

การออกแบบ และการพัฒนาเว็บเซอร์วิสสำหรับ
ระบบการตรวจสอบประสิทธิภาพการขาย

เศรษฐพงศ์ กตคุณไพศาล

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเว็บ วิทยาลัยครีเอทีฟดีไซน์

แอนด์ เอ็นเตอร์เทนเมนต์เทคโนโลยี

มหาวิทยาลัยธุรกิจบัณฑิตย์

พ.ศ. 2563

**A Design and Development of Web Services
for Sales Performance Monitoring**

Setthapong Katakunpaisarn

**Thematic Paper Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Web Engineering,
College of Creative Design and Entertainment Technology,
Dhurakij Pundit University**

2020

หัวข้อสารนิพนธ์	การออกแบบ และการพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย
ชื่อผู้เขียน	เศรษฐพงศ์ กตคุณ ไพศาล
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ ดร. วรสิทธิ์ ชูชัยวัฒนา
สาขาวิชา	สาขาวิชาวิศวกรรมเว็บ
ปีการศึกษา	2562

บทคัดย่อ

การจะประสบความสำเร็จในการดำเนินธุรกิจในปัจจุบัน สิ่งหนึ่งที่เป็นปัจจัยหลักคือ ข้อมูลที่ถูกต้องน่าเชื่อถือ และต้องมาในเวลาที่เหมาะสม เพื่อให้การตัดสินใจถูกต้องมากที่สุด ผู้ประกอบการ SMEs จำนวนไม่น้อยที่ยังคงตัดสินใจโดยปราศจากข้อมูลรองรับ สาเหตุเนื่องจากขาดองค์ความรู้ในการบริหารจัดการข้อมูลในองค์กร

สารนิพนธ์นี้จึงนำเสนอการวิเคราะห์ และออกแบบการพัฒนาเว็บเซอร์วิสสำหรับระบบตรวจสอบประสิทธิภาพการขาย เพื่อตอบสนองความต้องการของผู้ใช้งาน และสนับสนุนนักพัฒนาในฝั่งไคลเอนท์ให้สามารถเชื่อมต่อรับส่งข้อมูลเว็บเซอร์วิสได้อย่างสะดวก ซึ่งในการพัฒนาได้นำแนวคิด รวมถึงเทคโนโลยีการสื่อสารผ่านเครือข่ายที่เกี่ยวข้อง เช่น REST API และ รูปแบบข้อมูล JSON รวมถึงการนำ Open API Specification มาประยุกต์ใช้เพื่อให้เกิดประโยชน์ในการพัฒนาระบบซอฟต์แวร์เพื่อสนับสนุนการดำเนินธุรกิจได้อย่างมีประสิทธิภาพ

Thematic Paper Title	A Design and Development of Web Services for Sales Performance Monitoring
Author	Setthapong Katakunpaisarn
Individual Study Advisor	Asst. Prof. Dr. Worasit Choochaiwattana
Department	Web Engineering
Academic Year	2019

Abstract

To be successful in business today, one of the key factors is accurate, reliable, and timely information. In order to make the most accurate decisions, many SMEs still have to make decisions without supporting information. The main reason is a lack of knowledge in data management in the organization.

This thematic paper aims at analyzing, designing, and developing web services for sales performance monitoring systems. To meet the needs of users and to support developers on the client side, the web services are designed and developed for easy connecting by adopting the concept and communication technology such as REST API and JSON data formats. In addition, the Open API Specification is used to apply for the benefit of software system development to support business operations efficiently.

กิตติกรรมประกาศ

การจัดทำสารนิพนธ์เรื่องการออกแบบ และการพัฒนาเว็บไซต์สำหรับระบบการตรวจสอบประสิทธิภาพการขาย ครั้งนี้สำเร็จลุล่วงได้ด้วยดี โดยได้รับความช่วยเหลือ และการสนับสนุนจากผู้มีความรู้หลาย ๆ ท่าน โดยเฉพาะอย่างยิ่ง ผู้ช่วยศาสตราจารย์ ดร.วรสิทธิ์ ชูชัยวัฒนา ซึ่งเป็นอาจารย์ปรึกษา ที่ได้เสียสละเวลาให้คำแนะนำ ให้ความรู้ด้านวิชาการ และเทคนิคต่าง ๆ ตลอดจนข้อคิดเห็นที่เป็นประโยชน์ในการแก้ไขปรับปรุงผลงานสารนิพนธ์ฉบับนี้ ให้สมบูรณ์มากยิ่งขึ้น ผู้จัดทำซาบซึ่งในความกรุณาของท่านเป็นอย่างยิ่ง และขอกราบขอบพระคุณท่านเป็นอย่างสูง ขอขอบคุณเพื่อน ๆ ร่วมรุ่นทุก ๆ คน รวมถึงเพื่อน ๆ ที่ทำงาน ที่คอยให้ความช่วยเหลือ ซึ่งกัน และกันมาตลอดระยะเวลาการศึกษา

ขอกราบขอบพระคุณคุณแม่แม่ครอบครัว และบุคคลอันเป็นที่รักที่เป็นกำลังใจอันสำคัญ ยิ่งในการจัดทำสารนิพนธ์จนประสบความสำเร็จลุล่วงด้วยดี ซึ่งทุกท่านจะถูกจารึกไว้ในจิตใจ ของผู้จัดทำสารนิพนธ์ตลอดไป

ในท้ายที่สุดนี้ผู้จัดทำหวังว่าผลงานสารนิพนธ์ฉบับนี้จะเป็นประโยชน์กับผู้ที่ต้องการศึกษา ด้านการพัฒนาระบบจัดการสารสนเทศ และหากมีข้อผิดพลาดประการใดในงานสารนิพนธ์ฉบับนี้ ผู้จัดทำต้องขอกราบขออภัย เป็นอย่างสูงมา ณ ที่นี้ด้วย

เศรษฐพงศ์ กตคุณไพศาล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	๗
บทคัดย่อภาษาอังกฤษ.....	๘
กิตติกรรมประกาศ.....	๑
สารบัญตาราง.....	๗
สารบัญภาพ.....	๘
บทที่	
1. บทนำ.....	1
1.1 ความสำคัญ และที่มาของปัญหา.....	1
1.2 วัตถุประสงค์ของการศึกษา.....	2
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.4 ขอบเขตการศึกษา.....	2
2. ทฤษฎี และงานวิจัยที่เกี่ยวข้อง.....	4
2.1 สถาปัตยกรรมซอฟต์แวร์เชิงบริการ SOA (Service-Oriented Architecture).....	4
2.2 เว็บเซอร์วิส (Web service).....	7
2.3 Open API Specification (OAS).....	10
2.4 JavaScript Object Notation (JSON).....	21
2.5 NEST JS.....	23
2.6 Business Intelligence.....	23
3. ขั้นตอนการดำเนินงาน และเครื่องมือ.....	25
3.1 ศึกษาปัญหา และความต้องการของระบบ.....	25
3.2 วิเคราะห์ และออกแบบระบบ.....	27
3.3 เครื่องมือการพัฒนาระบบ.....	54
3.4 ระยะเวลาการดำเนินการ.....	55
4. ผลการดำเนินงาน.....	56
4.1 วิธีการทดสอบการพัฒนาระบบ.....	56
4.2 ผลการพัฒนาระบบ.....	59

สารบัญ (ต่อ)

บทที่	หน้า
5. สรุปอภิปรายผลการศึกษา และข้อเสนอแนะ.....	98
5.1 สรุป และอภิปรายผล.....	98
5.2 ปัญหา และอุปสรรค.....	99
5.3 ข้อเสนอแนะ.....	99
บรรณานุกรม.....	100
ภาคผนวก.....	102
ก. พจนานุกรมข้อมูล (Data Dictionary).....	103
ประวัติผู้เขียน.....	119

สารบัญตาราง

ตารางที่	หน้า
3.1 สรุป Use Case ของระบบการตรวจสอบประสิทธิภาพการขาย.....	29
3.2 Use Case ผู้ใช้เข้าสู่ระบบ.....	29
3.3 Use Case ผู้ใช้ตรวจสอบ Performance Overall.....	30
3.4 Use Case ผู้ใช้ตรวจสอบ Customer Overall.....	31
3.5 Use Case ผู้ใช้ตรวจสอบ Customer Focus.....	32
3.6 แสดงรายละเอียดตารางข้อมูลระบบการตรวจสอบประสิทธิภาพการขาย.....	34
3.7 Use Case (การออกแบบ API).....	36
3.8 อุปกรณ์ฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบ.....	54
3.9 ซอฟต์แวร์ที่ใช้ในการพัฒนาระบบ.....	55
3.10 ระยะเวลาการดำเนินการ.....	55
4.1 Test Case ที่นำมาทดสอบการทำงานของเมนู Performance Overall.....	57
4.2 Test Case ที่นำมาทดสอบการทำงานของเมนู Customer Overall.....	58
4.3 Test Case ที่นำมาทดสอบการทำงานของเมนู Customer Focus.....	58

สารบัญภาพ

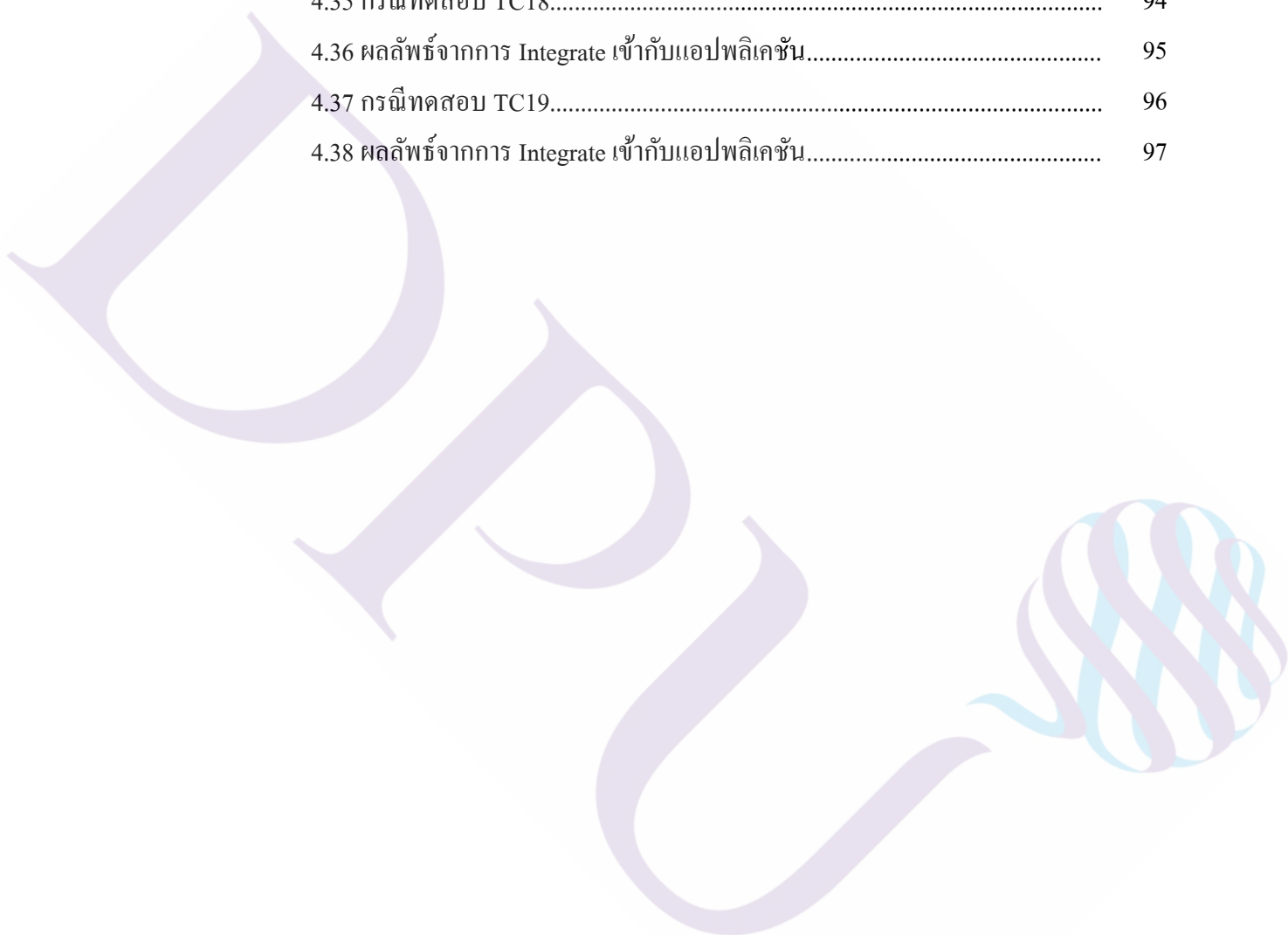
ภาพที่	หน้า
2.1 แสดง Silo-Oriented Architecture Layer.....	5
2.2 แสดง การทำงานของ Web Service.....	8
2.3 แสดงถึงหลักการดำเนินงานของ Open API Specification.....	11
2.4 แสดงหน้าจอการทำงานของ Swagger Editor.....	13
2.5 แสดงถึงส่วน Header ของ Swagger และ GET method.....	13
2.6 แสดงผลจากการ render Swagger ส่วนของ header.....	14
2.7 แสดงการกำหนด Model ของข้อมูลใน Swagger.....	15
2.8 แสดงผลลัพธ์ที่ได้จากการ Render Model.....	15
2.9 แสดง Parameters ที่อยู่ใน URL Path.....	16
2.10 แสดงผลลัพธ์ที่ได้จากการ RenderGETmethod.....	17
2.11 แสดงการกำหนด POST Method.....	17
2.12 แสดงผลลัพธ์ที่ได้จากการ Render POST method.....	18
2.13 แสดงตัวอย่าง Shell Script สำหรับ Generate Source Code.....	19
2.14 แสดงหน้าเว็บเพจเอกสาร API ที่ได้จาก Swagger UI.....	20
2.15 แสดงตัวอย่างข้อมูล JSON.....	22
3.1 ภาพรวมสถาปัตยกรรมของระบบเว็บเซอร์วิสสำหรับระบบการตรวจสอบ ประสิทธิภาพการขาย.....	27
3.2 Use Case Diagram สำหรับการตรวจสอบประสิทธิภาพภาพการขาย.....	28
3.4 แผนผังระบบการตรวจสอบประสิทธิภาพการขาย.....	35
3.5 วิเคราะห์ภาพรวมธุรกิจ (Dashboard) นำเสนอข้อมูล.....	36
3.6 วิเคราะห์ลูกค้าทั้งหมด (Customer Overview).....	39
3.7 วิเคราะห์รายลูกค้า (Customer Focus).....	42
4.1 กรณีทดสอบ TC01.....	59
4.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	62
4.3 กรณีทดสอบ TC02.....	63
4.4 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	64
4.5 กรณีทดสอบ TC03.....	65

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.6 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	66
4.7 กรณีทดสอบ TC04.....	66
4.8 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	67
4.9 กรณีทดสอบ TC05.....	68
4.10 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	69
4.11 กรณีทดสอบ TC06.....	70
4.12 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	71
4.13 กรณีทดสอบ TC07.....	72
4.14 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	73
4.15 กรณีทดสอบ TC08.....	74
4.16 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	75
4.17 กรณีทดสอบ TC09.....	76
4.18 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	77
4.19 กรณีทดสอบ TC10.....	78
4.20 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	79
4.21 กรณีทดสอบ TC11.....	80
4.22 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	81
4.23 กรณีทดสอบ TC12.....	82
4.24 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	83
4.25 กรณีทดสอบ TC13.....	84
4.26 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	85
4.27 กรณีทดสอบ TC14.....	86
4.28 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	87
4.29 กรณีทดสอบ TC15.....	88
4.30 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	89
4.31 กรณีทดสอบ TC16.....	90
4.32 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	91

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.33 กรณีทดสอบ TC17.....	92
4.34 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	93
4.35 กรณีทดสอบ TC18.....	94
4.36 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	95
4.37 กรณีทดสอบ TC19.....	96
4.38 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน.....	97



บทที่ 1

บทนำ

1.1 ที่มา และความสำคัญของงาน

จากประสบการณ์การทำงานในด้าน Supply Chain & Logistics มากกว่า 15 ปีในอุตสาหกรรมที่หลากหลาย อาทิ เช่น Packaging Products, Consumer Products, Medical Products, Engineering Products ในองค์กรชั้นนำของประเทศ การให้ความสำคัญในการนำเทคโนโลยีมาช่วยในการเพิ่มประสิทธิภาพทั้งในด้านการผลิต การขายและการตลาด การบริหารจัดการ Backoffice รวมถึงทรัพยากรบุคคล เป็นเรื่องสำคัญอย่างยิ่ง เพื่อเป็นแต้มต่อในการแข่งขัน และความยั่งยืนขององค์กร

ด้วยความตั้งใจที่ต้องการให้ผู้ประกอบการ SMEs ไทยได้เรียนรู้วิธีการนำเทคโนโลยีมาใช้ อย่างองค์กรชั้นนำทั่วไป การทำ Digital Transformation ด้วยวิถีไทยๆ เป็นเรื่องจำเป็น เนื่องจาก SMEs ไทยยังมีข้อจำกัด หลายด้าน เช่น ความพร้อมของบุคลากรภายในองค์กร ความพร้อมในการลงทุนเทคโนโลยี ยังคงใช้ซอฟต์แวร์ที่เป็นเทคโนโลยีเก่า รวมถึงผู้ประกอบการเอง ยังคงต้องทำหลายๆหน้าที่ในองค์กร ด้วยข้อจำกัดเหล่านี้ เมื่อปี 2559 จึงได้พัฒนา และเริ่มให้บริการระบบบริหารจัดการงานขายครบวงจร ชื่อว่า “iSmartSales” (ไอสมาร์ตเซลล์) ในรูปแบบ Software As A Services (SAAS) เป็น Mobile และ Web Applications ซึ่งสามารถต่อเชื่อมกับระบบงานเดิม (Legacy System) เพื่อให้ไม่เกิดการปรับตัวมากนักสำหรับ SMEs โดยล่าสุดปี 2562 มีการซื้อขายผ่านระบบประมาณ 7 พันล้านบาท

สารนิพนธ์นี้เป็นการพัฒนาต่อยอด ในเรื่องของการพัฒนา Web Services หรือ Web API เพื่อแลกเปลี่ยนข้อมูลระหว่างระบบ “iSmartSales” กับระบบการตรวจสอบประสิทธิภาพการขาย เพื่อให้ SMEs สามารถนำข้อมูลในองค์กรมาใช้ในการวิเคราะห์และตัดสินใจ บนระบบที่มีความถูกต้อง รวดเร็ว และเชื่อถือได้

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อปรับบริการข้อมูลการตรวจสอบประสิทธิภาพการขายที่มีอยู่ปัจจุบัน ให้เหมาะสมกับผู้ใช้ระดับ SMEs

1.2.2 เพื่อปรับบริการข้อมูลการตรวจสอบประสิทธิภาพการขายที่มีอยู่ปัจจุบัน ซึ่งใช้แพลตฟอร์มเมืองนอกในการพัฒนา ให้เป็นแบรนด์ของตนเอง

1.3 ประโยชน์และผลที่คาดว่าจะได้รับ

1.3.1 ชิ้นงานที่สามารถนำไปใช้ได้จริง และสามารถลดต้นทุนในการดำเนินการที่เดิมต้องใช้นักงานในการจัดเตรียมข้อมูลอย่างน้อยต้องใช้เวลาประมาณ 3-7 วันทำการ และยังสามารถเกิด Human Error ระหว่างดำเนินการ โดยระบบ Web API นี้จะช่วยทำให้ข้อมูลสามารถนำเสนอได้รวดเร็วขึ้น มีความถูกต้อง และเชื่อถือได้

1.3.2 Frontend Developer สามารถเรียกใช้ข้อมูลจาก Web API ในฝั่ง Server จากการจัดทำเอกสารอธิบาย REST API

1.4 ขอบเขตการศึกษา

การออกแบบ และพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย ได้นำเทคโนโลยีเว็บแอปพลิเคชัน มาประยุกต์ใช้เพื่อให้การทำงานของระบบเป็นไปตามวัตถุประสงค์ที่ตั้งไว้ ซึ่งมีขอบเขต ดังนี้

1.4.1 ทำการศึกษาระบบงานเดิม เพื่อเป็นต้นแบบ และคัดเลือกเฉพาะตัวชี้วัดที่สำคัญมาใช้

1.4.2 พัฒนาเว็บเซอร์วิส หรือเว็บ API เพื่อเป็นตัวกลางในการทำงานเชื่อมต่อรับส่งข้อมูลภายในระบบระหว่าง Server และ Client

1.4.3 นำเครื่องมือ และเทคโนโลยีด้านการจัดทำเอกสารการอธิบาย REST API มาใช้จัดทำเอกสาร เพื่ออำนวยความสะดวก และลดเวลาการทำงานให้ Frontend Developer ที่ต้องเรียกใช้เว็บ API ฝั่ง Server

1.4.4 เว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย ครอบคลุมการให้บริการข้อมูลเพื่อนำไปใช้เป็นตัวชี้วัด โดยให้ความสำคัญกับมุมมอง ดังนี้

1.4.4.1 ข้อมูลการตรวจสอบประสิทธิภาพโดยรวม (Performance Overall)

1.4.4.2 ข้อมูลการตรวจสอบประสิทธิภาพของลูกค้าทั้งหมด (Customer Performance Overall)

1.4.4.3 ข้อมูลการตรวจสอบประสิทธิภาพของลูกค้าเฉพาะราย (Customer Focus)



บทที่ 2

แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในการพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย ผู้วิจัยได้ทำการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง เพื่อเป็นแนวทางในการวิเคราะห์ และออกแบบระบบให้มีประสิทธิภาพ โดยเนื้อหาที่ผู้วิจัยศึกษามีดังนี้

1. สถาปัตยกรรมซอฟต์แวร์เชิงบริการ SOA (Service-Oriented Architecture)
2. เว็บเซอร์วิส (Web Service)
3. Open API Specification (OAS)
4. JavaScript Object Notation (JSON)
5. NEST JS
6. Business Intelligence

2.1 สถาปัตยกรรมซอฟต์แวร์เชิงบริการ SOA (Service-Oriented Architecture)

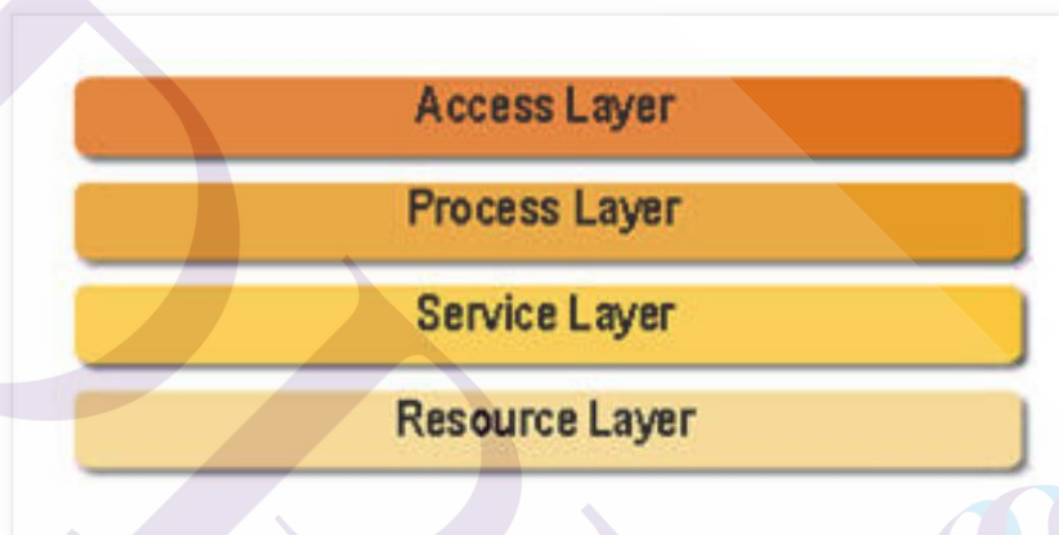
ระบบสถาปัตยกรรมเชิงบริการหรือ SOA เป็นแนวคิดในการจะออกแบบระบบไอทีในองค์กรให้เป็นระบบเชิงบริการ (Service-Oriented) ที่สามารถนำกลับมาใช้ใหม่ได้ ทั้งนี้ระบบไอทีขององค์กรต่างๆ ในปัจจุบันมักจะมีสถาปัตยกรรมแบบ Silo-Oriented Architecture ซึ่งการพัฒนา ระบบไอทีในแต่ละระบบต่างเป็นอิสระต่อกัน อาจมีระบบที่ใช้เทคโนโลยีที่แตกต่างกันเช่น Java, .NET, Oracle หรือ SAP เป็นต้น จึงทำให้ยากต่อการเชื่อมต่อ บำรุงรักษายาก มีค่าใช้จ่ายสูงปรับเปลี่ยนระบบได้ยาก และการพัฒนาระบบใหม่ๆ เป็นไปด้วยความล่าช้า

แนวคิดของระบบ SOA คือการจัดระบบ Silo-Oriented Architecture ใหม่ โดยการสร้างระบบไอทีให้เป็น 4 ชั้น (Layer)

1. Access Layer ซึ่งเป็นชั้นของการเรียกใช้กระบวนการทางธุรกิจที่พัฒนาขึ้น โดยอาจผ่านทางเว็บไซต์ (Web Site) หรือโทรศัพท์เคลื่อนที่ (Mobile Phone)
2. Process Layer ซึ่งเป็นชั้นของกระบวนการทางธุรกิจ (Business Process) ที่พัฒนาขึ้นมาจากการส่วนประกอบเซอร์วิสต่าง ๆ

3. Service Layer ซึ่งเป็นชั้นของส่วนประกอบเซอร์วิสต่าง ๆ ที่สามารถนำมาใช้ใหม่ได้ โดยส่วนประกอบเซอร์วิสเหล่านี้จะพัฒนามาจากโมดูล (Module) ต่าง ๆ ที่ทำงานบน Resource Layer เช่น โมดูลของฐานข้อมูล Oracle โมดูลของระบบโซลูชัน SAP หรือ PeopleSoft และโมดูลของโปรแกรมประยุกต์ที่อาจพัฒนาด้วย Java หรือ .NET เป็นต้น

4. Resource Layer ซึ่งจะเป็นชั้นของระบบโครงสร้างไอทีต่าง ๆ ในปัจจุบัน เช่น ระบบฐานข้อมูล Oracle ระบบโซลูชัน SAP หรือ PeopleSoft เป็นต้น



ภาพที่ 2.1 แสดง Silo-Oriented Architecture Layer

ระบบ SOA จะมีคุณลักษณะที่สำคัญดังนี้

1. การติดต่อสื่อสารระหว่างเซอร์วิส จะใช้เอกสารที่เป็น XML ที่นิยามผ่าน XML Schema (.xsd) ทำให้ไม่จำเป็นต้องทราบรายละเอียดของแพลตฟอร์ม และเทคโนโลยีของเซอร์วิสที่ใช้อยู่

2. เซอร์วิสจะมีตัวเชื่อมต่อ (Interface) ที่อธิบายเซอร์วิส เช่น Service Name, Input Parameter, Output Parameter และข้อมูลอื่น ๆ ในรูปแบบของไฟล์ XML ทำให้ไม่ขึ้นกับแพลตฟอร์ม และเทคโนโลยีที่เซอร์วิสนั้นใช้อยู่ โดยมากมักจะใช้มาตรฐาน WSDL (Web Service Description Language) ในการอธิบายเซอร์วิส

3. โปรแกรมประยุกต์ (Application) หรือกระบวนการทางธุรกิจต่าง ๆ สามารถพัฒนาขึ้นมาจากการใช้เซอร์วิสเดิมที่มีอยู่ ซึ่งมาตรฐานที่นิยมใช้คือ WS-BPEL (Web Service Business Process Execution Language)

4. SOA จะมี Registry ในการเก็บเซอร์วิสต่างๆ ที่มีอยู่ ซึ่ง Registry จะทำหน้าที่เหมือนไดเรกทอรีของเซอร์วิส โดยโปรแกรมประยุกต์หรือกระบวนการทางธุรกิจต่างๆ จะค้นหาและเรียกใช้เซอร์วิสจาก Registry นี้ มาตรฐานที่ใช้ในการเก็บ Registry ที่นิยมใช้คือ UDDI (Universal Description Definition and Integration)

5. เซอร์วิสแต่ละตัวจะมีส่วนการควบคุมคุณภาพที่เป็น QoS (Quality of Service) อาทิ เช่นการควบคุมความปลอดภัยด้าน Authentication, Authorization, Reliable Message และ Policy ประโยชน์ของการพัฒนา SOA

การพัฒนาระบบโครงสร้างไอทีในองค์กรให้เป็นระบบ SOA จะเกิดประโยชน์ในด้านต่าง ๆ ดังนี้

1. สามารถเชื่อมโยงธุรกิจต่าง ๆ

การพัฒนา SOA มีการเชื่อมโยงระบบไอทีต่าง ๆ ภายในองค์กร และภายนอกองค์กรที่อาจใช้เทคโนโลยีที่ต่างกัน ทำให้เราสามารถเชื่อมโยงธุรกิจต่าง ๆ ที่อาจอยู่ต่างระบบกัน และสามารถให้บริการกับลูกค้า คู่ค้า และบุคลากรในองค์กรได้

2. ระบบไอทีสามารถปรับเปลี่ยนได้ง่าย

การพัฒนา SOA สามารถที่จะทำให้น่าระบบไอทีเดิมมาใช้ใหม่ได้ ดังนั้นการปรับเปลี่ยนกระบวนการทางธุรกิจจึงเป็นไปได้อย่างรวดเร็ว และทำให้สามารถแข่งขันในตลาดธุรกิจได้อย่างรวดเร็ว

3. การลดค่าใช้จ่ายในการบำรุงรักษา และให้ผลตอบแทนการลงทุนที่คุ้มค่า

การพัฒนา SOA ทำให้องค์กรสามารถที่จะใช้เทคโนโลยีที่หลากหลาย จึงทำให้เราสามารถที่จะเลือกใช้เทคโนโลยีต่าง ๆ ได้ โดยไม่ต้องผูกติดกับเทคโนโลยีใดเทคโนโลยีหนึ่ง ทำให้ค่าใช้จ่ายด้านไอทีในระยะยาวลดลง

4. การทำงานของฝ่ายธุรกิจ และฝ่ายไอทีสอดคล้องกันมากขึ้น

การพัฒนา Business Process ของฝ่ายไอทีจะมีขั้นตอนที่ชัดเจนสามารถแสดงในเชิงกราฟฟิกได้ และเข้าใจง่าย ขึ้น และหน่วยงานทางธุรกิจที่ต้องเข้าใจด้านกระบวนการทางธุรกิจสามารถที่จะเข้ามา ร่วมทำการพัฒนาร่วมกับฝ่ายไอทีได้ดียิ่งขึ้น

2.2 เว็บเซอร์วิส

Web service (เว็บเซอร์วิส) คือการให้บริการแลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ในโลก internet ซึ่งถูกออกแบบมาตอนแรกเพื่อใช้ระหว่าง human-to-machine ผ่าน HTTP ต้องถูกพัฒนามาใช้กับ machine-to-machine ในรูปแบบ XML และ JSON format นั่นคือคำตอบว่า Web service คืออะไร

ในทางเทคนิค “Web Service” เป็นมาตรฐานในการเชื่อมต่อ Web-Base Application ที่วิ่งอยู่บน Internet Protocol โดยอาศัย XML, SOAP, WSDL และ UDDI

1. XML (Extensible Markup Language) เป็น format ที่จะบอกแหล่งที่มา และรายละเอียดข้อมูลได้
2. SOAP หรือ REST เป็น protocol ที่ใช้แลกเปลี่ยนข้อมูล
3. WSDL (Web Services Description Language) ใช้ในการอธิบายโครงสร้างของ service ทั้งหมด หรือ คู่มือการใช้งาน
4. UDDI (Universal Description, Discovery, and Integration) เป็นส่วนกลางให้ระบบอื่นมาลงทะเบียนไว้ โดยใช้ WSDL file ทำให้ฝั่ง Client สามารถดึงเอาข้อมูลไปใช้ติดต่อกับ web service ได้

จากข้อมูลอ้างอิงหลายที่ จะพบว่ามีข้อมูลเกี่ยวกับ Web Service หลากหลายมาก จึงขอสรุปเป็นหัวข้อดังนี้

1. เปิดให้ใช้งานได้ทั้ง Internet และ Intranet
2. ใช้มาตรฐาน XML format
3. ไม่ได้ยึดติดกับ Operating system ชนิดใด หรือ ภาษา programming ชนิดไหน
4. สามารถทำความเข้าใจคำสั่งด้วยลักษณะการทำงานแบบเดียวกับ XML

มี Software จำนวนมากที่ถูกเขียนในภาษา Programming ที่ต่างกัน แต่ทุกภาษาต่างก็มีความจำเป็นต้องแลกเปลี่ยนข้อมูลกับ Program อื่นซึ่งตัว Web Service เองก็ยอมให้ทำได้ด้วยการใช้ XML tag เป็นตัวเก็บข้อมูล ซึ่งโครงสร้างข้อมูลจะถูกเก็บไว้ที่ WSDL (Web Services Description Language) ภายใที่จะบอกถึงรายละเอียดการทำงานของ Web Service ดังนี้

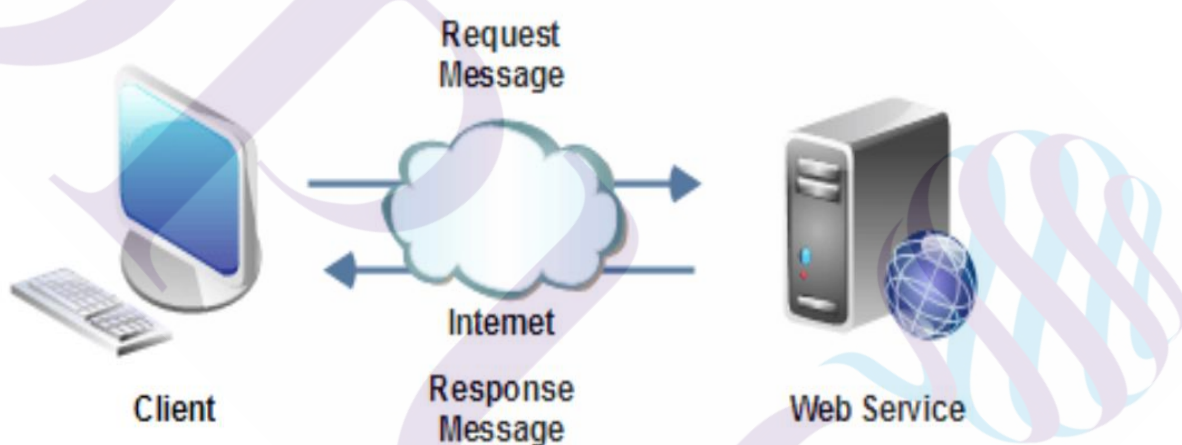
1. ระบบหนึ่งจะทำการส่งคำสั่งไปยังอีกระบบได้อย่างไร
2. Parameter ใหนที่ต้องใส่ในการส่งคำสั่ง
3. โครงสร้างของข้อมูล XML ต้องมีลักษณะอย่างไร
4. Error Message ที่แสดงหมายถึงคำสั่งที่ส่งมาผิดพลาดในส่วนไหน เพื่องานต่อการ

แก้ไข

ขั้นตอนการทำงานของ Web service

ส่วนมากนิยมเขียนกันด้วย Java และในการอธิบายตัวอย่างใช้ Unix Solaris ที่ทำการต่อกับ database เพื่อเก็บข้อมูล

1. ผู้ Client ทำการ โหลดข้อมูลต่าง ๆ ที่ใช้ในการติดต่อไปยัง server เข้าสู่ SOAP message
2. ส่ง SOAP message ไปยัง Web Service ด้วยการทำให้ HTTP POST
3. ผู้ Web service ก็จะทำการแกะข้อมูลออกจาก SOAP และเปลี่ยนเป็นคำสั่งที่ Application เข้าใจ
4. ในส่วน Application ก็จะนำเอาคำสั่งหรือข้อมูลที่ได้ไปทำงานจนได้ผลลัพธ์ ที่ต้องส่งกลับไปหาผู้ Client จัดใส่ใน HTTP format
5. ผู้ Client ก็จะแกะข้อมูลออกจาก SOAP Message เพื่อเอาผลลัพธ์



ภาพที่ 2.2 แสดง การทำงานของ Web Service

ข้อแตกต่างระหว่าง Web service กับ Web server

Web Server

1. เป็น software ที่ทำงานบน server ถูกออกแบบเพื่อรองรับการทำงานของ web page, web site รวมถึง web service
2. Web server ทำงานโดยอาศัย HTTP protocol
3. ข้อมูลที่วิ่งผ่าน web server คือ HTTP request และ HTTP response

4. Software ที่ใช้ส่วนใหญ่คือ IIS, Apache

Web Services

1. เป็น Application ที่ทำงานบน Web Server ทำหน้าที่รับส่งโครงสร้างข้อมูลให้กับ Program ที่เรียกใช้ ซึ่งเป็นมากกว่าการทำงานของ HTML ที่ Browser เรียกใช้งาน

2. สามารถใช้งานกับข้อมูลได้หลาย Format เช่น XML, JSON

3. Web Service ไม่จำเป็นต้องอาศัย HTTP Protocol

4. ปัจจุบันมี Software หลัก ๆ ที่ใช้เป็น Framework คือ Apache Axis, JWS หรือ .NET
ข้อดีของ Webservice

1. มีมาตรฐานในการใช้งาน

2. มีความยืดหยุ่นในการใช้งาน สามารถปรับแต่งตามความต้องการ และใช้งานกัน

แพร่หลาย

3. สามารถทำการ upgrade ได้อัตโนมัติ

4. ลดค่าใช้จ่ายในการเชื่อมต่อหรือสื่อสาร

5. ทำงานได้หลายหลายภาษา

6. มีมาตรฐานความปลอดภัยเพียงพอกับองค์กรขนาดใหญ่



2.3 Open API Specification

เนื่องจากการพัฒนาเว็บเซอร์วิสในรูปแบบ REST ยังไม่มีมาตรฐานที่กำหนดให้บังคับใช้งานเหมือน SOAP ซึ่งมาตรฐานต่าง ๆ เหล่านี้รวมไปถึงมาตรฐานที่ใช้อธิบายเว็บเซอร์วิส โดยในฝั่งของ SOAP เองจะมี WSDL (Web Services Description Language) ซึ่งจะเป็นเอกสารที่อยู่ในรูปแบบของ XML มีไว้ใช้อธิบายรายละเอียดของเว็บเซอร์วิส เพื่อให้แอปพลิเคชัน ที่ต้องการเรียกใช้เว็บเซอร์วิส ทราบรายละเอียดของบริการ และวิธีการเรียกใช้ โดย WSDL จะบรรจุรายละเอียดที่ต้องใช้ในการสร้าง Request message เพื่อร้องขอบริการเช่น ชื่อบริการ พารามิเตอร์ หรือโปรโตคอลที่ใช้ในการติดต่อสื่อสาร เพราะจะต้องมีเอกสาร Service Description เพื่อบอกรายละเอียด และวิธีการเรียกใช้ Service ซึ่งในฝั่งการทำงานของ REST นั้นเนื่องจากไม่มีมาตรฐานกำหนดไว้ปัญหาใหญ่ที่พบบ่อยในการพัฒนาระบบคือการทำเอกสารเพื่ออธิบาย API ต่างๆ ที่พัฒนาขึ้นจึงทำให้มีการรวมกลุ่มขึ้นมาเพื่อสร้างข้อกำหนด เพื่อเป็นกรอบหรือแนวทางการทำงานกับ RESTful Web Service เรียกว่า Open API Specification

2.3.1 ประวัติความเป็นมา

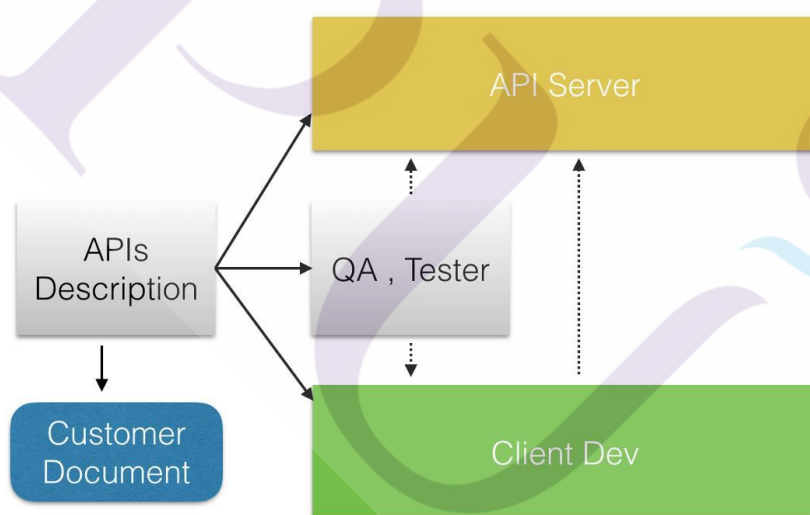
Open API Specification (OAS) หรือที่รู้จักกันครั้งแรกในชื่อ Swagger Specification ซึ่งเกิดขึ้นจากการรวมตัวของกลุ่มอุตสาหกรรม และนักพัฒนาซอฟต์แวร์ โดยใช้ชื่อว่า Open API Initiative วัตถุประสงค์คือการกำหนดมาตรฐาน และสร้างกรอบการทำงาน สำหรับ REST APIs ขึ้นมา โดย Open API Specification นั้น จะเป็นข้อกำหนดสำหรับการสร้างชุดข้อมูลที่สามารถอ่านและเข้าใจได้โดย มนุษย์และคอมพิวเตอร์ ซึ่งข้อมูลชุดนี้จะสามารถอธิบายโครงสร้างของ API ได้ว่ามีการทำงานอย่างไร มีการส่งข้อมูล และได้ผลตอบกลับอย่างไร รวมไปถึงความสามารถในการนำไปใช้ประโยชน์ในการทำงานอื่น ๆ เช่น การสร้างเอกสาร และการสร้างชุดของ Source code เพื่อรองรับการทำงานร่วมกับ API ในฝั่ง Server ซึ่งปัจจุบันได้มีการพัฒนาเครื่องมือหลายๆตัว ขึ้นมาเพื่อใช้ทำงานร่วมกัน ภายใต้ Open API Specification

Open API Specification เกิดขึ้นจากการริเริ่มของ Wordnik ซึ่งเป็นองค์กรไม่แสวงหาผลกำไร โดยตัว Wordnik เองเป็นแหล่งข้อมูลทางภาษาที่ให้บริการ Dictionary ภาษาอังกฤษแบบออนไลน์ รวมไปถึง Thesaurus เก็บรวบรวมข้อมูลอธิบายความหมายของคำที่มีความหมายเหมือนหรือคล้ายกัน โดยเริ่มจาก Swagger ซึ่งถูกพัฒนาขึ้นเพื่อใช้ในการทำงานภายในของ Wordnik เอง โดยมีการเริ่มพัฒนาขึ้นในช่วงต้น ปี ค.ศ. 2010 และในเดือนมีนาคม ปี 2015 บริษัท Smart Bear Software ได้เข้ามาพัฒนา Swagger API Specification ซึ่งเป็น Open Source Project ต่อจาก Reverb Technologies และ ในช่วงเดือนพฤศจิกายนของปี เดียวกัน Smart Bear ได้ประกาศว่า กำลังจะมีการจัดตั้งองค์กรใหม่ขึ้นด้วย การสนับสนุนจาก Linux Foundation โดยใช้ชื่อว่า Open API Initiative

(OAS) โดยมีหลายบริษัทชั้นนำ เป็นผู้ร่วมก่อตั้ง เช่น Google, IBM และ Microsoft ซึ่ง Smart Bear ได้แยก Swagger Specification ซึ่งมีเวอร์ชันล่าสุดคือ 2.0 ให้กับทางกลุ่ม ทั้งนี้ RAML และ API Blueprint ซึ่งเป็น Framework ที่มีวัตถุประสงค์การทำงานคล้ายกัน ก็อยู่ในรายชื่อที่ได้รับการพิจารณาจากกลุ่มด้วย และต่อมาเมื่อวันที่ 1 มกราคมปี 2016 Swagger Specification ได้ถูกเปลี่ยนชื่อมาเป็น Open API Specification อย่างเป็นทางการ และในปี 2017 ก็ได้ออก Preview เวอร์ชัน 3.0 โดยใช้ชื่อว่า Open API Specification 3.0 โดยมี ทาง MuleSoft ที่เป็นผู้ร่วมพัฒนาหลักของโปรเจกต์ทางเลือกอย่าง RESTful API Modeling Language (RAML) เข้ามาร่วมทำงานกับ OAS อีกด้วย

2.3.2 การนำ Open API Specification มาใช้งาน

แอปพลิเคชันที่ถูกพัฒนาขึ้นมาบนพื้นฐานของ Open API Specification จะสามารถสร้างเอกสารอธิบาย Method, Parameter และ Model ของ Response ในการเรียกใช้งานแต่ละ Service ได้โดยอัตโนมัติซึ่งช่วยให้ เอกสาร รวมถึง Client Library และ การทำงานในฝั่ง Server มีความสอดคล้องกัน โดยการกำหนดมาตรฐานนี้จะบังคับให้เราเขียน API Description ให้อยู่ในรูปแบบที่ได้กำหนดไว้ เพื่อที่จะใช้เป็นตัวกลางในการสื่อสารระหว่างผู้ที่เกี่ยวข้องตั้งแต่ Developer, Tester รวมถึงไปถึงคนทำเอกสารด้วย



ภาพที่ 2.3 แสดงถึงหลักการทำงานของ Open API Specification

เรายังสามารถนำ Swagger หรือ Open API เข้ามาช่วยในการพัฒนา API ของเราได้ดังต่อไปนี้

1. สำหรับการพัฒนาแบบ Design-first หรือการออกแบบ API ไว้ก่อน แล้วจึงลงมือเขียนโค้ดตามนั้นสามารถใช้ Swagger Codegen เพื่อสร้าง Server Stub สำหรับ API เพื่อเป็นการ Mock-up ข้อมูลเพื่อใช้ในการทดสอบได้ก่อนแล้วจึงเขียน Serverlogic ตามไปที่หลัง วิธีนี้สามารถทำให้การพัฒนาแอปพลิเคชันฝั่ง Server และ Client สามารถทำงานไปพร้อมกันได้

2. สามารถใช้ Swagger Codegen เพื่อสร้าง Library ในฝั่ง Client เพื่อเรียกใช้งาน API โดยมีภาษาที่รองรับมากกว่า 40 ภาษา

3. สามารถใช้ Swagger UI เพื่อสร้าง Interactive API documentation เพื่อให้ให้นักพัฒนาสามารถลองยิงทดสอบเรียก API ได้เลยผ่านเว็บเบราว์เซอร์

4. สามารถใช้ไฟล์ Spec ทำงานร่วมกับเครื่องมือที่เกี่ยวข้องกับ API อื่น ๆ เพื่อเชื่อมต่อเรียกใช้งาน API ของเราเช่นนำเข้าไฟล์ Spec ไปยัง SoapUI เพื่อทำ AutomatedTest กับ API ที่พัฒนาขึ้น

2.3.3 เครื่องมือที่ใช้กับ Open API Specification

เครื่องมือหลักที่ถูกพัฒนาขึ้นมาเพื่อใช้กับ Open API Specification มีดังต่อไปนี้

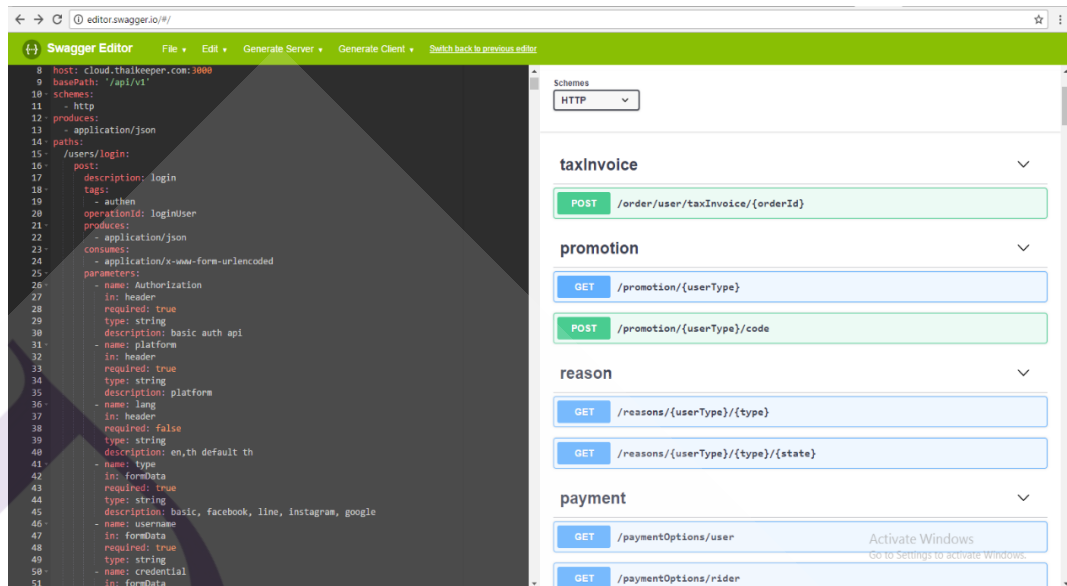
Swagger Editor

Swagger Codegen

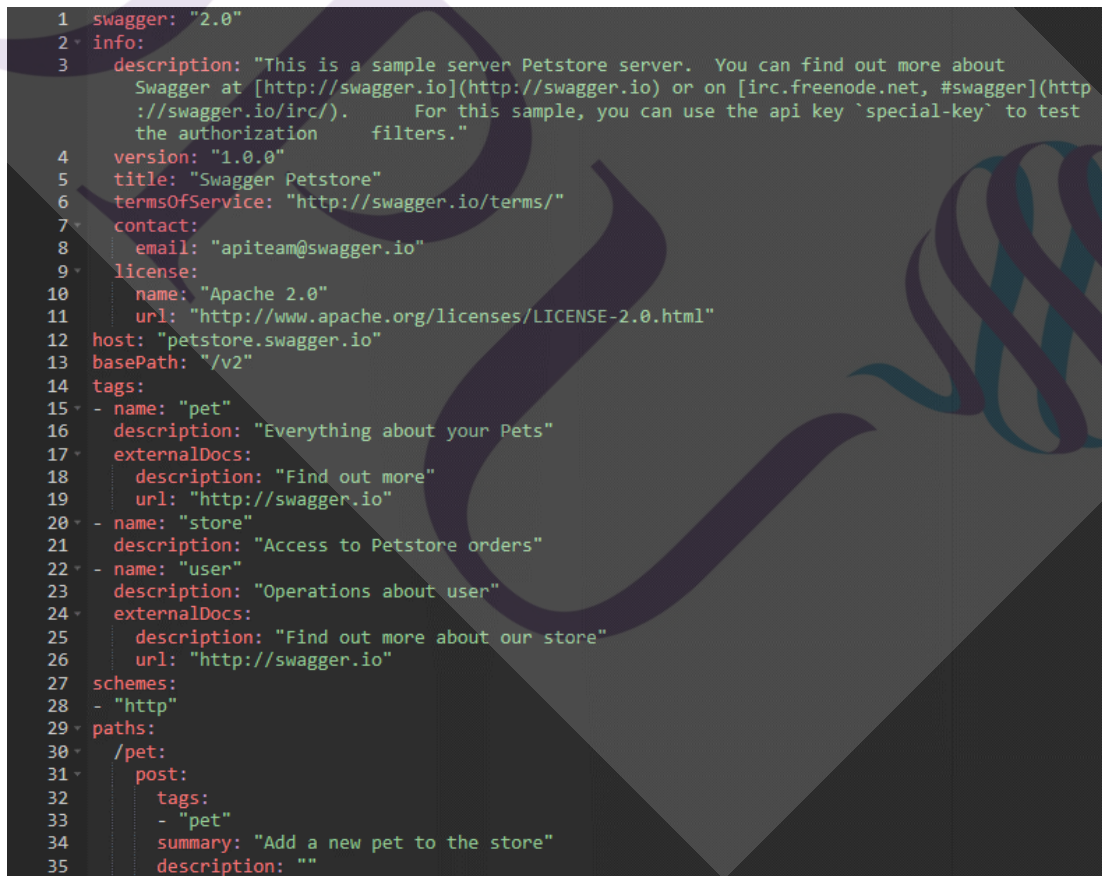
Swagger UI

2.3.3.1 Swagger Editor

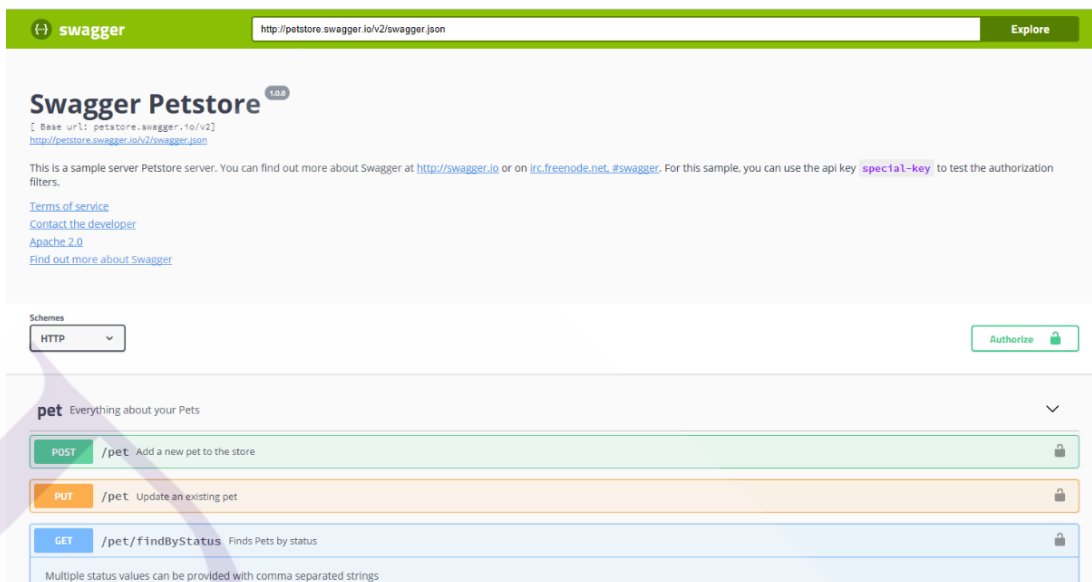
Swagger Editor เป็นเครื่องมือที่ไว้ช่วยเขียน API Description รองรับทั้งแบบ Yaml และ JSON ซึ่งนอกจากการทำตัวเป็น TextEditor แล้ว ทางด้านขวาจะมี Live-render จาก API Description ที่เขียนมาแสดงและยังสามารถส่ง Request ทดสอบ API ได้จริงอีกด้วย



ภาพที่ 2.4 แสดงหน้าการทำงานของ Swagger Editor



ภาพที่ 2.5 แสดงถึงส่วน Header ของ Swagger และ GET method



ภาพที่ 2.6 แสดงผลจากการ render Swagger ส่วนของ header

ใน swagger จะมีการกำหนด Model ของ Object เพื่อสามารถใช้กำหนดให้เป็นข้อมูล Response ที่ได้จากการเรียก API จากตัวอย่างจะเป็น Model ของ Order ที่ประกอบไปด้วย property id, petId, quantity เป็นต้น

```

578 definitions:
579   Order:
580     type: "object"
581     properties:
582       id:
583         type: "integer"
584         format: "int64"
585       petId:
586         type: "integer"
587         format: "int64"
588       quantity:
589         type: "integer"
590         format: "int32"
591       shipDate:
592         type: "string"
593         format: "date-time"
594       status:
595         type: "string"
596         description: "Order Status"
597         enum:
598           - "placed"
599           - "approved"
600           - "delivered"
601       complete:
602         type: "boolean"
603         default: false

```

ภาพที่ 2.7 แสดงการกำหนด Model ของข้อมูลใน Swagger



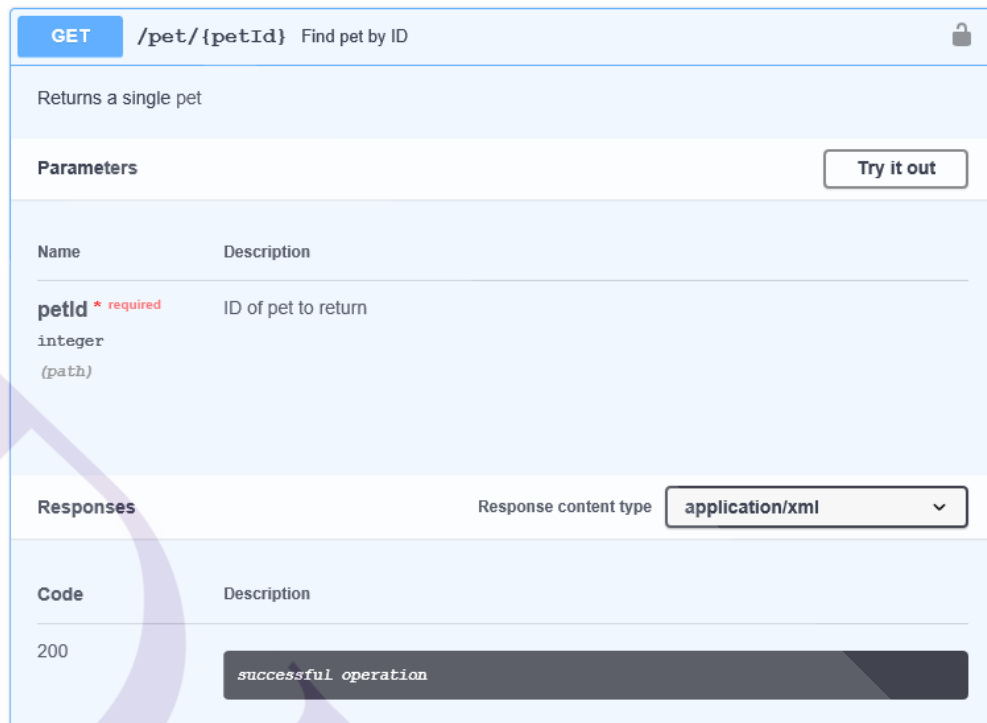
ภาพที่ 2.8 แสดงผลลัพธ์ที่ได้จากการ Render Model

ต่อไปจะเป็นการกำหนด GET Method ที่ใช้ model “Pet” มาเป็นค่า response รวมไปถึงการกำหนด Parameter ของ API แบบที่อยู่ใน URL Path ด้วย

```
157 /pet/{petId}:
158   get:
159     tags:
160     - "pet"
161     summary: "Find pet by ID"
162     description: "Returns a single pet"
163     operationId: "getPetById"
164     produces:
165     - "application/xml"
166     - "application/json"
167     parameters:
168     - name: "petId"
169       in: "path"
170       description: "ID of pet to return"
171       required: true
172       type: "integer"
173       format: "int64"
174     responses:
175     200:
176       description: "successful operation"
177       schema:
178         $ref: "#/definitions/Pet"
179     400:
180       description: "Invalid ID supplied"
181     404:
182       description: "Pet not found"
```

ภาพที่ 2.9 แสดง Parameters ที่อยู่ใน URL Path

Live render view จะแสดง GET Method ในรูปด้านล่าง จากตรงนี้หากต้องการทดสอบ
ลองยิง Request API ก็สามารกดกดที่ปุ่ม [Try it out]



ภาพที่ 2.10 แสดงผลลัพธ์ที่ได้จากการRenderGETmethod

```

29 paths:
30   /pet:
31     post:
32       tags:
33         - "pet"
34       summary: "Add a new pet to the store"
35       description: ""
36       operationId: "addPet"
37       consumes:
38         - "application/json"
39         - "application/xml"
40       produces:
41         - "application/xml"
42         - "application/json"
43       parameters:
44         - in: "body"
45           name: "body"
46           description: "Pet object that needs to be added to the store"
47           required: true
48         schema:
49           $ref: "#/definitions/Pet"
50       responses:
51         405:
52           description: "Invalid input"

```

ภาพที่ 2.11 แสดงการกำหนด POST Method

POST /pet Add a new pet to the store

Parameters Try it out

Name	Description
body * required <i>(body)</i>	Pet object that needs to be added to the store

Example Value Model

```

{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}

```

Parameter content type: **application/json**

Responses Response content type: **application/xml**

ภาพที่ 2.12 แสดงผลลัพธ์ที่ได้จากการ Render POST method

2.3.3.2 Swagger Codegen

Swagger Codegen เป็น Open Source Code-generator ใช้สำหรับสร้าง Server Stub และ SDK สำหรับฝั่ง Client จาก API Description ที่เราได้เขียนขึ้นมา วิธีการใช้งานหลังติดตั้งหลักๆ จะต้องเรียกเป็น Command line เพื่อ generate ตัว Source Code ออกมาหรือสามารถเขียนเป็น Shell Script เพื่อ Run คำสั่งเหล่านั้น ซึ่งใน Project Swagger Codegen ได้รวมตัวอย่างไฟล์ Script เหล่านี้ เข้าไว้ในตัวอย่างของ Project ด้วย โดยก่อนจะติดตั้ง และเรียกใช้งานจำเป็นต้องติดตั้ง Java เวอร์ชัน 7 ขึ้นไปก่อนจึงจะสามารถติดตั้ง และเรียกใช้งาน Swagger Codegen ได้ นักพัฒนาสามารถ Download และศึกษาวิธีการติดตั้งรวมถึงวิธีการใช้งานได้จาก GitHub: <https://github.com/swagger-api/swagger-codegen>



```

swift3-sol-rxswift - Notepad
File Edit Format View Help

#!/bin/sh

SCRIPT=""

while [ -h "$SCRIPT" ]; do
  ls=$(ls -ld "$SCRIPT")
  link=$(expr "$ls" : '.*-> \(.*\)')
  if expr "$link" : '/.*' > /dev/null; then
    SCRIPT="$link"
  else
    SCRIPT=$(dirname "$SCRIPT")/"$link"
  fi
done

if [ ! -d "${APP_DIR}" ]; then
  APP_DIR=$(dirname "$SCRIPT")/..
  APP_DIR=$(cd "$APP_DIR"; pwd)
fi

executable="modules/swagger-codegen-cli/target/swagger-codegen-cli.jar"

if [ ! -f "$executable" ]
then
  mvn clean package
fi

# if you've executed sbt assembly previously it will use that instead.
export JAVA_OPTS="${JAVA_OPTS} -XX:MaxPermSize=256M -Xmx1024M -DloggerPath=conf\log4j.properties"
ags="$@ generate -t modules/swagger-codegen/src/main/resources/swift3 -i swagger_sol_api.yaml -l swift3 -c swift3-sol-rxswift.json -o outputgen/sol/swift3/rxswift"

echo "Removing files and folders under samples/client/petstore/java/retrofit2/src/main"
rm -rf outputgen/sol/swift3/rxswift
find outputgen/swift3 -maxdepth 1 -type f ! -name "README.md" -exec rm {} +

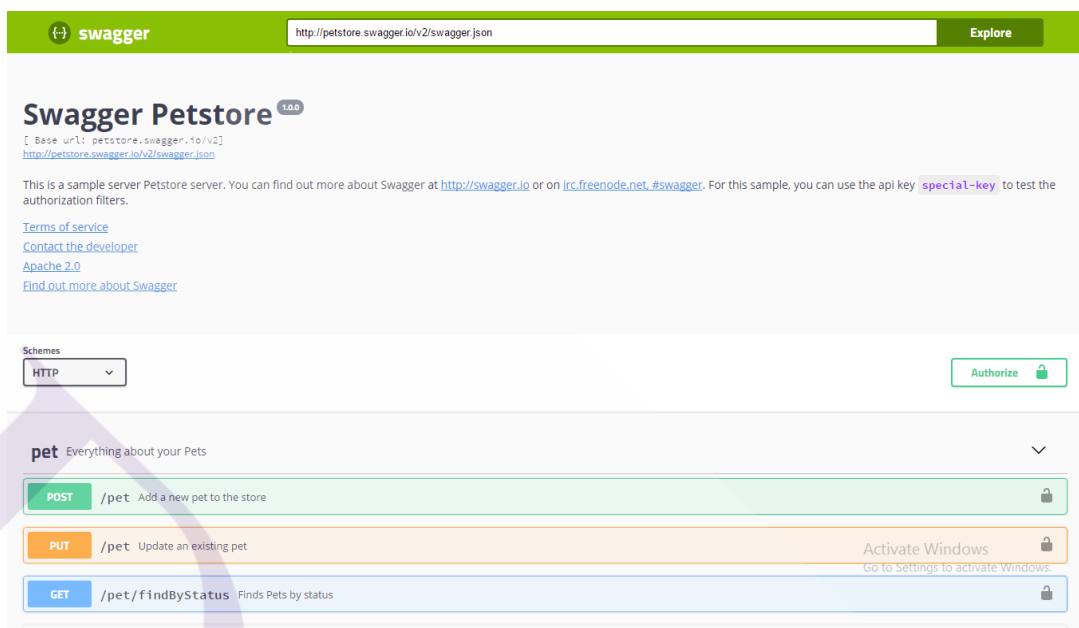
java $JAVA_OPTS -jar $executable $ags

```

ภาพที่ 2.13 แสดงตัวอย่าง Shell Script สำหรับ Generate Source Code

2.3.3.3 Swagger UI

Swagger UI เป็น Open Source Project ใช้สำหรับ Render เอกสารจาก API description ที่เขียนขึ้นให้ออกมาในรูปแบบเว็บเพจ และสามารถเรียกดูเอกสารได้ผ่าน Web Browser โดยสามารถ Download และศึกษาวิธีการติดตั้งรวมถึงวิธีการใช้งานได้จาก GitHub: <https://github.com/swagger-api/swagger-ui>



ภาพที่ 2.14 แสดงหน้าเว็บเพจเอกสาร API ที่ได้จาก Swagger UI

นอกจากเครื่องมือที่กล่าวมานี้ยังมีเครื่องมืออีกหลายตัวทั้งที่เป็น Open Source และที่ใช้ในเชิงพาณิชย์ ที่สามารถทำงานร่วมกับ Swagger Specification หรือ Open API Specification ได้อีกด้วย ซึ่งนักพัฒนาสามารถเข้าไปศึกษารายละเอียดได้จากเว็บไซต์หลักของ Swagger <http://swagger.io/> หรือจากเว็บไซต์ของ Open API Initiative <https://www.OpenAPIs.org/>

2.4 JavaScript Object Notation (JSON)

เจ ซัน (อังกฤษ : JSON: JavaScript Object Notation เ ลี ย ง อ่า น ภาษาอังกฤษ: /ˈdʒeɪsɒn/ แปลว่า สัญกรณ์วัตถุจาวาสคริปต์) เป็นฟอร์แมตสำหรับแลกเปลี่ยนข้อมูลคอมพิวเตอร์ ฟอร์แมต JSON นั้นอยู่ในรูปข้อความธรรมดา (plain text) ที่ทั้งมนุษย์ และ โปรแกรมคอมพิวเตอร์สามารถอ่านเข้าใจได้

มาตรฐานของฟอร์แมต JSON คือ RFC 4627 มี Internet media type เป็น application/json และมีนามสกุลของไฟล์เป็น .json

ปัจจุบัน JSON นิยมใช้ในเว็บแอปพลิเคชัน โดยเฉพาะ AJAX โดย JSON เป็นฟอร์แมตทางเลือกในการส่งข้อมูล นอกเหนือไปจาก XML ซึ่งนิยมใช้กันอยู่แต่เดิม สาเหตุที่ JSON เริ่มได้รับความนิยมเป็นเพราะกระชับ และเข้าใจง่ายกว่า XML

โครงสร้างของฟอร์แมต

เจสัน เป็นรูปแบบสายอักขระ (String) ชนิดหนึ่งที่ถูกจัดเก็บในรูปแบบที่สามารถอ่าน และเข้าใจได้ง่าย ถ้าจะให้เข้าใจง่าย ๆ ก็คือเจสันเปรียบเสมือนรูปแบบของ อาร์เรย์ (Array) ชนิดหนึ่ง ที่ใช้รับส่งข้อมูลผ่านอแจ็กต์เพราะซึ่งปกติแล้วถ้าเราต้องการรับ-ส่งข้อมูลผ่านอแจ็กต์ต่าง ๆ นั้น จะต้องรับ-ส่งมาในรูปแบบของสายอักขระทั้งก่อน และเมื่อฝั่งอแจ็กต์ทำการรับค่าที่ทำการส่งค่า กลับมาจากเซิร์ฟเวอร์ ก็จะต้องนำสายอักขระ เหล่านั้นมาตัดตำแหน่งที่ต้องการ เพื่อเอาสายอักขระ ตัวที่ต้องการมาใช้ แต่สำหรับเจสันแล้ว สามารถรับส่งชุดค่าตัวแปรได้ทั้งฝั่งไคลเอนต์ (Client) และ ฝั่งเซิร์ฟเวอร์ (Server) โดยทั้ง 2 ฝั่งสามารถทำการเข้ารหัส และถอดรหัสโดยใช้เจสันเอนโค้ด (Json Encode) และ เจสันดีโค้ด (Json Decode) เพื่ออ่านค่าตัวแปรเหล่านั้น และจะเรียกใช้งานมันได้ อย่างไร ซึ่งปกติแล้วจะอยู่ในรูปแบบของอาร์เรย์ และสำหรับตัวแปรเจสันนั้นไม่จำกัดแค่รับส่งข้อมูล ผ่านเว็บเบราว์เซอร์เท่านั้น แต่ยังสามารถนำเจสันไปประยุกต์กับการรับส่งข้อมูลในรูปแบบอื่น ๆ ได้ เช่นการจับเก็บข้อมูลในรูปแบบของ สายอักขระในข้อความหรือการรับส่งผ่านตัวให้บริการ เว็บไซต์ (Web Service) ก็สามารถทำได้เช่นเดียวกัน

JSON นั้นใช้ความสัมพันธ์ของภาษาจาวาสคริปต์ แต่ไม่ถูกมองว่าเป็นภาษาโปรแกรม กลับถูกมองว่าเป็นภาษาในการแลกเปลี่ยนข้อมูลมากกว่า ในปัจจุบันมีไลบรารีของภาษาโปรแกรม อื่น ๆ ที่ใช้ประมวลผลข้อมูลในรูปแบบ JSON มากมาย

โค้ดตัวอย่างของ JSON เป็นดังนี้

```

{
  "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": 10021
  },
  "phoneNumbers": [
    "212 555-1234",
    "646 555-4567"
  ]
}

```

ภาพที่ 2.15 แสดงตัวอย่างข้อมูล JSON

2.5 NEST JS

2.5.1 บทนำ

Nest (NestJS) เป็นเฟรมเวิร์กสำหรับการสร้างแอปพลิเคชันฝั่งเซิร์ฟเวอร์ Node.js อย่างมีประสิทธิภาพ โดยใช้ Progressive JavaScript และรองรับ TypeScript อย่างสมบูรณ์ (นักพัฒนาสามารถเขียนโค้ดด้วย Pure JavaScript ได้) และรวมองค์ประกอบของ OOP (การเขียนโปรแกรมเชิงวัตถุ), FP (การเขียนโปรแกรมเชิงฟังก์ชัน) และ FRP (Functional Reactive Programming)

Nest ใช้ประโยชน์จากเฟรมเวิร์ก HTTP Server ที่มีประสิทธิภาพเช่น Express (ค่าเริ่มต้น) และสามารถกำหนดค่าให้ใช้ Fastify ได้เช่นกัน!

Nest ให้ระดับความเป็นนามธรรมเหนือกรอบ Node.js ทั่วไป (Express / Fastify) เหล่านี้ แต่ยังแสดง APIs โดยตรงไปยังนักพัฒนา สิ่งนี้ช่วยให้นักพัฒนามีอิสระในการใช้โมดูลมากมายจาก Third Party Module ซึ่งมีให้สำหรับแพลตฟอร์มพื้นฐาน

2.5.2 ปรัชญา

ในช่วงไม่กี่ปีที่ผ่านมาต้องขอบคุณ Node.js ทำให้ JavaScript กลายเป็น“ lingua franca” ของเว็บสำหรับแอปพลิเคชันทั้งด้านหน้าและด้านหลัง สิ่งนี้ทำให้โครงการที่ยอดเยี่ยมเช่น Angular, React และ Vue ซึ่งปรับปรุงประสิทธิภาพการทำงานของนักพัฒนา และเปิดใช้งานแอปพลิเคชันส่วนหน้าที่รวดเร็วทดสอบได้ และขยายได้ อย่างไรก็ตามในขณะที่มีห้องสมุดผู้ช่วยและเครื่องมือมากมายสำหรับ Node (และ JavaScript ฝั่งเซิร์ฟเวอร์) แต่ก็ไม่มีใครแก้ปัญหาหลักของสถาปัตยกรรมได้อย่างมีประสิทธิภาพ

Nest นำเสนอสถาปัตยกรรมแอปพลิเคชันที่ไม่เหมือนใครซึ่งช่วยให้นักพัฒนาและทีมงานสามารถสร้างแอปพลิเคชันที่ปรับขนาดได้ทดสอบได้คู่กันอย่างยืดหยุ่น และบำรุงรักษาได้ง่าย สถาปัตยกรรมได้รับแรงบันดาลใจอย่างมากจาก Angular

2.6 Business Intelligence

ความชาญฉลาดทางธุรกิจ Business Intelligence (BI) สิ่งนี้เกิดขึ้นเมื่อนานมาแล้ว ซึ่งคอนเซ็ปต์ของมันคือการจัดการกับข้อมูลที่องค์กรหรือธุรกิจนั้น ๆ ถือครองอยู่ เพื่อให้การทำงานในส่วนต่างๆ ของระบบธุรกิจเป็นไปได้ง่ายขึ้น และช่วยลดข้อผิดพลาดในการทำงานลงอีกด้วย ทว่าปัจจุบันนี้มันสามารถทำได้มากกว่านั้น Modern BI สามารถช่วยให้การวิเคราะห์ข้อมูลทำได้รวดเร็วขึ้น นำเสนอเป็น Visualization ที่สวยงาม และเข้าใจได้ง่ายกว่าเมื่อก่อน มีระบบ Interactive ต่าง ๆ ที่ทำให้ดูข้อมูลได้ดีขึ้นไปจนถึงการตัดสินใจทางธุรกิจที่ยอดเยี่ยมจากการ predict ชุดข้อมูลต่าง ๆ ที่มี ซึ่งที่กล่าวมา มันคือ software ที่ทำหน้าที่ให้นำข้อมูลไปจัดทำรายงานในรูปแบบต่าง ๆ ให้เหมาะสมกับมุมมองในการวิเคราะห์ แสดงความสัมพันธ์ และทำนายผลลัพธ์ของแนวโน้มที่อาจเกิดขึ้นได้ตรงตามความต้องการขององค์กร เพื่อประโยชน์ในการวางแผนกลยุทธ์นั่นเอง

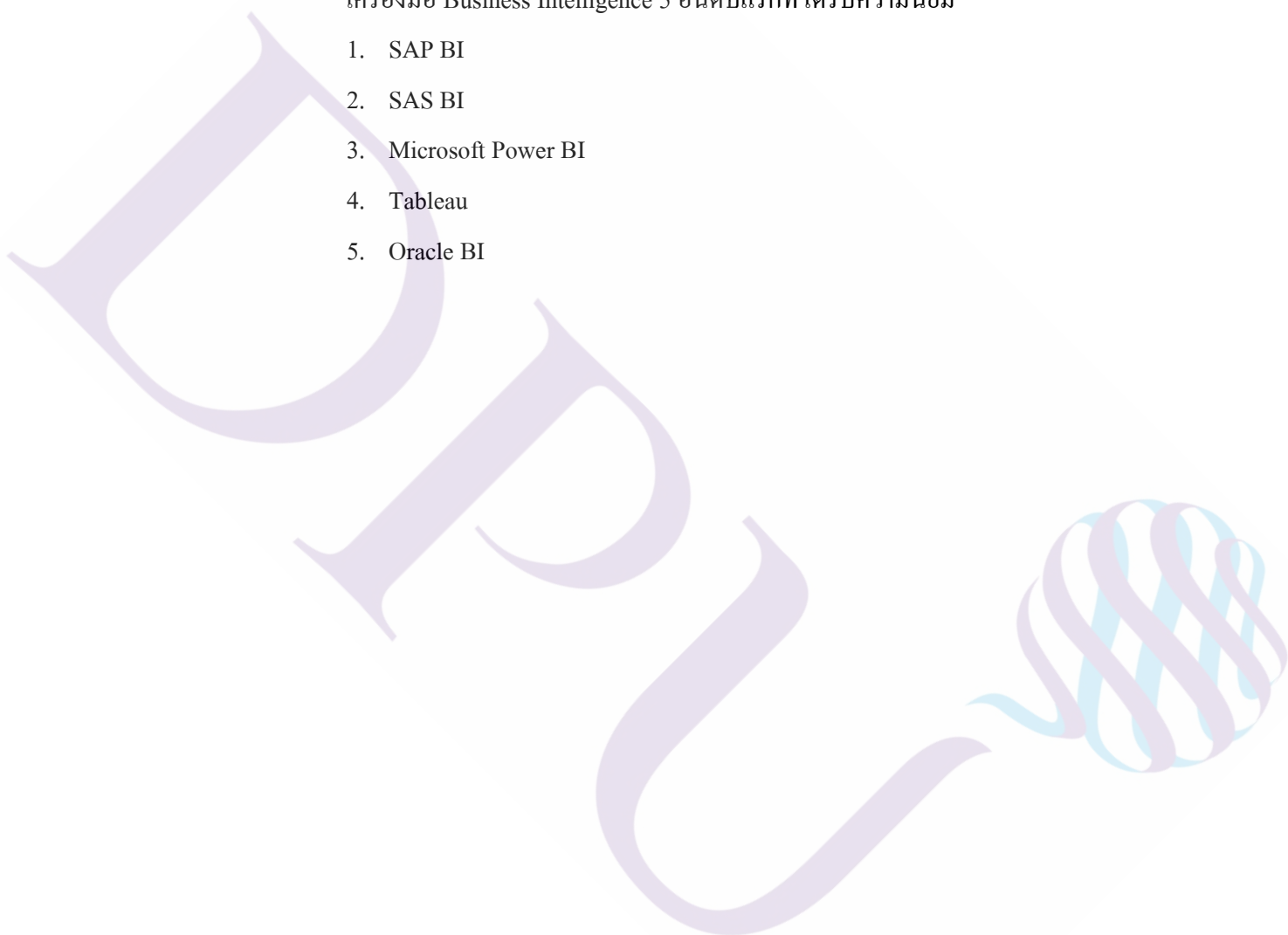
โปรแกรม BI เกิดขึ้นจากผสมผสาน business analytics, data mining, data visualization, data tools and infrastructure เข้าด้วยกัน และนำโมเดลที่ได้ไปฝึกจนสามารถคาดการณ์ผลลัพธ์ได้อย่างแม่นยำเพื่อให้องค์กรสามารถตัดสินใจได้อย่างดียิ่งขึ้น องค์ประกอบมีดังนี้

1. Data Warehouse คือ ฐานข้อมูลขนาดใหญ่ที่รวบรวมข้อมูลทั้งจากแหล่งข้อมูลภายในและภายนอกองค์กร รูปแบบข้อมูลที่ได้มาจำเป็นต้องทำ data preparation เพื่อให้เป็นไปตามเป้าหมายขององค์กร
2. Data Mart คือ คลังข้อมูลขนาดเล็กมีการเก็บข้อมูลที่มีลักษณะเฉพาะเจาะจง เช่น เก็บข้อมูลส่วนของการเงิน ส่วนของสินค้าคงคลัง ส่วนของการขาย เป็นต้น ซึ่งทำให้การวิเคราะห์ หาค่าความสัมพันธ์ และจัดการข้อมูลทำได้เหมาะสมยิ่งขึ้น
3. Data Mining คือ การนำคลังข้อมูลหลักมาประมวลผลใหม่ แสดงเฉพาะสิ่งที่สนใจโดยกระบวนการในการดึงข้อมูลออกจากฐานข้อมูลจะมีสูตรทางธุรกิจ Business Formula และเงื่อนไขต่าง ๆ เข้ามาเกี่ยวข้อง ผลลัพธ์ในรูปแบบที่แตกต่างกัน เช่น เป็นแผนภูมิในการตัดสินใจ Decision Trees เป็นต้น
4. Operations Research & Numerical Methods คือ การแก้ปัญหาทางคณิตศาสตร์
5. On-Line Analytical Processing – OLAP เครื่องมือที่ใช้ในการวิเคราะห์ข้อมูลในหลายมิติ แบบประมวลผลทันทีที่ป้อนข้อมูลเข้าไป คือ การสืบค้นข้อมูลที่ผู้ใช้สามารถเลือกผลลัพธ์ออกมาในรูปแบบของตารางหรือกราฟ อันสามารถวิเคราะห์ข้อมูลในมุมมองหลากหลายมิติ Multi-Dimensional โดยที่ผู้ใช้สามารถที่จะดูข้อมูลแบบเชิงลึกได้ตามต้องการ
6. Search & Report คือ ระบบสืบค้น และการแสดงผลการวิเคราะห์ให้เข้าใจ

ธุรกิจหรือองค์กรต่าง ๆ นั้นมีทั้งปัญหา และเป้าหมายที่ต้องการหาวิธีการในการแก้ และคำตอบมากมาย เพื่อที่จะตอบคำถามเหล่านี้ จึงได้ทำการรวบรวมข้อมูลที่เกี่ยวข้องจากทุกแหล่งเท่าที่เป็นไปได้ และกำหนดวิธีการที่จะนำไปสู่เป้าหมาย ในด้านเทคนิคข้อมูลดิบจะถูกรวบรวมจากกิจกรรมของธุรกิจ ประมวลผลข้อมูลแล้วจัดเก็บในคลังข้อมูล เมื่อจัดเก็บแล้วผู้ใช้จะสามารถเข้าถึงข้อมูลเริ่มกระบวนการวิเคราะห์เพื่อตอบคำถามทางธุรกิจ การทำงานเป็นไปตามแผนภาพนี้

เครื่องมือ Business Intelligence 5 อันดับแรกที่ได้รับความนิยม

1. SAP BI
2. SAS BI
3. Microsoft Power BI
4. Tableau
5. Oracle BI



บทที่ 3

วิธีการดำเนินการ และเครื่องมือ

ในบทนี้จะกล่าวถึงกระบวนการพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย ด้วยเทคโนโลยีการสื่อสารผ่าน REST API และรูปแบบข้อมูล JSON รวมถึงการนำเครื่องมือในการจัดทำเอกสาร Open API Specification มาประยุกต์ใช้เพื่อให้เกิดประโยชน์ในการพัฒนาระบบ โดยมีวิธีดำเนินงานและเครื่องมือ ดังนี้

3.1 ศึกษาปัญหาและความต้องการของระบบ

3.1.1 วิเคราะห์ความต้องการของผู้ใช้งาน

จากการเก็บบันทึกความต้องการของผู้ใช้ระบบบริหารงานขายครบวงจร “iSmartSales” ตั้งแต่ปี พ.ศ. 2559 โดยผู้ให้ข้อมูลคือผู้ใช้งานในส่วนต่างๆของระบบ ได้แก่ ตัวแทนขาย หัวหน้าฝ่ายขาย พนักงานสนับสนุนงานขาย พนักงานบัญชี รวมถึงผู้บริหารระดับสูง สรุปความต้องการข้อมูลสำหรับวิเคราะห์งานขาย ดังนี้

3.1.1.1 ข้อมูลวิเคราะห์ภาพรวมธุรกิจ

1. การตรวจสอบ Key Performance Indicator (KPI)
2. การตรวจสอบจำนวนลูกค้าที่มีการซื้อขาย
3. การตรวจสอบจำนวนการเข้าเยี่ยมลูกค้า
4. การติดตามยอดขายสะสมรายปี
5. การติดตามยอดขายสะสมรายไตรมาส
6. การติดตามยอดขายสะสมรายเดือน
7. การติดตามปริมาณการสั่งซื้อ
8. การติดตามยอดค้างส่งสินค้า
9. การติดตามยอดคืนสินค้า
10. การวิเคราะห์ยอดขายเปรียบเทียบช่วงเวลาเดียวกันของปีก่อน
11. การเปรียบเทียบการสั่งซื้อสินค้า และการส่งสินค้า
12. การเปรียบเทียบการส่งสินค้า และการคืนสินค้า

13. การเปรียบเทียบการสั่งซื้อสินค้าและคำสั่งสินค้า
14. การเปรียบเทียบสัดส่วนการขายตามลูกค้า
15. การเปรียบเทียบสัดส่วนการขายตามสินค้า
16. การเปรียบเทียบสัดส่วนการขายตามทีมขาย
17. การติดตามยอดสะสมมูลค่าการขายสินค้าเทียบการคืนสินค้า
18. การติดตามการขายรายจังหวัด
19. การสื่อสารภายในองค์กรผ่านกิจกรรมการทำงาน

3.1.1.2 ข้อมูลวิเคราะห์ภาพรวมลูกค้า

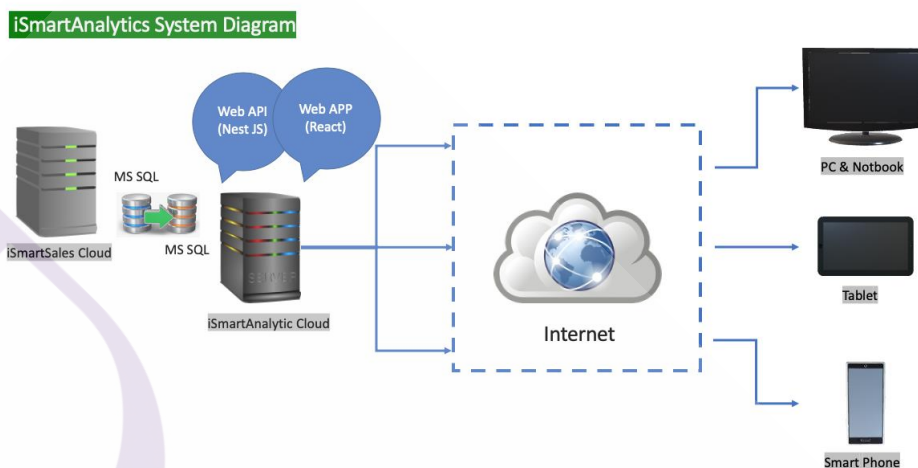
1. การติดตามโอกาสการขายใหม่จากตัวเลข
 - 1.1 ลูกค้าผู้มุ่งหวัง (Prospect Customer)
 - 1.2 ลูกค้าใหม่ (New Customer)
 - 1.3 ลูกค้าเก่ากลับมาซื้อใหม่ (ReEntry)
2. การติดตามความเคลื่อนไหวจำนวนลูกค้าที่ซื้อและไม่ซื้อ
3. การเปรียบเทียบสัดส่วนการขายแต่ละกลุ่มลูกค้า
4. การตรวจสอบ ติดตามลูกหนี้ค้างชำระ

3.1.1.3 ข้อมูลวิเคราะห์รายลูกค้า

1. รายละเอียดทะเบียนลูกค้า และการติดต่อ
2. อันดับการซื้อสินค้า
3. สถานะบิลค้างชำระ
4. การติดตามขอระบายสินค้าถึงผู้บริโภค
5. ประวัติการซื้อสินค้า
6. การเปรียบเทียบยอดสั่งซื้อสินค้ากับยอดคืนสินค้า
7. สินค้าที่มีการซื้อ และยังไม่ซื้อ
8. การติดตามรอบการสั่งซื้อให้เป็นไปตามรอบปกติ
9. การติดตามกิจกรรมในการสร้างความสัมพันธ์

3.2 วิเคราะห์ และออกแบบระบบ

3.2.1 สถาปัตยกรรมของระบบ (System Architecture)



ภาพที่ 3.1 ภาพรวมสถาปัตยกรรมของระบบเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย

ภาพรวมสถาปัตยกรรมของระบบเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย มีองค์ประกอบดังนี้

3.2.1.1 iSmartSales Cloud เป็น Web Server ระบบบริหารงานขายครบวงจร

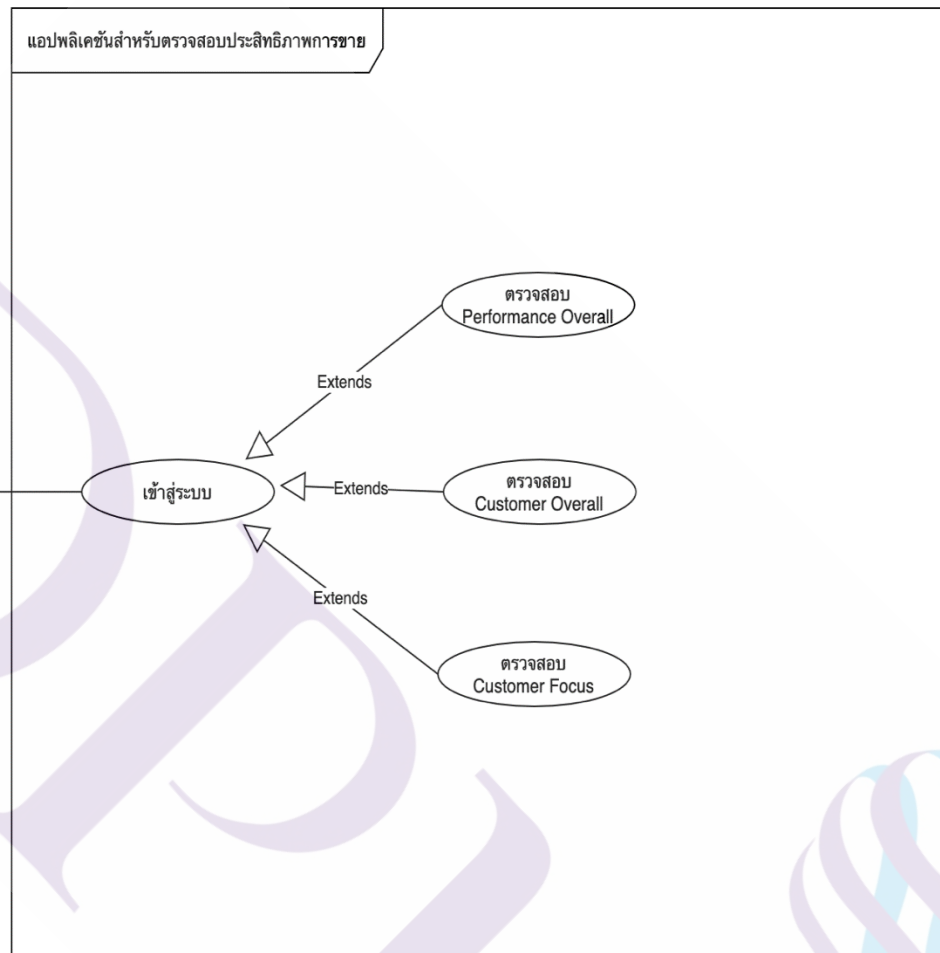
1. ระบบฐานข้อมูล Microsoft SQL Server

3.2.1.2 SPM Cloud เป็น Web Server ระบบการตรวจสอบประสิทธิภาพการขาย ซึ่งประกอบด้วย

1. ระบบฐานข้อมูล Microsoft SQL Server
2. ระบบ Web API พัฒนาด้วย Framework NEST JS
3. ระบบ Web Application พัฒนาด้วย React สำหรับผู้ใช้งานอุปกรณ์คอมพิวเตอร์

และอุปกรณ์พกพาต่าง ๆ

3.2.2 การออกแบบตาม Use Case การใช้งานของผู้ใช้



ภาพที่ 3.2 Use Case Diagram สำหรับการตรวจสอบประสิทธิภาพการขาย

ตารางที่ 3.1 สรุป Use Case ของระบบการตรวจสอบประสิทธิภาพการขาย

N o.	Use Case ID	Use Case Name	Actor	ตาราง
1	US0100	ผู้ใช้เข้าสู่ระบบ	ผู้ใช้	3.2
2	PO0100	ผู้ใช้ตรวจสอบ Performance Overall	ผู้ใช้	3.3
3	PO0200	ผู้ใช้ตรวจสอบ Customer Overall	ผู้ใช้	3.4
4	PO0300	ผู้ใช้ตรวจสอบ Customer Focus	ผู้ใช้	3.5

ตารางที่ 3.2 Use Case ผู้ใช้เข้าสู่ระบบ

Use Case ID	US0100	
Use Case Name	ผู้ใช้เข้าสู่ระบบ	
Actor	ผู้ใช้งาน	
Scenarios	1. ผู้ใช้ Login เข้าใช้งานระบบ	
Preconditions		
Postconditions	1. ระบบแสดงรายละเอียดผู้ใช้งาน	
Basic Flows	Actor	System
	1. ผู้ใช้ใส่รหัสผู้ใช้งาน 2. ผู้ใช้ใส่รหัสผ่าน	1. ระบบตรวจสอบความถูกต้องของรหัสผู้ใช้งานและรหัสผ่าน 2. ระบบตรวจสอบสิทธิ์ในการเข้าถึงข้อมูล

ตารางที่ 3.3 Use Case ผู้ใช้ตรวจสอบ Performance Overall

Use Case ID	PO0100	
Use Case Name	ผู้ใช้ตรวจสอบข้อมูล Performance Overall	
Actor	ผู้ใช้งาน	
Scenarios	1. ผู้ใช้เลือกเมนู Performance Overall	
Preconditions	1. ผ่านการทำงานตาม Use Case US0100 (ผู้ใช้เข้าสู่ระบบ)	
Postconditions	<ol style="list-style-type: none"> 1. ระบบแสดงข้อมูล KPI 2. ระบบแสดงจำนวนลูกค้าที่มีการซื้อขาย 3. ระบบแสดงจำนวนการเข้าเยี่ยมลูกค้า 4. ระบบแสดงยอดขายสะสมรายปี 5. ระบบแสดงยอดขายสะสมรายไตรมาส 6. ระบบแสดงยอดขายสะสมรายเดือน 7. ระบบแสดงปริมาณการสั่งซื้อ 8. ระบบแสดงยอดค้างส่งสินค้า 9. ระบบแสดงยอดคืนสินค้า 10. ระบบแสดงยอดขายเปรียบเทียบช่วงเวลาเดียวกันของปีก่อน 11. ระบบแสดงการสั่งซื้อสินค้าและการส่งสินค้า 12. ระบบแสดงการเปรียบเทียบการส่งสินค้าและการคืนสินค้า 13. ระบบแสดงเปรียบเทียบการสั่งซื้อสินค้าและค้างส่งสินค้า 14. ระบบแสดงเปรียบเทียบสัดส่วนการขายตามลูกค้า 15. ระบบแสดงเปรียบเทียบสัดส่วนการขายตามสินค้า 16. ระบบแสดงเปรียบเทียบสัดส่วนการขายตามทีมขาย 17. ระบบแสดงยอดสะสมมูลค่าการขายสินค้าเทียบการคืนสินค้า 18. ระบบแสดงการขายรายจังหวัด 19. ระบบแสดงกิจกรรมการทำ 	
Basic Flows	Actor	System
	1. ผู้ใช้เลือกเมนู Performance Overall	1. ระบบโหลดข้อมูลตามสิทธิ์ของผู้ใช้งาน ตาม Dashboard 19 รูปแบบ

ตารางที่ 3.4 Use Case ผู้ใช้ตรวจสอบ Customer Overall

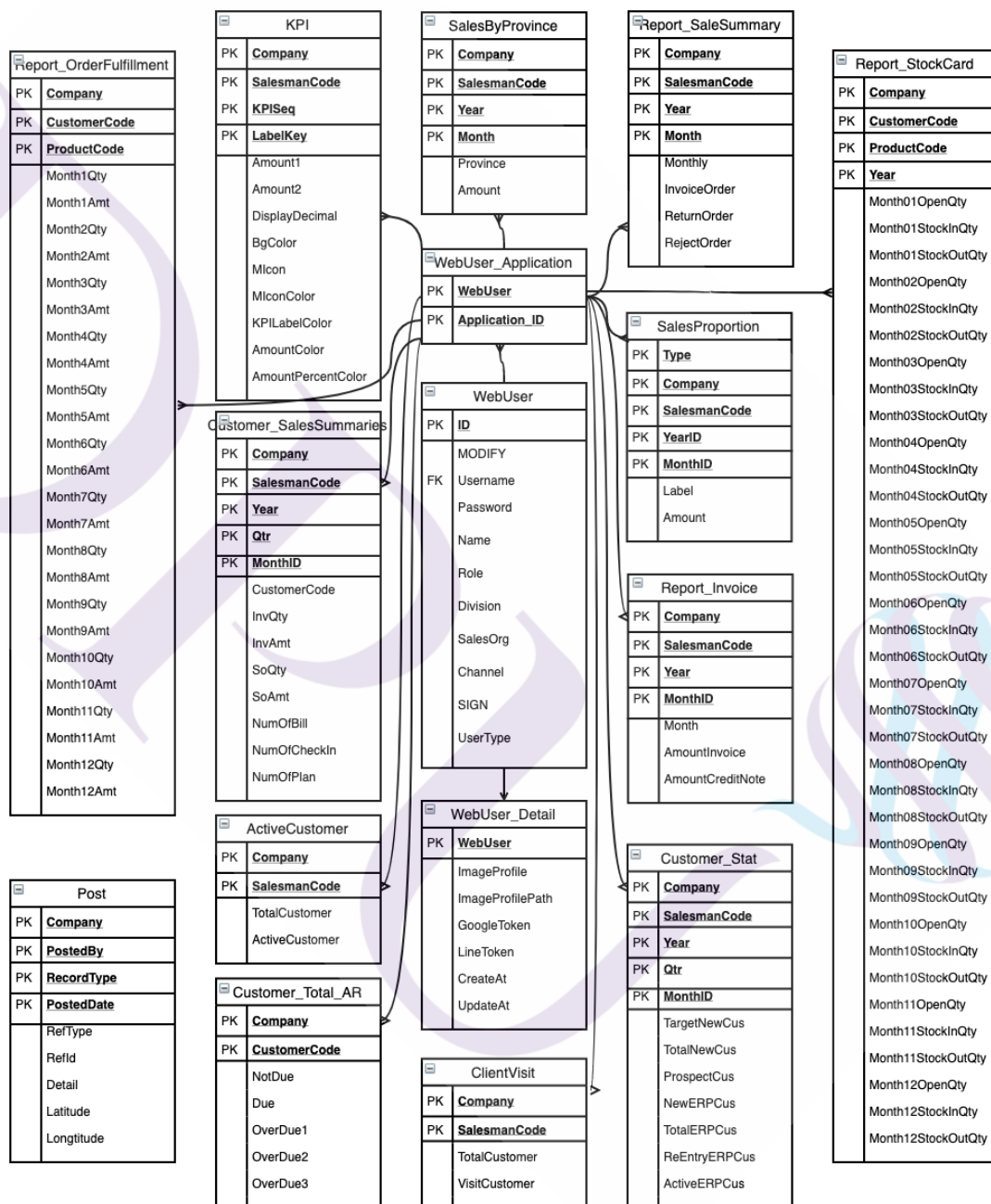
Use Case ID	PO0200	
Use Case Name	ผู้ใช้ตรวจสอบ Customer Overall	
Actor	ผู้ใช้งาน	
Scenarios	1. ผู้ใช้เลือกเมนู Customer Overall	
Preconditions	1. ผ่านการทำงานตาม Use Case US0100 (ผู้ใช้เข้าสู่ระบบ)	
Postconditions	<ol style="list-style-type: none"> 1. ระบบแสดงเปรียบเทียบลูกค้าใหม่ 2. ระบบแสดงความเคลื่อนไหวลูกค้าทุกสถานะ 3. ระบบแสดงเปรียบเทียบจำนวนลูกค้าซื้อ 4. ระบบแสดงความเคลื่อนไหวลูกค้าซื้อ/ลูกค้าใหม่ 5. ระบบแสดงความเคลื่อนไหวลูกค้าซื้อ 6. ระบบแสดงยอดขายตามกลุ่มลูกค้า 7. ระบบแสดงข้อมูลที่ซื้อรายลูกค้า 8. ระบบแสดงสถานะลูกหนี้ 	
Basic Flows	Actor	System
	1. ผู้ใช้เลือกเมนู Customer Overall	1. ระบบโหลดข้อมูลตามสิทธิ์ของผู้ใช้งาน ตาม Dashboard 8 รูปแบบ

ตารางที่ 3.5 Use Case ผู้ใช้ตรวจสอบ Customer Focus

Use Case ID	PO0300	
Use Case Name	ผู้ใช้ตรวจสอบ Customer Focus	
Actor	ผู้ใช้งาน	
Scenarios	1. ผู้ใช้เลือกเมนู Customer Overall	
Preconditions	1. ผ่านการทำงานตาม Use Case US0100 (ผู้ใช้เข้าสู่ระบบ)	
Postconditions	<ol style="list-style-type: none"> 1. ระบบแสดงข้อมูลหน้าบัญชี 2. ระบบแสดงยอดซื้อสะสม/ยอดส่งสินค้ารายปี 3. ระบบแสดงยอดซื้อสะสม/ยอดส่งสินค้ารายเดือน 4. ระบบแสดงรายละเอียดบิลคำสั่งชำระ 5. ระบบแสดงยอดระบายในตลาด 6. ระบบแสดงประวัติการขาย 7. ระบบแสดงปริมาณการขาย/คืน 8. ระบบแสดงเปรียบเทียบสัดส่วนการขาย 9. ระบบแสดงสินค้าที่ยังไม่มีการขาย 10. ระบบแสดงกิจกรรมที่ทำกับลูกค้า 	
Basic Flows	Actor	System
	1. ผู้ใช้เลือกเมนู Customer Focus	1. ระบบโหลดข้อมูลตามสิทธิ์ของผู้ใช้งาน ตาม Dashboard 10 รูปแบบ

3.2.3 การออกแบบฐานข้อมูล

จากการศึกษาข้อมูลความต้องการของระบบ ได้วิเคราะห์และออกแบบ ฐานข้อมูล ระบบการตรวจสอบประสิทธิภาพการขาย ดังแสดงได้ตาม E-R Diagram



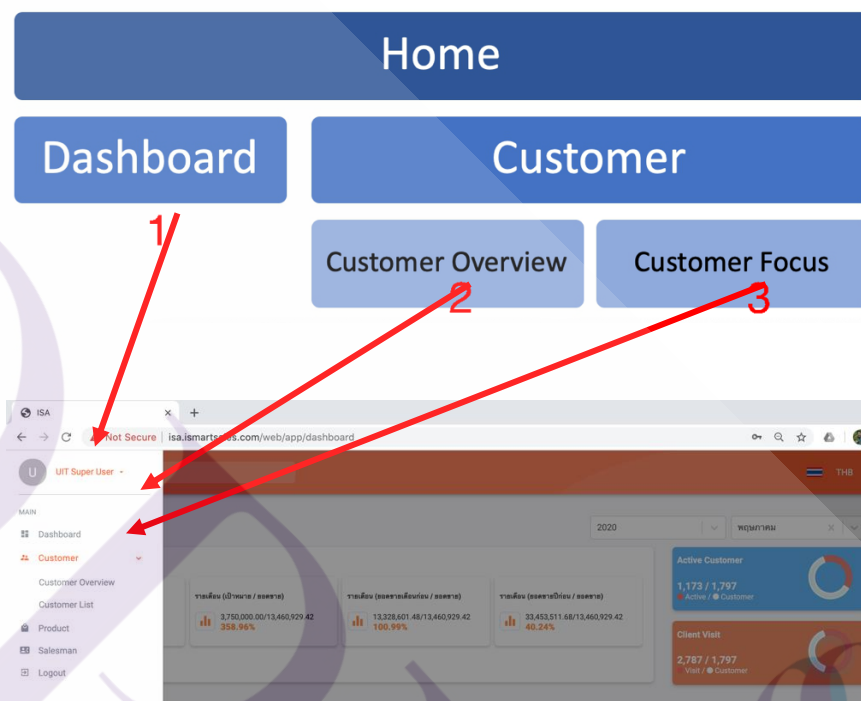
ภาพที่ 3.3 ER- Diagram ระบบการตรวจสอบประสิทธิภาพการขาย

ตารางที่ 3.6 แสดงรายละเอียดตารางข้อมูลระบบการตรวจสอบประสิทธิภาพการขาย

No.	Table	Table Description
1	WebUser	ตารางเก็บข้อมูลผู้ใช้งาน
2	WebUser_Detail	ตารางเก็บรูปภาพ และสิทธิ์ในการเข้าสู่ระบบ
3	WebUser_Application	ตารางแสดงสิทธิ์ในการเข้าถึงข้อมูลของผู้ใช้
4	SalesByProvince	ตารางเก็บข้อมูลยอดขายตามพื้นที่การขาย
5	ClientVisit	ตารางเก็บข้อมูล Check In ของพนักงานขาย
6	KPI	ตารางแสดง KPI
7	Customer_SalesSummaries	ตารางแสดงคำสั่งซื้อ เทียบยอดขาย เทียบกิจกรรม
8	ActiveCustomer	ตารางแสดงจำนวนลูกค้าที่มีการสั่งซื้อ
9	Customer_Total_AR	ตารางแสดงจำนวนบิลค้างชำระ
10	Report_OrderFulfillment	ตารางแสดงสถิติการสั่งซื้อ
11	Post	ตารางแสดงกิจกรรมการทำงาน
12	Report_SaleSummary	ตารางแสดงยอดขายเทียบยอดคืนและยอดค้างส่ง
13	SalesProportion	ตารางแสดงอัตราส่วนยอดขาย
14	Report_Invoice	ตารางแสดงยอดขายเทียบยอดคืน ตามพนักงานขาย
15	Customer_Stat	ตารางแสดงสถานะลูกค้า
16	Report_StockCard	ตารางแสดงสถิติยอดระบายสินค้า

3.2.4 แผนผังแอปพลิเคชันที่ทำงานในระบบ

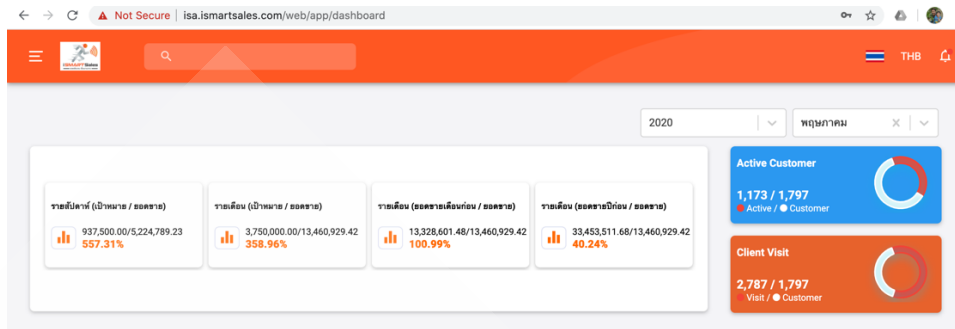
จากการศึกษาปัญหา และความต้องการของระบบ การออกแบบโครงสร้างระบบทำโดยพิจารณาจากสภาพการใช้งานระบบจริง ซึ่งแบ่งตามมุมมองในการวิเคราะห์ข้อมูลได้ดังนี้



ภาพที่ 3.4 แผนผังระบบการตรวจสอบประสิทธิภาพการขาย

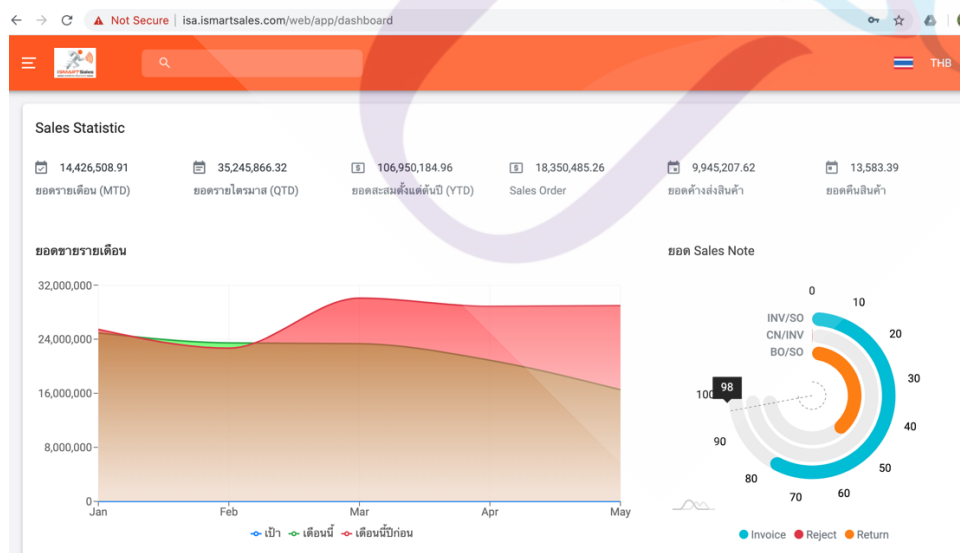
3.2.4.1 วิเคราะห์ภาพรวมธุรกิจ (Dashboard) นำเสนอข้อมูล ดังนี้

1. การตรวจสอบ Key Performance Indicator (KPI)
2. การตรวจสอบจำนวนลูกค้าที่มีการซื้อขาย
3. การตรวจสอบจำนวนการเข้าเยี่ยมชมลูกค้า



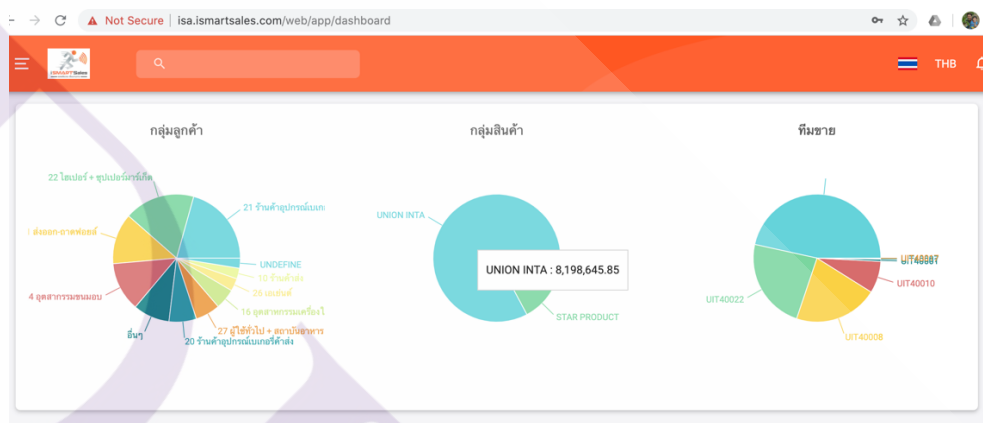
ภาพที่ 3.5 วิเคราะห์ภาพรวมธุรกิจ (Dashboard) นำเสนอข้อมูล

4. การติดตามยอดขายสะสมรายปี
5. การติดตามยอดขายสะสมรายไตรมาส
6. การติดตามยอดขายสะสมรายเดือน
7. การติดตามปริมาณการสั่งซื้อ
8. การติดตามยอดค้างส่งสินค้า
9. การติดตามยอดคืนสินค้า
10. การวิเคราะห์ยอดขายเปรียบเทียบช่วงเวลาเดียวกันของปีก่อน
11. การเปรียบเทียบการสั่งซื้อสินค้า และการส่งสินค้า
12. การเปรียบเทียบการส่งสินค้า และการคืนสินค้า
13. การเปรียบเทียบการสั่งซื้อสินค้า และค้างส่งสินค้า



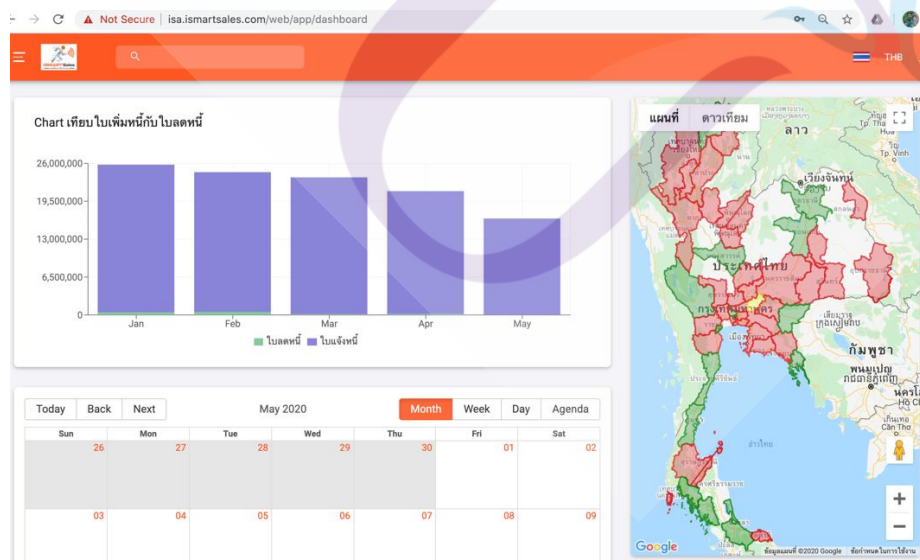
ภาพที่ 3.5 (ต่อ)

14. การเปรียบเทียบสัดส่วนการขายตามลูกค้า
15. การเปรียบเทียบสัดส่วนการขายตามสินค้า
16. การเปรียบเทียบสัดส่วนการขายตามทีมขาย



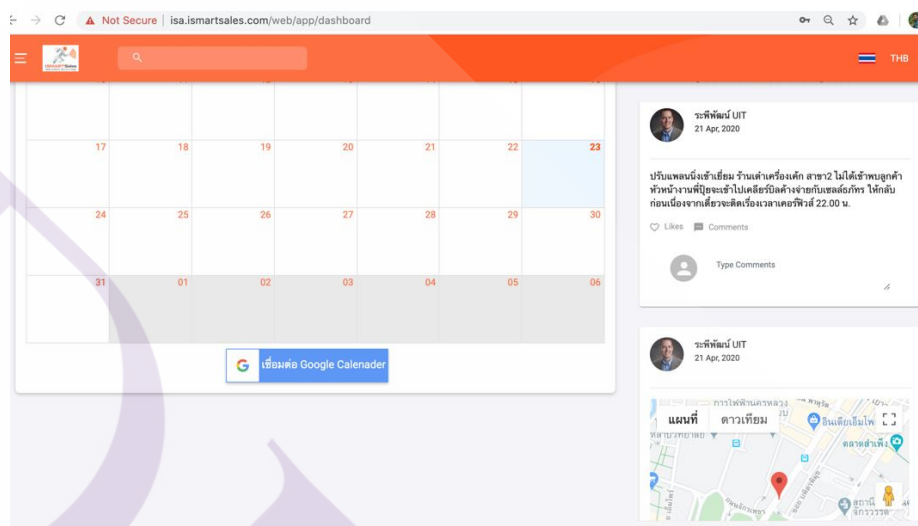
ภาพที่ 3.5 (ต่อ)

17. การติดตามยอดสะสมมูลค่าการขายสินค้าเทียบการคืนสินค้า
18. การติดตามการขายรายจังหวัด



ภาพที่ 3.5 (ต่อ)

19. การสื่อสารภายในองค์กรผ่านกิจกรรมการทำงาน



ภาพที่ 3.5 (ต่อ)

3.2.4.2 วิเคราะห์เจาะลึกมุมมองลูกค้า (Customer Focus)

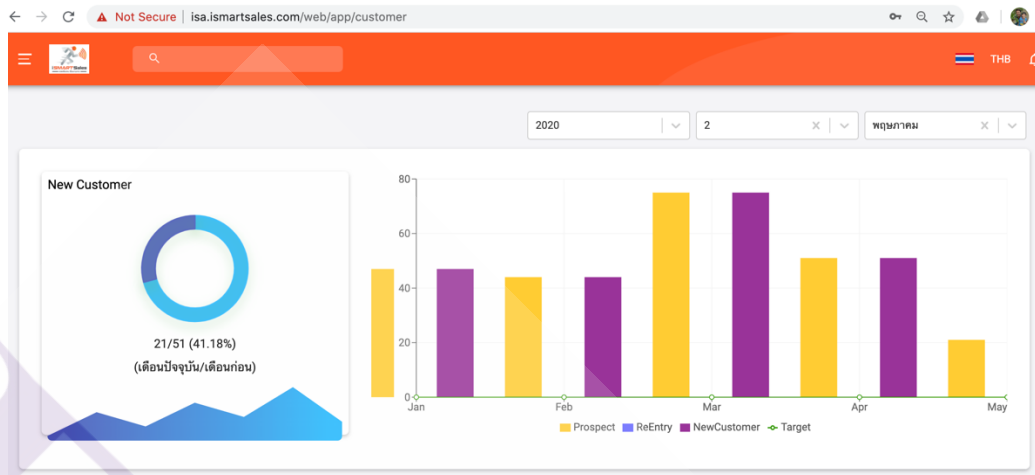
1. วิเคราะห์ลูกค้าทั้งหมด (Customer Overview)

1.1 การติดตามโอกาสการขายใหม่จากตัวเลข

1.1.1 ลูกค้าผู้มุ่งหวัง (Prospect Customer)

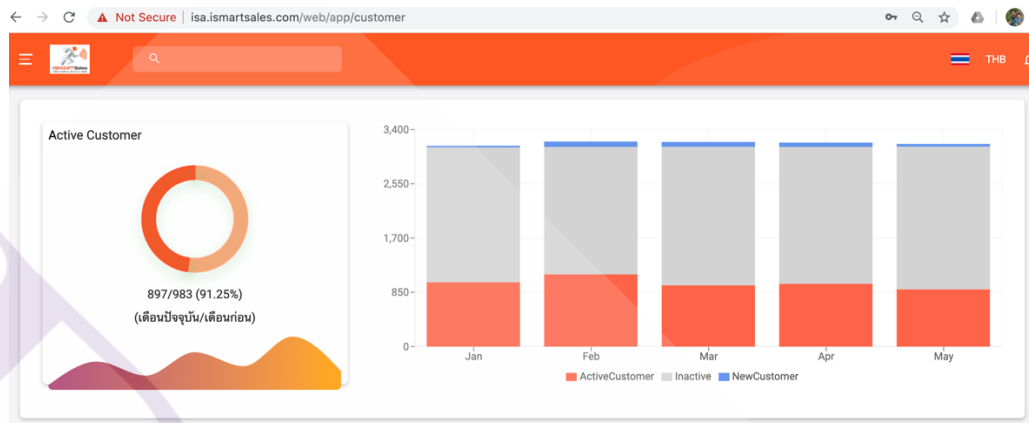
1.1.2 ลูกค้าใหม่ (New Customer)

1.1.3 ลูกค้าเก่ากลับมาซื้อใหม่ (ReEntry)



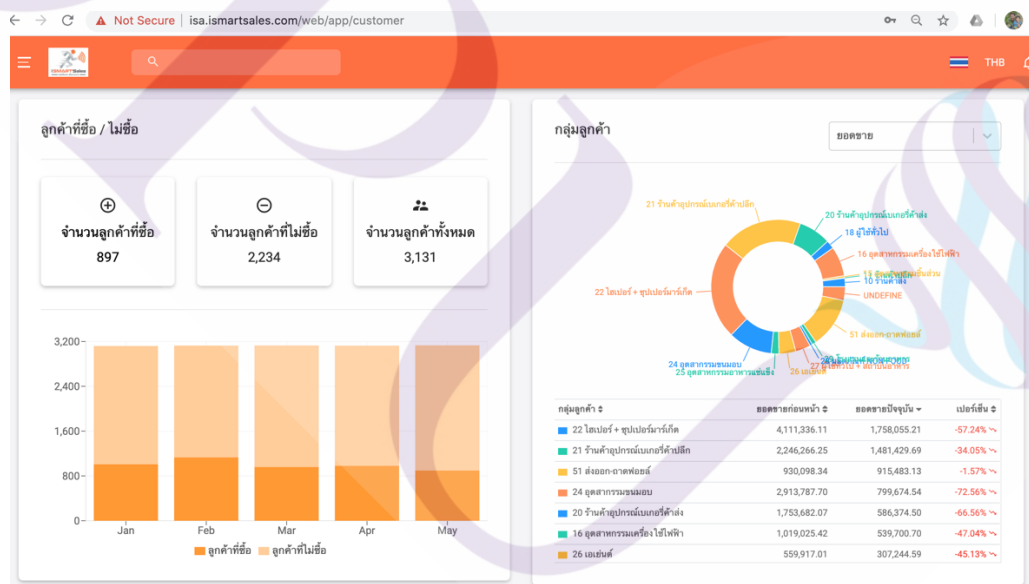
ภาพที่ 3.6 วิเคราะห์ลูกค้าทั้งหมด (Customer Overview)

1.2 การติดตามความเคลื่อนไหวจำนวนลูกค้าที่ซื้อ และไม่ซื้อ



ภาพที่ 3.6 (ต่อ)

1.3 การเปรียบเทียบสัดส่วนการขายแต่ละกลุ่มลูกค้า



ภาพที่ 3.6 (ต่อ)

1.4 การตรวจสอบ ติดตามลูกหนี้ค้างชำระ

ข้อมูลอายุลูกหนี้ 90 วัน

อายุลูกหนี้ 559 ราย / 34,222,489.95 บาท

รหัสลูกค้า	ชื่อลูกค้า	OverDue	Due	NotDue	รวม
UITA006	ANDRE ENTERPRISE LTD	358,247.97	0.00	0.00	358,247.97
UITA1	ลูกค้าใหม่ - กาศพอสส์	2,223.21	816.00	0.00	3,039.21
UITM004	MORECC CO., LTD.	445,948.79	0.00	0.00	445,948.79
UITS010	Siam Cineplex Co.,Ltd. สาขาสยามพารากอน	28,799.85	0.00	0.00	28,799.85
UITn012	เกียรติพาณิชย์	1,637.50	3,274.99	0.00	4,912.49
UITn017	การบินไทย จำกัด (มหาชน)	1,356,554.00	82,010.00	92,121.50	1,530,685.50
UITn045	เอกชัย(โตแดง)	120.00	0.00	6,086.00	6,206.00
UITn130	กรุงเทพ-โตคิวสรรพสินค้า จำกัด	927.39	0.00	0.00	927.39
UITn165	กิตติมา เหน่าประเสริฐ	5,250.00	0.00	0.00	5,250.00
UITn181	กะตะธานี จำกัด	19,319.92	0.00	0.00	19,319.92

ภาพที่ 3.6 (ต่อ)

2. วิเคราะห์รายลูกค้า (Customer Focus)

- 2.1 รายละเอียดทะเบียนลูกค้า และการติดต่อ
- 2.2 อันดับการซื้อสินค้า
- 2.3 สถานะบิลค้างชำระ

บางใหญ่ เบเกอรี่
 อันดับ : 133
 กลุ่ม : 20 ร้านค้าอุปกรณ์เบเกอรี่ห้าง
 สถานะ : Active
 ผลการเข้าเชื่อมโยง : ไม่มีแผนแต่ได้เข้าเชื่อมโยง

UIT40008 UIT40013

ยอดขายเฉลี่ยที่ทำได้การเชื่อมโยง : 8,908.00
 เข้าเชื่อมโยงกี่ครั้งถึงขายได้ : 1.00
 ยอดขายเฉลี่ยต่อบิล : 8,908.00

134,882.32/150,979.26 บาท
SO/IV
 Order This Year

41,098.00/8,908.00 บาท
SO/IV
 Order This Month

บิลค้างชำระ

ลำดับ	วันที่ปิดชำระ	วันที่กำหนดชำระ	เลขที่บิล	เลขที่ใบกำกับ	จำนวนเงินคงค้าง
1	08/11/2019	23/11/2019	VJP-IV/6211073	IV/6211073	4,800.00
2	08/12/2019	01/01/1900	STP-IVN6212254	IVN6212254	15,423.00
3	23/12/2019	07/01/2020	VJP-IV/6212326	IV/6212326	5,559.00
4	23/12/2019	01/01/1900	STP-IVN6212168	IVN6212168	8,908.00
5	28/12/2019	01/01/1900	STP-	IVN6212317	2,880.00

ABOUT

ที่อยู่
80/13 ตลาดบางใหญ่ อ.บางใหญ่

จังหวัด
จ.นนทบุรี

เบอร์โทร
02-926-3927

แผนที่ ดาวเทียม

ภาพที่ 3.7 วิเคราะห์รายลูกค้า (Customer Focus)

2.4 การติดตามขอระบายสินค้าถึงผู้บริโภค

สินค้า	มิถุนายน 2019			กรกฎาคม 2019			สิงหาคม 2019			กันยายน 2019			ตุลาคม 20	
	ออกมา	ซื้อ	ขาย	ออกมา	ซื้อ	ขาย	ออกมา	ซื้อ	ขาย	ออกมา	ซื้อ	ขาย	ออกมา	ซื้อ
กระดาษฟอยล์สตาร์โปรไฟล์ #1345 (1000/28/28000) แบบกระดาษ (สำหรับพิมพ์) 5217	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	28.00	28.00	0.00	0.00
กระดาษฟอยล์สตาร์โปรไฟล์ #1370 (1000/75/18000) แบบกระดาษ (สำหรับพิมพ์) 5220	0.00	0.00	0.00	0.00	0.00	0.00	0.00	18.00	18.00	0.00	0.00	0.00	0.00	0.00
กระดาษฟอยล์สตาร์โปรไฟล์ #1375 (1000/76/16000)แบบกระดาษ (สำหรับพิมพ์) 5338	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.00	16.00	0.00	0.00	0.00	0.00	0.00
โปรโมชัน A ซึ่งเป็นซีดี อุปกรณ์เงินสด 460 บาท S010/S160/S175/C365	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00
แผ่นสติกเกอร์ NO.001 ขนาด 35x50 ซม. สีขาว 5ชั้น/แผ่น 20จุดพิมพ์/กล่อง 4กล่อง/ลิ้ง	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.00	4.00	0.00	0.00
กระดาษฟอยล์สตาร์โปรไฟล์ #3105 (20/80/1600) พลาสติก	0.00	0.00	0.00	0.00	0.00	-70.00	70.00	0.00	70.00	0.00	0.00	0.00	0.00	0.00
กระดาษฟอยล์สตาร์โปรไฟล์ #3201 (20/100/2000) แบบห่อ	0.00	200.00	200.00	0.00	0.00	-100.00	100.00	300.00	400.00	0.00	200.00	200.00	0.00	0.00
กระดาษฟอยล์สตาร์โปรไฟล์ #3202 (20/75/1500) พลาสติก	0.00	0.00	0.00	0.00	0.00	-150.00	150.00	0.00	150.00	0.00	0.00	0.00	0.00	0.00
กระดาษฟอยล์สตาร์โปรไฟล์ #3203 (20/75/1500) พลาสติก	0.00	0.00	0.00	0.00	0.00	-95.00	95.00	0.00	95.00	0.00	0.00	0.00	0.00	0.00

ภาพที่ 3.7 (ต่อ)

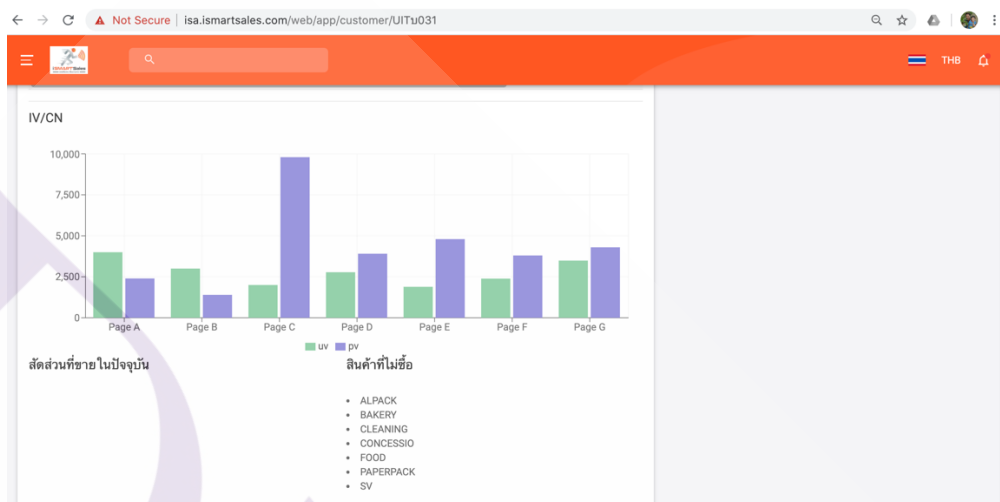
2.5 ประวัติการซื้อสินค้า

ProductLine	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
ALPACK	60,259.80	39,265.21	79,391.91	38,296.03	84,946.35	38,610.88	76,918.28	55,174.51	7,837.06
BAKERY	2,698.76	0.00	1,099.06	2,691.60	0.00	0.00	0.00	0.00	0.00
CLEANING	0.00	0.00	0.00	2,291.74	0.00	0.00	0.00	0.00	0.00
CONCESSIO	0.00	0.00	0.00	0.00	0.00	7,566.13	4,503.69	0.00	0.00
FOOD	3,861.91	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PAPERPACK	8,507.28	7,990.65	0.00	5,813.08	17,428.96	4,485.98	5,195.33	1,261.68	7,650.00
SV	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

ภาพที่ 3.7 (ต่อ)

2.6 การเปรียบเทียบยอดสั่งซื้อสินค้ากับยอดคืนสินค้า

2.7 สินค้าที่มีการซื้อ และยังไม่ซื้อ



ภาพที่ 3.7 (ต่อ)

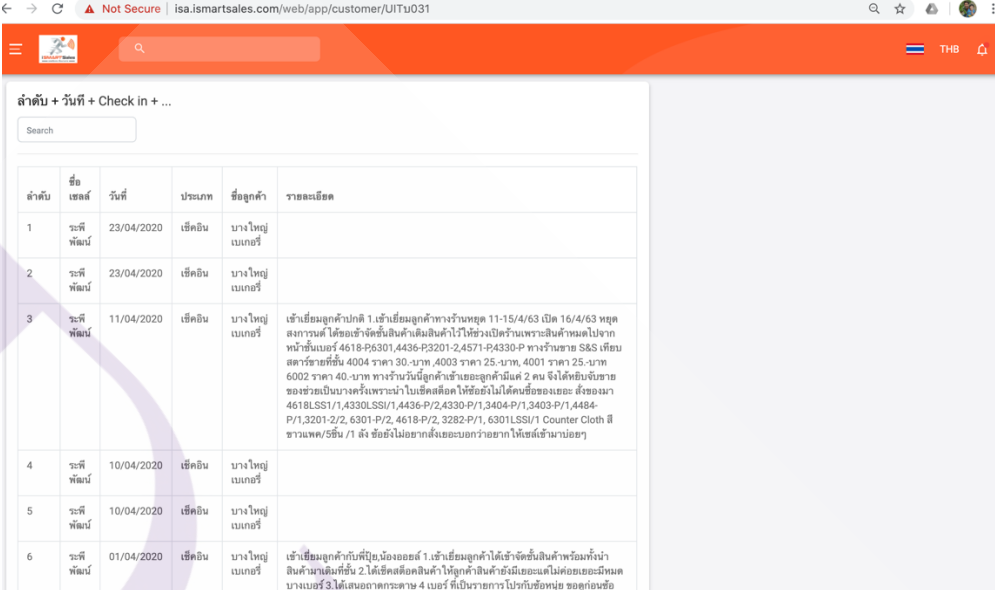
2.8 การติดตามรอบการสั่งซื้อให้เป็นไปตามรอบปกติ

รอบการสั่งซื้อสินค้า

ProductLine	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
ALPACK	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
BAKERY	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
CLEANING	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗
CONCESSION	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗
FOOD	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
PAPERPACK	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗
SV	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗

ภาพที่ 3.7 (ต่อ)

2.9 การติดตามกิจกรรมในการสร้างความสัมพันธ์



ลำดับ	ชื่อ	วันที่	ประเภท	ชื่อลูกค้า	รายละเอียด
1	ระพีพันธ์	23/04/2020	เช็คอิน	นางใหญ่ เบเกอร์	
2	ระพีพันธ์	23/04/2020	เช็คอิน	นางใหญ่ เบเกอร์	
3	ระพีพันธ์	11/04/2020	เช็คอิน	นางใหญ่ เบเกอร์	เข้าเยี่ยมชมลูกค้าปกติ 1 เข้าเยี่ยมชมลูกค้าทางบ้าน ชุด 11-15/4/63 เปิด 16/4/63 หยุดสงกรานต์ ได้ขอเข้าจัดรับเป็นค่าสัมปทานไปไว้ซึ่งมีปริมาณสินค้าหมดไปจากพื้นที่ชั้นเบอร์ 4618-P/6301, 4436-P/3201-2, 4571-P/4330-P ทางร้านขาย S&S เสื้อกันหนาวราคาชิ้น 4004 ราคา 30-บาท, 4003 ราคา 25-บาท, 4001 ราคา 25-บาท, 6002 ราคา 40-บาท ทางร้านวันที่ลูกค้าเข้ามาลูกค้ามีแค่ 2 คน จึงได้หยิบรายชื่อของช่างเป็นนางศรีเพราะนำไปเช็คสต็อกให้ชื่อยังไม่ได้นชื่อของชงเยะ ส่งของมา 4618LSSI/1, 4330LSSI/1, 4436-P/2, 4330-P/1, 3404-P/1, 3403-P/1, 4484-P/1, 3201-2/2, 6301-P/2, 4618-P/2, 3282-P/1, 6301LSSI/1 Counter Cloth สีขาวแพค/5ชิ้น /1 ลัง ชื่อยังไม่อยากส่งเลขยกกว่าอยากให้เช็คเข้ามาบ่อยๆ
4	ระพีพันธ์	10/04/2020	เช็คอิน	นางใหญ่ เบเกอร์	
5	ระพีพันธ์	10/04/2020	เช็คอิน	นางใหญ่ เบเกอร์	
6	ระพีพันธ์	01/04/2020	เช็คอิน	นางใหญ่ เบเกอร์	เข้าเยี่ยมชมลูกค้าที่ปฎิบัติงานของ 1 เข้าเยี่ยมชมลูกค้าได้เข้าจัดขึ้นสินค้าพร้อมทั้งนำสินค้ามาลงที่ชั้น 2 ได้เช็คสต็อกสินค้าให้ลูกค้าถึงค่าสัมปทานโดยชงเยะมีหมดบางเบอร์ 3 ได้เสนอราคาตาม 4 เบอร์ ที่เป็นรายการไปรับที่ชื่อทูลุ ซอตุ๋นชื่อ

ภาพที่ 3.7 (ต่อ)

3.2.5 การออกแบบ API

จากการศึกษาปัญหา และความต้องการของระบบ และจากการวิเคราะห์ ออกมาเป็น Use Case ทั้งหมด จึงสามารถออกแบบ API สำหรับเว็บเซอร์วิส ได้ดังนี้

ตารางที่ 3.7 Use Case (การออกแบบ API)

No.	API	Description
1	Customer	
1.1	GET /api/customer	เรียกดูข้อมูลลูกค้า
2	Active Customer	
2.1	GET /api/customer/activeCustomerChart	เรียกดูข้อมูลจำนวนลูกค้าที่มีการซื้อเทียบลูกค้าทั้งหมด
3	Visit Feed	
3.1	GET /api/customer/visitFeedChart	เรียกดูข้อมูลจำนวนลูกค้าที่มีการ Check In เทียบลูกค้าทั้งหมด
4	Active Customer Card	
4.1	GET /api/customer/ActiveCustomerCard	เรียกดูข้อมูลจำนวนลูกค้าที่มีการซื้อเดือนที่ต้องการเทียบเดือนก่อน
5	Active Customer Statistics	
5.1	GET /api/customer/ActiveCustomerStatChart	เรียกดูข้อมูลจำนวนลูกค้าที่มีการซื้อตั้งแต่ต้นปี
6	New Customer Card	
6.1	GET /api/customer/NewCustomerCard	/api/customer/NewCustomerCard

ตารางที่ 3.7 (ต่อ)

No.	API	Description
7	NewCustomerChart	
7.1	GET /api/customer/NewCustomerChart	เรียกดูข้อมูลจำนวนลูกค้าใหม่ตั้งแต่ต้นปี
8	Customer Product Card	
8.1	GET /api/customer/CustomerBuyProductCard	เรียกดูข้อมูลจำนวนลูกค้าซื้อ เทียบกับลูกค้าไม่ซื้อ และลูกค้าทั้งหมด ของเดือนปัจจุบัน
9	Customer Product Chart	
9.1	GET /api/customer/CustomerBuyProductChart	เรียกดูข้อมูลจำนวนลูกค้าซื้อ เทียบกับลูกค้าไม่ซื้อ และลูกค้าทั้งหมด ตั้งแต่ต้นปี
10	Login	
10.1	POST /api/auth/login	เช็คสิทธิ์การใช้งาน
11	Sales Summary	
11.1	GET /api/salesman/saleSummary	เรียกดูข้อมูลยอดขาย เทียบยอดสั่งซื้อ ยอดกิน และยกเลิก
12	Sales Summary 1	
12.1	GET /api/salesman/saleSummary1	เรียกดูข้อมูลยอดขายรายปี

ตารางที่ 3.7 (ต่อ)

No.	API		Description
13	Sales Summary 2		
13.1	GET	/api/salesman/saleSummary2	เรียกดูข้อมูลยอดขายรายไตรมาส
14	Sales Area Chart		
14.1	GET	/api/salesman/SalesAreaChart	เรียกดูข้อมูลยอดขายตั้งแต่ต้นปี
15	Sales Radial Bar Chart		
15.1	GET	/api/salesman/saleRadialBarChart	เรียกดูข้อมูลยอด Sales Note
16	Sales Group Pie Chart		
16.1	GET	/api/salesman/saleGroupPieChart	เรียกดูข้อมูลยอดขายตามกลุ่มลูกค้า
17	ProductGroupPieChart		
17.1	GET	/api/salesman/ProductGroupPieChart	เรียกดูข้อมูลยอดขายตามกลุ่มสินค้า
18	Sales Team Pie Chart		
18.1	GET	/api/salesman/saleTeamPieChart	เรียกดูข้อมูลยอดขายตามทีมขาย
19	Sales By Province		
19.1	GET	/api/salesman/saleByProvince	เรียกดูข้อมูลยอดขายตามจังหวัด
20	Invoice Bar Chart		

ตารางที่ 3.7 (ต่อ)

No.	API		Description
20.1	GET	/api/invoice/invoiceBar Chart	เรียกดูข้อมูลยอดขาย เทียบยอดคีนสินค้า ตั้งแต่ต้นปี
21	Post		
21.1	GET	/api/post	เรียกดูข้อมูลกิจกรรมของพนักงานขาย
22	Like		
22.1	POST	/api/post/like	ส่ง-รับ ข้อมูลการกดปุ่ม Like กิจกรรมของ พนักงานขาย
23	Comment		
23.1	GET	/api/post/comment	เรียกดูข้อมูล comment กิจกรรมของ พนักงานขาย
23.2	POST		ส่ง-รับข้อมูล comment กิจกรรมของ พนักงานขาย
23.3	DELETE	/api/post/comment/{id}	ลบข้อมูล comment กิจกรรมของพนักงาน ขาย
24	KPI		
24.1	GET	/api/kpi	เรียกดูข้อมูล KPI
25	Customer Group		
25.1	GET	/api/customer/customer Group	เรียกดูข้อมูลกลุ่มลูกค้า
26	Customer Top Sales		
26.1	GET	/api/customer/customer TopSales	เรียกดูข้อมูลลูกค้า เริ่มต้นจากอันดับยอดขาย มากที่สุด

ตารางที่ 3.7 (ต่อ)

No.	API		Description
27	Customer AR Aging Card		
27.1	GET	/api/customer-aging/CustomerARAgin gCard	เรียกดูข้อมูลลูกหนี้
28	Customer AR Due Card		
28.1	GET	/api/customer-aging/CustomerARDue Card	เรียกดูข้อมูลลูกหนี้
29	Customer AR Aging Card 1		
29.1	GET	/api/customer-aging/CustomerARAgin gCard1	เรียกดูข้อมูลอายุลูกหนี้ช่วงที่ 1
30	Customer AR Aging Card 2		
30.1	GET	/api/customer-aging/CustomerARAgin gCard2	เรียกดูข้อมูลอายุลูกหนี้ช่วงที่ 2
31	Customer AR Aging Card 3		
31.1	GET	/api/customer-aging/CustomerARAgin gCard3	เรียกดูข้อมูลอายุลูกหนี้ช่วงที่ 3
32	Customer AR Due Card 1		
32.1	GET	/api/customer-aging/CustomerARDueC ard1	เรียกดูข้อมูลลูกหนี้ ที่ Over Due

ตารางที่ 3.7 (ต่อ)

No.	API	Description
33	Customer AR Due Card 2	
33.1	GET /api/customer-aging/CustomerARDueCard2	เรียกดูข้อมูลลูกหนี้ ที่ Due
34	Customer AR Due Card 3	
34.1	GET /api/customer-aging/CustomerARDueCard3	เรียกดูข้อมูลลูกหนี้ ที่ Not Due
35	Customer AR Chart	
35.1	GET /api/customer-aging/CustomerARChart	เรียกดูข้อมูลยอดรวมหนี้ค้างตามวันครบกำหนดชำระ
36	Find Customer Contact	
36.1	GET /api/customer/findCustomerContact	เรียกดูข้อมูลรายละเอียดลูกค้า Latitude Longitude
37	Find Customer Contact Name	
37.1	GET /api/customer/findCustomerContactName	เรียกดูข้อมูลผู้ติดต่อของลูกค้า
38	Find Customer Salesman	
38.1	GET /api/customer/findCustomerSaleman	เรียกดูข้อมูลพนักงานขายที่ดูแลลูกค้า
39	Find Customer Ranking and Status	
39.1	GET /api/customer/findCustomerRankAndStatus	เรียกดูข้อมูลอันดับการซื้อสินค้า และสถานะ

ตารางที่ 3.7 (ต่อ)

No.	API	Description
40	Stock Card Report	
40.1	GET /api/customer/ReportStockCard	เรียกดูข้อมูล Stock Card
41	Customer AR Report	
41.1	GET /api/customer/CusARreport	เรียกดูข้อมูลรายละเอียดบิลค้างชำระ
42	Order Fulfillment Report	
42.1	GET /api/customer/orderFulfillment	เรียกดูข้อมูลรอบการซื้อตั้งแต่ต้นปี
43	findItemGroupMaster	
43.1	GET /api/customer/findItemGroupMaster	เรียกดูข้อมูลกลุ่มสินค้า (Item Group)
44	Find Product Group	
44.1	GET /api/customer/findProductGroup	เรียกดูข้อมูลประเภทสินค้า (Product Group)
45	Find Product Line	
45.1	GET /api/customer/findProductLine	เรียกดูข้อมูลหมวดสินค้า (Product Line)
46	Product Buy	
46.1	GET /api/customer/ProductBuy	เรียกดูข้อมูลสินค้าที่มีการซื้อเฉพาะราย

ตารางที่ 3.7 (ต่อ)

No.	API	Description
47	Product Not Buy	
47.1	GET /api/customer/ProductNotBuy	เรียกดูข้อมูลสินค้าที่ยังไม่เคยซื้อเฉพาะราย
48	Customer Plan Visit	
48.1	GET /api/customer/CustomerPlanVisit	เรียกดูข้อมูลประวัติการเยี่ยมชมเฉพาะราย
49	Activities List	
49.1	GET /api/customer/ActivitiesList	เรียกดูข้อมูลกิจกรรมของพนักงานขายกับลูกค้าเฉพาะราย
50	Customer Sales Summary	
50.1	GET /api/customer/customerSummary	เรียกดูข้อมูลสรุปยอดขายรายปีและรายเดือน
51	CustomerAvgSummary	
51.1	GET /api/customer/CustomerAvgSummary	เรียกดูประวัติการเข้าเยี่ยมชมว่ามีแผนหรือไม่มีแผนในการเข้าเยี่ยมชม

3.3 เครื่องมือการพัฒนาระบบ

เว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย พัฒนาโดยใช้เครื่องมือต่าง ๆ ดังนี้

3.3.1 อุปกรณ์ฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบ

ตารางที่ 3.8 อุปกรณ์ฮาร์ดแวร์ที่ใช้ในการพัฒนาระบบ

ลำดับ	รายการ	รายละเอียด
1	เครื่องคอมพิวเตอร์แม่ข่าย (Cloud Server) จำนวน 1 เครื่อง	Application Server / Database Server CPU 2 core RAM 6 GB Hard Disk 60 GB ระบบปฏิบัติการ Windows Server 2016 Standard
2	เครื่องคอมพิวเตอร์ส่วนบุคคล (Client) จำนวน 1 เครื่อง	คอมพิวเตอร์พกพา CPU 2.6 GHz Dual-Core Intel Core i5 RAM 8 GB Hard Disk 500 GB ระบบปฏิบัติการ macOS Catalina Version 10.15.4

บทที่ 4

ผลการดำเนินงาน

ในบทนี้จะกล่าวถึงผลดำเนินการพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย

4.1 วิธีการทดสอบการพัฒนาระบบ

วิธีการทดสอบผลดำเนินการพัฒนาแบ่งเป็นสองส่วนคือการทำ UnitTest โดยใช้เครื่องมือ ช่วยในการทดสอบ คือ Swagger และทำ Integration Test เข้ากับ แอปพลิเคชัน ของผู้ใช้ ในฝั่งไคลเอนท์ โดยใช้กรณีทดสอบ ดังในตารางที่ 4.1 โดยมีขั้นตอนการทดสอบด้วย Swagger แสดงตัวอย่างในภาพที่ 4.1 ดังต่อไปนี้

1. ใส่ URL ของ API
2. เลือก API ที่ต้องการทดสอบ
3. เลือก Method ที่ต้องการ
4. กดปุ่ม “Try it out”
5. ระบุ Paramater ที่ต้องการค้นหา
6. ดูผลลัพธ์การทำงาน (Response body)

และในส่วนของการทำ Integration Test จะแสดงภาพหน้าจอผลลัพธ์จากการเรียกใช้งานเซอร์วิสจากแอปพลิเคชันในฝั่งไคลเอนท์ทั้งในส่วนของผู้ใช้

ตารางที่ 4.1 Test Case ที่นำมาทดสอบการทำงานของเมนู Performance Overall

No.	Test Case ID	Test Case	Use Case ID
1	TC101	ผู้ใช้งานตรวจสอบข้อมูล KPI	PO0100
2	TC102	ผู้ใช้งานตรวจสอบจำนวนลูกค้าที่มีการซื้อขาย	PO0100
3	TC103	ผู้ใช้งานตรวจสอบจำนวนการเข้าเยี่ยมชมลูกค้า	PO0100
4	TC104	ผู้ใช้งานติดตามยอดขายสะสมรายปี	PO0100
5	TC105	ผู้ใช้งานติดตามยอดขายสะสมรายไตรมาส	PO0100
6	TC106	ผู้ใช้งานติดตามยอดขายสะสมรายเดือน	PO0100
7	TC107	ผู้ใช้งานติดตามปริมาณการสั่งซื้อ	PO0100
8	TC108	ผู้ใช้งานติดตามยอดค้างส่งสินค้า	PO0100
9	TC109	ผู้ใช้งานติดตามยอดคืนสินค้า	PO0100
10	TC110	ผู้ใช้งานวิเคราะห์ยอดขายเปรียบเทียบช่วงเวลาเดียวกันของปีก่อน	PO0100
11	TC111	ผู้ใช้งานเปรียบเทียบการสั่งซื้อสินค้าและการส่งสินค้า	PO0100
12	TC112	ผู้ใช้งานเปรียบเทียบการส่งสินค้าและการคืนสินค้า	PO0100
13	TC113	ผู้ใช้งานเปรียบเทียบการสั่งซื้อสินค้าและค้างส่งสินค้า	PO0100
14	TC114	ผู้ใช้งานเปรียบเทียบสัดส่วนการขายตามลูกค้า	PO0100
15	TC115	ผู้ใช้งานเปรียบเทียบสัดส่วนการขายตามสินค้า	PO0100
16	TC116	ผู้ใช้งานเปรียบเทียบสัดส่วนการขายตามทีมขาย	PO0100
17	TC117	ผู้ใช้งานติดตามยอดสะสมมูลค่าการขายสินค้าเทียบการคืนสินค้า	PO0100
18	TC118	ผู้ใช้งานติดตามการขายรายจังหวัด	PO0100
19	TC119	ผู้ใช้งานสื่อสารภายในองค์กรผ่านกิจกรรมการทำงาน	PO0100

ตารางที่ 4.2 Test Case ที่นำมาทดสอบการทำงานของเมนู Customer Overall

No.	Test Case ID	Test Case	Use Case ID
1	TC201	ผู้ใช้เปรียบเทียบลูกค้าใหม่	PO0200
2	TC202	ผู้ใช้ติดตามความเคลื่อนไหวลูกค้าทุกสถานะ	PO0200
3	TC203	ผู้ใช้เปรียบเทียบจำนวนลูกค้าซื้อ	PO0200
4	TC204	ผู้ใช้ติดตามความเคลื่อนไหวลูกค้าซื้อ/ลูกค้าใหม่	PO0200
5	TC205	ผู้ใช้ติดตามความเคลื่อนไหวลูกค้าซื้อ	PO0200
6	TC206	ผู้ใช้ติดตามยอดขายตามกลุ่มลูกค้า	PO0200
7	TC207	ผู้ใช้ติดตามข้อมูลที่ซื้อรายลูกค้า	PO0200
8	TC208	ผู้ใช้ติดตามสถานะลูกหนี้	PO0200

ตารางที่ 4.3 Test Case ที่นำมาทดสอบการทำงานของเมนู Customer Focus

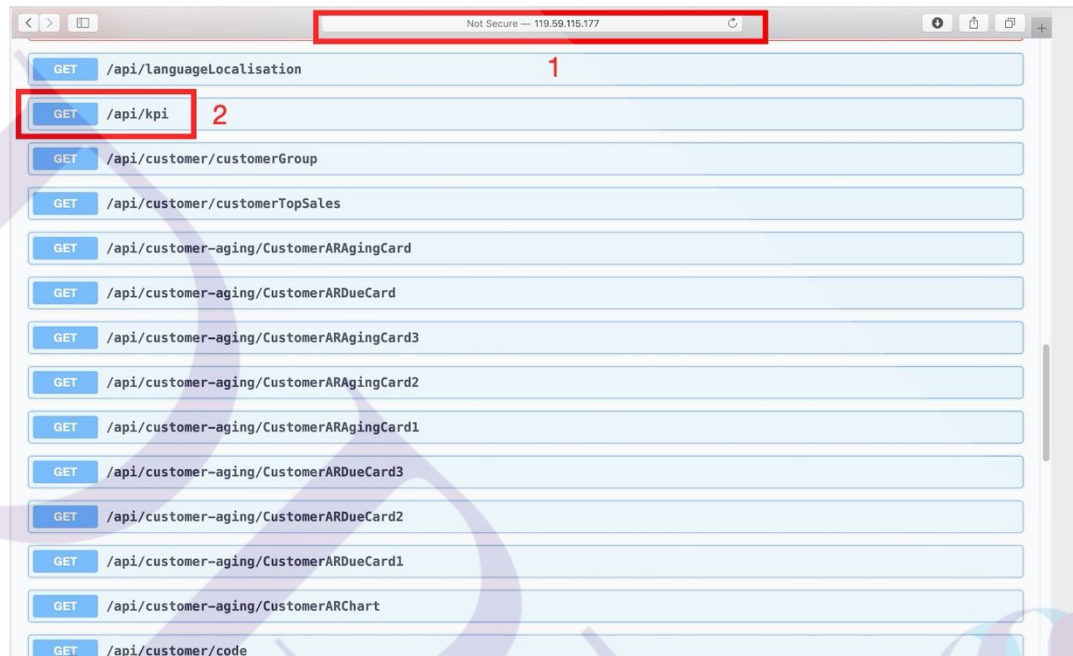
No.	Test Case ID	Test Case	Use Case ID
1	TC301	ผู้ใช้ตรวจสอบข้อมูลหน้าบัญชี	PO0300
2	TC302	ผู้ใช้ติดตามยอดซื้อสะสม/ยอดส่งสินค้ารายปี	PO0300
3	TC303	ผู้ใช้ติดตามยอดซื้อสะสม/ยอดส่งสินค้ารายเดือน	PO0300
4	TC304	ผู้ใช้ติดตามรายละเอียดบิลค้ำชำระ	PO0300
5	TC305	ผู้ใช้ติดตามยอดระบายในตลาด	PO0300
6	TC306	ผู้ใช้ติดตามประวัติการขาย	PO0300
7	TC307	ผู้ใช้ติดตามปริมาณการขาย/คืน	PO0300
8	TC308	ผู้ใช้เปรียบเทียบสัดส่วนการขาย	PO0300
9	TC307	ผู้ใช้ติดตามสินค้าที่ยังไม่มีการขาย	PO0300
10	TC308	ผู้ใช้ติดตามกิจกรรมที่ทำกับลูกค้า	PO0300

4.2 ผลการพัฒนาระบบ

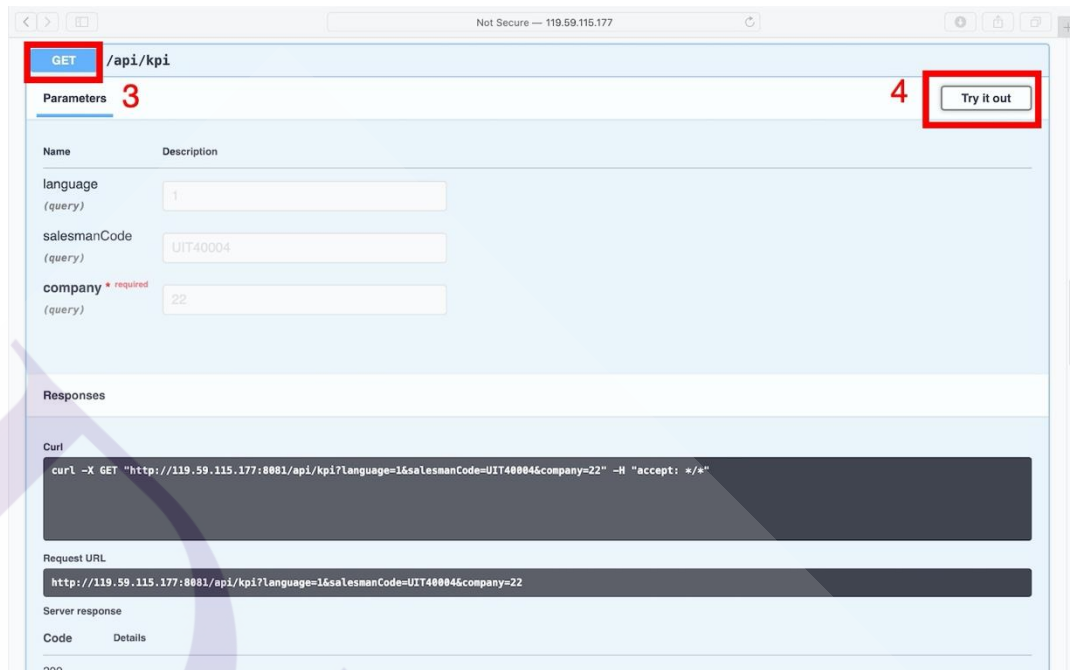
4.2.1 กรณีทดสอบ TC01

สำหรับผลการพัฒนาระบบในส่วนของข้อมูล KPI มีดังนี้

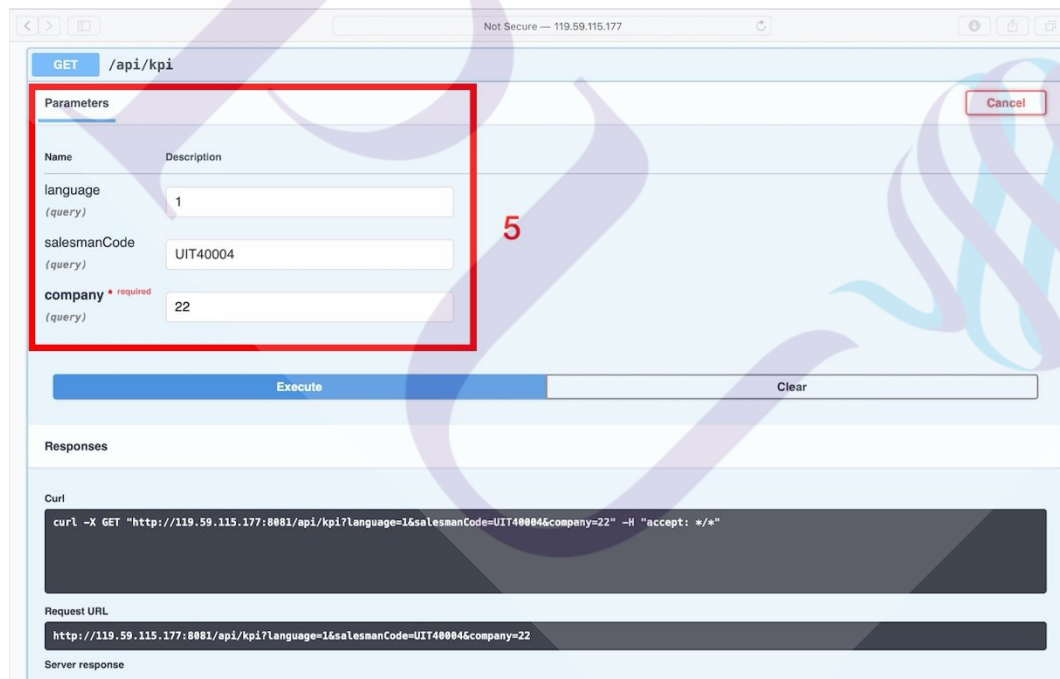
4.2.1.1 ทดสอบเรียกเซอรัวิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้



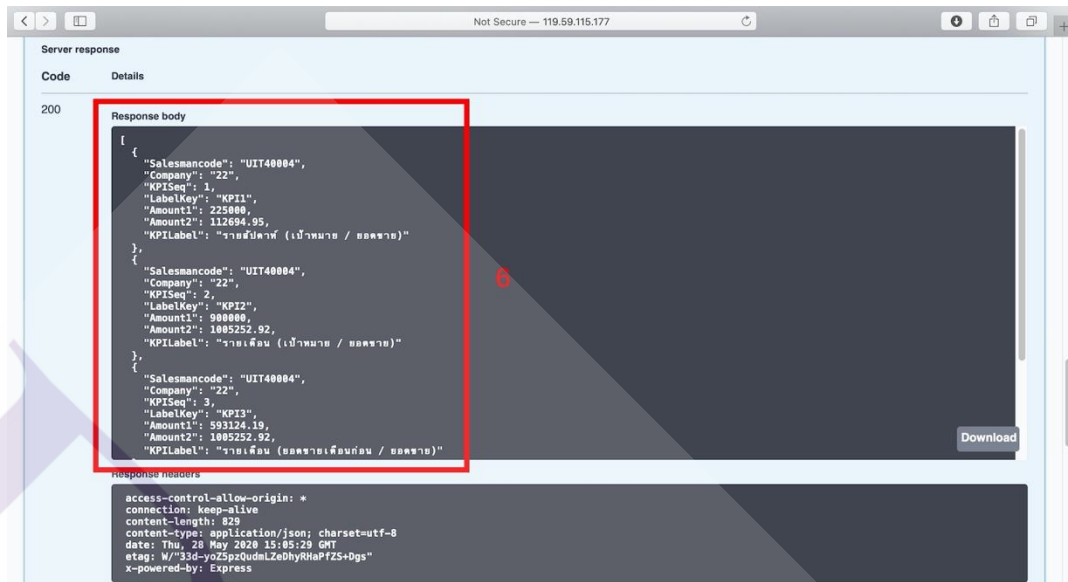
ภาพที่ 4.1 กรณีทดสอบ TC01



ภาพที่ 4.1 (ต่อ)



ภาพที่ 4.1 (ต่อ)



Server response

Code Details

200

Response body

```
[
  {
    "Salesmancode": "UIT40004",
    "Company": "22",
    "KPISeq": 1,
    "LabelKey": "KPI1",
    "Amount1": 225000,
    "Amount2": 112694.95,
    "KPILabel": "รายสินค้า (เป้าหมาย / ยอดขาย)"
  },
  {
    "Salesmancode": "UIT40004",
    "Company": "22",
    "KPISeq": 2,
    "LabelKey": "KPI2",
    "Amount1": 900000,
    "Amount2": 1005252.92,
    "KPILabel": "รายเดือน (เป้าหมาย / ยอดขาย)"
  },
  {
    "Salesmancode": "UIT40004",
    "Company": "22",
    "KPISeq": 3,
    "LabelKey": "KPI3",
    "Amount1": 503124.19,
    "Amount2": 1005252.92,
    "KPILabel": "รายเดือน (ยอดขายเดือนก่อน / ยอดขาย)"
  }
]
```

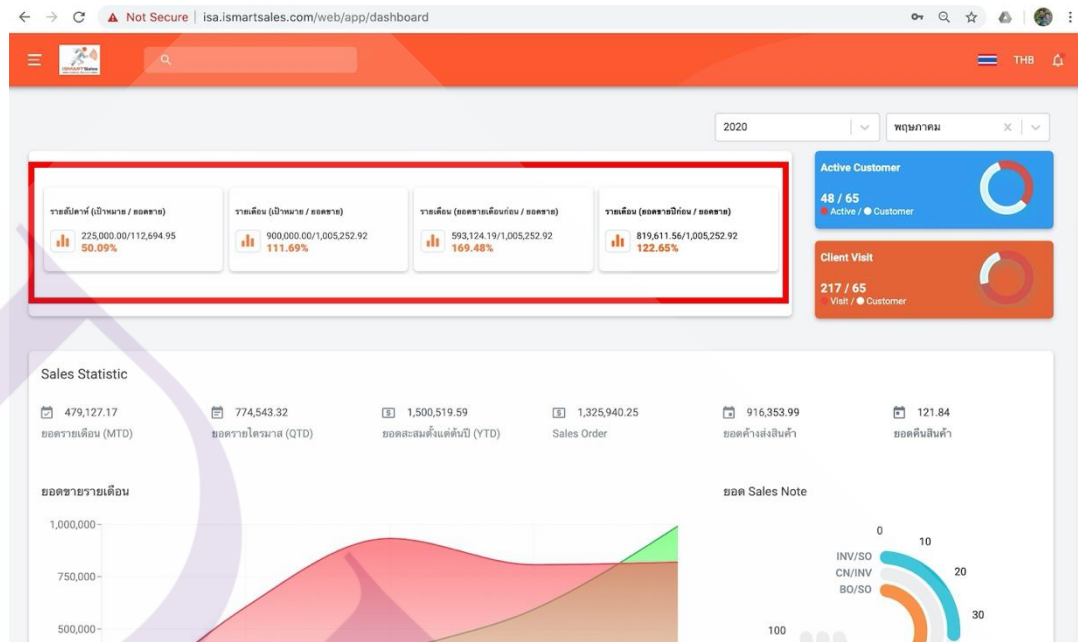
Download

Response headers

```
access-control-allow-origin: *
connection: keep-alive
content-length: 829
content-type: application/json; charset=utf-8
date: Thu, 28 May 2020 15:05:29 GMT
etag: W/"33d-y0Z5p20udmZeDhyRHpfZS+0gs"
x-powered-by: Express
```

ภาพที่ 4.1 (ต่อ)

4.2.1.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

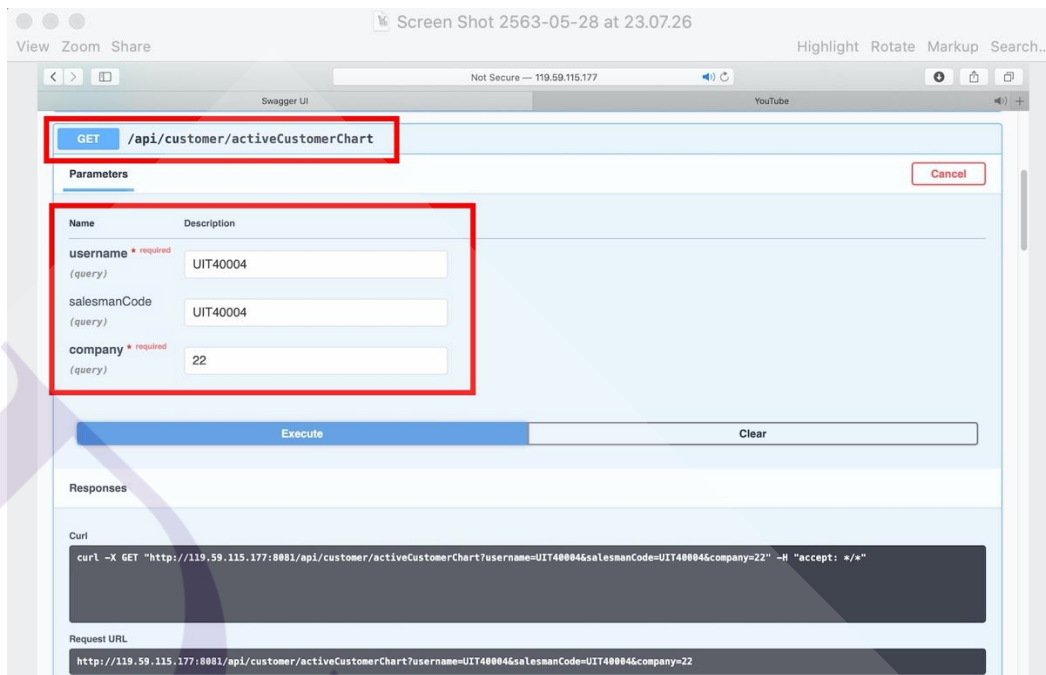


ภาพที่ 4.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

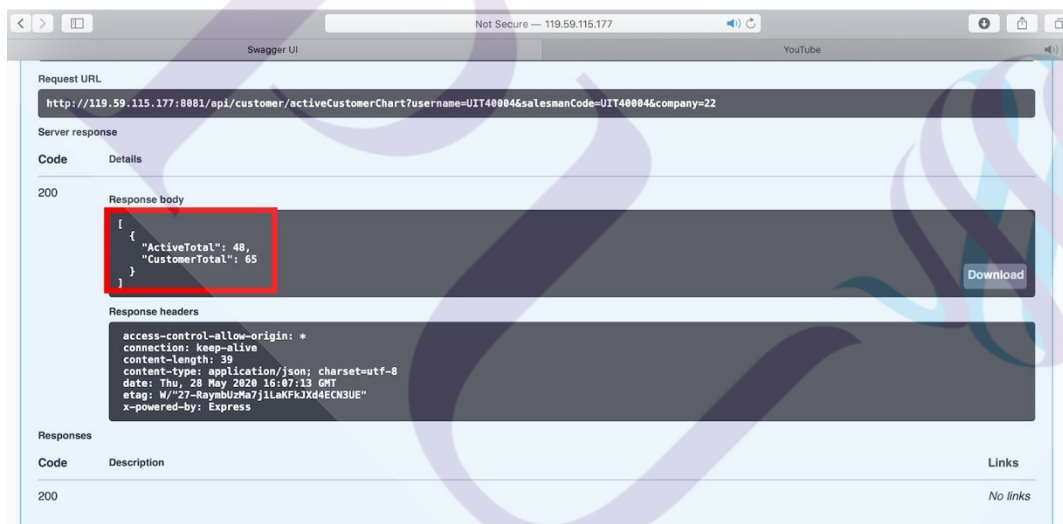
4.2.2 กรณีทดสอบ TC02

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.2.1 ทดสอบเรียกเซอร์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

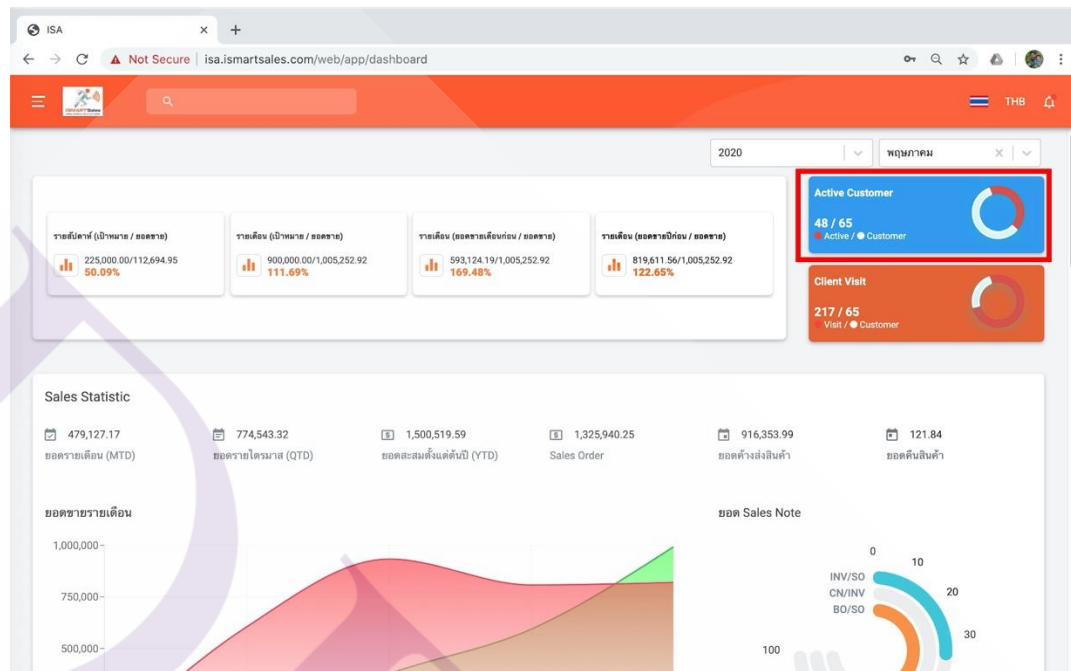


ภาพที่ 4.3 กรณีทดสอบ TC02



ภาพที่ 4.3 (ต่อ)

4.2.2.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

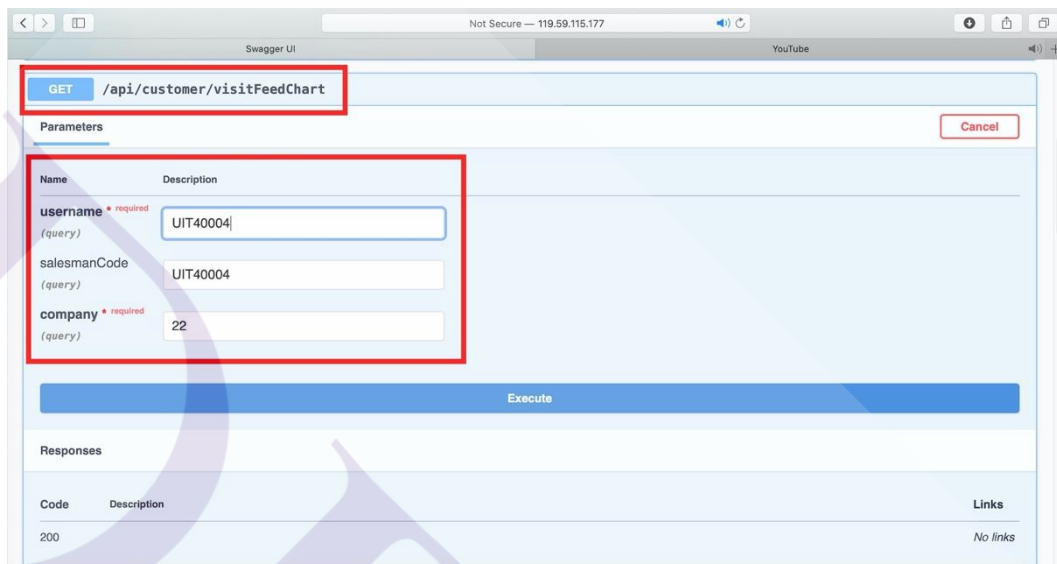


ภาพที่ 4.4 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

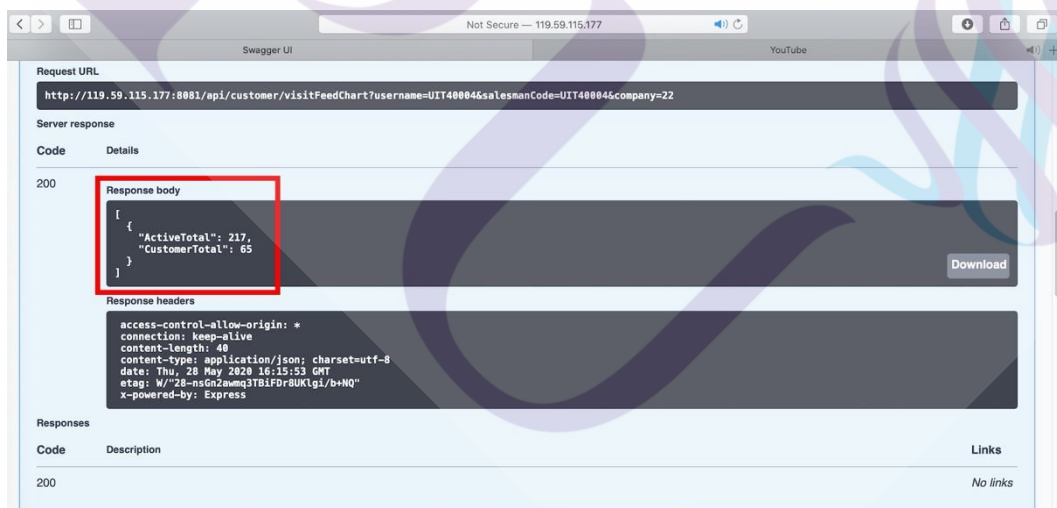
4.2.3 กรณีทดสอบ TC03

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.3.1 ทดสอบเรียกเซอรัวิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

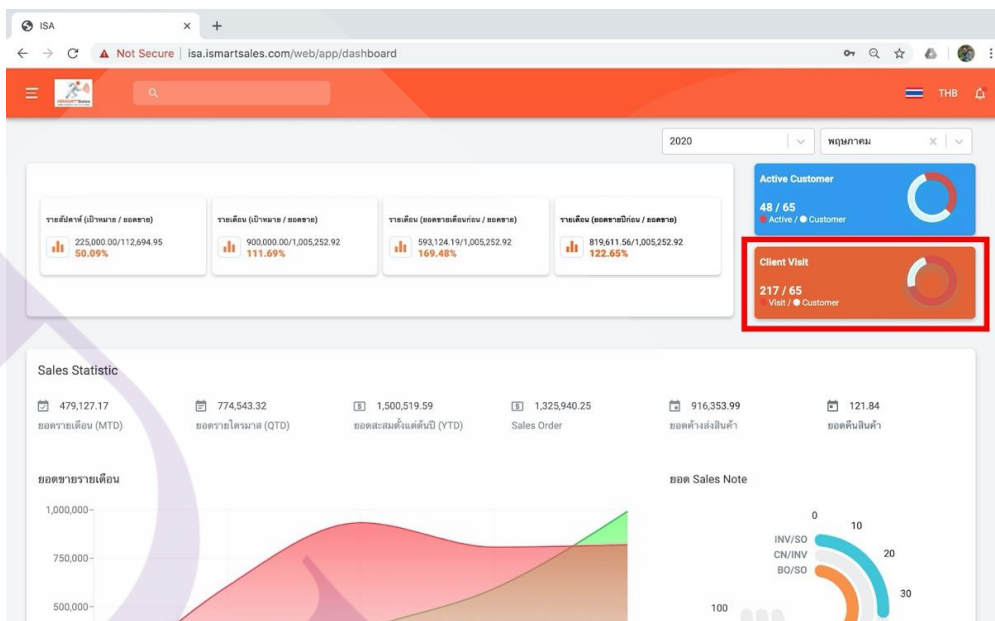


ภาพที่ 4.5 กรณีทดสอบ TC03



ภาพที่ 4.5 (ต่อ)

4.2.3.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

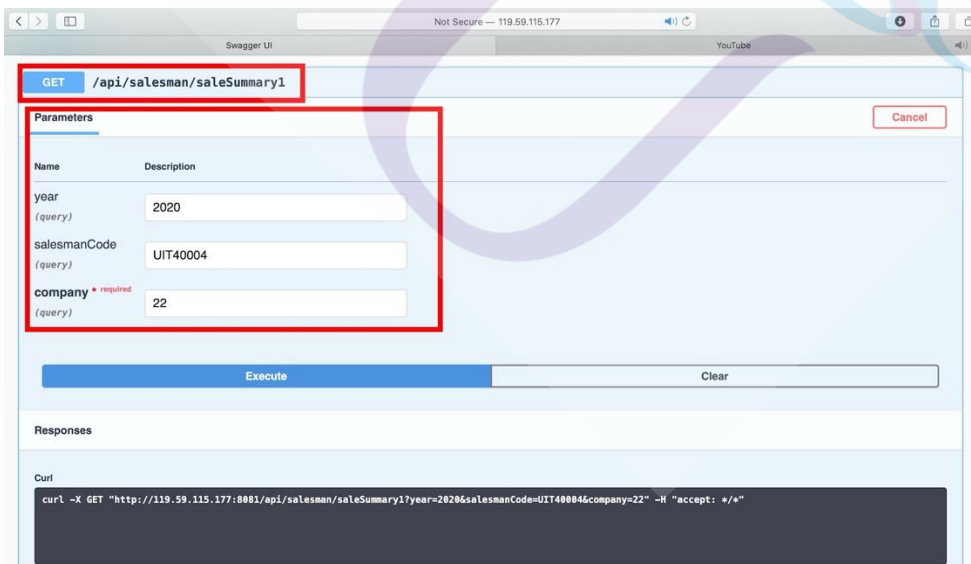


ภาพที่ 4.6 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

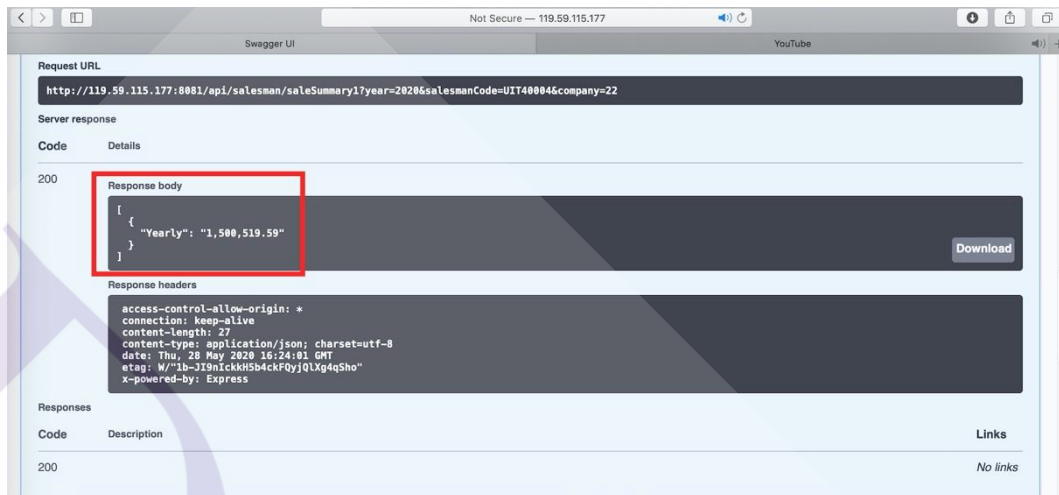
4.2.4 กรณีทดสอบ TC04

สำหรับผลการพัฒนาระบบในส่วนของการเชื่อมข้อมูล KPI มีดังนี้

4.2.4.1 ทดสอบเรียกเซอร์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

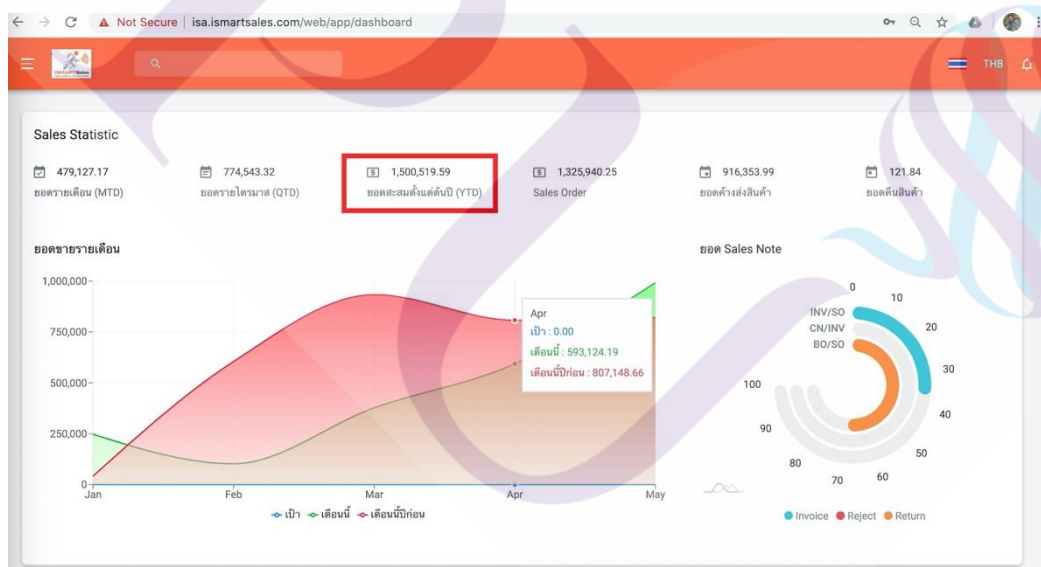


ภาพที่ 4.7 กรณีทดสอบ TC04



ภาพที่ 4.7 (ต่อ)

4.2.4.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

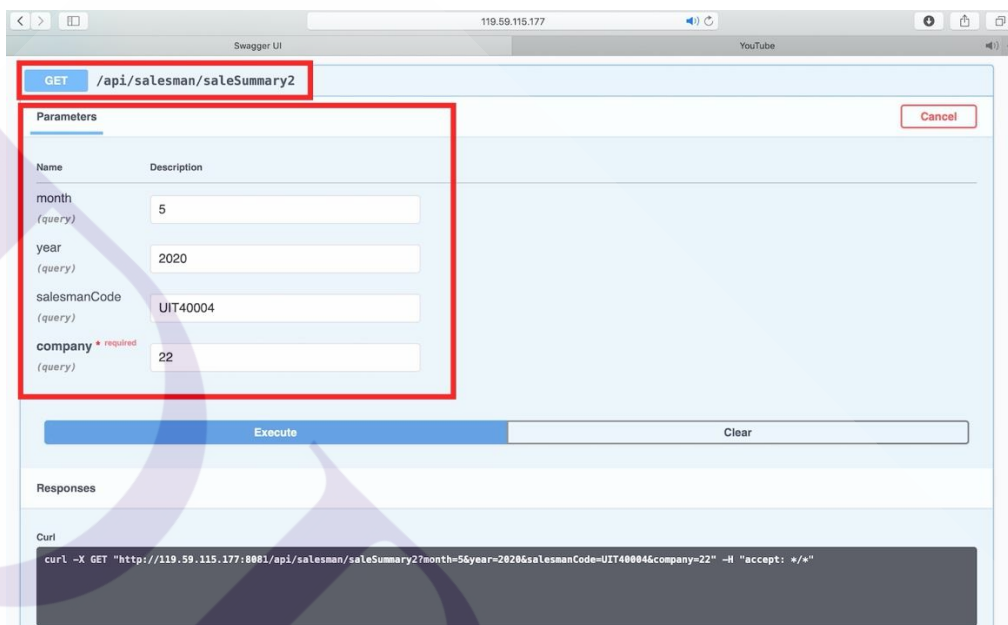


ภาพที่ 4.8 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

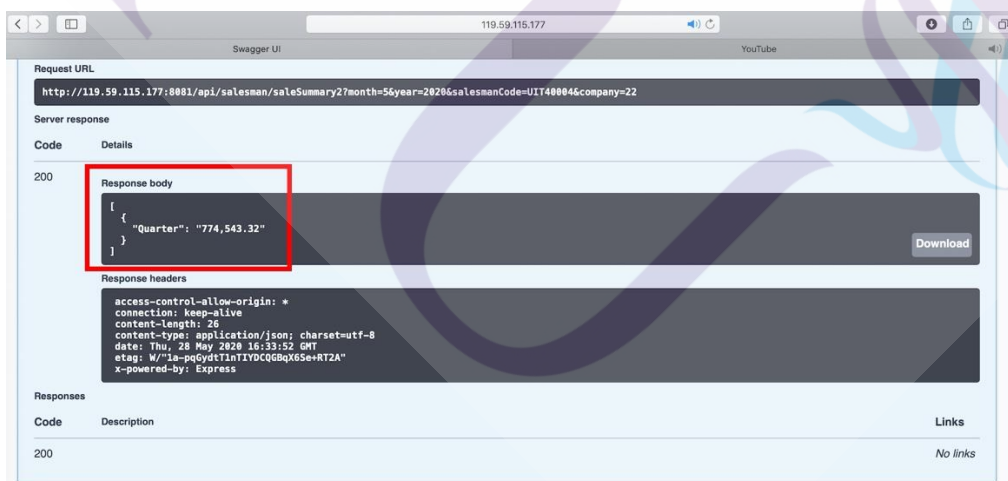
4.2.5 กรณีทดสอบ TC05

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.5.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

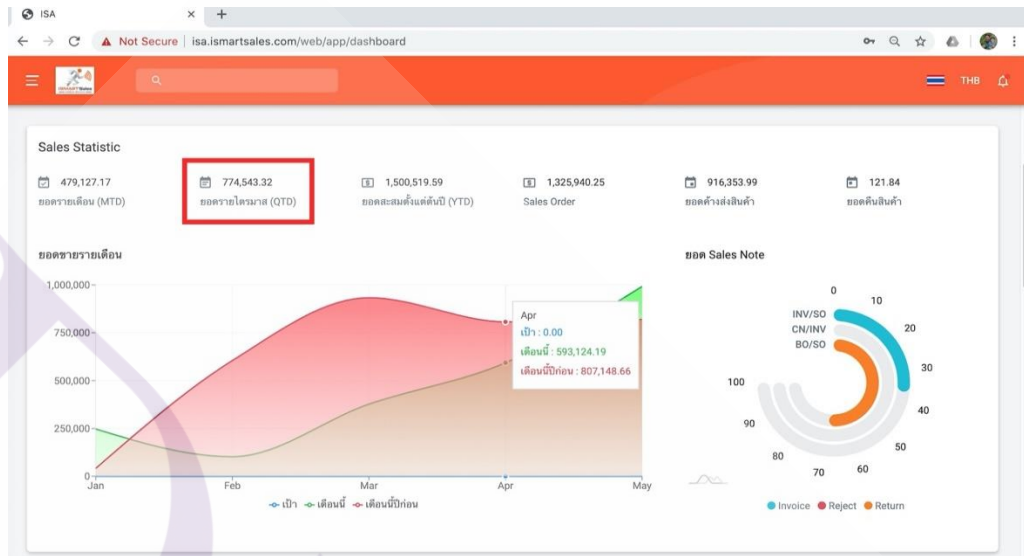


ภาพที่ 4.9 กรณีทดสอบ TC05



ภาพที่ 4.9 (ต่อ)

4.2.5.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

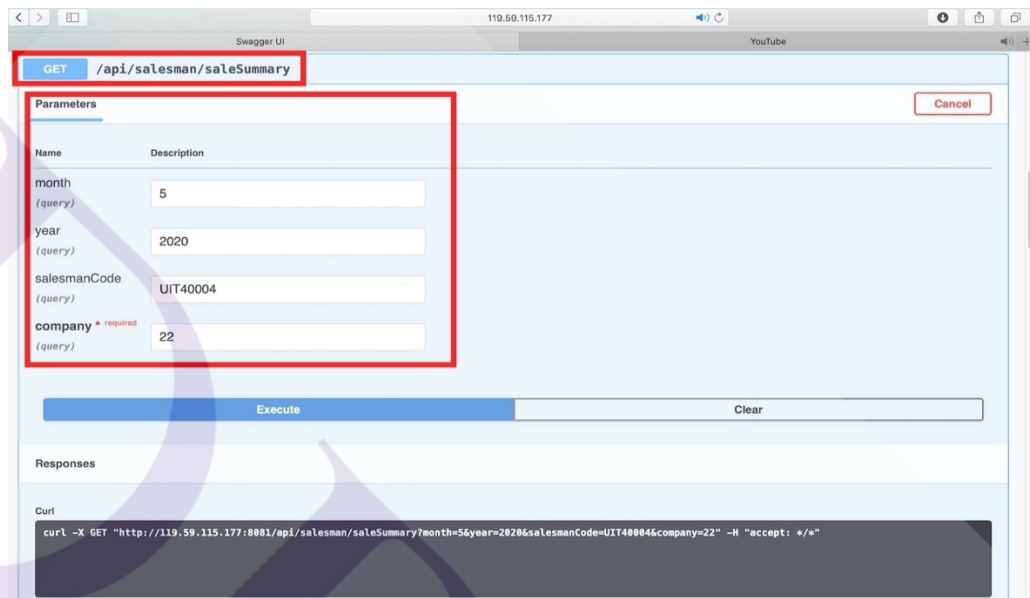


ภาพที่ 4.10 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

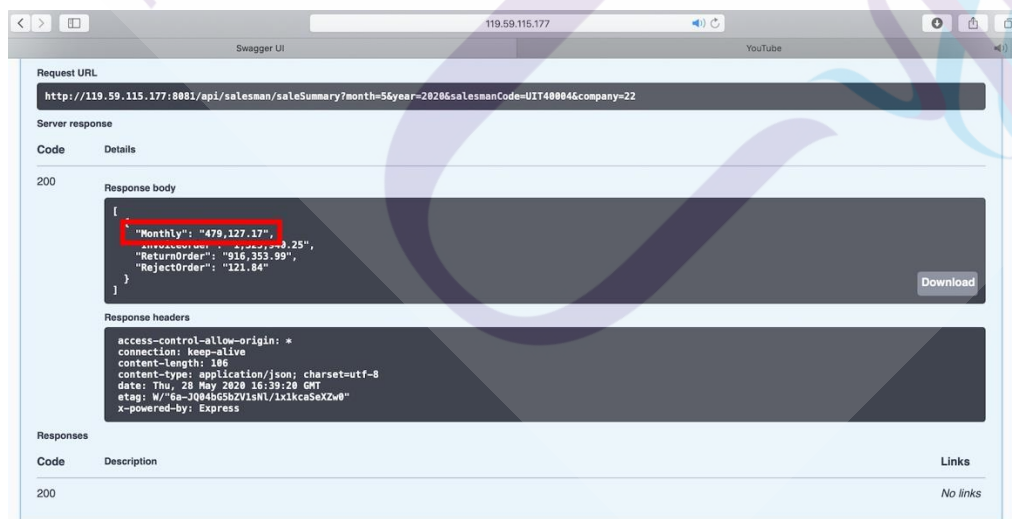
4.2.6 กรณีทดสอบ TC06

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.6.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

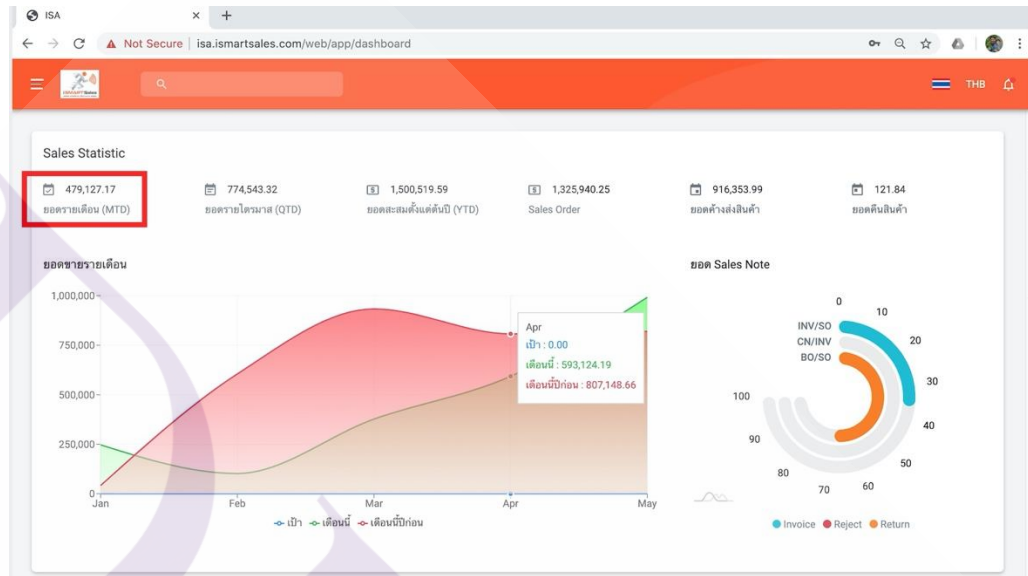


ภาพที่ 4.11 กรณีทดสอบ TC06



ภาพที่ 4.11 (ต่อ)

4.2.6.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

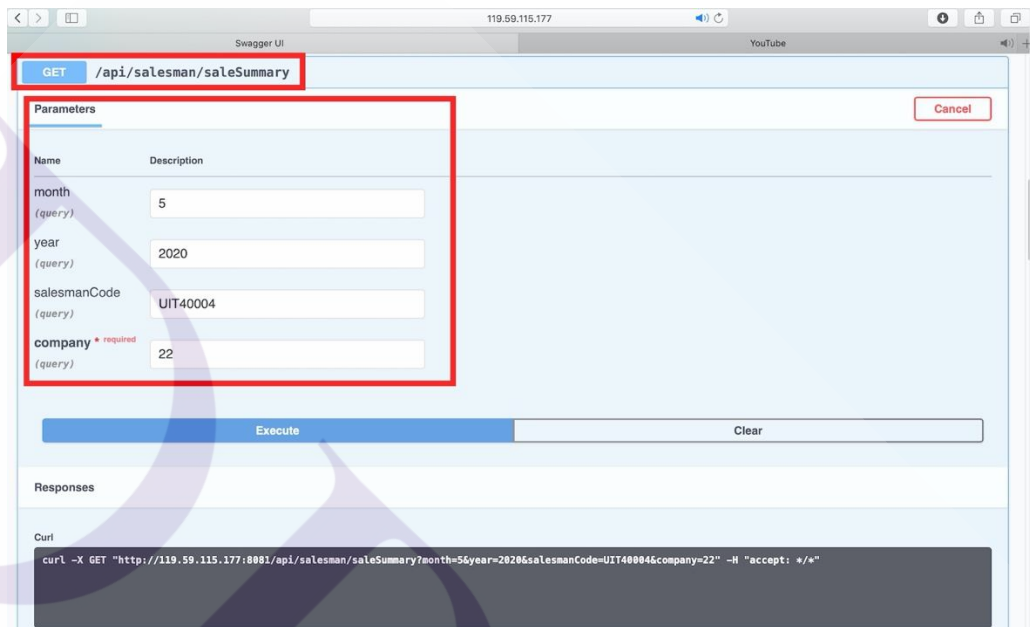


ภาพที่ 4.12 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

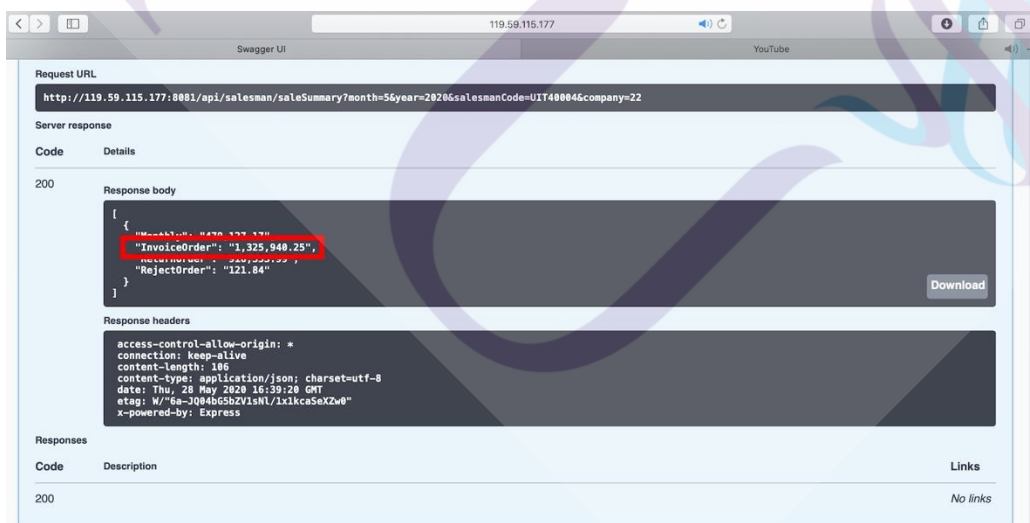
4.2.7 กรณีทดสอบ TC07

สำหรับผลการพัฒนาระบบในส่วนของข้อมูล KPI มีดังนี้

4.2.7.1 ทดสอบเรียกเซอรัวิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

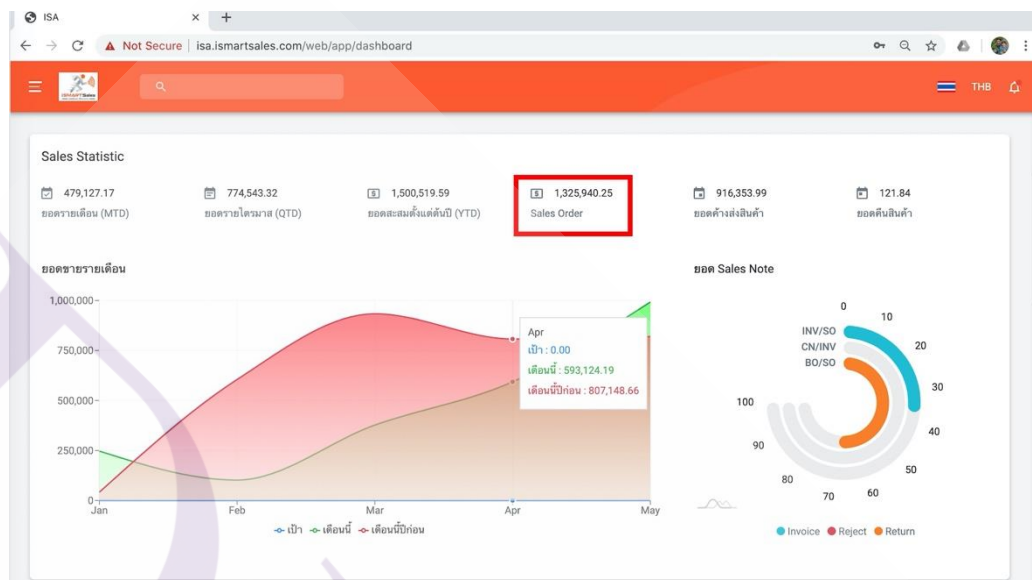


ภาพที่ 4.13 กรณีทดสอบ TC07



ภาพที่ 4.13 (ต่อ)

4.2.7.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

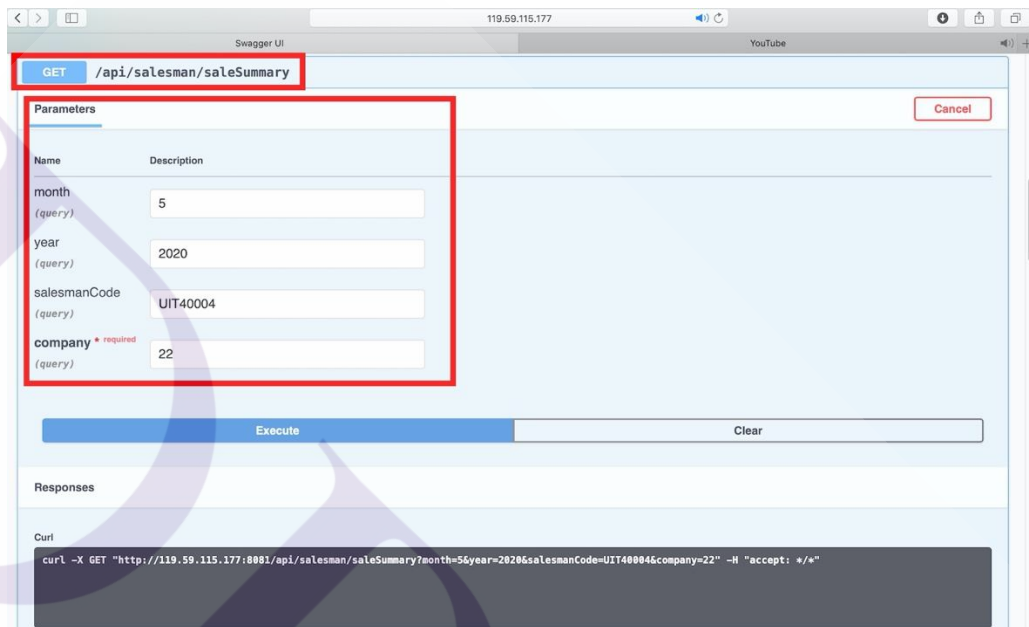


ภาพที่ 4.14 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

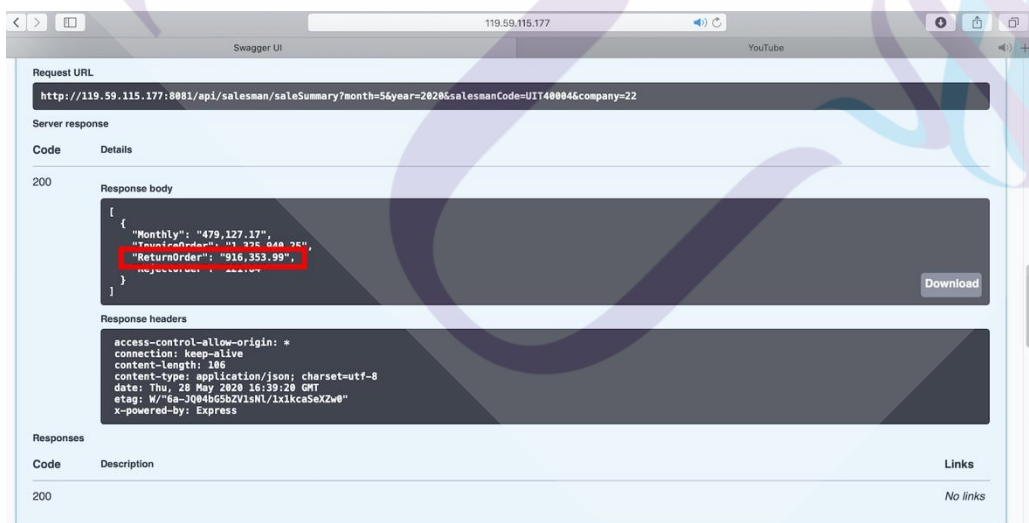
4.2.8 กรณีทดสอบ TC08

สำหรับผลการพัฒนาระบบในส่วนของข้อมูล KPI มีดังนี้

4.2.8.1 ทดสอบเรียกเซอร์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

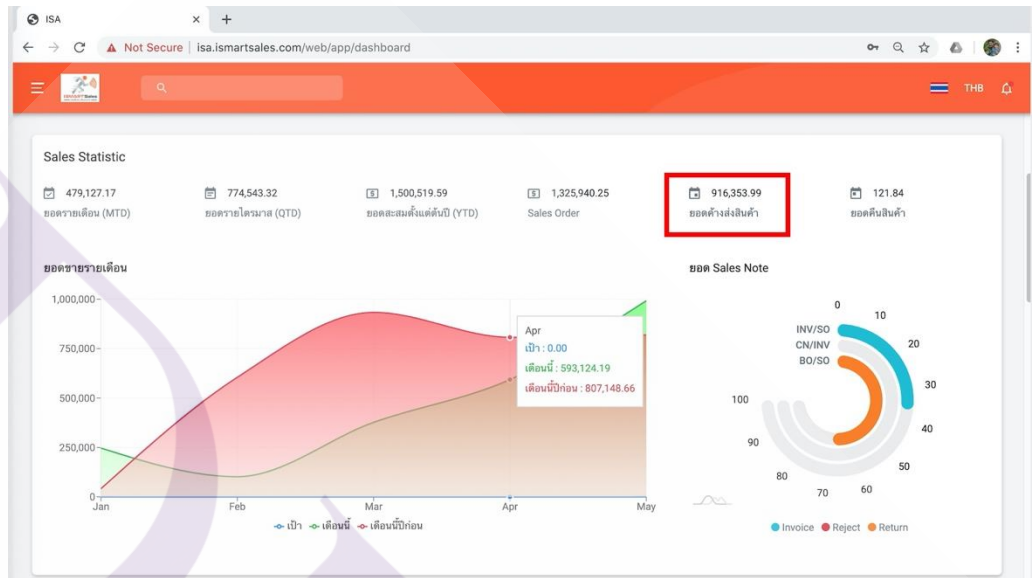


ภาพที่ 4.15 กรณีทดสอบ TC08



ภาพที่ 4.15 (ต่อ)

4.2.8.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

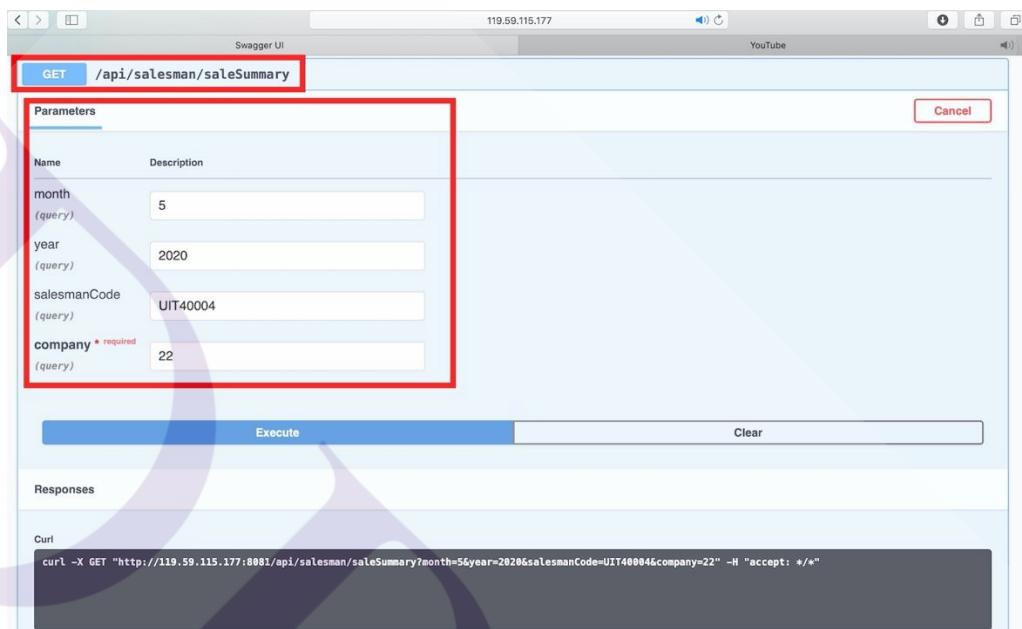


ภาพที่ 4.16 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

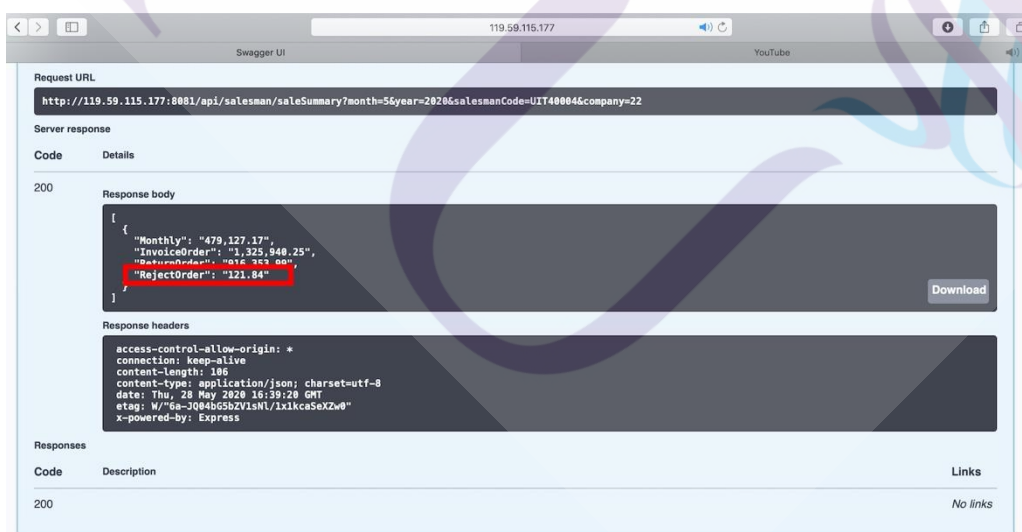
4.2.9 กรณีทดสอบ TC09

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.9.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

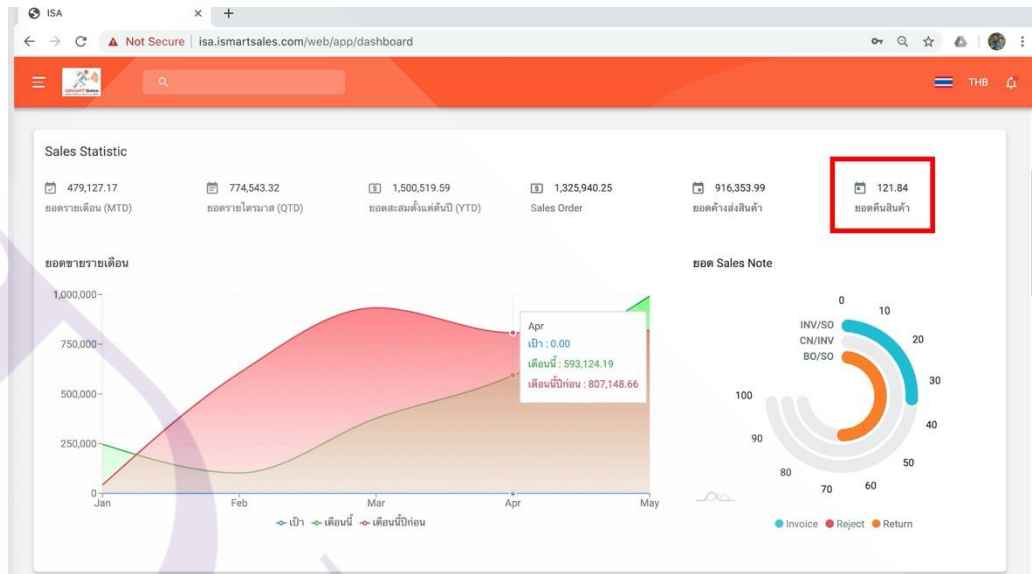


ภาพที่ 4.17 กรณีทดสอบ TC09



ภาพที่ 4.17 (ต่อ)

4.2.9.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

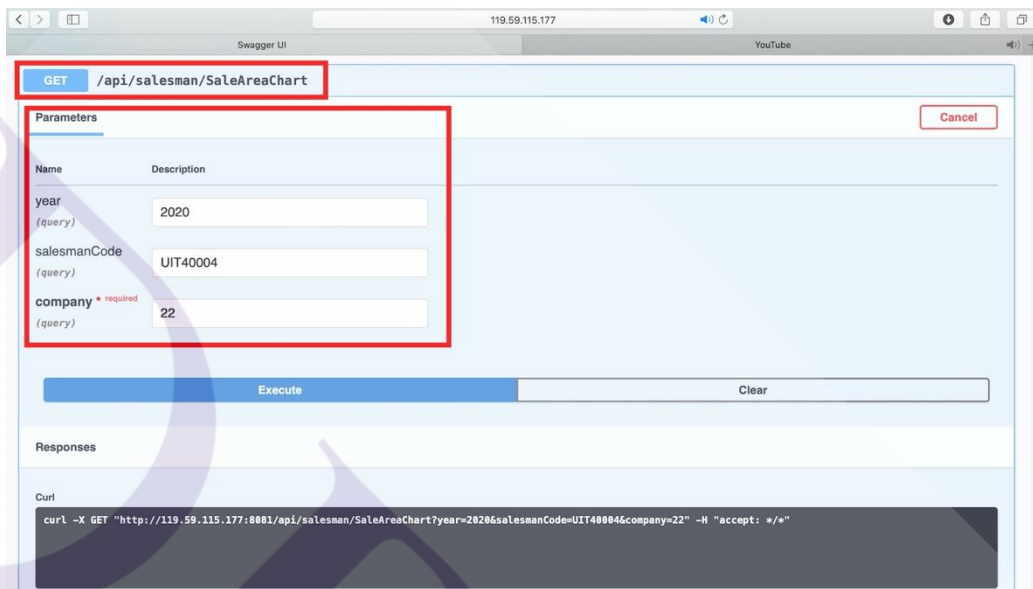


ภาพที่ 4.18 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

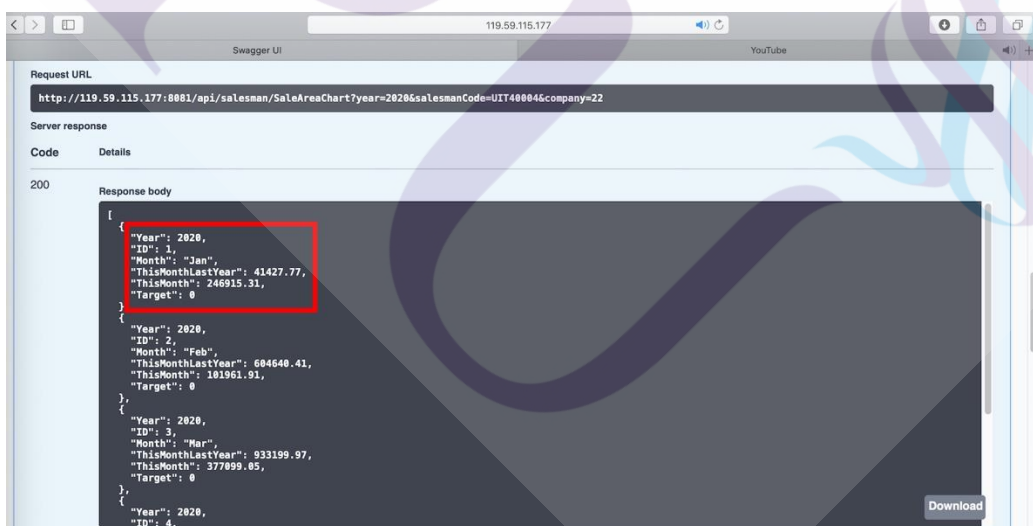
4.2.10 กรณีทดสอบ TC10

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.10.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้



ภาพที่ 4.19 กรณีทดสอบ TC10



ภาพที่ 4.19 (ต่อ)

4.2.10.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

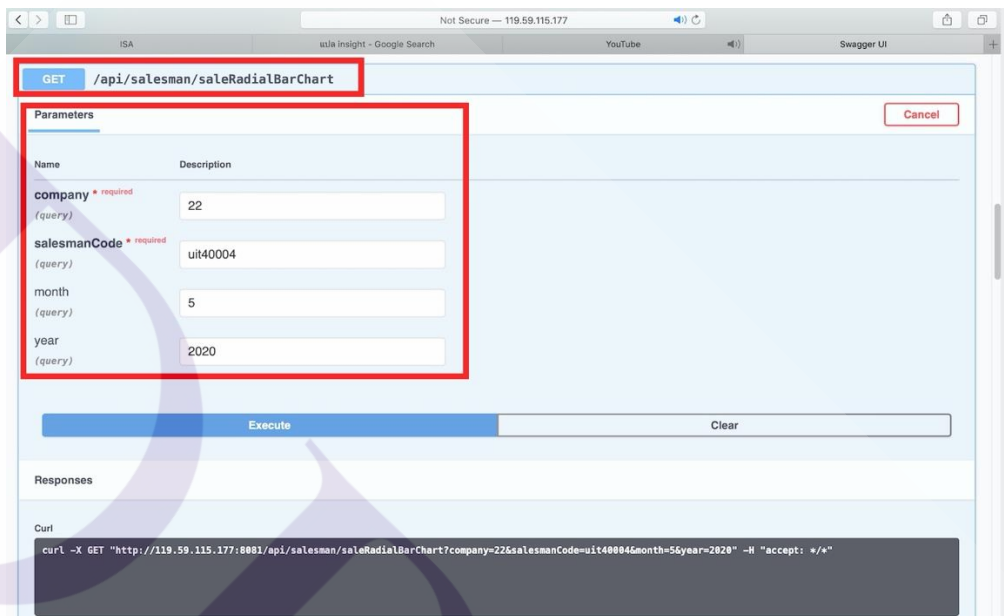


ภาพที่ 4.20 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

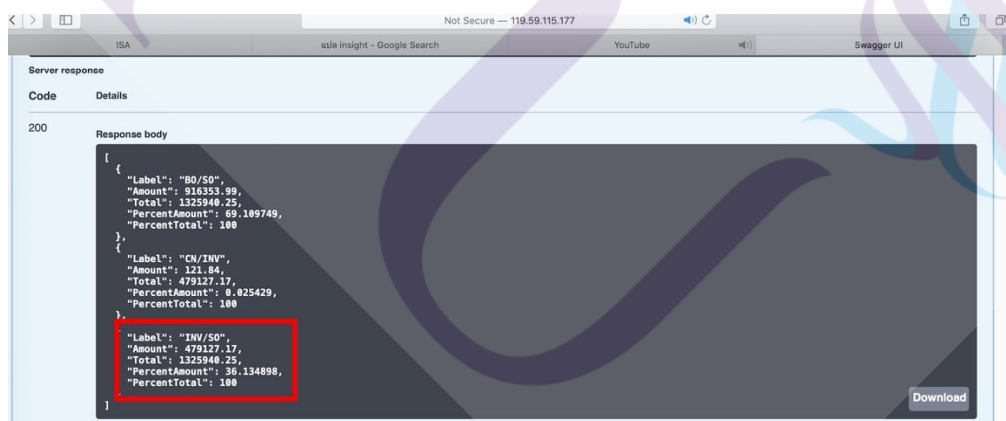
4.2.11 กรณีทดสอบ TC11

สำหรับผลการพัฒนาระบบในส่วนของข้อมูล KPI มีดังนี้

4.2.11.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

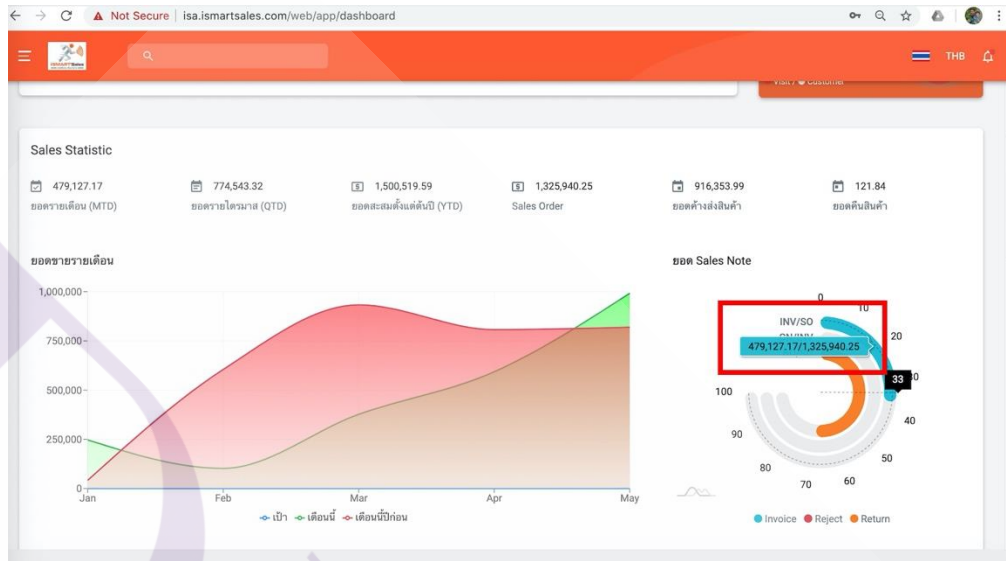


ภาพที่ 4.21 กรณีทดสอบ TC11



ภาพที่ 4.21 (ต่อ)

4.2.11.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

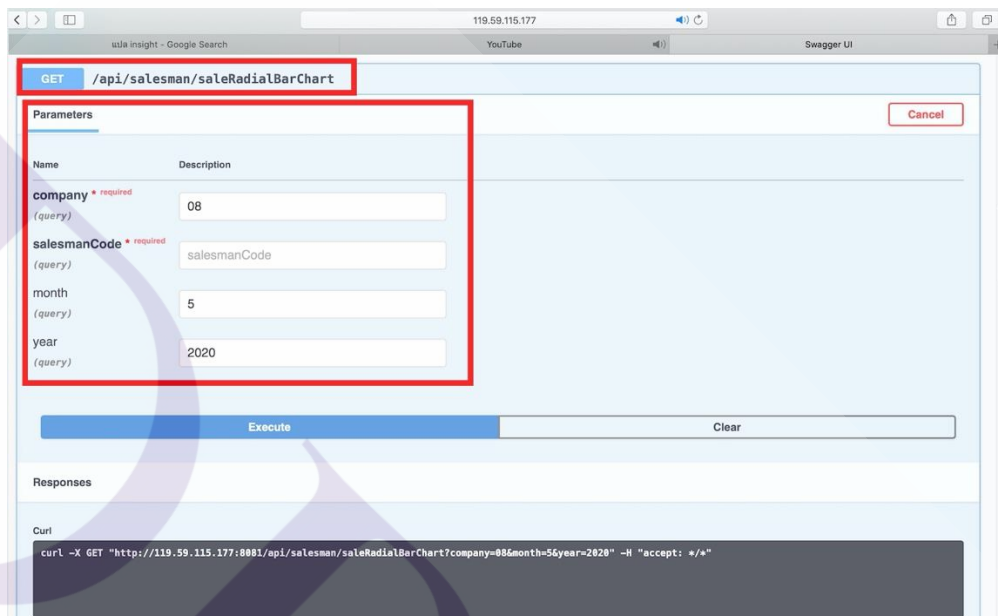


ภาพที่ 4.22 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

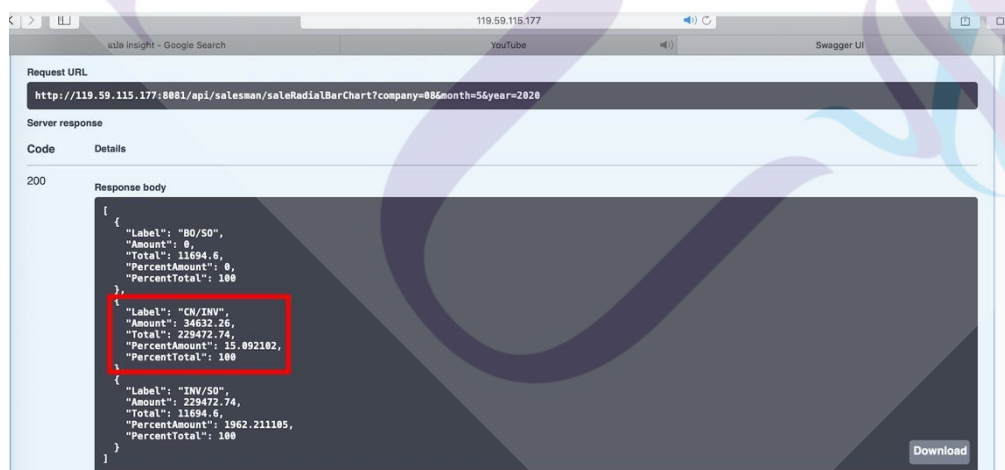
4.2.12 กรณีทดสอบ TC12

สำหรับผลการพัฒนาระบบในส่วนของข้อมูล KPI มีดังนี้

4.2.12.1 ทดสอบเรียกเซอร์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้



ภาพที่ 4.23 กรณีทดสอบ TC12



ภาพที่ 4.23 (ต่อ)

4.2.12.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

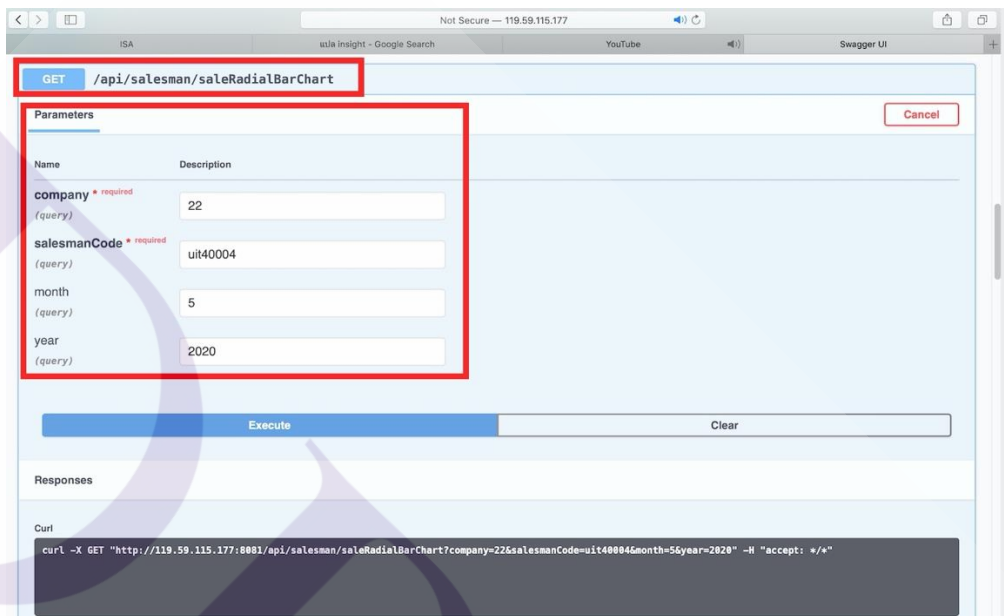


ภาพที่ 4.24 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

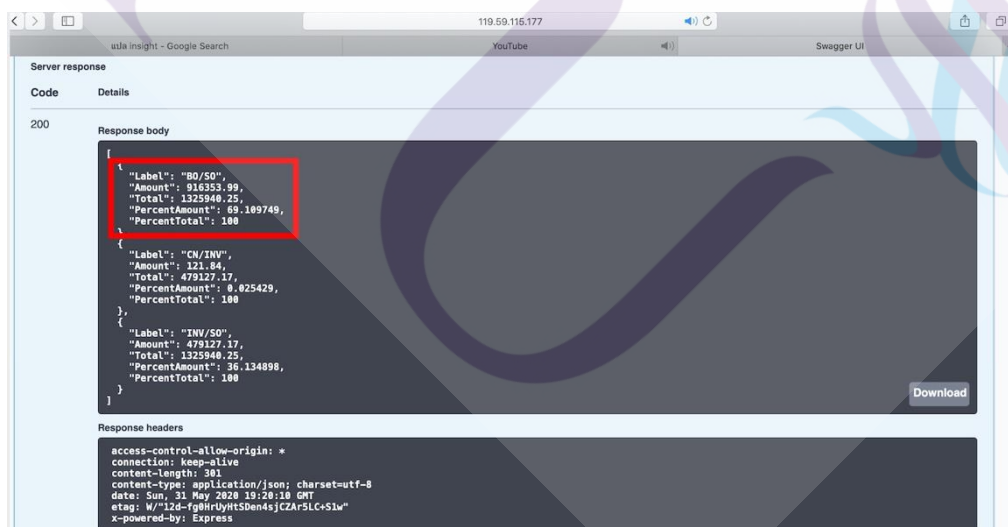
4.2.13 กรณีทดสอบ TC13

สำหรับผลการพัฒนาระบบในส่วนของข้อมูล KPI มีดังนี้

4.2.13.1 ทดสอบเรียกเซอร์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

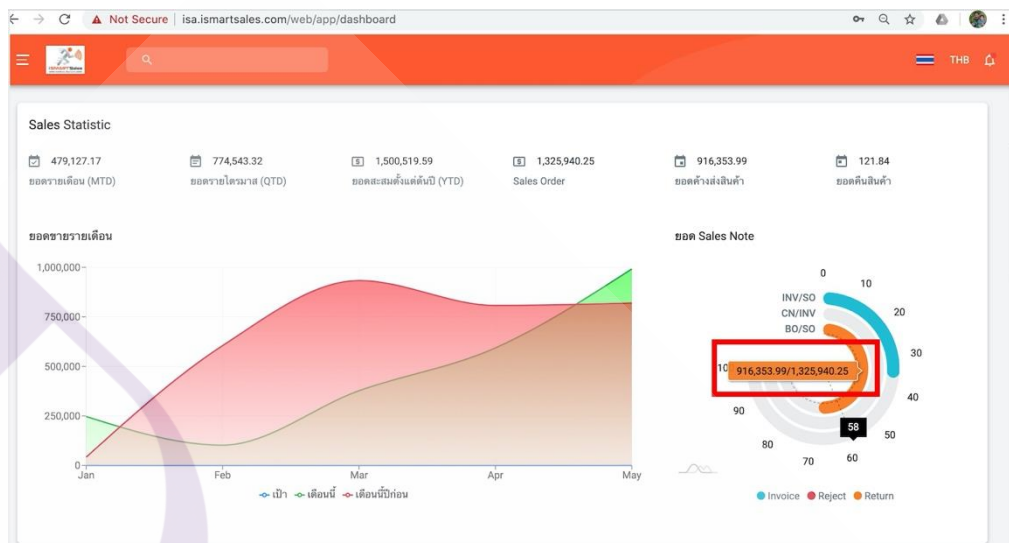


ภาพที่ 4.25 กรณีทดสอบ TC13



ภาพที่ 4.25 (ต่อ)

4.2.13.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

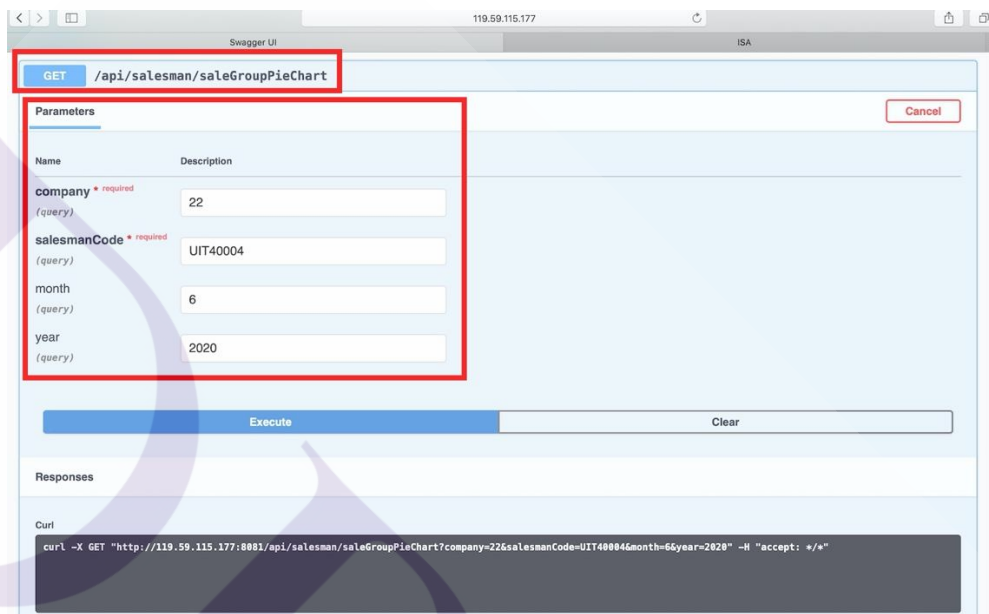


ภาพที่ 4.26 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

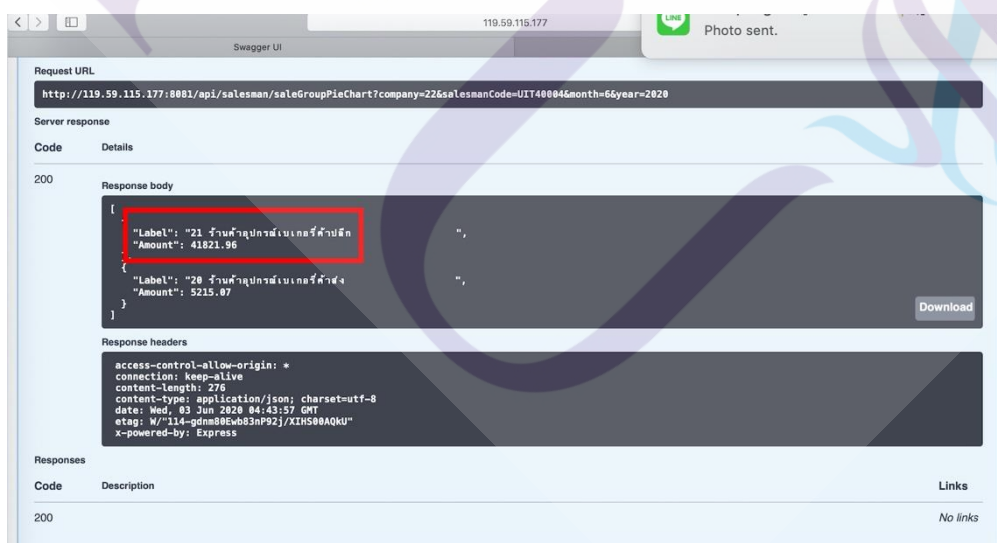
4.2.14 กรณีทดสอบ TC14

สำหรับผลการพัฒนาระบบในส่วนของคุณค่า KPI มีดังนี้

4.2.14.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

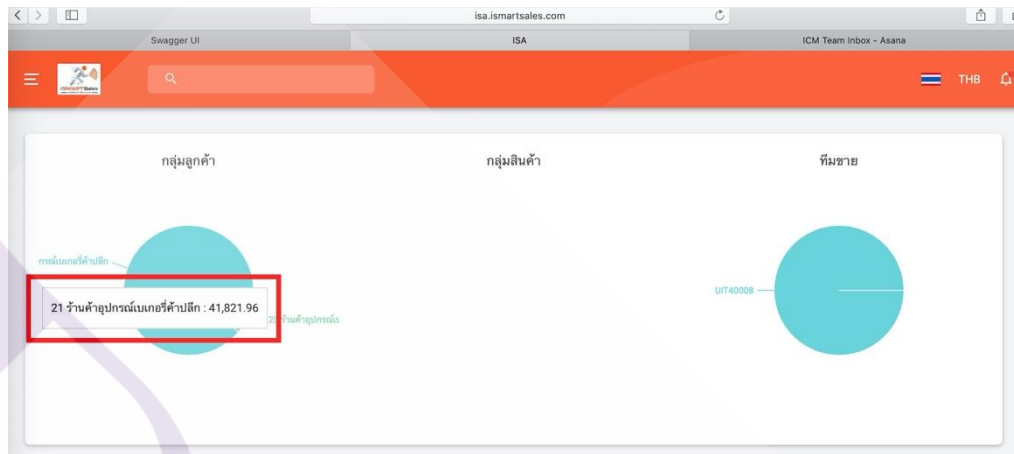


ภาพที่ 4.27 กรณีทดสอบ TC14



ภาพที่ 4.27 (ต่อ)

4.2.14.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

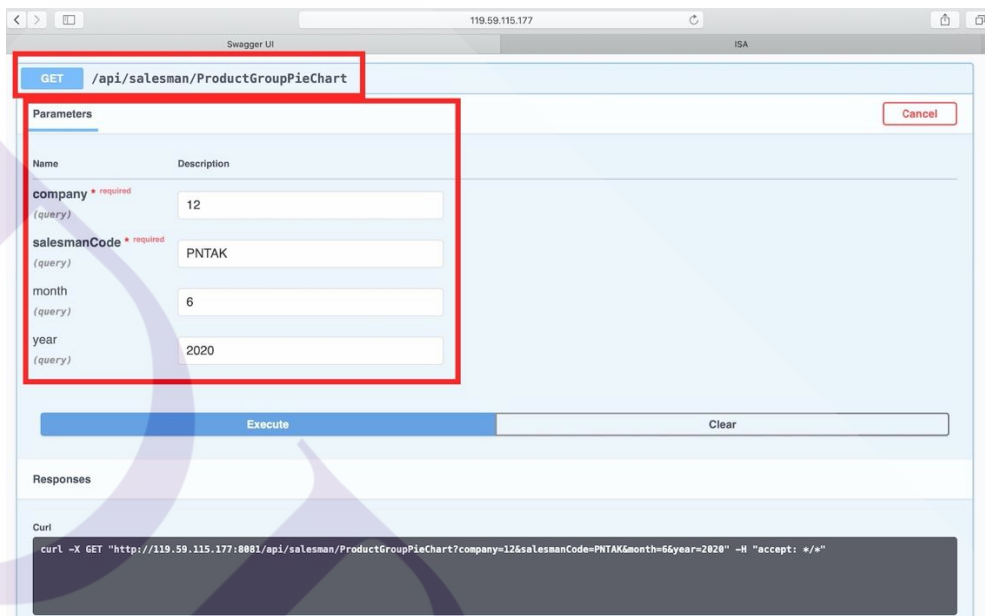


ภาพที่ 4.28 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

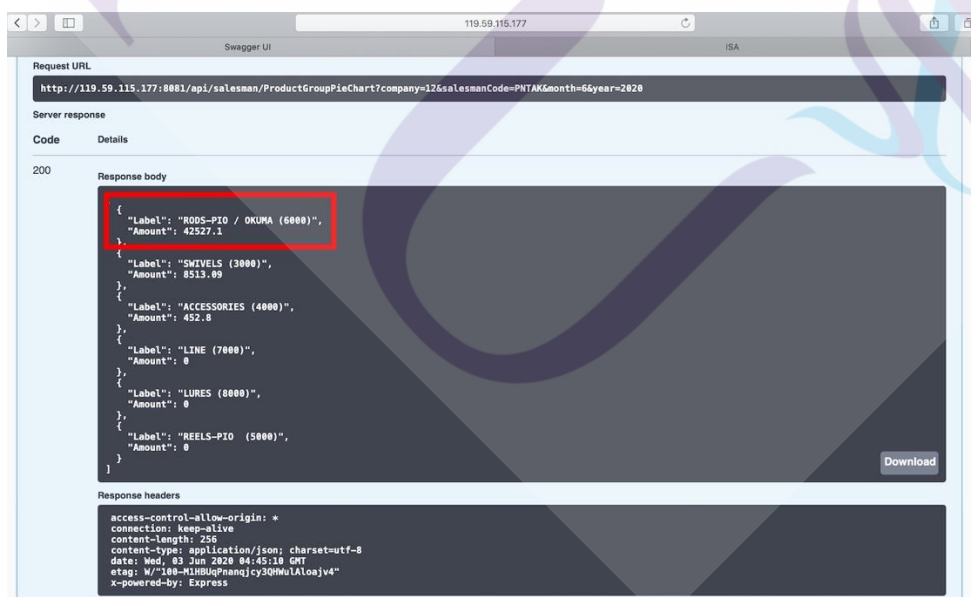
4.2.15 กรณีทดสอบ TC15

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.15.1 ทดสอบเรียกเซอ์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

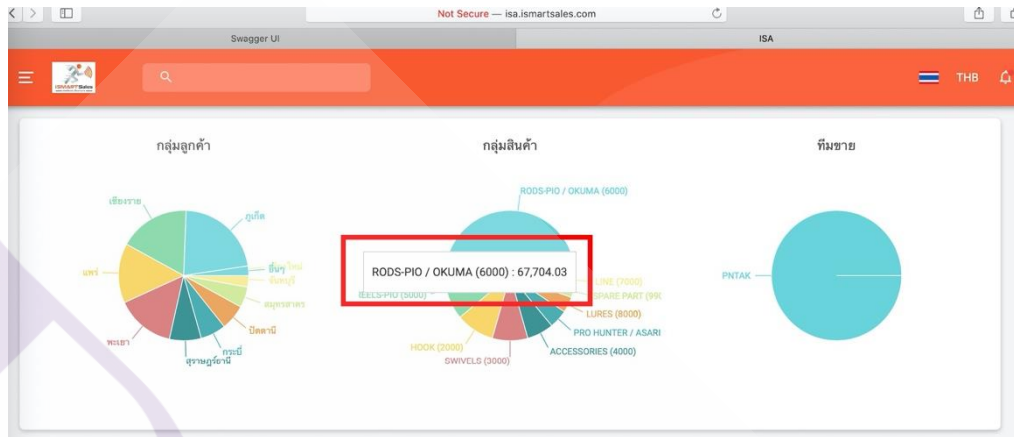


ภาพที่ 4.29 กรณีทดสอบ TC15



ภาพที่ 4.29 (ต่อ)

4.2.15.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

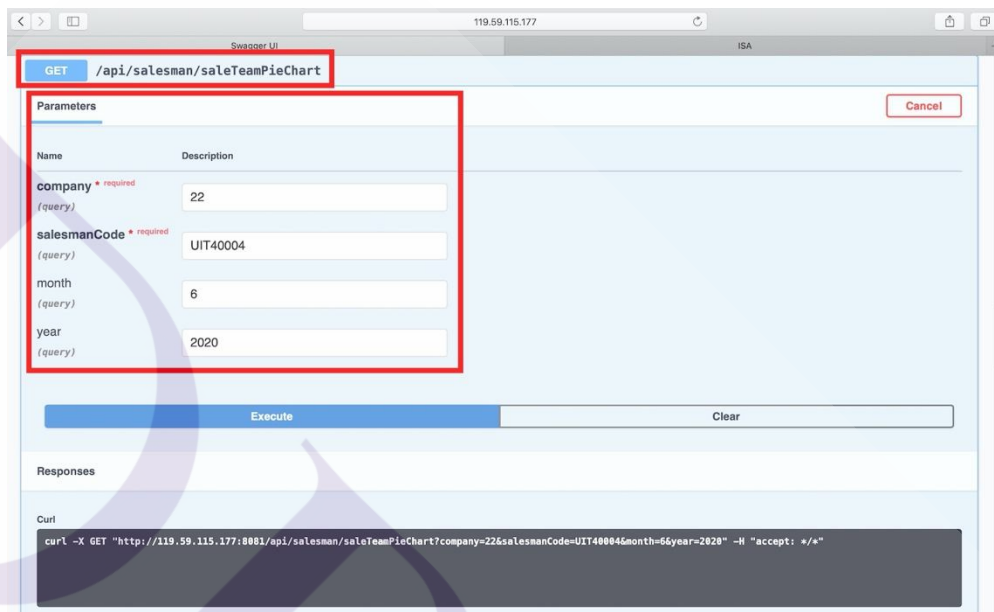


ภาพที่ 4.30 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

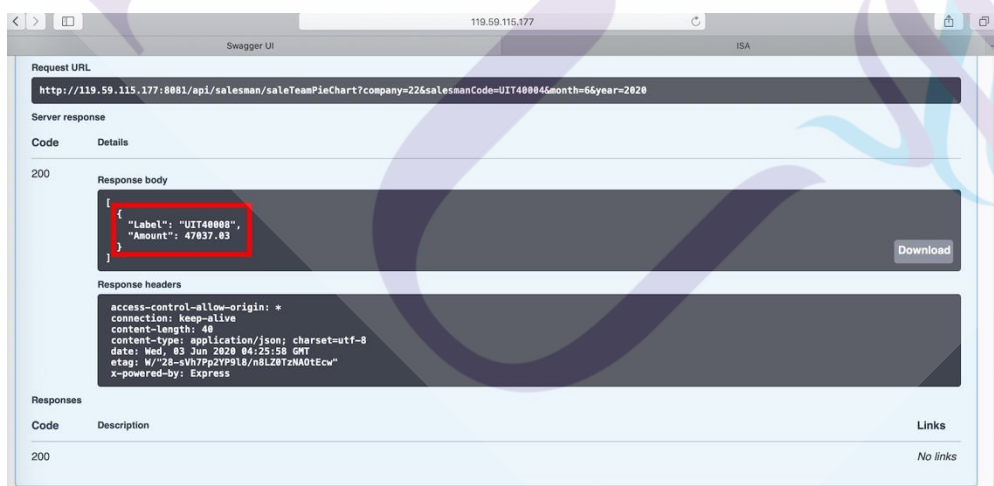
4.2.16 กรณีทดสอบ TC16

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.16.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

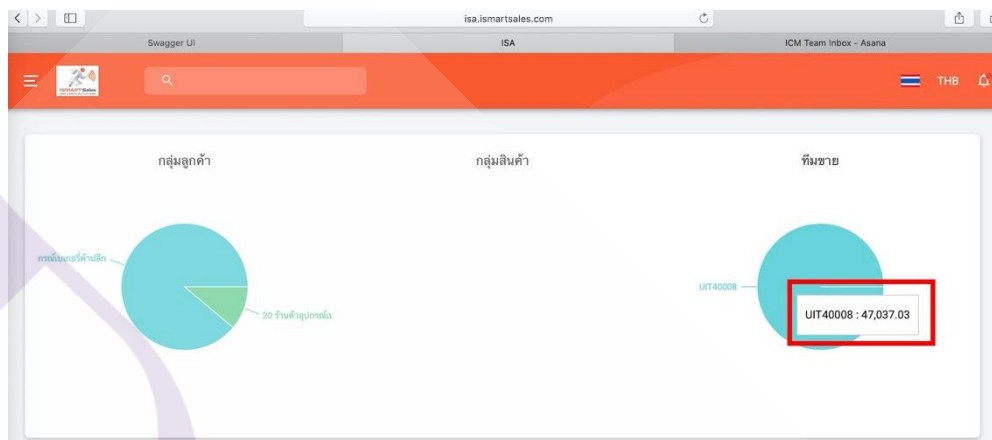


ภาพที่ 4.31 กรณีทดสอบ TC16



ภาพที่ 4.31 (ต่อ)

4.2.16.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

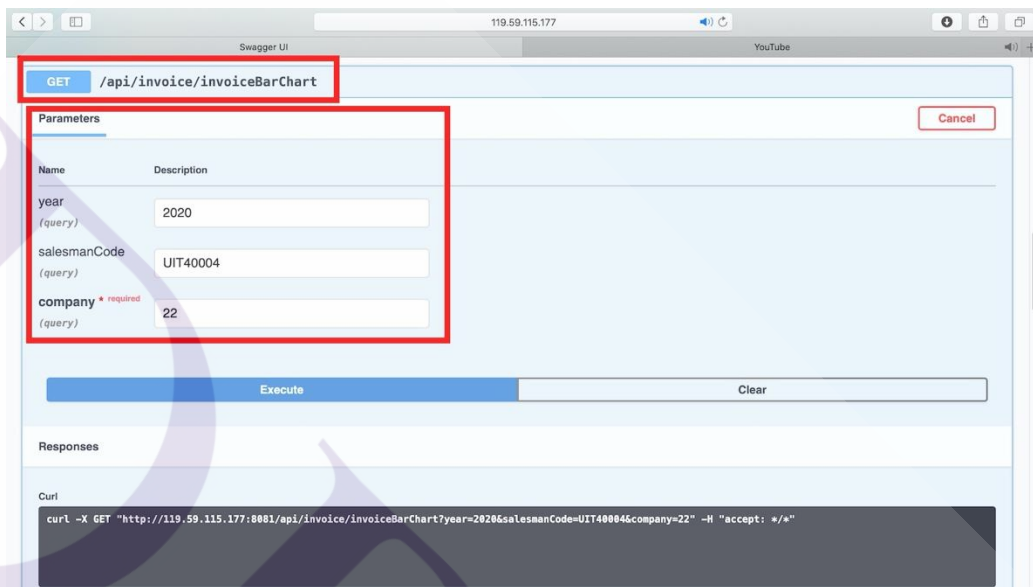


ภาพที่ 4.32 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

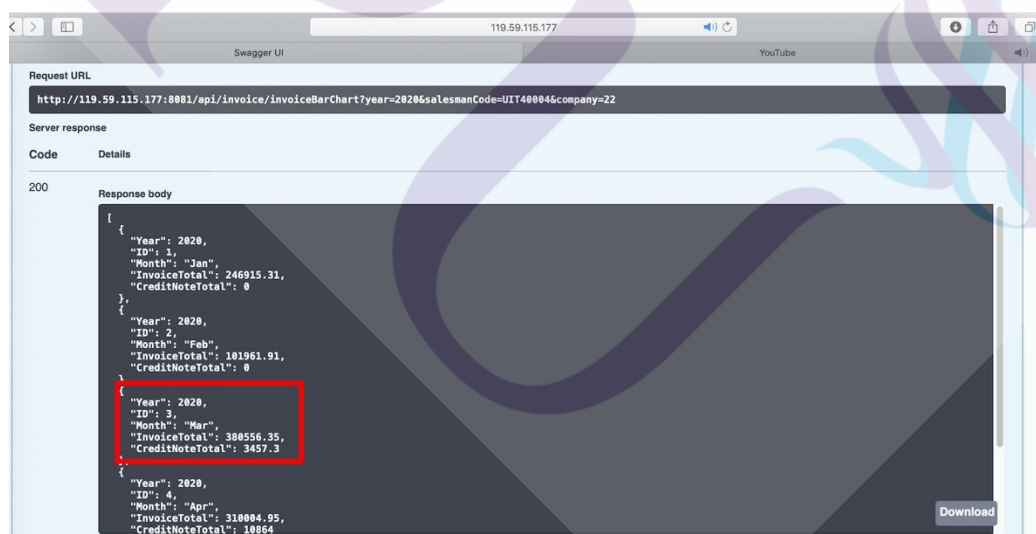
4.2.17 กรณีทดสอบ TC17

สำหรับผลการพัฒนาระบบในส่วนของคุณข้อมูล KPI มีดังนี้

4.2.17.1 ทดสอบเรียกเซอ์วิสด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

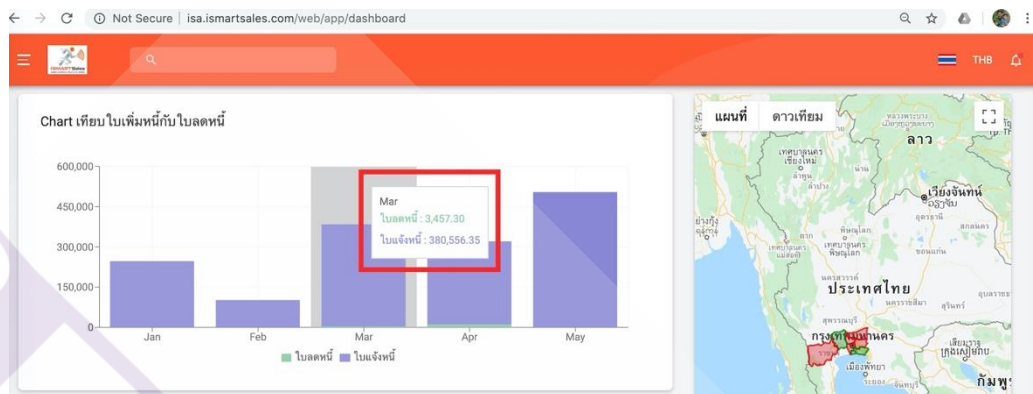


ภาพที่ 4.33 กรณีทดสอบ TC17



ภาพที่ 4.33 (ต่อ)

4.2.17.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

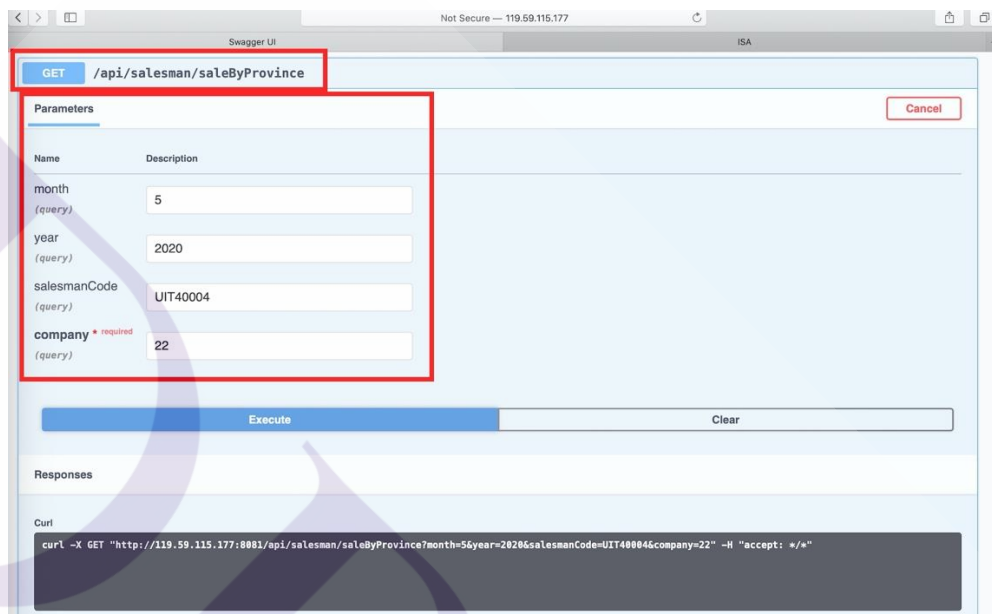


ภาพที่ 4.34 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

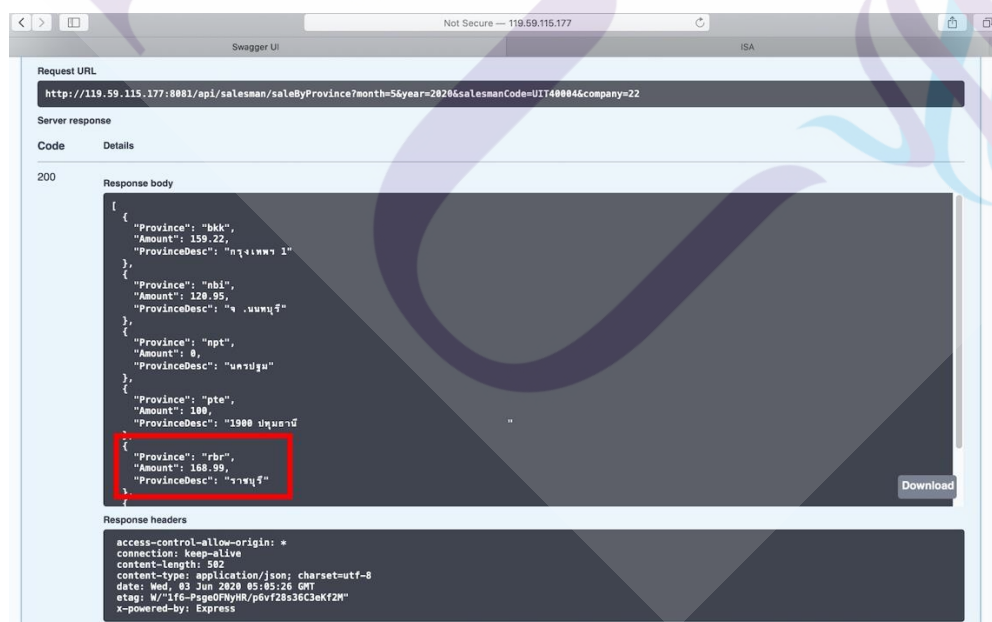
4.2.18 กรณีทดสอบ TC18

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.18.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

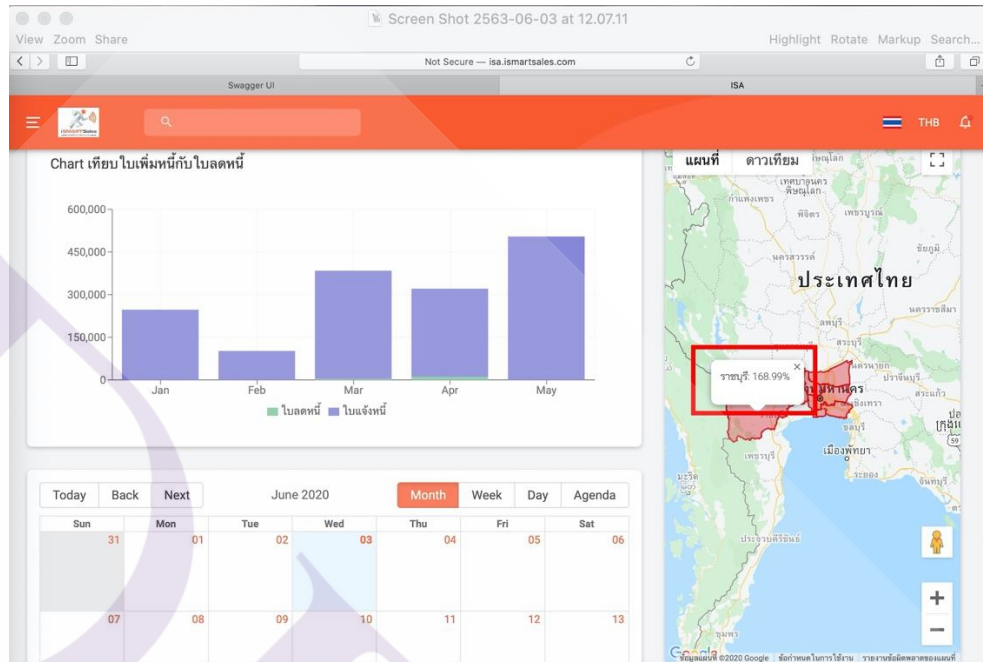


ภาพที่ 4.35 กรณีทดสอบ TC18



ภาพที่ 4.35 (ต่อ)

4.2.18.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

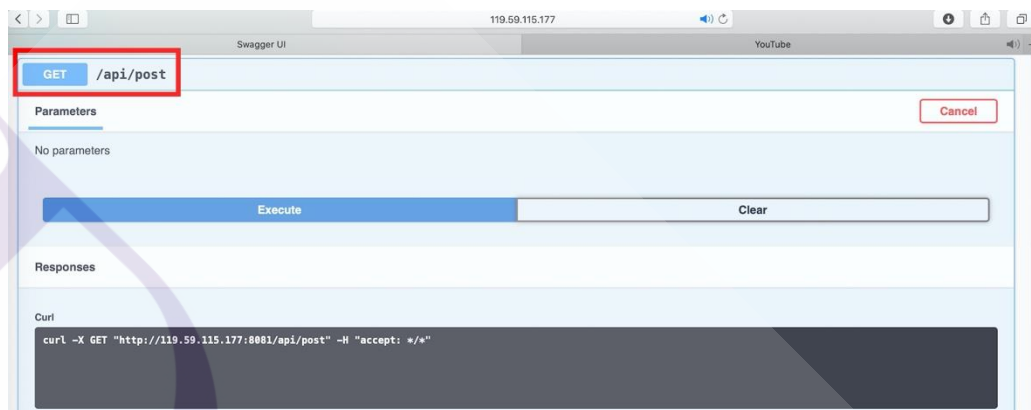


ภาพที่ 4.36 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

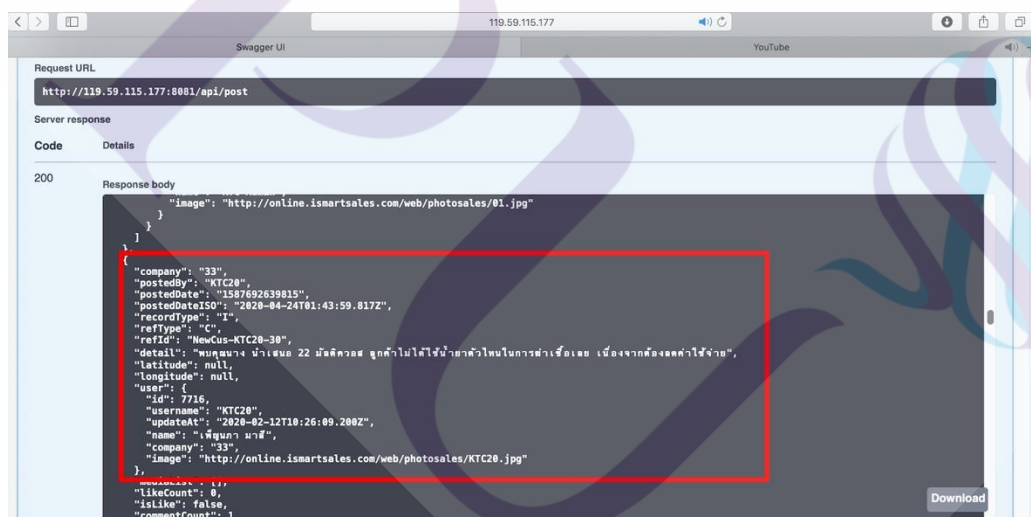
4.2.19 กรณีทดสอบ TC19

สำหรับผลการพัฒนาระบบในส่วนของคุณสมบัติ KPI มีดังนี้

4.2.19.1 ทดสอบเรียกเซิร์ฟเวอร์ด้วย SWAGGER ได้ผลลัพธ์ ดังนี้

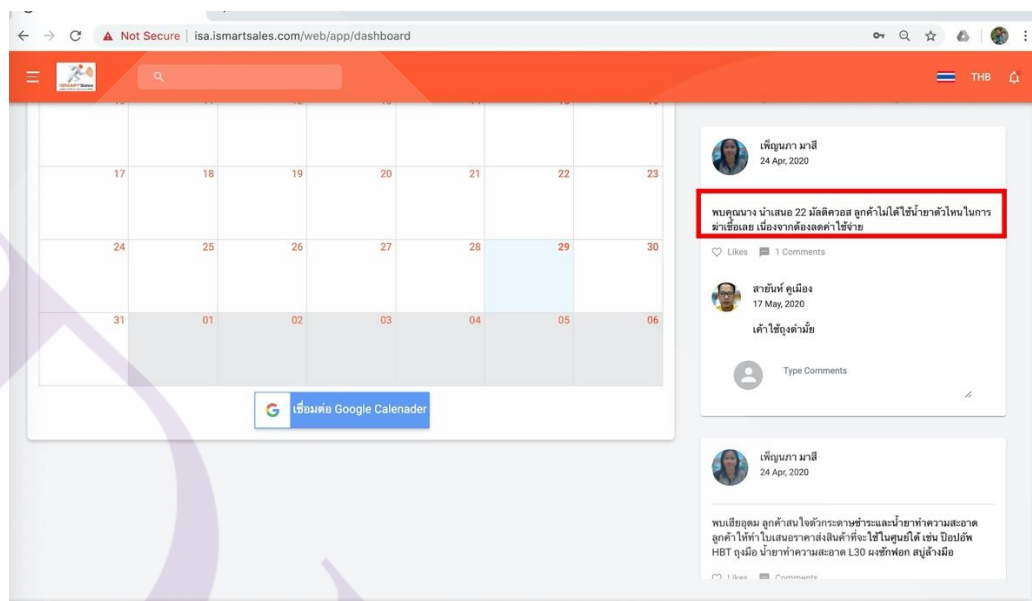


ภาพที่ 4.37 กรณีทดสอบ TC19



ภาพที่ 4.37 (ต่อ)

4.2.19.2 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน



ภาพที่ 4.38 ผลลัพธ์จากการ Integrate เข้ากับแอปพลิเคชัน

บทที่ 5

สรุปอภิปรายผลการศึกษา และข้อเสนอแนะ

ในบทนี้จะกล่าวถึงข้อสรุปจากการออกแบบ และพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย รวมทั้งปัญหาและอุปสรรค และข้อเสนอต่าง ๆ ดัง รายละเอียดต่อไปนี้

5.1 สรุป และอภิปรายผล

ในการดำเนินการเพื่อการออกแบบ และพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย จากการวิเคราะห์รวบรวมข้อมูล และออกแบบระบบรวมถึง ขั้นตอนการพัฒนา ระบบ ทำให้ได้ระบบที่ผู้ใช้สามารถวิเคราะห์ข้อมูลได้หลากหลายมิติ รวมถึงความถูกต้อง และรวดเร็วในการใช้งาน

จากการออกแบบ และพัฒนาระบบรวมถึงการนำมาใช้งานถือว่าประสบความสำเร็จเว็บเซอร์วิสสามารถทำงานได้และตรงตามวัตถุประสงค์ของการจัดทำ และตรงตามความต้องการของผู้ใช้ที่ได้ทำการสำรวจความต้องการไว้ในตอนต้น โดยมีรายละเอียดดังนี้

5.1.1 ผู้วิจัยได้ศึกษาปัญหา และได้ศึกษาเทคโนโลยีที่เกี่ยวข้อง และได้นำเทคโนโลยีที่เหมาะสมมาประยุกต์ใช้กับการพัฒนาเว็บเซอร์วิสสำหรับระบบการตรวจสอบประสิทธิภาพการขาย

5.1.2 ผู้วิจัยได้นำความรู้ และทฤษฎีที่เกี่ยวข้องที่ได้ศึกษามาทำการวิเคราะห์ และออกแบบระบบงาน ทำให้เว็บเซอร์วิสที่พัฒนาขึ้นมีประสิทธิภาพ และได้ระบบที่ตรงกับ ความต้องการของผู้ใช้งานในปัจจุบัน

5.1.3 เว็บเซอร์วิสที่พัฒนาขึ้นสามารถทำงานร่วมกับ แอปพลิเคชันในฝั่งไคลเอนต์ได้เป็นอย่างดี สามารถรับส่งข้อมูลไปมาระหว่างกันได้ และจากการนำ Framework ในการออกแบบ API อย่าง Open API Specification เข้ามาช่วยในการพัฒนาระบบพบว่าสามารถอำนวยความสะดวกในการพัฒนาระบบในฝั่งไคลเอนต์ได้เป็นอย่างดี สามารถลดข้อผิดพลาดในการทำงานร่วมกัน ระหว่าง เซอร์เวอร์ และไคลเอนต์ และลดเวลาในการพัฒนาได้มากเนื่องจากโค้ด ในส่วนของการเชื่อมต่อกับ API สามารถใช้ Swagger Codegen ช่วยสร้างออกมาได้

5.2 ปัญหา และอุปสรรค

5.2.1 เนื่องจากความต้องการในการวิเคราะห์ที่มีมุมมองที่หลากหลาย และมีมิติในการดูหลายด้าน ทำให้มีการปรับเปลี่ยน เพิ่มเติมความต้องการ ระหว่างการพัฒนา ทำให้ใน บางครั้งต้องกลับไปทบทวน และออกแบบระบบใหม่ในส่วนที่มีการเปลี่ยนแปลงทำให้ส่งผลกระทบต่อ การพัฒนา ระบบทั้งในส่วนของเซิร์ฟเวอร์ และ โคลเอนท์ ทำให้การพัฒนาล่าช้าออกไปไม่ตรงตาม แผนการที่วางไว้

5.2.2 เนื่องจากมีการนำ Open API Specification มาใช้ในการออกแบบ API และช่วยในการพัฒนา ระบบในฝั่ง โคลเอนท์ ซึ่งถือว่าเป็นเรื่องที่ค่อนข้างใหม่ ทำให้นักพัฒนาทั้งในฝั่งเซิร์ฟเวอร์ และ โคลเอนท์ต้องใช้เวลาในการศึกษาลงฝึก ลองถูกอยู่พอสมควรในช่วงแรก

5.2.3 และเนื่องจากการนำ Open API Specification มาใช้ และมีการส่งไฟล์ API Spec นี้ให้กับนักพัฒนา ระบบในฝั่ง โคลเอนท์นำไปใช้กับ เครื่องมือ Swagger Codegen ในการสร้างชุดคำสั่ง ในการเรียกใช้ API ดังนั้นเมื่อมีการเปลี่ยนแปลงแก้ไขหรือเพิ่ม API ใหม่รวมถึงเมื่อมีการสร้าง Spec ที่ผิดพลาดจำเป็นต้อง มีการแก้ไขไฟล์ Spec นี้ และส่งให้นักพัฒนาในฝั่ง โคลเอนท์ใหม่ทุกครั้ง เพื่อทำการสร้างชุดคำสั่งหรือโค้ดชุดใหม่ขึ้นมาทั้งหมด

5.3 ข้อเสนอแนะ

จากการทดสอบโดยผู้ใช้งานแต่ละกลุ่ม ทั้งผู้ใช้ระดับปฏิบัติการ รวมถึงผู้ใช้ระดับหัวหน้างาน ได้รับคำแนะนำ รวมถึงข้อเสนอแนะและส่วนที่ควรปรับปรุงของระบบมีดังนี้

5.3.1 เพื่อความสะดวกของผู้ใช้งาน ควรมีการทำ Single Sign On (SSO) จากระบบหลัก “iSmartSales” ให้ผู้ใช้สามารถเรียกใช้ได้รวดเร็วยิ่งขึ้น

5.3.2 สำหรับการเข้าใช้ตรงในระบบการตรวจสอบประสิทธิภาพการขาย ควรจัดให้มีการ Sign In ด้วย Social Account เช่น Facebook Google เป็นต้น เพื่อให้สะดวกรวดเร็วในการเข้าสู่ระบบ

5.3.3 เพื่อให้การวิเคราะห์ครอบคลุมการบริหารจัดการแบบ 360 องศา ควรจัดให้มีมิติการวิเคราะห์ทางการเงิน รวมถึงการบริหารบุคคลด้วย

5.3.4 จากปริมาณข้อมูลที่มีในระบบมากพอสมควร หากสามารถต่อยอดไปสู่การทำ AI (Artificial Intelligence) ก็จะทำให้ผู้ใช้สามารถได้รับคำแนะนำที่มีประโยชน์มากขึ้น



บรรณานุกรม

บรรณานุกรม

ภาษาไทย

ศหรัฐฐ(Saharat). *Service-OrientedArchitecture:SOA*. สืบค้น 8 กันยายน 2562 จาก

<http://saharat2537.blogspot.com/2017/03/soa-service-oriented-architecture.html>

Suphakit Annopornchai. *เว็บเซอร์วิส*. สืบค้น 8 กันยายน 2562 จาก

<https://saixiii.com/what-is-webservice/>

Intception. *NEST JS*. สืบค้น 8 กันยายน 2562 จาก

<https://blog.intception.me/dev/nestjs/write-backend-with-nestjs.html#service-หรือ-provider>

ภาษาต่างประเทศ

Open API Initiative. *About Open API Initiative* [Online]. Retrieve September 8, 2019 From

<https://www.openapis.org/about> [2017, June 22].

OpenAPI Specification. *The OpenAPI Specification* [Online]. Retrieve September 8, 2019 From

<https://github.com/OAI/OpenAPI-Specification> [2017, June 22].

SmartBear Software. *The Swagger documentation* [Online]. Retrieve September 8, 2019 From

<http://swagger.io/docs/> [2017, June 22].



ภาคผนวก

ภาคผนวก ก
พจนานุกรมข้อมูล (Data Dictionary)



ตารางที่ ก.1 ตาราง ActiveCustomer

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	TotalCustomer	INTEGER	จำนวนลูกค้าทั้งหมด	
4.	ActiveCustomer	INTEGER	จำนวนลูกค้าที่มีความเคลื่อนไหว	

ตารางที่ ก.2 ตาราง ClientVisit

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	TotalCustomer	INTEGER	จำนวนลูกค้าทั้งหมด	
4.	VisitCustomer	INTEGER	จำนวนครั้งที่เข้าเยี่ยมลูกค้า	

ตารางที่ ก.3 ตาราง Customer_SalesSummaries

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	INTEGER	ปี	
4.	Qtr	INTEGER	ไตรมาส	
5.	MonthID	INTEGER	เดือน	
6.	CustomerCode	NVARCHAR	รหัสลูกค้า	
7.	InvQty	INTEGER	จำนวนใบแจ้งหนี้	
8.	InvAmt	DECIMAL	จำนวนยอดเงินจากใบแจ้งหนี้	
9.	SoQty	INTEGER	จำนวนใบสั่งขาย	
10.	SoAmt	DECIMAL	จำนวนยอดเงินจากใบสั่งขาย	

ตารางที่ ก.3 (ต่อ)

11.	NumOfBill	INTEGER	จำนวนการวางบิล	
12.	NumOfCheckIn	INTEGER	จำนวนการเข้าเยี่ยมชมลูกค้า	
13.	NumOfPlan	INTEGER	จำนวนแผนการเข้าเยี่ยมชม	

ตารางที่ ก.4 ตาราง Customer_Stat

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	INTEGER	ปี	
4.	Qtr	INTEGER	ไตรมาส	
5.	MonthID	INTEGER	เดือน	
6.	TargetNewCus	INTEGER	เป้าหมายลูกค้าใหม่	
7.	TotalNewCus	INTEGER	จำนวนลูกค้าใหม่	
8.	ProspectCus	INTEGER	จำนวนลูกค้าที่คาดหวัง	
9.	NewERPCus	INTEGER	จำนวนลูกค้าใหม่จาก ERP	
10.	TotalERPCus	INTEGER	จำนวนลูกค้าทั้งหมด	
11.	ReEntryERPCus	INTEGER	จำนวนลูกค้าที่กลับมาซื้อสินค้าอีกครั้ง	
12.	ActiveERPCus	INTEGER	จำนวนลูกค้าที่มีความเคลื่อนไหว	

ตารางที่ ก.5 ตาราง Customer_Total_AR

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	CustomerCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	NotDue	DECIMAL	ยอดรวมลูกหนี้ไม่เกินกำหนด	
4.	Due	DECIMAL	ยอดรวมลูกหนี้ที่อยู่ในกำหนด	
5.	OverDue1	DECIMAL	ยอดรวมลูกหนี้ เกินระยะเวลาที่ กำหนด 1	
6.	OverDue2	DECIMAL	ยอดรวมลูกหนี้ เกินระยะเวลาที่ กำหนด 2	
7.	OverDue3	DECIMAL	ยอดรวมลูกหนี้ เกินระยะเวลาที่ กำหนด 3	

ตารางที่ ก.6 ตาราง Module

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	ModuleId	INTEGER	ชื่อโมดูล	PK
2.	ModuleParent	INTEGER	รายละเอียดของโมดูล	
3.	ModuleName	NVARCHAR	ชื่อคอมโพเน้นของโมดูล	
4.	ModuleDetail	NVARCHAR	พาทช์ของโมดูล	
5.	ModuleComponent	NVARCHAR	ระดับของโมดูล	
6.	ModulePath	NVARCHAR	ลำดับการแสดงผล	
7.	ModuleLevel	INTEGER	ประเภทของโมดูล	
8.	ModuleSort	INTEGER	ชื่อโมดูล	
9.	ModuleType	NVARCHAR	รายละเอียดของโมดูล	

ตารางที่ ก.7 ตาราง Post

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	PostedBy	NVARCHAR	ผู้ทำการบันทึกข้อมูล	PK
3.	RecordType	NVARCHAR	ประเภทการบันทึกข้อมูล	PK
4.	PostedDate	INTEGER	วันที่ทำการบันทึกข้อมูล	PK
5.	RefType	NVARCHAR	ประเภทผู้อ้างอิง	
6.	RefId	NVARCHAR	รหัสผู้อ้างอิง	
7.	Detail	NVARCHAR	ข้อความที่ทำการบันทึกข้อมูล	
8.	Latitude	DECIMAL	ละติจูด	
9.	Longitude	DECIMAL	ลองจิจูด	

ตารางที่ ก.8 ตาราง PostComment

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Id	NVARCHAR	รหัสแสดงความคิดเห็น	PK
2.	Company	NVARCHAR	รหัสบริษัท	PK
3.	PostedBy	NVARCHAR	ผู้ทำการบันทึกข้อมูล	PK
4.	RecordType	NVARCHAR	ประเภทการบันทึกข้อมูล	PK
5.	PostedDate	INTEGER	วันที่ทำการบันทึกข้อมูล	PK
6.	Comment	NVARCHAR	ข้อความแสดงความคิดเห็น	
7.	CommentedBy	NVARCHAR	ผู้แสดงความคิดเห็น	
8.	CreatedDate	DATETIME	วันที่ที่แสดงความคิดเห็น	

ตารางที่ ก.9 ตาราง PostLike

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	PostedBy	NVARCHAR	ผู้ทำการบันทึกข้อมูล	PK
3.	RecordType	NVARCHAR	ประเภทการบันทึกข้อมูล	PK
4.	PostedDate	INTEGER	วันที่ทำการบันทึกข้อมูล	PK
5.	LikedBy	INTEGER	ผู้ทำการไลค์	
6.	CreatedDate	DATETIME	วันที่ทำการไลค์	

ตารางที่ ก.10 ตาราง PostMedia

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Id	NVARCHAR	รหัสแสดงความคิดเห็น	PK
2.	Company	NVARCHAR	รหัสบริษัท	PK
3.	PostedBy	NVARCHAR	ผู้ทำการบันทึกข้อมูล	PK
4.	RecordType	NVARCHAR	ประเภทการบันทึกข้อมูล	PK
5.	PostedDate	INTEGER	วันที่ทำการบันทึกข้อมูล	PK
6.	MediaName	NVARCHAR	ชื่อรูปภาพที่ทำการโพสต์	
7.	FileLocation	NVARCHAR	ชื่อที่อยู่ของไฟล์	
8.	FileURLPath	NVARCHAR	ชื่อ URL ของที่อยู่ไฟล์	

ตารางที่ ก.11 ตาราง Province

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	ProvinceCode	NVARCHAR	รหัสจังหวัด	PK
3.	ISAProvinceCode	NVARCHAR	รหัสจังหวัดของโปรเจก ISA	PK

ตารางที่ ก.12 ตาราง Report_Invoice

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	NVARCHAR	ปี	PK
4.	MonthID	NVARCHAR	เดือน	PK
5.	Month	INTEGER	ชื่อเดือน	
6.	AmountInvoice	NVARCHAR	ยอดจากใบแจ้งหนี้	
7.	AmountCreditNote	NVARCHAR	ยอดจากใบลดหนี้	

ตารางที่ ก.13 ตาราง Report_Order_Fulfillment

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	CustomerCode	NVARCHAR	รหัสลูกค้า	PK
3.	ProductCode	NVARCHAR	รหัสสินค้า	PK
4.	Month1Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 11 เดือน	
5.	Month1Amt	DECIMAL	จำนวนยอดก่อนหน้า 11 เดือน	
6.	Month2Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 10 เดือน	
7.	Month2Amt	DECIMAL	จำนวนยอดก่อนหน้า 10 เดือน	
8.	Month3Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 9 เดือน	
9.	Month3Amt	DECIMAL	จำนวนยอดก่อนหน้า 9 เดือน	
10.	Month4Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 8 เดือน	
11.	Month4Amt	DECIMAL	จำนวนยอดก่อนหน้า 8 เดือน	
12.	Month5Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 7 เดือน	
13.	Month5Amt	DECIMAL	จำนวนยอดก่อนหน้า 7 เดือน	
14.	Month6Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 6 เดือน	
15.	Month6Amt	DECIMAL	จำนวนยอดก่อนหน้า 6 เดือน	
16.	Month7Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 5 เดือน	
17.	Month7Amt	DECIMAL	จำนวนยอดก่อนหน้า 5 เดือน	
18.	Month8Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 4 เดือน	
19.	Month8Amt	DECIMAL	จำนวนยอดก่อนหน้า 4 เดือน	
20.	Month9Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 3 เดือน	
21.	Month9Amt	DECIMAL	จำนวนยอดก่อนหน้า 3 เดือน	
22.	Month10Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 2 เดือน	
23.	Month10Amt	DECIMAL	จำนวนยอดก่อนหน้า 2 เดือน	
24.	Month11Qty	INTEGER	จำนวนออเดอร์ก่อนหน้า 1 เดือน	

ตารางที่ ก.13 (ต่อ)

25.	Month11Amt	DECIMAL	จำนวนยอดก่อนหน้า 1 เดือน	
26.	Month12Qty	INTEGER	จำนวนออเดอร์เดือนปัจจุบัน	
27.	Month12Amt	DECIMAL	จำนวนยอดเดือนปัจจุบัน	

ตารางที่ ก.14 ตาราง Report_SalesArea

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	INTEGER	ปี	PK
4.	MonthID	INTEGER	รหัสเดือน	PK
5.	Month	NVARCHAR	ชื่อเดือน	
6.	ThisMonthLastYear	DECIMAL	ยอดเดือนนี้ของปีก่อน	
7.	ThisMonth	DECIMAL	ยอดเดือนนี้ปีนี้	
8.	Target	DECIMAL	เป้าหมาย	

ตารางที่ ก.15 ตาราง Report_SaleSummary

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	INTEGER	ปี	PK
4.	Month	INTEGER	เดือน	PK
5.	Monthly	DECIMAL	ยอดรายเดือน	
6.	InvoiceOrder	DECIMAL	ยอดใบแจ้งหนี้	
7.	ReturnOrder	DECIMAL	ยอดคืนสินค้า	
8.	RejectOrder	DECIMAL	ยอดยกเลิกการสั่งซื้อ	

ตารางที่ ก.16 ตาราง Report_StockCard

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	CustomerCode	NVARCHAR	รหัสลูกค้า	PK
3.	ProductCode	NVARCHAR	รหัสสินค้า	PK
4.	Year	INTEGER	ปี	PK
5.	Month01OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 11 เดือน	
6.	Month01StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 11 เดือน	
7.	Month01StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 11 เดือน	
8.	Month02OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 10 เดือน	
9.	Month02StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 10 เดือน	
10.	Month02StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 10 เดือน	
11.	Month03OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 9 เดือน	
12.	Month03StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 9 เดือน	
13.	Month03StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 9 เดือน	
14.	Month04OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 8 เดือน	
15.	Month04StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 8 เดือน	

ตารางที่ ก.16 (ต่อ)

16.	Month04StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 8 เดือน	
17.	Month05OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 7 เดือน	
18.	Month05StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 7 เดือน	
19.	Month05StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 7 เดือน	
20.	Month06OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 6 เดือน	
21.	Month06StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 6 เดือน	
22.	Month06StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 6 เดือน	
23.	Month07OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 5 เดือน	
24.	Month07StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 5 เดือน	
25.	Month07StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 5 เดือน	
26.	Month08OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 4 เดือน	
27.	Month08StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 4 เดือน	
28.	Month08StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 4 เดือน	
29.	Month09OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 3 เดือน	

ตารางที่ ก.16 (ต่อ)

30.	Month09StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 3 เดือน	
31.	Month09StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 3 เดือน	
32.	Month10OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 2 เดือน	
33.	Month10StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 2 เดือน	
34.	Month10StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 2 เดือน	
35.	Month11OpenQty	DECIMAL	จำนวนสินค้ายกมาก่อนหน้า 1 เดือน	
36.	Month11StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อก่อนหน้า 1 เดือน	
37.	Month11StockOutQty	DECIMAL	จำนวนสินค้าที่ขายก่อนหน้า 1 เดือน	
38.	Month12OpenQty	DECIMAL	จำนวนสินค้ายกมาเดือนปัจจุบัน	
39.	Month12StockInQty	DECIMAL	จำนวนสินค้าที่ซื้อเดือนปัจจุบัน	
40.	Month12StockOutQty	DECIMAL	จำนวนสินค้าที่ขายเดือนปัจจุบัน	

ตารางที่ ก.17 ตาราง Sale_trx

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	INTEGER	ปี	PK
4.	Month	INTEGER	เดือน	PK
5.	Type	NVARCHAR	ชื่อประเภทของข้อมูล	PK
6.	Amount	DECIMAL	ยอดของประเภทแรก	
7.	Total	DECIMAL	ยอดของประเภทที่สอง	
8.	PercentAmount	DECIMAL	เปอร์เซ็นต์ยอดของประเภทแรก	
9.	TotalPercent	DECIMAL	เปอร์เซ็นต์ยอดของประเภทที่สอง	

ตารางที่ ก.18 ตาราง SalesByProvince

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Company	NVARCHAR	รหัสบริษัท	PK
2.	SalesmanCode	NVARCHAR	รหัสพนักงานขาย	PK
3.	Year	INTEGER	ปี	PK
4.	Month	INTEGER	เดือน	PK
5.	Province	NVARCHAR	รหัสจังหวัด	PK
6.	Amount	DECIMAL	เปอร์เซ็นต์ของยอดขาย	

ตารางที่ ก.19 ตาราง SalesPropotion

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	Type	NVARCHAR	ประเภท	PK
2.	Company	NVARCHAR	รหัสบริษัท	PK
3.	Salesmancode	NVARCHAR	รหัสพนักงานขาย	PK
4.	YearID	INTEGER	ปี	PK
5.	MonthID	INTEGER	เดือน	PK
6.	Label	NVARCHAR	รายละเอียดสัดส่วนการขาย	PK
7.	Amount	DECIMAL	ยอดขาย	

ตารางที่ ก.20 ตาราง UserPermission

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	UserGroup	INTEGER	กลุ่มผู้ใช้งาน	PK
2.	Module	INTEGER	รหัสโมดูล	PK
3.	CreatedBy	NVARCHAR	ผู้ที่บันทึกข้อมูล	PK
4.	CreatedAt	DATETIME	วันที่บันทึกข้อมูล	PK
5.	UpdatedBy	NVARCHAR	ผู้แก้ไขข้อมูล	PK
6.	UpdatedAt	DATETIME	วันที่แก้ไขข้อมูล	PK

ตารางที่ ก.21 ตาราง WebUser

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	ID	INTEGER	รหัสผู้ใช้งาน	PK
2.	MODIFY	DATETIME	วันที่ทำการแก้ไข	PK
3.	Username	NVARCHAR	ชื่อผู้ใช้งาน	PK
4.	Password	NVARCHAR	รหัสผ่าน	PK
5.	Name	NVARCHAR	ชื่อ - สกุลผู้ใช้งาน	
6.	Role	NVARCHAR	บทบาทของผู้ใช้งาน	
7.	Division	NVARCHAR	บริษัทของผู้ใช้งาน	
8.	SalesOrg	NVARCHAR	ทีมขาย	
9.	Channel	NVARCHAR	ช่องทางของการขาย	
10.	SIGN	IMAGE	ลายเซ็น	
11.	UserType	NVARCHAR	ประเภทของผู้ใช้งาน	

ตารางที่ ก.22 ตาราง WebUser_Application

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	WebUser	NVARCHAR	ชื่อผู้ใช้งานของหัวหน้าทีม	PK
2.	Application_ID	NVARCHAR	ชื่อผู้ใช้งานระบบ	PK

ตารางที่ ก.23 ตาราง WebUser_Detail

ลำดับ	แอทริบิวต์	ชนิด	คำอธิบาย	คีย์
1.	WebUser	INTEGER	ชื่อผู้ใช้งานระบบ	PK
2.	ImageProfile	DATETIME	ชื่อรูปภาพ	
3.	ImageProfilePath	NVARCHAR	ที่อยู่ของรูปภาพ	
4.	GoogleToken	NVARCHAR	รหัสการเข้าใช้งาน Google API	
5.	LineToken	NVARCHAR	รหัสการเข้าใช้งาน Line API	
6.	CreatedAt	NVARCHAR	วันที่บันทึกข้อมูล	
7.	UpdatedAt	NVARCHAR	วันที่แก้ไขข้อมูล	

ประวัติผู้เขียน

ชื่อ-นามสกุล

ประวัติการศึกษา

ตำแหน่ง และสถานที่ทำงานปัจจุบัน

เศรษฐพงศ์ กตคุณไพศาล

ปีการศึกษา 2536

สำเร็จการศึกษาระดับปริญญาตรี

สาขาวิทยาการคอมพิวเตอร์

มหาวิทยาลัยศรีนครินทรวิโรฒ ประสานมิตร

กรรมการผู้จัดการ

บริษัท ไอซีเอ็ม สمارท์ โซลูชั่น จำกัด

