ระบบจองคิวร้านเสริมสวย
Beauty salon queue reservation system

นางสาวปิยพร อาภรศรี

โครงงานนี้เป็นส่วนหนึ่งของการศึกษา
หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยอุบลราชธานี
ปีการศึกษา 2562

โครงงาน	:	ระบบจองคิวร้านเสริมสวย Beauty salon queue reservation system
โดย	:	นางสาวปิยพร อาภรศรี
อาจารย์ที่ปรึกษา	:	ดร.ทศพร จูฉิม
ระดับการศึกษา	:	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา	:	2562

ได้รับการพิจารณาให้เป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบประเมินความรู้โครงงานคอมพิวเตอร์

. . . . . อาจารย์ที่ปรึกษา
(ดร.ทศพร จูฉิม)

. . . . . กรรมการ
(อาจารย์ วาโย ปุยะติ)

. . . . . กรรมการ
(อาจารย์ วาสนา เหง้าเกษ)

. . . . . หัวหน้าภาควิชา
(ผศ.ดร. สุพจน์ สีบุตร)

วันที่ . . . / . . . / . . .

กิตติกรรมประกาศ

การพัฒนาโครงงานระบบจองคิวร้านเสริมสวย สำเร็จลุล่วงได้ด้วยความกรุณาแลความช่วยเหลือจากหลายๆ ท่าน ข้าพเจ้าขอขอพระคุณทุกท่าน ที่มีส่วนร่วมในการพัฒนาโครงงานนี้

ขอขอบพระคุณอาจารย์ ดร ทศพร จูฉิม อาจารย์ที่ปรึกษาโครงงานที่ได้แนะนำทฤษฎีและแนวทางในแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างการพัฒนาระบบ อีกครั้งยังคอยตรวจสอบความก้าวหน้าของการทำงานเป็นระยะ ๆ รวมทั้งสร้างกำลังใจให้ผู้พัฒนาอยู่เสมอ

ขอบพระคุณอาจารย์ประจำสาขาวิทยาการคอมพิวเตอร์ อาจารย์ประจำภาควิชาคณิตศาสตร์สถิติ และคอมพิวเตอร์ และอาจารย์ในคณะวิทยาศาสตร์ทุก ๆ ท่าน ที่คอยให้คำแนะนำ อบรมสั่งสอน และคอยช่วยเหลือข้าพเจ้าในการศึกษาตลอดมาขอบคุณเจ้าหน้าที่และบุคลากรของคณะวิทยาศาสตร์ ที่ได้อำนวยความสะดวกทางด้านอุปกรณ์และเครื่องมือต่าง ๆ

ขอบพระคุณบิดา มารดา ที่เป็นกำลังใจ คอยให้ความรักและความห่วงใยเสมอมา ตลอดจนคอยช่วยเหลือทุนทรัพย์ทางด้านการศึกษา และอุปกรณ์ในการพัฒนาโครงงาน

ขอบคุณเพื่อน ๆ สาขาวิทยาการคอมพิวเตอร์ชั้นปีที่ 4 ที่ได้คอยช่วยแก้ไขปัญหาและให้คำปรึกษาในการพัฒนาโครงงานครั้งนี้จนเสร็จสิ้น

นางสาวปิยพร อาภรศรี

17 มีนาคม 62

โครงงาน	:	ระบบจองคิวร้านเสริมสวย
โดย	:	นางสาวปิยพร อาภรศรี
อาจารย์ที่ปรึกษา	:	ดร.ทศพร จูฉิม
ระดับการศึกษา	:	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
ปีการศึกษา	:	2562

บทคัดย่อ

การเสริมสวยเป็นที่นิยมและเป็นธุรกิจที่ได้รับความนิยมจากลูกค้าเป็นจำนวนมากในปัจจุบัน โดยปกติเมื่อลูกค้าต้องการใช้บริการจะมาที่ร้านโดยไม่ได้นัดหมาย ซึ่งปัญหาคือร้านมีลูกค้าที่ใช้บริการอยู่ในขณะนั้นทำให้ต้องผู้ที่เข้ามาโดยไม่ได้นัดต้องรอคิวหรือถ้ามีลูกค้ากำลังรอรับบริการอยู่เป็นจำนวนมากอาจทำให้ต้องมาใช้บริการในวันอื่นแทน แม้ในบางกรณีที่ลูกค้าโทรมาสอบถามเพื่อทำการจองคิวล่วงหน้า แต่ช่างไม่สะดวกรับโทรศัพท์เนื่องจากกำลังให้บริการลูกค้าคนอื่นอยู่

ดังนั้นผู้พัฒนาจึงมีแนวคิดสร้างเว็บแอปพลิเคชันระบบการจองคิวร้านเสริมสวยขึ้น เพื่อช่วยจัดการปัญหาดังกล่าว ระบบนี้ถูกพัฒนาด้วย React framework , material ui , nodejs และ MySql โดยระบบสามารถให้ลูกค้าทำการจองคิว ดูคิวว่างของร้านเสริมสวย หาตำแหน่งของร้าน ดูข้อมูลทั่วไปของร้าน และเขียนรีวิวติชมได้ ในส่วนของเจ้าของร้าน สามารถเพิ่มข้อมูลทั่วไปของร้าน และจัดการการจองคิวของร้านเสริมสวยได้ ระบบการจองคิวรองรับการแสดงผลบนอุปกรณ์สมาร์ทโพนและเว็บบราวเซอร์

ระบบที่พัฒนาขึ้นจะช่วยอำนวยความสะดวกในการนัดหมายล่วงหน้า ลดการรอคิวของผู้ใช้บริการ และช่วยให้การจองคิวมีระเบียบมากขึ้น

คำสำคัญ: เว็บแอปพลิเคชัน , ระบบจองคิวร้านเสริมสวย

Topic	:	Beauty salon queue reservation system
Author	:	PIYAPHORN ARPHORNSRI
Advisor	:	TOSSAPORN JOOCHIM, Ph.D..
Degree	:	Bachelor of Science (Computer Science)
Academic Year	:	2019

Abstract

Beauty is popular and is a business that has been popular with many customers today. Usually when customers want to use the service, will come to the shop without an appointment. The problem is that the shop has customers who use the service at that time, causing those who enter without an appointment to wait in the queue. Or if there are many customers waiting to receive the service, may cause to use the service on another day instead Even in some cases where users call to inquire to reserve a queue in advance But the technician was not comfortable to answer the phone because he was serving other customers Therefore, the developer has the idea to create a salon queue reservation system. To help manage the said problem This system was developed with the React framework, material ui, nodejs and MySql. The system allows customers to reserve a queue. See the salon queue. Find a store location See general information of the store And can write a review In the part of the store owner Can add general information of the shop And can manage the beauty salon queue reservations The queue reservation system supports display on Smart Phon devices and web browsers. The developed system will help facilitate advance appointments. Reduce user waiting And helps to make queue reservations more organized.

Keywords: Web Application,Beauty salon queue reservation system

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ที่มาและเหตุผล

เนื่องจากปัจจุบัน การดำเนินธุรกิจร้านเสริมสวยที่ให้บริการเสริมความงาม เช่น การทำผม ตัดผม ออกแบบทรงผม อบไอน้ำ เวลาผู้ใช้บริการมาใช้บริการโดยจะมาที่ร้านเลยโดยที่ไม่จองคิว พบว่าร้านที่มาใช้บริการมีลูกค้าเป็นจำนวนมาก อาจจะทำให้ต้องรอคิวนานหรือต้องเสียเวลามาใช้บริการในวันอื่นบางครั้งผู้ใช้บริการมีเบอร์ของร้านเสริมสวยก็จะโทรมา สอบถามคิวและจองคิว แต่ช่างติดทำผมให้ลูกค้าท่านอื่นก็ไม่สามารถรับโทรศัพท์ได้ผู้พัฒนาจึงมีแนวคิดว่าจะทำระบบการจอง คิวร้านเสริมสวยขึ้น เพื่อแก้ปัญหาการรอคิวนานและให้มีความทันสมัยตลอดจนสามารถรองรับการแสดงผลบนอุปกรณ์ สมาร์ทโฟนในปัจจุบัน ทำให้สามารถจองคิวหรือติดต่อสื่อสารในเรื่องของการจองคิวทำผมกับทางร้านได้สะดวกมากยิ่งขึ้น

แนวทางการแก้ปัญหา จัดทำการพัฒนาเป็นเว็บแอปพิเคชัน ระบบจองคิวร้านเสริมสวยที่ถูกพัฒนาขึ้นเป็นเว็บแอปพิเคชัน จะช่วยเพิ่มระเบียบในการจัดการจองคิวให้เป็นระบบ ลดขั้นตอนการดำเนินงานที่ซับซ้อน ลดระยะเวลาในการดำเนินงาน ลดความผิดพลาดที่จะเกิดขึ้นในขั้นตอนการดำเนินงาน และช่วยเพิ่มประสิทธิภาพในการทำงาน

1.2 วัตถุประสงค์

1. เพื่อออกแบบและพัฒนาเว็บแอปพลิเคชั่น จองคิวร้านเสริมสวย
2. เพื่อแก้ปัญหาการรอคิวร้านเสริมสวย

1.3 ขอบเขตของโครงงาน

1.3.1 เจ้าของร้าน

- สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
- สามารถดูคิวที่ผู้ใช้บริการได้ทำการจองคิวไว้
- สามารถจัดการคิวได้

- สามารถ post ภาพผลงานทั้งหมดของร้านได้
- สามารถเพิ่ม แก้ไข และลบรายการให้บริการประจำร้านได้
- สามารถเพิ่ม แก้ไข และลบ ข้อมูลร้านได้
- สามารถเพิ่ม แก้ไข และลบข้อมูลตำแหน่งร้านได้

1.3.2 ช่างประจำร้าน

- ลงทะเบียนใช้ web ด้วย Email ได้
- สามารถดูตารางการทำงานของตนเองได้
- สามารถ post ภาพผลงานของตัวเองได้
- สามารถแก้ไขข้อมูลส่วนตัวได้

1.3.3 ผู้ใช้บริการ

- สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
- สามารถค้นหาร้านเสริมสวยได้
- สามารถจองคิวของร้านเสริมสวยได้
- สามารถดูคิวว่างของร้านเสริมสวยได้
- สามารถดูข้อมูลต่างๆของร้านเสริมสวยได้
- สามารถดูตำแหน่งของทางร้านได้
- สามารถดูผลงานของร้านได้
- สามารถเขียนรีวิว ติชม ได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยลดขั้นตอนการจองคิวร้านเสริมสวย
2. ช่วยลดปัญหาในการใช้บริการที่เกิดจากการรอคิวนาน
3. ช่วยให้การจองคิวมีระเบียบมากขึ้น

1.5 เครื่องมือที่ใช้ในการพัฒนา (Development tools)

1.5.1 ฮาร์ดแวร์

1. เครื่องคอมพิวเตอร์ส่วนบุคคล (Personal computer) เพื่อใช้ในการพัฒนาโมบายแอปพลิเคชัน โดยมีคุณสมบัติอย่างน้อยดังนี้

- ทำงานบนระบบปฏิบัติการ Elementary OS พื้นฐานการทำงานบน Windows 10
- หน่วยประมวลผลกลาง AMD Rezen(™) 5 3500U
- หน่วยประมวลผลกราฟฟิก AMD Radeon(™) Vega(8) Mobile Graphics
- หน่วยความจำหลักอย่างน้อย 8 กิกะไบต์ (Gigabyte, GB)
- หน่วยความจำสำรองอย่างน้อย 256 กิกะไบต์ (Gigabyte, GB)

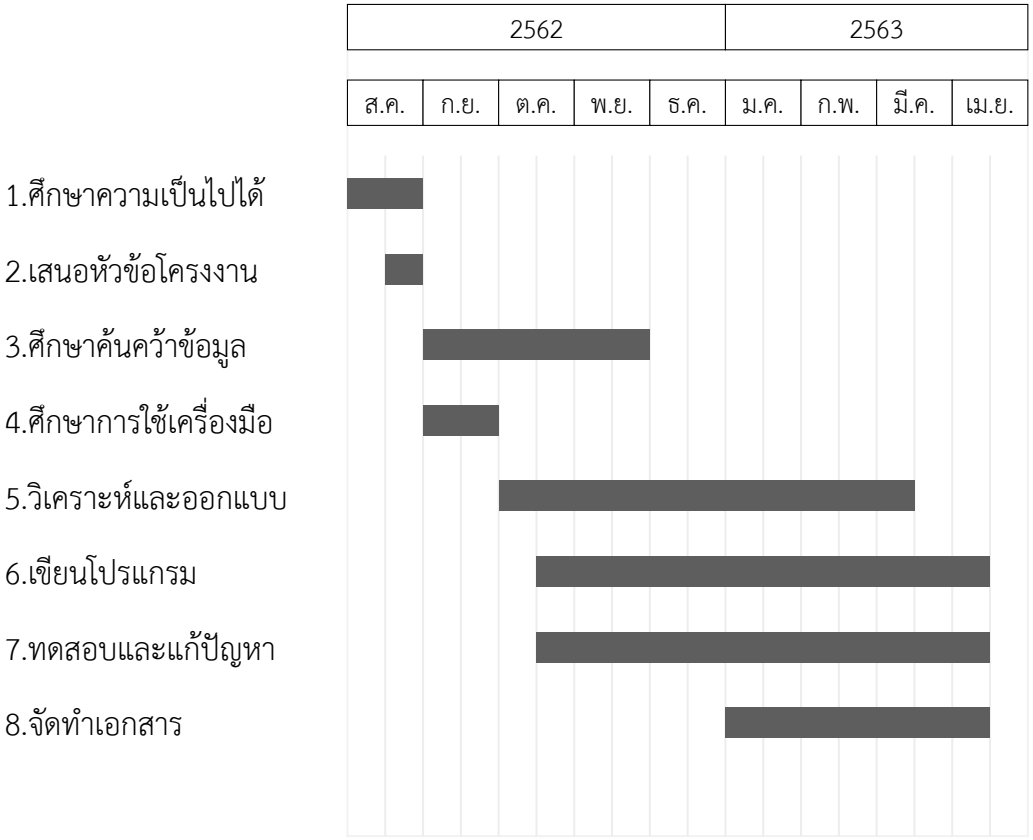
1.5.2 ซอฟต์แวร์ (Software)

1. ReactJS เป็น JavaScript Framework โดยมีชุดคำสั่งและไลบารี่ (Library) ให้ใช้งานมากมาย
2. Node.js คือ Cross Platform Runtime Environment หรือเรียกอีกอย่างว่า Backend Framework ใช้สำหรับเป็นเว็บเซิฟเวอร์ (Web Server) ซึ่งเขียนด้วยภาษา JavaScript
3. JavaScript เป็น ภาษาที่ใช้ในการพัฒนาเว็บ Environment)
4. Xampp เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อทดสอบระบบระหว่างพัฒนา
5. MySQL เป็น โปรแกรมระบบจัดการฐานข้อมูล
6. Google Map API เทคโนโลยีที่ใช้ในการใช้งานแผนที่
7. Visual Studio Code เครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชัน

1.5.3 แผนการดำเนินการ

ในการสร้างระบบแนะนำสถานที่ท่องเที่ยวในจังหวัดอุบลราชธานี ผู้พัฒนาได้แบ่งขั้นตอนการดำเนินงานไว้ด้วยกัน 8 ขั้นตอน ดังตารางที่ 1.1

ตารางที่ 1.1: ขั้นตอนการดำเนินงาน



บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงรายละเอียดเกี่ยวกับทฤษฎีและงานวิจัยที่เกี่ยวของกับการพัฒนาระบบจองคิวร้านเสริมสวย โดยแบ่งเนื้อหาออกเป็น 2 ส่วน ได้แก่ ส่วนที่หนึ่งเป็นเนื้อหาพื้นฐานเกี่ยวกับทฤษฎีการเขียนโปรแกรมและเทคโนโลยีที่นำมาใช้ในการพัฒนาในหัวข้อที่ 2.1 - 2.6 ได้แก่ ความรู้พื้นฐานเกี่ยวกับ React Node.js JavaScript MySQL Visual studio code Google maps API และในส่วนที่สองเป็นเนื้อหาเกี่ยวกับเว็ปแอพพลิเคชันที่เกี่ยวข้องกับโครงงานนี้เว็บแอพพลิเคชัน Gowabi

2.1 ความรู้พื้นฐานเกี่ยวกับ React

React เป็น JavaScript Library ที่ถูกสร้างโดย Facebook ซึ่ง React ทำหน้าที่เป็นเพียง User Interface (UI) ที่สร้างมาจากพื้นฐานแนวความคิดแบบ Model View Controller (MVC)

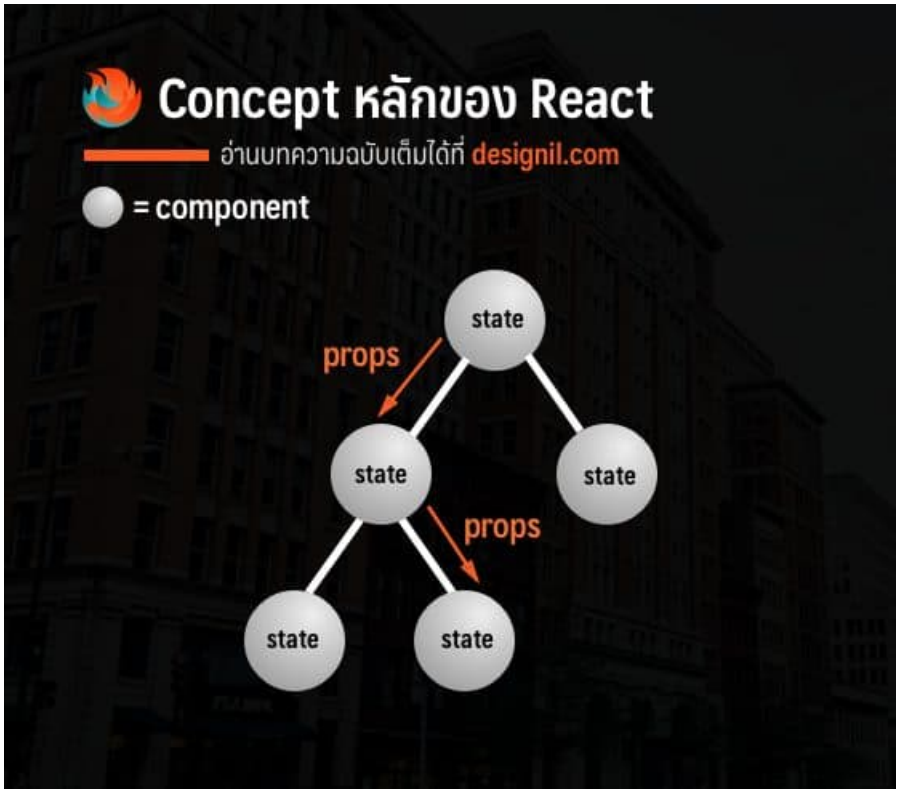
React ทำหน้าที่เฉพาะส่วน View (จาก Model View Controller) เหมาะกับงาน Web Front-End ที่สามารถแบ่งออกเป็น Web Component ย่อยๆ โดยหลักการวิเคราะห์ควรแยกให้ย่อยที่สุดเท่าที่จะทำได้ ซึ่งสามารถแบ่ง Component ออกเป็น 2 รูปแบบ คือ

- Container สำหรับบรรจุ Component หลักย่อยอื่นๆ ซึ่งไม่ควรมีการเก็บค่าใดๆ (สามารถทำหน้าที่เป็นตัวกลางในการส่งผ่านค่าได้) เน้นไปที่การจัด Layout
- Web Component คือ ส่วนที่ต้อง interact กับผู้ใช้จริงๆ เช่น ช่องกรอกข้อมูล ลิสต์แสดงข้อมูล ลาเบล (Label) และ ปุ่ม เป็นต้น ซึ่งอาจมีการเก็บค่าบางค่าเอาไว้ที่ สเตท (State) เพื่อนำมาแสดงผล

การเขียน React จำเป็นต้องมีความรู้ใน 3 ประเด็น ได้แก่

- Component – ส่วนประกอบต่างๆ ในเว็บ จะถูกมองเป็น Component
- State – ข้อมูลที่อยู่ใน Component แต่ละชิ้น เรียกว่า State
- Props – ข้อมูลที่ถูกส่งต่อจาก Component ชั้นบนลงไปชั้นล่าง ซึ่งเรียกว่า Props (Properties)

อธิบายประเด็นหลักทั้ง 3 ประเด็น ดังรูปที่ 2.1



รูปที่ 2.1: concept หลักของ React

ที่มา: https://www.designil.com/wp-content/uploads/2017/07/react-concept-designil.jpg

2.1.1 React Life Cycle

การเขียน render() ฟังก์ชันใน component [?] นั้น ควรจะเขียนในแบบ pure function ซึ่งจะไม่มีการเปลี่ยน state และสร้าง side effect ต่อภายนอกทั้งสิ้น อย่างเช่น การ call extenal service แบบ Ajax request, Firebase calling เป็นต้น เพราะหน้าที่ของ render() มีแค่การ render UI เท่านั้น หากไม่สามารถทำสิ่งดังกล่าวภายใน render() แล้ว กิจกรรมเหล่านั้นจึงสามารถทำได้ที่ life cycle ของ React

ตลอดช่วงวงจรชีวิต [?] สามารถควบคุมเหตุการณ์ต่างๆ ที่เกิดขึ้นในการแสดงผล UI การอัพเดทข้อมูล และการ re-rendering จนกระทั้งข้อมูลนั้นหายไป โดยที่ React ได้มีการเตรียมฟังก์ชันต่างๆ ไว้ สามารถอธิบายการทำงานของฟังก์ชันได้ ดังนี้

- componentWillMount() : คุณสมบัติของ componentWillMount ไม่มีอะไรเกี่ยวกับ

การใช้งาน component เพราะยังไม่มีการ mount อะไรขึ้นมา โดยมีหน้าที่ คือ การกำหนดค่าเริ่มต้นสำหรับการใช้งาน

- componentDidMount() : เกิดขึ้นเมื่อทำการ Mount เรียบร้อย พร้อมที่จะใช้งาน โดยปกติจะใช้ในการกำหนดค่าทุกอย่างที่ต้องใช้ DOM และรับข้อมูลที่ต้องการมาแสดงผล
- componentWillReceiveProps(nextProps) : เมื่อ Component ทำงาน จนกระทั้งมี pro-ps ใหม่เข้ามา เพื่อทำการเปลี่ยนแปลงข้อมูล componentWillReceive Props จะถูกเรียก โดยมี nextProps เป็นตัวแปรที่ถูกส่งเข้ามา
- shouldComponentUpdate(nextProps, nextState) : ถูกเรียกเมื่อ component มีการเปลี่ยนแปลงด้วย nextProps กับ nextState
- componentWillUpdate(nextProps, nextState) : ถูกเรียกก่อนที่จะ render หลังจากได้รับค่าใหม่ของ props หรือ state คุณสมบัติของคล้ายกับ componentWillReveiveProps
- componentDidUpdate(prevProps, prevState) : ถูกเรียกทันที่หลังจากเกิดการเปลี่ยน-แปลงของ component แต่จะไม่ถูกเรียกตอนครั้งแรกที่ render โดยที่ componentDidUpd-ate สามารถใช้งานได้เหมือน componentDidMount
- componentWillUnmount() : ถูกเรียกก่อนที่ component ทำการ unmount และ destroy โดยปกติแล้วจะใช้เพื่อทำการรีเซ็ต (reset) ค่าต่างๆ

2.1.2 React Life Cycle

การเขียน render() ฟังก์ชันใน component [?] นั้น ควรจะเขียนในแบบ pure function ซึ่งจะไม่มีการเปลี่ยน state และสร้าง side effect ต่อภายนอกทั้งสิ้น อย่างเช่น การ call extenal service แบบ Ajax request, Firebase calling เป็นต้น เพราะหน้าที่ของ render() มีแค่การ render UI เท่านั้น หากไม่สามารถทำสิ่งดังกล่าวภายใน render() แล้ว กิจกรรมเหล่านั้นจึงสา-มารถทำได้ที่ life cycle ของ React

ตลอดช่วงวงจรชีวิต [?] สามารถควบคุมเหตุการณ์ต่างๆ ที่เกิดขึ้นในการแสดงผล UI การอัพเดทข้อมูล และการ re-rendering จนกระทั้งข้อมูลนั้นหายไป โดยที่ React ได้มีการเตรียมฟังก์ชันต่างๆ ไว้ สามารถอธิบายการทำงานของฟังก์ชันได้ ดังนี้

- componentWillMount() : คุณสมบัติของ componentWillMount ไม่มีอะไรเกี่ยวกับการใช้งาน component เพราะยังไม่มีการ mount อะไรขึ้นมา โดยมีหน้าที่ คือ การกำหนด

ค่าเริ่มต้นสำหรับการใช้งาน

- componentDidMount() : เกิดขึ้นเมื่อทำการ Mount เรียบร้อย พร้อมที่จะใช้งาน โดยปกติจะใช้ในการกำหนดค่าทุกอย่างที่ต้องใช้ DOM และรับข้อมูลที่ต้องการมาแสดงผล
- componentWillReceiveProps(nextProps) : เมื่อ Component ทำงาน จนกระทั้งมี pro-ps ใหม่เข้ามา เพื่อทำการเปลี่ยนแปลงข้อมูล componentWillReceive Props จะถูกเรียก โดยมี nextProps เป็นตัวแปรที่ถูกส่งเข้ามา
- shouldComponentUpdate(nextProps, nextState) : ถูกเรียกเมื่อ component มีการเปลี่ยนแปลงด้วย nextProps กับ nextState
- componentWillUpdate(nextProps, nextState) : ถูกเรียกก่อนที่จะ render หลังจากได้รับค่าใหม่ของ props หรือ state คุณสมบัติของคล้ายกับ componentWillReveiveProps
- componentDidUpdate(prevProps, prevState) : ถูกเรียกทันที่หลังจากเกิดการเปลี่ยน-แปลงของ component แต่จะไม่ถูกเรียกตอนครั้งแรกที่ render โดยที่ componentDidUpd-ate สามารถใช้งานได้เหมือน componentDidMount
- componentWillUnmount() : ถูกเรียกก่อนที่ component ทำการ unmount และ destroy โดยปกติแล้วจะใช้เพื่อทำการรีเซ็ต (reset) ค่าต่างๆ

2.2 ความรู้พื้นฐานเกี่ยวกับ Node.js

Node.js [?] เป็นภาษาที่ทำงานอยู่ในฝั่งเซิร์ฟเวอร์ (server) ซึ่ง syntax ที่ใช้ในการเขียนคือ JavaScript และเป็นภาษาที่ออกแบบมาให้ทำงานแบบ Event-Driven หรือทำงานเมื่อเกิดเหตุการณ์ตามที่กำหนดไว้ และการทำงานแบบ Asynchronous ซึ่งสามารถทำงานในลำดับต่อไปโดยที่ไม่ต้องรอให้งานก่อนหน้าเสร็จก่อนแล้วจึงทำงานขั้นต่อไป แต่ก็สามารถกำหนดให้ทำงานแบบ Synchronous ได้เช่นกัน โดยการกำหนด Callback เมื่องานแรกทำงานเสร็จแล้ว นอกจากนี้ Node.js นั้นจะใช้ Compiler จาก Google JavaScript Engine V8

ส่วนใหญ่จะนิยมใช้ node.js ในงานที่ทำเป็นเบื้องหลัง คือ งานที่ประมวลผลสั่งเซิร์ฟเวอร์ ซึ่งเป็นงานที่อาจจะต้อง interface กับผู้ใช้ หรือไม่ต้อง interface กับผู้ใช้ ตัวอย่างงานที่ต้อง interface กับผู้ใช้ เช่น การทำตัวเองเป็น http server ในการดึงหน้าเว็บมาแสดงผลให้กับ user หรือว่า การเปิด socket เพื่อรับส่งข้อมูลกันระหว่างเซิร์ฟเวอร์กับผู้ใช้งาน เช่น ทำเป็นห้อง chat ทำเกม ทำระบบที่ป้อนข้อมูลเพื่อคำนวณผลลัพธ์ เป็นต้น ตัวอย่างงานที่ไม่ต้อง interface กับผู้ใช้

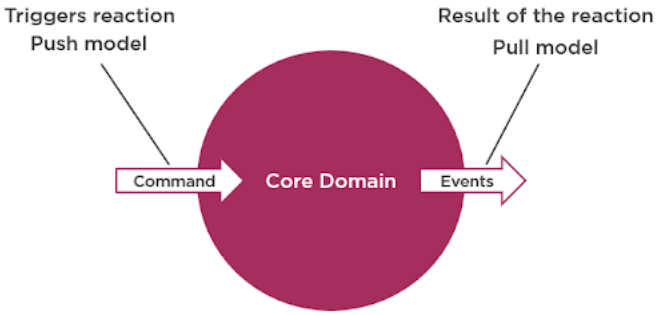
เช่น ทำ spider crawler เว็บ คือ การเปิดเว็บแล้วเก็บข้อมูลไปเรื่อยๆ หรือ โปรแกรมที่ รอรับค่าจาก streaming ต่างๆ เพื่อนำมาบันทึกไว้ ซึ่งการทำงานเหล่านี้ไม่จำเป็นต้อง interface กับผู้ใช้

node.js มีส่วนเสริมที่ชื่อว่า node package management (npm) ซึ่งเปรียบเหมือน google play ใน android หรือ app store ใน iOS ที่สามารถเอา package ที่คนอื่นเขียนเอาไว้แล้ว เพื่อแจกฟรี (free) มาต่อยอดเพื่อใช้ในงานของตนได้ โดยตัวอย่างที่ได้รับความนิยมจะเป็น underscore, async, request และ express เป็นต้น สำหรับการติดตั้ง ใช้คำสั่ง npm install ตามด้วยชื่อ package ที่ต้องการติดตั้ง [?]

node.js มีการทำงานเป็น Asynchronous คือ การทำงานบางอย่างไม่ต้องรอให้บรรทัดนั้นทำงานเสร็จ เช่น ส่งคำสั่งไป query ข้อมูลจากฐานข้อมูล แล้วสามารถข้ามไปทำงานบรรทัดต่อไปโดยไม่ต้องรอผลจากฐานข้อมูล เมื่อการทำงานนั้นทำงานเสร็จจึงค่อยรอผลลัพธ์กลับมา ดังนั้นปัญหาจะเกิดทันที ถ้าการทำงานต่อไปนำผลลัพธ์จากคำสั่งก่อนหน้านั้นมาใช้ต่อ ซึ่งส่งผลให้เกิดการทำงานผิดพลาด เพราะผลลัพธ์ยังไม่ได้รับกลับมา

2.2.1 node.js ทำงานแบบ event driven

การทำงานของ node [?] เรียกว่าเป็นการขับเคลื่อนด้วย event ต่างๆ ที่เกิดขึ้น ทำให้สามารถข้ามจาก event หนึ่งที่เสร็จแล้วไปยัง event อื่นได้ด้วยการสั่งงานต่อเนื่องกันไป หรือการสั่งให้ event หลาย event เริ่มทำงานในเวลาใกล้เคียงกัน ประโยชน์ที่ได้จาก event driven คือ การสั่งให้รอรับ event นั้นไปตลอดการณ์ โดยไม่เปลืองทรัพยากร เช่น การเชื่อมต่อไปยัง streaming channel ที่หนึ่ง ซึ่งอาจเป็น text หรือข้อมูลบางอย่าง เช่น ปริมาณน้ำฝน เอาไว้ หากต้นทางของ streaming ยังไม่มีข้อมูลส่งมา จะไม่เกิด event ใดๆ และ node.js จะรออยู่ แต่หากต้นทาง streaming มีข้อมูลมา node.js จะทำงานเพื่อตอบสนองต่อ event ที่เกิดขึ้นนั้นทันที สามารถแสดงการทำงานดังกล่าวได้ ดังรูปที่ 2.2



รูปที่ 2.2: การทำงานแบบ event driven

ที่มา: http://meewebfree.com/u/i/nodejs/node_js_stage.png

จากรูปที่ 2.2 สามารถอธิบายการทำงานได้ดังนี้ การทำงานของ stage1 เมื่อเรียกใช้การทำงานของ stage2 แล้ว ไม่จำเป็นต้องรอให้ stage2 ทำงานเสร็จก่อน ซึ่ง stage1 สามารถเรียกการทำงานของ stage3 ได้เลยโดยไม่จำเป็นต้องรอการทำงานของ stage2

2.2.2 ข้อดีของ Node.js

- มีการทำงานแบบ Event-Driven และ Asynchronous
- เหมาะกับการทำ Web แบบ Real time
- ประหยัดทรัพยากร ในการทำงาน
- มีการประมวลผลที่รวดเร็ว

2.3 ความรู้พื้นฐานเกี่ยวกับ JavaScript

JavaScript [?] คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต (Internet) ที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็น ภาษาสคริปต์ (script) เชิงวัตถุ ที่เรียกกันว่า ”สคริปต์” ซึ่งการใช้ JavaScript ในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) จะช่วยให้เว็บไซต์ดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น โดยมีวิธีการทำงานในลักษณะ ”แปลความและดำเนินงานไปทีละคำสั่ง” (interpret) หรือเรียกว่า โปรแกรมเชิงวัตถุ (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (client) และ ทางฝั่งเซิร์ฟเวอร์

JavaScript ถูกพัฒนาขึ้นโดย บริษัท เน็ตสเคปคอมมิวนิเคชันส์ (Netscape Communications Corporation) โดยใช้ชื่อว่า Live Script ออกมาพร้อมกับ Netscape Navigator 2.0 เพื่อใช้สร้างเว็บเพจ (Web page) โดยติดต่อกับเซิร์ฟเวอร์แบบ Live Wire ต่อมาเน็ตสเคปได้ร่วมมือกับ บริษัทซันไมโครซิสเต็มส์ (Sun Microsystems, Inc) ปรับปรุงระบบของเบราว์เซอร์ (Browser) เพื่อให้สามารถติดต่อใช้งานกับภาษาจาวา (Java) ได้ และได้ปรับปรุง LiveScript ใหม่เมื่อ ปี พ.ศ. 2538 แล้วตั้งชื่อใหม่ว่า JavaScript ซึ่ง JavaScript ทำให้การสร้างเว็บเพจมีลูกเล่นต่างๆ มากมาย และยังสามารถโต้ตอบกับผู้ใช้ได้อย่างทันที เช่น การใช้เมาส์คลิก หรือ การกรอกข้อความในฟอร์ม เป็นต้น

เนื่องจาก JavaScript ช่วยให้ผู้พัฒนา สามารถสร้างเว็บเพจได้ตรงกับความต้องการ และมีความน่าสนใจมากขึ้น ประกอบกับเป็นภาษาเปิด ที่ทุกคนสามารถนำไปใช้ได้ ดังนั้นจึงได้รับความนิยมเป็นอย่างสูง มีการใช้งานอย่างกว้างขวาง รวมทั้งได้ถูกกำหนดให้เป็นมาตรฐานโดย European Computer Manufacturer's Association (ECMA) การทำงานของ JavaScript จะต้องมีการแปลความคำสั่ง ซึ่งขั้นตอนนี้จะถูกจัดการโดยเบราว์เซอร์ (เรียกว่าเป็น client-side script) ดังนั้น JavaScript จึงสามารถทำงานได้เฉพาะบนเบราว์เซอร์ที่สนับสนุน ซึ่งปัจจุบันเบราว์เซอร์เกือบทั้งหมดสามารถสนับสนุน JavaScript แล้ว อย่างไรก็ดี สิ่งที่ต้องระวังคือ JavaScript มีการพัฒนาเป็นเวอร์ชันใหม่ๆ ออกมาด้วย ดังนั้น ถ้านำโค้ด (code) ของเวอร์ชันใหม่ ไปรันบนเบราว์เซอร์รุ่นเก่าที่ยังไม่สนับสนุน อาจจะทำให้เกิด error ได้

2.3.1 ประโยชน์ของ JavaScript

- JavaScript ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายได้ โดยไม่ต้องพึ่งภาษาอื่น
- JavaScript มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่น เมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox สามารถสั่งให้เปิดหน้าใหม่ได้ ทำให้เว็บไซต์มีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น
- JavaScript สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้โดยง่าย
- JavaScript สามารถใช้ตรวจสอบข้อมูลได้ เช่น เมื่อผู้ใช้งานกรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อบันทึกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่ากรอกผิด หรือลืมบันทึกอะไรบางอย่าง เป็นต้น

- JavaScript สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร
- JavaScript สร้าง Cookies (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

2.3.2 ข้อดีและข้อเสียของ Java JavaScript

JavaScript [?] ทำงานบนเว็บบราวเซอร์ (client-side script) จึงไม่มีข้อจำกัดว่าจะใช้เซิร์ฟเวอร์แบบไหนก็ตาม เพราะ JavaScript ทำงานเฉพาะในเครื่องของผู้ใช้งานเท่านั้น ซึ่งต่างกับภาษาสคริปต์อื่น เช่น PHP , ASP, JSP หรือ Perl ซึ่งต้องประมวลผลและทำงานที่เครื่องเซิร์ฟเวอร์ (server-side script) จึงจำเป็นต้องใช้บนเซิร์ฟเวอร์ ที่สนับสนุนภาษาเหล่านี้เท่านั้นจึงจะสามารถใช้งาน server-side script ได้ แต่อย่างไรก็ตาม จากลักษณะการทำงานที่กล่าวมาก็ทำให้ JavaScript มีข้อจำกัด กล่าวคือคือไม่สามารถรับและส่งข้อมูลต่างๆ กับเซิร์ฟเวอร์โดยตรง เช่น การอ่านไฟล์จากเซิร์ฟเวอร์ เพื่อนำมาแสดงบนเว็บเพจ หรือรับข้อมูลจากผู้ชม เพื่อนำไปเก็บบนเซิร์ฟเวอร์ เป็นต้น ดังนั้นงานลักษณะนี้ จึงยังคงต้องอาศัยภาษา server-side script อยู่ (ความจริงมี JavaScript ที่ทำงานบนเซิร์ฟเวอร์เช่นกัน ซึ่งต้องอาศัยเซิร์ฟเวอร์ที่สนับสนุนโดยเฉพาะเช่นกัน แต่ไม่เป็นที่นิยมนัก)

นักพัฒนาเว็บส่วนใหญ่จึงนิยมใช้ JavaScript ร่วมกับ ภาษา Server Script เพื่อทำการส่งข้อมูลระหว่าง เซิร์ฟเวอร์กับเครื่องของผู้ใช้งาน ซึ่งทำให้การแสดงผลของหน้าเว็บมีความสวยงามและราบรื่นมากยิ่งขึ้น

2.4 ความรู้พื้นฐานเกี่ยวกับ MySQL

MySQL [?] เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System : RDBMS) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเตอร์เน็ต สาเหตุเพราะว่า MySQL เป็นฟรีแวร์ ทางด้านของฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบฐานขอมูลที่เคยใช้ MySQL ต่างยอมรับในความสามารถความรวดเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมหาศาล ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย ไม่ว่าจะเป็น Unix , OS/2 , Mac , OS หรือ Windows ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้ งานร่วมกับ Web Development Platform ทั้งหลาย ไม่ว่าจะเป็น C,

C++, Java, Perl, PHP, Python, Tcl หรือ ASP ดังนั้น MySQL จึงได้รับความนิยมอย่างมากในปัจจุบัน และมีแนวโน้มสูงยิ่งขึ้นต่อไปในอนาคต

2.4.1 โครงสร้างของ MySQL

โครงสร้างภายในของ MySQL [?] คือ การออกแบบการทำงานในลักษณะของ Client Server นั่นเอง ซึ่งประกอบด้วย 2 ส่วน คือ ส่วนของผู้ให้บริการ (Server) และ ส่วนของผู้ใช้บริการ (Client) โดยในแต่ละส่วนจะมีโปรแกรมสำหรับการทำงานตามหน้าที่ของโปรแกรมนั้น ส่วนของผู้ให้บริการ (Server) จะเป็นส่วนที่ทำหน้าที่บริหารจัดการระบบฐานข้อมูล (MySQL Server) และเป็นที่จัดเก็บข้อมูลทั้งหมด ข้อมูลที่เก็บไว้นี้มีข้อมูลที่จำเป็นสำหรับการทำงานกับระบบฐานข้อมูล และข้อมูลที่เกิดจากการที่ผู้ใช้แต่ละคนสร้างขึ้นมา ส่วนของผู้ใช้บริการ (Client) คือ ส่วนที่ผู้ใช้ใช้งาน โดยโปรแกรมสำหรับใช้งานในส่วนนี้ได้แก่ MySQL , Client , Access . Web Development เป็นต้น

2.4.2 หลักการทำงานในลักษณะ Client Server

เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Client หรือ Server อาจจะอยู่บนเครื่องเดียวกันหรือแยกเครื่องกันก็ได้ทั้งนี้ขึ้นอยู่กับลักษณะการทำงาน หรือการกำหนดของผู้บริหารระบบตามปกติ ถ้าเป็นการทำงานลักษณะ Web-based มีการใช้ฐานข้อมูลขนาดไม่ใหญนัก ตัว MySQL Server และ Client มักจะมีอยู่บนเครื่องเดียวกัน โดยเครื่องคอมพิวเตอร์ดังกล่าวจะต้องมีทรัพยากรเพื่อการทำงาน เช่น เนื้อที่ฮาร์ดดิสก์ RAM มากพอสมควร แต่สำหรับการทำงานจริง (Real-world Application) มักจะแยก Client และ Server ออกเป็นคนละเครื่องกัน และสามารถรองรับงานได้ดีมากกว่า ดังนั้น ผู้บริหารระบบหรือผู้กำหนดนโยบายสำหรับการทำงานเครือข่ายจะต้อง คำนึงถึงเรื่องที่เกี่ยวข้องเหล่านี้ให้ดี เพื่อที่จะทำให้ระบบมีการทำงานรับการให้บริการแก่ผู้ใช้อย่างมีประสิทธิภาพ และข้อมูลมีความปลอดภัยมากที่สุด หลักการทำงานในลักษณะ Client Server มีดังนี้

- ที่ฝั่งของ Server จะมีโปรแกรมหรือระบบสำหรับจัดการฐานข้อมูลทำงานรออยู่ เพื่อเตรียมหรือรอ คอยการร้องขอการใช้บริการจาก Client
- เมื่อมีการรองขอการใช้บริการเข้ามา Server จะทำการตรวจสอบ เช่น อาจจะมีการให้ผู้ใช้บริการระบุชื่อและรหัสผ่าน และสำหรับ MySQL สามารถกำหนดได้ว่าจะอนุญาตหรือ

ปฏิเสธ Client ใดๆ ในระบบที่จะเข้าใช้บริการอีกด้วย

- ถ้าผ่านการตรวจสอบ Server ก็จะอนุมัติการให้บริการแก่ Client ที่รองขอการใช้ บริการนั้นๆ ต่อไปและถ้าในกรณีที่ไม่ได้ รับการอนุมัติServer ก็จะส่งข่าวสารความผิดพลาดแจ้งกลับไปที่ Client ที่รองขอการใช้ บริการนั้น

2.4.3 การทำงานของโปรแกรมของ MySQL

MySQL ถือเป็นระบบจัดการฐานข้อมูล (Database Management System : DBMS) ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเพิ่มเติม เข้าถึงหรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการ ฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการ ใช้งานเฉพาะ และรองรับการทำงานของแอพพลิเคชั่น (Application) ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูลและระบบจัดการฐานข้อมูล ดังนี้

- MySQL เป็นระบบจัดการฐานข้อมูลแบบ relational ฐานข้อมูลแบบ relational จะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์ เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น นอกจากนั้น แต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้สามารถรวมหรือจัด กลุ่มข้อมูลได้ตามต้องการ โดยอาศัยภาษา SQL ที่เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล
- MySQL แจกจ่ายให้ใช้งานแบบ Open Source นั่นคือ ผู้ใช้งาน MySQL ทุกคนสามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ สามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่าย

2.4.4 คุณสมบัติของ MySQL

1. สนับสนุน Cross-platform support
2. รองรับ Stored procedures
3. รองรับ Triggers และ Cursors
4. สนับสนุน Information schema
5. สนับสนุน SSL
6. รองรับการทำ Query caching

7. รองรับการทำ Sub-SELECTs
8. รอบรับการทำ Replication ทั้งแบบ Master-Master Replication และ Master-Slave Replication
9. รองรับ Unicode

2.5 ความรู้เกี่ยวกับ Visual Studio Code

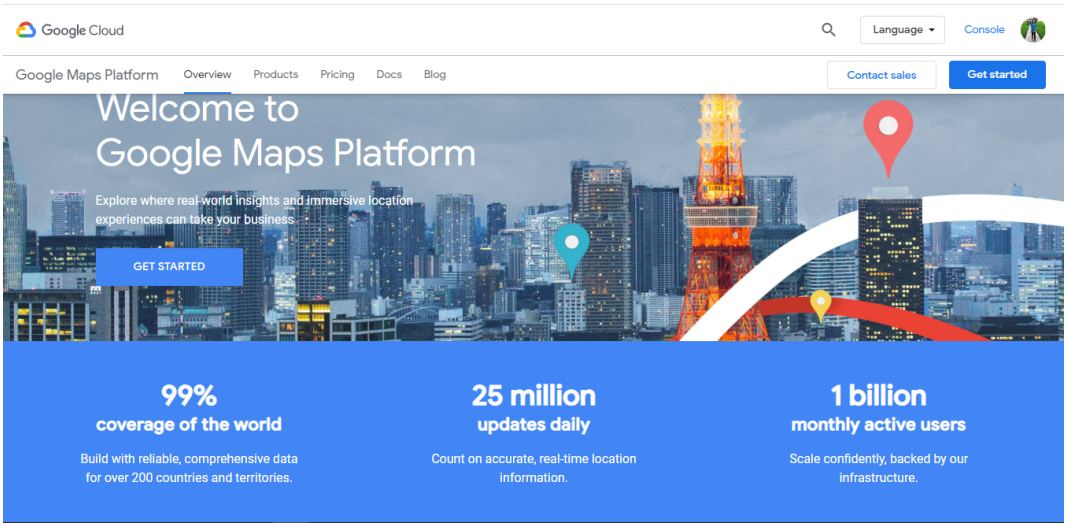
วิชวล สตูดิโอโค้ด (Visual Studio Code หรือ VSCode) [?] เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ดถูกพัฒนาโดยค่ายไมโครซอฟท์(Microsoft) มีการพัฒนาออกมาในรูปแบบของ OpenSource จึงสามารถนำมาใช้งานได้แบบไม่ต้องเสียค่าใช้จ่าย Visual Studio Code เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม (Cross-platform) โดยจะรองรับการใช้งานทั้งบนระบบปฏิบัติการ Windows,macOS และ Linux ซึ่งภาษาที่ Visual Studio Code รองรับการทำงานซึ่งมีมากกว่า 30 ภาษาโปรแกรม เช่น C++, C, CSS, Dockerfile, HTML, JavaScript, JSON, Less, Markdown, PHP, Python, Sass, TypeScript, Node.js และ Java เป็นต้น

2.6 ความรู้พื้นฐานเกี่ยวกับ Google Maps API

Google Maps API [?] เป็นชุด API ของ Google สำหรับพัฒนา web application และ mobile application (Android,ios) มีไว้สำหรับเรียกใช้แผนที่และชุด service ต่างๆ มากมายให้เรียกใช้ เช่น

- ชุดควบคุมแผนที่ (Map Control)
- การปรับแต่งแผนที่ (Styled Map)
- การนำทางจากจุดหนึ่งไปยังอีกจุดหนึ่ง (Directions Service)
- Street view
- การดึงข้อมูล POI (Point of Interest) คือข้อมูลสถานที่ต่างๆ ที่ Google รวบรวมไว้ เช่น โรงแรม ห้างสรรพสินค้า โรงเรียน สถานที่ราชการ เป็นต้น เพื่อให้ผู้ใช้งานนำมาใช้งานได้
- การแปลงที่อยู่เป็นพิกัด Latitude และ Longitude (Geocoding Service)
- แผนที่รองรับระบบ 3 มิติ ผู้ใช้งาน สามารถปรับมุมมองเป็นลักษณะ 3 มิติได้ ซึ่งระบบนี้มีให้บริการบางพื้นที่

การใช้งาน Google Maps API สามารถขอใช้งานโดยใช้บัญชี Gmail และเข้าไปที่เว็บ https://cloud.google.com/maps-platform/ ดังรูปที่ 2.3



รูปที่ 2.3: หน้าเว็บบริการของ Google Maps

ที่มา : https://cloud.google.com/maps-platform/

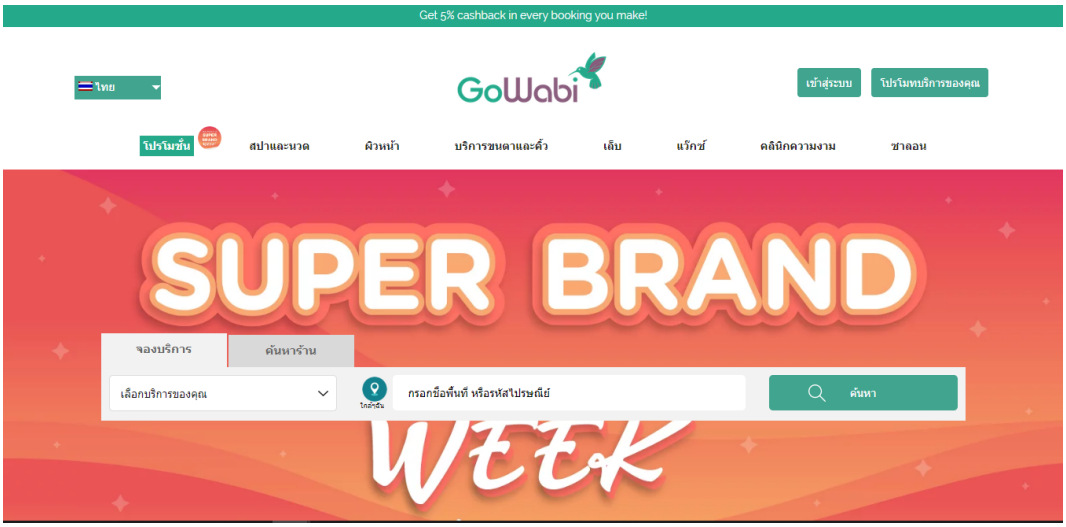
2.7 เอกสารและงานวิจัยที่เกี่ยวข้อง

2.7.1 เว็บแอพพลิเคชัน Gowabi

Gowabi[?] เป็นเว็บไซต์และแอปพลิเคชันที่ให้บริการเกี่ยวกับการค้นหาและจองคิวร้านเสริมสวย มีฟังก์ชันการทำงานพื้นฐานอันได้แก่ การค้นหา การจองคิว ดูข้อมูล เป็นต้น

2.7.2 ข้อแตกต่างระหว่างเว็บแอปพลิเคชั่น Gowabi กับเว็บของโครงงาน

เว็บแอปพลิเคชั่นของ Gowabi ยังไม่มีฟังก์ชันเลือกรายการ และช่างในการจองคิว ผู้พัฒนาจึงได้ทำฟังก์ชันเลือกรายหารและะช่างที่เหมาะสมตามรสนิยมของผู้ใช้เพิ่มลงในเว็บของโครงงาน



รูปที่ 2.4: หน้าแรกของเว็บไซต์ Gowabi

ที่มา :https://www.gowabi.com

บทที่ 3

การวิเคราะห์และออกแบบระบบ

การวิเคราะห์และออกแบบระบบก่อนดำเนินการจริงเป็นอีกหนึ่งขั้นตอนที่มีความสำคัญมาก เพราะการวิเคราะห์และออกแบบระบบนั้นเป็นการกระทำที่ทำให้ผู้พัฒนาเห็นรายละเอียดส่วนย่อยของงานทั้งหมด เพิ่มประสิทธิภาพในการวางแผน การทำงาน และยังช่วยลดปัญหาที่อาจจะเกิดขึ้นในระหว่างพัฒนา เพื่อให้ระบบมีความสมบูรณ์มากยิ่งขึ้น เนื่องจากการวิเคราะห์และออกแบบระบบนั้นจะช่วยให้ให้บริการ จัดการทรัพยากรได้อย่างคุ้มค่าและตรงตามความต้องการของระบบ

การวิเคราะห์และออกแบบระบบ การจองคิวร้านเสริมสวย ในบทนี้จะแบ่งออกเป็น 6 ขั้นตอนเพื่อให้เห็นการดำเนินงานอย่างมีระบบ ในหัวข้อแรกจะนำเสนอภาพรวมของระบบ ก่อนจะนำเสนอเอกสารแสดงความต้องการของระบบซึ่งจะทำให้เห็นที่มาของเพจต่าง ๆ ในขั้นตอนของการออกแบบในหัวข้อที่สาม ส่วนหัวข้อที่เหลือจะแสดงแผนภาพการการทำงานของระบบโดยใช้ 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

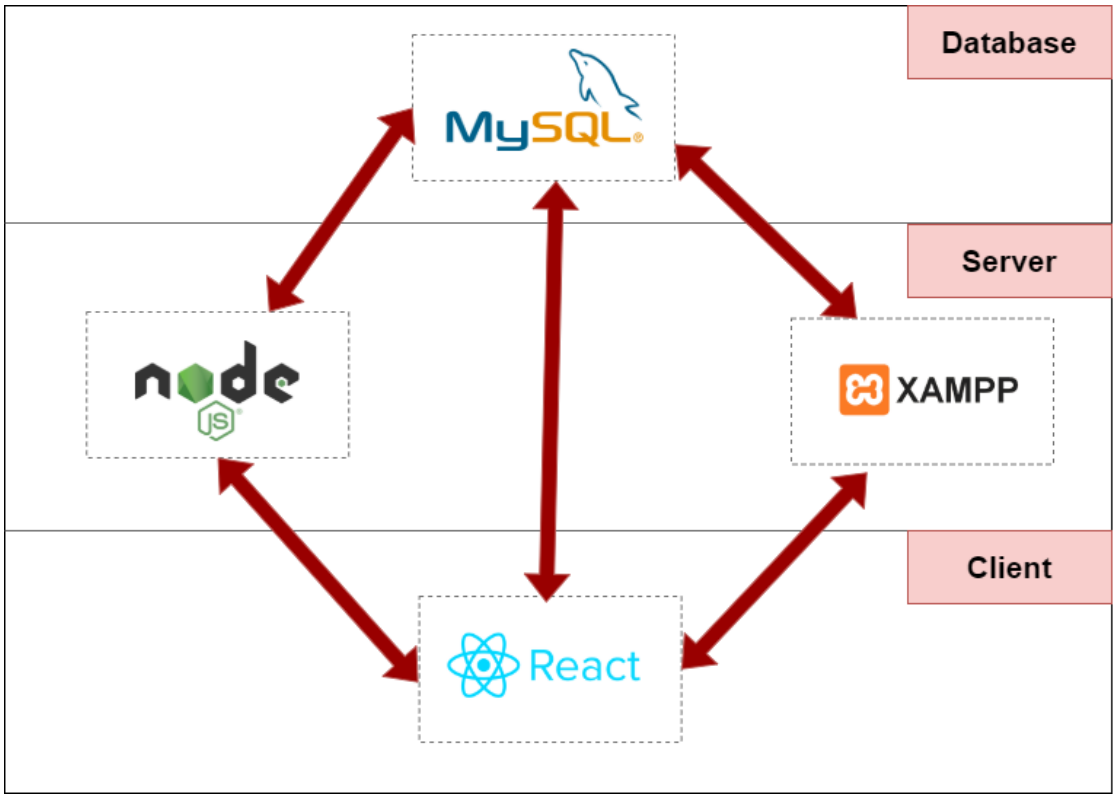
3.5 Class Diagram เป็นแผนภาพที่ใช้แสดง Class และความสัมพันธ์ระหว่าง Class

3.6 Sequence Diagram เป็นแผนภาพที่ใช้แสดงให้เห็นถึงการตอบโต้ข้อมูลระหว่างคลาส เรียงตามลำดับของเวลาที่เกิดเหตุการณ์จากน้อยไปมาก

3.1 โครงสร้างภาพรวมของระบบ

ความหมายของ System Architecture [11] หมายถึง กรอบโครงสร้างของระบบที่อธิบายความสัมพันธ์ขององค์ประกอบต่าง ๆ ไปจนถึงขั้นการเชื่อมต่อกันของระบบย่อยต่าง ๆ 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 ระบบการจองคิวร้านเสริมสวย

จากรูปที่ 3.1 สามารถอธิบายโครงสร้างและเทคโนโลยีของระบบโดยแบ่งเป็น 3 ส่วนหลักดังนี้

1. Database ระบบใช้บริการฐานข้อมูลแบบ MySQL ของไฟร์เบสชื่อ Relational Database Management System : RDBMS

2. Server กระบวนการทำงานในส่วนของเซิฟเวอร์ (server) แบ่งเป็น 2 ส่วนได้แก่
    - ชุดเผยแพร่สำหรับการพัฒนาเว็บไซต์ซึ่งในที่นี้ใช้ Node.jsและExpress ในการพัฒนา
    - ชุดบริการ phpmyadmin Api ใช้สำหรับการทำงานกับบริการต่าง ๆ ของ MySLQ
3. Client เว็บแอปพลิเคชันการจองคิวร้านเสริมสวย มีวัตถุประสงค์เพื่อรองรับการทำงานของผู้ใช้บนเว็บบราวเซอร์พัฒนาด้วย React

3.2 System Requirements

3.2.1 Functional Requirements

ระบบการจองคิวร้านเสริมสวย แบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

1. เจ้าของร้าน
    - สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
    - สามารถดูคิวที่ผู้ใช้บริการได้ทำการจองคิวไว้
    - สามารถจัดการคิวได้
    - สามารถ post ภาพผลงานทั้งหมดของร้านได้
    - สามารถเพิ่ม แก้ไข และลบรายการให้บริการประจำร้านได้
    - สามารถเพิ่ม แก้ไข และลบ ข้อมูลร้านได้
    - สามารถเพิ่ม แก้ไข และลบข้อมูลตำแหน่งร้านได้
2. ช่างประจำร้าน
    - ลงทะเบียนใช้ web ด้วย Email ได้
    - สามารถดูตารางการทำงานของตนเองได้
    - สามารถ post ภาพผลงานของตัวเองได้
    - สามารถแก้ไขข้อมูลส่วนตัวได้
3. ผู้ใช้บริการ
    - สามารถลงทะเบียนเข้าสู่ระบบด้วย Email ได้
    - สามารถค้นหาร้านเสริมสวยได้
    - สามารถจองคิวของร้านเสริมสวยได้
    - สามารถดูคิวว่างของร้านเสริมสวยได้

- สามารถดูข้อมูลต่างๆของร้านเสริมสวยได้
- สามารถดูตำแหน่งของทางร้านได้
- สามารถดูผลงานของร้านได้
- สามารถเขียนรีวิว ติชม ได้

3.2.2 Non-functional Requirements

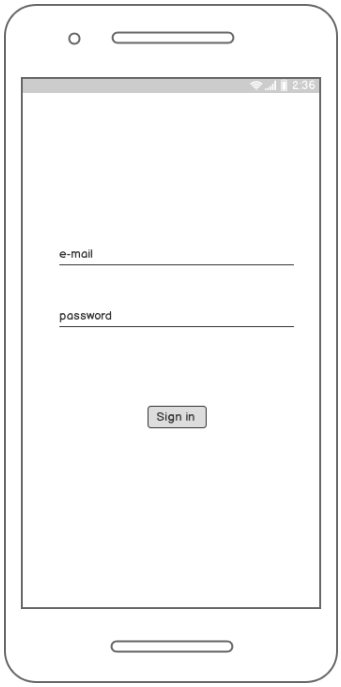
1. เว็บแอปพลิเคชัน
   - ใช้โปรโตคอล (Protocol) แบบ HTTPS (Hypertext Transfer Protocol Secure) ในการสื่อสารที่ช่วยรักษาความสมบูรณ์ถูกต้องของข้อมูลผู้ใช้และเก็บข้อมูลไว้เป็นความลับระหว่างคอมพิวเตอร์ของผู้ใช้กับเว็บไซต์
   - รองรับการใช้งานบนเว็บบราวน์เซอร์และสมาร์ทโฟน

3.3 User Interface Design

User Interface Design ของระบบการจองคิวร้านเสริมสวย มีดังนี้

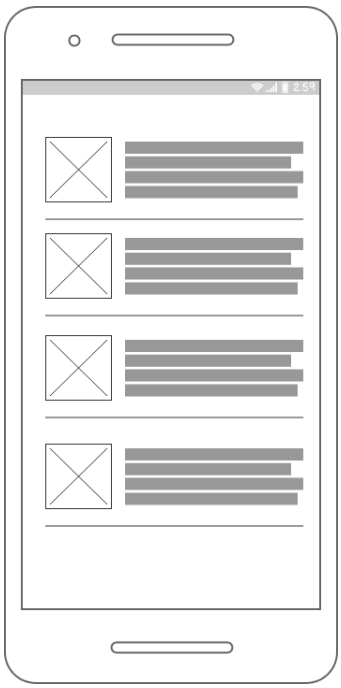
1. ผู้ใช้บริการ

   User Interface ผู้ใช้บริการ จะแบ่งออกแบบให้สะดวกต่อการใช้งาน



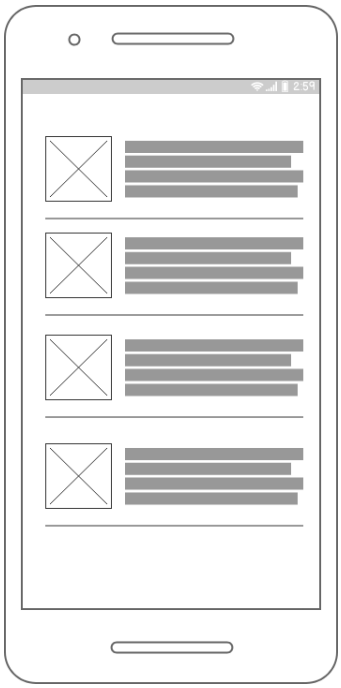
รูปที่ 3.2: หน้าจอเข้าสู่ระบบ

จากภาพที่ 3.2 แสดงหน้าจอสำหรับให้ผู้ใช้ทำการเข้าสู่ระบบ โดยจำเป็นต้องกรอกข้อมูลอีเมลและรหัสผ่านเพื่อใช้ในการเข้าสู่ระบบ



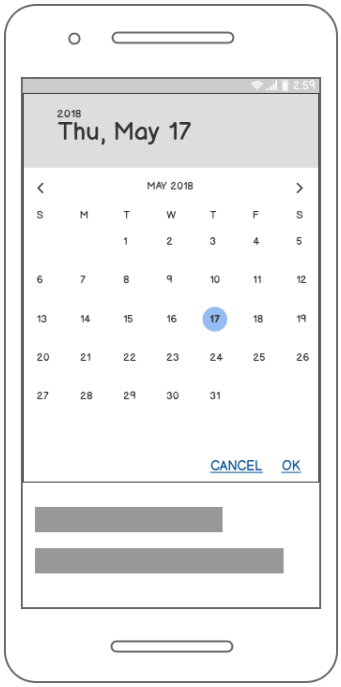
รูปที่ 3.3: หน้าจอข้อมูลร้าน

จากภาพที่ 3.3 แสดงหน้าจอข้อมูลร้านเพื่อใช้ตัดสินใจในกาเลือกรจองคิวร้านเสริมสวย



รูปที่ 3.4: หน้าจอข้อมูลแต่ละร้าน

จากภาพที่ 3.4 แสดงหน้าจอข้อมูลแต่ละร้าน เพื่อดูข้อมูลและการประชาสัมพันธ์ของร้าน



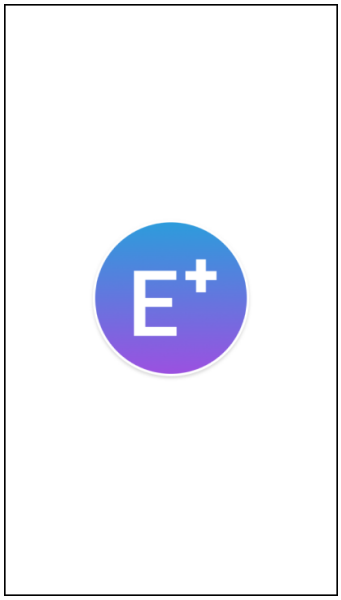
รูปที่ 3.5: หน้าจอการเลือกวันในการจองคิว

จากภาพที่ ?? แสดงหน้าจอปฏิทินกำหนดการการดำเนินการเพื่อให้ผู้ใช้สามารถตรวจสอบกำหนดการวันที่และเวลาในการดำเนินงานของกองทุนโดยหน้าจอมีการแสดงปฏิทินและแถวรายการกำหนดการ

2. High-fidelity wireframes

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

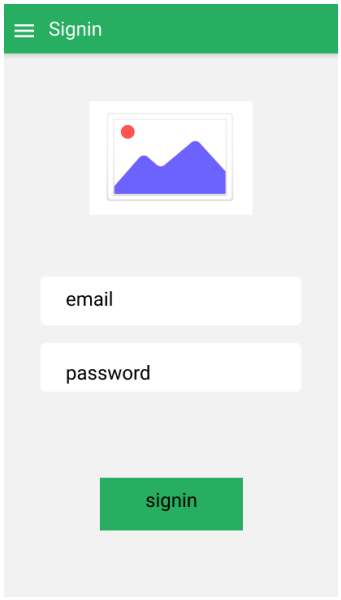
- โมบายแอปพลิเคชัน
  - การออกแบบหน้าจอ splash screen



รูปที่ 3.6: หน้าจอ splash screen

จากภาพที่ 3.6 แสดงหน้าจอ splash screen ใช้ในการแสดงทุกครั้งที่ผู้ใช้เปิดแอปพลิเคชันโดยวัตถุประสงค์การทำงานของหน้านี้คือเพื่อใช้แสดงขณะที่แอป-พลิเคชันทำการประมวลผลข้อมูลบนพื้นหลัง (Background process) เช่น การตรวจสอบสถานะการเข้าสู่ระบบของผู้ใช้คนปัจจุบัน เป็นต้น

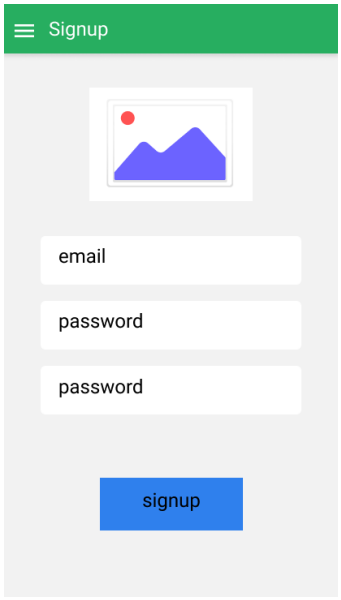
– การออกแบบหน้าจอเข้าสู่ระบบ



รูปที่ 3.7: หน้าจอเข้าสู่ระบบ

จากภาพที่ 3.7 แสดงหน้าจอสำหรับให้ผู้ใช้ทำการเข้าสู่ระบบเมื่อผู้ใช้ยังไม่ได้ทำการเข้าสู่ระบบ โดยจำเป็นต้องกรอกข้อมูลอีเมลและรหัสผ่านเพื่อใช้ในการเข้าสู่ระบบ ซึ่งการเข้าสู่ระบบจะทำเพียงครั้งเดียวเท่านั้น เมื่อผู้ใช้เปิดการทำงานแอปพลิชันใหม่ในครั้งถัดไประบบจะระบุข้อมูลของผู้ใช้งานอัตโนมัติ

- การออกแบบหน้าจอสมัครสมาชิก



รูปที่ 3.8: หน้าจอสมัครสมาชิก

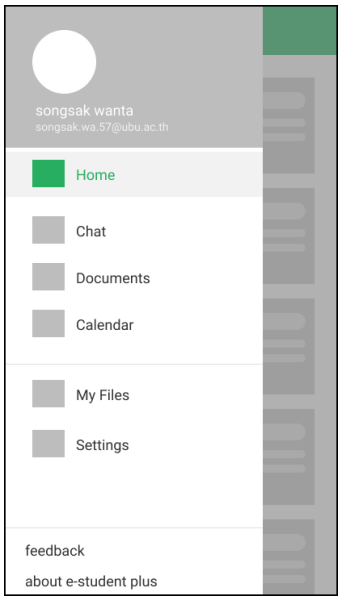
จากภาพที่ 3.8 แสดงหน้าจอสมัครสมาชิก หากผู้ใช้ยังไม่มีบัญชีในระบบผู้ใช้งานสามารถทำการสมัครสมาชิกเพื่อเข้าใช้งานระบบได้จากหน้าสมัครสมาชิก โดยผู้ใช้จำเป็นต้องกรอกข้อมูลอีเมล รหัสผ่านและรหัสนักศึกษาในการสมัครสมาชิก

– การออกแบบหน้าจอข่าวสารและประชาสัมพันธ์

รูปที่ 3.9: หน้าจอข่าวสารและประชาสัมพันธ์

จากภาพที่ 3.9 แสดงหน้าจอข่าวสารหรือประชาสัมพันธ์จากเจ้าหน้าที่หรือผู้ที่เกี่ยวข้อง

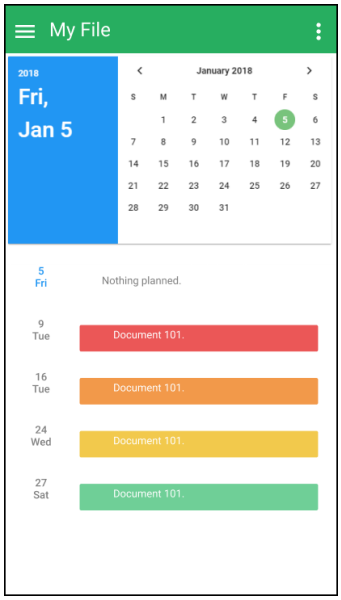
- การออกแบบหน้าจอเมนูนำทางหลัก



รูปที่ 3.10: หน้าจอเมนูนำทางหลัก

จากภาพที่ 3.10 แสดงเมนูนำทางหลักที่ใช้นำทางผู้ใช้งานไปยังหน้าจออื่นๆ ภายในแอปพลิเคชัน

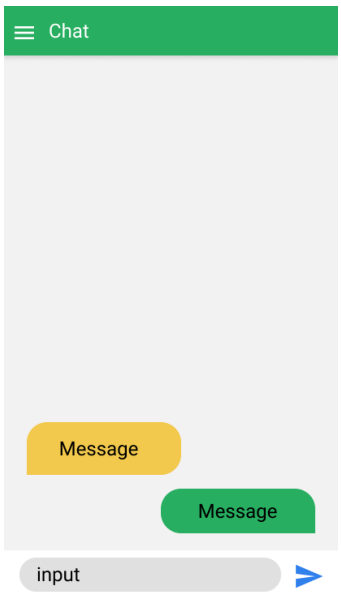
- การออกแบบหน้าจอปฏิทินกำหนดการการดำเนินการ



รูปที่ 3.11: หน้าจอปฏิทินกำหนดการการดำเนินการ

จากภาพที่ 3.11 แสดงหน้าจอปฏิทินกำหนดการการดำเนินการเพื่อให้ผู้ใช้สามารถตรวจสอบกำหนดการวันที่และเวลาในการดำเนินงานของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี

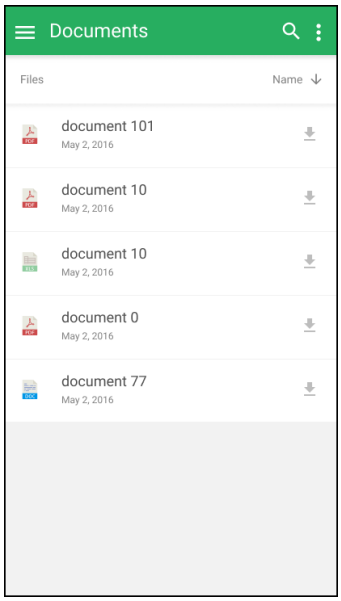
– การออกแบบหน้าจอสนทนา



รูปที่ 3.12: หน้าจอสนทนา

จากภาพที่ 3.12 นักศึกษาสามารถส่งข้อความไปยังเจ้าหน้าที่เพื่อติดต่อสอบถามข้อมูลกับทางเจ้าหน้าที่ได้โดยตรง

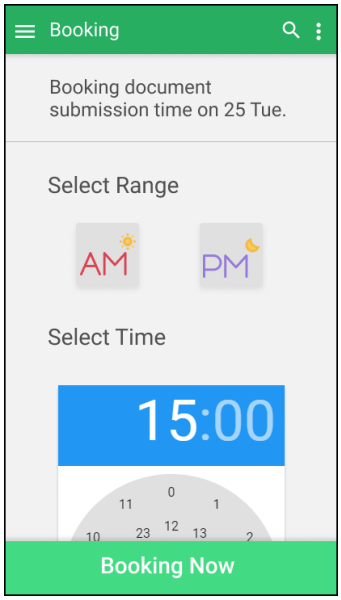
– การออกแบบหน้าจอเอกสารที่เกี่ยวข้อง



รูปที่ 3.13: หน้าจอเอกสารที่เกี่ยวข้อง

จากภาพที่ 3.13 นักศึกษาสามารถดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี เช่น ข้อกำหนดและคุณสมบัติของผู้กู้ยืม เป็นต้น ซึ่งเอกสารได้ถูกอัพโหลดไว้โดยเจ้าหน้าที่

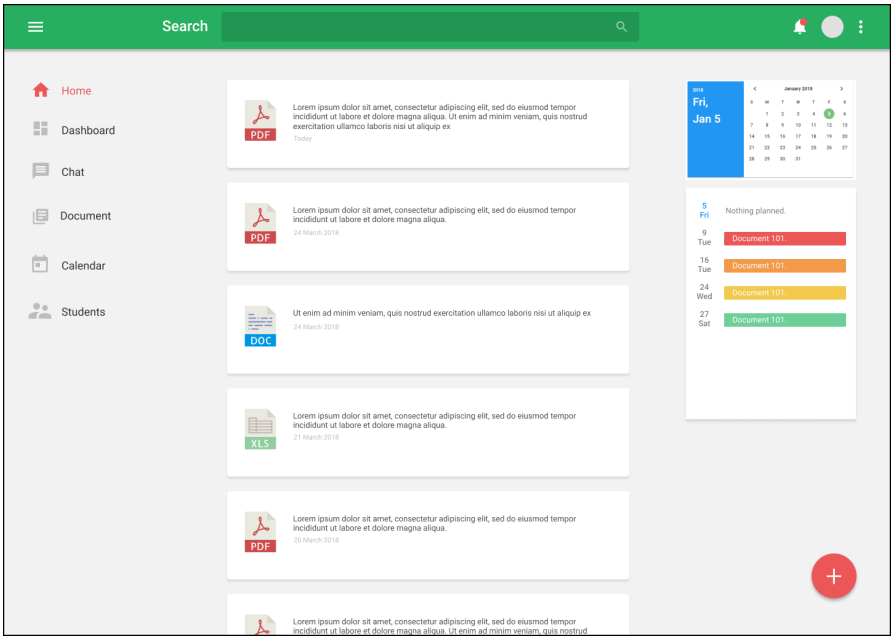
– การออกแบบหน้าจอจองวันที่และเวลาส่งเอกสาร



รูปที่ 3.14: หน้าจอจองวันที่และเวลาส่งเอกสาร

จากภาพที่ 3.14 เมื่อเจ้าหน้าที่ทำการเพิ่มวันที่และช่วงเวลาในการส่งเอกสาร นักศึกษาสามาจองวันที่และเวลาในการส่งเอกสารฉบับจริงของตนได้จากหน้าจอดังกล่าวโดยมีเงื่อนไขคือนักศึกษาผู้ที่จะทำการส่งเอกสารฉบับจริงจำเป็นต้องส่งภาพสำเนาเอกสารผ่านทางระบบเพื่อให้เจ้าหน้าที่ยืนยันความถูกต้องเสียก่อน

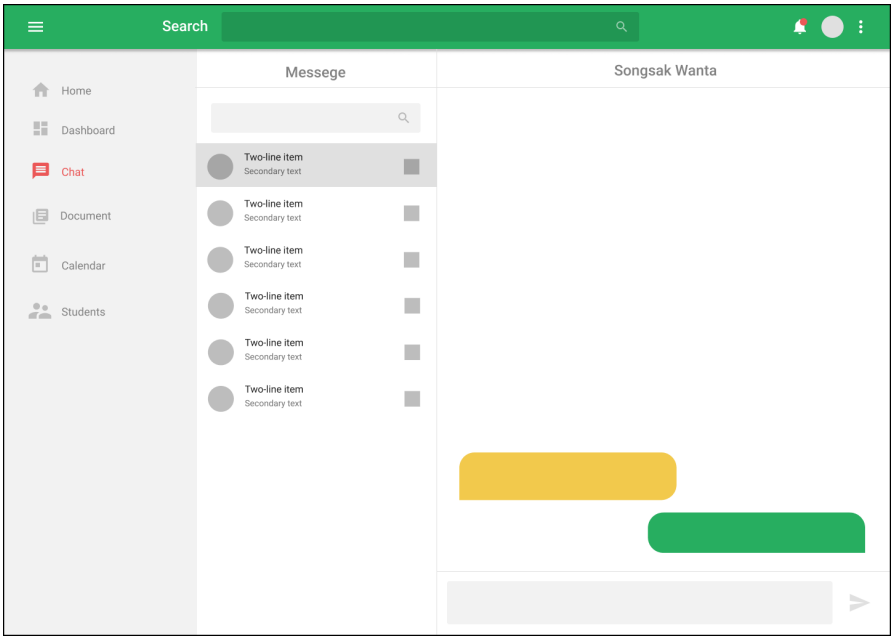
- เว็บแอปพลิเคชัน
  - การออกแบบหน้าจอหลัก



รูปที่ 3.15: หน้าจอหลัก

จากภาพที่ 4.6 แสดงหน้าจอหลักบนเว็บแอปพลิเคชัน เพื่ออำนวยความสะดวกต่อเจ้าหน้าที่ ในหน้าหลักได้รวบรวมข้อมูลสรุปและเมนูเข้าถึงด่วนซึ่งแบ่งเป็น 3 ส่วนหลักได้แก่ เมนูนำทาง ข่าวสารประชาสัมพันธ์และปฏิทินกำหนดการ

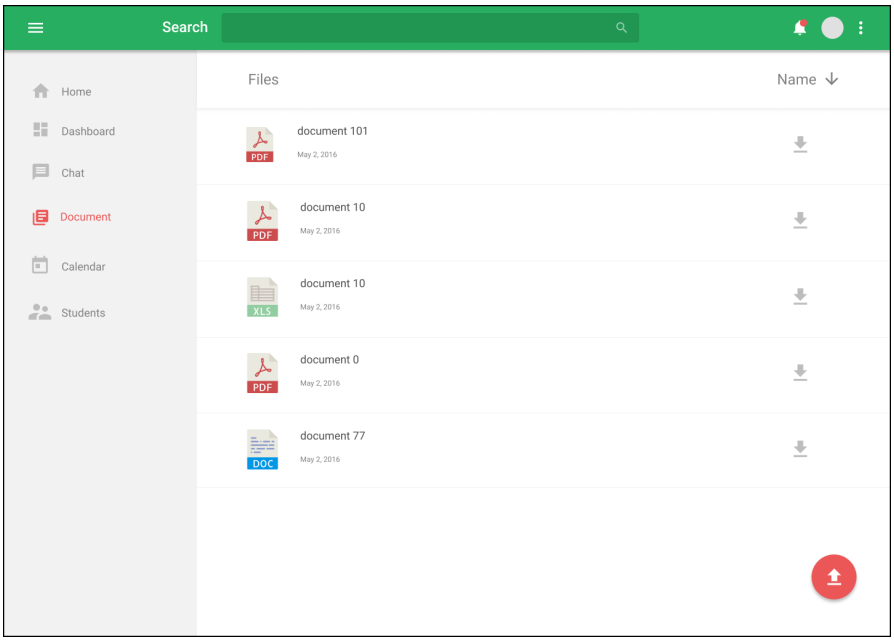
– การออกแบบหน้าจอสนทนา



รูปที่ 3.16: หน้าจอสนทนา

จากภาพที่ 3.16 แสดงหน้าจอสนทนามีการแสดงรายชื่อนักศึกษาและส่วนของห้องสนทนาด้วย

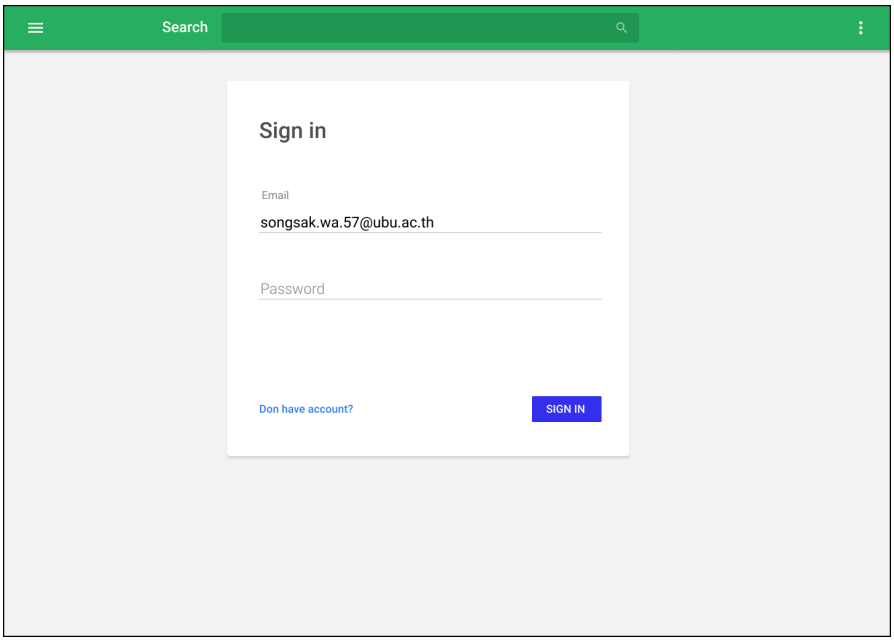
– การออกแบบหน้าจออัพโหลดเอกสาร



รูปที่ 3.17: หน้าจออัพโหลดเอกสาร

จากภาพที่ 3.17 แสดงหน้าจออัพโหลดเอกสารที่เกี่ยวข้องกับกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ทั้งนี้ผู้ที่มีสิทธิ์ในการอัพโหลดเอกสารมีเพียงเจ้าหน้าที่เท่านั้น

– การออกแบบหน้าจอเข้าสู่ระบบ



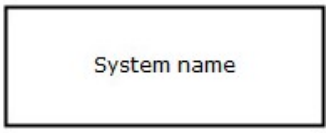
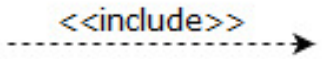
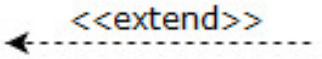
รูปที่ 3.18: หน้าจอเข้าสู่ระบบ

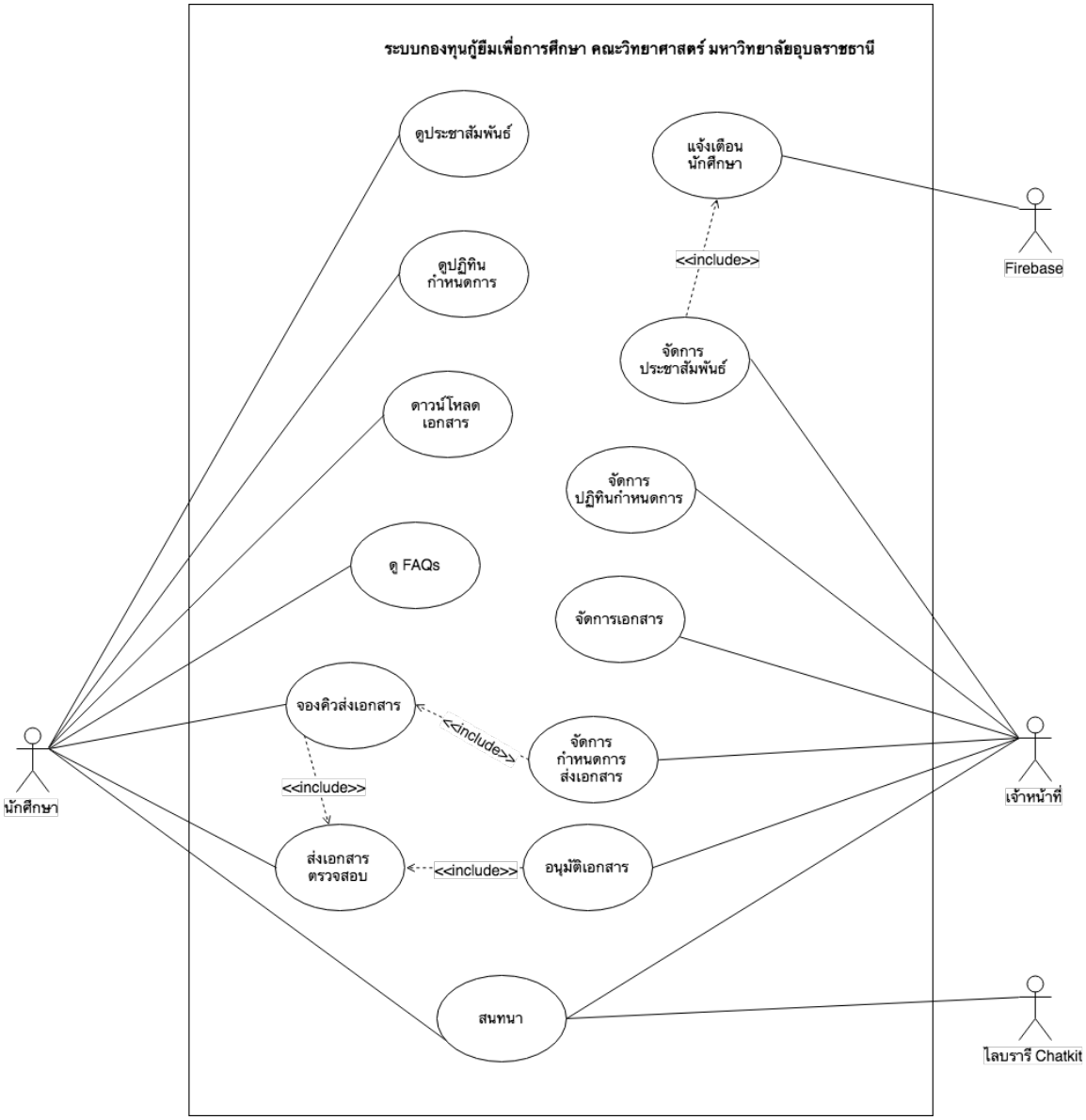
จากภาพที่ 3.18 แสดงหน้าจอการเข้าสู่ระบบของผู้ใช้โดยผู้ใช้จำเป็นต้องกรอกข้อมูลอีเมลและรหัสผ่านเพื่อเข้าใช้งานระบบ

3.4 Use Case Diagram

Use Case Diagram เป็นแผนผังเพื่อแสดงฟังก์ชันแสดงการทำงานของระบบโดยรวม แสดงส่วนประกอบในระบบและกิจกรรมที่เกิดขึ้นในระบบซึ่งในระบบระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา 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 คือบุคคลหรือระบบงานอื่นที่ใช้งานระบบหรือได้รับประโยชน์จากระบบซึ่งอยู่ภายนอกระบบ แทนด้วยรูปคนและมีชื่อบทบาทการใช้งานระบบ
	เส้นตรงที่แสดงถึงการใช้งาน Use case ของผู้กระทำ
	กรอบ สี่เหลี่ยม แสดง ถึง ขอบเขต ของ ระบบ โดย แสดง ชื่อ ระบบ ภายในหรือด้านบนกรอกสี่เหลี่ยม Use case อยู่ภายในกรอบสี่เหลี่ยม และ actor อยู่ภายนอกกรอบสี่เหลี่ยม
	ความสัมพันธ์แบบ «includes» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัณลักษณ์ลูกศรเส้นประ ซึ่ง Use case ที่หางลูกศรเรียกใช้งาน Use case ที่หัวลูกศรทุกครั้งที่มีการทำงาน
	ความสัมพันธ์แบบ «extend» แสดงว่า Use case หนึ่งดำเนินการตามขั้นตอนของ Use case อื่น โดยแทนด้วยสัญลักษณ์ลูกศรเส้นประ ซึ่ง Use case ที่หัวลูกศรเรียกใช้งาน Use case ที่หางลูกศร แต่การใช้งานไม่จำเป็นต้องเกิดขึ้นทุกครั้งขึ้นอยู่กับเงื่อนไขระหว่างการทำงาน



รูปที่ 3.19: Use Case Diagram ของระบบ XX

ตารางที่ 3.2: อธิบาย Use Case หน้าที่ของระบบ ในภาพที่ 3.19

Use Case	คำอธิบาย
ดูประชาสัมพันธ์	นักศึกษาสามารถดูประชาสัมพันธ์ได้โดยไม่จำเป็นต้องทำการเข้าสู่ระบบก่อน
ดูปฏิทินกำหนดการ	นักศึกษาสามารถเปิดดูปฏิทินวันที่และเวลากำหนดการของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์วิทยาลัยอุบลราชธานี โดยไม่จำเป็นต้องทำการเข้าสู่ระบบก่อน
ดาวน์โหลดเอกสาร	นักศึกษาดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนกู้ยืมการศึกษา คณะวิทยาศาสตร์มหาวิทยาลัยอุบลราชธานีได้โดยไม่จำเป็นต้องทำการเข้าสู่ระบบก่อน
ดู FAQs	หน้าแสดงรายการคำถามที่พบบ่อย
จองคิวส่งเอกสาร	เมื่อนักศึกษาเข้าสู่ระบบเรียบร้อยแล้วนักศึกษา สามารถจองคิววันที่และเวลาในการส่งเอกสารฉบับจริงหลังจากที่ได้รับการตรวจสอบโดยเจ้าหน้าที่เรียบร้อยแล้ว
ส่งเอกสารตรวจสอบ	นักศึกษาสามารถส่งเอกสารหน้าที่ตรวจสอบโดยการถ่ายรูปแล้วทำการอัพโหลดเข้าสู่ระบบ เมื่อเจ้าหน้าที่ตรวจสอบเรียบร้อยแล้วจะยืนยันสถานะของเอกสารอีกที
สนทนา	นักศึกษาสามารถสอบถามข้อมูลของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ได้จากช่องสนทนา
จัดการกำหนดการส่งเอกสาร	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบกำหนดการส่งเอกสารของนักศึกษา

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

Use Case	คำอธิบาย
จัดการประชาสัมพันธ์	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบประชาสัมพันธ์
จัดการปฏิทินกำหนดการ	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบกำหนดการการดำเนินการของกองทุน
จัดการเอกสาร	ใช้สำหรับเจ้าหน้าที่เพื่อ เพิ่ม แก้ไขหรือลบเอกสารที่เกี่ยวข้องของกองทุน
แจ้งเตือนนักศึกษา	ใช้เพื่อส่งแจ้งเตือนไปยังนักศึกษาเมื่อมีการเพิ่มประชาสัมพันธ์โดยเจ้าหน้าที่

ตารางที่ 3.4: Use Case ดูประชาสัมพันธ์

Use Case Title : ดูประชาสัมพันธ์	Use case Id : 1
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาดูประชาสัมพันธ์โดยไม่จำเป็นต้องเข้าสู่ระบบ	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูประชาสัมพันธ์ได้	

ตารางที่ 3.5: Use Case ดูปฏิทินกำหนดการ

Use Case Title : ดูปฏิทินกำหนดการ	Use case Id : 2
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาสามารถเปิดดูปฏิทินวันที่และเวลากำหนดการของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ไม่จำเป็นต้องเข้าสู่ระบบ	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูปฏิทินกำหนดการได้	

ตารางที่ 3.6: Use Case ดาวน์โหลดเอกสาร

Use Case Title : ดาวน์โหลดเอกสาร	Use case Id : 3
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนกู้ยืมการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานีได้ไม่จำเป็นต้องเข้าสู่ระบบ	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนได้	
Exceptional Flow ที่ 2 : หากผู้ใช้โมบายแอปพลิเคชันไม่เปิดสิทธิ์การอ่านและเขียนไฟล์บนความจำสำรอง จะไม่สามารถดาวน์โหลดเอกสารที่เกี่ยวข้องกับกองทุนได้	

ตารางที่ 3.7: Use Case ดู FAQs

Use Case Title : ดู FAQs	Use case Id : 4
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : ใช้เพื่อแสดงรายการคำถามที่พบบ่อย	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถดูคำถามที่พบบ่อยได้	

ตารางที่ 3.8: Use Case จองคิวส่งเอกสาร

Use Case Title : จองคิวส่งเอกสาร	Use case Id : 5
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow :เมื่อนักศึกษาเข้าสู่ระบบเรียบร้อยแล้วนักศึกษา สามารถจองคิววันที่และเวลาในการส่งเอกสารฉบับจริงหลังจากที่ได้รับการตรวจสอบโดยเจ้าหน้าที่เรียบร้อยแล้ว	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถจองคิวส่งเอกสารได้	

ตารางที่ 3.9: Use Case ส่งเอกสารตรวจสอบ

Use Case Title : ส่งเอกสารตรวจสอบ	Use case Id : 6
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : นักศึกษาสามารถส่งเอกสารหน้าที่ตรวจสอบโดยการถ่ายรูปแล้วทำการอัพโหลดเข้าสู่ระบบ เมื่อเจ้าหน้าที่ตรวจสอบเรียบร้อยแล้วจะยืนยันสถานะของเอกสารอีกที	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถส่งเอกสารตรวจสอบได้	
Exceptional Flow ที่ 1 : หากผู้ใช้โมบายแอปพลิเคชันไม่เปิดสิทธิ์ใช้งานกล้อง จะไม่สามารถถ่ายภาพเพื่อส่งเอกสารตรวจสอบได้	

ตารางที่ 3.10: Use Case สนทนา

Use Case Title : สนทนา	Use case Id : 7
Primary Actor : นักศึกษา	
Stakeholder Actor : เจ้าหน้าที่	
Main Flow : เมื่อนักศึกษาเข้าสู่ระบบแล้วจะสามารถสอบถามข้อมูลของกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี ได้จากช่องสนทนา	
Exceptional Flow ที่ 1 : หากผู้ใช้ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถสนทนาได้	

ตารางที่ 3.11: Use Case จัดการกำหนดการส่งเอกสาร

Use Case Title : จัดการกำหนดการส่งเอกสาร	Use case Id : 10
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบกำหนดการส่งเอกสารของนักศึกษา	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบกำหนดการส่งเอกสารของนักศึกษาได้	

ตารางที่ 3.12: Use Case จัดการประชาสัมพันธ์

Use Case Title : จัดการประชาสัมพันธ์	Use case Id : 11
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบปฏิทินกำหนดการในระบบได้	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบปฏิทินกำหนดการได้	

ตารางที่ 3.13: Use Case จัดการปฏิทินกำหนดการ

Use Case Title : จัดการปฏิทินกำหนดการ	Use case Id : 12
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบปฏิทินกำหนดการได้	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบปฏิทินกำหนดการได้	

ตารางที่ 3.14: Use Case จัดการเอกสาร

Use Case Title : จัดการเอกสาร	Use case Id : 13
Primary Actor : เจ้าหน้าที่	
Stakeholder Actor : -	
Main Flow : เจ้าหน้าที่ เพิ่ม แก้ไขหรือลบเอกสารที่เกี่ยวข้องกับกองทุนได้	
Exceptional Flow ที่ 1 : หากเจ้าหน้าที่ไม่เชื่อมต่ออินเทอร์เน็ต จะไม่สามารถ เพิ่ม แก้ไขหรือลบเอกสารที่เกี่ยวข้องกับกองทุนได้	

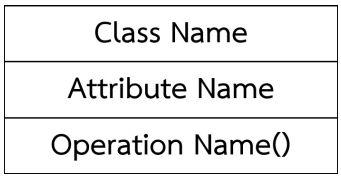
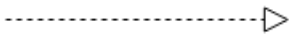
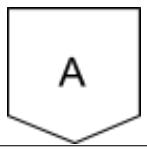
ตารางที่ 3.15: Use Case แจ้งเตือนนักศึกษา

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

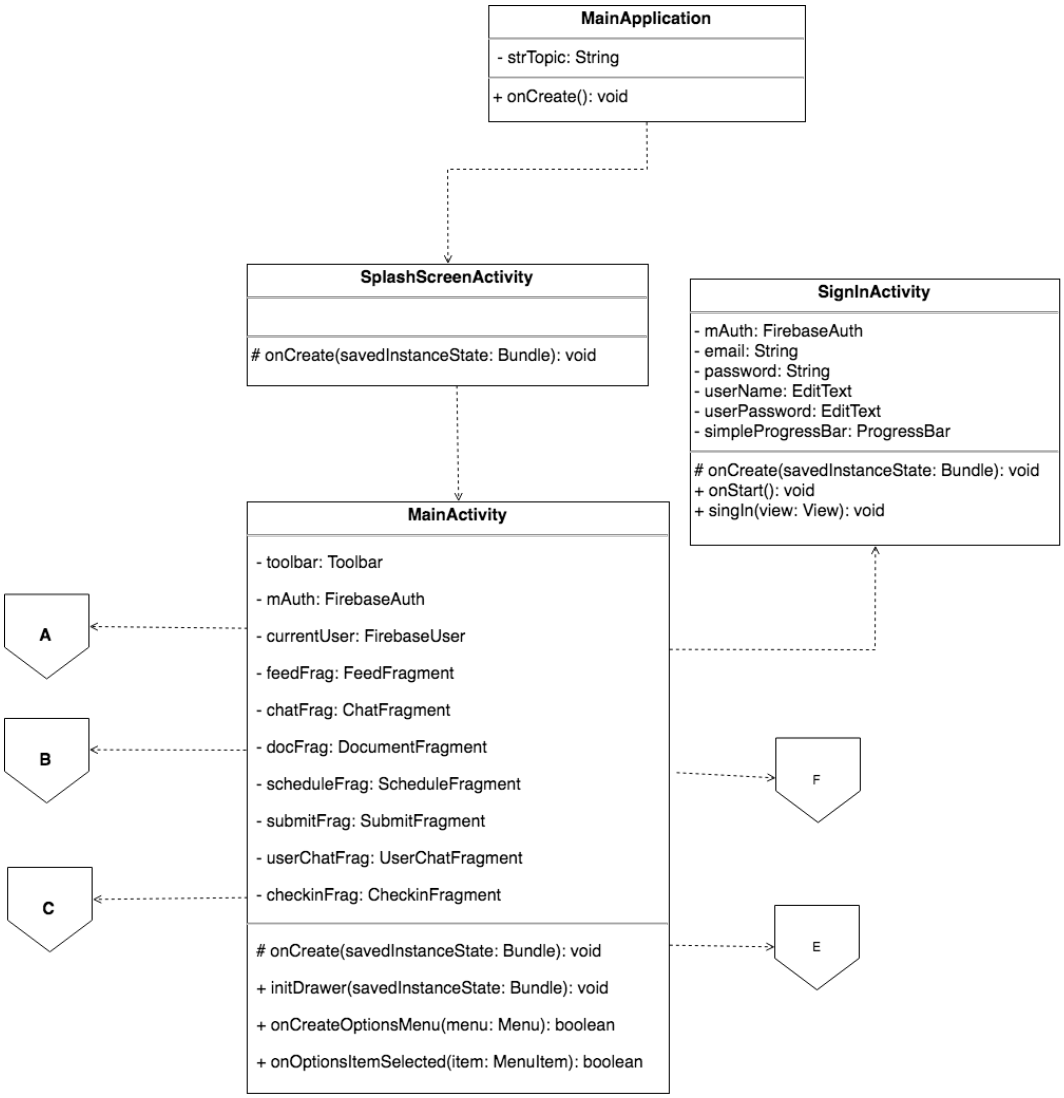
3.5 Class Diagram

Class Diagram คือแผนภาพที่ใช้แสดงคลาสและความสัมพันธ์ในแบบต่างๆ ระหว่างคลาส สัญลักษณ์ที่ใช้ในการเขียน Class Diagram แสดงในตารางที่ 3.16

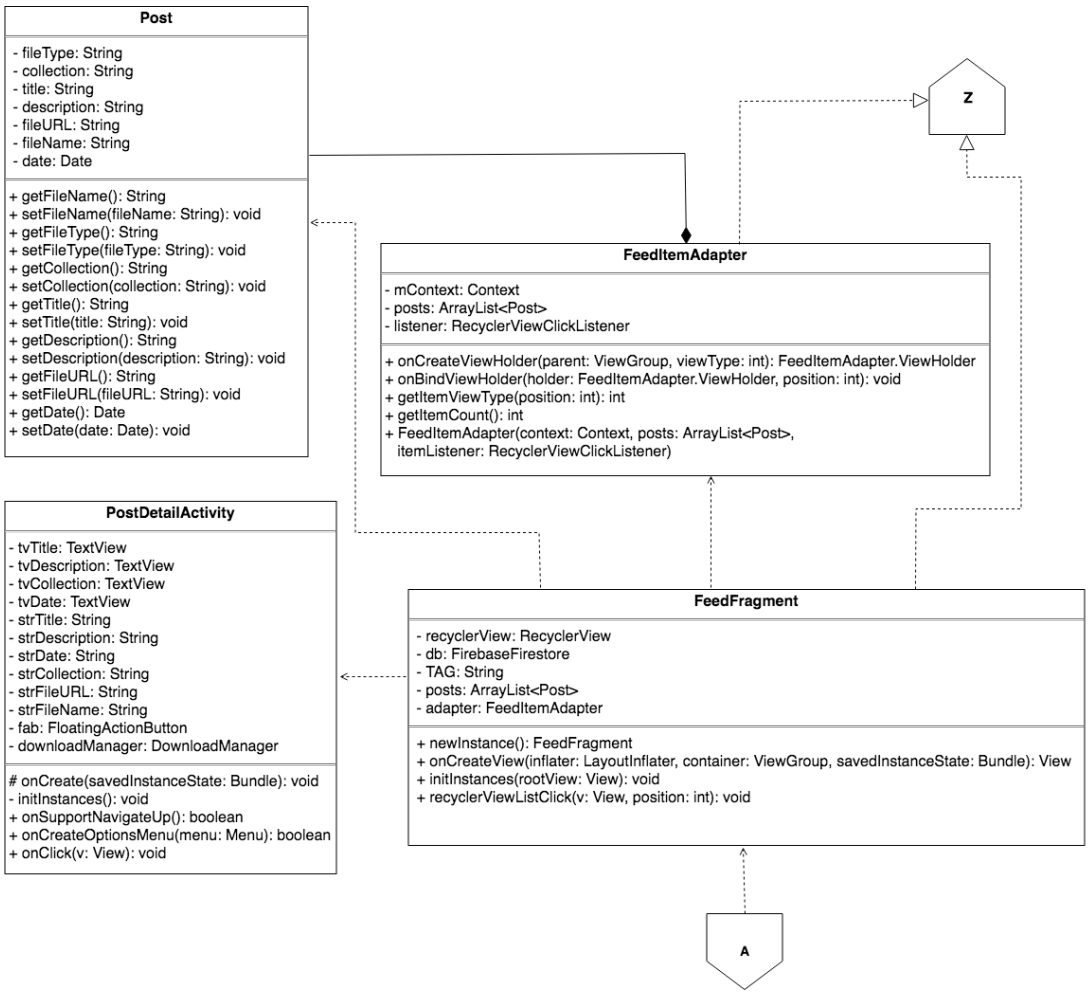
ตารางที่ 3.16: สัญลักษณ์ของ Class Diagram

สัญลักษณ์	การใช้งาน
	คลาส สัญลักษณ์แทนด้วยสี่เหลี่ยมแบ่งเป็น 3 ส่วน ส่วนบน เป็นชื่อของ class ส่วนกลาง เป็นชื่อ Attribute และส่วนล่างเป็น Operation Name หรือ Method ใช้สำหรับเขียนฟังก์ชันในการทำงานของคลาสนั้น ๆ ชนิดของ Visibility ของ Method และ Attribute แบ่งเป็น 3 ชนิด ได้แก่ 1. Public แทนสัญลักษณ์ด้วยเครื่องหมายบวก (+) 2. Private แทนสัญลักษณ์ด้วยเครื่องหมายลบ (-) 3. Protected แทนสัญลักษณ์ด้วยเครื่องหมายชาร์ป ()
	Dependency Relationship หมายความว่า คลาสที่อยู่ฝั่งต้นลูกศรสามารถเรียกใช้คลาสที่อยู่ฝั่งหัวลูกศร
	Composition Relationship เป็นความสัมพันธ์ระหว่างออบเจ็กต์หรือคลาสแบบขึ้นต่อกันและมีความเกี่ยวข้องกันเสมอ
	Realization Relationship เป็นความสัมพันธ์ระหว่าง Object หรือ Class ในลักษณะของการสืบทอดคุณสมบัติจาก Class หนึ่ง (Super class) ไปยังอีก Class หนึ่ง (Subclass)
	Connector เป็นสัญลักษณ์แทนด้วยรูปห้าเหลี่ยมและมีชื่ออยู่ตรงกลาง จะสร้างสัญลักษณ์นี้ไว้เมื่อต้องการเชื่อมต่อคลาสที่อยู่คนละหน้า

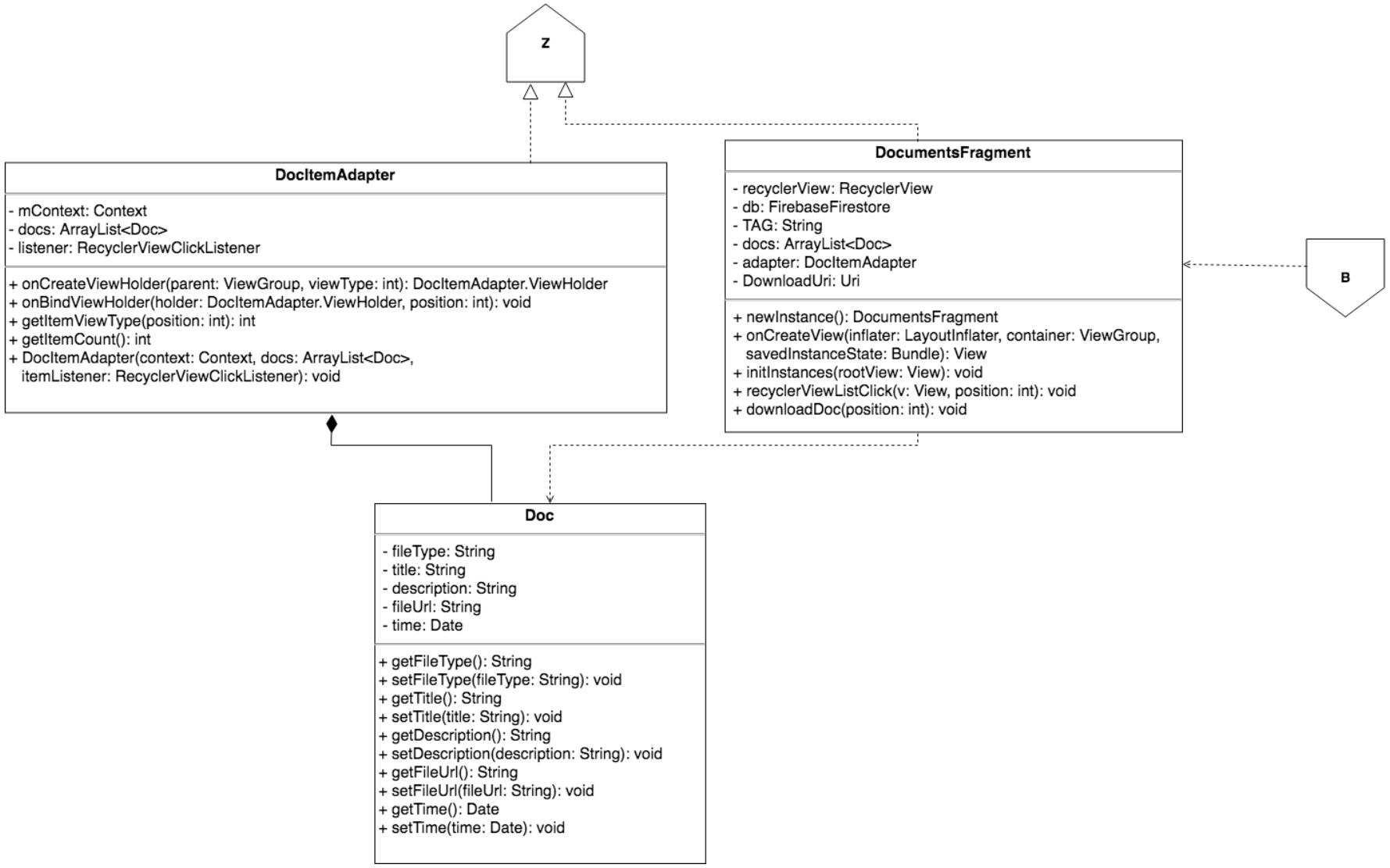
Class Diagram แสดงความสัมพันธ์ในรูปแบบต่างๆ ระหว่างคลาสของแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี อธิบายได้ตามภาพที่ 3.20 ดังต่อไปนี้



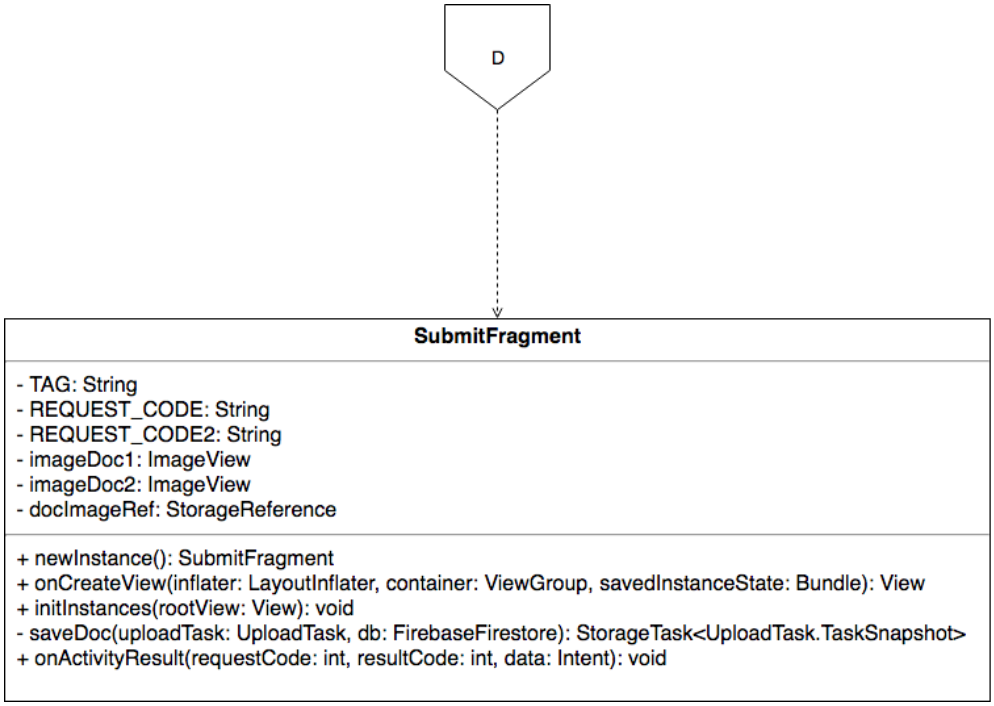
รูปที่ 3.20: Class Diagram ของแอปพลิเคชันระบบ XX



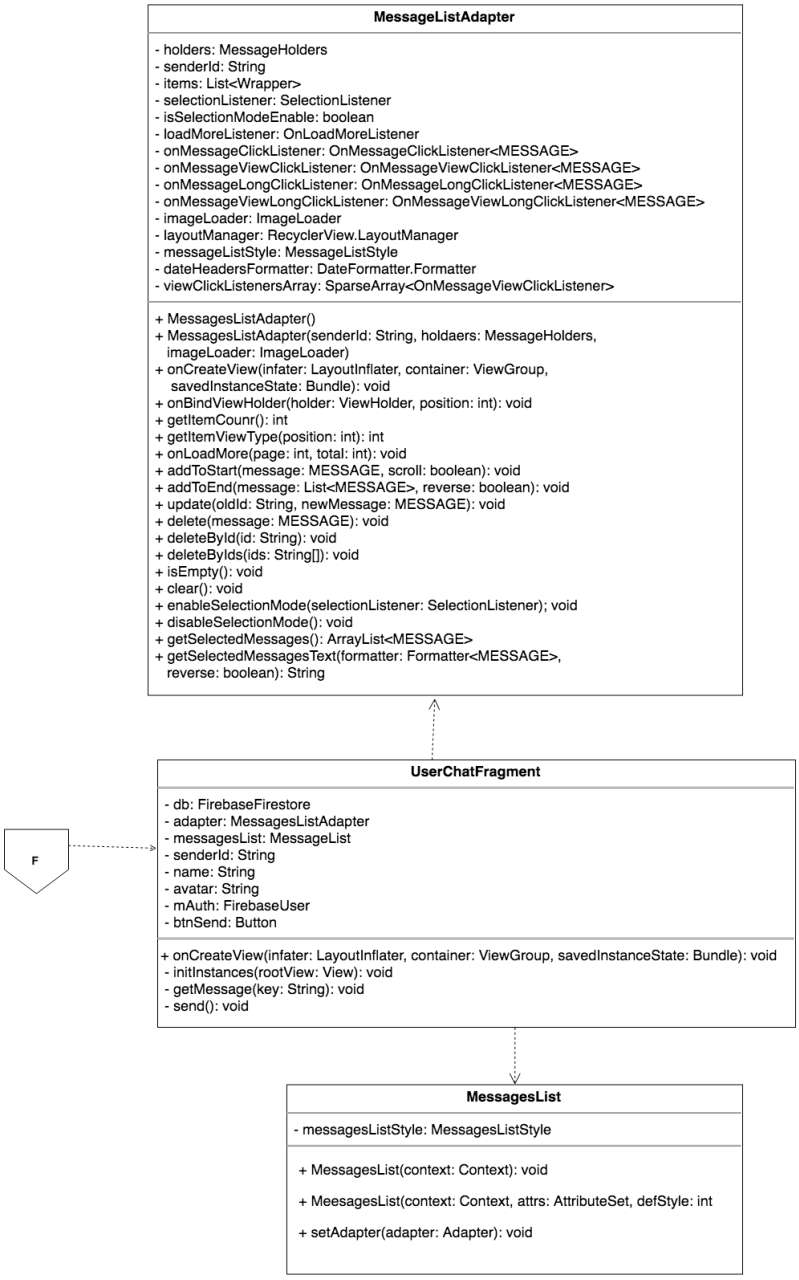
รูปที่ 3.21: Class Diagram ของแอปพลิเคชันระบบ XX



รูปที่ 3.22: Class Diagram ของแอปพลิเคชันระบบ XX



รูปที่ 3.23: Class Diagram ของแอปพลิเคชันระบบ XX



รูปที่ 3.24: Class Diagram ของแอปพลิเคชันระบบ XX

จากรูปภาพที่ 3.20 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.17: อธิบาย Class Diagram ของคลาสพื้นฐานของระบบ

Class Diagram	คำอธิบาย
MainApplication	คลาส MainApplication จะถูกเรียกใช้งานทุกครั้งเมื่อผู้ใช้เปิดแอปพลิเคชัน โดยวัตถุประสงค์การทำงานของคลาสนี้คือ เพื่อใช้จัดการทรัพยากรที่จำเป็นสำหรับการใช้งานในคลาสอื่น ๆ
SplashScreenActivity	คลาส SplashScreenActivity จะถูกเรียกใช้งานทุกครั้งเมื่อผู้ใช้เปิดแอปพลิเคชัน โดยวัตถุประสงค์การทำงานของคลาสคือ เพื่อใช้ตรวจสอบสถานะการเข้าสู่ระบบของผู้ใช้
MainActivity	คลาส MainActivity เป็นคลาสหลักที่ใช้ในการทำงานของแอป-พลิเคชันโดยการทำงานของคลาสนี้เน้นไปที่การสร้าง Fragment เพื่อใช้แสดงข้อมูลต่าง ๆ โดยองค์ประกอบการทำงานของคลาสนี้ประกอบบไปด้วยสองส่วนหลักๆ ได้แก่ เมนูนำทาง Drawer และ Fragment Container
SignInActivity	คลาส SignInActivity เป็นคลาสที่ใช้เพื่อให้สมาชิกที่ได้ลงทะเบียนกับระบบเข้าสู่ระบบเพื่อใช้งานบริการต่าง ๆ จากระบบ

จากรูปภาพที่ 3.21 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 เป็น คลาส ที่ มีหน้า ที่ ใน การ แสดง ข้อมูล ราย ละเอียด ของ ข่าวสาร แต่ละ แถว ที่ ได้ รับ จาก หน้า FeedFragment ที่จะส่งข้อมูลเมื่อผู้ใช้กดที่แถวรายการข่าวสาร
RecyclerViewClickListener	คลาส RecyclerViewClickListener เป็น คลาส อินเทอร์เฟส(Interface)ที่ใช้ในการสร้างแม่แบบเมื่อคลาสใด ๆ ต้องการใช้งานสำหรับการรับค่าเมื่อผู้ใช้กดแถวในลิสต์รายการ คลาสลูกที่ทำการสืบทอดคุณสมบัติจะสามารถรับข้อมูลตำแหน่งแถวที่ผู้ใช้กดบนลิสต์รายการได้

จากรูปภาพที่ 3.22 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 เป็นคลาสโมเดลที่กำหนดค่าต่าง ๆ ที่จำเป็นสำหรับใช้ในการสร้างลิสต์รายการของคลาส DocItemAdapter
RecyclerViewClickListener	คลาส RecyclerViewClickListener เป็นคลาสอินเทอร์เฟสที่ใช้ในการสร้างแม่แบบเมื่อคลาสใด ๆ ต้องการใช้งานสำหรับการรับค่าเมื่อผู้ใช้กดแถวในลิสต์รายการ คลาสลูกที่ทำการสืบทอดคุณสมบัติจะสามารถรับข้อมูลตำแหน่งแถวที่ผู้ใช้กดบนลิสต์รายการได้

จากรูปภาพที่ 3.23 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

ตารางที่ 3.20: อธิบาย Class Diagram ของส่วนของการส่งสำเนาเอกสาร

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

จากรูปภาพที่ 3.24 สามารถอธิบายแผนภาพ Class Diagram ได้ดังนี้

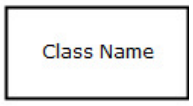
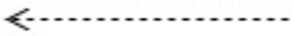
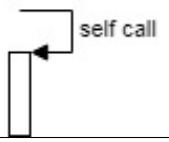
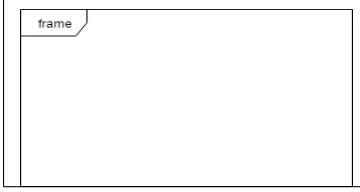
ตารางที่ 3.21: อธิบาย Class Diagram ของส่วนของการสนทนา

Class Diagram	คำอธิบาย
UserChatFragment	คลาส UserChatFragment เป็นคลาสที่ใช้ในการแสดงหน้าจอสนทนาสำหรับนักศึกษา เพื่อติดต่อสอบถามข้อมูลกับเจ้าหน้าที่ มีการสืบค้นข้อมูลประวัติการสนทนาเพื่อส่งไปแปลงเป็นข้อมูลลิสต์รายการที่คลาส MessagesListAdapter
MessagesListAdapter	คลาส MessagesListAdapter เป็นคลาสที่ใช้ในการแปลงชุดข้อมูลที่ได้รับจากคลาส UserChatFragment เป็นลิสต์รายการแล้วทำการคืนค่าลิสต์รายการที่ได้กลับไปยังคลาส UserChatFragment
MessagesList	คลาส MessagesList เป็นคลาสที่ใช้ในการจัดเก็บข้อมูลภายในคลาส UserChatFragment หลังจากที่ได้ทำการสืบค้นข้อมูลการสนทนาจากไฟร์เบสเพื่อส่งไปยังคลาส MessagesListAdapter
RecyclerViewClickListener	คลาส RecyclerViewClickListener เป็นคลาสอินเทอร์เฟสที่ใช้ในการสร้างแม่แบบเมื่อคลาสใด ๆ ต้องการใช้งานสำหรับการรับค่าเมื่อผู้ใช้กดแถวในลิสต์รายการ คลาสลูกที่ทำการสืบทอดคุณสมบัติจะสามารถรับข้อมูลตำแหน่งแถวที่ผู้ใช้กดบนลิสต์รายการได้

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 ในการส่งหรือรับข้อความ แทนด้วยสัญลักษณ์สี่เหลี่ยมมีชื่อคลาสอยู่ภายใน
	Lifeline หรือเส้นอายุขัย แสดงช่วงเวลาตั้งแต่เริ่มสร้าง object ในคลาสนั้น จนกระทั่ง object นั้นถูกทำลาย สัญลักษณ์แทนด้วยเส้นประ
	Focus of control หรือจุดควบคุม เป็นจุดควบคุมที่ object ใช้ทำการส่งหรือรับข้อความ สัญลักษณ์แทนด้วยสี่เหลี่ยม
	Message คือ ข้อความที่รับส่งระหว่าง Object สัญลักษณ์แทนด้วยลูกศรและประกอบด้วย 2 ส่วน คือ ข้อมูล (Data) และฟังก์ชัน (Function)
	Return Message เป็นข้อมูลที่ส่งกลับหลังจากทำงานเสร็จ
	Self call เป็นการเรียกฟังชันก์การทำงานภายในตัวเอง
	สร้างกรอบการทำงานของโปรแกรม เพื่อให้รู้ขอบเขตของการทำงานเช่น ลูป(loop)

รูปที่ 3.25: Sequence Diagram การแสดงข่าวสาร

จากภาพที่ 3.25 สามารถอธิบายแผนภาพ Sequence Diagram แสดงข่าวสาร ได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส FeedFragment เมื่อ FeedFragment ถูกติดตั้งบน MainActivity เมธอด callData() จะสืบค้นข้อมูลจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส FeedItemAdapter โดยมีการคืนค่าเป็นข้อมูลข่าวสารแต่ละแถวและในขั้นตอนสุดท้ายคลาส FeedFragment จะทำการแสดงรายการข้อมูลข่าวสารทั้งหมดออกทางหน้าจอ หากผู้ใช้มีการกดเลือกข่าวสารบางแถวคลาส FeedFragment จะทำการเรียกใช้ PostDetailActivity เพื่อแสดงรายละเอียดข้อมูลข่าวสารของแถวที่ถูกเลือก

รูปที่ 3.26: Sequence Diagram การแสดงปฏิทินกำหนดการ

จากภาพที่ 3.26 สามารถอธิบายแผนภาพ Sequence Diagram แสดงปฏิทินกำหนดการได้ดังนี้ เมื่อ ผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส ScheduleFragment เมื่อ ScheduleFragment ถูกติดตั้งบน MainActivity เมธอด callData() จะสืบค้นข้อมูลกำหนดการของวันปัจจุบันจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส Schedule-ItemAdapter โดยมีการคืนค่าเป็นข้อมูลกำหนดการแต่ละแถวและในขั้นตอนสุดท้ายคลาส Schedule-Fragment จะทำการแสดงรายการกำหนดการวันปัจจุบันออกทางหน้าจอ หากผู้ใช้มีการกดเลือกวันที่ที่ต้องการทราบกำหนดการจากปฏิทินคลาส ScheduleFragment จะทำการเรียกใช้ callData() อีกครั้งโดยสืบค้นข้อมูลกำหนดการของวันที่ถูกเลือกจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส ScheduleItemAdapter โดยมีการคืนค่าเป็นข้อมูลแต่กำหนดการละแถวและในขั้นตอนสุดท้ายคลาส ScheduleFragment จะทำการแสดงรายการกำหนดการวันที่ผู้ใช้เลือกออกทางหน้าจอ

รูปที่ 3.27: Sequence Diagram การแสดงดาวน์โหลดเอกสาร

จากภาพที่ 3.27 สามารถอธิบายแผนภาพ Sequence Diagram แสดงดาวน์โหลดเอกสาร ได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส DocumentsFragment เมื่อ DocumentsFragment ถูกติดตั้งบน MainActivity เมธอด initInstances() จะสืบค้นข้อมูลเอกสารทั้งหมดจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส DocItemAdapter โดยมีการคืนค่าเป็นข้อมูลเอกสารแต่ละแถวและในขั้นตอนสุดท้ายคลาส DocumentsFragment จะทำการแสดงรายการกำหนดการวันปัจจุบันออกทางหน้าจอ

รูปที่ 3.28: Sequence Diagram การแสดงบทสนทนา

จากภาพที่ 3.28 สามารถอธิบายแผนภาพ Sequence Diagram แสดงการสนทานา ได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส UserChatFragment เมื่อ UserChatFrag-ment ถูกติดตั้งบน MainActivity เมธอด getMessage() จะสืบค้นข้อมูลประวัติการสนทนาของผู้ใช้คนปัจจุบันทั้งหมดจากฐานข้อมูลบน Firebase FireStore และส่งข้อมูลที่ได้ไปแปลงที่คลาส MessagesListAdapter โดยมีการคืนค่าเป็นข้อมูลรายการประวัติการสนทนาทั้งหมดและในขั้นตอนสุดท้ายคลาส User-ChatFragment จะทำการแสดงรายการประวัติการสนทนาทั้งหมดออกทางหน้าจอ เมื่อผู้ใช้พิมพ์ข้อความและกดปุ่มส่งระบบจะเรียกให้เมธอด send() เพื่อทำการบันทึกข้อมูลไว้บน Firebase FireStore และทำการแสดงข้อมูลรายการประวัติการสนทนาทั้งหมดที่ถูกอัพเดท

รูปที่ 3.29: Sequence Diagram แสดงส่งเอกสารตรวจสอบ

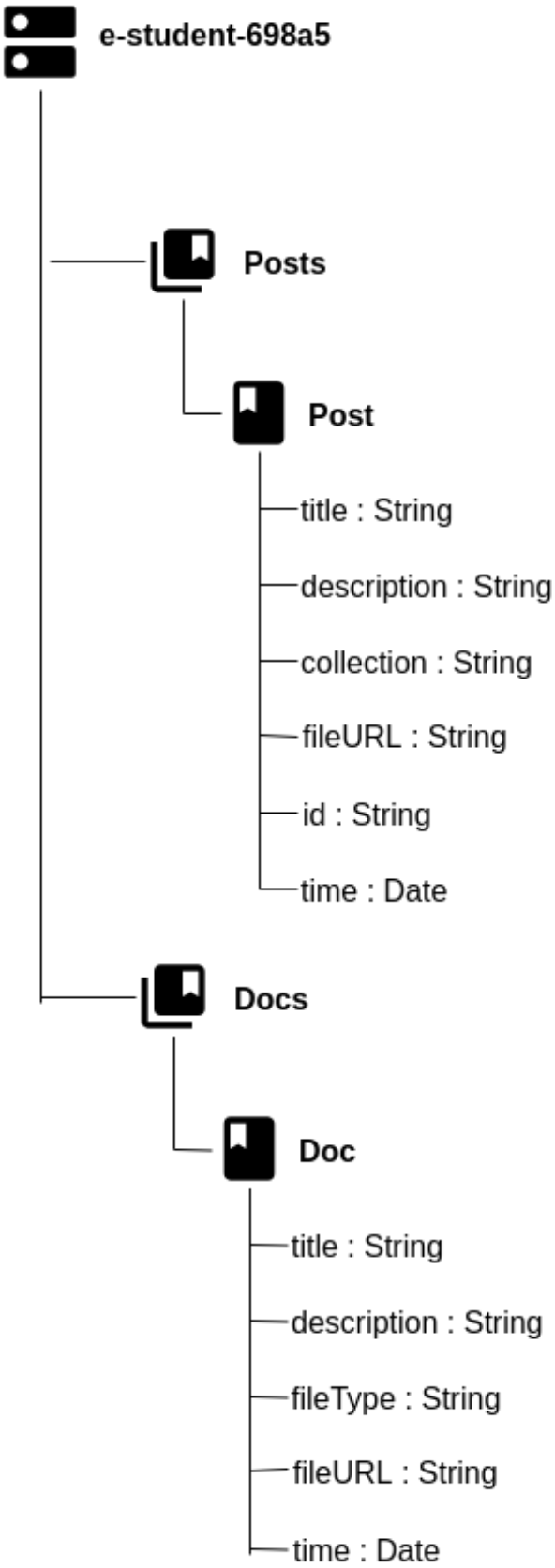
จากภาพที่ 3.29 สามารถอธิบายแผนภาพ Sequence Diagram แสดงส่งเอกสารตรวจสอบได้ดังนี้ เมื่อผู้ใช้เปิดโปรแกรมระบบจะเรียกใช้เมธอด onCreate() ที่คลาส MainActivity ระบบจะทำการสร้าง Fragment ขึ้นมาโดยใช้เมธอด onCreate() ที่คลาส SubmitFragment เมื่อ Submit-Fragment ถูกติดตั้งบน MainActivity เมธอด initInstances() จะถูกเรียกเพื่อสร้างหน้าจอแสงดผลเมื่อผู้ใช้กดปุ่มถ่ายรูประบบจะเรียกใช้ไลบรารี่ ScanConstants เพื่อถ่ายภาพเอกสารและรอให้ผู้ใช้ถ่ายครบทั้งสองแผ่นจึงจะแสดงปุ่มกดส่งเอกสารเพื่อตรวจสอบ

3.7 โครงสร้างฐานข้อมูลไฟร์เบส(Firebase Database Stucture)

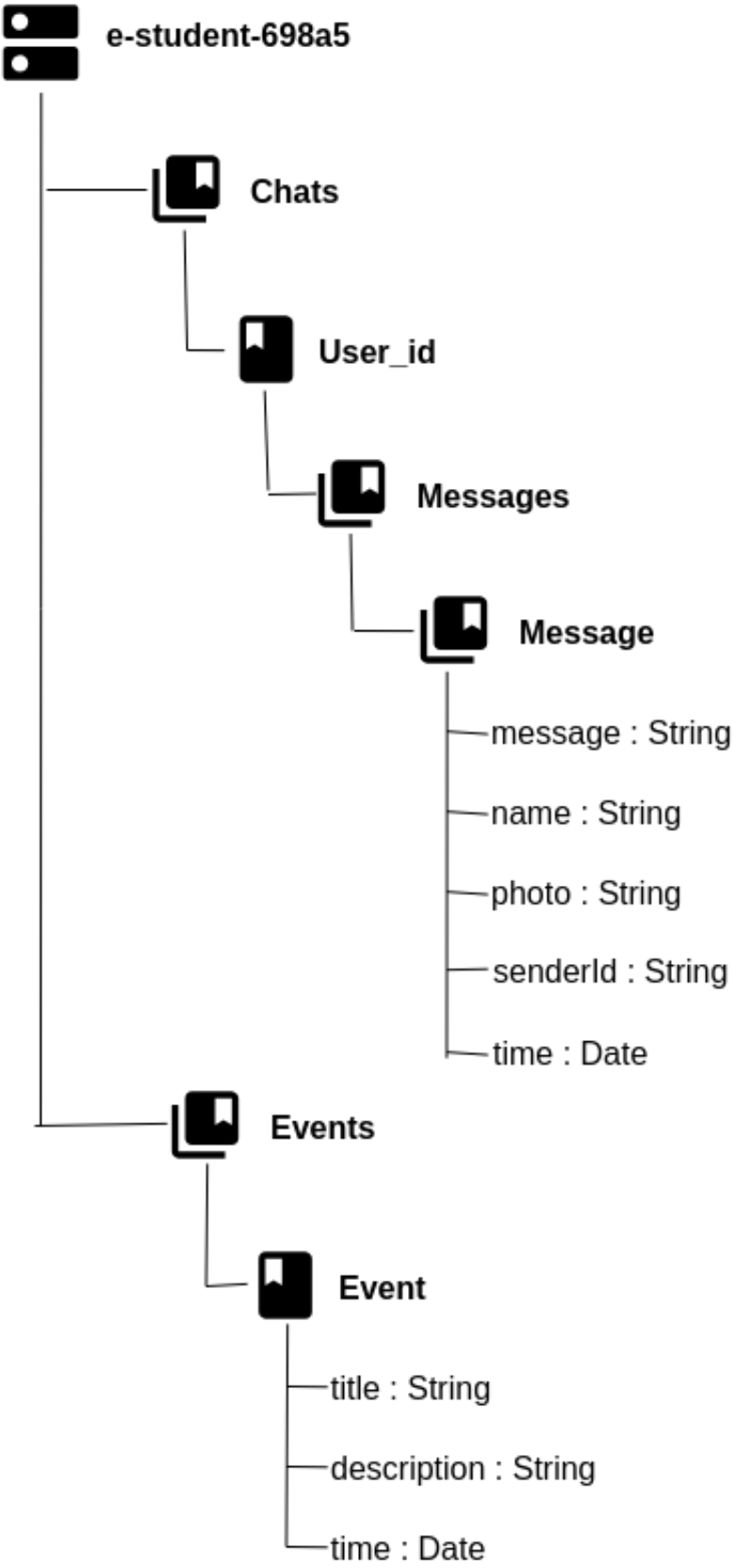
Firebase Database นั้นเป็น Database แบบ NoSQL และเป็น JSON database ที่มีโครงสร้างที่เป็น Key และ Value จัดเก็บข้อมูลในลักษณะโหนด หากต้องการเรียกงานจะเรียกใช้โดย การท่องไปยังโหนดที่ต้องการ ส่วนประกอบสัญลักษณ์ที่ใช้ในการเขียนโครงสร้างฐานข้อมูลแบบ Firebase แสดงดังตารางที่ 3.23

ตารางที่ 3.23: สัญลักษณ์ของโครงสร้างฐานข้อมูลแบบ Firebase

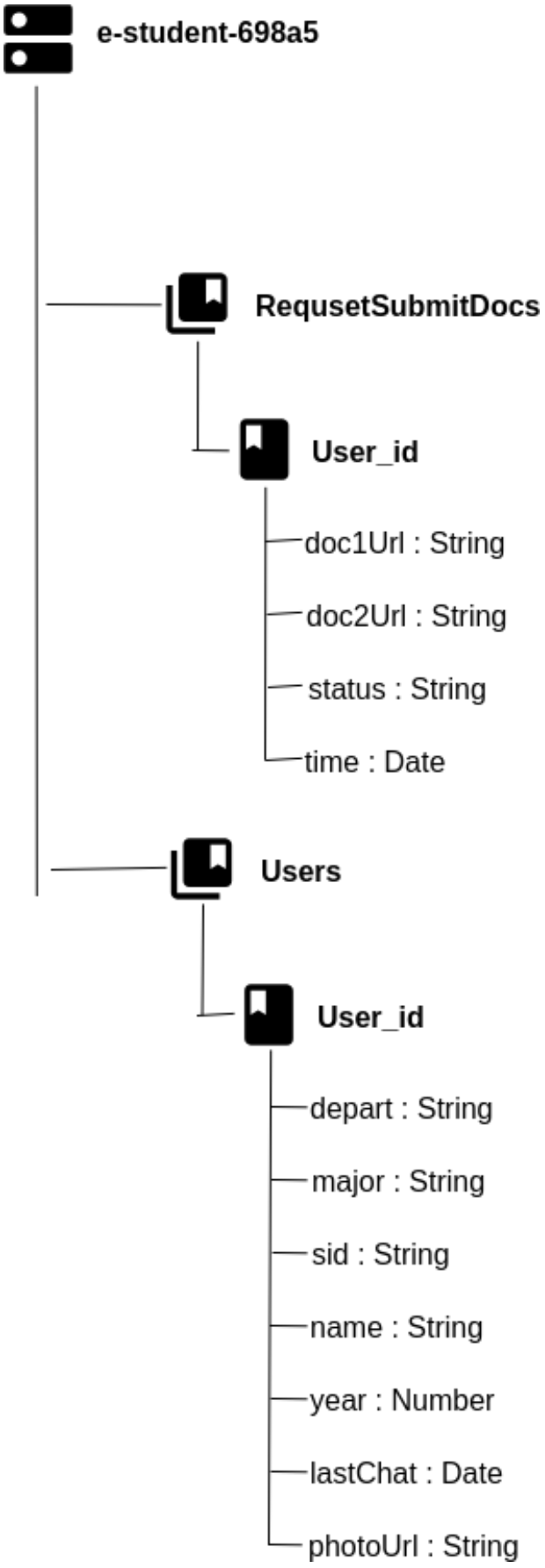
สัญลักษณ์	คำอธิบาย
	Database เป็นการเรียกชื่อแทนโหนด(Node)บนสุดที่ใช้ในการเก็บข้อมูล
	Collection เป็นการเรียกชื่อแทนของการเก็บหลาย ๆ เอกสารไว้ด้วยกัน
	Document เป็นการ เรียก ชื่อ แทน หน่วย การ เก็บ ของ ข้อมูล ใน Cloud Firestore ภายในจะประกอบไปด้วย ชื่อของ Document ชื่อของคีย์ (key) และ ค่าข้อมูล (value)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



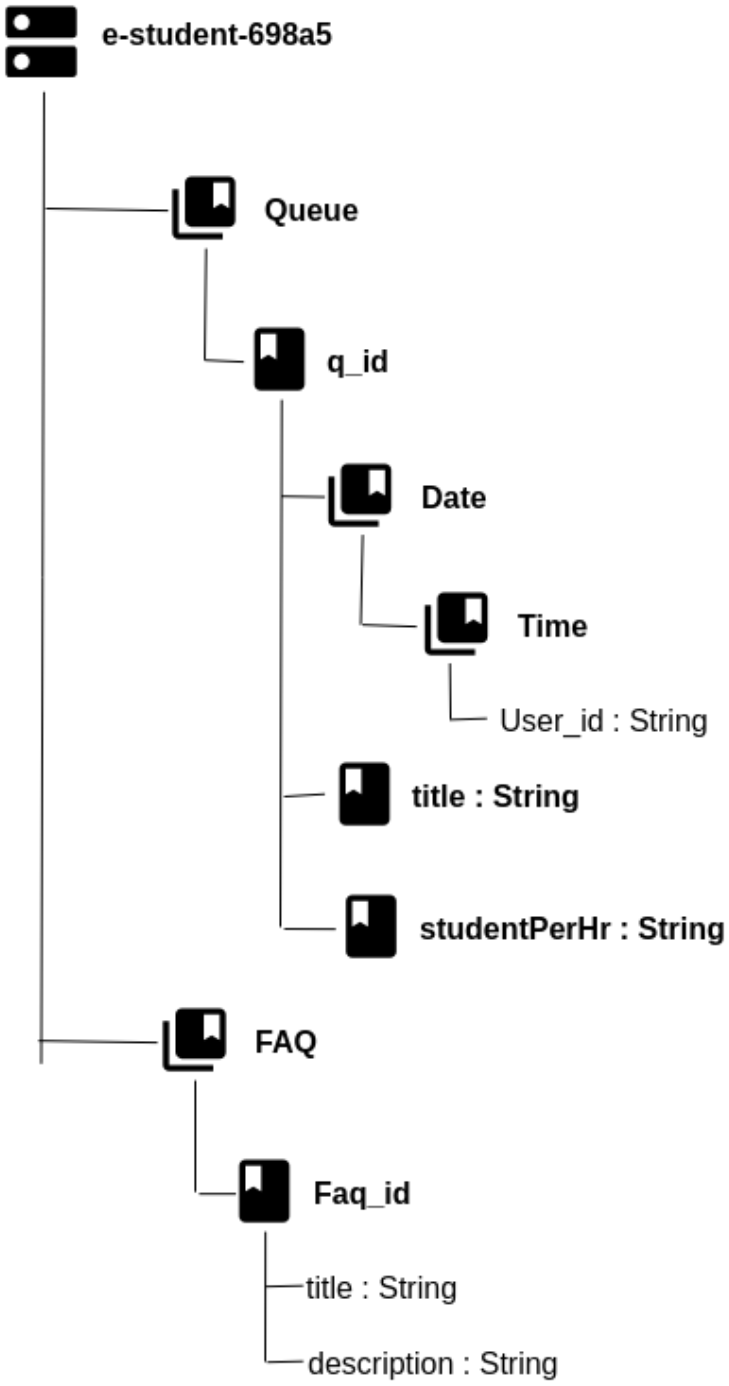
รูปที่ 3.30: โครงสร้างฐานข้อมูลแบบ Firebase



รูปที่ 3.31: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

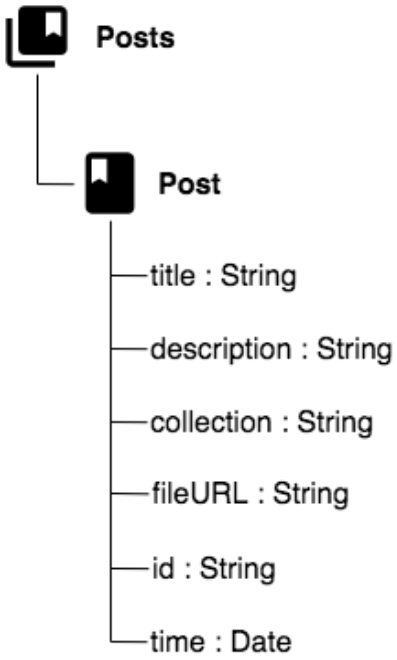


รูปที่ 3.32: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)



รูปที่ 3.33: โครงสร้างฐานข้อมูลแบบ Firebase(ต่อ)

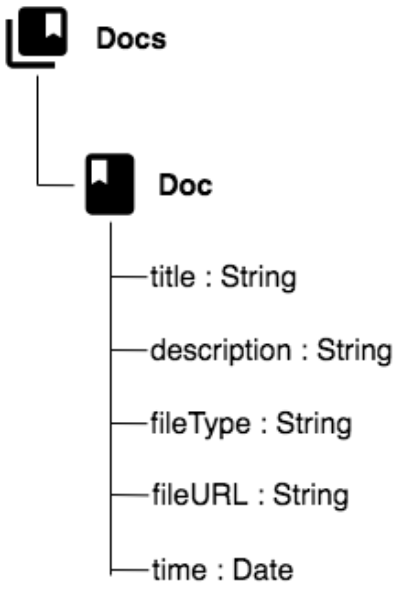
จากรูที่ 3.30-3.41 สามารถอธิบายโครงสร้างของข้อมูลได้ดังนี้



รูปที่ 3.34: โหนดเก็บข้อมูลประกาศ

ตารางที่ 3.24: อธิบายโหนดที่ใช้เก็บข้อมูลประกาศ

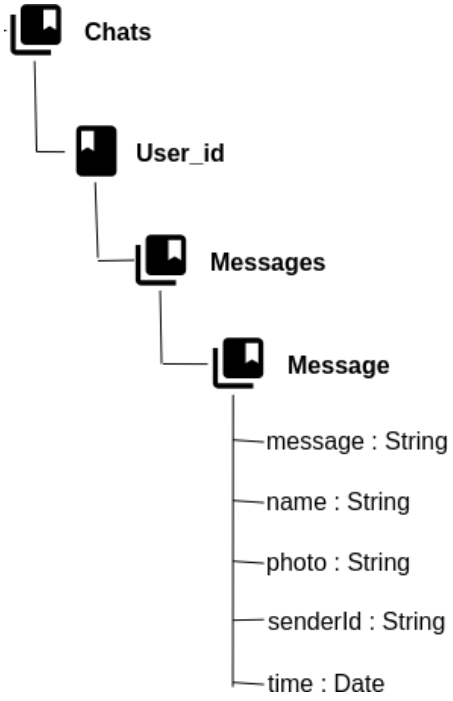
Key	คำอธิบาย
Posts	โหนดสำหรับเก็บข้อมูลประกาศทั้งหมด
Post	สำหรับเก็บข้อมูลแต่ละประกาศ
title	สำหรับเก็บชื่อหัวข้อประกาศ
description	สำหรับเก็บรายละเอียดประกาศ
collection	สำหรับเก็บประเภทของประกาศได้แก่ สาธารณะและเฉพาะบุคคล
fileURL	สำหรับเก็บ url ของเอกสารแนบประกาศ
id	สำหรับเก็บรหัสของประกาศ
time	สำหรับเก็บเวลาที่ประกาศ



รูปที่ 3.35: โหนดเก็บข้อมูลเอกสารที่เกี่ยวข้อง

ตารางที่ 3.25: อธิบายโหนดที่ใช้เก็บข้อมูลเอกสารที่เกี่ยวข้อง

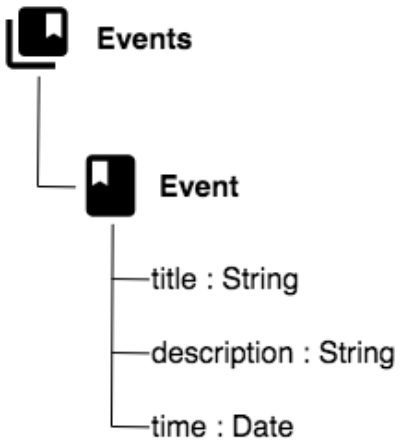
Key	คำอธิบาย
Docs	โหนดสำหรับเก็บข้อมูลของเอกสารที่เกี่ยวข้องทั้งหมด
Doc	สำหรับเก็บข้อมูลเอกสารแต่ละฉบับ
title	สำหรับเก็บชื่อหัวเรื่องของเอกสาร
description	สำหรับเก็บรายละเอียดของเอกสาร
fileType	สำหรับนามสกุลไฟล์เอกสาร เช่น .pdf .png เป็นต้น
fileURL	สำหรับเก็บ url ของเอกสาร
time	สำหรับเก็บเวลาที่ถูกอัพโหลดเข้าสู่ระบบโดยเจ้าหน้าที่



รูปที่ 3.36: โหนดเก็บข้อมูลประวัติการสนทนา

ตารางที่ 3.26: อธิบายโหนดที่ใช้เก็บข้อมูลประวัติการสนทนา

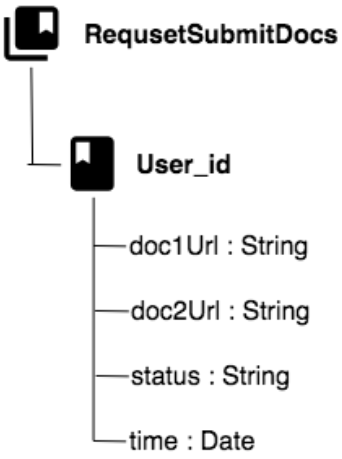
Key	คำอธิบาย
Chats	โหนดสำหรับเก็บข้อมูลประวัติการสนทนาทั้งหมด
User_id	สำหรับเก็บประวัติการสนทนาของผู้ใช้แต่ละคน
Messages	สำหรับเก็บประวัติการสนทนาทั้งหมดของผู้ใช้
Message	สำหรับเก็บข้อมูลของแต่ละข้อความ
message	สำหรับเก็บข้อความ
name	สำหรับเก็บชื่อของผู้ส่งข้อความ
photo	สำหรับเก็บ url รูปภาพของผู้ส่งข้อความ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



รูปที่ 3.37: โหนดเก็บข้อมูลกำหนดการ

ตารางที่ 3.27: อธิบายโหนดที่ใช้เก็บข้อมูลกำหนดการ

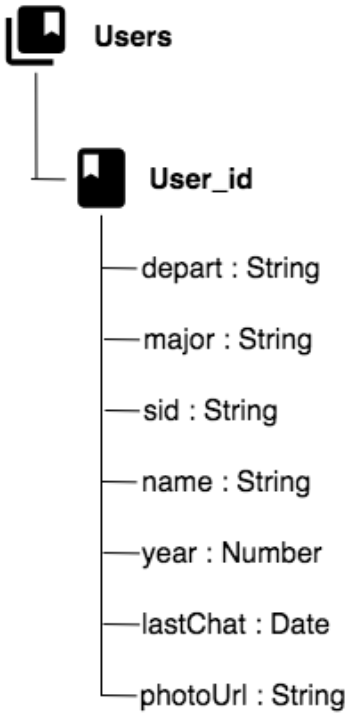
Key	คำอธิบาย
Events	โหนดสำหรับเก็บข้อมูลของกำหนดการทั้งหมด
Event	สำหรับเก็บข้อมูลของแต่ละกำหนดการ
title	สำหรับเก็บชื่อหัวข้อของกำหนดการ
description	สำหรับเก็บรายละเอียดของกำหนดการ
time	สำหรับเก็บเวลาของกำหนดการ



รูปที่ 3.38: โหนดเก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา

ตารางที่ 3.28: อธิบายโหนดที่ใช้เก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษา

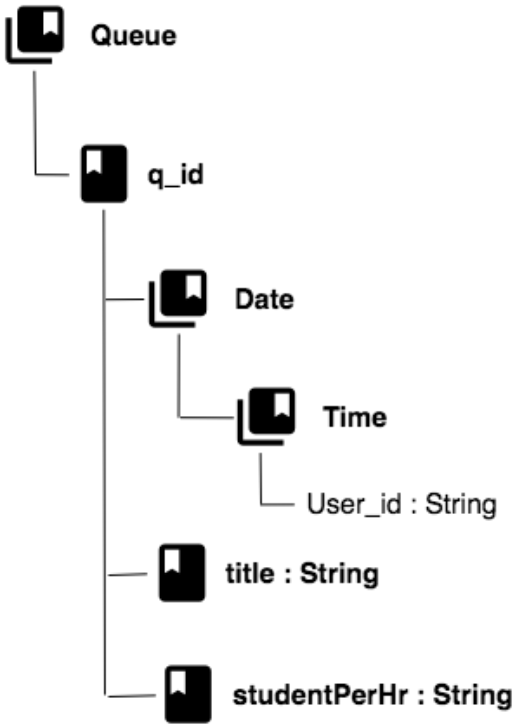
Key	คำอธิบาย
RusetSubmitDocs	โหนดสำหรับเก็บข้อมูลการยื่นสำเนาเอกสารเพื่อตรวจสอบของนักศึกษาทั้งหมด
User_id	สำหรับเก็บข้อมูลของแต่ละสำเนาเอกสารของนักศึกษาแต่ละคน
doc2	สำหรับเก็บ url ของภาพถ่ายสำเนาเอกสารฉบับที่ 1
doc2	สำหรับเก็บ url ของภาพถ่ายสำเนาเอกสารฉบับที่ 2
status	สำหรับเก็บผลการตรวจสอบของเจ้าหน้าที่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



รูปที่ 3.39: โหนดเก็บข้อมูลของนักศึกษา

ตารางที่ 3.29: อธิบายโหนดที่ใช้เก็บข้อมูลของนักศึกษา

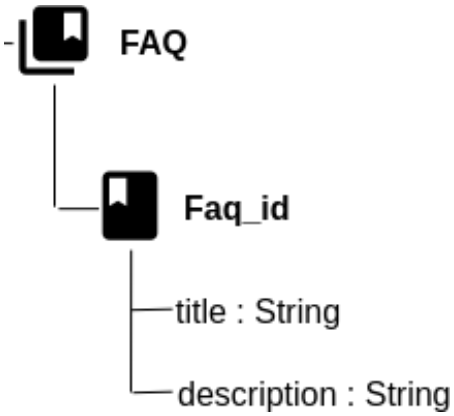
Key	คำอธิบาย
Users	โหนดสำหรับเก็บข้อมูลของนักศึกษา
User_id	สำหรับเก็บข้อมูลของนักศึกษาแต่ละคน
depart	สำหรับเก็บภาควิชาของนักศึกษา
major	สำหรับเก็บสาขาของนักศึกษา
sid	สำหรับเก็บรหัสประจำตัวนักศึกษา
name	สำหรับเก็บชื่อของนักศึกษา
year	สำหรับเก็บชั้นปีของนักศึกษา
lastChat	สำหรับเก็บเวลาที่สนทนากับเจ้าหน้าที่ล่าสุด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



รูปที่ 3.40: โหนดเก็บข้อมูลการจองคิวของนักศึกษา

ตารางที่ 3.30: อธิบายโหนดที่ใช้เก็บข้อมูลการจองคิวของนักศึกษา

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



รูปที่ 3.41: โหนดเก็บข้อมูลคำถามที่พบบ่อย

ตารางที่ 3.31: อธิบายโหนดที่ใช้เก็บข้อมูลคำถามที่พบบ่อย

Key	คำอธิบาย
Queue	โหนดสำหรับเก็บข้อมูลคำถามที่พบบ่อยทั้งหมด
Faq_id	สำหรับเก็บข้อมูลคำถามที่พบบ่อยแต่ละรายการ
title	สำหรับเก็บคำถาม
description	สำหรับเก็บคำตอบ

บทที่ 4

การพัฒนาระบบ

หลังจากที่ได้มีการเตรียมความพร้อมสำหรับการพัฒนาในด้านต่าง ไม่ว่าจะเป็นที่มาและความสำคัญของปัญหา เทคโนโลยีที่มีความเหมาะสมกับระบบ และการออกแบบระบบการทำงาน รวมไปถึงโครงสร้างของข้อมูล ในบทนี้จะเป็นการพูดถึงการสร้างระบบที่ได้มีการออกแบบไว้ในบทที่แล้วจะถูกนำเสนอในบทนี้ โดยการพัฒนาระบบแบ่งได้เป็นส่วนต่าง ๆ ดังนี้

4.1 การพัฒนาเว็บแอปพลิเคชัน

4.2 การพัฒนาเว็บแอปพลิเคชันแอนดรอยด์แอปพลิเคชัน

4.1 การพัฒนาเว็บแอปพลิเคชัน

การพัฒนาระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษาสำหรับเว็บแอปพลิเคชันนั้นวัตถุประสงค์หลังเพื่อสร้างความสะดวกต่อการกำงานของเจ้าหน้าที่อันเนื่องมาจากข้อจำกัดบางประการหากใช้ระบบทำงานบนอุปกรณ์สมาร์ทโฟนเพียงอย่างเดียว โดยตัวเว็บแอปพลิเคชันนี้ถูกพัฒนาขึ้นด้วย Vue.js มีรายละเอียดการทำงานดังนี้

4.1.1 การเชื่อมต่อ Cloud Firestore

bsection, ในการเชื่อมต่อเว็บแอปพลิเคชันกับไฟร์เบสเพื่อใช้บริการต่างๆ ของไฟร์เบส ทำ-ได้ดังนี้

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

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 เป็นการนำเข้าโมดูลตั้งค่าที่ได้จากรูปภาพที่ 4.2
- บรรทัดที่ 5 เป็นการส่งออกโมดูลไฟร์เบสเพื่อใช้ในไฟล์อื่น ๆ ซึ่งเมื่อถึงขั้นตอนนี้การเชื่อต่อบริการไฟร์เบสถือว่าเป็นอันเสร็จ

4.1.2 โครงสร้างของการสร้างหน้าเข้าสู่ระบบ

```
<template>
      <Row type="flex" justify="center" align="middle
        ">
      <Col span="8" class="col">
      <Card style="width:400px">
      <p slot="title">
      <Icon type="ios-person" size="20"></Icon>
      sign in
      </p>
      <a href="#" slot="extra" @click.prevent="
        SignUp">
      create account
      </a>
      <Form class="form" ref="formInline" :model="
        formInline" :rules="ruleInline">
      <FormItem prop="email">
            <Input type="text" v-model="formInline.
              email" placeholder="email">
            <Icon type="ios-email" slot="prepend"></
              Icon>
            </Input>
      </FormItem>
      <FormItem prop="password">
            <Input type="password" v-model="
              formInline.password" placeholder="
              password">
            <Icon type="ios-locked" slot="prepend
              "></Icon>
            </Input>
      </FormItem>
      <FormItem>
      <Button type="primary" :loading="loading"
        @click="handleSubmit('formInline')">
      <span v-if="!loading">sign in</span>
      <span v-else>signing in...</span>
      </Button>
      </FormItem>
      </Form>
      </Card>
      </Col>
      </Row>
</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 เป็นการกำหนดขนาดของเนื้อภายใน
- 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',
          trigger: 'blur' }
11    ],
12    password: [
13     { required: true, message: 'please fill password',
          trigger: 'blur' }
14    ]
15   }
16  }
17 }
```

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 เป็นการกำหนดกที่ใช้ในการตรวจสอบความถูกต้องของอีเมลและรหัสผ่าน

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

รูปที่ 4.5: การสร้างลอจิก (logic) ของหน้าเข้าสู่ระบบ SignIn.vue

จากภาพที่ 4.5 โครงสร้างลอจิกของหน้าเข้าสู่ระบบ 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 โครงสร้างของการสร้างหน้าข่าวสาร

```
<template>
<div style="padding: 16px;">
<Row>
<Col span="20" style="padding:16px;">
<Row v-for="(post,index) in postsData" :key="post.id" style="margin-bottom:16px;">
<Card>
<p slot="title">
<Icon type="social-rss-outline"></Icon>
post
<span style="font-size:11px; color: #95a5a6;">
{{ post.time }}
</span>
</p>
<a href="#" slot="extra" @click.prevent="showData(index)">
<!-- <Icon type="ios-loop-strong"></Icon> -->
detail
</a>
<p>
{{ post.title }}
</p>
</Card>
</Row>
</Col>
<Col span="4" style="padding:16px;">
<Timeline>
<TimelineItem v-for="(event, index) in eventsData" :key="index" >
<Icon type="trophy" slot="dot" v-if="index == 0"></Icon>
<p class="time">{{event.time}}</p>
<p class="content">{{event.title}}</p>
</TimelineItem>
</Timeline>
</Col>
</Row>
</div>
</template>
```

รูปที่ 4.6: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าข่าวสาร 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: การสร้างลอจิก(logic)ของหน้าข่าวสาร Home.vue

จากภาพที่ 4.9 โครงสร้างลอจิกของหน้าข่าวสาร 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   <img :src="post.fileURL"/>
7   </div>
8 </template>
```

รูปที่ 4.8: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้ารายละเอียดข่าวสาร ViewPost.vue

จากภาพที่ 4.8 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าดูรายละเอียดข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-8 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น HTML
- บรรทัดที่ 2-7 ครอบทับเนื้อหาทั้งหมดเพื่อให้ง่ายต่อการจัดการการแสดงผล
- บรรทัดที่ 3 เป็นการแสดงหัวข้อข่าวสาร
- บรรทัดที่ 4 เป็นการแสดงรายละเอียดข่าวสาร
- บรรทัดที่ 6 เป็นการแสดงไฟล์แนบ

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

รูปที่ 4.9: การสร้างลอจิกของหน้าดูรายละเอียดของข่าวสาร ViewPost.vue

จากภาพที่ 4.9 โครงสร้างลอจิกของหน้าดูรายละเอียดของข่าวสาร สามารถอธิบายการทำงานได้ดังนี้

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

4.1.5 โครงสร้างของการสร้างหน้าสนทนา

```
<template>
<Row :gutter="0" type="flex" justify="center" align="middle">
<Col span="16" style="padding: 0px;">
<Card style="min-height: 500px;max-height: 500px;" :padding="0">
<p v-if="!isAdmin" slot="title" style="text-align:center;">
ESP
</p>
<p v-else slot="title" style="text-align:center;">
{{ chatTitle }}
</p>
<Scroll style="background-color: #EEEEEE;">
<ul style="padding:6px;padding-right:8px;">
<li v-for="(item,index) in messages" :key="index" style="margin-bottom:8px;">
<Card v-if="user.uid !== item.id" :padding="6" style="text-align:left;display: inline-block;background-color: #FAFAFA;">
<div>
<p>{{ item.message }}</p>
</div>
</Card>
<div v-else style="text-align:right;">
<Card :padding="6" style="display: inline-block;background-color: #B2E281;">
<p>{{ item.message }}</p>
</Card>
</div>
</li>
</ul>
</Scroll>
<div style="padding:16px;text-align:right;" >
<Input type="textarea" v-model="formItem.message" placeholder="type..." v-on:keyup.enter="send"></Input>
<Button :loading="loading" type="primary" style="margin-top:10px;" size="large" @click="send">send</Button>
</div>
</Card>
</Col>
</Row>
</template>
```

รูปที่ 4.10: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสนทนา Message.vue

จากภาพที่ 4.10 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสนทนา สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-34 เป็นเทมเพลตที่ใช้เพื่อสื่อสารกับ Vue.js ให้แปลงข้อมูลดังกล่าวเป็น HTML
- บรรทัดที่ 4-22 แสดงหน้าต่างสนทนา
- บรรทัดที่ 9 แสดงชื่อคู่สนทนา
- บรรทัดที่ 11-26 แสดงข้อความสนทนา
- บรรทัดที่ 28 แสดงช่องกรอกข้อความสนทนา
- บรรทัดที่ 29 แสดงปุ่มกดส่งข้อความ

```
send() {
 var vm = this;
 vm.loading = true;
 this.formItem.time = new Date();

// check where data shulde update
 let key = "";
 if (this.isAdmin) {
  key = this.chatId;
   if (this.chatTitle === "" || this.chatTitle ===
      null) {
    return;
   }
 } else {
   key = this.formItem.senderId;
 }

 this.formItem.name = this.user.displayName;
 this.formItem.photo = this.user.photoURL

 db
 .collection("Chats")
 .doc(key)
 .collection("messages")
 .add(this.formItem)
 .then(function(docRef) {
   vm.loading = false;
   vm.formItem.message = "";
   db
   .collection("Users")
   .doc(key)
   .update({ lastChat: new Date() })
   .catch(function(error) {
    vm.$Message.error("send message fail");
    vm.loading = false;
   });
 });
}
```

รูปที่ 4.11: การสร้างลอจิกของหน้าสนทนา Message.vue

จากภาพที่ 4.11 โครงสร้างลอจิกของหน้าสนทนา สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1 ชื่อฟังก์ชัน
- บรรทัดที่ 2 ตรวจสอบไอดีของผู้ใช้
- บรรทัดที่ 3 ดึงค่าโปรไฟล์ผู้ใช้คนปัจจุบัน
- บรรทัดที่ 4-6 เขียนข้อมูลลงฐานข้อมูล Cloud Firestore โดยระบุ path ที่จะทำการจัดเก็บชุดข้อมูล
- บรรทัดที่ 28-35 อัพเดทข้อมูลเวลาสนทนาล่าสุดของผู้ใช้
- บรรทัดที่ 33 แสดงสถานะการอัพเดทข้อมูล

4.1.6 โครงสร้างของการสร้างหน้าปฏิทินแสดงกำหนดการ

```
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
           " type="date" size="large" placeholder="
           select date" style="margin-top:6px;"></
           DatePicker>
8      </Card>
9     </Col>
10    <Col span="18">
11     <h3>Schedule</h3>
12     <Table border
13       :loading="dataLoading"
14       :columns="columnsName"
15       :data="eventsData"
16       no-data-text="no schedule"
17       style="margin-top:6px;">
18     </Table>
19    </Col>
20   </Row>
21  </div>
22 </template>
```

รูปที่ 4.12: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าปฏิทินกำหนดการ Schedule.vue

จากภาพที่ 4.12 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าปฏิทินกำหนดการ สามารถอธิบายการทำงานได้ดังนี้

- 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
          ().time.getMonth()}-${doc.data().time.
          getFullYear()}‘
14    }
15    vm.eventsData.push(data)
16   })
17  })
18 }
```

รูปที่ 4.13: การสร้างลอจิกของหน้าปฏิทินกำหนดการ Schedule.vue

จากภาพที่ 4.13 โครงสร้างลอจิกของหน้าปฏิทินกำหนดการ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-18 ชื่อฟังก์ชันที่จะถูกเรียกทุกครั้งที่หน้าปฏิทินกำหนดการถูกเปิด
- บรรทัดที่ 3-17 สืบค้นกำหนดการจากฐานข้อมูลโดยมีการเรียงลำดับจากวันที่ล่าสุดไปยังวันที่ก่อนหน้า
- บรรทัดที่ 9-15 จัดเก็บข้อมูลที่สืบค้นได้ เพื่อใช้ในการแสดงผลบนหน้าจอ

4.1.7 โครงสร้างของการสร้างหน้าสร้างประกาศ

```
<Modal v-model="modalNewPost"
title="add post">
 <Form :model="formItem" :label-width="80">
  <FormItem label="title">
   <Input v-model="formItem.title" placeholder="
      Enter something..."></Input>
  </FormItem>
  <FormItem label="detail">
   <Input v-model="formItem.description" type="
      textarea" :autosize="{minRows: 2,maxRows: 5}"
      placeholder="Enter something..."></Input>
  </FormItem>
  <FormItem label="target">
   <Select v-model="formItem.collection">
    <Option value="public">public</Option>
    <Option value="group">group</Option>
    <Option value="volunteer">volunteer</Option>
   </Select>
  </FormItem>
  <FormItem label="contact list" v-if="tags.length >
      0">
    <Tag closable color="blue" v-for="tag in tags" :
       key="tag" @on-close="handleClose"> {{ tag }}
       </Tag>
  </FormItem>
  <FormItem label="file">
   <Upload :before-upload="handleUpload"
    action="https://shielded-earth-61349.herokuapp.
       com/">
    <Button :type="btnAddPostType" icon="ios-cloud-
       upload-outline"> {{ uploadBtnTile }} </Button>
   </Upload>
  </FormItem>
 </Form>
 <div slot="footer">
  <Button type="primary" :loading="loading" @click="
     newPost">save</Button>
  <Button type="ghost" style="margin-left: 8px"
     @click="modalNewPost = !modalNewPost">cancle</
     Button>
 </div>
</Modal>
```

รูปที่ 4.14: การสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ MgPost.vue

จากภาพที่ 4.8 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

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

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

รูปที่ 4.15: การสร้างลอจิกของหน้าหน้าสร้างประกาศ MgPost.vue

จากภาพที่ 4.15 โครงสร้างลอจิกของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

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

– บรรทัดที่ 13-19 เป็นการอัพเดทข้อมูล URL ที่ได้จากการอัพโหลดไฟล์แนบเข้าสู่ฐานข้อมูล Cloud Firestore

4.1.8 โครงสร้างของการสร้างหน้าอัพโหลดเอกสารที่เกี่ยวข้อง

```
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"> {{ uploadBtnTile }} </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>
```

รูป ที่ 4.16: การ สร้าง หน้า จอ ส่วน ติดต่อ ผู้ ใช้ ของ หน้า อัพโหลด เอกสาร ที่ เกี่ยวข้อง MgDocument.vue

จากภาพที่ 4.16 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าอัพโหลดเอกสารที่เกี่ยวข้อง สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-22 สร้างป๊อปอัพแสดงหน้าอัพโหลดไฟล์เอกสาร
- 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.
            lastIndexOf('.'))
16     }).then(() => {
17      vm.$Notice.success({
18        title: 'Success',
19       desc: ''
20     });
21   }).catch(function(error) {
22    vm.$Notice.warning({
23      title: 'Fail',
24      desc:''
25    });
26  });
27 }
```

รูปที่ 4.17: การสร้างลอจิกของหน้าหน้าอัพโหลดเอกสารที่เกี่ยวข้อง MgDocument.vue

จากภาพที่ 4.17 โครงสร้างลอจิกของหน้าสร้างประกาศ สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 6-26 เป็นการอัพเดทข้อมูลเอกสารเข้าสู่ฐานข้อมูล Cloud Firestore

- บรรทัดที่ 10-12 เป็นการเรียกใช้ไฟร์เบส Storage เพื่อทำการอัพโหลดไฟล์เอกสาร
- บรรทัดที่ 13-20 ใช้ในการอัพเดทข้อมูล URL ไปยังฐานข้อมูล Cloude Firestore

4.1.9 โครงสร้างของการสร้างหน้าสร้างกำหนดการจองคิวส่งเอกสาร

```
<Modal v-model="modalNewQueue">
 <p slot="header" style="color:#3498db;text-align:
    center">
  <Icon type="information-circled"></Icon>
  <span>Create sumbit document date</span>
 </p>
 <div>
  <Form :model="formItem" :label-width="80">
   <FormItem label="title">
    <Input v-model="formItem.title"></Input>
   </FormItem>
   <FormItem label="date">
    <DatePicker v-model="formItem.date" format="d-
       MMMM-yyyy" type="daterange" placement="bottom-
       end" placeholder="select" style="width: 200px
       "></DatePicker>
   </FormItem>
   <FormItem label="time">
    <TimePicker v-model="formItem.time" format="HH:mm
       " type="timerange" placement="bottom-end"
       placeholder="select" style="width: 200px"></
       TimePicker>
   </FormItem>
   <FormItem label="students per hr">
    <InputNumber :max="100" :min="1" v-model="
       formItem.count" style="width: 200px"></
       InputNumber>
   </FormItem>
  </Form>
 </div>
 <div slot="footer">
   <Button type="success" size="large" long @click="
      saveQueue">save</Button>
 </div>
</Modal>
```

รูปที่ 4.18: การ สร้าง หน้า จอ ส่วน ติดต่อ ผู้ใช้ ของ หน้า สร้าง กำหนดการ จอง คิว ส่ง เอกสาร MgQueue.vue

จากภาพที่ 4.18 โครงสร้างของการสร้างหน้าจอส่วนติดต่อผู้ใช้ของหน้าสร้างกำหนดการจองคิวส่งเอกสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 1-25 การสร้างหน้าต่างป๊อปอัพ
- บรรทัดที่ 4 ชื่อหน้าต่างป๊อปอัพ
- 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: การสร้างลอจิกของหน้าสร้างกำหนดการจองคิวส่งเอกสาร MgQueue.vue

จากภาพที่ 4.19 โครงสร้างลอจิกของหน้าสร้างกำหนดการจองคิวส่งเอกสาร สามารถอธิบายการทำงานได้ดังนี้

- บรรทัดที่ 3-15 ที่ได้จากการอัพโหลดไฟล์แนบเข้าสู่ฐานข้อมูล Cloud Firestore

4.2 การพัฒนาเว็บแอปพลิเคชันแอนดรอยด์แอปพลิเคชัน

การพัฒนาเว็บแอปพลิเคชันแอนดรอยด์แอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีวัตถุประสงค์หลักเพื่ออำนวยความสะดวกต่อนักศึกษาที่โดยส่วนใหญ่ใช้อุปกรณ์พกพาและต้องการใช้งานฮาร์ดแวร์(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 sheduleFrag =
     ScheduleFragment.newInstance();
6 private SubmitFragment submitFrag = SubmitFragment.
     newInstance();
7 private UserChatFragment userChatFrag =
     UserChatFragment.newInstance();
8 private CheckinFragment checkinFrag =
     CheckinFragment.newInstance();
```

รูปที่ 4.20: ตัวแปรในคลาส MainActivity

จากภาพที่ 4.20 ตัวแปรที่ประกาศขึ้นเพื่อใช้ในการทำงานของคลาส MainActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร mAuth ใช้ในการจัดเก็บสถานะและข้อมูลของผู้ใช้
- บรรทัดที่ 2 ตัวแปร feedFrag ใช้แสดงผลหน้าจอข่าวสาร
- บรรทัดที่ 3 ตัวแปร chatFrag ใช้แสดงผลหน้าจอสนทนาสำหรับเจ้าหน้าที่
- บรรทัดที่ 4 ตัวแปร chatFrag ใช้แสดงผลหน้าจอเอกสารที่เกี่ยวข้อง
- บรรทัดที่ 5 ตัวแปร scheduleFrag ใช้แสดงผลหน้าจอปฏิทินกำหนดการ
- บรรทัดที่ 6 ตัวแปร submitFrag ใช้แสดงผลหน้าจอส่งสำเนาเอกสาร
- บรรทัดที่ 7 ตัวแปร userChatFrag ใช้แสดงผลหน้าจอสนทนาสำหรับนักศึกษา
- บรรทัดที่ 8 ตัวแปร checkinFrag ใช้แสดงผลหน้าจอจองคิวส่งเอกสาร

```
new DrawerBuilder()
.addDrawerItems(
  feed, chat, event, doc, submit, checkin, faq,
     about, setting, account, logout
).withOnDrawerItemClickListener(new Drawer.
   OnDrawerItemClickListener() {
@Override
 public boolean onItemClick(View view, int position
    , IDrawerItem drawerItem) {
   long id = drawerItem.getIdentifier();
  if (id == 1) {
    getSupportFragmentManager().beginTransaction()
    .replace(R.id.contentContainer, feedFrag)
    .commit();
  } else if (id == 2) {
   if (currentUser.getEmail().contains("tapgabee"))
      {
     getSupportFragmentManager().beginTransaction()
     .replace(R.id.contentContainer, chatFrag)
     .commit();
   } else {
      getSupportFragmentManager().beginTransaction()
      .replace(R.id.contentContainer, userChatFrag)
      .commit();
   }
  } else if (id == 3) {
     getSupportFragmentManager().beginTransaction()
     .replace(R.id.contentContainer, sheduleFrag)
     .commit();
  } else if (id == 4) {
     getSupportFragmentManager().beginTransaction()
     .replace(R.id.contentContainer, docFrag)
     .commit();
  }else if (id == 5) {
     getSupportFragmentManager().beginTransaction()
     .replace(R.id.contentContainer, submitFrag)
     .commit();
  } else if (id == 6) {
     getSupportFragmentManager().beginTransaction()
     .replace(R.id.contentContainer, checkinFrag)
     .commit();
  } else if (id == 11) {
     mAuth.signOut();
     startActivity(new Intent(MainActivity.this,
        MainActivity.class));
     finish();
  }
}
```

รูปที่ 4.21: โค๊ดส่วนที่ใช้ในการสร้างเมนูนำทางหลักภายในคลาส MainActivity

จากภาพที่ 4.21 สามารถอธิบายการทำงานโค๊ดส่วนที่ใช้ในการสร้างเมนูนำทางหลักภายในคลาส MainActivity ได้ดังนี้

- บรรทัดที่ 1 เป็นการสร้างเมนูนำทาง
- บรรทัดที่ 2-3 เป็นการเพิ่ม Fragment ต่างๆ เข้าไปยังเมนูนำทาง
- บรรทัดที่ 4-6 เป็นการเพิ่มการดักจับอีเวนต์ (Event) เพื่อสลับหน้าจอการแสดงผลที่เกิดขึ้นเมื่อผู้ใช้กดที่เมนูนำทาง
- บรรทัดที่ 8-11 เป็นการแสดงผลหน้าข่าวสาร
- บรรทัดที่ 13-16 เป็นการแสดงผลหน้าสนทนาสำหรับเจ้าหน้าที่
- บรรทัดที่ 17-20 เป็นการแสดงผลหน้าสนทนาสำหรับนักศึกษา
- 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

```
db.collection("Posts")
.orderBy(getString(R.string.key_time), Query.
   Direction.DESCENDING)
.get()
.addOnCompleteListener(new OnCompleteListener<
   QuerySnapshot>() {
 @Override
 public void onComplete(@NonNull Task<QuerySnapshot
    > task) {
  if (task.isSuccessful() && isAdded()) {
   for (DocumentSnapshot document : task.getResult
      ()) {
    Log.d(TAG, document.getId() + " => " +
       document.getData());
    Map<String, Object> data = document.getData();
    Post post = new Post();
    post.setTitle(data.get(getString(R.string.
       key_title)).toString());
    post.setCollection(data.get(getString(R.string
       .key_collection)).toString());
    post.setDate((Date)data.get(getString(R.string
       .key_time)));
    post.setDescription(data.get(getString(R.
       string.key_description)) == null ? "" :
       data.get(getString(R.string.key_description
       )).toString());
    post.setFileURL(data.get(getString(R.string.
       key_fileURL)) == null ? "" : data.get(
       getString(R.string.key_fileURL)).toString())
       ;
    post.setFileName(data.get(getString(R.string.
       key_fileName)) == null ? "" : data.get(
       getString(R.string.key_fileName)).toString()
       );
    posts.add(post);
   }
  recyclerView.setLayoutManager(new
     LinearLayoutManager(getActivity()));
  recyclerView.setAdapter(adapter);
  adapter.notifyDataSetChanged();
 } else {
   Log.w(TAG, "Error getting documents.", task.
      getException());
 }
}
});
```

รูป ที่ 4.23: โค๊ด ส่วน ที่ ใช้ ใน การ สืบค้น ข้อมูล จาก Cloude Firestore ภายใน คลาส FeedFragment

จากภาพที่ 4.23 โค๊ดส่วนที่ใช้ในการสืบค้นข้อมูลจาก Cloude Firestore ภายในคลาส FeedFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-3 เริ่มทำการสืบค้นข้อมูลประกาศทั้งหมดพร้อมทั้งเรียงลำดับข้อมูลจากประกาศล่าสุดก่อน
- บรรทัดที่ 4-18 รับผลการสืบค้นพร้อมทั้งเพิ่มข้อมูลที่ได้แต่ละแถวเข้าไว้ที่ตัวแปร posts
- บรรทัดที่ 20-22 ทำการอัพเดทข้อมูลที่แสดงอยู่บน recyclerView

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

รูปที่ 4.24: โค๊ดส่วนที่ใช้ในการดักอีเวนต์เมื่อผู้ใช้กดที่แถวประกาศในคลาส FeedFragment

จากภาพที่ 4.24โค๊ดส่วนที่ใช้ในการดักอีเวนต์เมื่อผู้ใช้กดที่แถวประกาศในคลาส FeedFragment สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ประกาศตัวแปร intent ประเภทตัวแปร Intent เพื่อใช้กำหนดแอคทิวิตี้ปลายทางซึ่งในที่นี้คือ PostDetailActivity
- บรรทัดที่ 4-9 เป็นการเพิ่มข้อมูลเข้าเก็บไว้ที่ตัวแปร intent โดยดึงข้อมูลใน posts มาจากตำแหน่งแถวที่ถูกผู้ใช้กด

- บรรทัดที่ 11 เริ่มการทำงานของแอคทิวิตี้ PostDetailActivity

4.2.3 โครงสร้างของการสร้างหน้า PostDetailActivity

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

รูปที่ 4.25: โค๊ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส PostDetailActivity

จากภาพที่ 4.25 โค๊ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส PostDetailActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-4 เป็นการประกาศตัวแปรที่ใช้ในการเก็บข้อมูลประกาศ
- บรรทัดที่ 6-11 เป็นการดึงค่าที่ถูกส่งมาจากคลาส 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)
           getSystemService(Context.DOWNLOAD_SERVICE);
6      Uri uri = Uri.parse(strFileURl);
7      DownloadManager.Request request = new
           DownloadManager.Request(uri);
8      request.setNotificationVisibility(DownloadManager.
           Request.VISIBILITY_VISIBLE_NOTIFY_COMPLETED);
9      request.setDestinationInExternalPublicDir(
           Environment.DIRECTORY_DOWNLOADS, strFileName);
10    downloadManager.enqueue(request);
11   }
12 }
```

รูปที่ 4.26: โค๊ดส่วนที่ใช้ในการดาวน์โหลดเอกสารของคลาส PostDetailActivity

จากภาพที่ 4.26 โค๊ดส่วนที่ใช้ในการดาวน์โหลดเอกสารของคลาส 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: โค๊ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส ChatActivity

จากภาพที่ 4.27 โค๊ดส่วนที่ใช้ในการแสดงผลรายละเอียดประกาศของคลาส ChatActivity สามารถอธิบายได้ดังนี้

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

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

รูปที่ 4.28: โค๊ดส่วนที่ใช้ในการสืบค้นประวัติการสนทนาของคลาส ChatActivity

จากภาพที่ 4.28 โค๊ดส่วนที่ใช้ในการสืบค้นประวัติการสนทนาของคลาส ChatActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1-2 ทำการสืบค้นข้อมูลประวัติการสนทนาจาก Cloud Firestore
- บรรทัดที่ 11-14 สร้างตัวแปรเพื่อใช้ในการจัดเก็บข้อมูล

– บรรทัดที่ 16-24 ทำการอ่านค่าและเพิ่มเข้าลิสต์รายการ

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

รูปที่ 4.29: โค๊ดส่วนที่ใช้ในการส่งข้อความของคลาส ChatActivity

จากภาพที่ 4.29 โค๊ดส่วนที่ใช้ในการส่งข้อความของคลาส ChatActivity สามารถอธิบายได้ดังนี้

– บรรทัดที่ 1-6 สร้าง HashMap เพื่อใช้จัดเก็บข้อความ

– บรรทัดที่ 8-11 เป็นการเพิ่มชุดข้อมูลเข้าสู่ Cloud Firestore

– บรรทัดที่ 14-18 เป็นการอัพเดทเวลาสนทนาล่าสุดของผู้ใช้

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: โค๊ดส่วนที่ใช้ในการแสดงผลหน้าเข้าสู่ระบบของคลาส SignInActivity

จากภาพที่ 4.30 โค๊ดส่วนที่ใช้ในการแสดงผลหน้าเข้าสู่ระบบของคลาส 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.
      isEmpty() || password == null ){
4   Toast.makeText(SignInActivity.this, "Please fill
       data!", Toast.LENGTH_SHORT).show();
5   return;
6  }
7  simpleProgressBar.setVisibility(View.VISIBLE);
8  mAuth.signInWithEmailAndPassword(email, password)
9  .addOnCompleteListener(this, new OnCompleteListener
      <AuthResult>() {
10  @Override
11  public void onComplete(@NonNull Task<AuthResult>
       task) {
12   if (task.isSuccessful()) {
13    startActivity(new Intent(SignInActivity.this,
        MainActivity.class));
14    finish();
15   } else {
16    Toast.makeText(SignInActivity.this, "
        Authentication failed.",
17    Toast.LENGTH_SHORT).show();
18   }
19   simpleProgressBar.setVisibility(View.INVISIBLE);
20  }
21 }
22 );
```

รูปที่ 4.31: โค๊ดส่วนที่ใช้ในการเข้าสู่ระบบของคลาส SignInActivity

จากภาพที่ 4.31 โค๊ดส่วนที่ใช้ในการเข้าสู่ระบบของคลาส SignInActivity สามารถอธิบายได้ดังนี้

- บรรทัดที่ 1 ตัวแปร email จัดเก็บอีเมลของผู้ใช้
- บรรทัดที่ 2 ตัวแปร password จัดเก็บรหัสผ่านของผู้ใช้
- บรรทัดที่ 3 ตรวจสอบความถูกต้องของข้อมูล
- บรรทัดที่ 8-21 เป็นการเรียกใช้ไฟร์เบส 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: โค๊ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment

จากภาพที่ 4.34 โค๊ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment สามารถอธิบายได้ดังนี้

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

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

รูปที่ 4.33: โค๊ดส่วนที่ใช้ในการสืบค้นข้อมูลกำหนดการของคลาส ScheduleFragment

จากภาพที่ 4.33 โค๊ดส่วนที่ใช้ในการสืบค้นข้อมูลกำหนดการของคลาส 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: โค๊ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส ScheduleFragment

จากภาพที่ 4.34 โค๊ดส่วนที่ใช้ในการแสดงผลหน้าปฏิทินของคลาส 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

การทดสอบการทำงานขอแอนดรอยด์งแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานีและทดสอบการทำงานในส่วนของเว็บไซต์ โดยทำการทดสอบในลักษณะ Black-box Testing [12] หรือ Data-Driven testing ซึ่งเป็นการเทสแบบที่ไม่สนใจโปรเซส (Process) การทำงานภายในของโปรแกรมว่าทำงานอย่างไร แต่จะเน้นไปที่ Input และ Result ที่ได้มากกว่าว่าการทำงานต่าง ๆ ถูกต้องตามความต้องการ (Requirement) หรือไม่ ซึ่งการทดสอบการใช้งานแอนดรอยด์แอปพลิเคชัน และ การใช้งานเว็บแอปพลิเคชัน ได้ผลดังนี้

5.1 การทดสอบการใช้งานแอนดรอยด์แอปพลิเคชัน

- การทดสอบการใช้งานเมนูนำทางของแอนดรอยด์แอปพลิเคชัน การทดสอบเมนูนำทางของแอปพลิเคชันในการนำทางผู้ใช้งาน ซึ่งเมนูหลักประกอบด้วย เมนูหน้าประกาศ เมนูหน้าสนทนา เมนูหน้าปฏิทินกำหนดการ เมนูหน้าดาวน์โหลดเอกสาร เมนูส่งภาพถ่ายสำเนาเอก-สาร เมนูหน้าจองคิวส่งเอกสาร เมนูหน้าคำถามทีพบบ่อย เมนูหน้าเกี่ยวกับ เมนูหน้าข้อมูลส่วนตัวและเมนูออกจากระบบ ผลทดสอบดังตารางที่ 5.9-5.2

ตารางที่ 5.1: ผลการทดสอบเมนูนำทาง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เมนูประกาศ	กดปุ่มเมนูประกาศ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการประกาศทั้งหมด
เมนูสนทนา	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนา พร้อม ทั้ง แสดง ข้อมูล ประวัติการสนทนา
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงผลหน้าจอสนทนา พร้อม ทั้ง แสดง ข้อมูล ประวัติการสนทนา
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลดเอกสาร	ระบบ แสดง ผล หน้า จอ รายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนู หน้า ส่ง ภาพถ่าย สำเนาเอกสาร	กดปุ่มเมนูหน้าส่งภาพสำเนาเอกสาร	ระบบแสดงผลหน้าส่งเอกสารภาพสำเนาเอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

ตารางที่ 5.2: ผลการทดสอบเมนูนำทาง(ต่อ)

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เมนูหน้าจองคิวส่งเอกสาร	กดปุ่มเมนูหน้าจองคิวส่งเอกสาร	ระบบแสดงผลหน้าจองคิวส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ข่าวสาร พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าคำถามทีพบบ่อย	กด ปุ่ม เมนู หน้า คำถาม ที พบบ่อย	ระบบ แสดง หน้า จอง คิว ส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าเกี่ยวกับ	กดปุ่มเมนูหน้าเกี่ยวกับ	ระบบ แสดง ผล หน้า เกี่ยว กับ ซึ่งแสดงข้อมูลผู้พัฒนาวรมไปถึง แสดง เครดิต (credit) ไลบรารี ต่าง ๆ ที่ใช้งาน ภายในแอปพลิเคชัน
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าบัญชีผู้ใช้	กดปุ่มเมนูหน้าบัญชีผู้ใช้	ระบบ แสดง ผล หน้า จอ ข้อมูลส่วนตัวโดยมีข้อมูล รูปประจำตัว ชื่อผู้ใช้ สาขาวิชาและภาควิชา
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูออกจากระบบ	กดปุ่มเมนูออกจากระบบ	ทำการ ออก จาก ระบบ และแสดงหน้าจอข่าวสาร

- การทดสอบหน้ารายละเอียดประกาศ ในการแสดงผลหน้าจอรายละเอียดประกาศนั้นจะประกอบไปด้วยหัวเรื่องประกาศ รายละเอียดประกาศ วันที่ประกาศและเอกสารแนบ ผลการทดสอบดังตารางที่ 5.10

ตารางที่ 5.3: ผลการทดสอบหน้ารายละเอียดประกาศ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูประกาศ	ระบบ แสดง ผล หน้า จอ ประ-กาศ พร้อม ทั้ง แสดง รายการประกาศทั้งหมด
	กด ปุ่ม อ่าน ราย ละเอียด ประ-กาศ	ระบบแสดงผลหน้าจอรายละ-เอียดประกาศ
	กด ปุ่ม ดาวน์โหลด เอกสาร แนบ	ระบบแสดงผลการดาวน์โหลด เอกสารแนบ
	เมื่อ ดาวน์โหลด เสร็จ กด ปุ่ม เปิดเอกสาร	ระบบแสดงผลเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอรายละ-เอียดประกาศ
	กดปุ่มย้อนกลับอีกครั้ง	ระบบ แสดง ผล หน้า จอ ประ-กาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าสนทนา ในการแสดงผลหน้าจอสนทนานั้นจะประกอบไปด้วยรายการประวิติการสนทนา ช่องกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.11

ตารางที่ 5.4: ผลการการทดสอบหน้าสนทนา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาพร้อม ทั้ง แสดง รายการ ประวัติการสนทนา
	กดปุ่มที่ช่องกรอกข้อความ	ระบบ แสดง ตัว กระ พริบ (cursor) เพื่อ ชี้ ให้ รู้ ว่า ตำแหน่งของการพิมพ์อักขระ
	พิมพ์อักขระ	ระบบ แส ดง ผลอั ข ระ ที่ ถูก พิมพ์
	กดปุ่มส่งข้อความ	ระบบ แสดง ข้อความ ที่ ถูก พิมพ์ บน รายการ ประวัติสนทนาล่าสุด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าปฏิทินกำหนดการ ในการแสดงผลหน้าปฏิทินกำหนดการนั้นจะประกอบไปด้วยรายการประวิติการสนทนา ช่องกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.12

ตารางที่ 5.5: ผลการการทดสอบหน้าปฏิทินกำหนดการ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงหน้าจอปฏิทินกำหนดการ โดย มี การ แสดง กำหนดการของวันปัจจุบัน
	กดเลือกวันที่ต้องการดูกำหนดการในปฏิทิน	ระบบ แสดง กำหนดการ ของวันที่ถูกเลือก
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าดาวน์โหลดเอกสาร ในการแสดงผลหน้าจอดาวน์โหลดเอกสารนั้นจะประกอบไปด้วยรายการเอกสารโดยที่แต่ละฉบับบจะแสดงชื่อเอกสารและปุ่มดาวน์โหลดเอกสาร ผลการทดสอบดังตารางที่ 5.13

ตารางที่ 5.6: ผลการการทดสอบหน้าดาวน์โหลดเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลดเอกสาร	ระบบ แสดง ผล หน้า จอ รายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มดาวน์โหลดเอกสาร	ระบบแสดงผลการดาวน์โหลดเอกสาร
	เมื่อ ดาวน์โหลด เสร็จ กด ปุ่ม เปิดเอกสาร	ระบบแสดงผลเอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ รายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร ในการแสดงผลหน้าส่งภาพถ่ายสำเนาเอกสารนั้นจะประกอบไปด้วยปุ่มเพิ่มเอกสารฉบับที่ 1 ปุ่มเพิ่มเอกสารฉบับที่ 2 และปุ่มส่งเอกสาร ผลการทดสอบดังตารางที่ 5.7

ตารางที่ 5.7: ผลการทดสอบหน้าส่งภาพถ่ายสำเนาเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้า ส่ง ภาพถ่าย สำเนา เอก-สาร	กดปุ่มเมนูหน้าจองคิวส่งเอก-สาร	ระบบ แสดง หน้า จอง คิว ส่ง เอกสาร
	กดปุ่มเพิ่มเอกสารฉบับที่ 1	ระบบแสดงหน้าจอกล้องถ่าย ภาพ
	กดปุ่มถ่ายภาพเอกสาร	ระบบแสดงผลภาพเอกสาร
	กดปุ่มถัดไป	ระบบ แสดง ผล หน้า ปรับ แต่ง ภาพเอกสาร
	กดปุ่มยืนยัน	ระบบ แสดง ผล ภาพ หน้า ส่ง ภาพถ่าย สำเนา เอกสาร พร้อม ทั้ง แสดง ผล ภาพ เอกสาร ฉบับ ที่ 1
	กดปุ่มเพิ่มเอกสารฉบับที่ 2	ระบบแสดงหน้าจอกล้องถ่าย ภาพ
	กดปุ่มถ่ายภาพเอกสาร	ระบบแสดงผลภาพเอกสาร
	กดปุ่มถัดไป	ระบบ แสดง ผล หน้า ปรับ แต่ง ภาพเอกสาร
	กดปุ่มยืนยัน	ระบบ แสดง ผล ภาพ หน้า ส่ง ภาพถ่าย สำเนา เอกสาร พร้อม ทั้ง แสดง ผล ภาพ เอกสาร ฉบับ ที่ 1 และฉบับที่ 2
	กดส่งเอกสาร	ระบบแสดงผลการส่งเอกสาร และ แสดง สถานะ การ ตรวจ เอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประ-กาศ พร้อม ทั้ง แสดง รายการ ข่าวสารทั้งหมด

- การทดสอบหน้าจองคิวส่งเอกสาร ในการแสดงผลหน้าจองคิวส่งเอกสารนั้นจะประกอบไปด้วยปุ่มกดเลือกวันที่ ปุ่มกดเลือเวลา ผลการทดสอบดังตารางที่ 5.8

ตารางที่ 5.8: ผลการทดสอบหน้าจองคิวส่งเอกสาร

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าจองคิวส่งเอกสาร	กดปุ่มเมนูหน้าจองคิวส่งเอกสาร	ระบบแสดงผลหน้าจอกำหนดการส่งเอกสาร
	กดปุ่มเลือกวันที่ต้องการส่งเอกสาร	ระบบแสดงผลวันที่ถูกเลือก
	กดปุ่มเลือกเวลาที่ต้องการส่งเอกสาร	ระบบแสดงผลเวลาที่ถูกเลือกพร้อมทั้งแสดงปุ่มกดบันทึก
	กดปุ่มบันทึก	ระบบแสดงผลการจองวันที่ส่งเอกสาร
	กดปุ่มย้อนกลับ	ระบบแสดงผลหน้าจอประกาศพร้อมทั้งแสดงรายการข่าวสารทั้งหมด

5.2 การทดสอบการใช้งานเว็บแอปพลิเคชัน

- การทดสอบการใช้งานเมนูนำทางของเว็บแอปพลิเคชัน การทดสอบเมนูนำทางของเว็บแอปพลิเคชันในการนำทางผู้ใช้งาน ซึ่งเมนูหลักประกอบด้วย เมนูหน้าประกาศ เมนูหน้าสนทนา เมนูหน้าปฏิทินกำหนดการ เมนูหน้าดาวน์โหลดเอกสาร เมนูหน้าคำถามทีพบบ่อย เมนูหน้าเกี่ยวกับและเมนูออกจากระบบ ผลทดสอบดังตารางที่ 5.9-5.2

ตารางที่ 5.9: ผลการทดสอบเมนูนำทาง

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
เมนูประกาศ	กดปุ่มเมนูประกาศ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการประกาศทั้งหมด
เมนูสนทนา	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนา พร้อม ทั้ง แสดง ข้อมูล ประวัติการสนทนา
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบแสดงผลหน้าจอสนทนา พร้อม ทั้ง แสดง ข้อมูล ประวัติการสนทนา
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าดาวน์โหลดเอกสาร	กด ปุ่ม เมนู หน้า ดาวน์โหลด เอกสาร	ระบบ แสดง ผล หน้า จอ ตาราง รายการ เอกสาร ใน ระบบ พร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าคำถามทีพบบ่อย	กด ปุ่ม เมนู หน้า คำถาม ที พบ บ่อย	ระบบ แสดง หน้า จอง คิว ส่ง เอกสาร
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูหน้าเกี่ยวกับ	กดปุ่มเมนูหน้าเกี่ยวกับ	ระบบ แสดง ผล หน้า เกี่ยว กับ ซึ่งแสดงข้อมูลผู้พัฒนาวรมไปถึง แสดง เครดิต (credit) ไลบรารี ต่าง ๆ ที่ใช้ งาน ภายใน แอปพลิเคชัน
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด
เมนูออกจากระบบ	กดปุ่มเมนูออกจากระบบ	ทำการ ออก จาก ระบบ และ แสดงหน้าจอข่าวสาร

- การทดสอบหน้ารายละเอียดประกาศ ในการแสดงผลหน้าจอรายละเอียดประกาศนั้นจะประกอบไปด้วยหัวเรื่องประกาศ รายละเอียดประกาศ วันที่ประกาศและเอกสารแนบ ผลการทดสอบดังตารางที่ 5.10

ตารางที่ 5.10: ผลการทดสอบหน้ารายละเอียดประกาศ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูประกาศ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการประกาศทั้งหมด
	กด ปุ่ม อ่าน รายละเอียด ประกาศ	ระบบแสดงผลหน้าจอรายละเอียดประกาศ
	กด ปุ่ม ดาวน์โหลด เอกสารแนบ	ระบบแสดงผลการดาวน์โหลดเอกสารแนบ
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าสนทนา ในการแสดงผลหน้าจอสนทนานั้นจะประกอบไปด้วยรายการประวิติการสนทนา ช่องกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.11

ตารางที่ 5.11: ผลการการทดสอบหน้าสนทนา

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้ารายละเอียดประกาศ	กดปุ่มเมนูสนทนา	ระบบแสดงผลหน้าจอสนทนาพร้อม ทั้ง แสดง รายการ ประวัติการสนทนา
	กดปุ่มที่ช่องกรอกข้อความ	ระบบ แสดง ตัว กระ พริบ (cursor) เพื่อ ชี้ ให้ รู้ ว่า ตำแหน่งของการพิมพ์อักขระ
	พิมพ์อักขระ	ระบบ แส ดง ผลอั ข ระ ที่ ถูก พิมพ์
	กดปุ่มส่งข้อความ	ระบบ แสดง ข้อความ ที่ ถูก พิมพ์ บน รายการ ประวัติสนทนาล่าสุด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

- การทดสอบหน้าปฏิทินกำหนดการ ในการแสดงผลหน้าปฏิทินกำหนดการนั้นจะประกอบไปด้วยรายการประวิติการสนทนา ช่องกรอกข้อความและปุ่มส่งข้อความ ผลการทดสอบดังตารางที่ 5.12

ตารางที่ 5.12: ผลการการทดสอบหน้าปฏิทินกำหนดการ

การทำงาน	เงื่อนไขการทดสอบ	ผลการทดสอบ
หน้าปฏิทินกำหนดการ	กดปุ่มเมนูปฏิทินกำหนดการ	ระบบ แสดง ตาราง กำ หน ดาร ทั้งหมด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประ-กาศ พร้อม ทั้ง แสดง รายการ ข่าวสารทั้งหมด

- การทดสอบหน้าดาวน์โหลดเอกสาร ในการแสดงผลหน้าจอดาวน์โหลดเอกสารนั้นจะประกอบไปด้วยรายการเอกสารโดยที่แต่ละฉบับบจะแสดงชื่อเอกสารและปุ่มดาวน์โหลดเอกสาร 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 รายการเอกสารในระบบพร้อมทั้งแสดงปุ่มดาวน์โหลด
	กดปุ่มย้อนกลับ	ระบบ แสดง ผล หน้า จอ ประกาศ พร้อม ทั้ง แสดง รายการข่าวสารทั้งหมด

บทที่ 6

สรุปและข้อเสนอแนะ

การดำเนินโครงงานเพื่อพัฒนาระบบ XX นี้ พบว่าระบบสามารถทำงานได้ตามที่วิเคราะห์และออกแบบไว้ แต่ก็พบปัญหาและอุปสรรคระหว่างการพัฒนา ในบทนี้ผู้พัฒนาจึงขอสรุปความสามารถของระบบ ชี้แจงปัญหาและอุปสรรค พร้อมเสนอแนวทางในการพัฒนาระบบ XX ต่อ ตามลำดับ

6.1 สรุปความสามารถของระบบ

ระบบ XX ทั้งเว็บแอปพลิเคชันและโมบายแอปพลิเคชันสามารถสรุปความสามารถที่ระบบทำได้ดังนี้

6.1.1 เว็บแอปพลิเคชัน

ความสามารถหลักของเว็บแอปพลิเคชันนั้นเน้นสร้างความสะดวกต่อการจัดการเอกสารเรื่องข้อมูลต่างๆ ที่เกี่ยวข้องกับ XX โดยแบ่งความสามารถของระบบตามประเภทของผู้ใช้งานดังนี้

1. เจ้าหน้าที่
    - สร้างและแก้ไข XX ได้
    - สร้างกำหนดการการดำเนินงานของ XX ได้
    - สนทนากับ XX ได้
    - อัพโหลดเอกสารที่เกี่ยวข้องเข้าสู่ฐานข้อมูลได้
    - 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
    - อัพโหลดเอกสารที่เกี่ยวข้องเข้าสู่ฐานข้อมูลได้
2. นักศึกษา
    - ดูประกาศได้
    - ดูปฏิทินกำหนดการได้
    - ดูกำหนดการส่งเอกสารได้
    - ดาวน์โหลดเอกสารที่เกี่ยวข้องของ XX ได้
    - สนทนากับ XX ได้
    - ส่งภาพถ่ายสำเนาเอกสารได้
    - รับแจ้งเตือนต่างๆ ได้

6.2 ปัญหาและอุปสรรคในการพัฒนา

1. ไลบรารี (Library) ที่ใช้ในการกรอกข้อมูลลงเอกสาร pdf ไม่รองรับภาษาไทย
   แนวทางการแก้ไข : เปลี่ยนขั้นตอนการทำงานเป็นถ่ายภาพเอกสารฉบับจริงแล้วทำการส่งให้เจ้าหน้าที่ตรวจสอบเพื่อยืนยันความถูกต้อง
2. เนื่องจากทางผู้พัฒนามีความประสงค์ให้ระบบนี้สามารถใช้งานได้จริง ดังนั้น การพัฒนาในตอนนี้ยังมีข้อจำกัดเรื่องขนาดของเอกสารที่จัดเก็บบนไฟร์เบสที่สามารถอัพโหลดเข้าสู่ระบบสูงสุดเพียง 5 GB ซึ่งหากระบบถูกใช้งานจริงจำนวนข้อมูลในระบบจะเกินจำนวนที่ไฟร์เบสให้ใช้งานฟรี
   แนวทางการแก้ไข : ทำการบีบอัดข้อมูลให้มีขนาดเล็กลง ส่วนในอนาคตอาจจำเป็นต้องศึกษาแนวทางการสร้างเซิฟเวอร์ (Server) เป็นของระบบเอง

6.3 แนวทางการพัฒนาต่อ

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

บรรณานุกรม

[1] ทศพล ต้นสมบัติ. (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 สระอุบล. 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 ซึงถาวร. 2545. JAVA PROGRAMMING Volume I. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.

[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 เมษายา 2561. จาก https://www.studentloan.or.th/index.php/aboutus .

[9] งาน กองทุน เงิน ให้ กู้ ยืม เพื่อ การ ศึกษา มหาวิทยาลัย สงขลา นครินทร์ วิทยาเขต หาดใหญ่. (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

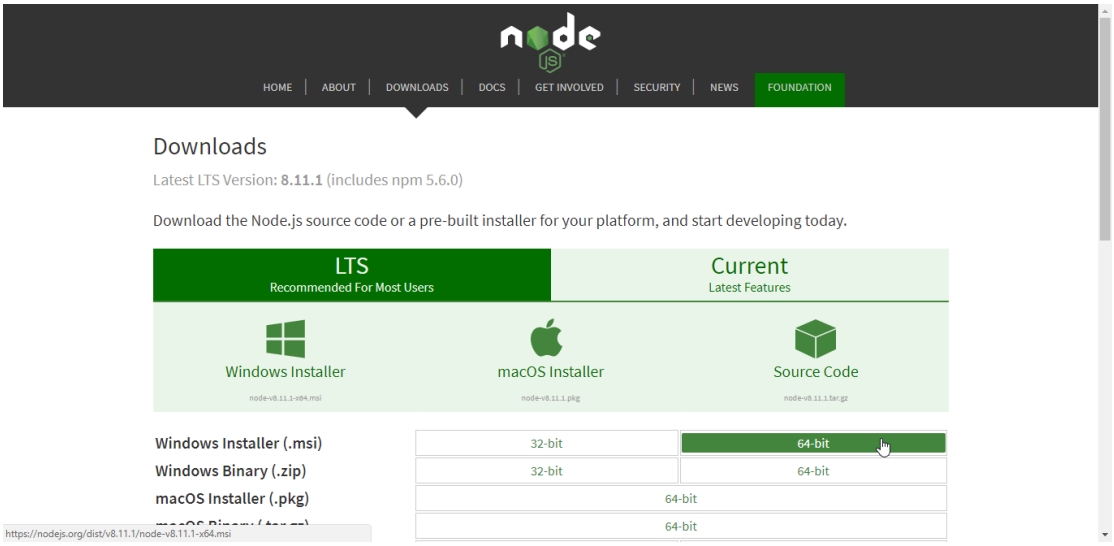
การติดตั้งเครื่องมือที่ใช้พัฒนาโปรแกรม

การติดตั้งเครื่องมือที่ใช้ในการพัฒนาระบบสหกิจศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี มีโปรแกรมที่จำเป็นในการพัฒนาระบบดังต่อไปนี้

- การติดตั้ง Node.js
- การติดตั้ง React.js
- การติดตั้ง XAMPP

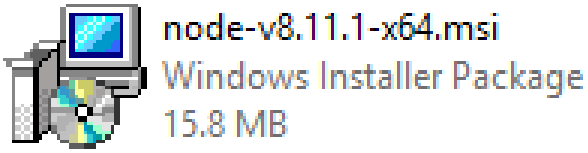
ก.1 การติดตั้ง Node.js

1) สามารถดาวน์โหลด Node.js ได้ที่ https://nodejs.org/en/download/ แสดงดังรูปที่ ก. 1



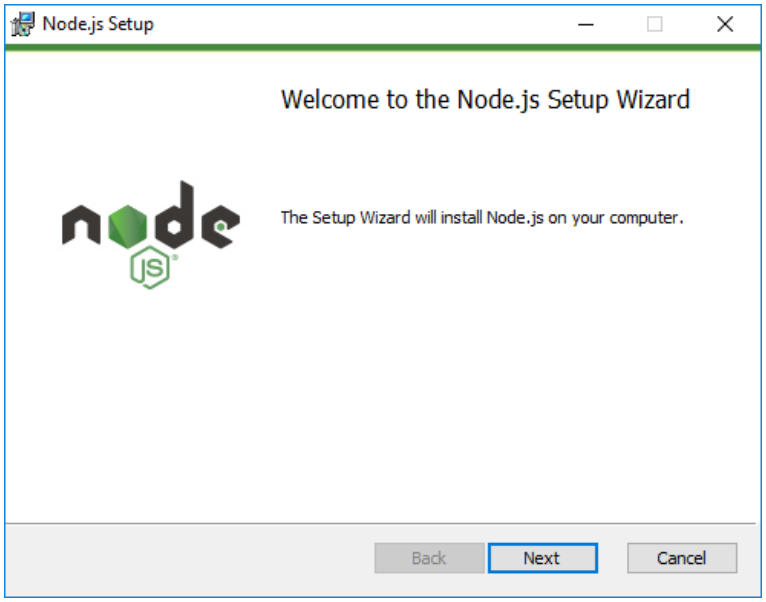
รูปที่ ก.1: หน้าเว็บดาวน์โหลด Node.js

2) เปิดไฟล์ติดตั้ง ชื่อ node-v10.15.3-x64.msi เพื่อติดตั้ง แสดงดังรูปที่ ก.2



รูปที่ ก.2: ไฟล์ติดตั้งสำหรับติดตั้ง Node.js

3) แสดงหน้าต่างตอนรับของ Node.js ให้กด Next แสดงดังรูปที่ ก.3

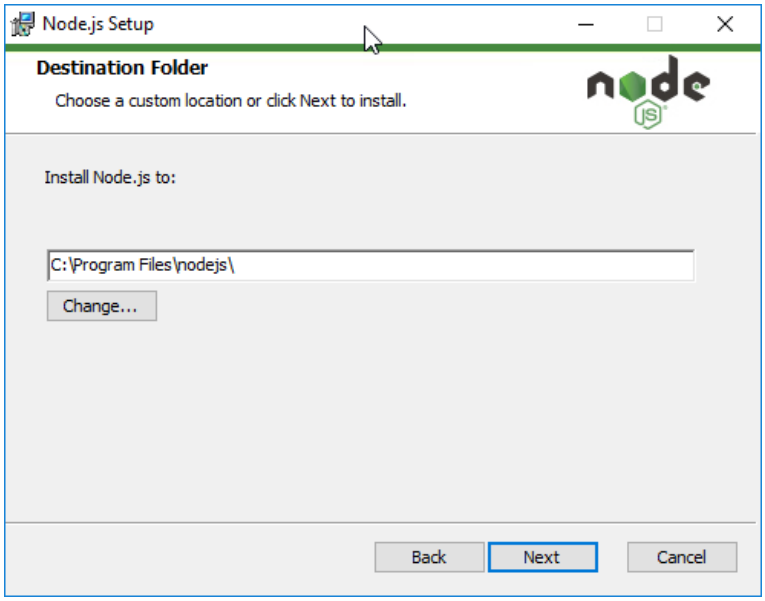


รูปที่ ก.3: หน้าต่างตอนรับของ Node.js

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

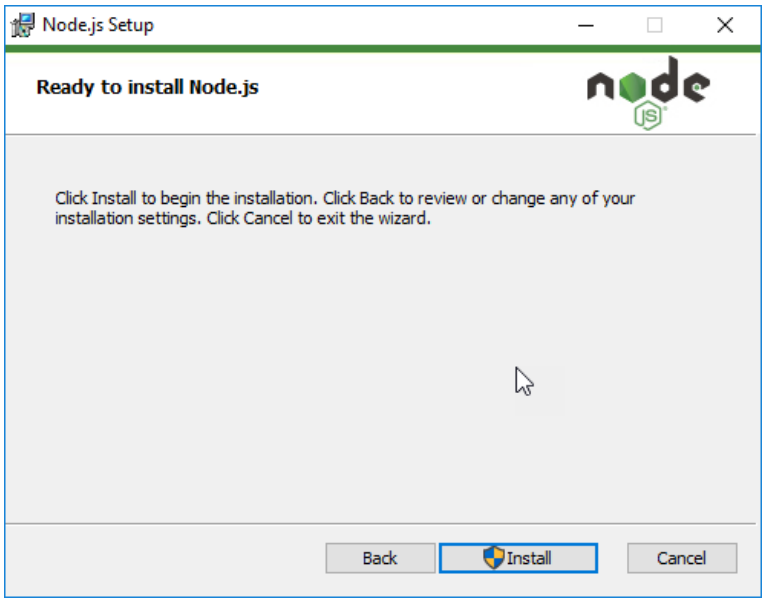
รูปที่ ก.4: หน้าต่างข้อตกลงในการใช้ Node.js

5) แสดงหน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง แสดงดังรูปที่ ก.5



รูปที่ ก.5: หน้าต่างเลือกโฟลเดอร์ที่จะทำการติดตั้ง Node.js

6) แสดงหน้าต่างสำหรับติดตั้ง Node.js ให้กด Install เพื่อทำงานติดตั้ง แสดงดังรูปที่ ก.6



รูปที่ ก.6: หน้าต่างติดตั้ง Node.js

ก.2 การติดตั้ง React.js

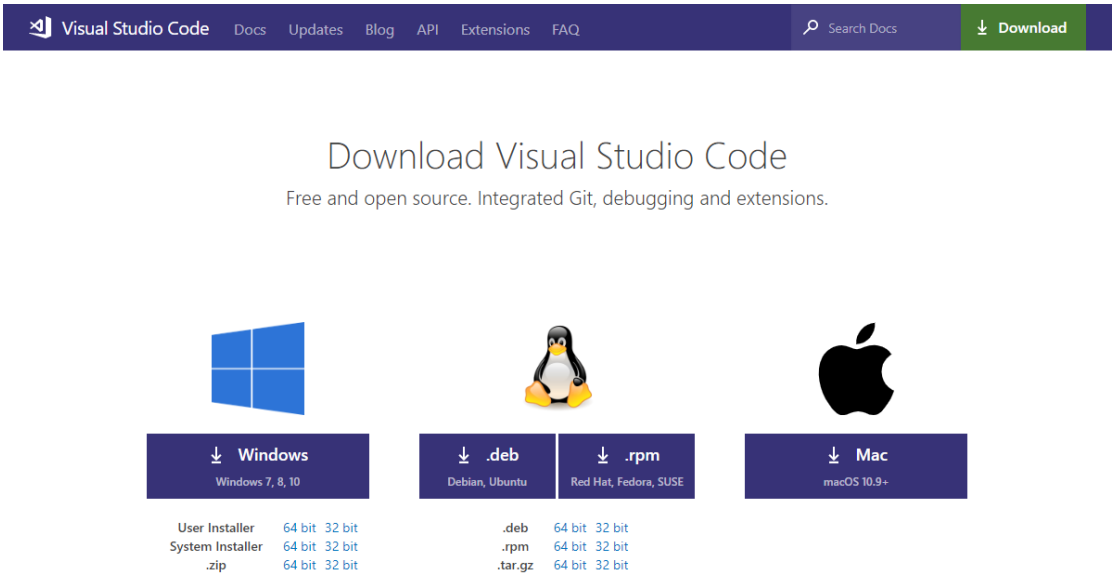
การติดตั้ง React.js สามารถทำผ่านคำสั่ง command line ได้ แสดงดังรูปที่ ก.7

```
npm install create-react-app --save
```

รูปที่ ก.7: คำสั่งสำหรับติดตั้ง React.js

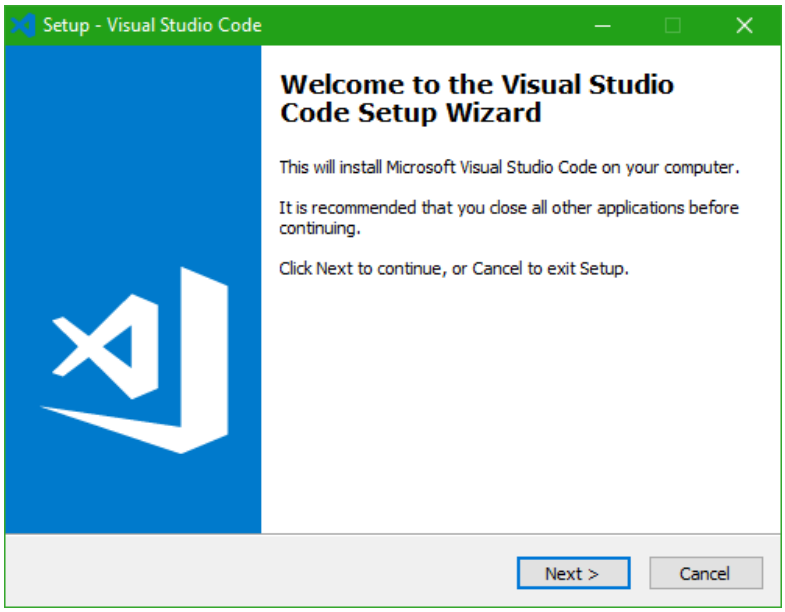
ก.3 การติดตั้ง Visual Studio Code

1. สามารถดาวน์โหลด Visual Studio Code ได้ที่ https://code.visualstudio.com/download ดังแสดงในรูปที่ ก.8



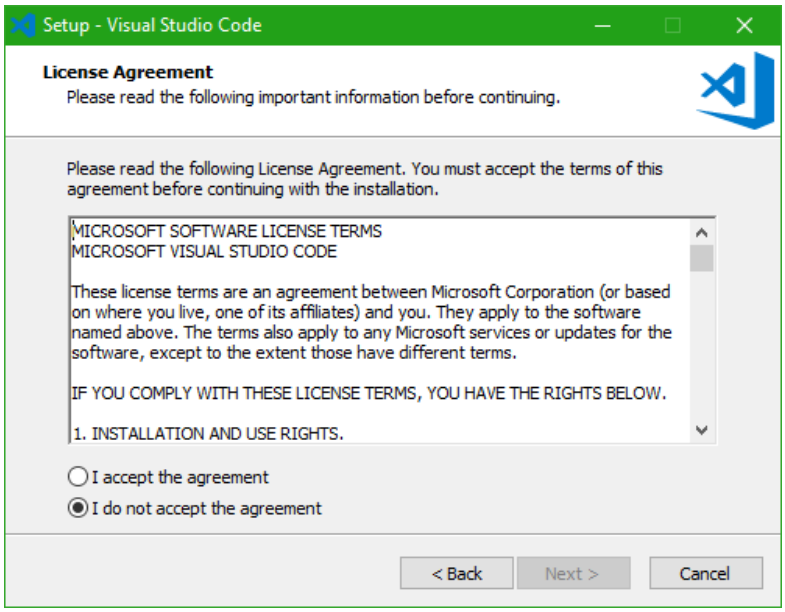
รูปที่ ก.8: หน้าเว็บดาวน์โหลด Visual Studio Code

2. เมื่อเปิดตัวติดตั้งขึ้นมาแล้ว จะแสดงหน้าจอ Welcome to the Visual Studio Code Setup Wizard ให้กดปุ่ม Next เพื่อเริ่มกระบวนการติดตั้ง ดังแสดงในรูปที่ ก.9



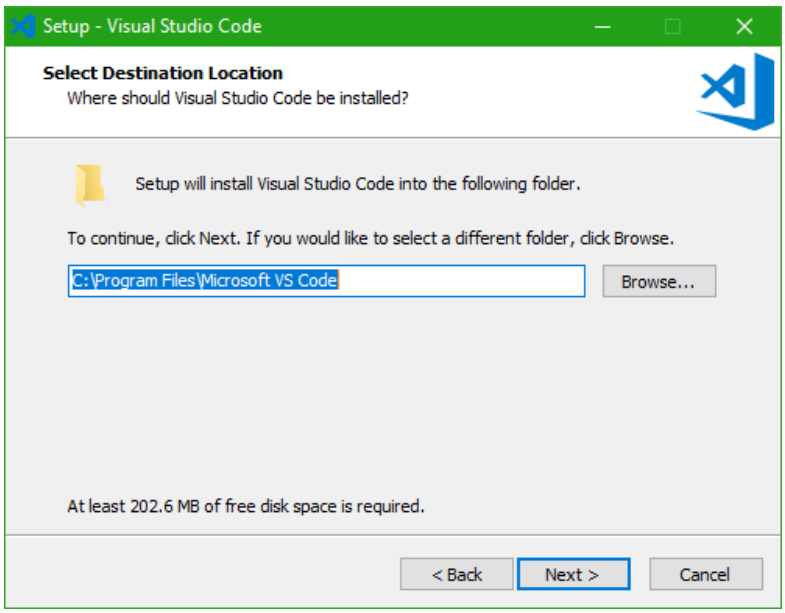
รูปที่ ก.9: หน้าต่างต้อนรับของ Visual Studio Code

3. หลังจากนั้นจะแสดงหน้าต่างข้อตกลงการใช้งาน Visual Studio Code ทำการติ๊กที่ I accept the areement แล้วกด Next ดังแสดงในรูปที่ ก.10



รูปที่ ก.10: หน้าต่างข้อตกลงการใช้งาน Visual Studio Code

4. จากนั้นจะแสดงหน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code ทำการกด Next ดังแสดงในรูปที่ ก.11

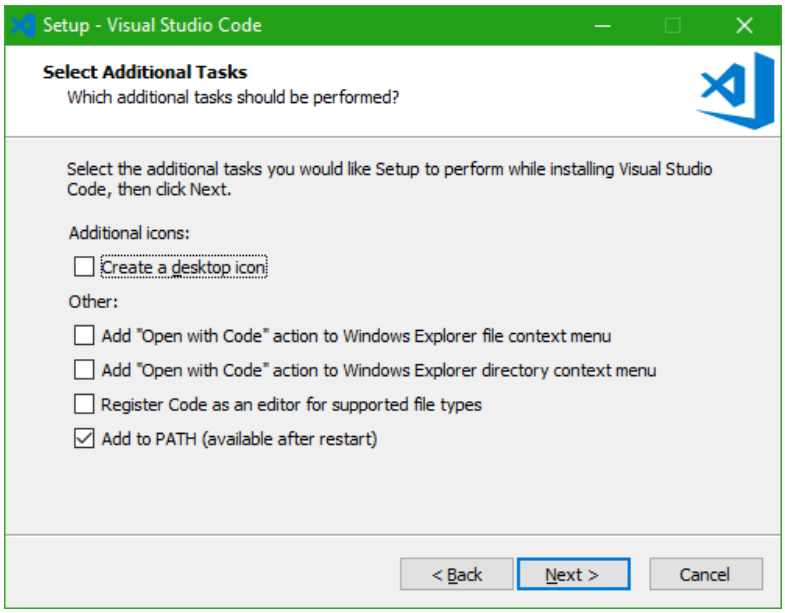


รูปที่ ก.11: หน้าต่างที่จัดเก็บไฟล์ต่างๆ ของ Visual Studio Code

5. จากนั้นจะแสดงหน้าต่างการจัดการซอร์ดคัทของ Visual Studio Code ทำการกด Next ดังแสดงในรูปที่ ก.12

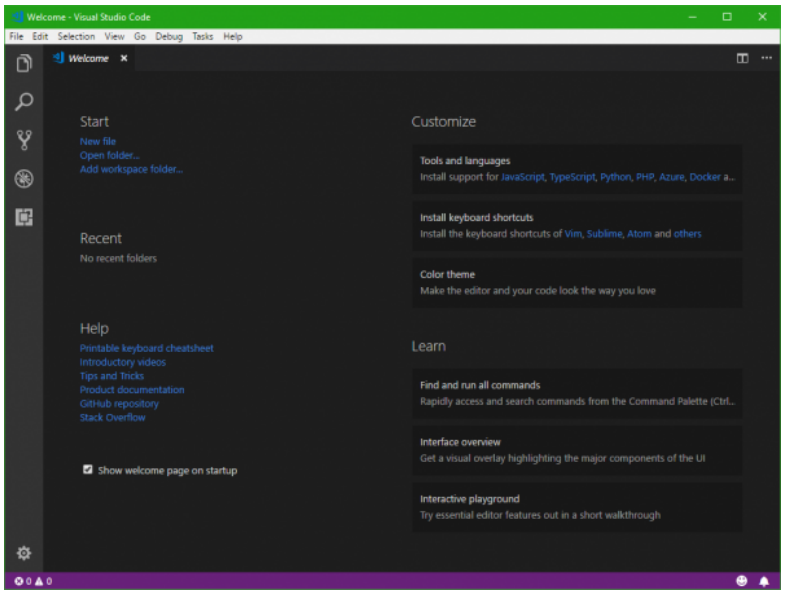
รูปที่ ก.12: หน้าต่างการจัดการซอร์ดคัท ของ Visual Studio Code

6. จากนั้นแสดงหน้าต่างเริ่มทำการติดตั้งทำการกด Next ดังแสดงในรูปที่ ก.13



รูปที่ ก.13: หน้าต่างเริ่มทำการติดตั้งทำการกด ของ Visual Studio Code

7. จากนั้นจะแสดงหน้าต่างเมื่อเข้าโปรแกรมหลังติตั้งเสร็จ ดังแสดงในรูปที่ ก.14



รูปที่ ก.14: หน้าต่างเมื่อเข้าโปรแกรมหลังติตั้งเสร็จ ของ Visual Studio Code

ภาคผนวก ข

คู่มือการติดตั้งระบบ

ในการติดตั้งเพื่อใช้งานแอปพลิเคชันระบบกองทุนเงินให้กู้ยืมเพื่อการศึกษา คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี หรือ ESP สามารถทำได้โดยมีขั้นตอนดังนี้

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

รูปที่ ข.1: หน้าเว็บดาวน์โหลด ESP installer package

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

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

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

รูปที่ ข.3: หน้าจอต้อนรับการติดตั้งแอปพลิเคชันบนอุปรกรณ์แอนดรอย์

4. เมื่อทำการติดตั้งแอปพลิเคชันสำเร็จระบบจะแสดงผล ดังแสดงในรูปที่ ข.4

รูปที่ ข.4: หน้าจอต้อนรับการติดตั้งแอปพลิเคชันบนอุปรกรณ์แอนดรอย์

ภาคผนวก ค

คู่มือการใช้งานระบบ

คู่มือการใช้งานทั้งหมดของระบบ สามารถแบ่งออกเป็น 2 ส่วน ดังนี้

1. ส่วนของหน้าเมนูแอปพลิเคชันสำหรับนักศึกษา
    - หน้าจอต้อนรับแสดงผลทุกครั้งเมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ ค.8

รูปที่ ค.1: หน้าจอต้อนรับ

    - เมื่อระบบทำการตรวจสอบว่ามีสิทธิ์(Permission)ในการใช้งานแอปพลิเคชันที่ผู้ใช้ยังไม่ได้อนุญาตให้เข้าถึง ระบบจะแสดงหน้าต่างขอสิทธิ์การเข้าถึง ดังแสดงในรูปที่ ค.9

รูปที่ ค.2: หน้าต่างขอสิทธิ์การเข้าถึง

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ หน้าต่างขอสิทธิ์การเข้าถึงข้อมูล
- หมายเลข 2 คือ ปุ่มให้สิทธิ์และยกเลิกการให้สิทธิ์

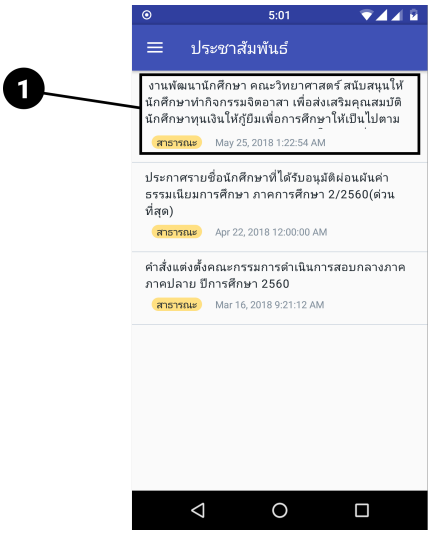
- ระบบทำการตรวจสอบทุกครั้งเมื่อผู้ใช้งานเปิดใช้งานแอปพลิเคชัน หากผู้ใช้งานคนปัจจุบันยังไม่ได้เข้าสู่ระบบ ระบบจะแสดงหน้าจอเข้าสู่ระบบโดยผู้ใช้งานจำเป็นต้องทำการกรอกข้อมูลคือ อีเมลและรหัสผ่าน ดังแสดงในรูปที่ ค.3

รูปที่ ค.3: หน้าจอเข้าสู่ระบบ

จากรูปที่ ค.3 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ส่วนของฟอร์มในการกรอกข้อมูลอีเมลและรหัสผ่าน
- หมายเลข 2 คือ ปุ่มกดเข้าสู่ระบบ

- หน้าแสดงข่าวสารประชาสัมพันธ์ซึ่งเป็นหน้าจอหลักของแอปพลิเคชัน ดังแสดงในรูปที่ ค.4



รูปที่ ค.4: หน้าแสดงข่าวสาร

จากรูปที่ ค.4 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ ข่าวสารที่มีข้อมูล หัวข้อข่าวสาร แท็ก(Tag)และวันที่ประการข่าวสาร

- เมื่อผู้ใช้กดเลือกดูรายละเอียดของข่าวสาร ระบบจะแสดงหน้าจอรายละเอียดข่าวสาร ดังแสดงในรูปที่ ค.5

รูปที่ ค.5: หน้ารายละเอียดของข่าวสาร

จากรูปที่ ค.5 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ หัวข้อข่าวสาร

– หมายเลข 1 คือ รายละเอียดของข่าวสาร

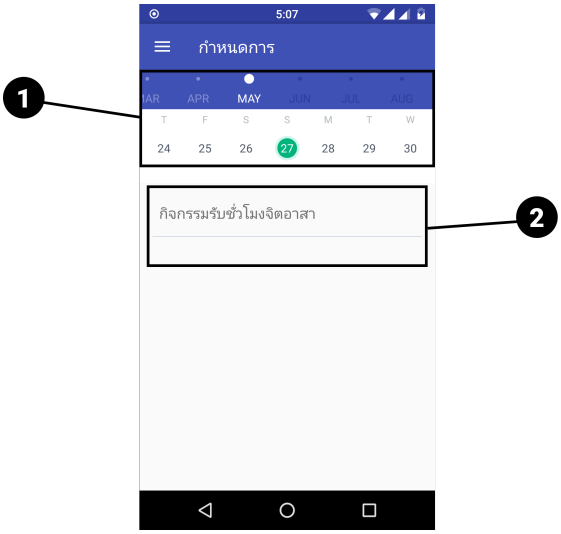
- เมื่อผู้ใช้กดเลือกเมนู Drawer ระบบจะแสดงเมนูนำทางหลักของแอปพลิเคชันซึ่งประกอบไปด้วยเมนู ประชาสัมพันธ์ ข้อความ กำหนดการ เอกสาร ส่งเอกสาร จองคิว คำถามที่พบบ่อย เกี่ยวกับเรา บัญชีผู้ใช้และออกจากระบบ ดังแสดงในรูปที่ ค.6

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
- หน้าจอต้อนรับแสดงผลทุกครั้งเมื่อผู้ใช้ทำการเปิดใช้งานแอปพลิเคชัน ดังแสดงในรูปที่ ค.8

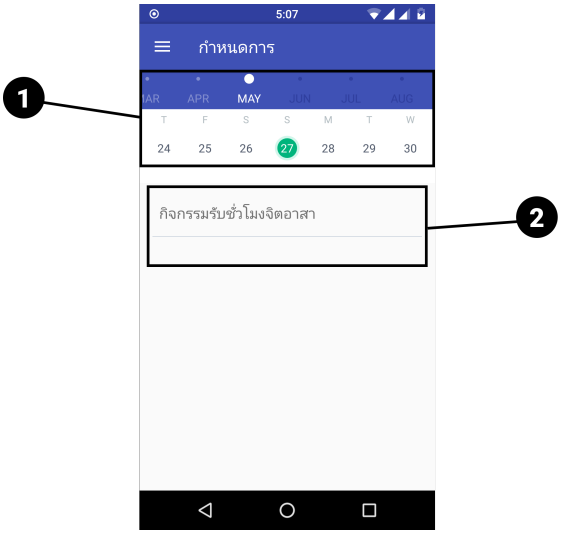
รูปที่ ค.8: หน้าจอต้อนรับ

- เมื่อระบบทำการตรวจสอบว่ามีสิทธิ์(Permission)ในการใช้งานแอปพลิเคชันที่ผู้ใช้ยังไม่ได้อนุญาตให้เข้าถึง ระบบจะแสดงหน้าต่างขอสิทธิ์การเข้าถึง ดังแสดงในรูปที่ ค.9

รูปที่ ค.9: หน้าต่างขอสิทธิ์การเข้าถึง

จากรูปที่ ค.9 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ หน้าต่างขอสิทธิ์การเข้าถึงข้อมูล
- หมายเลข 2 คือ ปุ่มให้สิทธิ์และยกเลิกการให้สิทธิ์



รูปที่ ค.10: หน้าแสดงกำหนดการ

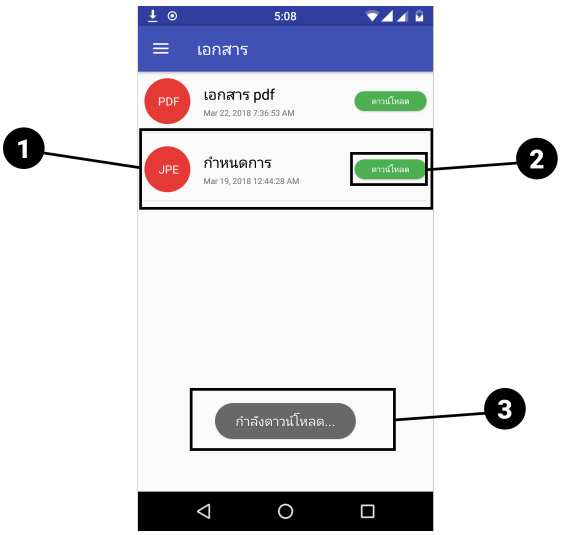
จากรูปที่ ค.10 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ ปฏิทินแสดงวันที่ปัจจุบันและผู้ใช้สามารถเลือกวันที่ต้องการ

เพื่อดูกำหนดการก่อนหน้า

- หมายเลข 2 คือ แสดงรายการกำหนดการของวันนั้น ๆ

- เมื่อผู้ใช้เลือกเมนูเอกสาร ระบบจะแสดงหน้าจอรายการเอกสาร ดังแสดงในรูปที่ ค. 11

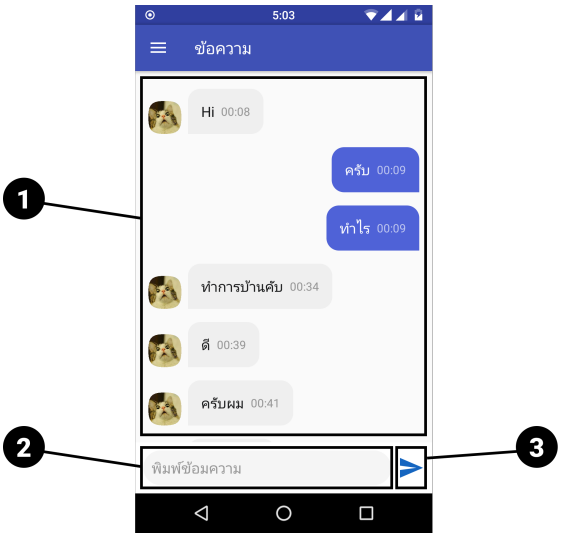


รูปที่ ค.11: หน้าจอเอกสาร

จากรูปที่ ค.11 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ รายการเอกสาร
- หมายเลข 2 คือ ปุ่มดาวน์โหลดเอกสาร
- หมายเลข 3 คือ สถานะการดาวน์โหลดเอกสาร

- เมื่อผู้ใช้เลือกเมนูข้อความ ระบบจะแสดงหน้าจอสนทนา ดังแสดงในรูปที่ ค.12

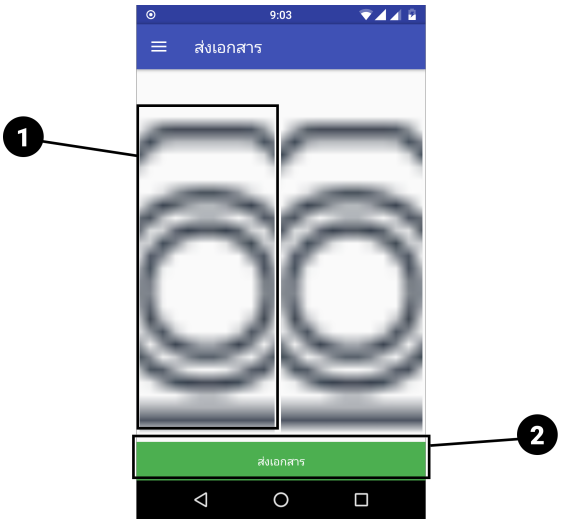


รูปที่ ค.12: หน้าจอสนทนา

จากรูปที่ ค.12 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ รายการประวัติการสนทนา
- หมายเลข 2 คือ ช่องกรอกข้อความเพื่อสนทนา
- หมายเลข 3 คือ ปุ่มส่งข้อความ

- เมื่อผู้ใช้เลือกเมนูส่งเอกสารระบบจะตรวจสอบข้อมูลว่าเจ้าหน้าที่ได้ทำการเปิดให้นักศึกษาส่งเอกสารได้หรือไม่ หากตรวจสอบแล้วพบว่าสามารถส่งได้ระบบจะแสดงหน้าจอส่งเอกสาร ดังแสดงในรูปที่ ค.13

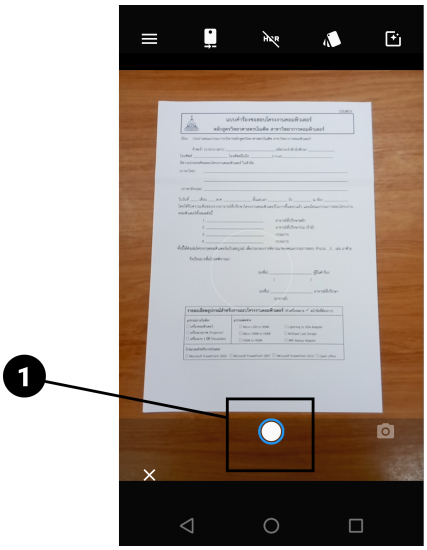


รูปที่ ค.13: หน้าจอส่งเอกสาร

จากรูปที่ ค.13 สามารถอธิบายการใช้งานได้ดังนี้

- หมายเลข 1 คือ เมื่อผู้ใช้เข้ามาครั้งแรกเมื่อผู้ใช้กดรูปภาพเอกสารระบบจะนำผู้ใช้ไปยังหน้าจอถ่ายภาพเอกสารและแสดงรูปภาพพรีวิว(Preview)ภาพถ่ายเอกสาร
- หมายเลข 2 คือ ปุ่มกดส่งเอกสาร

• เมื่อผู้ใช้กดที่ปุ่มถ่ายภาพเอกสารระบบจะแสดงหน้าจอถ่ายภาพเอกสาร ดังแสดงในรูปที่ ค.14

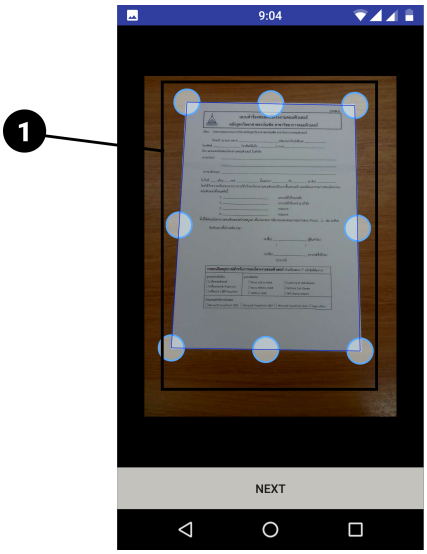


รูปที่ ค.14: หน้าจอถ่ายภาพเอกสาร

จากรูปที่ ค.14 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ ปุ่มกดถ่ายภาพสำเนาเอกสาร

• หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร ดังแสดงในรูปที่ ค.15

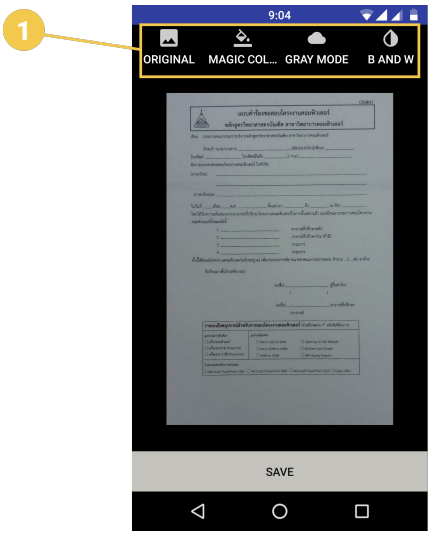


รูปที่ ค.15: หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.15 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ ปุ่มปรับมุมภาพทั้ง 8 มุมเพื่อปรับขนาดภาพถ่ายเอกสาร

- หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร ดังแสดงในรูปที่ ค.16

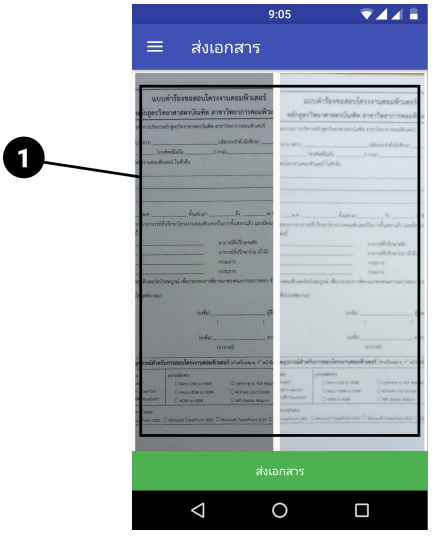


รูปที่ ค.16: หน้าจอแสดงปรับแต่งภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.16 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ เมื่อผู้ใช้ต้องการปรับความคมชัดของภาพสามารถทำได้จาก 4 ปุ่ม คือ ปุ่มภาพต้นฉบับ ปุ่มปรับสีอัตโนมัติ ปุ่มปรับปรับสีเป็นสีเทา ปุ่มปรับสีเป็นสีขาวดำ

- หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสารทั้งสองฉบับ ดังแสดงในรูปที่ ค.17

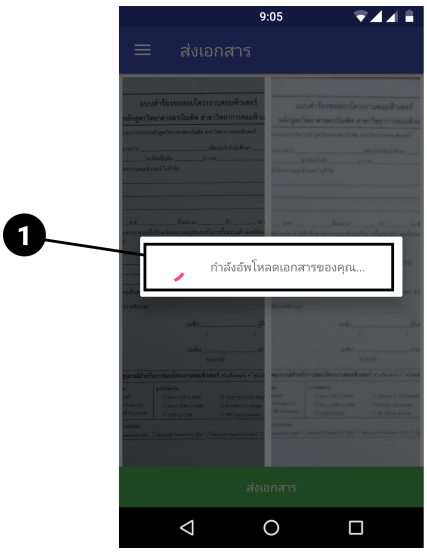


รูปที่ ค.17: หน้าจอแสดงภาพพรีวิวภาพถ่ายสำเนาเอกสาร

จากรูปที่ ค.17 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ แสดงภาพพรีวิวภาพถ่ายสำเนาเอกสารทั้งสองฉบับ

- เมื่อผู้ใช้กดปุ่มส่งเอกสาร ระบบจะแสดงหน้าต่างแสดงสถานะการอัพโหลดเอกสาร ดังแสดงในรูปที่ ค.18

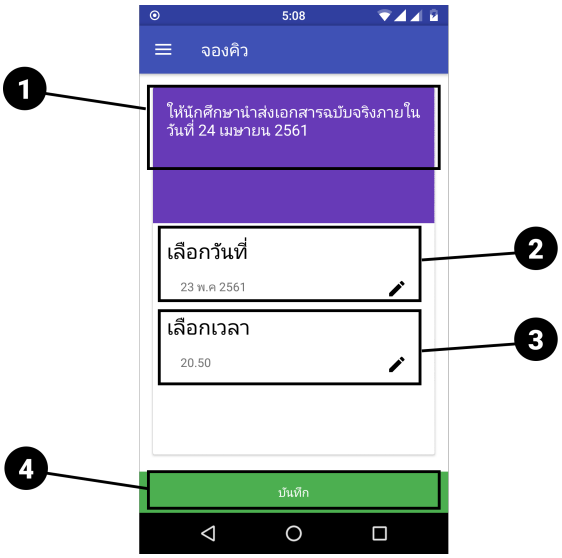


รูปที่ ค.18: หน้าต่างแสดงสถานะการอัพโหลดเอกสาร

จากรูปที่ ค.18 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ หน้าต่างสถานะการอัพโหลดเอกสาร

- เมื่อผู้ใช้กดเมนูจองคิว ระบบจะตรวจสอบถสานะการส่งเอกสารว่าถูกเจ้าหน้าที่ตรวจสอบแล้วหรือไม่ ถ้าเจ้าหน้าที่อนุมัติแล้วระบบจะแสดงหน้าจอจองวันที่ส่งเอกสาร ดังแสดงในรูปที่ ค.19

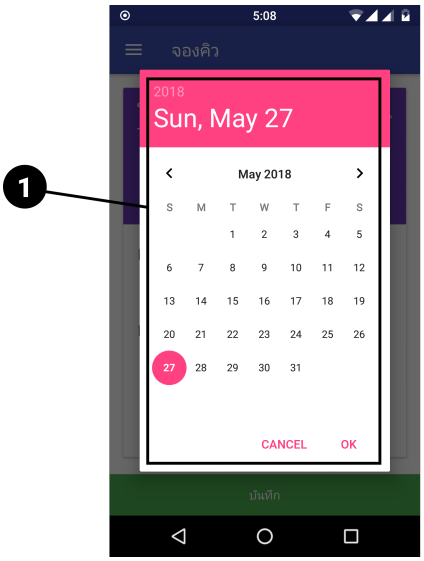


รูปที่ ค.19: หน้าจอจองคิว

จากรูปที่ ค.19 สามารถอธิบายการใช้งานได้ดังนี้

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

• เมื่อผู้ใช้กดปุ่มเลือกวันที่ที่ต้องการส่งเอกสารระบบจะแสดงหน้าต่างเลือกวันที่โดยจะแสดงเฉพาะวันที่ที่เจ้าหน้าที่เลือกไว้เท่านั้น ดังแสดงในรูปที่ ค.20

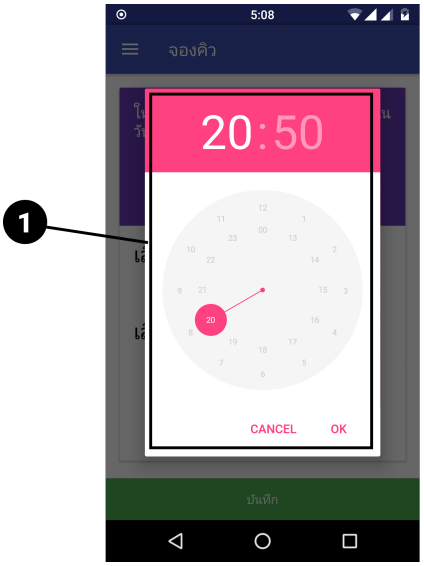


รูปที่ ค.20: หน้าต่างปฏิทินเลือกวันที่ต้องการส่งเอกสาร

จากรูปที่ ค.20 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ หน้าต่างปฏิทินเลือกวันที่ต้องการส่งเอกสาร

- เมื่อผู้ใช้กดปุ่มเลือกเวลาที่ต้องการส่งเอกสารระบบจะแสดงหน้าต่างเลือกเวลาโดยจะแสดงเฉพาะเวลาที่เจ้าหน้าที่เลือกไว้เท่านั้น ดังแสดงในรูปที่ ค.21



รูปที่ ค.21: หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร

จากรูปที่ ค.21 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ หน้าต่างนาฬิกาเลือกเวลาที่ต้องการส่งเอกสาร

- เมื่อผู้ใช้กดเมนูคำถามที่พบบ่อยระบบจะแสดงหน้าจอคำถามที่พบบ่อย ดังแสดงในรูปที่ ค.22

รูปที่ ค.22: หน้าจอคำถามที่พบบ่อย

จากรูปที่ ค.22 สามารถอธิบายการใช้งานได้ดังนี้

– หมายเลข 1 คือ คำถาม

– หมายเลข 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

ประวัติผู้พัฒนา

ชื่อ-สกุล: นางสาวปิยพร อาภรศรี

รหัสประจำตัวนักศึกษา: 59110440259

วัดเกิด: 12 06 2540

ที่อยู่ที่สามารถติดต่อได้: 7 ม.1 ต.หนองบก อ.เหล่าเสือโก้ก จ.อุบลราชธานี 34000

เบอร์โทรศัพท์: (+66) 99 468 2013

อิเมลล์: piyaphorn.ar.59@ubu.ac.th

ระดับมัธยมต้น: โรงเรียนหกสิบพรรษาวิทยาคมอุบลราชธานี จังหวัดอุบลราชธานี

ระดับมัธยมปลาย: โรงเรียนหกสิบพรรษาวิทยาคมอุบลราชธานี จังหวัดอุบลราชธานี

ระดับอุดมศึกษา: ภาควิชาคณิตศาสตร์ สถิติ และคอมพิวเตอร์ สาขาวิทยาการ คอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยอุบลราชธานี