

ชื่อระบบภาษาไทย

English Project Title

นาย(นางสาว)ชื่อ สกุล

โครงการนี้เป็นส่วนหนึ่งของการศึกษา

หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์  
ภาควิชาคณิตศาสตร์ สติทิ และคอมพิวเตอร์ คณะวิทยาศาสตร์  
มหาวิทยาลัยอุบลราชธานี

ปีการศึกษา 2561

ลิขสิทธิ์ของมหาวิทยาลัยอุบลราชธานี

โครงการ : ชื่อระบบภาษาไทย  
English Project Title  
โดย : นาย(นางสาว)ชื่อ สกุล  
อาจารย์ที่ปรึกษา : ดร. ชื่อ สกุล  
ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์  
ปีการศึกษา : 2561

---

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา  
ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงการคอมพิวเตอร์

..... อาจารย์ที่ปรึกษา  
(ดร. ชื่อ สกุล)

..... กรรมการ  
(ผศ. ชื่อ สกุล)

..... กรรมการ  
(อาจารย์ ชื่อ สกุล )

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

วันที่ .... / .... / ....

## กิตติกรรมประกาศ

การพัฒนาโครงงานระบบ XX สำเร็จลุล่วงได้ด้วยความกรุณาและความช่วยเหลือจากหลายๆ ท่าน ข้าพเจ้าขอขอบคุณทุกท่าน ที่มีส่วนร่วมในการพัฒนาโครงงานนี้

ขอขอบพระคุณอาจารย์ที่ปรึกษา ชื่อ สกุล อาจารย์ที่ปรึกษาโครงงานที่ได้แนะนำทฤษฎีและแนวทางในแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนาระบบ อีกครั้งยังคงยินดีตรวจสอบความถูกต้องของการทำงานเป็นระยะ ๆ รวมทั้งสร้างกำลังใจให้ผู้พัฒนาอยู่เสมอ

ขอขอบพระคุณ ชื่อ สกุล เจ้าหน้าที่งาน XX หน่วยงาน ผู้ให้คำปรึกษาเรื่องขั้นตอนการดำเนินงานและเอกสารที่เกี่ยวข้องของระบบ XX

ขอขอบพระคุณเจ้าหน้าที่ภาควิชาคณิตศาสตร์ สถิติและคอมพิวเตอร์ ที่เคยเอื้อประโยชน์ ความสะดวกทั้งเรื่องอุปกรณ์และสถานที่ต่อการปฏิบัติงานของผู้พัฒนา

ขอขอบคุณบริษัท XX จำกัด ที่ให้โอกาสและความรู้ที่สำคัญต้องในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์รวมไปถึงสอนองค์ความรู้ในการวางแผนงานอีกด้วย

ขอกราบขอบพระคุณบิดา มารดา ที่เคยให้กำลังใจ ค่อยให้ความรักและความห่วงใยเสมอมา ตลอดจนเคยช่วยเหลือทุนทรัพย์ทางด้านการศึกษาและอุปกรณ์ในการพัฒนาโครงงาน

นายชื่อ สกุล วันที่ XX เมษายน 62

โครงงาน : ชื่อระบบภาษาไทย  
โดย : นาย(นางสาว)ชื่อ สกุล  
อาจารย์ที่ปรึกษา : ดร. ชื่อ สกุล  
ระดับการศึกษา : วิทยาศาสตรบัณฑิต สาขาวิชาภาษาคอมพิวเตอร์  
ปีการศึกษา : 2561

---

### บทคัดย่อ

ระบบ XX abstract ภาษาไทย

คำสำคัญ: เว็บแอปพลิเคชัน, แอนดรอยด์แอปพลิเคชัน, วิจัยเอกสาร

Topic : English Project Title  
Author : FIRSTNAME LASTNAME  
Advisor : FIRTNAMES LASTNAME, Ph.D.  
Degree : Bachelor of Science (Computer Science)  
Academic Year : 2018

---

## **Abstract**

XX System English abstract

Keywords: Web Application, Android Application, Vue.js

# สารบัญ

	หน้า
กิตติกรรมประกาศ . . . . .	ค
บทคัดย่อภาษาไทย . . . . .	ง
บทคัดย่อภาษาอังกฤษ . . . . .	จ
สารบัญ . . . . .	ฉ
สารบัญตาราง . . . . .	ญ
สารบัญภาพ . . . . .	ภู
 บทที่	
1 บทนำ . . . . .	<b>1</b>
1.1 ที่มาและเหตุผล . . . . .	1
1.2 วัตถุประสงค์ . . . . .	1
1.3 ขอบเขตของโครงการ . . . . .	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ . . . . .	2
1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools) . . . . .	2
1.5.1 ฮาร์ดแวร์ . . . . .	2
1.5.2 ซอฟต์แวร์ (Software) . . . . .	2
1.5.3 แผนการดำเนินการ . . . . .	4
2 ทฤษฎีที่เกี่ยวข้อง . . . . .	<b>5</b>
2.1 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์ . . . . .	5
2.1.1 ประวัติความเป็นมาของระบบปฏิบัติการแอนดรอยด์ . . . . .	6
2.1.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์ . . . . .	7
2.1.3 การจัดการเกี่ยวกับวัสดุจัดการแอคทิวิตี้ของแอปพลิเคชัน . . . . .	9
2.1.4 กระบวนการเริ่มทำงานของแอคทิวิตี้ (Activity) . . . . .	10
2.1.5 ทำความรู้จักกับ Lifecycle Callback . . . . .	10
2.1.6 การกำหนด Launcher Activity ในแอปพลิเคชัน . . . . .	12
2.1.7 การสร้าง Instance ใหม่ . . . . .	13
2.1.8 การสิ้นสุดการทำงานของแอคทิวิตี้ (Destroy Activity) . . . . .	15
2.2 ความรู้พื้นฐาน Java . . . . .	15
2.2.1 ประวัติความเป็นมาของภาษา Java . . . . .	16

2.2.2	Java Compiler . . . . .	16
2.2.3	โครงสร้างของภาษา Java . . . . .	17
2.3	ความรู้พื้นฐาน Java Script . . . . .	18
2.3.1	ลักษณะการทำงานของ JavaScript . . . . .	19
2.3.2	JavaScript กับ HTML . . . . .	20
2.3.3	โครงสร้างภาษา . . . . .	20
2.4	ความรู้พื้นฐาน Vue.js Fronted Framework . . . . .	25
2.4.1	การติดตั้ง . . . . .	25
2.4.2	คุณลักษณะของวิวเจอส . . . . .	26
2.4.3	ข้อดีของ Vue.js . . . . .	27
2.5	ความรู้พื้นฐาน ScanLibrary . . . . .	27
2.5.1	ขั้นตอนการติดตั้งและใช้งาน . . . . .	27
2.5.2	ตัวอย่างการใช้งาน . . . . .	28
2.6	ความรู้พื้นฐานของระบบ XX เดิม . . . . .	30
2.6.1	ความเป็นมา . . . . .	30
2.6.2	วิสัยทัศน์และพันธกิจ . . . . .	30
2.6.3	ขั้นตอนการดำเนินการ XX . . . . .	30
2.7	เอกสารและงานวิจัยที่เกี่ยวข้อง . . . . .	31
2.7.1	เว็บไซต์ XX . . . . .	31
2.7.2	แอปพลิเคชัน XX . . . . .	31
3	การวิเคราะห์และออกแบบระบบ . . . . .	34
3.1	โครงสร้างภาพรวมของระบบ . . . . .	35
3.2	System Requirements . . . . .	36
3.2.1	Functional Requirements . . . . .	36
3.2.2	Non-functional Requirements . . . . .	37
3.3	User Interface Design . . . . .	38
3.4	Use Case Diagram . . . . .	50
3.5	Class Diagram . . . . .	58
3.6	Sequence Diagram . . . . .	68
3.7	โครงสร้างฐานข้อมูลไฟร์เบส(Firebase Database Structure) . . . . .	79

บทที่	หน้า
<b>4 การพัฒนาระบบ . . . . .</b>	<b>92</b>
<b>4.1 การพัฒนาเว็บแอปพลิเคชัน . . . . .</b>	<b>92</b>
<b>4.1.1 การเชื่อมต่อ Cloud Firestore . . . . .</b>	<b>92</b>
<b>4.1.2 โครงสร้างของการสร้างหน้าเข้าสู่ระบบ . . . . .</b>	<b>94</b>
<b>4.1.3 โครงสร้างของการสร้างหน้าข่าวสาร . . . . .</b>	<b>97</b>
<b>4.1.4 โครงสร้างของการสร้างหน้าดูรายละเอียดข่าวสาร . . . . .</b>	<b>99</b>
<b>4.1.5 โครงสร้างของการสร้างหน้าสนทนা . . . . .</b>	<b>101</b>
<b>4.1.6 โครงสร้างของการสร้างหน้าปฏิทินแสดงกำหนดการ . . . . .</b>	<b>104</b>
<b>4.1.7 โครงสร้างของการสร้างหน้าสร้างประกาศ . . . . .</b>	<b>106</b>
<b>4.1.8 โครงสร้างของการสร้างหน้าอัปโหลดเอกสารที่เกี่ยวข้อง . . . . .</b>	<b>109</b>
<b>4.1.9 โครงสร้างของการสร้างหน้าสร้างกำหนดการจองคิวส่งเอกสาร . . . . .</b>	<b>112</b>
<b>4.2 การพัฒนาเว็บแอปพลิเคชันแอนดรอยด์แอปพลิเคชัน . . . . .</b>	<b>113</b>
<b>4.2.1 โครงสร้างของการสร้างหน้า MainActivity . . . . .</b>	<b>114</b>
<b>4.2.2 โครงสร้างของการสร้างหน้า FeedFragment . . . . .</b>	<b>116</b>
<b>4.2.3 โครงสร้างของการสร้างหน้า PostDetailActivity . . . . .</b>	<b>119</b>
<b>4.2.4 โครงสร้างของการสร้างหน้า ChatActivity . . . . .</b>	<b>121</b>
<b>4.2.5 โครงสร้างของการสร้างหน้า SignInActivity . . . . .</b>	<b>124</b>
<b>4.2.6 โครงสร้างของการสร้างหน้า ScheduleFragment . . . . .</b>	<b>126</b>
<b>4.2.7 โครงสร้างของการสร้างหน้า ScheduleFragment . . . . .</b>	<b>128</b>
<b>5 การทดสอบระบบ . . . . .</b>	<b>129</b>
<b>5.1 การทดสอบการใช้งานแอนดรอยด์แอปพลิเคชัน . . . . .</b>	<b>129</b>
<b>5.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน . . . . .</b>	<b>137</b>
<b>6 สรุปและข้อเสนอแนะ . . . . .</b>	<b>143</b>
<b>6.1 สรุปความสามารถของระบบ . . . . .</b>	<b>143</b>
<b>6.1.1 เว็บแอปพลิเคชัน . . . . .</b>	<b>143</b>
<b>6.1.2 แอนดรอยด์พลิเคชัน . . . . .</b>	<b>144</b>
<b>6.2 ปัญหาและอุปสรรคในการพัฒนา . . . . .</b>	<b>144</b>
<b>6.3 แนวทางการพัฒนาต่อ . . . . .</b>	<b>145</b>

บทที่	หน้า
บรรณานุกรม . . . . .	146
ภาคผนวก . . . . .	149
ภาคผนวก ก การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม . . . . .	149
ก.1 การติดตั้ง Android Studio . . . . .	149
ก.2 การติดตั้ง Node.js . . . . .	152
ก.3 การติดตั้ง Vue.js Fronted Framework . . . . .	155
ภาคผนวก ข คู่มือการติดตั้งระบบ . . . . .	157
ภาคผนวก ค คู่มือการใช้งานระบบ . . . . .	160
ประวัติผู้เขียน . . . . .	177

## สารบัญตาราง

ตารางที่	หน้า
1.1 ขั้นตอนการดำเนินงาน . . . . .	4
3.1 สัญลักษณ์ของ Use case Diagram . . . . .	50
3.2 อธิบาย Use Case หน้าที่ของระบบ ในภาพที่ 3.18 . . . . .	52
3.3 อธิบาย Use Case หน้าที่ของระบบ(ต่อ) ในภาพที่ 3.3 . . . . .	53
3.4 Use Case ดูประชาสัมพันธ์ . . . . .	53
3.5 Use Case ดูปฏิทินกำหนดการ . . . . .	53
3.6 Use Case ดาวน์โหลดเอกสาร . . . . .	54
3.7 Use Case ดู FAQs . . . . .	54
3.8 Use Case จองคิวส่งเอกสาร . . . . .	54
3.9 Use Case ส่งเอกสารตรวจสอบ . . . . .	55
3.10 Use Case สนทนा . . . . .	55
3.11 Use Case จัดการกำหนดการส่งเอกสาร . . . . .	55
3.12 Use Case จัดการประชาสัมพันธ์ . . . . .	56
3.13 Use Case จัดการปฏิทินกำหนดการ . . . . .	56
3.14 Use Case จัดการเอกสาร . . . . .	56
3.15 Use Case แจ้งเตือนนักศึกษา . . . . .	57
3.16 สัญลักษณ์ของ Class Diagram . . . . .	58
3.17 อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ . . . . .	64
3.18 อธิบาย Class Diagram ของส่วนของการแสดงข่าวสาร . . . . .	65
3.19 อธิบาย Class Diagram ของส่วนของการแสดงรายการเอกสารในระบบ . . . . .	66
3.20 อธิบาย Class Diagram ของส่วนของการส่งสำเนาเอกสาร . . . . .	67
3.21 อธิบาย Class Diagram ของส่วนของการสนทนा . . . . .	67
3.22 สัญลักษณ์ของ Sequence Diagram . . . . .	68
3.23 สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Firebase . . . . .	79
3.24 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນປະກາສ . . . . .	84
3.25 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນເອກສາຣທີ່ເກີຍວ່າຂອງ . . . . .	85
3.26 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນປະວັດກາຮສນທາ . . . . .	86
3.27 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນກຳທັດກາຣ . . . . .	87
3.28 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນກາຮຍືນສຳນາເອກສາຣເພື່ອຕຽວຈອບຂອງນັກສຶກຫາ . . . . .	88
3.29 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນຂອງນັກສຶກຫາ . . . . .	89
3.30 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນກາຮຈອງຄິວຂອງນັກສຶກຫາ . . . . .	90
3.31 อธิบายໂທນດທີ່ໃຊ້ເກີບຂໍ້ມູນຄຳຄາມທີ່ພບບ່ອຍ . . . . .	91
5.1 ພລກາຣທດສອບເມນຸນໍາທາງ . . . . .	130
5.2 ພລກາຣທດສອບເມນຸນໍາທາງ(ຕ່ອ) . . . . .	131
5.3 ພລກາຣທດສອບໜ້າຮາຍລະເວີຍດປະກາສ . . . . .	132

5.4	ผลการการทดสอบหน้าสนใจ . . . . .	133
5.5	ผลการการทดสอบหน้าปฎิทินกำหนดการ . . . . .	134
5.6	ผลการการทดสอบหน้าดาวน์โหลดเอกสาร . . . . .	135
5.7	ผลการทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร . . . . .	136
5.8	ผลการทดสอบหน้าจองคิวส่งเอกสาร . . . . .	137
5.9	ผลการทดสอบเมนูนำทาง . . . . .	138
5.10	ผลการทดสอบหน้ารายละเอียดประกาศ . . . . .	139
5.11	ผลการการทดสอบหน้าสนใจ . . . . .	140
5.12	ผลการการทดสอบหน้าปฎิทินกำหนดการ . . . . .	141
5.13	ผลการการทดสอบหน้าดาวน์โหลดเอกสาร . . . . .	142

## สารบัญภาพ

รูปที่	หน้า
2.1 ส่วนแบ่งการตลาดระบบปฏิบัติการบนสมาร์ทโฟน . . . . .	6
2.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์ . . . . .	7
2.3 วัสดุจัดของแอคทิฟิตี้บนระบบปฏิบัติการแอนดรอยด์ . . . . .	10
2.4 การกำหนด Launcher Activity ในแอปพลิเคชัน . . . . .	12
2.5 การกำหนด Launcher Activity ในแอปพลิเคชัน . . . . .	12
2.6 การใช้งาน onCreate() . . . . .	14
2.7 การใช้งาน onDestroy() . . . . .	15
2.8 ตัวอย่างโปรแกรมภาษา Java ด้วย minted . . . . .	17
2.9 การประกาศคลาสในภาษา Java . . . . .	17
2.10 การประกาศเมธอด main ในภาษา Java . . . . .	17
2.11 ตัวแปรอาร์เรย์ . . . . .	22
2.12 การเปรียบเทียบว่าจริงหรือเท็จ . . . . .	23
2.13 ตัวดำเนินการเชิงข้อความ . . . . .	24
2.14 Components ของวีเจeos . . . . .	26
2.15 การเรียกใช้งาน . . . . .	27
2.16 การรับค่าจากไลบรารี . . . . .	28
2.17 ถ่ายภาพเอกสาร . . . . .	29
2.18 ปรับแต่งภาพถ่าย . . . . .	29
2.19 ผลการทำงาน . . . . .	30
2.20 หน้าแรกของเว็บไซต์ PSU Studentloan . . . . .	31
2.21 ข่าวสารจากหน่วยงาน . . . . .	33
3.1 System architecture ระบบ XX . . . . .	35
3.2 หน้าจอเข้าสู่ระบบ . . . . .	38
3.3 หน้าจอข่าวสารประชาสัมพันธ์ . . . . .	39
3.4 หน้าจอปฏิทินกำหนดการ . . . . .	39
3.5 หน้าจอ splash screen . . . . .	40
3.6 หน้าจอเข้าสู่ระบบ . . . . .	41
3.7 หน้าจอสมัครสมาชิก . . . . .	42
3.8 หน้าจอข่าวสารและประชาสัมพันธ์ . . . . .	42
3.9 หน้าจอเมนูนำทางหลัก . . . . .	43
3.10 หน้าจอปฏิทินกำหนดการการดำเนินการ . . . . .	44
3.11 หน้าจอสนทนา . . . . .	44
3.12 หน้าจอเอกสารที่เกี่ยวข้อง . . . . .	45
3.13 หน้าจอจองวันที่และเวลาส่งเอกสาร . . . . .	46
3.14 หน้าจอหลัก . . . . .	47

3.15	หน้าจอสนทนা . . . . .	48
3.16	หน้าจออัปโหลดเอกสาร . . . . .	48
3.17	หน้าจอเข้าสู่ระบบ . . . . .	49
3.18	Use Case Diagram ของระบบ XX . . . . .	51
3.19	Class Diagram ของแอปพลิเคชันระบบ XX . . . . .	59
3.20	Class Diagram ของแอปพลิเคชันระบบ XX . . . . .	60
3.21	Class Diagram ของแอปพลิเคชันระบบ XX . . . . .	61
3.22	Class Diagram ของแอปพลิเคชันระบบ XX . . . . .	62
3.23	Class Diagram ของแอปพลิเคชันระบบ XX . . . . .	63
3.24	Sequence Diagram การแสดงข่าวสาร . . . . .	69
3.25	Sequence Diagram การแสดงปฏิทินกำหนดการ . . . . .	71
3.26	Sequence Diagram การแสดงดาวน์โหลดเอกสาร . . . . .	73
3.27	Sequence Diagram การแสดงบทสนทนา . . . . .	75
3.28	Sequence Diagram แสดงส่งเอกสารตรวจสอบ . . . . .	77
3.29	โครงสร้างฐานข้อมูลแบบ Firebase . . . . .	80
3.30	โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ) . . . . .	81
3.31	โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ) . . . . .	82
3.32	โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ) . . . . .	83
3.33	โหนดเก็บข้อมูลประกาศ . . . . .	84
3.34	โหนดเก็บข้อมูลเอกสารที่เกี่ยวข้อง . . . . .	85
3.35	โหนดเก็บข้อมูลประวัติการสนทนา . . . . .	86
3.36	โหนดเก็บข้อมูลกำหนดการ . . . . .	87
3.37	โหนดเก็บข้อมูลการยื่นคำร้องเพื่อตรวจสอบของนักศึกษา . . . . .	88
3.38	โหนดเก็บข้อมูลของนักศึกษา . . . . .	89
3.39	โหนดเก็บข้อมูลการจองคิวของนักศึกษา . . . . .	90
3.40	โหนดเก็บข้อมูลคำถamentที่พบบ่อย . . . . .	91
4.1	ไฟล์ firebaseConfig.js . . . . .	92
4.2	ไฟล์ firebaseInit.js . . . . .	93
4.3	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue . . . . .	94
4.4	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue . . . . .	95
4.5	การสร้างโลจิก (logic) ของหน้าเข้าสู่ระบบ SignIn.vue . . . . .	96
4.6	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าข่าวสาร Home.vue . . . . .	97
4.7	การสร้างโลจิก(logic)ของหน้าข่าวสาร Home.vue . . . . .	98
4.8	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้ารายละเอียดข่าวสาร ViewPost.vue . . . . .	99
4.9	การสร้างโลจิกของหน้าดูรายละเอียดของข่าวสาร ViewPost.vue . . . . .	99
4.10	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสนทนา Message.vue . . . . .	101
4.11	การสร้างโลจิกของหน้าสนทนา Message.vue . . . . .	103
4.12	การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าปฏิทินกำหนดการ Schedule.vue . . . . .	104

4.13	การสร้างລອຈິກຂອງໜ້າປະລຸງທຶນກຳນົດການ Schedule.vue . . . . .	105
4.14	การสร้างໜ້າຈອສ່ວນຕິດຕ່ອງຜູ້ໃຊ້ຂອງໜ້າສ້າງປະກາດ MgPost.vue . . . . .	106
4.15	การสร้างລອຈິກຂອງໜ້າໜ້າສ້າງປະກາດ MgPost.vue . . . . .	108
4.16	การสร้างໜ້າຈອສ່ວນຕິດຕ່ອງຜູ້ໃຊ້ຂອງໜ້າອັພໂຫລດເອກສາທີ່ເກີ່ວຂ້ອງ MgDocument.vue	109
4.17	การสร้างລອຈິກຂອງໜ້າໜ້າອັພໂຫລດເອກສາທີ່ເກີ່ວຂ້ອງ MgDocument.vue . . . . .	110
4.18	การสร้างໜ້າຈອສ່ວນຕິດຕ່ອງຜູ້ໃຊ້ຂອງໜ້າສ້າງກຳນົດກາຮຈອງຄົວສ່າງເອກສາ MgQueue.vue	112
4.19	การสร้างລອຈິກຂອງໜ້າສ້າງກຳນົດກາຮຈອງຄົວສ່າງເອກສາ MgQueue.vue . . . . .	113
4.20	ຕັ້ງແປຣໃນຄລາສ MainActivity . . . . .	114
4.21	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮສ້າງເມຸນນຳທາງໜັກພາຍໃນຄລາສ MainActivity . . . . .	115
4.22	ຕັ້ງແປຣໃນຄລາສ FeedFragment . . . . .	116
4.23	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮສືບຄັນຂໍ້ມູນຈາກ Cloude Firestore ພາຍໃນຄລາສ FeedFragment	117
4.24	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮດັກອືເນຕົມເມື່ອຜູ້ໃຊ້ກົດທີ່ແກ່ປະກາດໃນຄລາສ FeedFragment . . .	118
4.25	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮແສດງຜລຮຽນລະເອີ້ດປະກາດຂອງຄລາສ PostDetailActivity . . .	119
4.26	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮດາວນໂຫລດເອກສາຂອງຄລາສ PostDetailActivity . . . . .	120
4.27	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮແສດງຜລຮຽນລະເອີ້ດປະກາດຂອງຄລາສ ChatActivity . . . . .	121
4.28	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮສືບຄັນປະວັດກາຮສນທາຂອງຄລາສ ChatActivity . . . . .	122
4.29	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮສັງຂໍ້ຄວາມຂອງຄລາສ ChatActivity . . . . .	123
4.30	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮແສດງຜລໜ້າເຂົ້າສູ່ຮະບບຂອງຄລາສ SignInActivity . . . . .	124
4.31	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮເຂົ້າສູ່ຮະບບຂອງຄລາສ SignInActivity . . . . .	125
4.32	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮແສດງຜລໜ້າປະລຸງທຶນຂອງຄລາສ ScheduleFragment . . . . .	126
4.33	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮສືບຄັນຂໍ້ມູນກຳນົດກາຮຂອງຄລາສ ScheduleFragment . . . . .	127
4.34	ໂຄີດສ່ວນທີ່ໃຊ້ໃນກາຮແສດງຜລໜ້າປະລຸງທຶນຂອງຄລາສ ScheduleFragment . . . . .	128
ກ.1	ໜ້າເວັບດາວນໂຫລດ Android Studio . . . . .	149
ກ.2	ໜ້າຕ່າງຕ້ອນຮັບຂອງ Android Studio . . . . .	150
ກ.3	ໜ້າຕ່າງຂໍອຕກລາງກາຮໃຊ້ຈານ Android Studio . . . . .	150
ກ.4	ໜ້າຕ່າງທີ່ຈັດເກີບໄຟລ໌ຕ່າງໆ ຂອງ Android Studio . . . . .	151
ກ.5	ໜ້າຕ່າງທີ່ຈັດເກີບໄຟລ໌ຕ່າງໆ ຂອງ Android Studio . . . . .	151
ກ.6	ໜ້າຕ່າງຜລກາຮຕິດຕັ້ງ Android Studio . . . . .	152
ກ.7	ໜ້າເວັບດາວນໂຫລດ Node.js . . . . .	152
ກ.8	ໄຟລ໌ຕິດຕັ້ງສໍາຫຼັບຕິດຕັ້ງ Node.js . . . . .	153
ກ.9	ໜ້າຕ່າງຕ້ອນຮັບຂອງ Node.js . . . . .	153
ກ.10	ໜ້າຕ່າງຂໍອຕກລາງໃນກາຮໃຊ້ Node.js . . . . .	154
ກ.11	ໜ້າຕ່າງເລືອກໄຟລ໌ເດືອກທີ່ຈະກຳນົດຕັ້ງ Node.js . . . . .	154
ກ.12	ໜ້າຕ່າງຕິດຕັ້ງ Node.js . . . . .	155
ກ.13	ຄໍາສັ່ງສໍາຫຼັບຕິດຕັ້ງ Vue.js Fronted Framework . . . . .	155
ກ.14	ໜ້າຕ່າງຜລກາຮຕິດຕັ້ງ . . . . .	156
ຂ.1	ໜ້າເວັບດາວນໂຫລດ ESP installer package . . . . .	157
ຂ.2	ໄຟລ໌ app-debug.apk ບນອຸປະກນົນ . . . . .	158

ข.3	หน้าจอต้อนรับการติดตั้งแอปพลิเคชันบนอุปกรณ์แอนดรอยด์ . . . . .	158
ข.4	หน้าจอต้อนรับการติดตั้งแอปพลิเคชันบนอุปกรณ์แอนดรอยด์ . . . . .	159
ค.1	หน้าจอต้อนรับ . . . . .	160
ค.2	หน้าต่างขอสิทธิ์การเข้าถึง . . . . .	161
ค.3	หน้าจอเข้าสู่ระบบ . . . . .	162
ค.4	หน้าแสดงข่าวสาร . . . . .	162
ค.5	หน้ารายละเอียดของข่าวสาร . . . . .	163
ค.6	เมนูนำทางหลักของแอปพลิเคชัน . . . . .	164
ค.7	หน้าจอกำหนดการ . . . . .	164
ค.8	หน้าจอต้อนรับ . . . . .	165
ค.9	หน้าต่างขอสิทธิ์การเข้าถึง . . . . .	166
ค.10	หน้าแสดงกำหนดการ . . . . .	166
ค.11	หน้าจอเอกสาร . . . . .	167
ค.12	หน้าจอสนทนา . . . . .	168
ค.13	หน้าจอส่งเอกสาร . . . . .	169
ค.14	หน้าจอถ่ายภาพเอกสาร . . . . .	170
ค.15	หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร . . . . .	170
ค.16	หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร . . . . .	171
ค.17	หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร . . . . .	171
ค.18	หน้าต่างแสดงสถานะการอัปโหลดเอกสาร . . . . .	172
ค.19	หน้าจอจองคิว . . . . .	173
ค.20	หน้าต่างปฏิทินเลือกวันที่ต้องการส่งเอกสาร . . . . .	174
ค.21	หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร . . . . .	174
ค.22	หน้าจอคำмарที่พับบ่อย . . . . .	175
ค.23	หน้าเกี่ยวกับ . . . . .	176

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและเหตุผล

เนื่องจากกระบวนการทำงานเดิม มีข้อจำกัดดังนี้ XX และในปัจจุบัน เทคโนโลยี XX สามารถทำให้ดีขึ้น ดังนี้ XX

ดังนั้น ผู้พัฒนาจึงเสนอระบบงาน XX โดยใช้เทคโนโลยี XX เพื่อแก้ปัญหา โดย มีความสามารถดังต่อไปนี้ XX

#### 1.2 วัตถุประสงค์

- เพื่อพัฒนาระบบแอปพลิเคชัน
- เพื่ออำนวยความสะดวกสิ่งอำนวยความสะดวกความสะดวกสำหรับ

#### 1.3 ขอบเขตของโครงการ

##### 1.3.1 เจ้าหน้าที่

- สามารถจัดการ X
- สามารถเพิ่ม X
- สามารถตรวจสอบอนุมัติ X

##### 1.3.2 นักศึกษา

- สามารถดู X ได้
- สามารถกำหนด X ได้
- สามารถดาวน์โหลด X ได้
- สามารถจอง X ได้
- สามารถส่ง X ได้
- สามารถสนทนากับ X ผ่านระบบได้

## 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยอำนวยความสะดวก XX
2. ช่วยกระจายงานของ XX
3. ช่วยจัดระบบ XX

## 1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)

### 1.5.1 ฮาร์ดแวร์

#### 1. สมาร์ทโฟน (Smart phone)

- ทำงานบนระบบปฏิบัติการแอนดรอยด์เวอร์ชัน 5.0 หรือ API Level 21
- หน่วยประมวลผลกลาง Mediatek MT6753 Octa-core ความเร็ว 1.3 กิกะเฮิร์ตซ์ (Gigahertz, GHz)
- หน่วยประมวลผลกราฟฟิกอย่างน้อย Mali-T720MP3
- หน่วยความจำหลักอย่างน้อย 2 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรองอย่างน้อย 16 กิกะไบต์ (Gigabyte, GB)
- หน้าจอแสดงผลความละเอียดอย่างน้อย 1080 x 1920 พิกเซล (Pixel)
- หน้าจอแสดงผลขนาดอย่างน้อย 5 นิ้ว
- กล้องถ่ายรูปความละเอียดอย่างน้อย 13 เมกกะพิกเซล (Megapixel)

#### 2. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer)

- ทำงานบนระบบปฏิบัติการ Elementary os พื้นฐานการทำงานบน Linux
- หน่วยประมวลผลกลาง Intel Core i3-3217U ความเร็ว 1.80 กิกะเฮิร์ตซ์ (Gigahertz, GHz)
- หน่วยประมวลผลกราฟฟิก NVIDIA GeForce GT 720M ความจำ 2 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำหลัก 4 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรอง 120 กิกะไบต์ (Gigabyte, GB)

### 1.5.2 ซอฟต์แวร์ (Software)

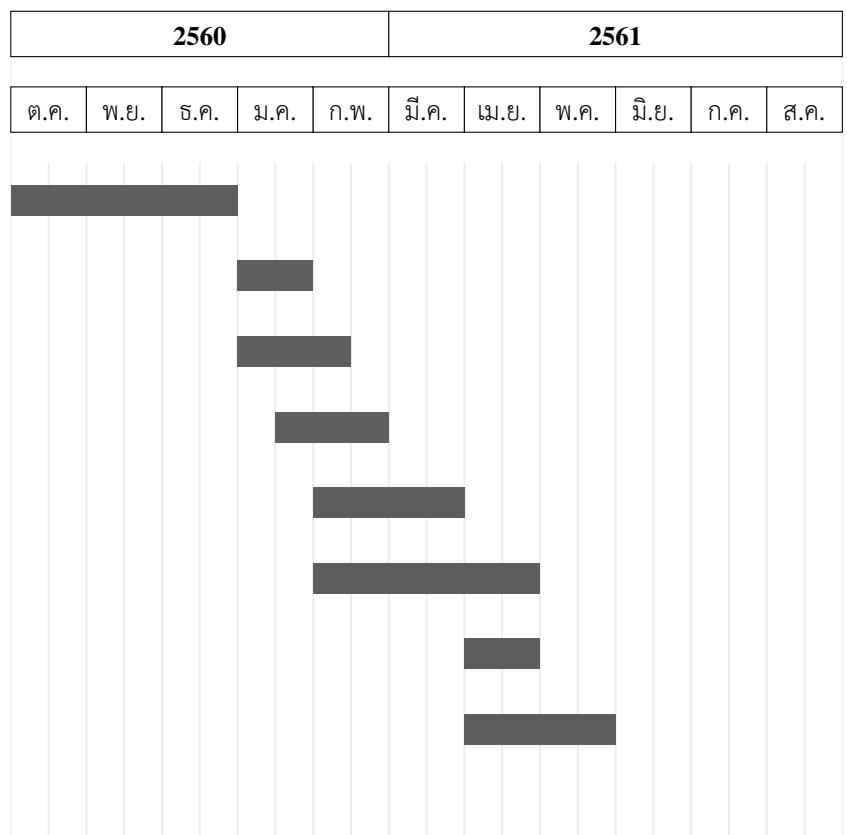
#### 1. XX.js ซึ่งเป็น Frontend Framework สำหรับพัฒนาเว็บไซต์

2. Node.js คือ Cross Platform Runtime Environment หรือเรียกอีกอย่างว่า Backend Framework ใช้สำหรับเป็นเว็บเซิฟเวอร์ (Web Server) ซึ่งเขียนด้วยภาษา JavaScript
3. Android Studio เป็น IDE (Integrated Development Environment) ใช้พัฒนาแอปพลิเคชันสำหรับระบบปฏิบัติการแอนดรอยด์
4. Android SDK ชุดของเครื่องมือที่ใช้ในการพัฒนาโปรแกรมสำหรับระบบปฏิบัติการแอนดรอยด์
5. Visual Studio Code เครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชัน

### 1.5.3 แผนการดำเนินการ

ในการสร้างระบบ XX ผู้พัฒนาได้แบ่งขั้นตอนการดำเนินงานไว้ด้วยกัน 7 ขั้นตอน ดังต่อไปนี้

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน



## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

บทนี้จะเป็นรายละเอียดเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวของกับการพัฒนาโปรแกรมในครั้งนี้ ประกอบไปด้วยหัวข้อหลักทั้งหมด 6 ข้อดังนี้

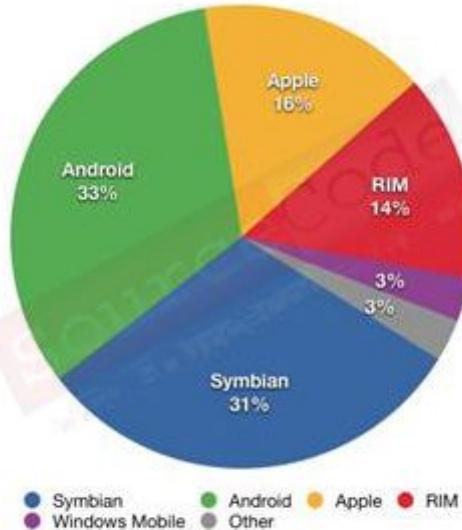
- 2.1 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์
- 2.2 ความรู้พื้นฐานภาษา Java
- 2.3 ความรู้พื้นฐาน Java Script
- 2.4 ความรู้พื้นฐาน Vue.js Fronted Framework
- 2.5 ความรู้พื้นฐานของระบบ XX เดิม
- 2.6 เอกสารและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ความรู้พื้นฐานระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ [1] คือระบบปฏิบัติการแบบเปิดเผยแพร่ต้นฉบับ (Open Source) โดยบริษัท กูเกิล (Google Inc.) ที่ได้รับความนิยมเป็นอย่างสูง เนื่องจากอุปกรณ์ที่ใช้ระบบปฏิบัติการแอนดรอยด์ มีจำนวนมาก อุปกรณ์มีหลากหลายระดับ หลายราคา รวมทั้งสามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอ และความละเอียดแตกต่างกันได้ ทำให้ผู้บริโภคสามารถเลือกได้ตามต้องการ และหากมองในทิศทางสำหรับนักพัฒนาโปรแกรม (Programmer) แล้วนักพัฒนาโปรแกรมเพื่อใช้งานบนระบบปฏิบัติการแอนดรอยด์ ไม่ใช่เรื่องยาก เพราะมีข้อมูลในการพัฒนาร่วมทั้ง 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 เปรียบได้ กับสภาพแวดล้อมที่โปรแกรมทำงานอยู่)

นอกจากนี้แล้วแอนดรอยด์ยังมีโปรแกรมแกรมที่เปิดเผยแพร่ต้นฉบับ (Open Source)

เป็นจำนวนมาก ทำให้เกิดพัฒนาที่สันใจสามารถนำชอร์ฟแวร์ต้นฉบับมาศึกษาได้ประกอบกับความนิยมของแอนดรอยด์ได้เพิ่มขึ้นอย่างมากในปัจจุบัน โดยดูได้จากส่วนแบ่งการตลาด ดังรูปที่ 2.1



รูปที่ 2.1: ส่วนแบ่งการตลาดระบบปฏิบัติการบนสมาร์ทโฟน

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

### 2.1.1 ประวัติความเป็นมาของระบบปฏิบัติการแอนดรอยด์

เริ่มต้นระบบปฏิบัติการแอนดรอยด์ถูกพัฒนามาจากบริษัทแอนดรอยด์ (Android Inc.) เมื่อปี พ.ศ 2546 โดยมีนาย แอนดี้ รูบิน (Andy Rubin) ผู้ให้กำเนิดระบบปฏิบัติการนี้และถูกบริษัทกูเกิลเข้าซื้อกิจการเมื่อเดือนสิงหาคม ปี พ.ศ 2548 โดยบริษัทแอนดรอยด์ได้ถูกซื้อมาเป็นบริษัทลูกของบริษัทกูเกิล

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

ต่อมาเมื่อเดือน พฤษภาคม ปี พ.ศ 2550 บริษัทกูเกิล ได้ทำการก่อตั้งสมาคม OHA (Open Handset Alliance) เพื่อเป็นหน่วยงานกลางในการกำหนดมาตรฐานกลาง ของอุปกรณ์พกพาและ

ระบบปฏิบัติการแอนดรอยด์ โดยมีสมาชิกในช่วงก่อนตั้งจำนวน 34 รายเข้าร่วม ซึ่งประกอบไปด้วย บริษัทชั้นนำที่ดำเนินธุรกิจด้านการสื่อสาร เช่น โรงงานผลิตอุปกรณ์พกพา บริษัทพัฒนาโปรแกรม ผู้ให้บริการสื่อสาร และผู้ผลิตอะไหล่อุปกรณ์ด้านสื่อสาร [2]

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

ช่วงต่อมาได้มีการออกผลิตภัณฑ์จากบริษัทต่าง ๆ ออกแบบหลากหลายรุ่น หลากหลายยี่ห้อ ตามการพัฒนาระบบปฏิบัติการแอนดรอยด์ ที่มีอยู่อย่างต่อเนื่อง ทำให้สินค้าของแอนดรอยด์ มีให้เลือกอยู่อย่างมากมาย

### 2.1.2 โครงสร้างของระบบปฏิบัติการแอนดรอยด์

การทำงานของระบบปฏิบัติการแอนดรอยด์ [3] ถือว่าเป็นสิ่งสำคัญ เพราะถ้าหากพัฒนาโปรแกรม สามารถมองภาพโดยรวมของระบบได้ทั้งหมด จะสามารถเข้าใจถึงกระบวนการทำงานได้ดียิ่งขึ้น และสามารถนำไปช่วยในการออกแบบโปรแกรมที่ต้องการพัฒนาเพื่อให้เกิดประสิทธิภาพในการทำงาน



รูปที่ 2.2: โครงสร้างของระบบปฏิบัติการแอนดรอยด์

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

จากโครงสร้างของระบบปฏิบัติการแอนดรอยด์ในรูปที่ 2.2 จะสังเกตได้ว่า มีการแบ่งออกเป็นส่วน 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++ โดยแบ่งชุดคำสั่งออกเป็นกลุ่มตาม

วัตถุประสงค์ของการใช้งาน เช่น Surface Manager จัดการเกี่ยวกับการแสดงผล Media Framework จัดการเกี่ยวกับการการแสดงภาพและเสียง OpenGL 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 ที่เป็นส่วนรวมคำสั่งและชุดคำสั่งสำคัญโดยถูกเขียนด้วยภาษาจาวา (Java Language)
5. Linux Kernel เป็นส่วนที่ทำหน้าที่หัวใจสำคัญในจัดการกับบริการหลักของระบบปฏิบัติการ เช่น เรื่องหน่วยความจำ พลังงาน ติดต่อกับอุปกรณ์ต่างๆ ความปลอดภัย เครือข่าย โดยแอนดรอยด์ได้นำเอาส่วนนี้มาจากระบบปฏิบัติการลินุกซ์ รุ่น 2.6 (Linux 2.6. Kernel) ซึ่งได้มีการออกแบบมาเป็นอย่างดี

### 2.1.3 การจัดการเกี่ยวกับวัฏจักรแอคทิวิตี้ของแอปพลิเคชัน

ขณะที่ผู้ใช้เปิดใช้งานแอปพลิเคชัน -> ออกจากแอปพลิเคชัน -> แล้วกลับเข้ามาในแอปพลิเคชันอีกครั้งแล้วทิวิตี้จะมีการร่าย Method ต่างๆ เกิดขึ้นในวัฏจักรแอคทิวิตี้ ยกตัวอย่างเช่น เมื่อแอคทิวิตี้เริ่มทำงานครั้งแรกจะแสดงขึ้นมาอยู่ด้านบนสุดของระบบ (Foreground) และรอรับการทำงานจากผู้ใช้ระหว่างกระบวนการนี้ระบบจะมีการเรียกใช้งาน Callback Method หรือ Method ที่ถูกเรียกใช้งานอัตโนมัติในแอคทิวิตี้ที่ได้กำหนดการทำงานให้กับ UI และส่วนติดต่ออื่น ๆ ไว้ ถ้าผู้ใช้มีการใช้งานใด ๆ ที่เป็นการเรียกแอคทิวิตี้อื่นขึ้นมาหรือสลับไปใช้งานแอปพลิเคชันอื่นระบบจะเรียก Callback Method อีกอันขึ้นมา เช่น ช่องแอปพลิเคชันไว้ด้านหลัง Background (ไม่แสดงแอคทิวิตี้แต่ Instance และ Method นั้นยังทำงานอยู่)

ภายใน Callback Method สามารถกำหนดการทำงานในแอคทิวิตี้เมื่อผู้ใช้ออกจากแอปพลิเคชันและกลับเข้ามาใช้งานแอปพลิเคชันใหม่อีกครั้งได้ ตัวอย่าง ถ้าแอปพลิเคชันเป็นแอปพลิเคชัน Streaming Video จะจะสั่งให้ทำการหยุด Video ช่วงระหว่าง และปิดการเชื่อมต่อ Network ไว้ก่อน เมื่อผู้ใช้สลับไปใช้แอปพลิเคชันอื่น และทันทีที่ผู้ใช้กลับมาใช้งานแอปพลิเคชันต่อ ก็ให้ทำการเชื่อมต่อ กับ Network และก้อนญาตให้ผู้ใช้กลับไปเล่น Video ในตำแหน่งที่ค้างต่อไปทันทีโดยที่ไม่

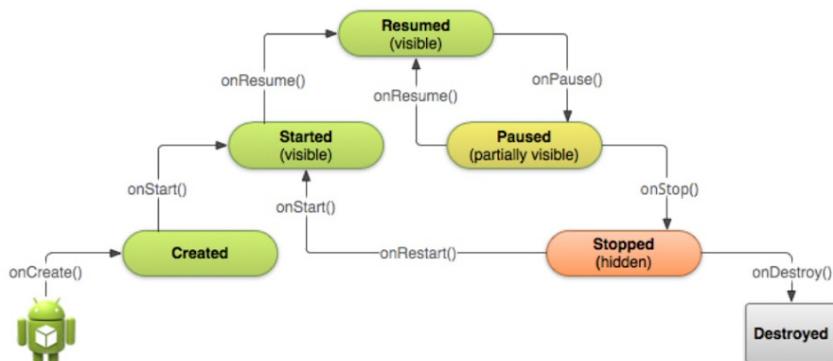
ต้องเริ่มต้นแอปพลิเคชันใหม่ เป็นต้น

#### 2.1.4 กระบวนการเริ่มทำงานของแอคทิวิตี้ (Activity)

ในระบบแอนดรอยด์การกำหนดโค้ดเริ่มต้นไว้ในแอคทิวิตี้โดยสัมพันธ์กับ Method ที่ถูกเรียกใช้งานอัตโนมัติ (Callback Method) อย่างเป็นลำดับ ตั้งแต่เริ่มต้นแอคทิวิตี้ไปจนถึงสิ้นสุดและปิดการทำงานของ Activity ลง

#### 2.1.5 ทำความรู้จักกับ Lifecycle Callback

ในขณะที่แอคทิวิตี้ [4] ทำงานระบบจะเรียกใช้ Callback Method ตามลำดับในลักษณะที่คล้ายกับการก่อพิรภพ นั่นคือ แต่ละขั้นตอนวัฏจักรของแอคทิวิตี้คือส่วนแยกอยู่แต่ละขั้นของพิรภพ เช่น เมื่อระบบสร้าง Instance ของแอคทิวิตี้ขึ้นมาใหม่ Method ที่เรียกใช้งานอัตโนมัติ (Callback Method) จะขยับ Activity Method ขึ้นมาด้านบนโดยด้านบนของพิรภพคือจุดที่แอคทิวิตี้กำลังทำงานแสดงอยู่ด้านหน้า (Foreground Activity) สุดและผู้ใช้กำลังใช้งานอยู่และเมื่อผู้ใช้กำลังจะออกจากแอคทิวิตี้ระบบจะเรียกใช้ Method อื่นซึ่งทำให้ Activity Method ถอยกลับไปอยู่ด้านล่างของพิรภพตามลำดับเพื่อหยุดการทำงานและลบแอคทิวิตี้ออกไป ในบางกรณีแอคทิวิตี้จะย้ายลงมาอยู่บางจุดและรอจังหวะที่จะถูกเรียกกลับขึ้นมาด้านบนอีก เช่น ในกรณีเมื่อผู้ใช้สลับไปใช้งานแอปพลิเคชันอื่นแล้วกลับมาใช้งานอีกครั้ง



รูปที่ 2.3: วัฏจักรของแอคทิวิตี้บนระบบปฏิบัติการแอนดรอยด์

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

จากรูปที่ 2.3 แสดงวัฏจักรของแอคทิวิตี้ในรูปแบบโครงสร้างพิรภพโดยแสดงให้เห็นว่า Method ที่เรียกใช้ งานอัตโนมัติ (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 กรณีมีสายโทรศัพท์เข้าหรือมีการสลับไปใช้งานแอปพลิเคชันอื่น
- ไม่ใช้ทรัพยากรที่มีค่าของระบบอย่างสูญเปล่า ถ้าไม่มีการใช้งานแอคทิวิตี้ใดๆ
- ไม่กระทบต่อกระบวนการในขั้นตอนการใช้งานของผู้ใช้กรณีออกจากแอปพลิเคชันแล้วกลับเข้ามาใช้งานอีกครั้ง
- ไม่หยุดการทำงานหรือระบบค้างที่กระทบการใช้งานของผู้ใช้กรณีมีการหมุนหน้าจอแนวอนอนและแนวตั้งสลับกัน

เหตุการณ์ที่แอคทิวิตี้มีการเปลี่ยน Method ต่าง ๆ ตามแสดงในรูปที่ 2.3 แต่มีอยู่ 3 Method เท่านั้นที่แอคทิวิตี้จะยังคงอยู่คung ที่ในช่วงเวลาระยะเวลาหนึ่งไม่เปลี่ยนไป Method อื่นในทันทีได้แก่

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

ในขณะที่ Method อื่น เช่น Created และ Started จะแสดงช่วงระหว่างระบบก็จะเปลี่ยนไป

Method อื่นในทันทีที่ Method ถูกเรียกใช้งานอัตโนมัติ นั่นคือ หลังจากที่ระบบเรียกใช้งาน onCreate() แล้ว ก็จะเรียกใช้งาน onStart() ทันทีและสุดท้ายตามด้วย onResumne() ซึ่งก็จะเข้าสู่ Resumed Method ทั้งหมดก็คือวัฏจักรแอคทิวิตี้เบื้องต้น

#### 2.1.6 การกำหนด Launcher Activity ในแอปพลิเคชัน

เมื่อผู้ใช้กดที่ Icon App จากหน้า Home Screen เพื่อใช้งาน ระบบก็จะเรียก onCreate() Method ขึ้นมาทำงานอัตโนมัติเพื่อที่จะเปิดใช้งานแอคทิวิตี้ที่ได้กำหนดให้เป็นแอคทิวิตี้หลัก ("launcher" หรือ "main") ซึ่งก็จะเป็นแอคทิวิตี้ที่เป็นหน้าหลักที่ผู้ใช้จะเห็นเมื่อเข้ามาใช้งานแอปพลิเคชัน สามารถกำหนดได้ว่าแอคทิวิตี้ใดที่จะใช้เป็น Activity หลักโดยกำหนดค่าในไฟล์ AndroidManifest.xml

แอคทิวิตี้หลักจะต้องกำหนดค่าใน manifest ด้วย <intent-filter>

```

1 <intent-filter>
2 <action android:name="android.intent.action.MAIN" />
3 <category android:name="android.intent.category.
   LAUNCHER" />
4 </intent-filter>

```

รูปที่ 2.4: การกำหนด Launcher Activity ในแอปพลิเคชัน

โดยประกอบไปด้วย MAIN action และ LAUNCHER category ดังตัวอย่างดังนี้

```

1 <Activity android:name=".MainActivity" android:label
   ="@string/app_name">
2 <intent-filter>
3 <action android:name="android.intent.action.MAIN" />
4 <category android:name="android.intent.category.
   LAUNCHER" />
5 </intent-filter>
6 </Activity>

```

รูปที่ 2.5: การกำหนด Launcher Activity ในแอปพลิเคชัน

ใน Android Studio เมื่อสร้างโปรเจก (Project) ขึ้นมาจะมีการประกาศค่าและกำหนดให้มีการเรียกใช้งาน filter ในไฟล์ AndroidManifest.xml มาให้เรียบร้อยแล้ว ถ้าไม่ได้กำหนด MAIN

action และ LAUNCHER category แล้ว App Icon ของจะไม่แสดงในหน้าหลักของลิสรายการแอปพลิเคชันบนหน้าจอสมาร์ทโฟนของผู้ใช้

### 2.1.7 การสร้าง Instance ใหม่

แอปพลิเคชันส่วนใหญ่จะประกอบไปด้วยแอคทิวิตี้หลายอันทำงานแยกต่างกันไป ไม่ว่าจะเป็นแอคทิวิตี้หลักที่ถูกสร้างขึ้นเมื่อผู้ใช้คลิกที่ icon app หรือแอคทิวิตี้ต่าง ๆ ที่ถูกเรียกใช้งานโดยผู้ใช้ ทั้งหมดล้วนแล้วเป็นสิ่งที่ระบบสร้าง instance ใหม่ของแอคทิวิตี้โดยการเรียกใช้ผ่าน `onCreate()` Method ต้องใช้ `onCreate()` Method ในเหตุผลเพื่อเริ่มต้นแอปพลิเคชันซึ่งควรจะเกิดขึ้นเพียงครั้งเดียวตลอดหนึ่งวัฏจักรของแอคทิวิตี้ เช่น ใช้ `onCreate()` กำหนดหน้าตาของแอปพลิเคชันหรือประกาศตัวแปรที่จะถูกใช้งานในคลาส (Class) เป็นต้น ตัวอย่างการใช้งาน `onCreate()` Method ตามตัวอย่างรูปที่ 2.6 แสดง code เพื่อให้เห็นการตั้งค่าเบื้องต้น เช่น การกำหนด User interface (โดยใช้ XML layout ไฟล์) การประกาศตัวแปรต่าง ๆ การตั้งค่ากำหนดเงื่อนไข UI

```

1 TextView mTextView; // การประกาศตัวแปรจำพวก text view ใน layout
2
3 @Override
4 public void onCreate(Bundle savedInstanceState) {
5     super.onCreate(savedInstanceState);
6
7 // การกำหนด user interface โดยใช้ XML layout ไฟล์
8 // ไฟล์ layout ใน project จะอยู่ที่ res/layout/main_Activity.xml
9     setContentView(R.layout.main_Activity);
10
11 // กำหนดค่าให้กับ TextView ดังนี้ สามารถเรียกใช้งานผ่านชื่อตัวแปรได้ภายหลัง
12     mTextView = (TextView) findViewById(R.id.text_message);
13
14 // ตั้งค่าสำหรับกำหนดเงื่อนไข UI
15 // ตรวจสอบเพื่อยืนยันว่ากำลังใช้งานบน android Honeycomb
16 // หรือสูงกว่าเพื่อเรียกใช้ ActionBar API
17     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.
18         HONEYCOMB) {
19         // กำหนดให้ icon ใน action bar ไม่ทำงานคล้ายปุ่ม Home
20         ActionBar actionBar = getActionBar();
21         actionBar.setHomeButtonEnabled(false);
22     }

```

รูปที่ 2.6: การใช้งาน onCreate()

เมื่อ onCreate() ทำงานเสร็จระบบจะเรียก onStart() และ onResume() method มาทำงานต่อในทันที แต่ถ้าต้องการที่จะไม่执行ใน Created Method หรือ Started Method ในทางเทคนิคแล้ว ก็ต้องตั้งเงื่อนไขไว้ใน onStart() ถูกเรียกใช้งานแต่ onResume() จะแสดงตามมาติด ๆ ในทันทีและจะอยู่ใน Resumed Method จนกว่าจะมีการเปลี่ยนแปลงบางอย่างเกิดขึ้น เช่น มีสายโทรศัพท์เข้าหรือผู้ใช้เปิดไปแอคทิวิตี้อื่นหรือปิดหน้าจอลง

โครงสร้างของวัสดุจักรแอคทิวิตี้ที่เน้นไปที่ 3 Method หลักที่ระบบเรียกใช้อัตโนมัติตามลำดับ เมื่อมีการสร้าง Instance ของแอคทิวิตี้ขึ้นมาใหม่ ได้แก่ onCreate(), onStart() และ onResume() เมื่อลำดับการทำงานเสร็จสิ้นลงแอคทิวิตี้จะ มาอยู่ที่ Resumed Method ซึ่งผู้ใช้งานอยู่จนกว่า จะเปลี่ยนไปเรียกใช้งานแอคทิวิตี้อื่นขึ้นมา

### 2.1.8 การสิ้นสุดการทำงานของแอคทิวิตี้ (Destroy Activity)

ในขณะที่วิจารณ์ของแอคทิวิตี้มี Method แรกคือ `onCreate()` และ Method สุดท้ายคือ `onDestroy()` ระบบจะเรียก Method นี้ในแอคทิวิตี้เป็นสัญญาณว่า Activity instance นี้จะถูกลบ ออกจากอย่างสมบูรณ์จากหน่วยความจำของระบบ (System memory)

แอปพลิเคชันส่วนใหญ่ไม่จำเป็นต้องกำหนด Method นี้ เพราะ local class ถูกทำลายไปพร้อมกับแอคทิวิตี้และแทบจะถูกลบไปอย่างสมบูรณ์ในระหว่างเรียกใช้งาน `onPause()` และ `onStop()` อย่างไรก็ตามถ้าแอคทิวิตี้ของมีการกำหนดให้มีการทำงานแบบ background (ทำงานอยู่เบื้องหลัง) ในชั้นตอน `onCreate()` ด้วยแล้วหรือมีการใช้งานทรัพยากรของระบบเป็นระยะเวลานาน เป็นเหตุให้ความจำถูกใช้งานจำนวนมาก หากไม่ทำการปิดการใช้งาน จึงควรอย่างยิ่งที่จะเรียกใช้งาน Method `onDestroy()` เพื่อคืนค่าหน่วยความจำให้ระบบ

```

1  @Override
2  public void onDestroy() {
3      super.onDestroy(); // เรียกใช้งานsuperclass ทุกครั้งเสมอ
4
5  // หยุดการติดตามหรือตรวจสอบMethod ของActivity
6      android.os.Debug.stopMethodTracing();
7 }
```

รูปที่ 2.7: การใช้งาน `onDestroy()`

ระบบจะเรียกใช้งาน `onDestroy()` หลังจากเรียกใช้งาน `onPause()` และ `onStop()` แล้วในทุกเหตุการณ์ ยกเว้นกรณีมีการกำหนดให้เรียกใช้ `finish()` ภายใน `onCreate()` Method ในกรณี เช่น Activity หยุดชั่วคราวเพื่อทำการเรียกใช้งานแอคทิวิตี้อื่นขึ้นมาแล้วทำการกำหนดให้มีการเรียกใช้งาน `finish()` ที่กำหนดใน `onCreate()` Method เพื่อทำลายแอคทิวิตี้ในกรณีระบบจะทำการเรียก `onDestroy()` ทันทีโดยไม่ผ่าน Callback Method อื่นเช่น ไม่ผ่าน `onPause()` หรือ `onStop()` แบบนี้เป็นต้น

## 2.2 ความรู้พื้นฐาน Java

Java [5] เป็นภาษาเขียนโปรแกรมเพื่อวัตถุประสงค์ทั่วไป โดยสามารถทำงานได้พร้อมกัน เป็นภาษาที่สร้างมาจากคลาส และสนับสนุนการเขียนโปรแกรมแบบออบเจกต์อย่างสมบูรณ์ และถูก

ออกแบบมาให้พร้อมสำหรับการใช้งานมากที่สุด ซึ่งมีเมธอดและคลาสต่าง ๆ อำนวยความสะดวก ให้ใช้มากมาย โดยภาษา Java นั้นมีความตั้งใจว่าจะทำให้นักพัฒนาออกแบบและพัฒนาโปรแกรม น้อยลง นั่นคือการเขียนเพียงครั้งเดียว แต่นำไปใช้งานได้ทุกที่หรือทุกแพลตฟอร์ม

แอปพลิเคชันของภาษา Java นั้นโดยปกติแล้วจะคอมไพล์เป็น bytecode ที่สามารถรันได้ใน Java virtual machine (JVM) ขึ้นกับสถาปัตยกรรมของคอมพิวเตอร์นั้นๆ และในปี 2016 ภาษา Java เป็นภาษาที่ได้รับความนิยมและใช้มากที่สุดในโลก โดยเฉพาะการใช้พัฒนาเว็บแอปพลิเคชัน

### 2.2.1 ประวัติความเป็นมาของภาษา Java

James Gosling Mike Sheridan และ Patrick Naughton ได้เริ่มก่อตั้งโครงการภาษา Java ของพวกเขาเมื่อปี 1991 โดยในตอนแรกมันถูกพัฒนาสำหรับทีวีที่สามารถมีปฏิสัมพันธ์ได้ เช่น เล่น เกมในทีวีได้ แต่มันยากเกินไปในการที่จะใช้งานกับสายเคเบิลของทีวีดิจิตอลในเวลานั้น ในตอนแรกภาษา Java ใช้ชื่อว่า Oak เพราะว่ามีต้นโอ๊คยืนออกใบยังออฟฟิศของ Gosling ต่อมาใช้ชื่อว่า Green และในตอนท้ายใช้ชื่อว่า Java มีที่มาจากการแฟ้ม Java

ภาษาได้รับการออกแบบให้มีรูปแบบทางภาษาเหมือนภาษา C และ C++ ซึ่งทำให้โปรแกรมเมอร์ส่วนมากนั้นคุ้นเคยกับมันได้ดีขึ้น และ Sun Microsystems เผยแพร่ Java 1.0 ในปี 1995 โดยมีคำกล่าวว่า ”Write Once, Run Anywhere” (WORA) ซึ่งมันพรี เขียนเพียงครั้งเดียว และสามารถนำไปรันได้บนทุกแพลตฟอร์ม

### 2.2.2 Java Compiler

ในการเขียนโปรแกรมในภาษา Java ต้องการ Java Compiler เพื่อทำการแปลงโค้ดของโปรแกรมที่เขียนเป็น bytecode เพื่อนำไปรันในแต่ละแพลตฟอร์มต่อไป โดยเรียกว่า Java Platform (JDK) ซึ่งประกอบไปด้วยคอมไпал์เลอร์ ในการแปลงโค้ดภาษา Java ให้เป็น Bytecode และ Java virtual machine (JVM) สำหรับรันโปรแกรมของภาษา Java ในแต่ละแพลตฟอร์ม สำหรับในบทเรียนนี้จะใช้ IDE ในการพัฒนาเพื่อความสะดวกและรวดเร็ว

### 2.2.3 โครงสร้างของภาษา Java

---

```

1 public class ClassName {
2     public static void main(String[] args) {
3         System.out.println("Hello สวัสดี");
4     }
5 }
```

---

รูปที่ 2.8: ตัวอย่างโปรแกรมภาษา Java ด้วย minted

รูปที่ 2.8 แสดงให้เห็นผลลัพธ์ของการใช้ minted เพื่อแสดงคำสั่งในเอกสาร Latex

- Package เป็นกลุ่มของคลาสหรือไลบรารี่มาตรฐานของภาษา Java ที่มีฟังก์ชันต่าง ๆ ให้ใช้มากมาย
- Class ในส่วนของการประกาศคลาส จะต้องประกาศคลาสให้ชื่อตรงกับไฟล์เสมอ นอกจาก Inner คลาสที่อยู่ในคลาสเดียวกัน โดยชื่อคลาสนั้นควรจะขึ้นต้นด้วยตัวใหญ่และถ้ามีหลายคำให้ใช้ตัวพิมพ์ใหญ่แบ่ง ดังตัวอย่างในรูปที่ 2.9

```

1 public class ClassName {
2 ...
3 }
```

รูปที่ 2.9: การประกาศคลาสในภาษา Java

- Method หลังจากคลาสร้างแล้ว จะเป็นประกาศเมธอดภายในคลาส โดยในการที่จะรันโปรแกรมได้จะต้องมีเมธอดที่ชื่อว่า Main ดังตัวอย่างในโปรแกรมด้านบน มันเป็นที่แรกที่โปรแกรมจะเริ่มทำงาน 2.9

```

1 public static void main (String[] args) {
2 ...
3 }
```

รูปที่ 2.10: การประกาศเมธอด main ในภาษา Java

- Methodments เป็นคำสั่งของโปรแกรมเพื่อให้โปรแกรมทำงานตามต้องการ เช่น System.out.println("Hello World!");; เป็นการแสดงผลข้อความออกทางหน้าจอ โดยปกติโปรแกรมมักจะมีหลายคำสั่ง
- Semicolon ทุกคำสั่งการทำงานของโปรแกรมในภาษา Java จะจบด้วยเครื่องหมาย Semicolon (;) นั่นหมายความว่าสามารถเขียนโปรแกรมแบบไหนก็ได้ โดยคอมไพล์อร์จะทราบอัตโนมัติว่าสิ้นสุดคำสั่งที่ไหน
- ในภาษา Java สามารถใช้ White space ได้อย่างอิสระตามที่ต้องการ โดย White space จะประกอบไปด้วย Space bar Tab และ Enter (return) เพราะว่าคอมไпал์อร์ตรวจการสิ้นสุดของคำสั่งด้วย ; ใช้ while space ทำให้โค้ดอ่านเข้าใจง่าย และเป็นระเบียบ
- Literals คือค่าของข้อมูลใดๆ ที่กำหนดให้กับตัวแปรได้ เรียกว่า Constant Literals ทุกค่าที่เป็นไปได้ เช่น "MarcusCode" เป็น String Literals 10 เป็น Integer Literals หรือ true เป็น Boolean Literals โดย Literals เป็นได้แค่ Primitive data type เท่านั้น ตัวอย่างการกำหนดค่าหรือ Literals ให้กับตัวแปร
- Expression เป็นการกระทำระหว่างตัวแปรกับตัวดำเนินการเพื่อให้ได้ผลลัพธ์ใหม่ เช่น 4 + 3 เป็น expression ของการบวกเลขและได้ได้ผลลัพธ์เท่ากับ 7 หรือ 1 == 1 เป็น expression ของการเปรียบเทียบระหว่างค่าสองค่าว่าเท่ากันหรือไม่ และได้ผลลัพธ์เป็น true
- Keyword คือคำที่สงวนไว้ในภาษา Java นั่นหมายความว่าไม่สามารถนำคำเหล่านี้ไปประกาศเป็นชื่อตัวแปร เมรอด หรือว่าคลาสได้ เพราะว่า Keyword ถูกใช้โดยคอมไпал์อร์เพื่อให้มันทำงานได้สมบูรณ์

### 2.3 ความรู้พื้นฐาน Java Script

JavaScript [6] คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็นภาษาสคริปต์เชิงวัตถุที่เรียกว่า "สคริปต์" (Script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (เช่นกับ HTML) เพื่อให้เว็บไซต์ของคุณมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (Interpret) หรือเรียกว่า อ็อบเจกต์ออเรียนเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้โดยทำงานร่วมกับภาษา HTML และภาษา Java Script ทำงานได้ทั้งทางฝั่งคุลเลนต์ (Client) และทางฝั่งเซิร์ฟเวอร์ (Server)

JavaScript ถูกพัฒนาขึ้นโดย เน็ตสเปคคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกแบบร่วมกับ Netscape Navigator2.0 เพื่อใช้สร้างเว็บเพจโดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเปคจึงได้ร่วมมือกับ บริษัทชั้นนำโคร์ซิสเต็มส์ปรับปรุงระบบของบราวเซอร์เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวาได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี 2538 แล้วตั้งชื่อใหม่ว่า JavaScript

JavaScript สามารถทำให้ การสร้างเว็บเพจ มีลูกเล่น ต่าง ๆ มากมายและสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิกหรือการกรอกข้อมูลในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนาสามารถสร้างเว็บเพจได้ตรงกับความต้องการและมีความน่าสนใจมากขึ้นประกอบกับเป็นภาษาเบ็ด ที่ใครก็สามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย ECMA การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยบราวเซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้ เฉพาะบนบราวเซอร์ที่สนับสนุน ซึ่งปัจจุบันบราวเซอร์เกือบทั้งหมดก็สนับสนุน JavaScript แล้ว อย่างไรก็ได้สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ ๆ อย่างต่อเนื่องดังนั้น ถ้าหากมีโค้ดของเวอร์ชันใหม่ไม่ได้รับการสนับสนุนอาจทำให้เกิดข้อผิดพลาด (Error) ได้

### 2.3.1 ลักษณะการทำงานของ JavaScript

JavaScript เป็นภาษาสคริปต์เชิงวัตถุหรือเรียกว่า อ็อบเจกต์ออเรียนเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนเอกสารด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ทำงานร่วมกับภาษา HTML และภาษาจาวาได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server) โดยมีลักษณะการทำงานดังนี้

- Navigator JavaScript เป็น Client-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลงทางฝั่งไคลเอนต์ หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ใช้ไม่ว่าจะเป็นเครื่องพีซี (Personal computer, PC) เครื่องแมคอินทอช (Macintosh) หรืออื่น ๆ จึงมีความเหมาะสมต่อการใช้งานของผู้ใช้ทั่วไปเป็นส่วนใหญ่
- LiveWire JavaScript เป็น Server-Side JavaScript ซึ่งหมายถึง JavaScript ที่ถูกแปลงทาง

ฝั่งเซิร์ฟเวอร์ (หมายถึงฝั่งเครื่องคอมพิวเตอร์ของผู้ให้บริการเว็บโดยอาจเป็นเครื่องของชั้นชิลล์คอมกราฟิกส์หรืออื่น ๆ) สามารถใช้ได้เฉพาะกับ LiveWire ของเน็ตสเคป โดยตรง

### 2.3.2 JavaScript กับ HTML

การเขียน JavaScript อาจเขียนรวมอยู่ในไฟล์เดียวกันกับ HTML ได้ ซึ่งแตกต่างจากการเขียนโปรแกรมภาษา Java ที่ต้อง เขียนแยกออกเป็นไฟล์ต่างหากไม่สามารถเขียนรวมอยู่ในไฟล์เดียวกับ HTML ได้ วิธีการเขียน JavaScript เพื่อสั่งให้เว็บเพจทำงาน มือญี่ด้วยกัน 2 วิธี ดังนี้

- เขียนด้วยชุดคำสั่งและฟังก์ชันของ JavaScript เอง
- เขียนตามเหตุการณ์ที่เกิดขึ้นตามการใช้งานจากชุดคำสั่งของ HTML

เมื่อเริ่มใช้งานโปรแกรมбраузอร์จะอ่านข้อมูลจากส่วนบนของไฟล์ HTML และทำงานไปตามลำดับจากบนลงล่าง (top-down) โดยเริ่มที่ส่วน `< HEAD >...< /HEAD >` ก่อนจากนั้นจึงทำงานในส่วน `< BODY >...< /BODY >` เป็นลำดับต่อมา การทำงานของ JavaScript ดูไม่แตกต่างไปจาก HTML เท่าไนด์ แต่ HTML จะวางแผนเอาต์โครงสร้างของอ็อบเจกต์ภายใน และส่วนเชื่อมโยงกับเว็บเพจเท่านั้น ในขณะที่ JavaScript สามารถเพิ่มเติมส่วนของการเขียนโปรแกรมและลอกิจเข้าไป

### 2.3.3 โครงสร้างภาษา

1. ตัวแปร (Variable) หมายถึง ชื่อหรือสัญลักษณ์ที่ตั้งขึ้นสำหรับการเก็บค่าใด ๆ ที่ไม่คงที่ โดยการจะเนื้อที่ในหน่วยความจำของระบบเครื่องที่เก็บข้อมูลซึ่งสามารถอ้างอิงได้ มีขนาดขึ้นอยู่กับชนิดของข้อมูลและค่าของข้อมูล ซึ่งค่าในตัวแปรนี้สามารถเปลี่ยนแปลงได้ตามคำสั่งในการประมวลผล
2. การตั้งชื่อ (Identifier or Name) เป็นชื่อที่ตั้งขึ้นมาเพื่อกำหนดให้เป็นชื่อของโปรแกรมหลัก, ฟังก์ชัน, ตัวแปร, ค่าคงที่, คำสั่ง และคำส่วน โดยมีหลักการตั้งชื่อว่า
  - ขึ้นต้นด้วยตัวอักษรในภาษาอังกฤษ ตามด้วยตัวอักษรหรือตัวเลขใด ๆ ก็ได้
  - ห้ามเว้นช่องว่าง
  - ห้ามใช้สัญลักษณ์พิเศษ ยกเว้นชีดล่าง (`_`) และดอลลาร์ (`$`)
  - สำหรับความยาวของชื่อใน JavaScript จะมีความยาวเท่าได้ก็ได้ แต่ที่นิยมใช้ ไม่เกิน

- การตั้งชื่อเมื่อพึงระวังว่า จะต้องไม่ซ้ำกับคำส่วน (Reserve word) และตัวอักษรของชื่อจะจำแนกแตกต่างกันระหว่างอักษรตัวพิมพ์เล็กกับอักษรตัวพิมพ์ใหญ่
- ควรจะตั้งชื่อโดยให้ชื่อนั้นมีสีความหมายให้เข้ากับข้อมูล สามารถอ่านและเข้าใจได้ตัวอย่างชื่อที่ถูกต้อง Hahaha, I\_Love\_you, Doll\$ เป็นต้น

- คำส่วน(Reserve word)เป็นคำที่มีความหมายเฉพาะตัวในภาษา JavaScript สงวนไว้ให้มีการตั้งชื่อซ้ำกับชื่อโปรแกรม, ฟังก์ชัน, ตัวแปร, ค่าคงที่ และคำสั่ง คำส่วน สามารถเรียกใช้ได้ทันทีโดยไม่ต้องมากำหนดความหมายใหม่แต่อย่างใด
- ชนิดของข้อมูลของตัวแปร (Data Type) เป็นการกำหนดประเภทค่าของข้อมูลให้กับตัวแปรเพื่อให้เหมาะสมกับการอ้างอิงข้อมูลจากตัวแปรในการใช้งาน ชนิดข้อมูลของตัวแปรนั้นมีอยู่ด้วยกัน 4 ชนิด ได้แก่
  - number หมายถึง ข้อมูลชนิดตัวเลข ประกอบด้วย เลขจำนวนเต็ม (Integer) และเลขจำนวนจริง (Floating)
  - logical หมายถึง ข้อมูลทางตรรกะ มี 2 สถานะ คือ จริง (True) และเท็จ (False)
  - string หมายถึง ข้อมูลที่เป็นข้อความ ซึ่งจะต้องกำหนดไว้ในเครื่องหมายคำพูด ("...")
  - null หมายถึง ไม่มีค่าข้อมูลใดๆ ซึ่งค่า null ใช้สำหรับการยกเลิกพื้นที่เก็บค่าของตัวแปรออกจากหน่วยความจำ
- การประกาศตัวแปร (Declarations) เป็นการกำหนดชื่อและชนิดข้อมูลให้กับตัวแปรเพื่อนำไปใช้ในโปรแกรม โดยการตั้งชื่อจะต้องคำนึงถึงค่าของข้อมูลและ ชนิดของข้อมูลที่อ้างอิง นอกจากนี้การตั้งชื่อควรให้สื่อความหมายของข้อมูล และอักษรของชื่อจะจำแนกแตกต่างกันระหว่างอักษรตัวพิมพ์เล็กกับอักษรตัวพิมพ์ใหญ่
 

รูปแบบ Var ชื่อตัวแปร; เป็นรูปแบบการประกาศตัวแปรปกติหรือ Var ชื่อตัวแปร = ข้อมูล; เป็นรูปแบบการกำหนดค่าเริ่มต้น ในกรณีที่ต้องการกำหนดตัวแปรหลายตัวในบรรทัดเดียว กันให้ใช้เครื่องหมาย คอมมา ( , ) คั่นระหว่างชื่อตัวแปรและปิดท้ายด้วยเครื่องหมายเชมิโคล่อน ( ; ) การกำหนดค่าให้กับตัวแปร

รูปแบบ ชื่อตัวแปร = ค่าของข้อมูล โดยที่ค่าของข้อมูล ได้แก่

  - ข้อมูลที่เป็นตัวเลข โดยกำหนดตัวเลขไปได้เลย เช่น num = 500
  - ข้อมูลในทางตรรกะ ได้แก่ จริง (True) หรือ เท็จ (False) เช่น test = True;
  - ข้อมูลสตริง ให้กำหนดอยู่ในเครื่องหมายคำพูด ("...") เช่น name = "Adisak";

ตัวแปรมี 2 จำพวก หากกำหนดชื่อตัวแปรไว้ที่โปรแกรมหลักโดยไม่ได้อยู่ภายในขอบเขตฟังก์ชันใด ๆ เรียกว่าเป็นตัวแปรแบบโกลบัล (Global) ตัวแปรจำพวกนี้จะมีค่าคงอยู่ในหน่วยความจำตลอดการทำงานของโปรแกรม ทำให้สามารถเรียกใช้ได้จากทุก ๆ ส่วนของโปรแกรม รวมถึงภายในฟังก์ชันต่าง ๆ ด้วย แต่ถ้ากำหนดตัวแปรไว้ภายในขอบเขตฟังก์ชันใด ๆ จะเรียกว่าเป็นตัวแปรแบบ โลคัล (Local) เพราะจะเป็นตัวแปรที่มีค่าคงอยู่ และสามารถเรียกใช้ได้เฉพาะ ภายในขอบเขตของฟังก์ชันนั้น ๆ เท่านั้น

6. ตัวแปรแบบอาร์เรย์ (Array) หมายถึงตัวแปรซึ่งมีค่าได้หลายค่าโดยใช้ชื่อ อ้างอิงเพียงชื่อเดียว ด้วยการใช้หมายเลขลำดับเป็นตัวจำแนกความแตกต่างของค่าตั้งแต่ละตัว ถ้าจะเรียกตัวแปรชนิดนี้ว่า ”ตัวแปรชุด” ก็เห็นจะไม่ผิดนัก ตัวแปรชนิดนี้มีประโยชน์มาก ลองคิดถึงค่าข้อมูลจำนวน 100 ค่า ที่ต้องการเก็บไว้ในตัวแปรจำนวน 100 ตัว อาจทำให้ต้องกำหนดตัวแปรที่แตกต่างกันมากถึง 100 ชื่อ กรณีอย่างนี้ควรจะทำอย่างไรดี แต่ด้วยการใช้สมบติอาร์เรย์ สามารถนำตัวแปรหลาย ๆ ตัวมาอยู่ร่วมเป็นชุดเดียวกันได้และสามารถเรียกใช้ตัวแปรทั้งหมดโดยระบุผ่านชื่อเพียงชื่อเดียวเท่านั้น ด้วยการระบุหมายเลขลำดับ หรือดัชนี (index) กำกับตามหลังชื่อตัวแปร ตัวแพรเพียงชื่อเดียวจึงมีความสามารถเทียบได้กับตัวแปรนับร้อยตัว พันตัว (ตัวที่ 1) ในตัวแปรแบบอาร์เรย์มีดัชนีเป็น 0 ส่วนตัวแปรต่อ ๆ ไปก็จะมีดัชนีเป็น 1,2,3,... ไปตามลำดับ เมื่อต้องการระบุชื่อตัวแปรแบบอาร์เรย์แต่ละตัว ก็จะใช้รูปแบบ name[0], name[1],... เรียงต่อกันไปเรื่อย ๆ สามารถสร้างตัวแปรอาร์เรย์ใหม่ด้วย `myArray = new Array()` ดังนี้

```

1 myArray[0] = 17;
2 myArray[1] = "Nun";
3 myArray[2] = "Stop";

```

รูปที่ 2.11: ตัวแปรอาร์เรย์

7. ค่าคงที่ (Literal หรือ Constant) หมายถึง ค่าของข้อมูลที่เมื่อกำหนดแล้วจะทำการเปลี่ยน-แปลงค่าเป็นอย่างอื่นไม่ได้ ชนิดข้อมูลของค่าคงที่ได้แก่

- เลขจำนวนเต็ม (Integer) เป็นตัวเลขที่ไม่มีเศษทศนิยม สามารถเขียนให้อยู่ในแบบเลขฐานสิบ (0-9), เลขฐานสิบหก (0-9, A-F) หรือ เลขฐานแปด (0-7) โดยการเขียนเลขฐานแปดให้ นำหน้าด้วย O (Octenary) ส่วนการเขียนเลขฐานสิบหกให้นำหน้า

### ด้วย Ox หรือ OX (Hexadenary)

- เลขจำนวนจริง (Floating) ใช้รูปแบบการเขียนโดยประกอบไปด้วยตัวเลข จุดทศนิยม และตัวเลขยกกำลัง E (Exponential) เช่น  $5.00E2$  จะหมายถึงค่า  $5.00$  คูณด้วย  $10$  ยกกำลัง  $2$  จะมีค่าเป็น  $500$ 。 $3.141E5$  จะหมายถึงค่า  $3.141$  คูณด้วย  $10$  ยกกำลัง  $5$  จะมีค่าเป็น  $314,1000$
- ค่าบูลิน (Boolean) เป็นค่าคงที่เชิงตรรกะ คือมีค่าเป็น จริง(True) และ เท็จ(False) เท่านั้น
- ข้อความสตริง (String) เป็นค่าคงที่แบบข้อความที่อยู่ภายใต้เครื่องหมายคำพูด ('...') หรือ ('...') เช่น ”บริษัท เอ็กซ์เพรส จำกัด”, ’นางนฤมล เวชตะระกุล’

8. รหัสคำสั่งพิเศษ (Character escape code) เป็นการกำหนดรหัสเพื่อควบคุมงานพิมพ์สตริง โดยใช้เครื่องหมาย Backslash นำหน้าตัวอักษรที่ต้องการกำหนดเป็นรหัส เพื่อให้กลับเป็น รหัสคำสั่งพิเศษ รหัส Character escape code

9. นิพจน์ (Expression) เป็นข้อคำสั่งที่ใช้กำหนดค่าของข้อมูล เช่น การบวกตัวเลข การเปรียบเทียบข้อมูล โดยการกำหนดชื่อของตัวแปร ตามด้วยเครื่องหมายที่ต้องการกระทำ (Operations) ต่อข้อมูลเป็นผลให้เกิดค่าข้อมูลใหม่ค่าหนึ่งให้กับตัวแปรเพื่อนำไปใช้งาน นิพจน์ JavaScript มีด้วยกัน 3 ชนิดดังนี้

- นิพจน์คณิตศาสตร์ (Arithmetic) เป็นนิพจน์ที่ใช้เครื่องหมายทางคณิตศาสตร์เป็นตัวกระทำ ผลลัพธ์ที่ได้จะมีค่าเป็นตัวเลขให้กับตัวแปร เช่น ให้ตัวแปร num เก็บตัวเลข  $5000$  จะเขียนได้ดังนี้  $num = 5000;$
- นิพจน์ตรรกะ (Logical) เป็นนิพจน์ในการเปรียบเทียบข้อมูลโดยใช้เครื่องหมายใน การเปรียบเทียบเพื่อตรวจสอบข้อมูลในการเปรียบเทียบว่าจริงหรือเท็จ เช่น กำหนดให้

1	$a = 50;$
2	$b = 70;$
3	$c = b > a;$

รูปที่ 2.12: การเปรียบเทียบว่าจริงหรือเท็จ

ผลลัพธ์ที่ได้คือ  $c$  จะมีค่าเป็นจริง (True)

- นิพจน์ข้อความ (String) เป็นนิพจน์เกี่ยวกับการกำหนดข้อความ การเชื่อมโยกข้อความ ใช้ประมวลผลข้อความในลักษณะต่าง ผลลัพธ์ที่ได้จะมีค่าเป็นตัวอักษรหรือข้อความเสมอ เช่น ให้ตัวแปร name เก็บชื่อ Adisak จะเขียนได้ดังนี้ name = "Adisak";
10. ตัวดำเนินการ (Operator) หมายถึง เครื่องหมายกำหนดกรรมวิธีทางคณิตศาสตร์, พีชคณิต, บูลีน, การเปรียบเทียบ ระหว่างข้อมูล 2 ตัว ซึ่งเรียกว่า โอลัรันเดอร์ (Operand) โดยอาจมีค่าเป็นตัวเลข ข้อความ ค่าคงที่ หรือตัวแปรต่าง ๆ
11. ชนิดของตัวดำเนินการ
- ตัวดำเนินการคณิตศาสตร์ (Arithmetic operator) หมายถึง ใช้สำหรับคำนวณโอลัรันเดอร์ที่เป็นค่าคงที่หรือตัวแปรที่ได้โดยให้ค่าผลลัพธ์เป็นตัวเลขค่าเดียว
  - ตัวดำเนินการเชิงเปรียบเทียบ (Comparison operator) หมายถึง เครื่องหมายในการเปรียบเทียบข้อมูล ผลลัพธ์ที่ได้จะมีค่าตրรบบูลีนเป็น จริง (True) และ เท็จ (False)
  - ตัวดำเนินการเชิงตรรกะ (Logical operator) เป็นเครื่องหมายที่ให้ค่าจริง (True) และ เท็จ (False) ในการเปรียบเทียบ
  - ตัวดำเนินการเชิงข้อความ (String operator) เป็นการเชื่อมโยกข้อความเข้าด้วยกัน (concatenation) โดยใช้เครื่องหมายบวก (+) เป็นตัวกระทำ เช่น

```

1 Name = "Bodin";
2 Say = "Hey "+Name;

```

รูปที่ 2.13: ตัวดำเนินการเชิงข้อความ

ผลลัพธ์ที่ได้ Say จะมีข้อความเป็น Hey Bodin

- ตัวดำเนินการระดับบิต (Bitwise operator) เป็นการดำเนินการเชิงตรรกะในระดับบิต โดยจะใช้มุมมองในแบบเลขฐาน 2 มาจัดการกับข้อมูล นั่นคือ ข้อมูลตัวเลขนั้นจะถูกแปลงเป็นเลขฐานสองในหน่วยความจำในขณะที่มีการดำเนินการเชิงตรรกะในระดับบิต ซึ่งโดยปกติแล้วการกระทำใน JavaScript จะอยู่ในระดับตัวอักษร ที่เรียกว่า ระดับไบต์ (byte)

## 2.4 ความรู้พื้นฐาน Vue.js Frontend Framework

Vue.js [7] เป็นเฟรมเวิร์คที่เน้นเรื่องการทำ User Interface คือเป็นภาษาสคริปต์เฟรมเวิร์คที่เข้ามาช่วยเรื่องการแสดงผล เป็นโอเพนซอร์ส (Open-Source) โดยวัตถุประสงค์หลังของการงานใช้วิจエอสคือ เพิ่มความสามารถการทำเว็บแอปพลิเคชันแบบ Single Page Application หรือ SPA

วิจエอสเป็น Frontend Framework ช่วยจัดการและทำให้การพัฒนาเว็บง่ายขึ้น เน้นพัฒนาส่วนติดต่อผู้ใช้งานเว็บ เช่น components, declarative UI, hot-reloading, time-travel debugging และอื่น ๆ ให้สะดวกต่อการใช้งาน พยายามลดสิ่งที่ไม่จำเป็นต่อการพัฒนาเว็บออกทำให้มีขนาดเล็กและง่ายต่อการนำไปใช้งานนักพัฒนา

มีสถาปัตยกรรมแบบ Adoptable สามารถนำไปประยุกต์ใช้จากหน้าเพจที่มีอยู่เดิม ทั้งนี้หากต้องการใช้งานความสารถขั้นสูงสำหรับแอปพลิเคชันที่ซับซ้อน จำเป็นต้องเพิ่มปลั๊กอิน (Plugin) อันได้แก่ Vue-router, Vuex และเครื่องมืออื่น ๆ ตามที่เราทางการของวิจエอสแนะนำ

### 2.4.1 การติดตั้ง

การติดตั้งเพื่อใช้งานวิจエอสทำได้หลายวิธีในที่นี้ขออธิบายด้วยกัน 2 วิธีได้แก่

- CDN (Content Delivery Network) ทำหน้าที่ให้การให้ User สามารถดาวน์โหลด Resource ต่าง ๆ บนเครือข่ายได้ เช่น jQuery, Bootstrap, jQuery UI, AngularJS เป็นต้น
- Vue Cli เป็นชุดคำสั่งที่ใช้ในการสร้าง Project ด้วย Vue.js ซึ่งได้มีการรวมชุดเครื่องมือรวมถึงไลบรารีต่าง ๆ ที่จำเป็นต่อการพัฒนาไว้ โดยเราสามารถเลือก Templates เพื่อใช้งานตามความเหมาะสมของงานได้
  1. ทำการใช้คำสั่ง `npm install -g vue-cli` เพื่อติดตั้ง Vue CLI
  2. ทำการใช้คำสั่ง `vue init webpack my-project` เพื่อการสร้าง Project
  3. ทำการใช้คำสั่ง `cd my-project` เพื่อเข้าไปที่ path ของ Project
  4. ทำการใช้คำสั่ง `npm install` เพื่อติดตั้ง node module ซึ่งเป็นชุดเครื่องมือที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชัน
  5. ทำการใช้คำสั่ง `npm run dev` เริ่มการทำงานของโปรแกรม

## 2.4.2 គុណលែកខ្លួននៃវិវាញផែដែ

- Templates วิวเจาเอกสารใช้งาน HTML เป็น syntax ซึ่งอนุญาตให้ผู้ใช้ทำการความคุมการแสดงผลภายใต้ DOM โดยทำผ่าน Vue instance's data ความสามารถหลักของ Templates คือสามารถความคุมการแสดงผลด้วยการเพิ่มคำสั่งหรือเงื่อนไขเข้าไปภายใต้แท็ก HTML ปกติ
  - ReActivity ความสามารถที่โดยเด่นที่เห็นได้ชัดเจนคือ การทำงานในส่วนของ re-rendering ที่รวดเร็ว
  - Components เป็นอีกหนึ่งความสามารถพิเศษของวิวเจาเอกสาร ในแอปพลิเคชันขนาดใหญ่จำเป็นต้องมีการแบ่งแอปพลิเคชันทั้งหมดออกเป็นส่วนย่อย ๆ เพื่อให้สามารถนำกลับมาใช้งานได้และทำให้ง่ายต่อการจัดการ

```
1 <div id="tuto">
2     <button@click v-bind:initial_count="0"></
3         buttonclicked>
4 </div>
5 <script>
6     Vue.component('buttonclicked', {
7         props: ["initial_count"],
8         data: function() {return {"count": 0} }
9             ,
10        template: '<button v-on:click="onclick">
11            Clicked {{ count }} times</button>',
12        Methods: {
13            "onclick": function() {
14                this.count = this.count + 1;
15            }
16        },
17        mounted: function() {
18            this.count = this.initial_count;
19        }
20    });
21
22 new Vue({
23     el: '#tuto',
24 });
</script>
```

รูปที่ 2.14: Components ของวิวเจอส

จากภาพที่ 2.14 แสดงวิวเจอส component โดยในบรรทัดที่ 5 ถึง 17 เป็นการสร้างวิวเจอส component

- Routing ในเทคโนโลยี SPA ข้อเสียหนึ่งข้อคือ ไม่สามารถส่งผ่านข้อมูลไปยัง Component อื่นได้ เนื่องจาก SPA เป็น URL-based response เพื่อแก้ปัญหาดังกล่าววิวเจอสได้สร้างส่วนประสานของตัวเองขึ้น คือ Routing ซึ่งใช้ส่งผ่านข้อมูลไปยัง Component อื่น

#### 2.4.3 ข้อดีของ Vue.js

- ช่วยแยก Logic การตัดสินใจออกจากโค๊ด (Code) การแสดงผล
- ช่วยแยกหน้าเว็บออกเป็น Component ทำให้การจัดการง่ายขึ้นและนำกลับมาใช้ได้
- ช่วยจัดการเรื่อง Dynamic data
- มีการเก็บสถานะต่าง ๆ ไว้ที่จุดเดียวเพื่อให้ง่ายต่อการจัดการและการเรียกใช้งานของ Component โดยการใช้งาน Vuex

### 2.5 ความรู้พื้นฐาน ScanLibrary

ScanLibrary [8] เป็นไลบรารีที่ใช้สแกนเอกสารต่าง ๆ บนแอนดรอยด์พัฒนามาจาก OpenCV ผลจากการประมวลผลจะได้เป็น Uri สิ่งสามารถนำมาแปลงเป็นข้อมูลรูปภาพเพื่อบันทึกลงฐานข้อมูลได้

#### 2.5.1 ขั้นตอนการติดตั้งและใช้งาน

- ทำการดาวน์โหลดไลบรารีจาก <https://github.com/jhansireddy/AndroidScannerDemo>
- ทำการนำเข้าไลบรารีผ่านคำสั่ง compile project(':scanlibrary')
- เมื่อต้องการใช้งาน ให้นำโค๊ดดังรูปที่ 2.15 ไปวางไว้ที่เมธอดที่ต้องการเริ่มการทำงาน

```

1 int REQUEST_CODE = 99;
2 int preference = ScanConstants.OPEN_CAMERA;
3 Intent intent = new Intent(this, ScanActivity.
4     class);
5 intent.putExtra(ScanConstants.
    OPEN_INTENT_PREFERENCE, preference);
startActivityForResult(intent, REQUEST_CODE);
```

รูปที่ 2.15: การเรียกใช้งาน

- ให้ทำการรับค่าที่ถูกส่งกลับจากไลบรารีด้วยคำสั่ง ดังรูปที่ 2.16

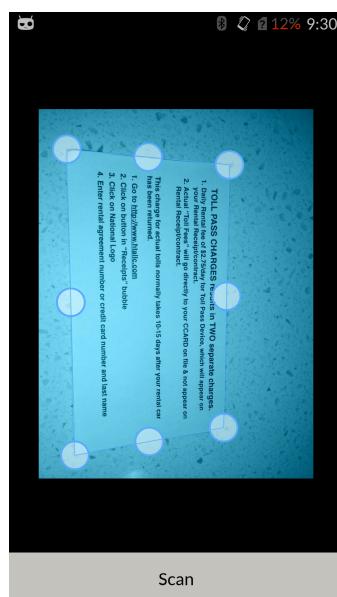
```

1  @Override
2  protected void onActivityResult(int requestCode, int
3      resultCode, Intent data) {
4      super.onActivityResult(requestCode, resultCode,
5          data);
6      if (requestCode == REQUEST_CODE && resultCode ==
7          Activity.RESULT_OK) {
8          Uri uri = data.getExtras().getParcelable(
9              ScanConstants.SCANNED_RESULT);
10         Bitmap bitmap = null;
11         try {
12             bitmap = MediaStore.Images.Media.
13                 getBitmap(getApplicationContext(), uri);
14             getContentResolver().delete(uri, null,
15                 null);
16             scannedImageView.setImageBitmap(bitmap);
17         } catch (IOException e) {
18             e.printStackTrace();
19         }
20     }
21 }
```

รูปที่ 2.16: การรับค่าจากไลบรารี

### 2.5.2 ตัวอย่างการใช้งาน

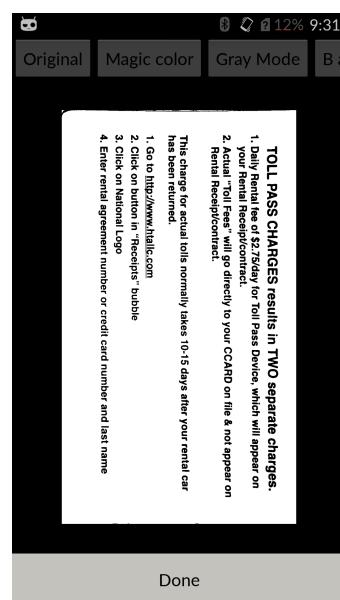
- ผู้ใช้ทำการถ่ายภาพเอกสารบนพื้นเรียบ



รูปที่ 2.17: ถ่ายภาพเอกสาร

ที่มา : <https://github.com/jhansireddy/AndroidScannerDemo>

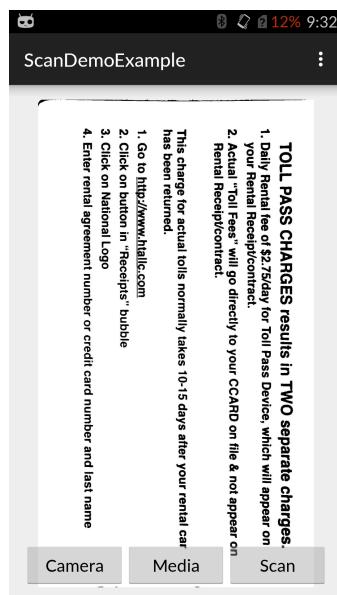
- ทำการปรับรูปภาพ เช่น การปรับสีรูปภาพ การปรับเป็นสีขาวดำ เป็นต้น



รูปที่ 2.18: ปรับแต่งภาพถ่าย

ที่มา : <https://github.com/jhansireddy/AndroidScannerDemo>

- ภาพที่ได้จากการประมวลผล



รูปที่ 2.19: ผลการทำงาน

ที่มา : <https://github.com/jhansireddy/AndroidScannerDemo>

## 2.6 ความรู้พื้นฐานของระบบ XX เดิม

### 2.6.1 ความเป็นมา

ระบบ XX เดิม มีประวัติดังนี้

### 2.6.2 วิสัยทัศน์และพันธกิจ

- วิสัยทัศน์ (Vision) ”เป็นองค์กรหลักที่ XX”
- พันธกิจ (Mission)

1. สนับสนุนและส่งเสริม XX

2. พัฒนาองค์กรสู่ความเป็นเลิศด้วยนวัตกรรมที่ทันสมัย โดยใช้หลักบริหารจัดการที่ดี

### 2.6.3 ขั้นตอนการดำเนินการ XX

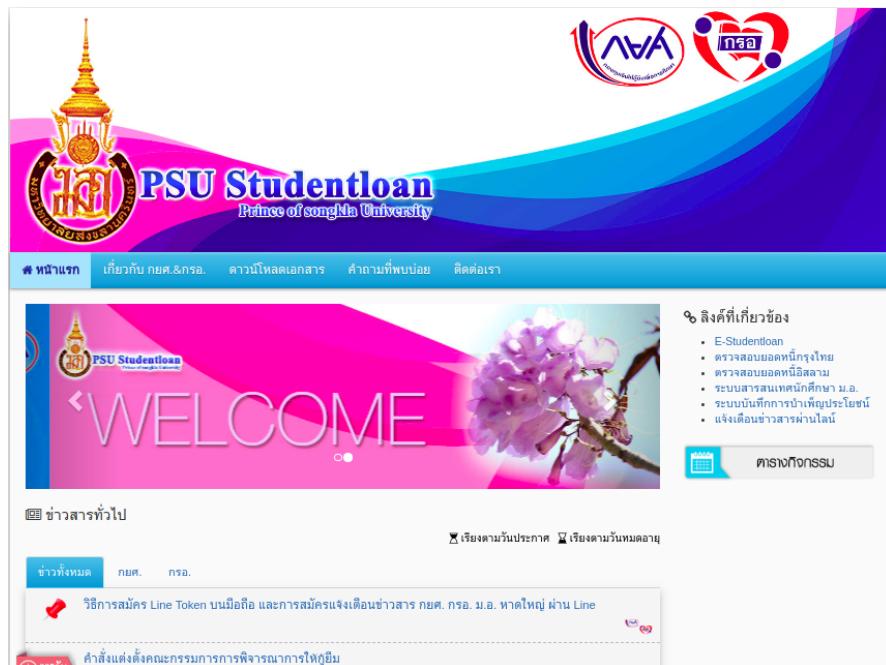
- อธิบายขั้นตอนที่ XX
- อธิบายขั้นตอนที่ XX
- อธิบายขั้นตอนที่ XX

## 2.7 เอกสารและงานวิจัยที่เกี่ยวข้อง

กล่าวถึงเอกสาร งานวิจัย หรือระบบงานที่คล้ายกันโดยแบ่งเป็น subsection โดยแต่ละหัวข้อให้อธิบายความสำคัญ พังก์ชันการทำงาน ข้อจำกัดหรือข้อแตกต่างจะระบบที่จะทำ เช่น

### 2.7.1 เว็บไซต์ XX

PSU Studentloan [9] เป็นเว็บไซต์ที่ให้บริการนักศึกษาของทุนเงินให้กู้ยืมเพื่อการศึกษา มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ สามารถเข้าใช้งานได้ที่ <https://studentloan.psu.ac.th/home> มีพังก์ชันการทำงานพื้นฐานอันได้แก่ การดูข่าวสารประชาสัมพันธ์ ประมวลภาพกิจกรรม ดาวน์โหลดเอกสารที่เกี่ยวข้อง เป็นต้น



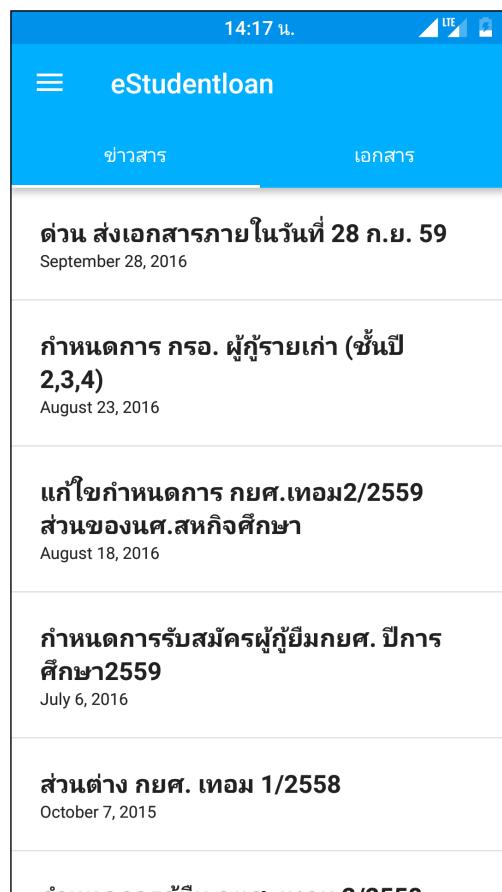
รูปที่ 2.20: หน้าแรกของเว็บไซต์ PSU Studentloan

ที่มา : <https://studentloan.psu.ac.th/home>

### 2.7.2 แอปพลิเคชัน XX

eStudentloan [10] เป็นแอปพลิเคชันที่พัฒนาโดยฝ่ายกิจการนักศึกษา สถาบันเทคโนโลยีไทย-ญี่ปุ่น เพื่อให้บริการนักศึกษาของทุนเงินให้กู้ยืมเพื่อการศึกษาสถาบันเทคโนโลยีไทย-ญี่ปุ่น สามารถดาวน์โหลดเพื่อใช้งานได้ที่ <https://play.google.com/store/apps/details?id=th.co.dest.anek.studentloan> โดยแอปพลิเคชันมีพังก์ชันการทำงาน คือ ติดตามข่าวสารจาก

หน่วยงาน



รูปที่ 2.21: ข่าวสารจากหน่วยงาน

ที่มา : <https://play.google.com/store/apps/details?id=th.co.dest.anek.studentloan>

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบก่อนดำเนินการจริงเป็นอีกหนึ่งขั้นตอนที่มีความสำคัญมาก เพราะการวิเคราะห์และออกแบบระบบนั้นเป็นการกระทำที่ทำให้ผู้พัฒนาเห็นรายละเอียดส่วนย่อยของงานทั้งหมด เพิ่มประสิทธิภาพในการวางแผน การทำงาน และยังช่วยลดปัญหาที่อาจจะเกิดขึ้นในระหว่างพัฒนา เพื่อให้ระบบมีความสมบูรณ์มากยิ่งขึ้น เนื่องจากการวิเคราะห์และออกแบบระบบนั้นจะช่วยให้บริการ จัดการทรัพยากรได้อย่างคุ้มค่าและตรงตามความต้องการของระบบ

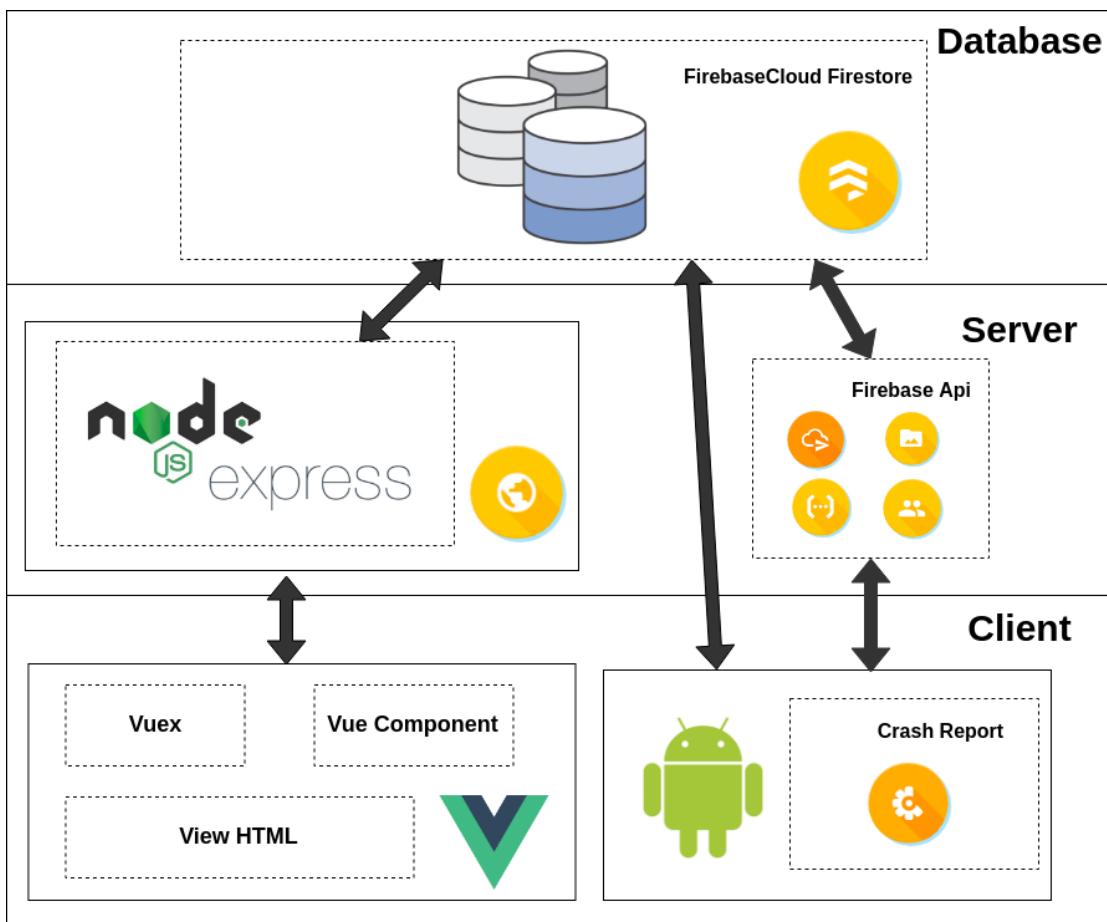
การวิเคราะห์และออกแบบระบบ XX ในบทนี้จะแบ่งออกเป็น X ขั้นตอนเพื่อให้เห็นการดำเนินงานอย่างมีระบบ ในหัวข้อแรกจะนำเสนอภาพรวมของระบบ ก่อนจะนำเสนอเอกสารแสดงความต้องการของระบบซึ่งจะทำให้เห็นถึงความต้องการของเพจต่าง ๆ ในขั้นตอนของการออกแบบในหัวข้อที่สาม ส่วนหัวข้อที่เหลือจะแสดงแผนภาพการทำงานของระบบโดยใช้ UML diagram ซึ่งประกอบไปด้วย Use Case, Class และ Sequence Diagram เพื่อแสดงรายละเอียดของระบบก่อนนำไปเขียนคำสั่งด้วยภาษาโปรแกรมในบทต่อไป

- 3.1 โครงสร้างภาพรวมของระบบ (System Architecture) เป็นการออกแบบภาพรวมและเทคโนโลยีของระบบ
- 3.2 System Requirements คือ ความต้องการหรือสิ่งที่ระบบควรจะทำ หรือหน้าที่หลักของระบบที่จะต้องทำ
- 3.3 User Interface Design เป็นการออกแบบส่วนต่อประสานกับผู้ใช้
- 3.4 Use Case Diagram เป็นแผนภาพที่ใช้แสดงให้ทราบว่าระบบทำงานหรือมีหน้าที่ใดบ้าง
- 3.5 Class Diagram เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class
- 3.6 Sequence Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นลักษณะการทำงานหรือขั้นตอนต่อเนื่องตามลำดับของเวลาที่เกิดเหตุการณ์จากน้อยไปมาก

### 3.1 โครงสร้างภาพรวมของระบบ

ความหมายของ System Architecture [11] หมายถึง ครอบคลุมโครงสร้างของระบบที่อธิบายความสัมพันธ์ขององค์ประกอบต่าง ๆ ไปจนถึงขั้นการเชื่อมต่อกันของระบบอย่างต่อเนื่อง โดยจัดกลุ่มองค์ประกอบไว้ในหลาย ๆ ลักษณะเพื่อให้ผู้เกี่ยวข้อง (Stakeholder) จากพื้นฐานสาขาอาชีพที่แตกต่าง กันสามารถทำความเข้าใจได้ง่าย เช่น การจัดแบ่งองค์ประกอบตามลักษณะการทำงานของระบบ (functional components) เป็นต้น

การออกแบบ System architecture แสดงภาพรวมและเทคโนโลยีของระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีรายละเอียดดังรูปที่ 3.1



รูปที่ 3.1: System architecture ระบบ XX

จากรูปที่ 3.1 สามารถอธิบายโครงสร้างและเทคโนโลยีของระบบโดยแบ่งเป็น 3 ส่วนหลัก ดังนี้

1. Database ระบบใช้บริการฐานข้อมูลแบบ NoSQL ของไฟร์เบสชื่อ Cloud Firestore
2. Server กระบวนการทำงานในส่วนของเซิฟเวอร์ (server) แบ่งเป็น 2 ส่วนได้แก่
  - ไฟร์เบส Hosting เป็นบริการสำหรับสร้างโฮสต์ (Host) สำหรับเก็บซอฟต์แวร์ (source code) และทำการเผยแพร่สำหรับการพัฒนาเว็บไซต์ ซึ่งในที่นี้ใช้ Node.js และ Express ในการพัฒนา
  - ชุดบริการไฟร์เบส Api ใช้สำหรับการทำงานกับบริการต่าง ๆ ของไฟร์เบสนั้นๆ เช่น Authentication ใช้สำหรับการจัดการข้อมูลผู้ใช้หรือไฟร์เบส Storage ที่ใช้สำหรับจัดเก็บไฟล์เอกสารและรูปภาพต่าง ๆ เป็นต้น
3. Client แอปพลิเคชันของระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี แบ่งเป็น 2 ส่วนตามการทำงานบนแพลตฟอร์ม ดังนี้
  - เว็บแอปพลิเคชันวัตถุประสงค์ในใช้งานเพื่อรับการทำงานของผู้ใช้งานบนบราวเซอร์โดยพัฒนาด้วย Vue.js
  - โมบายแอปพลิเคชันทำงานบนอุปกรณ์พกพา มีการใช้บริการ Crashlytics ของไฟร์เบสในการติดตามข้อผิดพลาดต่าง ๆ ที่เกิดขึ้น

## 3.2 System Requirements

### 3.2.1 Functional Requirements

ระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี แบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

1. เจ้าหน้าที่
  - สามารถทำการเข้าสู่ระบบได้
  - สามารถสร้างประกาศหรือประชาสัมพันธ์บนเว็บแอปพลิเคชันได้
  - สามารถกำหนดวันที่และช่วงเวลาในการส่งเอกสารการกู้ยืมของกองทุนเงินให้กู้ยืม เพื่อการศึกษาได้
  - สามารถตรวจสอบยืนยันความถูกต้องเอกสารกองทุนเงินให้กู้ยืมเพื่อการศึกษาจากนักศึกษาผ่านเว็บแอปพลิเคชันได้
  - สามารถทำการอัปโหลดฐานข้อมูลนักศึกษาของกองทุนเงินให้กู้ยืมเพื่อการศึกษาได้

- สามารถส่งข้อความเพื่อติดต่อนักศึกษาในระบบได้
- สามารถทำการอัปโหลดเอกสารกองทุนเงินให้กู้ยืมเพื่อการศึกษาที่เกี่ยวข้องเข้าสู่ฐานข้อมูลของระบบได้
- สามารถสร้างปฏิทินขั้นตอนการดำเนินการกองทุนเงินให้กู้ยืมเพื่อการศึกษาได้
- สามารถสร้างรายการคำตามที่พบบอยได้

## 2. นักศึกษา

- สามารถสมัครสมาชิกและเข้าสู่ระบบได้
- สามารถรับข้อมูลข่าวสารประชาสัมพันธ์ได้
- สามารถรับข้อมูลแจ้งเตือนเมื่อมีข่าวสารประชาสัมพันธ์ใหม่ ๆ ผ่านไปรษณีย์และแอปพลิเคชันได้
- สามารถดูปฏิทินกำหนดการการดำเนินงานกองทุนเงินให้กู้ยืมเพื่อการศึกษาได้
- สามารถส่งภาพสำเนาเอกสารกองทุนเงินให้กู้ยืมเพื่อการศึกษาเพื่อให้พนักงานตรวจสอบได้
- สามารถจองวันที่และเวลาในส่วนเอกสารกองทุนเงินให้กู้ยืมเพื่อการศึกษาฉบับจริงได้
- สามารถดาวน์โหลดเอกสารกองทุนเงินให้กู้ยืมเพื่อการศึกษาจากฐานข้อมูลของระบบได้
- สามารถส่งข้อความเพื่อติดต่อสอบถามกับเจ้าหน้าที่กองทุนเงินให้กู้ยืมเพื่อการศึกษาได้
- สามารถดูและแก้ไขข้อมูลส่วนตัวได้
- สามารถดูรายการคำตามที่พบบอยได้

### 3.2.2 Non-functional Requirements

#### 1. เว็บแอปพลิเคชัน

- ใช้โปรโตคอล (Protocol) แบบ HTTPS (Hypertext Transfer Protocol Secure) ใน การสื่อสารที่ช่วยรักษาความสมบูรณ์ถูกต้องของข้อมูลผู้ใช้และเก็บข้อมูลไว้เป็นความลับระหว่างคอมพิวเตอร์ของผู้ใช้กับเว็บไซต์
- รองรับการใช้งานของผู้ใช้พร้อมกันอย่างน้อย 100 คน

#### 2. แอ nondroydแอปพลิเคชัน

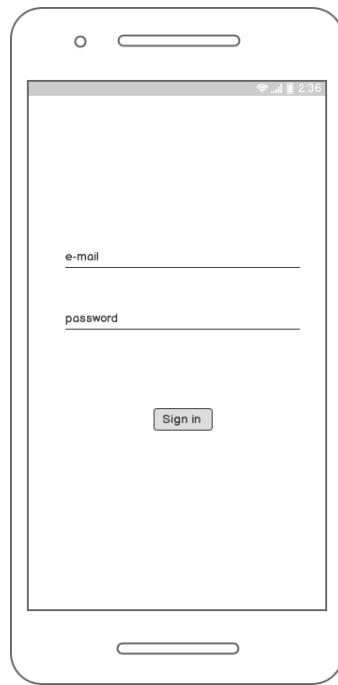
- การสอบถามข้อมูลและการบันทึกข้อมูลปลอดภัยโดยการใช้งาน Cloud FireStore
- รองรับการใช้งานของผู้ใช้พร้อมกันอย่างน้อย 100 คน

### 3.3 User Interface Design

ในการออกแบบ User Interface Design ของระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ใช้การออกแบบใน 2 ลักษณะ ได้แก่

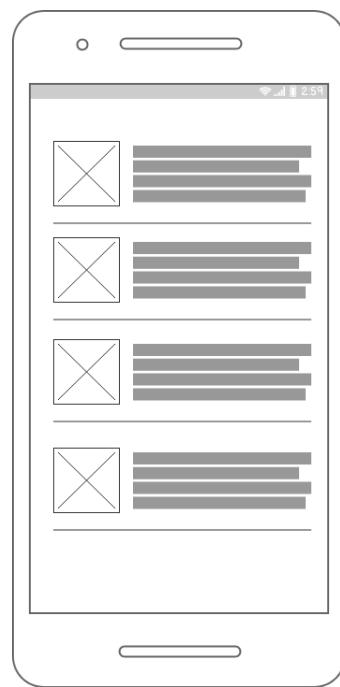
#### 1. Low-fidelity Wireframes

ใช้งาน Low-fidelity Wireframes สำหรับงานที่เน้นความรวดเร็ว สามารถสื่อสารเพื่อเข้าใจถึงการทำงานของระบบ ได้ ข้อดีคือ ได้ผลตอบรับอย่างรวดเร็ว สะดวกแก่การแก้ไขงาน แต่ข้อเสียคือ อาจจะได้รูปแบบการ Transition ระหว่างหน้าจอไม่ค่อยสมจริงเท่าไร



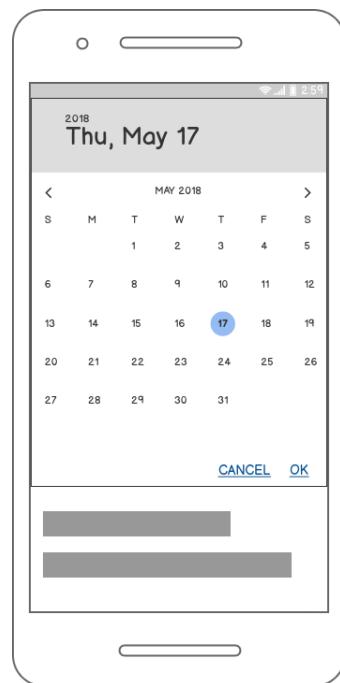
รูปที่ 3.2: หน้าจอเข้าสู่ระบบ

จากภาพที่ 3.2 แสดงหน้าจอสำหรับให้ผู้ใช้ทำการเข้าสู่ระบบ โดยจำเป็นต้องกรอกข้อมูล อีเมลและรหัสผ่านเพื่อใช้ในการเข้าสู่ระบบ



รูปที่ 3.3: หน้าจอข่าวสารประชาสัมพันธ์

จากภาพที่ 3.3 แสดงหน้าจอข่าวสารหรือประชาสัมพันธ์จากเจ้าหน้าที่หรือผู้ที่เกี่ยวข้องซึ่งแสดงข้อมูลและรายการข่าวสารหรือประชาสัมพันธ์



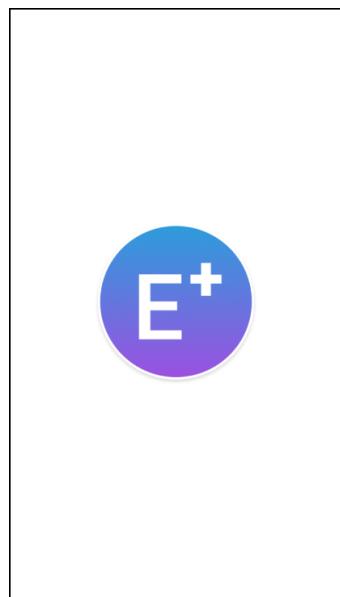
รูปที่ 3.4: หน้าจอปฏิทินกำหนดการ

จากภาพที่ 3.4 แสดงหน้าจอปฏิทินกำหนดการการดำเนินการเพื่อให้ผู้ใช้สามารถตรวจสอบกำหนดการวันที่และเวลาในการดำเนินงานของกองทุนโดยหน้าจอ มีการแสดงผลปฏิทินและรายละเอียดการดำเนินการ

## 2. High-fidelity wireframes

ใช้งาน High-fidelity wireframes สำหรับการนำเสนอไอเดีย (idea) หรือรูปแบบการ Action ให้แก่ Customer เสมือนงานจริงมากที่สุด ข้อดีคือ สามารถชี้นำการใช้งานจากหน้าหนึ่งไปยังอีกหนึ่งได้ดีด้วยการทำ Motion ระหว่างหน้ารวมทั้งสามารถทำ Interaction กับผู้ใช้งานซึ่งเป็นการสร้างการโต้ตอบการใช้งานกับผู้ใช้ได้ดี

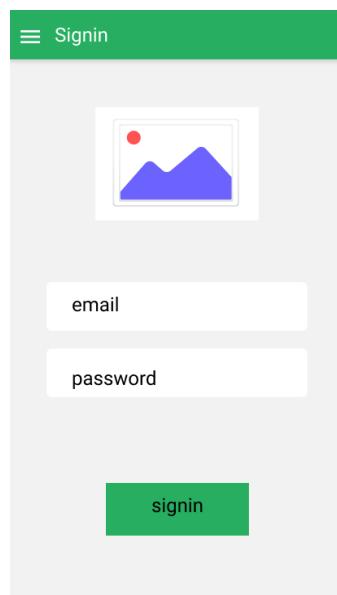
- ไม่บายแอปพลิเคชัน
  - การออกแบบหน้าจอ splash screen



รูปที่ 3.5: หน้าจอ splash screen

จากภาพที่ 3.5 แสดงหน้าจอ splash screen ใช้ในการแสดงทุกครั้งที่ผู้ใช้เปิดแอปพลิเคชันโดยวัตถุประสงค์การทำงานของหน้านี้คือเพื่อใช้แสดงขณะที่แอปพลิเคชันทำการประมวลผลข้อมูลบนพื้นหลัง (Background process) เช่น การตรวจสอบสถานะการเข้าสู่ระบบของผู้ใช้คนปัจจุบัน เป็นต้น

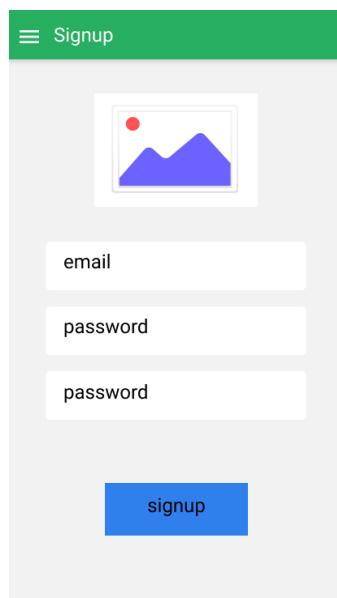
- การออกแบบหน้าจอเข้าสู่ระบบ



รูปที่ 3.6: หน้าจอเข้าสู่ระบบ

จากภาพที่ 3.6 แสดงหน้าจอสำหรับให้ผู้ใช้ทำการเข้าสู่ระบบเมื่อผู้ใช้งานไม่ได้ทำการเข้าสู่ระบบ โดยจำเป็นต้องกรอกข้อมูลอีเมลและรหัสผ่านเพื่อใช้ในการเข้าสู่ระบบ ซึ่งการเข้าสู่ระบบจะทำเพียงครั้งเดียวเท่านั้น เมื่อผู้ใช้เปิดการทำงานแอปพลิชันใหม่ในครั้งถัดไประบบจะระบุข้อมูลของผู้ใช้งานอัตโนมัติ

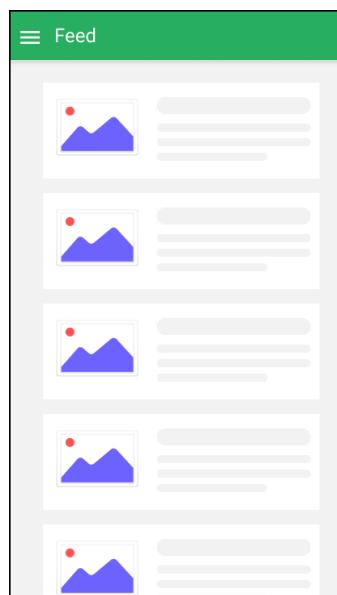
- การออกแบบหน้าจอสมัครสมาชิก



รูปที่ 3.7: หน้าจอสมัครสมาชิก

จากภาพที่ 3.7 แสดงหน้าจอสมัครสมาชิก หากผู้ใช้ยังไม่มีบัญชีในระบบผู้ใช้งานสามารถทำการสมัครสมาชิกเพื่อเข้าใช้งานระบบได้จากหน้าสมัครสมาชิก โดยผู้ใช้จำเป็นต้องกรอกข้อมูลอีเมล รหัสผ่านและรหัสนักศึกษาในการสมัครสมาชิก

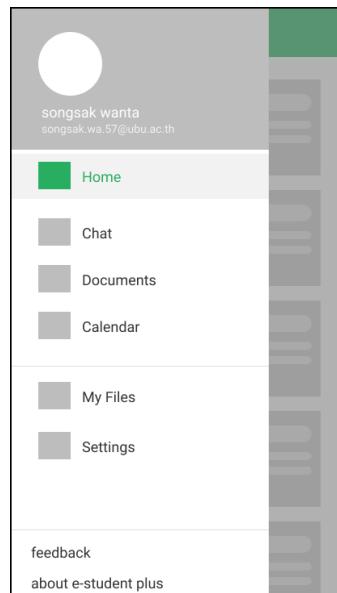
- การออกแบบหน้าจอข่าวสารและประชาสัมพันธ์



รูปที่ 3.8: หน้าจอข่าวสารและประชาสัมพันธ์

จากภาพที่ 3.8 แสดงหน้าจอข่าวสารหรือประชาสัมพันธ์จากเจ้าหน้าที่หรือผู้ที่เกี่ยวข้อง

- การอوكแบบหน้าจอเมนูนำทางหลัก



รูปที่ 3.9: หน้าจอเมนูนำทางหลัก

จากภาพที่ 3.9 แสดงเมนูนำทางหลักที่ใช้นำทางผู้ใช้งานไปยังหน้าจออื่นๆ ภายใน

แอปพลิเคชัน

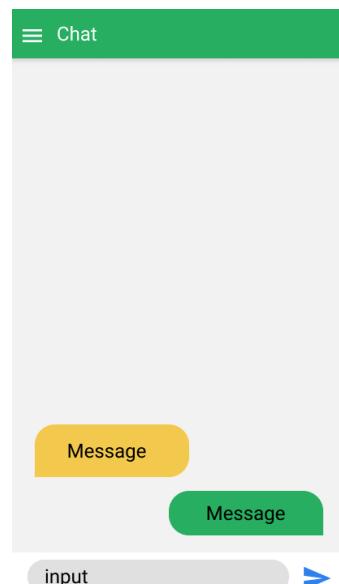
- การอุกแบบหน้าจอปฏิทินกำหนดการการดำเนินการ



รูปที่ 3.10: หน้าจอปฏิทินกำหนดการการดำเนินการ

จากภาพที่ 3.10 แสดงหน้าจอปฏิทินกำหนดการการดำเนินการเพื่อให้ผู้ใช้สามารถตรวจสอบกำหนดการวันที่และเวลาในการดำเนินงานของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี

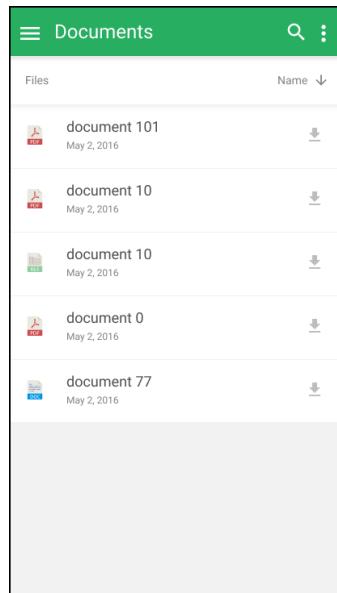
- การออกแบบหน้าจอสนทนากับผู้ใช้



รูปที่ 3.11: หน้าจอสนทนา

จากภาพที่ 3.11 นักศึกษาสามารถส่งข้อความไปยังเจ้าหน้าที่เพื่อติดต่อสอบถาม  
ข้อมูลกับทางเจ้าหน้าที่ได้โดยตรง

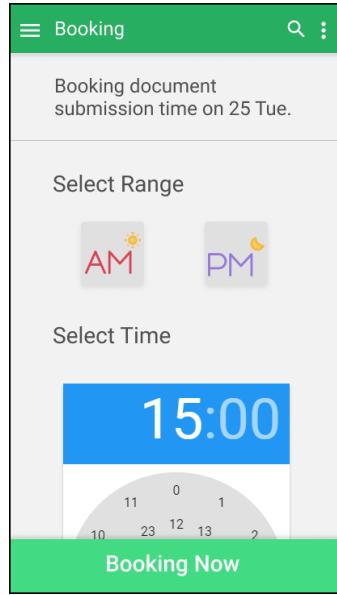
- การอุปแบบหน้าจอเอกสารที่เกี่ยวข้อง



รูปที่ 3.12: หน้าจอเอกสารที่เกี่ยวข้อง

จากภาพที่ 3.12 นักศึกษาสามารถดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนเงิน  
ให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี เช่น ข้อทำ-  
หนดและคุณสมบัติของผู้กู้ยืม เป็นต้น ซึ่งเอกสารได้ถูกอัพโหลดไว้โดยเจ้าหน้าที่

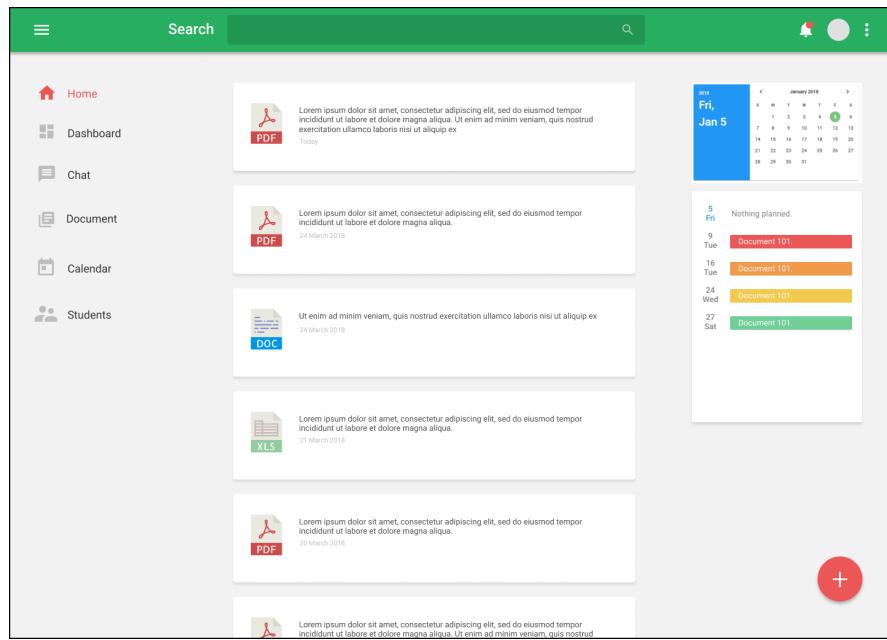
- การอุปแบบหน้าจอของวันที่และเวลาส่งเอกสาร



รูปที่ 3.13: หน้าจอของวันที่และเวลาส่งเอกสาร

จากภาพที่ 3.13 เมื่อเจ้าหน้าที่ทำการเพิ่มวันที่และช่วงเวลาในการส่งเอกสาร นักศึกษาสามารถจองวันที่และเวลาในการส่งเอกสารฉบับจริงของตนได้จากหน้าจอตั้งกล่าวโดยมีเงื่อนไขคือนักศึกษาผู้ที่จะทำการส่งเอกสารฉบับจริงจำเป็นต้องลงทะเบียนสำหรับผ่านทางระบบเพื่อให้เจ้าหน้าที่ยืนยันความถูกต้องเสียก่อน

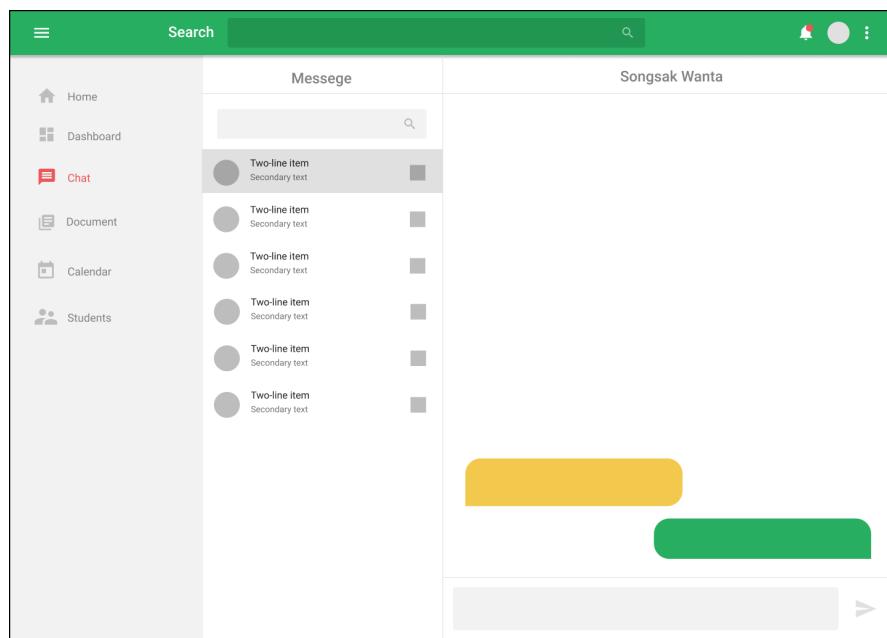
- เว็บแอปพลิเคชัน
  - การออกแบบหน้าจอหลัก



รูปที่ 3.14: หน้าจอหลัก

จากภาพที่ 4.6 แสดงหน้าจอหลักบนเว็บแอปพลิเคชัน เพื่ออำนวยความสะดวก ต่อเจ้าหน้าที่ ในหน้าหลักได้รวมรวมข้อมูลสรุปและเมนูเข้าถึงด่วนซึ่งแบ่งเป็น 3 ส่วนหลักได้แก่ เมนูนำทาง ข่าวสารประชาสัมพันธ์และปฏิทินกำหนดการ

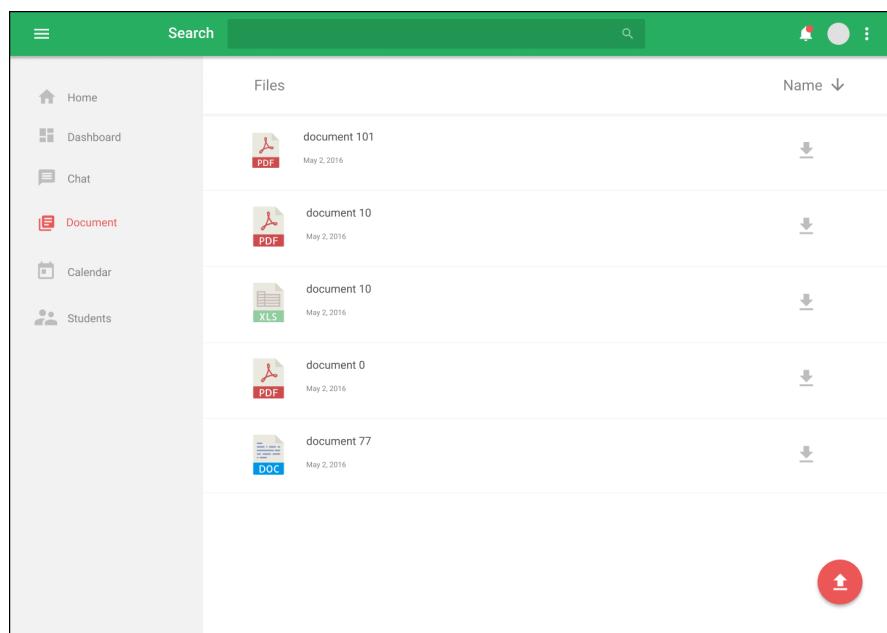
- การออกแบบหน้าจอสนทนา



รูปที่ 3.15: หน้าจอสนทนา

จากภาพที่ 3.15 แสดงหน้าจอสนทนา มีการแสดงรายชื่อนักศึกษาและส่วนของห้องสนทนาด้วย

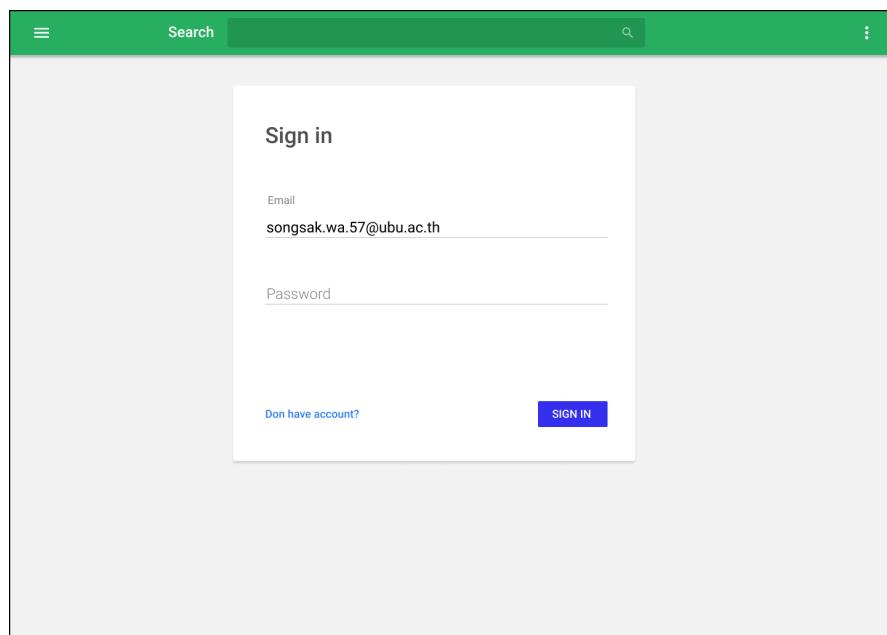
- การออกแบบหน้าจออัปโหลดเอกสาร



รูปที่ 3.16: หน้าจออัปโหลดเอกสาร

จากภาพที่ 3.16 แสดงหน้าจออัพโหลดเอกสารที่เกี่ยวข้องกับกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ทั้งนี้ผู้ที่มีสิทธิ์ในการอัพโหลดเอกสารมีเพียงเจ้าหน้าที่เท่านั้น

- การอุปโหลดแบบหน้าจอเข้าสู่ระบบ



รูปที่ 3.17: หน้าจอเข้าสู่ระบบ

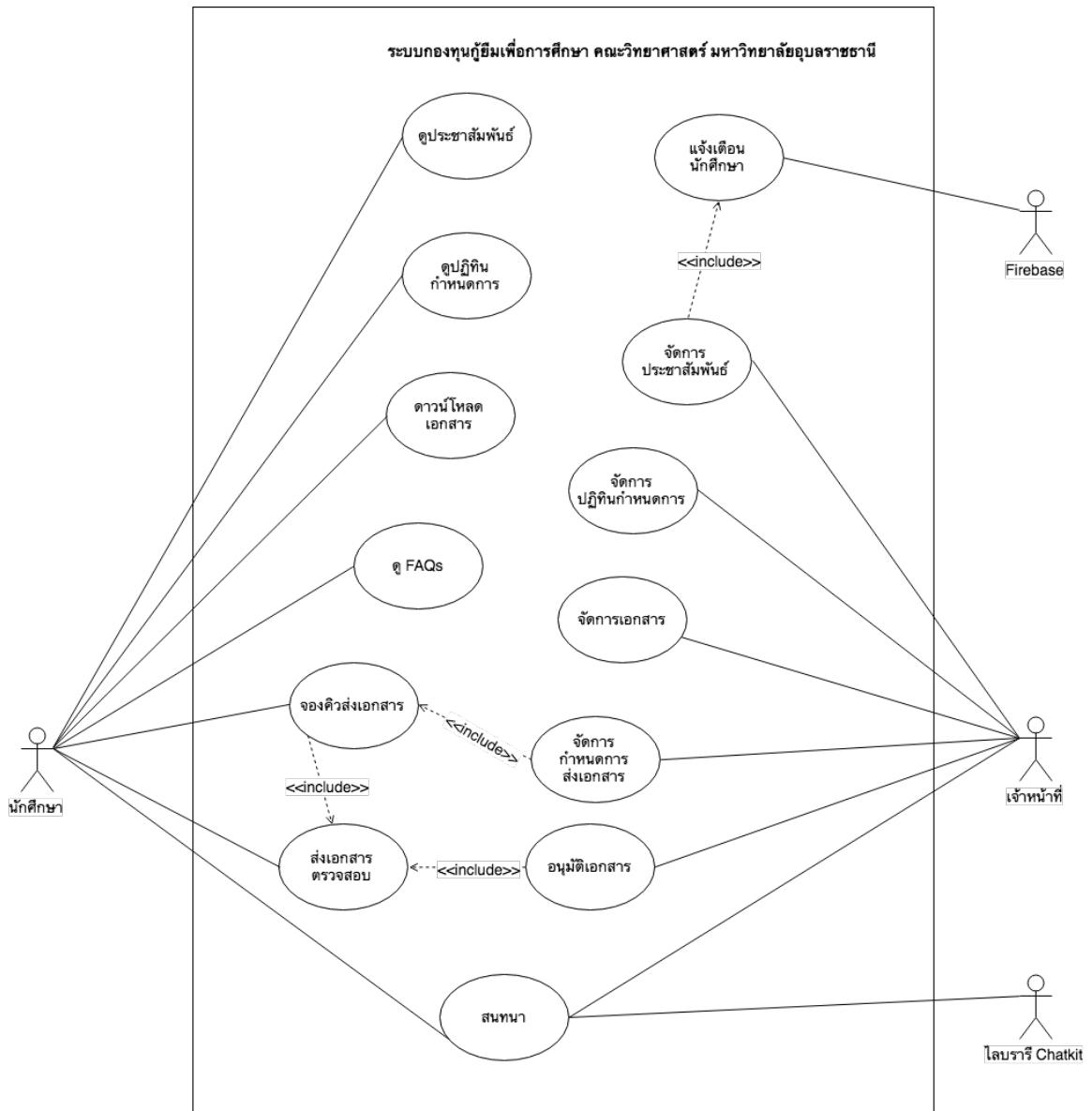
จากภาพที่ 3.17 แสดงหน้าจากการเข้าสู่ระบบของผู้ใช้โดยผู้ใช้จำเป็นต้องกรอกข้อมูลอีเมลและรหัสผ่านเพื่อเข้าใช้งานระบบ

### 3.4 Use Case Diagram

Use Case Diagram เป็นแผนผังเพื่อแสดงฟังก์ชันและถูกการทำงานของระบบโดยรวม แสดงส่วนประกอบในระบบและกิจกรรมที่เกิดขึ้นในระบบซึ่งในระบบของทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ผู้ใช้จำเป็นต้องเข้าสู่ระบบเพื่อใช้งานระบบ สัญลักษณ์ที่ใช้ในการเขียน Use Case Diagram แสดงในตารางที่ 3.1

ตารางที่ 3.1: สัญลักษณ์ของ Use case Diagram

สัญลักษณ์	การใช้งาน
Use case	Use case คือส่วนย่อยของระบบงาน แทนด้วยวงรีและชื่อของ Use case ภายในวงรี
	Actor คือบุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากการซึ่งอยู่ภายนอกระบบ แทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ
	เส้นตรงที่แสดงถึงการใช้งาน Use case ของผู้กระทำ
	กรอบ สีเหลี่ยม แสดงถึงขอบเขตของระบบ โดยแสดงชื่อระบบภายในหรือด้านบนกรอบสีเหลี่ยม Use case อยู่ภายในกรอบสีเหลี่ยม และ actor อยู่ภายนอกกรอบสีเหลี่ยม
	ความสัมพันธ์แบบ «includes» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประซึ่ง Use case ที่ทางลูกศรเรียกใช้งาน Use case ที่หัวลูกศรทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ «extend» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประซึ่ง Use case ที่หัวลูกศรเรียกใช้งาน Use case ที่ทางลูกศรแต่การใช้งานไม่จำเป็นต้องเกิดขึ้นทุกครั้งที่มีการเชื่อมโยงกับเจือนี้ระหว่างการทำงาน



รูปที่ 3.18: Use Case Diagram ของระบบ XX

ตารางที่ 3.2: อธิบาย Use Case หน้าที่ของระบบ ในภาพที่ 3.18

Use Case	คำอธิบาย
ดูประชาสัมพันธ์	นักศึกษาสามารถดูประชาสัมพันธ์ได้โดยไม่จำเป็นต้องทำการเข้าสู่ระบบก่อน
ดูปฏิทินกำหนดการ	นักศึกษาสามารถเปิดดูปฏิทินวันที่และเวลากำหนดการของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์วิทยาลัยอุบราชธานี โดยไม่จำเป็นต้องทำการเข้าสู่ระบบก่อน
ดาวน์โหลดเอกสาร	นักศึกษาดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนกู้ยืมการศึกษา คณะวิทยาศาสตร์มหาวิทยาลัย อุบราชธานีได้โดยไม่จำเป็นต้องทำการเข้าสู่ระบบก่อน
ดู FAQs	หน้าแสดงรายการคำถามที่พบบ่อย
จองคิวส่งเอกสาร	เมื่อนักศึกษาเข้าสู่ระบบเรียบร้อยแล้วนักศึกษา สามารถจองคิววันที่และเวลาในการส่งเอกสารฉบับจริงหลังจากที่ได้รับการตรวจสอบโดยเจ้าหน้าที่เรียบร้อยแล้ว
ส่งเอกสารตรวจสอบ	นักศึกษาสามารถส่งเอกสารหน้าที่ตรวจสอบโดยการถ่ายรูปแล้วทำการอัปโหลดเข้าสู่ระบบ เมื่อเจ้าหน้าที่ตรวจสอบเรียบร้อยแล้ว จะยืนยันสถานะของเอกสารอีกที
สนทนากลุ่ม	นักศึกษาสามารถสอบถามข้อมูลของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบราชธานี ได้จากช่องสนทนากลุ่ม
จัดการกำหนดการส่งเอกสาร	ใช้สำหรับเจ้าหน้าที่เพื่อเพิ่ม แก้ไขหรือลบกำหนดการส่งเอกสารของนักศึกษา

ตารางที่ 3.3: อธิบาย Use Case หน้าที่ของระบบ(ต่อ) ในภาพที่ 3.3

<b>Use Case</b>	<b>คำอธิบาย</b>
จัดการประชาสัมพันธ์	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบประชาสัมพันธ์
จัดการปฏิทินกำหนดการ	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบกำหนดการการดำเนินการของกองทุน
จัดการเอกสาร	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบเอกสารที่เกี่ยวข้องของกองทุน
แจ้งเตือนนักศึกษา	ใช้เพื่อส่งแจ้งเตือนไปยังนักศึกษาเมื่อมีการเพิ่มประชาสัมพันธ์โดยเจ้าหน้าที่

ตารางที่ 3.4: Use Case ดูประชาสัมพันธ์

<b>Use Case Title : ดูประชาสัมพันธ์</b>	<b>Use case Id : 1</b>
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาดูประชาสัมพันธ์โดยไม่จำเป็นต้องเข้าสู่ระบบ	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เขื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูประชาสัมพันธ์ได้	

ตารางที่ 3.5: Use Case ดูปฏิทินกำหนดการ

<b>Use Case Title : ดูปฏิทินกำหนดการ</b>	<b>Use case Id : 2</b>
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาสามารถเปิดดูปฏิทินวันที่และเวลากำหนดการของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบราชธานี ไม่จำเป็นต้องเข้าสู่ระบบ	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เขื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูปฏิทินกำหนดการได้	

ตารางที่ 3.6: Use Case ดาวน์โหลดเอกสาร

<b>Use Case Title :</b> ดาวน์โหลดเอกสาร	<b>Use case Id :</b> 3
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนกู้ยืมการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานีได้ไม่จำกัดจำนวนครั้ง	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้มต่ออินเทอร์เน็ต จะไม่สามารถดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนได้	
Exceptional Flow ที่ 2 : หากผู้ใช้ไม่มายแอปพลิเคชันไม่เปิดสิทธิ์การอ่านและเขียนไฟล์บันความจำสำรอง จะไม่สามารถดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนได้	

ตารางที่ 3.7: Use Case ดู FAQs

<b>Use Case Title :</b> ดู FAQs	<b>Use case Id :</b> 4
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : ใช้เพื่อแสดงรายการคำถามที่พบบ่อย	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้มต่ออินเทอร์เน็ต จะไม่สามารถดูคำถามที่พบบ่อยได้	

ตารางที่ 3.8: Use Case จองคิวส่งเอกสาร

<b>Use Case Title :</b> จองคิวส่งเอกสาร	<b>Use case Id :</b> 5
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : เมื่อนักศึกษาเข้าสู่ระบบเรียบร้อยแล้วนักศึกษา สามารถจองคิววันที่และเวลาในการส่งเอกสารฉบับจริงหลังจากที่ได้รับการตรวจสอบโดยเจ้าหน้าที่เรียบร้อยแล้ว	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เข้มต่ออินเทอร์เน็ต จะไม่สามารถจองคิวส่งเอกสารได้	

ตารางที่ 3.9: Use Case ส่งเอกสารตรวจสอบ

<b>Use Case Title :</b> ส่งเอกสารตรวจสอบ	<b>Use case Id :</b> 6
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาระบุส่วนตัวที่ต้องการส่งเอกสาร ระบุรายละเอียด เช่น ชื่อ นามสกุล วันเดือนปีเกิด สถานะ รหัสนักศึกษา แล้วคลิกปุ่มส่ง ระบบจะส่งเอกสารไปยังอีเมลของผู้รับ	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถส่งเอกสารตรวจสอบได้	
Exceptional Flow ที่ 2 : หากผู้ใช้ไม่มีบัญชีอีเมล หรือบัญชีไม่ถูกต้อง ระบบจะแจ้งเตือนผู้ใช้	

ตารางที่ 3.10: Use Case สนทนากลุ่ม

<b>Use Case Title :</b> สนทนากลุ่ม	<b>Use case Id :</b> 7
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : เมื่อนักศึกษาเข้าสู่ระบบแล้วจะสามารถสอบถามข้อมูลของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณานักศึกษาสามารถตอบกลับได้	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถสนทนากลุ่มได้	

ตารางที่ 3.11: Use Case จัดการกำหนดการส่งเอกสาร

<b>Use Case Title :</b> จัดการกำหนดการส่งเอกสาร	<b>Use case Id :</b> 10
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่เพิ่ม แก้ไขหรือลบกำหนดการส่งเอกสารของนักศึกษา	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถเพิ่ม แก้ไขหรือลบกำหนดการส่งเอกสารของนักศึกษาได้	

ตารางที่ 3.12: Use Case จัดการประชาสัมพันธ์

<b>Use Case Title :</b> จัดการประชาสัมพันธ์	<b>Use case Id :</b> 11
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบปฏิทินกำหนดการในระบบได้	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบ ปฏิทินกำหนดการได้	

ตารางที่ 3.13: Use Case จัดการปฏิทินกำหนดการ

<b>Use Case Title :</b> จัดการปฏิทินกำหนดการ	<b>Use case Id :</b> 12
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบปฏิทินกำหนดการได้	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบ ปฏิทินกำหนดการได้	

ตารางที่ 3.14: Use Case จัดการเอกสาร

<b>Use Case Title :</b> จัดการเอกสาร	<b>Use case Id :</b> 13
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบเอกสารที่เกี่ยวข้องกับกองทุนได้	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบ เอกสารที่เกี่ยวข้องกับกองทุนได้	

ตารางที่ 3.15: Use Case แจ้งเตือนนักศึกษา

<b>Use Case Title :</b> แจ้งเตือนนักศึกษา	<b>Use case Id :</b> 14
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : แจ้งเตือนไปยังนักศึกษาเมื่อมีการเพิ่มประชาสัมพันธ์โดยเจ้าหน้าที่	
Exceptional Flow ที่ 1 : หากนักศึกษาที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถรับแจ้งเตือนเมื่อมีการเพิ่มประชาสัมพันธ์โดยเจ้าหน้าที่	

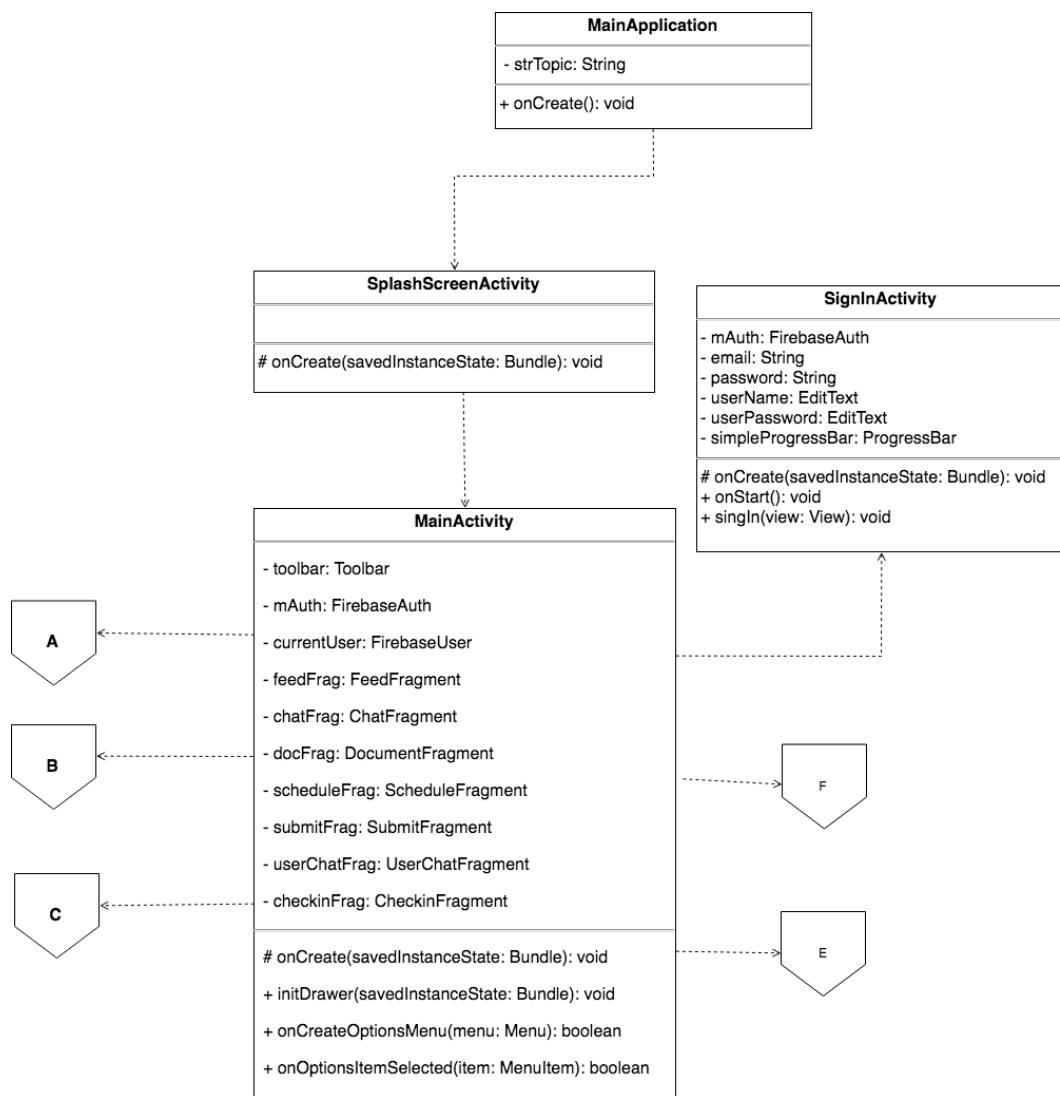
### 3.5 Class Diagram

Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแบบต่างๆ ระหว่างคลาส สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่ 3.16

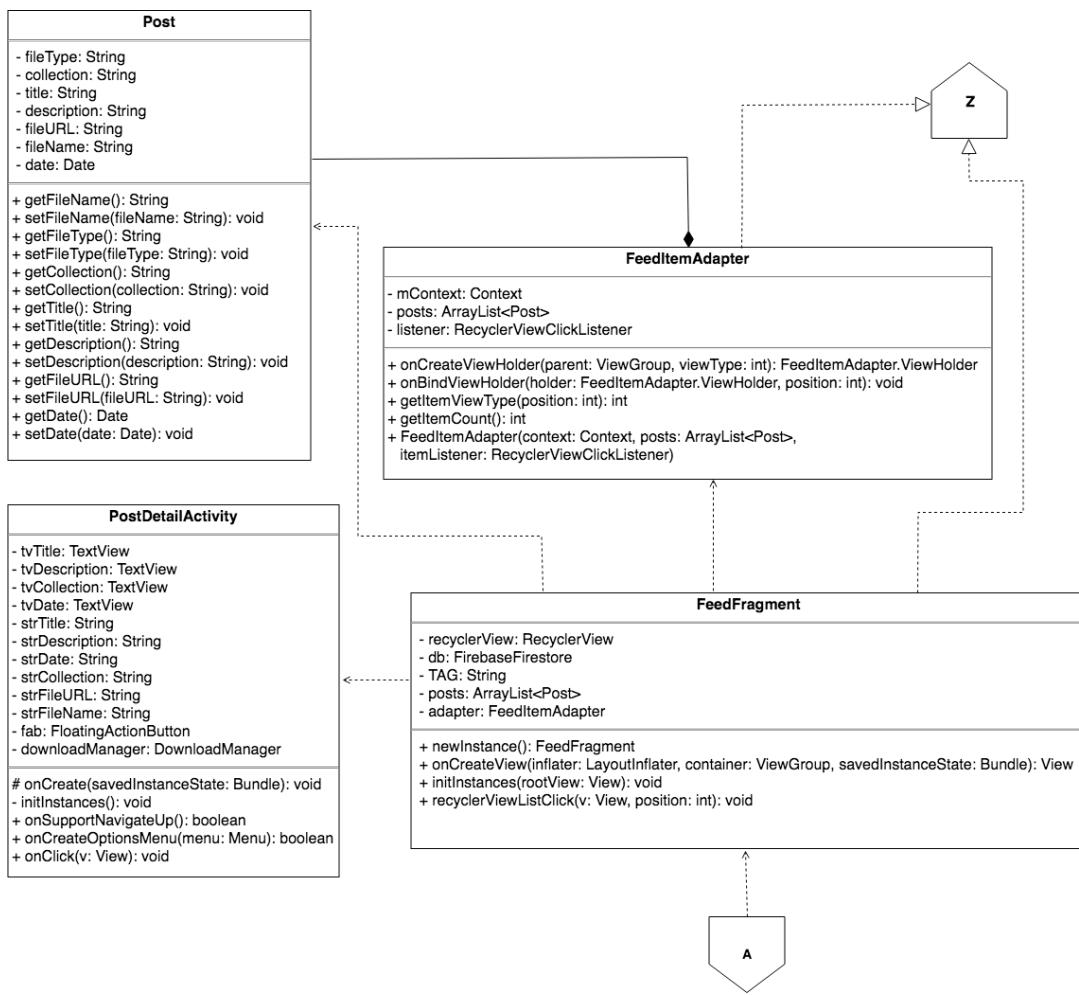
ตารางที่ 3.16: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	การใช้งาน
<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <b>Class Name</b>  <b>Attribute Name</b>  <b>Operation Name()</b> </div>	<p>คลาส สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบน เป็นชื่อของ class ส่วนกลาง เป็นชื่อ Attribute และ ส่วนล่าง เป็น Operation Name หรือ Method ใช้สำหรับเขียนฟังก์ชันในการทำงานของคลาสนั้น ๆ ชนิดของ Visibility ของ Method และ Attribute แบ่งเป็น 3 ชนิด ได้แก่</p> <ol style="list-style-type: none"> <li>1. Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+)</li> <li>2. Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-)</li> <li>3. Protected แทนสัญลักษณ์ด้วยเครื่องหมายชาร์ป ()</li> </ol>
	Dependency Relationship หมายความว่า คลาสที่อยู่ผู้ตั้งกฎศร สามารถเรียกใช้คลาสที่อยู่ผู้หัวกฎศร
	Composition Relationship เป็นความสัมพันธ์ระหว่างขอบเจ็ต หรือคลาสแบบขึ้นต่อ กันและมีความเกี่ยวข้องกันเสมอ
	Realization Relationship เป็นความสัมพันธ์ระหว่าง Object หรือ Class ในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Subclass)
	Connector เป็นสัญลักษณ์แทนด้วยรูปห้าเหลี่ยมและมีชื่ออุปทรง กลาง จะสร้างสัญลักษณ์นี้ไว้เมื่อต้องการเขียนต่อคลาสที่อยู่คนละหน้า

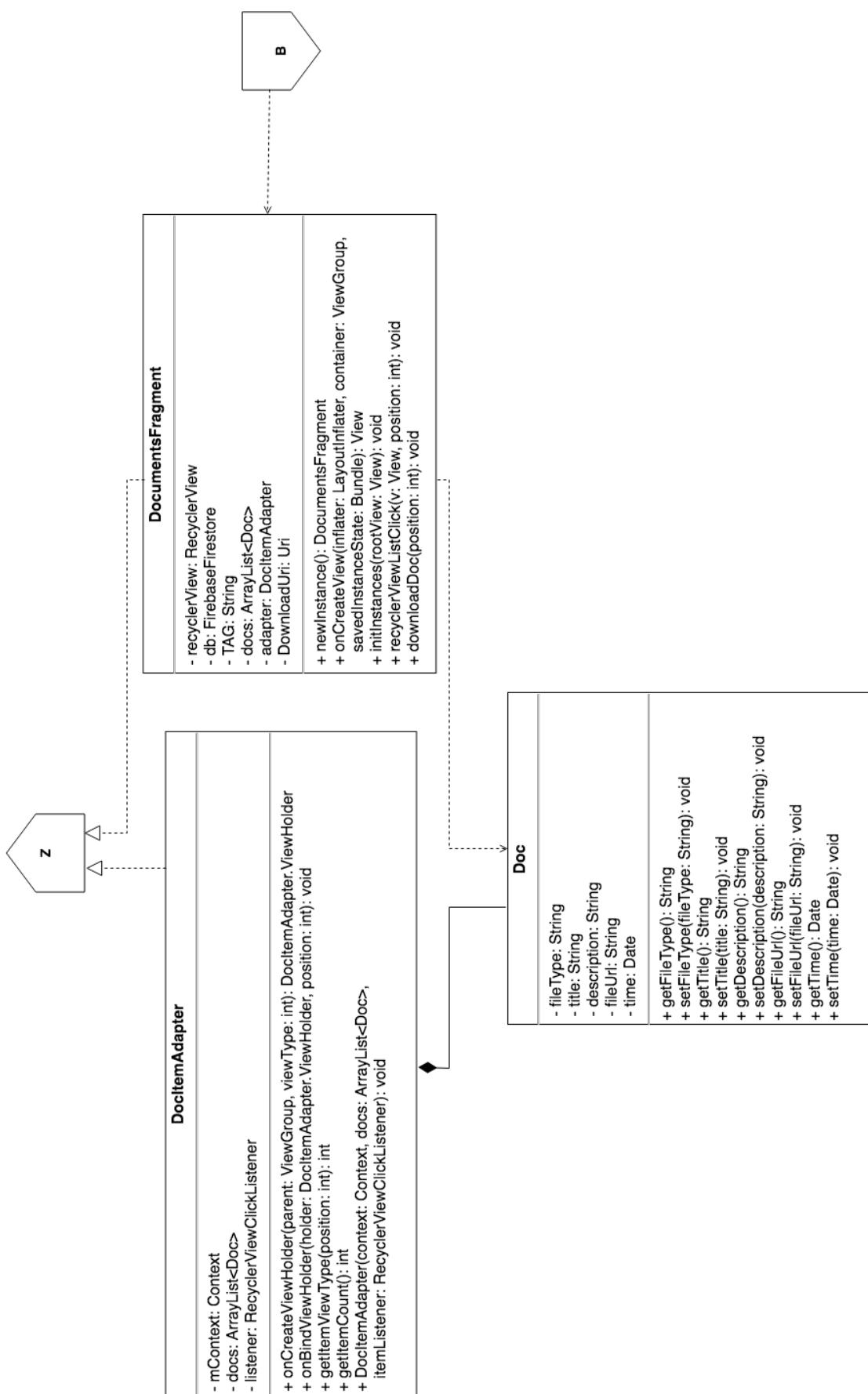
Class Diagram แสดงความสัมพันธ์ในรูปแบบต่างๆ ระหว่างคลาสของแอปพลิเคชันระบบ กองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี อธิบายได้ตามภาพ ที่ 3.19 ดังต่อไปนี้



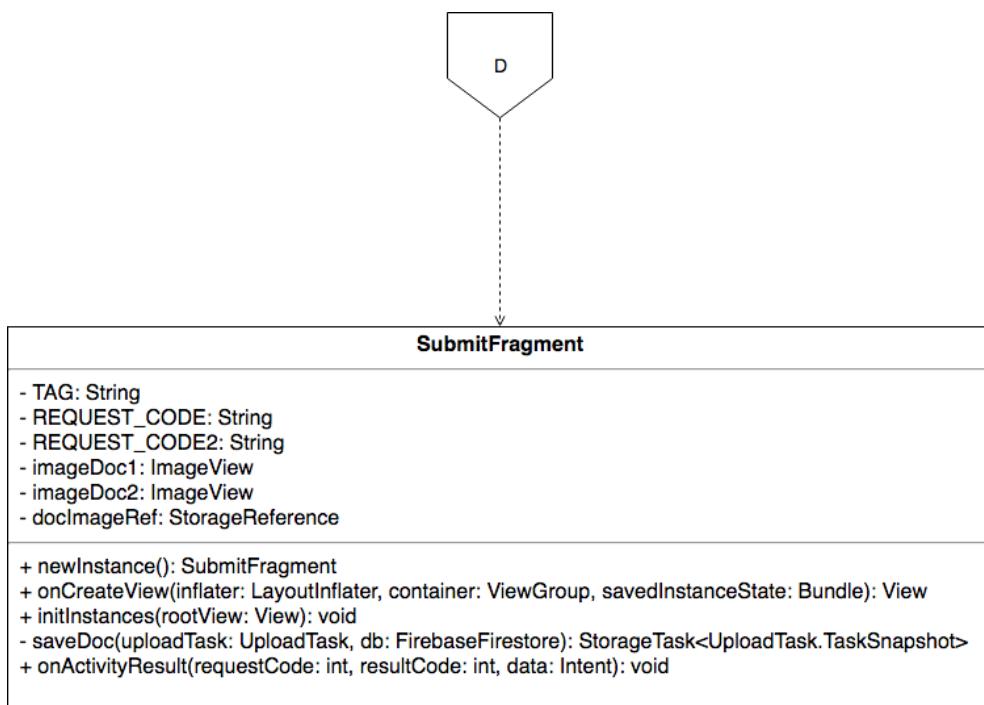
รูปที่ 3.19: Class Diagram ของแอปพลิเคชันระบบ XX



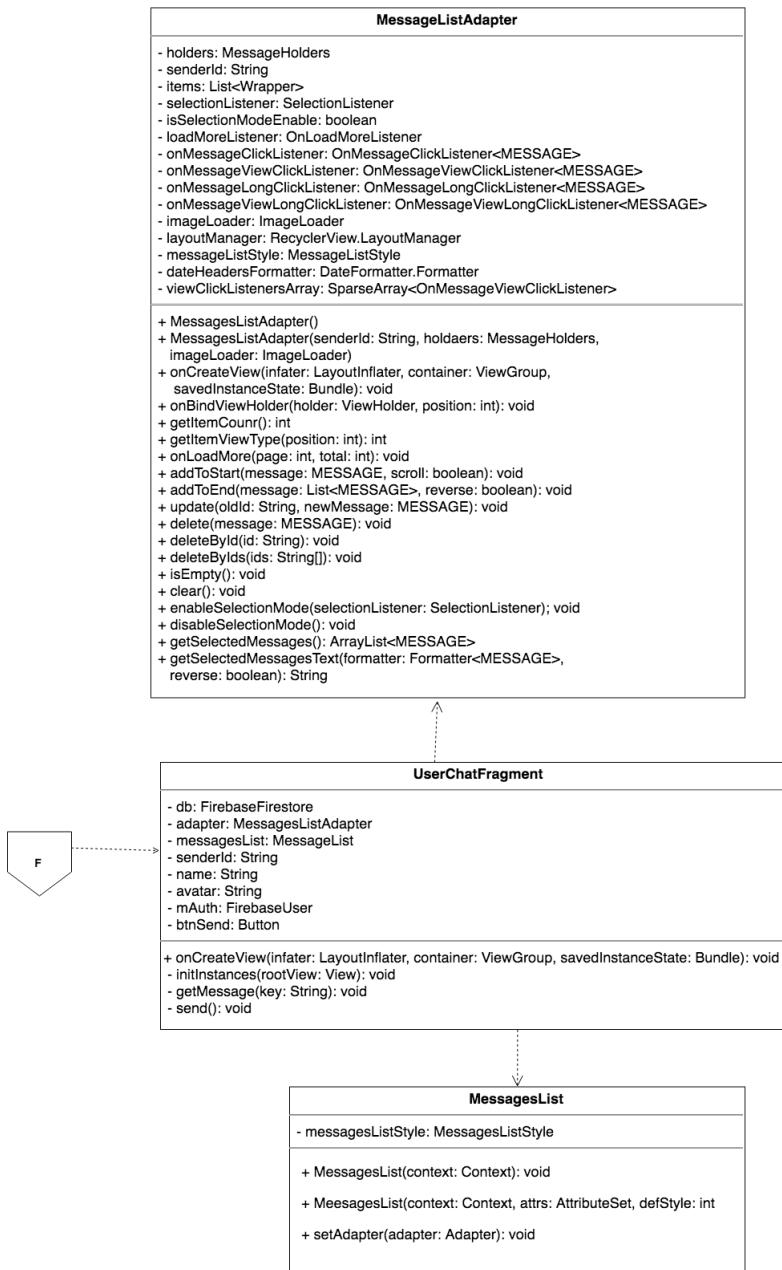
รูปที่ 3.20: Class Diagram ของแอปพลิเคชันระบบ XX



រូបទំនាក់ទំនង 3.21: Class Diagram ទូទៅពីគម្រោងបណ្តុះបណ្តាល XX



รูปที่ 3.22: Class Diagram ของแอปพลิเคชันระบบ XX



รูปที่ 3.23: Class Diagram ของแอปพลิเคชันระบบ XX

จากรูปภาพที่ 3.19 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.17: อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ

Class Diagram	คำอธิบาย
MainApplication	คลาส MainApplication จะถูกเรียกใช้งานทุกครั้ง เมื่อผู้ใช้เปิดแอปพลิเคชัน โดยวัตถุประสงค์การทำงานของคลาสนี้คือ เพื่อใช้จัดการทรัพยากรที่จำเป็นสำหรับการใช้งานในคลาสนี้ ๆ
SplashScreenActivity	คลาส SplashScreenActivity จะถูกเรียกใช้งานทุกครั้ง เมื่อผู้ใช้เปิดแอปพลิเคชัน โดยวัตถุประสงค์การทำงานของคลาสนี้คือ เพื่อใช้ตรวจสอบสถานะการเข้าสู่ระบบของผู้ใช้
MainActivity	คลาส MainActivity เป็นคลาสหลักที่ใช้ในการทำงานของแอปพลิเคชันโดยการทำงานของคลาสนี้เน้นไปที่การสร้าง Fragment เพื่อแสดงข้อมูลต่าง ๆ โดยองค์ประกอบการทำงานของคลาสนี้ประกอบไปด้วยสองส่วนหลักๆ ได้แก่ เมนูนำทาง Drawer และ Fragment Container
SignInActivity	คลาส SignInActivity เป็นคลาสที่ใช้เพื่อให้สมาชิกที่ได้ลงทะเบียน กับระบบเข้าสู่ระบบเพื่อใช้งานบริการต่าง ๆ จากระบบ

จากรูปภาพที่ 3.20 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.18: อธิบาย Class Diagram ของส่วนของการแสดงข่าวสาร

Class Diagram	คำอธิบาย
FeedFragment	คลาส FeedFragment เป็นคลาสหลักที่ใช้ในการแสดงข้อมูลข่าวสาร มีการทำงานหลักคือสืบค้นฐานข้อมูลจากไฟร์เบสเพื่อนำมาแสดง
FeedItemAdapter	คลาส FeedItemAdapter เป็นคลาสที่มีหน้าที่ในการแปลงชุดข้อมูลที่ได้จากคลาส FeedFragment แล้วคืนค่ากลับเป็นลิสต์รายการของชุดข้อมูลนั้น ๆ
Post	คลาส Post เป็นคลาสโมเดลที่กำหนดค่าต่างๆที่จำเป็นสำหรับใช้ในการสร้างลิสต์รายการของคลาส FeedItemAdapter
PostDetailActivity	คลาส PostDetailActivity เป็นคลาสที่มีหน้าที่ในการแสดงข้อมูลรายละเอียดของข่าวสาร แต่ละ แ Quinn ที่ได้รับ จากหน้า FeedFragment ที่จะส่งข้อมูลเมื่อผู้ใช้กดที่ແ夸รายการข่าวสาร
RecyclerViewClickListener	คลาส RecyclerViewClickListener เป็นคลาส อินเทอร์เฟส(Interface)ที่ใช้ในการสร้างแม่แบบเมื่อคลาสใด ๆ ต้องการใช้งานสำหรับการรับค่าเมื่อผู้ใช้กดແ夸ในลิสต์รายการ คลาสลูกที่ทำการสืบทอดคุณสมบัติจะสามารถรับข้อมูลตำแหน่งແ夸ที่ผู้ใช้กดบนลิสต์รายการได้

จากรูปภาพที่ 3.21 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.19: อธิบาย Class Diagram ของส่วนของการแสดงรายการเอกสารในระบบ

Class Diagram	คำอธิบาย
DocumentsFragment	คลาส DocumentsFragment เป็นคลาสที่ใช้ในการแสดงข้อมูลเอกสารที่ถูกอัพโหลดเข้าสู่ระบบโดยเจ้าหน้าที่ซึ่งจะถูกแสดงเป็นลิสต์รายการ
DocItemAdapter	คลาส DocItemAdapter เป็นคลาสที่มีหน้าที่ในการแปลงชุดข้อมูลที่ได้รับจากคลาส DocumentsFragment เป็นลิสต์รายการแล้วคืนกลับไปยังคลาส DocumentsFragment
Doc	คลาส Doc เป็นคลาสมोเดลที่กำหนดค่าต่าง ๆ ที่จำเป็นสำหรับใช้ในการสร้างลิสต์รายการของคลาส DocItemAdapter
RecyclerViewClickListener	คลาส RecyclerViewClickListener เป็นคลาสอินเทอร์เฟสที่ใช้ในการสร้างแม่แบบเมื่อคลาสใด ๆ ต้องการใช้งานสำหรับการรับค่าเมื่อผู้ใช้กดแฉวในลิสต์รายการ คลาஸลูกที่ทำการสืบทอดคุณสมบัติจะสามารถรับข้อมูลตำแหน่งแฉวที่ผู้ใช้กดบนลิสต์รายการได้

จากรูปภาพที่ 3.22 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.20: อธิบาย Class Diagram ของส่วนของการส่งสำเนาเอกสาร

Class Diagram	คำอธิบาย
SubmitFragment	คลาส SubmitFragment เป็นคลาสที่ใช้ในการแสดงหน้าจอส่งสำเนาเอกสาร โดยมีการดำเนินการภายในคลาสหลัก ๆ ได้แก่ การถ่ายภาพ แปลงภาพและบันทึกภาพเข้าสู่ระบบ

จากรูปภาพที่ 3.23 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

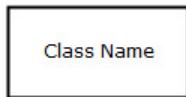
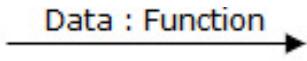
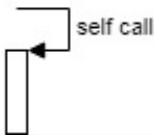
ตารางที่ 3.21: อธิบาย Class Diagram ของส่วนของการสนทนา

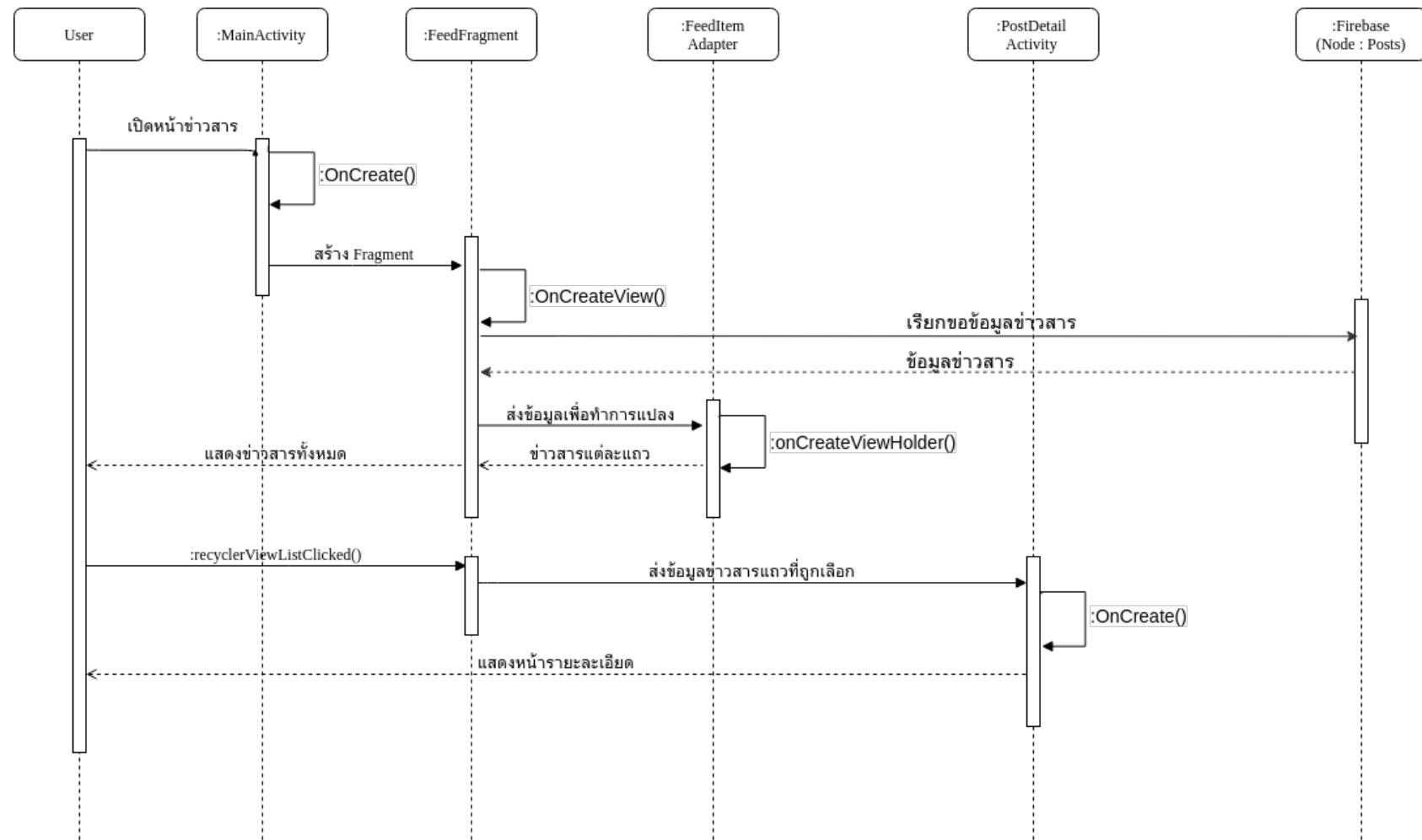
Class Diagram	คำอธิบาย
UserChatFragment	คลาส UserChatFragment เป็นคลาสที่ใช้ในการแสดงหน้าจอสนทนาสำหรับนักศึกษา เพื่อติดต่อสอบถามข้อมูลกับเจ้าหน้าที่ มีการสืบค้นข้อมูลประวัติการสนทนาเพื่อส่งไปแปลงเป็นข้อมูลลิสตรายการที่คลาส MessagesListAdapter
MessagesListAdapter	คลาส MessagesListAdapter เป็นคลาสที่ใช้ในการแปลงชุดข้อมูลที่ได้รับจากคลาส UserChatFragment เป็นลิสตรายการแล้วทำการคืนค่าลิสตรายการที่ได้กลับไปยังคลาส UserChatFragment
MessagesList	คลาส MessagesList เป็นคลาสที่ใช้ในการจัดเก็บข้อมูลภายในคลาส UserChatFragment หลังจากที่ได้ทำการสืบค้นข้อมูลการสนทนาจากไฟร์เบสเพื่อส่งไปยังคลาส MessagesListAdapter
RecyclerViewClickListener	คลาส RecyclerViewClickListener เป็นคลาสอินเทอร์เฟสที่ใช้ในการสร้างแม่แบบเมื่อคลาสใด ๆ ต้องการใช้งานสำหรับการรับค่าเมื่อผู้ใช้กดแควร์ในลิสตรายการ คลาสลูกที่ทำการสืบทอดคุณสมบัติจะสามารถรับข้อมูลตำแหน่งแควร์ที่ผู้ใช้กดบนลิสตรายการได้

### 3.6 Sequence Diagram

Sequence Diagram เป็น Diagram ที่แสดงขั้นตอนการทำงานของแต่ละ Use Case ระหว่าง Object ต่างๆ ที่ส่งข้อความถึงกันและกัน โดย Sequence Diagram จะช่วยให้มองเห็นการทำงานของภาพรวมของระบบ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียน Sequence Diagram แสดงดังตารางที่ 3.22

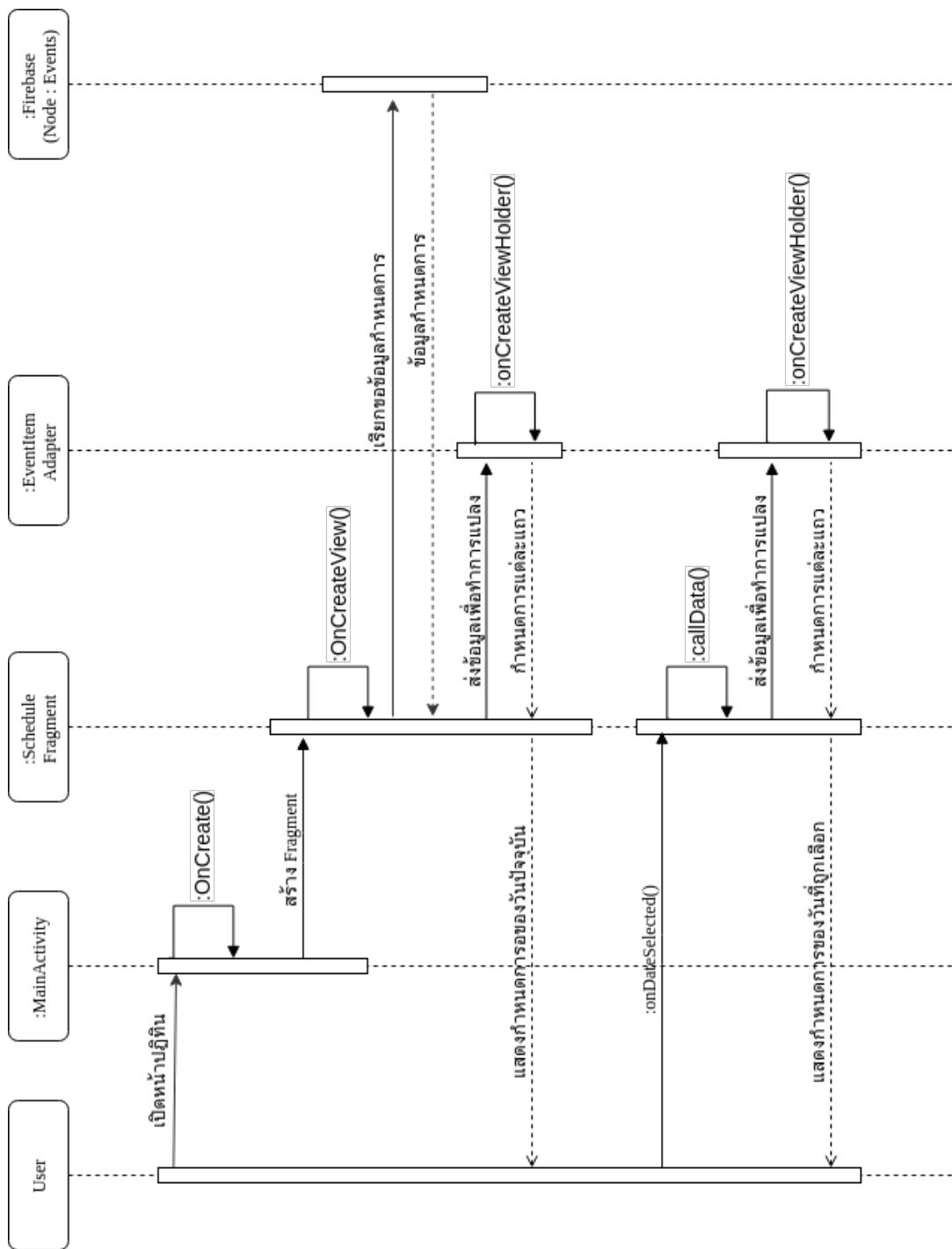
ตารางที่ 3.22: สัญลักษณ์ของ Sequence Diagram

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



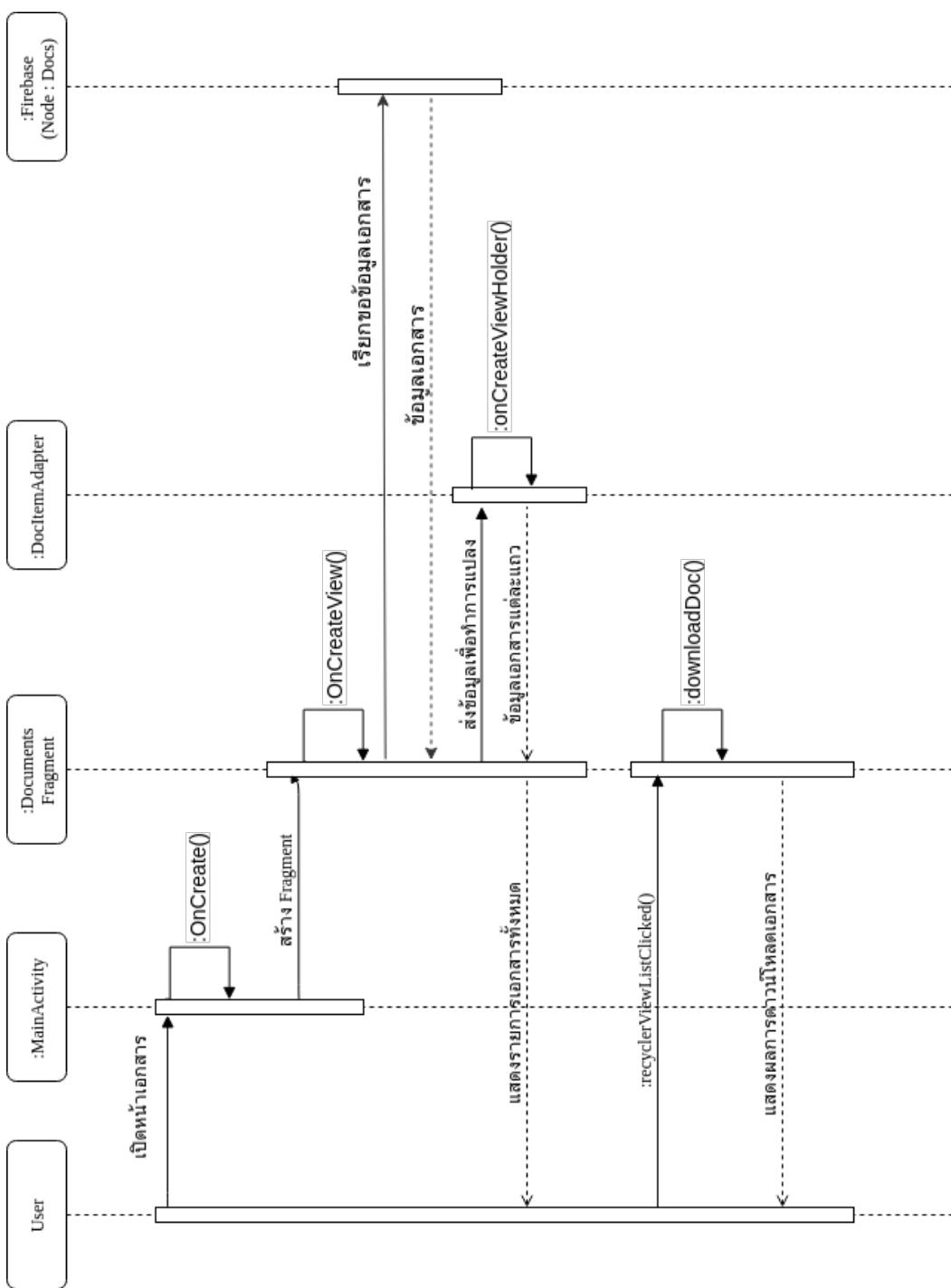
รูปที่ 3.24: Sequence Diagram การแสดงข่าวสาร

จากภาพที่ 3.24 สามารถอธิบายแผนภาพ Sequence Diagram และข่าวสารได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส FeedFragment เมื่อ FeedFragment ถูกติดตั้งบน MainActivity เมธอด callData() จะสืบค้นข้อมูลจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส FeedItemAdapter โดยมีการคืนค่าเป็นข้อมูลข่าวสารแต่ละแถวและในขั้นตอนสุดท้ายคลาส FeedFragment จะทำการแสดงรายการข้อมูลข่าวสารทั้งหมด ออกทางหน้าจอ หากผู้ใช้มีการกดเลือกข่าวสารบางแถวคลาส FeedFragment จะทำการเรียกใช้ PostDetailActivity เพื่อแสดงรายละเอียดข้อมูลข่าวสารของแถวที่ถูกเลือก



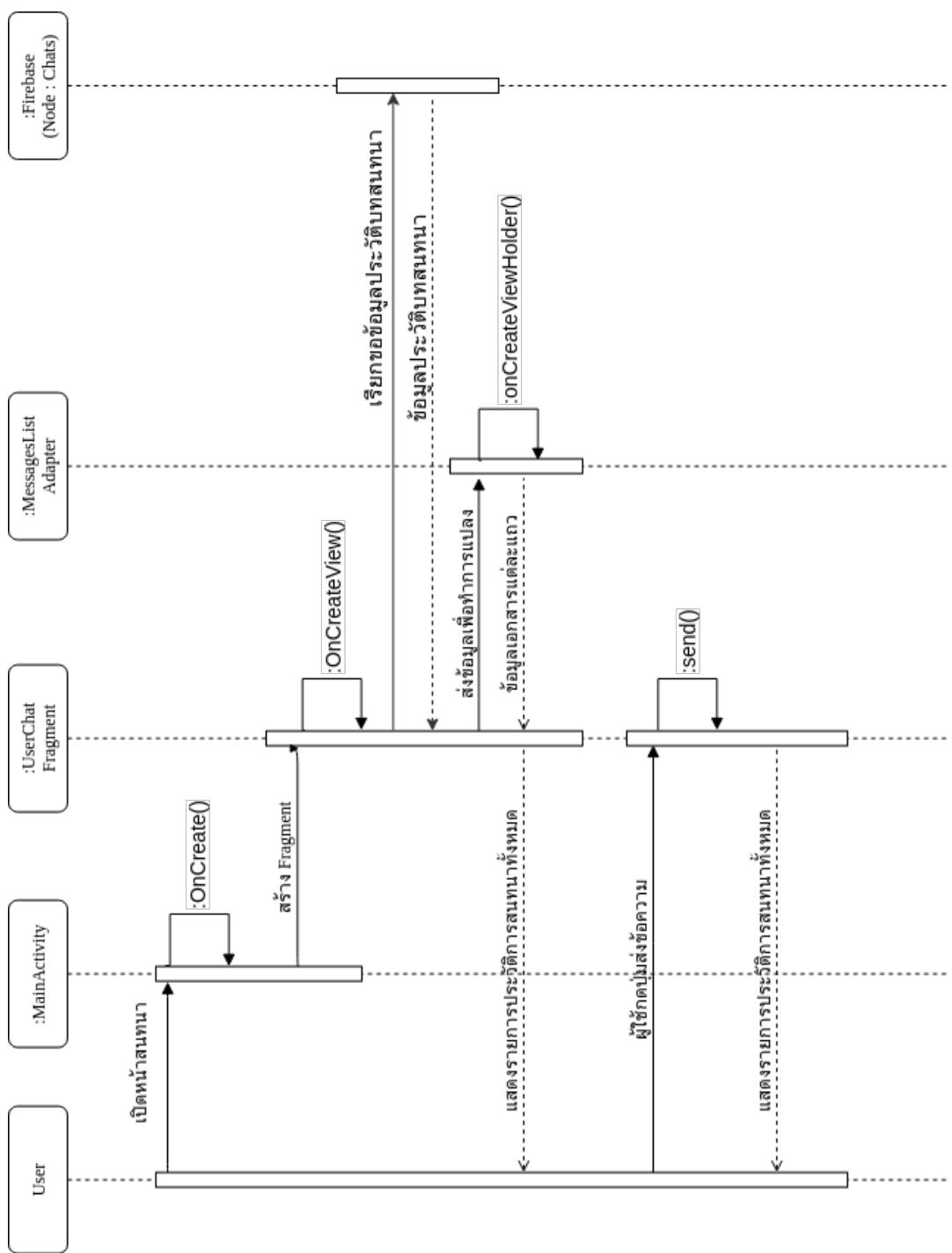
รูปที่ 3.25: Sequence Diagram การแสดงปฏิทินสำหรับการนัดการ

จากภาพที่ 3.25 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 ScheduleItemAdapter โดยมีการคืนค่าเป็นข้อมูลกำหนดการแต่ละแถวและในขั้นตอนสุดท้ายคลาส ScheduleFragment จะทำการแสดงรายการกำหนดการวันปัจจุบันออกทางหน้าจอ หากผู้ใช้มีการกดเลือกวันที่ต้องการทราบกำหนดการจากปฏิทินคลาส ScheduleFragment จะทำการเรียกใช้ callData() อีกครั้งโดยสืบค้นข้อมูลกำหนดการของวันที่ถูกเลือกจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส ScheduleItemAdapter โดยมีการคืนค่าเป็นข้อมูลแต่กำหนดการละ แถวและในขั้นตอนสุดท้ายคลาส ScheduleFragment จะทำการแสดงรายการกำหนดการวันที่ผู้ใช้เลือกออกทางหน้าจอ



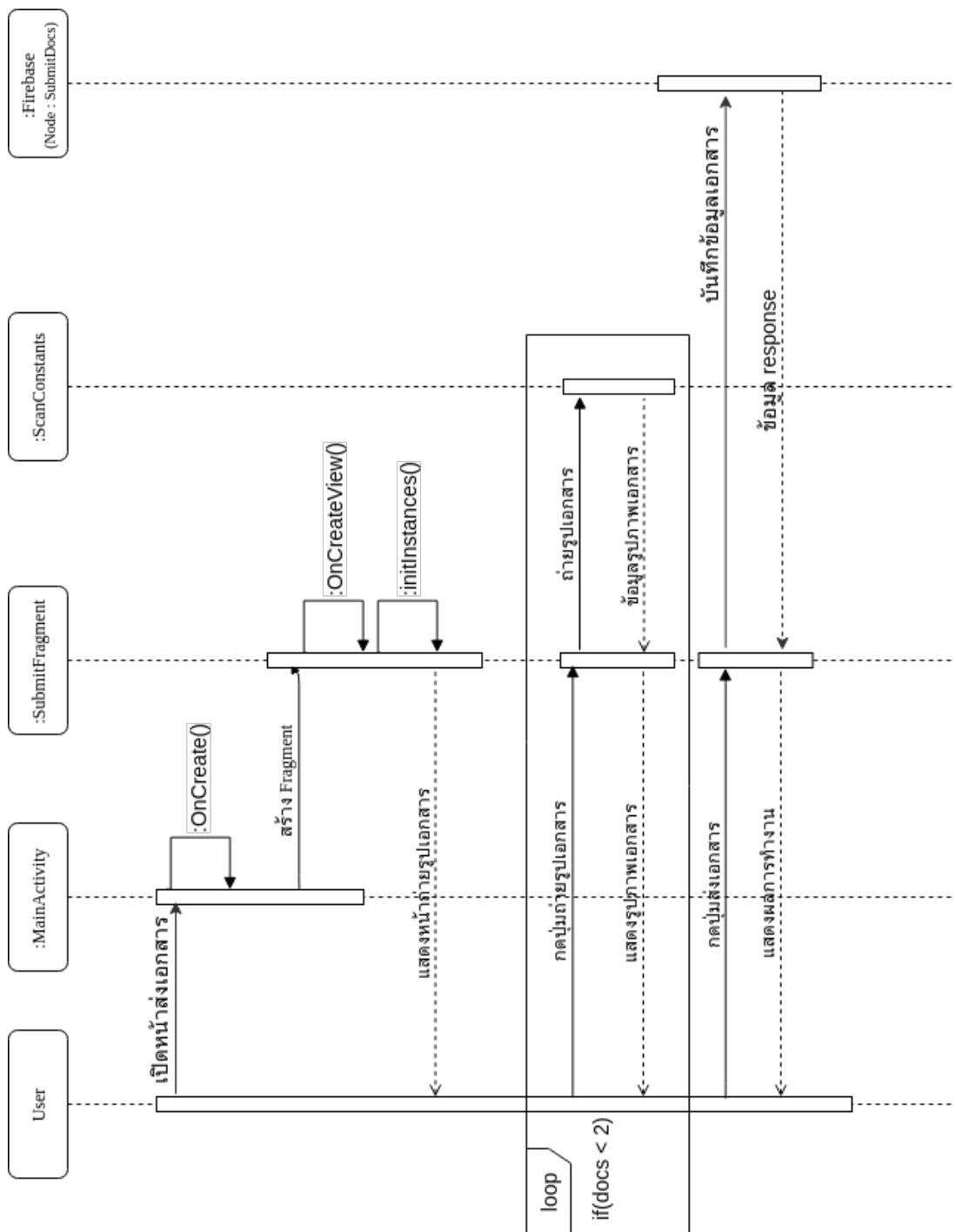
รูปที่ 3.26: Sequence Diagram การแสดงตามไฟล์ต่อไปนี้

จากภาพที่ 3.26 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 โดยมีการคืนค่าเป็นข้อมูลเอกสารแต่ละดวงและในขั้นตอนสุดท้ายคลาส Documents-Fragment จะทำการแสดงรายการรากำหนดร่วนปัจจุบันออกทางหน้าจอ



รูปที่ 3.27: Sequence Diagram การแสดงบทสนทนา

จากภาพที่ 3.27 สามารถอธิบายแผนภาพ Sequence Diagram แสดงการสนทนา ได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส UserChatFragment เมื่อ UserChatFragment ถูกติดตั้งบน MainActivity เมธอด getMessage() จะสืบค้นข้อมูลประวัติการสนทนาของผู้ใช้คนปัจจุบันทั้งหมดจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส MessagesListAdapter โดยมีการคืนค่าเป็นข้อมูลรายการประวัติการสนทนาทั้งหมดและในขั้นตอนสุดท้ายคลาส User-ChatFragment จะทำการแสดงรายการประวัติการสนทนาทั้งหมดโดยทางหน้าจอ เมื่อผู้ใช้พิมพ์ข้อความและกดปุ่มส่งระบบจะเรียกให้เมธอด send() เพื่อทำการบันทึกข้อมูลไว้บน Firebase FireStore และทำการแสดงข้อมูลรายการประวัติการสนทนาทั้งหมดที่ถูกอัพเดท



รูปที่ 3.28: Sequence Diagram แสดงถึง logic ของการตรวจสอบ

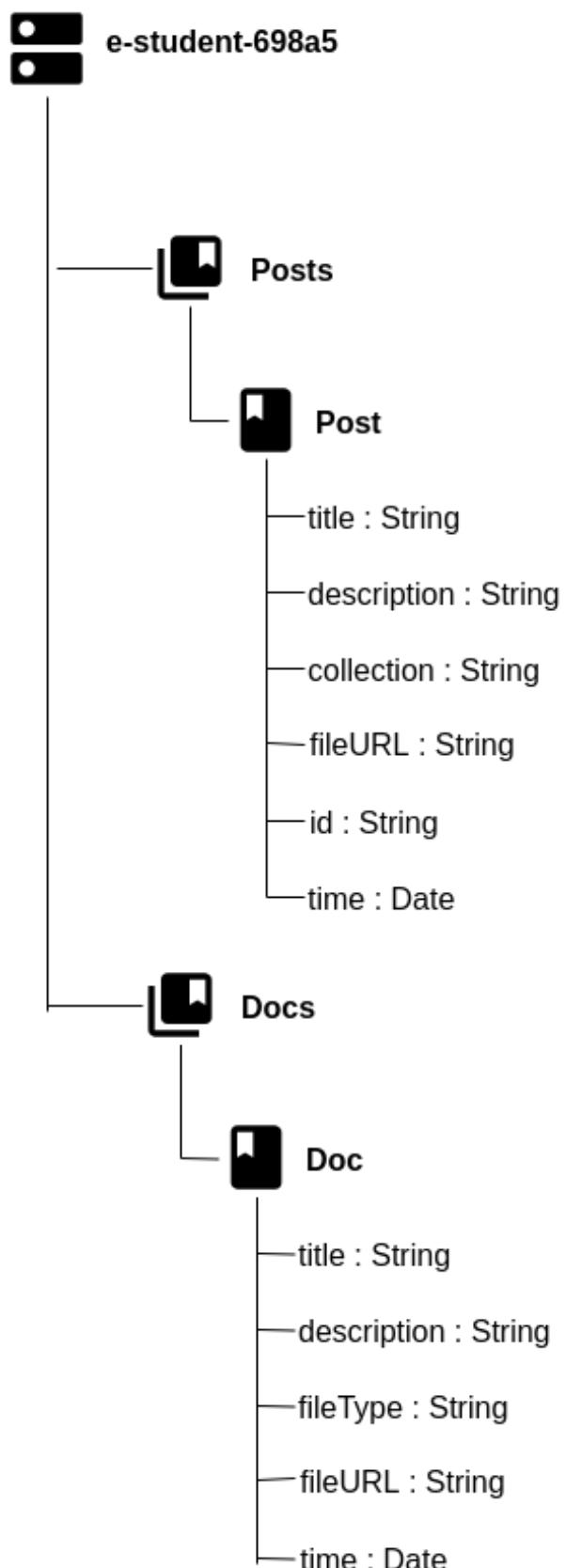
จากภาพที่ 3.28 สามารถอธิบายแผนภาพ Sequence Diagram แสดงสิ่งเอกสารตรวจสอบได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส SubmitFragment เมื่อ SubmitFragment ถูกติดตั้งบน MainActivity เมธอด initInstances() จะถูกเรียกเพื่อสร้างหน้าจอแสดงผลเมื่อผู้ใช้กดปุ่มถ่ายรูประบบจะเรียกใช้ไลบรารี ScanConstants เพื่อถ่ายภาพเอกสารและรอให้ผู้ใช้ถ่ายครบทั้งสองแผ่นจึงจะแสดงปุ่มกดส่งเอกสารเพื่อตรวจสอบ

### 3.7 โครงสร้างฐานข้อมูลไฟร์เบส(Firebase Database Structure)

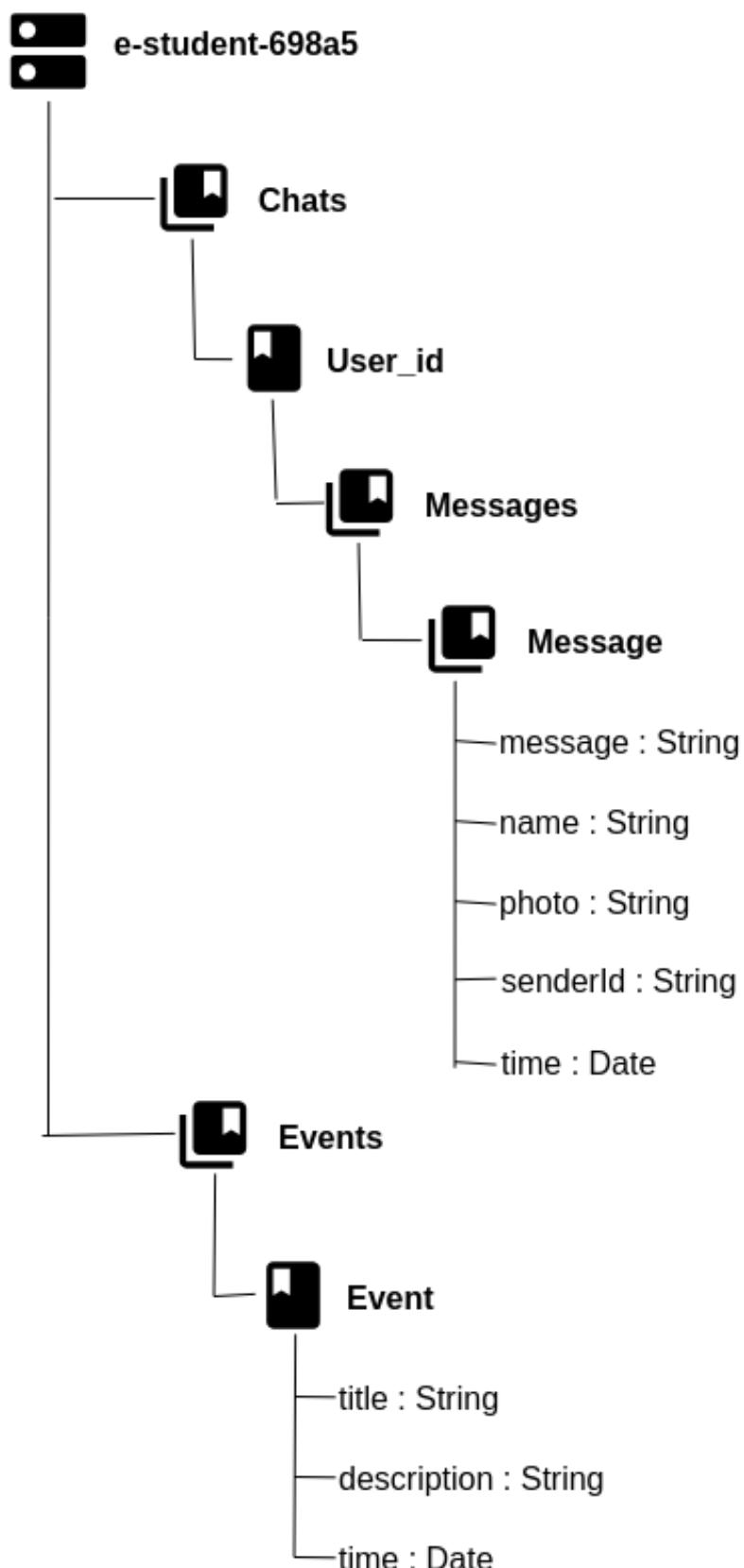
Firebase Database นั้นเป็น Database แบบ NoSQL และเป็น JSON database ที่มีโครงสร้างที่เป็น Key และ Value จัดเก็บข้อมูลในลักษณะโหนด หากต้องการเรียกงานจะเรียกใช้โดยการท่องไปยังโหนดที่ต้องการ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียนโครงสร้างฐานข้อมูลแบบ Firebase แสดงดังตารางที่ 3.23

ตารางที่ 3.23: สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Firebase

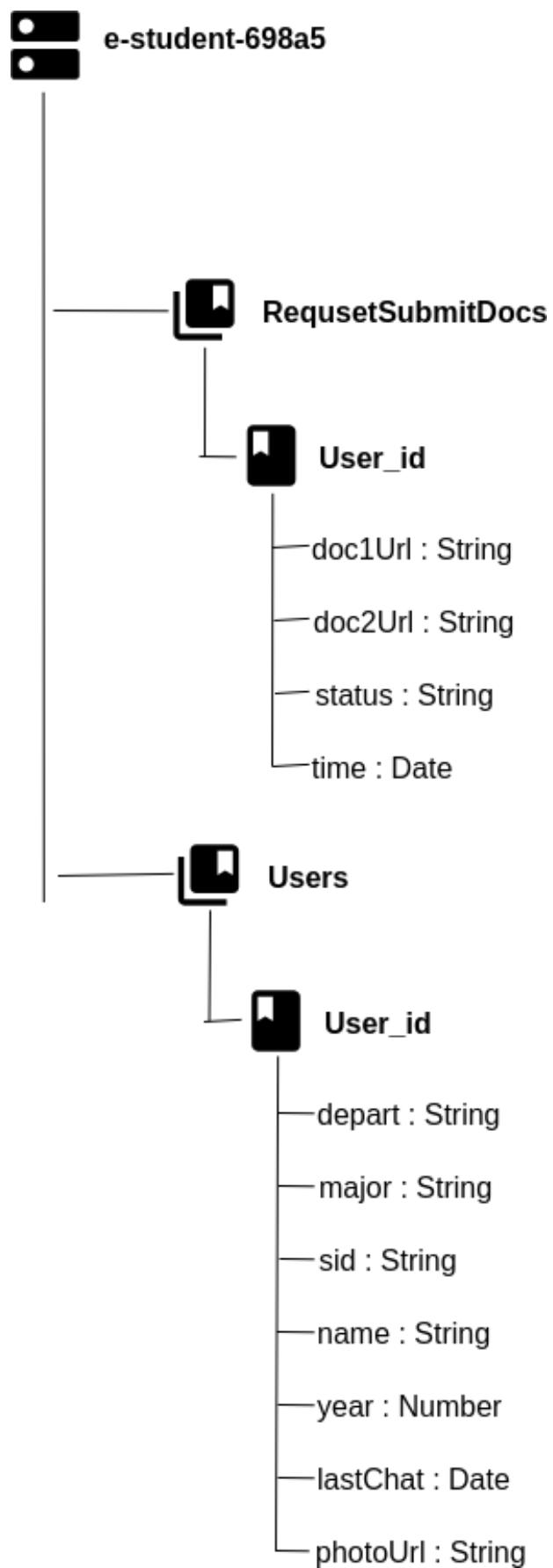
สัญลักษณ์	คำอธิบาย
	Database เป็นการเรียกชื่อแทนโหนด(Node)บนสุดที่ใช้ในการเก็บข้อมูล
	Collection เป็นการเรียกชื่อแทนของการเก็บหลาย ๆ เอกสารไว้ด้วยกัน
	Document เป็นการเรียกชื่อแทน หน่วย การเก็บ ของ ข้อมูล ใน Cloud Firestore ภายในจะประกอบไปด้วย ชื่อของ Document ชื่อของคีย์ (key) และ ค่าข้อมูล (value)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



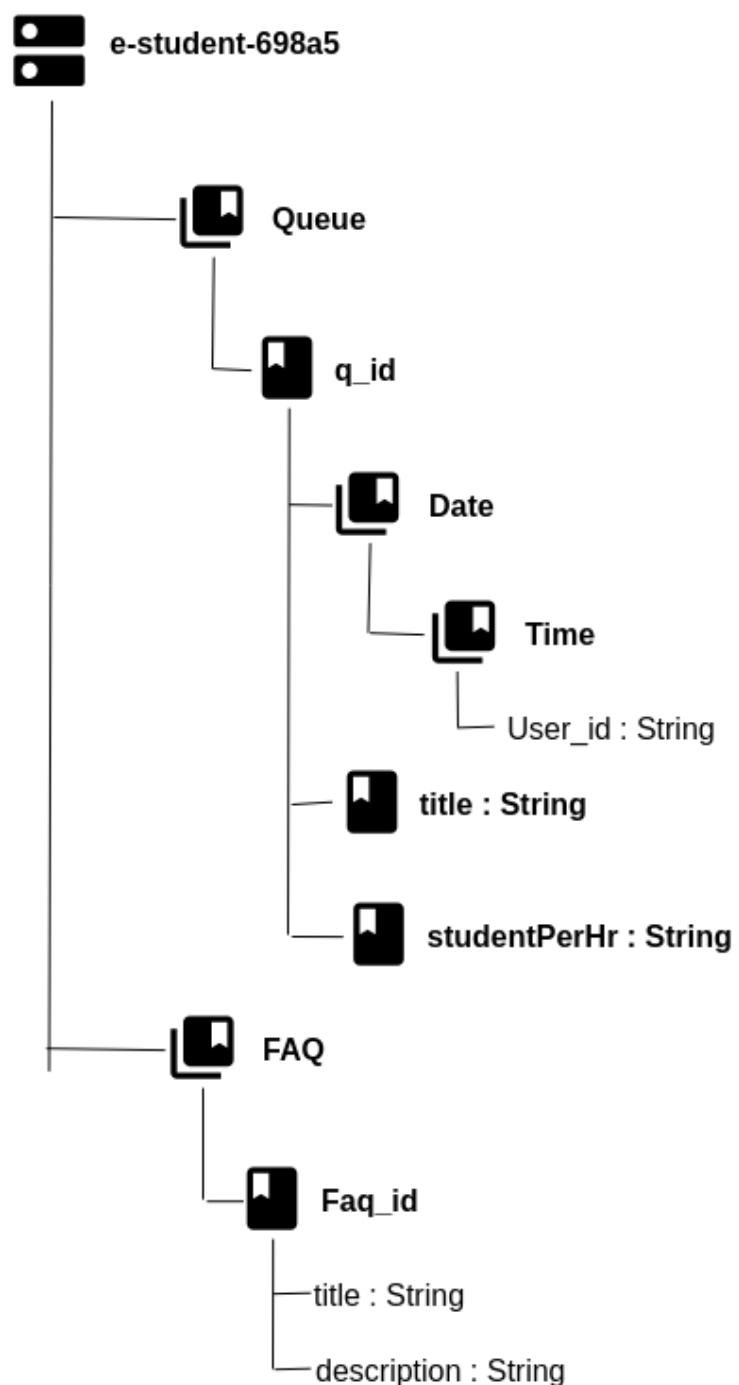
รูปที่ 3.29: โครงสร้างฐานข้อมูลแบบ Firebase



รูปที่ 3.30: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

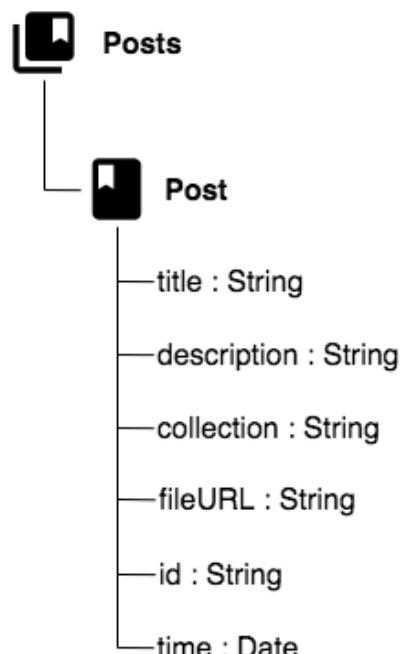


รูปที่ 3.31: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)



รูปที่ 3.32: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

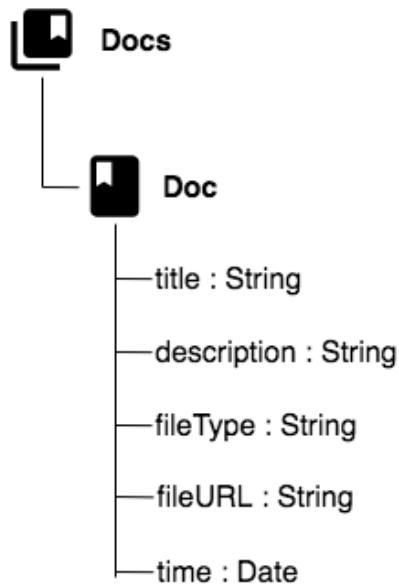
จากรูปที่ 3.29-3.40 สามารถอธิบายโครงสร้างของข้อมูลได้ดังนี้



รูปที่ 3.33: โหนดเก็บข้อมูลประกาศ

ตารางที่ 3.24: อธิบายโหนดที่ใช้เก็บข้อมูลประกาศ

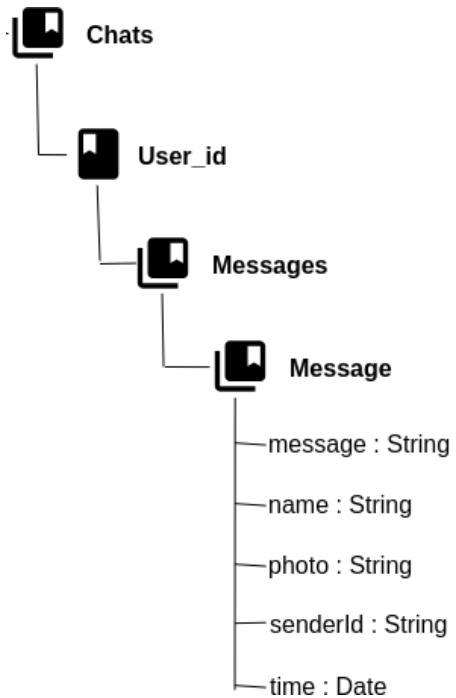
Key	คำอธิบาย
Posts	โหนดสำหรับเก็บข้อมูลประกาศทั้งหมด
Post	สำหรับเก็บข้อมูลแต่ละประกาศ
title	สำหรับเก็บชื่อหัวข้อประกาศ
description	สำหรับเก็บรายละเอียดประกาศ
collection	สำหรับเก็บประเภทของประกาศได้แก่ สารานุกรมและเฉพาะบุคคล
fileURL	สำหรับเก็บ url ของเอกสารแนบประกาศ
id	สำหรับเก็บรหัสของประกาศ
time	สำหรับเก็บเวลาที่ประกาศ



รูปที่ 3.34: โหนดเก็บข้อมูลเอกสารที่เกี่ยวข้อง

ตารางที่ 3.25: อธิบายโหนดที่ใช้เก็บข้อมูลเอกสารที่เกี่ยวข้อง

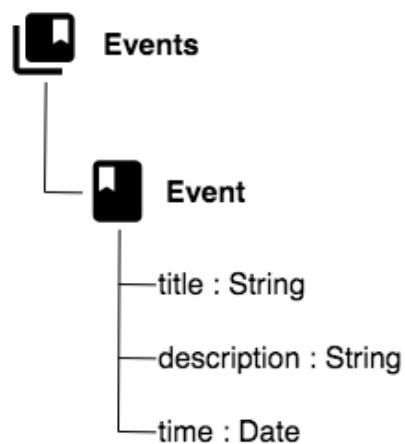
Key	คำอธิบาย
Docs	โหนดสำหรับเก็บข้อมูลของเอกสารที่เกี่ยวข้องทั้งหมด
Doc	สำหรับเก็บข้อมูลเอกสารแต่ละฉบับ
title	สำหรับเก็บชื่อหัวเรื่องของเอกสาร
description	สำหรับเก็บรายละเอียดของเอกสาร
fileType	สำหรับนามสกุลไฟล์เอกสาร เช่น .pdf .png เป็นต้น
fileURL	สำหรับเก็บ url ของเอกสาร
time	สำหรับเก็บเวลาที่ถูกอัพโหลดเข้าสู่ระบบโดยเจ้าหน้าที่



รูปที่ 3.35: โหนดเก็บข้อมูลประวัติการสนทนา

ตารางที่ 3.26: อธิบายโหนดที่ใช้เก็บข้อมูลประวัติการสนทนา

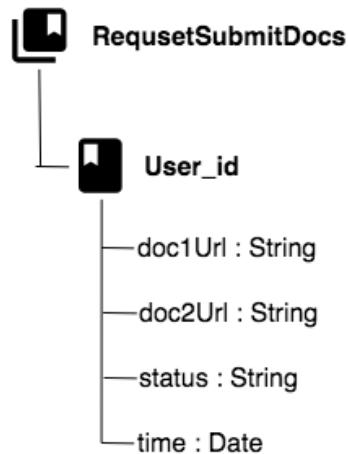
Key	คำอธิบาย
Chats	โหนดสำหรับเก็บข้อมูลประวัติการสนทนาทั้งหมด
User_id	สำหรับเก็บประวัติการสนทนาของผู้ใช้แต่ละคน
Messages	สำหรับเก็บประวัติการสนทนาทั้งหมดของผู้ใช้
Message	สำหรับเก็บข้อมูลของแต่ละข้อความ
message	สำหรับเก็บข้อความ
name	สำหรับเก็บชื่อของผู้ส่งข้อความ
photo	สำหรับเก็บ url รูปภาพของผู้ส่งข้อความ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



รูปที่ 3.36: โหนดเก็บข้อมูลกำหนดการ

ตารางที่ 3.27: อธิบายโหนดที่ใช้เก็บข้อมูลกำหนดการ

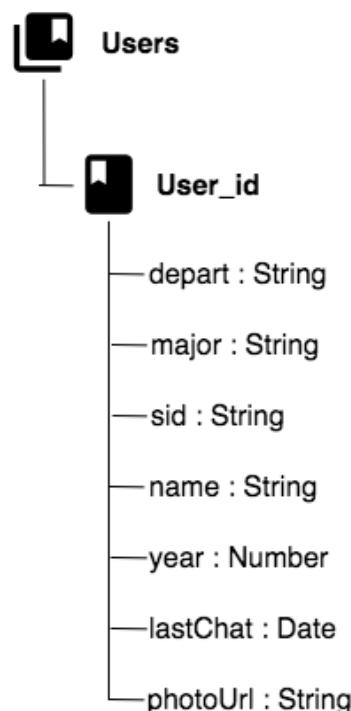
Key	คำอธิบาย
Events	โหนดสำหรับเก็บข้อมูลของกำหนดการทั้งหมด
Event	สำหรับเก็บข้อมูลของแต่ละกำหนดการ
title	สำหรับเก็บชื่อหัวข้อของกำหนดการ
description	สำหรับเก็บรายละเอียดของกำหนดการ
time	สำหรับเก็บเวลาของกำหนดการ



รูปที่ 3.37: โหนดเก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา

ตารางที่ 3.28: อธิบายโหนดที่ใช้เก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา

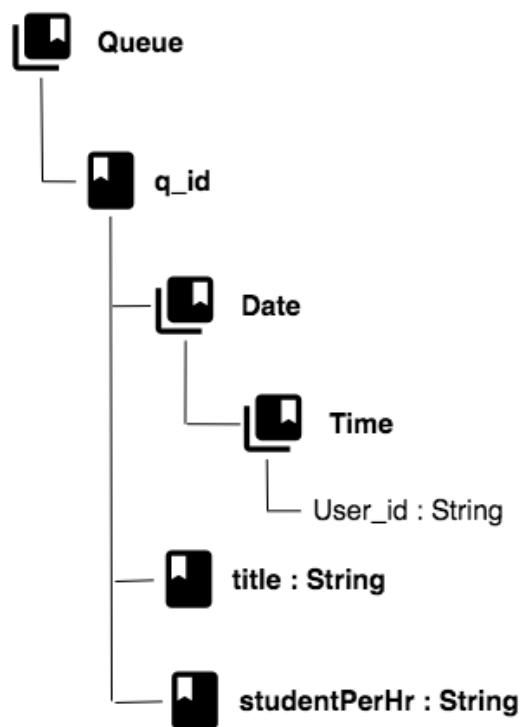
Key	คำอธิบาย
RusetSubmitDocs	โหนดสำหรับเก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษาทั้งหมด
User_id	สำหรับเก็บข้อมูลของแต่ละสำเนาเอกสารของนักศึกษาแต่ละคน
doc2	สำหรับเก็บ url ของภาพถ่ายสำเนาเอกสารฉบับที่ 1
doc2	สำหรับเก็บ url ของภาพถ่ายสำเนาเอกสารฉบับที่ 2
status	สำหรับเก็บผลการตรวจสอบของเจ้าหน้าที่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



รูปที่ 3.38: โหนดเก็บข้อมูลของนักศึกษา

ตารางที่ 3.29: อธิบายโหนดที่ใช้เก็บข้อมูลของนักศึกษา

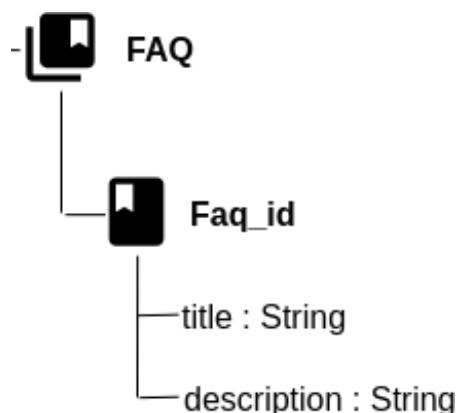
Key	คำอธิบาย
Users	โหนดสำหรับเก็บข้อมูลของนักศึกษา
User_id	สำหรับเก็บข้อมูลของนักศึกษาแต่ละคน
depart	สำหรับเก็บภาควิชาของนักศึกษา
major	สำหรับเก็บสาขาวิชาของนักศึกษา
sid	สำหรับเก็บรหัสประจำตัวนักศึกษา
name	สำหรับเก็บชื่อของนักศึกษา
year	สำหรับเก็บชั้นปีของนักศึกษา
lastChat	สำหรับเก็บเวลาที่สนทนากับเจ้าหน้าที่ล่าสุด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



รูปที่ 3.39: โน๊นดเก็บข้อมูลการจองคิวของนักศึกษา

ตารางที่ 3.30: อธิบายโน๊นดที่ใช้เก็บข้อมูลการจองคิวของนักศึกษา

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



รูปที่ 3.40: โหนดเก็บข้อมูลคำถามที่พับบอຍ

ตารางที่ 3.31: อธิบายโหนดที่ใช้เก็บข้อมูลคำถามที่พับบอຍ

Key	คำอธิบาย
Queue	โหนดสำหรับเก็บข้อมูลคำถามที่พับบอຍทั้งหมด
Faq_id	สำหรับเก็บข้อมูลคำถามที่พับบอຍแต่ละรายการ
title	สำหรับเก็บคำถาม
description	สำหรับเก็บคำตอบ

## บทที่ 4

### การพัฒนาระบบ

หลังจากที่ได้มีการเตรียมความพร้อมสำหรับการพัฒนาในด้านต่าง ไม่ว่าจะเป็นที่มาและความสำคัญของปัญหา เทคโนโลยีที่มีความเหมาะสมกับระบบ และการออกแบบระบบการทำงานรวมไปถึงโครงสร้างของข้อมูล ในบทนี้จะเป็นการพูดถึงการสร้างระบบที่ได้มีการออกแบบแบบไว้ในบทที่แล้วจะถูกนำเสนอในบทนี้ โดยการพัฒนาระบบแบ่งได้เป็นส่วนต่าง ๆ ดังนี้

#### 4.1 การพัฒนาเว็บแอปพลิเคชัน

#### 4.2 การพัฒนาเว็บแอปพลิเคชันแยกรอยด์แอปพลิเคชัน

#### 4.1 การพัฒนาเว็บแอปพลิเคชัน

การพัฒนาระบบของทุนเงินให้กู้ยืมเพื่อการศึกษาสำหรับเว็บแอปพลิเคชันนั้นวัตถุประสงค์หลังเพื่อสร้างความสะดวกต่อการทำงานของเจ้าหน้าที่อันเนื่องมาจากข้อจำกัดบางประการหากใช้ระบบทำงานบนอุปกรณ์สมาร์ทโฟนเพียงอย่างเดียว โดยตัวเว็บแอปพลิเคชันนี้ถูกพัฒนาขึ้นด้วย Vue.js มีรายละเอียดการทำงานดังนี้

##### 4.1.1 การเชื่อมต่อ Cloud Firestore

bsection, ในการเชื่อมต่อเว็บแอปพลิเคชันกับไฟร์เบสเพื่อใช้บริการต่างๆ ของไฟร์เบส ทำได้ดังนี้

```
1 export default {
2     apiKey: "
3         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
4     authDomain: "e-student-698a5.firebaseio.com",
5     databaseURL: "https://e-student-698a5.
6         firebaseio.com",
7     projectId: "e-student-698a5",
8     storageBucket: "e-student-698a5.appspot.com",
9     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 เป็นการนำเข้าโมดูลตั้งค่าที่ได้จากการรูปภาพที่ 4.2
- บรรทัดที่ 5 เป็นการส่งออกโมดูลไฟร์เบสเพื่อใช้ในไฟล์อื่น ๆ ซึ่งเมื่อถึงขั้นตอนนี้การเชื่อมต่อบริการไฟร์เบสลือว่าเป็นอันเสร็จ

#### 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 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น HTML
- บรรทัดที่ 2-32 เป็นการความคุมลักษณะการแสดงผลบนหน้าจอ
- บรรทัดที่ 3-31 เป็นการกำหนดขนาดของเนื้อหาใน
- 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     }
18   }
19 }
```

รูปที่ 4.4: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ SignIn.vue

จากภาพที่ 4.4 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าเข้าสู่ระบบ สามารถอธิบายการทำงานได้ดังนี้

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

```

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 ໂຄຣສ້າງຂອງການສ້າງໜ້າຂ່າວສາຮ

```

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: ການສ້າງໜ້າຈອສ່ວນຕິດຕໍ່ຜູ້ໃຊ້ຂອງໜ້າຂ່າວສາຮ Home.vue

จากภาพที่ 4.6 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-33 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ 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: การสร้างโลจิก(logic)ของหน้าข่าวสาร Home.vue

จากภาพที่ 4.9 โครงสร้างโลจิกของหน้าข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

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

พร้อมทั้งเรียงลำดับตามวันที่ประกาศ

- 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: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้ารายละเอียดข่าวสาร ViewPost.vue

จากภาพที่ 4.8 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าดูรายละเอียดข่าวสารสามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-8 เป็นแทมเพลตที่ใช้เพื่อสื่อสารกับ 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: การสร้างลอกิจของหน้าดูรายละเอียดของข่าวสาร ViewPost.vue

จากภาพที่ 4.9 โครงสร้างโลจิกของหน้าดูรายละเอียดของข่าวสาร สามารถอธิบายการทำ-

งานได้ดังนี้

- บรรทัดที่ 1 เป็นการฟังก์ชันที่ถูกเรียกทุกครั้งที่ผู้ใช้เปิดหน้าดูรายละเอียดข่าวสาร
- บรรทัดที่ 2 ดึงค่าไอดีของประกาศที่ถูกส่งมาจากหน้าแสดงข่าวสาร
- บรรทัดที่ 3 ทำการสืบค้นข้อมูลข่าวสารจาก Cloud Firestore จากไอดีของประกาศ
- บรรทัดที่ 4-6 เป็นการตรวจสอบว่ามีประกาศดังกล่าวอยู่ในฐานข้อมูล Cloude Firestore หรือไม่

#### 4.1.5 โครงสร้างของการสร้างหน้าสนทนากลุ่ม

```

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 }}
10      </p>
11      <Scroll style="background-color: #EEEEEE;">
12        <ul style="padding:6px;padding-right:8px;">
13          <li v-for="(item, index) in messages" :key="index" style="margin-bottom:8px;">
14            <Card v-if="user.uid !== item.id" :padding="6" style="text-align:left;display: inline-block; background-color: #FAFAFA;">
15              <div>
16                <p>{{ item.message }}</p>
17              </div>
18            </Card>
19            <div v-else style="text-align:right;">
20              <Card :padding="6" style="display: inline-block; background-color: #B2E281;">
21                <p>{{ item.message }}</p>
22              </Card>
23            </div>
24          </li>
25        </ul>
26      </Scroll>
27      <div style="padding:16px;text-align:right;">
28        <Input type="textarea" v-model="formItem.message" placeholder="Type..." v-on:keyup.enter="send"/>
29        <Button :loading="loading" type="primary" style="margin-top:10px;" size="large" @click="send">Send</Button>
30      </div>
31    </Card>
32  </Col>
33 </Row>
34 </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 โครงสร้างລອຈິກຂອງໜ້າສນທນາ ສາມາຮອດຮັບຍາກວຽກທຳການໄດ້ດັ່ງນີ້

- บรรทัดທີ 1 ຂຶ້ອຳພັດທະນາ
- บรรทัดທີ 2 ຕຽບສອບໄວດືຂອງຜູ້ໃຊ້
- บรรทัดທີ 3 ດຶງຄ່າໂປຣໄຟລ໌ຜູ້ໃຊ້ຄົນປ່າງຈຸບັນ
- บรรทัดທີ 4-6 ເພີ້ນຂໍ້ມູນລາຍງານຂໍ້ມູນ Cloud Firestore ໂດຍຮະບຸ path ທີ່ຈະກຳກັນຈັດເກີບ  
ຊຸດຂໍ້ມູນ
- บรรทัดທີ 28-35 ອັບເດທຂໍ້ມູນເວລາສນທນາລ່າສຸດຂອງຜູ້ໃຊ້
- บรรทัดທີ 33 ແສດສຕານະກາວອັບເດທຂໍ້ມູນ

#### 4.1.6 โครงสร้างของการสร้างໜ້າປົກລົງສະແດງກຳທັນດາການ

```

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: ການສ້າງໜ້າຈອສ່ວນຕິດຕ່ອຸ້ນຂອງໜ້າປົກລົງກຳທັນດາການ Schedule.vue

จากภาพที่ 4.12 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าปฏิทินกำหนดการสามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-22 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ 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: การสร้างລອງຈິກຂອງหน້າປະລິບີນກຳທັນດກສະເໜາຮົມບາຍການທຳມະນີ

จากภาพที่ 4.13 โครงสร้างລອງຈິກຂອງหน້າປະລິບີນກຳທັນດກສະເໜາຮົມບາຍການທຳມະນີ

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

รูปที่ 4.14: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ MgPost.vue

จากภาพที่ 4.8 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-31 เป็นการสร้างหน้าเพิ่มข่าวสารเป็นปีอปอพ (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 เป็นการสร้างปุ่มอปโหลดไฟล์
- 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         }

```

รูปที่ 4.15: การสร้างລອຈິກຂອງໜ້າໜ້າສ້າງປະກາສ MgPost.vue

จากภาพที่ 4.15 ໂຄງສ້າງລອຈິກຂອງໜ້າສ້າງປະກາສ ສາມາລອອືບາຍການທຳມາດີດັ່ງນີ້

- ບຣທັດທີ 1 ຂຶ້ອຳພັ້ນກໍ່ໜັນ
- ບຣທັດທີ 4-5 ຊົມຸລຂອງປະກາສ
- ບຣທັດທີ 6-30 ເຮັດໃຫ້ຈານ Cloud Firestore ເພື່ອບັນທຶກປະກາສລົງຈູານຊື່ອຸມຸລ
- ບຣທັດທີ 10-24 ເປັນສ່ວນທີ່ໃໝ່ໃນກາວອັບໂຫດເອກສາຣແນບໄປຍັງໄຟຣີເບສ Storage

- บรรทัดที่ 13-19 เป็นการอัพเดทข้อมูล URL ที่ได้จากการอัปโหลดไฟล์แนบเข้าสู่ฐานข้อมูล

### 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.com/">
14          <Button :type="type" icon="ios-cloud-upload-outline" >{{ uploadBtnTitle }}</Button>
15        </Upload>
16      </FormItem>
17    </Form>
18    <div slot="footer">
19      <Button type="primary" :loading="loading" @click="newDoc">upload</Button>
20      <Button type="ghost" style="margin-left: 8px" @click="modalNewDoc = !modalNewDoc">cancle</Button>
21    </div>
22  </Modal>

```

รูปที่ 4.16: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าอัปโหลดเอกสารที่เกี่ยวข้อง MgDocument.vue

จากภาพที่ 4.16 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าอัปโหลดเอกสารที่เกี่ยวข้องสามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-22 สร้างปีอปอัพแสดงหน้าอัพโหลดไฟล์เอกสาร
- 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         });
28       }
29     }
30   }
31 }

```

รูปที่ 4.17: การสร้างlogicของหน้าหน้าอัพโหลดเอกสารที่เกี่ยวข้อง MgDocument.vue

จากภาพที่ 4.17 โครงสร้างlogicของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 6-26 เป็นการอัพเดทข้อมูลเอกสารเข้าสู่ฐานข้อมูล Cloud Firestore

- บรรทัดที่ 10-12 เป็นการเรียกใช้เฟร์เบส Storage เพื่อทำการอัปโหลดไฟล์เอกสาร
- บรรทัดที่ 13-20 ใช้ในการอัปเดตข้อมูล URL ไปยังฐานข้อมูล Cloude Firestore

#### 4.1.9 โครงสร้างของการสร้างหน้าสร้างกำหนดการของคิวส่งเอกสาร

```

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: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างกำหนดการของคิวส่งเอกสาร

MgQueue.vue

จากภาพที่ 4.18 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างกำหนดการ ของคิวส่งเอกสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-25 การสร้างหน้าต่างปีอปอัพ
- บรรทัดที่ 4 ชื่อหน้าต่างปีอปอัพ
- 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: การสร้างlogicของหน้าสร้างกำหนดการของคิวส่งเอกสาร MgQueue.vue

จากภาพที่ 4.19 โครงสร้างlogicของหน้าสร้างกำหนดการของคิวส่งเอกสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 3-15 ที่ได้จากการอัปโหลดไฟล์แนบเข้าสู่ฐานข้อมูล Cloud Firestore

## 4.2 การพัฒนาเว็บแอปพลิเคชันออนไลน์และแอปพลิเคชัน

การพัฒนาเว็บแอปพลิเคชันออนไลน์และแอปพลิเคชันระบบบกонтุเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีวัตถุประสงค์หลักเพื่ออำนวยความสะดวกในการศึกษาที่โดยส่วนใหญ่ใช้อุปกรณ์พกพาและต้องการใช้งานฮาร์ดแวร์(Hardware) เช่น กล้องที่ใช้ใน

## การถ่ายภาพสำเนาเอกสาร เป็นต้น

### 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 scheduleFrag =
    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 ใช้แสดงผลหน้าจอสนทนาร่วมเจ้าหน้าที่
- บรรทัดที่ 4 ตัวแปร chatFrag ใช้แสดงผลหน้าจอเอกสารที่เกี่ยวข้อง
- บรรทัดที่ 5 ตัวแปร scheduleFrag ใช้แสดงผลหน้าจอปฏิทินกำหนดการ
- บรรทัดที่ 6 ตัวแปร submitFrag ใช้แสดงผลหน้าจอส่งสำเนาเอกสาร
- บรรทัดที่ 7 ตัวแปร userChatFrag ใช้แสดงผลหน้าจอสนทนารับนักศึกษา
- บรรทัดที่ 8 ตัวแปร checkinFrag ใช้แสดงผลหน้าจอจดจำคิวส่งเอกสาร

```

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) เพื่อสับหน้าจอการแสดงผลที่เกิดขึ้นเมื่อผู้ใช้กดที่เมนูนำทาง
- บรรทัดที่ 8-11 เป็นการแสดงผลหน้าข่าวสาร
- บรรทัดที่ 13-16 เป็นการแสดงผลหน้าสนใจสำหรับเจ้าหน้าที่
- บรรทัดที่ 17-20 เป็นการแสดงผลหน้าสนใจสำหรับนักศึกษา
- บรรทัดที่ 22-25 เป็นการแสดงผลหน้าปฏิทินกำหนดการ
- บรรทัดที่ 26-29 เป็นการแสดงผลหน้าดาวน์โหลดเอกสาร
- บรรทัดที่ 30-33 เป็นการแสดงผลหน้าส่งสำเนาเอกสาร
- บรรทัดที่ 34-37 เป็นการแสดงผลหน้าจองคิวส่งเอกสาร
- 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 ใช้ในการแสดงข้อมูลลิสต์รายการข่าวสาร
- บรรทัดที่ 2 ตัวแปร db ใช้ในการสืบค้นข้อมูลจากข่าวสารจาก Cloud Firestore
- บรรทัดที่ 3 ตัวแปร posts ใช้ในการเก็บชุดข้อมูลที่ได้จากการสืบค้นข้อมูล
- บรรทัดที่ 4 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลเป็นลิสต์รายการเพื่อแสดงบน 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 เริ่มทำการสืบค้นข้อมูลประกาศทั้งหมดพร้อมทั้งเรียงลำดับข้อมูลจากประกาศล่าสุดก่อน
- บรรทัดที่ 4-18 รับผลการสืบค้นพร้อมทั้งเพิ่มข้อมูลที่ได้แต่ละแกรเว็บไว้ที่ตัวแปร 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: โค้ดส่วนที่ใช้ในการดักอีเวนต์เมื่อผู้ใช้กดที่แกรเว็บประกาศในคลาส FeedFragment

จากภาพที่ 4.24 โค้ดส่วนที่ใช้ในการดักอีเวนต์เมื่อผู้ใช้กดที่แกรเว็บประกาศในคลาส FeedFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ประกาศตัวแปร intent ประเภทตัวแปร Intent เพื่อใช้กำหนดแอกทิวิตีปลายทางซึ่งในที่นี้คือ PostDetailActivity
- บรรทัดที่ 4-9 เป็นการเพิ่มข้อมูลเข้าเก็บไว้ที่ตัวแปร intent โดยดึงข้อมูลใน posts มาจากตำแหน่งแกร์ที่ถูกผู้ใช้กด

- บรรทัดที่ 11 เริ่มการทำงานของแอคทิวิตี้ PostDetailActivity

#### 4.2.3 โครงสร้างของการสร้างหน้า PostDetailActivity

```

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

```

รูปที่ 4.25: โค้ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส PostDetailActivity

จากภาพที่ 4.25 โค้ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส PostDetailActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-4 เป็นการประกาศตัวแปรที่ใช้ในการเก็บข้อมูลประกาศ
- บรรทัดที่ 6-11 เป็นการตั้งค่าที่ถูกส่งมาจากคลาส FeedFragment ผ่านทาง Intent
- บรรทัดที่ 12-15 เป็นการแสดงผลข้อมูลต่างๆ ออกทางหน้าจอแสดงผล

```

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 โค้ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส ChatActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร db ใช้ในการสืบค้นข้อมูลจาก Cloud Firestore
- บรรทัดที่ 2 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลที่ได้จากการสืบค้นเป็นลิสตรายการ
- 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: โค้ดส่วนที่ใช้ในการสีบคืนประวัติการสนทนาของคลาส ChatActivity

จากภาพที่ 4.28 โค้ดส่วนที่ใช้ในการสีบคืนประวัติการสนทนาของคลาส ChatActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-2 ทำการสีบคืนข้อมูลประวัติการสนทนาจาก 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 }
24 );

```

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 เป็นการอัพเดทเวลาสนทนala สุดของผู้ใช้

#### 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 ใช้ในการแสดงลิสตรายการกำหนดการ
- บรรทัดที่ 2 ตัวแปร db ใช้ในการสืบค้นข้อมูลจาก Cloud Firestore
- บรรทัดที่ 3 ตัวแปร events จัดเก็บข้อมูลกำหนดการ
- บรรทัดที่ 4 ตัวแปร adapter ใช้ในการแปลงชุดข้อมูลกำหนดการเป็นลิสตรายการเพื่อใช้แสดงใน 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

การทดสอบการทำงานของแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี และทดสอบการทำงานในส่วนของเว็บไซต์ โดยทำการทดสอบในลักษณะ Black-box Testing [12] หรือ Data-Driven testing ซึ่งเป็นการทดสอบแบบที่ไม่สนใจโปรเซส (Process) การทำงานภายในของโปรแกรมว่าทำงานอย่างไร แต่จะเน้นไปที่ Input และ Result ที่ได้มากกว่าการทำงานต่าง ๆ ถูกต้องตามความต้องการ (Requirement) หรือไม่ ซึ่งการทดสอบการใช้งานแอปพลิเคชัน และ การใช้งานเว็บแอปพลิเคชัน ได้ผลดังนี้

#### 5.1 การทดสอบการใช้งานแอปพลิเคชัน

- การทดสอบการใช้งานเมนูนำทางของแอปพลิเคชัน การทดสอบเมนูนำทางของแอปพลิเคชันในการนำทางผู้ใช้งาน ซึ่งเมนูหลักประกอบด้วย เมนูหน้าประกาศ เมนูหน้าสถานะ เมนูหน้าปฏิทินกำหนดการ เมนูหน้าดาวน์โหลดเอกสาร เมนูส่งภาพถ่ายสำเนาเอกสาร เมนูหน้าจองคิวส่งเอกสาร เมนูหน้าคำตามที่พบร้อย 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
เมนูหน้าคำตามทีพบบ่อย	กดปุ่ม เมนู หน้า คำตาม ที พบ บ่อย	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

- การทดสอบหน้ารายละเอียดประกาศ ในการแสดงผลหน้าจอรายละเอียดประกาศนั้นจะประกอบไปด้วยหัวเรื่องประกาศ รายละเอียดประกาศ วันที่ประกาศและเอกสารแนบ ผลการทดสอบดังตารางที่ 5.10

ตารางที่ 5.3: ผลการทดสอบหน้ารายละเอียดประกาศ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศทั้งหมด
	กดปุ่มอ่านรายละเอียดประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กดปุ่มดาวน์โหลด เอกสารแนบ	ระบบแสดงผลการดาวน์โหลดเอกสารแนบ
	เมื่อดาวน์โหลดเสร็จ กดปุ่มเปิดเอกสาร	ระบบแสดงผลเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กดปุ่มย้อนกลับอีกครั้ง	ระบบแสดงผลหน้าจอประ-

- การทดสอบหน้าสนทนา ในการแสดงผลหน้าจอสนทนานั้นจะประกอบไปด้วยรายการประวัติการสนทนา ซึ่งกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.11

ตารางที่ 5.4: ผลการการทดสอบหน้าสนทนา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาพร้อมทั้งแสดงรายการประวัติการสนทนา
	กดปุ่มที่ซ่องกรอกข้อความ	ระบบแสดงตัวกรอบ(cursor) เพื่อชี้ให้รู้ว่าตำแหน่งของการพิมพ์อักขระ
	พิมพ์อักขระ	ระบบแสดงผลอักษรที่ถูกพิมพ์
	กดปุ่มส่งข้อความ	ระบบแสดงข้อความที่ถูกพิมพ์บนรายการประวัติสนทนาล่าสุด
	กดปุ่มย้อนกลับ	ระบบแสดงผู้ลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าปฏิทินกำหนดการ ใน การแสดงผลหน้าปฏิทินกำหนดการนั้นจะประกอบไปด้วยรายการประวัติการสนทน่า ซึ่งกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.12

ตารางที่ 5.5: ผลการการทดสอบหน้าปฏิทินกำหนดการ

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

- การทดสอบหน้าดาวน์โหลดเอกสาร ในการแสดงผลหน้าจอดาวน์โหลดเอกสารนั้นจะประกอบไปด้วยรายการเอกสารโดยที่แท็ลเลอร์บบจะแสดงชื่อเอกสารและปุ่มดาวน์โหลดเอกสาร ผลการทดสอบดังตารางที่ 5.13

ตารางที่ 5.6: ผลการการทดสอบหน้าดาวน์โหลดเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลดเอกสาร	ระบบแสดงผลหน้าจอ รายรายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มดาวน์โหลดเอกสาร	ระบบแสดงผลการดาวน์โหลดเอกสาร
	เมื่อ ดาวน์โหลด เสร็จ กด ปุ่ม เปิดเอกสาร	ระบบแสดงผลเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอ รายรายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอ ประกาศ พร้อมทั้งแสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร ในการแสดงผลหน้าส่งภาพถ่ายสำเนาเอกสาร นั้นจะประกอบไปด้วยปุ่มเพิ่มเอกสารฉบับที่ 1 ปุ่มเพิ่มเอกสารฉบับที่ 2 และปุ่มส่งเอกสาร ผลการทดสอบดังตารางที่ 5.7

ตารางที่ 5.7: ผลการทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้า ส่ง ภาพถ่าย สำเนา เอกสาร	กดปุ่มเมนูหน้าจอคงคิวส่งเอกสาร	ระบบ แสดง หน้า จอ คิว ส่งเอกสาร
	กดปุ่มเพิ่มเอกสารฉบับที่ 1	ระบบแสดงหน้าจอกล้องถ่ายภาพ
	กดปุ่มถ่ายภาพเอกสาร	ระบบแสดงผลภาพเอกสาร
	กดปุ่มถัดไป	ระบบ แสดง ผล หน้า ปรับ แต่ง ภาพเอกสาร
	กดปุ่มยืนยัน	ระบบ แสดง ผล ภาพ หน้า ส่ง ภาพถ่ายสำเนาเอกสารพร้อม ทั้งแสดงผลภาพเอกสารฉบับที่ 1
	กดปุ่มเพิ่มเอกสารฉบับที่ 2	ระบบแสดงหน้าจอกล้องถ่ายภาพ
	กดปุ่มถ่ายภาพเอกสาร	ระบบแสดงผลภาพเอกสาร
	กดปุ่มถัดไป	ระบบ แสดง ผล หน้า ปรับ แต่ง ภาพเอกสาร
	กดปุ่มยืนยัน	ระบบ แสดง ผล ภาพ หน้า ส่ง ภาพถ่ายสำเนาเอกสารพร้อม ทั้งแสดงผลภาพเอกสารฉบับที่ 1 และฉบับที่ 2
กดส่งเอกสาร		ระบบแสดงผลการส่งเอกสาร และ แสดง สถานะ การ ตรวจ เอกสาร
กดปุ่มย้อนกลับ		ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการ ข่าวสารทั้งหมด

- การทดสอบหน้าจอของคิวส่งเอกสาร ใน การแสดงผลหน้าจอของคิวส่งเอกสารนั้นจะประกอบไปด้วยปุ่มกดเลือกวันที่ ปุ่มกดเลือเวลา ผลการทดสอบดังตารางที่ 5.8

ตารางที่ 5.8: ผลการทดสอบหน้าจอของคิวส่งเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าจอของคิวส่งเอกสาร	กดปุ่มเมนูหน้าจอของคิวส่งเอกสาร	ระบบแสดงผลหน้าจอกำหนดการส่งเอกสาร
	กดปุ่มเลือกวันที่ ต้องการ ส่งเอกสาร	ระบบแสดงผลวันที่ถูกเลือก
	กดปุ่มเลือกเวลาที่ต้องการส่งเอกสาร	ระบบแสดงผลเวลาที่ถูกเลือกพร้อมทั้งแสดงปุ่มกดบันทึก
	กดปุ่มบันทึก	ระบบแสดงผลการจองวันที่ส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

## 5.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน

- การทดสอบการใช้งานเมนูนำทางของเว็บแอปพลิเคชัน การทดสอบเมนูนำทางของเว็บแอปพลิเคชันในการนำทางผู้ใช้งาน ซึ่งเมนูหลักประกอบด้วย เมนูหน้าประกาศ เมนูหน้าสนทนามenuหน้าปฏิทินกำหนดการ เมนูหน้าดาวน์โหลดเอกสาร เมนูหน้าคำาถามทีพบป้อย เมนูหน้าเกี่ยวกับและเมนูอุကจาระบบ 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
เมนูหน้าคำานทีพบป้อย	กด ปุ่ม เมนู หน้า คำานที 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

- การทดสอบหน้ารายละเอียดประกาศ ในการแสดงผลหน้าจอรายละเอียดประกาศนั้นจะประกอบไปด้วยหัวเรื่องประกาศ รายละเอียดประกาศ วันที่ประกาศและเอกสารแนบ ผลการทดสอบดังตารางที่ 5.10

ตารางที่ 5.10: ผลการทดสอบหน้ารายละเอียดประกาศ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูประกาศ	ระบบแสดงผู้ล หน้าจอประกาศ พร้อม ทั้งแสดงรายการประกาศทั้งหมด
	กดปุ่มอ่านรายละเอียดประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กดปุ่มดาวน์โหลดเอกสารแนบ	ระบบแสดงผลการดาวน์โหลดเอกสารแนบ
	กดปุ่มย้อนกลับ	ระบบแสดงผู้ล หน้าจอประกาศ พร้อม ทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าสนทนา ในการแสดงผลหน้าจอสนทนานั้นจะประกอบไปด้วยรายการประวัติการสนทนา ซึ่งกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.11

ตารางที่ 5.11: ผลการการทดสอบหน้าสนทนา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาพร้อมทั้งแสดงรายการประวัติการสนทนา
	กดปุ่มที่ซ่องกรอกข้อความ	ระบบแสดงตัวกรอบ(cursor) เพื่อชี้ให้รู้ว่าตำแหน่งของการพิมพ์อักขระ
	พิมพ์อักขระ	ระบบแสดงผลอักษรที่ถูกพิมพ์
	กดปุ่มส่งข้อความ	ระบบแสดงข้อความที่ถูกพิมพ์บนรายการประวัติสนทนาล่าสุด
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าปฏิทินกำหนดการ ใน การแสดงผลหน้าปฏิทินกำหนดการนั้นจะประกอบไปด้วยรายการประวัติการสนทน่า ซึ่งกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.12

ตารางที่ 5.12: ผลการการทดสอบหน้าปฏิทินกำหนดการ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงตารางกำหนดการทั้งหมด
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอ ประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

- การทดสอบหน้าดาวน์โหลดเอกสาร ในการแสดงผลหน้าจอดาวน์โหลดเอกสารนั้นจะประกอบไปด้วยรายการเอกสารโดยที่แท่นจะบีบจะแสดงชื่อเอกสารและปุ่มดาวน์โหลดเอกสาร ผลการทดสอบดังตารางที่ 5.13

ตารางที่ 5.13: ผลการการทดสอบหน้าดาวน์โหลดเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลดเอกสาร	ระบบ แสดง ผล หน้า จอ ตาราง รายการ เอกสาร ใน ระบบ พร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มดาวน์โหลดเอกสาร	ระบบแสดงผลการดาวน์โหลดเอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ รายการเอกสารในระบบพร้อมทั้ง แสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผู้ หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการ ข่าวสารทั่วหมู่

## บทที่ 6

### สรุปและข้อเสนอแนะ

การดำเนินโครงการเพื่อพัฒนาระบบ XX นี้ พบร่วมระบบสามารถทำงานได้ตามที่วิเคราะห์ และออกแบบไว้ แต่ก็พบปัญหาและอุปสรรคระหว่างการพัฒนา ในบทนี้ผู้พัฒนาจึงขอสรุปความสามารถของระบบ ซึ่งจะช่วยให้ผู้ใช้งานเข้าใจการทำงานของระบบ XX มากยิ่งขึ้น พร้อมเสนอแนวทางในการพัฒนาระบบ XX ต่อ ตามลำดับ

#### 6.1 สรุปความสามารถของระบบ

ระบบ XX ทั้งเว็บแอปพลิเคชันและโมบายแอปพลิเคชันสามารถสรุปความสามารถที่ระบบทำได้ดังนี้

##### 6.1.1 เว็บแอปพลิเคชัน

ความสามารถหลักของเว็บแอปพลิเคชันนั้นเน้นสร้างความสะดวกต่อการจัดการเอกสารเรื่องข้อมูลต่างๆ ที่เกี่ยวข้องกับ XX โดยแบ่งความสามารถของระบบตามประเภทของผู้ใช้งาน ดังนี้

###### 1. เจ้าหน้าที่

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

###### 2. นักศึกษา

- สมัครสมาชิกได้
- ดูประกาศได้
- ดูปฏิทินกำหนดการได้
- ดูกำหนดการส่งเอกสารได้

- แก้ไขข้อมูลส่วนตัวได้

#### 6.1.2 แอนดรอยด์พลิเคชัน

เพื่อสร้างความสะดวกในการติดต่อและติดตามประกาศเนื่องจากทำงานบนอุปกรณ์พกพา ทั้งนี้ยังมีบางฟังก์ชันที่จำเป็นต้องทำงานบนอุปกรณ์พกพาด้วย เช่น กล้องถ่ายภาพ เป็นต้น

##### 1. เจ้าหน้าที่

- ดูประกาศได้
- ดูกำหนดการการดำเนินงานของ XX ได้
- สนทนากับ XX ได้
- อัปโหลดเอกสารที่เกี่ยวข้องเข้าสู่ฐานข้อมูลได้

##### 2. นักศึกษา

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

#### 6.2 ปัญหาและอุปสรรคในการพัฒนา

- ไลบรารี (Library) ที่ใช้ในการกรอกข้อมูลลงเอกสาร pdf ไม่รองรับภาษาไทย  
แนวทางการแก้ไข : เปลี่ยนชั้นตอนการทำงานเป็นถ่ายภาพเอกสารฉบับจริงแล้วทำการส่งให้เจ้าหน้าที่ตรวจสอบเพื่อยืนยันความถูกต้อง
- เนื่องจากทางผู้พัฒนามีความประสงค์ให้ระบบนี้สามารถใช้งานได้จริง ดังนั้น การพัฒนาในตอนนี้ยังมีข้อจำกัดเรื่องขนาดของเอกสารที่จัดเก็บบนไฟร์เบสที่สามารถอัปโหลดเข้าสู่ระบบสูงสุดเพียง 5 GB ซึ่งหากระบบถูกใช้งานจริงจำนวนข้อมูลในระบบจะเกินจำนวนที่ไฟร์เบสให้ใช้งานพรี  
แนวทางการแก้ไข : ทำการบีบอัดข้อมูลให้มีขนาดเล็กลง ส่วนในอนาคตอาจจำเป็นต้อง

ศึกษาแนวทางการสร้างเซิฟเวอร์ (Server) เป็นของระบบเอง

### 6.3 แนวทางการพัฒนาต่อ

1. สร้าง Web server ของระบบซึ่งเป็นโปรแกรมที่มีหน้าที่ให้บริการด้านการจัดการเว็บไซต์ และ Database server ซึ่งเป็นโปรแกรมที่ทำหน้าที่ให้บริการด้านการจัดการดูแลข้อมูลต่าง ๆ ภายใต้เว็บไซต์ โปรแกรมที่มีการใช้งานส่วนใหญ่เป็น mysql, postgresql, DB2
2. การพัฒนาส่วนแจ้งเตือนล่วงหน้าก่อนถึงกำหนดการต่าง ๆ
3. การพัฒนาส่วนการกรอกข้อมูลตามแบบฟอร์มของ XX ผ่านระบบได้
4. การพัฒนาให้ระบบเป็นระบบ e-document ทั้งระบบโดยเกี่ยวข้องกับผู้ใช้ได้แก่ XX YY และ ZZ

## บรรณานุกรม

- [1] ทศพล ตันสมบัติ. (ม.ป.ป.). ระบบปฏิบัติการ android [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <https://beerkung.wordpress.com/ระบบปฏิบัติการรุ่นล่าส/ระบบปฏิบัติการ-android.html> .
- [2] Google. (2557). Open handset alliance [ออนไลน์]. สืบค้นเมื่อ 20 พฤษภาคม 2561. จาก <http://www.openhandsetalliance.com/> .
- [3] กอบเกียรติ สารอุบล. 2549. การพัฒนา App Android. กรุงเทพฯ: มีเดีย เนทเวิร์ค.
- [4] Sleeping For Less. (2557). Activity life cycle [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <http://www.akexorcist.com/2016/04/why-do-we-need-to-know-about-activity-life-cycle-th.html> .
- [5] ดร.วีระศักดิ์ ชิงถาวร. 2545. JAVA PROGRAMMING Volume I. กรุงเทพฯ: ชีเอ็ดยู เคชั่น.
- [6] mindphp. (2555). Javascript คืออะไร [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <https://goo.gl/FAeTb2> .
- [7] Vue.js Team. (2560). Introduction [ออนไลน์]. สืบค้นเมื่อ 2 พฤษภาคม 2561. จาก <https://vuejs.org/v2/guide/> .
- [8] jhansireddy. (2557). Androidscannerdemo [ออนไลน์]. สืบค้นเมื่อ 3 เมษายน 2561. จาก <https://www.studentloan.or.th/index.php/aboutus> .
- [9] งานกองทุนเงินให้กู้ยืมเพื่อการศึกษา มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่. (2554). กยศ.ม.อ(psu studentloan) [ออนไลน์]. สืบค้นเมื่อ 20 พฤษภาคม 2561. จาก <https://studentloan.psu.ac.th/home> .

- [10] Tni.Student. (2560). แอปพลิเคชัน estudentloan [ออนไลน์]. สืบค้น เมื่อ 20 พฤษภาคม 2561. จาก <https://play.google.com/store/apps/details?id=th.co.dest.anek.studentloan>.
- [11] Kunchit Phiu-Nual. (2557). ความหมายและความสำคัญของ system architecture [ออนไลน์]. สืบค้นเมื่อ 12 พฤษภาคม 2561. จาก <https://goo.gl/6ZhGQo>.
- [12] Atthaboon S. (2555). Black-box testing strategy [ออนไลน์]. สืบค้นเมื่อ 20 พฤษภาคม 2561. จาก <http://everybitsconsult.com/blog/2015/06/22/black-box-testing.html>.

ภาคผนวก

## ภาคผนวก ก

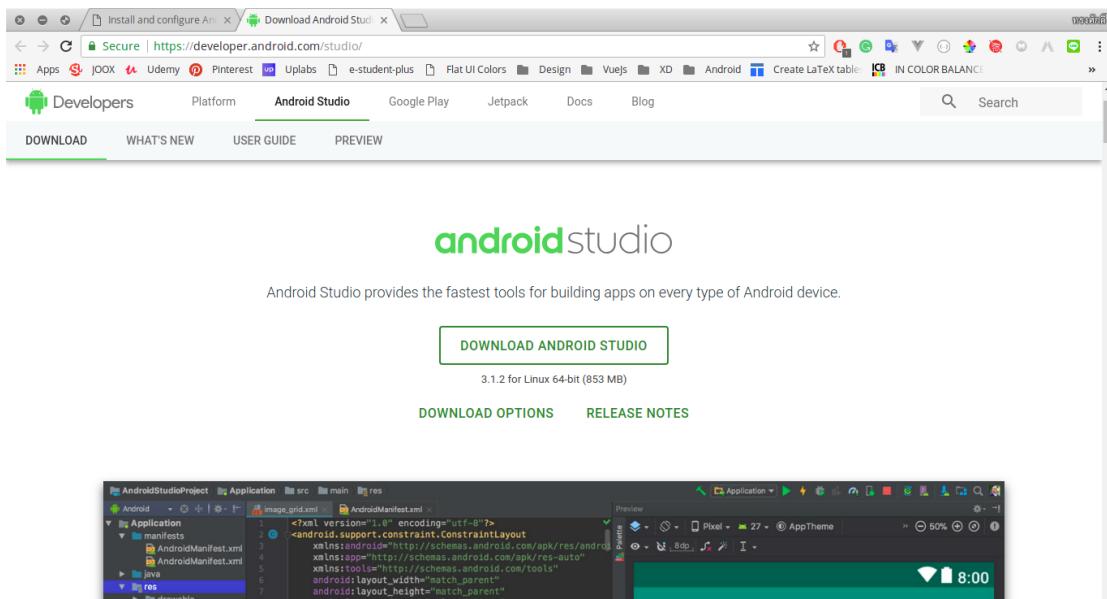
### การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้

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

#### ก.1 การติดตั้ง Android Studio

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



รูปที่ ก.1: หน้าเว็บดาวน์โหลด Android Studio

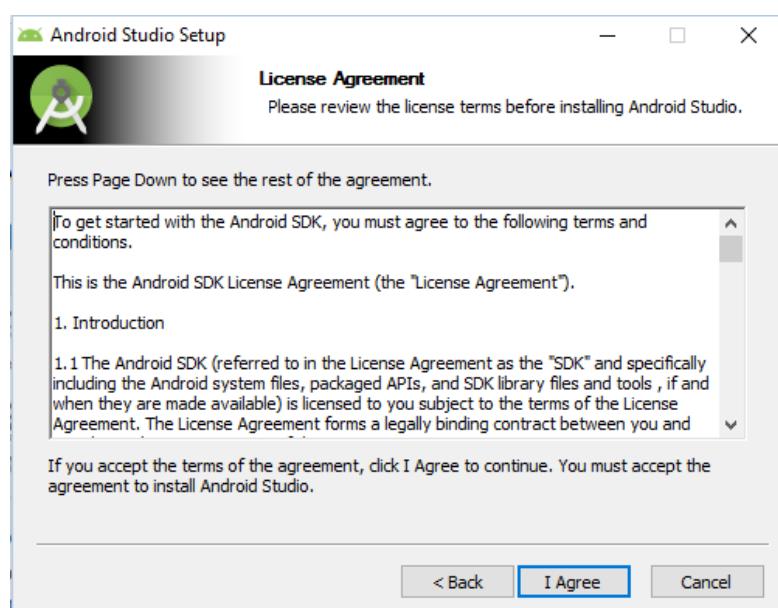
2. แสดงหน้าต่างต้อนรับของ Android Studio ทำการกด Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.2



รูปที่ ก.2: หน้าต่างต้อนรับของ Android Studio

3. แสดงหน้าต่างข้อตกลงการใช้งาน Android Studio ทำการกด I Agree ดังแสดงในรูปที่ ก.3

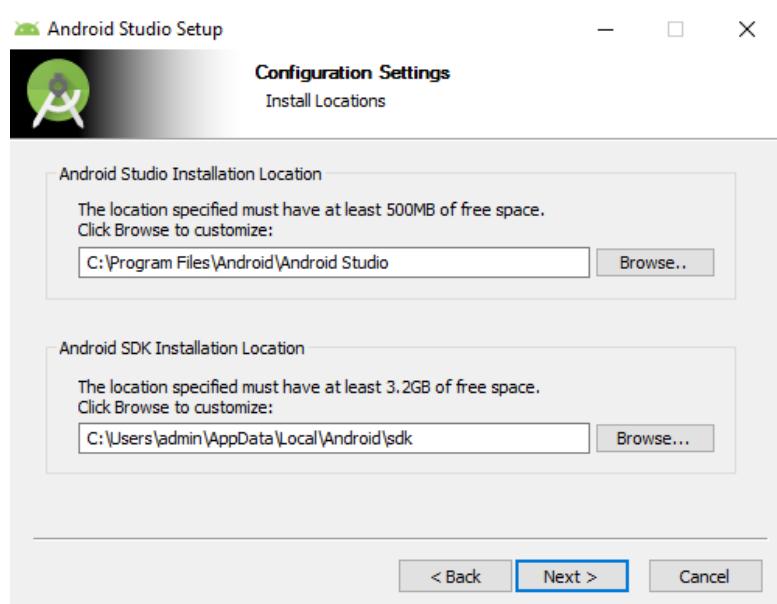
3



รูปที่ ก.3: หน้าต่างข้อตกลงการใช้งาน Android Studio

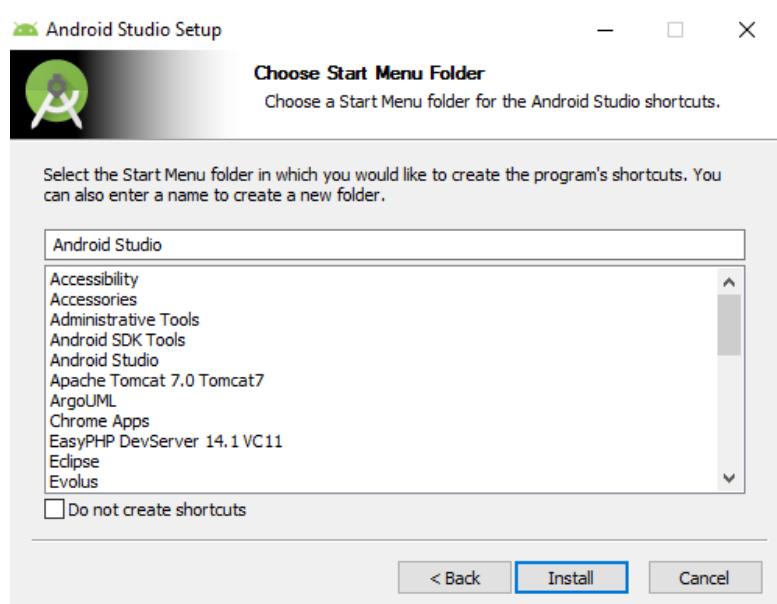
4. แสดงหน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Android Studio ทำการกด Next ดังแสดงในรูปที่ ก.

4



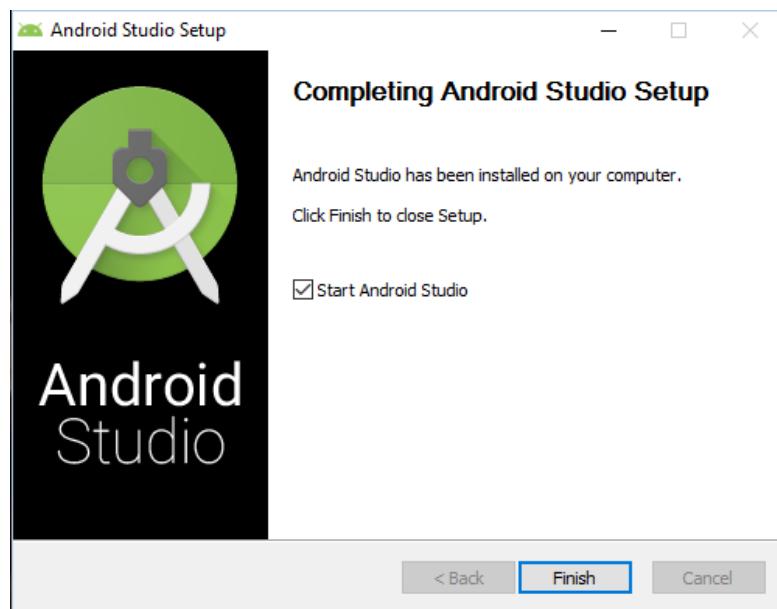
รูปที่ ก.4: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Android Studio

5. หน้าต่างเริ่มทำการติดตั้งทำการกด Install ดังแสดงในรูปที่ ก.5



รูปที่ ก.5: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Android Studio

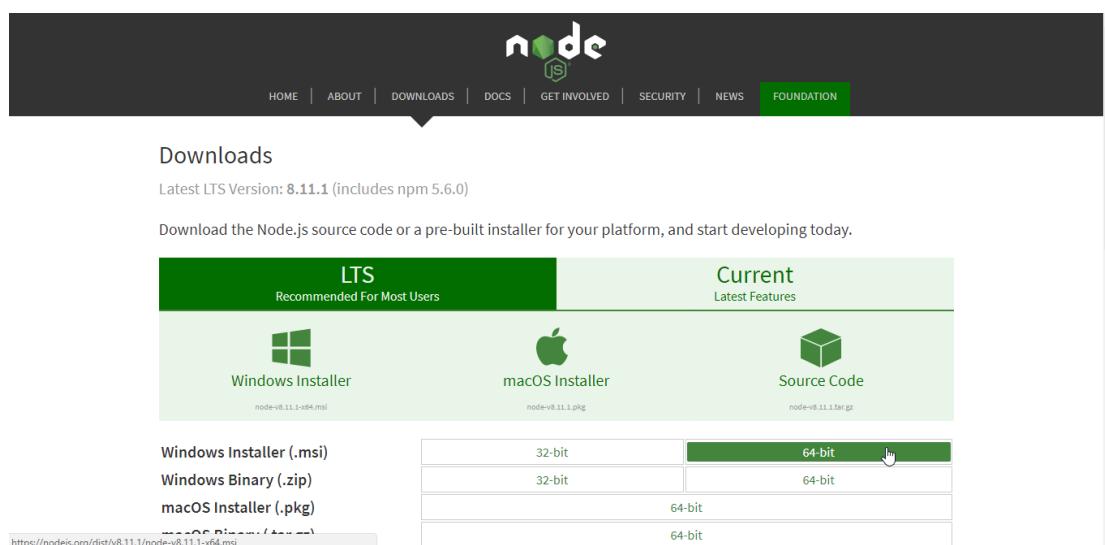
6. หน้าต่างผลการติดตั้ง Android Studio ดังแสดงในรูปที่ ก.6



รูปที่ ก.6: หน้าต่างผลการติดตั้ง Android Studio

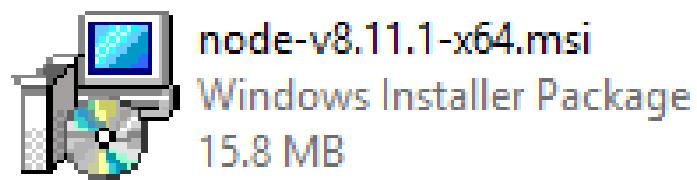
## ก.2 การติดตั้ง Node.js

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



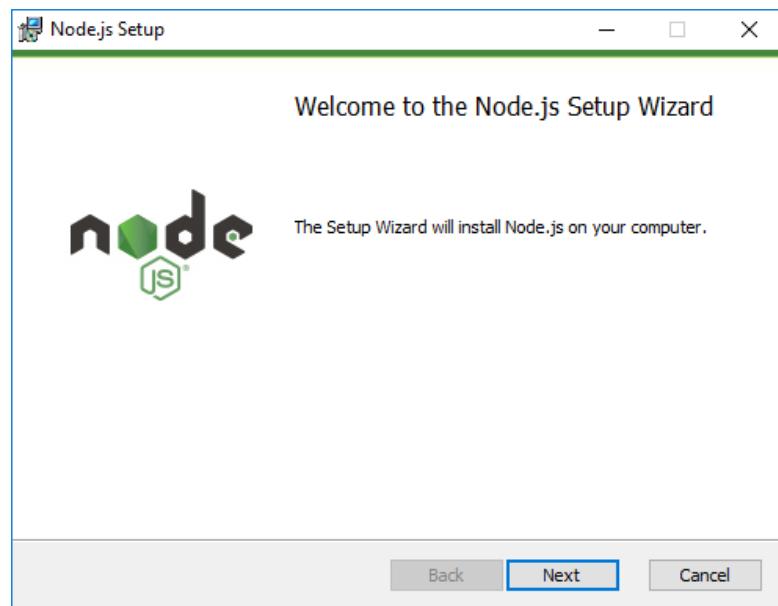
รูปที่ ก.7: หน้าเว็บดาวน์โหลด Node.js

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



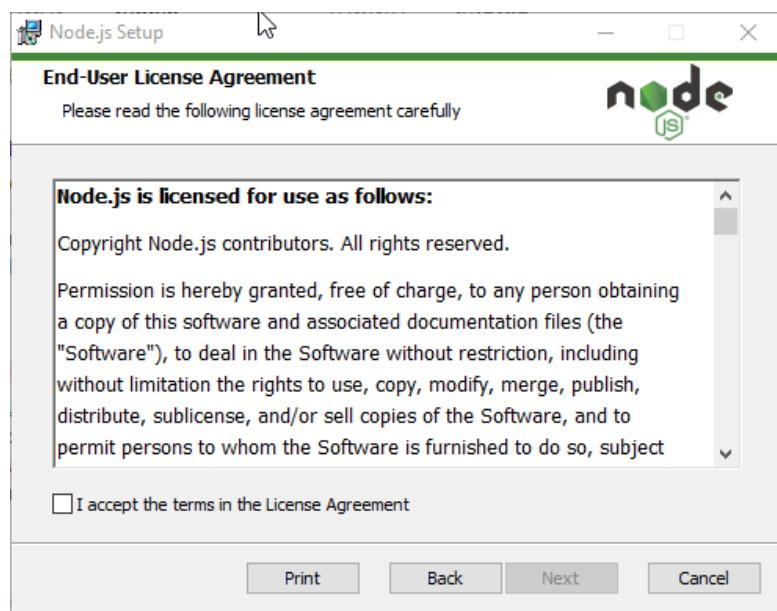
รูปที่ ก.8: ไฟล์ติดตั้งสำหรับติดตั้ง Node.js

3. แสดงหน้าต่างต้อนรับของ Node.js ทำการกด Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.9



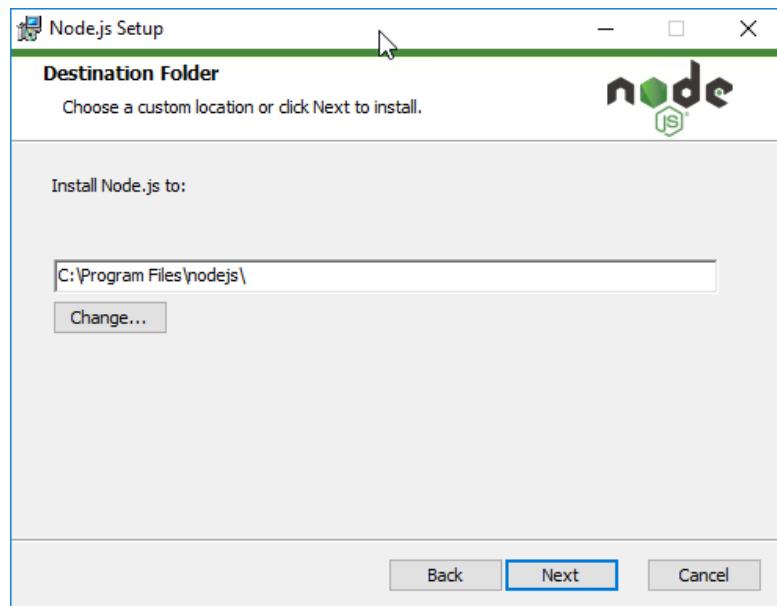
รูปที่ ก.9: หน้าต่างต้อนรับของ Node.js

4. แสดงหน้าต่างข้อตกลงในการใช้ Node.js ให้เลือกช่อง I accept the terms in the License Agreement และกด Next ดังแสดงในรูปที่ ก.10



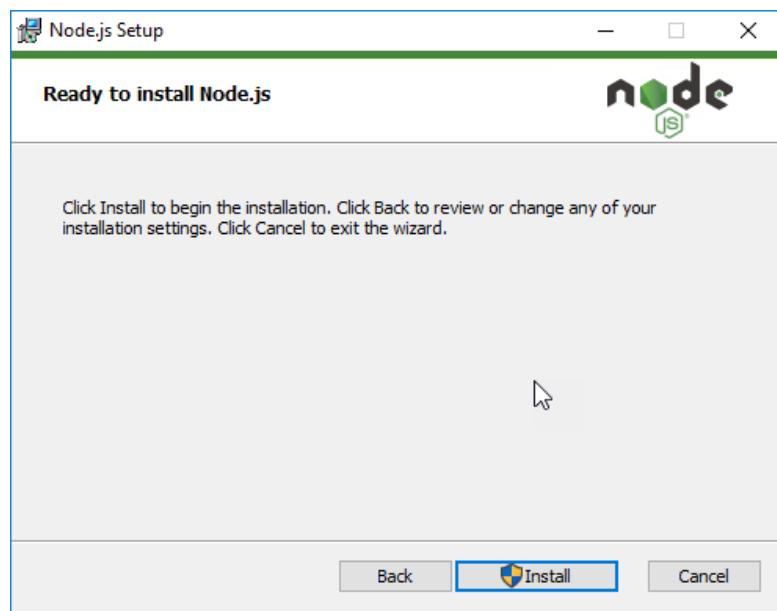
รูปที่ ก.10: หน้าต่างข้อตกลงในการใช้ Node.js

5. แสดงหน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง ดังแสดงในรูปที่ ก.11



รูปที่ ก.11: หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js

6. แสดงหน้าต่างสำหรับติดตั้ง Node.js ทำการกด Install เพื่อทำการติดตั้ง ดังแสดงในรูปที่ ก.12



รูปที่ ก.12: หน้าต่างติดตั้ง Node.js

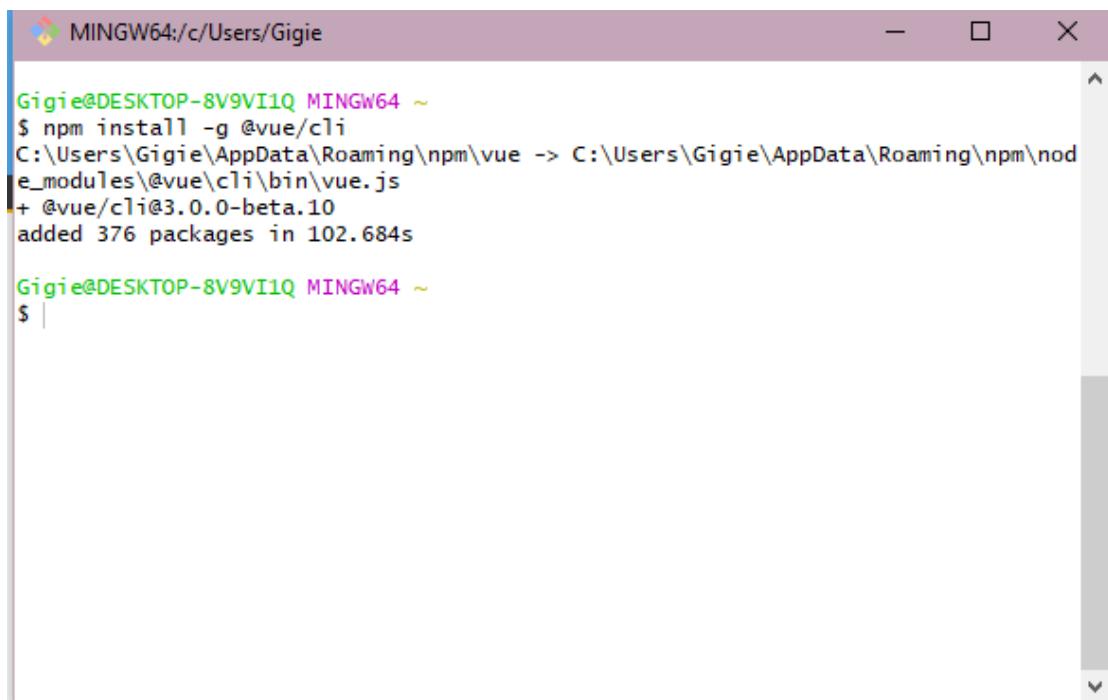
### ก.3 การติดตั้ง Vue.js Fronted Framework

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

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

รูปที่ ก.13: คำสั่งสำหรับติดตั้ง Vue.js Fronted Framework

ผลการติดตั้ง Vue.js Frontend Framework ดังแสดงในรูปที่ ก.14



```
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 ~
$ |
```

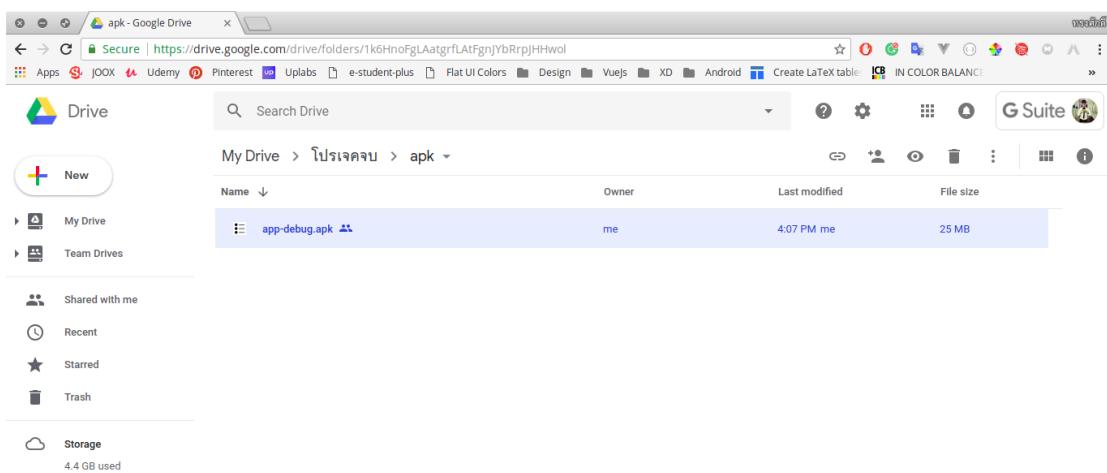
รูปที่ ก.14: หน้าต่างผลการติดตั้ง

## ภาคผนวก ข

### คู่มือการติดตั้งระบบ

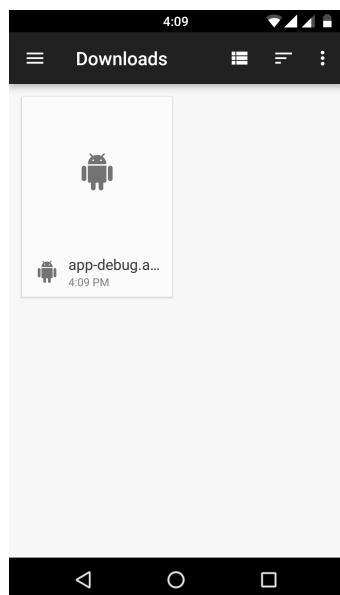
ในการติดตั้งเพื่อใช้งานแอปพลิเคชันระบบของทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี หรือ ESP สามารถทำได้โดยมีขั้นตอนดังนี้

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



รูปที่ ข.1: หน้าเว็บดาวน์โหลด ESP installer package

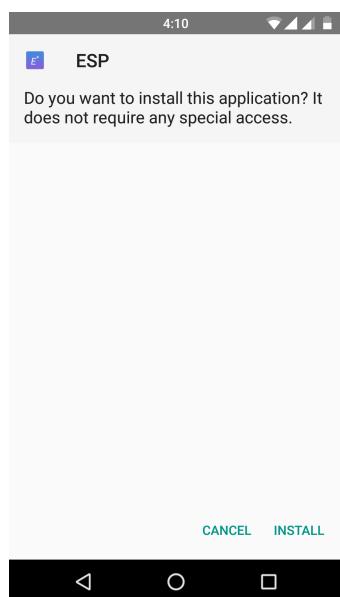
- ตัดลอกไฟล์ app-debug.apk ที่อยู่ในแฟ้มงาน(Folder)ที่อยู่บนคอมพิวเตอร์ไปไว้ในหน่วยความจำบนอุปกรณ์ที่ต้องการ ดังแสดงในรูปที่ ข.2



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

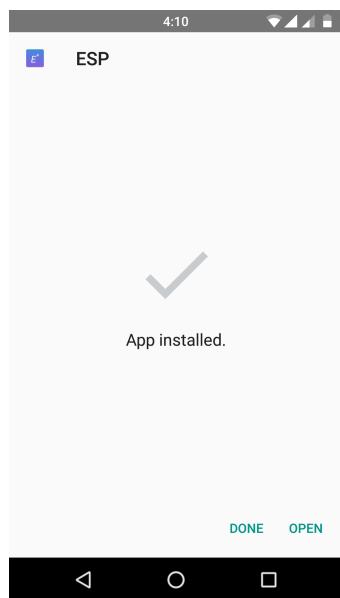
3. ทำการเปิดไฟล์ app-debug.apk และกด INSTALL เพื่อทำการติดตั้ง ดังแสดงในรูปที่ ข.

3



รูปที่ ข.3: หน้าจอต้อนรับการติดตั้งแอปพลิเคชันบนอุปกรณ์เอนдрอย์

4. เมื่อทำการติดตั้งแอปพลิเคชันสำเร็จระบบจะแสดงผล ดังแสดงในรูปที่ ข.4



รูปที่ ข.4: หน้าจอติดตั้งแอปพลิเคชันบนอุปกรณ์แอนดรอย์

## ภาคผนวก ค

### คู่มือการใช้งานระบบ

คู่มือการใช้งานทั้งหมดของระบบ สามารถแบ่งออกเป็น 2 ส่วน ดังนี้

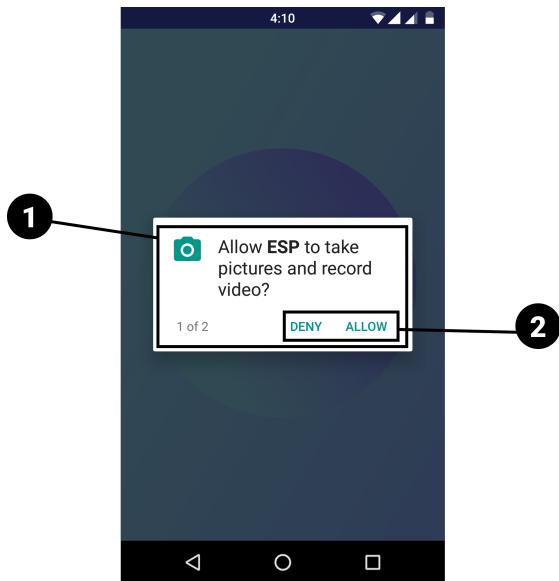
#### 1. ส่วนของหน้าเมนูแอปพลิเคชันสำหรับนักศึกษา

- หน้าจอต้อนรับแสดงผลทุกรอบ เมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ ค.8



รูปที่ ค.1: หน้าจอต้อนรับ

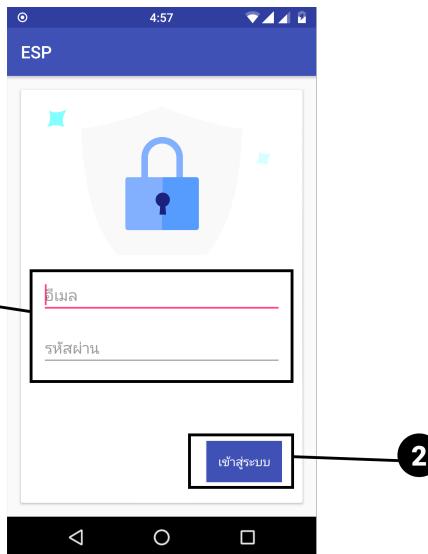
- เมื่อระบบทำการตรวจสอบว่ามีสิทธิ์(Permission)ในการใช้งานแอปพลิเคชันที่ผู้ใช้ยังไม่ได่อนุญาตให้เข้าถึง ระบบจะแสดงหน้าต่างขอสิทธิ์การเข้าถึง ดังแสดงในรูปที่ ค.9



รูปที่ ค.2: หน้าต่างขอสิทธิ์การเข้าถึง

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

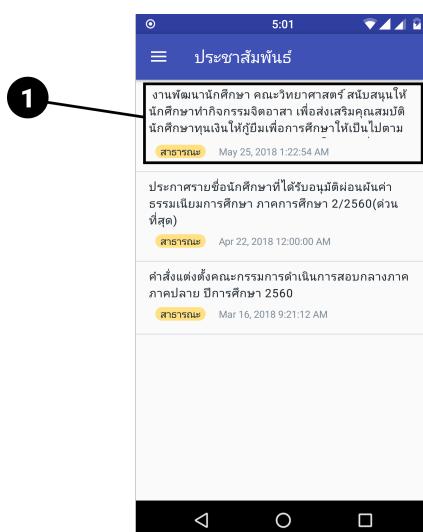
- หมายเลขอ 1 คือ หน้าต่างขอสิทธิ์การเข้าถึงข้อมูล
- หมายเลขอ 2 คือ ปุ่มให้สิทธิ์และยกเลิกการให้สิทธิ์
- ระบบทำการตรวจสอบทุกครั้งเมื่อผู้ใช้งานเปิดใช้งานแอปพลิเคชัน หากผู้ใช้งานคนปัจจุบันยังไม่ได้เข้าสู่ระบบ ระบบจะแสดงหน้าจอเข้าสู่ระบบโดยผู้ใช้งานจำเป็นต้องทำการกรอกข้อมูลคือ อีเมลและรหัสผ่าน ดังแสดงในรูปที่ 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

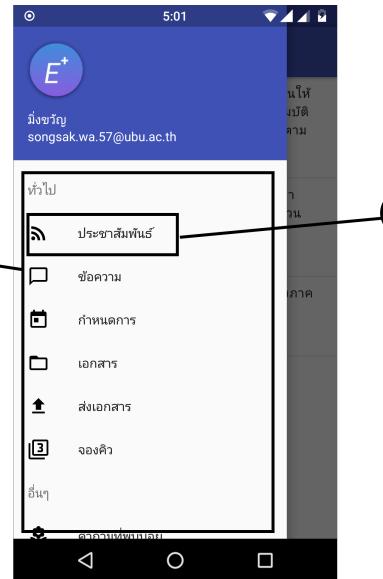
#### จากรูปที่ ค.4 สามารถอธิบายการใช้งานได้ดังนี้

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



รูปที่ ค.5: หน้ารายละเอียดของข่าวสาร

- จากรูปที่ ค.5 สามารถอธิบายการใช้งานได้ดังนี้
- หมายเลขอ 1 คือ หัวข้อข่าวสาร
  - หมายเลขอ 1 คือ รายละเอียดของข่าวสาร
  - เมื่อผู้ใช้กดเลือกเมนู Drawer ระบบจะแสดงเมนูนำทางหลักของแอปพลิเคชันซึ่งประกอบไปด้วยเมนู ประชาสัมพันธ์ ข้อความ กำหนดการ เอกสาร ส่งเอกสาร จอง คิว คำถามที่พบบ่อย เกี่ยวกับเรา บัญชีผู้ใช้และออกจากระบบ ดังแสดงในรูปที่ ค.6

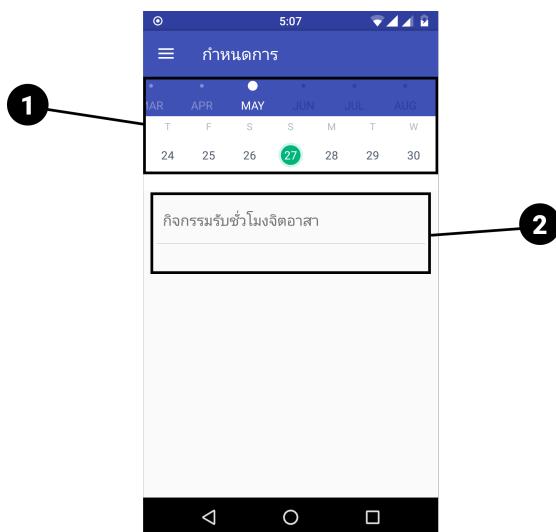


รูปที่ ค.6: เมนูนำทางหลักของแอปพลิเคชัน

จากรูปที่ ค.6 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ รายการเมนูนำทางหลักของแอปพลิเคชัน
- หมายเลข 2 คือ เมนูนำทาง
- เมื่อผู้ใช้เลือกเมนูกำหนดการ ระบบจะแสดงหน้าจอกำหนดการ ดังแสดงในรูปที่ ค.

10



รูปที่ ค.7: หน้าจอกำหนดการ

- ส่วนของหน้าเมนูแอปพลิเคชันสำหรับนักศึกษา
- หน้าจอต้อนรับแสดงผลทุกรูปเมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ ค.8



รูปที่ ค.8: หน้าจอต้อนรับ

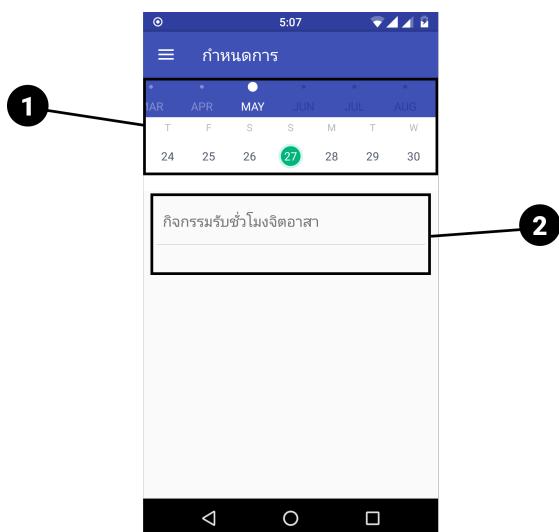
- เมื่อระบบทำการตรวจสอบว่ามีสิทธิ์(Permission)ในการใช้งานแอปพลิเคชันที่ผู้ใช้ยังไม่ได้ออนุญาตให้เข้าถึง ระบบจะแสดงหน้าต่างขอสิทธิ์การเข้าถึง ดังแสดงในรูปที่ ค.9



รูปที่ ค.9: หน้าต่างขอสิทธิ์การเข้าถึง

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ หน้าต่างขอสิทธิ์การเข้าถึงข้อมูล
- หมายเลข 2 คือ ปุ่มให้สิทธิ์และยกเลิกการให้สิทธิ์



รูปที่ ค.10: หน้าแสดงกำหนดการ

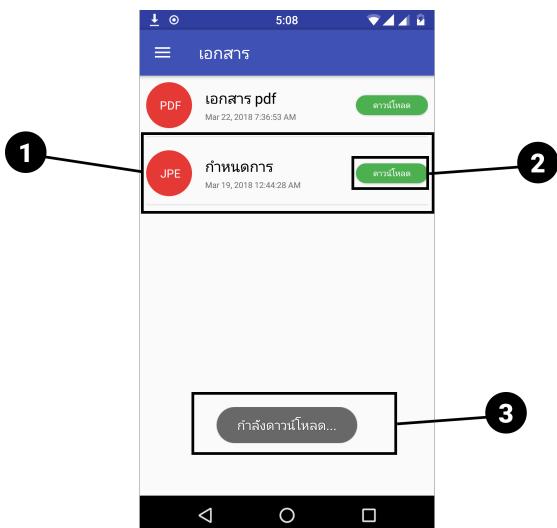
จากรูปที่ ค.10 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปฏิทินแสดงวันที่ปัจจุบันและผู้ใช้สามารถเลือกวันที่ต้องการเพื่อ

### ดูกำหนดการก่อนหน้า

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

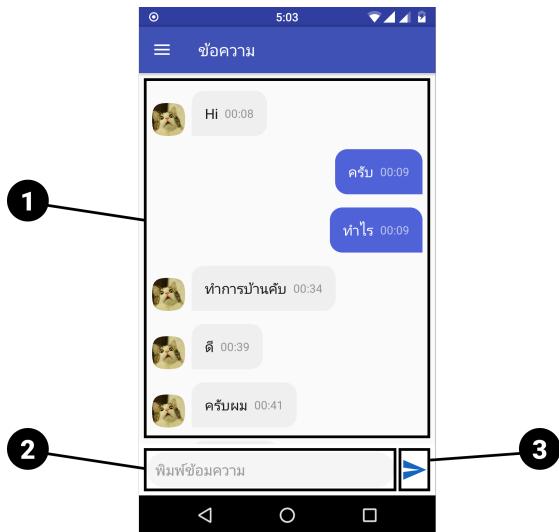
11



รูปที่ ค.11: หน้าจอเอกสาร

### จากรูปที่ ค.11 สามารถอธิบายการใช้งานได้ดังนี้

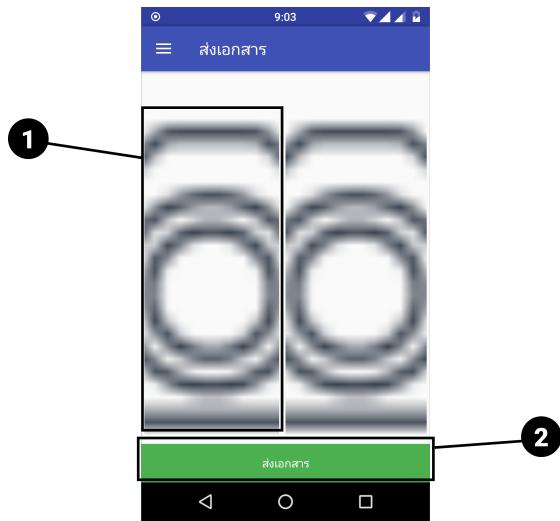
- หมายเลข 1 คือ รายการเอกสาร
- หมายเลข 2 คือ ปุ่มดาวน์โหลดเอกสาร
- หมายเลข 3 คือ สถานะการดาวน์โหลดเอกสาร
- เมื่อผู้ใช้เลือกเมนูข้อความ ระบบจะแสดงหน้าจอสนทนา ดังแสดงในรูปที่ ค.12



รูปที่ ค.12: หน้าจอสนทนากู

จากรูปที่ ค.12 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ รายการประวัติการสนทนา
- หมายเลข 2 คือ ช่องกรอกข้อความเพื่อสนทนา
- หมายเลข 3 คือ ปุ่มส่งข้อความ
- เมื่อผู้ใช้เลือกเมนูส่งเอกสารระบบจะตรวจสอบข้อมูลว่าเจ้าหน้าที่ได้ทำการเปิดให้นักศึกษาส่งเอกสารได้หรือไม่ หากตรวจสอบแล้วพบว่าสามารถส่งได้ระบบจะแสดงหน้าจอส่งเอกสาร ดังแสดงในรูปที่ ค.13



รูปที่ ค.13: หน้าจอส่งเอกสาร

จากรูปที่ ค.13 สามารถอธิบายการใช้งานได้ดังนี้

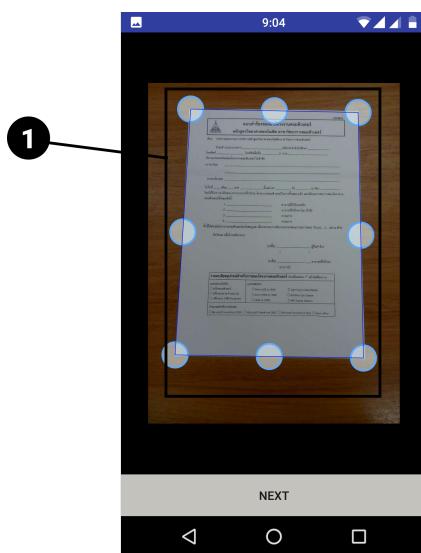
- หมายเลข 1 คือ เมื่อผู้ใช้เข้ามาครั้งแรกเมื่อผู้ใช้กดรูปภาพเอกสารระบบจะนำผู้ใช้ไปยังหน้าจอถ่ายภาพเอกสารและแสดงรูปภาพพรีวิว(Preview)ภาพถ่ายเอกสาร
- หมายเลข 2 คือ ปุ่มกดส่งเอกสาร
- เมื่อผู้ใช้กดที่ปุ่มถ่ายภาพเอกสารระบบจะแสดงหน้าจอถ่ายภาพเอกสาร ตั้งแสดงในรูปที่ ค.14



รูปที่ ค.14: หน้าจอถ่ายภาพเอกสาร

จากรูปที่ ค.14 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มกดถ่ายภาพสำเนาเอกสาร
- หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร ดังแสดงในรูปที่ ค.15

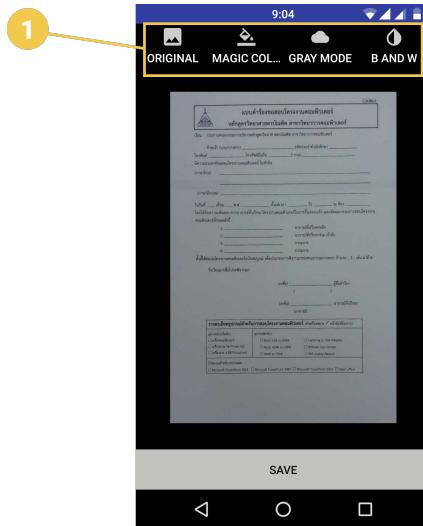


รูปที่ ค.15: หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.15 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปุ่มปรับมุมภาพทั้ง 8 มุมเพื่อปรับขนาดภาพถ่ายเอกสาร

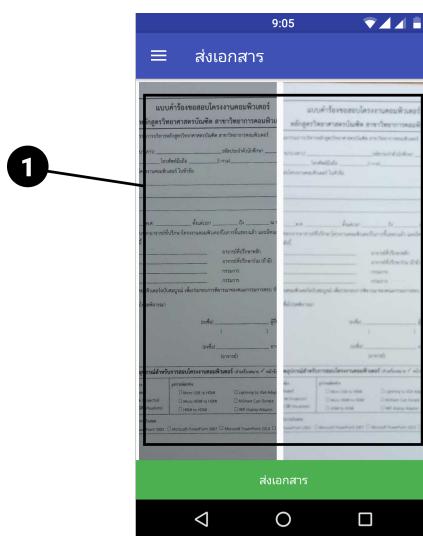
- หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร ดังแสดงในรูปที่ ค.16



รูปที่ ค.16: หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.16 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขอื่นๆ คือ เมื่อผู้ใช้ต้องการปรับความคมชัดของภาพสามารถทำได้จาก 4 ปุ่ม คือ ปุ่มภาพตันฉบับ ปุ่มปรับสีอัตโนมัติ ปุ่มปรับปรับสีเป็นสีเทา ปุ่มปรับสีเป็นสีขาวดำ
- หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสารทั้งสองฉบับ ดังแสดงในรูปที่ ค.17



รูปที่ ค.17: หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.17 สามารถอธิบายการใช้งานได้ดังนี้

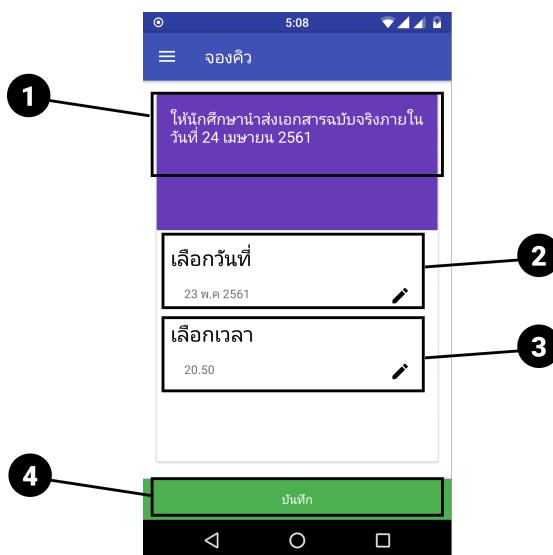
- หมายเลข 1 คือ แสดงภาพพรีวิวภาพถ่ายสำเนาเอกสารทั้งสองฉบับ
- เมื่อผู้ใช้กดปุ่มส่งเอกสาร ระบบจะแสดงหน้าต่างแสดงสถานะการอัพโหลดเอกสาร ดังแสดงในรูปที่ ค.18



รูปที่ ค.18: หน้าต่างแสดงสถานะการอัพโหลดเอกสาร

จากรูปที่ ค.18 สามารถอธิบายการใช้งานได้ดังนี้

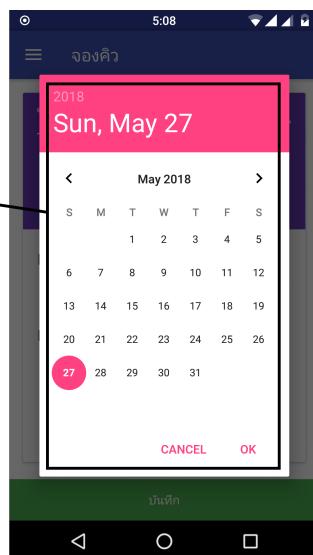
- หมายเลข 1 คือ หน้าต่างสถานะการอัพโหลดเอกสาร
- เมื่อผู้ใช้กดเมนูจองคิว ระบบจะตรวจสอบสถานะการส่งเอกสารว่าถูกเจ้าหน้าที่ตรวจ  
สอบแล้วหรือไม่ ถ้าเจ้าหน้าที่อนุมัติแล้วระบบจะแสดงหน้าจอของวันที่ส่งเอกสาร ดัง  
แสดงในรูปที่ ค.19



รูปที่ ค.19: หน้าจอจ่องคิว

จากรูปที่ ค.19 สามารถอธิบายการใช้งานได้ดังนี้

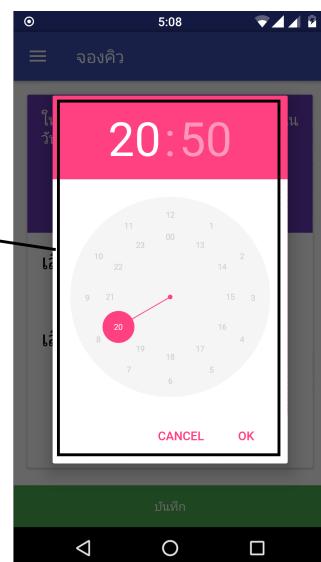
- หมายเลข 1 คือ รายละเอียด
- หมายเลข 2 คือ ส่วนของการเลือกวันที่ที่ต้องการส่งเอกสาร
- หมายเลข 3 คือ ส่วนของการเลือกเวลาที่ต้องการส่งเอกสาร
- หมายเลข 4 คือ ปุ่มกดส่งเอกสาร
- เมื่อผู้ใช้กดปุ่มเลือกวันที่ที่ต้องการส่งเอกสารระบบจะแสดงหน้าต่างเลือกวันที่โดยจะแสดงเฉพาะวันที่ที่เจ้าหน้าที่เลือกไว้เท่านั้น ดังแสดงในรูปที่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

- หมายเลขอ 1 คือ หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร
- เมื่อผู้ใช้กดเมนูคำถามที่พับบอยระบบจะแสดงหน้าจอคำถามที่พับบอย ดังแสดงในรูปที่ ค.22



รูปที่ ค.22: หน้าจอคำถามที่พับบอย

จากรูปที่ ค.22 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลขอ 1 คือ คำถาม
- หมายเลขอ 2 คือ คำตอบ
- เมื่อผู้ใช้กดปุ่มเมนูเกี่ยวกับเราระบบจะแสดงข้อมูลรายละเอียดของงานพัฒนานักศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ดังแสดงในรูปที่ ค.23



รูปที่ ค.23: หน้าเกี่ยวกับ

จากรูปที่ ค.23 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ข้อมูลติดต่องานพัฒนานักศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี
- ปุ่มกดสำหรับเปิดกลุ่มงานพัฒนานักศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานีบน Facebook

## ประวัติผู้เขียน

ชื่อ-สกุล: นายชื่อ สกุล

รหัสประจำตัวนักศึกษา: 5811400000

วัดเกิด: XX YY 25ZZ

ที่อยู่ที่สามารถติดต่อได้: XX หม.XX ต.XX อ.XX จ.XX 34XXX

เบอร์โทรศัพท์: (+66) XX XXX XXXX

อีเมลล์: xxxxxxxx.yy.59@ubu.ac.th

ระดับมัธยมต้น: โรงเรียน XX จังหวัด XX

ระดับมัธยมปลาย: โรงเรียน XX จังหวัด XX

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สาขาวิชาระบบทั่วไป คณะพัฒนา  
วิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี