



รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบประกันภัยบนโมบายล์แพลตฟอร์ม
Insurance System on Mobile Platform



นายรัตน์ แผงสุด
นายสุทธิรัตน์ กระจ่างทอง
นางสาวปรียานุช อันประเสริฐ

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559



T148563

รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบประกันภัยบนโมบายล์แพลตฟอร์ม

Insurance System on Mobile Platform

นายรัตน์ะ แพงสุด

นายสุทธิรัตน์ กระจดาชทอง

นางสาวปรียานุช อันประเสริฐ

ส.ท.
ร.374 ร

เลขหมู่.....2559
เลขทะเบียน.....148563
วันเดือนปี..... 6 พ.ย. 2560

b. 108371494
i.

ภาควิชา วิศวกรรมคอมพิวเตอร์

สาขาวิชา วิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2559

ชื่อโครงการสหกิจศึกษา ระบบประกันภัยบนโมบายล์แพลตฟอร์ม

ชื่อ-สกุล นักศึกษา นายรัตนะ แพงสุด

นายสุทธิรัตน์ กระดาษทอง

นางสาวปรียานุช อันประเสริฐ

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมคอมพิวเตอร์

สาขาวิชา วิศวกรรมสารสนเทศ

ชื่อ-สกุล อาจารย์นิเทศ ดร.พิกุลแก้ว ตังติสานนท์

ชื่อ-สกุล ผู้นิเทศงาน ดร.พิกุลแก้ว ตังติสานนท์

สถานประกอบการ บริษัท ออคตาگون อินเทอร์เน็ตคอฟี จำกัด

บทคัดย่อ

รูปแบบการให้บริการประกันภัยในปัจจุบัน ยังมีความล่าช้า ความซับซ้อน ความยุ่งยากในการใช้บริการอยู่ ทั้งในมุมมองของลูกค้าผู้เอาประกัน เช่น การใช้งานบัตรประกันภัย การดูรายละเอียด และผลประโยชน์กรมธรรม์ การซื้อและต่ออายุกรมธรรม์ ซึ่งต้องทำผ่านตัวแทนของบริษัทประกันภัย ทั้งหมด และตัวบริษัทประกันภัยเอง การให้บริการแจ้งเคลม การขายประกัน การติดต่อกับลูกค้า ก็ยังทำได้ไม่ดีนัก

ในปัจจุบันมีการใช้งานสมาร์ตโฟนอย่างกว้างขวาง การนำบริการด้านประกันภัยมาให้บริการในรูปแบบโมบายล์แอปพลิเคชันจึงเป็นตัวเลือกที่เหมาะสมกับรูปแบบการใช้ชีวิตในปัจจุบัน ที่ต้องการความรวดเร็วและสะดวกสบาย ดังนั้น ปรินูญานินพนธ์ฉบับนี้จึงได้พัฒนาระบบการให้บริการประกันภัยบนสมาร์ตโฟนในรูปแบบของแอปพลิเคชัน ซึ่งระบบการให้บริการประกันภัยแบบใหม่นี้ จะช่วยให้ทั้งลูกค้าของบริษัทประกันภัยใช้บริการประกันภัยได้อย่างสะดวกสบาย และลูกค้าที่เป็นตัวบริษัทประกันภัยเอง จะสามารถให้บริการลูกค้าของบริษัทได้อย่างรวดเร็วเช่นกัน รวมถึงลดค่าใช้จ่ายที่เกิดจากการใช้งานระบบแบบเก่าได้ โดยใช้เทคโนโลยีต่าง ๆ ที่มีบนสมาร์ตโฟนระบบปฏิบัติการ แอนดรอยด์ (Android) ทำงานร่วมกับส่วนต่อประสานต์โปรแกรมประยุกต์ แบบเรสฟูล (Representational state transfer Application Program Interface : RESTful API) เพื่อให้ได้ระบบที่ทำงานร่วมกันได้แบบออนไลน์ตลอดเวลา มาพัฒนาระบบดังกล่าว

คำสำคัญ : ประกันภัย, แอปพลิเคชัน, สมาร์ตโฟน, โปรแกรมประยุกต์แบบเรสฟูล

Cooperative Title: Insurance System on Mobile Platform

Student intern name: Rattana Fangsut

Suttirat Kradatong

Preeyanuch Anprasert

Faculty: Engineering

Department: Computer Engineering

Program: Information Engineering

Advisor name: Dr. Pikulkaew Tangtisanon

Mentor name: Dr. Pikulkaew Tangtisanon

Company: Octagon Interactive co. Ltd

ABSTRACT

Nowadays, Insurance service system has several duplicated working processes which have affected consumer directly such as how complicated that consumer facing when they want to check their insurance detail and benefit or when they want to renew the insurance period.

Presently, Smartphone is playing a major role in daily life. The combination of Insurance service system and Smartphone application would consider to be an appropriated choice for Insurance Business to serve their customer in these days. So this Project will represent about the development of Insurance mobile application on Android operation system which is widely used currently and using RESTful API technology to provide the real-time interactive service for customer.

Eventually this Insurance mobile application would help consumer to get the service faster and more comfortable. Moreover the Insurance mobile application could replace the gap between consumer and business and it can reduce cost in the business side as well.

Keyword: Insurance service, Application, Smartphone, RESTful AP

กิตติกรรมประกาศ

ตามที่คณะผู้จัดทำได้ปฏิบัติงานสหกิจศึกษา ณ บริษัท ออกดากอน อินเทอร์แอกทีฟ (Octagon Interactive) ตั้งแต่วันที่ 8 สิงหาคม ถึงวันที่ 30 พฤษภาคม พ.ศ. 2559 ในหัวข้อโครงการระบบประกันภัยบนโมบายล์แพลตฟอร์ม (Insurance System on Mobile Platform) ทำให้คณะผู้จัดทำได้รับความรู้และประสบการณ์ในการทำงานที่นอกเหนือจากในตำราเรียน เกิดความรู้ความเข้าใจในการทำงานร่วมกับบุคคลอื่นที่มีแตกต่างกันทางความคิด ทักษะ ทักษะ ช่วงอายุ ประสบการณ์ และการทำงานร่วมกันเป็นทีมในระดับองค์กร ซึ่งเป็นสิ่งสำคัญที่ควรคำนึงถึงในการทำงานทุกสายงานอาชีพ การปฏิบัติงานสหกิจศึกษาครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความช่วยเหลือและสนับสนุนจากหลายฝ่ายดังนี้

- | | |
|-----------------------------|--------------------------------------|
| 1. คุณจิระพงษ์ มุคำ | ตำแหน่ง CTO/Co-founder |
| 2. คุณบุศ อินชัย | ตำแหน่ง UX/UI Designer |
| 3. คุณรุจิรา วงศ์เบญจทรัพย์ | ตำแหน่ง Web Developer |
| 4. คุณพีระพัฒน์ พงษ์ผลา | ตำแหน่ง Backend Developer |
| 5. คุณตติวัฒน์ บวรรัตนากิจ | ตำแหน่ง Mobile Application Developer |
| 6. คุณปณิธิ ดาวนนท์ | ตำแหน่ง Android Developer |

รวมไปถึงบุคลากรท่านอื่นๆ ที่มีได้กล่าวถึง ที่ได้ให้คำแนะนำและช่วยเหลือในการจัดทำโครงการนี้คณะผู้จัดทำขอขอบพระคุณ ดร.พิกุลแก้ว ดังดีसानนท์ ที่มีส่วนในการให้ข้อมูลและเป็นพี่ปรึกษาในการจัดทำปฏิญญาพันธบัตรฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้คำแนะนำและช่วยเหลือทั้งในเรื่องการทำงานและการวางตัวให้เหมาะสมในระดับองค์กร ซึ่งล้วนเป็นประโยชน์ในการปฏิบัติงานสหกิจและการทำงานในอนาคต ทางคณะผู้จัดทำขอขอบพระคุณไว้ ณ ที่นี้ด้วย

สุดท้ายนี้คณะผู้จัดทำขอขอบพระคุณบิดามารดา และครอบครัว ซึ่งเปิดโอกาสให้ได้รับการศึกษาเล่าเรียน ตลอดจนดูแลช่วยเหลือและให้กำลังใจคณะผู้จัดทำเสมอมา

นายรัตน์ะ แพงสุด
นายสุทธิรัตน์ กระจดาชทอง
นางสาวปริยานุช อันประเสริฐ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	X
บทที่ 1	1
1.1. ความเป็นมาและความสำคัญของงานวิจัย	1
1.2. วัตถุประสงค์ของงานวิจัย.....	1
1.3. ขอบเขตของการวิจัย (ส่วนการออกแบบ).....	2
1.4. ขอบเขตของการวิจัย (ส่วนการพัฒนาแอปพลิเคชัน).....	3
1.5. ขอบเขตของการวิจัย (ส่วนระบบหลังบ้าน).....	3
1.6. วิธีการดำเนินงานวิจัย	4
1.7. อุปกรณ์สำหรับการดำเนินงานวิจัย	4
1.8. ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2	7
2.1. ทฤษฎีกระบวนการคิดเชิงออกแบบ (Design Thinking).....	7
2.2. อไจล์ (Agile) แนวคิดในการพัฒนาผลิตภัณฑ์	10
2.3. ทฤษฎีเอ็มวีพี (MVP)	13
2.4. ทฤษฎีการออกแบบยูเอ็กซ์ยูไอ (UX/UI) ด้วยเมททีเรียลดีไซน์ (Material Design)	16
2.5. กระบวนการในการสร้างยูเอ็กซ์ยูไอ (UX/UI) ให้กับแอปพลิเคชัน (Application)	41
2.6. แอนดรอยด์ (Android).....	44
2.7. ภาษาจาวา	54
2.8. ภาษาเอกซ์เอ็มแอล.....	57
2.9. แนวคิดการเขียนโปรแกรมแบบเอ็มวีซี	59
2.10. การเขียนผังงาน (Flowchart).....	60
2.11. ยูเอ็มแอล (Unified Modeling Language : UML).....	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

ภาพที่	หน้า
2.12. ภาษาพีเอชพี (PHP)	65
2.13. ลาลาเวลเฟรมเวิร์ค (Laravel).....	71
2.14. ส่วนต่อประสานโปรแกรมประยุกต์แบบ เรสฟูล (RESTful API)	73
2.15. ระบบปฏิบัติการลินุกซ์ (Linux).....	76
2.16. โปรแกรม Postman	77
บทที่ 3	78
3.1. การออกแบบ	78
3.2. ส่วนหน้าบ้าน-โมบายล์แอปพลิเคชัน	85
3.3. ออกแบบและพัฒนาระบบหลังบ้านและส่วนต่อประสานโปรแกรมประยุกต์	110
บทที่ 4	142
4.1. ผลของการออกแบบแอปพลิเคชัน	142
บทที่ 5	156
5.1. การใช้งานแอปพลิเคชัน	156
บทที่ 6	173
6.1. ผลการดำเนินงานในขั้นตอนการพัฒนา	173
6.2. การจำลองการร้องขอเพื่อทดสอบส่วนต่อประสานโปรแกรมประยุกต์ด้วยโปรแกรม Postman	193
6.3. ผลลัพธ์การทำงานของส่วนต่อประสานโปรแกรมประยุกต์ที่ถูกต้อง	196
6.4. ผลลัพธ์การทำงานของส่วนต่อประสานโปรแกรมประยุกต์เมื่อเกิดข้อผิดพลาด	217
บทที่ 7	223
7.1. บทสรุปการวิจัย	223
7.2. ปัญหาที่พบในระหว่างการทำงาน	223
7.3. แนวทางการแก้ไข	224
7.4. แนวทางการพัฒนาต่อและนำไปใช้	224

สารบัญตาราง

ตารางที่	หน้า
ตารางที่ 2.1 ตารางแสดงค่า dpi เทียบกับขนาดหน้าจอ.....	26
ตารางที่ 2.2 ตารางรุ่นต่างๆ ของแอนดรอยด์และจำนวนผู้ใช้งานในแต่ละรุ่น.....	45
ตารางที่ 2.3 สัญลักษณ์ของไฟล์ชาร์ต	60
ตารางที่ 2.4 สัญลักษณ์ที่ใช้ในแผนภาพยูสเคส	63
ตารางที่ 2.5 สัญลักษณ์ที่ใช้ในแผนภาพลำดับเหตุการณ์.....	64
ตารางที่ 3.1 ยูสเคสสมัครสมาชิก	92
ตารางที่ 3.2 ยูสเคสล็อกอิน.....	93
ตารางที่ 3.3 ยูสเคสซื้อประกันออนไลน์.....	94
ตารางที่ 3.4 ยูสเคสติดต่อเจ้าหน้าที่ขายประกัน	95
ตารางที่ 3.5 ยูสเคสขอข้อมูลเพิ่มเติม	96
ตารางที่ 3.6 ยูสเคสค้นหาสถานที่	97
ตารางที่ 3.7 ยูสเคสโทรฉุกเฉิน	98
ตารางที่ 3.8 ยูสเคสแจ้งเคลม	99
ตารางที่ 3.9 ยูสเคสติดต่อบริษัท	100
ตารางที่ 3.10 ยูสเคสบัตรประกันภัย.....	101
ตารางที่ 4.1 ตารางแสดงความพึงพอใจและข้อเสนอแนะของลูกค้าและนักพัฒนาต่อพีเจอร์	142
ตารางที่ 4.2 ตารางแสดงความพึงพอใจและข้อเสนอแนะของลูกค้าต่อแบบจำลองแอปพลิเคชัน... ..	144
ตารางที่ 6.1 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับล็อกอิน	196
ตารางที่ 6.2 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับล็อกอิน.....	196
ตารางที่ 6.3 รูปแบบของ attribute ภายในresponse ของ API สำหรับล็อกอิน.....	196
ตารางที่ 6.4 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับสมัครสมาชิก.....	197
ตารางที่ 6.5 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับสมัครสมาชิก... ..	197
ตารางที่ 6.6 รูปแบบของ attribute ภายในresponse ของ API สำหรับสมัครสมาชิก.....	197
ตารางที่ 6.7 รูปแบบ request และ HTTP Verb เพื่อร้องขอไปยัง API สำหรับกู้รหัสผ่าน	198
ตารางที่ 6.8 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับกู้รหัสผ่าน.....	198
ตารางที่ 6.9 รูปแบบของ attribute ภายใน response ของ API สำหรับกู้รหัสผ่าน	198
ตารางที่ 6.10 รูปแบบกริยา HTTP เพื่อร้องขอไปยัง API สำหรับหน้าหลัก.....	200
ตารางที่ 6.11 รูปแบบของ Header ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าหลัก.....	200
ตารางที่ 6.12 รูปแบบของ attribute ในการตอบกลับของ API สำหรับหน้าหลัก.....	200
ตารางที่ 6.13 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแสดงรายละเอียด โปรโมชั่น.....	201
ตารางที่ 6.14 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแสดงรายละเอียด โปรโมชั่น.....	201

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
ตารางที่ 6.15 รูปแบบของ attribute ภายใน response ของAPI สำหรับหน้าแสดงรายละเอียดโปรโมชัน.....	202
ตารางที่ 6.16 รูปแบบ request และ HTTP Verb เพื่อส่งไปยังAPI สำหรับหน้าซื้อประกันออนไลน์.....	203
ตารางที่ 6.17 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าซื้อประกันออนไลน์.....	203
ตารางที่ 6.18 รูปแบบของ attribute ภายใน response ของAPI สำหรับหน้าซื้อประกันออนไลน์.....	203
ตารางที่ 6.19 request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าบัตรประกันของฉฉ.....	204
ตารางที่ 6.20 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าบัตรประกันของฉฉ.....	204
ตารางที่ 6.21 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าบัตรประกันของฉฉ.....	204
ตารางที่ 6.22 รูปแบบ request และ HTTP Verb เพื่อส่งไปยังAPI สำหรับหน้าบัตรประกันจำลอง.....	205
ตารางที่ 6.23 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าบัตรประกันจำลอง.....	205
ตารางที่ 6.24 รูปแบบของ attribute ภายใน response ของAPI สำหรับหน้าบัตรประกันจำลอง.....	205
ตารางที่ 6.25 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแผนที่สถานที่แต่ละแบบ.....	207
ตารางที่ 6.26 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่สถานพยาบาล.....	207
ตารางที่ 6.27 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่อยู่ช่อมรถ.....	208
ตารางที่ 6.28 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวแต่ละแบบ.....	210
ตารางที่ 6.29 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อดึงข้อมูลส่วนตัวปัจจุบัน.....	210
ตารางที่ 6.30 รูปแบบของ attribute ภายใน response ของAPI สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อดึงข้อมูลส่วนตัวปัจจุบัน.....	210
ตารางที่ 6.31 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อบันทึกข้อมูลส่วนตัว.....	211

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
ตารางที่ 6.32 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อบันทึกข้อมูลส่วนตัว	211
ตารางที่ 6.33 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูลส่วนตัว	212
ตารางที่ 6.34 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัย	212
ตารางที่ 6.35 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเพื่อบันทึกข้อมูลที่อยู่อาศัยปัจจุบัน	213
ตารางที่ 6.36 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเมื่อบันทึกข้อมูลที่อยู่อาศัยปัจจุบัน	213
ตารางที่ 6.37 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเพื่อบันทึกข้อมูลที่อยู่อาศัย	214
ตารางที่ 6.38 รูปแบบของ attribute ที่ใช้ส่งค่าใน HTTP body ไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเพื่อบันทึกข้อมูลที่อยู่อาศัย	214
ตารางที่ 6.39 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูลส่วนตัว	214
ตารางที่ 6.40 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าตั้งค่า	215
ตารางที่ 6.41 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าตั้งค่าเพื่อขอข้อมูลการตั้งค่าปัจจุบัน	215
ตารางที่ 6.42 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าตั้งค่าเพื่อขอข้อมูลการตั้งค่าปัจจุบัน	215
ตารางที่ 6.43 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าตั้งค่าเพื่อบันทึกการตั้งค่าการแจ้งเตือน	216
ตารางที่ 6.44 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าตั้งค่าเพื่อบันทึกการตั้งค่าการแจ้งเตือน	216
ตารางที่ 6.45 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง สำหรับแสดงความคิดเห็น	216
ตารางที่ 6.46 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง สำหรับแสดงความคิดเห็น	216
ตารางที่ 6.47 รูปแบบของ attribute ภายใน response ของ API สำหรับแสดงความคิดเห็น	217

สารบัญภาพ

ภาพที่	หน้า
ภาพที่ 2.1 ขั้นตอนกระบวนการคิด Design thinking ของ Stanford d.school.....	8
ภาพที่ 2.2 ขั้นตอนกระบวนการคิด Design thinking ของ UK Design Council.....	8
ภาพที่ 2.3 Scrum Software Development Process.....	11
ภาพที่ 2.4 ความแตกต่างระหว่าง Agile กับ Waterfall.....	12
ภาพที่ 2.5 หนังสืออีบุ๊ก Minimum Viable Product จากเว็บไซต์ UXPin.....	13
ภาพที่ 2.6 How to Build a MVP.....	14
ภาพที่ 2.7 ภาพแสดงตัวอย่าง Material Design.....	16
ภาพที่ 2.8 ภาพแสดงความรู้สึก แสง และเงาของ Material Design.....	17
ภาพที่ 2.9 ภาพแสดงความชัดเจน และการแยกแยะวัตถุของ Material Design.....	17
ภาพที่ 2.10 ภาพแสดงการเคลื่อนไหวและการเปลี่ยนของ Material Design.....	18
ภาพที่ 2.11 ภาพแสดงการเคลื่อนไหวในรูปแบบที่แตกต่างกัน.....	19
ภาพที่ 2.12 การเคลื่อนไหวแบบระลอกคลื่นหมึก (ink ripples).....	19
ภาพที่ 2.13 แสดงการเคลื่อนไหวการเชื่อมต่อกันระหว่างพื้นผิวและวัตถุ.....	20
ภาพที่ 2.14 กราฟแสดงความแตกต่างการเคลื่อนไหวแบบ No Easing และ Easing.....	20
ภาพที่ 2.15 ภาพการเคลื่อนไหวแบบ Aware.....	21
ภาพที่ 2.16 ภาพการเคลื่อนไหวแบบ Intentional.....	21
ภาพที่ 2.17 ภาพแสดงรูปแบบการใช้สีของ Material Design.....	22
ภาพที่ 2.18 ภาพการใช้สีเพื่อใช้เน้นปุ่มและข้อความ.....	23
ภาพที่ 2.19 การใช้สีตัวอักษรกับพื้นหลังสว่าง.....	23
ภาพที่ 2.20 การใช้สีตัวอักษรกับพื้นหลังมืด.....	23
ภาพที่ 2.21 ภาพแสดงตัวอย่างไอคอนของ Material Design.....	24
ภาพที่ 2.22 ฟอนต์ Roboto และฟอนต์ Noto.....	24
ภาพที่ 2.23 ภาพแสดงความแตกต่างของหน้าจอเทียบกับหน่วย dp.....	25
ภาพที่ 2.24 ตัวอย่างเส้นกริดพื้นฐานของที่ใช้กับหน้าจอ.....	26
ภาพที่ 2.25 ตัวอย่างเส้นกริดพื้นฐานที่ใช้กับข้อความ.....	27
ภาพที่ 2.26 ตัวอย่างการใช้เส้นกริดบนหน้าจอมือถือ.....	27
ภาพที่ 2.27 ภาพโครงสร้างยูไอ (UI) บนโทรศัพท์มือถือ.....	28
ภาพที่ 2.28 ตัวอย่างการซ่อนและแสดงผลของเมนูในหน้าจอทั้งสองแบบ.....	29
ภาพที่ 2.29 ตัวอย่างการซ่อนและแสดงผลของข้อมูลบนการ์ดวิวในหน้าจอทั้งสองแบบ.....	29
ภาพที่ 2.30 ตัวอย่างการเปลี่ยนรูปแบบของข้อมูลในหน้าจอทั้งสองแบบ.....	30
ภาพที่ 2.31 ตัวอย่างการเปลี่ยนรูปแบบเมนูตั้งค่าในหน้าจอทั้งสองแบบ.....	30
ภาพที่ 2.32 ตัวอย่างการแบ่งพื้นที่ว่างในหน้าจอทั้งสองแบบ.....	31
ภาพที่ 2.33 ตัวอย่างการจัดเลย์เอาต์ที่เหมาะสมในหน้าจอทั้งสองแบบ.....	31
ภาพที่ 2.34 ตัวอย่างยูไอของนาฬิกาในหน้าจอทั้งสองแบบ.....	31
ภาพที่ 2.35 ตัวอย่างการขยายของเลย์เอาต์ในหน้าจอทั้งสองแบบ.....	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 2.36 ตัวอย่างการขยายของเลย์เอาต์ในหน้าจอทั้งสองแบบ.....	32
ภาพที่ 2.37 ภาพปุ่มเมนูด้านล่างของหน้าจอมือถือ แท็บเล็ต และคอมพิวเตอร์ส่วนบุคคล	33
ภาพที่ 2.38 ตัวอย่างการใช้งานปุ่มเมนูด้านล่าง (bottom navigation)	34
ภาพที่ 2.39 ตัวอย่างการใช้งานปุ่มเมนูด้านล่าง (bottom navigation)	34
ภาพที่ 2.40 ปุ่มลอย (Floating action button).....	35
ภาพที่ 2.41 ปุ่มแบน (Flat button).....	35
ภาพที่ 2.42 ปุ่มยก (Raised button).....	36
ภาพที่ 2.43 การ์ด (Cards).....	36
ภาพที่ 2.44 กล่องแจ้งเตือน (Dialog)	37
ภาพที่ 2.45 ลิสต์ (List).....	37
ภาพที่ 2.46 ประเภทของลิสต์	38
ภาพที่ 2.47 พิกเจอร์แสดงวันที่และเวลาที่ถูกล็อกแบบ	38
ภาพที่ 2.48 ช่องทำเครื่องหมาย (Checkboxes).....	39
ภาพที่ 2.49 ปุ่มตัวเลือก (Radio buttons)	39
ภาพที่ 2.50 สวิตช์ (Switches).....	40
ภาพที่ 2.51 แท็บ (Tabs).....	40
ภาพที่ 2.52 แท็บเครื่องมือ (Toolbars).....	41
ภาพที่ 2.53 กระบวนการการออกแบบแอปพลิเคชัน (Process of designing).....	41
ภาพที่ 2.54 แบบร่าง (Wireframe).....	42
ภาพที่ 2.55 แบบจำลอง (Mockup).....	43
ภาพที่ 2.56 โปรโตไทป์ (Prototype).....	44
ภาพที่ 2.57 สถาปัตยกรรมของแอนดรอยด์.....	46
ภาพที่ 2.58 เข้าสู่แอปพลิเคชันและออกจากแอปพลิเคชัน	48
ภาพที่ 2.59 การเปลี่ยนสถานะ onPause และ onResume.....	49
ภาพที่ 2.60 การเปลี่ยนสถานะ onStop และ onStart	50
ภาพที่ 2.61 การสเกลหน้าจอมือถือให้พอดีกับแท็บเล็ต.....	50
ภาพที่ 2.62 ตัวอย่างการจัดเลย์เอาต์ให้เหมาะกับอุปกรณ์แท็บเล็ต.....	51
ภาพที่ 2.63 ตัวอย่างแฟร็กเมนต์บนมือถือ	52
ภาพที่ 2.64 โครงสร้างการทำงานแฟร็กเมนต์.....	52
ภาพที่ 2.65 วงจรชีวิตของแฟร็กเมนต์	53
ภาพที่ 2.66 แสดงองค์ประกอบของภาษาจาวา	55
ภาพที่ 2.67 ขั้นตอนการทำงานภาษาจาวา	56
ภาพที่ 2.68 รูปแบบการเขียนโปรแกรมจาวา.....	57
ภาพที่ 2.69 โครงสร้างเอกสารเอกซ์เอ็มแอล	58
ภาพที่ 2.70 ขั้นตอนการทำงานของ MVC	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 2.71 ผังงานแบบเรียงลำดับ	61
ภาพที่ 2.72 ผังงานที่มีการกำหนดเงื่อนไข	61
ภาพที่ 2.73 ผังงานที่มีการทำงานแบบวนรอบ	62
ภาพที่ 2.74 ความสัมพันธ์แบบเรียกใช้ในแผนภาพยูสเคส	63
ภาพที่ 2.75 ความสัมพันธ์แบบขยายในแผนภาพยูสเคส	64
ภาพที่ 2.76 สัญลักษณ์ของภาษาพีเอชพี.....	65
ภาพที่ 2.77 ตัวอย่างการฝังภาษาสคริปต์ PHP ลงใน HTML	66
ภาพที่ 2.78 ตัวอย่างการประกาศตัวแปรในภาษา PHP	68
ภาพที่ 2.79 Taylor Otwell ผู้พัฒนา Laravel Framework.....	71
ภาพที่ 2.80 ภาพหน้าต่างโปรแกรม Postman.....	77
ภาพที่ 3.1 ความสัมพันธ์ของแต่ละหน้าจอใน Wireframe	80
ภาพที่ 3.2 ตัวอย่างแบบจำลองแอปพลิเคชัน (Mockup).....	81
ภาพที่ 3.3 ตัวอย่างโปรแกรม Flinto	82
ภาพที่ 3.4 ตัวอย่างหน้าจอตอนแสดงผลแอนิเมชันโปรแกรม Flinto	83
ภาพที่ 3.5 บรรยายการทำทดสอบประสบการณ์ผู้ใช้งาน (Usability Test) ที่บริษัทประกันภัย..	84
ภาพที่ 3.6 ตัวอย่างหน้าจอแอปพลิเคชันที่อัปเดตบนโปรแกรม Zeplin	84
ภาพที่ 3.7 ผังงานการทำงานหน้าลือคอิน	85
ภาพที่ 3.8 ผังงานการทำงานหน้าหลัก	86
ภาพที่ 3.9 ผังงานการทำงานหน้าบัตรประกัน.....	87
ภาพที่ 3.10 ผังงานการทำงานหน้าเคลม.....	88
ภาพที่ 3.11 ผังงานการทำงานหน้าแผนที่	89
ภาพที่ 3.12 ผังงานการทำงานหน้าโทรฉุกเฉิน	90
ภาพที่ 3.13 แผนภาพยูสเคสแอปพลิเคชันประกันภัย	91
ภาพที่ 3.14 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการสมัครสมาชิก	91
ภาพที่ 3.15 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการลือคอิน.....	93
ภาพที่ 3.16 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการซื้อประกันออนไลน์....	94
ภาพที่ 3.17 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันติดต่อเจ้าหน้าที่ขายประกัน	95
ภาพที่ 3.18 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการขอข้อมูลเพิ่มเติม	96
ภาพที่ 3.19 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการค้นหาสถานที่	97
ภาพที่ 3.20 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการโทรฉุกเฉิน	98
ภาพที่ 3.21 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการแจ้งเคลม	99
ภาพที่ 3.22 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันติดต่อ SAGI	100
ภาพที่ 3.23 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการใช้บัตรประกันภัย	101
ภาพที่ 3.24 แผนภาพลำดับเหตุการณ์การลือคอิน.....	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ขออนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 3.25 แผนภาพลำดับเหตุการณ์การซื้อประกันออนไลน์.....	103
ภาพที่ 3.26 แผนภาพลำดับเหตุการณ์การติดต่อเจ้าหน้าที่.....	104
ภาพที่ 3.27 แผนภาพลำดับเหตุการณ์การขอข้อมูลเพิ่มเติม.....	104
ภาพที่ 3.28 แผนภาพลำดับเหตุการณ์การค้นหาสถานที่.....	105
ภาพที่ 3.29 แผนภาพลำดับเหตุการณ์การโทรฉุกเฉิน.....	105
ภาพที่ 3.30 แผนภาพลำดับเหตุการณ์การแจ้งเคลม.....	106
ภาพที่ 3.31 แผนภาพลำดับเหตุการณ์การใช้งานบัตร.....	106
ภาพที่ 3.32 แผนภาพลำดับเหตุการณ์การใช้งานบัตร.....	107
ภาพที่ 3.33 แพ็กเกจ model.....	107
ภาพที่ 3.34 แพ็กเกจ view.....	108
ภาพที่ 3.35 แพ็กเกจ fragment.....	108
ภาพที่ 3.36 แพ็กเกจ activity.....	108
ภาพที่ 3.37 แพ็กเกจ util.....	109
ภาพที่ 3.38 แพ็กเกจ service.....	109
ภาพที่ 3.39 แพ็กเกจ DAO.....	109
ภาพที่ 3.40 หน้าตาโปรแกรม Zeplin แสดงการออกแบบไวต์เฟรมของแอปพลิเคชัน.....	110
ภาพที่ 3.41 หน้าจอแอปพลิเคชันขณะล็อกอิน.....	111
ภาพที่ 3.42 หน้าจอแอปพลิเคชันขณะล็อกอิน.....	111
ภาพที่ 3.43 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าสลิปรหัสผ่าน.....	112
ภาพที่ 3.44 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าหลัก.....	112
ภาพที่ 3.45 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแสดงรายละเอียดโปรโมชั่น.....	113
ภาพที่ 3.46 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าซื้อประกันออนไลน์.....	113
ภาพที่ 3.47 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าบัตรประกันของฉัน.....	114
ภาพที่ 3.48 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าบัตรประกันจำลอง.....	115
ภาพที่ 3.49 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแผนที่.....	115
ภาพที่ 3.50 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแก้ไขข้อมูลส่วนตัว.....	116
ภาพที่ 3.51 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแก้ไขข้อมูลที่อยู่อาศัย.....	116
ภาพที่ 3.52 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าตั้งค่า.....	117
ภาพที่ 3.53 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแสดงความคิดเห็น.....	118
ภาพที่ 3.54 Use case diagram ของระบบสวนต่อประสานตโปรแกรมประยุกต์.....	118
ภาพที่ 3.55 Activity diagram แสดงการทำงานของ API ในการลงชื่อเข้าใช้.....	119
ภาพที่ 3.56 Activity diagram แสดงการทำงานของ API ในการกู้รหัสผ่าน.....	119
ภาพที่ 3.57 Activity diagram แสดงการทำงานของ API ในหน้าหลัก.....	120
ภาพที่ 3.58 Activity diagram แสดงการทำงานของ API ในการกดเลือกโปรโมชั่น.....	121
ภาพที่ 3.59 Activity diagram แสดงการทำงานของ API ในการกดเลือกผลิตภัณฑ์.....	122

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 3.60 Activity diagram แสดงการทำงานของ API ในหน้าบัตรประกันของฉัน.....	123
ภาพที่ 3.61 Activity diagram แสดงการทำงานของ API ในหน้าบัตรประกันจำลอง.....	124
ภาพที่ 3.62 Activity diagram แสดงการทำงานของ API ในหน้าแผนที่ขณะเลือกสถานที่เป็นศูนย์ ซ่อม	125
ภาพที่ 3.63 Activity diagram แสดงการทำงานของ API ในหน้าแผนที่ขณะเลือกสถานที่เป็น โรงพยาบาล	126
ภาพที่ 3.64 Activity diagram แสดงการทำงานของ API ในหน้าขณะอยู่ในหน้าแก้ไขข้อมูลส่วนตัว	127
ภาพที่ 3.65 Activity diagram แสดงการทำงานของ API ในหน้าขณะกดบันทึกในหน้าแก้ไขข้อมูล ส่วนตัว	128
ภาพที่ 3.66 Activity diagram แสดงการทำงานของ API ในหน้าขณะอยู่ในหน้าแก้ไขข้อมูลที่อยู่ อาศัย	129
ภาพที่ 3.67 Activity diagram แสดงการทำงานของ API ในขณะกดบันทึกในหน้าแก้ไขข้อมูลที่อยู่ อาศัย	130
ภาพที่ 3.68 Activity diagram แสดงการทำงานของ API ในขณะเข้าสู่หน้าตั้งค่า.....	131
ภาพที่ 3.69 Activity diagram แสดงการทำงานของ API ในขณะมีการเปลี่ยนแปลงการตั้งค่า....	132
ภาพที่ 3.70 Activity diagram แสดงการทำงานของ API ในขณะกดออกจากระบบ.....	133
ภาพที่ 3.71 Activity diagram แสดงการทำงานของ API ในขณะกดส่งความคิดเห็น	134
ภาพที่ 3.72 หน้าต่างโปรแกรม Putty	136
ภาพที่ 3.73 หน้าต่าง console เมื่อ remote SSH ไปยังเครื่องแม่ข่าย.....	137
ภาพที่ 3.74 หน้าต่าง console ขณะสร้างโปรเจกต์ลาเวสด้วยคำสั่ง composer.....	137
ภาพที่ 3.75 หน้าเว็บไซต์ Bitbucket.org หลังจากสร้าง Repository ใหม่.....	138
ภาพที่ 3.76 หน้าต่างโปรแกรม Source Tree.....	139
ภาพที่ 3.77 หน้าต่างโปรแกรม XAMPP	140
ภาพที่ 3.78 แผนภาพแสดงการทำงานในรูปแบบของ Client-Server	140
ภาพที่ 3.79 แผนภาพแสดงการทำงานภายในฝั่ง Server	141
ภาพที่ 4.1 ตัวอย่างหน้าสมัครสมาชิก และหน้าเข้าสู่ระบบ.....	144
ภาพที่ 4.2 ตัวอย่างหน้าตั้งค่า และหน้าลิ้มรสผ่าน.....	145
ภาพที่ 4.3 ตัวอย่างชื่อประกันหรือหน้าหลัก	146
ภาพที่ 4.4 ตัวอย่างหน้าเว็บประกันเมื่อชื่อประกันด้วยแอปพลิเคชัน	146
ภาพที่ 4.5 ตัวอย่างหน้าบัตรประกันสำหรับผู้ที่ยังไม่มีประกัน และหน้ายืนยันตัวตน.....	147
ภาพที่ 4.6 ตัวอย่างหน้าบัตรประกันภัย สำหรับผู้ใช้ที่มีประกันแล้ว.....	148
ภาพที่ 4.7 ตัวอย่างหน้าบัตรประกันภัย สำหรับผู้ใช้ที่มีประกันแล้ว (พีเจอร์ที่จะสร้างในอนาคต)..	148
ภาพที่ 4.8 ตัวอย่างหน้าเคลม (Claim).....	149
ภาพที่ 4.9 ตัวอย่างหน้าแจ้งเคลม แบบไม่มีคู่มือ (1).....	150

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ขออนุญาต

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 4.10 ตัวอย่างหน้าแจ้งเคลม แบบไม่มีคู่กรณี (2).....	150
ภาพที่ 4.11 ตัวอย่างหน้าแจ้งเคลม แบบมีคู่กรณี.....	151
ภาพที่ 4.12 ตัวอย่างหน้าแสดงแผนที่อู่ซ่อมรถ และโรงพยาบาล.....	152
ภาพที่ 4.13 ตัวอย่างหน้านำทาง และโทรฉุกเฉิน.....	152
ภาพที่ 4.14 ตัวอย่างหน้าโปรไฟล์ (1).....	153
ภาพที่ 4.15 ตัวอย่างหน้าโปรไฟล์ (2).....	153
ภาพที่ 4.16 ตัวอย่างแบบจำลองแอปพลิเคชันทั้งหมดที่ส่งต่อให้นักพัฒนาบนโปรแกรม Zeplin (1)	154
ภาพที่ 4.17 ตัวอย่างแบบจำลองแอปพลิเคชันทั้งหมดที่ส่งต่อให้นักพัฒนาบนโปรแกรม Zeplin (2)	154
ภาพที่ 4.18 ตัวอย่างแบบจำลองแอปพลิเคชันทั้งหมดที่ส่งต่อให้นักพัฒนาบนโปรแกรม Zeplin (3)	155
ภาพที่ 5.1 แสดงไอคอนแอปพลิเคชันบนจอโทรศัพท์เคลื่อนที่.....	156
ภาพที่ 5.2 ภาพแสดงวิธีการใช้งานแอปพลิเคชัน.....	157
ภาพที่ 5.3 ภาพแสดงการเข้าสู่ระบบ.....	158
ภาพที่ 5.4 ภาพแสดงการเข้าสู่ระบบผ่านเฟซบุ๊ก.....	159
ภาพที่ 5.5 เลือกรุ่นสีรถสำหรับ.....	159
ภาพที่ 5.6 แสดงหน้าจอสําหรับสมัครสมาชิก.....	160
ภาพที่ 5.7 แสดงหน้าหลัก.....	161
ภาพที่ 5.8 แสดงหน้าโปรไฟล์.....	162
ภาพที่ 5.9 หน้าแก้ไขโปรไฟล์ข้อมูลส่วนตัว.....	163
ภาพที่ 5.10 แสดงการแจ้งเตือน เมื่อกดปุ่มบันทึก.....	163
ภาพที่ 5.11 แสดงหน้าตั้งค่า.....	164
ภาพที่ 5.12 แสดงหน้าแสดงความคิดเห็น.....	165
ภาพที่ 5.13 หน้าซื้อประกันออนไลน์.....	165
ภาพที่ 5.14 หน้ายืนยันตัวตน กรณีใช้งานบัตรครั้งแรก.....	166
ภาพที่ 5.15 หน้าบัตรประกัน กรณียังไม่มีประกัน.....	167
ภาพที่ 5.16 แสดงการแจ้งเตือนเมื่อใส่หมายเลขบัตรประชาชนผิด.....	167
ภาพที่ 5.17 หน้าบัตรประกันของฉัน.....	168
ภาพที่ 5.18 หน้าบัตรประกัน.....	168
ภาพที่ 5.19 แสดงหน้าเลือกลักษณะอุบัติเหตุ.....	169
ภาพที่ 5.20 แสดงหน้าแจ้งเคลม กรณียังไม่มีประกัน.....	170
ภาพที่ 5.21 หน้าแจ้งเคลม.....	170
ภาพที่ 5.22 หน้าแผนที่.....	171
ภาพที่ 5.23 แสดงหน้าโทรฉุกเฉิน.....	172

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 6.1 หน้าต่างโปรแกรม Putty	173
ภาพที่ 6.2 หน้าต่าง console เมื่อ remote SSH ไปยังเครื่องแม่ข่าย	174
ภาพที่ 6.3 หน้าต่าง consoleขณะสร้างโปรเจกต์ลาลาเวลด้วยคำสั่ง composer	174
ภาพที่ 6.4 หน้าต่าง console หลังจากสร้างโปรเจกต์ลาลาเวลด้วยคำสั่ง composer เสร็จสมบูรณ์	175
ภาพที่ 6.5 หน้าต่าง consoleขณะใช้คำสั่ง vimเพื่อเปิดไฟล์ด้วย vim editor.....	175
ภาพที่ 6.6 หน้าต่าง consoleขณะใช้คำสั่ง cd เพื่อเข้าไปภายในโฟลเดอร์โปรเจกต์.....	175
ภาพที่ 6.7 หน้าต่าง consoleขณะเปิดไฟล์ .env ด้วย vim editor.....	176
ภาพที่ 6.8 หน้าต่าง consoleขณะแก้ไขไฟล์ .env ด้วย vim editor.....	176
ภาพที่ 6.9 หน้าเว็บไซต์ Bitbucket.orgหลังจากสร้าง Repository ใหม่.....	177
ภาพที่ 6.10 หน้าต่าง console ขณะสร้าง Local Repository ด้วยคำสั่ง git init	177
ภาพที่ 6.11 หน้าต่าง console ขณะสั่ง Stage ไฟล์ด้วยคำสั่ง git add.....	177
ภาพที่ 6.12 หน้าต่าง consoleขณะใช้คำสั่ง git commit.....	178
ภาพที่ 6.13 หน้าเว็บไซต์ Bitbucket.org ซึ่งมีขั้นตอนการเชื่อม Bitbucket Repositoryกับ Local Repository อธิบายไว้.....	178
ภาพที่ 6.14 ภาพหน้าต่าง console หลังจากการเชื่อม Bitbucket Repository กับ Local Repositoryด้วยคำสั่ง git remote และทำการอัปเดตโปรเจกต์ด้วยคำสั่ง git push	179
ภาพที่ 6.15 หน้าเว็บไซต์ Bitbucket.org ปรากฏกิจกรรมที่ได้ทำที่แท็บ Recent Activity ใน ด้านขวา และรายละเอียดคร่าวๆของโปรเจกต์ที่อัปเดตขึ้นไปบน Bitbucket Git Server	179
ภาพที่ 6.16 หน้าต่างโปรแกรม Source Tree	180
ภาพที่ 6.17 หน้าต่างโปรแกรม Source Tree ขณะทำการ clone โปรเจกต์ลงบนเครื่อง คอมพิวเตอร์	180
ภาพที่ 6.18 โปรแกรม Eclipse Mars ขณะสร้างโปรเจกต์จากโฟลเดอร์โปรเจกต์ที่โคลนจาก Git.	181
ภาพที่ 6.19 หน้าต่างโปรแกรม Eclipse Mars หลังจาก import โปรเจกต์เรียบร้อยแล้ว	182
ภาพที่ 6.20 หน้าต่างโปรแกรม XAMPP	182
ภาพที่ 6.21 หน้าต่างบราวเซอร์แสดงผลการทำงานของลาลาเวลเฟรมเวิร์คบนเครื่อง คอมพิวเตอร์ จากการโคลนโปรเจกต์จากเครื่องแม่ข่าย ผ่านระบบ Git.....	183
ภาพที่ 6.22 หน้าจอโปรแกรมขณะทำการติดตั้งโปรแกรม PostgreSQL.....	183
ภาพที่ 6.23 หน้าจอติดตั้งขณะตั้งค่ารหัสผ่านสำหรับเข้าใช้ระบบฐานข้อมูล PostgreSQL	184
ภาพที่ 6.24 หน้าต่างแสดงโฟลเดอร์ phpadminที่สร้างไว้ใน โฟลเดอร์ xampp	184
ภาพที่ 6.25 ขณะแก้ไขไฟล์ config.inc.php โดยตั้งค่า extra_login_security เป็น falseเพื่อ อนุญาตให้มีการ login เข้าระบบฐานข้อมูลด้วยผู้ใช้ postgres	185
ภาพที่ 6.26 หน้าต่างโปรแกรม Xamppขณะเข้าไปตั้งค่าไฟล์ php.ini	185
ภาพที่ 6.27 ขณะแก้ไขไฟล์ php.iniโดยการลบ ; เพื่อใช้งานส่วนขยาย php_pdo_pgsql และ php_pgsql.....	186

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 6.28 ขณะแก้ไขไฟล์ httpd-xampp.conf โดยการเพิ่มโค้ดตามกรอบสีเหลี่ยมสีแดงเพื่อให้สามารถใช้งาน phpadmin ผ่านหน้าบราวเซอร์บน Apache Webserver ใน XAMPP ได้.....	186
ภาพที่ 6.29 ภาพบราวเซอร์ขณะเข้าสู่หน้า phpadmin บน localhost.....	187
ภาพที่ 6.30 แผนภาพแสดงการทำงานแบบ client -server ที่ติดต่อสื่อสารผ่าน API.....	187
ภาพที่ 6.31 ภาพแสดงส่วนประกอบต่างๆ ระบบที่ทำงานร่วมกันบนเครื่อง server ที่พัฒนาด้วย ลาลาเวล	188
ภาพที่ 6.32 หน้าจอ command prompt ขณะใช้คำสั่ง cd เข้าไปยังโฟลเดอร์โปรเจกต์	188
ภาพที่ 6.33 หน้าจอ command prompt ขณะใช้คำสั่ง Artisan ในการสร้างโมเดล.....	189
ภาพที่ 6.34 หน้าจอโปรแกรม Eclipse Mars แสดงโครงสร้างของโมเดลที่ได้จากการใช้คำสั่ง Artisan	189
ภาพที่ 6.35 หน้าจอ command prompt ขณะใช้คำสั่ง Artisan ในการสร้างคอนโทรลเลอร์.....	189
ภาพที่ 6.36 หน้าจอโปรแกรม Eclipse Mars แสดงโครงสร้างของคอนโทรลเลอร์ที่ได้จากการใช้ คำสั่ง Artisan	190
ภาพที่ 6.37 ภาพแสดงไฟล์โมเดลทั้งหมดที่จำเป็นในการพัฒนาระบบประกันภัยบนโมไบล์ แพลตฟอร์ม.....	190
ภาพที่ 6.38 ภาพแสดงไฟล์คอนโทรลเลอร์ทั้งหมดที่จำเป็นในการพัฒนาระบบประกันภัยบนโมไบล์ แพลตฟอร์ม.....	191
ภาพที่ 6.39 ภาพแสดงไฟล์โมเดลที่เชื่อมต่อกับฐานข้อมูลตาราง timeline.....	191
ภาพที่ 6.40 ภาพแสดงไฟล์คอนโทรลเลอร์ที่ประกอบด้วยเมธอดต่างๆ.....	192
ภาพที่ 6.41 ภาพแสดงไฟล์ route.php ที่ทำหน้าที่ในการกำหนดเส้นทางในการเข้าถึง API	192
ภาพที่ 6.42 ภาพแสดงรูปแบบการเขียน route ในลาลาเวลเฟรมเวิร์ค.....	193
ภาพที่ 6.43 หน้าต่างโปรแกรม Postman แสดงส่วนประกอบสำคัญที่ใช้ในการทดสอบระบบ API	193
ภาพที่ 6.44 โปรแกรม Postman ขณะกำหนด HTTP Verb และ URL ปลายทางให้กับ request	194
ภาพที่ 6.45 โปรแกรม Postman ขณะกำหนด attribute ที่ต้องการส่งค่าไปพร้อมกับ request	194
ภาพที่ 6.46 โปรแกรม Postman ขณะกำหนด Headers ให้กับ request.....	195
ภาพที่ 6.47 โปรแกรม Postman ขณะแสดงผลลัพธ์ที่ได้จาก response	195
ภาพที่ 6.48 response ที่ได้รับกลับมาจาก API สำหรับล็อกอิน.....	197
ภาพที่ 6.49 response ที่ได้รับกลับมาจาก API สำหรับสมัครสมาชิก.....	198
ภาพที่ 6.50 response ที่ได้รับกลับมาจาก API สำหรับกู้รหัสผ่านด้วยอีเมล	198
ภาพที่ 6.51 response ที่ได้รับกลับมาจาก API สำหรับกู้รหัสผ่านด้วยเบอร์โทรศัพท์	198
ภาพที่ 6.52 เมื่อเลือกกู้รหัสผ่านโดยใส่ username เป็นอีเมล API จะตั้งรหัสใหม่แล้วส่งรหัสผ่านไป ยังSMS.....	199

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 6.53 เมื่อเลือกผู้รหัสผ่านโดยใส่ username เป็นอีเมล API จะตั้งรหัสใหม่แล้วส่งรหัสผ่านไป ยังอีเมล.....	199
ภาพที่ 6.54 response ที่ได้รับกลับมาจาก API สำหรับหน้าหลัก	201
ภาพที่ 6.55 response ที่ได้รับกลับมาจาก API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น	202
ภาพที่ 6.56 response ที่ได้รับกลับมาจาก API สำหรับหน้าซื้อประกันออนไลน์	204
ภาพที่ 6.57 response ที่ได้รับกลับมาจาก API สำหรับหน้าบัตรประกันของฉัน	205
ภาพที่ 6.58 response ที่ได้รับกลับมาจาก API สำหรับหน้าหน้าบัตรประกันจำลอง.....	206
ภาพที่ 6.59 response ที่ได้รับกลับมาจาก API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่ สถานพยาบาล	208
ภาพที่ 6.60 response ที่ได้รับกลับมาจาก API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่อยู่ ซ่อมรถ.....	209
ภาพที่ 6.61 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อขอข้อมูลส่วนตัว	211
ภาพที่ 6.62 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูล ส่วนตัวเรียบร้อยแล้ว	212
ภาพที่ 6.63 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเมื่อดึงข้อมูลที่อยู่ อาศัย	213
ภาพที่ 6.64 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูล ส่วนตัวเรียบร้อยแล้ว	214
ภาพที่ 6.65 response ที่ได้รับกลับมาจาก API สำหรับหน้าตั้งค่าเพื่อขอข้อมูลการตั้งค่าปัจจุบัน	215
ภาพที่ 6.66 response ที่ได้รับกลับมาจาก API สำหรับหน้าตั้งค่าเพื่อบันทึกการตั้งค่าการแจ้งเตือน	216
ภาพที่ 6.67 response ที่ได้รับกลับมาจาก API สำหรับแสดงความคิดเห็น.....	217
ภาพที่ 6.68 response ที่ได้กลับมาเมื่อไม่พบชื่อผู้ใช้หรือรหัสไม่ถูกต้องเมื่อมีการเรียกใช้ API สำหรับ ล็อกอิน	217
ภาพที่ 6.69 response ที่ได้กลับมาเมื่อไม่พบข้อมูลที่จำเป็นในการสมัครสมาชิกเมื่อมีการเรียกใช้ API สำหรับสมัครสมาชิก	217
ภาพที่ 6.70 response ที่ได้กลับมาเมื่อตรวจพบการลงทะเบียนเบอร์โทรศัพท์ซ้ำ หรือรูปแบบเบอร์ โทรศัพท์ผิดปกติเมื่อมีการเรียกใช้ API สำหรับสมัครสมาชิก	218
ภาพที่ 6.71 response ที่ได้กลับมาเมื่อตรวจพบรูปแบบเบอร์อีเมลผิดปกติเมื่อมีการเรียกใช้ API สำหรับสมัครสมาชิก	218
ภาพที่ 6.72 response ที่ได้กลับมาเมื่อไม่พบข้อมูลสำหรับยืนยันตัวตนในระบบเมื่อมีการเรียกใช้ API สำหรับผู้รหัสผ่าน.....	218
ภาพที่ 6.73 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าหลัก	218

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
ภาพที่ 6.74 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น.....	219
ภาพที่ 6.75 response ที่ได้กลับมาเมื่อไม่พบรายละเอียดของโปรโมชั่นในระบบที่อ้างอิงด้วย id เมื่อมีการเรียกใช้ API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น.....	219
ภาพที่ 6.76 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าซื้อประกันออนไลน์.....	219
ภาพที่ 6.77 response ที่ได้กลับมาเมื่อไม่พบรายละเอียดของข้อมูลผลิตภัณฑ์ในระบบที่อ้างอิงด้วย id เมื่อมีการเรียกใช้ API สำหรับหน้าซื้อประกันออนไลน์.....	219
ภาพที่ 6.78 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าบัตรประกันของฉัน.....	220
ภาพที่ 6.79 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าบัตรประกันจำลอง.....	220
ภาพที่ 6.80 response ที่ได้กลับมาเมื่อไม่พบการยืนยันบัตรประชาชนเมื่อมีการเรียกใช้ API สำหรับหน้าบัตร.....	220
ภาพที่ 6.81 response ที่ได้กลับมาเมื่อไม่พบประเภทของกรมธรรม์ที่ตรงกับบัตรประชาชนเมื่อมีการเรียกใช้ API สำหรับหน้าบัตรประกันจำลอง.....	220
ภาพที่ 6.82 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าแก้ไขข้อมูลส่วนตัว.....	221
ภาพที่ 6.83 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับแก้ไขข้อมูลที่อยู่อาศัย.....	221
ภาพที่ 6.84 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับสำหรับตั้งค่า.....	221
ภาพที่ 6.85 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับสำหรับแสดงความคิดเห็น.....	221

บทที่ 1

บทนำ

1.1. ความเป็นมาและความสำคัญของงานวิจัย

บริษัทออกตาคอนอินเทอร์แอกทีฟ (Octagon interactive) เป็นบริษัทที่ดำเนินธุรกิจด้านการพัฒนาซอฟต์แวร์บนแพลตฟอร์มต่าง ๆ ทั้งบนเว็บไซต์และแอปพลิเคชันบนสมาร์ทโฟน รวมถึงการวางระบบต่าง ๆ เพื่อให้ซอฟต์แวร์ที่พัฒนาต่อลูกค้าซึ่งคือองค์กรชั้นนำต่าง ๆ ที่ต้องการซอฟต์แวร์หรือระบบที่จะเข้าไปช่วยในการดำเนินธุรกิจเป็นไปตามเป้าหมายของบริษัทนั้น ๆ

ปัจจุบัน บริษัทออกตาคอนอินเทอร์แอกทีฟ ได้รับการว่าจ้างจากบริษัทเอกชนรายหนึ่ง ที่ดำเนินธุรกิจเกี่ยวกับประกันภัย ซึ่งทางบริษัทเอกชนรายนี้มีความต้องการที่จะดำเนินธุรกิจประกันภัยในรูปแบบใหม่ที่มีความสะดวก เข้าถึงได้ง่าย และให้บริการได้อย่างรวดเร็ว จึงเล็งเห็นว่าการใช้แพลตฟอร์มโมบายล์เข้ามาช่วยในการดำเนินธุรกิจ เป็นสิ่งที่ตอบโจทย์อย่างมากในปัจจุบันทั้งต่อตัวลูกค้าบริษัทประกันภัย และบริษัทประกันภัยเอง

การให้บริการประกันภัยผ่านโมบายล์แพลตฟอร์ม จะต้องมียุทธศาสตร์ประกอบหลายอย่างที่ทำางานร่วมกัน เพื่อให้ได้ประสิทธิผลตามที่บริษัทประกันต้องการ ไม่ว่าจะเป็นการสร้างแอปพลิเคชันสำหรับสมาร์ทโฟนบนระบบปฏิบัติการแอนดรอยด์ (Android) ให้รองรับกับความหลากหลายของอุปกรณ์สมาร์ทโฟนในปัจจุบัน ไปจนถึงการออกแบบพัฒนา และทดสอบให้ผู้ใช้สามารถใช้งานได้ง่าย ไม่ยุ่งยากซับซ้อน รวดเร็ว เชื่อถือได้ การสร้างระบบแม่ข่าย (Server) เพื่อให้แอปพลิเคชันสามารถสื่อสารกับบริษัทประกันได้ทันทีตามความต้องการ และยังคงคำนึงถึงความปลอดภัยในการออกแบบการสื่อสารผ่านส่วนต่อประสานโปรแกรมประยุกต์ (Application Program Interface : API) อีกด้วย

งานวิจัยชิ้นนี้จึงมีรายละเอียดในการดำเนินงานที่ค่อนข้างมากผู้วิจัยจึงแบ่งขั้นตอนในการพัฒนาออกเป็น 3 ส่วนหลัก คือ

- การออกแบบ (Design Phase)
- การพัฒนาโมบายล์แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ (Mobile Application Development)
- การพัฒนาระบบหลังบ้าน (Backend Development)

1.2. วัตถุประสงค์ของงานวิจัย

- เพื่อสร้างแอปพลิเคชันประกันภัยบนระบบปฏิบัติการแอนดรอยด์ให้มีประสิทธิภาพตามความต้องการของลูกค้าบริษัทประกันภัย
- เพื่อความสะดวกสบายในการใช้บริการตามวัตถุประสงค์ของบริษัทประกันภัย
 - การซื้อกรมธรรม์
 - การใช้งานประกันภัย
 - การแจ้งเคลม
 - ค้นหาสถานที่

- เพื่อเพิ่มทางเลือกในการให้บริการกับลูกค้าบริษัทประกันภัย
 - บัตรกรมธรรม์จำลอง
 - ซื้อประกันออนไลน์ผ่านแอปพลิเคชัน
 - แจ้งเคลมแบบเร่งด่วนระบุตำแหน่งด้วย จีพีเอส (GPS)
- เพื่อสร้างรูปแบบใหม่ๆ ในการให้บริการประกันภัย
- เพื่อเพิ่มทักษะด้านความรู้ ความสามารถและประสบการณ์ใหม่ๆ จากการปฏิบัติงานจริง
- เพื่อพัฒนาศักยภาพด้านทัศนคติ ความรับผิดชอบและการทำงานร่วมกับผู้อื่นในองค์กร
- เพื่อนำทักษะและศักยภาพที่ได้รับการพัฒนามาประยุกต์ใช้ในการทำงานและศึกษาต่อในอนาคต

1.3. ขอบเขตของการวิจัย (ส่วนการออกแบบ)

1.3.1. เก็บรวบรวมความต้องการของลูกค้าเพื่อระดมความคิด

- เข้าประชุมเพื่อรับฟังความต้องการของลูกค้าเกี่ยวกับแอปพลิเคชันและจัดบันทึก
- สรุปความต้องการของลูกค้าในที่ประชุม พร้อมทั้งระดมความคิดของทุกคนในทีม

1.3.2. สร้างพีเจอร์ของแอปพลิเคชันโดยการสร้างแบบร่าง (Wireframe)

- วาดพีเจอร์ของแอปพลิเคชันลงบนกระดาษ
- สร้างแบบร่างของแอปพลิเคชันลงในโปรแกรม
- นำเสนอแบบร่างในที่ประชุมและแก้ไขแบบร่างตามคำแนะนำ
- นำเสนอแบบร่างให้ลูกค้าดู
- แก้ไขแบบร่างตามคำแนะนำของลูกค้า พร้อมทั้งปรึกษากับนักพัฒนาเกี่ยวกับความเป็นไปได้ในการสร้างพีเจอร์นั้น ๆ

1.3.3. พัฒนาแบบร่างให้เกิดเป็นแบบจำลองแอปพลิเคชัน (Mockup)

- เพิ่มข้อมูลจริง สี รูปภาพ ให้กับแบบร่างเพื่อทำให้เกิดแบบจำลองของแอปพลิเคชัน
- นำเสนอแบบจำลองในที่ประชุมและแก้ไขแบบจำลองตามคำแนะนำ
- นำเสนอแบบจำลองให้ลูกค้าดู
- แก้ไขแบบจำลองตามคำแนะนำของลูกค้า

1.3.4. ทดสอบประสบการณ์การใช้งานแอปพลิเคชันโดยการสร้างโปรโตไทป์ (Prototype)

- สร้างโปรโตไทป์จากแบบจำลองแอปพลิเคชัน
- สร้างเอกสารทดสอบผู้ใช้งาน
- นำโปรโตไทป์ไปทดสอบการใช้งานแอปพลิเคชันกับผู้ใช้งานจริง
- รับฟังคำแนะนำของลูกค้า แก้ไข และสร้างเป็นไฟนอลดีไซน์
- อัปเดตไฟล์งานที่ออกแบบให้กับนักพัฒนา

1.4. ขอบเขตของการวิจัย (ส่วนการพัฒนาแอปพลิเคชัน)

- พัฒนาแอปพลิเคชันให้สามารถสมัครสมาชิกจากแอปพลิเคชัน
- พัฒนาแอปพลิเคชันให้สามารถดูข่าวสาร รายละเอียด และสิทธิพิเศษ จากแอปพลิเคชัน
- พัฒนาแอปพลิเคชันให้สามารถใช้งานบัตรจำลองแสดงกรมธรรม์ และประกันที่มีอยู่ แทนการใช้บัตรจริงจากแอปพลิเคชัน
- พัฒนาแอปพลิเคชันให้สามารถดูรายละเอียดความคุ้มครองของกรมธรรม์ จากแอปพลิเคชัน
- พัฒนาแอปพลิเคชันให้สามารถค้นหาโรงพยาบาลในเครือข่ายของบริษัทประกันหรือโรงพยาบาลอื่นๆ ศูนย์ซ่อมในเครือข่ายของบริษัทประกันและศูนย์ซ่อมอื่นๆจากแอปพลิเคชัน

1.5. ขอบเขตของการวิจัย (ส่วนระบบหลังบ้าน)

1.5.1. ระบบหลังบ้าน และส่วนติดต่อประสานตัวโปรแกรมประยุกต์ (API)

- API สำหรับสมัครสมาชิกจากแอปพลิเคชัน
- API สำหรับดูข่าวสาร รายละเอียดและสิทธิพิเศษจากแอปพลิเคชัน
- API สำหรับใช้งานบัตรจำลองแสดงกรมธรรม์และประกันที่มีอยู่แทนการใช้บัตรจริงจากแอปพลิเคชัน
- API สำหรับดูรายละเอียดความคุ้มครองของกรมธรรม์จากแอปพลิเคชัน
- API สำหรับแจ้งเคลมได้อย่างรวดเร็วผ่านแอปพลิเคชัน
- API ค้นหาโรงพยาบาลในเครือข่ายของบริษัทประกันและโรงพยาบาลอื่นๆจากแอปพลิเคชัน
- API ค้นหาศูนย์ซ่อมในเครือข่ายของบริษัทประกัน และศูนย์ซ่อมอื่นๆจากแอปพลิเคชัน
- API ค้นหาเบอร์โทรศัพท์ฉุกเฉินจากแอปพลิเคชัน

1.5.2. ระบบฐานข้อมูล

- ฐานข้อมูลเก็บรายละเอียดลูกค้า
- ฐานข้อมูลสินค้า บริการ และสิทธิประโยชน์
- ฐานข้อมูลบัตรกรมธรรม์จำลอง
- ฐานข้อมูลสถานที่
- ฐานข้อมูลการทำธุรกรรม

1.5.3. เครื่องแม่ข่าย

- ติดตั้งและปรับตั้งค่า ภาษาพีเอชพี ให้เหมาะสมต่อการพัฒนา และการทำงาน
- ติดตั้งและปรับตั้งค่า ลาลาเวลเฟรมเวิร์ค ให้เหมาะสมต่อการพัฒนา และการทำงาน
- ติดตั้งและปรับตั้งค่า อะแพชี เว็บเซิร์ฟเวอร์ (Apache Webserver) ให้เหมาะสมต่อการพัฒนาและการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6. วิธีการดำเนินงานวิจัย

1.6.1. การออกแบบ (Design Phase)

- ออกแบบลักษณะการทำงานของแอปพลิเคชันให้เหมาะกับจุดประสงค์และความต้องการของบริษัทประกันภัย
- ออกแบบภาพรวมของแอปพลิเคชัน (Wide frame)
- ออกแบบลักษณะการใช้งานให้เหมาะสมกับผู้ใช้ (User Experience : UX)
- ออกแบบเค้าโครงและหน้าต่างส่วนติดต่อกับผู้ใช้ (User Interface : UI)
- สร้างตัวต้นแบบ (Prototype) นำไปให้ผู้ทดสอบ (Tester) ทดลองใช้งานและสังเกตข้อผิดพลาดในการออกแบบและปรับแก้

1.6.2. การพัฒนาโมบายล์แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

- วาดเลย์เอาท์ที่ได้จากการออกแบบบนโปรแกรมแอนดรอยด์สตูดิโอ (Android Studio)
- สร้างส่วนติดต่อกับการทำงานร่วมกับเครื่องแม่ข่ายผ่านเอพีไอแบบ RESTful

1.6.3. การพัฒนาระบบหลังบ้าน (Backend Development)

- วิเคราะห์ความต้องการเอพีไอในการใช้ติดต่อสื่อสารระหว่างเครื่องแม่ข่ายกับแอปพลิเคชัน
- สร้างความพร้อมบนเครื่องเซิร์ฟเวอร์ระบบปฏิบัติการ ลินุกซ์ (Linux)
- สร้างโครงสร้างระบบหลังบ้านและเอพีไอด้วยภาษาพีเอชพี (PHP) และลาตลาเวลเฟรมเวิร์ค (Laravel Framework)
- พัฒนาระบบโดยใช้สถาปัตยกรรมออกแบบแบบ โมเดล วิว คอนโทรลเลอร์ (Model View Controller : MVC)
- ทดสอบและปรับปรุงประสิทธิภาพการใช้งานด้วยโปรแกรมโพสต์แมน (Postman)

1.7. อุปกรณ์สำหรับการดำเนินงานวิจัย

1. ฮาร์ดแวร์ และอุปกรณ์

- เครื่องแมคบุ๊ก (MacBook) ที่สามารถเชื่อมต่ออินเทอร์เน็ตได้
- เครื่องคอมพิวเตอร์ที่สามารถเชื่อมต่ออินเทอร์เน็ตได้
- โทรศัพท์มือถือที่ใช้งานระบบปฏิบัติการแอนดรอยด์
- เครื่องเซิร์ฟเวอร์ระบบปฏิบัติการลินุกซ์

2. ซอฟต์แวร์โปรแกรม

- โปรแกรม Sketch
- โปรแกรม Adobe Photoshop CC
- โปรแกรม Adobe Illustrator CC

- โปรแกรม Flinto
- โปรแกรม Zeplin
- โปรแกรม Andriod Studio
- โปรแกรม Eclipse for PHP
- โปรแกรม Putty
- โปรแกรม Xampp
- โปรแกรม Postman
- โปรแกรม Postgresql

3. ซอฟต์แวร์อื่น ๆ

- Source Tree (Git Client)
- Bitbucket (Git Server)
- Laravel Framework

1.8. ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับจากการฝึกงานสหกิจกับบริษัท ออกตาคอน อินเทอร์เน็ตทีฟ

1. ประโยชน์ต่อบริษัท

- เพิ่มรูปแบบแพลตฟอร์มซอฟต์แวร์การให้บริการด้านประกัน ซึ่งสามารถนำไปประยุกต์ใช้กับงานอื่น ๆ ที่มีลักษณะใกล้เคียงกันได้ในอนาคต
- ศึกษารูปแบบการทำงานของเครื่องมือและซอฟต์แวร์ใหม่ๆ เช่น โปรแกรมสเก็ต (Sketch) และโปรแกรมฟลินโต้ (Flinto) ลาลาเวลเฟรมเวิร์ค (Laravel Framework)

2. ประโยชน์ต่อผู้วิจัย

- ได้เรียนรู้วัฒนธรรมองค์กร ฝึกการปรับตัวให้เข้ากับสังคมภายในองค์กรและการทำงานร่วมกันเป็นทีมภายในองค์กร
- ได้เรียนรู้แนวคิดใหม่ๆ ในการพัฒนาโมบายล์แอปพลิเคชัน
- ได้เรียนรู้วิธีการพัฒนาซอฟต์แวร์ในรูปแบบของไจล์ (Agile)
- ได้รับความรู้ในการใช้งานซอฟต์แวร์ต่างๆ มากมาย เช่น Git, Sketch, Flinto, Zeplin Android Studio, Putty, Postman
- ได้รับความรู้ และทักษะในการออกแบบแอปพลิเคชันให้ตรงต่อความต้องการของลูกค้าและง่ายต่อนักพัฒนา
- ได้รับความรู้ และทักษะในการพัฒนาแอปพลิเคชันและระบบหลังบ้านให้มีประสิทธิภาพสูงสุด
- ได้เรียนรู้การสื่อสารกันระหว่างนักออกแบบและนักพัฒนา
- ได้พบปะพูดคุยกับลูกค้าที่เป็นบริษัทและองค์กรขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ได้ฝึกวินัยและความรับผิดชอบในหน้าที่ของตนเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

2.1. ทฤษฎีกระบวนการคิดเชิงออกแบบ (Design Thinking)

2.1.1. นิยามของคำว่าออกแบบ (Design)

การออกแบบ (Design) ในความหมายที่คนทั่วไปเข้าใจคือการสร้างสิ่งใหม่ๆ การรู้จักวางแผนกำหนดขั้นตอน รู้จักเลือกใช้วัสดุและวิธีการเพื่อทำตามที่ต้องการนั้น โดยให้สอดคล้องกับลักษณะรูปแบบ และคุณสมบัติของวัสดุแต่ละชนิด ตามความคิดสร้างสรรค์ และการสร้างสรรค์สิ่งใหม่ขึ้นมา เช่น การวาดรูป การสร้างบ้าน การสร้างของใช้ และสร้างเฟอร์นิเจอร์ ซึ่งจะทำโต๊ะขึ้นมาซักหนึ่งตัว จะต้องวางแผนไว้เป็นขั้นตอน โดยต้องเริ่มต้นจากการเลือกวัสดุที่จะใช้ในการทำโต๊ะนั้นว่าจะใช้วัสดุอะไรที่เหมาะสม ในการยึดต่อระหว่างจุดต่างๆนั้นควรใช้ กาว ตะปู สกรู หรือใช้ข้อต่อแบบใด รู้ถึงวัตถุประสงค์ของการนำไปใช้งาน ความแข็งแรงและการรองรับน้ำหนักของโต๊ะสามารถรองรับได้มากน้อยเพียงใด สีสนักรวใช้สีอะไรจึงจะสวยงาม เป็นต้น และในอีกหนึ่งความหมายของการออกแบบคือ การปรับปรุงแบบ ผลงานหรือสิ่งต่างๆ ที่มีอยู่แล้วให้เหมาะสม และดูดีมีความแปลกใหม่ขึ้น เพื่อตอบสนองความต้องการในสิ่งใหม่ๆของมนุษย์ เป็นศิลปะของมนุษย์เนื่องจากการสร้างค่านิยมทางความงาม และสนองคุณประโยชน์ทางกายภาพให้แก่มนุษย์ด้วย ซึ่งส่วนใหญ่เพื่อการดำรงชีวิตให้อยู่รอด และสร้างความสะดวกสบายมากยิ่งขึ้น แต่ความเข้าใจนี้ถือว่าถูกเพียงส่วนหนึ่งเท่านั้น

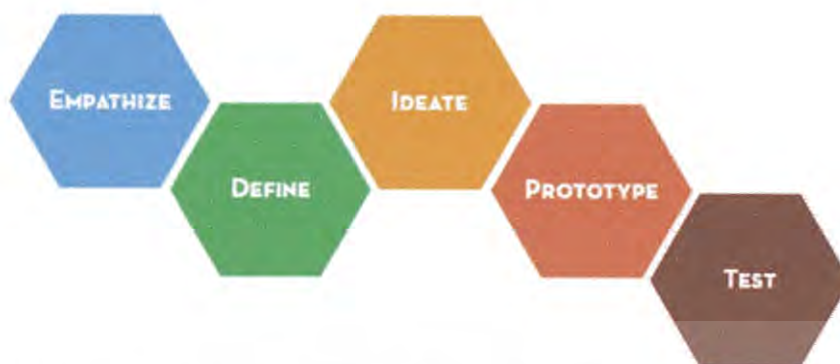
นิยามคำว่า การออกแบบ (Design) อีกหนึ่งความหมายได้ถูกกล่าวไว้โดย Simon (1968) ว่า การออกแบบ ไม่ได้เป็นแค่การสร้างสิ่งของหรือสิ่งประดิษฐ์ใหม่ๆ เท่านั้น แต่คือความพยายามที่จะเปลี่ยนแปลงสถานการณ์หรือสภาพแวดล้อมในปัจจุบันให้เป็นไปตามที่ได้คาดหวังไว้ในอนาคต หากมอย้อนไปในสมัยก่อนที่ยังไม่มีเครื่องบิน ผู้คนต่างใช้เรือหรือรถเป็นพาหนะหลักในการเดินทางในระยะที่ไกล การสร้างเครื่องบินไม่ใช่เป็นเพียงการทำให้การเดินทางได้เร็วขึ้นเท่านั้น แต่ยังเป็นการออกแบบประสบการณ์ในการเดินทางแบบใหม่ แก้ปัญหาการขนส่ง รวมไปถึงเพิ่มโอกาสต่างๆ อีกมากมายให้กับโลกใบนี้ ซึ่งก็คือความหมายของคำว่า การออกแบบ (Design) ที่แท้จริงคือการพัฒนาและสร้างการเปลี่ยนแปลงให้เกิดขึ้น

2.1.2. ความหมายของกระบวนการคิดเชิงออกแบบ (Design Thinking)

Design Thinking คือ กระบวนการคิดเชิงออกแบบที่ใช้การทำความเข้าใจในปัญหาต่างๆ อย่างลึกซึ้ง โดยคำนึงผู้ใช้เป็นศูนย์กลาง และนำเอาความคิดสร้างสรรค์ร่วมกับมุมมองคนหลายๆสายอาชีพ มาสร้างไอเดีย แนวทางการแก้ไข และนำเอาแนวทางต่างๆ นั้นมาทดสอบและพัฒนา เพื่อให้ได้แนวทางหรือนวัตกรรมที่ตอบโจทย์กับผู้ใช้และสถานการณ์นั้นๆ ซึ่ง Design Thinking ได้ถูกนำมาใช้ในองค์กรชั้นนำของโลกมากมายทั้งที่มีทั้งขนาดใหญ่และขนาดเล็ก อาทิเช่น Google, Apple, Phillips, P&G และ Airbnb เป็นต้น โดยองค์กรต่างๆเหล่านี้ ได้นำ Design Thinking มาใช้เป็นเครื่องมือหลักหลัก เพื่อสร้างนวัตกรรมต่างๆ ไม่ว่าจะเป็น Product and Service, Operational Process, Business Strategy และรวมไปถึง Business Model โดยจะขอเสนอตัวอย่าง Design Thinking ของสององค์กรให้ดูเพื่อให้เข้าใจถึงกระบวนการทำ Design Thinking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

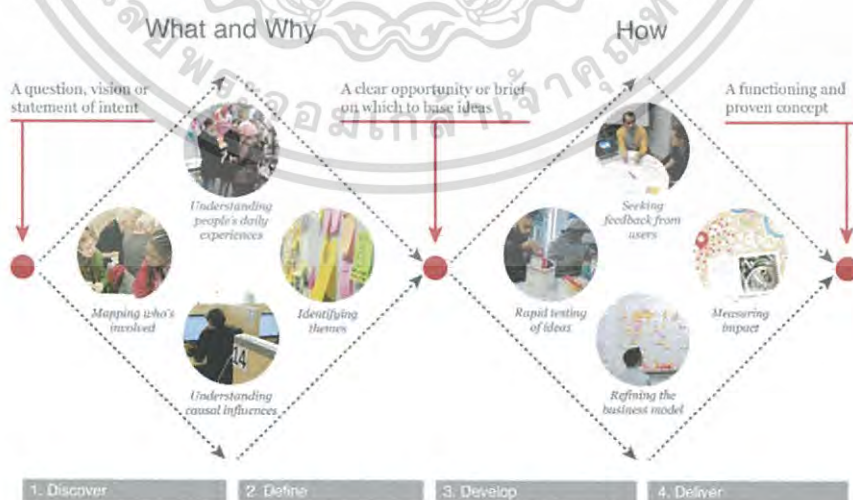
2.1.2.1. กระบวนการคิดเชิงออกแบบของ Stanford d.school



ภาพที่ 2.1 ขั้นตอนกระบวนการคิด Design thinking ของ Stanford d.school

จากภาพที่ 2.1 เป็น Design thinking ของ Stanford d.school ซึ่งได้แบ่งขั้นตอนกระบวนการคิดออกเป็น 5 ขั้นตอน ได้แก่ Empathize, Define, Ideate, Prototype, และ Test จากทั้ง 5 ขั้นตอนนี้ จะเห็นได้ว่า ขั้นตอนที่หนึ่งและสอง (Empathize และ Define) เป็นขั้นตอนทำความเข้าใจปัญหาที่ได้รับมาจากผู้ใช้และตีความปัญหานั้นอย่างลึกซึ้ง ขั้นตอนที่สาม (Ideate) คือขั้นตอนในการใช้ความคิดสร้างสรรค์และมุมมองจากหลายๆ ด้านมาสร้างเป็นไอเดียที่อยู่บนพื้นฐานความต้องการของผู้ใช้งาน และขั้นตอนที่สี่และห้า (Prototype และ Test) คือขั้นตอนในการทดสอบแนวคิดและพัฒนาต้นแบบที่เป็นตัวอย่างแนวคิดจากไอเดียที่ได้สร้างขึ้น เพื่อให้ได้แนวทางหรือนวัตกรรมที่ตอบโจทย์กับสถานการณ์หรือปัญหาที่เกิดขึ้นกับผู้ใช้งาน

2.1.2.2. กระบวนการคิดเชิงออกแบบของ UK Design Council



ภาพที่ 2.2 ขั้นตอนกระบวนการคิด Design thinking ของ UK Design Council

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Design Thinking หรือ The Double Diamond Design Process ของ UK Design Council นั้นแบ่งขั้นตอนออกเป็น 4 ขั้นตอนด้วยกัน ได้แก่ Discover Define Develop และ Deliver จะเห็นได้ว่า กระบวนการคิดเชิงออกแบบของ UK Design Council จะคล้ายกันกับแนวคิดของ d.school อยู่มาก ก็คือ ขั้นตอนหนึ่งและสอง (Discover และ Define) เป็นขั้นตอนทำความเข้าใจปัญหาที่ได้รับมาจากผู้ใช้และตีความปัญหานั้นอย่างลึกซึ้ง รวมถึงการวางแผนโครงการ ขั้นตอนสาม (Develop) คือขั้นตอนในการใช้ความคิดสร้างสรรค์และมุมมองจากหลากหลายด้านมาสร้างไอเดียที่หลากหลายและพัฒนาให้เกิดเป็นภาพเพื่อทดสอบไอเดียต่างๆ และขั้นตอนสี่ (Deliver) คือขั้นตอนในการทดสอบช่วงสุดท้ายก่อนที่จะนำเอานวัตกรรมหรือไอเดียนั้นออกสู่ตลาด หรือนำเอาไปใช้ประโยชน์จริง

2.1.3. สรุปขั้นตอนง่ายๆในการทำกระบวนการคิดเชิงออกแบบ (Design Thinking)

กระบวนการ Design Thinking หรือ กระบวนการคิดเชิงออกแบบ เป็นกระบวนการคิดสร้างสรรค์นวัตกรรมอย่างเป็นระบบ Design Thinking ไม่ใช่เรื่องใหม่ในปัจจุบัน แต่เป็นกระบวนการคิดสร้างสรรค์ที่ใช้กันมานานในสายงานออกแบบผลิตภัณฑ์ หรือในสายงานสถาปัตยกรรม Design Thinking มีบทบาทมากขึ้นเรื่อยๆในธุรกิจทุกวันนี้เพราะถูกนำมาปรับใช้ในการแก้ปัญหาและสร้างสรรค์นวัตกรรมผ่านผลิตภัณฑ์และบริการใหม่ๆ ซึ่งวิธีการเรียนรู้ Design Thinking ที่ดีที่สุด คือ การทดลองนำ Design Thinking ไปใช้ในการทำงานจริง ซึ่งสามารถสรุปแนวคิดได้เป็น 3 ขั้นตอนหลักๆ

1.) เข้าใจปัญหาให้ถูกต้อง (Understand) คือ

การใช้เวลาทำความเข้าใจปัญหาอย่างลึกซึ้ง มีบทบาทมากต่อการกำหนดทิศทางในการแก้ปัญหา การเข้าใจปัญหาที่ถูกต้องและตั้งคำถามที่ถูกต้องเท่านั้นที่จะนำไปสู่การแก้ปัญหาที่ตรงประเด็น จริงๆแล้ว Design Thinking มีชื่อเรียกอีกชื่อคือ “Human-Centered Design” หรือการออกแบบโดยการเอา “คน” ที่ต้องการจะแก้ปัญหามาเป็นศูนย์กลาง ตัวอย่างเช่น หากครูต้องการพัฒนาการเรียนการสอนของตัวเอง สิ่งแรกที่ควรทำคือการทำความเข้าใจคือนักเรียน ถ้าเป็นหมอหรือพยาบาลที่ต้องการสร้างสรรค์บริการที่ดี ก็ต้องเริ่มต้นที่การทำความเข้าใจความต้องการของคนไข้เป็นสิ่งแรก เช่นเดียวกับการทำธุรกิจ หากจะทำธุรกิจใด ก็ต้องเริ่มต้นที่การทำความเข้าใจกับปัญหาที่ลูกค้ามีอย่างลึกซึ้ง เพราะถ้าหากพัฒนาสร้างสรรค์ผลิตภัณฑ์และบริการใหม่ออกแล้วไม่สามารถหารายได้จากผลิตภัณฑ์และบริการได้ นั่นเป็นเพราะการสร้างสรรค์ผลิตภัณฑ์และบริการใหม่จากสมมุติฐาน ไม่ได้สร้างจากความต้องการที่แท้จริงของลูกค้า

2.) คิดแบบไม่มีกรอบ (Brainstorm) คือ

การสร้างไอเดีย หรือการต่อยอดไอเดียจากหลากหลายมุมมองของคนภายในทีม เพื่อสร้างสรรค์ไอเดีย นวัตกรรมที่แปลกใหม่และตอบโจทย์การแก้ไขปัญหาต่างๆ อย่างถูกต้อง ถึงแม้จะเข้าใจปัญหาอย่างถูกต้องแล้ว อีกอุปสรรคสำคัญที่ทำให้เกิดนวัตกรรมใหม่ๆ ไม่เกิดคือ กรอบความคิดของแต่ละคน บางครั้งไอเดียใหม่ๆ ที่แปลกหูแปลกตาจะมาพร้อมกับความเสี่ยงต่อการโดนประเมินหรือโดนตัดออก ดังนั้น กระบวนการ Design Thinking จึงเน้นการแยกกระบวนการสร้างสรรค์ไอเดีย (Idea Generation) ออกจากการประเมิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Idea Evaluation) คือ คิดไอเดียออกมาให้มากพอก่อน แล้วจึงมาประเมินว่าไอเดียไหน เป็นไปได้มากน้อยอย่างไร ตรงตามความต้องการของลูกค้าหรือไม่ รวมไปถึงสามารถพัฒนา ผลิตภัณฑ์ออกมาได้จริงหรือไม่ เพราะบ่อยครั้งไอเดียที่แปลกใหม่ก็มีโอกาสที่จะไม่สามารถ นำมาสร้างหรือพัฒนาเป็นผลิตภัณฑ์ได้จริง

3.) เรียนรู้ผ่านการทดลองลงมือทำ (Prototype) คือ

การพัฒนาไอเดีย สร้างต้นแบบ และนำไปทดลองกับกลุ่มเป้าหมาย รับคำแนะนำ และความคิดเห็น เพื่อนำไปพัฒนาต้นแบบ จนกระทั่งคนภายในทีมและกลุ่มเป้าหมายพึงพอใจ แล้วจึงนำเอานวัตกรรมนั้นไปใช้จริง เพราะเสน่ห์ที่สำคัญอีกข้อของกระบวนการ Design Thinking คือการเปลี่ยนไอเดียดีให้เป็นรูปเป็นร่างที่สามารถจับต้องจริงได้ ด้วยการ สร้างต้นแบบหรือแบบจำลองง่ายๆที่สื่อสารแนวคิด เพราะการสร้างต้นแบบเพื่อนำไปทดลอง ใช้จะทำให้เกิดการเรียนรู้ถึงข้อดีและข้อเสียของไอเดีย ทำให้เห็นชัดเจนขึ้นว่าไอเดียที่คิด ขึ้นมานั้นจะตอบโจทย์กลุ่มเป้าหมายหรือไม่ กระบวนการ Design Thinking ไม่ใช่ศาสตร์ที่ ซับซ้อน และสามารถนำมาปรับใช้ได้หลากหลาย เช่น

- Design Thinking ในการออกแบบบริการ: การออกแบบประสบการณ์การ บริการ ที่เกิดจากความเข้าใจความต้องการที่แท้จริงของลูกค้า
- Design Thinking ในการพัฒนาสังคม: องค์กรประชาสังคมมากมายนำ Design Thinking ไปใช้ในการออกแบบวิธีการแก้ปัญหาสังคม
- Design Thinking กับการใช้ชีวิต: สามารถประยุกต์ใช้ Design Thinking กับ ตัวเองได้ เพราะที่มหาวิทยาลัยสแตนฟอร์ด (Stanford d. school) ได้นำ Design Thinking มาปรับใช้กับการ ออกแบบชีวิตผ่านวิชา Design Happiness

โดยสรุปแล้ว กระบวนการ Design Thinking คือ การปรับเปลี่ยนมุมมองความคิด พิจารณา “ปัญหา” ให้เป็น “โอกาสในการแก้ปัญหา”

2.2. อจิลล์ (Agile) แนวคิดในการพัฒนาผลิตภัณฑ์

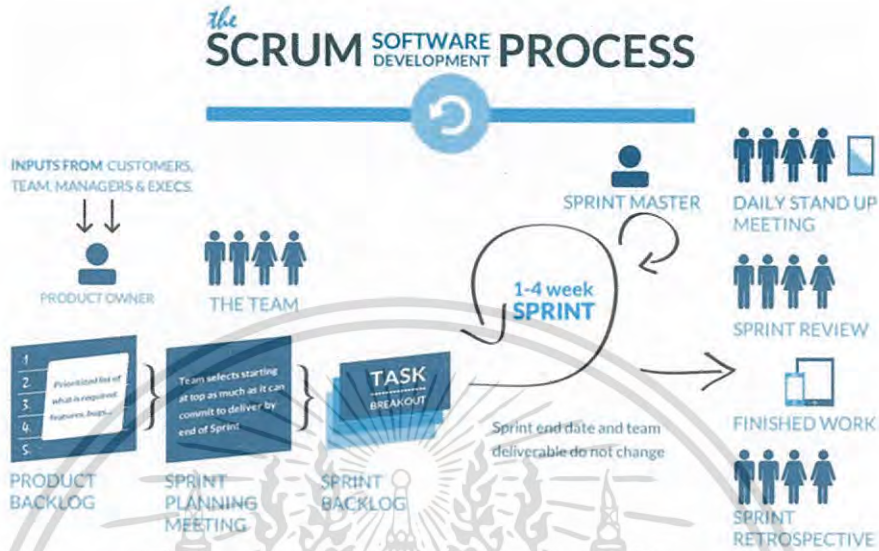
2.2.1. ประวัติความเป็นมาของอจิลล์ (Agile)

แนวคิดของอจิลล์ (Agile) นั้นได้เกิดขึ้นมาตั้งแต่ช่วงปี 1970 แล้ว จากการวิจัยได้ พบว่า แนวคิดได้เกิดขึ้น มาเพราะการพัฒนาซอฟต์แวร์ในรูปแบบเดิม (traditional method) นั้น มี ข้อเสียมากมาย โดยเฉพาะเรื่องความพึงพอใจต่อลูกค้า เวลาในการพัฒนา และงบประมาณ ดังนั้น นักพัฒนาซอฟต์แวร์จึงได้แนะนำให้ เปลี่ยนไปใช้อจิลล์ ซึ่งมีแนวคิดดังกล่าวนี้ ตอบสนองต่อการ เปลี่ยนแปลงความต้องการของลูกค้า การทำงานร่วมกับลูกค้า

อจิลล์ (Agile) คือกลุ่มของผู้สร้างแนวคิดการพัฒนาซอฟต์แวร์ (Software methodologies) ที่มีแนวคิดเดียวกัน มีลักษณะการพัฒนาแบบการทำซ้ำ (iteration) คือ แบ่ง ส่วนย่อยๆ แล้วพัฒนาต่อเรื่อยๆ แต่ละกระบวนการพัฒนาซอฟต์แวร์แบบทำซ้ำ (iteration) จะต้อง เกิดซอฟต์แวร์ที่ใช้งานได้จริงในแต่ละช่วงของการพัฒนา (working software) และต้องพัฒนาต่อ เรื่อยๆ มีลักษณะให้ความสำคัญกับคนที่ทำมากกว่า มีการส่งมอบซอฟต์แวร์ที่นำไปใช้งานได้จริง

มากกว่าการทำเอกสารที่ครบสมบูรณ์ มีการเจอลูกค้าในการพัฒนาตลอดเวลา มากกว่าการทำตามสัญญา และต้องยอมรับการเปลี่ยนแปลง มากกว่าการทำตามแผนที่ได้วางไว้

2.2.2. ประวัติความเป็นมาของอไจล์ (Agile)



ภาพที่ 2.3 Scrum Software Development Process

อไจล์ (Agile) นั้นมีหลายวิธีการ (method) ให้เลือกใช้ในการพัฒนามากมาย สำหรับวิธีการที่ได้รับความนิยมมากที่สุดคือ สกรัม (Scrum) วิธีการพัฒนาผลิตภัณฑ์แบบอไจล์ที่ชื่อว่า สกรัม (Scrum) คือ รูปแบบการทำงาน (framework) ของอไจล์ที่มีไว้สำหรับการพัฒนาซอฟต์แวร์ชนิดหนึ่ง โดยกลุ่มของคนที่ทำให้อซอฟต์แวร์ (software) เกิดเป็นผลิตภัณฑ์ (product) ไม่จำเป็นต้องเป็นหน้าที่ใครคนใดคนหนึ่ง แต่สามารถทำหน้าที่นั้นได้ทุกคน ซึ่ง Scrum มีกระบวนการพัฒนาโดยย่อคือ

- 1.) Product backlog คือ การทำบอร์ดที่รวบรวม requirement ทั้งหมดของลูกค้าทั้งหมด
- 2.) Sprint backlog คือ การทำบอร์ดที่รวบรวมงาน (task) ของการพัฒนาในแต่ละช่วง (iteration) นั้นๆ ว่าต้องทำอะไรบ้าง
- 3.) Sprint คือ การทำลงมือพัฒนาในช่วงนั้นๆ (iteration development) เพื่อทำให้เกิดซอฟต์แวร์ที่สามารถใช้งานจริงได้ในเบื้องต้น (working software) ขึ้นมาเป็นช่วงๆ
- 4.) Stand-up meeting เป็นการประชุมหรือพูดคุยกันถึงปัญหาและความคืบหน้าของการพัฒนา โดยหน้า Sprint backlog จะอธิบายว่า ใครทำอะไรบ้าง และทำไปถึงไหนแล้ว ความถี่ของการทำ Stand-up meeting อาจทำทุกวัน หรือ ทุกสัปดาห์ ขึ้นอยู่กับองค์กร
- 5.) ควรจะทำ Sprint ทุกสัปดาห์ หรือ อาจน้อยกว่านี้ได้ตามความเหมาะสม จนเกิดซอฟต์แวร์ที่สามารถใช้งานจริงได้ในแต่ละช่วง working software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3. ข้อดีและข้อเสียของอไจล์ (Agile)

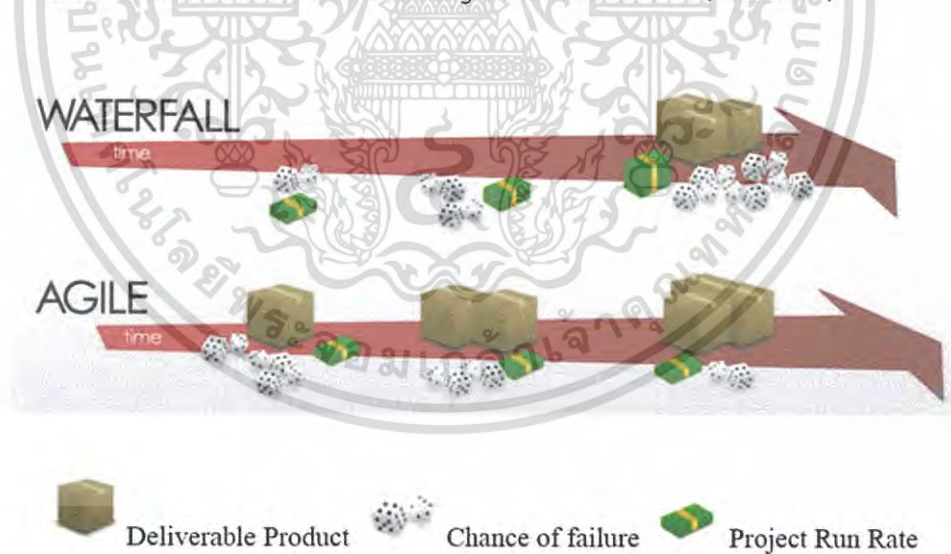
ข้อดีคือ

- รองรับการเปลี่ยนแปลงความต้องการของลูกค้า (Requirement) ได้ดี
- สนุกกับการพัฒนาซอฟต์แวร์ (software)
- มีความยืดหยุ่นในการพัฒนาฟีเจอร์ (Flexible feature)
- ลูกค้าเห็นผลิตภัณฑ์ (Product) ได้ไว
- มีความเสี่ยงน้อย โอกาสที่โปรเจกต์จะสำเร็จสูง (Project success)
- มีการเกิดผลิตภัณฑ์ที่มากกว่าการทำกระบวนการพัฒนาแบบวอเตอร์ฟอล (Waterfall)

ข้อเสียคือ

- ทุกคนในทีมเท่าเทียมกัน ไม่มีผู้นำที่บริหารจัดการทีม แต่ทุกคนในทีมต้องช่วยกันซึ่งอาจเป็นข้อเสียสำหรับองค์กรที่ต้องการผู้นำ
- ไม่เหมาะกับงานหน่วยงานราชการ ที่เน้นทำเอกสารไว้ก่อน
- ในการสร้างผลิตภัณฑ์จะต้องฟังความเห็นของทุกคนในทีม ซึ่งบางครั้งสิ่งที่ทุกคนในทีมเห็นด้วยเป็นส่วนมาก อาจไม่ตรงตามความต้องการของลูกค้า หรือไม่สามารถจะพัฒนาได้จริงเสมอ

2.2.4. ความแตกต่างระหว่างอไจล์ (Agile) กับวอเตอร์ฟอล (Waterfall)



ภาพที่ 2.4 ความแตกต่างระหว่าง Agile กับ Waterfall

จากรูปที่ 2.4 จะเห็นความแตกต่างได้ว่า วอเตอร์ฟอล (Waterfall) ตลอดระยะเวลาการพัฒนาจะไม่มีผลิตภัณฑ์ (product) ส่งออกมาให้ลูกค้าได้เห็นเลย ซึ่งลูกค้าจะสามารถเห็นผลิตภัณฑ์ (product) จริงๆ ได้ เมื่อถึงระยะช่วงสุดท้ายของกระบวนการแล้ว ในขณะที่ Agile ส่งซอฟต์แวร์ที่ใช้งานได้จริง (working software) ให้ดูเป็นระยะๆ ว่า นี่คือนี่ที่ลูกค้าต้องการหรือไม่ เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลูกค้าจะสามารถเปลี่ยนความต้องการ (requirement) ได้ทันที ดั่งนั้นอไจล์ (Agile) จึงเป็นกระบวนการพัฒนาซอฟต์แวร์ที่รองรับการเปลี่ยนแปลงความต้องการของลูกค้า (requirement) ได้ดี

วอเตอร์ฟอล (Waterfall) มีค่าใช้จ่ายในการพัฒนาโครงการ (project) เพิ่มขึ้นตลอด เพราะความต้องการของลูกค้า (requirement) เปลี่ยนแปลงอยู่ตลอดเวลา หากความต้องการ (requirement) ถูกเปลี่ยนในขณะที่อยู่ในกระบวนการเขียนโปรแกรม (phase coding) หรือกระบวนการออกแบบ (design) จะมีค่าใช้จ่ายเพิ่มขึ้นกว่าช่วงแรกๆ ยิ่งถ้าผลิตภัณฑ์ (product) นั้นเสร็จแล้ว มีการเปลี่ยนแปลงความต้องการของลูกค้า (requirement) เพิ่มเติมในภายหลัง จะทำให้ค่าใช้จ่ายสูงมาก ในขณะที่อไจล์ (Agile) จะกำหนดค่าใช้จ่ายในการรันโครงการ (project) ไว้คงที่ตลอดการพัฒนาผลิตภัณฑ์ (product)

วอเตอร์ฟอล (Waterfall) มีความเสี่ยงในการพัฒนาระยะสุดท้ายมากเพราะมีโอกาสที่ผลิตภัณฑ์ (product) ออกมาแล้วไม่ใช่สิ่งที่ลูกค้าต้องการ ทำให้มีโอกาสที่โครงการจะล้มเหลว (project fail) ได้สูงพอสมควร ในขณะที่อไจล์ (Agile) รับความเสี่ยงในเรื่องนี้ตั้งแต่ต้น เพราะการส่งมอบผลิตภัณฑ์ (product) ครั้งแรกอาจไม่ใช่สิ่งที่ลูกค้าต้องการ ดังนั้นจึงปรับเปลี่ยนตลอดการพัฒนาโครงการ จนความเสี่ยงลดลง

2.3. ทฤษฎีเอ็มวีพี (MVP)

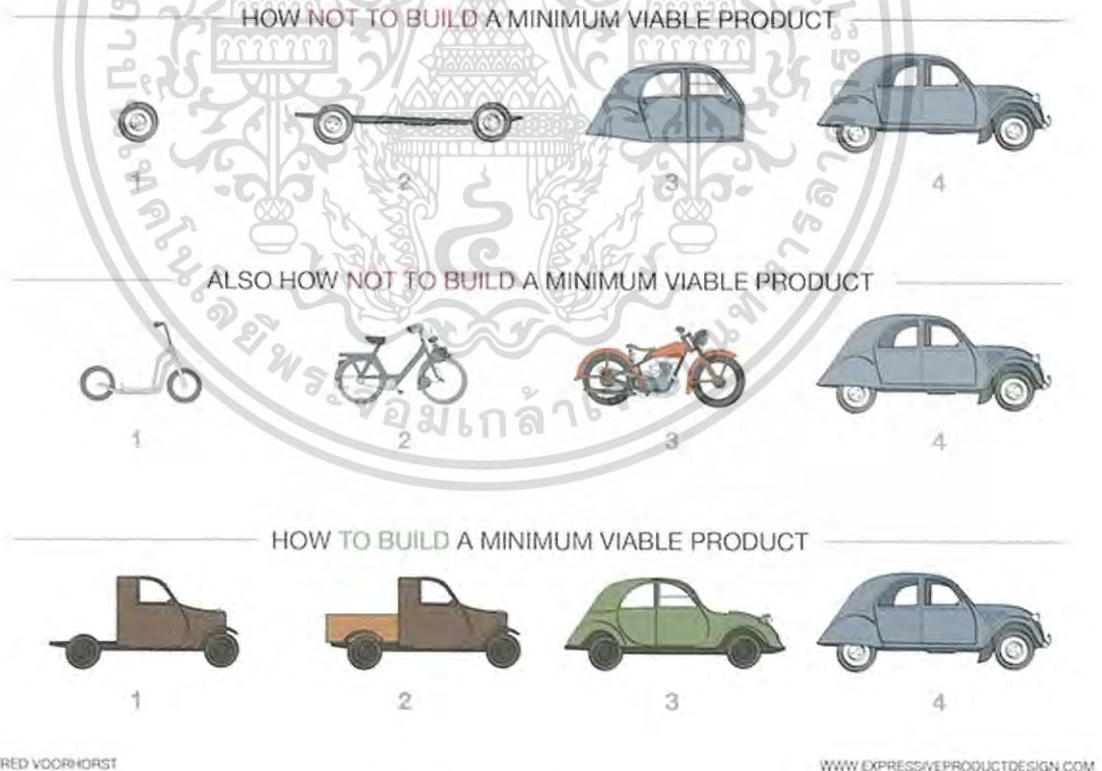
2.3.1. เอ็มวีพี (MVP) คืออะไร



ภาพที่ 2.5 หนังสืออีบุ๊ก Minimum Viable Product จากเว็บไซต์ UXPin

MVP ย่อมาจาก Minimum Viable Product ซึ่งความหมายของ MVP คือการสร้างผลิตภัณฑ์ (product) ที่มีฟีเจอร์ (feature) ของผลิตภัณฑ์เพียงพอสำหรับการทดสอบดูว่าผลิตภัณฑ์นี้สามารถทำงานได้ในตลาด เพื่อให้ในการสร้างผลิตภัณฑ์บรรลุเป้าหมายนี้ จะยังไม่ใส่ฟีเจอร์ (feature) ที่ยังไม่จำเป็นให้กับผลิตภัณฑ์และใส่เฉพาะฟีเจอร์ (feature) ที่ถือว่าเป็นหัวใจหลักของผลิตภัณฑ์ (product) หรือ แอปพลิเคชัน (application) เท่านั้น โดยสรุป MVP คือ ชุดฟีเจอร์จำนวนน้อยที่สุด (a set of minimum features) ที่เพียงพอให้ซอฟต์แวร์สามารถใช้งานได้ เพื่อที่จะนำซอฟต์แวร์นั้นออกสู่ตลาดได้อย่างรวดเร็วที่สุด ซึ่งคำนิยามเกี่ยวกับ MVP ที่ได้รับการยอมรับมากที่สุด นิยามโดย Steve Blank และ Eric Ries โดย MVP คือ ฟีเจอร์อย่างน้อยที่สุดของผลิตภัณฑ์ที่สามารถใช้มันประเมินตลาดได้

- Minimum: หมายถึงแอปพลิเคชัน (application) ที่มีแต่ฟีเจอร์ (feature) ที่เป็นหัวใจหลักเท่านั้น และทุกสิ่งที่ยังไม่จำเป็นจะต้องไม่นำไปใส่ในผลิตภัณฑ์
- Viable: หมายถึงผลิตภัณฑ์ (product) ที่มีโอกาสดีให้ผู้ใช้เข้ามาใช้งาน นั่นคือการสร้างคุณค่าให้กับผู้ใช้งาน ความหมายกว้างๆ ของมูลค่า ตัวอย่างเช่น เกมสกีให้ความบันเทิง ซึ่งเป็นคุณค่า โดยปกติแล้วจะพิจารณาผลิตภัณฑ์ (product) นี้ ถ้ามันสร้างรายได้มากพอที่จะคุ้มค่าพอที่จะครอบคลุมค่าใช้จ่ายในการพัฒนาผลิตภัณฑ์ (product) นี้
- Product: หมายถึงผลิตภัณฑ์ที่ผลิตสินค้าดิจิทัล (digital) ให้ผู้คนได้ใช้งาน



ภาพที่ 2.6 How to Build a MVP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 2.6 ถ้าเป้าหมาย คือ การสร้างรถยนต์เพื่อช่วยผู้คนที่เดินทาง สะดวกสบายขึ้น ในรูปแบบสุด ถ้าหากเริ่มทำล้อก่อน จะสามารถนำไปใช้ประเมินตลาดได้หรือไม่ อะไร คือคุณค่าของล้อที่จะช่วยแก้ปัญหาให้กับผู้คนที่เดินทางได้สะดวกสบายขึ้น และกว่าจะเป็นรถยนต์ทั้ง คันค่อนข้างจะใช้เวลานานมาก สุดท้ายถ้าหากลงทุน ลงแรงและเวลาไปตั้งมาก แต่ไม่รู้ว่าจะมีคน ต้องการมันจริงหรือไม่ รูปทรงกลาง เริ่มจากสร้างรถจักรยานแล้วผลิตภัณฑ์สุดท้ายคือรถยนต์ก็ไม่ใช่ หลักการของ MVP เพราะมีการเปลี่ยนโครงสร้างของผลิตภัณฑ์มากเกินไป ส่วนรูปร่างสุด เริ่มจาก การที่สร้างรถยนต์ขึ้นมาแบบง่ายๆ สามารถที่จะช่วยให้ผู้คนที่เดินทางสะดวกสบายขึ้นเป็นสิ่งแรก สามารถนำไปทดสอบตลาดได้เร็ว เพื่อดูว่าผลิตภัณฑ์สามารถตอบโจทย์กลุ่มเป้าหมายได้หรือไม่ ก่อนที่จะนำมาต่อยอด ปรับปรุง และทดสอบตลาดไป จนกลายเป็นรถยนต์ที่สมบูรณ์ที่สุดและตลาด ยอมรับ

2.3.2. วิธีการกำหนดเอ็มวีพี (MVP)

ก่อนที่จะสามารถเริ่มต้นพัฒนาผลิตภัณฑ์ จะต้องกำหนด MVP ขึ้นมาก่อนและ กำหนดแผนพัฒนาผลิตภัณฑ์ (product roadmap) อะไรคือฟีเจอร์ (feature) ที่ต้องมี (nice-to-haves?) ซึ่งมันเป็นเรื่องสำคัญมากที่จะมีวัตถุประสงค์ที่เป็นไปได้อยู่ในกระบวนการทำงาน บางฟีเจอร์ (feature) ที่ใส่ใจมากอาจไม่ใช่หัวใจสำคัญของผลิตภัณฑ์ที่จะสร้างขึ้น (core of the product) ดังนั้น ต้องตัดสินใจว่าอะไรคือฟีเจอร์ (feature) ที่ต้องมี (must-have) และทำการเขียนกลุ่มของฟีเจอร์ที่ ต้องการให้มีในผลิตภัณฑ์ (feature set) ซึ่งเอกสารนี้ทำให้เข้าใจภาพรวมในสิ่งที่ต้องสร้าง โดย เอกสารนี้จะเป็นตัวอย่างที่ใช้สรุปร่วมกับนักออกแบบ และนักลงทุน เป็นการวางวิสัยทัศน์ลงบน กระดาษ นอกจากนี้ยังสามารถทำความเข้าใจฟีเจอร์ (feature) ในทางเทคนิค (technical) โดยปกติ แล้วกลุ่มของฟีเจอร์ที่ต้องการให้มีในผลิตภัณฑ์ (feature set) นั้นจะรวมขอบเขตของเทคนิคที่ใช้ใน การพัฒนา (technical scope) เข้าไปด้วย ซึ่งจะมีประโยชน์มากถ้าหากทำงานร่วมกับนักพัฒนา หลายคน

ขั้นตอนต่อไปจะต้องกำหนดความสำคัญของฟีเจอร์ (feature) ว่าอะไรคือฟีเจอร์ (feature) ที่สำคัญที่สุดและเป็นสิ่งที่สร้างมูลค่าให้มากที่สุดโดยใช้เวลาน้อยที่สุด หนึ่งในสิ่งที่ต้อง กำหนดนั้น คือต้องกำหนดว่าฟีเจอร์ (feature) อะไรที่จะสร้างเป็นอันดับแรกก่อนที่จะส่งผลิตภัณฑ์ (product) ออกไปสู่ตลาด

2.3.3. ข้อดีของการทำเอ็มวีพี (MVP)

ในการทำ MVP มีข้อดีดังต่อไปนี้

- สามารถทดสอบสมมติฐาน โดยใช้ทรัพยากรน้อยที่สุด: การสร้างฟีเจอร์ (feature) น้อยที่สุดเพื่อทดสอบตลาดหรือสมมติฐาน ช่วยให้ธุรกิจไม่จำเป็นต้อง ลงทุนเงินและเวลามากมายไปกับการพัฒนาซอฟต์แวร์ออกสู่ตลาด ซึ่งมีความ เสี่ยงคืออาจไม่มีคนใช้
- เร่งการเรียนรู้: การทดสอบสมมติฐานช่วยให้ได้คำตอบเกี่ยวกับผลิตภัณฑ์ (product) ที่จะทำสามารถตอบโจทย์ความต้องการของลูกค้าหรือไม่ ซึ่งในจุดนี้ จะช่วยให้สามารถปรับปรุง ต่อยอด หรือทดสอบสมมติฐานใหม่ๆ ได้อย่าง รวดเร็ว

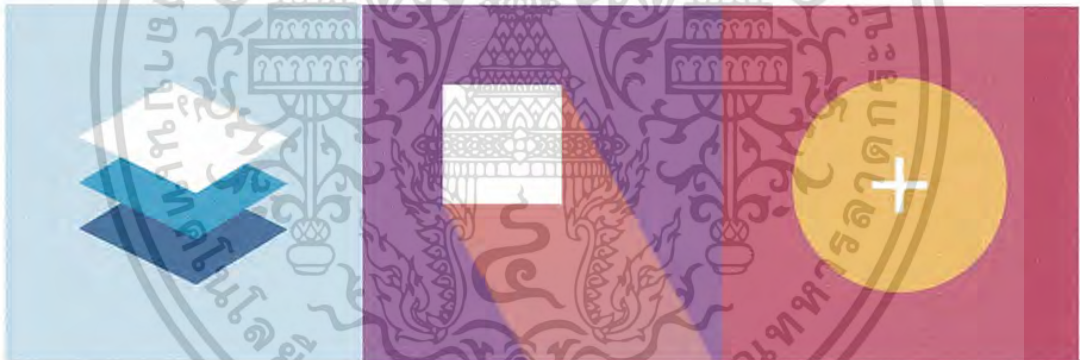
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลดการเสียเวลาที่ต้องใช้ในการพัฒนา: การลดเวลาการเขียนโปรแกรม (coding) หรือการเขียนซอฟต์แวร์ โดยมุ่งเน้นเฉพาะฟีเจอร์ (feature) ที่น้อยที่สุดซึ่งเป็นแก่นสำคัญของผลิตภัณฑ์ (product) เท่านั้น
- ส่งผลิตภัณฑ์ให้ถึงมือลูกค้าเร็วที่สุดเท่าที่จะทำได้: เหตุที่ควรส่งผลิตภัณฑ์ให้ถึงมือลูกค้าเร็วที่สุด ทั้งนี้เพื่อเก็บข้อเสนอแนะ (feedback) จากลูกค้า รวมถึงเป็นการทดสอบสมมติฐานด้วยว่า สุดท้ายแล้วลูกค้าจะยอมรับผลิตภัณฑ์ (product) ที่สร้างขึ้นหรือไม่

2.4. ทฤษฎีการออกแบบยูเอชไอ (UX/UI) ด้วยเมททีเรียลดีไซน์ (Material Design)

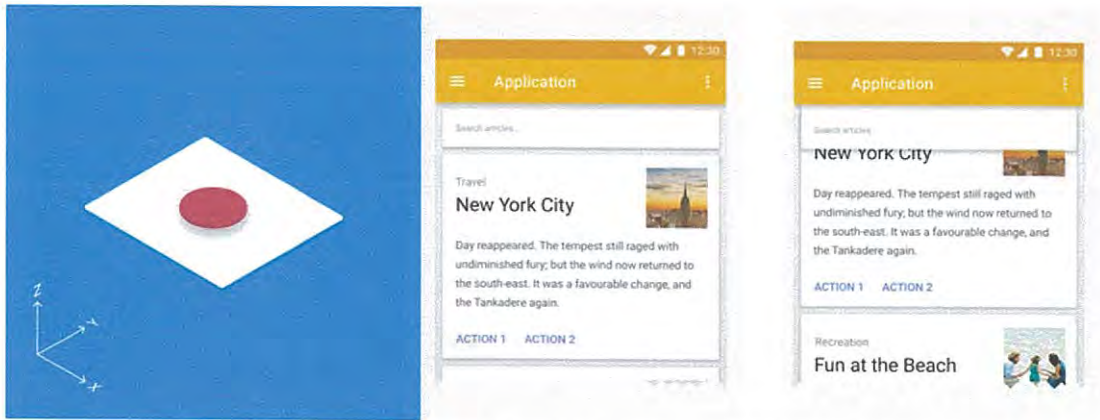
2.4.1. ทฤษฎีพื้นฐานของเมททีเรียลดีไซน์ (Material Design)

เมททีเรียลดีไซน์ (Material Design) ได้นำมาเปิดตัวในงาน Google I/O ซึ่งเป็นภาษาดีไซน์ (Design Language) ที่ถูกพัฒนาโดยกูเกิล (Google) และจะถูกนำมาใช้แทนโฮโล (Holo) โดยที่เมททีเรียลดีไซน์ มาจากแนวคิดของกระดาษและหมึกที่ไปสัมพันธ์กับความสามารถในการรับรู้โดยอัตโนมัติของมนุษย์ ซึ่งถือได้ว่าเป็นภาษาดีไซน์ (Design Language) ที่มี User Experience (UX) ดีมาก ถือเป็นการต่อยอด Flat Design ของทางฝั่ง iOS และ Modern UI ของทางฝั่ง Microsoft ได้ดีไม่แพ้กัน



ภาพที่ 2.7 ภาพแสดงตัวอย่าง Material Design

ในเมททีเรียลดีไซน์ (Material Design) เชื่อว่า ขอบ (edges) และ พื้นผิว (surfaces) เป็นสิ่งที่มีอยู่ในชีวิตจริงและมีการเรียนรู้อยู่ในสมองมนุษย์อยู่แล้ว จึงควรจะใช้ขอบเพื่อให้มนุษย์สามารถแยกแยะวัตถุได้โดยอัตโนมัติและไม่ว่าจะแสดงผลเป็นพิกเซล (pixel) 2 มิติ แต่เมททีเรียลดีไซน์ มองว่าโลกแห่งความจริงนั้นเป็น 3 มิติเสมอ มีแสงเงาและการเคลื่อนไหวตามกฎของฟิสิกส์ เพื่อสร้างการรับรู้ที่เกิดขึ้นเองโดยธรรมชาติ



ภาพที่ 2.8 ภาพแสดงความลึก แสง และเงาของ Material Design

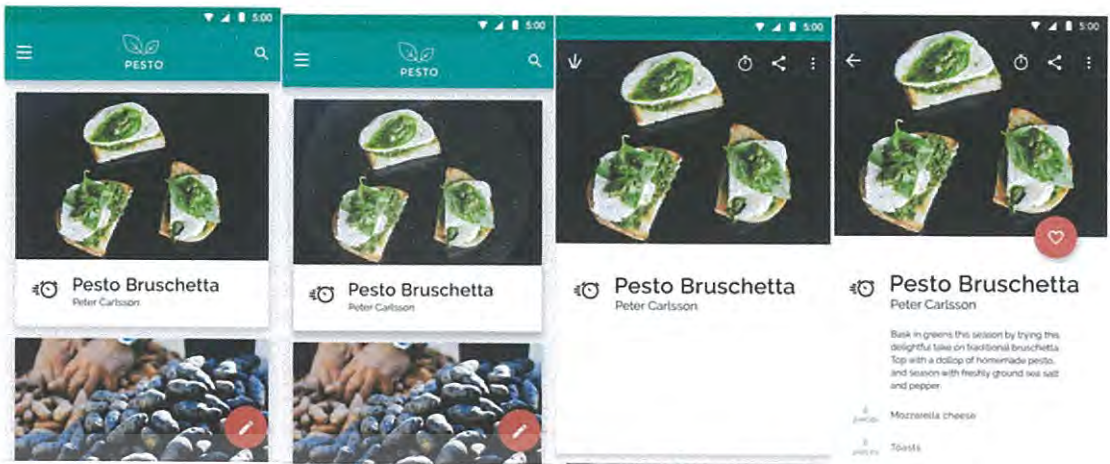
จากภาพที่ 2.8 ภาษาดีไซน์ (Design Language) ตัวนี้จึงมี ความลึก (depth) ของวัตถุ ซึ่งแสดงถึงแสงและเงา โดยพื้นฐานของแสง พื้นผิวและการเคลื่อนไหวเป็นกุญแจสำคัญในการนำพาว่าวัตถุเคลื่อนย้ายเพื่อโต้ตอบกับผู้ใช้ งานที่เหมือนจริงจะแสดงให้เห็นถึงขอบเขต พื้นที่ และแสดงให้เห็นชั้นส่วนของวัตถุที่กำลังเคลื่อนไหวอย่างชัดเจน โดยในตัว Android Lollipop ก็จะมีตัวแปร Z ขึ้นมาเพื่อกำหนดความลึกของ View นั้นๆ ได้ พร้อมเรนเดอร์เงาและระดับชั้นของวัตถุให้โดยอัตโนมัติ เพิ่มความสะดวกในการพัฒนา



ภาพที่ 2.9 ภาพแสดงความชัดเจน และการแยกแยะวัตถุของ Material Design

จะเห็นได้ว่าจากภาพที่ 2.9 ภาษาดีไซน์ (Design Language) ตัวนี้ให้ความสำคัญกับการแยกแยะวัตถุมาก หนึ่งในแนวคิดที่ถูกนำมาใช้คือ ความชัดเจน (bold) โดยวัตถุต้องไม่เนี้ยนเรียบไปกับวัตถุอันอื่นๆ แต่ต้องถูกแสดงให้เห็นอย่างชัดเจน ใช้สีที่ตัดกัน ใช้ขนาดที่ชัดเจน ไอคอนที่สื่อความหมายชัดเจน ไม่ต้องมาคิดว่าวัตถุชิ้นนี้กดได้หรือไม่ ยกเว้นปุ่มบางประเภทที่ต้องมีลักษณะเนี้ยนเรียบเพื่อไม่ให้ดึงความสนใจของผู้ใช้ในหน้านั้นมากเกินไป

148563



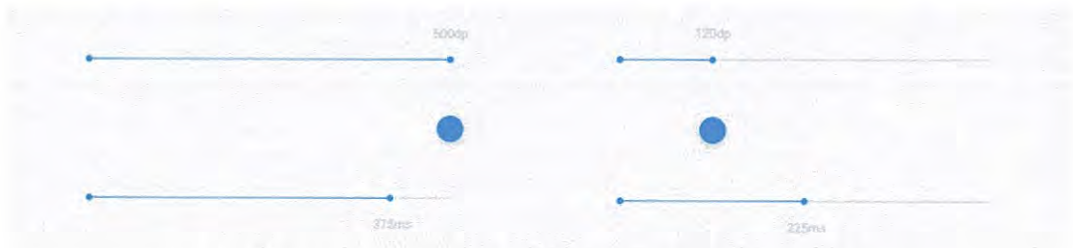
ภาพที่ 2.10 ภาพแสดงการเคลื่อนไหวและการเปลี่ยนของ Material Design

อีกหนึ่งแนวคิดของแมททีเรียลดีไซน์ (Material Design) คือ การกระทำของผู้ใช้งาน (user) เป็นใหญ่ ถ้าหากผู้ใช้งานทำอะไร ก็ควรจะมีการโต้ตอบ (Interact) จากจุดนั้นๆ เช่น ถ้ากดที่ปุ่มตรงกลางจอ ก็ควรจะมีอะไรแสดงผลออกมาจากกลางจอ ไม่ใช่แสดงผลออกมาทางขอบของจอ การเคลื่อนไหวของวัตถุ (animation) และการเปลี่ยนของวัตถุ (transition) ต้องมีความสั้นไหล การเปลี่ยนจากอีกหน้าไปยังอีกหน้าโดยตัดไปเลยไม่ใช่วิธีที่ดี ควรจะมีการเปลี่ยน (transition) ไปแบบเนียนๆ เช่น ค่อยๆ เปลี่ยนแปลงรูปร่าง ค่อยๆ ปรากฏ หรือค่อยๆ หายไป และสุดท้ายการเปลี่ยนของวัตถุ ทุกอย่างต้องมีความหมาย การขยับเร็วช้า ความเฉื่อย ความแรงของการเปลี่ยน ล้วนมีผลต่อการรับรู้ของผู้ใช้ทั้งหมด และด้วยแมททีเรียลดีไซน์นี้ กูเกิ้ลไม่ได้นำมาใช้บน Android Lollipop (version 5.0) เท่านั้น แต่ยังถูกนำไปใช้บนทุกบริการของกูเกิ้ล ไม่ว่าจะเป็นเว็บไซต์ ทีวี หรือแม้กระทั่งรถยนต์ ซึ่งเป็นมาตรฐานการออกแบบ (Unify Design) ที่ออกแบบครั้งเดียวสามารถทำไปใช้ได้บนทุกผลิตภัณฑ์

2.4.2. การเคลื่อนไหวของวัตถุในแมททีเรียลดีไซน์ (Motion)

สาเหตุที่วัตถุ (view) ต่างๆ บนแอปพลิเคชันต้องมีการเคลื่อนไหวที่สื่อความหมายก็เพื่อแสดงให้เห็นว่าแอปพลิเคชันถูกจัดระเบียบมาแล้ว เพราะการเคลื่อนไหวจะทำให้ผู้ใช้งานโฟกัสไปที่วัตถุ (view) นั้น และการเคลื่อนไหวนั้นจะโต้ตอบกับผู้ใช้ได้เองจากการกระทำของผู้ใช้กับวัตถุ (view) นั้นๆ ซึ่งการเคลื่อนไหว (motion) นี้ จะเป็นตัวช่วยลำดับความสัมพันธ์กันระหว่างวัตถุ (view) แต่ละชั้นเพราะช่วยไม่ทำให้ผู้ใช้งานเกิดความสับสนในการใช้งานแอปพลิเคชัน โดยการเคลื่อนไหวของวัตถุแบ่งออกเป็น

1.) ตอบสนองกับผู้ใช้งาน (Responsive)



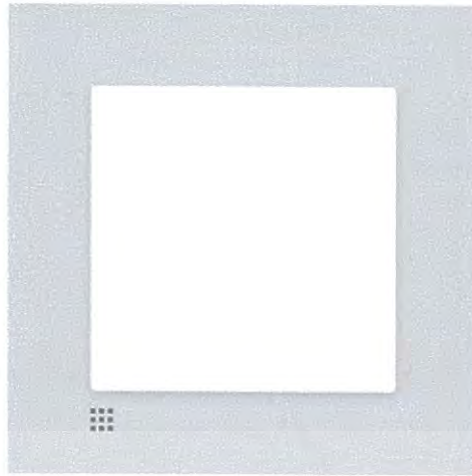
ภาพที่ 2.11 ภาพแสดงการเคลื่อนไหวในรูปแบบที่แตกต่างกัน

จากภาพที่ 2.11 แสดงลักษณะการเคลื่อนไหวแบบใหญ่ซึ่งใช้เวลาเคลื่อนไหวบนโทรศัพท์มือถือประมาณ 300 - 400 ms ส่วนลักษณะการเคลื่อนไหวแบบเล็กจะใช้เวลาสั้นประมาณ 150 - 200 ms ลักษณะการเคลื่อนไหวยาวหรือสั้นกว่านี้สามารถรู้สึกช้าหรือเร็วแตกต่างกัน



ภาพที่ 2.12 การเคลื่อนไหวแบบระลอกคลื่นหมึก (ink ripples)

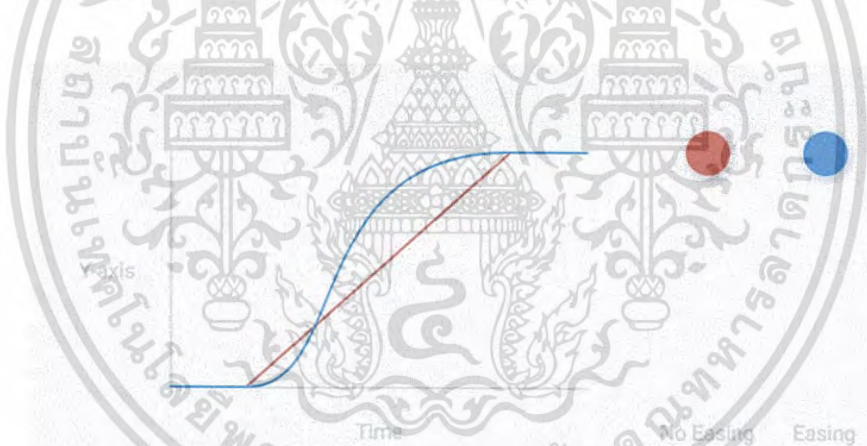
จากภาพที่ 2.12 คืออระลอกคลื่นหมึก (ink ripples) เป็นการเคลื่อนไหวที่เป็นจุดเด่นสำคัญของ แมททีเรียลดีไซน์ (Material Design) ที่จะช่วยยืนยันการป้อนข้อมูลของผู้ใช้งาน โดยลักษณะของการเคลื่อนไหวจะขยายออกจากจุดสัมผัสเป็นลักษณะระลอกคลื่นออกไป ซึ่งจะช่วยบอกผู้ใช้งานว่าการดิว (card view) โบนั้นเข้าสู่สถานะทำงานแล้ว (active state)



ภาพที่ 2.13 แสดงการเคลื่อนไหวการเชื่อมต่อกันระหว่างพื้นผิวและวัตถุ

จากภาพที่ 2.13 แสดงให้เห็นถึงลักษณะการเคลื่อนไหวการที่เชื่อมต่อกันระหว่างพื้นผิว (surfaces) และวัตถุ (element) ซึ่งในรูปเมื่อผู้ใช้งานกดที่ปุ่มเมนูแล้วการ์ดวิว (card view) ขยายออกมาจากเมนูซึ่งเป็นพื้นผิวเดียวกับพื้นหลัง

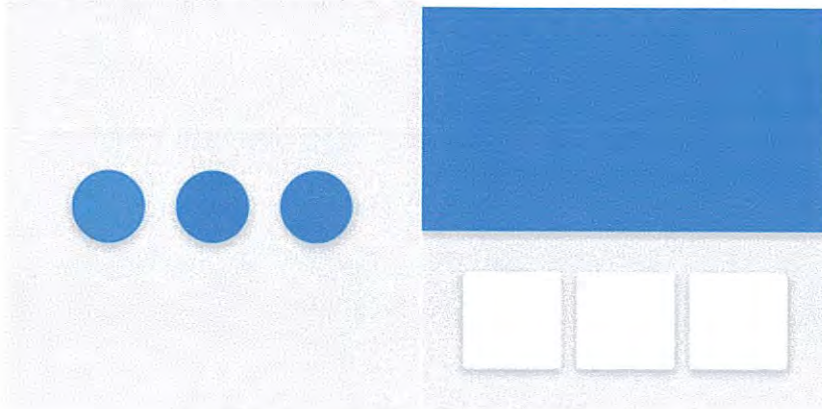
2.) เคลื่อนไหวแบบเป็นธรรมชาติ (Natural)



ภาพที่ 2.14 กราฟแสดงความแตกต่างการเคลื่อนไหวแบบ No Easing และ Easing

ในโลกแห่งความจริงความสามารถของวัตถุที่จะเพิ่มหรือลดความเร็วลงได้อย่างรวดเร็วเป็นผลมาจากน้ำหนักและแรงเสียดทานของพื้นผิว โดยแมททีเรียลดีไซน์ นำลักษณะการเคลื่อนไหวแบบ Easing มาเป็นมาตรฐานในการเคลื่อนไหวของวัตถุ ซึ่งเป็นการเคลื่อนไหวที่เหมือนธรรมชาติ โดยได้แรงบันดาลใจมาจากทฤษฎีแรงโน้มถ่วงประกอบกับการเคลื่อนไหวเป็นเส้นโค้งมากกว่าเส้นตรง

3.) เคลื่อนไหวแบบมีความสัมพันธ์กันสภาพแวดล้อมรอบข้าง (Aware)



ภาพที่ 2.15 ภาพการเคลื่อนไหวแบบ Aware

จากภาพที่ 2.15 แอปที่เรียลตี้ไซน์ จะแสดงการเคลื่อนไหวแบบที่มีความสัมพันธ์กันกับวัตถุรอบๆ ข้าง ซึ่งในภาพตัวอย่างเมื่อผู้ใช้งานกดที่ปุ่มสีฟ้าทางซ้ายสุดปุ่มสีฟ้าที่เหลืออีกสองปุ่มทางขวามือจะลอยมารวมกันกับปุ่มที่ผู้ใช้งานกดพร้อมขยายเป็นสี่เหลี่ยมผืนผ้าสีฟ้า และจากนั้นในเวลาเดียวกันจะมีการ์ดวิว (card view) สามชิ้นพุ่งขึ้นมาจากด้านล่าง ซึ่งเป็นการเคลื่อนไหว (animation) ที่ช่วยดึงดูดผู้ใช้งานได้มาก เพราะการเคลื่อนไหวแบบนี้มักทำให้ผู้ใช้งานรู้สึกประทับใจ

4.) เคลื่อนไหวได้ตั้งใจคิด (Intentional)



ภาพที่ 2.16 ภาพการเคลื่อนไหวแบบ Intentional

จากรูปที่ 2.16 การเคลื่อนไหวของแอปที่เรียลตี้ไซน์ จะดึงให้ผู้ใช้งานโฟกัสไปจุดใดจุดหนึ่ง ณ เวลาที่เหมาะสม เพราะการเคลื่อนไหวที่ทำให้ผู้ใช้งานโฟกัสที่จุดใดจุดหนึ่งจะทำให้ผู้ใช้งานเข้าใจได้เองจากการเคลื่อนไหว (animation) ว่าต้องทำอะไรต่อไป

5.) การเคลื่อนไหวต้องรวดเร็ว (Motion is quick)

การเคลื่อนไหวที่ดีในแบบของแอปที่เรียลตี้ไซน์ จะต้องมีการเคลื่อนไหวที่ตอบสนองกับผู้ใช้งานในเวลาที่ไม่นานเกินความจำเป็น คือผู้ใช้งานจะไม่รู้สึกว่าการเคลื่อนไหวช้าไปหรือไม่ เพราะอาจทำให้ผู้ใช้งานคิดว่าแอปพลิเคชันตัวนี้ทำงานช้าไป

6.) การเคลื่อนไหวต้องรวดเร็ว (Motion is quick)

การเคลื่อนไหวและการเปลี่ยนของวัตถุในแมททีเรียลดีไซน์ ต้องมีความชัดเจน เรียบง่าย และมีความสอดคล้องกัน โดยการเคลื่อนไหวในแต่ละครั้งจะหลีกเลี่ยงการเปลี่ยนของวัตถุที่มีมากจนเกินไปในแต่ละครั้ง

7.) การเคลื่อนไหวแบบเป็นกลุ่ม (Motion is cohesive)

วัตถุต่างของแมททีเรียลดีไซน์ ในหนึ่งหน้าจะมีการเคลื่อนไหวแบบเป็นปึกแผ่นหรือมีการเคลื่อนไหวพร้อมกันเป็นกลุ่ม โดยแต่ละวัตถุจะมีความเร็ว การตอบสนองที่ดึงดูดผู้ใช้งานอย่างสอดคล้องกัน ซึ่งจะทำให้การปรับแต่งแอปพลิเคชันมีการเคลื่อนไหวที่สอดคล้องกันทั้งแอปพลิเคชัน

2.4.3. ลักษณะของวัตถุในแมททีเรียลดีไซน์ (Style)

ลักษณะของวัตถุในแมททีเรียลดีไซน์ จำแนกได้ดังนี้

1.) รูปแบบของสี

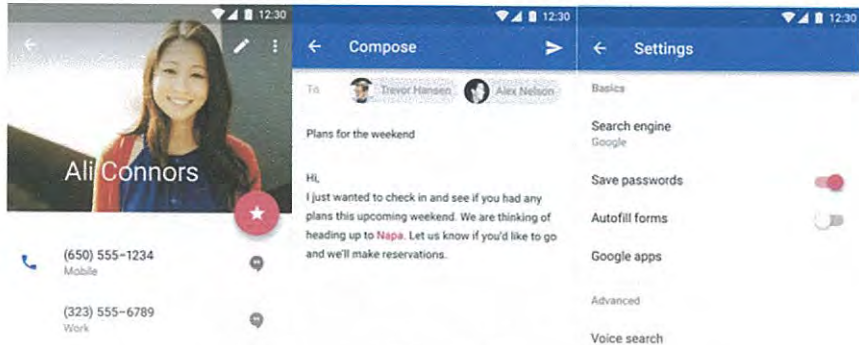


ภาพที่ 2.17 ภาพแสดงรูปแบบการใช้สีของ Material Design

การใช้สีในแมททีเรียลดีไซน์ (Material Design) จะถูกกำหนดให้เหมาะสมตามแบรนด์ของผลิตภัณฑ์ที่สร้างขึ้น โดยการใช้สีควรจะเลือกให้เหมาะสมและมีความชัดเจนระหว่างองค์ประกอบภายในแอปพลิเคชัน เพื่อให้เห็นถึงความชัดเจนของวัตถุในแอปพลิเคชัน ซึ่งข้อจำกัดในการเลือกสีเพื่อนำมาใช้ร่วมกับวัตถุอยู่ที่สามเฉดสี โดยมีการจัดลำดับของสีหลักและสีรองดังภาพที่ 2.17

- สีหลัก (Primary color): เป็นสีที่เลือกมาเป็นจุดเด่นของแอปพลิเคชัน จะต้องเป็นสีประจำของแบรนด์นั้นๆ โดยสีหลักจะถูกนำไปใช้ในทุกหน้าจอ และในหลายๆ วัตถุบนแอปพลิเคชัน
- สีรอง (Secondary color): เป็นสีที่ถูกนำมาใช้เพื่อบอกถึงส่วนที่เป็นข้อมูลที่เกี่ยวข้องต่างๆ บนแอปพลิเคชัน โดยรูปแบบของสีอาจจะเข้มหรือสว่างกว่าสีหลักก็ได้ตามความเหมาะสม
- สีที่ใช้เพื่อเน้น (Accent color): สีที่ใช้สำหรับเน้นจะถูกนำไปใช้กับปุ่มประเภทที่ลอยขึ้นมา (floating button) หรือวัตถุที่ต้องการโต้ตอบกับผู้ใช้ เช่น

ข้อความที่ต้องการเน้น หรือปุ่มที่กำลังทำงานอยู่หรือทำงานไปแล้ว ดังภาพที่ 2.18



ภาพที่ 2.18 ภาพการใช้สีเพื่อเน้นปุ่มและข้อความ

2.) สีของตัวอักษรและพื้นหลัง

Primary text	#000000	87%
Secondary text	#000000	54%
Disabled / Hint text	#000000	38%
Primary color	#3E50B4	100%
Accent color	#FF3F80	100%

ภาพที่ 2.19 การใช้สีตัวอักษรกับพื้นหลังสว่าง

จากภาพที่ 2.19 พื้นหลังแบบสว่างจะใช้สีของข้อความแบบมืดเพื่อให้เกิดการตัดกันอย่างชัดเจน ซึ่งข้อความหลักจะต้องมีความโปร่งใสของตัวอักษรอยู่ที่ 87% ข้อความรองจะต้องมีความโปร่งใสอยู่ที่ 54% ส่วนข้อความประเภทที่ให้เป็นคำแนะนำจะใช้ความโปร่งใสอยู่ที่ 38%

Primary text	#FFFFFF	100%
Secondary text	#FFFFFF	70%
Disabled / Hint text	#FFFFFF	50%
Primary color	#3E50B4	100%
Accent color	#FF3F80	100%

ภาพที่ 2.20 การใช้สีตัวอักษรกับพื้นหลังมืด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานภาษาอังกฤษและฟอนต์ Noto เป็นฟอนต์มาตรฐานสำหรับภาษาอื่น เช่น ไทย จีน ญี่ปุ่น เกาหลี เป็นต้น

2.4.4. รูปแบบการจัดวางเลย์เอาต์ของวัตถุในแมททีเรียลดีไซน์ (Layout)

แมททีเรียลดีไซน์ (Material Design) ได้มีการดึงเอา Grid มาช่วยในการออกแบบเลย์เอาต์ (layout) เพื่อใช้สำหรับใช้ข้ามแพลตฟอร์ม มีการตีเส้น สร้างขอบเขตของส่วนต่างๆ ให้มีจังหวะและน้ำหนักของการออกแบบให้เป็นไปในทางเดียวกันในทุกๆ หน้าจอและอุปกรณ์

1.) หน่วย Density-independent pixels (dp)

เนื่องจากแอนดรอยด์นั้นมีขนาดหน้าจอที่ต่างกันมาก ไม่ว่าจะเป็นขนาด 480x320, 800x480, 960x540, 1280x720 ดังนั้นการเขียนแอปพลิเคชันจะกำหนดพิกเซลที่ตายตัวไม่ได้ เพราะไม่เช่นนั้นแล้ว ถ้าหากนำไปเปิดบนหน้าจอที่มีความละเอียดที่แตกต่างกัน ก็จะทำให้การแสดงผลของภาพผิดเพี้ยนไปจากเดิมที่กำหนดไว้ ดังนั้นทางผู้พัฒนาระบบแอนดรอยด์จึงได้คิดค้นหน่วยหนึ่งขึ้นมา เพื่อให้ความละเอียดหน้าจอที่แตกต่างกันเมื่อแปลงเป็นหน่วยนี้ จะได้ค่าออกมาใกล้เคียงกันที่สุด ซึ่งได้ถูกนำมาใช้กับแมททีเรียลดีไซน์ ด้วยเพื่อที่ออกแบบแอปพลิเคชันใดก็ตาม ที่ใช้หน่วยนี้จะได้เกิดการคลาดเคลื่อนของดีไซน์น้อยที่สุด



ภาพที่ 2.23 ภาพแสดงความแตกต่างของหน้าจอเทียบกับหน่วย dp

จากภาพที่ 2.23 จะเห็นได้ว่าจอขนาดเล็ก (small) จะมีขนาดจอย่างน้อย 426 x 320 dp จอขนาดกลาง (normal) จะมีขนาดจอย่างน้อย 470 x 320 dp จอขนาดใหญ่ (large) จะมีขนาดจอย่างน้อย 640 x 480 dp และจอขนาดที่ใหญ่มาก (xlarge) จะมีขนาดจอย่างน้อย 960 x 720

ตารางที่ 2.1 ตารางแสดงค่า dpi เทียบกับขนาดหน้าจอ

ขนาดหน้าจอ	dpi	ขนาดของภาพในหน่วยพิกเซล (px)
xxxhdpi	640	400 x 400
xxhdpi	480	300 x 300
xhdpi	320	200 x 200
hdpi	240	150 x 150
mdpi	160	100 x 100

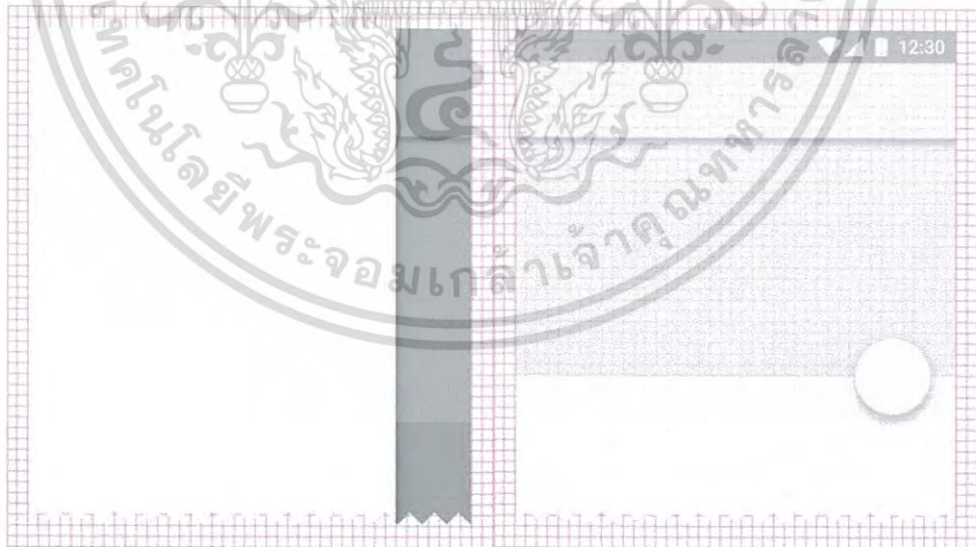
โดยทางกูเกิ้ล (Google) ได้กำหนดค่า dpi ไว้เพื่อเป็นค่าคงมาตรฐานของความหนาแน่น (density) ดังตารางที่ 2.1 ยกตัวอย่างถ้าจะคำนวณหาค่า dp จาก Samsung Nexus 10 ซึ่งเครื่องนี้จะเป็น xlarge-xhdpi ดังนั้นค่า dpi จะเท่ากับ 320 และเครื่องนี้มีความละเอียดหน้าจอเท่ากับ 2560 x 1600

$$2560 \text{ px} : 2560 \times (160 / 320) = 1280 \text{ dp}$$

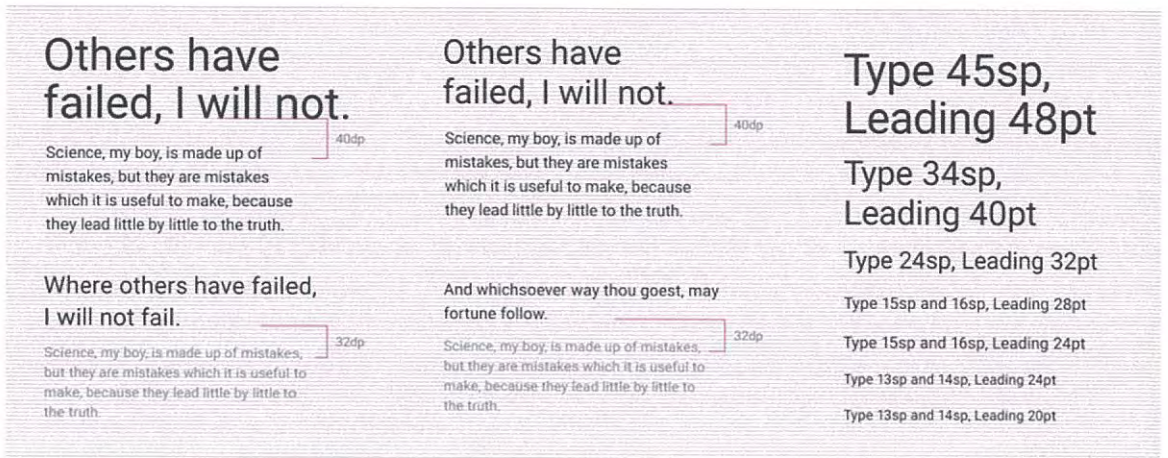
$$1600 \text{ px} : 1600 \times (160 / 320) = 800 \text{ dp}$$

จะได้ขนาดหน้าจอ 1280 x 800 dp ซึ่งตรงกับขนาด xlarge (โดยมีขนาดอย่างน้อย 960 x 720 dp)

2.) เส้นกริดพื้นฐาน (Baseline grids)

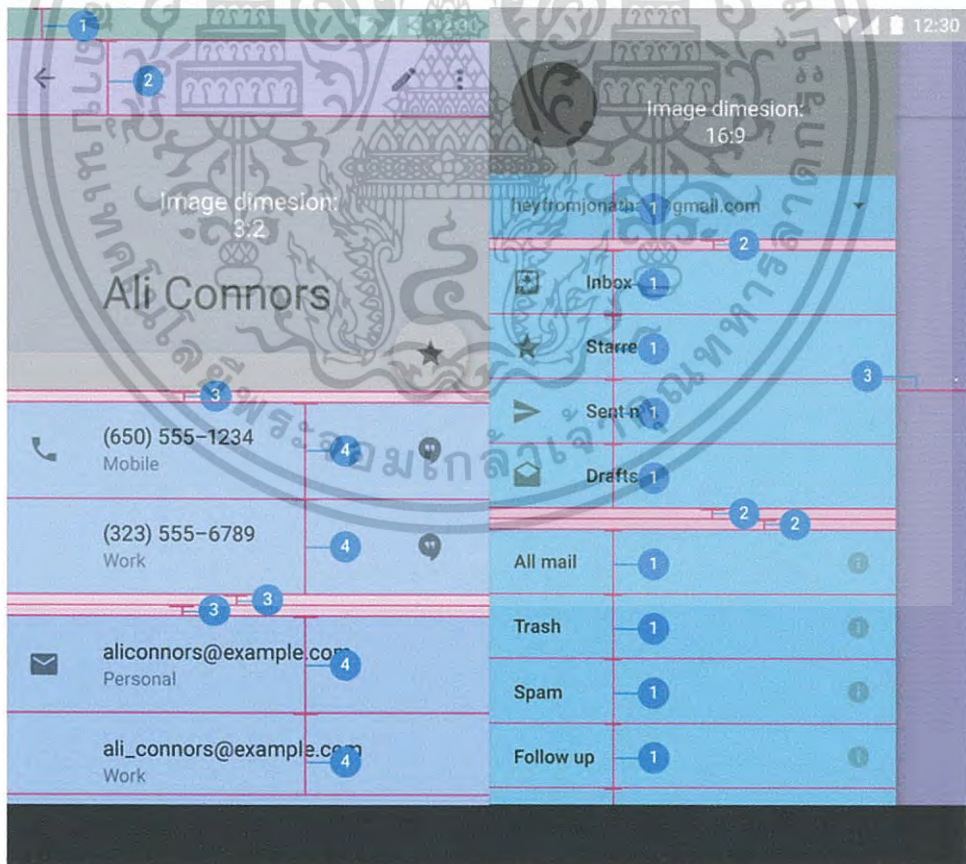


ภาพที่ 2.24 ตัวอย่างเส้นกริดพื้นฐานของที่ใช้กับหน้าจอ



ภาพที่ 2.25 ตัวอย่างเส้นกริดพื้นฐานที่ใช้กับข้อความ

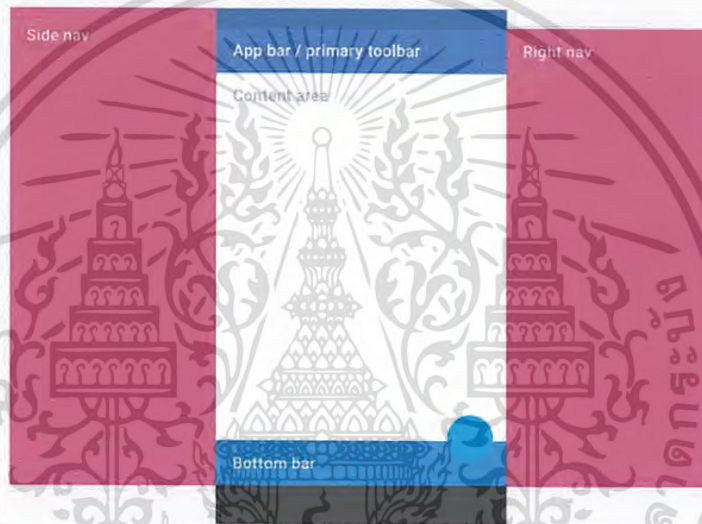
เส้นกริดพื้นฐานของแมททีเรียลดีไซน์ (Material Design) ถูกออกแบบเป็นรูปสี่เหลี่ยมจัตุรัสขนาด 8 dp สำหรับจอโทรศัพท์มือถือ (mobile) จอแท็บเล็ต (tablet) และจอคอมพิวเตอร์ส่วนบุคคล (desktop) ส่วนข้อความต่างๆ ใช้กริดขนาด 4 dp ดังภาพที่ 2.24 และภาพที่ 2.25



ภาพที่ 2.26 ตัวอย่างการใช้เส้นกริดบนหน้าจอมือถือ

จากภาพที่ 2.26 รูปทางด้านซ้ายกำหนดความสูงของสเตตัสบาร์ (Status bar) เป็น 24dp (ตำแหน่งที่ 1) กำหนดความสูงของทูลบาร์ (Toolbar) เป็น 56dp (ตำแหน่งที่ 2) กำหนดช่องว่างระหว่างข้อความ (Space between content areas) เป็น 8dp (ตำแหน่งที่ 3) และกำหนดความสูงของส่วนที่เป็นข้อความ (List item) เป็น 72dp (ตำแหน่งที่ 4) ส่วนรูปทางขวามือเป็นลักษณะเมนูของแมททีเรียลดีไซน์ (Material Design) ซึ่งกำหนดส่วนที่ใส่อีเมล (Account menu and list items) เป็น 48dp (ตำแหน่งที่ 1) กำหนดช่องว่างระหว่างข้อความ (Space between content areas) เป็น 8dp (ตำแหน่งที่ 2) และกำหนดช่องว่างระหว่างขอบจอทางด้านขวากับเมนู (Navigation right margin) เป็น 56dp (ตำแหน่งที่ 3)

3.) โครงสร้างยูไอ (UI) บนโทรศัพท์มือถือ



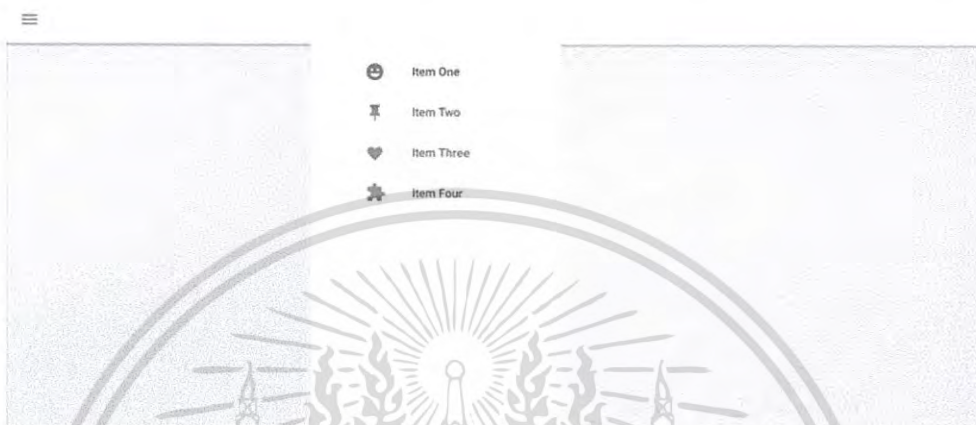
ภาพที่ 2.27 ภาพโครงสร้างยูไอ (UI) บนโทรศัพท์มือถือ

โครงสร้างของยูไอ (UI) บนมือถือจะประกอบด้วยสเตตัสบาร์ (Status bar) และแอปบาร์ (App bar) อยู่ชิดด้านบนสุดของจอ ส่วนบาร์ด้านล่าง (Button bar) ปัจจุบันทางกูเกิ้ล (Google) ได้เปลี่ยนเป็นเมนูบาร์ ซึ่งถือว่าเป็นยูเอกซ์ (UX) ที่ดีในปัจจุบัน และอีกหนึ่งปุ่มที่ทางกูเกิ้ล (Google) กำหนดไว้ด้านล่างคือ ปุ่มโฟลติง (floating button) ซึ่งเป็นปุ่มที่ไว้ใช้งานฟังก์ชันหลักในหน้านั้นๆ ส่วนทางซ้ายจะเป็นไซด์บาร์ (Side nav) ซึ่งมีไว้เก็บเมนูรองโดยจะมีแฮมเบอร์เกอร์ไอคอน (hamburger icon) เป็นปุ่มสำหรับสื่อถึงเมนู เมื่อกดที่แฮมเบอร์เกอร์ไอคอน (hamburger icon) แล้วไซด์บาร์ (Side nav) จะเลื่อนออกมาจากทางซ้าย ส่วนบาร์ทางขวา (Right nav) จะมีไว้สำหรับเป็นหน้าต่างแจ้งเตือน ซึ่งมักไม่ค่อยพบเห็นบนแอปพลิเคชันนัก เพราะแค่ไซด์บาร์ (Side nav) ก็สามารถเก็บเมนูรองได้ค่อนข้างครบถ้วนแล้ว

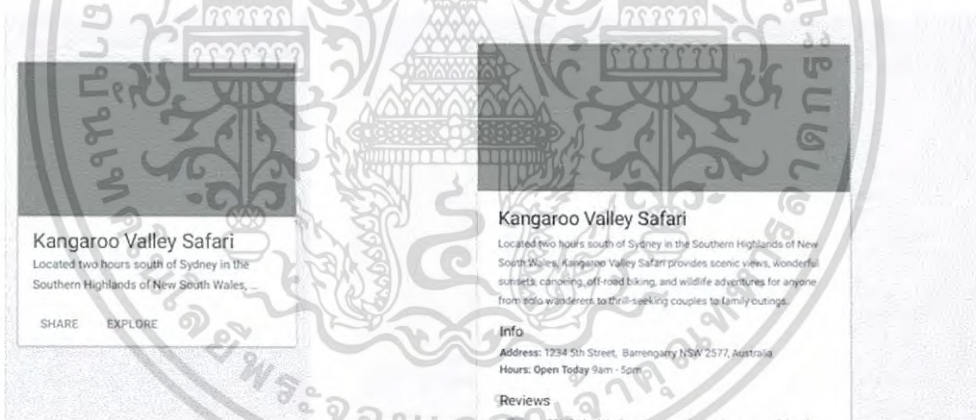
4.) รูปแบบการตอบสนองผู้ใช้ตามขนาดของหน้าจอ

เนื่องการขนาดหน้าจอที่มีมากในระบบปฏิบัติการแอนดรอยด์ทำให้ต้องออกแบบยูไอ (UI) ให้รองรับกับหน้าจอทุกขนาด จึงทำให้ต้องมีการจัดวางเลย์เอาต์ (Layout) ที่แตกต่างกันไปในแต่ละหน้าจอ เพื่อให้ผู้ใช้งานสามารถใช้งานได้ง่ายและสะดวกที่สุด มีรูปแบบการตอบสนองดังนี้

- ซ่อนและแสดงผล (Reveal)



ภาพที่ 2.28 ตัวอย่างการซ่อนและแสดงผลของเมนูในหน้าจอทั้งสองแบบ



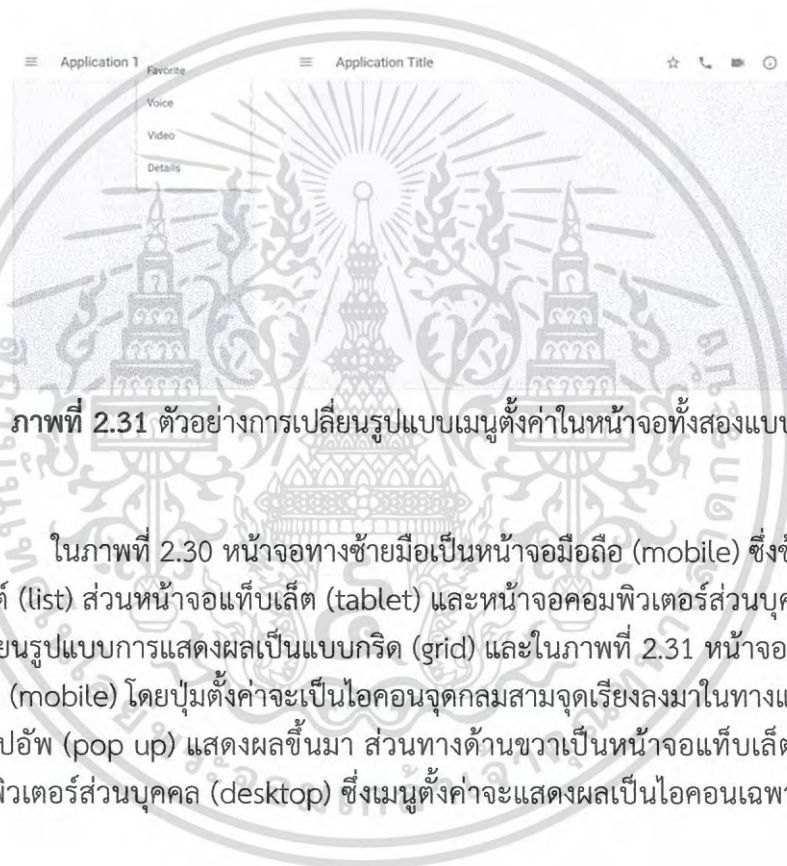
ภาพที่ 2.29 ตัวอย่างการซ่อนและแสดงผลของข้อมูลบนการ์ดวิวในหน้าจอทั้งสองแบบ

ในภาพที่ 2.28 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) ซึ่งเมนูจะถูกซ่อนไว้ เมื่อผู้ใช้งานกดที่แฮมเบอร์เกอร์ไอคอน (hamburger icon) จะมีเมนูแสดงผลออกมา ซึ่งต่างจากหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) จะมีเมนูแสดงผลออกมาให้เห็นเลย ส่วนภาพที่ 2.29 ข้อมูลจะถูกซ่อนไว้บางส่วนในหน้าจอมือถือ (mobile) และใช้การแสดงผลแบบให้กดเข้าไปอ่านเพิ่มเติมแทนและจะแสดงผลทั้งหมดบนหน้าจอจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop)

- เปลี่ยนรูปแบบ (Transform)



ภาพที่ 2.30 ตัวอย่างการเปลี่ยนรูปแบบของข้อมูลในหน้าจอทั้งสองแบบ



ภาพที่ 2.31 ตัวอย่างการเปลี่ยนรูปแบบเมนูตั้งค่าในหน้าจอทั้งสองแบบ

ในภาพที่ 2.30 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) ซึ่งข้อมูลจะถูกเรียงลงมาเป็นลิสต์ (list) ส่วนหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) ข้อมูลถูกเปลี่ยนรูปแบบการแสดงผลเป็นแบบกริด (grid) และในภาพที่ 2.31 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) โดยปุ่มตั้งค่าจะเป็นไอคอนจุดกลมสามจุดเรียงลงมาในทางแนวนตั้ง เมื่อกดที่ปุ่มนี้จะมีป๊อปอัพ (pop up) แสดงผลขึ้นมา ส่วนทางด้านขวาเป็นหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) ซึ่งเมนูตั้งค่าจะแสดงผลเป็นไอคอนเฉพาะแยกกันอย่างชัดเจน

- การแบ่งพื้นที่ว่างของเนื้อหา (Divide)



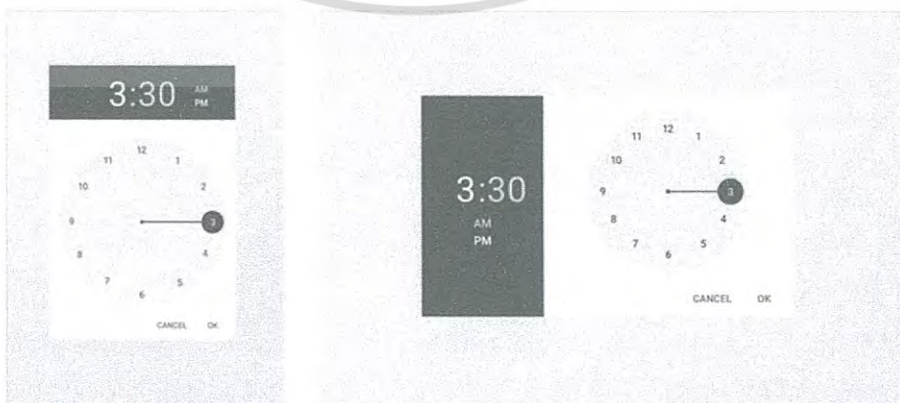
ภาพที่ 2.32 ตัวอย่างการแบ่งพื้นที่ว่างในหน้าจอทั้งสองแบบ

ในภาพที่ 2.32 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) มีการแบ่งพื้นที่ว่างสำหรับใส่ข้อมูลให้เหมาะสมกับหน้าจอ ส่วนหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) มีการแบ่งพื้นที่ว่างมากกว่าหน้าจอมือถือ เนื่องจากมีพื้นที่ว่างของหน้าจอมากจึงสามารถแบ่งพื้นที่ไว้สำหรับเนื้อหาได้มากกว่า

- จัดการยูไอให้เหมาะสม (Reflow)



ภาพที่ 2.33 ตัวอย่างการจัดเลย์เอาต์ให้เหมาะสมในหน้าจอทั้งสองแบบ

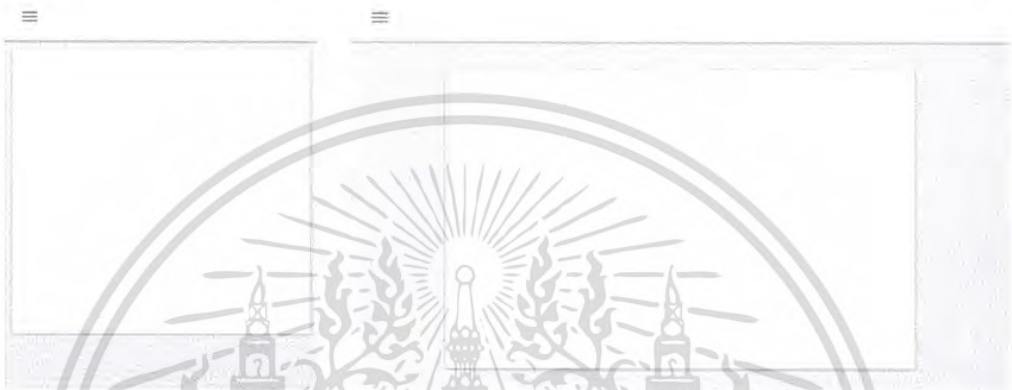


ภาพที่ 2.34 ตัวอย่างยูไอของนาฬิกาในหน้าจอทั้งสองแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาพที่ 2.33 แสดงให้เห็นถึงความแตกต่างระหว่างการจัดรูปแบบเลย์เอาต์ (layout) ของหน้าจอมือถือ (mobile) หน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) ส่วนภาพที่ 2.34 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) มีลักษณะการวางยูไอ (UI) โดยให้เลขบอกเวลาแบบดิจิทัลอยู่ด้านบนและให้หน้าปัดของนาฬิกาอยู่ด้านล่าง ส่วนหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ตั้งโต๊ะ (desktop) มีลักษณะการวางยูไอ โดยให้เลขบอกเวลาแบบดิจิทัลอยู่ทางด้านซ้ายและให้หน้าปัดของนาฬิกาอยู่ด้านขวา

- ขยาย (Expand)



ภาพที่ 2.35 ตัวอย่างการขยายของเลย์เอาต์ในหน้าจอทั้งสองแบบ

ในภาพที่ 2.35 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) จะเห็นว่าเลย์เอาต์ (layout) มีการจัดวางเป็นสี่เหลี่ยมจัตุรัสขนาดเกือบเต็มจอ ส่วนหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) มีการเลย์เอาต์ (layout) มีการจัดวางเป็นสี่เหลี่ยมผืนผ้า จะเห็นว่าเลย์เอาต์ (layout) มีการขยายออกจากเดิม เนื่องจากมีพื้นที่ว่างของหน้าจอมากกว่านั่นเอง

- เปลี่ยนตำแหน่ง (Position)



ภาพที่ 2.36 ตัวอย่างการขยายของเลย์เอาต์ในหน้าจอทั้งสองแบบ

ในภาพที่ 2.36 หน้าจอทางซ้ายมือเป็นหน้าจอมือถือ (mobile) จะเห็นปุ่มสี่เหลี่ยมที่ลอยขึ้นมา (floating button) ถูกจัดอยู่มุมขวาล่างของจอ ส่วนหน้าจอแท็บเล็ต (tablet) และหน้าจอคอมพิวเตอร์ส่วนบุคคล (desktop) ถูกจัดปุ่มนี้อยู่ทางขวาบนตรงเส้นกั้นระหว่างส่วนหัวและส่วนเนื้อหาของหน้าเนื่องจากมีพื้นที่ของหน้าจอมากกว่านั่นเอง

2.4.5. องค์ประกอบในแมททีเรียลดีไซน์ (Components)

กูเกิล (Google) ได้วางมาตรฐานองค์ประกอบของแมททีเรียลดีไซน์ (Material Design) ไว้ตั้งแต่เปิดตัวในงาน Google I/O ซึ่งปัจจุบันได้มีการเปลี่ยนแปลงองค์ประกอบ (Components) ต่างๆ นี้ไปตามยุคสมัยและผลวิจัยการใช้งานของผู้ใช้ ซึ่งองค์ประกอบ (Components) ที่ทางกูเกิลกำหนดไว้มีมาก ทางคณะผู้จัดทำจึงขอเสนอเพียงองค์ประกอบ (Components) ที่นำมาใช้กับแอปพลิเคชันนี้เท่านั้น

1.) ปุ่มเมนูด้านล่าง (Bottom Navigation)

เดิมทีแล้วแอนดรอยด์ได้เคยกำหนดยูเอกซ์ (UX) ของเรื่องเมนู (navigation) หลักๆ ก่อนหน้านี้ไว้โดยมีแท็บด้านบน (Tabs) และแท็บที่เลื่อนออกมาจากทางซ้าย (navigation drawer) ด้วยการกดปุ่มแฮมเบอร์เกอร์ไอคอน (hamburger icon) ที่เป็นเมนูหลัก (main navigation) หลักของแอปพลิเคชันโดยได้กำหนดไว้ว่า “ไม่ควรมีวิดจอะไรวางไว้ด้านล่าง” เพราะจะทำให้ผู้ใช้งานมีโอกาสกดไปโดนปุ่มของเครื่องด้านล่างที่ไม่ต้องการจะกดได้ แต่ปัจจุบันกูเกิลได้กำหนดปุ่มเมนูด้านล่าง (Bottom navigation) ลงได้ในหน้าจอรูปร่างแบบดีไซน์ที่แนะนำ (Material Design Guideline) โดยมีลักษณะคล้ายกับแท็บ (Tabs) แต่ไม่สามารถเลื่อนได้ ดังภาพที่ 2.37



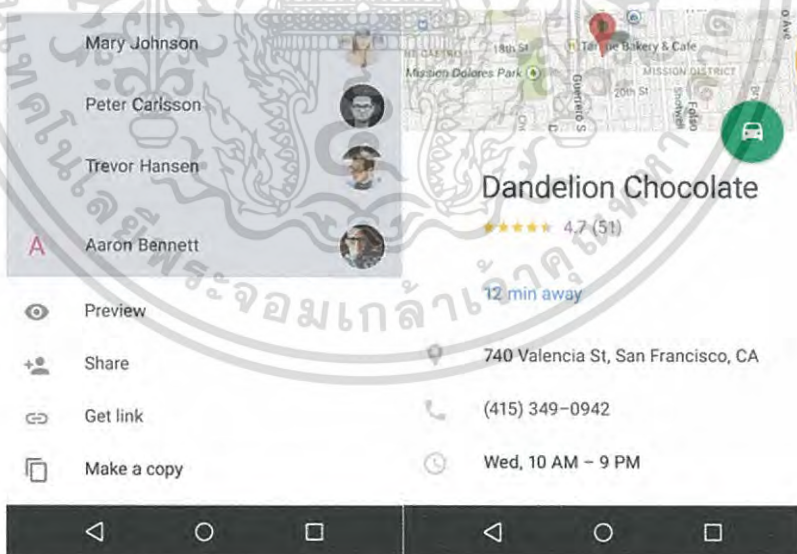
ภาพที่ 2.37 ภาพปุ่มเมนูด้านล่างของหน้าจอมือถือ แท็บเล็ต และคอมพิวเตอร์ส่วนบุคคล



ภาพที่ 2.38 ตัวอย่างการใช้งานปุ่มเมนูด้านล่าง (bottom navigation)

ในภาพที่ 2.38 เป็นการใช้งานปุ่มเมนูด้านล่าง (bottom navigation) ซึ่งมีไว้สำหรับใช้เปลี่ยนหน้าหลักของแอปพลิเคชัน ซึ่งเมื่อเปลี่ยนหน้าไปแล้วปุ่มเมนูด้านล่าง (bottom navigation) นี้จะอยู่เหมือนเดิม และสามารถกดสลับกลับไปทีหน้าเดิมได้ตลอด แต่สิ่งที่ปุ่มนี้ต่างจากปุ่มแท็บ (Tabs) ไป คือ ปุ่มนี้เหมาะสำหรับใช้สลับหน้าเพียง 3 - 5 หน้าเท่านั้น โดยที่แต่ละไอคอนจะถูกกำหนดตำแหน่งไว้อย่างคงที่ (position fix) และสำหรับเรื่องสี ไม่ควรจะใช้สีที่หลากหลายนเกินไป ควรจะใช้สีที่ตรงกับผลิตภัณฑ์หรือใช้สีขาวกับสีดำเท่านั้น โดยกำหนดสีสำหรับปุ่มที่ถูกเลือก และปุ่มที่ไม่ถูกเลือกให้ชัดเจนเพื่อป้องกันไม่ให้ผู้ใช้งานสับสน ส่วนข้อความควรใช้เป็นเพียงคำสั้นๆ เพื่อให้แสดงผลจบภายในบรรทัดเดียว

2.) แผ่นแสดงเนื้อหาทางด้านล่าง (Bottom Sheets)



ภาพที่ 2.39 ตัวอย่างการใช้งานปุ่มเมนูด้านล่าง (bottom navigation)

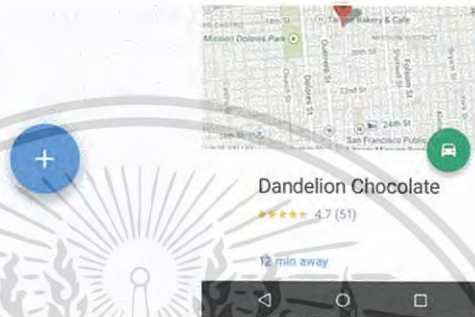
จากภาพที่ 2.39 เป็นแผ่นแสดงเนื้อหาทางด้านล่าง (Bottom Sheets) ซึ่งมีไว้ใช้แสดงข้อมูลจำนวนมากๆ เมื่อกดปุ่มให้แสดงผลขึ้นมา โดยถ้าเป็นแผ่นที่แสดงผลเนื้อหาที่อยู่ภายในแอปพลิเคชันเองจะมีชื่อเรียกว่า Persistent Bottom Sheets ส่วนแผ่นที่แสดงผลเนื้อหาให้เห็นถึงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมโยงไปยังเว็บไซต์หรือแอปพลิเคชันอื่นจะเรียกว่า Modal Bottom Sheets โดยการเคลื่อนไหว (animation) ของแผ่นแสดงเนื้อหาทางด้านล่าง (Bottom Sheets) นี้จะเป็นรูปแบบการสไลด์ขึ้นมาจากด้านล่างสุดของหน้าจอ

3.) ปุ่ม (Button)

ปุ่มถูกออกแบบเป็นมาตรฐานของแมททีเรียลดีไซน์ (Material Design) มีอยู่ด้วยกัน 3 แบบ คือ

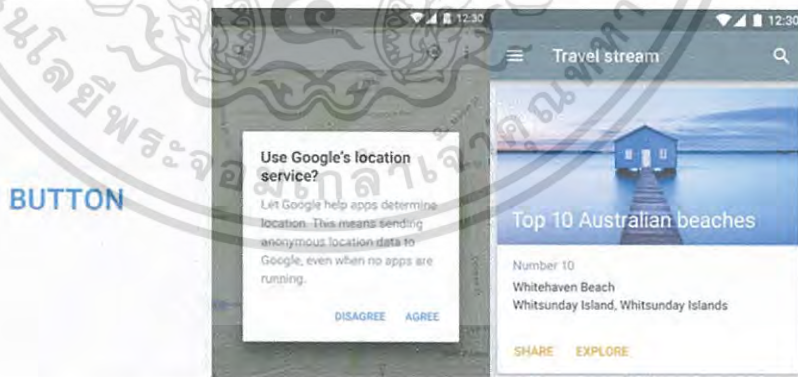
- ปุ่มลอย (Floating action button)



ภาพที่ 2.40 ปุ่มลอย (Floating action button)

เป็นปุ่มกลมที่ถูกออกแบบมาให้มีลักษณะลอยอยู่บนหน้าจอ โดยมีหน้าที่เป็นปุ่มหลักที่แสดงให้ผู้ใช้งานรับรู้ถึงฟังก์ชันหลักในหน้านั้นๆ เช่น ถ้าหากเข้าไปที่กูเกิ้ลแมพ (Google Map) เพื่อให้แอปพลิเคชันนำทาง ปุ่มนี้จะเป็นปุ่มหลักที่ลอยขึ้นมาบนหน้าจอ ดังภาพที่ 2.40

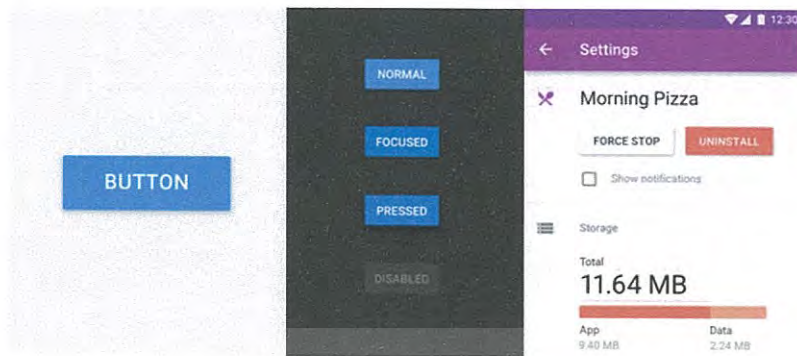
- ปุ่มแบน (Flat button)



ภาพที่ 2.41 ปุ่มแบน (Flat button)

เป็นปุ่มแสดงเพียงตัวอักษรเท่านั้น โดยปุ่มจะไม่มีกรวยให้ลอยขึ้น แต่จะให้สีที่เหมาะสมและตัดกับพื้นหลังอย่างชัดเจน เพื่อให้เข้าใจว่าสามารถกดได้ โดยมีหน้าที่เป็นปุ่มแอคชั่นที่ผู้ใช้สามารถกดได้บนไดอะล็อกบ็อกซ์ (Dialog Box) หรือเป็นปุ่มแอคชั่นที่อยู่ใต้ข้อความในการควิ โดยมีเทคนิคการใช้สีที่ทำให้ปุ่มโดดเด่นขึ้นเพื่อให้ผู้ใช้งานมองเห็นได้ง่าย ดังภาพที่ 2.41

- ปุ่มยก (Raised button)

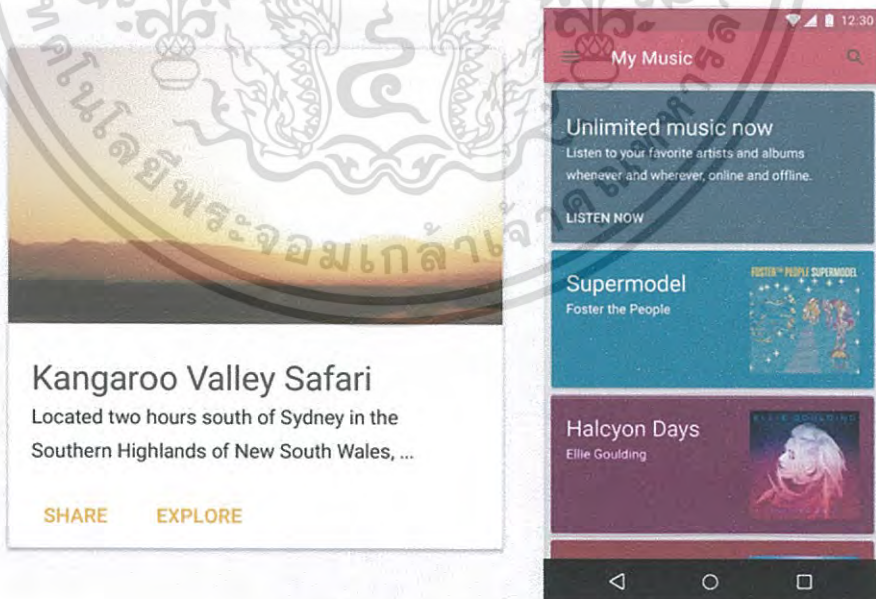


ภาพที่ 2.42 ปุ่มยก (Raised button)

เป็นปุ่มที่ออกแบบมาโดยใส่กรอบ และเพิ่มมิติให้กับปุ่ม ทำให้ดูเหมือนกับว่าปุ่มลอยขึ้นมาเล็กน้อย ซึ่งถูกใช้เน้นปุ่มฟังก์ชันที่ต้องการให้เห็นชัดเจนในพื้นที่ว่างที่ดูกว้างกว่าปกติ เพื่อให้รู้สึกว่ามีพื้นที่แฉกนั้นไม่โล่งจนเกินไป เช่น ปุ่ม Uninstall ถ้าหากใช้ปุ่มแบนจะทำให้รู้สึกโล่งผิดปกติ โดยปุ่มยก (Raised button) จะมีลักษณะที่เปลี่ยนแปลงไปตามพฤติกรรมกรกดปุ่มของผู้ใช้งาน ดังภาพที่ 2.42

4.) การ์ด (Cards)

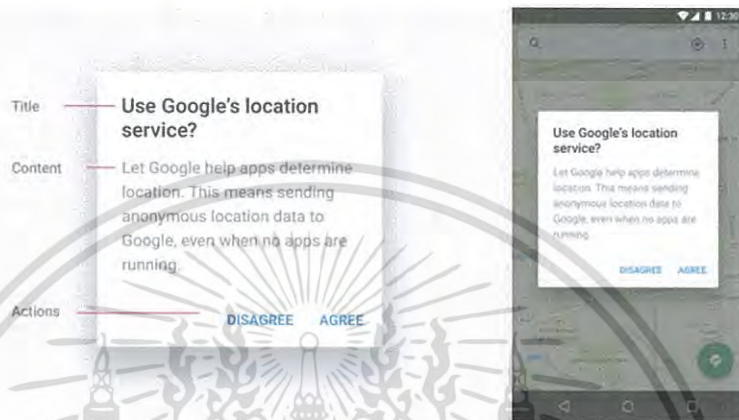
การ์ดเป็นตัวที่ช่วยแสดงเนื้อหาที่มีความแตกต่างของวัตถุในการ์ดเดียวกันได้ดี ไม่ว่าจะเป็นการแสดงผลภาพ ข้อความ หรือปุ่มต่างๆ ที่มีขนาดหรือสีที่แตกต่างกัน



ภาพที่ 2.43 การ์ด (Cards)

การที่จะใช้เลย์เอาท์ (Layout) ในรูปแบบการดั้นนั้นจะต้องดูว่า ในกลุ่มของวัตถุนั้นจะต้องประกอบไปด้วยภาพ ข้อความ และปุ่มต่างๆ รวมกันอยู่ ไม่ใช่มีเพียงภาพหรือตัวหนังสืออย่างเดียวเพียงอย่างหนึ่ง ข้อความที่อยู่ภายในการดั้นสามารถรองรับข้อความยาวๆ ได้ เช่น การแสดงความคิดเห็น เป็นต้น สำหรับข้อความที่มีความยาวมาก จะซ่อนข้อความบางกลุ่มทางท้ายสุดไปเป็นลักษณะจุดสามจุดแทน (...) และการดิวที่ดีจะต้องมีลักษณะโค้งที่มุมการดั้นขนาดเป็น 2 dp ดังภาพที่ 2.43

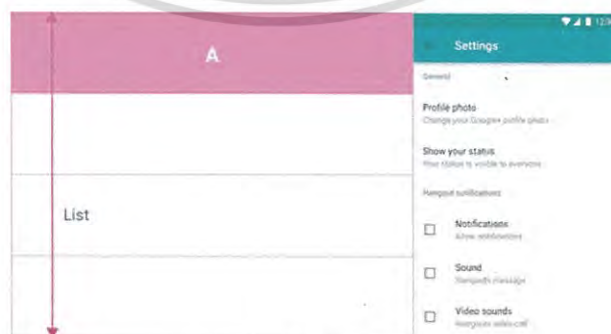
5.) กล่องแจ้งเตือน (Dialog)



ภาพที่ 2.44 กล่องแจ้งเตือน (Dialog)

จากภาพที่ 2.44 เป็นกล่องแจ้งเตือน (Dialog) ที่ถูกเิกิดได้ออกแบบไว้ในรูปแบบดีไซน์ที่แนะนำ (Material Design Guideline) โดยส่วนประกอบของกล่องแจ้งเตือนจะประกอบไปด้วยหัวข้อ (Title) ข้อความ (Content) และปุ่มแอคชัน (Action) เพื่อมีไว้ตอบโต้กับผู้ใช้งาน การใช้งานกล่องแจ้งเตือน (Dialog) นั้นไม่ควรจะมีการใช้บ่อยจนเกินไป เพราะการที่มีกล่องแจ้งเตือนขึ้นมาจะทำให้ผู้ใช้งานหยุดแล้วไปอ่านที่กล่องแจ้งเตือนนี้ ดังนั้นการเลือกกล่องแจ้งเตือนควรใช้เฉพาะจุดที่ต้องการให้ผู้ใช้งานรับรู้ข้อมูลนั้นจริงๆ และตัดสินใจที่จะทำอะไรต่อกับแอปพลิเคชัน โดยลักษณะของกล่องแจ้งเตือน (Dialog) นี้จะต้องไม่บดบังหน้าจอทั้งหมดเวลาแสดงผล ดังรูป

6.) ลิสต์ (List)



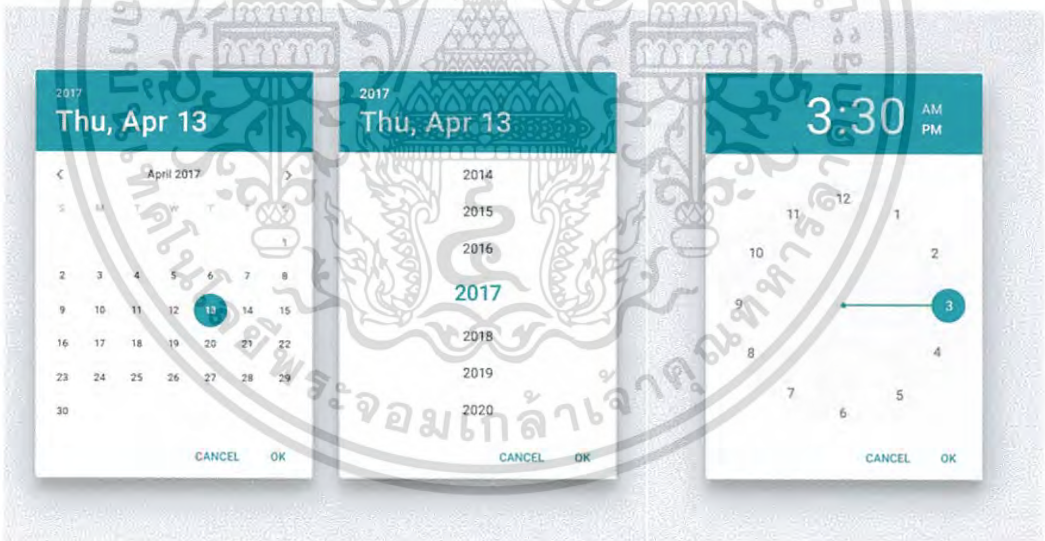
ภาพที่ 2.45 ลิสต์ (List)

Single-line item	Two-line item Secondary text
Single-line item	Two-line item Secondary text
Single-line item	Two-line item Secondary text

ภาพที่ 2.46 ประเภทของลิสต์

จากภาพที่ 2.45 ลิสต์ (List) หรือรายการเป็นองค์ประกอบส่วนที่ส่วนใหญ่แล้วแอปพลิเคชันจะมี เพราะมีไว้สำหรับให้ผู้ใช้งานอ่านเรียงลงมาหรือใช้แสดงข้อมูลจำนวนมากเป็นลำดับ เช่น รายการแสดงชื่อเพื่อนบนเฟซบุ๊ก (Facebook) หรือรายการการตั้งค่าในหน้าการตั้งค่า (Settings) เป็นต้น โดยแต่ละรายการ (List) จะต้องมัลักษณะการจัดรูปแบบที่เหมือนกัน เพื่อให้ผู้ใช้รู้สึกว่าเป็นลิสต์เดียวกัน ซึ่งลักษณะของลิสต์ที่กูเกิ้ล (Google) กำหนดขึ้นมีด้วยกันอยู่ 2 ประเภท ได้แก่ ลิสต์แบบบรรทัดเดียว (Single-line list) และลิสต์แบบสองบรรทัด (Two-line list) ดังภาพที่ 2.46

7.) พิคเกอร์ (Pickers)



ภาพที่ 2.47 พิคเกอร์แสดงวันที่และเวลาที่ถูกลอกแบบ

จากภาพที่ 2.47 เป็นพิกเกอร์ (Pickers) ที่ทางกูเกิ้ล (Google) ออกแบบมาเพื่อเป็นไดอะล็อก (Dialog) แสดงผลปฏิทินและแสดงผลเวลาเฉพาะของแมททีเรียลดีไซน์ (Material Design) ซึ่งทางกูเกิ้ลบอกว่าเป็นยูไอ (UI) ที่ออกแบบมาแล้วเหมาะสมที่สุดในการแสดงผลบนหน้าจอมือถือ (Best suited for display)

8.) ตัวเลือก (Selection controls)

ลักษณะของปุ่มตัวเลือกโดยมาตรฐานมีด้วยกันอยู่ 3 แบบ คือ ช่องทำเครื่องหมาย (Checkboxes), ปุ่มตัวเลือก (Radio buttons) และสวิตช์ (Switches)

- ช่องทำเครื่องหมาย (Checkboxes)



ภาพที่ 2.48 ช่องทำเครื่องหมาย (Checkboxes)

ช่องทำเครื่องหมาย (Checkboxes) ถูกออกแบบให้ใช้สำหรับเลือกข้อมูลหลายรายการจากตัวเลือกเพียงหนึ่งชุด ถ้าหากมีข้อมูลเพียงรายการเดียวไม่เหมาะสมจะใช้ช่องทำเครื่องหมาย (Checkboxes)

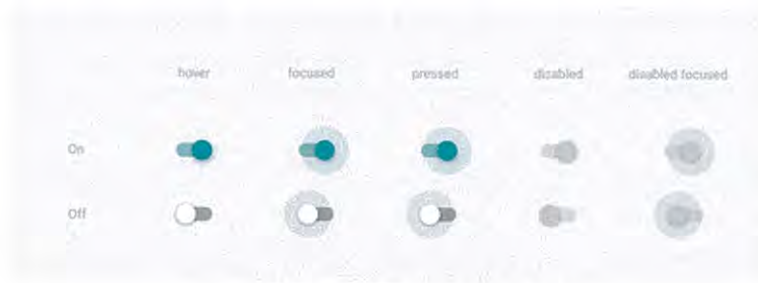
- ปุ่มตัวเลือก (Radio buttons)



ภาพที่ 2.49 ปุ่มตัวเลือก (Radio buttons)

ปุ่มตัวเลือก (Radio buttons) ถูกออกแบบให้ใช้สำหรับเลือกข้อมูลเพียงตัวเดียวจากตัวเลือกเพียงหนึ่งชุด ถ้าหากผู้ใช้งานเปลี่ยนไปเลือกข้อมูลอีกตัวในข้อมูลชุดนั้น ข้อมูลชุดเก่าที่เคยถูกเลือกไว้จะถูกยกเลิกการเลือกไป

- สวิตช์ (Switches)



ภาพที่ 2.50 สวิตช์ (Switches)

สวิตช์ (Switches) ถูกออกแบบสำหรับใช้สลับสถานะของตัวเลือกนั้นๆ เช่น ปุ่มสำหรับ เปิด/ปิด (on/off) การแจ้งเตือนของแอปพลิเคชัน เป็นต้น สำหรับการใช้งานปุ่มสวิตช์ เพียงแค่ผู้ใช้งานใช้นิ้วกดไปทางขวามือเพื่อให้สวิตช์เข้าสู่สถานะเปิดใช้งาน เมื่อเข้าสู่สถานะเปิดใช้งาน ปุ่มนี้จะแสดงสีขึ้นมาให้รู้ว่าอยู่ในสถานะเปิดใช้งานแล้ว หากต้องการสลับกลับไปยังสถานะปิดการใช้งาน ก็สามารถทำได้โดยการปัดกลับไปทางซ้ายมือ

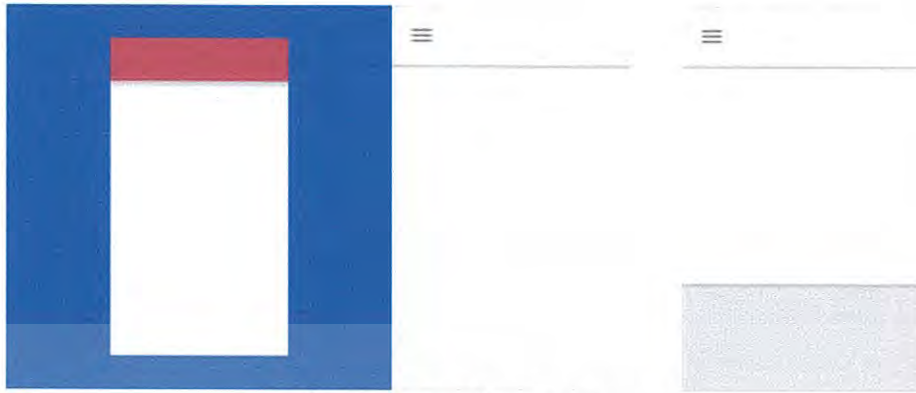
9.) แท็บ (Tabs)



ภาพที่ 2.51 แท็บ (Tabs)

แท็บ (Tabs) เป็นองค์ประกอบที่มีหน้าที่สำหรับสลับหน้าตาหรือมุมมองที่ใช้แสดงเนื้อหาที่มีความแตกต่างกัน ลักษณะของแท็บที่ดีที่ทางกูเกิ้ลออกแบบมาจะต้องมีเพียงแถวเดียว ซึ่งสามารถเลื่อนไปทางซ้ายและขวา (scroll) ได้ขึ้นอยู่กับกรออกแบบของแต่ละคน ซึ่งแท็บ (Tabs) เป็นรูปแบบของยูเอ็กซ์ (UX) ที่ดีละเอียดง่ายที่สุดในการใช้งาน

10.) แท็บเครื่องมือ (Toolbars)

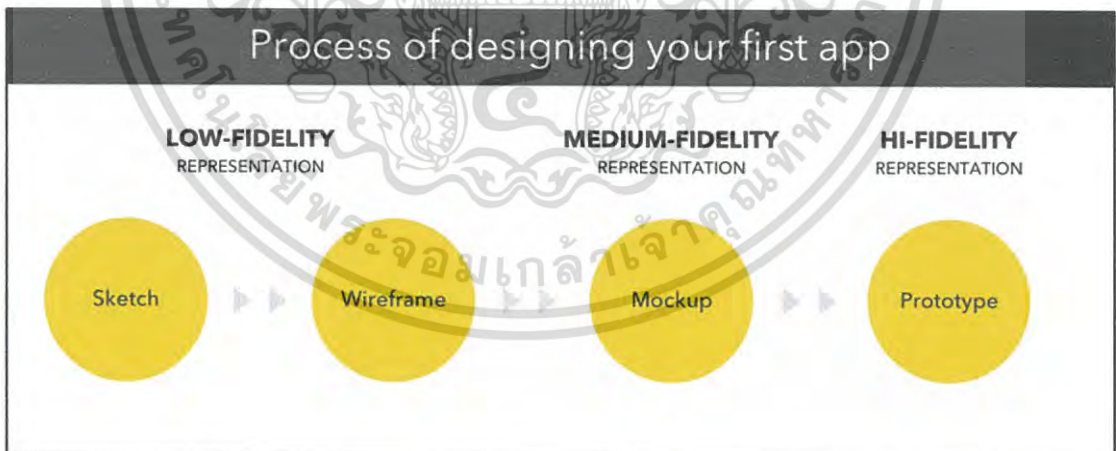


ภาพที่ 2.52 แท็บเครื่องมือ (Toolbars)

แท็บเครื่องมือ (Toolbars) ทำหน้าที่แสดงหัวข้อ (Title) ในหน้านั้นๆ ของแอปพลิเคชัน รวมไปถึงแสดงผลเมนูหลักที่จำเป็นของแอปพลิเคชัน โดยซ่อนเมนูด้วยแฮมเบอร์เกอร์ไอคอน (hamburger icon)

2.5. กระบวนการในการสร้างยูเอกซ์ยูไอ (UX/UI) ให้กับแอปพลิเคชัน (Application)

เมื่อรวบรวมปัญหาและความต้องการ (requirement) ของลูกค้าได้ในระยะแรก แล้วนำสร้างเป็นแนวคิด (Concept Ideation) เรียบร้อยแล้ว ขั้นตอนการออกแบบและสร้างผลิตภัณฑ์จะเป็นไปตามขั้นตอนต่อไปนี้



ภาพที่ 2.53 กระบวนการการออกแบบแอปพลิเคชัน (Process of designing)

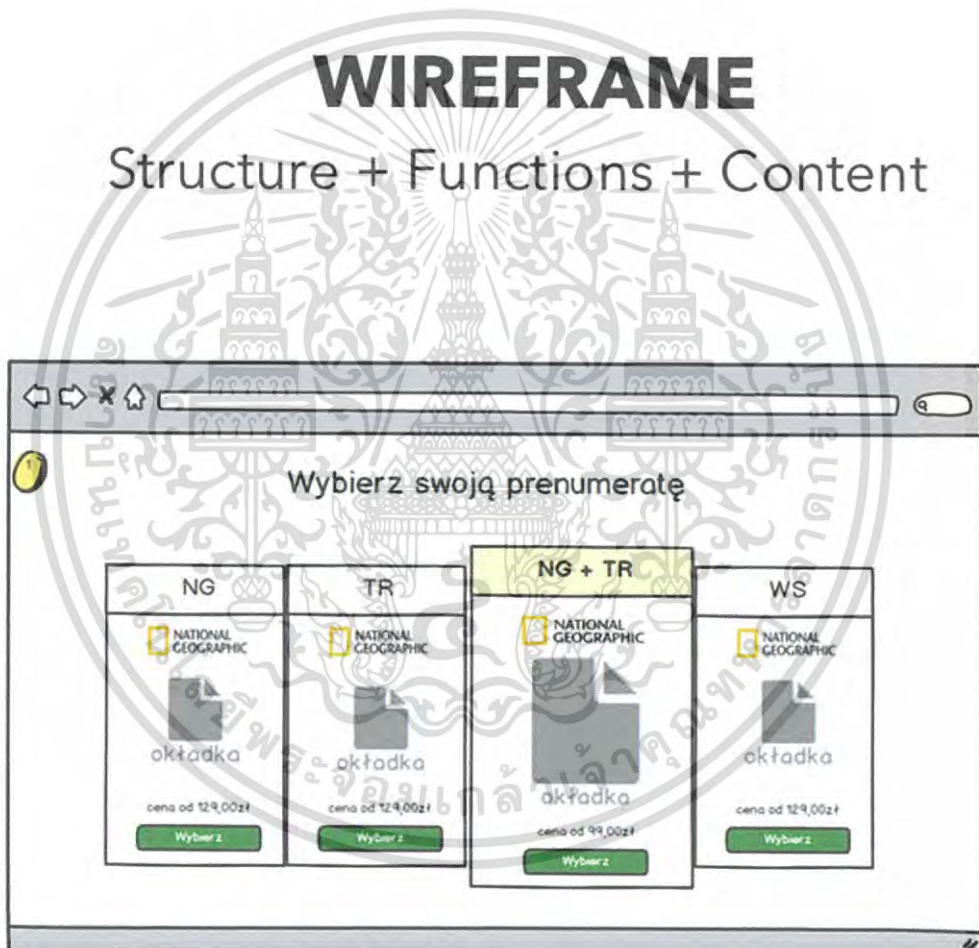
จากภาพที่ 2.53 แสดงให้เห็นถึงกระบวนการการออกแบบแอปพลิเคชันซึ่งประกอบไปด้วยสามช่วง ในช่วงแรกประกอบไปด้วยขั้นตอนการวาดลงบนกระดาษ (Sketch) และการสร้างแบบร่าง (Wireframe) ในช่วงที่สองคือขั้นตอนการสร้างแบบจำลองของแอปพลิเคชัน (Mockup) ขึ้นมา และช่วงสุดท้ายเป็นขั้นตอนการนำแบบจำลองแต่ละหน้าของแอปพลิเคชันมาทำให้สามารถใช้งานได้งานเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบื้องต้นเสมือนได้พัฒนาแอปพลิเคชันจริงเรียบร้อยแล้ว ซึ่งเรียกขั้นตอนนี้ว่า โปรโตไทป์ (Prototype) เพื่อนำไปทดสอบกับผู้ใช้งาน (Usability Testing)

2.5.1. วาดลงบนกระดาษ (Sketch)

การเริ่มต้นขึ้นโครงร่างของแอปพลิเคชันที่ง่ายที่สุดคือการวาดลงบนกระดาษเปล่า เพื่อจัดระบบของเนื้อหาและฟังก์ชันต่างๆ ว่าควรอยู่ที่หน้าใด เนื่องจากที่ต้องเริ่มต้นด้วยวิธีการวาด เพราะเป็นวิธีช่วยที่ระดมความคิดได้รวดเร็วที่สุด (fastest for brainstorming) เมื่อได้แนวความคิดที่หลากหลายมากพอให้กับแอปพลิเคชันแล้ว จะต้องมาเลือกแนวคิดที่เหมาะสมและสามารถพัฒนาได้จริงเพื่อทำการสร้างแบบร่าง (Wireframe) ต่อไป

2.5.2. สร้างแบบร่าง (Wireframe)



ภาพที่ 2.54 แบบร่าง (Wireframe)

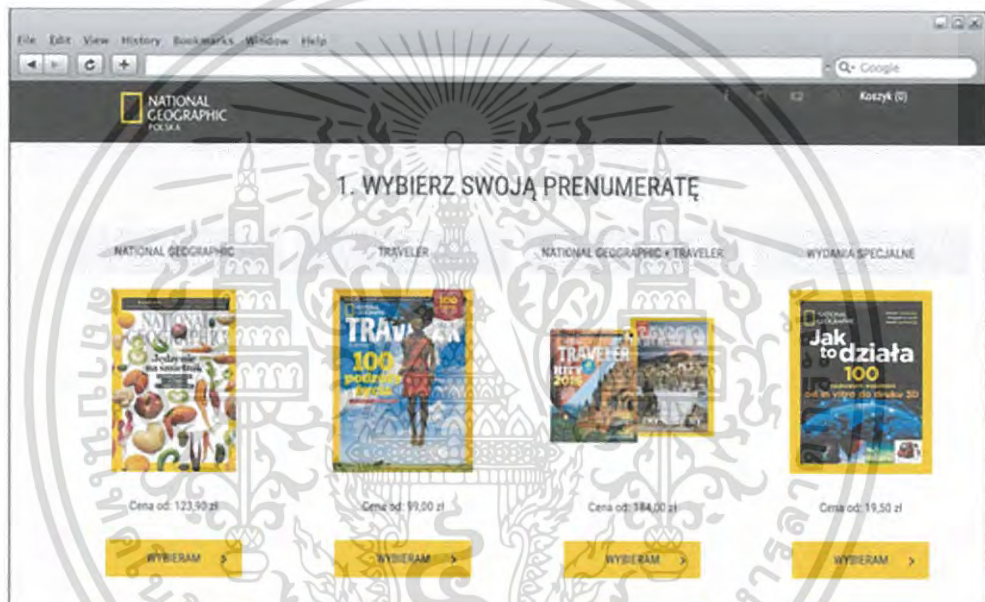
เมื่อผ่านขั้นตอนการวาดแนวคิดลงบนกระดาษมาแล้ว ต่อไปเป็นขั้นตอนคือการสร้างแบบร่าง (Wireframe) ซึ่งเป็นการวางโครงสร้างของแอปพลิเคชัน พร้อมกับกำหนดฟังก์ชันต่างๆ โดยขั้นตอนนี้จะมีการจัดวางยูเอไอ (UI) และวิเคราะห์เรื่องยูเอเอ็กซ์ (UX) ในเบื้องต้น เพื่ออธิบายหลักการ ทำงานของผลิตภัณฑ์ ความสัมพันธ์ของแต่ละหน้าจอในแอปพลิเคชัน รวมไปถึงการใส่ข้อมูลเบื้องต้น ซึ่งขั้นตอนนี้จะไม่ใส่ใจเรื่องของการออกแบบผลิตภัณฑ์ให้เกิดความสวยงาม แต่จะใช้สีหลักของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลิตภัณฑ์เพียงหนึ่งถึงสองสีเท่านั้นเพื่อให้ลูกค้าสามารถมองภาพออกในตอนไปนำเสนอเพื่อรับฟังคำแนะนำ

2.5.3. สร้างแบบจำลอง (Mockup)

MOCKUP

Style + Colours + Right Content



ภาพที่ 2.55 แบบจำลอง (Mockup)

ขั้นตอนต่อจากการสร้างแบบร่าง (Wireframe) คือการสร้างแบบจำลอง (Mockup) แอปพลิเคชันขึ้นมา ซึ่งการจะเริ่มทำขั้นตอนนี้ได้จะต้องผ่านการแก้ไขแบบร่างตามคำแนะนำของลูกค้ามาก่อน จึงจะสามารถเริ่มสร้างแบบจำลองต่อได้ โดยขั้นตอนนี้เป็นการสร้างหน้าจอของแอปพลิเคชัน (User Interface) ให้สมบูรณ์มากยิ่งขึ้น ได้แก่ การเพิ่มสี การใส่ภาพ การสร้างไอคอน การใส่โลโก้ของผลิตภัณฑ์ และการใส่ข้อมูลจริงลงไปในแอปพลิเคชัน ซึ่งขั้นตอนนี้จะเริ่มทำให้ลูกค้ามองเห็นรูปแบบของผลิตภัณฑ์มากขึ้น และเป็นขั้นตอนที่กินเวลาพอสมควร เนื่องจากเมื่อลูกค้าได้เห็นผลิตภัณฑ์ที่ใกล้เคียงกับของจริง ลูกค้าจะเริ่มมีไอเดียและมีความต้องการเพิ่มมากขึ้น ทำให้ขั้นตอนนี้มีการแก้ไขบ่อยครั้งและใช้เวลาในการพัฒนานานกว่าขั้นตอนอื่นๆ แต่ตามหลักแล้วนักออกแบบจะแก้ไขให้เฉพาะรูปแบบของแอปพลิเคชันเท่านั้น แต่จะไม่เปลี่ยนแปลงโครงสร้างของพีเจอร์แอปพลิเคชัน เพราะจะส่งผลให้เกิดความล่าช้าในการพัฒนา

2.5.4. โพรโตไทป์ (Prototype)



ภาพที่ 2.56 โพรโตไทป์ (Prototype)

โพรโตไทป์เป็นขั้นตอนสุดท้ายในการออกแบบผลิตภัณฑ์ที่สำคัญมากในการรวบรวมคำแนะนำและความต้องการของลูกค้าที่ดีที่สุดในปัจจุบัน แต่ก็ไม่ได้เป็นขั้นตอนที่จำเป็นต้องทำเสมอไป ขั้นตอนนี้เป็นการนำเอาแบบจำลอง (Mockup) ของแอปพลิเคชันมาสร้างให้เกิดการเคลื่อนไหวและเกิดการโต้ตอบกับผู้ใช้งานเพื่อทดสอบพฤติกรรมการใช้งานแอปพลิเคชันของผู้ใช้ (User Experience) โดยตรง เหตุที่ขั้นตอนนี้สามารถรวบรวมคำแนะนำและความต้องการของลูกค้าที่ดีที่สุด เพราะโพรโตไทป์ทำให้ลูกค้าหรือผู้ใช้งานรู้สึกเหมือนกับว่ากำลังใช้งานแอปพลิเคชันนี้จริง ซึ่งในบางองค์กรมองว่าขั้นตอนนี้สำคัญเพราะเป็นขั้นตอนที่ช่วยลดความเสี่ยงในการพัฒนาแอปพลิเคชันจริงได้ดี

2.6. แอนดรอยด์ (Android)

แอนดรอยด์ (Android) เป็น ระบบปฏิบัติการสำหรับอุปกรณ์พกพา เช่น โทรศัพท์มือถือ แท็บเล็ตคอมพิวเตอร์ เน็ตบุ๊ก ทำงานบนลินุกซ์ เคอร์เนลด์ จากนั้นบริษัทแอนดรอยด์ถูกซื้อโดย กูเกิล (Google) และนำแอนดรอยด์ไปพัฒนาต่อ

แอนดรอยด์ ยัง เป็นระบบปฏิบัติการแบบเปิด (Open Source) ทำให้นักพัฒนาที่สนใจสามารถนำซอร์ฟแวร์ต้นฉบับ มาศึกษาได้อย่างไม่ยาก

2.6.1. พัฒนาการของแอนดรอยด์และรุ่น

ประวัติรุ่นของแอนดรอยด์ เริ่มตั้งแต่การเผยแพร่รุ่นทดลองในเดือนพฤศจิกายน พ.ศ. 2550 และได้มีการเปิดตัวแอนดรอยด์ 1.0 ในเดือนกันยายน พ.ศ. 2551 ภายใต้การพัฒนาอย่างต่อเนื่องของ

กูเกิล และ โอเพนแฮนด์เซตอัลไลแอนซ์ ซึ่งจะเห็นได้จากการอัปเดตของระบบปฏิบัติการในช่วงแรกนับตั้งแต่เดือนเมษายน พ.ศ. 2552 รุ่นของแอนดรอยด์พัฒนาภายใต้โคเดนัม (ชื่อเฉพาะในการพัฒนา) และเผยแพร่โดยเรียงตามตัวอักษรภาษาอังกฤษ

ตารางที่ 2.2 ตารางรุ่นต่างๆ ของแอนดรอยด์และจำนวนผู้ใช้งานในแต่ละรุ่น

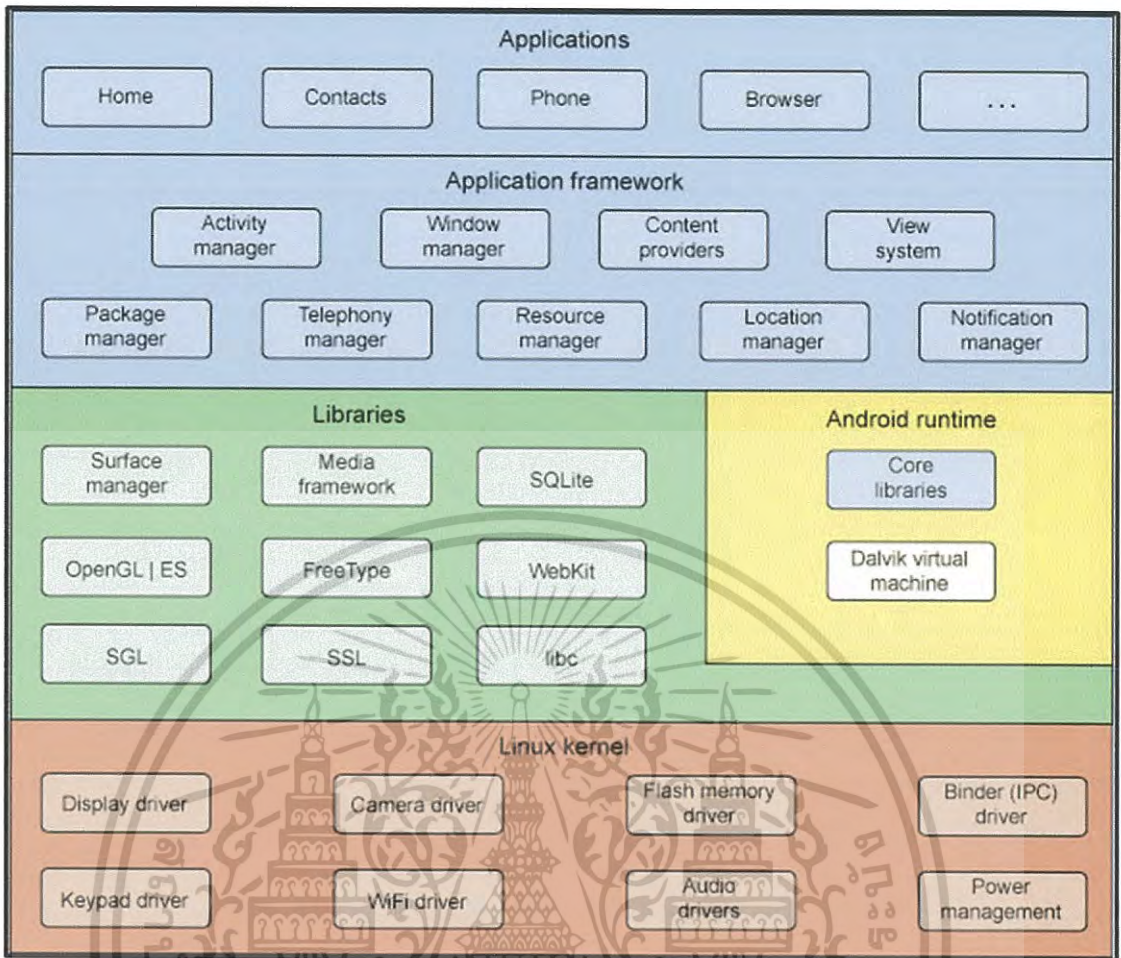
Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.3%
4.1.x	Jelly Bean	16	4.9%
4.2.x		17	6.8%
4.3		18	2.0%
4.4	KitKat	19	25.2%
5.0	Lollipop	21	11.3%
5.1		22	22.8%
6.0	Marshmallow	23	24.0%
7.0	Nougat	24	0.3%

ข้อมูลเมื่อวันที่ 7 พฤศจิกายน 2559

รุ่นอื่นๆที่มีจำนวนผู้ใช้งานน้อยกว่า 0.1 เปอร์เซ็นต์ จะไม่ถูกแสดง

2.6.2. สถาปัตยกรรมของแอนดรอยด์

นักพัฒนาแอนดรอยด์มีสิ่งจำเป็นที่จะต้องรู้คือ โครงสร้างของระบบปฏิบัติการแอนดรอยด์โดยโครงสร้างดังกล่าวจะถูกแบ่งออกเป็นส่วนๆ ดังนี้



ภาพที่ 2.57 สถาปัตยกรรมของแอนดรอยด์

1.) ชั้นแอปพลิเคชัน (Applications)

ส่วนแอปพลิเคชันหรือส่วนของโปรแกรมที่มีมากับระบบปฏิบัติการที่มีมาอยู่แล้ว หรือจากการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่างๆ ได้โดยตรง ซึ่งการทำงานจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนไว้ในโปรแกรม

โครงสร้างของแอปพลิเคชัน คือส่วนที่มีขึ้นเพื่อให้เกิดความสะดวกแก่ผู้พัฒนาโปรแกรมรวมถึงความมีประสิทธิภาพของโปรแกรม การที่ผู้พัฒนาโปรแกรมได้รู้โครงสร้างของแอปพลิเคชันจะช่วยให้เกิดการทุ่มเวลาและมีความเข้าใจมากยิ่งขึ้น

2.) ชั้นแอปพลิเคชันเฟรมเวิร์ค (Application Framework)

เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมากๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งานแอปพลิเคชันเฟรมเวิร์คในส่วนที่ต้องการใช้งาน แล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น

- ส่วนจัดการแอกทิวิตี (Activities Manager) เป็นส่วนจัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม (Activity)

- คอนเทนต์โพรไวเดอร์ (Content Providers) เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้ เช่น สามารถพัฒนาโปรแกรมเพื่อไปดึงข้อมูลรายชื่อที่อยู่บนสมุดโทรศัพท์ได้
- ระบบการแสดงผล (View System) ประกอบด้วยส่วนติดต่อผู้ใช้ (user interface) ชนิดต่างๆที่ใช้ในการพัฒนาโปรแกรมเช่น รายการ ตาราง กล่องข้อความ ปุ่ม รวมไปถึงเหตุการณ์และเว็บเบราว์เซอร์
- ส่วนจัดการทรัพยากร (Resource Manager) เป็นตัวจัดการในการเข้าถึงข้อมูลที่เป็น ข้อความ, รูปภาพ

3.) ชั้นไลบรารี (Libraries)

เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการทำงาน เช่น Surface Manage จัดการเกี่ยวกับการแสดงผล, Media Framework จัดการเกี่ยวกับการการแสดงผลภาพและเสียง, SQLite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

4.) ชั้นแอนดรอยด์รันไทม์ (Android Runtime)

จะมีดาโลวิคเวอร์ชวลแมชชีน (Dalvik Virtual Machine) ที่ถูกออกแบบมาเพื่อให้ทำงานบนอุปกรณ์ที่มีหน่วยความจำ (Memory), หน่วยประมวลผลกลาง (CPU) และพลังงาน (Battery) ที่จำกัด ซึ่งการทำงานของดาโลวิคเวอร์ชวลแมชชีนจะทำการแปลงไฟล์ที่ต้องการทำงานไปเป็นไฟล์ .DEX ก่อนเพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับหน่วยประมวลผลกลางที่มีความเร็วไม่มาก ส่วนต่อมาคือ Core Libraries ที่เป็นส่วนรวบรวมคำสั่งและชุดคำสั่งสำคัญ โดยถูกเขียนด้วยภาษาจาวา (Java Language)

5.) ลินุกซ์เคอร์เนล (Linux Kernel)

เป็นส่วนที่ทำหน้าที่หัวใจสำคัญ ในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 26. Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.6.3. ส่วนประกอบของแอปพลิเคชัน

2.6.3.1. แอคทิวิตี (Activity)

เป็นส่วนของการติดต่อกับผู้ใช้ โดยมีการแสดงผลออกมาเพื่อให้ผู้ใช้งานได้เห็นและได้ใช้งาน ในแต่ละแอปพลิเคชันอาจจะมีแอคทิวิตี (Activity) เดียวหรือหลายๆ แอคทิวิตีก็ได้ หรืออาจจะไม่มีแอคทิวิตีเลยก็ได้ ตัวอย่างเช่น รายการแสดงรายการเมนู นักพัฒนาสามารถเลือกให้รายการเมนูที่แสดงออกมามีภาพและคำบรรยายได้ภาพได้ ส่วนที่อยู่ในแอคทิวิตีนั้นเรียกว่า วิว (View) ซึ่งก็มีอยู่ในหลายรูปแบบ เช่น ปุ่ม (Buttons) ฟิลด์ข้อความ (Text Fields) แถบเลื่อน (Scroll Bars) รายการเมนู (Menu Items) ตรวจสอบกล่อง (Check Boxes) เป็นต้น

2.6.3.2. ผู้ให้บริการ (Service)

คือกลุ่มของข้อมูลที่สร้างขึ้นจากแอปพลิเคชัน เพื่อให้แอปพลิเคชัน อื่นๆ ได้นำไปใช้ โดยการจัดเก็บข้อมูลผู้ให้บริการเนื้อหา (Content Provider) นั้นจะอยู่ในลักษณะของไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลและอื่นๆ ตัวอย่างเช่น แอปพลิเคชันที่ใช้งานผู้ให้บริการเนื้อหาที่เห็นได้ชัดเจนที่สุดคือ โปรแกรมที่แสดงรายชื่อในการติดต่อ เป็นต้น

2.6.3.3. ระบบรับทราบเหตุการณ์ (Content Provider)

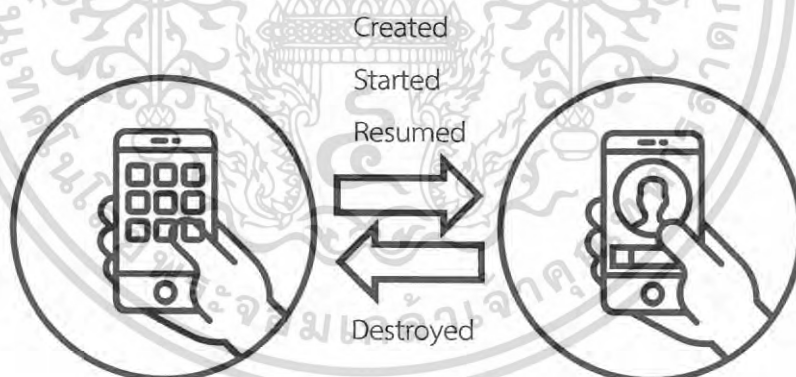
คือกลุ่มของข้อมูลที่สร้างขึ้นจากแอปพลิเคชัน เพื่อให้แอปพลิเคชัน อื่นๆ ได้นำไปใช้ โดยการจัดเก็บข้อมูลผู้ให้บริการเนื้อหา (Content Provider) นั้นจะอยู่ในลักษณะของไฟล์ ฐานข้อมูลและอื่นๆ ตัวอย่างเช่น แอปพลิเคชันที่ใช้งานผู้ให้บริการเนื้อหาที่เห็นได้ชัดเจนที่สุดคือ โปรแกรมที่แสดงรายชื่อในการติดต่อ เป็นต้น

2.6.3.4. ผู้ให้บริการด้านข้อมูล (Broadcast Receiver)

คือส่วนที่ใช้สำหรับคอยรับและตอบสนองต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น เมื่อแบตเตอรี่ต่ำ ผู้ใช้ทำการเปลี่ยนภาษา มีการโทรออก มีข้อความเข้า และอื่นๆ ถึงแม้การ รับส่ง ข้อมูล (Broadcast Receiver) จะไม่มีส่วนของการแสดงผล แต่ก็สามารถที่จะเรียกแอคทिवิตี ขึ้นมา แสดงผลให้ผู้ใช้ได้ หรืออาจเป็นตัวแจ้งเตือนในรูปแบบ การสั่น การแสดงไฟกระพริบ

2.6.4. วงจรชีวิตของแอปพลิเคชัน

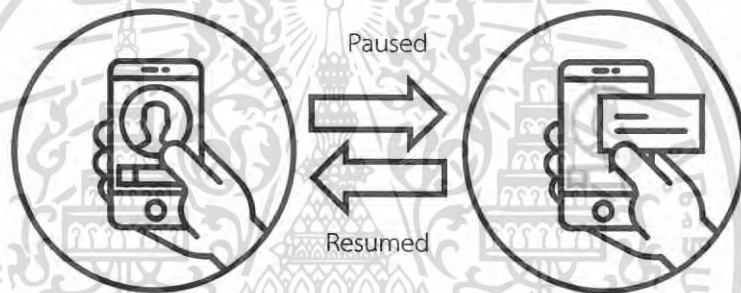
โดยปกติแอปพลิเคชันจะทำงานแยกกัน ในแต่ละโพรเซส และแต่ละโพรเซสอาจจะมีแอคทिवิตีหรือบริการ ทำงานอยู่มากกว่า 1 แอคทिवิตี หรือบริการ ดังนั้น ในแต่ละแอปพลิเคชัน อาจจะมี มากกว่า 1 แอคทिवิตี โดยในแต่ละแอคทिवิตีจะมีวงรอบชีวิต (Life Cycle) ที่แยกจากกันโดย ชัดเจน ซึ่งมี สถานะการทำงานหลักดังนี้



ภาพที่ 2.58 เข้าสู่แอปพลิเคชันและออกจากแอปพลิเคชัน

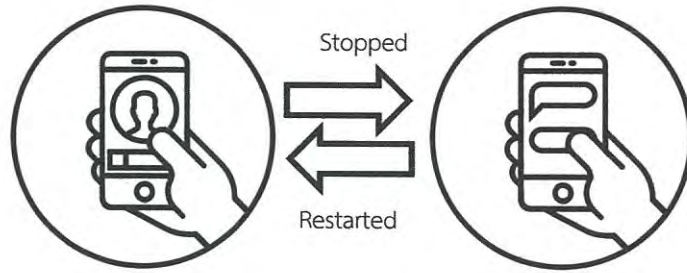
- onCreate (Bundle savedInstanceState) ส่วนนี้ จะถูกเรียกใช้งานเมื่อเปิด แอปพลิเคชันขึ้นมา แอปพลิเคชันเฟรมเวิร์คจะนำ Bundle object ที่บันทึกไว้ก่อน แอปฯ ถูกปิดมาคืนค่าต่างๆ ก่อนหน้าที่แอคทिवิตีจะทำงาน จากนั้นจะตามด้วย ฟังก์ชัน onStart() เช่น ผู้ใช้กดดูหน้าโปรไฟล์ของเพื่อนอยู่จากนั้นกดปุ่มโฮม เมื่อ ต้องการเข้าแอปพลิเคชันอีกครั้งก็จะคืนค่าหน้าโปรไฟล์เพื่อนมาให้

- onStart() ส่วนนี้เป็นการระบุว่าแอกทิวิตีนั้นๆ จะถูกแสดงขึ้นมาจากนั้น สถานะจะถูกย้ายไปเป็นสถานะ onResume แต่ถ้าแอกทิวิตีนั้นๆ ไม่สามารถทำงานได้ด้วยเหตุผลบางอย่างสถานะจะถูกย้ายไปเป็นสถานะ onStop
- onResume() ส่วนนี้จะถูกเรียกเมื่อแอกทิวิตีนั้นๆ มีการติดต่อกับผู้ใช้งาน เช่น นักพัฒนาต้องการเรียกแอกทิวิตีนั้นๆ ขึ้นมาทำงานอีกรอบหนึ่ง หลังจากที่แอกทิวิตีนั้นๆ อยู่ในสถานะ onPause
- onDestroy() ส่วนนี้จะถูกเรียกเมื่อมีการปิดการทำงานของแต่ละแอกทิวิตี เช่น ผู้ใช้ออกจากแอปพลิเคชัน ดังภาพที่ 2.58
- onPause() ส่วนนี้จะถูกเรียกใช้เมื่อแอกทิวิตีนั้นๆ จะถูกเปลี่ยนไปเป็นการทำงานทางเบื้องหลัง (Background) เช่น ขณะใช้งานหน้าโปรไฟล์ ไดอะล็อกสำหรับตอบข้อความก็เด้งขึ้นมา หน้าโปรไฟล์จึงเปลี่ยนไปทำงานทางเบื้องหลัง และเมื่อกดปิดกล่องข้อความนั้น หน้าโปรไฟล์จะเข้าสู่สถานะ onResume ดังภาพที่ 2.59



ภาพที่ 2.59 การเปลี่ยนสถานะ onPause และ onResume

- onStop() ส่วนนี้จะถูกเรียกใช้งานเมื่อผู้ใช้ไม่ต้องการใช้งานแอกทิวิตีนั้นๆ ในระยะเวลาหนึ่งๆ เช่น ผู้ใช้ใช้งานหน้าโปรไฟล์แล้วได้กดปุ่มเปลี่ยนไปยังหน้าข้อความ ซึ่งผู้ใช้จะไม่เป็นหน้าโปรไฟล์ขณะที่ใช้งานหน้าข้อความอยู่ ดังภาพที่ 2.60
- onRestart() ส่วนนี้จะเป็นการระบุว่าแอกทิวิตีนั้นๆ จะถูกแสดงขึ้นมาอีกครั้งหนึ่ง ซึ่งจะตามด้วยสถานะ onStart() เช่น เมื่อหน้าโปรไฟล์อยู่ในสถานะ onStop อยู่ ผู้ใช้ซึ่งใช้งานหน้าข้อความต้องการเปลี่ยนไปยังหน้าโปรไฟล์ ก็จะไปเข้าสู่สถานะ onRestart ดังภาพที่ 2.60

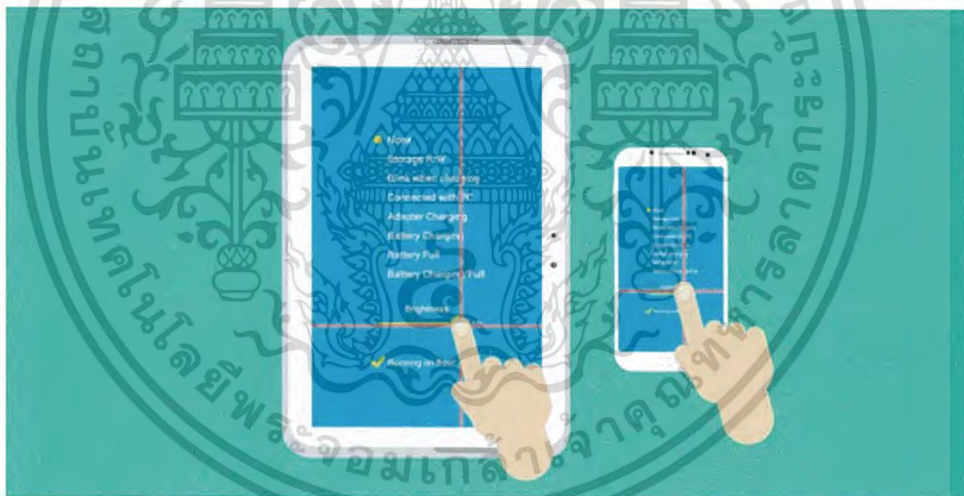


ภาพที่ 2.60 การเปลี่ยนสถานะ onStop และ onStart

2.6.5. แฟร็กเมนต์ (fragment)

แฟร็กเมนต์ (Fragment) นั้นถูกนำเสนอขึ้นมาตั้งแต่ 3.0 (API 11) ซึ่งเป็นยุคเริ่มต้นของแท็บเล็ต (Tablet) ที่มีหน้าจอขนาดใหญ่ขึ้น การใช้งานแตกต่างไปจากมือถือ ดังนั้นในการออกแบบ UI แบบเดิมๆที่เคยมีอยู่ก็จะเปลี่ยนแปลงไปสิ้นเชิง

ดังนั้นจึงมีนักพัฒนาหลายๆคนที่ใช้วิธีสเกล จากหน้าจอแบบมือถือให้พอดีกับแบบแท็บเล็ตแทน เพื่อที่ว่าจะได้ไม่ต้องแก้ไขอะไรมาก



ภาพที่ 2.61 การสเกลหน้าจอมือถือให้พอดีกับแท็บเล็ต

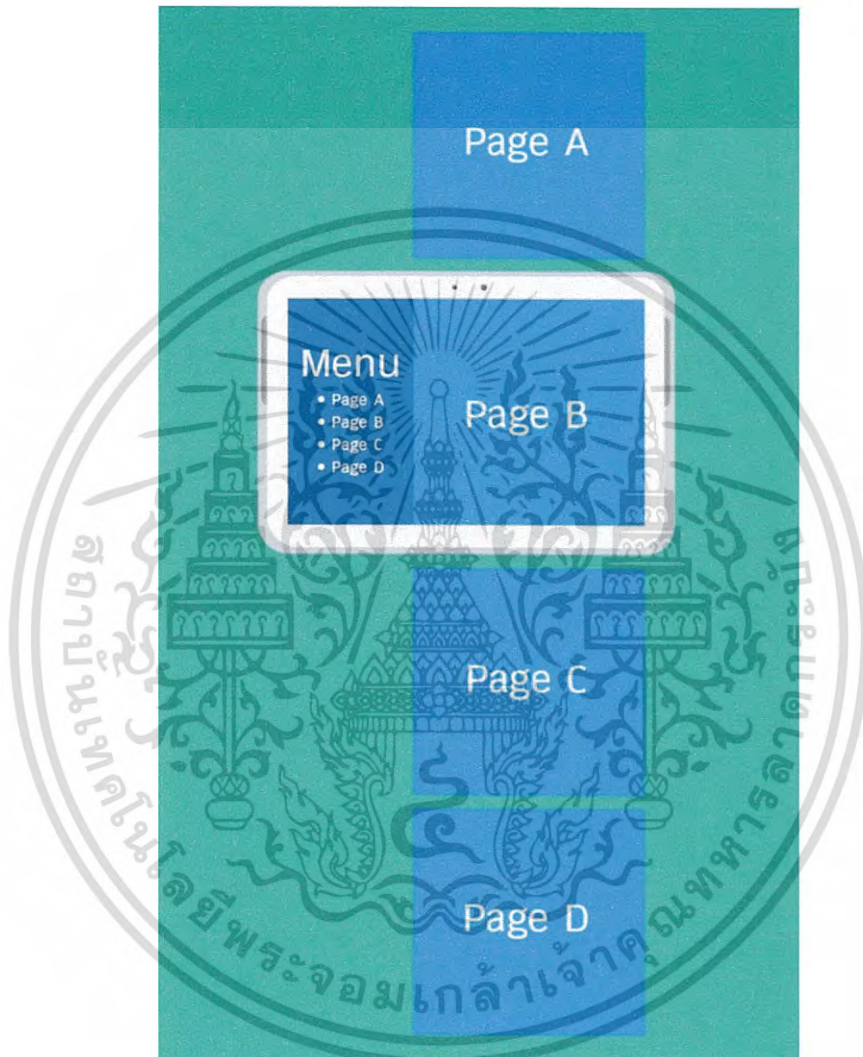
ถึงแม้จะแก้ปัญหาเรื่องหน้าจอได้แต่ก็ไม่ใช่วิธีที่ถูกต้อง เพราะยังสเกลให้ใหญ่ขึ้น ยูสเซอร์อินเตอร์เฟซก็จะมีขนาดใหญ่ตาม จึงเกิดพื้นที่ไร้ประโยชน์เยอะมาก ผู้ใช้ต้องขยับนิ้วไปมาบนหน้าจอไกลขึ้นกว่าเดิม

ดังนั้นเพื่อให้เกิดประโยชน์สูงสุดสำหรับแท็บเล็ตจึงต้องเปลี่ยนวิธีดีไซน์หน้าตาของแอปฯให้เหมาะสมกับแต่ละอุปกรณ์ยิ่งขึ้น โดยที่สร้างแอปพลิเคชันแค่ตัวเดียว แต่รันได้บนอุปกรณ์ทุกๆ แบบ ไม่ต้องทำแยกกันจึงนำแฟร็กเมนต์มาช่วยรองรับการทำงาน

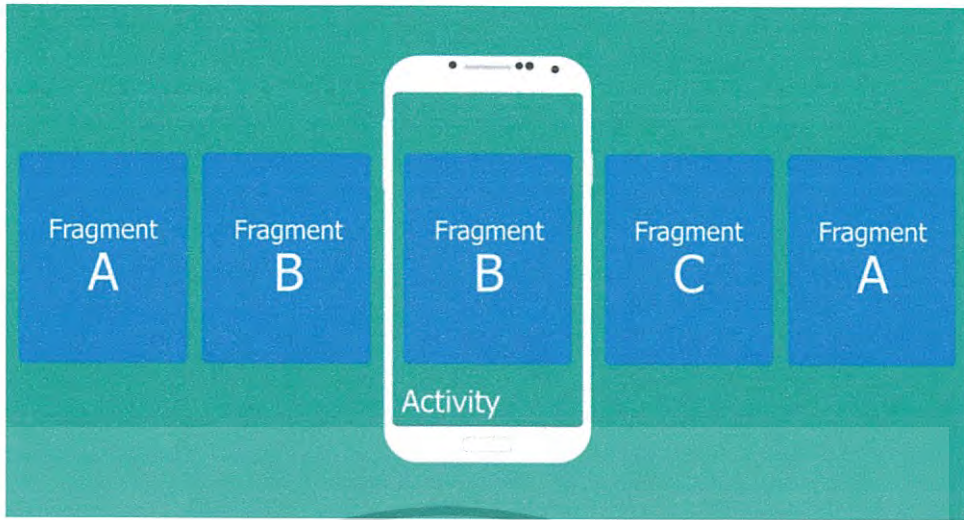
หลักการของแฟร็กเมนต์ คือการออกแบบยูสเซอร์อินเตอร์เฟซให้มีการทำงานลักษณะแยกออกเป็นส่วนๆ คล้ายๆกับแบ่งออกเป็นช่องหน้าต่างด้านซ้ายและด้านขวา แต่ละช่อง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ในวงจำกัด ไม่อนุญาตให้เผยแพร่ไปยังบุคคลอื่นโดยไม่ได้รับอนุญาต

เรียกว่า หนึ่งแฟร็กเมนต์ โดยแต่ละแฟร็กเมนต์จะมีความทำงานเป็นของตัวเอง คล้ายๆกับเป็น แอคทิวิตีย่อย (Sub-activity) สองตัวแยกกัน ทำงานอยู่ ภายในหนึ่งแอคทิวิตีหลัก (main activity) เช่น แอปพลิเคชันสำหรับอ่านข่าว ก็จะมีแฟร็กเมนต์ด้านซ้ายหน้าที่ทำงานคือลิสต์รายชื่อหัวข้อข่าว ส่วนช่องแฟร็กเมนต์ ด้านขวาทำหน้าที่เปิดดูรายละเอียดข่าวตามที่ใช้และที่หัวข้อข่าวด้านซ้าย ซึ่งทั้ง 2 แฟร็กเมนต์ทำงานแยกกันคนละแอคทิวิตีแต่สามารถรับส่งข้อมูลหากันได้



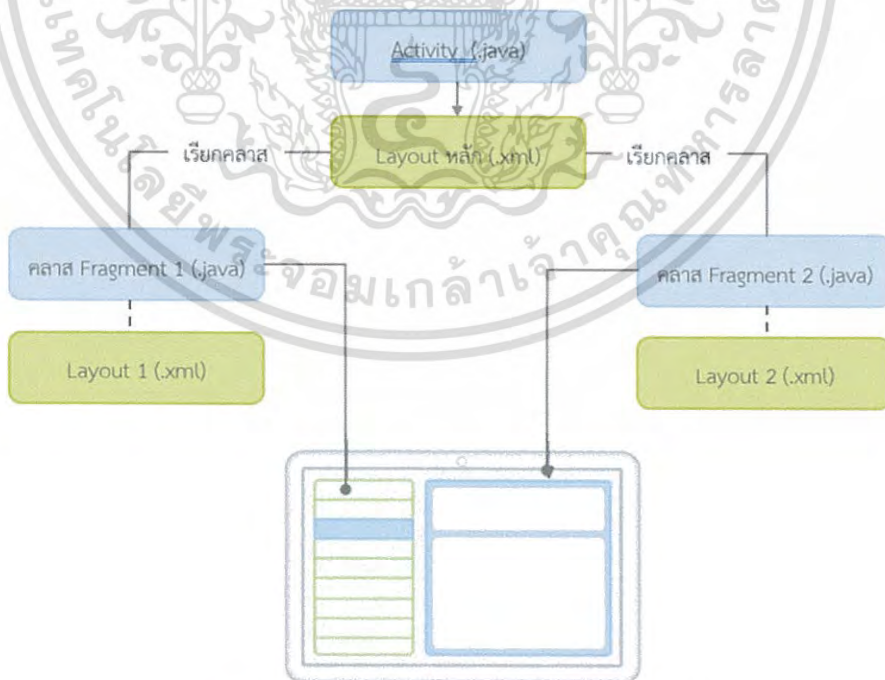
ภาพที่ 2.62 ตัวอย่างการจัดเลย์เอาต์ที่เหมาะสมกับอุปกรณ์แท็บเล็ต



ภาพที่ 2.63 ตัวอย่างแฟร็กเมนต์บนมือถือ

2.6.5.1. โครงสร้างแฟร็กเมนต์

แฟร็กเมนต์ ส่วนใหญ่มักออกแบบให้ทำงานหน้าที่ต่างกัน เช่น ช่องซ้าย ลิสต์ ช่องขวาเปิดดูรายละเอียด ดังนั้น แต่ละแฟร็กเมนต์ จะมีคลาส (โปรแกรม java ควบคุมการทำงาน) และมีเลย์เอาท์ แยกกันโดยมีแอกทิวิตีหลัก เช่น MainActivity.java ควบคุมอยู่ด้านบนสุดอีกทีหนึ่ง เท่ากับว่า อย่างน้อยสุดในแอปพลิเคชันที่ออกแบบใช้แฟร็กเมนต์จะต้องมี 3 คลาส ดังภาพที่ 2.64

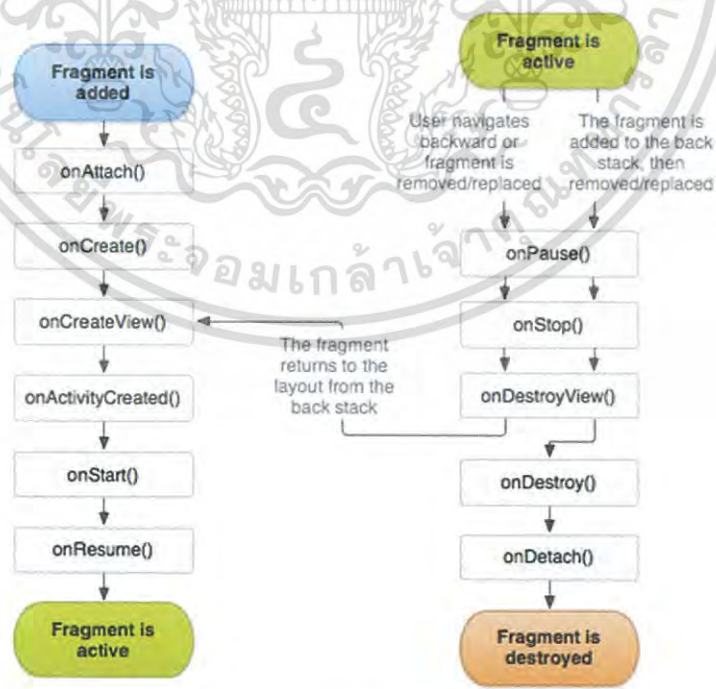


ภาพที่ 2.64 โครงสร้างการทำงานแฟร็กเมนต์

2.6.5.2. วงจรชีวิตแฟร็กเมนต์

Fragment แต่ละตัวให้ทำงานต่างกันได้เต็มที่ และมี Life Cycle ที่ช่วยให้ Manage การทำงานหลายอย่างได้สะดวกขึ้น

- onAttach จะถูกเรียกเมื่อแอกทิวิตีเรียกใช้งานแฟร็กเมนต์นั้นๆ (ผูกแฟร็กเมนต์เข้ากับแอกทิวิตี)
- onCreate จะถูกเรียกเมื่อทำการสร้างแฟร็กเมนต์ขึ้นมา (Initial)
- onCreateView จะถูกเรียกเมื่อกำหนดเลย์เอาท์ที่จะใช้กับแฟร็กเมนต์ (ผูกเลย์เอาท์เข้ากับแฟร็กเมนต์)
- onActivityCreated แอกทิวิตีถูกสร้างขึ้นและผูกแฟร็กเมนต์เข้ากับแอกทิวิตีเรียบร้อยแล้ว
- onStart แฟร็กเมนต์แสดงขึ้นมาให้เห็นบนหน้าจอเรียบร้อยแล้ว
- onResume แฟร็กเมนต์พร้อมสำหรับติดต่อกับผู้ใช้
- onPause แฟร็กเมนต์หยุดติดต่อกับผู้ใช้
- onStop แฟร็กเมนต์หยุดแสดงให้เห็นบนหน้าจอ
- onDestroyView เคลียร์รีซอร์ส (Resource) ที่เกี่ยวข้องกับการแสดงผล
- onDestroy เคลียร์ข้อมูลทั้งหมดของแฟร็กเมนต์นั้นๆ
- onDetach แอกทิวิตีเลิกใช้งานแฟร็กเมนต์นั้นๆ (แฟร็กเมนต์กับแอกทิวิตีไม่เชื่อมต่อกันแล้ว)



ภาพที่ 2.65 วงจรชีวิตของแฟร็กเมนต์

2.7. ภาษาจาวา

จาวา คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซัน ไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษาซีพลัสพลัส (C++) โดยรูปแบบที่เพิ่มเติมขึ้น คล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตาม ต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้วภายหลังจึงเปลี่ยนไปใช้ชื่อ “จาวา” ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษาจาวา อยู่ที่ผู้เขียนโปรแกรมสามารถใช้หลักการของโปรแกรมเชิงวัตถุ (Object-Oriented Programming) มาพัฒนาโปรแกรมของตนด้วยจาวาได้

โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้นคลาสคือที่เก็บเมทอด (Method) หรือ พฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

2.7.1. ความหมาย

องค์ประกอบหลัก แบ่งเป็น 3 ส่วน ดังนี้

2.7.1.1. Java Virtual Machine (JVM)

เป็นเทคโนโลยีจาวาที่ทำหน้าที่เป็นตัวแปลภาษา (interpreter) โดยจะทำการแปลจาวาไบต์โค้ด (JAVA Bytecode) ให้เป็นภาษาที่เครื่องเข้าใจและสามารถรันได้หลายแพลตฟอร์ม (Platform)

2.7.1.2. Java Runtime Environment (JRE)

เป็นส่วนที่รับโปรแกรมไบต์โค้ดที่แปลมาจาก JVM มาทำการรันโปรแกรม โดยจะทำงาน 3 ขั้นตอน ดังนี้

- 1.) โหลดไบต์โค้ด ขั้นตอนนี้จะเป็นการโหลดคลาสทุกคลาสที่เกี่ยวข้องในการรันโปรแกรม โดยใช้ Class Loader
- 2.) ตรวจสอบไบต์โค้ดโดยใช้ Byte code Verifier คือการตรวจสอบว่าโปรแกรม จะต้องไม่มีคำสั่งที่ทำให้เกิดความผิดพลาดกับระบบ เช่น การแปลงข้อมูลที่ผิดพลาด หรือมี การแทรกแซงเข้าสู่ระบบภายใน เป็นต้น
- 3.) รันไบต์โค้ด โดยใช้ Runtime Interpreter

2.7.1.3. Java 2 Software Developer Kit (J2SDK)

Java 2 Software Developer Kit (J2SDK) เป็นชุดพัฒนาโปรแกรม ภาษาจาวา เดิมเรียกว่า JDK ต่อมาเปลี่ยนชื่อเป็นจาวา 2 ประกอบไปด้วยโปรแกรมต่างๆ เช่น โปรแกรมคอมไพเลอร์ (javac.exe) โปรแกรมอินเทอร์พรีเตอร์ (java.exe) และโปรแกรมดีบักเกอร์ แต่ชุดโปรแกรม JDK จะเป็นเพียงโปรแกรมย่อยของโปรแกรมประเภท SDK เนื่องจาก JDK จะไม่มี โปรแกรมอีดิเตอร์ (Editor) สำหรับการเขียนซอร์สโค้ด (Source code) หรือนำโปรแกรมดั่งนั้นผู้ใช้ จะต้องใช้โปรแกรม SDK อื่นเพื่อช่วยในการเขียนซอร์สโค้ด

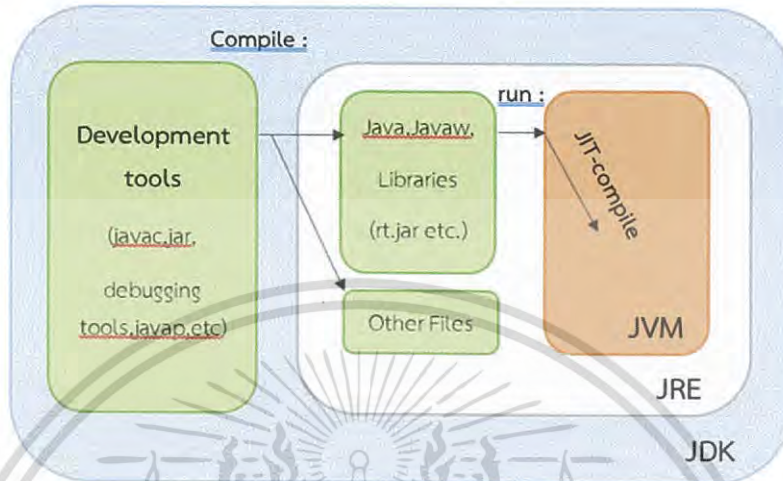
ส่วน SDK สำหรับการพัฒนาโปรแกรมภาษาจาวาที่มีอยู่ในปัจจุบันจะมีทั้ง ที่เป็นโปรแกรมแบบฟรีแวร์ (Freeware) หรือโปรแกรมเพื่อการค้า (Commercial software) เช่น

- NetBeans ของบริษัท ซันไมโครซิสเต็มส์
- Eclipse ของบริษัท IBM
- Jdeveloper ของบริษัท Oracle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- IntelliJ IDEA ของบริษัท Jet Brains

โปรแกรม SDK เหล่านี้จะมีเครื่องมือเพื่อช่วยในการสร้างโปรแกรมภาษา java ในรูปแบบต่างๆได้ง่ายขึ้น เช่น เครื่องมือช่วยในการออกแบบโปรแกรมกราฟฟิก



ภาพที่ 2.66 แสดงองค์ประกอบของภาษาจาวา

2.7.2. ขั้นตอนการทำงานภาษาจาวา

2.7.2.1. เขียนโปรแกรม

สร้างโปรแกรมซอร์สโค้ดด้วยการพิมพ์คำสั่งต่างๆ ตามหลักการเขียนโปรแกรมด้วยภาษาจาวาโดยใช้ editor หรือโปรแกรมที่สามารถพิมพ์ข้อความ และสามารถบันทึกไฟล์เป็นรหัสแอสกี (ASCII) หลังจากเขียนโปรแกรมเสร็จเรียบร้อยแล้วต้องทำการบันทึกข้อมูลเป็นไฟล์ที่มีชื่อเดียวกันกับชื่อคลาสของจาวา และใช้นามสกุลไฟล์เป็น .java

2.7.2.2. คอมไพล์โปรแกรม

แปลภาษาคอมไพเลอร์ระดับสูงหรือภาษาโปรแกรมที่มนุษย์เขียนขึ้นให้เป็นภาษาแอสเซมบลีหรือภาษาเครื่องที่เครื่องคอมพิวเตอร์เข้าใจ ตัวแปลภาษาแบ่งออกเป็น 2 แบบ

- 1.) คอมไพเลอร์ ตัวแปลภาษาประเภทนี้จะแปลชุดคำสั่งในซอร์สโค้ดทั้งหมดให้เป็นโปรแกรม executable code ตัวอย่างเช่นแปล source code ภาษา c จากโปรแกรม hello.c ให้เป็นโปรแกรม executable code ที่ชื่อ hello.exe โดยโปรแกรม executable code ที่ได้จากการแปลภาษาคอมไพเลอร์ระดับสูงจะสามารถทำงานได้อย่างรวดเร็วเนื่องจากโค้ดอยู่ในรูปของเลขฐานสองที่สามารถติดต่อกับเครื่องคอมพิวเตอร์ได้โดยตรง โดยคอมไพเลอร์ของภาษาจาวาคือ คอมไพเลอร์โดยจะทำการอ่านโปรแกรมภาษาจาวาทั้งหมดตั้งแต่ต้นจนจบแล้วทำการแปลผลทีเดียว
- 2.) อินเตอร์พรีเตอร์ ตัวแปลภาษาประเภทนี้จะแปลชุดคำสั่งของภาษาคอมไพเลอร์ระดับสูงทีละคำสั่งให้เป็นโปรแกรม executable code แล้วจะสั่งให้เครื่องคอมพิวเตอร์ทำงานทันทีโดยจะทำการอ่านและแปลโปรแกรมทีละบรรทัด เมื่อแปลผลบรรทัดหนึ่งเสร็จก็จะทำตามคำสั่งในบรรทัดนั้นแล้วจึงทำการแปลผลตามคำสั่งในบรรทัดถัดไป หลักการที่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นประโยชน์จะเขียนขึ้นใหม่

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์พรีเตอร์ใช้เรียกว่าอินเทอร์พรีต (interpret) การคอมไพล์โปรแกรมซอร์สโค้ดทำได้โดยการใช้คำสั่ง javac.exe ที่มากับการติดตั้ง JDK แล้วมีรูปแบบคำสั่งคือ javac FileName.java เมื่อ FileName.java คือชื่อไฟล์ใดๆ ที่มีนามสกุล .java ถ้าไม่มีชื่อผิดพลาดใดๆ ผลลัพธ์ที่ได้จากการคอมไพล์จะได้ไฟล์ไบนารีโค้ดที่ชื่อเดียวกับชื่อคลาส เช่น javac TestJava.java หลังจากการคอมไพล์จะได้ไฟล์ TeestJava.class สิ่งสำคัญในการคอมไพล์ไฟล์ซอร์สโค้ดคือ ต้องพิมพ์ชื่อไฟล์พร้อมนามสกุลเป็น .java เสมอและต้องพิมพ์ชื่อไฟล์ด้วยตัวอักษรตัวใหญ่หรือตัวเลขให้ถูกต้องตามการตั้งชื่อคลาส

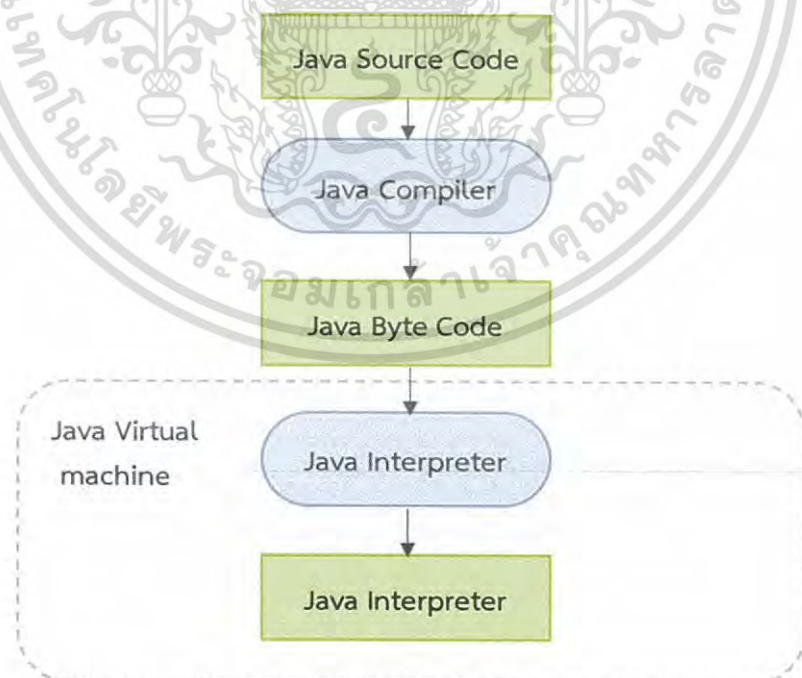
ดังนั้นการรันโปรแกรมเพียงแค่พิมพ์ชื่อไฟล์ไม่ต้องพิมพ์นามสกุลของไฟล์ และต้องพิมพ์ชื่อไฟล์ด้วยตัวอักษรใหญ่หรือตัวเล็กให้ถูกต้องตามชื่อคลาส ตัวอย่างเช่น java TestJava เมื่อ TestJava คือชื่อไฟล์ TestJava.class

2.7.2.3. รันโปรแกรม

ทำการรันโปรแกรมเพื่อดูผลลัพธ์ทางจอภาพ โดยการไฟล์ไบนารีโค้ดจากการใช้คำสั่ง javac.exe ที่มากับการติดตั้ง JDK แล้วซึ่งมีรูปแบบคำสั่งคือ java FileName เมื่อ FileName คือชื่อไฟล์ไม่ต้องมีนามสกุล ดังนั้นการรันโปรแกรมเพียงแค่พิมพ์ชื่อไฟล์ไม่ต้องพิมพ์นามสกุลของไฟล์ และต้องพิมพ์ชื่อไฟล์ด้วยตัวอักษรใหญ่หรือตัวเล็กให้ถูกต้องตามชื่อคลาส ตัวอย่างเช่น java TestJava เมื่อ TestJava คือชื่อไฟล์ TestJava.class

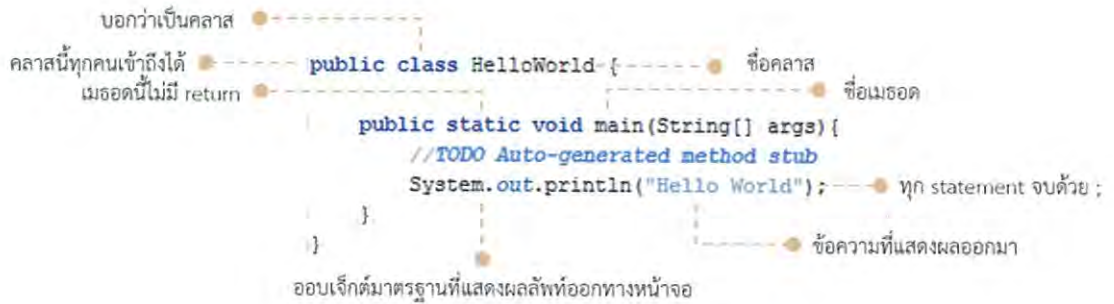
2.7.2.4. ประมวลผล

เมื่อนำ executable program จากขั้นตอนที่ 3 มาประมวลก็จะได้ผลลัพธ์ (output) ของโปรแกรมออกมา



ภาพที่ 2.67 ขั้นตอนการทำงานภาษาจาวา

2.7.3. รูปแบบการเขียนโปรแกรมภาษาจาวา



ภาพที่ 2.68 รูปแบบการเขียนโปรแกรมจาวา

ภาษาจาวามีรูปแบบการเขียนโปรแกรมที่ประกอบด้วยคลาสอย่างน้อยหนึ่งคลาสที่มีเมธอด `main()` ซึ่งเปรียบเสมือนจุดเริ่มต้นของโปรแกรม โดยมีรูปแบบการเขียนโปรแกรมดังภาพที่ 2.68 ซึ่งสามารถอธิบายได้ดังนี้

- 1.) เป็นการสร้างหรือนิยามคลาสขึ้นในชื่อ `HelloWorld` ชื่อไฟล์จึงต้องเป็น `HelloWorld.java`
- 2.) คลาส `HelloWorld` มีเมธอด `main()` เมื่อ execute โปรแกรมนี้ JRE จะค้นหา เมธอดที่มีชื่อ `main()` เพื่อเริ่มประมวลผลตามชุดคำสั่งที่เขียนไว้
- 3.) เมธอด `main()` จะต้องมีคีย์เวิร์ด `public`, `static` และ `void` อยู่ รวมไปถึงพารามิเตอร์ที่เป็นอาร์เรย์ ของข้อมูลประเภทข้อความ (`String`)
- 4.) ชุดคำสั่งที่ต้องการให้เกิดการประมวลผล จะเขียนไว้ในเครื่องหมายวงเล็บ `{}` ของเมธอด `main()`
- 5.) ทุกคำสั่งต้องจบด้วยเครื่องหมาย `;` เสมอ

2.8. ภาษาเอกซ์เอ็มแอล

เอกซ์เอ็มแอล (XML) เป็นภาษาที่ใช้ในการแสดงผลข้อมูล โดยสามารถกำหนดชื่อแท็ก (อิลิเมนต์) และชื่อแอตทริบิวต์ ได้ตามความต้องการของผู้สร้างเอกสารเอกซ์เอ็มแอลโดยเอกสารนั้นจะต้องมีความเป็น Well-formed เอกสาร XML เป็นแค่แท็กซีไฟล์ชนิดหนึ่ง ที่มีแท็กเปิดและแท็กปิดครอบข้อมูลไว้ตรงกลาง ทำให้เอกสาร XML ถูกใช้ในการติดต่อกับระบบที่ต่างกัน เนื่องจากความง่ายในการสร้างเอกสาร การนำเอกสาร XML ไปใช้งาน จะสนใจแต่ข้อมูลที่ถูกเน้นด้วยแท็กมากกว่า

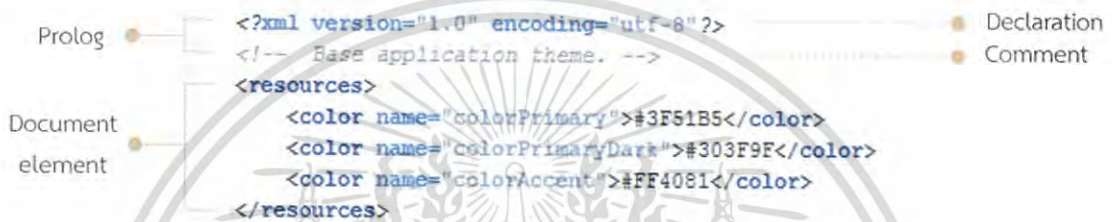
การสร้างเอกสารที่มีรูปแบบถูกต้อง (Well – formed XML Document) เป็นกฎที่ต้องปฏิบัติตามสำหรับการสร้างเอกสาร XML เมื่อสร้างเอกสาร XML ที่มีรูปแบบถูกต้อง จะทำให้สามารถกำหนดตำแหน่งต่าง ๆ ภายในเอกสาร และสามารถจัดการกับอิลิเมนต์ได้ตามต้องการ ตัวอย่างเช่น เอกสาร XML ต้องเริ่มต้นด้วย `<?xml version="1.0" ?>` เอกสาร XML 1 เอกสาร จะต้องมีแท็กรูทเพียงแท็กเดียว หมายความว่า แท็กและข้อมูลต่างๆ จะต้องอยู่ภายในแท็กแรกสุดเพียงแท็กเดียว การเปิดและปิดแท็กจะต้องไม่มีการคร่อมกัน เช่น `ตัวหนา<i>และเอียง</i>`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก XML สามารถกำหนดชื่อแท็ก และชื่อแอตทริบิวต์ได้ตามความต้องการ ทำให้ในการเน้นข้อมูลใดข้อมูลหนึ่ง สามารถมีเอกสาร XML หลายรูปแบบ หากว่าเอกสาร XML นั้นถูกนำไปใช้ติดต่อกับระบบอื่นๆ อาจทำให้สื่อความหมายไม่ตรงกัน ดังนั้นจึงต้องมีการกำหนดรูปแบบที่เป็นมาตรฐานขึ้น โดยตัวกำหนดโครงสร้างของเอกสาร (DTD) และ สคีม่า (Schema)

2.8.1. โครงสร้างของเอกสารเอกซ์เอ็มแอล

XML ให้ความสำคัญกับโครงสร้างข้อมูล และไวยากรณ์ในการเขียนแท็กคำสั่ง ซึ่งเป็นส่วนในการกำหนดรูปแบบและตำแหน่งของแท็กคำสั่งต่างๆ ด้วย พิจารณาโครงสร้างของเอกสาร XML ดังภาพที่ 2.69



ภาพที่ 2.69 โครงสร้างเอกสารเอกซ์เอ็มแอล

โครงสร้างของเอกสาร XML ประกอบด้วย 2 ส่วนหลัก ดังนี้

- 1.) ส่วนของ Prolog อยู่ที่ตำแหน่งเริ่มต้นของเอกสาร XML ทำหน้าที่สั่งให้ XML Parser ทำการแปลความหมายของ Element และ Content ต่างๆ
- 2.) ส่วนของ Document Element หรือ Root Element เป็นส่วนที่บรรจุ Element เพิ่มเติมในเอกสาร XML ได้ โดยที่ RootElement จะประกอบด้วย Element ย่อยจำนวนมาก ซึ่งใน Element ย่อยเหล่านั้นจะบรรจุแท็กคำสั่ง และข้อความ (Text) ที่ต้องการแสดงผลบนหน้าเว็บเพจ

2.8.2. กฎพื้นฐานของ XML

กฎพื้นฐานในการเขียน XML (Well – Formed)

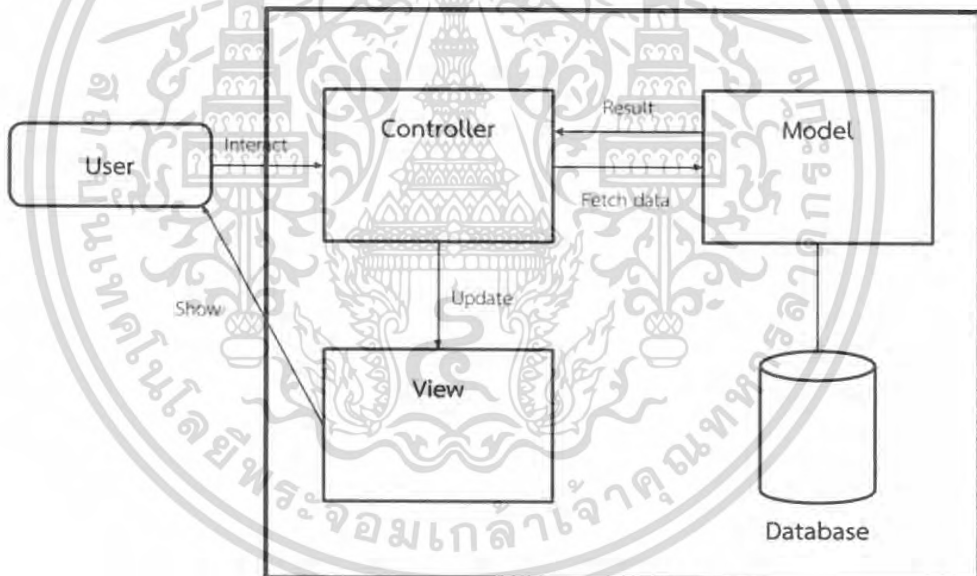
- 1.) Element ใน XML สามารถมีได้หลายตัว แต่จะต้องมี Element เพียง 1 ตัว ทำหน้าที่เป็น Root และ Element อื่นๆ จะทำหน้าที่เป็น Child
- 2.) การกำหนดแท็กทุกแท็กต้องมีแท็กปิดตามหลังเสมอ
- 3.) อักขรตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ใน XML จะถือว่า ต่างกัน (Case Sensitive)
- 4.) ค่าที่กำหนดให้กับแอททริบิวต์หรือ Element ใดๆ ต้องใส่เครื่องหมาย “ ” หรือ ‘ ‘ ครอบค่านั้น
- 5.) ส่วนของการประกาศ XML จะเรียกว่า Processing Instructions (PI) จะต้องมีการเขียนแท็กระบุเวอร์ชันที่เรียกใช้งาน ดังนี้ <?xml version = “1.0”?>
- 6.) ชื่อ Element ของเอกสาร XML สามารถเป็นอักขระ ตัวเลข และอักขระพิเศษอื่นๆ ได้ และห้ามเว้นช่องว่างระหว่างชื่อของ Element นั้น

2.9. แนวคิดการเขียนโปรแกรมแบบเอ็มวีซี

เนื่องจากการพัฒนาแอปพลิเคชันขึ้นมาให้ใช้งานได้นั้น ภายในจะประกอบไปด้วยไฟล์หลายๆไฟล์ซึ่งยังไม่รวมกับส่วนที่จะเพิ่มเติมในอนาคต ดังนั้นสิ่งสำคัญอย่างแรกของการสร้างโปรเจกต์ขึ้นมาคือการวางโครงสร้างของโค้ด (Code Structure)

เอ็มวีซี (MVC) คือ โครงสร้างโค้ดชนิดหนึ่ง ที่เขียนง่าย รูปแบบของเอ็มวีซีใช้เพื่อแยกส่วนซอฟต์แวร์ในส่วน ตรรกะเนื้อหา (domain logic) ได้แก่ความเข้าใจในระบบของผู้ใช้ และส่วนการป้อนข้อมูลและแสดงผล (GUI) ซึ่งช่วยให้การพัฒนา การทดสอบ และการดูแลรักษาซอฟต์แวร์แยกออกจากกัน ที่มีการแบ่งแยกระบบออกเป็น 3 ส่วนหลักๆ คือ

- 1.) โมเดล (Model) คือ ส่วนที่ใช้ในการติดต่อกับฐานข้อมูล ซึ่งทำหน้าที่ในการดึงข้อมูลขึ้นมาจากฐานข้อมูลขึ้นมา และทำการจัดการข้อมูลต่างๆ ไว้ในรูปแบบที่เหมาะสม
- 2.) วิว (View) คือส่วนที่จะนำข้อมูลจากโมเดลไปใช้แสดงผลให้ผู้ใช้งานเห็นผลลัพธ์ออกมาในยูสเซอร์อินเตอร์เฟส
- 3.) คอนโทรลเลอร์(Controller) คือส่วนที่จะคอยรับอินพุตจากผู้ใช้งานเข้ามาแล้วนำคำสั่งไปประมวลผล เพื่อสั่งงานวิวและ โมเดล ให้ประมวลผลออกมาเป็นอย่างไร



ภาพที่ 2.70 ขั้นตอนการทำงานของ MVC

2.9.1. ขั้นตอนการทำงานของเอ็มวีซี

จากรูปที่ 2.69 สามารถอธิบายการทำงานของเอ็มวีซีได้ดังนี้

- 1.) เริ่มจากผู้ใช้งานรีเควสให้กับแอปพลิเคชัน เช่น ผู้ใช้ต้องการคำนวณบางอย่าง ซึ่งจะถูกส่งต่อให้คอนโทรลเลอร์ทำการตรวจสอบข้อมูลที่มาให้
- 2.) แล้วคอนโทรลเลอร์จะเรียกเมธอดให้ทำงานเพื่อจัดการรีเควสนั้น
- 3.) โมเดลจะทำการคำนวณและอาจติดต่อกับฐานข้อมูลเพื่อจัดการกับรีเควสนั้น แล้วส่งผลลัพธ์กลับไปให้คอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4.) เมื่อคอนโทรลเลอร์ได้ผลลัพธ์จากโมเดลแล้วก็ใช้ผลลัพธ์นั้นส่งต่อให้วิทำงาน
- 5.) วิจะแสดงผลลัพธ์นั้นไปให้ผู้ใ้

2.10. การเขียนผังงาน (Flowchart)

ผังงาน (Flowchart) คือรูปภาพหรือสัญลักษณ์ที่ใช้เขียนแทนคำอธิบาย ข้อความหรือคำพูดที่ใช้ในอัลกอริทึม เพราะการนำเสนอขั้นตอนของงานให้เข้าใจตรงกัน ระหว่างผู้เกี่ยวข้อง ด้วยคำพูด หรือข้อความทำได้ยากกว่าการใช้รูปภาพหรือสัญลักษณ์ หางาน

สามารถแบ่งได้เป็น 2 ประเภทใหญ่คือ

- 1.) ผังงานระบบ (System Flowchart) คือ ผังงานที่แสดงขั้นตอนการทำงานในระบบอย่างกว้าง ๆ แต่ไม่เจาะลงในระบบงานย่อย
- 2.) ผังงานโปรแกรม (Program Flowchart) คือ ผังงานงานที่แสดงถึงขั้นตอนในการทำงานของโปรแกรม ตั้งแต่รับข้อมูล คำนวณ จนถึงแสดงผลลัพธ์








2.10.1. รูปแบบของผังงาน

การเขียนผังงานจะเป็นการถ่ายทอดความเข้าใจที่ได้จากการวิเคราะห์งานให้อยู่ในรูปภาพหรือสัญลักษณ์ผู้ที่เขียนโปรแกรมจะสามารถเข้าใจลำดับขั้นตอนการเขียนโปรแกรมได้รวดเร็วและง่ายขึ้น โดยดูจากการเชื่อมโยงของสัญลักษณ์ และยังช่วยให้ง่ายต่อการตรวจสอบความถูกต้องของลำดับขั้นตอนในวิธีการประมวลผลด้วย

2.10.2. สัญลักษณ์ของผังงาน

การเขียนผังงานเบื้องต้นจะใช้สัญลักษณ์ดังต่อไปนี้คือ

ตารางที่ 2.3 สัญลักษณ์ของโพลัวชาร์ต

สัญลักษณ์	ความหมาย
	จุดเริ่มต้น และจุด สิ้นสุดของโปรแกรม (เป็นสิ่งแรกและสิ่งสุดท้ายที่จะต้องวาดในการเขียน flow chart)
	เส้นทางการไหลของโปรแกรม เพื่อช่วยในการเชื่อมแต่ละขั้นของโปรแกรม
	การประมวลผล การทำงาน การคิดคำนวณ
	รับข้อมูล/ส่งออกข้อมูล เช่นรับข้อมูลจากผู้ใช้ หรือแสดงผลค่าตัวแปร
	เงื่อนไข เป็นจุดที่มีเงื่อนไขให้เลือกทำ หรือเช็คค่าตามเงื่อนไข
	จุดเชื่อมต่อ คือจุดรวมจากหลายเส้นทางของโปรแกรมเข้ามาเส้นทางเดียวในกรณีที่ขั้นตอนต่อไปจะทำงานเหมือนกัน
	ขึ้นหน้าถัดไป ในกรณีที่เขียนหน้าเดียวไม่เพียงพอ

2.10.3. ขั้นตอนการเขียนโฟลว์ชาร์ต

การเขียนโฟลว์ชาร์ตจะใช้รายละเอียดจากวิธีการประมวลผลของการวิเคราะห์งาน ซึ่งประกอบด้วยขั้นตอนสำคัญคือการรับข้อมูล input การประมวลผล process การแสดงผล output นอกจากนี้ต้องใช้สัญลักษณ์ให้ตรงกับความหมายด้วย

2.10.4. การเขียนโฟลว์ชาร์ตแบบโครงสร้าง

การเขียนโฟลว์ชาร์ตแบบโครงสร้างมีประโยชน์คือทำให้การใส่ขั้นตอนการทำงานได้ง่ายและเป็นระเบียบซึ่งมีหลักการเขียนอยู่ 3 ข้อคือ

1.) รูปแบบเรียงลำดับ (Sequence Structure)

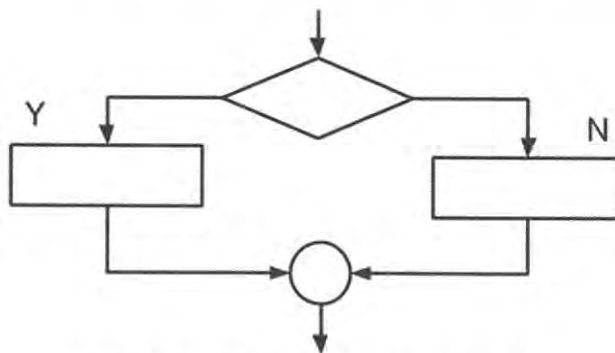
เป็นการทำงานแบบเรียงลำดับตั้งแต่ต้นจนจบ เป็นรูปแบบง่ายๆไม่มีการเปรียบเทียบใดใด มีทิศทางการไหลของข้อมูลเพียงทางเดียว ซึ่งอาจจะเป็นแบบบนลงล่างหรือจากซ้ายไปขวาก็ได้ จะเขียนเป็นผังงานได้ดังภาพที่ 2.71



ภาพที่ 2.71 ผังงานแบบเรียงลำดับ

2.) รูปแบบที่มีการกำหนดเงื่อนไขหรือให้เลือก (Decision Structure)

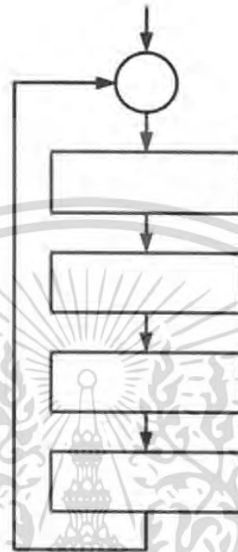
รูปแบบนี้จะยาวกว่ารูปแบบแรก เพราะจะมีการสร้างเงื่อนไขเพื่อให้เลือกทำงาน ถ้าหากเลือกทางใดก็จะไปทำงานในเงื่อนไขที่เลือก ซึ่งเงื่อนไขที่กำหนดขึ้นมีแต่เขียนอยู่ในสัญลักษณ์การตัดสินใจ



ภาพที่ 2.72 ผังงานที่มีการกำหนดเงื่อนไข

3.) รูปแบบที่มีการทำงานแบบวนรอบ หรือ Loop (Iteration Structure)

การทำงานของรูปแบบนี้จะเป็นการทำงานซ้ำๆ กันหลายๆ ครั้งเท่าที่ต้องการ ซึ่งการทำงานหรือจะขึ้นอยู่กับเงื่อนไขที่กำหนดให้ และจะมีการนำเอาลักษณะของการตัดสินใจมาช่วยว่าจะมีการทำงานซ้ำอีกหรือไม่



ภาพที่ 2.73 ผังงานที่มีการทำงานแบบวนรอบ

2.11. ยูเอ็มแอล (Unified Modeling Language : UML)

ยูเอ็มแอล (UML) คือโมเดลมาตรฐานที่ใช้หลักการออกแบบโปรแกรมเชิงวัตถุ (Object oriented analysis and design) รูปแบบของภาษา ยูเอ็มแอลจะมีลักษณะเป็นสัญลักษณ์ที่นำไปใช้ในโมเดลต่างๆ ยูเอ็มแอลจะมีข้อกำหนดกฎระเบียบต่างๆ โดยกฎระเบียบต่างๆ จะมีความหมายต่อการเขียนโปรแกรม ดังนั้น การใช้ยูเอ็มแอลจะต้องทราบความหมายของสัญลักษณ์ต่างๆ สิ่งเหล่านี้มีความจำเป็นอย่างยิ่งต่อการตีความของการออกแบบระบบก่อนนำไปพัฒนาระบบงานจริงซึ่งภาษาแผนภาพที่ใช้แสดงนั้นมีหลายแบบด้วยกัน ได้แก่ แผนภาพยูสเคส แผนภาพคลาส แผนภาพแสดงลำดับเหตุการณ์ ซึ่งมีหลักการในการออกแบบดังนี้

2.11.1. แผนภาพยูสเคส (Use case diagram)

แผนภาพยูสเคส คือ แผนภาพที่แสดงการทำงานของผู้ใช้ระบบและความสัมพันธ์กับระบบย่อยภายในระบบใหญ่ ในการเขียนแผนภาพยูสเคสผู้ใช้ระบบหรือผู้ที่เกี่ยวข้องกับระบบจะถูกกำหนดว่าเป็น แอคเตอร์ (Actor) โดยในแผนภาพจะอธิบายว่าแอคเตอร์นั้นสามารถใช้งานแอปพลิเคชันได้บ้าง หรือมีหน้าที่อะไรใน แอปพลิเคชันเพราะฉะนั้นแผนภาพยูสเคสจะบอกได้ถึงภาพรวมการใช้งานทั้งหมดภายในระบบหรือก็คือกิจกรรมทั้งหมดของแอคเตอร์ซึ่งจะทำให้มองภาพรวมของระบบได้ชัดเจนยิ่งขึ้น

2.11.1.1. สัญลักษณ์ที่ใช้ในแผนภาพยูสเคส

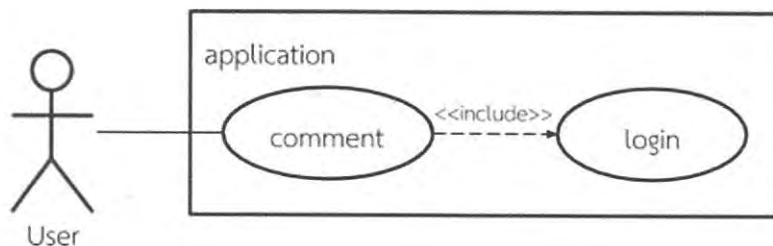
ตารางที่ 2.4 สัญลักษณ์ที่ใช้ในแผนภาพยูสเคส

สัญลักษณ์	ความหมาย
 Actor	ผู้ที่จะทำกับระบบอาจเป็นผู้ที่ทำการส่งข้อมูลรับข้อมูลหรือแลกเปลี่ยนข้อมูลกับระบบนั้นๆ
 Use case	หน้าที่หรืองานต่างๆในระบบ
 Connection	เส้นเชื่อมระหว่างยูสเคสกับแอกเตอร์
 System name System Boundary	ขอบเขตของระบบที่สนใจ เป็นเส้นขอบแบ่งระหว่างส่วนภายในระบบกับแอกเตอร์

2.11.1.2. ความสัมพันธ์ของแผนภาพยูสเคส

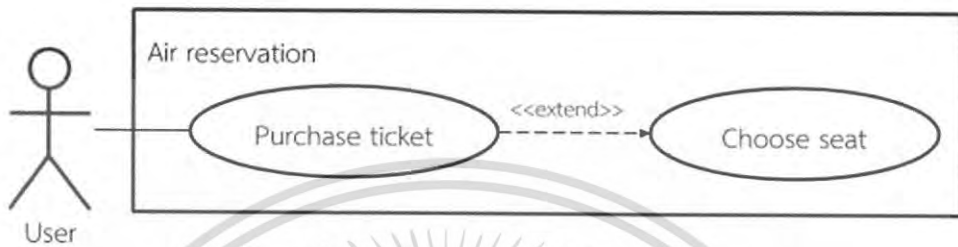
ความสัมพันธ์ระหว่างยูสเคส หมายถึง ความสัมพันธ์ที่แต่ละยูสเคสภายในระบบมีความสัมพันธ์กันโดยความสัมพันธ์ของยูสเคสนั้นสามารถแบ่งออกได้ 2 แบบ คือความสัมพันธ์แบบเรียกใช้ (include) และความสัมพันธ์แบบขยาย (Extend)

ความสัมพันธ์แบบเรียกใช้ หมายถึงการที่ยูสเคสหนึ่งเรียกใช้งานยูสเคสอีกอันหนึ่งคล้ายกับการเรียกใช้โปรแกรมย่อยโดยโปรแกรมหลักการเขียนสัญลักษณ์แทนความสัมพันธ์แบบเรียกใช้ของยูสเคสทำได้โดยใช้สัญลักษณ์เส้นประพร้อมหัวลูกศรชี้ไปยังยูสเคสที่ถูกเรียกใช้งาน และมีคำว่า <<include>> หรือ <<uses>> กำกับอยู่บนเส้นดังกล่าวที่ 2.74



ภาพที่ 2.74 ความสัมพันธ์แบบเรียกใช้ในแผนภาพยูสเคส

ความสัมพันธ์แบบขยาย หมายถึง การที่ยูสเคสหนึ่งไปมีผลต่อการทำงานตามปกติของอีกยูสเคสหนึ่ง นั่นหมายความว่ายูสเคสที่มีความสัมพันธ์แบบขยายนั้นจะมีผลทำให้การทำงานของยูสเคสที่ถูกความสัมพันธ์แบบขยายจะมีกิจกรรมที่เปลี่ยนแปลงไปสัญลักษณ์ที่ใช้แทนความสัมพันธ์แบบขยายในแผนภาพยูสเคส หรือใช้สัญลักษณ์ลูกศรโดยเริ่มจากยูสเคสที่มีความสัมพันธ์แบบขยายไปยังยูสเคสและมีคำว่า <<extend>> กำกับไว้ดังภาพที่ 2.75



ภาพที่ 2.75 ความสัมพันธ์แบบขยายในแผนภาพยูสเคส

2.11.2. แผนภาพลำดับเหตุการณ์ (ซีควเอนซ์ไดอะแกรม)

Sequence Diagram เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่างอ็อบเจกต์ของคลาส โดยเฉพาะการส่งข้อความ (message) ระหว่างอ็อบเจกต์ตามลำดับของเวลา (Sequence) ที่เกิดเหตุการณ์ขึ้นจากน้อยไปมาก โดยจะมีสัญลักษณ์แสดงให้เห็นลำดับของการข้อความ

2.11.2.1. สัญลักษณ์ที่ใช้ในแผนภาพลำดับเหตุการณ์

ตารางที่ 2.5 สัญลักษณ์ที่ใช้ในแผนภาพลำดับเหตุการณ์

สัญลักษณ์	ชื่อ	ความหมาย
	Object	อ็อบเจกต์ที่ต้องทำหน้าที่ตอบสนองต่อแอกเตอร์
	Lifeline	สัญลักษณ์แทนระยะเวลาทั้งหมดของอ็อบเจกต์
	Activation	สัญลักษณ์แทนระยะเวลาการทำงานของอ็อบเจกต์
	Message Synchronous	ข้อความหรือการติดต่อแบบรอคอย คำตอบหรือการตอบกลับก่อนที่ทำงานอื่นๆต่อไป

←-----	Message Return	ข้อความที่เกิดขึ้นในกรณีที่ต้นทางเริ่มจะติดต่อแล้วปลายทางต้องการติดต่อกลับด้วย
└─┘	Self-Delegation	การประมวลผลและคืนค่าที่ได้ภายในอีอบเจกต์เดียวกัน

2.12. ภาษาพีเอชพี (PHP)

เป็นภาษาสคริปต์ที่ทำงานฝั่งเครื่องแม่ข่าย ที่มีคุณสมบัติที่เหมาะสมต่อการนำมาพัฒนา ระบบ เพราะสามารถพัฒนาได้เร็ว และยืดหยุ่น รวมถึงมีไลบรารีและ เฟรมเวิร์คต่างๆรองรับมากมาย ในปัจจุบัน



ภาพที่ 2.76 สัญลักษณ์ของภาษาพีเอชพี

2.12.1. ภาษาพีเอชพี คืออะไร

PHP ย่อมาจากคำว่า Hypertext Preprocessor เป็นภาษาสคริปต์ (script) ที่ถูกพัฒนาแบบเปิด (Open-source) วัตถุประสงค์เพื่อใช้เป็นภาษาสคริปต์ในการพัฒนา ที่เหมาะสมสำหรับเว็บไซต์ และสามารถฝังตัว (embedded) ร่วมกับภาษา HTML ได้

```

<!DOCTYPE HTML>
<html>
  <head>
    <title>Example</title>
  </head>
  <body>

    <?php
      echo "Hi, I'm a PHP script!";
    ?>

  </body>
</html>

```

ภาพที่ 2.77 ตัวอย่างการฝังภาษาสคริปต์ PHP ลงใน HTML

การใช้งานภาษาสคริปต์ PHP จะช่วยทดแทนการเขียนโปรแกรมที่ซับซ้อน (เทียบกับภาษาซี (C) หรือ ภาษาเพอร์ (Perl) ในการสร้าง HTML) ในการเขียนเว็บไซต์ด้วย PHP สามารถเขียนภาษาด้วยภาษา HTML ปกติได้และสามารถฝังคำสั่งภาษา PHP บางอย่างลงไปได้ด้วย ดังในตัวอย่างตามภาพที่ 2.77 สิ่งที่ฝังลงไปคือคำสั่ง echo ในภาษา PHP เพื่อให้ทำการแสดงผลลัพธ์ออกมาเป็นคำว่า “Hi, I’m a PHP script!” โค้ดภาษา PHP ที่เขียนลงไปจะถูกล้อมรอบด้วย อักขระพิเศษคือ “<?php” ซึ่งเป็นตัวสั่งให้เริ่มการประมวลผล และ “?” ซึ่งเป็นตัวสั่งให้หยุดการประมวลผลบนภาษา PHP

สิ่งหนึ่งที่ทำให้ภาษาสคริปต์ PHP แตกต่างจากภาษาสคริปต์อื่นๆ เช่น client-side JavaScript นั่นคือ การทำงานของภาษาสคริปต์ PHP จะถูกสั่งให้ทำงาน (Executed) บนเครื่องแม่ข่าย (Server) จากนั้นจึงส่งผลลัพธ์ที่ได้เป็นภาษา HTML กลับมายังเครื่องลูกข่าย (Client) ที่เครื่องลูกข่ายจะได้ผลลัพธ์ที่เกิดจากการทำงานของสคริปต์เท่านั้น และจะไม่ทราบการทำงานของโปรแกรมที่เขียนด้วยภาษาสคริปต์ PHP ดังนั้นผู้ใช้งานจะไม่มีทางรู้ได้เลยถึงกระบวนการทำงานต่างๆ ก่อนที่จะได้ผลลัพธ์ซึ่งเขียนด้วยภาษาสคริปต์ PHP จะต่างจาก client-side JavaScript ที่การทำงานทั้งหมดจะถูกทำบนฝั่ง เครื่องลูกข่าย

ข้อดีที่ได้จากภาษาสคริปต์ PHP คือความง่ายในการโปรแกรม ทำให้เหมาะสมสำหรับผู้เริ่มต้น แต่ความง่ายของมันยังสามารถต้องโจทย์ความต้องการขั้นสูง รองรับการโปรแกรมที่ซับซ้อนสำหรับผู้เชี่ยวชาญ ด้วยคุณสมบัติและฟังก์ชันต่างๆ มากมายที่มีในภาษาสคริปต์ PHP

อย่างไรก็ตาม ภาษาสคริปต์ PHP จะพุ่งเป้าไปที่การพัฒนาการโปรแกรมให้ทำงานบนเครื่องแม่ข่าย (server-side) ซึ่งภาษาสคริปต์ PHP ยังมีความสามารถอีกหลากหลายด้านที่เหมาะสมกับการพัฒนาโปรแกรมฝั่งเครื่องแม่ข่าย

2.12.2. ความสามารถและจุดเด่นของภาษาพีเอชพี

ภาษา PHP จะเจาะจงไปที่การเขียนโปรแกรมในฝั่งเครื่องแม่ข่าย ภาษา PHP มีความสามารถเหมือนกับ ภาษาในการพัฒนาเว็บแอปพลิเคชัน (GCI : Common Gateway Interface) อื่นๆ เช่น การรับข้อมูลจากฟอร์มข้อมูลของภาษา HTML การสร้างหน้าเว็บแบบโต้ตอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(dynamic content) และ การทำงานร่วมกับ คุกกี้ (Cookies) เป็นต้น แต่ความสามารถของภาษา PHP ยังมีอีกมากมาย ซึ่งแบ่งเป็นการใช้งานใหญ่ๆได้ 3 แบบดังนี้

- 1.) การใช้งานฝั่งเครื่องแม่ข่าย (Server-side scripting)
- 2.) การใช้งานผ่าน Command line
- 3.) การใช้งานเขียนโปรแกรมบน คอมพิวเตอร์

ภาษา PHP สามารถใช้งานได้บนระบบปฏิบัติการที่หลากหลาย เช่น Linux, Unix (รวมถึงระบบปฏิบัติการที่พัฒนามาบนพื้นฐานของ ระบบปฏิบัติการ Unix), Windows, Mac OS X, RISC OS และอื่นๆ และสามารถทำงานบน เว็บเซิร์ฟเวอร์ (Web Server) ที่เป็นที่ยอมรับ เช่น Apache, IIS, lighttpd, nginx และอื่นๆ ซึ่งทำงานเป็น โมดูล หรือ GCI ก็ได้ ดังนั้นจึงมีอิสระเต็มที่ในการเลือกระบบปฏิบัติการ และ เว็บเซิร์ฟเวอร์ เพื่อพัฒนาแอปพลิเคชันด้วยภาษา PHP นอกเหนือจากนั้น ยังสามารถเลือกใช้การพัฒนาโปรแกรมแบบ การเขียนโปรแกรมเชิงกระบวนการ (Procedural Programming) หรือ การเขียนโปรแกรมเชิงวัตถุ (Object-oriented programming, OOP) หรือใช้ร่วมกันทั้งสองแบบก็ได้

ภาษา PHP ไม่ได้จำกัดความสามารถเพียงแค่การแสดงผลในรูปแบบของภาษา HTML แต่ยังสามารถสร้างผลลัพธ์เป็นสื่อในรูปแบบต่างๆได้ เช่น รูปภาพ, เอกสาร pdf หรือกระทั่ง ภาพเคลื่อนไหวแบบ flash แอนิเมชัน รวมถึงการเก็บไฟล์อักขระ เช่น XHTML, XML เพื่อจัดเก็บลงบนระบบ แทนที่การ พิมพ์เอกสารจริงๆ และยังสามารถนำข้อมูลภายในเอกสารไฟล์เหล่านั้นมาใช้ในอนาคตได้อีกด้วย

หนึ่งในความสามารถที่สำคัญ และเป็นจุดแข็งของภาษา PHP คือความสามารถในการรองรับการทำงานร่วมกับระบบฐานข้อมูลหลากหลายแบบ การเขียนหน้าเว็บให้ทำงานร่วมกับฐานข้อมูลสามารถทำได้ง่ายๆ เพียงแค่เพิ่มส่วนขยายเพื่อให้ทำงานกับระบบฐานข้อมูลแบบต่างๆ เช่น MySQL โดยใช้ PHP Data Object (PDO) เข้ามาช่วยจัดการข้อมูล หรือเชื่อมต่อกับฐานข้อมูลที่รองรับมาตรฐานการเชื่อมต่อฐานข้อมูลผ่าน Open Database Connectivity (ODBC)

นอกจากนั้น ภาษา PHP ยังสามารถเชื่อมต่อไปยังบริการอื่นผ่าน โพรโทคอลต่างๆ ได้เช่น LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM และอื่นๆ และยังสามารถสร้างการเชื่อมต่อแบบ socket เชื่อมต่อกับบริการและโปรโตคอลอื่นๆได้

2.12.3. ประวัติความเป็นมาของภาษาพีเอชพี

ภาษาพีเอชพี (PHP language) เกิดขึ้นในปี 1994 โดย Rasmus Lerdorf โปรแกรมเมอร์ชาวอเมริกันได้คิดค้นสร้างเครื่องมือ Common Gateway Interface (CGI) ที่ใช้ในการพัฒนาเว็บส่วนตัว (Personal Home Page) โดยใช้ข้อดีของภาษาซี (C) และ ภาษาเพอร์ล และยังได้พัฒนาให้เว็บส่วนตัวของสามารถติดต่อกับฐานข้อมูลได้ โดยใช้ชื่อว่า Form Interpreter (FI) รวมทั้งสองส่วนเรียกว่า "Personal Home Page/Forms Interpreter" หรือ PHP/FI ซึ่งถือเป็นจุดเริ่มต้นของ ภาษา PHP หลังจากการพัฒนา

2.12.4. เวอร์ชันของภาษาพีเอชพี

2.12.4.1. เวอร์ชัน 1

เวอร์ชัน 1 (PHP V.1) หลังจากการพัฒนา PHP/FI ได้สร้างตัวอย่างเว็บไซต์ที่ทำงานแบบไดนามิก (Dynamic Web Applications) เพื่อค้นหาข้อบกพร่อง (Bug) ในการทำงานของเว็บไซต์ และนำไปปรับปรุงแก้ไขการทำงานของ PHP/FI ที่สร้างขึ้นมา

Rasmus Lerdorf ได้ประกาศออก PHP เวอร์ชันที่ 1 ในชื่อ "Personal Home Page Tools (PHP Tools) version 1.0" บนกลุ่มการสนทนา Usenet เมื่อ 8 มิถุนายน 1995 ในเวอร์ชันนี้ ประกอบด้วยฟังก์ชันพื้นฐานต่างๆ การประกาศตัวแปรมีรูปแบบคล้ายกับในภาษา Perl มีความสามารถในการรับข้อมูลที่ส่งจากฟอร์มของภาษา HTML ความสามารถในการสร้างและฝังโค้ด HTML ภายในภาษา PHP โดยโครงสร้างประโยค (Syntax) ที่ใช้จะมีความคล้ายคลึงกับในภาษา Perl



```
$a = 1;
```

ภาพที่ 2.78 ตัวอย่างการประกาศตัวแปรในภาษา PHP

2.12.4.2. เวอร์ชัน 2

เวอร์ชัน 2 (PHP V.2) Rasmus Lerdorf ไม่ได้ตั้งใจที่จะให้ PHP เป็นภาษาคอมพิวเตอร์ใหม่ โดย Rasmus Lerdorf กล่าวว่า “I don’t know how to stop it, there was never any intent to write a programming language. I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way.” แต่ด้วยความนิยมที่เกิดจากข้อดีของภาษานี้ Rasmus Lerdorf จึงได้สร้างทีม เพื่อเริ่มพัฒนาทดสอบปรับปรุงในเวอร์ชันพัฒนา (beta) ให้มีความสามารถจัดการกับข้อมูลที่ถูกสร้างมาจากภาษา HTML ถัดมาอีก 2 เดือน จึงได้ออกเวอร์ชันที่ 2 (PHP/FI Version 2) ในเดือน พฤศจิกายน ปี 1997

2.12.4.3. เวอร์ชัน 3

เวอร์ชัน 3 (PHP V.3) Zeev Suraski และ Andi Gutmans ได้พัฒนาตัวแจงส่วน (Parser) ซึ่งเป็นพื้นฐานของ PHP เวอร์ชัน 3 และได้เปลี่ยนชื่อเป็น PHP : Hypertext Preprocessor จากนั้นได้ออกรุ่นทดลอง ออกสู่สาธารณะ และได้ออกรุ่นทางการ (Official) เมื่อเดือน มิถุนายน ปี 1998 จากนั้นทั้ง 2 คนได้พัฒนาส่วนกลางของ PHP ขึ้นใหม่ด้วย Zend Engine ในปี 1999 และเป็นผู้ก่อตั้ง บริษัท Zend Technologies ซึ่งตั้งอยู่ที่ประเทศ อิสราเอล

2.12.4.4. เวอร์ชัน 4

เวอร์ชัน 4 (PHP V.4) ซึ่งพัฒนาด้วย Zend Engine 1.0 ปล่อยออกสู่สาธารณะ เมื่อวันที่ 22 พฤษภาคม ปี 2000 z และได้ถูกพัฒนาอย่างต่อเนื่อง จนถึงเวอร์ชัน 4.4.9 และจึงได้หยุดพัฒนาและปรับปรุงความปลอดภัยลงในปี 2008

2.12.4.5. เวอร์ชัน 5

เวอร์ชัน 5 (PHP version 5) ซึ่งพัฒนาด้วย Zend Engine 2 ได้เผยแพร่ ออกสู่สาธารณะเมื่อ 13 กรกฎาคม ปี 2004 เพิ่มความสามารถใหม่ๆเข้ามา เช่น รองรับการเขียน โปรแกรมเชิงวัตถุ (Object-oriented programming : OOP), ส่วนเสริม PHP Data Object (POD) ซึ่งจะใช้ในการกำหนดคีย์ค่า เพื่อเข้าถึงฐานข้อมูล และ ปรับปรุงประสิทธิภาพหลายๆด้าน

ในปี 2008 หลังจากที่ PHP Version 4 ได้มีการหยุดการพัฒนาลง จึงเหลือแค่ PHP Version 5 เท่านั้นที่ยังอยู่ภายใต้การพัฒนา หลายโปรเจกต์ที่เป็น การพัฒนาแบบเปิด (Open-Source) หยุดรองรับ PHP Version 4 และหันไปรองรับ PHP Version 5 เพราะได้มีการออก เครื่องมือแปลงการทำงานของ PHP Version 4 เป็น PHP Version 5 ชื่อว่า GoPHP5 ซึ่งสร้างโดย สมาคมนักพัฒนา PHP

ต่อมา ตัวแปลภาษา PHP (PHP Interpreters) ซึ่งจะทำให้สามารถใช้งาน ภาษา PHP บนระบบปฏิบัติการต่างๆ ได้พัฒนาให้รองรับระบบปฏิบัติการส่วนใหญ่ ทั้งแบบ 32 บิต (x86) และ 64 บิต (x64)

2.12.4.6. เวอร์ชัน 6

PHP ในตอนแรกยังไม่รองรับ ภาษาที่เป็น Native Unicode ในปี 2005 Andrei Zmievski หัวหน้าโปรเจกต์ในการพัฒนาให้ PHP รองรับภาษา Native Unicode โดยการนำไลบรารี International Components for Unicode (ICU) ฝังลงใน PHP ซึ่งจะทำให้ภาษา PHP รองรับภาษาที่เป็น Native Unicode ตามมาตรฐาน UTF-16 จะทำให้รองรับในเรื่องของภาษาต่างๆ การเปลี่ยนแปลงนี้จะถูกเพิ่มเข้ามาใน PHP version 6 ร่วมกับคุณสมบัติใหม่ๆอีกหลายประการที่พัฒนาขึ้น

อย่างไรก็ตาม การพัฒนาที่เร่งรีบรวมถึงขาดนักพัฒนาที่เข้าใจ ปัญหาเรื่อง ประสิทธิภาพจึงเกิดขึ้นเนื่องจากการรองรับ Unicode UTF-16 ซึ่งแทบจะยังไม่มีการใช้งานบนบริบท ของเว็บไซต์ โปรเจกต์นี้จึงถูกเลื่อนออกไปก่อน จึงทำให้ต้องออก PHP version 5.3 แทน PHP version 6 ในปี 2009 PHP Version 5.3 ซึ่งเป็นการแปลงกลับจาก PHP Version 6 พร้อมกับ คุณสมบัติใหม่ๆ ยกเว้นคุณสมบัติเรื่องการรองรับ Unicode

PHP Version 6 ที่จะสามารถรองรับ Unicode ก็ถูกปล่อยปะละเลย จนกระทั่งเดือน มีนาคม ปี 2010 ได้มีการเผยแพร่ PHP Version 5.4 อีกครั้ง ซึ่งมีการพัฒนาเพิ่ม คุณสมบัติที่จะมีใน PHP Version 6 เข้ามาไว้ใน PHP Version 5.4 นี้ยกเว้นแต่ คุณสมบัติในการ รองรับ Unicode เช่นเดิม แนวคิดที่จะเพิ่มการรองรับ Unicode ใน PHP Version 6 ก็ยังไม่มีการพูด ถึงอย่างเป็นทางการจนกระทั่งปี 2014

2.12.4.7. เวอร์ชัน 7

ระหว่างปี 2014 ถึง 2015 การเปลี่ยนแปลงใหญ่ของ PHP อยู่ภายใต้การพัฒนา แต่ก็เป็นที่ถกเถียงกันในเรื่องเลขเวอร์ชันที่จะใช้บน PHP Version 7 เพราะ PHP Version 6 ยังอยู่ในขั้นการพัฒนาและทดสอบ ยังไม่มีการเผยแพร่ออกสู่สาธารณะ แต่ข้อถกเถียงที่จะนำชื่อ PHP Version 6 มาใช้อีกครั้งก็ตกไปหลังจากการโหวต เพราะอาจจะทำให้เกิดความสับสนในการเรียกเรียงข้อมูล และการอ้างอิงบนหนังสือ

พื้นฐานของของ PHP Version 7 ถูกขนานนามให้เป็น PHP next generation (PHPNG) และถูกแยกเป็นแขนงหนึ่งของ PHP ซึ่งถูกสร้างโดย Dmitry Stogov, Xinchen Hui และ Nikita Popov ซึ่งการพัฒนา PHP ในเวอร์ชันที่ 7 มุ่งเป้าไปที่การเพิ่มประสิทธิภาพ ด้วยการแก้ไข ปรับปรุง และพัฒนา (Refactoring) Zend Engine แต่ยังคงความเข้ากันได้กับภาษา PHP ปกติมากที่สุด

ได้มีการทดสอบทำการวัดผล (Benchmark) ด้วยการใช้ชุดทดสอบ WordPress-Base Benchmarks เมื่อวันที่ 14 มิถุนายน 2014 ซึ่งผลที่ได้แสดงให้เห็นว่าการทำงานของ PHP Version 7 มีประสิทธิภาพที่ดีขึ้นกว่าเดิมเกือบ 100% ผู้พัฒนาคาดหวังจะทำให้ PHP Version 7 สามารถใช้งานได้ง่าย มีประสิทธิภาพที่ดีขึ้นไปอีกในอนาคต เช่นเดียวกับ การพัฒนาโครงสร้างข้อมูล (Data Structure) ให้มีความกระชับรัดกุมกว่าเดิม การเปลี่ยนแปลงอื่นๆ เช่นการทำงานร่วมกับคอมไพเลอร์แบบ just-in-time (JIT) และเนื่องจากการเปลี่ยนที่เกิดขึ้นกับ Zend Engine อย่างมีนัยสำคัญ จึงเปลี่ยนชื่อเป็น Zend Engine 3 เพื่อรับช่วงต่อจาก Zend Engine 2 ที่ใช้ใน PHP Version 5 การเปลี่ยนแปลงอย่างมากใน PHPNG จึงได้รับชื่อใหม่เป็น PHP Version 7 แทนที่จะเป็น PHP Version 5.x.x ตามปกติ

โดยปกติภาษา PHP แต่ละเวอร์ชันจะรองรับการทำงานและรูปการเขียน (Coding) และความเข้ากันได้ (Compatibility) ของ PHP เวอร์ชันที่เก่ากว่า แต่ PHP version 7 พัฒนาไปเกินกว่าที่จะเข้ากันได้กับ PHP เวอร์ชันเก่าในบางส่วน โดยเฉพาะอย่างยิ่ง การเปลี่ยนแปลงดังนี้

- การแจ้งเตือนการทำงานผิดพลาดที่ร้ายแรง (Fatal) หลากหลายอย่าง จะถูกแทนที่ด้วยข้อมูลข้อยกเว้นเชิงวัตถุ (Object-oriented Exceptions)
- ยกเลิกการรองรับ (Deprecated) รูปแบบการประกาศ Constructor เมธอดใน PHP Version 4
- รูปแบบการทำงานของ foreach
- บางคลาส (Class) ที่มากับ PHP จากเดิมเมื่อมีความผิดพลาดเกิดขึ้นในการทำงาน (Execute) จะส่งค่ากลับออกมาเป็นค่าว่างเปล่า (null) แต่จะเปลี่ยนเป็นการโยนข้อมูลความผิดพลาดกลับมาแทน (throw Error Exception)

จากเปลี่ยนแปลงข้างต้นเป็นเพียงส่วนหนึ่งของการเปลี่ยนแปลงสำคัญที่เกิดขึ้นกับ PHP Version 7 ที่จะต้องคำนึงถึงความเข้ากันได้หากมีการใช้งานรูปแบบการเขียนของเวอร์ชันเก่าบน PHP Version 7

2.13. ลาลาเวลเฟรมเวิร์ค (Laravel)

2.13.1. ลาลาเวลเฟรมเวิร์คคืออะไร

Laravel เป็นโปรเจกต์ เว็บเฟรมเวิร์ค ที่มีการพัฒนาแบบเปิด (Open-source) โดยผู้เริ่มพัฒนาที่ชื่อ Taylor Otwell ซึ่งมีจุดประสงค์ในการทำให้โครงสร้างการเขียน เว็บแอปพลิเคชันเป็นไปตามรูปแบบ สถาปัตยกรรมการออกแบบระบบแบบ โมเดล วิว คอนโทรลเลอร์ (Model View Controller : MVC)

Laravel จะมีลักษณะการทำงานที่เป็นแบบแยก คือ จะมีการแบ่งการทำงาน ออกเป็นโมดูล (Module) ที่ทำหน้าที่แตกต่างกันไป มีการบริหารจัดการฐานข้อมูลที่แยกการเข้าถึงข้อมูลแต่ละข้อมูลออกจากกัน รวมถึงเครื่องมืออื่นๆที่ช่วยในการพัฒนา และปรับปรุงแอปพลิเคชัน ที่จะทำให้การพัฒนาแอปพลิเคชัน เป็นไปอย่างเรียบง่าย และ มีระบบ (syntactic-sugar)

ในเดือน มีนาคม ปี 2015 Laravel เฟรมเวิร์ค เป็นหนึ่งใน PHP เฟรมเวิร์คที่ได้รับ ความนิยมอย่างมาก เช่นเดียวกับ เฟรมเวิร์คอื่นๆ (Symfony2, Nette, Codelgniter, Yii2)



ภาพที่ 2.79 Taylor Otwell ผู้พัฒนา Laravel Framework

2.13.2. ประวัติและความเป็นมา

2.13.2.1. เวอร์ชัน 1

Laravel เฟรมเวิร์ค ได้ออกเวอร์ชัน (beta) ครั้งแรก ในวันที่ 9 มิถุนายน ปี 2011 และได้ออกเวอร์ชันจริง ในเดือนเดียวกัน ในเวอร์ชันแรก (Laravel 1) ประกอบไปด้วยความสามารถ และกลไกต่างๆเช่น การยืนยันตัวตน (Authentication), การรองรับภาษา (Localization), โมเดล (Model), วิว (View), เซสชัน (Session), การนำทาง (Routing) และกลไกอื่นๆ แต่ยังคงขาดกลไกการทำงานแบบ คอนโทรลเลอร์ (Controller) ซึ่งนั่นทำให้ใน Laravel 1 ยังไม่ใช่เฟรมเวิร์คที่มีสถาปัตยกรรมระบบแบบ MVC จริงๆ

เอกสารนี้เขียนและเผยแพร่โดยอาจารย์ ดร. อธิวัฒน์ อธิวัฒน์ อาจารย์สอนวิชาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.2.2. เวอร์ชัน 2

Laravel 2 ถูกเผยแพร่อย่างเป็นทางการในเดือน กันยายน 2011 นำข้อเสนอแนะต่างๆในชุมชนนักพัฒนาที่มีต่อ Laravel 1 มาปรับปรุงเพื่อเพิ่มประสิทธิภาพและความสามารถ และได้เพิ่มกลไก คอนโทรลเลอร์ (Controller) เข้ามาแล้ว ทำให้กลายเป็นเฟรมเวิร์คที่มีสถาปัตยกรรมระบบแบบ MVC อย่างแท้จริง

2.13.2.3. เวอร์ชัน 3

Laravel 3 ถูกเผยแพร่อย่างเป็นทางการเมื่อเดือน กุมภาพันธ์ ปี 2012 มีการเพิ่มคุณสมบัติหลายอย่างเช่น การรองรับคำสั่งผ่านหน้าต่างคำสั่งควบคุม (Command-line Interface : CLI) โดยใช้คำสั่งที่ชื่อว่า “อาร์ติซาน” (Artisan) รองรับการจัดการฐานข้อมูล (Database Management Systems : DBMS) ภายในตัวเฟรมเวิร์คเอง รองรับการปรับปรุงโครงสร้างฐานข้อมูล (Migrations) แบบเวอร์ชันคอนโทรล (version control) ความสามารถในการจับความเปลี่ยนแปลง (handling-event) และความสามารถในการเพิ่ม ส่วนขยายให้กับระบบ (Bundles) Laravel 3 ได้รับความนิยมเพิ่มขึ้นเป็นอย่างมากในเวอร์ชันนี้

2.13.2.4. เวอร์ชัน 4

Laravel 4 มีรหัสว่า Illuminate ถูกเผยแพร่ออกสู่สาธารณะเมื่อเดือน พฤษภาคม ปี 2013 ได้มีการปรับปรุงเฟรมเวิร์คทั้งหมดใหม่ แกะไขการปรับปรุงโครงสร้างเฟรมเวิร์คให้แยกส่วนต่างๆของเฟรมเวิร์ค ออกเป็นแพ็คเกจย่อยให้สามารถจัดการเฟรมเวิร์คได้จากคำสั่ง “คอมโพสเซอร์” (Composer) ซึ่งเป็นคำสั่งที่ใช้จัดการโครงสร้างและแพ็คเกจภายในแอปพลิเคชัน ที่ทำงานอยู่บนเครื่องแม่ข่าย (server) เพิ่มการทำงานที่ชื่อว่า “ดาต้าเบส ซีตติ้ง” (database seeding) เพื่อใช้ในการกำหนดข้อมูลเริ่มต้นให้กับฐานข้อมูล และได้มีการวางแผนการปรับปรุงประสิทธิภาพให้ดีขึ้นและออกเวอร์ชันย่อย ในทุกๆ 6 เดือน รองรับการสื่อสารแบบไม่สอดคล้องกัน (Message Queues) รองรับการลบข้อมูลแบบชั่วคราวเรียกว่า “ซอฟต์ดีลิต” (Soft Deleting)

2.13.2.5. เวอร์ชัน 5

Laravel 5 ถูกเผยแพร่สู่สาธารณะเมื่อเดือน กุมภาพันธ์ ปี 2015 หลังจากการปรับปรุงรุ่นที่ 4 จบลงที่เวอร์ชัน 4.3 (Laravel 4.3) เพิ่มความสามารถในการกำหนดตารางการทำงานล่วงหน้า (Corn) ภายในตัวเฟรมเวิร์คเองผ่านแพ็คเกจที่เรียกว่า “Scheduler” รองรับการจัดการทรัพยากรภายในแพ็คเกจผ่าน “Elixir” และยังเพิ่มความสามารถในด้านความน่าเชื่อถือ (authentication) ผ่านแพ็คเกจที่ชื่อว่า “Socialite”

2.13.2.6. เวอร์ชัน 6

เผยแพร่เมื่อ มิถุนายน ปี 2015 ในเวอร์ชันนี้เป็นเวอร์ชันแรกที่ได้รับการดูแลระยะยาว (long-term support : LTS) ซึ่งจะประกอบด้วย การปรับปรุงข้อบกพร่อง เป็นระยะเวลา 2 ปี นับจากวันเผยแพร่ การปรับปรุงด้านความปลอดภัย เป็นระยะเวลา 3 ปี นับจากเผยแพร่ และจะเผยแพร่เวอร์ชันใหม่ทุกๆ 2 ปี

2.13.2.7. เวอร์ชัน 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เผยแพร่มื่อ สิงหาคม ปี 2016 คุณสมบัติใหม่ใน Laravel เวอร์ชัน 5.3 จะมุ่งเป้าไปที่ความเร็วในการพัฒนา ด้วยการเพิ่มฟังก์ชันและคำสั่งที่มีความต้องการ ในการใช้งานเข้ามาในเฟรมเวิร์คเพื่อเพิ่มความสะดวกในการพัฒนา

2.13.3. ความสามารถและจุดเด่นของ ลาลาเวลเฟรมเวิร์ค

Bundle คุณสมบัติในการเพิ่มโมดูลแพ็คเกจเข้ามาในระบบ เพื่อใช้งานร่วมกับเฟรมเวิร์ค ซึ่งถูกเพิ่มเข้ามาตั้งแต่ Laravel 3 และใน Laravel 4 จะใช้คำสั่ง Composer เข้ามาช่วยในการจัดการ โมดูลและแพ็คเกจต่างๆ จาก Packagist repository

Eloquent ORM (object-relational mapping) เป็นฟังก์ชันขั้นสูง ที่จะผลสานการทำงานระหว่างคลาสกับฐานข้อมูล โดยจะสร้างก้อนข้อมูลออกมา จากข้อมูลภายในฐานข้อมูล ซึ่งสามารถกำหนดข้อจำกัดต่างๆได้จาก เมธอดภายในคลาส โดยก้อนข้อมูลที่ได้จากคลาสหนึ่งก้อนจะผูกกับแถวหนึ่งแถวของตารางในฐานข้อมูล

Query Builder ถูกเพิ่มครั้งแรกใน Laravel 4 ทำหน้าที่ในการให้บริการเข้าถึงข้อมูลภายในฐานข้อมูลโดยตรงแทนที่การใช้งาน Eloquent ORM และทดแทนการใช้คำสั่ง Query โดยตรงและยังเพิ่มความสามารถเข้าไป เช่น การ เก็บผลลัพธ์ในการเรียกดู (caching) การทำ Soft Delete เป็นต้น

Application logic เป็นส่วนที่จะอำนวยความสะดวกในการพัฒนาแอปพลิเคชัน และกำหนดการใช้งานคอนโทรลเลอร์ผ่านการกำหนดค่าภายในเราต์ (Routes)

Restful Controller เพิ่มคุณสมบัติในการรองรับ และกำหนดการรองรับข้อแบบ restful ทั้ง HTTP GET, HTTP POST, HTTP PUT, HTTP DELETE เป็นต้น

Migrations ให้บริการการควบคุมระบบเวอร์ชันของการแปลงโครงสร้างฐานข้อมูล โดยการเขียนการเปลี่ยนแปลงโครงสร้างตามที่ต้องการลงในไฟล์ database schemas แทนที่การเขียน Query แบบปกติ ทำให้สามารถควบคุมการเปลี่ยนแปลงโครงสร้างของฐานข้อมูลได้ง่ายขึ้น

Database Seeding ทำหน้าที่ในการสร้างตารางและข้อมูลต้นแบบภายในฐานข้อมูลเพื่อใช้ในการทดสอบระหว่างการพัฒนาแอปพลิเคชัน

2.14. ส่วนต่อประสานโปรแกรมประยุกต์แบบ เรสฟูล (RESTful API)

Representational state transfer หรือ RESTful เป็นบริการติดต่อสื่อสารบนเว็บ (Web Service) แบบหนึ่ง ที่ให้บริการการแลกเปลี่ยนข้อมูลระหว่างเครื่องคอมพิวเตอร์ ผ่านระบบเครือข่ายอินเทอร์เน็ต ซึ่ง Web Service จะแตกต่างจาก Web Application ในเรื่องลักษณะของการใช้งาน Web Application สามารถเข้าใช้งานบริการต่างๆผ่านทางหน้าจอส่วนติดต่อผู้ใช้งาน web browser ได้ทันที ส่วนการใช้งาน Web Service จะไม่มีส่วนติดต่อผู้ใช้โดยตรงต้องมีการติดต่อสื่อสารผ่านโปรโตคอลอื่นๆแทนโดยมากแล้วจะเป็นการติดต่อสื่อสารกันระหว่างเครื่องคอมพิวเตอร์ ซึ่งแต่ละ Web Service ก็จะมีรูปแบบการติดต่อสื่อสารที่แตกต่างกันขึ้นอยู่กับผู้พัฒนา RESTful เป็นรูปแบบหนึ่งที่อนุญาตให้มีการร้องขอ (Request) เข้ามายังระบบ Web Service เพื่อจัดการกับข้อมูล หรือทรัพยากรต่างๆบนคอมพิวเตอร์ โดยใช้ชุดรูปแบบคำสั่งการร้องขอ ที่เป็นอิสระต่อกัน (stateless) stateless ใน API หมายถึงในแต่ละการร้องขอที่ถูกส่งมายัง API จะไม่มีผลกระทบต่อกันหรืออธิบายง่าย ๆ ก็คือจะไม่มีการเก็บสถานะการประมวลผลจากแต่ละ request ไว้ใน cache ของระบบเพื่อนำไป

ประมวลผลกับ request อื่นๆนั่นเอง นอกจากบริการบนเว็บแบบ RESTful แล้วยังมีรูปแบบอื่นๆของการให้บริการแลกเปลี่ยนข้อมูลเช่น WSDL และ SOAP

แหล่งข้อมูลบนเว็บ (Web resource) ในยุคแรกที่มีการใช้ World Wide Web : WWW แหล่งข้อมูลบนเว็บ หมายถึง เอกสาร หรือ ไฟล์ ต่างๆซึ่งจะสามารถเข้าถึงเอกสารและไฟล์เหล่านั้นได้จากการระบุ URLs แต่ปัจจุบันสามารถเข้าถึงทุกสิ่งทุกอย่างได้ ไม่ใช่แค่เพียง ไฟล์ หรือเอกสาร แต่ยังคงรวมไปถึง ชื่อ ที่อยู่ ทุกสิ่งทุกอย่างที่อยู่บน WWW ด้วยรูปแบบที่ง่ายและยังสามารถจัดการกับสิ่งเหล่านั้นได้ด้วย

ใน RESTful การร้องขอจะสร้างสิ่งที่เรียกว่า URI เพื่อร้องขอการตอบกลับ (Response) ซึ่งการตอบกลับดังกล่าวอาจจะอยู่ในรูปแบบเช่น XML, HTML, JSON หรือบางสิ่งบางอย่างตามที่กำหนด การตอบกลับที่ได้จะเป็นการยืนยันว่าได้ทำสิ่งที่ร้องขอเรียบร้อยแล้วหรือไม่ การร้องขอจะผ่านโปรโตคอล HTTP ซึ่งลักษณะของการร้องขอจะถูกกำหนดผ่าน คำกริยาของโปรโตคอล HTTP คือ GET, POST, PUT, DELETE และใช้รูปแบบการทำงานที่เป็นอิสระต่อกันในแต่ละการร้องขอเพื่อให้ได้ประสิทธิภาพที่รวดเร็วในการทำงาน มีความน่าเชื่อถือ และสามารถรองรับการเติบโตของระบบได้ สามารถปรับปรุงหรืออัปเดตได้โดยไม่ส่งผลกระทบต่อระบบ หรือกระทั่งสามารถปรับปรุงได้ทันทีในขณะที่ระบบกำลังทำงาน

แนวคิดของ REST ถูกคิดค้นและกำหนดขึ้นในปี 2000 โดย Roy Fielding ในวิทยานิพนธ์ปริญญาเอกของ Roy Fielding โดยใช้ REST ออกแบบ HTTP 1.1 และ URI ขึ้นมา เพื่อตั้งใจที่จะทำให้เกิดภาพการออกแบบ เว็บเซอร์วิสที่ดี โดยผู้ใช้สามารถเข้าถึงทรัพยากรบนเว็บได้ด้วยการคลิกลิงก์ เช่น /user/tom และกำหนดตัวดำเนินการเป็นคำกริยาต่างๆตามที่ต้องการเช่น GET หรือ DELETE และจะมีผลลัพธ์ตอบกลับมายังผู้ใช้เพื่อให้ผู้ใช้แต่ละคนที่ร้องขอทราบและหรือนำผลลัพธ์นั้นไปดำเนินการในขั้นต่อไป

2.14.1. ประวัติและความเป็นมา

REST ถูกกำหนดขึ้นในปี 2000 โดย Roy Thomas Fielding ในวิทยานิพนธ์ปริญญาเอกชื่อ "Architectural Styles and the Design of Network-based Software Architectures" ณ University of California, Irvine ซึ่งได้ออกแบบสถาปัตยกรรม REST ควบคู่ไปกับการพัฒนา HTTP 1.1 ในช่วงปี 1996-1999 โดยพัฒนาจากรูปแบบของ HTTP 1.0 ที่มีอยู่แล้วตั้งแต่ปี 1996

2.14.2. คุณสมบัติและข้อกำหนด

- Client-server

การทำงานร่วมกันระหว่างภาระงานของส่วนติดต่อผู้ใช้ กับภาระงานของส่วนจัดการข้อมูล จะต้องแยกกันและมีการทำงานในรูปแบบ client-server เพื่อให้การทำงานข้ามแพลตฟอร์มทำได้ง่ายขึ้น และมีความสามารถในการขยายระบบได้ง่ายขึ้น ซึ่งเป็นสิ่งสำคัญในการใช้งานเว็บ

- Stateless

การทำงานร่วมกันระหว่าง client-server จะต้องไม่มีการเก็บบริบทของแต่ละการร้องขอไว้บนเซิร์ฟเวอร์เพื่อใช้ร่วมกับบริบทการร้องขออื่นๆ การร้องขอจาก client จะต้องประกอบด้วยสิ่งที่จำเป็นในการใช้งานเว็บเซอร์วิสแล้วภายในแต่ละการร้องขอ

เอกสารนี้เป็นเอกสารที่จำเป็นในการใช้งานเว็บเซอร์วิสแล้วภายในแต่ละการร้องขอ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Cacheable

ต้องไม่มีการเก็บผลลัพธ์ในการตอบกลับไว้ เพื่อป้องกันการความผิดพลาดในการใช้ข้อมูลที่ค้างอยู่ หรือข้อมูลที่ไม่เหมาะสม กับการตอบกลับในอนาคต

- Layered system client

จะไม่สามารถเชื่อมต่อกับ Server ตรงๆได้ จะต้องมีส่วนกลางในการติดต่อสื่อสาร เพื่อให้มีประสิทธิภาพในการขยายระบบที่ดี รวมถึงช่วยในเรื่องการกระจายการทำงาน (load-balancing) และการควบคุมความปลอดภัยด้วย

- Code on demand

server จะสามารถกำหนดการทำงานบน client ได้ด้วยการส่งชุดคำสั่งที่สามารถทำงานบนเครื่อง client เช่น java applets และชุดคำสั่งจําพวก client-side script เช่น JavaScript เป็นต้น

- Uniform interface

เป็นข้อบังคับพื้นฐานของระบบที่ให้บริการแบบ REST การใช้ uniform interface จะช่วยลดความยุ่งยากและแยกข้อมูลหรือทรัพยากรแต่ละอันออกจากกัน เพื่อลดความยุ่งยากซับซ้อนในการเข้าถึงหรือ กระทำการใดๆกับข้อมูลหรือทรัพยากรที่ต้องการ โดยอาศัยข้อกำหนดต่างๆ ต่อไปนี้

- Identification of resource: ในแต่ละการร้องขอจะมีการระบุ ถึงข้อมูลหรือทรัพยากร เช่นการใช้ URIs ในการอ้างอิงไปยังข้อมูลบนระบบที่ให้บริการแบบ REST (/user/tom) ข้อมูลหรือทรัพยากรต่างก็มีวิถีทางในการแสดงผลลัพธ์ที่แตกต่างกัน ยกตัวอย่างเช่น Server ส่งข้อมูลหน้าเว็บกลับไปยัง client ในรูปแบบ HTML หรือส่งข้อมูลศิลปินกลับมายัง client ในรูปแบบ XML หรือ JSON เป็นต้น
- Manipulation of resources through representations: ในการร้องขอจะต้องมีการกำหนดกริยาที่จะกระทำต่อข้อมูลหรือทรัพยากรที่ต้องการ เช่นใน มาตรฐาน HTTP ใช้กริยา GET เพื่อแสดงว่าต้องการที่จะดึงข้อมูลหรือทรัพยากรของ URI ที่ระบุไว้
- Self-descriptive messages: ในการร้องขอหรือตอบกลับ จะต้องมิลักษณะที่สามารถอธิบายได้ว่าข้อมูลดังกล่าวคือข้อมูลอะไร โดยที่ผู้รับไม่จำเป็นต้องทราบถึงโครงสร้างข้อมูลของผู้ส่ง
- Hypermedia as the engine of application state (HATEOAS): ในการตอบกลับจาก server จะต้องประกอบด้วย hyperlink สำหรับการทำงานอื่นๆที่สามารถทำได้ต่อจากปัจจุบันเพื่อให้ระบบไม่จำเป็นต้องกำหนดค่าตายตัวในการทำงานขั้นต่อไปด้วยตัวเอง ยกตัวอย่างเช่น webpage เมื่อมีการร้องขอหน้า webpage ผลลัพธ์ที่ตอบกลับจาก server จะประกอบด้วย hyperlink มากมายเพื่อทำงานขั้นต่อไป

2.15. ระบบปฏิบัติการลินุกซ์ (Linux)

ลินุกซ์ เป็นระบบปฏิบัติการ (Operating system) เช่นเดียวกับ ไมโครซอฟต์วินโดวส์ ดอส หรือ ยูนิกซ์ลินุกซ์เป็นระบบปฏิบัติการที่มีการพัฒนาแบบเปิด นั่นหมายถึงว่าผู้ที่ใช้งานระบบปฏิบัติการ ลินุกซ์ นั้นเมื่อเกิดปัญหาในการใช้งานสามารถที่จะพัฒนาและแก้ไขในส่วนต่างๆ ได้ทันที

ในปัจจุบันระบบปฏิบัติการลินุกซ์ ได้รับความนิยมเป็นอย่างมาก เนื่องจากเป็นระบบที่ทำงานได้อย่างอิสระ ปลอดภัย เชื่อถือได้ และราคาถูก จึงถูกองค์กรชั้นนำต่างๆ นำ ลินุกซ์ ไปพัฒนาเป็นระบบปฏิบัติการและปรับใช้กับเครื่องคอมพิวเตอร์เซิร์ฟเวอร์และคอมพิวเตอร์ส่วนบุคคล

เริ่มแรกลินุกซ์ พัฒนาให้สามารถใช้งานบนเครื่องคอมพิวเตอร์ อินเทล 386 ไมโครโพรเซสเซอร์ หลังจากนั้นมีการนำลินุกซ์ ไปพัฒนาให้รองรับสถาปัตยกรรมคอมพิวเตอร์ในระบบต่างๆ แม้กระทั่งในโทรศัพท์มือถือ และกล้องบันทึกวีดีโอ

2.15.1. ประวัติ

ระบบปฏิบัติการลินุกซ์ถือกำเนิดขึ้นในประเทศฟินแลนด์ ในปี ค.ศ. 1980 โดย Linus Trovalds นักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยเฮลซิงกิ Linus เล็งเห็นว่าระบบปฏิบัติการ มินิกซ์ ซึ่งพัฒนาโดย Andrew S. Tanenbaum ความสามารถยังมีข้อจำกัดอยู่ จึงได้เริ่มพัฒนาระบบปฏิบัติการของตนเองขึ้นมา เพื่อใช้งานและศึกษาทำความเข้าใจในวิชา ระบบปฏิบัติการคอมพิวเตอร์ที่ Linus เรียนด้วย เมื่อการพัฒนาดำเนินไปสักระยะหนึ่ง Linus จึงได้ชักชวนให้นักพัฒนาคณะอื่นๆ เข้ามาช่วยในการพัฒนาระบบปฏิบัติการของ Linus ที่มีชื่อว่า ลินุกซ์

Linus เป็นผู้ที่พัฒนาและคอยรวบรวมโปรแกรมที่นักพัฒนาคณะอื่นๆ ได้ร่วมกันพัฒนาขึ้นมาผ่านทางเครือข่ายอินเทอร์เน็ต และได้แจกจ่ายให้ทดลองใช้เพื่อทดสอบและหาข้อบกพร่อง ปัจจุบันรุ่นล่าสุดของ ลินุกซ์ คือรุ่น 4.9.6 เผยแพร่เมื่อ 26 มกราคม ปี ค.ศ. 2017

2.15.2. การนำระบบปฏิบัติการ ลินุกซ์ มาใช้งาน

เดิมทีการใช้งาน ลินุกซ์ คือใช้เป็นระบบปฏิบัติการสำหรับเครื่องคอมพิวเตอร์เซิร์ฟเวอร์ ด้วยคุณสมบัติของ ลินุกซ์ ที่มีความยืดหยุ่นและสามารถปรับแก้และพัฒนาต่อให้เหมาะสมตามที่ต้องการได้ ต่อมาจึงมีการนำไปประยุกต์ใช้กับงานหลายๆประเภท เนื่องจากต้นทุนในการใช้งาน ลินุกซ์ มีต้นทุนที่ต่ำและความสามารถในการปรับแต่งได้อย่างอิสระ จึงถูกนำมาใช้ในระบบฝังตัว เช่น เครื่องรับสัญญาณโทรทัศน์ อุปกรณ์พกพา กระทั่ง โทรศัพท์มือถือ ลินุกซ์ เป็นคู่แข่งสำคัญของระบบปฏิบัติการ ซิมเบียนโอเอส (Symbian OS) ซึ่งถูกใช้ในโทรศัพท์มือถือเป็นจำนวนมาก ในช่วงปี ค.ศ. 2000 ถึง ค.ศ. 2010 และนำมาใช้ทดแทน วินโดวส์ซีอี และปาล์มโอเอส อีกด้วย ไฟร์วอลล์และเราเตอร์บางรุ่นของ Linksys ก็นำระบบปฏิบัติการ ลินุกซ์ ที่ดัดแปลงให้เหมาะสมต่อการทำงานมาใช้

ในระยะหลังมีการนำระบบปฏิบัติการ ลินุกซ์ มาพัฒนาและใช้งานบนซูเปอร์คอมพิวเตอร์มากขึ้น ในรายชื่อซูเปอร์คอมพิวเตอร์ที่มีสมรรถนะสูงสุด 500 อันดับ สืบค้นเมื่อเดือนพฤษภาคม พ.ศ. 2548 มีซูเปอร์คอมพิวเตอร์ถึง 371 เครื่องที่ใช้ระบบปฏิบัติการ ลินุกซ์ รวมถึงซูเปอร์คอมพิวเตอร์อันดับที่หนึ่งและสองด้วย

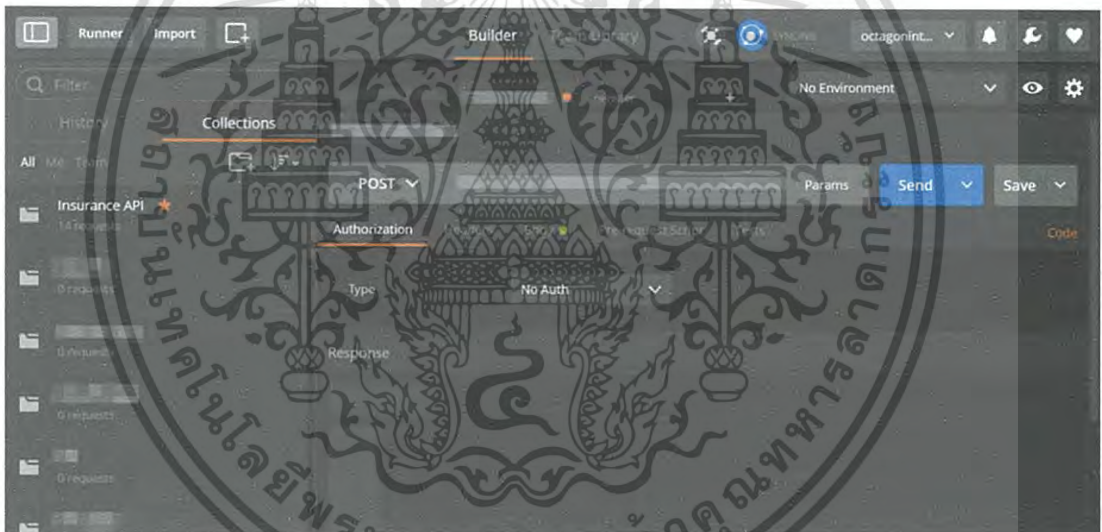
ลินุกซ์ ถือเป็นส่วนสำคัญของซอฟต์แวร์ที่ทำงานบนเซิร์ฟเวอร์เรียกว่า LAMP ย่อมาจาก ลินุกซ์ , Apache, MySQL, Per/PHP/Python ซึ่งเป็นที่นิยมใช้เป็นซอฟต์แวร์ในการพัฒนาเอกสารเว็บเซิร์ฟเวอร์ และพบมากที่สุดระบบหนึ่งของการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16. โปรแกรม Postman

Postman เป็นซอฟต์แวร์ชนิด REST client ที่มีประสิทธิภาพ ใช้สำหรับทดสอบการทำงาน ของ web service สร้างโดย Abhinav Asthana ซึ่งเป็นโปรแกรมเมอร์และนักออกแบบชาวอินเดีย โปรแกรม Postman เป็นโปรแกรมที่ทำให้การทดสอบเว็บเซอร์วิส การพัฒนา และจัดทำเอกสารคู่มือในการใช้งานส่วนต่อประสานโปรแกรมประยุกต์เป็นเรื่องง่าย โปรแกรม Postman เปิดให้ดาวน์โหลดผ่าน Google chrome web store

2.16.1. หลักการทำงาน

โปรแกรม Postman เป็นโปรแกรมที่จำลองตัวเองเป็น client ที่เชื่อมต่อไปยังเว็บเซอร์วิสที่อยู่บนเครื่อง server ด้วยการสร้าง HTTP Request และแสดงผลการทำงานของเว็บเซอร์วิสที่ได้จาก Response ให้ผู้ใช้งานเห็น โปรแกรม Postman สามารถกำหนดค่าต่างๆ ในการส่ง request ได้มากมายหลายแบบทั้ง HTTP Body, HTTP Header, URL, HTTP Method และสามารถเลือกแสดงผลที่ได้ในรูปแบบต่างๆ ทั้ง JSON, XML, HTML หรือแสดงเป็นข้อมูลดิบก็ได้ ปัจจุบันมีผู้ใช้งานโปรแกรม Postman มากกว่า 348,000 คน



ภาพที่ 2.80 ภาพหน้าต่างโปรแกรม Postman

บทที่ 3

ขั้นตอนการดำเนินงาน

ในขั้นตอนการดำเนินงาน จะแบ่งส่วนของการดำเนินงานหลักๆออกได้เป็น 3 ส่วนคือ

- ส่วนออกแบบ (Design Phase)
- ส่วนหน้าบ้าน-โมบิลแอปพลิเคชัน (Frontend Mobile Application)
- ส่วนระบบหลังบ้าน (Backend)

ถัดไปจะกล่าวถึงขั้นตอนการดำเนินงานส่วนออกแบบ ซึ่งจะอธิบายกระบวนการในแต่ละขั้นตอนอย่างละเอียด

3.1. การออกแบบ

3.1.1. เก็บรวบรวมความต้องการของลูกค้า

ขั้นตอนนี้คือการเก็บรวบรวมความต้องการของลูกค้า (Requirement) ก่อนเริ่มการพัฒนาระบบ หรือเริ่มพัฒนาแอปพลิเคชัน ซึ่งกระบวนการนี้เป็นขั้นตอนที่ค่อนข้างสำคัญมาก โดยที่คณะผู้จัดทำและนักพัฒนาที่บริษัทจะพูดคุยกับลูกค้าในเรื่องว่า งานจะเสร็จในระยะเวลาใด ควรใช้เทคโนโลยีแบบใด พูดคุยถึงข้อตกลงต่างๆ ซึ่งในการพัฒนาจะสามารถเริ่มวิเคราะห์ได้จากขั้นตอนนี้ โดยได้รวบรวมความต้องการของลูกค้าได้ดังนี้

- ต้องการให้ผู้ใช้งานซื้อประกันผ่านทางแอปพลิเคชันได้ โดยให้ระบบการซื้อบนเว็บไซต์มีความสัมพันธ์กับบนแอปพลิเคชัน
- ต้องการให้แอปพลิเคชันสามารถแจ้งเตือนได้
- ถ่ายรูปจุดเสียหายของรถได้
- ต้องการให้มีบัตรประกันแบบดิจิทัลการ์ดบนมือถือ
- ต้องการให้หน้าแสดงข้อมูลของผู้ซื้อประกัน (Profile) มีข้อมูลครบถ้วนตรงตามเว็บไซต์
- ต้องการให้มีแผนที่แสดง อุโมงค์มรดก โรงพยาบาลให้เครือ และโรงพยาบาลนอกเครือ
- ต้องการให้สามารถโทรติดต่อเจ้าหน้าที่ได้ในกรณีฉุกเฉินผ่านแอปพลิเคชัน
- ต้องการให้สามารถแชร์โปรโมชั่นต่างๆ ผ่านทางโซเชียลเน็ตเวิร์คได้

ทั้งหมดนี้คือความต้องการหลักของลูกค้าที่ทางคณะผู้จัดทำได้เข้าประชุมและเก็บรวบรวมมาได้ ซึ่งจริงๆแล้วลูกค้ายังไม่มีไอเดียที่คิดไว้มากเท่าไร ลูกค้าต้องการให้คณะผู้จัดทำช่วยคิดฟีเจอร์ใหม่ๆที่คิดว่าควรจะมีให้ในแอปพลิเคชัน และขึ้นโครงสร้างของแอปมาให้ลูกค้าได้ดูก่อน

3.1.2. ระดมความคิด เพื่อสร้างฟีเจอร์ให้แอปพลิเคชัน

ขั้นตอนนี้เป็นขั้นตอนการรวบรวมความคิด (Brainstorm) จากทุกคนในทีมพัฒนา เพื่อให้ได้ฟีเจอร์ของแอปพลิเคชันที่ตรงตามต้องการของลูกค้าและสามารถพัฒนาได้จริง โดยทุกคนจะได้รับกระดาษแผ่นเล็กๆคนละ 5 แผ่นหรือมากกว่านั้นหากต้องการเขียนเพิ่ม เพื่อเขียนสิ่งที่อยากให้ออปพลิเคชันนี้มี แล้วนำไปติดไว้ที่บอร์ด เพื่อดูว่าฟีเจอร์ของใครน่าสนใจและสามารถทำได้จริง ซึ่งคณะผู้จัดทำสรุปฟีเจอร์ที่จะทำบนแอปพลิเคชันได้ดังนี้

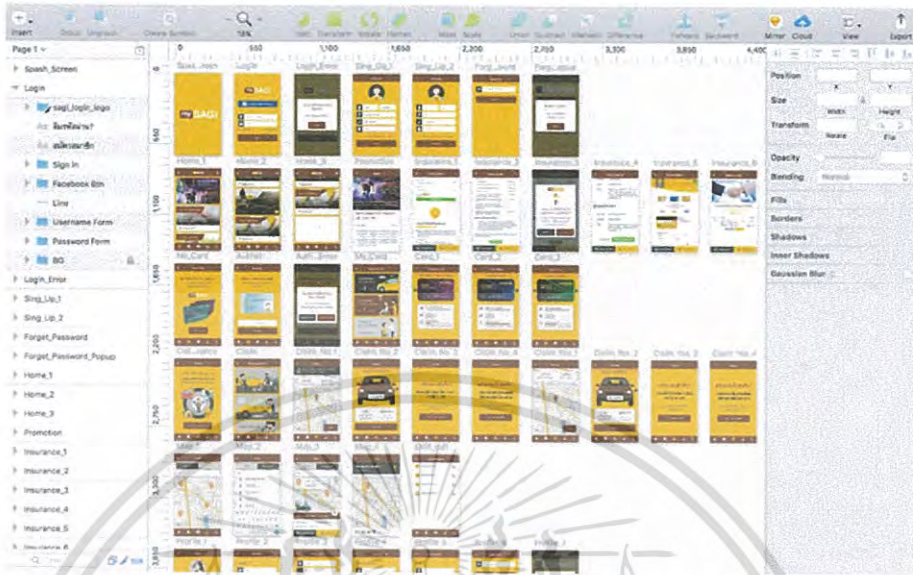
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และแจ้งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จะให้เมนูแรกมีสำหรับขายประกันออนไลน์ โดยแสดงเป็นลักษณะการ์ด (Card View) แบ่งแยกประเภทของประกันโดยเรียงตามประกันที่ผู้ใช้ซื้อมากที่สุด
- ในระหว่างการซื้อ จะมีพีเจอร์แสดงเบอร์ของเจ้าหน้าที่ให้โทรไปสอบถามรายละเอียดได้ หรือ กดสนใจเพื่อให้เจ้าหน้าที่โทรมาหาก็ได้
- เมนูที่สองจะทำให้บัตรประกันเป็นแบบบัตรดิจิทัลการ์ดที่อยู่บนแอปพลิเคชัน โดยบัตรใบนี้เสมือนบัตรจริง ซึ่งสามารถต่ออายุออนไลน์ได้ ดูความคุ้มครองได้ ดูอายุของกรมธรรม์ได้ สามารถสแกนบาร์โค้ด (Bar Code) หรือคิวอาร์โค้ด (QR Code) ได้เพื่อดูข้อมูลผู้มีประกันในกรณีที่ใช้กับโรงพยาบาล
- เมนูที่สามเป็นเมนูแจ้งเคลม โดยแบ่งการแจ้งเคลมเป็นสองประเภทคือ มีคู่กรณีและไม่มีคู่กรณี สามารถถ่ายรูปจุดเสียหายและสภาพของรถตามที่ทางบริษัทประกันต้องการ โดยการแจ้งนี้ไม่มีสองรูปแบบ คือ จำเป็นต้องให้เจ้าหน้าที่มาที่จุดเกิดเหตุ และไม่จำเป็นต้องให้เจ้าหน้าที่มาที่จุดเกิดเหตุ ตามรูปแบบของการแจ้งเคลม
- เมนูที่สี่เป็นหน้าแสดงแผนที่ อยู่ซ่อมรถ โรงพยาบาลให้เครื่อ และโรงพยาบาลนอกเครื่อ โดยจะมีแท็บแยกประเภทเพื่อแสดงพินของสถานที่ที่มีสีต่างกัน รวมถึงจะมีช่องค้นหาสถานที่
- หน้าโปรไฟล์มีข้อมูลทุกอย่างตรงตามเว็บไซต์ แต่จะแบ่งตามประเภทของข้อมูล

3.1.3. สร้างแบบร่าง ให้กับแอปพลิเคชัน

ขั้นตอนนี้คณะผู้จัดทำเลือกใช้โปรแกรม Sketch ในการสร้างแบบร่าง (Wireframe) โดยจะนำพีเจอร์ที่คณะผู้จัดทำสรุปได้ทั้งหมดมาสร้างเป็น User Interface (UI) เบื้องต้น โดยขั้นตอนนี้จะไม่มีการใช้สีที่ทำให้เกิดความสวยงามของ UI จะเป็นการวางพีเจอร์บนแอปพลิเคชันเท่านั้น ซึ่งจะต้องใช้ความรู้ด้านประสบการณ์การใช้งานแอปพลิเคชันเข้ามาช่วย ได้แก่ การวิเคราะห์ว่าปุ่มต่างๆ ควรจะอยู่ตรงส่วนไหนของหน้าจอเพื่อให้ง่ายต่อการใช้งาน และวิเคราะห์ว่าหากผู้ใช้กดปุ่มดังกล่าวนี้แล้วจะเชื่อมต่อไปยังหน้าจอไหนหรือจะมีการตอบสนองกับผู้ใช้งานอย่างไร โดยแต่ละหน้าจอจะมีความสัมพันธ์กันดังภาพที่ 3.1

3.1.4. สร้างแบบร่าง (Wireframe) ให้กับแอปพลิเคชัน



ภาพที่ 3.2 ตัวอย่างแบบจำลองแอปพลิเคชัน (Mockup)

จากภาพที่ 3.2 ขั้นตอนนี้เป็นขั้นตอนที่ทำหลังจากที่แบบร่าง (Wireframe) ไม่มีการเปลี่ยนแปลงแล้ว อาจมีการเพิ่มความต้องการของลูกค้าบ้างในส่วนนี้ แต่คณะผู้จัดทำจะไม่แก้ไขส่วนที่ส่งผลกระทบต่อโครงสร้างหลักที่ได้ออกแบบไว้ในแบบร่าง (Wireframe) โดยในขั้นตอนการสร้างแบบจำลองของแอปพลิเคชันนี้คณะผู้จัดทำจะทำให้ยูสเซอร์อินเตอร์เฟส (User Interface) ที่ได้วางไว้เกิดความสวยงามมากขึ้นจากการใส่สีจริง ใส่ข้อมูลจริง ใส่ภาพที่ต้องการใช้ลงไปจริงให้ตรงตามความต้องการของลูกค้า

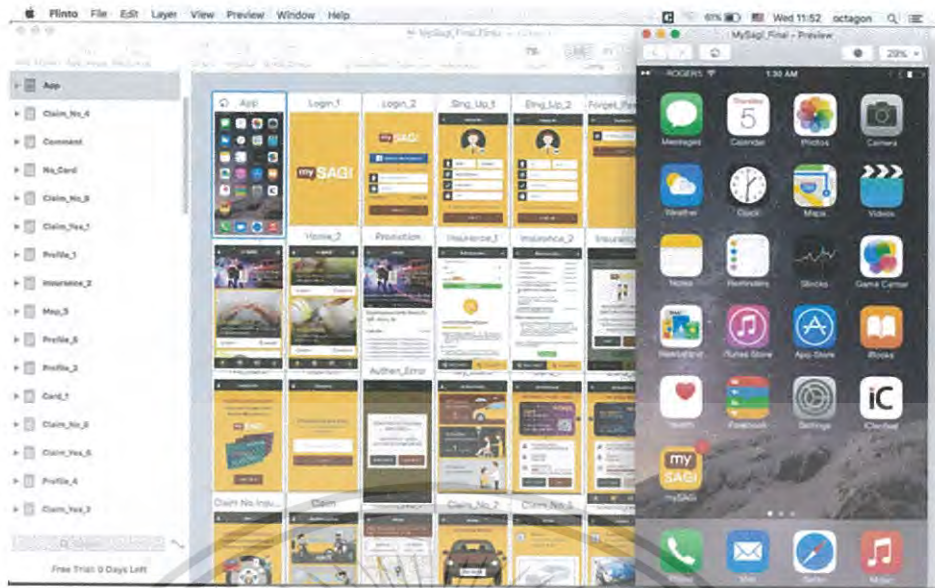
3.1.5. สร้างแบบจำลอง เพื่อทดสอบพฤติกรรมการใช้งานแอปพลิเคชัน



ภาพที่ 3.3 ตัวอย่างโปรแกรม Flinto

ขั้นตอนนี้เป็นกรนำเอาแบบจำลองแอปพลิเคชัน (Mockup) ที่ได้สร้างไว้มาทำให้เกิดการตอบสนองต่อผู้ใช้งานเพื่อใช้ในการทดสอบประสบการณ์การใช้งานแอปพลิเคชันของผู้ใช้ โดยโปรแกรมที่คณะผู้จัดทำเลือกมาใช้ คือโปรแกรม Flinto โปรแกรมนี้มีความง่ายตรงที่สามารถนำเข้าไฟล์งานจากโปรแกรม Sketch ได้เลย และโปรแกรมนี้ก็สามารถใช้ได้เฉพาะบนระบบปฏิบัติการแม็ค (MacBook) เท่านั้น

จากภาพที่ 3.3 เป็นหน้าตาของโปรแกรม Flinto ซึ่งจะมีหน้าตาคล้ายกับโปรแกรม Sketch ในระดับหนึ่ง แต่โปรแกรมตัวนี้ไม่มีความสามารถในการวาดหรือสร้างวัตถุต่างๆแบบโปรแกรม Sketch แต่มีความสามารถในการสร้างแอนิเมชันให้กับวัตถุที่สร้างขึ้นจากโปรแกรม Sketch เพื่อให้แอปพลิเคชันสามารถตอบสนองกับผู้ใช้งานได้เมื่อผู้ใช้งานกดที่ปุ่มต่างๆ ปิดหน้าจอไปทางซ้ายมือหรือขวามือ รวมไปถึงการเลื่อนหน้าจอขึ้นลง เป็นต้น ซึ่งสามารถสร้างแอนิเมชันได้ค่อนข้างอิสระ และตั้งค่าต่างๆได้ค่อนข้างละเอียดมาก รวมถึงสามารถนำแอนิเมชันที่ใส่แล้วนำกลับมาใช้ใหม่กับหน้าจออื่นๆได้ด้วย



ภาพที่ 3.4 ตัวอย่างหน้าจอตอนแสดงผลแอนิเมชันโปรแกรม Flinto

จากภาพที่ 3.4 หน้าจอทางขวาเป็นหน้าจอแสดงผลแอนิเมชันหลังจากสร้างเสร็จ ซึ่งสามารถดูได้จากบนเครื่องแม็ค (MacBook) หรืออ็อปโทดโฟลด์นี้จากเครื่องไปยังแอปพลิเคชัน Flinto เพื่อดูผ่านมือถือได้โดยไม่ต้องเชื่อมต่ออินเทอร์เน็ตตลอดเวลา แต่จะต้องเป็นมือถือที่เป็นระบบปฏิบัติการ iOS เท่านั้น ได้แก่ iPhone 5, iPhone 6 หรือ iPhone 7 เป็นต้น จึงจะสามารถอ็อปโทดโฟลด์ขึ้นไปเพื่อทดสอบใช้งานแอนิเมชันได้

นำเสนอโปรโตไทป์ พร้อมสัมภาษณ์เรื่องการใช้งานแอปพลิเคชัน กับลูกค้าพร้อมรับฟังคำแนะนำมาปรับปรุงดีไซน์ให้ถูกต้องตามความต้องการของลูกค้าก่อนลงมือพัฒนาจริง

ขั้นตอนการทำการทดสอบประสบการณ์ผู้ใช้งาน (Usability Test) เป็นขั้นตอนที่คณะผู้จัดทำจะนำโปรโตไทป์ที่สร้างขึ้นไปให้ผู้ใช้งานทดสอบหรือลองใช้งานแอปพลิเคชัน โดยเตรียมการ คือ ทางทีมดีไซน์จะสร้างเอกสารสำหรับทดสอบขึ้นมาให้เท่ากับจำนวนผู้ทดสอบ ซึ่งในเอกสารจะมีคำถามที่ทางทีมดีไซน์ต้องถามผู้ใช้งานไปพร้อมกับให้ผู้ใช้งานใช้งานแอปพลิเคชัน เช่น ถ้าหากลูกค้าจะซื้อประกันลูกค้าต้องทำอะไร เป็นต้น ซึ่งจะถามคำถามในลักษณะนี้กับทุกๆ หน้าจอที่ทีมดีไซน์ได้ทำโปรโตไทป์ไปจนหมด แล้วให้ผู้ใช้งานประเมินพร้อมให้คำแนะนำในเรื่องของการใช้งานของแต่ละหน้าจอ หากผู้ใช้งานต้องการจะเสนอแนะ สุดท้ายคณะผู้จัดทำจะมีคำถามปลายเปิดถามผู้ใช้งานเพื่อสำรวจว่าผู้จัดทำและพัฒนา ควรจะสร้างแอปพลิเคชันนี้ขึ้นมาหรือไม่ เพราะจะวัดว่าถ้าหากแอปพลิเคชันนี้เสร็จ จะมีคนใช้งานหรือไม่ เช่น ถ้าหากลูกค้ามีโอกาสได้ใช้งานแอปพลิเคชันนี้ลูกค้าจะใช้งานมันหรือไม่, ลูกค้าคิดอย่างไรกับการใช้งานบัตรประกันภัยแบบดิจิทัลการ์ด เป็นต้น ดังภาพที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.5 บรรยากาศการทำทดสอบประสบการณ์ผู้ใช้งาน (Usability Test) ที่บริษัทประกันภัย

หลังจากทำทดสอบประสบการณ์ผู้ใช้งาน (Usability Test) คณะผู้จัดทำจะต้องนำคำแนะนำของผู้ใช้งานมาเข้าที่ประชุมที่บริษัทเพื่อคุยกันว่าอะไรที่ทีมพัฒนาทำได้และทำไม่ได้ เพื่อที่จะได้รายงานลูกค้าถึงให้ทราบถึงข้อจำกัดในการพัฒนาและลดความเสี่ยงที่จะเกิดโอกาสโปรเจกต์ล้มเหลว ก่อนที่จะสร้าง สุดท้ายคณะผู้จัดทำสร้างไฟล์นอลตีไซน์ (Final Design) เพื่อเป็นต้นแบบของแอปพลิเคชันให้นักพัฒนาลงมือพัฒนาจริงในเวอร์ชันแรก หลังจากสร้างไฟล์นอลตีไซน์ (Final Design) เสร็จคณะผู้จัดพิมพ์ทีมดีไซน์ที่จะต้องส่งให้ทางแผนกรับผิดชอบด้านการตลาดของบริษัทประกันภัยตรวจสอบเรื่องสีของผลิตภัณฑ์ เรื่องโลโก้ของผลิตภัณฑ์ และอื่นๆที่เกี่ยวข้องกับผลิตภัณฑ์ก่อนเข้าสู่ช่วงโปรแกรมมิ่งหรือลงมือพัฒนาแอปพลิเคชันจริง โดยทีมดีไซน์จะทำการอัปเดตไฟล์งานที่ออกแบบทั้งหมดขึ้นไปบนโปรแกรม Zeplin ดังภาพที่ 3.6



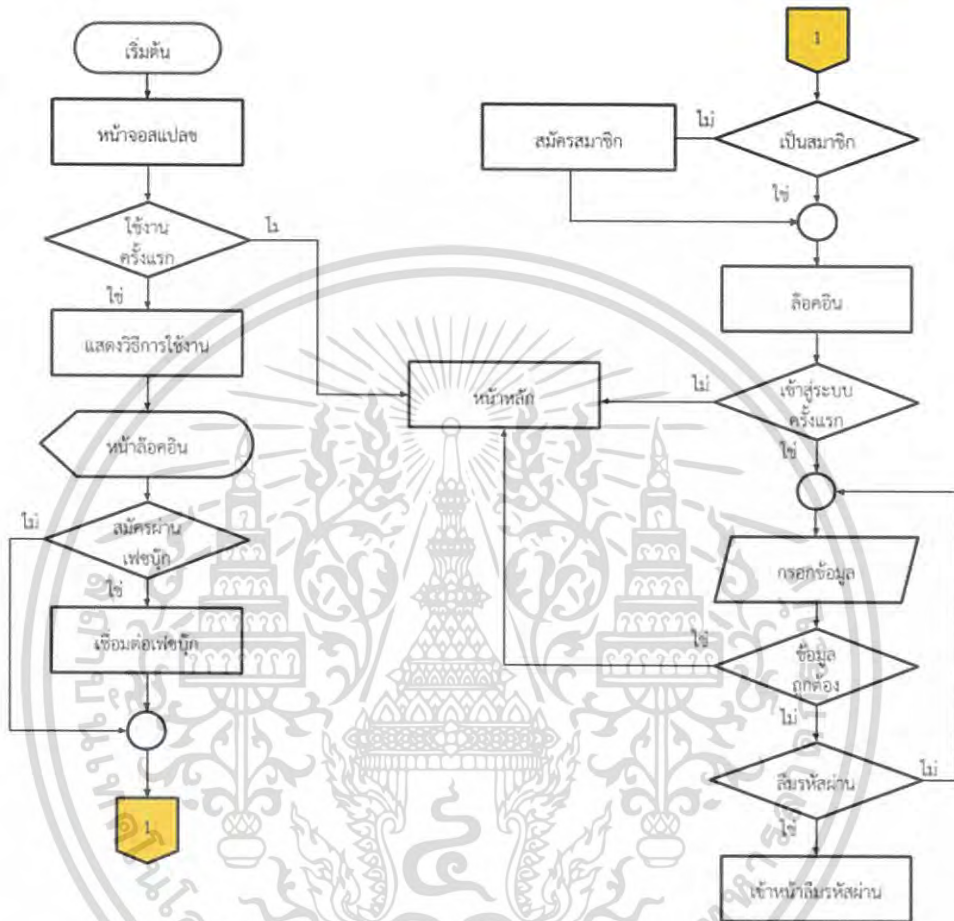
ภาพที่ 3.6 ตัวอย่างหน้าจอแอปพลิเคชันที่อัปขึ้นบนโปรแกรม Zeplin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2. ส่วนหน้าบ้าน-โมบายล์แอปพลิเคชัน

3.2.1. ออกแบบขั้นตอนการทำงานของโปรแกรม

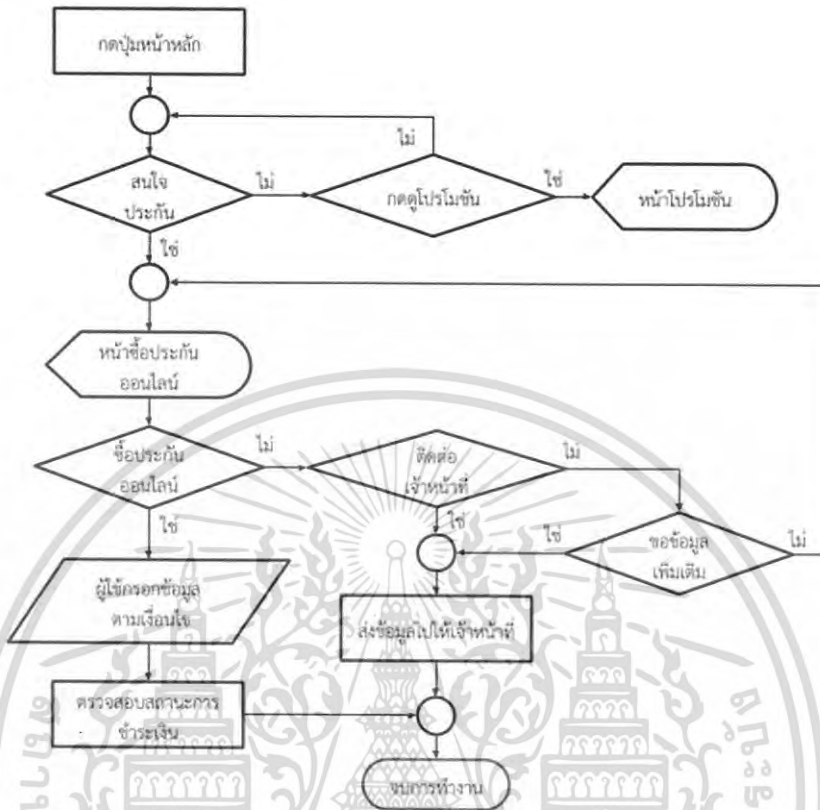
3.2.1.1. ผังงานการทำงานหน้าลือคอิน



ภาพที่ 3.7 ผังงานการทำงานหน้าลือคอิน

จากภาพที่ 3.7 แสดงผังงานขั้นตอนการทำงานหน้าลือคอิน เริ่มจากเข้าสู่หน้าจอสแปลชสกรีน ระบบจะทำการตรวจสอบว่าใช้งานแอปพลิเคชันครั้งแรกหรือไม่ ถ้าเข้าใช้งานครั้งแรกระบบจะไปที่หน้าลือคอิน ถ้าผู้ใช้ยังไม่ได้เป็นสมาชิกจะต้องทำการสมัครสมาชิกก่อน ซึ่งสามารถสมัครผ่านเฟซบุ๊กหรือสมัครสมาชิกแบบปกติ หลังจากนั้นให้ทำการลือคอินเข้าสู่ระบบ กรณีที่ข้อมูลไม่ถูกต้องผู้ใช้ต้องกรอกข้อมูลใหม่อีกครั้ง หรือถ้าผู้ใช้ลืมรหัสผ่านให้กดเข้าไปที่ลืมรหัสผ่านเพื่อให้ระบบส่งรหัสผ่านมาทางอีเมล เมื่อผู้ใช้ลือคอินสำเร็จ ก็จะไปยังหน้าหลัก ทั้งนี้ถ้าผู้ใช้เข้าแอปพลิเคชันครั้งต่อไปก็จะเข้าสู่หน้าหลักทันที

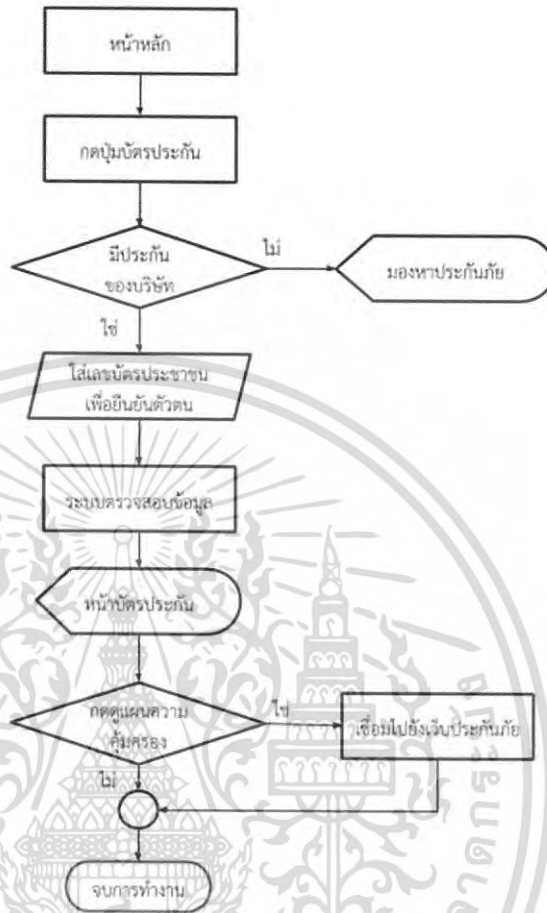
3.2.1.2. ผังงานการทำงานหน้าหลัก



ภาพที่ 3.8 ผังงานการทำงานหน้าหลัก

จากภาพที่ 3.8 แสดงผังงานขั้นตอนการทำงานหน้าหลักเมื่อผู้ใช้เข้าสู่ระบบเรียบร้อยแล้วจะเข้าสู่หน้า หลักทันที ภายในหน้านี้จะประกอบไปด้วยส่วนที่เป็นโปรโมชั่นและประกันภัยภาพแบบต่างๆ ซึ่งสามารถกดเข้าไปดูข้อมูลต่างๆได้ ในกรณีที่ผู้ใช้สนใจประกันและต้องการซื้อประกันภัยนั้น ระบบจะทำการแสดงเว็บวิวขึ้นมาให้ผู้ใช้กรอกข้อมูลตามเงื่อนไข และชำระเงินให้ถูกต้อง ก็เป็นอันเสร็จสมบูรณ์ ส่วนผู้ใช้ที่ยังไม่ได้ตัดสินใจหรือต้องการทราบรายละเอียดเพิ่มเติม สามารถติดต่อเจ้าหน้าที่ขายประกันของบริษัท หรือ ขอข้อมูลเพิ่มเติม ระบบก็จะทำการส่งข้อมูลของผู้ใช้ไปยังเจ้าหน้าที่ให้ติดต่อกลับมา

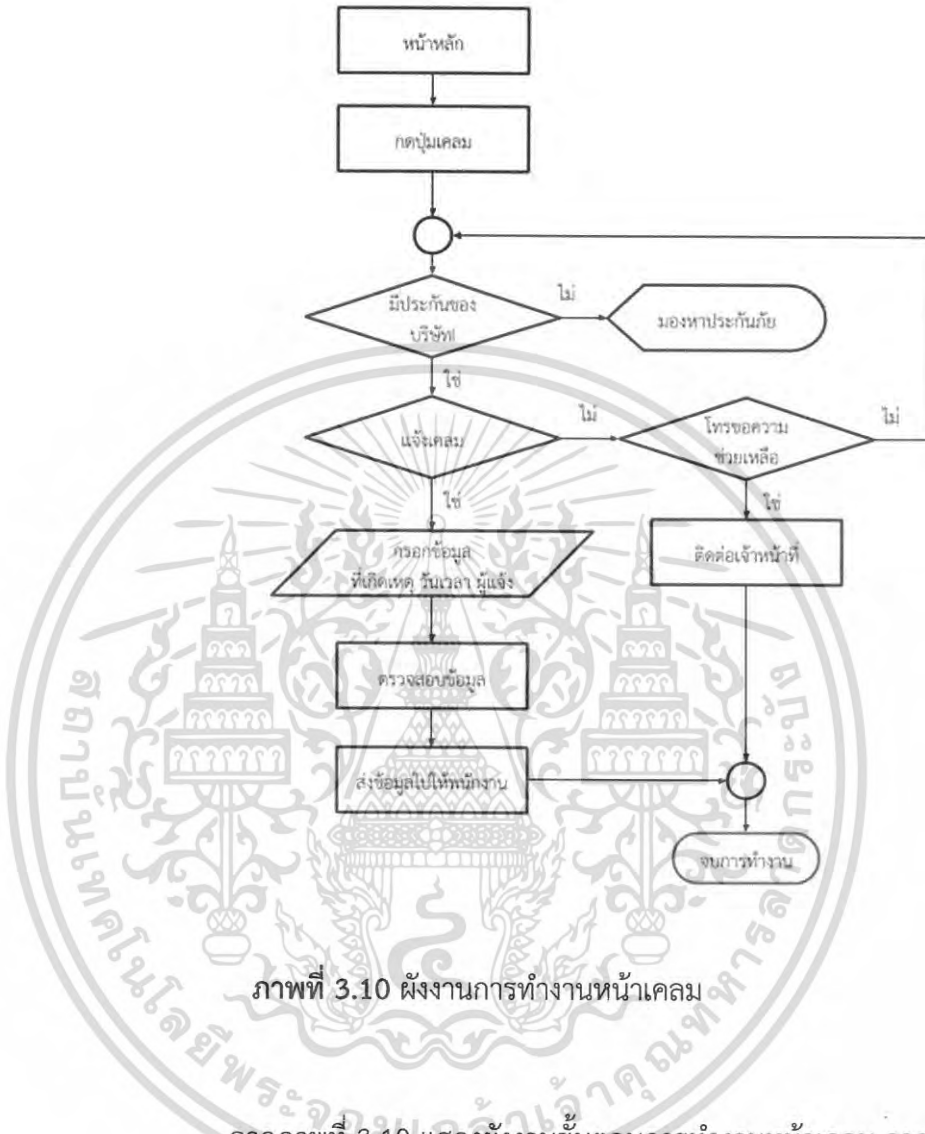
3.2.1.3. ผังงานการทำงานหน้าบัตรประกัน



ภาพที่ 3.9 ผังงานการทำงานหน้าบัตรประกัน

จากภาพที่ 3.9 แสดงผังงานขั้นตอนการทำงานหน้าบัตรประกัน จากหน้าหลัก เมื่อผู้ใช้กดปุ่มบัตรประกันระบบจะแสดงหน้าบัตรประกัน ถ้ายังไม่มีประกันภัย ระบบจะแสดงหน้ามองหาประกันภัย เมื่อมีประกันภัยแล้วระบบจะให้ยืนยันการใช้งานบัตรจำลองโดยการกรอกเลขบัตรประชาชนเพียงครั้งแรกที่ใช้งาน ถ้าข้อมูลยืนยันถูกต้อง ก็จะแสดงหน้าบัตรจำลองขึ้นมา ผู้ใช้สามารถกดดูแผนความคุ้มครองได้

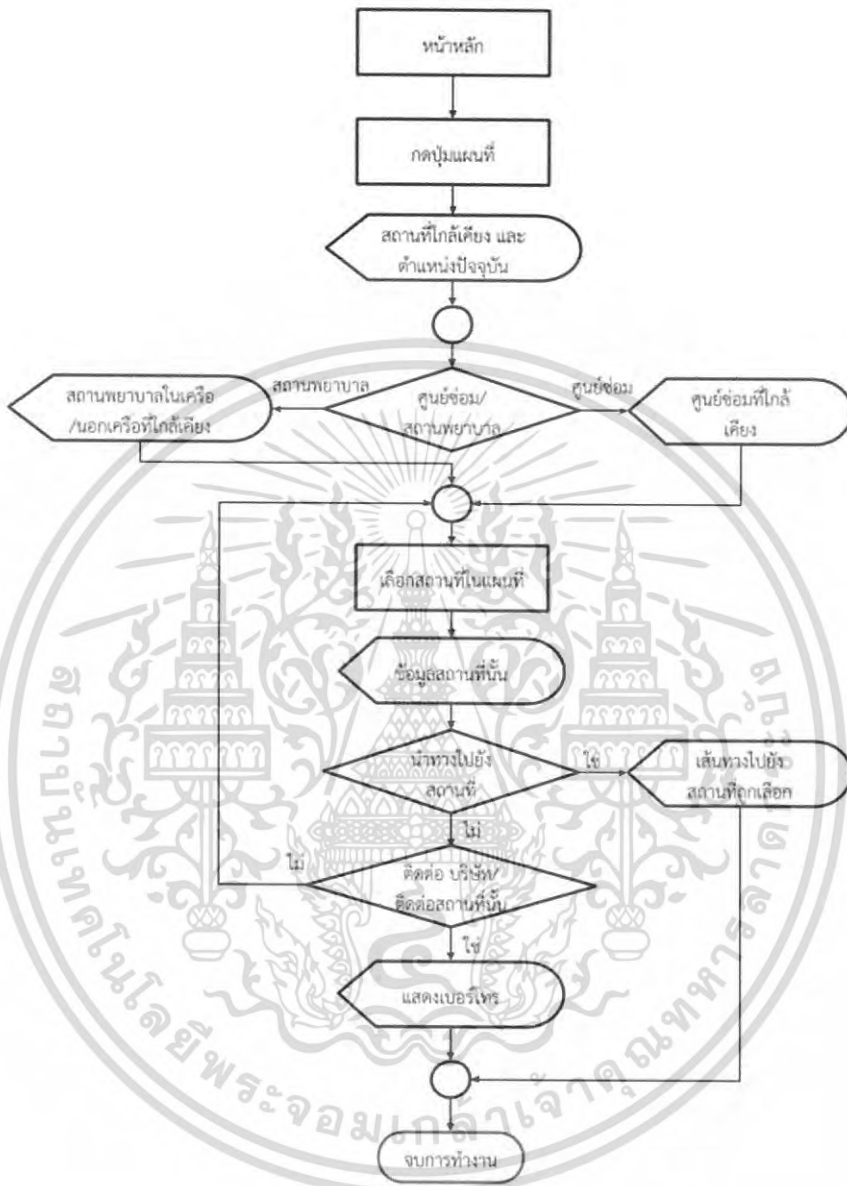
3.2.1.4. ผังงานการทำงานหน้าเคลม



ภาพที่ 3.10 ผังงานการทำงานหน้าเคลม

จากภาพที่ 3.10 แสดงผังงานขั้นตอนการทำงานหน้าเคลม จากหน้าหลัก เมื่อผู้ใช้กดปุ่มเคลม ระบบจะแสดงหน้าเคลม ก่อนที่จะใช้งานการแจ้งเคลมได้นั้น ผู้ใช้จะต้องมีประกันภัยก่อน ถ้าผู้ใช้อยังไม่มีประกันภัย ระบบจะแสดงหน้ามองหาประกันภัย แต่ถ้ามีแล้วสามารถแจ้งเคลมหรือโทรขอความช่วยเหลือจากเจ้าหน้าที่ กรณีต้องการแจ้งเคลม ผู้ใช้ก็ต้องกรอกข้อมูลให้ครบตามที่กำหนด จากนั้นระบบจะส่งข้อมูลไปให้พนักงานเพื่อให้พนักงานไปยังจุดเกิดเหตุ

3.2.1.5. ผังงานการทำงานหน้าแผนที่



ภาพที่ 3.11 ผังงานการทำงานหน้าแผนที่

จากภาพที่ 3.11 แสดงผังงานขั้นตอนการทำงานหน้าแผนที่ จากหน้าหลัก เมื่อผู้ใช้กดปุ่มแผนที่ ระบบจะแสดงหน้าแผนที่ ขึ้นมา ซึ่งเป็นที่อยู่ปัจจุบันของผู้ใช้และสถานที่ใกล้เคียง ผู้ใช้สามารถเลือกได้ว่าจะค้นหาสถานพยาบาลหรือศูนย์ซ่อม จากนั้นเลือกสถานที่ที่ต้องการ โดยระบบจะแสดงข้อมูล เช่น ชื่อสถานที่ ที่อยู่ ระยะทาง เวลาเดินทาง เบอร์ติดต่อ สามารถกดนำทางไปยังสถานที่นั้นหรือติดต่อกับบริษัท

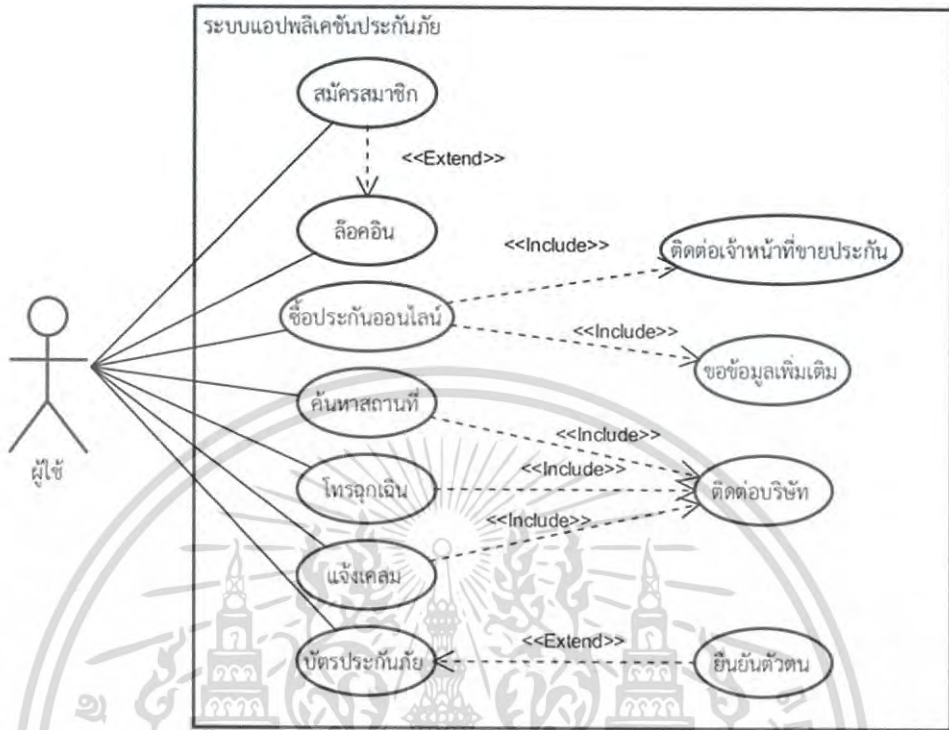
3.2.1.6. ผังงานการทำงานหน้าโทรฉุกเฉิน



ภาพที่ 3.12 ผังงานการทำงานหน้าโทรฉุกเฉิน

จากภาพที่ 3.12 แสดงผังงานขั้นตอนการทำงานหน้าโทรฉุกเฉิน จากหน้าหลักเมื่อผู้ใช้กดปุ่มโทรฉุกเฉิน ระบบจะแสดงหน้าเบอร์ฉุกเฉิน เช่น เบอร์ติดต่อบริษัท เรียกรถพยาบาล ติดต่อสถานีตำรวจ เป็นต้น ผู้ใช้จะทำการกดเบอร์ที่ต้องการเพื่อโทรออก

3.2.2. แผนภาพยูสเคส (Use case diagram)



ภาพที่ 3.13 แผนภาพยูสเคสแอปพลิเคชันประกันภัย

ดังนั้น

จากแผนภาพยูสเคสข้างต้น จะเห็นได้ว่าระบบแอปพลิเคชันประกันภัยมี 10 ยูสเคส

3.2.2.1. สมัครสมาชิก

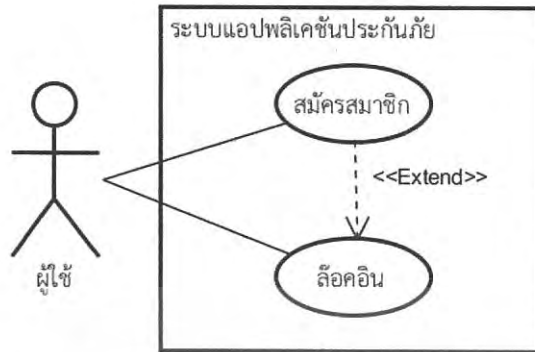


ภาพที่ 3.14 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการสมัครสมาชิก

ตารางที่ 3.1 ยูสเคสสมัครสมาชิก

Use case ID	01	
Use Case Name	สมัครสมาชิก	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย สมัครสมาชิกทางเว็บแล้ว	
Post-Condition	ผู้ใช้สมัครสมาชิกสำเร็จ สามารถล็อกอินได้	
Brief Description	ผู้ใช้นั้นขอสมัครสมาชิก	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1.เข้าสู่หน้าจอล็อกอิน 3. กดสมัครสมาชิก 5. กรอกรายละเอียดในการสมัครสมาชิก <ol style="list-style-type: none"> 5.1 สมัครผ่าน Facebook <ol style="list-style-type: none"> 5.1.1) ขึ้นหน้าให้กรอกข้อมูลแต่มีข้อมูลบางส่วนให้แล้ว 5.2 สมาชิกใหม่ <ol style="list-style-type: none"> 5.1.2) ขึ้นหน้าให้กรอกข้อมูล 7. สามารถนำ Username ไปใช้ในการล็อกอิน ได้ 	<ol style="list-style-type: none"> 2.แสดงหน้าจอล็อกอิน 4. แสดงหน้าจอสมัครสมาชิก 6. ตรวจสอบ Username ว่าซ้ำไหม <ol style="list-style-type: none"> 6.1) ถ้า Username ไม่ซ้ำ <ol style="list-style-type: none"> 6.1.1) สมัครสมาชิกสำเร็จ 6.2) ถ้า Username ซ้ำ <ol style="list-style-type: none"> 6.2.1) แจ้งเตือนว่า Username ซ้ำ
Exceptions	<ol style="list-style-type: none"> 1. กรอกข้อมูลไม่ครบ 2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต 3. Username ซ้ำ 	

3.2.2.2. ล็อกอิน

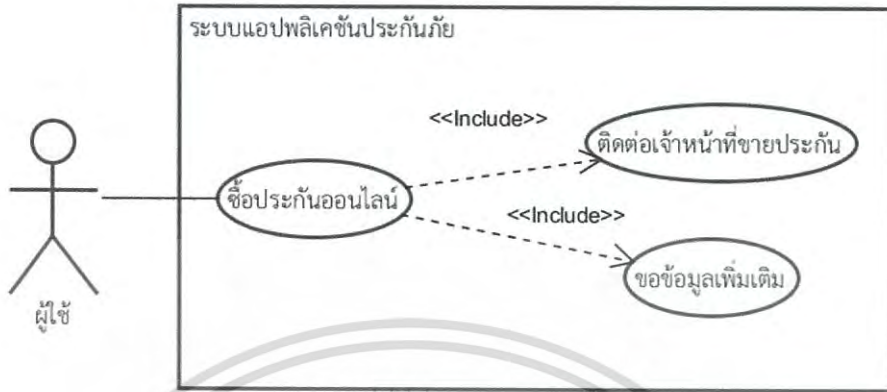


ภาพที่ 3.15 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการล็อกอิน

ตารางที่ 3.2 ยูสเคสล็อกอิน

Use case ID	02	
Use Case Name	ล็อกอิน	
Actor	ผู้ใช้	
Pre-Condition	ต้องสมัครสมาชิกก่อน	
Post-Condition	ผู้ใช้สามารถใช้งานแอปพลิเคชันได้	
Brief Description	ผู้ใช้ยืนยันตัวตนเข้าสู่ระบบ	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. เข้าสู่หน้าจอล็อกอิน 3. กรอกข้อมูล Username และ Password หรือ เชื่อมต่อผ่าน Facebook 4. ล็อกอิน 6. สามารถใช้งานแอปพลิเคชันได้ 	<ol style="list-style-type: none"> 2. แสดงหน้าจอล็อกอิน 5. ตรวจสอบ <ol style="list-style-type: none"> 5.1) Username หรือ Password ผิด <ol style="list-style-type: none"> 5.1.1) แสดงข้อความแจ้งเตือน 5.2) Username หรือ Password ถูกต้อง <ol style="list-style-type: none"> 5.2.1) แสดงหน้าจอหลัก 5.3) ถ้าเชื่อมต่อผ่าน Facebook ระบบเชื่อมโยงไปยัง Facebook
Exceptions	<ol style="list-style-type: none"> 1. กรอกข้อมูลไม่ครบ 2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต 	

3.2.2.3. ซื้อประกันออนไลน์

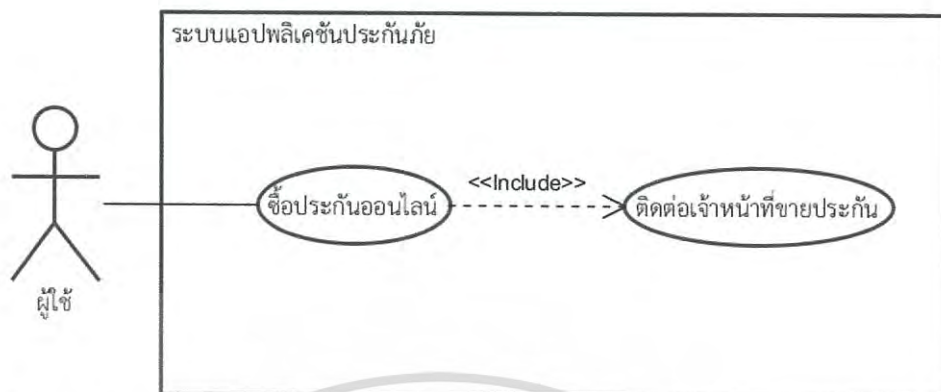


ภาพที่ 3.16 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการซื้อประกันออนไลน์

ตารางที่ 3.3 ยูสเคสซื้อประกันออนไลน์

Use case ID	03	
Use Case Name	ซื้อประกันออนไลน์	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว	
Post-Condition	ผู้ซื้อประกันคุ้มครอง สามารถใช้งานบัตรได้	
Brief Description	ผู้ใช้ซื้อประกันออนไลน์ได้ โดยการกรอกข้อมูลตามเงื่อนไขที่กำหนดให้ครบ	
Flow of Event	Actor	System
	1. เข้าสู่แอปพลิเคชัน 2. กดไปที่สนใจประกันนี้ 4. ผู้ใช้ซื้อประกันผ่านเว็บวิว 7. ผู้ใช้สามารถใช้งานบัตรได้	3. ระบบเปิดหน้าเว็บไซต์ประกัน ขึ้นมา 5. ระบบตรวจสอบข้อมูลฐานข้อมูล 6. ระบบส่งอีเมลยืนยันการซื้อ
Exceptions	3. ไม่ได้เชื่อมต่ออินเทอร์เน็ต 4. ข้อมูลบัตรเครดิตไม่ถูกต้อง	

3.2.2.4. ติดต่อเจ้าหน้าที่หน้าซื้อประกันออนไลน์

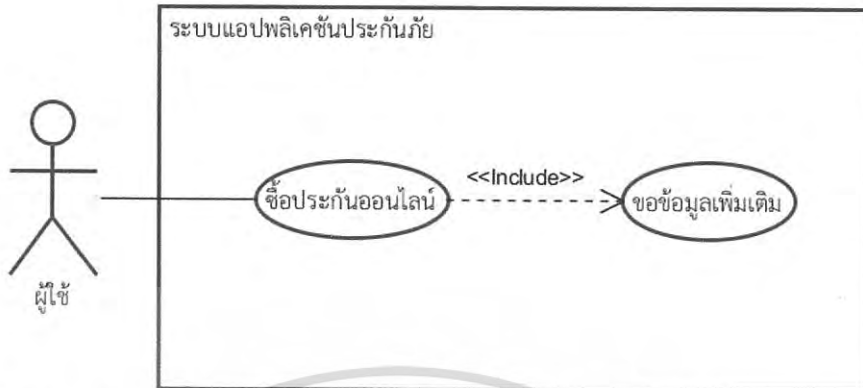


ภาพที่ 3.17 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันติดต่อเจ้าหน้าที่ขายประกัน

ตารางที่ 3.4 ยูสเคสติดต่อเจ้าหน้าที่ขายประกัน

Use case ID	04	
Use Case Name	ติดต่อเจ้าหน้าที่หน้าซื้อประกันออนไลน์	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว เข้าหน้าซื้อประกันออนไลน์	
Post-Condition	ติดต่อเจ้าหน้าที่ผ่านโทรศัพท์	
Brief Description	ผู้ใช้ต้องการซื้อประกันออนไลน์แต่ได้เลือกติดต่อเจ้าหน้าที่	
Flow of Event	Actor	System
	1. เข้าสู่แอปพลิเคชัน 2. กดไปที่สนใจประกันนี้ 3. กดติดต่อเจ้าหน้าที่ 5. กดโทรออก	4. ระบบดึงเบอร์มาจากฐานข้อมูล
Exceptions	2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต	

3.2.2.5. ขอข้อมูลเพิ่มเติม

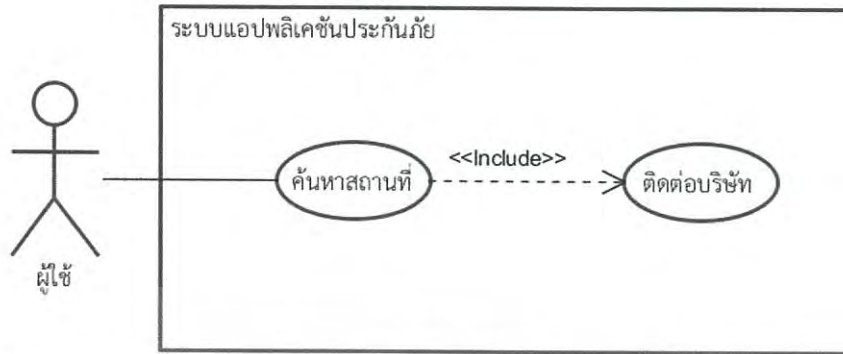


ภาพที่ 3.18 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการขอข้อมูลเพิ่มเติม

ตารางที่ 3.5 ยูสเคสขอข้อมูลเพิ่มเติม

Use case ID	05	
Use Case Name	ขอข้อมูลเพิ่มเติม	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว เข้าหน้าซื้อประกันออนไลน์	
Post-Condition	มีเจ้าหน้าที่ติดต่อมา	
Brief Description	ผู้ใช้ต้องการซื้อประกันออนไลน์แต่ได้เลือกให้เจ้าหน้าที่ติดต่อกลับ	
Flow of Event	Actor	System
	1. เข้าสู่แอปพลิเคชัน 2. กดไปที่สนใจประกันนี้ 3. กดขอข้อมูลเพิ่มเติม	4. ระบบแจ้งเจ้าหน้าที่ผ่านทางอีเมล 5. เจ้าหน้าที่ติดต่อกลับไปยังผู้ใช้
Exceptions	2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต 5. ผู้ใช้ไม่ได้ใช้เมลหรือเบอร์โทรตามข้อมูลที่กรอกไว้	

3.2.2.6. ค้นหาสถานที่

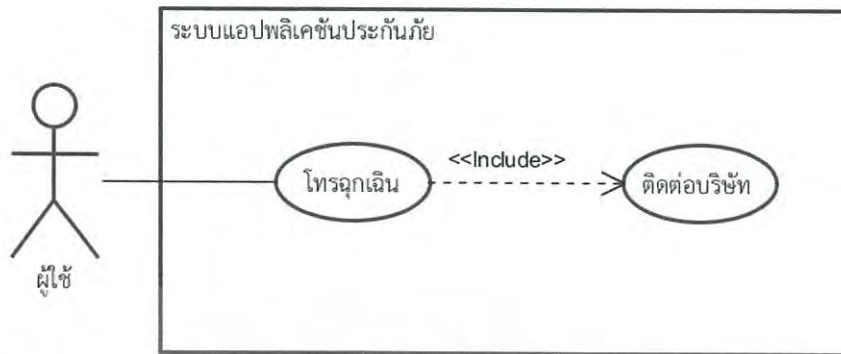


ภาพที่ 3.19 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการค้นหาสถานที่

ตารางที่ 3.6 ยูสเคสค้นหาสถานที่

Use case ID	06	
Use Case Name	ค้นหาสถานที่	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว	
Post-Condition	ผู้ใช้ค้นหาสถานที่	
Brief Description	แสดงสถานที่	
Flow of Event	Actor	System
Exceptions	3. ไม่ได้เชื่อมต่ออินเทอร์เน็ต	

3.2.2.7. โทรฉุกเฉิน

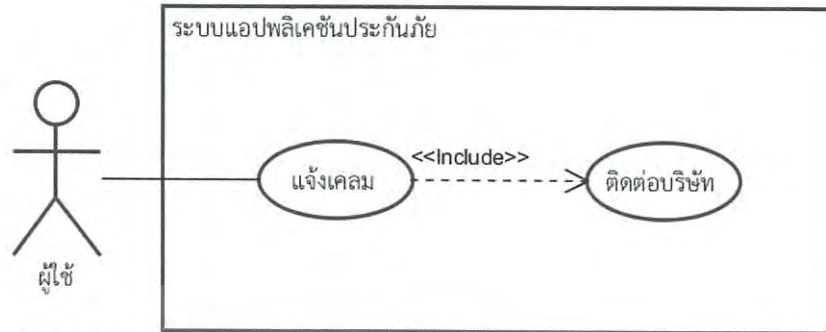


ภาพที่ 3.20 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการโทรฉุกเฉิน

ตารางที่ 3.7 ยูสเคสโทรฉุกเฉิน

Use case ID	07	
Use Case Name	โทรฉุกเฉิน	
Actor	ผู้ใช้, เจ้าหน้าที่คอลเซ็นเตอร์, จนท.สถานพยาบาล, จนท.อยู่ซ่อม	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว เข้าหน้าโทรฉุกเฉิน	
Post-Condition	ติดต่อกับเจ้าหน้าที่	
Brief Description	ผู้ใช้ต้องการติดต่อขอความช่วยเหลือ จากเจ้าหน้าที่	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. เข้าสู่แอปพลิเคชัน 2. กดปุ่มโทรฉุกเฉิน 3. กดเบอร์ฉุกเฉินที่ต้องการ 	<ol style="list-style-type: none"> 4. ค้นหาเบอร์และโทรออก
Exceptions		

3.2.2.8. แจ็งเคลม



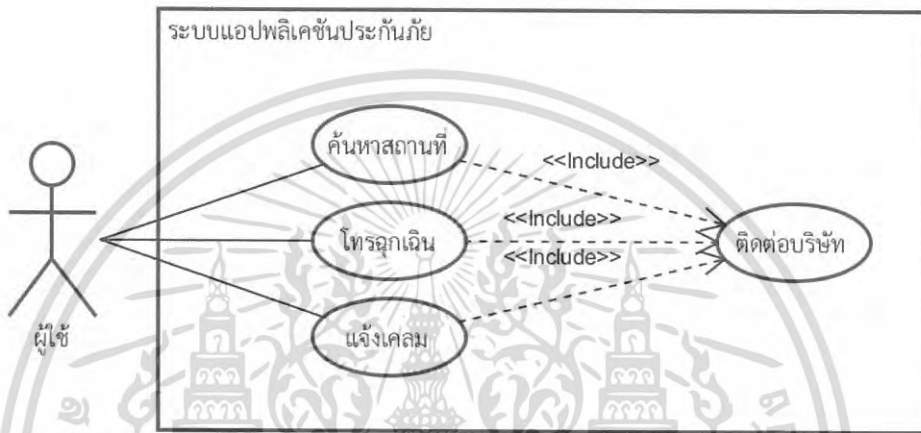
ภาพที่ 3.21 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการแจ็งเคลม

ตารางที่ 3.8 ยูสเคสแจ็งเคลม

Use case ID	08	
Use Case Name	แจ็งเคลม	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว	
Post-Condition	เจ้าหน้าที่จะติดต่อกลับมาภายในห้านาทีหรือมาอย่างที่ก่เกิดเหตุ	
Brief Description	ผู้ใช้ต้องการแจ็งเคลม ถ้ามีคู่กรณีเจ้าหน้าที่จะมาอย่างที่ก่เกิดเหตุ แต่ถ้าไม่มีคู่กรณีเจ้าหน้าที่จะติดต่อกลับมาภายในห้านาที	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. เข้าสู่แอปพลิเคชัน 2. กดปุ่มเคลม 4. กรอกข้อมูลตามเงื่อนไขที่กำหนด 	<ol style="list-style-type: none"> 3. ระบบตรวจสอบข้อมูลผู้ใช้งาน <ol style="list-style-type: none"> 3.1) กรณีมีประกัน <ol style="list-style-type: none"> 3.1.1) แสดงหน้าให้กรอกข้อมูล 3.2) กรณีไม่มีประกัน <ol style="list-style-type: none"> 3.2.1) แสดงหน้าคุณยังไม่มีประกันภัย 5. ระบบตรวจสอบข้อมูล 6. ระบบส่งข้อมูลไปแจ้งเจ้าหน้าที่ 7. เจ้าหน้าที่ติดต่อกลับไปยังผู้ใช้ <ol style="list-style-type: none"> 7.1) กรณีมีคู่กรณี <ol style="list-style-type: none"> 7.1.1) เจ้าหน้าที่มาอย่างที่ก่เกิดเหตุ

		7.2) กรณีไม่มีคู่กรณี 7.2.1) เจ้าหน้าที่ ติดต่อ กลับมาภายในห้านาที
Exceptions	2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต 4. กรอกข้อมูลไม่ถูกต้อง	

3.2.2.9. ติดต่อบริษัท

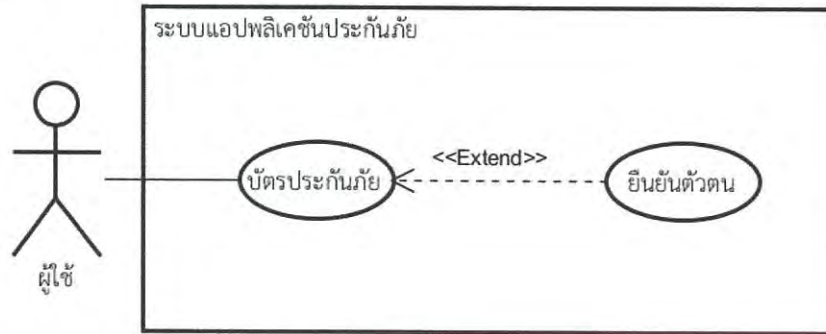


ภาพที่ 3.22 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันติดต่อ SAGI

ตารางที่ 3.9 ยูสเคสติดต่อบริษัท

Use case ID	09	
Use Case Name	ติดต่อบริษัท	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว เข้าหน้าค้นหาสถานที่และเลือกสถานที่ที่ต้องการ	
Post-Condition	ติดต่อเจ้าหน้าที่ผ่านโทรศัพท์	
Brief Description	ผู้ใช้ค้นหาสถานที่ที่ต้องการและได้เลือกติดต่อบริษัท	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> เข้าสู่แอปพลิเคชัน กดไปที่สนใจประกันนี้ กดติดต่อเจ้าหน้าที่ 	<ol style="list-style-type: none"> ค้นหาเบอร์และโทรออก
Exceptions	2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต	

3.2.2.10. บัตรประกันภัย



ภาพที่ 3.23 แผนภาพยูสเคสแสดงความสัมพันธ์ระหว่างผู้ใช้และฟังก์ชันการใช้บัตรประกันภัย

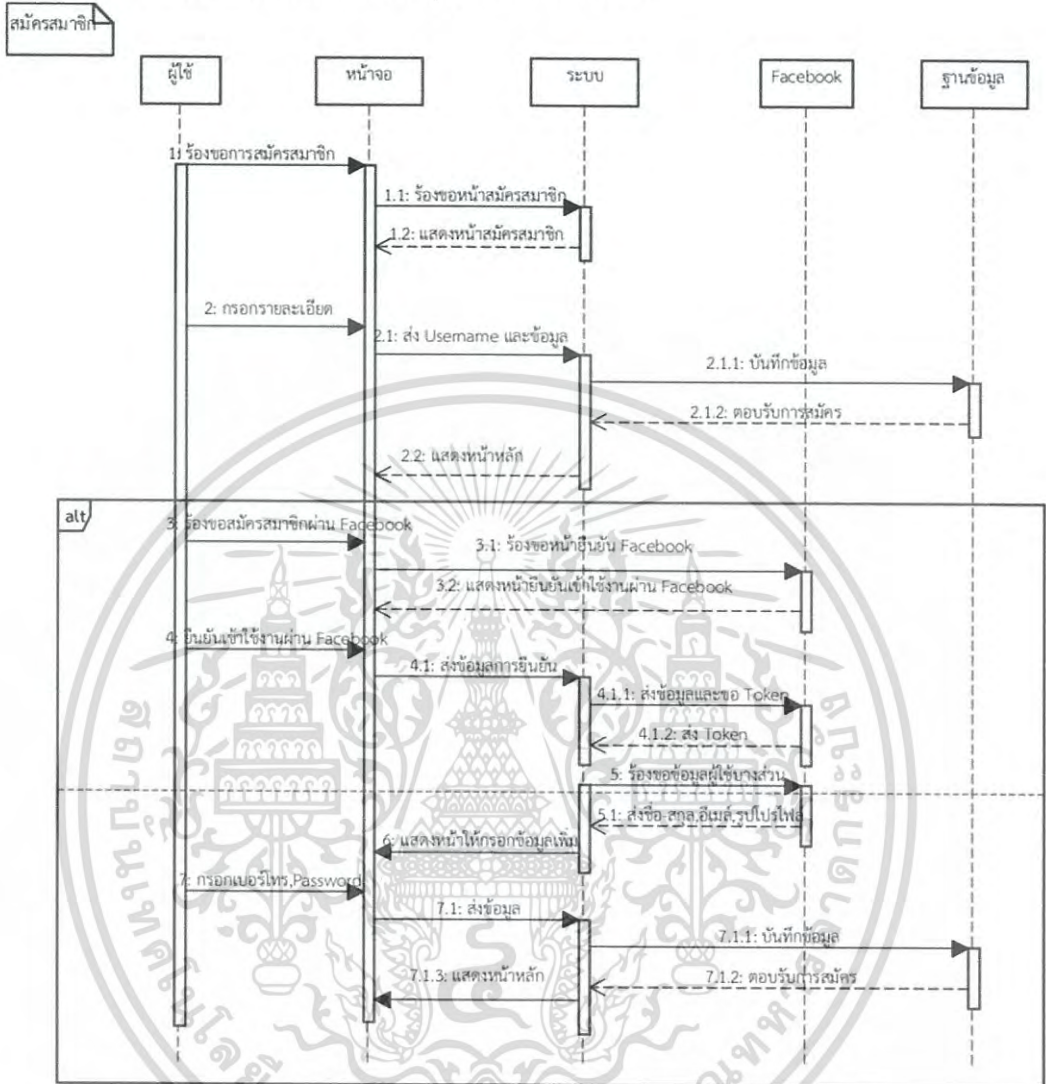
ตารางที่ 3.10 ยูสเคสบัตรประกันภัย

Use case ID	10	
Use Case Name	บัตรประกันภัย	
Actor	ผู้ใช้	
Pre-Condition	ต้องมีแอปพลิเคชันประกันภัย ล็อกอินเข้าสู่ระบบแล้ว มีประกันภัยของบริษัท ยืนยันการใช้งานบัตร	
Post-Condition	สามารถดูข้อมูลความคุ้มครอง ระยะเวลาคุ้มครอง	
Brief Description	ผู้ใช้ดูข้อมูลความคุ้มครอง ระยะเวลาคุ้มครอง	
Flow of Event	Actor	System
	<ol style="list-style-type: none"> 1. เข้าสู่แอปพลิเคชัน 2. กดปุ่มบัตรประกันภัย 4. กรอกเลขบัตรประชาชนเพื่อ ยืนยันการใช้งานบัตร 6. ผู้ใช้ใช้งานบัตรได้ 	<ol style="list-style-type: none"> 3. ระบบตรวจสอบข้อมูลผู้ใช้งาน <ol style="list-style-type: none"> 3.1) กรณีไม่มีประกัน <ol style="list-style-type: none"> 3.1.1) แสดงหน้าคุณยังไม่ มีประกันภัยของเรา 3.2) กรณีมีประกัน <ol style="list-style-type: none"> 3.2.1) แสดงหน้ายืนยัน ตัวตน 5. ระบบตรวจสอบข้อมูล
Exceptions	<ol style="list-style-type: none"> 2. ไม่ได้เชื่อมต่ออินเทอร์เน็ต 4. กรอกข้อมูลไม่ถูกต้อง แสดงหน้าไม่พบหมายเลขบัตรประชาชนในระบบ 	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

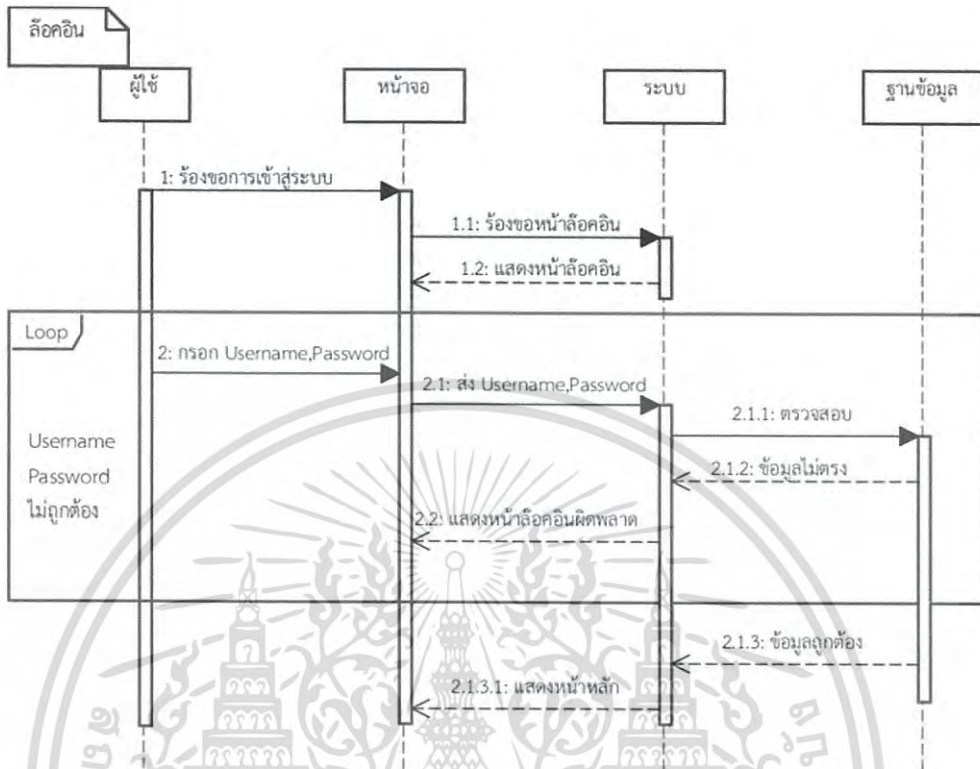
3.2.3. แผนภาพลำดับเหตุการณ์ (Sequence diagram)

3.2.3.1. แผนภาพลำดับเหตุการณ์สมัครสมาชิก



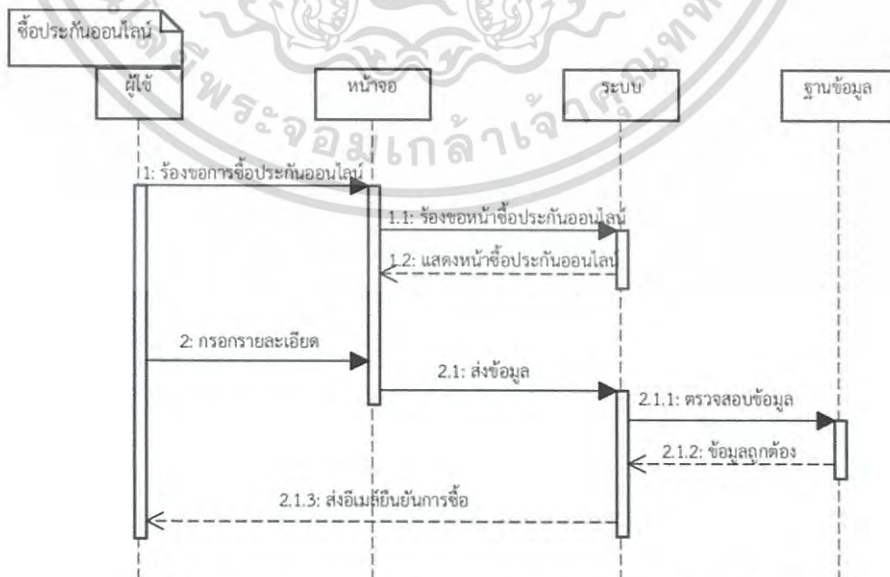
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2. แผนภาพลำดับเหตุการณ์ลือคอิน



ภาพที่ 3.24 แผนภาพลำดับเหตุการณ์การลือคอิน

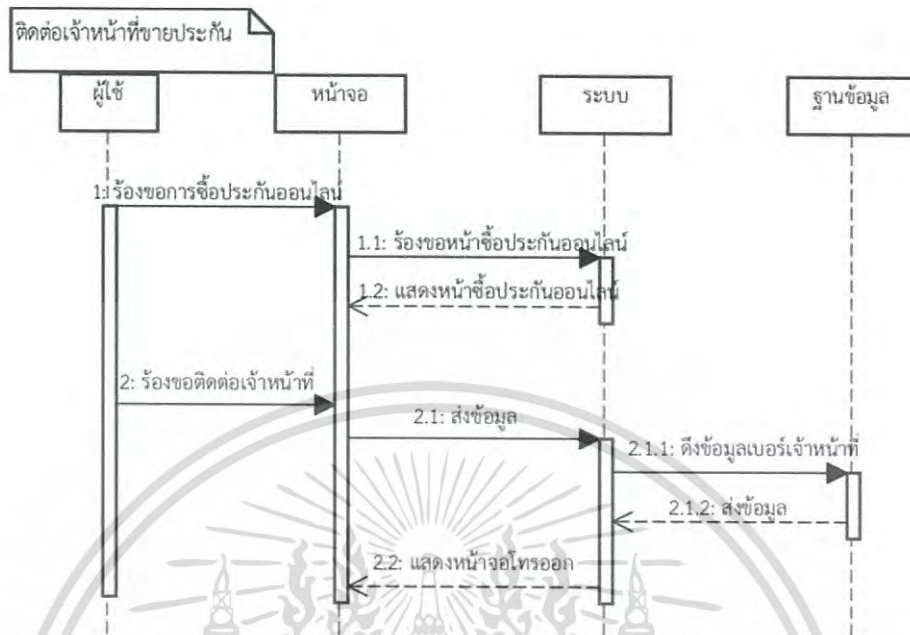
3.2.3.3. แผนภาพลำดับเหตุการณ์ซื้อประกันออนไลน์



ภาพที่ 3.25 แผนภาพลำดับเหตุการณ์การซื้อประกันออนไลน์

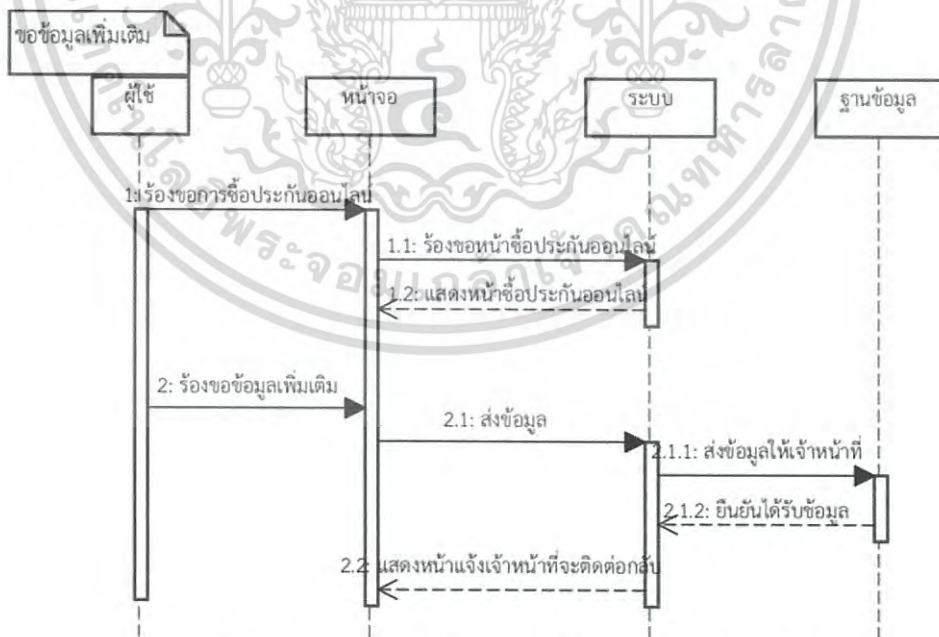
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.4. แผนภาพลำดับเหตุการณ์ติดต่อเจ้าหน้าที่ขายประกัน



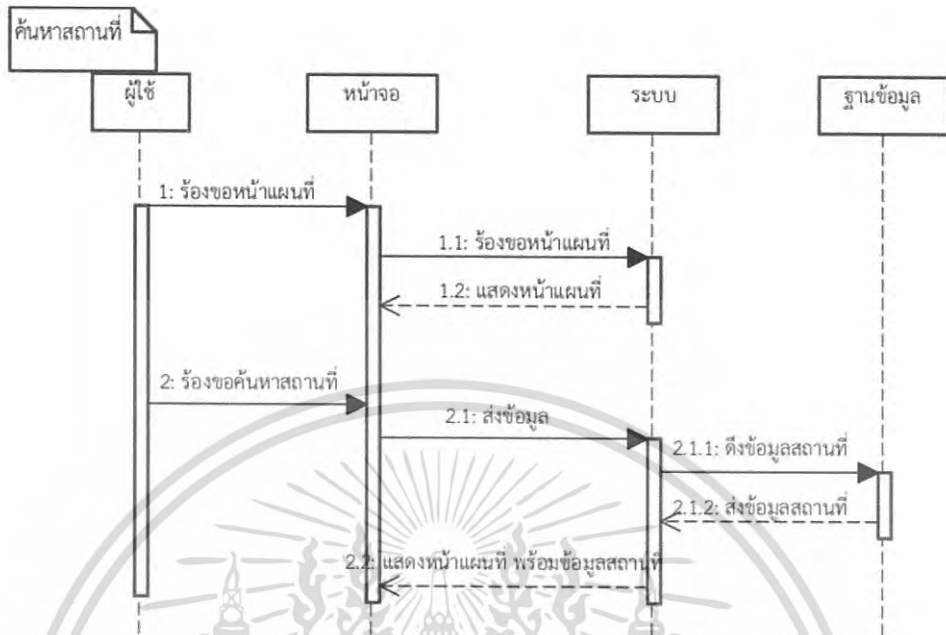
ภาพที่ 3.26 แผนภาพลำดับเหตุการณ์การติดต่อเจ้าหน้าที่

3.2.3.5. แผนภาพลำดับเหตุการณ์ขอข้อมูลเพิ่มเติม



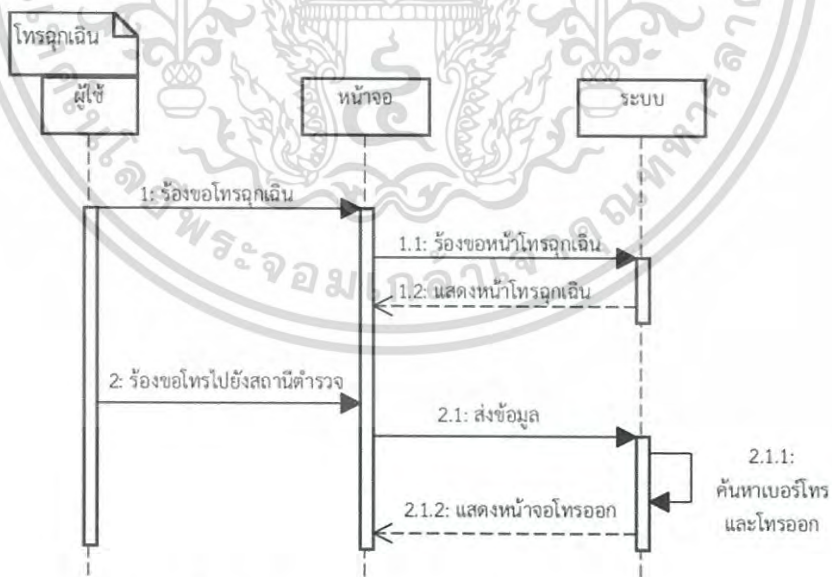
ภาพที่ 3.27 แผนภาพลำดับเหตุการณ์การขอข้อมูลเพิ่มเติม

3.2.3.6. แผนภาพลำดับเหตุการณ์ค้นหาสถานที่



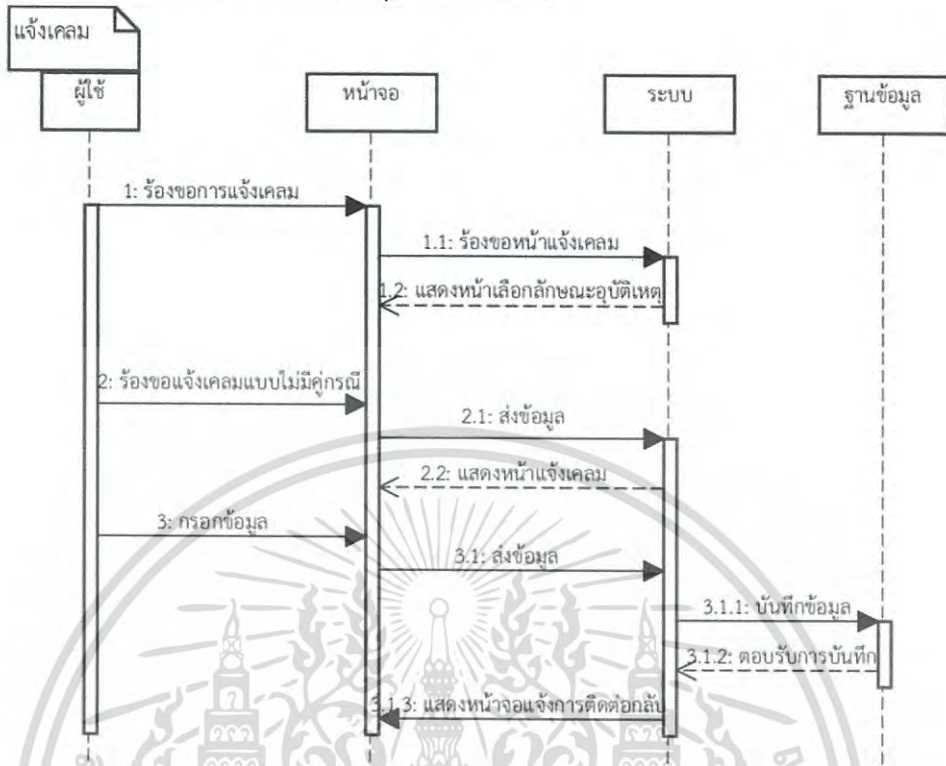
ภาพที่ 3.28 แผนภาพลำดับเหตุการณ์การค้นหาสถานที่

3.2.3.7. แผนภาพลำดับเหตุการณ์โทรฉุกเฉิน



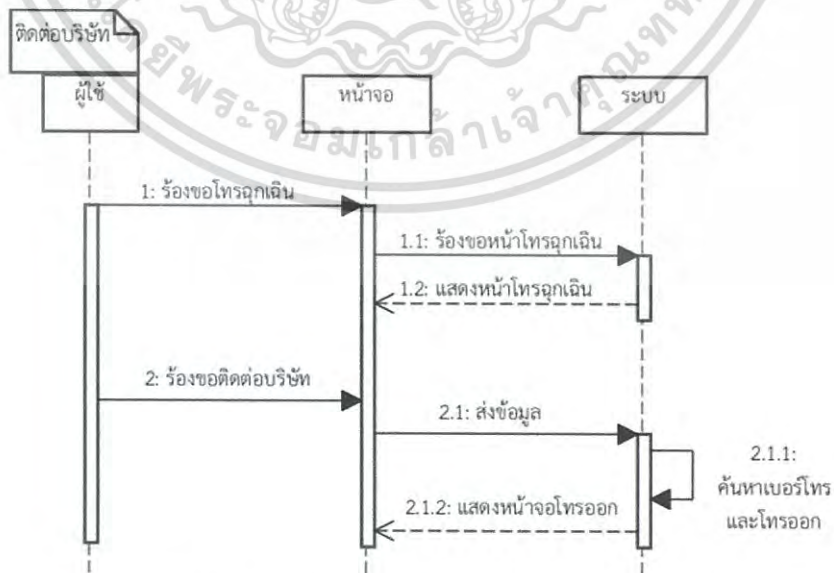
ภาพที่ 3.29 แผนภาพลำดับเหตุการณ์การโทรฉุกเฉิน

3.2.3.8. แผนภาพลำดับเหตุการณ์แจ้งเตือน



ภาพที่ 3.30 แผนภาพลำดับเหตุการณ์การแจ้งเตือน

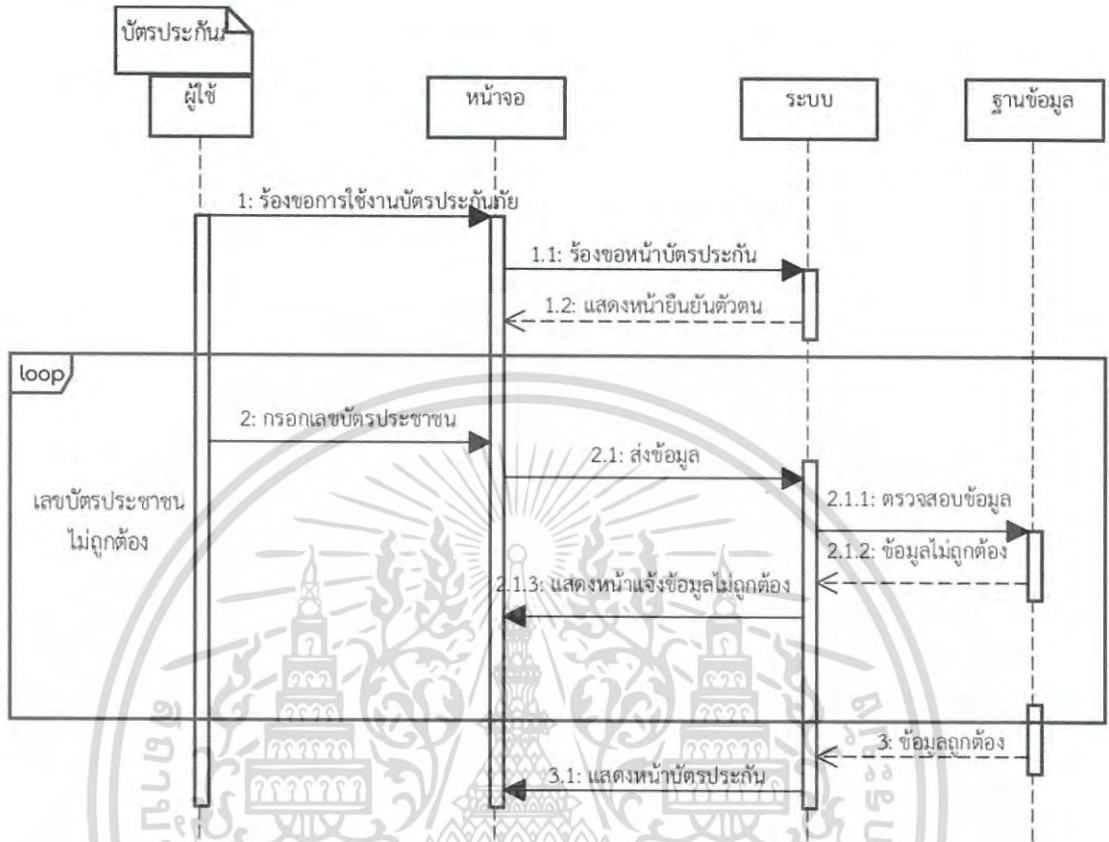
3.2.3.9. แผนภาพลำดับเหตุการณ์ติดต่อบริษัท



ภาพที่ 3.31 แผนภาพลำดับเหตุการณ์การใช้งานบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.10. แผนภาพลำดับเหตุการณ์บัตรประกันภัย



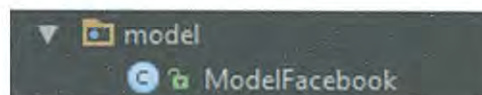
ภาพที่ 3.32 แผนภาพลำดับเหตุการณ์การใช้งานบัตร

3.2.4. การออกแบบแอปพลิเคชันด้วยโครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC

ในขั้นตอนการออกแบบแอปพลิเคชันได้นำโครงสร้างสถาปัตยกรรมซอฟต์แวร์แบบ MVC (Model-View-Controller) มาประยุกต์ใช้เพื่อให้ง่ายต่อการแก้ไขโปรแกรมซึ่งได้แบ่งไฟล์ต่างๆ ออกเป็น 3 กลุ่ม ดังนี้

1.) กลุ่มโมเดล ได้แก่

- Model ไว้เก็บคลาสจำพวก Business Logic หรือ การคำนวณ เช่น ModelFacebook ดังภาพที่ 3.33

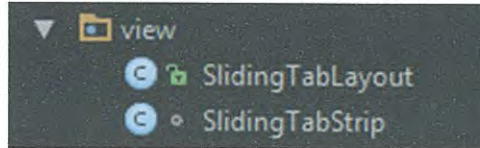


ภาพที่ 3.33 แพ็กเกจ model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) กลุ่มวิว ได้แก่

- View ไว้เก็บคลาสจำพวก Custom view group เช่น SlidingTabLayout ดังภาพที่ 3.34



ภาพที่ 3.34 แพ็กเกจ view

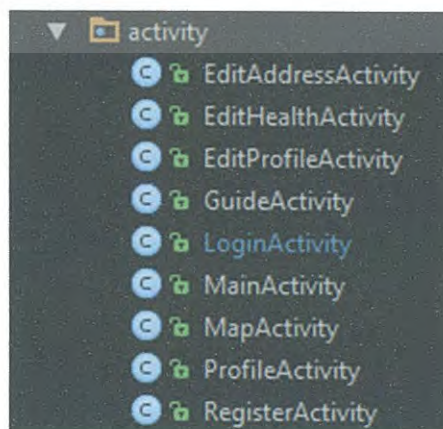
- Fragment ไว้เก็บคลาสจำพวก Fragment เช่น LoginFragment ดังภาพที่ 3.35



ภาพที่ 3.35 แพ็กเกจ fragment

3.) กลุ่มคอนโทรลเลอร์ ได้แก่

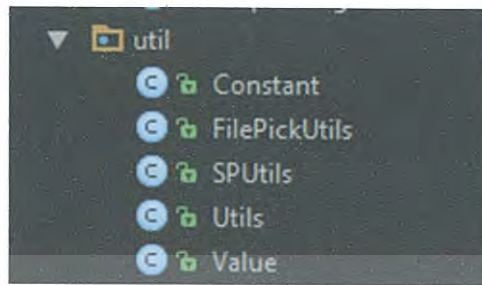
- Activity ไว้เก็บคลาสจำพวก Activity เช่น MainActivity LoginActivity ดังภาพที่ 3.36



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภาพที่ 3.36 แพ็กเกจ activity

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

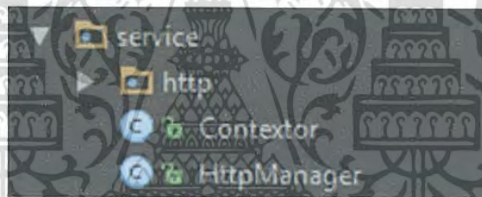
- Util ไว้เก็บคลาสจำพวก ดึงข้อมูลแบบง่ายๆ เช่น Value เก็บค่า String, Constant เก็บค่าคงที่, SPUtil บันทึกข้อมูลตัวแปรไว้ในแอปพลิเคชัน ดังภาพที่ 3.37



ภาพที่ 3.37 แพ็กเกจ util

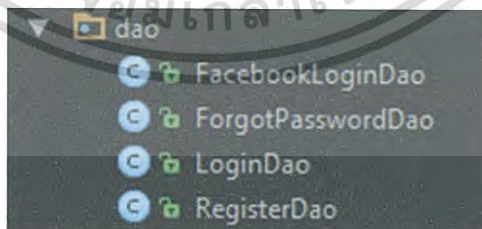
4.) ส่วนที่เพิ่มเติมจากรูปแบบเอ็มวีซี ได้แก่

- Service ไว้เก็บคลาสจำพวก ทำหน้าที่ในการประมวลผลเข้าถึงข้อมูลที่อยู่ในระบบฐานข้อมูล ดังภาพที่ 3.38



ภาพที่ 3.38 แพ็กเกจ service

- DAO เป็น service อันหนึ่ง ไว้เก็บคลาส Data Access Logic ที่ช่วยให้การเข้าถึงฐานข้อมูลได้ง่ายขึ้น ดังภาพที่ 3.39

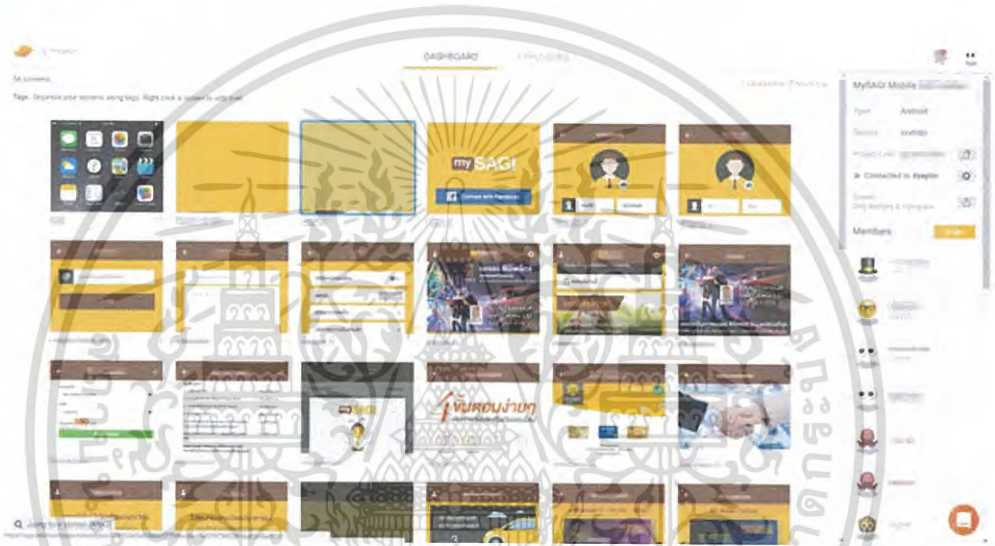


ภาพที่ 3.39 แพ็กเกจ DAO

3.3. ออกแบบและพัฒนาระบบหลังบ้านและส่วนต่อประสานโปรแกรมประยุกต์

3.3.1. ศึกษาการทำงานของแอปพลิเคชันจากไวต์เฟรม

จากขั้นตอนการดำเนินงานในส่วนของการออกแบบ หลังได้ข้อสรุปจากการออกแบบไวต์เฟรมและนำเสนอลูกค้าคือ บริษัท ศรีอยุธยา เจนเนอรัล ประกันภัย จำกัด (มหาชน) จึงนำรูปแบบที่ทีมออกแบบได้ออกแบบไว้ มาวิเคราะห์เพื่อสรุปโครงสร้างของโปรเจกต์และระบบที่ควรจะมี โดยติดต่อสื่อสารกับทีมออกแบบด้วย โปรแกรม Zeplin เพื่อดูรายละเอียดการออกแบบรวมถึงการทำงานที่ได้ออกแบบไว้ดังภาพที่ 3.40 จากนั้นวิเคราะห์การทำงานร่วมกับทีมพัฒนาแอปพลิเคชันเพื่อกำหนดการทำงานและกำหนดส่วนต่อประสานโปรแกรมประยุกต์ (Application Programming Interface : API) ที่ละหน้า



ภาพที่ 3.40 หน้าตาโปรแกรม Zeplin แสดงการออกแบบไวต์เฟรมของแอปพลิเคชัน

3.3.1.1. หน้าล็อกอิน

- มีการทำงานร่วมกับ API ในส่วนของการเข้าสู่ระบบ
- มีการทำงานร่วมกับ API ในส่วนของการเชื่อมต่อกับเฟซบุ๊ก



ภาพที่ 3.41 หน้าจอแอปพลิเคชันขณะล็อกอิน

3.3.1.2. หน้าสมัครสมาชิก

- มีการทำงานร่วมกับ API ในส่วนของการสมัครสมาชิก



ภาพที่ 3.42 หน้าจอแอปพลิเคชันขณะล็อกอิน

3.3.1.3. หน้าสื่مرتผ่าน

- มีการทำงานร่วมกับ API ในส่วนของการตั้งค่ารหัสใหม่และส่งรหัสไปยังอีเมลหรือเบอร์โทรศัพท์ลูกค้า



ภาพที่ 3.43 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าสื่مرتผ่าน

3.3.1.4. หน้าหลัก

- มีการทำงานร่วมกับ API ในส่วนของการโหลดโปรโมชั่นและผลิตภัณฑ์ในหน้าหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาพที่ 3.44 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าหลักไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.5. หน้าแสดงรายละเอียดโปรโมชั่น

- มีการทำงานร่วมกับ API ในส่วนของการดึงรายละเอียดของโปรโมชั่น ทั้งรูปภาพ หัวข้อ และคำอธิบาย



ภาพที่ 3.45 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแสดงรายละเอียดโปรโมชั่น

3.3.1.6. หน้าซื้อประกันออนไลน์

- มีการทำงานร่วมกับ API ในส่วนของการดึงลิงก์บนเว็บไซต์ขายประกันของบริษัท ประกันเพื่อแสดงผลหน้าเว็บสำหรับซื้อประกันออนไลน์ภายในแอปพลิเคชัน



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ภาพที่ 3.46 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าซื้อประกันออนไลน์ โยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.7. หน้าบัตรประกันของฉัน

- มีการทำงานร่วมกับ API ในส่วนของการดึงข้อมูลกรมธรรม์ของลูกค้าที่มีอยู่มาแสดง



ภาพที่ 3.47 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าบัตรประกันของฉัน

3.3.1.8. หน้าบัตรประกันจำลอง

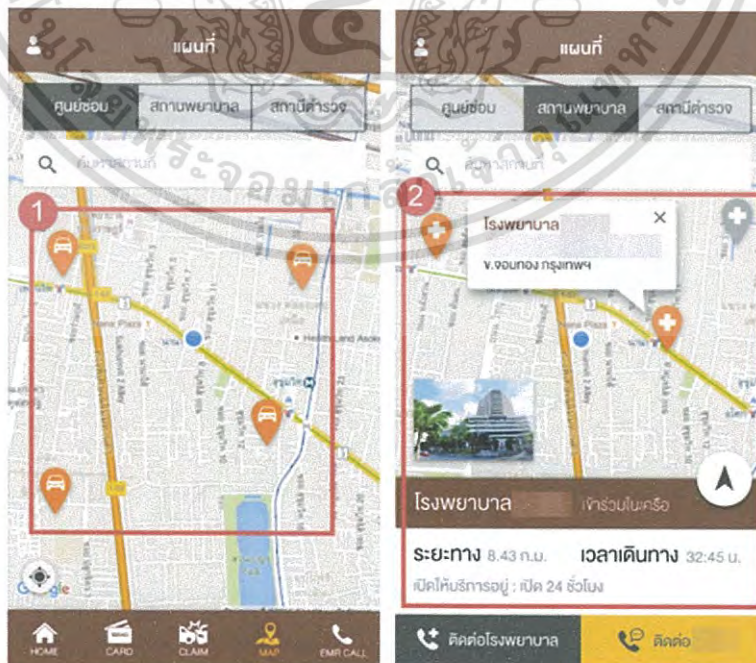
- มีการทำงานร่วมกับ API ในส่วนของการดึงข้อมูลกรมธรรม์แต่ละประเภทของลูกค้าที่มีอยู่ทั้งหมายเลขกรมธรรม์ เบี้ยประกัน ระยะเวลาคุ้มครอง และอื่นๆ มาแสดง



ภาพที่ 3.48 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าบัตรประกันจำลอง

3.3.1.9. หน้าแผนที่

- มีการทำงานร่วมกับ API ในส่วนของการดึงข้อมูลศูนย์ซ่อมในเครือและนอกเครือบริษัทประกันมาแสดง
- มีการทำงานร่วมกับ API ในส่วนของการดึงข้อมูลโรงพยาบาลในเครือและนอกเครือบริษัทประกันมาแสดง

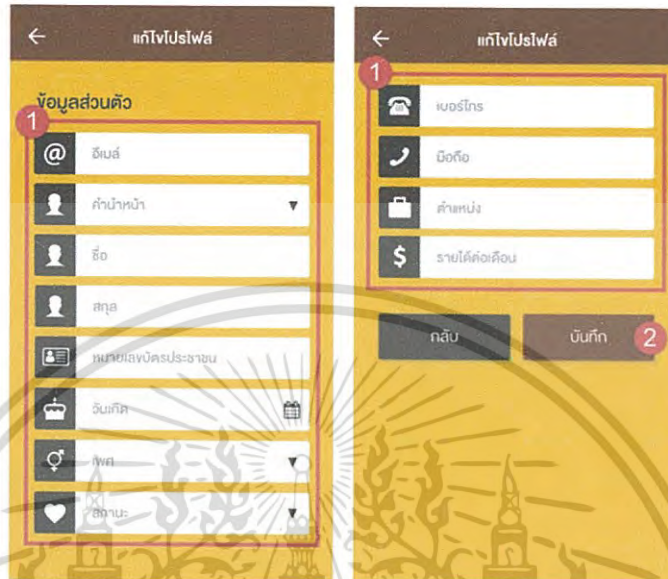


ภาพที่ 3.49 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแผนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.10. หน้าแก้ไขข้อมูลส่วนตัว

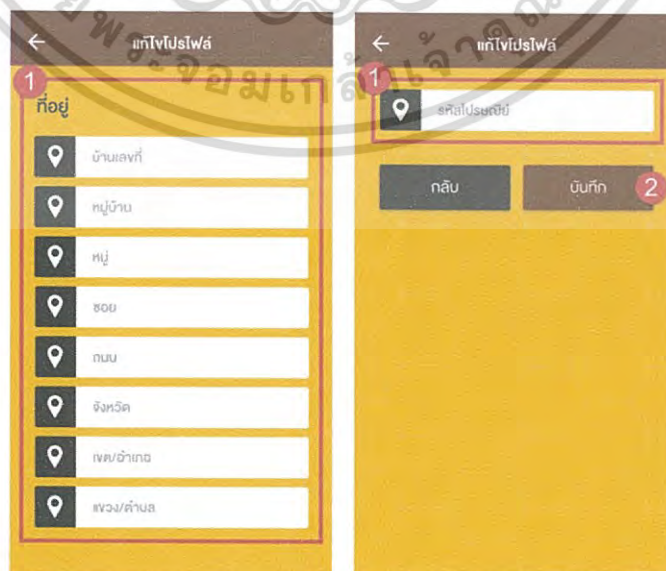
- มีการทำงานร่วมกับ API ในส่วนของการดึงข้อมูลส่วนตัวของลูกค้ามาแสดง
- มีการทำงานร่วมกับ API ในส่วนของการบันทึกข้อมูลส่วนตัวของลูกค้าลงฐานข้อมูล



ภาพที่ 3.50 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแก้ไขข้อมูลส่วนตัว

3.3.1.11. หน้าแก้ไขข้อมูลที่อยู่อาศัย

- มีการทำงานร่วมกับ API ในส่วนของการดึงข้อมูลส่วนตัวของลูกค้ามาแสดง
- มีการทำงานร่วมกับ API ในส่วนของการบันทึกข้อมูลส่วนตัวของลูกค้าลงฐานข้อมูล



ภาพที่ 3.51 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแก้ไขข้อมูลที่อยู่อาศัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.12. หน้าตั้งค่า

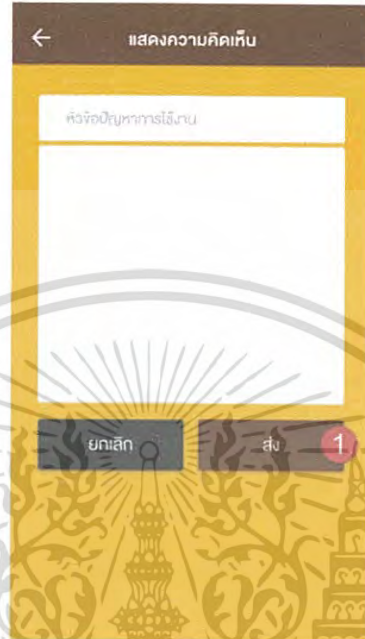
- มีการทำงานร่วมกับส่วนต่อประสานโปรแกรมประยุกต์ในส่วนของการดึงข้อมูล การตั้งค่า
- มีการทำงานร่วมกับส่วนต่อประสานโปรแกรมประยุกต์ในส่วนของการบันทึกการตั้งค่า
- มีการทำงานร่วมกับส่วนต่อประสานโปรแกรมประยุกต์ในส่วนของการออกจาก ระบบ



ภาพที่ 3.52 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าตั้งค่า

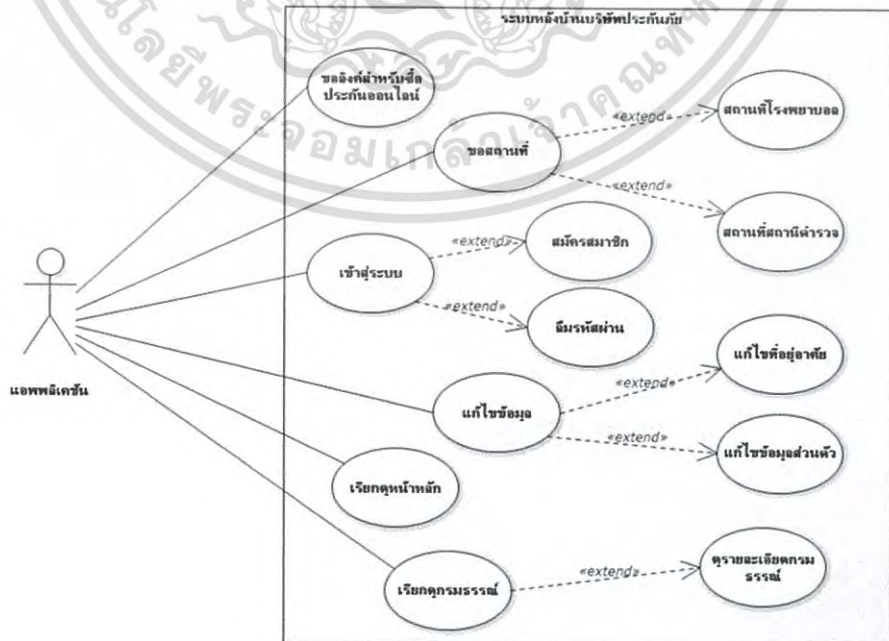
3.3.1.13. หน้าแสดงความคิดเห็น

- มีการทำงานร่วมกับส่วนต่อประสานโปรแกรมประยุกต์ในส่วนการบันทึกความคิดเห็น



ภาพที่ 3.53 หน้าจอแอปพลิเคชันขณะอยู่ในหน้าแสดงความคิดเห็น

3.3.2. แผนภาพยูสเคส (Use case diagram)



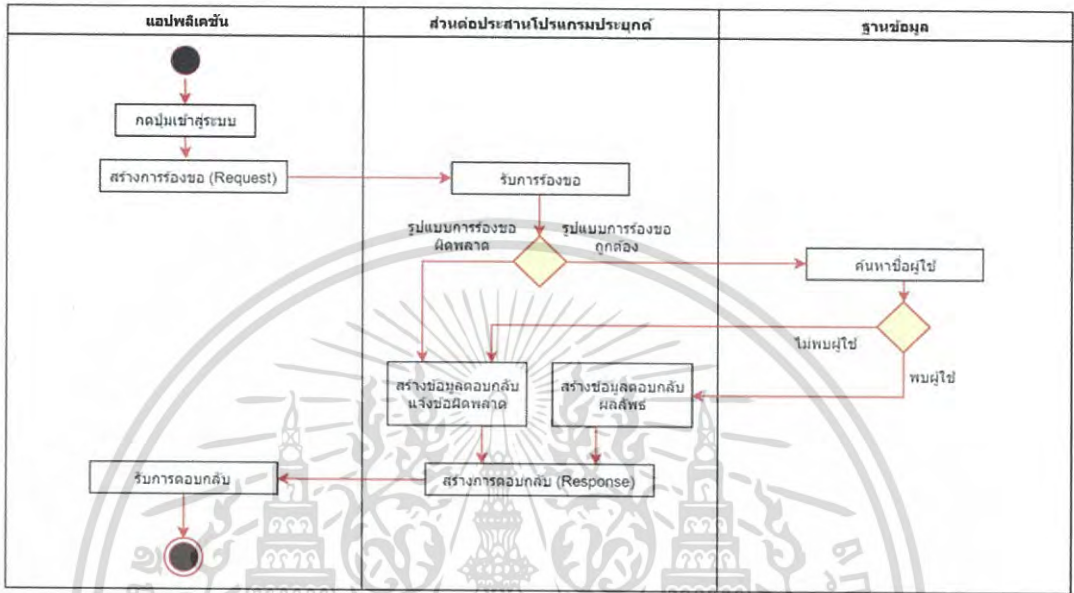
ภาพที่ 3.54 Use case diagram ของระบบส่วนต่อประสานโปรแกรมประยุกต์ด้านการค้า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทประกันภัยเพื่อใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาตจากบริษัทฯ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3. แผนภาพกิจกรรม (Activity diagram)

3.3.3.1. หน้าล็อกอิน

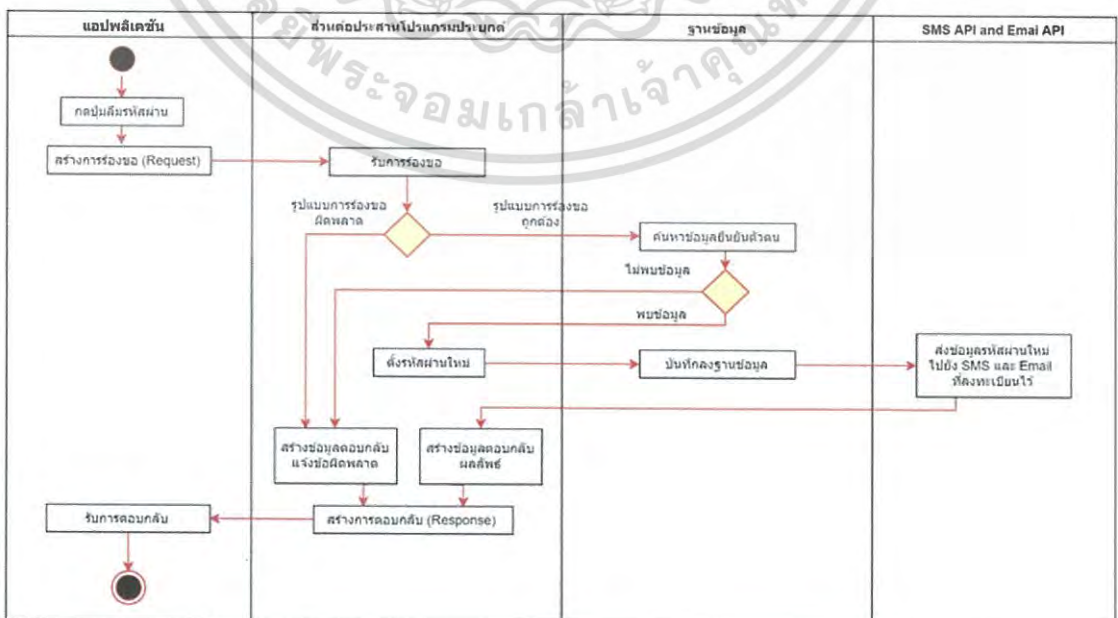
- ส่วนของการเข้าสู่ระบบ
- ส่วนของการเชื่อมต่อกับเฟซบุ๊ก



ภาพที่ 3.55 Activity diagram แสดงการทำงานของ API ในการลงชื่อเข้าใช้

3.3.3.2. หน้าสมัครรหัสผ่าน

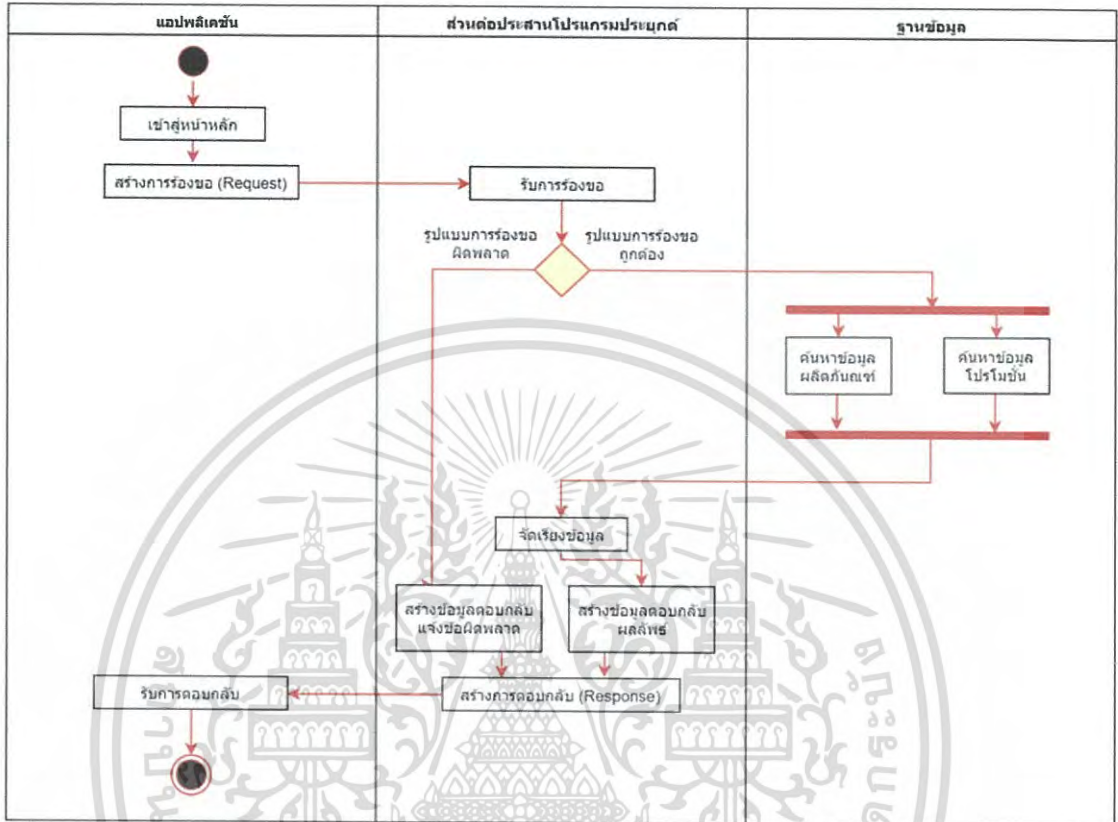
- ส่วนของการสมัครรหัสผ่าน



ภาพที่ 3.56 Activity diagram แสดงการทำงานของ API ในการกู้รหัสผ่าน

3.3.3.3. หน้าหลัก

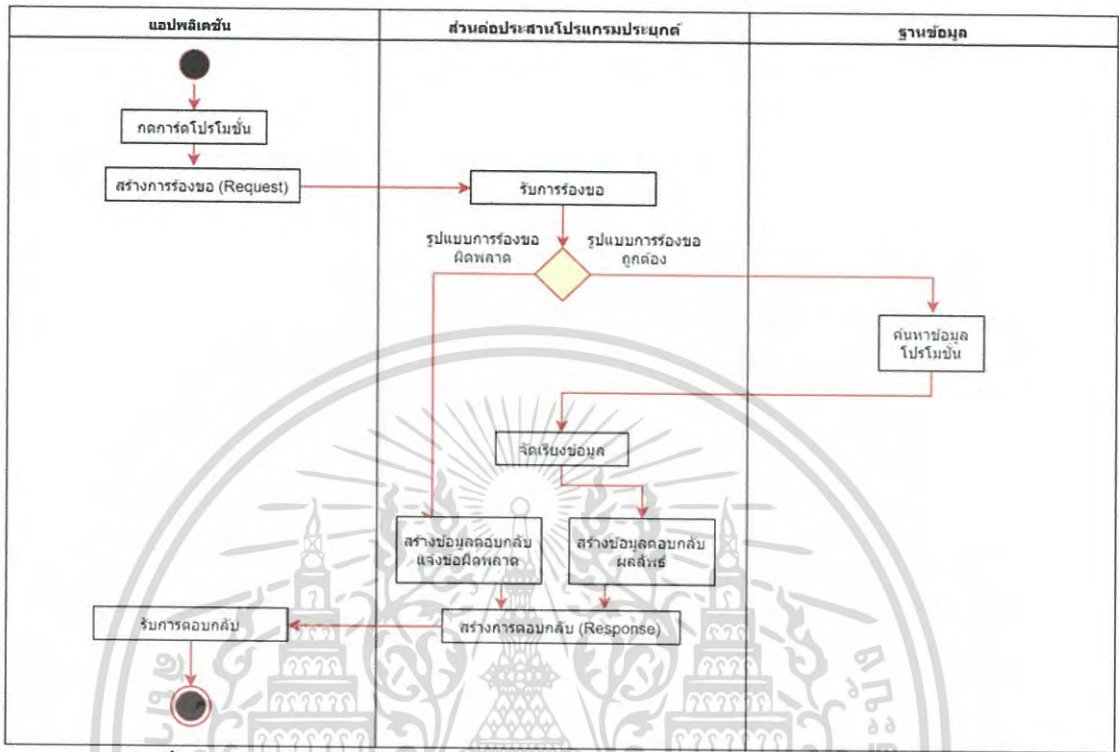
- ส่วนของการไหลข้อมูลโปรโมชันและผลิตภัณฑ์ เพื่อแสดงในหน้าหลัก



ภาพที่ 3.57 Activity diagram แสดงการทำงานของ API ในหน้าหลัก

3.3.3.4. หน้าแสดงรายละเอียดโปรโมชั่น

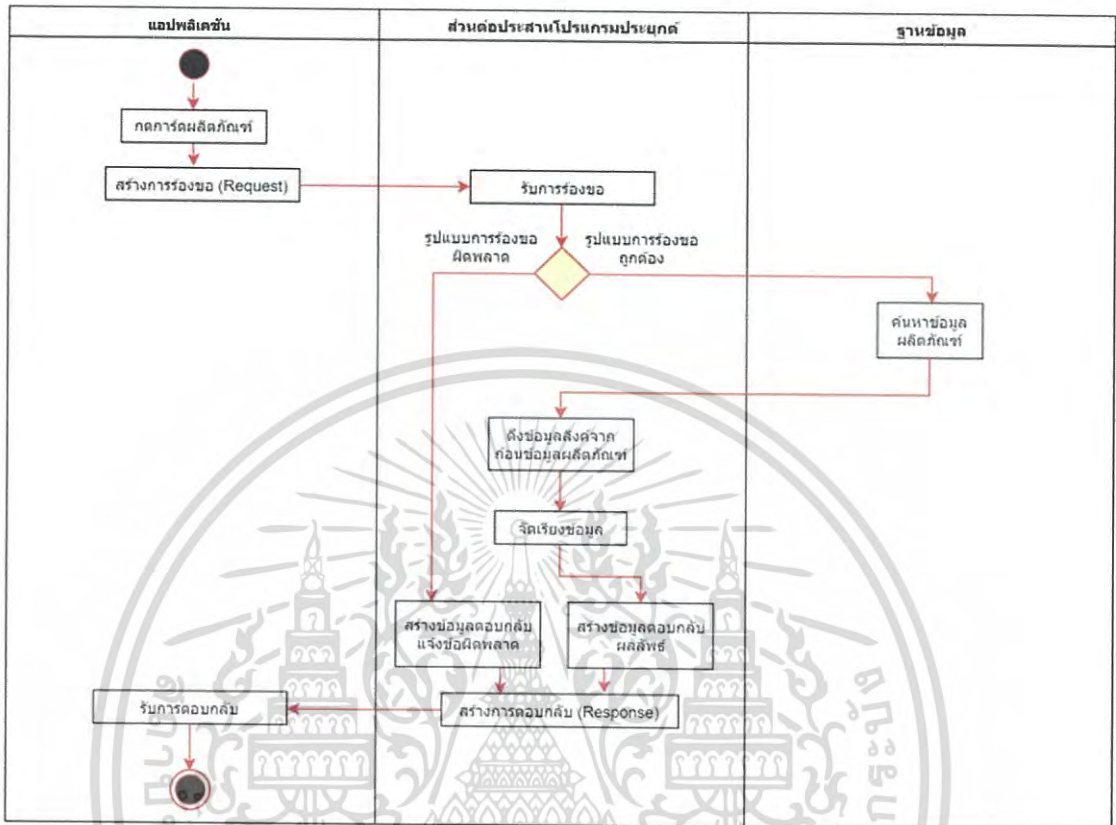
- ส่วนของการไหลทรายละเอียดโปรแกรม



ภาพที่ 3.58 Activity diagram แสดงการทำงานของ API ในการกดเลือกโปรโมชั่น

3.3.3.5. หน้าซื้อประกันออนไลน์

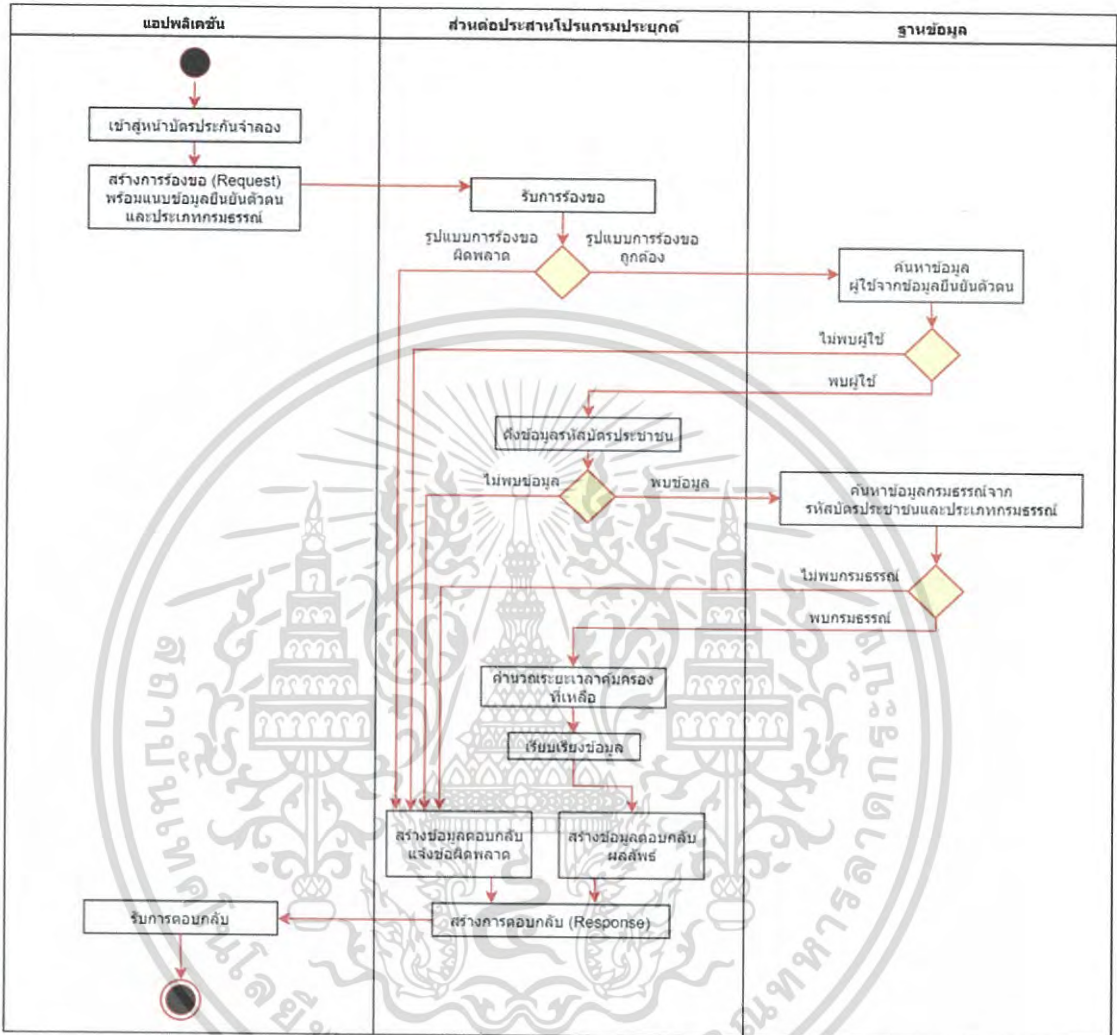
- ส่วนของการไหลข้อมูลผลิตภัณฑ์



ภาพที่ 3.59 Activity diagram แสดงการทำงานของ API ในการกดเลือกผลิตภัณฑ์

3.3.3.7. หน้าบัตรประกันจำลอง

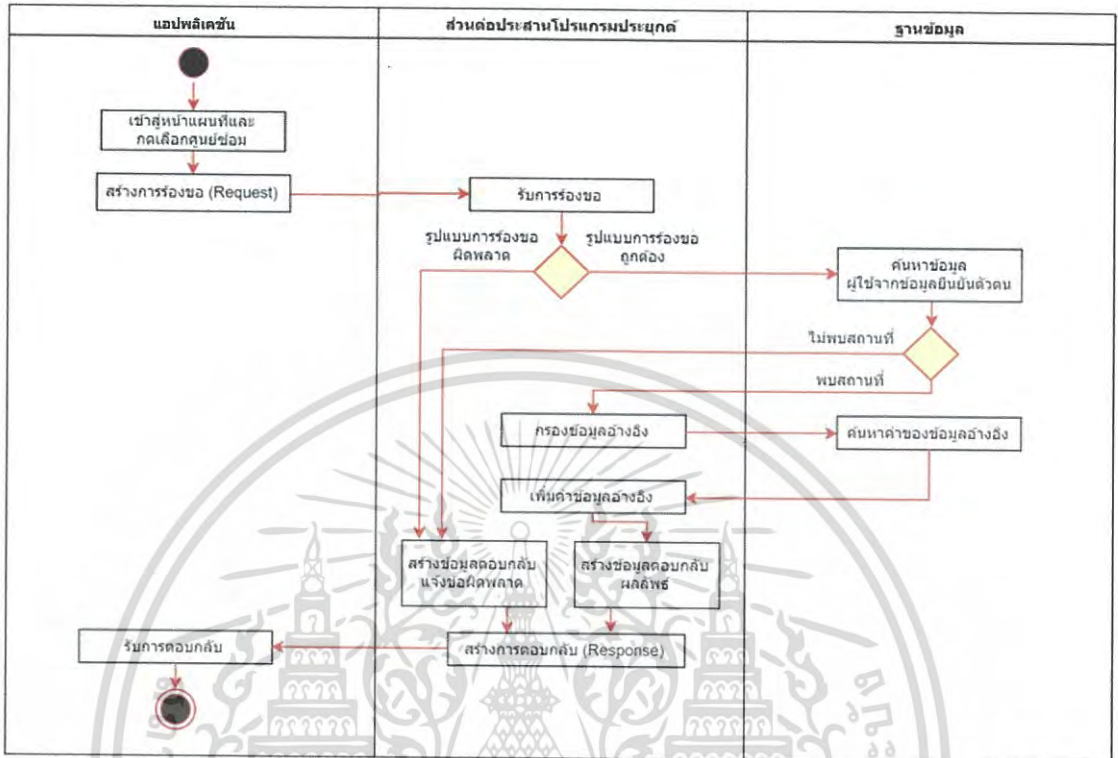
- ส่วนของการดึงข้อมูลกรมธรรม์แต่ละประเภทของลูกค้าที่มีอยู่ทั้งหมายเลขกรมธรรม์ เบี้ยประกัน ระยะเวลาคุ้มครอง และข้อมูลอื่นๆ



ภาพที่ 3.61 Activity diagram แสดงการทำงานของ API ในหน้าบัตรประกันจำลอง

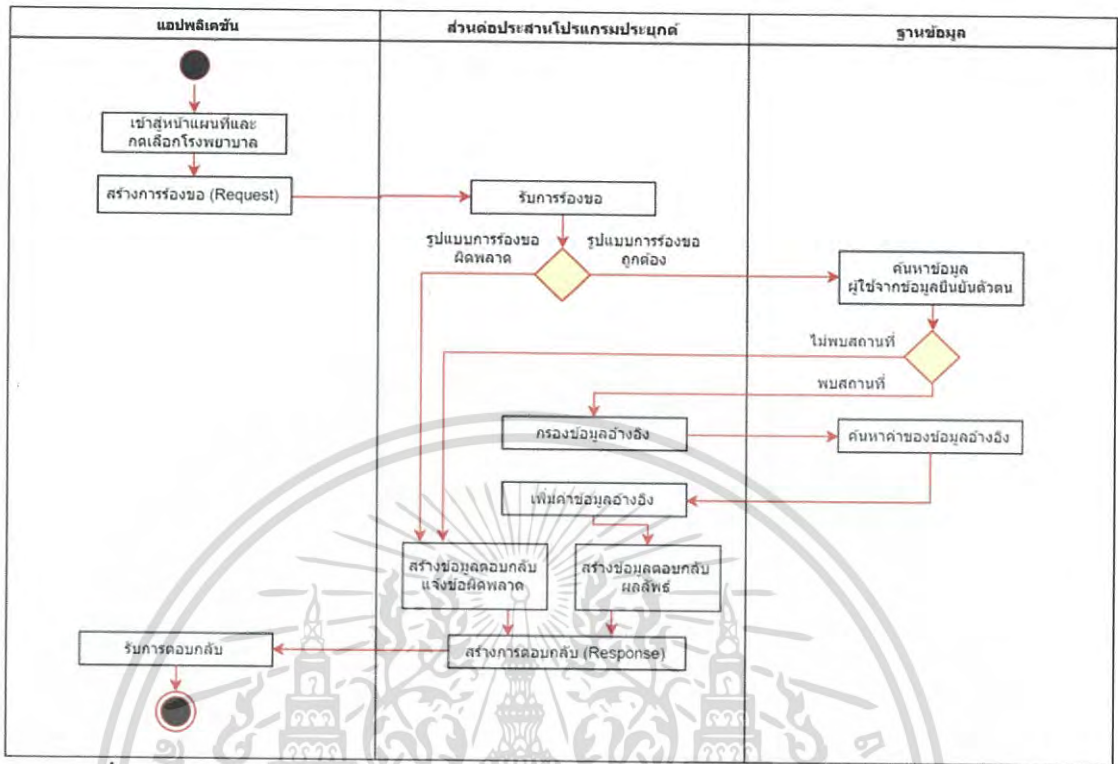
3.3.3.8. หน้าแผนที่

- ส่วนของการดึงข้อมูลศูนย์ซ่อมในเครื่องและนอกเครื่องบริษัทประกันมาแสดง



ภาพที่ 3.62 Activity diagram แสดงการทำงานของ API ในหน้าแผนที่ขณะเลือกสถานที่เป็นศูนย์ซ่อม

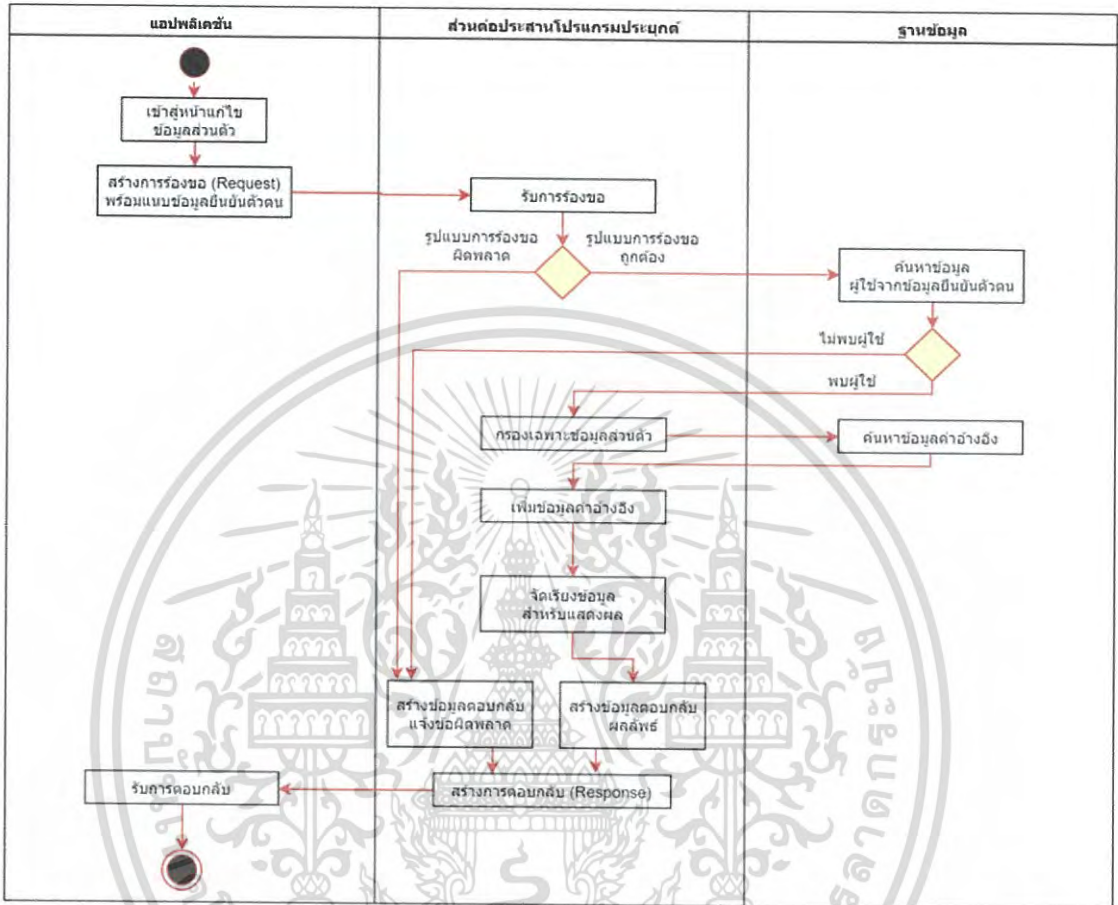
- ส่วนของการดึงข้อมูลโรงพยาบาลในเครือและนอกเครือบริษัทประกันมาแสดง



ภาพที่ 3.63 Activity diagram แสดงการทำงานของ API ในหน้าแผนที่ขณะเลือกสถานที่ที่เป็นโรงพยาบาล

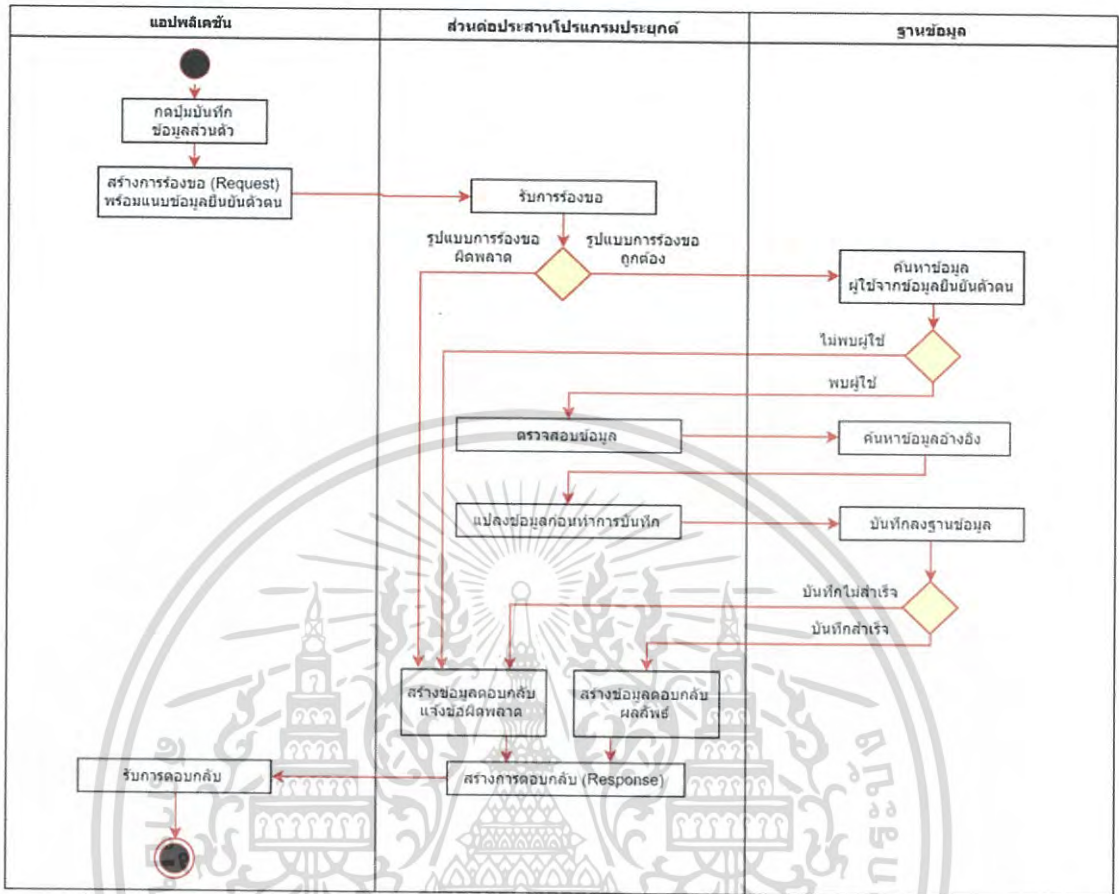
3.3.3.9. หน้าแก้ไขข้อมูลส่วนตัว

- ส่วนของการดึงข้อมูลส่วนตัวของลูกค้ามาแสดง



ภาพที่ 3.64 Activity diagram แสดงการทำงานของ API ในหน้าขณะอยู่ในหน้าแก้ไขข้อมูลส่วนตัว

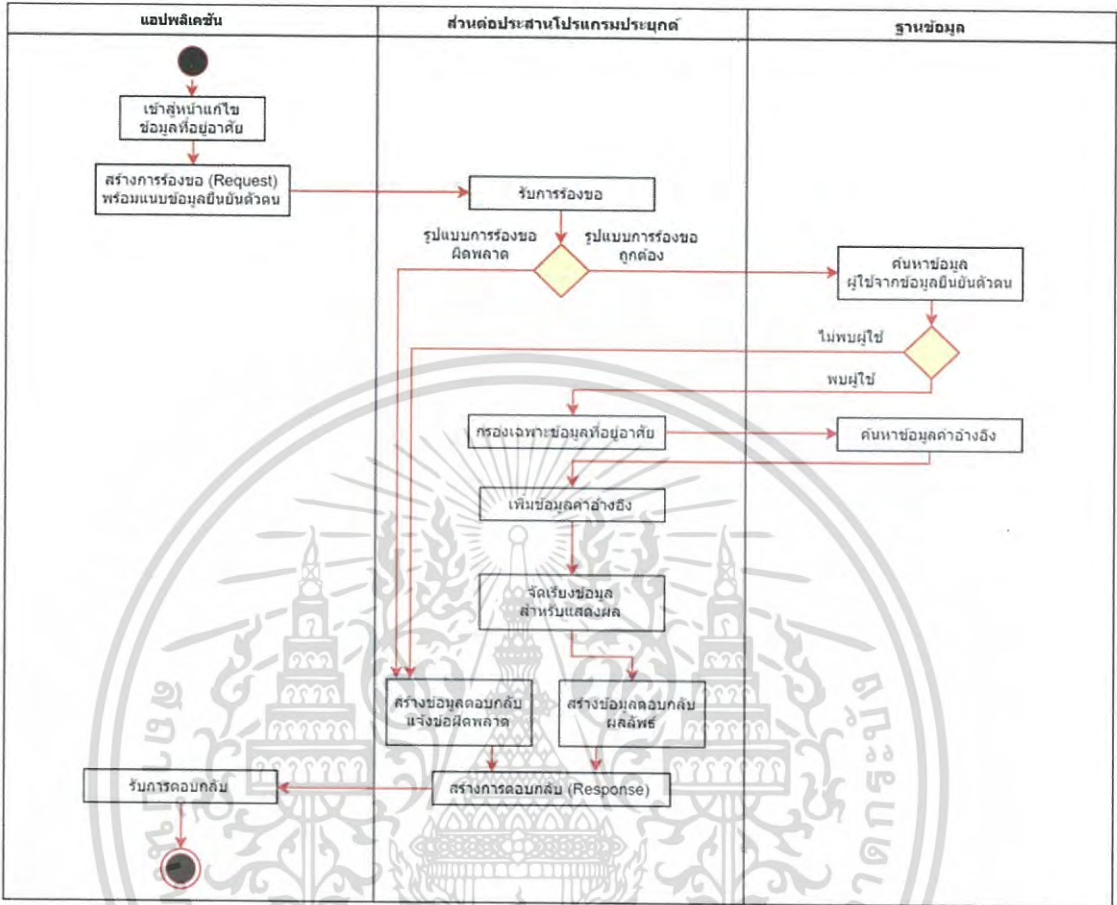
- ส่วนของการบันทึกข้อมูลส่วนตัวของลูกค้าฐานข้อมูล



ภาพที่ 3.65 Activity diagram แสดงการทำงานของ API ในหน้าขณะกดบันทึกในหน้าแก้ไขข้อมูลส่วนตัว

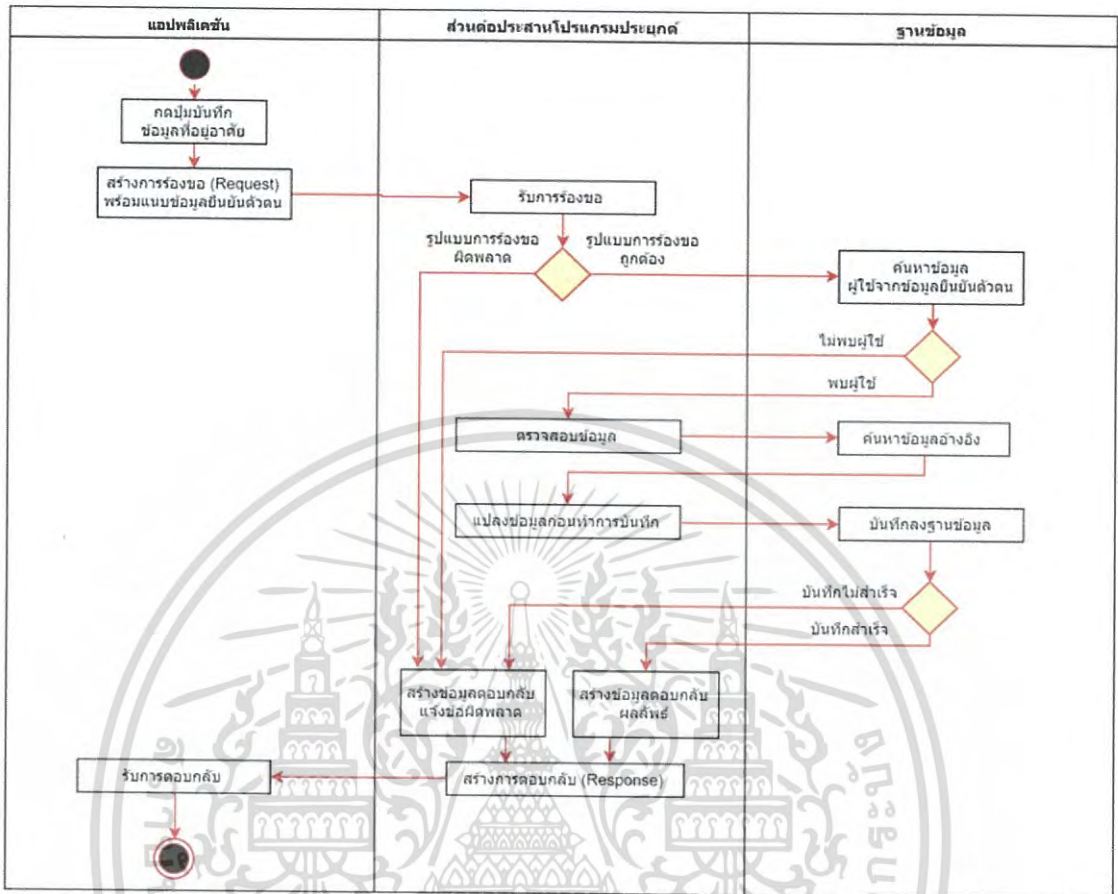
3.3.3.10. หน้าแก้ไขข้อมูลที่อยู่อาศัย

- ส่วนของการดึงข้อมูลที่อยู่อาศัยของลูกค้ามาแสดง



ภาพที่ 3.66 Activity diagram แสดงการทำงานของ API ในหน้าขณะอยู่ในหน้าแก้ไขข้อมูลที่อยู่อาศัย

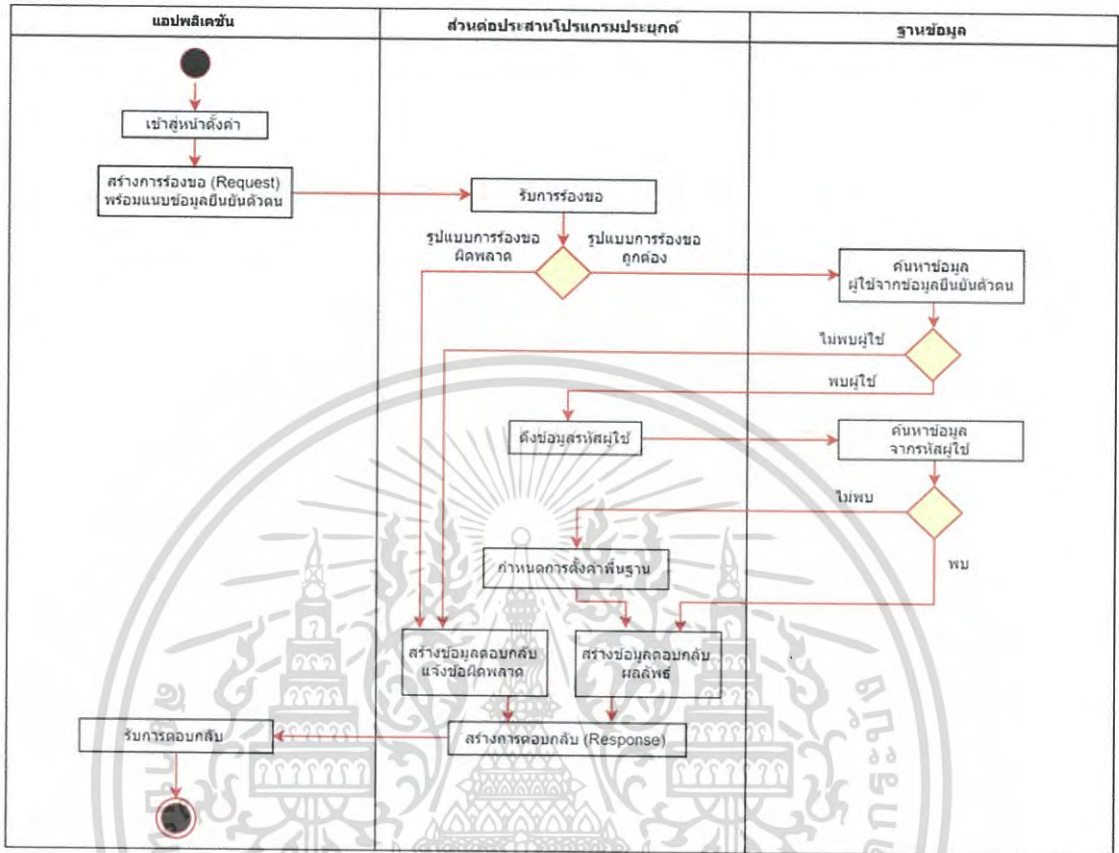
- ส่วนของการบันทึกข้อมูลที่อยู่อาศัยของลูกค้ำลงฐานข้อมูล



ภาพที่ 3.67 Activity diagram แสดงการทำงานของ API ในขณะกดบันทึกในหน้าแก้ไขข้อมูลที่อยู่อาศัย

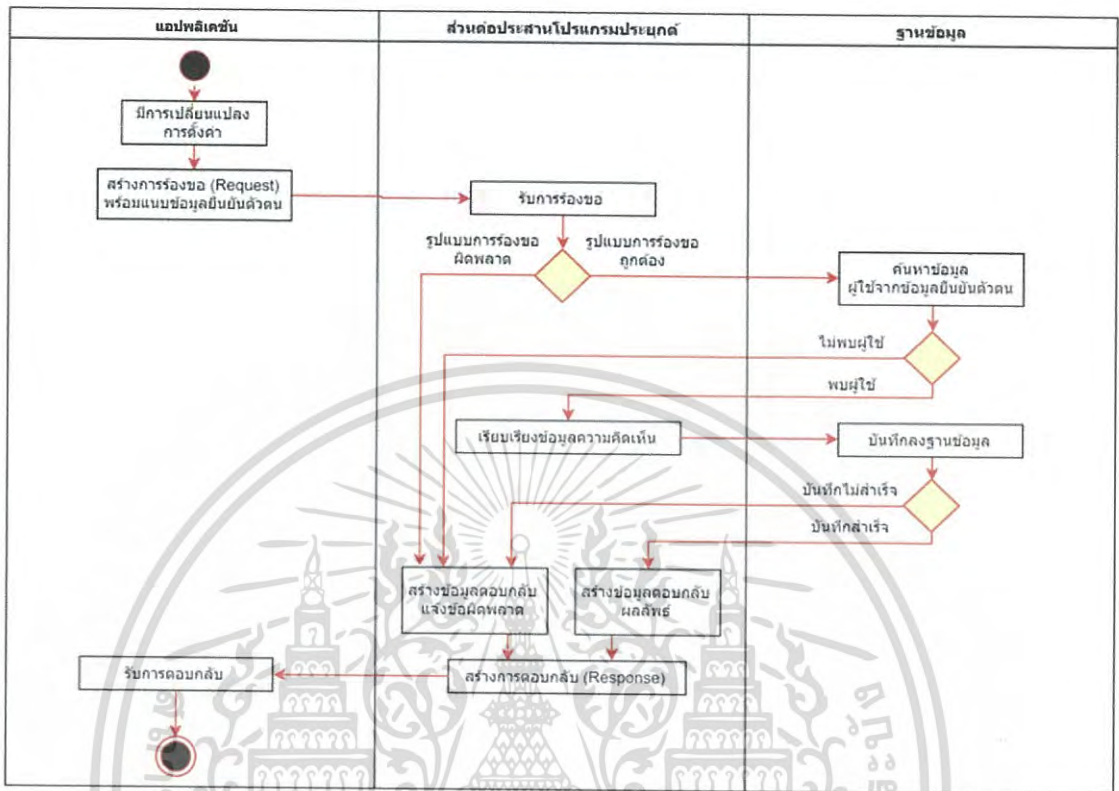
3.3.3.11. หน้าตั้งค่า

- ส่วนของการดึงข้อมูลการตั้งค่า



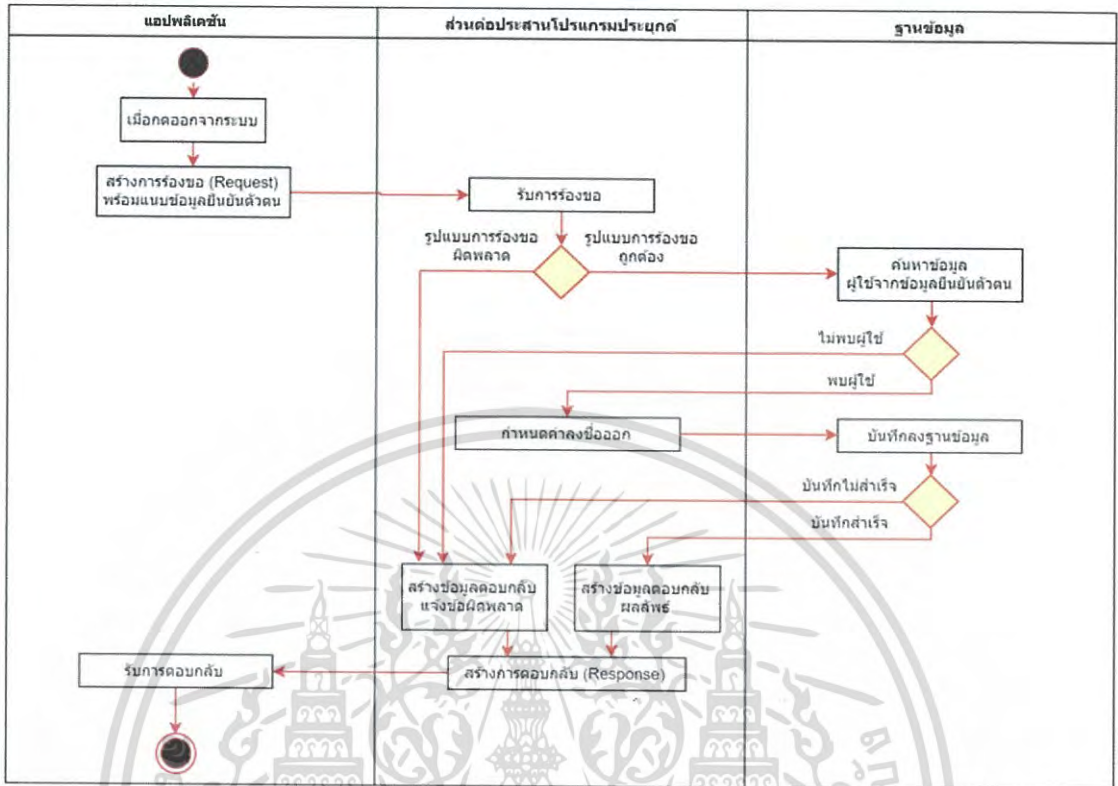
ภาพที่ 3.68 Activity diagram แสดงการทำงานของ API ในขณะที่เข้าสู่หน้าตั้งค่า

- ส่วนของการบันทึกการตั้งค่า



ภาพที่ 3.69 Activity diagram แสดงการทำงานของ API ในขณะมีการเปลี่ยนแปลงการตั้งค่า

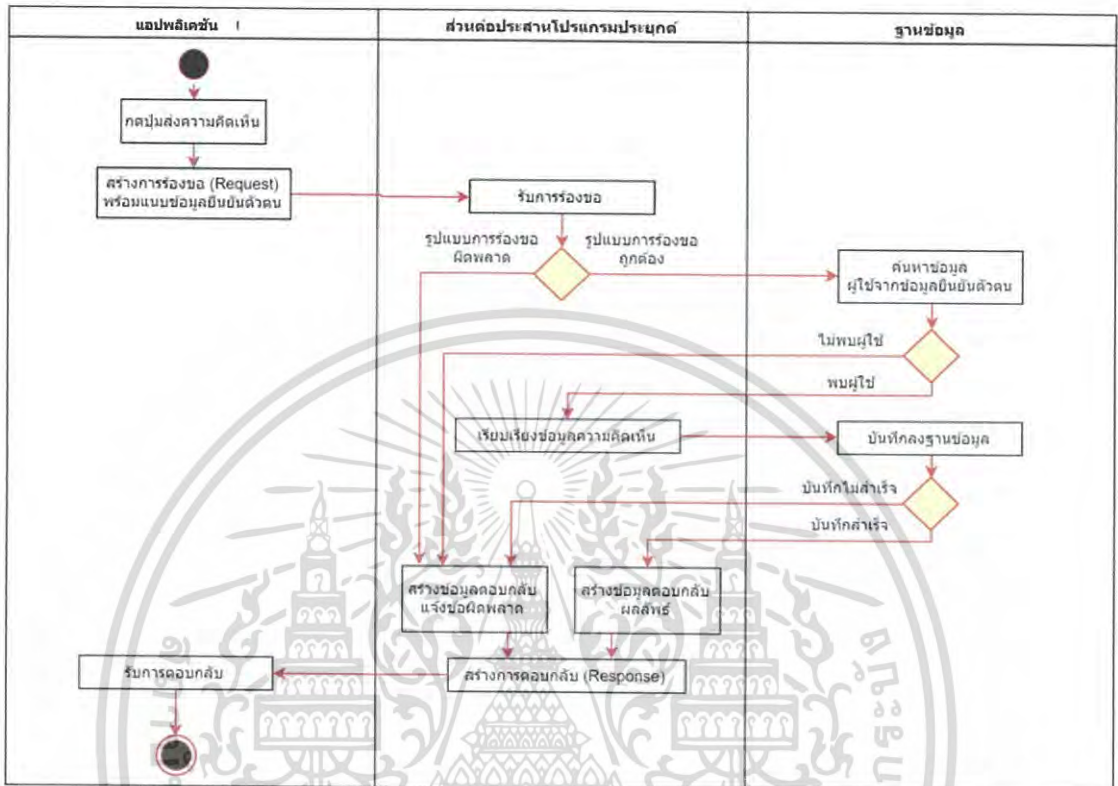
- ส่วนของการออกจากระบบ



ภาพที่ 3.70 Activity diagram แสดงการทำงานของ API ในขณะคดออกจากระบบ

3.3.3.12. หน้าแสดงความคิดเห็น

- ส่วนการบันทึกความคิดเห็น



ภาพที่ 3.71 Activity diagram แสดงการทำงานของ API ในขณะที่กดส่งความคิดเห็น

3.3.4. เตรียมความพร้อมสภาพแวดล้อมให้เหมาะสมต่อการพัฒนา

ก่อนเริ่มการพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ ต้องมีการเตรียมความพร้อมสภาพแวดล้อม (Environment) ให้เหมาะสมกับการพัฒนาเสียก่อน เพราะการพัฒนาาระบบใดๆ ก็ตาม ต้องมีการใช้งานเครื่องมือหลากหลายอย่างในการพัฒนา ไม่ว่าจะเป็นอุปกรณ์ฮาร์ดแวร์ หรือซอฟต์แวร์ต่างๆ ต้องมีการเตรียมความพร้อมในการพัฒนาให้มีสภาพแวดล้อมภายในระบบที่เหมาะสมต่อการพัฒนา เพื่อเป็นการหลีกเลี่ยงปัญหาที่อาจจะเกิดขึ้นในขั้นตอนการพัฒนา ซึ่งอาจจะทำให้เกิดความล่าช้า และปัญหาอื่นๆ ตามมาในอนาคต

เมื่อระบบมีสภาพแวดล้อมที่เหมาะสมต่อการพัฒนาแล้ว ก็จะสามารถเริ่มการพัฒนาได้โดยการพัฒนาในช่วงแรกจะเน้นไปที่การวางโครงสร้างของระบบหลังบ้านก่อน โดยการใช้ซอฟต์แวร์ต่างๆ เข้ามาช่วยในการสร้างระบบขึ้นมา (Build) รวมถึงทดสอบการทำงานเบื้องต้นเพื่อนำไปสู่การพัฒนาและทดสอบในขั้นต่อไป

การพัฒนาจะแบ่งเป็นสองฝั่งคือ การพัฒนาบนเครื่องคอมพิวเตอร์ และการพัฒนาบนเครื่องแม่ข่าย โดยการพัฒนาจะเป็นในลักษณะเริ่มพัฒนาบนเครื่องคอมพิวเตอร์ จากนั้นจึงนำสิ่งที่พัฒนาบนเครื่องคอมพิวเตอร์ย้ายขึ้นไปพัฒนา ทดสอบ และปรับปรุงต่อบนเครื่องแม่ข่าย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อใดๆ ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการนี้จะเกิดขึ้นตลอดการพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ ดังนั้นจึงต้องมีการเตรียมความพร้อมสภาพแวดล้อมให้เหมาะสมต่อการพัฒนาทั้งสองฝั่ง

3.3.5. การเตรียมอุปกรณ์ ฮาร์ดแวร์ และซอฟต์แวร์ สำหรับการพัฒนา

3.3.5.1. ฝั่งการพัฒนาบนเครื่องคอมพิวเตอร์

1.) อุปกรณ์ หรือ ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ที่สามารถเชื่อมต่ออินเทอร์เน็ตได้ขณะพัฒนาโปรแกรม

2.) โปรแกรมและซอฟต์แวร์

- ระบบปฏิบัติการ Windows 10
- โปรแกรม eclipse for PHP
- โปรแกรม XAMPP
- โปรแกรม Source Tree
- โปรแกรม Putty
- โปรแกรม Git
- โปรแกรม Google Chrome
- โปรแกรม Postman

3.3.5.2. ฝั่งการพัฒนาบนเครื่องแม่ข่าย

1.) อุปกรณ์ หรือ ฮาร์ดแวร์

- เครื่องคอมพิวเตอร์ประสิทธิภาพสูงสำหรับทำงานเป็นเครื่องแม่ข่าย

2.) โปรแกรมและซอฟต์แวร์

- ระบบปฏิบัติการ Linux

3.3.5.3. ติดตั้ง และตั้งค่า ซอฟต์แวร์สำหรับการพัฒนา

1.) เครื่องคอมพิวเตอร์

- Eclipse for PHP

2.) เครื่องคอมพิวเตอร์แม่ข่าย

- อะแพชี เว็บเซิร์ฟเวอร์ 2

อะแพชี เว็บเซิร์ฟเวอร์ 2 (Apache webserver 2) เป็นซอฟต์แวร์ที่มีการพัฒนาแบบเปิด (open source) เปลี่ยนเครื่องคอมพิวเตอร์ให้สามารถให้บริการเว็บไซต์ได้โดยการเรียกใช้งานจากเครื่องคอมพิวเตอร์เครื่องอื่นๆ (client) ความพิเศษของอะแพชีคือมีการพัฒนาโมดูลขึ้นมาทำงานร่วมกับอะแพชีมากมาย ซึ่งแต่ละโมดูลก็จะมีความสามารถต่างๆ เช่น mod_php, mod_perl, mod_python ซึ่งจะทำให้อะแพชีสามารถทำงานร่วมกับภาษาอื่นๆได้นอกจากเอชทีเอ็มแอล (HTML) เป็นต้น อะแพชีสามารถทำงานได้บนระบบปฏิบัติการที่หลากหลายทั้ง Windows, Linux ในปริญญาโทฉบับนี้จะนำอะแพชีเว็บเซิร์ฟเวอร์มาติดตั้งบนเครื่องแม่ข่ายเพื่อให้บริการ API บนโพรโทคอล HTTP พอร์ต 80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- พีเอชพี 5.6

เอชพี 5.6 (PHP version 5.6) เป็นซอฟต์แวร์ที่จะต้องลงบนเครื่องคอมพิวเตอร์แม่ข่าย เพื่อให้เครื่องคอมพิวเตอร์แม่ข่าย รู้จักและสามารถประมวลผลภาษาพีเอชพีได้บนเครื่องคอมพิวเตอร์ระบบปฏิบัติการลินุกซ์

- กิท

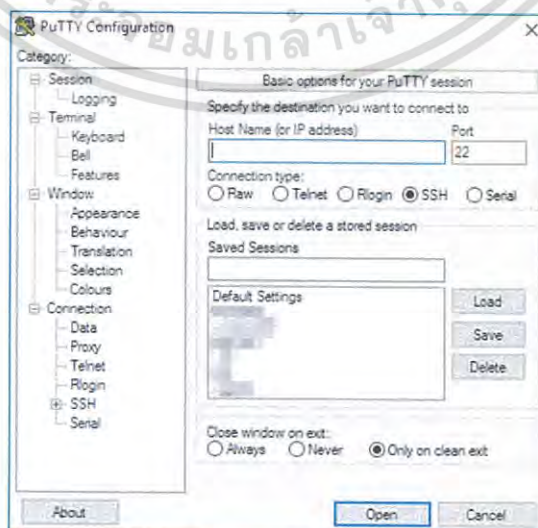
กิท (Git) เป็นซอฟต์แวร์ใช้ในการทำเวอร์ชันคอนโทรล (version control) ซึ่งใช้คู่กับการพัฒนาโปรเจกต์ไปตลอดเวลา ข้อดีของการทำเวอร์ชันคอนโทรล คือ สามารถจัดการการพัฒนาได้อย่างเป็นระบบ โปรเจกต์การพัฒนาทั้งหมดจะอยู่บนระบบคลาวด์ (Git cloud server) สามารถเข้าถึงโปรเจกต์ได้จากทุกที่ที่มีซอฟต์แวร์กิทอยู่ สามารถจัดการการพัฒนาได้หลายแบบ ด้วยความสามารถ บรานช์ (Branch) เพื่อแยกการทดสอบพัฒนา (development) ออกจากการใช้งานจริง (production) โดยสามารถรวมบรานช์เข้าด้วยกันได้ด้วยความสามารถ เมิร์จ (Merge) เมื่อต้องการนำสิ่งที่พัฒนาในบรานช์ development ไปลงบนบรานช์ production รวมถึงทำให้การทำงานร่วมกับทีมสะดวกยิ่งขึ้นไม่จำเป็นต้องคัดลอกและวางโค้ดอีกต่อไป ในปฏิญญาฉบับนี้ จะใช้ซอฟต์แวร์กิทในการทำเวอร์ชันคอนโทรล ทำงานร่วมกับทีม และย้ายโค้ดโปรเจกต์ที่พัฒนาจากเครื่องคอมพิวเตอร์ลงบนเครื่องแม่ข่าย

3.3.6. ขึ้นโครงสร้างและพัฒนา

การขึ้นโครงสร้างเพื่อลงมือพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ จะต้องมีขั้นตอนในการทำงานหลายขั้นตอนรวมถึงต้องใช้ซอฟต์แวร์มากมายในการพัฒนาให้เกิดระบบหลังบ้านและ API ที่สามารถรองรับการทำงานตามที่ได้ออกแบบไว้ได้

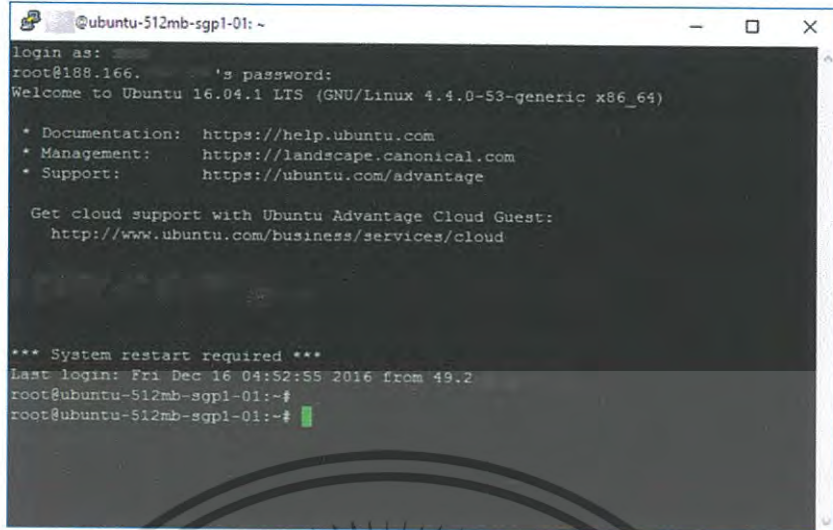
- คอมมานด์ไลน์

การขึ้นโครงสร้างจะใช้วิธีการพิมพ์คำสั่งผ่านหน้าต่างคอมมานด์ไลน์ (command line) โดยทำการ รีโมทซีเคียวเชล (Remote Secure Shell : Remote SSH) ผ่านโปรแกรม Putty ไปยังเครื่องแม่ข่ายและพิมพ์คำสั่งผ่าน หน้าต่างคอนโซล (Console) ของโปรแกรม Putty ดังภาพที่ 3.73



ภาพที่ 3.72 หน้าต่างโปรแกรม Putty

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

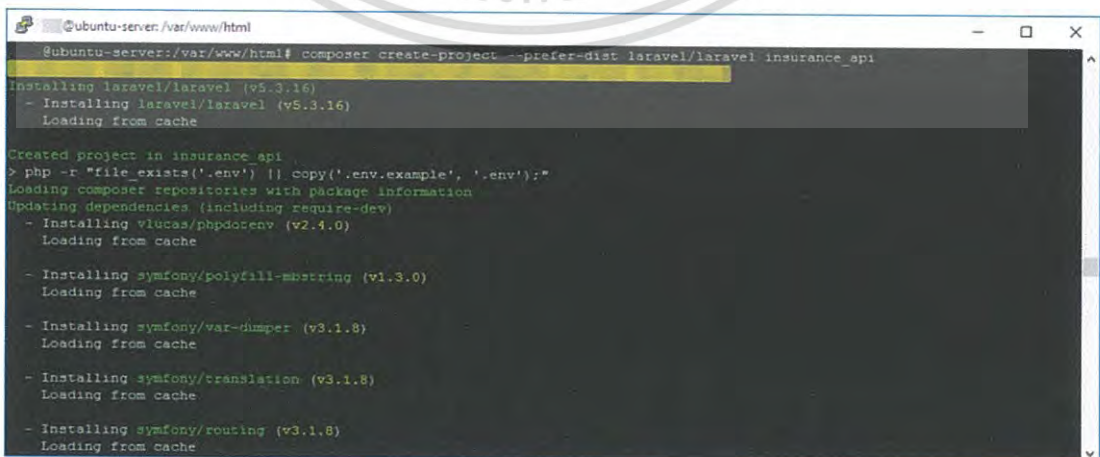


ภาพที่ 3.73 หน้าต่าง console เมื่อ remote SSH ไปยังเครื่องแม่ข่าย

- คำสั่งคอมโพสเซอร์

Composer คือระบบจัดการไลบรารีภายนอกของ php ถูกสร้างขึ้นมาโดยนำแนวคิดมาจากระบบจัดการไลบรารีของ node.js,ruby เดิมระบบจัดการไลบรารีของ php คือ pear ซึ่งไม่มีการพัฒนานานมากแล้ว และแนวคิดของ composer ต่างจาก pear ตรงที่ตัวไลบรารีจะไม่ถูกติดตั้งในระดับที่สามารถเรียกใช้จากที่ไหนก็ได้ แต่จะใช้โฟลเดอร์ vendor สำหรับการเก็บไลบรารีต่างๆ ซึ่ง composer จะจัดการ dependency ที่เขียนด้วยภาษา php โดยมีแหล่งรวม packages ต่างๆอยู่ที่ <http://packagist.org> ซึ่งเป็นแหล่งรวม repository ส่วนของ dependency library จากหลายๆ ที่ไม่ว่าจะเป็น github, bitbucket เป็นต้น

ตัวอย่างการใช้คำสั่ง composer ในการสร้างโปรเจกต์ด้วยสาลาเวลเฟรมเวิร์ค “composer create-project --prefer-dist laravel/laravel your_project_name” ดังภาพที่ 3.74



ภาพที่ 3.74 หน้าต่าง console ขณะสร้างโปรเจกต์สาลาเวลด้วยคำสั่ง composer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.7. เชื่อมต่อกับฐานข้อมูล

การทำงานของระบบจะต้องมีการทำงานควบคู่กับฐานข้อมูล ซึ่งชนิดของฐานข้อมูลก็ขึ้นอยู่กับระบบฐานข้อมูลเดิมของบริษัทประกันว่า เดิมแล้วใช้ฐานข้อมูลชนิดใด ซึ่งบริษัทเอกชนผู้ว่าจ้างในการทำระบบประกันบนโมไบล์แพลตฟอร์มนั้น ใช้ระบบฐานข้อมูล PostgreSQL

PostgreSQL เป็นระบบการจัดการฐานข้อมูลเชิงวัตถุสัมพันธ์ (object-relational) ซึ่งมีการทำงานแบบ ORDBMS สามารถใช้ชุดคำสั่งภาษา SQL ในการเรียกดู บันทึก และแก้ไขข้อมูลได้เกือบทั้งหมด นอกจากนี้ยังเป็นระบบฐานข้อมูลที่มีการพัฒนาแบบเปิด (open-source) ที่ทันสมัยสามารถนำไปใช้งานได้โดยไม่เสียค่าใช้จ่ายใดๆ มีการพัฒนาโดยมหาวิทยาลัยแคลิฟอร์เนีย (Berkeley Computer Science department, University of California.)

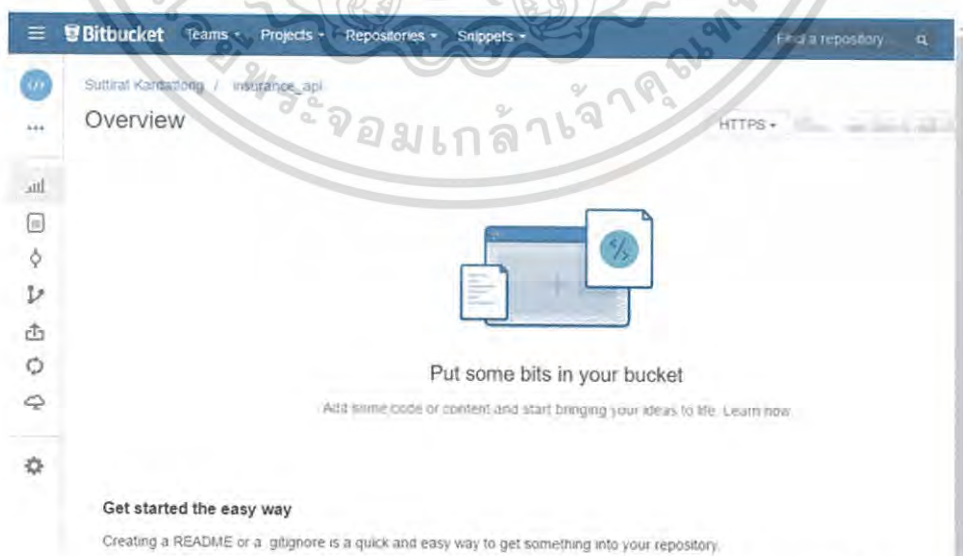
PostgreSQL สามารถทำงานบนระบบปฏิบัติการได้ทั้ง Linux ระบบปฏิบัติการที่พัฒนาจาก UNIX เช่น AIX, BSD, HP-UX, SGI Irix, Mac OS X, Solaris, Tru64 และ Windows

3.3.8. ระบบเวอร์ชันคอนโทรล Git

Git เป็นระบบการทำ Version Control ที่มีหน้าที่ในการจัดเก็บการเปลี่ยนแปลงของไฟล์ในโปรเจกต์ สามารถที่จะเรียกดูหรือย้อนกลับไปดูเวอร์ชันต่างๆของโปรเจกต์เวลาใดก็ได้ หรือแม้แต่ว่าแต่ละไฟล์ใครเป็นคนเพิ่มหรือแก้ไข หรือดูว่าไฟล์นั้นๆ ถูกเขียนโดยใครบ้างก็สามารถทำได้ การนำระบบ Git มาใช้งานจะทำให้การพัฒนาซอฟต์แวร์สะดวกมากยิ่งขึ้น การใช้งานจะต้องอาศัย Git Server และ Git Client

- Bitbucket.org

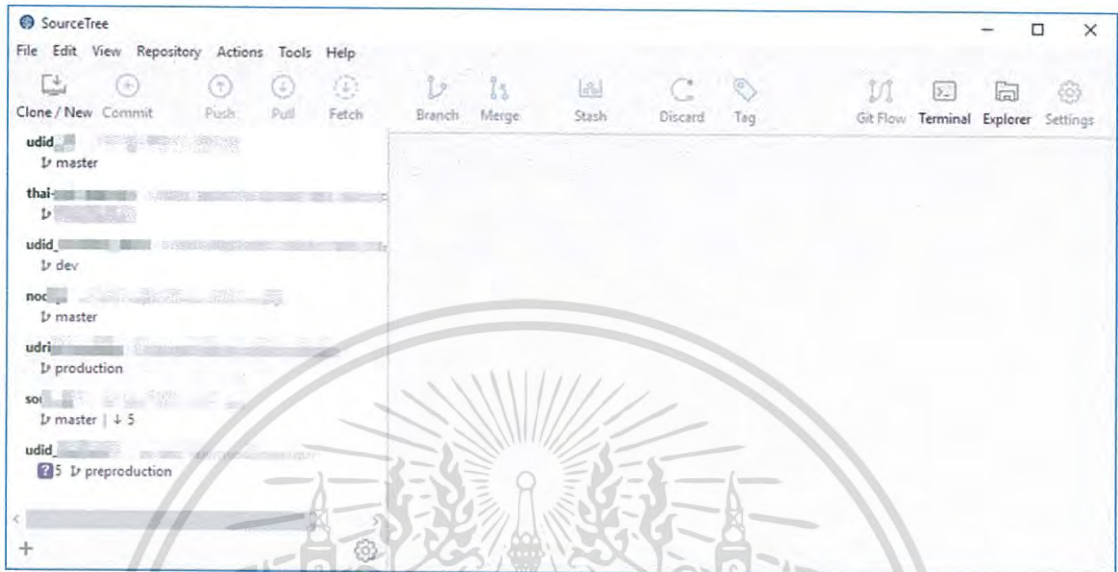
การอัปโหลดไฟล์ขึ้นระบบ Git จะต้องมี Git Server ซึ่งบริษัท Octagon Interactive เลือกใช้ Bitbucket.org ซึ่งสามารถสร้าง Git Repository ไว้สำหรับเก็บไฟล์โปรเจกต์ไว้บนระบบคลาวด์ได้



ภาพที่ 3.75 หน้าเว็บไซต์ Bitbucket.org หลังจากสร้าง Repository ใหม่

- Source Tree

การโคลนโปรเจกต์จาก Git Server เลือกใช้โปรแกรม Source Tree ดังภาพที่ 3.76 ซึ่งเป็นโปรแกรมประเภท Git Client ที่ได้รับความนิยมสูง



ภาพที่ 3.76 หน้าต่างโปรแกรม Source Tree

3.3.9. ระบบเว็บเซิร์ฟเวอร์สำหรับเครื่องคอมพิวเตอร์ในการพัฒนา

ในระหว่างขั้นตอนการพัฒนา จะต้องมีการทดสอบโปรแกรมที่เขียนขึ้น จึงจำเป็นต้องจำลองเครื่องแม่ข่ายขึ้นบนเครื่องคอมพิวเตอร์เพื่อทดสอบการทำงานในการเขียนฟังก์ชันต่างๆ ก่อนที่จะทดสอบการทำงานบนเครื่องแม่ข่ายจริง

- โปรแกรม XAMPP

XAMPP คือโปรแกรมสำหรับจำลองเครื่องคอมพิวเตอร์ ให้ทำงานในลักษณะของ Web Server นั่นคือเครื่องคอมพิวเตอร์ที่ใช้งาน XAMPP จะเป็นทั้งเครื่อง Server และเครื่อง Client ในเครื่องเดียวกัน ทำให้ไม่ต้องเชื่อมต่อกับ Internet เพื่อทดสอบเว็บไซต์หรือระบบ API ที่สร้างขึ้น

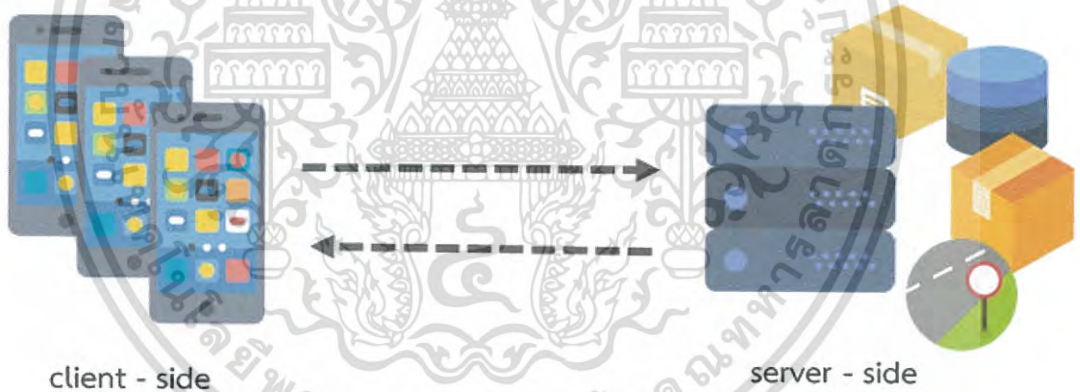
XAMPP ประกอบด้วย Apache, PHP, MySQL, PHP MyAdmin, Perl ซึ่งเป็นชุดโปรแกรม สำหรับสร้างเว็บไซต์ที่ได้รับความนิยม XAMPP รองรับระบบปฏิบัติการ เช่น Windows, Linux, Apple ทำงานได้ทั้งบนระบบปฏิบัติการแบบ 32 bit และ 64 bit



ภาพที่ 3.77 หน้าต่างโปรแกรม XAMPP

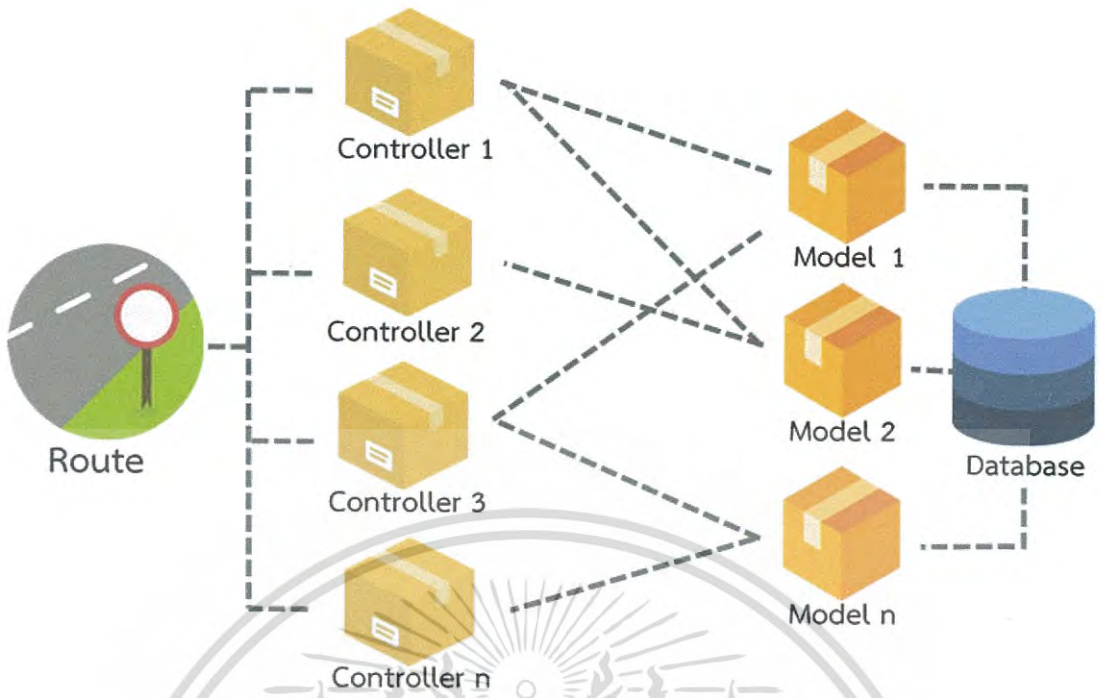
3.3.10. ระบบภายใต้สถาปัตยกรรม โมเดล วิว คอนโทรลเลอร์

ระบบประกันภัยบนโมบายล์แพลตฟอร์ม ที่ได้ออกแบบและพัฒนาจะใช้หลักการทำงานแบบ Client-Server และจะใช้การติดต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลระหว่างทั้งสองฝั่งผ่าน API ในการรับส่ง request และ response ดังภาพที่ 3.78

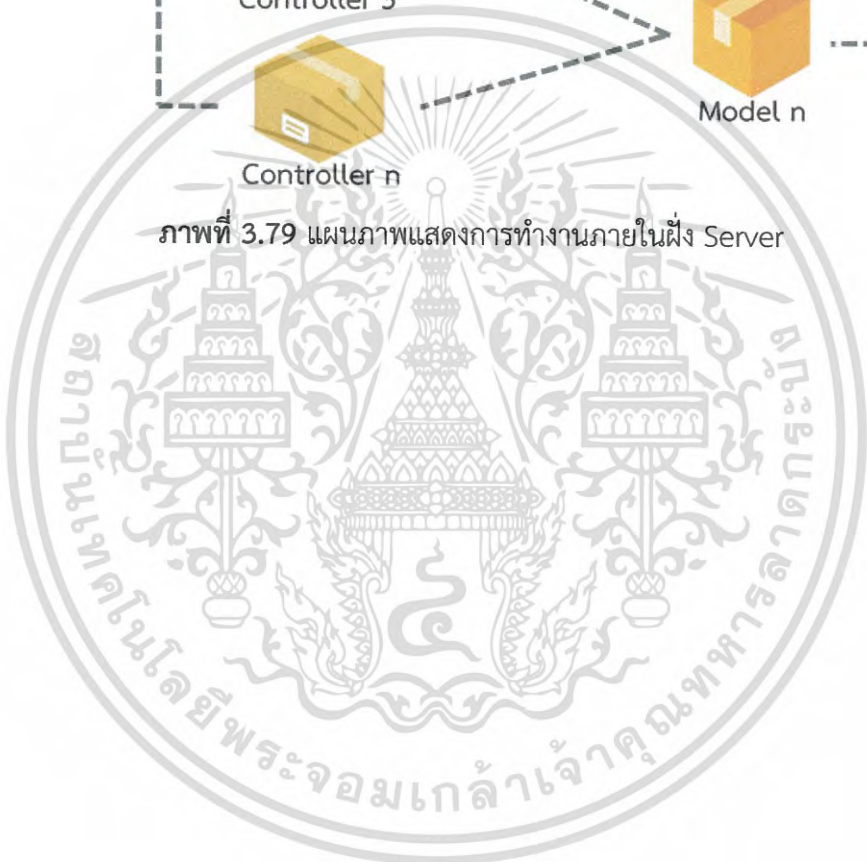


ภาพที่ 3.78 แผนภาพแสดงการทำงานในรูปแบบของ Client-Server

ซึ่งภายในฝั่งเซิร์ฟเวอร์ (Server) จะประกอบด้วยการทำงานในหลายๆส่วนภายใต้การพัฒนาระบบด้วยสถาปัตยกรรมออกแบบ โมเดล วิว คอนโทรลเลอร์ (MVC) ซึ่งในระบบที่พัฒนาจะประกอบด้วย เราท (Route) โมเดล (Model) คอนโทรลเลอร์ (Controller) จะทำงานสอดคล้องกันเพื่อให้บริการเว็บเซอร์วิสแก่ฝั่ง ไคลเอนต์ (client) ซึ่งเป็นโมบายล์แอปพลิเคชัน ดังภาพที่ 3.79



ภาพที่ 3.79 แผนภาพแสดงการทำงานภายในฝั่ง Server



บทที่ 4 ผลการวิจัย (ส่วนการออกแบบ)

ในส่วนของการออกแบบมีผลที่ได้ออกมาสี่ช่วง ได้แก่ ผลของการสร้างพีเจอร์ให้กับแอปพลิเคชันผลของการสร้างแบบจำลองแอปพลิเคชัน ผลของแบบจำลองแอปพลิเคชันที่ออกมาในแต่ละหน้าจอ และผลของทำทดสอบประสิทธิภาพการใช้งานแอปพลิเคชันของผู้ใช้งาน ซึ่งจะต้องออกแบบให้ตรงตามความต้องการของลูกค้าและถูกต้องตรงตามหลักการใช้งานของผู้ใช้ที่ถูกวางมาตรฐานไว้โดยผู้เก็ลในแต่ละยุคสมัย แต่ก็สามารถประยุกต์ใช้ได้ตามความเหมาะสมและตามไอเดียของนักออกแบบ โดยวัดจากตัวแปรที่สำคัญคือ การออกแบบที่ตรงตามความต้องการของลูกค้าและง่ายต่อการพัฒนาแอปพลิเคชัน

4.1. ผลของการออกแบบแอปพลิเคชัน

4.1.1. ผลของการสร้างพีเจอร์ให้กับแอปพลิเคชัน

ตัวแปรที่ใช้ในการวัดการสร้างพีเจอร์ คือ การออกแบบที่ทำให้ลูกค้าพัฒนาของนักพัฒนาแอปพลิเคชันเกิดความพึงพอใจ ถ้าหากส่วนไหนที่ลูกค้าไม่พึงพอใจ นักออกแบบควรจะปรึกษานักพัฒนาแล้วว่าถ้าหากปรับเปลี่ยนเป็นรูปแบบนี้จะส่งผลต่อการพัฒนาหรือไม่ เพื่อที่จะไปบอกข้อจำกัดในการพัฒนาให้ลูกค้าเข้าใจว่าเหตุใดนักออกแบบจึงออกแบบมาในลักษณะนี้ ดังตารางที่ 4.1

ตารางที่ 4.1 ตารางแสดงความพึงพอใจและข้อเสนอแนะของลูกค้าและนักพัฒนาต่อพีเจอร์

รายการพีเจอร์	ความพึงพอใจและข้อเสนอแนะ	
	ลูกค้า	นักพัฒนา
1. ให้เมนูหลักอยู่ด้านล่างสุดของจอ	พอใจ	พอใจ
2. หน้าหลักเป็นหน้าซื้อประกันภัย โดยมีลักษณะการจัดวางเป็นลักษณะการเรียงลงมา	พอใจ	ต้องการให้ลักษณะของการจัดเป็นรูปแบบเดียวกันทั้งหมด
3. หน้าซื้อประกันจะมีปุ่มให้สามารถโทรไปสอบถามรายละเอียดประกันภัยกับทางบริษัทได้โดยตรง หรือต้องการให้ทางบริษัทโทรมาก็ได้	พอใจ	พอใจ
4. หน้าโปรโมชันสามารถแชร์ผ่านทางโซเชียลเน็ตเวิร์คได้	พอใจ	พอใจ
5. หน้าเคลมมีขั้นตอนการแจ้งจุดเกิดเหตุ เวลาและรถที่เกิดเหตุ และมีการถ่ายรูปตามลำดับก่อนแจ้งเคลม	ตัดขั้นตอนการถ่ายรูปออกไปก่อนในช่วงแรก	ขึ้นอยู่กับบริษัทประกัน

6. หน้าบัตรประกันจะใช้เป็นรูปแบบดิจิทัล การ์ด ซึ่งสามารถต่ออายุกรรมธรรม์ได้ ดูความคุ้มครองได้ และสแกนคิวอาร์โค้ดได้	พอใจ	ควรยืนยันตัวตนเข้าใช้งานครั้งแรก/ตัดการสแกนคิวอาร์โค้ดออกในการพัฒนาช่วงแรก
7. หน้าแผนที่แสดงอู่ซ่อมรถในเครือ และโรงพยาบาล โดยสามารถกดนำทางได้ และสามารถค้นหาสถานที่ได้ด้วยตัวเอง	พอใจ	พอใจ
8. หน้าโทรถูกเงินจะแสดงเบอร์สำหรับเรียก รถพยาบาล โทรแจ้งตำรวจ และโทรหาบริษัทประกัน	เพิ่มเบอร์ฉุกเฉิน ส่วนตัวที่ผู้ใช้งาน	พอใจ
9. หน้าโปรไฟล์จะแสดงข้อมูลส่วนตัวและที่อยู่ ที่ตรงตามข้อมูลบนเว็บไซต์ โดยสามารถเข้าเมนูนี้ได้จากทุกหน้า	ต้องการให้เพิ่มข้อมูล ด้านสุขภาพ	ขึ้นอยู่กับบริษัทประกัน
10. หน้าตั้งค่าจะสามารถเปิดปิดการแจ้งเตือนของแอปพลิเคชันได้ แสดงความคิดเห็น และออกจากระบบได้ โดยสามารถเข้าเมนูนี้ได้จาก หน้าหลักเท่านั้น	พอใจ	พอใจ
11. ให้ชื่อแสดงเมนูอยู่ชัดเจนตามมาตรฐานที่ กูเกิ้ลออกแบบไว้	ขอให้อยู่กึ่งกลางจอ	ขึ้นอยู่กับบริษัทประกัน
12. ใส่มาสคอตเป็นรูปฝั่งของทางบริษัท ประกันเวลาแสดงไต่อะล็อกแจ้งเตือนบาง ประเภท	พอใจ	ขึ้นอยู่กับบริษัทประกัน
13. หน้าสมัครสมาชิกให้กรอกข้อมูลให้น้อย ที่สุดเพียง 5 ช่อง คือ ชื่อ, สกุล, อีเมล, เบอร์ โทร และรหัสผ่าน เพื่อลดเวลาในการเข้าถึง แอปพลิเคชันของผู้ใช้งาน	พอใจ	พอใจ
14. หน้าเข้าสู่ระบบจะมีช่องให้ใส่อีเมลและ รหัสผ่าน หรือสามารถเข้าสู่ระบบผ่านทางเฟซบุ๊กได้ และปุ่มลิมรหัสผ่าน	พอใจ	พอใจ

4.1.2. ผลของการสร้างแบบจำลองแอปพลิเคชัน

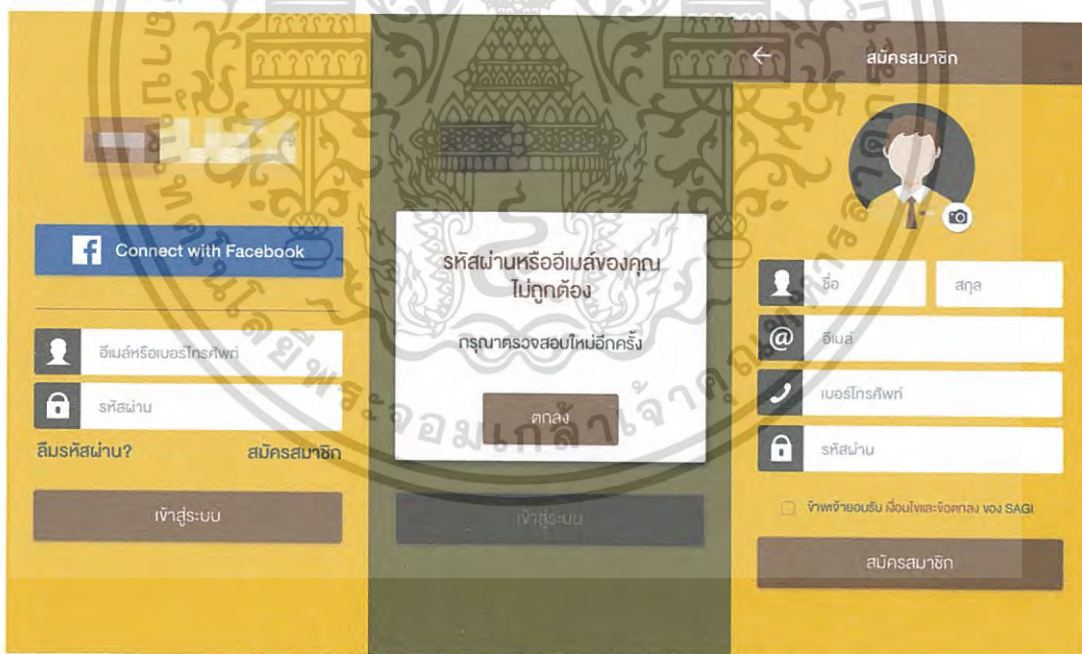
ตัวแปรที่ใช้ในการวัดการสร้างแบบจำลองจะยึดหลักเดียวกันกับการสร้างพีเจอร แต่จะวัด จากความพึงพอใจของลูกค้าเพียงอย่างเดียวเท่านั้น เช่น การใช้สีที่ตรงกับแบรนด์ของสินค้าหรือ บริษัท ตำแหน่งการจัดวางของปุ่ม รูปภาพ ไอคอน และข้อมูลต่างๆ บนแอปพลิเคชัน ก่อนที่จะส่งต่อ ให้กับนักพัฒนานลงมือพัฒนาแอปพลิเคชันจริง

ตารางที่ 4.2 ตารางแสดงความพึงพอใจและข้อเสนอแนะของลูกค้ำต่อแบบจำลองแอปพลิเคชัน

รายการ	ความพึงพอใจและข้อเสนอแนะของลูกค้ำ
1. ใช้สีหลักสองสีคือ เหลือง และน้ำตาล	พอใจ
2. ใช้พื้นหลังเป็นลายรังผึ้ง	พอใจ
3. ใช้ภาพให้สื่อถึงประเภทของประกัน	พอใจ
4. ใช้ไอคอนมาตรฐานที่ผู้ใช้งานเห็นแล้วรู้ว่าต้องการสื่อถึงอะไร	พอใจในบางส่วนหนึ่ง บางไอคอนอยากให้ใช้เป็นตัวอักษรเพื่อถ่ายทอดผู้สูงอายุ
5. ใช้สีที่แตกต่างกันเพื่อแบ่งประเภทของบัตรประกันภัย	พอใจ
6. หน้าแสดงแผนที่อยู่ช่อมรดกและโรงพยาบาลใช้ไอคอนที่สื่อถึงประเภทของสถานที่และสีของพื้นที่แตกต่างกัน	ต้องการให้ใช้พื้นที่เดียวกัน

4.1.3. ผลของแบบจำลองแอปพลิเคชันที่ออกมาในแต่ละหน้าจอ

4.1.3.1. หน้าสมัครสมาชิก และหน้าเข้าสู่ระบบ

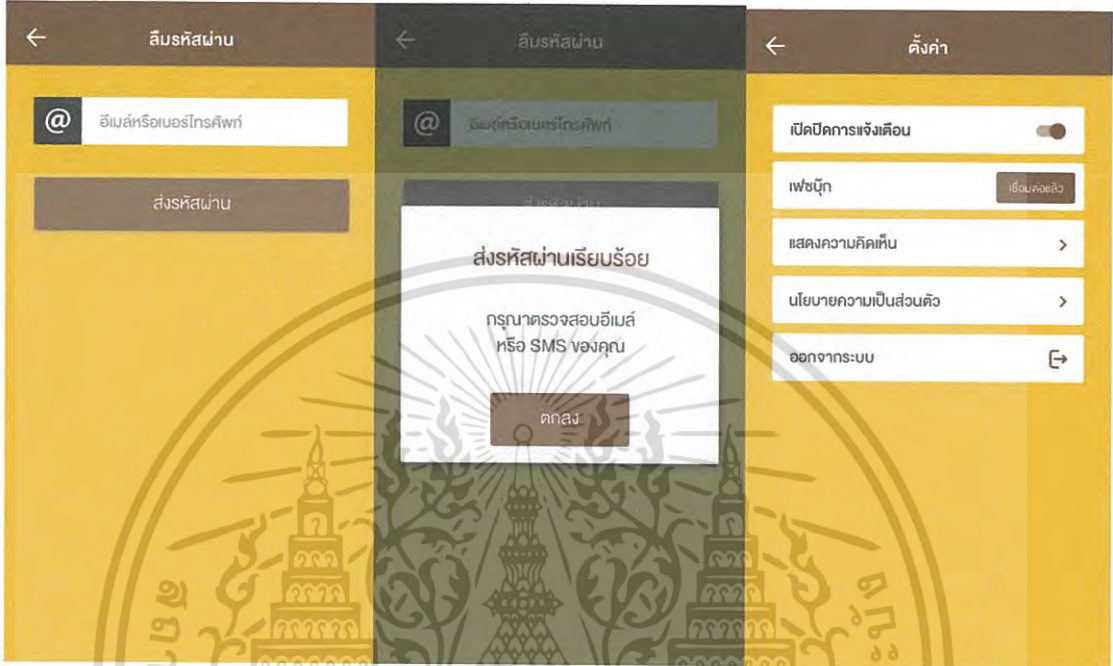


ภาพที่ 4.1 ตัวอย่างหน้าสมัครสมาชิก และหน้าเข้าสู่ระบบ

จากภาพที่ 4.1 หน้าเข้าสู่ระบบจะประกอบไปด้วยช่องใส่อีเมลหรือเบอร์โทรศัพท์ และช่องใส่ รหัสผ่าน ปุ่มสมัครสมาชิก ปุ่มลืรหัสผ่าน ปุ่มเข้าสู่ระบบ และปุ่มเข้าสู่ระบบผ่านทางเฟซบุ๊ก (Facebook) ด้านบนสุดจะเป็นโลโก้ของแอปพลิเคชัน ในหน้าสมัครสมาชิก จะมีช่องสำหรับใส่ชื่อ นามสกุล อีเมล เบอร์โทรศัพท์ และรหัสผ่าน ซึ่งเป็นมาตรฐานที่แอปส่วนใหญ่ต้องมีเป็นเอกสารณเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้มาใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติ ถ้าหากกดสมัครผ่านทาง เฟซบุ๊ก (Facebook) ช่องเหล่านี้จะมีการดึงข้อมูลมาจากเฟซบุ๊ก แล้วให้ผู้ใช้งานใส่เพียงรหัสผ่านเท่านั้น ส่วนด้านบนสุดเป็นรูปโปรไฟล์ของผู้ใช้งานซึ่งสามารถอัปโหลดได้เองจากบนมือถือของผู้ใช้งาน หรือดึงภาพมาจากเฟซบุ๊กก็ได้ เมื่อกรอกข้อมูลครบถ้วนแล้วผู้ใช้งานจะต้องยอมรับเงื่อนไขของทางบริษัทประกันก่อน จึงจะสามารถกดปุ่มสมัครสมาชิกได้

4.1.3.2. หน้าตั้งค่า และหน้าลิ้มรสผ่าน



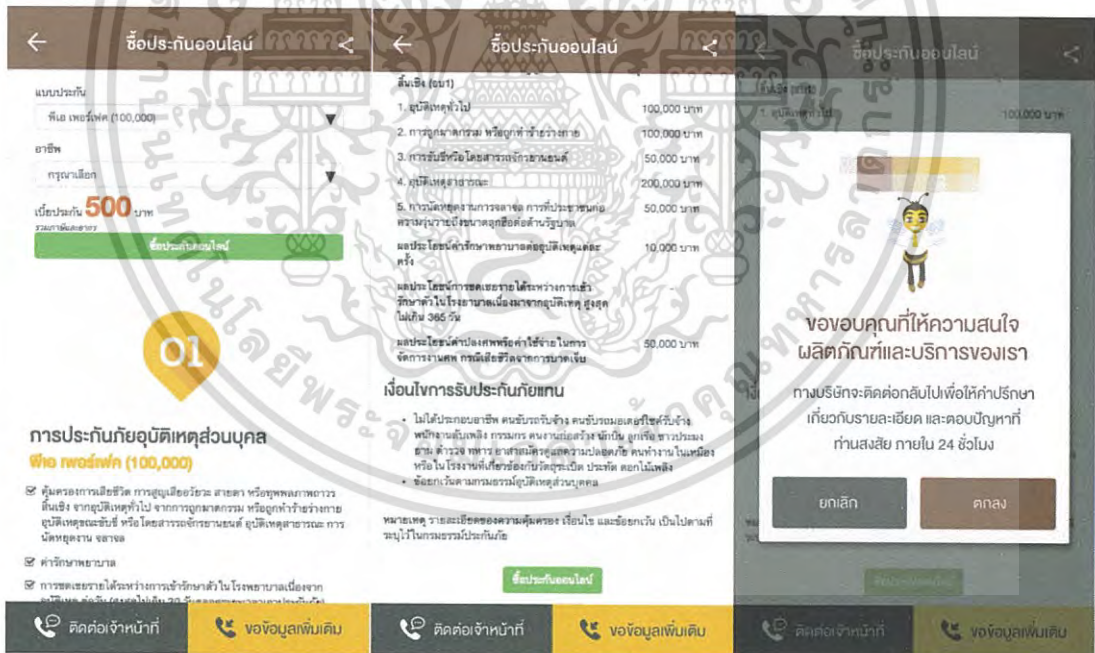
ภาพที่ 4.2 ตัวอย่างหน้าตั้งค่า และหน้าลิ้มรสผ่าน

จากภาพที่ 4.2 หน้าตั้งค่าจะมีปุ่มสำหรับกดเปิดหรือปิดการแจ้งเตือนทุกอย่างบนแอปพลิเคชัน ปุ่มแสดงว่าขณะนี้ผู้ใช้งานเชื่อมต่อกับเฟซบุ๊กอยู่หรือไม่ ปุ่มสำหรับแสดงความคิดเห็นเกี่ยวกับแอปพลิเคชัน ปุ่มนโยบายความเป็นส่วนตัว และปุ่มออกจากระบบ ซึ่งหน้าตั้งค่านี้อาจสามารถเข้าได้จากการกดไอคอนบนหน้าหลักเท่านั้น ส่วนหน้าลิ้มรสผ่าน จะมีช่องสำหรับให้กรอกอีเมลหรือเบอร์โทรศัพท์ที่ต้องการจะให้ส่งรหัสผ่านไป

4.1.3.3. หน้าหลักและหน้าซื้อประกันผ่านเว็บไซต์ด้วยแอปพลิเคชัน



ภาพที่ 4.3 ตัวอย่างซื้อประกันหรือหน้าหลัก

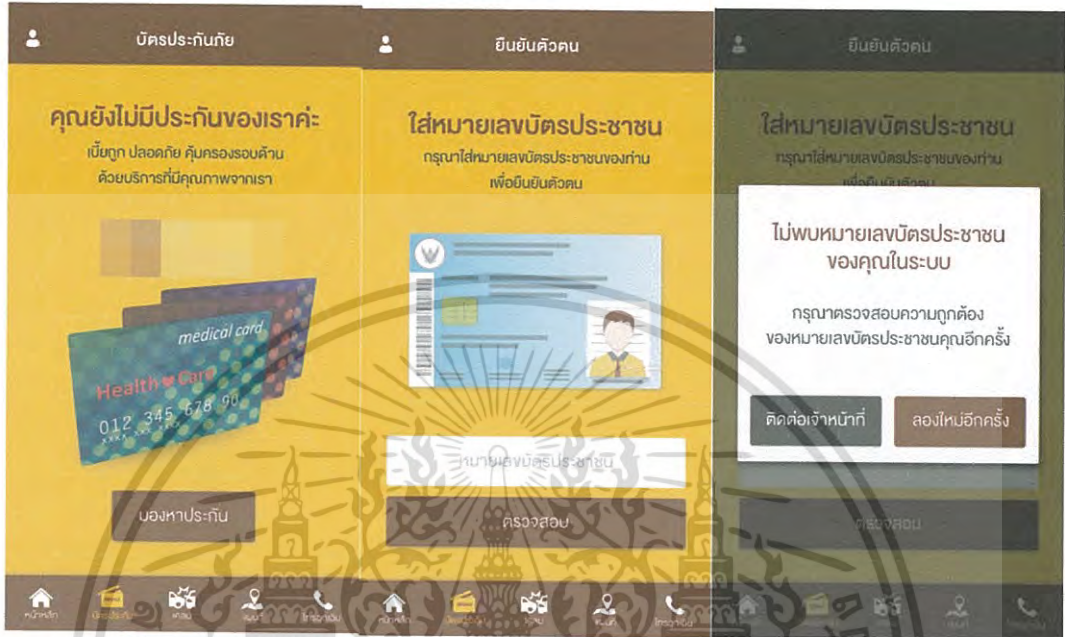


ภาพที่ 4.4 ตัวอย่างหน้าเว็บประกันเมื่อซื้อประกันด้วยแอปพลิเคชัน

จากภาพที่ 4.3 เป็นหน้าหลักซึ่งเป็นหน้าสำหรับซื้อประกันออนไลน์ผ่านทางแอป โดยจะเรียงกันเป็นลักษณะการ์ด (Card View) แบ่งเป็นประเภทของประกัน และเรียงจากประเภทของประกันที่มีคนซื้อมากที่สุดลงมาถึงประกันที่มีคนซื้อน้อยที่สุดตามลำดับ ด้านบนสุดของหน้าจะเป็นโปรโมชันของทางบริษัทประกันภัย และโปรโมชันอื่นๆก็สามารถไปแทรกขึ้นระหว่างการ์ด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

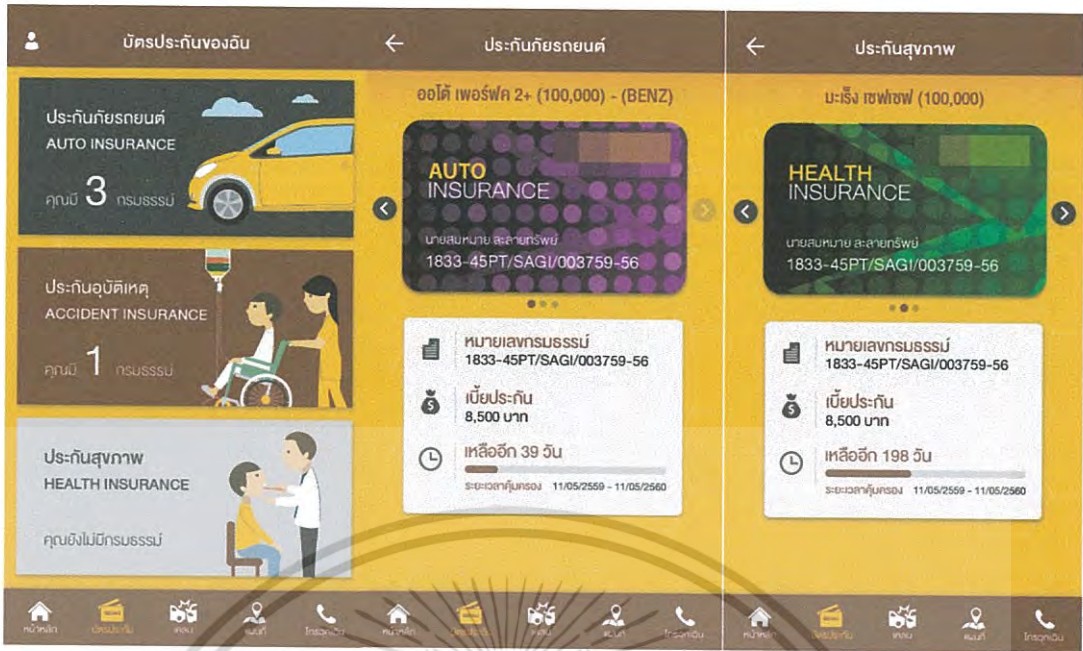
ได้ ส่วนในภาพที่ 4.4 เป็นหน้าเว็บไซต์ของทางบริษัทประกันที่จะแสดงเป็นลักษณะเว็บวิว (Web View) บนแอปพลิเคชัน โดยจะมีปุ่มสำหรับโทรสอบถามรายละเอียดกับเจ้าหน้าที่หรือให้เจ้าหน้าที่โทรมาแนะนำเกี่ยวกับประกันก็ได้เช่นกัน

4.1.3.4. หน้าบัตรประกันภัยและหน้ายืนยันตัวตน

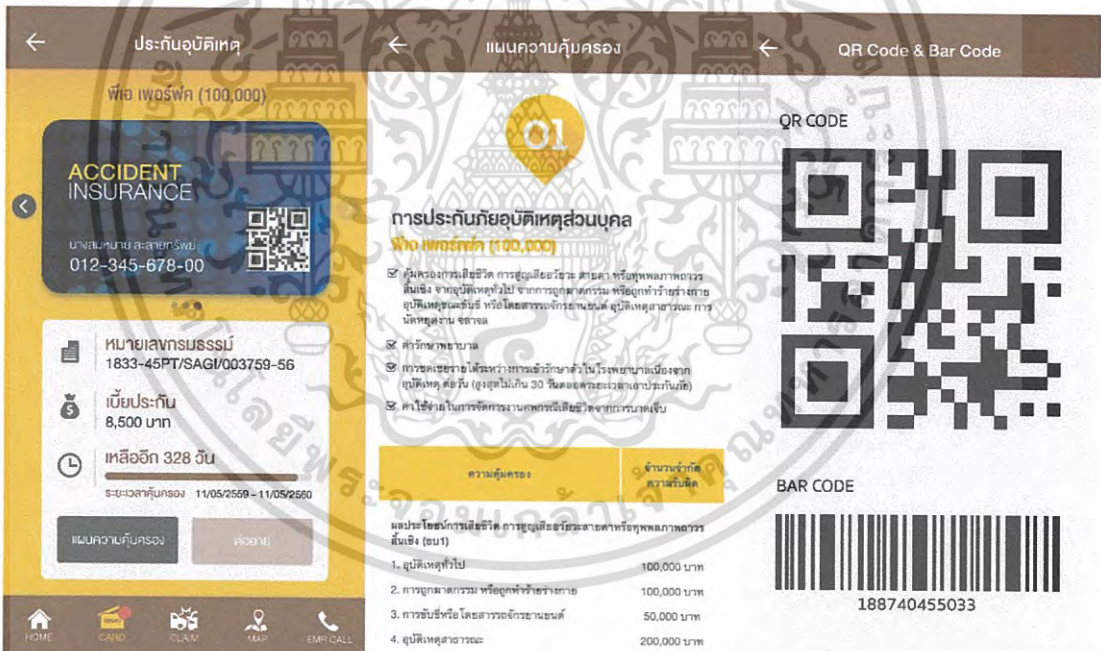


ภาพที่ 4.5 ตัวอย่างหน้าบัตรประกันสำหรับผู้ที่ยังไม่มีประกัน และหน้ายืนยันตัวตน

จากภาพที่ 4.5 ในหน้าบัตรประกันภัยจะแสดงให้เห็นให้ผู้ใช้งานรู้ว่าผู้ใช้งานยังไม่มีกรมธรรม์ของทางบริษัทประกันภัย ซึ่งจะมีปุ่มลัดสำหรับกดกลับไปซื้อประกันที่หน้าหลัก ถ้าหากผู้ใช้งานถือกรมธรรม์เรียบร้อยแล้ว การเข้าใช้งานในครั้งแรกจะต้องกรอกรหัสบัตรประชาชนเพื่อยืนยันตัวตนก่อนเข้าใช้งานพีเจอาร์บัตรประกันภัยนี้



ภาพที่ 4.6 ตัวอย่างหน้าบัตรประกันภัย สำหรับผู้ใช้ที่มีประกันแล้ว



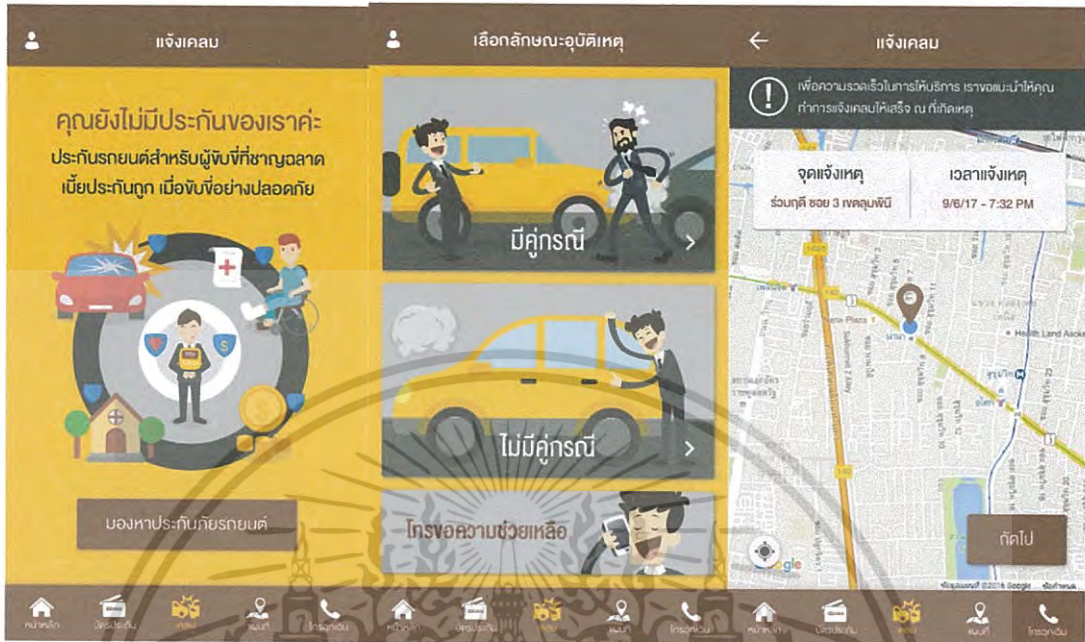
ภาพที่ 4.7 ตัวอย่างหน้าบัตรประกันภัย สำหรับผู้ใช้ที่มีประกันแล้ว (ฟีเจอร์ที่จะสร้างในอนาคต)

จากภาพที่ 4.6 เป็นหน้าสำหรับผู้ใช้งานที่ถือครองกรมธรรม์เรียบร้อยแล้ว มีการกรอกรหัสบัตรประชาชนเพื่อยืนยันตัวตนอย่างถูกต้อง เมื่อเข้ามาจะเห็นว่าผู้ใช้งานมีประกันภัยประเภทไหนอยู่ก็กรมธรรม์ เมื่อกดเข้าไปอีกก็จะเป็นหน้าบัตรประกันภัยของผู้ใช้งานซึ่งสามารถดูรายละเอียดต่างๆได้ ดูความคุ้มครองได้ ดูอายุกรมธรรม์ของผู้ใช้งาน และสามารถต่ออายุประกันผ่านแอปพลิเคชันได้ (ส่วนฟีเจอร์ที่คาดว่าจะเพิ่มในอนาคต คือ ฟีเจอร์สแกนคิวอาร์โค้ดและบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้ทางสถานพยาบาลตรวจสอบข้อมูลของผู้ใช้งานได้ รวมไปถึงฟีเจอร์ดูความคุ้มครอง และฟีเจอร์ต่ออายุกรมธรรม์ เป็นต้น)

4.1.3.5. หน้าแจ้งเคลม

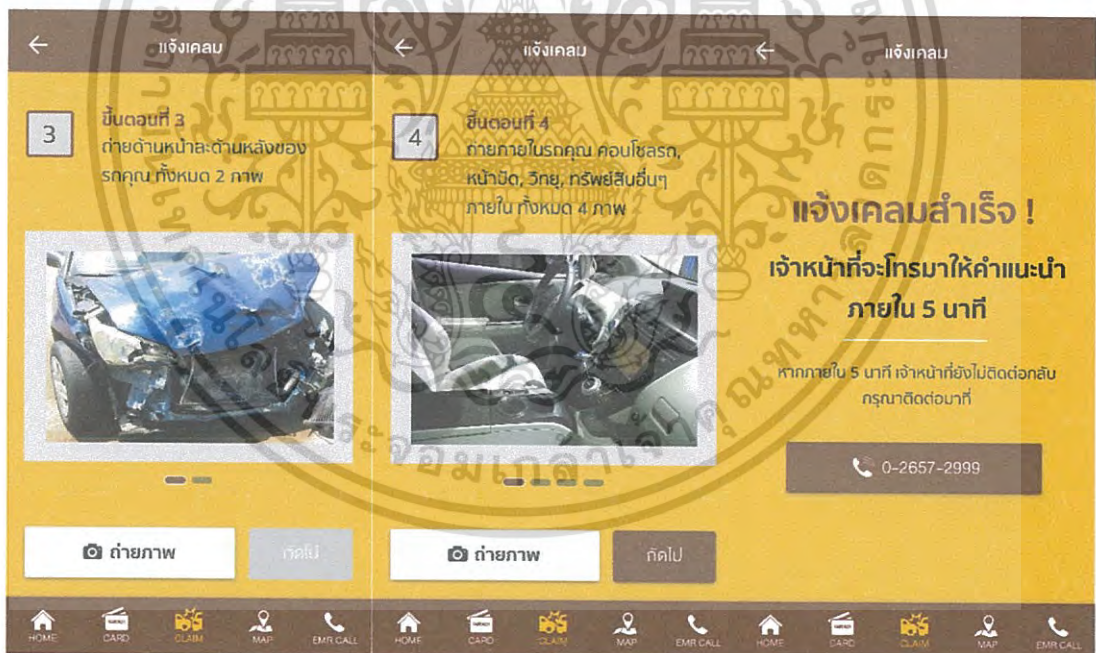


ภาพที่ 4.8 ตัวอย่างหน้าเคลม (Claim)

จากภาพที่ 4.8 ในภาพทางซ้ายสุดเป็นหน้าสำหรับผู้ใช้งานที่ยังไม่มีประกันรถยนต์ ซึ่งด้านล่างจะมีปุ่มลัดสำหรับกลับไปหน้าซื้อประกันภัย ถ้าหากผู้ใช้งานที่มีประกันรถยนต์แล้วมีการยืนยันตัวตนด้วยหมายเลขบัตรประชาชนเพื่อเข้าใช้งานในครั้งแรกแล้ว จะมีหน้าการแจ้งเคลมแบบมีคู่กรณี และไม่มีคู่กรณี ขึ้นมา และจะมีปุ่มด้านล่างสุดสำหรับโทรขอความช่วยเหลือจากเจ้าหน้าที่เคลม



ภาพที่ 4.9 ตัวอย่างหน้าแจ้งเคลม แบบไม่มีคู่มือ (1)



ภาพที่ 4.10 ตัวอย่างหน้าแจ้งเคลม แบบไม่มีคู่มือ (2)

จากภาพที่ 4.8 - ภาพที่ 4.10 เป็นหน้าแจ้งเคลมแบบไม่มีคู่มือ โดยวิธีการแจ้งเคลมนั้นจะเริ่มด้วยการแจ้งจุดเกิดเหตุ แจ้งทะเบียนรถยนต์ที่เกิดอุบัติเหตุ แจ้งชื่อและเบอร์โทรศัพท์ของคนขับและผู้แจ้งเคลม เหตุที่ออกแบบแยกไว้สองช่อง เพราะมีโอกาสที่ผู้แจ้งเคลมและคนขับจะไม่ใช่คนเดียวกัน เมื่อผ่านขั้นตอนนี้ จะเป็นขั้นตอนของการถ่ายรูปจุดเสียหายและสภาพของรถยนต์ที่เกิดอุบัติเหตุตามมุมต่างๆที่ทางบริษัทประกันต้องการ เมื่ออกดแจ้งเคลมเจ้าหน้าที่จะ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนข้อมูลการใช้งานโดยไม่เปิดเผยต่อสาธารณชน

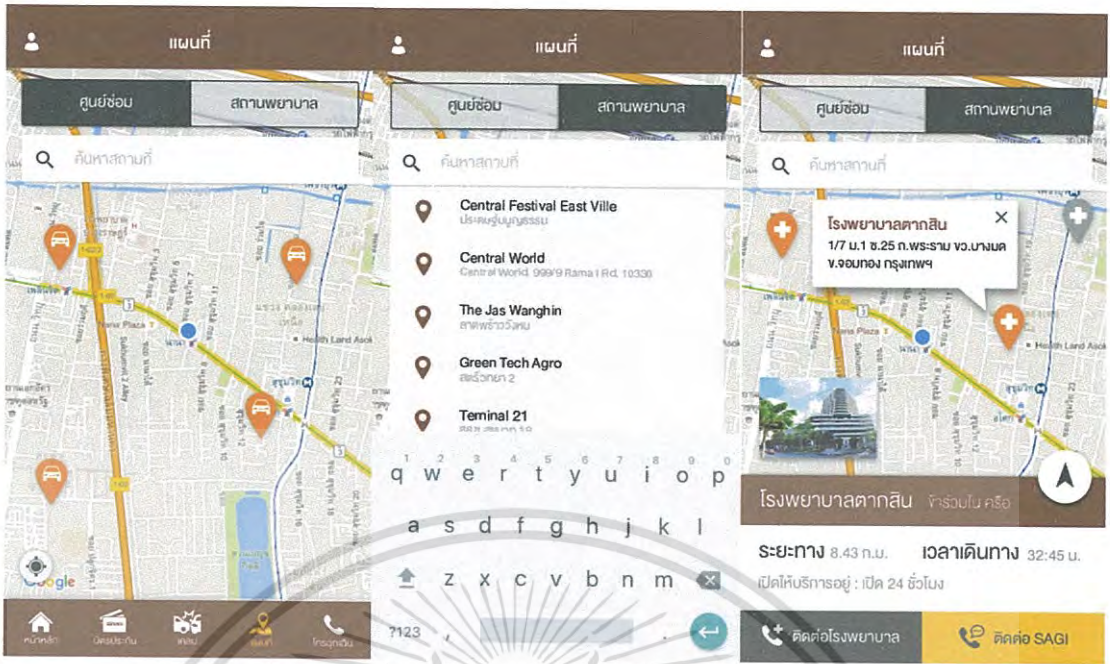
ติดต่อกลับมาภายใน 5 นาที โดยที่ลูกค้าไม่จำเป็นต้องรอเจ้าหน้าที่มาที่จุดเกิดเหตุเอง (พีเจอาร์ถ่ายภาพนี้ยังไม่มีการพัฒนาในเวอร์ชันแรก จะมีการพัฒนาต่อในอนาคตหรือเวอร์ชันต่อไป จึงต้องให้เจ้าหน้าที่มายังจุดแจ้งเหตุแบบปกติก่อน)



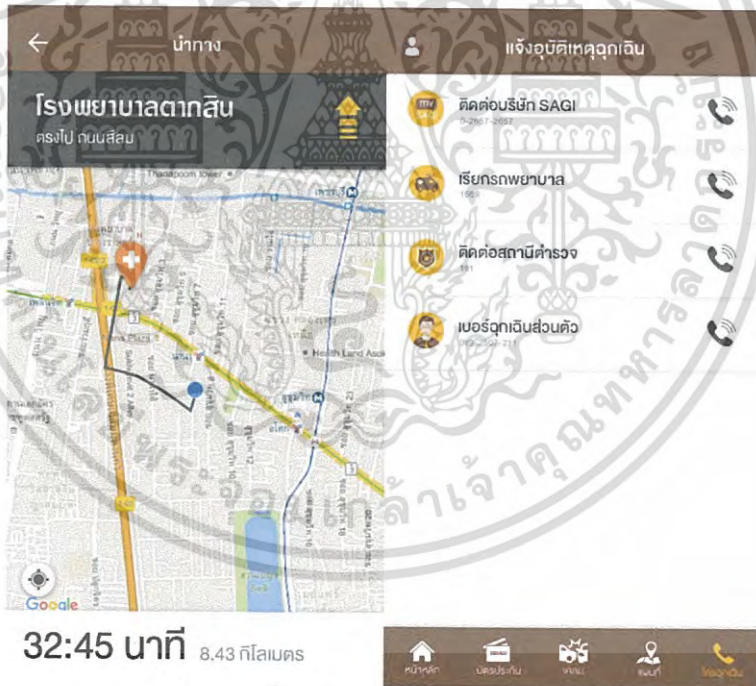
ภาพที่ 4.11 ตัวอย่างหน้าแจ้งเคลม แบบมีคู่มือ

จากภาพที่ 4.11 เป็นหน้าการแจ้งเคลมแบบมีคู่มือ ซึ่งการแจ้งเคลมในแบบนี้ทางบริษัทประกันจำเป็นต้องให้เจ้าหน้าที่มาที่จุดเกิดเหตุเพื่อตกลงว่าใครผิดและใครถูก ในแอปพลิเคชันจึงออกแบบให้แจ้งเคลมโดยกรอกข้อมูลที่จำเป็นเท่านั้น เช่น สถานที่เกิดเหตุ เวลาเกิดเหตุ รถที่เสียหาย คนขับ และผู้แจ้งเคลม เป็นต้น ส่วนเรื่องการถ่ายรูปเจ้าหน้าที่จะมาดำเนินการในที่เกิดเหตุ พร้อมทั้งมาตกลงกันว่าใครผิดและใครถูกด้วยตัวเจ้าหน้าที่เอง

4.1.3.6. หน้าแสดงแผนที่ อุช่อมารถ โรงพยาบาล และหน้าโทรฉุกเฉิน



ภาพที่ 4.12 ตัวอย่างหน้าแสดงแผนที่ที่อยู่ช่อมรถ และโรงพยาบาล



ภาพที่ 4.13 ตัวอย่างหน้านำทาง และโทรฉุกเฉิน

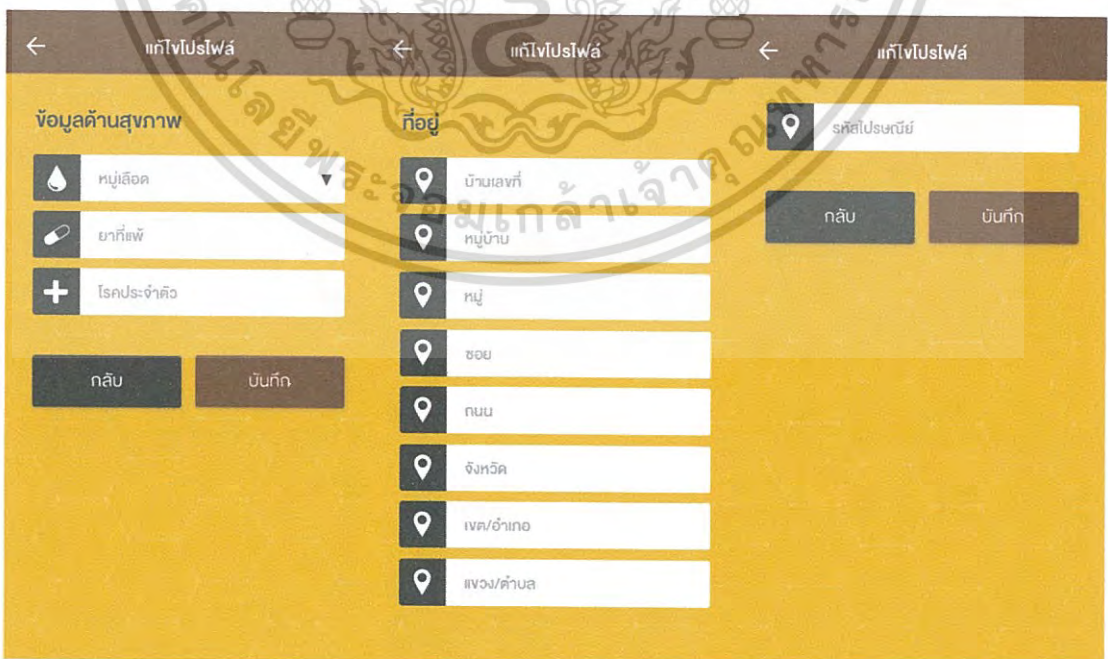
จากภาพที่ 4.12 เป็นหน้าแสดงแผนที่ โดยคณะผู้จัดทำจะแยกสถานที่สำคัญเป็นแท็บอยู่ด้านบน เพื่อให้ง่ายต่อการใช้งาน ได้แก่ อยู่ช่อมรถ และสถานพยาบาล โดยสถานพยาบาลจะมีสองประเภทคือ โรงพยาบาลในเครือ และโรงพยาบาลนอกเครือ เมื่อผู้ใช้งานกดที่พินสถานที่ (Location Pin) ก็จะสามารถกดนำทางได้จากปุ่มลูกศรที่ลอยขึ้นมา พร้อมทั้งมีปุ่มสำหรับโทรติดต่อสถานพยาบาล อยู่ช่อมรถ หรือบริษัทประกันภัยได้ ได้แก่แท็บของอยู่ช่อมรถ และเอกสารเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ใช้งานกดเข้าไปจะขยับขึ้นหน้าการคัดไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานพยาบาล จะมีชื่อกันหาสถานที่ซึ่งสามารถค้นหาสถานที่ได้จากกูเกิ้ลแมพ (Google Map) แลในภาพที่ 4.13 จะเป็นหน้าส่วนโทรฉุกเฉิน ซึ่งทางบริษัทประกันภัยของให้ทำแยกออกมาไว้เพื่อความสะดวกในกรณีเกิดเหตุฉุกเฉิน สำหรับผู้ที่ต้องการติดต่อเพื่อเรียกรถพยาบาลหรือแจ้งเหตุกับตำรวจ

4.1.3.7. หน้าโปรไฟล์



ภาพที่ 4.14 ตัวอย่างหน้าโปรไฟล์ (1)

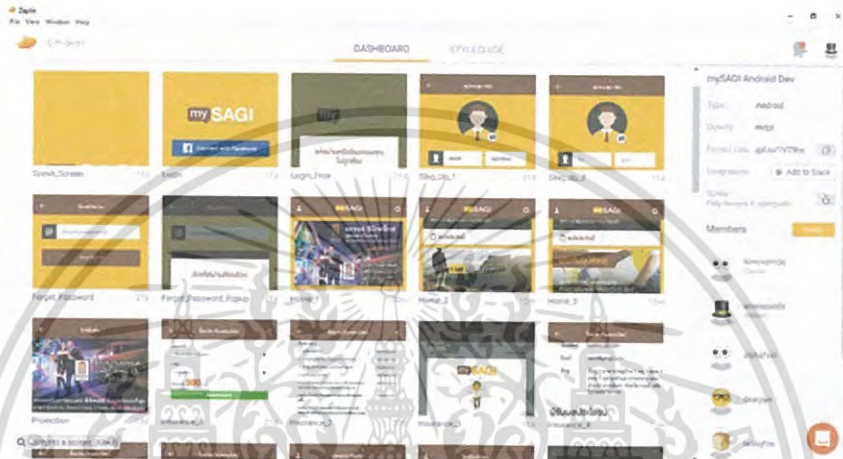


ภาพที่ 4.15 ตัวอย่างหน้าโปรไฟล์ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพที่ 4.14 และภาพที่ 4.15 เป็นหน้าแสดงโปรไฟล์ของผู้ใช้งาน ซึ่งด้านบนผู้ใช้งานจะสามารถเปลี่ยนรูปโปรไฟล์ของตัวเองได้โดยกดที่ไอคอนกล้องถ่ายรูป ถัดลงมาจะมีปุ่มข้อมูลส่วนตัว ปุ่มข้อมูลด้านสุขภาพ และปุ่มที่อยู่ ซึ่งกดเข้าไปจะมีรายละเอียดของผู้ใช้งานตามประเภทที่แยกไว้ โดยข้อมูลต่างๆนี้จะถูกดึงมาจากฐานข้อมูล และผู้ใช้งานสามารถแก้ไขข้อมูลส่วนตัวได้จากแอปพลิเคชัน เมื่อกดปุ่มบันทึกจะมีไดอะล็อกแสดงขึ้นมาถามว่าต้องการจะบันทึกข้อมูลหรือไม่

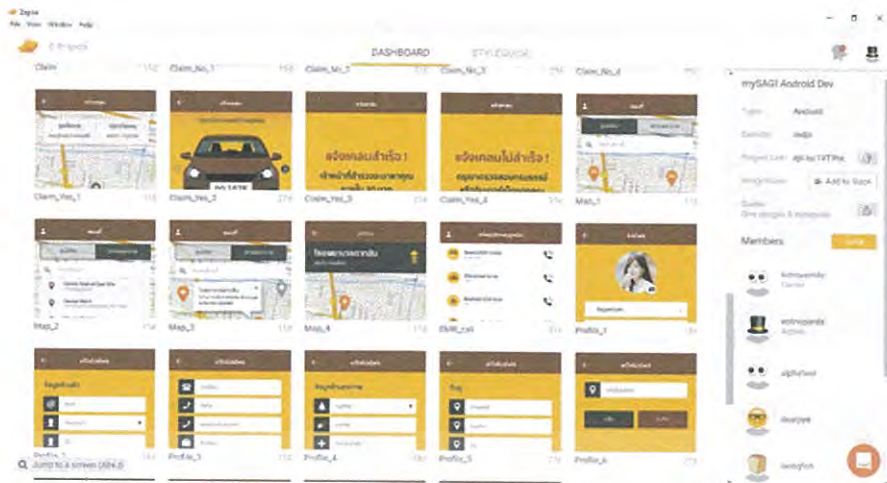
4.1.3.8. หน้าจอทั้งหมดหลังอัปโหลดไฟล์ขึ้นไปบนโปรแกรม Zeplin



ภาพที่ 4.16 ตัวอย่างแบบจำลองแอปพลิเคชันทั้งหมดที่ส่งต่อให้นักพัฒนาบนโปรแกรม Zeplin (1)



ภาพที่ 4.17 ตัวอย่างแบบจำลองแอปพลิเคชันทั้งหมดที่ส่งต่อให้นักพัฒนาบนโปรแกรม Zeplin (2)



ภาพที่ 4.18 ตัวอย่างแบบจำลองแอปพลิเคชันทั้งหมดที่ส่งต่อให้นักพัฒนาโปรแกรม Zeplin (3)

4.1.4. ผลของการทำทดสอบประสบการณ์การใช้งานแอปพลิเคชันของผู้ใช้งาน

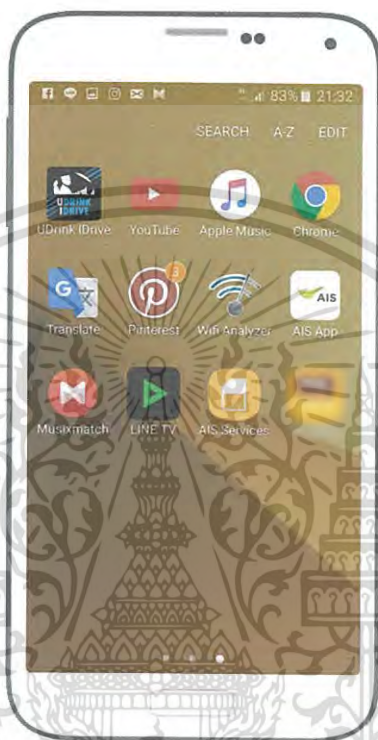
ผลของการทำทดสอบประสบการณ์การใช้งานแอปพลิเคชัน (Usability Test) เป็นไปด้วยดี ลูกค้าที่เข้ามาร่วมทดสอบ 90 เปอร์เซ็นต์ สามารถใช้งานแอปพลิเคชันได้อย่างราบรื่นตามที่ทีมดีไซน์ได้ออกแบบไว้ จะมีเพียงส่วนน้อยประมาณ 10 เปอร์เซ็นต์ เท่านั้นที่ใช้งานแอปพลิเคชันได้อย่างไม่ราบรื่นเนื่องจากประสบการณ์การใช้งานแอปพลิเคชันของกลุ่มผู้ใช้งานใน 10 เปอร์เซ็นต์นี้มีน้อย อาจเป็นเพราะเป็นกลุ่มของผู้ใช้ที่มีอายุราวๆ 50 ปีขึ้นไปหรือเป็นกลุ่มของผู้สูงอายุ

บทที่ 5

ผลการวิจัย (ส่วนการพัฒนาแอปพลิเคชัน)

5.1. การใช้งานแอปพลิเคชัน

ผู้ใช้งานสามารถเข้าสู่แอปพลิเคชัน ผ่านทางระบบปฏิบัติการแอนดรอยด์ หลังการดาวน์โหลดและติดตั้ง แอปพลิเคชันแล้ว เลือกที่ไอคอนดังแสดงในภาพที่ 5.1



ภาพที่ 5.1 แสดงไอคอนแอปพลิเคชันบนจอโทรศัพท์เคลื่อนที่

หน้าแรกของแอปพลิเคชัน จะอธิบายการใช้งานของแอปพลิเคชัน ส่วนข้างล่างของหน้าจอโทรศัพท์เคลื่อนที่ จะแสดงปุ่ม skip ซึ่งการทำงานจะเหมือนกันในทุกหน้าของการอธิบายวิธีใช้งานของแอปพลิเคชัน ดังแสดงในภาพที่ 5.2 มีดังนี้คือ

- ปุ่ม skip

ใช้สำหรับการเปลี่ยนไปหน้าถัดไป ผู้ที่ใช้งานเป็นครั้งแรกสามารถอ่านวิธีใช้งานของ แอปพลิเคชัน ได้ทีละหน้า ในหน้าสุดท้ายปุ่ม skip ใช้สำหรับการข้ามวิธีใช้งานของแอปพลิเคชันทั้งหมด เพื่อก้าวสู่การใช้งานแอปพลิเคชันจริง



ภาพที่ 5.2 ภาพแสดงวิธีการใช้งานแอปพลิเคชัน

ส่วนหน้าจอแสดงช่องกรอกข้อความ 2 ช่องและปุ่ม 4 ปุ่ม ดังแสดงในภาพที่ 5.3 ได้แก่

- ช่องกรอกข้อความ ใช้สำหรับใส่อีเมลหรือเบอร์โทรศัพท์
- ช่องกรอกข้อความ สำหรับใส่รหัสผ่าน
- ปุ่ม Login with Facebook ใช้สำหรับเข้าสู่ระบบผ่านเฟซบุ๊ก
- ปุ่ม ลืมรหัสผ่าน ใช้สำหรับส่งรหัสผ่านมาทางอีเมลหรือเบอร์โทรศัพท์เมื่อลืมรหัสผ่าน
- ปุ่ม สมัครสมาชิก ใช้สำหรับกรอกข้อมูลเพื่อสมัครสมาชิก
- ปุ่ม เข้าสู่ระบบ ใช้สำหรับการเข้าใช้งานโดยผู้ใช้งานได้สมัครสมาชิกแล้ว



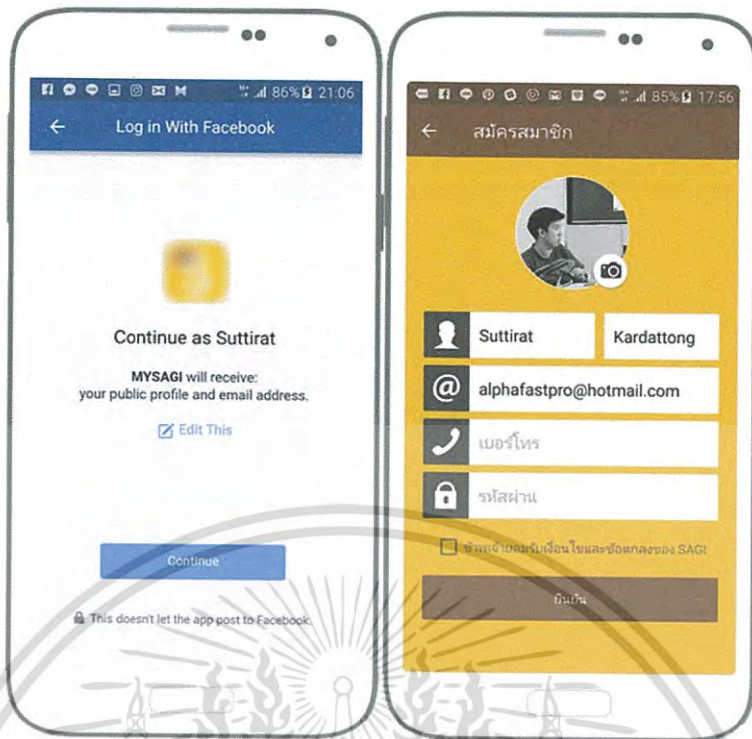
ภาพที่ 5.3 ภาพแสดงการเข้าสู่ระบบ

เลือกปุ่ม Login with Facebook แสดงหน้าสำหรับยืนยันการเข้าถึงข้อมูล ดังภาพที่ 5.4 (ซ้าย) โดยกดปุ่ม continue เพื่อให้ระบบดึงข้อมูลจากเฟซบุ๊กมา

จากนั้นกรอกข้อมูลที่เหลือเพื่อทำการสมัครสมาชิก ข้อมูลที่ต้องกรอก ได้แก่

- เบอร์โทร
- รหัสผ่าน คือรหัสผ่านสำหรับเข้าใช้งาน
- เลือก ยอมรับเงื่อนไขและข้อตกลง

เมื่อกรอกข้อมูลเรียบร้อยแล้ว เลือกปุ่ม ยืนยัน เพื่อทำการบันทึกข้อมูล



ภาพที่ 5.4 ภาพแสดงการเข้าสู่ระบบผ่านเฟซบุ๊ก

เลือกปุ่ม ลืมรหัสผ่าน แสดงหน้าสำหรับผู้ลืมรหัสผ่าน ดังภาพที่ 5.5 โดยข้อมูลที่กรอก คือ อีเมลหรือเบอร์โทรศัพท์ จากนั้นกดปุ่มส่งรหัส เพื่อให้ระบบส่งรหัสมาทางอีเมลหรือเบอร์ที่ผู้ใช้กรอกไป



ภาพที่ 5.5 เลือกปุ่มลืมรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกปุ่ม สมาชิก หน้าจอจะแสดงให้ผู้ใช้กรอกข้อมูล ดังภาพที่ 5.6 โดยข้อมูลที่ผู้ใช้กรอกกรอกรมีดังนี้

- ชื่อ คือ ชื่อจริงของผู้ใช้งาน โดยชื่อที่กรอกจะนำไปแสดงบนบัตรจำลอง
- สกุล คือ นามสกุลของผู้ใช้งาน โดยนามสกุลที่กรอกจะนำไปแสดงบนบัตรจำลอง
- อีเมล คือ อีเมลสำหรับเข้าใช้งาน
- เบอร์โทรศัพท์ คือ เบอร์โทรศัพท์สำหรับเข้าใช้งาน
- รหัสผ่าน คือ รหัสผ่านสำหรับเข้าใช้งาน

เมื่อกรอกข้อมูลเรียบร้อยแล้ว เลือกปุ่ม ยืนยัน เพื่อทำการบันทึกข้อมูล



ภาพที่ 5.6 แสดงหน้าจอสำหรับสมัครสมาชิก



ภาพที่ 5.7 แสดงหน้าหลัก

เมื่อผู้ใช้เข้าสู่ระบบเรียบร้อยแล้ว จะแสดงหน้าหลัก ดังภาพที่ 5.7 หน้าจอส่วนบน ประกอบไปด้วย 2 ปุ่มได้แก่

- ปุ่มโปรไฟล์ ใช้สำหรับบันทึกข้อมูลส่วนตัวผู้ใช้ สามารถแก้ไขข้อมูลได้ ดังภาพที่ 5.8
- ปุ่มตั้งค่า ใช้สำหรับตั้งค่า การเปิดปิดแจ้งเตือน การเชื่อมต่อเฟซบุ๊ก แสดงความคิดเห็น นโยบายความเป็นส่วนตัว แก้ไขรหัสผ่าน และออกจากระบบ ดังภาพที่ 5.11

ส่วนกลางหน้าจอจะประกอบไปด้วยแสดงโปรโมชั่น และข้อมูลประกันภัย เมื่อผู้ใช้กดสนใจประกันนี้ หน้าจอจะแสดงดังภาพที่ 5.13

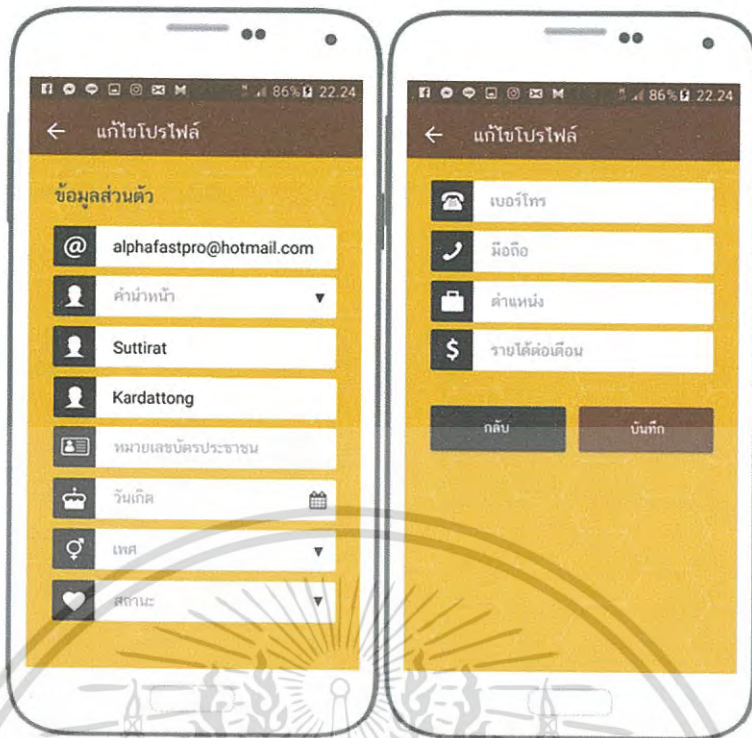
ส่วนล่างของหน้าจอแสดงปุ่ม 5 ปุ่ม ดังนี้

- ปุ่ม หน้าหลัก ใช้สำหรับเชื่อมโยงไปยังหน้าหลัก
- ปุ่ม บัตรประกันภัย ใช้สำหรับเชื่อมโยงไปยังหน้าบัตรจำลอง
- ปุ่ม เคลม ใช้สำหรับเชื่อมโยงไปยังหน้าเคลม
- ปุ่ม แผนที่ ใช้สำหรับเชื่อมโยงไปยังหน้าแผนที่
- ปุ่ม โทรณุกเงิน ใช้สำหรับเชื่อมโยงไปยังหน้าโทรณุกเงิน



ภาพที่ 5.8 แสดงหน้าโปรไฟล์

เลือก ข้อมูลส่วนตัว จะแสดงหน้าแก้ไขโปรไฟล์ ซึ่งมีกล่องข้อความและปุ่ม ดังภาพที่ 5.9 ผู้ใช้จะต้องกรอกข้อมูลส่วนตัว เช่น อีเมล ชื่อ สกุล เลขบัตรประชาชน วันเกิด เพศ มือถือ เป็นต้น เมื่อกดปุ่ม บันทึก หน้าจอจะแสดงดังภาพที่ 5.10 จากนั้นกดปุ่ม ตกลง เพื่อบันทึกข้อมูล



ภาพที่ 5.9 หน้าแก้ไขโปรไฟล์ข้อมูลส่วนตัว



ภาพที่ 5.10 แสดงการแจ้งเตือน เมื่อกดปุ่มบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.11 แสดงหน้าตั้งค่า

- เมื่อผู้ใช้กดปุ่ม แสดงความคิดเห็น หน้าจอจะแสดงดังภาพที่ 5.12 หน้าจอประกอบด้วย
- กล่องข้อความ สำหรับใส่หัวข้อปัญหาการใช้งาน
 - ก่อขข้อความ สำหรับพิมพ์แสดงความคิดเห็น
 - ปุ่ม ยกเลิก คือ ยกเลิกข้อความที่พิมพ์
 - ปุ่ม ส่ง คือ จะส่งข้อความที่พิมพ์ไปยังบริษัท



ภาพที่ 5.12 แสดงหน้าแสดงความคิดเห็น



ภาพที่ 5.13 หน้าชื่อประกันออนไลน์

ส่วนกลางหน้าจอก็จะเป็นการนำเว็บวิมาแสดง เป็นการซื้อประกันออนไลน์ผ่านเว็บวิ
ส่วนล่างของหน้าจอประกอบไปด้วยปุ่ม 2 ปุ่ม ได้แก่
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่ม ติดต่อเจ้าหน้าที่ คือ ผู้ใช้กดเพื่อโทรหาเจ้าหน้าที่
- ปุ่ม ขอข้อมูลเพิ่มเติม คือ ผู้ใช้กดเมื่อต้องการสอบถามข้อมูล รายละเอียดเพิ่มเติม โดยเจ้าหน้าที่จะเป็นฝ่ายติดต่อกลับมา



ภาพที่ 5.14 หน้ายืนยันตัวตน กรณีใช้งานบัตรครั้งแรก

เลือกปุ่ม บัตรประกัน เมื่อเข้าใช้งานหน้านี้ครั้งแรกจะแสดงหน้ายืนยันตัวตน ดังภาพที่ 5.14 ข้อมูลที่ผู้ใช้ต้องกรอก คือ หมายเลขบัตรประชาชน จากนั้นกดปุ่ม ตรวจสอบ เพื่อส่งข้อมูลให้ระบบตรวจสอบ

กรณีที่ผู้ใช้ยังไม่มีประกันภัย จะแสดงดังภาพที่ 5.15 หรือหมายเลขบัตรประชาชนไม่ถูกต้อง จะแสดงกล่องข้อความแจ้งเตือนไม่พบหมายเลขบัตรประชาชน ดังภาพที่ 5.16 ให้ผู้ใช้ตรวจสอบข้อมูล โดยมีปุ่มให้เลือก ดังนี้

- ปุ่ม ลองใหม่อีกครั้ง ให้ผู้ใช้กลับไปยังหน้ายืนยันแล้วกรอกข้อมูลใหม่อีกครั้ง
- ปุ่ม ติดต่อเจ้าหน้าที่ ใช้สำหรับโทรติดต่อเจ้าหน้าที่

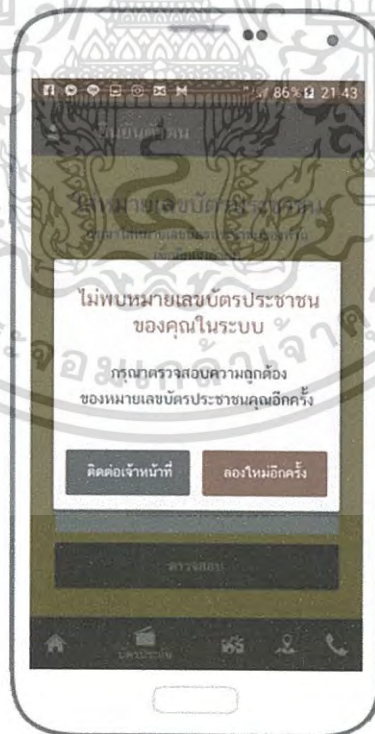
เมื่อผู้ยืนยันตัวตนสำเร็จและมีประกันภัยของบริษัทแล้ว จะเข้าสู่หน้า บัตรประกันของฉันทดงภาพที่ 5.17 เป็นปุ่มแสดงข้อมูลเล่มกรมธรรม์ที่ลูกค้าถืออยู่ ซึ่งปุ่มถูกแบ่งตามประเภทประกันได้แก่

- ปุ่ม ประกันภัยรถยนต์
- ปุ่ม ประกันอุบัติเหตุ
- ปุ่ม ประกันสุขภาพ

เมื่อผู้ใช้กดปุ่มประกันภัยรถยนต์ หน้าจอจะแสดงดังภาพที่ 5.18



ภาพที่ 5.15 หน้าบัตรประกัน กรณียังไม่มีประกัน



ภาพที่ 5.16 แสดงการแจ้งเตือนเมื่อใส่หมายเลขบัตรประชาชนผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.17 หน้าบัตรประกันของฉัน



ภาพที่ 5.18 หน้าบัตรประกัน

เลือกปุ่ม เกล็ด จะเข้าสู่หน้า เลือกลักษณะอุบัติเหตุ ดังภาพที่ 5.19 เพื่อให้ผู้ใช้เลือกตาม
 กรณีของตนเอง ได้แก่
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีคู่กรณี คือ พนักงานจะต้องมาตรวจสอบ ณ ที่เกิดเหตุ
- ไม่มีคู่กรณี คือ พนักงานไม่ต้องมาตรวจสอบ ณ ที่เกิดเหตุ ผู้ขับขี่สามารถนำรถประกันเข้าไปติดต่อซ่อมที่อู่ในเครือได้
- ปุ่ม โทรขอความช่วยเหลือ ไว้ใช้สำหรับติดต่อเจ้าหน้าที่ทันที

กรณีผู้ใช้ที่ยังไม่มีประกัน เมื่อกดปุ่ม เคลม หน้าจะแสดงดังภาพที่ 5.20 ผู้ใช้สามารถกดปุ่ม มองหาประกันภัยรถยนต์ เพื่อไปยังหน้าซื้อประกันออนไลน์

เมื่อผู้ใช้มีประกัน หลังจากที่เกิดปุ่มเลือก มีคู่กรณีหรือไม่มีคู่กรณี หน้าจะแสดงดังภาพที่ 5.21 ซึ่งประกอบไปด้วย

- ข้อมูลทะเบียนรถยนต์ตามเล่มกรมธรรม์ที่ลูกค้ามี
- ชื่อคนขับพร้อมเบอร์โทร
- กล่องข้อความ ให้กรอกชื่อผู้แจ้งเคลม
- ปุ่ม แจ้งเคลม กดเพื่อไปยังขั้นตอนต่อไป



ภาพที่ 5.19 แสดงหน้าเลือกลักษณะอุบัติเหตุ



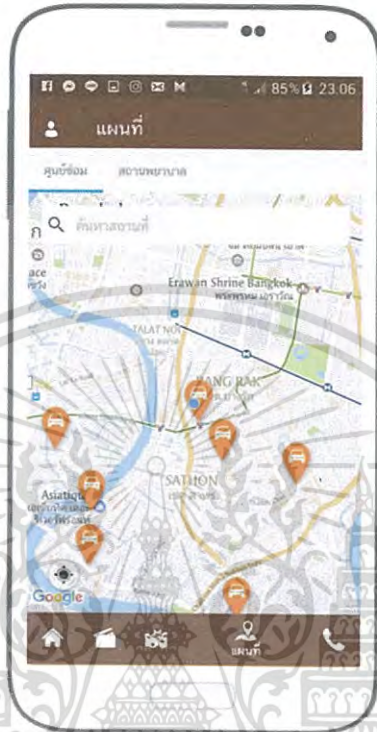
ภาพที่ 5.20 แสดงหน้าแจ้งเตือน กรณียังไม่มีประกัน



ภาพที่ 5.21 หน้าแจ้งเคลม

เลือกปุ่ม แผนที่ หน้าจอจะแสดงดังภาพที่ 5.22 ไว้สำหรับค้นหาสถานที่ศูนย์ซ่อมรถ หรือสถานพยาบาล ภายใ้หน้าจอประกอบไปด้วย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

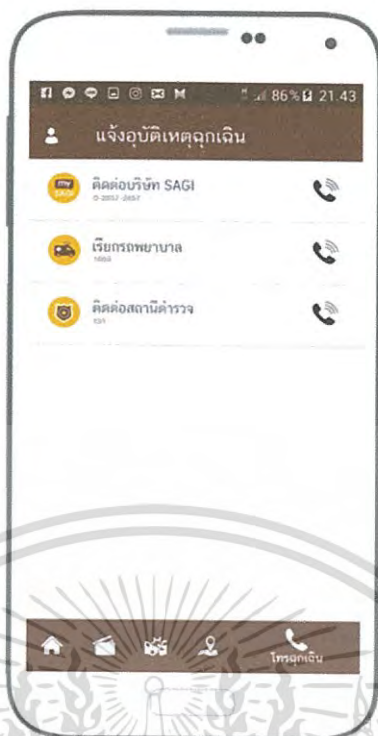
- แถบเลือกระหว่างศูนย์ซ่อม หรือสถานพยาบาล
- กล่องข้อความ สำหรับระบุสถานที่ ที่ต้องการค้นหา
- แผนที่ แสดงสถานที่ศูนย์ซ่อม หรือสถานพยาบาลที่มีในฐานข้อมูล



ภาพที่ 5.22 หน้าแผนที่

เลือกปุ่ม โทรฉุกเฉิน หน้าจอจะแสดง ดังภาพที่ 5.23 เป็นปุ่มให้ผู้ใช้กดเพื่อโทรออกได้ทันทีประกอบไปด้วย ปุ่มดังนี้

- ปุ่ม ติดต่อบริษัท คือ จะโทรไปยังเบอร์ของบริษัท
- ปุ่ม เรียกรถพยาบาล คือ โทรไปยังเบอร์ 1669
- ปุ่ม ติดต่อสถานีตำรวจ คือ โทรไปยังเบอร์ 191



ภาพที่ 5.23 แสดงหน้าโทรฉุกเฉิน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

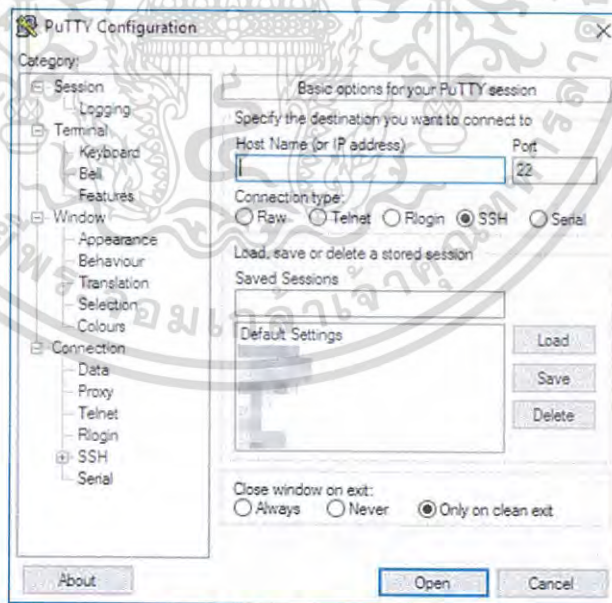
ผลการวิจัย (ส่วนการพัฒนาระบบหลังบ้านและส่วนต่อประสานโปรแกรมประยุกต์)

จากการดำเนินงานตลอดจนสิ้นสุดระยะเวลาโครงการสหกิจศึกษา ผลการวิจัยที่ได้สามารถแสดงให้เห็นถึงผลลัพธ์ที่เกิดขึ้นระหว่างขั้นตอนการพัฒนาผลลัพธ์การทำงานของระบบหลังบ้านและส่วนต่อประสานโปรแกรมประยุกต์ (API) ที่ได้พัฒนาขึ้น เพื่อทำงานร่วมกับ แอปพลิเคชันแอนดรอยด์ สำหรับให้บริการระบบประกันบนโมบายล์แพลตฟอร์ม โดยผลการดำเนินงานจะแสดงให้เห็นถึงการพัฒนาระบบหลังบ้าน และการทำงานของ API ในการรับการร้องขอ (Request) และการส่งการตอบกลับ (Response) การร้องขอของแต่ละแบบในแต่ละ API โดยใช้การจำลองส่ง Request ไปยังระบบหลังบ้านบนเครื่องแม่ข่าย (Server) และแสดงผล Response ในรูปของ JSON ด้วยโปรแกรม Postman และเปรียบเทียบให้เห็นถึงรูปแบบการตอบกลับการร้องขอที่ถูกต้อง และการตอบกลับการร้องขอที่ผิดพลาด

6.1. ผลการดำเนินงานในขั้นตอนการพัฒนา

6.1.1. ขึ้นโครงสร้างด้วย าลาเวลเฟรมเวิร์ค

การขึ้นโครงสร้างจะใช้วิธีการพิมพ์คำสั่งผ่านหน้าต่างคอมมาต์ไลน์ (command line) โดยทำการ รีโมทซีเคียวซิล (Remote Secure Shell : Remote SSH) ผ่านโปรแกรม Putty ดังภาพที่ 6.1 ไปยังเครื่องแม่ข่ายและพิมพ์คำสั่งผ่านหน้าต่างคอมโซล (Console) ของโปรแกรม Putty ดังภาพที่ 6.2



ภาพที่ 6.1 หน้าต่างโปรแกรม Putty

```
@ubuntu-512mb-sgpl-01: ~
login as:
root@188.166. .... 's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-53-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

*** System restart required ***
Last login: Fri Dec 16 04:52:55 2016 from 49.2
root@ubuntu-512mb-sgpl-01:~#
root@ubuntu-512mb-sgpl-01:~#
```

ภาพที่ 6.2 หน้าต่าง console เมื่อ remote SSH ไปยังเครื่องแม่ข่าย

ใช้คำสั่ง `composer` ในการสร้างโปรเจกต์ด้วยลาลาเวลเฟรมเวิร์ค “`composer create-project --prefer-dist laravel/laravel your_project_name`” ดังภาพที่ 6.3

```
@ubuntu-server: /var/www/html
@ubuntu-server: /var/www/html# composer create-project --prefer-dist laravel/laravel insurance api
Installing laravel/laravel (v5.3.16)
- Installing laravel/laravel (v5.3.16)
  Loading from cache
Created project in insurance api
> php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing vlucas/phpdotenv (v2.4.0)
  Loading from cache
- Installing symfony/polyfill-mbstring (v1.3.0)
  Loading from cache
- Installing symfony/var-dumper (v3.1.8)
  Loading from cache
- Installing symfony/translation (v3.1.8)
  Loading from cache
- Installing symfony/routing (v3.1.8)
  Loading from cache
```

ภาพที่ 6.3 หน้าต่าง console ขณะสร้างโปรเจกต์ลาลาเวลด้วยคำสั่ง `composer`

หลังจากสร้างโปรเจกต์เสร็จจะได้รับรหัส Application key เพื่อเอาไว้ใช้เข้ารหัสการทำงานบนลาลาเวลเฟรมเวิร์คซึ่งจะต้องจำรหัสนี้ไว้เพื่อใช้ในการกำหนดค่า key ในไฟล์ `app.php` ดังภาพที่ 6.4

```

root@ubuntu-server: /var/www/html
laravel/framework suggests installing aws/aws-sdk-php (Required to use the SQS queue driver and SES mail driver (~3.0).)
laravel/framework suggests installing doctrine/dbal (Required to rebase columns and drop SQLite columns (~2.4).)
laravel/framework suggests installing guzzlehttp/guzzle (Required to use the Mailgun and Mandrill mail drivers and the ping m
ethods on schedules (~5.3)-6.0.)
laravel/framework suggests installing league/flysystem-aws-s3-v3 (Required to use the Flysystem S3 driver (~1.0).)
laravel/framework suggests installing league/flysystem-rackspace (Required to use the Flysystem Rackspace driver (~1.0).)
laravel/framework suggests installing pda/phpstan (Required to use the beanstalk queue driver (~3.0).)
laravel/framework suggests installing predis/predis (Required to use the redis cache and queue drivers (~1.0).)
laravel/framework suggests installing pusher/pusher-php-server (Required to use the Pusher broadcast driver (~2.0).)
laravel/framework suggests installing symfony/psr-http-message-bridge (Required to use psr7 bridging features (0.2.*).)
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit-mock-objects suggests installing ext-soap (*)
phpunit/php-code-coverage suggests installing ext-xdebug (>=2.4.0)
phpunit/phpunit suggests installing phpunit/php-invoker (~1.1)
phpunit/phpunit suggests installing ext-xdebug (*)
Writing lock file
Generating autoload files
> Illuminate\Foundation\ComposerScripts::postUpdate
> php artisan optimize
Generating optimized class loader
The compiled class file has been removed.
> php artisan key:generate
Application key [base64:wj0112oYKQp...KzE=] set successfully.
root@ubuntu-server: /var/www/html#

```

ภาพที่ 6.4 หน้าต่าง console หลังจากสร้างโปรเจกต์ลาลาเวลด้วยคำสั่ง composer เสร็จสมบูรณ์

ใช้คำสั่ง `cd` เข้าไปที่โฟลเดอร์ที่เก็บข้อมูลโปรเจกต์ “`cd /var/www/html/your_project_name/config`” จากนั้นเปิดไฟล์ `app.php` เพื่อเพิ่ม Application key ที่ได้ ด้วยคำสั่ง “`vim app.php`” ดังภาพที่ 6.5

```

root@ubuntu-server: /var/www/html/insurance_api/config
@ubuntu-server:/var/www/html# cd insurance_api/config
@ubuntu-server:/var/www/html/insurance_api/config# vim app.php

```

ภาพที่ 6.5 หน้าต่าง console ขณะใช้คำสั่ง `vim` เพื่อเปิดไฟล์ด้วย `vim` editor

เลื่อนลงเพื่อค้นหาและแก้ไขบรรทัด `'key' => env('APP_KEY')` โดยกด `|` และเพิ่ม Application key ที่ได้ลงไปเป็น `'key' => env('APP_KEY', 'putYourApplicationKey')` จากนั้นกด `ESC` พิมพ์ `:wq` เพื่อบันทึกไฟล์และออกจากโปรแกรม `vim`

6.1.2. เชื่อมต่อระบบเข้ากับฐานข้อมูลบนเครื่องแม่ข่าย

การทำงานของระบบจะต้องมีการทำงานควบคู่กับฐานข้อมูลด้วย ซึ่งชนิดของฐานข้อมูลก็ขึ้นอยู่กับระบบฐานข้อมูลเก่าของลูกค้าบริษัทประกันว่า เดิมแล้วใช้ฐานข้อมูลชนิดใด ในครั้งนี้บริษัทเอกชนผู้ว่าจ้างในการทำระบบประกันบนโมบายล์แพลตฟอร์มนั้น ใช้ระบบฐานข้อมูล PostgreSQL ซึ่งสามารถทำงานร่วมกับ ลาลาเวลเฟรมเวิร์คได้

เริ่มจากการแก้ไขการตั้งค่าในไฟล์ `.env` ของลาลาเวลเฟรมเวิร์ค โดยการใช้คำสั่ง `cd /var/www/html/your_project_name` ดังภาพที่ 6.6

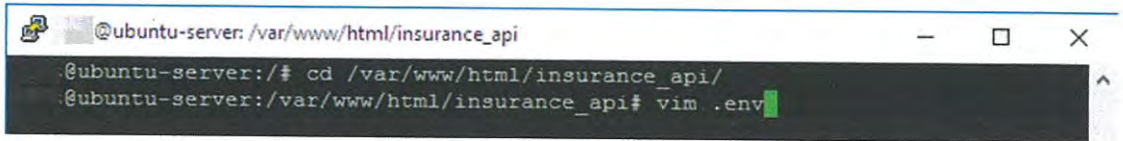
```

@ubuntu-server:/var/www/html/insurance_api
@ubuntu-server:/# cd /var/www/html/insurance_api/
@ubuntu-server:/var/www/html/insurance_api#

```

ภาพที่ 6.6 หน้าต่าง console ขณะใช้คำสั่ง `cd` เพื่อเข้าไปภายในโฟลเดอร์โปรเจกต์

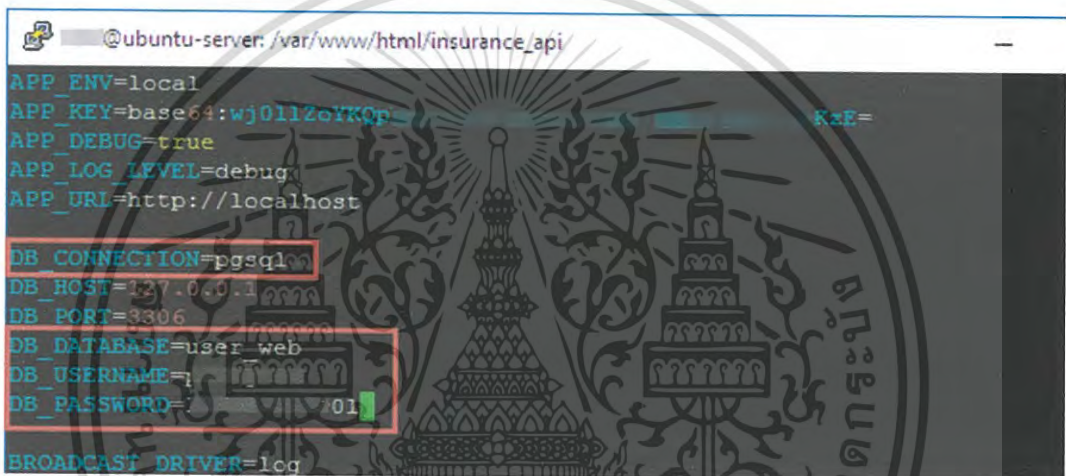
เอกสารนี้เป็นเอกสารหลังจากนั้นใช้คำสั่ง `vim .env` เพื่อเปิดไฟล์ `.env` ด้วย `vim` editor ดังภาพที่ 6.7 นี้ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และส่งอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
@ubuntu-server: /var/www/html/insurance_api
@ubuntu-server:~# cd /var/www/html/insurance_api/
@ubuntu-server:/var/www/html/insurance_api# vim .env
```

ภาพที่ 6.7 หน้าต่าง console ขณะเปิดไฟล์ .env ด้วย vim editor

แก้ไขไฟล์ .env ที่ปิดขึ้นโดยการกด I และแก้ไขข้อความดังนี้ DB_CONNECTION = postgres เพื่อกำหนดชนิดของฐานข้อมูล DB_DATABASE = your_db_name เพื่อกำหนดชื่อดาต้าเบสที่จะเชื่อมต่อ DB_USERNAME = username ใส่ชื่อผู้ใช้และ DB_PASSWORD = password สำหรับรหัสผ่านดังภาพที่ 6.8 จากนั้นกด ESC แล้วพิมพ์ :wq เพื่อบันทึกไฟล์และออกจาก vim editor



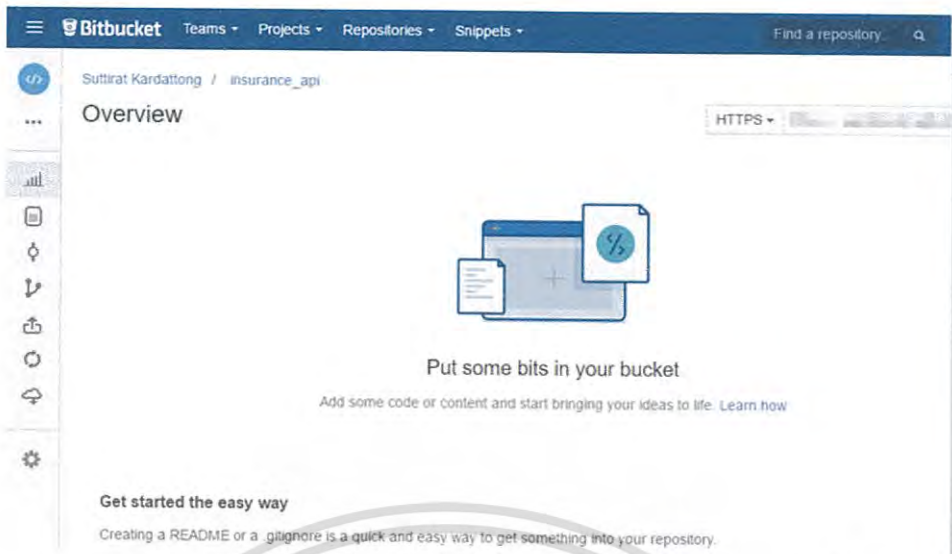
```
@ubuntu-server: /var/www/html/insurance_api
APP_ENV=local
APP_KEY=base64:wj0112oYKCP
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://localhost

DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=user_web
DB_USERNAME=root
DB_PASSWORD=12345678901
BROADCAST_DRIVER=log
```

ภาพที่ 6.8 หน้าต่าง console ขณะแก้ไขไฟล์ .env ด้วย vim editor

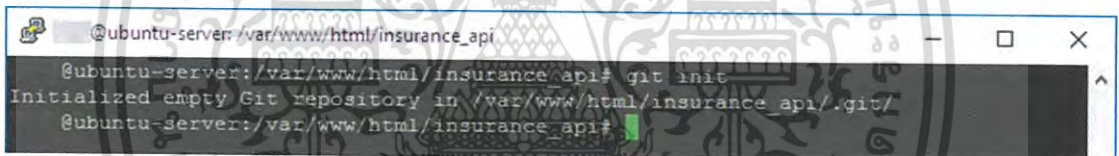
6.1.3. อัปเดตโปรเจกต์ขึ้นระบบ Git

การอัปเดตไฟล์ขึ้นระบบ Git จะต้องมี Git Server ซึ่งบริษัท Octagon Interactive เลือกใช้ Git Server ของ Bitbucket.org โดยเริ่มจากเข้าไปสร้าง Git Repository ไว้สำหรับเก็บไฟล์โปรเจกต์ที่สร้างขึ้นดังภาพที่ 6.9



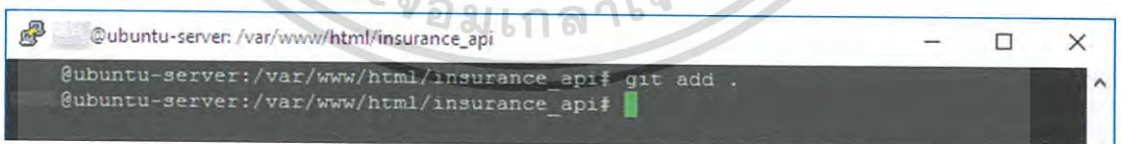
ภาพที่ 6.9 หน้าเว็บไซต์ Bitbucket.org หลังจากสร้าง Repository ใหม่

ที่หน้า console ของโปรแกรม putty ใช้คำสั่ง “cd /var/www/html/your_project_name” กลับไปยังโฟลเดอร์โปรเจกต์ และใช้คำสั่ง git init เพื่อสร้าง Local Repository ไว้ภายในโฟลเดอร์โปรเจกต์ดังภาพที่ 6.10



ภาพที่ 6.10 หน้าต่าง console ขณะสร้าง Local Repository ด้วยคำสั่ง git init

ใช้คำสั่ง git add . เพื่อทำให้ทุกไฟล์ที่อยู่ในโฟลเดอร์ อยู่ในสถานะ stage เพื่อเตรียม commit ไฟล์ทั้งโปรเจกต์ไปยัง Local Repository ดังภาพที่ 6.11



ภาพที่ 6.11 หน้าต่าง console ขณะส่ง Stage ไฟล์ด้วยคำสั่ง git add

ใช้คำสั่ง git commit -m 'put_your_description_here' เพื่อ commit ไฟล์ไปยัง Local Repository ที่สร้างไว้ คำสั่ง -m คือการใส่รายละเอียดเป็นข้อความเพื่อกำกับการ commit ไว้ดังภาพที่ 6.12

```
@ubuntu-server: /var/www/html/insurance_api
@ubuntu-server:/var/www/html/insurance_api# git commit -m 'initial insurance_api on git'
[master (root@ubuntu-server:localdomain) 2e5d4ee] initial insurance_api on git
Committer: root <root@ubuntu-server.localdomain>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

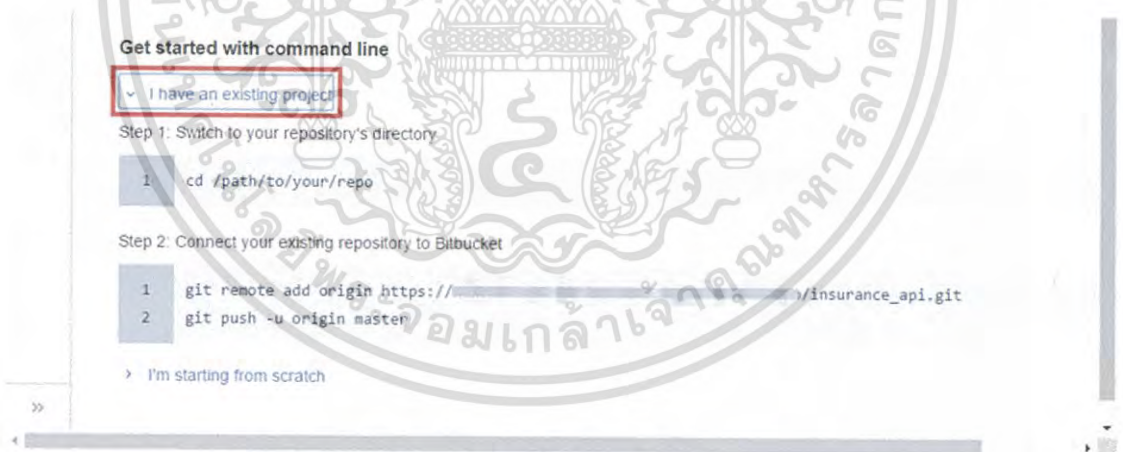
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

81 files changed, 6387 insertions(+)
create mode 100644 .env.example
create mode 100644 .gitattributes
create mode 100644 .gitignore
create mode 100755 app/Console/Kernel.php
create mode 100755 app/Exceptions/Handler.php
create mode 100755 app/Http/Controllers/Auth/ForgotPasswordController.php
create mode 100755 app/Http/Controllers/Auth/LoginController.php
create mode 100755 app/Http/Controllers/Auth/RegisterController.php
```

ภาพที่ 6.12 หน้าต่าง console ขณะใช้คำสั่ง git commit

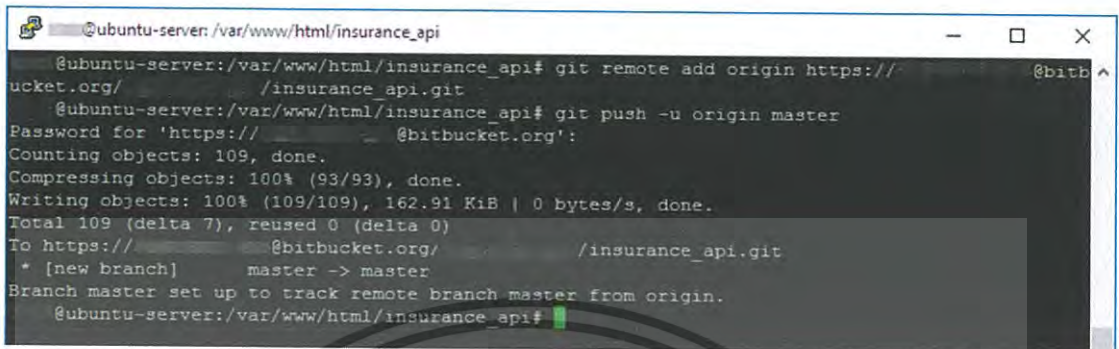
หลังจาก commit ไฟล์บน Local Repository แล้วขั้นตอนต่อไปจะเป็นการเชื่อม Local Repository เข้ากับ Cloud Repository ที่ได้สร้างไว้บน Bitbucket.org บน Bitbucket.org จะมีตัวเลือกในการเชื่อมต่อได้ 2 แบบ คือสำหรับโปรเจกต์ที่มีอยู่แล้ว กับยังไม่เคยมีโปรเจกต์ ให้ใช้ตัวเลือกสำหรับโปรเจกต์ที่มีอยู่แล้วโดยคลิกที่ I have an existing project ดังภาพที่ 6.13



ภาพที่ 6.13 หน้าเว็บไซต์ Bitbucket.org ซึ่งมีขั้นตอนการเชื่อม Bitbucket Repository กับ Local Repository อธิบายไว้

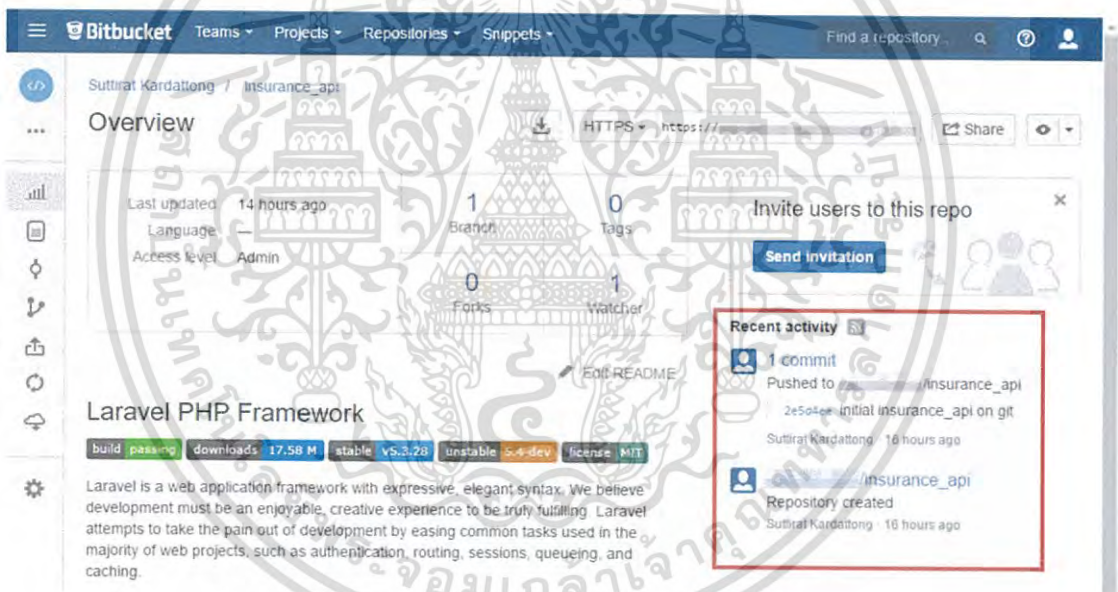
เมื่อคลิกจะปรากฏขั้นตอนการเชื่อมต่อ Bitbucket Repository ขึ้นมาให้พิมพ์คำสั่งเหล่านั้นบนหน้าต่าง console โปรแกรม Putty เริ่มจากพิมพ์คำสั่ง “cd /var/www/html/your_project_name” เพื่อเข้าไปยังโฟลเดอร์โปรเจกต์ จากนั้นพิมพ์คำสั่ง git remote add origin https://your_bitbucket/repository_name.git เพื่อเชื่อม Bitbucket Repository เข้ากับ Local Repository และคำสั่ง git push -u origin master เพื่อ push โปรเจกต์ที่ได้ commit ไว้ในตอนแรกขึ้นไปยัง Bitbucket.org โดยคำสั่ง origin master เป็นการกำหนด branch ที่ต้องการจะ

push โปรเจกต์ขึ้นไปที master จากนั้นที่หน้าต่าง console จะปรากฏข้อความให้ใส่รหัสเพื่อยืนยันตัวตน เมื่อใส่รหัสที่ถูกต้องแล้วจะทำการอัปโหลดดังภาพที่ 6.14 ไฟล์โปรเจกต์จะถูกอัปโหลดขึ้นบน Bitbucket.org ทันที ดังภาพที่ 6.15



```
@ubuntu-server: /var/www/html/insurance_api
@ubuntu-server: /var/www/html/insurance_api# git remote add origin https://bitbucket.org/insurance_api.git
@ubuntu-server: /var/www/html/insurance_api# git push -u origin master
Password for 'https://bitbucket.org':
Counting objects: 109, done.
Compressing objects: 100% (93/93), done.
Writing objects: 100% (109/109), 162.91 KiB | 0 bytes/s, done.
Total 109 (delta 7), reused 0 (delta 0)
To https://bitbucket.org/insurance_api.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
@ubuntu-server: /var/www/html/insurance_api#
```

ภาพที่ 6.14 ภาพหน้าต่าง console หลังจากการเชื่อม Bitbucket Repository กับ Local Repository ด้วยคำสั่ง git remote และทำการอัปโหลดโปรเจกต์ด้วยคำสั่ง git push

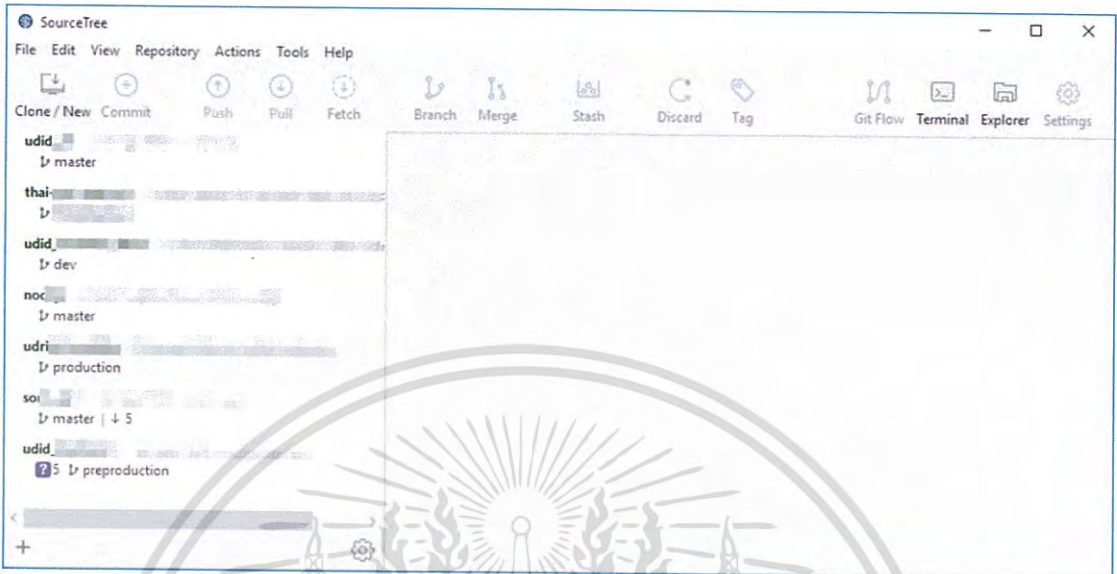


ภาพที่ 6.15 หน้าเว็บไซต์ Bitbucket.org ปรากฏกิจกรรมที่ได้ทำที่แท็บ Recent Activity ในด้านขวา และรายละเอียดคร่าวๆของโปรเจกต์ที่อัปโหลดขึ้นไปบน Bitbucket Git Server

หลังจากการขึ้นโครงโปรเจกต์บนเครื่องแม่ข่ายและอัปโหลดขึ้นระบบ Git แล้วนั้น ในขั้นตอนนี้จะเป็นขั้นตอนสำหรับเตรียมการพัฒนาระบบบนเครื่องคอมพิวเตอร์ทั่วไป เพื่อให้เกิดความรวดเร็วในการพัฒนาและหลีกเลี่ยงความล่าช้าในการทดสอบการทำงานขั้นพื้นฐานขณะพัฒนาจากการอัปโหลดระบบขึ้นไปทดสอบบนเครื่องแม่ข่าย

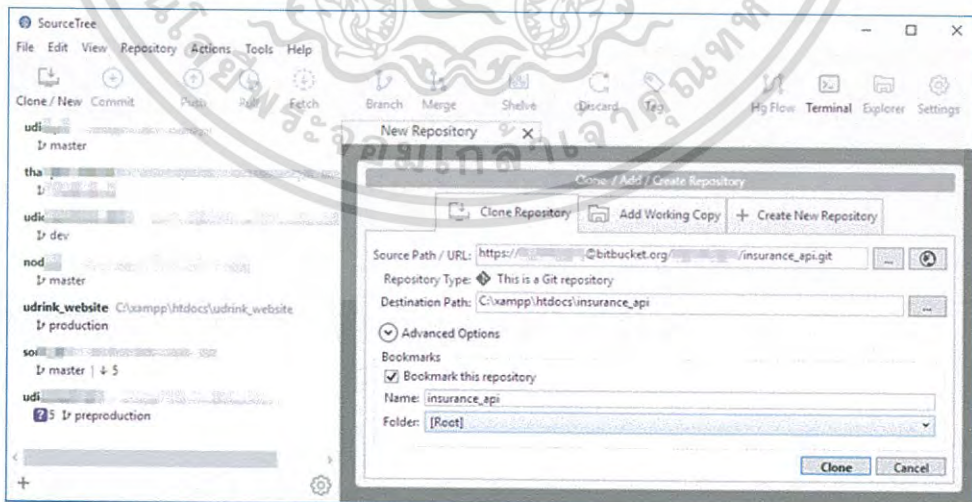
การพัฒนาบนคอมพิวเตอร์ทั่วไปนั้น ก็จำเป็นที่จะต้องจองลงสภาพแวดล้อมให้เหมาะสมต่อการพัฒนาเช่นกันโดยการใช้โปรแกรมต่างๆเข้ามาช่วย เช่น การจำลองการทำงานของ Web Server ด้วย โปรแกรม XAMPP และการจำลองสภาพแวดล้อมอื่นๆ ก่อนทำการพัฒนา

การโคลนโปรเจกต์จาก Git ได้เลือกใช้โปรแกรม Source Tree ดังภาพที่ 6.16 ซึ่งเป็นโปรแกรมประเภท Git Client ที่ได้รับความนิยมสูง



ภาพที่ 6.16 หน้าต่างโปรแกรม Source Tree

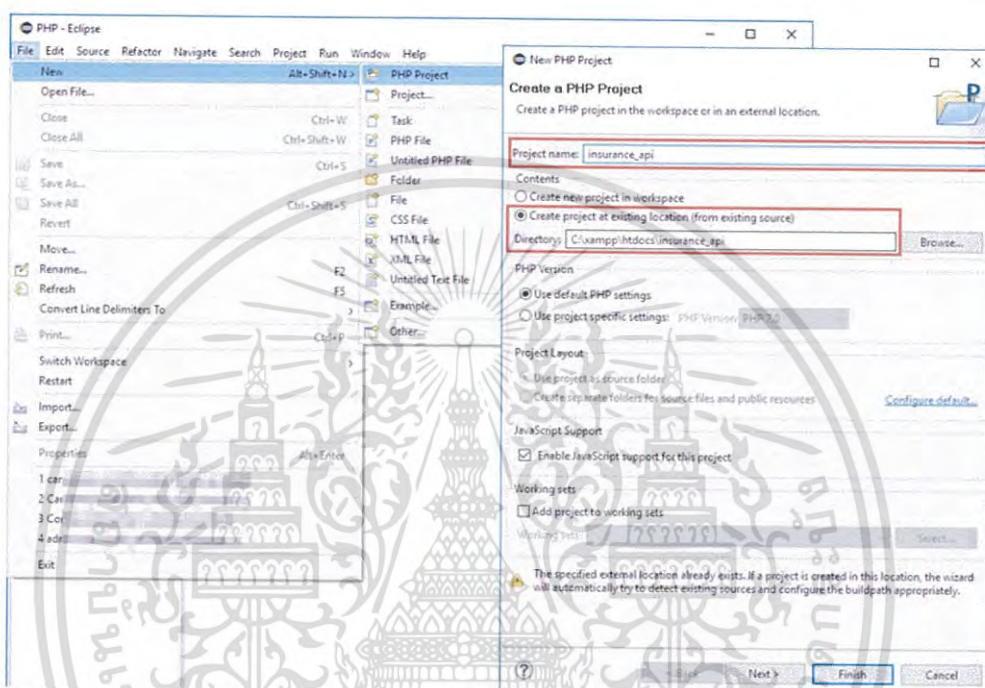
จากนั้นกดเลือกที่ Clone/New เพื่อโคลนโปรเจกต์จาก Bitbucket.org ใส่ URL ของ Bitbucket Repository ที่ ต้องการจะโคลนลงไป แล้วกดเลือก Destination Path ไปที่ C:/xampp/htdocs/your_project_name ซึ่งจะเป็นพาหุโพลเดอร์ที่โปรแกรม XAMPP สร้างไว้ใช้จำลองการทำงานของ Web Server (เปรียบเสมือน /var/www/html/your_project_name บนเครื่องแม่ข่าย) จากนั้นกดปุ่ม clone ดังภาพที่ 6.17



ภาพที่ 6.17 หน้าต่างโปรแกรม Source Tree ขณะทำการ clone โปรเจกต์ลงบนเครื่องคอมพิวเตอร์

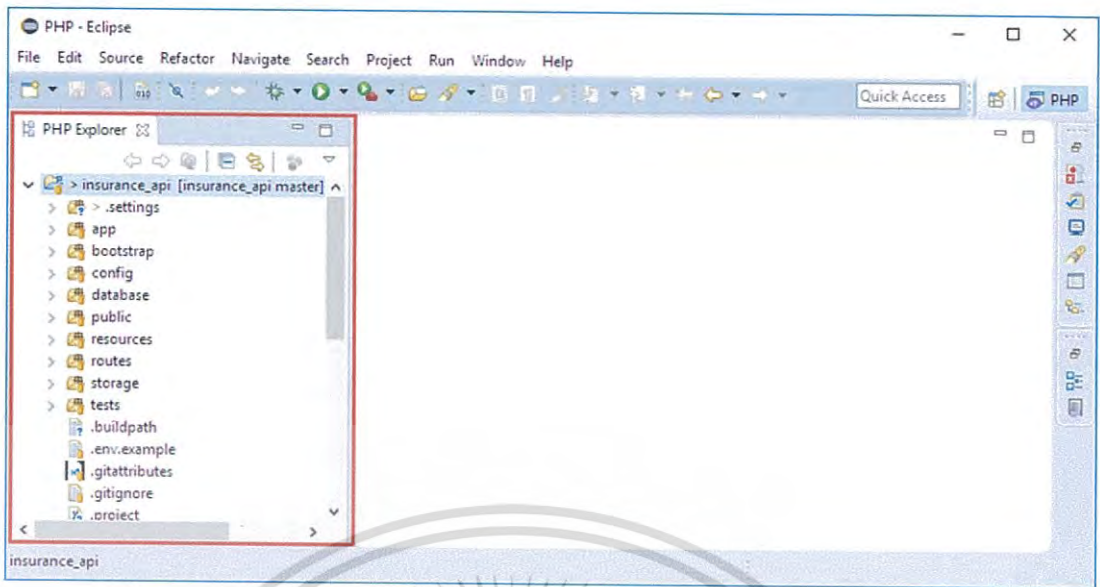
เมื่อโคลนโปรเจกต์เสร็จเรียบร้อยแล้วขั้นตอนต่อไปจะเป็นการ import project บนโปรแกรม Eclipse Mars เป็นโปรแกรมประเภท Editor ใช้สำหรับการเขียนโค้ดเพื่อพัฒนาระบบ

เมื่อเปิดโปรแกรม Eclipse Mars ขึ้นมาเลือก File > New > PHP Project จากนั้นในหน้าต่าง Create a PHP project กำหนดชื่อให้กับโปรเจกต์ในช่อง Project name ในหัวข้อ Content กดเลือก Create project at existing location และกดเลือก Directory ให้ชี้ไปยังพาทที่ได้โคลนโปรเจกต์จาก Git ไว้ จากนั้นกด Finish ดังภาพที่ 6.18



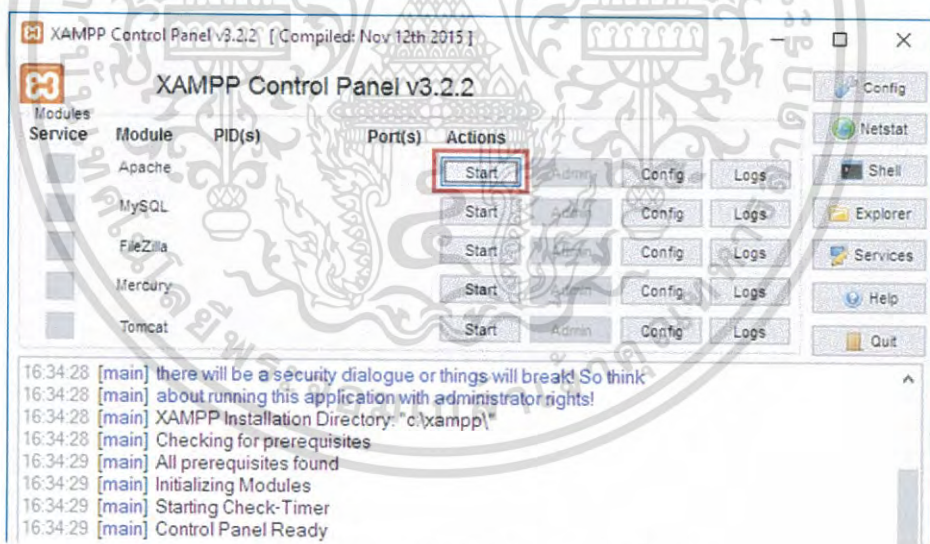
ภาพที่ 6.18 โปรแกรม Eclipse Mars ขณะสร้างโปรเจกต์จากโฟลเดอร์โปรเจกต์ที่โคลนจาก Git

เมื่อสร้างโปรเจกต์เรียบร้อยแล้วจะสามารถมองเห็นโครงสร้างของโปรเจกต์ได้จากหน้าต่างด้านซ้ายของโปรแกรม Eclipse Mars ดังภาพที่ 6.19 เป็นอันเสร็จสิ้นการ import โปรเจกต์เข้าในโปรแกรม Eclipse Mars



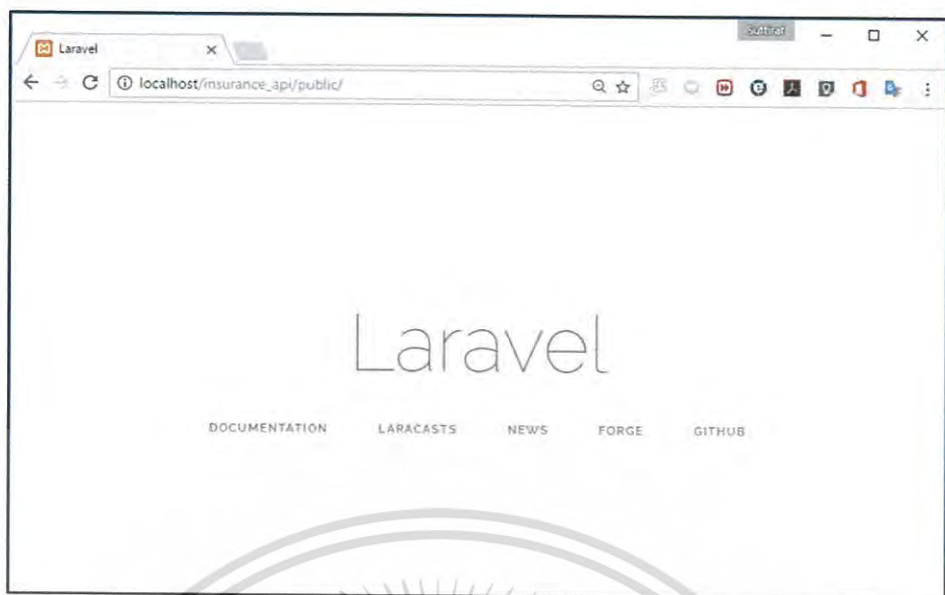
ภาพที่ 6.19 หน้าต่างโปรแกรม Eclipse Mars หลังจาก import โปรเจกต์เรียบร้อยแล้ว

หลังจากนั้นจะทำการทดสอบการทำงานของระบบที่จำลองสภาพแวดล้อมให้เหมาะสมต่อการทำงานขึ้นมาบนเครื่องคอมพิวเตอร์โดยเปิดโปรแกรม XAMPP ขึ้นมาและสั่งเปิดการทำงานของ Apache Webserver โดยการคลิกปุ่ม start ที่หัวข้อ Apache ดังภาพที่ 6.20



ภาพที่ 6.20 หน้าต่างโปรแกรม XAMPP

เมื่อ Apache ทำงานแล้วจะแสดงเป็นแถบสีเขียว ให้ทดสอบด้วยการเปิดเว็บเบราว์เซอร์ แล้วเข้าไปที่ http://localhost/your_project_name/public/ จะพบกับหน้าตาสีขาว ดังภาพที่ 6.21 แสดงว่าการทำงานในขั้นตอนนี้เสร็จสมบูรณ์แล้ว



ภาพที่ 6.21 หน้าต่างเบราว์เซอร์แสดงผลการทำงานของลาเวลเฟรมเวิร์กบนเครื่องคอมพิวเตอร์ จากการโคลนโปรเจกต์จากเครื่องแม่ข่าย ผ่านระบบ Git

6.1.4. ตั้งค่าฐานข้อมูลเพื่อพัฒนาบนเครื่องคอมพิวเตอร์

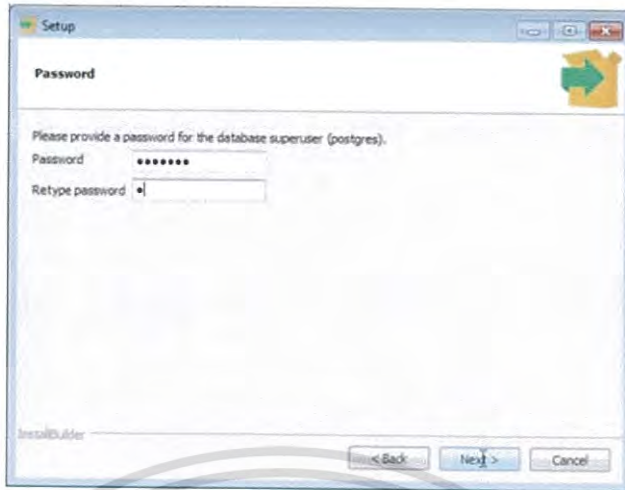
หลังจากการโคลนโปรเจกต์ลงบนเครื่องคอมพิวเตอร์สำหรับพัฒนาต่อไปจะเป็นการตั้งค่าระบบฐานข้อมูลสำหรับการพัฒนาบนเครื่องคอมพิวเตอร์ เริ่มจากดาวน์โหลดและติดตั้งโปรแกรม pgAdminIII ดังภาพที่ 6.22



ภาพที่ 6.22 หน้าจอโปรแกรมขณะทำการติดตั้งโปรแกรม PostgreSQL

ระหว่างขั้นตอนการติดตั้ง จะต้องมีการตั้งค่ารหัสสำหรับเข้าใช้งาน ฐานข้อมูลดังภาพที่ 6.23 จากนั้นเลือกตำแหน่งการติดตั้งให้อยู่ภายในโฟลเดอร์ของโปรแกรม XAMPP ที่ได้ติดตั้งไว้เช่น C:/xampp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.23 หน้าจอติดตั้งขณะตั้งค่ารหัสผ่านสำหรับเข้าใช้ระบบฐานข้อมูล PostgreSQL

ขั้นตอนต่อไปจะเป็นการติดตั้งหน้า phpPgadmin สำหรับเว็บให้สามารถทำงานบน Apache Webserver ได้

ดาวน์โหลดไฟล์สำหรับติดตั้ง phpPgadmin จากนั้นสร้างโฟลเดอร์ในโฟลเดอร์ที่ติดตั้งโปรแกรม XAMPP ชื่อ phppgadmin ดังภาพที่ 6.24 แล้วทำการแตกไฟล์ที่โหลดมาไปยังโฟลเดอร์

	phpMyAdmin	13/10/2559 14:44	File folder
<input checked="" type="checkbox"/>	phppgadmin	11/8/2559 21:24	File folder
	PostareSQL	13/10/2559 14:29	File folder

ภาพที่ 6.24 หน้าต่างแสดงโฟลเดอร์ phppgadmin ที่สร้างไว้ใน โฟลเดอร์ xampp

เมื่อทำการแตกไฟล์เสร็จเรียบร้อยแล้วให้เปิดไฟล์ config.inc.php ในโฟลเดอร์ conf เพื่อแก้ไข extra_login_security เป็น false ดังภาพที่ 6.25 หลังจากนั้นบันทึกการเปลี่ยนแปลง

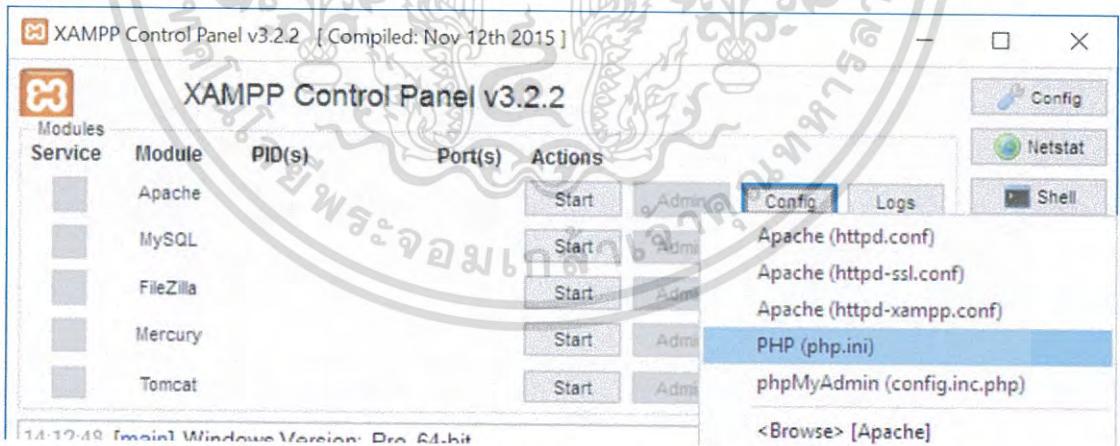
```

96 // 'default off' enables AutoComplete but turns it off by default.
97 // 'disable' disables AutoComplete.
98 $conf['autocomplete'] = 'default on';
99
100 // If extra login security is true, then logins via phpPgAdmin with no
101 // password or certain usernames (pgsql, postgres, root, administrator)
102 // will be denied. Only set this false once you have read the FAQ and
103 // understand how to change PostgreSQL's pg_hba.conf to enable
104 // passworded local connections.
105 $conf['extra_login_security'] = false;
106
107 // Only show owned databases?
108 // Note: This will simply hide other databases in the list - this does
109 // not in any way prevent your users from seeing other database by
110 // other means. (e.g. Run 'SELECT * FROM pg_database' in the SQL area.)
111 $conf['owned_only'] = false;
112
113 // Display comments on objects? Comments are a good way of documenting
114 // a database, but they do take up space in the interface.
115 $conf['show_comments'] = true;
116
117 // Display "advanced" objects? Setting this to true will show

```

ภาพที่ 6.25 ขณะแก้ไขไฟล์ config.inc.php โดยตั้งค่า extra_login_security เป็น false เพื่ออนุญาตให้มีการ login เข้าระบบฐานข้อมูลด้วยผู้ใช้ postgres

จากนั้นเปิดโปรแกรม XAMPP คลิกที่ปุ่ม config ของ Apache เลือก PHP (php.ini) ดังภาพที่ 6.26 จากนั้นแก้ไขโดยการลบ ; ด้านหน้าบรรทัด extension=php_pdo_pgsql.dll และ extension=php_pgsql.dll ออกดังภาพที่ 6.27 จากนั้นบันทึกการเปลี่ยนแปลง



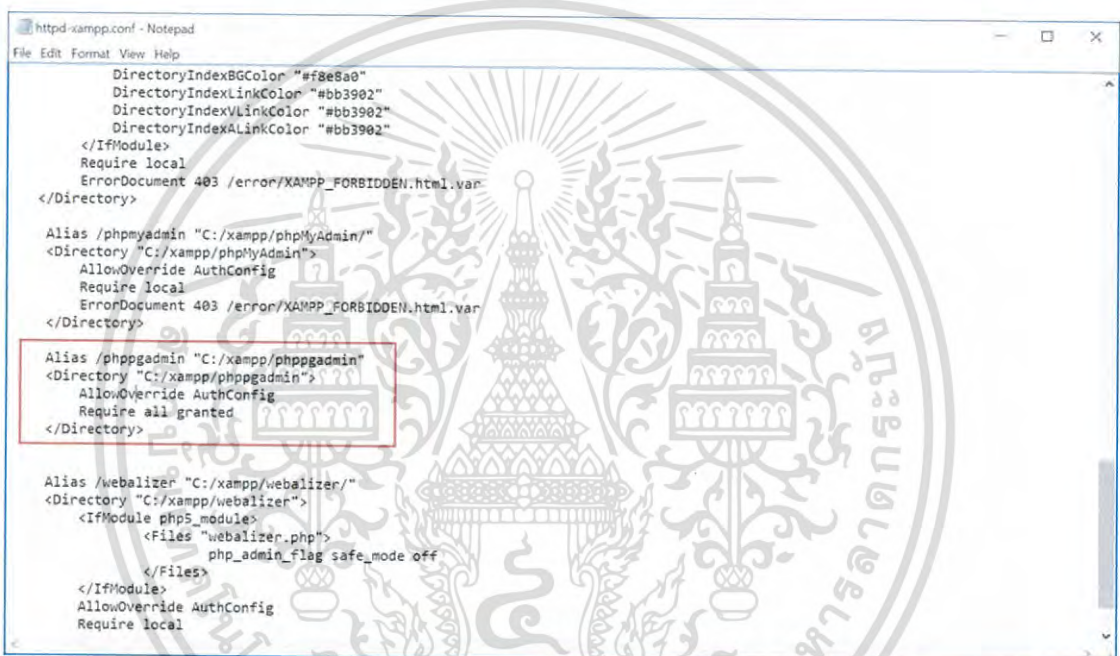
ภาพที่ 6.26 หน้าต่างโปรแกรม Xampp ขณะเข้าไปตั้งค่าไฟล์ php.ini

```
;extension=php_pdo_oci.dll
;extension=php_pdo_odbc.dll
extension=php_pdo_pgsql.dll
extension=php_pdo_sqlite.dll
extension=php_pgsql.dll
;extension=php_shmop.dll
```

; The MIBS data available in the PHP distribution must be installed.

ภาพที่ 6.27 ขณะแก้ไขไฟล์ php.ini โดยการลบ ; เพื่อใช้งานส่วนขยาย php_pdo_pgsql และ php_pgsql

กลับมาที่โปรแกรม XAMPP คลิกที่ปุ่ม config ของ Apache เลือก Apache (httpdxampp.conf) จากนั้นเพิ่มข้อความดังภาพที่ 6.28 และบันทึกการเปลี่ยนแปลง



ภาพที่ 6.28 ขณะแก้ไขไฟล์ httpd-xampp.conf โดยการเพิ่มโค้ดตามกรอบสี่เหลี่ยมสีแดงเพื่อให้สามารถใช้งาน phppgadmin ผ่านหน้าบราวเซอร์บน Apache Webserver ใน XAMPP ได้

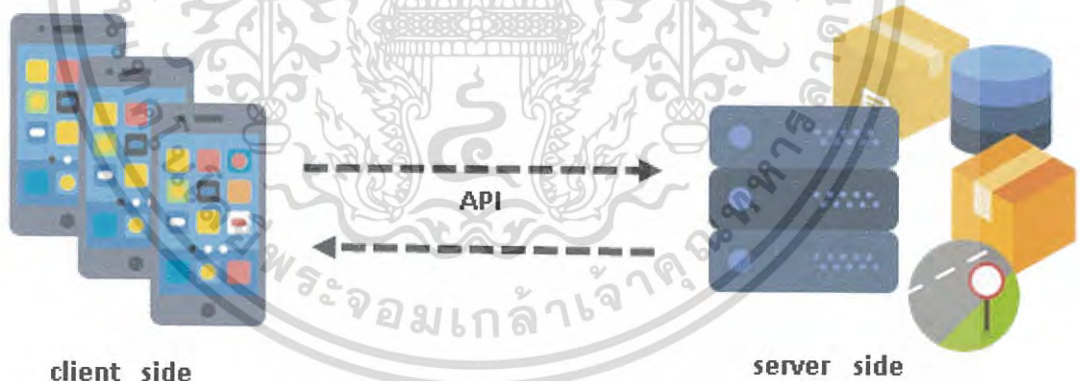
เริ่มการทำงานของ Apache webserver ใหม่โดยการกด stop และ start ในโปรแกรม XAMPP จากนั้นทดลองเปิดหน้า phppgadmin โดยการเข้าเว็บเบราว์เซอร์แล้วไปที่ localhost/phppgadmin จะปรากฏดังภาพที่ 6.29



ภาพที่ 6.29 ภาพบราวเซอร์ขณะเข้าสู่หน้า phppgadmin บน localhost

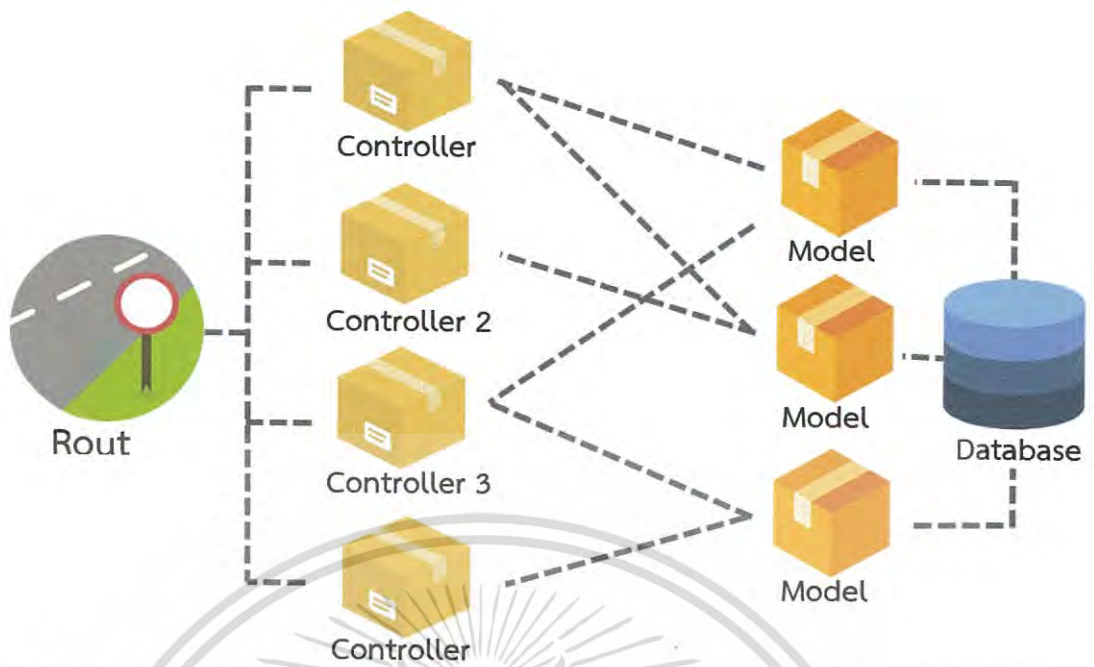
6.1.5. พัฒนาระบบภายใต้สถาปัตยกรรม โมเดล วิว คอนโทรลเลอร์

ระบบประกันภัยบนโมบายล์แพลตฟอร์ม ที่ได้ออกแบบและพัฒนาจะใช้หลักการทำงานแบบ client – server และจะใช้การติดต่อสื่อสารเพื่อแลกเปลี่ยนข้อมูลระหว่างทั้งสองฝั่งผ่าน API ในการรับส่ง request และ response ดังภาพที่ 6.30



ภาพที่ 6.30 แผนภาพแสดงการทำงานแบบ client -server ที่ติดต่อสื่อสารผ่าน API

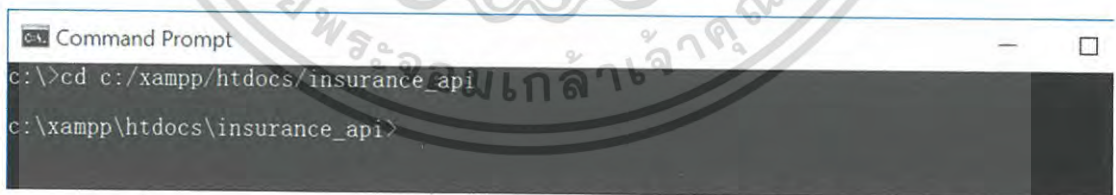
ภายในฝั่ง Server จะประกอบด้วยการทำงานในหลายๆส่วนภายใต้การพัฒนา ระบบด้วยสถาปัตยกรรมออกแบบ โมเดล วิว คอนโทรลเลอร์ (MVC) ซึ่งในระบบที่พัฒนาจะประกอบด้วย เร้าท์ (Route) โมเดล (Model) คอนโทรลเลอร์ (Controller) จะทำงานสอดคล้องกัน เพื่อให้บริการเว็บเซอร์วิสแก่ฝั่ง client ซึ่งเป็นโมบายล์แอปพลิเคชันดังภาพที่ 6.31



ภาพที่ 6.31 ภาพแสดงส่วนประกอบต่างๆ ระบบที่ทำงานร่วมกันบนเครื่อง server ที่พัฒนาด้วย ลาลาเวล

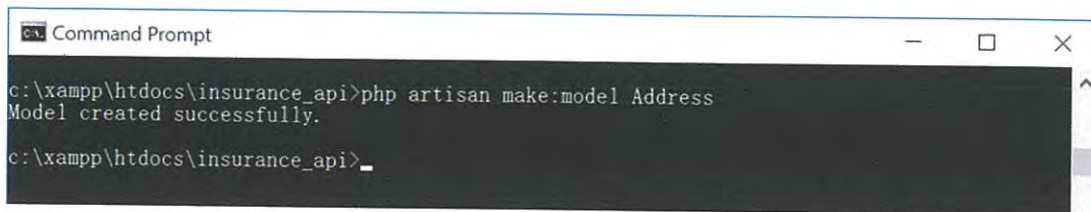
การพัฒนาเว็บด้วยลาลาเวลเฟรมเวิร์คจะต้องพัฒนาระบบภายใต้สถาปัตยกรรม การออกแบบแบบ MVC ซึ่งตัวลาลาเวลเฟรมเวิร์คเองสามารถสร้างโมเดล วิว และ คอนโทรลเลอร์ได้ ด้วยการใช้คำสั่ง Artisan ของลาลาเวลเฟรมเวิร์คเอง คำสั่ง Artisan เป็นฟังก์ชัน php สามารถใช้งานได้ผ่าน command line การใช้งานบน windows สามารถใช้ได้ผ่านโปรแกรม command prompt

การสร้างไฟล์ต่างๆด้วยคำสั่ง Artisan ให้เปิดโปรแกรม command prompt ขึ้นมา จากนั้น เข้าไปที่โฟลเดอร์ไฟล์โปรเจกต์ด้วยคำสั่ง `cd c:/your/directory/your_project_name` ดัง ภาพที่ 6.32



ภาพที่ 6.32 หน้าจอ command prompt ขณะใช้คำสั่ง cd เข้าไปยังโฟลเดอร์โปรเจกต์

การสร้างไฟล์โมเดล ใช้คำสั่ง `php artisan make:model YourModelName` เมื่อสร้างไฟล์สำเร็จจะปรากฏผลลัพธ์ดัง และจะปรากฏไฟล์โมเดลที่สร้างในโฟลเดอร์ app ภายในโฟลเดอร์โปรเจกต์ เช่น `c:/your/directory/your_project_name/app/YourModel.php`



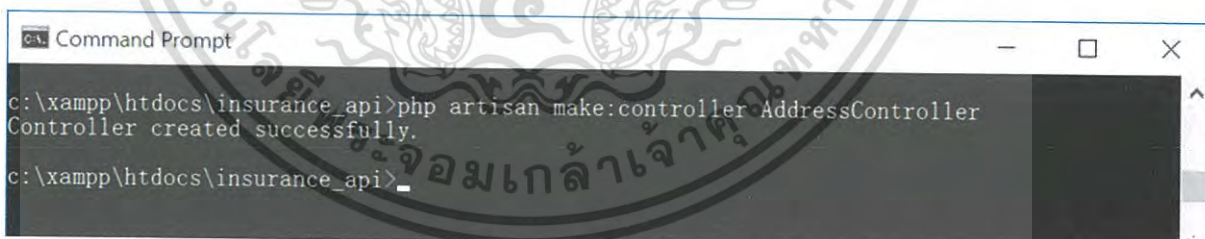
```
c:\xampp\htdocs\insurance_api>php artisan make:model Address
Model created successfully.
c:\xampp\htdocs\insurance_api>_
```

ภาพที่ 6.33 หน้าจอ command prompt ขณะใช้คำสั่ง Artisan ในการสร้างโมเดล



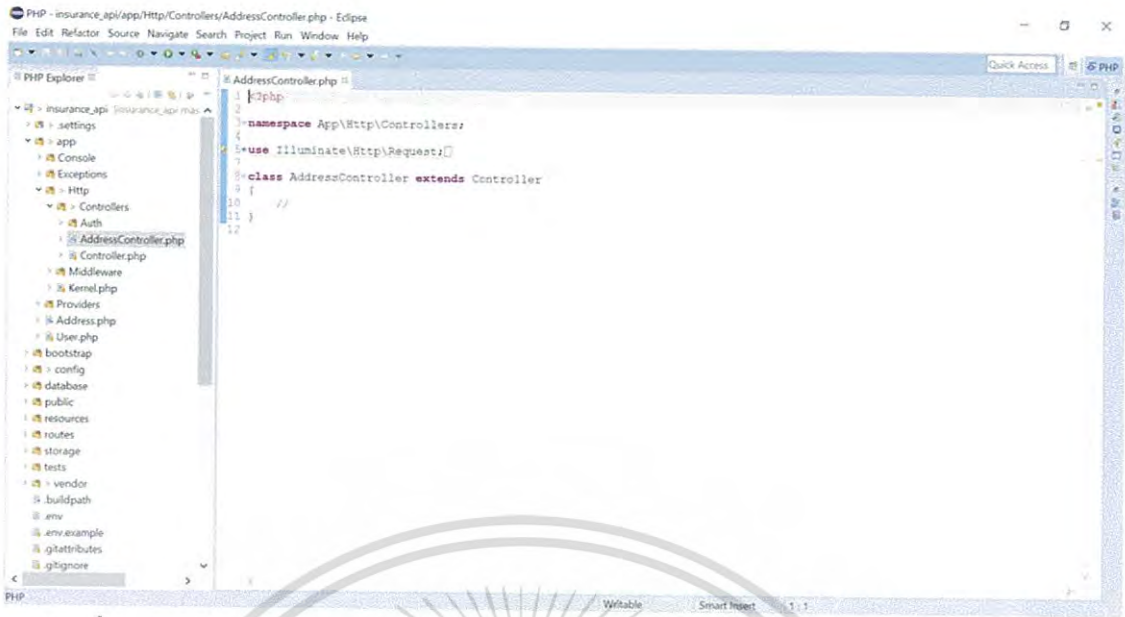
ภาพที่ 6.34 หน้าจอโปรแกรม Eclipse Mars แสดงโครงสร้างของโมเดลที่ได้จากการใช้คำสั่ง Artisan

การสร้างไฟล์คอนโทรลเลอร์ ใช้คำสั่ง `php artisan make :controller YourControllerName` เมื่อสร้างไฟล์สำเร็จจะปรากฏผลลัพธ์ดังภาพที่ 6.35 จะปรากฏไฟล์โมเดลที่สร้างในโฟลเดอร์ `app` ภายในโฟลเดอร์โปรเจกต์ เช่น `c:/your/directory /your_project_name/app/http/controllers/AddressController.php`



```
c:\xampp\htdocs\insurance_api>php artisan make:controller AddressController
Controller created successfully.
c:\xampp\htdocs\insurance_api>_
```

ภาพที่ 6.35 หน้าจอ command prompt ขณะใช้คำสั่ง Artisan ในการสร้างคอนโทรลเลอร์

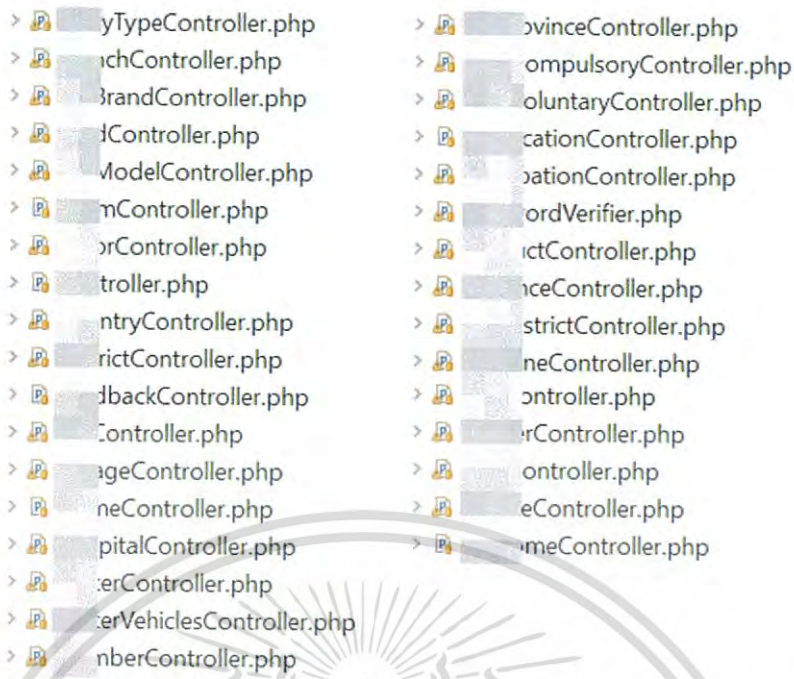


ภาพที่ 6.36 หน้าจอโปรแกรม Eclipse Mars แสดงโครงสร้างของคอนโทรลเลอร์ที่ได้จากการใช้คำสั่ง Artisan

หลังจากนั้นสร้างไฟล์โมเดล และคอนโทรลเลอร์ ที่จำเป็นในการทำงานของระบบที่ได้ออกแบบไว้ตาม activity diagram ทั้งหมดจนครบดังภาพที่ 6.37 และภาพที่ 6.38



ภาพที่ 6.37 ภาพแสดงไฟล์โมเดลทั้งหมดที่จำเป็นในการพัฒนาระบบประกันภัยบนโมบายล์แพลตฟอร์ม



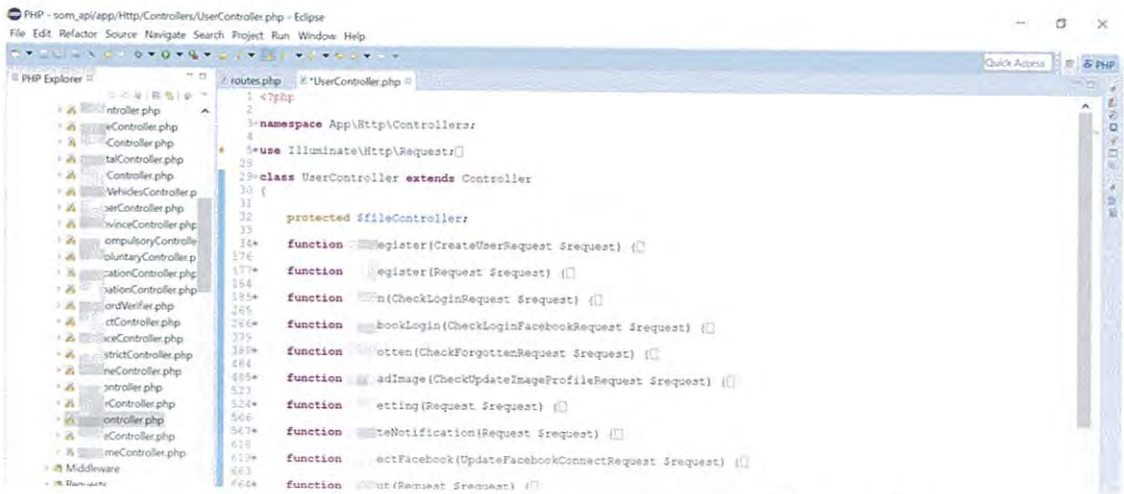
ภาพที่ 6.38 ภาพแสดงไฟล์คอนโทรลเลอร์ทั้งหมดที่จำเป็นในการพัฒนาระบบประกันภัยบนโมบายล์แพลตฟอร์ม

หลังจากนั้นเขียนโค้ดเพื่อทำการเชื่อมต่อโมเดลเข้ากับฐานข้อมูล โดยการกำหนดค่าในคลาสโมเดลแต่ละโมเดล ให้เชื่อมต่อไปยังฐานข้อมูลแต่ละตาราง ดังภาพที่ 6.39 และทำเช่นนี้กับทุกโมเดลตามที่ได้ออกแบบไว้เพื่อให้โมเดลทำหน้าที่ในการให้บริการข้อมูลแก่ส่วนอื่นๆ ในระบบ แทนการเข้าถึงฐานข้อมูลโดยตรง



ภาพที่ 6.39 ภาพแสดงไฟล์โมเดลที่เชื่อมต่อกับฐานข้อมูลตาราง timeline

ในการเขียนโค้ดการทำงานของระบบ เช่น ตรรกะการทำงานต่างๆ เงื่อนไข และการประมวลผลข้อมูลส่วนใหญ่จะถูกเขียนภายใน คลาสคอนโทรลเลอร์ โดยแยกเขียนแต่ละตรรกะการทำงานเป็นเมธอดดัง ภาพที่ 6.40 ให้สอดคล้องตามที่ได้ออกแบบไว้ใน activity diagram



ภาพที่ 6.40 ภาพแสดงไฟล์คอนโทรลเลอร์ที่ประกอบด้วยเมธอดต่างๆ

หลังจากการเขียนโค้ดการทำงานลงในคลาสคอนโทรลเลอร์ครบตามที่ได้ออกแบบไว้แล้ว การเรียกใช้คอนโทรลเลอร์เหล่านี้ จะถูกเรียกใช้ด้วย เร้าท์ (Route) ซึ่งเร้าท์จะทำหน้าที่ในการรับ request และนำ request แต่ละอันไปยังเป้าหมายที่ถูกตั้งตามที่กำหนด การกำหนดปลายทางของ request จะถูกกำหนดด้วย URL ซึ่งภายในเร้าท์นี้เองที่สามารถออกแบบให้ API มีลักษณะการเรียกใช้งานเป็นไปตามการออกแบบแบบ RESTful API ไฟล์เร้าท์จะอยู่ในโฟลเดอร์ Http ภายในโฟลเดอร์โปรเจกต์ เช่น c:/your/directory/your_project_name/app/Http/route.php ดังภาพที่ 6.41



ภาพที่ 6.41 ภาพแสดงไฟล์ route.php ที่ทำหน้าที่ในการกำหนดเส้นทางในการเข้าถึง API

รูปแบบการกำหนดการทำงานของ route จะกำหนดด้วยการเขียนโค้ด เพื่อระบุเส้นทางให้เร้าท์นำทางไปยังการประมวลผลถัดไป เมื่อมีการส่ง request เข้ายังระบบหลังบ้านดังภาพที่ 6.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

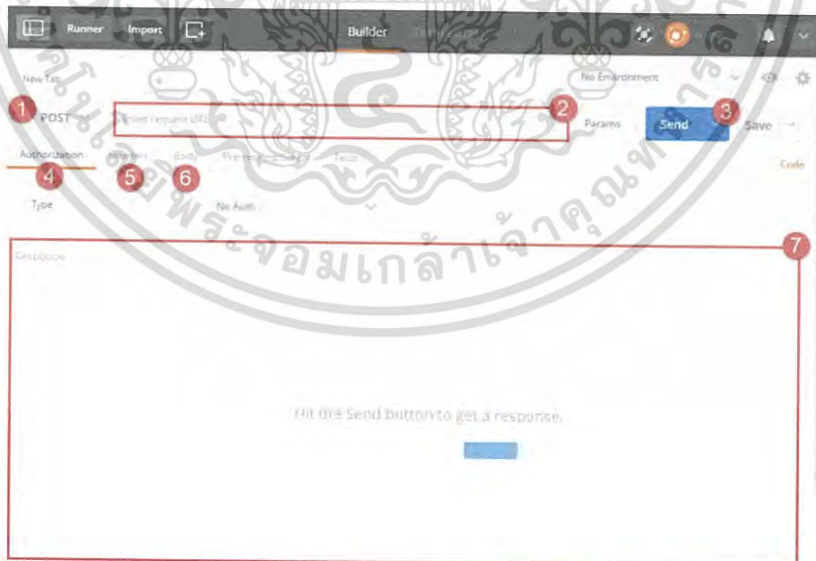
1 Route::group(array('prefix' => 'user'), function()
{
    Route::post('/test1', 'TestController@test1');
    Route::post('/test2', 'TestController@test2');
});
    3      4      5      6
  
```

ภาพที่ 6.42 ภาพแสดงรูปแบบการเขียน route ในลาลาเวลเฟรมเวิร์ค

จากภาพที่ 6.42 ในกรอบที่ 1 เป็นคำสั่งสำหรับระบุให้ route ที่อยู่ภายใน group จะเข้าถึงได้ผ่านเส้นทาง user เช่น api.example.com/user/test1 เป็นต้น

ในกรอบที่ 2 เป็นคำสั่งสำหรับกำหนด route โดยแต่ละ route จะมีส่วนประกอบทั้งหมด 4 ส่วนคือ จุดที่ 3 สำหรับกำหนด HTTP Verb ที่จะใช้ในการเข้าถึง route เช่น GET หรือ POST เป็นต้น จุดที่ 4 สำหรับกำหนดเส้นทางที่จะเข้าถึง route เช่น /test2 หมายถึงจะสามารถเข้าถึง route นี้ได้จาก api.example.com/test2 เป็นต้น แต่จากตัวอย่างข้างต้น route ถูกเขียนไว้ภายใต้ group การเข้าถึง route จึงต้องเข้าที่ api.example.com/user/test2 แทน จุดที่ 5 และ 6 ใช้สำหรับกำหนด callback หลังจากมีการเข้าถึง route โดยจุดที่ 5 จะกำหนดให้เรียก controller ที่กำหนดมาทำงาน และจุดที่ 6 จะกำหนดเมธอด ที่ต้องการให้ทำงาน ซึ่งจะต้องเป็นเมธอดที่อยู่ภายใน controller ที่กำหนดในจุดที่ 5 ด้วย

6.2. การจำลองการร้องขอเพื่อทดสอบส่วนต่อประสานตีโปรแกรมประยุกต์ด้วยโปรแกรม Postman



ภาพที่ 6.43 หน้าต่างโปรแกรม Postman แสดงส่วนประกอบสำคัญที่ใช้ในการทดสอบระบบ API

6.2.1. ส่วนประกอบของโปรแกรม Postman

จากภาพที่ 6.43 แสดงให้เห็นหน้าต่างของโปรแกรม Postman ที่จะใช้ในการทดสอบการทำงานซึ่งมีส่วนประกอบสำคัญต่าง ๆ ดังต่อไปนี้

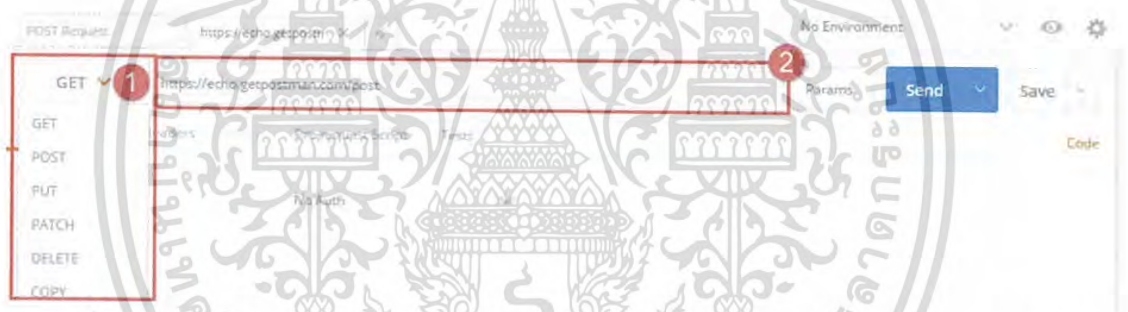
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.) ส่วนกำหนดกริยา (HTTP verb) ในการส่ง request ไปยังปลายทาง
- 2.) ส่วนกำหนด URL ในการส่ง request เพื่อกำหนดปลายทางในการส่ง
- 3.) ปุ่มกดสำหรับส่ง request
- 4.) ส่วนกำหนดรูปแบบการอนุญาต (Authorization) ให้กับ request
- 5.) ส่วนกำหนดข้อมูลในส่วนหัว (Headers) ของ request
- 6.) ส่วนกำหนดคุณลักษณะ (Attribute) ของ request
- 7.) ส่วนแสดงผลลัพธ์ แสดงผลของ response ในรูปแบบต่างๆ

6.2.2. การใช้งานโปรแกรม Postman

ในการใช้งานโปรแกรม Postman เพื่อทดสอบ API จะต้องมีการกำหนดค่าในการส่ง request ให้เหมาะสมตาม API ที่จะทำการทดสอบซึ่งโดยปกติแล้วรูปแบบการกำหนดค่าสามารถดูได้จากคู่มือการใช้งาน API ซึ่งจะกำหนดรายละเอียดรูปแบบการส่ง request ที่ถูกต้องไว้

เมื่อทราบรูปแบบการ request แล้ว จึงกำหนดรูปแบบในโปรแกรม Postman ขั้นแรกเลือก HTTP Verb ตามที่รูปแบบ request ที่ต้องการโดยคลิกจุดที่ 1 จากนั้นกำหนด URL ปลายทางที่ต้องการส่ง request ไปโดยพิมพ์ URL ลงในจุดที่ 2 ดังภาพที่ 6.44



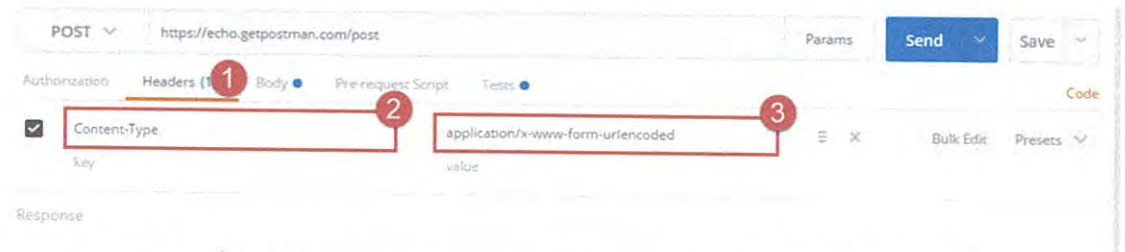
ภาพที่ 6.44 โปรแกรม Postman ขณะกำหนด HTTP Verb และ URL ปลายทางให้กับ request

หากรูปแบบการ request สามารถส่ง attribute ไปกับ request ได้ (เช่น HTTP Verb เป็น POST) ให้คลิกที่ body (จุดที่ 1) เพื่อกำหนด HTTP Body จากนั้นทำการกำหนดรูปแบบข้อมูลที่将会ทำการส่ง (จุดที่ 2) ชื่อ attribute ที่ต้องการลงในช่อง key (จุดที่ 3) และกำหนดค่าของ attribute ในช่อง value (จุดที่ 4) ดังภาพที่ 6.45



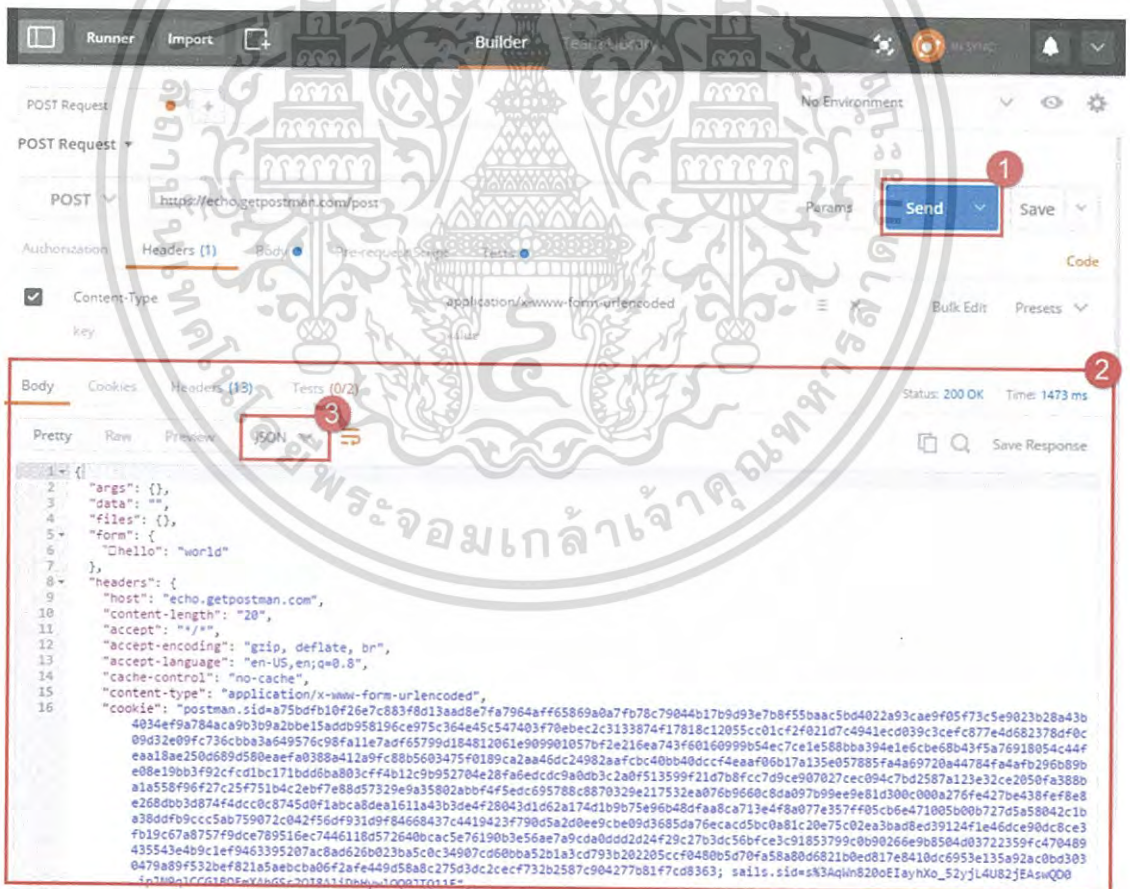
ภาพที่ 6.45 โปรแกรม Postman ขณะกำหนด attribute ที่ต้องการส่งค่าไปพร้อมกับ request

หากรูปแบบการ request ต้องการให้กำหนดส่วน Headers ของ request ให้คลิกที่ Headers (จุดที่ 1) จากนั้นกำหนด header-key ที่ต้องการลงในช่อง key (จุดที่ 2) และกำหนดค่าในช่อง value (จุดที่ 3) ดังภาพที่ 6.46



ภาพที่ 6.46 โปรแกรม Postman ขณะกำหนด Headers ให้กับ request

หลังจากกำหนดรูปแบบของ request เรียบร้อย กดปุ่ม send (จุดที่ 1) เพื่อทำการส่ง request ไปยังปลายทาง จากนั้น response ที่ได้รับจะแสดงอยู่ด้านล่าง (กรอบที่ 2) และสามารถเลือกรูปแบบการแสดงผล response ที่ได้ในรูปแบบต่างๆ ได้ (จุดที่ 3) ดังภาพที่ 6.47



ภาพที่ 6.47 โปรแกรม Postman ขณะแสดงผลลัพธ์ที่ได้จาก response

6.3. ผลลัพธ์การทำงานของส่วนต่อประสานโปรแกรมประยุกต์ที่ถูกต้อง

6.3.1. API สำหรับหน้าล็อกอิน

API สำหรับล็อกอินกำหนดรูปแบบrequest แสดงดัง

ตารางที่ 6.1 รูปแบบของ HTTP body ดังตารางที่ 6.2 รูปแบบของ response ดังตารางที่ 6.3 และผลลัพธ์ดังภาพที่ 6.48

ตารางที่ 6.1 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับล็อกอิน

กริยา HTTP	URL	คำอธิบาย
POST	api.dev.mysagi.com/public/api/v1/user/login	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับสมัครสมาชิก

ตารางที่ 6.2 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับล็อกอิน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	username	เบอร์โทรศัพท์มือถือ, อีเมล	0812345678 หรือ test@email.com
2	password	รหัสผ่าน	password
3	fcm_id (ไม่จำเป็น)	รหัสประจำเครื่องบนระบบปฏิบัติการ android	*อักขระจากระบบ
4	apple_id (ไม่จำเป็น)	รหัสประจำเครื่องบนระบบปฏิบัติการ iOS	*อักขระจากระบบ

ตารางที่ 6.3 รูปแบบของ attribute ภายใน response ของ API สำหรับล็อกอิน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	ข้อความอธิบาย	เข้าสู่ระบบสำเร็จ
2	status	ตัวเลขบอกสถานะ	1
3	image_profile	url รูปโปรไฟล์	*URL รูปภาพ
4	access_token	รหัสสำหรับใช้ยืนยันตัวตนชั่วคราว	*อักขระจากระบบ

```

1 {
2   "message": "เข้าสู่ระบบสำเร็จ",
3   "status": 1,
4   "image_profile": "http://[redacted]/public/files/a9d87ba765676e77021813d18b14ae1f4d98fe37.png",
5   "access_token": "Bearer AfW0qgi [redacted] sV3jh7dFD9uN#iSXgt5"
6 }

```

ภาพที่ 6.48 response ที่ได้รับกลับมาจาก API สำหรับล็อกอิน

6.3.2. API สำหรับสมัครสมาชิก

API สำหรับสมัครสมาชิกกำหนดรูปแบบ request แสดงดังตารางที่ 6.4 รูปแบบของ HTTP body ดังตารางที่ 6.5 รูปแบบของ response ดังตารางที่ 6.6 และผลลัพธ์ดังภาพที่ 6.49

ตารางที่ 6.4 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับสมัครสมาชิก

กริยา	URL	คำอธิบาย
HTTP POST	api.dev.mysagi.com/public/api/v1/user/register	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับสมัครสมาชิก

ตารางที่ 6.5 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับสมัครสมาชิก

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	mobile	เบอร์โทรศัพท์มือถือ	0812345678
2	password	รหัสผ่าน	password
3	name	ชื่อ	suttirat
4	last_name	นามสกุล	kradatong
5	email	อีเมล	test@email.com
6	file	ไฟล์ภาพนามสกุล .jpg .png .gif เท่านั้น	Profile.png

ตารางที่ 6.6 รูปแบบของ attribute ภายใน response ของ API สำหรับสมัครสมาชิก

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	ข้อความอธิบาย	เข้าสู่ระบบสำเร็จ
2	status	ตัวเลขบอกสถานะ	1
3	image_profile	url รูปโปรไฟล์	http://xxx.com/file/aDBanner.png
4	access_token	รหัสสำหรับใช้ยืนยันตัวตนชั่วคราว	Bearer 5kJYZzicplxxxxxWpWsu8...

```

1 {
2   "message": "สมัครสมาชิกสำเร็จ",
3   "status": 1,
4   "image_profile": "http://.../public/files/a9d87ba765676e77021813d18b14ae1f4d98fe37.png",
5   "access_token": "Bearer ..."
6 }

```

ภาพที่ 6.49 response ที่ได้รับกลับมาจาก API สำหรับสมัครสมาชิก

6.3.3. API สำหรับหน้าลิ้มรสผ่าน

API สำหรับสมัครสมาชิกกำหนดรูปแบบ request แสดงดังตารางที่ 6.7 รูปแบบของ HTTP body แสดงดังตารางที่ 6.8 รูปแบบของ response ดังตารางที่ 6.9 และผลลัพธ์ดังภาพที่ 6.50 และ ภาพที่ 6.51

ตารางที่ 6.7 รูปแบบ request และ HTTP Verb เพื่อร้องขอไปยัง API สำหรับกู้รหัสผ่าน

กริยา	URL	คำอธิบาย
HTTP		
POST	api.dev.mysagi.com/public/api/v1/user/forgotten	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับกู้รหัสผ่าน

ตารางที่ 6.8 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับกู้รหัสผ่าน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	username	เบอร์โทรศัพท์มือถือหรืออีเมล	0812345678หรือ test@email.com

ตารางที่ 6.9 รูปแบบของ attribute ภายใน response ของ API สำหรับกู้รหัสผ่าน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	ข้อความอธิบาย	กรุณาตรวจสอบอีเมลของคุณ
2	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "message": "กรุณาตรวจสอบอีเมลของคุณ",
3   "status": 1
4 }

```

ภาพที่ 6.50 response ที่ได้รับกลับมาจาก API สำหรับกู้รหัสผ่านด้วยอีเมล

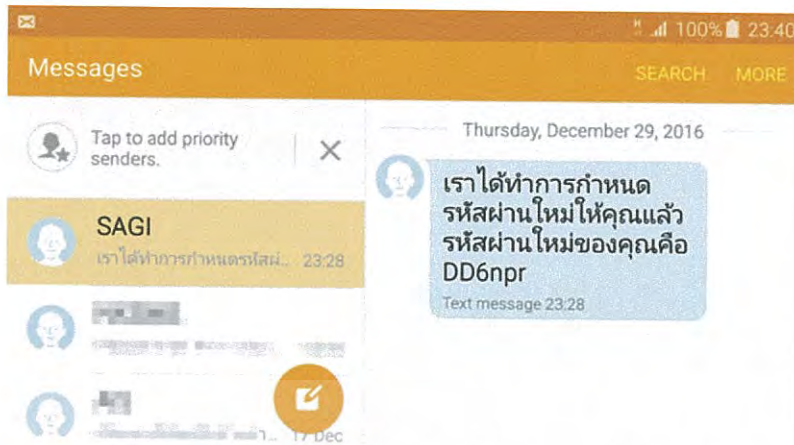
```

1 {
2   "message": "กรุณาตรวจสอบ SMS ของคุณ",
3   "status": 1
4 }

```

ภาพที่ 6.51 response ที่ได้รับกลับมาจาก API สำหรับกู้รหัสผ่านด้วยเบอร์โทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 6.52 เมื่อเลือกกู้รหัสผ่านโดยใส่ username เป็นอีเมล API จะตั้งรหัสใหม่แล้วส่งรหัสผ่านไป
ยังSMS



ghjbG6

ภาพที่ 6.53 เมื่อเลือกกู้รหัสผ่านโดยใส่ username เป็นอีเมล API จะตั้งรหัสใหม่แล้วส่งรหัสผ่านไป
ยังอีเมล

6.3.4. API สำหรับหน้าหลัก

API สำหรับหน้าหลักกำหนดรูปแบบ request แสดงดังตารางที่ 6.10 และกำหนด HTTP header ดังตารางที่ 6.11 รูปแบบของ response ดังตารางที่ 6.12 และผลลัพธ์ดังภาพที่ 6.54 ตารางที่ 6.10 รูปแบบกริยา HTTP เพื่อร้องขอไปยัง API สำหรับหน้าหลัก

กริยา HTTP	URL	คำอธิบาย
POST	api.dev.mysagi.com/public/api/v1/timeline/card	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับสมัครสมาชิก

ตารางที่ 6.11 รูปแบบของ Header ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าหลัก

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxXXXXXWpWsu8...

ตารางที่ 6.12 รูปแบบของ attribute ในการตอบกลับของ API สำหรับหน้าหลัก

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Timeline_card	กลุ่มของข้อมูลของการ์ดในหน้าหลักแต่ละใบ	"timeline_card": [กลุ่มข้อมูล 1.1.-1.5.]
1.1.	id	ไอดีของการ์ด	1
1.2.	type	ประเภทของการ์ด	promotion, product
1.3.	title	ชื่อการ์ด	Test
1.4.	description	รายละเอียด	xxxxxxxxxxxxxxxxxxxx...
1.5.	img	url รูปภาพบนการ์ด	http://xxx.com/file/aDBanner.png
2	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "timeline_card": [
3     {
4       "id": 3,
5       "type": "promotion",
6       "title": "Test 1 Promotion",
7       "description": "promotion promotion promotion promotion promotion promot
8       "img": "http://[redacted]/files/aDBanner.png"
9     },
10    {
11      "id": 4,
12      "type": "promotion",
13      "title": "Test 2 Promotion",
14      "description": "promotion promotion promotion promotion promotion promot
15      "img": "http://[redacted]/files/aDBanner.png"
16    },
17    {
18      "id": 5,
19      "type": "promotion",
20      "title": "Test",
21      "description": "http://develop.mysagi.com/index.php?r=store%2Fplanmobile&id=pa9aa.4",
22      "img": "http://[redacted]/files/healthCardView.png"
23    }
24  ],
25  "status": 1
26 }

```

ภาพที่ 6.54 response ที่ได้รับกลับมาจาก API สำหรับหน้าหลัก

6.3.5. API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น

API สำหรับสมัครสมาชิกกำหนดรูปแบบ request แสดงดังตารางที่ 6.13 และกำหนด HTTP headers ดังตารางที่ 6.14 รูปแบบของ response ดังตารางที่ 6.15 และผลลัพธ์ดังภาพที่ 6.55

ตารางที่ 6.13 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/timeline/carddetail/{id}	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น และแทน {id} ด้วย ไอดีของการ์ดชนิดโปรโมชั่นที่ต้องการ

ตารางที่ 6.14 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kYZzicplxxxxxWpWsu8...

ตารางที่ 6.15 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	card_detail	กลุ่มของข้อมูลของการ์ดเฉพาะไอดีที่ต้องการ	"card_detail": [กลุ่มข้อมูล 1.1.-1.7.]
1.1.	id	ไอดีของการ์ด	1
1.2.	type	ประเภทของการ์ด	promotion
1.3.	title	ชื่อการ์ด	Test
1.4.	description	รายละเอียด	xxxxxxxxxxxxxxxxxxxx...
1.5.	img	url รูปภาพบนการ์ด	http://xxx.com/file/aDBanner.png
1.6.	created_at	เวลาที่สร้าง	2016-11-11 00:00:00.000000
1.7.	updated_at	เวลาที่อัปเดตล่าสุด	2016-11-11 00:00:00.000000
2	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "card_detail": {
3     "id": 3,
4     "type": "promotion",
5     "title": "Test 1 Promotion",
6     "description": "promotion promotion promotion promotion promotion promotion promotion promotion promotion",
7     "img": "http://xxx.com/file/aDBanner.png",
8   },
9   "created_at": {
10    "date": "2016-11-11 00:00:00.000000",
11    "timezone_type": 3,
12    "timezone": "Asia/Bangkok"
13  },
14  "updated_at": {
15    "date": "2016-11-11 00:00:00.000000",
16    "timezone_type": 3,
17    "timezone": "Asia/Bangkok"
18  }
19 },
20 "status": 1

```

ภาพที่ 6.55 response ที่ได้รับกลับมาจาก API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น

6.3.6. API สำหรับหน้าซื้อประกันออนไลน์

API สำหรับสมัครสมาชิกกำหนดรูปแบบ request แสดงดังตารางที่ 6.16 และกำหนด HTTP headers ดังตารางที่ 6.17 รูปแบบของ response ดังตารางที่ 6.18 และผลลัพธ์ดังภาพที่ 6.56

ตารางที่ 6.16 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าซื้อประกันออนไลน์

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/timeline/carddetail/{id}	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าแสดงรายละเอียดโปรโมชั่น และแทน {id} ด้วย ไอดีของการ์ดชนิดผลิตภัณฑ์ที่ต้องการ

ตารางที่ 6.17 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าซื้อประกันออนไลน์

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.18 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าซื้อประกันออนไลน์

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	card_detail	กลุ่มของข้อมูลของการ์ดเฉพาะไอดีที่ต้องการ	"card_detail": [กลุ่มข้อมูล 1.1.-1.7.]
1.1.	id	ไอดีของการ์ด	1
1.2.	type	ประเภทของการ์ด	product
1.3.	title	ชื่อการ์ด	Product name
1.4.	description	url หน้าซื้อประกันออนไลน์	http://xxx.com/index.php?r=store%2Fplanmobile&id=xxx.x
1.5.	img	url รูปภาพบนการ์ด	http://xxx.com/file/aDBanner.png
1.6.	created_at	เวลาที่สร้าง	2016-11-11 00:00:00.000000
1.7.	updated_at	เวลาที่อัปเดตล่าสุด	2016-11-11 00:00:00.000000
2	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "card_detail": {
3     "id": 1,
4     "type": "product",
5     "title": "Test 2 Produc",
6     "description": "http://.../index.php?r=...&id=pa9aa.4",
7     "img": "http://.../files/accidentCardView.png",
8     "created_at": {
9       "date": "2016-11-11 00:00:00.000000",
10      "timezone_type": 3,
11      "timezone": "Asia/Bangkok"
12    },
13    "updated_at": {
14      "date": "2016-11-11 00:00:00.000000",
15      "timezone_type": 3,
16      "timezone": "Asia/Bangkok"
17    }
18  },
19  "status": 1
20 }

```

ภาพที่ 6.56 response ที่ได้รับกลับมาจาก API สำหรับหน้าซื้อประกันออนไลน์

6.3.7. API สำหรับหน้าบัตรประกันของฉัน

API สำหรับหน้าบัตรประกันของฉันกำหนดรูปแบบ request แสดงดังตารางที่ 6.19 และกำหนด HTTP headers ดังตารางที่ 6.20 รูปแบบของ response ดังตารางที่ 6.21 และผลลัพธ์ดังภาพที่ 6.57

ตารางที่ 6.19 request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าบัตรประกันของฉัน

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/ card/list	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าบัตรประกันของฉัน

ตารางที่ 6.20 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าบัตรประกันของฉัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kYZZicplxxxxxWpWsu8...

ตารางที่ 6.21 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าบัตรประกันของฉัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	identity_id	รหัสบัตรประชาชน	1100501183333
2	pa (ไม่จำเป็น)	จำนวนกรมธรรม์ ประเภท pa ที่มีอยู่	2
3	motor (ไม่จำเป็น)	จำนวนกรมธรรม์ ประเภท motor ที่มีอยู่	2

4	health (ไม่จำเป็น)	จำนวนกรมธรรม์ ประเภท health ที่มีอยู่	2
5	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "identity_id": "3410862202737",
3   "pa": 3,
4   "motor": 3,
5   "health": 0,
6   "status": 1
7 }

```

ภาพที่ 6.57 response ที่ได้รับกลับมาจาก API สำหรับหน้าบัตรประกันของฉัน

6.3.8. API สำหรับหน้าบัตรประกันจำลอง

API สำหรับหน้าบัตรประกันจำลองกำหนดรูปแบบ request ดังตารางที่ 6.22 และกำหนด HTTP headers ดังตารางที่ 6.23 รูปแบบของ response ดังตารางที่ 6.24 และผลลัพธ์ดังภาพที่ 6.58

ตารางที่ 6.22 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าบัตรประกันจำลอง

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/ api/v1/card/detail/ {insurance_type}	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าบัตรประกันของฉันและแทน {insurance_type} ด้วยประเภทกรมธรรม์

ตารางที่ 6.23 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าบัตรประกันจำลอง

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.24 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าบัตรประกันจำลอง

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	identity_id	รหัสบัตรประชาชน	1100501183333
2	type	ประเภทกรมธรรม์	pa
3	detail	กลุ่มของข้อมูล กรมธรรม์เฉพาะประเภทที่ระบุที่	"detail": [กลุ่มข้อมูล 3.1.- 3.11.]

3.1	name	ชื่อ	สมศรี
3.2	last_name	นามสกุล	ดีใจ
3.3	id_order	ไอดีสั่งซื้อ	75e68e7a-e7d0-42db-a969-xxxxx..
3.4	type	ประเภทกรมธรรม์	pa
3.5	code	รหัสประเภทกรมธรรม์	xx9xx.x
3.6	price	ราคา	500
3.7	insurance_start	ระยะเวลาเริ่มต้นความคุ้มครอง	2016-10-06 00:00:00
3.8	insurance_end	ระยะเวลาสิ้นสุดความคุ้มครอง	2017-10-06 00:00:00
3.9	policy_no	เลขกรมธรรม์	001-XX-XXXX-XXXXXXXX (XX)
3.10	remain_day	ระยะเวลาคุ้มครองที่เหลือ (วัน)	200
3.11	product_name	ชื่อกรมธรรม์	ทีเอ เพอร์เฟค XXXXXX
4	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "identity_id": "341086-XXXX",
3   "type": "pa",
4   "detail": [
5     {
6       "name": "สมศรี",
7       "last_name": "ดีใจ",
8       "id_order": "6a2d5856-4dc9-4b26-83d2-XXXX",
9       "type": "pa",
10      "code": "pa",
11      "price": 500,
12      "insure_start": "2016-10-06 00:00:00",
13      "insure_end": "2017-10-06 00:00:00",
14      "policy_no": "001-MS-2016-0000119 (PA)",
15      "remain_day": 279,
16      "product_name": "ทีเอ เพอร์เฟค XXXXXX"
17    },
18    {
19      "name": "สมศรี",
20      "last_name": "ดีใจ",
21      "id_order": "75e68e7a-e7d0-42db-a969-XXXX",
22      "policy_no": "001-MS-2016-0000119 (PA)",
23      "remain_day": 279,
24      "product_name": "ทีเอ เพอร์เฟค XXXXXX"
25    }
26  ],
27   "status": 1
28 }

```

ภาพที่ 6.58 response ที่ได้รับกลับมาจาก API สำหรับหน้าหน้าบัตรประกันจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3.9. สำหรับหน้าแผนที่

API สำหรับหน้าแผนที่ที่มีรูปแบบ request สองแบบคือ request ข้อมูลแผนที่สถานพยาบาล และข้อมูลแผนที่อยู่ช่อมรถ ซึ่งแสดงดังตารางที่ 6.25 รูปแบบของ response เมื่อเรียก API เพื่อขอแผนที่สถานพยาบาลแสดงดังตารางที่ 6.26 และ response ดัง

ภาพที่ 6.59 ส่วนรูปแบบของ response เมื่อเรียก API เพื่อขอแผนที่อยู่ช่อมรถแสดงดังตารางที่ 6.27 และ response ดังภาพที่ 6.63

ตารางที่ 6.25 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแผนที่สถานที่แต่ละแบบ

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/hospital/list	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าแผนที่เพื่อค้นหาสถานพยาบาล
GET	api.dev.mysagi.com/public/api/v1/garage/list	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าแผนที่เพื่อค้นหาอยู่ช่อมรถ

ตารางที่ 6.26 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่สถานพยาบาล

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	hospital_list	กลุ่มของข้อมูลสถานพยาบาล	"hospital_list": [กลุ่มข้อมูล 1.1.- 1.18.]
1.1	name	ชื่อสถานพยาบาล	pa
1.2	name_en	ชื่อสถานพยาบาลภาษาอังกฤษ	โรงพยาบาลกรุงเทพ
1.3	type_th	ประเภทสถานพยาบาล	โรงพยาบาล
1.4	type_en	ประเภทสถานพยาบาลภาษาอังกฤษ	Hospital
1.5	phone	เบอร์โทรศัพท์สถานพยาบาล	023103000,023180066
1.6	latlng	ละติจูด,ลองติจูด ของสถานพยาบาล	13.7485861,100.5810701
1.7	area_th	พื้นที่ให้บริการ	กรุงเทพและปริมณฑล
1.8	area_en	พื้นที่ให้บริการภาษาอังกฤษ	Bangkok and Vicinity
1.9	accident	* ขึ้นกับบริษัทประกันภัย	false
1.10	health	* ขึ้นกับบริษัทประกันภัย	false

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.11	is_ipd	* ขึ้นกับบริษัทประกันภัย	false
1.12	is_opd	* ขึ้นกับบริษัทประกันภัย	false
1.13	care_card	* ขึ้นกับบริษัทประกันภัย	false
1.15	provider	* ขึ้นกับบริษัทประกันภัย	TPA
1.16	display	* ขึ้นกับบริษัทประกันภัย	true
1.17	province_id	ไอดีจังหวัด	0
1.18	province_name	ชื่อจังหวัด	กรุงเทพมหานคร
2.	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "hospital_list": [
3     {
4       "name": "โรงพยาบาล",
5       "name_en": "GENERAL HOSPITAL",
6       "type_th": "โรงพยาบาล",
7       "type_en": "Hospital",
8       "phone": "02-02",
9       "latlng": "13.100.",
10      "area_th": "กรุงเทพและปริมณฑล",
11      "area_en": "Bangkok and vicinity",
12      "accident": false,
13      "health": false,
14      "is_ipd": false,
15      "is_opd": false,
16      "care_card": false,
17      "provider": "TPA",
18      "display": true,
19      "province_id": "0",
20      "province_name": "กรุงเทพมหานคร"
21    },
22    {
23      "name": "โรงพยาบาล",
24      "name_en": "HOSPITAL",
25      "type_th": "โรงพยาบาล",
26      "type_en": "Hospital",
27      "provider": "TPA",
28      "display": true,
29      "province_id": "30",
30      "province_name": "ปัตตานี"
31    }
32  ],
33  "status": 1
34 }

```

ภาพที่ 6.59 response ที่ได้รับกลับมาจาก API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่สถานพยาบาล

ตารางที่ 6.27 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่อยู่ซ่อมรถ

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	garage_list	กลุ่มของข้อมูลอยู่	"garage_list": [กลุ่มข้อมูล 1.1.-1.9.]
1.1	name	ชื่ออยู่ซ่อมรถ	สิรินาย
1.2	phone	เบอร์โทรศัพท์อยู่ซ่อมรถ	022771232,022773314

1.3	fax	เบอร์แฟกซ์	026918770
1.4	address	ที่อยู่ช่อมรถ	56 ซ.วิภาวดีรังสิต6 ถ.วิภาวดีรังสิต แขวงดินแดง
1.5	observe_point	จุดนัดพบ	เข้าซ.วิภาวดีรังสิต6 ประมาณ 100 ม. อยู่ด้
1.6	emcs	* ขึ้นกับบริษัทประกันภัย	ไม่มี
1.7	latlng	ละติจูด,ลองติจูด ของอยู่ ช่อม	13.784449,100.5605689
1.8	province_id	ไอดีจังหวัด	0
1.9	district_id	ไอดีตำบล	37
2	status	ตัวเลขบอกสถานะ	1

```

1 - {}
2 - "garage_list": [
3 - {
4 -   "name": "สิริ...",
5 -   "phone": "02...,02...",
6 -   "fax": "02...",
7 -   "address": "56 ซ. ..., แขวงดินแดง เขตดินแดง",
8 -   "observe_point": "เข้าซ. ... ประมาณ 100 ม. อยู่ด้านซ้ายมือ",
9 -   "emcs": "ไม่มี",
10 -  "latlng": "13. ..., 100. ...",
11 -  "province_id": "0",
12 -  "district_id": "37"
13 - },
14 - {
15 -   "name": "ขลุ...",
16 -   "phone": "02...,0889...",
17 -   "fax": "02...",
18 -   "latlng": "19...,99...",
19 -   "province_id": "71",
20 -   "district_id": null
21 - }
22 - ],
23 - "status": 1
24 - }

```

ภาพที่ 6.60 response ที่ได้รับกลับมาจาก API สำหรับหน้าแผนที่เมื่อส่ง request เพื่อขอแผนที่อยู่ช่อมรถ

6.3.10. API สำหรับหน้าแก้ไขข้อมูลส่วนตัว

API สำหรับหน้าแก้ไขข้อมูลส่วนตัวมีรูปแบบ request สองแบบคือร้องขอข้อมูลส่วนตัวที่อยู่บนฐานข้อมูล และบันทึกข้อมูลส่วนตัวลงฐานข้อมูลแสดงดังตารางที่ 6.28 โดยการขอข้อมูลส่วนตัวกำหนด HTTP headers ดังตารางที่ 6.29 และมีรูปแบบของ response เมื่อเรียก API ขอข้อมูลส่วนตัว ดังตารางที่ 6.30 และ response ดังภาพที่ 6.61 ส่วนการบันทึกข้อมูลส่วนตัวกำหนด HTTP headers ดังตารางที่ 6.37 มีรูปแบบ response เมื่อเรียก API บันทึกข้อมูลส่วนตัวแสดงดังตารางที่ 6.38 และ response ดังภาพที่ 6.62

ตารางที่ 6.28 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัว แต่ละแบบ

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/user/viewinfo	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อดึงข้อมูลส่วนตัวปัจจุบัน
POST	api.dev.mysagi.com/public/api/v1/user/updateinfo	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อบันทึกข้อมูลส่วนตัว

ตารางที่ 6.29 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัว เพื่อดึงข้อมูลส่วนตัวปัจจุบัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kYZZicplxxxxxWpWsu8...

ตารางที่ 6.30 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อดึงข้อมูลส่วนตัวปัจจุบัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	email	อีเมลลูกค้า	test@email.com
2	title	รหัสคำนำหน้าชื่อ	39
3	name	ชื่อลูกค้า	สุทธิรัตน์
4	last_name	นามสกุลลูกค้า	กระดาษทอง
5	identity_id	เลขบัตรประชาชน	1100501182xxx
6	birthdate	วันเกิด	1994-03-24
7	gender	เพศ	ชาย
8	user_status	สถานะสมรส	โสด
9	phone	เบอร์โทรศัพท์บ้าน	023456789
10	mobile	เบอร์โทรศัพท์มือถือ	0812345678
11	position	ตำแหน่ง	Senior Developer
12	salary	เงินเดือน	30001-50000
13	emergency_phone	เบอร์ติดต่อกรณีฉุกเฉิน	0888888888
14	title_name	คำนำหน้าชื่อ	นาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15	status	ตัวเลขบอกสถานะ	1
----	--------	----------------	---

```

1 {
2   "email": "peelc[REDACTED]",
3   "title": "39",
4   "name": "พระ[REDACTED]",
5   "last_name": "[REDACTED]",
6   "identity_id": "[REDACTED]",
7   "birthdate": "19[REDACTED]",
8   "gender": "ชาย",
9   "user_status": "โสด",
10  "phone": "",
11  "mobile": "091[REDACTED]",
12  "position": "",
13  "salary": "20001 - 30000",
14  "emergency_phone": "088[REDACTED]",
15  "title_name": "นาย",
16  "status": 1
17 }

```

ภาพที่ 6.61 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อขอข้อมูลส่วนตัว

ตารางที่ 6.31 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัว เพื่อบันทึกข้อมูลส่วนตัว

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.32 รูปแบบของ attribute การส่งค่าใน HTTP body ไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัว เพื่อบันทึกข้อมูลส่วนตัว

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	email	อีเมลลูกค้า	test@email.com
2	title	รหัสค่านำหน้าชื่อ	39
3	name	ชื่อลูกค้า	สุทธิรัตน์
4	last_name	นามสกุลลูกค้า	กระดาษทอง
5	identity_id	เลขบัตรประชาชน	1100501182xxx
6	birthdate	วันเกิด	1994-03-24
7	gender	เพศ	ชาย
8	user_status	สถานะสมรส	โสด
9	phone	เบอร์โทรศัพท์บ้าน	023456789
10	mobile	เบอร์โทรศัพท์มือถือ	0812345678
11	position	ตำแหน่ง	Senior Developer

12	salary	เงินเดือน	30001-50000
13	emergency_phone	เบอร์ติดต่อกรณีฉุกเฉิน	0888888888
14	title_name	คำนำหน้าชื่อ	นาย
15	status	ตัวเลขบอกสถานะ	1

ตารางที่ 6.33 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูลส่วนตัว

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	ข้อความอธิบาย	ทำการบันทึกข้อมูลเรียบร้อยแล้วค่ะ
2	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "message": "ทำการบันทึกข้อมูลเรียบร้อยแล้ว",
3   "status": 1
4 }
```

ภาพที่ 6.62 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูลส่วนตัวเรียบร้อยแล้ว

6.3.11. API สำหรับแก้ไขข้อมูลที่อยู่อาศัย

API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัย มีรูปแบบการร้องขอสองแบบคือร้องขอข้อมูลที่อยู่อาศัยที่อยู่บนฐานข้อมูล และบันทึกข้อมูลที่อยู่อาศัยลงฐานข้อมูลแสดงดังตารางที่ 6.34 โดยการขอข้อมูลที่อยู่อาศัยกำหนด HTTP headers ดังตารางที่ 6.35 และมีรูปแบบของ response เมื่อเรียก API ขอข้อมูลที่อยู่อาศัยดังตารางที่ 6.36 ผลลัพธ์ดังภาพที่ 6.63 การบันทึกขอข้อมูลที่อยู่อาศัยกำหนด HTTP headers ดังตารางที่ 6.37 และมีรูปแบบการร้องขอเรียก API บันทึกขอข้อมูลที่อยู่อาศัยแสดงดังตารางที่ 6.38 และผลลัพธ์ดังภาพที่ 6.64

ตารางที่ 6.34 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัย

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/user/viewaddr	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อดึงข้อมูลที่อยู่อาศัย
POST	api.dev.mysagi.com/public/api/v1/user/updateaddress	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเพื่อบันทึกข้อมูลที่อยู่อาศัย

ตารางที่ 6.35 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเพื่อดึงข้อมูลที่อยู่อาศัยปัจจุบัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.36 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเมื่อดึงข้อมูลที่อยู่อาศัยปัจจุบัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	ads_no	บ้านเลขที่	7/6
2	ads_village	หมู่บ้าน	ลัดดาแลนด์
3	ads_moo	หมู่	1
4	ads_soi	ซอย	ลัดดาแลนด์
5	ads_road	ถนน	รังสิต-นครนายก
6	ads_subdistrict	ไอดีตำบล	3
7	ads_district	ไอดีอำเภอ	9
8	ads_province	ไอดีจังหวัด	5
9	ads_postcode	รหัสไปรษณีย์	40160
10	ads_province_name	ชื่อจังหวัด	ขอนแก่น
11	ads_district_name	ชื่ออำเภอ	มัญจาคีรี
12	ads_subdistrict_name	ชื่อตำบล	โพนเพ็ก
13	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "ads_no": " ",
3   "ads_village": " ",
4   "ads_moo": " ",
5   "ads_soi": " ",
6   "ads_road": " ",
7   "ads_subdistrict": "3",
8   "ads_district": "9",
9   "ads_province": "5",
10  "ads_postcode": "40160",
11  "ads_province_name": "ขอนแก่น",
12  "ads_district_name": "มัญจาคีรี",
13  "ads_subdistrict_name": "",
14  "status": 1
15 }

```

ภาพที่ 6.63 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเมื่อดึงข้อมูลที่อยู่อาศัย

ตารางที่ 6.37 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเพื่อบันทึกข้อมูลที่อยู่อาศัย

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.38 รูปแบบของ attribute ที่ใช้ส่งค่าใน HTTP body ไปยัง API สำหรับหน้าแก้ไขข้อมูลที่อยู่อาศัยเพื่อบันทึกข้อมูลที่อยู่อาศัย

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	ads_no	อีเมลลูกค้า	test@email.com
2	ads_village	รหัสค่านำหน้าชื่อ	39
3	ads_moo	ชื่อลูกค้า	สุทธิรัตน์
4	ads_soi	นามสกุลลูกค้า	กระดาศทอง
5	ads_road	เลขบัตรประชาชน	1100501182xxx
6	ads_subdistrict	วันเกิด	1994-03-24
7	ads_district	เพศ	ชาย
8	user_status	สถานะสมรส	โสด
9	ads_province	เบอร์โทรศัพท์บ้าน	023456789
10	ads_postcode	เบอร์โทรศัพท์มือถือ	0812345678

ตารางที่ 6.39 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูลส่วนตัว

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	ข้อความอธิบาย	ทำการบันทึกข้อมูลเรียบร้อยแล้วค่ะ
2	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "message": "ทำการบันทึกข้อมูลเรียบร้อยแล้ว",
3   "status": 1
4 }
```

ภาพที่ 6.64 response ที่ได้รับกลับมาจาก API สำหรับหน้าแก้ไขข้อมูลส่วนตัวเมื่อบันทึกข้อมูลส่วนตัวเรียบร้อยแล้ว

6.3.12. API สำหรับตั้งค่า

API สำหรับหน้าสำหรับตั้งค่า มีรูปแบบการร้องขอ 2 แบบคือร้องขอข้อมูลการตั้งค่าที่อยู่บนฐานข้อมูล และบันทึกการตั้งค่าการแจ้งเตือนลงฐานข้อมูลแสดงดังตารางที่ 6.40 โดยการขอข้อมูลการตั้งค่ากำหนด HTTP headers ดังตารางที่ 6.41 และมีรูปแบบของ response เมื่อเรียก API ขอข้อมูลการตั้งค่าดังตารางที่ 6.42 ผลลัพธ์ภาพที่ 6.65 และการบันทึกการตั้งค่าการแจ้งเตือน กำหนด HTTP headers ดังตารางที่ 6.43 และมีรูปแบบการร้องขอเรียก API บันทึกการตั้งค่าการแจ้งเตือนแสดงดังตารางที่ 6.44 และผลลัพธ์ดังรูปที่ภาพที่ 6.66

ตารางที่ 6.40 รูปแบบ request และ HTTP Verb เพื่อส่งไปยัง API สำหรับหน้าตั้งค่า

กริยา HTTP	URL	คำอธิบาย
GET	api.dev.mysagi.com/public/api/v1/user/setting	ใช้ HTTP Get ในการร้องขอไปยัง API สำหรับขอข้อมูลการตั้งค่า
POST	api.dev.mysagi.com/public/api/v1/user/notification	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับบันทึกการตั้งค่าการแจ้งเตือน

ตารางที่ 6.41 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าตั้งค่าเพื่อขอข้อมูลการตั้งค่าปัจจุบัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.42 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าตั้งค่าเพื่อขอข้อมูลการตั้งค่าปัจจุบัน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	คำอธิบาย	สำเร็จ
2	facebook_connect	เชื่อมต่อเฟซบุ๊กอยู่หรือไม่	true
3	noti_status	เปิดการตั้งค่าอยู่หรือไม่	false
4	status	ตัวเลขบอกสถานะ	1

```

1 {
2   "message": "สำเร็จ",
3   "status": 1,
4   "facebook_connect": true,
5   "noti_status": false
6 }

```

ภาพที่ 6.65 response ที่ได้รับกลับมาจาก API สำหรับหน้าตั้งค่าเพื่อขอข้อมูลการตั้งค่าปัจจุบัน

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.43 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง API สำหรับหน้าตั้งค่าเพื่อบันทึกการตั้งค่าการแจ้งเตือน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.44 รูปแบบของ attribute ภายใน response ของ API สำหรับหน้าตั้งค่าเพื่อบันทึกการตั้งค่าการแจ้งเตือน

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	ข้อความอธิบาย	อัปเดตข้อมูลแล้ว
2	status	ตัวเลขบอกสถานะ	1

```

1 - {
2   "message": "อัปเดตข้อมูลแล้ว",
3   "status": 1
4 }

```

ภาพที่ 6.66 response ที่ได้รับกลับมาจาก API สำหรับหน้าตั้งค่าเพื่อบันทึกการตั้งค่าการแจ้งเตือน

6.3.13. API สำหรับแสดงความคิดเห็น

API สำหรับแสดงความคิดเห็นกำหนดรูปแบบ request แสดงดังตารางที่ 6.45 กำหนด HTTP headers ดังตารางที่ 6.46 รูปแบบของ response ดังตารางที่ 6.47 และผลลัพธ์ดังภาพที่ 6.67

ตารางที่ 6.45 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง สำหรับแสดงความคิดเห็น

กริยา HTTP	URL	คำอธิบาย
POST	api.dev.mysagi.com/public/api/v1/user/notification	ใช้ HTTP Post ในการร้องขอไปยัง API สำหรับบันทึกการตั้งค่าการแจ้งเตือน

ตารางที่ 6.46 รูปแบบของ Headers ที่จำเป็นในการส่งค่าไปยัง สำหรับแสดงความคิดเห็น

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	Authorization	access_token ที่ได้จากการล็อกอินหรือสมัครสมาชิก	Bearer 5kJYZzicplxxxxxWpWsu8...

ตารางที่ 6.47 รูปแบบของ attribute ภายใน response ของ API สำหรับแสดงความคิดเห็น

ลำดับ	ชื่อ Attribute	คำอธิบาย	ตัวอย่าง
1	message	คำอธิบาย	ทำการส่งข้อเสนอแนะของท่านเรียบร้อยแล้ว
2	status	เลขบอกสถานะ	1

```

1 {
2   "message": "ทำการส่งข้อเสนอแนะของท่านเรียบร้อยแล้ว",
3   "status": 1
4 }

```

ภาพที่ 6.67 response ที่ได้รับกลับมาจาก API สำหรับแสดงความคิดเห็น

6.4. ผลลัพธ์การทำงานของส่วนต่อประสานโปรแกรมประยุกต์เมื่อเกิดข้อผิดพลาด

6.4.1. API สำหรับหน้าล็อกอิน

หากไม่พบชื่อผู้ใช้หรือรหัสผ่านไม่ถูกต้องดังภาพที่ 6.68

```

1 {
2   "message": "กรุณาตรวจสอบใหม่อีกครั้ง",
3   "status": 0
4 }

```

ภาพที่ 6.68 response ที่ได้กลับมาเมื่อไม่พบชื่อผู้ใช้หรือรหัสไม่ถูกต้องเมื่อมีการเรียกใช้ API สำหรับล็อกอิน

6.4.2. API สำหรับสมัครสมาชิก

มีการตรวจสอบข้อมูลที่จำเป็นในการสมัครสมาชิก หากพบความผิดพลาดจะได้ผลลัพธ์ดังภาพที่ 6.69

```

1 {
2   "message": "กรุณากรอกเบอร์โทรศัพท์ของท่านคะ",
3   "status": 0
4 }
1 {
2   "message": "กรุณากรอกชื่อของท่านคะ",
3   "status": 0
4 }
1 {
2   "message": "กรุณากรอกนามสกุลของท่านคะ",
3   "status": 0
4 }
1 {
2   "message": "กรุณากรอกรหัสผ่านของท่านคะ",
3   "status": 0
4 }

```

ภาพที่ 6.69 response ที่ได้กลับมาเมื่อไม่พบข้อมูลที่จำเป็นในการสมัครสมาชิกเมื่อมีการเรียกใช้ API สำหรับสมัครสมาชิก

มีการตรวจสอบข้อมูลเบอร์โทรศัพท์ หากมีการใช้งานไปแล้ว รวมถึงรูปแบบของเบอร์โทรศัพท์ หากพบข้อผิดพลาดจะแสดงผลดังภาพที่ 6.70

```
1 {
2   "message": "เบอร์โทรศัพท์ไม่ได้ถูกใช้งานไปแล้ว",
3   "status": 0
4 }
1 {
2   "message": "เบอร์โทรศัพท์ของท่านควรมี 10 หลัก กรุณาตรวจสอบอีกครั้งค่ะ",
3   "status": 0
4 }
```

ภาพที่ 6.70 response ที่ได้กลับมาเมื่อตรวจพบการลงทะเบียนเบอร์โทรศัพท์ซ้ำ หรือรูปแบบเบอร์โทรศัพท์ผิดปกติเมื่อมีการเรียกใช้ API สำหรับสมัครสมาชิก

มีการตรวจสอบรูปแบบอีเมล หากพบความผิดพลาดจะได้ผลลัพธ์ดังภาพที่ 6.71

```
1 {
2   "message": "อีเมลของท่านไม่ถูกต้อง กรุณาตรวจสอบอีเมลของท่านอีกครั้งค่ะ",
3   "status": 0
4 }
```

ภาพที่ 6.71 response ที่ได้กลับมาเมื่อตรวจพบรูปแบบเบอร์อีเมลผิดปกติเมื่อมีการเรียกใช้ API สำหรับสมัครสมาชิก

6.4.3. API สำหรับกู้รหัสผ่าน

การตรวจสอบข้อมูลที่ใช้ในการยืนยันตัวตนเพื่อกู้รหัสผ่าน หากไม่พบเบอร์โทรศัพท์หรืออีเมลที่ลงทะเบียนไว้ จะแสดงผลลัพธ์ดังภาพที่ 6.72

```
1 {
2   "message": "กรุณาตรวจสอบอีกครั้ง",
3   "status": 0
4 }
```

ภาพที่ 6.72 response ที่ได้กลับมาเมื่อไม่พบข้อมูลสำหรับยืนยันตัวตนในระบบเมื่อมีการเรียกใช้ API สำหรับกู้รหัสผ่าน

6.4.4. API สำหรับหน้าหลัก

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพธ์ดังภาพที่ 6.73

```
1 {
2   "error": "access_denied",
3   "error_description": "The resource owner or authorization server denied the request."
4 }
```

ภาพที่ 6.73 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบเมื่อมีการเรียกใช้ API สำหรับหน้าหลัก

6.4.5. API สำหรับหน้าแสดงรายละเอียดโปรโมชัน

มีการตรวจสอบสิทธิ์การเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลผิดพลาดดังภาพที่ 6.74

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.74 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าแสดงรายละเอียดโปรโมชัน

มีการตรวจสอบข้อมูลโปรโมชัน หากไม่พบรายละเอียดของโปรโมชันในระบบที่อ้างอิงด้วย id จะแสดงผลผิดพลาดดังภาพที่ 6.75

```
1 {  
2   "message": "Unsuccessfully. not found timeline card for id: 12.",  
3   "status": 0  
4 }
```

ภาพที่ 6.75 response ที่ได้กลับมาเมื่อไม่พบรายละเอียดของโปรโมชันในระบบที่อ้างอิงด้วย id เมื่อมีการเรียกใช้ API สำหรับหน้าแสดงรายละเอียดโปรโมชัน

6.4.6. API สำหรับหน้าซื้อประกันออนไลน์

มีการตรวจสอบสิทธิ์การเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลผิดพลาดดังภาพที่ 6.76

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.76 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าซื้อประกันออนไลน์

มีการตรวจสอบข้อมูลผลิตภัณฑ์ หากไม่พบรายละเอียดของข้อมูลผลิตภัณฑ์ในระบบที่อ้างอิงด้วย id จะแสดงผลผิดพลาดดังภาพที่ 6.77

```
1 {  
2   "message": "Unsuccessfully. not found timeline card for id: 12.",  
3   "status": 0  
4 }
```

ภาพที่ 6.77 response ที่ได้กลับมาเมื่อไม่พบรายละเอียดของข้อมูลผลิตภัณฑ์ในระบบที่อ้างอิงด้วย id เมื่อมีการเรียกใช้ API สำหรับหน้าซื้อประกันออนไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4.7. API สำหรับหน้าบัตรประกันของฉัน

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพธ์ดังภาพที่ 6.78

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.78 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าบัตรประกันของฉัน

6.4.8. API สำหรับหน้าบัตรประกันจำลอง

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพธ์ดังภาพที่ 6.79

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.79 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าบัตรประกันจำลอง

มีการตรวจสอบข้อมูลบัตรประชาชนหากลูกค้ายังไม่มีการยืนยันเลขบัตรประชาชนในระบบจะแสดงผลลัพธ์ดังภาพที่ 6.80

```
1 {  
2   "message": "Unsuccessfully. Not found or something wrong with identity id. Please check your infomation.",  
3   "status": 0  
4 }
```

ภาพที่ 6.80 response ที่ได้กลับมาเมื่อไม่พบการยืนยันบัตรประชาชนเมื่อมีการเรียกใช้ API สำหรับหน้าบัตร

มีการตรวจสอบข้อมูลกรมธรรม์ในระบบซึ่งอ้างอิงด้วยประเภทของกรมธรรม์ (type) หากไม่พบกรมธรรม์ของลูกค้าที่มีประเภทเดียวกับที่ร้องขอข้อมูล จะแสดงผลลัพธ์ดังภาพที่ 6.81

```
1 {  
2   "message": "Unsuccessfully. Not found any insurance for this identity id.",  
3   "status": 0  
4 }
```

ภาพที่ 6.81 response ที่ได้กลับมาเมื่อไม่พบประเภทของกรมธรรม์ที่ตรงกับบัตรประชาชนเมื่อมีการเรียกใช้ API สำหรับหน้าบัตรประกันจำลอง

6.4.9. API สำหรับหน้าแก้ไขข้อมูลส่วนตัว

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพท์ดังภาพที่ 6.82

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.82 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับหน้าแก้ไขข้อมูลส่วนตัว

6.4.10. API สำหรับแก้ไขข้อมูลที่อยู่อาศัย

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพท์ดังภาพที่ 6.83

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.83 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับแก้ไขข้อมูลที่อยู่อาศัย

6.4.11. API สำหรับตั้งค่า

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพท์ดังภาพที่ 6.84

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.84 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ เมื่อมีการเรียกใช้ API สำหรับสำหรับตั้งค่า

6.4.12. API สำหรับแสดงความคิดเห็น

มีการตรวจสอบสิทธิการเข้าถึงด้วย access_token ในการใช้งาน API หากไม่พบหรือไม่ตรงกับ access_token บนระบบจะแสดงผลลัพท์ดังภาพที่ 6.85

```
1 {  
2   "error": "access_denied",  
3   "error_description": "The resource owner or authorization server denied the request."  
4 }
```

ภาพที่ 6.85 response ที่ได้กลับมาเมื่อไม่พบข้อมูล access_token สำหรับยืนยันตัวตนในระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเรียกใช้ API สำหรับแสดงความคิดเห็นไปประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการวิจัยและข้อเสนอแนะ

จากการดำเนินงานพัฒนางานวิจัยระบบประกันภัยบนโมบายล์แพลตฟอร์ม ตลอดจนถึงสิ้นสุดระยะเวลาโครงการสหกิจศึกษา รวมถึงผลการวิจัยที่ได้สามารถสรุปผลให้เห็นถึงภาพรวม ผลลัพธ์ ประโยชน์ที่เกิดขึ้นจากการพัฒนางานวิจัยชิ้นนี้ รวมถึงปัญหาที่พบระหว่างขั้นตอนการพัฒนางานวิจัย และข้อเสนอแนะในแง่มุมต่างๆ ที่ผู้วิจัยเห็นว่าสามารถปรับแก้ได้พร้อมแนวทางการแก้ไข โดยสรุปเป็นหัวข้อดังต่อไปนี้

7.1. บทสรุปการวิจัย

ระบบประกันภัยบนโมบายล์แพลตฟอร์มที่ได้พัฒนา ภายใต้กรอบระยะเวลาการฝึกงานสหกิจ ให้ผลลัพธ์ที่น่าพึงพอใจ สามารถทำงานตอบสนองตรงกับความต้องการของบริษัทประกันได้เป็นอย่างดี ในส่วนการออกแบบ สามารถตอบสนองความต้องการของบริษัทประกันและลูกค้าของบริษัทประกันได้ตามเป้าหมาย การออกแบบเป็นไปตามหลักของ Material Design ซึ่งเป็นหลักการออกแบบที่ในเรื่องของประสบการณ์การใช้งาน เหมาะสมกับการใช้งานบนโทรศัพท์มือถือ เป็นการออกแบบที่กำลังได้รับความนิยมในปัจจุบัน ซึ่งสามารถตอบสนองความต้องการทั่วไปของบริษัทประกันได้แม้จะไม่สามารถทดแทนการให้บริการแบบเดิมได้ทั้งหมด แต่ก็ช่วยลดขั้นตอนการให้บริการในมุมมองของบริษัทประกัน รวมถึงในมุมมองของลูกค้าก็สามารถลดขั้นตอนการให้บริการไปได้อย่างมาก

การพัฒนาโมบายล์แอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ สามารถทำงานตอบสนองความต้องการในการให้บริการประกันภัยได้เป็นอย่างดี โมบายล์แอปพลิเคชันเป็นส่วนสำคัญอย่างมากในงานวิจัยระบบประกันภัยบนโมบายล์แพลตฟอร์ม เพราะเป็นส่วนที่จะต้องติดต่อกับผู้ใช้งาน นั่นคือลูกค้าบริษัทประกันภัยโดยตรง และต้องติดต่อกับระบบเว็บเซิร์ฟเวอร์เพื่อเป็นสื่อกลางระหว่างบริษัทประกันภัยและลูกค้าเพื่อให้บริการ บริการด้านประกันภัยต่างๆ เช่น การซื้อกรมธรรม์ การดูรายละเอียดกรมธรรม์ การใช้งานบัตรประกันจำลอง การค้นสถานพยาบาล และศูนย์ซ่อมในเครือ

การพัฒนาเว็บเซิร์ฟเวอร์ทำได้ตรงตามเป้าหมาย สามารถให้บริการประกันภัยซึ่งทำงานร่วมกับโมบายล์แอปพลิเคชันได้เป็นอย่างดี การพัฒนาเว็บเซิร์ฟเวอร์ใช้ภาษา PHP และลาลาเวลเฟรมเวิร์ค ซึ่งมีคุณสมบัติที่เพียงพอและเหมาะสม ในการพัฒนาในแง่เวลาในการพัฒนาที่จำกัด ประสิทธิภาพที่ต้องการ และสามารถตอบสนองความต้องการของระบบได้เป็นอย่างดี รวมถึงสามารถทำงานร่วมกับระบบฐานข้อมูลได้หลากหลายชนิด ทำให้สามารถรวมการทำงานเข้ากับระบบฐานข้อมูลเดิมของบริษัทประกันได้ง่ายยิ่งขึ้น มีการประเมินประสิทธิภาพการทำงานของเว็บเซิร์ฟเวอร์เพื่อแก้ไขกระบวนการทำงานหากมีความล่าช้าในการประมวลผล

7.2. ปัญหาที่พบในระหว่างการทำงาน

- ไม่เข้าใจกระบวนการในการทำงานประสานกันระหว่างนักออกแบบและนักพัฒนาในช่วงแรก
- ขาดประสบการณ์ในการใช้งานโปรแกรมออกแบบแอปพลิเคชัน (Sketch) เนื่องจากเป็นโปรแกรมที่ทางบริษัทนำเข้ามาใช้ใหม่ ทำให้ต้องใช้เวลาในการเรียนรู้สักกระยะหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เนื่องด้วยโปรแกรมที่ใช้ในการออกแบบมีข้อจำกัดเรื่องฮาร์ดแวร์ คือสามารถใช้ได้บนเครื่องแมคบุ๊ก (MacBook) เท่านั้น จึงทำให้ต้องขอยืมใช้เครื่องที่บริษัท
- การเปลี่ยนแปลงความต้องการของลูกค้าที่มีบ่อยครั้งมาก ทำให้ช่วงเวลาในการออกแบบแอปพลิเคชันค่อนข้างกินเวลามาก ส่งผลให้เกิดความล่าช้าในการพัฒนาแอปพลิเคชันจริง
- การจำลองสภาพแวดล้อมในการพัฒนาระบบหลังบ้านและ API ที่มีความยุ่งยาก ซับซ้อน และขั้นตอนที่ค่อนข้างมาก
- การพัฒนาระบบหลังบ้านและ API ร่วมกับทีมพัฒนาระบบของบริษัทประกัน ยังมีความล่าช้าในการสื่อสารอยู่
- กระบวนการออกแบบและวิเคราะห์ความต้องการของลูกค้า ใช้เวลามากทำให้ส่วนการพัฒนาระบบหลังบ้านและ API รวมถึงส่วนการพัฒนาโมบายล์แอปพลิเคชันล่าช้าตามไปด้วย
- ในการพัฒนาระบบหลังบ้านและ API จะต้องมีการเชื่อมต่ออินเทอร์เน็ตตลอดเวลา ทำให้บางครั้งไม่สามารถพัฒนาระบบได้อย่างต่อเนื่อง เนื่องจากความเสถียรของระบบเครือข่ายอินเทอร์เน็ต

7.3. แนวทางการแก้ไข

- พยายามศึกษา ทดลองใช้งานโปรแกรม และทำความเข้าใจตัวอย่างที่ได้มาจากผู้เชี่ยวชาญ ทำให้สามารถนำไปปรับและประยุกต์ใช้กับงาน ให้ตรงความต้องการของตนเองและลูกค้าได้รวดเร็วขึ้น
- ศึกษาหาความรู้จากอินเทอร์เน็ตเพิ่มเติม เมื่อเกิดความไม่เข้าใจในการใช้งานโปรแกรมการโค้ด โดยหลักๆแล้วจะศึกษาจากทางออนไลน์คอร์สที่บางบริษัทซื้อไว้ให้
- ต้องมีความพยายาม และความสามารถในการค้นหาข้อมูลด้วยกูเกิ้ล (Google) ซึ่งจำเป็นมากต่อนักออกแบบและนักพัฒนา
- หาแรงบันดาลใจจากงานของผู้เชี่ยวชาญ เพื่อนำแนวคิดใหม่ๆไปประยุกต์ใช้ในการออกแบบและพัฒนาแอปพลิเคชัน
- ในการจำลองสภาพแวดล้อมในการพัฒนาระบบ สามารถใช้ซอฟต์แวร์ Vagrant เพื่อจัดการสภาพแวดล้อมแทนการจำลองสภาพแวดล้อมด้วยตนเอง
- ใช้การพัฒนาและทดสอบบนเครื่องคอมพิวเตอร์ แทนการพัฒนาและทดสอบบนเครื่องแม่ข่ายโดยตรงเพื่อหลีกเลี่ยงปัญหาอินเทอร์เน็ตไม่เสถียร

7.4. แนวทางการพัฒนาต่อและนำไปใช้

- พัฒนาแอปพลิเคชันให้มียูสเซอร์อินเตอร์เฟซ (User Interface) ที่ใช้งานง่ายขึ้น และพัฒนาฟีเจอร์อื่น ๆ ที่ได้ออกแบบเข้าไปอีก เช่น การถ่ายรูปจุดต่างๆของรถ การสแกนคิวอาร์โค้ดเพื่อให้ข้อมูลประกันแก่แพทย์ เป็นต้น
- สามารถนำระบบหลังบ้านและ API ไปประยุกต์ใช้กับบริษัทประกันภัยอื่นๆได้ รวมถึงระบบที่มีความคล้ายคลึงกันเช่น ระบบบริหารจัดการบิลค่าบริการ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้า

- เพิ่มบริการอื่นๆ เข้าไปในระบบเพื่อตอบสนองตามความต้องการ ขงบริษัทประกันและลูกค้าของบริษัทประกัน



เอกสารอ้างอิง

Octagon Interactive. (2559). [ออนไลน์].

แหล่งที่มา: <http://www.8interactive.co.th> [4 ตุลาคม 2559].

Medium. (2559). **DESIGN THINKING คืออะไร (OVERVIEW)**. [ออนไลน์].

แหล่งที่มา: <https://medium.com/@dexstudioth/design-thinking-คืออะไร-overview-dc8c8e7547db#.bzt7yz3v9> [5 ตุลาคม 2559].

Storylog. (2558). **"Design Thinking" – กระบวนการออกแบบนวัตกรรม**. [ออนไลน์].

แหล่งที่มา: <https://storylog.co/story/56a321f8f69f51246bce4045> [5 ตุลาคม 2559].

Behavioural Design Lab. (2559). **Approach — Behavioural Design Lab**. [online].

แหล่งที่มา: <http://www.behaviouraldesignlab.org/work/approach> [5 Oct. 2559].

d.school. (2559). **Welcome to the Virtual Crash Course in Design Thinking**.

[online].

แหล่งที่มา: <http://dschool.stanford.edu/dgift> [5 Oct. 2559].

TCDC. (2559). **TCDC : Design Thinking for Creative Business [เก็บตกเวิร์คชอป]**.

[ออนไลน์]

แหล่งที่มา: [http://www.tcdc.or.th/articles/design-creativity/20302/#Design-Thinking-for-Creative-Business-\[เก็บตกเวิร์คชอป\]](http://www.tcdc.or.th/articles/design-creativity/20302/#Design-Thinking-for-Creative-Business-[เก็บตกเวิร์คชอป]) [5 ตุลาคม 2559]

River Park. (2559). **SDLC (Software Development Life Cycle) using Agile**. [ออนไลน์]

แหล่งที่มา: <http://www.riverpark.co.th/blog/agilesdlc.html> [6 ตุลาคม 2559].

Agilemethodology (2008). **The Agile Movement**. [online].

แหล่งที่มา: <http://agilemethodology.org> [6 Oct. 2559].

Start IT Up. (2558). **MVP เรื่องที่คนทำสตาร์ทอัพควรรู้**. [ออนไลน์].

แหล่งที่มา: <http://startitup.in.th/mvp-the-startup-needs-to-know/> [8 ตุลาคม 2559].

เอกสารอ้างอิง (ต่อ)

iTron. (2558). จะสร้าง Minimum Viable Product ได้อย่างไร. [ออนไลน์].

แหล่งที่มา: <http://www.itorn.net/2557/11/02/จะสร้าง-minimum-viable-product-ได้อย่างไร> [8 ตุลาคม 2559].

Uiblogazine. (2557). ทำน้อย สำเร็จได้ (Minimum Viable Product).[ออนไลน์].

แหล่งที่มา: <http://www.uiblogazine.com/design-excite/ทำน้อย-สำเร็จได้-minimum-viable-product> [8 ตุลาคม 2559].

Leanstack. (2559). Minimum Viable Product. [online].

แหล่งที่มา: <https://leanstack.com/minimum-viable-product> [8 Oct. 2559].

Quora. (2557). What is a minimum viable product?. [online].

แหล่งที่มา: <https://www.quora.com/What-is-a-minimum-viable-product> [8 Oct. 2559].

UX Academy. (2559). Hooked by Nir Eyal . [ออนไลน์].

แหล่งที่มา: <https://blog.uxacademy.in.th//hooked-by-nir-eyal-ตอนที่-1-8e378518e4c1#jj8e15lw> [11 ตุลาคม 2559].

Tech Sauce. (2559). 4 ขั้นตอน Hook กระตุ้นให้ผู้ใช้ติดหนึบกับผลิตภัณฑ์ของคุณ. [ออนไลน์].

แหล่งที่มา: <http://techsauce.co/startup101/4-steps-of-hook-to-design-engaging-product> [11 ตุลาคม 2559].

True Incube. (2557). Hooked: สร้างสินค้าอย่างไรให้คนใช้ “ติดหนึบ”. [ออนไลน์].

แหล่งที่มา: <https://trueincube.wordpress.com/2557/06/20/hooked-by-nir-ayal> [11 ตุลาคม 2559].

Nir & Far (2554). Behavior Can Be Designed. [online].

แหล่งที่มา: <http://www.nirandfar.com/> [11 Oct. 2559].

Material Design. (2557). [online].

แหล่งที่มา: <https://material.google.com> [12 Oct. 2559].

NuuNeoi. (2557). Material Design ภาษาดีไซน์ใหม่จาก Google ทฤษฎีสู่ปฏิบัติ.[ออนไลน์].

แหล่งที่มา: https://nuuneoi.com/blog/blog.php?read_id=692 [12 ตุลาคม 2559].

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง (ต่อ)

NuuNeoi. (2559). รู้จัก Bottom Navigation ... UX แบบใหม่ที่เพิ่งเพิ่มเข้ามาในแอนดรอยด์
อย่างเป็นทางการ. [ออนไลน์].

แหล่งที่มา: https://nuuneoi.com/blog/blog.php?read_id=859 [12 ตุลาคม 2559].

Blognone. (2557). รู้จัก Material Design แนวทางการออกแบบ UI ของกูเกิลในยุคต่อไป.
[ออนไลน์].

แหล่งที่มา: <https://www.blognone.com/node/57820> [12 ตุลาคม 2559].

Sleeping For Less. (2556). [Android Design] แท้จริงแล้วหน่วย dp คืออะไร?. [ออนไลน์].

แหล่งที่มา: <http://www.akexorcist.com/2013/03/android-design-dp.html> [12 ตุลาคม 2559].

Sleeping For Less. (2558). [Android Design] ทำไมแอนดรอยด์ต้องใช้หน่วย DP?. [ออนไลน์].

แหล่งที่มา: <http://www.akexorcist.com/2015/07/android-design-density-independent-pixels.html> [12 ตุลาคม 2559].

Brain Hub. (2557). What Is The Difference Between Wireframe, Mockup and
Prototype?. [online].

แหล่งที่มา: <https://brainhub.eu/blog/2016/04/20/difference-between-wireframe-mockup-prototype> [18 Oct. 2559].

invision. (2558). MASTERING YOUR PERFECT DESIGN PROCESS. [online].

แหล่งที่มา: <http://blog.invisionapp.com/mastering-your-perfect-design-process> [18 Oct. 2559].

Blognone. (2558). แนะนำ Slack โดะกลางวางแผน สื่อประสานทุกโปรเจกต์ให้ลุล่วง. [ออนไลน์].

แหล่งที่มา: <https://www.blognone.com/node/65551> [10 พฤษภาคม 2559].

digimolek.com. (2558). Slack แอปคุยในองค์กรระดับหมื่นล้าน ที่คิดนอกกรอบ และมาจากการ
ที่คาดไม่ถึง. [ออนไลน์]

แหล่งที่มา: <http://www.digimolek.com/2015/05/31/slack-billionaire-app> [10 พฤษภาคม 2559]

เอกสารอ้างอิง (ต่อ)

Slack. (2558). [online]

แหล่งที่มา: <https://slack.com> [10 พฤศจิกายน. 2559].

designil. (2558). **แนะนำ Sketch 3 โปรแกรมออกแบบ UI Design สุดฮิตที่ใช้กันทั่วโลก!!**. [ออนไลน์].

แหล่งที่มา: <http://www.designil.com/sketch-3-ui-design.html> [11 พฤศจิกายน 2559].

Bow Kraivanich. (2558). **แนะนำ Sketch 3 แอปสำหรับออกแบบเว็บไซต์ หรือ user interface ที่ใช้ง่ายมากๆ**. [ออนไลน์].

แหล่งที่มา: <http://bowkraivanich.com/sketch-3-preview> [11 พฤศจิกายน 2559].

Medium. (2557). **What is new in Sketch 3**. [online].

แหล่งที่มา: <https://medium.com/sketch-app/what-is-new-in-sketch-3-4b92d8b25f3#segx0zb2k> [11 พฤศจิกายน. 2559].

Sketch. (2559). [online].

แหล่งที่มา: <https://www.sketchapp.com> [11 พฤศจิกายน. 2559].

Zeplin. (2559). [online].

แหล่งที่มา: <https://zeplin.io> [11 พฤศจิกายน. 2559].

Sleeping For Less. (2559). **[Mobile App Design] มารู้จักกับ Prototyping Tools สำหรับงาน Mobile Application กันเถอะ**. [ออนไลน์].

แหล่งที่มา: <http://www.akexorcist.com/2016/02/prototyping-tools-the-fast-way-to-get-started-the-mobile-app-development.html> [11 พฤศจิกายน 2559].

Flinto. (2559). [online].

แหล่งที่มา: <https://zeplin.io> [11 พฤศจิกายน. 2559].

กอบเกียรติ สระอุบล. (2556). **พัฒนา App Android**. กรุงเทพมหานคร: มีเดีย เนทเวิร์ก.

Eduzones. (2558). **สถาปัตยกรรมของ Android**. [ออนไลน์]

แหล่งที่มา : <http://blog.eduzones.com/cazii/84053> [25 พฤศจิกายน 2559].

เอกสารอ้างอิง (ต่อ)

- Utramod. 2552. **ความรู้เบื้องต้นภาษาจาวา**. [ออนไลน์]
แหล่งที่มา : <http://happyerevrytime.exteen.com/20090608/entry> [25 พฤษภาคม 2559].
- Sleeping For Less. (2559) . [Android Code] **Fragment Principle**. [ออนไลน์]
แหล่งที่มา : <http://www.akexorcist.com/2014/10/android-fragment-principle-fragment.html> [25 พฤษภาคม 2559].
- Sleeping For Less. (2559) . [Android Code] **Let's Fragment - วงจรชีวิตของ Fragment (Fragment Lifecycle)**. [ออนไลน์]
แหล่งที่มา : <http://www.akexorcist.com/2014/11/lets-fragment-lifecycle.html> [25 พฤษภาคม 2559].
- วิกิพีเดีย สารานุกรมเสรี.(2558) **เอกซ์เอ็มแอล**. [ออนไลน์]
แหล่งที่มา : <https://th.wikipedia.org/wiki/เอกซ์เอ็มแอล> [25 พฤษภาคม 2559].
- ระบายนความรู้. (2559). **MVC แนวคิดการเขียนโปรแกรม ที่หลายคนอาจจะไม่รู้จัก**. [ออนไลน์]
แหล่งที่มา : <http://sundryanything.blogspot.com/2014/03/mvc.html> [25 พฤษภาคม 2559].
- M.Wannaporn C. **WebBlog เพื่อการเรียนรู้การสอน. (2558). การเขียนผังงาน (Flowchart)**. [ออนไลน์]
แหล่งที่มา : <https://stwannaorn.wordpress.com/2014/06/29/การเขียนผังงาน-flowchart> [14 ธันวาคม 2559].
- whoknown.com. (2557). **การออกแบบเชิงออบเจกต์**. [ออนไลน์]
แหล่งที่มา : http://www.whoknown.com/2014/02/blog-post_26.html [14 ธันวาคม 2559].
- personal information. (2557). **Flowchart(รูปแบบการเขียนผังงาน)**. [ออนไลน์]
แหล่งที่มา : <http://panuphunboonsuebsakoon.blogspot.com/2014/12/flowchart.html> [25 พฤษภาคม 2559].

เอกสารอ้างอิง (ต่อ)

git-scm.com. (2559). **Git ขั้นพื้นฐาน**. [ออนไลน์].

แหล่งที่มา : <https://git-scm.com/book/th/v1/เริ่มต้นใช้งาน-Git-ขั้นพื้นฐาน> [14 ธันวาคม 2559].

getcomposer.org. **Getting Started with Composer** [ออนไลน์].

แหล่งที่มา : <https://getcomposer.org/doc/00-intro.md#globally> [18 พฤษภาคม 2559].

Jeremy Davis, Alon Swartz. (2009, May 17). **Failed to extract file : Missing unzip command**. [ออนไลน์].

แหล่งที่มา : <https://www.turnkeylinux.org/forum/support/20090517/failed-extract-file-missing-unzip-command-fix-noobs-me> [18 พฤษภาคม 2559].

laravel.com. **Laravel 5.3 Installation**. [ออนไลน์].

แหล่งที่มา : <https://laravel.com/docs/5.3/installation> [18 พฤษภาคม 2559].

Rahul K. (2016, August 30). **How To Install Laravel 5 Framework on Ubuntu 16.04**. [ออนไลน์].

แหล่งที่มา : <https://tecadmin.net/install-laravel-framework-on-ubuntu> [18 พฤษภาคม 2559].

Saimok. (2015, June 17). **บทเรียน LARAVEL 5 การใช้งาน MIGRATIONS และ SEEDING**. [ออนไลน์].

แหล่งข้อมูล : <http://www.saimok.com/2015/06/17/thai-laravel-5-migrations-seeding/> [20 พฤษภาคม 2559].

fixer1234, RoseHosting, 0decimal0. (2015, February 2). **How to recompile the PHP with --with-pgsql**. [ออนไลน์].

แหล่งข้อมูล : <https://superuser.com/questions/607546/how-to-recompile-the-php-with-with-pgsql> [20 พฤษภาคม 2559].

Muhammad Arul. (2016). **How to Install PostgreSQL and phpPgAdmin on Ubuntu 16.04**. [ออนไลน์].

แหล่งข้อมูล : <https://www.howtoforge.com/tutorial/ubuntu-postgresql-installation/> [28 พฤษภาคม 2559].

เอกสารอ้างอิง (ต่อ)

rosehosting.com. (2016, August 24) **Install PostgreSQL with phpPgAdmin on Ubuntu 16.04.** [ออนไลน์].

แหล่งที่มา : <https://www.rosehosting.com/blog/install-postgresql-with-phpPgAdmin-on-ubuntu/> [4 ธันวาคม 2559].

confluence.atlassian.com. **Set up a repository.** [ออนไลน์].

แหล่งที่มา : <https://confluence.atlassian.com/bitbucket/set-up-a-repository-877174034.html> [4 ธันวาคม 2559].

en.wikipedia.org. (2017, March 23). **PHP.** [ออนไลน์].

แหล่งที่มา : <https://en.wikipedia.org/wiki/PHP> [23 มีนาคม 2560].

en.wikipedia.org. (2017, March 4). **Laravel.** [ออนไลน์].

แหล่งที่มา : <https://en.wikipedia.org/wiki/Laravel> [23 มีนาคม 2560].

softmelt.com. **REST (Representational State Transfer) คืออะไร.** [ออนไลน์].

แหล่งที่มา : <http://www.softmelt.com/article.php?id=134> [24 ตุลาคม 2559].

Nuttavut Thongjor. (2016, April 21). **ออกแบบ REST API ยังไงดี? แนะนำ jsonapi.** [ออนไลน์].

แหล่งที่มา : <https://www.babelcoder.com/blog/posts/rest-api-design> [24 ตุลาคม 2559].

en.wikipedia.org. (2017, March 22). **Representational state transfer.** [ออนไลน์].

แหล่งที่มา : https://en.wikipedia.org/wiki/Representational_state_transfer [23 มีนาคม 2560].

th.wikipedia.org. (2017, March 11). **ลินุกซ์.** [ออนไลน์].

แหล่งที่มา : <https://th.wikipedia.org/wiki/ลินุกซ์> [23 มีนาคม 2560].

getpostman.com. **Developing APIs is hard.Postman makes it easy.** [ออนไลน์].

แหล่งที่มา : <https://www.getpostman.com/> [18 ตุลาคม 2559].

getpostman.com. **Getting Started.** [ออนไลน์].

แหล่งที่มา : <https://www.getpostman.com/docs/> [18 ตุลาคม 2559].

เอกสารอ้างอิง (ต่อ)

en.wikipedia.org. (2017, March 18). PuTTY. [ออนไลน์].

แหล่งที่มา : <https://en.wikipedia.org/wiki/PuTTY> [23 มีนาคม 2560].

putty.org. Download PuTTY. [ออนไลน์].

แหล่งข้อมูล : <http://www.putty.org/> [17 ตุลาคม 2559].

