

แอปพลิเคชันสูงวัยมายเฟรนด์

OLD MY FRIENDS

นายนรากร วิเชียรไชย

โครงการนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

ภาควิชาคณิตศาสตร์ สติทิ และคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2561

ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงงาน : แอปพลิเคชันสูงวัยมายเฟรนด์
OLD MY FRIENDS
โดย : นายนรากร วิเชียรไชย
อาจารย์ที่ปรึกษา : ผศ.ชยaphr แก่นสาร์
ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์
ปีการศึกษา : 2561

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงงานคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา
(ผศ.ชยaphr แก่นสาร์)

..... กรรมการ
(ดร.สุภาวดี ทิรัญพงศ์สิน)

..... กรรมการ
(ดร.วิชิต สมบัติ)

..... หัวหน้าภาควิชา
(ดร.ชัชวิน นามมั่น)

วันที่/...../.....

กิตติกรรมประกาศ

การพัฒนาแอปพลิเคชันสูงวัยมายเพรนด์ (OLD MY FRIENDS) สำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลือจากหลายท่าน ข้าพเจ้าขอขอบคุณทุกท่าน ที่มีส่วนร่วมในการพัฒนาโครงการนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ผศ. ชาญพร แก่นสาร อาจารย์ที่ปรึกษาโครงการที่ได้แนะนำทฤษฎีและแนวทางในแก่ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนาระบบ อีกครั้งยังคงอยู่ตรวจสอบความก้าวหน้าของการทำงานเป็นระยะ ๆ รวมทั้งสร้างกำลังใจในการพัฒนาโครงการนี้ตลอดจนโครงการเสร็จสมบูรณ์

ขอขอบพระคุณอาจารย์ประจำสาขาวิชาการคอมพิวเตอร์ อาจารย์ประจำภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ และอาจารย์ในคณะวิทยาศาสตร์ทุก ๆ ท่าน ที่เคยให้คำแนะนำ อบรมสั่งสอน และค่อยช่วยเหลือข้าพเจ้า ในการศึกษาตลอดมา ขอบคุณเจ้าหน้าที่และบุคลากรของคณะวิทยาศาสตร์ ที่ได้อำนวยความสะดวกทางด้านอุปกรณ์และเครื่องมือต่าง ๆ

ขอกราบขอบพระคุณบิดา มารดา ที่ค่อยให้กำลังใจ ค่อยให้ความรักและความห่วงใยเสมอมา ตลอดจนค่อยช่วยเหลือทุนทรัพย์ทางด้านการศึกษาและอุปกรณ์ในการพัฒนาโครงการ

ขอบคุณเพื่อน ๆ สาขาวิชาการคอมพิวเตอร์ชั้นปีที่ 4 ที่ได้ค่อยช่วยแก่ไขปัญหาและให้คำปรึกษาในการพัฒนาโครงการครั้งนี้จนเสร็จสิ้น

นายธนากร วิเชียรไชย วันที่ 26 เมษายน 62

โครงงาน	:	แอปพลิเคชันสูงวัยมายเฟรนด์
โดย	:	นายนรากร วิเชียรไชย
อาจารย์ที่ปรึกษา	:	ผศ.ชยាមร แก่นสาร์
ระดับการศึกษา	:	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา	:	2561

บทคัดย่อ

แอปพลิเคชันสูงวัย มายเฟรนด์ (OLD MY FRIENDS) ได้ถูกพัฒนาขึ้นโดยสามารถทำงานได้ทั้งบนระบบปฏิบัติการแอนดรอยด์ (Android) และไอโอเอос (IOS) ซึ่งจะเป็นแอปพลิเคชันที่ช่วยอำนวยความสะดวกสำหรับกลุ่มผู้สูงอายุ โดยประกอบด้วยฟังก์ชันการทำงานที่จำเป็น ซึ่งได้แก่ระบบแข็งบอทเพื่อถามตอบเรื่องโรค จะช่วยให้เข้าใจในโรคที่เกิดกับผู้สูงอายุจำนวน 5 โรค โดยสามารถให้ข้อมูลสาเหตุ อาการ วิธีป้องกันและดูแลรักษาเบื้องต้นได้ และระบบยังสามารถแจ้งเตือนการทานยาได้ด้วย นอกจากนี้ยังมีส่วนให้ผู้สูงอายุสามารถโพสต์กระทู้หรือหัวข้อสนทนากัน เพื่อช่วยแชร์เรื่องราวที่สนใจกับเพื่อนหรือกลุ่มได้ ทั้งยังมีฟังก์ชันของการติดตามตำแหน่งของผู้ใช้งาน เพื่อการติดตามหรือแจ้งเตือนกรณีที่หลงทาง โดยในการพัฒนาจะใช้ ionic 3 และ dialogflow ซึ่งเป็นส่วนในการสร้างแข็งบอทเพื่อการพูดคุยโต้ตอบ โดยประโยชน์ของแอปพลิเคชันสูงวัย มายเฟรนด์ (OLD MY FRIENDS) คือช่วยให้คำแนะนำการส่งเสริมพฤติกรรมสุขภาพที่ดีผ่านระบบแข็งบอท และสร้างกลุ่มเพื่อนผ่านการพูดคุยทางสังคมออนไลน์ทำให้ช่วยลดความเหงาและสร้างกิจกรรมให้กับผู้สูงอายุได้

คำสำคัญ: แข็งบอท ไอโอนิก แอนดรอยด์ ไอโอเอส ผู้สูงอายุ

Topic	:	OLD MY FRIENDS
Author	:	MR.NARAKORN VICHIANCHAI
Advisor	:	Chayaporn Kaensar, Asst. Prof.
Degree	:	Bachelor of Science (Computer Science)
Academic Year	:	2018

Abstract

OLD MY FRIENDS is a cross-platform mobile application to facilitate elderly which runs on both Android operating system and iOS operating system. It contains functionality that is needed. The chatbot system to ask and answer about the disease. To help understand the diseases of the elderly 5 disease by can provide information on the causes, symptoms, protection and basic care, Azusa Temperature data logging function private health system can alert about taking care of yourself each disease. Such as exercise, eating, etc., also can ask and answer about the weather. There is also a section for the elderly to post threads or topics. To help share the story interested with friends or groups. The function of tracking of the user to the track or alert in case of lost by using ionic 3 and development dialogflow, which is to create a chatbot to interactive talking by taking advantage of OLD MY FRIENDS. Is to give instructions to promote good health habits via chatbot. And create a group of friends through social talk online make reduce loneliness and construction activities for the elderly.

Keywords: chatbot, ionic, android, ios, elderly

สารบัญ

	หน้า
กิตติกรรมประกาศ	ค
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
สารบัญ	ฉ
สารบัญตาราง	ภ
สารบัญภาพ	ภ
 บทที่	
1 บทนำ	1
1.1 ที่มาและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	3
1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)	3
1.5.1 ฮาร์ดแวร์	3
1.5.2 ซอฟต์แวร์ (Software)	3
1.5.3 แผนการดำเนินการ	4
2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 Hybrid Mobile Application	5
2.1.1 ข้อดีของ Hybrid Mobile Application	6
2.1.2 ข้อเสียของ Hybrid Mobile Application	6
2.2 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์	7
2.2.1 ประวัติความเป็นมาของระบบปฏิบัติการแอนดรอยด์	8
2.2.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์	9
2.2.3 ข้อเด่นของระบบปฏิบัติการแอนดรอยด์	11
2.2.4 การจัดการเกี่ยวกับวัสดุจัดแอนดรอยด์	12
2.2.5 กระบวนการเริ่มทำงานของแอคทิวิตี้ (Activity)	12
2.2.6 ทำความสะอาดรู้จักกับ Lifecycle Callback	12
2.3 ความรู้พื้นฐาน Ionic Framework	14

2.3.1	เทคโนโลยีที่ใช้ในการพัฒนา Ionic Framework	14
2.3.2	การทำงานของ Cordova Application	16
2.3.3	ความแตกต่างระหว่าง PhoneGap/Cordova	17
2.3.4	เริ่มต้นการใช้งาน	17
2.4	การใช้งาน Firebase เป็นฐานข้อมูล	18
2.4.1	บริการหลักของเฟร์เบส	19
2.4.2	การพัฒนา Cloud Firestore	20
2.5	Dialogflow	22
2.5.1	เริ่มต้นใช้งาน Dialogflow	22
2.6	Libraries moment.js	27
2.7	Google Maps API	29
2.7.1	Google Maps คืออะไร	29
2.7.2	ประโยชน์ของการใช้งาน Google Maps	29
2.7.3	การนำ Google Maps API มาใช้งาน	30
2.8	เอกสารและงานวิจัยที่เกี่ยวข้อง	31
2.8.1	แอปพลิเคชันเครือข่ายสังคมออนไลน์เพื่อผู้สูงอายุ (OLDSTER)	31
2.8.2	แอปพลิเคชัน KINYA	32
3	การวิเคราะห์และออกแบบระบบ	35
3.1	โครงสร้างภาพรวมของระบบ	35
3.2	การวิเคราะห์ความต้องการของระบบ	37
3.2.1	ความต้องการหลักของระบบ (Functional Requirements)	37
3.2.2	Non-functional Requirements	37
3.3	การออกแบบส่วนติดต่อผู้ใช้	37
3.3.1	หน้าเข้าสู่ระบบ	38
3.4	แผนภาพโดยรวม	55
3.4.1	แผนภาพยูสเคส (Use Case Diagram)	55
3.4.2	แผนภาพคลาส (Class Diagram)	62
3.5	ซีเควนไ/doeagram (Sequence Diagram)	67
3.6	โครงสร้างฐานข้อมูลไฟร์เบส(Firebase Database Structure)	78
4	การพัฒนาระบบ	87
4.1	การพัฒนาเว็บแอปพลิเคชัน	87

4.1.1 การเขียนต่อ Cloud Firestore	87
4.1.2 โครงสร้างของการสร้างหน้าเข้าสู่ระบบ	89
4.1.3 โครงสร้างของการสร้างหน้าข่าวสาร	92
4.1.4 โครงสร้างของการสร้างหน้าดูรายละเอียดข่าวสาร	94
4.1.5 โครงสร้างของการสร้างหน้าสนทนา	96
4.1.6 โครงสร้างของการสร้างหน้าปฏิทินแสดงกำหนดการ	99
4.1.7 โครงสร้างของการสร้างหน้าสร้างประกาศ	101
4.1.8 โครงสร้างของการสร้างหน้าอัปโหลดเอกสารที่เกี่ยวข้อง	104
4.1.9 โครงสร้างของการสร้างหน้าสร้างกำหนดการของคิวส่งเอกสาร	106
4.2 การพัฒนาเว็บแอปพลิเคชันแอนดรอยด์แอปพลิเคชัน	107
4.2.1 โครงสร้างของการสร้างหน้า MainActivity	108
4.2.2 โครงสร้างของการสร้างหน้า FeedFragment	110
4.2.3 โครงสร้างของการสร้างหน้า PostDetailActivity	113
4.2.4 โครงสร้างของการสร้างหน้า ChatActivity	114
4.2.5 โครงสร้างของการสร้างหน้า SignInActivity	118
4.2.6 โครงสร้างของการสร้างหน้า ScheduleFragment	120
4.2.7 โครงสร้างของการสร้างหน้า ScheduleFragment	122
5 การทดสอบระบบ	123
5.1 การทดสอบการใช้งานแอนดรอยด์แอปพลิเคชัน	123
5.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน	131
6 สรุปและข้อเสนอแนะ	137
6.1 สรุปความสามารถของระบบ	137
6.1.1 เว็บแอปพลิเคชัน	137
6.1.2 แอนดรอยด์พลิเคชัน	138
6.2 ปัญหาและอุปสรรคในการพัฒนา	138
6.3 แนวทางการพัฒนาต่อ	139
ภาคผนวก	141
ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม	141
ก.1 การติดตั้ง Android Studio	141
ก.2 การติดตั้ง Node.js	144

ก.3 การติดตั้ง Vue.js Fronted Framework 147

บทที่	หน้า
ภาคผนวก ข คู่มือการติดตั้งระบบ	149
ภาคผนวก ค คู่มือการใช้งานระบบ	152
ประวัติผู้เขียน	169

สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน	4
3.1 สัญลักษณ์ของ Use case Diagram	55
3.2 อธิบาย Use Case หน้าที่ของแอปพลิเคชันสูงวัยมายเฟรนด์ในภาพที่ 3.16	57
3.3 Use Case เข้าสู่ระบบ	57
3.4 Use Case ออกจากระบบ	58
3.5 Use Case จัดการข้อมูลส่วนตัว	58
3.6 Use Case ดูเพิ่ม แก้ไข หรือลบโพสท์	58
3.7 Use Case โทรหีอส่งข้อความมาชิกครอบครัว	59
3.8 Use Case ดูตำแหน่งสมาชิกครอบครัว	59
3.9 Use Case จัดการครอบครัว	59
3.10 Use Case จัดการกลุ่ม	60
3.11 Use Case จัดการเพื่อน	60
3.12 Use Case เพิ่ม หรือลบการแจ้งเตือน	60
3.13 Use Case ดูวิธีการใช้งาน	61
3.14 Use Case คุยกับเซาฟบอท	61
3.15 สัญลักษณ์ของ Class Diagram	62
3.16 อธิบาย Class Diagram ของแอปพลิเคชันสูงวัยมายเฟรนด์	66
3.17 สัญลักษณ์ของ Sequence Diagram	67
3.18 สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Firebase	78
3.19 อธิบายโหนดที่ใช้เก็บข้อมูลประภาก	80
3.20 อธิบายโหนดที่ใช้เก็บข้อมูลเอกสารที่เกี่ยวข้อง	81
3.21 อธิบายโหนดที่ใช้เก็บข้อมูลประวัติการสนทนา	82
3.22 อธิบายโหนดที่ใช้เก็บข้อมูลกำหนดการ	83
3.23 อธิบายโหนดที่ใช้เก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา	84
3.24 อธิบายโหนดที่ใช้เก็บข้อมูลของนักศึกษา	85
3.25 อธิบายโหนดที่ใช้เก็บข้อมูลการจองคิวของนักศึกษา	86
3.26 อธิบายโหนดที่ใช้เก็บข้อมูลคำตามที่พูบบ่อຍ	86
5.1 ผลการทดสอบเมนูนำทาง	124
5.2 ผลการทดสอบเมนูนำทาง(ต่อ)	125
5.3 ผลการทดสอบหน้ารายละเอียดประกาศ	126
5.4 ผลการทดสอบหน้าสนทนา	127
5.5 ผลการทดสอบหน้าปฏิทินกำหนดการ	128
5.6 ผลการทดสอบหน้าดาวน์โหลดเอกสาร	129
5.7 ผลการทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร	130
5.8 ผลการทดสอบหน้าจองคิวส่งเอกสาร	131

5.9 ผลการทดสอบเมนูนำทาง	132
5.10 ผลการทดสอบหน้ารายละเอียดประกาศ	133
5.11 ผลการทดสอบหน้าสนทนา	134
5.12 ผลการทดสอบหน้าปฏิทินกำหนดการ	135
5.13 ผลการทดสอบหน้าดาวน์โหลดเอกสาร	136

สารบัญภาพ

รูปที่	หน้า
2.1 Hybrid Mobile Application คืออะไร	5
2.2 ส่วนแบ่งการตลาดระบบปฏิบัติการบนสมาร์ทโฟน	8
2.3 โครงสร้างของระบบปฏิบัติการแอนดรอยด์	9
2.4 วัสดุจัดของแอคทิวิตี้บนระบบปฏิบัติการแอนดรอยด์	13
2.5 การทำงานของ Ionic Framework	15
2.6 Cordova Application	16
2.7 Cordova Build Application	16
2.8 แสดงการติดตั้ง cli	18
2.9 Firebase 2.0	19
2.10 เว็บ https://firebase.google.com	20
2.11 รูปแบบการส่งข้อมูลผู้ใช้ไปยังเซทบอท	22
2.12 หน้าหลักของ Dialogflow	23
2.13 หน้า Create Agent	23
2.14 หน้า Intent ส่วน Training phrases	24
2.15 หน้า Intent ส่วน Responses	25
2.16 ทดสอบคุยกับบอท	26
2.17 ทดสอบคุยกับบอท	27
2.18 คำสั่งในการติดตั้ง moment.js	27
2.19 รูปแบบการใช้งาน moment.js	28
2.20 เรียกใช้ API KEY	30
2.21 เรียกใช้งาน google maps ใน ionic framework	31
2.22 หน้าหลักของ OLDSTER	32
2.23 หน้าหลักของ KINYA	34
3.1 System architecture แอปพลิเคชันสูงวัยมายเฟรนด์	36
3.2 หน้าจอเข้าสู่ระบบ	38
3.3 หน้าจอหลัก	39
3.4 หน้าแซทบอทคุยกับปุ่มจองที่นั่น	41
3.5 หน้าจอกระดานข่าว	42
3.6 หน้าครอบครัว	44
3.7 หน้าแสดงตำแหน่งของครอบครัว	45
3.8 หน้าฉุกเฉิน	46
3.9 หน้าเลือกครอบครัว	47
3.10 หน้าวิธีใช้งาน	48
3.11 หน้าจอแสดงกลุ่ม	49
3.12 หน้าจอแสดงเพื่อน	50

3.13	หน้าแสดงรายการแจ้งเตือนการทำงานยา	51
3.14	หน้าแสดงการเพิ่มการแจ้งเตือนทานยา	52
3.15	หน้าจอแสดงข้อมูลส่วนตัวของผู้ใช้	53
3.16	Use Case Diagram ของแอปพลิเคชันสูงวัยมายเฟรนด์	56
3.17	Class Diagram ของแอปพลิเคชันสูงวัยมายเฟรนด์	64
3.18	Sequence Diagram การแสดงข่าวสาร	68
3.19	Sequence Diagram การแสดงปฏิทินกำหนดการ	70
3.20	Sequence Diagram การแสดงดาวน์โหลดเอกสาร	72
3.21	Sequence Diagram การแสดงบทสนทนา	74
3.22	Sequence Diagram แสดงส่งเอกสารตรวจสอบ	76
3.23	โครงสร้างฐานข้อมูลแบบ Firebase	78
3.24	โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)	78
3.25	โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)	78
3.26	โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)	79
3.27	โหนดเก็บข้อมูลประกาศ	80
3.28	โหนดเก็บข้อมูลเอกสารที่เกี่ยวข้อง	81
3.29	โหนดเก็บข้อมูลประวัติการสนทนা	82
3.30	โหนดเก็บข้อมูลกำหนดการ	83
3.31	โหนดเก็บข้อมูลการยืนยันสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา	84
3.32	โหนดเก็บข้อมูลของนักศึกษา	85
3.33	โหนดเก็บข้อมูลการจองคิวของนักศึกษา	86
3.34	โหนดเก็บข้อมูลคำถามที่พบบ่อย	86
4.1	ไฟล์ firebaseConfig.js	87
4.2	ไฟล์ firebaseInit.js	88
4.3	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue	89
4.4	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue	90
4.5	การสร้างโลจิก (logic) ของหน้าเข้าสู่ระบบ SignIn.vue	91
4.6	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าข่าวสาร Home.vue	92
4.7	การสร้างโลจิก(logic)ของหน้าข่าวสาร Home.vue	93
4.8	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้ารายละเอียดข่าวสาร ViewPost.vue	94
4.9	การสร้างโลจิกของหน้าดูรายละเอียดของข่าวสาร ViewPost.vue	94
4.10	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสนทนา Message.vue	96
4.11	การสร้างโลจิกของหน้าสนทนา Message.vue	98
4.12	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าปฏิทินกำหนดการ Schedule.vue	99
4.13	การสร้างโลจิกของหน้าปฏิทินกำหนดการ Schedule.vue	100
4.14	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ MgPost.vue	101
4.15	การสร้างโลจิกของหน้าหน้าสร้างประกาศ MgPost.vue	103
4.16	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าอัพโหลดเอกสารที่เกี่ยวข้อง MgDocument.vue	104

4.17	การสร้างລອຈິກຂອງໜ້າທີ່ເປົ້າໂສ່ງຂອງ MgDocument.vue	105
4.18	การສ້າງໜ້າຈົບສ່ວນຕິດຕໍ່ຜູ້ໃຊ້ຂອງໜ້າສ້າງກຳທັນດາກຈອງຄົວສ່າງເອກສານ MgQueue.vue	106
4.19	การສ້າງລອຈິກຂອງໜ້າສ້າງກຳທັນດາກຈອງຄົວສ່າງເອກສານ MgQueue.vue	107
4.20	ຕັ້ງແປຣໃນຄລາສ MainActivity	108
4.21	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການສ້າງເມນູນໝາທາງໜ້າໃນຄລາສ MainActivity	109
4.22	ຕັ້ງແປຣໃນຄລາສ FeedFragment	110
4.23	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການສືບຄັນຂໍ້ມູນຈາກ Cloude Firestore ກາຍໃນຄລາສ FeedFragment	111
4.24	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການດັກອື່ນຕົ້ນມື່ອຜູ້ໃຊ້ກົດທີ່ແກວປະກາສໃນຄລາສ FeedFragment . .	112
4.25	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການແສດງຜລຮຽນລະເອີຍດປະກາສຂອງຄລາສ PostDetailActivity .	113
4.26	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການດາວນໂລດເອກສານຂອງຄລາສ PostDetailActivity	114
4.27	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການແສດງຜລຮຽນລະເອີຍດປະກາສຂອງຄລາສ ChatActivity	114
4.28	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການສືບຄັນປະວິທີການສັນທະນາຂອງຄລາສ ChatActivity	116
4.29	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການສ່າງຂໍ້ຄວາມຂອງຄລາສ ChatActivity	117
4.30	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການແສດງຜລໜ້າເຂົ້າສູ່ຮບບຂອງຄລາສ SignInActivity	118
4.31	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການເຂົ້າສູ່ຮບບຂອງຄລາສ SignInActivity	119
4.32	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການແສດງຜລໜ້າປົງປົງທຶນຂອງຄລາສ ScheduleFragment	120
4.33	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການສືບຄັນຂໍ້ມູນກຳທັນດາກຈອງຄລາສ ScheduleFragment	121
4.34	ໂຄີດສ່ວນທີ່ໃຊ້ໃນການແສດງຜລໜ້າປົງປົງທຶນຂອງຄລາສ ScheduleFragment	122
ກ.1	ໜ້າເວັບດາວນໂລດ Android Studio	141
ກ.2	ໜ້າຕ່າງໆທີ່ຕ້ອນຮັບຂອງ Android Studio	142
ກ.3	ໜ້າຕ່າງໆຂໍ້ອຕກລາງການໃຊ້ຈານ Android Studio	142
ກ.4	ໜ້າຕ່າງໆທີ່ຈັດເກີບໄຟລ໌ຕ່າງໆ ຂອງ Android Studio	143
ກ.5	ໜ້າຕ່າງໆທີ່ຈັດເກີບໄຟລ໌ຕ່າງໆ ຂອງ Android Studio	143
ກ.6	ໜ້າຕ່າງໆຜລກາຮັດຕິດຕໍ່ Android Studio	144
ກ.7	ໜ້າເວັບດາວນໂລດ Node.js	144
ກ.8	ໄຟລ໌ຕິດຕໍ່ສໍາຫຼັບຕິດຕໍ່ Node.js	145
ກ.9	ໜ້າຕ່າງໆຕອນຮັບຂອງ Node.js	145
ກ.10	ໜ້າຕ່າງໆຂໍ້ອຕກລາງໃນການໃຊ້ Node.js	146
ກ.11	ໜ້າຕ່າງໆເລືອກໄຟລ໌ເຊື້ອງທີ່ຈະທຳການຕິດຕໍ່ Node.js	146
ກ.12	ໜ້າຕ່າງໆຕິດຕໍ່ Node.js	147
ກ.13	ຄໍາສັ່ງສໍາຫຼັບຕິດຕໍ່ Vue.js Fronted Framework	147
ກ.14	ໜ້າຕ່າງໆຜລກາຮັດຕິດຕໍ່	148
ຂ.1	ໜ້າເວັບດາວນໂລດ ESP installer package	149
ຂ.2	ໄຟລ໌ app-debug.apk ບນອຸປະກອນ	150
ຂ.3	ໜ້າຈອງຕ້ອນຮັບການຕິດຕໍ່ແອປລິເຄີບນັບອຸປະກອນແອນດຣອຍ໌	150
ຂ.4	ໜ້າຈອງຕ້ອນຮັບການຕິດຕໍ່ແອປລິເຄີບນັບອຸປະກອນແອນດຣອຍ໌	151
ຄ.1	ໜ້າຈອງຕ້ອນຮັບ	152
ຄ.2	ໜ້າຕ່າງໆຂອ່ສິຫຼືການເຂົ້າຖື	153

ค.3	หน้าจอเข้าสู่ระบบ	154
ค.4	หน้าแสดงข่าวสาร	154
ค.5	หน้ารายละเอียดของข่าวสาร	155
ค.6	เมนูนำทางหลักของแอปพลิเคชัน	156
ค.7	หน้าจอกำหนดการ	156
ค.8	หน้าจอต้อนรับ	157
ค.9	หน้าต่างขอสิทธิ์การเข้าถึง	158
ค.10	หน้าแสดงกำหนดการ	158
ค.11	หน้าจอเอกสาร	159
ค.12	หน้าจอสนทนา	160
ค.13	หน้าจอส่งเอกสาร	161
ค.14	หน้าจอถ่ายภาพเอกสาร	162
ค.15	หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร	162
ค.16	หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร	163
ค.17	หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร	164
ค.18	หน้าต่างแสดงสถานะการอัปโหลดเอกสาร	164
ค.19	หน้าจอของคิว	165
ค.20	หน้าต่างปฏิทินเลือกวันที่ต้องการส่งเอกสาร	166
ค.21	หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร	166
ค.22	หน้าจอคำนวณที่พับบ่อย	167
ค.23	หน้าเกี่ยวกับ	168

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

ในภาวะสังคมปัจจุบัน ประเทศไทยในปี 2561 มีผู้สูงอายุทั้งเพศและเพศหญิงเฉลี่ยประมาณ 16.06 เปอร์เซ็นต์ ของประชากรทั้งหมดในประเทศไทย ผู้สูงอายุส่วนใหญ่อยู่บ้านเพียงลำพัง อันเนื่องสาเหตุมาจากการลูกหลานต้องออกไปทำงานที่ต่างจังหวัด หรือในกรุงเทพมหานคร ส่งผลให้ผู้สูงอายุ สนใจที่จะใช้โซเชียลในการติดต่อสื่อสาร หรือแชร์เรื่องราวที่น่าสนใจในคนวัยเดียวกัน ในปัจจุบันมีเทคโนโลยีในการสื่อสารมากมาย เช่น เฟสบุ๊ค ไลน์ แอปพลิเคชันเหล่านี้ถูกออกแบบมาสำหรับคนรุ่นใหม่มากกว่าผู้สูงอายุ ปัญหาคือการใช้งานมีความซับซ้อน ตัวหนังสือที่มีขนาดเล็ก รวมไปถึงไม่มีฟังก์ชันที่ช่วยอำนวยความสะดวกให้ผู้สูงอายุในกรณีฉุกเฉินได้เฉพาะเจาะจง

ดังนั้นผู้พัฒนาจึงต้องการเสนอแอปพลิเคชันสูงวัยมายเฟรนด์ (OLD MY FRIENDS) ที่ช่วยให้ผู้สูงอายุสามารถสื่อสารหรือแชร์เรื่องราวที่น่าสนใจ กับคนวัยเดียวกัน และยังสามารถให้ความรู้เบื้องต้น เช่น สาเหตุ อาการ วิธีป้องกัน วิธีรักษา อาการแทรกซ้อน ในเรื่องโรคที่เกิดขึ้นบ่อย กับผู้สูงอายุจำนวน 5 โรค ได้แก่ โรคเบาหวาน โรคซีมเศร้า โรคความดันโลหิตสูง โรคข้อเสื่อม โรคอัลไซเมอร์ ที่จะเป็นประโยชน์สำหรับผู้สูงอายุ และผู้สูงอายุยังสามารถดู ตำแหน่งปัจจุบันของคนในครอบครัว และยังสามารถแจ้งเหตุเมื่อเกิดเหตุฉุกเฉินด้วยการโทรหรือส่งข้อความไปยังคนในครอบครัวได้ นอกจากนี้ แอปพลิเคชัน ยังประกอบด้วยฟังก์ชันแจ้งเตือนการทานยา ซึ่งจะช่วยให้ผู้สูงอายุสามารถทานยาได้ถูกต้อง และตรงตามเวลา

1.2 วัตถุประสงค์

- เพื่อพัฒนาระบบแอปพลิเคชันสำหรับผู้สูงอายุให้สามารถแบ่งปันเรื่องราวที่น่าสนใจได้
- เพื่อพัฒนาแอปพลิเคชันอำนวยความสะดวกในกรณีฉุกเฉิน
- เพื่อพัฒนาแอปพลิเคชันให้ความรู้พื้นฐานเรื่องโรคที่เกิดกับผู้สูงอายุผ่านระบบซอฟต์แวร์

1.3 ขอบเขตของโครงการ

1.3.1 ผู้ใช้งาน

- แอปพลิเคชันทำงานบน
- สามารถโพสต์ได้
- สามารถเพิ่มคอมเม้นท์ได้
- สามารถถูกใจโพสต์ได้
- สามารถแก้ไขโพสต์ได้
- สามารถแก้ไขคอมเม้นท์ได้
- สามารถลบโพสต์ได้
- สามารถเพิ่มคอมเม้นท์ได้
- สามารถเลือกประเภทโพสต์ได้
- สามารถตอบโต้กับแซทบอทได้
- สามารถพิมพ์คำสั่งในแซทบอทด้วยเสียงได้
- สามารถเพิ่มเพื่อนได้
- สามารถลบเพื่อนได้
- สามารถแซทกับเพื่อนได้
- สามารถสร้างกลุ่มได้
- สามารถแซทกลุ่มได้
- สามารถดูprofile ได้
- สามารถแก้ไขprofile ได้
- สามารถเปลี่ยนรูปprofile ได้
- สามารถเพิ่มครอบครัวได้
- สามารถแสดงตำแหน่งครอบครัวได้
- สามารถแจ้งเตือนผ่านโทรศัพท์ได้
- สามารถแจ้งเตือนผ่าน SMS ได้
- สามารถแจ้งเตือนการทานยาได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยให้ผู้สูงอายุสามารถแบ่งปันประสบการณ์ในคนวัยเดียวกันได้
2. ช่วยให้อ่านความสะดวกในการติดต่อสื่อสารกับครอบครัวได้
3. ช่วยให้ผู้สูงอายุได้รับความรู้ของโรคพื้นฐานที่พบในผู้สูงอายุ

1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)

1.5.1 ฮาร์ดแวร์

1. สมาร์ทโฟน (Smart phone)

- ทำงานบนระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 4.4 ขึ้นไป
- ทำงานบนระบบปฏิบัติการไอโอเอสเวอร์ชัน 10 ขึ้นไป

2. เดสก์ท็อป (Desktop)

- Chrome
- Safari
- Edge
- Firefox

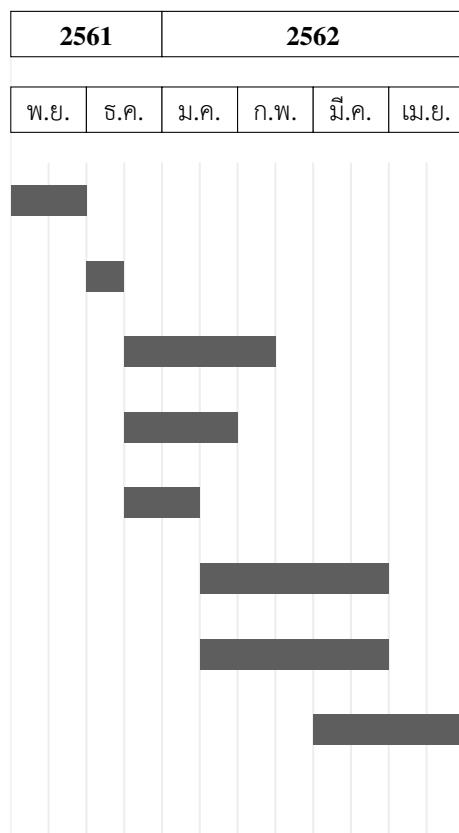
1.5.2 ซอฟต์แวร์ (Software)

1. Ionic 3 ซึ่งเป็น Frontend Framework สำหรับพัฒนาเว็บแอปพลิเคชัน
2. Firebase คือ Platform ที่รวมเครื่องมือต่าง ๆ ที่ใช้จัดการ Backend หรือ Server side
3. Dialogflow หรือ Api.ai เป็นแพลตฟอร์มที่ใช้ในการสร้างแชทบอทที่รองรับการทำ Natural Language understanding
4. Node Package Manager หรือ NPM เป็นซอฟต์แวร์ที่มาพร้อมกับ Node ที่ช่วยให้สามารถนำเข้าโมดูลต่าง ๆ ภาษาใน Node ได้
5. Library moment.js เป็น JavaScript Library สำหรับจัดการ Date Time
6. Google Maps APIs เป็น API ของ Google ไว้สำหรับเรียกใช้แผนที่
7. Visual Studio Code เครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชัน

1.5.3 แผนการดำเนินการ

ในการสร้างแอปพลิเคชันสูงวัยมายเฟรนด์ ผู้พัฒนาได้แบ่งขั้นตอนการดำเนินงานไว้ด้วยกัน 8 ขั้นตอน ดังต่อไปนี้

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะอธิบายถึงองค์ความรู้และทฤษฎีที่จำเป็นต่อการพัฒนาแอปพลิเคชันรวมทั้งงานวิจัยที่เกี่ยวข้อง มีรายละเอียดดังนี้

2.1 Hybrid Mobile Application

- 2.2 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์
- 2.3 ความรู้พื้นฐานของระบบปฏิบัติการไอโอเอส
- 2.4 ความรู้พื้นฐาน Ionic Framework
- 2.5 การใช้งาน Firebase เป็นฐานข้อมูล
- 2.6 Dialogflow
- 2.7 Libraries moment.js
- 2.8 Google Maps API
- 2.9 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 Hybrid Mobile Application



รูปที่ 2.1: Hybrid Mobile Application คืออะไร

ที่มา : <https://www.mindphp.com/คุณมี/73-คืออะไร/3663-hybrid-application-ไฮบริด-แอปพลิเคชัน-หรือ-hybrid-app-ไฮบริด-แอพ-คืออะไร.html>

Hybrid Application คือ แอปพลิเคชันที่ถูกพัฒนาขึ้นมาเพื่อให้สามารถทำงานได้บนระบบปฏิบัติการทั้งหมดโดยพัฒนาแค่ครั้งเดียว โดยจำเป็นต้องผ่านเฟรมเวิร์คต่างๆ เพื่อให้สามารถทำ-

งานบน OS นั้นๆได้ เช่น PhoneGap (โฟนเกป) ซึ่งเป็น Open source framework ด้วยการพัฒนาแอปด้วยเทคโนโลยีเว็บ html, CSS และ Java Script เป็นต้น

การพัฒนา Mobile Application แต่เดิมนั้น ถ้าจะเริ่มก็คงต้องแต่ยุค J2ME อาจจะเก่ามาก จนมาถึงยุคสมัยของ IOS และ Android รวมไปถึงน้องสุดท้องอย่าง Windows Phone ซึ่งแต่ก่อนก็มีพัฒนาจาก Window CE มาก่อนหน้านี้ จนผู้ใช้งานแต่ละฝั่ง platform เริ่มมีความสำคัญใกล้เคียงกัน ดังนั้น การพัฒนาแอป เฉพาะของ iOS หรือ Android เพียงอย่างเดียวถือเป็นการเสียโอกาสทางธุรกิจเป็นอย่างมาก จนมีคนเริ่มคิดหาวิธีทำให้ชีวิตต่างขึ้นโดยการเขียน HTML5 + CSS3 + JavaScript และใช้วิธีทำงานผ่าน Web View Component เป็นส่วนของหน้าเบราว์เซอร์ในแอปอีกด้วย ที่ของแต่ละ Platform จะกล้ายมาเป็นโครงการ Cordova และได้มีการพัฒนาส่วนขยาย Plug-In เพิ่มเรื่อยๆ ทำให้ปัจจุบันเราสามารถเข้าถึง Hardware หรือ Sensor ซึ่งโดยปกติ HTML5 ธรรมดามิ่งสามารถเข้าถึงได้

จากได้อ่านส่วนบุนma การเขียนแอปแบบ Hybrid Application คือ การพัฒนาโดยอาศัย Framework หรือ SDK ที่ถูกสร้างมาจากหลากหลายภาษาและมีเครื่องมือที่เหมาะสมกับ Framework หรือ SDK นั้น ๆ ให้เลือกใช้ในการพัฒนาที่หลากหลายตัวอย่างเช่น codova SDK ใช้ภาษา LUA , Acrobat AIR ใช้ภาษา ACTION SCRIPT 3 หรือ UNITY ใช้ C และ JAVASCRIPT ซึ่งการเขียนในรูปแบบนี้เราสามารถแปลงไปใช้กับระบบปฏิบัติการอื่น ๆ ได้และใช้เวลาอ้อยในการเพื่อพัฒนาหลาย ๆ แอปพลิเคชัน

2.1.1 ข้อดีของ Hybrid Mobile Application

1. พัฒนาด้วยภาษา HTML, CSS และ JavaScript ทำให้ง่ายและเรียนรู้ได้อย่างรวดเร็ว
2. พัฒนาครั้งเดียวสามารถใช้ได้หลาย Platform ทั้ง iOS, Android และ Window Phone
3. ใช้ต้นทุนในการพัฒนาน้อยกว่า Native App

2.1.2 ข้อเสียของ Hybrid Mobile Application

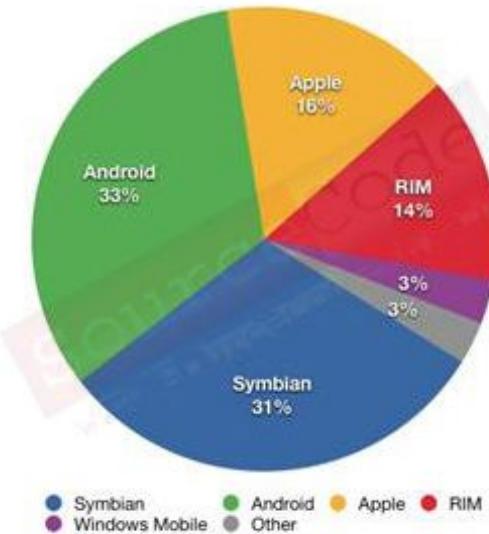
1. ประสิทธิภาพการทำงานจะด้อยกว่า Native App
2. ในบางกรณีอาจจะใช้ความสามารถของอุปกรณ์ได้ไม่เต็มที่ เนื่องจากต้องขึ้นอยู่กับ Framework ที่เลือกในการพัฒนานั้นมี Component ที่ต้องการหรือไม่ ดังนั้น Hybrid App จึงมีจุดเด่นในเรื่องความง่ายและพัฒนาได้รวดเร็ว และ Cross-Platforms คือพัฒนาครั้งเดียวแต่สามารถ

นำไปติดตั้งในหลาย Platforms แต่เมื่อพูดถึงเรื่องประสิทธิภาพในการทำงาน เช่นความเร็ว หรือการเรียกใช้หรือติดต่อ feature ต่าง ๆ ของอุปกรณ์ ก็ต้องยอมรับว่าอาจจะยังด้อยกว่าแอปพลิเคชันที่พัฒนาด้วย Native App ในบางลักษณะการทำงานอยู่ดี

2.2 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ (Android) คือระบบปฏิบัติการแบบเปิดเผยแพร่องค์รวมต้นฉบับ (Open Source) โดยบริษัท กูเกิล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลายราคา รวมทั้งสามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอ และความละเอียดแตกต่างกันได้ ทำให้ผู้บริโภคสามารถเลือกได้ตามต้องการและห้ามมองในทิศทางสำหรับนักพัฒนาโปรแกรม (Programmer) แล้วนั้นการพัฒนาโปรแกรมเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ ไม่ใช่เรื่องยาก เพราะมีข้อมูลในการพัฒนารวมทั้ง Android SDK (Software Development Kit) เตรียมไว้ให้กับนักพัฒนาได้เรียนรู้ และเมื่อนักพัฒนาต้องการจะเผยแพร่หรือจำหน่ายโปรแกรมที่พัฒนาแล้วเสร็จแอนดรอยด์ก็ยังมีตลาดในการเผยแพร่โปรแกรม Google PlayStore แต่หากจะกล่าวถึงโครงสร้างภาษาที่ใช้ในการพัฒนานั้น สำหรับ Android SDK จะยึดโครงสร้างของภาษาจาวา (Java language) ในการเขียนโปรแกรม เพราะโปรแกรมที่พัฒนามาได้จะต้องทำงานอยู่ภายใต้ Dalvik Virtual Machine เช่นเดียว กับโปรแกรมจาวา ที่ต้องทำงานอยู่ภายใต้ Java Virtual Machine (Virtual Machine เปรียบได้กับสภาพแวดล้อมที่โปรแกรมทำงานอยู่)

นอกจากนี้แล้วแอนดรอยด์ยังมีโปรแกรมแกรมที่เปิดเผยแพร่องค์รวมต้นฉบับ (Open Source) เป็นจำนวนมาก ทำให้นักพัฒนาที่สนใจสามารถนำชอร์ฟแวร์ต้นฉบับมาศึกษาได้ประกอบกับความนิยมของแอนดรอยด์ได้เพิ่มขึ้นอย่างมากในปัจจุบัน โดยดูได้จากส่วนแบ่งการตลาด ดังรูปที่ 2.2



รูปที่ 2.2: ส่วนแบ่งการตลาดระบบปฏิบัติการบนสมาร์ทโฟน

ที่มา : <https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าส/ระบบปฏิบัติการ-android.html>

2.2.1 ประวัติความเป็นมาของระบบปฏิบัติการแอนดรอยด์

เริ่มต้นระบบปฏิบัติการแอนดรอยด์ถูกพัฒนามาจากบริษัทแอนดรอยด์ (Android) เมื่อปี พ.ศ 2546 โดยมีนาย แอนดี้ รูบิน (Andy Rubin) ผู้ให้กำเนิดระบบปฏิบัติการนี้และถูกบริษัทกูเกิล เข้าซื้อกิจการเมื่อเดือนสิงหาคม ปี พ.ศ 2548 โดยบริษัทแอนดรอยด์ได้กลายมาเป็นบริษัทลูกของบริษัทกูเกิล

ระบบปฏิบัติการแอนดรอยด์ เป็นระบบปฏิบัติการที่พัฒนามาจากการนำเอาแกนกลางของระบบปฏิบัติการลินุกซ์ (Linux Kernel) ซึ่งเป็นระบบปฏิบัติการที่ออกแบบมาเพื่อทำงานเป็นเครื่องให้บริการ (Server) มาพัฒนาต่อ เพื่อให้กลายเป็นระบบปฏิบัติการบนอุปกรณ์พกพา (Mobile Operating System)

ต่อมาเมื่อเดือน พฤษภาคม ปี พ.ศ 2550 บริษัทกูเกิล ได้ทำการก่อตั้งสมาคม OHA (Open Handset Alliance) เพื่อเป็นหน่วยงานกลางในการกำหนดมาตรฐานกลาง ของอุปกรณ์พกพาและระบบปฏิบัติการแอนดรอยด์ โดยมีสมาชิกในช่วงก่อนตั้งจำนวน 34 รายเข้าร่วม ซึ่งประกอบไปด้วย บริษัทชั้นนำที่ดำเนินธุรกิจด้านการสื่อสาร เช่น โรงแรมผลิตอุปกรณ์พกพา บริษัทพัฒนาโปรแกรม ผู้

ให้บริการสื่อสาร และผู้ผลิตอะไหล่อุปกรณ์ด้านสื่อสาร [?]

หลังจากนั้น เมื่อเดือนตุลาคม ปี พ.ศ 2551 บริษัท ภูเก็ล ได้เปิดตัวมือถือตัวแรกที่ใช้ระบบปฏิบัติการแอนดรอยด์ ที่ชื่อ T-Mobile G1 หรืออีกชื่อนึงคือ HTC Dream โดยใช้แอนดรอยด์รุ่น 1.1 และหลังจากนั้น ได้มีการปรับพัฒนาระบบปฏิบัติการเป็นรุ่นใหม่ มาเป็นลำดับ

ซึ่งต่อมาได้มีการออกผลิตภัณฑ์จากบริษัทต่าง ๆ ออกมากหลากหลายรุ่น หลากหลายยี่ห้อ ตามการพัฒนาระบบปฏิบัติการแอนดรอยด์ ที่มีอยู่อย่างต่อเนื่อง ทำให้สินค้าของแอนดรอยด์ มีให้เลือกอยู่อย่างมากมาย

2.2.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

การทำความเข้าใจโครงสร้างของระบบปฏิบัติการแอนดรอยด์ [?] ถือว่าเป็นสิ่งสำคัญ เพราะถ้าหากพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบได้ทั้งหมด จะสามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนา เพื่อให้เกิดประสิทธิภาพในการทำงาน



รูปที่ 2.3: โครงสร้างของระบบปฏิบัติการแอนดรอยด์

ที่มา : <https://www.theandroid-mania.com/android-architecture/>

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ในรูปที่ 2.3 จะสังเกตได้ว่า มีการแบ่งออก

เป็นส่วน ๆ ที่มีความเกี่ยวเนื่องกัน โดยส่วนบนสุดเป็นส่วนที่ผู้ใช้งานทำการติดต่อโดยตรงซึ่งคือส่วนของ Applications ลำดับถัดมาเป็นองค์ประกอบอื่น ๆ ตามลำดับ และสุดท้ายเป็นส่วนที่ติดต่อกับอุปกรณ์โดยผ่านทาง Linux Kernel โครงสร้างของแอนดรอยด์สามารถอธิบายได้ดังนี้

1. Applications ส่วนแอปพลิเคชันหรือส่วนของโปรแกรมที่มากับระบบปฏิบัติการ หรือเป็นกลุ่มของโปรแกรมที่ผู้ใช้งานได้ทำการติดตั้งไว้ โดยผู้ใช้งานสามารถเรียกใช้โปรแกรมต่าง ๆ ได้โดยตรงซึ่งการทำงานของแต่ละโปรแกรมจะเป็นไปตามที่ผู้พัฒนาโปรแกรมได้ออกแบบและเขียนโค้ด (Code) โปรแกรมเอาไว้
2. Application Framework เป็นส่วนที่มีการพัฒนาขึ้นเพื่อให้นักพัฒนาสามารถพัฒนาโปรแกรมได้สะดวก และมีประสิทธิภาพมากยิ่งขึ้น โดยนักพัฒนาไม่จำเป็นต้องพัฒนาในส่วนที่มีความยุ่งยากมาก ๆ เพียงแค่ทำการศึกษาถึงวิธีการเรียกใช้งาน Application Framework ในส่วนที่ต้องการใช้งานแล้วนำมาใช้งาน ซึ่งมีหลายกลุ่มด้วยกัน ตัวอย่างเช่น
 - Activities Manager เป็นกลุ่มของชุดคำสั่งที่จัดการเกี่ยวกับวงจรการทำงานของหน้าต่างโปรแกรม (Activity)
 - Content Providers เป็นกลุ่มของชุดคำสั่ง ที่ใช้ในการเข้าถึงข้อมูลของโปรแกรมอื่น และสามารถแบ่งปันข้อมูลให้โปรแกรมอื่นเข้าถึงได้
 - View System เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับการจัดการโครงสร้างของหน้าจอที่แสดงผลในส่วนที่ติดต่อกับผู้ใช้งาน (User Interface)
 - Telephony Manager เป็นกลุ่มของชุดคำสั่งที่ใช้ในการเข้าถึงข้อมูลด้านโทรศัพท์ เช่น หมายเลขโทรศัพท์ เป็นต้น
 - Resource Manager เป็นกลุ่มของชุดคำสั่งในการเข้าถึงข้อมูลที่เป็นข้อความและรูปภาพ
 - Location Manager เป็นกลุ่มของชุดคำสั่งที่เกี่ยวกับตำแหน่งทางภูมิศาสตร์ที่ระบบปฏิบัติการได้รับค่าจากอุปกรณ์
 - Notification Manager เป็นกลุ่มของชุดคำสั่งที่จะถูกเรียกใช้เมื่อโปรแกรมต้องการแสดงผลให้กับผู้ใช้งาน ผ่านทางแถบสถานะ (Status Bar) ของหน้าจอ
3. Libraries เป็นส่วนของชุดคำสั่งที่พัฒนาด้วย C/C++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตามวัตถุประสงค์ของการใช้งาน เช่น Surface Manager จัดการเกี่ยวกับการแสดงผล Media

Framework จัดการเกี่ยวกับการการแสดงภาพและเสียง Open GL|ES และ SGL จัดการเกี่ยวกับภาพ 3 มิติ และ 2 มิติ SQLite จัดการเกี่ยวกับระบบฐานข้อมูล เป็นต้น

4. Android Runtime จะมี Darvik Virtual Machine ที่ถูกออกแบบมาเพื่อให้ทำงานบนอุปกรณ์ที่มีหน่วยความจำ (Memory) หน่วยประมวลผลกลาง (CPU) และพลังงาน (Battery) ที่จำกัดซึ่งการทำงานของ Darvik Virtual Machine จะทำการแปลงไฟล์ที่ต้องการทำงานไปเป็นไฟล์ .DEX ก่อนการทำงานเหตุผลเพื่อให้มีประสิทธิภาพเพิ่มขึ้นเมื่อใช้งานกับหน่วยประมวลผลกลางที่มีความเร็วไม่มากส่วนต่อมาคือ Core Libraries ที่เป็นส่วนรวมรวมคำสั่งและชุดคำสั่งสำคัญโดยถูกเขียนด้วยภาษาจาวา (Java Language)
5. Linux Kernel เป็นส่วนที่ทำหน้าที่หัวใจสำคัญในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 2.6. Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

2.2.3 ข้อเด่นของระบบปฏิบัติการแอนดรอยด์

เนื่องจากระบบปฏิบัติการแอนดรอยด์มีการเจริญเติบโตอย่างรวดเร็วและมีส่วนแบ่งตลาดของอุปกรณ์ด้านนี้ขึ้นทุกขณะ ทำให้กลุ่มผู้ใช้งานและกลุ่มนักพัฒนาโปรแกรมให้ความสำคัญกับระบบปฏิบัติการแอนดรอยด์เพิ่มมากขึ้น

เมื่อมองในด้านของกลุ่มผลิตภัณฑ์บริษัทที่มีการพัฒนาผลิตภัณฑ์รุ่นใหม่ ได้มีการนำเอาระบบปฏิบัติการแอนดรอยด์ไปใช้ในสินค้าของตนเองพร้อมทั้งยังมีการปรับแต่งให้ระบบปฏิบัติการมีความสามารถ การจัดวาง โปรแกรมและลูกเล่นใหม่ ๆ ที่แตกต่างจากคู่แข่งในท้องตลาดโดยเฉพาะอย่างยิ่งกลุ่มสินค้าที่เป็นมือถือรุ่นใหม่ (SmartPhone) และอุปกรณ์จอสัมผัส(Touch Screen)โดยมีลักษณะแตกต่างกันไป เช่น ขนาดหน้าจอ ระบบโทรศัพท์ ความเร็วของหน่วยประมวลผล ปริมาณหน่วยความจำ แม้กระทั่งอุปกรณ์ตรวจจับ(Sensor)ต่าง ๆ

หากมองในด้านของการพัฒนาโปรแกรม ทางบริษัท Google ได้มีการพัฒนา Application Framework ไว้สำหรับนักพัฒนาใช้งานได้อย่างสะดวกและไม่เกิดปัญหาเมื่อนำชุดโปรแกรมที่พัฒนาขึ้นมา ไปใช้กับอุปกรณ์ที่มีลักษณะต่างกัน เช่น ขนาดจออุปกรณ์ไม่เท่ากัน ก็ยังสามารถใช้งานโปรแกรมได้เหมือนกัน เป็นต้น

2.2.4 การจัดการเกี่ยวกับวัสดุจัดแอลกอกทิวิตี้ของแอปพลิเคชัน

ขณะที่ผู้ใช้เปิดใช้งานแอปพลิเคชัน -> ออกจากแอปพลิเคชัน -> แล้วกลับเข้ามาในแอป-พลิเคชันอีกครั้งแอลกอกทิวิตี้จะมีการร่าย Method ต่างๆ เกิดขึ้นในวัสดุจัดแอลกอกทิวิตี้ ยกตัวอย่างเช่น เมื่อแอลกอกทิวิตี้เริ่มทำงานครั้งแรกจะแสดงข้อความอยู่ด้านบนสุดของระบบ (Foreground) และรอรับการทำงานจากผู้ใช้ในระหว่างกระบวนการนี้ระบบจะมีการเรียกใช้งาน Callback Method หรือ Method ที่ถูกเรียกใช้งานอัตโนมัติในแอลกอกทิวิตี้ที่ได้กำหนดการทำงานให้กับ UI และส่วนติดต่ออื่น ๆ ไว้ ถ้าผู้ใช้มีการใช้งานใด ๆ ที่เป็นการเรียกแอลกอกทิวิตี้อื่นขึ้นมาหรือสลับไปใช้งานแอปพลิเคชันอื่น ระบบจะเรียก Callback Method อีกอันขึ้นมา เช่น ช่องแอปพลิเคชันไว้ด้านหลัง Background (ไม่แสดงแอลกอกทิวิตี้แต่ Instance และ Method นั้นยังทำงานอยู่)

ภายใน Callback Method สามารถกำหนดการทำงานในแอลกอกทิวิตี้เมื่อผู้ใช้ออกจากแอปพลิเคชันและกลับเข้ามาใช้งานแอปพลิเคชันใหม่อีกครั้งได้ ตัวอย่าง ถ้าแอปพลิเคชันเป็นแอปพลิเคชัน Streaming Video จะจะสั่งให้ทำการหยุด Video ชั่วคราว และปิดการเชื่อมต่อ Network ไว้ก่อน เมื่อผู้ใช้สลับไปใช้แอปพลิเคชันอื่น และทันทีที่ผู้ใช้กลับมาใช้งานแอปพลิเคชันต่อ ก็ให้ทำการเชื่อมต่อ กับ Network และก้อนคุณภาพให้ผู้ใช้กลับไปเล่น Video ในตำแหน่งที่ค้างต่อไปทันทีโดยที่ไม่ต้องเริ่มต้นแอปพลิเคชันใหม่ เป็นต้น

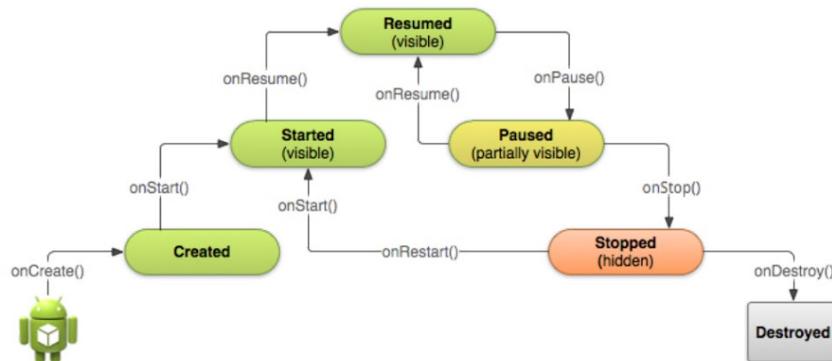
2.2.5 กระบวนการเริ่มการทำงานของแอลกอกทิวิตี้ (Activity)

ในระบบแอนดรอยด์การกำหนดโค้ดเริ่มต้นไว้ในแอลกอกทิวิตี้โดยสัมพันธ์กับ Method ที่ถูกเรียกใช้งานอัตโนมัติ (Callback Method) อย่างเป็นลำดับ ตั้งแต่เริ่มต้นแอลกอกทิวิตี้ไปจนถึงสิ้นสุดและปิดการทำงานของ Activity ลง

2.2.6 ทำความรู้จักกับ Lifecycle Callback

ในขณะที่แอลกอกทิวิตี้ [?] ทำงานระบบจะเรียกใช้ Callback Method ตามลำดับในลักษณะที่คล้ายกับการกรอกพิธีมิตร นั่นคือ แต่ละขั้นตอนวัสดุจัดแอลกอกทิวิตี้คือส่วนแยกอย่างแต่ละขั้นของพิธีมิตร เช่น เมื่อระบบสร้าง Instance ของแอลกอกทิวิตี้ขึ้นมาใหม่ Method ที่เรียกใช้งานอัตโนมัติ (Callback Method) จะขับ Activity Method ขึ้นมาด้านบนโดยด้านบนของพิธีมิตรคือจุดที่แออลกอกทิวิตี้กำลังทำงานแสดงอยู่ด้านหน้า (Foreground Activity) สุดและผู้ใช้กำลังใช้งานอยู่และเมื่อผู้ใช้กำลังจะออกจากแอลกอกทิวิตี้ระบบจะเรียกใช้ Method อื่นซึ่งทำให้ Activity Method ถอยกลับไป

อยู่ด้านล่างของพีระมิดตามลำดับเพื่อหยุดการทำงานและลบแอคทิวิตี้ออกไป ในบางกรณีแอคทิวิตี้จะย้ายลงมาอยู่บางจุดและรอจังหวะที่จะถูกเรียกกลับขึ้นมาด้านบนอีก เช่น ในกรณีเมื่อผู้ใช้สลับไปใช้งานแอปพลิเคชันอื่นแล้วกลับมาใช้งานอีกครั้ง



รูปที่ 2.4: วัฏจักรของแอคทิวิตี้บนระบบปฏิบัติการแอนดรอยด์

ที่มา : <https://www.dev2qa.com/android-activity-lifecycle-example/>

จากรูปที่ 2.7 แสดงวัฏจักรของแอคทิวิตี้ในรูปแบบโครงสร้างพีระมิดโดยแสดงให้เห็นว่า Method ที่เรียกใช้งานอัตโนมัติ (Callback Method) ได้แก่ onCreate(), onStart(), onResume() และ onRestart() จะขยับแอคทิวิตี้ขึ้นไปด้านบนสุดที่ Resumed Method และมี Method ได้แก่ onPause(), onStop() และ onDestroy() ที่จะขยับแอคทิวิตี้ลงมาด้านล่าง แอคทิวิตี้ยังสามารถกลับไปทำงานที่ตำแหน่ง Resumed Method จากตำแหน่ง Paused และ Stopped ได้อีกด้วย

ในบางครั้งไม่จำเป็นต้องเรียกใช้งาน Callback Method ทั้งหมดเสมอไปขึ้นกับความซับซ้อนของแอคทิวิตี้ อย่างไรก็ตามเป็นสิ่งสำคัญที่นักพัฒนาควรทำความเข้าใจแต่ละ Method เพื่อให้มั่นใจได้ว่าแอปพลิเคชันของที่ได้พัฒนาตอบสนองเป็นไปตามที่ผู้ใช้คาดหวัง ดังนั้น ในการใช้งาน Callback Method ที่ถูกวิธีจะช่วยให้แอปพลิเคชันทำงานได้เป็นอย่างดี ดังนี้

- ไม่หยุดการทำงานหรือค้าง กรณีมีสายยังคงเข้าหรือมีการสลับไปใช้งานแอปพลิเคชันอื่น
- ไม่ใช้ทรัพยากรที่มีค่าของระบบอย่างสูญเปล่า ถ้าไม่มีการใช้งานแอคทิวิตี้ใดๆ
- ไม่กระทบต่อกระบวนการในขั้นตอนการใช้งานของผู้ใช้กรณีออกจากแอปพลิเคชันแล้วกลับเข้ามาใช้งานอีกครั้ง
- ไม่หยุดการทำงานหรือระบบค้างที่กระทบการใช้งานของผู้ใช้กรณีมีการหมุนหน้าจอแนวตั้ง

นอนและแนวตั้งสลับกัน

เหตุการณ์ที่แอคทิวิตี้มีการเปลี่ยน Method ต่าง ๆ ตามแสดงในรูปที่ 2.7 แต่มีอยู่ 3 Method เท่านั้นที่แอคทิวิตี้จะยังคงอยู่คุณที่ในช่วงเวลาระยะเวลาหนึ่งไม่เปลี่ยนไป Method อื่นในทันทีได้แก่

- Resumed (แสดงอยู่ ทำงานอยู่) ใน Method นี้แอคทิวิตี้จะแสดงอยู่ด้านหน้าสุดและผู้ใช้กำลังใช้งานอยู่ บ่อยครั้งจะเรียกว่า Running Method
- Paused (แสดงหน้าจอบางส่วน ไม่ถูกบังสนิท) ใน Method นี้แอคทิวิตี้จะถูกบดบังด้วยแอคทิวิตี้อื่น เช่น แอคทิวิตี้อื่นที่อยู่ด้านหน้าสุดที่แสดงในลักษณะกึ่งโปร่งใสหรือไม่ได้แสดงแบบเต็มหน้าจอ แอคทิวิตี้ในสถานะนี้จะไม่สามารถรับคำจากผู้ใช้และทำงานคำสั่งใด ๆ ได้
- Stopped (แสดงหน้าจอแบบ Background ผู้ใช้มองไม่เห็น) ใน Method นี้ แอคทิวิตี้จะถูกบดบังอย่างสมบูรณ์และผู้ใช้มองไม่เห็นโดยจะถูกบังไปอยู่ด้านหลังในขณะที่อยู่ใน Method นี้ ค่า Activity Instance และตัวแปรทั้งหมดจะยังคงอยู่แต่จะไม่สามารถถูกเรียกมาใช้งานจากโค้ดได ๆ ได้

ในขณะที่ Method อื่น เช่น Created และ Started จะแสดงช่วงระหว่างแล้วระบบ ก็จะเปลี่ยนไป Method อื่นในทันทีที่ Method ถูกเรียกใช้งานอัตโนมัติ นั่นคือ หลังจากที่ระบบเรียกใช้งาน onCreate() แล้วก็จะเรียกใช้งาน onStart() ทันทีและสุดท้ายตามด้วย onResumne() ซึ่งก็จะเข้าสู่ Resumed Method ทั้งหมดก็คือวัฏจักรแอคทิวิตี้เบื้องต้น

2.3 ความรู้พื้นฐาน Ionic Framework

Ionic Framework คือเครื่องมือในการสร้าง Mobile Application เป็นเครื่องมือสร้างแอปมือถือที่สามารถสร้างที่เดียว สามารถใช้งานได้บนระบบปฏิบัติการ iOS, Android และ Windows ซึ่งก็จะใช้งานร่วมกับ Framework ตัวอื่น ๆ ได้ คือ Angular และ Apache Cordova ในตอนสุดท้ายเพื่อให้ทั้งแอปที่เขียนมาใช้ได้กับทุกระบบปฏิบัติการ

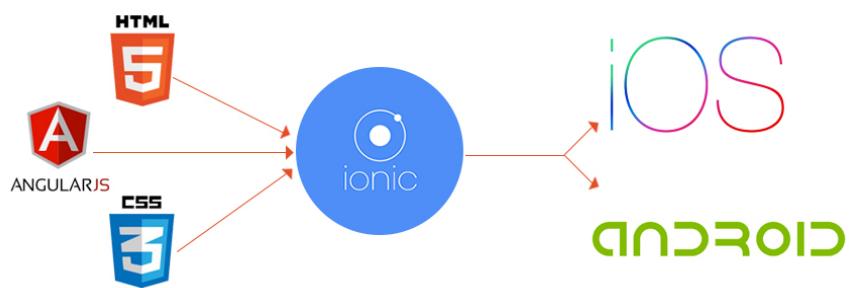
2.3.1 เทคโนโลยีที่ใช้ในการพัฒนา Ionic Framework

Ionic Framework พัฒนา Frontend ด้วยภาษา HTML , CSS , JavaScript และถูก Build เป็น Application ด้วย Cordova

HTML5 คือ ภาษาマーกรกอปป์ ที่ใช้สำหรับเขียน website ซึ่ง HTML5 นี้เป็นภาษาที่ถูกพัฒนาต่อมาจากภาษา HTML และพัฒนาขึ้นมาโดย WHATWG (The Web Hypertext Application Technology Working Group) โดยได้มีการปรับเพิ่ม Feature หลายอย่างเข้ามาเพื่อให้ผู้พัฒนาสามารถใช้งานได้ง่ายมากยิ่งขึ้น

CSS3 คือ ส్టాటిల్స్‌చీ పెన్ภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลของ HTML พุดจాయๆ คือทำให้การแสดงผลของ HTML ให้สวยงาม

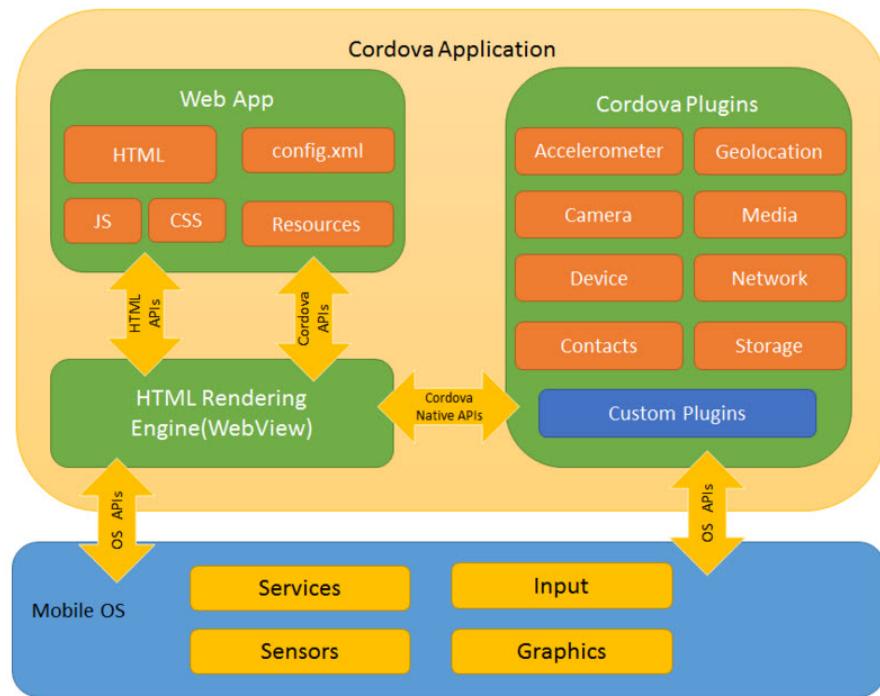
AngularJs คือ JavaScript Framework รูปแบบหนึ่งที่พัฒนามาจาก Google หน้าที่ของมันคือ เป็น engine ที่ใช้ควบคุมในส่วน front end ของเว็บได้เป็นอย่างดีมีการทำงานแบบ Model View Controller (MVC)



รูปที่ 2.5: การทำงานของ Ionic Framework

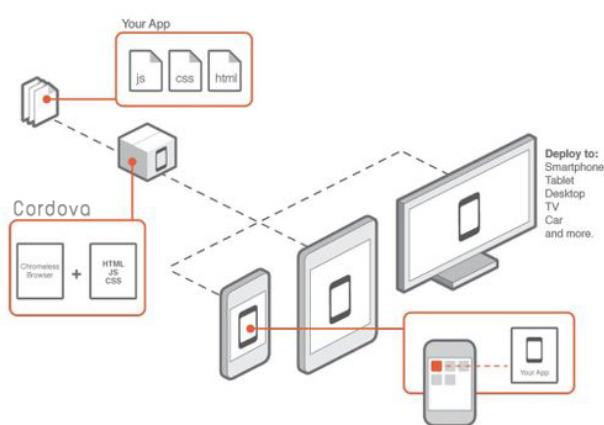
ที่มา : <http://blog.prscreative.com/what-is-ionic/>

2.3.2 การทำงานของ Cordova Application



รูปที่ 2.6: Cordova Application

ที่มา : <http://blog.prscreative.com/what-is-ionic/>



รูปที่ 2.7: Cordova Build Application

ที่มา : <http://blog.prscreative.com/what-is-ionic/>

Cordova มีหน้าที่ห่อหุ้มแอปพลิเคชันไว้ และทำหน้าที่ติดต่อกับ Hardware ของ Mobile เป็นหลัก เพราะมี API ติดต่อกับ Hardware โดยตรง เช่น Camera, Contacts, Media, Network

2.3.3 ความแตกต่างระหว่าง PhoneGap/Cordova

PhoneGap และ Cordova มีลักษณะคล้ายกัน เนื่องจากมันเกือบจะเหมือนกัน ต่างกันเพียง เรื่องของลิขสิทธิ์ และการนำไปใช้งาน

PhoneGap : ในยุคแรกๆพัฒนาโดย Nitobi เปิดให้ใช้งานแบบ Open Source ซึ่งได้รับความนิยมในการนำมาใช้เป็นเทคโนโลยี Hybrid ซึ่งต่อมาถูกซื้อโดยบริษัท Adobe เพื่อนำมาเสริมทัพให้กับโปรแกรม Adobe Dreamweaver เพื่อให้สามารถ Build app จากโปรแกรม Dreamweaver ให้รองรับหลาย Platform ได้ แต่มีค่าลิขสิทธิ์โปรแกรม

Cordova : เกิดขึ้นจากการตกลงกันระหว่าง Adobe และ Nitobi ด้วยแนวคิดที่อยากให้ PhoneGap เป็น Open Source ต่อไป จึงได้มีการตกลงกันให้นำโค้ดของ PhoneGap ไปตั้งเป็นชื่อใหม่ นั่นก็คือ Cordova เพื่อมอบให้ Apache Foundation ไปดูแลถูกนำไปใช้ในโครงการ Hybrid mobile application หลายโครงการ เช่น AppGyver, Ionic framework

2.3.4 เริ่มต้นการใช้งาน

การเริ่มต้นใช้งาน Ionic Framework สามารถศึกษาการติดตั้งได้ที่ <https://ionicframework.com/docs/v3/intro/installation/> ซึ่งเริ่มต้นเราจะต้องทำการติดตั้ง Cordova CLI และ Ionic CLI ผ่าน npm ก่อน

```

1 $ npm install -g cordova
2 # ติดตั้งCordova CLI
3 $ npm install -g ionic
4 # ติดตั้งIonic CLI
5 $ ionic start ชื่อโปรเจค
6 # สร้างโปรเจคไอโอนิกมีลักษณะซึ่งเริ่มต้นเป็นBlank
7 $ cd ชื่อโปรเจค
8 # เข้าไปในโปรเจคที่เราสร้าง
9 $ ionic serve
10 # รันไอโอนิกแบบlocalhost:8000

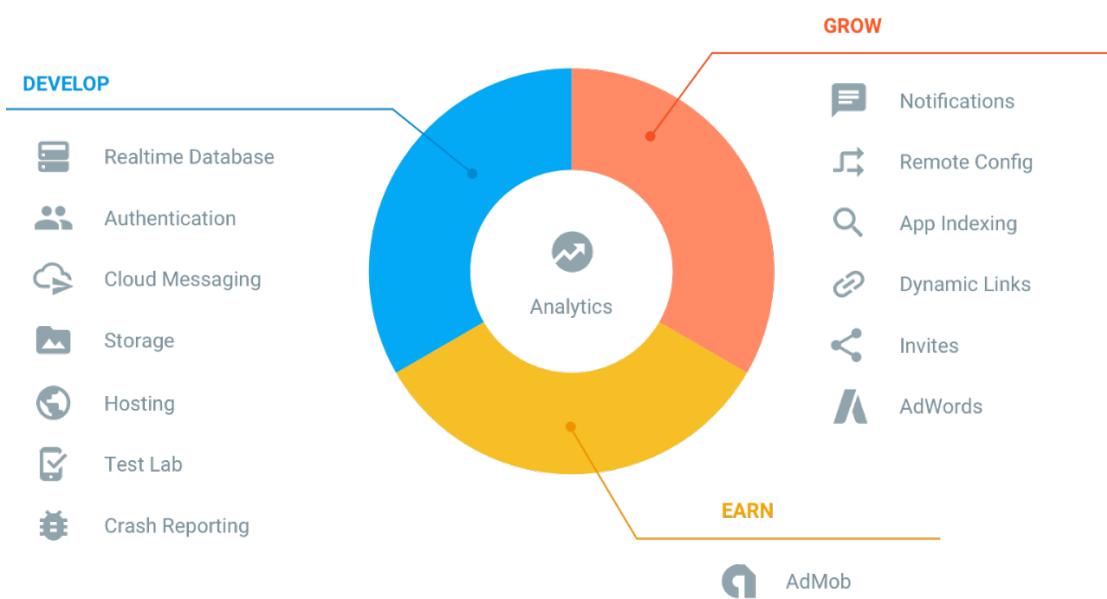
```

รูปที่ 2.8: แสดงการติดตั้ง cli

2.4 การใช้งาน Firebase เป็นฐานข้อมูล

Firebase [?] คือ บริการ Backend และ แพลตฟอร์ม ครอบคลุมสำหรับนักพัฒนาแอปพลิเคชัน และโปรแกรมประยุกต์บนเว็บแพลตฟอร์มที่มีเครื่องมือและโครงสร้างพื้นฐานที่ได้รับการออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันพลิเคชันที่มีคุณภาพสูง Firebase (ไฟร์เบส) ถูกสร้างขึ้นจากคุณสมบัติเสริมว่านักพัฒนาสามารถทดสอบและจับคู่เพื่อให้พอดีกับความต้องการของตน บริษัท ก่อตั้งขึ้นในปี 2011 โดยแอนดรูว์และเจมส์ เทมป์ลิน สินค้าเริ่มต้น Firebase เป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล ดังรูป 2.9

Google Firebase 2.0 มีการพัฒนาจากการ Backend เก็บข้อมูลอย่างเดียว มาเป็น แพลตฟอร์มครอบคลุมสำหรับนักพัฒนาแอปพลิเคชัน (รองรับ iOS, Android, Web) และรองรับบริการทุกอย่างที่นักพัฒนาแอปต้องการใช้งาน



รูปที่ 2.9: Firebase 2.0

ที่มา : www.mindphp.com/คุ้มครอง/73-คืออะไร/3921-what-is-firebase-backend.html

2.4.1 บริการหลักของเฟร์เบส

- Realtime Database จัดเก็บและซิงค์ข้อมูลระหว่างผู้ใช้และอุปกรณ์ต่างๆแบบเรียลไทม์โดยใช้ฐานข้อมูล NoSQL ที่ออกแบบมาสำหรับคลาวด์ ซิงค์ข้อมูลที่อัปเดตระหว่างอุปกรณ์ที่เชื่อมต่อเป็นมิลลิวินาทีและข้อมูลจะยังคงมีอยู่ถ้าแอพพลิเคชันอफ์ไลน์
- Authentication จัดการบัญชีผู้ใช้ด้วย Firebase Auth ซึ่งใช้งานง่ายและปลอดภัยมีวิธีการหลายในการสร้างบัญชีผู้ใช้และตรวจสอบความถูกต้อง ได้แก่ อีเมล/รหัสผ่าน, ผู้ให้บริการบุคคลที่สาม เช่น Google หรือ Facebook
- Cloud Storage จัดเก็บภาพเสียงวิดีโอหรือเนื้อหาอื่น ๆ เช่น รูปภาพโปรดิฟ์ผู้ใช้ หรือวีดีโอทัศน์ต่างๆ เป็นต้น ซึ่งมีความปลอดภัยในการอัปโหลดไฟล์และดาวน์โหลดสำหรับแอพพลิเคชัน
- Hosting ใช้ในการเผยแพร่เว็บไซต์ โดยเนื้อหาภายในเว็บเป็นเดชบอร์ด รายงานข้อต่างๆ ของผู้ใช้ ซึ่งต้องทำการเข้าสู่ระบบก่อน
- Crashlytics เป็นบริการล่าสุดที่กูเกิลได้เข้าควบรวมเข้ามาไว้ในบริการเฟร์เบส สามารถรายงานข้อขัดข้องได้อย่างมีประสิทธิภาพ เปิดใช้งานการวิเคราะห์แบบเรียลไทม์เพื่อช่วยให้

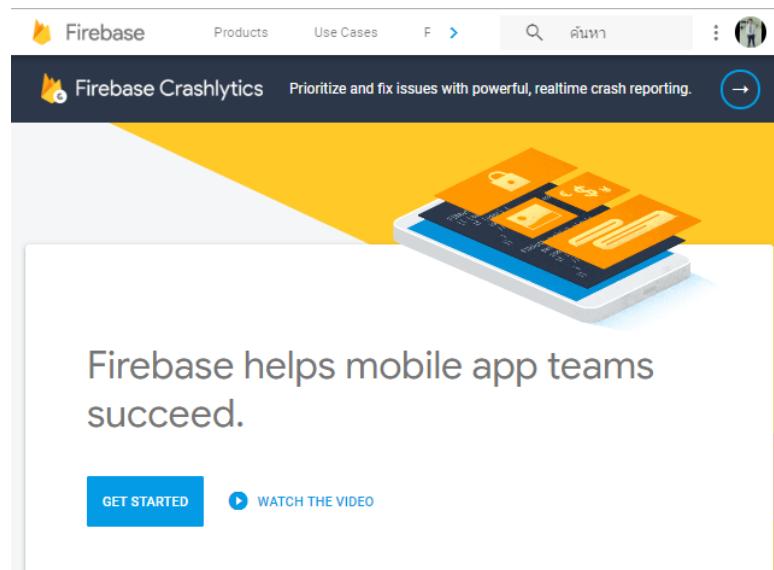
เข้าใจสิ่งที่เกิดขึ้นในแอพพลิเคชัน เครื่องมือวิเคราะห์ข้อมูลจะให้ข้อมูลเชิงลึก

- Cloud Firestore เป็นบริการในส่วนของ Database ที่ใช้ระบบฐานข้อมูลแบบ NoSQL ที่เป็นแบบ Document Database และเป็นการนำเอาข้อดีต่างๆ ของบริการด้านฐานข้อมูลของ Realtime Database มาปรับปรุงพัฒนาต่อและเพิ่มความสามารถขึ้นไปมากขึ้น ซึ่งผู้เขียนจะได้กล่าวถึงในบทถัดไป

2.4.2 การพัฒนา Cloud Firestore

การพัฒนา Cloud Firestore แบ่งออกแบบ 5 ขั้นตอน ดังนี้

- การสร้าง Cloud Firestore เพื่อใช้งานในโครงการ ในขั้นตอนแรกทำการสร้าง Database เพื่อที่จะใช้งาน Cloud Firestore ก่อน โดยใช้บัญชี Gmail ที่มาเข้าไปที่เว็บ <https://firebase.google.com> ดูรูปภาพที่ 2.17



รูปที่ 2.10: เว็บ <https://firebase.google.com>

ที่มา :<https://medium.com/20scoops-cnx/เข้มข้นกับ-firebase-cloud-firestore-ระบบฐานข้อมูลที่เปิดตัวใหม่ล่าสุดจาก-firebase-แบบจัดเต็ม-d001e43e2be7>

- ติดตั้ง SDKs เพื่อใช้งาน Cloud Firestore โดย SDKs ที่ Firebase ได้เตรียมไว้ให้ สามารถดูรายละเอียดได้ที่ <https://firebase.google.com/docs/firestore/quickstart>
- ออกแบบโครงสร้างและการจัดการข้อมูล

- ระบบฐานข้อมูลของ Cloud Firestore จะเป็น NoSQL แบบ Document ซึ่งจะแตกต่างจากระบบฐานข้อมูลแบบ SQL โดยจะไม่มีตาราง ไม่มีเวลา แต่เก็บข้อมูล ภายใน Document จะเก็บแบบ Key-value โดยแต่ละ Document จะถูกเก็บไว้ใน Collection ซึ่งใน Document สามารถมี Subcollection ได้
- Collection เป็นการเรียกชื่อแทนของการเก็บห้องๆ เอกสารไว้ด้วยกัน เช่น เก็บข้อมูลของ User จำนวนมากไว้ด้วยกัน จึงตั้งชื่อ Collection ว่า Users ซึ่งใน Collection เดียว กันผู้ใช้งานสามารถใส่ข้อมูลที่แตกต่างชนิดกันในแต่ละ Key และ Document ได้ โดยในแต่ละ Key และ Document จะมีอิสระในการใส่ข้อมูล แต่ควรใส่ข้อมูลในแต่ละ Key ของ Document เป็นประเภทเดียวกัน เพราะจะทำให้การค้นหาและการจัดเรียงลำดับของข้อมูลนั้นง่ายขึ้น
- Subcollection สามารถสร้าง Subcollection ของ Subcollection ไปได้เรื่อยๆ โดย Cloud Firestore ว่าสามารถซ้อนกันไปได้ 100 ลำดับขึ้น

4. การรับและส่งข้อมูล การรับและส่งข้อมูลจาก Cloud Firestore จะมี 2 วิธี โดยจะสามารถใช้ได้ทั้งการรับข้อมูลและการส่งข้อมูล

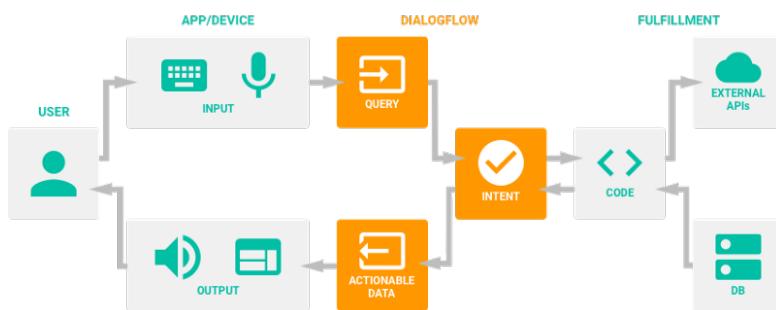
- การรับข้อมูลเพียงครั้งเดียวจะเป็นการรับข้อมูลเมื่อมีจุดประสงค์ที่จะไม่ต้องการรับรู้การเปลี่ยนแปลงของข้อมูล ซึ่ง ณ ขณะนั้นข้อมูลมีค่าเป็นอะไรมีจะได้ค่านั้นมา หากมีการเปลี่ยนแปลงข้อมูลในภายหลัง ผู้ใช้งานต้องเป็นผู้จัดการรับข้อมูลล่าสุดเอง โดยวิธีการรับข้อมูลเพียงครั้งเดียวจะใช้ Method Get()
- การรับข้อมูลแบบ Realtime update จะเป็นการรับข้อมูลเมื่อผู้ใช้งานมีจุดประสงค์ที่จะต้องการรับรู้การเปลี่ยนแปลงของข้อมูล ซึ่ง ณ ขณะที่ข้อมูลเกิดการเปลี่ยนแปลง จะมีการรับข้อมูลที่เกิดการเปลี่ยนแปลงโดยอัตโนมัติ โดยในครั้งแรกที่มีการรับข้อมูล จะสร้าง Initialize Instance และทุกครั้งที่ข้อมูลมีการเปลี่ยนแปลงก็จะส่ง Callback ให้กับ Listener โดยผ่าน Method onSnapshot()

5. การป้องกันและความปลอดภัยของข้อมูลข้อมูลใน Cloud Firestore ได้มีการออกแบบให้สามารถกำหนดกฎของความปลอดภัยต่างๆ ได้โดยผ่าน Firebase Console ซึ่งหากใช้ Cloud Firestore ผู้ใช้งานสามารถมาทำเรื่องการป้องกันและรักษาความปลอดภัยของข้อมูลเพียงที่เดียว สามารถใช้ได้ทั้งหมดไม่ว่าจะเป็น Web และ Mobile ส่วนฝั่ง Server ก็สามารถใช้ IAM ใน Google Cloud Platform มาจัดการความปลอดภัยสำหรับ Cloud Firestore ได้

2.5 Dialogflow

Dialogflow คือ platform สำหรับสร้าง chatbot ของ Google ที่ใช้ machine learning ด้าน Natural Language Processing (NLP) มาช่วยในการทำความเข้าใจถึงความต้องการ (intent) และสิ่งที่ต้องการ (entity) ในประโยคสนทนากลางผู้ใช้งาน และตอบคำถามตามความต้องการของผู้ใช้งาน ตามกฎ หรือ flow ที่ผู้พัฒนาวางแผนไว้ ซึ่ง Dialogflow จะช่วยเพิ่มความยืดหยุ่นของประโยคที่ chatbot รับมา ว่าไม่จำเป็นต้องตรงตามเงื่อนไข แบบ rule based ก็สามารถเข้าใจถึงความต้องการของผู้ใช้งานได้

2.5.1 เริ่มต้นใช้งาน Dialogflow

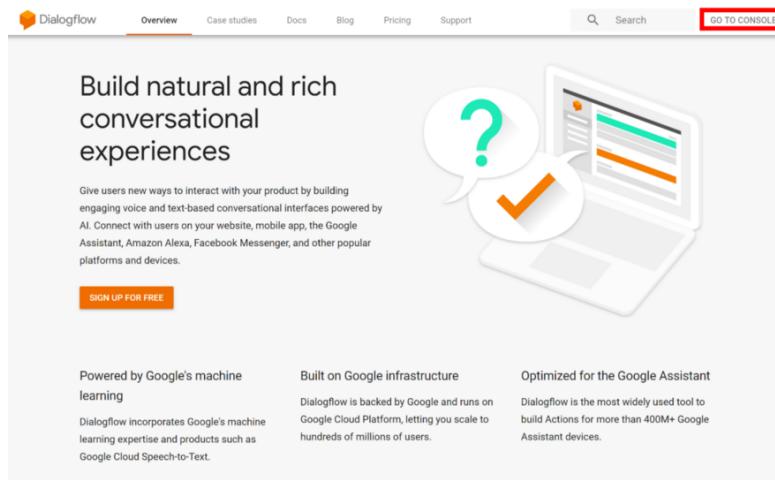


รูปที่ 2.11: รูปแบบการส่งข้อมูลผู้ใช้ไปยังแซทบอท

ที่มา :<https://dialogflow.com/docs/agents>

1. ลงทะเบียน/ล็อกอินเข้า Dialogflow

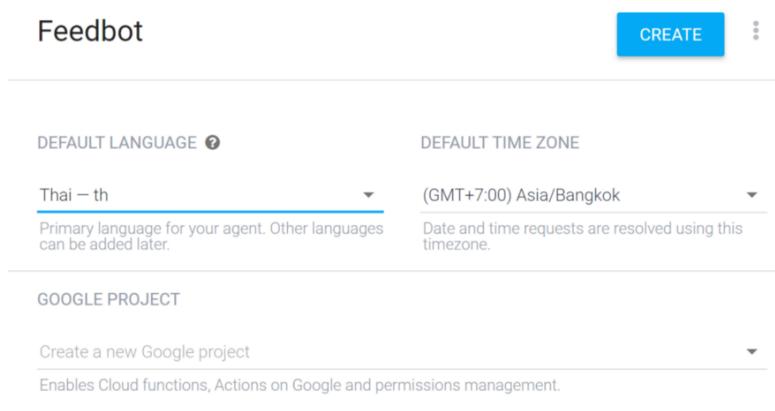
ในการสร้าง Agent เราต้องลงทะเบียนเข้าใช้งานก่อนนะ โดยไปยังหน้าเว็บของ Dialogflow และกดที่ Go Console จากนั้นก็เข้าสู่ขั้นตอนการ Login หรือลงทะเบียน



รูปที่ 2.12: หน้าหลักของ Dialogflow

2. สร้าง Agent

หลังจาก Login สำเร็จเราจะเจอกับ Workplace ในการทำแชทบอทலะ ให้ไปที่เมนูด้านซ้าย และเลือก Create Agent ก็จะพบกับหน้าจอสำหรับตั้งค่าแชทบอทของเรา โดยต้องสามารถตั้งชื่อ ภาษา และ Timezone ที่ต้องการ



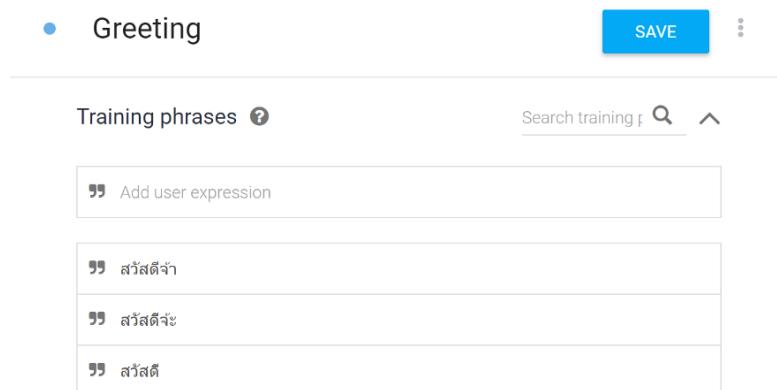
รูปที่ 2.13: หน้า Create Agent

3. สอนบอทให้พูดทักษะ

เมื่อสร้างเสร็จแล้วเราจะพบกับ Default Intents มา 2 ตัวก็คือ Default Welcome Intent และ Default Fallback Intent มาให้ ตอนนี้ยังไม่ต้องสนใจมันนะ เพราะเราจะลองสร้าง

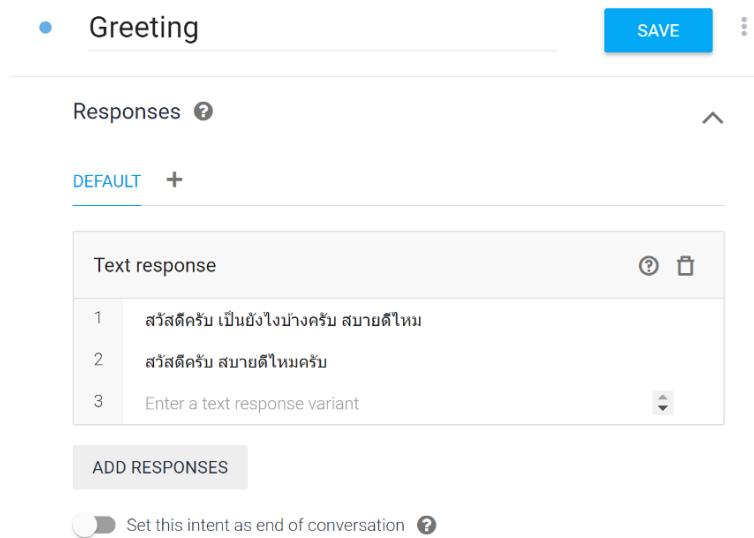
Intent ใหม่โดยตั้งชื่อว่า Greeting

ในการสร้างให้กดที่ปุ่ม Create Intent และตั้งชื่อ Intent นี้ว่า Greeting โดยเราตั้งใจให้ Intent นี้ โตตอบกับผู้ใช้งาน เวลาที่ผู้ใช้ต้องการที่จะทักทายกับzechbot ที่เราสร้างขึ้นมา จากนั้นไปที่ Training phrases หรือแนวประโยคที่เราจะให้zechbotเข้าใจว่า ถ้าพูดด้วย ประโยคประมาณนี้ แสดงว่าผู้ใช้งานตั้งใจจะสื่อถึง Intent นี้ ถ้าดูจากตัวอย่างจะพบว่ามีการระบุ phrases ไว้ว่า สวัสดี, สวัสดีจ้ะ, สวัสดีจ้า



รูปที่ 2.14: หน้า Intent ส่วน Training phrases

จากนั้นเราจะลองไปตั้งค่า Responses หรือประโยคที่เราต้องการให้zechbotตอบกลับ ในกรณีที่บอทสามารถจับได้ว่าผู้ใช้งานตั้งใจจะสื่อถึง Intent นี้ สำหรับตัวอย่างจะพบว่า ถ้าผู้ใช้พิมพ์ สวัสดี, สวัสดีจ้ะ, สวัสดีจ้า ตาม Training phrases เราจะให้zechbotของเราตอบกลับว่า สวัสดีครับ เป็นยังไงบ้างครับ สบายดีไหม หรือ สวัสดีครับ สบายดีไหมครับ โดยจะสุ่มขึ้นมาว่าจะตอบอันไหน

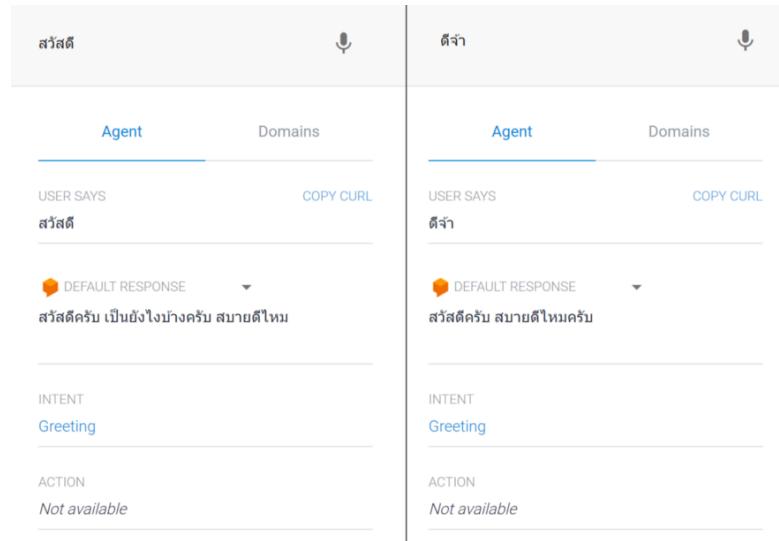


รูปที่ 2.15: หน้า Intent ส่วน Responses

ตรงส่วนของ Responses เราสามารถเพิ่มข้อความ หรือเพิ่ม balloon message ให้ต่อ กัน หลายๆ อันได้ โดยกดที่ปุ่ม Add Responses และถ้าต้องการตั้งค่าว่า intent นี้เป็น intent สุดท้ายในการสนทนากัน ก็สามารถเปิด Checkbox Set this intent as end of conversation ซึ่งเดียวเราค่อนมาคุยกันแบบละเอียดอีกครั้ง ตอนที่ต้องทำ Contexts กันอีกครั้ง

4. ทดสอบคุยกับบอท

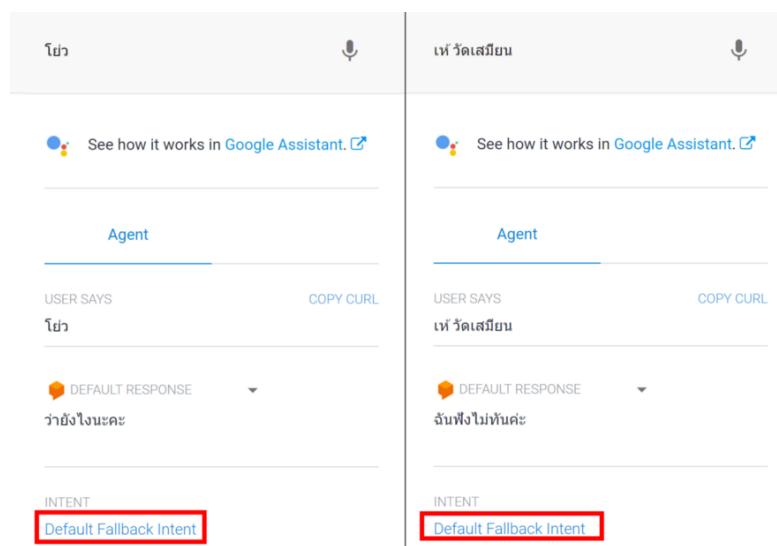
หลังจากเราลองทำ Greeting Intent เสร็จ ก็ถึงเวลาที่เราจะลองทดสอบการใช้งานกัน ซึ่ง เราสามารถทดสอบได้ผ่านกล่องสนทนาที่อยู่ทางด้านขวา โดยลองพิมพ์คำว่า สวัสดี ลงไป ก็ จะพบว่า textbooks จะตอบเรากลับมาว่า สวัสดีครับ เป็นยังไงบ้างครับ สบายดีไหม ตามที่เรา ตั้งค่าไว้ใน Responses นั้นเอง



รูปที่ 2.16: ทดสอบคุยกับบอท

ถ้าดูจากภาพจะพบว่า ถ้าเราพิมพ์คำบางคำที่ไม่ได้มีอยู่ใน Training phrases อุ่งคำว่า ดีเจ ตัว Dialogflow ก็ฉลาดพอที่จะจับได้ว่านี่คือคำที่อยู่ในกลุ่มเดียวกับ สวัสดี ซึ่งเป็นคำทักทาย ที่เรากำหนดไว้มันคือ Intent Greeting นั้นเอง

แต่ในขณะเดียวกัน คำบางคำ หรือประโยคบางประโยคตัวแซบทบทของเราก็อาจจะยังไม่เข้าใจ ว่าสิ่งที่ผู้ใช้งานต้องการจะสื่อสารอกรกما มันคือ Intent อะไร ซึ่งเวลาสร้าง Agent Dialogflow ก็จะสร้าง Default Fallback Intent ขึ้นมาให้ พร้อมกับ Responses บางส่วนในกรณีที่แซบทบทไม่สามารถหา Intent ที่เหมาะสมได้ ก็จะมาตกที่คेसนี้ทั้งหมดตามภาพนั้นเอง



ຮູບທີ 2.17: ທດສອບຄຸຍກັບບອທ

2.6 Libraries moment.js

moment.js គື່ອ javascript library ສໍາຮັບໃໝ່ສໍາຮັບຈັດການ Date ແລະ Time ທີ່ຈະຊ່ວຍ
ອໍານວຍຄວາມສະດວກໃນເຮືອງການຈັດ Format Date ໄຫ້ຮຽກກັບທີ່ເຮົາຕ້ອງການ ມີ Feature ທີ່ໜາກໜາຍ
ກຣອບຄຸມທີ່ໃຊ້ງານ Format ທີ່ Date , Time , Timezone , Standard Time , Local Time
ເປັນຕົ້ນ ຜຶ່ງການນຳ library moment.JavaScript ນາໃຊ້ງານສາມາຮັດຕິດຕັ້ງໄດ້ດັ່ງນີ້

1. ຄຳສັ່ງໃນການຕິດຕັ້ງ

```

1 /* install dependency */
2 npm install moment --save # npm
3 yarn add moment # Yarn
4 Install-Package Moment.js # NuGet
5 spm install moment --save # spm
6 meteor add momentjs:moment # meteor
7 bower install moment --save # bower (deprecated)

```

ຮູບທີ 2.18: ຄຳສັ່ງໃນການຕິດຕັ້ງ moment.js

2. ຮູ່ປະບົບການໃຊ້ງານ moment.js

```

1 const moment = require('moment');
2
3 const SLASH_DMY = 'DD/MM/YYYY';
4 const SLASH_DMYHMS = 'DD/MM/YYYY HH:mm:ss';
5 const SLASH_YMD = 'YYYY/MM/DD';
6 const SLASH_YMDHMS = 'YYYY/MM/DD HH:mm:ss';
7 const DASH_DMY = 'DD-MM-YYYY';
8 const DASH_DMYHMS = 'DD-MM-YYYY HH:mm:ss';
9 const DASH_YMD = 'YYYY-MM-DD';
10 const DASH_YMDHMS = 'YYYY-MM-DD HH:mm:ss';
11
12 console.log('sysdate ::==', moment());
13
14 console.log('sysdate ::==', moment().format(SLASH_DMY));
15 console.log('sysdate ::==', moment().format(
    SLASH_DMYHMS));
16
17 console.log('sysdate ::==', moment().format(SLASH_YMD));
18 console.log('sysdate ::==', moment().format(
    SLASH_YMDHMS));
19
20 console.log('sysdate ::==', moment().format(DASH_DMY));
21 console.log('sysdate ::==', moment().format(
    DASH_DMYHMS));
22
23 console.log('sysdate ::==', moment().format(DASH_YMD));
24 console.log('sysdate ::==', moment().format(
    DASH_YMDHMS));
25 /*
26 sysdate ::== moment("2018-07-03T21:08:38.248")
27 sysdate ::== 03/07/2018
28 sysdate ::== 03/07/2018 21:08:38
29 sysdate ::== 2018/07/03
30 sysdate ::== 2018/07/03 21:08:38
31 sysdate ::== 03-07-2018
32 sysdate ::== 03-07-2018 21:08:38
33 sysdate ::== 2018-07-03
34 sysdate ::== 2018-07-03 21:08:38
35 */

```

รูปที่ 2.19: รูปแบบการใช้งาน moment.js

ทั้งหมดที่ยังไม่ใช้ทั้งหมดที่ Moment JS ทำได้ยังมีความสามารถอีกหลายอย่างที่ยังไม่ได้กล่าวถึงที่ Moment ทำได้เข้าไปดูได้ที่ <https://momentjs.com/docs/>

2.7 Google Maps API

2.7.1 Google Maps คืออะไร

Google Maps คือ บริการแผนที่ของ Google ซึ่งให้บริการ Services ที่เกี่ยวข้องกับแผนที่ทั้งหมด โดยในปัจจุบันแผนที่ของ Google นั้นมีอยู่หลากหลายประเภท อาทิ เช่น ที่เราใช้บริการแผนที่บนเว็บไซต์ หรือ App บน Smartphone โดย Services เหล่านี้เรามารู้ว่า เราสามารถเรียกใช้ได้ฟรีในกรณีที่ผ่าน Application ทั่วไป แต่ถ้าในกรณีที่เราจะมีการเรียกใช้งานในเว็บไซต์ หรือแอปพลิเคชันที่พัฒนาขึ้นเอง Google Maps ก็จะมี API ให้ใช้งานได้ เช่นเดียวกัน แต่ Services ของ Google นั้นมีข้อจำกัดในการใช้งาน ถ้าต้องการใช้ในปริมาณที่สูงขึ้น ก็จะต้องซื้อ Package ที่ทาง Google Maps มีมาให้ โดยปกติจะมีการจำกัดจำนวนที่ Request เข้ามาเรียกใช้งาน สำหรับเรียกใช้แผนที่และชุด service ต่าง ๆ ของ Google เพื่อพัฒนา Application ได้เหมือนกับ Google มีบริการ features ให้เรียกใช้ดังต่อไปนี้

- การปรับแต่งแผนที่ (Styled Map)
- ชุดควบคุมแผนที่ (Map Control)
- ชุดเครื่องมือวาดภาพบนแผนที่ (Drawing)
- การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service)
- การคำนวณความสูงของจุดพิกัด (Elevation Service)
- การแปลงที่อยู่เป็นพิกัด Latitude และ Longitude (GeoCoding Service)
- การดึงข้อมูล POI (Point of Interest)
- Street View

2.7.2 ประโยชน์ของการใช้งาน Google Maps

- สามารถค้นหาสถานที่ต่าง ๆ ได้อย่างสะดวกรวดเร็ว
- สามารถค้นหาชื่อถนนและสีแยกได้
- สามารถค้นหาร้านอาหารในพื้นที่ที่ต้องการได้
- สามารถที่จะประชาสัมพันธ์สถานประกอบการทางธุรกิจ
- สามารถย่อหรือขยายแผนที่ทั่วโลกให้เล็กลงได้

- สามารถวางแผนเส้นทางการเดินทางไปยังพื้นที่ต่าง ๆ ได้
- สามารถใช้งานด้านระบบวิทยาในการค้นหาแหล่งเพร่เชื้อ
- สามารถทำแผนที่หรือเส้นทางไปบ้านของตนเองได้
- สามารถแสดงตำแหน่งของตนเองได้
- สามารถดูและมองเห็นแผนที่ต่าง ๆ ทั่วโลกได้อย่างรวดเร็ว

2.7.3 การนำ Google Maps API มาใช้งาน

สิ่งที่ต้องทำเพื่อเรียก Google Maps API มาใช้งานดังนี้

- (a) ทำการสมัครขอ API KEY โดยสามารถเข้าไปสมัครได้ที่ <https://console.developers.google.com>
- (b) เพิ่มคำสั่งในบรรทัดที่ 2 ไว้ในไฟล์ index.html

```

1 <head>
2 <script src="https://maps.googleapis.com/maps/api/js
   ?key={API_KEY}" async defer></script>
3 </head>
```

รูปที่ 2.20: เรียกใช้ API KEY

- (c) ขั้นตอนการเรียกใช้งานสำหรับแสดง Google Maps

```

1 <-- IN HTML -->
2
3 <div #map id="map"></div> // เรียกใช้ map เพื่อแสดง
4
5 <-- IN TYPESCRIPT -->
6
7 map:any; กำหนดตัวแปร// map
8
9 ionViewDidLoad() {
10   this.initMap();
11 } // เรียกใช้งานเมื่อหน้าจอเรียก
12
13 initMap() {
14   this.map = new google.maps.Map(this.mapElement.nativeElement, {
15     zoom: 7,
16     center: {lat: 41.85, lng: -87.65}
17   });
18   this.directionsDisplay.setMap(this.map);
19 } // ฟังก์ชันสำหรับกำหนด map
20
21 <-- IN SCSS -->
22   #map {
23     height: 100%;
24   } // ขนาดของ map

```

รูปที่ 2.21: เรียกใช้งาน google maps ใน ionic framework

(d) เรียกใช้งานด้วยคำสั่ง Google Maps เพื่อใช้งานตามต้องการ

2.8 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.8.1 แอปพลิเคชันเครือข่ายสังคมออนไลน์เพื่อผู้สูงอายุ (OLDSTER)

แอปพลิเคชันเครือข่ายสังคมออนไลน์เพื่อผู้สูงอายุ (OLDSTER) [?] เป็นแอปพลิเคชันที่ถูกออกแบบมาเป็นเครือข่ายสังคมของผู้สูงอายุ โดยแอปถูกออกแบบเพื่อผู้สูงอายุเป็นหลัก จึงง่ายต่อการใช้งาน มีรูปแบบที่ไม่ยุ่งยาก ไม่ซับซ้อน ตัวหนังสือเป็นแบบที่จ่ายต่อการอ่าน และขนาดไม่เล็กเกินไป มีฟังก์ชันการทำงาน 9 ฟังก์ชันดังนี้ ข้อมูลส่วนตัว กระดาษข่าว เช ระหว่างคน 2 คน การแจ้งเตือน วิธีใช้งาน ข่าวสารในหมวดต่าง ๆ เป็นต้น ความแตกต่างระหว่างแอปพลิเคชัน OLDSTER กับ OLD MY FRIENDS อยู่ตรงที่แอป OLD MY FRIENDS จะมีการให้ความรู้ในเรื่องโรคที่ผู้สูงอายุมักพบเจอในรูปแบบของเชทบอท และ

ยังมีการแข่งแบบกลุ่ม การค้นหาตำแหน่งของคนในครอบครัว รวมทั้งมีการแจ้งเตือนการท่านยาอีกด้วย



รูปที่ 2.22: หน้าหลักของ OLDSTER

2.8.2 แอปพลิเคชัน KINYA

KINYA [?] เป็นแอปพลิเคชันที่ให้ความรู้ในเรื่องโรคต่าง ๆ ด้วยบทความ และช่วยแจ้งเตือนการท่านยาไว้สำหรับป้องกันการลืมทานยาของผู้สูงอายุ อีกทั้งยังมีฟังก์ชันการนัดพบแพทย์โดยผู้ใช้จะต้องเป็นผู้กรอกข้อมูลเองด้วย ความแตกต่างระหว่าง KINYA กับ OLD MY FRIENDS คือเป้าหมายที่แตกต่างกันนั่นคือ KINYA จะเน้นการการแจ้งเตือนการ

ท่านยาและการพบแพที่ ส่วน OLD MY FRIENDS จะเน้นให้ความรู้ในเรื่องโรคโดยใช้
เชิงบอท และการแบ่งปันประสบการณ์ด้วยกระดาษข่าว



รูปที่ 2.23: หน้าหลักของ KINYA

ที่มา : <https://play.google.com/store/apps/details?id=com.idoctor.kinyahl=th>

บทที่ 3

การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบก่อนดำเนินการจริงเป็นอีกหนึ่งขั้นตอนที่มีความสำคัญมาก เพราะการวิเคราะห์และออกแบบระบบนั้นเป็นการกระทำที่ทำให้ผู้พัฒนาเห็นรายละเอียดส่วนย่อยของงานทั้งหมด เพิ่มประสิทธิภาพในการวางแผนการทำงาน และยังช่วยลดปัญหาที่อาจจะเกิดขึ้นในระหว่างพัฒนา เพื่อให้ระบบมีความสมบูรณ์มากยิ่งขึ้น เนื่องจาก การวิเคราะห์และออกแบบระบบนั้นจะช่วยให้ให้บริการ จัดการทรัพยากรได้อย่างคุ้มค่า และตรงตามความต้องการของระบบ

การวิเคราะห์และออกแบบแอปพลิเคชันสูงวัยมายเฟรนด์ ในบทนี้จะแบ่งออกเป็น 6 ขั้นตอนเพื่อให้เห็นการดำเนินงานอย่างมีระบบ ในหัวข้อแรกจะนำเสนอภาพรวมของระบบ ก่อนจะนำเสนอเอกสารแสดงความต้องการของระบบซึ่งจะทำให้เห็นที่มาของเพจต่าง ๆ ใน ขั้นตอนของการออกแบบในหัวข้อที่สาม ส่วนหัวข้อที่เหลือจะแสดงแผนภาพการทำงานของระบบโดยใช้ UML diagram ซึ่งประกอบไปด้วย Use Case, Class และ Sequence Diagram เพื่อแสดงรายละเอียดของระบบก่อนนำไปเขียนคำสั่งด้วยภาษาโปรแกรมในบทต่อไป

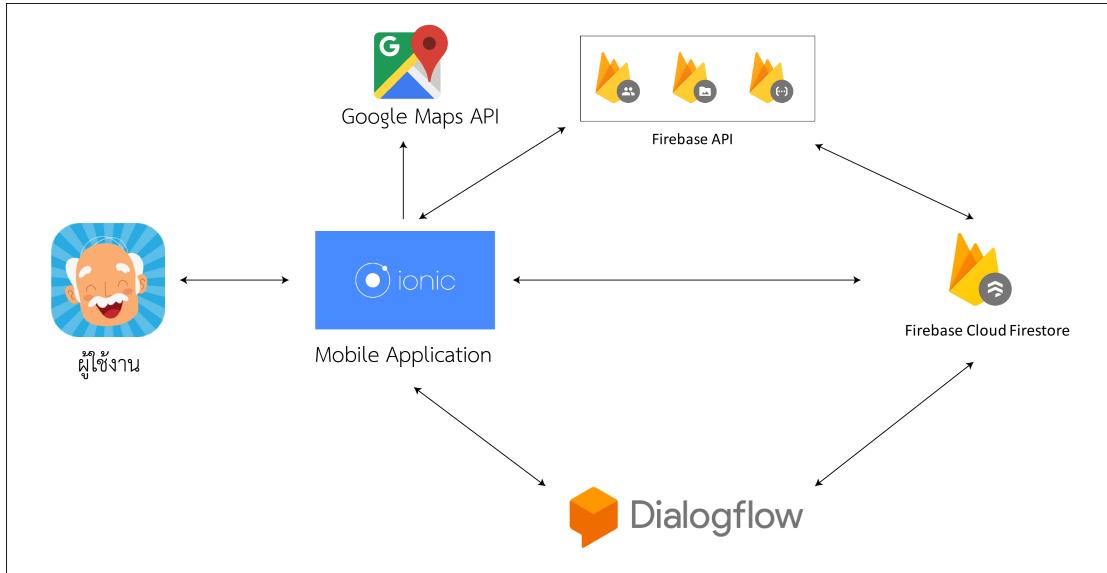
- 3.1 โครงสร้างภาพรวมของระบบ (System Architecture)
- 3.2 การวิเคราะห์ความต้องการของระบบ (System Requirements)
- 3.3 การออกแบบส่วนติดต่อผู้ใช้ (User Interface Design)
- 3.4 แผนภาพโดยรวม
 - 3.3.1 ยูสเคสโดยรวม (Use Case Diagram)
 - 3.3.2 คลาสโดยรวม (Class Diagram)
 - 3.3.3 ซีเควสโดยรวม (Sequence Diagram)

3.1 โครงสร้างภาพรวมของระบบ

ความหมายของ System Architecture [?] หมายถึง กรอบโครงสร้างของระบบที่อธิบายความสัมพันธ์ขององค์ประกอบต่าง ๆ ไปจนถึงขั้นการเชื่อมตอกันของระบบอย่างต่อเนื่อง โดยจัดกลุ่มองค์ประกอบไว้ในหลาย ๆ ลักษณะเพื่อให้ผู้เกี่ยวข้อง (Stakeholder) จากพื้นฐาน

สาขาอาชีพที่แตกต่าง กันสามารถทำความเข้าใจได้ง่าย เช่น การจัดแบ่งองค์ประกอบตามลักษณะการทำงานของระบบ (functional components) เป็นต้น

การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของแอปพลิเคชันสูงวัยมายเฟรนด์ มีรายละเอียดดังรูปที่ 3.1



รูปที่ 3.1: System architecture แอปพลิเคชันสูงวัยมายเฟรนด์

จากรูปที่ 3.1 สามารถอธิบายโครงสร้างและเทคโนโลยีของระบบโดยแบ่งเป็น 3 ส่วนหลักดังนี้

- Database ระบบใช้บริการฐานข้อมูลแบบ NoSQL ของเฟร์เบสชื่อ Cloud Firestore
- Server กระบวนการทำงานในส่วนของเซิฟเวอร์ (server) แบ่งเป็น 3 ส่วนได้แก่
 - Dialogflow เป็น Platform ไว้สำหรับจัดการการโต้ตอบอัตโนมัติหรือเชทบอท
 - Google Maps Api เป็น Api ของ Google ไว้สำหรับเรียกใช้งาน Google Maps เพื่อใช้งานแผนที่
 - ชุดบริการเฟร์เบส Api ใช้สำหรับการทำงานกับบริการต่าง ๆ ของเฟร์เบสบนแพลตฟอร์มที่แตกต่างกัน เช่น Authentication ใช้สำหรับการจัดการข้อมูลผู้ใช้ หรือเฟร์เบส Storage ที่ใช้สำหรับจัดเก็บไฟล์เอกสารและรูปภาพต่าง ๆ เป็นต้น
- Client App Mobile แอปพลิเคชันทำงานบนอุปกรณ์พกพาสามารถใช้ได้ทั้งแอนดรอยด์และไอโอเอส

3.2 การวิเคราะห์ความต้องการของระบบ

3.2.1 ความต้องการหลักของระบบ (Functional Requirements)

แอปพลิเคชันสูงวัยมายเฟรนด์ แบ่งความสามารถของระบบดังนี้

(a) ผู้ใช้งาน

- สามารถสมัครสมาชิกและเข้าสู่ระบบได้
- สามารถดู สร้าง แก้ไข ลบ โพสท์ได้
- สามารถคุยกับตอบกับแขกบอทได้
- สามารถดูตำแหน่งของคนในครอบครัวได้
- สามารถส่งแจ้งเตือนรูปแบบการโทร หรือข้อความ ไปหาคนในครอบครัวได้
- สามารถเพิ่ม ลบ เพื่อนได้
- สามารถรับการแจ้งเตือนจากการทานยาได้
- สามารถดูและแก้ไขข้อมูลส่วนตัวได้

3.2.2 Non-functional Requirements

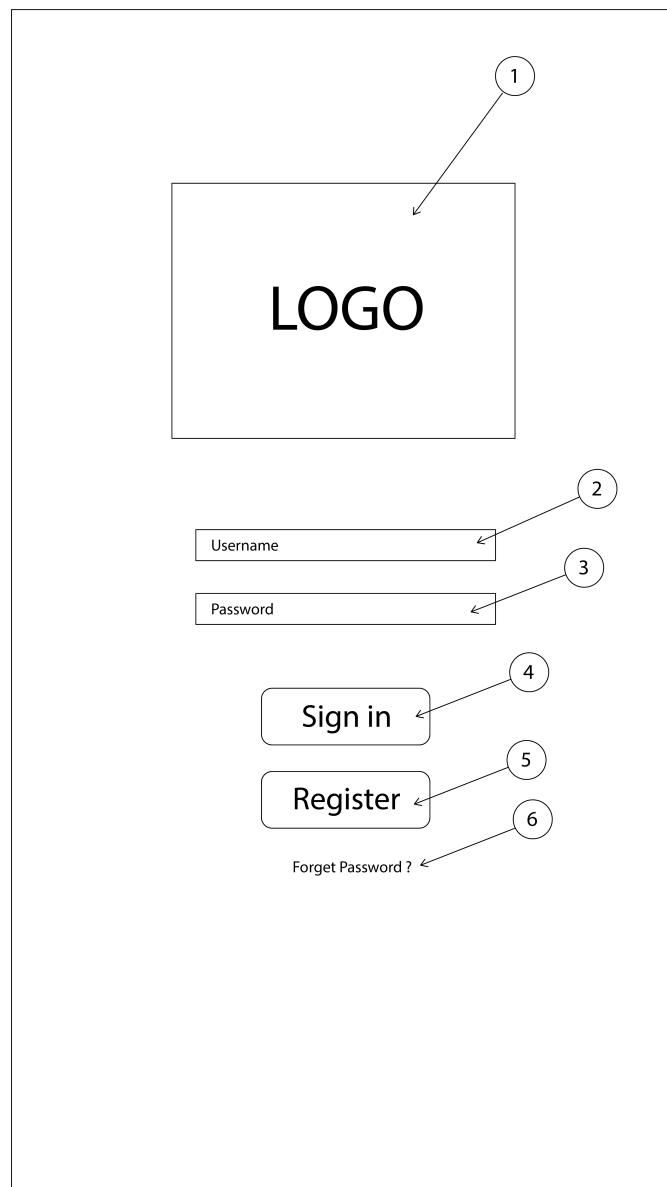
(a) แอปพลิเคชัน

- แขกบอทสามารถตอบโต้ได้ตรงประเด็น
- มีความรวดเร็วในการกดไลท์
- สามารถใช้งานได้รวดเร็ว
- สามารถดูตำแหน่งของคนในครอบครัวแบบ Realtime
- สามารถแจ้งเตือนการทานยาได้เมื่อปิดแอปพลิเคชันอย่างสมบูรณ์

3.3 การออกแบบส่วนติดต่อผู้ใช้

ในการออกแบบส่วนติดต่อผู้ใช้ของแอปพลิเคชันสูงวัยมายเฟรนด์ ประกอบด้วยส่วนต่าง ๆ ดังนี้

3.3.1 หน้าจอเข้าสู่ระบบ

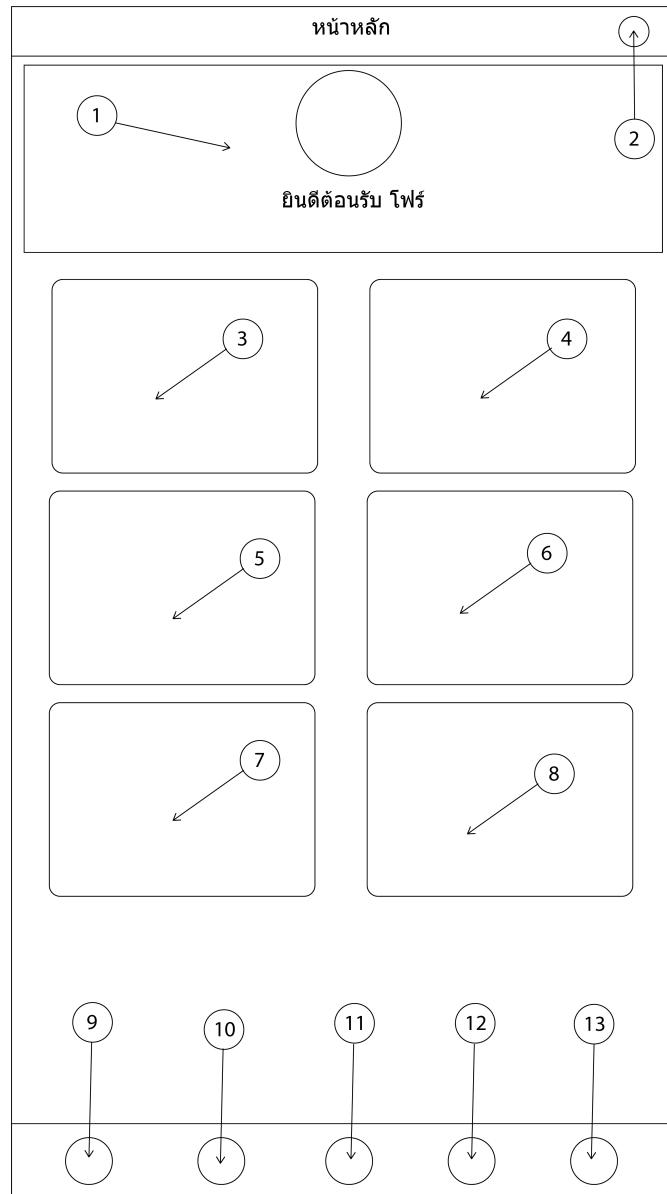


รูปที่ 3.2: หน้าจอเข้าสู่ระบบ

จากภาพที่ 3.2 การออกแบบหน้าจอเข้าสู่ระบบประกอบไปด้วย 6 ส่วนดังนี้

- ส่วนที่ 1 โลโก้ของแอปพลิเคชัน
- ส่วนที่ 2 ช่องสำหรับกรอกชื่อผู้ใช้
- ส่วนที่ 3 ช่องสำหรับกรอกรหัสผ่าน
- ส่วนที่ 4 ปุ่มสำหรับเข้าสู่ระบบ

- ส่วนที่ 5 ปุ่มสำหรับสมัครสมาชิก
- ส่วนที่ 6 ข้อความสำหรับกดเมื่อรีบส่งผ่าน

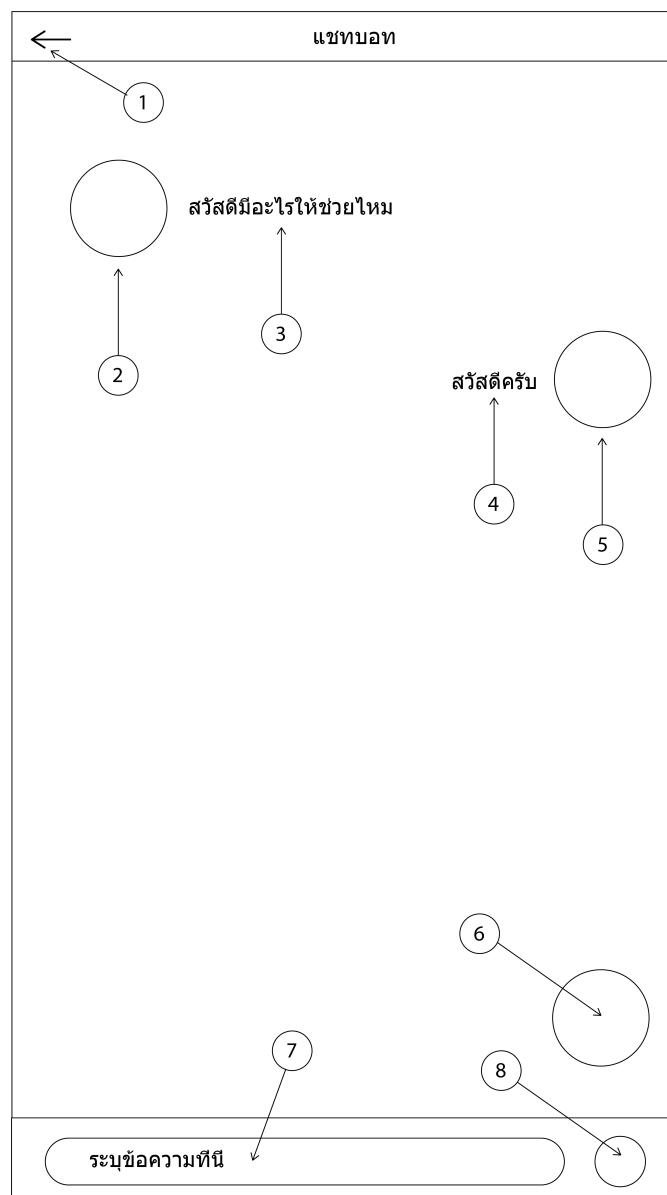


รูปที่ 3.3: หน้าจอหลัก

จากภาพที่ 3.3 การออกแบบหลักประกอบไปด้วย 13 ส่วนดังนี้

- ส่วนที่ 1 ข้อมูลของผู้ใช้ประกอบไปด้วยรูปประจำตัวและชื่อผู้ใช้
- ส่วนที่ 2 ปุ่มสำหรับออกจากระบบ
- ส่วนที่ 3 ปุ่มสำหรับไปยังหน้าปัจจุบันที่เป็นแซทบอท

- ส่วนที่ 4 ปุ่มสำหรับไปยังหน้ากระดานข่าว
- ส่วนที่ 5 ปุ่มสำหรับไปยังหน้าครอบครัว
- ส่วนที่ 6 ปุ่มสำหรับไปยังหน้าตำแหน่งของครอบครัว
- ส่วนที่ 7 ปุ่มสำหรับไปยังหน้าฉุกเฉิน
- ส่วนที่ 8 ปุ่มไปยังหน้าวิธีใช้งาน
- ส่วนที่ 9 ปุ่มไปยังหน้าหลักเราจะอยู่หน้านี้ทุกครั้งเมื่อเราเข้าสู่แอปพลิเคชัน
- ส่วนที่ 10 ปุ่มสำหรับไปยังหน้าเพื่อนและครอบครัว
- ส่วนที่ 11 ปุ่มเพื่อโพสท์ข้อความหรือรูปภาพ
- ส่วนที่ 12 ปุ่มสำหรับไปหน้าตั้งค่า
- ส่วนที่ 13 ปุ่มสำหรับไปยังหน้าໂປຣີຟັບໜຸລສ່ວນຕົວ

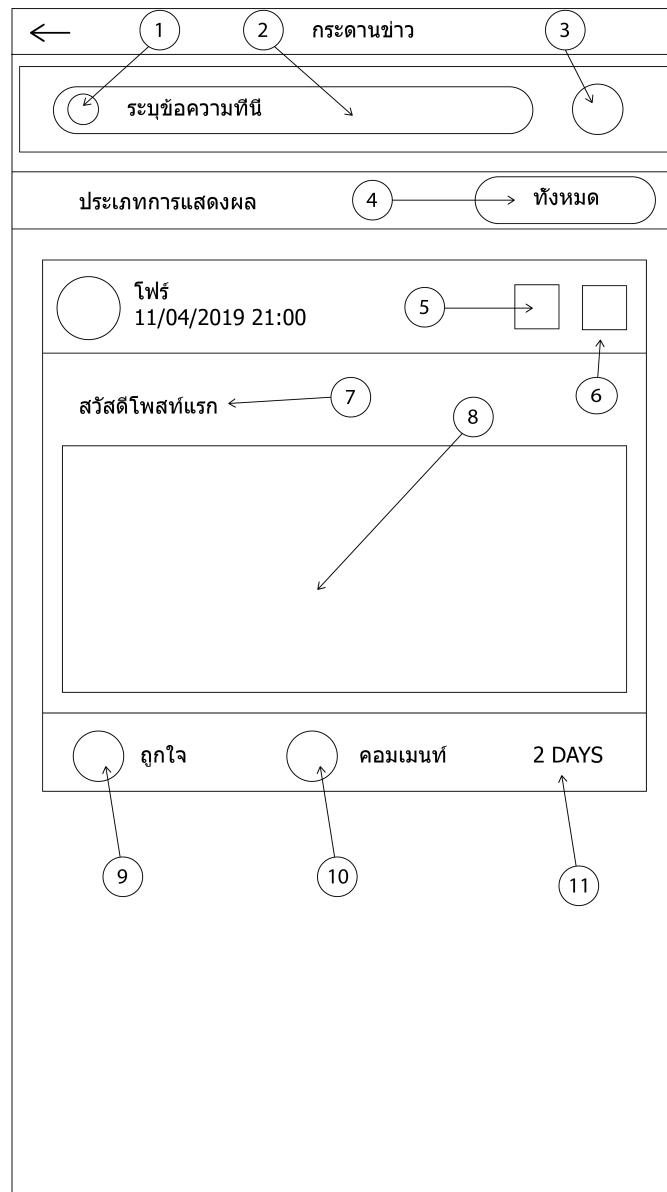


ຮູບທີ່ 3.4: ຮຳເນັດທບອທຄຸຍກັບປູ່ຈອ້ອນ

ຈາກພາກທີ່ 3.4 ການອອກແບບໜັກປະກອບໄປດ້ວຍ 8 ສ່ວນດັ່ງນີ້

- ສ່ວນທີ່ 1 ປູ່ມຸລືກສຽກລັບໄປຢັງໜ້າຈອ້ອນ
- ສ່ວນທີ່ 2 ຮູ່ປະຈຳຕົວຂອງປູ່ຈອ້ອນ
- ສ່ວນທີ່ 3 ຂ້ອຄວາມທີ່ປູ່ຈອ້ອນຕອບກັບຜູ້ໃຊ້
- ສ່ວນທີ່ 4 ຂ້ອຄວາມຂອງຜູ້ໃຊ້
- ສ່ວນທີ່ 5 ຮູ່ປະຈຳຕົວຂອງຜູ້ໃຊ້

- ส่วนที่ 6 ปุ่มสำหรับพิมพ์ข้อความด้วยเสียง (Speech to text)
- ส่วนที่ 7 ช่องสำหรับกรอกข้อความที่จะแสดงในส่วนที่ 4
- ส่วนที่ 8 ปุ่มสำหรับยืนยันข้อความส่วนที่ 7

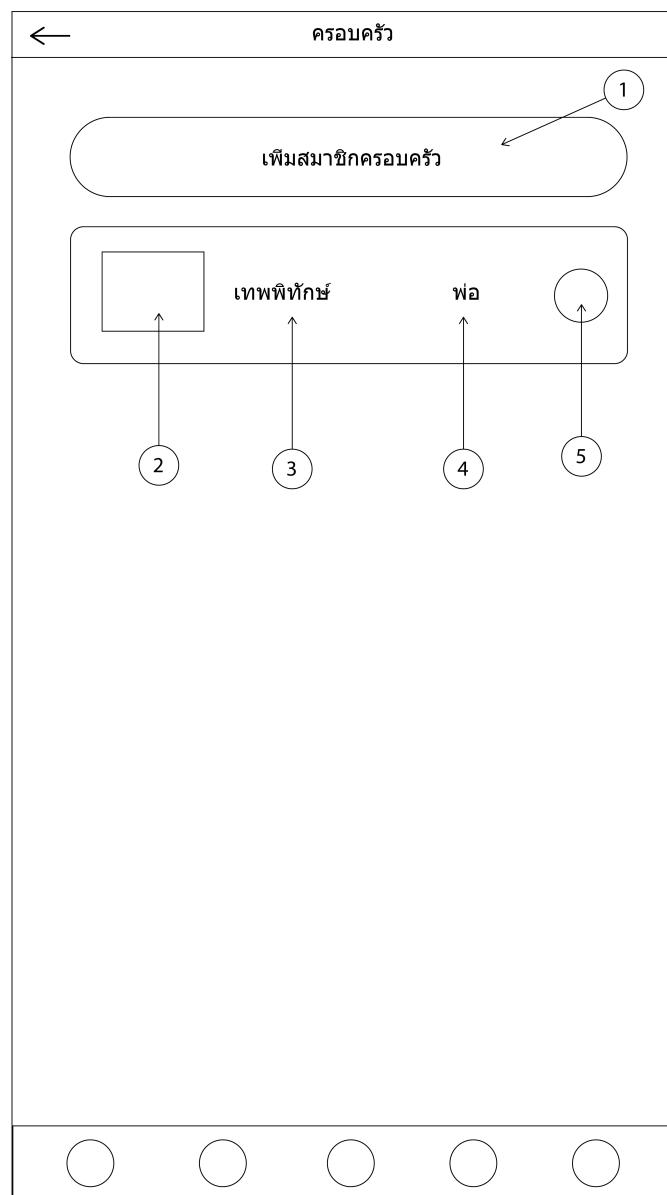


รูปที่ 3.5: หน้าจอกระดาษขาว

จากภาพที่ 3.5 การออกแบบหลักประกอบไปด้วย 11 ส่วนดังนี้

- ส่วนที่ 1 ปุ่มสำหรับเพิ่มรูปภาพสำหรับโพสท์มีสองส่วนคือ ถ่ายรูปและเลือกรูปจากแกลอรี่

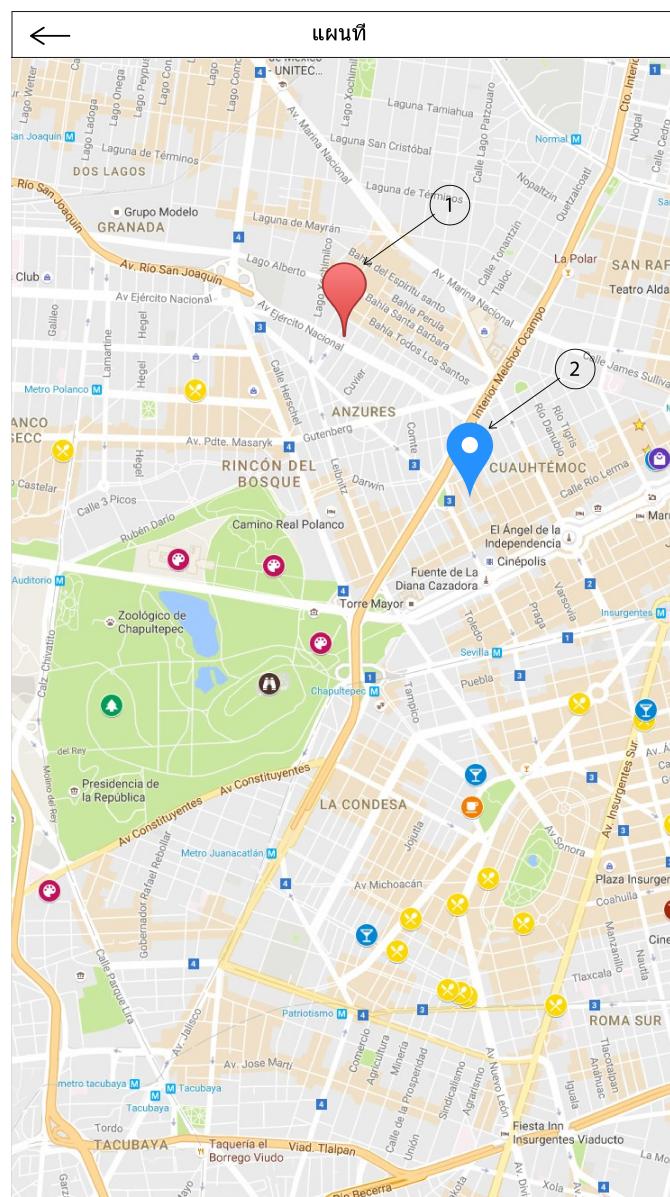
- ส่วนที่ 2 ช่องสำหรับกรอกข้อความที่จะโพสท์
- ส่วนที่ 3 ปุ่มสำหรับยืนยันการโพสท์หลังจากคลิกจะมีการเลือกประเภทได้ 3 แบบคือ กีฬา, ศาสนา, ดนตรี
- ส่วนที่ 4 ลิสท์สำหรับเลือกแสดงเฉพาะประเภทที่ต้องการมี 4 แบบได้แก่ ทั้งหมด, กีฬา, ศาสนา, ดนตรี
- ส่วนที่ 5 ปุ่มสำหรับแก้ไขโพสท์
- ส่วนที่ 6 ปุ่มสำหรับลบโพสท์
- ส่วนที่ 7 ข้อความโพสท์ของผู้ใช้
- ส่วนที่ 8 รูปภาพโพสท์ของผู้ใช้
- ส่วนที่ 9 ปุ่มสำหรับกดถูกใจ
- ส่วนที่ 10 ปุ่มสำหรับแสดงความคิดเห็น
- ส่วนที่ 11 ข้อความแสดงระยะเวลาหลังจากโพสท์ถูกสร้าง



รูปที่ 3.6: หน้าครอบครัว

จากภาพที่ 3.6 การออกแบบหลักประกอบไปด้วย 5 ส่วนดังนี้

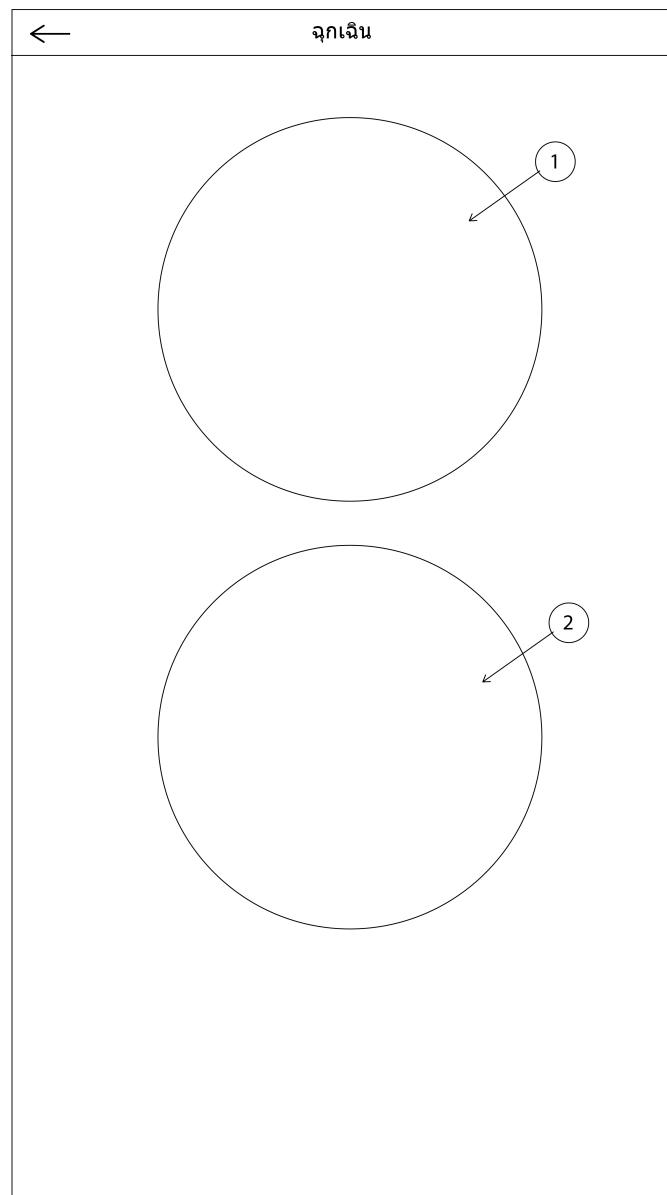
- ส่วนที่ 1 ปุ่มสำหรับเพิ่มสมาชิกในครอบครัว
- ส่วนที่ 2 รูปภาพของสมาชิกในครอบครัว
- ส่วนที่ 3 ชื่อของสมาชิกในครอบครัว
- ส่วนที่ 4 สถานะของครอบครัว
- ส่วนที่ 5 ปุ่มสำหรับจัดการสมาชิกครอบครัวได้แก่ แก้ไขสถานะ ลบสมาชิก



รูปที่ 3.7: หน้าแสดงตำแหน่งของครอบครัว

จากภาพที่ 3.7 การออกแบบหลักประกอบไปด้วย 2 ส่วนดังนี้

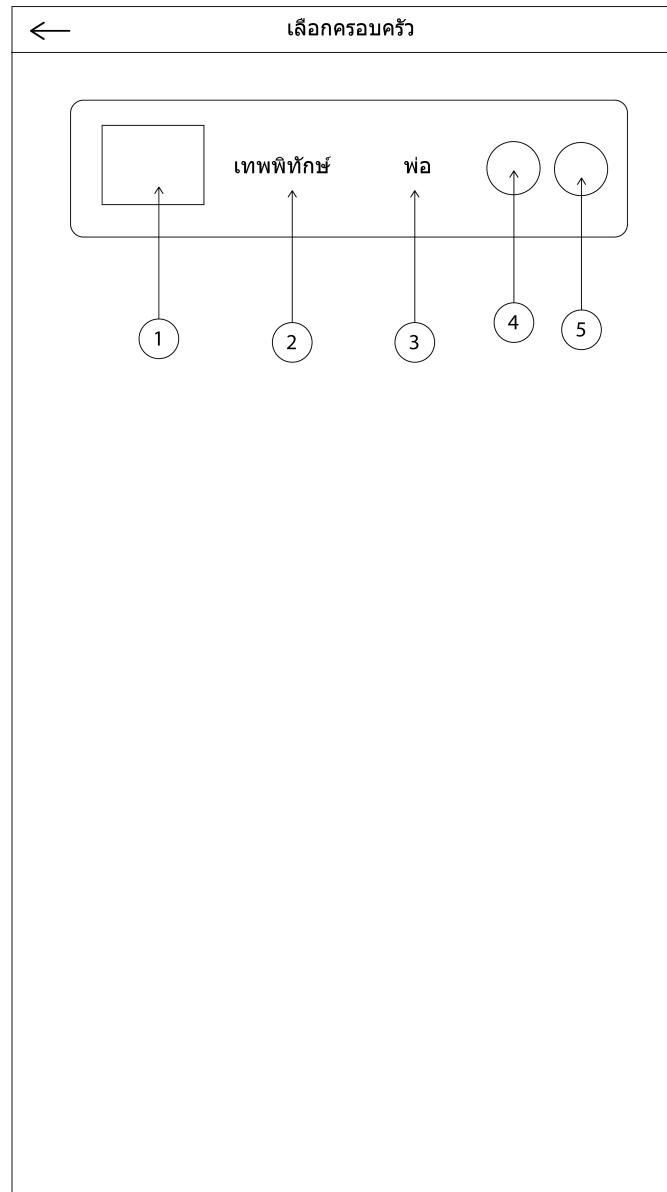
- ส่วนที่ 1 แสดงตำแหน่งของผู้ใช้
- ส่วนที่ 2 แสดงตำแหน่งของสมาชิกในครอบครัว



รูปที่ 3.8: หน้าฉกเฉิน

จากภาพที่ 3.8 การออกแบบหลักประกอบไปด้วย 2 ส่วนดังนี้

- ส่วนที่ 1 ปุ่มสำหรับแสดงเสียงเพื่อขอความช่วยเหลือ
- ส่วนที่ 2 ปุ่มสำหรับป้ายหน้าเลือกครอบครัวเพื่อขอความช่วยเหลือจากครอบครัว

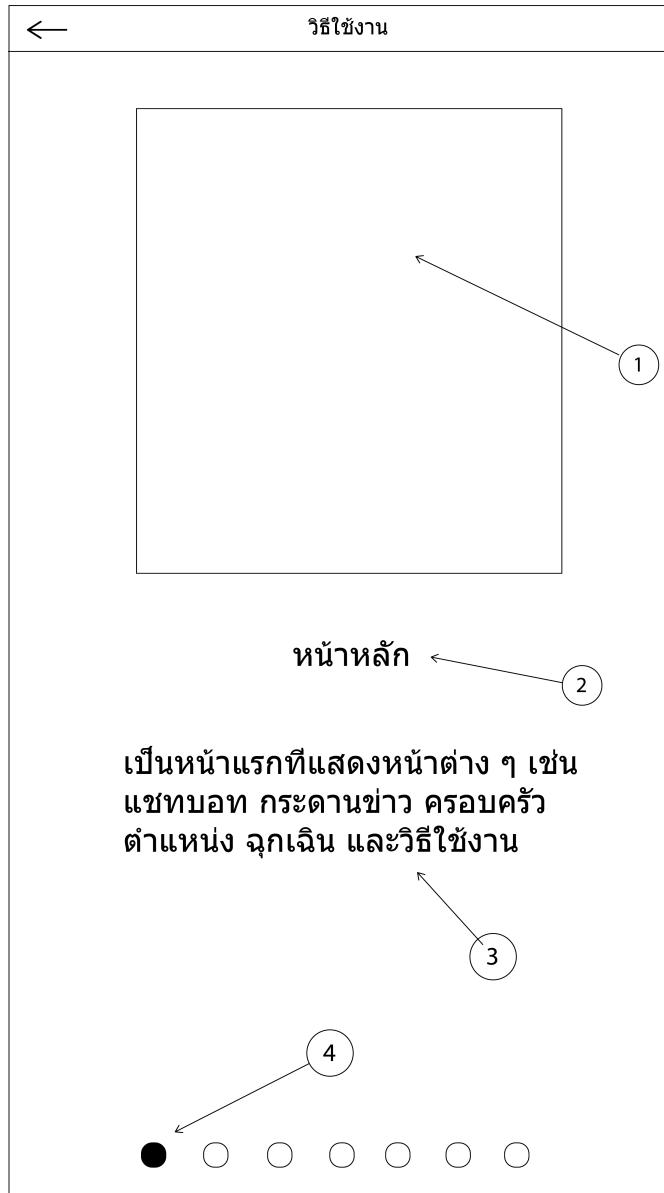


รูปที่ 3.9: หน้าเลือกครอบครัว

จากภาพที่ 3.9 การออกแบบหลักประกอบไปด้วย 5 ส่วนดังนี้

- ส่วนที่ 1 รูปประจำตัวของสมาชิกในครอบครัว
- ส่วนที่ 2 ชื่อของสมาชิกในครอบครัว
- ส่วนที่ 3 สถานะของครอบครัว
- ส่วนที่ 4 ปุ่มสำหรับໂທไปยังหมายเลขของสมาชิกในครอบครัว
- ส่วนที่ 5 ปุ่มสำหรับส่งข้อความไปยังสมาชิกในครอบครัวมีข้อความดังนี้ ”ช่วยด้วย !!!”

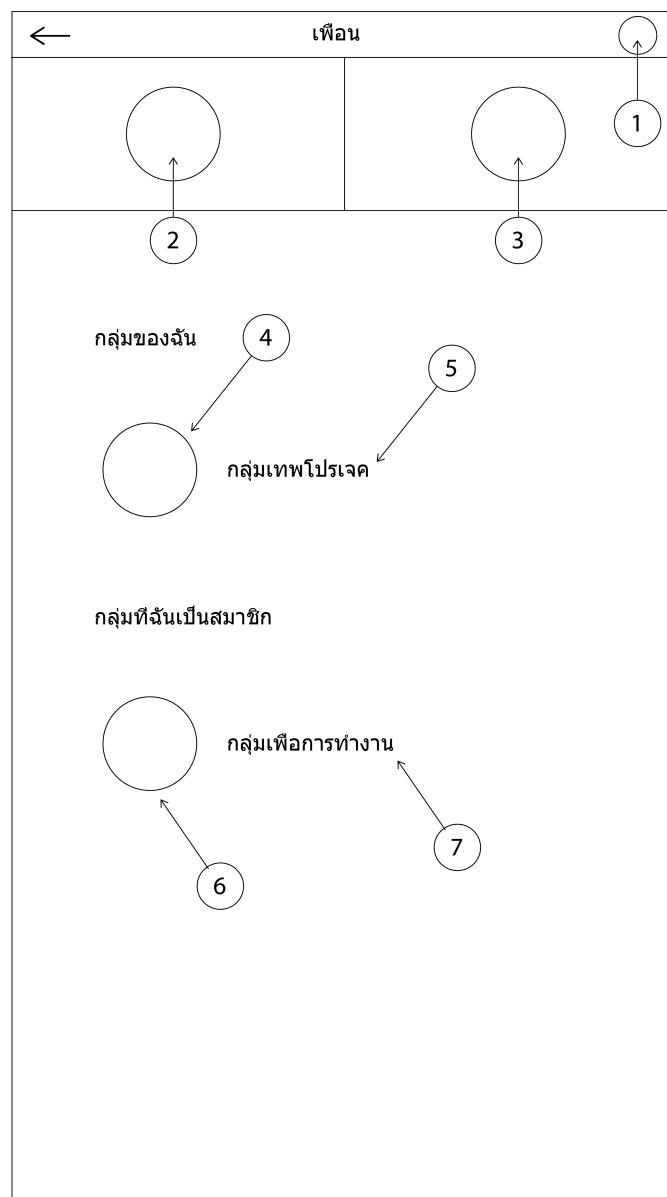
នឹង [ចិះដូងឲ្យ] ឡើង ពួននីមីបញ្ចាថេទិន្នន័យតិចតែកលប់មាតិ [បេរុះដូងឲ្យ] តាមនាម គោល។”



រូបទី 3.10: អាហារិធម្មង់

จากภาพទី 3.10 ការអកបែបលកប្រភកបែបដោយ 4 សំណង់នេះ

- សំណង់ទី 1 រូបភាពព័វយ៉ាងនៃការអកលកជាន់នៅ
- សំណង់ទី 2 ចិះខែនការណ៍នៅ
- សំណង់ទី 3 រាយឥឡូវបិះពិមិននៃការអកជាន់នៅ
- សំណង់ទី 4 តុដសាមរែបនៃការអកនៃការអកជាន់នៅ

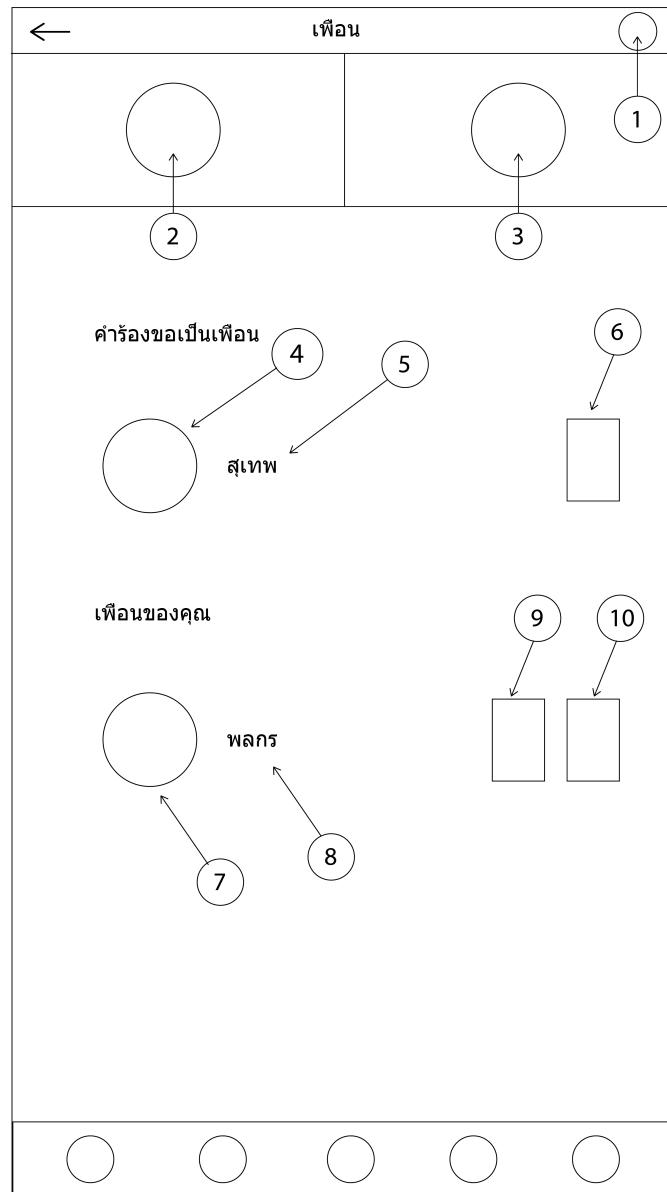


รูปที่ 3.11: หน้าจอแสดงกลุ่ม

จากภาพที่ 3.11 การออกแบบหลักประกอบไปด้วย 7 ส่วนดังนี้

- ส่วนที่ 1 ปุ่มสำหรับเพิ่มเพื่อน
- ส่วนที่ 2 ปุ่มสำหรับไปยังหน้าจอแสดงกลุ่ม
- ส่วนที่ 3 ปุ่มสำหรับไปยังหน้าจอแสดงเพื่อน
- ส่วนที่ 4 รูปภาพของกลุ่มที่เราเป็นคนสร้าง
- ส่วนที่ 5 ชื่อของกลุ่มที่เราเป็นคนสร้าง

- ส่วนที่ 6 รูปภาพของกลุ่มที่เราเป็นสมาชิก
- ส่วนที่ 7 ชื่อของกลุ่มที่เราเป็นสมาชิก



รูปที่ 3.12: หน้าจอแสดงเพื่อน

จากภาพที่ 3.12 การออกแบบหลักประกอบไปด้วย 8 ส่วนดังนี้

- ส่วนที่ 1 ปุ่มสำหรับเพิ่มเพื่อน
- ส่วนที่ 2 ปุ่มสำหรับไปยังหน้าแสดงกลุ่ม
- ส่วนที่ 3 ปุ่มสำหรับไปยังหน้าจอแสดงเพื่อน

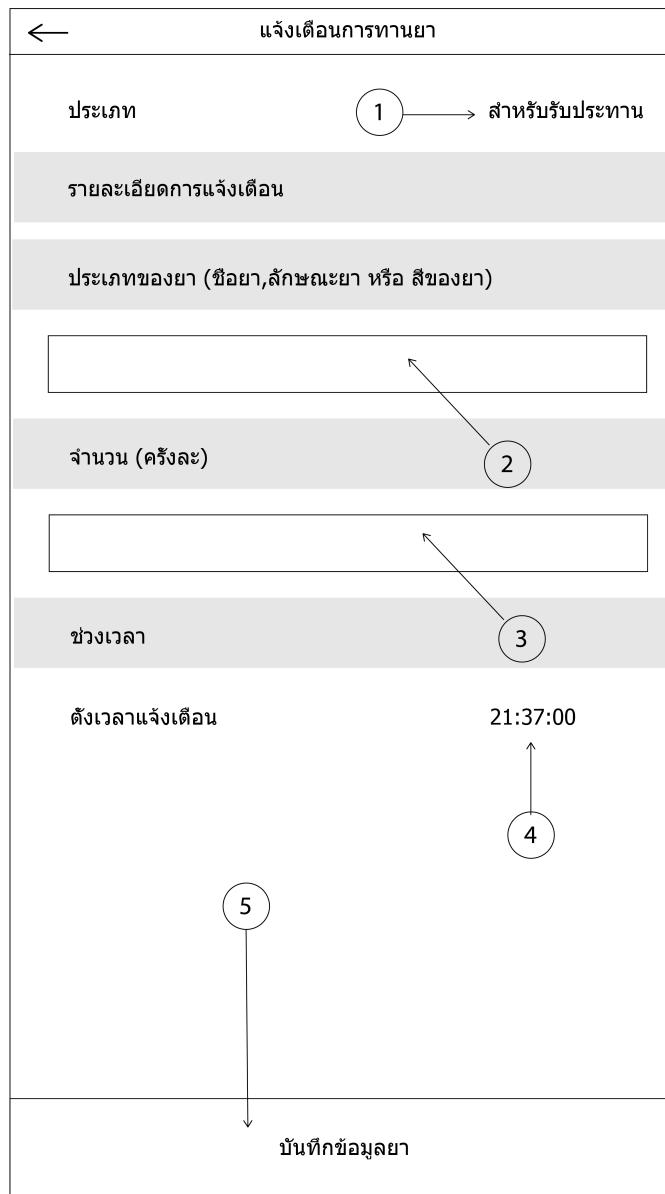
- ส่วนที่ 4 รูปภาพของเพื่อนที่รอการยืนยัน
- ส่วนที่ 5 ชื่อของเพื่อนที่รอการยืนยัน
- ส่วนที่ 6 ปุ่มสำหรับยกเลิกคำขอร้องเป็นเพื่อน
- ส่วนที่ 7 รูปภาพของเพื่อน
- ส่วนที่ 8 ชื่อของเพื่อน
- ส่วนที่ 9 ปุ่มสำหรับเช็คกับเพื่อน
- ส่วนที่ 10 ปุ่มสำหรับลบเพื่อน



รูปที่ 3.13: หน้าแสดงรายการแจ้งเตือนการทานยา

จากภาพที่ 3.13 การออกแบบหลักประกอบไปด้วย 4 ส่วนดังนี้

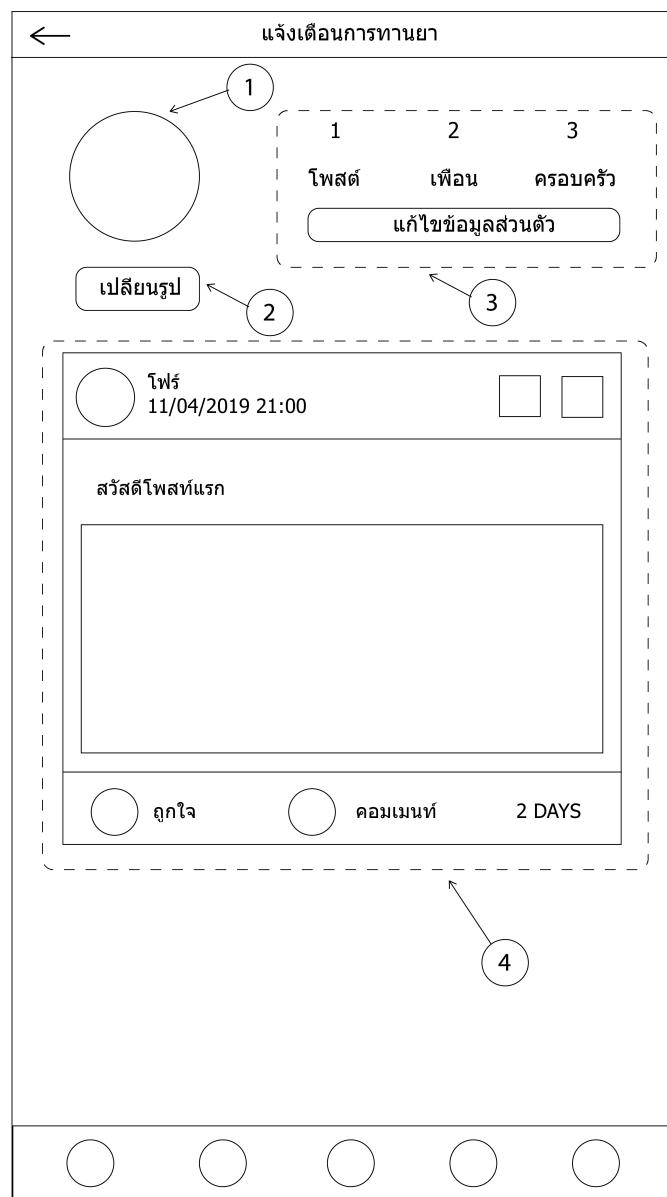
- ส่วนที่ 1 ข้อความแสดงชื่อของยา
- ส่วนที่ 2 ข้อความแสดงจำนวนการรับประทานต่อครั้ง
- ส่วนที่ 3 ข้อความแสดงเวลาในการแจ้งเตือน
- ส่วนที่ 4 ปุ่มสำหรับเพิ่มการแจ้งเตือนการทานยา



รูปที่ 3.14: หน้าแสดงการเพิ่มการแจ้งเตือนทานยา

จากภาพที่ 3.14 การออกแบบแบบหลักประกอบไปด้วย 5 ส่วนดังนี้

- ส่วนที่ 1 ลิสท์สำหรับเลือกประเภทการรับประทานมี 2 ประเภทได้แก่ สำหรับรับประทานและสำหรับฉีด
- ส่วนที่ 2 ช่องสำหรับกรอกชื่อยา ลักษณะของยา หรือสีของยา
- ส่วนที่ 3 ช่องสำหรับกรอกจำนวนครั้งในการทานยา
- ส่วนที่ 4 ลิสท์สำหรับกำหนดเวลาการแจ้งเตือนการทานยา
- ส่วนที่ 5 ปุ่มสำหรับยืนยันการเพิ่มการแจ้งเตือนทานยา



รูปที่ 3.15: หน้าจอแสดงข้อมูลส่วนตัวของผู้ใช้

จากภาพที่ 3.15 การออกแบบหลักประกอบไปด้วย 4 ส่วนดังนี้

- ส่วนที่ 1 รูปประจำตัวของผู้ใช้
- ส่วนที่ 2 ปุ่มสำหรับเปลี่ยนรูปประจำตัว
- ส่วนที่ 3 แสดงจำนวนโพสท์ เพื่อน ครอบครัว และปุ่มแก้ไขข้อมูลส่วนตัว
- ส่วนที่ 4 โพสท์ของผู้ใช้

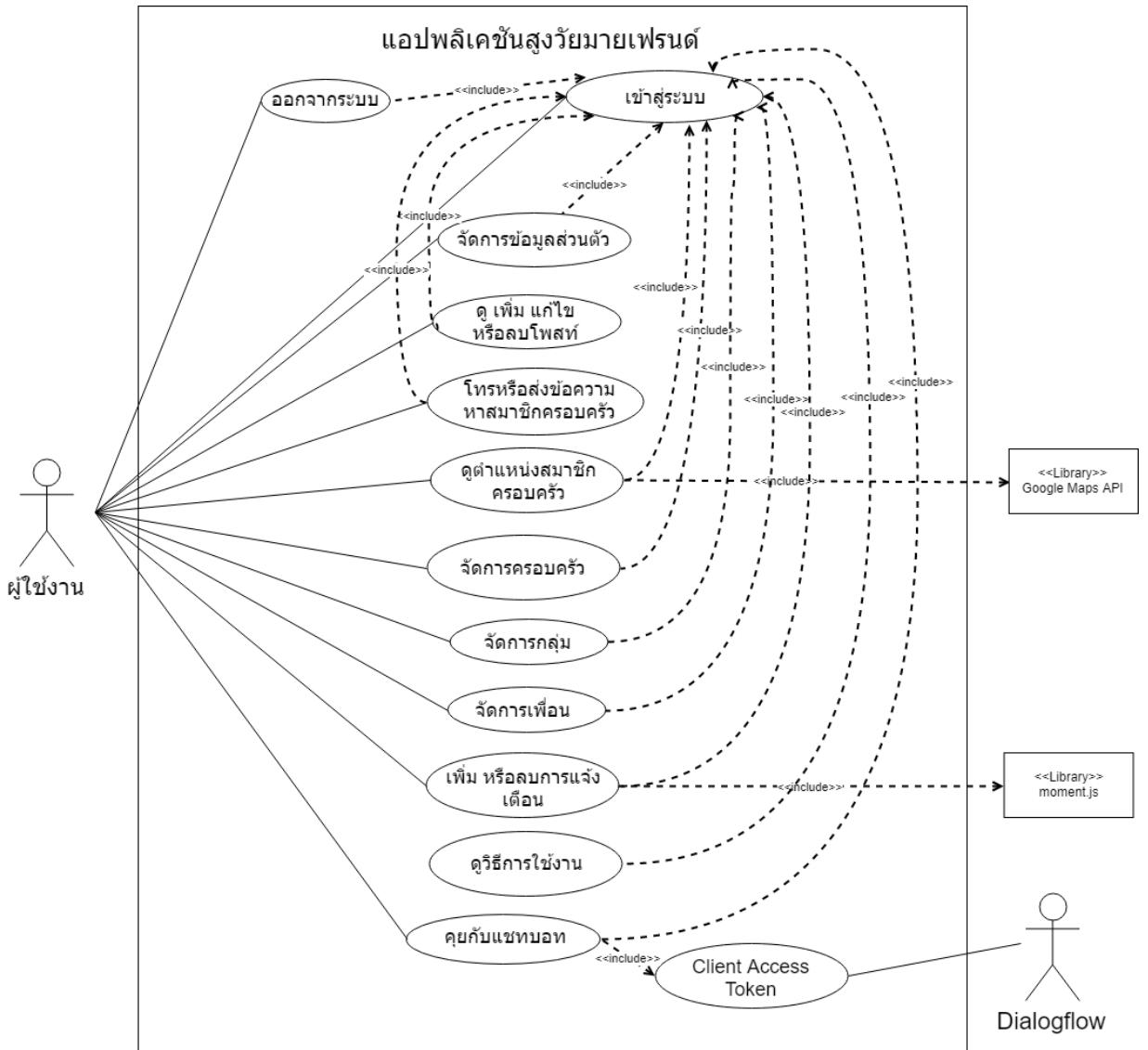
3.4 แผนภาพໄດ້ອະແກນ

3.4.1 แผนภาพຢູ່ສເຄສ (Use Case Diagram)

Use Case Diagram เป็นແຜນຜັງເພື່ອແສດງຟັງກ່ານແສດງກາರທຳມະນາຂອງຮບບໂດຍຮົມ ແສດງສ່ວນປະກອບໃນຮບບແລະ ກິຈກຣມທີ່ເກີດຂຶ້ນໃນຮບບຊື່ໃນຮບບຮບບກອງທຸນເງິນໃຫ້ກູ້ຍືມເພື່ອກາຮືກໜາ ຄະວິທາຍາສຕ່ຣ ມາຫວິທາລໍຍອຸບລາຮາຈຮານີ ຜູ້ໃຈຈະເປັນຕ້ອງເຂົ້າສູ່ຮບບເພື່ອໃຊ້ຈານຮບບ ສັນລັກໝົນທີ່ໃຊ້ໃນກາຮືກໜາ Use Case Diagram ແສດງໃນຕາຮາງທີ່ 3.1

ຕາຮາງທີ່ 3.1: ສັນລັກໝົນຂອງ Use case Diagram

ສັນລັກໝົນ	ການໃຊ້ຈານ
Use case	Use case ອີ່ສ່ວນຍ່ອຍຂອງຮບບຈານ ແກ່ນດ້ວຍງວງຮີແລະ ຊົ່ວໂອງ Use case ກາຍໃນງວງຮີ
	Actor ອີ່ບຸຄຄລ໌ທີ່ຮົບປະໂຫຍດຈາກຮບບຊື່ອຸ່ງກາຍນອກຮບບ ແກ່ນດ້ວຍຮູບປານແລະ ມີຊື່ອົບທຳການໃຊ້ຈານຮບບ
	ເສັ້ນຕຽງທີ່ແສດງຕິດການໃຊ້ຈານ Use case ຂອງຜູ້ກະທຳ
	ກຮອບ ສື່ເໜີ່ຍືມ ແສດງ ຕິ່ງ ຂອບເຂດ ຂອງ ຮະບບ ໂດຍ ແສດງ ຊົ່ວ ຮະບບ ກາຍໃນ ອີ່ດ້ານບນກຮອບ ສື່ເໜີ່ຍືມ Use case ອູ່ກາຍໃນກຮອບ ສື່ເໜີ່ຍືມ ແລະ actor ອູ່ກາຍນອກກຮອບ ສື່ເໜີ່ຍືມ
	ຄວາມສັ້ນພັນຮີແບບ «includes» ແສດງວ່າ Use case ບໍ່ໄດ້ດໍາເນີນການ ຕາມຂັ້ນຕອນຂອງ Use case ອີ່ນ ໂດຍແກ່ນດ້ວຍສັນລັກໝົນລູກຄຣເສັ້ນ ປະ ຊົ່ງ Use case ທີ່ໜ້າລູກຄຣເຮັດໃຊ້ຈານ Use case ທີ່ໜ້າລູກຄຣທຸກ ຄັ້ງທີ່ມີການທຳມະນາ
	ຄວາມສັ້ນພັນຮີແບບ «extend» ແສດງວ່າ Use case ບໍ່ໄດ້ດໍາເນີນການ ຕາມຂັ້ນຕອນຂອງ Use case ອີ່ນ ໂດຍແກ່ນດ້ວຍສັນລັກໝົນລູກຄຣເສັ້ນ ປະ ຊົ່ງ Use case ທີ່ໜ້າລູກຄຣເຮັດໃຊ້ຈານ Use case ທີ່ໜ້າລູກຄຣ ແຕ່ ການໃຊ້ຈານໄມ່ຈະເປັນຕ້ອງເກີດຂຶ້ນທຸກຄັ້ງຂຶ້ນອູ່ກັບເຈື່ອນໄຂຮ່ວ່າການ ທຳມະນາ



รูปที่ 3.16: Use Case Diagram ของแอปพลิเคชันสูงวัยมายเฟรนด์

ตารางที่ 3.2: อธิบาย Use Case หน้าที่ของแอปพลิเคชันสูงวัยมายเฟรนด์ในภาพที่ 3.16

Use Case	คำอธิบาย
เข้าสู่ระบบ	ใช้สำหรับให้ผู้ใช้เข้าใช้งานแอปพลิเคชัน
ออกจากระบบ	ผู้ใช้ที่เข้าสู่ระบบแล้วสามารถออกจากระบบ
จัดการข้อมูลส่วนตัว	ใช้งานเพื่อให้ผู้ใช้ แก้ไขข้อมูลที่ต้องการเปลี่ยนแปลง
ดู เพิ่ม แก้ไข หรือลบโพสท์	ใช้งานเพื่อให้ผู้ใช้สามารถดู เพิ่ม แก้ไข หรือลบโพสท์ของตัวเอง
โทรหีอส่งข้อความหาสมาชิกในครอบครัว	ใช้สำหรับให้ผู้ใช้สามารถโทรหีอส่งข้อความหาคนในครอบครัว และสามารถเปิดเสียงฉุกเฉิน
ดูตำแหน่งสมาชิกในครอบครัว	ใช้งานเพื่อให้ผู้ใช้สามารถดูตำแหน่งปัจจุบันของสมาชิกในครอบครัวด้วยแผนที่
จัดการครอบครัว	ใช้สำหรับเพิ่ม แก้ไข หรือลบข้อมูลสมาชิกครอบครัว
จัดการกลุ่ม	ใช้สำหรับเพิ่ม แซท แก้ไข หรือลบกลุ่มได้ และเพิ่ม แก้ไข หรือลบสมาชิกในกลุ่มได้
จัดการเพื่อน	ใช้สำหรับยืนยันคำร้องขอเป็นเพื่อน รวมถึงพูดคุย เพิ่มและลบเพื่อน
เพิ่ม หรือลบการแจ้งเตือน	ใช้สำหรับเพิ่ม หรือลบการแจ้งเตือนการทำงานยา
ดูวิธีการใช้งาน	ใช้สำหรับให้ผู้ใช้ดูรายละเอียดของหน้าต่าง ๆ ในแอปพลิเคชัน
คุยกับเซทบทอท	ใช้งานเพื่อให้ผู้ใช้สามารถถามเรื่องโรคหรือเรื่องที่สนใจ

ตารางที่ 3.3: Use Case เข้าสู่ระบบ

Use Case Title : เข้าสู่ระบบ	Use case Id : 1
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชัน	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถเข้าสู่ระบบได้	

ตารางที่ 3.4: Use Case ออกรายงาน

Use Case Title : ออกรายงาน	Use case Id : 2
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถออกจากรายงานได้	
Exceptional Flow ที่ 1 : หากยังไม่เข้าสู่ระบบจะไม่สามารถออกจากรายงานได้	

ตารางที่ 3.5: Use Case จัดการข้อมูลส่วนตัว

Use Case Title : จัดการข้อมูลส่วนตัว	Use case Id : 3
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถดู เพิ่ม แก้ไขข้อมูลส่วนตัวได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถดู เพิ่ม แก้ไขข้อมูลส่วนตัวได้	

ตารางที่ 3.6: Use Case ดู เพิ่ม แก้ไข หรือลบโพสท์

Use Case Title : ดู เพิ่ม แก้ไข หรือลบโพสท์	Use case Id : 4
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถดู เพิ่ม แก้ไข หรือลบโพสท์ได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถดู เพิ่ม แก้ไข หรือลบโพสท์	

ตารางที่ 3.7: Use Case โทรหรือส่งข้อความหาสมาชิกครอบครัว

Use Case Title : โทรหรือส่งข้อความหาสมาชิกครอบครัว	Use case Id : 5
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถโทรศัพท์หรือส่งข้อความหาสมาชิกครอบครัว และสามารถเปิดเสียงฉุกเฉินได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถโทรศัพท์หรือส่งข้อความหาสมาชิกครอบครัว และสามารถเปิดเสียงฉุกเฉินได้	

ตารางที่ 3.8: Use Case ดูตำแหน่งสมาชิกครอบครัว

Use Case Title : ดูตำแหน่งสมาชิกครอบครัว	Use case Id : 6
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถดูตำแหน่งสมาชิกในครอบครัวได้ด้วยแผนที่	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูตำแหน่งสมาชิกในครอบครัวได้	

ตารางที่ 3.9: Use Case จัดการครอบครัว

Use Case Title : จัดการครอบครัว	Use case Id : 7
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถเพิ่ม แก้ไข ลบสมาชิกในครอบครัวได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถเพิ่ม แก้ไข ลบสมาชิกในครอบครัวได้	
Exceptional Flow ที่ 2 : หากผู้ใช้ไม่มีผู้ใช้งานเป็นเพื่อน จะไม่สามารถเพิ่มเข้ามาในครอบครัวได้	

ตารางที่ 3.10: Use Case จัดการกลุ่ม

Use Case Title : จัดการกลุ่ม	Use case Id : 8
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถเพิ่ม พูดคุย แก้ไข หรือลบกลุ่ม เพิ่ม แก้ไข หรือลบสมาชิกในกลุ่มได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถเพิ่ม พูดคุย แก้ไข หรือลบกลุ่ม เพิ่ม แก้ไข หรือลบสมาชิกในกลุ่มได้	
Exceptional Flow ที่ 2 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ตจะไม่สามารถเพิ่ม พูดคุย แก้ไข หรือลบกลุ่ม เพิ่ม แก้ไข หรือลบสมาชิกในกลุ่มได้	

ตารางที่ 3.11: Use Case จัดการเพื่อน

Use Case Title : จัดการเพื่อน	Use case Id : 9
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถเพิ่ม พูดคุย แก้ไข หรือลบเพื่อนได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถเพิ่ม พูดคุย แก้ไข หรือลบเพื่อนได้	
Exceptional Flow ที่ 2 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ตจะไม่สามารถเพิ่ม พูดคุย แก้ไข หรือลบเพื่อนได้	

ตารางที่ 3.12: Use Case เพิ่ม หรือลบการแจ้งเตือน

Use Case Title : เพิ่ม หรือลบการแจ้งเตือน	Use case Id : 10
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถเพิ่ม หรือลบการแจ้งเตือนการทานยาได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถเพิ่ม หรือลบการแจ้งเตือนได้	

ตารางที่ 3.13: Use Case ดูวิธีการใช้งาน

Use Case Title : ดูวิธีการใช้งาน	Use case Id : 11
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถดูรายละเอียดของหน้าจอต่าง ๆ ได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้าสู่ระบบจะไม่สามารถดูรายละเอียดของหน้าจอต่าง ๆ ได้	

ตารางที่ 3.14: Use Case คุยกับzechboph

Use Case Title : คุยกับzechboph	Use case Id : 12
Primary Actor : ผู้ใช้งาน	
Stakeholder Actor : -	
Main Flow : สามารถคุยโต้ตอบกับzechbophได้	
Exceptional Flow ที่ 1 : หากผู้ใช้งานไม่เชื่อมต่ออินเทอร์เน็ตจะไม่คุยกับzechbophได้	
Exceptional Flow ที่ 2 : หากผู้ใช้งานไม่เข้าสู่ระบบจะไม่คุยกับzechbophได้	

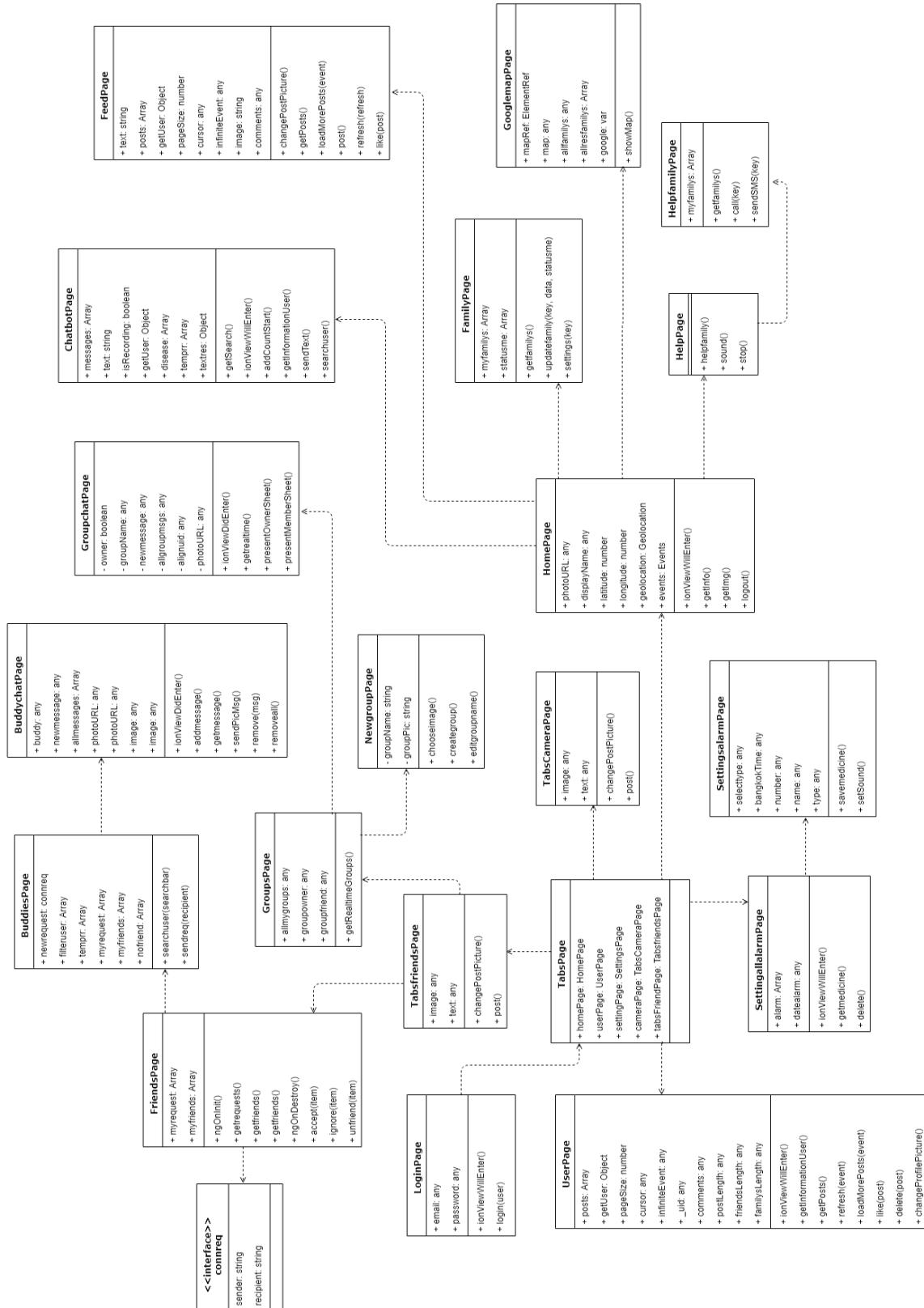
3.4.2 แผนภาพคลาส (Class Diagram)

Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแบบต่างๆ ระหว่างคลาส สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่ 3.15

ตารางที่ 3.15: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	การใช้งาน			
<table border="1"> <tr> <td>Class Name</td> </tr> <tr> <td>Attribute Name</td> </tr> <tr> <td>Operation Name()</td> </tr> </table>	Class Name	Attribute Name	Operation Name()	<p>คลาส สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบน เป็นชื่อของ class ส่วนกลาง เป็นชื่อ Attribute และส่วนล่าง เป็น Operation Name หรือ Method ใช้สำหรับเขียนฟังก์ชันในการทำงานของคลาสนั้น ๆ ชนิดของ Visibility ของ Method และ Attribute แบ่งเป็น 3 ชนิด ได้แก่</p> <ul style="list-style-type: none"> (a) Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+) (b) Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-) (c) Protected แทนสัญลักษณ์ด้วยเครื่องหมายชาร์ป ()
Class Name				
Attribute Name				
Operation Name()				
	Dependency Relationship หมายความว่า คลาสที่อยู่ผู้ฝึกศร สามารถเรียกใช้คลาสที่อยู่ผู้หัวลูกศร			
	Composition Relationship เป็นความสัมพันธ์ระหว่างขอบเจ็กต์ หรือคลาสแบบขึ้นต่อกันและมีความเกี่ยวข้องกันเสมอ			
	Realization Relationship เป็นความสัมพันธ์ระหว่าง Object หรือ Class ในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Subclass)			
	Connector เป็นสัญลักษณ์แทนด้วยรูปห้าเหลี่ยมและมีชื่ออู่ตระ กลาง จะสร้างสัญลักษณ์นี้ไว้เมื่อต้องการเชื่อมต่อคลาสที่อยู่คนละ หน้า			

Class Diagram แสดงความสัมพันธ์ในรูปแบบต่างๆ ระหว่างคลาสของแอปพลิเคชันสูงวัย
นายเฟรนด์ อธิบายได้ตามภาพที่ ?? ดังต่อไปนี้



รูปที่ 3.17: Class Diagram ของแอปพลิเคชันสังคมฯภาษาเพอร์ล

จากรูปภาพที่ 3.17 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.16: อธิบาย Class Diagram ของแอปพลิเคชันสูงวัยมายเฟรนด์

Class Diagram	คำอธิบาย
LoginPage	คลาส LoginPage เป็นคลาสที่ใช้เพื่อให้ผู้ใช้ที่ได้ลงทะเบียนกับระบบไว้แล้วเข้าระบบเพื่อใช้งานแอปพลิเคชัน
TabsPage	คลาส TabsPage เป็นคลาสที่ถูกเรียกเพื่อจัดการ Tabs
TabsfriendsPage	คลาส TabsfriendsPage เป็นคลาสที่ใช้แสดงสมาชิกกลุ่ม คำร้องขอเพิ่มเพื่อนและเพื่อน
FriendsPage	คลาส FriendsPage เป็นคลาสที่ใช้แสดงคำร้องขอเพื่อนและเพื่อน
connreq	คลาส connreq เป็นคลาสที่ใช้เก็บข้อมูลคำร้องขอเพื่อน
BuddiesPage	คลาส BuddiesPage เป็นคลาสที่ใช้สำหรับการเพิ่มเพื่อน
BuddychatPage	คลาส BuddychatPage เป็นคลาสที่ใช้สำหรับแชทกับเพื่อน
BuddychatPage	คลาส BuddychatPage เป็นคลาสที่ใช้สำหรับแชทกับเพื่อน
GroupsPage	คลาส GroupsPage เป็นคลาสที่แสดงกลุ่มของฉันและกลุ่มที่เป็นสมาชิก
NewgroupPage	คลาส NewgroupPage เป็นคลาสที่ใช้สร้างกลุ่ม
GroupchatPage	คลาส GroupchatPage เป็นคลาสสำหรับการแชทกลุ่ม
SettingallalarmPage	คลาส SettingallalarmPage เป็นคลาสที่แสดงการแจ้งเตือนการทانยาทั้งหมด
SettingsalarmPage	คลาส SettingsalarmPage เป็นคลาสที่ใช้บันทึกการแจ้งเตือนการทันยา
TabsCameraPage	คลาส TabsCameraPage เป็นคลาสที่ใช้สำหรับสร้างโพสท์
HomePage	คลาส HomePage เป็นคลาสสำหรับการแสดงหน้าเมนูหลักของแอปพลิเคชัน
ChatbotPage	คลาส ChatbotPage เป็นคลาสที่ใช้สำหรับพูดคุยกับแชทบอท
FeedPage	คลาส FeedPage เป็นคลาสที่แสดงโพสท์ทั้งหมด
FamilyPage	คลาส FamilyPage เป็นคลาสที่แสดงสมาชิกในครอบครัว
GooglemapPage	คลาส GooglemapPage เป็นคลาสที่แสดงตำแหน่งของสมาชิกในครอบครัวด้วย Google Map

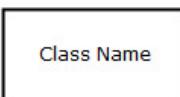
ตารางที่ 3.17: อธิบาย Class Diagram ของแอปพลิเคชันสูงวัยมายเฟรนด์

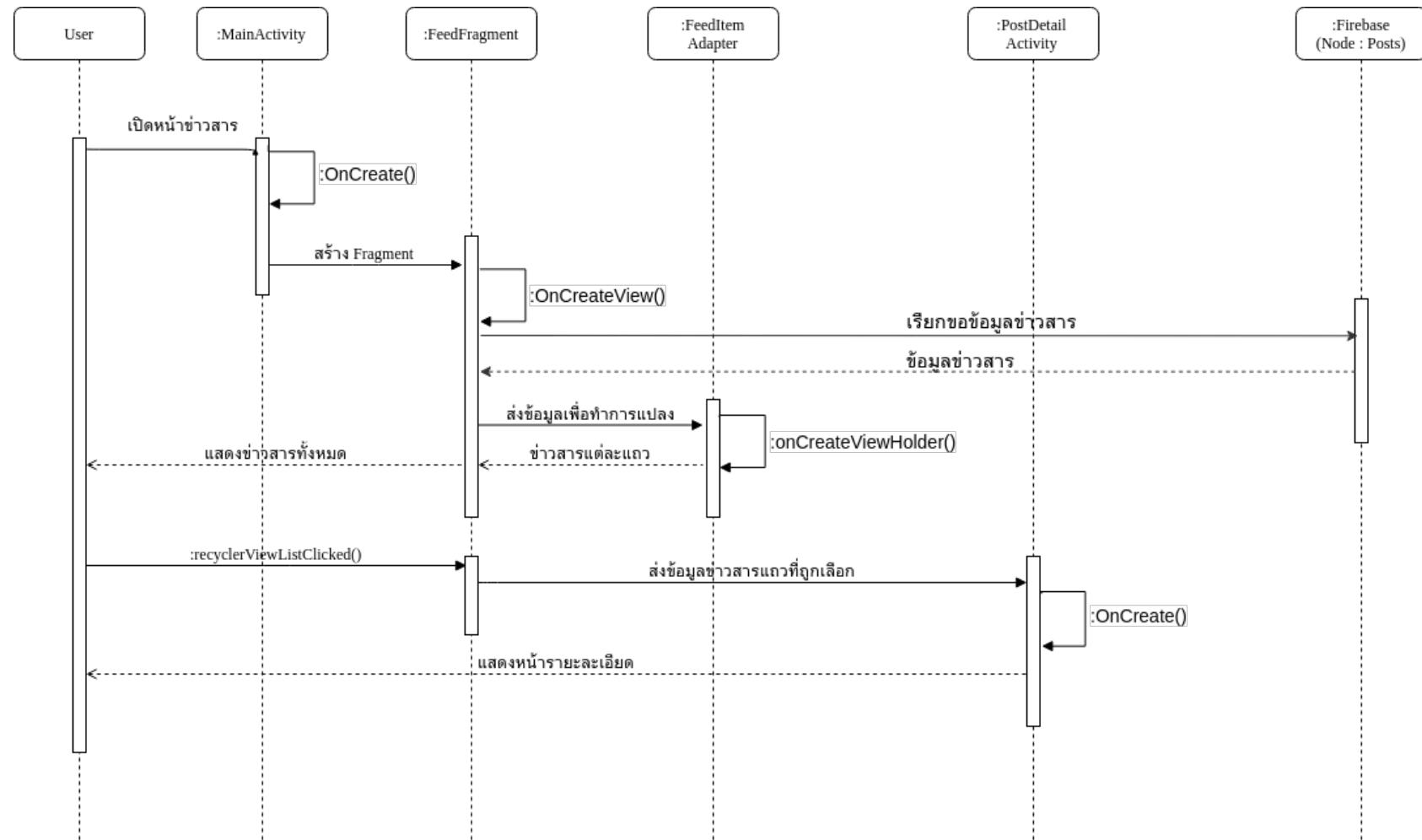
Class Diagram	คำอธิบาย
FamilyPage	คลาส FamilyPage เป็นคลาสที่แสดงสมาชิกในครอบครัว
GooglemapPage	คลาส GooglemapPage เป็นคลาสที่แสดงตำแหน่งของสมาชิกในครอบครัวด้วย Google Map
HelpPage	คลาส HelpPage เป็นคลาสที่แสดงรูปภาพขอความช่วยเหลือ 2 แบบ คือ แบบติดต่อครอบครัว และแบบส่งเสียงฉุกเฉิน
HelpfamilyPage	คลาส HelpfamilyPage เป็นคลาสแสดงสมาชิกในครอบครัวเพื่อขอความช่วยเหลือ มี 2 แบบได้แก่ โทร และส่งข้อความ

3.5 ชีเควนไ/doeagram (Sequence Diagram)

Sequence Diagram เป็น Diagram ที่แสดงขั้นตอนการทำงานของแต่ละ Use Case ระหว่าง Object ต่างๆ ที่ส่งข้อความถึงกันและกัน โดย Sequence Diagram จะช่วยให้มองเห็นการทำงานของภาพรวมของระบบ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียน Sequence Diagram แสดงดังตารางที่ 3.17

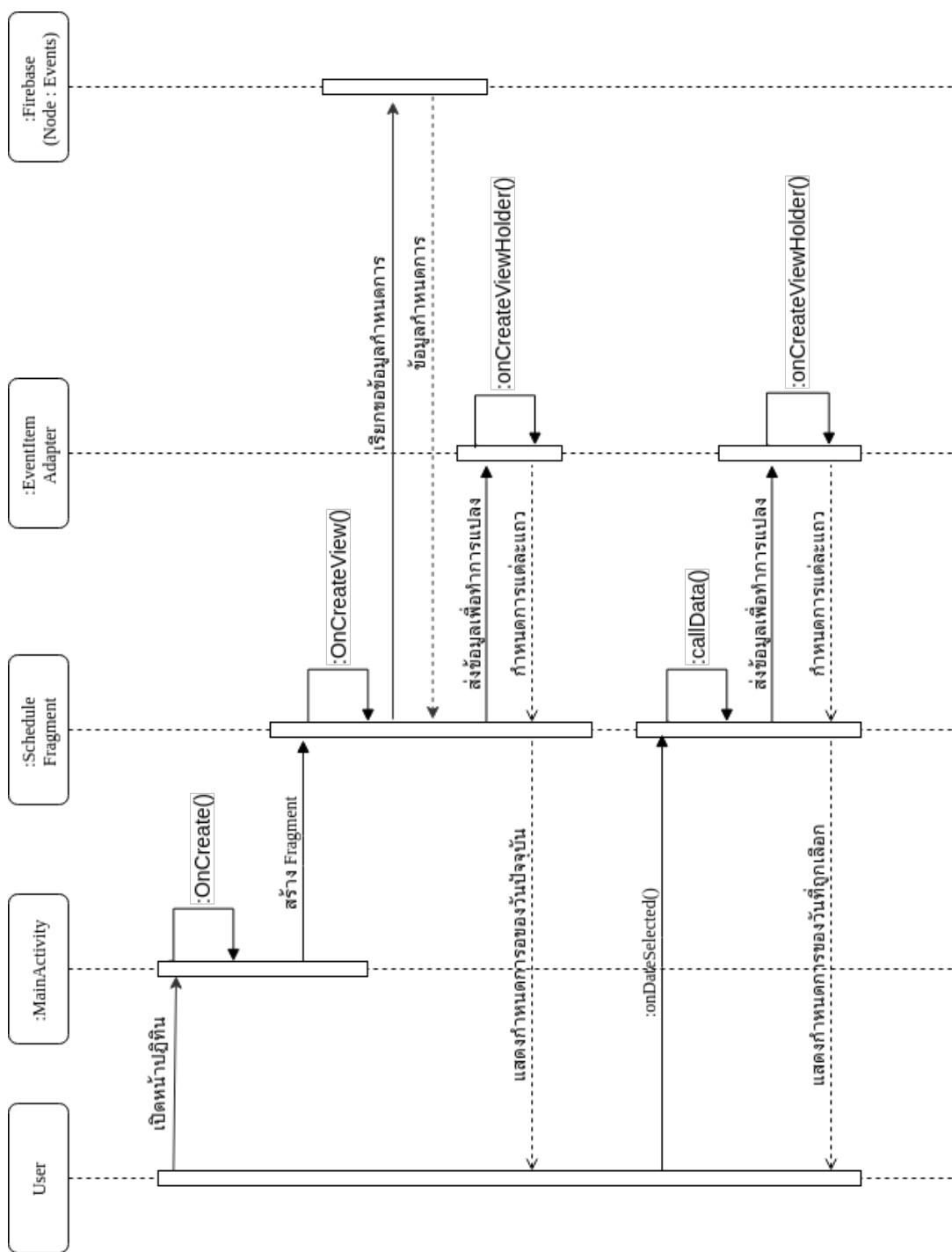
ตารางที่ 3.18: สัญลักษณ์ของ Sequence Diagram

สัญลักษณ์	การใช้งาน
	Class แสดงถึงการทำงานของ Use Case ใน การส่งหรือรับข้อความ แทนด้วยสัญลักษณ์สี่เหลี่ยมมีชื่อคลาสอยู่ภายใน
	Lifeline หรือเส้นอายุขัย แสดงช่วงเวลาตั้งแต่เริ่มสร้าง object ในคลาสนั้น จนกระทั่ง object นั้นถูกทำลาย สัญลักษณ์แทนด้วยเส้นประ
	Focus of control หรือจุดควบคุม เป็นจุดควบคุมที่ object ใช้ทำการส่งหรือรับข้อความ สัญลักษณ์แทนด้วยสี่เหลี่ยม
	Message คือ ข้อความที่รับส่งระหว่าง Object สัญลักษณ์แทนด้วยลูกศรและประกอบด้วย 2 ส่วน คือ ข้อมูล (Data) และฟังก์ชัน (Function)
	Return Message เป็นข้อมูลที่ส่งกลับหลังจากทำงานเสร็จ
	Self call เป็นการเรียกฟังก์กการทำงานภายในตัวเอง
	สร้างกรอบการทำงานของโปรแกรม เพื่อให้รู้ขอบเขตของการทำงาน เช่น ลูป(loop)



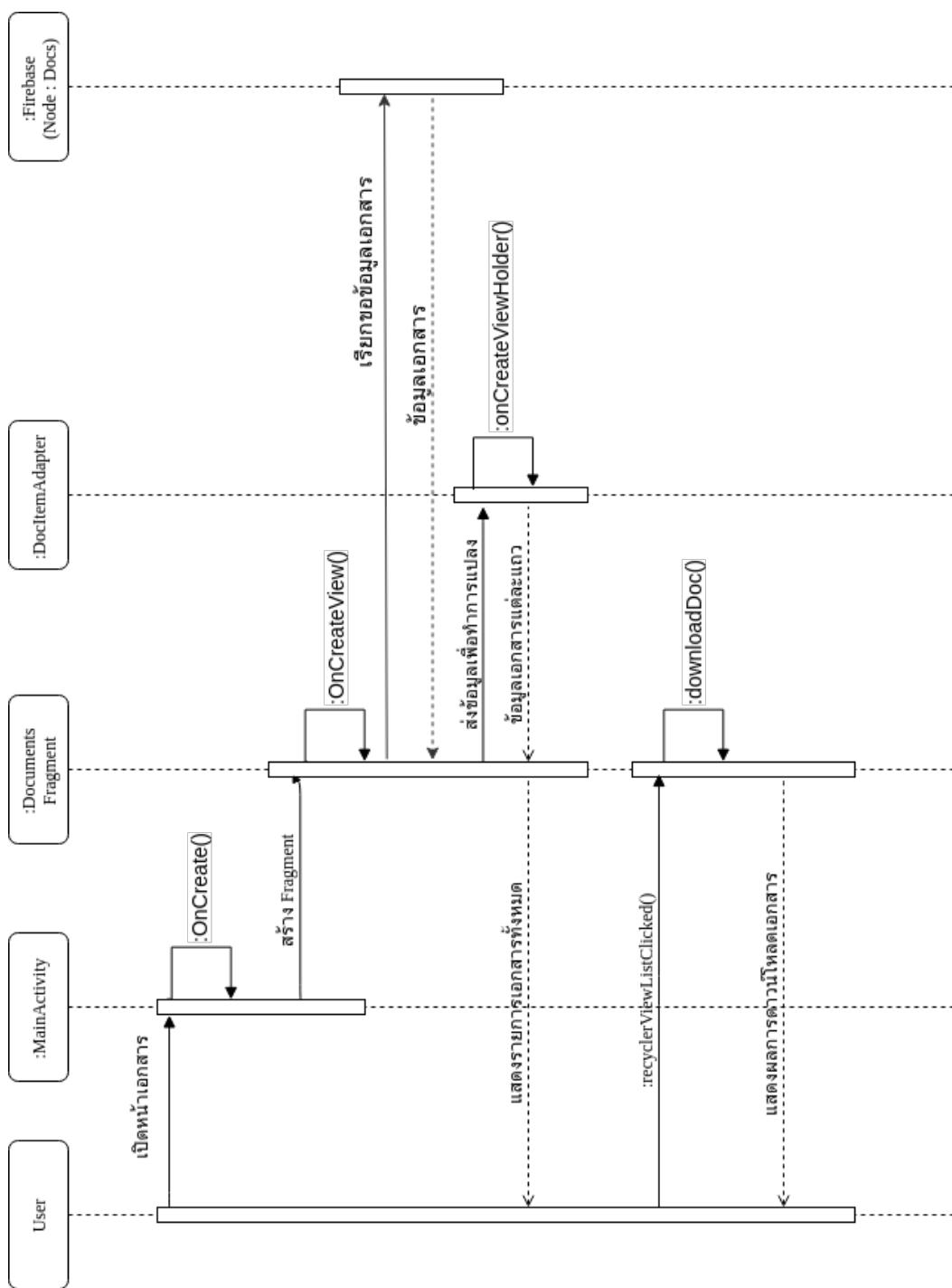
รูปที่ 3.18: Sequence Diagram การแสดงข่าวสาร

จากภาพที่ 3.18 สามารถอธิบายแผนภาพ Sequence Diagram แสดงข่าวสาร ได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส FeedFragment เมื่อ FeedFragment ถูกติดตั้งบน MainActivity เมธอด callData() จะสืบค้นข้อมูลจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส FeedItemAdapter โดยมีการคืนค่าเป็นข้อมูลข่าวสารแต่ละแถวและในขั้นตอนสุดท้ายคลาส FeedFragment จะทำการแสดงรายการข้อมูลข่าวสารทั้งหมดออกทางหน้าจอ หากผู้ใช้มีการกดเลือกข่าวสารบางแถวคลาส FeedFragment จะทำการเรียกใช้ PostDetailActivity เพื่อแสดงรายละเอียดข้อมูลข่าวสารของแถวที่ถูกเลือก



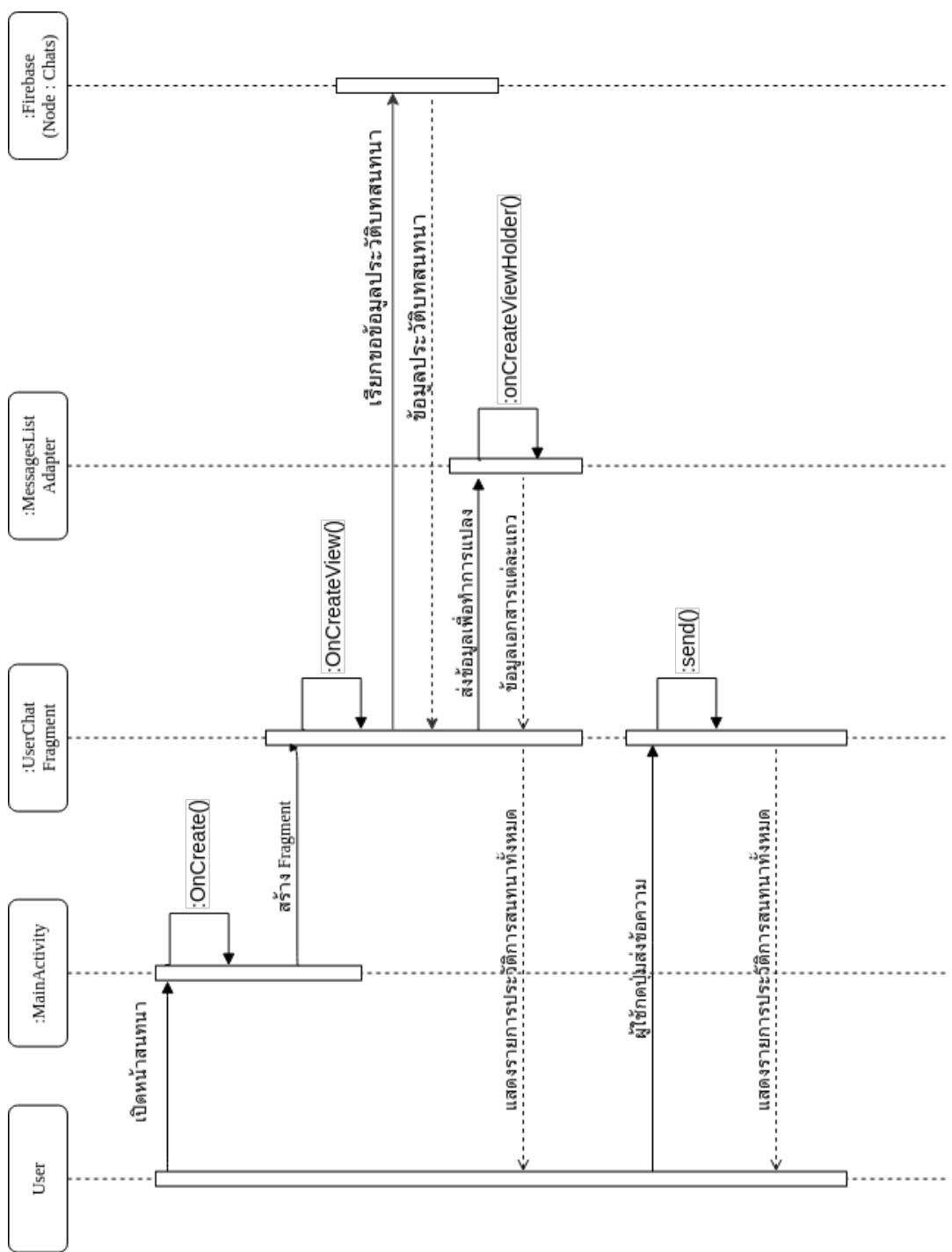
รูปที่ 3.19: Sequence Diagram การแสดงแบบปฏิทินก้าวหน้าของการ

จากภาพที่ 3.19 สามารถอธิบายแผนภาพ Sequence Diagram แสดงปฏิทินกำหนดการได้ดังนี้ เมื่อ ผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส ScheduleFragment เมื่อ ScheduleFragment ถูกติดตั้งบน MainActivity เมธอด callData() จะสืบค้นข้อมูลกำหนดการของวันปัจจุบันจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส Schedule-ItemAdapter โดยมีการคืนค่าเป็นข้อมูลกำหนดการแต่ละแถว และในขั้นตอนสุดท้ายคลาส Schedule-Fragment จะทำการแสดงรายการกำหนดการวันปัจจุบันออกทางหน้าจอ หากผู้ใช้มีการกดเลือกวันที่ที่ต้องการทราบกำหนดการจากปฏิทิน คลาส ScheduleFragment จะทำการเรียกใช้ callData() อีกครั้งโดยสืบค้นข้อมูลกำหนดการของวันที่ถูกเลือกจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส ScheduleItemAdapter โดยมีการคืนค่าเป็นข้อมูลแต่กำหนดการละแถวและในขั้นตอนสุดท้ายคลาส ScheduleFragment จะทำการแสดงรายการกำหนดการวันที่ผู้ใช้เลือกออกทางหน้าจอ



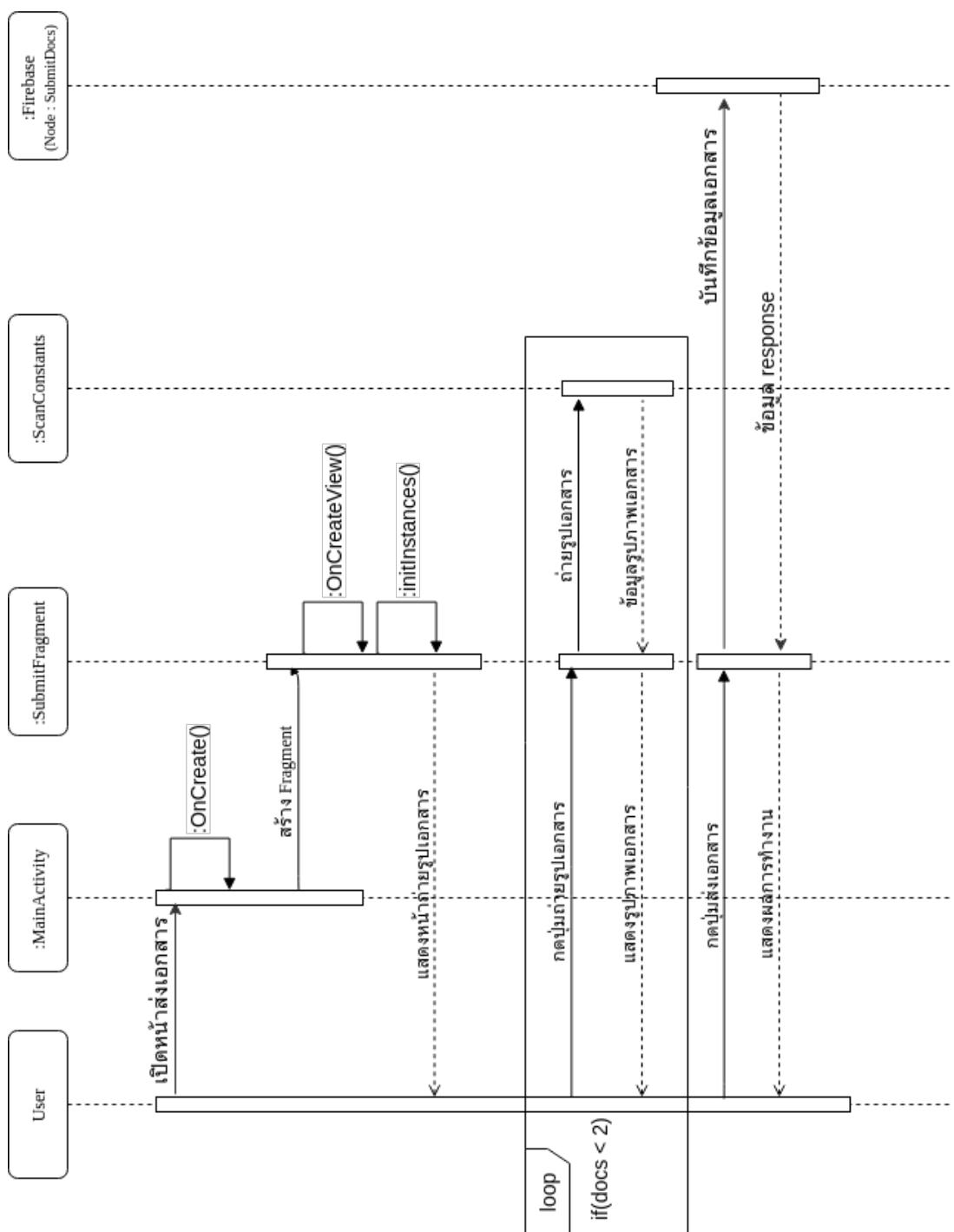
รูปที่ 3.20: Sequence Diagram การแสดงตามไฟล์เอกสาร

จากภาพที่ 3.20 สามารถอธิบายแผนภาพ Sequence Diagram แสดงดาวน์โหลดเอกสารได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส DocumentsFragment เมื่อ DocumentsFragment ถูกติดตั้งบน MainActivity เมธอด initInstances() จะสืบค้นข้อมูลเอกสารทั้งหมดจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส DocItem-Adapter โดยมีการคืนค่าเป็นข้อมูลเอกสารแต่ละແควและในขั้นตอนสุดท้ายคลาส Documents-Fragment จะทำการแสดงรายการกำหนดการวันปัจจุบันออกทางหน้าจอ



รูปที่ 3.21: Sequence Diagram การแสดงบทสนทนา

จากภาพที่ 3.21 สามารถอธิบายแผนภาพ Sequence Diagram แสดงการสนทนา ได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบ จะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส UserChatFragment เมื่อ UserChatFrag-ment ถูกติดตั้งบน MainActivity เมธอด getMessage() จะสืบค้น ข้อมูลประวัติการสนทนาของผู้ใช้คนปัจจุบันทั้งหมดจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส MessagesListAdapter โดยมีการคืนค่าเป็นข้อมูลรายการประวัติการสนทนาทั้งหมดและในขั้นตอนสุดท้ายคลาส User-ChatFragment จะทำการแสดงรายการประวัติการสนทนาทั้งหมดออกจากหน้าจอ เมื่อผู้ใช้พิมพ์ข้อความแลกดปุ่มส่งระบบจะเรียกให้เมธอด send() เพื่อทำการบันทึกข้อมูลไว้บน Firebase FireStore และทำการแสดงข้อมูลรายการประวัติการสนทนาทั้งหมดที่ถูกอัพเดท



รูปที่ 3.22: Sequence Diagram แสดงถึง logic ของการตรวจสอบ

จากภาพที่ 3.22 สามารถอธิบายแผนภาพ Sequence Diagram และสิ่งที่เอกสารตรวจสอบได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส SubmitFragment เมื่อ Submit-Fragment ถูกติดตั้งบน MainActivity เมธอด initInstances() จะถูกเรียก เพื่อสร้างหน้าจอแสดงผลเมื่อผู้ใช้กดปุ่มถ่ายรูประบบจะเรียกใช้ไลบรารี ScanConstants เพื่อถ่ายภาพเอกสารและรอให้ผู้ใช้ถ่ายครบทั้งสองแผ่นจึงจะแสดงปุ่มกดส่งเอกสารเพื่อตรวจสอบ

3.6 โครงสร้างฐานข้อมูลไฟร์เบส(Firebase Database Structure)

Firebase Database นั้นเป็น Database แบบ NoSQL และเป็น JSON database ที่มีโครงสร้างที่เป็น Key และ Value จัดเก็บข้อมูลในลักษณะโหนด หากต้องการเรียกใช้โดย การท่องไปยังโหนดที่ต้องการ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียนโครงสร้างฐานข้อมูลแบบ Firebase แสดงดังตารางที่ 3.18

ตารางที่ 3.19: สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Firebase

สัญลักษณ์	คำอธิบาย
	Database เป็นการเรียกชื่อแทนโหนด(Node)บนสุดที่ใช้ในการเก็บข้อมูล
	Collection เป็นการเรียกชื่อแทนของการเก็บหลาย ๆ เอกสารไว้ด้วยกัน
	Document เป็นการเรียกชื่อแทน หน่วย การเก็บ ของ ข้อมูล ใน Cloud Firestore ภายในจะประกอบไปด้วย ชื่อของ Document ซึ่งของคีย์ (key) และ ค่าข้อมูล (value) โดยชื่อของ Document ห้ามซ้ำกัน ซึ่งใน Cloud Firestore สามารถระบุประเภทของข้อมูลได้ 9 ประเภทได้แก่ boolean, number, string, geo point, timestamp, array, object, reference และ null

รูปที่ 3.23: โครงสร้างฐานข้อมูลแบบ Firebase

รูปที่ 3.24: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

รูปที่ 3.25: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

รูปที่ 3.26: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

จากรูปที่ 3.23-3.34 สามารถอธิบายโครงสร้างของข้อมูลได้ดังนี้

รูปที่ 3.27: โหนดเก็บข้อมูลประกาศ

ตารางที่ 3.20: อธิบายโหนดที่ใช้เก็บข้อมูลประกาศ

Key	คำอธิบาย
Posts	โหนดสำหรับเก็บข้อมูลประกาศทั้งหมด
Post	สำหรับเก็บข้อมูลแต่ละประกาศ
title	สำหรับเก็บชื่อหัวข้อประกาศ
description	สำหรับเก็บรายละเอียดประกาศ
collection	สำหรับเก็บประเภทของประกาศได้แก่ สารานุกรมและเฉพาะบุคคล
fileURL	สำหรับเก็บ url ของเอกสารแนบประกาศ
id	สำหรับเก็บรหัสของประกาศ
time	สำหรับเก็บเวลาที่ประกาศ

รูปที่ 3.28: โหนดเก็บข้อมูลเอกสารที่เกี่ยวข้อง

ตารางที่ 3.21: อธิบายโหนดที่ใช้เก็บข้อมูลเอกสารที่เกี่ยวข้อง

Key	คำอธิบาย
Docs	โหนดสำหรับเก็บข้อมูลของเอกสารที่เกี่ยวข้องทั้งหมด
Doc	สำหรับเก็บข้อมูลเอกสารแต่ละฉบับ
title	สำหรับเก็บชื่อหัวเรื่องของเอกสาร
description	สำหรับเก็บรายละเอียดของเอกสาร
fileType	สำหรับนามสกุลไฟล์เอกสาร เช่น .pdf .png เป็นต้น
fileURL	สำหรับเก็บ url ของเอกสาร
time	สำหรับเก็บเวลาที่ถูกอัพโหลดเข้าสู่ระบบโดยเจ้าหน้าที่

รูปที่ 3.29: โหนดเก็บข้อมูลประวัติการสนทนา

ตารางที่ 3.22: อธิบายโหนดที่ใช้เก็บข้อมูลประวัติการสนทนา

Key	คำอธิบาย
Chats	โหนดสำหรับเก็บข้อมูลประวัติการสนทนาทั้งหมด
User_id	สำหรับเก็บประวัติการสนทนาของผู้ใช้แต่ละคน
Messages	สำหรับเก็บประวัติการสนทนาทั้งหมดของผู้ใช้
Message	สำหรับเก็บข้อมูลของแต่ละข้อความ
message	สำหรับเก็บข้อความ
name	สำหรับเก็บชื่อของผู้ส่งข้อความ
photo	สำหรับเก็บ url รูปภาพของผู้ส่งข้อความ
senderId	สำหรับเก็บรหัสของผู้ส่งข้อความ
time	สำหรับเก็บเวลาที่ข้อความถูกส่ง

รูปที่ 3.30: โนนดเก็บข้อมูลกำหนดการ

ตารางที่ 3.23: อธิบายโนนดที่ใช้เก็บข้อมูลกำหนดการ

Key	คำอธิบาย
Events	โนนดสำหรับเก็บข้อมูลของกำหนดการทั้งหมด
Event	สำหรับเก็บข้อมูลของแต่ละกำหนดการ
title	สำหรับเก็บชื่อหัวข้อของกำหนดการ
description	สำหรับเก็บรายละเอียดของกำหนดการ
time	สำหรับเก็บเวลาของกำหนดการ

รูปที่ 3.31: โหนดเก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา

ตารางที่ 3.24: อธิบายโหนดที่ใช้เก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา

Key	คำอธิบาย
RusetSubmitDocs	โหนดสำหรับเก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษาทั้งหมด
User_id	สำหรับเก็บข้อมูลของแต่ละสำเนาเอกสารของนักศึกษาแต่ละคน
doc2	สำหรับเก็บ url ของภาพถ่ายสำเนาเอกสารฉบับที่ 1
doc2	สำหรับเก็บ url ของภาพถ่ายสำเนาเอกสารฉบับที่ 2
status	สำหรับเก็บผลการตรวจสอบของเจ้าหน้าที่
time	สำหรับเก็บเวลาที่สำเนาเอกสารถูกเพิ่มเข้าสู่ระบบ

รูปที่ 3.32: โหนดเก็บข้อมูลของนักศึกษา

ตารางที่ 3.25: อธิบายโหนดที่ใช้เก็บข้อมูลของนักศึกษา

Key	คำอธิบาย
Users	โหนดสำหรับเก็บข้อมูลของนักศึกษา
User_id	สำหรับเก็บข้อมูลของนักศึกษาแต่ละคน
depart	สำหรับเก็บภาควิชาของนักศึกษา
major	สำหรับเก็บสาขาวิชาของนักศึกษา
sid	สำหรับเก็บรหัสประจำตัวนักศึกษา
name	สำหรับเก็บชื่อของนักศึกษา
year	สำหรับเก็บชั้นปีของนักศึกษา
lastChat	สำหรับเก็บเวลาที่สนทนากับเจ้าหน้าที่ล่าสุด
photoUrl	สำหรับเก็บ url รูปภาพโปรไฟล์ (Profile)

รูปที่ 3.33: โหนดเก็บข้อมูลการจองคิวของนักศึกษา

ตารางที่ 3.26: อธิบายโหนดที่ใช้เก็บข้อมูลการจองคิวของนักศึกษา

Key	คำอธิบาย
Queue	โหนดสำหรับเก็บข้อมูลการจองคิวของนักศึกษาทั้งหมด
q_id	สำหรับเก็บข้อมูลของการจองคิวแต่ละครั้งที่เปิดจองคิว
Date	สำหรับเก็บวันที่สำหรับส่งเอกสาร
Time	สำหรับเก็บรายชื่อของนักศึกษาที่ทำการจองคิวในส่งเอกสารเวลา นั้น ๆ
User_id	สำหรับเก็บรหัสของนักศึกษา
title	สำหรับเก็บชื่อหัวเรื่องกำหนดการการจองคิว
studentPerHr	สำหรับเก็บจำนวนนักศึกษาต่อชั่วโมง

รูปที่ 3.34: โหนดเก็บข้อมูลคำถามที่พับบออย

ตารางที่ 3.27: อธิบายโหนดที่ใช้เก็บข้อมูลคำถามที่พับบออย

Key	คำอธิบาย
Queue	โหนดสำหรับเก็บข้อมูลคำถามที่พับบออยทั้งหมด
Faq_id	สำหรับเก็บข้อมูลคำถามที่พับบออยแต่ละรายการ
title	สำหรับเก็บคำถาม
description	สำหรับเก็บคำตอบ

บทที่ 4

การพัฒนาระบบ

หลังจากที่ได้มีการเตรียมความพร้อมสำหรับการพัฒนาในด้านต่าง ไม่ว่าจะเป็นที่มาและความสำคัญของปัญหา เทคโนโลยีที่มีความเหมาะสมกับระบบ และการออกแบบระบบ การทำงานรวมไปถึงโครงสร้างของข้อมูล ในบทนี้จะเป็นการพูดถึงการสร้างระบบที่ได้มีการออกแบบไว้ในบทที่แล้วจะถูกนำเสนอในบทนี้ โดยการพัฒนาระบบแบ่งได้เป็นส่วนต่าง ๆ ดังนี้

- 4.1 การพัฒนาเว็บแอปพลิเคชัน
- 4.2 การพัฒนาเว็บแอปพลิเคชันแอนดรอยด์แอปพลิเคชัน

4.1 การพัฒนาเว็บแอปพลิเคชัน

การพัฒนาระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษาสำหรับเว็บแอปพลิเคชันนั้นวัตถุประสงค์ หลังเพื่อสร้างความสะดวกต่อการทำงานของเจ้าหน้าที่อันเนื่องมาจากข้อจำกัดบางประการ หากใช้ระบบทำงานบนอุปกรณ์สมาร์ทโฟนเพียงอย่างเดียว โดยตัวเว็บแอปพลิเคชันนี้ถูก พัฒนาขึ้นด้วย Vue.js มีรายละเอียดการทำงานดังนี้

4.1.1 การเชื่อมต่อ Cloud Firestore

ในการเชื่อมต่อเว็บแอปพลิเคชันกับไฟร์เบสเพื่อใช้บริการต่างๆ ของไฟร์เบส ทำได้ดังนี้

```
1 export default {
2     apiKey: "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
3     authDomain: "e-student-698a5.firebaseio.com",
4     databaseURL: "https://e-student-698a5.
5         firebaseio.com",
6     projectId: "e-student-698a5",
7     storageBucket: "e-student-698a5.appspot.com",
8     messagingSenderId: "000000000000"
```

รูปที่ 4.1: ไฟล์ firebaseConfig.js

จากภาพที่ 4.2 โครงสร้างของไฟล์ firebaseConfig.js สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการส่งออกโมดูลเพื่อใช้งานในไฟล์อื่น
- บรรทัดที่ 2 -7 เป็นการตั้งค่าระบุตัวตนเพื่อใช้งานบริการไฟร์เบส

```

1 import firebase from 'firebase'
2 import 'firebase/firestore'
3 import firebaseConfig from './firebaseConfig'
4
5 export default firebase.initializeApp(firebaseConfig)

```

รูปที่ 4.2: ไฟล์ firebaseInit.js

จากภาพที่ 4.2 โครงสร้างของไฟล์ firebaseInit.js สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการนำเข้าไลบรารีของไฟร์เบส
- บรรทัดที่ 2 เป็นการนำเข้าบริการ Cloud Firestore ของไฟร์เบส
- บรรทัดที่ 8 เป็นการนำเข้าโมดูลตั้งค่าที่ได้จำกัดมาในไฟล์ config.js
- บรรทัดที่ 5 เป็นการส่งออกโมดูลไฟร์เบสเพื่อใช้ในไฟล์อื่น ๆ ซึ่งเมื่อถึงขั้นตอนนี้การใช้อัตราบริการไฟร์เบสถือว่าเป็นอันเสร็จ

4.1.2 โครงสร้างของการสร้างหน้าเข้าสู่ระบบ

```

1 <template>
2   <Row type="flex" justify="center" align="middle">
3     <Col span="8" class="col">
4       <Card style="width:400px">
5         <p slot="title">
6           <Icon type="ios-person" size="20"/></Icon>
7           sign in
8         </p>
9         <a href="#" slot="extra" @click.prevent=""
10            SignUp">
11            create account
12          </a>
13         <Form class="form" ref="formInline" :model=""
14            formInline" :rules="ruleInline">
15           <FormItem prop="email">
16             <Input type="text" v-model="formInline.
17               email" placeholder="email">
18               <Icon type="ios-email" slot="prepend"></
19                 Icon>
20               </Input>
21           </FormItem>
22           <FormItem prop="password">
23             <Input type="password" v-model=""
24               formInline.password" placeholder=""
25               password">
26               <Icon type="ios-locked" slot="prepend"
27                 "></Icon>
28             </Input>
29           </FormItem>
30         <Button type="primary" :loading="loading"
31            @click="handleSubmit('formInline')">
32           <span v-if="!loading">sign in</span>
33           <span v-else>signing in...</span>
34         </Button>
35       </Form>
36     </Card>
37     </Col>
38   </Row>
39 </template>

```

รูปที่ 4.3: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue

จากภาพที่ 4.4 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-33 เป็น템เพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น HTML

– บรรทัดที่ 2-32 เป็นการความคุมลักษณะการแสดงผลบนหน้าจอ

– บรรทัดที่ 3-31 เป็นการกำหนดขนาดของเนื้อภายนอก

– บรรทัดที่ 4-30 เป็นการแสดงเนื้อหาในรูปแบบการ์ด (Card)

– บรรทัดที่ 5-8 เป็นส่วนที่ใช้สำหรับกำหนดหัวเรื่องของการ์ด

– บรรทัดที่ 12 เป็นสร้างฟอร์ม (Form)

– บรรทัดที่ 13 เป็นสร้างช่องกรอกข้อมูลอีเมล (e-mail) จากผู้ใช้

– บรรทัดที่ 18 เป็นสร้างช่องกรอกข้อมูลรหัสผ่าน (password) จากผู้ใช้

– บรรทัดที่ 24 สร้างปุ่มเข้าสู่ระบบ

```

1 data () {
2   return {
3     alert: false,
4     formInline: {
5       email: '',
6       password: ''
7     },
8     ruleInline: {
9       email: [
10         { required: true, message: 'please fill email',
11           trigger: 'blur' }
12       ],
13       password: [
14         { required: true, message: 'please fill password',
15           trigger: 'blur' }
16     ]
17   }
}

```

รูปที่ 4.4: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue

จากภาพที่ 4.4 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ สามารถ

อธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-7 เป็นการสร้างชุดข้อมูลที่ใช้สำหรับการเข้าสู่ระบบ
- บรรทัดที่ 3 ค่าที่ใช้เก็บสถานะของการเข้าสู่ระบบ
- บรรทัดที่ 4-7 เป็นการเก็บข้อมูลอีเมลและรหัสผ่านในรูปแบบ json
- บรรทัดที่ 8-15 เป็นการกำหนดค่าที่ใช้ในการตรวจสอบความถูกต้องของอีเมลและรหัสผ่าน

```

1 userSignIn({commit}, payload) {
2   commit('setLoading', true)
3   firebase.auth().signInWithEmailAndPassword(payload.
4     email, payload.password)
5   .then(firebaseUser => {
6     commit('setUser', firebaseUser)
7     commit('setLoading', false)
8     commit('setError', null)
9   })
10  .catch(error => {
11    commit('setError', error.message)
12    commit('setLoading', false)
13  })

```

รูปที่ 4.5: การสร้างโลจิก (logic) ของหน้าเข้าสู่ระบบ SignIn.vue

จากภาพที่ 4.5 โครงสร้างโลจิกของหน้าเข้าสู่ระบบ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการสร้างฟังก์ชันสำหรับรับข้อมูลที่ในการเข้าสู่ระบบ
- บรรทัดที่ 2 เรียกใช้ฟังก์ชันอื่นเพื่อทำการอัพเดทสถานะการเข้าสู่ระบบ
- บรรทัดที่ 3-12 เป็นการเรียกใช้บริการไฟร์เบส Authentication พร้อมส่งค่า email และ password เพื่อทำการเข้าสู่ระบบ
- บรรทัดที่ 5-7 เป็นการอัพเดทสถานะเมื่อเข้าสู่ระบบสำเร็จ
- บรรทัดที่ 9-12 เป็นการอัพเดทสถานะเมื่อเข้าสู่ระบบไม่สำเร็จ

4.1.3 ໂຄງສ່ຽງຂອງສ່ຽງໜ້າຂ່າວສາຮ

```

1 <template>
2 <div style="padding: 16px;">
3 <Row>
4 <Col span="20" style="padding:16px;">
5 <Row v-for=" (post, index) in postsData" :key="post.id"
      " style="margin-bottom:16px;">
6 <Card>
7 <p slot="title">
8 <Icon type="social-rss-outline"></Icon>
9 post
10 <span style="font-size:11px; color: #95a5a6;">
11 { post.time } 
12 </span>
13 </p>
14 <a href="#" slot="extra" @click.prevent="showData(
    index)">
15 <!-- <Icon type="ios-loop-strong"></Icon> -->
16 detail
17 </a>
18 <p>
19 { post.title } 
20 </p>
21 </Card>
22 </Row>
23 </Col>
24 <Col span="4" style="padding:16px;">
25 <Timeline>
26 <TimelineItem v-for=" (event, index) in eventsData" :
      key="index" >
27 <Icon type="trophy" slot="dot" v-if="index == 0"></
      Icon>
28 <p class="time">{ event.time }</p>
29 <p class="content">{ event.title }</p>
30 </TimelineItem>
31 </Timeline>
32 </Col>
33 </Row>
34 </div>
35 </template>
```

ຮູບທີ 4.6: ການສ່ຽງໜ້າຈອສ່ວນຕິດຕໍ່ຜູ້ໃຊ້ຂອງໜ້າຂ່າວສາຮ Home.vue

จากภาพที่ 4.6 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-33 เป็น템เพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น HTML

- บรรทัดที่ 3-33 เป็นการความคุมลักษณะการแสดงผลบนหน้าจอ
- บรรทัดที่ 6-21 เป็นการแสดงเนื้อหาในรูปแบบการ์ด (Card)
- บรรทัดที่ 10-12 เป็นการแสดงเวลาที่ประกาศข่าว
- บรรทัดที่ 18-20 เป็นการแสดงหัวข้อข่าวสาร
- บรรทัดที่ 25-31 เป็นการแสดงปฏิทินกำหนดการขนาดย่อ

```

1 created() {
2   var vm = this;
3   vm.postsData = [];
4   db.collection("Posts")
5     .orderBy("time", "desc")
6     .get()
7     .then(function(querySnapshot) {
8       querySnapshot.forEach(function(doc) {
9         const data = {
10           id: doc.id,
11           title: doc.data().title,
12           description: doc.data().description,
13           time: doc.data().time.toLocaleString(),
14           fileUrl: doc.data().fileURL[0]
15         }
16         if (vm.postsData) {
17           vm.postsData.push(data);
18         }
19       })
20     })
21   }

```

รูปที่ 4.7: การสร้างโลジค(logic)ของหน้าข่าวสาร Home.vue

จากภาพที่ 4.9 โครงสร้างโลจิกของหน้าข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการฟังก์ชันที่ถูกเรียกทุกครั้งที่ผู้ใช้เปิดหน้าข่าวสาร
- บรรทัดที่ 4-20 เรียกใช้บริการ Cloude Firestore เพื่อทำการสืบค้นข้อมูลข่าวสาร

ทั้งหมดพร้อมทั้งเรียงลำดับตามวันที่ประกาศ

- บรรทัดที่ 9-18 เป็นการเพิ่มข้อมูลเข้าสู่ลิสต์รายการเพื่อใช้ในการแสดงบนหน้าจอ

4.1.4 โครงสร้างของการสร้างหน้าดูรายละเอียดข่าวสาร

```

1 <template>
2   <div>
3     <h2>{{ post.title }}</h2>
4     <p style="font-size:14px;">{{ post.description
      }}</p>
5     <br>
6     
7   </div>
8 </template>
```

รูปที่ 4.8: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าดูรายละเอียดข่าวสาร ViewPost.vue

จากภาพที่ 4.8 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าดูรายละเอียดข่าวสารสามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-8 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น HTML

- บรรทัดที่ 2-7 ครอบทับเนื้อหาทั้งหมดเพื่อให้ง่ายต่อการจัดการการแสดงผล
- บรรทัดที่ 3 เป็นการแสดงหัวข้อข่าวสาร
- บรรทัดที่ 4 เป็นการแสดงรายละเอียดข่าวสาร
- บรรทัดที่ 6 เป็นการแสดงไฟล์แนบ

```

1 created() {
2   let id = this.$route.params.id
3   db.collection('Posts').doc(id).get().then((doc) =>
4     {
5       if(doc.exists) {
6         this.post = doc.data()
7       }
8     })
}
```

รูปที่ 4.9: การสร้างlogicของหน้าดูรายละเอียดของข่าวสาร ViewPost.vue

จากภาพที่ 4.9 โครงสร้างโลจิกของหน้าดูรายละเอียดของข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 เป็นการฟังก์ชันที่ถูกเรียกทุกครั้งที่ผู้ใช้เปิดหน้าดูรายละเอียดข่าวสาร
- บรรทัดที่ 2 ดึงค่าไอดีของประกาศที่ถูกส่งมาจากการหน้าแสดงข่าวสาร
- บรรทัดที่ 3 ทำการสืบค้นข้อมูลข่าวสารจาก Cloud Firestore จากไอดีของประกาศ
- บรรทัดที่ 4-6 เป็นการตรวจสอบว่ามีประกาศตั้งกล่าวอยู่ในฐานข้อมูล Cloude Firestore หรือไม่

4.1.5 โครงสร้างของการสร้างหน้าสนทนา

```

1 <template>
2 <Row :gutter="0" type="flex" justify="center" align="middle">
3 <Col span="16" style="padding: 0px;">
4 <Card style="min-height: 500px;max-height: 500px;" :padding="0">
5 <p v-if="!isAdmin" slot="title" style="text-align: center;">
6 ESP
7 </p>
8 <p v-else slot="title" style="text-align:center;">
9 {{ chatTitle }}</p>
10 <Scroll style="background-color: #EEEEEE;">
11 <ul style="padding:6px;padding-right:8px;">
12 <li v-for="(item, index) in messages" :key="index" style="margin-bottom:8px;">
13 <Card v-if="user.uid !== item.id" :padding="6" style="text-align:left;display: inline-block; background-color: #FAFAFA;">
14 <div>
15 <p>{{ item.message }}</p>
16 </div>
17 </Card>
18 <div v-else style="text-align:right;">
19 <Card :padding="6" style="display: inline-block; background-color: #B2E281;">
20 <p>{{ item.message }}</p>
21 </Card>
22 </div>
23 </li>
24 </ul>
25 </Scroll>
26 <div style="padding:16px;text-align:right;">
27 <Input type="textarea" v-model="formItem.message" placeholder="type..." v-on:keyup.enter="send"></Input>
28 <Button :loading="loading" type="primary" style="margin-top:10px;" size="large" @click="send">send</Button>
29 </div>
30 </Card>
31 </Col>
32 </Row>
33 </template>
```

รูปที่ 4.10: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสนทนา Message.vue

จากภาพที่ 4.10 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสนใจ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-34 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น **HTML**

- บรรทัดที่ 4-22 แสดงหน้าต่างสนใจ
- บรรทัดที่ 9 แสดงชื่อคู่สนใจ
- บรรทัดที่ 11-26 แสดงข้อความสนใจ
- บรรทัดที่ 28 แสดงช่องกรอกข้อความสนใจ
- บรรทัดที่ 29 แสดงปุ่มกดส่งข้อความ

```

1 send() {
2   var vm = this;
3   vm.loading = true;
4   this.formItem.time = new Date();
5
6   // check where data shulde update
7   let key = "";
8   if (this.isAdmin) {
9     key = this.chatId;
10    if (this.chatTitle === "" || this.chatTitle ===
11      null) {
12      return;
13    }
14  } else {
15    key = this.formItem.senderId;
16  }
17
18  this.formItem.name = this.user.displayName;
19  this.formItem.photo = this.user.photoURL
20
21  db
22    .collection("Chats")
23    .doc(key)
24    .collection("messages")
25    .add(this.formItem)
26    .then(function(docRef) {
27      vm.loading = false;
28      vm.formItem.message = "";
29      db
30        .collection("Users")
31        .doc(key)
32        .update({ lastChat: new Date() })
33        .catch(function(error) {
34          vm.$Message.error("send message fail");
35          vm.loading = false;
36        });
37    });
}

```

รูปที่ 4.11: การสร้างลอกิจของหน้าสนทนา Message.vue

จากภาพที่ 4.11 โครงสร้างลอกิจของหน้าสนทนາ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ชื่อฟังก์ชัน
- บรรทัดที่ 2 ตรวจสอบໄວดีของผู้ใช้
- บรรทัดที่ 3 ดึงค่าໂປຣັບໝູນໃຊ້ຄູນປັຈຸບັນ
- บรรทัดที่ 4-6 ເຢີນຂໍ້ມູນລອງຮູນຂໍ້ມູນ Cloud Firestore ໂດຍຮະບຸ path ທີ່ຈະກຳກັນ
- บรรทัดที่ 7 ຈັດເກີບຊຸດຂໍ້ມູນ
- บรรทัดที่ 8-11 ວິທີກຳນົດຂໍ້ມູນເວລາສນຫາລ່າສຸດຂອງຜູ້ໃຊ້
- บรรทัดที่ 12 ແສດສານການວິທີກຳນົດຂໍ້ມູນ

4.1.6 โครงสร้างของการสร้างหน้าປົງທຶນແສດງກຳນົດການ

```

1 <template>
2   <div>
3     <Row :gutter="16">
4       <Col span="6">
5         <Card>
6           <h3>search</h3>
7           <DatePicker v-model="filterDate" format="d-M-yyyy"
8             type="date" size="large" placeholder="select date" style="margin-top:6px;"></
9             DatePicker>
10        </Card>
11      </Col>
12      <Col span="18">
13        <h3>Schedule</h3>
14        <Table border
15          :loading="dataLoading"
16          :columns="columnsName"
17          :data="eventsData"
18          no-data-text="no schedule"
19          style="margin-top:6px;">
20        </Table>
21      </Col>
22    </Row>
23  </div>
24 </template>
```

ຮູບທີ 4.12: ການສ້າງໜ້າຈອສ່ວນຕິດຕໍ່ອື່ນຂອງໜ້າປົງທຶນກຳນົດການ Schedule.vue

จากภาพที่ 4.12 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าปฎิทินกำหนดการสามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-22 เป็น템เพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น

HTML

- บรรทัดที่ 4-9 แสดงหน้าต่างเลือกวันที่เพื่อค้นหา
- บรรทัดที่ 11 แสดงชื่อตาราง
- บรรทัดที่ 12-18 แสดงตารางกำหนดการ

```

1 created() {
2   var vm = this;
3   db.collection("Events")
4     .orderBy("time")
5     .onSnapshot(function(querySnapshot) {
6       vm.dataLoading = false;
7       vm.eventsData = [];
8       querySnapshot.forEach((doc) => {
9         const data = {
10           'id': doc.id,
11           'title': doc.data().title,
12           'description': doc.data().description,
13           'time': `${doc.data().time.getDate()}-${doc.data()
14             .time.getMonth()}-${doc.data().time.
15             getFullYear()}`
16         }
17         vm.eventsData.push(data)
18       })
19     })
20   }

```

รูปที่ 4.13: การสร้างລອຈິກຂອງหน้าປົງປີທຶນກໍາທັນດກສະເໜາ

จากภาพที่ 4.13 โครงสร้างລອຈິກຂອງหน้าປົງປີທຶນກໍາທັນດກສະເໜາອີເມວໄດ້

- บรรทัดที่ 1-18 ຊື່ຝຶກໜັນທີຈະຄູກເຮີຍກຸກຄັ້ງທີ່หน้าປົງປີທຶນກໍາທັນດກຄຽກເປີດ
- บรรทัดที่ 3-17 ສີບຄົນກໍາທັນດກຈາກຮູ້ນ້ອມມູລໂດຍມີການເຮີຍລຳດັບຈາກວັນທີລ່າສຸດ
- ໄປຢັ້ງວັນທີກ່ອນໜ້າ
- บรรทัดທີ່ 9-15 ຈັດເກີບຂໍ້ມູນທີ່ສີບຄົນໄດ້ ເພື່ອໃຫ້ໃນການແສດງຜລບນໜ້າຈອ

4.1.7 โครงสร้างของการสร้างหน้าสร้างประกาศ

```

1 <Modal v-model="modalNewPost"
2 title="add post">
3   <Form :model="formItem" :label-width="80">
4     <FormItem label="title">
5       <Input v-model="formItem.title" placeholder="Enter something..."></Input>
6     </FormItem>
7     <FormItem label="detail">
8       <Input v-model="formItem.description" type="textarea" :autosize="{minRows: 2, maxRows: 5}" placeholder="Enter something..."></Input>
9     </FormItem>
10    <FormItem label="target">
11      <Select v-model="formItem.collection">
12        <Option value="public">public</Option>
13        <Option value="group">group</Option>
14        <Option value="volunteer">volunteer</Option>
15      </Select>
16    </FormItem>
17    <FormItem label="contact list" v-if="tags.length > 0">
18      <Tag closable color="blue" v-for="tag in tags" :key="tag" @on-close="handleClose"> {{ tag }} </Tag>
19    </FormItem>
20    <FormItem label="file">
21      <Upload :before-upload="handleUpload" action="https://shielded-earth-61349.herokuapp.com/">
22        <Button :type="btnAddPostType" icon="ios-cloud-upload-outline"> {{ uploadBtnTitle }} </Button>
23      </Upload>
24    </FormItem>
25  </Form>
26  <div slot="footer">
27    <Button type="primary" :loading="loading" @click="newPost">save</Button>
28    <Button type="ghost" style="margin-left: 8px" @click="modalNewPost = !modalNewPost">cancle</Button>
29  </div>
30 </Modal>
31

```

รูปที่ 4.14: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ MgPost.vue

จากภาพที่ 4.8 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-31 เป็นการต้องการให้แสดงหน้าเพิ่มข่าวสารเป็นป็อปอัพ (Pop up) จึงใช้แท็ก(tag) <Model></Model>
- บรรทัดที่ 2-7 ครอบทับเนื้อหาทั้งหมดเพื่อให้ง่ายต่อการจัดการการแสดงผล
- บรรทัดที่ 3-26 เป็นการสร้างฟอร์มรับข้อมูล
- บรรทัดที่ 4-25 เป็นสร้างช่องรับข้อมูลประกาศ
- บรรทัดที่ 5 เป็นการรับค่าหัวเรื่องประกาศ
- บรรทัดที่ 8 เป็นการรับค่ารายละเอียดประกาศ
- บรรทัดที่ 11-15 เป็นการรับค่ากลุ่มเป้าหมายของประกาศนั้นๆ
- บรรทัดที่ 21-24 เป็นการสร้างปุ่มอัพโหลดไฟล์
- บรรทัดที่ 28 เป็นการสร้างปุ่มบันทึกประกาศ
- บรรทัดที่ 29 เป็นการสร้างปุ่มยกเลิกประกาศ

```

1 newPost() {
2   let key = '';
3   var vm = this;
4   vm.formItem.time = new Date();
5   vm.formItem.tags = vm.tags
6   db.collection("Posts").add(this.formItem)
7   .then(function(docRef) {
8     key = docRef.id;
9     if(vm.file != null){
10       let storageRef = storage.ref('Posts/'+key);
11       let fileRef = storageRef.child(vm.file.name+"");
12       fileRef.put(vm.file).then(function(snapshot) {
13         db.collection("Posts").doc(key)
14         .update(
15           {
16             fileURL: snapshot.metadata.downloadURLs[0],
17             fileType: vm.file.name.slice(vm.file.name.
18               lastIndexOf('.')) + "",
19             fileName: vm.file.name
20           }
21         ) .then(() => {
22           vm.$Notice.success({
23             title: 'Your post was created',
24             desc: ''
25           })
26         .catch(function(error) {
27           vm.$Notice.warning({
28             title: 'create post fail',
29             desc:''
30           });
31         })
32       })
33     }
34   })
35 }

```

รูปที่ 4.15: การสร้างลอกิจของหน้าหน้าสร้างประกาศ MgPost.vue

จากภาพที่ 4.15 โครงสร้างลอกิจของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ชื่อฟังก์ชัน
- บรรทัดที่ 4-5 ข้อมูลของประกาศ
- บรรทัดที่ 6-30 เรียกใช้งาน Cloud Firestore เพื่อบันทึกประกาศลงฐานข้อมูล
- บรรทัดที่ 10-24 เป็นส่วนที่ใช้ในการอัพโหลดเอกสารแนบไปยังไฟร์เบส Storage

- บรรทัดที่ 13-19 เป็นการอัพเดทข้อมูล URL ที่ได้จากการอัปโหลดไฟล์แบบเข้าสู่ฐานข้อมูล Cloud Firestore

4.1.8 โครงสร้างของการสร้างหน้าอัปโหลดเอกสารที่เกี่ยวข้อง

```

1 <Modal v-model="modalNewDoc"
2   title="Upload File">
3   <Form :model="formItem" :label-width="80">
4     <FormItem label="Document title">
5       <Input v-model="formItem.title"></Input>
6     </FormItem>
7     <FormItem label="Document detail">
8       <Input v-model="formItem.description" type="textarea" :autosize="{minRows: 2, maxRows: 5}"></Input>
9     </FormItem>
10    <FormItem label="Select file">
11      <Upload
12        :before-upload="handleUpload"
13        action="https://shielded-earth-61349.herokuapp.
14          com/">
15        <Button :type="type" icon="ios-cloud-upload-
16          outline" > {{ uploadBtnTitle }} </Button>
17      </Upload>
18    </FormItem>
19  </Form>
20  <div slot="footer">
21    <Button type="primary" :loading="loading" @click
22      ="newDoc">upload</Button>
23    <Button type="ghost" style="margin-left: 8px"
24      @click="modalNewDoc = !modalNewDoc">cancle</
25      Button>
26  </div>
27 </Modal>
```

รูปที่ 4.16: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าอัปโหลดเอกสารที่เกี่ยวข้อง MgDocument.vue

จากภาพที่ 4.16 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าอัปโหลดเอกสารที่เกี่ยวข้องสามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-22 สร้างปุ่มอัปโหลดหน้าอัปโหลดไฟล์เอกสาร

- บรรทัดที่ 3-17 สร้างฟอร์ม
- บรรทัดที่ 11-15 สร้างปุ่มอัพโหลดเอกสาร
- บรรทัดที่ 19 สร้างปุ่มบันทึกเอกสาร
- บรรทัดที่ 20 สร้างปุ่มยกเลิกการอัพโหลดเอกสาร

```

1 newDoc () {
2   var vm = this;
3   vm.loading = true;
4   let key = '';
5   this.formItem.time = new Date();
6   db.collection("Docs").add(this.formItem)
7   .then(function(docRef) {
8     key = docRef.id;
9     if(vm.file != null) {
10       let storageRef = storage.ref('Docs/' +key);
11       let fileRef = storageRef.child(vm.file.name+"");
12       fileRef.put(vm.file).then(function(snapshot) {
13         db.collection("Docs").doc(key).update({
14           fileURL: snapshot.metadata.downloadURLs[0],
15           fileType: vm.file.name.slice(vm.file.name.
16             lastIndexOf('.')))
17         }).then(() => {
18           vm.$Notice.success({
19             title: 'Success',
20             desc: ''
21           });
22         }).catch(function(error) {
23           vm.$Notice.warning({
24             title: 'Fail',
25             desc:''
26           });
27         })
}

```

รูปที่ 4.17: การสร้างlogicของหน้าหน้าอัพโหลดเอกสารที่เกี่ยวข้อง MgDocument.vue

จากภาพที่ 4.17 โครงสร้างlogicของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 6-26 เป็นการอัพเดทข้อมูลเอกสารเข้าสู่ฐานข้อมูล Cloud Firestore
- บรรทัดที่ 10-12 เป็นการเรียกใช้ไฟร์เบส Storage เพื่อทำการอัพโหลดไฟล์เอกสาร
- บรรทัดที่ 13-20 ใช้ในการอัพเดทข้อมูล URL ไปยังฐานข้อมูล Cloude Firestore

4.1.9 โครงสร้างของการสร้างหน้าสร้างกำหนดการจองคิวส่งเอกสาร

```

1 <Modal v-model="modalNewQueue">
2   <p slot="header" style="color:#3498db;text-align:
3     center">
4     <Icon type="information-circled"></Icon>
5     <span>Create sumbit document date</span>
6   </p>
7   <div>
8     <Form :model="formItem" :label-width="80">
9       <FormItem label="title">
10      <Input v-model="formItem.title"></Input>
11    </FormItem>
12    <FormItem label="date">
13      <DatePicker v-model="formItem.date" format="d-
14        MMMM-yyyy" type="daterange" placement="bottom-
15        end" placeholder="select" style="width: 200px
16        "></DatePicker>
17    </FormItem>
18    <FormItem label="time">
19      <TimePicker v-model="formItem.time" format="HH:mm
20        " type="timerange" placement="bottom-end"
21        placeholder="select" style="width: 200px"></
22        TimePicker>
23    </FormItem>
24    <FormItem label="students per hr">
25      <InputNumber :max="100" :min="1" v-model="
          formItem.count" style="width: 200px"></
          InputNumber>
26  </Form>
27 </div>
28 <div slot="footer">
29   <Button type="success" size="large" long @click="
30     saveQueue">save</Button>
31 </div>
32 </Modal>
```

รูปที่ 4.18: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างกำหนดการจองคิวส่งเอกสาร

MgQueue.vue

จากภาพที่ 4.18 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างกำหนดการ ของคิวส่งเอกสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-25 การสร้างหน้าต่างปีอปอพ
- บรรทัดที่ 4 ชื่อหน้าต่างปีอปอพ
- บรรทัดที่ 7-20 เป็นการสร้างฟอร์ม
- บรรทัดที่ 23 เป็นการสร้างปุ่มบันทึกกำหนดการ

```

1 saveQueue() {
2   var vm = this;
3   db.collection("Queue").add(this.formItem)
4     .then(function(docRef) {
5       vm.modalNewQueue = false;
6       vm.$Notice.success({
7         title: 'success',
8         desc: ''
9       });
10    }).catch(function() {
11      vm.modalNewQueue = false;
12      vm.$Notice.warning({
13        title: 'fial',
14        desc:''
15      );
16    })
17 }

```

รูปที่ 4.19: การสร้างโลจิกของหน้าสร้างกำหนดการของคิวส่งเอกสาร MgQueue.vue

จากภาพที่ 4.19 โครงสร้างโลจิกของหน้าสร้างกำหนดการของคิวส่งเอกสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 3-15 ที่ได้จากการอัปโหลดไฟล์แบบเข้าสู่ฐานข้อมูล Cloud Firestore

4.2 การพัฒนาเว็บแอปพลิเคชันออนไลน์แบบออฟไลน์

การพัฒนาเว็บแอปพลิเคชันออนไลน์แบบออฟไลน์ คือการพัฒนาเว็บแอปพลิเคชันที่สามารถทำงานได้แม้ไม่มีการเชื่อมต่ออินเทอร์เน็ต หรือเครือข่ายที่เสื่อมเสีย แต่ยังคงสามารถใช้งานได้ต่อเนื่อง ไม่ว่าจะในสถานะไร้สัญญาณอินเทอร์เน็ต หรือเครือข่ายที่ช้าๆ ก็ตาม ทำให้ผู้ใช้งานสามารถเข้าถึงข้อมูลและฟังก์ชันต่างๆ ได้โดยไม่ต้องมีการเชื่อมต่ออินเทอร์เน็ตอย่างต่อเนื่อง ซึ่งเป็นประโยชน์อย่างมากสำหรับผู้ใช้งานที่ต้องเดินทางหรืออยู่ในพื้นที่ที่ไม่มีอินเทอร์เน็ต เช่น บนเครื่องบิน หรือในรถตู้ ที่ต้องการเข้าถึงข้อมูลและฟังก์ชันต่างๆ ของแอปพลิเคชันได้โดยไม่ต้องมีการเชื่อมต่ออินเทอร์เน็ตอย่างต่อเนื่อง

4.2.1 โครงสร้างของการสร้างหน้า MainActivity

```

1 private FirebaseAuth mAuth;
2 private FeedFragment feedFrag = FeedFragment .
    newInstance();
3 private ChatFragment chatFrag = ChatFragment .
    newInstance();
4 private DocumentsFragment docFrag =
    DocumentsFragment.newInstance();
5 private ScheduleFragment sheduleFrag =
    ScheduleFragment.newInstance();
6 private SubmitFragment submitFrag = SubmitFragment .
    newInstance();
7 private UserChatFragment userChatFrag =
    UserChatFragment.newInstance();
8 private CheckinFragment checkinFrag =
    CheckinFragment.newInstance();

```

รูปที่ 4.20: ตัวแปรในคลาส MainActivity

จากภาพที่ 4.20 ตัวแปรที่ประกาศขึ้นเพื่อใช้ในการทำงานของคลาส MainActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร mAuth ใช้ในการจัดเก็บสถานะและข้อมูลของผู้ใช้
- บรรทัดที่ 2 ตัวแปร feedFrag ใช้แสดงผลหน้าจอข่าวสาร
- บรรทัดที่ 3 ตัวแปร chatFrag ใช้แสดงผลหน้าจอสนทนารับเจ้าหน้าที่
- บรรทัดที่ 4 ตัวแปร chatFrag ใช้แสดงผลหน้าจอเอกสารที่เกี่ยวข้อง
- บรรทัดที่ 5 ตัวแปร scheduleFrag ใช้แสดงผลหน้าจอปฏิทินกำหนดการ
- บรรทัดที่ 6 ตัวแปร submitFrag ใช้แสดงผลหน้าจอส่งสำเนาเอกสาร
- บรรทัดที่ 7 ตัวแปร userChatFrag ใช้แสดงผลหน้าจอสนทนารับนักศึกษา
- บรรทัดที่ 8 ตัวแปร checkinFrag ใช้แสดงผลหน้าจอของคิวส่งเอกสาร

```

1 new DrawerBuilder()
2 .addDrawerItems(
3     feed, chat, event, doc, submit, checkin, faq,
4         about, setting, account, logout
5 ).withOnDrawerItemClickListener(new Drawer.
6     OnDrawerItemClickListener() {
7     @Override
8     public boolean onItemClick(View view, int position
9         , IDrawerItem drawerItem) {
10        long id = drawerItem.getIdentifier();
11        if (id == 1) {
12            getSupportFragmentManager().beginTransaction()
13                .replace(R.id.contentContainer, feedFrag)
14                .commit();
15        } else if (id == 2) {
16            if (currentUser.getEmail().contains("tagabee"))
17            {
18                getSupportFragmentManager().beginTransaction()
19                    .replace(R.id.contentContainer, chatFrag)
20                    .commit();
21            } else {
22                getSupportFragmentManager().beginTransaction()
23                    .replace(R.id.contentContainer, userChatFrag)
24                    .commit();
25            }
26        } else if (id == 3) {
27            getSupportFragmentManager().beginTransaction()
28                .replace(R.id.contentContainer, schedleFrag)
29                .commit();
30        } else if (id == 4) {
31            getSupportFragmentManager().beginTransaction()
32                .replace(R.id.contentContainer, docFrag)
33                .commit();
34        } else if (id == 5) {
35            getSupportFragmentManager().beginTransaction()
36                .replace(R.id.contentContainer, submitFrag)
37                .commit();
38        } else if (id == 6) {
39            getSupportFragmentManager().beginTransaction()
40                .replace(R.id.contentContainer, checkinFrag)
41                .commit();
42    } else if (id == 11) {
43        mAuth.signOut();
        startActivity(new Intent(MainActivity.this,
            MainActivity.class));
        finish();
    }
}

```

รูปที่ 4.21: โค้ดส่วนที่ใช้ในการสร้างเมนูนำทางหลักภายในคลาส MainActivity

จากภาพที่ 4.21 สามารถอธิบายการทำงานโค้ดส่วนที่ใช้ในการสร้างเมนูนำทางหลักภายในคลาส MainActivity ได้ดังนี้

- บรรทัดที่ 1 เป็นการสร้างเมนูนำทาง
- บรรทัดที่ 2-3 เป็นการเพิ่ม Fragment ต่างๆ เข้าไปยังเมนูนำทาง
- บรรทัดที่ 4-6 เป็นการเพิ่มการดักจับอีเวนต์ (Event) เพื่อสลับหน้าจอการแสดงผลที่เกิดขึ้นเมื่อผู้ใช้กดที่เมนูนำทาง
- บรรทัดที่ 8-11 เป็นการแสดงผลหน้าข่าวสาร
- บรรทัดที่ 13-16 เป็นการแสดงผลหน้าสนทนาร่วมกันที่
- บรรทัดที่ 17-20 เป็นการแสดงผลหน้าสนทนาร่วมกันศึกษา
- บรรทัดที่ 22-25 เป็นการแสดงผลหน้าปฏิทินกำหนดการ
- บรรทัดที่ 26-29 เป็นการแสดงผลหน้าดาวน์โหลดเอกสาร
- บรรทัดที่ 30-33 เป็นการแสดงผลหน้าส่งสำเนาเอกสาร
- บรรทัดที่ 34-37 เป็นการแสดงผลหน้าของคิวส่งเอกสาร
- บรรทัดที่ 38-41 เป็นการรีเฟรช(refresh)หน้าจอเมื่อผู้ใช้กดปุ่มออกจากระบบ

4.2.2 โครงสร้างของการสร้างหน้า FeedFragment

```

1 private RecyclerView recyclerView;
2 private FirebaseFirestore db;
3 private ArrayList<Post> posts;
4 private FeedItemAdapter adapter;
```

รูปที่ 4.22: ตัวแปรในคลาส FeedFragment

จากภาพที่ 4.22 ตัวแปรที่ประกาศขึ้นเพื่อใช้ในการทำงานของคลาส FeedFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร recyclerView ใช้ในการแสดงข้อมูลลิสตรายการข่าวสาร
- บรรทัดที่ 2 ตัวแปร db ใช้ในการสืบค้นข้อมูลจากข่าวสารจาก Cloud Firestore
- บรรทัดที่ 3 ตัวแปร posts ใช้ในการเก็บชุดข้อมูลที่ได้จากการสืบค้นข้อมูล
- บรรทัดที่ 4 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลเป็นลิสตรายการเพื่อแสดงบน recyclerView

```

1 db.collection("Posts")
2 .orderBy(getString(R.string.key_time), Query.
    Direction.DESCENDING)
3 .get()
4 .addOnCompleteListener(new OnCompleteListener<
    QuerySnapshot>() {
5     @Override
6     public void onComplete(@NonNull Task<QuerySnapshot
        > task) {
7         if (task.isSuccessful() && isAdded()) {
8             for (DocumentSnapshot document : task.getResult
                 ()) {
9                 Log.d(TAG, document.getId() + " => " +
                    document.getData());
10                Map<String, Object> data = document.getData();
11                Post post = new Post();
12                post.setTitle(data.get(getString(R.string.
                    key_title)).toString());
13                post.setCollection(data.get(getString(R.string
                    .key_collection)).toString());
14                post.setDate((Date) data.get(getString(R.string
                    .key_time)));
15                post.setDescription(data.get(getString(R.
                    string.key_description)) == null ? "" :
                    data.get(getString(R.string.key_description
                    )).toString());
16                post.setFileURL(data.get(getString(R.string.
                    key_fileURL)) == null ? "" : data.get(
                    getString(R.string.key_fileURL)).toString());
17                post.setFileName(data.get(getString(R.string.
                    key_fileName)) == null ? "" : data.get(
                    getString(R.string.key_fileName)).toString());
18                posts.add(post);
19            }
20            recyclerView.setLayoutManager(new
                LinearLayoutManager(getActivity()));
21            recyclerView.setAdapter(adapter);
22            adapter.notifyDataSetChanged();
23        } else {
24            Log.w(TAG, "Error getting documents.", task.
                getException());
25        }
26    }
27 });

```

รูปที่ 4.23: โค้ดส่วนที่ใช้ในการสืบค้นข้อมูลจาก Cloude Firestore ภายในคลาส FeedFragment

จากภาพที่ 4.23 โค้ดส่วนที่ใช้ในการสืบค้นข้อมูลจาก Cloude Firestore ภายในคลาส FeedFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-3 เริ่มทำการสืบค้นข้อมูลประการทั้งหมดพร้อมทั้งเรียงลำดับข้อมูลจากประการล่าสุดก่อน
- บรรทัดที่ 4-18 รับผลการสืบค้นพร้อมทั้งเพิ่มข้อมูลที่ได้แต่ละแฉวเข้าไว้ที่ตัวแปร posts
- บรรทัดที่ 20-22 ทำการอัพเดทข้อมูลที่แสดงอยู่บน recyclerView

```

1  @Override
2  public void recyclerViewListClicked(View v, int
3      position) {
4      Intent intent = new Intent(getActivity(),
5          PostDetailActivity.class);
6      intent.putExtra(getString(R.string.key_title),
7          posts.get(position).getTitle());
8      intent.putExtra(getString(R.string.key_collection)
9          , posts.get(position).getCollection());
10     intent.putExtra(getString(R.string.key_time),
11         posts.get(position).getDate());
12     intent.putExtra(getString(R.string.key_description)
13         , posts.get(position).getDescription());
14     intent.putExtra(getString(R.string.key_fileURL),
15         posts.get(position).getFileURL());
16     intent.putExtra(getString(R.string.key_fileName),
17         posts.get(position).getFileName());
18     if (getActivity() != null)
19         getActivity().startActivity(intent);
20 }
```

รูปที่ 4.24: โค้ดส่วนที่ใช้ในการดักอีเวนต์เมื่อผู้ใช้กดที่ແລວประการในคลาส FeedFragment

จากภาพที่ 4.24 โค้ดส่วนที่ใช้ในการดักอีเวนต์เมื่อผู้ใช้กดที่ແລວประการในคลาส FeedFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ประการตัวแปร intent ประเภทตัวแปร Intent เพื่อใช้กำหนดแอคชันที่วิธีปลายทางซึ่งในนี้คือ PostDetailActivity
- บรรทัดที่ 4-9 เป็นการเพิ่มข้อมูลเข้าเก็บไว้ที่ตัวแปร intent โดยดึงข้อมูลใน posts

มาจากตำแหน่งແກ່ວິທີຜູ້ໃຊ້ດີ

- ບຽບທັດທີ 11 ເປັນການທຳມະນຸດຂອງແອຄຫິວດີ່ PostDetailActivity

4.2.3 ໂຄງສໍາງຂອງການສ້າງໜ້າ PostDetailActivity

```

1 private TextView tvTitle, tvDescription,
      tvCollection, tvDate;
2 private String strTitle, strDescription, strDate,
      strCollection, strFileURL, strFileName;
3 private FloatingActionButton fab;
4 private DownloadManager downloadManager;
5
6 strTitle = getIntent().getStringExtra(getString(R.
      string.key_title));
7 strDescription = getIntent().getStringExtra(
      getString(R.string.key_description));
8 strDate = getIntent().getStringExtra(getString(R.
      string.key_time));
9 strCollection = getIntent().getStringExtra(getString
      (R.string.key_collection));
10 strFileURL = getIntent().getStringExtra(getString(R.
      string.key_fileURL));
11 strFileName = getIntent().getStringExtra(getString(R
      .string.key_fileName));
12 tvTitle.setText(strTitle);
13 tvDescription.setText(strDescription);
14 tvDate.setText(strDate);
15 tvCollection.setText(strCollection);

```

ຮູບທີ 4.25: ໂຄງສໍາງທີ່ໃໝ່ໃນການແສດງຜລຣາຍລະເອີຍດປະກາສຂອງຄລາສ PostDetailActivity

ຈາກພາບທີ 4.25 ໂຄງສໍາງທີ່ໃໝ່ໃນການແສດງຜລຣາຍລະເອີຍດປະກາສຂອງຄລາສ PostDetailActivity
ສາມາດອອີບາຍໄດ້ຕັ້ງນີ້

- ບຽບທັດທີ 1-4 ເປັນການປະກາສຕັ້ງແປຣທີ່ໃໝ່ໃນການເກີບຂໍ້ມູນປະກາສ
- ບຽບທັດທີ 6-11 ເປັນການຕຶງຄ່າທີ່ຄູກສ່າງມາຈາກຄລາສ FeedFragment ຜ່ານທາງ Intent
- ບຽບທັດທີ 12-15 ເປັນການແສດງຜລຂໍ້ມູນຕ່າງໆ ອອກທາງໜ້າລວມແສດງຜລ

```

1  @Override
2  public void onClick(View v) {
3      int id = v.getId();
4      if (id == R.id.fab) {
5          downloadManager = (DownloadManager)
6              getSystemService(Context.DOWNLOAD_SERVICE);
7          Uri uri = Uri.parse(strFileURL);
8          DownloadManager.Request request = new
9              DownloadManager.Request(uri);
10         request.setNotificationVisibility(DownloadManager.
11             Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
12         request.setDestinationInExternalPublicDir(
13             Environment.DIRECTORY_DOWNLOADS, strFileName);
14         downloadManager.enqueue(request);
15     }
16 }
```

รูปที่ 4.26: โค้ดส่วนที่ใช้ในการดาวน์โหลดเอกสารของคลาส PostDetailActivity

จากภาพที่ 4.26 โค้ดส่วนที่ใช้ในการดาวน์โหลดเอกสารของคลาส PostDetailActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-2 เช็คว่าอีเวนต์ที่เกิดขึ้นมาจากปุ่มดาวน์โหลดเอกสารหรือไม่
- บรรทัดที่ 5-9 เป็นการเตรียมความพร้อมสำหรับการดาวน์โหลดเอกสาร
- บรรทัดที่ 10 ทำการเริ่มดาวน์โหลดเอกสาร

4.2.4 โครงสร้างของการสร้างหน้า ChatActivity

```

1 private FirebaseFirestore db;
2 private MessagesListAdapter adapter;
3 private MessagesList messagesList;
4 private String senderId;
5 private String name;
6 private String avatar;
7 private FirebaseAuth mAuth;
8 private EditText tvMessage;
9 private FirebaseUser currentUser;
10 private Button btnSend;
```

รูปที่ 4.27: โค้ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส ChatActivity

จากภาพที่ 4.27 โค้ดส่วนที่ใช้ในการแสดงผลรายละเอียดประการของคลาส ChatActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร db ใช้ในการสืบค้นข้อมูลจาก Cloud Firestore
- บรรทัดที่ 2 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลที่ได้จากการสืบค้นเป็นลิสต์รายการ
- บรรทัดที่ 3 ตัวแปร messagesList ใช้ในการแสดงบทสนทนา
- บรรทัดที่ 4 ตัวแปร senderId ใช้ในการจัดเก็บไอดีของผู้ใช้
- บรรทัดที่ 5 ตัวแปร name ใช้ในการจัดเก็บชื่อของผู้ใช้
- บรรทัดที่ 6 ตัวแปร avatar ใช้ในการจัดเก็บ url รูปภาพของผู้ใช้
- บรรทัดที่ 7 ตัวแปร mAuth ใช้ในการจัดเก็บสถานะและข้อมูลของผู้ใช้
- บรรทัดที่ 8 ตัวแปร tvMessage ใช้ในการกรอกข้อความ
- บรรทัดที่ 9 ตัวแปร currentUser ใช้ในการจัดเก็บสถานะและข้อมูลของผู้ใช้คนปัจจุบัน
- บรรทัดที่ 10 ตัวแปร btnSend เป็นปุ่มที่ใช้สำหรับกดส่งข้อความ

```

1 db.collection("Chats").document(key).collection("messages")
2 .orderBy("time")
3 .addSnapshotListener(new EventListener<QuerySnapshot>() {
4 @Override
5 public void onEvent(QuerySnapshot documentSnapshots,
6   FirebaseFirestoreException e) {
7   if (e != null) {
8     System.err.println("Listen failed:" + e);
9     return;
10 }
11 String id;
12 String usrId;
13 String text;
14 Date createdAt;
15 adapter.clear();
16 for (DocumentSnapshot document : documentSnapshots)
17   {
18     Map<String, Object> data = document.getData();
19     id = document.getId();
20     text = data.get("message").toString();
21     createdAt = (Date) data.get("time");
22     usrId = data.get("senderId").toString();
23     Message m = new Message(id, text, createdAt, new
24       User(usrId, name, avatar));
25     adapter.addToStart(m, true);
26     adapter.notifyDataSetChanged();
27   }
28 }
29 );

```

รูปที่ 4.28: โค้ดส่วนที่ใช้ในการสืบค้นประวัติการสนทนาของคลาส ChatActivity

จากภาพที่ 4.28 โค้ดส่วนที่ใช้ในการสืบค้นประวัติการสนทนาของคลาส ChatActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-2 ทำการสืบค้นข้อมูลประวัติการสนทนาจาก Cloud Firestore
- บรรทัดที่ 11-14 สร้างตัวแปรเพื่อใช้ในการจัดเก็บข้อมูล
- บรรทัดที่ 16-24 ทำการอ่านค่าและเพิ่มเข้าลิสต์รายการ

```

1 final Map<String, Object> map = new HashMap<>();
2 map.put("message", tvMessage.getText().toString());
3 map.put("time", new Date());
4 map.put("senderId", currentUser.getUid());
5 map.put("name", currentUser.getDisplayName());
6 map.put("photo", currentUser.getPhotoUrl().toString()
7     ());
8 tvMessage.setText("");
9 db.collection("Chats").document(senderId)
10 .collection("messages")
11 .add(map)
12 .addOnCompleteListener(new OnCompleteListener<
13     DocumentReference>() {
14     @Override
15     public void onComplete(@NonNull Task<
16         DocumentReference> task) {
17         Map<String, Object> map1 = new HashMap<>();
18         map1.put("lastChat", new Date());
19         db.collection("Users")
20         .document(senderId)
21         .update(map1);
22     }
23 });

```

รูปที่ 4.29: โค้ดส่วนที่ใช้ในการส่งข้อความของคลาส ChatActivity

จากภาพที่ 4.29 โค้ดส่วนที่ใช้ในการส่งข้อความของคลาส ChatActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-6 สร้าง HashMap เพื่อใช้จัดเก็บข้อความ
- บรรทัดที่ 8-11 เป็นการเพิ่มชุดข้อมูลเข้าสู่ Cloud Firestore
- บรรทัดที่ 14-18 เป็นการอัพเดทเวลาสนทนาร่วมกับผู้ใช้

4.2.5 โครงสร้างของการสร้างหน้า SignInActivity

```

1 private FirebaseAuth mAuth;
2 private String email, password;
3 private EditText userName, userPassword;
4 private ProgressBar simpleProgressBar;
```

รูปที่ 4.30: โค้ดส่วนที่ใช้ในการแสดงผลหน้าเข้าสู่ระบบของคลาส SignInActivity

จากภาพที่ 4.30 โค้ดส่วนที่ใช้ในการแสดงผลหน้าเข้าสู่ระบบของคลาส SignInActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร mAuth ใช้ในการจัดเก็บสถานะการเข้าสู่ระบบของผู้ใช้
- บรรทัดที่ 2 ตัวแปร email จัดเก็บอีเมลของผู้ใช้
- บรรทัดที่ 2 ตัวแปร password จัดเก็บรหัสผ่านของผู้ใช้
- บรรทัดที่ 3 ตัวแปร userName ใช้ในการรับค่าอีเมลจากผู้ใช้
- บรรทัดที่ 3 ตัวแปร userPassword ใช้ในการรับค่ารหัสผ่านจากผู้ใช้

```

1  email = userName.getText().toString();
2  password = userPassword.getText().toString();
3  if(email.isEmpty() || email == null || password.
4      isEmpty() || password == null ){
5      Toast.makeText(SignInActivity.this, "Please fill
6          data!", Toast.LENGTH_SHORT).show();
7      return;
8  }
9  simpleProgressBar.setVisibility(View.VISIBLE);
10 mAuth.signInWithEmailAndPassword(email, password)
11     .addOnCompleteListener(this, new OnCompleteListener
12         <AuthResult>() {
13             @Override
14             public void onComplete(@NonNull Task<AuthResult>
15                 task) {
16                 if (task.isSuccessful()) {
17                     startActivity(new Intent(SignInActivity.this,
18                         MainActivity.class));
19                     finish();
20                 } else {
21                     Toast.makeText(SignInActivity.this, "
22                         Authentication failed.",
23                         Toast.LENGTH_SHORT).show();
24                 }
25                 simpleProgressBar.setVisibility(View.INVISIBLE);
26             }
27         }
28     );

```

รูปที่ 4.31: โค้ดส่วนที่ใช้ในการเข้าสู่ระบบของคลาส SignInActivity

จากภาพที่ 4.31 โค้ดส่วนที่ใช้ในการเข้าสู่ระบบของคลาส SignInActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร email จัดเก็บอีเมลของผู้ใช้
- บรรทัดที่ 2 ตัวแปร password จัดเก็บรหัสผ่านของผู้ใช้
- บรรทัดที่ 3 ตรวจสอบความถูกต้องของข้อมูล
- บรรทัดที่ 8-21 เป็นการเรียกใช้เฟร์เบส Authentication เพื่อเข้าใช้งานระบบ

4.2.6 โครงสร้างของการสร้างหน้า ScheduleFragment

```

1 private RecyclerView recyclerView;
2 private FirebaseFirestore db;
3 private ArrayList<Event> events;
4 private EventItemAdapter adapter;
5 private DatePickerTimeline datePicker;

```

รูปที่ 4.32: โค้ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment

จากภาพที่ 4.34 โค้ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร recyclerView ใช้ในการแสดงลิสต์รายการกำหนดการ
- บรรทัดที่ 2 ตัวแปร db ใช้ในการสืบค้นข้อมูลจาก Cloud Firestore
- บรรทัดที่ 3 ตัวแปร events จัดเก็บข้อมูลกำหนดการ
- บรรทัดที่ 4 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลกำหนดการเป็นลิสต์รายการเพื่อใช้แสดงใน recyclerView
- บรรทัดที่ 5 ตัวแปร datePicker ใช้ในการแสดงปฏิทิน

```

1 db.collection("Events")
2 .orderBy(getString(R.string.key_time), Query.
    Direction.DESCENDING)
3 .get()
4 .addOnCompleteListener(new OnCompleteListener<
    QuerySnapshot>() {
5     @Override
6     public void onComplete(@NonNull Task<QuerySnapshot>
        task) {
7         if (task.isSuccessful() && isAdded()) {
8             for (DocumentSnapshot document : task.getResult
                 ()) {
9                 Log.d(TAG, document.getId() + " => " +
                    document.getData());
10                Map<String, Object> data = document.getData();
11                Date dbDate = (Date) data.get(getString(R.
                    string.key_time));
12                if (dbDate.getDate() == datePicker.
                    getSelectedDay() && dbDate.getMonth() ==
                    datePicker.getSelectedMonth()) {
13                    Event event = new Event();
14                    event.setTitle(data.get(getString(R.string.
                        key_title)).toString());
15                    event.setDescription(data.get(getString(R.
                        string.key_description)).toString());
16                    event.setTime((Date) data.get(getString(R.
                        string.key_time)));
17                    events.add(event);
18                }
19            }
20            recyclerView.setLayoutManager(new
                LinearLayoutManager(getActivity()));
21            recyclerView.setAdapter(adapter);
22            adapter.notifyDataSetChanged();
23        } else {
24            Log.w(TAG, "Error getting documents.", task.
                getException());
25        }
26    }
27}
28);

```

รูปที่ 4.33: โค้ดส่วนที่ใช้ในการสืบค้นข้อมูลกำหนดการของคลาส ScheduleFragment

จากภาพที่ 4.33 โค้ดส่วนที่ใช้ในการสืบค้นข้อมูลกำหนดการของคลาส ScheduleFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-4 เป็นการสืบค้นข้อมูลกำหนดการโดยเรียงลำดับข้อมูลล่าสุดก่อน
- บรรทัดที่ 7-19 เป็นการนำข้อมูลที่ได้จากการสืบค้นแปลงเป็นลิสต์รายการและแสดงผล
- บรรทัดที่ 20-22 เป็นการอัพเดทลิสต์รายการ

4.2.7 โครงสร้างของการสร้างหน้า ScheduleFragment

```

1 private RecyclerView recyclerView;
2 private FirebaseFirestore db;
3 private ArrayList<Event> events;
4 private EventItemAdapter adapter;
5 private DatePickerTimeline datePicker;
```

รูปที่ 4.34: โค้ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment

จากภาพที่ 4.34 โค้ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร recyclerView ใช้ในการแสดงลิสต์รายการกำหนดการ
- บรรทัดที่ 2 ตัวแปร db ใช้ในการสืบค้นข้อมูลจาก Cloud Firestore
- บรรทัดที่ 3 ตัวแปร events จัดเก็บข้อมูลกำหนดการ
- บรรทัดที่ 4 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลกำหนดการเป็นลิสต์รายการเพื่อใช้แสดงใน recyclerView
- บรรทัดที่ 5 ตัวแปร datePicker ใช้ในการแสดงปฏิทิน

บทที่ 5

การทดสอบระบบ

การทดสอบการทำงานของแอปพลิเคชันระบบ กองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานีและทดสอบการทำงานในส่วนของเว็บไซต์ โดยทำการทดสอบในลักษณะ Black-box Testing [?] หรือ Data-Driven testing ซึ่งเป็นการทดสอบแบบที่ไม่สนใจโปรเซส (Process) การทำงานภายในของโปรแกรมว่าทำงานอย่างไร แต่จะเน้นไปที่ Input และ Result ที่ได้มากกว่าการทำงานต่าง ๆ ถูกต้องตามความต้องการ (Requirement) หรือไม่ ซึ่งการทดสอบการใช้งานแอปพลิเคชัน หรือไม่ ซึ่งการทดสอบการใช้งานแอปพลิเคชัน ได้ผลดังนี้

5.1 การทดสอบการใช้งานแอปพลิเคชัน

- การทดสอบการใช้งานเมนูนำทางของแอปพลิเคชัน การทดสอบเมนูนำทางของแอปพลิเคชันในการนำทางผู้ใช้งาน ซึ่งเมนูหลักประกอบด้วย เมนูหน้าประกาศ เมนูหน้าสนทนา เมนูหน้าปฏิทินกำหนดการ เมนูหน้าดาวน์โหลดเอกสาร เมนูส่งภาพถ่ายสำเนาเอกสาร เมนูหน้าจองคิวส่งเอกสาร เมนูหน้าคำทำที่พับบอym เมนูหน้าเกี่ยวกับ เมนูหน้าข้อมูลส่วนตัวและเมนูออกจากระบบ ผลทดสอบดังตารางที่ 5.9-5.2

ตารางที่ 5.1: ผลการทดสอบเมนูนำทาง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เมนูประกาศ	กดปุ่มเมนูประกาศ	ระบบแสดงผลหน้าจอประกาศ พร้อมทั้งแสดงรายการประกาศทั้งหมด
เมนูสนทนა	กดปุ่มเมนูสนทนა	ระบบแสดงผลหน้าจอสนทนາ พร้อมทั้งแสดงข้อมูลประวัติการสนทนາ
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอสนทนາ พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงผลหน้าจอสนทนາ พร้อมทั้งแสดงข้อมูลประวัติการสนทนາ
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอสนทนາ พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลด เอกสาร	ระบบแสดงผลหน้าจอรายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอสนทนາ พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนู หน้า ส่ง ภาพถ่าย สำเนา เอกสาร	กดปุ่มเมนูหน้าส่งภาพสำเนา เอกสาร	ระบบแสดงผลหน้าส่งเอกสาร ภาพสำเนาเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอสนทนາ พร้อมทั้งแสดงรายการข่าวสารทั้งหมด

ตารางที่ 5.2: ผลการทดสอบเมนูนำทาง(ต่อ)

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เมนูหน้าจองคิวส่งเอกสาร	กดปุ่มเมนูหน้าจองคิวส่งเอกสาร	ระบบแสดงผลหน้าจองคิวส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอข่าวสารพร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าคำตามที่พับบออย	กดปุ่ม เมนู หน้า คำตาม ที่ พับ บออย	ระบบแสดงหน้าจองคิวส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าเกี่ยวกับ	กดปุ่มเมนูหน้าเกี่ยวกับ	ระบบแสดงผลหน้าเกี่ยวกับซึ่งแสดงข้อมูลผู้พัฒนารวมไปถึงแสดงเครดิต (credit) ไลบรารีต่าง ๆ ที่ใช้งานภายในแอปพลิเคชัน
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าบัญชีผู้ใช้	กดปุ่มเมนูหน้าบัญชีผู้ใช้	ระบบแสดงผลหน้าจอข้อมูลส่วนตัวโดยมีข้อมูลรูปประจำตัว ชื่อผู้ใช้ สาขาวิชาและภาควิชา
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูออกจากระบบ	กดปุ่มเมนูออกจากระบบ	ทำการออก จาก ระบบ และแสดงหน้าจอข่าวสาร

- การทดสอบหน้ารายละเอียดประกาศในการแสดงผลหน้าจอรายละเอียดประกาศนั้น จะประกอบไปด้วยหัวเรื่องประกาศ รายละเอียดประกาศ วันที่ประกาศและเอกสาร แบบ ผลการทดสอบดังตารางที่ 5.10

ตารางที่ 5.3: ผลการทดสอบหน้ารายละเอียดประกาศ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศทั้งหมด
	กดปุ่มอ่านรายละเอียดประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กดปุ่มดาวน์โหลด เอกสาร แบบ	ระบบแสดงผลการดาวน์โหลดเอกสารแบบ
	เมื่อดาวน์โหลดเสร็จ กดปุ่มเปิดเอกสาร	ระบบแสดงผลเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กดปุ่มย้อนกลับอีกครั้ง	ระบบแสดงผลหน้าจอประ-

- การทดสอบหน้าสนทนา ในการแสดงผลหน้าจอสนทนานั้นจะประกอบไปด้วยราย-การประวัติการสนทนา ซึ่งกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.11

ตารางที่ 5.4: ผลการการทดสอบหน้าสนทนา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาพร้อมทั้งแสดงรายการประวัติการสนทนา
	กดปุ่มที่ซึ่งกรอกข้อความ	ระบบแสดงตัวกระพิรบ (cursor) เพื่อชี้ให้รู้ว่าตำแหน่งของการพิมพ์อักษรจะ
	พิมพ์อักษร	ระบบแสดงผลอักษรที่ถูกพิมพ์
	กดปุ่มส่งข้อความ	ระบบแสดงข้อความที่ถูกพิมพ์บนรายการประวัติสนทนาล่าสุด
	กดปุ่มย้อนกลับ	ระบบแสดงผู้ลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าปฏิทินกำหนดการ ในการแสดงผลหน้าปฏิทินกำหนดการนั้นจะประกอบไปด้วยรายการประวัติการสนทน่า ซ่องกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.12

ตารางที่ 5.5: ผลการการทดสอบหน้าปฏิทินกำหนดการ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงหน้าจอปฏิทินกำหนดการ โดยมี การ แสดง กำหนดการของวันปัจจุบัน
	กดเลือกวันที่ต้องการดูกำหนดการในปฏิทิน	ระบบ แสดง กำหนดการ ของ วันที่ถูกเลือก
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประ- กานต์ พร้อม ทั้ง แสดง รายการ ข่าวสารทั้งหมด

- การทดสอบหน้าดาวน์โหลดเอกสาร ในการแสดงผลหน้าจอดาวน์โหลดเอกสารนั้นจะ ประกอบไปด้วยรายการเอกสารโดยที่แต่ละฉบับจะแสดงชื่อเอกสารและปุ่มดาวน์โหลดเอกสาร ผลการทดสอบดังตารางที่ 5.13

ตารางที่ 5.6: ผลการการทดสอบหน้าดาวน์โหลดเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลด เอกสาร	ระบบ แสดง ผล หน้า จอ ราย- การเอกสารในระบบพร้อมทั้ง แสดงปุ่มดาวน์โหลด
	กดปุ่มดาวน์โหลดเอกสาร	ระบบแสดงผลการดาวน์โหลด เอกสาร
	เมื่อ ดาวน์โหลด เสร็จ กด ปุ่ม เปิดเอกสาร	ระบบแสดงผลเอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ราย- การเอกสารในระบบพร้อมทั้ง แสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผู้ล หน้า จอ ประ- กษา พร้อม ทั้ง แสดง รายการ ข่าวสารทั้งหมด

- การทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร ในการแสดงผลหน้าส่งภาพถ่ายสำเนาเอกสารนั้นจะประกอบไปด้วยปุ่มเพิ่มเอกสารฉบับที่ 1 ปุ่มเพิ่มเอกสารฉบับที่ 2 และปุ่มส่งเอกสาร ผลการทดสอบดังตารางที่ 5.7

ตารางที่ 5.7: ผลการทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้า ส่ง ภาพถ่าย สำเนา เอกสาร	กดปุ่มเมนูหน้าจอของคิวส่งเอกสาร	ระบบ แสดง หน้า จอ คิว ส่งเอกสาร
	กดปุ่มเพิ่มเอกสารฉบับที่ 1	ระบบแสดงหน้าจอกล่องถ่ายภาพ
	กดปุ่มถ่ายภาพเอกสาร	ระบบแสดงผลภาพเอกสาร
	กดปุ่มถัดไป	ระบบ แสดง ผล หน้า ปรีป แต่งภาพเอกสาร
	กดปุ่มยืนยัน	ระบบ แสดง ผล ภาพ หน้า ส่ง ภาพถ่าย สำเนา เอกสาร พร้อม ทั้ง แสดง ผล ภาพ เอกสาร ฉบับที่ 1
	กดปุ่มเพิ่มเอกสารฉบับที่ 2	ระบบ แสดง หน้า จอกล่องถ่ายภาพ
	กดปุ่มถ่ายภาพเอกสาร	ระบบ แสดง ผล ภาพเอกสาร
	กดปุ่มถัดไป	ระบบ แสดง ผล หน้า ปรีป แต่งภาพเอกสาร
	กดปุ่มยืนยัน	ระบบ แสดง ผล ภาพ หน้า ส่ง ภาพถ่าย สำเนา เอกสาร พร้อม ทั้ง แสดง ผล ภาพ เอกสาร ฉบับที่ 1 และฉบับที่ 2
กดส่งเอกสาร	กดส่งเอกสาร	ระบบ แสดง ผล การ ส่ง เอกสาร และ แสดง สถานะ การ ตรวจ เอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผู้ หน้า จอ ประ กากศ พร้อม ทั้ง แสดง รายการ ข่าวสาร ทั้งหมด

- การทดสอบหน้าจอคิวส่งเอกสาร ในการแสดงผลหน้าจอคิวส่งเอกสารนั้นจะประกอบไปด้วยปุ่มกดเลือกวันที่ ปุ่มกดเลือเวลา ผลการทดสอบดังตารางที่ 5.8

ตารางที่ 5.8: ผลการทดสอบหน้าจอคิวส่งเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าจอคิวส่งเอกสาร	กดปุ่มเมนูหน้าจอคิวส่งเอกสาร	ระบบแสดงผลหน้าจอกำหนดการส่งเอกสาร
	กดปุ่มเลือกวันที่ ต้องการ ส่งเอกสาร	ระบบแสดงผลวันที่ถูกเลือก
	กดปุ่มเลือกเวลาที่ต้องการส่งเอกสาร	ระบบแสดงผลเวลาที่ถูกเลือกพร้อมทั้งแสดงปุ่มกดบันทึก
	กดปุ่มบันทึก	ระบบแสดงผลการจองวันที่ส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

5.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน

- การทดสอบการใช้งานเมนูนำทางของเว็บแอปพลิเคชัน การทดสอบเมนูนำทางของเว็บแอปพลิเคชันในการนำทางผู้ใช้งาน ซึ่งเมนูหลักประกอบด้วย เมนูหน้าประกาศ เมนูหน้าสนทนา เมนูหน้าปฏิทินกำหนดการ เมนูหน้าดาวน์โหลดเอกสาร เมนูหน้าคำถามทีพับบอย เมนูหน้าเกี่ยวกับและเมนูออกจากระบบ ผลทดสอบดังตารางที่ 5.9-

5.2

ตารางที่ 5.9: ผลการทดสอบเมนูนำทาง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เมนูประ公示	กดปุ่มเมนูประ公示	ระบบแสดงผลหน้าจอประ公示พร้อมทั้งแสดงรายการประ公示ทั้งหมด
เมนูสนทนา	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาร่วมทั้งแสดงข้อมูลประวัติการสนทนากลับ
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประ公示พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงผลหน้าจอสนทนาร่วมทั้งแสดงข้อมูลประวัติการสนทนากลับ
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประ公示พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลด เอกสาร	ระบบแสดงผลหน้าจอตารางรายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประ公示พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าคำตามที่พบป้อย	กด ปุ่ม เมนู หน้า คำตาม ที่ พบ ป้อย	ระบบแสดงหน้าจองคิวส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประ公示พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูหน้าเกี่ยวกับ	กดปุ่มเมนูหน้าเกี่ยวกับ	ระบบแสดงผลหน้าเกี่ยวกับซึ่งแสดงข้อมูลผู้พัฒนารวมไปถึงแสดงเครดิต (credit) ไลบรารีต่าง ๆ ที่ใช้งานภายในแอปพลิเคชัน
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประ公示พร้อมทั้งแสดงรายการข่าวสารทั้งหมด
เมนูออกจากระบบ	กดปุ่มเมนูออกจากระบบ	ทำการออก จาก ระบบ และแสดงหน้าจอข่าวสาร

- การทดสอบหน้ารายละเอียดประกาศในการแสดงผลหน้าจอรายละเอียดประกาศนั้น จะประกอบไปด้วยหัวเรื่องประกาศ รายละเอียดประกาศ วันที่ประกาศและเอกสาร แบบ ผลการทดสอบดังตารางที่ 5.10

ตารางที่ 5.10: ผลการทดสอบหน้ารายละเอียดประกาศ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูประกาศ	ระบบแสดงผู้ล หน้าจอประกาศ พร้อม ทั้งแสดงรายการประกาศทั้งหมด
	กดปุ่มอ่านรายละเอียดประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กดปุ่มดาวน์โหลดเอกสารแบบ	ระบบแสดงผลการดาวน์โหลดเอกสารแบบ
	กดปุ่มย้อนกลับ	ระบบแสดงผู้ล หน้าจอประกาศ พร้อม ทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าสนทนา ในการแสดงผลหน้าจอสนทนานั้นจะประกอบไปด้วยราย-การประวัติการสนทนา ซึ่งกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.11

ตารางที่ 5.11: ผลการการทดสอบหน้าสนทนา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาพร้อมทั้งแสดงรายการประวัติการสนทนา
	กดปุ่มที่ซึ่งกรอกข้อความ	ระบบแสดงตัวกระพิรบ (cursor) เพื่อชี้ให้รู้ว่าตำแหน่งของการพิมพ์อักษรจะ
	พิมพ์อักษร	ระบบแสดงผลอักษรที่ถูกพิมพ์
	กดปุ่มส่งข้อความ	ระบบแสดงข้อความที่ถูกพิมพ์บนรายการประวัติสนทนาล่าสุด
	กดปุ่มย้อนกลับ	ระบบแสดงผู้ลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าปฏิทินกำหนดการ ในการแสดงผลหน้าปฏิทินกำหนดการนั้นจะประกอบไปด้วยรายการประวัติการสนทนากับกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.12

ตารางที่ 5.12: ผลการการทดสอบหน้าปฏิทินกำหนดการ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงตารางกำหนดการทั้งหมด
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประวัติพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าดาวน์โหลดเอกสาร ในการแสดงผลหน้าจอดาวน์โหลดเอกสารนั้นจะ ประกอบไปด้วยรายการเอกสารโดยที่แต่ละฉบับจะแสดงชื่อเอกสารและปุ่มดาวน์โหลดเอกสาร ผลการทดสอบดังตารางที่ 5.13

ตารางที่ 5.13: ผลการการทดสอบหน้าดาวน์โหลดเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลด เอกสาร	ระบบ แสดง ผล หน้า จอ ตารา- ง รายการ เอกสาร ใน ระบบ พร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มดาวน์โหลดเอกสาร	ระบบแสดงผลการดาวน์โหลด เอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ราย- การเอกสารในระบบพร้อมทั้ง แสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผู้ หน้า จอ ประ- การ พร้อม ทั้ง แสดง รายการ ข่าวสารทั่วหมด

บทที่ 6

สรุปและข้อเสนอแนะ

การดำเนินโครงการเพื่อพัฒนาระบบ XX นี้ พบร่วมกับระบบสามารถทำงานได้ตามที่วิเคราะห์ และออกแบบไว้ แต่ก็พบปัญหาและอุปสรรคระหว่างการพัฒนา ในบทนี้ผู้พัฒนาจึงขอสรุป ความสามารถของระบบ ซึ่งจะช่วยให้ผู้ใช้งานสามารถนำร่องและประเมินแนวทางในการพัฒนาระบบ XX ต่อ ตามลำดับ

6.1 สรุปความสามารถของระบบ

ระบบ XX ทั้งเว็บแอปพลิเคชันและโมบายแอปพลิเคชันสามารถสรุปความสามารถที่ระบบ ทำได้ดังนี้

6.1.1 เว็บแอปพลิเคชัน

ความสามารถหลักของเว็บแอปพลิเคชันนั้นเน้นสร้างความสะดวกต่อการจัดการเอกสาร เรื่องข้อมูลต่างๆ ที่เกี่ยวข้องกับ XX โดยแบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

(a) เจ้าหน้าที่

- สร้างและแก้ไข XX ได้
- สร้างกำหนดการการดำเนินงานของ XX ได้
- สนทนากับ XX ได้
- อัปโหลดเอกสารที่เกี่ยวข้องเข้าสู่ฐานข้อมูลได้
- สร้างกำหนดการส่งสำเนาเอกสารได้

(b) นักศึกษา

- สมัครสมาชิกได้
- ดูประกาศได้
- ดูปฏิทินกำหนดการได้
- ดูกำหนดการส่งเอกสารได้
- แก้ไขข้อมูลส่วนตัวได้

6.1.2 แอนดรอยด์พลีเคชัน

เพื่อสร้างความสะดวกในการติดต่อและติดตามประกาศเนื่องจากทำงานบนอุปกรณ์พกพา ทั้งนี้ยังมีบางฟังก์ชันที่จำเป็นต้องทำงานบนอุปกรณ์พกพาด้วย เช่น กล้องถ่ายภาพ เป็นต้น

(a) เจ้าหน้าที่

- ดูประกาศได้
- ดูกำหนดการการทำงานของ XX ได้
- สนทนากับ XX ได้
- อัปโหลดเอกสารที่เกี่ยวข้องเข้าสู่ฐานข้อมูลได้

(b) นักศึกษา

- ดูประกาศได้
- ดูปฏิทินกำหนดการได้
- ดูกำหนดการส่งเอกสารได้
- ดาวน์โหลดเอกสารที่เกี่ยวข้องของ XX ได้
- สนทนากับ XX ได้
- ส่งภาพถ่ายสำเนาเอกสารได้
- รับแจ้งเตือนต่างๆ ได้

6.2 ปัญหาและอุปสรรคในการพัฒนา

(a) ไลบรารี (Library) ที่ใช้ในการกรอกข้อมูลลงเอกสาร pdf ไม่รองรับภาษาไทย

แนวทางการแก้ไข : เปลี่ยนขั้นตอนการทำงานเป็นถ่ายภาพเอกสารฉบับจริงแล้วทำการส่งให้เจ้าหน้าที่ตรวจสอบเพื่อยืนยันความถูกต้อง

(b) เนื่องจากทางผู้พัฒนามีความประสงค์ให้ระบบนี้สามารถใช้งานได้จริง ดังนั้น การพัฒนาในตอนนี้ยังมีข้อจำกัดเรื่องขนาดของเอกสารที่จัดเก็บบนไฟร์เบสที่สามารถอัปโหลดเข้าสู่ระบบสูงสุดเพียง 5 GB ซึ่งหากระบบถูกใช้งานจริงจำนวนข้อมูลในระบบจะเกินจำนวนที่ไฟร์เบสให้ใช้งานพรี

แนวทางการแก้ไข : ทำการบีบอัดข้อมูลให้มีขนาดเล็กลง ส่วนในอนาคตอาจจำเป็นต้องศึกษาแนวทางการสร้างเซิฟเวอร์ (Server) เป็นของระบบเอง

6.3 แนวทางการพัฒนาต่อ

- (a) สร้าง Web server ของระบบซึ่งเป็นโปรแกรมที่มีหน้าที่ให้บริการด้านการจัดการ เว็บไซต์และ Database server ซึ่งเป็นโปรแกรมที่ทำหน้าที่ให้บริการด้านการจัด- การดูแลข้อมูลต่าง ๆ ภายในเว็บไซต์ โปรแกรมที่มีการใช้งานส่วนใหญ่เป็น mysql, postgresql, DB2
- (b) การพัฒนาส่วนแจ้งเตือนล่วงหน้าก่อนถึงกำหนดการต่าง ๆ
- (c) การพัฒนาส่วนการกรอกข้อมูลตามแบบฟอร์มของ XX ผ่านระบบได้
- (d) การพัฒนาให้ระบบเป็นระบบ e-document ทั้งระบบโดยเกี่ยวข้องกับผู้ใช้ได้แก่ XX YY และ ZZ

ภาคผนวก

ภาคผนวก ก

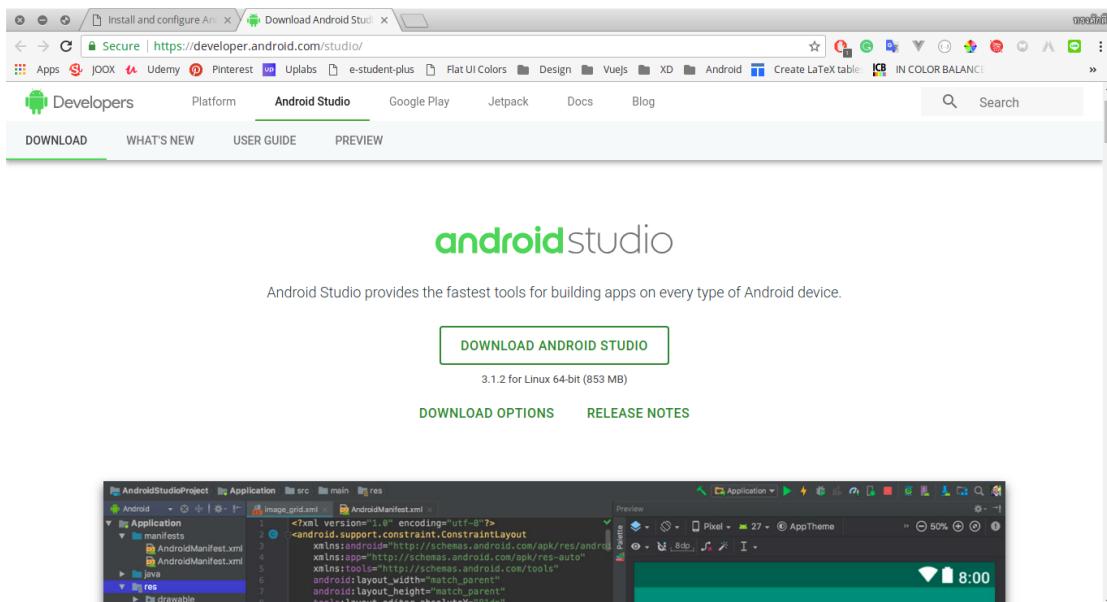
การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้

- การติดตั้ง Android Studio
- การติดตั้ง Node.js
- การติดตั้ง Vue.js Frontend Framework

ก.1 การติดตั้ง Android Studio

- (a) สามารถดาวน์โหลด Android Studio installer package ได้ที่ <https://developer.android.com/studio/> ดังแสดงในรูปที่ ก.1



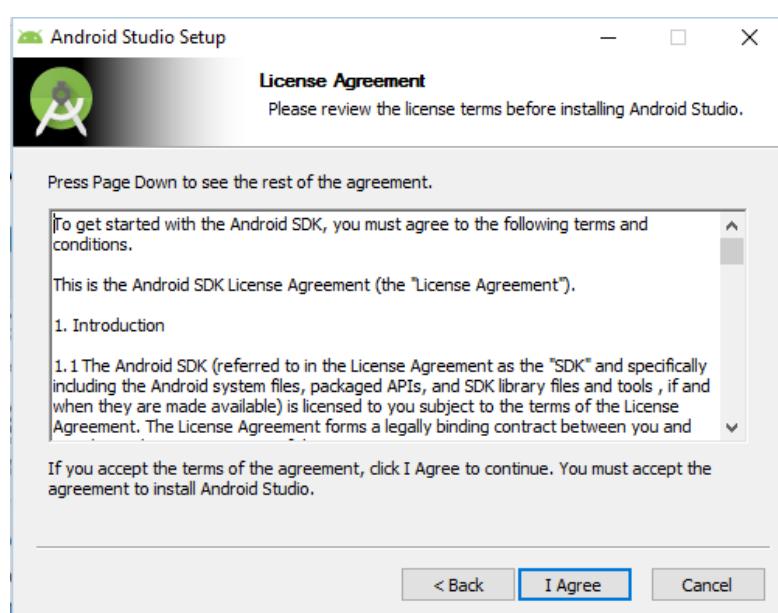
รูปที่ ก.1: หน้าเว็บดาวน์โหลด Android Studio

- (b) แสดงหน้าต่างต้อนรับของ Android Studio ทำการกด Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.2



รูปที่ ก.2: หน้าต่างต้อนรับของ Android Studio

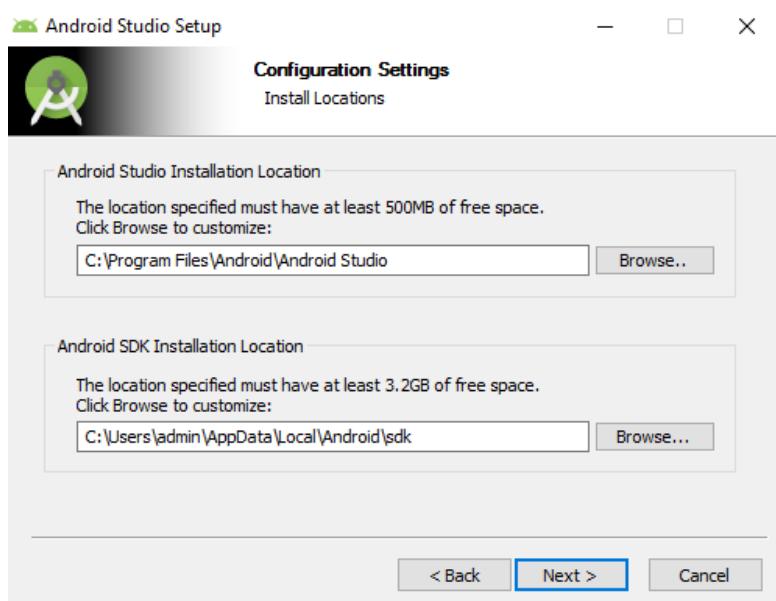
- (c) แสดงหน้าต่างข้อตกลงการใช้งาน Android Studio ทำการกด I Agree ดังแสดงในรูปที่ ก.3



รูปที่ ก.3: หน้าต่างข้อตกลงการใช้งาน Android Studio

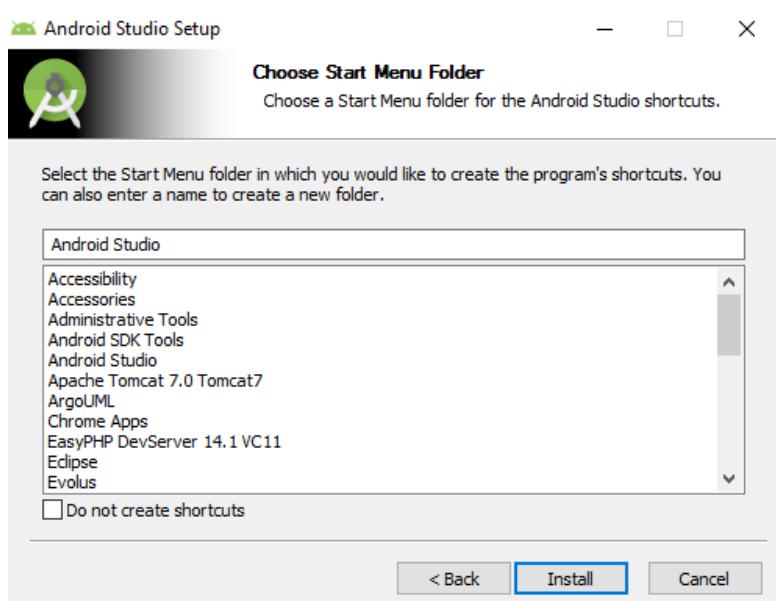
- (d) แสดงหน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Android Studio ทำการกด Next ดังแสดงใน

รูปที่ ก.4



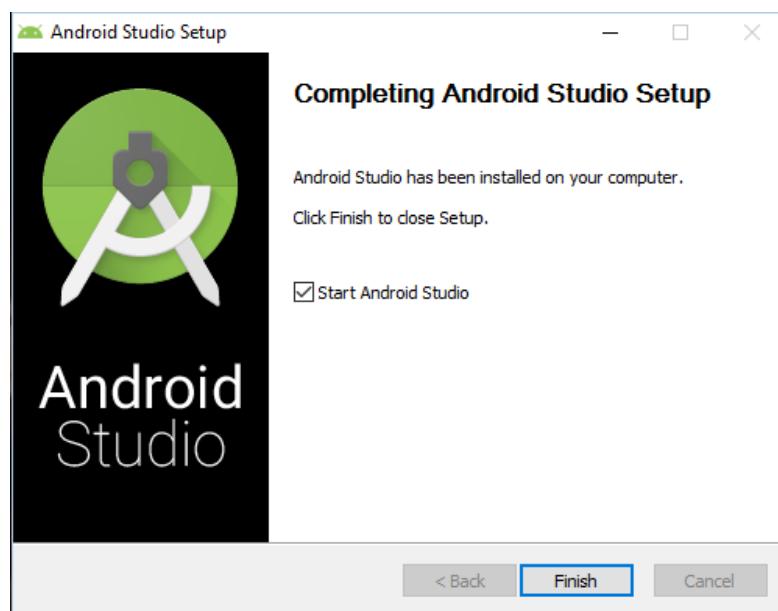
รูปที่ ก.4: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Android Studio

(e) หน้าต่างเริ่มทำการติดตั้งทำการกด Install ดังแสดงในรูปที่ ก.5



รูปที่ ก.5: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Android Studio

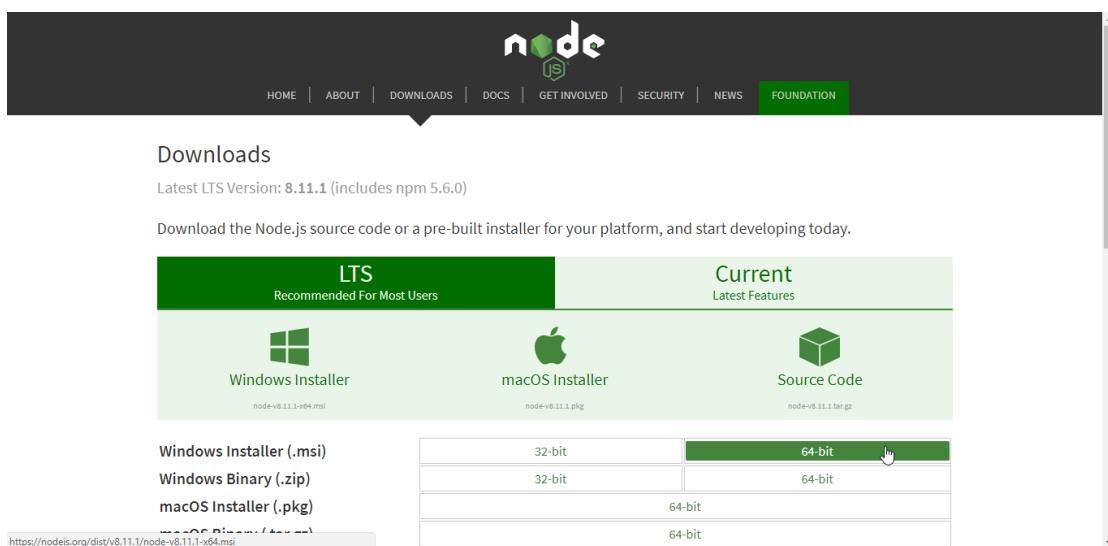
(f) หน้าต่างผลการติดตั้ง Android Studio ดังแสดงในรูปที่ ก.6



รูปที่ ก.6: หน้าต่างผลการติดตั้ง Android Studio

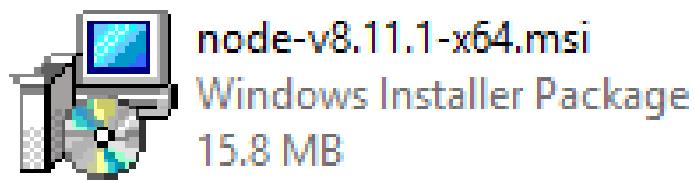
ก.2 การติดตั้ง Node.js

- (a) สามารถดาวน์โหลด Node.js installer package ได้ที่ <https://nodejs.org/en/download/> ดังแสดงในรูปที่ ก.7



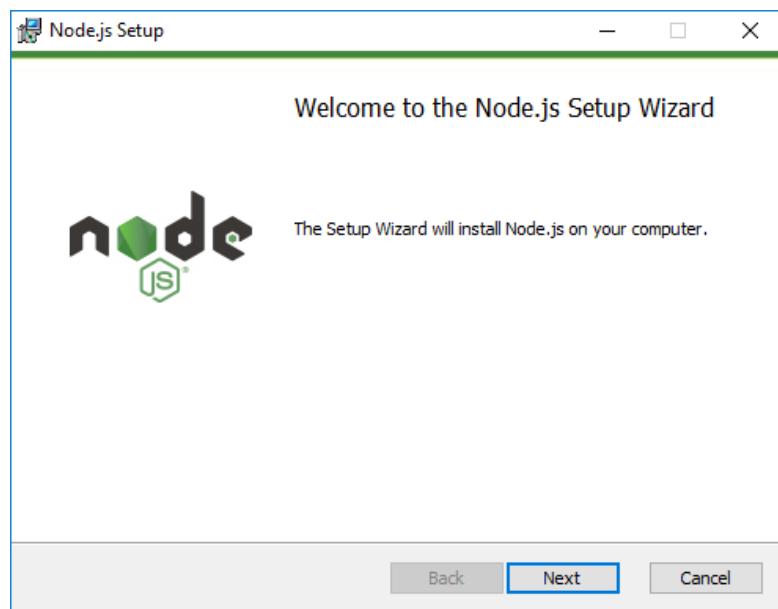
รูปที่ ก.7: หน้าเว็บดาวน์โหลด Node.js

- (b) เปิดไฟล์ติดตั้ง ชื่อ node-vx.xx.x-x64.msi เพื่อทำการติดตั้ง ดังแสดงในรูปที่ ก.8



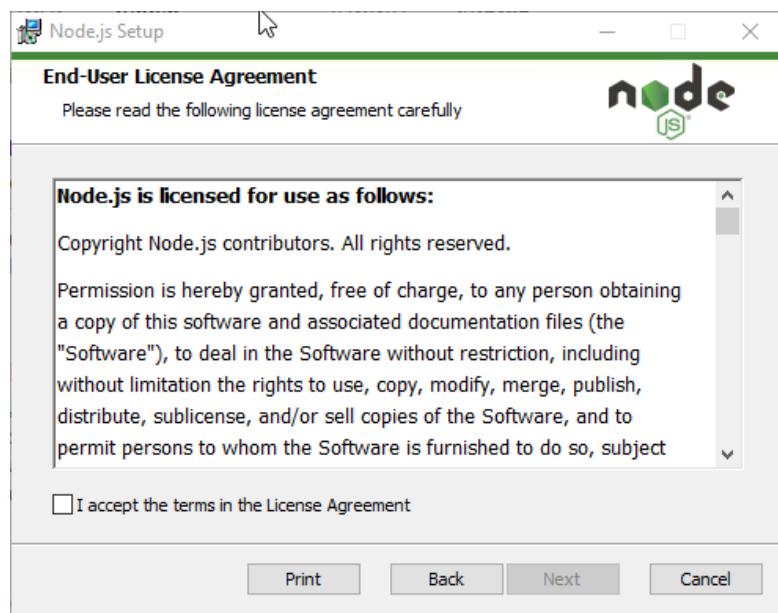
รูปที่ ก.8: ไฟล์ติดตั้งสำหรับติดตั้ง Node.js

- (c) แสดงหน้าต่างต้อนรับของ Node.js ทำการกด Next เพื่อเริ่มกระบวนการติดตั้ง ดัง
แสดงในรูปที่ ก.9



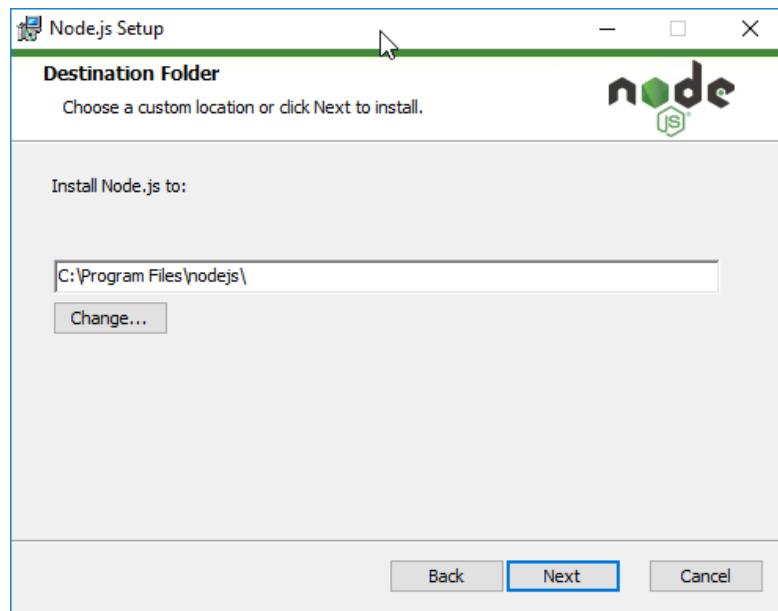
รูปที่ ก.9: หน้าต่างต้อนรับของ Node.js

- (d) แสดงหน้าต่างข้อตกลงในการใช้ Node.js ให้เลือกช่อง I accept the terms in the License Agreement และกด Next ดังแสดงในรูปที่ ก.10



รูปที่ ก.10: หน้าต่างข้อตกลงในการใช้ Node.js

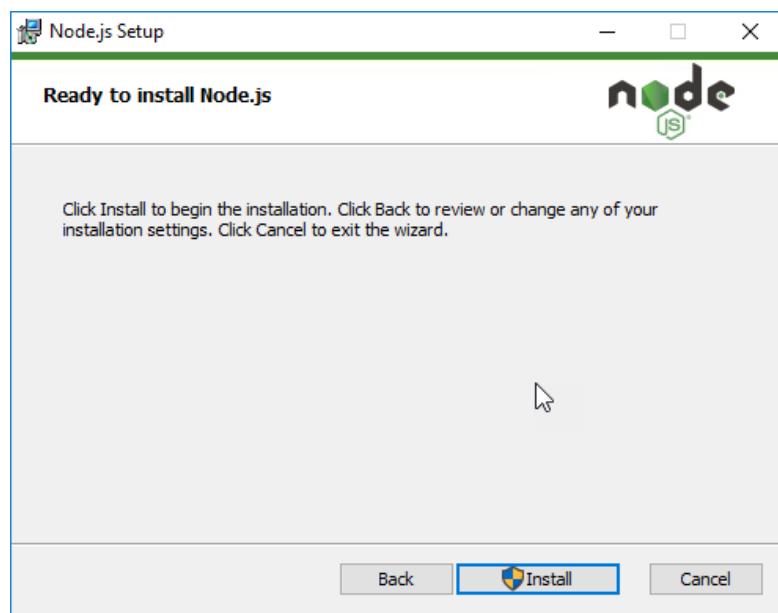
(e) แสดงหน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง ดังแสดงในรูปที่ ก.11



รูปที่ ก.11: หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js

(f) แสดงหน้าต่างสำหรับติดตั้ง Node.js ทำการกด Install เพื่อทำการติดตั้ง ดังแสดงใน

รูปที่ ก.12



รูปที่ ก.12: หน้าต่างติดตั้ง Node.js

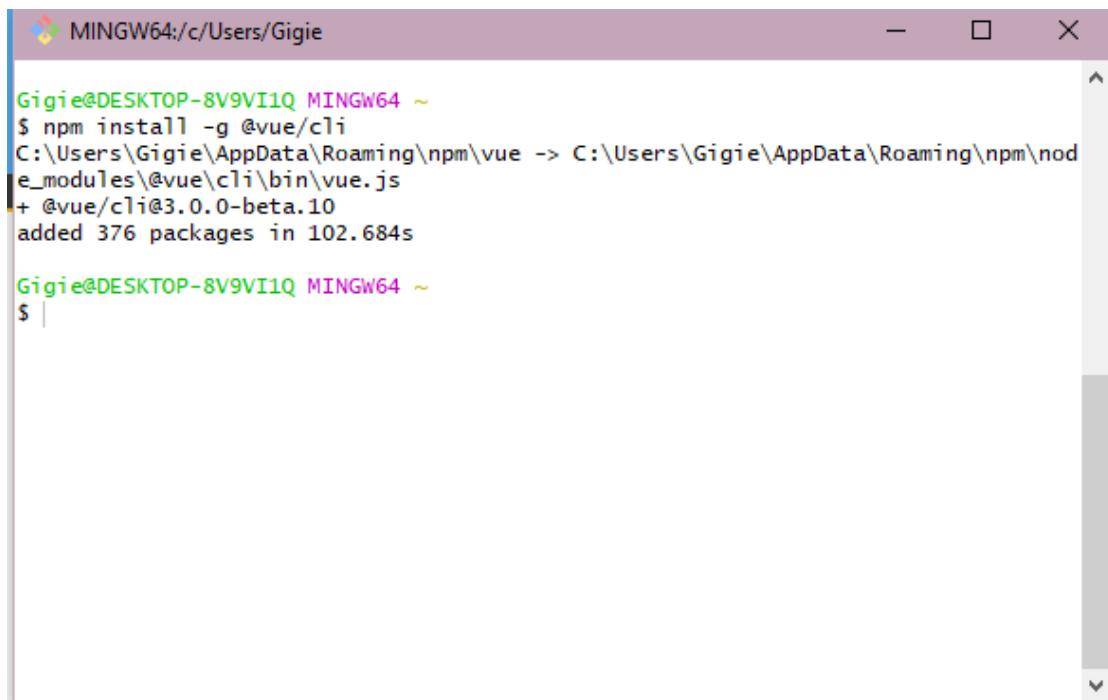
ก.3 การติดตั้ง Vue.js Frontend Framework

การติดตั้ง Vue.js Frontend Framework สามารถทำผ่านคำสั่ง command line ได้โดยจำเป็นต้องทำการติดตั้ง Node.js ก่อนเพื่อใช้ในกระบวนการติดตั้งนี้ ดังแสดงในรูปที่ ก.13

```
npm install -g @vue/cli
```

รูปที่ ก.13: คำสั่งสำหรับติดตั้ง Vue.js Frontend Framework

ผลการติดตั้ง Vue.js Fronted Framework ดังแสดงในรูปที่ ก.14



The screenshot shows a terminal window titled "MINGW64:/c/Users/Gigie". The command entered was "\$ npm install -g @vue/cli". The output indicates that the command was run from C:\Users\Gigie\AppData\Roaming\npm\vue and redirected to C:\Users\Gigie\AppData\Roaming\npm\node_modules\@vue\cli\bin\vue.js. A new version of @vue/cli (3.0.0-beta.10) was added, along with 376 other packages, in 102.684 seconds.

```
Gigie@DESKTOP-8V9VI1Q MINGW64 ~
$ npm install -g @vue/cli
C:\Users\Gigie\AppData\Roaming\npm\vue -> C:\Users\Gigie\AppData\Roaming\npm\node_modules\@vue\cli\bin\vue.js
+ @vue/cli@3.0.0-beta.10
added 376 packages in 102.684s

Gigie@DESKTOP-8V9VI1Q MINGW64 ~
$ |
```

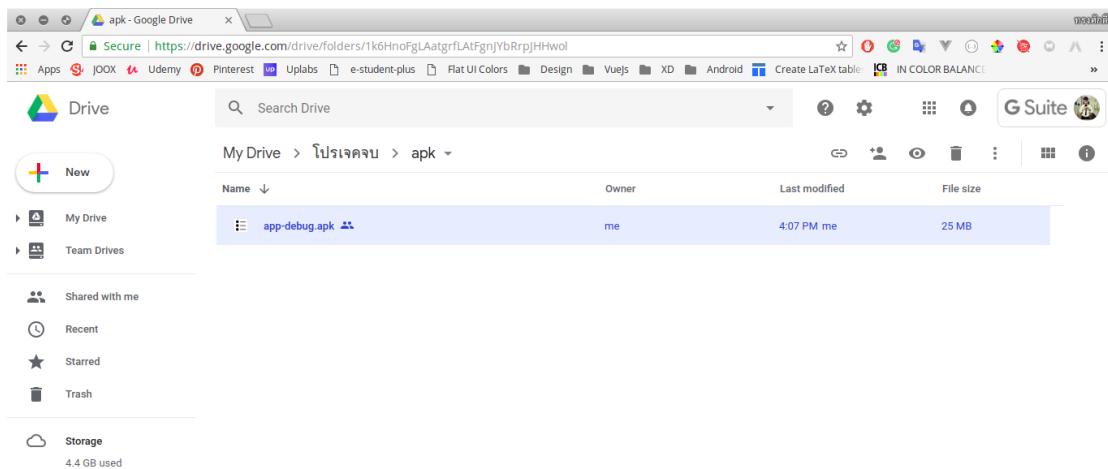
รูปที่ ก.14: หน้าต่างผลการติดตั้ง

ภาคผนวก ข

คู่มือการติดตั้งระบบ

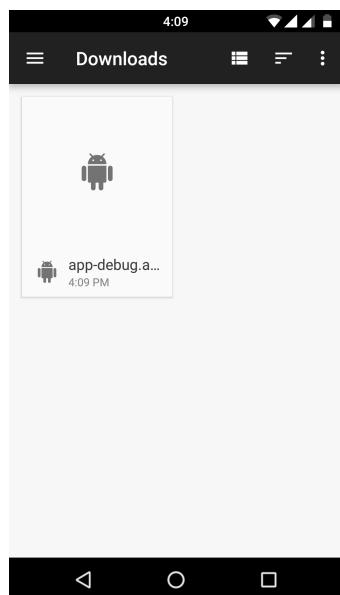
ในการติดตั้งเพื่อใช้งานแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี หรือ ESP สามารถทำได้โดยมีขั้นตอนดังนี้

- (a) สามารถดาวน์โหลด ESP installer package ได้ที่ <https://drive.google.com/drive/folders/1k6HnoFgLAatgrfLAtFgnJYbRrpJHHwol> ดังแสดงในรูปที่ ข.1



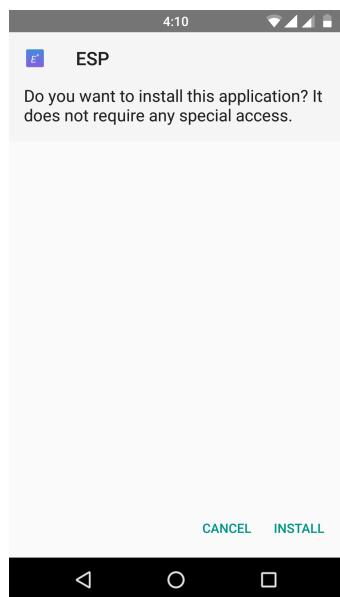
รูปที่ ข.1: หน้าเว็บดาวน์โหลด ESP installer package

- (b) ดัดлокไฟล์ app-debug.apk ที่อยู่ในแฟ้มงาน(Folder)ที่อยู่บนคอมพิวเตอร์ไปไว้ในหน่วยความจำบนอุปกรณ์ที่ต้องการ ดังแสดงในรูปที่ ข.2



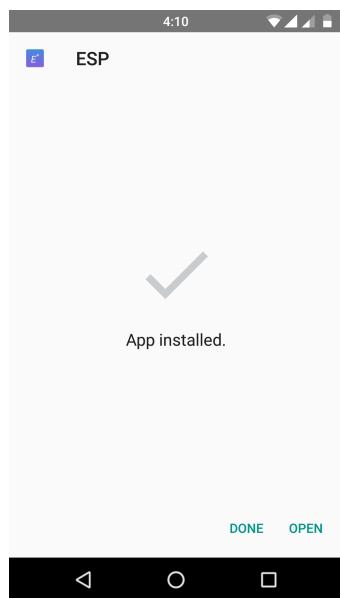
รูปที่ ข.2: ไฟล์ app-debug.apk บนอุปกรณ์

- (c) ทำการเปิดไฟล์ app-debug.apk และกด INSTALL เพื่อทำการติดตั้ง ดังแสดงในรูปที่ ข.3



รูปที่ ข.3: หน้าจอต้อนรับการติดตั้งแอปพลิเคชันบนอุปกรณ์เอนдрอย์

- (d) เมื่อทำการติดตั้งแอปพลิเคชันสำเร็จระบบจะแสดงผล ดังแสดงในรูปที่ ข.4



รูปที่ ข.4: หน้าจอติดตั้งแอปพลิเคชันบนอุปกรณ์แอนดรอย์

ภาคผนวก ค

คู่มือการใช้งานระบบ

คู่มือการใช้งานทั้งหมดของระบบ สามารถแบ่งออกเป็น 2 ส่วน ดังนี้

(a) ส่วนของหน้าเมนูแอปพลิเคชันสำหรับนักศึกษา

- หน้าจอต้อนรับแสดงผลทุกครั้งเมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ ค.8



รูปที่ ค.1: หน้าจอต้อนรับ

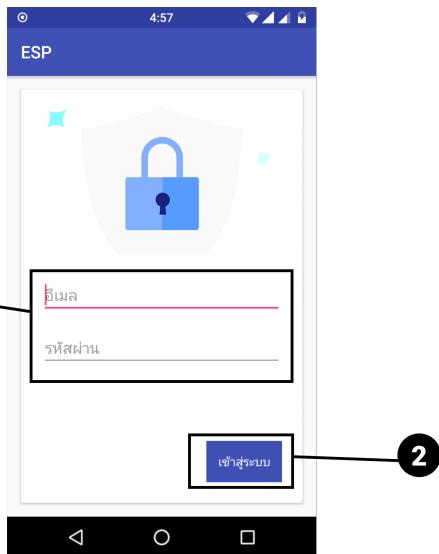
- เมื่อระบบทำการตรวจสอบว่ามีสิทธิ์(Permission)ในการใช้งานแอปพลิเคชันที่ผู้ใช้งานไม่ได่อนุญาตให้เข้าถึง ระบบจะแสดงหน้าต่างขอสิทธิ์การเข้าถึง ดังแสดงในรูปที่ ค.9



รูปที่ ค.2: หน้าต่างขอสิทธิ์การเข้าถึง

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

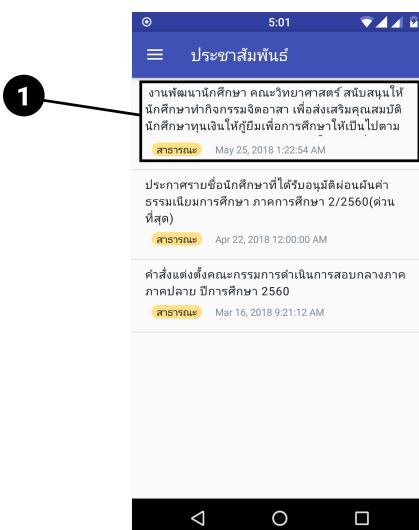
- หมายเลข 1 คือ หน้าต่างขอสิทธิ์การเข้าถึงข้อมูล
- หมายเลข 2 คือ ปุ่มให้สิทธิ์และยกเลิกการให้สิทธิ์
- ระบบทำการตรวจสอบทุกครั้งเมื่อผู้ใช้งานเปิดใช้งานแอปพลิเคชัน หากผู้ใช้งานคนปัจจุบันยังไม่ได้เข้าสู่ระบบ ระบบจะแสดงหน้าจอเข้าสู่ระบบโดยผู้ใช้งานจำเป็นต้องทำการกรอกข้อมูลคือ อีเมลและรหัสผ่าน ดังแสดงในรูปที่ ค.3



รูปที่ ค.3: หน้าจอเข้าสู่ระบบ

จากรูปที่ ค.3 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ส่วนของฟอร์มในการกรอกข้อมูลอีเมลและรหัสผ่าน
- หมายเลข 2 คือ ปุ่มกดเข้าสู่ระบบ
- หน้าแสดงข่าวสารประชาสัมพันธ์ซึ่งเป็นหน้าจอหลักของแอปพลิเคชัน ดังแสดงในรูปที่ ค.4



รูปที่ ค.4: หน้าแสดงข่าวสาร

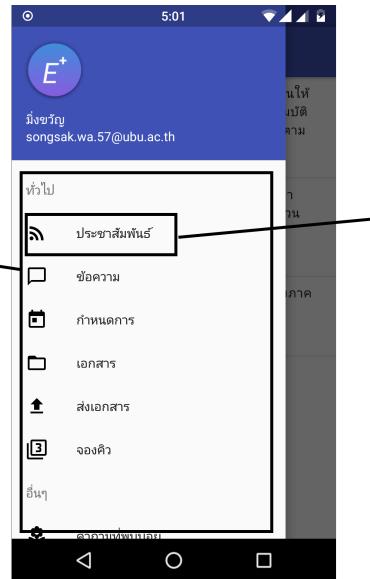
จากรูปที่ ค.4 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขอ 1 คือ ข่าวสารที่มีข้อมูล หัวข้อข่าวสาร แท็ก(Tag)และวันที่ประการข่าวสาร
- เมื่อผู้ใช้กดเลือกดูรายละเอียดของข่าวสาร ระบบจะแสดงหน้าจอรายละเอียดข่าวสาร ดังแสดงในรูปที่ ค.5



รูปที่ ค.5: หน้ารายละเอียดของข่าวสาร

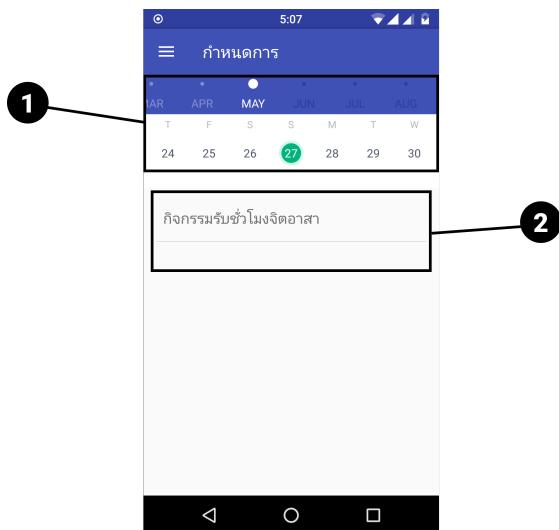
- จากรูปที่ ค.5 สามารถอธิบายการใช้งานได้ดังนี้
- หมายเลขอ 1 คือ หัวข้อข่าวสาร
 - หมายเลขอ 1 คือ รายละเอียดของข่าวสาร
 - เมื่อผู้ใช้กดเลือกเมนู Drawer ระบบจะแสดงเมนูนำทางหลักของแอปพลิเคชัน ซึ่งประกอบไปด้วยเมนู ประชาสัมพันธ์ ข้อความ กำหนดการ เอกสาร ส่งเอกสาร จองคิว คำถามที่พบบ่อย เกี่ยวกับเรา บัญชีผู้ใช้ และออกจากระบบ ดังแสดงในรูปที่ ค.6



รูปที่ ค.6: เมนูนำทางหลักของแอปพลิเคชัน

จากรูปที่ ค.6 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ รายการเมนูนำทางหลักของแอปพลิเคชัน
- หมายเลข 2 คือ เมนูนำทาง
- เป็นผู้ใช้เลือกเมนูกำหนดการ ระบบจะแสดงหน้าจอกำหนดการ ตั้งแสดงในรูปที่ ค.10



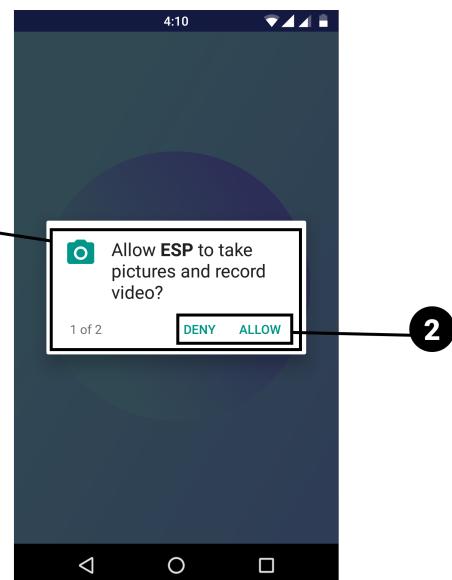
รูปที่ ค.7: หน้าจอกำหนดการ

- ส่วนของหน้าเมนูแอปพลิเคชันสำหรับนักศึกษา
- หน้าจอต้อนรับแสดงผลทุกครั้งเมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ ค.8



รูปที่ ค.8: หน้าจอต้อนรับ

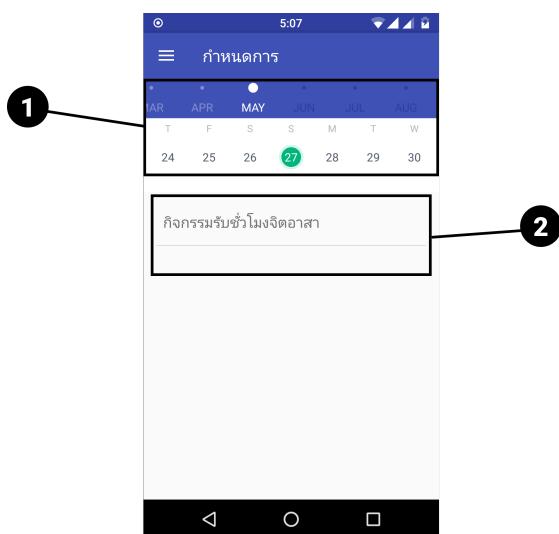
- เมื่อระบบทำการตรวจสอบว่ามีสิทธิ์(Permission)ในการใช้งานแอปพลิเคชันที่ผู้ใช้งานไม่ได่อนุญาตให้เข้าถึง ระบบจะแสดงหน้าต่างขอสิทธิ์การเข้าถึง ดังแสดงในรูปที่ ค.9



รูปที่ ค.9: หน้าต่างขอสิทธิ์การเข้าถึง

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ หน้าต่างขอสิทธิ์การเข้าถึงข้อมูล
- หมายเลข 2 คือ ปุ่มให้สิทธิ์และยกเลิกการให้สิทธิ์



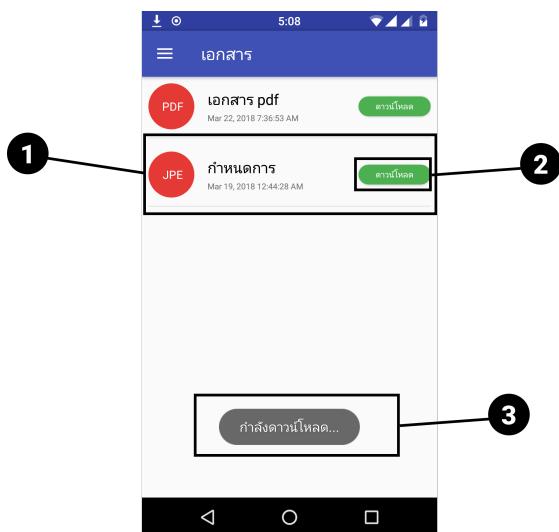
รูปที่ ค.10: หน้าแสดงกำหนดการ

จากรูปที่ ค.10 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปฏิทินแสดงวันที่ปัจจุบันและผู้ใช้สามารถเลือกวันที่ต้อง-

การเพื่อดูกำหนดการก่อนหน้า

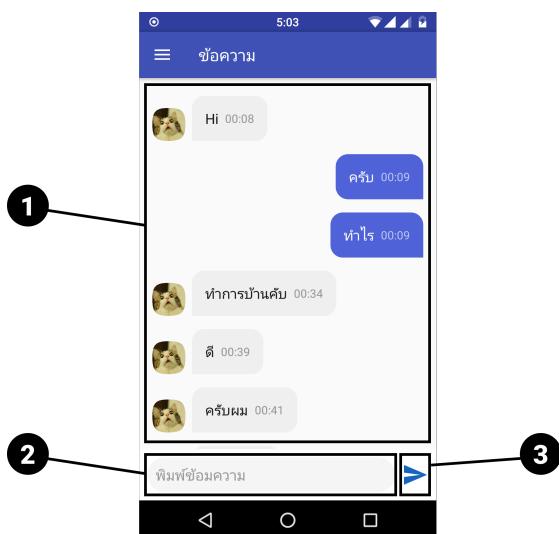
- หมายเลข 2 คือ แสดงรายการกำหนดการของวันนี้ ๆ
- เมื่อผู้ใช้เลือกเมนูเอกสาร ระบบจะแสดงหน้าจอรายการเอกสาร ดังแสดงในรูปที่ ค.11



รูปที่ ค.11: หน้าจอเอกสาร

จากรูปที่ ค.11 สามารถอธิบายการใช้งานได้ดังนี้

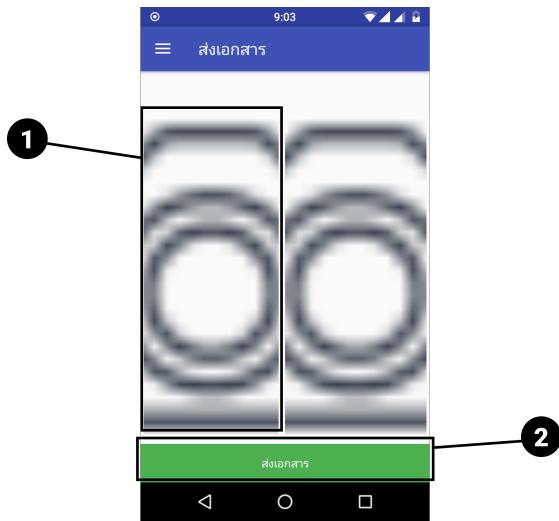
- หมายเลข 1 คือ รายการเอกสาร
- หมายเลข 2 คือ ปุ่มดาวน์โหลดเอกสาร
- หมายเลข 3 คือ สถานะการดาวน์โหลดเอกสาร
- เมื่อผู้ใช้เลือกเมนูข้อความ ระบบจะแสดงหน้าจอสนทนา ดังแสดงในรูปที่ ค.12



รูปที่ ค.12: หน้าจอสนทนากู

จากรูปที่ ค.12 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ รายการประวัติการสนทนากู
- หมายเลข 2 คือ ช่องกรอกข้อความเพื่อสนทนากู
- หมายเลข 3 คือ ปุ่มส่งข้อความ
- เมื่อผู้ใช้เลือกเมนูส่งเอกสารระบบจะตรวจสอบข้อมูลว่าเจ้าหน้าที่ได้ทำการเปิดให้นักศึกษาส่งเอกสารได้หรือไม่ หากตรวจสอบแล้วพบว่าสามารถส่งได้ระบบจะแสดงหน้าจอส่งเอกสาร ดังแสดงในรูปที่ ค.13



รูปที่ ค.13: หน้าจอส่งเอกสาร

จากรูปที่ ค.13 สามารถอธิบายการใช้งานได้ดังนี้

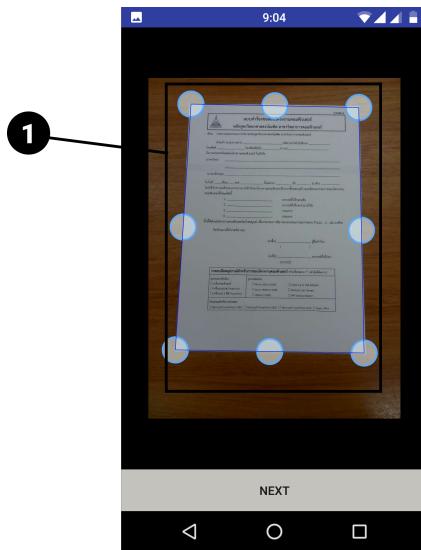
- หมายเลข 1 คือ เมื่อผู้ใช้เข้ามาครั้งแรกเมื่อผู้ใช้กดรูปภาพเอกสารระบบจะนำผู้ใช้ไปยังหน้าจอถ่ายภาพเอกสารและแสดงรูปภาพพรีวิว(Preview)ภาพถ่ายเอกสาร
- หมายเลข 2 คือ ปุ่มกดส่งเอกสาร
- เมื่อผู้ใช้กดที่ปุ่มถ่ายภาพเอกสารระบบจะแสดงหน้าจอถ่ายภาพเอกสาร ตั้งแสดงในรูปที่ ค.14



รูปที่ ค.14: หน้าจอถ่ายภาพเอกสาร

จากรูปที่ ค.14 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มกดถ่ายภาพสำเนาเอกสาร
- หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร ตั้งแสดงในรูปที่ ค.15

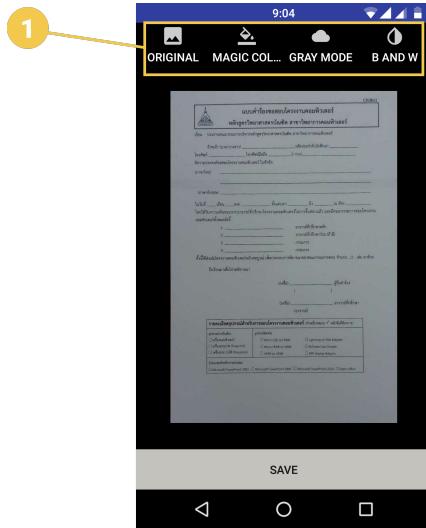


รูปที่ ค.15: หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.15 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มปรับมุมภาพทั้ง 8 มุมเพื่อปรับขนาดภาพถ่ายเอกสาร

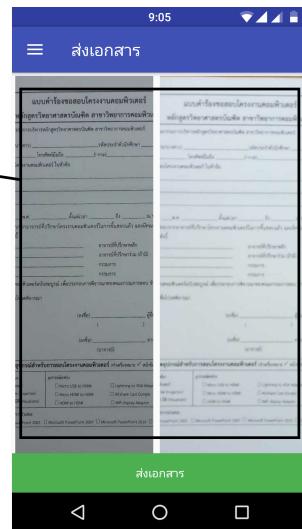
- หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร ดังแสดงในรูปที่ ค.16



รูปที่ ค.16: หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.16 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขอ 1 คือ เมื่อผู้ใช้ต้องการปรับความคมชัดของภาพสามารถทำได้จาก 4 ปุ่ม คือ ปุ่มภาพตันฉบับ ปุ่มปรับสีอัตโนมัติ ปุ่มปรับปรับสีเป็นสีเทา ปุ่มปรับสีเป็นสีขาวดำ
- หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสารทั้งสองฉบับ ดังแสดงในรูปที่ ค.



รูปที่ ค.17: หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.17 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ แสดงภาพพรีวิวภาพถ่ายสำเนาเอกสารทั้งสองฉบับ
- เมื่อผู้ใช้กดปุ่มส่งเอกสาร ระบบจะแสดงหน้าต่างแสดงสถานะการอัพโหลดเอกสาร ดังแสดงในรูปที่ ค.18

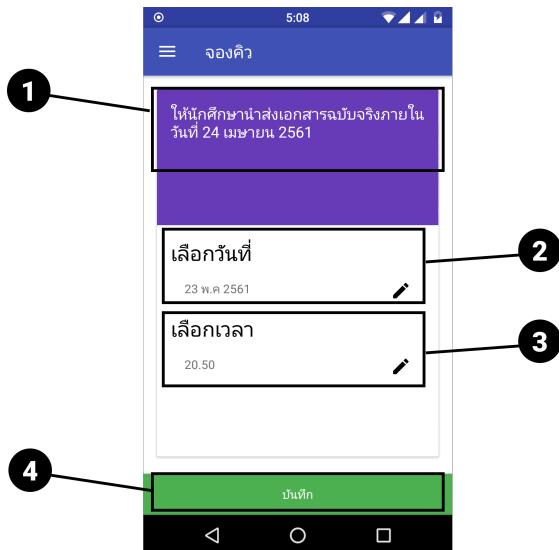


รูปที่ ค.18: หน้าต่างแสดงสถานะการอัพโหลดเอกสาร

จากรูปที่ ค.18 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ หน้าต่างสถานะการอัพโหลดเอกสาร

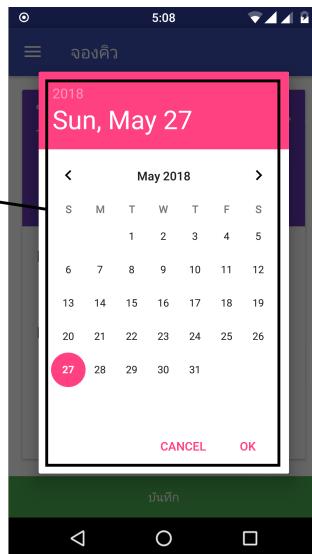
- เมื่อผู้ใช้กดเมนูจองคิว ระบบจะตรวจสอบสถานะการส่งเอกสารว่าถูกเจ้าหน้าที่ตรวจสอบแล้วหรือไม่ ถ้าเจ้าหน้าที่อนุมัติแล้วระบบจะแสดงหน้าจอของวันที่ส่งเอกสาร ดังแสดงในรูปที่ ค.19



รูปที่ ค.19: หน้าจอจองคิว

จากรูปที่ ค.19 สามารถอธิบายการใช้งานได้ดังนี้

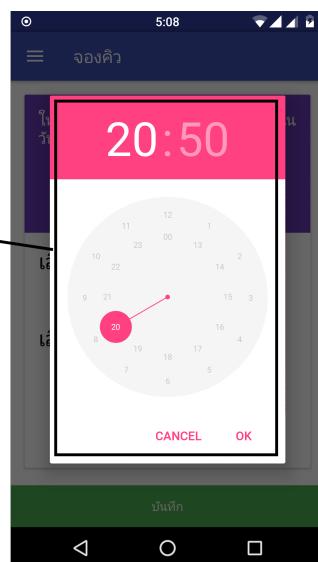
- หมายเลข 1 คือ รายละเอียด
- หมายเลข 2 คือ ส่วนของการเลือกวันที่ที่ต้องการส่งเอกสาร
- หมายเลข 3 คือ ส่วนของการเลือกเวลาที่ต้องการส่งเอกสาร
- หมายเลข 4 คือ ปุ่มกดส่งเอกสาร
- เมื่อผู้ใช้กดปุ่มเลือกวันที่ที่ต้องการส่งเอกสารระบบจะแสดงหน้าต่างเลือกวันที่โดยจะแสดงเฉพาะวันที่ที่เจ้าหน้าที่เลือกไว้เท่านั้น ดังแสดงในรูปที่ ค.20



รูปที่ ค.20: หน้าต่างปฏิทินเลือกวันที่ต้องการส่งเอกสาร

จากรูปที่ ค.20 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ หน้าต่างปฏิทินเลือกวันที่ต้องการส่งเอกสาร
- เมื่อผู้ใช้กดปุ่มเลือกเวลาที่ต้องการส่งเอกสารระบบจะแสดงหน้าต่างเลือกเวลาโดยจะแสดงเฉพาะเวลาที่เจ้าหน้าที่เลือกไว้เท่านั้น ดังแสดงในรูปที่ ค.21



รูปที่ ค.21: หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร

จากรูปที่ ค.21 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขอรบ 1 คือ หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร
- เมื่อผู้ใช้กดเมนูคำถามที่พับบอกระบบจะแสดงหน้าจอคำถามที่พับบอย ดังแสดงในรูปที่ ค.22



รูปที่ ค.22: หน้าจอคำถามที่พับบอย

จากรูปที่ ค.22 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขอรบ 1 คือ คำถาม
- หมายเลขอรบ 2 คือ คำตอบ
- เมื่อผู้ใช้กดปุ่มเมนูเกี่ยวกับระบบ จะแสดงข้อมูลรายละเอียดของงานพัฒนา
นักศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ดังแสดงในรูปที่ ค.23



รูปที่ ค.23: หน้าเกี่ยวกับ

จากรูปที่ ค.23 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ข้อมูลติดต่อของงานพัฒนานักศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี
- ปุ่มกดสำหรับเปิดกลุ่มงานพัฒนานักศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัย อุบลราชธานีบน Facebook

ประวัติผู้เขียน

ชื่อ-สกุล: นายนรากร วิเชียรไชย

รหัสประจำตัวนักศึกษา: 5811403626

วันเกิด: 06 11 2539

ที่อยู่ที่สามารถติดต่อได้: 74 ม.1 ต.ภูดิน อ.เมือง จ.กาฬสินธุ์ 46000

เบอร์โทรศัพท์: (+66) 88 766 5841

อีเมลล์: narakorn.vi.58@ubu.ac.th

ระดับมัธยมต้น: โรงเรียน สหสัขนรศึกษา จังหวัด กาฬสินธุ์

ระดับมัธยมปลาย: โรงเรียน สหสัขนรศึกษา จังหวัด กาฬสินธุ์

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สถาบัน และคอมพิวเตอร์ สาขาวิชาการ คอมพิวเตอร์

คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี