

รายงานผลการวิจัย

เรื่อง

การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่

ระบบปฏิบัติการแอนดรอยด์

เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด

A Development of on an Android Application for Shortest Bangkok Bus

Route Searching

โดย

ผศ.ดร.เนืองวงศ์ ทวยเจริญ

รายงานวิจัยนี้ได้รับทุนส่งเสริมและสนับสนุนการวิจัยจาก

มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2559

ชื่อเรื่อง : การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อ
ค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
ผู้วิจัย : ผศ.ดร.เนืองวงศ์ ทวยเจริญ สถาบัน : มหาวิทยาลัยธุรกิจบัณฑิต
ปีที่พิมพ์ : 2559 สถานที่พิมพ์ : มหาวิทยาลัยธุรกิจบัณฑิต
แหล่งที่เก็บรายงานการวิจัยฉบับสมบูรณ์: มหาวิทยาลัยธุรกิจบัณฑิต
จำนวนหน้างานวิจัย : 83 หน้า
คำสำคัญ : ขสมก. โปรแกรมประยุกต์บนโทรศัพท์ แอนดรอยด์ ระยะทางที่สั้นที่สุด รถ
สาธารณะ
ลิขสิทธิ์ : มหาวิทยาลัยธุรกิจบัณฑิต

บทคัดย่อ

ปัจจุบัน ข้อมูลของระบบขนส่งมวลชนของ ขสมก.ที่มีอยู่เป็นข้อมูลที่ไม่ถูกต้องนัก เพราะไม่มี
การปรับปรุงข้อมูลให้เป็นปัจจุบันเป็นเวลานาน และไม่ได้มีการรวบรวมข้อมูลอย่างเป็นระบบ งานวิจัย
นี้ได้กล่าวถึงการพัฒนาระบบเก็บข้อมูลป้ายรถประจำทางด้วยโทรศัพท์ระบบปฏิบัติการแอนดรอยด์
และสร้างข้อมูลเส้นทางรถโดยสาร โดยมีการบันทึกการเดินรถของรถแต่ละสายที่สร้างจาก My maps
ของ Google จากนั้นนำเส้นทางที่สร้างไปคำนวณด้วยโปรแกรมบนเครื่องแม่ข่ายและแปลงพิกัดป้ายกับ
พิกัดเส้นทางที่เก็บรวบรวมได้ให้กลายเป็นเส้นทางรถประจำทางแต่ละสายตามลำดับป้ายที่ผ่าน
โดยอัตโนมัติ ผลการทดสอบระบบพบว่าระบบสามารถเก็บข้อมูลป้ายรถและสร้างเส้นทางเดินรถได้
และโปรแกรมการสร้างเส้นทางเดินรถผ่านป้ายรถใช้เวลาคำนวณเฉลี่ย 5.75 ms ต่อหนึ่งพิกัดเส้นทางรถ
ข้อมูลจากระบบดังกล่าวได้ถูกนำไปใช้ในการพัฒนาเอพีไอและโปรแกรมบนโทรศัพท์เคลื่อนที่บน
ระบบปฏิบัติการแอนดรอยด์เพื่อให้ข้อมูลการเดินทางด้วยรถประจำทางแก่คนทั่วไป เอพีไอที่พัฒนาขึ้น
ประกอบด้วยฐานข้อมูลเส้นทางรถในสังกัด ขสมก. ข้างต้น และขั้นตอนวิธีในการค้นหา
เส้นทางที่สั้นที่สุดที่พิจารณาทั้งการเปลี่ยนรถและการเดินอีกด้วย ผลการทดสอบระบบพบว่าเอพีไอที่
สร้างขึ้นใช้หน่วยความจำน้อยกว่า 6 MB และใช้เวลาน้อยกว่า 0.025ms เมื่อคำนวณเส้นทางในกราฟที่มี
ป้ายรถจำนวน 10,000 ป้าย เมื่อเรียกใช้เอพีไอในโปรแกรมประยุกต์ระบบปฏิบัติการแอนดรอยด์พบว่า
โปรแกรมประยุกต์สามารถตอบกลับภายในเวลา 4 วินาที ซึ่งเป็นเวลาที่ผู้ใช้ยอมรับได้ ผลการสำรวจ
ความคิดเห็นของผู้ใช้งานจำนวน 64 คน จาก 8 เขตการเดินรถ พบว่าผู้ใช้งานมีระดับความพึงพอใจมาก
ในระดับ 4.16

Title : A Development of on an Android Application for Shortest Bangkok Bus Route Searching

Researcher : Asst.Prof.Dr.Nuengwong Tuaycharoen Institution: Dhurakij Pundit University

Year of Publication: 2016 Publisher: Dhurakij Pundit University

Sources : Dhurakij Pundit University

No. of page : 83 pages

Keywords : BMTA, Mobile Application, Android, Shortest Path, Public Transportation

Abstract

Currently, the transit system information provided by BMTA is not accurate because the information is not updated for decades. Furthermore, there is no systematic way of gathering and updating the information. This research describes the development of bus stop collecting and bus route creating system. We use an Android smartphone as a tool to collect the bus stop coordinates and send them to the system's database server. We also create bus route coordinates with Google's My Maps. Then, a program on the server converts bus route coordinates into a line of bus stops automatically. The experimental results show that the system can collect bus stops, and the server's programs can map bus stops to a bus route with an average of 5.75 ms per bus route coordinate. The information created in this system is used to develop an API and an Android Application to inform general bus passengers. The API includes the BMTA bus database and a shortest path algorithm with bus transits and walking consideration. The experimental results show that the API takes less than 6 MB of memory and under 0.025ms of processing time for 10,000 nodes. Including in an android application, our application's response times are under 4 seconds, which is acceptable for user experience. Our survey on 64 passengers in 8 Bangkok Metropolitan Bus Areas finds that the users rank our application with high satisfaction level of 4.16.

สารบัญ

สารบัญรูป	จ
สารบัญตาราง	ฉ
บทที่ 1_บทนำ	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	5
1.3 ขอบเขตของงาน	5
1.4 ประโยชน์ที่คาดว่าจะได้รับ	6
1.5 ข้อยกเว้นของการวิจัย	6
1.6 ข้อตกลงเบื้องต้น	7
1.7 การออกแบบระบบในภาพรวม	7
1.8 ภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนา	9
1.9 อุปกรณ์และซอฟต์แวร์ที่ใช้ในการพัฒนา	9
1.10 แผนการดำเนินงาน	10
บทที่ 2_ทฤษฎีและงานที่เกี่ยวข้อง	11
2.1 ทฤษฎีที่เกี่ยวข้อง	11
2.2 งานวิจัยที่เกี่ยวข้อง	12
2.3 งานที่เกี่ยวข้อง	13
บทที่ 3_การออกแบบและพัฒนา	15
3.1 การออกแบบฐานข้อมูล	15
3.2 พจนานุกรมข้อมูลของฐานข้อมูลระบบ	17
3.3 Use-case	20
3.4 สถาปัตยกรรมระบบ	20
3.5 ข้อกำหนดซอฟต์แวร์	23

3.6 Class Diagram	28
3.7 Sequence Diagram	31
3.8 State Diagram	35
3.9 ขั้นตอนวิธีการคำนวณเส้นทางเดินรถ	46
3.10 ขั้นตอนวิธีการคำนวณเส้นทางรถโดยสารประจำทางที่สั้นที่สุด	47
บทที่ 4 ผลการดำเนินงาน	49
4.1 โปรแกรมตรวจสอบป้ายรถประจำทาง	49
4.2 โปรแกรมสำรวจเส้นทางเดินรถประจำทาง	51
4.3 โปรแกรมกรอกข้อมูลเส้นทางเดินรถประจำทางบนเว็บ	54
4.4 โปรแกรมค้นหาเส้นทางที่สั้นที่สุดบนแอนดรอยด์	57
4.5 ผลการทดสอบประสิทธิภาพของระบบ	65
4.6 ผลการสำรวจความคิดเห็นต่อการใช้งานของแอปพลิเคชันค้นหาเส้นทางรถประจำทางบน โทรศัพท์ระบบปฏิบัติการแอนดรอยด์	69
บทที่ 5 สรุป	71
บรรณานุกรม	72

สารบัญรูป

รูปที่	หน้า
รูปที่ 1.1 จำนวนผู้โดยสารที่ใช้บริการรถโดยสารประจำทาง ขสมก. ตั้งแต่ปี พ.ศ. 2535-2554	2
รูปที่ 1.2 จำนวนรถยนต์นั่งส่วนบุคคลและรถจักรยานยนต์ที่จดทะเบียนในกรุงเทพมหานครสะสม ตั้งแต่ปี พ.ศ. 2535-2556	3
รูปที่ 1.3 จำนวนผู้โดยสารรถไฟฟ้าขนส่งมวลชนเฉลี่ยต่อวัน ตั้งแต่ปี พ.ศ. 2542-2556 [4]	3
รูปที่ 1.4 ภาพรวมของระบบ	7
รูปที่ 3.1 แผนภาพอ็วาร์ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด	15
รูปที่ 3.2 แผนภาพอ็วาร์ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด	16
รูปที่ 3.3 use-case ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด	20
รูปที่ 3.4 สถาปัตยกรรมฮาร์ดแวร์ของระบบ	21
รูปที่ 3.5 สถาปัตยกรรมซอฟต์แวร์ของระบบ	22
รูปที่ 3.6 class diagram ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด สำหรับ User	28
รูปที่ 3.7 class diagram ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด สำหรับUser(ต่อ)	29
รูปที่ 3.8 class diagram ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด สำหรับ Admin	30
รูปที่ 3.9 Sequence diagram ของ use-case เลือกสายทิ้งไปและกลับ (User)	31
รูปที่ 3.10 Sequence diagram ของ use-case ค้นหาปลายทาง (User)	31
รูปที่ 3.11 Sequence diagram ของ use-case ข้อมูลป้าย (User)	32
รูปที่ 3.12 Sequence diagram ของ use-case ค้นหาข้อมูลรอบตัว(User)	32

สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 3.13 Sequence diagram ของ use-case แก้ไขป้าย (Admin)	33
รูปที่ 3.14 Sequence diagram ของ use-case เช็คป้ายถูกต้อง (Admin)	33
รูปที่ 3.15 Sequence diagram ของ use-case ลบป้าย (Admin)	34
รูปที่ 3.16 Sequence diagram ของ use-case เก็บข้อมูลสายรถ (Admin)	34
รูปที่ 3.17 state diagram ของ class MainActivity (User)	35
รูปที่ 3.18 state diagram ของ class Tab (User)	35
รูปที่ 3.19 state diagram ของ class Directory (User)	35
รูปที่ 3.20 state diagram ของ class bus (User)	35
รูปที่ 3.21 state diagram ของ class result CustomListAdapterbus (User)	36
รูปที่ 3.22 state diagram ของ class result Buslistbusstop (User)	36
รูปที่ 3.23 state diagram ของ class result BusstopGo (User)	36
รูปที่ 3.24 state diagram ของ class result CustomListAdapterbusstop_list (User)	36
รูปที่ 3.25 state diagram ของ class BusstopListBlack (User)	37
รูปที่ 3.26 state diagram ของ class CustomListAdapterbusstop_listB (User)	37
รูปที่ 3.27 state diagram ของ class railway (User)	37
รูปที่ 3.28 state diagram ของ class CustomListAdapterrailway (User)	38
รูปที่ 3.29 state diagram ของ class ferry (User)	38
รูปที่ 3.30 state diagram ของ class CustomListAdapterferry (User)	39
รูปที่ 3.31 state diagram ของ class map (User)	39
รูปที่ 3.32 state diagram ของ class CustomListAdaptermap (User)	39
รูปที่ 3.33 state diagram ของ class mapbus_busstop (User)	39
รูปที่ 3.34 state diagram ของ class route (User)	39
รูปที่ 3.35 state diagram ของ class markerroute (User)	40
รูปที่ 3.36 state diagram ของ class near (User)	40
รูปที่ 3.37 state diagram ของ class savebusstop (User)	40
รูปที่ 3.38 state diagram ของ class DBnamebusstop (User)	40

สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 3.39 state diagram ของ class Server (User)	41
รูปที่ 3.40 state diagram ของ class main(Admin)	41
รูปที่ 3.41 state diagram ของ class tag (Admin)	42
รูปที่ 3.42 state diagram ของ class tagging (Admin)	43
รูปที่ 3.43 state diagram ของ class tagmap (Admin)	43
รูปที่ 3.44 state diagram ของ class tagend (Admin)	44
รูปที่ 3.45 state diagram ของ class editbusstopM (Admin)	44
รูปที่ 3.46 state diagram ของ class editmenu (Admin)	45
รูปที่ 3.47 state diagram ของ class editbusstop (Admin)	45
รูปที่ 3.48 state diagram ของ class database (Admin)	46
รูปที่ 4.1 หน้าจอ หมุดป้ายสายรถประจำทาง	49
รูปที่ 4.2 หน้าจอ หมุดที่ทำการเช็ค	50
รูปที่ 4.3 หน้าจอตรวจสอบป้าย	50
รูปที่ 4.4 หน้าจอ ยืนยันความถูกต้อง	51
รูปที่ 4.5 หน้าจอ เช็คการเปิดระบบการบอกตำแหน่ง (GPS)	51
รูปที่ 4.6 หน้าจอ การกรอกข้อมูลและ ประเภทสายรถประจำทาง	52
รูปที่ 4.7 หน้าจอ การเปิดการทำงานเบื้องหลัง	52
รูปที่ 4.8 หน้าจอ วาดเส้นทางสายรถประจำทาง	53
รูปที่ 4.9 หน้าจอ บอกข้อมูลและส่งข้อมูลขึ้นฐานข้อมูล	54
รูปที่ 4.10 หน้าจอ ป้ายสายรถประจำทางใน กรุงเทพมหานคร และปริมณฑล	55
รูปที่ 4.11 หน้าจอ เลือกประเภทในการเดินทางไปยังป้าย	55
รูปที่ 4.12 หน้าจอ ระบุสายรถประจำทาง	56
รูปที่ 4.13 หน้าจอ เลือกประเภทการเดินทางของสายรถประจำทาง	56
รูปที่ 4.14 หน้าจอ ข้อมูลเส้นทางสายรถประจำทาง	57
รูปที่ 4.15 หน้าจอดาวน์โหลดป้ายรถประจำทาง	57
รูปที่ 4.16 หน้าจอแสดงสายรถประจำทางทั้งหมด	58

สารบัญรูป(ต่อ)

รูปที่	หน้า
รูปที่ 4.17 หน้าจอ แสดงป้ายของสายรถประจำทาง (ขาไป)	59
รูปที่ 4.18 หน้าจอ แสดงป้ายของสายรถประจำทาง (ขากลับ)	59
รูปที่ 4.19 หน้าจอ แสดงสายรถไฟ	60
รูปที่ 4.20 หน้าจอ แสดงการเดินทาง	60
รูปที่ 4.21 หน้าจอ บอกแสดงบริการสายรถประจำทาง	61
รูปที่ 4.22 หน้าจอ บอกข้อมูลสายรถประจำทาง	61
รูปที่ 4.23 หน้าจอ ค้นหาเส้นทางรถประจำทาง	62
รูปที่ 4.24 หน้าจอ เลือกการเดินทางด้วยสายรถประจำทาง	62
รูปที่ 4.25 หน้าจอ แสดงเส้นทางการเดินทางที่สั้นที่สุด	63
รูปที่ 4.26 หน้าจอ แสดงเส้นทางการเดินทางมีจุดเปลี่ยนรถน้อยที่สุด	64
รูปที่ 4.27 หน้าจอ แสดงป้ายรอบตัว.	64
รูปที่ 4.28 ค่าความแม่นยำ (Accuracy) ที่วัดได้จากโทรศัพท์แอนดรอยด์ทั้ง 3 รุ่น	65
รูปที่ 4.29 การใช้งานพื้นที่หน่วยความจำของเอพีไอ	67
รูปที่ 4.30 เวลาที่เอพีไอใช้ในการประมวลผล	68

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน	10
ตารางที่ 3.1 ตาราง Busstop	17
ตารางที่ 3.2 ตาราง Bus	18
ตารางที่ 3.3 ตาราง Busgraph	18
ตารางที่ 3.4 ตาราง Busroute	19
ตารางที่ 3.5 ตาราง Structured Language Specification ของ Use-case แก้ไขป้าย	23
ตารางที่ 3.6 ตาราง Structured Language Specification ของ Use-case Map ป้าย	23
ตารางที่ 3.7 ตาราง Structured Language Specification ของ Use-case เช็คป้าย	24
ตารางที่ 3.8 ตาราง Structured Language Specification ของ Use-case เก็บข้อมูลสายรถ	24
ตารางที่ 3.9 ตาราง Structured Language Specification ของ Use-case ลบป้าย	25
ตารางที่ 3.10 ตาราง Structured Language Specification ของ Use-case ค้นหาปลายทาง	25
ตารางที่ 3.11 ตาราง Structured Language Specification ของ Use-case ค้นหาป้ายรอบตัว	26
ตารางที่ 3.12 ตาราง Structured Language Specification ของ Use-case ค้นหาสายรถประจำทาง	26
ตารางที่ 3.13 ตาราง Structured Language Specification ของ Use-case ข้อมูลป้าย	27
ตารางที่ 4.1 ประสิทธิภาพทางเวลาของการสร้างเส้นทางรถประจำทางผ่านป้ายรถ	66
ตารางที่ 4.2 เวลาตอบกลับของแอนดรอยแอปพลิเคชัน	68
ตารางที่ 4.3 ระดับความคิดเห็นต่อโปรแกรมบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ที่ได้พัฒนาขึ้น	69

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

จากข้อมูลในเดือนมกราคม 2558 ของระบบสถิติทางการทะเบียน [1] ระบุว่า ในกรุงเทพมหานครมีประชากรอยู่ทั้งสิ้น 5.6 ล้านคน แต่หากรวมประชากรแฝง เช่น แรงงานข้ามชาติ แรงงานจากต่างจังหวัด และคนไร้บ้านที่ยังไม่ได้อยู่ในระบบแล้ว จะมีประชากรราว 15 ล้านคน[2] ในเมืองใหญ่ที่มีประชากรหนาแน่นอย่างกรุงเทพมหานครนี้ จำเป็นต้องมีการบริการขนส่งสาธารณะที่สะดวก เพียงพอ และมีราคาที่เหมาะสมกับรายได้ของคนเมือง

ระบบขนส่งสาธารณะในกรุงเทพมหานคร ประกอบด้วย รถไฟฟ้าบีทีเอส รถไฟฟ้าเอ็มอาร์ที รถไฟฟ้าแอร์พอร์ตลิงค์ รถไฟไทย รถบีอาร์ที รถเมล์ รถผู้ร่วมบริการ เรือ มอเตอร์ไซค์รับจ้าง ตุ๊กตุ๊ก และรถแท็กซี่ ซึ่งในบรรดาการเดินทางด้วยระบบขนส่งสาธารณะเหล่านี้ รถโดยสารประจำทาง (ขสมก.) เป็นระบบที่ให้บริการตลอดทั้งคืนและรถที่บริการตลอดวัน ถือเป็นระบบขนส่งสาธารณะที่ครอบคลุมมากที่สุด และเป็นระบบขนส่งที่มีราคาถูกที่สุด แม้ว่าได้มีการปรับขึ้นค่าโดยสาร เป็น 12-24 บาทแล้วก็ตาม ในขณะที่รถธรรมดา (รถเมล์แดง) ยังคงราคาอยู่ที่ 6.50 บาท และรถปรับอากาศ (รถเมล์ส้ม) ยังคงเริ่มต้นที่ 12 บาท ซึ่งหากจะมีการปรับค่าโดยสาร ต้องรออนโยบายจากกระทรวงคมนาคมก่อน [1]

องค์การขนส่งมวลชนกรุงเทพ (ขสมก.) ได้จัดตั้งตามพระราชกฤษฎีกาจัดตั้งองค์การขนส่งมวลชนกรุงเทพ พ.ศ. 2519 เมื่อวันที่ 19 ตุลาคม 2519 [3] เป็นหน่วยงานรัฐวิสาหกิจประเภทสาธารณูปโภคสังกัดกระทรวงคมนาคม มีภารกิจและขอบเขตความรับผิดชอบในการจัดบริการรถโดยสารประจำทางวิ่งรับส่งผู้โดยสารในเขตกรุงเทพมหานคร และจังหวัดใกล้เคียง คือ นนทบุรี ปทุมธานี สมุทรปราการ สมุทรสงคราม และนครปฐม นอกจากนี้ ยังมีหน้าที่ในด้านประกอบการอื่น ๆ ที่เกี่ยวกับหรือต่อเนื่องกับการประกอบการขนส่งบุคคล เนื่องจากกิจการเดินรถโดยสารประจำทางจัดเป็นสาธารณูปการชนิดหนึ่งของรัฐที่ให้บริการแก่ประชาชนผู้มีรายได้น้อยและปานกลางเป็นหลัก การดำเนินการจึงมุ่งสนองตอบนโยบายของรัฐบาลในด้านการให้ความช่วยเหลือแก่ผู้มีรายได้น้อยโดยไม่หวังผลกำไร การจัดเก็บอัตราค่าโดยสารจึงอยู่ในอัตราต่ำกว่าต้นทุนตามที่รัฐบาลเป็นผู้กำหนด

นโยบาย การให้บริการของ ขสมก. ที่มุ่งในด้านความสะดวก รวดเร็ว ปลอดภัย และประหยัดค่าใช้จ่าย ในการเดินทางของผู้โดยสารเป็นหลัก ตลอดจนเป็นผู้ควบคุมและดูแลผู้ประกอบการเดินทางเอกชนที่ ให้บริการรถประเภทรถโดยสารขนาดใหญ่ (ทั้งรถธรรมดาและรถปรับอากาศ) รถมินิบัส รถเล็กวิ่งใน ซอย และรถตู้โดยสารปรับอากาศ

ในปี 2554 ขสมก. มีจำนวนรถโดยสารประจำทางรวมทุกประเภท 445 เส้นทาง คัน แบ่งเป็น รถของ ขสมก. เป็นผู้ให้บริการ 108 เส้นทาง รถโดยสารที่บริษัทเอกชนร่วมวิ่งบริการ จำนวน 101 เส้นทาง

สถิติของสำนักงานจราจรและขนส่ง [2] ในปี 2556 ระบุว่า ขสมก. มีสถิติจำนวนผู้โดยสารที่ใช้ บริการรถเมล์ ขสมก.ต่อวันเท่ากับ 935,726 คน แบ่งออกเป็นรถปรับอากาศ 538,218 คน รถธรรมดา 397,507 คน โดยมีจำนวนรถโดยสารประจำทางรวมทุกประเภท 15,821 คัน แบ่งเป็นรถของ ขสมก. เป็นผู้ให้บริการจำนวน 3,416 คัน

นอกจากนี้ ยังมีรถโดยสารที่บริษัทเอกชนร่วมวิ่งบริการ จำนวนรวม 12,405 คัน โดยแบ่งเป็น รถเล็กในซอย 2,266 คัน รถตู้โดยสาร 5,242 คัน รถมินิบัส 1,028 คัน และรถร่วมบริการ 3,869 คัน ถึงแม้ว่า ขสมก. จะมีจำนวนผู้โดยสารเกือบล้านคนต่อวัน แต่จำนวนผู้โดยสารมีแนวโน้มลดลง ดัง แสดงในรูปที่ 1.1 ในขณะที่ สถิติจำนวนรถยนต์ส่วนบุคคลและรถจักรยานยนต์ที่จดทะเบียนใน กรุงเทพมหานครกลับมาแนวโน้มเพิ่มมากขึ้นทุกปี ดังแสดงในรูปที่ 1.2

จำนวนผู้โดยสารที่ใช้บริการรถโดยสารประจำทาง ขสมก. ตั้งแต่ปี พ.ศ. 2535-2554



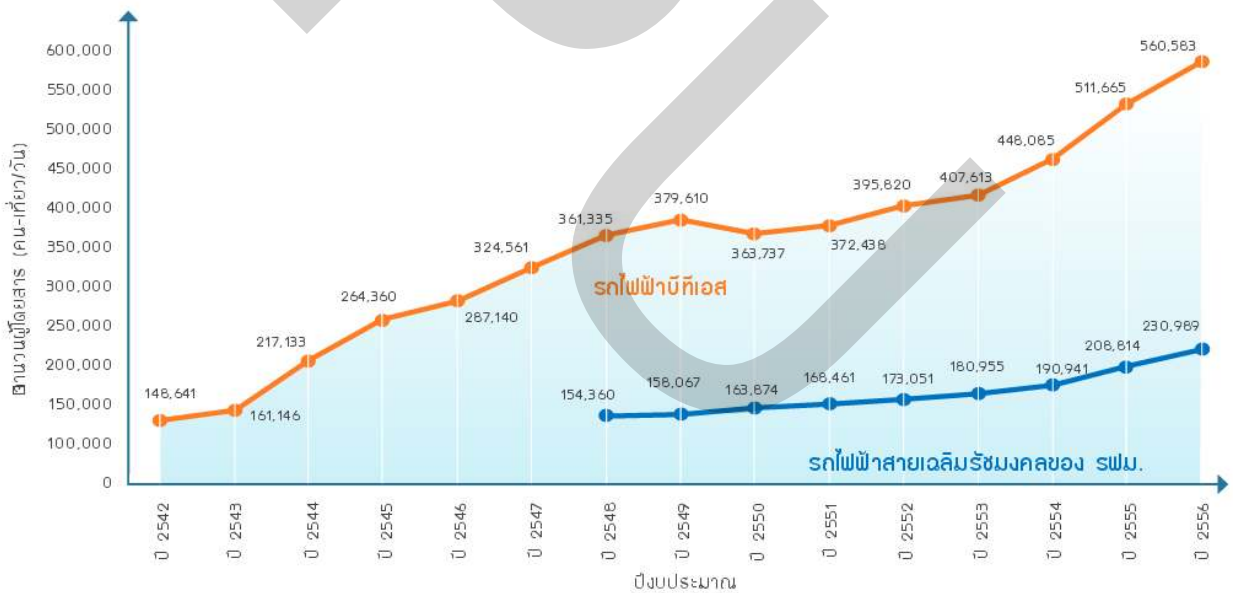
รูปที่ 1.1 จำนวนผู้โดยสารที่ใช้บริการรถโดยสารประจำทาง ขสมก. ตั้งแต่ปี พ.ศ. 2535-2554 [4]

จำนวนรถยนต์นั่งส่วนบุคคลและรถจักรยานยนต์ที่จดทะเบียนในกรุงเทพมหานครสะสมตั้งแต่ปี พ.ศ. 2535-2556



รูปที่ 1.2 จำนวนรถยนต์นั่งส่วนบุคคลและรถจักรยานยนต์ที่จดทะเบียนในกรุงเทพมหานครสะสมตั้งแต่ปี พ.ศ. 2535-2556 [4]

จำนวนผู้โดยสารรถไฟฟ้าขนส่งมวลชนเฉลี่ยต่อวัน ตั้งแต่ปี พ.ศ. 2542-2556



รูปที่ 1.3 จำนวนผู้โดยสารรถไฟฟ้าขนส่งมวลชนเฉลี่ยต่อวัน ตั้งแต่ปี พ.ศ. 2542-2556 [4]

จากการรวบรวมและวิเคราะห์ข้อมูล ปี 2535-2556 ที่เกี่ยวข้องกับการเดินทางในกรุงเทพมหานครของการรถไฟฟ้าขนส่งมวลชนแห่งประเทศไทย[4] พบว่า การเดินทางโดยรถส่วน

บุคคลทั้งรถยนต์และรถจักรยานยนต์ได้เพิ่มขึ้นอย่างต่อเนื่องและมีแนวโน้มเพิ่มขึ้นต่อไปในอนาคต โดยในระหว่างปี 2535-2556 รถยนต์เพิ่มขึ้นเฉลี่ยร้อยละ 6.75 ต่อปี และรถจักรยานยนต์เพิ่มขึ้นเฉลี่ย ร้อยละ 6.62 ต่อปี ในทางตรงข้าม จำนวนผู้ใช้บริการรถโดยสารสาธารณะของ ขสมก. มีแนวโน้มลดลงอย่างต่อเนื่อง โดยลดลงเฉลี่ยร้อยละ 6.8 ต่อปี

สถิติดังกล่าวแสดงให้เห็นว่าประชาชนในเขตกรุงเทพมหานครนิยมเดินทางโดยรถยนต์ส่วนบุคคล บีทีเอส และเอ็มอาร์ที เพิ่มขึ้นอย่างต่อเนื่อง เนื่องจากประชาชนต้องการความสะดวกสบายในการเดินทาง แต่ระบบขนส่งมวลชนของ ขสมก. ยังให้ความสะดวกได้ไม่สมบูรณ์นัก ดังนั้น การขนส่งสาธารณะของ ขสมก. จะต้องพัฒนาให้เพียงพอต่อความต้องการของประชาชน โดยให้เพิ่มจำนวนเที่ยว การเดินรถ เพิ่มการให้ข้อมูลการเดินทางในระบบของ ขสมก. และเพิ่มการให้ข้อมูลการเชื่อมต่อ เครื่องอำนวยความสะดวกโดยสารสาธารณะ สิ่งเหล่านี้จะทำให้ประชาชนมาใช้บริการมากขึ้น และสะดวกต่อการเดินทางยิ่งขึ้น

ในปัจจุบัน โทรศัพท์เคลื่อนที่อัจฉริยะ (Smart phone) เป็นที่นิยมอย่างมาก มีการแข่งขันจากหลายบริษัทผู้ผลิต ทำให้ผู้ผลิตจำนวนมากได้ผลิตโทรศัพท์อัจฉริยะออกมาให้เลือกใช้อย่างหลากหลาย ตั้งแต่ราคาหลายหมื่นบาท จนถึงราคาไม่กี่พันบาท จึงทำให้โทรศัพท์อัจฉริยะมีราคาถูกลงและคนทั่วไปสามารถซื้อหากันได้ คุณสมบัติหนึ่งของโทรศัพท์อัจฉริยะคือ ผู้ใช้งานสามารถเข้าถึงและเลือกติดตั้งโปรแกรมได้หลากหลาย ไม่ได้จำกัดอยู่เพียงโปรแกรมประยุกต์ที่ติดตั้งมาพร้อมกับโทรศัพท์เท่านั้น ดังนั้นผู้ใช้งานสามารถติดตั้งโปรแกรมที่เหมาะสมกับการดำเนินชีวิตของผู้ใช้เฉพาะบุคคลได้

สถิติในเดือนมีนาคม 2558 [5] ระบุว่าระบบปฏิบัติการบนโทรศัพท์อัจฉริยะที่เป็นที่นิยมทั่วโลกซึ่งเห็นได้จากส่วนแบ่งการตลาด [4] ได้แก่ แอนดรอยด์ (Android) (78.0%), iOS ของ iPhone (18.3%), MS Windows Phone (2.7%), และระบบอื่นๆ (1.0%) จะเห็นได้ว่าระบบปฏิบัติการที่ได้รับความนิยมที่สุดคือระบบปฏิบัติการ แอนดรอยด์ และได้รับความนิยมเพิ่มขึ้นอย่างต่อเนื่อง โดยมี Samsung เป็นผู้ผลิตโทรศัพท์เคลื่อนที่อัจฉริยะรายใหญ่ที่สุด โดยเน้นที่การผลิตโทรศัพท์รุ่นประหยัด ตั้งแต่ราคาระดับกลางจนถึงราคาสูง

ดังนั้น ผู้วิจัยจึงต้องการวิจัยและพัฒนาโปรแกรมประยุกต์บนเคลื่อนที่ในระบบปฏิบัติการแอนดรอยด์ซึ่งเป็นระบบปฏิบัติการที่ได้รับความนิยมสูงสุด โดยเฉพาะในกลุ่มผู้มีรายได้น้อยจนถึงผู้มีรายได้ปานกลางซึ่งเป็นผู้ใช้รถโดยสารประจำทางอยู่เป็นประจำ โดยโปรแกรมประยุกต์ที่พัฒนาขึ้นสามารถแสดงข้อมูลเส้นทางสายรถประจำทางตามที่ใช้เลือก แสดงข้อมูลสายรถประจำทางตามป้ายรถประจำ

ทางที่เลือก และค้นหาสายรถประจำทางที่มีระยะทางสั้นที่สุด โดยผู้ใช้กำหนดพิกัดจุดเริ่มต้นและปลายทางบนแผนที่

ถึงแม้ว่าในปัจจุบัน จะมีโปรแกรมประยุกต์ที่ทำหน้าที่คล้ายกับ โปรแกรมประยุกต์ที่จะพัฒนา ซึ่งแบ่งเป็น โปรแกรมประยุกต์บนเว็บ และ โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ โปรแกรมประยุกต์ที่จะพัฒนาในงานวิจัยฉบับนี้แตกต่างจากโปรแกรมประยุกต์อื่นที่มีการแสดงผลบนแผนที่ทำให้ผู้ใช้สามารถใช้งานได้ง่าย และผู้ใช้สามารถค้นหาสายรถโดยการระบุพิกัดจุดเริ่มต้นและปลายทาง โดยโปรแกรมจะคำนวณเส้นทางที่มีระยะทางสั้นที่สุดที่รวมถึงเส้นทางที่มีการต่อสายรถให้อัตโนมัติ โปรแกรมจะถูกพัฒนาขึ้นด้วยภาษา Java และ XML บน Android Studio ที่ติดต่อกับโปรแกรมโปรแกรมประยุกต์บนเว็บที่เขียนขึ้นด้วยภาษา PHP ติดต่อกับฐานข้อมูล MySQL โดยภาษาและเครื่องมือทั้งหมดเป็น Open Source จึงไม่มีค่าใช้จ่ายในการใช้งาน นอกจากนี้ ยังมีการใช้อัลกอริทึมของ Dijkstra [10] ในการหาเส้นทางที่มีระยะทางสั้นที่สุด

1.2 วัตถุประสงค์

เพื่อพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ที่สามารถ

- 1.2.1 ให้ข้อมูลสายรถประจำทางในการเดินทางด้วยรถโดยสารประจำทางในสังกัด ขสมก. ได้
- 1.2.2 ให้ข้อมูลป้ายรถประจำทางในการเดินทางด้วยรถโดยสารประจำทางในสังกัด ขสมก. ได้
- 1.2.3 ค้นหาสายรถประจำทางที่ให้เส้นทางที่มีระยะทางสั้นที่สุด เมื่อผู้ใช้ระบุพิกัดจุดเริ่มต้นและปลายทางบนแผนที่ได้

1.3 ขอบเขตของงาน

โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ที่ต้องการวิจัยและพัฒนามีวัตถุประสงค์เพื่อให้ความช่วยเหลือให้กับผู้โดยสารรถประจำทางของ ขสมก. ได้เดินทางได้สะดวกมากขึ้น ซึ่งโปรแกรมค้นหาเส้นทางการเดินทางด้วยรถประจำทางประกอบด้วยคุณสมบัติต่างๆดังนี้

- เมื่อผู้ใช้เลือกสายรถประจำทาง โปรแกรมสามารถแสดงเส้นทางบนแผนที่ ทั้งขาไปและขากลับ พร้อมพิกัดป้ายรถ รายการแสดงป้ายรถ และสัญลักษณ์การเชื่อมต่อกับระบบขนส่งชนิดอื่น ได้
- แสดงป้ายรถในแผนที่ ที่อยู่รัศมี 500 เมตรจากตำแหน่งของผู้ใช้ได้

- เมื่อผู้ใช้เลือกป้ายรถในแผนที่ โปรแกรมสามารถแสดงรายการหมายเลขสายรถประจำทางที่ผ่านป้ายรถที่เลือกได้
- เมื่อผู้ใช้ระบุพิกัดจุดเริ่มต้นและปลายทางบนแผนที่ โปรแกรมสามารถค้นหาสายรถประจำทางที่ให้เส้นทางที่มีระยะทางสั้นที่สุดได้โดยใช้อัลกอริทึมของ Dijkstra หรืออัลกอริทึมอื่นที่เหมาะสมได้ โดยอาจมีการเปลี่ยนสายรถตามความเหมาะสม
- วัดประสิทธิภาพทางเวลาของโปรแกรมบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ 1 รุ่น เมื่อใช้งานผ่าน 3G หรือ WiFi โดยเวลาในการตอบสนองไม่ควรเกินกว่า 5 วินาที ซึ่งเป็นเวลาที่ผู้ใช้ยอมรับได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ผู้โดยสารรถประจำทางของ ขสมก. ที่ใช้งานโปรแกรมสามารถเดินทางด้วยรถประจำทางได้สะดวกมากขึ้น
- 1.4.2 เป็นช่องทางในการให้ข้อมูลการเดินทางด้วยรถประจำทางในเขตกรุงเทพมหานคร
- 1.4.3 เป็นการประชาสัมพันธ์ให้ผู้โดยสารในเขตกรุงเทพมหานครมาใช้งานรถโดยสารประจำทางมากขึ้น
- 1.4.4 เป็นช่องทางการประชาสัมพันธ์มหาวิทยาลัยให้กับบุคคลภายนอกได้ทราบถึงความเข้มแข็งทางวิชาการ

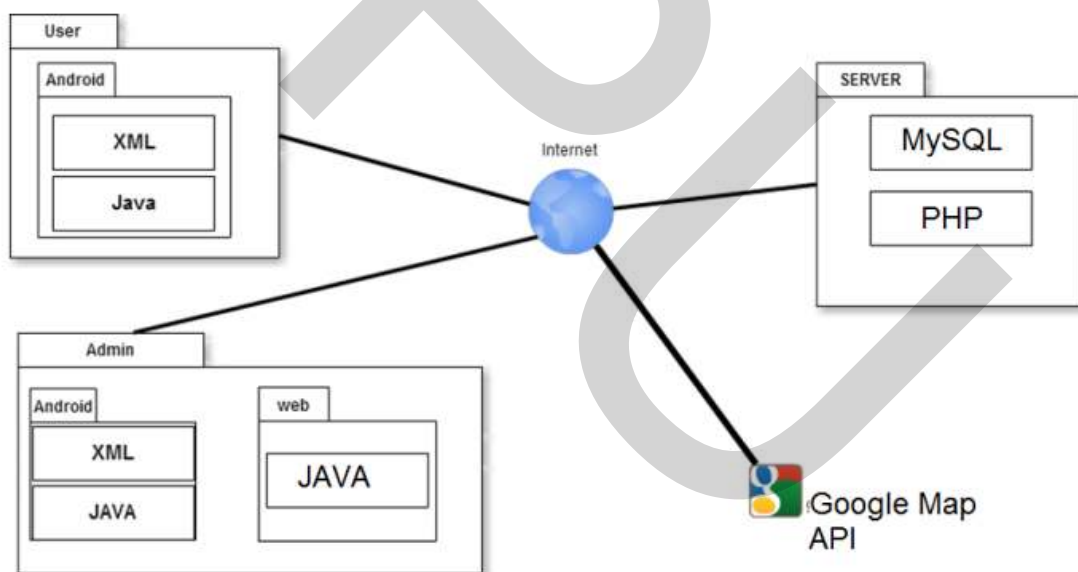
1.5 ข้อจำกัดของการวิจัย

- โปรแกรมสามารถใช้งานได้กับโทรศัพท์อัจฉริยะที่ติดตั้งระบบปฏิบัติการ Android Version 4.4 ขึ้นไปและโทรศัพท์ต้องรองรับการใช้งาน GPS, 3G และ WiFi
- การวัดประสิทธิภาพทางเวลาของโปรแกรม จะเก็บข้อมูลจากโทรศัพท์ 1 รุ่น คือ Samsung Note 4 ที่ใช้ Android v 4.4.4 เนื่องจากข้อจำกัดทางงบประมาณ ได้แก่ ราคาของโทรศัพท์ที่ใช้ในการเก็บข้อมูลและความเสี่ยงที่จะเกิดความเสียหายต่อโทรศัพท์ระหว่างการเก็บข้อมูลการใช้งานจริง

1.6 ข้อตกลงเบื้องต้น

โปรแกรมประยุกต์ที่จะพัฒนาในข้อเสนอวิจัยฉบับนี้ ถูกพัฒนาขึ้นตาม MOU ระหว่างคณะวิศวกรรมศาสตร์ และบริษัท SmartVC International โดยมีคณะวิศวกรรมศาสตร์รับหน้าที่พัฒนาโปรแกรม ในขณะที่บริษัท SmartVC รับผิดชอบการเก็บรวบรวมข้อมูล การตรวจสอบข้อมูล การดูแลฐานข้อมูล การออกแบบส่วนติดต่อผู้ใช้ และการประชาสัมพันธ์โครงการให้ผู้เกี่ยวข้องได้รับทราบ โดยทางบริษัทไม่มีการสนับสนุนด้านเงินทุนการวิจัยและพัฒนาโปรแกรมแต่อย่างใด โดยหลังการพัฒนาลิขสิทธิ์ของโปรแกรมจะเป็นของบริษัท SmartVC International แต่จะมีตราสัญลักษณ์ของมหาวิทยาลัยธุรกิจบัณฑิตย์ปรากฏในโปรแกรมประยุกต์ในฐานะผู้พัฒนา ตามที่ได้ตกลงกันเป็นลายลักษณ์อักษรไว้ใน MOU

1.7 การออกแบบระบบในภาพรวม



รูปที่ 1.4 ภาพรวมของระบบ

ภาพรวมของระบบเก็บข้อมูลป้ายรถและคำนวณเส้นทางการเดินทางรถแสดงในรูปที่ 1.4 โดยระบบประกอบด้วยส่วนโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์สำหรับ

การเก็บข้อมูลพิกัดป้ายรถ โดยผู้ที่ใช้เป็นอาสาสมัคร ที่เขียนด้วยภาษา XML และ Java และ โปรแกรมที่ทำงานบนเครื่องแม่ข่ายสำหรับการคำนวณว่ารถวิ่งผ่านป้ายรถใดบ้างซึ่งเขียนด้วยภาษา Java

ส่วนทางฝั่งเครื่องแม่ข่ายใช้ MySQL เป็นฐานข้อมูล โดยใช้ภาษา PHP สร้างเว็บเซอร์วิสเพื่อให้แอนดรอยด์เรียกใช้ข้อมูลจากฐานข้อมูลได้

ผู้ที่ใช้เป็นอาสาสมัครจะทำการตรวจเช็คข้อมูลพิกัดป้ายรถที่ได้จาก ขสมก. โดยเดินทางไปทีป้ายรถ เปิดโปรแกรมบนแอนดรอยด์ เพื่อดึงข้อมูลป้ายรถจากเครื่องแม่ข่าย หากข้อมูลพิกัดป้ายแสดงผลถูกต้องบนแผนที่ของโปรแกรม ผู้ใช้กดยืนยันข้อมูล หากข้อมูลไม่ถูกต้อง ผู้ใช้สามารถปิดหมุดลงบนแผนที่ของโปรแกรมเพื่อกำหนดพิกัดให้ป้ายใหม่ได้ โปรแกรมจะอัปเดตข้อมูลที่ตรวจเช็คแล้วไปยังฐานข้อมูลบนเครื่องแม่ข่ายทันที ในขณะเดียวกัน ผู้ดูแลระบบสามารถตรวจสอบว่ามีป้ายใดที่ยังไม่ได้มีการตรวจเช็คข้อมูลผ่านโปรแกรมบนแอนดรอยด์ได้อีกด้วย

ส่วนผู้ดูแลระบบจะสร้างเส้นทางการเดินรถประจำทางได้ 3 วิธีคือ

1. สร้างโดยใช้โปรแกรมบนโทรศัพท์แอนดรอยด์ที่ผู้วิจัยได้พัฒนาขึ้นด้วยภาษาจาวา โดยวางโทรศัพท์ไว้บนรถ เมื่อรถออกวิ่งที่ต้นสาย โปรแกรมจะบันทึกพิกัดตลอดเส้นทางการเดินรถจนกระทั่งผู้ใช้กดปุ่มหยุดที่สุดสาย จากนั้นผู้ใช้สามารถอัปเดตพิกัดทั้งหมดไปเก็บไว้ในฐานข้อมูล

2. สร้างโดยใช้โปรแกรม My Maps ของ Google [8] ตามเส้นทางการเดินรถของรถประจำทางแต่ละสาย My Maps จะให้ค่าพิกัดบนเส้นทางที่สร้างให้อัตโนมัติในรูปไฟล์ kml จากนั้นผู้ดูแลระบบจะเรียกใช้โปรแกรมภาษา Java แปลงข้อมูลในไฟล์ kml เป็นพิกัดเส้นทางที่แต่ละจุดห่างกันไม่เกิน 5 เมตรนำมาเป็นข้อมูลเส้นทางการเดินรถ

พิกัดที่ได้จากโปรแกรมในข้อ 1 และ 2 จะถูกนำมาคำนวณร่วมกับข้อมูลป้ายที่ได้จากโทรศัพท์แอนดรอยด์ว่ารถแต่ละสายวิ่งผ่านป้ายใดบ้าง โดยใช้ขั้นตอนวิธีที่อธิบายในบทที่ 3 ต่อไป

3. สร้างโดยใช้โปรแกรมบนเว็บที่ผู้วิจัยพัฒนาขึ้นด้วยภาษา PHP และ Google Map เพื่อลากเส้นเชื่อมระหว่างป้ายรถแต่ละป้าย ระบุสายรถที่ผ่าน แล้วบันทึกลงฐานข้อมูล

ส่วนเอพีไอ(API) และ โปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางที่สั้นที่สุดนั้น โปรแกรมภาษา Java บนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์จะเรียกใช้ข้อมูลเส้นทางในฐานข้อมูล MySQL ผ่านเว็บเซอร์วิสที่เขียนขึ้นด้วยโปรแกรมภาษา PHP ซึ่งคำนวณเส้นทางที่สั้นที่สุดด้วยขั้นตอนวิธีของ Dijkstra ซึ่งทำงานบนเครื่องแม่ข่าย โดยส่งผลลัพธ์การคำนวณในรูปแบบ JSON ไปให้โปรแกรมบนโทรศัพท์แอนดรอยด์แสดงผล

1.8 ภาษาคอมพิวเตอรืที่ใช้ในการพัฒนา

- JAVA
- PHP
- SQL
- XML
- JSON
- Java Script และ JQuery

1.9 อุปกรณ์และซอฟต์แวร์ที่ใช้ในการพัฒนา

1.9.1 อุปกรณ์ที่ใช้ในการพัฒนา

- คอมพิวเตอร์โน้ตบุ๊ก Asus
- คอมพิวเตอร์โน้ตบุ๊ก MSI
- โทรศัพท์ Android รุ่น Samsung Note 4
- โทรศัพท์ Android รุ่น Samsung Galaxy S6 Edge
- โทรศัพท์ Android รุ่น Asus Fonepad 7

1.9.2 ซอฟต์แวร์ที่ใช้ในการพัฒนา

- JDK (Java Development kit 8)
- Android Studio v 1.0.2
- Android SD (Android Software Development kit) v 24.0.2
- ADT (Android Development Tool) v 4.4
- Atom V 1.2.4

บทที่ 2

ทฤษฎีและงานที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

เส้นทางการเดินทางประจำทางที่ระบบที่พัฒนาขึ้นในงานวิจัยนี้ต้องการ เป็นลำดับของพิกัดป้ายรถประจำทาง แต่ข้อมูลที่ระบบนี้เก็บได้ เป็นข้อมูลของพิกัดเส้นทางเดินรถที่เก็บทุก 5 เมตร และพิกัดของป้ายรถทั้งหมดในเขตกรุงเทพมหานครและปริมณฑลซึ่งมีจำนวน 4,781 ป้าย ดังนั้นจึงต้องมีการคำนวณว่าป้ายใดอยู่บนเส้นทางเดินรถแต่ละสายบ้าง ซึ่งมีทฤษฎีที่เกี่ยวข้องดังนี้

2.1.1 ขั้นตอนวิธีของ Dijkstra [10]

ขั้นตอนวิธีของ Dijkstra เป็นขั้นตอนวิธีที่ให้เส้นทางที่สั้นที่สุดในโครงสร้างข้อมูลชนิดกราฟแบบมีทิศทางที่มีน้ำหนักที่ไม่เป็นค่าลบบนแต่ละเส้นเชื่อม โดยจะเริ่มกำหนดค่าให้โหนดเริ่มต้นเป็นศูนย์ และโหนดอื่นเป็นอนันต์ จากนั้นกำหนดให้โหนดเริ่มต้นเป็นโหนดปัจจุบัน แล้วจะพิจารณาเส้นเชื่อมที่เชื่อมต่อกับโหนดนั้นทั้งหมดที่เชื่อมต่อไปยังโหนดที่ยังไม่ไปเยี่ยม แล้วคำนวณน้ำหนักสะสมของเส้นเชื่อมนั้น จากนั้นเลือกโหนดที่ให้ค่าน้ำหนักสะสมรวมน้อยที่สุดเป็นโหนดปัจจุบัน ขยายการคำนวณออกไปเรื่อย ๆ จนกระทั่งพบโหนดปลายทาง หรือไปเยี่ยมโหนดครบทุกโหนดในกราฟเรียบร้อยแล้ว

2.1.2 การคำนวณระยะทางจากกลุ่มพิกัดบนโลก

Google API Team[5] ได้แนะนำวิธีการคำนวณโดยใช้ฐานข้อมูล MySQL แบบ MYISAM และใช้ SQL ด้านล่างกับตาราง markers ที่ประกอบด้วยคอลัมน์ id, name, address, lat, lng ที่สร้างขึ้นเพื่อเก็บพิกัดต่าง ๆ จำนวนมาก

```
SELECT id, ( 6371 * acos( cos( radians(37) ) * cos( radians( lat ) ) * cos( radians( lng ) - radians(-122) ) + sin( radians(37) ) * sin( radians( lat ) ) ) ) ) AS distance FROM markers HAVING distance < 25 ORDER BY distance LIMIT 0 , 20
```

โดย 6371 คือ ระยะทางเป็นกิโลเมตรต่อพิกัดหนึ่งองศา และผลลัพธ์ที่ได้จะเป็นระยะเป็นพิกัดจำนวน 20 ระยะที่มีระยะทางห่างจากพิกัด 37,-122 ไม่เกิน 25 กิโลเมตร

อย่างไรก็ตาม เนื่องจากฐานข้อมูลของระบบที่พัฒนาขึ้นในบทความนี้เป็นฐานข้อมูลแบบมีความสัมพันธ์ ดังจะกล่าวในบทที่ 3 ทำให้ไม่สามารถใช้งานฐานข้อมูลแบบ MYISAM ที่ไม่มีความสัมพันธ์ได้ ดังนั้นจึงไม่สามารถใช้วิธีการดังกล่าวได้

2.1.3 Haversine formula

Haversine formula [6] เป็นสูตรที่ใช้คำนวณระยะห่างระหว่างพิกัด 2 พิกัดบนพื้นโลก (d) โดยไม่สนใจความสูงของพื้นราบ โดยมีสูตรดังนี้

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (2)$$

$$d = R \cdot c \quad (3)$$

โดย φ คือละติจูด, λ คือลองจิจูดของพิกัด, R คือรัศมีโลก ซึ่งมีค่าเฉลี่ยที่ 6,371 กม. และค่ามุมต่าง ๆ อยู่ในหน่วยเรเดียน

2.2 งานวิจัยที่เกี่ยวข้อง

การวางแผนเส้นทาง (Route Planning) คือการแก้ปัญหาเพื่อหาเส้นทางที่ดีที่สุดในการเดินทางจากสถานที่หนึ่งไปยังอีกสถานที่หนึ่ง ซึ่งแต่ละเส้นทางอาจมีระยะทางและค่าใช้จ่ายแตกต่างกัน เป็นที่ทราบกันดีว่าปัญหาประเภทนี้สามารถแก้ได้โดยใช้ขั้นตอนวิธีของ Dijkstra [10] ซึ่งเป็นขั้นตอนวิธีที่ให้เส้นทางที่สั้นที่สุดในโครงสร้างข้อมูลชนิดกราฟแบบมีทิศทางที่มีน้ำหนักบนแต่ละเส้นเชื่อมไม่เท่ากัน โดยจะเริ่มที่โหนดเริ่มต้น จากนั้นจะพิจารณาเส้นเชื่อมที่เชื่อมต่อกับโหนดนั้นทั้งหมด ขยายออกไปเรื่อย ๆ จนกระทั่งพบโหนดปลายทาง

อย่างไรก็ตาม การแก้ปัญหาวางแผนเส้นทางสำหรับเครือข่ายรถสาธารณะมีความแตกต่างจากการแก้ปัญหาค้นหาเส้นทางที่สั้นที่สุดทั่วไป โดยเฉพาะอย่างยิ่ง มีความแตกต่างจากการแก้ปัญหาวางแผนเส้นทางสำหรับเครือข่ายถนน [12] ในขณะที่ เครือข่ายถนนมักเป็นเครือข่ายขนาดใหญ่ครอบคลุมพื้นที่กว้าง เช่น ถนนในทวีปยุโรป หรือ อเมริกาเหนือ ทำให้ต้องจัดการกับข้อมูลนำเข้าจำนวนมาก ขั้นตอนวิธีสมัยใหม่ [13,14,15,16,17] มักแก้ปัญหาที่มีข้อมูลนำเข้าจำนวนมากโดยใช้ 2

ขั้นตอน คือ (1) การคำนวณล่วงหน้าแบบออฟไลน์ (offline preprocessing) และ (2) การสอบถามแบบเวลาจริง จึงจะทำให้การแก้ปัญหาสำหรับเครือข่ายถนนสามารถประมวลผลได้ภายในเวลาที่ผู้ใช้งานยอมรับได้

ในขณะที่เครือข่ายรถสาธารณะต้องพิจารณาทั้งเวลาที่ใช้เดินทาง จำนวนครั้งในการเปลี่ยนรถ เวลาที่ใช้รอการเปลี่ยนรถ และราคาค่าโดยสาร ทำให้งานวิจัยสำหรับเครือข่ายรถสาธารณะมักเน้นไปที่การใช้ทฤษฎีกราฟในการแก้ปัญหา โดยเฉพาะอย่างยิ่ง แบบขยายเวลา (time-expanded) และแบบที่ขึ้นอยู่กับเวลา (time-dependent) [18,19,20]. นอกจากนี้ ยังมีงานวิจัยที่แก้ปัญหานี้โดยที่ไม่ใช้ทฤษฎีกราฟ ได้แก่ RAPTOR[21] ซึ่งใช้การกรองและการคำนวณแบบขนานด้วยโปรเซสเซอร์ที่มีหลายคอร์ ซึ่งสามารถทำงานภายใน 8 ms กับป้ายรถประจำทาง 20,000 ป้ายและเวลาออกรถ 5 ล้านครั้ง

2.3 งานที่เกี่ยวข้อง

ในปัจจุบันมีโปรแกรมประยุกต์ที่ทำหน้าที่ให้ข้อมูลเส้นทางการเดินทางประจำทางในเขตกรุงเทพมหานคร ซึ่งแบ่งเป็น โปรแกรมประยุกต์บนเว็บ และ โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ ระบบปฏิบัติการแอนดรอยด์ ดังนี้

โปรแกรมประยุกต์บนเว็บ ได้แก่

เว็บไซต์ siamtraffic.net [6] ให้ผู้ใช้ระบุจุดเริ่มต้นและปลายทางเป็นชื่อป้ายรถประจำทาง และแสดงผลเป็นข้อความชื่อป้ายที่ต้องขึ้น/ลงรถ พร้อมระยะทาง แต่ไม่มีการแสดงผลบนแผนที่ ซึ่งถ้าหากผู้ใช้งานไม่ทราบชื่อป้าย ก็ไม่อาจทราบได้ว่าต้องขึ้นลงรถที่จุดใด

ในเว็บไซต์ของ ขสมก. [7] ผู้ใช้สามารถค้นหาสายรถ โดยระบุหมายเลขสายรถ ต้นทาง ปลายทาง สถานีที่วิ่งผ่าน หรือคุณสมบัติอื่นๆ โดยแสดงผลเพียง ข้อมูลชื่อป้ายที่แต่ละสายวิ่งผ่าน อย่างไรก็ตาม ผู้ใช้ไม่สามารถค้นหาโดยใช้จุดเริ่มต้นและปลายทางที่ต้องการเดินทางได้ และไม่มีการแสดงผลบนแผนที่ จึงทำให้การใช้งานไม่สะดวกนัก

Google Maps [8] เป็นบริการของ Google ที่สามารถให้ผู้ใช้กำหนดพิกัดจุดเริ่มต้นและปลายทางบนแผนที่ จากนั้นผู้ใช้สามารถเลือกบริการการขนส่งที่ต้องการ โปรแกรมจะแสดงวิธีการเดินทาง เช่นสายรถและจุดเปลี่ยนสายรถให้ อย่างไรก็ตาม ข้อมูลบนแผนที่ google ถูกเก็บโดยใช้หลักการ crowd sourcing ซึ่งไม่มีการตรวจสอบความถูกต้อง

ส่วนโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ [9] มีดังต่อไปนี้

“รถเมล์”, “สายรถเมล์”, “จอดป้ายหน้า”, และ “รถโดยสาร” เป็นแอปพลิเคชันที่นำข้อมูลจากเว็บไซต์ ขสมก. มาแสดงผลบนหน้าจอโทรศัพท์ และสามารถค้นหาโดยการกรอกสายรถ หรือ ชื่อป้ายรถ ได้ แต่ไม่สามารถค้นหาสายรถในการเดินทางที่ต้องมีการต่อรถได้ และไม่มีผลการแสดงผลบนแผนที่ ทำให้ผู้ใช้ใช้งานได้ไม่สะดวกนัก

ส่วน “แผนที่รถเมล์”, Bangkok Buses, และ RoadMay สามารถแสดงผลบนแผนที่ ตามชื่อสายรถ ชื่อป้ายรถ หรือชื่อถนนที่ผู้ใช้เลือกได้ เมื่อผู้ใช้เลือกสายรถ โปรแกรมจะแสดงเส้นทางบนแผนที่ ทั้งขาไปและขากลับ และแสดงป้ายรถใกล้กับตำแหน่งของผู้ใช้ได้ แต่ไม่สามารถค้นหาเส้นทางโดยกำหนดจุดเริ่มต้นและปลายทางที่ต้องการได้

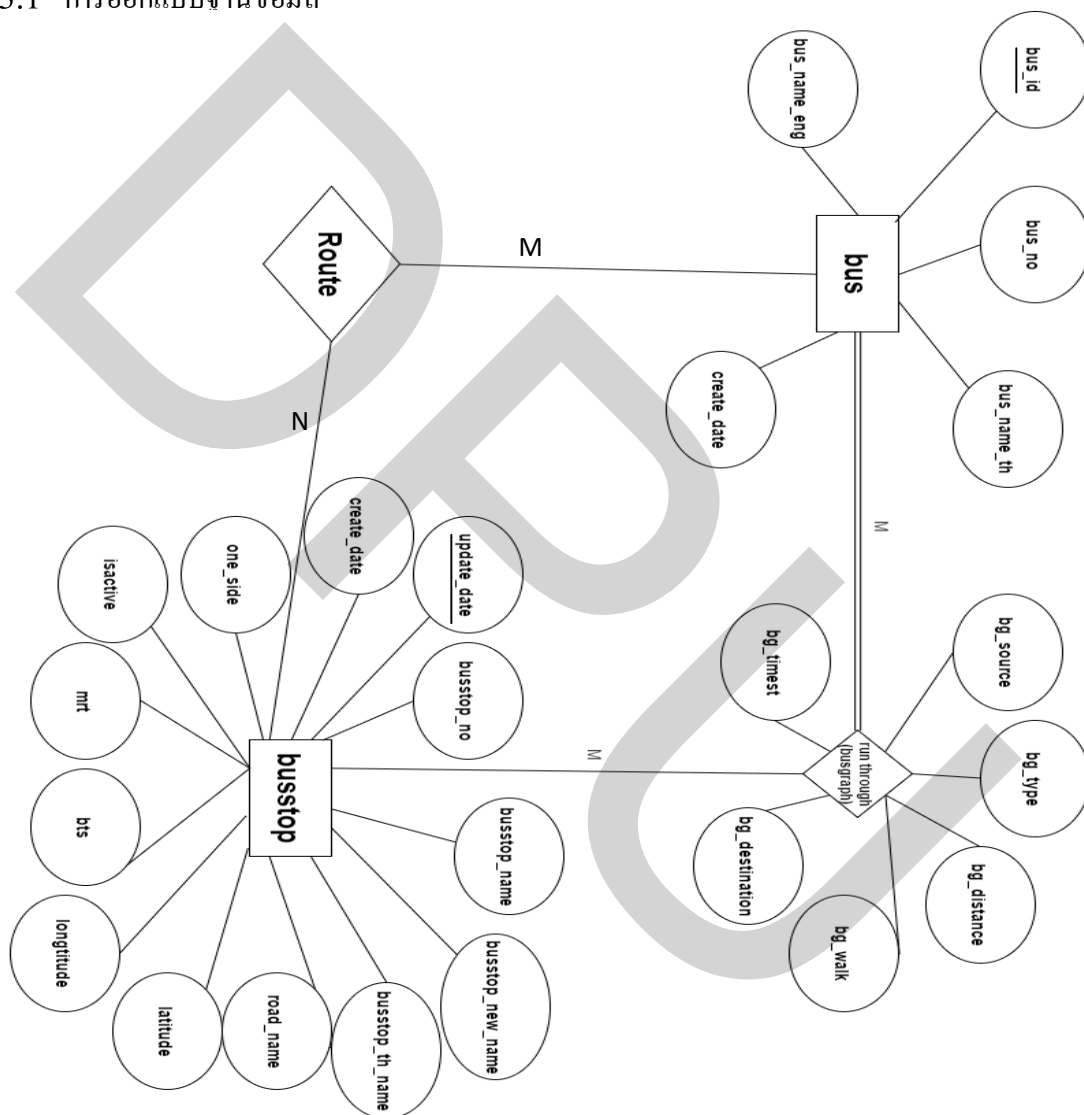
ส่วน โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์ที่เป็นที่นิยมในต่างประเทศ ได้แก่ NAVITIME ที่สามารถใช้งานในประเทศต่าง ๆ ในเอเชีย สหรัฐอเมริกาและยุโรป แต่ครอบคลุมเพียงรถไฟฟ้านั้น ส่วน Transit App ที่สามารถใช้งานในเมืองต่าง ๆ ในสหรัฐอเมริกา แคนาดา ยุโรป และออสเตรเลีย และรวมเอาการเดินทางสาธารณะหลายชนิดมาคำนวณ อย่างไรก็ตาม แอปพลิเคชันเหล่านี้ ได้ใช้ข้อมูลตารางเวลาที่พาหนะแต่ละประเภทมาถึงป้ายรถมาคำนวณด้วย ซึ่งต่างจากการเดินทางในกรุงเทพมหานคร ที่ไม่สามารถทราบได้ว่ารถจะมาถึงป้ายเมื่อไร จึงใช้แนวคิดต่างกัน ในการพัฒนาขึ้นตอนวิธี

อย่างไรก็ตาม โปรแกรมประยุกต์เหล่านี้ใช้ข้อมูลจากฐานข้อมูลของ ขสมก. ซึ่งไม่ได้มีการปรับปรุงข้อมูลให้เป็นปัจจุบันเป็นเวลานาน ทำให้ข้อมูลที่ให้กับผู้ใช้คลาดเคลื่อนจากความเป็นจริง รวมทั้งผู้ใช้ไม่สามารถส่งข้อมูลที่ถูกต้องกลับไปยังโปรแกรมประยุกต์เหล่านี้ได้ ดังนั้นจึงมีความจำเป็นอย่างยิ่งที่ต้องทำการสำรวจข้อมูลให้ถูกต้องโดยระบบที่พัฒนาขึ้นในบทความนี้ ก่อนที่จะนำข้อมูลไปใช้งาน โดยโปรแกรมประยุกต์ต่าง ๆ ข้างต้น

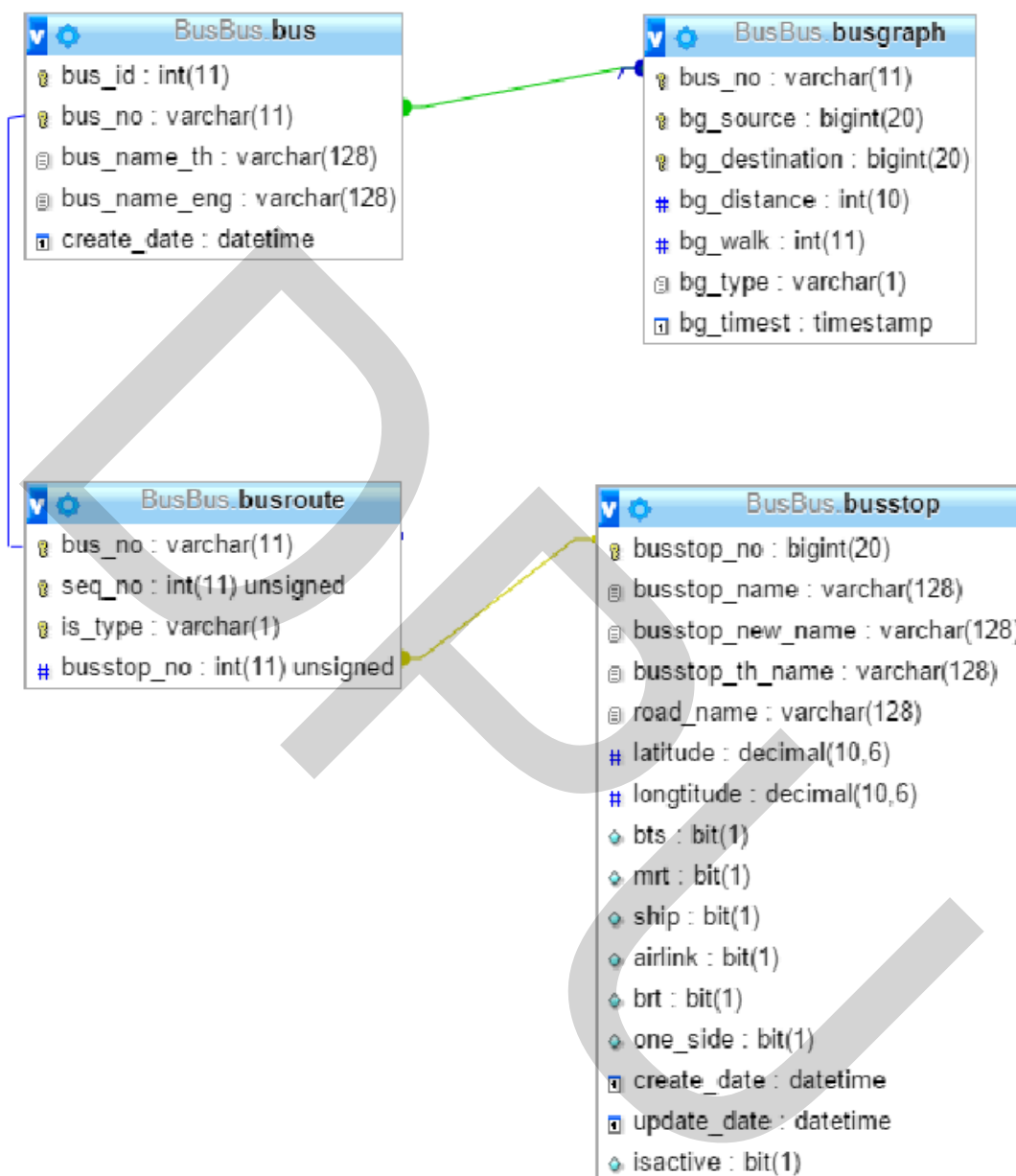
บทที่ 3

การออกแบบและพัฒนา

3.1 การออกแบบฐานข้อมูล



รูปที่ 3.1 แผนภาพอีอาร์ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด



รูปที่ 3.2 แผนภาพอ็อร์ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด

รูปที่ 3.1 และ รูปที่ 3.2 แสดงแผนภาพอ็อร์ของระบบ โดยในรูปที่ 3.2 แสดงชนิดและขนาดของข้อมูลในแต่ละแอทริบิวต์ รวมถึงคีย์หลักของแต่ละตารางความสัมพันธ์

3.2 พจนานุกรมข้อมูลของฐานข้อมูลระบบ

แต่ละตารางความสัมพันธ์เก็บข้อมูลในแต่ละแอทริบิวต์ ดังต่อไปนี้

ตารางที่ 3.1 ตาราง Busstop

Column	Type	Comments
busstop_no	bigint(20)	เลขป้าย
busstop_name	varchar(128)	ชื่อป้าย
busstop_new_name	varchar(128)	ชื่อป้ายใหม่
busstop_th_name	varchar(128)	ชื่อไทย
road_name	varchar(128)	ชื่อถนน
latitude	decimal(10,6)	ละติจูด
longtitude	decimal(10,6)	ลองจิจูด
bts	bit(1)	เชื่อมต่อกับ bts หรือไม่
mrt	bit(1)	เชื่อมต่อกับ mrt หรือไม่
ship	bit(1)	เชื่อมต่อกับเรือหรือไม่
airlink	bit(1)	เชื่อมต่อกับ airport link หรือไม่
brt	bit(1)	เชื่อมต่อกับ brt หรือไม่
one_side	bit(1)	เดินรถทางเดียวหรือไม่
create_date	datetime	เวลาสร้างระเบียบ
update_date	datetime	เวลาอัปเดต
isactive	bit(1)	การใช้งานป้าย

ตารางที่ 3.2 ตาราง Bus

Column	Type	Links to	Comments
<u>bus_id</u>	int(11)		เลขสายรถประจำทาง
bus_no	varchar(11)	busgraph -> bus_no	เลขเส้นสายรถประจำทาง
bus_name_th	varchar(128)		ชื่อสายรถประจำทางภาษาไทย
bus_name_eng	varchar(128)		ชื่อสายรถประจำทางภาษาอังกฤษ
create_date	datetime		เวลาตอนสร้างระเบียบ

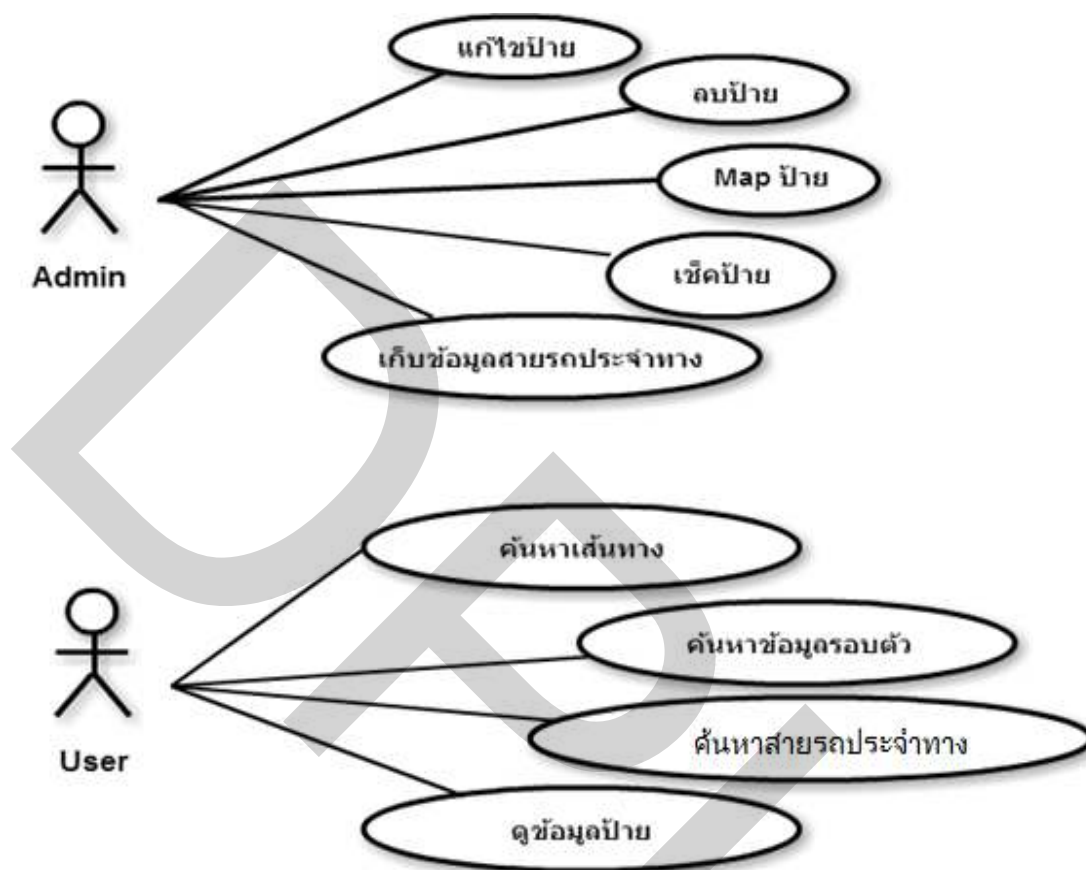
ตารางที่ 3.3 ตาราง Busgraph

Column	Type	Links to	Comments
<u>bus_no</u>	varchar(11)	busroute -> bus_no	เลขสายรถประจำทาง
bg_source	bigint(20)		ป้ายต้นทาง
bg_destination	bigint(20)		ป้ายปลายทาง
bg_distance	int(10)		ระยะทางของรถประจำทาง
bg_walk	int(11)		เช็คว่าเดินได้ไหม
bg_type	varchar(1)		ชนิดของการกระทำ
bg_timest	timestamp		

ตารางที่ 3.4 ตาราง Busroute

Column	Type	Links to	Comments
<u>bus_no</u>	varchar(11)	bus -> bus_no	เลขเส้นสายรถประจำทาง
seq_no	int(11)		หมายเลขของเส้นทาง
is_type	varchar(1)		ชนิด
<u>busstop_no</u>	int(11)	busstop -> busstop_no	เลขป้าย

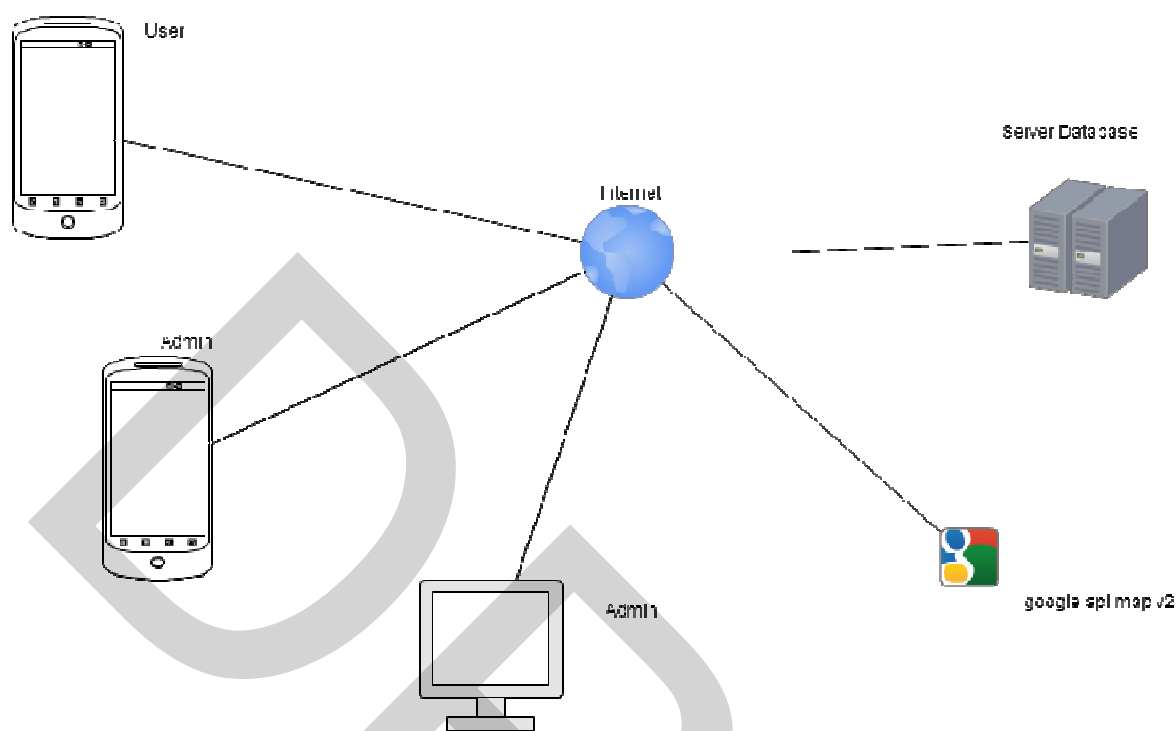
3.3 Use-case



รูปที่ 3.3 use-case ของระบบ การพัฒนา โปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด

3.4 สถาปัตยกรรมระบบ

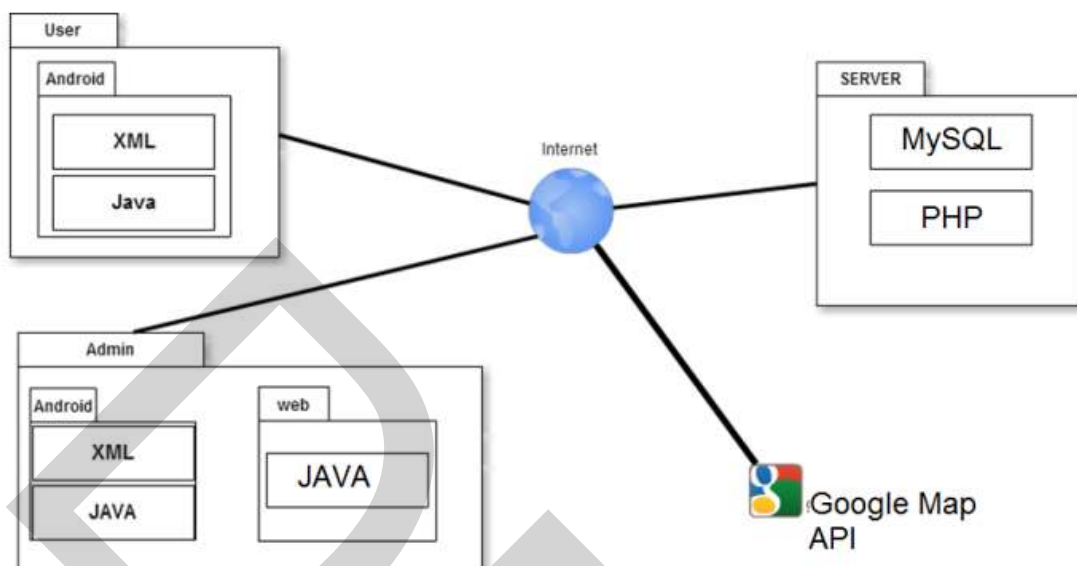
ผู้ใช้ทั่วไป (User) ที่ใช้ระบบปฏิบัติการแอนดรอยด์สามารถดูข้อมูลสายรถประจำทางและสามารถค้นหาสายรถที่จะเดินทางด้วยเส้นทางที่สั้นที่สุด ส่วนผู้ดูแลระบบ (Admin) สามารถเก็บข้อมูลสายรถประจำทางและป้ายรถประจำทางลงฐานข้อมูลในเครื่องแม่ข่ายได้ผ่านโทรศัพท์แอนดรอยด์ นอกจากนี้ผู้ดูแลระบบยังสามารถแก้ไขข้อมูลบน web application ได้อีกด้วย



รูปที่ 3.4 สถาปัตยกรรมฮาร์ดแวร์ของระบบ

เพื่อตรวจสอบข้อมูลพิกัดป้ายรถที่ได้จาก ขสมก. ผู้ใช้ที่เป็นอาสาสมัครจะเดินทางไปป้ายรถด้วยตนเอง แล้วเปิดโปรแกรมบนแอนดรอยด์ เพื่อดึงข้อมูลป้ายรถจากเครื่องแม่ข่ายผ่าน Web Service หากข้อมูลพิกัดป้ายแสดงผลถูกต้องบนแผนที่ของโปรแกรม ผู้ใช้กดยืนยันข้อมูล หากข้อมูลไม่ถูกต้อง ผู้ใช้สามารถปัดหมุดลงบนแผนที่ของโปรแกรมเพื่อกำหนดพิกัดให้ป้ายใหม่ได้ โปรแกรมจะอัปเดตข้อมูลที่ตรวจสอบแล้วไปยังฐานข้อมูลบนเครื่องแม่ข่ายทันที ในขณะเดียวกัน ผู้ดูแลระบบสามารถตรวจสอบว่ามีป้ายใดที่ยังไม่ได้มีการตรวจเช็คข้อมูลผ่านโปรแกรมบนแอนดรอยด์ได้อีกด้วย

เพื่อสร้างข้อมูลเส้นทางการเดินทางของรถประจำทางแต่ละสาย ผู้ดูแลระบบ (Admin) จะใช้โปรแกรมสำรวจเส้นทางการเดินทางของรถประจำทางบนแอนดรอยด์เก็บข้อมูลพิกัดการเดินทางแต่ละสาย นอกจากนี้ เพื่อเพิ่มความสะดวกในการเก็บข้อมูล ผู้สำรวจยังสามารถกรอกข้อมูลการเดินทางแต่ละสายเพิ่มเติมบนเว็บได้อีกด้วย



รูปที่ 3.5 สถาปัตยกรรมซอฟต์แวร์ของระบบ

สถาปัตยกรรมซอฟต์แวร์ของระบบแสดงในรูปที่ 3.5 โดยระบบประกอบด้วย 5 ส่วนได้แก่

1. โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์สำหรับการเก็บข้อมูลพิกัดป้ายรถโดยผู้ใช้ที่เป็นอาสาสมัคร (User) ที่เขียนด้วยภาษา XML และ Java และเรียกใช้ Google Map API เพื่อแสดงผลพิกัดป้ายรถบนแผนที่
2. โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์สำหรับผู้ดูแลระบบ (Folder Admin//Android) ใช้เก็บข้อมูลสายรถประจำทาง
3. โปรแกรมสำหรับผู้ดูแลระบบที่ทำงานบนเครื่องแม่ข่าย (Folder Admin//Web) สำหรับสร้างเส้นทางสายรถประจำทางว่าผ่านป้ายรถใดบ้างซึ่งเขียนด้วยภาษาจาวา
4. โปรแกรมสำหรับผู้ดูแลระบบที่ทำงานบนเครื่องแม่ข่าย (Server) สำหรับกรอกข้อมูลการเดินทางของสายรถประจำทางซึ่งเขียนด้วยภาษา PHP
5. ฐานข้อมูลฝั่งเครื่องแม่ข่าย (Server) ใช้ MySQL เป็นโปรแกรมจัดการฐานข้อมูล โดยใช้ภาษา PHP สร้างเว็บเซอร์วิสเพื่อให้โปรแกรมประยุกต์บนระบบปฏิบัติการแอนดรอยด์เรียกใช้ฐานข้อมูลได้

3.5 ข้อกำหนดซอฟต์แวร์

ตารางที่ 3.5-3.13 แสดงข้อกำหนดซอฟต์แวร์ผ่าน Structured Language Specification ซึ่งอธิบายตาม Use-case แต่ละอันที่ได้ระบุไว้ในรูปที่ 3.3 โดยตารางที่ 3.5-3.9 อธิบายการทำงานของผู้อัปเดตระบบ และตารางที่ 3.10-3.13 อธิบายการทำงานของผู้ใช้ทั่วไป

ตารางที่ 3.5 ตาราง Structured Language Specification ของ Use-case แก้ไขป้าย	
System	การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	แก้ไขป้าย
Actor	Admin
Action	การแก้ไขป้ายที่ผิดพลาดหรือข้อมูลป้ายไม่ตรง 1. การเลือกป้าย 2. ทำการแก้ไขข้อมูล
Stimulus	ผู้ใช้ทำการเลือกป้ายที่ต้องการแก้ไข
Response	ข้อมูลที่ถูกแก้ไขจะถูกส่งไปยัง Server และ แสดงสีเขียวที่ป้ายบนแผนที่

ตารางที่ 3.6 ตาราง Structured Language Specification ของ Use-case Map ป้าย	
System	การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	Map ป้าย (web site)
Actor	Admin
Action	1. ผู้ใช้ทำการเลือกสายรถที่ต้องการ Map ป้าย 2. กดปุ่มป้ายที่ต้องการ Mapping 3. ทำการส่งข้อมูล
Stimulus	ผู้ใช้เลือก Mapping สายรถ
Response	ระบบส่งข้อมูลไปยัง Server Data base แล้ว Update

ตารางที่ 3.7 ตาราง Structured Language Specification ของ Use-case เช็คป้าย	
System	การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	เช็คป้าย
Actor	Admin
Action	<p>ผู้ใช้ทำการเลือกป้ายที่ต้องการและ set ค่า</p> <ol style="list-style-type: none"> 1. ผู้ใช้เลือกป้าย 2. กดถูกต้อง
Stimulus	ทำการเลือกป้ายที่ต้องการ
Response	ระบบส่งข้อมูลไปยัง Server และแสดงสีเขียวที่ป้ายบนแผนที่

ตารางที่ 3.8 ตาราง Structured Language Specification ของ Use-case เก็บข้อมูลสายรถ	
System	การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	เก็บข้อมูลสายรถ
Actor	Admin
Action	<p>ระบบทำการเก็บข้อมูล Latitude Longitude ที่รถประจำทางวิ่งผ่านทั้งหมด ทุก ๆ 1 วินาที และนำไปเก็บไว้ในตัวเครื่อง</p> <ol style="list-style-type: none"> 1. ผู้ใช้ทำการกดเริ่ม 2. ผู้ใช้กดหยุด ระบบจะเก็บข้อมูลใน 3. ผู้ใช้กดปุ่มเพื่อส่งข้อมูลไปเก็บบน Server
Stimulus	ผู้ใช้เริ่ม โดยทำการกดเริ่ม
Response	ระบบเก็บตำแหน่งข้อมูลบนสมาร์ตโฟนและส่งไปเก็บบน Server เมื่อผู้ใช้อัปโหลด

ตารางที่ 3.9 ตาราง Structured Language Specification ของ Use-case ลบป้าย	
System	การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	ลบป้าย
Actor	Admin
Action	<p>ผู้ใช้ทำการเช็คป้ายถูกต้องหรือไม่ ถ้าไม่สามารถทำการกดปุ่มลบป้าย ออกจาก map</p> <ol style="list-style-type: none"> 1. กดที่ปุ่มลบป้าย 2. กดตกลง
Stimulus	ผู้ใช้กดปุ่มเพื่อลบป้าย
Response	ระบบทำการลบป้ายออกจากฐานข้อมูล Data base

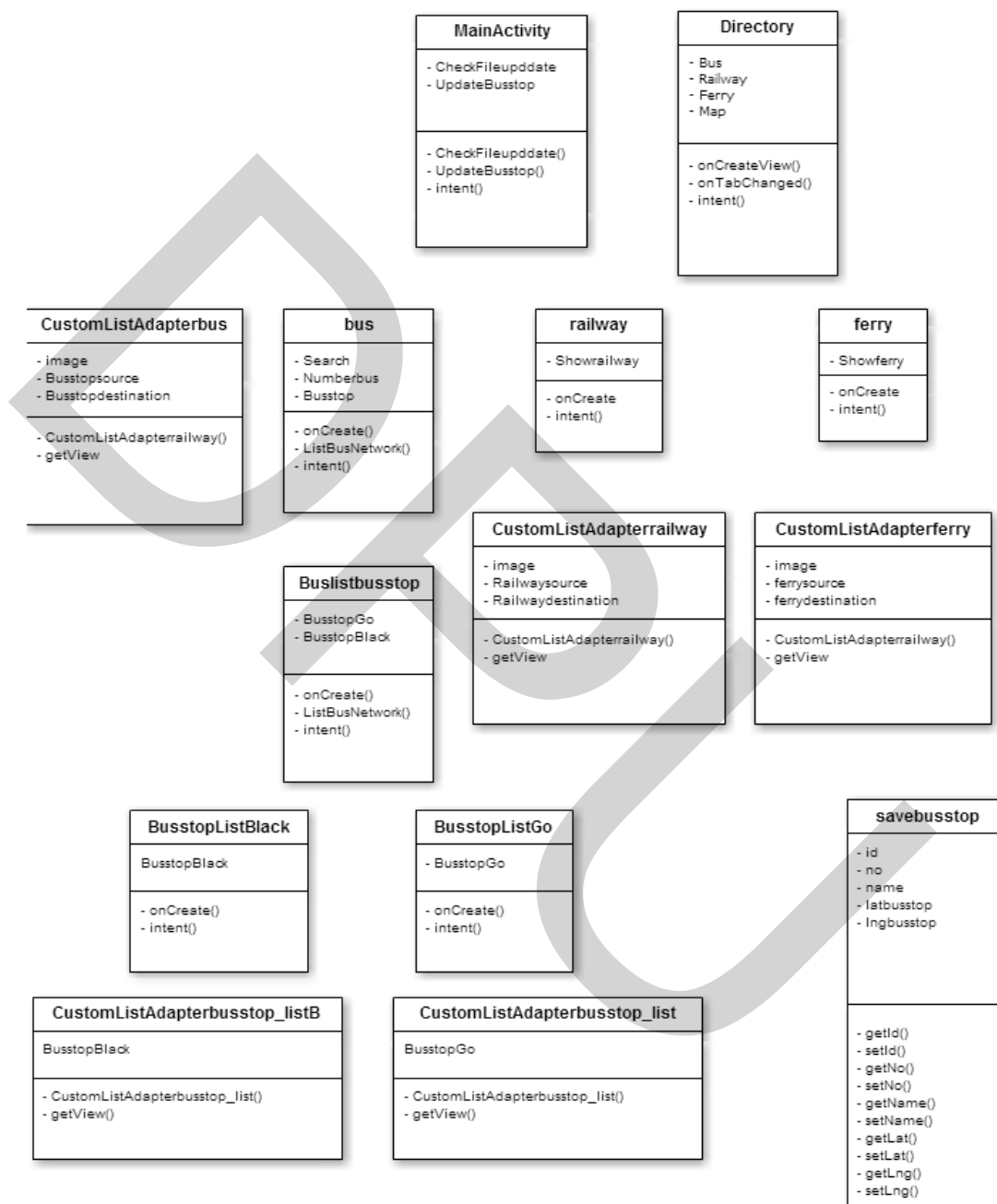
ตารางที่ 3.10 ตาราง Structured Language Specification ของ Use-case ค้นหาปลายทาง	
System	การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	ค้นหาปลายทาง
Actor	User
Action	<p>ผู้ใช้สามารถเลือกปลายทางที่จะไป เพื่อดูเส้นทางและสายรถประจำทางได้</p> <ol style="list-style-type: none"> 1. ผู้ใช้สามารถเลือกปลายทางที่จะไป 2. กด route เส้นทาง 3. App จะแสดงเส้นทาง และ สายรถประจำทางที่จะไป
Stimulus	ผู้ใช้เลือกจุดหมายที่จะไป
Response	ระบบจะคำนวณหาเส้นทางและสายรถที่ต้องขึ้น

ตารางที่ 3.11 ตาราง Structured Language Specification ของ Use-case ค้นหาป้ายรอบตัว	
System	การพัฒนาโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	ค้นหาป้ายรอบตัว
Actor	User
Action	<p>ผู้ใช้สามารถหาป้ายรอบ ๆ ตัวหาข้อมูลรอบ ๆ ตัว</p> <ol style="list-style-type: none"> กดเพื่อหาข้อมูลรอบ ๆ ตัว Interface จะโชว์ข้อมูลรอบ ๆ ตัวที่ผู้ใช้อยู่
Stimulus	ผู้ใช้งานกดหาข้อมูลรอบ ๆ ตัว
Response	ระบบจะแสดงข้อมูลรอบ ๆ ตัวที่ผู้ใช้อยู่ในระยะ 500 เมตร

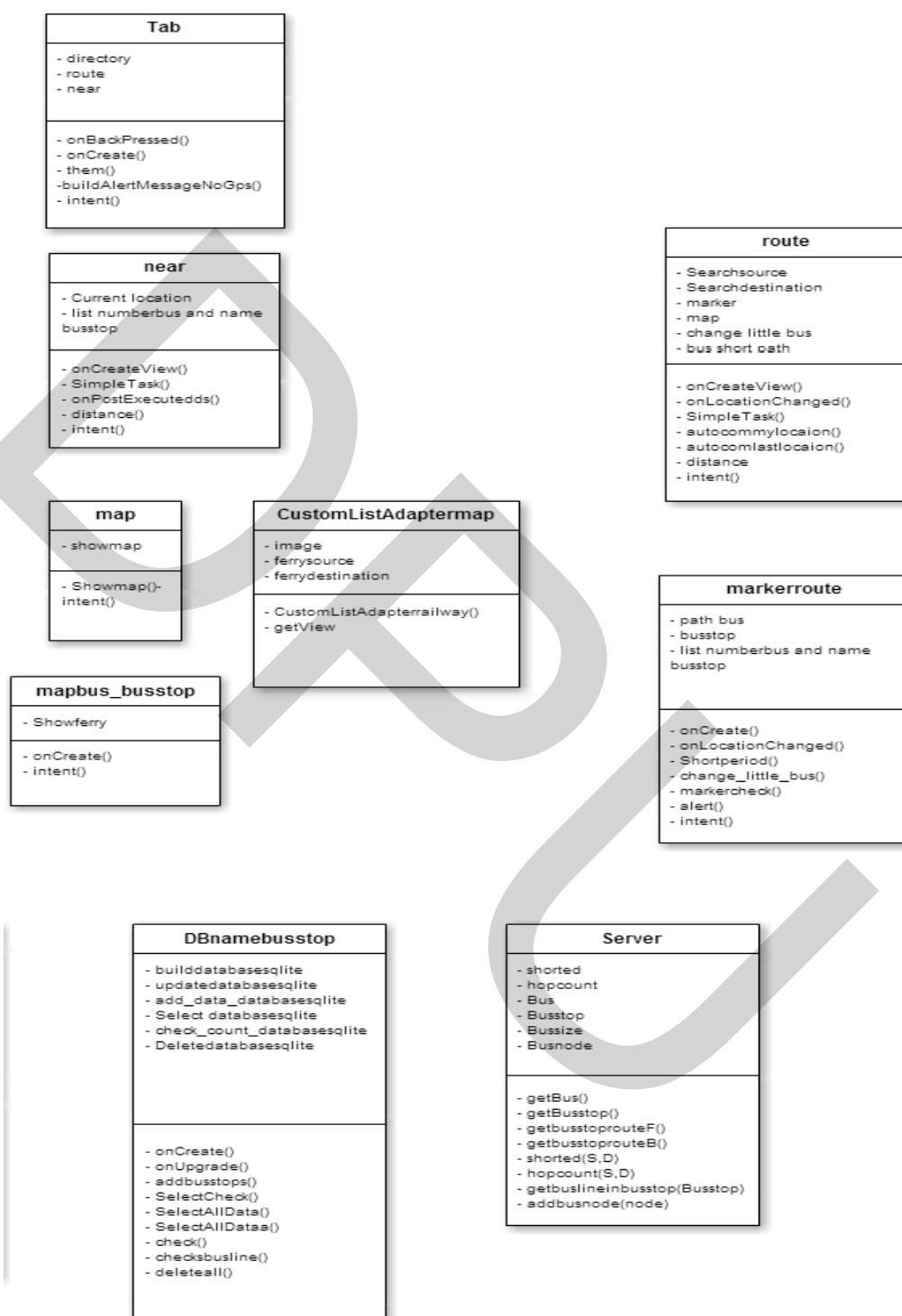
ตารางที่ 3.12 ตาราง Structured Language Specification ของ Use-case ค้นหาสายรถประจำทาง	
System	การพัฒนาโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	ค้นหาสายรถประจำทาง
Actor	User
Action	<p>ผู้ใช้สามารถค้นหาสายรถโดยค้นหาจากเลขสายรถประจำทางที่ต้องการเพื่อดูข้อมูลก่อนเดินทาง</p> <ol style="list-style-type: none"> แสดงสายรถประจำทางเพื่อดูว่าจะผ่านป้ายไหนบ้าง
Stimulus	ผู้ใช้งานกดหาเลขสายรถประจำทางเพื่อดูว่ารถประจำทางผ่านป้ายไหนบ้าง
Response	ระบบจะแสดงเลขสายรถประจำทาง

ตารางที่ 3.13 ตาราง Structured Language Specification ของ Use-case ข้อมูลป้าย	
System	การพัฒนาโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด
Use-case	ข้อมูลป้าย
Actor	User
Action	หน้าจอ interface จะโชว์ป้ายรถประจำทางทั้งหมดเพื่อให้ผู้ใช้ดูว่าป้ายนี้มีสายรถประจำทางไหนวิ่งผ่านบ้าง <ol style="list-style-type: none"> 1. กดที่ป้ายที่ผู้ใช้ต้องการ 2. ป้ายจะโชว์ข้อมูลทั้งหมด
Stimulus	ผู้ใช้ดูได้ว่าป้ายนี้ มีรถประจำทางสายไหนผ่านบ้าง
Response	ระบบจะแสดงข้อมูลของป้ายและสายรถประจำทางที่ผ่านได้

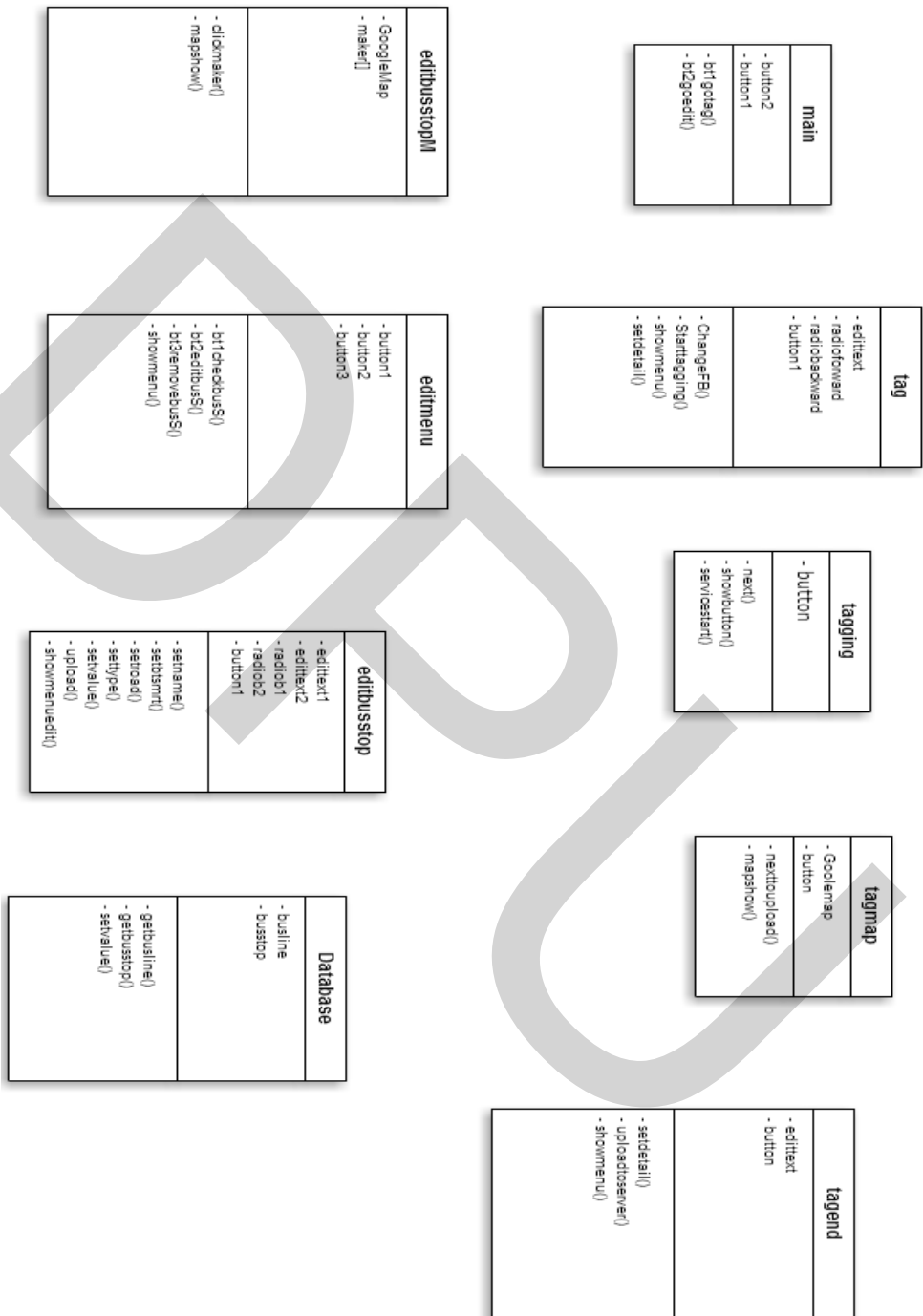
3.6 Class Diagram



รูปที่ 3.6 class diagram ของระบบ การพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด สำหรับ User

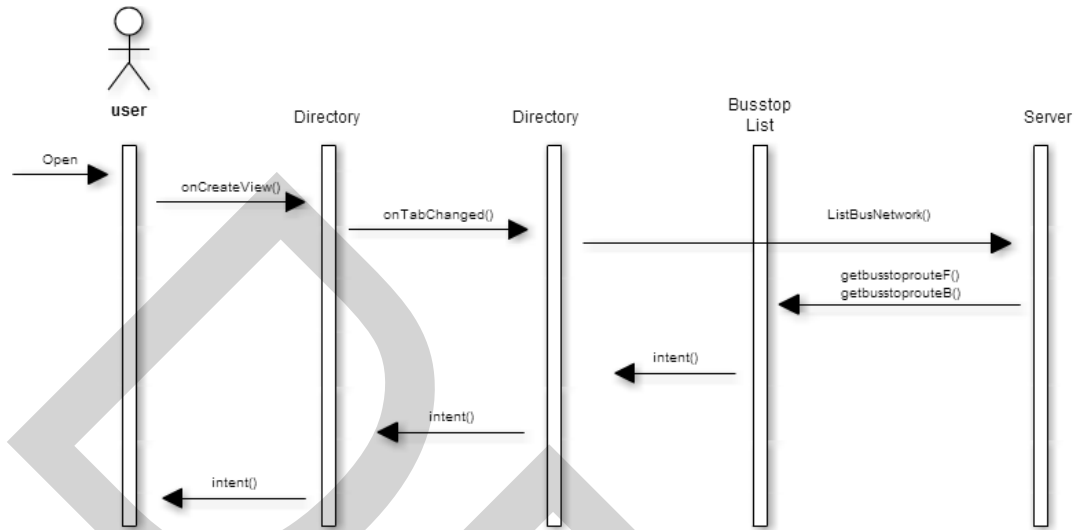


รูปที่ 3.7 class diagram ของระบบ การพัฒนาโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการ แอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด สำหรับUser(ต่อ)

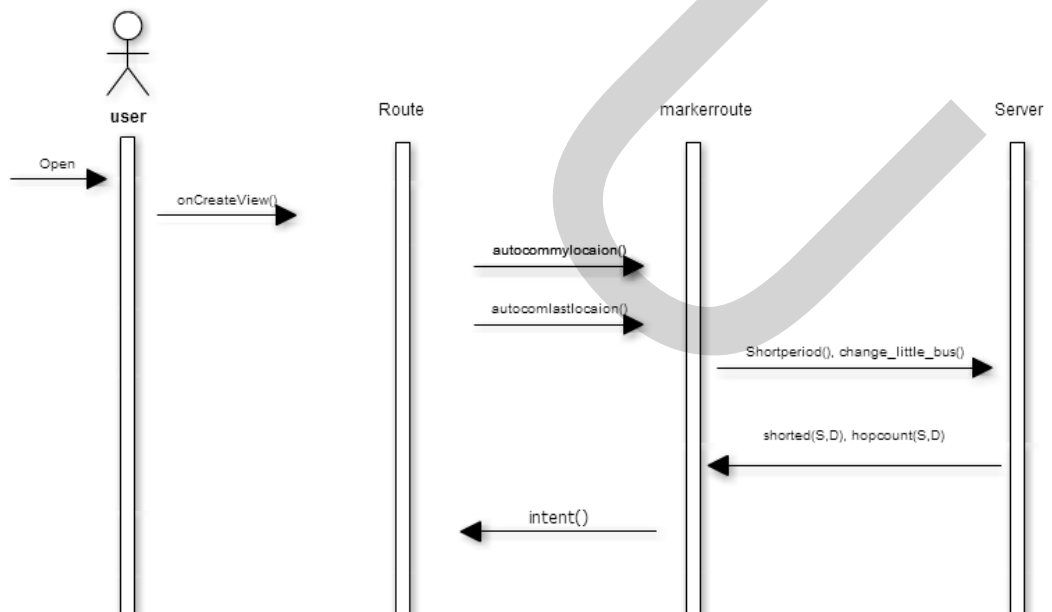


รูปที่ 3.8 class diagram ของระบบ การพัฒนาโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ระบบปฏิบัติการ แอนดรอยด์เพื่อค้นหาเส้นทางรถประจำทางในกรุงเทพมหานครที่มีระยะทางสั้นที่สุด สำหรับ Admin

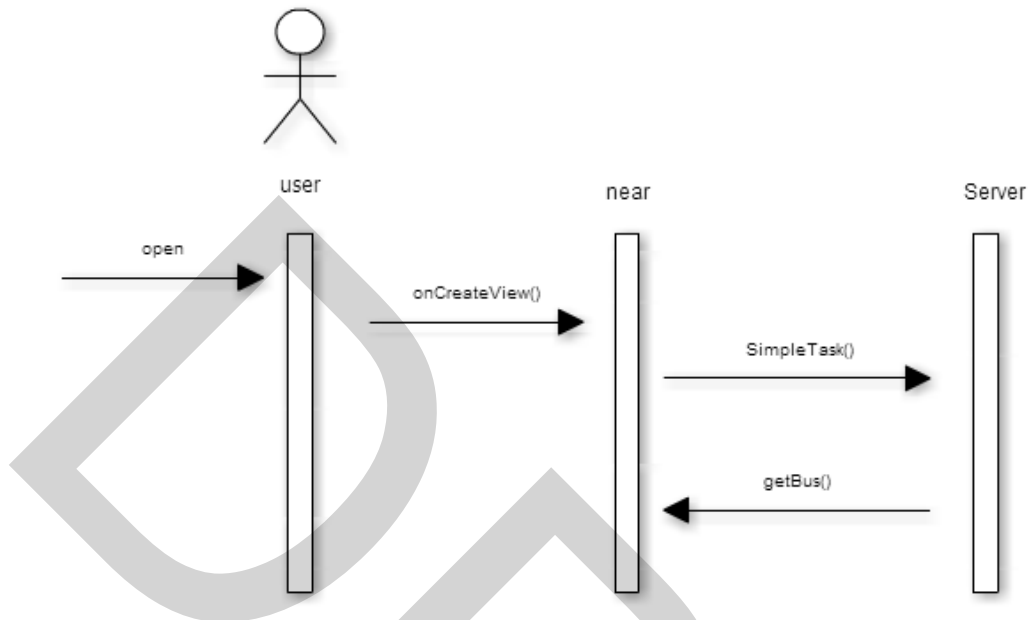
3.7 Sequence Diagram



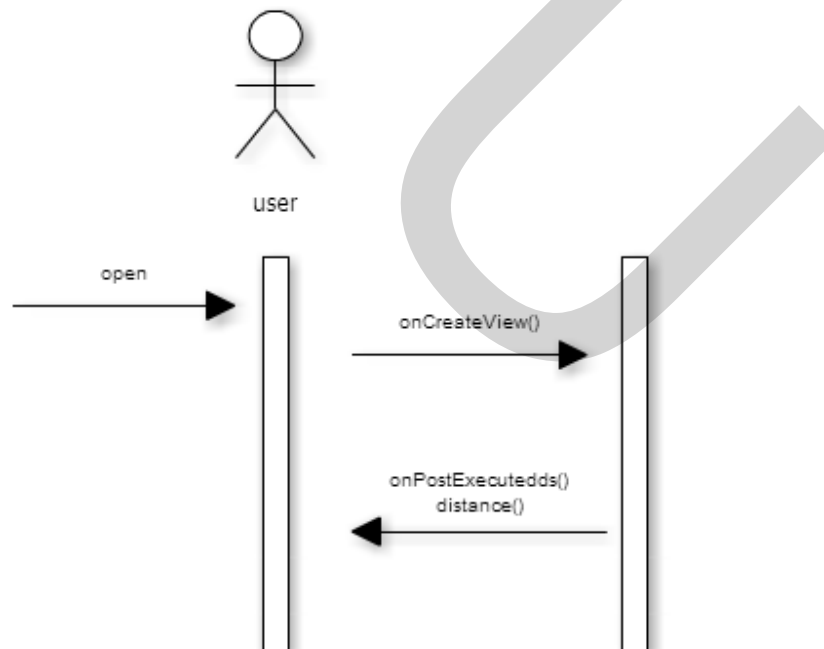
รูปที่ 3.9 Sequence diagram ของ use-case เลือกสายทั้งไปและกลับ (User)



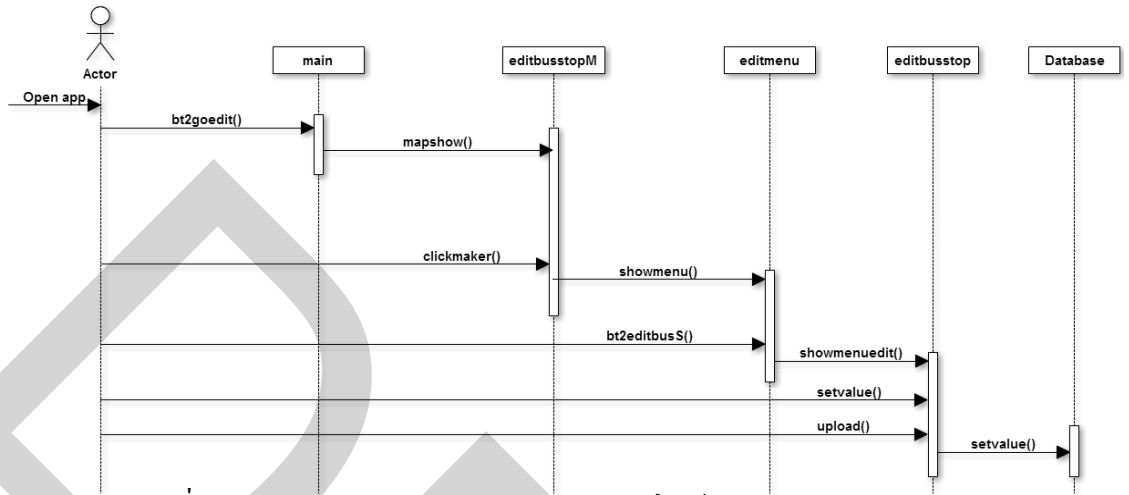
รูปที่ 3.10 Sequence diagram ของ use-case ค้นหาปลายทาง (User)



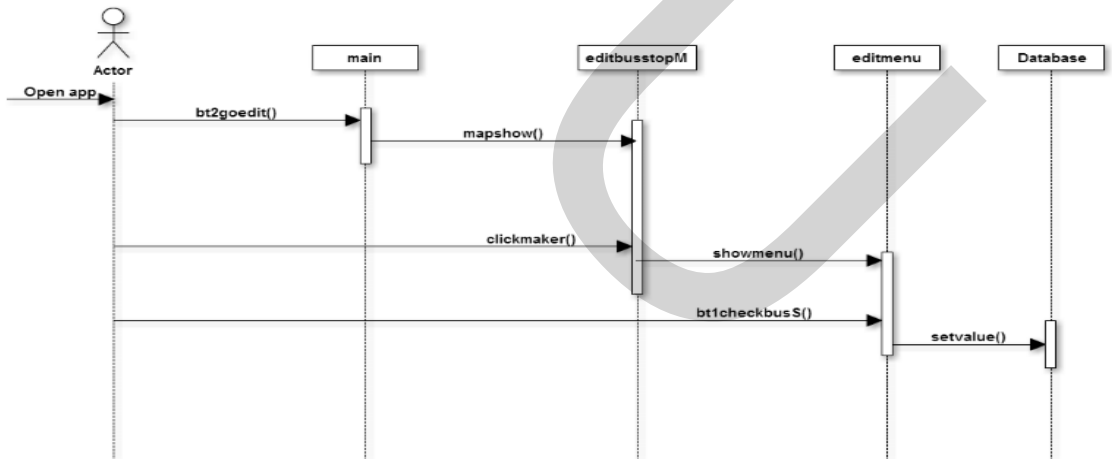
รูปที่ 3.11 Sequence diagram ของ use-case ข้อมูลป้าย (User)



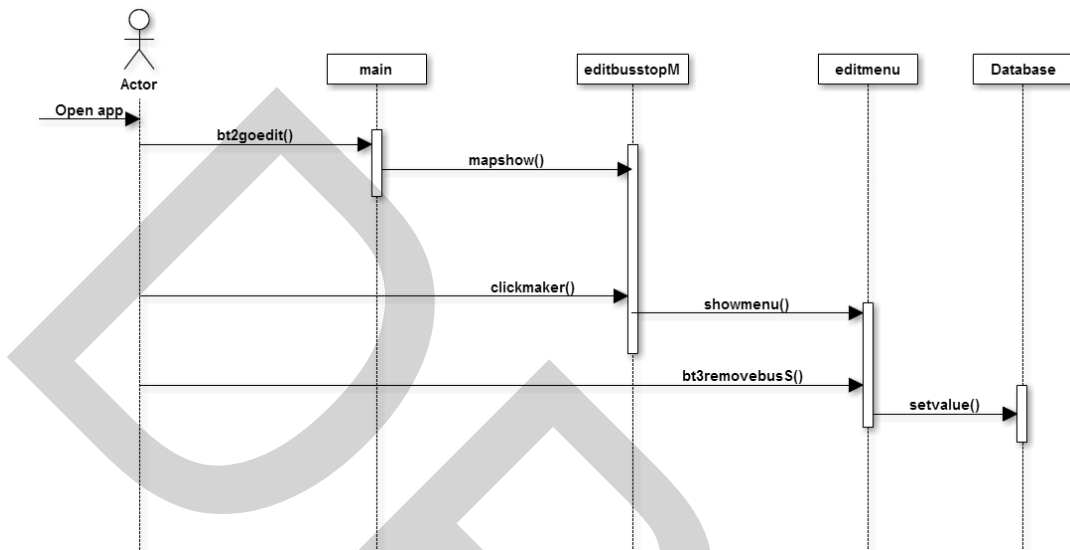
รูปที่ 3.12 Sequence diagram ของ use-case ค้นหาข้อมูลรอบตัว(User)



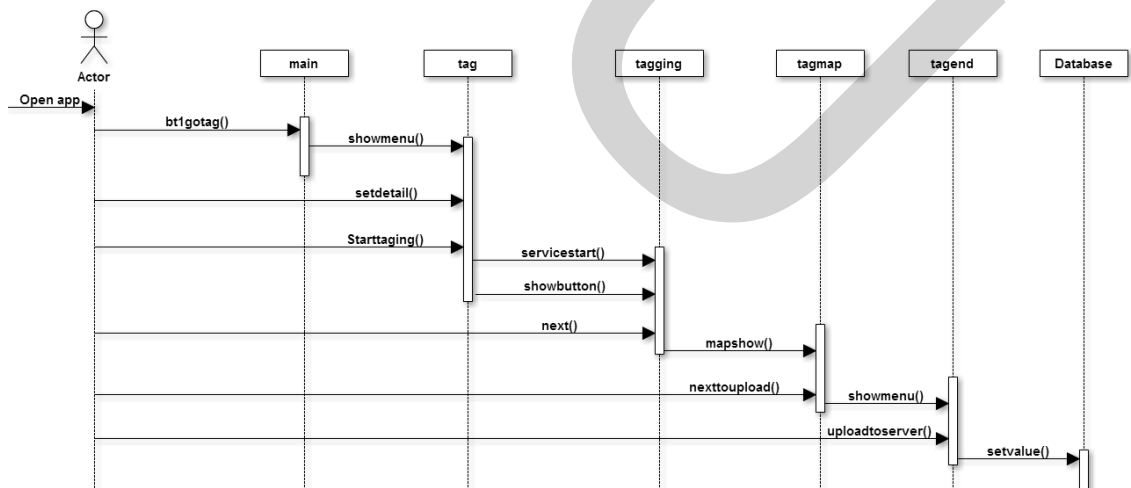
รูปที่ 3.13 Sequence diagram ของ use-case แก้ไขป้าย (Admin)



รูปที่ 3.14 Sequence diagram ของ use-case เช็คน้ำยาถูกต้อง (Admin)

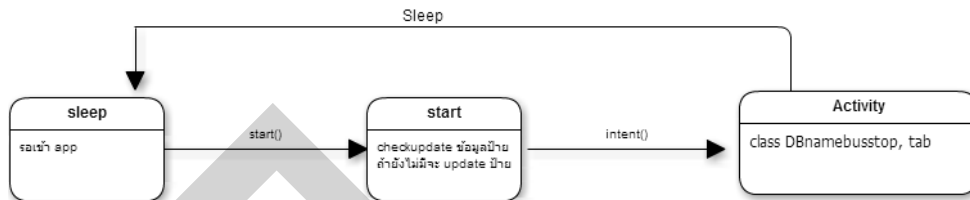


รูปที่ 3.15 Sequence diagram ของ use-case ลบป้าย (Admin)

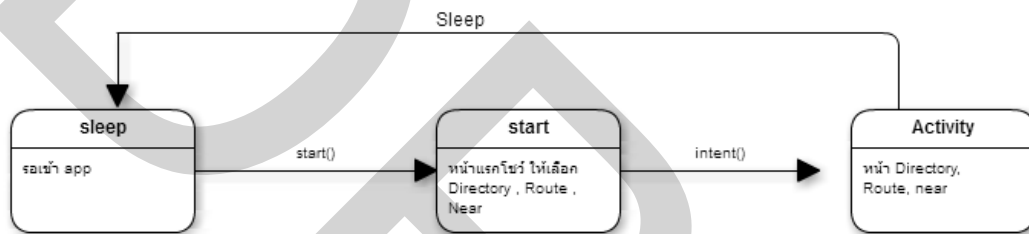


รูปที่ 3.16 Sequence diagram ของ use-case เก็บข้อมูลสายรถ (Admin)

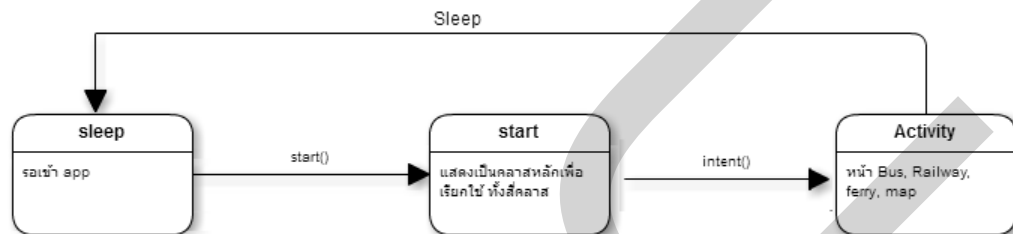
3.8 State Diagram



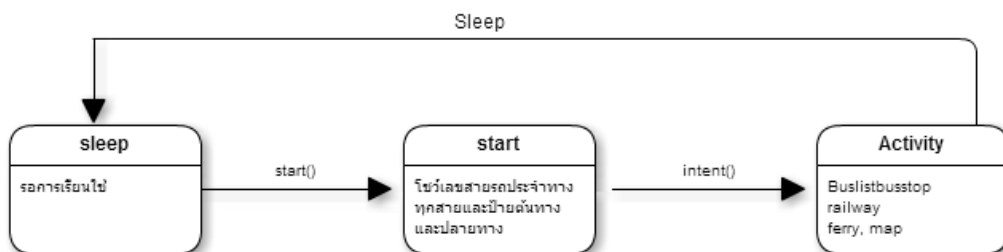
รูปที่ 3.17 state diagram ของ class MainActivity (User)



รูปที่ 3.18 state diagram ของ class Tab (User)



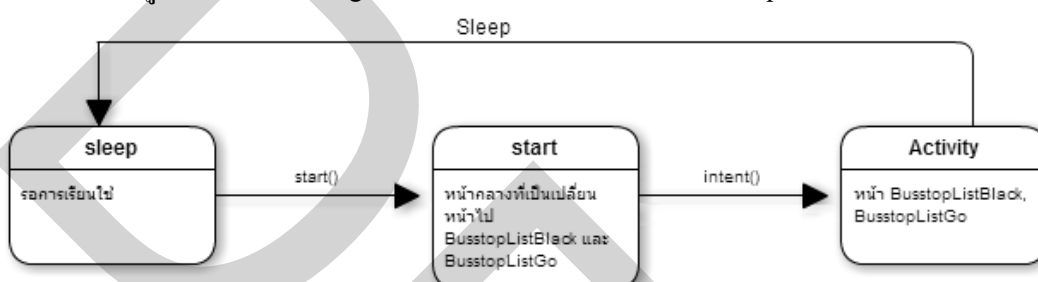
รูปที่ 3.19 state diagram ของ class Directory (User)



รูปที่ 3.20 state diagram ของ class bus (User)



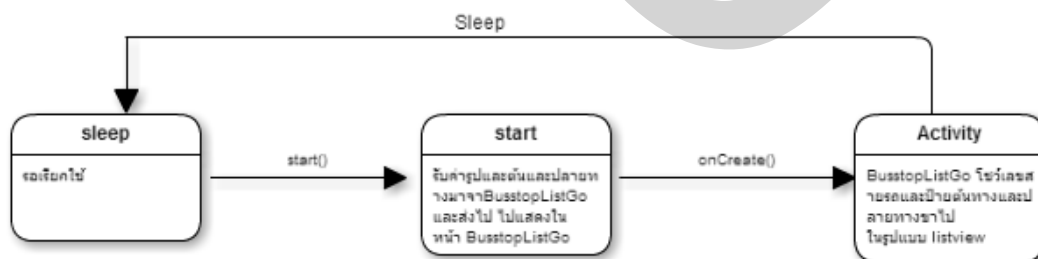
รูปที่ 3.21 state diagram ของ class result CustomListAdapterbus (User)



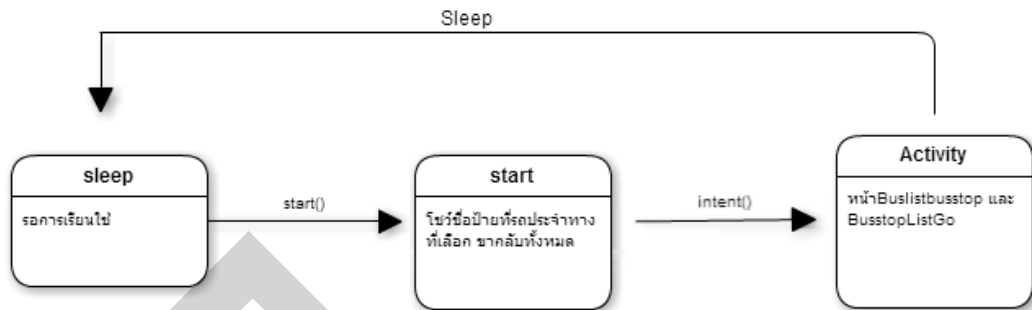
รูปที่ 3. 22 state diagram ของ class result Buslistbusstop (User)



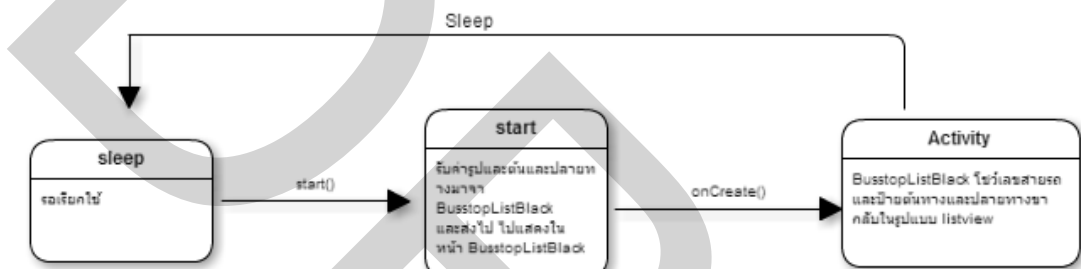
รูปที่ 3.23 state diagram ของ class result BusstopGo (User)



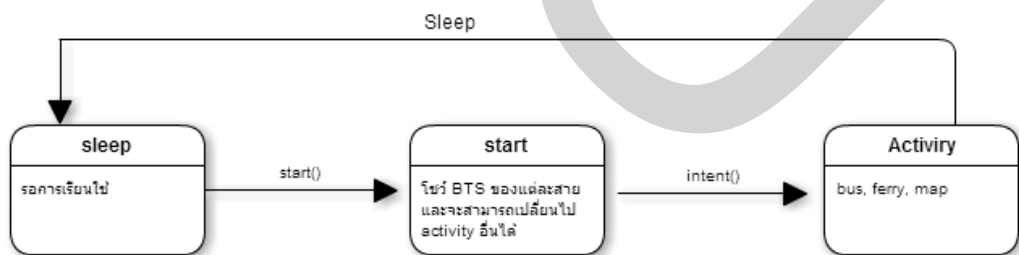
รูปที่ 3.24 state diagram ของ class result CustomListAdapterbusstop_list (User)



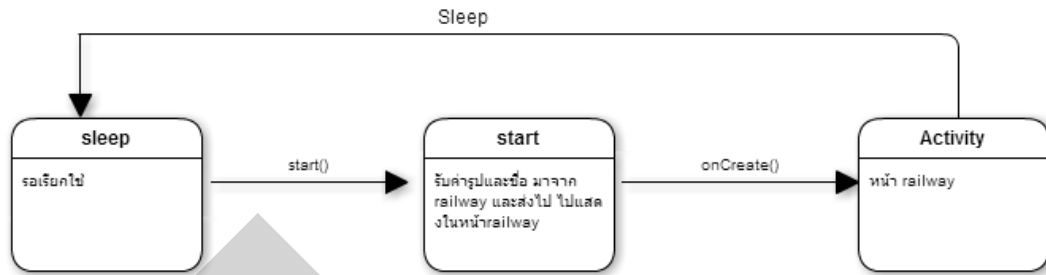
รูปที่ 3.25 state diagram ของ class BusstopListBlack (User)



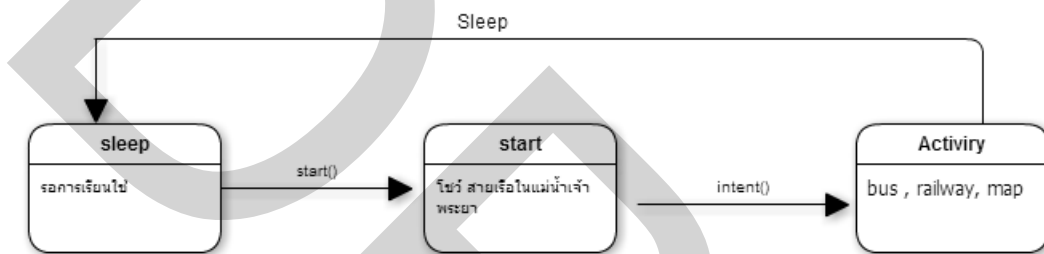
รูปที่ 3.26 state diagram ของ class CustomListAdapterbusstop_listB (User)



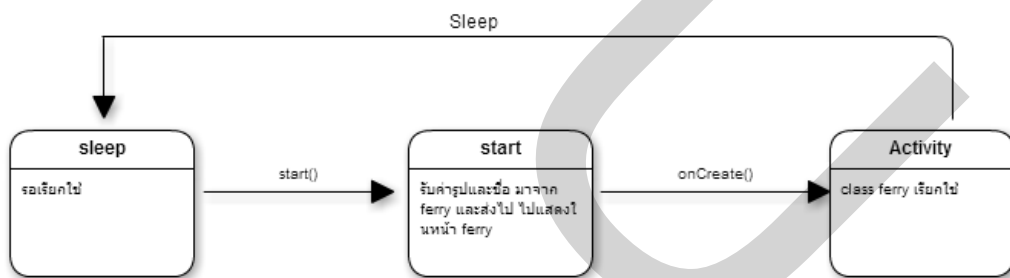
รูปที่ 3.27 state diagram ของ class railway (User)



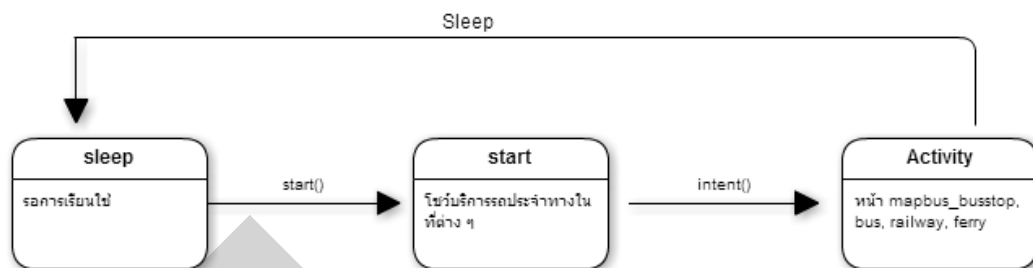
รูปที่ 3.28 state diagram ของ class CustomListAdapterrailway (User)



รูปที่ 3.29 state diagram ของ class ferry (User)



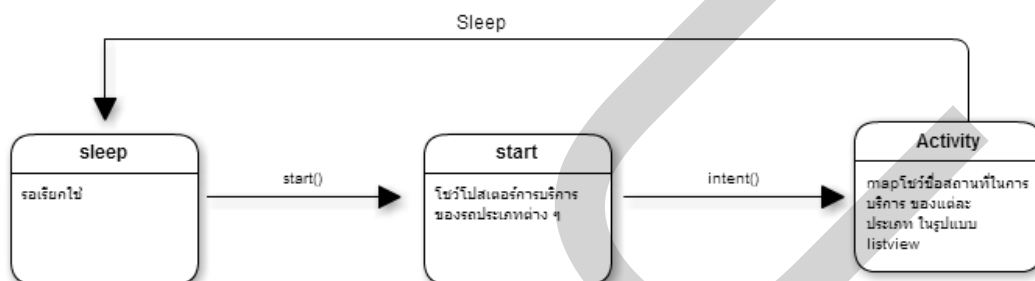
รูปที่ 3.30 state diagram ของ class CustomListAdapterferry (User)



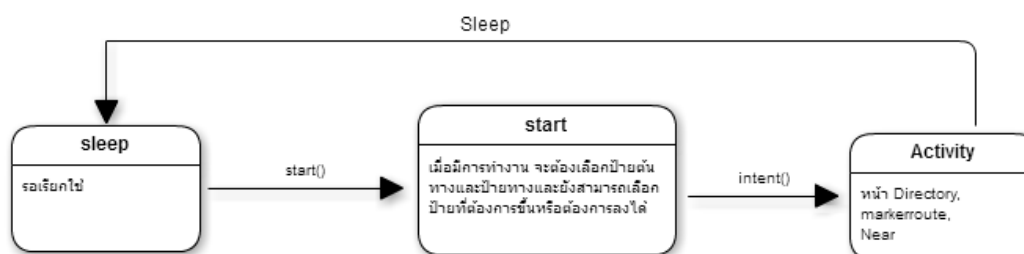
รูปที่ 3.31 state diagram ของ class map (User)



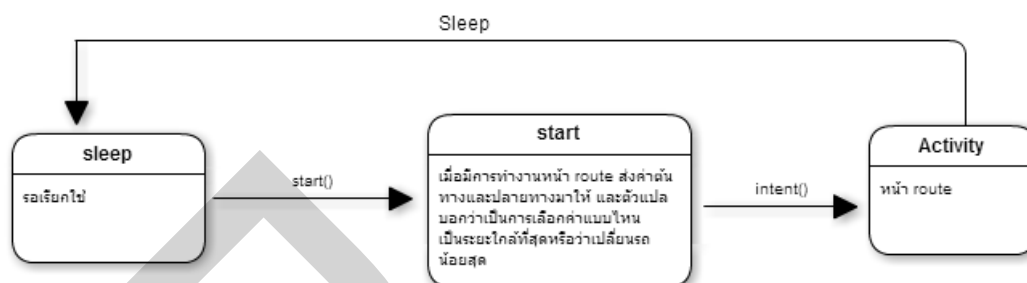
รูปที่ 3.32 state diagram ของ class CustomListAdaptermap (User)



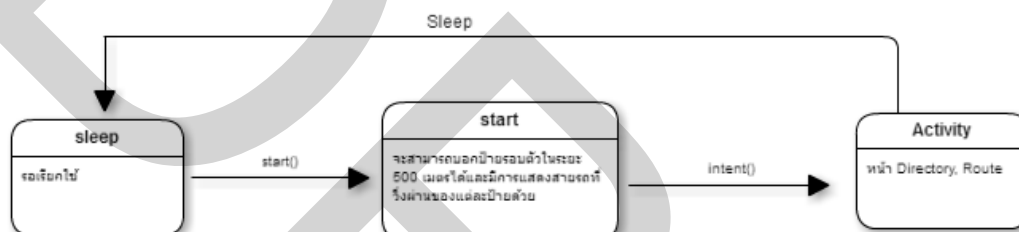
รูปที่ 3.33 state diagram ของ class mapbus_busstop (User)



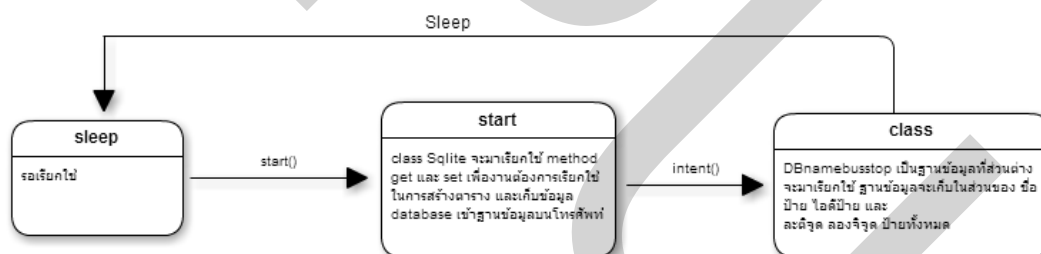
รูปที่ 3.34 state diagram ของ class route (User)



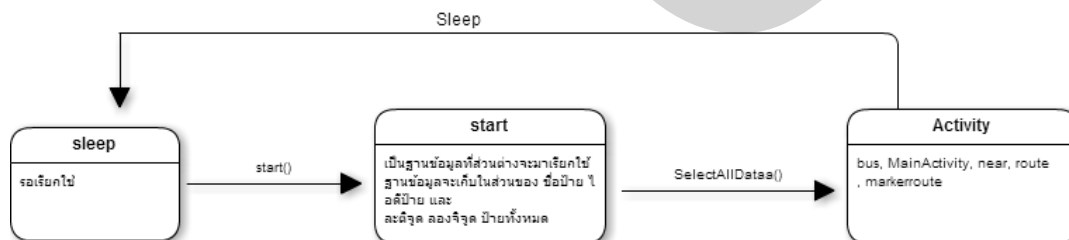
รูปที่ 3.35 state diagram ของ class markerroute (User)



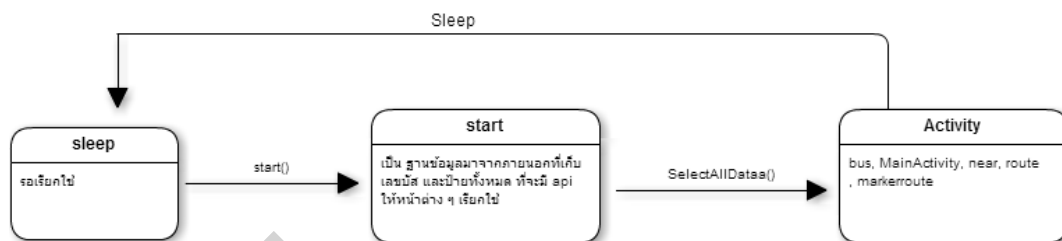
รูปที่ 3.36 state diagram ของ class near (User)



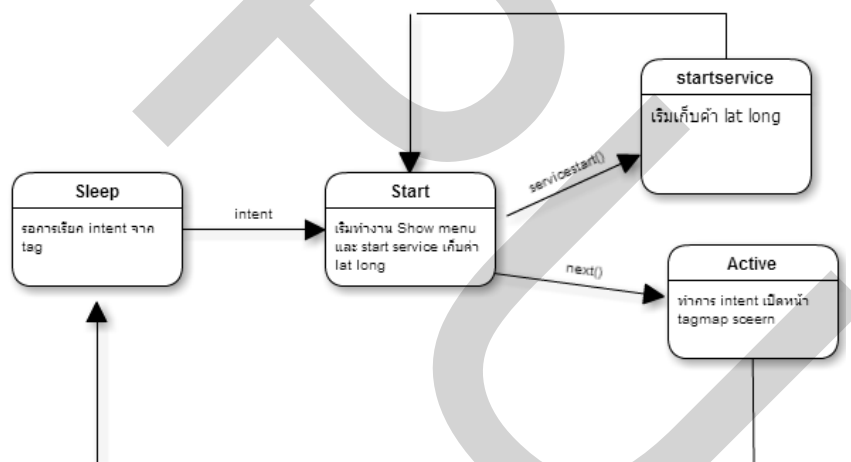
รูปที่ 3.37 state diagram ของ class savebusstop (User)



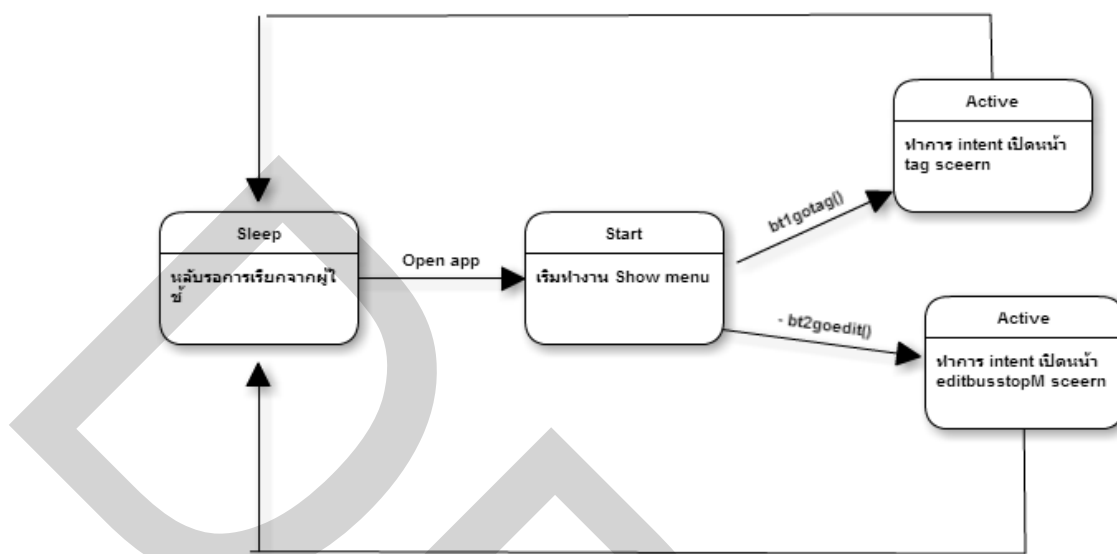
รูปที่ 3.38 state diagram ของ class DBnamebusstop (User)



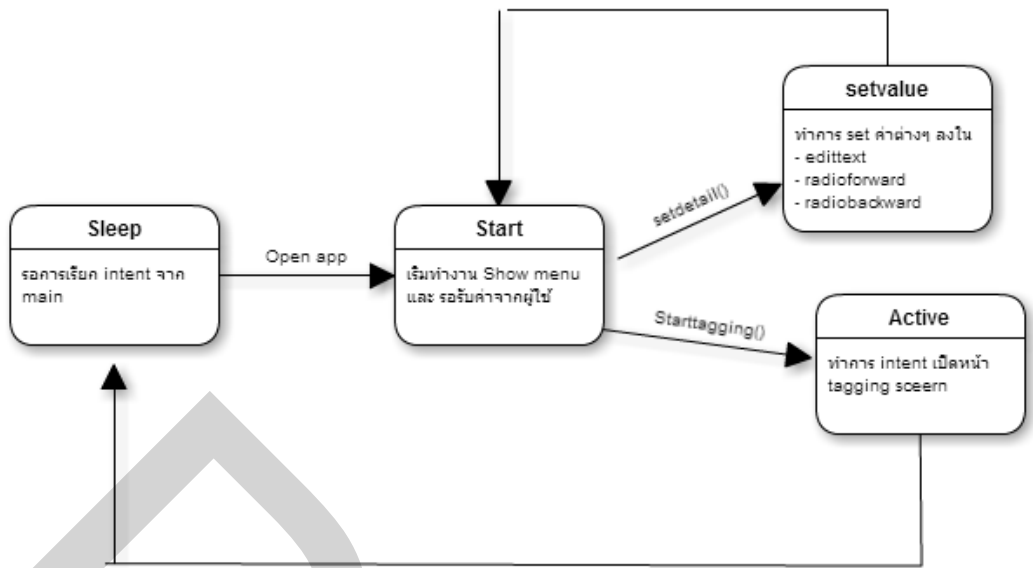
รูปที่ 3.39 state diagram ของ class Server (User)



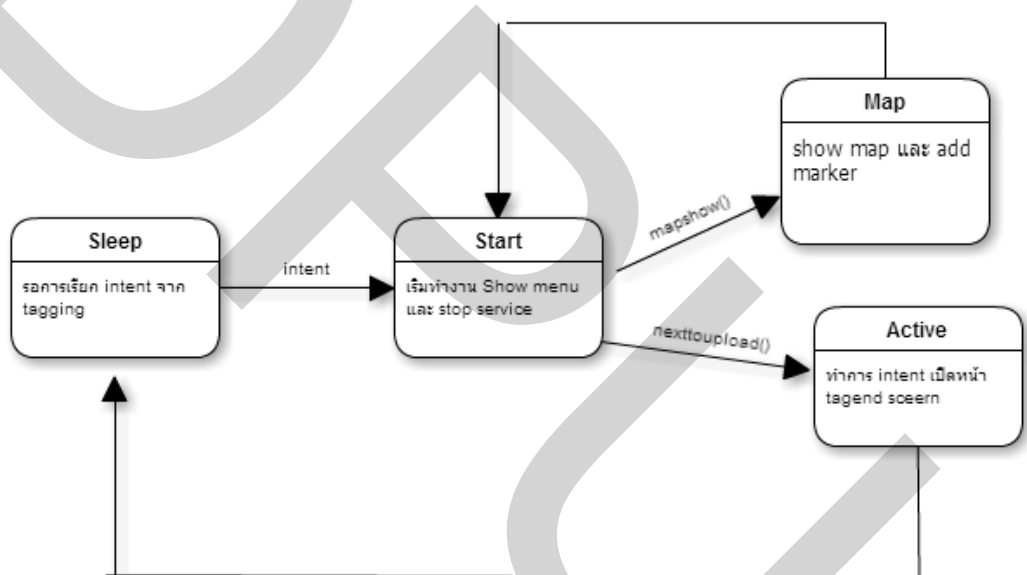
รูปที่ 3.40 state diagram ของ class main(Admin)



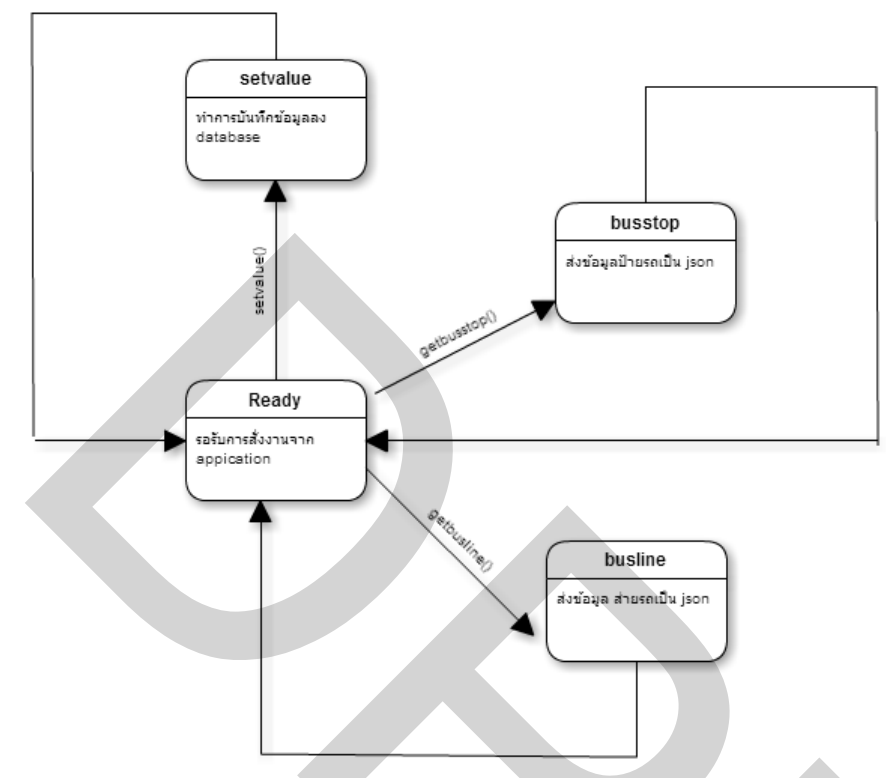
รูปที่ 3.41 state diagram ของ class tag (Admin)



รูปที่ 3.42 state diagram ของ class tagging (Admin)



รูปที่ 3.43 state diagram ของ class tagmap (Admin)



รูปที่ 3.48 state diagram ของ class database (Admin)

3.9 ขั้นตอนวิธีการคำนวณเส้นทางเดินรถ

หลังจากได้ข้อมูลการสำรวจเส้นทางรถประจำทางแต่ละสายมาแล้ว ซึ่งอยู่ในรูปแบบของลำดับของพิกัดบนถนน ซึ่งยังไม่สามารถนำมาใช้ในการคำนวณเส้นทางที่สั้นที่สุดในการโดยสารรถประจำทางได้ ระบบต้องนำมาแปลงเป็นลำดับของป้ายรถที่รถแต่ละสายวิ่งผ่าน โดยคำนวณว่ารถแต่ละสายวิ่งผ่านป้ายใดบ้าง ผู้วิจัยใช้วิธีดึงข้อมูลพิกัดรถสายหนึ่ง และพิกัดของป้ายทั้งหมดจากฐานข้อมูลแล้วทำการคำนวณระยะห่างระหว่างป้ายกับพิกัดของรถที่เก็บได้ ว่ามีพิกัดใดให้ค่าระยะห่างน้อยกว่า 500 เมตรหรือไม่ โดยพิจารณาเฉพาะพิกัดป้ายที่อยู่ทางด้านซ้ายของพิกัดเส้นทางเดินรถเท่านั้น จากนั้นทำการเรียงลำดับป้ายโดยหาระยะทางจากป้ายที่รถวิ่งผ่านกับจุดเริ่มต้นของเส้นทางเดินรถ แล้วเรียงป้ายตามระยะทางจากน้อยไปมาก ก็จะได้เส้นทางรถเดินรถผ่านป้ายรถ

อย่างไรก็ตาม เนื่องจากพิกัดที่ได้เป็นพิกัดบนพื้นราบเท่านั้น ทำให้การตัดสินใจในกรณีพิเศษต่าง ๆ ทำได้ยาก เช่น ในกรณีที่ป้ายรถอยู่ใต้ทางด่วนแต่รถประจำทางวิ่งบนทางด่วนนั้น โปรแกรมอาจ

เข้าใจว่าป้ายดังกล่าวอยู่บนเส้นทางเดินรถด้วย ดังนั้นเมื่อโปรแกรมทำการคำนวณเสร็จแล้ว จะต้องมีการตรวจสอบความถูกต้องอีกครั้งหนึ่งก่อนนำไปจัดเก็บลงในฐานข้อมูล

3.10 ขั้นตอนวิธีการคำนวณเส้นทางการเดินทางด้วยรถประจำทางที่สั้นที่สุด

ผู้วิจัยได้พัฒนาเอพีไอ (API) ที่สร้างโครงสร้างข้อมูลชนิดกราฟแบบมีทิศทางขึ้น โดยใช้เอนทิตี Busgraph ในฐานข้อมูลและเรียกใช้ขั้นตอนวิธีของ Dijkstra ผ่านไลบรารีจาก OrientDB PHP Library [22] โครงสร้างข้อมูลกราฟถูกสร้างขึ้นโดยมีป้ายรถเป็นโหนด และรถประจำทางที่วิ่งผ่านป้ายหนึ่งไปป้ายหนึ่งเป็นเส้นเชื่อมที่ใช้ระยะทางเป็นน้ำหนักของเส้นเชื่อม เนื่องจากป้ายรถของ ขสมก. มีจำนวนประมาณ 4,000 ป้าย เราจึงเลือกใช้ขั้นตอนวิธีของ Dijkstra ซึ่งให้การคำนวณที่มีความเร็วเพียงพอสำหรับปัญหาที่ ดังที่ได้อภิปรายในหัวข้อ 2.2 ไปแล้ว เอพีไอที่พัฒนาขึ้นได้เรียกใช้เมธอดที่เป็นขั้นตอนวิธีของ Dijkstra และส่งป้ายรถเริ่มต้นและป้ายรถปลายทางให้เป็นพารามิเตอร์ ซึ่งเป็นป้ายรถที่ผู้ใช้งานแอปพลิเคชันเลือก

นอกจากนี้ ผู้วิจัยยังได้รวมเอาเส้นทางเดินทางเข้าเป็นส่วนหนึ่งของเส้นทางคำตอบ ถ้าการเดินทางสามารถให้เส้นทางที่ดีกว่าเมื่อเทียบกับการโดยสารรถประจำทาง ตัวอย่างเช่น ในกรณีที่ปลายทางอยู่ฝั่งตรงข้ามข้าม แทนที่จะระบุให้ผู้โดยสารรถประจำทางไปจนสุดทางแล้วย้อนกลับมาฝั่งตรงข้าม แอปพลิเคชันที่พัฒนาจากเอพีไอนี้จะแนะนำให้เดินข้ามถนนแทน ซึ่งการทำแบบนี้จะต้องเพิ่มเส้นเชื่อมการเดินทางลงในฐานข้อมูล ผู้วิจัยได้เพิ่มเส้นเชื่อมการเดินทางให้กับทุกป้ายรถที่มีระยะห่างไม่เกิน 500 เมตร โดยมีน้ำหนักของเส้นเชื่อมเป็น 4 เท่าของระยะทางจริงเพื่อสะท้อนว่าการเดินช้ากว่าการโดยสารรถในกรุงเทพฯ 4 เท่า [23] จึงทำให้ขั้นตอนวิธีของ Dijkstra เลือกการโดยสารรถก่อนการเดินทางระหว่างป้ายคู่เดียวกัน

ส่วนการค้นหาเส้นทางที่เปลี่ยนรถน้อยที่สุด จะใช้แนวคิดเดียวกัน แต่จะให้ค่าน้ำหนักของเส้นเชื่อมแต่ละเส้นเป็น 1 และหากมีการเปลี่ยนรถจะเพิ่มน้ำหนักของเส้นเชื่อมขึ้นอีก 1 ทำให้ขั้นตอนวิธีของ Dijkstra เลือกเส้นทางที่เปลี่ยนรถน้อยกว่าเป็นคำตอบนั่นเอง

เมื่อขั้นตอนวิธีของ Dijkstra คืนลำดับของป้ายรถที่ให้เส้นทางที่สั้นที่สุดออกมา เอพีไอจะต้องนำมาสร้างเป็นลำดับของสายรถประจำทางที่ผู้ใช้งานต้องโดยสาร ดังนั้น เมื่อได้รับผลลัพธ์จากขั้นตอนวิธีของ Dijkstra แล้ว เอพีไอจะต้องนำมาสร้างเป็นลำดับของสายรถที่ผู้ใช้โดยสาร โดยมีการเปลี่ยนรถน้อยที่สุด นอกจากนี้ ถ้าระหว่างคู่ของป้ายใด ๆ มีรถประจำทางหลายสายผ่าน ต้องขอมอบให้ผู้เลือกใช้รถสายใดก็ได้ ขั้นตอนวิธีการเลือกสายรถของผู้วิจัยดำเนินการดังนี้

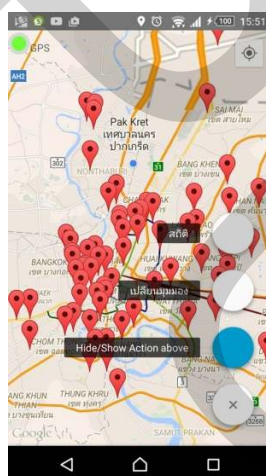
1. ที่ป้ายรถที่เป็นจุดเริ่มต้น บันทึกสายรถทุกสายที่ผ่านป้ายนี้ลงอะเรย์ชื่อ buses จากนั้น เลื่อนไป ยังป้ายถัดไปในคำตอบที่ได้จากขั้นตอนวิธีของ Dijkstra ซึ่งเป็นเส้นทางที่สั้นที่สุด
2. ที่ป้ายรถใด ๆ ที่อยู่ในคำตอบที่ได้จากขั้นตอนวิธีของ Dijkstra ให้เปรียบเทียบสายรถที่ผ่าน ป้ายนั้นกับอะเรย์ buses ถ้าสายรถในอะเรย์ buses ไม่ผ่านป้ายนั้น ให้ลบสายรถนั้นออก จากอะเรย์ buses ผลลัพธ์ที่ได้จากขั้นตอนนี้คือ สายรถในอะเรย์ buses เป็นสายรถที่ผู้ใช้งาน สามารถโดยสารระหว่างป้ายนี้กับป้ายก่อนหน้านั้นเอง
3. ถ้าไม่มีสายรถเหลืออยู่ในอะเรย์ buses หมายความว่าผู้ใช้งานอาจต้องเปลี่ยนสายรถ หรือ ต้อง เดินจากป้ายก่อนหน้ามายังป้ายปัจจุบัน ดังนั้นอะเรย์ buses จะเก็บสายรถทุกสายที่ผ่านป้ายก่อน หน้า ซึ่งอาจรวมถึงการเดินก็ได้
4. เลื่อนไปยังป้ายถัดไปในคำตอบของ Dijkstra แล้วทำข้อ 2 และข้อ 3 จนกว่าจะถึงป้ายปลายทาง
5. ผลลัพธ์คืออะเรย์ buses ซึ่งเป็นลำดับของสายรถที่ผู้ใช้งานสามารถโดยสารจากป้ายเริ่มต้น ไป ยังป้ายปลายทาง ผลลัพธ์อยู่ในรูปของ JSON ที่สามารถส่งต่อให้แอปพลิเคชันบนแอนดรอยด์ แสดงผลได้

บทที่ 4

ผลการดำเนินงาน

ในบทที่ 4 จะอธิบายถึงผลการดำเนินงานของโปรแกรมที่ได้พัฒนาขึ้นทั้งหมด ซึ่งประกอบด้วย โปรแกรมตรวจสอบป้ายรถบนแอนดรอยด์ โปรแกรมสำรวจเส้นทางรถประจำทางบนแอนดรอยด์ โปรแกรมกรอกข้อมูลเส้นทางการเดินทางรถประจำทางบนเว็บ และโปรแกรมค้นหาเส้นทางบนแอนดรอยด์ ผลการทดสอบประสิทธิภาพของระบบ และ ผลการสำรวจความคิดเห็นของผู้ใช้งานแอปพลิเคชันบนโทรศัพท์แอนดรอยด์เพื่อค้นหาเส้นทางที่สั้นที่สุด

4.1 โปรแกรมตรวจสอบป้ายรถประจำทาง



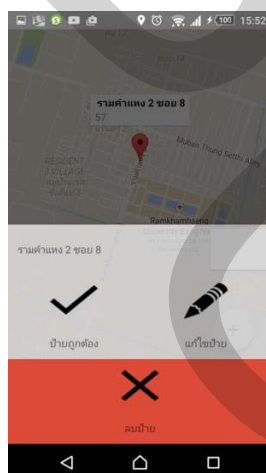
รูปที่ 4.1 หน้าจอ หมุดป้ายสายรถประจำทาง

รูปที่ 4.1 หน้าจอ ส่วนของการแสดงป้ายรถประจำทางทั้งหมดของ ขสมก. ที่ต้องมีการตรวจสอบความถูกต้อง จากการทดสอบโปรแกรมพบว่าการแสดงตำแหน่งปัจจุบันของผู้ใช้ใน รูปแบบของจุดสีฟ้าบนแผนที่ Google Map และจะแสดงความเสถียรของ GPS ได้



รูปที่ 4.2 หน้าจอ หมุดที่ทำการเช็คอิน

เมื่อผู้ใช้ขยายแผนที่ จะสามารถเห็นรายละเอียดในบริเวณต่าง ๆ ได้ โดยรูปที่ 4.2 แสดงการปักหมุดป้ายรถประจำทางบนแผนที่ที่ต้องการตรวจสอบความถูกต้อง



รูปที่ 4.3 หน้าจอตรวจสอบป้าย

เมื่อผู้ใช้กดปุ่มเมนู จะมีเมนูปรากฏดังแสดงในรูปที่ 4.3 ซึ่งแสดงหน้าจอเมนูการตรวจสอบข้อมูลป้ายรถประจำทาง ซึ่งมี 3 ทางเลือก ได้แก่ ป้ายถูกต้อง แก้ไขป้าย และ ลบป้าย



รูปที่ 4.4 หน้าจอ ยืนยันความถูกต้อง

เมื่อผู้ใช้เลือก แก้ไขป้าย จะแสดงฟอร์มดังแสดงในรูปที่ 4.4 เพื่อให้ผู้ใช้แก้ไขข้อมูลป้ายรถประจำทาง ผู้ใช้สามารถกรอกข้อมูลและกดปุ่ม CONFIRM เพื่อบันทึกไว้ในฐานข้อมูล

4.2 โปรแกรมสำรวจเส้นทางเดินรถประจำทาง



รูปที่ 4.5 หน้าจอ เช็การเปิดระบบการบอกตำแหน่ง (GPS)

เมื่อผู้ใช้เปิดการใช้งานโปรแกรม จะมีการตรวจสอบการเปิดใช้งานระบบ GPS ของโทรศัพท์ หากผู้ใช้ยังไม่ได้เปิดใช้งานระบบ GPS ของโทรศัพท์ โปรแกรมจะแสดงหน้าจอดังรูป 4.5 ซึ่งจะมีการ

เดือนขึ้นมาในครั้งแรกเมื่อยังไม่มีระบบการบอกตำแหน่งเท่านั้น แต่ถ้าหากมีการเปิดระบบ GPS แล้วจะไม่มีแจ้งเตือนนี้



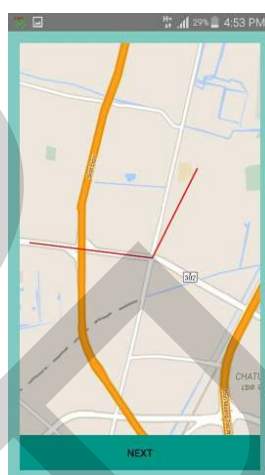
รูปที่ 4.6 หน้าจอ การกรอกข้อมูลและ ประเภทสายรถประจำทาง

รูปที่ 4.6 แสดงหน้าจอการกรอกข้อมูลสายรถประจำทาง จะมีการกรอก ข้อมูลเลขสายรถประจำทาง เลขข้างรถสายรถประจำทาง ระบุขาไปหรือขากลับ หรือรถประจำทางอาจมีการวิ่งเพียงครึ่งสาย โดยจะไม่วิ่งไปจนสุดสาย จะสามารถกรอกตามที่รถสายนั้นจะวิ่งได้ เพื่อเก็บเป็นข้อมูลของแต่ละสายรถประจำทาง และสามารถเลือกประเภทของรถประจำทางว่าเป็นรถประเภทอะไร



รูปที่ 4.7 หน้าจอ การเปิดการทำงานเบื้องหลัง

จากนั้น โปรแกรมจะทำงานอยู่เบื้องหลัง (Service) โดยจะแสดงหน้าจอ ดังรูปที่ 4.7 ซึ่งผู้ใช้สามารถปิดหน้าจอที่ใช้ในการเก็บละติจูดและลองจิจูดของรถที่วิ่งในแต่ละสายได้ โดยผู้ใช้สามารถวางโทรศัพท์ไว้ในรถ และโปรแกรมจะทำงานเบื้องหลังตลอดเวลาจนกว่าจะสุดสายและมีการกดปุ่มหยุด (END) เพื่อระบุว่าถึงปลายทางแล้ว



รูปที่ 4.8 หน้าจอ วาดเส้นทางสายรถประจำทาง

เมื่อผู้ใช้กดปุ่มหยุด(END) โปรแกรมจะแสดงเส้นทางการเดินทางรถที่ได้บนแผนที่ดังแสดงในรูปที่ 4.8 ซึ่งแสดงหน้าจอการวาดเส้นลงบนแผนที่ Google Map ตาม ละติจูดและลองจิจูดที่เก็บมาจากการทำงานเบื้องหลังที่เก็บละติจูดและลองจิจูดทั้งหมด มาวาดลงบนเส้นเพื่อดูว่าสายรถประจำทางสายที่เก็บมาวิ่งไปบนทางเส้นไหนเพื่อให้ผู้ใช้ได้ตรวจสอบความถูกต้องของข้อมูลที่สำรวจได้ก่อนเก็บลงฐานข้อมูล

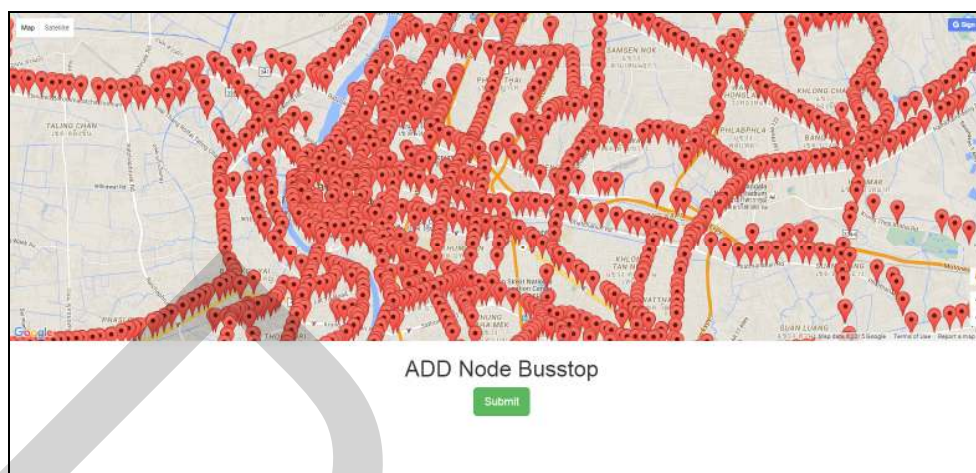


รูปที่ 4.9 หน้าจอ บอกรายละเอียดและส่งข้อมูลขึ้นฐานข้อมูล

จากนั้น ผู้ใช้สามารถกดปุ่ม upload เพื่อให้โปรแกรมเก็บข้อมูลลงฐานข้อมูลในเครื่องแม่ข่าย ดังแสดงในรูปที่ 4.9 ซึ่งแสดงข้อมูลของรถที่ทำการกรอกไว้แล้วก่อนหน้านี้ คือ สายรถประจำทาง เลข ข้างสายรถประจำทาง ข้อมูลสายที่วิ่ง ประเภทสายที่วิ่ง และยังสามารถใส่ ข้อมูลที่เก็บ เช่น เวลา วัน เดือน ปี ที่เก็บข้อมูลสายรถประจำทางนั้น

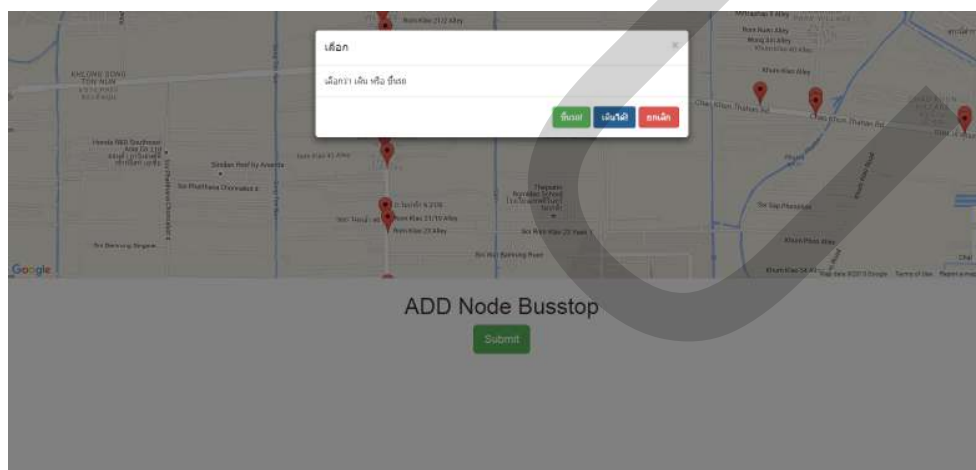
4.3 โปรแกรมกรอกข้อมูลเส้นทางเดินรถประจำทางบนเว็บ

เนื่องจากการเก็บข้อมูลเส้นทางรถประจำทางด้วยแอนดรอยด์ มีข้อจำกัดหลายประการ ได้แก่ ต้องใช้โทรศัพท์ที่ให้ค่าพิกัดที่แม่นยำ การใช้งานจำนวนมากในการสำรวจเส้นทางรถแต่ละสาย และใช้เวลาบนรถนาน ทำให้ผู้วิจัยได้พัฒนาโปรแกรมเพื่อกรอกข้อมูลเส้นทางเดินรถบนเว็บทดแทน เพื่อให้สามารถเก็บข้อมูลการเดินทางโดยไม่ต้องเดินทางไปยังสถานที่จริง และสามารถกรอกข้อมูลเมื่อไรก็ได้ ไม่จำเป็นต้องอยู่บนรถเท่านั้น



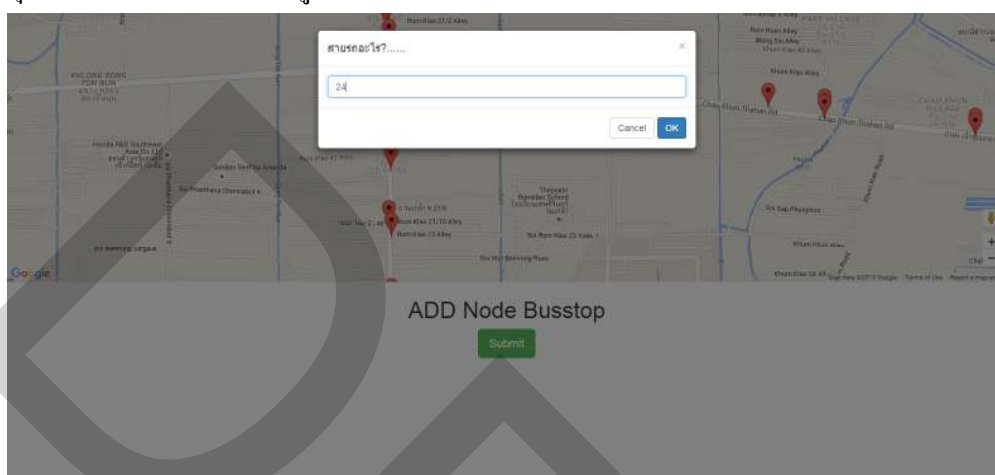
รูปที่ 4.10 หน้าจอ ป้ายสายรถประจำทางใน กรุงเทพมหานคร และปริมณฑล

รูปที่ 4.10 แสดงหน้าเว็บซึ่งแสดงป้ายรถทั้งหมดคในระบบในกรุงเทพมหานคร และปริมณฑล บนแผนที่ Google Map เพื่อให้ผู้ใช้ระบุเส้นทางการเดินทางของสายรถประจำทางแต่ละสาย ที่มีการวิ่ง ผ่านในแต่ละป้ายในกรุงเทพมหานคร และปริมณฑล



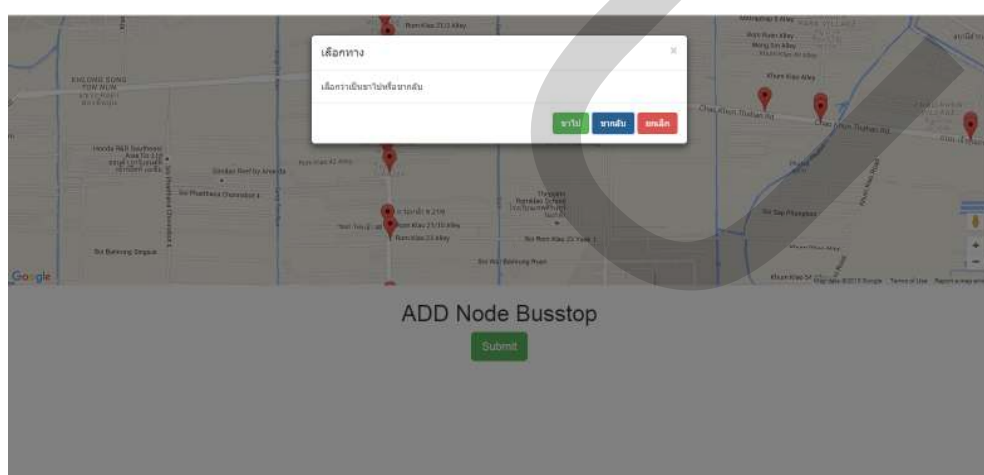
รูปที่ 4.11 หน้าจอ เลือกประเภทในการเดินทางไปยังป้าย

เมื่อผู้ใช้คลิกเลือกป้าย 2 ป้ายที่รถประจำทางวิ่งผ่าน โปรแกรมจะแสดงหน้าจอการเลือกประเภทในการเดินทางไปยังป้ายที่เลือก ดังแสดงในรูปที่ 4.11 โดยผู้ใช้จะต้องมีการเลือกหมวดต้นทางและหมวดปลายทางที่ไปที่ทิศทางที่ถูกต้องเพื่อที่จะสามารถใช้ในการค้นหาเส้นทางได้



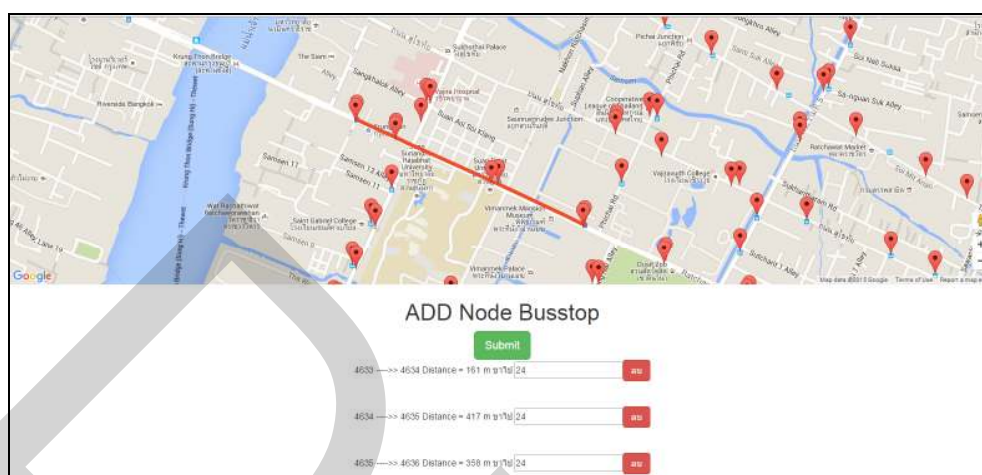
รูปที่ 4.12 หน้าจอ ระบุสายรถประจำทาง

จากนั้น ผู้ใช้ต้องระบุสายรถประจำทางที่จะวิ่งผ่านป้ายที่เลือกลงไป ดังรูปที่ 4.12 เพื่อที่จะนำไปทำแผนที่เส้นทางสายรถประจำทางของแต่ละสายที่ใช้ระบุไปใช้ในการค้นหาเส้นทางได้



รูปที่ 4.13 หน้าจอ เลือกประเภทการเดินทางของสายรถประจำทาง

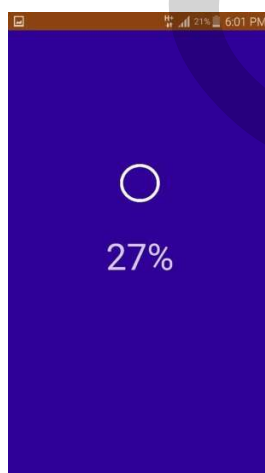
จากนั้นผู้ใช้เลือกกว่าเส้นทางที่ระบุเป็นขาไปหรือขากลับดังแสดงในรูปที่ 4.13



รูปที่ 4.14 หน้าจอ ข้อมูลเส้นทางสายรถประจำทาง

ข้อมูลเส้นทางที่ผู้ใช้กรอกจะแสดงเป็นเส้นบนแผนที่และแสดงข้อมูลได้แผนที่ ผู้ใช้สามารถตรวจสอบความถูกต้องของข้อมูลก่อนส่งไปเก็บในฐานข้อมูลในเครื่องแม่ข่าย โดยต้องมีการลากเส้นทางไปในทางทิศเดียวกันและต้องต่อเนื่องกัน จึงสามารถนำไปใช้ในการค้นหาเส้นทางได้

4.4 โปรแกรมค้นหาเส้นทางที่สั้นที่สุดบนแอนดรอยด์



รูปที่ 4.15 หน้าจอดาวน์โหลดป้ายรถประจำทาง

หลังจากที่ได้ข้อมูลเส้นทางรถโดยสารเรียบร้อยแล้ว ผู้ใช้ทั่วไปสามารถเรียกใช้งาน โปรแกรม ค้นหาเส้นทางที่สั้นที่สุดที่ทำงานบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ โดยเมื่อเปิดโปรแกรมขึ้นมา ในครั้งแรก โปรแกรมจะดาวน์โหลดข้อมูลป้ายรถประจำทางทั้งหมดที่มีอยู่ในฐานข้อมูล เพื่อนำไปใช้ในกรณีที่ไม่ได้มีการเชื่อมต่ออินเทอร์เน็ต สำหรับฟังก์ชันค้นหาป้ายรถรอบพิกัดผู้ใช้



รูปที่ 4.16 หน้าจอแสดงสายรถประจำทางทั้งหมด

รูปที่ 4.16 แสดงหน้าจอที่แสดงหมายเลขสายรถประจำทางทั้งหมดที่มีอยู่ในกรุงเทพมหานคร และปริมณฑล แต่ละลำดับจะมีการแสดงชื่อป้ายต้นทางและป้ายปลายทาง และจะสามารถทำการค้นหาเลขสายรถประจำทางที่ต้องการได้โดยพิมพ์ชื่อสายในช่องค้นหาค้านบน และผู้ใช้สามารถดูป้ายที่รถประจำทางแต่ละสายวิ่งผ่านทั้งขาไปและขากลับได้อีกด้วย

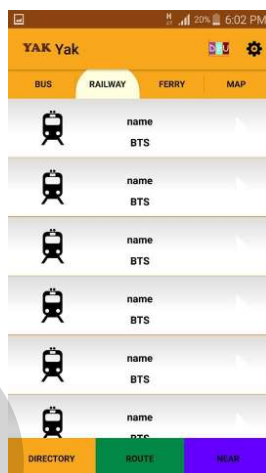


รูปที่ 4.17 หน้าจอ แสดงป้ายของสายรถประจำทาง (ขาไป)



รูปที่ 4.18 หน้าจอ แสดงป้ายของสายรถประจำทาง (ขากลับ)

รูปที่ 4.17 แสดงหน้าจอที่แสดงเส้นทางการเดินรถของสายรถประจำทางที่ผู้ใช้เลือก โดยจะแสดงการเดินทางของแต่ละสายรถโดยมีการระบุว่ามีการผ่านป้ายใดบ้างตามลำดับ ส่วนรูปที่ 4.18 แสดงเส้นทางการเดินรถขากลับ เมื่อผู้ใช้เลือกแท็บขากลับด้านบน



รูปที่ 4.19 หน้าจอ แสดงสายรถไฟฟ้า

รูปที่ 4.19 แสดงหน้าจอลิสต์รายการการเดินทางของ BTS, MRT, AIRPORT LINK ในรูปแบบโปสเตอร์ โดยจะมีการแสดงในทุกสาย ของรถไฟฟ้าเพื่อเป็นข้อมูลในการเดินทาง



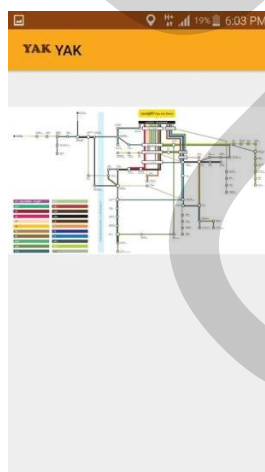
รูปที่ 4.20 หน้าจอ แสดงการเดินทางเรือ

รูปที่ 4.20 แสดงหน้าจอที่แสดงลิสต์รายการข้อมูลการเดินทางเรือแต่ละสายในรูปแบบโปสเตอร์ เพื่อให้ข้อมูลการเดินทางโดยเรือในแม่น้ำเจ้าพระยา



รูปที่ 4.21 หน้าจอ บอกแสดงบริการสายรถประจำทาง

รูปที่ 4.21 แสดงหน้าจอที่แสดงแผนที่การเดินทางสำหรับสถานที่สำคัญต่าง ๆ ที่จัดทำโดยบริษัท YAK/SmartVC โดยในปัจจุบันมีแผนที่รถโดยสารชนิดต่าง ๆ รอบอนุสาวรีย์ชัยสมรภูมิ



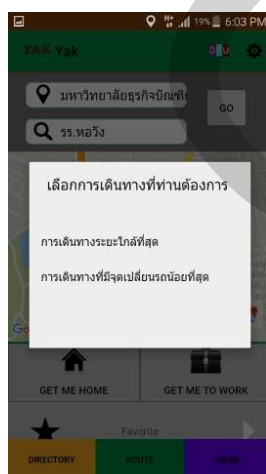
รูปที่ 4.22 หน้าจอ บอกข้อมูลสายรถประจำทาง

เมื่อคลิกที่ชื่อแผนที่ในรูปที่ 4.21 จะแสดงแผนที่ที่แสดงจุดให้บริการของสายรถโดยสารชนิดต่าง ๆ ในรูปแบบโปสเตอร์ เช่น รถคู่ รถประจำทาง รถประจำทางมินิบัส และจะแสดงรถประจำทางแบบบริการการเดินทางออกต่างจังหวัดด้วย



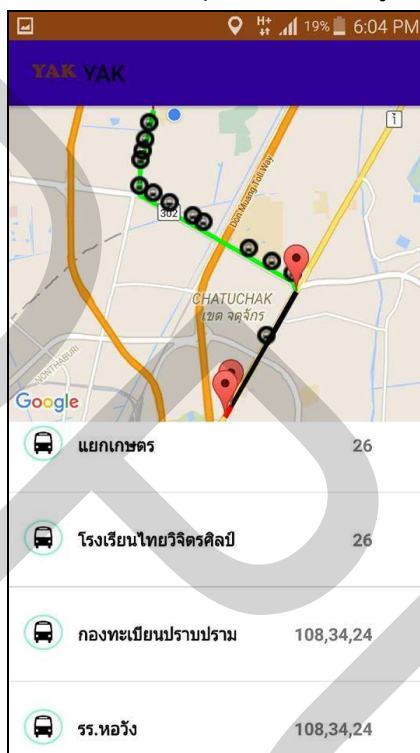
รูปที่ 4.23 หน้าจอ ค้นหาเส้นทางรถประจำทาง

ในส่วนการค้นหาเส้นทางรถประจำทางที่สั้นที่สุด แสดงในรูปที่ 4.23 โดยจะแสดงป้ายจุดเริ่มต้นเป็นละติจูดและลองจิจูดปัจจุบัน และยังสามารถค้นหาต้นทางที่ต้องการได้โดยพิมพ์ค้นหาสถานที่ต้นทางและยังสามารถเลือกป้ายที่ต้องขึ้นรถประจำทางได้ ส่วนป้ายปลายทางเราสามารถค้นหาสถานที่ที่ต้องการไปได้และจะสามารถเลือกป้ายที่ต้องการลงรถได้ โดยขอบเขตการค้นหาป้ายจะอยู่ในเขตกรุงเทพฯ และปริมณฑล เท่านั้น



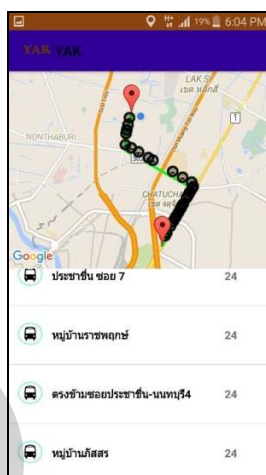
รูปที่ 4.24 หน้าจอ เลือกการเดินทางด้วยสายรถประจำทาง

เมื่อเลือกป้ายต้นทางและปลายทางแล้ว โปรแกรมจะแสดงเมนูให้เลือกการเดินทางเพื่อให้เหมาะกับสถานการณ์ในการเดินทางของผู้ใช้ ดังแสดงในรูปที่ 4.24 โดยจะมีการเดินทางให้เลือกสองแบบ คือ การเดินทางระยะทางที่สั้นที่สุด ซึ่งอาจต้องมีการเปลี่ยนรถเพื่อลดระยะทางการเดินทาง ส่วนแบบที่สองเป็นการเดินทางที่มีจุดเปลี่ยนรถน้อยที่สุด ซึ่งเหมาะกับผู้เดินทางที่ไม่ต้องการขึ้นลงรถบ่อย



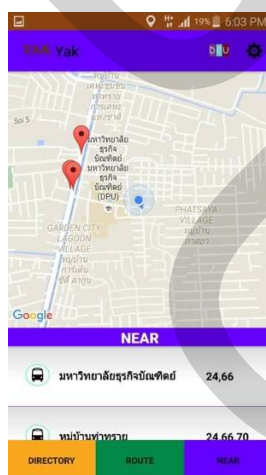
รูปที่ 4.25 หน้าจอ แสดงเส้นทางการเดินทางที่สั้นที่สุด

หากผู้ใช้เลือกการเดินทางด้วยเส้นทางที่สั้นที่สุด โปรแกรมจะแสดงหน้าจอ ดังรูปที่ 4.25 โดยจะแสดงป้ายที่ต้องผ่านจากต้นทางไปยังปลายทางและแสดงจุดที่ต้องเปลี่ยนสายรถประจำทางเพื่อต่อรถประจำทางไปยังปลายทาง นอกจากนี้ยังแสดงการเดิน เพื่อขึ้นรถประจำทาง ลงบนแผนที่ โปรแกรมจะมีการแสดงเส้นทางเป็นสีต่าง ๆ เพื่อแสดงการเปลี่ยนสายรถประจำทางหรือการเดิน และจะมีการแสดงในรูปแบบลิสต์รายการเป็นลำดับการเดินทางที่ต้องผ่านในแต่ละป้าย จากป้ายแรกไปยังป้ายสุดท้าย



รูปที่ 4.26 หน้าจอ แสดงเส้นทางการเดินทางมีจุดเปลี่ยนรถน้อยที่สุด

รูปที่ 4.26 แสดงหน้าจอที่แสดงเส้นทางการเดินทางที่มีจุดเปลี่ยนรถน้อยที่สุด ซึ่งมีการแสดงผลเหมือนรูปที่ 4.25 แต่ใช้วิธีการคำนวณต่างกัน



รูปที่ 4.27 หน้าจอ แสดงป้ายรอบตัว

นอกจากนี้ ผู้ใช้ยังสามารถตรวจสอบป้ายรถประจำทางที่อยู่ในบริเวณใกล้เคียงกับตำแหน่งปัจจุบันของผู้ใช้ โดยจะแสดงป้ายรอบตัวในระยะ 500 เมตรพร้อมชื่อป้ายและสายรถประจำทางที่รถประจำทางวิ่งผ่านทั้งหมดที่ผ่านป้ายนั้น ๆ บนแผนที่ และจะแสดงลิสต์รายการชื่อป้ายในระยะทั้งหมดด้านล่าง

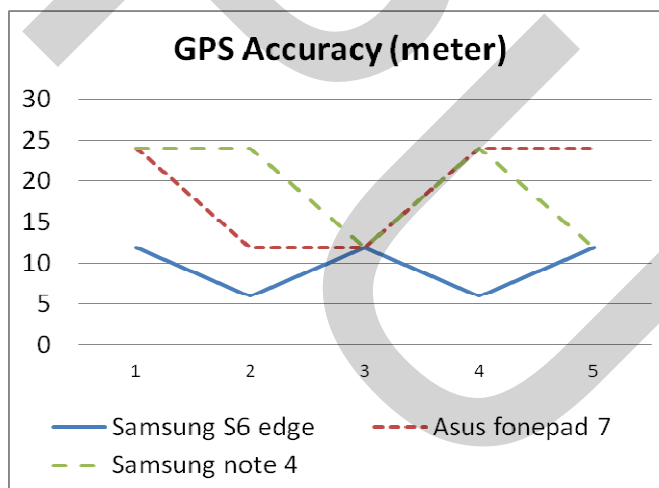
4.5 ผลการทดสอบประสิทธิภาพของระบบ

4.5.1 ผลการทดสอบความแม่นยำของ GPS ของโทรศัพท์แอนดรอยด์

ผู้วิจัยได้ทำการทดสอบความแม่นยำ (Accuracy) ของค่าพิกัดที่ได้จาก GPS ของโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ที่ต่างกันจำนวน 3 รุ่น ได้แก่

1. Samsung Galaxy S6 Edge เป็นตัวแทนของโทรศัพท์รุ่นที่ดีที่สุดในปัจจุบัน,
2. Samsung Note 4 เป็นตัวแทนของโทรศัพท์รุ่นกลาง
3. Asus Fonepad 7 เป็นตัวแทนของโทรศัพท์ราคาประหยัด

ผู้วิจัยได้ทำการทดสอบโดยการยื่นที่ป้ายรถบริเวณหน้ามหาวิทยาลัย และบันทึกผลที่ได้จากโปรแกรม GPS Test [9] บนโทรศัพท์ทั้ง 3 เครื่อง เครื่องละ 5 ครั้ง โดยไม่มีการขยับร่างกาย ผลค่าความแม่นยำของ GPS ของโทรศัพท์แต่ละรุ่นเป็นดังแสดงในรูปที่ 4.28 โดยแกน x แสดงหมายเลขครั้งที่บันทึกข้อมูล และแกน y แสดงค่าความแม่นยำมีหน่วยเป็นเมตร ซึ่งค่าความแม่นยำที่น้อยกว่าจะหมายถึงค่าที่วัดได้แต่ละครั้งจะใกล้เคียงค่าที่ถูกต้องมากกว่า



รูปที่ 4.28 ค่าความแม่นยำ (Accuracy) ที่วัดได้จากโทรศัพท์แอนดรอยด์ทั้ง 3 รุ่น

จากผลการทดสอบพบว่าโทรศัพท์รุ่น Samsung Galaxy S6 edge ให้ค่าความแม่นยำอยู่ระหว่าง 6-12 เมตรซึ่งเป็นค่าที่ดีกว่า(ค่าต่ำกว่า) ส่วนโทรศัพท์อีกสองรุ่นกลับให้ค่าความแม่นยำใกล้เคียงกันอยู่ระหว่าง 12-24 เมตร ดังนั้น รุ่นของโทรศัพท์ซึ่งใช้อุปกรณ์ GPS ที่ต่างกัน มีผลต่อความแม่นยำของพิกัดที่สำรวจได้ ดังนั้นการใช้โทรศัพท์รุ่นที่ดีที่สุดจะช่วยลดความคลาดเคลื่อนของพิกัดได้

อย่างไรก็ตามการเก็บข้อมูลพิกัดป้ายรถประจำทางไม่จำเป็นต้องใช้อุปกรณ์ GPS ที่ให้ความแม่นยำในระดับเซนติเมตร เนื่องจาก ในการใช้งานจริง ขนาดของป้ายรถเมล์รวมที่พิกผู้โดยสารมีขนาดใหญ่กว่า 6 เมตร และหากป้ายรถประจำทางอยู่ในระยะที่ผู้ใช้งานสามารถมองเห็นได้ (ประมาณ 6 เมตร เช่นกัน) เนื่องจากสายตาของมนุษย์ในระดับ 20/20 สามารถมองเห็นอย่างชัดเจนในระยะประมาณ 20 ฟุต หรือ 6 เมตร) ดังนั้น GPS ที่มีความแม่นยำ 6-12 เมตรก็เพียงพอต่อการใช้งาน

4.5.2 ผลการทดสอบโปรแกรมสร้างเส้นทางเดินรถภาษาจาวา

จากการทดสอบโปรแกรมการแปลงเส้นทางรถเป็นลำดับของป้ายที่รถวิ่งผ่าน ที่เขียนขึ้นด้วยภาษาจาวา ซึ่งทำงานบนเครื่องคอมพิวเตอร์ MSI GT70 ที่มี CPU Intel Core i7 2.40 GHz หน่วยความจำ 8GB โดยจับเวลาที่ใช้ในการคำนวณโดยใช้คำสั่ง System.nanoTime() ในภาษาจาวาจับเวลาการคำนวณ ได้ผลการทดสอบโปรแกรมดังแสดงในตารางที่ 4.1 โดยผู้วิจัยได้ทำการทดสอบโปรแกรมกับสายรถจำนวน 4 สาย แต่ละสายมีพิกัดเส้นทางรถมากกว่า 5,000 พิกัด

ตารางที่ 4.1 ประสิทธิภาพทางเวลาของการสร้างเส้นทางรถประจำทางผ่านป้ายรถ

สายรถ	จำนวนพิกัดบนเส้นทาง	เวลาที่ใช้ (ms)
510	10,722	60,622
24	5,572	32,816
70	5,712	32,775
66	5,596	31,961

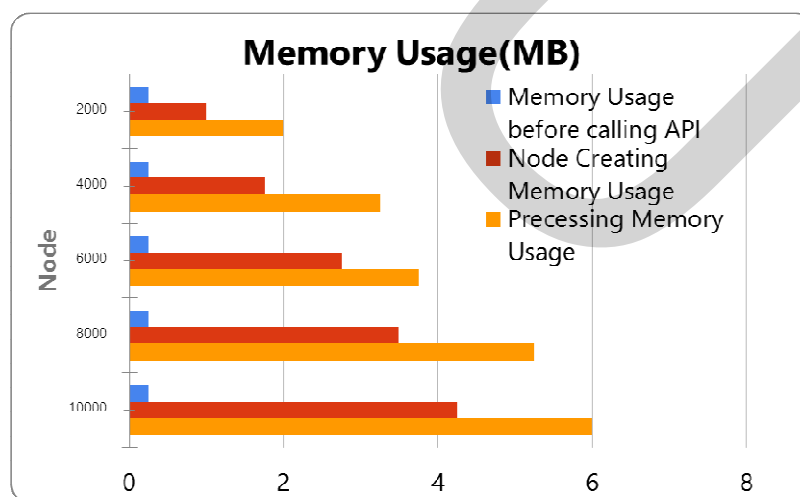
ตารางที่ 1 แสดงประสิทธิภาพทางเวลาของโปรแกรมเมื่อทำการสร้างเส้นทางรถประจำทางสายต่าง ๆ ผ่านป้ายรถ ซึ่งมีจำนวนทั้งสิ้น 4,781 ป้าย พบว่าแต่ละสายรถมีจำนวนพิกัดบนเส้นทางที่เก็บข้อมูลมาไม่เท่ากัน ทำให้การคำนวณในเส้นทางของแต่ละสายใช้เวลาไม่เท่ากัน ขึ้นอยู่กับจำนวนพิกัดของเส้นทางเดินรถที่เก็บมาได้ โดยโปรแกรมใช้เวลาโดยเฉลี่ยที่ 5.75 ms ในการคำนวณหนึ่งพิกัดบนเส้นทางเดินรถ

อย่างไรก็ตาม การสร้างเส้นทางรถประจำทางด้วยวิธีดังกล่าว ยังไม่สามารถให้เส้นทางที่ถูกต้อง 100 เปอร์เซ็นต์ได้ เนื่องจากในบางกรณี ถึงแม้ว่าป้ายจะอยู่ห่างจากเส้นทางไม่มาก แต่อาจเป็นป้ายที่อยู่ได้ทางด่วนที่รถวิ่งผ่าน ทำให้การสร้างเส้นทางผิดพลาดได้ จึงต้องมีการตรวจสอบข้อมูลโดยผู้เชี่ยวชาญอีกครั้งหนึ่งก่อนนำไปใช้งานจริง

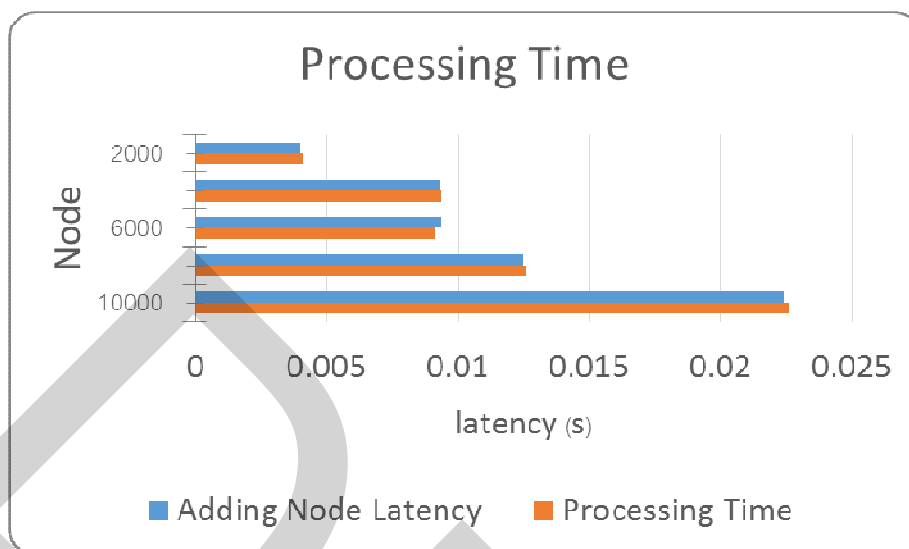
ถึงแม้ว่าการคำนวณนี้จะใช้เวลาค่อนข้างนาน แต่ก็เป็นการทำงานแบบ pre-processing เพื่อเตรียมข้อมูลสำหรับการพัฒนาโปรแกรมเพื่อให้ข้อมูลการเดินทางโดยรถประจำทางต่อไป ดังนั้นเมื่อนำข้อมูลไปใช้ในโปรแกรมการค้นหาเส้นทางการเดินทางต่อไปในอนาคต ผู้ใช้งานทั่วไปจะไม่ต้องเสียเวลารอการคำนวณนี้

4.5.3 ผลการทดสอบประสิทธิภาพของแอปพลิเคชันค้นหาเส้นทางรถประจำทางที่สั้นที่สุด

เพื่อวัดประสิทธิภาพของเอพีไอค้นหาเส้นทางรถประจำทางที่สั้นที่สุดที่ได้สร้างขึ้น ผู้วิจัยได้ทำการทดลองวัดเวลาในการประมวลผลและการใช้พื้นที่หน่วยความจำบนเครื่องแม่ข่าย โดยทดลองให้เอพีไอสร้างกราฟขนาด 2,000 ถึง 10,000 โหนดแบบสุ่มพิกลัด แล้วสุ่มค้นหาเส้นทางที่สั้นที่สุด ดังผลการทดลองในรูปที่ 4.29 และรูปที่ 4.30 จากผลการทดลองพบว่าเอพีไอที่สร้างขึ้นใช้หน่วยความจำน้อยกว่า 6 MB และใช้เวลาในการประมวลผลน้อยกว่า 0.025 วินาที เมื่อประมวลผลโครงสร้างข้อมูลกราฟที่ประกอบด้วยโหนดจำนวน 10,000 โหนด



รูปที่ 4.29 การใช้งานพื้นที่หน่วยความจำของเอพีไอ



รูปที่ 4.30 เวลาที่เอพีไอใช้ในการประมวลผล

นอกจากนี้ ผู้วิจัยยังวัดเวลาตอบกลับของแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ที่เรียกใช้ไอพีโอเน็ต ดังแสดงในตารางที่ 4.2 โดยวัดเวลาตอบกลับของแอปพลิเคชันในช่วงเวลาต่างๆของวัน เนื่องจากปริมาณการใช้เวลาอินเทอร์เน็ตในช่วงเวลาต่าง ๆ ของวันมีผลต่อเวลาตอบกลับค่อนข้างมาก จากผลการทดลองพบว่าเวลาตอบกลับอยู่ในช่วงที่ผู้ใช้ยอมรับได้ คือส่วนใหญ่ไม่น้อยกว่า 4 วินาที [11]

ตารางที่ 4.2 เวลาตอบกลับของแอนดรอยด์แอปพลิเคชัน

จำนวนป้ายรถในคำตอบ	เวลาตอบกลับ(วินาที)		
	9.30-11.30am	3-5pm	8.30-11pm
10	0.912	1.220	3.125
15	0.945	1.267	3.569
20	1.123	1.458	3.498
30	1.145	1.461	4.652
40	1.230	1.784	3.681

4.6 ผลการสำรวจความคิดเห็นต่อการใช้งานของแอปพลิเคชันค้นหาเส้นทางรถประจำทางบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์

จากการสำรวจความคิดเห็นต่อการใช้งานแอปพลิเคชันค้นหาเส้นทางบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ ระหว่างวันที่ 18 พฤศจิกายน ถึง 11 ธันวาคม 2558 โดยทำการสำรวจประชาชนที่อาศัยอยู่ในเขตการเดินรถประจำทางทั้ง 8 เขต เขตละ 8 คน โดยแบ่งเป็นผู้หญิง 32 คน ชาย 32 คน และแบ่งตามช่วงอายุ 3 ช่วง ได้แก่ น้อยกว่า 20 ปีจำนวน 29 คน, 20-55 ปีจำนวน 34 คน, 55 ปีขึ้นไปจำนวน 1 คน เป็นผู้ใช้งานรถประจำทางทุกวัน 22 คน ใช้งานสัปดาห์ละ 1-2 ครั้ง 25 คน และใช้งานนาน ๆ ครั้ง 17 คน

ผู้วิจัยได้ออกเดินทางไปยังเขตการเดินรถทั้ง 8 เขต และให้ผู้ใช้งานโปรแกรมที่ติดตั้งบนโทรศัพท์ที่เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์รุ่น Samsung Note 4 โดยตรง จากนั้นให้ผู้ใช้งานตอบแบบประเมินที่สามารถให้ระดับความพึงพอใจได้ 5 ระดับ คือ เห็นด้วยมากที่สุด(5 คะแนน), เห็นด้วยมาก(4 คะแนน), เห็นด้วยปานกลาง(3 คะแนน), เห็นด้วยน้อย(2 คะแนน) และ เห็นด้วยน้อยที่สุด(1 คะแนน) ผู้ใช้งานมีความคิดเห็นเกี่ยวกับโปรแกรมที่ได้พัฒนาขึ้น ดังในตารางที่ 4.3

ตารางที่ 4.3 ระดับความคิดเห็นต่อโปรแกรมบนโทรศัพท์ระบบปฏิบัติการแอนดรอยด์ที่ได้พัฒนาขึ้น

คำถาม	ระดับความคิดเห็นเฉลี่ย	ค่าเบี่ยงเบนมาตรฐาน
คุณคิดว่า icon สอดคล้องกับลักษณะกับการใช้งานใช่หรือไม่	4.13	0.65
คุณคิดว่าสีพื้นหลัง สีตัวอักษร และปุ่มใช้งานเหมาะสมกันใช่หรือไม่	4.05	0.71
คุณคิดว่าหน้าจอการใช้งานของแอปพลิเคชันเข้าใจง่ายหรือไม่	4.19	0.61
แอปพลิเคชันทำให้การใช้งานรถประจำทางง่ายขึ้นใช่หรือไม่	4.52	0.61
แอปพลิเคชันสามารถนำไปใช้งานได้จริงใช่หรือไม่	4.47	0.66
คุณคิดว่าฟังก์ชันในแอปพลิเคชันครบถ้วนใช่หรือไม่	3.81	0.79
คุณคิดว่าข้อมูล ในแอปพลิเคชันถูกต้องใช่หรือไม่	3.92	0.78
คุณคิดว่า แอปพลิเคชันนี้มีประโยชน์ต่อประชาชนในกรุงเทพฯอยู่ในระดับใด	4.44	0.58
คุณคิดว่าประสิทธิภาพโดยรวมของแอปพลิเคชันอยู่ในระดับใด	4.02	0.62

คำถาม	ระดับความคิดเห็นเฉลี่ย	ค่าเบี่ยงเบนมาตรฐาน
คุณคิดว่าความง่ายในการใช้งานแอปพลิเคชันอยู่ในระดับใด	4.03	0.74
เฉลี่ยรวม	4.16	0.72

นอกจากนี้ผู้ใช้งานยังให้ข้อเสนอแนะอื่น ๆ ในการปรับปรุงโปรแกรมต่อไป ดังนี้

- ควรปรับหน้าจอการใช้งานให้มีสีสันสวยงาม และจัดกลุ่มข้อมูล เพื่อให้ user ใช้งานได้ง่ายขึ้น (2 คน)
- อยากให้ใส่รูปรถเมล์แต่ละสาย พร้อมระบุว่าเป็นเอกชนหรือรัฐบาล เพื่อให้คนที่ใช้แอปพลิเคชันหรือชาวต่างชาติสังเกตเห็นได้
- ควรมีสภาพการจราจรและเวลาในการเดินทางโดยประมาณ
- ควรเพิ่มเวลาในการเดินรถ คันแรกคันสุดท้ายหรือวิ่งทั้งคืน (3 คน)
- ต้องมีภาษาทางเลือก
- ความง่ายต่อการใช้งานไม่ต้องเปิด GPS จะดีกว่านี้
- เป็นประโยชน์ต่อกลุ่มที่ไม่ได้ใช้รถเป็นประจำมากกว่า อาจมีปัญหาเรื่องสัญญาณอินเทอร์เน็ต และ GPS
- ควรให้มีสัญลักษณ์กำหนดสถานที่จุดสำคัญ ๆ ไว้ด้วย เช่น วัด โรงพยาบาล สถานที่ราชการ ฯลฯ

บทที่ 5

สรุป

ข้อมูลของระบบขนส่งมวลชนของ ขสมก. ไม่มีการปรับปรุงข้อมูลให้เป็นปัจจุบันเป็นเวลานาน และไม่ได้มีการรวบรวมข้อมูลอย่างเป็นระบบ จึงทำให้การพัฒนาการให้ข้อมูลการเดินทางในระบบของ ขสมก. ทำได้ยากลำบาก งานวิจัยนี้ได้กล่าวถึงการพัฒนาการเก็บข้อมูลป้ายรถประจำทาง และสร้างข้อมูลเส้นทางรถโดยสาร โดยใช้โทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์เป็นเครื่องมือในการเก็บข้อมูลพิกัดป้ายส่งไปยังเครื่องแม่ข่ายฐานข้อมูล จากนั้น โปรแกรมในระบบจะทำการแปลงพิกัดป้ายกับเส้นทางที่เก็บรวบรวมได้ให้กลายเป็นเส้นทางรถประจำทางแต่ละสายโดยอัตโนมัติ ผลการทดสอบระบบพบว่าระบบสามารถเก็บข้อมูลป้ายรถและเส้นทางรถได้ และ โปรแกรมการสร้างเส้นทางรถผ่านป้ายรถใช้เวลาคำนวณเฉลี่ย 5.75 ms ต่อหนึ่งพิกัดเส้นทางรถ ข้อมูลจากระบบดังกล่าวจะถูกนำไปใช้ในการสร้างระบบเพื่อให้ข้อมูลการเดินทางด้วยรถประจำทางแก่คนทั่วไปต่อไป

นอกจากนี้ งานวิจัยยังได้พัฒนาแอปพลิเคชันสำหรับการค้นหาเส้นทางที่สั้นที่สุดในการเดินทางด้วยรถประจำทางของ ขสมก. โดยใช้ข้อมูลที่เก็บได้จากระบบข้างต้น และได้พัฒนาแอปพลิเคชันบนโทรศัพท์เคลื่อนที่ระบบปฏิบัติการแอนดรอยด์สำหรับการค้นหาเส้นทางที่สั้นที่สุดโดยใช้ขั้นตอนวิธีของ Dijkstra บนโครงสร้างข้อมูลแบบกราฟแบบมีทิศทางโดยให้ป้ายรถเป็นโหนด และเส้นเชื่อมระหว่างโหนดคือสายรถหรือการเดินทาง โดยระบบจะสร้างเส้นเชื่อมแบบการเดินทางให้อัตโนมัติ ถ้าป้ายรถห่างกันไม่เกิน 500 เมตร และให้ค่าน้ำหนักการเดินทางเป็นสี่เท่าของการนั่งรถ

จากผลการทดสอบประสิทธิภาพของระบบพบว่าแอปพลิเคชันที่ใช้พื้นที่หน่วยความจำน้อยกว่า 6 MB และใช้เวลาน้อยกว่า 0.025 วินาทีในการประมวลผลเส้นทางที่อยู่บนกราฟที่มี 10,000 โหนด ส่วนแอปพลิเคชันบนแอนดรอยด์สามารถทำงานได้ภายในเวลา 4-5 วินาทีซึ่งเป็นเวลาที่ผู้ใช้งานยอมรับได้ ผลการสำรวจความคิดเห็นของผู้ใช้งานจำนวน 64 คน จาก 8 เขตการเดินทาง พบว่าผู้ใช้งานมีความพึงพอใจมากในระดับ 4.16 โดยเห็นว่าแอปพลิเคชันที่พัฒนาขึ้นทำให้การเดินทางด้วยรถประจำทางสะดวกขึ้น สามารถใช้งานได้จริง และเป็นประโยชน์ในระดับมากที่สุด

บรรณานุกรม

- [1] Thai Publica (17 กุมภาพันธ์ 2558). สํารวจค่าใช้จ่ายค่าเดินทางคนกรุงเทพฯ รถไฟ รถเมล์ เรือ วินมอเตอร์ไซด์ ตุ๊กตุ๊ก และแท็กซี่, [ออนไลน์]. เข้าถึงได้จาก: <http://thaipublica.org/2015/02/bangkok-public-transportation/> (วันที่ค้นข้อมูล : 5 มิถุนายน 2558)
- [2] สํานักการจราจรและขนส่ง, กรุงเทพมหานคร. สถิติจราจร ปี 2556, [ออนไลน์]. <http://office.bangkok.go.th/dotat/StatBook/b2556.pdf> (วันที่ค้นข้อมูล : 5 มิถุนายน 2558)
- [3] กองนโยบายและแผนงาน, สํานักผังเมืองกรุงเทพมหานคร. รายงานการศึกษาเรื่อง:ข้อมูลผู้ใช้บริการระบบขนส่งมวลชนกรุงเทพ, [ออนไลน์]. เข้าถึงได้จาก : http://cpd.bangkok.go.th:90/web2/strategy/DATA54_55/2MASST.pdf (วันที่ค้นข้อมูล : 5 มิถุนายน 2558)
- [4] การรถไฟฟ้าขนส่งมวลชนแห่งประเทศไทย. รายงานประจำปี 2556, [ออนไลน์]. เข้าถึงได้จาก : <http://www.mrta.co.th/th/aboutMRTA/annualReport/All2556.pdf> (วันที่ค้นข้อมูล : 5 มิถุนายน 2558)
- [5] IDC.com, “Smartphone OS Market Share Q1 2015”, [ออนไลน์]. เข้าถึงได้จาก : <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> (วันที่ค้นข้อมูล : 5 มิถุนายน 2558)
- [6] SiamTraffic.NET [ออนไลน์]. เข้าถึงได้จาก : <http://www.siamtraffic.net> (วันที่ค้นข้อมูล : 5 มิถุนายน 2558).
- [7] องค์การขนส่งมวลชนกรุงเทพ. สายรถประจำทาง [ออนไลน์]. เข้าถึงได้จาก : <http://www.bmta.co.th/?q=th/bus-lines>. (วันที่ค้นข้อมูล : 5 มิถุนายน 2558).
- [8] Google Maps. [ออนไลน์]. เข้าถึงได้จาก : <http://maps.google.co.th/>. (วันที่ค้นข้อมูล : 5 มิถุนายน 2558).
- [9] Google Play. [ออนไลน์]. เข้าถึงได้จาก : <http://play.google.com/>. (วันที่ค้นข้อมูล : 5 มิถุนายน 2558).
- [10] Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford (2001). "Section 24.3: Dijkstra's algorithm". Introduction to Algorithms (Second ed.). MIT Press and McGraw–Hill. pp. 595–601. ISBN 0-262-03293-7
- [11] Kissmetrics, “How loading time affects your bottom line”. <http://goo.gl/kosva>
- [12] H. Bast, et. al. Route Planning in Transportation Networks. Technical Report

- [13] I. Abraham, et. al. A Hub-Based Labeling Algorithm for ShortestPaths on Road Networks. In P. M. Pardalos and S. Rebennack, editors, Experimental Algorithms (SEA), volume 6630 of LNCS, pages 230–241. Springer, 2011.
- [14] R. Geisberger, et. al. Exact Routing in Large Road Networks Using Contraction Hierarchies. *Transportation Science*, 46(3):388–404, 2012.
- [15] A. V. Goldberg, et. al. Reach for A*: Shortest Path Algorithms with Preprocessing. In C. Demetrescu, A. V. Goldberg, and D. S. Johnson, editors, The Shortest Path Problem: Ninth DIMACS Implementation Challenge, volume 74 of DIMACS Book, pages 93–139. AMS, 2009.
- [16] M. Hilger, et. al. Fast Point-to-Point Shortest Path Computations with Arc-Flags. In C. Demetrescu, A. V. Goldberg, and D. S. Johnson, editors, The Shortest Path Problem: Ninth DIMACS Implementation Challenge, volume 74 of DIMACS Book, pages 41–72. AMS, 2009.
- [17] H. Bast, et. al. Fast Routing in Road Networks with Transit Nodes. *Science*, 316(5824):566, 2007.
- [18] E. Pyrga, et. al. Efficient Models for Timetable Information in Public Transportation Systems. *ACM Journal of Experimental Algorithmics*, 2007.
- [19] Delling, D., et. al. Parallel computation of best connections in public transportation networks. *ACM J. Exp. Algor.* 17, 4, Article 4.4 (October 2012), 26 pages.
- [20] Pyrga, E., et. al. 2007. Efficient models for timetable information in public transportation systems. *ACM J. Exp. Algor.* 12, Article 2.4 (2007).
- [21] D. Delling, et. al. Round-Based Public Transit Routing. In *Algorithm Engineering and Experiments (ALENEX)*, pages 130–140. SIAM, 2012.
- [22] OrientDB[online]. Available: <https://github.com/doctrine/orientdb-odm>. (last accessed : June 5, 2015).
- [23] เดลินิวส์ (2555)[ออนไลน์]. เข้าถึงได้จาก: <http://www.dailynews.co.th/Content/regional/140023>. (วันที่ค้นข้อมูล: 1 มีนาคม 2556).