THE DEVELOPMENT OF GDSS TO SUPPORT GROUP DECISION MAKING BY INCREASING THE PARTICIPATION OF THAI GRADUATE STUDENTS

By

JETSURA VONGVICHIEN

A Thesis Submitted in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy in Business Informatics
DPU International College
Dhurakij Pundit University
Year 2010

Abstract

The development of Group Decision Support System (GDSS) software with updated functionalities that may enhance group interaction in a face-to-face brainstorming mode is discussed in this research. Some of the functionalities include multi-language simultaneous input capabilities, including Thai, and the utilization of generic keyboard connections via USB. The developed software aims to facilitate, enhance and improve the level and efficiency of participatory learning of mixed groups of Thai and foreign students. This enhanced GDSS may also be used in a business group decision-making environment. Furthermore, to overcome the shy nature of Thai people, the option of non-identification of participant features is integrated into this GDSS. A pilot study has been conducted to exchange ideas, comments or solutions regarding assigned topics in an actual classroom setting / brainstorming session in order to investigate the learning effectiveness and the overall users' satisfaction with the software support. The enhanced software makes the designed GDSS more generic and suitable not only for graduate classroom environments but also for most types of group decision support approaches, such as consensus business decision making, strategic goal planning, particularly in multicultural and multilingual environments. The research outcomes may also lead to the identification of the preferred features of a GDSS facilitator in the future.

Keywords: Group Decision Support (GDS), Group Decision Support System (GDSS), Computer Mediated Collaboration, Computer-based Support System.

Acknowledgements

This dissertation would not have been possible without the guidance and the assistance of several individuals who in one way or another contributed and extended their valuable support until the completion of this study

First and foremost, my utmost appreciation to my principal supervisor, Professor Charles S. Newton, whose encouragement, guidance and support from the initial to the final level enabled me to continue with the project and develop an understanding of the subject

I would like to thank Dr. Adisorn Na Ubon, for his enthusiastic and unfailing support as my adviser

I am also grateful to my group of fellow PhD candidates at DPUIC, for moral support and experience sharing. It would have been much more difficult during the tough times without all of you.

Last but not the least I would like to thank my family and parents. I am wholeheartedly thankful for your patience, strength, and being by my side all the way.

Jetsura Vongvichien

Table of Contents

Table of Figures

List of Tables

Chapter 1

Introduction

This doctoral dissertation is primarily concerned with computer technology associated with group decision making and peer group learning. This research considers the integration of Group Decision Support Systems (GDSS) into the young Thai educational environment in order to contribute to the development of young Thais' thinking ability, to enhance group productivities, and to improve learning performance. Although GDSS is not a new field of information technology-based research, it is new when considering Thai education. This chapter explains: 1) the background and scope of the study, 2) the research objectives and questions, 3) expected contributions, 4) and the structure of the thesis.

1.1 Background and scope of the study

In an era of rapid growth in information technology (IT); advancements in newly developed electronic communication, the development of faster CPUs, recently invented peripheral devices, fourth generation computer languages and decision support have resulted in dramatic improvements in many areas including group interaction and decision making.

Since making the correct decision in any problematic situation or strategic planning is of crucial importance for any group of people, methods and techniques with the objective of giving support for decision making have received more attention. Organizations and institutions around the world search for and exploit sources of competitive advantage to generate cumulative knowledge and information

to improve decision quality and increase performance. Additionally, not only do they have to deal with their competitors, but they also have to deal with the changing and new demands from their dynamic environments (Huber, 1984).

One of the obvious technological advancements is the group decision support system (GDSS) or group support system (GSS) concept. The excitement about GDSS for both practitioners and researchers alike is the result of technologies that promise to add significant value to the users or groups of users that adopt them. As compared to group brainstorming using other tools and intervention mechanisms, it has been found that group consensus is higher in GDSS solutions (Quaddus & Tung, 2002). A key factor in the effectiveness of such a system is the manner in which users or decision makers are supported by applying this technology (DeSanctis & Gallupe, 1987). Because of the success and benefit of utilizing GDSS technology to support business groups, researchers have begun to explore ways to apply GSS in the learning arena. Various empirical studies indicate that GDSS supported collaborative learning significantly increases the level of student learning and 'groupthink' (e.g., Aiken, Martin, Vanjani & Sexton, 1994-1995; Leidner & Jarvenpaa, 1995; Kwok & Khalifa, 1998). GDSS can be used in a classroom setting to provide either "same time, same place" or "same time, different place" assisted by parallel communication, discussion, or brainstorming through supported features. Everyone in the GDSS session can talk or present ideas by typing into his or her message box, which the GDSS then immediately submits to the other participants. All participants have an equal chance to contribute. Further developments of GDSS functionalities lend themselves to a design science research approach.

GDSS can have a wide variety of tasks. It may simply seek to organize and communicate results to differently skilled groups of people to build a shared understanding because GDSS allows participants to focus better on the problem-solving situation with fewer distractions. It may be built on large databases or models or both. There are two basic types of GDSS.

1. Some are developed to be used as an adjunct to Face-to-Face discussion (FtF) in a discussion room (e.g., Osborn, 1957; Shaw 1976; Steiner, 1972; McGrath, 1984; McGrath, O'Connor, & Hollingshead, 1993) and

2. Some are constructed to support decision making in a distributed communications environment such as a computer conference (Pinsonneault & Kraemer, 1989; Johansen, 1988).

Various GDSSs have been developed, the purposes of which have been directed towards both commercial applications such as; Team Focus by IBM, Online collaboration for decision making (www.facilitate.com), and MindJet Maid Manager Pro (www.mindjet.com), and towards experimental activities, such as Group Storm at UNSW@ADFA, Australia, Plexsys System of Arizona University, USA, and Free Mind. Each category can be broken down into two sub-groups "GDSS" and "Others". The "GDSS" sub-group is the system that was mainly developed for practical GDSS usage while the "Others" sub-group comprises systems that use research platforms for further study into GDSS.

In addition to distributed GDSS developments, a number of GDSSs also try to integrate automatic language translation functionality into the system to overcome the language barrier. Language can have an important influence on the group decision-making process (Alderton & Frey, 1986; Meyers, Seibold, & Brashers, 1991;

Holtgraves & Lasky, 1999) and interaction (Bradac & Mulac, 1984; Bradac & Street, 1990; Adkins & Brashers, 1995). Adkins & Brashers (1995) pointed out that, for GDSS applications, language is an important subject because the information exchange process is conversational; a sender can only encode a textual message. Aiken, Martin, Paolillo, & Shirani (1994) stated the following:

> "Communication in multilingual groups is very difficult. Even if all participants in the group know a common language, it may be a first language for some of the group and a second language for others. Communication in such situations is not equal for all group members. (p. 155)

Through GDSS, relationships and discussion or interpersonal perceptions by group members are primarily based on the textual language. Kirkpatrick (1993) had shown that in face-to-face group meetings 20% of the people do 80% of the talking because some of the group members are shy, or too polite, the consequence of which leads to a lower rate of participation and a reduction of critical thinking. Supporting language manipulation in GDSS can therefore be a major factor of participative effectiveness. While English is the international and main language for major groupware, it has been recognized that different cultures operate in different dialects and expression. A question arises as to whether the available language of the application matters and whether it affects the decision or learning outcomes.

One solution for GDSS designers is the incorporation of language translators into the software (Aiken & Vanjani, 2009; Aiken, Park, & Lindblom, 2010; Aiken, Wang, Gu, & Paolillo, 2011). Language translation is a feature that assists participants in translating from one language to another when a word, phrase or

sentence is input. Although the translations are generally very good, translation errors can occur due to mis-spellings, poor grammar, slang, euphemisms, and idioms. The accuracy of automatic language translation limits successful implementation in commercial GDSS. In 2002, Aiken reported that despite medium level accuracy, understandability is at a very high level; "Several studies of mixed, English and Spanish speaking groups have shown that the Spanish to English grammatical accuracy has ranged from 45% to 74% and the understandability ranged from 85% to 94% (Ackermann, Morton, & Valerie, 2003; Aiken, 1993), Even though the grammar and word choices were sometimes incorrect; the meaning of the comment was still understood to a large extent. The English to Spanish grammatical accuracy was consistently close to 75% and the understandability was about 97%. In a study of groups with many English speakers and only a few Spanish spelling correctly, however, the understandability was 100%, and only 10% of the English speaking subjects were able to guess correctly which comments had been translated from Spanish (Aiken, Martin, Reithel, Shirani & Singleton, 1992). Though the understanding was claimed to be 100%, the understanding may be uncertain especially in critical meetings. The translation accuracy problem can lead to misinterpretation and may lead to improper responses in complex idea generation. To emphasize the accuracy concern, Kock (2007) identified that translations between some languages, such as German to French, are not as good as between others, such as English to German. The reasons for varying degrees of translatability are the peculiarities of different languages, for example, the frequent use of contractions in French, the adjective following the noun in Spanish, and the tonal differences of words in some languages that change the meaning or interpretation. In 2009, Aiken &

Vanjanim adverted the accuracies of Google Translation, a commercial online translation program. They examined reverse translations from 33 foreign languages to English with ten sentences from an electronic meeting. Understandable was evaluated by a native English speaker. The overall results were shown that 13% were verbatim, 73% were not verbatim but understood, and 14% were considered as not understood. Although, many of translations were 'understood', interestingly the results of 5 Asia languages gained higher percentage of dubious understanding. Only 4% of the reverse translations were verbatim, 70% of the reverse translations were not verbatim but understood, and 26% of the reverse translations were not understood.

The process of translation from an Asian language to European language is complex. The complexities are from the characters, idiom, jargon, and word specific meaning. Duff (1989) cited that "idiomatic expressions are notoriously untranslatable" (p. 11). Explicit translations are not always equivalent (Kock, 2007). This may be one of the reasons why GDSS technology is not widely utilized in the Thai educational environment. Supporting multiple languages has therefore, become a possible concern in GDSS design and development.

Wealthy Asian families often send their children to study abroad to give their children the opportunity to master a foreign language and also to gain more confidence and learn a new way of thinking through a participative educational style (Teerakul, 2001). This allows the development of important skills for today's business environment. Traditionally, Thai children are taught to pay respect to seniors and also decision making is not delegated to junior levels in organizations and private environments (Thanasankit, 2002) both of which complicate the issue. Young Thais are not supposed to argue and push their own ideas with their parents, teachers, or

supervisors and this may restrict the ability of teachers or employers to identify gaps in their knowledge. Since this mindset has remained largely unchanged for many generations, it is not easy to change.

Thai culture has hence become a major obstacle to the learning ability of young Thais who undertake their education locally and has also influenced the way Thai people think, tackle problems, and make decisions (Rohitratana, 1998). For instance, because some group members are too shy or polite, there is a decline in participation, thereby reducing critical thinking and the generation of creative ideas. One possible method for making the change can start in the provision of educational facilities, especially at the university level.

1.2 Research objectives and questions

Given the importance of group decision support systems (GDSS), it is worthwhile investigating further the language supportive concept to complement existing research. This doctoral study has two main research objectives. First, it assesses and analyzes to develop a GDSS artifact with a multi-language aiding tool relying on the advancement of technologies to explain the antecedents and consequences of the system on the international academic students' learning process and outcomes including improvements in participation and contribution. Second, it intends to test the system empirically in a non-Western setting, which will help the researcher to examine the external validity of Western-developed theories (i.e. the applicability of theories in other contexts). This includes the re-evaluation of the dimensionality of constructs, and the assessment of certain assumptions associated with findings reported in extant literature (e.g. the connection between GSS and a

human context). In particular, this study seeks to answer the following research question:

Research Question:

Can a generic international group of academic students improve their group learning performances and participation, and be satisfied with the classroom atmosphere through the use of multi-language supportive GSS?

Learning performance can be analyzed in terms of the quality, efficiency, and effectiveness of a classroom discussion process and outcome, while classroom atmosphere can be analyzed in terms of participation rates and the perception of students on electronic classroom experience, such as satisfaction with the process and outcomes.

1.3 Contribution of this study

Overall, this study contributes three areas of knowledge. First, supported by an extensive literature review, this work describes the challenges for the situational evaluation of a group decision support system (GDSS) or group support system (GSS) for multinational culture and language classroom environments. Second, elucidation of how the study methods can be applied to overcome some of the challenges is sought. In particular, design science research methodology has been shown to help establish a set of requirements for designing and building a GSS tool. Various factors are examined to determine the essential GSS features which support classroom learning, and educators' participation, and satisfaction. Finally, the exploration of the effects of application design and advancement of technology-support on the examined subject is derived through prototyping, and through lab and observation studies.

In summary, by achieving the aforementioned objectives, the two main contributions of this thesis are:

- Extending meaningfully the utilization of GDSS in the international educational field and on Thai culture to offer practical insights for Asian academics.
- Developing an experimental new GSS artifact – BrainEx as a technology-based tool for hypothesis testing of multi-language-support GSS.

Design Science, Positivism, and Interpretive methods were used to develop the artifact and test the participants' performance and satisfaction in collaboration with a classroom teacher or a facilitator.

1.4 Thesis structure

This dissertation is structured in six chapters as follows:

- Chapter 1 Introduction – This chapter presents the rationale of this research, which goals can be achieved, why this dissertation presents new research contributions, and what the major contributions of this dissertation are.
- Chapter 2 GDSS Evolution – This chapter reviews and investigates existing GDSS literature on antecedents, concepts, and consequences of group (decision) support systems. It begins with a discussion of foundation, definitions and characteristics of G(D)SS. Studies on GDSS evolution and model antecedents and consequences from different research disciplines are then presented. The chapter concludes by identifying gaps in existing research and outlining the possible uses of the GDSS concept with target groups identified in this dissertation.
- Chapter 3 Research Methodology and Method – Research methodology techniques employed in the study are discussed. A brief review of

research approaches in the information system research area is first presented. The overview of design research, positivist and interpretivist research designs including the research setting and data collection procedures are then elaborated upon.

- Chapter 4 Conceptual Framework and Hypotheses – Here the conceptual model of the research is described and the development of hypotheses is explained. The linkages between the designed system and variables from both concepts are explained. Finally, the impact of the system on studied subjects is explained and hypotheses are presented.

- Chapter 5 Research Findings – The techniques selected to test the developed artifact and hypotheses are presented. It describes the pilot and laboratory experimental testing, and the related underlying assumptions are highlighted and described. Observation of qualitative data and the use of questionnaire surveys involving quantitative data collection methods have been used to contribute to the findings from testing studies. The preliminary results of measurement scale development using literature searches, focus groups, and analyses of the questionnaires are shown. The chapter concludes by showing the outcomes of the hypotheses testing.

- Chapter 6 Conclusion – Several conclusions are drawn from the research studies and outcomes in respect to the research question. The limitations of the research and recommendations for further research are also presented.

Chapter 2

GDSS Evolution

2.1 Introduction

With the recent advancement of technologies, such as electronic board-rooms, local area networks, programming language software, hardware and communication devices, together with the increasing need for group decision making, meetings and new learning initiatives, a growing interest has emerged in technological support for groups as shown through increased activity in the Group Support System (GSS) or Group Decision Support System (GDSS) research areas. In this chapter, different kinds of GDSS are introduced and the objective of the thesis is identified. The chapter begins with a discussion regarding the foundation, and the different definitions and characteristics of GDSS. Experimental GDSS studies are reviewed, which will lead to research gaps being identified and discussed.

2.2 Foundation

Over a quarter of a century ago, team-based organizations have been a principle characteristic of work. High-level coordinating groups have an important role within organizations (Mintzberg, 1979). Management discussion and group projects have become a part of everyday life in the business world. Meetings have become vital media for management and communication because the proper running of a meeting can save time, create new ideas, increase motivation, productivity, and solve problems. Nunamaker, Briggs, Mittleman, Vogel, & Balthazard (1997) cited research by Panko (1994) that twenty percent of managers' time was spent in formal

meetings and up to eighty-five percent was for communication. Later in 1996, Pollard and Hayne reported that companies hold team meetings as their main coordination and communication activities. While the percentage of group meetings has increased over time, the research of Robert & Economy (1995) has shown that over 50 percent of this meeting time is wasted. Similarly, in 2001, Romano and Nunamaker reported that the amount of unproductive time in meetings has been estimated to be between 11 percent and 73 percent. It is not therefore surprising that many researchers have tried to find alternative solutions to the unproductive meeting situation or to achieve meeting purposes. An attractive approach pursued by researchers is the implementation of technology-based tools. Computer-supported systems are integrated into the operation of most organizations (Aiken *et al.,* 1995; Nunamaker *et al.*, 1997) to assist work groups in completing their tasks and improving their performance.

Technology is defined as the "practical implementations of intelligence" (Ferre, 1988, p.26). It includes tools (e.g., electronic messaging systems, collaborative systems, and group (decision) support systems), materials (e.g., computers, software, LCD projectors, and peripheral devices), and communication media (e.g., local networking, and Internet). An important result of the rapid growth of computer technology; such as minicomputers, timeshare operating systems, and distributed computing, is the support provided to decision makers. In the early 1960s, many organizations began to computerize operational business. Information systems were implemented to perform and improve organizational processing and management. Collaboration through such technologies, which allows meeting participants to share and exchange data, to save information generated during their

meetings automatically and to support problem solving in group meetings, has been developing and has captured researchers' attention. A well-known technology-supporting concept for group work and decision making is Group Decision Support Systems (GDSS).

GDSS has been extended from the scope of decision support systems (DSS). The study of Eom (1999) identified that GDSS has become a central part of DSS since 1991. The place of GSS in DSS evolution is shown in Figure 2.1. Over the last 30 years, improvements in the decision-making process have enabled productive actions, prompt problem solving, and better performance. DSS theory developments began in the early 1970s through Michael S. Scott Morton (Sprague, 1980; Arnott & Pervan, 2005). Over the past three decades, the DSS term has been defined differently by different authors. They perceive the DSS field from various perspectives, and there is some disagreement about what is really meant by the term and what is important. From time to time, new entrants to DSS research have adjusted and extended the definition. A widely accepted definition of DSS, proposed by Sprague (1980), is

> "DSS comprise a class of information systems that draws on transaction processing systems and interacts with the other parts of the overall information system to support the decision-making activities of managers and other knowledge workers in organizations". (p.6)

Some authors define DSS as an interactive computer-based system that is intended to help individual decision makers solve complex or unstructured problems and make decisions through direct interactions with analytical models and data

(Bonczek, Holsapple, & Whinston, 1981; Huber, 1984). DSS includes any system that supports a decision or makes some contribution towards decision making. It involves model support. The mission of DSS was established to cover the creation of approaches for utilizing information system technology to increase the decision makers' effectiveness in situations where the computer system can assist and enhance human judgment in task performance with elements which cannot be specified ahead (Holsapple & Whinston, 1996) and can help reduce cognitive effort (Gallupe & McKeen, 1990).

Research in this area normally focuses on how information technology can improve decision efficiency and effectiveness (Shim, Warkentin, Courtney, Power, Sharda, & Carlsson, 2002). Two main DSS research areas are discussed: firstly, the theoretical studies of organizational decision making and technical work (Keen & Morton, 1978) and secondly, major organizational decision making research was conducted during the late 1950s and early 1960s at Carnegie Institute of Technology and significant technical work was achieved in 1960s at MIT (Keen & Morton, 1978). DSS are typically developed for individual stand-alone computers. It is comprised of (i) a large number of data or sophisticated database management capabilities with access to internal and external data, (ii) powerful modeling functions accessed by a model management system, and (iii) user interface design which enables functions of queries, reporting, and graphing (Shim *et al.*, 2002). With the evolution of computing, DSS has expanded the scope from personal use to group level. GDSS or GSS, providing a structure and automated interconnected network of computers or communication facility, allows workgroups to brainstorm, and to share complex decision-making information more easily in order to accomplish a collective set of

tasks. Group members in a face-to-face (FtF) meeting may be seated in a semi-circle or U-shape seating arrangement in a workspace or in distributed meetings where participants in different locations interact or work together.

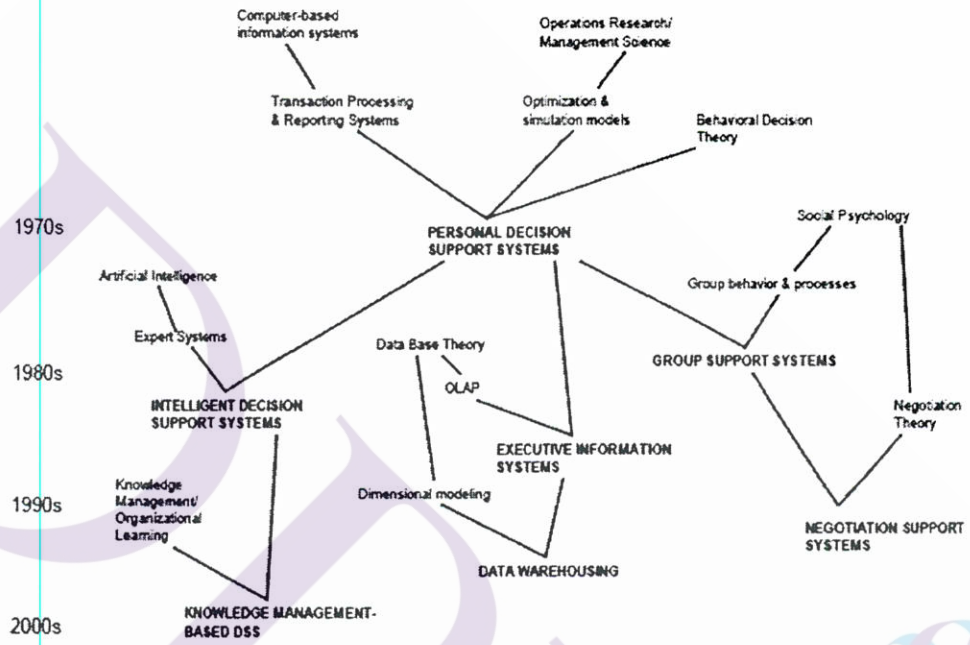


Figure 2.1 The Evolution of the Decision Support Systems Field (Arnott & Pervan, 2005)

2.3 Group Decision Support Systems: Definition and Related Terms

A typical GDSS consists of a meeting room with computer terminals and decision technologies, and input is by keyboard, mouse, or combination of these devices to support problem finding, formulation and problem solution by groups or teams. Participants can create displays on their terminals or on a large public display screen (Huber, 1984). GDSS involves the provision of electronic brainstorming, idea generation, and communication-support features, in addition to simple computer messaging, to exchange ideas, opinions, and preferences. DeSanctis & Gallupe

(1987) defined a GDSS as a system "which combines communication, computers, and technologies to support problem formulation and solutions in group meetings" (p.589).

Huber (1984) provides another definition. A Group Decision Support System (GDSS) is a computer based information system that enhances group decision making by facilitating the exchange and use of information by group members, and interactions between the group and the computer, to formulate and solve unstructured problems.

However, GDSS has developed beyond its original impact on decision making. An additional term is identified as "a computer-based system that supports groups of people engaged in a common task and that provides an interface to a shared environment" (Aiken *et al.*, 1994, p.157). As a result, GDSS is also called a group support system (GSS) (Shim *et al.*, 2002) which focuses on group communication support.

Nunamaker (1995) concluded in the "Electronic Meeting System: Ten Years of Lessons Learned" paper that a GSS is a collection of different software tools, each of which focuses and structures group thinking, and affects group dynamics in a unique manner.

Most GDSS or GSS (the terms are used depending upon the specific study) applications are designed and aimed to improve the productivity and effectiveness of the group *(decision making)* process by removing communication barriers, exchanging ideas and opinions (brainstorming), and providing structured decision analysis. They focus on *modeling* support of group problem solving, or group task performance. Several GDSS research papers have formed the foundation for

subsequent GDSS empirical research. They proposed architectures of GDSS, and multidimensional practices of GDSS which impact on decision quality, and meeting or activities outcomes. An experiment by Nunamaker, Dennis, & Valacich (1990a) studied the effect of anonymity and critical tone on outcomes. The study also cited that group size has also had impacts upon outcomes. Various computer systems were employed to deal with the structured parts of the DSS problem (such as decision modeling methods, and rules for group discussion), but the unstructured parts were managed by decision-maker judgment.

2.4 Group Decision Making: Definition

One of the accepted decision making processes that influences many research works (see, for example, Sprague & Watson, 1979 & 1996; Gray, Vogel, & Beauclair.,1990; Kersten, 1997) is the decision making model by Simon (1960). Simon distinguishes three primary phases of decision making process; intelligence, design and choice. The intelligence phase involves the environmental investigation and problem recognition. Besides identifying the problem, classification of the problems might also need to be decided upon within this phase. Decomposition of complex or large problems into sub-problems sometimes reduces the difficulty in obtaining solutions. The result of the intelligence phase is a problem statement. It is defined in order to understand a problematic situation about which actions need to be taken as the starting point of the design phase. In the design phase, alternative solutions are created and then models are developed to analyze those alternatives. These alternatives are then evaluated by predicting the outcomes. The next phase is the choice phase where one of the different alternatives is selected to solve the defined

problem identified in the intelligence phase. Later a fourth phase was added called the implementation phase. This phase describes the implementation of the preferred solution selected during the choice phase. According to Turban & Aronsson (1998), implementation can be defined as "putting a recommended solution to work" (p.59).

Generally, a decision-making process is not an individual activity. A group of decision makers can attempt to generate relevant alternatives. A group decision making process is defined by DeSanctis & Gallupe (1987) as follows:

> "Two or more people who are jointly responsible for detecting a problem, elaborating on the nature of the problem, generating possible solutions, evaluating potential solutions, or formulating strategies for implementing solutions. ... The members of the group are aware of one another and perceive themselves to be a part of the group which is making the decision." (p. 89)

A group decision-making activity involves exchanging personal opinions within a group in order to proceed with problem solving. The successful or unsuccessful outcomes of the group meeting on decision-making depend on many factors.

Kelley & Thibaut (1969) stated that several authors have indicated that the *quality of information available to a group* is one of the most important determinants of successful group decision making. Another factor that influences the group decision-making performance is the *quality of effort* by the group members. For instance, Janis & Mann (1977) found that groups are more likely to reach high-quality decisions when their decision-making processes are carefully characterized and thoroughly examined. The third factor that impacts on group decision-making

performance is the *quality of thinking*. Studies have reported that participants' ability to deliver appropriate thoughtful conclusions from available decision-relevant information to them affected the decision quality (Hirokawa & Poole, 1996; Hew & Cheung, 2003).

Although it is clear that various factors affect the group decision-making, articles of Hirokawa & Pace (1983), Hackman & Morris (1975), and McGrath (1984) suggest that the communication quality is also an important influence on the decision-making effectiveness and accomplishment. In this sense, the communication technology of GDSS is another interesting point that has a dramatic impact on the decision-making process and presumably on the decision outcomes.

2.5 The study of group (decision) support systems

The studies of interdisciplinary GDSS are not new. Experiments on GDSS have been presented since the mid-1980s. More than 200 research articles on GDSS or GSS have been identified and analyzed. GDSS has emerged and received heightened attention as computers and communication technologies have become common tools or equipment in the workplace, home, and classroom and have created new opportunities for users. An additional major reason that has driven research efforts is the benefit of participation within a group brought by the system, such as decision quality, productivity improvement, and increasing degree of participation and learning.

At an early stage, most Group Decision Support System (GDSS) research initiatives focused on 'decision rooms' and were limited to organizational contexts relevant to management decision-making support (e.g., Pinsonneault & Kraemer,

1989; Benbasat & Nault, 1990; Dennis & Gallupe, 1993). The system was seen as a strategic management tool, involving electronic brainstorming, idea consolidation, and scenario finalization. It has been expanded continuously into the areas of information science (Keen & Scott Morton, 1987; Goodman, 1986), management science, computer science, and social psychology (Lewicki, Weiss, & Lewin, 1992; Simon, 1997; Eom, 1999). Existing GSS support functions include idea gathering or brainstorming, rating, ranking, voting, analysis, allocating models, and grouping (Dennis, George, Jessup, Nunamaker, & Vogel, 1988). Topics studied included the impact of computer-based systems and environment on group decision quality; problem solving effectiveness and efficiency and satisfaction (DeSanctis & Gallupe, 1987; Hiltz, Johnson, & Turoff, 1991; Tyran & Shepherd, 1998); the influences on behavior of GSS on group decision-making (Zigurs, Poole, & DeSanctis, 1988); the effect of group sizes on electronic group decision making outcomes (Gallupe, Dennis, Cooper, Valacich, Bastianutti, & Nunamaker, 1992; Dennis, Valacich, & Nunamaker, 1990); the influence of facilitative support on group meeting effectiveness (Dickson, Robinson,Heath, & Lee, 1988; McGoff & Ambrose, 1991; Dickson, Partridge, & Robinson, 1993) and the impact of anonymity on idea sharing (Jessup *et al.*, 1990; Kwok & Khalifa, 1998). Each of these experiments has demonstrated the performance advantages of using group decision-making software.

However, the impact of computer-based systems has been shown to depend on various factors such as the stage of computerization, the particular task involved, the existing organizational structure, and the amount of user experience with the system or type of system used (Attewell & Rule, 1984; Rice, 1980). Like a jigsaw, GDSS is a complex subject which cannot be understood by reviewing only one area.

Rigopoulos, Psarras, & Askounis (2008) stated,"...building an appropriate Group Decision Support System (GDSS) is not a straightforward process" (p. 407).

The tools have evolved in several directions, such as the design of the system, the group size (small or large), the type of task, and the communication mode (same place or different place or dispersed), all of which influence the research outcomes. Empirical investigations on the critical factors leading to the success of systems and also the usage and features of GDSS have been reported by Gallupe (1985), Quinn, Rohrbaugh, & McGrath (1985), DeSanctis & Gullupe, 1987, Gallupe *at al.* (1992), Kwok & Khalifa (1998), Chen, Hsu, Orwig, Hoopes, & Nunamaker (1994), Ackermann & Eden (2001), and others. For instance, Ackermann, (1996) stated that the success of GDSS depends on the quality of the facilitator. Jessup *et al.* (1990) investigated the concept of anonymity of GDSS when conducting idea generating tasks. Hence, when developing a GDSS research investigation, the purpose, characteristics and environment of a system are considered as crucial steps.

2.6 GDSS Characteristics

Even though group technology support research findings have appeared in literature since 1982 (Fjermestad & Hiltz, 1998), there is still no general taxonomy categorization. One of the most popular papers written by Johansen, Martin, Mittman, Saffo, Gibbet, & Benson (1991) provides the foundation of a group meeting matrix (see Table 2.1). Four conditions of communication modes are discussed; same time and same place (synchronous), same time but different place, different time but same place (asynchronous), and different time and different place. Table 2.1, adapted from Aiken, Mahesh, & James (1995), presents a brief illustration of these four modes

of GDSS environmental settings for different sizes of group.

Table 2.1 Four types of GDSS Environmental Setting, Aiken *et al.*, 1995

Meeting Type	Setting
1.Decision Room	A small group meeting in a same time & same place to address a set of problems in a Face-to-Face (FtF) configuration.
2.Local Area Decision Network / Synchronously	Small groups meeting at the same time whose members are dispersed in limited geographical areas or different places. Electronic or human facilitators may participate to guide the session in accordance with a predefined agenda.
3.Project Room / Asynchronously	A large group meeting which is too large for a Decision Room. The setting can be applied for same place, different time situations.
4.Computer-Mediated Conference	A large group meeting where the members are dispersed geographically. The setting can be applied both in same time & different place and in different time & different place configurations.

Selecting any of these modes involves exchanging messages pertaining to preferences and values. As information technologies are utilized to overcome space and time constraints. Different categorizations of GSS including GDSS, Project Room, computer-mediated communication system (CMCs), and Distributed Group

Support System (DGSS) have been established regarding their modes of support (see Figure 2.2).

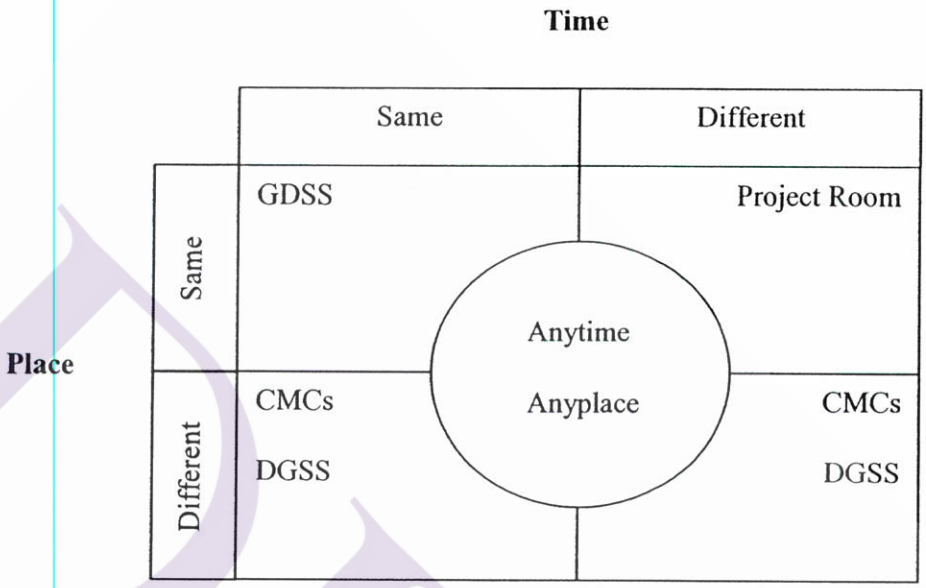


Figure 2.2 The modes of GSS, Johensen *et al.*, 1991

The main focus of GDSS studies, since the mid-twentieth century, has been on the small group Face-to-Face (FtF) mode or same time same place condition (Hiltz *et al.*, 1991). Face-to-Face electronic meetings usually take place in a specially equipped computer laboratory where everyone can talk in parallel or in sequence depending on the application design. Several of them have examined the effects of automated groups compared with non-automated groups on the group task performance (see for example, Kull, 1982; Turoff & Hiltz, 1982; Siegel, Dubrovsky, Kiesler, & McGuire, 1986; Jarvenpaa, Rao, & Huber, 1988). CMCS are frequently used for text-based communication and focused on different time or place environments. The most common forms are electronic mail, and bulletin board systems. While GDSS is widely used as a means for distributed study and occurs or when participants are geographically or temporally separated. Users may access to

the GDSS to do the real time or same time brainstorming (e.g. video conference) or to do the off-line or different time ideas posting.

However, group meeting conditions are not the only focus point of GDSS researchers. There are some other important distinctions, variables and factors that have been studied in a variety of GDSS researches to analyze the improvement and performance of group learning and/or decision making, and to remove or decrease barriers to idea and information generation resulting from applying group support technology. A number of research models have been proposed for the study of GDSS and group effectiveness (DeSanctis & Gallupe, 1987; DeSanctis & Poole, 1994; Jessup & Valacich, 1993; Nunamaker *et al.*, 1991b; Pinsonneault & Kraemer, 1989). These models generally take an input-process-output approach.

Based on the *input-process-output model* of 40 years ago (McGrath, 1984; Hackman & Morries, 1978; Jessup & Valacich 1993), the typical input variables of the model are technology (hardware, and software included media), group traits (e.g. size, facilitator behavior, national culture, etc.), the nature of the task (e.g. type of task, degree of task structure, and difficulty), and the context (e.g., culture and behavior norms, decision-making time frame) (Gallupe, 1992; Fjermestad & Hiltz, 1997; Kwok & Khalifa, 1998), which are expected to enhance the group decision-making process and outcome. The process represents either behaviors during the meeting (e.g. patterns of participation) or more fleeting characteristics of the group (e.g., cohesion); and outputs include various measures of group effectiveness.

From time to time, GSS technologies and researches, as information systems research, have been expanded to investigate and examine the different ways in which GSS technologies are used and differences in their effectiveness. They are

conducted in a multi-disciplinary and multi-cultural context. A number of frameworks have been identified consisting of a mixture of characteristics of the group. For example, Osborn (1957) noted that groups can improve their meeting process and outcomes with certain structures, such as strict agenda that enforce rational brainstorming. Much research has focused on how anonymity of input affects GDSS interactions (e.g. Connolly, Jessup, & Valacich, 1990a & 1990b; Jessup *et al.*, 1990; Jessup & Tansik, 1991). Numerous GDSS studies found mixed support for the claim that anonymity increases group performance (Dennis & Gallupe, 1993). Valacich, Jessup, Denis, & Nunamaker (1992) reported anonymity can improve idea generation within the GDSS environment. The process variables can be independent or dependent variables, such as communication characteristics and the structure imposed by technology or the groups experience from the process construction.

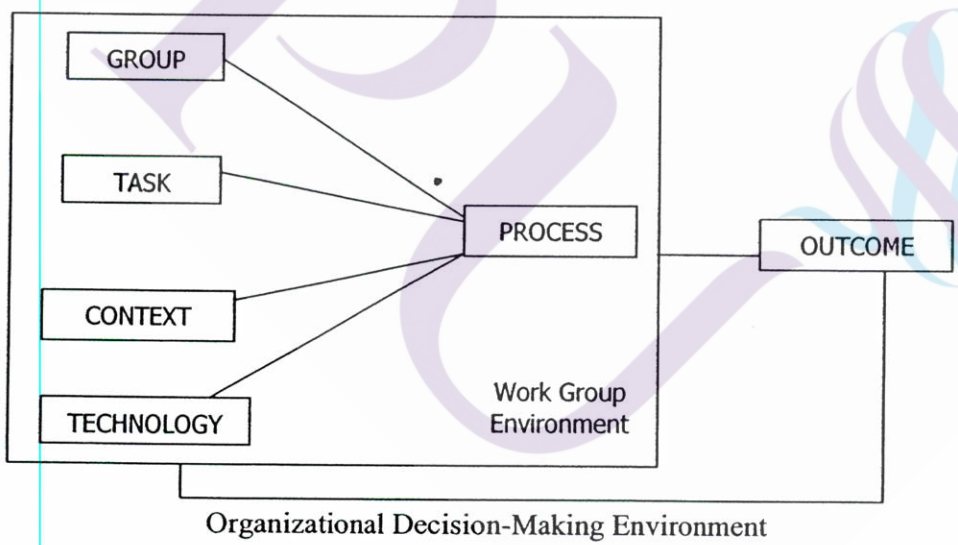


Figure 2.3 Input-process-output model of a group work

2.6.1 Four general types of input

2.6.1.1 Technology: Under the subject of technology, it touches on various types of existing GDSS research (Fjermestad & Hiltz, 1997).

Task Support - Tools are based on a computer-based system. The brainstorming technique (more than 35 studies) is the most frequently used tool for generating ideas and opinions of group members who are specialist in different areas for solving or finding a solution to a specific problem followed by voting or preference rating.

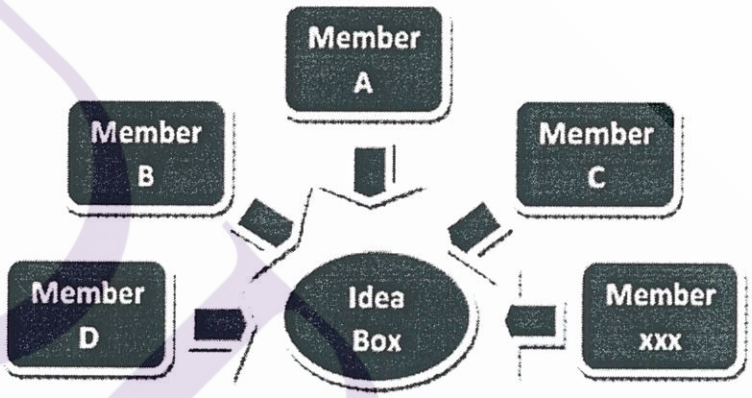


Figure 2.4 Brainstorming concept

Process Support - Technology which generally has been classified into three primary types: DSS, GDSS, or CMC. A DSS has been developed to support an individual decision maker. A GDSS has been developed for a decision room or synchronous (same time) condition. The GDSS is composed of structured tools to assist decision making (e.g., voting tools). CMC has been developed focusing on group discussion which may or may not include GDSS tools. Using a GDSS has been the focus of the majority of studies.

Process Structure – this is about the "level" of sophistication in terms of GDSS features. DeSanctis & Gallupe (1987, p. 593-594) have classified

GDSS into three levels, each with a high level of interference on the group exchange and with increasing corresponding features of the system. Level 1, provides technical features aimed at removing common communication barriers, such as large screens for instantaneous display of ideas, voting solicitation and compilation, anonymous input of ideas and preferences, and electronic message exchange between members. Normally, this level is referred to as "a computer-supported conference room" or "an electronic boardroom" (DeSanctis & Gallupe, 1987) in which participants can write and display their planning, ideas, and selections. Most GDSS studies use this level (Fjermestad & Hiltz, 1997). Level 2 or the second level of sophistication "provides decision modeling and group decision making techniques aimed at reducing uncertainty and 'noise' that occur in the group's decision process". The Delphi method is a general technique used at this level. The most advanced level, the third level, is "characterized by machine-induced group communication patterns and can include expert advice in the selection and arrangement of rules to be applied during a meeting".

2.6.2 Task: "The general variable group task type' is emerging as an especially important variable, often accounting for as much as 50% of the variance in group performance". Poole, Seibold, & McPhee (1985, p. 88)

The task influences the group's motivation to accomplish the objective (Guzzo, 1986). It is an important determinant of the needs for information exchange and the consequent communication practices of the decision-making group. In some GDSS literature the task is among the most important influencing factors explaining the effects of using GSS. Some categorization schemes have been proposed in the literature. For example, the study by McGrath (1984) which is one of the most widely

cited works, identifies eight types of task in four categories. Tasks are categorized in accordance with what the group must accomplish during the meeting. Fjermestad and Hiltz (1998) state in their research that task type does reflect GSS use:

> "Adding task type as an additional control variable, we observe more positive results when Computer-Mediated Communication (CMC) systems (technology-support in the different place / same-time and different place / different time) with task type 4 (decision making), and GSS (same-time / same place) with task type 2 (idea generation), are compared to the face-to-face condition." (p. 6)

The complexity of the task is another factor. The decisions made for highly complex tasks were better with GDSS compared with FtF (Bui and Sivasankaran, 1990). From the studies of Simon (1973), tasks could be compared in terms of their level of structure. Highly ill-structured tasks will be complex, difficult, and involve a relatively high number of preference issues. In contrast, well-structured tasks are less complex, easier, and have a relatively low number of preference issues.

2.6.3 Group

Not one of the definitions of GDSS mentions anything about the *size of the group* of decision makers. Since the use of a GDSS never was intended for a single individual decision maker, group size becomes a concerning attribute for many researchers (Dennis, 1990; Gallupe, 1992; Tyran & Shepherd, 1998). Group size has been argued to have an impact on the GDSS outcome (Nunamaker *et al.*, 1991; Dennis *et al.*, 1988). Nunamaker & Deokar (2008) cited researches by Fjermestad & Hiltz (1998) and Dennis & Wixom (2002) that larger groups can positively affect or

outperform more than smaller groups in GDSS environments. However, the primary choice of the experiments reviewed utilized small to medium-size groups - 6 to 12 group members followed by four group members. There are very few studies that used groups larger than 12. This can be because a GSS system itself may not be able to scale to larger groups. With very large groups, the volume of information that users must process grows dramatically and is challenging to manage effectively. For example, Hoffmann (1979) found larger groups have more difficulty reaching a consensus. This also lead to one important factor which lies within the group size factor. The task gain or loss and differentiation can depend upon the group size. Nevertheless, it is not clear what the suitable size is for a learning environment using GSS (Tyran & Shepherd, 1998).

2.6.4 Context

Only a few context variables have been studied. In the GDSS field, facilitator context is a crucial resource considered in computer support decision-making group studies (e.g., Dickson *et al.*, 1988; McGoff & Ambrose, 1991; Clawson, Bostrom, & Anson, 1993; Antunes & Ho, 2001). These studies focus on the role of facilitation and measure the effectiveness of groups using facilitators (Pinsonneault & Kreamer, 1991; Dennis & Garfield, 2003). Group facilitation is a process in which a person who is acceptable to all the members of a group intervenes to help improve the way the group identifies and solves problems and makes decisions (Schwarz 1994). The facilitator can encourage an open and anonymous debate of decision alternatives.

Many empirical researches report on the positive outcomes of integrating facilitation into collaborative technology environments (Ackermann, 1996; Dickson *et*

al., 1993). The two main benefits of having a facilitator to assist in the use of a GDSS are that the facilitator eases the group into the use of unfamiliar technology and provides a good process or guideline for the group (Dickson *et al.*, 1993). According to Nunamaker *et al.*, (1997) a GDSS facilitator executes four functions:

(1) Provides technical support by initiating and terminating specific software tools;

(2) Chairs the meeting, maintaining and updating the agenda;

(3) Assists in agenda planning; and

(4) Provides organizational continuity, setting rules and maintaining an organizational repository.

As such, the facilitator serves to help the participants to use the GSS appropriately and impels the participants to be more participative along the process and finally reaching agreed idea outcomes.

Likewise, several empirical studies show that under the same controlling factor(s), the way in which a GSS effect works is not always consistent across the group in terms of idea generation (e.g., Markus, 1992; Zack & McKenney, 1995). The impact may depend on the team's social structure. Culture or group behavior norms therefore have become another significant factor of GDSS study. To date, a multicultural workforce is a reality (Adler, 1991). An individual's cultural values may influence his/her perception of information and impact decision making (Adler & Ghadar, 1989; Hofstede, 1984). In face-to-face group meetings, 20 percent of people do 80 percent of the talking. The reason is that some group members are too shy or polite to participate (Kirkpatrick, 1992). This leads to low levels of participation and a decrease in critical thinking. Many GSS studies concern the relationship between

culture and process outcomes. GSS has been used to improve the balance of group process gain and loss, where "process gain" refers to synergy effects and "process loss" refers to reductions in group performance due to inefficiencies with the group process (Nunamaker, 1993). Various GSS have been designed with support features aimed at addressing and solving the sources of group process loss, such as the cultural behavior of participants, including how to make social status in groups become a less dominating factor so that all participants can feel they have equal opportunities to contribute and participate.

When culture is discussed, obviously language content is also investigated. It is accepted that language is a part of culture. Cited by Jiang (2000, p. 328) "…without language, culture would not be possible". Brown (1994, p. 165) describes culture and language as follows: "a language is a part of a culture and culture is a part of a language; the two are intricately interwoven so that one cannot separate the two without losing the significance of either language or culture." Jiang also presents one metaphor of language and culture from a communicative view (p. 329): "Communication is swimming, language is the swimming skill, and culture is the water. Without language, communication would remain very limited (in very shallow water); without culture, there would be no communication at all."

Clearly, communication in multilingual groups is the direction of current business and education trends. The increasing numbers of international meetings and classroom programs involving multiple language groups who do not share a common language lead to communication difficulties and problems. Even if all group members know a common language, which may be a first language for some members and a second language for other members, group communication is still not

equal for all participants. GDSS studies have demonstrated various solutions to overcome this lingual barrier especially in multilingual group communication (e.g., Aiken *et al.*, 1994).

This study has therefore been developed with the belief that with appropriate matching among participants, tasks, context and group support technology, GDSS would be more successful and some benefits can be gained.

2.7 Sample of GDSS types

The analysis of the work by Aiken *et al.* (1995) and of the foundation of DSS shows that research has been mainly dedicated to four areas: Personal Decision Support Systems (PDSS), Group Support Systems (GSS), Intelligent Decision Support Systems (IDSS) and large data-centered systems (e.g., EIS and data warehouse). In these areas, a number of sub-types have been developed. The study by Arnott and Pervan (2005) to review and analyze 1,020 DSS articles published in 14 major journals from 1990 to 2003 identified eight different types of DSS; PDSS, GSS, EIS (includes BI), Data warehouse, Intelligent DSS, Knowledge management-based DSS, and Negotiation support systems. They found that PDSS primarily and GSS secondarily, dominated research activities with 35.3% and 29.2% respectively.

2.8 Gaps in GDSS / GSS research

The review of literature reveals several directions for GDSS design and development. The effectiveness and performance of the system may be influenced by various types of constituent. Hirokowa and Poole (1996) expressed that a *decision is a social product of groups and the communication process* and it is the

communication between group members that helps them work out both the form and content of decisions. Over time, a set of social structures or culture emerges to influence interactions between groups of people. Cultural differences between countries may be evident from one part to another in (multinational) organizations (Hofstede, 1984; Thanasankit, 1999).

2.8.1 Implications of National Culture on G(D)SS

National culture has been defined by many researchers. Kluckholn (1962) defines culture as consisting of "patterns, explicit, and implicit, of and for behavior acquired and transmitted by symbols, constituting the distinctive achievement of human groups, including their embodiments in artifacts" (p.73). Hofstede (1997) defines culture as "the collective programming of the mind which distinguishes the members of one group or category of people from another" (p.5). "Mental programming … patterns of thinking and feeling and potential acting" (Hofstede 1991a, p. 4). Dennis, Tyran, Vogel, & Nunamaker (1997) states that culture depicts the way members of a group relate to each other and to the environment. Various dimensions of culture have been examined in the literature. The purpose of the dimensions is "to add some structure to a mass of cultural information that otherwise is too complex to grasp" (Hodgetts, 1993, p.54). There are four basic dimensions of national culture adopted by Blanning & Reinig (2002), whereas Hofstede (1991), the most widely cited researcher in previous papers, had suggested five dimensions.

2.8.1.1 Five culture dimensions

1. Power distance refers to the perceived level of dominance of one group over another. Defined by Hofstede *et al.*, (2010), power distance is the

extent to which society accepts and expects the fact that power in institutions and organizations across countries and cultures is unevenly distributed. A "high power distance" society is characterized when it emphasizes the gap between junior and superior, for example teacher over student (Hofstede, 1997) and societies of most Asian countries (Hofstede *et al.*, 2010).

2. Individualism-Collectivism measures whether people prefer to work alone or in groups. Individualism indicates a preference for a loose-knit social framework in society in which individuals are only supposed to take care of themselves and their immediate families; as opposed to collectivism, which implies a preference for a tightly knit social framework in which individuals can expect their relatives and clan to protect them in exchange for loyalty. For a collectivist country, the perception that the will of the group is seen as stronger than the will of individual predominates. In the international affairs, associated with the individualism-collectivism dimension, difference in values will continue to exist and to play a big role (Hofstede *et al.*, 2010)

3. Uncertainty avoidance is defined as "the extent to which the members of a culture feel threatened by uncertain or unknown situations" (Hofstede, 1991, p.113; Hofstede *et al.*, 2010, p.191) which leads them to support beliefs promising certainty and to maintain institutions protecting conformity. This dimension is based upon a degree of a society's discomfort with ambiguity and the pursuit of stabilizing activities. Hofstede (1998) cites Dawar, Parker, & Price's study (1996) which states that uncertainty-avoiding cultures tend to be more expressive and are distrustful of new ideas.

4. Masculinity indicates a preference for achievement, heroism, assertiveness and material success; as opposed to femininity which implies a preference for relationships, modesty, caring for the weak and the quality of life.

5. This fifth dimension was Confucian work dynamic developed by Michael Bond who attempted to identify Chinese culture values and their impact on the workspace (Mullin, 2007). Confucian work dynamic distinguishes between long-term and short-term orientation, formulated by Hofstede (1998). Long-term oriented cultures emphasize the fostering of virtues oriented toward future rewards (Hofstede *et al.*, 2010) or persistence with slow results, ordering relationships by status and observing order, thrift, saving, adaptation of traditions to a modern context and a concern for virtue rather than truth. On the other hand, short-term oriented cultures is the fostering of virtues related to the past and present in particular, respect for traditions and social status regardless of cost, and are concerned with saving face (Hofstede *et al.*, 2010). They tend to see things in right and wrong terms and expect quick results (Hofstede, 1993).

Later in 2010 edition of Hofstede *et al.* book 'Cultures and Organization: Software of the mind', the new dimension 'Indulgence versus Restraint' was added. This dimension is focusing on happiness and life control. "Indulgence stands for a society that allows relatively free gratification and natural human drives related to enjoying life and having fun" whereas, restraint indicates a society by regulating it with means of strict social norms (Hofstede *et al.*, 2010, p. 281).

Researchers of cross-culture noted that power-distance is inversely related to individualism (Hofsted, 1991; Kim, Triandis, Kagitcibasi, Choi, & Yoon, 1994) and the impact of the power-distance and individualism dimensions together has been

shown to be most relevant to group behavior and group outcomes (Tan, Wai, Watson, Clapper, & McLean, 1998, Tan, Wai, & Watson, 1999; Watson, Ho, Ramann, 1994). In addition, the influences of cultural differences not only contribute toward the form of human groups' perceptions and decision making but also toward information technology development. Hales (1995) states that "...culture approaches to computing technologies are rather high profile" (p.105). Hofstede's model of cultural differentiation is frequently used to explain the influence of national culture on the use and design of information technology or groupware (Sondergaard, 1994; Mejias, Shepherd, Vogel, & Lazaneo, 1997). When developing a successful information system, developers have to apply universal rules according to the situation in a particular country by evaluating the appropriateness of the methodologies and adapting them to suite the national culture (Korpela, Soriyan, Olufokunbi, & Mursu 1998). Palvia & Hunter (1996) also indicate that the different uses of the same methods of information technology in countries can be caused by the differentiation of cultures. In summary, local culture can change the way group decision making is generated as well as the technology which is used.

Many GSS researchers mentioned that national culture may encourage differences in GSS usage by groups (e.g., Nunamaker *et al.*, 1997; Vreede, Jones, & Mgaya, 1998; Vreede, Mgaya, Schuuring, Cornelissen, & Feenstra, 2000; Watson, 1994). The studies clearly show that the socio-cultural environment indeed influences the way in which GSS is employed. As discussed earlier, culture and language are related. The magnitude of developing GSS to increase productivity and eliminate the communication barrier in multi-cultural and multilingual groups is gaining greater interest from GSS researchers (Aiken & Vanjain, 2009; Aiken *et al.*, 2010; Aiken,

Balan, & Hazarika, 2011). Nevertheless, both culture and language are still an area of great neglect in the research of GDSS context variables. Only a few articles have looked at and assessed these factors particularly in terms of linguistic differences (e.g., Fjermestad & Hiltz, 1997; Landry, 1995; Reinig & Mejias, 2002; Quaddus & Tung, 2002; Aiken *et al.,* 1994; Aiken, 2008; Aiken *et al.,* 2010).

2.8.2 Implications of using GDSS in the Education Industry

While GDSS or GSS studies for university classroom education have been widely explored in Western countries for various activities, there is little research in non-Western countries where national cultures are distinctly different from Western cultures. GDSS in particular may be suitable or workable in Western cultures, but it may not be appropriate in certain Asian cultures.

Since the introduction of GDSS over three decades ago and because of the success of using G(D)SS to support organizational collaborative decision making and activities in effective ways, there has been recent interest in using GSS in the academic area to promote and improve student learning, such as classroom discussion and project teams. GSS has also been introduced into the classroom as educators realized the potential of collaborative learning activities, such as improving the participation rate and the productive rate of group performance. Since GSS was first introduced into the educational environment, GSS has been gaining attention from many researchers. A number of research studies of GSS in classrooms have been published. Most of the studies have attempted to address the basic question of whether GSS can improve "learning". To answer this question, the impacts of GSS on learning performance and satisfaction have been investigated (see, example, Kwok & Khalifa, 1998; Aiken, 1994-1995; Alavi, 1994). Another significant goal of using

GSS technology in the educational process is to provide students with a richer and more valuable educational experience compared to other tools (Manheim, Vlahos, & Xie, 1995).

Apart from being exploratory in nature to address the basic question of whether GSS can improve "learning" or the improvement of student productivity, some studies have also investigated other issues related to the learning process. With today's educational techniques, educators are prompted to develop cooperative learning and communication skills including critical thinking in an interactive environment (Glynn, Qiontana, Cunningham, & Cooper, 1998), using a group-oriented approach (Tyran & Shepherd, 2001). A substantial finding of Gahr (2003) is evidence suggesting that active learning techniques improve learning, achievement, and other related outcomes regardless of the subject being taught—compared to traditional lecturing approaches. Group-oriented learning approaches have been used successfully for learners ranging from elementary grades to the university level (Schmuck & Schmuck, 1988). While the group-oriented learning method has shown the potential of learning improvement, it still has several traps associated with using such a method. There are a variety of factors that influence student perceptions of the group learning process and classroom participation. Problems that can reduce the effectiveness of learning are limitations on the communication "air time", fear of speaking up, and free riding (Salomon & Globerson, 1989). These problems become the motivation to promote GSS in classroom environments.

Many researchers have found similar outcomes. GSS experiences in learning environments are very inspiring in that they have led to higher levels of study performance and better skill development. Callan and Whymark (2002) show great

success in increased student participation, motivation, and the quality of outcomes by applying GSS technology into the classroom. Briggs, Ramesh, Romano, & Latimer (1994-1995) paper reported that compared to 'traditional' students, GSS supported students participated more in lecture sessions, created essentially higher quality and quantity answers, and were more interested in the lectures. The observation results of Leidner & Jarvenpaa (1993), Reining (1997), and Tyran (1997) demonstrated that GSS technology-supported learning could motivate participation among students by inviting students to conduct exploratory analyzes of course subjects and this could produce a higher participation rate.

Since the learning process is defined and executed by groups, GSS can bring a number of benefits to the process, such as (Haake & Wang, 1997):

- Enabling all participants to work simultaneously;
- Providing equal opportunities for participation;
- Enabling groups to effectively bring more information, knowledge, and skills to bear on task;
- Offering access to external information
- Supporting the development of an organized memory, and
- Increasing satisfaction with classroom experience.

GSS technologies can transform a traditional instructor-oriented classroom into a student-centered classroom. In addition, the role of computers in education has been broadened to include problem solving and group brainstorming. Because of the advantages of GSS gained in the learning process, this area has received attention from researchers. Over the past thirty years, research investigating the effect of using collaborative technology in classrooms has continuously been conducted. For

example, Aiken *et al.* (1994-1995) studied the appropriateness of GSS in classrooms. Comparative outcomes between traditional synchronous discussion and a GSS approach were reported. Glynn *et al.* (1998) examined the application of GSS technology in high schools. In 2001, Vreede and Mgaya investigated the benefits of electronic meeting support in educational environments. Although there already exist a number of researches on information exchange in Face-to-Face unstructured group decisions, there is still a need for further research on information exchange and idea generation within GSS (McGranth & Hollingshead, 1994). "GroupStorm" was developed at UNSW@ADFA with the objective of utilizing GDSS technology together with the advantages of having facilitators present. The study has examined the classroom contribution using the Face-to-Face (FtF) mode (the same time, same place) application. The key concepts and perceptions of GroupStorm provided a good foundation and strong motivation for further enhancement, in particular to search for tools and methods to promote favorable information systems for group interaction in the Thai educational environment.

There are few empirical studies that have investigated the effectiveness of GDSS in an international education society environment in Asian countries. Hence, to close the gap along with concurrent improvements in computer technology, this study proposes to investigate GDSS support in regard to the Thai educational environment. The design, development and use of a particular GSS to support group-oriented learning, has been implemented in the Thai graduate classroom. This has brought about the design science development of a software-supported ability for a group of international students – a multi-native language and culture group including Thais, Chinese, and Vietnamese to generate their ideas in their native language in

preference to using English as the mode of communication. The anonymity feature was also included in the application development in order to enable the students to voice their opinions freely without fearing a confrontation with the other students and to increase their participation for the exchange of ideas which may translate into improvements of learning performance, decision quality and higher satisfaction of users with the application usage outcomes.

2.9 Conclusion

At the beginning of this chapter, the foundation and importance of GDSS or GSS as a research field was described, and emphasis was placed on the definition and the characteristics of the system. Various studies and techniques for the utilization of GDSS software were also presented. In the research study that forms this dissertation, all key input variables of a group work model including technology, task, group, and context, were utilized for the design and development of a new GDSS tool. These variables were chosen because they have an influence on the focus group and the expected outcome of increasing the group learning process or group task performance and the quality of multi-cultural and multi-language classrooms. The Face-to-Face (FtF) communication mode is the main scope of the study. Based on the traditional Thai educational method, same time same place student participation is the general environment of Thai classrooms.

The next chapter will discuss the research philosophies and approaches that could be adopted for this study in detail as they are an important basis for the research.

Chapter 3

Research Methodology

3.1 Introduction

The review of the literature carried out in the previous chapter stated that research involving Decision Support Systems (DSS) or Group Support Systems (GSS) is interdisciplinary. It is categorized as a sub-area of the information systems (IS) discipline. Developing a GDSS system is a complicated process and, as a result, may vary in desirable functionalities, depending on the research contexts and settings. The previous chapter also pointed out some opportunities for further research. One of the most important concerns is that despite the complexities of GDSS research studies and of the design research approach, which will be possible only with suitable supporting technologies and the skills of the developer / researcher, it still does not guarantee high performance and beneficial outcomes.

Relying on one dimension alone to explain a complex phenomenon such as decision making may portray an incomplete view. For example, if one applies just technology advancement to design and enhance a computer-based system's contribution to decision-making effectiveness and efficiency, one can fail to recognize the role that nature and human / social aspects have on idea generation, decision quality, and satisfaction. In other words, user perceptions and attitudes, environmental and cultural differences have been major themes for research in determining whether they are dominant factors when groups of people interact to make a decision or perform a task or project. Kwok & Khalifa (1998) stated that

> "Recent advances in computer and communication technologies have created new opportunities for use of these technologies in education.... Group support systems (GSS) are expected to reduce group process losses associated with information overload, social pressure, attention blocking and other difficulties." (p. 463)

As information system (IS) research is derived from multi-disciplines (Fjermestad and Hiltz, 1988), this chapter explains the details and the components of the chosen research methodologies; design science, positivism and interpretivism, as well as the methods employed. Specifically, the descriptions of reasoning about the philosophical stance and design science approach on which this thesis is based will be reviewed. The presentation of philosophies here is essential because they are the grounds on which this study is built. The next section provides a brief discussion of the research setting target group.

3.2 Research Setting: Country

Research setting selection is an important part of the successful research project (Baker, 1994, Bernard, 2000). An appropriate research setting can help a researcher in effectively taking notes when examining proposed theories and in drawing conclusions about empirical tests (Doktor, Tung, & Glinow, 1991). By clearly defining the examination subject, validity, and scope of the particular research question, this can assist in the selection of a research methodology for the investigation.

The majority of past studies regarding IS research, including GSS has been undertaken in western countries (such as US, UK, etc.). Much less has focused on investigations regarding other regions (such as Asia, or Africa). This may limit theory generalization. In order to bridge this gap and test the external validity of western-developed theories by collecting data in a non-western country, Thailand, a country in Southeast Asia, has been selected for this study. The main reason for this choice is that Thailand is culturally similar to its neighbors and is considered to be substantially different to western countries (McGill, 1995). Furthermore, Thailand is the only country in South East Asia that has never been under a western colonial power which affects the development of culture (Bhabha, 1994) and also has an impact on the ability to communicate in a foreign language (Chieochan, Lindley, & Dunn, 2003; Corbitt, 1999). Conducting this research in Thailand therefore allows the assessment of the validity of theories to meet the set research goal.

Referring to Chapter 2, national culture has been defined by various researchers in a number of related ways. In short, culture depicts the way members of a society relate to each other and to the environment (Trompenaars, 1998). The culture makes one social environment different from another and molds its own shape (Kroeber & Kluckhohn, 1952). Komin (1991) defines the national culture as one that characterizes national traits. Of course, this does not mean that every person in the culture arranges all the characteristic dimensions in the same order of importance. Therefore, in describing Thai national characteristics, we are only referring to the common characteristic elements within the Thai culture. (p.687) Hofstede's (1991) investigation of western and non-western countries indicates that Thai society has a high power distance as reflected in its strongly hierarchical and bureaucratic society.

This means that junior people will not do anything until they get approval from senior or higher status people. Rohitratana (1998) also found that Thai subordinates are not likely to make decisions. They avoid taking risks, because risk means more uncertainty on getting approval.

Face saving or ego orientation or the criticism-avoidance value also plays a very important role in Thai culture as part of the way power is practiced. Thais try to avoid conflict and criticism at all times because of the face saving value. The study of Komin (1991) mentions that saving face or the avoidance of criticism is extremely important in Thailand. He states that

> "The 'face' is identical with 'ego' and is very sensitive. Since the Thais give tremendous emphasis on 'face' and 'ego', preserving one another's 'ego' is the basic rule of all Thai interactions both on the continuum of familiarity-unfamiliarity and the continuum of superior-inferior, with difference only in degree." (p.135)

Thailand also has high uncertainty avoidance (Hofstede 1997). Societies that are high in uncertainty avoidance hold a narrow-mindedness for and anxiety towards uncertain or ambiguous situations. With such a relationship within a Thai group, it demands that no one be placed in an embarrassing situation; the group takes every effort to avoid causing anyone of them to lose face.

Under the individualism dimension, Asian, South American and African cultures mostly tend towards collectivism (Hofstede, 1991). The impact of this value in Thailand is that Thai people do not contribute much in group work or meetings. They always think that their contribution as an individual is

not important. In addition, Thailand obtained a very low score for individualism. From the survey of Hofstede comparing countries (cited in Holmes & Tangtongtavy, 1997, p.17), Thailand ranks ninth of fifty-three cultures which indicates a strong 'collectivism' society and therefore there is a low degree to which individuals are integrated into groups. For instance, group decisions are considered better than individual decisions (Hofstede, 1980).

Another important behavior of Thais considered to impact on group decision-making outcomes is uncertainty avoidance. Uncertainty avoidance relates to the degree to which people are comfortable with ambiguous events and the lack of ability to predict future events with accuracy. Hofstede's survey (1984) indicates that Thais present a moderate degree of uncertainty avoidance ranking 22nd out of 40 countries.

In addition to the national social-culture, education trends in Thailand are another significant factor (Prpic & Kanjananapanyakom, 2004). Thai education has evolved through many changes over time due to educational value being recognized as the most basic requirement of any economic and technological upgrading (Hobday, 2000). Today, educating students in Thailand is a big challenge due to the globalization of education converging towards international standards of curricula and assessment, and the need for the ability to communicate in a second language. International education is now of major importance to Thailand. Internationalization includes the practices of knowledge and language acquisition, enhancing the curriculum with international content, and incorporating the understanding of cultural issues. In academic programs for international students, there is the coming together of both local and foreign students. The English language has become the language

adopted as the global lingua franca. However, it is generally known that the English language is a key source of difficulty in learning for Asian people or non-native speakers of English, especially Thai students who have a level of English proficiency lower than most Asia countries (e.g. Malaysia, Philippines, and Singapore). Oral communication or participation in English in multi-national student classroom discussions or learning is difficult. Students in a particular class do not have the same level of English language expertise. They do not recognize the words in English. They manifest a lack of confidence with English. They feel unhappy with oral performances. These have intimidated or obstructed the students when required to ask or answer a question or to give a comment to avoid a situation where they may lose 'face' from using incorrect English words or pronunciation in front of the evaluative eyes of other native speakers or participants who have strong English language skills. One obvious impact is lower knowledge or idea sharing. Sawir (2005) states that "prior research indicates that many international students from Asia, studying in Australia, face serious learning difficulties and lack confidence in speaking, and refrain from taking a proactive role in classrooms" (p.567).

Offering the use of GSS with the appropriate features that create opportunities, and encourage multi-national students in classrooms to engage in more equal and enthusiastic participation should therefore produce more learning processes, higher learning efficiency and effectiveness, and better decision quality.

3.3 Research Framework

The major purpose of this study is the development of a new GSS technology artifact to be utilized in international classrooms for engaging and

increasing Thai graduate group participation and learning level in a multi-cultural and multi-language environment. The key features of the artifact focuses on using portable hardware, such as a computer notebook and USB keyboards, and the integration of multi-criteria analysis of group task brainstorming that may encourage the students to generate more ideas leading to higher quality outcomes of group decisions. To achieve the purpose, this research is carried out based on a combination of the design science paradigm and the widely accepted research philosophies in the information systems research field: positivist and interpretivist paradigms. These methodologies serve as a common framework. They are used for examining the area of designing and developing a GSS application, a pilot study, and a laboratory experiment, attempting to bridge a research gap by incorporating the social perspective, task-related activities and information technology influencing the level of group learning performance, decision quality, classroom participation, and perceptions of the student experience.

3.3.1 Philosophy in Information Systems Research

Shanks, Rouse, & Arnott (1993) defined research as a systematic process of acquiring new knowledge, based upon some form of empirical evidence. In the IS discipline, researchers can observe a wide-ranging discussion of research paradigms (Weber, 1987). Information technology capabilities and characteristics are often developed and implemented in response to human purposes or specific task requirements by using practical knowledge. As a result, IS research studies commonly deal with human actions. For more than three decades, the conceptual model has been one of the key research topics in the field of IS (Wand & Weber, 2002). A conceptual model represents a specific domain which serves as a vital step in the

software development process. Nevertheless, due to the difficulty of conceptual modeling, it has become less popular in the past ten years (Wand & Weber, 2002). Experimental research (Batra, Hoffer, & Bostrom, 1990) and survey research (Batra & Marakas, 1995; Hitchman, 1995) supports these views.

It has long been known that there are two main types of research; *positivist* and *interpretivist*. When we discuss epistemological paradigms, positivism and interpretivism are emphasised (Lee, 1991; Mingers, 2001; Probert, 2001; Weber, 2004). Positivist research approaches are objective, while interpretivist research approaches are more subjective. The IS literature has provided an extensive discussion on positivism and interpretivism since the 1970s. For instance, Shanks *et al.* (1993) explained that research methods in the IS field are associated with two theoretical paradigms; positivism and interpretivism, depending on different philosophical assumptions about the nature of science and the nature of social reality. Orlikowski and Baroudi (1991) consider three broad IS research paradigms: positivist, interpretivist, and critical theory. They found that from 1983 to 1988 97% of 155 IS research works applied a positivist position. Since then, a range of non-positivist investigations, particularly based on interpretivism, has been growing. Consequently, interpretivist approaches are being employed for conducting and evaluating IS research (Klein & Myers, 1999).

At present, one of the most accepted philosophies in IS research is the *'design science'* approach, also referred to as *'design theory'* by Markus, Majchrzak, & Gasser (2002) and Walls, Widmeyer, & El Sawy (1992) or as the *'sciences of the artificial'* by Simon. Design science research is the third paradigm of IS research that complements the interpretivism and positivism approaches (Hevner, March, Park, &

Ram, 2004; Vaishnavi & Kuechler, 2006; Livari, 2003). Simon (1988) classifies two different ways of conducting research in the IS discipline; *'natural sciences'* (the concept of behavioral sciences) and *'sciences of the artificial'* (the root of design-sciences). His study clearly identifies design as central to what humans are and what they do. Simon (1996) also indicated that humans need a science of design that is a "tough, analytic, partly formalizable, partly empirical, teachable doctrine (p.113)." The *'sciences of the artificial'* focus on how to design and construct artificial systems and researchers or innovators seek to extend the human and organizational capacity frontiers by creating new and innovative artifacts; on the other hand, the natural science approach focuses on how "things" are and how they work and researchers search to develop and verify theories by explaining or predicting human or organizational behavior. Jarvinen (2004) also makes the distinction that natural sciences stress "what is the reality" and design sciences stress "utility of artifacts".

Even though different concepts have been used, March & Smith (1995) and Hevner *et al.* (2004) present similar distinctions by introducing two paradigms for IS research: behavioral science and design science which are understood as "problem solving" paradigms. The design science paradigm addresses two distinct phases of a problem-oriented process: understanding and solving it. To be useful for problem solving and to change the IS usage in practice, IT artifacts are considered and generated as an effective tool (Orlikowski & Iacono, 2001). In 2003, Benbasat & Zmud suggested that the core of IS discipline should be the IT artifact.

3.3.2 Philosophy in GDSS research

After carefully analyzing previous sections and considering the scope of this thesis, design science can lead to appropriate artifact functionality design that

helps to develop a GDSS tool which can reduce the interruption of the flow of ideas, remove the language barrier of idea generation and improve the effectiveness of international group learning and processes. George, Easton, Nunamaker, & Northcraft (1990) conceptualize GDSS as the IT artifact. Their study investigated the impacts of a set of group communication tools with specifiable features that are hypothesized to produce more effective group outcomes than would result from Face-to-Face communications without those tools. Lewis (1994) states that GSS research is a convergence of three reference disciplines: behavioral sciences, decision science, and information systems. Furthermore, in the GSS literature, an extension of natural sciences in the positivist paradigm can be widely found (e.g., Chen & Hirschheim 2004; Klein & Myers, 1999; Orlikowski & Baroudi 1991; Trauth & Jessup, 2000).

Dennis *et al.* (1988) proposed that group support system (GSS) research should adopt both a positivistic approach in order to provide control and rigor and also an interpretivist approach to ensure relevance. Eden (1995) went further than Dennis *et al.* (1988) to claim that these two approaches of GSS actually exist. He termed the "narrow-band" approach, which is characterized by generating positivistic experimental evidence from laboratory-based examination, the approach to test the performance change of student subjects resulting from changes in the process in which they operate, while he termed "wide-band" approach, which has sometimes been claimed to be unproven, as the method through which evaluation is based on interpretivist paradigms which provide qualitative reflections of real-world events.

Trauth & Jessup (2000) attempt to compare and contrast positivist and interpretivist positions applied to GSS research. Subsequent frameworks for GSS research have included the task as an important variable (Benbasat & Lim, 1993;

Nunamaker *et al.*, 1991a). Zigurs & Buckland (1998) utilized a positivist approach to describe the theory of task/technology fit used to assess the relationships between group tasks and GSS technology to achieve GSS usage. There are also numerous GDSS researches that approach the social psychological position (Diener, Fraser, Beaman, & Kelem, 1976; Gergen, Gergen, & Barton, 1973; Huber 1984). Additionally, the study of Chen & Hirschheim (2004) on 1,893 IS articles published in eight major IS publications from 1991 to 2001 reported that 81% of articles were of positivist orientation and only 19% used the interpretivist approach. In the research article "A critical analysis of decision support systems research" of Arnott & Pervan (2005), their analysis shows the tendency of DSS research paradigms from 1990 to 2003 that 92.2% of DSS researches applied a positivist approach; only 7.6% used an interpretivist approach. This means GSS research is dominated by positivism.

However, there is no clear consensus on which taxonomy of philosophical positions is appropriate for IS research. The research approach selection depends on the researcher's philosophical assumptions. Nonetheless, research studies based on a pure methodology are rarely found. Technology and behavior are inseparable (Lee, 2000). Reeves, Herrington, & Oliver, (2005) recommend design research as a proper approach to socially responsible inquiry, particularly in the field of educational studies. For the research in this thesis, not only the creation of a new GDSS IT artifact for educational area was undertaken, but also pilot and experimental tests as positivist and interpretivist paradigms were integrated. The objectives are to investigate the fundamental behavior of people in group phenomena and evaluate the outcomes of social interaction based on the quality and efficiency of learning and critical thinking though a prototype 'BrainEx' group-supported information system

application. The next section will provide the introduction of design science methodology.

3.4 Overview of Design Science Methodology

> The design-science methodology is fundamentally a '*problem-solving paradigm*'. It seeks to create innovations that define the ideas, practices, technical capabilities, and products through which the analysis, design, implementation, and use of information systems can be effectively and efficiently accomplished.
>
> (Tsichritzis 1997; Denning 1997)
>
> "The design science paradigm seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts."
>
> (Hevner *et al.*, 2004, p. 75)

"Design science" or "design research" is a powerful opportunity and has gained strong interest especially in the relevant study of Information Systems and Engineering fields (Au 2001; Ball 2001; Burstein & Gregor, 1999; Hevner *et al.*, 2004; March & Smith, 1995; Markus *et al.*, 2002; Nunamaker *et at.*, 1991a; Walls *et al.*, 1992). To understand and appreciate the design science approach, it is necessary to comprehend all related key words. The importance of "design" is recognized in IS academia. Existing researches (e.g., Edelson, 2002) suggest that design implies solving problems, creating something new, understanding how things work and why, and offering unique learning lessons. It is a key activity in fields like architecture,

engineering, and urban planning (Schon, 1993). Simon (1988) states that designers devise "courses of action aimed at changing existing situations into a preferred one (p. 130)." Benbasat & Zmud (1999, p. 5) claim that IS research is directly related to its applicability in design, stating that the implications of empirical IS research should "be implementable, ... synthesize an existing body of research, ... [or] stimulate critical thinking" among IS practitioners.

The objective of design-science research is not only "to develop technology-based solutions to important and relevant business problems" (Hevner *et al.*, 2004, p.87), but also to develop empirically grounded theories by combining studies of both the process of learning and the means that support that process (diSessa & Cobb, 2004; Gravemeijer, 1994). This is a reason why sometimes design science is also called design theory. Design theory is a prescriptive theory based on social and natural science theories that is focused on prediction and goal achievement (Markus *et al.*, 2002; Walls *et al.*, 1992). Various guidelines and methodologies for design research have been established. A design research approach applies design as a strategy for developing and clarifying theories (Edelson, 2002; Barab & Squire, 2004). These are some examples of proposed guidelines: the theories of design (Walls *et al.*, 1992) and guidelines for academicians conducting design-centric projects (Hevner *et al.*, 2004). The guideline and methodology objectives are to create and evaluate new knowledge like IT artifacts in order to dignify efforts in clarifying the outputs of such research (March & Smith 1995).

Herbert Simon (1996) identified 'Design Science' as an inventive or creative problem solving activity, one in which new technologies are the primary products. The 'design science research result' produces and applies knowledge of

task in order to establish an effective ***information technology (IT) artifact*** that defines the ideas, practices, and technical capabilities (Tsichritzis, 1997) and can improve individual, organizational, and societal capabilities (Hevner *et al.*, 2004; March and Smith 1995). The term *artifact* is used to describe something that is artificial, or constructed by humans, as opposed to something that occurs naturally (Simon 1996). Design science artifacts are information technology-oriented and they are considered against value and utility (Purao, 2002). IT, defined by March and Smith (1995), is technology used to acquire and process information in support of human purposes. Artifacts, by Walls *et al.* (1992) and Markus *et al.* (2002), rely on existing "kernel theories" that are applied, tested, modified, and extended through the experience, creativity, intuition, and problem solving capabilities of the researcher. Orlikowski and Iacono (2001) define IT artifacts as "those bundles of material and cultural properties packaged in some socially recognizable form such as hardware and/or software" (p.121).

In 2003, Benbasat and Zmud described the IT artifact as "the application of IT to enable or support some task(s) embedded within a structure(s) that itself is embedded within a context(s)" (p. 186). They also identified four elements of an IT artifact: information technology, the tasks, the task structure, and the task context within which it is used. These four elements of an IT artifact exploit knowledge from various disciplines including computer science, society, human factors, and organizational psychology. Designers of usable technologies therefore need to consider the broader social contexts – technologies can be used as long as users adopt design principles.

Many studies (e.g., Akker, 1999; Cobb, Confrey, diSessa, Lehrer, & Schauble, 2003; Kelly, 2003) outline possible characteristics that apply to most design research as:

- Interventionist: to design an intervention in the real world.
- Iterative: to incorporate a cyclic approach of design, evaluation and revision.
- Process-oriented: to focus on understanding and improving interventions.
- Utility-oriented: to measure the merit of a design, in part, by its practicality for users in real contexts.
- Theory-oriented: to build a theory, the design is (at least partly) based upon theoretical propositions; and field testing of the design contribution.

3.4.1 Design Research Process

Design research is a collaborative process that draws upon multiple theories, methods and disciplines to develop a *technology-enhanced learning* environment (Bannan-Ritland, 2003). "Design research integrates the development of solutions to practical problems in learning environments with the identification of reusable design principles" (Reeves, 2006, p. 86). With its phenomena, design research has gained momentum and has become an important research concept since it was first presented in the early 90s. It is an emerging form of inquiry in the educational field in order to support learning performance through the design of learning environments (Baek, 2007).

However, designing useful artifacts is complex due to the needs for creativity advancement in the domain areas in which existing theory is often

insufficient. Since the main objective of design science researchers is to produce objective knowledge, they facilitate its enactment by applying knowledge of tasks or situations in order to create "artifacts" that are innovative and valuable. They are represented in a structured form that may vary from software. Researchers have to adequately understand the inherent problems for development or problem solving, and the solution approach (Klein & Meyers, 1999) in terms of both the quality of the product and the design process.

Design sciences include both products of design and processes of design covering the testing and evaluation of new artifacts to show improved performance (e.g. efficiency, satisfaction). March and Smith (1995), building on Simon's work, identify two basic activities in design research field: *build* and *evaluate* activities. "Building is the process of constructing an artifact for a specific purpose, while evaluation is the process of determining how effectively the artifact performs" (p.254). The building process recognizes the important steps from design to construction. Evaluation is a complex activity because of the fact that performance is related to intended use and the range of tasks. For example, in a decision making situation through an application, not only is the artifact evaluated but also evaluation criteria must be determined under the particular environment. This build-and-evaluate loop is normally iterated many times before the final design artifact is generated (Markus *et al.*, 2002). Based-on Rossi & Sein (2003), there are five steps to process a design science research activity.

3.4.1.1 Five steps of design science research

1. "Identify a need" including identifying deficiencies in current systems and problems in the field and also searching for previous relevant topics.

2. "Development" covers designing the system, applying software engineering principles, and getting the best tools or building an artifact based on a perceived need. The measures of success are also defined in this step.

3. "Evaluation" includes analysis of the built systems and processing field trials and determines whether the artifact solves the problem it was intended to solve. The evaluation should be done against the defined measures. In this step, Rossi and Sein (2003) propose both internal and external criteria. The internal criterion consists of the "Match between the artifact and the 'abstract idea', the match with generally accepted principles of designed artifacts, and is the artifact representing a "good system" as defined by the field" (p. 8). An external criterion is the advancement of design theory.

4. "Learned" and

5. "Theorized"

The last two steps reflect on the process and product, generate findings and confirm or reject the original assumptions.

In addition, Takeda, Veerkamp, Tomiyama, & Yoshikawam in 1990 discussed the building of a computable design process model by applying the 'design experiment' method. From a problem-solving point of view, they constructed and analyzed the observable knowledge flow occurring in the course of the general computable design cycle drawn in Figure 3.1. The observable knowledge flow was generalized into a process inherent analysis step for any artifact design effort and has become widely accepted by researchers as "The General Methodology of Design Research". Design research methodology comprises five steps: awareness of the problem, suggestions, development, evaluation, and conclusion.

With this model, the general design cycle is well-suited and chosen for this GSS thesis since there are many areas of GSS that are yet to be explored. Examples of GSS issues that could profit from research are culture and native language differences in international education initiatives.

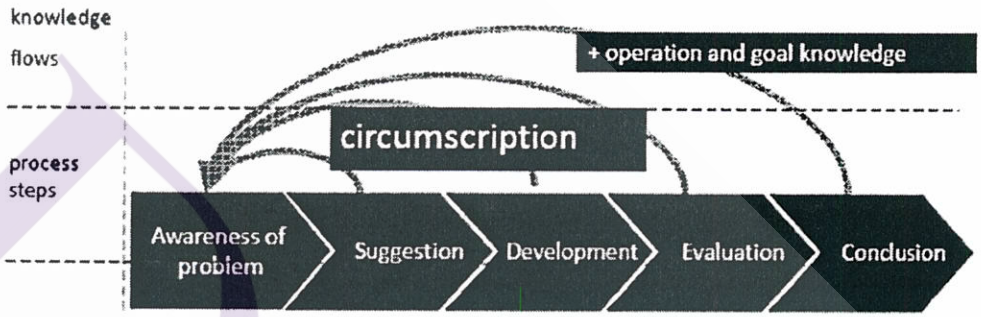


Figure 3.1 The General Methodology of Design Research, Takeda *et al.*, 1990, p.45

Starting from the *Awareness of a Problem*, this designation emphasizes the problem- solving/performance-improving nature of the activity. The output of this step is a proposal for a new research effort. *Suggestions* for a problem solution are abductively drawn from the existing knowledge/theory base for the problem area (Peirce, 1931-35). In this step, new functionality is envisioned in accordance with a configuration of existing or new and existing elements. *Development* will attempt to implement an artifact in accordance with the suggested solution. The implementation technique will vary depending on the artifact construction. The *Evaluation* phase contains an analytic sub-phase which evaluates the created artifact in accordance with criteria that are made explicit in the awareness of problem phase. The final phase, *conclusion*, typically indicates the completion of a specific design project.

In addition to understanding design research, the *Circumscription* process is important because it generates *understanding that could only be gained from the*

specific act of construction. Circumscription is a formal logical method (McCarthy, 1980) that assumes every fragment of knowledge is valid only in certain situations.

This study also applies the design science framework proposed by Hevner *et al*. in 2004 that describes the design development process and how behavioral and design sciences interact. Three major components are identified in the framework; the environment, IS research and the knowledge base. The environment includes elements such as people (task), organization, and technology. Knowledge base provides the knowledge foundation of existing behavioral and natural theories, as described in the literature review, to instruct the IS research components which focus on the development and evaluation of the artifact. The evaluation results in refinements to the design theory and iterative build-evaluate cycle (see Figure 5) to the additional knowledge base.

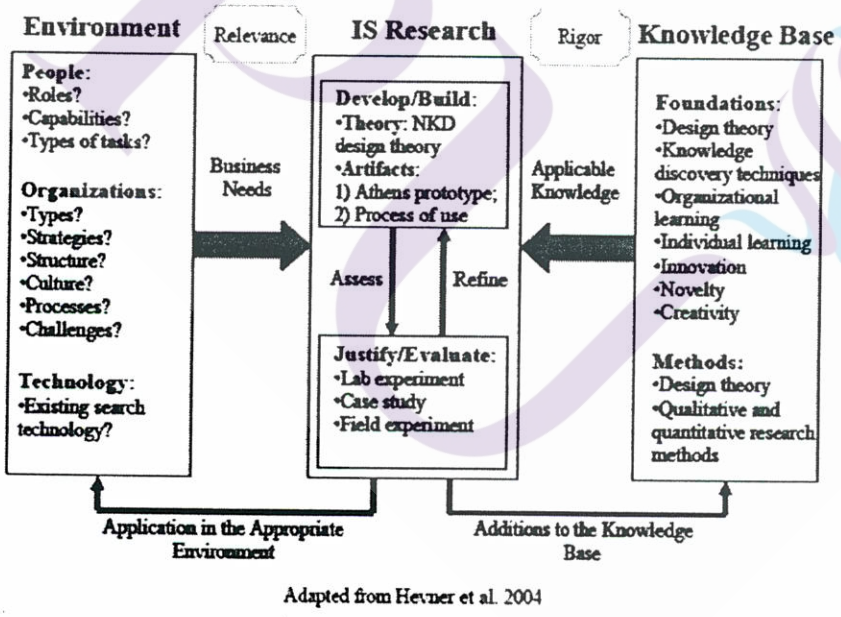


Figure 3.2 IS Research Framework, Hevner *et al.*, 2004, p. 80

3.4.2 Output of Design Research

In design science, the output is an artifact. There are four types of artifact product: a construct, a model, a method, and an instantiation or an implementation (March & Smith, 1995; Hevner *et al.*, 2004). *Construct* means the language (vocabulary and/or symbols) where problems and solutions are defined and communicated (Schon 1983). That is to say, constructs are the conceptualization and vocabulary used to clarify the problem and solution. *Models* use constructs to represent a real world situation – the design problem and its solution space (Simon 1996). Ball (2001) briefly described that constructs and models are the relationship of the artifact to its environment. *Methods* provide a set of steps (algorithms or guidance) for performing goal directed activities. They are composites comprising a process or algorithms similar to the constructs to describe the results of the application (Brinkkemper, 1996). Finally, *instantiations* show that constructs, models or methods can be physically implemented in a working system. An instantiation is the realization of the artifact in an environment. It has a physical existence in the real world, as a piece of hardware or software, or a series of physical actions. Instantiations demonstrate feasibility, enabling concrete assessment of an artifact's suitability to its intended purpose of performing certain tasks (Hevner *et al.*, 2004). In addition to technology-oriented IT artifacts, design science can also focus on organization, and work practice (Hevner *et al.*, 2004; Venable, 2006).

After understanding the basis of a design science research framework, it asserts that design science is a viable mode of research which has a significant contribution in the study and practice of GSS. Design science can lead to the development of methods that guide how an artifact like a GSS application should be

built and the next challenges are how to achieve building such an artifact that can solve the known problems and how to evaluate whether or not the established artifact meets the identified criteria or the research question. Whitten, Bentley, & Dittman (2004) presented two main stages of the IT artifact life cycle: *during* IT development and *after* IT development

3.5 Review of Positivism Approach

Since the mid-1970s, the IS literature provides an extensive discussion on positivism (Burrell & Morngan, 1979; Chen & Hirschheim, 2004; Lee, 1991; Weber 2004). Positivism means 'scientific'. The positivist approach is typically put into a practical view of science known as 'logical positivism' or 'logical empiricism'. This approach has been explicitly recognized and advocated as the natural science model. It is desirable to study social behavior in ways similar to those used by a natural scientist.

Positivism has been defined by Neuman (2006) as "an organized method for combining deductive logic with precise empirical observations of individual behavior in order to discover and confirm a set of probabilistic causal laws that can be used to predict general patterns of human activity" (p.81). This is to establish facts, and identify relationships or cause and effects between its elements from an external point of view. Researches based on the positivist approach look for measures and objectives by testing hypotheses with carefully analysis of statistical numbers from the measures. Concluded by Orlikowski & Baroudi (1991), four criteria for classifying positivism were stated;

1. "Evidence of formal propositions."

2. "Quantifiable measures of variables."

3. "Hypothesis testing."

4. "Drawing of inferences about a phenomenon to increase the predictive understanding of phenomena."

Research based on a positivist approach in GSS research normally intends to understand whether the technology can be helpful, how the technology can best be deployed, and what the marginal contributions of the various technology components are (DeSanctis, 1993). Some authors conduct a positivist paradigm in GSS to concede that by controlling many elements of the problem situation, the results cannot be generalized to the real world (see in Cook & Campbell, 1979; Mandviwalla & Gray, 1998; Trauth & Jessup, 2000). The positivistic research, also called the quantitative approach, embraces various techniques such as experimental, forecasting, and large-scale survey methods to verify hypothetical-deductive formalization. Most quantitative research methods involve either field or lab experiments or by asking and obtaining answers to questions by conducting a survey using questionnaires and/or interviews. Quantitative approaches are frequently subject to particular analytical techniques. They are methods for yielding and analyzing numeric information in the form of statistical analyses. By using quantitative methods, it is possible to make the statistical generality, with a degree of certainty, to depict results regarding the whole population from where the sample population is drawn. The quantitative method has therefore been applied to transform evidence or information of the relationships between GSS inputs, process, and outcomes and their strengths from laboratory testing and questionnaires into numbers and amounts.

Although the dominance of positivism has strongly gained momentum in mainstream information system research, including GSS, the positivist approach does not address all of the issues relevant to the real world (Mandviwalla & Gray 1998; Trauth & Jessup 2000) and in particular ignores social complexity and cultural contexts (Kaplan & Duchon 1988). This had led some researchers to adopt research approaches that focus on human interpretations and meaning (Myers, 1997). Interpretivism can add value to the positivist research approach with the main research emphasis being on understanding the *'social'* phenomena.

3.6 Review of Interpretivism Approach

> "Interpretive research can help IS researchers to understand human thought and action in social and organizational contexts; it has the potential to produce deep insights into information systems phenomena including the management of information systems and information systems development."
>
> (Klein & Myer, 1999, p.67)

The interpretive approach is defined as "the systematic analysis of socially meaningful action through the direct detailed observation of people in natural settings in order to arrive at understanding and interpreting how people create and maintain their social world" (Neuman, 2006, p. 88). Interpretive methods are subjective and adopt the knowledge of the reality of social construction and the context of human actors. In other words, rather than studying the causes and effects or relationships between individual variables as positivist research, the interpretivist approach seeks to generate a "form of social enquiry that focuses on the way people interpret and make

sense of their experiences and the world in which they live" (Holloway, 1997, p.1). Practically, this research method uses the researcher's interpretation of information which cannot or should not be translated into numbers or amounts. Traditionally, the outcome of interpretive researches intends to improve the understanding of social phenomena of the research questions. For the information system field, interpretive studies are "aimed at producing an understanding of the context of the information system, and the process whereby the information system influences and is influenced by the context" (Walsham, 1993, p. 4-5) and are stances for conducting and evaluating IS research (Klein & Myers, 1999; Walsham, 1995).

The preferred method of the interpretivist philosophical framework mainly focuses on qualitative aspects. The terms 'interpretivist' and 'qualitative' research are sometimes used interchangeably. For some, the former term is often taken to mean all approaches in human sciences that do not take a hypothetico-deductive approach to investigation, whilst others maintain more narrowly that qualitative research is itself characterized by an interpretive approach (Denzin & Lincoln, 1995). The three fundamental data collection methods in qualitative research so well known to scholars are, (a) interview, (b) participant observation, and (c) document analysis.

Interview as a form of data collection is a social interaction where the researcher meets or talks to the respondent either directly or via phone. The key advantages of the interviewing technique is that they are of short duration, have a high response frequency, and it is possible to address follow-up questions. However, there are a couple of disadvantages of this method such as the difficulty of addressing sensitive questions, and the possibility of interview effects referring to situations

where the interviewer may influence the respondent to answer in a certain way (Wiedersheim & Eriksson, 1991).

Observation as a data collection procedure is the most favorable in studying natural courses of events and behaviors that are hard to question or are sensitive to discuss (Fredman, Horndahl, & Stromberg, 1999). Two principal forms of observation recognized by Holme & Solvang (1991) are open observations and hidden observations. In open observations, the participants know that they are being observed and accept the observation. On the other hand, in hidden observation, the participants are not informed about the information collected or about what the researcher is investigating and do not know they are being observed. An advantage of the open observations method is that the researcher may be able to conduct practical research in the form of taking notes freely during explorative studies. This can sometimes create a problem. Since the study groups know that they are being observed, they may not act as freely and naturally as in the case of hidden observations.

The third data collection is document analysis or analysis of sources. Existing documents, for example pieces of paper, films or photos, can be used to answer questions regarding either relevant conditions – actual circumstances or actual courses of events, or the experiences of personal information.

As for the purpose of this thesis, regarding the comprehension of the impact of designing a new GSS - BrainEx application in the areas of non-participative culture and multi-native language classroom brainstorming sessions, this research applied an open observation approach as the qualitative data collection method. The focuses of this observation have been on the focus group participation rates and actions on using

BrainEx to express their ideas freely in the individual native language as well as in English.

3.7 Combining Proposed Methodologies

> "...design science creates artifacts, giving rise to phenomena that can be the targets of **natural science research**. **Group decision support systems**, for example, foster user behaviors that are the subject of natural science investigations."
>
> (March and Smith, 1995, p. 254)
>
> IS research based on the natural science or behavior science paradigm, should be complemented with research based on the design science paradigm.
>
> (Walls *et al.*, 1992; March & Smith, 1995; Hevner *et al.*, 2004)

Particularly in system development, many information system researchers are concerned that there is a broadening gap between research and practice (Galliers, 1994). IS researches generally merge people, organizations, and technology together (Devis & Olson, 1985; Lee, 2001). It deals with human creations for which researchers require understanding of human interactions. Lee (2001, p iii) states that IS research "examines more than just the technology system, or just the social system, or even the two side by side..." This is to address the inherent problems in order to develop and implement successful information systems within organizations (March & Smith, 1995; Nunamaker *et al.*, 1991a). It is therefore important to examine how the science of human beings could inform technology design.

The complementary *'problem understanding paradigm'* can be referred to as natural science (Hevner *et al.*, 2004, Simon, 1981). Natural science typically consists of traditional research in physical, biological, social and behavioral domains. The emphasis is on theories that explain human behavior as it is the root of the behavioral-sciences paradigm. It is viewed as two activities, discovery and justification (Kaplan, 1964). "Discovery is the process of generating scientific claims (e.g. theories, law) and justification includes activities by which such claims are tested for validity" (March & Smith, 1995, p.253).

Additionally, Hevner's study (2007) identified "a three cycle view of design science research. The Relevance Cycle bridges the contextual environment of the research project with the design science activities. The Rigor Cycle connects the design science activities with the knowledge base of scientific foundations, experience, and expertise that informs the research project. The central Design Cycle iterates between the core activities of building and evaluating the design artifacts and processes of the research (p.88). The researcher also stated that good design science research always starts with identifying and representing opportunities and problems in the software environment. In other words, understanding human and social groups are important parts of IS research success. The major reason is that the phenomena are artifacts that are designed and built by man to accomplish the purpose of man. For example, Briggs, de Vreede, & Nunamaker (2003) explain the discontinuation of empirically successful GSS applications stating that success depends on skilled facilitators. As a result, the system involves more than the computerized tools. Reviewing Gero (1999), the author noted that "using theories of material behavior, it has been possible to develop formal methods of analysis of the behavior of

configurations of materials (ie designs) under a variety of environmental conditions" (p.1).

Incorporating design science together with both positivist and interpretivist methods has been reported in social science literature since the early 1990s. The combination has aimed to extend and add value to the research. The Klein & Myer (1999) GSS is one of the best examples of a parallel design employing positivist and interpretive analysis. It was carried out separately on the same set of information. but the summaries were integrated to generate richer knowledge (Mingers, 2001). Positivism assumes the existence of a 'real world' that achieves knowledge of reality; while, interpretivism assumes achieving knowledge influenced by the subject or individual. As such, design research may require the combination of positivism and interpretivism.

In accordance with the purpose of this study, as described in an earlier chapter, utilizing the design science approach on building a groupware artifact, *'BrainEx'* GSS consists of a set of software, hardware, and language components and procedures supporting groups of people in learning, project management, or decision-related meetings, through brainstorming and idea generation. The system evolves from a personal decision support system by using computer-based systems and theories of group behavior and processes and behavioral decision theory. These factors address organizational application characteristics and practical relevance in which there is a need for reasonable balance. Through the work of March and Smith (1995), who developed a framework to demonstrate the relationship, activities, and outputs of design and natural science research in information technology, an idea for

designing a GSS artifact has therefore been created and arranged to realize the intended behavior of target users or focused groups.

As mentioned by March & Smith (1995), the difficulty of design science on artifact performance results is that it is dependent on the environment in which the software / system is operated. Hence, to employ or propose IS research, a researcher requires starting from pre-scientific observation and technology usage in practice, and theories about information system related issues. These are intended to predict human behavior, system or software available functions and the relationship between both aspects. The 'BrainEx' will therefore be developed based on a key research methodology - design science. To meet the overall purposes of this research, the mix of positivist and interpretivist methodologies were adopted and used in order to analyze the usability of the BrainEx GSS artifact concerning the pre-research assumption and expectations about the research target group.

An advantage of using both quantitative and qualitative methods in combination pointed out by Jick (1979) is validity. The validity of the analysis of the collected information is high and the analysis results are likely to be correct. The concept of validity can be defined as the measurement of what is supposed to be measured. This study has tried to increase validity by choosing different approaches regarding research methods and data collection processes. Pilot and laboratory experimental testing have been carried out and after completing the experiments, questionnaires were distributed to participants, a positivist approach, to analyze and evaluate the effect of BrainEx on the participation process and outcome performance, measured as numerical information generated in parallel during the experimental session. This approach of observation during data collection, the qualitative approach,

is suited for this research purpose. The approach has been utilized to investigate the particular participants' reactions about the artifact.

3.8 Conclusion

In summary, this chapter explicitly outlined the framework of the research approach which will be used for testing the artifact after the development phase has been completed. The challenge of this research topic is not only the difficulty of technical development but also in terms of finding appropriate testing methodology in a limited timeframe so that the research question can be properly answered. This research aims to generate a system or electronic group idea-generator to support the brainstorming of decision making tasks in the Thai educational classroom environment. A participation session as a Face-to-Face discussion (FtF) was conducted at a single location and processed through the initiated BrainEx system. Since technology-support for group work has been explored, in particular by using 'GroupStorm' developed by the Australian Defence Force Academy (UNSW@ADFA) together with the pros of having facilitation, this approach was deemed to be an appropriate foundation for further enhancement in particular in a Thai educational classroom environment. The limitations of GroupStorm mostly relate to the programming design, technology integration and language capability problems, principally the Thai language, described as follows:

- The current version has been developed using the Windows 98 operating system
- Not a completely Modular design
- User Interface is not user-friendly and attractive

- The coding is in an early version of programming standards which are not upward compatible to work on new hardware devices such as USB
- Cannot be used remotely or in multi-rooms
- New meeting process, concept or interested topic must be configured at the programming level.

The literature review in this chapter suggests a combination of design science, positivism and interpretivism can answer this thesis question. For the case of this thesis, BrainEx – the prototype of a GDSS tool – was built and evaluated against the research problem and design principles. Evaluating the tool using both qualitative and quantitative methods through pilot and lab experimental tests is described in Chapter 5. A questionnaire and observation of actual classroom settings will be adopted as the data collection methodology. The next chapter discusses the design development undertaken.

Chapter 4

BrainEx Artifact Design and Development

4.1 Introduction

In the previous chapter, the research methodologies that form this thesis were explained. The approaches are reflected in different stages of the research process to provide a synergistic combination of data relevant to the research questions. In reference to the list of deliverable IT artifacts that design science can encompass, which includes constructs, models, methods, and instantiations, two common design science frameworks can focus on either the design product or design process. In this study, the design principles surrounding the design productare are focused upon.

The inspiration of GroupStorm, the experimental GDSS from ADFA (UNSW@ADFA), became the starting point of this research. The artifact in this research is called 'BrainEx – Brain Exercise', because the first objective of this development is to provide a tool that allows discussion groups to overcome the challenge of a language barrier and cultural difference that may prevent knowledge sharing among the participants. "Culture is often seen as the key inhibitor of effective knowledge sharing" (McDermott & O'Dell, 2001, p. 76). By having that goal, it means that BrainEx must have the capabilities to allow all of the participants in the group to input ideas in their own languages at the same time. Secondly, this artifact will adopt the concept of a common display which all the participants can view and they can input and interact with each other on the same display. This concept is called 'Single display'. Thirdly, this artifact will be developed based upon software only.

There will be no special hardware requirement except for current industry standard connections which is the Universal Serial Bus (USB). This will improve the mobility of the facilitator. Fourthly, this artifact should be amenable for future development. Following these four main objectives, the development of BrainEx is unique and very challenging. Because of the fact that, in terms of development, there are many issues to be resolved in order to achieve these four main objectives i.e. How can the artifact be capable of accepting multiple language inputs simultaneously? How can the artifact differentiate the input information from various input devices? These critical issues will be discussed in this chapter.

4.2 Rationale and conceptual design

4.2.1 Rationale

Because of dysfunctional characteristics, many individuals are uncomfortable to participate in a group decision process which will inhibit full participation (Burdett, 2000, p. 3)

- Usually 20 percent of the participants dominate 80 percent of the time.
- People are often afraid to say what they think, or they feel pressured to conform to the group.
- Good ideas can be stifled and lost while people await their turn to speak.
- Ideas are often judged by who said them rather than by the value of what was said.
- Because people feel dissatisfied with the process, they are less likely to support the outcomes.

Regarding Thai students, Deveney (2005) noticed "the Thai students tended to be passive, responded only to direct questioning, were poor participants in class discussions and rarely put their hands up to answer questions, despite knowing the answers. Such behaviour was generally not displayed by either western students or other East Asian students from Taiwan, Japan and Korea" (p. 153-154). Inhibition affecting participation is likely to be an even greater issue when combined with the Thai cultural effect and multi-lingual discussion. Deveney (2005) had concluded in his studies that Thai culture does strongly impact on Thai students' learning. To improve this awareness, creating supportive environment where all students can develop confidence and participate in class discussion is a considerable key.

> A supportive environment is "safety, acceptance, and harmony so that students and teachers are able to manifest and learn from diverse values, perspectives and ways of interacting.. an environment that encourages students to be their authentic selves and to take the risks that are fundamental to personal development and academic performance."
>
> Ginsbert and Wlodkowski (2000, p. 51)

Nevertheless, various existing GDSS with multi-language capability or supportive suchlike GroupStorm - ADFA (UNSW@ADFA), smartSpeed Connect, CoFEEE, etc. have offered a great potential to transform international education barriers come in the way of productive or achieving learning. There are widespread barriers to successfully integrate technology in the classroom (Keengwe *et al.*, 2008; Ozden, 2007), for instance, lack of infrastructure includes network, and Internet, lack of accessibility includes insufficient computers, peripherals, and software, or other

resource-related issues includes time, problems with space, poor funding, curriculum translocation or migration difficulties, and lack of technical support (Bingimlas, 2009; Schoepp 2005). These are inspirations for developing the BrainEx GDSS which can improve the participation of Thai students not only in a Thai classroom environment, but also in a multi lingual environment where the limitations on access to hardware, software, and infrastructure and technical support are significant. Generic, inexpensive, and easy to use technologies are researched.

Being shy to participate or influence in English or any other language does not mean lack of knowledge. As such, the idea of using GDSS to improve participation will not focus so much on the linguistic competency but more on capturing as much as possible ideas and knowledge in any language from idea originators during the discussion process. The artifact development objectives are therefore derived as follows:

- To reduce the interruption of the flow of ideas during the brainstorming.
- To reduce the language barrier which may cause interruption of the idea flow.
- To help improve the effectiveness of discussion among participants who can be stimulated or elaborate on an interesting point put forward by the facilitator.
- To improve the utilization of those who are more capable to provide appropriate assistance with the subject matter to help the less capable persons. This may be due to the design of the artifact which uses only a single screen for discussion.
- To employ simple application that can utilize industrial standards such as USB.

- To provide stand-alone GDSS that are easily implemented in small and nonsupported network areas.

4.2.2 Conceptual design

Based on the rationale, objectives and the literature review, a major conceptual design of this study is concentrated on Process Structure GDSS type Level 1 which provides technical features designed to remove common communication and participation barriers in a same time/same place conference. The BrainEx application was designed to provide main features for leading group interaction and may promote more information exchange and participative interaction among group members. Highlighted characteristics of BrainEx are as follows;

- Sharing a single screen
- Up to 12 participants message boxes plus 1 facilitator dialog box
- Supports multi languages
- Simultaneous multi language input selection by users
- USB keyboard connection capabilities
- Ease of use – minimal set up and usage learning time
- Anonymity capability – participate without attaching names
- All generated ideas captured in database

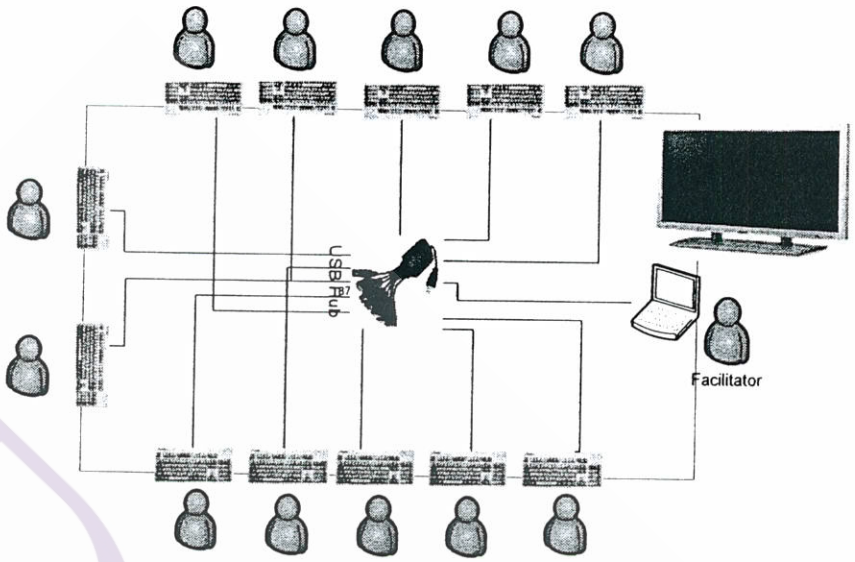


Figure 4.1 BrainEx – Classroom / Group Meeting set

4.3 Development methodology

This research adopted RAD (Rapid Application Development) as the methodology for developing the "BrainEx" application. This is due to the limitation in resources for conducting research. RAD is a mixture of various software development methodologies (Whitten *et al.* 2004). There are two main factors which are key drivers for RAD adoption of this research: time and previous development initiatives.

The time allotted for development was not unlimited. In order to develop this artifact, a variety of technical programming knowledge was required, for instance, USB connection programming and research on alternative and suitable options and Microsoft Windows internationalization capabilities and programming options. Hence, the development needed to be iterative and incremental due to the existence of unknown factors and the insufficient knowledge of a single developer. As such, RAD and prototyping techniques have been adopted to accelerate the development process.

Previous development and study are the main drivers which govern RAD methodology adoption. Single display which is one of the required design features of BrainEx is the first and a major challenge in the early stage of development. Single display is the capability of software that is similar to the traditional blackboard in face to face discussion mode. The difficulty level of single screen display alone can be treated as another project. In fact, single display development has its own field of study. There are two options for achieving the single display development in this research, which is either to develop from ground up or to re-use the existing and available findings from other research studies. No matter which option is chosen, the development technique still has to be iterative and incremental. Because of the two main drivers, RAD has been adopted for this research.

4.4 Development challenge

Since the development of the technology in the mid 20th century, many digital computers, software, peripherals, and communication systems have been made available, such as personal computers, networking, mouse, keyboard, communication ports (e.g., serial, parallel, and USB), operating system (OS) and programming language. The innovations have not only enabled more powerful organizational developments and capabilities like group-work and decision-making but also evolved significantly into various other field studies and have continued to be a significant stage of IS academia. Presented in a previous chapter, a well-known system that supports a group environment reflected in technology evolution is the decision support system (DSS) or group support system (GSS). As mentioned by Arnott & Pervan (2005) "Another dimension to the evolution of DSS is improvement in

technology, as the emergence of each of the DSS types has usually been associated with the deployment of new information technologies" (p. 71). Their report also presents that between 1990 and 2003 research projects about decision support systems mainly focused on information technology and system development. Power (2004) also identified the dominant technology component or driver as the primary dimension of DSS.

In the philosophy dicussed in Chapter 3, this research proposes a new FtF mode Electronic Brainstorming (level 1) artifact by utilizing technology advancement in order to improve the learning processes at the group level of multilingual graduate students. The critical learning processes include interpretation and integration. Interpreting is the step where a shared understanding of ideas occurs while integrating is the part of decision making regarding the ideas to act upon. The group decision-making research challenge stems from many factors, for instance, the variety of different tasks, individual characteristics, and the types of available technologies which may affect idea generation. As such, to design and develop BrainEx, there were critical technical issues which needed to be overcome as follows:

4.4.1 Single Display

BrainEx was designed to be used in face-to-face (FtF) group meetings involved in decision or class room brainstorming in order to support participatory, and equal idea sharing involving two to twelve person groups. To contribute to learning or understanding through a discussion environment, one key factor is active and open view expression through broad participation. A large monitor at the front of the room displaying group information such as vote tallies or idea lists generated during the meeting through a range of procedural control options is necessary. For example,

only one projector connected to the facilitator or technician machine presents all terminals' activities on the screen in real time.

Single display is one of the vital features which supports the FtF mode of group discussion. The basic idea of using a single display is to simulate the look and feel of traditional black or white boards. Although this concept has been developed for many years, the availability of the tools for further development of GDSS is very limited and confined only to a small select group of single display researchers and developers. The tool which has been developed for further development and enhancement is called 'Single Display Groupware - (SDG)'. "Computer programs which enable co-present users to collaborate via a shared computer with a single shared display and simultaneous use of multiple input devices" (Bederson, Stewart, & Druin, 1999). In order to develop SDG for research, advanced knowledge of programming is required and it can be time consuming. Extensive research has been carried out in order to find readymade SDG which will allow public usage and can be enhanced for further research investigations. Only one was found. This SDG was developed in the year 2002 by Edward Tse and Saul Greenberg, Department of Computer Science, University of Calgary and is called the "SDG Toolkit: A toolkit for Rapid Prototyping Single Display Groupware". After seeking permission to use the SDG Toolkit from Edward Tse, permission was granted together with the source code under the Academic Free License. SDG Toolkit has been developed using the C programming language. The limitations of the SDG Toolkit are:

- No support available, only source code is provided.
- No international keyboard input capability.
- Lack of selective keystroke control.

With these weaknesses, it makes it difficult even to do the basic activities of generating an application. For example, when an application requires multiple mice or keyboards plugged into a computer, the developer has to do low-level device manipulation to retrieve and process the various inputs as separate streams. The SDG Toolkit therefore needs to be thoroughly reviewed and enhanced to overcome the previously mentioned limitations or weaknesses. Explanation on technical development will be described later in this chapter.

4.4.2 Simultaneous multi-language input capabilities

One of BrainEx's objectives is to provide the collaborating GSS level 1 tool which provides structure for communication that can enhance the participation of students in a bi or multi lingual co-located (face-to-face) meeting environment. This can be achieved only by providing simultaneous multi-language input capability. Conceptually it does not seem too difficult because most operating systems software (OS) have built-in multi language selection features.

Although multi language selection and inputs are provided in an OS like the Microsoft Windows Platform, UNIX, MAC and others, that only allows one user at a time for each language being used as an input. However, BrainEx aims to provide simultaneous multi-language inputs from multiple keyboards in order to create a continuous idea flow. It is necessary to allow each member to control their individual keyboard input language which allows them to switch to any language they prefer at any time. Simultaneity is therefore not the only key challenge as independent language selection for each keyboard is also a critical challenge. The first stage before developing the artifact involves developers or researchers addressing the issues of rudimentary OS and programming language to complement and extend the regular

OS international language support. Two basic questions arise: What OS should be utilized? What programming language should be employed?

4.4.2.1 What OS should be utilized?

GDSS is moving towards running on Windows operating systems replacing DOS and UNIX environments (Shim *et al.*, 2002).

The *Operating System* is a program that acts as an intermediary between a user of a computer and the computer hardware. The main functions of OS are to control and coordinate the use of the hardware among the various application programs for the various users. The base OS for this thesis was selected based upon the features that were deemed necessary for BrainEx. These features include human–computer interaction (HCI) support, multi-tasking support, multi-language support, and virtual memory support. The HCI is integrated, and this is due largely to the fact that technology alone may not win user acceptance. Normally, interface design seeks to discover the most efficient way to design understandable electronic messages (Shneiderman, 1998). A typical application applies the user interface component to convert technical complexity to usable product. The User Experience, or how the user experiences the end product, is the key to acceptance. Microsoft Windows platform is therefore chosen to achieve all of the defined attributes. Other advantages of MS Windows include a much larger selection of available software programs and utilities, and broader hardware driver support.

4.4.2.2 What programming language should be employed?

The GDSS arena rapidly develops and grows at almost the same pace as technology advancement. It is important to realize that the electronic activities described above are mediated through computer applications that have been

written. This particular form of writing is called computer programming. Programming is also the only process through which the creation of computer software is possible. To communicate with a computer, a developer must use a computer language. Hence, programming language is also a key contributor in terms of IT enhancement for GDSS artifact development. Later development of GDSS artifacts can incorporate new functions and features into its design because of the advancement and evolution of programming languages.

Definition of programming by Gal-Ezer and Harel (1998)

"...we take programming here in a rather broad sense, covering not only the coding act itself, but also the design of the algorithms underlying the programs and, to some extent, considerations of correctness and efficiency. To some, this interpretation of programming might be the obvious one to adopt, but experience shows the point ought to be made more explicitly". (p. 82)

Programming language creation began in the early 1950's. Coding or programming used to be a technical job requiring a lot of time and knowledge for any delivery of computer software. Each progression in technology has produced newer languages to enhance the ability of making a clean coding which empowers and makes programming more effortless, easier to maintain, and more efficient and technically requires less ordering or grammatical consistency. Although there are many programming languages from which to choose, a predominant method has been using Object Oriented Programming (OOP). Object-oriented frameworks are increasingly popular as a means to simplify and automate the development and configuration process associated with complex application domains such as graphical user interfaces (Linton & Calder, 1987, Fayad & Schmidt, 1997), databases (Batory &

O'Malley, 1992), operating system kernels (Campbell, Russo, & Johnson, 1987), and communication sub-systems (Zweig, 1990). The components in a framework typically include a full set of features from *class libraries* and *objects* to build Windows-based applications and enhance reusability. Reusability leverages the know-how and previous experience of developers in defining generic components that can be reapplied to create new applications in order to avoid re-creating and revalidating common solutions (Faryad, 1997).

With the latest Object Oriented Programming language, the researcher or developer can enjoy the reusability features which allow them to share or re-use any available development. Since this research aims to develop the GDSS artifact which can simultaneous input multiple languages into a single screen application, internationalization capability is one of the key considerations for programming language selection.

A programming language that has realized the object-oriented concept is Microsoft C#. It has become one of the cornerstone choices of professional programmers for sophisticated applications. The C# language provides concepts of component-based software technology which defines properties and events or object members of components (Platt, 2001; Archer, 2001). The attribute concept allows programmers to define software component development (Burke, 2000). This object oriented concept allows the C# language to be used together with extended GUI control, and deal effectively with real world problems because the object model can encapsulate real world objects. Objects are combined into namespaces, which allow accessing everything programmatically. In addition, C# has used an allied notion of

internationalization support which implies that computer applications can be adapted to different languages and regional differences (Hejlsberg, 2003).

With the advantages of OOP, the re-usability of SDG which was developed as a C programming object and the capabilities and characteristics of C# are the major reasons why this research employs C# as the programming language to develop BrainEx. An additional brief of C# will be reviewed later in 4.5.

4.4.3 Data differentiation from multiple input devices (multiple USB keyboards)

BrainEx has been designed to enhance face-to-face classroom communication, collaboration and coordination. Investigation in functionality design and development of BrainEx will not be sufficient for software advancement development. The selection of hardware, which comprises not only the personal computer but also the peripheral communication tools to bring about the justified degree of ease of use and ease to find, is also necessary.

From the characteristics of a GDSS, for any set of people who sit together in a meeting room or the space captured by the group through computer support, the system requires numbers of input devices like keyboards to be equal to the number of participants. The key function of each input device is that it must be independently controlled and operated. Furthermore, as described by Mandviwalla and Olfman in 1994, a generic set of GSS characteristics is required to accommodate different group development methods. General requirements include the use of techniques to influence behavioral processes. They also mention that the collaborations sustain different behavioral processes and facilitate different methods of processing information.

Based upon the development of GroupStorm by ADFA, which was one of the GDSS development researches, the researcher had to design and invent a specific physical connector host and particular keyboards to plug in to the host in order to utilize the researcher's group decision application and enable the parallel group communication feature. This might be a barrier for an organization or an institute that would like to implement the software. All related special devices must be built and carried. However, recent evolution of information technologies has removed the barriers of GDSS usage.

In this thesis, the research focuses on designing technology-support for assisting group communication or participation and improving the abilities of learning and idea generation in group performance in international classrooms in Thailand. Recently, a broader range of technologies has been widely used. Technology infrastructure parallels the three eras of growth in the computer industry: the data processing (DP) era, the microcomputer era, and the network era (Nolan and Croson, 1995).

While microcomputers are becoming faster and smaller, one cannot neglect the improvement of personal computer interconnect interfaces that can support simultaneous attachments of multiple portable devices, like Universal Serial Bus (USB) technology.

The compatibility of new hardware interface standards, such as the Universal Serial Bus (USB), opens up opportunities for software developers to build new applications without having to be too concerned about hardware proprietary interfaces. USB is one of those significant technological innovations. An explicit improvement as a result of USB introduction in this research on GSS artifact

development is multi-USB interface keyboards. Keyboards have been used as a general low-cost input device for years but are limited to only one keyboard per PC. This is due to the PC hardware constraint which provides only one PS/2 keyboard port. Therefore, by using USB keyboards, a single PC or laptop computer can have multiple keyboards attached at the same time while previous GDSS artifact development involved the design of special hardware (Multiplexer – MUX) to cater for multiple keyboard connection.

USB is a personal interconnecting device/port that can be plugged into the system at any time. Normally, it is used with many peripherals such as mouse, scanner and printer to improve the plug-and-play capabilities – the ability to use the devices without requiring specific driver installation. Additionally, USB features allow devices to be connected and disconnected without the need of computer rebooting and also possess the ability to provide power to low-consumption devices without the need of an external power supply.

A USB port works differently from a parallel or serial port. Because it operates on a shared bus, it does not have unique port addresses as parallel and serial ports do. When a USB device is attached to a USB port, the operating system requests information from the device and assigns a bus address and a class device driver. Generally, an operating system provides a lower-level driver which manages the details of communication on the bus.

In addition, with the advancement of communication technology using wireless technology, USB wireless keyboards and their receivers work by plugging wireless receivers into any USB port of any computer or USB hub allowing the USB wireless keyboard to communicate with the computer. This allows the computer users

to use the keyboard freely without the hassle of tangled wires. BrainEx has been developed to take advantage of wireless keyboard capabilities. Hence, the advent of USB keyboards has given rise to this research application. They can be plugged into the system at any time and as many as possible through the available USB hosts.

Nevertheless, multiple USB integration is still be one of major technical challenge in BrainEx Development. Some of these technical development challenges are:

- Windows OS will always assign new bus address whenever the new USB keyboard is connected to the USB Hub, therefore, programming mechanism requires in BrainEx in order to resolve this issue in particular to resuming unfinished session so all the idea from previous session can bring up to the right participants for subsequent session.
- When all the connected USB keyboards are simultaneously typed at the same time, how can BrainEx distinguish which information comes from which USB keyboard. This is one of the issues which some GDSS avoid by using special design Multiplexor.

4.4.4 Information overflow and incomplete one-time discussion

Because of multiple and simultaneous idea inputs and the real world of decision-making, these can be potential problems which can be caused from either the continuous data flow into the application via individual limited message display space or the meeting time limitation or discussion topic complexity.

BrainEx core design aims to be an idea generation tool group writer that needs to provide thirteen message boxes for the different sources of information; one from the facilitator and the others from each participant. As mentioned earlier, all input boxes must be displayed within a single window. This is to prevent participant confusion and the interruption of idea flows and idea sharing. The other key point of concern regarding BrainEx's development is that all written or shared messages have to be captured automatically and saved for further analysis and to support decision-making. In a group meeting, participants often enter or generate a large number of ideas simultaneously. The article of Chen *et al.* (1994) mentioned that "in a typical meeting of 10 – 20 participants, several hundred electronic brainstorming (EBS) comments can be generated in less than an hour" (p.58). This extreme amount of information can make the group fail to comprehend what was discussed or may be unable to process the information immediately after successful and effective participation. They may need to spend more time to access their computer assisted discussion and go through the records to organize the ideas into categories and to better understand the meaning of each contributed idea. As a result, automatic logging should emphasize more on the removal of the information flow barrier. Another fact of decision making or learning condition is that it is difficult to end the session by finding a conclusion within a couple hours. Users may need to call or play back their incomplete session or previous information at any time to continue the discussion. As people cannot memorize everything they said or thought, taking notes manually might help, but commonly it does not cover all the content. This automated discussion log will therefore also have to overcome the time impediment and support the development of participant memory from meeting to meeting. To store them with

the ability of idea owner identification, BrainEx integrates a small relational database application of the Microsoft platform - Microsoft Access.

In the step of data retrieval for analysis, BrainEx combines Microsoft Word as a reporting tool and as an aide for idea organization. With the available properties of MS Word, group members can later browse their ideas or comments and learn from reading classmates' contributions during brainstorming. The facilitator can highlight or underline the significant ideas, categorize or group the information, and summarize or conclude the ideas for a final decision.

4.5 Technical resolution of the development challenge

Because of the challenging specification of BrainEx's functionality, the developer needs to understand what functionalities the program is expected to have. Primarily, the components of the solution are identified while designing the input and output methodology and for dividing the components into subtasks. The written algorithms are then initiated to describe how to accomplish each subtask. Finally, each algorithm is coded in the chosen programming language - C#.

4.5.1 Learning about the C# language

C# is one of the main programming languages under the .NET framework. As mentioned earlier, C# provides special language concepts to define properties and events of components. The C# consolidates unique concepts for type-safe function pointers and events (Archer, 2001). Function pointers can be handled like objects. Events in C# are object members and are declared as delegate types.

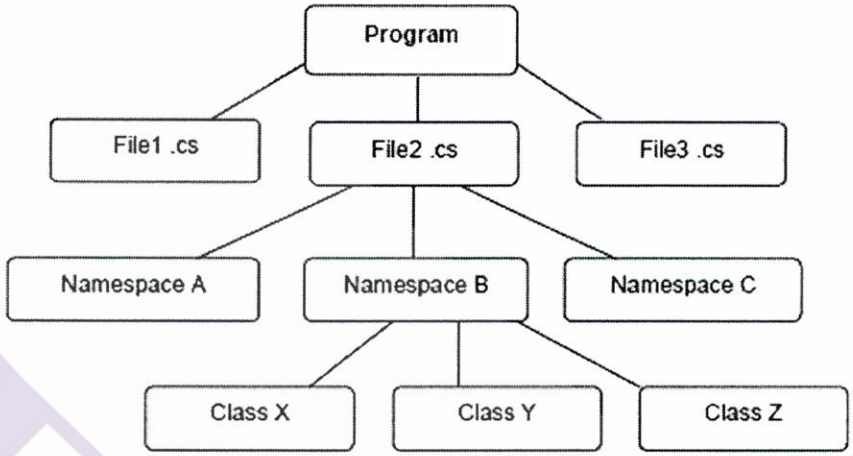


Figure 4.2 C# language architecture

The structure of an application developed in C# consists of files ".cs" that can have unlimited numbers in a project. A ".cs" file contains source codes or programming commands under one or more "namespaces". Practically, varieties of function definitions, such as class, interface, delegate and enumeration are distributed inside a namespace. A significant advancement of C#, compared to other programming languages, is that it bundles listed sets of classes called *class libraries* that can be used to develop Windows-based applications to supporting features. These libraries are analyzed and used for design-time and run-time presentation and for the adaption of the components in BrainEx programming to solve the research problems.

4.5.2 BrainEx Programming

BrainEx is designed to be a general communication media integrated decision support tool that can be adapted by both skilled and unskilled computer users. It is obvious that the application feature is not the only significant factor influencing user performance and productivity. The aspect of user interface

appearance is clearly important. The basic required element of BrainEx is therefore to be a graphic user interface (GUI) or Windows-based application. GUI will help particular users perform their tasks effectively. C# offers several graphical support namespaces such as '*System.Windows.Forms*'; namespace that contains classes for creating Windows-based applications that take full advantage of the rich user interface features together with the *'System.ComponetModel*'; namespace which provides classes used to implement the run-time and design-time behavior of components. These two namespaces have been manipulated in BrainEx in order to extend application GUI appearances, behaviors, and controls.

> An application framework itself is not complete. Developers have to complete or extend the framework abilities by establishing their particular applications. Developers can add their own classes.
>
> (Froehich, Hoover, Liew & Sorenson, 1999; Dallal & Sorenson, 2006)

Namespaces are employed into all eleven program files (.cs) developed in BrainEx: Program.cs, MDIParent1.cs, ItemObject.cs, KeyboardControl.cs, LogInControls.cs, UserSetup.cs, TemplateSetup.cs, NewSession.cs, OpenSession.cs, ViewSession.cs, and DataSetBrain.cs. These developed files include other particular namespaces to gain access to the classes and objects contained within each of them. New object classes are added within these files to be able to respond to underlying research purpose needs included granting the opportunity of international language participation in local areas using only USB keyboards to do the text communication

with each other as well as the support of anonymous/synonymous group processes, structured/unstructured tasks, and recurrence group meetings,

C# language provides many generic tools to support the multi-language issue. The BrainEx program utilizes *'InputLanguage'* class bundling in the '*System.Window.Forms'* namespace. This class provides methods and fields to manage the input language. It is designed to call the *'CurrentInputLanguage'* property to get the current input language and change the current input language to a different language. The input language is dependent on Windows locale identifier. Hence, '*InstalledInputLanguages*' is another property to be called to allow the use of all the installed languages in the operating system. Another import class is also implemented into the application namely *"CultureInfo"*. This class is one of the classes offered in the *'System.Globalization'* namespace that "defines culture-related information, including the language, the country/region, the calendars in use, the format patterns for dates, currency, and numbers, and the sort order for strings" (http://msdn.microsoft.com/en-us/library/system.globalization.aspx). All expressed namespaces and classes are adapted into BrainEx programming to serve the independent language selection.

```
private void KeyboardControl_Load(object sender, EventArgs e)
    {
      //Set Default Input Language
      _languageCulture = InputLanguage.CurrentInputLanguage.Culture.ToString();
    }
/// <summary>
    /// Change language input
    /// </summary>
    public void ChangeLanguage()
    {
      try
      {
        for (int i = 0; i < InputLanguage.InstalledInputLanguages.Count; i++)
        {
          if (this.LanguageCulture == InputLanguage.InstalledInputLanguages[i].Culture.ToString())
          {
            if (i == (InputLanguage.InstalledInputLanguages.Count - 1))
            {
              i = 0;
            }
            else
            {
              i++;
            }

            this.LanguageCulture = InputLanguage.InstalledInputLanguages[i].Culture.ToString();
            InputLanguage.CurrentInputLanguage = InputLanguage.InstalledInputLanguages[i];

            return;
          }
        }
      }
      catch (Exception ex)
      {
        String err = ex.Message;
      }
```

Figure 4.3 Sample of program coding of keyboard control for multi-language input support

In order to advance the quality and usability of BrainEx by allowing the artifact to be able to conduct a variety of simultaneous activities which will help decision-makers overcome their constraints during the group decision making process, the artifact development requires creating new object classes. As a result, new object classes are established for this purpose. Although these new object classes are unique, they can be shared among various functions within the program. This can be achieved by using the "Global variable" declaration. As such, the use of global

variables and mechanisms and the use of synchronized access to shared objects are required. Each module that needs to reference other objects must include definitions for the global variables. For example, BrainEx provides any point of time reporting service by using Microsoft Word. This MS Word application must be defined as a class that can be displayed under the BrainEx window or be called to present as a separated component. Technically, MS Word applications must be announced publicly as a global variable.

The last important element is automated record keeping. BrainEx automatically records information shared and voted by a group into a database file. To serve this automatic log keeping, data sets and typed data sets are integrated. The .NET Framework contains the ADO technology extension (Esposito, 2001) to provide access to external data sources, such as databases. However, it comes with different sets of libraries for different modes to communicate with a data source. These libraries are called 'Data Providers', usually named for the protocol or data source type to interact with. As earlier identified, MS Access is selected to be the data storage application. The 'OLEDB' Data provider is used to connect to and keep data in the database.

```
public partial class OpenSession : Form
  {
    public Word.ApplicationClass wordAppMain;
    public Word._Document wordDoc;
    public Word.Application wordApp;
                          .
                          .
                          .
    private void ButtonWord_Click(object sender, EventArgs e)
    {
      object fileName = "c:\\" + this.Text + ".doc";
      object readOnly = false;
      object isVisible = true;
      object missing = System.Reflection.Missing.Value;
      if (sender != null)
      {
        MessageBox.Show("Please wait while the document is being displayed");
      }
      //Close Word
      try
      {
        objWinWordControl.CloseControl();
        object dummy = null;
        object dummy2 = (object)false;
        wordDoc.Close(ref dummy, ref dummy, ref dummy);
        // Change the line below.
        wordApp.Quit(ref dummy2, ref dummy, ref dummy);
      }
      catch { }
      finally
      {
        objWinWordControl.document = null;
        WinWordControl.WinWordControl.wd = null;
        WinWordControl.WinWordControl.wordWnd = 0;
      }
      try
      {
        //delete old file
        if (File.Exists(fileName.ToString()))
        {
          File.Delete(fileName.ToString());
        }
        //if (WindowWord)
        if (((MDIParent1)this.MdiParent).GetWindowWord())
        {
          //Open word in new window
          object file = "";
          object newTemplate = false;
          object docType = 0;
          isVisible = true;
          wordAppMain = new Word.ApplicationClass();
          wordDoc = wordAppMain.Documents.Add(ref file, ref newTemplate, ref docType, ref isVisible);
          wordApp = wordDoc.Application;
          wordApp.Visible = true;
          //wordDoc.Activate();
          //focus this form
          ((MDIParent1)this.ParentForm).Activate();
        }
        else
        {
          //Open word in this window
          objWinWordControl.LoadDocument("");
          wordDoc = objWinWordControl.document;
          wordApp = wordDoc.Application;
        }
                          .
                          .
                          .
```

Figure 4.4 Sample of program coding of calling MS Word

4.6 Features and functionality of BrainEx

> "The effectiveness of the technology depends on its appropriate design and use by the group."
>
> (DeSanctis & Gallupe, 1987, p. 589)
>
> "Design principles identify the "ideals to be achieved by the selection of specific technologies and the development of particular features [of the IT artifact]."
>
> (Boland, Tenkasi, & Teeni 1994, p. 464)
>
> An approach of design science to research is building and evaluating an artifact to address an existing yet unsolved problem.
>
> (Hevner *et al.*, 2004)

After researched data has been obtained from the empirical GDSS literature review, they have to be analyzed carefully in order to derive answers to research questions and to be presented in research software features. BrainEx is a software application designed especially for multi-native language group decision making and learning. According to) Jacobsen *et al.* (2002), the real challenge is to "develop fluency with teaching and learning with technology, not just with technology, itself'' (p. 44). Rather than to viewing technology as merely a tool for delivery, it should be seen as a means to improve learning

The basic features of the software:

- Easy to use user interface that allows one to twelve people to generate their ideas through their own message box and also enable them to feel like they are chatting. BrainEx is designed to consist with twelve message boxes because group

size makes a difference in the effectiveness of brainstorming groups. Several researches reported that brainstorming was most effective for groups of up to twelve members (Osborn, 1957; Bouchard and Hare, 1970; Fern, 1982a; Hackman & Vidmar, 1970; Lewis, Sadosky, & Connolly, 1975). Information sharing is the most typical aspect of group engaging activities, and most GSS facilities include one public screen to give the group a common point to share public information and a focus point to view the information.

- Allow every user to express in parallel their personal ideas by typing and exchanging written comments simultaneously through the BrainEx GSS in any preferable language (e.g. English, Thai, etc.) that are available or installed in the computer's operating system (OS). As mention earlier, BrainEx GSS was designed and developed to be implemented in the multi-national classroom in order to enhance students' learning and support equal participation. Parallelism allows participants to partake in information exchange at the same time because they can key in messages or ideas simultaneously. This reduces and eliminates time spent waiting for others, in contrast with manual face-to-face meetings. As such, groups interacting within GSS can to an extent avoid process losses or mitigate production blocking that accrue to verbal groups as a result of waiting for their turn to speak (Gallupe, Cooper, Grize, & Bastianutti, 1994; Valacich, Dennis, & Nunamaker, 1991). Furthermore, keying in concurrently can increase participation and group energy. Group members will be able to work through a greater amount of data within a shorter time, process information, see the contributions of others and contribute new information relative to others within the group and may be able to use an idea in a manner that the originator

did not anticipate because participants have different levels of information or language skills.

- Provide functions for the facilitator or team leader who provides direct intervention into the meeting process such as controlling a personal mouse (provided only to facilitator), moving forward to the next topic or backward to a previous topic, and transferring discussion messages or ideas of participants to a simple document and also for participants to control their individual USB keyboards connected directly to a single PC machine.

- Can create a new session either in a structured or un-structured format. BrainEx provides a template setup feature which allows users to define the topic and unlimited sub-topics. This will make the participation proceed structurally.

- BrainEx also allows a moderator / facilitator not to select a template. This will support the un-predefined session. The moderator / facilitator, the person who chairs the meeting, does not need to complete the session in a single discussion. An incomplete session can be re-opened as long as the session status is not changed to "close session".

- A database is provided to memorize all written ideas. All electronic contributions are saved automatically onto disk file which can be recalled or retrieved in real time or any other time and supports the development of organizational memory from meeting to meeting or from class brainstorming to class brainstorming. The facilitator is freed from note-taking and conducting process structure tasks manually or mentally keeping track of what was said. Hardcopy reports can be provided at any stage in the meeting.

- Another vital feature in BrainEx for effective implementation face-to-face GSS mode is the "anonymity" concept. Anonymity or anonymous refers to "the ability of members to input information without attaching an identity to the information" (Vathanaphas & Liang, 2007, p. 1678) which is not possible when contributions are made verbally. With the anonymity component of BrainEx, it confers a number of advantages and can positively affect group outcomes. It eliminates many elements which can limit a group's productivity and produce a higher number of original ideas and more solutions and critical ideas or comments in comparison with non-anonymous systems (Jessup *et al.*, 1990). Anonymity enables group members to increase the level of participation by speaking freely and contributing ideas openly and honestly, especially in the non-participative culture or when the participants feel concerned about their personal security. The fear of embarrassment and social disapproval is greatly reduced. Shy individuals can participate more and generate more ideas because the fear of negative evaluation that can cause them to withhold opinion and information is reduced. With the ability to provide anonymity for participating individuals, significantly improved meeting effectiveness, productivity, and individual participation levels through the reduction of power-distance levels can be achieved. The anonymity and contemporary input features of GDSS support low power-distance cultural norms of group behavior (Watson *et al.*, 1994).

Mentioned in the findings of Tyran's research paper (1997), "Obviously, a student cannot learn from a classmate's contribution if the contribution is not read" (p.627). To support and improve computer-based discussion process and performance, BrainEx provides a technique to combine all discussed messages into

the Microsoft Word application. At any time or any point of the session, the facilitator can open the Microsoft Word application within BrainEx and select to activate the options of either throwing all typed messages of every participants' window into the opening Microsoft Word or allowing specific or all participant(s) to transfer their individual message-by-message into the word document. With this feature, all participants can easier view or read other contributions in a single page document. This may promote or encourage additional ideas from the participants in the form of a response, feedback or support of other ideas.

4.7 Application Interface Design

Now connecting a person to a computer with a cable to transfer information is not enough. People receive inputs mainly in the forms of sight, touch, sound, smell, and taste (collectively called the "5 senses"). Doing the best to make communication between human and computers more effective and more efficient is to implement user interfaces that interact with human senses. As a *GDSS* affects a multi-user group, rather than just an individual, paying more attention to the user/system interface with multi-user access is significant. The researcher has to understand the problem of the subject matter to design the appropriate structure starting from physical, logical, conceptual to operational. This section addresses the issue of user interface design and related issues of quality properties of BrainEx user interface.

User interface development usually involves designing communication and discourse, graphical and textual material, and information and tasks. Johnson

(1992) defined three perspectives to be considered in interface design: functional perspective, aesthetic perspective, and a structural perspective.

1. The *functional perspective* is concerned with whether or not the design is serviceable for its intended purpose. This perspective concentrates on the issues of usability and task completeness.

2. The *aesthetic perspective* is concerned with whether or not the design is pleasing in its appearance and conforms to any accepted notions of design. This perspective concentrates on the design of visual appearance and interface. It considers the layout on the screen, icons, graphical and textual figures, the style of menus and buttons, animation, and interactive video. For example, color is used to assist decision-makers to the perceived information.

3. The *structural perspective* is concerned with whether or not the design has been built in a manner that will make it reliable and efficient to use and can be easily maintained and extended. One solution is the use of object-oriented programming (see detail in 4.4.2.2 section) which enables user interface design to be based on sets of primitive objects that have connections to others.

4.7.1 Interface Design

As a GDSS, BrainEx is concerned with how to provide decision guidance, how to remove barriers of idea generating in multi-language brainstorming and how to help the decision maker utilize the brainstorming ideas efficiently. Based on Johnson's interface designing perspectives and a consideration of GDSS features and human computer interaction concepts, five major interfaces of BrainEx functionalities were developed:

1. Create/Remove facilitator and participants. Building a communication group is the first step for any team-based activity. BrainEx offers an interface for creating or removing communication group members. Before starting a brainstorming session, the instructor can set up username(s) and password(s) including the assignment of the user role, facilitator or participant. After finishing setting up the participant team, the name of group members will be displayed in the use list table. Modifying the existing element is also available if the element is selected and editable. The objective is to support anonymity but to be able to know who is the ideas owner from direct records in the application database.

2. Setup discussion or task template. A fundamental feature of all aspects of GDSS is enabling tasks to be executed in a correct way. In the task of managing GDSS, extending the pre-meeting support or the framework which begins with the agenda setting and includes all the various process elements from brainstorming through to final decision making is necessary. A template can comprise a parent node and multiplex child nodes or unlimited child nodes level. The tree diagram of the created agenda hierarchy is also provided to help the creator not lose sight of the parts of discussion subjects.

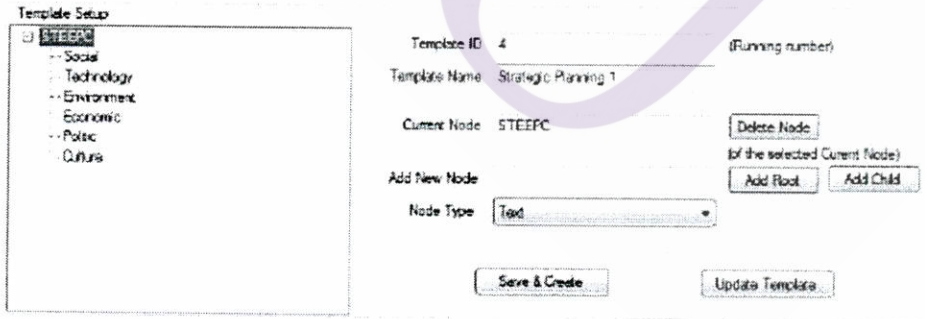


Figure 4.5: Template Setup Interface

3. Establish new discussion session. Due to the consideration of information re-use, whenever the user begins a new series of communication or participation by clicking menu items, there will be a pre-defined data to tell the instructor that the records are available. BrainEx offers three menu items to select discussion group members and facilitators, discussion template, and discussion technique (synonymous or anonymous group member name displaying).

4. Serve parallel multi language text messages expression and consolidate to a common widow. This function is the primary part of every GDSS to provide the interacting channel to the users. Challenges of BrainEx are the ability to provide various brainstorming facilities to satisfy various discussion requirements in a flexible fashion. The facilities in BrainEx include serving twelve text message boxes, freely switching input language at any point of time via personal USB keyboard, no users' memorization needed, and a discussion board to enable each participant views all contents and ideas posed through a common window.

The interface designs consist of twelve individual message boxes containing two small windows; one for typing and the other for re-visiting what they have typed on the left part of screen (see Figure 4.6). Before starting a discussion session, each participant is required to entry their assigned username and password. This is to grant and reserve a message box to a particular user and make them know which message box belongs to them. Each participant is allowed to type his words onto their personal message boxes as long as they prefer. Green and red light is considered as the common color to determine active or inactive status. Another property aiming to provide discussed information dashboard is integrating MS Words as one internal attribute of BrainEx to receive, consolidate and present all recorded

information into one window. The MS Words will be displayed in the right part of screen (see Figure 4.6) if the element buttons are pressed.

5. Continue unclosed discussion sessions. Doing a decision making, it is possible to take more than one time meeting. To support this, BrainEx provides an interface for calling back unclosed or uncompleted session. An important feature of this function is that it provides the list of deployed sessions. User just selects and clicks to continue or re-work; all recorded information will be re-loaded in the original message boxes.

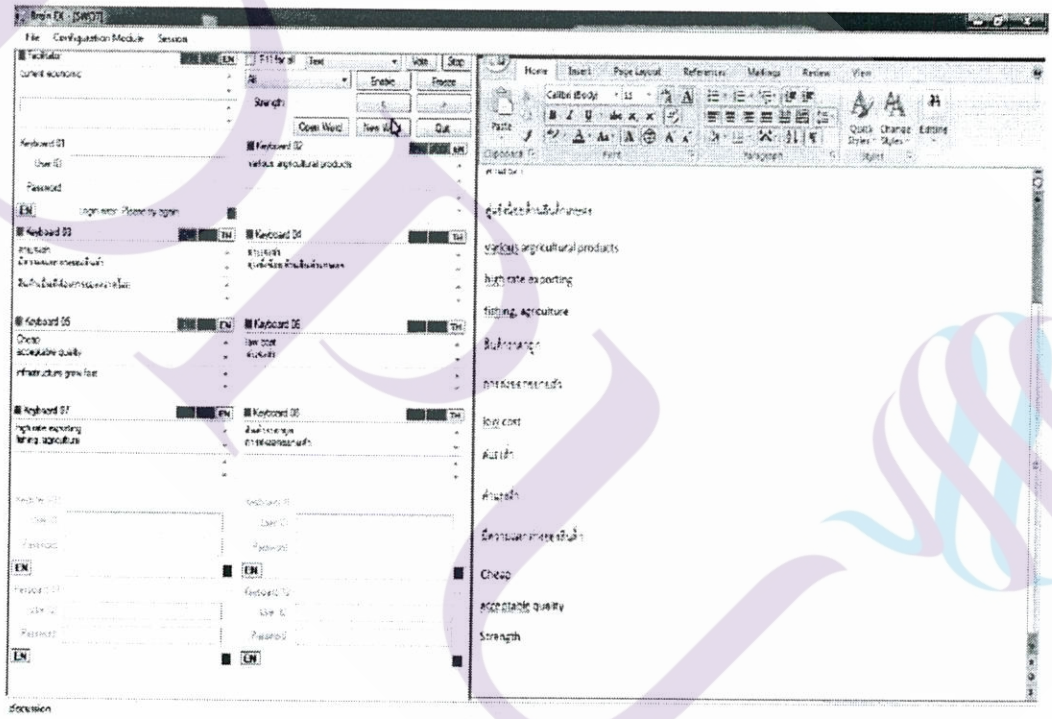


Figure 4.6: BrainEx Discuss Screen

4.8 Conclusion

BrainEx has been developed with the main objective of utilizing a generic GDS approach to enhance the participation of Thai students in a multi-lingual environment classroom. Therefore, not only generic GDSS functionalities are

provided in BrainEx but also, a most important functionality, the capability to capture multiple languages from different keyboards simultaneously. This functionality will allow all participants to participate in face-to-face brainstorming sessions with any preferred language, even their own native language. As a result, the idea flow or generation should be improved due to language barrier reduction. Another major difference of BrainEx to other software available nowadays is fully operational software without requiring any special hardware. This will increase the portability and mobility to the facilitator for setting up group meetings in any location or any room type with low cost. BrainEx is also designed to have many other useful features which provide opportunities for future research in various aspect of GDSS research, for instance, re-usable facilitator templates, anonymous meeting mode which can capture and differentiate each individual set of input ideas after session analysis. In the next chapter, BrainEx will be pilot tested and investigated through experimental tests for further detailed evaluation in actual class room settings.

Chapter 5

Artifact Evaluation

5.1 Introduction

In the previous chapter, a set of design principles to address the challenges of new knowledge discovery was proposed. It is important to understand how such an information system GSS artifact should be designed, developed and used in order to be effectively adopted in international and multi-cultural academic environments. Besides the design science concept, in order to address the research question of this study, artifact evaluation is one of the critical steps in design science research where the main goal is to see whether the developed artifact can contribute as it is intended to or not. Examinations were conducted in Thai universities and concentrated on one specific aspect of group support, "the Face-to-Face group brainstorm". The reason to select brainstorming technique and not any other aspect is because it is a normal class room setting of graduate learning (i.e. interactive discussion based) using both wide- and narrow-band approaches. Adopting Eden's terminology, identified in Chapter 3, the narrow-band approach has been characterized by a research preference for running pilot and experimental, reproducible, laboratory-based brainstorms with students groups, who are working on a problem which is completely new to them. The group members can raise and write down their idea on the whiteboard or computer screen, then, narrow down the ideas to create a potential or finalized strategy. Questionnaires were also used for the statistical analysis of the outputs. Although these two techniques are more positivistic, some phenomena could not be answered or explained adequately by quantitative analysis alone.

The general idea behind the pilot test and the laboratory experimental test reported in this chapter is to evaluate the influence of the design artifact, BrainEx GSS. Chisholm (1994) suggested that the real face-to-face contact cannot be replaced with electronic media alone especially when group efforts are beginning or concluding. Common values of fact-to-face contact up front for individuals not only to reduce the difficulty of sharing risky ideas but also to achieve a final decision from end of a meeting analysis and negotiation. Hence, face-to-face classroom environment setting is the context of an entire workshop and the discussion room space is limited. With the approval from subjects that they are willing to participate before starting the class, this research collected data using an observational approach as well. Qualitative observation data were collected to create a chain of evidence to describe the students' work or learning processes.

A pilot group of participating students had been assigned to work using the STEEPC[1] paradigm, adapted from the STEEPV[2] concept. The laboratory experiments of tests 1 and 2 were discussed using the STEEPV concept. The third experiment was performed using the STEEPC approach. The outcomes of the brainstorming sessions using BrainEx as a discussion tool will be discussed and used in their learning experience.

The following topics are covered in this chapter:

- Design evaluation principle
- Data collection methodology

---

1 STEEPC – Social, Technological, Economic, Environmental/Ecological, Political and Culture-based issues

2 STEEPV - Social, Technological, Economic, Environmental/Ecological, Political and Value-based issues

- Pilot test setting
- Experimental test setting
- Result discussion

5.2 Design evaluation principle

As noted earlier, integrating electronic support into the classroom aims to facilitate the conversation and information flows between students and instructors, and among students (Leidner & Jarvenpaa, 1995), to contribute to the effectiveness of collaborative learning in the classroom (Kwok & Khalifa, 1998), and to improve classroom participation (Walsh, Briggs, Ayoub, Vanderboom, & Glynn, 1996; Tyran, 1997) and learning through discussion (Alavi, 1994). These results are largely based on studies in Western or developed countries. There are few examples about the use of GSS originally developed for enhancing the learning processes and idea generation and idea sharing in developing countries' educational industries such as Thailand.

Considerations of the objective of this research lead to the design and development of a new low constraint on training, space and investment and supportive multi-language groupware artifact – BrainEx GSS. BrainEx was developed based upon the academic research perspective by applying advancement of technology, inexpensive and portable computer system, hardware, and software and including GSS key features that solve the research problem. The research question was initiated to determine the overall effects of the BrainEx GSS among focus groups which consisted of academic level students in international colleges in Asia who normally have their own native language and use the English language as a second language. The investigation was to see whether BrainEx can increase the learning

process and performance; to examine whether the BrainEx system is suitable and can improve Asian or non-participative groups in regards to participation rates and quantity of proposed ideas, and to evaluate whether there is any satisfactory improvement of the focus groups' processes and results. In particular to be covered was the relationship between cultural practice and utilization of GSS attributes among multi-national language students. The assessment of the BrainEx system's utilization effectiveness for a brainstorming discussion-room session for a focus group was therefore conducted.

5.2.1 BrainEx Participants / Unit of Analysis

The unit of analysis is the entity whose characteristics are the focus of study (Corbetta, 2003). It can be an individual, group, company, institution, community or artifact (Baker, 1994). The appropriate unit of analysis choice is dictated by the research objective and question (Baker, 1994), and it is linked to the method of data collection. In accordance with the aforementioned research questions, the unit of analysis of this study is the "international academic students". .

Brain-Ex was implemented in international classroom situations (same time/same place) and was used by groups of graduate level students who made up the focus groups. International graduate students were examined in groups consisting of students from various nations (for example, China, Vietnam, Indonesia, Malaysia, and Thailand) where English is learned as a foreign language. Furthermore, the learning pattern of graduate level education is based not only on gathering knowledge from course books and written communication but also from practical real world examples, business which requires significant exchange of information and knowledge gained from experience and interactions between students themselves as well as from

business professionals. The key focus is to explain the effect of GSS on meaningful learning which overcomes the barriers of information exchange and idea and opinion generation so as to enable students to think more creatively, discuss actively, and get a deeper understanding on a whole range of assigned subjects and problems. The expected deliverables from a GSS include a higher level of focus group satisfaction, enjoyment in the process and outcome, and the effectiveness, quantity and quality of student collaborative learning, and idea generation. The predictions of the proposed BrainEx GSS are verified in a number of classroom situations by setting up four tests; one pilot test, and three laboratory experimental tests. They were held in four different international college classrooms in Thailand. In each situation, twelve Asian Masters students from the Department of Business Administration and one facilitator were organized for the brainstorming sessions. Each focus group comprised both male and female students who have a native language different to English.

5.2.2 The Participation Setup

Appropriate research settings can support the researcher in effectively observing social phenomena, examining proposed theories, and confidently drawing conclusions about empirical tests (Doktor *et al*, 1991; Eisenhardt, 1989). In terms of classrooms or teaching methods, collaborative learning in a face-to-face setting is the general form to gain a highly effective learning strategy (Alavi *et al.*, 1997). Moreover, although there is a variety of laboratory experimental techniques (e.g. Brainstorming, and Assumption Reversals), group brainstorming is the most adopted and studied (Hender, Rodgers, Dean, & Nunamaker, 2001). Group brainstorming is a generic method to generate new ideas around a specific area of interest or problem-solving that is normally used in face-to-face decision-room or group working

sessions. Brainstorming is defined as "a group process in which group members collectively contribute their ideas in a creative atmosphere" (Moran, Talbot, & Benson, 1990, p. 3). The main intention is to remove inhibitions and allow participants to think more freely and move into new areas of thought, create new ideas and solutions, and also improve collaborative learning. By means of carefully prepared questions, participants submit their ideas as they occur to them and then build on the ideas raised by others. All the ideas are gathered without being criticized and collected to provide as many thoughts as possible. There are many papers describing instances where brainstorming groups have generated an impressive number of ideas in a limited period of time (e.g., Osborn, 1957).

All experimental classes were equipped with a computer notebook and twelve USB keyboards, but there were differences in the operating system (OS) usage. BrainEx GSS used in the pilot test and the first laboratory test was installed on Microsoft Windows XP operating system while in the second and third laboratory tests BrainEx was installed on Microsoft Windows 7. They were chosen differently to prove that no matter Microsoft Windows version, the developed BrainEx GSS, from the design science approach, is still able to be implemented smoothly. The desks at which the students sat were arranged in a U-shape to enable face-to-face interaction. Another key feature of the setting was that the software was set up for anonymity. It allowed students to communicate simultaneously (many-to-many pattern) by throwing their individual ideas into the un-named messages boxes. For this treatment, students could see the ideas of each other without knowing whose ideas were presented to them to decrease the ability of group domination and increase the members' contribution.

5.3 Data collection methodology

After the BrainEx GSS had been designed, developed and implemented, output data had to be analyzed carefully and presented in a suitable form otherwise, the results would be useless to readers. Hence, data analysis is a very important phase of this research. Measurements of performance for the focus groups were developed using the combination of positivist and interpretivist approaches.

5.3.1 Positivistic Analysis

In terms of positivism, which corresponds to a quantitative approach; often a controlled laboratory experimental group is involved. Ordinarily, laboratory experiments are used to examine the relationships, causes and effects of a small number of variables on each other. A small number of variables are inspected using quantitative techniques. The quantitative technique associates the statistical correlations or numerical representation of data in order to explain the phenomenon reflected by the data in order to derive answers to the research question. The data collection and measurements came from handing out questionnaires after each brainstorming session. Questionnaires, containing all potential measures, had been distributed in person to a conveniently sampled group of students (Henderson, Giese, & Cote, 2004).

The questionnaires asked members of the groups to rate their opinions and the scores calculated from each part were analyzed to understand the effect of BrainEx GSS on FtF group discussion, learning satisfaction and performance including participation, quality, effectiveness, and the efficiency level of individuals and the group. Within the questionnaire, there are two sets of questions. The first set was to measure the feelings of the focus group on their group discussion activities

through the electronic brainstorming system, BrainEx, using a five point Likert scale. While the questions in the first part were to investigate the participation or discussion quality and process of the group, the questions in the second part were aimed to measure the attitude level of participants toward the assigned task outcomes, and the usefulness of the key features of BrainEx GSS. A seven point Likert scale between 1 = disagreement to 7 = agreement was used to identify the opinions.

In addition, to evaluate the advantage of multi-lingual supportive GSS, counting numbers or quantities are used to:

a) Compare the number of ideas presented in solely English language sessions and the number of ideas presented in combination of local and English language and

b) Compare the numbers of ideas presented in traditional oral brainstorming and the number of ideas from using GSS. This measurement was contributes only in laboratory 2 and 3 sessions

Even though this research type allows close examination of the variables under investigation, it is frequently criticized because it does not reflect the real world as it may be seen as a contrived experiment. However, as this is evaluating a software/hardware artifact and not a qualitative concept, the findings may indicate the advantages of using BrainEx realistically.

5.3.2 Interpretivistic Analysis

Since positivist and interpretivist methods should not be viewed separately, the qualitative data collection method was designed and aligned with the quantitative data collection strategy. This study uses an interpretivistic approach to observe and understand data generated in a natural context by indicating the actions of focus

groups (international program students) according to their own subjective perception, rather than to explain regular or numerical patterns. Recently, a number of researchers (e.g. Easton, Easton, & Belch, 2003) have expanded the domain of GSS to qualitative research by calling for the adoption of GSS to support focus groups.

Focus groups are interactive discussion groups used for generating knowledge and hypotheses, exploring opinions, and identifying and pretesting questionnaire items (Fern, 1982b). In GSS research, focus group methods are seen as a group discussion of a particular topic under the direction of a facilitator who promotes group participation and interaction and manages the discussion through a series of topics (Ackermann, 1996). They are used in testing the effectiveness or efficacy of providing an environment that encourages the expression of different opinions (Morrison *et al.*, 2002) and to investigate the potential benefits of increasing the overall productivity of a diversified focus group (Tse, 1999). Focus groups make explicit use of group member interactions to produce information and insights that would be impossible without the interaction of the group (Stewart & Shamdasani, 1990). Four key purposes of focus groups have been categorized as (Morgan, 1997):

- Using the insights of group members to generate hypotheses,
- Evaluating different research sites or study populations,
- Developing questionnaires or interview schedules, and
- Exploring group members' perceptions and interpretations of results from other studies.

In accordance with the identified properties of focus groups, many GSS researches have applied GSS to focus groups to explore the potential benefits of technology-support to improve the degree of group performance, including larger

quantity and better quality of discussion data, ideas and processes (e.g., Clapper & Massery, 1996; Parent, Gallupe, Salisbury, & Handelman, 2000; Klein, Tellefsen, & Herskovitz, 2007) and in addressing the shortcomings as they provide solutions that may be a part of the group dynamics. Focus groups can be used to gather data through a combination of quantitative and qualitative approaches (Morgan, 1997): the focus group method has therefore also been used in this research to validate measurement. Another main advantage of using focus groups in this research is that the researcher can use the social interaction within the group to encourage more equal participation of participants.

In each class, the observation method was adopted to achieve a more realistic consideration. The observation method is most commonly used for collecting primary data. Using the observation method, the information is sought by the researcher by directly observing individuals without asking for their concurrence. For this study, observational data were collected and notes were kept during the experimental testing to provide a richer picture by providing evidence not found in the positivist analysis and to build a chain of evidence to describe the group learning processes, effectiveness and efficiency of BrainEx and the satisfaction of the participants. All meetings were observed for about two-thirds of the meeting duration. Detailed notes were taken and reports finished within 2 days of each testing. The qualitative data collected from observation were aligned with the quantitative data collection strategy and analyzed in an iterative process of refining the outcomes in the group learning process, effectiveness, efficiency and idea generation. In general, Wendy (1998) suggests "observation of use in the field as an important input to future development"

5.4 Empirical Investigation

To answer the research question presented earlier, an empirical study was conducted. In general, the purpose of an experiment is to test predictions derived from theory and to study relationships under controlled conditions so that confusion in the interpretation of the results can be minimized. There are two major types of experiment: laboratory and field (Benbasat, 1990). Stone (1978) specified the laboratory experiment characteristics as follows: 1) the setting is planned and artificial, 2) participants are assigned randomly to the treatment and control environments, 3) one or more independent variables are manipulated and the impact on the dependent variable of interest is measured, and 4) all other independent variables and intervening variables are controlled. On the other hand, field experiments are different as 1) the setting is in the field instead of being artificial, 2) there is less control over the other independent variables and intervening variables, and 3) participants are not assigned randomly (Shadish *et al.,* 2003). The main benefit of experiments is that they are very strong in enhancing internal *validity*, which refers to the validity of inferences made regarding causality.

Hence, for this thesis, the experimental tests in the laboratory were set. There are several important elements to consider when designing experiments like the method of assignment, subject, task, as well as the use of pilot, pre-tests, and post-tests to understand how the treatment can be used in real world settings. The pilot and laboratory experiment tests evolve through brainstorming groups of international academic students engaged in collaborative learning. The tests were run to investigate whether the new designed GSS – BrainEx, enriched the understanding of the group, as part of the design sciences approach to address the research question and

to see if the results matched the research expectations. This study took an input-process-output approach focusing on the influence of BrainEx on the participation outcomes for the focus groups over a series of same-place-same-time physical configuration brainstorming classroom sessions. Multiple sources of focus groups were used to improve the validity of research question testing. The inputs in this research referred to the independent variables. As discussed in Chapter 2, the most commonly accepted model of GSS suggests four typical GSS inputs; group, task, technology, and context. The process represents the behaviors or the interaction of the participant group during the brainstorming session in a shared environment. Outputs include various measures of dependent variables (e.g., participation rate, group effectiveness, and quality and quantity of the generated ideas).

5.4.1 Independent Variable

The primary objective of the BrainEx development is to study the effect that multi-language and anonymity featured GSS-support has on increasing the overall productivity and satisfaction of a diversified international classroom in Thailand. Focus group members were allowed to simultaneously and anonymously share information on a specific topic. They had an opportunity to generate a list of ideas and use the discussion language independently. Hence, the key independent variables – the treatment conditions – used or considered in this experiment were the available technology features and the participant context or culture:

- Key concerned technology features were using common IT equipment like the USB tool and the capacity of parallel and individual communication language usage on the task performed by the group during each session. All participants in the BrainEx-support focus groups were assigned to the conditions and interacted and

shared ideas relating to the case study questions. Each student's input was entered through their individual keyboard, although not all participants had equal keyboarding skills. Three treatment conditions were used; English, Thai, and combination of both English and Thai, which were used to compare the effect of the tool on individual idea contributions. A minimum of 20 minutes per condition was required.

- The across national boundaries and cultures of the focus groups. This researcher's interests were in exploring how BrainEx GSS affects focus group learning processes and outcomes in non-participative cultures such as Asia. In accordance with the focus group communication styles, power distance, and face saving, the anonymous context of input ideas provided by separate windows for participants to address their different ideas was also adapted as a part of an individual variable.

5.4.2 Dependent Variables

For the manipulation checks, the key dependent variables, the used languages, were measured during the experimental test and at the post-test and compared between treatments. The final product of the task was measured and assessed in three categories:

- Participation rate of the class. Participation was assessed by counting the number of contributions made by students. To evaluate this variable, the group member actions on typing information through the BrianEx application were observed and the total number of generated ideas was counted.

- Usability of the multi-lingual supportive feature. Quantities of records under different language usage options were counted to identify the impact of individual language switching capability of BrainEx on international academic

brainstorming achievement and learning. As well as, observation as the simplest of all usability methods (Nielsen, 1998) was integrated to understand and analyze for information about the true field usability of installed system.

- The task performance covered the quality, effectiveness and efficiency of the outcomes of the assigned task using the BrainEx GSS artifact. This includes the improvement of focus groups' learning processes, attributes of the idea proposed to address the task, the creativity level of ideas and quality of generated ideas.

- The satisfaction of focus groups with the BrainEx GSS. Meeting satisfaction has proven to be an important construct to GSS researchers over a twenty-year period (Fjermestad & Hiltz, 1998-1999). The measures of process satisfaction assessed individual satisfaction with the process of idea generation and quality of contributions and learning by using both questionnaire results and observation during the discussion. Satisfaction with the brainstorming process is measured by the following question: How would you describe your group's participation process? (Scale anchored with 1 = satisfied and 5 = dissatisfied) and the satisfaction of the brainstorming outcomes was measured by the following three statements: (1) The outcome of the discussion was? (Scale anchored with 1 = unsatisfactory and 5 = satisfactory), (2) People in the case who would be affected by our conclusions would probably be satisfied with them? (scale anchored with 1 = strongly disagree and 7 = strongly agree), and (3) I am satisfied with my group's conclusions and assumptions scale anchored with 1 = strongly disagree and 7 = strongly agree).

- Most measures of product creativity are subjective in nature, relying on the judgment of an expert panel (e.g., Amabile, 1983; Couger & Dengate, 1992; Ocker, Hiltz, Turoff, & Fjermestad, 1996). An objective analysis of product

creativity is challenging since it requires a measurable set and verifiable criteria (Amabile, 1983; Couger & Dengate, 1992). Accordingly, the potential benefits of ideas were assessed in this study by two independent evaluators after the completion of the experiment. Measuring and comparing the results of two independents evaluators across the three treatment conditions helped to understand the effect of BrainEx features towards research problems or dependent variables. Hence, the research outcomes can have at least partially contributed to the knowledge for improving the quality of similar brainstorming or learning session settings.

INPUT	PROCESS	OUTPUT
Features of GDSS- BrainEx (Supportive multi-language, Physical & Anonymous settings)	Brainstorming Interaction Assigned brainstorming topic	Idea Generation / Participation rate Usability Classroom Learning
International students (Multi-language & culture)		Effectiveness & Efficiency Satisfaction

Table 5.1 Research Variables

5.4.3 Sample

To investigate the potential influences of the BrainEx GSS system features on the brainstorming situation processes, usability and performance (quality, effectiveness, and efficiency) and on user satisfaction, user testing with real users is the most fundamental method. Four experimental tests were conducted on focus group classrooms. Because one of BrainEx main features is supporting multi-language participation in a face-to-face environment, the efforts in each laboratory workshop were undertaken with multi-national students in an academic classroom

setting. The sample included a maximum of twelve students and one facilitator. Some participants were doctoral students and some of them were Master's students. All samples were mixed gender groups and the ages ranged between 22 and 55. None of the participants had previously participated in a computer-based brainstorming experiment; therefore, none were familiar with GSS. Individuals were randomly assigned to mixed gender groups in limited space, and network access environment.

5.4.4 Task

Empirical sessions were run in classrooms or in a decision-room (same time/same place) for three to four hours. The BrainEx-focus groups were given the same tasks and performed the tasks in the same sequence. For this study, the topic "what would Thailand look like in 25 years" was discussed using the STEEPV/STEEPC conceptual approach. The specific task involved a future scenario of Thailand in which focus groups were assigned to brainstorming and generate ideas using the STEEPC / STEEPV dimensions:

- Social (S),
- Technology (T),
- Economics (E),
- Ecology (E),
- Politics (P), and
- Culture (C) / Value (V).

All groups were provided with an introduction to the objective of the focus group, what topics would be covered and how the outcomes of the groups would be used by the researcher. Treatment conditions were assigned and could be changed

before each particular session relating to the topic began. Each participant in the group had to complete the experiment.

Participants were given approximately twenty minutes to complete the brainstorming on each issue of either the STEEPC or STEEPV model. The communication mode that they would be discussing these issues in was explained to the participants at the beginning of the class. Participants were asked to use BrainEx to discuss the assigned topic. They were told to use and switch different languages supported by the tool to help generate their ideas during the brainstorming session.

5.5 Pilot test

A small pilot study was conducted in 2009 with twelve graduate students, as participants, from the Master of Business Administration program, Dhurakij Pundit University International College. The participants were Asian students who have been described as collectivistic and having high power-distance cultures. All of them have their own native language and use the English language as a second language. The BrainEx (version 1.0) was installed on a stand-alone computer notebook and run in a fifteen square-meter meeting room to support the synchronous face-to-face brainstorming classroom. Twelve USB keyboards did connect to the notebook via several USB hubs. The computer was connected to a single screen projector and was controlled by the facilitator. All participants were free to move to different seats during the session.

Before starting the session, brief information and the procedures of the application usage were given to the participants in ten minutes. Since the BrainEx was designed based on easy-to-use apprehension, both facilitator and participants

were not required time to learn or train how to use the hardware and software. A pre-defined template was created to support and structure the session at the topic level. Unstructured sub-topics were given for each topic during the discussion. The session was clustered into 6 themes for analytical thinking about current and future situations using the STEEPC framework (i.e., society, technology, economics, environment, politics, and culture).

Anonymous contribution was used in the testing to reduce social status cues or avoid the idea resistant. The users can see the data without knowing who originated the ideas. At the start of the session, the facilitator had signed up all twelve students with individual usernames and passwords. After logging on to the application, BrainEx randomly assigned a message box for each keyboard and the message box names were identified as keyboard 1, keyboard 2, …, keyboard 12. All participants were allowed to submit brainstorming ideas through the individual keyboard appeared on individual window on the screen. The course instructor and BrainEx facilitators worked together to plan the GSS-supported class. The first set of tasks was to 'Identify the present STEEPC situation in Thailand'. The second task was to 'Identify the future STEEPC scenario of Thailand in next 15 years'. The instructor controlled the time and managed the task flow. Members of the group learn the concept of brainstorming technique and share ideas through the BrainEx system. The first topic was appointed to type in English, and all the rest were in either English or Thai. The questionnaire survey which was to analyze student evaluations and was also used to analyze test scores was piloted. The feedback and results from this pilot test were important for the laboratory experimental test's success. It allowed the facilitator to practice both the conceptual and operational changes and to work on

some technical tuning of the BrainEx system. Conceptually, the facilitator learned how to guide the classroom brainstorming and discovery. Operationally, the facilitator learned new techniques of using BrainEx's available features in order to enhance the information sharing during the discussion session, such as using the active F11 function key which allows participants to type their messages directly to the Microsoft Word document. This helped all the participants to view their own and others' input. For functional fine-tuning, the researcher learn how the report structure should be designed and organized by utilizing the Microsoft Word template feature to handle multi-level discussion topics so that the generated report can be in a readily usable format for subsequent activities, i.e. content analysis.

5.6 Laboratory experimental setting

The laboratory experiments were set up with the aim to implement a newly designed GSS with a multi-language supportive feature, BrainEx, in an international classroom environment to examine the influence of the system on performances of the group brainstorming process and outcomes. Participants' satisfaction of the discussion outcomes and usability of the available features of BrainEx at the both individual and group level were investigated. The reaction and participation rates of the students were also monitored.

5.6.1 Experiments Test 1 & 2:

The first experiment was conducted four months after the pilot study. The second experiment was carried out four months later. The aspects were similar to the pilot study. Both experiments were setup as simulated real classroom environments and used multiple equivalent focus groups – international academic program students

in Thailand universities. There were twelve international students, one facilitator, and one course instructor in each laboratory class that was held in a small classroom. It should be noted that all the participants in experiments 1 and 2 had English as a second language. In experiment 1, nine students were Thai (75%), two were Chinese (17%), and one was Vietnamese (8%). In experiment 2, ten were Thai (84%) and the others are Vietnamese (16%). The desks were arranged in a U-shape to enable a face-to-face environment among the students. The group of students and instructor in experiment 1 were not the same as in experiment 2, but the GSS facilitator was the same. The course instructor and GSS facilitator had worked together in preparing the course material and planning for the class process. Every lab experiment began with the instructor of the class using a standard script to explain the purpose and importance of the session and the instructions that were to be followed when brainstorming or generating ideas. The main instructions were to 'generate as many ideas as you can based on your experience and understanding of the assigned topic. All ideas will be discussed, investigated and summarized later when all the series of discussion sessions are complete.' All groups were assigned the same task involved asking each participant to present ideas to evaluate the current and future situations in Thailand using the six themes of STEEPV.

To understand how a computer-based classroom benefits and aids the focus groups in knowledge creation and group discussion performance, experiments 1 and 2 were built on two treatment designs. The two designs were one, traditional classroom brainstorming where group members verbally generated ideas and two, GSS support where group members used the BrainEx system to generate, share, and record ideas. All focus groups performed both treatments by dividing each testing into two

sections. Starting with the traditional approach, all the brainstorming was orally conducted. The instructor directly led the session and allowed 15 to 20 minutes for each theme discussion. Beginning of the session, the instructor announced that there is no enforcement to take part in the discussions. Everyone was allowed to participate as many times as they would like to. However, due to the class was mixed between Thai and foreign students, this environment required the students to discuss in English only and had to raise their hands to speak. In the case of several students raised their hand at the same time, the instructor prioritized one-by-one to present the idea. The facilitator recorded those ideas on the white board at the front of the room.

This was then followed by the GSS-support treatment, using the same technology setting and support in both experiments. BrainEx (version 1.1) software was run on a computer notebook where twelve keyboards were connected via USB hubs. A large screen projection system was also attached, so that all participants could see all the posted ideas of every participant in real time. Two input languages were enabled in the computer; Thai and English. To switch the input language, the 'grave accent' keyboard button was used as a hot key. The key features of BrainEx were used. F11 and F12 hot keys to enable pushing typing messages to MS Word are activated. The anonymous feature was preserved to verify the reflection of students' activities on idea generation and idea sharing compared with no BrainEx support, because group members did not know whose ideas were being presented to them. A pre-defined STEEPV template was initiated to provide the structural discussion and output report. A parallel communication channel was active which allowed all group members to generate and post their ideas in their individual message boxes at any time without the need to raise their hands. A group facilitator had taken

approximately ten minutes to instruct all participants what are and how to use these key functions. The facilitator was stationed at the head of the room and managed the BrainEx system and the experiment procedures and also assisted participants in its use. The same set of twelve students using the traditional approach was randomly appointed to the provided keyboards. They were allowed to spend approximately 15 to 20 minutes to complete each STEEPV sub-topic in the determination of Thailand's future situation. To comprehend how multi-language supportive function increases the volume of recorded ideas, the participants were assigned to use only English language in the Social (S) theme, but for other themes they were allowed to use mixed languages of English and Thai.

5.6.2 Experimental Test 3:

The third experiment took place in the year 2010. The primary objective of this test is to confirm the research question results. The general physical environment was similarly constructed as experiments 1 and 2; a computer, twelve USB keyboards, and a projection device to display the contents of the computer screen. The key properties of the selected focus group remained. Mixed genders of twelve international graduate students participated. There were seven Thai (58%), three Chinese (25%), and two Japanese (17%) enrolled in the class.

The course instructor and GSS facilitator had worked together in planning the class processes. The instructor planned to assign the task involved generating personal ideas of the current and future situations regarding Thailand using the six themes of STEEPC. Therefore, the STEEPC template was prepared with the activation of the anonymous methodology feature. Due to the characteristics of the anonymity feature, BrainEx randomly appointed a keyboard for each message box,

and the facilitator agreed with the instructor to do the sign-in for all seats before the class started. As the primary group members are Thai, beside English, the common language, Thai language was agreed to be added into input language list. The generic hot key to switch the input language 'grave accent' keyboard button was selected.

The experiment began with the facilitator giving brief information and guidelines on how to use BrainEx, and then the instructor of the class explained the purpose and importance of the session and instructed about what students had to brainstorm or generate ideas. All ideas were recorded and used for investigation. These ideas were summarized later when the series of discussion sessions were completed. A group facilitator was stationed at the back of the room and managed the BrainEx system and assisted participants in its use.

Ideas of the current situation using STEEPC were generated first. The length of each session was approximately one and a half hours – 15 minutes per theme. To promote learning via interaction, after finishing the Social theme, the F11 feature that allows participants to send their typed messages into an open Microsoft word document was activated. Students could view their classmates' opinions or ideas on a single document page. The input made by the others could be built upon generating additional knowledge and comments.

5.7 Post-test

Understanding and evaluating the relevance of the results is an important step in the knowledge-discovery process. One of the challenges with knowledge discovery is that the individual must perceive the discovery results or outcomes as relevant and important (Schulz, 2001). Design science is focused on developing a

class of solutions to address the class of problems identified in the environment. The effectiveness of a design science approach is usually evaluated against the effectiveness of other technologies or designs (Venable, 2006).

To evaluate the outcomes, there are several methodological pitfalls. A set of questionnaires had been developed and handed-out to all participants after the series of the brainstorming sessions was completed. During the classes, observations and note-taking were also made to elaborate the outcomes and gain usability insights. Recording file of user interaction sequences help by showing why the users paused or generated low number of ideas. After the brainstorming session was over, all the ideas were summarized and evaluated.

5.7.1 Overall Results of the Focus Group Discussion / Learning Process:

The first examination of the research questionnaire was analyzed as to whether there was any positivity in overall performance and satisfaction with the focus group process or focus group results through the new designed and developed GSS. Assessing BrainEx in terms of the qualitative observations of the focus group's brainstorming process and the tool usability, a similar pattern to pilot testing could be observed when analyzing the focus groups' participation rates where the groups undertook the same task. It was obvious that the BrainEx - GSS could easily implement and use. It took average thirty minutes to setup and configures the hardware and software, and took ten minutes for usage training. It allowed students to freely express their thoughts or ideas. They could contribute whatever they wanted to in any wording and in any language they wanted to. All students enjoyed typing their ideas in without stopping to wait for other group members to finish typing their ideas. Seeing other ideas on the screen during discussion, the students could emphasize the

ideas they also went along with and the new ideas came up all the time. Within the assigned timeframe (10 – 15 minutes) for single discussion, over fifty ideas about each of the discussion topics or sub-topic were contributed. Those ideas or keywords were directly inputted and stored in the BrainEx.

After class, students were assigned to construct a briefing report. The feedback from the students had been identified in the report indicating that they were very satisfied with this application. Simultaneous input encourages students to participate equally. In addition, both the multi-language support and the anonymous feature of the BrainEx software were keys in making the focus groups participate enthusiastically in the discussions. Participants had more confidence and enjoyment to contribute their ideas and generate knowledge represented by the total number of ideas. For example, one student of pilot group specified that "At first, I do not think the idea will come too much but after typing and see the other idea on the screen, the new idea come up. This is very useful for us to develop the scenario…The BrainEx tool is a wonderful tool, I felt not shy to typing the idea down because the other did not know which one is mind and one more thing I could type in Thai language too (see detail in Appendix II).

Nevertheless, the group in experiment 1 was marginally more satisfied with the process than those in experiment 2. In all cases, the results of the exploratory study positively supported the predictive research question which was 'Can a generic international group of academic students improve their group learning performances and participation, and be satisfied with the classroom environment through the use of multi-language supportive GSS?'

Table 5.2: The participation rate of BrainEx GSS and Non-BrainEx GSS brainstorming

Variable	Topic	Usage time (minutes)	Numbers of generated ideas (each theme of STEEPV)	% of participants (n = 12)
Traditional/Oral or non-GSS Brainstorming	Social	20	26	67
	Technology	20	29	84
	Economic	15	22	67
	Environment	15	20	58
	Political	15	16	58
	Cultural	15	18	67
GSS Brainstorming	Social	20	48	100
	Technology	20	59	100
	Economic	15	52	100
	Environment	15	52	100
	Political	15	42	100
	Cultural	15	50	100

From the experimental test 1 and 2 results, Table 5.2 show variance in independent variables (GSS or non-GSS) and explains the variance in the dependent variables (idea generating / participation rate and usability). One-hundred percent (100%) of participants (n = 12) input their ideas when they used BrainEx while average only sixty-six percent (66%) raised their hands to share ideas when the participation was done without the BrainEx support. With the same amount of time, the GSS session could record more than two times the ideas than traditional brainstorming. These indicate that anonymity and supportive parallel communication in BrainEx GSS lead to better perceptions of participants of the sense behind the idea and therefore not stopping the idea flow and this can reduce many of negative effects of verbal meeting. Qualitative observation of students' behavior and reactions in

traditional setting also support the indication. Several students avoid raising their hands. They just nodded in assent if they agreed with other ideas or just whispered words.

In addition to understanding more on the effect of multi-language input capability of the BrainEx, any records taken of how much use participants made of the feature were reviewed. Comparing numbers of language usage, 79.8% of records were in English language. Only 20.2% were in Thai. Even though English language gained more percentage than Thai language, when analyze deeply into the specific topic. This study found in difficult topic that under the environment, politic, and culture or value areas, which are required special wording/terms that users were not familiar with or were not generally used in daily life Thai language were used more, such as **รัฐบาลไม่ทำตามแผนยุทธศาสตร์ของชาติ** (government did not follow country strategic plan), **การเช่าพระเครื่องปลอม** (buying fake Buddha image), **ความแห้งแล้ง** (drought condition), **ความไม่มีน้ำใจช่วยเหลือซึ่งกันและกัน** (heartless), and **เด็กไม่นอบน้อมต่อผู้ใหญ่** (disrespect to elderly). With this information, multilingual supportive GSS can definitely be useful to increase and improve baseline information for group participants on developing the suitable scenario.

From the questionnaires scoring calculation of the first set of questions that mainly focus on the reflective feelings of the participating students with their group brainstorming process including the quality, efficiency, effectiveness and satisfaction using the BrainEx application, all focus groups were asked to rate their perceptions of whether the use of the BrainEx made their group discussion perform positively in regards to quality, effectiveness, and efficiency through the questions in

the questionnaire. Satisfaction with the idea generation process is also rated. Noted from observation,

In terms of overall quality, efficiency and effectiveness, the results from each experiment showed no significant differences from the overall results of these three measures between the experimental groups, and the statistical analysis provided marginal support for the research hypothesis which meant multi-language supportive GSS can increase the quantity of information or knowledge sharing. Participants in each focus group using the BrainEx scored significantly higher; the average means and standard deviations of all experiments were 3.80 and 0.75 respectively (see detail in Appendix VI). Similar to the results of satisfaction of all focus groups for the brainstorming process using BrainEx to generate ideas and knowledge, there were small differences in this measure. Table 5.3 shows all groups were significantly satisfied with their brainstorming process. Although those groups in experiments 1 and 2 were marginally more satisfied with the process than those in the pilot test and experiment 3, this might be because the groups had first brainstormed in a traditional way before brainstorming through BrainEx. Nevertheless, these also evidently supported the assumption in this research that features in BrainEx improved the participation rate of students. Moreover, the average mean of the conducted experiments was 4.02 and average standard deviation was 0.63 which also statistically supported the research assumption (see detail in Appendix VI).

Table 5.3 Overall measures of the STEEPV/C brainstorming session process (S indicates supportive research question)

Outcomes	Laboratory Experiments							
	Pilot (n12)		Lab 1 (n12)		Lab 2 (n12)		Lab 3 (n11)	
	M	S.D.	M	S.D.	M	S.D.	M	S.D.
Observation								
High participation rate	S		S		S		S	
Questionnaire								
Perceived quality, efficiency and effectiveness of brainstorming process	3.77	0.70	3.80	0.80	3.82	0.70	3.79	0.81
Perceived satisfaction of brainstorming process	3.88	0.69	4.13	0.64	4.08	0.70	4.00	0.63

5.7.2 Perception towards the Overall Brainstorming Outcomes from BrainEx session

The second set of questions was to gather information regarding users' attitude towards the resolution of group learning or discussion outcomes from the brainstorming session using the BrainEx GSS and the usability of BrainEx. A similar pattern of each BrainEx user groups' expressions or opinions was recognized after analyzing the group's perception of the outcomes of the focus group. The groups had rated their agreements on the discussion performance outcomes of the assigned tasks, on the satisfaction of the outcomes, and on the key features of BrainEx GSS to overcome the language and cultural contexts.

The results in Table 5.4 indicate there was no significant difference in the three measures of quality, effective, and efficiency with the brainstorming outcomes between the four focus groups. Groups in the experiments 2 and 3 (Exp 1 & 2), however, gave slightly higher mean scores than those groups in experiment 3 and the pilot test. This might be because the participants in Exp 1 & 2 participated in a traditional brainstorming session (without using BrainEx) so they intuitively compared the results from BrainEx session with the results from traditional F-to-F brainstorming mode. Overall results indicate that the majority of participants of all groups were confident in generating ideas and thought the conclusion arrived at by their group from the information in BrainEx was efficient which supports the research proposition.

Focus groups were also asked to identify their individual satisfaction level as to whether the use of BrainEx support made them more satisfied with the outcomes from the generated ideas. Analysis of the questionnaire feedback from the four experiments showed that there were no statistical differences in the scores between groups in their satisfaction with the focus group brainstorming results. Groups significantly agreed with their groups' conclusion which was derived from the posted ideas in the BrainEx.

Involving the multi-language support concept, which is the major feature of the BrainEx, there were no significant different results between the groups. Analyzing the means of the results in Table 5.4 (see detail in Appendix VI), those focus groups reported high levels of agreement on the positive effects of a GSS which provides the ability of language selection to support the groups in their idea generating flow. Interestingly, participants in all groups agreed that the opportunity

of using either international or native language at any point of the brainstorming can help them to overcome the language barrier or difficulties and allow them to contribute more to the discussion or brainstorming session without worrying about language capabilities or skills. The overall results also indicated and strongly supported the positive effect of using BrainEx in the brainstorming session.

Table 5.4 Overall measures of the STEEPV/C brainstorming session outcomes

Outcomes	Laboratory Experiments							
	Pilot (n12)		Lab 1 (n12)		Lab 2 (n12)		Lab 3 (n11)	
	M	S.D.	M	S.D.	M	S.D.	M	S.D.
Questionnaire								
Perceived quality, efficiency and effectiveness of brainstorming outcomes	4.61	1.05	4.99	1.09	5.03	0.99	5.08	1.08
Perceived satisfaction of brainstorming outcomes	4.96	0.90	5.25	0.75	5.33	0.87	5.32	0.86
Usability of multi-language supportive GSS	5.39	1.08	5.52	1.14	5.55	1.13	5.56	1.17

5.8 Conclusion

This chapter reported on the pilot and laboratory experiments conducted using a new GSS artifact - BrainEx to answer the research question – "Can a generic international group of academic students improve their group learning performances and participation, and be satisfied with the classroom environment through the use of multi-language supportive GSS?" Four focus groups took part in order to examine the use of BrainEx and evaluate the effects of the major features such as anonymity and multi-language support of the BrainEx GSS application, designed and developed from

design sciences research methodology for focus groups. The tasks which the focus groups covered were both idea generation and evaluation activities. All international program student groups completed their tasks in classroom face-to-face brainstorming environments (same time / same place) using the dimensions of STEEPC / V. Analysis was focused on taking both quantitative and qualitative assessments.

The results of integrating the BrainEx GSS into the classroom laboratory experiments, using both observation and questionnaire statistical data collection techniques, show that when looking at the overall participation process performance, two dimensions of participant satisfaction: satisfaction with the brainstorming process and the degree to which participants were satisfied with the brainstorming outcomes, and the usability of BrainEx to overcome the communicating language barrier, that the findings were consistent across all four focus groups. There is little difference in the means and standard deviations between the groups. These numbers provide strong support to the thesis research question which was "Can advanced multi-language supportive GSS improve group learning and performance and the satisfaction of participants" and the objective of developing the new GSS artifact – BrainEx.

The next chapter concludes the thesis. Learning about research methodologies and the areas of GSS development are articulated. Limitations of the current version of the BrainEx are identified. Additional application features for greater effectiveness of group processes and outcomes using the technology-support are also investigated.

Chapter 6

Conclusion

6.1 Introduction

This chapter summarizes the materials in this thesis, research findings, and major contribution, so that the limitations of the research and recommendations for future research can be determined.

At the start of this research paper, an introduction to the area of group support system (GSS) or group decision support system (GDSS) evolution, characteristics, and empirical researches were presented. Even though GSS as a research field is not new, it still has several gaps to be investigated. This current research represents one of new emerging information system (IS) research paradigms known as 'Design Science'. The important of the Design Science approach in GSS research was described. With the knowledge contributed by the Design Science perception, this study can bridge the gap between the GSS and multi-nation or multi-native language users which has prevented the effective and efficient use of GSS especially in face-to-face (F-to-F) or same-time and same-place discussion room in a non-English speaking environment.

Since there are more and more international academic study programs in both western and non-western countries and also an improvement of technological hardware and software, BrainEx, an IT artifact, has been designed and developed to meet the needs of focus groups who have different native language requirements and who are from non-participative cultures. The focus group concept was identified and how the focus group used the artifact for group participation, brainstorming or

meeting was presented. This study had adopted the groupware GSS BrainEx into the educational process where FtF group discussion is the common environment setting in order to answer the research question: Can a generic international group of academic students improve their group learning performances and participation, and be satisfied with the classroom environment through the use of multi-language supportive GSS?

The key features of the software included: 1) a parallel communication channel (through individual USB keyboards) which allows all classmates to participate equally at the same time, 2) anonymity which allows the participants to safely display their ideas, 3) multi-language switching which allows each participant to freely select and switch their communication language at any point of brainstorming, and 4) group idea recording which allows all electronic messages to be saved and made available after the brainstorming session. Several methods and techniques for the evaluation of BrainEx GSS were then introduced, used and tested to find out the answers for the research question. Of all the evaluation methods, emphasis made on heuristic techniques, laboratory experiment settings, user observation analysis and questionnaires were chosen and explored in detail. The details of pilot and laboratory experiments that made use of focus groups were presented. Open observations and survey questionnaires as the ways to getting direct and unbiased information from the members of the GSS supported groups were used for not only group performance evaluation but also for users' satisfaction evaluation.

Normally, GSS is regarded as a computer-based system that provides an interface to a shared environment. It is designed to support groups of people engaged in a common task or idea exchange. Therefore, the performance and user satisfaction must be evaluated if quality, efficiency, and effective communication, discussion

process and outcome are desired. In addition, for this study, it is equally important to evaluate the usability of the significant feature of BrainEx which is the multi-language support. The property of the multi-language support feature makes BrainEx different from existing GSS. It allows each participant synchronously to share their ideas in any preferable language including freely switching the language at any point of the discussion through their own portable and low cost input machines - USB keyboards that are connected to a single computer.

Information collected from utilizing BrainEx into classroom brainstorming environments was collected and investigated using both quantitative and qualitative data collection approaches. Open or participant observation technique, one of the popular qualitative approaches, can provide effective monitoring and measures of performance and usability. Relevant information regarding this technique was presented comprehensively. Measures of performance provided by open observation tracking were discussed as well as the participation rates of members in focus groups.

Besides the observation approach, the questionnaire and its statistical analysis is another technique which can enhance BrainEx performance measurement. User observation and questionnaire techniques are both powerful, but emphasize different aspects of performance evaluation. While the participant observation method focuses on the meaning of the participant existence as seen from the standpoint of insiders (Spradley, 1980), questionnaire focuses on personal expression or opinion as seen from the given answers. Therefore, it is better to combine both to gain better understanding rather than using only a single method. This study distributed the same set of questions to all participants after finishing the brainstorming session. Many questions representing the same measurement criterion

were asked to the evaluators to emphasize the results to gain more valuable information and a thorough understanding from questionnaire responses.

In addition, users' satisfaction on the application usage and outcomes is also an issue concerning many information technologists. If users are not satisfied with the application, it may mean that the application is not as effective as the technologists' expectancy. Each and every user satisfaction must be evaluated in one way or another. Asking users' the right questions via questionnaire techniques was used to evaluate their satisfaction of the discussion process and the outcomes from the electronic brainstorming session which incorporated BrainEx.

6.2 Summary of the Main Results

This study began by examining the advantages of GSS for educational purposes especially in an international face-to-face classroom environment. After understanding the key concepts of GSS application and the gaps of integrating GSS effectively and efficiency into an Asian environment where Asian people have a native language different from English and therefore need to use English as the second language, a new GSS artifact was developed to close the gap. To investigate whether the application can perform up to the research expectation or not, one pilot and three experiments were utilized, corresponding to the two evaluation techniques chosen previously. In the pilot experiment, it was discovered through user observation that users are highly motivated to participate in class. As members of this focus group are mostly Thai students, it appears that they are entirely appreciative of the language switching feature. The number of ideas was continuously posted into every message box in both English and Thai. This presents success in reducing the

perceived blocking of ideas. In addition, from the questionnaires, participants tend to prefer the performance of the discussion process through the use of BrainEx GSS. As well as being evident from the results of the satisfaction questionnaires, the students liked the electronic classroom experience, the way their group performed, and the results from their brainstorming. Each of the satisfaction indicators received high scores. These findings were later supported by users' preferential responses in all three experiments.

As the testing procedure of experiments 1 and 2 are slightly differently from the pilot experiment, the findings show slightly different scores of both means and standard deviations. The experiments were conducted into two treatment designs; traditional verbal group brainstorming and BrainEx GSS support group brainstorming. Comparing electronic idea generation sessions to the traditional oral idea generation session, overall results obtained from these exploratory studies support the research expectation. The numbers of ideas / knowledge generated in BrainEx GSS support session are greater and the participation rates are significantly higher. It was also uncovered in these experiments that the applicability of both anonymity and freedom of communicating language selection lead to higher student satisfaction with this type of brainstorming environment as was evident from the results of the satisfaction questionnaires. The scores are greater than the scores from the pilot and third experiments which the focus groups work only in GSS support treatment. There is no language difficulty situation to be compared.

The results of the final experiments support the purpose of BrainEx. BrainEx was found to influence the brainstorming process performance, usability and the satisfaction of the users.

Overall, there was strong support for BrainEx, as a means of producing or improving classroom learning / discussion process and user satisfaction on both dimensions of satisfaction with the electronic brainstorming process and satisfaction with the discussion outcomes. However, the research also highlighted some special issues to be taken into consideration by GSS developers.

6.3 Limitation and Suggestions for Future Research

The main feature of BrainEx about multi language support was developed by adapting the multi-language support capacity of the Microsoft Windows operating system. Therefore, to use BrainEx, the software installation must be done only on a Windows platform machine. Furthermore, when users would like to switch from default or first language to the other languages, the alternative languages must be pre-installed or activated in the Windows language option. Another limitation that happens from depending on Windows language abilities and can possibly block the ideas generation flow is the case of installing more than two languages. The more languages installed, the more ideas are blocked. The user is required to press the hot keys for the input language (e.g. Grave Accent) till the required language is activated. The differences in native languages of the different participants and the feature of non-network and Internet access required GSS can also be other obstacles that may cause lower BrainEx productivity and usage e.g. the class members were a combination of Chinese, Vietnamese, and Thai. Once a Thai participates or inputs Thai characters into the application, the Chinese and Vietnamese do not in most cases understand the sentence. Even though, real time language translation or similar

features can be investigated to overcome this issue. The present state of online real time translators is not efficient enough to use yet. The accuracy still varies.

A number of facets of the research undertaken indicated several directions for future research on the topic covered by this study. More detailed explanations of some phenomena identified in this study, for example differences in the native languages of group participants, the physical setting of the BrainEx system including the Internet access capability, or the minimum level of language translation accuracy can be considered by future research concentrating on these issues, regarding the usability of the BrainEx system. To enhance BrainEx GSS technology more efficiently and effectively designed and implemented, language translation online functions, such as Google Translate supporting 51 languages in 2,550 language-pair combinations (Aiken *et al.*, 2009), may be investigated together the importance of translation accuracy with the topic and task determination for example medical or legal matters require more attention than do informal, ad hoc communication. Appreared in Fügen *et al.* (2007) study, in some cases, interpreters might be expected to be 80% accurate in meetings, and in other situations, such as intelligence analysis, just 40% accuracy might be enough (Caulfield & Reeder, 2001).

Finally, last limitation would be the questionnaire which can be improved by adding more open-ended questions in order to cross check and avoid any bias issues that could cause from scoring questionnaire.

References

Ackermann, F., & Eden, C. (1994). Issues in computer and non-computer supported GDSSs. *Decision Support Systems*, *12*, 381-390.

Ackermann, F. (1996). Participants' Perceptions on the Role of Facilitators Using Group Decision Support Systems. *Group Decision and Negotiation*, *5*, 93–112.

Ackermann, F., & Eden, C. (2001). Contrasting Single User and Networked Group Decision Support Systems for Strategy Making. *Group Decision and Negotiation, 10,* 47-66.

Ackermann, F., Morton, A., & Valerie, B. (2003). Technology-driven and model-driven approached to group decision support focus, research philosophy, and key concepts. *European Journal of Information Systems*, *12*, 110 – 126.

Adkins, M., & Brashers, D.E. (1995). The power of language in computer-mediated groups. *Management Communication Quarterly*, *8*, 289-322.

Adler, N. J. (1991). *International dimensions of organizational behavior* (2nd ed.). Boston: PWS-Kent.

Adler, N. J., & Ghadar, F. (1989). Globalization and human resource management. In A. M.Rugman (Ed.), *Research in global strategic management: A Canadian perspective* (pp. 179-205). Greenwich, CT: JAI Press.

Aiken, M. W. (1993). GDSS: theory and practice at Mississippi. *ACM SIGOIS Bullentin*, *14*, 56-57.

Aiken, M.W. (2002). Multilingual communication in electronic meeting. *ACM SIGGROUP Bulletin*, *23*, 18-19.

Aiken, M.W., Balan, S. & Hazarik, B. (2011), Exploring The Feasiblity of Mobmile Multilingual Electronec Meetings. *Issues in Information Systems, XII*, 16-22.

Aiken, M.W., Mahesh, V. & James, K. (1995). Group decision support systems. *Review of Business*, *16*, 38-42.

Aiken, M. W., Martin, J.S., Paolillo, J.G.P., & Shirani, A.I. (1994). A group decision support system for multilingual group. *Information & Management*, *26*, 155-161.

Aiken, M. W., Martin, J. S., Reithel, B., Shirani, A., & Singleton, T. (1992). Using a group decision support system for multicultural and multilingual communication. *Proceedings of the 23rd Annual Meeting of the Decision Sciences Institute, 2,* 792-794.

Aiken, M.W., Martin, J.S., Vanjani M., & Sexton R. (1994-1995). Group decision support systems in higher education. *Journal of Educational Technology Systems*, *23*, 3–12.

Aiken, M., Park, M., & Lindblom, T. (2010). Integrating machine translation with group support systems. *International Journal of Business and Management*, *5*, 25-35.

Aiken, M. & Vanjani, M. (2009). Polyglot: A multilingual group support system. *Issues in Information Systems, 10*, 101-106.

Aiken, M., Wang, J., Gu, L., & Paolillo, J. (2011). An exploratory study of how technology supports communication in multilingual groups. *International Journal of e-Collaboration*, *7*, 17-29.

Alavi, M. (1994). Computer-mediated collaborative learning: An empirical evaluation. *MIS Quarterly*, *18*, 159–174.

Alavi, M., Yoo, Y., & Vogel, D.R. (1997). Using Information Technology to Add Value to Management Education. *Academy of Management Journal*, *40*, 159-174.

Al Dallal, J. & Sorenson, P. (2006). Generating Class-Based Test Cases for Interface Classes of Object-Oriented Black Box Frameworks. *World Academy of Science, Engineering and Technology*, *16*, 90 -95.

Alderton, S. M., & Frey, L. R. (1986). Argumentation in small groups decision-making. In R. Y. Hirokawa & M. S. Poole (Eds), *Group Decision-Making and Communication* (pp. 157–173). Beverly Hills, CA: Sage.

Amabile, T.M., (1983). The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, *45*, 357-376.

Arnott, D., & Pervan, G. (2005). A critical analysis of decision support systems research. *Journal of Information Technology*, *20*, 67-87.

Antunes, P., & Ho, T. (2001). The Design of a GDSS Meeting Preparation Tool. *Group Decision and Negotiation*, *10*, 5-25.

Archer, T. (2001). *Inside C#*, MS Press.

Attwell, P., & Rule, J. (1984). Computing and organizations: What we know and what we don't know. *Communication of the ACM*, *27*, 1184-1191.

Au, Y.A. (2001). Design Science I: The Role of Design Science in Electronic Commerce Research. *Communications of the AIS, 7,* Available at: http://aisel.aisnet.org/cais/vol7/iss1/1.

Baek, J. K. (2007). Exploring the Assessment of Geological Observation with Design Research. *ProQuest Dissertations & Theses*, *60*(01). (UMI No. 3277152)

Baker, T. (1994). *Doing Social Research* (2nd ed.). McGraw-Hill: Singapore.

Ball, N.L. (2001). Design Science II: The Impact of Design Science on Electronic Commerce Research and Practice. *Communications of the AIS,* 7, Available at: http://aisel.aisnet.org/cais/vol7/iss1/2.

Bannan-Ritland, B. (2003). The role of design in research: The Integrative Learning Design Framework. *Educational Researcher*, *32*, 21-24.

Barab, S., & Squire, K. (2004). Design-based research: Putting a stake in the ground. *Journal of the Learning Sciences, 13*, 1-14.

Batra, D., Hoffer, J. A., & Bostrom, R. P. A. (1990). Comparison of User Performance between the Relational and the Extended Entity Relationship Models in the Discovery Phase of Database Design. *Communications of the ACM*, *33*, 126-139.

Batra, D., & Marakas, G.M. (1995). Conceptual data modeling in theory and practice. *European Journal of Information Systems*, *4*, 185-193.

Batory, D., & O'Malley, S.W. (1992). The Design and Implementation of Hierarchical Software Systems Using Reusable Components. *ACM Transactions on Software Engineering and Methodology*, *1*, 355–398.

Bederson, B. B., Stewart, J., & Druin, A. (1999). *Single Display Groupware* (Technical Report CS-TR-4086). University of Maryland, USA.

Benbasat, I. (1990). Laboratory experiments in information systems with a focus on individuals: A critical appraisal, in I. Benbasat (ed.). *The information systems research challenge: experimental research methods* (pp. 97-111). Harvard Business School, 33-47.

Benbasat, I., & Lim, L.H. (1993). The Effects of Group, Task, Context and Technology Variables on the Usefulness of Group Support Systems: A Meta-Analysis of Experimental Studies. *Small Group Research*, *24*, 430-462.

Benbasat, I., & Nault, B. (1990). An Evaluation of Empirical Research in Managerial Support Systems. *Decision Support Systems*, *6*, 203–226.

Benbasat, I., & Zmud, R.W. (1999). Empirical Research in Information Systems: The Practice of Relevance. *MIS Quarterly, 23*, 3-16.

Benbasat, I., & Zmud, R.W. (2003). The identity crisis within the IS discipline: Defining and communicating the discipline's core properties. *MIS Quarterly*, *27*, 183-194.

Bernard, H. (2000). *Social Research Methods: Qualitative and Quantitative Approaches*. Sage: London.

Bhabha, H. L. (1994). *The location of culture*. UK, Routledge.

Bingimlas, K. A. (2009). Barriers to the Successful Integration of ICT in Teaching and Learning Environments: A Review of Literature. *Eurasia Journal of Mathematics, Science & Technology Education,* 6, 235-245.

Blanning, R.W., & Reinig, B.A. (2002). Multiperiod scenario development using group support systems: An application to the future of Hong Kong. *Journal of Organizational Computing and Electronic Commerce*, *12*, 105-119.

Boland, R.J., Tenkasi, R.V., & Teeni, D. (1994). Designing Information Technology to Support Distributed Cognition. *Organization Science, 5*, 456-475.

Bouchard, T. J., & Hare, M. (1970). Size, performance, and potential in brainstorming groups. *Journal of Applied Psychology*, *54*, 51-55.

Bonczek, R. H., Holsapple, C.W., & Whinston, A.B. (1981). *Foundations of Decision Support Systems*. Academic Press, New York.

Bradac, J. J., & Mulac, A. (1984). A molecular view of powerful and powerless speech styles: Attributional consequences of specific language features and communicator intentions. *Communication Monographs*, *51*, 307-319.

Bradac, J. J., & Street, R.L. (1990). Powerful and powerless styles of talk: A theoretical analysis of language and impression formation. *Research on Language and Social Interactions*, *23*, 195-242.

Briggs, R.O., de Vreede, G.J., & Nunamaker, Jr., J.F. (2003). Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems. *Journal of Management Information Systems*, *9*, 31-63.

Briggs, R.O, Ramesh V., Romano N.C. JR., Latimer J. (1994-95). The Exemplar Project, *Journal of Educational Technology Systems*, *23*, 277-291.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, *38*, 275-280.

Brown, H.D. (1994). *Principles of Language Learning and Teaching* (3rd ed.). Englewood Cliffs, NJ: Prentice Hall Regents.

Burdett, J. (2000). Changing Channels: Using the Electronic Meeting System to Increase Equity in Decision Making. *Information Technology, Learning, and Performance Journal*, *18*, 3-12.

Burke, S. (2000). Creating Designable Components for Microsoft Visual Studio .NET Designers, Microsoft Corporation, msdn.microsoft.com/library.

Burrell, G., & Morgan, G. (1979). *Sociological Paradigms and Organizational Analysis*. Heinemann: London

Burstein, F., & Gregor, S. (1999). The Systems Development or Engineering Approach to Research in Information Systems: An Action Research Perspective. *10th Australasian Conference on Information Systems*, 122-134.

Callan, J. L., & Whymark, G. K. (2002). From boardroom to mainstream classroom: Educational informatics to enhance instruction and learning. *Paper presented at the 5th World Multi-Conference on Systems, Cybernetics and Informatics*, Orlando, FL.

Campbell, R., Russo, V., & Johnson, G. (1987). The Design of a Multiprocessor Operating System. in *Proceedings of the USENIX C++Workshop*, 109–126.

Caulfield, H. & Reeder, F. (2001). Evaluation of endogenous systems. MITRE Technical Papers. Retrieved from http://mitre-corp.org.

Chen, H., Hsu, P., Orwig, R., Hoopes, L., & Nunamaker, J.F. (1994). Automatic Concept Classification of Text From Electronic Meeting. *Communications Of The ACM, 37*, 57-73

Chen, W., & Hirschheim, R. (2004). A paradigmatic and methodological examination of information systems research from 1991-2001, *Information Systems Journal, 14*, 197-235.

Chieochan, O, Lindley, D., & Dunn, T. (2003). The adoption of information technology: a foundation of Ecommerce development in Thai culture. In T. Thanasakit, *E-Commerce and Culture Values* (pp. 17-50). IDEA Group Publishing.

Chisholm, J. (1994). Electronic meeting systems, *Unix Review, 12*, 13-18.

Clapper, D. L., & Massey, A. P. (1996). Electronic focus groups: A framework for exploration. *Information and Management, 30*, 43-50.

Clawson, V.K., Bostrom, R.P., & Anson, R. (1993). The Role of Facilitator in Computer Supported Meetings. *Small Group Research, 24*, 547-565.

Cobb, P., Confrey, J., diSessa, A., Lehrer, R., & Schauble, L. (2003). Design experiments in educational research. *Educational Researcher, 32,* 9-13.

Cook, T. D., & Campbell, D. T. (1979). *Quasi-Experimentation: Design and Analysis Issues for Field Settings*. Boston, MA, Houghton Miffin.

Connolly, T., Jessup, L. M., & Tansik, D. A. (1990a). Toward a Theory of Automated Group Work: The Deindividuating Effects of Anonymity. *Small Group Research, 21*, 333- 348.

Connolly, T., Jessup, L. M., & Valacich, J. S. (1990b). Effects of Anonymity and Evaluative Tone on Idea Generation in Computer-Mediated Groups. *Management Science, 36*, 689-703.

Corbetta, P. (2003). *Social Research: Theory, Methods, and Techniques*. Sage: London.

Corbitt, B. J. (1999). Exploring the social construction of IT policy-Thailand and Singapore, *Prometheus, 17*, 309-321.

Couger, J. D., & Dengate, G. (1992). Measurement of Creativity in I.S. Products, in Hawaii, *International Conference on Systems Sciences*, 288-298. Lihue: HI.

Davis, G. B., & Olson, M. H. (1985). *Management Information Systems: Conceptual Foundations, Structure and Development* (2nd ed.). McGraw-Hill: New York.

Dawar, N., Parker, P.M., & Price, L. J. (1996). A Cross-Cultural Study of Interpersonal Information Exchange. *Journal of International Business Studies*, *27,* 497-516.

Denning, P. J. (1997). A New Social Contract for Research. *Communications of the ACM, 40,* 132-134.

Dennis, A. R. (1996). Information exchange and use in group decision making: You can lead a group to information but you can't make it think, *MIS Quarterly*, *20,* 433-457.

Dennis, A. R., & Gallupe, R. B. (1993). A History of Group Support Systems Empirical Research: Lessons learned and future directions, In L.M. Jessup and J.S. Valacich (Eds.) *Group Support Systems: New Perspectives* (pp. 281-293), New York: Macmillan.

Dennis, A. R., George, J. F., Jessup, L. M., Nunamaker, J. F. Jr., & Vogel, D. R. (1988). Information technology to support group meeting. *MIS Quarterly*, *12,* 591-624.

Dennis, A. R., Tyran, C. K., Vogel, D. R. & Nunamaker, J. F. (1997). Group support systems for strategic planning. *Journal of Management Information Systems, 14*, 155–184.

Dennis, A. R., Valacich, J. S., & Nunamaker, J. F., Jr. (1990). An experimental investigation of the effects of group size in an electronic meeting environment. *IEEE*, *20,* 1049-1057.

Dennis, A.R., and Wixom, B.H. (2002). Investigating the Moderators of the Group Support Systems Use with Meta-Analysis, *Journal of Management Information System*, *18,* 235–257.

Denzin, N. K. & Lincoln, Y. S. (1995). Editor's introduction. *Qualitative Inquiry*, *1*, 3-6.

Delbecq, A. L., Van deVen, A. H., & Gustafson, D.H. (1975). *Group Techniques for Program Planning: A Guide to Nominal Group and Delphi Processes*. Scott-Foresman & Company, Glenview, Illinois.

DeSanctis, G. (1993). Theory and Research: Goals, Priorities, and Approaches, *MIS Quarterly*, *17*, vi-viii.

DeSanctis, G., & Gallupe, R. B. (1985). Group decision support systems: A new frontier. *Data Base*, *16*, 3-10.

DeSanctis, G., & Gallupe, R. B. (1987). A foundation for the study of group decision support systems. *Management Science*, *33*, 589-609.

DeSanctis, G. L., & Poole, M. S. (1987). Group decision making and group decision support Systems (*Working paper MISRC-WP-88-02*). Management Information Systems Research Center, University of Minnesota, Minneapolis.

Desanctis, G., & Poole, M. S. (1994). Capturing the Complexity in Advanced Technology Use: Adaptive Structuration Theory. *Organization Science*, *5*, 121-147.

Deveney, B. (2005). An investigation into aspects of Thai culture an its impact on Thai students in a international school in Thailand, *Journal of Research International Education, 4*, 153-171. Dickson, G. W., Partridge, J., & Robinson, L. (1993). Exploring Modes of Facilitative Support for GDSS Technology. *MIS Quarterly*, *7*, 173 -1 94.

Dickson, G. W., Robinson, L., Heath, R. & Lee, J. E. (1988). Observations on GDSS Interactions: Chauffeured, Facilitated, and User-Driven Systems. *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences*, *3*, 337-343.

Diener, E. Fraser, S. C., Beaman, A. L., & Kelem, R. T. (1976). Effects of deindividuation variables on stealing among Halloween trick-or-treaters. *Journal of Personality and Social Psychology*, *33*, 178-183.

DiSessa, A. A., & Cobb, P. (2004). Ontological Innovation and Role of Theory in Design Experiments. *The Journal of The Learning Sciences*, *13*, 77-103.

Doktor, R., Tung, R., & Glinow, M. (1991). Future Directions for Management Theory Development. *Academy of Management Review*, *16*, 362–366.

Duff, A. (1989). *Translation.* Oxford. Oxford University Press.

Easton, G., Easton, A., & Belch, M. (2003). An experimental investigation of electronic focus groups. *Information & Management*, *40*, 717-727

Edelson, D. C. (2002). Design Research: What We Learn When We Engage in Design. *The Journal of The Learning Science*, *11*, 105-121.

Eden, C. (1995). On evaluating the performance of wide-band GDSS. *European Journal of Operational Research, 81*, 302-311.

Eisenhardt, K. (1989). Building Theories from Case Study Research. *Academy of Management Review*, *14*, 532-550.

Eom, S.B. (1999). Decision Support Systems Research: Current state and trends. *Industrial Management and Data System*, *99*, 213-220.

Esposito, D. (2001). ADO .NET for the ADO Programmer, MSDN Online Library, Microsoft Corporation, msdn.microsoft.com/library.

Fayad, M., & Schmidt, D. (1997). Object Oriented Application Frameworks. *Communications of the ACM*, *40*, 32-38

Fern, E. F. (1982a). The use of focus groups for idea generation: The effects of group size, acquaintanceship, and moderator on response quantity and quality. *Journal of Marketing Research, 19*, 1-13.

Fern, E. F. (1982b). Why do focus groups work: a review and integration of small group process theories, In A. Mitchell (Ed.), *Advances in Consumer Research* (pp. 444-451). Association for Consumer Research, St. Louis: MO, 444 – 415.

Ferre, F. (1988). *Philosophy of Technology*, Englewood Cliffs, NJ: Prentice Hall.

Fjermestad, J., & Hiltz, S. R. (1998). An Assessment of Group Support Systems Experimental Research: Methodology and Results. *Journal of Management Information Systems*, *15*, 7-149.

Fox, N. J. (2005). Cultures of Ageing in Thailand and Australia. (What Can and Ageing Body Do?). *Sociology*, *39*, 481-498.

Fredman J, Horndahl M, & Strömberg LT. (1999). *A Framework for the Design of Organizational Decision Support Systems*, Department of Informatics, Göte-borg School of Economics, Goteborg University.

Froehich, G., Hoover, J. H., Liew, W., & Sorenson, P. G. (1999). Application framework issues when evolving business applications for electronic commerce, *Information Systems*, *24*, 457-473.

Fügen, C., Waibel, A., & Kolss, M. (2007). Simultaneous translation of lectures and speeches. *Machine Translation, 21*, 209-252.

Gahr, A.A. (2003). Cooperative chemistry. *Journal of College Science Teaching*, *32*, 311-315.

Gal-Ezer, J., & Harel, D. (1998). What (Else) Should CS Educators Know?. *Communications of the ACM*, *41*, 77-84.

Galliers, R.D. (1994). Relevance and rigour in information systems research: Some personal reflections on issues facing the information systems research community. In B.C. Glasson, I.T. Hawryszkiewycz, B.A. Underwood and R. Weber (Eds.). *Business process re-engineering: Information systems opportunities and challenges* (pp. 93-101). Amsterdam: Elsevier North-Holland.

Gallupe, R. B. (1985). *The Impact of Task Difficulty on the Use of a Group Decision Support System*. Unpublished doctoral dissertation, University of Minnesota.

Gallupe, R. B., Cooper, W. H., Grize, M. L., & Bastianutti, L. M. (1994). Blocking Electronic Brain-storming. *Journal of Applied Psychology*, *79*, 77-86.

Gallupe, R. B., Dennis, A., Cooper, W., Valacich, J., Bastianutti, L., & Nunamaker, J. (1992). Electronic Brainstorming and Group Size. *Academy of Management Journal*, *35*, 35-369.

Gallupe, R. B., & McKeen, J. D. (1990). Enhancing Computer-Mediated Communication: An Experimental Investigation into the Use of a Group Decision Support System for Face-to- Face Versus Remote Meetings. *Information & Management, 18*, 1-13.

George, J. F, Easton, G. K. Nunamaker, J. F., & Northcraft, G. B. (1990). A Study of Collaborative Group Work With and Without Computer-Based Support. *Information Systems Research*, *1*, 394-415.

Gergen, K. J., Gergen, M. M., & Barton, W.N. (1973). Deviance in the dark. *Psychology Today*, *7*, 129-130.

Gero, J. S. (1999). Research Methods for Design Science Research: Computational and Cognitive Approaches, *ANZAScA Conference*: Sydney.

Ginsberg, M. and Wlodkowski, R. (2000), *Creating Highly Motivating Classrooms for all Students: A Schoolwide Approach to Powerful Teaching with Diverse Learners* San Francisco: Jossey-Bass.

Glynn, M.S., Quintana, J., Cunningham, D., & Dr. Cooper, S. (1998). GSS For Jury Deliberations: Applying Technology in the High School Courtroom, *IEEE*, *1*, 309-317.

Goodman, P.S. (1986). *Designing Effective Work Groups*. San Francisco: Jossey-Bass.

Gravemeijer, K. (1994). *Developing Realistic Mathematics Education*. Utrecht: Cdß-Press.

Gray, P., Vogel, D., & Beauclair, R. (1990). Assessing GDSS Empirical Research. *European Journal of Operational Research, 46*, 162-176.

Guzzo, R.A. (1986). *Group Decision-Making and Group Effectiveness in Organizations*, In P.S. Goodman and Associates, *Designing Effective Work Groups* (pp. 34-71). San Francisco, CA: Jossey-Bass Inc.

Haake, J., & Wang, W. (1997). Flexible support for business processes: Extending cooperative hypermedia with process support, in *Proceedings of the International ACM SIGGROUP - GROUP97, 341-350.*

Hackman, J. R., & Charles, G. M. (1975). Group Tasks, Group Interaction Process, and Group Performance Effectiveness: A Review and Proposed Integration, In L.

Berkowitz (Ed.). *Advances in Experimental Social Psychology* (pp. 45-99). Academic Press, New York.

Hackman, J., R., & Vidmar, N. (1970). Effects of size and task type on group performance and member reactions, *Sociometry, 33*, 37-54.

Hales, M. (1995). Information Systems Strategy, a Cultural Borderland, Some Monstrous Behavior. In: S. L. Star (Ed.). *The Cultures of Computing* (pp. 103–117). Oxford, UK: Blackwell Publishers.

Hejlsberg, A., Wiltanumth, S., & Golde, P. (2003). *The C# Programming Language*. 2nd Microsoft Corporation, MA.: Pearson Education, Inc.

Hender, J. M., Rodgers, T. L., Dean, D. L., & Nunamaker, J. F. (2001). Improving Group Creativity: Brainstorming Versus Non-brainstorming Techniques in a GSS Environment. *In Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS'01)*, Hawaii, USA.

Henderson, P., Giese, J., & Cote, J. (2004). Impression Management Using Typeface Design, *Journal of Marketing*, *68*, 60-72.

Hevner, A. R. (2007). A three-cycle view of design science research. *Scandinavian Journal of Information Systems*, *19*, 87–92.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Sytems Research. *MIS Quarterly*, *28*, 75-105.

Hew, K. F. & Cheung, W. S. (2003). Evaluating the Participation and Quality of Thinking of Pre-Service Teachers in Asynchronous Online Discussion Environment: Part I. *International Journal of Instructional Media*, *30*, 247-262.

Hiltz, S.R., Johnson, K., & Turoff, M. (1991). Group Decision Support: The Effects of Designated Human Leaders and Statistical Feedback in Computerized Conferences. *Journal of Management Information Systems*, *8*, 81-108.

Hirokawa, R. Y., & Pace, R. A. (1983). A Descriptive Investigation *at* the Possible Communication-based Reasons of Effective and Ineffective Group Decision Making. *Communication Monographs, 50*, 363-379.

Hirokawa, R. Y., & Poole, M. S. (1996). *Communication and Group Decision Making*, 2nd Beverley Hills, CA: Sage Publications.

Hitchman, S. (1995). Practitioner perceptions of the use of some semantic concepts in the entity relationship model. *European Journal of Information Systems, 4*, 31-40.

Hobday, M. (2000). East versus Southeast Asian innovation systems: comparing OEM- and TNC-led growth in electronics. In: Kim, L., Nelson, R. (Eds.), *Technology, Learning, and Innovation: Experiences of Newly Industrializing Economies* (pp. 129–169). Cambridge University Press, Cambridge.

Hodgetts, R. (1993). A Conversation With Geert Hofstede. *Organizational Dynamics*, *21*, 53-61.

Hoffman, L. (1979). *The Group Problem-Solving Process*, Praeger, New York.

Hofstede, G.H. (1980). *Culture Consequences: International Differences in Work-related Values*. London: Sage Publications.

Hofstede, G.H. (1983). The cultural relativity of organizational practices and theories. *Journal of International Business Studies*, *Fall,* 76-88.

Hofstede, G.H. (1984). Cultural dimensions in management and planning, *Asia Pacific Journal of Management*, *1*, 81-99.

Hofstede, G. H. (1991). *Cultures and organizations: Software of the Mind*. New York: McGraw-Hill.

Hofstede, G. H. (1993). Cultural constraints in management theories. In J.T. Wren (Ed.). The leader's companion (pp. 253-270). New York: Free Press.

Hofstede, G. H. (1997). *Cultures and organizations: software of the mind*. London: MaGraw Hill.

Hofstede, G. H. (1998). Think Locally, Act Globally: Cultural Constraints in Personnel Management. *Management International Review*, *38*, 7–26.

Hofstede, G. H., Hofstede, G. J. & Minkov, M. (2010). *Cultures and Organizations Software of the mind*, The McGraw-Hill. NY.

Holloway, I. (1997). *Basic Concepts for Qualitative Research*. London: Blackwell Scientific.

Holme, I. M., & Solvang, B. K. (1991). Forskningsmetodik. Om Kvalitation och Kvantitativa metoder, Lund, Studenttlitterratur (In Swedish).

Holmes, H., & Tangtongtavy, S. (1997). *Working with the Thais: A Guide to Managing Thailand*. Bangkok: White Lotus Press.

Holsapple, C. W., & Whinston, A. B. (1996). Exploring the next generation of decision support. *Decision Support Systems*, *14*, 185– 186.

Holsapple, C. W., & Whinston, A. B. (2000). *Decision Support Systems: A Knowledge-based Approach*. West Publishing Company, ME.

Holtgraves, T., & Lasky, B. (1999). Linguistic Power and Persuasion. *Journal of Language and Social Psychology*, *18*, 196-205.

Huber, G. P. (1984). Issues in the Design of Group Decision Support Systems. *MIS Quarterly*, *8*, 195-204.

Jacobsen M, Clifford P, & Friesen S (2002) Preparing teachers for technology integration: creating a culture of inquiry in the context of use. Contemporary issues in technology and teacher education, 2(3). Retrieved May 12, 2011 from http://www.citejournal.org/vol2/iss3/currentpractice/article2.cfm

Janis, I. L., & L. Mann. (1977). *Decision Making: A Psychological Analysis of Conflict, Choice, and Commitment*. New York: The Free Press.

Jarvenpaa, S. L., Rao, V. S., & Huber, G. P. (1988). Computer support for meetings of groups working on structured problems: A field experiment. *MIS Quarterly*, *12*, 645–666.

Jarvinen, P.H. (2004): *On Research Methods*. University of Tampere.

Jessup, L. M., Connolly, T., & Galegher, J. (1990). The Effects of Anonymity on GDSS Group Process with an Idea-Generating Task. *MIS Quarterly*, *14*, 313 – 321.

Jessup, L. M., & Tansik, D. A. (1991). Decision making in an automated environment: the effects of anonymity and proximity with a group decision support system. *Decision Sciences*, *22*, 266-279.

Jessup, L.M., & Valacich, J.S. (1993). *Macmillan*, New York, NY.

Jiang, W. (2000). The relationship between culture and language, *ELT Journal, 54*, 328 – 334.

Jick, T.D. (1979). Mixing qualitative and quantitative methods: Triangulation in action. *Administrative Science Quarterly*, *24*, 602-611.

Johansen, R. (1988*). Groupware: Computer support for business teams*. New York: Free Press.

Johansen, R., Martin, A., Mittman, R., Saffo, P. I., Gibbet, D., & Benson, S. (1991). *Leading Business Teams.* Addison-Wesley, Reading, MA.

Johnson, P. (1992). *Human-Computer Interaction*, Mcgraw-Hill Book Company, London.

Kaplan, A. (1964). *The conduct of Inquiry*. New York: Crowell.

Kaplan, B., & Duchon, D. (1988). Combining Qualitative and Quantitative Methods in Information Systems Research: A Case Study. *MIS Quarterly*, *12*, 571-586.

Keen, P. G. W., & Morton, M. S. (1978). *Decision Support Systems: An Organizational Perspective*. MA: Addison-Wesley.

Keengwe, J., Onchwari, G., & Wachira, P. (2008). Computer technology integration and student learning: barriers and promise. *Journal of Science Education and Technology, 17,* 560–565.

Kelly, A. E. (2003). Research as design. Theme issue: The role of design in educational research. *Educational Researcher, 32*, 3-4.

Kelley, H., & Thibaut, J. (1969). Group problem solving. In G. Lindzey & E. Aronson (Eds.). *The handbook of social psychology:* Vol. 4 (2nd ed.). Reading, MA: Addison-Wesley.

Kersten, G. (1997). Support for group decisions and negotiations. An overview. In J. Clímaco (Ed). *Multicriteria Analysis* (pp. 332-346). Springer-Verlag.

Kim, U., Triandis, H.C., Kagitcibasi, C. Choi, S.C., & Yoon, G. (1994). *Individualism and Collectivism: Theory, Methods and Applications*. Thousand Oaks, CA.: Sage Publications

Kirkpatrick, D. (1993). *Here comes the payoff from PCs*. Decision support systems (3rd ed.). Prentice-Hall, Inc.

Klein, H. K., & Myers, M. D. (1999). A Set of Principles for Conducting and Evaluating Interpretive Field Studies in Information Systems. *MIS Quarterly*, *23*, 76-94.

Klein, E. E., Tellefsen, T., & Herskovitz, P. J. (2007). The Use of Group Support Systems in Focus Groups: Information Technology Meets Qualitative Research. *Computers in Human Behavior*, *23*, 2113–2132.

Kluckholn, C. (1962). *Culture and behaviour*. New York: The Free Press of Glencoe.

Kock, N.F. (2007). *Encyclopedia of E-Collaboration*. Idea Group Inc (IGI) Global, PA.

Komin, S. (1991). *Psychology of the Thai People*, National Institute of Development Administration (NIDA), Bangkok, Thailand.

Korpela, M., Soriyan, H. A., Olufokunbi, K. C. & Mursu, A. (1998). Blueprint of an African Systems Development Methodology: An Action Research Project in the Health Sector. In: C. Avgerou, C. (Ed.*). Proceedings of the Fifth International Working Conference of EP WG 9.4 Implementation and Evaluation of Information Systems in Developing Countries* (pp. 273–285). London School of Economics and Political Science and Asian Institute of Technology Bangkok, IFIP.

Kroeber, A. L., & Kluckhohn, C. (1952). *Culture: A critical review of concepts and definitions*. Harvard University Peabody Museum of American Archeology and Ethnology Papers

Kwok, R. C., & Khalifa, M. (1998). Effect of GSS on Meaningful Learning, hicss, Thirty-First Annual Hawaii International Conference on System Sciences, 1, 463.

Kull, D. (1982). Group decisions: Can a computer help?. *Computer Decisions*, *14*, 70-84.

Landry, J.R. (1995). Culture and the formalization of organizational decision support systems. *IEEE, 4*, 913-916.

Lee, A. S. (1991). Integrating Positivist and Interpretive Approaches to Organizational Research. *Organization Science, 2*, 342-365.

Lee, A. S. (2000). Systems *Thinking, Design Science, and Paradigms: Heeding Three Lessons from the Past to Resolve Three Dilemmas in the Present to Direct a Trajectory for Future Research in the Information Systems Field, Keynote Address*, Eleventh International Conference on Information Management, Taiwan, available online at http://www.people.vcu.edu/~aslee/ICIM-keynote-2000.

Lee, A.S. (2001). Editorial. *MIS Quarterly, 25*, pp. iii-vii.

Leidner, D. E., & Jarvenpaa, S. L. (1993). The Information Age Confronts Education. *Information Systems Research, 4*, 24-54.

Leidner, D. E., & Jarvenpaa, S. L. (1995). The Use of Information Technology to Enhance Management School Education: A Theoretical View. *MIS Quarterly, 19*, 265-291.

Lewicki, R., Weiss, S., & Lewin, D. (1992). Models of conflict, negotiation and third party interventions: A review and synthesis. *Journal of Organizational Behavior, 13*, pp. 209-252.

Lewis, A. C., Sadosky, T. L., & Connolly, T. (1975). The effectiveness of group brainstorming in engineering problem solving. *IEEE Transactions on Engineering Management, 22*, 119-124.

Lewis, L.F. (1994). *A Brief Introduction to Group Support Systems*. Bellingham, W.A, MeetingWorks Associates.

Linton, M. A., & Calder, P. R. (1987). The Design and Implementation of InterViews, in *Proceedings of the USENIX C++ Workshop*.

Livari, J. (2003). The IS Core – VII: Towards Information Systems as a Science of Meta-Artifacts. *Communications of the Association for Information Systems 12*, 568-581.

Mandviwalla, M., & Gray, P. (1998). Is IS Research on GSS Relevant? *Information Resources Management Journal, 11*, 29-37.

Mandviwalla, M., & Olfman, L. (1994). What do groups need? A proposed set of generic groupware requirements. *ACM Transactions on Computer-Human Interaction, 1*, 245-268.

Manheim, M. L., Vlahos, N. J., & Xie, Y. (1995). Designing team support applications to meet business objectives, In Coleman, D., and Khanna, R. (Eds.). *Groupware - Technology and Applications* (pp. 323 – 377). Prentice Hall, Upper Saddle River, NY.

March, S.T., & Smith, G.F. (1995). Design and Natural Science Research on Information Technology. *Decision Support Systems, 15*, 251-266.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design Theory for Systems that Support Emergent Knowledge Processes. *MIS Quarterly, 26*, 179-212.

McCarthy, J. (1980). Circumscription - A Form of Non-Monotonic Reasoning. *Artificial Intelligence, 13*, 27-39.

McDermott, R., & O'Dell, C. (2001). Overcoming cultural barriers to sharing knowledge. *Journal of Knowledge Management, 5*, 76 – 85.

McGill, A. (1995). American and Thai Managers' Explanations for Poor Company Performance: Role of Perspective and Culture in Causal Selection. *Organizational Behavior and Human Decision Processes*, *61*, 16-27.

McGoff, C. J., & Ambrose, L. (1991). Empirical Information from the Field: A Practitioner's View of Using GDSS in Business. *In the Proceedings of the 24th Annual Hawaii International Conference on System Sciences* (Vol. 3). Los Alamitos, CA: Society Press.

McGrath, J. E. (1984). *Groups: Interaction and Performance*. Englewood Cliffs, NJ: Prentice-Hall.

McGrath, J. E., O'Connor, K. M., & Hollingshead, A. B. (1993), Group task performance and communication technology: a longitudinal study of computer-mediated versus face-to-face groups. *Small Group Research, 24*, 307-333.

Mejias, R. J., Shepherd, M. M., Vogel, D.R., & Lazaneo, L. (1997). Consensus and satisfaction levels: A cross-cultural comparison of GSS and non-GSS outcomes within and between the United State and Mexico. *Journal of Management Information Systems*, *13*, 137-161.

Meyers, R. A., Seibold, D. R., & Brashers, D. E. (1991). Argument in initial group decision-making discussion: A refinement of a coding scheme and a descriptive quantitative analysis. *Western Journal of Speech Communication*, *55*, 47-68.

Mingers, J. (2001). Combining IS Research Methods: Towards a Pluralist Methodology. *Information Systems Research, 12*, 240-259.

Mintzberg, H. (1979). *The Structuring of Organizations*., Englewood Cliffs, NJ.: Prentice Hall.

Moran, J. W., Talbot, R. P., & Benson, R. M. (1990). *A guide to graphical problem-solving*. Milwaukee, WI: ASQC Quality Press.

Morgan, D. L. (1997). *Focus Groups as Qualitative Research* (2nd Ed,). London: Sage Publications.

Morison, J. (2004). E-democracy: on-line civic space and the renewal of democracy?. *Canadian Journal of Law and Jurisprudence*, *XVII*, 129–142.

Myers, M. D. (1997). Interpretive Research in Information Systems. *Information Systems: An emerging discipline?* J. Mingers and F. A. Stowell, McGraw-Hill.

Neuman, W. L. (2006). *Research Methods: Qualitative and Quantitative Approaches*. United States.

Niederman, F. & Bryson, J. (1998). Influence of Computer-Based Meeting Support on Process and Outcomes for a Divisional Coordinating Group. *Group Decision and Negotiation, 7*, 293-325.

Nielsen, J. (1993). *Usability Engineering*, San Diego, CA: Academic Press.

Nolan, R. L. & Croson, D. C. (1995). *Creative Destruction*. Boston, MA.: Harvard Business School Press.

Nunamaker, J. F., Briggs, R. O., & Mittleman, D. D. (1995). Electronic meeting systems: ten years of lessons learned. *Journal of Management Information Systems, 13*, 163-207

Nunamaker, J. F., Briggs, R. O, Mittleman, D. D., Vogel, D., & Balthazard, P. (1997). Lessons from a dozen years of group support systems research: A discussion of lab and field findings. *Journal of Management Information Systems*, *13*, 163-207.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S., & Vogel, D. R. and George, J. (1991a), Electronic meeting systems to support group work. *Communications of*

the ACM, *7*, 40-61.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S. & Vogel, D. R. (1991b). Information technology for negotiating groups: generating options for mutual gain. *Management Science*, *37*, 1325-1346.

Nunamaker, J. F., & Deokar, A.V. (2008), *GDSS Parameters and Benefits*, Handbook On Decision Support System 1. Springer Berin Heidelberg, 391-414.

Ocker, R., Hiltz, S. R., Turoff, M., & Fjermestad, J. (1996). The effects of distributed group support and process structuring on software requirements development teams: Results on creativity and quality. *Journal of Management Information Systems*, *12*, 127-153.

Orlikowski, W. J., & Baroudi, J. J. (1991). Studying Information Technology in Organizations: Research approaches and assumptions. *Information Systems Research*, *2*, 1–28.

Orlikowski, W. J., & Iacono, C. S. (2001). Research Commentary: Desperately Seeking the .IT. In IT Research.A Call to Theorizing the IT Artifact. *Information Systems*, *12*, 121-134.

Osborn, A. F. (1957). *Applied Imagination* (rev. ed.). New York: Scribner's.

Ozden, M. (2007). Problems with science and technology education in Turkey. *Eurasia Journal of Mathematics, Science & Technology Education, 3*, 157-161.

Palvia, S., & Hunter, G. (1996). Information Systems Development: A Conceptual Model and a Comparison of Methods Used in Singapore, USA, and Europe. *Journal of Global Information Management, 4*, 5–16.

Panko, R. R. (1994). Managerial communication patterns. *Journal of Organizational Computing*, *2*, 95-122.

Parent, M., Gallupe, R.B., Salisbury, W.D., & Handelman, J.M. (2000). Knowledge Creation in Focus Groups: Can Group Technologies Help? *Information and Management*, *38*, 47–58

Peirce, C.S. (1931-35). *Collected Papers*, ed. C. Hartshorene, and P. Weiss. Harvard University Press, Cambridge, MA.

Pinsonneault, A., & K.L. Kraemer. (1989). The Impact of Technological Support on Groups: An Assessment of the Empirical Research. *Decision Support Systems, Special Issue on Group Decision Support Systems*, 197-216.

Platt, D. S. (2001). Introduction Microsoft .NET. MS Press.

Pollard, C., & Hayne, S. (1996). The Reality of Meeting and the Use of Automated Meeting Tools. Americas Information Systems Conference.

Poole, M. S., Seibold D. R., & McPhee, R. D. (1985). Group Decision-Making as a Structurational Process. *Quart. J. Speech*, *71*, 74-102.

Power, D. J. (2004). Decision Support Systems: From the Past to the Future. *Proceeding of the Americas Conference on Information Systems*. New York, NY, 2025-2031.

Prpic, J. K. & Kanjananapanyakom, R. (2004). The impact of cultural values and norms on higher education in Thailand. *Proceedings HERDSA Conference, Retrieved 26 May 2008 from* http://herdsa2004.curtin.edu.my/Contributions/NRPapers/P064-jt.pdf.

Probert, S. K. (2001). Contemporary Epistemology and IS Methodology: An Interpretive Framework, *7th Americas Conference on Information Systems (AMCIS 2001)*, Boston, MA.

Purao, S. (2002). Design Research in the Technology of Information Systems: Truth or Dare. *GSU Department of CIS Working Paper*. Atlanta.

Quaddus, M. A., & Tung, L. L. (2002). Explaining cultural differences in decision conferencing. *Communiction of the ACM, 45*, 93-98.

Quinn, R. E., Rohrbaugh, J., & McGrath M. R. (1985). Automated Decision Conferencing: How it Works. *Personnel, 62*, 49-55.

Reeves, T. C. (2006). Design research from a technology perspective. In J. van den Akker, K. Gravemeijer, S. McKenney & N. Nieveen (Eds.). *Educational Design Research* (pp. 52-66). London: Routledge.

Reeves, T. C., Herrington, J., & Oliver, R. (2005). Design research: A socially responsible approach to instructional technology research in higher education *Journal of Computing in Higher Education, 16*, 96-115.

Rice, R. (1980). The impacts of computer-mediated organizational and interpersonal communication. *Annual Review of Information Science and Technology, 15*, 221-249

Rigopoulos, G., Psarras, J., & Askounis, D. T. (2008). Web Support System for Group Collaborative Decisions. *Journal of Applied Sciences, 8*, 407-419.

Robert B. & Economy, P. (1995). *Better Business Meetings*. Burr Ridge, IL: Irwin Inc.

Rohitratana, K. (1998). The role of Thai values in managing information systems: a case study of implementing an MRP systems. In Proceedings of the Fifth International Working Conference of IFIPWG9.4 – Implementation and Evaluation of Information Systems in developing Countries (Avgerou, C. Ed), 188-201. London School of Economics and Political Science and Asian Institute of Technology Bangkok.

Romano, N.C., & Nunamaker, J. F. (2001). Meeting Analysis: Findings from Research and Practice. *Proceedings of the Thirty-Fourth Hawaii International Conference on System Sciences*. IEEE Computer Press.

Rossi, M., & Sein, M. K. (2003). Design research workshop: a proactive research approach, in: *26th Information Systems Research Seminar in Scandinavia*. Haikko Finland: The IRIS Association.

Salomon, G., & Globerson, T. (1989). When teams do not function the way they ought to International. *Journal of Educational Research, 13*, 89–98.

Sawir, E. (2005). Language difficulties of international students in Australia The effects of prior learning experience. *International Education Journal, 6*, 567-580.

Schmuck, R. A., & Schmuck, P.A. (1988). *Group Processes in the Classroom* (5th Edition). William C. Brown, Dubuque, IA.

Schoepp, K. (2005). Barriers to technology integration in a technology rich environment: Learning and teaching in higher education. Retrieved August 12, 2011 from http://www.zu.ac.ae/lthe/vol2no1/lthe02_05.html.

Schon, D. A. (1983). *The Reflective Practitioner*. New York, NY.: Basic Books Inc.

Schon, D. A. (1993). *The Reflective Practitioner: How Professionals Think in Action*, , New York: Basic Books.

Schwarz, R. (1994). *The skilled facilitator*. Jossey-Bass Publishers.

Schulz, M. (2001). The uncertain relevance of newness: Organizational learning and knowledge flows. *Academy of Management Journal, 44*, 661-681.

Scott Morton, M. S. (1971). *Management Decision System: Computer-based support for decision making*. Boston, MA: Harvard University.

Shadish, W. R., Cook, T. D., & Campbell, D. T. (2003) *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Boston, MA.: Houghton Mifflin Company.

Shanks, G., Rouse, A., & Arnott, D. (1993). A Review of approaches to research and scholarship in information systems. *Working Paper Series*. Caulfield East, Australia: Monash University Printing Services.

Shaw, M. E. (1976). *Group Dynamics: The Psychology of Small Group Behavior*. New York: McGraw-Hill.

Shim, J. P., Warkentin, M., Courtney, J.F., Power, D.J., Sharda. R., & Carlsson, C. (2002). Past, present, and future of decision support technology. *Decision Support System*, *33*, 111-126.

Shneiderman, B. (1998). *Designing the user interface: Strategies for effective human-computer interaction* (3rd Ed.), Reading, MA: Addison-Wesley Publishing.

Siegel, J., Dubrovsky, V., Kiesler, S., & McGuire, T. W. (1986). Group Processes in Computer-Mediated Communication. *Organizational Behavior and Human Decision Processes*, *37*, 157-187.

Simon, H. A. (1981). *Sciences of the Artificial* (2nd Ed). Cambridge, MA: MIT Press.

Simon, H. A. (1996). The Sciences of the Artificial, (3rd Ed). Cambridge, MA.: MIT Press.

Simon, H.A. (1960). *The New Science of Management Decision*. Harper Brothers, New York.

Simon, H. A. (1997). *Administrative behavior: a study of decision-making processes in administrative organizations* (4th Ed.). Simon & Schuster Inc.

Simon, H.A. (1988). *The Sciences of the Artificial* (2nd Ed.). Cambridge, MA.: MIT Press.

Sondergaard, M. (1994). Hofstede's consequences: A study of reviews, citations, and replications. *Organizational Studies*, *25*, 447-456.

Spradley, J. (1980). *Participant Observation*. New York: Holt, Rinehart, and Winston.

Sprague, R, H., Jr. (1980), A Framework for the Development of Decision Support Systems. *MIS Quarterly*, *4*, 1-26.

Sprague, R. H., Jr. & Watson H. J. (1979). *Bit by Bit: Toward Decision Support Systems*. California Management Review, XXII, 1, Fall, 60-68.

Sprague, R. H., Jr. & Watson H. J. (editors). (1996). *Decision Support for Management*. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Steiner, I .D. (1972). *Group Process and Productivity*. New York: Academic Press.

Stewart, D. W., & Shamdasani, P. N. (1990). *Focus groups: Theory and practice*, Newbury Park: Sage Publications.

Stone, E. F. (1978). *Research Methods in Organizational Behaviour*. Santa Monica: Goodyear Publishing Co.

Tan, B. C., Wei, K., & Watson, R. T. (1999). The Equalizing Impact of a Group Support System on Status Differentials. *ACM Transactions on Information Systems*, *17*, 77-100.

Tan, B. C., Wai, K., Watson, R. T., Clapper, D. L. ,& McLean, E. R. (1998). Computer-Mediated Communication and Majority Influence: Assessing the

Impact in an Individualistic and Collectivistic Culture. *Management Science*, *44*, 1263 – 1278.

Takeda, H., Veerkamp, P., Tomiyama, T., & Yoshikawam, H. (1990). Modeling Design Processes, in: *AI Magazine*, 37-48.

Teerakul, S. (2001). The study of Thai graduate students' opinion toward educational environment in Taksin University.

Thanasankit, T., (2002). Requirements engineering-exploring the influence of power and Thai values. *European Journal of Information Systems*, *11*, 128-141.

Trauth, E. M., & Jessup, L. M. (2000). Understanding information technology usage: A test of competing models. *Information System Research*, *6*, 144-176.

Trompenaars, F. *Riding the Waves of Culture*. New York: McGraw-Hill.

Tse, A. C. B., (1999). Conducting electronic focus group discussions among Chinese respondents. *Journal of the Market Research Society*, *41*, 407–415.

Tse, E., & Greenberg, S. (2002). SDGToolkit: A Toolkit for Rapidly Prototyping Single Display Groupware. Poster in ACM CSCW '2002 Conference on Computer Supported Cooperative Work.

Turban, E., & Aronson, J. E. (1998). *Decision Support Systems and Intelligent Systems*. London: Prentice-Hall International Inc.

Turoff, M. & Hiltz, S. R. (1982). Computer Support for Group versus Individual Decisions. *IEEE Transactions on Communications*, *30*, 82-90.

Tsichritzis, D. (1997). The Dynamics of Innovation. *Beyond Calculation: The Next Fifty Years of Computing*, Copernicus, 259-265.

Tyran, C. K. (1997). GSS to Support Classroom Discussion: Opportunities and Pitfalls, Proceedings of the 30th Annual Hawaii International Conference on Systems Sciences. *IEEE, 2,* 260-629.

Tyran, C. & Shepherd, M. (1998). GSS and Learning Research: A Review and Assessment of the Early Studies. *IEEE, 1*, 470-179.

Vaishnavi, V. and Kuechler, W. (2004/5). Design Research in Information Systems January 20, 2004, last updated August 16, 2009. URL: http://desrist.org/design-research-in-information-systems.

Valacich, J. S., Dennis, A. R., & Nunamaker, J. F. (1991). Electronic support for meetings: The group systems concept. *International Journal of Man Machine Studies*, *34*, 261-282.

Valacich, J. S., Jessup, L., Dennis, A. R., & Nunamaker, J. F. (1992). A conceptual framework of anonymity in group support. *Group Decision and Negotiation*, *1*, 219-241.

Van den Akker, J. (1999). Principles and methods of development research. In J. Van den Akker, R. M. Branch, K. Gustafson, N. Nieveen, & T. Plomp (Eds.). *Design approaches and tools in education and training* (pp. 1-14). Dordrecht: Kluwer Academic Publishers.

Vathanophas, V. & Liang, S.Y. (2007). Enhancing information sharing in group support systems (GSS). *Computers in Human Behavior*, *23*, 1675-1691.

Venable, J. (2006). The role of theory and theorising in design science research. In: Proceedings of the 1st International Conference on Design Science in Information Systems and Technology, Claremont, USA, 1-18.

Vreede, G. J. DE, Jones, N., & Mgaya, R.J.S. (1998). Exploring the Application and Acceptance of Group Support Systems in Africa. *Journal of MIS*, *15*, 197-234.

Vreede, G. J. DE, Mgaya, R., Schuuring, J., Cornelissen, B., & Feenstra, R. A. (2000). Comparison of the Impact of Electronic Meeting Technologies in Two Less Developed Countries. *Proceedings of IFIP9.4*, 24-26.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an Information System Design Theory for Vigilant EIS. *Information Systems Research, 3*, 36-59.

Walsh, K. ., Briggs, R. O., Ayoub, J., Vanderboom, C., & Glynn, M. S. (1996). Learning with GSS: A Case Study. *Proceedings of the 29th Annual HICSS.* IEEE Press, Los Alamitos, California, 283-292.

Walsham, G. (1993). *Interpreting Information Systems in Organizations*. Wiley: Chichester.

Walsham, G. (1995). Interpretive Case Studies in IS Research: Nature and Method. *European Journal of Information systems*, *4*, 74-81.

Wand, Y., & Weber, R. (2002). Research Commentary: Information System and Conceptual Modeling – A Research Agenda. *Information Systems Research*, *13*, 363-376.

Watson, R. T., Ho, T. H., & Ramann, K. S. (1994). Culture: A forth dimension of group support systems. *Communications of the ACM*, *37*, 44-55.

Weber, R. (1987). Toward a Theory of Artifacts: A Paradigmatic Base for Information Systems Research. *Journal of Information Systems, 1*, 3-19.

Weber, R. (2004). The Rhetoric of Positivism Versus Interpretivism. *MIS Quarterly*, *28*, iii-xii.

Wendy E. M. (1998). Triangulation within and across HCI disciplines. *Human-Computer Interaction*, *13*, 310–315.

Wiedersheim-Paul, F. & Eriksson, L. (1991). *Att Utreda, Forska och Rapportera*. Liber-Hermods: Karlshamn.

Whitten, J., L. Bentley, & Dittman, K. (2004). *Systems Analysis and Design Methods*, (6th Ed.). Boston: McGraw-Hill Irwin.

Zigurs, I., & Buckland, B. K. (1998). A theory of task/technology fit and group support systems effectiveness. *MIS Quarterly*, *3*, 313-335.

Zigurs, I., Poole, M. S., & DeSanctis, G. L. (1988). A study of influence in computer-mediated group decision making. *MIS Quarterly*, *12*, 625–644.

Zweig, J. M. (1990). The Conduit: a Communication Abstraction in C++, in Proceedings of the 2nd USENIX C++ Conference (pp. 191–203). USENIX Association.

Appendix I

A Sample of BrainEx Generated Report

This is excerpted from actual test session

STEEPC - Present

Social

Age gap

We are thE WOrLd

First: Knowledge Management

put on line training

second: close age gap

third: set up key sectors: HR... IC

red Shirt

Red & Yellow fight

civil unrest

more companies work at home..

easy going

facebook twitter

on line training is be very effecrtive

fast life style

internet is wisely used inThailand

Thai edUCATION SYSTem and its critiCAL PROBLEMS

blems arise so HR will be mroe important cosidERation

ECOnomic crisis

MOre demand on people skill toward the multi taskiNg.

unemployment

LET's Go to SET INDEX

Thai like westeners.

Less spending pOWER

E-learning course always available on the internet, peoPLE somehow do not interent TO attend ANY courses IN class

the rising of illigal abandonment

Thai like Janpese and Korean fashion

Quality oF skill due to focus not on the knolwedge but rather more on getting the degree

Traffic in Bangkok ALWAYS COngested

TrafFic jam iN big CITY

trAFFIC jams

locALs go out counTry, OUTsidERS COME into thE couNtry

cuSTOMEr PREFER onsite training

heavy traffic in downtown

the rising of COST of living in bangKOK aND thailand

too many cars

scholL RIvalRIEs in the lateST news

many copanies haVE own internal training course by themselve and they someHOW Do not interest to get external company to provide the training course

new hrend oF HR MIghT sPin to oUt source system

hai are not that punctual

people drive very quickly

new fashion trend with some identity crisis

Thai are not that hard-working

less tourists in Thalland due to poloitical situatiOn of 20!0

Trend towarDS sustainable toUrism

given social unrest under negative circuMstances in other aspects such as P,E,T, the training subjects of conflict resolution and Human Right Issue will l play more important roles in the future

how Can NEW graduateS be accepted by companies??

exchange rate is low now

traffic very bad

Bangkok kids love to play with I-phones, and other technologies

too many new graduates are looking for jobs

kingdom of pirated DVD's and games

pressure on finding a job for new graduate

IS early retirement good or bad??

EaRlY retire is EsSential

the rising of illigal work man leak from boundary

tourists fear for safety

more people like to study chinese

Jum will buy this program to enhance thier training capability

multi-cultural life-style

The transportation doesn't have reliable

many natonality in thailand ANd therefore the training course or training coMPAny should have multi-language

customer, especially well educated ones, arE more critical towards new PRoducts

Is it good for thai companies to hire foreigners?

consumer ConfiDEnce decreases.

otheR PRovinces of Thailand rather neglected by the gOvernment ex. flood issues

employement rate??

crime commit by non-state people

MAny peolpe come frim Different coutry doing bUsnless here

un-employment rate, pople lose their jobs.

lack of training in International standards of service.

Technology: T

Problems of publi of bus, boat or train

Technology

stilL no sufficIENt 3G

Steve jobs aleady rules this planet

moblie tEchnologies

telecommunication system in general Really bad quality

T always chanGE quickly

lomg distance LERANING CAPABILITY

Rich kids RELY ON TECHNOLOGIEs Too mucH

i pad was sold YEsterdar 15000

Security becomes more sensitive

Today the technology is growing too fast ,the coMPANY Might lack of the newtechnology

WI-FI

technology is moving sO FAST

online meeting

traditional meeng will be gone

modern technologies APPLIED TOO QuICKLY, meaning NOT Much time to learn it in certain aREAS

expensive cost of new arrival tECHNOlogy for thai PEOPLPE

n addtion to traditional ways of training, explore every other MeANS OF COMMUNICAtioN, i.e CD DVD on line- compter, set up own TV CHANNEL mobiLE Phone..

TEChnologies must be applied to the where it makes seNSe

a lot OF online games

internet community is so popular in Thaiand, peoplE do not want to attend the class

IT devicees R Very cheap

a Lot of porn sites

high technology helps People get closer

online crimEs and viruses threaten the world

The government haven't yet verified the modern techNology liKE 3G

Online learning might become more popular.

THERE isn't any Thai technology company to support the country's development

people get more entertainments

increase in social networking through the Internet

worK with STratigic partner in telecom to strengthen THE CompeteNcy

online shopping

a lot of people surf the internet

Lack of performance officer in the ICT

WIRELESS technology are strongly important

unqualify techniCIAn

can online in many place

corruption in Thaiand

more opportunities for professional web designers to work as freElancer

virus

internet encourages self study

forergn technologies dominate Thailand

VIRUS!11

get some supportfinancially through ADs

moRE PREFERENces for self study instead of classroom only exercises

powerpoint is the trend !!!!!

high tech. make things work faster and smoOTHer

adavance tECHNOloGY maKE people forget the basicof liviNg

where is Thai technology R&D center

censorship of internet by the evil evil ICT peeps

limItation Of interenet in Thailand, interent cost is still expensive comparing with other countires

censorship of the evil evil ICT peeps

insufficient infrastructure is an obstacLe FOR ADVAnced technology importeD to Thailand

new anD avance technology is set up only in the big city

seLF-paced study is wisely used in thailand

when the systemS IN technology goes wrong there is no crisis manageMent measurements

Does Thai government support universities to develop new technologies by their own? Does the government have special budget to tose universities?

lack of huMAN vaLues and more Relience on too much techsnologies

All things about database matters should be in the Smart Card e.g. driver licence, SOcail assurance, ATM based on government bank

it has both good way and worse way

Environment

The global warming

bangkok's famous traffic jams contribute a lot to bad environment

flood at ayuttaya

ธรรมชาติเลื่อมโทรม

air, water, soil, pollution

getting hoter and hoter

traffic congested in bangkok and some other big cities in Thailand

พื้นที่ป่าลดลง

noise pollution via construction

toxic keeping place

provinces outside bangkok are endangered by floods and drought

dirty water

หน้าร้อนแล้ง หน้าฝนน้ำท่วม

too many A/C generate much more heat than they can actually cooling down the rooms or what so ever

a leckage of people's preception of environmental protection

ความแห้งแล้ง

PEOPLE get sick because of pollution

beautiful View

limitation of transportation

เชื้อโรคเต็มไปหมดในอากาศ

air conditions are too cold And absorb a lot of energy

thai buses exhaust a Lot of gas fumes

not much effort put into developing the public transportation

สายไฟฟ้าโยงพันกัน

tab water should consumable properly

กรมควบคุมมลพิษ มีฤทธิ์ได้ทุกอุตสาหกรรม

รถติดฉากกกกก

รถเยอะทั้งในเมืองนอกเมือง

too much construction cause MORe traffic jams

nuclear poWER for energy

are there any government divisions actually protect the Environment???

ถนนมาแทนที่แม่น้ำ

less rain but hoter

ทุกภาคอากาศไม่เหมือนกัน เหมือนคนละประเทศ

pollution affects the street food's quality

Less naturaral resoures such as oil iron and woods.

too many plastic bags have been used and distributed by supermarkets, 7-11, and others

vernment have clearly neglected environmental disaster

sustainable strategies havebeen considered in people's mind

limit emission from the factury outside the big city

เต็มไปด้วยนิคมอุตสาหกรรม

pollution of the sea due too release of chemical garbage by factorieS

no plan to deal with waste water

no plan about environmental issues

น้ำเน่า อากาศเน่า คนเน่า ทุกอย่างเน่า

environmental issues make people and employees sick

กองขยะสูงเป็นภูเขา

ชายหาดมีแต่ขยะ

recycling system

cabon rate policy

many people in Thailand still do not understand about the global warming and how to help country or world

limit high power air-conditioner prodeced

Economic

unempolyment rate in thailand are still high

a lot oF graduates but high percentage of unemployment

GDP

decrease IN INTernational business establishment

LESS TOURISTS COMinG TO THAILAND

high cost of LIVING IN bANGKOK cOMPARE WITH THE CURRENT INCOME

Effect from powerful countries such as USA , CHAiNA aND JAPAN.

trend goES TOWardS Sustainable tourism or cheaper accomodation

imbalance iN rICH And poor, ... the gap goes higher and higher

Thai BaHT is wEAk

AGING socieTy

thai baht unsaTAble

coRRUption !!!!

dollar is cheap

some areas are non balance of living cost even in the same city

euro is unstable and cheap, too

Very large number of Buddhist temple FUnd

low spending power

Technology not affordable in most schools

IIMport product rate is higher than export products,

Stock market down turn

Unstable politics

unSATIfaction from new iNvestor due to politacal situation

HK$ r vERy cheap

high volume of classROOMS IN THai SCHools

teacHERS AND lECTUreRS not PAID very WEll

problems of PIGs + B, QE2 and KOreaN WAR will move funds to asia

employement rete infulences the economy

Lack of promotions of local product to the world market properly

unrealistic Expectations of Parents and authorities from their children

China and India will receive most benefits from the fund floW

social pressure OF young generation

SET rate is unstand, up & down always

bad Crisis management

corruption

PURe gold 99.99 vs 96.5

slum still exsited in city

over pay trend of thai prople

thai market will have more spending in iomproving their knowldge from various training courses

set 50

set100

rote learning at most thai schools instead of critical thinking

spending more / more demand on JUM

general public purchaSing power

um can raise fund to expand their businesses

increASED PRIces in almost every aspects

foreign investments keep increasing

the comparision between Thai GDP in 2008 and in 2009

to increase human capital as well as other partners: stratergic capital

credit card debt from underpay employees

in crease network to cover Thailand and GMS

education system should bring VALUES and social behavior first instead of academics...

waht are the most popular products in Thai market?

debt rate is very higH for many employess becausethey cAN get money so easily from credit card, Personal loan, etc and at the end, they care unable to return back to banK

tap venture caplTal fund/ list in mAI

illgal credit company

tourism business and inDUStry helps geneates amount of money

whoever is rich is lucky

incresing of Industrail estate place in thsilsnd for 10000 rai in eastern seaboard.

Politic

แบ่งเป็นก๊ก เป็นเหล่า

CoRruPTION

i LOVE samak sOndaravet

money is gOd

ต้องการเป็นใหญ่

Oe needs tO support more ed

I loke Thaksin

เสื่อมโทรม

i love taksin

incompetent politiciANs

sick of unstable in thailand

social hierarchy is thailand's downFall

Unsatble politics in Thailand

unstable sTatus

รัฐธรรมนูญเปลี่ยนไปตามรัฐบาล

i love democrasy

รัฐธรรมนูเพื่อรัฐไม่ใช่เพื่อประชาชน

Yellow & red always fight

เสื้อหลากสี

I love king

ซื้อเสียง โกงเลือกตั้ง

Political Group leaders forget about human rights

No Common on color shirts but they blows somehow

รัฐบาลไม่ทำตามแผนยุทธศาสตร์ของชาติ เมื่อเปลี่ยนรัฐบาลใหม่ก็ทำตามแผนของพรรคตัวเอง

politics is an issue in universities

รัฐประหารเป็นเรื่องปกติ

recognise the credential of the trainee who pass the courses

คณะรัฐบาลไม่มั่นคง

stateless people

undergraduated and stupid politicians

complicated visa procedures for foreigners

nepotism due to family ties AMONG POLITICIANS

ทหารแตงโม

Police officers are to get involve in corruption

dirty corruption in the government officers

นักการเมืองแบ่งที่นั่งในสภา

more party in thailand election

unclear visa regulations

government parties always change and making country has no direction

free trade policy

money can buy power and rights

not much involvement into politics by students

top to the bottom approach

Value

generation y has the most contact to western culture

ความไม่มีน้ำใจช่วยเหลือซึ่งกันและกัน

copy western countries

รับวัฒนธรรมตะวันตกมากเกิน

multi-cultural

korae trend

บ้า korae series

ชอบอะไรเป็นพักๆ แป๊ปๆ

buddhism country

easier to make friends online and exchange cultural values

เด็กไม่นอบน้อมต่อผู้ใหญ่

younger like to go out and stay with friends rather than family

should realize the benefits of the training courses

the internet becomes the ultimate connecter of people

หาคู่ผ่านโลก online

no clear self identification

spend a little time to know each other then get married

international

สนใจแต่วัตถุมากกว่าคุณค่าทางใจ

traditional cultural stuffs still maintained

parent prefer to send kids to study in international school

training the trainer to be more capable

Differentiation between adult and young generation.

thai people is very sensible and affect able easily to their mind

เกรงใจ

Actors and actress have high impact on teenagers.

old cultural behaviors with new trend

foreign cultures have strong effect to traditional Thai culture

ธรรมกาย จัด บวชพระ100000รูป

thai culture is known as Baeutiful culture

พระเกย์

diversify to monKhood area

more selfish people in Bangkok

พระไม่อยู่ในกรอบวินัย

independent from family

thai culture values กลัวเสียหน้า

การเช่าพระเครื่องปลอม

People still with different cultural background may leads to conflicts

discremination

คิดถึงแต่ตัวเอง

high context vs low context societies

ต้องจบโทถึงมีงานทำ

learning in schools as in the text book instead of critical thinking and understand the concept

value of thai's ancient remains and antiques

pub & bar every weekend

เลิกเหล้าเข้าพรรษา

fear of expressing oneself ideas

time is not an issue

hard to take things seriously

open mind

no outside of the box thinking

afraid to share the idea in the public

built up incentive for younger generation such as gold, silver, bronze, platinum medals when one takes series of training courses

certain

gender crisis is not a big issue

support the right for gay in some forbiden areas or matters

opportunities for higher education is only for the wealthy society

medals will support credential when applying for jobs

open minded nothing's unacceptable

not enough allocation of high education institutes accross the country,... but mostly in bangkoK

เณะแต่วเกลื่อน

many of universities in Thailand and some are just giving degree to people to easily.>>>some of the university do not Certified

un-distinct sex classification

พระเรียนปริญญาเอก

sex education is a taboo

nationalism ,non-colony state

new ideas and thoughts can be accepted easily

avoidance of critical problems

eating along the street is should be arranged in the right place

a buddhist country permit prostitution, but restrict gambling

prostitution business legally

operated

Appendix II

A Sample of Summarized Report from BrainEx Usage

The brainstorming process using time around 5-6 hours get many ideas from 13 students. At first I don't think the idea will come too much but after typing and see the other idea on the screen, the new idea come up all the time. That is very useful for us to develop the scenario I saw other people think about the situation of Thailand in next 15 years. And we helped to narrow down the situation which Thailand will be in the next 15 years.

The BrainEX tools is a wonderful tool, I felt not shy to typing the idea down because the other didn't know which one is mind and one more thing I could type in Thai language too.

Result

The idea form the brainstorming session came up with huge data, 43 pages long around 1,000 up idea.

Here is the summarized idea which each group was present;

Social

Categories	Keyword Present	Keyword Future
1. Family	High race of divorce, the decreasing of family time involving, the violence in the family,	Leave old people, family size is smaller, age extending, birthrate decrease, decreasing of member family, less family, less time with family
2. Life Style	Single parents, scruple, more old people, decrease birthrate, careless neighborhood, disorder, rush, stressed out, more single, late married, confident, sex in young age group, technology addicted, pregnant before married, low morality, kid start to work early, focus on money, look for convenient, Korean addicted in young age, selfish, living in condominium, influence by TV, Radio, Internet, jobless	Pre sex, less married, technology addicted, care on environment, people go to temple more than previous, independent, gay marriage, more gay, gay acceptance, better morality, people more obese, higher suicide rate, more care environment, do online business, more single
3. Education	level of education is higher, school is the main key to taking care young student	Higher education, multi skip people, more school/university in the country
4.Crime & Drug	Crime problem, drugs, street racing, rape, foreigner labor problem, violent of female	Drug increasing, high-low crime, more homeless
5. Social Gap - Age gap - Class gap - Adaption	Human right, social network, community of senior, the gap between rich-poor, the gap between young and old, conservative with the old group, poor people without chance, racism war in the south,	Conflict of people between young and old people, multi culture society, less farmer, decreasing moving rate into city, more community, unemployment, the disable body rate decrease, separate south, more new disease, middle class increasing, less human right, less labor,

Political

Key Word	What is happening?	Key Word	What will happen?
Political System	Constitution Monarchy	Political System	- Constitution Monarchy - President system - Military detector
Policy	Populism	Policy	
Political situations	- Different political idea - Corruption - Conflict within Government parties - Weak of King institution - Conflict with Cambodia - Problem in southern part of Thailand	Political situations	- Civil war - Peace full - Corruption - Less corruption
		Government Parties	- Prachatipat is the government - Taksin parties is the government
		Constitution	- New constitution - Same constitution

Economic

1. Thai Khem Kheang
2. Self Sufficiency Economic
3. Global Economic
4. Gold, Gas world price

Thai Khem Kheang

Support keywords:

- Government support old people
- Thai retires people can live better
- Thai brand developed to be strong brand in the world
- SME grow up
- Individual investment grow up
- Government support money for free education until graduate level
- Thailand GDP will increase
- Main income is from service industry
- TAT will be more successful
- Travel is the big export product

Self Sufficiency Economic

- Less import, produce in country
- The economic will be good in future if Thai people feel enough for life
- Agriculture will be more important

Global Economic

- Thailand will be hub of Asia
- Hamburger crisis
- Dubai crisis
- Agriculture industry will be stronger and empower to the others
- Thailand will be economic leader if we find petroleum
- China will be Thailand main export partner
- Less organization in stock market (only huge company can survive)
- 3 organizations (America, Europe, Asia) can help each other
- China will be the main factory in the world
- The company will invest in overseas asset

Gold, Gas world price

- Gold prices will be more expensive
- Nobody use gas anymore

Technology

1. Expansion of network and security system: E-business (E-commerce, E-shopping, E-marketing, E-entertainment, E-banking), E-education, E-library, E-book, Video-phone, Wide internet coverage area, 4G wireless, High data protection.

2. Hardware & software development: Chip mobile, Voice Command, Small miniature sensor, Multi-software.

3. Innovation: Nano Technology, Bio-technology, Robot, Green IT, Electronic vehicles (Cars, Trains)

4. Medical Technology: tailor-made medicine, better sensory equipment such as

Cultural

1. Everybody accepts in the third-sex.
2. Using 4 main languages
3. People don't want to married.
4. Children believe in western and pop culture.

The finalized scenario:

From the above summary and from Professor Charles, I have sum up which the scenario which will be in next 15 years in Thailand.

Social	- Unemployment rate in Thailand is 8%. - The older generation (over 60s) now make up 22% of Thailand's population. - Thailand still has problems with drug usage and HIV/AIDS, although there have been some medical breakthroughs. - People care more about their health, more healthy food. - The increase of birthrate not more than 2 per family. - Unmarried rate is increase. - Higher education, more skill, English and Chinese is the 2nd and 3rd language
Economic	- Thailand will be the important member group of ASEAN. - Bangkok will lose the tourists to Myanmar, Cambodia and Vietnam as it is cheaper. - Thailand will be the place for healthy tourist. SPA, Thai-Medical. - Thailand will be the kitchen of the world. Provide the better quality of food to other countries. - The financial plan will be more popular. Because people caring more about the retirement time. - The income of people will increase. - The tax rate will increase. - The retirement plan of the government will be better.
Political	- There are three major political parties and five minor parties represented in the Thai parliament.
Environment	- The river more clean cause the government run the campaign to motivate the people. - More tree and wood in Thailand. - Thailand will use the natural power instead gas and oil.
Cultural	- Thai cultural will fade out, the foreigner cultural replace. - Less warm hospitality - The young people start to work earlier for taking care themselves. - The Thai social will be more western, small family, young people will leave on their own - Third sex is more acceptance and can getting married like western country

Appendix III

BrainEx Application Flow Diagram

BrainEx includes 6 main functions: User Setup, Template Setup, New Session, Continue Session, Complete Session and MS Word Window

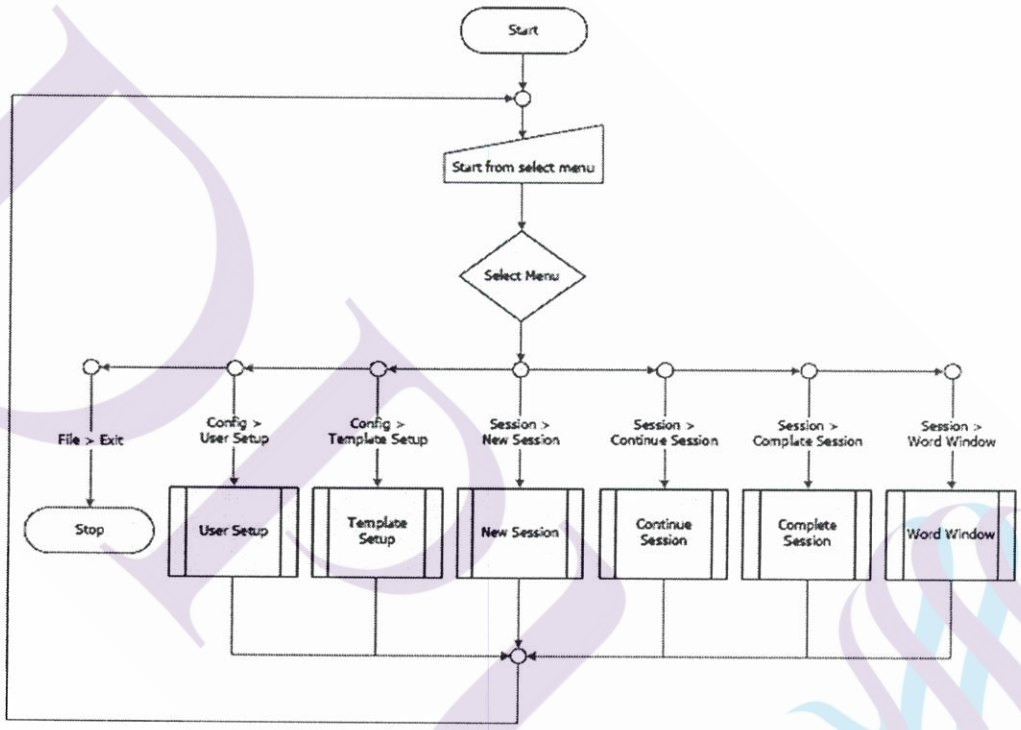


Figure Appendix 1.1: Overall BrainEx Functional Diagram

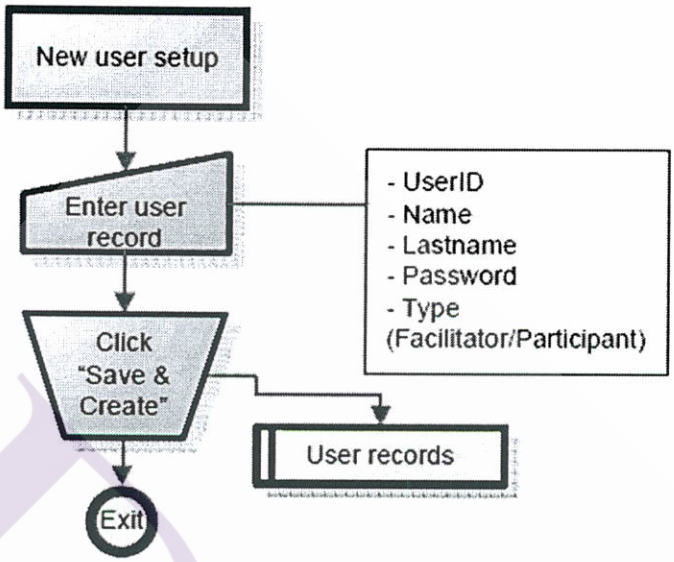


Figure Appendix 1.2: User setup

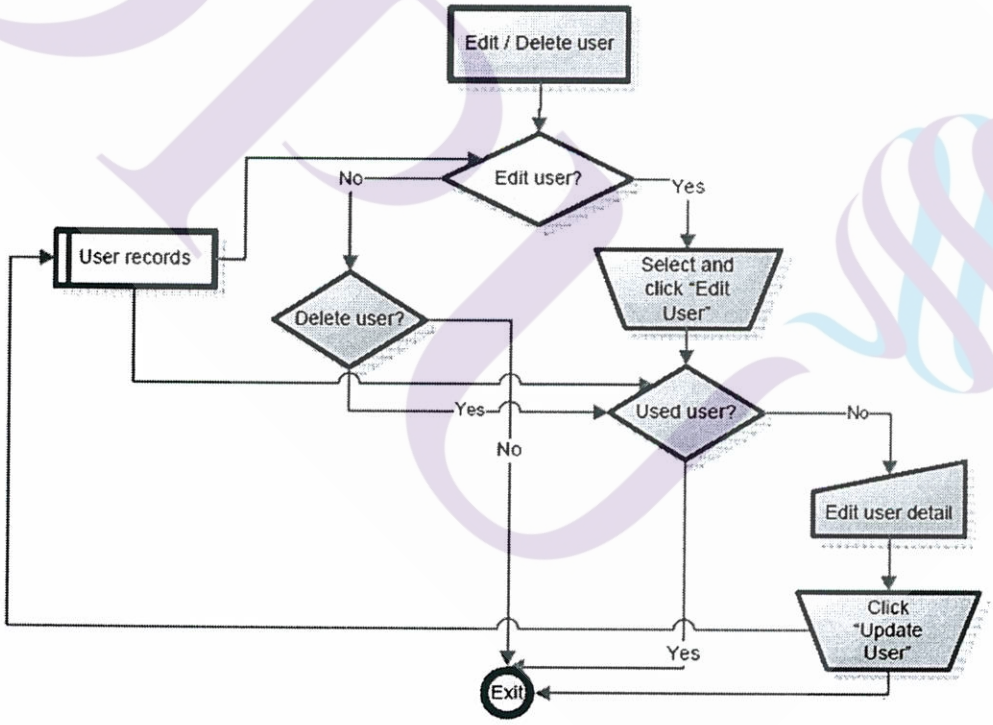


Figure Appendix 1.3: Edit / Delete user

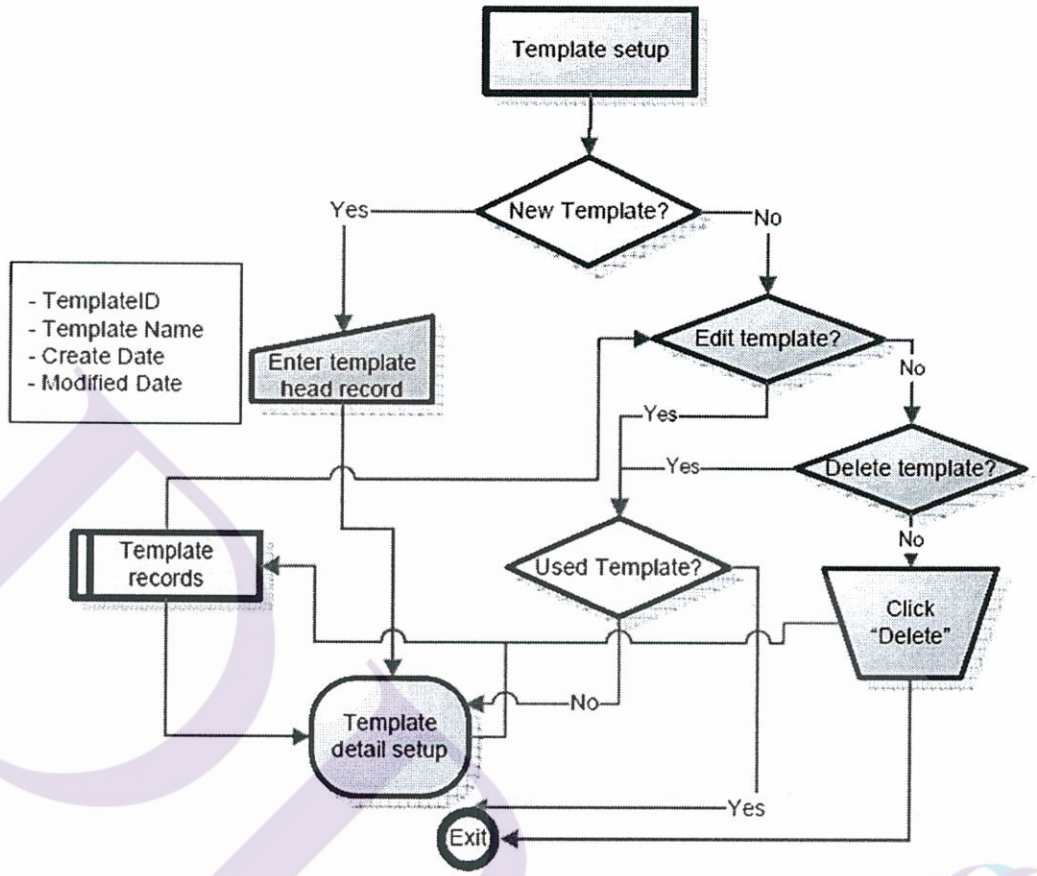


Figure Appendix 1.4: Template Setup

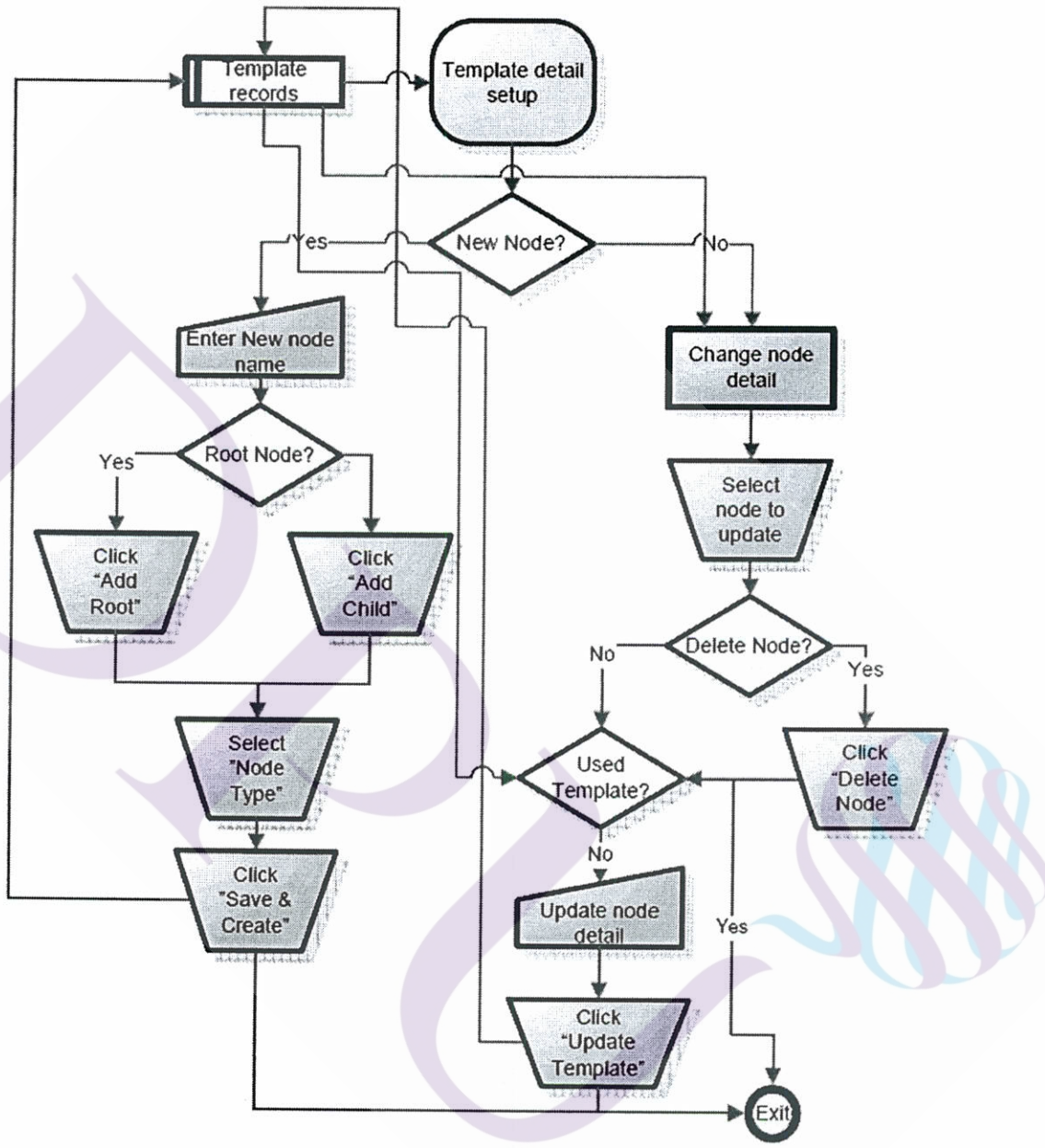


Figure Appendix 1.5 Template detail setup

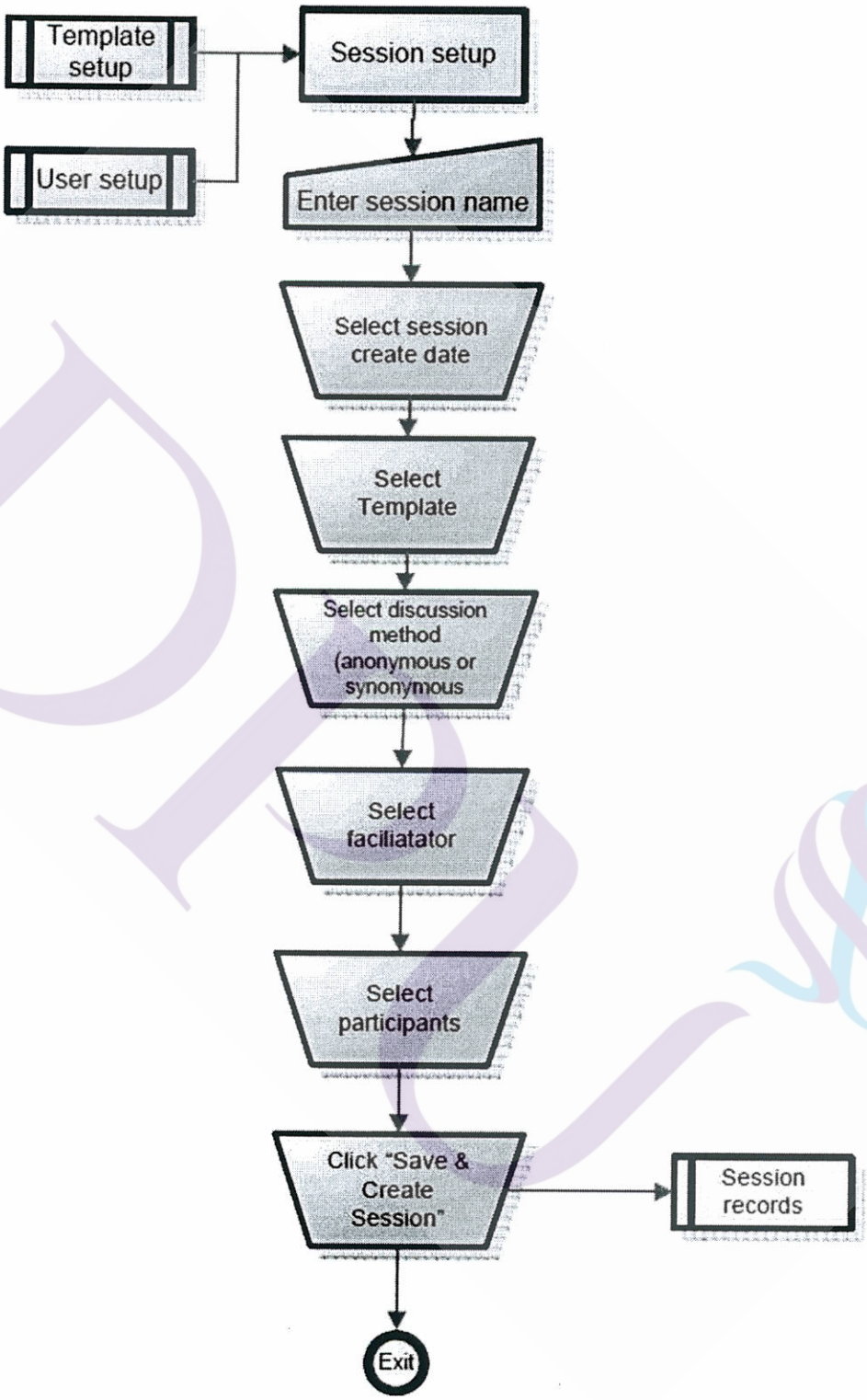


Figure Appendix 1.6: Create new session

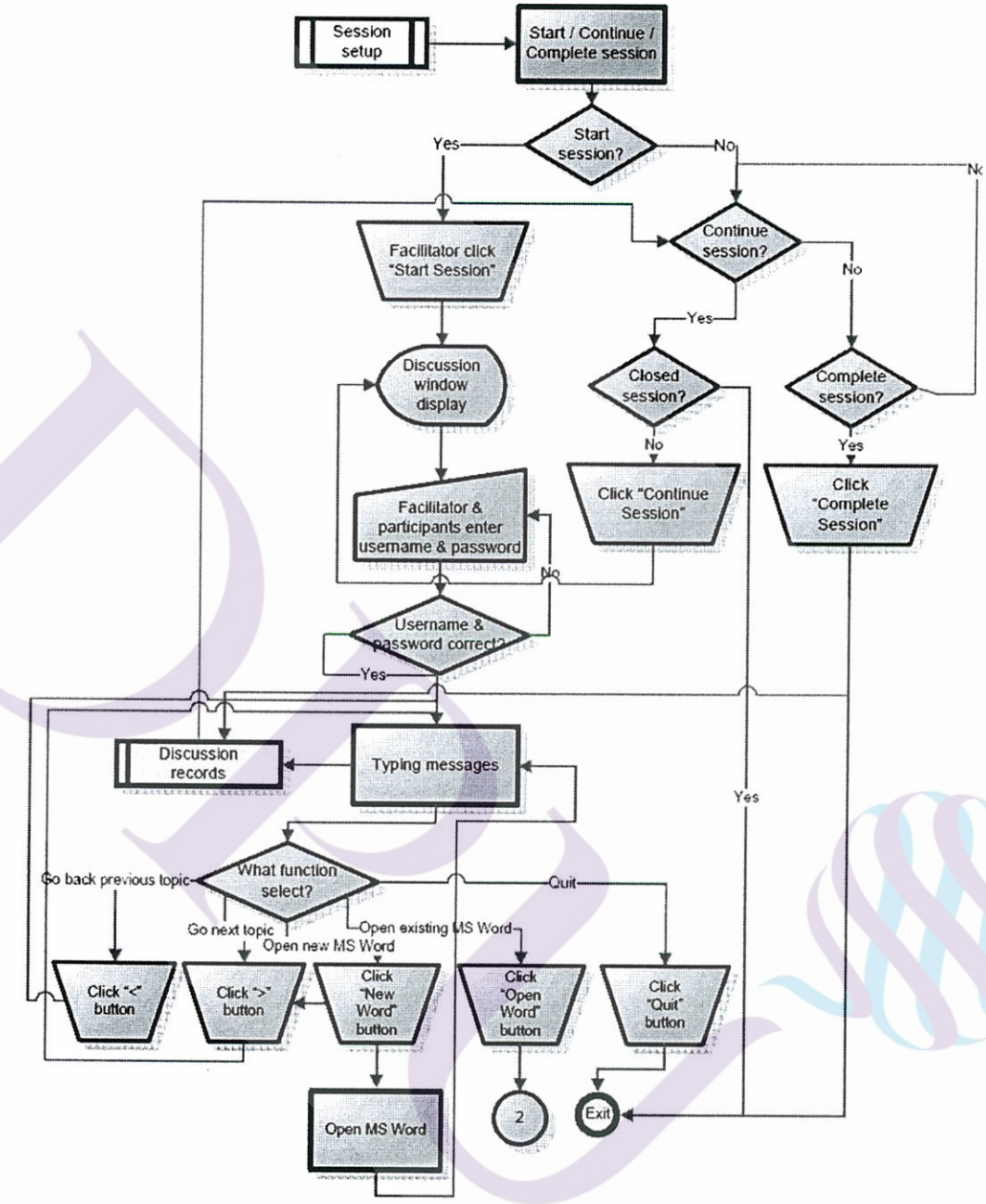


Figure Appendix 1.7: Start / Continue / Delete session

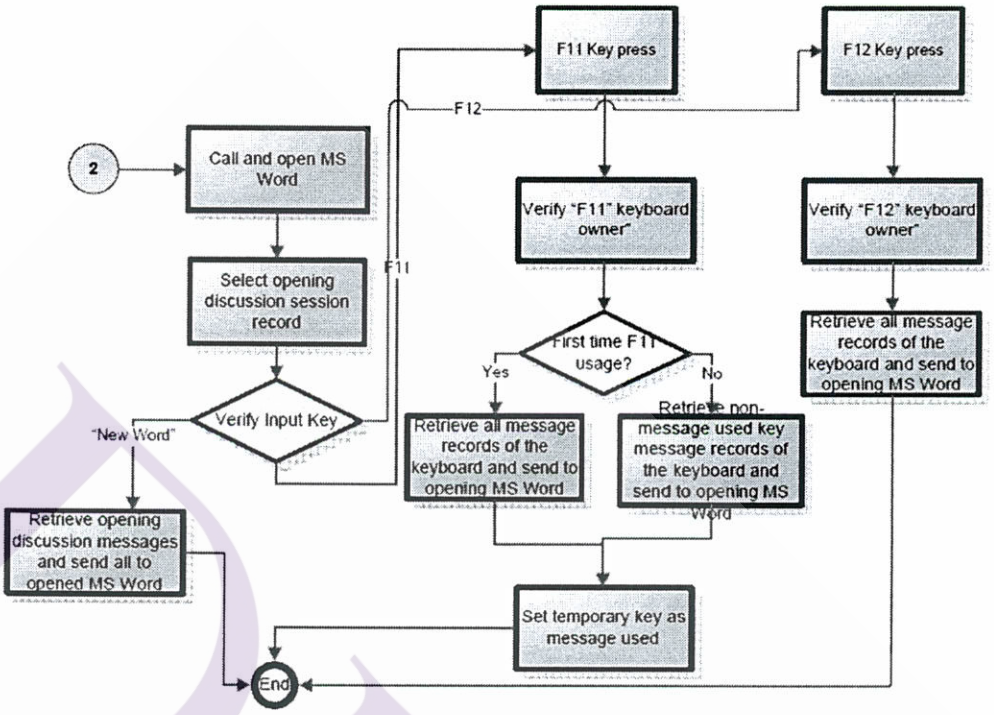


Figure Appendix 1.8: Open discussion message in MS Word and F11 & F12 Function

Appendix IV

BrainEx Group Support System Manual

The GSS used in the international graduate focus group laboratory experiment was BrainEx. The GSS is based on developing an agenda which can be completed by groups in the meeting room, face-to-face participation, or same time/same place discussion.

BrainEx is a window based application with a client computer either personal computer or laptop and participant keyboards. The computer was used by the researcher to create and control the meeting agenda and to run the discussion. For those setup meetings in the classroom, the computer/BrainEx screen was projected at the front of the meeting room so that students could see the input. Each student has an individual keyboard connected to BrainEx where they anonymously enter their ideas on the agenda tasks and questions in each personal message box. The generated ideas are collected and displayed on the BrainEx common public screen and the group input can be viewed on the public screen.

Screenshots of BrainEx:

The following screens illustrate the steps of using BrainEx

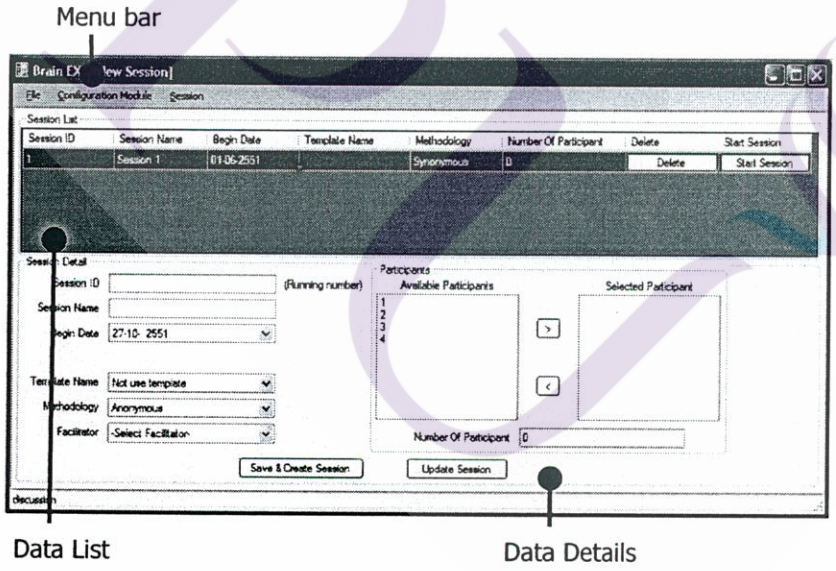
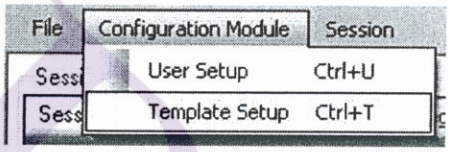


Figure Appendix 2.1 presents BrainEx Start page

When using BrainEx for the first time, the researcher has to create user accounts foreach participant. The objective of this feature mainly supports anonymity and continuous meeting session.

Figure Appendix 2.2 Create a new BrainEx user account

The agenda template also needs to be prepared before creating a session of each focus group classroom situation.



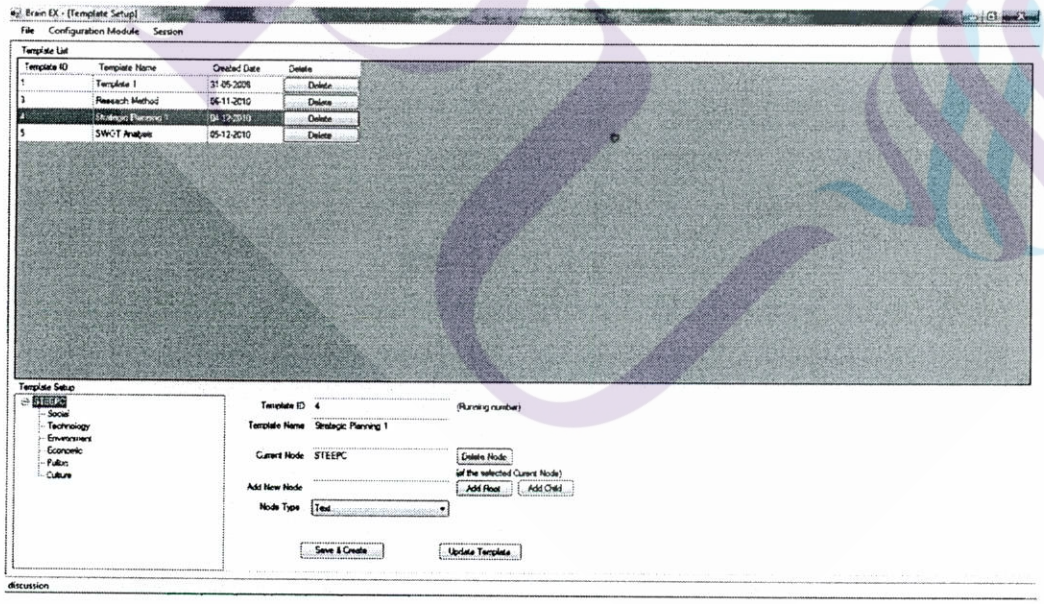


Figure Appendix 2.3 Create a new BrainEx participant template

Before focus group participation starts, the researcher has to prepare a new session with the agenda template, facilitator, and participants selection.

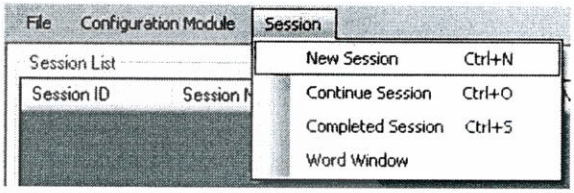


Figure Appendix 2.4 Create a new BrainEx session

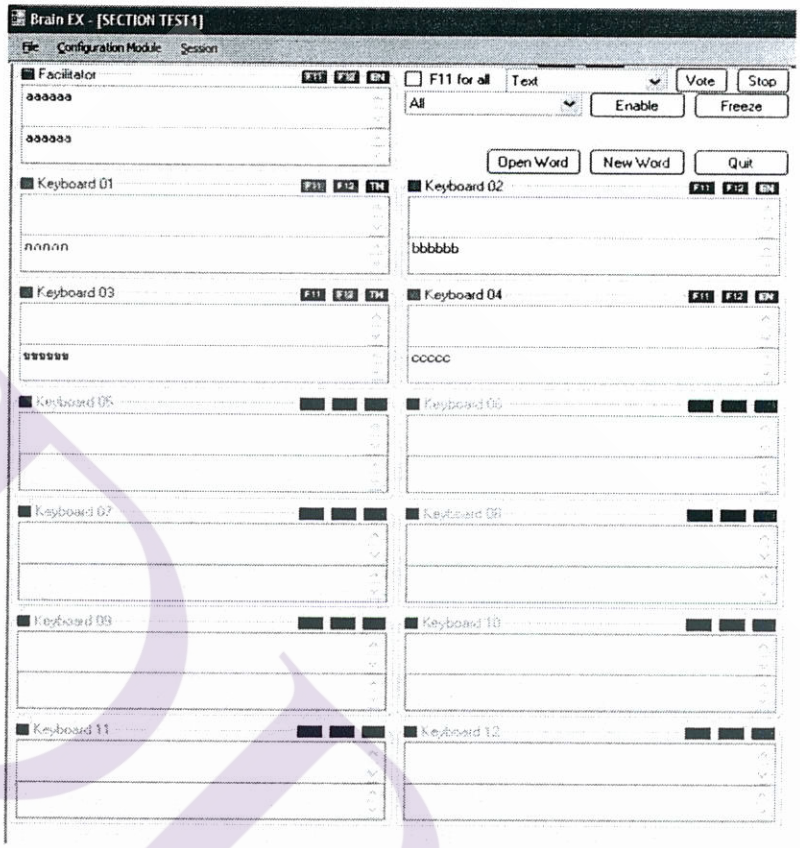


Figure Appendix 2.5 Discussion screen

BrainEx allows displaying all generated ideas in the Microsoft Word application alongside BrainEx discussion screen.

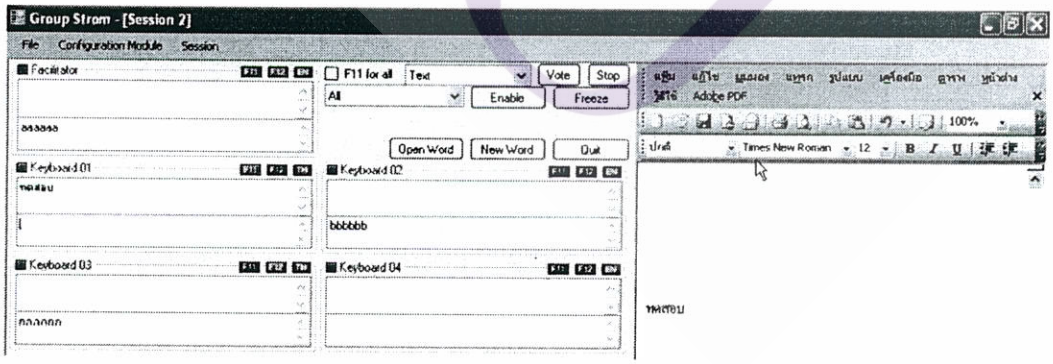


Figure Appendix 2.6 Discussion screen with opening Microsoft Word

Appendix V

Questionnaire

The research questionnaire applied and adapted based on the questionnaire set of Neiderman & Bryson (1998) in Influence of Computer-Based Meeting Support on Process and Outcomes for a Divisional Coordinating Group, Group Decision and Negotiation, p. 318-319.

This research is aimed at analyzing the classroom learning performance in terms of quality, efficiency, and effectiveness of classroom discussion process and outcome and also perception of students on electronic classroom experience such as satisfaction on the process and outcomes. By completing the questionnaire, this research is given valuable input to the investigation.

Guarantee of Confidentiality

This questionnaire will be used only for the evaluation of the BrainEx purpose only. All the results will not be presented as the raw information. No personal information will be released and strictly keep confidential.

Propose of Questionnaire

Post meeting questionnaire

Date_____ Interviewee Name/Title_____

We are interested in how your group approached its task. Given below are certain statements which reflect feelings about your group activities. Please indicate in the space provided the degree to which each statement applies to you or your group. Indicate your choice by circling the appropriate marker. There is no right or wrong answer. Work quickly – just record your first impressions.

1. The overall quality of the discussion was: |----|----|----|----|----|
   Poor Good

2. The discussion, on the whole, was: |----|----|----|----|----|
   Ineffective Effective

3. The outcome of the discussion was: |----|----|----|----|----|
   Unsatisfactory Satisfactory

4. The discussion was: |----|----|----|----|----|
   Incompetently Competently

5. The issues explored in this discussion were:

executed executed
|----|----|----|----|----|
Trivial Substantial

6. The content of the discussion was:

|----|----|----|----|----|
Carelessly developed Carefully developed

7. The manner in which the participants examined issues was:

|----|----|----|----|----|
Nonconstructive Constructive

8. The group's movement towards reaching a conclusion on the discussion question, under the circumstances, was:

|----|----|----|----|----|
Insignificant Significant

9. How would you describe your group's participation process?

a. |----|----|----|----|----|
Efficient Inefficient
b. |----|----|----|----|----|
Coordinated Uncoordinated
c. |----|----|----|----|----|
Fair Unfair
d. |----|----|----|----|----|
Understandable Confusing
e. |----|----|----|----|----|
Satisfying Dissatisfying

We would like to gather information about your attitude towards your group problem solving / Decision making / Brainstorming process and the resolution of this agenda item. Given below are certain statements to reflect your attitudes. Please indicate your level of agreement or disagreement with each statement by circling one of the appropriate markers. You might respond to each statement with one of the following responses

1	2	3	4	5	6	7
Strongly disagree	Greatly disagree	Disagree	Neutral	Agree	Greatly agree	Strongly agree

1. Our group's conclusions are good ones. 1---2---3---4---5---6---7

2. People in the case who would be affected by our conclusions would probably be satisfied with them. 1---2---3---4---5---6---7

3. I am not sure our conclusions are appropriate. 1---2---3---4---5---6---7

4. We could easily justify our conclusion.	1---2---3---4---5---6---7
5. I am not confident about our conclusion.	1---2---3---4---5---6---7
6. I am committed to my group's conclusions and assumptions.	1---2---3---4---5---6---7
7. I am satisfied with my group's conclusions and assumptions.	1---2---3---4---5---6---7
8. I am in complete agreement with my group's conclusions and assumptions.	1---2---3---4---5---6---7
9. The group decision process uncovered valid considerations that I did not think of individually.	1---2---3---4---5---6---7
10. The group decision process made me critically reevaluate the validity of considerations that I had thought of individually.	1---2---3---4---5---6---7
11. The meeting enabled me to better understand the positions and opinions of other members of my group	1---2---3---4---5---6---7
12. Compared to when the meeting started, I have a much better understanding of how other members of my group view this issue.	1---2---3---4---5---6---7
13. I feel personally committed to participate in implementing the group's next steps regarding this task.	1---2---3---4---5---6---7
14. I believe that the group's next steps will be successfully completed.	1---2---3---4---5---6---7
15. I believe that the successful completion of the group's next steps will be helpful.	1---2---3---4---5---6---7
16. I feel that the GDSS (BrainEx) help me understand more on the context and process of the today's subject	1---2---3---4---5---6---7
17. At which degree of involvement which you have participated or involved in the today's subject before.	1---2---3---4---5---6---7
18. If you have to be involved in the similar learning process of any subject, will you use / prefer to use this GDSS (BrainEx)?	1---2---3---4---5---6---7

19. Will you recommend others to use this GDSS (BrainEx) in the future?	1---2---3---4---5---6---7
20. At what degree, do you think this GDSS (BrainEx) helps you or allow you to contribute to overcome the language barrier or difficulties?	1---2---3---4---5---6---7
21. At what degree, do you think this GDSS (BrainEx) enhance more on discussion or brainstorming session without worrying on language capabilities?	1---2---3---4---5---6---7
22. At what degree, do you think this GDSS (BrainEx) useful brainstorming in multi language session or workshop?	1---2---3---4---5---6---7

Appendix VI

Questionnaire Results

Pilot Test:

Number of Particip = 12

Part 1

1	2	3	4	5	Questions	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
		3	8	1	The overall quality of the discussion	3	3	3	4	4	4	4	4	4	4	4	5	3.83	0.58
		3	8	1	The discussion, on the whole was	3	3	3	4	4	4	4	4	4	4	4	5	3.83	0.58
		4	6	2	The outcome of the discussion was	3	3	3	3	4	4	4	4	4	4	5	5	3.83	0.72
		4	6	2	The discussion was fit	3	3	3	3	4	4	4	4	4	4	5	5	3.83	0.72
		2	6	2	The issues explored in this discussion were	3	3	4	4	4	4	4	4	4	4	5	5	4.00	0.60
		3	7	2	The content of the discussion was	3	3	3	4	4	4	4	4	4	4	5	5	3.92	0.67
		3	7	2	The manner in which the participants examined issues was	3	3	3	4	4	4	4	4	4	4	5	5	3.92	0.67
		4	6	2	The group's movement towards reaching a conclusion on the discussion question, under the	3	3	3	3	4	4	4	4	4	4	5	5	3.83	0.72
5	**4**	**3**	**2**	**1**	**How would you describe your group's participation process**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**M**	**SD**
1	6	3	1	1	a. Efficient - Inefficient	5	4	4	4	4	4	4	3	3	3	2	1	3.42	1.08
	7	4	1		b. Coordinated - Uncoordinated	4	4	4	4	4	4	4	3	3	3	3	2	3.50	0.67
1	7	3	1		c. Fair - Unfair	5	4	4	4	4	4	4	4	3	3	3	2	3.67	0.78
1	7	4			d. Understandable -Confusing	5	4	4	4	4	4	4	4	3	3	3	3	3.75	0.62
2	7	3			e. Satisfying - Dissatisfying	5	5	4	4	4	4	4	4	4	3	3	3	3.92	0.67

Number of Participation = 12

Part 2

1	2	3	4	5	6	7	Questions	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
			5	4	3		Our group's conclusions are good ones	4	4	4	4	4	5	5	5	5	6	6	6	4.83	0.83
			3	4	5		People in the case who would be affected by our conclusions would probably be satisfied with them	4	4	4	5	5	5	5	6	6	6	6	6	5.17	0.83
	1	3	4	2	2		I am not sure our conclusions are appropriate	2	3	3	3	4	4	4	4	5	5	6	6	4.08	1.24
		2	4	3	3		We could easily justify our conclusion	3	3	4	4	4	4	5	5	5	6	6	6	4.58	1.08
	1	2	4	3	2		I am not confident about our conclusion	2	3	3	4	4	4	4	5	5	5	6	6	4.25	1.22
		3	4	4	1		I am committed to my group's conclusions and assumptions	3	3	3	4	4	4	4	5	5	5	5	6	4.25	0.97
		1	4	4	3		I am satisfied with my group's conclusions and assumptions	3	4	4	4	4	5	5	5	5	6	6	6	4.75	0.97
		1	3	3	5		I am in complete agreement with my group's conclusions and assumptions	3	4	4	4	5	5	5	6	6	6	6	6	5.00	1.04
		2	5	3	2		The group decision process uncovered valid considerations that I did not think of	3	3	4	4	4	4	4	5	5	5	6	6	4.42	1.00
		1	6	3	2		The group decision process made me critically reevaluate the validity of considerations that I had thought of individually	3	4	4	4	4	4	4	5	5	5	6	6	4.50	0.90
		2	4	3	2	1	The meeting enabled me to better understand the positions and opinions of other members of my group	3	3	4	4	4	4	5	5	5	6	6	7	4.67	1.23
		1	3	4	3	1	Compared to when the meeting started. I have a much better understanding of how other members of my group view this issue	3	4	4	4	5	5	5	5	6	6	6	7	5.00	1.13
			5	4	2	1	I feel personally committed to participate in implementing the group's next steps regarding this task	4	4	4	4	4	5	5	5	5	6	6	7	4.92	1.00
		1	4	4	3		I believe that the group's next steps will be successfully completed	3	4	4	4	4	5	5	5	5	6	6	6	4.75	0.97
		2	3	4	3		I believe that the successful completion of the group's next steps will be helpful	3	3	4	4	4	5	5	5	5	6	6	6	4.67	1.07
			4	5	2	1	I feel that the GDSS (BrainEx) help me understand more on the context and process of the today's subject	4	4	4	4	5	5	5	5	5	6	6	7	5.00	0.95
		1	2	4	5		At which degree of involvement which you have participated or involved in the today's subject before	3	4	4	5	5	5	5	6	6	6	6	6	5.08	1.00
			2	3	5	2	If you have to be involved in the similar learning process of any subject, will you use / prefer to use this GDSS (BrainEx)?	4	4	5	5	5	6	6	6	6	6	7	7	5.58	1.00
			2	3	4	3	Will you recommend others to use this GDSS (BrainEx) in the future?	4	4	5	5	5	6	6	6	6	7	7	7	5.67	1.07
			2	4	3	3	At what degree, do you think this GDSS (BrainEx) helps you to overcome the language barrier or difficulties?	4	4	5	5	5	5	6	6	6	7	7	7	5.58	1.08
		1	2	4	2	3	At what degree, do you think this GDSS (BrainEx) enhance or allow you to contribute more on discussion or brainstorming session without worrying on language capabilities	3	4	4	5	5	5	5	6	6	7	7	7	5.33	1.30
			3	3	3	3	At what degree, do you think this GDSS (BrainEx) useful in multi language brainstorming session or workshop	4	4	4	5	5	5	6	6	6	7	7	7	5.50	1.17

Experimental 1:

Number of Participatio = 12

Part 1

1	2	3	4	5	Questions	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
		2	7	3	The overall quality of the discussion was	3	3	4	4	4	4	4	4	4	5	5	5	4.08	0.67
		3	6	3	The discussion, on the whole was	3	3	3	4	4	4	4	4	4	5	5	5	4.00	0.74
			9	3	The outcome of the discussion was	4	4	4	4	4	4	4	4	4	5	5	5	4.25	0.45
		3	6	3	The discussion was fit	3	3	3	4	4	4	4	4	4	5	5	5	4.00	0.74
	1	4	5	2	The issues explored in this discussion were	2	3	3	3	3	4	4	4	4	4	5	5	3.67	0.89
		3	7	2	The content of the discussion was	3	3	3	4	4	4	4	4	4	4	5	5	3.92	0.67
		2	7	3	The manner in which the participants examined issues was	3	3	4	4	4	4	4	4	4	5	5	5	4.08	0.67
		3	5	4	The group's movement towards reaching a conclusion on the discussion question, under the circumstances, was	3	3	3	4	4	4	4	4	5	5	5	5	4.08	0.79

5	4	3	2	1	How would you describe your group's participation process	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
1	5	3	2	1	a. Efficient - Inefficient	5	4	4	4	4	4	3	3	3	2	2	1	3.25	1.14
1	7	3	1		b. Coordinated - Uncoordinated	5	4	4	4	4	4	4	4	3	3	3	2	3.67	0.78
1	6	3	2		c. Fair - Unfair	5	4	4	4	4	4	4	3	3	3	2	2	3.50	0.90
1	6	4	1		d. Understandable -Confusing	5	4	4	4	4	4	4	3	3	3	3	2	3.58	0.79
2	8	2			e. Satisfying - Dissatisfying	5	5	4	4	4	4	4	4	4	4	3	3	4.00	0.60

Number of Participation = 12

Part 2

1	2	3	4	5	6	7	Questions	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
			2	4	6		Our group's conclusions are good ones	4	4	5	5	5	5	6	6	6	6	6	6	5.33	0.78
			2	5	5		People in the case who would be affected by our conclusions would probably be satisfied with them	4	4	5	5	5	5	5	6	6	6	6	6	5.25	0.75
	1	3	4	1	2	1	I am not sure our conclusions are appropriate	2	3	3	3	4	4	4	4	5	6	6	7	4.25	1.48
		2	1	6	3		We could easily justify our conclusion	3	3	4	5	5	5	5	5	5	6	6	6	4.83	1.03
	1	5	3	1	1	1	I am not confident about our conclusion	2	3	3	3	3	3	4	4	4	5	6	7	3.92	1.44
			2	8	2		I am committed to my group's conclusions and assumptions	4	4	5	5	5	5	5	5	5	5	6	6	5.00	0.60
			2	5	5		I am satisfied with my group's conclusions and assumptions	4	4	5	5	5	5	5	6	6	6	6	6	5.25	0.75
			3	3	5	1	I am in complete agreement with my group's conclusions and assumptions	4	4	4	5	5	5	6	6	6	6	6	7	5.33	0.98
		1	4	3	3	1	The group decision process uncovered valid considerations that I did not think of individually	3	4	4	4	4	5	5	5	6	6	6	7	4.92	1.16
		1	4	3	3	1	The group decision process made me critically reevaluate the validity of considerations that I had thought of individually	3	4	4	4	4	5	5	5	6	6	6	1	4.42	1.44
			4	2	4	2	The meeting enabled me to better understand the positions and opinions of other members of my group	4	4	4	4	5	5	6	6	6	6	7	7	5.33	1.15
		1	2	3	3	3	Compared to when the meeting started, I have a much better understanding of how other members of my group view this issue	3	4	4	5	5	5	6	6	6	7	7	7	5.42	1.31
			2	4	6		I feel personally committed to participate in implementing the group's next steps regarding this task	4	4	5	5	5	5	6	6	6	6	6	6	5.33	0.78
		1	1	4	4	2	I believe that the group's next steps will be successfully completed	3	4	5	5	5	5	6	6	6	6	7	7	5.42	1.16
			3	2	7		I believe that the successful completion of the group's next steps will be helpful	4	4	4	5	5	6	6	6	6	6	6	6	5.33	0.89
			2	6	2	2	I feel that the GDSS (BrainEx) help me understand more on the context and process of the today's subject	4	4	5	5	5	5	5	5	6	6	7	7	5.33	0.98
		1	3	3	5		At which degree of involvement which you have participated or involved in the today's subject before	3	4	4	4	5	5	5	6	6	6	6	6	5.00	1.04
			2	3	4	3	If you have to be involved in the similar learning process of any subject, will you use / prefer to use this GDSS (BrainEx)?	4	4	5	5	5	6	6	6	6	7	7	7	5.67	1.07
			2	3	2	5	Will you recommend others to use this GDSS (BrainEx) in the future?	4	4	5	5	5	6	6	7	7	7	7	7	5.83	1.19
			2	3	2	5	At what degree, do you think this GDSS (BrainEx) helps you to overcome the language barrier or difficulties?	4	4	5	5	5	6	6	7	7	7	7	7	5.83	1.19
		1	1	5	1	4	At what degree, do you think this GDSS (BrainEx) enhance or allow you to contribute more on discussion or brainstorming session without worrying on language capabilities	3	4	5	5	5	5	5	6	7	7	7	7	5.50	1.31
			3	3	3	3	At what degree, do you think this GDSS (BrainEx) useful in multi language brainstorming session or workshop	4	4	4	5	5	5	6	6	6	7	7	7	5.50	1.17

Experimental 2:

Number of Participation = 12

Part 1

1	2	3	4	5	Questions	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
		3	8	1	The overall quality of the discussion was	3	3	3	4	4	4	4	4	4	4	4	5	3.83	0.58
		3	7	2	The discussion, on the whole was	3	3	3	4	4	4	4	4	4	4	5	5	3.92	0.67
		2	7	3	The outcome of the discussion was	3	3	4	4	4	4	4	4	4	5	5	5	4.08	0.67
		3	6	3	The discussion was fit	3	3	3	4	4	4	4	4	4	5	5	5	4.00	0.74
		3	7	2	The issues explored in this discussion were	3	3	3	4	4	4	4	4	4	4	5	5	3.92	0.67
		4	6	2	The content of the discussion was	3	3	3	3	4	4	4	4	4	4	5	5	3.83	0.72
		2	8	2	The manner in which the participants examined issues was	3	3	4	4	4	4	4	4	4	4	5	5	4.00	0.60
		2	6	4	The group's movement towards reaching a conclusion on the discussion question, under the circumstances, was	3	3	4	4	4	4	4	4	5	5	5	5	4.17	0.72

5	4	3	2	1	How would you describe your group's participation process	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
1	6	4	1		a. Efficient - Inefficient	5	4	4	4	4	4	4	3	3	3	3	2	3.58	0.79
	6	5	1		b. Coordinated - Uncoordinated	4	4	4	4	4	4	3	3	3	3	3	2	3.42	0.67
1	7	3	1		c. Fair - Unfair	5	4	4	4	4	4	4	4	3	3	3	2	3.67	0.78
1	7	3	1		d. Understandable -Confusing	5	4	4	4	4	4	4	4	3	3	3	2	3.67	0.78
3	7	2			e. Satisfying - Dissatisfying	5	5	5	4	4	4	4	4	4	4	3	3	4.08	0.67

Number of Participation = 12

Part 2

1	2	3	4	5	6	7	Questions	1	2	3	4	5	6	7	8	9	10	11	12	M	SD
			2	5	5		Our group's conclusions are good ones	4	4	5	5	5	5	5	6	6	6	6	6	5.25	0.75
			2	5	5		People in the case who would be affected by our conclusions would probably be satisfied with them	4	4	5	5	5	5	5	6	6	6	6	6	5.25	0.75
	1	3	4	2	1	1	I am not sure our conclusions are appropriate	2	3	3	3	4	4	4	4	5	5	6	7	4.17	1.40
		2	2	5	3		We could easily justify our conclusion	3	3	4	4	5	5	5	5	5	6	6	6	4.75	1.06
	1	1	3	5	2		I am not confident about our conclusion	2	3	4	4	4	5	5	5	5	5	6	6	4.50	1.17
			2	8	2		I am committed to my group's conclusions and assumptions	4	4	5	5	5	5	5	5	5	5	6	6	5.00	0.60
			2	5	3	2	I am satisfied with my group's conclusions and assumptions	4	4	5	5	5	5	5	6	6	6	7	7	5.42	1.00
			3	3	6		I am in complete agreement with my group's conclusions and assumptions	4	4	4	5	5	5	6	6	6	6	6	6	5.25	0.87
		1	4	3	4		The group decision process uncovered valid considerations that I did not think of individually	3	4	4	4	4	5	5	5	6	6	6	6	4.83	1.03
		1	4	3	4		The group decision process made me critically reevaluate the validity of considerations that I had thought of individually	3	4	4	4	4	5	5	5	6	6	6	6	4.83	1.03
			4	2	4	2	The meeting enabled me to better understand the positions and opinions of other members of my group	4	4	4	4	5	5	6	6	6	6	7	7	5.33	1.15
			3	2	5	2	Compared to when the meeting started, I have a much better understanding of how other members of my group view this issue	4	4	4	5	5	6	6	6	6	6	7	7	5.50	1.09
			2	3	5	1	I feel personally committed to participate in implementing the group's next steps regarding this task	4	4	5	5	5	6	6	6	6	6	6	7	5.50	0.90
			3	4	4	1	I believe that the group's next steps will be successfully completed	4	4	4	5	5	5	5	6	6	6	6	7	5.25	0.97
			3	3	6		I believe that the successful completion of the group's next steps will be helpful	4	4	4	5	5	5	6	6	6	6	6	6	5.25	0.87
			2	6	2	2	I feel that the GDSS (BrainEx) help me understand more on the context and process of the today's subject	4	4	5	5	5	5	5	5	6	6	7	7	5.33	0.98
			2	3	6	1	At which degree of involvement which you have participated or involved in the today's subject before	4	4	5	5	5	6	6	6	6	6	6	7	5.50	0.90
		1	2	2	4	3	If you have to be involved in the similar learning process of any subject, will you use / prefer to use this GDSS (BrainEx)?	3	4	4	5	5	6	6	6	6	7	7	7	5.50	1.31
			2	3	4	3	Will you recommend others to use this GDSS (BrainEx) in the future?	4	4	5	5	5	6	6	6	6	7	7	7	5.67	1.07
			2	3	3	4	At what degree, do you think this GDSS (BrainEx) helps you to overcome the language barrier or difficulties?	4	4	5	5	5	6	6	6	7	7	7	7	5.75	1.14
		1	1	4	2	4	At what degree, do you think this GDSS (BrainEx) enhance or allow you to contribute more on discussion or brainstorming session without worrying on language capabilities	3	4	5	5	5	5	6	6	7	7	7	7	5.58	1.31
			3	3	3	3	At what degree, do you think this GDSS (BrainEx) useful in multi language brainstorming session or workshop	4	4	4	5	5	5	6	6	6	7	7	7	5.50	1.17

Experimental 3:

Number of Participation = 11

Part 1

1	2	3	4	5	Questions	1	2	3	4	5	6	7	8	9	10	11	M	SD
		2	7	2	The overall quality of the discussion was	3	3	4	4	4	4	4	4	4	5	5	4.00	0.63
		3	5	3	The discussion, on the whole was	3	3	3	4	4	4	4	4	5	5	5	4.00	0.77
		2	7	2	The outcome of the discussion was	3	3	4	4	4	4	4	4	4	5	5	4.00	0.63
		3	5	3	The discussion was fit	3	3	3	4	4	4	4	4	5	5	5	4.00	0.77
	1	3	5	2	The issues explored in this discussion were	2	3	3	3	4	4	4	4	4	5	5	3.73	0.90
		3	6	2	The content of the discussion was	3	3	3	4	4	4	4	4	4	5	5	3.91	0.70
		2	6	3	The manner in which the participants examined issues was	3	3	4	4	4	4	4	4	5	5	5	4.09	0.70
		3	4	4	The group's movement towards reaching a conclusion on the discussion question, under the circumstances, was	3	3	3	4	4	4	4	5	5	5	5	4.09	0.83
5	**4**	**3**	**2**	**1**	**How would you describe your group's participation process**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**M**	**SD**
1	4	3	2	1	a. Efficient - Inefficient	5	4	4	4	4	3	3	3	2	2	1	3.18	1.17
	7	3	1		b. Coordinated - Uncoordinated	4	4	4	4	4	4	4	3	3	3	2	3.55	0.69
1	6	2	2		c. Fair - Unfair	5	4	4	4	4	4	4	3	3	2	2	3.55	0.93
1	6	3	1		d. Understandable -Confusing	5	4	4	4	4	4	4	3	3	3	2	3.64	0.81
2	7	2			e. Satisfying - Dissatisfying	5	5	4	4	4	4	4	4	4	3	3	4.00	0.63

Number of Participation = 11

Part 2

1	2	3	4	5	6	7	Questions	1	2	3	4	5	6	7	8	9	10	11	M	SD
			2	4	5		Our group's conclusions are good ones	4	4	5	5	5	5	6	6	6	6	6	5.27	0.79
			2	4	5		People in the case who would be affected by our conclusions would probably be satisfied with them	4	4	5	5	5	5	6	6	6	6	6	5.27	0.79
	1	3	4		2	1	I am not sure our conclusions are appropriate	2	3	3	3	4	4	4	4	6	6	7	4.18	1.54
		2	1	5	3		We could easily justify our conclusion	3	3	4	5	5	5	5	5	6	6	6	4.82	1.08
	1		3	5	1	1	I am not confident about our conclusion	2	4	4	4	5	5	5	5	5	6	7	4.73	1.27
			2	8	1		I am committed to my group's conclusions and assumptions	4	4	5	5	5	5	5	5	5	5	6	4.91	0.54
			2	4	4	1	I am satisfied with my group's conclusions and assumptions	4	4	5	5	5	5	6	6	6	6	7	5.36	0.92
			3	2	5	1	I am in complete agreement with my group's conclusions and assumptions	4	4	4	5	5	6	6	6	6	6	7	5.36	1.03
		1	4	2	3	1	The group decision process uncovered valid considerations that I did not think of individually	3	4	4	4	4	5	5	6	6	6	7	4.91	1.22
		1	4	3	2	1	The group decision process made me critically reevaluate the validity of considerations that I had thought of individually	3	4	4	4	4	5	5	5	6	6	7	4.82	1.17
			4	2	3	2	The meeting enabled me to better understand the positions and opinions of other members of my group	4	4	4	4	5	5	6	6	6	7	7	5.27	1.19
		1	2	2	3	3	Compared to when the meeting started, I have a much better understanding of how other members of my group view this issue	3	4	4	5	5	6	6	6	7	7	7	5.45	1.37
			2	3	6		I feel personally committed to participate in implementing the group's next steps regarding this task	4	4	5	5	5	6	6	6	6	6	6	5.36	0.81
		1		4	4	2	I believe that the group's next steps will be successfully completed	3	5	5	5	5	6	6	6	6	7	7	5.55	1.13
			3	1	7		I believe that the successful completion of the group's next steps will be helpful	4	4	4	5	6	6	6	6	6	6	6	5.36	0.92
			2	6	1	2	I feel that the GDSS (BrainEx) help me understand more on the context and process of the today's subject	4	4	5	5	5	5	5	5	6	7	7	5.27	1.01
		1	2	3	5		At which degree of involvement which you have participated or involved in the today's subject before	3	4	4	5	5	5	6	6	6	6	6	5.09	1.04
			2	2	4	3	If you have to be involved in the similar learning process of any subject, will you use / prefer to use this GDSS (BrainEx)?	4	4	5	5	6	6	6	6	7	7	7	5.73	1.10
			2	2	2	5	Will you recommend others to use this GDSS (BrainEx) in the future?	4	4	5	5	6	6	7	7	7	7	7	5.91	1.22
			2	3	1	5	At what degree, do you think this GDSS (BrainEx) helps you to overcome the language barrier or difficulties?	4	4	5	5	5	6	7	7	7	7	7	5.82	1.25
		1	1	4	1	4	At what degree, do you think this GDSS (BrainEx) enhance or allow you to contribute more on discussion or brainstorming session without worrying on language capabilities	3	4	5	5	5	5	6	7	7	7	7	5.55	1.37
			3	2	3	3	At what degree, do you think this GDSS (BrainEx) useful in multi language brainstorming session or workshop	4	4	4	5	5	6	6	6	7	7	7	5.55	1.21

Appendix VII

BrainEx Programming

Using dll

- Interop.Office.dll
- Interop.VBIDE.dll
- Interop.Word.dll
- Sdgt 2.0.1.0.dll
- WinWordControl.dll

AssemblyInfo.cs

```
using System.Reflection;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;

// General Information about an assembly is controlled through the following
// set of attributes. Change these attribute values to modify the information
// associated with an assembly.
[assembly: AssemblyTitle("Brain EX")]
[assembly: AssemblyDescription("")]
[assembly: AssemblyConfiguration("")]
[assembly: AssemblyCompany("")]
[assembly: AssemblyProduct("Brain EX")]
[assembly: AssemblyCopyright("")]
[assembly: AssemblyTrademark("")]
[assembly: AssemblyCulture("")]

// Setting ComVisible to false makes the types in this assembly not visible
// to COM components.  If you need to access a type in this assembly from
// COM, set the ComVisible attribute to true on that type.
[assembly: ComVisible(false)]

// The following GUID is for the ID of the typelib if this project is exposed to COM
[assembly: Guid("75e3de74-83ad-459d-a0a9-17f4a297b86a")]

// Version information for an assembly consists of the following four values:
//
//      Major Version
//      Minor Version
//      Build Number
//      Revision
//
[assembly: AssemblyVersion("1.0.0.0")]
[assembly: AssemblyFileVersion("1.0.0.0")]
```

Program.cs

```
using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace GroupStorm
{
```

```
    static class Program
    {
      /// <summary>
      /// The main entry point for the application.
      /// </summary>

      [STAThread]
      static void Main()
      {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);
        Application.Run(new MDIParent1());
      }
    }
}
```

MDIParent1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace GroupStorm
{
  public partial class MDIParent1 : Form
  {
    public MDIParent1()
    {
      InitializeComponent();
    }

    /// <summary>
    /// Click Exit menu
    /// </summary>
    private void ExitToolsStripMenuItem_Click(object sender, EventArgs e)
    {
      //Exit Application
      Application.Exit();
    }

    /// <summary>
    /// Click Display Status Bar menu
    /// </summary>
    private void StatusBarToolStripMenuItem_Click(object sender, EventArgs e)
    {
      //Set display or hide status bar
      statusStrip.Visible = statusBarToolStripMenuItem.Checked;
    }

    /// <summary>
    /// Set Text Status
    /// </summary>
    /// <param name="Status"></param>
```

```
public void SetStatus(string Status)
{
   toolStripStatusLabel.Text = Status;
}

/// <summary>
/// Click UserSetup menu
/// </summary>
private void UserSetupToolStripMenuItem_Click_1(object sender, EventArgs e)
{
   bool bDisplay = false;

   foreach (Form childForm in this.MdiChildren) //Get all opened form
   {
      if (childForm.GetType() == typeof(UserSetup)) //UserSetup form opened
      {
         //Display UserSetup form
         childForm.Focus();
         bDisplay = true;
      }
      else
      {
         //Close other form
         childForm.Close();
      }
   }

   if (!bDisplay)
   {
      //Display UserSetup form
      UserSetup childForm = new UserSetup();
      childForm.MdiParent = this;
      childForm.WindowState = FormWindowState.Maximized;
      childForm.Show();
   }
}

/// <summary>
/// Click Template Setup menu
/// </summary>
private void TemplateSetupToolStripMenuItem_Click(object sender, EventArgs e)
{
   bool bDisplay = false;

   foreach (Form childForm in this.MdiChildren) //Get all opened form
   {
      if (childForm.GetType() == typeof(TemplateSetup)) //TemplateSetup form opened
      {
         //Display TemplateSetup form
         childForm.Focus();
         bDisplay = true;
      }
      else
      {
         //Close other form
         childForm.Close();
      }
```

```
        }

        if (!bDisplay)
        {
          //Display TemplateSetup form
          TemplateSetup childForm = new TemplateSetup();
          childForm.MdiParent = this;
          childForm.WindowState = FormWindowState.Maximized;
          childForm.Show();
        }
      }

      /// <summary>
      /// Click New Session menu
      /// </summary>
      private void newSessionToolStripMenuItem_Click(object sender, EventArgs e)
      {
        bool bDisplay = false;

        foreach (Form childForm in this.MdiChildren) //Get all opened form
        {
          if ((childForm.GetType() == typeof(NewSession)) && childForm.Text == "New Session")
//New Session form opened
          {
            //Display New Session form
            childForm.Focus();
            bDisplay = true;
          }
          else
          {
            //Close other form
            childForm.Close();
          }
        }

        if (!bDisplay)
        {
          //Display New Session form
          NewSession childForm = new NewSession();
          childForm.MdiParent = this;
          childForm.WindowState = FormWindowState.Maximized;
          childForm.Text = "New Session";
          childForm.Status = "New";
          childForm.Show();
        }
      }

      /// <summary>
      /// Click Continue Session menu
      /// </summary>
      private void continueSessionToolStripMenuItem_Click(object sender, EventArgs e)
      {
        bool bDisplay = false;

        foreach (Form childForm in this.MdiChildren) //Get all opened form
        {
```

```
            if ((childForm.GetType() == typeof(NewSession)) && childForm.Text == "Continue
Session") //Continue Session form opened
            {
              //Display Continue Session form
              childForm.Focus();
              bDisplay = true;
            }
            else
            {
              //Close other form
              childForm.Close();
            }
          }

          if (!bDisplay)
          {
            //Display Continue Session form
            NewSession childForm = new NewSession();
            childForm.MdiParent = this;
            childForm.WindowState = FormWindowState.Maximized;
            childForm.Text = "Continue Session";
            childForm.Status = "Pending";
            childForm.Show();
          }
        }

        /// <summary>
        /// Click Completed Session menu
        /// </summary>
        private void completedSessionToolStripMenuItem_Click(object sender, EventArgs e)
        {
          bool bDisplay = false;

          foreach (Form childForm in this.MdiChildren) //Get all opened form
          {
            if ((childForm.GetType() == typeof(NewSession)) && childForm.Text == "Completed
Session") //Completed Session form opened
            {
              //Display Completed Session form
              childForm.Focus();
              bDisplay = true;
            }
            else
            {
              //Close other form
              childForm.Close();
            }
          }

          if (!bDisplay)
          {
            //Display Completed Session form
            NewSession childForm = new NewSession();
            childForm.MdiParent = this;
            childForm.WindowState = FormWindowState.Maximized;
            childForm.Text = "Completed Session";
            childForm.Status = "Completed";
```

```
            childForm.Show();
        }
    }

    /// <summary>
    /// Start Session to brain strom
    /// </summary>
    /// <param name="SessionID"></param>
    /// <param name="Status"></param>
    /// <param name="Methodology"></param>
    public void OpenKeyboardForm(int SessionID, string Status, string Methodology)
    {
        bool bDisplay = false;

        foreach (Form childForm in this.MdiChildren) //Get all opened form
        {
            //Close all form
            childForm.Close();
        }

        if (!bDisplay)
        {
            //Display Brain Strom form
            OpenSession childForm = new OpenSession();
            childForm.MdiParent = this;
            childForm.WindowState = FormWindowState.Maximized;
            childForm.SessionID = SessionID;
            childForm.Status = Status;
            childForm.Methodology = Methodology;

            childForm.Show();
        }
    }


    /// <summary>
    /// Start Session to brain strom
    /// </summary>
    /// <param name="SessionID"></param>
    /// <param name="Status"></param>
    /// <param name="Methodology"></param>
    public void ViewKeyboardForm(int SessionID, string Status, string Methodology)
    {
        bool bDisplay = false;

        foreach (Form childForm in this.MdiChildren) //Get all opened form
        {
            //Close all form
            childForm.Close();
        }

        if (!bDisplay)
        {
            //Display Brain Strom form
            ViewSession childForm = new ViewSession();
            childForm.MdiParent = this;
            childForm.WindowState = FormWindowState.Maximized;
```

```
            childForm.SessionID = SessionID;
            childForm.Status = Status;
            childForm.Methodology = Methodology;

            childForm.Show();
        }
    }

    /// <summary>
    /// Open New Session Form
    /// </summary>
    public void OpenNewSessionForm()
    {
        //Open New Session Form
        newSessionToolStripMenuItem_Click(null, null);
    }

    private void MDIParent1_Load(object sender, EventArgs e)
    {
        //Set Application to full screen
        this.WindowState = FormWindowState.Maximized;

        //Open New Session Form
        newSessionToolStripMenuItem_Click(this, e);
    }

    public bool GetWindowWord()
    {
        return wordWindowToolStripMenuItem.Checked;
    }

    private void openWordToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Word.ApplicationClass wordApp;
        Word._Document wordDoc;

        object missing = System.Reflection.Missing.Value;
        //object fileName = "normal.doc";
        object fileName = "";
        object newTemplate = false;
        object docType = 0;
        object isVisible = true;

        //Open Empty word and activate it
        wordApp = new Word.ApplicationClass();
        wordDoc = wordApp.Documents.Add(ref fileName, ref newTemplate, ref docType, ref
isVisible);
        wordApp.Visible = true;
        wordDoc.Activate();

        //Set Paragraph Alignment
        wordApp.Selection.ParagraphFormat.Alignment =
Word.WdParagraphAlignment.wdAlignParagraphLeft;
        //Set Font Style
        wordApp.Selection.Font.Bold = (int)Word.WdConstants.wdToggle;
        //Display Text to word
        wordApp.Selection.TypeText("Name : Mod");
```

```
            //Display new Paragraph
            wordApp.Selection.TypeParagraph();
        }
    }
}
```

MDIParent1.Designer.cs

```
namespace GroupStorm
{
    partial class MDIParent1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.menuStrip = new System.Windows.Forms.MenuStrip();
            this.fileMenu = new System.Windows.Forms.ToolStripMenuItem();
            this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.viewMenu = new System.Windows.Forms.ToolStripMenuItem();
            this.statusBarToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.configurationModuleToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
            this.UserSetupToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.TemplateSetupToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.sessionToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.newSessionToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.continueSessionToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.completedSessionToolStripMenuItem = new
System.Windows.Forms.ToolStripMenuItem();
```

```
            this.wordWindowToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.openWordToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
            this.statusStrip = new System.Windows.Forms.StatusStrip();
            this.toolStripStatusLabel = new System.Windows.Forms.ToolStripStatusLabel();
            this.ToolTip = new System.Windows.Forms.ToolTip(this.components);
            this.menuStrip.SuspendLayout();
            this.statusStrip.SuspendLayout();
            this.SuspendLayout();
            //
            // menuStrip
            //
            this.menuStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.fileMenu,
            this.viewMenu,
            this.configurationModuleToolStripMenuItem,
            this.sessionToolStripMenuItem});
            this.menuStrip.Location = new System.Drawing.Point(0, 0);
            this.menuStrip.Name = "menuStrip";
            this.menuStrip.Size = new System.Drawing.Size(632, 24);
            this.menuStrip.TabIndex = 0;
            this.menuStrip.Text = "MenuStrip";
            //
            // fileMenu
            //
            this.fileMenu.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.exitToolStripMenuItem});
            this.fileMenu.ImageTransparentColor = System.Drawing.SystemColors.ActiveBorder;
            this.fileMenu.Name = "fileMenu";
            this.fileMenu.Size = new System.Drawing.Size(35, 20);
            this.fileMenu.Text = "&File";
            //
            // exitToolStripMenuItem
            //
            this.exitToolStripMenuItem.Name = "exitToolStripMenuItem";
            this.exitToolStripMenuItem.Size = new System.Drawing.Size(92, 22);
            this.exitToolStripMenuItem.Text = "E&xit";
            this.exitToolStripMenuItem.Click += new
System.EventHandler(this.ExitToolsStripMenuItem_Click);
            //
            // viewMenu
            //
            this.viewMenu.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.statusBarToolStripMenuItem});
            this.viewMenu.Name = "viewMenu";
            this.viewMenu.Size = new System.Drawing.Size(41, 20);
            this.viewMenu.Text = "&View";
            this.viewMenu.Visible = false;
            //
            // statusBarToolStripMenuItem
            //
            this.statusBarToolStripMenuItem.Checked = true;
            this.statusBarToolStripMenuItem.CheckOnClick = true;
            this.statusBarToolStripMenuItem.CheckState = System.Windows.Forms.CheckState.Checked;
            this.statusBarToolStripMenuItem.Name = "statusBarToolStripMenuItem";
            this.statusBarToolStripMenuItem.Size = new System.Drawing.Size(124, 22);
            this.statusBarToolStripMenuItem.Text = "&Status Bar";
```

```
            this.statusBarToolStripMenuItem.Click += new
 System.EventHandler(this.StatusBarToolStripMenuItem_Click);
            //
            // configurationModuleToolStripMenuItem
            //
            this.configurationModuleToolStripMenuItem.DropDownItems.AddRange(new
 System.Windows.Forms.ToolStripItem[] {
            this.UserSetupToolStripMenuItem,
            this.TemplateSetupToolStripMenuItem});
            this.configurationModuleToolStripMenuItem.Name =
 "configurationModuleToolStripMenuItem";
            this.configurationModuleToolStripMenuItem.Size = new System.Drawing.Size(121, 20);
            this.configurationModuleToolStripMenuItem.Text = "&Configuration Module";
            //
            // UserSetupToolStripMenuItem
            //
            this.UserSetupToolStripMenuItem.Name = "UserSetupToolStripMenuItem";
            this.UserSetupToolStripMenuItem.ShortcutKeys =
 ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control |
 System.Windows.Forms.Keys.U)));
            this.UserSetupToolStripMenuItem.Size = new System.Drawing.Size(187, 22);
            this.UserSetupToolStripMenuItem.Text = "&User Setup";
            this.UserSetupToolStripMenuItem.Click += new
 System.EventHandler(this.UserSetupToolStripMenuItem_Click_1);
            //
            // TemplateSetupToolStripMenuItem
            //
            this.TemplateSetupToolStripMenuItem.Name = "TemplateSetupToolStripMenuItem";
            this.TemplateSetupToolStripMenuItem.ShortcutKeys =
 ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control |
 System.Windows.Forms.Keys.T)));
            this.TemplateSetupToolStripMenuItem.Size = new System.Drawing.Size(187, 22);
            this.TemplateSetupToolStripMenuItem.Text = "&Template Setup";
            this.TemplateSetupToolStripMenuItem.Click += new
 System.EventHandler(this.TemplateSetupToolStripMenuItem_Click);
            //
            // sessionToolStripMenuItem
            //
            this.sessionToolStripMenuItem.DropDownItems.AddRange(new
 System.Windows.Forms.ToolStripItem[] {
            this.newSessionToolStripMenuItem,
            this.continueSessionToolStripMenuItem,
            this.completedSessionToolStripMenuItem,
            this.wordWindowToolStripMenuItem,
            this.openWordToolStripMenuItem});
            this.sessionToolStripMenuItem.Name = "sessionToolStripMenuItem";
            this.sessionToolStripMenuItem.Size = new System.Drawing.Size(55, 20);
            this.sessionToolStripMenuItem.Text = "&Session";
            //
            // newSessionToolStripMenuItem
            //
            this.newSessionToolStripMenuItem.Name = "newSessionToolStripMenuItem";
            this.newSessionToolStripMenuItem.ShortcutKeys =
 ((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control |
 System.Windows.Forms.Keys.N)));
           this.newSessionToolStripMenuItem.Size = new System.Drawing.Size(202, 22);
           this.newSessionToolStripMenuItem.Text = "&New Session";
```

```
            this.newSessionToolStripMenuItem.Click += new
System.EventHandler(this.newSessionToolStripMenuItem_Click);
            //
            // continueSessionToolStripMenuItem
            //
            this.continueSessionToolStripMenuItem.Name = "continueSessionToolStripMenuItem";
            this.continueSessionToolStripMenuItem.ShortcutKeys =
((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control |
System.Windows.Forms.Keys.O)));
            this.continueSessionToolStripMenuItem.Size = new System.Drawing.Size(202, 22);
            this.continueSessionToolStripMenuItem.Text = "C&ontinue Session";
            this.continueSessionToolStripMenuItem.Click += new
System.EventHandler(this.continueSessionToolStripMenuItem_Click);
            //
            // completedSessionToolStripMenuItem
            //
            this.completedSessionToolStripMenuItem.Name = "completedSessionToolStripMenuItem";
            this.completedSessionToolStripMenuItem.ShortcutKeys =
((System.Windows.Forms.Keys)((System.Windows.Forms.Keys.Control |
System.Windows.Forms.Keys.S)));
            this.completedSessionToolStripMenuItem.Size = new System.Drawing.Size(202, 22);
            this.completedSessionToolStripMenuItem.Text = "Completed &Session";
            this.completedSessionToolStripMenuItem.Click += new
System.EventHandler(this.completedSessionToolStripMenuItem_Click);
            //
            // wordWindowToolStripMenuItem
            //
            this.wordWindowToolStripMenuItem.CheckOnClick = true;
            this.wordWindowToolStripMenuItem.Name = "wordWindowToolStripMenuItem";
            this.wordWindowToolStripMenuItem.Size = new System.Drawing.Size(202, 22);
            this.wordWindowToolStripMenuItem.Text = "Word Window";
            //
            // openWordToolStripMenuItem
            //
            this.openWordToolStripMenuItem.Name = "openWordToolStripMenuItem";
            this.openWordToolStripMenuItem.Size = new System.Drawing.Size(202, 22);
            this.openWordToolStripMenuItem.Text = "Open Word";
            this.openWordToolStripMenuItem.Visible = false;
            this.openWordToolStripMenuItem.Click += new
System.EventHandler(this.openWordToolStripMenuItem_Click);
            //
            // statusStrip
            //
            this.statusStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
            this.toolStripStatusLabel});
            this.statusStrip.Location = new System.Drawing.Point(0, 431);
            this.statusStrip.Name = "statusStrip";
            this.statusStrip.Size = new System.Drawing.Size(632, 22);
            this.statusStrip.TabIndex = 2;
            this.statusStrip.Text = "StatusStrip";
            //
            // toolStripStatusLabel
            //
            this.toolStripStatusLabel.Name = "toolStripStatusLabel";
            this.toolStripStatusLabel.Size = new System.Drawing.Size(55, 17);
            this.toolStripStatusLabel.Text = "discussion";
            //
```

```
            // MDIParent1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(632, 453);
            this.Controls.Add(this.statusStrip);
            this.Controls.Add(this.menuStrip);
            this.IsMdiContainer = true;
            this.MainMenuStrip = this.menuStrip;
            this.Name = "MDIParent1";
            this.Text = "Brain EX";
            this.Load += new System.EventHandler(this.MDIParent1_Load);
            this.menuStrip.ResumeLayout(false);
            this.menuStrip.PerformLayout();
            this.statusStrip.ResumeLayout(false);
            this.statusStrip.PerformLayout();
            this.ResumeLayout(false);
            this.PerformLayout();

        }
        #endregion


        private System.Windows.Forms.MenuStrip menuStrip;
        private System.Windows.Forms.StatusStrip statusStrip;
        private System.Windows.Forms.ToolStripStatusLabel toolStripStatusLabel;
        private System.Windows.Forms.ToolStripMenuItem fileMenu;
        private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem viewMenu;
        private System.Windows.Forms.ToolStripMenuItem statusBarToolStripMenuItem;
        private System.Windows.Forms.ToolTip ToolTip;
        private System.Windows.Forms.ToolStripMenuItem configurationModuleToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem UserSetupToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem TemplateSetupToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem sessionToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem newSessionToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem continueSessionToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem completedSessionToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem wordWindowToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem openWordToolStripMenuItem;
    }
}
```

ItemObject.cs

```
amespace GroupStorm
{
    /// <summary>
    /// Class containing key and value of some entity.
    /// </summary>
    public class ItemObject
    {
        private string key;
```

```
        private object valueOfKey;

        /// <summary>
        /// Overloaded constructor.
        /// </summary>
        /// <param name="key">Key of object.</param>
        /// <param name="valueOfKey">Value of object.</param>
        public ItemObject(string key, object valueOfKey)
        {
            this.key = key;
            this.valueOfKey = valueOfKey;
        }

        /// <summary>
        /// Default constructor
        /// </summary>
        public ItemObject()
        {
            key = string.Empty;
            valueOfKey = string.Empty;
        }

        ///<summary>
        ///Returns a <see cref="T:System.String"></see> that represents the current <see
cref="T:System.Object"></see>.
        ///</summary>
        ///
        ///<returns>
        ///A <see cref="T:System.String"></see> that represents the current <see
cref="T:System.Object"></see>.
        ///</returns>
        public override string ToString()
        {
            return key;
        }

        ///<summary>
        ///Serves as a hash function for a particular type.
        ///</summary>
        ///
        ///<returns>
        ///A hash code for the current <see cref="T:System.Object"></see>.
        ///</returns>
        public override int GetHashCode()
        {
            return ToString().GetHashCode();
        }

        /// <summary>
        /// Gets or sets Key of object.
        /// </summary>
        public string Key
        {
            get { return key; }
            set { key = value; }
        }
```

```
        /// <summary>
        /// Gets or sets Value of object.
        /// </summary>
        public object ValueOfKey
        {
            get { return valueOfKey; }
            set { valueOfKey = value; }
        }
    }
}
```

KeyboardControl.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Runtime.InteropServices;

namespace GroupStorm
{
    public partial class KeyboardControl : UserControl
    {
        [DllImport("user32.dll", CharSet = CharSet.Auto, CallingConvention =
CallingConvention.StdCall)]
        //public static extern void mouse_event(long dwFlags, long dx, long dy, long cButtons, long
dwExtraInfo);
        public static extern void mouse_event(Int32 dwFlags, Int32 dx, Int32 dy, Int32 cButtons, Int32
dwExtraInfo);

        private const int MOUSEEVENTF_LEFTDOWN = 0x02;
        private const int MOUSEEVENTF_LEFTUP = 0x04;
        private const int MOUSEEVENTF_RIGHTDOWN = 0x08;
        private const int MOUSEEVENTF_RIGHTUP = 0x10;

        //Move and Click Mouse at position x,y
        public void DoMouseClick(int x, int y)
        {
            Cursor.Position = new Point(x, y);
            //Call the imported function with the cursor's current position
            int X = Cursor.Position.X;
            int Y = Cursor.Position.Y;
            mouse_event(MOUSEEVENTF_LEFTDOWN | MOUSEEVENTF_LEFTUP, X, Y, 0, 0);
        }

        private string _keyboardName = "";
        private string _keyboardID = "";
        private string _languageCulture = "";
        private int _newText = 0;
        private bool _Freeze = false;
        private int _sessionID = 0;
```

```
private int _templateDetailID = 0;
public int _selectedLine = 0;
private bool _enableF11 = false;
private bool _enableF12 = false;
public bool farcilitor = false;
public string _sessionStatus = "New";

public KeyboardControl()
{
  InitializeComponent();
}

public string KeyboardName
{
  set
  {
    _keyboardName = value;
    this.GroupBoxKeyboard.Text = "   " + value;
  }
  get
  {
    return _keyboardName;
  }
}

public string KeyboardID
{
  set
  {
    _keyboardID = value;
  }
  get
  {
    return _keyboardID;
  }
}

public string LanguageCulture
{
  set
  {
    _languageCulture = value;
    if (!string.IsNullOrEmpty(value))
    {
      this.LabelLanguage.Text = value.Substring(0, 2).ToUpper();
    }
  }
  get
  {
    return _languageCulture;
  }
}

private void KeyboardControl_Load(object sender, EventArgs e)
{
  //Set Default Input Language
  _languageCulture = InputLanguage.CurrentInputLanguage.Culture.ToString();
```

```
        }

        /// <summary>
        /// Copy text to clipboard
        /// </summary>
        private void copyToolStripMenuItem_Click(object sender, EventArgs e)
        {
            if (this.TextBoxDisplay.Focused) //Copy text from display box
            {
                if (!string.IsNullOrEmpty(this.TextBoxDisplay.SelectedText))
                {
                    //Save text to clipboard
                    Clipboard.SetText(this.TextBoxDisplay.SelectedText);
                }
            }
            else if (this.TextBoxInput.Focused) //Copy text from input box
            {
                if (!string.IsNullOrEmpty(this.TextBoxInput.SelectedText))
                {
                    //Save text to clipboard
                    Clipboard.SetText(this.TextBoxInput.SelectedText);
                }
            }
        }

        /// <summary>
        /// Paste text from clipboard to display box
        /// </summary>
        public void pasteToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //Goto new line
            if (this.TextBoxDisplay.Text != "") this.TextBoxDisplay.Text += Environment.NewLine;

            //Paste text from clipboard to display box
            this.TextBoxDisplay.Text += Clipboard.GetText();
            //focus on last line
            this.TextBoxDisplay.Select(this.TextBoxDisplay.Text.Length, 0);
            this.TextBoxDisplay.ScrollToCaret();

            //Save transaction message to DB
            SaveTransaction(_sessionID, _templateDetailID, _keyboardID,
((OpenSession)this.ParentForm).CopyString, DateTime.Now);

            //Set index of saved line of display box
            _selectedLine = this.TextBoxDisplay.Lines.Length - 1;
        }

        /// <summary>
        /// Paste text from clipboard to all display box
        /// </summary>
        private void pasteAllToolStripMenuItem_Click(object sender, EventArgs e)
        {
            //Call paste string from OpenSession Form
            ((OpenSession)this.ParentForm).PasteString();
        }

        /// <summary>
```

```
        /// Keydown on input box
        /// </summary>
        private void TextBoxInput_KeyDown(object sender, KeyEventArgs e)
        {
          int tempLength = 0;

          if (farcilitor == false)
          {
            //if (this.ParentForm.ParentForm.MainMenuStrip.Focused == true)
            //{
            //   e.SuppressKeyPress = true;
            //}

            //for (int i = 0;i<this.ParentForm.ParentForm.MainMenuStrip.Items.Count;i++)
            //{
            //   if (this.ParentForm.ParentForm.MainMenuStrip.Items[i].Selected == true)
            //   {
            //      e.SuppressKeyPress = true;
            //      return;
            //   }
            //}
          }

          switch (e.KeyCode)
          {
            #region "Press Enter"
            case Keys.Enter: //Press enter
              //Goto new line
              if (this.TextBoxDisplay.Text != "") this.TextBoxDisplay.Text += Environment.NewLine;

              //display new text in display box
              this.TextBoxDisplay.Text += this.TextBoxInput.Text.ToString();

              //focus on last line
              this.TextBoxDisplay.Focus();
              this.TextBoxDisplay.ScrollToCaret();

              //Save transaction message to DB
              SaveTransaction(_sessionID, _templateDetailID, _keyboardID, this.TextBoxInput.Text,
DateTime.Now);

              //Clear input box
              this.TextBoxInput.Clear();
              this.TextBoxInput.Focus();

              //Set index of saved line of display box
              _selectedLine = this.TextBoxDisplay.Lines.Length - 1;
              break;
            #endregion

            #region "Press F11"
            case Keys.F11: //Press F11
              try
              {
                if (farcilitor)
                {
                  if (e.Control && e.Alt) //Press Ctrl + Alt + F11
```

```
            {
              ((OpenSession)this.ParentForm).ShortCutF11forAll();
              return;
            }
            else if (e.Control) //Press Ctrl + F11
            {
              EnableF11 = !EnableF11;
              return;
            }
          }

          if (_enableF11) //Can use F11
          {
            if (farcilitor) //This user is farcilitor
            {
              if (((OpenSession)this.ParentForm).forAll) //select use F11 for all participant
              {
                //All participant do F11 function
                ((OpenSession)this.ParentForm).DoF11();
              }
              else
              {
                //Do F11 function
                DoF11();
              }
            }
            else //This user is participant
            {
              //Do F11 function
              DoF11();
            }
          }
        }
        catch (Exception ex)
        {
          String err = ex.Message;
        }
        break;
      #endregion

      #region "Press F12"
      case Keys.F12: //Press F12
        try
        {
          if (farcilitor)
          {
            if (e.Control) //Press Control + F12
            {
              EnableF12 = !EnableF12;
              return;
            }
          }

          if (_enableF12) //Can use F12
          {
            //Remove keyboard data
            e.SuppressKeyPress = true;
```

```
                    if ((((OpenSession)this.ParentForm).wordDoc != null) //Word Opened
                    {
                        Object oConst1 = Word.WdGoToItem.wdGoToLine;
                        Object oConst2 = Word.WdGoToDirection.wdGoToLast;
                        Object oMissing = System.Reflection.Missing.Value;

                        //Focus to last line
                        ((OpenSession)this.ParentForm).wordApp.Selection.GoTo(ref oConst1, ref
oConst2, ref oMissing, ref oMissing);

                        //Word Activate
                        ((OpenSession)this.ParentForm).wordApp.Activate();
                        DoMouseClick(900, 500);
                    }
                }
                }
                catch (Exception ex)
                {
                    String err = ex.Message;
                }
                break;
            #endregion

            #region "Press Up"
            //case Keys.Up: //Press Up
            //    //Scroll display box up
            //    if (_selectedLine > 0) _selectedLine -= 2;
            //    if (_selectedLine < 0) _selectedLine = 0;

            //    tempLength = 0;
            //    for (int i = 0; i < _selectedLine; i++)
            //    {
            //        if (i == 0) tempLength -= 2;
            //        tempLength += this.TextBoxDisplay.Lines[i].Length + 2;
            //    }

            //    this.TextBoxDisplay.Select(tempLength, 0);
            //    this.TextBoxDisplay.ScrollToCaret();

            //    //Remove keyboard data
            //    e.SuppressKeyPress = true;
            //    break;
            #endregion

            #region "Press Down"
            //case Keys.Down: //Press Down
            //    //Scroll display box down
            //    if (_selectedLine < this.TextBoxDisplay.Lines.Length - 1) _selectedLine += 2;
            //    if (_selectedLine > this.TextBoxDisplay.Lines.Length - 1) _selectedLine =
this.TextBoxDisplay.Lines.Length - 1;

            //    tempLength = 0;
            //    for (int i = 0; i <= _selectedLine; i++)
            //    {
            //        if (i == 0) tempLength += 1;
            //        tempLength += this.TextBoxDisplay.Lines[i].Length + 2;
```

```
//   }
//   this.TextBoxDisplay.Select(tempLength, 0);
//   this.TextBoxDisplay.ScrollToCaret();

//   //Remove keyboard data
//   e.SuppressKeyPress = true;
//   break;
#endregion

#region "Press PgUp"
case Keys.PageUp: //Press PgUp
  //Scroll display box up
  if (_selectedLine > 0) _selectedLine -= 2;
  if (_selectedLine < 0) _selectedLine = 0;

  tempLength = 0;
  for (int i = 0; i < _selectedLine; i++)
  {
    if (i == 0) tempLength -= 2;
    tempLength += this.TextBoxDisplay.Lines[i].Length + 2;
  }

  this.TextBoxDisplay.Select(tempLength, 0);
  this.TextBoxDisplay.ScrollToCaret();

  //Remove keyboard data
  e.SuppressKeyPress = true;
  break;
#endregion

#region "Press PgDown"
case Keys.PageDown: //Press PgDown
  //Scroll display box down
  if (_selectedLine < this.TextBoxDisplay.Lines.Length - 1) _selectedLine += 2;
  if (_selectedLine > this.TextBoxDisplay.Lines.Length - 1) _selectedLine =
this.TextBoxDisplay.Lines.Length - 1;

  tempLength = 0;
  for (int i = 0; i < _selectedLine; i++)
  {
    if (i == 0) tempLength += 1;
    tempLength += this.TextBoxDisplay.Lines[i].Length + 2;
  }
  this.TextBoxDisplay.Select(tempLength, 0);
  this.TextBoxDisplay.ScrollToCaret();

  //Remove keyboard data
  e.SuppressKeyPress = true;
  break;
#endregion

#region "Press V"
case Keys.V: //Press V
  try
  {
    if (farcilitor)
    {
```

```
            if (e.Control && e.Alt) //Press Ctrl + Alt + V
            {
              ((OpenSession)this.ParentForm).ShortCutVote();
              return;
            }
          }
        }
        catch (Exception ex)
        {
          String err = ex.Message;
        }
        break;
      #endregion

      #region "Press E"
      case Keys.E: //Press E
        try
        {
          if (farcilitor)
          {
            if (e.Control) //Press Ctrl + E
            {
              ((OpenSession)this.ParentForm).ShortCutUnfreeze();
              return;
            }
          }
        }
        catch (Exception ex)
        {
          String err = ex.Message;
        }
        break;
      #endregion

      #region "Press F"
      case Keys.F: //Press F
        try
        {
          if (farcilitor)
          {
            if (e.Control) //Press Ctrl + F
            {
              ((OpenSession)this.ParentForm).ShortCutFreeze();
              return;
            }
          }
        }
        catch (Exception ex)
        {
          String err = ex.Message;
        }
        break;
      #endregion

      #region "Press ,"
      case Keys.Oemcomma: //Press ,
        try
```

```
    {
      if (farcilitor)
      {
        if (e.Control) //Press Ctrl + ,
        {
          ((OpenSession)this.ParentForm).ShortCutPrevious();
          return;
        }
      }
    }
    catch (Exception ex)
    {
      String err = ex.Message;
    }
    break;
#endregion

#region "Press ."
case Keys.OemPeriod: //Press .
  try
  {
    if (farcilitor)
    {
      if (e.Control) //Press Ctrl + .
      {
        ((OpenSession)this.ParentForm).ShortCutNext();
        return;
      }
    }
  }
  catch (Exception ex)
  {
    String err = ex.Message;
  }
  break;
#endregion

#region "Press W"
case Keys.W: //Press W
  try
  {
    if (farcilitor)
    {
      if (e.Control && e.Alt) //Press Ctrl + Alt + W
      {
        ((OpenSession)this.ParentForm).ShortCutOpenWord();
        return;
      }
      else if (e.Control) //Press Ctrl + W
      {
        ((OpenSession)this.ParentForm).ShortCutNewWord();
        return;
      }
    }
  }
  catch (Exception ex)
  {
```

```
            String err = ex.Message;
        }
        break;
    #endregion

    #region "Press Q"
    case Keys.Q: //Press Q
        try
        {
            if (farcilitor)
            {
                if (e.Control) //Press Ctrl + Q
                {
                    ((OpenSession)this.ParentForm).ShortCutQuit();
                    return;
                }
            }
        }
        catch (Exception ex)
        {
            String err = ex.Message;
        }
        break;
    #endregion

    #region "Press Esc"
    case Keys.Escape: //Press Esc
        try
        {
            if (farcilitor)
            {
                return;
            }
            else
            {
                e.SuppressKeyPress = true;
            }
        }
        catch (Exception ex)
        {
            String err = ex.Message;
        }
        break;
    #endregion

    #region "Press Ctrl"
    case Keys.ControlKey: //Press Ctrl
        try
        {
            if (farcilitor)
            {
                return;
            }
            else
            {
                e.SuppressKeyPress = true;
            }
```

```
    }
    catch (Exception ex)
    {
       String err = ex.Message;
    }
    break;
#endregion

#region "Press Ctrl + "
case Keys.Control: //Press Ctrl
    try
    {
       if (farcilitor)
       {
          return;
       }
       else
       {
          e.SuppressKeyPress = true;
       }
    }
    catch (Exception ex)
    {
       String err = ex.Message;
    }
    break;
#endregion

#region "Press Alt"
case Keys.Alt: //Press Alt
    try
    {
       if (farcilitor)
       {
          return;
       }
       else
       {
          e.SuppressKeyPress = true;
       }
    }
    catch (Exception ex)
    {
       String err = ex.Message;
    }
    break;
#endregion

#region "Press F4"
case Keys.F4: //Press F4
    try
    {
       if (farcilitor)
       {
          return;
       }
       else
```

```
                {
                    e.SuppressKeyPress = true;
                }
            }
            catch (Exception ex)
            {
                String err = ex.Message;
            }
            break;
        #endregion
    }

    if (e.Alt == true)
    {
        if (farcilitor)
        {
            return;
        }
        else
        {
            e.SuppressKeyPress = true;
        }
    }

    Console.WriteLine(e.KeyValue.ToString());

    //For vote
    if (TextBoxInput.MaxLength == 1)
    {
        //if vote type is Yes/No, participant can press only Y or N
        if (((OpenSession)this.ParentForm).cbxVoteType.SelectedItem.ToString() == "Yes/No")
        {
            if ((e.KeyCode != Keys.Y) && (e.KeyCode != Keys.N) && (e.KeyCode != Keys.Back))
            {
                //Remove keyboard data
                e.SuppressKeyPress = true;
            }
        }
    }
}

private void GroupBoxKeyboard_EnabledChanged(object sender, EventArgs e)
{
    if (this.Enabled)
    {
        this.LabelStatus.BackColor = Color.Lime;
    }
    else
    {
        this.LabelStatus.BackColor = Color.Red;
    }
}

public bool Freeze
{
    set
    {
```

```
            _Freeze = value;
            this.GroupBoxKeyboard.Enabled = !_Freeze;
        }
        get
        {
            return _Freeze;
        }
    }

    public void SetNewTextLast()
    {
        _newText = this.TextBoxDisplay.Lines.Length;
    }

    public int SessionID
    {
        get { return _sessionID; }
        set { _sessionID = value; }
    }

    public int TemplateDetailID
    {
        get { return _templateDetailID; }
        set { _templateDetailID = value; }
    }

    private void KeyboardControl_VisibleChanged(object sender, EventArgs e)
    {
        if (this.Visible)
        {
            //Set old transaction text of this user
            if (_keyboardID != "")
            {
                //get transaction text
                DataSetBrain.SessionTransactionsDataTable STDataTable = new
DataSetBrain.SessionTransactionsDataTable();
                DataSetBrainTableAdapters.SessionTransactionsTableAdapter STTableAdapter = new
DataSetBrainTableAdapters.SessionTransactionsTableAdapter();
                STDataTable = STTableAdapter.GetDataBy_UserID(_sessionID, _keyboardID);

                foreach (DataRow myRow in STDataTable.Rows)
                {
                    //Display old transaction text
                    this.TextBoxDisplay.Text += Environment.NewLine +
myRow["TransactionInput"].ToString();
                }

                //focus on last line
                this.TextBoxDisplay.Focus();
                this.TextBoxDisplay.ScrollToCaret();
                //focus on input box
                this.TextBoxInput.Focus();
            }
        }
    }
        //EnableF11
    public bool EnableF11
```

```
{
    set
    {
        _enableF11 = value;
        if (_enableF11)
        {
            lblF11.BackColor = Color.Lime;
        }
        else
        {
            lblF11.BackColor = Color.Red;
        }
    }
    get
    {
        return _enableF11;
    }
}
        //Enable F12
public bool EnableF12
{
    set
    {
        _enableF12 = value;
        if (_enableF12)
        {
            lblF12.BackColor = Color.Lime;
        }
        else
        {
            lblF12.BackColor = Color.Red;
        }
    }
    get
    {
        return _enableF12;
    }
}

private void lblF11_Click(object sender, EventArgs e)
{
    //EnableF11 = !_enableF11;
}

private void lblF12_Click(object sender, EventArgs e)
{
    //EnableF12 = !_enableF12;
}

private void TextBoxInput_KeyUp(object sender, KeyEventArgs e)
{
    //Clear input box if it empty
    if (string.IsNullOrEmpty(TextBoxInput.Text.Trim()))
    {
        TextBoxInput.Clear();
    }
}
```

```
        private void lblF11_MouseDown(object sender, MouseEventArgs e)
        {
            EnableF11 = !_enableF11;
        }

        private void lblF12_MouseDown(object sender, MouseEventArgs e)
        {
            EnableF12 = !_enableF12;
        }

        /// <summary>
        /// Save text from input box to DB
        /// and display it on Word
        /// </summary>
        public void DoF11()
        {
            if (!string.IsNullOrEmpty(TextBoxInput.Text.Trim()))
            {
                //Goto new line
                if (this.TextBoxDisplay.Text != "") this.TextBoxDisplay.Text += Environment.NewLine;
                //display new text in display box
                this.TextBoxDisplay.Text += this.TextBoxInput.Text.ToString();
                //focus on last line
                this.TextBoxDisplay.Focus();
                this.TextBoxDisplay.ScrollToCaret();
                //Save transaction message to DB
                SaveTransaction(_sessionID, _templateDetailID, _keyboardID, TextBoxInput.Text,
DateTime.Now);

                //Clear and focus input box
                this.TextBoxInput.Clear();
                this.TextBoxInput.Focus();
            }

            if (((OpenSession)this.ParentForm).wordDoc != null) //if word opened
            {
                try
                {
                    //Close Word Doc only
                    if (((OpenSession)this.ParentForm).wordApp.Documents.Count == 0)
                    {
                        //Open Word
                        ((OpenSession)this.ParentForm).KBControlOpenWord();
                    }
                }
                catch (Exception ex)
                {
                    Console.WriteLine(ex.Message);

                    //Open Word
                    ((OpenSession)this.ParentForm).KBControlOpenWord();
                }

                Object oConst1 = Word.WdGoToItem.wdGoToLine;
                Object oConst2 = Word.WdGoToDirection.wdGoToLast;
                Object oMissing = System.Reflection.Missing.Value;
```

```
            //Focus to last line
            ((OpenSession)this.ParentForm).wordApp.Selection.GoTo(ref oConst1, ref oConst2, ref
oMissing, ref oMissing);

            if (((OpenSession)this.ParentForm).Methodology == "Anonymous") //Methodology is
Anonymous
            {
               //Display transaction text in word and not show owner's name
               for (int i = _newText; i < this.TextBoxDisplay.Lines.Length; i++)
               {

((OpenSession)this.ParentForm).wordApp.Selection.TypeText(this.TextBoxDisplay.Lines[i]);
                  ((OpenSession)this.ParentForm).wordApp.Selection.TypeText(Environment.NewLine);

                  if (((OpenSession)this.ParentForm)._vote)
                  {
                     //Set Vote Text
                     ((OpenSession)this.ParentForm)._votingText += Environment.NewLine +
this.TextBoxDisplay.Lines[i];
                  }
               }
            }
            else //Methodology is Synonymous
            {
               if (_templateDetailID != 0) //Use template
               {
                  //Display transaction text in word
                  //Display owner's name
                  for (int i = _newText; i < this.TextBoxDisplay.Lines.Length; i++)
                  {
                     ((OpenSession)this.ParentForm).wordApp.Selection.TypeText(_keyboardName + " :
" + this.TextBoxDisplay.Lines[i]);

((OpenSession)this.ParentForm).wordApp.Selection.TypeText(Environment.NewLine);

                     if (((OpenSession)this.ParentForm)._vote)
                     {
                        //Set Vote Text
                        ((OpenSession)this.ParentForm)._votingText += Environment.NewLine +
_keyboardName + " : " + this.TextBoxDisplay.Lines[i];
                     }
                  }
               }
               else //Not use template
               {
                  //Display transaction text in word and not show owner's name
                  for (int i = _newText; i < this.TextBoxDisplay.Lines.Length; i++)
                  {

((OpenSession)this.ParentForm).wordApp.Selection.TypeText(this.TextBoxDisplay.Lines[i]);

((OpenSession)this.ParentForm).wordApp.Selection.TypeText(Environment.NewLine);

                     if (((OpenSession)this.ParentForm)._vote)
                     {
                        //Set Vote Text
```

```
                    ((OpenSession)this.ParentForm)._votingText += Environment.NewLine +
this.TextBoxDisplay.Lines[i];
                }
            }
        }
    }
    else //word not open
    {
        //Open Word
        ((OpenSession)this.ParentForm).KBControlOpenWord();
    }

    // Save Last Old Text Index
    SetNewTextLast();
}

/// <summary>
/// Save transaction text to DB
/// </summary>
/// <param name="SessionID"></param>
/// <param name="TemplateDetailID"></param>
/// <param name="UserID"></param>
/// <param name="Message"></param>
/// <param name="Date"></param>
private void SaveTransaction(int SessionID, int TemplateDetailID, string UserID, string Message,
DateTime Date)
{
    //Save transaction text to DB
    DataSetBrainTableAdapters.SessionTransactionsTableAdapter STTableAdapter = new
DataSetBrainTableAdapters.SessionTransactionsTableAdapter();
    STTableAdapter.Insert(SessionID, TemplateDetailID, UserID, Message, Date);

    //Set Session status to pending
    if (_sessionStatus == "New")
    {
        DataSetBrainTableAdapters.SessionsTableAdapter sessionsTableAdapter = new
DataSetBrainTableAdapters.SessionsTableAdapter();
        sessionsTableAdapter.Update_Status("Pending", SessionID);

        _sessionStatus = "Pending";
    }
}

/// <summary>
/// Change language input
/// </summary>
public void ChangeLanguage()
{
    try
    {
        for (int i = 0; i < InputLanguage.InstalledInputLanguages.Count; i++)
        {
            if (this.LanguageCulture ==
InputLanguage.InstalledInputLanguages[i].Culture.ToString())
            {
                if (i == (InputLanguage.InstalledInputLanguages.Count - 1))
```

```
                {
                    i = 0;
                }
                else
                {
                    i++;
                }

                this.LanguageCulture = InputLanguage.InstalledInputLanguages[i].Culture.ToString();
                InputLanguage.CurrentInputLanguage = InputLanguage.InstalledInputLanguages[i];

                return;
            }
        }
    }
    catch (Exception ex)
    {
        String err = ex.Message;
    }
  }
 }
}
```

KeyboardControl.Designer.cs

```
namespace GroupStorm
{
  partial class KeyboardControl
  {
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
    protected override void Dispose(bool disposing)
    {
      if (disposing && (components != null))
      {
        components.Dispose();
      }
      base.Dispose(disposing);
    }

    #region Component Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
```

```
{
    this.components = new System.ComponentModel.Container();
    this.GroupBoxKeyboard = new System.Windows.Forms.GroupBox();
    this.lblF12 = new System.Windows.Forms.Label();
    this.lblF11 = new System.Windows.Forms.Label();
    this.LabelStatus = new System.Windows.Forms.Label();
    this.LabelLanguage = new System.Windows.Forms.Label();
    this.TextBoxInput = new System.Windows.Forms.TextBox();
    this.contextMenuStrip1 = new System.Windows.Forms.ContextMenuStrip(this.components);
    this.copyToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.pasteToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.pasteAllToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
    this.TextBoxDisplay = new System.Windows.Forms.TextBox();
    this.GroupBoxKeyboard.SuspendLayout();
    this.contextMenuStrip1.SuspendLayout();
    this.SuspendLayout();
    //
    // GroupBoxKeyboard
    //
    this.GroupBoxKeyboard.Controls.Add(this.lblF12);
    this.GroupBoxKeyboard.Controls.Add(this.lblF11);
    this.GroupBoxKeyboard.Controls.Add(this.LabelStatus);
    this.GroupBoxKeyboard.Controls.Add(this.LabelLanguage);
    this.GroupBoxKeyboard.Controls.Add(this.TextBoxInput);
    this.GroupBoxKeyboard.Controls.Add(this.TextBoxDisplay);
    this.GroupBoxKeyboard.Dock = System.Windows.Forms.DockStyle.Fill;
    this.GroupBoxKeyboard.Location = new System.Drawing.Point(0, 0);
    this.GroupBoxKeyboard.Name = "GroupBoxKeyboard";
    this.GroupBoxKeyboard.Padding = new System.Windows.Forms.Padding(5, 3, 5, 3);
    this.GroupBoxKeyboard.Size = new System.Drawing.Size(300, 90);
    this.GroupBoxKeyboard.TabIndex = 0;
    this.GroupBoxKeyboard.TabStop = false;
    this.GroupBoxKeyboard.Text = "   Keyboard";
    this.GroupBoxKeyboard.EnabledChanged += new
System.EventHandler(this.GroupBoxKeyboard_EnabledChanged);
    //
    // lblF12
    //
    this.lblF12.AutoSize = true;
    this.lblF12.BackColor = System.Drawing.Color.Red;
    this.lblF12.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
    this.lblF12.Font = new System.Drawing.Font("Microsoft Sans Serif", 6.5F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
    this.lblF12.ForeColor = System.Drawing.Color.Black;
    this.lblF12.Location = new System.Drawing.Point(246, 1);
    this.lblF12.Name = "lblF12";
    this.lblF12.Size = new System.Drawing.Size(26, 14);
    this.lblF12.TabIndex = 5;
    this.lblF12.Text = "F12";
    this.lblF12.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
    this.lblF12.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.lblF12_MouseDown);
    //
    // lblF11
    //
    this.lblF11.AutoSize = true;
    this.lblF11.BackColor = System.Drawing.Color.Red;
```

```
        this.lblF11.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
        this.lblF11.Font = new System.Drawing.Font("Microsoft Sans Serif", 6.5F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
        this.lblF11.ForeColor = System.Drawing.Color.Black;
        this.lblF11.Location = new System.Drawing.Point(219, 1);
        this.lblF11.Name = "lblF11";
        this.lblF11.Size = new System.Drawing.Size(26, 14);
        this.lblF11.TabIndex = 4;
        this.lblF11.Text = "F11";
        this.lblF11.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
        this.lblF11.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.lblF11_MouseDown);
        //
        // LabelStatus
        //
        this.LabelStatus.BackColor = System.Drawing.Color.Lime;
        this.LabelStatus.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
        this.LabelStatus.Location = new System.Drawing.Point(5, 2);
        this.LabelStatus.Name = "LabelStatus";
        this.LabelStatus.Size = new System.Drawing.Size(10, 10);
        this.LabelStatus.TabIndex = 3;
        //
        // LabelLanguage
        //
        this.LabelLanguage.BackColor = System.Drawing.SystemColors.GradientInactiveCaption;
        this.LabelLanguage.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
        this.LabelLanguage.Font = new System.Drawing.Font("Microsoft Sans Serif", 6.5F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
        this.LabelLanguage.ForeColor = System.Drawing.Color.Black;
        this.LabelLanguage.Location = new System.Drawing.Point(273, 1);
        this.LabelLanguage.Name = "LabelLanguage";
        this.LabelLanguage.Size = new System.Drawing.Size(22, 14);
        this.LabelLanguage.TabIndex = 2;
        this.LabelLanguage.Text = "EN";
        this.LabelLanguage.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
        //
        // TextBoxInput
        //
        this.TextBoxInput.ContextMenuStrip = this.contextMenuStrip1;
        this.TextBoxInput.Location = new System.Drawing.Point(5, 52);
        this.TextBoxInput.Multiline = true;
        this.TextBoxInput.Name = "TextBoxInput";
        this.TextBoxInput.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
        this.TextBoxInput.Size = new System.Drawing.Size(290, 33);
        this.TextBoxInput.TabIndex = 1;
        this.TextBoxInput.Tag = "";
        this.TextBoxInput.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.TextBoxInput_KeyDown);
        this.TextBoxInput.KeyUp += new
System.Windows.Forms.KeyEventHandler(this.TextBoxInput_KeyUp);
        //
        // contextMenuStrip1
        //
        this.contextMenuStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
        this.copyToolStripMenuItem,
        this.pasteToolStripMenuItem,
        this.pasteAllToolStripMenuItem});
```

```
            this.contextMenuStrip1.Name = "contextMenuStrip1";
            this.contextMenuStrip1.Size = new System.Drawing.Size(116, 70);
            //
            // copyToolStripMenuItem
            //
            this.copyToolStripMenuItem.Name = "copyToolStripMenuItem";
            this.copyToolStripMenuItem.Size = new System.Drawing.Size(115, 22);
            this.copyToolStripMenuItem.Text = "Copy";
            this.copyToolStripMenuItem.Click += new
System.EventHandler(this.copyToolStripMenuItem_Click);
            //
            // pasteToolStripMenuItem
            //
            this.pasteToolStripMenuItem.Name = "pasteToolStripMenuItem";
            this.pasteToolStripMenuItem.Size = new System.Drawing.Size(115, 22);
            this.pasteToolStripMenuItem.Text = "Paste";
            this.pasteToolStripMenuItem.Click += new
System.EventHandler(this.pasteToolStripMenuItem_Click);
            //
            // pasteAllToolStripMenuItem
            //
            this.pasteAllToolStripMenuItem.Name = "pasteAllToolStripMenuItem";
            this.pasteAllToolStripMenuItem.Size = new System.Drawing.Size(115, 22);
            this.pasteAllToolStripMenuItem.Text = "Paste All";
            this.pasteAllToolStripMenuItem.Click += new
System.EventHandler(this.pasteAllToolStripMenuItem_Click);
            //
            // TextBoxDisplay
            //
            this.TextBoxDisplay.AllowDrop = true;
            this.TextBoxDisplay.BackColor = System.Drawing.SystemColors.Info;
            this.TextBoxDisplay.ContextMenuStrip = this.contextMenuStrip1;
            this.TextBoxDisplay.Dock = System.Windows.Forms.DockStyle.Top;
            this.TextBoxDisplay.Location = new System.Drawing.Point(5, 16);
            this.TextBoxDisplay.Multiline = true;
            this.TextBoxDisplay.Name = "TextBoxDisplay";
            this.TextBoxDisplay.ReadOnly = true;
            this.TextBoxDisplay.ScrollBars = System.Windows.Forms.ScrollBars.Vertical;
            this.TextBoxDisplay.Size = new System.Drawing.Size(290, 38);
            this.TextBoxDisplay.TabIndex = 0;
            //
            // KeyboardControl
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ContextMenuStrip = this.contextMenuStrip1;
            this.Controls.Add(this.GroupBoxKeyboard);
            this.Name = "KeyboardControl";
            this.Size = new System.Drawing.Size(300, 90);
            this.Load += new System.EventHandler(this.KeyboardControl_Load);
            this.VisibleChanged += new System.EventHandler(this.KeyboardControl_VisibleChanged);
            this.GroupBoxKeyboard.ResumeLayout(false);
            this.GroupBoxKeyboard.PerformLayout();
            this.contextMenuStrip1.ResumeLayout(false);
            this.ResumeLayout(false);

        }
```

```
        #endregion

        private System.Windows.Forms.GroupBox GroupBoxKeyboard;
        private System.Windows.Forms.Label LabelLanguage;
        public System.Windows.Forms.TextBox TextBoxInput;
        private System.Windows.Forms.ContextMenuStrip contextMenuStrip1;
        private System.Windows.Forms.ToolStripMenuItem copyToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem pasteToolStripMenuItem;
        private System.Windows.Forms.Label LabelStatus;
        public System.Windows.Forms.TextBox TextBoxDisplay;
        private System.Windows.Forms.ToolStripMenuItem pasteAllToolStripMenuItem;
        private System.Windows.Forms.Label lblF12;
        private System.Windows.Forms.Label lblF11;
    }
}
```

LogInControl.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Data;
using System.Text;
using System.Windows.Forms;

namespace GroupStorm
{
    public partial class LogInControl : UserControl
    {
        private string _keyboardName = "";
        private string _languageCulture = "";
        private string _userID = "";
        private string _name = "";
        private string _lastname = "";
        private string _type = "";
        private string _methodology = "Synonymous";
        private KeyboardControl _keyboardControl;
        private bool _focusUserID = true;

        public LogInControl()
        {
            InitializeComponent();
        }

        public string KeyboardName
        {
            set
            {
                _keyboardName = value;
                this.groupBox1.Text = value;
            }
```

```
    get
    {
      return _keyboardName;
    }
  }

  public string LanguageCulture
  {
    set
    {
      _languageCulture = value;
      if (!string.IsNullOrEmpty(value))
      {
        this.LabelLanguage.Text = value.Substring(0, 2).ToUpper();
      }
    }
    get
    {
      return _languageCulture;
    }
  }

  private void LogInControl_Load(object sender, EventArgs e)
  {
    //Set Default Input Language
    _languageCulture = InputLanguage.CurrentInputLanguage.Culture.ToString();

    //Focus to userid box
    this.TextBoxUserID.Focus();
  }

  private void LogInControl_EnabledChanged(object sender, EventArgs e)
  {
    if (this.Enabled)
    {
      this.LabelStatus.BackColor = Color.Lime;
    }
    else
    {
      this.LabelStatus.BackColor = Color.Red;
    }
  }

  /// <summary>
  /// Keydown on userid box
  /// </summary>
  private void TextBoxUserID_KeyDown(object sender, KeyEventArgs e)
  {
    if (e.KeyCode == Keys.Enter) //Press 'Enter' key
    {
      //focus on password box
      _focusUserID = false;
      this.TextBoxPassword.Focus();
    }
  }

  /// <summary>
```

```
        /// Keydown on password box
        /// </summary>
        private void TextBoxPassword_KeyDown(object sender, KeyEventArgs e)
        {
            if (e.KeyData == Keys.Enter) //Press Enter
            {
                // Check Login
                DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
                DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();

                usersDataTable = usersTableAdapter.GetDataBy_LogIn(this.TextBoxUserID.Text.Trim(),
this.TextBoxPassword.Text.Trim(), ((OpenSession)this.ParentForm).SessionID);

                if (usersDataTable.Rows.Count > 0)
                {
                    string LogOn = usersDataTable.Rows[0]["LogOn"].ToString();
                    if (LogOn != "Y")
                    {
                        _userID = usersDataTable.Rows[0]["UserID"].ToString();
                        _name = usersDataTable.Rows[0]["Name"].ToString();
                        _lastname = usersDataTable.Rows[0]["LastName"].ToString();
                        _type = usersDataTable.Rows[0]["Type"].ToString();

                        //if Methodology is Synonymous, display farcilitor or participant's name
                        if (_methodology == "Synonymous")
                        {
                            _keyboardControl.KeyboardName = _name + " " + _lastname;
                        }

                        //Set userID to keyboardControl
                        _keyboardControl.KeyboardID = _userID;

                        //Trick event VisibleChanged of keyboardControl
                        _keyboardControl.Visible = false;
                        _keyboardControl.Visible = true;

                        //hide login form
                        this.Visible = false;

                        //Set Status LogOn
                        usersTableAdapter.SetLogOn("Y", _userID);
                    }
                    else
                    {
                        //Logon error, clear all textbox
                        this.LabelError.Visible = true;
                        this.TextBoxUserID.Text = "";
                        this.TextBoxPassword.Text = "";
                        this.TextBoxUserID.Focus();
                        _focusUserID = true;
                    }
                }
                else
                {
                    //Logon error, clear all textbox
                    this.LabelError.Visible = true;
```

```
                this.TextBoxUserID.Text = "";
                this.TextBoxPassword.Text = "";
                this.TextBoxUserID.Focus();
                _focusUserID = true;
            }
        }
    }

    public string Methodology
    {
        get { return _methodology; }
        set { _methodology = value; }
    }

    public KeyboardControl KeyboardControl
    {
        get { return _keyboardControl; }
        set { _keyboardControl = value; }
    }

    public bool FocusUserID
    {
        get { return _focusUserID; }
        set { _focusUserID = value; }
    }

    /// <summary>
    /// Change language input
    /// </summary>
    public void ChangeLanguage()
    {
        try
        {
            for (int i = 0; i < InputLanguage.InstalledInputLanguages.Count; i++)
            {
                if (this.LanguageCulture ==
InputLanguage.InstalledInputLanguages[i].Culture.ToString())
                {
                    if (i == (InputLanguage.InstalledInputLanguages.Count - 1))
                    {
                        i = 0;
                    }
                    else
                    {
                        i++;
                    }

                    this.LanguageCulture = InputLanguage.InstalledInputLanguages[i].Culture.ToString();
                    InputLanguage.CurrentInputLanguage = InputLanguage.InstalledInputLanguages[i];

                    return;
                }
            }
        }
        catch (Exception ex)
        {
            String err = ex.Message;
```

```
        }
      }
    }
}
```

LogInControl.Designer.cs

```
namespace GroupStorm
{
  partial class LogInControl
  {
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
    protected override void Dispose(bool disposing)
    {
      if (disposing && (components != null))
      {
        components.Dispose();
      }
      base.Dispose(disposing);
    }

    #region Component Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.groupBox1 = new System.Windows.Forms.GroupBox();
      this.LabelError = new System.Windows.Forms.Label();
      this.label2 = new System.Windows.Forms.Label();
      this.label1 = new System.Windows.Forms.Label();
      this.TextBoxPassword = new System.Windows.Forms.TextBox();
      this.TextBoxUserID = new System.Windows.Forms.TextBox();
      this.LabelStatus = new System.Windows.Forms.Label();
      this.LabelLanguage = new System.Windows.Forms.Label();
      this.groupBox1.SuspendLayout();
      this.SuspendLayout();
      //
      // groupBox1
      //
      this.groupBox1.Controls.Add(this.LabelError);
      this.groupBox1.Controls.Add(this.label2);
      this.groupBox1.Controls.Add(this.label1);
      this.groupBox1.Controls.Add(this.TextBoxPassword);
```

```
            this.groupBox1.Controls.Add(this.TextBoxUserID);
            this.groupBox1.Controls.Add(this.LabelStatus);
            this.groupBox1.Controls.Add(this.LabelLanguage);
            this.groupBox1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.groupBox1.Location = new System.Drawing.Point(0, 0);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Padding = new System.Windows.Forms.Padding(5, 3, 5, 3);
            this.groupBox1.Size = new System.Drawing.Size(300, 90);
            this.groupBox1.TabIndex = 0;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Keyboard";
            //
            // LabelError
            //
            this.LabelError.AutoSize = true;
            this.LabelError.ForeColor = System.Drawing.Color.Red;
            this.LabelError.Location = new System.Drawing.Point(85, 72);
            this.LabelError.Name = "LabelError";
            this.LabelError.Size = new System.Drawing.Size(141, 13);
            this.LabelError.TabIndex = 6;
            this.LabelError.Text = "Login error. Please try again.";
            this.LabelError.Visible = false;
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(16, 48);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(53, 13);
            this.label2.TabIndex = 3;
            this.label2.Text = "Password";
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(26, 22);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(43, 13);
            this.label1.TabIndex = 2;
            this.label1.Text = "User ID";
            //
            // TextBoxPassword
            //
            this.TextBoxPassword.BackColor = System.Drawing.SystemColors.Window;
            this.TextBoxPassword.Location = new System.Drawing.Point(75, 45);
            this.TextBoxPassword.Name = "TextBoxPassword";
            this.TextBoxPassword.PasswordChar = '*';
            this.TextBoxPassword.Size = new System.Drawing.Size(209, 20);
            this.TextBoxPassword.TabIndex = 1;
            this.TextBoxPassword.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.TextBoxPassword_KeyDown);
            //
            // TextBoxUserID
            //
            this.TextBoxUserID.BackColor = System.Drawing.SystemColors.Window;
            this.TextBoxUserID.Location = new System.Drawing.Point(75, 19);
            this.TextBoxUserID.Name = "TextBoxUserID";
```

```
            this.TextBoxUserID.Size = new System.Drawing.Size(209, 20);
            this.TextBoxUserID.TabIndex = 0;
            this.TextBoxUserID.KeyDown += new
    System.Windows.Forms.KeyEventHandler(this.TextBoxUserID_KeyDown);
            //
            // LabelStatus
            //
            this.LabelStatus.BackColor = System.Drawing.Color.Lime;
            this.LabelStatus.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
            this.LabelStatus.Location = new System.Drawing.Point(284, 74);
            this.LabelStatus.Name = "LabelStatus";
            this.LabelStatus.Size = new System.Drawing.Size(13, 13);
            this.LabelStatus.TabIndex = 5;
            //
            // LabelLanguage
            //
            this.LabelLanguage.AutoSize = true;
            this.LabelLanguage.BackColor = System.Drawing.SystemColors.GradientInactiveCaption;
            this.LabelLanguage.BorderStyle = System.Windows.Forms.BorderStyle.FixedSingle;
            this.LabelLanguage.Dock = System.Windows.Forms.DockStyle.Bottom;
            this.LabelLanguage.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
    System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.LabelLanguage.ForeColor = System.Drawing.Color.Black;
            this.LabelLanguage.Location = new System.Drawing.Point(5, 72);
            this.LabelLanguage.Name = "LabelLanguage";
            this.LabelLanguage.Size = new System.Drawing.Size(26, 15);
            this.LabelLanguage.TabIndex = 4;
            this.LabelLanguage.Text = "EN";
            this.LabelLanguage.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
            //
            // LogInControl
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.Controls.Add(this.groupBox1);
            this.Name = "LogInControl";
            this.Size = new System.Drawing.Size(300, 90);
            this.Load += new System.EventHandler(this.LogInControl_Load);
            this.EnabledChanged += new System.EventHandler(this.LogInControl_EnabledChanged);
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Label LabelStatus;
        private System.Windows.Forms.Label LabelLanguage;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label LabelError;
        public System.Windows.Forms.TextBox TextBoxUserID;
        public System.Windows.Forms.TextBox TextBoxPassword;
    }
}
```

UserSetup.cs

```
sing System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace GroupStorm
{
   public partial class UserSetup : Form
   {
      private DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
      private DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
GroupStorm.DataSetBrainTableAdapters.UsersTableAdapter();
      private string _UserID = "";

      public UserSetup()
      {
         InitializeComponent();
      }

      private void UserSetup_Load(object sender, EventArgs e)
      {
         //Set User Type Combo
         this.ComboBoxType.SelectedIndex = 0;

         //Display User List
         RefreshData();
      }

      /// <summary>
      /// Display selected user detail
      /// </summary>
      private void DataGridViewUser_CellClick(object sender, DataGridViewCellEventArgs e)
      {
         if (e.RowIndex != -1) //Not Header
         {
            //Set User Detail
            _UserID = this.DataGridViewUser[0, e.RowIndex].Value.ToString();
            this.TextBoxUserID.Text = this.DataGridViewUser[0, e.RowIndex].Value.ToString();
            this.TextBoxNewUserID.Text = this.DataGridViewUser[0, e.RowIndex].Value.ToString();
            this.TextBoxName.Text = this.DataGridViewUser[1, e.RowIndex].Value.ToString();
            this.TextBoxLastname.Text = this.DataGridViewUser[2, e.RowIndex].Value.ToString();
            this.TextBoxPassword.Text = this.DataGridViewUser[3, e.RowIndex].Value.ToString();
            this.ComboBoxType.SelectedItem = this.DataGridViewUser[4,
e.RowIndex].Value.ToString();
         }
      }
```

```
        /// <summary>
        /// Delete User
        /// </summary>
        private void DataGridViewUser_CellContentClick(object sender, DataGridViewCellEventArgs e)
        {
          try
          {
            if ((e.RowIndex != -1) && (e.ColumnIndex == 5)) //Click Delete button
            {
              if (MessageBox.Show("Do you want to delete?", "Delete User",
MessageBoxButtons.YesNo) == DialogResult.Yes) //Confirm Dateta
              {
                DataSetBrainTableAdapters.SessionParticipantsTableAdapter SPTableAdapter = new
GroupStorm.DataSetBrainTableAdapters.SessionParticipantsTableAdapter();
                if (SPTableAdapter.GetUserInSession(_UserID) > 0)
                {
                  MessageBox.Show("The User ID has been added into a session. Cannot delete.");
                }
                else
                {
                  //Delete User
                  usersTableAdapter.Delete(_UserID);
                  MessageBox.Show("Data was successfully deleted.");

                  //Clear User Detail
                  _UserID = "";
                  this.TextBoxUserID.Text = "";
                  this.TextBoxNewUserID.Text = "";
                  this.TextBoxName.Text = "";
                  this.TextBoxLastname.Text = "";
                  this.TextBoxPassword.Text = "";
                  this.ComboBoxType.SelectedIndex = 0;

                  //Display User List
                  RefreshData();
                }
              }
            }
          }
          catch (Exception ex)
          {
            MessageBox.Show(ex.Message);
          }
        }

        /// <summary>
        /// Add New User
        /// </summary>
        private void ButtonCreate_Click(object sender, EventArgs e)
        {
          //Add New User
          try
          {
            //Get User Data
            string UserID = this.TextBoxNewUserID.Text.Trim();
            string Name = this.TextBoxName.Text.Trim();
            string Lastname = this.TextBoxLastname.Text.Trim();
```

```
            string Password = this.TextBoxPassword.Text.Trim();
            string Type = this.ComboBoxType.SelectedItem.ToString();

            if (CheckSave(true)) //Varidate Data
            {
              if (CheckDup(UserID)) //Dupplicate UserID
              {
                //Display Error
                this.errorProvider1.SetError(this.TextBoxNewUserID, "Dupplicate UserID. Please enter
new \"UserID\".");
              }
              else
              {
                //Add New User
                usersTableAdapter.Insert(UserID, Name, Lastname, Password, Type, "");

                MessageBox.Show("Data was successfully saved.");

                //Clear User Detail
                _UserID = "";
                this.TextBoxUserID.Text = "";
                this.TextBoxNewUserID.Text = "";
                this.TextBoxName.Text = "";
                this.TextBoxLastname.Text = "";
                this.TextBoxPassword.Text = "";
                this.ComboBoxType.SelectedIndex = 0;

                //Display User List
                RefreshData();

                //Clear Error
                this.errorProvider1.Clear();
              }
            }
          }
          catch (Exception ex)
          {
            MessageBox.Show(ex.Message);
          }
        }

        /// <summary>
        /// Update User
        /// </summary>
        private void ButtonUpdate_Click(object sender, EventArgs e)
        {
          //Update User
          try
          {
            //Get User Data
            string Name = this.TextBoxName.Text.Trim();
            string Lastname = this.TextBoxLastname.Text.Trim();
            string Password = this.TextBoxPassword.Text.Trim();
            string Type = this.ComboBoxType.SelectedItem.ToString();

            if (CheckSave(false)) //Varidate Data
            {
```

```
            DataSetBrainTableAdapters.SessionParticipantsTableAdapter SPTableAdapter = new
GroupStorm.DataSetBrainTableAdapters.SessionParticipantsTableAdapter();
            if (SPTableAdapter.GetUserInSession(_UserID) > 0) //User in use
            {
              if (Type == DataGridViewUser.CurrentRow.Cells[4].Value.ToString()) //User Type not
change
              {
                //Update User
                usersTableAdapter.Update(_UserID, Name, Lastname, Password, Type, _UserID);
              }
              else
              {
                MessageBox.Show("The User ID has been added into a session. Type cannot be
changed.");
                return;
              }
            }
            else
            {
              //Update User
              usersTableAdapter.Update(_UserID, Name, Lastname, Password, Type, _UserID);
            }

            MessageBox.Show("Data was successfully saved.");

            //Clear User Detail
            _UserID = "";
            this.TextBoxUserID.Text = "";
            this.TextBoxNewUserID.Text = "";
            this.TextBoxName.Text = "";
            this.TextBoxLastname.Text = "";
            this.TextBoxPassword.Text = "";
            this.ComboBoxType.SelectedIndex = 0;

            //Display User List
            RefreshData();

            //Clear Error
            this.errorProvider1.Clear();
          }
        }
        catch (Exception ex)
        {
          MessageBox.Show(ex.Message);
        }
      }

      /// <summary>
      /// Varidate Data
      /// </summary>
      private bool CheckSave(bool NewUser)
      {
        bool result = true;

        if (NewUser) //New User
        {
          if (string.IsNullOrEmpty(this.TextBoxNewUserID.Text.Trim())) //Empty UserID
```

```
        {
            //Display Error
            this.errorProvider1.SetError(this.TextBoxNewUserID, "Please enter \"UserID\".");
            result = false;
        }
    }
    if (string.IsNullOrEmpty(this.TextBoxName.Text.Trim())) //Empty Name
    {
        //Display Error
        this.errorProvider1.SetError(this.TextBoxName, "Please enter \"Name\".");
        result = false;
    }
    if (string.IsNullOrEmpty(this.TextBoxLastname.Text.Trim())) // Empty Lastname
    {
        //Display Error
        this.errorProvider1.SetError(this.TextBoxLastname, "Please enter \"Lastname\".");
        result = false;
    }
    if (string.IsNullOrEmpty(this.TextBoxPassword.Text.Trim())) //Empty Password
    {
        //Display Error
        this.errorProvider1.SetError(this.TextBoxPassword, "Please enter \"Password\".");
        result = false;
    }
    if (string.IsNullOrEmpty(this.ComboBoxType.SelectedItem.ToString())) //Empty user type
    {
        //Display Error
        this.errorProvider1.SetError(this.ComboBoxType, "Please select \"Type\".");
        result = false;
    }

    return result;
}

/// <summary>
/// Check Dupplicate UserID
/// </summary>
private bool CheckDup(string UserID)
{
    bool result = false;

    if (usersTableAdapter.CheckDup(UserID) > 0)
    {
        result = true;
    }

    return result;
}

/// <summary>
/// Display User List
/// </summary>
private void RefreshData()
{
    usersDataTable = usersTableAdapter.GetData();
    this.DataGridViewUser.AutoGenerateColumns = false;
    this.DataGridViewUser.DataSource = usersDataTable;
```

```
        }
    }
}
```

UserSetup.Designer.cs

```
namespace GroupStorm
{
    partial class UserSetup
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            this.DataGridViewUser = new System.Windows.Forms.DataGridView();
            this.UserIDTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.NameTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.LastnameTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.PasswordTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
            this.TypeTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.DeleteButtonColumn = new System.Windows.Forms.DataGridViewButtonColumn();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.label8 = new System.Windows.Forms.Label();
            this.label7 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
            this.TextBoxUserID = new System.Windows.Forms.TextBox();
            this.ButtonUpdate = new System.Windows.Forms.Button();
```

```
            this.ButtonCreate = new System.Windows.Forms.Button();
            this.label5 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.TextBoxNewUserID = new System.Windows.Forms.TextBox();
            this.ComboBoxType = new System.Windows.Forms.ComboBox();
            this.TextBoxName = new System.Windows.Forms.TextBox();
            this.TextBoxPassword = new System.Windows.Forms.TextBox();
            this.TextBoxLastname = new System.Windows.Forms.TextBox();
            this.groupBox3 = new System.Windows.Forms.GroupBox();
            this.errorProvider1 = new System.Windows.Forms.ErrorProvider(this.components);
            ((System.ComponentModel.ISupportInitialize)(this.DataGridViewUser)).BeginInit();
            this.groupBox2.SuspendLayout();
            this.groupBox3.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.errorProvider1)).BeginInit();
            this.SuspendLayout();
            //
            // DataGridViewUser
            //
            this.DataGridViewUser.AllowUserToAddRows = false;
            this.DataGridViewUser.AllowUserToDeleteRows = false;
            this.DataGridViewUser.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            this.DataGridViewUser.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
            this.UserIDTextBoxColumn,
            this.NameTextBoxColumn,
            this.LastnameTextBoxColumn,
            this.PasswordTextBoxColumn,
            this.TypeTextBoxColumn,
            this.DeleteButtonColumn});
            this.DataGridViewUser.Dock = System.Windows.Forms.DockStyle.Fill;
            this.DataGridViewUser.Location = new System.Drawing.Point(3, 16);
            this.DataGridViewUser.MultiSelect = false;
            this.DataGridViewUser.Name = "DataGridViewUser";
            this.DataGridViewUser.ReadOnly = true;
            this.DataGridViewUser.RowHeadersVisible = false;
            this.DataGridViewUser.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
            this.DataGridViewUser.Size = new System.Drawing.Size(705, 424);
            this.DataGridViewUser.TabIndex = 0;
            this.DataGridViewUser.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.DataGridViewUser_CellClick);
            this.DataGridViewUser.CellContentClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.DataGridViewUser_CellContentClick);
            //
            // UserIDTextBoxColumn
            //
            this.UserIDTextBoxColumn.DataPropertyName = "UserID";
            this.UserIDTextBoxColumn.HeaderText = "User ID";
            this.UserIDTextBoxColumn.Name = "UserIDTextBoxColumn";
            this.UserIDTextBoxColumn.ReadOnly = true;
            this.UserIDTextBoxColumn.Width = 68;
            //
            // NameTextBoxColumn
```

```
            //
            this.NameTextBoxColumn.DataPropertyName = "Name";
            this.NameTextBoxColumn.HeaderText = "Name";
            this.NameTextBoxColumn.Name = "NameTextBoxColumn";
            this.NameTextBoxColumn.ReadOnly = true;
            //
            // LastnameTextBoxColumn
            //
            this.LastnameTextBoxColumn.DataPropertyName = "Lastname";
            this.LastnameTextBoxColumn.HeaderText = "Lastname";
            this.LastnameTextBoxColumn.Name = "LastnameTextBoxColumn";
            this.LastnameTextBoxColumn.ReadOnly = true;
            //
            // PasswordTextBoxColumn
            //
            this.PasswordTextBoxColumn.DataPropertyName = "Password";
            this.PasswordTextBoxColumn.HeaderText = "Password";
            this.PasswordTextBoxColumn.Name = "PasswordTextBoxColumn";
            this.PasswordTextBoxColumn.ReadOnly = true;
            //
            // TypeTextBoxColumn
            //
            this.TypeTextBoxColumn.DataPropertyName = "Type";
            this.TypeTextBoxColumn.HeaderText = "Type";
            this.TypeTextBoxColumn.Name = "TypeTextBoxColumn";
            this.TypeTextBoxColumn.ReadOnly = true;
            //
            // DeleteButtonColumn
            //
            this.DeleteButtonColumn.AutoSizeMode =
System.Windows.Forms.DataGridViewAutoSizeColumnMode.DisplayedCells;
            this.DeleteButtonColumn.HeaderText = "Delete";
            this.DeleteButtonColumn.Name = "DeleteButtonColumn";
            this.DeleteButtonColumn.ReadOnly = true;
            this.DeleteButtonColumn.Text = "Delete";
            this.DeleteButtonColumn.UseColumnTextForButtonValue = true;
            this.DeleteButtonColumn.Width = 44;
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.label8);
            this.groupBox2.Controls.Add(this.label7);
            this.groupBox2.Controls.Add(this.label6);
            this.groupBox2.Controls.Add(this.TextBoxUserID);
            this.groupBox2.Controls.Add(this.ButtonUpdate);
            this.groupBox2.Controls.Add(this.ButtonCreate);
            this.groupBox2.Controls.Add(this.label5);
            this.groupBox2.Controls.Add(this.label4);
            this.groupBox2.Controls.Add(this.label3);
            this.groupBox2.Controls.Add(this.label2);
            this.groupBox2.Controls.Add(this.label1);
            this.groupBox2.Controls.Add(this.TextBoxNewUserID);
            this.groupBox2.Controls.Add(this.ComboBoxType);
            this.groupBox2.Controls.Add(this.TextBoxName);
            this.groupBox2.Controls.Add(this.TextBoxPassword);
            this.groupBox2.Controls.Add(this.TextBoxLastname);
            this.groupBox2.Dock = System.Windows.Forms.DockStyle.Bottom;
```

```
this.groupBox2.Location = new System.Drawing.Point(5, 448);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(711, 225);
this.groupBox2.TabIndex = 0;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "User Setup";
//
// label8
//
this.label8.AutoSize = true;
this.label8.ForeColor = System.Drawing.Color.Red;
this.label8.Location = new System.Drawing.Point(432, 49);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(58, 13);
this.label8.TabIndex = 25;
this.label8.Text = "For create.";
//
// label7
//
this.label7.AutoSize = true;
this.label7.ForeColor = System.Drawing.Color.Red;
this.label7.Location = new System.Drawing.Point(432, 22);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(61, 13);
this.label7.TabIndex = 24;
this.label7.Text = "For update.";
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(197, 22);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(43, 13);
this.label6.TabIndex = 23;
this.label6.Text = "User ID";
//
// TextBoxUserID
//
this.TextBoxUserID.AcceptsReturn = true;
this.TextBoxUserID.Location = new System.Drawing.Point(246, 19);
this.TextBoxUserID.MaxLength = 50;
this.TextBoxUserID.Name = "TextBoxUserID";
this.TextBoxUserID.ReadOnly = true;
this.TextBoxUserID.Size = new System.Drawing.Size(180, 20);
this.TextBoxUserID.TabIndex = 22;
//
// ButtonUpdate
//
this.ButtonUpdate.AutoSize = true;
this.ButtonUpdate.Location = new System.Drawing.Point(345, 182);
this.ButtonUpdate.Name = "ButtonUpdate";
this.ButtonUpdate.Size = new System.Drawing.Size(110, 23);
this.ButtonUpdate.TabIndex = 21;
this.ButtonUpdate.Text = "Update User";
this.ButtonUpdate.UseVisualStyleBackColor = true;
this.ButtonUpdate.Click += new System.EventHandler(this.ButtonUpdate_Click);
//
```

```
// ButtonCreate
//
this.ButtonCreate.Location = new System.Drawing.Point(163, 182);
this.ButtonCreate.Name = "ButtonCreate";
this.ButtonCreate.Size = new System.Drawing.Size(110, 23);
this.ButtonCreate.TabIndex = 20;
this.ButtonCreate.Text = "Save && Create User";
this.ButtonCreate.UseVisualStyleBackColor = true;
this.ButtonCreate.Click += new System.EventHandler(this.ButtonCreate_Click);
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(209, 153);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(31, 13);
this.label5.TabIndex = 19;
this.label5.Text = "Type";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(187, 127);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(53, 13);
this.label4.TabIndex = 18;
this.label4.Text = "Password";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(187, 101);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(53, 13);
this.label3.TabIndex = 17;
this.label3.Text = "Lastname";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(205, 75);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(35, 13);
this.label2.TabIndex = 16;
this.label2.Text = "Name";
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(197, 49);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(43, 13);
this.label1.TabIndex = 15;
this.label1.Text = "User ID";
//
// TextBoxNewUserID
//
```

```
            this.TextBoxNewUserID.AcceptsReturn = true;
            this.TextBoxNewUserID.Location = new System.Drawing.Point(246, 46);
            this.TextBoxNewUserID.MaxLength = 50;
            this.TextBoxNewUserID.Name = "TextBoxNewUserID";
            this.TextBoxNewUserID.Size = new System.Drawing.Size(180, 20);
            this.TextBoxNewUserID.TabIndex = 10;
            //
            // ComboBoxType
            //
            this.ComboBoxType.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.ComboBoxType.FormattingEnabled = true;
            this.ComboBoxType.Items.AddRange(new object[] {
            "Facilitator",
            "Participant"});
            this.ComboBoxType.Location = new System.Drawing.Point(246, 150);
            this.ComboBoxType.Name = "ComboBoxType";
            this.ComboBoxType.Size = new System.Drawing.Size(180, 21);
            this.ComboBoxType.TabIndex = 14;
            //
            // TextBoxName
            //
            this.TextBoxName.Location = new System.Drawing.Point(246, 72);
            this.TextBoxName.MaxLength = 50;
            this.TextBoxName.Name = "TextBoxName";
            this.TextBoxName.Size = new System.Drawing.Size(180, 20);
            this.TextBoxName.TabIndex = 11;
            //
            // TextBoxPassword
            //
            this.TextBoxPassword.AcceptsReturn = true;
            this.TextBoxPassword.Location = new System.Drawing.Point(246, 124);
            this.TextBoxPassword.MaxLength = 50;
            this.TextBoxPassword.Name = "TextBoxPassword";
            this.TextBoxPassword.Size = new System.Drawing.Size(180, 20);
            this.TextBoxPassword.TabIndex = 13;
            //
            // TextBoxLastname
            //
            this.TextBoxLastname.Location = new System.Drawing.Point(246, 98);
            this.TextBoxLastname.MaxLength = 50;
            this.TextBoxLastname.Name = "TextBoxLastname";
            this.TextBoxLastname.Size = new System.Drawing.Size(180, 20);
            this.TextBoxLastname.TabIndex = 12;
            //
            // groupBox3
            //
            this.groupBox3.Controls.Add(this.DataGridViewUser);
            this.groupBox3.Dock = System.Windows.Forms.DockStyle.Fill;
            this.groupBox3.Location = new System.Drawing.Point(5, 5);
            this.groupBox3.Name = "groupBox3";
            this.groupBox3.Size = new System.Drawing.Size(711, 443);
            this.groupBox3.TabIndex = 1;
            this.groupBox3.TabStop = false;
            this.groupBox3.Text = "User List";
            //
            // errorProvider1
```

```
            //
            this.errorProvider1.ContainerControl = this;
            //
            // UserSetup
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(721, 678);
            this.ControlBox = false;
            this.Controls.Add(this.groupBox3);
            this.Controls.Add(this.groupBox2);
            this.Name = "UserSetup";
            this.Padding = new System.Windows.Forms.Padding(5);
            this.Text = "User Setup";
            this.Load += new System.EventHandler(this.UserSetup_Load);
            ((System.ComponentModel.ISupportInitialize)(this.DataGridViewUser)).EndInit();
            this.groupBox2.ResumeLayout(false);
            this.groupBox2.PerformLayout();
            this.groupBox3.ResumeLayout(false);
            ((System.ComponentModel.ISupportInitialize)(this.errorProvider1)).EndInit();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.DataGridView DataGridViewUser;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.TextBox TextBoxNewUserID;
        private System.Windows.Forms.ComboBox ComboBoxType;
        private System.Windows.Forms.TextBox TextBoxName;
        private System.Windows.Forms.TextBox TextBoxPassword;
        private System.Windows.Forms.TextBox TextBoxLastname;
        private System.Windows.Forms.Button ButtonUpdate;
        private System.Windows.Forms.Button ButtonCreate;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.ErrorProvider errorProvider1;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.TextBox TextBoxUserID;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.DataGridViewTextBoxColumn UserIDTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn NameTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn LastnameTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn PasswordTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn TypeTextBoxColumn;
        private System.Windows.Forms.DataGridViewButtonColumn DeleteButtonColumn;
    }
}
```

TemplateSetup.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace GroupStorm
{
  public partial class TemplateSetup : Form
  {
    private DataSetBrain.TemplatesDataTable templatesDataTable = new
DataSetBrain.TemplatesDataTable();
    private DataSetBrainTableAdapters.TemplatesTableAdapter templatesTableAdapter = new
GroupStorm.DataSetBrainTableAdapters.TemplatesTableAdapter();
    private DataSetBrain.TemplateDetailsDataTable templateDetailsDataTable = new
DataSetBrain.TemplateDetailsDataTable();
    private DataSetBrainTableAdapters.TemplateDetailsTableAdapter templateDetailsTableAdapter =
new GroupStorm.DataSetBrainTableAdapters.TemplateDetailsTableAdapter();
    private int _TemplateID = 0;
    private bool _TreeChange = false;

    public TemplateSetup()
    {
      InitializeComponent();
    }

    private void TemplateSetup_Load(object sender, EventArgs e)
    {
      //Set Default NodeType ComboBox
      this.ComboBoxNodeType.SelectedIndex = 0;

      //Display list data
      RefreshData();
    }

    /// <summary>
    /// Display selected Template Detail
    /// </summary>
    private void DataGridViewTemplate_CellClick(object sender, DataGridViewCellEventArgs e)
    {
      try
      {
        if (e.RowIndex != -1) //Not Header
        {
          //Display Template Detail
          _TemplateID = (int)this.DataGridViewTemplate[0, e.RowIndex].Value;
          this.TextBoxTemplateID.Text = this.DataGridViewTemplate[0,
e.RowIndex].Value.ToString();
          this.TextBoxTemplateName.Text = this.DataGridViewTemplate[1,
e.RowIndex].Value.ToString();
```

```
            //Get Node Topic
            DataSetBrain.TemplateTreeDataTable templateTreeDataTable = new
DataSetBrain.TemplateTreeDataTable();
            DataSetBrainTableAdapters.TemplateTreeTableAdapter templateTreeTableAdapter = new
GroupStorm.DataSetBrainTableAdapters.TemplateTreeTableAdapter();
            templateTreeDataTable =
templateTreeTableAdapter.GetDataBy_TemplateID(_TemplateID);

            //Dsplay Node Topic Tree
            DisplayTree(templateTreeDataTable);

            _TreeChange = false;
          }
        }
        catch (Exception ex)
        {
          MessageBox.Show(ex.Message);
        }
      }

      /// <summary>
      /// Delete Template
      /// </summary>
      private void DataGridViewTemplate_CellContentClick(object sender,
DataGridViewCellEventArgs e)
      {
        try
        {
          if ((e.RowIndex != -1) && (e.ColumnIndex == 3)) //Click on Delete button
          {
            if (MessageBox.Show("Do you want to delete?", "Delete template",
MessageBoxButtons.YesNo) == DialogResult.Yes) //Confirm Delete
            {
              if (!UsedTemplate(Convert.ToInt32(_TemplateID))) //template not use
              {
                //Delete template
                templatesTableAdapter.Delete(_TemplateID);
                MessageBox.Show("Data was successfully deleted.");

                //Clear display template detail
                _TemplateID = 0;
                this.TextBoxTemplateID.Text = "";
                this.TextBoxTemplateName.Text = "";
                this.TextBoxCurrentNode.Text = "";
                this.TextBoxNewNode.Text = "";
                this.ComboBoxNodeType.SelectedIndex = 0;
                this.treeView1.Nodes.Clear();

                //Display new list data
                RefreshData();
              }
              else
              {
                MessageBox.Show("This Template has been used in a session. Cannot delete.");
              }
            }
          }
```

```
    }
    catch (Exception ex)
    {
      MessageBox.Show(ex.Message);
    }
  }

  /// <summary>
  /// Add Root Topic
  /// </summary>
  private void ButtonAddRoot_Click(object sender, EventArgs e)
  {
    try
    {
      //Get topic node data
      string NodeName = this.TextBoxNewNode.Text.Trim();
      string NodeType = this.ComboBoxNodeType.SelectedItem.ToString();

      if (NodeName == "") //Empty node name
      {
        MessageBox.Show("Please enter Node Name.");
      }
      else
      {
        TreeNode[] tempNode = this.treeView1.Nodes.Find(NodeName, true);
        if (tempNode.Length > 0) //Dupplicate Node Name
        {
          MessageBox.Show("Dupplicate Node Name. Please enter new \"Node Name\".");
        }
        else
        {
          //Add New node to tree
          this.treeView1.Nodes.Add(NodeName, NodeName, NodeType);
          this.TextBoxNewNode.Text = "";
          this.ComboBoxNodeType.SelectedIndex = 0;

          _TreeChange = true;
        }
      }
    }
    catch (Exception ex)
    {
      MessageBox.Show(ex.Message);
    }
  }

  /// <summary>
  /// Add Child topic node
  /// </summary>
  private void ButtonAddChild_Click(object sender, EventArgs e)
  {
    try
    {
      //Get Child Node datail
      string CurrentNode = this.TextBoxCurrentNode.Text;
      string NodeName = this.TextBoxNewNode.Text.Trim();
      string NodeType = this.ComboBoxNodeType.SelectedItem.ToString();
```

```
        if (CurrentNode == "") //Empty parent node
        {
          MessageBox.Show("Please select Node.");
        }
        else if (NodeName == "") //Empty node name
        {
          MessageBox.Show("Please enter Node Name.");
        }
        else
        {
          TreeNode[] tempNode = this.treeView1.Nodes.Find(NodeName, true);
          if (tempNode.Length > 0) //Dupplicate Node Name
          {
            MessageBox.Show("Dupplicate Node Name. Please enter new \"Node Name\".");
          }
          else
          {
            //Add child node to tree
            this.treeView1.SelectedNode.Nodes.Add(NodeName, NodeName, NodeType);
            this.TextBoxNewNode.Text = "";
            this.ComboBoxNodeType.SelectedIndex = 0;

            //Display child node
            this.treeView1.SelectedNode.Expand();

            _TreeChange = true;
          }
        }
      }
      catch (Exception ex)
      {
        MessageBox.Show(ex.Message);
      }
    }

    /// <summary>
    /// Delete node
    /// </summary>
    private void ButtonDeleteNode_Click(object sender, EventArgs e)
    {
      try
      {
        string CurrentNode = this.TextBoxCurrentNode.Text;
        if (CurrentNode == "") //empty selected node
        {
          MessageBox.Show("Please select Node.");
        }
        else
        {
          //remove selected node
          this.treeView1.SelectedNode.Remove();
          if (this.treeView1.SelectedNode == null)
          {
            this.TextBoxCurrentNode.Text = "";
          }
```

```
            _TreeChange = true;
          }
        }
        catch (Exception ex)
        {
          MessageBox.Show(ex.Message);
        }
      }

      /// <summary>
      /// display topic name of selected node
      /// </summary>
      private void treeView1_AfterSelect(object sender, TreeViewEventArgs e)
      {
        this.TextBoxCurrentNode.Text = e.Node.Name;
      }

      /// <summary>
      /// Display Topic Tree
      /// </summary>
      private void DisplayTree(DataTable myTable)
      {
        //set empty tree
        this.treeView1.Nodes.Clear();

        foreach (DataRow myRow in myTable.Rows)
        {
          if (string.IsNullOrEmpty(myRow["ParentTopicName"].ToString()))
          {
            //Add main topic name
            this.treeView1.Nodes.Add(myRow["TopicName"].ToString(),
myRow["TopicName"].ToString(), myRow["KeyMeasurement"].ToString());
          }
          else
          {
            //Add Child topic node to parent topic node
            TreeNode[] treeNode =
this.treeView1.Nodes.Find(myRow["ParentTopicName"].ToString(), true);
            if (treeNode.Length > 0)
            {
              treeNode[0].Nodes.Add(myRow["TopicName"].ToString(),
myRow["TopicName"].ToString(), myRow["KeyMeasurement"].ToString());
            }
          }
        }

        //Show Tree
        this.treeView1.ExpandAll();
      }

      /// <summary>
      /// Display Template List Data
      /// </summary>
      private void RefreshData()
      {
        templatesDataTable = templatesTableAdapter.GetData();
        this.DataGridViewTemplate.AutoGenerateColumns = false;
```

```
        this.DataGridViewTemplate.DataSource = templatesDataTable;
    }

    /// <summary>
    /// Check template is use
    /// </summary>
    private bool UsedTemplate(int templateID)
    {
        DataSetBrainTableAdapters.SessionsTableAdapter sessionsTableAdapter = new
GroupStorm.DataSetBrainTableAdapters.SessionsTableAdapter();
        if (sessionsTableAdapter.GetUseTemplate(templateID) > 0)
        {
            return true;
        }
        else
        {
            return false;
        }
    }

    /// <summary>
    /// Create new template
    /// </summary>
    private void ButtonCreate_Click(object sender, EventArgs e)
    {
        // Create Template
        try
        {
            string TemplateName = this.TextBoxTemplateName.Text.Trim();

            if (CheckSave()) //Varidate template data
            {
                if (CheckDup(TemplateName)) //Duppicate Template Name
                {
                    //Display Error
                    this.errorProvider1.SetError(this.TextBoxTemplateName, "Dupplicate Template Name.
Please enter new \"Template Name\".");
                }
                else
                {
                    if (this.treeView1.Nodes.Count > 0) //have tree node
                    {
                        //Create template
                        templatesTableAdapter.Insert(TemplateName, DateTime.Now, DateTime.Now);
                        int TemplateID = (int)templatesTableAdapter.GetMaxID();

                        //Create Template Detail
                        SaveTree(this.treeView1.Nodes, 0, TemplateID);

                        MessageBox.Show("Data was successfully saved.");

                        //Clear template detail
                        _TemplateID = 0;
                        this.TextBoxTemplateID.Text = "";
                        this.TextBoxTemplateName.Text = "";
                        this.TextBoxCurrentNode.Text = "";
                        this.TextBoxNewNode.Text = "";
```

```
                    this.ComboBoxNodeType.SelectedIndex = 0;
                    this.treeView1.Nodes.Clear();

                    //Display template list data
                    RefreshData();

                    //Clear Error
                    this.errorProvider1.Clear();
                }
                else
                {
                    MessageBox.Show("Please enter Template Detail.");
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    /// <summary>
    /// Update template
    /// </summary>
    private void ButtonUpdate_Click(object sender, EventArgs e)
    {
        //Update Template
        try
        {
            string TemplateName = this.TextBoxTemplateName.Text.Trim();

            if (CheckSave()) //Varidate template data
            {
                if (CheckDup(TemplateName) && (TemplateName !=
this.DataGridViewTemplate.CurrentRow.Cells[1].Value.ToString())) //Dupplicate Template Name
                {
                    //Display Error
                    this.errorProvider1.SetError(this.TextBoxTemplateName, "Dupplicate Template Name.
Please enter new \"Template Name\".");
                }
                else
                {
                    if (!UsedTemplate(Convert.ToInt32(_TemplateID))) //Not used template
                    {
                        if (this.treeView1.Nodes.Count > 0) //have tree node
                        {
                            //Update Template
                            templatesTableAdapter.Update(TemplateName, DateTime.Now, _TemplateID);

                            // Delete Template Detail
                            templateDetailsTableAdapter.Delete(_TemplateID);

                            // Create Template Detail
                            SaveTree(this.treeView1.Nodes, 0, _TemplateID);
                        }
                        else
```

```
                {
                    MessageBox.Show("Please enter Template Detail.");
                    return;
                }
            }
            else
            {
                if (!_TreeChange)
                {
                    if (this.treeView1.Nodes.Count > 0)
                    {
                        //Update Template
                        templatesTableAdapter.Update(TemplateName, DateTime.Now, _TemplateID);

                        // Delete Template Detail
                        templateDetailsTableAdapter.Delete(_TemplateID);

                        // Create Template Detail
                        SaveTree(this.treeView1.Nodes, 0, _TemplateID);
                    }
                    else
                    {
                        MessageBox.Show("Please enter Template Detail.");
                        return;
                    }
                }
                else
                {
                    MessageBox.Show("The Template ID has been added into a session. Template
Detail Tree cannot be changed.");
                    return;
                }
            }

            MessageBox.Show("Data was successfully saved.");

            _TemplateID = 0;
            this.TextBoxTemplateID.Text = "";
            this.TextBoxTemplateName.Text = "";
            this.TextBoxCurrentNode.Text = "";
            this.TextBoxNewNode.Text = "";
            this.ComboBoxNodeType.SelectedIndex = 0;
            this.treeView1.Nodes.Clear();

            RefreshData();

            this.errorProvider1.Clear();
        }
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

/// <summary>
```

```
        /// Varidate Template Data
        /// </summary>
        /// <returns></returns>
        private bool CheckSave()
        {
            bool result = true;

            if (string.IsNullOrEmpty(this.TextBoxTemplateName.Text.Trim())) //Empty Template name
            {
                this.errorProvider1.SetError(this.TextBoxTemplateName, "Please enter \"Template
Name\".");
                result = false;
            }

            return result;
        }

        /// <summary>
        /// Check Dupplicate Template name
        /// </summary>
        private bool CheckDup(string TemplateName)
        {
            bool result = false;

            if (templatesTableAdapter.CheckDup(TemplateName) > 0)
            {
                result = true;
            }

            return result;
        }

        /// <summary>
        /// Save Topic Tree
        /// </summary>
        private void SaveTree(TreeNodeCollection myNode, int Parent, int TemplateID)
        {
            foreach (TreeNode Node in myNode)
            {
                //Save Topic Node
                templateDetailsTableAdapter.Insert(TemplateID, Parent, Node.Name, Node.ImageKey);
                int myIndex = (int)templateDetailsTableAdapter.GetMaxID();

                if (Node.GetNodeCount(false) > 0)
                {
                    SaveTree(Node.Nodes, myIndex, TemplateID);
                }
            }
        }
    }
}
```

TemplateSetup.Designer.cs

```
namespace GroupStorm
```

```
{
    partial class TemplateSetup
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.components = new System.ComponentModel.Container();
            System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle1 = new
System.Windows.Forms.DataGridViewCellStyle();
            this.groupBox1 = new System.Windows.Forms.GroupBox();
            this.ButtonUpdate = new System.Windows.Forms.Button();
            this.ButtonCreate = new System.Windows.Forms.Button();
            this.ButtonAddChild = new System.Windows.Forms.Button();
            this.label7 = new System.Windows.Forms.Label();
            this.ComboBoxNodeType = new System.Windows.Forms.ComboBox();
            this.label6 = new System.Windows.Forms.Label();
            this.ButtonDeleteNode = new System.Windows.Forms.Button();
            this.ButtonAddRoot = new System.Windows.Forms.Button();
            this.TextBoxCurrentNode = new System.Windows.Forms.TextBox();
            this.label5 = new System.Windows.Forms.Label();
            this.TextBoxNewNode = new System.Windows.Forms.TextBox();
            this.label4 = new System.Windows.Forms.Label();
            this.label1 = new System.Windows.Forms.Label();
            this.label3 = new System.Windows.Forms.Label();
            this.treeView1 = new System.Windows.Forms.TreeView();
            this.TextBoxTemplateID = new System.Windows.Forms.TextBox();
            this.TextBoxTemplateName = new System.Windows.Forms.TextBox();
            this.label2 = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.DataGridViewTemplate = new System.Windows.Forms.DataGridView();
            this.errorProvider1 = new System.Windows.Forms.ErrorProvider(this.components);
            this.TemplateIDTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
```

```
        this.TemplateNameTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
        this.CreateDateTextBoxColumn = new
System.Windows.Forms.DataGridViewTextBoxColumn();
        this.DeleteButtonColumn = new System.Windows.Forms.DataGridViewButtonColumn();
        this.groupBox1.SuspendLayout();
        this.groupBox2.SuspendLayout();
        ((System.ComponentModel.ISupportInitialize)(this.DataGridViewTemplate)).BeginInit();
        ((System.ComponentModel.ISupportInitialize)(this.errorProvider1)).BeginInit();
        this.SuspendLayout();
        //
        // groupBox1
        //
        this.groupBox1.Controls.Add(this.ButtonUpdate);
        this.groupBox1.Controls.Add(this.ButtonCreate);
        this.groupBox1.Controls.Add(this.ButtonAddChild);
        this.groupBox1.Controls.Add(this.label7);
        this.groupBox1.Controls.Add(this.ComboBoxNodeType);
        this.groupBox1.Controls.Add(this.label6);
        this.groupBox1.Controls.Add(this.ButtonDeleteNode);
        this.groupBox1.Controls.Add(this.ButtonAddRoot);
        this.groupBox1.Controls.Add(this.TextBoxCurrentNode);
        this.groupBox1.Controls.Add(this.label5);
        this.groupBox1.Controls.Add(this.TextBoxNewNode);
        this.groupBox1.Controls.Add(this.label4);
        this.groupBox1.Controls.Add(this.label1);
        this.groupBox1.Controls.Add(this.label3);
        this.groupBox1.Controls.Add(this.treeView1);
        this.groupBox1.Controls.Add(this.TextBoxTemplateID);
        this.groupBox1.Controls.Add(this.TextBoxTemplateName);
        this.groupBox1.Controls.Add(this.label2);
        this.groupBox1.Dock = System.Windows.Forms.DockStyle.Bottom;
        this.groupBox1.Location = new System.Drawing.Point(5, 199);
        this.groupBox1.Name = "groupBox1";
        this.groupBox1.Size = new System.Drawing.Size(782, 249);
        this.groupBox1.TabIndex = 0;
        this.groupBox1.TabStop = false;
        this.groupBox1.Text = "Template Setup";
        //
        // ButtonUpdate
        //
        this.ButtonUpdate.AutoSize = true;
        this.ButtonUpdate.Location = new System.Drawing.Point(564, 207);
        this.ButtonUpdate.Name = "ButtonUpdate";
        this.ButtonUpdate.Size = new System.Drawing.Size(110, 23);
        this.ButtonUpdate.TabIndex = 23;
        this.ButtonUpdate.Text = "Update Template";
        this.ButtonUpdate.UseVisualStyleBackColor = true;
        this.ButtonUpdate.Click += new System.EventHandler(this.ButtonUpdate_Click);
        //
        // ButtonCreate
        //
        this.ButtonCreate.Location = new System.Drawing.Point(382, 207);
        this.ButtonCreate.Name = "ButtonCreate";
        this.ButtonCreate.Size = new System.Drawing.Size(110, 23);
        this.ButtonCreate.TabIndex = 22;
        this.ButtonCreate.Text = "Save && Create Template";
```

```
this.ButtonCreate.UseVisualStyleBackColor = true;
this.ButtonCreate.Click += new System.EventHandler(this.ButtonCreate_Click);
//
// ButtonAddChild
//
this.ButtonAddChild.Location = new System.Drawing.Point(667, 123);
this.ButtonAddChild.Name = "ButtonAddChild";
this.ButtonAddChild.Size = new System.Drawing.Size(75, 23);
this.ButtonAddChild.TabIndex = 15;
this.ButtonAddChild.Text = "Add Child";
this.ButtonAddChild.UseVisualStyleBackColor = true;
this.ButtonAddChild.Click += new System.EventHandler(this.ButtonAddChild_Click);
//
// label7
//
this.label7.AutoSize = true;
this.label7.Location = new System.Drawing.Point(583, 108);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(146, 13);
this.label7.TabIndex = 14;
this.label7.Text = "(of the selected Curent Node)";
//
// ComboBoxNodeType
//
this.ComboBoxNodeType.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
this.ComboBoxNodeType.FormattingEnabled = true;
this.ComboBoxNodeType.Items.AddRange(new object[] {
"Text",
"Multiple Choice",
"Yes/No"});
this.ComboBoxNodeType.Location = new System.Drawing.Point(397, 152);
this.ComboBoxNodeType.Name = "ComboBoxNodeType";
this.ComboBoxNodeType.Size = new System.Drawing.Size(180, 21);
this.ComboBoxNodeType.TabIndex = 13;
//
// label6
//
this.label6.AutoSize = true;
this.label6.Location = new System.Drawing.Point(326, 155);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(60, 13);
this.label6.TabIndex = 12;
this.label6.Text = "Node Type";
//
// ButtonDeleteNode
//
this.ButtonDeleteNode.Location = new System.Drawing.Point(586, 82);
this.ButtonDeleteNode.Name = "ButtonDeleteNode";
this.ButtonDeleteNode.Size = new System.Drawing.Size(75, 23);
this.ButtonDeleteNode.TabIndex = 11;
this.ButtonDeleteNode.Text = "Delete Node";
this.ButtonDeleteNode.UseVisualStyleBackColor = true;
this.ButtonDeleteNode.Click += new System.EventHandler(this.ButtonDeleteNode_Click);
//
// ButtonAddRoot
//
```

```
this.ButtonAddRoot.Location = new System.Drawing.Point(586, 124);
this.ButtonAddRoot.Name = "ButtonAddRoot";
this.ButtonAddRoot.Size = new System.Drawing.Size(75, 23);
this.ButtonAddRoot.TabIndex = 10;
this.ButtonAddRoot.Text = "Add Root";
this.ButtonAddRoot.UseVisualStyleBackColor = true;
this.ButtonAddRoot.Click += new System.EventHandler(this.ButtonAddRoot_Click);
//
// TextBoxCurrentNode
//
this.TextBoxCurrentNode.Location = new System.Drawing.Point(397, 84);
this.TextBoxCurrentNode.MaxLength = 255;
this.TextBoxCurrentNode.Name = "TextBoxCurrentNode";
this.TextBoxCurrentNode.ReadOnly = true;
this.TextBoxCurrentNode.Size = new System.Drawing.Size(180, 20);
this.TextBoxCurrentNode.TabIndex = 9;
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(321, 87);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(70, 13);
this.label5.TabIndex = 8;
this.label5.Text = "Current Node";
//
// TextBoxNewNode
//
this.TextBoxNewNode.Location = new System.Drawing.Point(397, 126);
this.TextBoxNewNode.MaxLength = 50;
this.TextBoxNewNode.Name = "TextBoxNewNode";
this.TextBoxNewNode.Size = new System.Drawing.Size(180, 20);
this.TextBoxNewNode.TabIndex = 7;
//
// label4
//
this.label4.AutoSize = true;
this.label4.Location = new System.Drawing.Point(309, 129);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(80, 13);
this.label4.TabIndex = 6;
this.label4.Text = "Add New Node";
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(326, 22);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(65, 13);
this.label1.TabIndex = 1;
this.label1.Text = "Template ID";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(583, 22);
this.label3.Name = "label3";
```

```
            this.label3.Size = new System.Drawing.Size(91, 13);
            this.label3.TabIndex = 5;
            this.label3.Text = "(Running number)";
            //
            // treeView1
            //
            this.treeView1.Dock = System.Windows.Forms.DockStyle.Left;
            this.treeView1.Location = new System.Drawing.Point(3, 16);
            this.treeView1.Name = "treeView1";
            this.treeView1.Size = new System.Drawing.Size(278, 230);
            this.treeView1.TabIndex = 0;
            this.treeView1.AfterSelect += new
System.Windows.Forms.TreeViewEventHandler(this.treeView1_AfterSelect);
            //
            // TextBoxTemplateID
            //
            this.TextBoxTemplateID.Location = new System.Drawing.Point(397, 19);
            this.TextBoxTemplateID.Name = "TextBoxTemplateID";
            this.TextBoxTemplateID.ReadOnly = true;
            this.TextBoxTemplateID.Size = new System.Drawing.Size(180, 20);
            this.TextBoxTemplateID.TabIndex = 2;
            //
            // TextBoxTemplateName
            //
            this.TextBoxTemplateName.Location = new System.Drawing.Point(397, 45);
            this.TextBoxTemplateName.MaxLength = 50;
            this.TextBoxTemplateName.Name = "TextBoxTemplateName";
            this.TextBoxTemplateName.Size = new System.Drawing.Size(180, 20);
            this.TextBoxTemplateName.TabIndex = 4;
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(309, 48);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(82, 13);
            this.label2.TabIndex = 3;
            this.label2.Text = "Template Name";
            //
            // groupBox2
            //
            this.groupBox2.Controls.Add(this.DataGridViewTemplate);
            this.groupBox2.Dock = System.Windows.Forms.DockStyle.Fill;
            this.groupBox2.Location = new System.Drawing.Point(5, 5);
            this.groupBox2.Name = "groupBox2";
            this.groupBox2.Size = new System.Drawing.Size(782, 194);
            this.groupBox2.TabIndex = 1;
            this.groupBox2.TabStop = false;
            this.groupBox2.Text = "Template List";
            //
            // DataGridViewTemplate
            //
            this.DataGridViewTemplate.AllowUserToAddRows = false;
            this.DataGridViewTemplate.AllowUserToDeleteRows = false;
            this.DataGridViewTemplate.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
```

```
        this.DataGridViewTemplate.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
        this.TemplateIDTextBoxColumn,
        this.TemplateNameTextBoxColumn,
        this.CreateDateTextBoxColumn,
        this.DeleteButtonColumn});
        this.DataGridViewTemplate.Dock = System.Windows.Forms.DockStyle.Fill;
        this.DataGridViewTemplate.Location = new System.Drawing.Point(3, 16);
        this.DataGridViewTemplate.MultiSelect = false;
        this.DataGridViewTemplate.Name = "DataGridViewTemplate";
        this.DataGridViewTemplate.ReadOnly = true;
        this.DataGridViewTemplate.RowHeadersVisible = false;
        this.DataGridViewTemplate.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
        this.DataGridViewTemplate.Size = new System.Drawing.Size(776, 175);
        this.DataGridViewTemplate.TabIndex = 0;
        this.DataGridViewTemplate.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.DataGridViewTemplate_CellClick);
        this.DataGridViewTemplate.CellContentClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.DataGridViewTemplate_CellContentCli
ck);
        //
        // errorProvider1
        //
        this.errorProvider1.ContainerControl = this;
        //
        // TemplateIDTextBoxColumn
        //
        this.TemplateIDTextBoxColumn.DataPropertyName = "TemplateID";
        this.TemplateIDTextBoxColumn.HeaderText = "Template ID";
        this.TemplateIDTextBoxColumn.Name = "TemplateIDTextBoxColumn";
        this.TemplateIDTextBoxColumn.ReadOnly = true;
        //
        // TemplateNameTextBoxColumn
        //
        this.TemplateNameTextBoxColumn.DataPropertyName = "TemplateName";
        this.TemplateNameTextBoxColumn.FillWeight = 150F;
        this.TemplateNameTextBoxColumn.HeaderText = "Template Name";
        this.TemplateNameTextBoxColumn.Name = "TemplateNameTextBoxColumn";
        this.TemplateNameTextBoxColumn.ReadOnly = true;
        this.TemplateNameTextBoxColumn.Width = 150;
        //
        // CreateDateTextBoxColumn
        //
        this.CreateDateTextBoxColumn.DataPropertyName = "CreatedDate";
        dataGridViewCellStyle1.Format = "dd-MM-yyyy";
        this.CreateDateTextBoxColumn.DefaultCellStyle = dataGridViewCellStyle1;
        this.CreateDateTextBoxColumn.HeaderText = "Created Date";
        this.CreateDateTextBoxColumn.Name = "CreateDateTextBoxColumn";
        this.CreateDateTextBoxColumn.ReadOnly = true;
        //
        // DeleteButtonColumn
        //
        this.DeleteButtonColumn.HeaderText = "Delete";
        this.DeleteButtonColumn.Name = "DeleteButtonColumn";
        this.DeleteButtonColumn.ReadOnly = true;
        this.DeleteButtonColumn.Text = "Delete";
```

```
            this.DeleteButtonColumn.UseColumnTextForButtonValue = true;
            //
            // TemplateSetup
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(792, 453);
            this.ControlBox = false;
            this.Controls.Add(this.groupBox2);
            this.Controls.Add(this.groupBox1);
            this.Name = "TemplateSetup";
            this.Padding = new System.Windows.Forms.Padding(5);
            this.Text = "Template Setup";
            this.Load += new System.EventHandler(this.TemplateSetup_Load);
            this.groupBox1.ResumeLayout(false);
            this.groupBox1.PerformLayout();
            this.groupBox2.ResumeLayout(false);
            ((System.ComponentModel.ISupportInitialize)(this.DataGridViewTemplate)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.errorProvider1)).EndInit();
            this.ResumeLayout(false);

        }

        #endregion

        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.DataGridView DataGridViewTemplate;
        private System.Windows.Forms.TreeView treeView1;
        private System.Windows.Forms.TextBox TextBoxTemplateName;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.TextBox TextBoxTemplateID;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.TextBox TextBoxNewNode;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.TextBox TextBoxCurrentNode;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Button ButtonDeleteNode;
        private System.Windows.Forms.Button ButtonAddRoot;
        private System.Windows.Forms.ComboBox ComboBoxNodeType;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.Button ButtonAddChild;
        private System.Windows.Forms.Button ButtonUpdate;
        private System.Windows.Forms.Button ButtonCreate;
        private System.Windows.Forms.ErrorProvider errorProvider1;
        private System.Windows.Forms.DataGridViewTextBoxColumn TemplateIDTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn TemplateNameTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn CreateDateTextBoxColumn;
        private System.Windows.Forms.DataGridViewButtonColumn DeleteButtonColumn;
    }
}
```

NewSession.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace GroupStorm
{
   public partial class NewSession : Form
   {
      private DataSetBrain.SessionListDataTable sessionListDataTable = new
DataSetBrain.SessionListDataTable();
      private DataSetBrainTableAdapters.SessionListTableAdapter sessionListTableAdapter = new
DataSetBrainTableAdapters.SessionListTableAdapter();
      private DataSetBrain.SessionsDataTable sessionsDataTable = new
DataSetBrain.SessionsDataTable();
      private DataSetBrainTableAdapters.SessionsTableAdapter sessionsTableAdapter = new
DataSetBrainTableAdapters.SessionsTableAdapter();
      private DataSetBrain.SessionParticipantsDataTable sessionParticipantsDataTable = new
DataSetBrain.SessionParticipantsDataTable();
      private DataSetBrainTableAdapters.SessionParticipantsTableAdapter
sessionParticipantsTableAdapter = new
DataSetBrainTableAdapters.SessionParticipantsTableAdapter();
      private string _Status = "New";
      private int _SessionID = 0;
      private int _MaxParticipant = 12;
      private ItemObject _OldItem;

      public NewSession()
      {
         InitializeComponent();
      }

      public String Status
      {
         get { return _Status; }
         set { _Status = value; }
      }

      private void NewSession_Load(object sender, EventArgs e)
      {
         if (_Status == "New") //New Session
         {
            //Hide End Date Column
            this.DataGridViewSession.Columns[3].Visible = false;
            //Hide Delete Session Button
            this.DataGridViewSession.Columns[8].Visible = false;
            //Hide View Session Button
            this.DataGridViewSession.Columns[11].Visible = false;

            //Hide end date session detail
            this.LabelEndDate.Visible = false;
            this.TextBoxEndDate.Visible = false;
         }
         else //Continue or Completed Session
```

```
        {
            //Hide Update Session Button
            this.ButtonUpdate.Visible = false;
            //Hide Delete Session Button
            this.DataGridViewSession.Columns[7].Visible = false;

            if (_Status == "Completed") //Completed Session
            {
                //Hide Close Session Button
                this.DataGridViewSession.Columns[8].Visible = false;
                //Hide Start Session Button
                this.DataGridViewSession.Columns[9].Visible = false;
            }
            else
            {
                //Hide View Session Button
                this.DataGridViewSession.Columns[11].Visible = false;
            }
        }

        //Set Template Combobox
        GetTemplateName();
        //Set Facilitator Combobox
        GetFacilitator();
        //Set Available Combobox
        GetAvailableUser(_SessionID);
        //Set Selected Combobox
        GetSelectedUser(_SessionID);
        //Display Session List
        RefreshData();

        //Set Default Value
        this.ComboBoxTemplateName.SelectedIndex = 0;
        this.ComboBoxMethodology.SelectedIndex = 0;
        this.DateTimePickerBeginDate.Value = DateTime.Now;
        this.ComboBoxFacilitator.SelectedIndex = 0;
        _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;
    }

    /// <summary>
    /// Display Session List
    /// </summary>
    private void RefreshData()
    {
        //Get Session data
        sessionListDataTable = sessionListTableAdapter.GetDataBy_Status(_Status);

        //Display session data on DataGridViewSession
        this.DataGridViewSession.AutoGenerateColumns = false;
        this.DataGridViewSession.DataSource = sessionListDataTable;
    }

    /// <summary>
    /// User select a session on session list
    /// </summary>
    private void DataGridViewSession_CellClick(object sender, DataGridViewCellEventArgs e)
    {
```

```
        if (e.RowIndex != -1) //Click on data list ,not header
        {
          //Display Session Detail
          _SessionID = (int)this.DataGridViewSession[0, e.RowIndex].Value;
          this.TextBoxSessionID.Text = this.DataGridViewSession[0, e.RowIndex].Value.ToString();
          this.TextBoxSessionName.Text = this.DataGridViewSession[1,
e.RowIndex].Value.ToString();
          this.DateTimePickerBeginDate.Value = (DateTime)this.DataGridViewSession[2,
e.RowIndex].Value;
          this.TextBoxEndDate.Text = this.DataGridViewSession[3, e.RowIndex].Value.ToString();

          // Set Template Name
          this.ComboBoxTemplateName.SelectedIndex = 0;
          for (int i = 0; i < this.ComboBoxTemplateName.Items.Count; i++)
          {
            ItemObject Item = (ItemObject)this.ComboBoxTemplateName.Items[i];
            if (Item.Key.ToString() == this.DataGridViewSession[4, e.RowIndex].Value.ToString())
            {
              this.ComboBoxTemplateName.SelectedIndex = i;
            }
          }

          this.ComboBoxMethodology.SelectedItem = this.DataGridViewSession[5,
e.RowIndex].Value.ToString();
          this.TextBoxNumberOfParticipant.Text = this.DataGridViewSession[6,
e.RowIndex].Value.ToString();

          // Set Facilitator
          this.ComboBoxFacilitator.SelectedIndex = 0;
          sessionParticipantsDataTable =
sessionParticipantsTableAdapter.GetDataBy_Facilitator(_SessionID);
          if (sessionParticipantsDataTable.Rows.Count > 0)
          {
            for (int i = 0; i < this.ComboBoxFacilitator.Items.Count; i++)
            {
              ItemObject Item = (ItemObject)this.ComboBoxFacilitator.Items[i];
              if (Item.ValueOfKey.ToString() ==
sessionParticipantsDataTable.Rows[0]["UserID"].ToString())
              {
                this.ComboBoxFacilitator.SelectedIndex = i;
              }
            }
          }

          // Reset ListBoxAvailableParticipant and ListBoxSelectedParticipant
          for (int i = 0; i < this.ListBoxSelectedParticipant.Items.Count; i++)
          {
            this.ListBoxAvailableParticipant.Items.Add(this.ListBoxSelectedParticipant.Items[i]);
          }
          this.ListBoxSelectedParticipant.Items.Clear();

          // Set Participants
          sessionParticipantsDataTable =
sessionParticipantsTableAdapter.GetDataBy_Participant(_SessionID);
          foreach (DataRow myRow in sessionParticipantsDataTable.Rows)
          {
            SwapUser(myRow["UserID"].ToString());
```

```
        }
      }
    }

    /// <summary>
    /// User Click Start, Close, Delete Button on session list
    /// </summary>
    private void DataGridViewSession_CellContentClick(object sender, DataGridViewCellEventArgs
e)
    {
      try
      {
        if (e.RowIndex != -1)
        {
          switch (e.ColumnIndex)
          {
            case 7:
              // Delete Session
              if (MessageBox.Show("Do you want to delete?", "Delete Session",
MessageBoxButtons.YesNo) == DialogResult.Yes)
              {
                sessionParticipantsTableAdapter.Delete(_SessionID);
                sessionsTableAdapter.Delete(_SessionID);

                MessageBox.Show("Data was successfully deleted.");

                //Clear Session Detail
                _SessionID = 0;
                this.TextBoxSessionName.Text = "";
                this.DateTimePickerBeginDate.Value = DateTime.Now;
                this.ComboBoxTemplateName.SelectedIndex = 0;
                this.ComboBoxMethodology.SelectedIndex = 0;
                this.TextBoxNumberOfParticipant.Text = "0";
                this.ComboBoxFacilitator.SelectedIndex = 0;
                _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

                GetAvailableUser(_SessionID);
                GetSelectedUser(_SessionID);

                //Display New data on Session List
                RefreshData();
              }
              break;
            case 8:
              // Close Session
              if (MessageBox.Show("Do you want to close session?", "Close Session",
MessageBoxButtons.YesNo) == DialogResult.Yes)
              {
                //Set Data to save
                string SessionName = this.DataGridViewSession[1,
e.RowIndex].Value.ToString();
                DateTime BeginDate = (DateTime)this.DataGridViewSession[2,
e.RowIndex].Value;
                DateTime EndDate = DateTime.Now;
                int TemplateID = (int)this.DataGridViewSession[10, e.RowIndex].Value;
                string Methodology = this.DataGridViewSession[5, e.RowIndex].Value.ToString();
                int NoOfParticipant = (int)this.DataGridViewSession[6, e.RowIndex].Value;
```

```
                    //Save data to DB
                    sessionsTableAdapter.Update(SessionName, BeginDate, TemplateID,
Methodology, NoOfParticipant, _SessionID);
                    sessionsTableAdapter.Update_EndDate(DateTime.Now, _SessionID);
                    sessionsTableAdapter.Update_Status("Completed", _SessionID);

                    MessageBox.Show("Session was successfully closed.");

                    //Clear Session Detail
                    _SessionID = 0;
                    this.TextBoxSessionName.Text = "";
                    this.DateTimePickerBeginDate.Value = DateTime.Now;
                    this.ComboBoxTemplateName.SelectedIndex = 0;
                    this.ComboBoxMethodology.SelectedIndex = 0;
                    this.TextBoxNumberOfParticipant.Text = "0";
                    this.ComboBoxFacilitator.SelectedIndex = 0;
                    _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

                    GetAvailableUser(_SessionID);
                    GetSelectedUser(_SessionID);

                    //Display New data on Session List
                    RefreshData();
                }
                break;

            case 9:
                // Start Session
                string KeepMethodology = this.DataGridViewSession[5,
e.RowIndex].Value.ToString();

                // Open Session to Brain strom
                MDIParent1 TempMDI = (MDIParent1)this.MdiParent;
                TempMDI.OpenKeyboardForm(_SessionID, _Status, KeepMethodology);

                // Set to default
                _SessionID = 0;
                this.TextBoxSessionName.Text = "";
                this.DateTimePickerBeginDate.Value = DateTime.Now;
                this.ComboBoxTemplateName.SelectedIndex = 0;
                this.ComboBoxMethodology.SelectedIndex = 0;
                this.TextBoxNumberOfParticipant.Text = "0";
                this.ComboBoxFacilitator.SelectedIndex = 0;
                _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

                GetAvailableUser(_SessionID);
                GetSelectedUser(_SessionID);

                //Display New data on Session List
                RefreshData();
                break;

            case 11:
                // View Session
                string KeepMethodology2 = this.DataGridViewSession[5,
e.RowIndex].Value.ToString();
```

```
                    // Open Session to Brain strom
                    MDIParent1 TempMDI2 = (MDIParent1)this.MdiParent;
                    TempMDI2.OpenKeyboardForm(_SessionID, _Status, KeepMethodology2);

                    // Set to default
                    _SessionID = 0;
                    this.TextBoxSessionName.Text = "";
                    this.DateTimePickerBeginDate.Value = DateTime.Now;
                    this.ComboBoxTemplateName.SelectedIndex = 0;
                    this.ComboBoxMethodology.SelectedIndex = 0;
                    this.TextBoxNumberOfParticipant.Text = "0";
                    this.ComboBoxFacilitator.SelectedIndex = 0;
                    _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

                    GetAvailableUser(_SessionID);
                    GetSelectedUser(_SessionID);

                    //Display New data on Session List
                    RefreshData();
                    break;
                }
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }

    private void GetTemplateName()
    {
        //Get Template Data
        DataSetBrain.TemplatesDataTable templatesDataTable = new
DataSetBrain.TemplatesDataTable();
        DataSetBrainTableAdapters.TemplatesTableAdapter templatesTableAdapter = new
DataSetBrainTableAdapters.TemplatesTableAdapter();

        templatesDataTable = templatesTableAdapter.GetData();

        //Set Template Data to ComboBoxTemplateName
        this.ComboBoxTemplateName.Items.Clear();
        this.ComboBoxTemplateName.Items.Add(new ItemObject("Not use template", "0"));
        foreach (DataRow myRow in templatesDataTable.Rows)
        {
            //ItemObject(Display, Value)
            this.ComboBoxTemplateName.Items.Add(new
ItemObject(myRow["TemplateName"].ToString(), myRow["TemplateID"].ToString()));
        }
    }

    private void GetFacilitator()
    {
        //Get Facilitator Data
        DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
        DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();
```

```
        usersDataTable = usersTableAdapter.GetDataBy_Facilitator();

        //Set Facilitator Data to ComboBoxFacilitator
        this.ComboBoxFacilitator.Items.Clear();
        this.ComboBoxFacilitator.Sorted = true;
        foreach (DataRow myRow in usersDataTable.Rows)
        {
          //ItemObject(Display, Value)
          this.ComboBoxFacilitator.Items.Add(new ItemObject(myRow["UserID"].ToString(),
myRow["UserID"].ToString()));
        }

        this.ComboBoxFacilitator.Items.Add(new ItemObject("-Select Facilitator-", ""));
      }

      private void GetAvailableUser(int SessionID)
      {
        //Get Available Participant
        DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
        DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();

        usersDataTable = usersTableAdapter.GetDataBy_AvailableUser(SessionID);

        //Set Available Participant to ListBoxAvailableParticipant
        this.ListBoxAvailableParticipant.Items.Clear();
        this.ListBoxAvailableParticipant.Sorted = true;
        foreach (DataRow myRow in usersDataTable.Rows)
        {
          //ItemObject(Display, Value)
          this.ListBoxAvailableParticipant.Items.Add(new ItemObject(myRow["UserID"].ToString(),
myRow["UserID"].ToString()));
        }
      }

      private void GetSelectedUser(int SessionID)
      {
        //Get Selected Participant
        DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
        DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();

        usersDataTable = usersTableAdapter.GetDataBy_SelectedUser(SessionID);

        //Set Selected Participant to ListBoxSelectedParticipant
        this.ListBoxSelectedParticipant.Items.Clear();
        this.ListBoxSelectedParticipant.Sorted = true;
        foreach (DataRow myRow in usersDataTable.Rows)
        {
          //ItemObject(Display, Value)
          this.ListBoxSelectedParticipant.Items.Add(new ItemObject(myRow["UserID"].ToString(),
myRow["UserID"].ToString()));
        }
      }

      /// <summary>
```

```
        /// Select Participant and Set Participant No. of Session
        /// </summary>
        private void button1_Click(object sender, EventArgs e)
        {
          //Select Participant
          while (this.ListBoxAvailableParticipant.SelectedItems.Count > 0)
          {
            int NoOfParticipant = Convert.ToInt32(this.TextBoxNumberOfParticipant.Text) + 1;
            if (NoOfParticipant <= _MaxParticipant)
            {
              this.TextBoxNumberOfParticipant.Text = NoOfParticipant.ToString();

this.ListBoxSelectedParticipant.Items.Add(this.ListBoxAvailableParticipant.SelectedItem);

this.ListBoxAvailableParticipant.Items.Remove(this.ListBoxAvailableParticipant.SelectedItem);

              if (NoOfParticipant == _MaxParticipant)
              {
                this.button1.Enabled = false;
                return;
              }
            }
          }
        }

        /// <summary>
        /// Unselect Participant and Set Participant No. of Session
        /// </summary>
        private void button2_Click(object sender, EventArgs e)
        {
          //Unselect Participant
          while (this.ListBoxSelectedParticipant.SelectedItems.Count > 0)
          {
            int NoOfParticipant = Convert.ToInt32(this.TextBoxNumberOfParticipant.Text) - 1;
            if (NoOfParticipant < _MaxParticipant)
            {
              this.button1.Enabled = true;
              this.TextBoxNumberOfParticipant.Text = NoOfParticipant.ToString();

this.ListBoxAvailableParticipant.Items.Add(this.ListBoxSelectedParticipant.SelectedItem);

this.ListBoxSelectedParticipant.Items.Remove(this.ListBoxSelectedParticipant.SelectedItem);
            }
          }
        }

        /// <summary>
        /// Create New Session
        /// </summary>
        private void ButtonCreate_Click(object sender, EventArgs e)
        {
          // Create Session
          try
          {
            //Set Data
            string SessionName = this.TextBoxSessionName.Text.Trim();
            DateTime BeginDate = this.DateTimePickerBeginDate.Value;
```

```
            ItemObject SelectedItem = (ItemObject)this.ComboBoxTemplateName.SelectedItem;
            int TemplateID = Convert.ToInt32(SelectedItem.ValueOfKey);
            string Methodology = this.ComboBoxMethodology.SelectedItem.ToString();
            SelectedItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;
            string Facilitator = SelectedItem.ValueOfKey.ToString();
            int NoOfParticipant = Convert.ToInt32(this.TextBoxNumberOfParticipant.Text);

            if (CheckSave()) //Varidate Data
            {
              if (CheckDup(SessionName)) //Dupplicate SessionName
              {
                //Display error
                this.errorProvider1.SetError(this.TextBoxSessionName, "Dupplicate Session Name.
Please enter new \"Session Name\".");
              }
              else
              {
                // Create Session
                if (TemplateID <= 0)
                {
                  Methodology = "Anonymous";
                  NoOfParticipant = 0;
                }

                //Save Session Detail
                sessionsTableAdapter.Insert(SessionName, BeginDate, TemplateID, Methodology,
NoOfParticipant, "New");

                if (TemplateID > 0)
                {
                  // Save Session Facilitator
                  int SessionID = (int)sessionsTableAdapter.GetMaxID();
                  sessionParticipantsTableAdapter.Insert(SessionID, Facilitator, "Facilitator");
                  // Save Session Participant
                  for (int i = 0; i < this.ListBoxSelectedParticipant.Items.Count; i++)
                  {
                    ItemObject Item = (ItemObject)this.ListBoxSelectedParticipant.Items[i];
                    sessionParticipantsTableAdapter.Insert(SessionID, Item.ValueOfKey.ToString(),
"Participant");
                  }
                }

                MessageBox.Show("Data was successfully saved.");

                //Clear Session Detail
                _SessionID = 0;
                this.TextBoxSessionName.Text = "";
                this.DateTimePickerBeginDate.Value = DateTime.Now;
                this.ComboBoxTemplateName.SelectedIndex = 0;
                this.ComboBoxMethodology.SelectedIndex = 0;
                this.TextBoxNumberOfParticipant.Text = "0";
                this.ComboBoxFacilitator.SelectedIndex = 0;
                _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

                GetAvailableUser(_SessionID);
                GetSelectedUser(_SessionID);
```

```
                //Display New data on Session List
                RefreshData();

                //Clear Error
                this.errorProvider1.Clear();
              }
            }
          }
          catch (Exception ex)
          {
            MessageBox.Show(ex.Message);
          }
        }

        /// <summary>
        /// Update Selected Session
        /// </summary>
        private void ButtonUpdate_Click(object sender, EventArgs e)
        {
          // Update Session
          try
          {
            //Set Data
            string SessionName = this.TextBoxSessionName.Text.Trim();
            DateTime BeginDate = this.DateTimePickerBeginDate.Value;
            ItemObject SelectedItem = (ItemObject)this.ComboBoxTemplateName.SelectedItem;
            int TemplateID = Convert.ToInt32(SelectedItem.ValueOfKey.ToString());
            string Methodology = this.ComboBoxMethodology.SelectedItem.ToString();
            SelectedItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;
            string Facilitator = SelectedItem.ValueOfKey.ToString();
            int NoOfParticipant = Convert.ToInt32(this.TextBoxNumberOfParticipant.Text);

            if (CheckSave()) //Varidate Data
            {
              if (CheckDup(SessionName) && (SessionName !=
this.DataGridViewSession.CurrentRow.Cells[1].Value.ToString())) //Dupplicate SessionName
              {
                //Display Error
                this.errorProvider1.SetError(this.TextBoxSessionName, "Dupplicate Session Name.
Please enter new \"Session Name\".");
              }
              else
              {
                if (TemplateID <= 0)
                {
                  Methodology = "Anonymous";
                  NoOfParticipant = 0;
                }

                // Update Session
                sessionsTableAdapter.Update(SessionName, BeginDate, TemplateID, Methodology,
NoOfParticipant, _SessionID);
                // Delete Session Facilitator and Participant
                sessionParticipantsTableAdapter.Delete(_SessionID);

                if (TemplateID > 0)
                {
```

```
                    // Save Session Facilitator
                    sessionParticipantsTableAdapter.Insert(_SessionID, Facilitator, "Facilitator");
                    // Save Session Participant
                    for (int i = 0; i < this.ListBoxSelectedParticipant.Items.Count; i++)
                    {
                        ItemObject Item = (ItemObject)this.ListBoxSelectedParticipant.Items[i];
                        sessionParticipantsTableAdapter.Insert(_SessionID, Item.ValueOfKey.ToString(),
"Participant");
                    }
                }

                MessageBox.Show("Data was successfully saved.");

                //Clear Session Detail
                _SessionID = 0;
                this.TextBoxSessionName.Text = "";
                this.DateTimePickerBeginDate.Value = DateTime.Now;
                this.ComboBoxTemplateName.SelectedIndex = 0;
                this.ComboBoxMethodology.SelectedIndex = 0;
                this.TextBoxNumberOfParticipant.Text = "0";
                this.ComboBoxFacilitator.SelectedIndex = 0;
                _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

                GetAvailableUser(_SessionID);
                GetSelectedUser(_SessionID);

                //Display New data on Session List
                RefreshData();

                //Clear Error
                this.errorProvider1.Clear();
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

/// <summary>
/// Varidate Data
/// </summary>
private bool CheckSave()
{
    bool result = true;

    //Session Name can not empty
    if (string.IsNullOrEmpty(this.TextBoxSessionName.Text.Trim()))
    {
        this.errorProvider1.SetError(this.TextBoxSessionName, "Please enter \"Session Name\".");
        result = false;
    }

    ItemObject SelectedItem = (ItemObject)this.ComboBoxTemplateName.SelectedItem;
    int TemplateID = Convert.ToInt32(SelectedItem.ValueOfKey);
    if (TemplateID > 0)
```

```
            {
                //Facilitator can not empty
                SelectedItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;
                if (SelectedItem.ValueOfKey.ToString() == "")
                {
                    this.errorProvider1.SetError(this.ComboBoxFacilitator, "Please select \"Facilitator\".");
                    result = false;
                }

                //Participant can not empty
                int NoOfParticipant = Convert.ToInt32(this.TextBoxNumberOfParticipant.Text);
                if (NoOfParticipant <= 0)
                {
                    this.errorProvider1.SetError(this.TextBoxNumberOfParticipant, "Please select
\"Participant\".");
                    result = false;
                }
            }
            return result;
        }

        /// <summary>
        /// Check Dupplicate SessionName
        /// </summary>
        private bool CheckDup(string SessionName)
        {
            bool result = false;

            if (sessionsTableAdapter.CheckDup(SessionName) > 0)
            {
                result = true;
            }

            return result;
        }

        private void ComboBoxFacilitator_SelectedIndexChanged(object sender, EventArgs e)
        {
            //Remove Facilitator User from ListBoxAvailableParticipant and ListBoxSelectedParticipant

            ItemObject selectedItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;
            //MessageBox.Show(selectedItem.ValueOfKey.ToString());

            for (int i = 0; i < this.ListBoxAvailableParticipant.Items.Count; i++)
            {
                ItemObject Item = (ItemObject)this.ListBoxAvailableParticipant.Items[i];
                if (selectedItem.ValueOfKey.ToString() == Item.ValueOfKey.ToString())
                {

this.ListBoxAvailableParticipant.Items.Remove(this.ListBoxAvailableParticipant.Items[i]);
                    break;
                }
            }

            for (int i = 0; i < this.ListBoxSelectedParticipant.Items.Count; i++)
            {
                ItemObject Item = (ItemObject)this.ListBoxSelectedParticipant.Items[i];
```

```
            if (selectedItem.ValueOfKey.ToString() == Item.ValueOfKey.ToString())
            {
              this.ListBoxSelectedParticipant.Items.Remove(this.ListBoxSelectedParticipant.Items[i]);

              int NoOfParticipant = Convert.ToInt32(this.TextBoxNumberOfParticipant.Text) - 1;
              this.TextBoxNumberOfParticipant.Text = NoOfParticipant.ToString();
              break;
            }
          }

          if (_OldItem != null)
          {
            if (_OldItem.ValueOfKey.ToString() != "")
            {
              this.ListBoxAvailableParticipant.Items.Add(_OldItem);
            }
            _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;
          }
        }

        /// <summary>
        /// Swap Item between ListBoxAvailableParticipant and ListBoxSelectedParticipant
        /// </summary>
        private void SwapUser(string UserID)
        {
          //Swap Item between ListBoxAvailableParticipant and ListBoxSelectedParticipant
          for (int i = 0; i < this.ListBoxAvailableParticipant.Items.Count; i++)
          {
            ItemObject Item = (ItemObject)this.ListBoxAvailableParticipant.Items[i];
            if (Item.ValueOfKey.ToString() == UserID)
            {
              this.ListBoxSelectedParticipant.Items.Add(this.ListBoxAvailableParticipant.Items[i]);

this.ListBoxAvailableParticipant.Items.Remove(this.ListBoxAvailableParticipant.Items[i]);
              return;
            }
          }
        }

        private void ComboBoxTemplateName_SelectedIndexChanged(object sender, EventArgs e)
        {
          //if not use template, reset ListBoxAvailableParticipant and ListBoxSelectedParticipant
          ItemObject SelectedItem = (ItemObject)this.ComboBoxTemplateName.SelectedItem;
          int TemplateID = Convert.ToInt32(SelectedItem.ValueOfKey);
          if (TemplateID <= 0)
          {
            this.ComboBoxMethodology.SelectedItem = "Anonymous";
            this.ComboBoxFacilitator.SelectedIndex = 0;
            _OldItem = (ItemObject)this.ComboBoxFacilitator.SelectedItem;

            for (int i = 0; i < this.ListBoxSelectedParticipant.Items.Count; i++)
            {
              this.ListBoxAvailableParticipant.Items.Add(this.ListBoxSelectedParticipant.Items[i]);
            }
            this.ListBoxSelectedParticipant.Items.Clear();

            this.TextBoxNumberOfParticipant.Text = "0";
```

```
        }
      }
    }
}
```

NewSession.Designer.cs

```
namespace GroupStorm
{
   partial class NewSession
   {
      /// <summary>
      /// Required designer variable.
      /// </summary>
      private System.ComponentModel.IContainer components = null;

      /// <summary>
      /// Clean up any resources being used.
      /// </summary>
      /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
      protected override void Dispose(bool disposing)
      {
         if (disposing && (components != null))
         {
            components.Dispose();
         }
         base.Dispose(disposing);
      }

      #region Windows Form Designer generated code

      /// <summary>
      /// Required method for Designer support - do not modify
      /// the contents of this method with the code editor.
      /// </summary>
      private void InitializeComponent()
      {
         this.components = new System.ComponentModel.Container();
         System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle1 = new
System.Windows.Forms.DataGridViewCellStyle();
         System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle2 = new
System.Windows.Forms.DataGridViewCellStyle();
         this.groupBox1 = new System.Windows.Forms.GroupBox();
         this.ComboBoxFacilitator = new System.Windows.Forms.ComboBox();
         this.label6 = new System.Windows.Forms.Label();
         this.DateTimePickerBeginDate = new System.Windows.Forms.DateTimePicker();
         this.ButtonUpdate = new System.Windows.Forms.Button();
         this.ButtonCreate = new System.Windows.Forms.Button();
         this.groupBox3 = new System.Windows.Forms.GroupBox();
         this.label9 = new System.Windows.Forms.Label();
         this.label10 = new System.Windows.Forms.Label();
         this.TextBoxNumberOfParticipant = new System.Windows.Forms.TextBox();
```

```
            this.label8 = new System.Windows.Forms.Label();
            this.button2 = new System.Windows.Forms.Button();
            this.button1 = new System.Windows.Forms.Button();
            this.ListBoxSelectedParticipant = new System.Windows.Forms.ListBox();
            this.ListBoxAvailableParticipant = new System.Windows.Forms.ListBox();
            this.label7 = new System.Windows.Forms.Label();
            this.ComboBoxMethodology = new System.Windows.Forms.ComboBox();
            this.TextBoxEndDate = new System.Windows.Forms.TextBox();
            this.LabelEndDate = new System.Windows.Forms.Label();
            this.label5 = new System.Windows.Forms.Label();
            this.ComboBoxTemplateName = new System.Windows.Forms.ComboBox();
            this.label4 = new System.Windows.Forms.Label();
            this.TextBoxSessionName = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.TextBoxSessionID = new System.Windows.Forms.TextBox();
            this.label1 = new System.Windows.Forms.Label();
            this.groupBox2 = new System.Windows.Forms.GroupBox();
            this.DataGridViewSession = new System.Windows.Forms.DataGridView();
            this.errorProvider1 = new System.Windows.Forms.ErrorProvider(this.components);
            this.SessionIDTextBoxColumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.SessionNameTextBoxColumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.BeginDateTextBoxColumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.EndDateTextBoxColumn = new System.Windows.Forms.DataGridViewTextBoxColumn();
            this.TemplateNameTextBoxColumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.MethodologyTextBoxCloumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.NoOfParticipantTextBoxColumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.DeleteButtonColumn = new System.Windows.Forms.DataGridViewButtonColumn();
            this.CloseButtonColumn = new System.Windows.Forms.DataGridViewButtonColumn();
            this.StartButtonColumn = new System.Windows.Forms.DataGridViewButtonColumn();
            this.TemplateIDTextBoxColumn = new
        System.Windows.Forms.DataGridViewTextBoxColumn();
            this.ViewButtonColumn = new System.Windows.Forms.DataGridViewButtonColumn();
            this.groupBox1.SuspendLayout();
            this.groupBox3.SuspendLayout();
            this.groupBox2.SuspendLayout();
            ((System.ComponentModel.ISupportInitialize)(this.DataGridViewSession)).BeginInit();
            ((System.ComponentModel.ISupportInitialize)(this.errorProvider1)).BeginInit();
            this.SuspendLayout();
            //
            // groupBox1
            //
            this.groupBox1.Controls.Add(this.ComboBoxFacilitator);
            this.groupBox1.Controls.Add(this.label6);
            this.groupBox1.Controls.Add(this.DateTimePickerBeginDate);
            this.groupBox1.Controls.Add(this.ButtonUpdate);
            this.groupBox1.Controls.Add(this.ButtonCreate);
            this.groupBox1.Controls.Add(this.groupBox3);
            this.groupBox1.Controls.Add(this.label7);
            this.groupBox1.Controls.Add(this.ComboBoxMethodology);
            this.groupBox1.Controls.Add(this.TextBoxEndDate);
```

```
            this.groupBox1.Controls.Add(this.LabelEndDate);
            this.groupBox1.Controls.Add(this.label5);
            this.groupBox1.Controls.Add(this.ComboBoxTemplateName);
            this.groupBox1.Controls.Add(this.label4);
            this.groupBox1.Controls.Add(this.TextBoxSessionName);
            this.groupBox1.Controls.Add(this.label3);
            this.groupBox1.Controls.Add(this.label2);
            this.groupBox1.Controls.Add(this.TextBoxSessionID);
            this.groupBox1.Controls.Add(this.label1);
            this.groupBox1.Dock = System.Windows.Forms.DockStyle.Bottom;
            this.groupBox1.Location = new System.Drawing.Point(5, 233);
            this.groupBox1.Name = "groupBox1";
            this.groupBox1.Size = new System.Drawing.Size(775, 241);
            this.groupBox1.TabIndex = 0;
            this.groupBox1.TabStop = false;
            this.groupBox1.Text = "Session Detail";
            //
            // ComboBoxFacilitator
            //
            this.ComboBoxFacilitator.DropDownStyle =
    System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.ComboBoxFacilitator.FormattingEnabled = true;
            this.ComboBoxFacilitator.Location = new System.Drawing.Point(100, 177);
            this.ComboBoxFacilitator.Name = "ComboBoxFacilitator";
            this.ComboBoxFacilitator.Size = new System.Drawing.Size(180, 21);
            this.ComboBoxFacilitator.TabIndex = 26;
            this.ComboBoxFacilitator.SelectedIndexChanged += new
    System.EventHandler(this.ComboBoxFacilitator_SelectedIndexChanged);
            //
            // label6
            //
            this.label6.AutoSize = true;
            this.label6.Location = new System.Drawing.Point(42, 180);
            this.label6.Name = "label6";
            this.label6.Size = new System.Drawing.Size(52, 13);
            this.label6.TabIndex = 25;
            this.label6.Text = "Facilitator";
            //
            // DateTimePickerBeginDate
            //
            this.DateTimePickerBeginDate.CustomFormat = "dd-MM-yyyy";
            this.DateTimePickerBeginDate.Format =
    System.Windows.Forms.DateTimePickerFormat.Custom;
            this.DateTimePickerBeginDate.Location = new System.Drawing.Point(100, 71);
            this.DateTimePickerBeginDate.Name = "DateTimePickerBeginDate";
            this.DateTimePickerBeginDate.Size = new System.Drawing.Size(180, 20);
            this.DateTimePickerBeginDate.TabIndex = 24;
            //
            // ButtonUpdate
            //
            this.ButtonUpdate.AutoSize = true;
            this.ButtonUpdate.Location = new System.Drawing.Point(423, 212);
            this.ButtonUpdate.Name = "ButtonUpdate";
            this.ButtonUpdate.Size = new System.Drawing.Size(110, 23);
            this.ButtonUpdate.TabIndex = 23;
            this.ButtonUpdate.Text = "Update Session";
            this.ButtonUpdate.UseVisualStyleBackColor = true;
```

```
this.ButtonUpdate.Click += new System.EventHandler(this.ButtonUpdate_Click);
//
// ButtonCreate
//
this.ButtonCreate.Location = new System.Drawing.Point(241, 212);
this.ButtonCreate.Name = "ButtonCreate";
this.ButtonCreate.Size = new System.Drawing.Size(136, 23);
this.ButtonCreate.TabIndex = 22;
this.ButtonCreate.Text = "Save && Create Session";
this.ButtonCreate.UseVisualStyleBackColor = true;
this.ButtonCreate.Click += new System.EventHandler(this.ButtonCreate_Click);
//
// groupBox3
//
this.groupBox3.Controls.Add(this.label9);
this.groupBox3.Controls.Add(this.label10);
this.groupBox3.Controls.Add(this.TextBoxNumberOfParticipant);
this.groupBox3.Controls.Add(this.label8);
this.groupBox3.Controls.Add(this.button2);
this.groupBox3.Controls.Add(this.button1);
this.groupBox3.Controls.Add(this.ListBoxSelectedParticipant);
this.groupBox3.Controls.Add(this.ListBoxAvailableParticipant);
this.groupBox3.Location = new System.Drawing.Point(383, 6);
this.groupBox3.Name = "groupBox3";
this.groupBox3.Size = new System.Drawing.Size(386, 200);
this.groupBox3.TabIndex = 13;
this.groupBox3.TabStop = false;
this.groupBox3.Text = "Participants";
//
// label9
//
this.label9.AutoSize = true;
this.label9.Location = new System.Drawing.Point(29, 16);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(108, 13);
this.label9.TabIndex = 8;
this.label9.Text = "Available Participants";
//
// label10
//
this.label10.AutoSize = true;
this.label10.Location = new System.Drawing.Point(252, 16);
this.label10.Name = "label10";
this.label10.Size = new System.Drawing.Size(102, 13);
this.label10.TabIndex = 7;
this.label10.Text = "Selected Participant";
//
// TextBoxNumberOfParticipant
//
this.TextBoxNumberOfParticipant.AcceptsReturn = true;
this.TextBoxNumberOfParticipant.Location = new System.Drawing.Point(162, 174);
this.TextBoxNumberOfParticipant.Name = "TextBoxNumberOfParticipant";
this.TextBoxNumberOfParticipant.ReadOnly = true;
this.TextBoxNumberOfParticipant.Size = new System.Drawing.Size(180, 20);
this.TextBoxNumberOfParticipant.TabIndex = 5;
this.TextBoxNumberOfParticipant.Text = "0";
//
```

```
            // label8
            //
            this.label8.AutoSize = true;
            this.label8.Location = new System.Drawing.Point(45, 177);
            this.label8.Name = "label8";
            this.label8.Size = new System.Drawing.Size(111, 13);
            this.label8.TabIndex = 4;
            this.label8.Text = "Number Of Participant";
            //
            // button2
            //
            this.button2.Location = new System.Drawing.Point(179, 117);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(27, 23);
            this.button2.TabIndex = 3;
            this.button2.Text = "<";
            this.button2.UseVisualStyleBackColor = true;
            this.button2.Click += new System.EventHandler(this.button2_Click);
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(179, 56);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(27, 23);
            this.button1.TabIndex = 2;
            this.button1.Text = ">";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.Click += new System.EventHandler(this.button1_Click);
            //
            // ListBoxSelectedParticipant
            //
            this.ListBoxSelectedParticipant.FormattingEnabled = true;
            this.ListBoxSelectedParticipant.Location = new System.Drawing.Point(225, 32);
            this.ListBoxSelectedParticipant.Name = "ListBoxSelectedParticipant";
            this.ListBoxSelectedParticipant.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;
            this.ListBoxSelectedParticipant.Size = new System.Drawing.Size(154, 134);
            this.ListBoxSelectedParticipant.TabIndex = 1;
            //
            // ListBoxAvailableParticipant
            //
            this.ListBoxAvailableParticipant.FormattingEnabled = true;
            this.ListBoxAvailableParticipant.Location = new System.Drawing.Point(6, 32);
            this.ListBoxAvailableParticipant.Name = "ListBoxAvailableParticipant";
            this.ListBoxAvailableParticipant.SelectionMode =
System.Windows.Forms.SelectionMode.MultiExtended;
            this.ListBoxAvailableParticipant.Size = new System.Drawing.Size(154, 134);
            this.ListBoxAvailableParticipant.TabIndex = 0;
            //
            // label7
            //
            this.label7.AutoSize = true;
            this.label7.Location = new System.Drawing.Point(26, 153);
            this.label7.Name = "label7";
            this.label7.Size = new System.Drawing.Size(68, 13);
            this.label7.TabIndex = 12;
            this.label7.Text = "Methodology";
```

```
            //
            // ComboBoxMethodology
            //
            this.ComboBoxMethodology.DropDownStyle =
    System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.ComboBoxMethodology.FormattingEnabled = true;
            this.ComboBoxMethodology.Items.AddRange(new object[] {
            "Anonymous",
            "Synonymous"});
            this.ComboBoxMethodology.Location = new System.Drawing.Point(100, 150);
            this.ComboBoxMethodology.Name = "ComboBoxMethodology";
            this.ComboBoxMethodology.Size = new System.Drawing.Size(180, 21);
            this.ComboBoxMethodology.TabIndex = 11;
            //
            // TextBoxEndDate
            //
            this.TextBoxEndDate.AcceptsReturn = true;
            this.TextBoxEndDate.Location = new System.Drawing.Point(100, 97);
            this.TextBoxEndDate.Name = "TextBoxEndDate";
            this.TextBoxEndDate.ReadOnly = true;
            this.TextBoxEndDate.Size = new System.Drawing.Size(180, 20);
            this.TextBoxEndDate.TabIndex = 10;
            //
            // LabelEndDate
            //
            this.LabelEndDate.AutoSize = true;
            this.LabelEndDate.Location = new System.Drawing.Point(42, 100);
            this.LabelEndDate.Name = "LabelEndDate";
            this.LabelEndDate.Size = new System.Drawing.Size(52, 13);
            this.LabelEndDate.TabIndex = 9;
            this.LabelEndDate.Text = "End Date";
            //
            // label5
            //
            this.label5.AutoSize = true;
            this.label5.Location = new System.Drawing.Point(12, 126);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(82, 13);
            this.label5.TabIndex = 8;
            this.label5.Text = "Template Name";
            //
            // ComboBoxTemplateName
            //
            this.ComboBoxTemplateName.DropDownStyle =
    System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.ComboBoxTemplateName.FormattingEnabled = true;
            this.ComboBoxTemplateName.Location = new System.Drawing.Point(100, 123);
            this.ComboBoxTemplateName.Name = "ComboBoxTemplateName";
            this.ComboBoxTemplateName.Size = new System.Drawing.Size(180, 21);
            this.ComboBoxTemplateName.TabIndex = 7;
            this.ComboBoxTemplateName.SelectedIndexChanged += new
    System.EventHandler(this.ComboBoxTemplateName_SelectedIndexChanged);
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Location = new System.Drawing.Point(34, 75);
```

```
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(60, 13);
this.label4.TabIndex = 5;
this.label4.Text = "Begin Date";
//
// TextBoxSessionName
//
this.TextBoxSessionName.Location = new System.Drawing.Point(100, 45);
this.TextBoxSessionName.Name = "TextBoxSessionName";
this.TextBoxSessionName.Size = new System.Drawing.Size(180, 20);
this.TextBoxSessionName.TabIndex = 4;
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(19, 48);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(75, 13);
this.label3.TabIndex = 3;
this.label3.Text = "Session Name";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Location = new System.Drawing.Point(286, 22);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(91, 13);
this.label2.TabIndex = 2;
this.label2.Text = "(Running number)";
//
// TextBoxSessionID
//
this.TextBoxSessionID.Location = new System.Drawing.Point(100, 19);
this.TextBoxSessionID.Name = "TextBoxSessionID";
this.TextBoxSessionID.ReadOnly = true;
this.TextBoxSessionID.Size = new System.Drawing.Size(180, 20);
this.TextBoxSessionID.TabIndex = 1;
//
// label1
//
this.label1.AutoSize = true;
this.label1.Location = new System.Drawing.Point(36, 22);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(58, 13);
this.label1.TabIndex = 0;
this.label1.Text = "Session ID";
//
// groupBox2
//
this.groupBox2.Controls.Add(this.DataGridViewSession);
this.groupBox2.Dock = System.Windows.Forms.DockStyle.Fill;
this.groupBox2.Location = new System.Drawing.Point(5, 5);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(775, 228);
this.groupBox2.TabIndex = 1;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "Session List";
```

```
            //
            // DataGridViewSession
            //
            this.DataGridViewSession.AllowUserToAddRows = false;
            this.DataGridViewSession.AllowUserToDeleteRows = false;
            this.DataGridViewSession.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
            this.DataGridViewSession.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
            this.SessionIDTextBoxColumn,
            this.SessionNameTextBoxColumn,
            this.BeginDateTextBoxColumn,
            this.EndDateTextBoxColumn,
            this.TemplateNameTextBoxColumn,
            this.MethodologyTextBoxCloumn,
            this.NoOfParticipantTextBoxColumn,
            this.DeleteButtonColumn,
            this.CloseButtonColumn,
            this.StartButtonColumn,
            this.TemplateIDTextBoxColumn,
            this.ViewButtonColumn});
            this.DataGridViewSession.Dock = System.Windows.Forms.DockStyle.Fill;
            this.DataGridViewSession.Location = new System.Drawing.Point(3, 16);
            this.DataGridViewSession.MultiSelect = false;
            this.DataGridViewSession.Name = "DataGridViewSession";
            this.DataGridViewSession.ReadOnly = true;
            this.DataGridViewSession.RowHeadersVisible = false;
            this.DataGridViewSession.SelectionMode =
System.Windows.Forms.DataGridViewSelectionMode.FullRowSelect;
            this.DataGridViewSession.Size = new System.Drawing.Size(769, 209);
            this.DataGridViewSession.TabIndex = 0;
            this.DataGridViewSession.CellClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.DataGridViewSession_CellClick);
            this.DataGridViewSession.CellContentClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.DataGridViewSession_CellContentClic
k);
            //
            // errorProvider1
            //
            this.errorProvider1.ContainerControl = this;
            //
            // SessionIDTextBoxColumn
            //
            this.SessionIDTextBoxColumn.DataPropertyName = "SessionID";
            this.SessionIDTextBoxColumn.HeaderText = "Session ID";
            this.SessionIDTextBoxColumn.Name = "SessionIDTextBoxColumn";
            this.SessionIDTextBoxColumn.ReadOnly = true;
            //
            // SessionNameTextBoxColumn
            //
            this.SessionNameTextBoxColumn.DataPropertyName = "SessionName";
            this.SessionNameTextBoxColumn.HeaderText = "Session Name";
            this.SessionNameTextBoxColumn.Name = "SessionNameTextBoxColumn";
            this.SessionNameTextBoxColumn.ReadOnly = true;
            //
            // BeginDateTextBoxColumn
            //
```

```
this.BeginDateTextBoxColumn.DataPropertyName = "SessionBeginDate";
dataGridViewCellStyle1.Format = "dd-MM-yyyy";
dataGridViewCellStyle1.NullValue = null;
this.BeginDateTextBoxColumn.DefaultCellStyle = dataGridViewCellStyle1;
this.BeginDateTextBoxColumn.HeaderText = "Begin Date";
this.BeginDateTextBoxColumn.Name = "BeginDateTextBoxColumn";
this.BeginDateTextBoxColumn.ReadOnly = true;
//
// EndDateTextBoxColumn
//
this.EndDateTextBoxColumn.DataPropertyName = "SessionEndDate";
dataGridViewCellStyle2.Format = "dd-MM-yyyy";
dataGridViewCellStyle2.NullValue = null;
this.EndDateTextBoxColumn.DefaultCellStyle = dataGridViewCellStyle2;
this.EndDateTextBoxColumn.HeaderText = "End Date";
this.EndDateTextBoxColumn.Name = "EndDateTextBoxColumn";
this.EndDateTextBoxColumn.ReadOnly = true;
//
// TemplateNameTextBoxColumn
//
this.TemplateNameTextBoxColumn.DataPropertyName = "TemplateName";
this.TemplateNameTextBoxColumn.FillWeight = 120F;
this.TemplateNameTextBoxColumn.HeaderText = "Template Name";
this.TemplateNameTextBoxColumn.Name = "TemplateNameTextBoxColumn";
this.TemplateNameTextBoxColumn.ReadOnly = true;
this.TemplateNameTextBoxColumn.Width = 120;
//
// MethodologyTextBoxCloumn
//
this.MethodologyTextBoxCloumn.DataPropertyName = "Methodology";
this.MethodologyTextBoxCloumn.HeaderText = "Methodology";
this.MethodologyTextBoxCloumn.Name = "MethodologyTextBoxCloumn";
this.MethodologyTextBoxCloumn.ReadOnly = true;
//
// NoOfParticipantTextBoxColumn
//
this.NoOfParticipantTextBoxColumn.DataPropertyName = "NoOfParticipant";
this.NoOfParticipantTextBoxColumn.FillWeight = 135F;
this.NoOfParticipantTextBoxColumn.HeaderText = "Number Of Participant";
this.NoOfParticipantTextBoxColumn.Name = "NoOfParticipantTextBoxColumn";
this.NoOfParticipantTextBoxColumn.ReadOnly = true;
this.NoOfParticipantTextBoxColumn.Width = 135;
//
// DeleteButtonColumn
//
this.DeleteButtonColumn.HeaderText = "Delete";
this.DeleteButtonColumn.Name = "DeleteButtonColumn";
this.DeleteButtonColumn.ReadOnly = true;
this.DeleteButtonColumn.Text = "Delete";
this.DeleteButtonColumn.UseColumnTextForButtonValue = true;
//
// CloseButtonColumn
//
this.CloseButtonColumn.HeaderText = "Close Session";
this.CloseButtonColumn.Name = "CloseButtonColumn";
this.CloseButtonColumn.ReadOnly = true;
this.CloseButtonColumn.Text = "Close Session";
```

```
this.CloseButtonColumn.UseColumnTextForButtonValue = true;
//
// StartButtonColumn
//
this.StartButtonColumn.HeaderText = "Start Session";
this.StartButtonColumn.Name = "StartButtonColumn";
this.StartButtonColumn.ReadOnly = true;
this.StartButtonColumn.Text = "Start Session";
this.StartButtonColumn.UseColumnTextForButtonValue = true;
//
// TemplateIDTextBoxColumn
//
this.TemplateIDTextBoxColumn.DataPropertyName = "TemplateID";
this.TemplateIDTextBoxColumn.HeaderText = "Template ID";
this.TemplateIDTextBoxColumn.Name = "TemplateIDTextBoxColumn";
this.TemplateIDTextBoxColumn.ReadOnly = true;
this.TemplateIDTextBoxColumn.Visible = false;
//
// ViewButtonColumn
//
this.ViewButtonColumn.HeaderText = "View Session";
this.ViewButtonColumn.Name = "ViewButtonColumn";
this.ViewButtonColumn.ReadOnly = true;
this.ViewButtonColumn.Resizable = System.Windows.Forms.DataGridViewTriState.True;
this.ViewButtonColumn.Text = "View Session";
this.ViewButtonColumn.UseColumnTextForButtonValue = true;
//
// NewSession
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(785, 479);
this.ControlBox = false;
this.Controls.Add(this.groupBox2);
this.Controls.Add(this.groupBox1);
this.Name = "NewSession";
this.Padding = new System.Windows.Forms.Padding(5);
this.Text = "New Session";
this.Load += new System.EventHandler(this.NewSession_Load);
this.groupBox1.ResumeLayout(false);
this.groupBox1.PerformLayout();
this.groupBox3.ResumeLayout(false);
this.groupBox3.PerformLayout();
this.groupBox2.ResumeLayout(false);
((System.ComponentModel.ISupportInitialize)(this.DataGridViewSession)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.errorProvider1)).EndInit();
this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.GroupBox groupBox1;
private System.Windows.Forms.GroupBox groupBox2;
private System.Windows.Forms.DataGridView DataGridViewSession;
private System.Windows.Forms.TextBox TextBoxSessionID;
private System.Windows.Forms.Label label1;
```

```
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.TextBox TextBoxSessionName;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label7;
        private System.Windows.Forms.ComboBox ComboBoxMethodology;
        private System.Windows.Forms.TextBox TextBoxEndDate;
        private System.Windows.Forms.Label LabelEndDate;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.ComboBox ComboBoxTemplateName;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.ListBox ListBoxSelectedParticipant;
        private System.Windows.Forms.ListBox ListBoxAvailableParticipant;
        private System.Windows.Forms.Button ButtonUpdate;
        private System.Windows.Forms.Button ButtonCreate;
        private System.Windows.Forms.TextBox TextBoxNumberOfParticipant;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.DateTimePicker DateTimePickerBeginDate;
        private System.Windows.Forms.Label label10;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.ErrorProvider errorProvider1;
        private System.Windows.Forms.ComboBox ComboBoxFacilitator;
        private System.Windows.Forms.Label label6;
        private System.Windows.Forms.DataGridViewTextBoxColumn SessionIDTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn SessionNameTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn BeginDateTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn EndDateTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn TemplateNameTextBoxColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn MethodologyTextBoxCloumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn
NoOfParticipantTextBoxColumn;
        private System.Windows.Forms.DataGridViewButtonColumn DeleteButtonColumn;
        private System.Windows.Forms.DataGridViewButtonColumn CloseButtonColumn;
        private System.Windows.Forms.DataGridViewButtonColumn StartButtonColumn;
        private System.Windows.Forms.DataGridViewTextBoxColumn TemplateIDTextBoxColumn;
        private System.Windows.Forms.DataGridViewButtonColumn ViewButtonColumn;

    }
}
```

OpenSession.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;
```

```
namespace GroupStorm
{
   public partial class OpenSession : Form
   {
      public Word.ApplicationClass wordAppMain;
      public Word._Document wordDoc;
      public Word.Application wordApp;

      private DataSetBrain.TemplateDetailsDataTable templateDetailsDataTable = new
DataSetBrain.TemplateDetailsDataTable();
      private DataSetBrainTableAdapters.TemplateDetailsTableAdapter templateDetailsTableAdapter =
new GroupStorm.DataSetBrainTableAdapters.TemplateDetailsTableAdapter();
      private DataSetBrain.SessionsDataTable sessionsDataTable = new
DataSetBrain.SessionsDataTable();
      private DataSetBrainTableAdapters.SessionsTableAdapter sessionsTableAdapter = new
DataSetBrainTableAdapters.SessionsTableAdapter();
      private DataSetBrain.SessionParticipantsDataTable sessionParticipantsDataTable = new
DataSetBrain.SessionParticipantsDataTable();
      private DataSetBrainTableAdapters.SessionParticipantsTableAdapter
sessionParticipantsTableAdapter = new
DataSetBrainTableAdapters.SessionParticipantsTableAdapter();
      private DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
      private DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();

      KeyboardControl[] AllKeyboardControl = new KeyboardControl[13];
      LogInControl[] AllLogInControl = new LogInControl[13];

      private int _sessionID = 0;
      private string _status = "New";
      private string _methodology = "Anonymous";
      private int CurrentRow = 0;
      private int MaxRow = 0;
      private string _copyString = "";
      public bool forAll = false;
      public bool WindowWord = false;
      public string _votingText = "";
      public bool _vote = false;

      public OpenSession()
      {
         InitializeComponent();
      }

      private void OpenSession_Load(object sender, EventArgs e)
      {
         //Clear Status LogOn
         usersTableAdapter.ClearLogOn();

         // Get Session
         sessionsDataTable = sessionsTableAdapter.GetDataBy_SessionID(_sessionID);
         this.Text = sessionsDataTable.Rows[0]["SessionName"].ToString();
         int TemplateID = (int)sessionsDataTable.Rows[0]["TemplateID"];

         //Set default vote type
         cbxVoteType.SelectedIndex = 0;
```

```
            //Initial All Keyboard Control
            SetKeyboardControl();
            //Initial All Login Control
            SetLogInControl(_methodology);
            //Set Keyboard Combobox
            SetComboBoxKeyboard();

            // Get TemplateDetail
            if (TemplateID > 0)
            {
              templateDetailsDataTable =
templateDetailsTableAdapter.GetDataBy_TemplateID(TemplateID);
              MaxRow = templateDetailsDataTable.Rows.Count;
              this.LabelTemplateDetail.Text =
templateDetailsDataTable.Rows[CurrentRow]["TopicName"].ToString();

              //Get Last transaction data of this session
              DataSetBrain.SessionTransactionsDataTable STDataTable = new
DataSetBrain.SessionTransactionsDataTable();
              DataSetBrainTableAdapters.SessionTransactionsTableAdapter STTableAdapter = new
DataSetBrainTableAdapters.SessionTransactionsTableAdapter();
              STDataTable = STTableAdapter.GetDataBy_LastTemplateDetail(_sessionID);

              for (int i = 0; i < MaxRow; i++)
              {
                if (templateDetailsDataTable.Rows[i]["TemplateDetailID"].ToString() ==
STDataTable.Rows[0]["TemplateDetailID"].ToString())
                {
                  this.LabelTemplateDetail.Text =
templateDetailsDataTable.Rows[i]["TopicName"].ToString();
                  CurrentRow = i;
                  break;
                }
              }

              //Set Topic ID to All Keyboard Control
              foreach (KeyboardControl KB in AllKeyboardControl)
              {
                KB.TemplateDetailID =
Convert.ToInt32(templateDetailsDataTable.Rows[CurrentRow]["TemplateDetailID"]);
              }
            }

            //Free key
            if (TemplateID <= 0)
            {
              //Disable Login Control
              for (int i = 0; i < AllLogInControl.Length; i++)
              {
                AllLogInControl[i].Visible = false;
                AllLogInControl[i].Enabled = false;
              }

              //Disable Template
              this.LabelTemplateDetail.Visible = false;
              //Disable Previous & Next Button
              this.ButtonPrevious.Visible = false;
```

```
            this.ButtonNext.Visible = false;
        }

        // Use 1 mouse
        if (sdgManager1.Mice.Count > 1)
        {
            for (int i = 1; i < sdgManager1.Mice.Count; i++)
            {
                sdgManager1.Mice[i].Visible = false;
            }
        }

        //Set open word type (true:open word in new window, false:use same window)
        WindowWord = ((MDIParent1)this.MdiParent).GetWindowWord();
    }

    /// <summary>
    /// Quit this form
    /// </summary>
    private void button1_Click(object sender, EventArgs e)
    {
        //Open new session form
        MDIParent1 TempMDI = (MDIParent1)this.MdiParent;
        TempMDI.OpenNewSessionForm();

        //Close this form
        this.Close();
    }

    /// <summary>
    /// Press keyboard down
    /// </summary>
    private void sdgManager1_KeyDown(object sender, Sdgt.SdgKeyEventArgs e)
    {
        //Set used Keyboard ID
        int UseKeyboardID = e.ID;

        if (AllLogInControl[UseKeyboardID].Visible) //Login form
        {
            if (AllLogInControl[UseKeyboardID].FocusUserID)
            {
                //Focus userid box to used Keyboard ID
                AllLogInControl[UseKeyboardID].TextBoxUserID.Focus();
            }
            else
            {
                //Focus password box to used Keyboard ID
                AllLogInControl[UseKeyboardID].TextBoxPassword.Focus();
            }
        }
        else //Keyboard form
        {
            if (AllKeyboardControl[UseKeyboardID].Freeze) //Freeze Keyboard Control
            {
                this.TextBoxTemp.Focus();
            }
            else
```

```
{
  //Focus input box to used Keyboard ID
  AllKeyboardControl[UseKeyboardID].TextBoxInput.Focus();

  if (AllKeyboardControl[UseKeyboardID].farcilitor == false)
  {
    //if (this.ParentForm.MainMenuStrip.Focused == true)
    //{
    //   AllKeyboardControl[UseKeyboardID].TextBoxInput.Focus();
    //}

    //for (int i = 0; i < this.ParentForm.MainMenuStrip.Items.Count; i++)
    //{
    //   if (this.ParentForm.MainMenuStrip.Items[i].Selected == true)
    //   {
    //      AllKeyboardControl[UseKeyboardID].TextBoxInput.Focus();
    //      return;
    //   }
    //}

    switch (e.KeyCode)
    {
      #region "Press Esc"
      case Keys.Escape: //Press Esc
        try
        {
          //e.SuppressKeyPress = true;
          //e.Handled = false;
        }
        catch (Exception ex)
        {
          String err = ex.Message;
        }
        break;
      #endregion

      #region "Press Ctrl"
      case Keys.ControlKey: //Press Ctrl
        try
        {
          //e.SuppressKeyPress = true;
          //e.Handled = false;
          this.TextBoxTemp.Focus();
        }
        catch (Exception ex)
        {
          String err = ex.Message;
        }
        break;
      #endregion

      #region "Press Ctrl + "
      case Keys.Control: //Press Ctrl
        try
        {
          //e.SuppressKeyPress = true;
          //e.Handled = false;
```

```
                        this.TextBoxTemp.Focus();
                    }
                    catch (Exception ex)
                    {
                        String err = ex.Message;
                    }
                    break;
                #endregion

                #region "Press Alt"
                case Keys.Alt: //Press Alt
                    try
                    {
                        //e.SuppressKeyPress = true;
                        //e.Handled = false;
                        //this.TextBoxTemp.Focus();
                    }
                    catch (Exception ex)
                    {
                        String err = ex.Message;
                    }
                    break;
                #endregion
            }

            if (((Keys.Control & e.Modifiers) > 0))
            {
                this.TextBoxTemp.Focus();
            }
        }
    }

    if (e.KeyCode == Keys.Oemtilde)
    {
        if (AllLogInControl[UseKeyboardID].Visible)
        {
            AllLogInControl[UseKeyboardID].ChangeLanguage();
        }
        else
        {
            AllKeyboardControl[UseKeyboardID].ChangeLanguage();
        }
    }

    //Set default language for used Keyboard ID
    InputLanguage.CurrentInputLanguage =
InputLanguage.FromCulture(System.Globalization.CultureInfo.GetCultureInfo(AllKeyboardControl[U
seKeyboardID].LanguageCulture));
}

/// <summary>
/// press mouse down
/// </summary>
private void sdgManager1_MouseDown(object sender, Sdgt.SdgMouseEventArgs e)
{
```

```
        }

        /// <summary>
        /// Input Language Change
        /// </summary>
        private void OpenSession_InputLanguageChanged(object sender,
InputLanguageChangedEventArgs e)
        {
          //foreach (KeyboardControl KB in AllKeyboardControl)
          //{
          //   if (KB.TextBoxInput.Focused)
          //   {
          //      //Set Input Language to used keyboard control+
          //      KB.LanguageCulture = InputLanguage.CurrentInputLanguage.Culture.ToString();
          //      break;
          //   }
          //}
          //AllKeyboardControl[_UseKeyboardID].LanguageCulture =
InputLanguage.CurrentInputLanguage.Culture.ToString();
        }

        /// <summary>
        /// Open new Word
        /// </summary>
        private void ButtonWord_Click(object sender, EventArgs e)
        {
          object fileName = "c:\\" + this.Text + ".doc";
          object readOnly = false;
          object isVisible = true;
          object missing = System.Reflection.Missing.Value;

          if (sender != null)
          {
            MessageBox.Show("Please wait while the document is being displayed");
          }

          //Close Word
          try
          {
            objWinWordControl.CloseControl();

            object dummy = null;
            object dummy2 = (object)false;
            wordDoc.Close(ref dummy, ref dummy, ref dummy);
            // Change the line below.
            wordApp.Quit(ref dummy2, ref dummy, ref dummy);

          }
          catch { }
          finally
          {
            objWinWordControl.document = null;
            WinWordControl.WinWordControl.wd = null;
            WinWordControl.WinWordControl.wordWnd = 0;
          }

          try
```

```
            {
              //delete old file
              if (File.Exists(fileName.ToString()))
              {
                File.Delete(fileName.ToString());
              }

              //if (WindowWord)
              if (((MDIParent1)this.MdiParent).GetWindowWord())
              {
                //Open word in new window
                object file = "";
                object newTemplate = false;
                object docType = 0;
                isVisible = true;
                wordAppMain = new Word.ApplicationClass();
                wordDoc = wordAppMain.Documents.Add(ref file, ref newTemplate, ref docType, ref
isVisible);
                wordApp = wordDoc.Application;
                wordApp.Visible = true;
                //wordDoc.Activate();

                //focus this form
                ((MDIParent1)this.ParentForm).Activate();
              }
              else
              {
                //Open word in this window
                objWinWordControl.LoadDocument("");
                wordDoc = objWinWordControl.document;
                wordApp = wordDoc.Application;
              }

              try
              {
                //Display Menu Bar
                wordDoc.ActiveWindow.Application.CommandBars["Menu Bar"].Enabled = true;
              }
              catch (Exception ex)
              {
                Console.Write(ex.Message);
              }

              // Write Session Name
              wordApp.Selection.Font.Bold = 1;
              wordApp.Selection.TypeText(this.Text);
              wordApp.Selection.TypeText(Environment.NewLine);
              wordApp.Selection.Font.Bold = 0;

              //Get transaction data of this session
              DataSetBrain.SessionTransactionsDataTable STDataTable = new
DataSetBrain.SessionTransactionsDataTable();
              DataSetBrainTableAdapters.SessionTransactionsTableAdapter STTableAdapter = new
DataSetBrainTableAdapters.SessionTransactionsTableAdapter();
              STDataTable = STTableAdapter.GetDataBy_SessionID(_sessionID);

              string TempTemplateDetailName = "";
```

```
            if (_methodology == "Anonymous") //Methodology is Anonymous
            {
              foreach (DataRow myRow in STDataTable.Rows)
              {
                if (this.LabelTemplateDetail.Visible)
                {
                  DataRow[] dr = templateDetailsDataTable.Select("TemplateDetailID = " +
myRow["TemplateDetailID"].ToString());
                  if (dr.Length > 0)
                  {
                    //Display Topic Name
                    if (TempTemplateDetailName != dr[0]["TopicName"].ToString())
                    {
                      wordApp.Selection.Font.Bold = 1;
                      wordApp.Selection.TypeText(dr[0]["TopicName"].ToString());
                      wordApp.Selection.TypeText(Environment.NewLine);
                      TempTemplateDetailName = dr[0]["TopicName"].ToString();
                      wordApp.Selection.Font.Bold = 0;
                    }
                  }
                }

                //Display transaction text
                wordApp.Selection.TypeText(myRow["TransactionInput"].ToString());
                wordApp.Selection.TypeText(Environment.NewLine);
              }
            }
            else //Methodology is Synonymous
            {
              foreach (DataRow myRow in STDataTable.Rows)
              {
                if (this.LabelTemplateDetail.Visible)
                {
                  DataRow[] dr = templateDetailsDataTable.Select("TemplateDetailID = " +
myRow["TemplateDetailID"].ToString());
                  if (dr.Length > 0)
                  {
                    //Display Topic Name
                    if (TempTemplateDetailName != dr[0]["TopicName"].ToString())
                    {
                      wordApp.Selection.TypeText(dr[0]["TopicName"].ToString());
                      wordApp.Selection.TypeText(Environment.NewLine);
                      TempTemplateDetailName = dr[0]["TopicName"].ToString();
                    }
                  }
                }

                if (!string.IsNullOrEmpty(myRow["UserID"].ToString()))
                {
                  string Name = "";
                  //Get Name
                  DataSetBrain.UsersDataTable usersDataTable = new
DataSetBrain.UsersDataTable();
                  DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();
```

```
                    usersDataTable =
usersTableAdapter.GetDataBy_UserID(myRow["UserID"].ToString());

                    if (usersDataTable.Rows.Count > 0)
                    {
                        Name = usersDataTable.Rows[0]["Name"].ToString() + " " +
usersDataTable.Rows[0]["LastName"].ToString();
                    }

                    //Display transaction text
                    //Display owner
                    wordApp.Selection.TypeText(Name + " : " +
myRow["TransactionInput"].ToString());
                    wordApp.Selection.TypeText(Environment.NewLine);
                }
                else
                {
                    //Display transaction text
                    wordApp.Selection.TypeText(myRow["TransactionInput"].ToString());
                    wordApp.Selection.TypeText(Environment.NewLine);
                }
            }
        }

        //Display Topic Name
        if (this.LabelTemplateDetail.Visible)
        {
            wordApp.Selection.TypeText(LabelTemplateDetail.Text);
            wordApp.Selection.TypeText(Environment.NewLine);
        }

        // Save File
        wordDoc.SaveAs(ref fileName, ref missing, ref missing, ref missing, ref missing, ref
missing, ref readOnly, ref missing, ref missing, ref missing, ref missing);

        SetKeyboardInputNew();
      }
      catch (Exception ex) { String err = ex.Message; }
    }

    /// <summary>
    /// Open new Word
    /// </summary>
    public void KBControlOpenWord()
    {
      ButtonWord_Click(null, null);
    }

    /// <summary>
    /// Initial All Keyboard Control
    /// </summary>
    private void SetKeyboardControl()
    {
      //Count keyboard
      int KeyboardCount = this.sdgManager1.Keyboards.Count;
      //Set keyboard
      AllKeyboardControl[0] = this.keyboardControl0;
```

```
        AllKeyboardControl[1] = this.keyboardControl1;
        AllKeyboardControl[2] = this.keyboardControl2;
        AllKeyboardControl[3] = this.keyboardControl3;
        AllKeyboardControl[4] = this.keyboardControl4;
        AllKeyboardControl[5] = this.keyboardControl5;
        AllKeyboardControl[6] = this.keyboardControl6;
        AllKeyboardControl[7] = this.keyboardControl7;
        AllKeyboardControl[8] = this.keyboardControl8;
        AllKeyboardControl[9] = this.keyboardControl9;
        AllKeyboardControl[10] = this.keyboardControl10;
        AllKeyboardControl[11] = this.keyboardControl11;
        AllKeyboardControl[12] = this.keyboardControl12;

        //Set facilitator keyboard to first index
        KeyboardControl Temp = new KeyboardControl();
        Temp = AllKeyboardControl[0];
        for (int i = 0; i < KeyboardCount - 1; i++)
        {
          AllKeyboardControl[i] = AllKeyboardControl[i + 1];
        }
        AllKeyboardControl[KeyboardCount - 1] = Temp;

        //Set session id to all keyboard
        foreach (KeyboardControl KB in AllKeyboardControl)
        {
          KB.SessionID = _sessionID;
        }

        //Disable non use keyboard control
        for (int i = sdgManager1.Keyboards.Count; i < 13; i++)
        {
          AllKeyboardControl[i].Enabled = false;
        }

        //Set farcilitor to KeyboardControl
        AllKeyboardControl[KeyboardCount - 1].farcilitor = true;
        //Enable F11
        AllKeyboardControl[KeyboardCount - 1].EnableF11 = true;
        //Enable F12
        AllKeyboardControl[KeyboardCount - 1].EnableF12 = true;
    }

    /// <summary>
    /// Initial all login control
    /// </summary>
    private void SetLogInControl(string Methodology)
    {
        //Count keyboard
        int KeyboardCount = this.sdgManager1.Keyboards.Count;
        //Set all login control
        AllLogInControl[0] = this.logInControl0;
        AllLogInControl[1] = this.logInControl1;
        AllLogInControl[2] = this.logInControl2;
        AllLogInControl[3] = this.logInControl3;
        AllLogInControl[4] = this.logInControl4;
        AllLogInControl[5] = this.logInControl5;
        AllLogInControl[6] = this.logInControl6;
```

```
        AllLogInControl[7] = this.logInControl7;
        AllLogInControl[8] = this.logInControl8;
        AllLogInControl[9] = this.logInControl9;
        AllLogInControl[10] = this.logInControl10;
        AllLogInControl[11] = this.logInControl11;
        AllLogInControl[12] = this.logInControl12;

        //Set methodology to all login control
        for (int i = 0; i<AllLogInControl.Length; i++)
        {
            AllLogInControl[i].Methodology = Methodology;
        }

        //set facilitator login control to first index
        LogInControl Temp = new LogInControl();
        Temp = AllLogInControl[0];
        for (int i = 0; i < KeyboardCount - 1; i++)
        {
            AllLogInControl[i] = AllLogInControl[i + 1];
        }
        AllLogInControl[KeyboardCount - 1] = Temp;

        //Disable non use login control
        for (int i = sdgManager1.Keyboards.Count; i < 13; i++)
        {
            AllLogInControl[i].Enabled = false;
        }
    }

    /// <summary>
    /// Save and Close word when close this form
    /// </summary>
    private void OpenSession_FormClosing(object sender, FormClosingEventArgs e)
    {
        try
        {
            if (wordDoc != null)
            {
                //Save word
                wordDoc.Save();

                //Close Word
                object dummy = null;
                object dummy2 = (object)false;
                wordDoc.Close(ref dummy, ref dummy, ref dummy);
                wordApp.Quit(ref dummy2, ref dummy, ref dummy);
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }

    public int SessionID
    {
        set
```

```
            {
                _sessionID = value;
            }
            get
            {
                return _sessionID;
            }
        }

        public string Status
        {
            set
            {
                _status = value;
            }
            get
            {
                return _status;
            }
        }

        public string Methodology
        {
            set
            {
                _methodology = value;
            }
            get
            {
                return _methodology;
            }
        }

        /// <summary>
        /// Set all Keyboard to ComboBox
        /// except facilitator keyboard
        /// </summary>
        private void SetComboBoxKeyboard()
        {
            int KeyboardCount = this.sdgManager1.Keyboards.Count;
            this.ComboBoxKeyboard.Items.Clear();
            this.ComboBoxKeyboard.Sorted = true;

            this.ComboBoxKeyboard.Items.Add(new ItemObject("All", "All"));
            for (int i = 0; i < KeyboardCount - 1; i++)
            {
                if (AllKeyboardControl[i].Enabled)
                {
                    this.ComboBoxKeyboard.Items.Add(new
ItemObject(AllKeyboardControl[i].KeyboardName, i));
                }
            }

            this.ComboBoxKeyboard.SelectedIndex = 0;
        }

        /// <summary>
```

```
        /// Unfreeze keyboard
        /// </summary>
        private void ButtonEnable_Click(object sender, EventArgs e)
        {
          //Get selected keybord to unfreeze
          int KeyboardCount = this.sdgManager1.Keyboards.Count;
          ItemObject SelectedItem = (ItemObject)this.ComboBoxKeyboard.SelectedItem;
          string SelectedValue = SelectedItem.ValueOfKey.ToString();

          if (SelectedValue == "All") //Unfreeze all keyboard
          {
            for (int i = 0; i < KeyboardCount - 1; i++)
            {
              if (AllKeyboardControl[i].Freeze)
              {
                AllKeyboardControl[i].Freeze = false;
              }
            }

            //Set text Statusbar
            ((MDIParent1)this.MdiParent).SetStatus("all keyboard enable");
          }
          else
          {
            //Unfreeze selected keybord
            AllKeyboardControl[Convert.ToInt32(SelectedValue)].Freeze = false;

            //Set text Statusbar

((MDIParent1)this.MdiParent).SetStatus(AllKeyboardControl[Convert.ToInt32(SelectedValue)].Keyb
oardName.Trim() + " enable");
          }
        }

        /// <summary>
        /// Freeze keyboard
        /// </summary>
        private void ButtonFreeze_Click(object sender, EventArgs e)
        {
          //Get selected keybord to freeze
          int KeyboardCount = this.sdgManager1.Keyboards.Count;
          ItemObject SelectedItem = (ItemObject)this.ComboBoxKeyboard.SelectedItem;
          string SelectedValue = SelectedItem.ValueOfKey.ToString();

          if (SelectedValue == "All") //Freeze all keyboard
          {
            for (int i = 0; i < KeyboardCount - 1; i++)
            {
              if (AllKeyboardControl[i].Enabled)
              {
                AllKeyboardControl[i].Freeze = true;
              }
            }

            //Set text Statusbar
            ((MDIParent1)this.MdiParent).SetStatus("all keyboard freezing");
          }
```

```
            else
            {
              //Freeze selected keybord
              AllKeyboardControl[Convert.ToInt32(SelectedValue)].Freeze = true;

              //Set text Statusbar

((MDIParent1)this.MdiParent).SetStatus(AllKeyboardControl[Convert.ToInt32(SelectedValue)].Keyb
oardName.Trim() + " freezing");
            }
        }

        /// <summary>
        /// Select provious topic
        /// </summary>
        private void ButtonPrevious_Click(object sender, EventArgs e)
        {
          if (CurrentRow > 0) //Not first topic
          {
            CurrentRow--;
            //Set topic name
            this.LabelTemplateDetail.Text =
templateDetailsDataTable.Rows[CurrentRow]["TopicName"].ToString();

            //Set topic id to all keyboard
            foreach (KeyboardControl kb in AllKeyboardControl)
            {
              kb.TemplateDetailID =
Convert.ToInt32(templateDetailsDataTable.Rows[CurrentRow]["TemplateDetailID"].ToString());
            }

            if (wordDoc != null) //Word is open
            {
              try
              {
                Object oConst1 = Word.WdGoToItem.wdGoToLine;
                Object oConst2 = Word.WdGoToDirection.wdGoToLast;
                Object oMissing = System.Reflection.Missing.Value;

                //Goto last line
                wordApp.Selection.GoTo(ref oConst1, ref oConst2, ref oMissing, ref oMissing);

                wordApp.Selection.TypeText("----------------------------------------");
                wordApp.Selection.TypeText(Environment.NewLine);

                //Display Topic Name
                wordApp.Selection.Font.Bold = 1;
                wordApp.Selection.TypeText(LabelTemplateDetail.Text);
                wordApp.Selection.TypeText(Environment.NewLine);
                wordApp.Selection.Font.Bold = 0;
              }
              catch (Exception ex)
              {
                Console.WriteLine(ex.Message);
              }
            }
          }
```

```
        }

        /// <summary>
        /// Select next topic
        /// </summary>
        private void ButtonNext_Click(object sender, EventArgs e)
        {
          if (CurrentRow < (MaxRow - 1)) //not last topic
          {
            CurrentRow++;
            //Display topic name
            this.LabelTemplateDetail.Text =
  templateDetailsDataTable.Rows[CurrentRow]["TopicName"].ToString();
            //set topic id to all keyboard
            foreach (KeyboardControl kb in AllKeyboardControl)
            {
              kb.TemplateDetailID =
  Convert.ToInt32(templateDetailsDataTable.Rows[CurrentRow]["TemplateDetailID"].ToString());
            }

            if (wordDoc != null) //Word is open
            {
              try
              {
                Object oConst1 = Word.WdGoToItem.wdGoToLine;
                Object oConst2 = Word.WdGoToDirection.wdGoToLast;
                Object oMissing = System.Reflection.Missing.Value;

                //Goto last line
                wordApp.Selection.GoTo(ref oConst1, ref oConst2, ref oMissing, ref oMissing);

                wordApp.Selection.TypeText("----------------------------------------");
                wordApp.Selection.TypeText(Environment.NewLine);

                //Display topic name
                wordApp.Selection.Font.Bold = 1;
                wordApp.Selection.TypeText(LabelTemplateDetail.Text);
                wordApp.Selection.TypeText(Environment.NewLine);
                wordApp.Selection.Font.Bold = 0;
              }
              catch (Exception ex)
              {
                Console.WriteLine(ex.Message);
              }
            }
          }
        }

        /// <summary>
        /// Open old word document
        /// </summary>
        private void ButtonOpenWord_Click(object sender, EventArgs e)
        {
          //Display Dialog to open word document
          OpenFileDialog openFileDialog = new OpenFileDialog();
          openFileDialog.InitialDirectory =
  Environment.GetFolderPath(Environment.SpecialFolder.Personal);
```

```
openFileDialog.Filter = "Word Files (*.doc)|*.doc|All Files (*.*)|*.*";
if (openFileDialog.ShowDialog(this) == DialogResult.OK)
{
  string FileName = openFileDialog.FileName;
  // TODO: Add code here to open the file.
  MessageBox.Show("Please wait while the document is being displayed");

  //Close word
  try
  {
    objWinWordControl.CloseControl();

    object dummy = null;
    object dummy2 = (object)false;
    wordDoc.Close(ref dummy, ref dummy, ref dummy);
    // Change the line below.
    wordApp.Quit(ref dummy2, ref dummy, ref dummy);
  }
  catch { }
  finally
  {
    objWinWordControl.document = null;
    WinWordControl.WinWordControl.wd = null;
    WinWordControl.WinWordControl.wordWnd = 0;
  }

 try
  {
    object missing = System.Reflection.Missing.Value;
    //object fileName = "normal.doc";
    object fileName = FileName;
    object newTemplate = false;
    object docType = 0;
    object isVisible = true;
    object readOnly = false;

    //Open word
    //if (WindowWord)
    if (((MDIParent1)this.MdiParent).GetWindowWord())
    {
      //Open word on new window
      wordApp = new Word.ApplicationClass();
      wordDoc = wordApp.Documents.Add(ref fileName, ref newTemplate, ref docType, ref
isVisible);
    }
    else
    {
      //Open word on this window
      objWinWordControl.LoadDocument(FileName);
      wordDoc = objWinWordControl.document;
      wordApp = wordDoc.Application;
    }

    try
    {
      //Display menu bar
      wordDoc.ActiveWindow.Application.CommandBars["Menu Bar"].Enabled = true;
```

```
            }
            catch (Exception ex)
            {
               Console.Write(ex.Message);
            }

            //Save word document
            wordDoc.SaveAs(ref fileName, ref missing, ref missing, ref missing, ref missing, ref
missing, ref readOnly, ref missing, ref missing, ref missing, ref missing);

            //Set new message of all display box to last line
            SetKeyboardInputNew();
         }
         catch (Exception ex) { String err = ex.Message; }
      }
   }

   /// <summary>
   /// Set new message of all display box to last line
   /// </summary>
   private void SetKeyboardInputNew()
   {
      for (int i = 0; i < AllKeyboardControl.Length; i++)
      {
         AllKeyboardControl[i].SetNewTextLast();
      }
   }

   /// <summary>
   /// Set Copy text
   /// </summary>
   public string CopyString
   {
      get
      {
         return _copyString;
      }

      set
      {
         _copyString = value;
      }

   }

   /// <summary>
   /// paste text to all display box
   /// </summary>
   public void PasteString()
   {
      foreach (KeyboardControl KB in AllKeyboardControl)
      {
         if (KB.Enabled)
         {
            //Paste text to display box
            KB.pasteToolStripMenuItem_Click(null, null);
         }
```

```
    }
  }

  private void chbxF11ForAll_MouseDown(object sender, MouseEventArgs e)
  {
    chbxF11ForAll.Checked = !chbxF11ForAll.Checked;

    forAll = chbxF11ForAll.Checked;
  }

  private void ButtonEnable_MouseDown(object sender, MouseEventArgs e)
  {
    ButtonEnable_Click(null, null);
  }

  private void ButtonFreeze_MouseDown(object sender, MouseEventArgs e)
  {
    ButtonFreeze_Click(null, null);
  }

  private void ButtonPrevious_MouseDown(object sender, MouseEventArgs e)
  {
    ButtonPrevious_Click(null, null);
  }

  private void ButtonNext_MouseDown(object sender, MouseEventArgs e)
  {
    ButtonNext_Click(null, null);
  }

  private void ButtonOpenWord_MouseDown(object sender, MouseEventArgs e)
  {
    ButtonOpenWord_Click(null, null);
  }

  private void ButtonWord_MouseDown(object sender, MouseEventArgs e)
  {
    ButtonWord_Click(null, null);
  }

  private void button1_MouseDown(object sender, MouseEventArgs e)
  {
    button1_Click(null, null);
  }

  /// <summary>
  /// Do F11 function to all keyboard
  /// </summary>
  public void DoF11()
  {
    //Count keyboard
    int KeyboardCount = this.sdgManager1.Keyboards.Count;
    AllKeyboardControl[KeyboardCount - 1].DoF11();

    for (int i = 0; i < AllKeyboardControl.Length; i++)
    {
      if (AllKeyboardControl[i].Enabled)
```

```
            {
                AllKeyboardControl[i].DoF11();
            }
        }
    }

    /// <summary>
    /// Vote Start
    /// </summary>
    private void btnVote_MouseDown(object sender, MouseEventArgs e)
    {
        // Do F11 function to all keyboard
        DoF11();

        if (wordDoc == null)
        {
            //Open Word
            KBControlOpenWord();
        }

        //Object oConst1 = Word.WdGoToItem.wdGoToLine;
        //Object oConst2 = Word.WdGoToDirection.wdGoToLast;
        //Object oMissing = System.Reflection.Missing.Value;

        //Goto last line
        //wordApp.Selection.GoTo(ref oConst1, ref oConst2, ref oMissing, ref oMissing);
        //Display 'Vote'
        //wordApp.Selection.TypeText("Vote");
        //wordApp.Selection.TypeText(Environment.NewLine);

        //Save Start Vote Message
        AllKeyboardControl[0].TextBoxInput.Text = "Vote";
        AllKeyboardControl[0].DoF11();

        //if Vote type is "Multiple Choice" or "Yes/No", all participant can press only 1 charactor
        if ((cbxVoteType.SelectedItem.ToString() == "Multiple Choice") ||
(cbxVoteType.SelectedItem.ToString() == "Yes/No"))
        {
            foreach(KeyboardControl kb in AllKeyboardControl)
            {
                if (!kb.farcilitor)// is participant
                {
                    //Set input box to keep 1 character
                    kb.TextBoxInput.MaxLength = 1;
                }
            }
        }

        //Disable Vote type
        cbxVoteType.Enabled = false;
        //Disable start vote button
        btnVote.Enabled = false;
        //Disable Previous & Next Topic button
        ButtonPrevious.Enabled = false;
        ButtonNext.Enabled = false;

        //Set text Statusbar
```

```
    ((MDIParent1)this.MdiParent).SetStatus("voting");

    //Set Vote Text
    _votingText = Environment.NewLine + "Vote";

    //vote status
    _vote = true;
}

/// <summary>
/// Stop Vote
/// </summary>
private void btnStopVote_MouseDown(object sender, MouseEventArgs e)
{
    if (_vote)
    {
        // Do F11 function to all keyboard
        DoF11();

        //if (wordDoc == null)
        //{
        //    //Open Word
        //    KBControlOpenWord();
        //}

        //Object oConst1 = Word.WdGoToItem.wdGoToLine;
        //Object oConst2 = Word.WdGoToDirection.wdGoToLast;
        //Object oMissing = System.Reflection.Missing.Value;

        ////Goto last line
        //wordApp.Selection.GoTo(ref oConst1, ref oConst2, ref oMissing, ref oMissing);
        ////Display 'Vote'
        //wordApp.Selection.TypeText("Stop Vote");
        //wordApp.Selection.TypeText(Environment.NewLine);

        //Save Stop Vote Message
        AllKeyboardControl[0].TextBoxInput.Text = "Stop Vote";
        AllKeyboardControl[0].DoF11();

        //if Vote type is "Multiple Choice" or "Yes/No", all participant can press more character
        if ((cbxVoteType.Text == "Multiple Choice") || (cbxVoteType.Text == "Yes/No"))
        {
            foreach (KeyboardControl kb in AllKeyboardControl)
            {
                //Set input box to keep more character
                kb.TextBoxInput.MaxLength = 0;
            }
        }

        //Enable vote type
        cbxVoteType.Enabled = true;
        //Enable Start vote button
        btnVote.Enabled = true;
        //Enable Previous & Next topic button
        ButtonPrevious.Enabled = true;
        ButtonNext.Enabled = true;
```

```
        {
            ButtonFreeze_Click(null, null);
        }

        public void ShortCutPrevious()
        {
            ButtonPrevious_Click(null, null);
        }

        public void ShortCutNext()
        {
            ButtonNext_Click(null, null);
        }

        public void ShortCutNewWord()
        {
            ButtonWord_Click(null, null);
        }

        public void ShortCutOpenWord()
        {
            ButtonOpenWord_Click(null, null);
        }

        public void ShortCutQuit()
        {
            button1_Click(null, null);
        }
    }
}
```

OpenSession.Designer.cs

```
namespace GroupStorm
{
    partial class OpenSession
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
```

```
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
    this.components = new System.ComponentModel.Container();
    this.sdgManager1 = new Sdgt.SdgManager(this.components);
    this.splitContainer1 = new System.Windows.Forms.SplitContainer();
    this.btnStopVote = new System.Windows.Forms.Button();
    this.btnVote = new System.Windows.Forms.Button();
    this.cbxVoteType = new System.Windows.Forms.ComboBox();
    this.chbxF11ForAll = new System.Windows.Forms.CheckBox();
    this.ButtonOpenWord = new System.Windows.Forms.Button();
    this.LabelTemplateDetail = new System.Windows.Forms.Label();
    this.ButtonNext = new System.Windows.Forms.Button();
    this.ButtonPrevious = new System.Windows.Forms.Button();
    this.ComboBoxKeyboard = new System.Windows.Forms.ComboBox();
    this.ButtonFreeze = new System.Windows.Forms.Button();
    this.ButtonEnable = new System.Windows.Forms.Button();
    this.button1 = new System.Windows.Forms.Button();
    this.ButtonWord = new System.Windows.Forms.Button();
    this.TextBoxTemp = new System.Windows.Forms.TextBox();
    this.objWinWordControl = new WinWordControl.WinWordControl();
    this.logInControl12 = new GroupStorm.LogInControl();
    this.keyboardControl12 = new GroupStorm.KeyboardControl();
    this.logInControl11 = new GroupStorm.LogInControl();
    this.keyboardControl11 = new GroupStorm.KeyboardControl();
    this.logInControl10 = new GroupStorm.LogInControl();
    this.keyboardControl10 = new GroupStorm.KeyboardControl();
    this.logInControl9 = new GroupStorm.LogInControl();
    this.keyboardControl9 = new GroupStorm.KeyboardControl();
    this.logInControl8 = new GroupStorm.LogInControl();
    this.keyboardControl8 = new GroupStorm.KeyboardControl();
    this.logInControl7 = new GroupStorm.LogInControl();
    this.keyboardControl7 = new GroupStorm.KeyboardControl();
    this.logInControl6 = new GroupStorm.LogInControl();
    this.keyboardControl6 = new GroupStorm.KeyboardControl();
    this.logInControl5 = new GroupStorm.LogInControl();
    this.keyboardControl5 = new GroupStorm.KeyboardControl();
    this.logInControl4 = new GroupStorm.LogInControl();
    this.keyboardControl4 = new GroupStorm.KeyboardControl();
    this.logInControl3 = new GroupStorm.LogInControl();
    this.keyboardControl3 = new GroupStorm.KeyboardControl();
    this.logInControl2 = new GroupStorm.LogInControl();
    this.keyboardControl2 = new GroupStorm.KeyboardControl();
    this.logInControl1 = new GroupStorm.LogInControl();
    this.keyboardControl1 = new GroupStorm.KeyboardControl();
    this.logInControl0 = new GroupStorm.LogInControl();
    this.keyboardControl0 = new GroupStorm.KeyboardControl();
    this.splitContainer1.Panel1.SuspendLayout();
    this.splitContainer1.Panel2.SuspendLayout();
```

```
            this.splitContainer1.SuspendLayout();
            this.SuspendLayout();
            //
            // sdgManager1
            //
            this.sdgManager1.EmulateSystemMouseMode =
Sdgt.EmulateSystemMouseModes.FollowMouse;
            this.sdgManager1.Keyboards = null;
            this.sdgManager1.Mice = null;
            this.sdgManager1.MouseToFollow = 0;
            this.sdgManager1.ParkSystemMouseLocation = new System.Drawing.Point(350, 350);
            this.sdgManager1.RelativeTo = this;
            this.sdgManager1.KeyDown += new
Sdgt.SdgKeyEventHandler(this.sdgManager1_KeyDown);
            this.sdgManager1.MouseDown += new
Sdgt.SdgMouseEventHandler(this.sdgManager1_MouseDown);
            //
            // splitContainer1
            //
            this.splitContainer1.Dock = System.Windows.Forms.DockStyle.Fill;
            this.splitContainer1.FixedPanel = System.Windows.Forms.FixedPanel.Panel1;
            this.splitContainer1.Location = new System.Drawing.Point(0, 0);
            this.splitContainer1.Name = "splitContainer1";
            //
            // splitContainer1.Panel1
            //
            this.splitContainer1.Panel1.Controls.Add(this.btnStopVote);
            this.splitContainer1.Panel1.Controls.Add(this.btnVote);
            this.splitContainer1.Panel1.Controls.Add(this.cbxVoteType);
            this.splitContainer1.Panel1.Controls.Add(this.chbxF11ForAll);
            this.splitContainer1.Panel1.Controls.Add(this.ButtonOpenWord);
            this.splitContainer1.Panel1.Controls.Add(this.LabelTemplateDetail);
            this.splitContainer1.Panel1.Controls.Add(this.ButtonNext);
            this.splitContainer1.Panel1.Controls.Add(this.ButtonPrevious);
            this.splitContainer1.Panel1.Controls.Add(this.ComboBoxKeyboard);
            this.splitContainer1.Panel1.Controls.Add(this.ButtonFreeze);
            this.splitContainer1.Panel1.Controls.Add(this.ButtonEnable);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl12);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl11);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl10);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl9);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl8);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl7);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl6);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl5);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl4);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl3);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl2);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl1);
            this.splitContainer1.Panel1.Controls.Add(this.logInControl0);
            this.splitContainer1.Panel1.Controls.Add(this.button1);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl0);
            this.splitContainer1.Panel1.Controls.Add(this.ButtonWord);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl11);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl12);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl9);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl10);
```

```
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl7);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl8);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl5);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl6);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl3);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl4);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl2);
            this.splitContainer1.Panel1.Controls.Add(this.keyboardControl1);
            this.splitContainer1.Panel1.Controls.Add(this.TextBoxTemp);
            this.splitContainer1.Panel1MinSize = 612;
            //
            // splitContainer1.Panel2
            //
            this.splitContainer1.Panel2.Controls.Add(this.objWinWordControl);
            this.splitContainer1.Panel2MinSize = 0;
            this.splitContainer1.Size = new System.Drawing.Size(1016, 688);
            this.splitContainer1.SplitterDistance = 612;
            this.splitContainer1.SplitterWidth = 1;
            this.splitContainer1.TabIndex = 9;
            //
            // btnStopVote
            //
            this.btnStopVote.Location = new System.Drawing.Point(567, 0);
            this.btnStopVote.Name = "btnStopVote";
            this.btnStopVote.Size = new System.Drawing.Size(42, 23);
            this.btnStopVote.TabIndex = 37;
            this.btnStopVote.TabStop = false;
            this.btnStopVote.Text = "Stop";
            this.btnStopVote.UseVisualStyleBackColor = true;
            this.btnStopVote.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.btnStopVote_MouseDown);
            //
            // btnVote
            //
            this.btnVote.Location = new System.Drawing.Point(519, 0);
            this.btnVote.Name = "btnVote";
            this.btnVote.Size = new System.Drawing.Size(42, 23);
            this.btnVote.TabIndex = 36;
            this.btnVote.TabStop = false;
            this.btnVote.Text = "Vote";
            this.btnVote.UseVisualStyleBackColor = true;
            this.btnVote.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.btnVote_MouseDown);
            //
            // cbxVoteType
            //
            this.cbxVoteType.DropDownStyle = System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.cbxVoteType.FormattingEnabled = true;
            this.cbxVoteType.Items.AddRange(new object[] {
            "Text",
            "Multiple Choice",
            "Yes/No",
            "Number"});
            this.cbxVoteType.Location = new System.Drawing.Point(387, 2);
            this.cbxVoteType.Name = "cbxVoteType";
            this.cbxVoteType.Size = new System.Drawing.Size(126, 21);
            this.cbxVoteType.TabIndex = 35;
```

```
            this.cbxVoteType.TabStop = false;
            //
            // chbxF11ForAll
            //
            this.chbxF11ForAll.AutoSize = true;
            this.chbxF11ForAll.Location = new System.Drawing.Point(309, 4);
            this.chbxF11ForAll.Name = "chbxF11ForAll";
            this.chbxF11ForAll.Size = new System.Drawing.Size(72, 17);
            this.chbxF11ForAll.TabIndex = 34;
            this.chbxF11ForAll.TabStop = false;
            this.chbxF11ForAll.Text = "F11 for all";
            this.chbxF11ForAll.UseVisualStyleBackColor = true;
            this.chbxF11ForAll.MouseDown += new
    System.Windows.Forms.MouseEventHandler(this.chbxF11ForAll_MouseDown);
            //
            // ButtonOpenWord
            //
            this.ButtonOpenWord.Location = new System.Drawing.Point(372, 72);
            this.ButtonOpenWord.Name = "ButtonOpenWord";
            this.ButtonOpenWord.Size = new System.Drawing.Size(75, 23);
            this.ButtonOpenWord.TabIndex = 32;
            this.ButtonOpenWord.TabStop = false;
            this.ButtonOpenWord.Text = "Open Word";
            this.ButtonOpenWord.UseVisualStyleBackColor = true;
            this.ButtonOpenWord.MouseDown += new
    System.Windows.Forms.MouseEventHandler(this.ButtonOpenWord_MouseDown);
            //
            // LabelTemplateDetail
            //
            this.LabelTemplateDetail.AutoSize = true;
            this.LabelTemplateDetail.Location = new System.Drawing.Point(318, 51);
            this.LabelTemplateDetail.Name = "LabelTemplateDetail";
            this.LabelTemplateDetail.Size = new System.Drawing.Size(35, 13);
            this.LabelTemplateDetail.TabIndex = 31;
            this.LabelTemplateDetail.Text = "label1";
            //
            // ButtonNext
            //
            this.ButtonNext.Location = new System.Drawing.Point(534, 46);
            this.ButtonNext.Name = "ButtonNext";
            this.ButtonNext.Size = new System.Drawing.Size(75, 23);
            this.ButtonNext.TabIndex = 30;
            this.ButtonNext.TabStop = false;
            this.ButtonNext.Text = ">";
            this.ButtonNext.UseVisualStyleBackColor = true;
            this.ButtonNext.MouseDown += new
      System.Windows.Forms.MouseEventHandler(this.ButtonNext_MouseDown);
            //
            // ButtonPrevious
            //
            this.ButtonPrevious.Location = new System.Drawing.Point(453, 46);
            this.ButtonPrevious.Name = "ButtonPrevious";
            this.ButtonPrevious.Size = new System.Drawing.Size(75, 23);
            this.ButtonPrevious.TabIndex = 29;
            this.ButtonPrevious.TabStop = false;
            this.ButtonPrevious.Text = "<";
            this.ButtonPrevious.UseVisualStyleBackColor = true;
```

```
            this.ButtonPrevious.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.ButtonPrevious_MouseDown);
            //
            // ComboBoxKeyboard
            //
            this.ComboBoxKeyboard.DropDownStyle =
System.Windows.Forms.ComboBoxStyle.DropDownList;
            this.ComboBoxKeyboard.FormattingEnabled = true;
            this.ComboBoxKeyboard.Location = new System.Drawing.Point(309, 22);
            this.ComboBoxKeyboard.Name = "ComboBoxKeyboard";
            this.ComboBoxKeyboard.Size = new System.Drawing.Size(138, 21);
            this.ComboBoxKeyboard.Sorted = true;
            this.ComboBoxKeyboard.TabIndex = 28;
            this.ComboBoxKeyboard.TabStop = false;
            //
            // ButtonFreeze
            //
            this.ButtonFreeze.Location = new System.Drawing.Point(534, 22);
            this.ButtonFreeze.Name = "ButtonFreeze";
            this.ButtonFreeze.Size = new System.Drawing.Size(75, 23);
            this.ButtonFreeze.TabIndex = 27;
            this.ButtonFreeze.TabStop = false;
            this.ButtonFreeze.Text = "Freeze";
            this.ButtonFreeze.UseVisualStyleBackColor = true;
            this.ButtonFreeze.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.ButtonFreeze_MouseDown);
            //
            // ButtonEnable
            //
            this.ButtonEnable.Location = new System.Drawing.Point(453, 22);
            this.ButtonEnable.Name = "ButtonEnable";
            this.ButtonEnable.Size = new System.Drawing.Size(75, 23);
            this.ButtonEnable.TabIndex = 26;
            this.ButtonEnable.TabStop = false;
            this.ButtonEnable.Text = "Enable";
            this.ButtonEnable.UseVisualStyleBackColor = true;
            this.ButtonEnable.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.ButtonEnable_MouseDown);
            //
            // button1
            //
            this.button1.Location = new System.Drawing.Point(534, 72);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(75, 23);
            this.button1.TabIndex = 3;
            this.button1.TabStop = false;
            this.button1.Text = "Quit";
            this.button1.UseVisualStyleBackColor = true;
            this.button1.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.button1_MouseDown);
            //
            // ButtonWord
            //
            this.ButtonWord.Location = new System.Drawing.Point(453, 72);
            this.ButtonWord.Name = "ButtonWord";
            this.ButtonWord.Size = new System.Drawing.Size(75, 23);
            this.ButtonWord.TabIndex = 7;
```

```
            this.ButtonWord.TabStop = false;
            this.ButtonWord.Text = "New Word";
            this.ButtonWord.UseVisualStyleBackColor = true;
            this.ButtonWord.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.ButtonWord_MouseDown);
            //
            // TextBoxTemp
            //
            this.TextBoxTemp.Location = new System.Drawing.Point(372, 72);
            this.TextBoxTemp.Name = "TextBoxTemp";
            this.TextBoxTemp.Size = new System.Drawing.Size(75, 20);
            this.TextBoxTemp.TabIndex = 33;
            this.TextBoxTemp.TabStop = false;
            //
            // objWinWordControl
            //
            this.objWinWordControl.BackColor = System.Drawing.SystemColors.Highlight;
            this.objWinWordControl.Dock = System.Windows.Forms.DockStyle.Fill;
            this.objWinWordControl.Location = new System.Drawing.Point(0, 0);
            this.objWinWordControl.Name = "objWinWordControl";
            this.objWinWordControl.Size = new System.Drawing.Size(403, 688);
            this.objWinWordControl.TabIndex = 0;
            //
            // logInControl12
            //
            this.logInControl12.FocusUserID = true;
            this.logInControl12.KeyboardControl = this.keyboardControl12;
            this.logInControl12.KeyboardName = "Keyboard 12";
            this.logInControl12.LanguageCulture = "en-US";
            this.logInControl12.Location = new System.Drawing.Point(307, 578);
            this.logInControl12.Methodology = "Synonymous";
            this.logInControl12.Name = "logInControl12";
            this.logInControl12.Size = new System.Drawing.Size(300, 90);
            this.logInControl12.TabIndex = 25;
            //
            // keyboardControl12
            //
            this.keyboardControl12.EnableF11 = false;
            this.keyboardControl12.EnableF12 = false;
            this.keyboardControl12.Freeze = false;
            this.keyboardControl12.KeyboardID = "";
            this.keyboardControl12.KeyboardName = "Keyboard 12";
            this.keyboardControl12.LanguageCulture = "en-US";
            this.keyboardControl12.Location = new System.Drawing.Point(307, 578);
            this.keyboardControl12.Name = "keyboardControl12";
            this.keyboardControl12.SessionID = 0;
            this.keyboardControl12.Size = new System.Drawing.Size(300, 90);
            this.keyboardControl12.TabIndex = 12;
            this.keyboardControl12.TemplateDetailID = 0;
            //
            // logInControl11
            //
            this.logInControl11.FocusUserID = true;
            this.logInControl11.KeyboardControl = this.keyboardControl11;
            this.logInControl11.KeyboardName = "Keyboard 11";
            this.logInControl11.LanguageCulture = "en-US";
            this.logInControl11.Location = new System.Drawing.Point(3, 578);
```

```
this.logInControl11.Methodology = "Synonymous";
this.logInControl11.Name = "logInControl11";
this.logInControl11.Size = new System.Drawing.Size(300, 90);
this.logInControl11.TabIndex = 24;
//
// keyboardControl11
//
this.keyboardControl11.EnableF11 = false;
this.keyboardControl11.EnableF12 = false;
this.keyboardControl11.Freeze = false;
this.keyboardControl11.KeyboardID = "";
this.keyboardControl11.KeyboardName = "Keyboard 11";
this.keyboardControl11.LanguageCulture = "en-US";
this.keyboardControl11.Location = new System.Drawing.Point(3, 578);
this.keyboardControl11.Name = "keyboardControl11";
this.keyboardControl11.SessionID = 0;
this.keyboardControl11.Size = new System.Drawing.Size(300, 90);
this.keyboardControl11.TabIndex = 11;
this.keyboardControl11.TemplateDetailID = 0;
//
// logInControl10
//
this.logInControl10.FocusUserID = true;
this.logInControl10.KeyboardControl = this.keyboardControl10;
this.logInControl10.KeyboardName = "Keyboard 10";
this.logInControl10.LanguageCulture = "en-US";
this.logInControl10.Location = new System.Drawing.Point(307, 482);
this.logInControl10.Methodology = "Synonymous";
this.logInControl10.Name = "logInControl10";
this.logInControl10.Size = new System.Drawing.Size(300, 90);
this.logInControl10.TabIndex = 23;
//
// keyboardControl10
//
this.keyboardControl10.EnableF11 = false;
this.keyboardControl10.EnableF12 = false;
this.keyboardControl10.Freeze = false;
this.keyboardControl10.KeyboardID = "";
this.keyboardControl10.KeyboardName = "Keyboard 10";
this.keyboardControl10.LanguageCulture = "en-US";
this.keyboardControl10.Location = new System.Drawing.Point(307, 482);
this.keyboardControl10.Name = "keyboardControl10";
this.keyboardControl10.SessionID = 0;
this.keyboardControl10.Size = new System.Drawing.Size(300, 90);
this.keyboardControl10.TabIndex = 10;
this.keyboardControl10.TemplateDetailID = 0;
//
// logInControl9
//
this.logInControl9.FocusUserID = true;
this.logInControl9.KeyboardControl = this.keyboardControl9;
this.logInControl9.KeyboardName = "Keyboard 09";
this.logInControl9.LanguageCulture = "en-US";
this.logInControl9.Location = new System.Drawing.Point(3, 482);
this.logInControl9.Methodology = "Synonymous";
this.logInControl9.Name = "logInControl9";
this.logInControl9.Size = new System.Drawing.Size(300, 90);
```

```
this.logInControl9.TabIndex = 22;
//
// keyboardControl9
//
this.keyboardControl9.EnableF11 = false;
this.keyboardControl9.EnableF12 = false;
this.keyboardControl9.Freeze = false;
this.keyboardControl9.KeyboardID = "";
this.keyboardControl9.KeyboardName = "Keyboard 09";
this.keyboardControl9.LanguageCulture = "en-US";
this.keyboardControl9.Location = new System.Drawing.Point(3, 482);
this.keyboardControl9.Name = "keyboardControl9";
this.keyboardControl9.SessionID = 0;
this.keyboardControl9.Size = new System.Drawing.Size(300, 90);
this.keyboardControl9.TabIndex = 9;
this.keyboardControl9.TemplateDetailID = 0;
//
// logInControl8
//
this.logInControl8.FocusUserID = true;
this.logInControl8.KeyboardControl = this.keyboardControl8;
this.logInControl8.KeyboardName = "Keyboard 08";
this.logInControl8.LanguageCulture = "en-US";
this.logInControl8.Location = new System.Drawing.Point(307, 386);
this.logInControl8.Methodology = "Synonymous";
this.logInControl8.Name = "logInControl8";
this.logInControl8.Size = new System.Drawing.Size(300, 90);
this.logInControl8.TabIndex = 21;
//
// keyboardControl8
//
this.keyboardControl8.EnableF11 = false;
this.keyboardControl8.EnableF12 = false;
this.keyboardControl8.Freeze = false;
this.keyboardControl8.KeyboardID = "";
this.keyboardControl8.KeyboardName = "Keyboard 08";
this.keyboardControl8.LanguageCulture = "en-US";
this.keyboardControl8.Location = new System.Drawing.Point(307, 386);
this.keyboardControl8.Name = "keyboardControl8";
this.keyboardControl8.SessionID = 0;
this.keyboardControl8.Size = new System.Drawing.Size(300, 90);
this.keyboardControl8.TabIndex = 8;
this.keyboardControl8.TemplateDetailID = 0;
//
// logInControl7
//
this.logInControl7.FocusUserID = true;
this.logInControl7.KeyboardControl = this.keyboardControl7;
this.logInControl7.KeyboardName = "Keyboard 07";
this.logInControl7.LanguageCulture = "en-US";
this.logInControl7.Location = new System.Drawing.Point(3, 386);
this.logInControl7.Methodology = "Synonymous";
this.logInControl7.Name = "logInControl7";
this.logInControl7.Size = new System.Drawing.Size(300, 90);
this.logInControl7.TabIndex = 20;
//
// keyboardControl7
```

```
//
this.keyboardControl7.EnableF11 = false;
this.keyboardControl7.EnableF12 = false;
this.keyboardControl7.Freeze = false;
this.keyboardControl7.KeyboardID = "";
this.keyboardControl7.KeyboardName = "Keyboard 07";
this.keyboardControl7.LanguageCulture = "en-US";
this.keyboardControl7.Location = new System.Drawing.Point(3, 386);
this.keyboardControl7.Name = "keyboardControl7";
this.keyboardControl7.SessionID = 0;
this.keyboardControl7.Size = new System.Drawing.Size(300, 90);
this.keyboardControl7.TabIndex = 7;
this.keyboardControl7.TemplateDetailID = 0;
//
// logInControl6
//
this.logInControl6.FocusUserID = true;
this.logInControl6.KeyboardControl = this.keyboardControl6;
this.logInControl6.KeyboardName = "Keyboard 06";
this.logInControl6.LanguageCulture = "en-US";
this.logInControl6.Location = new System.Drawing.Point(307, 290);
this.logInControl6.Methodology = "Synonymous";
this.logInControl6.Name = "logInControl6";
this.logInControl6.Size = new System.Drawing.Size(300, 90);
this.logInControl6.TabIndex = 19;
//
// keyboardControl6
//
this.keyboardControl6.EnableF11 = false;
this.keyboardControl6.EnableF12 = false;
this.keyboardControl6.Freeze = false;
this.keyboardControl6.KeyboardID = "";
this.keyboardControl6.KeyboardName = "Keyboard 06";
this.keyboardControl6.LanguageCulture = "en-US";
this.keyboardControl6.Location = new System.Drawing.Point(307, 290);
this.keyboardControl6.Name = "keyboardControl6";
this.keyboardControl6.SessionID = 0;
this.keyboardControl6.Size = new System.Drawing.Size(300, 90);
this.keyboardControl6.TabIndex = 6;
this.keyboardControl6.TemplateDetailID = 0;
//
// logInControl5
//
this.logInControl5.FocusUserID = true;
this.logInControl5.KeyboardControl = this.keyboardControl5;
this.logInControl5.KeyboardName = "Keyboard 05";
this.logInControl5.LanguageCulture = "en-US";
this.logInControl5.Location = new System.Drawing.Point(3, 290);
this.logInControl5.Methodology = "Synonymous";
this.logInControl5.Name = "logInControl5";
this.logInControl5.Size = new System.Drawing.Size(300, 90);
this.logInControl5.TabIndex = 18;
//
// keyboardControl5
//
this.keyboardControl5.EnableF11 = false;
this.keyboardControl5.EnableF12 = false;
```

```
this.keyboardControl5.Freeze = false;
this.keyboardControl5.KeyboardID = "";
this.keyboardControl5.KeyboardName = "Keyboard 05";
this.keyboardControl5.LanguageCulture = "en-US";
this.keyboardControl5.Location = new System.Drawing.Point(3, 290);
this.keyboardControl5.Name = "keyboardControl5";
this.keyboardControl5.SessionID = 0;
this.keyboardControl5.Size = new System.Drawing.Size(300, 90);
this.keyboardControl5.TabIndex = 5;
this.keyboardControl5.TemplateDetailID = 0;
//
// logInControl4
//
this.logInControl4.FocusUserID = true;
this.logInControl4.KeyboardControl = this.keyboardControl4;
this.logInControl4.KeyboardName = "Keyboard 04";
this.logInControl4.LanguageCulture = "en-US";
this.logInControl4.Location = new System.Drawing.Point(307, 194);
this.logInControl4.Methodology = "Synonymous";
this.logInControl4.Name = "logInControl4";
this.logInControl4.Size = new System.Drawing.Size(300, 90);
this.logInControl4.TabIndex = 17;
//
// keyboardControl4
//
this.keyboardControl4.EnableF11 = false;
this.keyboardControl4.EnableF12 = false;
this.keyboardControl4.Freeze = false;
this.keyboardControl4.KeyboardID = "";
this.keyboardControl4.KeyboardName = "Keyboard 04";
this.keyboardControl4.LanguageCulture = "en-US";
this.keyboardControl4.Location = new System.Drawing.Point(307, 194);
this.keyboardControl4.Name = "keyboardControl4";
this.keyboardControl4.SessionID = 0;
this.keyboardControl4.Size = new System.Drawing.Size(300, 90);
this.keyboardControl4.TabIndex = 4;
this.keyboardControl4.TemplateDetailID = 0;
//
// logInControl3
//
this.logInControl3.FocusUserID = true;
this.logInControl3.KeyboardControl = this.keyboardControl3;
this.logInControl3.KeyboardName = "Keyboard 03";
this.logInControl3.LanguageCulture = "en-US";
this.logInControl3.Location = new System.Drawing.Point(3, 194);
this.logInControl3.Methodology = "Synonymous";
this.logInControl3.Name = "logInControl3";
this.logInControl3.Size = new System.Drawing.Size(300, 90);
this.logInControl3.TabIndex = 16;
//
// keyboardControl3
//
this.keyboardControl3.EnableF11 = false;
this.keyboardControl3.EnableF12 = false;
this.keyboardControl3.Freeze = false;
this.keyboardControl3.KeyboardID = "";
this.keyboardControl3.KeyboardName = "Keyboard 03";
```

```
this.keyboardControl3.LanguageCulture = "en-US";
this.keyboardControl3.Location = new System.Drawing.Point(3, 194);
this.keyboardControl3.Name = "keyboardControl3";
this.keyboardControl3.SessionID = 0;
this.keyboardControl3.Size = new System.Drawing.Size(300, 90);
this.keyboardControl3.TabIndex = 3;
this.keyboardControl3.TemplateDetailID = 0;
//
// logInControl2
//
this.logInControl2.FocusUserID = true;
this.logInControl2.KeyboardControl = this.keyboardControl2;
this.logInControl2.KeyboardName = "Keyboard 02";
this.logInControl2.LanguageCulture = "en-US";
this.logInControl2.Location = new System.Drawing.Point(307, 98);
this.logInControl2.Methodology = "Synonymous";
this.logInControl2.Name = "logInControl2";
this.logInControl2.Size = new System.Drawing.Size(300, 90);
this.logInControl2.TabIndex = 15;
//
// keyboardControl2
//
this.keyboardControl2.EnableF11 = false;
this.keyboardControl2.EnableF12 = false;
this.keyboardControl2.Freeze = false;
this.keyboardControl2.KeyboardID = "";
this.keyboardControl2.KeyboardName = "Keyboard 02";
this.keyboardControl2.LanguageCulture = "en-US";
this.keyboardControl2.Location = new System.Drawing.Point(307, 98);
this.keyboardControl2.Name = "keyboardControl2";
this.keyboardControl2.SessionID = 0;
this.keyboardControl2.Size = new System.Drawing.Size(300, 90);
this.keyboardControl2.TabIndex = 2;
this.keyboardControl2.TemplateDetailID = 0;
//
// logInControl1
//
this.logInControl1.FocusUserID = true;
this.logInControl1.KeyboardControl = this.keyboardControl1;
this.logInControl1.KeyboardName = "Keyboard 01";
this.logInControl1.LanguageCulture = "en-US";
this.logInControl1.Location = new System.Drawing.Point(3, 98);
this.logInControl1.Methodology = "Synonymous";
this.logInControl1.Name = "logInControl1";
this.logInControl1.Size = new System.Drawing.Size(300, 90);
this.logInControl1.TabIndex = 14;
//
// keyboardControl1
//
this.keyboardControl1.EnableF11 = false;
this.keyboardControl1.EnableF12 = false;
this.keyboardControl1.Freeze = false;
this.keyboardControl1.KeyboardID = "";
this.keyboardControl1.KeyboardName = "Keyboard 01";
this.keyboardControl1.LanguageCulture = "en-US";
this.keyboardControl1.Location = new System.Drawing.Point(3, 98);
this.keyboardControl1.Name = "keyboardControl1";
```

```
            this.keyboardControl1.SessionID = 0;
            this.keyboardControl1.Size = new System.Drawing.Size(300, 90);
            this.keyboardControl1.TabIndex = 1;
            this.keyboardControl1.TemplateDetailID = 0;
            //
            // logInControl0
            //
            this.logInControl0.FocusUserID = true;
            this.logInControl0.KeyboardControl = this.keyboardControl0;
            this.logInControl0.KeyboardName = "Facilitator";
            this.logInControl0.LanguageCulture = "en-US";
            this.logInControl0.Location = new System.Drawing.Point(3, 3);
            this.logInControl0.Methodology = "Synonymous";
            this.logInControl0.Name = "logInControl0";
            this.logInControl0.Size = new System.Drawing.Size(300, 90);
            this.logInControl0.TabIndex = 13;
            //
            // keyboardControl0
            //
            this.keyboardControl0.EnableF11 = false;
            this.keyboardControl0.EnableF12 = false;
            this.keyboardControl0.Freeze = false;
            this.keyboardControl0.KeyboardID = "";
            this.keyboardControl0.KeyboardName = "Facilitator";
            this.keyboardControl0.LanguageCulture = "en-US";
            this.keyboardControl0.Location = new System.Drawing.Point(3, 3);
            this.keyboardControl0.Name = "keyboardControl0";
            this.keyboardControl0.SessionID = 0;
            this.keyboardControl0.Size = new System.Drawing.Size(300, 90);
            this.keyboardControl0.TabIndex = 0;
            this.keyboardControl0.TemplateDetailID = 0;
            //
            // OpenSession
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;
            this.AutoSize = true;
            this.ClientSize = new System.Drawing.Size(1016, 688);
            this.ControlBox = false;
            this.Controls.Add(this.splitContainer1);
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
            this.Name = "OpenSession";
            this.Text = "Brain Strom";
            this.Load += new System.EventHandler(this.OpenSession_Load);
            this.InputLanguageChanged += new
System.Windows.Forms.InputLanguageChangedEventHandler(this.OpenSession_InputLanguageChan
ged);
            this.FormClosing += new
System.Windows.Forms.FormClosingEventHandler(this.OpenSession_FormClosing);
            this.splitContainer1.Panel1.ResumeLayout(false);
            this.splitContainer1.Panel1.PerformLayout();
            this.splitContainer1.Panel2.ResumeLayout(false);
            this.splitContainer1.ResumeLayout(false);
            this.ResumeLayout(false);

        }
```

```
        #endregion

        private Sdgt.SdgManager sdgManager1;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Button ButtonWord;
        private System.Windows.Forms.SplitContainer splitContainer1;
        private KeyboardControl keyboardControl2;
        private KeyboardControl keyboardControl1;
        private KeyboardControl keyboardControl11;
        private KeyboardControl keyboardControl12;
        private KeyboardControl keyboardControl9;
        private KeyboardControl keyboardControl10;
        private KeyboardControl keyboardControl7;
        private KeyboardControl keyboardControl8;
        private KeyboardControl keyboardControl5;
        private KeyboardControl keyboardControl6;
        private KeyboardControl keyboardControl3;
        private KeyboardControl keyboardControl4;
        private KeyboardControl keyboardControl0;
        private LogInControl logInControl0;
        private LogInControl logInControl1;
        private LogInControl logInControl2;
        private LogInControl logInControl4;
        private LogInControl logInControl3;
        private LogInControl logInControl6;
        private LogInControl logInControl5;
        private LogInControl logInControl8;
        private LogInControl logInControl7;
        private LogInControl logInControl12;
        private LogInControl logInControl11;
        private LogInControl logInControl10;
        private LogInControl logInControl9;
        private System.Windows.Forms.Button ButtonFreeze;
        private System.Windows.Forms.Button ButtonEnable;
        private System.Windows.Forms.ComboBox ComboBoxKeyboard;
        private System.Windows.Forms.Button ButtonNext;
        private System.Windows.Forms.Button ButtonPrevious;
        private System.Windows.Forms.Label LabelTemplateDetail;
        private System.Windows.Forms.Button ButtonOpenWord;
        private System.Windows.Forms.TextBox TextBoxTemp;
        private System.Windows.Forms.CheckBox chbxF11ForAll;
        private System.Windows.Forms.Button btnVote;
        private System.Windows.Forms.Button btnStopVote;
        public System.Windows.Forms.ComboBox cbxVoteType;
        public WinWordControl.WinWordControl objWinWordControl;
    }
}
```

ViewSession.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.IO;

namespace GroupStorm
{
  public partial class ViewSession : Form
  {
    public Word.ApplicationClass wordAppMain;
    public Word._Document wordDoc;
    public Word.Application wordApp;

    private DataSetBrain.TemplateDetailsDataTable templateDetailsDataTable = new
DataSetBrain.TemplateDetailsDataTable();
    private DataSetBrainTableAdapters.TemplateDetailsTableAdapter templateDetailsTableAdapter =
new GroupStorm.DataSetBrainTableAdapters.TemplateDetailsTableAdapter();
    private DataSetBrain.SessionsDataTable sessionsDataTable = new
DataSetBrain.SessionsDataTable();
    private DataSetBrainTableAdapters.SessionsTableAdapter sessionsTableAdapter = new
DataSetBrainTableAdapters.SessionsTableAdapter();
    private DataSetBrain.SessionParticipantsDataTable sessionParticipantsDataTable = new
DataSetBrain.SessionParticipantsDataTable();
    private DataSetBrainTableAdapters.SessionParticipantsTableAdapter
sessionParticipantsTableAdapter = new
DataSetBrainTableAdapters.SessionParticipantsTableAdapter();
    private DataSetBrain.UsersDataTable usersDataTable = new DataSetBrain.UsersDataTable();
    private DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();

    private int _sessionID = 0;
    private string _status = "New";
    private string _methodology = "Anonymous";
    private int CurrentRow = 0;
    private int MaxRow = 0;
    private string _copyString = "";
    public bool forAll = false;
    public bool WindowWord = false;
    public string _votingText = "";
    public bool _vote = false;
    private string _TopicName = "";

    public ViewSession()
    {
      InitializeComponent();
    }

    private void ViewSession_Load(object sender, EventArgs e)
    {
      // Get Session
      sessionsDataTable = sessionsTableAdapter.GetDataBy_SessionID(_sessionID);
      this.Text = sessionsDataTable.Rows[0]["SessionName"].ToString();
      int TemplateID = (int)sessionsDataTable.Rows[0]["TemplateID"];

      //Set open word type (true:open word in new window, false:use same window)
      WindowWord = ((MDIParent1)this.MdiParent).GetWindowWord();
```

```
object fileName = "c:\\" + this.Text + ".doc";
object readOnly = false;
object isVisible = true;
object missing = System.Reflection.Missing.Value;

if (sender != null)
{
   MessageBox.Show("Please wait while the document is being displayed");
}

//Close Word
try
{
   objWinWordControl.CloseControl();

   object dummy = null;
   object dummy2 = (object)false;
   wordDoc.Close(ref dummy, ref dummy, ref dummy);
   // Change the line below.
   wordApp.Quit(ref dummy2, ref dummy, ref dummy);

}
catch { }
finally
{
   objWinWordControl.document = null;
   WinWordControl.WinWordControl.wd = null;
   WinWordControl.WinWordControl.wordWnd = 0;
}

try
{
   //delete old file
   if (File.Exists(fileName.ToString()))
   {
      File.Delete(fileName.ToString());
   }

   //if (WindowWord)
   if (((MDIParent1)this.MdiParent).GetWindowWord())
   {
      //Open word in new window
      object file = "";
      object newTemplate = false;
      object docType = 0;
      isVisible = true;
      wordAppMain = new Word.ApplicationClass();
      wordDoc = wordAppMain.Documents.Add(ref file, ref newTemplate, ref docType, ref
isVisible);
      wordApp = wordDoc.Application;
      wordApp.Visible = true;
      //wordDoc.Activate();

      //focus this form
      ((MDIParent1)this.ParentForm).Activate();
   }
```

```
            else
            {
               //Open word in this window
               objWinWordControl.LoadDocument("");
               wordDoc = objWinWordControl.document;
               wordApp = wordDoc.Application;
            }

            try
            {
               //Display Menu Bar
               wordDoc.ActiveWindow.Application.CommandBars["Menu Bar"].Enabled = true;
            }
            catch (Exception ex)
            {
               Console.Write(ex.Message);
            }

            // Write Session Name
            wordApp.Selection.TypeText(this.Text);
            wordApp.Selection.TypeText(Environment.NewLine);

            //Get transaction data of this session
            DataSetBrain.SessionTransactionsDataTable STDataTable = new
DataSetBrain.SessionTransactionsDataTable();
            DataSetBrainTableAdapters.SessionTransactionsTableAdapter STTableAdapter = new
DataSetBrainTableAdapters.SessionTransactionsTableAdapter();
            STDataTable = STTableAdapter.GetDataBy_SessionID(_sessionID);

            string TempTemplateDetailName = "";

            if (_methodology == "Anonymous") //Methodology is Anonymous
            {
               foreach (DataRow myRow in STDataTable.Rows)
               {
                  if (TemplateID > 0)
                  {
                     DataRow[] dr = templateDetailsDataTable.Select("TemplateDetailID = " +
myRow["TemplateDetailID"].ToString());
                     if (dr.Length > 0)
                     {
                        //Display Topic Name
                        if (TempTemplateDetailName != dr[0]["TopicName"].ToString())
                        {
                           wordApp.Selection.TypeText(dr[0]["TopicName"].ToString());
                           wordApp.Selection.TypeText(Environment.NewLine);
                           TempTemplateDetailName = dr[0]["TopicName"].ToString();
                        }
                     }
                  }

                  //Display transaction text
                  wordApp.Selection.TypeText(myRow["TransactionInput"].ToString());
                  wordApp.Selection.TypeText(Environment.NewLine);
               }
            }
            else //Methodology is Synonymous
```

```
            {
                foreach (DataRow myRow in STDataTable.Rows)
                {
                    if (TemplateID > 0)
                    {
                        DataRow[] dr = templateDetailsDataTable.Select("TemplateDetailID = " +
myRow["TemplateDetailID"].ToString());
                        if (dr.Length > 0)
                        {
                            //Display Topic Name
                            if (TempTemplateDetailName != dr[0]["TopicName"].ToString())
                            {
                                wordApp.Selection.TypeText(dr[0]["TopicName"].ToString());
                                wordApp.Selection.TypeText(Environment.NewLine);
                                TempTemplateDetailName = dr[0]["TopicName"].ToString();
                            }
                        }
                    }

                    if (!string.IsNullOrEmpty(myRow["UserID"].ToString()))
                    {
                        string Name = "";
                        //Get Name
                        DataSetBrain.UsersDataTable usersDataTable = new
DataSetBrain.UsersDataTable();
                        DataSetBrainTableAdapters.UsersTableAdapter usersTableAdapter = new
DataSetBrainTableAdapters.UsersTableAdapter();
                        usersDataTable =
usersTableAdapter.GetDataBy_UserID(myRow["UserID"].ToString());

                        if (usersDataTable.Rows.Count > 0)
                        {
                            Name = usersDataTable.Rows[0]["Name"].ToString() + " " +
usersDataTable.Rows[0]["LastName"].ToString();
                        }

                        //Display transaction text
                        //Display owner
                        wordApp.Selection.TypeText(Name + " : " +
myRow["TransactionInput"].ToString());
                        wordApp.Selection.TypeText(Environment.NewLine);
                    }
                    else
                    {
                        //Display transaction text
                        wordApp.Selection.TypeText(myRow["TransactionInput"].ToString());
                        wordApp.Selection.TypeText(Environment.NewLine);
                    }
                }
            }

            // Save File
            wordDoc.SaveAs(ref fileName, ref missing, ref missing, ref missing, ref missing, ref
missing, ref readOnly, ref missing, ref missing, ref missing, ref missing);
        }
        catch (Exception ex) { String err = ex.Message; }
    }
```

```
        public int SessionID
        {
            set
            {
                _sessionID = value;
            }
            get
            {
                return _sessionID;
            }
        }

        public string Status
        {
            set
            {
                _status = value;
            }
            get
            {
                return _status;
            }
        }

        public string Methodology
        {
            set
            {
                _methodology = value;
            }
            get
            {
                return _methodology;
            }
        }
    }
}
```

ViewSession.Designer.cs

```
namespace GroupStorm
{
    partial class ViewSession
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
```

```
        /// <param name="disposing">true if managed resources should be disposed; otherwise,
    false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            this.objWinWordControl = new WinWordControl.WinWordControl();
            this.SuspendLayout();
            //
            // objWinWordControl
            //
            this.objWinWordControl.BackColor = System.Drawing.SystemColors.Highlight;
            this.objWinWordControl.Dock = System.Windows.Forms.DockStyle.Fill;
            this.objWinWordControl.Location = new System.Drawing.Point(0, 0);
            this.objWinWordControl.Name = "objWinWordControl";
            this.objWinWordControl.Size = new System.Drawing.Size(1016, 688);
            this.objWinWordControl.TabIndex = 1;
            //
            // ViewSession
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(96F, 96F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Dpi;
            this.AutoSize = true;
            this.ClientSize = new System.Drawing.Size(1016, 688);
            this.ControlBox = false;
            this.Controls.Add(this.objWinWordControl);
            this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedSingle;
            this.Name = "ViewSession";
            this.Text = "Brain Strom";          this.Load += new
    System.EventHandler(this.ViewSession_Load);
            this.ResumeLayout(false);

        }

        #endregion

        public WinWordControl.WinWordControl objWinWordControl;
    }
}
```

Thesis Title: The development of GDSS to support group decision making by increasing the participation of Thai graduate students

Author: ……Jetsura Vongvichien……..……

Thesis Principle Supervisor: ……Professor Charles.S. Newton.....…….....

Thesis Supervisor: ……Dr Adisorn Na Ubon…….....……..

Program: ……PhD BI……..

Academic Year: …2010……..……(*Year of graduation*)