



รายงานสหกิจศึกษาฉบับสมบูรณ์

แอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้า
Order Service and Delivering Application for Grocery Store

ณัฐภัทร สุวิสิษฐ์
NATTHAPATARA SUWISITH

สาขาวิชาวิศวกรรมสารสนเทศ
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561



รายงานสหกิจศึกษาฉบับสมบูรณ์

แอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้า

Order Service and Delivering Application for Grocery Store

ณัฐภัทร สุวิสิษฐ์

NATTHAPATARA SUWISITH

สาขาวิชาวิศวกรรมสารสนเทศ

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา แอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้า

ชื่อ-สกุล นักศึกษา นายณัฐภัทร สุวิศิษฐ์

คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร.พิกุลแก้ว ดั่งติสานนท์

ชื่อ-สกุล ผู้นิเทศงาน นายคุณาวุฒิ เจริญรุ่ง

สถานประกอบการ บริษัท โกซอฟท์ (ประเทศไทย) จำกัด

บทคัดย่อ

ปริญญานิพนธ์นี้ได้จัดทำขึ้นมาเพื่อศึกษาและพัฒนาแอปพลิเคชันที่สามารถรับรายการสินค้าของร้านสะดวกซื้อที่ส่งเข้ามาผ่านทางเว็บแอปพลิเคชัน โดยในระบบนั้นจะมีส่วนเว็บแอปพลิเคชันที่เป็นหน้าที่ผู้ใช้งานทำการสั่งซื้อสินค้าต่างๆ เมื่อลูกค้าสั่งซื้อสินค้าเสร็จสิ้นรายการที่ส่งจะถูกส่งมายังพนักงานร้านสะดวกซื้อ จากนั้นจะแสดงให้พนักงานร้านสะดวกซื้อในรูปแบบแอปพลิเคชันบนโทรศัพท์มือถือระบบแอนดรอยด์ แล้วจึงนำมาจัดส่งถึงบ้านหรือสำหรับผู้ที่ต้องการมารับที่ร้าน ซึ่งในปฏิบัติงานครั้งนี้ได้รับมอบหมายให้ทำในส่วนแอปพลิเคชันบนโทรศัพท์มือถือระบบแอนดรอยด์ โดยแอปพลิเคชันนี้จะทำการรับรายการสินค้ามาจากลูกค้าที่สั่งซื้อรายการสินค้าของร้านสะดวกซื้อผ่านเว็บแอปพลิเคชัน ว่าในรายการนั้นมีสินค้าใดบ้าง มีจำนวนกี่ชิ้น ราคาของสินค้านั้นราคาเท่าไร และรายการสินค้าจะประกอบไปด้วยข้อมูลของผู้ส่งด้วย เช่น ประเภทของการชำระเงิน ที่อยู่จัดส่ง และเบอร์ติดต่อของลูกค้า ตามที่ลูกค้าทำการกรอกเข้ามา แล้วพนักงานจะทำการยอมรับรายการสินค้าหรือจะทำการยกเลิกรายการสินค้านั้น เมื่อทำการยอมรับรายการสินค้านั้นพนักงานก็จะทำการจัดเตรียมของเพื่อทำการส่งไปยังลูกค้า โดยมีการพัฒนาโดยใช้ภาษา Java และ XML ซึ่งใช้ในการเขียนแอนดรอยด์ และมีระบบ server โดยใช้ NodeJS โดยใช้ภาษา JavaScript ในการเรียกใช้ web service

คำสำคัญ : เว็บแอปพลิเคชัน, แอปพลิเคชัน, แอนดรอยด์

Cooperative Title: Order Service and Delivering Application for Grocery Store

Student intern name: Mr. Natthapatara Suwisith

Faculty: Bachelor of Engineering **Department:** Computer Engineering

Program: Information Engineering

Advisor Name: Asst. Prof. Dr. Pikulkaew Tangtisanon

Mentor Name: Kunawut Jaruenrang

Company: Gosoft (Thailand) Co.,Ltd

ABSTRACT

The objective of this project is to build an android application that users use to receive orders from customers who pass the order via the web application. After the customer completes the order, the order will be sent to the user that work as sellers in the grocery store. The users can slide through list of orders from customers who order items from web applications. The orders contains product's name, amount and price also include customers information such as payment type, address and contact. The sellers can choose to accept the orders or cancel them. If the sellers accept them, they will prepare the items and delivery to the customer. This application has been developed by Java and XML languages that are used to code android application. A server is run based on NodeJS and JavaScript language is implemented to run the web service.

Keyword: web application, application, android

กิตติกรรมประกาศ

ปริญญาานิพนธ์เล่มนี้สำเร็จลุล่วงได้ด้วยความกรุณาช่วยเหลือ แนะนำ ให้คำปรึกษา ตรวจสอบแก้ไข ข้อบกพร่องต่างๆ ด้วยความเอาใจใส่อย่างดียิ่งจาก บริษัท โกซอฟท์ (ประเทศไทย) จำกัด และพี่พนักงาน บริษัท โกซอฟท์ (ประเทศไทย) จำกัด

ข้าพเจ้าขอขอบพระคุณ บริษัท โกซอฟท์ (ประเทศไทย) จำกัด ที่ได้ให้โอกาสข้าพเจ้าในการเป็น นักศึกษาโครงการสหกิจศึกษา ช่วงระยะเวลาตั้งแต่วันที่ 6 สิงหาคม 2561 จนถึง 23 พฤศจิกายน 2561

ขอขอบพระคุณ คุณคุณาวุฒิ เจริญรุ่ง ตำแหน่ง ซีเนียร์ ซอฟต์แวร์ เอ็นจิเนียร์ (Senior Software Engineer) ของ บริษัท โกซอฟท์ (ประเทศไทย) จำกัด และพี่ๆที่อยู่ในทีม mobile ทุกคนที่ให้ความรู้ และ ถ่ายทอดประสบการณ์ เกี่ยวกับการดำเนินการพัฒนาโปรแกรมแอปพลิเคชันบนมือถือแพลตฟอร์มแอนดรอยด์ และสอนแนวคิดการพัฒนาเพื่อไว้สำหรับการต่อยอดในอนาคต

ขอขอบพระคุณ ผศ.ดร.พิกุลแก้ว ตั้งติสานนท์ อาจารย์ที่ปรึกษาที่คอยความช่วยเหลือ ให้คำชี้แนะช่วย แก้ปัญหาตลอดจนให้ความรู้ ประสบการณ์ที่ดีแก่ผู้จัดทำ และดูแลอย่างใกล้ชิดทำให้ปริญญาานิพนธ์เล่มนี้สำเร็จ สมบูรณ์

ณัฐภัทร สุวิสิษฐ์

สารบัญ

บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป.....	X
สารบัญตาราง	XI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 วิธีดำเนินการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	4
2.1 ระบบปฏิบัติการ Android	4
2.2 Android Studio	5
2.3 ภาษา Java	6
2.4 Android SDK	6
2.5 Java JDK	7
2.6 Android Support Library	7
2.7 Visual Studio Code	8
2.8 NodeJS	9
2.9 Web Service	10
2.10 RESTful หรือ REST	10
2.11 JSON	11

2.12 API	12
2.13 Database	13
2.14 SQL	13
2.15 Xampp	14
2.16 แอปพลิเคชันตัวทดลอง หรือ Prototype (Mockup)	14
2.17 ภาษา XML	16
2.18 สถาปัตยกรรมซอฟต์แวร์แบบ Model-View-Controller หรือ MVC Pattern	17
2.19 Third party library สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android	19
2.20 การออกแบบ User Experience และ User Interface (UX/UI)	20
2.21 Firebase	22
2.22 OneSignal	24
บทที่ 3 วิธีดำเนินการ	26
3.1 การวิเคราะห์ระบบ	26
3.2 โครงสร้างฐานข้อมูลระบบ (ER Diagram)	31
3.3 การพัฒนาระบบ	37
บทที่ 4 ผลการทดลอง	39
4.1 การเข้าใช้งานระบบผ่านการล็อกอิน	40
4.2 การแสดงรายการสินค้า	40
4.3 การดูข้อมูลของรายการสินค้า	41
4.4 การดูข้อมูลสินค้าที่สั่ง	41
4.5 การยกเลิกรายการสินค้า	42
4.6 การยืนยันรายการสินค้า	43
4.7 การแสดงประวัติรายการสินค้าทั้งหมด	44
4.8 การแสดงประวัติรายการสินค้าที่สำเร็จ	44
4.9 การแสดงประวัติรายการสินค้าที่ยกเลิก	45

4.10 การดูสถานะสินค้าในประวัติรายการสินค้า	46
4.11 แสดงช่วงระยะเวลาที่ระบบเปิด-ปิด	46
4.12 ตั้งเวลาเปิด-ปิดระบบ	47
4.13 แก้ไขช่วงเวลาเปิด-ปิดระบบ	48
4.14 ลบช่วงเวลาเปิด-ปิดระบบ	48
4.15 ได้รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่	50
4.16 ได้รับการแจ้งเตือนเมื่อผู้ใช้งานยกเลิกรายการสินค้า	50
4.17 ส่งการแจ้งเตือนกลับไปหาผู้ใช้งานเพื่อบอกความคืบหน้าของรายการสินค้า	51
4.18 ดูสินค้าในรายการผ่านการสแกน QR code	51
บทที่ 5 สรุปผลการทดลอง	53
5.1 สรุปผลการทดลอง	53
5.2 ปัญหาที่พบระหว่างการทดลอง	53
5.3 แนวทางการแก้ไขปัญหา	53
5.4 แนวทางการพัฒนาต่อ	53
เอกสารอ้างอิง	54
ภาคผนวก ก การทดสอบ RESTful หรือการเรียกใช้งาน web service	57
ภาคผนวก ข การทดสอบการส่งการแจ้งเตือน	67
ประวัติผู้เขียน	76

สารบัญรูป

รูปที่ 2.1 วิวัฒนาการของโลโก้ระบบปฏิบัติการ Android	4
รูปที่ 2.2 สถิติจำนวนแอปพลิเคชันบน Google Play Store	5
รูปที่ 2.3 โลโก้ Android Studio	5
รูปที่ 2.4 โลโก้ของภาษา Java	6
รูปที่ 2.5 ตัวอย่าง Android Virtual Device ที่มีมาให้ใน Android SDK	7
รูปที่ 2.6 โลโก้ของ Visual Studio Code	8
รูปที่ 2.7 ภาพตัวอย่างหน้าต่างการโปรแกรม Visual Studio Code	9
รูปที่ 2.8 ภาพโลโก้ของ NodeJS	10
รูปที่ 2.9 ภาพจำลองการทำงานของ REST	11
รูปที่ 2.10 ตัวอย่างข้อมูลรูปแบบ JSON	11
รูปที่ 2.11 ภาพตัวอย่างการให้บริการของ Google APIs	12
รูปที่ 2.12 ตัวอย่างคำสั่งภาษา SQL	13
รูปที่ 2.13 ภาพแสดง Prototype ของแอปพลิเคชัน	15
รูปที่ 2.14 ภาพตัวอย่างภาษา XML ที่ใช้ในการพัฒนาแอปพลิเคชัน	16
รูปที่ 2.15 ภาพตัวอย่างส่วนการตั้งเวลาเปิดปิดระบบ-ของแอปพลิเคชัน ที่ถูกกำหนดด้วยภาษา XML.....	17
รูปที่ 2.16 ภาพตัวอย่างการแบ่งส่วนของซอฟต์แวร์ตามสถาปัตยกรรมซอฟต์แวร์แบบ MVC ของแอปพลิเคชัน	18
รูปที่ 2.17 ภาพแสดง Workflow Diagram ของสถาปัตยกรรมซอฟต์แวร์แบบ MVC	18
รูปที่ 2.18 ภาพแสดงหลักการทำงานของซอฟต์แวร์ที่ใช้โครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC ...	19
รูปที่ 2.19 ภาพตัวอย่าง Third party library “glide”	20
รูปที่ 2.20 ภาพตัวอย่างหน้าเว็บไซต์ www.github.com ที่เป็นแหล่งรวบรวม Third party library	20
รูปที่ 2.21 ภาพตัวอย่างเปรียบเทียบการออกแบบให้ได้ User Experience ที่ดีและไม่ดี	21
รูปที่ 2.22 logo Firebase	23

รูปที่ 2.23 logo ของ OneSignal	24
รูปที่ 2.24 ตัวอย่างเว็บ OneSignal	25
รูปที่ 3.1 แผนภาพ Use Case Diagram	26
รูปที่ 3.2 Activity Diagram การยืนยันรายการสินค้า	27
รูปที่ 3.3 Activity Diagram การยกเลิกรายการสินค้า	28
รูปที่ 3.4 Activity Diagram การค้นหาสินค้าในรายการสินค้าผ่านการสแกน QR code	29
รูปที่ 3.5 Flow chart ของระบบหลัก	30
รูปที่ 3.6 แผนภาพการไหลของข้อมูล(Data flow diagram Level 0)	31
รูปที่ 3.7 แผนภาพ ER diagram	31
รูปที่ 3.8 แผนภาพทฤษฎีการวิเคราะห์และออกแบบระบบ	38
รูปที่ 4.1 การใช้งานระบบผ่านการล็อกอิน	40
รูปที่ 4.2 การแสดงรายการสินค้า	41
รูปที่ 4.3 รายละเอียดข้อมูลของรายการสินค้า.....	41
รูปที่ 4.4 การดูข้อมูลสินค้าที่สั่ง.....	42
รูปที่ 4.5 การยกเลิกรายการสินค้า.....	42
รูปที่ 4.6 การยืนยันรายการสินค้า.....	43
รูปที่ 4.7 ผลลัพธ์เมื่อทำการยืนยันสถานะของรายการสินค้าจะถูกเปลี่ยนไป และจะถูกนำไปแสดงอีกหน้า	43
รูปที่ 4.8 การแสดงประวัติรายการสินค้าทั้งหมด	44
รูปที่ 4.9 การแสดงประวัติรายการสินค้าที่สำเร็จ	45
รูปที่ 4.10 การแสดงประวัติรายการสินค้าที่ยกเลิก	45
รูปที่ 4.11 การดูสถานะสินค้าในประวัติรายการสินค้า	46
รูปที่ 4.12 แสดงช่วงระยะเวลาที่ระบบเปิด-ปิด	47
รูปที่ 4.13 ตั้งเวลาเปิดปิดระบบ-	47

รูปที่ 4.14 ตัวอย่างการตั้งเวลาเปิดปิดระบบ-	48
รูปที่ 4.15 แก๊ไขช่วงเวลาเปิด-ปิดระบบ	48
รูปที่ 4.16 ลบช่วงเวลาเปิดปิดระบบ-	49
รูปที่ 4.17 ผลลัพธ์หลังจากลบช่วงเวลาเปิดปิดระบบ-	49
รูปที่ 4.18 ได้รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่	50
รูปที่ 4.19 ได้รับการแจ้งเตือนเมื่อผู้ใช้งานยกเลิกรายการสินค้า	50
รูปที่ 4.20 ส่งการแจ้งเตือนกลับไปหาผู้ใช้งานเพื่อบอกความคืบหน้าของรายการสินค้า	51
รูปที่ 4.21 ความคืบหน้าของรายการสินค้า	51
รูปที่ 4.22 การสแกน QR code	52
รูปที่ 4.23 ดูสินค้าในรายการผ่านการสแกน QR code	52
รูปที่ ก.1 ตัวอย่างการเชื่อมต่อ database ฝั่ง server โดยใช้ภาษา PHP.....	58
รูปที่ ก.2 การ implement retrofit2 เข้ามายังแอปพลิเคชัน	59
รูปที่ ก.3 ตัวอย่างการสร้าง model เพื่อรองรับข้อมูลที่ดึงมาจากฐานข้อมูล.....	59
รูปที่ ก.4 ตัวอย่างส่วนที่ใช้ในการ query เพื่อดึงข้อมูลจากฐานข้อมูล	60
รูปที่ ก.5 ตัวอย่างส่วนที่กำหนด HTTP verb และลักษณะของ URL.....	61
รูปที่ ก.6 ตัวอย่างชุดข้อมูล JSON	61
รูปที่ ก.7 ตัวอย่างการเปิด server เพื่อใช้งาน localhost บน Xampp.....	62
รูปที่ ก.8 ตัวอย่างโปรแกรม Postman.....	63
รูปที่ ก.9 ตัวอย่างผลลัพธ์ของ service ที่ต้องการดึงช่วงเวลาเปิด-ปิดร้านของร้านรหัส 00914.....	63
รูปที่ ก.10 ตัวอย่างการตั้งค่า base URL ในแอปพลิเคชัน	64
รูปที่ ก.11 ตัวอย่างของ interface ของการเรียกใช้ web service	65
รูปที่ ก.12 ตัวอย่างรูปแบบการเขียนโดยใช้ retrofit2.....	66
รูปที่ ก.13 ตัวอย่างผลลัพธ์ของการเรียกใช้ web service ในการแสดงรายละเอียดสินค้าในฝั่ง client	66
รูปที่ ข.1 ตัวอย่างหน้าเว็บ OneSignal.....	68

รูปที่ ข.2 ตัวอย่างหน้าเว็บ OneSignal หลังกด ADD APP	68
รูปที่ ข.3 ตัวอย่างหน้าเว็บ OneSignal ที่ให้เลือก platform การใช้งาน	69
รูปที่ ข.4 ตัวอย่างหน้าเว็บ OneSignal ที่ให้กรอกข้อมูลรายละเอียดของ Firebase.....	69
รูปที่ ข.5 ตัวอย่างหน้าเว็บ OneSignal เลือก platform เป้าหมายที่ต้องการจะทำการทดสอบ	70
รูปที่ ข.6 ตัวอย่างหน้าเว็บ OneSignal ที่ให้ Your App ID เพื่อนำไปใช้ในขั้นตอนถัดไป.....	70
รูปที่ ข.7 เป็นการติดตั้ง Android SDK	71
รูปที่ ข.8 โค้ดที่ต้องเพิ่มในส่วนที่ 1	71
รูปที่ ข.9 โค้ดที่ต้องเพิ่มในส่วนที่ 2	72
รูปที่ ข.10 โค้ดที่ต้องเพิ่มในส่วนที่ 3	72
รูปที่ ข.11 ตัวอย่างเว็บในการทดสอบการส่งการแจ้งเตือนไปยัง subscriber	73
รูปที่ ข.12 โค้ดที่ไว้ใช้เริ่มต้น OneSignal ในแอปพลิเคชัน	73
รูปที่ ข.13 โค้ดตัวอย่างของการกำหนด tag.....	73
รูปที่ ข.14 โค้ดตัวอย่างสำหรับการส่งการแจ้งเตือน	74
รูปที่ ข.15 โค้ดตัวอย่างสำหรับการส่งการแจ้งเตือน (ต่อ).....	74
รูปที่ ข.16 โค้ดส่วน String JSON Body	75
รูปที่ ข.17 โค้ดตัวอย่าง NotificationReceivedHandler	75

สารบัญตาราง

ตารางที่ 2.1 ตารางเปรียบเทียบ Firebase กับ MongoDB	23
ตารางที่ 3.1 ตารางพนักงาน (employee).....	32
ตารางที่ 3.2 ตารางผู้ใช้งาน (member).....	32
ตารางที่ 3.3 ตารางประเภทการส่ง (delivery_type).....	32
ตารางที่ 3.4 ตารางรายการสินค้า (order).....	33
ตารางที่ 3.5 ตารางสถานะความคืบหน้า(order_operation_s).....	34
ตารางที่ 3.6 ตารางการเชื่อมโยงสินค้า (order_product_mapping).....	34
ตารางที่ 3.7 ตารางสถานะของรายการสินค้า (order_s).....	35
ตารางที่ 3.8 ตารางการจ่ายเงิน (payment).....	35
ตารางที่ 3.9 ตารางสินค้า (product)	35
ตารางที่ 3.10 ตารางประเภทสินค้า (product_type).....	36
ตารางที่ 3.11 ตารางร้านค้า (shop).....	36
ตารางที่ 3.12 ตารางคลังสินค้า (stock).....	36
ตารางที่ 3.13 ตารางเวลา (time).....	37

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท โกซอฟท์ (ประเทศไทย) จำกัด เป็นบริษัทในกลุ่มธุรกิจของบริษัท ซี.พี.ออลส์ จำกัด (มหาชน) ภายใต้สัญลักษณ์ทางการค้า 7-11 ก่อตั้งด้วยการนำองค์กรด้านไอที ที่มีประสบการณ์จากหลายสายงานในกลุ่ม ให้เป็นหนึ่งเดียวกันโดยมีวัตถุประสงค์ของบริษัทฯ คือเป็นบริษัทชั้นนำในการให้คำปรึกษา และบริการด้านสารสนเทศ (Information System) ในประเทศไทย ซึ่งทำให้การครอบคลุมถึงการพัฒนาโปรแกรมใช้งานภายในธุรกิจ (In-house Application Development) ติดตั้งระบบโปรแกรมชั้นนำ (System Integrator) ระบบเครือข่าย (Networking) ระบบรักษาความปลอดภัยข้อมูล (Security System) และอื่นๆ นอกจากนี้ บริษัทยังมีศูนย์ข้อมูล (Data Center) และศูนย์ Call Service ให้บริการแก่ลูกค้าด้วย

ในปัจจุบันองค์ประกอบสำคัญในดำรงชีวิตประจำวันจำเป็นต้องมีสิ่งอำนวยความสะดวกต่างๆ ไม่ว่าจะเป็น อนุภาค หรือ บริโภค ล้วนจำเป็นต้องซื้ออยู่เสมอ

ซึ่งเว็บแอปพลิเคชันนี้ทำเพื่ออำนวยความสะดวกในการซื้อของภายในร้านสะดวกซื้อ สามารถทำการสั่งซื้อสินค้าออนไลน์เพื่อให้ทางร้านนำส่งสินค้ามายังผู้สั่งซื้อ หรือไปรับกับทางร้านได้ และทำให้ระบบการสั่งซื้อเป็นระบบและมีประสิทธิภาพที่เพิ่มสูงขึ้น ซึ่งจะลดปัญหาไม่มีเวลาในการไปเลือกซื้อ หรือเพราะคนในร้านสะดวกซื้อมีคนจำนวนมาก จึงเลือกที่จะไม่ซื้อ ณ เวลานั้นแล้ว ยังเพิ่มยอดขายให้แก่ร้านสะดวกซื้อได้ ซึ่งในบางช่วงเวลาจำนวนลูกค้าค่อนข้างน้อย จึงนำโปรเจกต์ชิ้นนี้มาใช้ในช่วงเวลานั้น

โดยในส่วนปฏิบัติงานครั้งนี้ได้รับมอบหมายให้ทำการรับรายการสินค้าจากหน้าเว็บแอปพลิเคชันมาแสดงผลบนแอปพลิเคชันบนโทรศัพท์มือถือแพลตฟอร์มแอนดรอยด์ ว่าในรายการสินค้านั้นมีสินค้าใดบ้าง เบอร์ติดต่อ รายละเอียดที่จัดส่ง และเวลาที่ทำการส่ง เพื่อให้พนักงานไปหยิบสินค้าภายในร้านมาจัดเตรียมเพื่อรอลูกค้ามารับหรือนำไปจัดส่งตามสถานที่ที่ระบุไว้ในรายการสินค้า

1.2 วัตถุประสงค์

- 1.2.1 เพื่ออำนวยความสะดวกแก่พนักงานร้านสะดวกซื้อในการตรวจเช็คสินค้าในรายการที่สั่งมา
- 1.2.2 ลดปัญหาด้านการผิดพลาดของพนักงานในการจัดเรียงคิวรายการสินค้า
- 1.2.3 เพื่อสะดวกในการตรวจสอบสินค้าในรายการที่อยู่ในระหว่างการดำเนินการ รายการที่ถูกต้อง ยกเลิก และรายการที่เสร็จสมบูรณ์ได้

- 1.2.4 มีการแจ้งเตือนกลับไปยังผู้สั่งซื้อเพื่อให้ผู้สั่งซื้อสามารถดูความคืบหน้าของรายการสินค้านั้นๆ ได้

1.3 ขอบเขตของโครงการ

- 1.3.1 สามารถรับรายการสินค้าที่ถูกส่งผ่านทางเว็บแอปพลิเคชันเพื่อนำมาแสดงผลบนหน้าจอ โทรศัพท์มือถือแท็บเล็ตฟอร์มแอนดรอยด์ได้
- 1.3.2 สามารถดูข้อมูลของรายการสินค้า
- 1.3.3 สามารถดูข้อมูลของสินค้าภายในรายการที่สั่งได้
- 1.3.4 สามารถยกเลิกรายการสินค้าที่สั่งมาได้
- 1.3.5 สามารถค้นหารายการสินค้าตามหมายเลขรายการสินค้า
- 1.3.6 สามารถตั้งเวลาเปิด-ปิดระบบได้
- 1.3.7 สามารถแก้ไขช่วงเวลาเปิด-ปิดระบบได้
- 1.3.8 สามารถลบช่วงเวลาเปิด-ปิดระบบได้
- 1.3.9 สามารถแสดงช่วงเวลาเปิด-ปิดระบบได้
- 1.3.10 สามารถได้รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่ได้
- 1.3.11 สามารถได้รับการแจ้งเตือนเมื่อผู้สั่งซื้อทำการยกเลิกรายการสินค้า
- 1.3.12 สามารถส่งการแจ้งเตือนกลับไปหาผู้สั่งว่ารายการสินค้านี้มีความคืบหน้าอย่างไรได้
- 1.3.13 สามารถประมวลผล QR code เพื่อดูหมายเลขรายการสินค้าและสินค้าภายในรายการสินค้านั้นได้

1.4 วิธีการดำเนินงาน

- 1.4.1 ศึกษาเกี่ยวกับพื้นฐานการเขียนแอปพลิเคชันแอนดรอยด์
- 1.4.2 ศึกษาหลักการการออกแบบฐานข้อมูลและข้อมูลที่จะนำมาใช้
- 1.4.3 ศึกษาการใช้งาน web service
- 1.4.4 วิเคราะห์และออกแบบระบบ
- ออกแบบซอฟต์แวร์ของระบบ
 - ออกแบบฐานข้อมูลของระบบ
 - ออกแบบหน้าตาของแอปพลิเคชันสำหรับผู้ใช้
- 1.4.5 เขียนแอปพลิเคชันโดยแบ่งแต่ละหน้าตามสถานะของรายการสินค้าและความสามารถในการทำงาน
- หน้าแสดงรายการสินค้าที่อยู่ในคิว

- หน้าแสดงรายการสินค้าที่อยู่ในระหว่างการดำเนินการจัดส่งหรือรอลูกค้ามารับสินค้า
- หน้าแสดงประวัติรายการสินค้าจะจัดเก็บรายการสินค้าที่สำเร็จและถูกยกเลิกไว้ในช่วงระยะเวลา 2 วันของการส่งรายการนั้น
- หน้าแสดงช่วงเวลาการเปิด-ปิดระบบ
- หน้าสแกน QR code

1.4.6 ทดสอบและแก้ไขข้อผิดพลาดของแอปพลิเคชัน

1.4.7 จัดทำรูปเล่มและไฟล์นำเสนอ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการเข้าร่วมโครงการสหกิจกับทางบริษัท โกซอฟท์ (ประเทศไทย) จำกัด

จำแนกเป็น 3 ส่วนคือ

1.5.1 ประโยชน์ต่อบริษัท

1.5.1.1 นำโปรเจกต์ไปพัฒนาต่อให้กับ Seven Eleven

1.5.1.2 นำไปประยุกต์ใช้กับ Application อื่นๆ

1.5.2 ประโยชน์ต่อตนเอง

1.5.2.1 ได้รับประสบการณ์การทำงาน

1.5.2.2 ฝึกวินัยและการตรงต่อเวลา

1.5.2.3 พัฒนาทักษะการเขียน Android และ NodeJS

1.5.2.4 พัฒนาทักษะการเขียน web service

1.5.3 ประโยชน์ต่อมหาลัย

1.5.3.1 ได้เรียนรู้รูปแบบการทำงานและสามารถนำมาพัฒนาหลักสูตรให้กับรุ่น

ต่อไป

บทที่ 2

แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบปฏิบัติการ Android

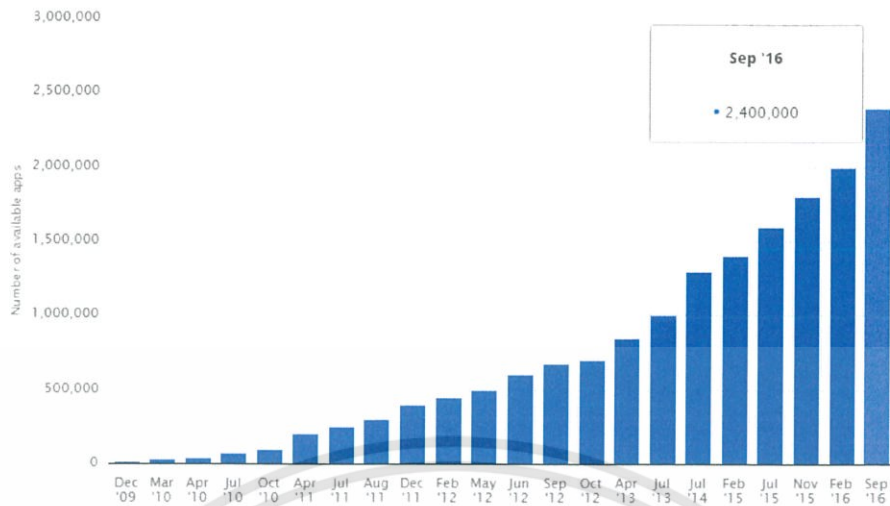
Android คือ ระบบปฏิบัติการบนสมาร์ตโฟน (Smart Phone) และแท็บเล็ต (Tablet) แบบ Open source ที่มีพื้นฐานอยู่บนระบบปฏิบัติการ Linux อีกชั้นหนึ่ง (Base on Linux) ซึ่งในปัจจุบัน Android นับเป็นระบบปฏิบัติการบนสมาร์ตโฟนที่ได้รับความนิยมมากที่สุด นอกจากนี้แล้วในปัจจุบันยังถูกนำไปใช้บนอุปกรณ์อิเล็กทรอนิกส์อื่นๆอีก เช่น กล้องดิจิตอลระบบ Android, Smart TV ระบบ Android, Android Wear (เช่น นาฬิกาข้อมือ, แว่นตา) เป็นต้น โดยมีการแสดงวิวัฒนาการของโลโก้ระบบปฏิบัติการ Android ตั้งแต่ ปี 2008 – 2013 ดังรูปที่ 2.1



รูปที่ 2.1 วิวัฒนาการของโลโก้ระบบปฏิบัติการ Android ตั้งแต่ ปี 2008 – 2013

ที่มา : <http://www.xda-developers.com/wp-content/uploads/2013/11/android-evolution.jpg>

Android เป็นระบบปฏิบัติการแบบเปิดหรือที่เรียกว่า Open source ซึ่งอนุญาตให้ผู้ผลิต, นักพัฒนา และผู้ให้บริการเครือข่าย สามารถปรับแต่งและจัดจำหน่ายได้ นอกจากนี้การพัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการ Android ยังใช้ภาษา Java เป็นภาษาหลัก ทำให้ง่ายต่อการพัฒนาและด้วยตัวช่วยหลายๆ อย่างที่ทาง Google ได้จัดสรรไว้สำหรับนักพัฒนาทุกระดับได้ใช้ในการพัฒนาแอปพลิเคชันทำให้ใครๆ ก็สามารถพัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการ Android ได้ ส่งผลให้บน Google Play Store ที่เป็นแหล่งให้ดาวน์โหลดแอปพลิเคชันสำหรับระบบปฏิบัติการ Android ซึ่งในปัจจุบันมีจำนวนแอปพลิเคชันสูงถึง 2,000,000 แอปพลิเคชันที่ใช้ระบบปฏิบัติการนี้ ซึ่งเป็นเหตุผลให้ในการปฏิบัติงานครั้งนี้ได้เลือกใช้งานระบบ Android แทนการเลือกใช้งานระบบ IOS โดยมีรูปสถิติจำนวนแอปพลิเคชันบน Google Play Store ตั้งแต่ปี 2009 – 2016 ดังรูปที่ 2.2



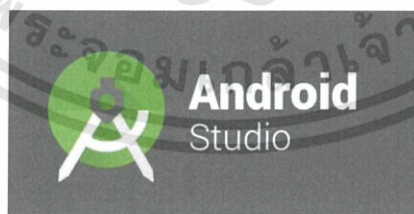
รูปที่ 2.2 สถิติจำนวนแอปพลิเคชันบน Google Play Store ตั้งแต่ปี 2009 – 2016

ที่มา : <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>

2.2 Android Studio

Android Studio เป็น Tool ไร้พัฒนา Android โดยเฉพาะ วัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา Application บน Android โดยเฉพาะให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI (Graphical User Interface) ที่ช่วยให้สามารถแสดงผล Application ในมุมมองที่แตกต่างกันบน Smart Phone แต่ละรุ่น โดยสามารถแสดงผลได้ทันทีโดยไม่ต้องทำการรัน Application บน Emulator

ซึ่งในปฏิบัติการครั้งนี้ได้เลือกใช้ Tool ตัวนี้เนื่องจากมีผู้ใช้เป็นจำนวนมาก และมีคู่มือการใช้ที่หลากหลาย เมื่อเกิดปัญหาขณะปฏิบัติงานสามารถหาวิธีการแก้ไขได้ง่าย โดยมีโลโก้ ดังรูปที่ 2.3



รูปที่ 2.3 โลโก้ Android Studio

ที่มา : <http://arctech.in.th/articles/125>

2.3 ภาษา Java

ภาษา Java เป็นภาษาเพื่อการพัฒนาโปรแกรมเชิงวัตถุหรือ Object Oriented Programming (OOP) ซึ่งมีจุดประสงค์หลักในการพัฒนาภาษา Java เพื่อนำมาใช้แทนภาษา C++ และจากรูปแบบของภาษา Java ที่ถูกเพิ่มขึ้นมาทำให้ตัวภาษาส่วนที่เพิ่มเติมขึ้นมามีความคล้ายกับภาษา Objective-C

ชื่อภาษา Java มีความคล้ายคลึงกับชื่อภาษา Javascript แต่ตัวภาษาและการนำไปใช้นั้นแตกต่างกันเป็นอย่างมาก ทั้งภาษา Java และภาษา Javascript ไม่มีส่วนเกี่ยวข้องใดๆกันทั้งสิ้น โดยมีโลโก้ ดังรูปที่ 2.4



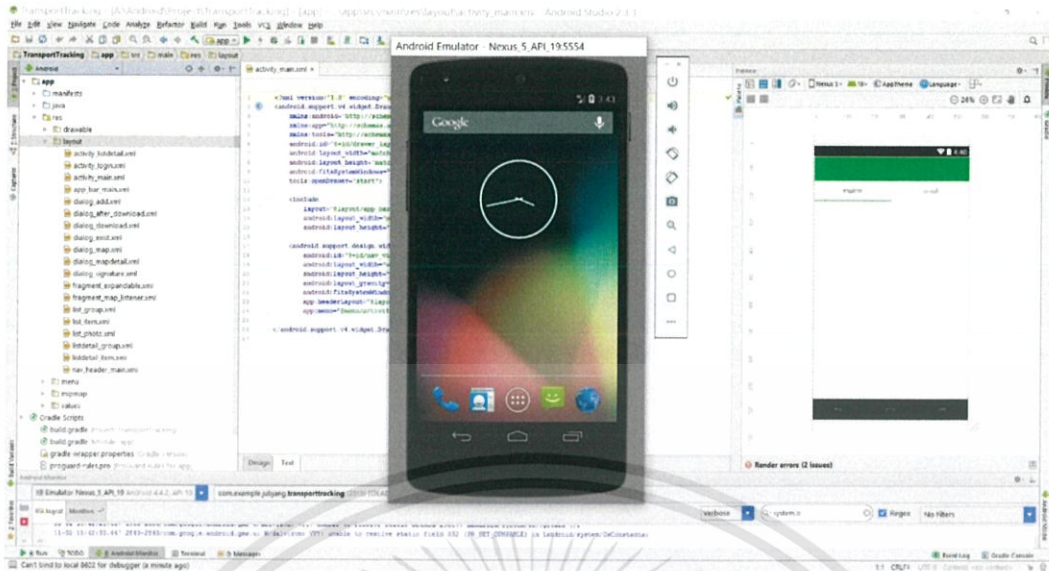
รูปที่ 2.4 โลโก้ของภาษา Java

ที่มา : <https://www.kinvey.com/wp-content/uploads/2013/11/java-logo.jpg?w=300>

2.4 Android SDK

Android SDK หรือ Android Software Development Kit เป็นชุดโปรแกรมที่ได้ถูกคิดค้นและพัฒนาขึ้นเพื่อให้นักพัฒนาได้ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยทางบริษัท Google ได้แจกจ่าย Android SDK ให้ใช้โดยไม่คิดค่าใช้จ่ายใดๆทั้งสิ้น ด้วยสาเหตุนี้เองจึงทำให้ระบบปฏิบัติการ Android มีแอปพลิเคชันเพิ่มขึ้นอย่างรวดเร็ว

ใน Android SDK นี้จะประกอบไปด้วยโปรแกรมช่วยเหลือและไลบรารีต่างๆที่จำเป็นและช่วยให้การพัฒนาแอปพลิเคชันทำได้สะดวกสบายยิ่งขึ้น เช่น Android Virtual Device หรือ Emulator ที่ช่วยให้ผู้ที่พัฒนาแอปพลิเคชันสามารถทดสอบแอปพลิเคชันบน Emulator ได้โดยไม่ต้องมีสมาร์ตโฟน เพราะ Android Virtual Device นี้มีการจำลองสภาพแวดล้อม, ทรัพยากรตัวเครื่องของสมาร์ตโฟนระบบปฏิบัติการ Android ที่สามารถใช้งานได้ใกล้เคียงกับสมาร์ตโฟนระบบปฏิบัติการ Android เครื่องจริงเกือบทั้งหมด โดยมีตัวอย่าง Android Virtual Device ที่มีมาให้ใน Android SDK ดังรูปที่ 2.5



รูปที่ 2.5 ตัวอย่าง Android Virtual Device ที่มาให้ใน Android SDK

2.5 Java JDK

Java JDK หรือ Java Development Kit เปรียบเสมือนชุดเครื่องมือที่ใช้ในการพัฒนาโปรแกรมด้วยภาษา Java ที่ประกอบไปด้วย Java compiler, Java debugger, Java doc และ Java interpreter หรือ Java VM ถ้าหากนักพัฒนาไม่ทำการติดตั้ง Java JDK แล้วจะไม่สามารถทำการ compile หรือ run โปรแกรมหรือแอปพลิเคชันได้ ทั้งนี้ Java SDK แบ่งย่อยได้ 3 รุ่น ดังนี้

- Java SE (Standard Edition) สำหรับพัฒนาโปรแกรมบนคอมพิวเตอร์ทั่วไป
- Java ME (Micro Edition) สำหรับพัฒนาโปรแกรมบนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ หรือแท็บเล็ต ส่วนมากใช้ในการพัฒนาเกม
- Java EE (Enterprise Edition) สำหรับพัฒนาโปรแกรมในองค์กรใหญ่ๆ หรือมีขอบเขตของโครงการกว้างมาก

2.6 Android Support Library

ด้วยสาเหตุที่ระบบปฏิบัติการ Android นั้นมีการพัฒนาอย่างต่อเนื่องและรวดเร็วมาก ทำให้การพัฒนาแอปพลิเคชันเพื่อให้รองรับกับระบบปฏิบัติการ Android เวอร์ชันใหม่เสมอ ในขณะเดียวกัน ก็ยังมีผู้ใช้ระบบปฏิบัติการ Android เวอร์ชันเก่าก็ยังคงมีอยู่เป็นจำนวนมาก ทางนักพัฒนาจึงจำเป็นต้องพัฒนาแอปพลิเคชันให้รองรับทั้งระบบปฏิบัติการ Android เกือบทุกเวอร์ชัน แต่บางคำสั่งในการพัฒนาแอปพลิเคชันเพื่อให้รองรับกับระบบปฏิบัติการ Android ในเวอร์ชันที่แตกต่างกัน ก็อาจจะต้องใช้คำสั่งในการพัฒนาที่แตกต่างกัน

ปัญหาดังกล่าวถือเป็นปัญหาสำคัญในการพัฒนาแอปพลิเคชันเป็นอย่างมาก ทางบริษัท Google จึงได้คิดค้นและพัฒนา Android Support Library ที่เปรียบเสมือนชุดคำสั่งที่เป็นมาตรฐานในการพัฒนาเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

แอปพลิเคชันบนระบบปฏิบัติการ Android ที่ช่วยให้นักพัฒนาเขียนคำสั่งเพียงชุดเดียวก็สามารถใช้คำสั่งนี้ได้กับระบบปฏิบัติการ Android ได้แบบครอบคลุมเกือบทุกเวอร์ชัน และถ้าหากมีการพัฒนาระบบปฏิบัติการ Android เวอร์ชันใหม่่ออกมาอีก ทางนักพัฒนาแอปพลิเคชัน ก็เพียงแค่อัพเดท Android Support Library ให้เป็นเวอร์ชันใหม่ตามคำสั่งในแอปพลิเคชันก็จะถูกปรับปรุงให้ตัวระบบปฏิบัติการเข้าใจได้เอง โดยที่นักพัฒนาแอปพลิเคชัน ไม่จำเป็นต้องไปแก้ไขในระดับชั้น Source code แต่อย่างใด

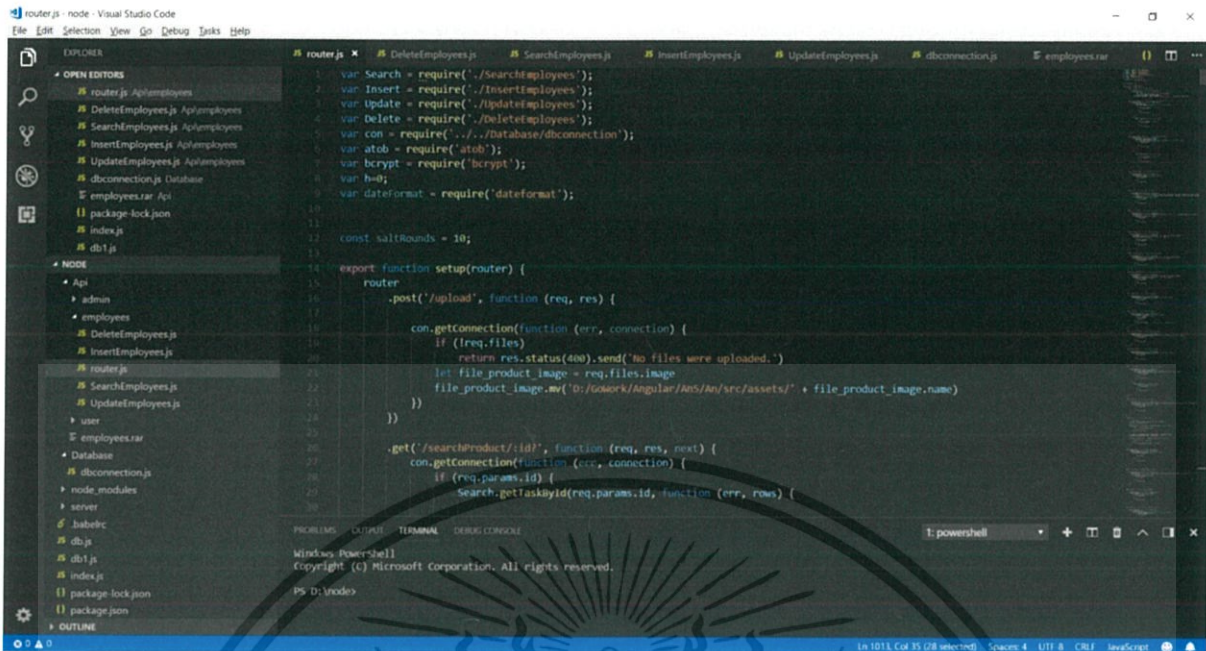
2.7 Visual Studio Code

Visual Studio Code เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบฟรีๆ ที่ต้องการความเป็นมืออาชีพ ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่างๆ ให้เลือกใช้อย่างมากมาย ไม่ว่าจะเป็น 1. การเปิดใช้งานภาษาอื่นๆ ทั้งภาษา C++, C#, Java, Python, PHP หรือ Go 2. Themes 3. Debugger 4. Commands เป็นต้น โดยมีโลโก้ดังรูปที่ 2.6 และ ตัวอย่างหน้าต่างการโปรแกรม Visual Studio Code ดังรูปที่ 2.7



รูปที่ 2.6 โลโก้ของ Visual Studio Code

ที่มา : https://en.wikipedia.org/wiki/Visual_Studio_Code



รูปที่ 2.7 ตัวอย่างหน้าต่างการโปรแกรม Visual Studio Code

2.8 NodeJS

NodeJS เป็นเทคโนโลยีฝั่ง Sever Side ที่ถูกพัฒนาด้วยภาษา JavaScript แต่เดิมภาษา JavaScript ทำงานฝั่ง Client เป็นหลัก แต่ที่จริงแล้ว NodeJS เป็น Client หรือ Server ก็ขึ้นอยู่กับจุดประสงค์ของแอปพลิเคชันนั้น แต่จุดเริ่มต้นมาจาก Server Side เป็นหลัก NodeJS คือ JavaScript มีการ Compiled เป็น Byte Code ซึ่ง NodeJS สามารถ Debug ได้ต่างจาก JavaScript ในยุคแรกๆ ทำให้แก้ปัญหาง่ายขึ้น และสามารถรันได้บนทุกระบบปฏิบัติการยอดนิยม เนื่องจากมาพร้อมกับเทคโนโลยีที่เรียกว่า Non – Blocking I/O ปัจจุบัน NodeJS ถูกนำมาเป็น Web Server , Mobile Hybrid , IOT , Webkit , TVOS , OS เป็นต้น โดยมีโลโก้ ดังรูปที่ 2.8

เหตุผลที่ได้รับความนิยมหลักๆคือ

1. เขียนโค้ดเข้าใจได้ง่าย
2. มี Library ฟรีมากมาย
3. ใช้ทรัพยากรน้อย
4. เรียนรู้ได้เร็วสำหรับโปรแกรมเมอร์ทุกระดับ



รูปที่ 2.8 โลโก้ของNodeJS

ที่มา : <https://en.wikipedia.org/wiki/Node.js>

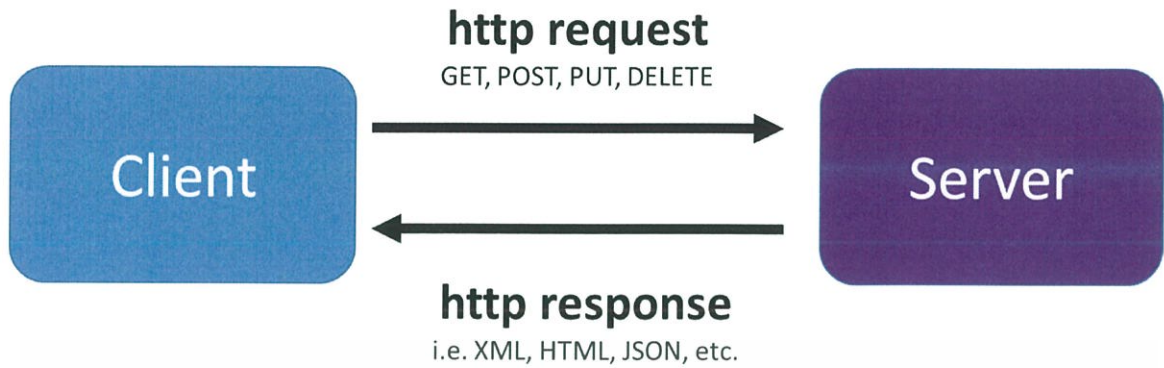
2.9 Web Service

Web service คือมาตรฐานในการเชื่อมต่อ Web-Base Application บน Internet Protocol โดยอาศัย XML, SOAP, WSDL และ UDDI

- XML (Extensible Markup Language) หรือ JSON (JavaScript Object Notation) เป็นformat ที่จะบอกแหล่งที่มาและรายละเอียดข้อมูลได้
- SOAP หรือ REST เป็น protocol ที่ใช้แลกเปลี่ยนข้อมูล
- WSDL (Web Services Description Language) ใช้ในการอธิบายโครงสร้างของ Serviceทั้งหมดหรือคู่มือการใช้งาน
- UDDI (Universal Description, Discovery, and Integration) เป็นส่วนกลางให้ระบบอื่นมาลงทะเบียนไว้ โดยใช้ WSDL file ทำให้ฝั่ง Client สามารถดึงเอาข้อมูลไปใช้ติดต่อกับ Web Service ได้
-

2.10 RESTful หรือ REST

REST หรือ Representational state transfer คือ การสร้าง Web Service ชนิดหนึ่งที่ใช้สื่อสารกันบน Internet ใช้หลักการแบบ Stateless คือไม่มี Session ซึ่งต่างจาก Web Service แบบอื่น เช่น WSDL และ SOAP การทำงานของ RESTful Web Service จะอาศัย URI/URL ของ Request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูปแบบ XML, HTML, JSON โดย Response ที่ตอบกลับจะเป็นการยืนยันผลของคำสั่งที่ส่งมา และสามารถพัฒนาด้วยภาษา programming ได้หลากหลาย คำสั่งก็จะมีตาม HTTP verbs ซึ่งก็คือ GET, POST, PUT และ DELETE โดยมี ภาพจำลองการทำงานของ REST ดังรูปที่ 2.9



รูปที่ 2.9 ภาพจำลองการทำงานของ REST

2.11 JSON

JSON หรือ JavaScript Object Notation คือ Format ที่เป็น text ใช้ในการสร้าง object เพื่อส่งข้อมูลระหว่าง Application หรือ Applications Program Interface (API) โดย Format จะมีรูปแบบเป็นคู่ Key-Value หรือเป็นแบบ Array ซึ่งสามารถนำมาใช้แทน XML format ได้ JSON เป็น format ที่ได้รับการใช้งานจาก JavaScript มาก่อน

JSON Schema ใช้สำหรับแสดง Format โครงสร้างของ JSON เพื่อทำ Validation, Documentation และ Interaction Control ซึ่ง Schema จะเป็นตัวบอกว่าข้อมูลต้องมีอะไรบ้าง ซึ่งใช้หลักการเดียวกับ XML Schema (XSD) แต่ไม่มีมาตรฐานของ File Extension โดยมี ตัวอย่างข้อมูลรูปแบบ JSON ดังรูปที่ 2.10

```
{
  "available_travel_modes" : [ "DRIVING", "WALKING" ],
  "geocoded_waypoints" : [
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJr7GPpXvDwoARYOEOnjwQmJ8",
      "types" : [ "establishment", "food", "point_of_interest", "restaurant" ]
    },
    {
      "geocoder_status" : "OK",
      "place_id" : "ChIJN_BnVtKY4jAR7JfDmY5gdPs",
      "types" : [ "route" ]
    }
  ],
  "routes" : [],
  "status" : "ZERO_RESULTS"
}
```

รูปที่ 2.10 ตัวอย่างข้อมูลรูปแบบ JSON

ที่มา : <https://saixiii.com/what-is-json>

2.12 API

API หรือ Application Program Interface คือ เป็นช่องทางสำหรับขอใช้บริการคำสั่ง จาก Operation System (OS) หรือ Application อื่นๆ ซึ่งการใช้งานจะต้องทำการติดตั้ง Function และเรียกใช้งานตาม Document ที่เขียนไว้ โดยมีตัวอย่างการให้บริการของ Google APIs ดังรูปที่ 2.11 APIs สร้างขึ้นมา จาก 2 อย่าง คือ

1. ข้อกำหนดที่จะอธิบายการแลกเปลี่ยนข้อมูลระหว่าง Program ซึ่งทำออกมาในลักษณะ Document เพื่อบอกว่า Request/Response ต้องเป็นอย่างไร
2. Software ที่เขียนขึ้นตามข้อกำหนดและทำการเผยแพร่ออกไปให้ใช้งาน

ซึ่งรูปแบบการนำเอา API ไปใช้งานมีดังนี้

1. Libraries and frameworks ซึ่งเขียนขึ้นตาม Document ในรูปแบบภาษา Program ที่ต่างกันออกไปตามความเหมาะสมกับงาน เพื่อเอาไปทำเป็น framework ให้กับระบบที่ใช้ในการสื่อสารหากัน
2. Operating Systems API สามารถใช้งานในการสื่อสารระหว่าง Application และ Operating System
3. Remote APIs ซึ่งทำให้ Developer สามารถเข้าไปควบคุมทรัพยากรผ่านทาง Protocol เพื่อให้มีมาตรฐานการสื่อสารเดียวกัน ถึงแม้ว่าจะเป็นคนละ Technology
4. Web APIs นิยมใช้กันมากในปัจจุบัน เพราะอยู่ในกลุ่มของ HTTP และขยายออกไปสู่รูปแบบ XML และ JSON ซึ่งโดยรวมแล้วก็คืออยู่บน Web Service



The screenshot shows the Google APIs Explorer interface. It features a sidebar on the left with navigation options like 'API และบริการ' and 'API และบริการที่เปิดใหม่'. The main area displays a list of APIs with columns for 'API', 'สถานะ', 'วันที่ออกสู่สาธารณะ', 'เวลาในการตอบสนอง', and 'เวลาในการตอบสนอง 99%'. The table lists APIs such as Google Maps Android API, Google Maps Directions API, and Google Places API for Android.

API	สถานะ	วันที่ออกสู่สาธารณะ	เวลาในการตอบสนอง	เวลาในการตอบสนอง 99%	
Google Maps Android API	-	-	-	-	ดูได้
Google Maps Directions API	-	-	-	-	ดูได้
Google Places API for Android	-	-	-	-	ดูได้

รูปที่ 2.11 ตัวอย่างการให้บริการของ Google APIs

ที่มา : <https://saixiii.com/what-is-json>

2.13 Database

Database หรือ ฐานข้อมูล คือส่วนที่ใช้เก็บข้อมูลเพื่อให้สามารถบริการการจัดเก็บและการเข้าถึงข้อมูลส่วนกลางได้สะดวกรวดเร็ว ซึ่งข้อมูลจะถูกเก็บในรูปแบบของ rows, columns และ tables โดยอาศัยดัชนี (indexed) เพื่อให้สามารถค้นหาข้อมูลได้รวดเร็ว นอกจากนี้ยังสามารถทำการเพิ่ม (insert), ลบ (delete) หรือ เปลี่ยนแปลง (update) ข้อมูลใหม่ได้อีกด้วย

2.14 SQL

SQL หรือ Structured Query Language คือ ภาษาเพื่อการจัดการฐานข้อมูล (Database) เป็นภาษามาตรฐานบนระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) สำหรับสร้าง, แก้ไขและเรียกใช้ฐานข้อมูล เช่นคำสั่ง SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP เป็นต้น

อีกทั้งภาษา SQL ยังเป็นระบบเปิด (Open System) หมายถึง สามารถใช้คำสั่ง SQL กับฐานข้อมูลชนิดใดก็ได้ ภาษา SQL จึงมีประโยชน์มากๆ ในการพัฒนาระบบงานต่างๆ ไม่ว่าจะเป็นพัฒนา Web Database Application, Windows Application, Web Service, Android Application ที่จะต้องทำงานกับข้อมูล โดยมีตัวอย่างคำสั่งภาษา SQL ดังรูปที่ 2.12

```
SELECT column_name,column_name FROM table_name;
```

```
INSERT INTO table_name VALUES (value1,value2,value3,...);
```

```
UPDATE table_name SET column1=value1,column2=value2,...
```

```
WHERE some_column=some_value;
```

```
DELETE FROM table_name WHERE some_column=some_value;
```

รูปที่ 2.12 ตัวอย่างคำสั่งภาษา SQL

ที่มา : <http://www.9experttraining.com/articles/ภาษา-sql-structured-query-language-และ-transact-sql>

2.15 Xampp

Xampp เป็นโปรแกรม Apache web sever ไว้จำลอง web server เพื่อไว้ทดสอบ สคริป หรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใดๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP, MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL, phpMyadmin ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe ซึ่งนี่เป็นเหตุผลที่ใช้ในการปฏิบัติงานในครั้งนี้

โปรแกรม Xampp สามารถใช้งานได้ 4 OS ได้แก่

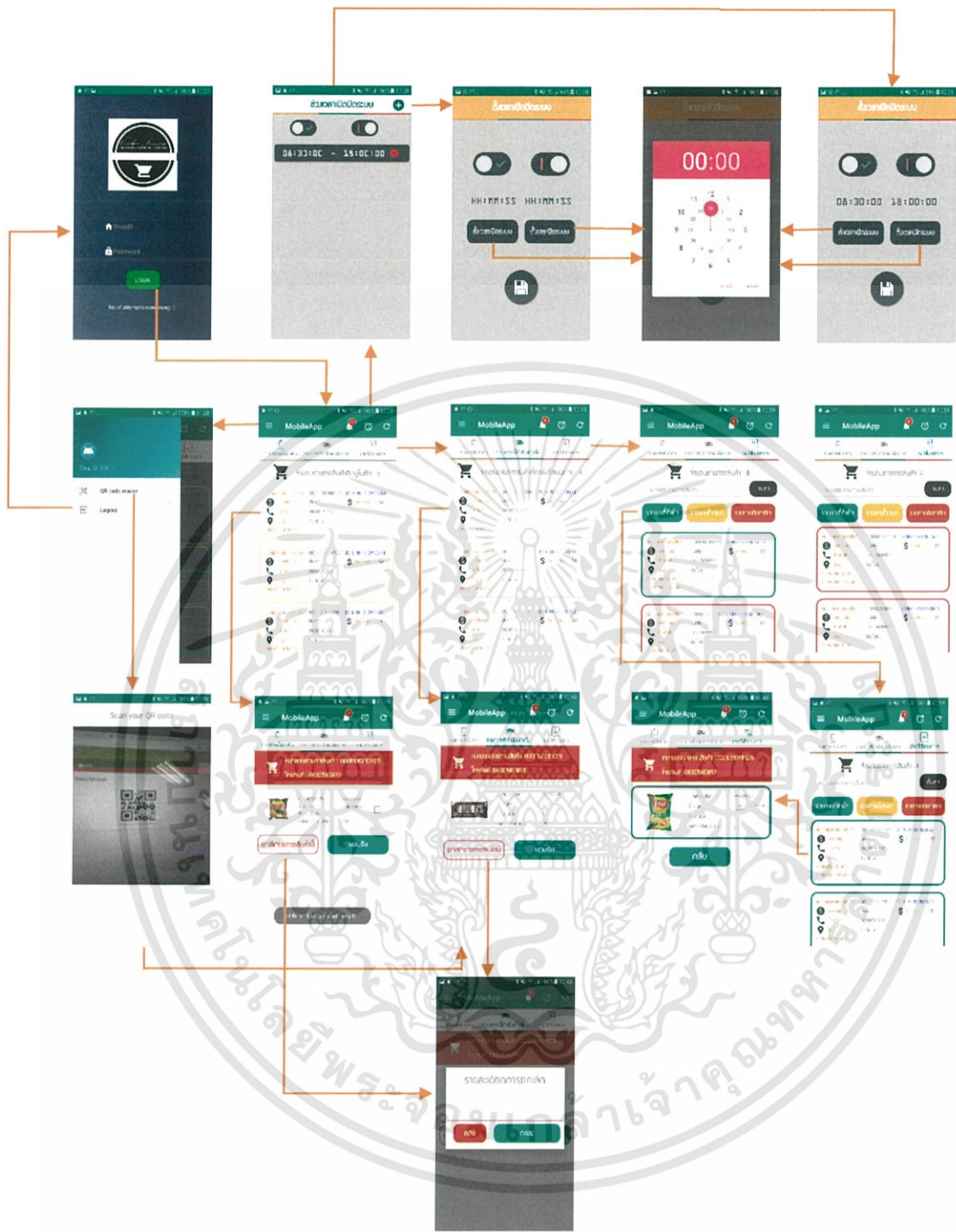
1. Window สามารถใช้งานได้กับ window รุ่น 2000, 2003, xp,vista,window 7 และจะมาพร้อมกับ
2. Linux สำหรับ SuSE, RedHat, ManDrake, Debian และ Ubuntu
3. Mac OS X
4. Solaris สำหรับ Solaris 8 และ Solaris 9

2.16 แอปพลิเคชันตัวทดลอง หรือ Prototype (Mockup)

Prototype หรือ Mockup เปรียบเสมือนตัวต้นแบบของโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ ที่มีการตอบสนองต่อการใช้งานของผู้ใช้ เช่น กดปุ่มเปลี่ยนหน้าแล้วหน้าแอปพลิเคชันเปลี่ยนหน้า, กดไฮเปอร์ลิงค์บนเว็บไซต์แล้วเปิดหน้าเว็บไซต์ใหม่ขึ้นมา เป็นต้น ซึ่งหลังจากออกแบบ Prototype เสร็จสิ้นแล้วจะต้องมีการนำไปทดสอบกับกลุ่มผู้ใช้งาน ผู้ใช้เข้าใจการออกแบบ สามารถใช้งานโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์ ได้จริง หากผู้ใช้งานรู้สึกว่าการออกแบบ, แอปพลิเคชัน, เว็บไซต์นี้ยังใช้งานยากอยู่ อาจจะต้องย้อนกลับไปออกแบบใหม่

การออกแบบ Prototype และนำไปทดสอบกับกลุ่มผู้ใช้งานก่อนทำให้ไม่ต้องแก้ไขในส่วนหน้าติดต่อผู้ใช้บ่อยๆ ซึ่งจะทำให้การพัฒนาเป็นไปอย่างล่าช้าและมีความซับซ้อนเป็นอย่างมาก

การออกแบบ Prototype สามารถทำได้หลายวิธี ไม่ว่าจะเป็นการออกแบบโดยใช้โปรแกรมบนคอมพิวเตอร์, ออกแบบโดยใช้แอปพลิเคชันบนสมาร์ตโฟนที่มีกันอย่างแพร่หลายในปัจจุบัน, เว็บไซต์สำหรับออกแบบ Prototype หรือแม้แต่การพัฒนาโปรแกรม, แอปพลิเคชัน หรือเว็บไซต์แบบมีการตอบสนองต่อการใช้งานของผู้ใช้ ออกมาก็สามารถทำได้ โดยในปฏิบัติการครั้งนี้มี prototype ดังรูปที่ 2.13



รูปที่ 2.13 Prototype ของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.17 ภาษา XML

ภาษา XML หรือ ภาษา Extensible Markup Language เป็นภาษาที่ใช้ในการแสดงผลข้อมูล ซึ่งภาษา XML จะมีความสามารถใกล้เคียงกับภาษา HTML แต่ภาษา XML จะมีความสามารถที่มากกว่า HTML ตรงที่ภาษา XML ไม่ใช่เพียงแค่แสดงผลข้อมูลแต่สามารถเก็บข้อมูลได้ด้วย หรือสามารถเก็บรายละเอียดของข้อมูลที่แสดงที่เรียกว่า Meta data เพราะจุดประสงค์ของภาษา XML จริงๆแล้วถูกคิดค้นมาเพื่อเก็บข้อมูล ส่วนการแสดงผลจริงๆจะใช้ภาษาเฉพาะที่เรียกว่า ภาษา XSL หรือภาษา Extensible Stylesheet Language

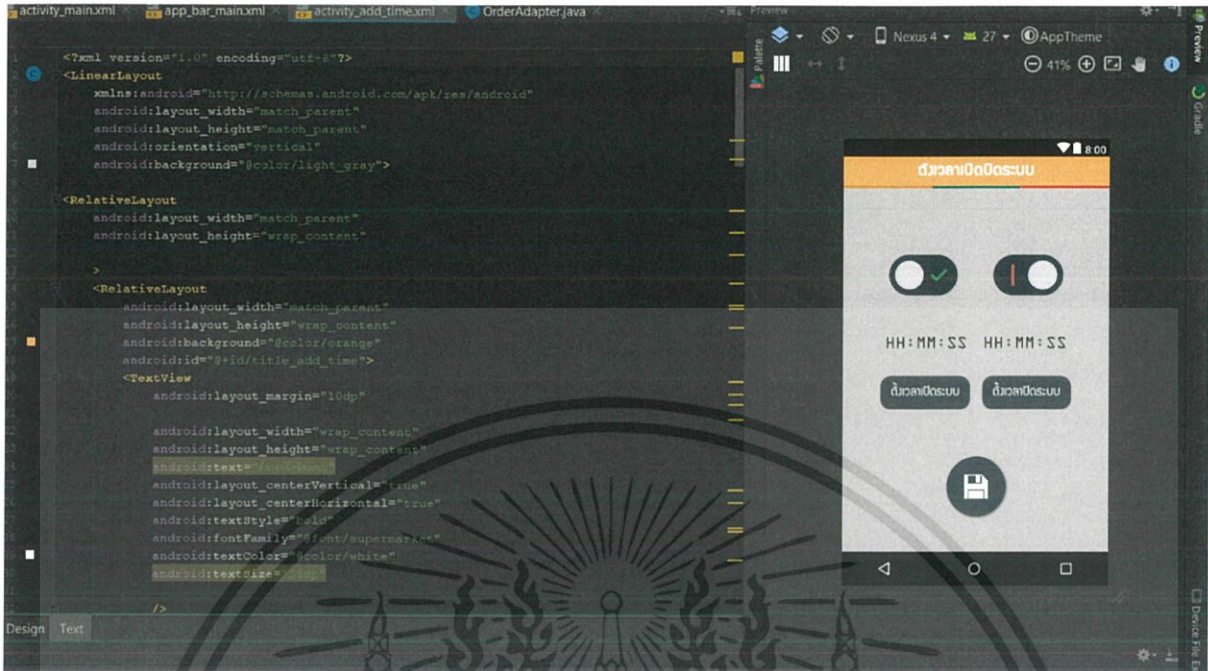
อาจกล่าวได้ว่าภาษา XML เป็นส่วนเสริมของภาษาต่างๆ เพราะภาษา XML ไม่สามารถแสดงผลข้อมูลได้ด้วยตัวเอง เพราะฉะนั้นหากต้องการจะแสดงผลข้อมูลจำเป็นต้องใช้ร่วมกับภาษาอื่นๆ เช่น HTML, PHP, JSP หรือภาษาอื่นๆที่สนับสนุน

ในปัจจุบันนอกจากมีการนำภาษา XML มาใช้ร่วมกับภาษา HTML เพื่อแสดงผลของข้อมูล และติดต่อกับผู้ใช้จากโครงสร้างของข้อมูลแล้ว ยังมีการนำมาใช้ร่วมกับภาษา Java เพื่อการแสดงผลเป็นส่วนติดต่อกับผู้ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android อีกด้วย โดยจะสามารถกำหนดรายละเอียดการแสดงผลต่างๆได้อย่างละเอียด เช่น ขนาดตัวอักษร, ตัวอักษรหนา เอียง บาง, ขนาดความกว้างของหน้าจอ แอปพลิเคชัน, ซ่อนหรือแสดงเลย์เอาต์ต่างๆภายในแอปพลิเคชัน เป็นต้น ซึ่งบางอย่างสามารถกำหนดค่าได้ด้วยภาษา Java แต่หลายอย่างภาษา Java ไม่สามารถทำได้ จึงจำเป็นต้องมีภาษา XML เข้ามาเป็นส่วนเติมเต็มในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยมีตัวอย่างภาษา XML ที่ใช้ในการพัฒนาแอปพลิเคชัน ดังรูปที่ 2.14 และรูปที่ 2.15

```
RelativeLayout ImageView
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.a58010420.gccs.mobileappshop.activity.MainActivity"
    android:background="@drawable/gradient">
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="230dp"
        android:layout_height="180dp"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="59dp"
        app:srcCompat="@drawable/logo_2" />
    <EditText
        android:id="@+id/etshop_id"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_below="@+id/imageView2"
        android:layout_marginTop="80dp"
        android:background="#11000000"
        android:ems="10"
        android:drawableLeft="@drawable/ic_home_white_24dp"
        android:textColor="#ffffff"
        android:textColorHint="@color/gray"
        android:hint="shop ID"
        android:inputType="textPersonName">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

รูปที่ 2.14 ตัวอย่างภาษา XML ที่ใช้ในการพัฒนาแอปพลิเคชัน



ภาพที่ 2.15 ตัวอย่างส่วนการตั้งเวลาเปิด-ปิดระบบของแอปพลิเคชัน ที่ถูกกำหนดด้วยภาษา XML

2.18 สถาปัตยกรรมซอฟต์แวร์แบบ Model-View-Controller หรือ MVC Pattern

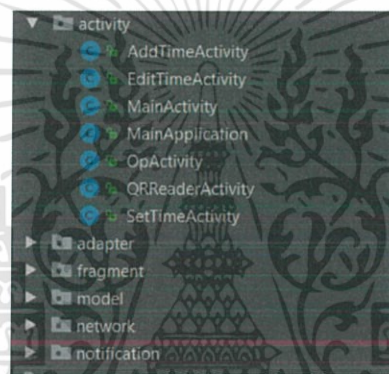
สถาปัตยกรรมซอฟต์แวร์แบบ Model-View-Controller หรือที่นิยมเรียกกันว่า MVC Pattern หรือที่นักพัฒนามักจะนิยมเรียกกันว่า MVC เป็นโครงสร้างสถาปัตยกรรมซอฟต์แวร์ที่นิยมใช้เป็นอย่างมากในการพัฒนาซอฟต์แวร์ในปัจจุบัน เพราะโครงสร้างแบบ MVC นี้จะมีการแยกส่วนของโปรแกรมเป็น 3 ส่วนคือ ส่วน Model, View และ Controller ซึ่งทั้ง 3 ส่วนจะแบ่งแยกหน้าที่การทำงานกันอย่างชัดเจน ซึ่งช่วยให้การพัฒนา การทดสอบ และการดูแลรักษาซอฟต์แวร์เป็นไปได้โดยง่ายมากขึ้น แม้ผู้ที่เข้ามาทำการพัฒนาต่อ หรือทดสอบซอฟต์แวร์นั้นๆ ก็สามารถตรวจสอบและเข้าใจโครงสร้างของซอฟต์แวร์นั้นได้ง่ายขึ้น โดยหน้าที่หลักของทั้ง 3 ส่วนนั้นมีดังนี้

- Model เป็นส่วนของซอฟต์แวร์ที่มีไว้เพื่อทำงานร่วมกับตัวข้อมูลของซอฟต์แวร์ เช่น การเก็บข้อมูล การส่งต่อข้อมูลไปยังส่วนอื่นของโปรแกรม เป็นต้น หรือจะเปรียบได้ว่า Model เป็นเสมือนระบบฐานข้อมูลของซอฟต์แวร์นั้นๆ ทั่วๆ ไป และการจัดการเกี่ยวกับข้อมูลทุกอย่าง จะถูกจัดการในส่วนของ Model ทั้งหมด
- View เป็นส่วนของซอฟต์แวร์ที่ทำหน้าที่แสดงผลและเป็นส่วนติดต่อกับผู้ใช้ของซอฟต์แวร์นั้นๆ ไม่สามารถทำอะไรได้มากไปกว่านั้น
- Controller เป็นส่วนของซอฟต์แวร์ที่ทำหน้าที่รับค่าข้อมูลต่างๆ จาก Model มาประมวลผลจนเสร็จสิ้นสมบูรณ์ ก่อนจะส่งต่อไปแสดงผลที่ส่วนของ View หรือ

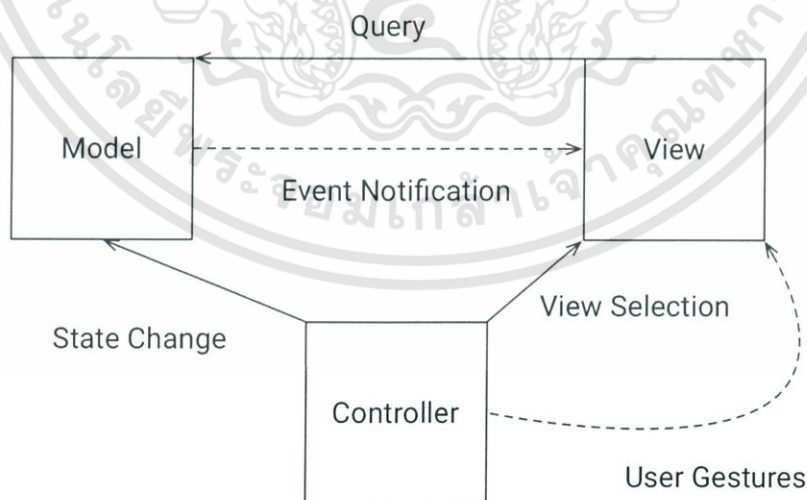
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

อาจจะรับค่ามาจากส่วน View แล้วทำหน้าที่ประมวลผล ก่อนส่งเข้าไปจัดเก็บลงใน ส่วนของ Model

จะเห็นแล้วว่าซอฟต์แวร์ที่ใช้โครงสร้างสถาปัตยกรรมแบบ MVC ในการพัฒนาจะทำให้ส่วนต่างๆของซอฟต์แวร์แบ่งหน้าที่กันอย่างชัดเจน และไม่มีการข้ามไปทำหน้าที่ของส่วนอื่นๆของซอฟต์แวร์ได้ และด้วยการแบ่งแยกส่วนต่างๆและหน้าที่กันอย่างชัดเจน ทำให้ผู้ที่ไม่เคยเห็นโค้ดที่ถูกพัฒนาขึ้นมาก่อนก็สามารถเข้าใจโครงสร้างของซอฟต์แวร์นั้นๆได้ง่ายขึ้นเป็นอย่างมาก และทำให้การพัฒนาต่อยอดซอฟต์แวร์ หรือการปรับปรุง ดูแลรักษา ตรวจสอบความถูกต้องของซอฟต์แวร์ก็สามารถทำได้ง่ายขึ้นเช่นกัน โดยมีตัวอย่างการแบ่งส่วนของซอฟต์แวร์ตามสถาปัตยกรรมซอฟต์แวร์แบบ MVC ของแอปพลิเคชัน ดังรูปที่ 2.16 และมี Workflow Diagram ของสถาปัตยกรรมซอฟต์แวร์แบบ MVC ดังรูปที่ 2.17



รูปที่ 2.16 ตัวอย่างการแบ่งส่วนของซอฟต์แวร์ตามสถาปัตยกรรมซอฟต์แวร์แบบ MVC ของแอปพลิเคชัน

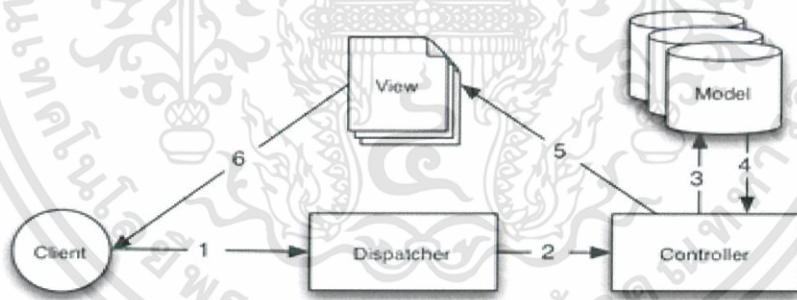


ภาพที่ 2.17 Workflow Diagram ของสถาปัตยกรรมซอฟต์แวร์แบบ MVC

ที่มา : https://i0.wp.com/www.arnondora.in.th/wp-content/uploads/2015/11/mvc_model.png

สำหรับซอฟต์แวร์ที่มีการใช้โครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC นั้นจะมีหลักการ
ทำงานคล้ายๆกัน ดังนี้

1. เมื่อผู้ใช้ ทำการร้องขอ(Request) เข้ามายังซอฟต์แวร์ ซอฟต์แวร์จะส่งคำขอ
ดังกล่าวไปที่ Controller ก่อนเพื่อให้ Controller พิจารณาว่าต้องทำหน้าที่อะไร
2. หากคำขอที่ส่งเข้ามามีความต้องการใช้ข้อมูล Controller จะส่งหน้าที่การทำงาน
ต่อไปยัง Model เพื่อไปหาข้อมูลภายใน Model มาให้
3. เมื่อ Controller ได้รับข้อมูลที่ได้รับการประมวลผลเพื่อนำไปใช้งานต่อแล้ว
Controller จะส่งข้อมูลที่ได้รับการประมวลผลแล้วนั้นไปให้ View
4. สุดท้าย View ทำการแสดงผลให้กับผู้ใช้
5. และเมื่อผู้ใช้ทำการใช้งานซอฟต์แวร์ส่วนต่างๆต่อไป ซอฟต์แวร์ก็จะมีการทำงานตาม
ขั้นตอนที่ 1 – 4 ทุกครั้งที่มีการส่งคำขอไปยังซอฟต์แวร์ หรือต้องติดต่อกับข้อมูล
ภายในซอฟต์แวร์นั้นๆ
6. กรณีที่ไม่จำเป็นต้องใช้งานข้อมูลที่เก็บอยู่ภายใน Model เมื่อ Controller รับคำขอ
จากผู้ใช้แล้ว Controller จะทำการประมวลผลแล้วส่งผลลัพธ์ที่ได้ไปยัง View เลย
โดยไม่ต้องมีการทำงานร่วมกับส่วน Model



ภาพที่ 2.18 หลักการทำงานของซอฟต์แวร์ที่ใช้โครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC ที่มา :

http://2.bp.blogspot.com/_DHOYWBAmhO4/TRQro3TSCLI/AAAAAAAAAEo/Ec472SdLcLO/s1600/basic_mvc.png

2.19 Third party library สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ

Android

Third party library หรือที่นิยมเรียกกันว่า “ไลบรารีนอก” คือส่วนเสริมหรือ Function สำเร็จรูปที่มีนักพัฒนาทำไว้สำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android ซึ่ง Third party library ในปัจจุบันก็มีให้เลือกใช้มากมาย และ Third party library แต่ละอย่างมีความสามารถที่แตกต่างกัน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะมีตั้งแต่ช่วยแปลงข้อมูลระหว่าง Object และ JSON, ช่วยจัดการการรับส่งข้อมูลจาก Web Service, การสแกนบาร์โค้ด หรือช่วยจัดการหาเส้นทางบนแผนที่ก็สามารถทำได้

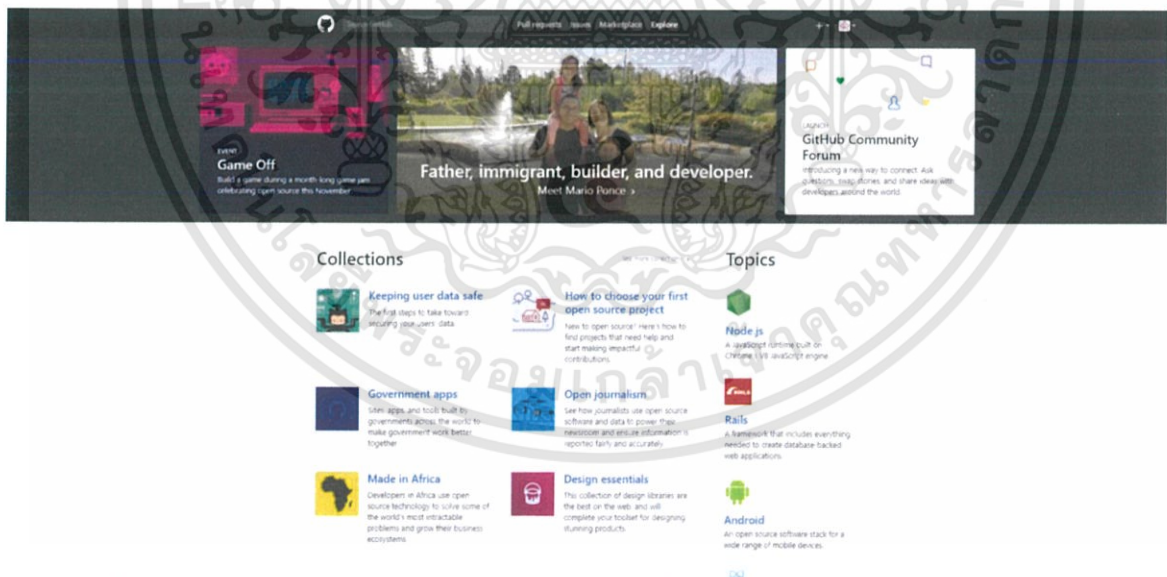
สำหรับการหา Third party library มาใช้งานในปัจจุบันนั้นสามารถหาได้ง่ายๆตามเว็บไซต์ต่างๆเช่น www.github.com หรือหาได้จากแอปพลิเคชันที่รวบรวม Third party library ไว้ให้แล้ว เช่น libraries for developers ก็มีให้เลือกใช้มากมาย

ซึ่งในปฏิบัติการครั้งนี้ได้ใช้งานในส่วนของ glide library, recycle view library และ card view library ซึ่งเป็นส่วนช่วยในการจัดเรียงการแสดงผลข้อมูลให้สวยงามและเป็นระเบียบมากยิ่งขึ้น โดยมีตัวอย่าง Third party library “glide” ดังรูปที่ 2.19 และตัวอย่างหน้าเว็บไซต์ www.github.com ที่เป็นแหล่งรวบรวม Third party library ดังรูปที่ 2.20



ภาพที่ 2.19 ตัวอย่าง Third party library “glide”

ที่มา : <https://www.androidhive.info/2016/04/android-glide-image-library-building-image-gallery-app/>



รูปที่ 2.20 ตัวอย่างหน้าเว็บไซต์ www.github.com ที่เป็นแหล่งรวบรวม Third party library

2.20 การออกแบบ User Experience และ User Interface (UX/UI)

ในการออกแบบซอฟต์แวร์หรือเว็บไซต์นั้นจำเป็นต้องมีการออกแบบอยู่ 2 ส่วนใหญ่ๆ ที่ผู้ที่ออกแบบจำเป็นต้องคำนึงถึง คือส่วนของ User Interface ที่เป็นส่วนติดต่อกับผู้ใช้ และ User Experience ที่เอกสารนี้เป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเหมือนศาสตร์ของการออกแบบแขนงหนึ่งที่กำลังได้รับความนิยมอย่างมากในการพัฒนาซอฟต์แวร์หรือเว็บไซต์ สำหรับความหมายและหลักการในการออกแบบของทั้ง 2 ส่วน คือ

- User Interface (UI) Design หรือการออกแบบส่วนติดต่อผู้ใช้ ส่วนติดต่อผู้ใช้เป็นส่วนสำคัญอย่างมากในการดึงดูดให้ผู้ใช้สนใจมาใช้ซอฟต์แวร์หรือเว็บไซต์ เพราะถ้าซอฟต์แวร์หรือเว็บไซต์ได้มีการออกแบบ UI ออกมาได้สวยงาม สบายตาผู้ใช้ ผู้ใช้ก็อยากจะใช้งานซอฟต์แวร์หรือเว็บไซต์นั้นๆ มากกว่าซอฟต์แวร์หรือเว็บไซต์ที่มีการออกแบบ UI แบบไม่โดดเด่นหรือน่าดึงดูดให้ใช้ แต่อย่างไรก็ตามในปัจจุบันมีการทดลองมากมายที่พิสูจน์ว่าการออกแบบ UI ให้สวยงามเพียงอย่างเดียว ไม่สามารถทำให้ผู้ใช้สนใจใช้ซอฟต์แวร์หรือเว็บไซต์นั้นๆ ได้มากพอ เพราะฉะนั้นจำเป็นต้องมีการออกแบบ User Experience ควบคู่กันไปด้วย
- User Experience (UX) หรือประสบการณ์ของผู้ใช้ นับตั้งแต่มีการคิดค้นและวิจัยเรื่อง User Experience จนกระทั่งปัจจุบันยังไม่สามารถให้คำนิยามที่ตายตัวของ UX ได้ โดยผู้ที่ได้ศึกษาเกี่ยวกับ UX มักให้คำอธิบายโดยรวมว่า UX ไม่ใช่เพียงความสวยงามของซอฟต์แวร์หรือเว็บไซต์ แต่เป็นความพึงพอใจของผู้ใช้ที่มีต่อซอฟต์แวร์หรือเว็บไซต์นั้นๆ ด้วย ความพึงพอใจของผู้ใช้จึงเกิดจากหลายปัจจัยที่เกี่ยวข้องกับซอฟต์แวร์หรือเว็บไซต์นั้นๆ มีด้วยกัน 3 ข้อหลักๆ ดังนี้ มีความสวยงาม, ใช้งานง่าย, ผู้ใช้สามารถเรียนรู้การใช้งานได้อย่างรวดเร็วโดยอาศัยการเรียนรู้จากคู่มือที่น้อยที่สุด หรืออาจไม่จำเป็นต้องศึกษาวิธีใช้จากคู่มือเลย หากซอฟต์แวร์หรือเว็บไซต์ที่มีการออกแบบ UX ที่ดีจำเป็นต้องมีปัจจัยทั้ง 3 อย่างที่ได้กล่าวไป

จะเห็นได้ว่าการทำให้ผู้ใช้พึงพอใจในซอฟต์แวร์หรือเว็บไซต์จำเป็นต้องมอบ User Experience ที่ดีให้กับผู้ใช้ และการที่ซอฟต์แวร์หรือเว็บไซต์จะสามารถมอบ User Experience ที่ดีให้กับผู้ใช้ได้จำเป็นต้องมีการออกแบบ User Interface ที่ดีเป็นหนึ่งในปัจจัยหลักด้วย ซึ่งในปฏิบัติการในครั้งนี้ก็ได้นำปัจจัยเหล่านี้มาใช้เช่นกัน โดยมีตัวอย่างเปรียบเทียบการออกแบบให้ได้ User Experience ที่ดีและไม่ดี ดังรูปที่ 2.21

<p>Sign Up Sign In</p> <ul style="list-style-type: none"> - Same Verbs - Similar Prepositions - Same Syllables - Same Style <p>Confuses Users</p>	<p>Sign Up Log In</p> <ul style="list-style-type: none"> - Different Verbs - Similar Prepositions - Same Syllables - Same Style <p>More Distinct</p>	<p>Register Sign In</p> <ul style="list-style-type: none"> - Different Verbs - Different Phrases - Different Syllables - Different Style <p>Very Distinct</p>
<p>✘ Bad</p>	<p>– Okay</p>	<p>✓ Good</p>

ภาพที่ 2.21 ตัวอย่างเปรียบเทียบการออกแบบให้ได้ User Experience ที่ดีและไม่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่มา : <https://s-media-cache-ak0.pinimg.com/originals/13/f0/79/13f07917b0dc1fb8644316aac0c52b3a.png>

2.21 Firebase

Firebase คือ Project ที่ถูกออกแบบมาให้เป็น API และ Cloud Storage สำหรับ Real time Application รองรับหลาย Platform ทั้ง IOS App, Android, Web App

Firebase ถูกสร้างขึ้นเพื่อให้นักพัฒนาสามารถผสมและจับคู่เพื่อให้พอดีกับความต้องการของตน เริ่มต้น Firebase เป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและชิงค์ข้อมูล

Firebase มีการให้บริการ ดังนี้

1. Firebase Analysis บริการวิเคราะห์ข้อมูล ดึงเทคโนโลยีมาจาก Google Analytics แคมป์ยังเปิดให้ใช้ฟรีแบบไม่จำกัดปริมาณข้อมูลใดๆ
2. Firebase Cloud Messaging (FCM) ระบบส่งข้อความแจ้งเตือน ใช้งานฟรีไม่จำกัดปริมาณข้อความ
3. Firebase Storage บริการพื้นที่เก็บข้อมูล เอาไว้เก็บภาพ วิดีโอ หรือไฟล์ขนาดใหญ่จากแอปพลิเคชันของผู้ใช้สร้างอยู่บน Google Cloud Storage
4. Firebase Remote Config ตัวช่วยอัปเดตคอนฟิกของแอปพลิเคชัน สำหรับปรับแต่งค่าต่างๆ ในแอปพลิเคชันจากระยะไกล (เช่น เกมที่อยากปรับสมดุลของเกมตลอดเวลา) สามารถใช้ร่วมกับ Firebase Analytics เพื่อกำหนดผู้ใช้งานแยกเป็นกลุ่มๆ ได้
5. Firebase Crash Reporting ตัวรายงานการ Clash ของแอปพลิเคชัน รองรับทั้ง IOS และ Android
6. Firebase Test Lab for Android บริการทดสอบแอปพลิเคชันบนฮาร์ดแวร์จริง
7. Firebase Notifications เป็นคอนโซลสำหรับนักพัฒนา เพื่อยิงข้อความผ่าน FCM ไปยังผู้ใช้ สำหรับโฆษณาหรือกระตุ้นให้ผู้ใช้กลับมาเปิดแอปพลิเคชันของเรา (เช่น แจกของในเกม)
8. Firebase Dynamic Links บริการ URL กลางที่สามารถชี้ทางไปยังหน้าต่างๆ แปรผันตามอุปกรณ์หรือคุณสมบัติของผู้ใช้ (เช่น แต่ละประเทศกดลิงค์เดียวกัน เข้าคนละหน้ากัน)
9. Firebase Invites ระบบเชิญเพื่อนมาใช้แอปพลิเคชัน มีพีเจอร์ referral คนชวนได้สิทธิประโยชน์

10. Firebase App Indexing เปลี่ยนชื่อมาจาก Google App Indexing ที่ช่วยให้ Google Search ค้นเจอเนื้อหาภายในแอปพลิเคชัน

ดังนั้น Firebase จึงครอบคลุมทุกการบริการสำหรับพัฒนา Real time Application บริการเกือบทุกตัวของ Firebase ใช้งานได้ฟรีแบบไม่จำกัดปริมาณ ยกเว้น Test Lab, Storage, Real time Database, Hosting ที่คิดเงิน

เนื่องจากในปฏิบัติงานในครั้งนี้มีระบบการแจ้งเตือน เมื่อได้รับรายการสินค้าเข้ามาใหม่ แอปพลิเคชันตัวนี้จะได้รับการแจ้งเตือนซึ่งเป็นเหตุผลให้เลือกใช้งาน firebase ในการ database โดยมีโลโก้ ดังรูปที่ 2.22



รูปที่ 2.22 logo ของ Firebase

ที่มา : <https://firebase.google.com/>

ซึ่งก่อนที่ในปฏิบัติการในครั้งนี้ได้เลือกใช้ firebase นั้นได้ลองทำการค้นคว้า database ตัวอื่นมาแล้วเช่นกันซึ่งตัวที่ได้ลองทำการค้นคว่านั้นก็คือ MongoDB ซึ่ง MongoDB นั้นเป็น Database แบบ NoSQL โดยไม่ต้องเตรียมสร้าง table , field เอาไว้รอก่อน เหมือนกับ database ปกติ

ซึ่งความแตกต่างระหว่าง Firebase กับ MongoDB นั้นมี ดังตารางที่ 2.1

ตารางที่ 2.1 ตารางเปรียบเทียบ Firebase กับ MongoDB

	Firestore	MongoDB
ประสิทธิภาพ	Firestore ไม่สามารถจัดสรรประสิทธิภาพสูงเหมือน MongoDB ได้	MongoDB สามารถจัดสรรประสิทธิภาพสูงได้
ผู้พัฒนา	Google นั้นเป็นผู้พัฒนา Firestore	MongoDB Inc เป็นผู้พัฒนา MongoDB
ภาษาโปรแกรมที่รองรับ	รองรับ ภาษา C,Java และ JavaScript	รองรับภาษาที่หลากหลายกว่า เช่น C,C#,Java,JavaScript เป็นต้น
ความปลอดภัย	ไม่มีระบบรักษาความปลอดภัยเหมือนMongoDB	มีการจัดการระบบรักษาความปลอดภัยที่ดีกว่า Firestore
แอปพลิเคชัน	เหมาะสำหรับแอปพลิเคชันที่มีขนาดเล็ก	เหมาะกับแอปพลิเคชันที่มีขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางดังกล่าวจะเห็นได้ว่า MongoDB นั้นมีข้อดีที่ดีกว่า Firebase อยู่หลายด้าน แต่เหตุผลที่ปฏิบัติการครั้งนี้เลือกใช้ Firebase นั้นเป็นเพราะว่าปฏิบัติงานนี้เป็นเพียงส่วนประกอบของ ชิ้นงานขนาดใหญ่ซึ่งในส่วนอื่น ๆ นั้นได้มีงานใช้งานในรูปแบบ Firebase รวมถึงการใช้ OneSignal นั้น จำเป็นที่ต้องใช้ Firebase ในการเข้าสู่ระบบ

2.22 OneSignal

เป็นการบริการ push notifications ไปยังผู้ใช้งานที่รองรับหลาย platform จุดประสงค์ ต้องการให้ผู้ใช้งานส่ง notification ไปยังกลุ่มเป้าหมายหลากหลาย segment โดยการใช้วิธี filter กลุ่มผู้ที่ เข้ามาใช้งานเว็บ

โดยการส่งข้อความเพียงแค่ว่า create app เพื่อให้ได้ app id บน OneSignal แล้วนำ script ไปติดตั้งที่เว็บไซต์ของเรา OneSignal จะเก็บ session ของผู้ใช้งานไว้ให้เรานำมา filter เพื่อสร้าง กลุ่มเป้าหมาย และสามารถส่งข้อความ notification ไปยังกลุ่มเป้าหมายได้ที่เว็บของ OneSignal ได้เลย

แต่เราจะใช้บริการ OneSignal ในการทำ desktop notification ไปยังเจ้าหน้าที่ซึ่งต้องมีการ ตั้งค่าเฉพาะเพิ่มเติมให้กับกลุ่ม user เพื่อให้สามารถแจ้งเตือนไปยังเจ้าหน้าที่ได้ถูกต้อง

ซึ่งในปฏิบัติการจะมีการแจ้งเตือนอยู่ 2 ส่วนก็คือส่วนที่รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่ กับส่งการแจ้งเตือนกลับไปหาเว็บแอปพลิเคชันว่ารายการสินค้านี้ได้ถูกยกเลิกหรือรายการสินค้านี้กำลัง เตรียมพร้อมสำหรับการจัดส่ง โดยมีโลโก้ ดังรูปที่ 2.23 และตัวอย่างเว็บ OneSignal ดังรูปที่ 2.24



รูปที่ 2.23 logo ของ OneSignal

ที่มา : <https://www.mobilhanem.com/android-one-signal-push-notification/>

OneSignal (/) SETTINGS MESSAGES USERS DELIVERY MOBILEAPP ?

Users

Segments All Users Test Users

SEGMENTS: SUBSCRIBED USERS Player ID

IMPORT/EXPORT

ACTIONS	CHANNEL	SUBSCRIBED	LAST ACTIVE	FIRST SESSION	DEVICE	SESSIONS	APP VERSION	COUNTRY	ROOTED	LOCATION POINT	USAGE DURATION	LANGUAGE CODE	PLAYER ID
OPTIONS -		✓	11/15/18, 12:11:20 pm	10/10/18, 6:06:55 pm	SM-G610F (7.0)	415	1	TH	No		78800	th	8197c68-c-
OPTIONS -		✓	11/15/18, 9:59:54 am	10/20/18, 10:56:32 am	Win32 (7.0)	2		TH	No		0	th	3c98dba9-1
OPTIONS -		✓	11/09/18, 1:50:54 pm	11/09/18, 1:50:54 pm	Win32 (7.0)	1		TH	No		0	en	45ec2429-5
OPTIONS -		✓	11/09/18, 11:53:39 am	11/09/18, 11:53:39 am	Win32 (7.0)	1		TH	No		0	en	8106a929-d
OPTIONS -		✓	11/09/18, 11:53:16 am	11/09/18, 11:53:16 am	Win32 (7.0)	1		TH	No		0	en	5c1f1158-b
OPTIONS -		✓	11/08/18, 2:05:32 pm	10/12/18, 2:14:46 pm	Win32 (7.0)	28		TH	No		0	th	912d8b1f-8
OPTIONS -		✓	11/01/18, 3:49:26 pm	11/01/18, 2:08:47 pm	Android SDK built for x86 (8.0.0)	2	1	TH	Yes		0	en	fae54866-2
OPTIONS -		✓	10/19/18, 10:46:22 am	10/16/18, 10:35:53 am	Win32 (6.9)	18		TH	No		0	en	e56459b9-a



รูปที่ 2.24 ตัวอย่างเว็บ OneSignal

บทที่ 3

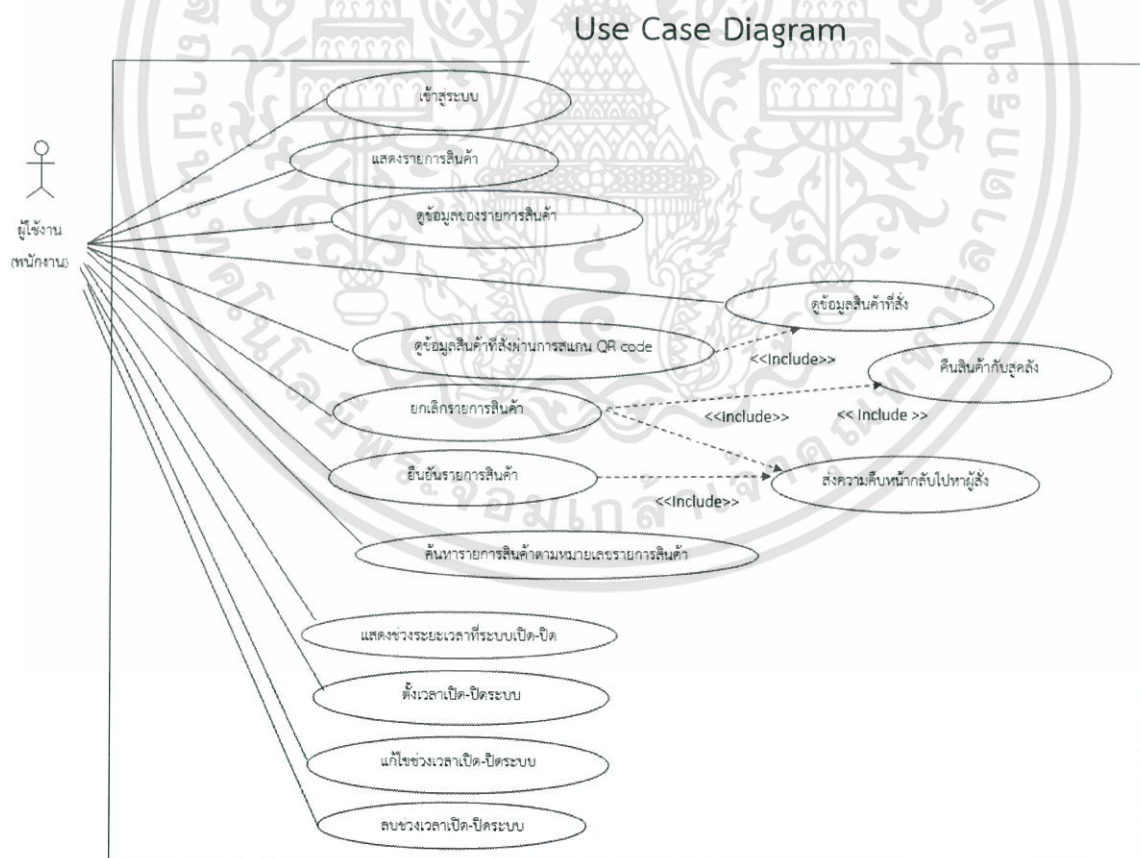
วิธีดำเนินการ

ขั้นตอนพัฒนาแอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้า ซึ่งมีทฤษฎีต่างๆที่ใช้เป็นข้อมูลประกอบการจัดทำโครงการ

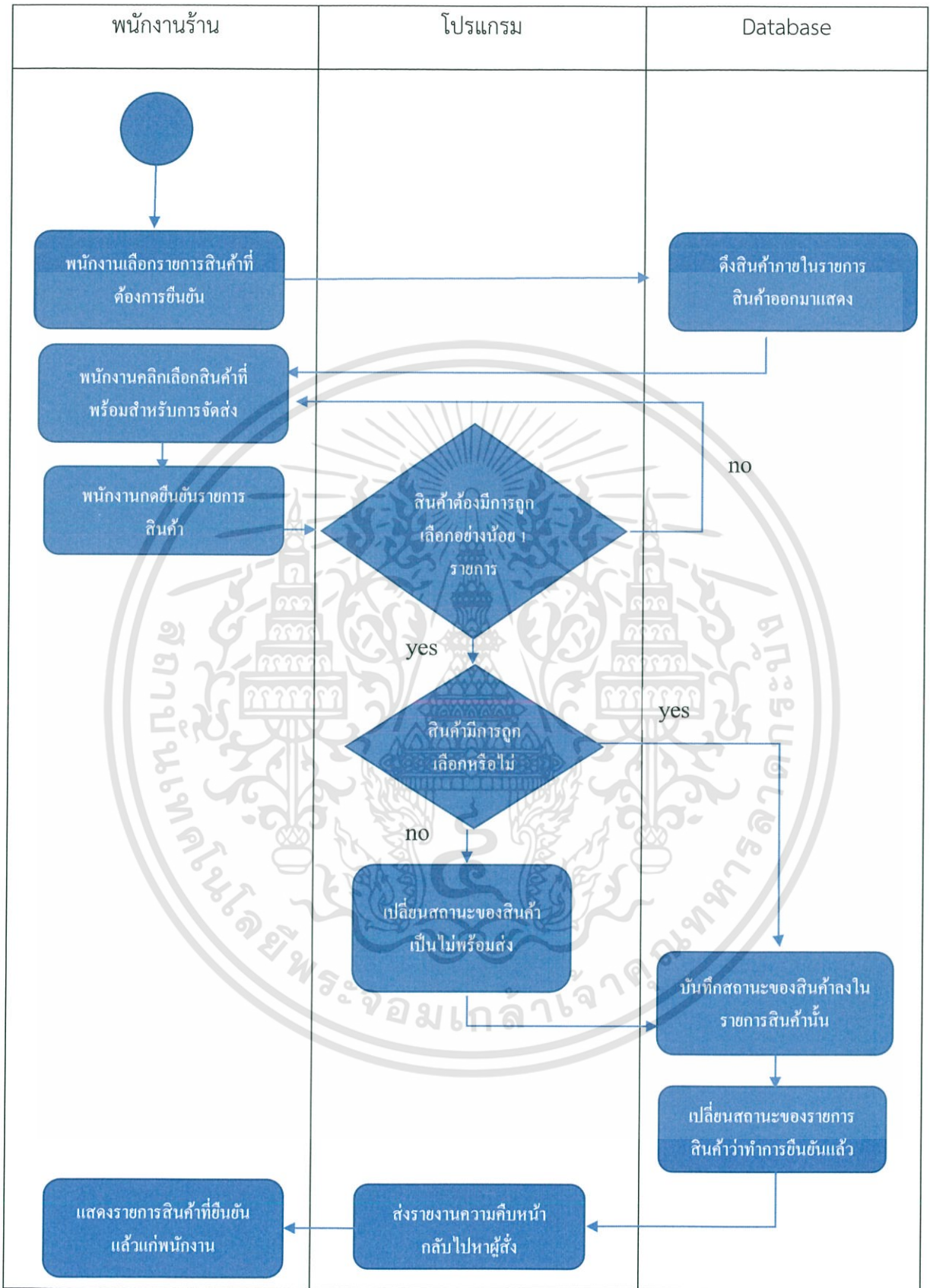
- 3.1 การวิเคราะห์ระบบ
- 3.2 โครงสร้างฐานข้อมูล
- 3.3 การพัฒนาระบบ

3.1 การวิเคราะห์ระบบ

กระบวนการต่อไปนี้จะเป็นการวิเคราะห์ระบบงาน เพื่อทำการวิเคราะห์และอธิบายถึงปัญหาที่เกิดขึ้น โดยขั้นตอนในการวิเคราะห์ระบบนั้น โดยแอปพลิเคชันได้นำเครื่องมือในการวิเคราะห์ระบบ ซึ่งประกอบด้วย แผนภาพ Use case diagram ดังรูปที่ 3.1, Activity Diagram ของการยืนยันรายการสินค้า ดังรูปที่ 3.2, Activity Diagram การยกเลิกรายการสินค้า ดังรูปที่ 3.3, Activity Diagram การค้นหาสินค้าในรายการสินค้า ผ่านการสแกน QR code ดังรูปที่ 3.4, flow chart หลักของระบบ ดังรูปที่ 3.5, แผนภาพการไหลของข้อมูล DFD Level 0 ดังรูปที่ 3.6 และแผนภาพ ER diagram ดังรูปที่ 3.7

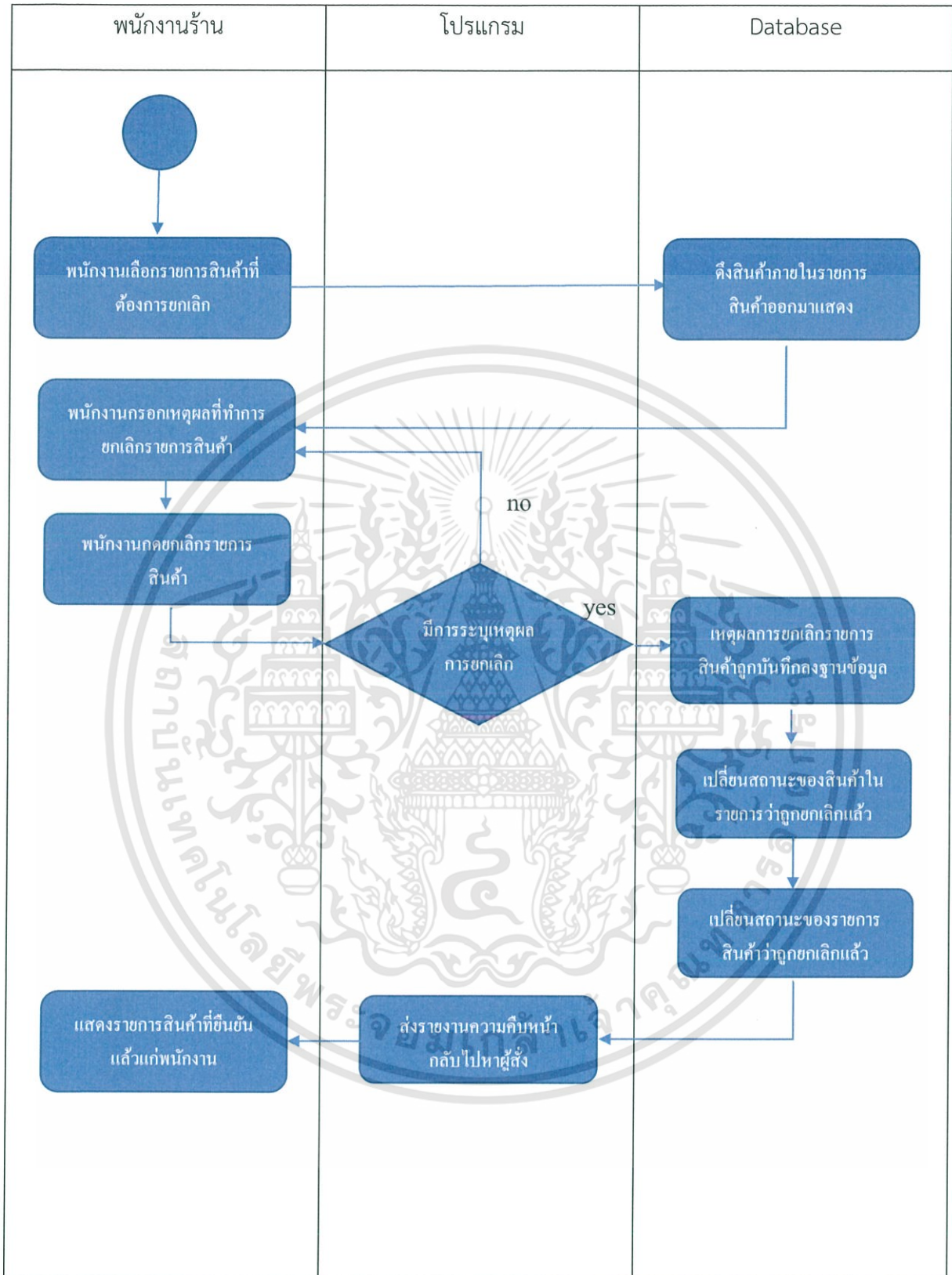


รูปที่ 3.1 แผนภาพ Use Case Diagram

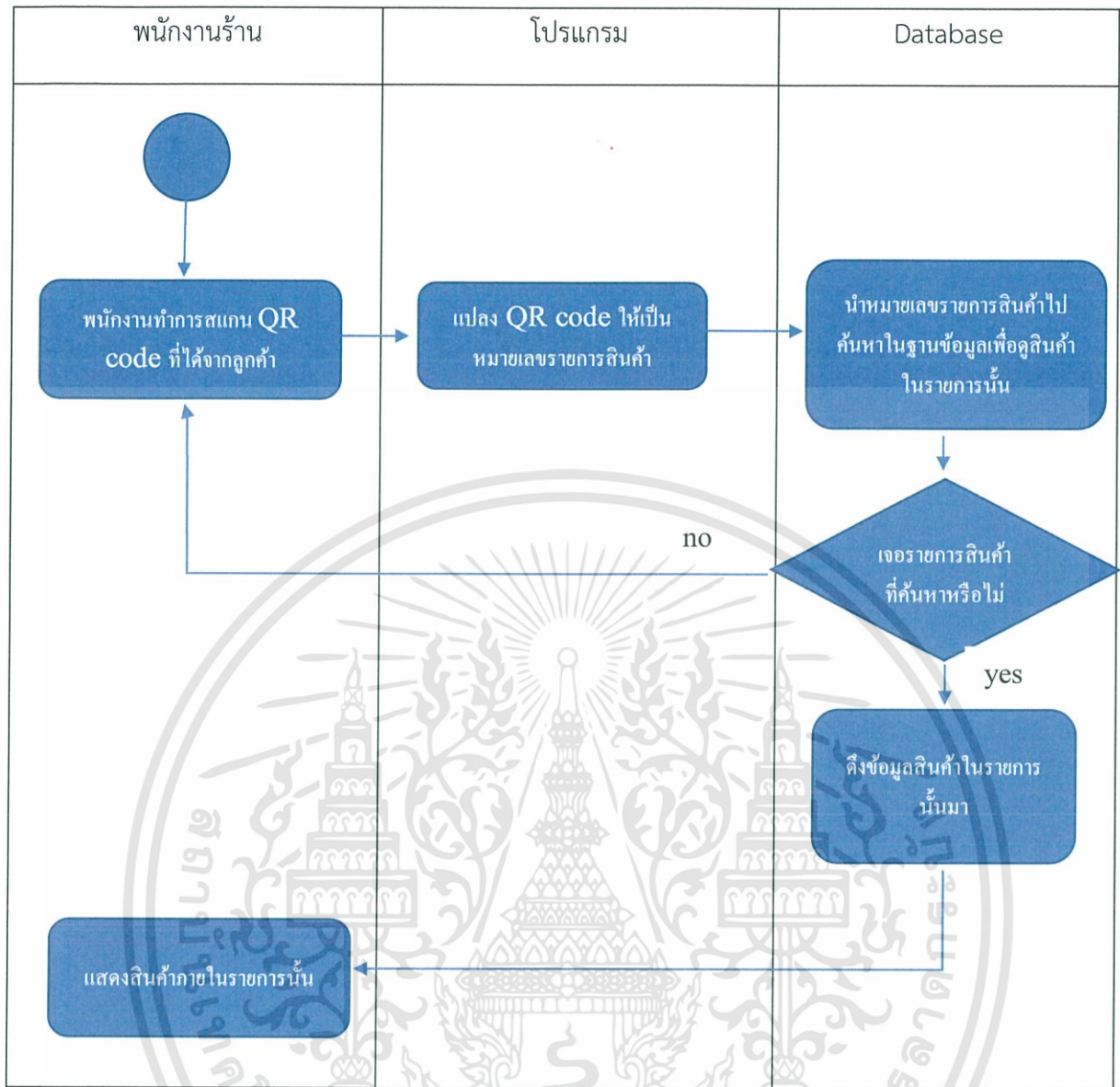


รูปที่ 3.2 Activity Diagram การยืนยันรายการสินค้า

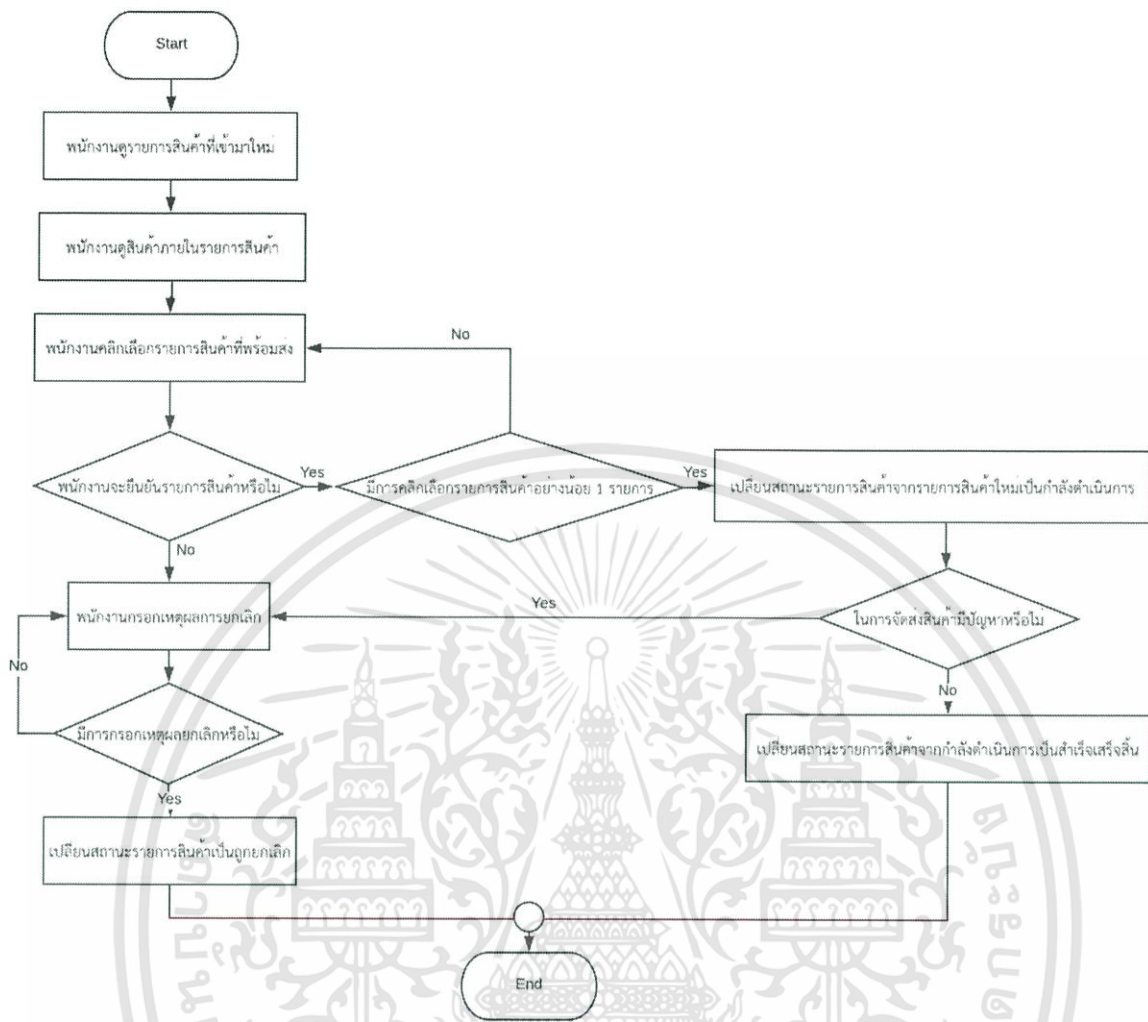
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Activity Diagram การยกเลิกรายการสินค้า

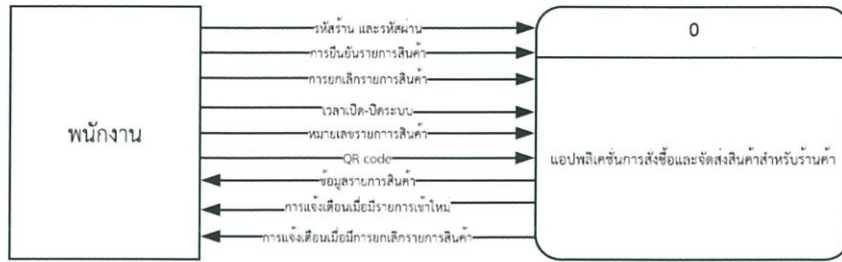


รูปที่ 3.4 Activity Diagram การค้นหาสินค้าในรายการสินค้า
ผ่านการสแกน QR code



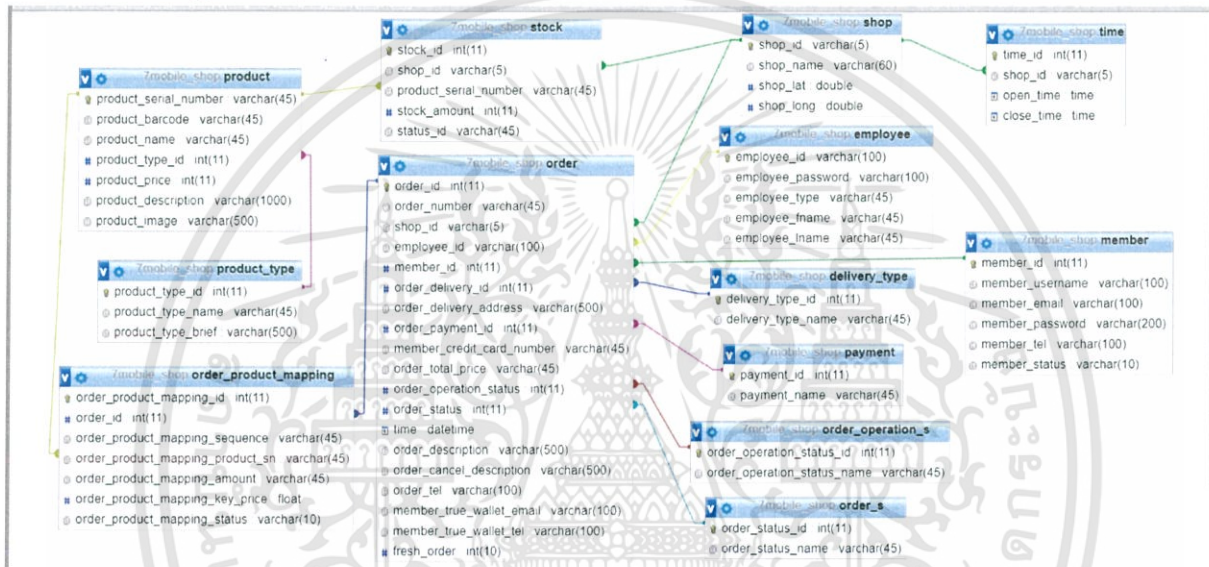
รูปที่ 3.5 Flow chart ของระบบหลัก

Data Flow Diagram



รูปที่ 3.6 แผนภาพการไหลของข้อมูล(Data flow diagram Level 0)

3.2 โครงสร้างฐานข้อมูลระบบ (ER Diagram)



รูปที่ 3.7 แผนภาพ ER diagram

ตารางที่ 3.1 ตารางพนักงาน (employee)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
employee_id	varchar(100)	PK	หมายเลขประจำตัวพนักงาน
employee_password	varchar(100)		รหัสการเข้าใช้ของพนักงาน
employee_type	varchar(45)		ประเภทของพนักงาน
employee_fname	varchar(45)		ชื่อจริงพนักงาน
employee_lname	varchar(45)		นามสกุลพนักงาน

ตารางที่ 3.2 ตารางผู้ใช้งาน (member)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
member_id	int(11)	PK	หมายเลขประจำตัวของผู้ใช้งาน
member_username	varchar(100)		ชื่อผู้ใช้งาน
member_email	varchar(100)		อีเมลผู้ใช้งาน
member_password	varchar(200)		รหัสผ่านผู้ใช้งาน
member_tel	varchar(100)		เบอร์โทรศัพท์ผู้ใช้งาน
member_status	varchar(10)		สถานะผู้ใช้งาน

ตารางที่ 3.3 ตารางประเภทการส่ง (delivery_type)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
delivery_type_id	int(11)	PK	หมายเลขประจำตัวประเภทการส่ง
delivery_type_name	varchar(45)		ประเภทของการส่ง

ตารางที่ 3.4 ตารางรายการสินค้า (order)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
order_id	int(11)	PK	หมายเลขประจำตัวรายการ สินค้า
order_number	varchar(45)		หมายเลขรายการสินค้า
shop_id	varchar(5)	FK	หมายเลขประจำตัวร้านค้า
employee_id	varchar(100)	FK	หมายเลขประจำตัวพนักงาน
member_id	int(11)	FK	หมายเลขประจำตัวผู้ใช้งาน
order_delivery_id	int(11)	FK	หมายเลขประจำตัวประเภทการ ส่ง
order_delivery_address	varchar(500)		ที่อยู่ที่จะรับสินค้า
order_payment_id	int(11)	FK	หมายเลขประจำตัวประเภทการ จ่ายเงิน
member_credit_card_number	varchar(45)		หมายเลขบัตรเครดิต
order_total_price	varchar(45)		ราคารวมของสินค้า
order_operation_status	int(11)	FK	หมายเลขสถานะของรายการ สินค้า
order_status	int(11)	FK	หมายเลขสถานะของรายการ สินค้า
time	datetime		เวลาที่สั่งซื้อสินค้า
order_description	varchar(100)		รายละเอียดเพิ่มเติมของรายการ สินค้า
order_cancel_description	varchar(500)		รายละเอียดการยกเลิกรายการ สินค้า
order_tel	varchar(100)		หมายเลขโทรศัพท์ที่ไว้ติดต่อ
member_true_wallet_email	varchar(100)		อีเมลทรูมันนี่ วอลเล็ทของ ผู้ใช้งาน

member_true_wallet_tel	varchar(100)		เบอร์โทรศัพท์ทรูมันนี่ วอลเล็ต ของผู้ใช้งาน
fresh_order	int(10)		หมายเลขสถานะของรายการ สินค้า

ตารางที่ 3.5 ตารางสถานะความคืบหน้า(order_operation_s)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
order_operation_status_id	int(11)	PK	หมายเลขประจำตัว สถานะ
order_operation_status_name	varchar(45)		รายละเอียดสถานะ

ตารางที่ 3.6 ตารางการเชื่อมโยงสินค้า (order_product_mapping)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
order_product_mapping_id	int(11)	PK	หมายเลขประจำตัว การเชื่อมโยงสินค้า
order_id	int(11)	FK	หมายเลขประจำตัว รายการสินค้า
order_product_mapping_sequence	varchar(45)		ลำดับหมายเลขการ เชื่อมโยงสินค้า
order_product_mapping_product_sn	varchar(45)	FK	หมายเลขสินค้า
order_product_mapping_amount	varchar(45)		จำนวนสินค้า
order_product_mapping_key_price	float		ราคาของสินค้า
order_product_mapping_status	varchar(10)		สถานะของสินค้า

ตารางที่ 3.7 ตารางสถานะของรายการสินค้า (order_s)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
order_status_id	int(11)	PK	หมายเลขประจำตัวสถานะ
order_status_name	varchar(45)		รายละเอียดสถานะ

ตารางที่ 3.8 ตารางการจ่ายเงิน (payment)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
payment_id	int(11)	PK	หมายเลขประจำตัว ประเภทการจ่ายเงิน
payment_name	varchar(45)		ชื่อประเภทการจ่ายเงิน

ตารางที่ 3.9 ตารางสินค้า (product)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
product_serial_number	varchar(45)	PK	หมายเลขสินค้า
product_barcode	varchar(45)		รหัสบาร์โค้ดของสินค้า
product_name	varchar(45)		ชื่อสินค้า
product_type_id	int(11)	FK	หมายเลขประจำตัว ประเภทของสินค้า
product_price	int(11)		ราคาสินค้า
product_description	varchar(1000)		รายละเอียดสินค้า
product_image	varchar(500)		รูปภาพสินค้า

ตารางที่ 3.10 ตารางประเภทสินค้า (product_type)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
product_type_id	int(11)	PK	หมายเลขประจำตัวประเภทสินค้า
product_type_name	varchar(45)		ชื่อประเภทสินค้า
product_type_brief	varchar(100)		คำอธิบายของประเภทสินค้า

ตารางที่ 3.11 ตารางร้านค้า (shop)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
shop_id	varchar(5)	PK	หมายเลขประจำตัวร้านค้า
shop_name	varchar(60)		ชื่อสาขาของร้านค้า
shop_lat	double		ตำแหน่งละติจูดของร้านค้า
shop_long	double		ตำแหน่งลองจิจูดของร้านค้า

ตารางที่ 3.12 ตารางคลังสินค้า (stock)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
stock_id	int(11)	PK	หมายเลขประจำตัวคลังสินค้า
shop_id	varchar(5)	FK	หมายเลขประจำตัวร้านค้า
product_serial_number	varchar(45)	FK	หมายเลขสินค้า
stock_amount	int(11)		จำนวนสินค้าในคลัง
status_id	varchar(45)		สถานะของสินค้า

ตารางที่ 3.13 ตารางเวลา (time)

ชื่อคอลัมน์ (Column Name)	ชนิดข้อมูล (Data Type)	ความสัมพันธ์ (Relation)	รายละเอียด (Detail)
time_id	int(11)	PK	หมายเลขประจำตัวเวลา
shop_id	varchar(5)	FK	หมายเลขประจำตัวร้านค้า
open_time	time		เวลาเปิดระบบ
close_time	time		เวลาปิดระบบ

3.3 การพัฒนาระบบ

การพัฒนาซอฟต์แวร์ ตามปกติแล้วจะประกอบไปด้วยกลุ่มกิจกรรม 3 ส่วนหลักๆด้วยกัน คือ การวิเคราะห์ (Analysis) การออกแบบ (Design) และการนำไปใช้ (Implementation) ซึ่งกิจกรรมทั้งสามนี้สามารถใช้งานได้ดีกับโครงการซอฟต์แวร์ขนาดเล็ก ในขณะที่โครงการซอฟต์แวร์ขนาดใหญ่ มักจำเป็นต้องใช้แบบแผนการพัฒนาซอฟต์แวร์ตามแนวทางของ SDLC (Systems Development Life Cycle) จนครบทุกกิจกรรม โดยมีแผนภาพทฤษฎีการวิเคราะห์และออกแบบระบบ ดังรูปที่ 3.3 ขั้นตอนที่ใช้ศึกษาขบวนการวิเคราะห์และออกแบบระบบ ประกอบด้วย

3.3.1 การทำความเข้าใจกับปัญหา (Project Identification and Selection)

ศึกษาปัญหาซึ่งเกิดจากไม่เป็นระบบในการสั่งซื้อสินค้า เนื่องจากในอดีตเป็นรูปแบบการโทรเข้ามาสั่งสินค้า ซึ่งทำให้เกิดปัญหาคือ การจัดเรียงคิวลูกค้า ความถูกต้องของรายการสินค้า ซึ่งทำให้พนักงานที่ร้านทำงานหนักขึ้น จึงนำแอปพลิเคชันมาเพิ่มความอำนวยความสะดวกให้แก่พนักงาน และสามารถเพิ่มจำนวนยอดขายให้แก่ร้านค้าได้เช่นกัน

3.3.2 การรวบรวมข้อมูล (Project Initiating and Planning)

รวบรวมข้อมูลจากอินเทอร์เน็ตถึงวิธีเขียนแอปพลิเคชันที่เหมาะสมกับงาน และสอบถามพี่เลี้ยงและทีมงานว่าเขียนแบบไหนถึงจะเหมาะกับการใช้งาน เหมาะกับบุคลากรในหน่วยงาน รวมถึงรวบรวมข้อมูลจากส่วนเว็บแอปพลิเคชันที่จะส่งข้อมูลเข้ามา และต้องสะดวกหากพี่เขาจะแก้ไข ปรับปรุงในภายหลัง ในปฏิบัติงานครั้งนี้จึงใช้ Android studio และ NodeJS ในฝั่ง Web service ที่มีการใช้ภาษา JAVA, JAVASCRIPT, XML และ SQL มาใช้งาน

3.3.3 การวิเคราะห์ระบบ (Analysis)

จากนั้นเราจึงเขียนแผนภาพทิศทางการไหลของข้อมูล (Data Flow Diagram) และผังงานระบบ (System Flowchart) เพื่อให้เห็นภาพรวมและขั้นตอนการเขียนแอปพลิเคชันมากขึ้นและทำให้สะดวกในขั้นตอนต่อไปในการปฏิบัติงาน

3.3.4 การออกแบบระบบ (Logical Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ 37 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบระบบซึ่งเป็นส่วนที่ทำการติดต่อกับผู้ใช้งานระบบ จะต้องออกแบบให้เหมาะสมและ
ผู้ใช้งานระบบจะต้องใช้งานง่ายไม่ซับซ้อนจนเกินไป และการออกแบบระบบ จะเป็นการออกแบบในส่วนของ
ยูสเคสไดร์แกรม(Use Case Diagram) ยูสเคสเทมเพลท (Use Case Template) แอคทิวิตีไดร์แกรม (Activity
Diagram) และดาต้าดิคชันนารี Data Dictionary) เพื่อให้เราเห็นภาพรวมของแอปพลิเคชันและจะทำให้
พัฒนาระบบง่ายขึ้น

3.3.5 การพัฒนาระบบ (Physical Design)

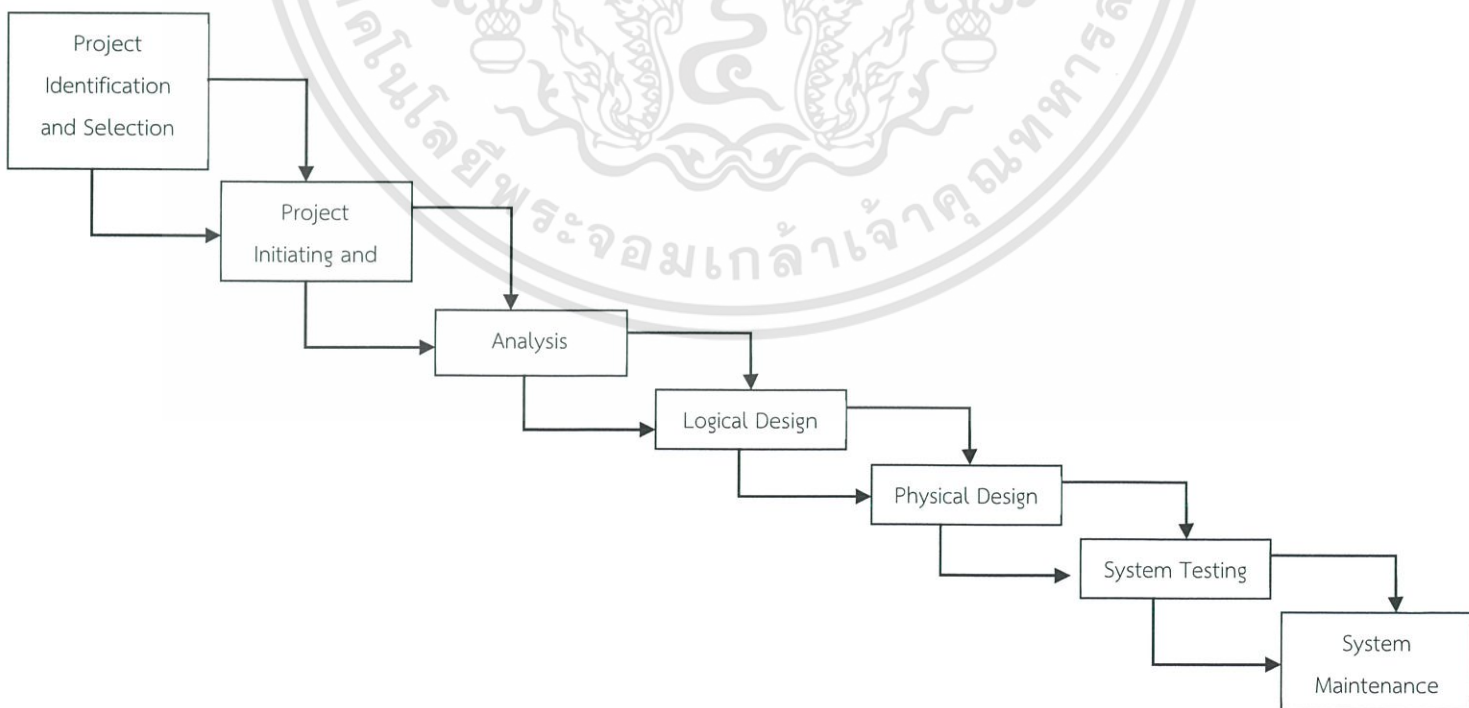
พัฒนาระบบตามแผนงานที่วางเอาไว้ในขั้นตอนที่ 4 เริ่มเขียนแอปพลิเคชันในโปรแกรม Android
Studio โดยใช้ภาษา Java และ XML เพราะมีความเหมาะสมกับระบบงาน และสามารถปรับปรุงแก้ไข ฟีเจอร์
ที่มเคยใช้ Android Studio ในการเขียนมาก่อน จึงสามารถถามเมื่อพบเจอปัญหาได้ เพิ่มเติมแอปพลิเคชันใน
ภายหลังได้ง่าย และมีคู่มือในการใช้เป็นจำนวนมากเพราะจำนวนผู้ใช้งานโปรแกรมนี้มีจำนวนมาก และในส่วน
เชื่อมต่อ Web service ได้ใช้ โปรแกรม Visual Studio Code โดยใช้ภาษา JAVASCRIPT ในการเขียน เป็น
รูปแบบ NodeJS ใช้ภาษา SQL ในการเชื่อมต่อกับ database โดยมี API เป็นตัวกลางในการเชื่อมต่อกับ
ฐานข้อมูล

3.3.6 การทดสอบระบบ (System Testing)

ตรวจสอบบัค (Bug) ของแอปพลิเคชันด้วยตนเองแล้วให้พี่เลี้ยงและทีมงานทดลองใช้แอปพลิเคชันดู
ว่าแอปพลิเคชันสมควรมีการแก้ไข เพิ่มเติมส่วนไหนบ้าง และกีดำเนินการแก้ไข

3.3.7 การบำรุงรักษาระบบ (System Maintenance)

ส่งมอบแอปพลิเคชันให้กับพี่เลี้ยงและทีมงานไปดำเนินการบำรุงรักษาและแก้ไขแอปพลิเคชันหลังที่ใช้
งานจริง แล้วนำไปพัฒนาให้สมบูรณ์ยิ่งขึ้นในอนาคต



รูปที่ 3.8 แผนภาพทฤษฎีการวิเคราะห์และออกแบบระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ 38 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

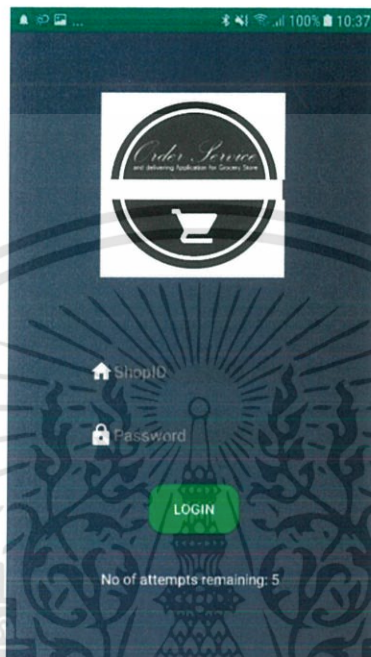
ผลการทดลอง

จากการพัฒนาแอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้าเสร็จ จำเป็นต้องมีการทดสอบการทำงานของแอปพลิเคชันเพื่อตรวจสอบว่าจะให้ผลลัพธ์นั้นสามารถอำนวยความสะดวกซื้อในการตรวจเช็คสินค้าในรายการที่สั่งมา ลดปัญหาด้านการผิดพลาดของพนักงานในการจัดเรียงคิวรายการสินค้า เพื่อสะดวกในการตรวจสอบสินค้าในรายการที่อยู่ในระหว่างการดำเนินการ รายการที่ถูกยกเลิก และรายการที่เสร็จสมบูรณ์ได้ มีการแจ้งเตือนกลับไปยังผู้สั่งซื้อเพื่อให้ผู้สั่งซื้อสามารถดูความคืบหน้าของรายการสินค้านั้นๆ ได้และมีประสิทธิภาพเหมาะสมกับการนำไปใช้งานจริง

- 4.1 สามารถเข้าใช้งานระบบผ่านการล็อกอินได้
- 4.2 สามารถดูการแสดงรายการสินค้าได้
- 4.3 สามารถดูข้อมูลของรายการสินค้าได้
- 4.4 สามารถดูข้อมูลสินค้าที่สั่งได้
- 4.5 สามารถยกเลิกรายการสินค้าได้
- 4.6 สามารถยืนยันรายการสินค้าได้
- 4.7 สามารถดูการแสดงประวัติรายการสินค้าทั้งหมดได้
- 4.8 สามารถดูการแสดงประวัติรายการสินค้าที่สำเร็จได้
- 4.9 สามารถดูการแสดงประวัติรายการสินค้าที่ยกเลิกได้
- 4.10 สามารถดูสถานะสินค้าในประวัติรายการสินค้าได้
- 4.11 สามารถดูการแสดงช่วงระยะเวลาที่ระบบเปิด-ปิดได้
- 4.12 สามารถตั้งเวลาเปิด-ปิดระบบได้
- 4.13 สามารถแก้ไขช่วงเวลาเปิด-ปิดระบบได้
- 4.14 สามารถลบช่วงเวลาเปิด-ปิดระบบได้
- 4.15 สามารถได้รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่ได้
- 4.16 สามารถได้รับการแจ้งเตือนเมื่อผู้ใช้งานยกเลิกรายการสินค้าได้
- 4.17 สามารถส่งการแจ้งเตือนกลับไปหาผู้ใช้งานเพื่อบอกความคืบหน้าของรายการสินค้าได้
- 4.18 สามารถดูสินค้าในรายการผ่านการสแกน QR code ได้

4.1 การเข้าใช้งานระบบผ่านการล็อกอิน

พนักงานต้องทำการกรอกรหัสร้าน และรหัสผ่านก่อนใช้งานระบบ ซึ่งระบบจะมีการรักษาความปลอดภัย คือระบบจะอนุญาตให้พนักงานลองเข้าระบบได้จำนวน 5 ครั้งดังรูปที่ 4.1



รูปที่ 4.1 การเข้าใช้งานระบบผ่านการล็อกอิน

4.2 การแสดงรายการสินค้า

เมื่อพนักงานเข้าสู่ระบบมาแล้วจะมีการแสดงรายการสินค้าของร้านที่ได้ทำการเข้าสู่ระบบ โดยมีการแสดงสถานะอยู่ 2 รูปแบบ คือ รายการสินค้าที่ใหม่ กับรายการสินค้าที่เคยถูกเปิดอ่านดูแล้ว ดังรูปที่ 4.2 โดยรายการสินค้าเข้ามาใหม่นั้นจะมีข้อสี่เหลี่ยมเพื่อแสดงถึงสถานะของรายการสินค้านั้นว่ายังไม่เคยถูกเปิดดู และในส่วนหน้าแสดงรายการสินค้าจะมีตัวเลขแสดงจำนวนรายการสินค้าด้วย



รูปที่ 4.2 การแสดงรายการสินค้า

4.3 การดูข้อมูลของรายการสินค้า

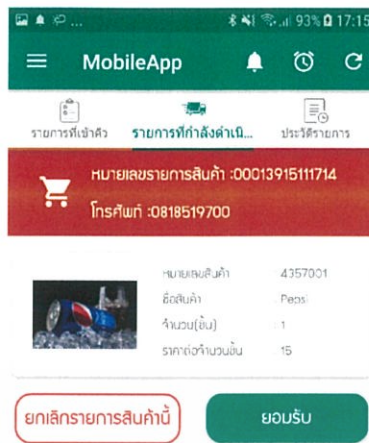
โดยจะมีรายละเอียดของรายการสินค้า ดังรูปที่ 4.3 โดยรายละเอียดจะประกอบด้วย หมายเลขรายการสินค้า วันที่-เวลาที่สั่งรายการสินค้า ประเภทการชำระเงิน ราคารวม เบอร์ติดต่อ ที่อยู่จัดส่ง และรายละเอียดเพิ่มเติมสำหรับการจัดส่ง



รูปที่ 4.3 รายละเอียดข้อมูลของรายการสินค้า

4.4 การดูข้อมูลสินค้าที่สั่ง

เมื่อพนักงานกดเลือกดูรายการสินค้าจะมีข้อมูลของสินค้าภายในรายการสินค้านั้นขึ้นมาแสดง ดังรูปที่ 4.4 โดยรายละเอียดของสินค้านั้นจะประกอบไปด้วย หมายเลขสินค้า ชื่อสินค้า จำนวนที่สั่งซื้อ และราคาต่อจำนวนชิ้น พร้อมมีรูปของสินค้าประกอบด้วย



รูปที่ 4.4 รายละเอียดข้อมูลสินค้าที่สั่ง

4.5 การยกเลิกรายการสินค้า

เมื่อพนักงานต้องการที่จะยกเลิกรายการสินค้านั้นๆ อาจจะมีสาเหตุ เนื่องจากสินค้าอาจจะความชำรุดหรือสินค้าหมด ก็จะมีหน้าสำหรับการกรอกสาเหตุนั้นๆดังรูปที่ 4.5 โดยพนักงานจำเป็นต้องกรอกสาเหตุทุกครั้งที่มีการยกเลิกแล้วสามารถใส่เหตุผลได้เกิน 2 บรรทัด



รูปที่ 4.5 หน้าตาสำหรับการกรอกรายละเอียดการยกเลิก

4.6 การยืนยันรายการสินค้า

เมื่อพนักงานได้ทำการเตรียมของเสร็จเรียบร้อยแล้วพร้อมที่จะดำเนินการส่ง พนักงานจะทำการกดยืนยันรายการสินค้าเพื่อเปลี่ยนสถานะรายการสินค้าและส่งการแจ้งเตือนกลับไปหาผู้ส่งให้รู้ถึงสถานะที่เปลี่ยนแปลงได้ ดังรูปที่ 4.6 และเมื่อยืนยันรายการเสร็จแล้วจะมีการแสดง ดังรูปที่ 4.7

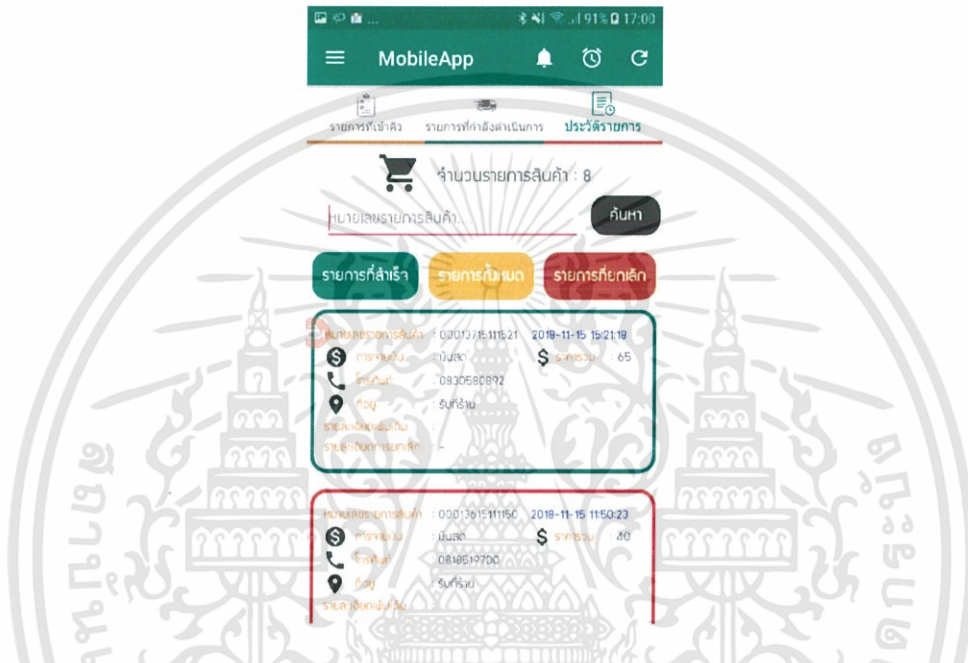


รูปที่ 4.6 การยืนยันรายการสินค้า

รูปที่ 4.7 ผลลัพธ์เมื่อทำการยืนยันสถานะของรายการสินค้าจะถูกเปลี่ยนไป และจะถูกนำไปแสดงอีกหน้า

4.7 การแสดงประวัติรายการสินค้าทั้งหมด

เมื่อพนักงานต้องการที่จะเรียกดูรายการสินค้าที่ส่งสำเร็จแล้ว ก็จะมีหน้าแสดงประวัติรายการสินค้าทั้งหมด ดังรูปที่ 4.8 โดยรายการสินค้าที่แสดงนั้นจะแสดงเฉพาะรายการที่ถูกสั่งในวันที่ดูและ 1วันก่อนวันที่ดูเท่านั้น โดยรายการที่ส่งสำเร็จจะมีขอบสีเขียว และรายการที่ถูกยกเลิกจะมีขอบสีแดง และมีบอกจำนวนรายการสินค้ากำกับด้วย และมีฟังก์ชันสำหรับในการค้นหาโดยการกรอกหมายเลขรายการสินค้าเพื่อทำการค้นหา



รูปที่ 4.8 การแสดงประวัติรายการสินค้าทั้งหมด

4.8 การแสดงประวัติรายการสินค้าที่สำเร็จ

พนักงานสามารถเลือกดูประวัติรายการสินค้าเฉพาะรายการสินค้าที่สำเร็จเท่านั้นได้ ดังรูปที่ 4.9 โดยจะแสดงเฉพาะรายการที่มีกรอบสีเขียวเท่านั้น



รูปที่ 4.9 การแสดงประวัติรายการสินค้าที่สำเร็จ

4.9 การแสดงประวัติรายการสินค้าที่ยกเลิก

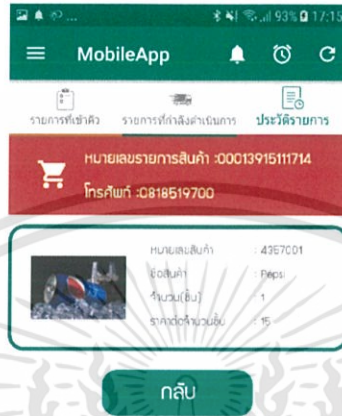
พนักงานสามารถเลือกดูประวัติรายการสินค้าเฉพาะรายการสินค้าที่ยกเลิกเท่านั้นได้ ดังรูปที่ 4.10 โดยจะแสดงเฉพาะรายการที่มีกรอบสีแดงเท่านั้น



รูปที่ 4.10 การแสดงประวัติรายการสินค้าที่ยกเลิก

4.10 การดูสถานะสินค้าในประวัติรายการสินค้า

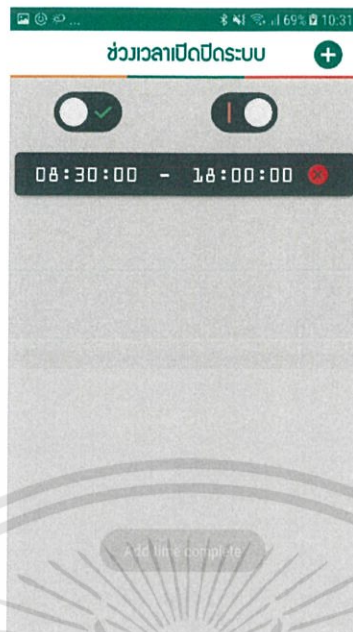
พนักงานสามารถดูสถานะสินค้าในประวัติรายการสินค้าได้ ดังรูปที่ 4.11 โดยรายละเอียดของสินค้าก็จะประกอบไปด้วยหมายเลขสินค้า จำนวนที่สั่งซื้อ ราคาจำนวนต่อชิ้น และสถานะของสินค้าถ้าเป็นสีเขียว แสดงว่ารายการสินค้านี้ถูกจัดส่งเป็นที่เรียบร้อยแล้วหากเป็นสีแดงแสดงว่ามีการยกเลิกสินค้านั้น



รูปที่ 4.11 การดูสถานะสินค้าในประวัติรายการสินค้า

4.11 แสดงช่วงระยะเวลาที่ระบบเปิด-ปิด

พนักงานสามารถดูช่วงระยะเวลาเปิด-ปิดระบบได้ ดังรูปที่ 4.12 เนื่องจากร้านแต่ละร้านจะมีเวลาเปิด-ปิดระบบไม่ตรงกันจึงทำให้ร้านแต่ละร้านต้องมากำหนดช่วงระยะเวลาเปิด-ปิดระบบเองโดยสามารถกำหนดมากกว่า 1 ช่วงได้โดยสามารถคลิกที่รูป + เพื่อทำการเพิ่มช่วงเวลาเปิด-ปิดระบบ



รูปที่ 4.12 แสดงช่วงระยะเวลาที่ระบบเปิด-ปิด

4.12 ตั้งเวลาเปิด-ปิดระบบ

พนักงานสามารถตั้งเวลาเปิด-ปิดระบบได้โดยมีการแสดงหน้าต่าง ดังรูปที่ 4.13 และตั้งรูปที่ 4.14 โดยจะเป็นการกำหนดชั่วโมง และนาที



รูปที่ 4.13 ตั้งเวลาเปิด-ปิดระบบ



รูปที่ 4.14 ตัวอย่างการตั้งเวลาเปิด-ปิดระบบ

4.13 แก้ไขช่วงเวลาเปิด-ปิดระบบ

พนักงานสามารถแก้ไขช่วงเวลาเปิด-ปิดระบบ โดยเมื่อคลิกไปในช่วงระยะเวลาที่แก้ไขจะได้หน้าต่างแสดงดังรูปที่ 4.15 โดยจะมีแสดงช่วงเวลาที่ยังก่อนได้รับการแก้ไขด้วย

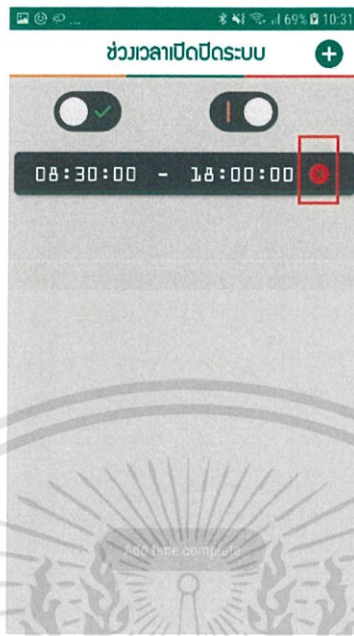


รูปที่ 4.15 แก้ไขช่วงเวลาเปิด-ปิดระบบ

4.14 ลบช่วงเวลาเปิด-ปิดระบบ

พนักงานสามารถลบช่วงเวลาเปิด-ปิดระบบโดยกดที่รูปภากระบาทสี่แดงได้ ดังรูปที่ 4.16 และผลลัพธ์จะแสดงดังรูปที่ 4.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



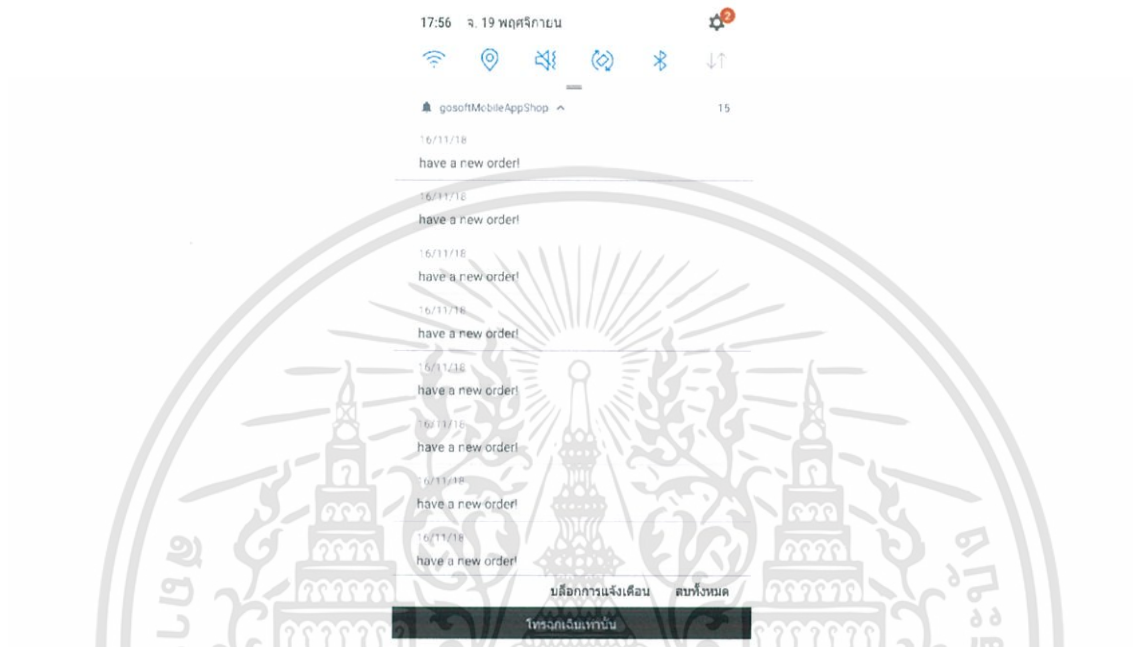
รูปที่ 4.16 ลบช่วงเวลาเปิด-ปิดระบบ



รูปที่ 4.17 ผลลัพธ์หลังจากลบช่วงเวลาเปิด-ปิดระบบ

4.15 ได้รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่

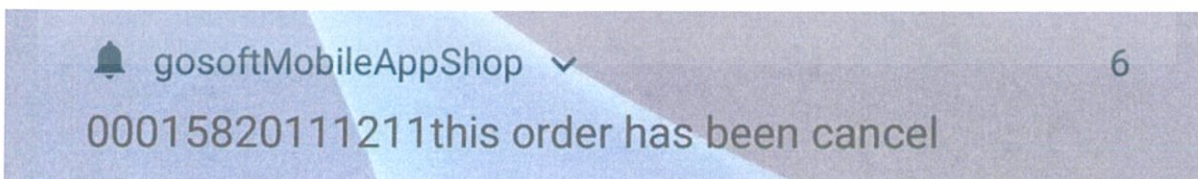
พนักงานสามารถดูการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่ได้ โดยจะแสดงดังรูปที่ 4.18 โดยจะมีข้อความส่งมาบอกว่า have a new order !



รูปที่ 4.18 ได้รับการแจ้งเตือนเมื่อมีรายการสินค้าเข้ามาใหม่

4.16 ได้รับการแจ้งเตือนเมื่อผู้ใช้งานยกเลิกรายการสินค้า

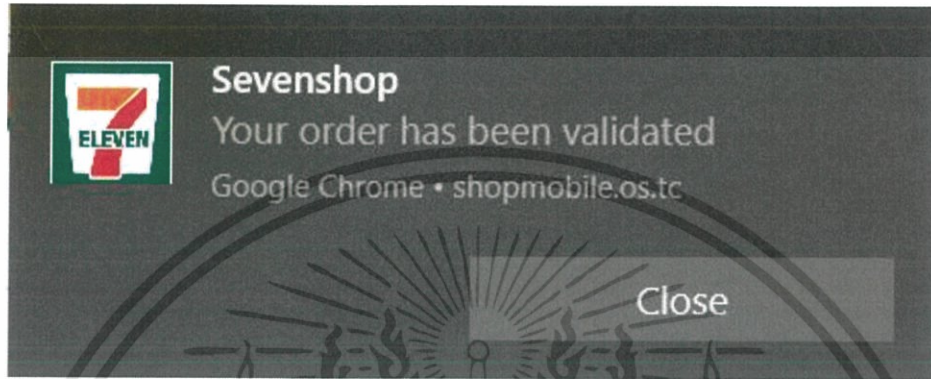
พนักงานสามารถดูการแจ้งเตือนเมื่อผู้สั่งยกเลิกรายการสินค้าได้ โดยแสดงดังรูปที่ 4.19 เมื่อลูกค้าได้ทำการยกเลิกรายการสินค้า ร้านที่รับผิดชอบรายการสินค้านั้นจะได้รับการแจ้งเตือนโดยข้อความที่ส่งนั้นประกอบไปด้วยหมายเลขรายการสินค้าที่ยกเลิก และข้อความว่า this order has been cancel



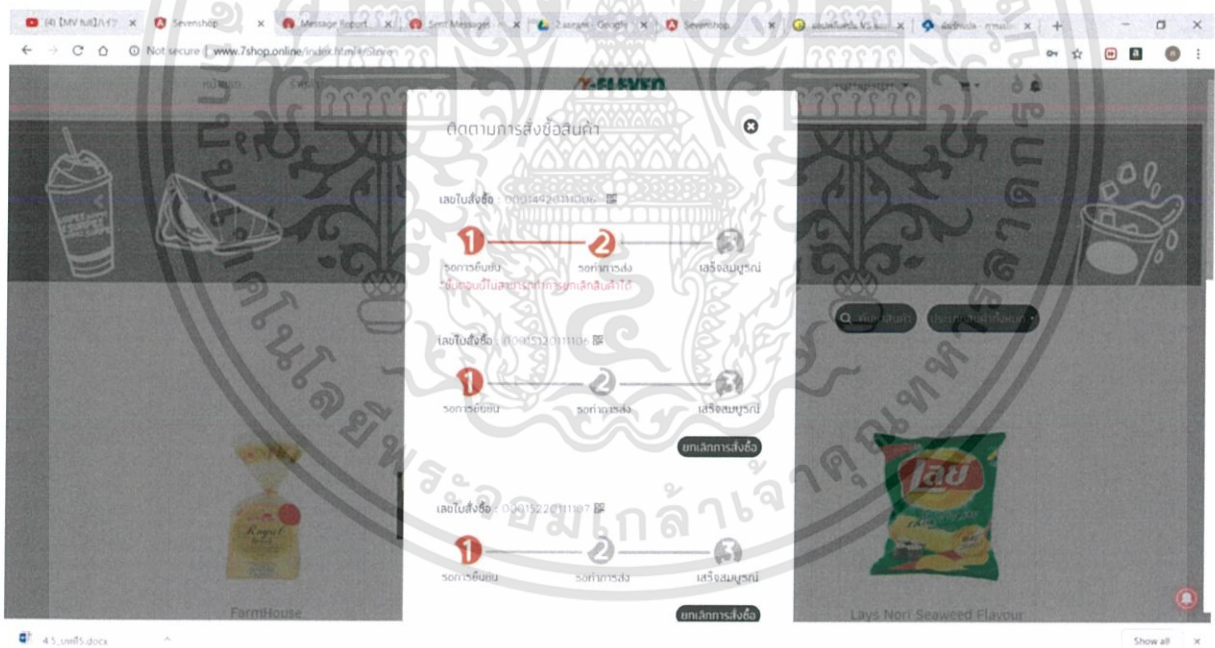
รูปที่ 4.19 ได้รับการแจ้งเตือนเมื่อผู้ใช้งานยกเลิกรายการสินค้า

4.17 ส่งการแจ้งเตือนกลับไปหาผู้ใช้งานเพื่อบอกความคืบหน้าของรายการสินค้า

และระบบจะทำการส่งการแจ้งเตือนกลับไปหาผู้สั่งเพื่อบอกความคืบหน้าของรายการสินค้าได้ ดังรูปที่ 4.20 และไปแสดงผลบนเว็บแอปพลิเคชัน ดังรูปที่ 4.21 โดยในรูปนี้เป็นตัวอย่างของเว็บแอปพลิเคชันที่ได้นำการแจ้งเตือนที่รับเข้ามา มาแสดงผลให้ฝั่งลูกค้าได้ดูความคืบหน้าของรายการสินค้าได้



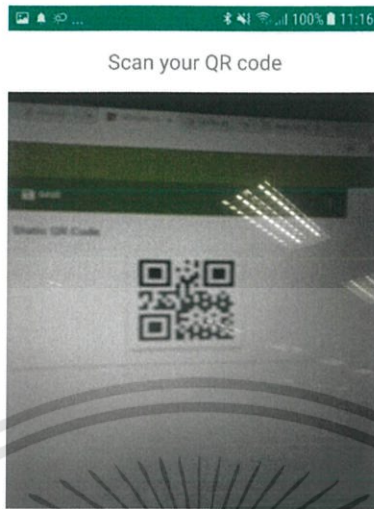
รูปที่ 4.20 ส่งการแจ้งเตือนกลับไปหาผู้ใช้งานเพื่อบอกความคืบหน้าของรายการสินค้า



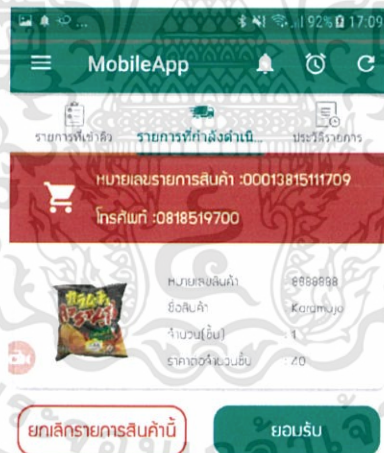
รูปที่ 4.21 ความคืบหน้าของรายการสินค้า

4.18 ดูสินค้าในรายการผ่านการสแกน QR code

พนักงานสามารถดูสินค้าในรายการสินค้าผ่านทาง QR code ได้โดย แอปพลิเคชันจะมีระบบการสแกน ดังรูปที่ 4.22 เมื่อสแกนเสร็จสิ้นจะแสดงสินค้าในรายการนั้น ดังรูปที่ 4.23 หากสแกนแล้วนำค่าที่ได้ไปค้นหาแล้วไม่เจอรายการสินค้าก็จะขึ้นข้อความว่าค้นหาไม่เจอรายการสินค้านั้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 การสแกน QR code



รูปที่ 4.23 ดูสินค้าในรายการผ่านการสแกน QR code

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

แอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้าสามารถรับรายการสินค้าจากทางด้านเว็บ แอปพลิเคชันและนำรายการสินค้ามาแสดงบนหน้าจอโทรศัพท์แพลตฟอร์มแอนดรอยด์ได้ เพื่ออำนวยความสะดวกให้แก่พนักงานที่ร้านสะดวกซื้อในการรับรายการสินค้า และสามารถหลีกเลี่ยงปัญหาด้านการจัดเรียงคิวรายการสินค้าที่เกิดขึ้นได้ รวมถึงมีการส่งความคืบหน้ากลับไปหาผู้สั่งได้ เพื่อให้ผู้สั่งสามารถติดตามรายการสินค้าของตนได้

5.2 ปัญหาที่พบระหว่างการทดลอง

- 1) ขนาดหน้าจอของโทรศัพท์มือถือที่นำมาใช้ไม่ตรงกับที่ตั้งค่าไว้ในตอนแรก
- 2) รูปแบบโมเดลที่นำมาใช้ในการรับข้อมูลไม่ตรงกับ response ที่ส่งมาจากทางด้านฐานข้อมูล
- 3) โครงสร้างฐานข้อมูลที่นำมาใช้ในการทดสอบไม่ตรงกับในระบบจริง
- 4) ผู้จัดทำไม่มีโทรศัพท์มือถือแพลตฟอร์มแอนดรอยด์ จึงต้องใช้ simulation ในการทำซึ่งเกิดปัญหาไม่สามารถทดสอบระบบการแจ้งเตือน และระบบสแกน QR code ได้

5.3 แนวทางการแก้ไขปัญหา

- 1) ปรับแก้ไขให้เหมาะสมกับเครื่องที่จะนำไปใช้จริง
- 2) ตรวจสอบผลลัพธ์ที่ได้จากการ response และปรับโมเดลให้ตรงกับที่ได้รับมา
- 3) แก้ไขฐานข้อมูลให้เหมือนกับระบบจริง
- 4) ขอยืมโทรศัพท์มือถือแอนดรอยด์ที่พี่ใช้ในการทดสอบ

5.4 แนวทางการพัฒนาต่อ

- 1) ระบบในการ Log in สามารถดึงข้อมูลมาจากรฐานข้อมูลของทางบริษัทได้
- 2) พัฒนาระบบการสแกน QR code ให้ภาพชัดมากยิ่งขึ้น
- 3) ให้สแกน QR code สามารถแสดงผลข้อมูลในการค้นหาได้ละเอียดยิ่งขึ้น
- 4) สามารถรองรับได้ในโทรศัพท์มือถือทุกรุ่น ทุกขนาดได้

เอกสารอ้างอิง

- [1] พร้อมเลิศ หล่อวิจิตร. คู่มือเขียนแอป Android ด้วย Android Studio. พิมพ์ครั้งที่ 3.
- [2] Samisa Abeysinghe, RESTful PHP Web Services, 2008
- [3] What is OneSignal เข้าถึงได้จาก : <https://documentation.onesignal.com/docs>
- [4] What is API เข้าถึงได้จาก : <https://medium.freecodecamp.org/what-is-an-api-in-english-please-b880a3214a82>
- [5] What is Node.js เข้าถึงได้จาก : https://www.w3schools.com/nodejs/nodejs_intro.asp
- [6] What is XML เข้าถึงได้จาก : https://www.w3schools.com/xml/xml_what.asp
- [7] What is SQL เข้าถึงได้จาก : https://www.w3schools.com/sql/sql_intro.asp





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรมที่นำมาใช้ในการปฏิบัติงานในครั้งนี้นั้นได้แก่

- Android Studio
- Visual Studio Code
- Genymotion
- Postman
- Xampp





ภาคผนวก ก

การทดสอบ RESTful หรือการเรียกใช้งาน web service ผ่าน API

ซึ่งจากเดิมก่อนที่มาใช้รูปแบบการเรียกใช้ web service นั้นจะเป็นการ query ข้อมูลผ่าน database โดยตรง โดยมีการใช้ภาษา SQL ซึ่งไม่มีตัวกลางในการเชื่อมต่อระหว่าง client กับ server ซึ่งทำให้ผู้พัฒนาส่วนมากจะต้องพัฒนาฝั่ง client ควบคู่ กับฝั่ง server ซึ่งหากฝั่ง client มีการเปลี่ยนแปลง ฝั่ง server ก็มีการเปลี่ยนแปลงเช่นกันซึ่งเป็นปัญหาในการพัฒนาของผู้พัฒนา โดยแต่เดิมนั้นได้ใช้ภาษา PHP เป็นภาษาที่ใช้ในการเขียนฝั่ง server จะมีการเชื่อมต่อ database ต่างๆ ดังรูปที่ ก.1

```

<?php
$objConnect = mysql_connect("localhost","root","root");
$objDB = mysql_select_db("mydatabase");

// $_POST["txtKeyword"] = "a"; // for Sample

$strKeyword = $_POST["txtKeyword"];
$strSQL = "SELECT * FROM customer WHERE 1 AND Name LIKE '%" . $strKeyword . "%' ";

$objQuery = mysql_query($strSQL);
$numFields = mysql_num_fields($objQuery);
$resultArray = array();
while($objResult = mysql_fetch_array($objQuery))
{
    $arrCol = array();
    for($i=0;$i<$numFields;$i++)
    {
        $arrCol[mysql_field_name($objQuery,$i)] = $objResult[$i];
    }
    array_push($resultArray,$arrCol);
}
mysql_close($objConnect);

echo json_encode($resultArray);
?>

```

รูปที่ ก.1 ตัวอย่างการเชื่อมต่อ database ฝั่ง server โดยใช้ภาษา PHP

ซึ่งการเชื่อมต่อฐานข้อมูลในปฏิบัติงานในครั้งนี้นั้นใช้รูปแบบของ RESTful การทำงานของ RESTful Web Service จะอาศัย URI/URL ของ Request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูป XML, HTML, JSON โดย Response ที่ตอบกลับจะเป็นการยืนยันผลของคำสั่งที่ส่งมา และสามารถพัฒนาด้วยภาษา programming ได้หลากหลาย คำสั่งก็จะมีตาม HTTP verbs ซึ่งก็คือ GET, POST, PUT และ DELETE

ในปฏิบัติงานในครั้งนี้ได้เลือกใช้ retrofit2 เป็นรูปแบบการเขียน RESTful รูปแบบหนึ่ง โดยก่อนที่จะใช้นั้นจำเป็นต้อง implement retrofit เข้ามายังแอปพลิเคชันก่อน ดังรูปที่ ก.2 โดยในการจะใช้ web service ดังกล่าวจำเป็นที่จะต้องคำนึงถึงข้อมูลที่ต้องการนำมาใช้เสียก่อนว่าจะต้องใช้ข้อมูลอะไรบ้าง ของตารางไหน แล้วต้องทำการเตรียม model สร้าง object เพื่อมารองรับค่าเหล่านั้นเสียก่อน ดังรูปที่ ก.3

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support:design:27.1.1'
    implementation 'com.android.support.constraint:constraint-layout:1.1.2'
    implementation 'com.android.support:support-annotations:27.1.1'
    testImplementation 'junit:junit:4.12'
    compile 'com.android.support:cardview-v7:27.1.1'
    compile 'com.android.support:recyclerview-v7:27.1.1'
    androidTestImplementation 'com.android.support.test:runner:1.0.2'
    androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
    compile 'com.squareup.retrofit2:retrofit:2.1.0'
    compile 'com.squareup.retrofit2:converter-gson:2.1.0'
    compile 'me.dm7.barcodescanner:zxing:1.8.4'
    compile 'com.google.android.gms:play-services-vision:15.0.0'
    //glide
    implementation 'com.github.bumptech.glide:glide:4.7.1'
    // //circleimageview
    // implementation 'de.hdodenhof:circleimageview:2.2.0'
    implementation 'com.onesignal:OneSignal:[3.9.1, 3.99.99]'
    // implementation 'com.google.firebase:firebase-core:16.0.0'
    // implementation 'com.google.android.gms:play-services:12.0.1'
}
```

รูปที่ ก.2 การ implement retrofit2 เข้ามายังแอปพลิเคชัน

```
public class Time {
    @SerializedName("time_id")
    private int time_id;
    @SerializedName("open_time")
    private String open_time;
    @SerializedName("close_time")
    private String close_time;

    public Time(int time_id, String open_time, String close_time) {
        this.open_time = open_time;
        this.close_time = close_time;
    }

    public int getTime_id() { return time_id; }

    public void setTime_id(int time_id) { this.time_id = time_id; }

    public String getOpen_time() { return open_time; }

    public void setOpen_time(String open_time) { this.open_time = open_time; }

    public String getClose_time() { return close_time; }

    public void setClose_time(String close_time) { this.close_time = close_time; }
}
```

รูปที่ ก.3 ตัวอย่างการสร้าง model เพื่อรับข้อมูลที่ดึงมาจากฐานข้อมูล

เมื่อได้เตรียม model เพื่อรอรับข้อมูลเสร็จแล้ว ต่อมาจะไปดูฝั่ง server โดยฝั่ง server นั้นได้ใช้ ภาษา JavaScript ในการเขียน โดยใช้โปรแกรม Visual Studio Code ซึ่งใช้ Node.JS เป็นหลัก โดยใน ปฏิบัติงานในครั้งนี้ได้ออกแบบฝั่ง server เป็น 2 ฝั่งนั้นคือส่วนที่ใช้ในการ query คำสั่ง SQL เพื่อเชื่อมต่อกับ ฐานข้อมูล ดังรูปที่ ก.4 และส่วนที่ดูแลในเรื่องของ URL ซึ่งเป็นส่วนที่กำหนด HTTP verb เช่น get, post, put และ delete และยังกำหนดลักษณะของ URL ว่ามีการส่งค่าผ่านทาง URL มาหรือไม่ แล้วดูแลในเรื่อง ของประเภทการส่งค่ากลับว่าเป็นลักษณะไหน ดังรูปที่ ก.5 โดยในปฏิบัติงานนี้ได้เลือกประเภทการส่งค่าไป- กลับเป็นรูปแบบ JSON ดังรูปที่ ก.6

```

1 var db = require("../Database/dbconnection")
2 var bcrypt = require('bcrypt');
3 var Search = {
4   getAllTasks:function(callback){
5     return db.query("select * from product inner join product type on product type id=product.product_type_id inner join stock on stock.product_serial_nu
6   },
7   getTaskById:function(id,callback){
8     return db.query("select * from product inner join product type on product type id=product.product_type_id inner join stock on stock.product_serial_nu
9   },
10  getTaskByShopId:function(shop,id,callback){
11    return db.query("select * from product inner join product type on product type id=product.product_type_id inner join stock on stock.product_serial_nu
12  },
13  getProduct:function(callback){
14    return db.query("select * from product order by product name",callback);
15  },
16  getProductById:function(id,callback){
17    return db.query("select * from product where product serial number = ?",[id],callback);
18  },
19  getProductType:function(callback){
20    return db.query("select * from product type",callback);
21  },
22  getAllProductType:function(callback){
23    return db.query("select * from product type DESC",callback);
24  },
25  getLatestProductId:function(callback){
26    return db.query("select product serial number from product order by product serial number DESC ",callback)
27  },
28  getStock:function(callback){
29    return db.query("select * from stock",callback)
30  },

```

รูปที่ ก.4 ตัวอย่างส่วนที่ใช้ในการ query เพื่อดึงข้อมูลจากฐานข้อมูล

```

    .get('/getTime/:shop_id', function (req, res, next) {
      res.header("Access-Control-Allow-Origin", "*");
      res.header("Access-Control-Allow-Headers", "Origin, X-Requested-With, Content-Type, Accept");
      con.getConnection(function (err, connection) {
        Search.getTime(req.params.shop_id,function (err, rows) {
          connection.release();
          if (err) {
            res.json(err);
            console.log(err);
          }
          else {
            res.json(rows);
            console.log(rows);
          }
        })
      })
    })
  )
)
function test(req, tmp, j, callback) {

```

รูปที่ ก.5 ตัวอย่างส่วนที่กำหนด HTTP verb และลักษณะของ URL



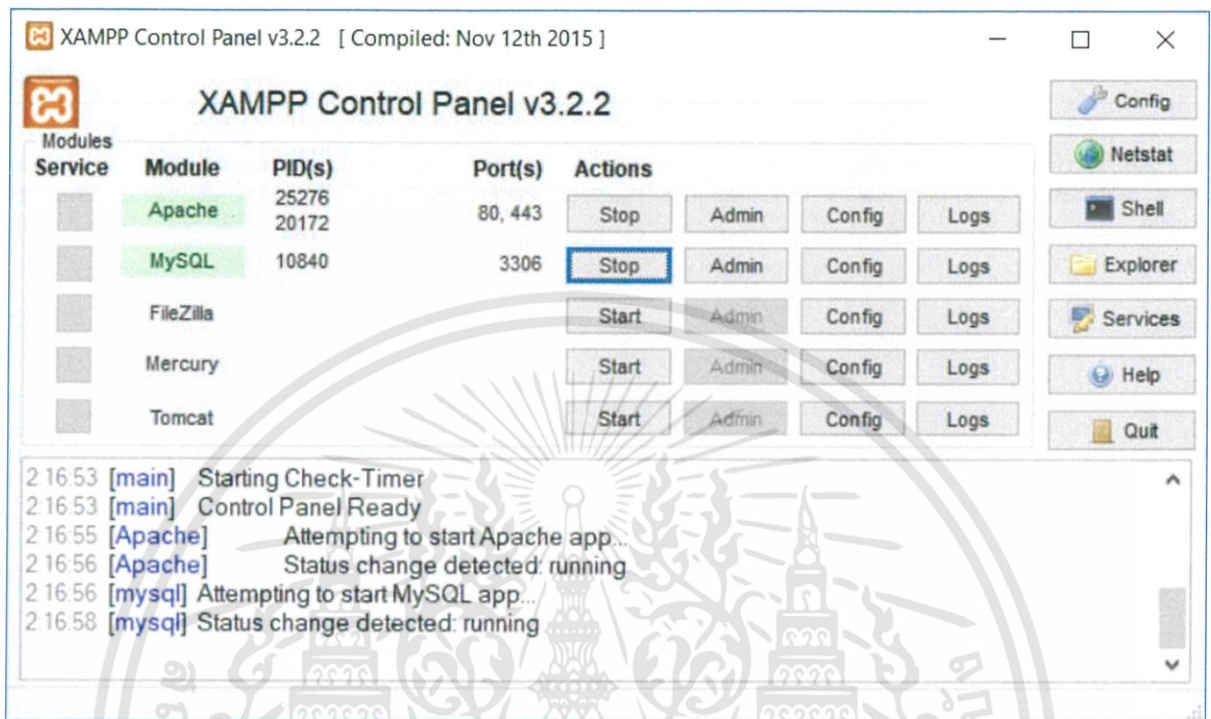
รูปที่ ก.6 ตัวอย่างชุดข้อมูล JSON

เมื่อได้เตรียม ฟัง server เสร็จสิ้นแล้วนั้นจะทำการทดสอบการเรียกใช้ web service โดยเรียกผ่าน URL ซึ่งผู้จัดทำนั้นได้ใช้ โปรแกรม Postman ในการทดสอบ และทดสอบ server ใน localhost ซึ่งสิ่งที่ต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

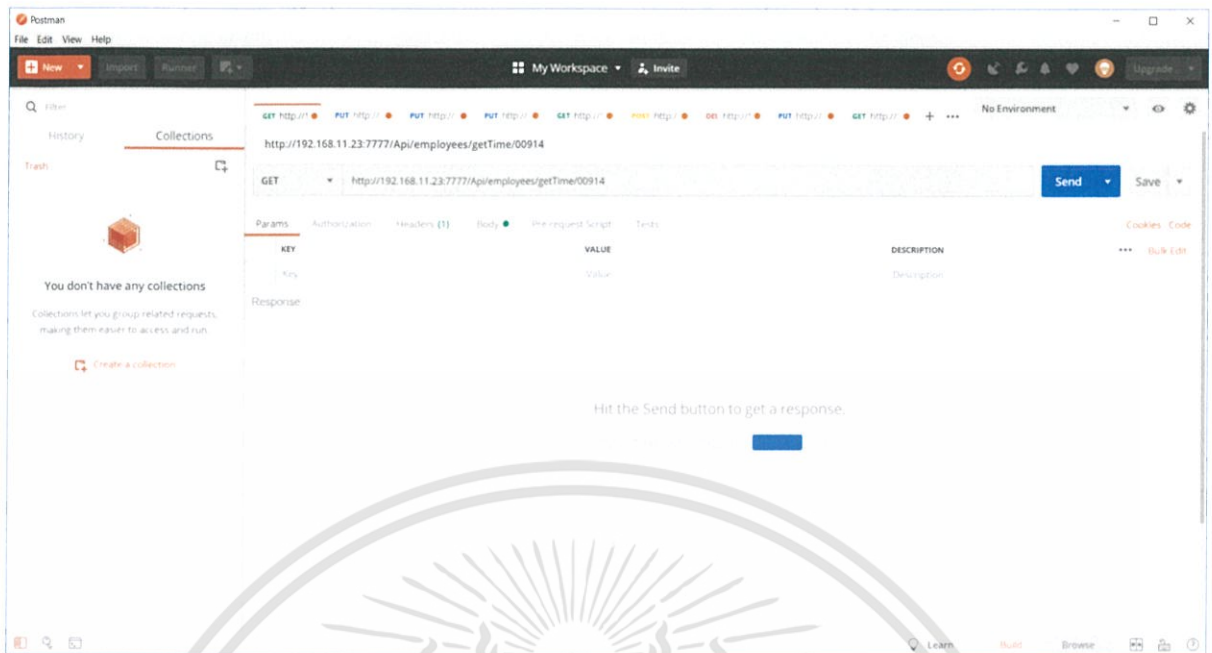
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำก่อนที่จะทดสอบก็คือทำการเปิด server localhost ซึ่งทางผู้จัดทำนั้นได้เลือก Xampp เป็นตัวจัดการในเรื่องของ localhost โดนต้องทำการเปิด server localhost ดังรูปที่ ก.7 จึงจะสามารถทดสอบได้



รูปที่ ก.7 ตัวอย่างการเปิด server เพื่อใช้งาน localhost บน Xampp

เมื่อทำการเปิดแล้ว ก็เปิดโปรแกรม Postman เพื่อทำการทดสอบ การเรียกใช้งาน web service ผ่าน URL โดยตัวอย่างหน้าต่างโปรแกรม Postman นั้นเป็นดังรูปที่ ก.8 โดยจะมีส่วนที่กำหนด HTTP verb ว่า จะใช้ method ไหน และส่วนที่เป็น URL เพื่อระบุ Path ที่ต้องการเรียกใช้งาน service โดนในส่วนของ Postman นั้นมันเป็นการทดสอบการเชื่อมต่อฐานข้อมูลฝั่ง server ว่าสามารถเพิ่ม แก้ไข เรียกดู หรือลบ ข้อมูลในฐานข้อมูลได้หรือไม่



รูปที่ ก.8 ตัวอย่างโปรแกรม Postman

ซึ่งเมื่อทำใส่ URL และค่าที่ต้องการจะส่งเข้าไปในการ query นั้นแล้วจากในรูปที่ ก.8 นั้นได้ยกตัวอย่างการเรียกใช้งาน web service เป็นการดึงช่วงระยะเวลาเปิด-ปิดร้านของร้านที่มีรหัส 00914 และผลลัพธ์ที่ได้นั้นจะถูกจัดให้อยู่ในรูปแบบของ JSON ซึ่ง response ของ service ตัวนี้นั้นมีผลลัพธ์ ดังรูปที่ ก.9

```
[
  {
    "time_id": 28,
    "open_time": "20:00:00",
    "close_time": "23:00:00"
  },
  {
    "time_id": 31,
    "open_time": "02:00:00",
    "close_time": "01:00:00"
  },
  {
    "time_id": 32,
    "open_time": "04:00:00",
    "close_time": "08:00:00"
  },
  {
    "time_id": 33,
    "open_time": "12:00:00",
    "close_time": "14:00:00"
  }
]
```

รูปที่ ก.9 ตัวอย่างผลลัพธ์ของ service ที่ต้องการดึงช่วงเวลาเปิด-ปิดร้านของร้านรหัส 00914

ซึ่งปัจจัยหลักของ RESTful นั้นก็คือการทำแยกส่วนกันระหว่างฝั่ง client และ ฝั่ง server โดย ฝั่ง server นั้นจะกำหนดว่าจะต้องมี URL หน้าตาแบบนี้มีค่าที่ส่งเข้ามาเป็นประเภทนี้ แล้วผลลัพธ์ที่ได้จะมีหน้าตาแบบนี้กลับไป เพราะฉะนั้นฝั่ง client จึงต้องเตรียม model หรือ object เพื่อรับผลลัพธ์ตรงนั้นให้ถูกต้อง

ซึ่งเมื่อทำการทดสอบฝั่ง server เป็นที่เรียบร้อยแล้วนั้นจากนั้นจะเป็นส่วนที่ทดสอบฝั่ง client ว่าสามารถดึงข้อมูลมาแสดงบนแอปพลิเคชันได้หรือไม่ โดยก่อนที่จะทดสอบนั้นต้องทำการเชื่อมต่อให้อยู่ในวง lan เดียวกันกับ server ก่อน จากนั้นทำการตั้งค่า base URL ที่ใช้ในแอปพลิเคชันดังรูปที่ ก.10

```
package com.example.a30010420.g0901mobileappshop.network;

import ...

/**
 * Created by Matt on 29/8/2561.
 */
public class RetrofitInstance {
    private static Retrofit retrofit;
    private static final String BASE_URL = "http://172.20.10.2:7777";
    // private static final String BASE_URL = "http://www.7shop.online:7777";

    public static Retrofit getRetrofitInstance(){
        if (retrofit == null) {
            retrofit = new retrofit2.Retrofit.Builder()
                .baseUrl(BASE_URL)
                .addConverterFactory(GsonConverterFactory.create())
                .build();
        }
        return retrofit;
    }
}
```

รูปที่ ก.10 ตัวอย่างการตั้งค่า base URL ในแอปพลิเคชัน

ซึ่งเมื่อทำการตั้งค่า base URL เป็นที่เรียบร้อยแล้วนั้น ในปฏิบัติงานนี้ได้ทำการออกแบบวิธีการเขียนเรียกใช้ web service โดยเป็นการสร้าง interface เพื่อให้ ในส่วนของ activity นั้นได้เรียกนำไปใช้ โดยตัวอย่างของ interface มีลักษณะดังรูปที่ ก.11

```

public interface GetService {
    @GET("/Api/employees/getQueueOrder2/{shop_id}")
    Call<ArrayList<Order>> getOrder(@Path("shop_id") String shop_id);

    @GET("/Api/employees/getProductInQueue/{id}")
    Call<ArrayList<ProductInQueue>> getProductInQueue(@Path("id") int id);

    @PUT("/Api/employees/updatePendingOrderOpStatus/{id}")
    Call<UpdatePendingResponse> UpdatePendingResponse(@Path("id") int id);

    @PUT("/Api/employees/updateCancelOrderStatus/{id}")
    Call<UpdateCancelOrderOpStatusResponse> UpdateCancelOrderStatusResponse (@Path("id") int id);

    @PUT("/Api/employees/updateSuccessOrderStatus/{id}")
    Call<UpdateSuccessOrderOpStatusResponse> UpdateSuccessOrderStatusResponse (@Path("id") int id);

    @GET("/Api/employees/getPendingOrder/{shop_id}")
    Call<ArrayList<Order>> getPendingOrder(@Path("shop_id") String shop_id);

    @PUT("/Api/employees/updateProductInQueue/{shop_id}")
    Call<UpdateProductInQueueResponse> UpdateProductInQueue(@Path("shop_id") String shop_id, @Body ArrayList<ProductInQueue> uncheckeddataList);

    @PUT("/Api/employees/updateProductInQueue/{shop_id}")
    Call<UpdateProductInPendingResponse> UpdateProductInPending(@Path("shop_id") String shop_id, @Body ArrayList<ProductInPending> dataList);

    @PUT("/Api/employees/updateProductStatus/{order_id}")
    Call<UpdateProductStatusResponse> UpdateProductStatus(@Path("order_id") int order_id, @Body ArrayList<ProductInQueue> checkeddataList);

    @GET("/Api/employees/getProductInPending/{id}/{number}")
    Call<ArrayList<ProductInPending>> getProductInPending(@Path("id") int id, @Path("number") String number);
}

```

รูปที่ ก.11 ตัวอย่างของ interface ของการเรียกใช้ web service

ใน interface ส่วนนี้นั้นจะประกอบไปด้วย ประเภท method ที่ใช้ในการเรียกใช้ web service โดยมีการกำหนด URL ที่ต้องตรงกับ URL ฝั่ง server เพื่อเป็นการเรียกใช้งาน web service นั้นๆ โดยมีการระบุ path ค่าที่ส่งไปพร้อมกับ URL และค่าที่ส่งใน body รวมถึงเมื่อได้รับผลลัพธ์หรือ response แล้วจะนำข้อมูลนั้นไปใส่ใน model ไหน

และในส่วนของแอปพลิเคชันนั้นในการจะเรียกใช้ web service จะมีรูปแบบการเขียนโดยใช้ retrofit2 ซึ่งมีลักษณะดังรูปที่ ก.12 โดยจากรูปดังกล่าวในการเรียกใช้งาน web service นั้นจะถูกแบ่งเป็น 2 ส่วนคือ onResponse และ onFailure ซึ่งถ้าหากทางฝั่ง client มีการกำหนด URL กำหนดประเภทค่าที่ส่งไปตามที่ฝั่ง server กำหนดถูกต้องแล้วฝั่ง server จะทำการส่งผลลัพธ์ หรือ response กลับมาหา client จากนั้น client จะมีการนำผลลัพธ์นั้นไปยัง model ที่กำหนดถูกกำหนด แล้วจากนั้นการทำงานจะเข้าไปสู่ส่วน onResponse ซึ่งเราจะนำข้อมูลนั้นไปทำอะไรต่อก็ใส่ในส่วนนี้ แต่ถ้าหากเกิดข้อผิดพลาดต่างๆ เกิดขึ้น อย่างเช่น การส่งค่าของฝั่ง client ไม่ตรงกับฝั่งของ server หรือ model ที่รองรับของทางฝั่ง client นั้นไม่ตรงกับผลลัพธ์ที่ส่งมาจากทางฝั่ง server หรือ การเชื่อมต่ออินเทอร์เน็ตนั้นมีปัญหา ก็จะเข้าสู่การทำงานในส่วน onFailure ซึ่งเราก็สามารถแจ้งเตือนกลับไปหา ผู้ใช้งานได้ว่าเกิดข้อผิดพลาดจากการดึงข้อมูลเนื่องจากปัญหาอะไรได้

```

GetService service = RetrofitInstance.getRetrofitInstance().create(GetService.class);

Call<AddTimeResponse> call = service.insertTime(shop_id,timeData);

Log.wtf( tag: "URL Called",  msg: call.request().url()+"");

call.enqueue(new Callback<AddTimeResponse>() {
    @Override
    public void onResponse(Call<AddTimeResponse> call, Response<AddTimeResponse> response){
        Toast.makeText( context: AddTimeActivity.this, text: "Add time complete", Toast.LENGTH_SHORT).show();
        System.out.println("Add time complete");
        queue.setText("กำลังรวมรายการสินค้า");
        loadingDialog.dismiss();
        finish();
    }
    @Override
    public void onFailure(Call<AddTimeResponse> call,Throwable t){
        Toast.makeText( context: AddTimeActivity.this, text: "Add time failed",Toast.LENGTH_SHORT).show();
        System.out.println(t);
        loadingDialog.dismiss();
        Toast.makeText( context: AddTimeActivity.this, text: "No internet connection",Toast.LENGTH_SHORT).show();
        AlertDialog.Builder mBuilder = new AlertDialog.Builder( context: AddTimeActivity.this);
        View mView =getLayoutInflater().inflate(R.layout.no_internet_connection_dialog, root: null);
        TextView tv_no_internet_connection = (TextView) mView.findViewById(R.id.tv_no_internet_connection);
        Button btn_no_internet_connection_back = (Button) mView.findViewById(R.id.btn_no_internet_connection_back);
        mBuilder.setView(mView);
        final AlertDialog onInternetDialog = mBuilder.create();
        onInternetDialog.show();
        btn_no_internet_connection_back.setOnClickListener((v) -> {
            onInternetDialog.dismiss();
        });
    }
});

```

รูปที่ ก.12 ตัวอย่างรูปแบบการเขียนโดยใช้ retrofit2

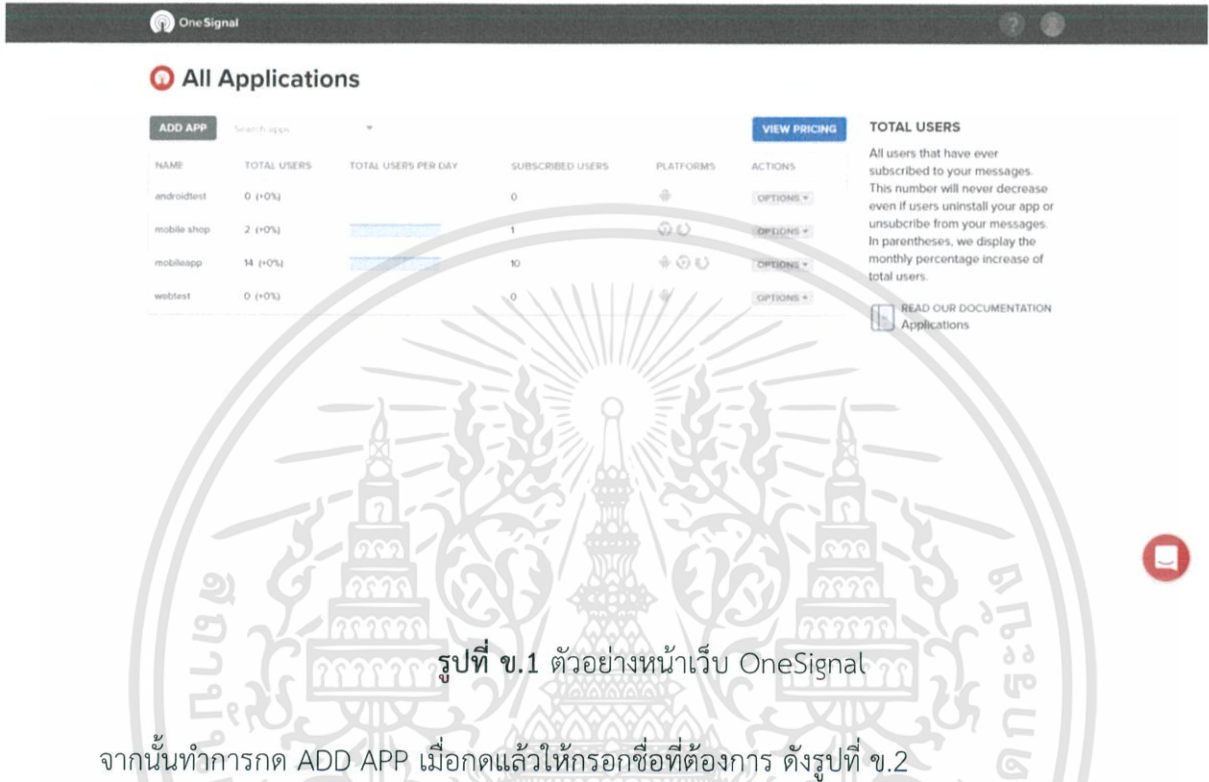
และเมื่อทุกอย่างไม่มีข้อผิดพลาดนั้นก็จะสามารถเรียกใช้งาน web service และนำข้อมูลจากในฐานข้อมูลนำมาแสดงบนแอปพลิเคชันได้ ยกตัวอย่างรูปที่ ก.13 เป็นการดึงข้อมูลรายการสินค้าขึ้นมาแสดงบนหน้าจอโดยมีรายละเอียดรายการสินค้าต่างๆ



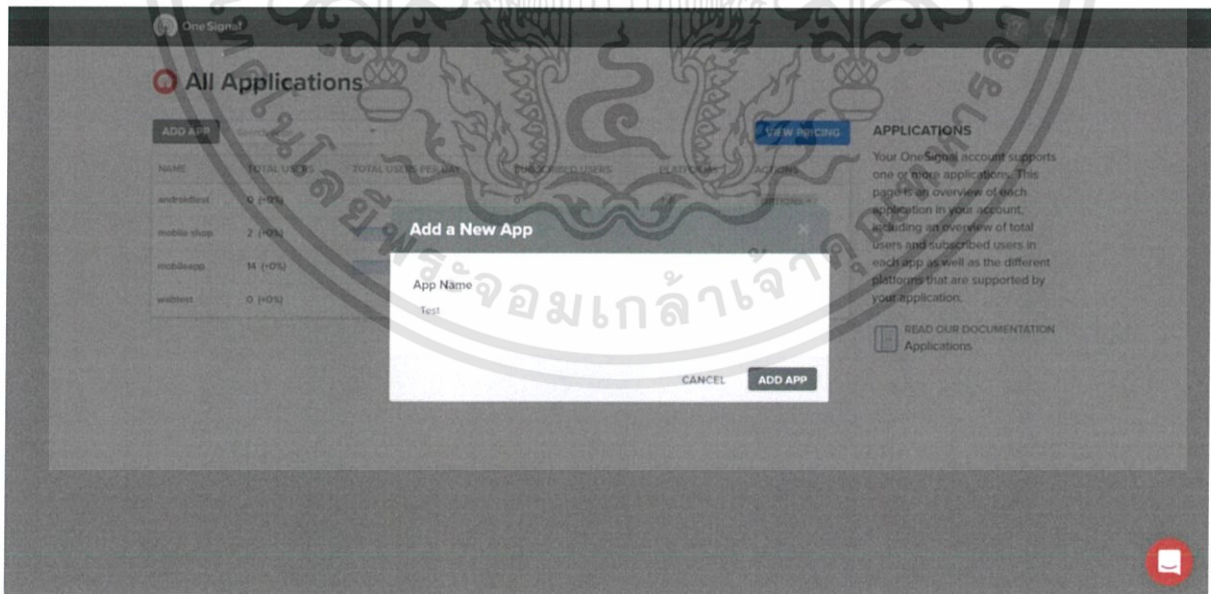
รูปที่ ก.13 ตัวอย่างผลลัพธ์ของการเรียกใช้ web service ในการแสดงรายละเอียดสินค้าในฝั่ง client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปฏิบัติงานนี้ได้เลือกใช้ OneSignal ในการรับ-ส่งการแจ้งเตือน ซึ่งก่อนที่จะสามารถรับหรือส่งการแจ้งเตือนได้นั้นจำเป็นที่จะต้องเข้าสู่ระบบของ OneSignal เสียก่อน เพื่อทำการลงทะเบียนแอปพลิเคชันเพื่อจะใช้งาน OneSignal โดยหน้าตาเว็บ OneSignal เป็นดังรูปที่ ข.1



จากนั้นทำการกด ADD APP เมื่อกดแล้วให้กรอกชื่อที่ต้องการ ดังรูปที่ ข.2

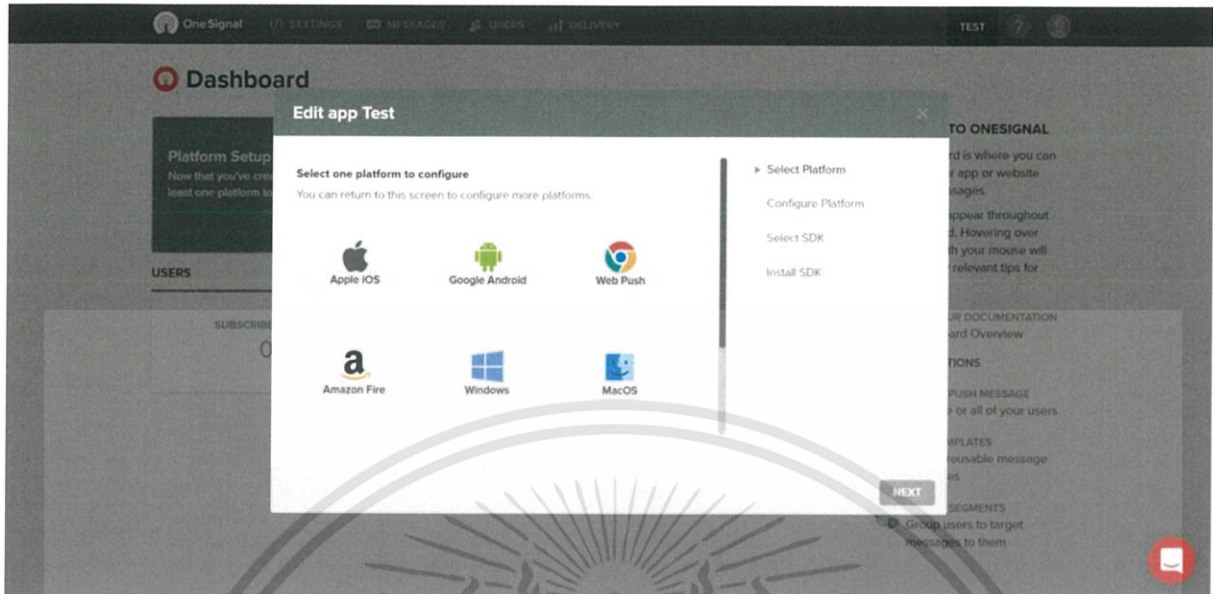


จากนั้นเมื่อทำการกด add เป็นที่เรียบร้อยแล้วนั้นจะได้หน้าตาดังรูปที่ ข.3 จะมีให้เลือกหน้าตา platform ที่จะใช้งาน OneSignal



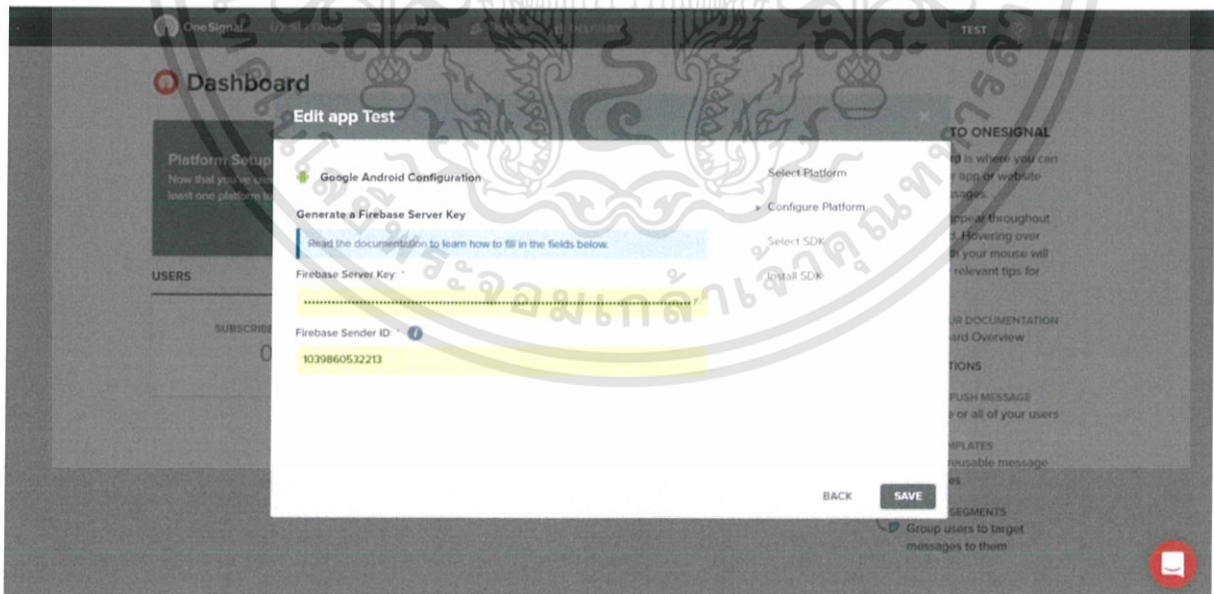
ภาคผนวก ข

การทดสอบการส่งการแจ้งเตือน



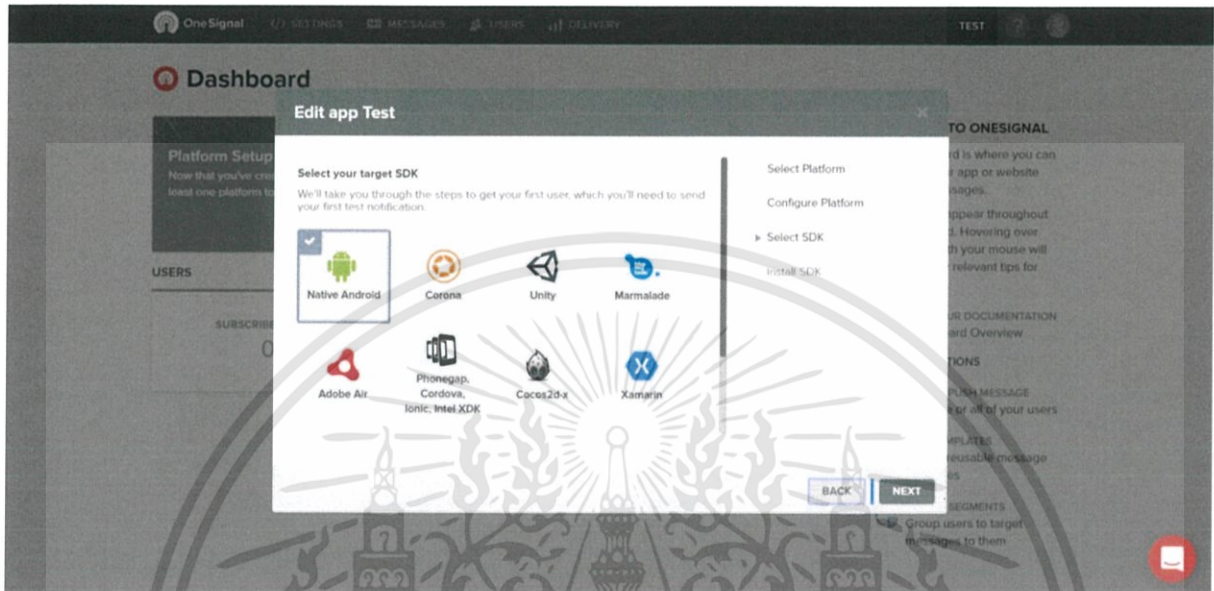
รูปที่ ข.3 ตัวอย่างหน้าเว็บ OneSignal ที่ให้เลือก platform การใช้งาน

ในปฏิบัติการนี้ก็ได้เลือก platform เป็นของ Google Android เมื่อทำการเลือก platform เป็นที่เรียบร้อยแล้วนั้น กด next จะพบหน้าต่างที่ให้กรอก Firebase server key และ firebase Sender ID ซึ่งจะได้ 2 อย่างนี้ก็ต่อเมื่อทำการสมัคร firebase ก่อนแล้วทาง firebase จะส่ง server key ผ่านทาง email มาให้กรอกนั่นเอง โดยหน้าต่างเว็บมีลักษณะดังรูปที่ ข.4



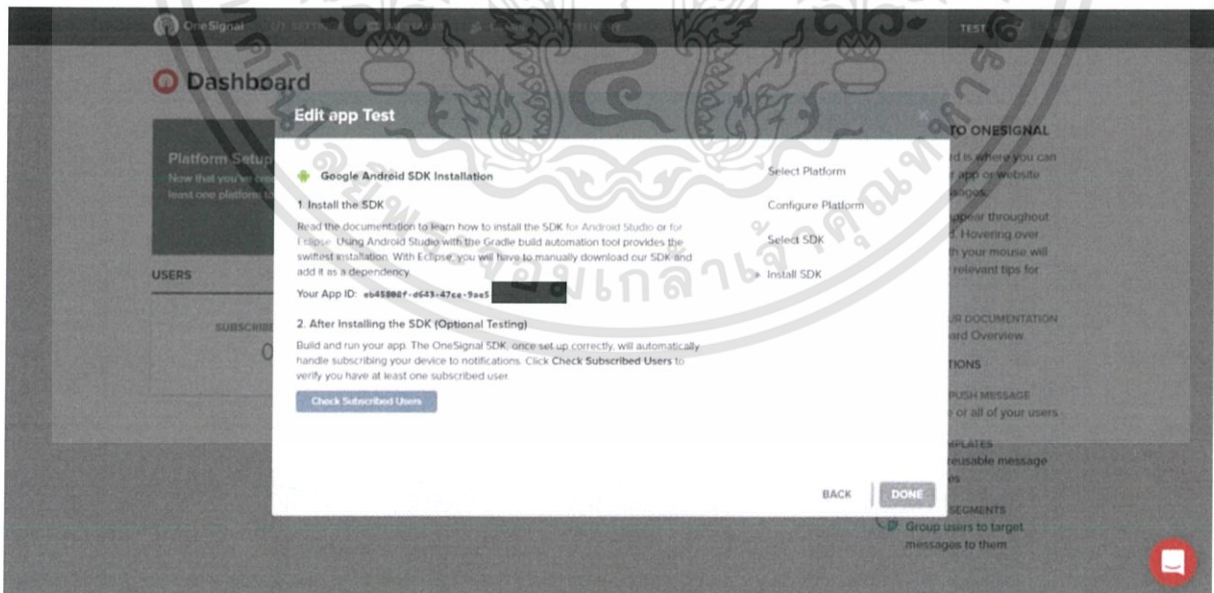
รูปที่ ข.4 ตัวอย่างหน้าเว็บ OneSignal ที่ให้กรอกข้อมูลรายละเอียดของ Firebase

เมื่อทำการกรอกเสร็จเป็นที่เรียบร้อยแล้วนั้น กด save จะพบกับหน้าต่างเว็บดังรูปที่ ข.5 โดยหน้านี้จะให้เลือก platform เป้าหมายที่ต้องการจะทดสอบการส่งการแจ้งเตือนว่าต้องการที่จะส่งการแจ้งเตือนนั้นไปยัง platform ไหนนั่นเอง



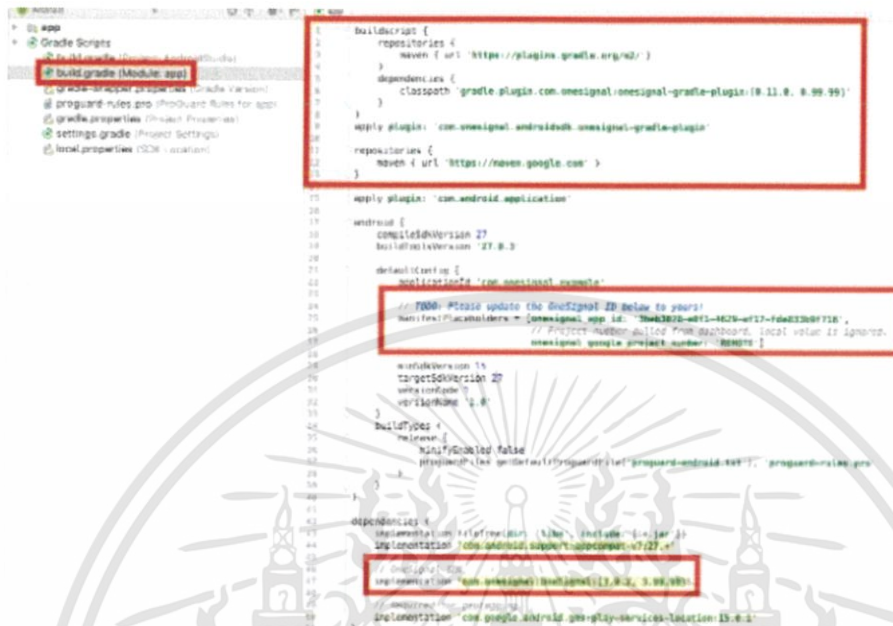
รูปที่ ข.5 ตัวอย่างหน้าเว็บ OneSignal เลือก platform เป้าหมายที่ต้องการจะทำการทดสอบ

เมื่อทำการเลือก platform เป้าหมายเรียบร้อยแล้วนั้นจะพบกับหน้าต่างเว็บ ดังรูปที่ ข.6 โดยจะมีการระบุ Your App ID มาให้เพื่อจะนำไปใช้ในขั้นตอนถัดไป



รูปที่ ข.6 ตัวอย่างหน้าเว็บ OneSignal ที่ให้ Your App ID เพื่อจะนำไปใช้ในขั้นตอนถัดไป

จากนั้นจะเป็นส่วนที่จะต้องนำมาเพิ่มในแอปพลิเคชันเป็นการติดตั้ง Android SDK โดยจะแทรกโค้ดเข้าไปใน gradle ของ android ดังรูปที่ ข.7



รูปที่ ข.7 เป็นการติดตั้ง Android SDK

โดยมีส่วนที่ต้องเพิ่มมีอยู่ 3 ส่วน ส่วนแรกนั้นจะต้องเพิ่มโค้ดดังรูปที่ ข.8

```
buildscript {
    repositories {
        maven { url 'https://plugins.gradle.org/m2/' }
    }
    dependencies {
        classpath 'gradle.plugin.com.onesignal:onesignal-gradle-plugin:[0.11.0., 0.99.99]'
    }
}
apply plugin: 'com.onesignal.androidsdk.onesignal-gradle-plugin'

repositories {
    maven { url 'https://maven.google.com' }
}
```

รูปที่ ข.8 โค้ดที่ต้องเพิ่มในส่วนที่ 1

โดยมีส่วนที่ 2 นั้นมีโค้ดดังรูปที่ ข.9

```
dependencies {  
    implementation 'com.onesignal:OneSignal:[3.9.1, 3.99.99]'  
}
```

รูปที่ ข.9 โค้ดที่ต้องเพิ่มในส่วนที่ 2

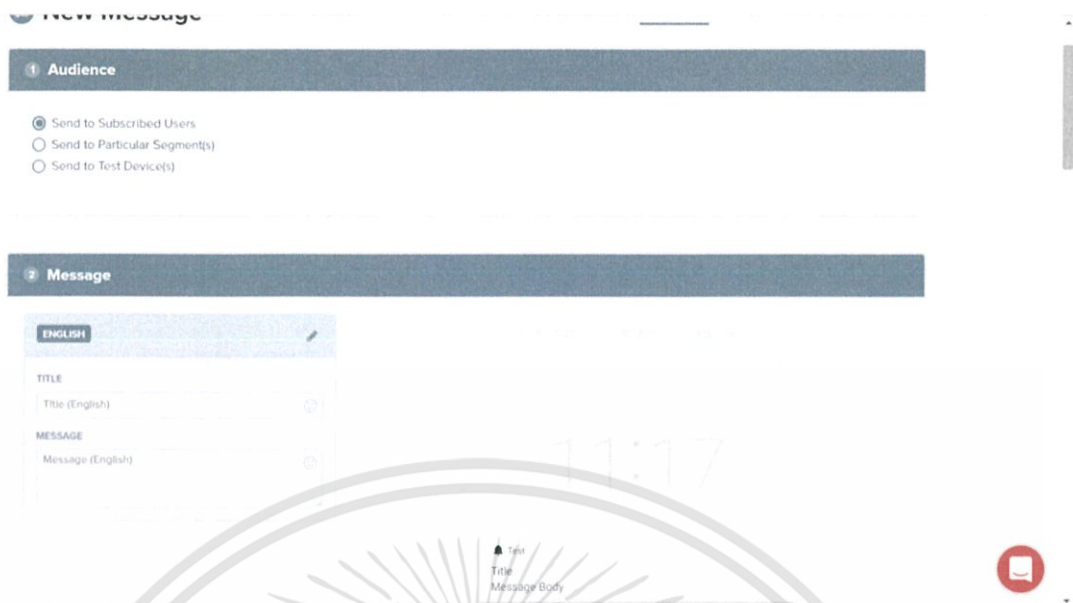
โดยมีส่วนที่ 3 นั้นมีโค้ดดังรูปที่ ข.10 โดยในส่วนที่จะต้องใส่ App ID ที่ได้มาจากขั้นตอนข้างต้นนั้นมาใส่ใน ‘ ‘

```
android {  
    defaultConfig {  
        manifestPlaceholders = [  
            onesignal_app_id: 'PUT YOUR ONESIGNAL APP ID HERE',  
            // Project number pulled from dashboard, local value is ignored.  
            onesignal_google_project_number: 'REMOTE'  
        ]  
    }  
}
```

รูปที่ ข.10 โค้ดที่ต้องเพิ่มในส่วนที่ 3

เมื่อทำขั้นตอนนี้เสร็จสิ้นเป็นที่เรียบร้อยแล้ว ส่วนถัดมาที่จะต้องทำนั้นก็จะเป็น subscriber ของ App ที่สร้างใน OneSignal เนื่องจากหลักการของ OneSignal ในการส่งการแจ้งเตือนนั้นจะเป็นการส่งไปยัง Subscriber ทุกคนโดย Subscriber อาจจะกำหนด tag ไว้สำหรับให้ผู้ส่งสามารถกำหนด tag ผู้รับได้ เช่น โทรศัพท์ เครื่องที่ 1 มี tag เป็น 01 และโทรศัพท์ เครื่องที่ 2 มี tag เป็น 02 เมื่อผู้ที่ต้องการส่งการแจ้งเตือนอยากส่งการแจ้งเตือนไปหาเพียงโทรศัพท์ เครื่องที่ 1 เท่านั้นผู้ส่งเพียงกำหนด tag เป็น 01 เพียงเท่านั้นผู้ส่งก็จะส่งการแจ้งเตือนไปหาโทรศัพท์ เครื่องที่ 1 เพียงเครื่องเดียวเท่านั้น

เมื่อได้ทำตามถึงขั้นตอนของรูปที่ ข.10 แล้วนั้น แอปพลิเคชันนั้นจะเป็น subscriber ของ App ใน OneSignal เป็นที่เรียบร้อยซึ่งในเว็บ OneSignal นั้นสามารถที่จะทำการทดสอบส่ง Message ไปยัง subscriber ทุกเครื่องได้โดยตัวอย่างเว็บ OneSignal ดังรูปที่ ข.11



รูปที่ ข.11 ตัวอย่างเว็บในการทดสอบการส่งการแจ้งเตือนไปยัง subscriber

จากนั้นเมื่อต้องการให้แอปพลิเคชันของเราสามารถส่งการแจ้งเตือนหรือมีการรับการแจ้งเตือนจากผู้อื่นได้จำเป็นที่จะต้องเพิ่มโค้ดดังรูปที่ ข.12 ไปยังแอปพลิเคชัน

```
public class YourAppClass extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        OneSignal.startInit(this).init();
    }
}
```

รูปที่ ข.12 โค้ดที่ไว้ใช้เริ่มต้น OneSignal ในแอปพลิเคชัน

จากนั้นถ้าอยากกำหนด tag ให้กับเครื่องตัวเองแล้วนั้นจำเป็นตัวมีคำสั่งนี้ ดังรูปที่ ข.13 โดยจะต้องกำหนด key และค่า value

```
OneSignal.sendTag("key", "value");
```

รูปที่ ข.13 โค้ดตัวอย่างของการกำหนด tag

ซึ่งถ้าต้องการส่งการแจ้งเตือนไปยังผู้ใช้นั้นนั้นจะมีโค้ดตัวอย่างดังรูปที่ ข.14 และรูปที่ ข.15

```
class CallOneSignalValidated extends AsyncTask<String,String,String>{

    @Override
    protected String doInBackground(String... strings) {
        try {
            String jsonResponse;
            System.out.println("Callllllllllllllllllllllllllllllllll");
            URL url = new URL("https://onesignal.com/api/v1/notifications");
            HttpURLConnection con = (HttpURLConnection)url.openConnection();
            con.setUseCaches(false);
            con.setDoOutput(true);
            con.setDoInput(true);

            con.setRequestProperty("Content-Type", "application/json; charset=UTF-8");
            con.setRequestProperty("Authorization", "Basic MjJiMWE4OWYtYTNhNy00YTg5LTk5MTQzNTIzOGQ1MDRmYTkw");
            con.setRequestMethod("POST");

            String strJsonBody = "{
                + \"app_id\": \"83dd8ed7-35a9-48c3-8601-84505c652ace\",
                + \"included_segments\": [\"All\"],
                + \"filters\": [{\"field\": \"tag\", \"key\": \"member_id\", \"relation\": \"=\", \"value\": \"\"+member_id+\"\"},
                + \"data\": {\"foo\": \"Bar\"},
                + \"contents\": {\"en\": \"Your order has been validated\"}
            }";

            // System.out.println("filters: [{\"field\": \"tag\", \"key\": \"member_id\", \"relation\": \"=\", \"value\": \"\"+member_id+\"\"}");
            System.out.println("member id in AsyncTask : "+member id);
            System.out.println("strJsonBody:\n" + strJsonBody);

            byte[] sendBytes = strJsonBody.getBytes("UTF-8");
            con.setFixedLengthStreamingMode(sendBytes.length);

            OutputStream outputStream = con.getOutputStream();
            outputStream.write(sendBytes);

            int httpResponse = con.getResponseCode();
            System.out.println("httpResponse: " + httpResponse);

            // System.out.println("filters: [{\"field\": \"tag\", \"key\": \"member_id\", \"relation\": \"=\", \"value\": \"\"+member_id+\"\"}");
            System.out.println("member id in AsyncTask : "+member id);
            System.out.println("strJsonBody:\n" + strJsonBody);

            byte[] sendBytes = strJsonBody.getBytes("UTF-8");
            con.setFixedLengthStreamingMode(sendBytes.length);

            OutputStream outputStream = con.getOutputStream();
            outputStream.write(sendBytes);

            int httpResponse = con.getResponseCode();
            System.out.println("httpResponse: " + httpResponse);

            if (httpResponse >= HttpURLConnection.HTTP_OK
                && httpResponse < HttpURLConnection.HTTP_BAD_REQUEST) {
                Scanner scanner = new Scanner(con.getInputStream(), "UTF-8");
                jsonResponse = scanner.useDelimiter("\\A").hasNext() ? scanner.next() : "";
                scanner.close();
            }
            else {
                Scanner scanner = new Scanner(con.getErrorStream(), "UTF-8");
                jsonResponse = scanner.useDelimiter("\\A").hasNext() ? scanner.next() : "";
                scanner.close();
            }
            System.out.println("jsonResponse:\n" + jsonResponse);

        } catch (Throwable t) {
            t.printStackTrace();
            System.out.println("thrown one signal "+t);
        }

        return null;
    }
}
```

รูปที่ ข.14 โค้ดตัวอย่างสำหรับการส่งการแจ้งเตือน

รูปที่ ข.15 โค้ดตัวอย่างสำหรับการส่งการแจ้งเตือน(ต่อ)

ซึ่งส่วนที่เปลี่ยนแปลงจะมีเพียงส่วนที่เป็น String JSON Body เท่านั้นดังรูปที่ ข.16

```
String strJsonBody = "{"
    + "\"app_id\": \"83dd8ed7-35a9-48e3-8601-84505c652aee\",",
    + "\"included_segments\": [\"All\"],",
    + "\"filters\": [{\"field\": \"tag\", \"key\": \"member_id\", \"relation\": \"=\", \"value\": \"\"+member_id+\"}],",
    + "\"data\": {\"foo\": \"bar\"},",
    + "\"contents\": {\"en\": \"Your order has been validated\"}",
    + "}";

System.out.println("\"filters\": [{"field": \"tag\", \"key\": \"member_id\", \"relation\": \"=\", \"value\": \"\"+member_id+\""}");
System.out.println("member_id in AsyncTask : "+member_id);
System.out.println("strJsonBody:\n" + strJsonBody);
```

รูปที่ ข.16 โค้ดส่วน String JSON Body

โดยส่วนที่ต้องแก้นั้นจะเป็นส่วน app_id ให้ตรงกับ app_id ในเว็บ OneSignal ที่ต้องการ subscriber ในส่วนถัดมาเป็นส่วน filter ซึ่งจะทำให้การตรวจสอบ tag ของ subscriber ว่ามี tag ที่ตรงกับเงื่อนไขของ filter ก็จะสามารถส่งการแจ้งเตือนได้ และสามารถกำหนดข้อความที่จะส่งไปในส่วนของ content ได้ ทั้งหมดนี้เป็นเพียงส่วนภาคส่งเท่านั้นเป็นการส่งการแจ้งเตือนไปยัง subscriber ที่ต้องการ

แต่ในส่วนของภาครับนั้นแล้วแต่ความต้องการของผู้พัฒนาที่ต้องการที่จะออกแบบคำสั่งว่าเมื่อได้รับการแจ้งเตือนแล้วจะจัดการกับมันอย่างไรต่อ โดยในปฏิบัติงานนี้ได้้นำการรับการแจ้งเตือนมาทำการรีเฟรชหน้ารายการสินค้า ซึ่งโค้ดตัวอย่างที่ใช้ในการรับการแจ้งเตือนนั้นเป็นการดึงคำสั่งของ OneSignal มาใช้คือ NotificationReceivedHandler ดังรูปที่ ข.17

```
public class NotificationReceiveHandler implements OneSignal.NotificationReceivedHandler {
    Context ctx;

    public NotificationReceiveHandler(Context context) { this.ctx = context; }

    @Override
    public void notificationReceived(OSNotification notification) {
        System.out.println("New message ja");
        // System.out.println("notificationID : "+notificationID);
        // broadcastIntent();
    }

    public void broadcastIntent() {
        Intent intent = new Intent();
        intent.setAction("com.gosoftmobileappshop.CUSTOM_INTENT");
        ctx.sendBroadcast(intent);
    }
}
```

รูปที่ ข.17 โค้ดตัวอย่าง NotificationReceivedHandler

ประวัติผู้เขียน

หัวข้อโครงการ แอปพลิเคชันการสั่งซื้อและจัดส่งสินค้าสำหรับร้านค้า

ชื่อ-สกุล นายณัฐภัทร สุวิสิษฐ์

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์ สาขาวิชา วิศวกรรมสารสนเทศ

ประวัติส่วนตัว เกิดวันที่ 23 มิถุนายน 2540 ที่อยู่ 39/13 ม.อรุณนิเวศน์ ซอย นวมินทร์ 163

ถนน นวลจันทร์ เขตบึงกุ่ม แขวง คลองกุ่ม กรุงเทพฯ 10240

ประวัติการศึกษา

2558- ปัจจุบัน สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2552-2558 โรงเรียน อัสสัมชัญ บางรัก