

การพัฒนาฟีเจอร์การสนทนาสำหรับลูกค้าผู้ใช้บริการบน
แพลตฟอร์มคลาวด์คิทเชน

THE DEVELOPMENT OF CHAT FEATURE FOR
CONSUMERS ON CLOUD KITCHEN PLATFORM



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตร

บัณฑิต (วิทยาการคอมพิวเตอร์)

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน **ปีการศึกษา 2565** อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DEVELOPMENT OF CHAT FEATURE FOR CONSUMERS ON CLOUD KITCHEN PLATFORM





THITIRAT PHASOMSAP

A COOPERATIVE EDUCATION SUBMITTED IN PARTIAL
FULFILLMENT OF THE REQUIREMENT FOR
THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)
DEPARTMENT OF COMPUTER SCIENCE, SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพัฒนาฟีเจอร์การสนทนาสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเช่น THE DEVELOPMENT OF CHAT FEATURE FOR CUSTOMERS ON CLOUD KITCHEN PLATFORM
ชื่อนักศึกษา	นางสาวฐิติรัตน์ ผสมทรัพย์ รหัสนักศึกษา 62050147
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร.อินทราพร อรัณยธนา

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2565

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.อินทราพร อรัณยธนา กรรมการและอาจารย์ที่ปรึกษา	
ผศ.ดร.ปัทมา เจริญพร กรรมการ	

ลิขสิทธิของคณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพัฒนาฟีเจอร์การสนทนาสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเชน THE DEVELOPMENT OF CHAT FEATURE FOR CUSTOMERS ON CLOUD KITCHEN PLATFORM
ชื่อนักศึกษา	นางสาวฐิติรัตน์ ผสมทรัพย์ รหัสนักศึกษา 62050147
ปริญญา	วิทยาศาสตร์บัณฑิต (วิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร.อินทราพร อรัณยธนา

บทคัดย่อ

โครงการสหกิจศึกษานี้ มีวัตถุประสงค์เพื่อร่วมพัฒนาระบบการสนทนาบนแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ซึ่งแพลตฟอร์มคลาวด์คิทเชนเป็นแพลตฟอร์มที่อำนวยความสะดวกให้กับลูกค้าผู้ใช้บริการที่มีความต้องการสั่งอาหาร เนื่องจากมีความไม่สะดวกในการเดินทางไปรับประทานอาหารที่ร้าน และยังเป็นโอกาสให้ลูกค้าผู้ใช้บริการมีตัวเลือกในการเลือกร้านอาหารที่หลากหลายมากยิ่งขึ้น เพื่อเป็นการเปิดประสบการณ์ใหม่ ๆ ให้กับลูกค้าผู้ใช้บริการ โดยระบบการสนทนามีวัตถุประสงค์เพื่อเพิ่มทางเลือกช่องทางในการติดต่อสื่อสารระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้า และเพิ่มความถูกต้องของรายละเอียดในรายการอาหารที่ลูกค้าผู้ใช้บริการได้ทำการสั่งกับทางผู้ประกอบการร้านค้า ซึ่งจากการร่วมพัฒนา ผู้จัดทำได้ศึกษา ค้นคว้า พัฒนา และทดสอบการพัฒนาระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการที่ทำงานอยู่บนระบบปฏิบัติการ iOS บนแพลตฟอร์มคลาวด์คิทเชนและใช้ภาษา Swift ในการพัฒนา โดยระบบการสนทนานี้ ลูกค้าผู้ใช้บริการสามารถรับและส่งข้อความตัวอักษรและข้อความรูปภาพไปยังผู้ประกอบการร้านค้าได้

คำสำคัญ : Cloud Kitchen Platform, ระบบการสนทนา, ระบบปฏิบัติการ iOS, ภาษา Swift

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	THE DEVELOPMENT OF CHAT FEATURE FOR CUSTOMERS ON CLOUD KITCHEN PLATFORM
Students	Miss Thitirat Phasomsap Student ID 62050147
Department	Computer Science
School	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Asst. Prof. Inthraporn Aranyanak, Ph.D.

Abstract

The objective of this cooperative education project is to develop a chat feature for consumers on Cloud Kitchen Platform, a consumer-service platform that allows customers to easily order food from a variety of restaurants for a new experience. The chat feature will provide an additional communication channel between consumers and merchants and improve the accuracy of food item details for consumer purchases. The chat feature, which will be developed using Swift for the iOS operating system, will allow consumers to send and receive text and picture messages to the merchant.

Keywords: Cloud Kitchen Platform, Chat feature, iOS operating system, Swift language

กิตติกรรมประกาศ

การจัดทำสหกิจศึกษาครั้งนี้ บรรลุผลสำเร็จล่วงไปได้ด้วยดี โดยได้รับความกรุณา ความช่วยเหลือและการสนับสนุน จากบุคลากรหลายท่าน ผู้จัดทำขอขอบพระคุณบุคคลต่าง ๆ ที่ได้ให้ความช่วยเหลือและเสียสละเวลาในการให้คำแนะนำ คำปรึกษามาโดยตลอด อันได้แก่

ครอบครัวอันได้แก่ บิดา มารดา พี่สาวและพี่ชายที่คอยสนับสนุนในด้านการศึกษาและตลอดจนปัจจัยด้านอื่น ๆ ทั้งการดูแลร่างกาย การเป็นกำลังใจที่สำคัญ

ผศ.ดร.อินทราพร อรัณยธนาค และ ผศ.ดร.ปัทมา เจริญพร ที่ได้เสียสละเวลาให้คำแนะนำ ปรึกษา ให้ข้อเสนอแนะและวิธีการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นตลอดในการจัดทำสหกิจศึกษาฉบับนี้ อีกทั้งยังคอยติดตามความคืบหน้าของการดำเนินงานของสหกิจศึกษาครั้งนี้ และช่วยตรวจสอบแก้ไขให้สหกิจศึกษาบรรลุผลสำเร็จไปได้ด้วยดี

บริษัท ฟูลทิม จำกัด และพี่เลี้ยงในระหว่างที่ได้สหกิจศึกษาทุกท่านที่ได้มอบโอกาสและประสบการณ์อันดีในการเป็นนักศึกษาฝึกงานในตำแหน่งผู้พัฒนาแอปพลิเคชันบนระบบปฏิบัติการ iOS รวมทั้งให้ความรู้เกี่ยวกับการทำงานในองค์กร ชีวิตในการทำงาน ความรู้ทางด้านวิทยาการคอมพิวเตอร์ ตั้งแต่พื้นฐานของการเขียนแอปพลิเคชันจนถึงการทดสอบระบบ ให้คำแนะนำปรึกษา ให้กำลังใจ และคอยช่วยเหลือในทุก ๆ ด้านตลอดการสหกิจศึกษา เพื่อให้สามารถนำความรู้และประสบการณ์ไปใช้ได้จริง มีประสิทธิภาพ และทำให้ผลงานที่เกิดขึ้นสามารถปฏิบัติงานได้อย่างราบรื่น

คณะอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ และสาขาอื่น ๆ ที่ท่านได้อบรมสั่งสอน ได้ให้ความรู้ ความสามารถ ทั้งการทำงานและการใช้ชีวิตในอนาคต ตลอดระยะเวลา 4 ปีที่ได้ศึกษา ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สุดท้ายนี้ ยังมีบุคคลท่านอื่น ๆ ที่ไม่ได้กล่าวถึงไว้ ณ ที่นี้ ขอขอบพระคุณทุกท่าน ที่ได้สละเวลาส่วนตนมาเพื่อช่วยเหลือ ให้คำปรึกษา ให้การสนับสนุน และเป็นกำลังใจ ในการจัดทำสหกิจศึกษาฉบับนี้ให้สัมฤทธิ์ผลด้วยดีทุกประการ

ฐิติรัตน์ ผสมทรัพย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ	ง
สารบัญตาราง	ช
สารบัญรูป.....	ซ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของงานวิจัย	2
1.3 ขอบเขตของงานวิจัย	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง	3
2.1 ทฤษฎีที่เกี่ยวข้องกับกระบวนการดำเนินงาน.....	3
2.1.1 การพัฒนาระบบด้วยแนวคิดแบบ Agile	3
2.2 ทฤษฎีที่เกี่ยวข้องกับกระบวนการการพัฒนาและกรอบงานของระบบ.....	7
2.2.1 Clean Swift Architecture	7
2.2.2 View Controller Life cycle	8
2.2.3 Design Pattern	11
2.2.4 การเขียนโปรแกรมแบบโมดูลาร์.....	12
2.2.5 การแลกเปลี่ยนข้อมูลผ่าน API.....	13
2.2.6 Testing in Xcode with XCTest.....	15
2.2.7 การเขียน Unit Test	17
2.3 ทฤษฎีที่เกี่ยวข้องเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนา	18
2.3.1 Xcode	18
2.3.2 Cocoapods	19
2.3.3 macOS Terminal	19
2.3.4 Sourcetree	20
2.3.5 Figma	20
2.3.6 Postman	21

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่น

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.7 Firebase.....	21
2.3.8 WebSocket	22
2.3.9 Realm	22
2.4 ทฤษฎีที่เกี่ยวข้องกับภาษาโปรแกรมที่ใช้ในการพัฒนา	23
2.4.1 ภาษา Swift.....	23
2.5 งานวิจัยที่เกี่ยวข้อง.....	23
2.5.1 แนวคิด Conversational Commerce ในธุรกิจ E-Business.....	23
2.5.2 งานวิจัยที่เกี่ยวข้องกับการออกแบบการพัฒนาแอปพลิเคชันแชท	25
บทที่ 3 วิธีการดำเนินงานวิจัย.....	27
3.1 การวางแผนและการวิเคราะห์ระบบ	27
3.1.1 ศึกษาความเป็นไปได้และเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับการดำเนินงาน.....	27
3.1.2 วิเคราะห์ความต้องการของระบบ.....	27
3.1.3 การวิเคราะห์ระบบงานเดิม	28
3.1.4 การออกแบบ (Design).....	30
3.1.5 การดำเนินงาน.....	30
3.1.6 การทดสอบระบบ.....	30
3.2 ขั้นตอนการออกแบบระบบ	31
3.2.1 แผนภาพแสดงขั้นตอนการทำงานและส่วนต่าง ๆ ของระบบ.....	31
3.2.2 แผนภาพแสดงการทำงานของผู้ใช้ระบบ (Use Case Diagram)	34
3.2.3 แผนภาพแสดงถึงการปฏิสัมพันธ์ (Sequence Diagram).....	37
3.2.4 แผนภาพกิจกรรม (Activity Diagram).....	43
บทที่ 4 ผลการวิจัยและการอภิปรายผล	44
4.1 หน้าหลักของห้องสนทนา.....	45
4.1.1 หน้าหลักของห้องสนทนา.....	45
4.1.2 การส่งข้อความตัวอักษรและแสดงข้อความในหน้าห้องสนทนา	47
4.1.3 การส่งข้อความรูปภาพและแสดงข้อความในหน้าห้องสนทนา	51
4.1.4 แสดงสถานะการอ่านข้อความ	53
4.1.5 การแจ้งเตือนข้อความใหม่.....	54

เอกสารนี้เป็นเอกสาร 4.2 การทดสอบระบบการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกฏ 56

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	68
5.1 สรุปผลการวิจัย	68
5.2 ข้อเสนอแนะ.....	68
เอกสารอ้างอิง.....	70
ภาคผนวก	73
ภาคผนวก ก	74
ภาคผนวก ข	77



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
3.1 Use Case Description ของส่งข้อความตัวอักษร.....	35
3.2 Use Case Description ของส่งข้อความรูปภาพ.....	35
3.3 Use Case Description ของพรีวิรูรูปภาพ.....	35
3.4 Use Case Description ของดูประวัติการสนทนา.....	36
3.5 Use Case Description ของเช็คสถานะการอ่าน.....	36
3.6 Use Case Description ของแจ้งเตือนข้อความ.....	36
4.1 Test Case การดูประวัติการสนทนา.....	57
4.2 Test Case การกดส่งข้อความ.....	60
4.3 Test Case การรับข้อความ.....	61
4.4 Test Case การส่งข้อความสำเร็จ.....	62
4.5 Test Case การส่งข้อความไม่สำเร็จ.....	64
4.6 Test Case การส่งข้อความใหม่อีกครั้ง.....	65
4.7 Test Case การแสดงสถานะการอ่านข้อความ.....	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แนวคิดการทำงานรูปแบบ Agile	3
2.2 กระบวนการทำงานรูปแบบ Scrum	5
2.3 รูปภาพโครงสร้างของ Clean Swift	7
2.4 ตัวอย่างการเรียกใช้งานทั้ง 5 ฟังก์ชัน	8
2.5 ตัวอย่างการทำงานเมื่อเปิดหน้า View Controller ในครั้งแรก	9
2.6 ตัวอย่างการทำงานเมื่อทำการกดปุ่มเพื่อไปหน้าถัดไป	9
2.7 ตัวอย่างการทำงานเมื่อกลับกดปุ่มกลับมายังหน้าแรก	10
2.8 แผนภาพแสดงแนวคิดของ MVVM ใน iOS	11
2.9 แผนภาพการทำงานแบบโมดูลาร์	12
2.10 Test Pyramid	15
2.11 โปรแกรม Xcode	18
2.12 MacOS Terminal	19
2.13 กราฟแสดงการใช้งานแอปพลิเคชันแซท	24
2.14 ตารางเปรียบเทียบความแตกต่างของแต่ละแอปพลิเคชัน	24
2.15 Use Case Diagram ของการส่งข้อความ	26
2.16 Sequence Diagram ของการส่งข้อความ	26
3.1 แผนภาพแสดงขั้นตอนการทำงานของระบบ	31
3.2 Use Case Diagram	34
3.3 Sequence Diagram ของการดูประวัติการสนทนา	37
3.4 Sequence Diagram ของการส่งข้อความตัวอักษร	38
3.5 Sequence Diagram ของการส่งข้อความรูปภาพ	39
3.6 Sequence Diagram ของการส่งข้อความที่ส่งไม่สำเร็จ	40
3.7 Sequence Diagram ของการเช็คสถานะการอ่าน	41
3.8 Sequence Diagram ของการแจ้งเตือนข้อความ	42
3.9 Activity Diagram ของการดูประวัติการสนทนา	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.1 หน้าหลักของห้องสนทนา.....	45
4.2 หน้า Empty State	46
4.3 การส่งข้อความตัวอักษร.....	47
4.4 แสดงข้อความตัวอักษร	48
4.5 แสดงข้อความตัวอักษรที่ส่งไม่สำเร็จ	49
4.6 เมื่ออินเทอร์เน็ทกลับมาใช้งานได้ จะทำการส่งข้อความใหม่.....	50
4.7 การส่งข้อความรูปภาพ.....	51
4.8 แสดงพรีวิวรูปภาพ	52
4.9 แสดงสถานะการอ่านข้อความ	53
4.10 การแจ้งเตือนผ่าน Notification.....	54
4.11 แสดงแจ้งเตือนว่ามีข้อความที่ยังไม่ได้เปิดอ่าน	55
ก.1 หน้าผลการค้นหาโปรแกรม Xcode ใน App Store.....	74
ก.2 หน้า App Store แสดงปุ่ม Open เพื่อเปิด App.....	75
ก.3 หน้าจอแสดงข้อความขออนุญาตก่อนติดตั้งแอปพลิเคชัน	75
ก.4 หน้าต่างโปรแกรม Xcode.....	76
ข.1 ติดตั้ง Cocoapods ลงในเครื่อง.....	77
ข.2 สร้าง Podfile	78
ข.3 ไฟล์โปรเจคหลังสร้าง Podfile.....	79
ข.4 ติดตั้ง Cocoapods	80
ข.5 ไฟล์ทั้งหมดในโปรเจค.....	80
ข.6 พิมพ์ Library ที่ต้องการติดตั้ง.....	81
ข.7 ติดตั้ง Library ลงโปรเจค.....	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากทางบริษัทแห่งหนึ่งได้มีการจัดทำแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ซึ่งเป็นแพลตฟอร์มที่อำนวยความสะดวกให้กับลูกค้าผู้ใช้บริการที่มีความต้องการสั่งอาหาร เนื่องจากมีความไม่สะดวกในการเดินทางไปรับประทานอาหารที่ร้าน โดยมีสาเหตุมาจากหลายปัจจัย เช่น ระยะเวลาที่จำกัด เป็นต้น นอกจากนี้ยังเป็นการเปิดโอกาสให้ลูกค้ามีตัวเลือกในการเลือกร้านอาหารที่หลากหลายมากยิ่งขึ้นเพื่อเป็นการเปิดประสบการณ์ใหม่ ๆ ให้กับลูกค้า และแอปพลิเคชันยังอำนวยความสะดวกให้แก่ผู้ประกอบการร้านค้าที่ต้องการขายอาหารบนแพลตฟอร์มออนไลน์ โดยให้ผู้ประกอบการร้านค้ามีทางเลือกในการขายอาหารได้มากยิ่งขึ้น รวมถึงมีการจัดหาบริการจัดส่งอาหารสำหรับร้านค้าที่ไม่มีผู้จัดส่งด้วยตัวเอง โดยในแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) นี้ ประกอบไปด้วยแอปพลิเคชันทั้งหมด 3 แอปพลิเคชัน ได้แก่ 1) แอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการ (Consumer) เป็นแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการที่มีความต้องการที่จะสั่งอาหารตามร้านต่าง ๆ 2) แอปพลิเคชันสำหรับผู้ประกอบการร้านค้า (Merchant) เป็นแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า โดยผู้ประกอบการร้านค้าสามารถที่จะใส่ที่อยู่ของร้านค้า เมนูอาหาร และกำหนดราคาอาหาร รวมไปถึงมีการจัดหาผู้จัดส่งให้กับร้านค้า และ 3) แอปพลิเคชันสำหรับผู้จัดส่ง (Rider) เป็นแอปพลิเคชันสำหรับผู้จัดส่งอาหาร โดยผู้จัดส่งสามารถเลือกรับงานการจัดส่งอาหารจากร้านค้า ติดต่อกับร้านค้า รวมไปถึงมีการแสดงเส้นทางการเดินทางเพื่อจัดส่งอาหารให้แก่ลูกค้า และเนื่องจากแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการ ยังต้องพัฒนาเพิ่มเติมในส่วนของการพัฒนาระบบการสนทนาเพื่อให้ลูกค้าผู้ใช้บริการ และผู้ประกอบการร้านค้ามีช่องทางการติดต่อที่สะดวกมากขึ้น

ดังนั้นจึงจำเป็นต้องมีการพัฒนาระบบการสนทนาของแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการ เพื่อใช้ในการแจ้งรายละเอียดต่าง ๆ เกี่ยวกับเมนูอาหารที่ได้ทำการสั่งซื้อ และยังเป็นการพัฒนาเพิ่มช่องทางการติดต่อสื่อสารระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้า

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อพัฒนาระบบการสนทนา ในแอปพลิเคชันบนมือถือสำหรับลูกค้าผู้ใช้บริการ (Consumer Application)
- 2) เพื่อเพิ่มทางเลือกช่องทางในการติดต่อสื่อสารระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้า
- 3) เพื่อเพิ่มความถูกต้องของรายละเอียดในรายการอาหารที่ลูกค้าผู้ใช้บริการได้ทำการสั่งกับทางผู้ประกอบการร้านค้า

1.3 ขอบเขตของงานวิจัย

- 1) แอปพลิเคชันบนอุปกรณ์เคลื่อนที่ (Mobile Application) จะทำงานบนระบบปฏิบัติการ iOS
- 2) แอปพลิเคชันบนอุปกรณ์เคลื่อนที่ (Mobile Application) ทำงานผ่านเครือข่ายอินเทอร์เน็ต
- 3) เป็นแอปพลิเคชันที่ใช้สำหรับลูกค้าผู้ใช้บริการ (Consumer Application) เท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) มีระบบการสนทนาในแอปพลิเคชันบนมือถือสำหรับลูกค้าผู้ใช้บริการ
- 2) ลูกค้าผู้ใช้บริการมีทางเลือกในการติดต่อสื่อสารกับผู้ประกอบการร้านค้ามากขึ้น
- 3) เพิ่มความถูกต้องของรายละเอียดในรายการอาหารที่ลูกค้าผู้ใช้บริการได้ทำการสั่งกับทางร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

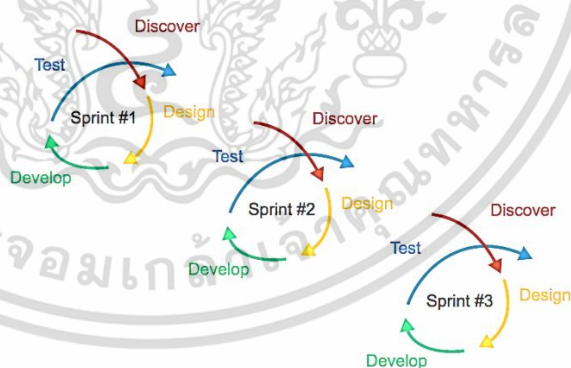
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงแนวคิดทฤษฎีความรู้ และหลักการที่เกี่ยวข้องกับการพัฒนาระบบสารสนเทศของแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) โดยแพลตฟอร์มนี้เป็นแพลตฟอร์มที่อยู่บนแอปพลิเคชันระบบปฏิบัติการ iOS

2.1 ทฤษฎีที่เกี่ยวข้องกับกระบวนการดำเนินงาน

2.1.1 การพัฒนาระบบด้วยแนวคิดแบบ Agile

ในกระบวนการพัฒนาระบบ Software Development ในองค์กรมักนิยมใช้แนวคิดการทำงานแบบ Agile ซึ่งเป็นแนวทางและวิธีการการพัฒนาซอฟต์แวร์ที่ใช้กันอย่างแพร่หลายในปัจจุบัน มีแนวคิดสำคัญคือไม่เคร่งในระเบียบวิธีมากเกินไป ไม่เน้นการจัดทำเอกสาร แต่เน้นผลผลิตหรือตัวซอฟต์แวร์เป็นหลัก และให้ความสำคัญกับการสื่อสารในการทำงานร่วมกัน โดยจะแบ่งแผนงานและเวลาการพัฒนาออกเป็นช่วง ๆ แต่ละช่วงมีระยะเวลาไม่เกิน 1-4 สัปดาห์ เพื่อให้ยืดหยุ่น พร้อมทั้งรับความเปลี่ยนแปลง ลดความเสี่ยงในการพัฒนา และเมื่อมีการเปลี่ยนแปลงหรืออุปสรรคเกิดขึ้นจะสามารถปรับแผนการพัฒนาได้อย่างทันที่



รูปที่ 2.1 แนวคิดการทำงานรูปแบบ Agile

จากรูปที่ 2.1 เป็นรูปที่แสดงถึงแนวคิดการทำงานรูปแบบ Agile ซึ่ง Agile Methodology เป็นโมเดลที่ไม่มีระเบียบวิธีตายตัว แต่สามารถที่จะนำไปปรับไปใช้ให้เหมาะสมกับองค์กร ซึ่ง Agile อาจจะเป็นส่วนขยายของกระบวนการพัฒนาซอฟต์แวร์อื่น ๆ ที่มีอยู่เดิมให้มีความยืดหยุ่นและประสิทธิภาพมากขึ้น ทั้งนี้ Agile มีหลักการและแนวทางสำคัญ ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้งานด้านการศึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) เน้นให้ความสำคัญกับทีมเวิร์ค สร้างสิ่งแวดล้อมให้สนับสนุนการสื่อสาร จัดให้สมาชิกในทีมมาจากหลายสายงาน เพื่อช่วยกันสร้างความเข้าใจในรายละเอียดของงานในแง่มุมต่าง ๆ
- 2) เน้นการสร้างซอฟต์แวร์ที่ใช้งานได้จริงมากกว่าการยึดถือเอกสารหรือรายงานแบ่งโครงการเป็นงานย่อย ๆ เพื่อกำหนดเป้าหมายที่ชัดเจนในระยะเวลาดสั้น ๆ
- 3) เน้นการทำงานร่วมระหว่างนักพัฒนาและลูกค้าหรือผู้ใช้ตลอดโครงการ เนื่องจาก Feedback ของผู้ใช้คือสิ่งเดียวที่จะบอกได้ว่าซอฟต์แวร์นั้นถูกพัฒนาไปในแนวทางที่ถูกต้อง
- 4) เน้นความยืดหยุ่น พร้อมรับการเปลี่ยนแปลงได้ตลอดเวลา เรียนรู้จากข้อผิดพลาดและผลสำเร็จ เพื่อปรับปรุงการทำงานช่วยให้โครงการสามารถดำเนินต่อไปได้อย่างราบรื่น

ซึ่งแนวคิดแบบ Agile มักจะมากคู่กับกรอบการทำงาน (Framework) แบบ Scrum เป็นวิธีการทำงานที่ได้รับความนิยมมากที่สุดสำหรับการทำงานภายใต้แนวคิดนี้ คือ วิธีการทำงานที่ให้ “ทีมช่วยกันรุมงาน” ซึ่งวิธีการทำงานที่ให้ทุกคนในทีมมาช่วยกันรุมงาน โดยวิธีการทำงานแบบ Scrum จะไม่มีตำแหน่งหรือลำดับขั้นตอนมากมาย จะมีเพียง 3 ตำแหน่งที่สำคัญเท่านั้น คือ

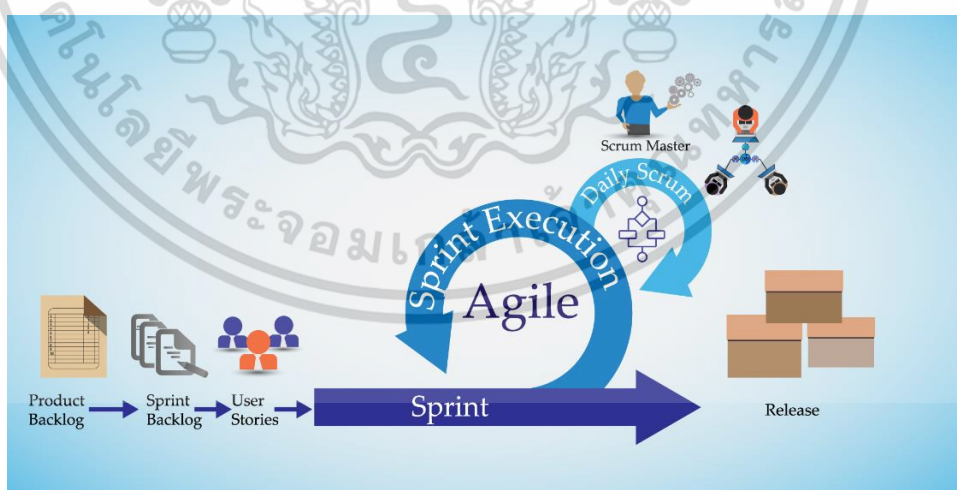
- 1) Product Owner (PO) เป็นผู้มีหน้าที่ประเมินคุณค่าของงาน (Values) และจัดลำดับความสำคัญของงาน (Priorities ของ Tasks ต่าง ๆ) ให้กับทีม ตำแหน่งนี้คือผู้ที่ใกล้ชิดกับลูกค้ามากที่สุด ดังนั้นจึงทำหน้าที่เป็นเสมือนเสียงของลูกค้า คอยตรวจสอบและสังเกตว่าอะไรที่น่าจะตอบโจทย์กลุ่มลูกค้าได้มากที่สุด เพื่อสามารถแก้ไขและปรับเปลี่ยนงานได้อย่าง Agile สิ่งสำคัญที่ Product Owner ต้องโฟกัสก็คือ Product Backlog Item (สิ่งที่ลูกค้าต้องการ) โดยทำให้ทั้งทีมเข้าใจวัตถุประสงค์ตรงกันอย่างโปร่งใส และตรวจสอบได้
- 2) Scrum Master เป็นเสมือนโค้ชผู้ทำให้การทำงานเป็นไปอย่างลื่นไหล ตำแหน่งนี้ไม่ได้ทำหน้าที่เป็นผู้นำทีมที่จะตัดสินใจสิ่งต่าง ๆ แต่คือคนที่ช่วยให้คนในทีมได้ตัดสินใจร่วมกัน ทำให้คนในทีมเข้าใจการทำงานแบบ Scrum ได้มากขึ้น และคอยกำจัดอุปสรรคที่ขัดขวางไม่ให้ทีมบรรลุเป้าหมาย โดยสิ่งที่ Scrum Master สามารถทำเพื่อช่วยให้ทีมทำงานบรรลุเป้าหมายนั้นได้แก่ กำหนด Timebox ซึ่ง Timebox คือการประเมินช่วงเวลาที่เหมาะสมสำหรับแต่ละกิจกรรม ถัดมาคือ ระบุ Definition of Done โดยนิยามของคำว่า “งานเสร็จ” ที่เข้าใจตรงกันทั้งทีม เช่น งานเสร็จหมายถึงผ่านการทดสอบเรียบร้อยแล้ว ก่อนที่ Product Owner จะเข้ามาตรวจสอบหรือตัดสินใจปล่อย Prototype ให้ลูกค้า และจัดประชุม Daily Scrum Meeting คือ การประชุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่หรือใช้เพื่อประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามทำซ้ำโดยไม่ได้รับอนุญาต และต้องยกย่องชื่อเจ้าของเอกสารทุกครั้ง

สำคัญมากสำหรับการทำงานแบบ Scrum เพื่อให้แน่ใจว่าทีมงานกำลังทำงานอยู่ในเป้าหมายหรือทิศทางที่กำหนดไว้ โดยเป็นการประชุมสั้น ๆ 10 - 15 นาทีในทุก ๆ เช้า เพื่อถามตัวเองว่า เมื่อวานทำอะไร วันนี้จะทำอะไร มีปัญหาอะไรบ้างที่ต้องแก้ไข แล้วงานของเราใกล้เคียงเป้าหมายหรือยัง

- 3) Development Team สำหรับ Development Team คีย์เวิร์ดสำคัญคือคำว่า Cross-Functional และ Self-Organized หมายความว่า ในทีม ๆ หนึ่งสามารถทำงานร่วมกันได้ตั้งแต่ต้นจนจบ โดยไม่จำเป็นต้องทำงานข้ามแผนกให้เสียเวลา ในหนึ่งทีมจะประกอบไปด้วยทุก ๆ ตำแหน่งรวมกัน เช่น Designer, Programmer, UI/UX, Testing โดยแต่ละแผนกจะเน้นทำงานแบบ Autonomy ทุกคนมีอำนาจในการตัดสินใจ ไม่ต้องรอขออนุญาต เน้นความว่องไวรวดเร็วและมีประสิทธิภาพเป็นพื้นฐานสำคัญ

ซึ่งรอบเวลาในการทำงานแบบ Scrum ซึ่งเป็นระยะเวลาสั้น ๆ จะแบ่งเป็น Sprint โดยความยาวของ Sprint อยู่ตั้งแต่ระหว่าง 1 – 4 สัปดาห์ เมื่อครบกำหนด Sprint ทีมงานจะต้องเสร็จงานตามที่ได้วางแผนไว้โดยไม่จำเป็นต้องรอให้งานเสร็จทั้งหมดสามารถนำบางส่วนที่ใช้งานก่อนได้ให้ลูกค้าหรือผู้ใช้งานแล้วค่อยทยอยส่วนอื่น ๆ เพิ่มเติมในภายหลัง ลักษณะแบบนี้เรียกว่า MVP หรือ Minimum Viable Product ถือเป็นเครื่องมือสำคัญในการประเมินตลาดว่าสินค้านั้นจะรุ่งหรือจะร่วงได้ดีทีเดียว ซึ่งในแต่ละ Sprint จะมีการทำกิจกรรมของ Scrum เพื่อให้การดำเนินงานเป็นขั้นตอนชัดเจน ตรวจสอบ วัตถุประสงค์และลดการประชุมที่ไม่จำเป็น กระบวนการทำงานรูปแบบ Scrum จะเป็นดังรูปที่ 2.2



รูปที่ 2.2 กระบวนการทำงานรูปแบบ Scrum

จากรูปที่ 2.2 เป็นรูปที่แสดงถึงกระบวนการทำงานในรูปแบบของ Scrum ซึ่งภายในกระบวนการทำงานแบบ Scrum จะประกอบไปด้วยกิจกรรมต่าง ๆ ดังนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

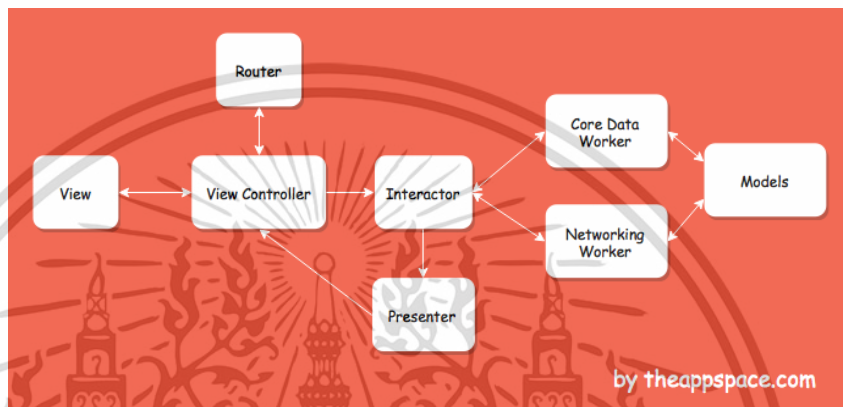
- 1) Sprint Goal การระบุวัตถุประสงค์ของ Sprint เช่น ต้องการทำให้มีผู้ใช้ใหม่เพิ่มขึ้นกี่เปอร์เซ็นต์หรือ ต้องการแนะนำสินค้าที่เกี่ยวข้องให้แก่ผู้ใช้ สิ่งสำคัญคือ การทำให้ทีมเห็นเป้าหมายตรงกัน รู้ว่าต้องทำงานอะไรบ้าง และรู้ว่าต้องทำงานชิ้นนี้ไปเพื่ออะไร
- 2) Sprint Planning กำหนดกิจกรรมหรืองานที่ต้องทำภายในระยะเวลาที่กำหนดสั้น ๆ ไม่เกิน 2 - 4 สัปดาห์ เป้าหมายก็คือการทำงานให้เสร็จภายในเวลาอันรวดเร็ว เน้นใช้งานได้จริงเป็นหลัก
- 3) Sprint Review คือการตรวจสอบผลลัพธ์ของ Sprint ตรวจสอบ วิเคราะห์และรับ Feedback เพื่อนำไปปรับปรุงแก้ไขเพิ่มเติม ส่วนมากจะใช้เวลาไม่เกิน 2 ชั่วโมงสำหรับ Sprint 2 สัปดาห์ ในขั้นตอนนี้คือ ขั้นตอนที่ต้องแสดงผลลัพธ์ของงานให้ผู้ที่เกี่ยวข้อง (Stakeholders) ตรวจสอบด้วย เพื่อรับฟังข้อเสนอแนะและทบทวน Product Backlog
- 4) Sprint Retrospective การตรวจสอบการดำเนินงานสำหรับ Sprint ที่จบลง ทั้งในเรื่องที่ทีมงาน ความสัมพันธ์ภายในทีม และสภาพแวดล้อมในการทำงาน โดยแต่ละคนในทีมต้องให้ Feedback กันรอบวงว่า สำหรับ Sprint ที่ผ่านมามีอะไรที่ดีและไม่ดีบ้าง จัดลำดับความสำคัญของผลกระทบที่เกิดขึ้นแล้วนำไปปรับปรุง โดย Scrum Master เข้าร่วมเสนอความคิดเห็นด้วยในฐานะสมาชิกของทีมและคอยควบคุมเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทฤษฎีที่เกี่ยวข้องกับกระบวนการการพัฒนาและกรอบงานของระบบ

2.2.1 Clean Swift Architecture

Clean Swift คือ Design Pattern หนึ่ง ซึ่งจะเป็นการลดความแออัดที่เกิดขึ้นใน View Controller โดยการนำสิ่งที่ไม่จำเป็นต้องอยู่ใน View Controller มาแบ่งฟังก์ชันการทำงาน อย่างเหมาะสม จุดประสงค์คือ ทำให้การ debug ในอนาคตสามารถทำได้ง่ายขึ้น โดยถูกแบ่ง Flow การทำงานเป็นดังนี้



รูปที่ 2.3 รูปภาพโครงสร้างของ Clean Swift

จากรูปที่ 2.3 เป็นโครงสร้างของ Clean Swift โดยที่แต่ละส่วนจะมีหน้าที่ที่ต่างกันออกไป ดังนี้

- View เป็นส่วนที่แสดงผลให้ผู้ใช้งานเห็น โดยภายในงานจะใช้ Storyboard ในการวางส่วนประกอบต่าง ๆ บนหน้าจอ
- View Controller เป็นส่วนที่จะนำข้อมูลไปแสดงผลบนหน้า Interface และรับค่าจาก presenter และเป็นส่วนที่มี Life Cycle
- Interactor เป็นตัวที่จะทำการส่ง Request ไปยัง Worker และเมื่อ Worker ส่งค่า Response กลับมาจะถูกส่งค่าต่อไปยัง Presenter
- Presenter เป็นตัวที่รับค่ามาจาก Interactor แล้วจะทำการแปลงผลและส่งค่าไปยัง View Controller เพื่อทำการแสดงผลที่หน้าจอ
- Worker เป็นเหมือน API ซึ่งเป็น Service ตัวกลางเชื่อมต่อระหว่าง Interactor กับ Model
- Model ส่วนที่จัดการข้อมูลที่จะนำไปแสดงผล โดยจะแบ่งด้วยกัน 3 ส่วน ก็คือ Request, Response, ViewModel
- Router เป็นตัวกำหนดทิศทางว่าจะสามารถไปยังหน้าไหนได้ และเป็นส่วนที่รับค่าต่าง ๆ จากหน้าอื่น ๆ และส่งค่าไปยังหน้าอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการขโมยข้อมูลโดยไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 View Controller Life cycle

Life cycle คือวงจรชีวิต หรืออาจจะเรียกได้ว่า วัฏจักรของหน้าจอ ซึ่ง Life cycle จะทำให้สามารถรู้ได้ว่าควรที่จะเรียก API หรือว่า update UI ตอนไหน โดย Life cycle จะประกอบไปด้วย 5 ฟังก์ชันหลัก ๆ ดังนี้

- 1) viewDidLoad
- 2) viewWillAppear
- 3) viewDidAppear
- 4) viewWillDisappear
- 5) viewDidDisappear

โดยหน้าที่ของแต่ละตัวและจะถูกเรียกใช้ตอนไหน สามารถดูได้ดังนี้

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        print("วิว 1: หลังจากโหลดหน้าแล้ว")
    }

    override func viewWillAppear(_ animated: Bool) {
        print("วิว 1: ก่อนที่วิวจะปรากฏ")
    }

    override func viewDidAppear(_ animated: Bool) {
        print("วิว 1: หลังจากวิวปรากฏแล้ว")
    }

    override func viewWillDisappear(_ animated: Bool) {
        print("วิว 1: ก่อนที่วิวจะหายไป")
    }

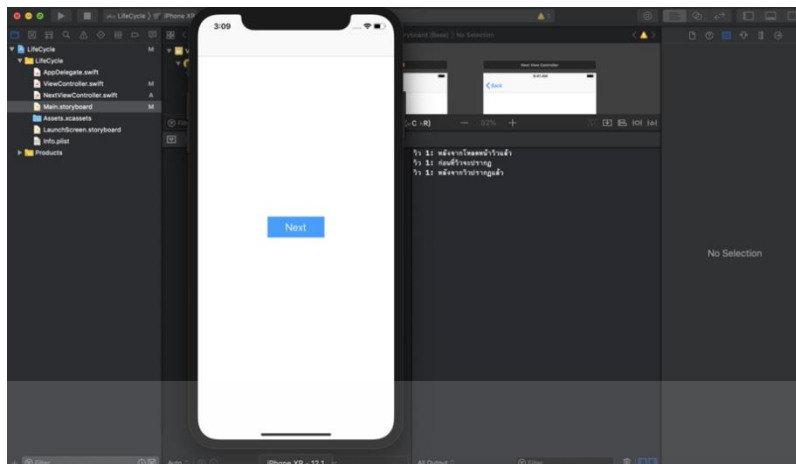
    override func viewDidDisappear(_ animated: Bool) {
        print("วิว 1: หลังจากวิวหายไปแล้ว")
    }

}
```

รูปที่ 2.4 ตัวอย่างการเรียกใช้งานทั้ง 5 ฟังก์ชัน

จากรูปที่ 2.4 เมื่อลองเรียกใช้งานฟังก์ชันแต่ละตัว หลังจากนั้นจะลองรันโค้ดดูเพื่อผลลัพธ์ ซึ่งจะได้ผลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างการทำงานเมื่อเปิดหน้า View Controller ในครั้งแรก

จากรูปที่ 2.5 จะเห็นได้ว่า เมื่อเปิดหน้าจอมาหน้าแรกจะเข้าฟังก์ชัน viewDidLoad เพื่อทำการแสดงผลลัพธ์ แล้วหลังจากนั้นจะเข้าฟังก์ชัน viewWillAppear และ viewDidAppear ตามลำดับ



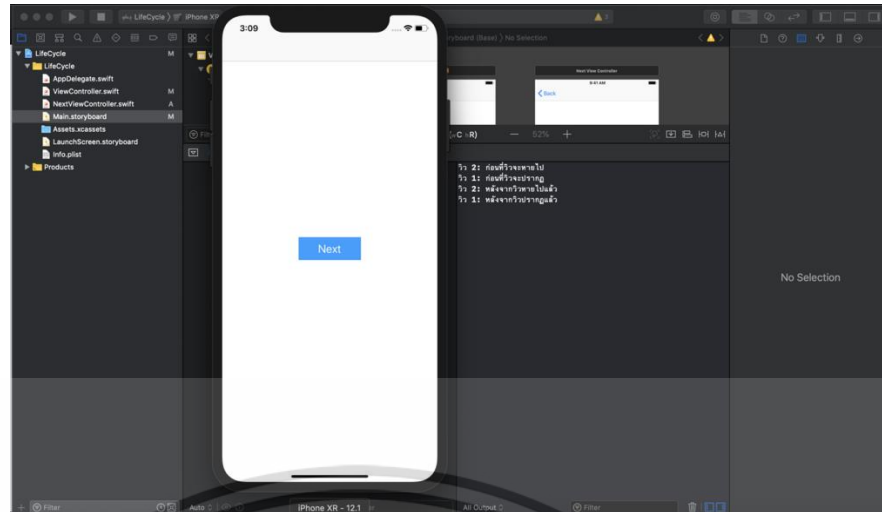
รูปที่ 2.6 ตัวอย่างการทำงานเมื่อทำการกดปุ่มเพื่อไปหน้าถัดไป

จากที่รูปที่ 2.6 เมื่อได้ทำการกดที่ปุ่ม next จะเห็นได้วก่อนที่จะเข้าฟังก์ชัน viewWillDisappear ของหน้าแรก จะเข้าฟังก์ชัน viewDidLoad ของหน้าที่ 2 ก่อน แล้วจึงตามด้วยฟังก์ชัน viewWillDisappear ของหน้าแรก หลังจากนั้นจะเข้าฟังก์ชัน viewWillAppear ของหน้าที่ 2 ก่อน จึงจะเข้าฟังก์ชัน viewDidDisappear ของหน้าแรก

หลังจากกดเข้าหน้าที่ 2 มาแล้ว จะทำตกกลับเพื่อกลับไปยังหน้าแรก เพื่อดู Life

Cycle ซึ่งจะได้ผลดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ตัวอย่างการทำงานเมื่อคลิกปุ่มกลับมายังหน้าแรก

จากรูปที่ 2.7 จะเห็นได้ว่า เมื่อคลิกกลับมายังหน้าแรก จะเข้าฟังก์ชัน `viewWillDisappear` ของหน้าที่ 2 แล้วจึงตามด้วยฟังก์ชัน `viewWillAppear` ของหน้าแรก หลังจากนั้นจะเข้าฟังก์ชัน `viewDidDisappear` ของหน้าที่ 2 ก่อน จึงจะเข้าฟังก์ชัน `viewWillAppear` ของหน้าแรก จากข้างต้น แสดงให้เห็นว่าจะมีการเข้าฟังก์ชัน `viewDidLoad` เพียงครั้งเดียว ซึ่งส่วนนี้เป็นส่วนที่สำคัญในการที่จะเรียก API หรือทำการ Update UI ในกรณีที่ต้องการให้มีการเรียก API หรือ Update UI ทุกครั้งที่เข้ามายังหน้าแรก ไม่ว่าจะเป็นการเปิดหน้าแรกขึ้น หรือแม้แต่การกลับมาจากหน้าถัดไปให้ทำในฟังก์ชัน `viewWillAppear` ส่วนในกรณีที่เราต้องการเรียก API หรือ Update UI แค่ครั้งเดียวในตอนเปิดหน้านั้น ๆ และไม่ต้องการให้ทำอีกในกรณีที่กลับมาจากหน้าถัดไปให้ทำใน `viewDidLoad`

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 Design Pattern

MVVM Pattern ประกอบไปด้วย Model-View-ViewModel ซึ่ง MVVM เป็นการแยก View และ Model ออกจากกัน โดย MVVM นั้นจะมี ViewModel เพิ่มขึ้นมา ViewModel จะทำหน้าที่เก็บข้อมูลทั้งหมดที่ View ต้องการ โดยระหว่าง View กับ ViewModel จะถูก Implement โดยสิ่งที่เรียกว่า Data - binding ซึ่งถ้า View มีการเปลี่ยนแปลงก็จะส่งผลต่อ ViewModel ด้วย ในทางกลับกัน การเปลี่ยนแปลงของ ViewModel ก็ส่งผลถึง View เช่นกัน ใน MVVM ส่วนของ View จะไม่รู้อะไรเลยนอกจากการแสดงผล ส่วน Logic อื่น ๆ จะอยู่ใน ViewModel ทั้งหมด

ใน MVVM ส่วนของ View จะถือ Reference ของ ViewModel ไว้ แต่ ViewModel จะไม่รู้จัก View เลย ดังนั้นความสัมพันธ์ระหว่าง ViewModel กับ View จะเป็นแบบ One-to-many โดยที่ ViewModel หนึ่งตัวจะสามารถใช้กับ View หลาย ๆ ตัวได้ โดยใน MVVM ส่วนของ View จะไม่รู้จัก Model เลย และ Model ก็จะไม่รู้จัก View เช่นกัน ซึ่งการใช้ MVVM ใน iOS มีข้อดีคือ ทำให้ง่ายต่อการเขียน Unit Test และง่ายต่อการ Maintain

การเขียน MVVM กับ iOS จะมี 4 ส่วนหลัก ๆ ดังนี้



รูปที่ 2.8 แผนภาพแสดงแนวคิดของ MVVM ใน iOS

จากรูปที่ 2.8 เป็นแผนภาพที่แสดงแนวคิดของ MVVM ใน iOS ซึ่งจะประกอบไปด้วย 3 ส่วน คือ View (ViewController), ViewModel และ Model ซึ่งจะเห็นได้ว่าทั้ง 3 ส่วนนี้ มีการรับและส่งข้อมูลอยู่ตลอดเวลา เปรียบเสมือนท่อที่เปิดเพื่อรับและส่งข้อมูลไปมาระหว่างกันอยู่ตลอดเวลา ซึ่งแต่ละส่วนจะมีหน้าที่ที่แตกต่างกัน ดังนี้

- 1) View หรือ ViewController เป็นส่วนที่ไว้ใช้ในการควบคุมการแสดงผลและแสดงผลให้กับผู้ใช้ โดย View หรือ ViewController จะสามารถส่งและรับ

ข้อมูลไปยัง ViewModel ผ่านการใช้ Data binding และรับค่าที่มีการ Update

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

จากหน้า viewModel มาแสดงผล

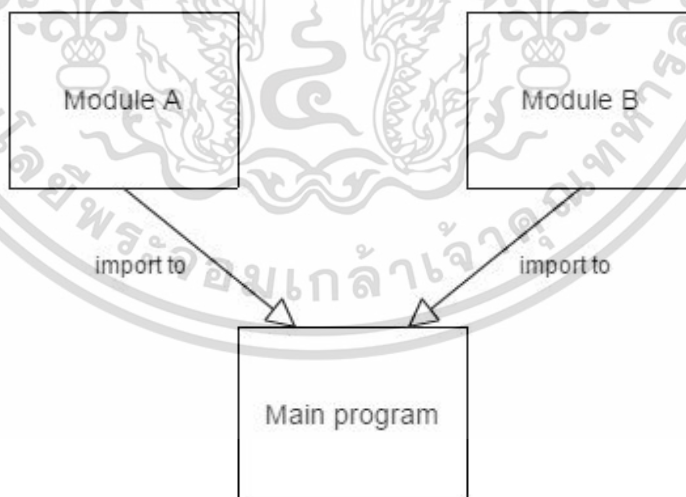
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) ViewModel เป็นส่วนที่รับข้อมูลจาก View และส่งค่ากลับไปหา View ซึ่ง ViewModel จะทำหน้าที่ในส่วนของ Logic ทั้งหมด และมีหน้าที่ในการเรียก API เพื่อดึงข้อมูลมาจาก Server มาจัดการและส่งข้อมูลกลับไปยัง View เพื่อทำการแสดงผลบนหน้าจอ เมื่อ viewModel มีการเปลี่ยนแปลงค่าก็จะทำการอัปเดตค่าใน Model ด้วย
- 3) Model เป็นส่วนที่ใช้เพื่อเก็บค่าหรือตัวแปรที่ใช้ต่าง ๆ โดยเมื่อ ViewModel มีการเปลี่ยนแปลงค่าและค่าใน Model มีการอัปเดต Model ก็ส่งค่ากลับไปให้ ViewModel

2.2.4 การเขียนโปรแกรมแบบโมดูลาร์

โมดูล (Module) คือ ไฟล์หรือส่วนของโปรแกรมที่ใช้สำหรับกำหนดตัวแปร ฟังก์ชัน หรือคลาส โดยแบ่งย่อยอีกหน่วยหนึ่งจากโปรแกรมหลัก และในโมดูลยังสามารถประกอบไปด้วยคำสั่งประมวลผลการทำงานได้

โมดูล คือการแยกส่วนของโปรแกรมออกเป็น ส่วน และสามารถเรียกใช้ได้เมื่อต้องการ ซึ่งโมดูลเป็นเหมือนไลบรารีของฟังก์ชันและคลาสต่าง ๆ เพราะว่าเมื่อโปรแกรมมีขนาดใหญ่ก็สามารถแบ่งส่วนต่าง ๆ ของโปรแกรมออกเป็นโมดูลย่อย ๆ เพื่อให้ง่ายต่อการจัดการและการทำงาน โดยการทำงานแบบโมดูลจะเป็นไปตามรูปที่ 2.9



รูปที่ 2.9 แผนภาพการทำงานแบบโมดูลาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 การแลกเปลี่ยนข้อมูลผ่าน API

API (Application Programming Interface) คือ ตัวกลางที่เชื่อมต่อระหว่าง “ผู้ใช้บริการ” กับ “ผู้ให้บริการ” เมื่อผู้ใช้บริการต้องการข้อมูลบางอย่าง API จะทำหน้าที่ส่งต่อคำขอไปยังผู้ให้บริการและส่งข้อมูลกลับมาที่ผู้ใช้บริการโดยอัตโนมัติ โดยระบบการสนทนาจะใช้ RESTful Web Service

RESTful Web Service (RWS) คือ Web Service ที่ใช้สถาปัตยกรรม Rest ซึ่งเป็นรูปแบบสถาปัตยกรรมซอฟต์แวร์ (Architecture) ที่ใช้ประโยชน์จากเทคโนโลยี Web Protocol เพื่อใช้ในการสร้าง Web Service โดย RWS อนุญาตให้ระบบ Request และเข้าถึง Resource บนเว็บ โดยใช้ชุดคำสั่งที่กำหนดเอาไว้ล่วงหน้า ซึ่งการโต้ตอบของระบบที่ใช้ REST จะอยู่บนพื้นฐานของ Hypertext Transfer Protocol (HTTP) ซึ่ง Request จะส่งคำขอไปยัง URL ที่กำหนด เพื่อนำไปค้นหาและประมวลผลแล้วจะได้ Response กลับมาในรูปแบบ XML หรือ JSON โดย RESTful จะประกอบไปด้วย Client คือ ผู้ที่เข้ามาเป็น Request Resource และ Server คือ ผู้ที่ให้บริการ Resource

2.2.5.1 HTTP Request

คำสั่ง HTTP Request คือคำสั่งที่ Client ส่งไปยัง Server เพื่อให้ Server ส่งข้อมูลตอบกลับมา โดยภายใน HTTP Request จะประกอบไปด้วย

- 1) URL เป็นส่วนที่ใช้ระบุที่อยู่ของ Server ที่ Client ต้องการขอข้อมูล
- 2) HTTP Request Headers คือส่วนที่ระบุข้อมูล และกฎต่าง ๆ ในการเชื่อมต่อ
- 3) HTTP Method คือส่วนที่ใช้กำหนดประเภทของคำร้องขอ ซึ่งตอนออกแบบ RESTful Web service จะต้องกำหนดการเข้าถึงข้อมูลแต่ละแบบว่าจะให้ใช้ Method ใด โดยประเภทของคำสั่ง Request ที่ใช้จะมีอยู่ 4 Methods ที่ใช้งานบ่อย ได้แก่

- GET ใช้สำหรับขอข้อมูลจาก Server ซึ่ง method GET ควรใช้ในการดึงข้อมูลเท่านั้น
- POST ใช้สำหรับการสร้างข้อมูลใหม่โดยส่งข้อมูลผ่าน body
- PUT ใช้สำหรับแทนที่ข้อมูลที่มีทั้งหมดด้วยข้อมูลใหม่ที่ส่งไป
- DELETE ใช้ลบข้อมูลที่มีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) HTTP Request Body เป็นส่วนสุดท้ายของ Request message จะเป็นส่วนที่ใช้สำหรับส่งข้อมูลให้กับ Server อาจจะมีหรือไม่มีก็ได้

2.2.5.2 HTTP Response

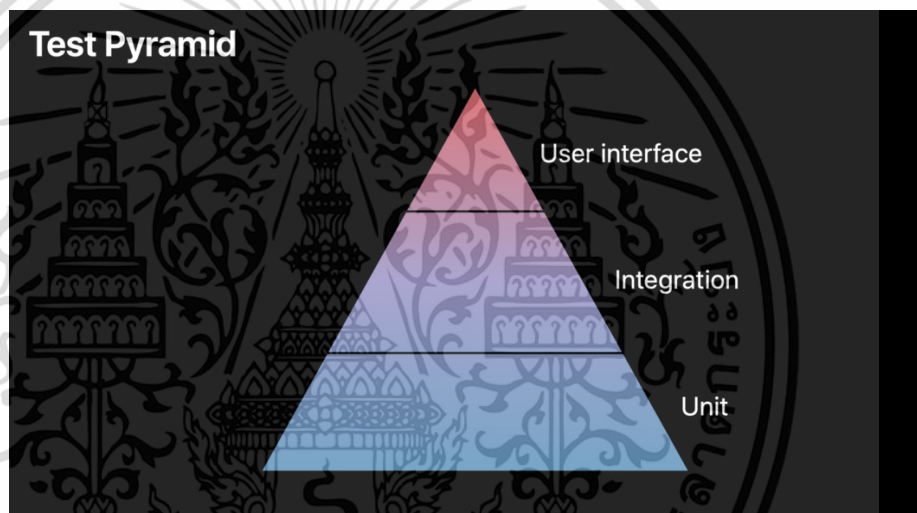
การ Response จะเกิดขึ้นหลังจากส่ง Request ไปที่ Server แล้ว Server ก็จะมีการตอบรับกลับมา ซึ่งข้อความที่ตอบกลับมาจะเรียกว่า HTTP Response message โดย Response message จะประกอบไปด้วย 2 ส่วนหลัก ได้แก่

- 1) HTTP Response Header จะเป็นส่วนของการระบุข้อมูลรายละเอียดที่บอกข้อมูลทางเทคนิค เช่น ขนาดของข้อมูล ประเภทของเนื้อหา เป็นต้น
- 2) HTTP Response Body จะเป็นส่วน HTML หรือข้อมูลใน Format ใด ๆ เช่น JSON, XML เป็นต้น ที่เราร้องขอไปกลับมา ซึ่งในส่วนนี้จะมี Status Code ส่งกลับมาให้ โดย Status Code ซึ่งเป็นรหัสบอกสถานะว่า การ Request ครั้งนั้นได้รับการตอบรับเป็นอย่างไร Status Code ที่พบได้บ่อย ๆ เช่น 200 OK มีความหมายว่า Request ดังกล่าวทำงานสำเร็จ และข้อมูลถูกส่งกลับมายัง Client ได้อย่างถูกต้อง และ 500 Internal Server Error หมายถึง เกิดการทำงานที่ไม่สมบูรณ์ขึ้นที่ Server ทำให้ไม่สามารถตอบกลับ Client ตามที่ร้องขอมาได้

2.2.6 Testing in Xcode with XCTest

ใน XCode นั้นมี XCTest ซึ่งเป็น Framework ที่ช่วยให้สามารถเขียน Automation Test บน Xcode ได้โดยไม่ต้องไปมองหา Framework อื่น ซึ่ง XCTest นั้นเริ่มมีให้ใช้งานมาตั้งแต่ Xcode 5 (ปี 2013) โดย Concept การทำเทสที่ดีควรที่จะ Apply Test Pyramid Concept ในการพัฒนา Software Project เพื่อให้การเขียนเทสนั้นมีประสิทธิภาพที่ดีที่สุด

Test Pyramid คือหลักการพื้นฐานที่จะช่วยให้สามารถ Balance ของการทำ Automation Test ที่เหมาะสม ทั้งในแง่ของ Thoroughness, Quality และ Execution Speed ซึ่งในแต่ละ Test Level บน Test Pyramid ก็จะมีสัดส่วนของทั้งสามองค์ประกอบที่แตกต่างกันไป ซึ่งดูได้จากรูปที่ 2.10



รูปที่ 2.10 Test Pyramid

จากรูปที่ 2.10 Test Pyramid จะประกอบไปด้วย 3 องค์ประกอบ คือ Unit Test Integration Test และ User Interface Test ซึ่งแต่ละส่วนจะมีความสำคัญในการทดสอบที่แตกต่างกัน ดังนี้

Unit Test เป็นการทำเทสในหน่วยย่อยที่สุด ระดับ Single Method ซึ่งเป็น Foundation ของการเขียนเทสทั้งหมดในแอปพลิเคชัน โดย Unit Test นั้นสามารถรันได้อย่างรวดเร็วและให้ผลลัพธ์ที่ชัดเจนว่าโค้ดมีปัญหาตรงไหน (Easy to Investigation) ดังนั้นจึงจำเป็นต้องเขียน Unit Test ให้กับแอปพลิเคชันมากที่สุด

Integration Test คือการเทสเพื่อตรวจสอบดูว่า Component ต่าง ๆ ที่อยู่ภายในแอปพลิเคชันสามารถทำงานร่วมกันได้อย่างถูกต้อง ซึ่งจะรันได้ช้ากว่า Unit Test แต่เน้นการดูการ Integrate Component ย่อย ๆ ของแอปพลิเคชันว่าสามารถทำงานร่วมกันได้ไหม

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User Interface Test หรือ UI Test เป็นการทดสอบเพื่อดูว่าแอปพลิเคชันทำงานได้เหมือนกับที่ต้องการหรือไม่ ซึ่งเป็นการทดสอบที่เหมือนกับที่ User ใช้งานจริง ซึ่ง UI Test นั้นต้องการ Effort ในการ Maintenance สูง เพราะ UI นั้นมีโอกาสเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นจึงควรมี UI Test ในโปรเจกต์น้อยที่สุด

บน Xcode นั้นก็มี Test Target อยู่ 3 แบบคือ

- 1) Unit Tests ใช้เน้นดูการทำงานของ Source Code ว่าทำงานถูกต้องหรือไม่
- 2) UI Tests ใช้ทดสอบในลักษณะที่เป็นมุมมองการใช้งานของ User จริง ๆ ดู Workflow การทำงานของแอปพลิเคชัน
- 3) Performance Tests สามารถใช้ใน Benchmark การทำงานของแอปพลิเคชันได้ เช่น CPU, Memory Usage

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 การเขียน Unit Test

Unit Test คือการทดสอบเฉพาะจุดที่สนใจ เช่น ทดสอบฟังก์ชันที่มี Business Logic ที่สำคัญ (Business Logic จะอยู่ใน View Model) ซึ่ง Unit Test สามารถรันและให้ผลทดสอบได้อย่างรวดเร็วและให้ผลลัพธ์ที่ชัดเจนว่าโค้ดมีปัญหาตรงไหน (Easy to Investigation) โดยการเขียน Unit Test สามารถเขียนได้หลายรูปแบบ ภายในโปรเจกต์จะเขียน Unit Test โดยใช้แบบ BDD

BDD หรือชื่อเต็ม คือ Behavior-Driven Development คือการเขียน Unit Test โดยคำนึงถึงผลลัพธ์ให้เป็นไปตาม User Story หรือ Requirement เพราะแทนที่จะทดสอบในระดับ Detail ของแต่ละฟังก์ชัน ก็เปลี่ยนไปทดสอบ User Story หรือ Requirement ของงานนั้น ๆ ว่าต้องการผลลัพธ์ยังไงแทน ซึ่งถ้าผลลัพธ์ออกมาถูก จะช่วยยืนยันส่วนหนึ่งว่าการทำงานภาพรวมก่อนผลลัพธ์นั้นจะออกมา มันทำงานได้ถูกต้อง ถึงแม้ว่าจะไม่ได้ไปเขียนทดสอบส่วนนั้น ๆ ก็ตาม ทำให้ Unit Test นั้น เขียนง่าย อ่านง่าย แก้ไขง่าย และทำตัวเป็นกึ่งเอกสารไปในตัวด้วย เพราะเมื่อไหร่ก็ตามที่ไม่แน่ใจว่าในโค้ดชุดนี้มันถูกสร้างไว้ทำอะไร มีเงื่อนไขอะไรบ้าง เนื่องจากทำไว้นานแล้ว หรือต้องมาแก้ไขโค้ดของคนอื่น ก็สามารถมาดู Unit Test นี้ได้ว่ามัน Expect อะไร มีเงื่อนไขอะไรบ้าง ซึ่งการเขียน Unit Test แบบ BDD ภายใน 1 เคสที่ต้องการทดสอบ จะประกอบไปด้วย 3 ส่วนหลัก ๆ คือ

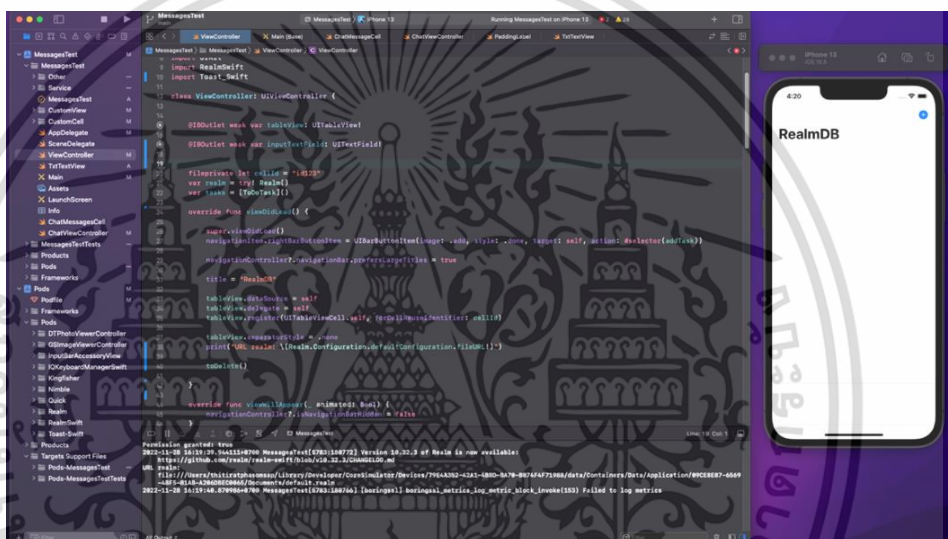
- 1) Given คือ การกำหนดว่าจะเกิดเหตุการณ์อะไรขึ้นบ้าง และมีการ Mock Data หรือสร้างข้อมูลจำลองขึ้นมา
- 2) When คือ การนำข้อมูลที่จำลองขึ้นมาไปทดสอบกับ Code Logic ใน ViewModel
- 3) Then คือ ส่วนที่ใช้เขียนเพื่อทดสอบว่า User ควรที่จะได้รับผลแบบไหน ซึ่งจะต้องทำการตรวจสอบ (Assert) ค่าว่าตรงตามที่ได้กำหนดไว้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ทฤษฎีที่เกี่ยวข้องเกี่ยวกับเครื่องมือที่ใช้ในการพัฒนา

2.3.1 Xcode

Xcode เป็นเครื่องมือที่ใช้สำหรับพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ iOS, iPadOS, macOS, tvOS และ watchOS โดย Xcode มาพร้อมกับเครื่องมือทุกอย่างที่จำเป็นต้องใช้เพื่อสร้างสรรค์แอปพลิเคชัน อาทิ หน้าจอสำหรับการเขียนคำสั่ง หน้าจอสำหรับการออกแบบส่วนติดต่อกับผู้ใช้ และส่วนสำหรับจำลองการทำงาน (Simulator) ของอุปกรณ์ต่าง ๆ ทำให้สามารถรันแอปพลิเคชันบนเครื่องจำลอง เช่น iPhone iPad หรือแม้กระทั่ง Apple Watch ได้ทันที โดย Xcode มีหน้าตาโปรแกรมดังรูป 2.11



รูปที่ 2.11 โปรแกรม Xcode

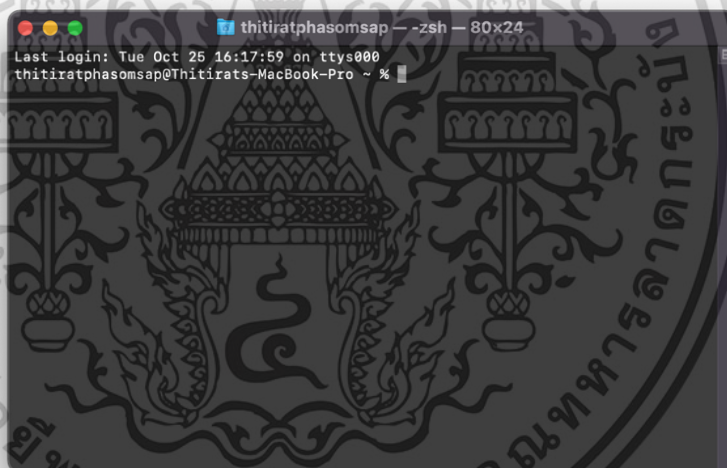
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 Cocoapods

CocoaPods คือ ตัวจัดการ Library และ Dependencies ต่าง ๆ ให้กับแอปพลิเคชันที่พัฒนาโดยใช้ภาษา Swift หรือโดยเฉพาะในระบบ iOS ทำให้ไม่ต้องไปดาวน์โหลดไฟล์ Library ต่าง ๆ จากแหล่งอื่น และ CocoaPods จะช่วยอำนวยความสะดวกให้กับผู้ใช้ นั่นคือทำให้สามารถดึง Library ใหม่ ๆ มาใช้ หรือจะเป็นการ Maintain ระบบได้อย่างง่ายดาย สะดวก และที่สำคัญคือไม่ต้องไปยุ่งกับ Environment เดิมของ Project อีกด้วย

2.3.3 macOS Terminal

Terminal คือระบบ Command Line ในระบบปฏิบัติการ macOS โดยระบบการทำงานจะคล้ายกับระบบปฏิบัติการ UNIX เพื่อใช้ควบคุมระบบปฏิบัติการ เช่น การเปิดไฟล์ หรือสร้างไฟล์โดยใช้แอปพลิเคชันพื้นฐานและความสามารถอื่น ๆ อีกมากมาย ซึ่งในที่นี้จะใช้ในการติดตั้ง Cocoapods โดยตัวอย่างโปรแกรม macOS Terminal จะเป็นดังรูปที่ 2.12



รูปที่ 2.12 MacOS Terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 Sourcetree

Sourcetree เป็น Software ตัวหนึ่งที่ใช้ GUI แทนการ Run ผ่านหน้าต่าง CMD ใน Windows และหน้าต่าง TERMINAL ใน Linux เพื่อให้ผู้ใช้งานใช้งานได้สะดวกขึ้น ซึ่ง Git คือ Version Control แบบ Distributed ตัวหนึ่ง ซึ่งเป็นระบบที่ใช้ Backup code และควบคุมการเปลี่ยนแปลง Files (เช่น Text File, Binary File, Python File โดยจะเรียกรวมว่า Source Code) ภายใน Project เมื่อมีการจัดเก็บ Source Code เข้าในระบบ Git จะเรียกว่า Git Repository โดยระบบจะจัดเก็บและจดจำการเปลี่ยนแปลงไฟล์ในแต่ละ version เพื่อให้สามารถย้อนกลับไปใช้ version เก่าได้ อีกทั้งยังสามารถดูว่าใครเป็นคนแก้ไขไฟล์ได้อีกด้วย และระบบของ Git สามารถแบ่งกันทำงานคนละ Branches ได้โดยการ Push code ขึ้นไปเก็บที่ Remote Repository (Git Hosting) เมื่อเพื่อนร่วมทีม Pull Branches ลงมาก็จะได้ version ล่าสุดไปพัฒนาต่อได้ โดยจุดเด่นของ sourcetree มีดังนี้

- 1) ง่ายต่อการใช้งาน ผ่านหน้าต่างโปรแกรมที่เป็นสัดส่วนชัดเจน
- 2) ง่ายต่อการดู Git History เนื่องจากมีแผนผังแสดงประวัติให้ดูได้อย่างชัดเจน
- 3) มีหน้าจอแสดงการเปลี่ยนแปลงของ Source code อย่างชัดเจน ทำให้รู้ได้ว่าแต่ละ Commit ได้ทำอะไรลงไปบ้าง
- 4) สามารถใช้งานระดับ Advance ผ่านเมนู Terminal ที่มาพร้อมกับโปรแกรมได้

2.3.5 Figma

Figma คือเครื่องมือออกแบบที่สามารถใช้ออกแบบได้ตั้งแต่เว็บไซต์ แอปพลิเคชัน หรือใช้สำหรับการแบบโลโก้ Artwork ต่าง ๆ โดย Figma ให้ความสำคัญในเรื่องการทำงานร่วมกันภายในทีม ทำให้ทีม UX/UI Design สามารถทำงานร่วมกันได้ง่ายขึ้น รวมไปถึงส่งเสริมการทำงานระหว่างทีม ที่ช่วยให้ Designer ส่งต่องานกับ Developer ได้ง่ายมากยิ่งขึ้น โดยสิ่งแรกที่เห็นได้ชัดคือ ตัวโปรแกรม Figma ที่ใช้งานในรูปแบบ Browser - Based ที่ทุกคนสามารถทำงานพร้อมกันได้ทุกที่ ทุกเวลา และยังมี Features อีกมากมายที่ช่วยให้การส่งต่องานระหว่างทีม ทำให้สามารถทำงานได้ง่ายกว่าเครื่องมือออกแบบอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 Postman

Postman เป็นเครื่องมือที่ไว้ใช้สำหรับ API Developers ซึ่งหากให้อธิบายในความหมายง่าย ๆ ก็คือ โดยปกติแล้วเมื่อเราเขียน API Service ขึ้นมาเรามักจะใช้ Postman ที่เป็น API Testing tool ในการส่ง Request และดู Response ที่ได้ต่าง ๆ กลับมา

2.3.7 Firebase

Firebase คือ Platform ที่รวบรวมเครื่องมือต่าง ๆ สำหรับการจัดการในส่วน Backend หรือ Server side ซึ่งทำให้สามารถ Build Mobile Application ได้อย่างมีประสิทธิภาพ และยังลดเวลาและค่าใช้จ่ายของการทำ Server side หรือการวิเคราะห์ข้อมูลให้อีกด้วย โดยมีทั้งเครื่องมือที่ฟรี และเครื่องมือที่มีค่าใช้จ่าย (สำหรับการ Scale) ซึ่ง firebase มีบริการหลายอย่างให้ใช้ โดยในการพัฒนาระบบการสนทนา จะใช้ Firebase Cloud Message (FCM)

Firebase Cloud Messaging (FCM) คือ บริการส่งข้อความแจ้งเตือนแบบข้ามแพลตฟอร์ม (cross-platform messaging) โดยจะเป็นตัวที่ทำให้ Mobile Application สามารถรับ Notification ได้ โดยจะส่ง message ไปหาได้ทุก Platform ทั้ง Android, iOS และ Web แบบฟรี ๆ

2.3.8 WebSocket

WebSocket เป็นเทคโนโลยีเพื่อใช้ติดต่อสื่อสารระหว่าง Web Server กับ Client แบบ Real Time โดย Client ที่สามารถรับข้อมูลทางฝั่ง Server มาแสดงผลได้ผ่าน Protocol TCP/IP โดยไม่ต้อง Refresh หน้า ซึ่งข้อดีของ WebSocket ไม่ต้องส่ง Request ใหม่ เพื่อส่งคำร้องขอไปยัง Server และรอรับ Response จากทางฝั่ง Server ทำให้ WebSocket ประหยัดทั้งเวลาและปริมาณข้อมูลที่ส่งไปมาระหว่าง Web Server กับ Client โดยการทำงานของ WebSocket มันจะรอรับ Message ทางฝั่ง Web Server อยู่ตลอดเวลา หลังจากที่ได้ส่งคำสั่ง ติดต่อ WebSocket Server ไปแล้วในครั้งแรก และมันจะรอรับ Message จนกระทั่งจะสั่ง Close หรือหยุดการติดต่อ โดยในการใช้ Socket บน mobile application จะใช้ Socket.io ซึ่ง Socket.io เป็น JavaScript frameworks ที่เอาไว้เรียกใช้งาน WebSocket เพื่อคอยรับส่งข้อมูลจาก client-browser และ server และทำงานในแบบ Real-time ตัวอย่างของการใช้งาน WebSocket เช่น Chat room, Document collaboration ซึ่งจะแตกต่างจากรูปแบบการทำงานของเว็บทั่วไป หรือ RESTful HTTP ซึ่งจะมีลักษณะการทำงานแบบ Request-Response นั่นคือ เมื่อ Client ทำการส่ง Request หน้าเว็บไป ตัว Server ก็จะตอบกลับมาด้วย Response และจะทำงานในลักษณะนี้ไปเรื่อย ๆ แต่ในการทำงานของ WebSocket นั้น จะมีลักษณะเป็นแบบ Event - Based Communication นั่นคือ จะทำงานตามที่เราเขียน Code ไว้ เช่น ถ้าเกิดเหตุการณ์ A ก็ให้ส่งข้อมูล B ไป เป็นต้น

2.3.9 Realm

Realm เป็นฐานข้อมูลสำหรับเก็บข้อมูลบน Mobile Application โดยจุดเด่นของ Realm มีดังนี้

- มีระบบการจัดการและเก็บบันทึกข้อมูลของตัวเอง
- มีการออกแบบให้ใช้งานได้ง่ายสำหรับนักพัฒนา
- เน้นทำงานได้เร็วมาก
- สามารถใช้งานข้ามแพลตฟอร์มได้ โดยปัจจุบันรองรับในแอปที่สร้างด้วย Java, Objective-C, Swift, React Native และล่าสุดคือ Xamarin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ทฤษฎีที่เกี่ยวข้องกับภาษาโปรแกรมที่ใช้ในการพัฒนา

2.4.1 ภาษา Swift

Swift คือ ภาษาการเขียนโปรแกรมที่ได้รับการสร้างสรรค์โดย Apple เพื่อใช้ในการสร้างแอปพลิเคชันสำหรับอุปกรณ์ iOS, Mac, Apple TV และ Apple Watch เรียกได้ว่าเป็นภาษาที่ออกแบบมาเพื่อให้นักพัฒนามีอิสระยิ่งกว่าที่เคย นอกจากนี้ Swift ยังใช้งานง่ายและเป็นโอเพ่นซอร์ส โดย Swift เป็นภาษาที่ทำงานได้เร็วและมีประสิทธิภาพ จึงสามารถส่งฟีดแบ็คแบบเรียลไทม์ และทำงานร่วมกับโค้ด Objective-C ที่มีอยู่แล้วได้ ไม่ว่าจะเป็นไลบรารีและเฟรมเวิร์คที่เขียนโดยภาษา Objective-C ดังนั้น นักพัฒนาจึงสามารถเขียนโค้ดที่ปลอดภัยขึ้น เสถียรขึ้น ทั้งยังประหยัดเวลาและเพิ่มประสบการณ์การใช้งานได้อีกด้วย ซึ่งแอปที่เขียนด้วย Swift นั้นสามารถทำงานได้รวดเร็ว เช่น อัลกอริทึมการค้นหาทั่วไป จะทำงานเสร็จเร็วกว่าเมื่อใช้ Swift

Swift เป็นภาษาโอเพ่นซอร์สที่เปิดให้ใช้งานฟรีสำหรับผู้ที่สนใจหลากหลายกลุ่ม ไม่ว่าจะเป็นนักพัฒนา นักการศึกษาหรือนักศึกษา ภายใต้สิทธิ์การใช้งานแบบโอเพ่นซอร์ส Apache 2.0 โดยได้จัดเตรียมไบนารีสำหรับ macOS และ Linux ที่นำไปคอมไพล์โค้ดสำหรับ iOS, macOS, watchOS, tvOS และ Linux ได้ นอกจากนี้ เพื่อเป็นการช่วยให้ Swift เติบโตและเป็นภาษาที่ทรงพลังมากยิ่งขึ้น จึงได้มีการสร้างชุมชนขึ้นมาเพื่อให้ผู้มีส่วนช่วยพัฒนาซอร์สโค้ดของ Swift ได้โดยตรง

2.5 งานวิจัยที่เกี่ยวข้อง

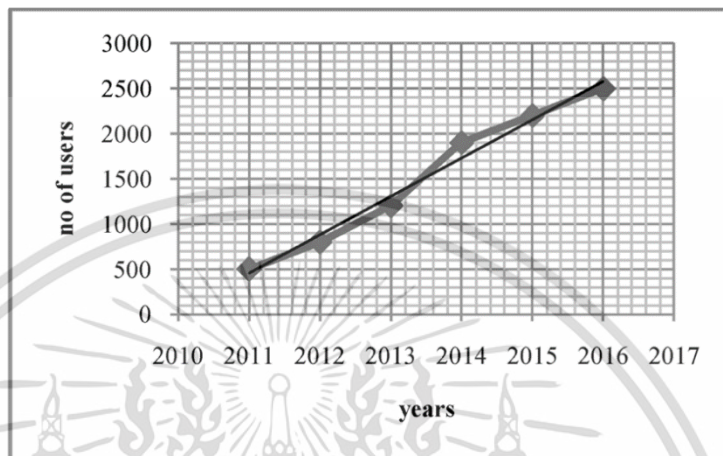
2.5.1 แนวคิด Conversational Commerce ในธุรกิจ E-Business

ธุรกิจแบบ E-commerce เป็นการสร้างประสบการณ์การซื้อขายสินค้าออนไลน์ให้กับทั้งผู้ซื้อและผู้ขายสินค้า โดย 52% ของประชากรเลือกที่จะซื้อสินค้าออนไลน์มากกว่าการไปซื้อสินค้าที่ห้างหรือร้านค้า ซึ่งในงานวิจัยของ Piyush, Choudhury และ Kumar จะพูดถึงการรวมแอปพลิเคชันเข้ากับการธุรกิจการขายสินค้า โดยการใช้ Conversational Commerce เป็นแพลตฟอร์มที่เข้าถึงง่าย ผู้ซื้อสามารถติดต่อสื่อสารกับทางร้านค้าได้โดยตรง ทั้งการขอความช่วยเหลือ สอบถามข้อสงสัยต่าง ๆ เกี่ยวกับสินค้า และทางร้านค้าเองก็จะได้ฟีดแบ็คและรีวิวกจากผู้ซื้อ เพื่อสร้างความน่าเชื่อถือให้กับร้านค้าเช่นเดียวกัน

Conversational Commerce เป็นการซื้อขายสินค้าแบบแชทออนไลน์ โดยผู้ซื้อ มีการพูดคุยกับแบรินด์ หรือผู้ขายผ่านแชทออนไลน์ ซึ่งจะนำไปสู่ขั้นตอนที่ผู้ซื้อตัดสินใจซื้อ และมีการยืนยันการสั่งของกับแบรินด์ หรือผู้ขายผ่านโปรแกรมแชท ซึ่งจากการสำรวจและ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ประโยชน์อื่นใดได้
วิเคราะห์ข้อมูลของ Piyush, Choudhury และ Kumar ในปี 2016 พบว่าผู้คนที่ส่วนมากไม่ทราบว่าใครคือผู้ขาย อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มักใช้แอปพลิเคชัน Facebook Messenger และ What's app ในการซื้อสินค้าออนไลน์มากที่สุด เนื่องจากแอปพลิเคชันทั้งสองเป็นแอปแชทออนไลน์ ซึ่งการซื้อสินค้าออนไลน์ผ่านแพลตฟอร์มนี้ ทำให้ผู้ซื้อรู้สึกว่าร้านค้ามีความน่าเชื่อถือ เข้าถึงง่าย และประหยัดเวลา ซึ่งการใช้ Conversational Commerce ในธุรกิจจะทำให้ผู้ขายสามารถเข้าใจผู้ซื้อมากยิ่งขึ้น



รูปที่ 2.13 กราฟแสดงการใช้งานแอปพลิเคชันแชท

จากรูปที่ 2.13 จะเห็นได้ว่ามีการใช้งานแอปพลิเคชันแชทเพิ่มมากขึ้น ซึ่งแสดงให้เห็นว่าการสื่อสารผ่านการแชทหรือส่งข้อความเป็นวิธีที่ดีและเป็นที่ต้องการใน E - markets แอปพลิเคชันแชทหลาย ๆ แอปได้อนุญาตให้แบรนด์สินค้าหรือบริษัทต่าง ๆ ใช้แอปพลิเคชันในการขายผลิตภัณฑ์ และลูกค้าก็สามารถเข้าถึงได้ง่ายมากยิ่งขึ้น โดยตัวอย่างแอปพลิเคชันมีดังรูปที่ 2.14

Application	Usage	Introduced in Year	Website Viability	User Preference	Associated with (Brand)
Facebook messenger	Fb chatting	2004	yes	High	Uber and many more
What's app	chatting	2010	No	High	Ola,Zomato and many more
We chat	chatting	2011	No	High	Online order by different sellers
Twitter	Post and news feed	2006	yes	High	Online order by different sellers
Slack	Bot chat and messaging	2014	No	Medium	Taco bell
Amazon echo	Bot device	2016	No	Medium	Alexa(information, and control)
Telegram	Chatting	2015	No	Low	Online order by different sellers
Apple Siri	Bot	2011	No	High	Different types of commands
Magic	For online shopping	2014	yes	Low	Takes any type of order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 2.14 ตารางเปรียบเทียบความแตกต่างของแต่ละแอปพลิเคชัน
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.14 จะเห็นได้ว่ามีแอปพลิเคชันหลายแอปที่ให้แบรนด์สินค้าหรือบริษัทใช้แอปพลิเคชันในการขายผลิตภัณฑ์ เช่น Facebook messenger ซึ่งบริษัทที่เข้ามาร่วมทำงานด้วยรายแรกคือ Uber โดย Uber จะให้ลูกค้าทำการจองคนขับ (Rider) ผ่านทางแอป Facebook messenger เพียงแอปเดียว โดยลูกค้าสามารถส่งโลเคชันและยืนยันการจองได้ภายในแอป ซึ่งในอนาคตลูกค้าสามารถที่จะใช้ messenger ในการ สั่งซื้อสินค้าออนไลน์ที่มีขอบเขตที่ใหญ่ขึ้น เป็นต้น

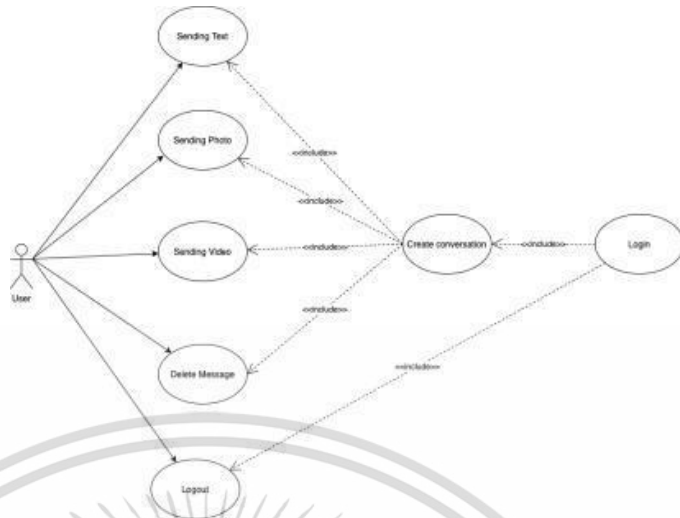
2.5.2 งานวิจัยที่เกี่ยวข้องกับการออกแบบการพัฒนาแอปพลิเคชันแชท

Chuong Tran (2021) ได้ทำวิจัยเกี่ยวกับการออกแบบการพัฒนาแอปพลิเคชันแชทบนระบบปฏิบัติการ iOS ซึ่งในการพัฒนาแอปพลิเคชันนี้เป็นแอปพลิเคชันที่สามารถใช้ได้บนอุปกรณ์ iPhone และ iPad โดยมีการทำงานหลัก ๆ คือ การลงชื่อเข้าใช้ (Login) การส่งข้อความตัวอักษร ข้อความรูปภาพ และข้อความที่เป็นวิดีโอ มีเทคโนโลยีที่ใช้ในการพัฒนา ได้แก่

- 1) Firebase ใช้ในการเก็บข้อมูล เพื่อให้แอปพลิเคชันสามารถเชื่อมต่อกับ Backend Cloud storage และ API
- 2) ภาษา Swift เป็นภาษาที่ใช้ในการพัฒนาแอปพลิเคชันสำหรับอุปกรณ์ของ Apple
- 3) Facebook Login SDK เป็นบริการของ Facebook ใช้ในการ Login เพื่อให้ผู้ใช้สามารถเข้าใช้งานแอปพลิเคชันผ่านแอคเคาท์ Facebook ได้ และแอปพลิเคชันจะได้ข้อมูลของผู้ใช้จาก Facebook เช่น ชื่อผู้ใช้ อีเมล เป็นต้น
- 4) Google Sign-in SDK เป็นบริการของ Google ใช้ในการ Login เพื่อให้ผู้ใช้สามารถเข้าใช้งานแอปพลิเคชันผ่านแอคเคาท์ Google เช่น ชื่อผู้ใช้ อีเมล เป็นต้น

ในงานวิจัยเกี่ยวกับการพัฒนาแอปพลิเคชันแชทบนระบบปฏิบัติการ iOS ได้มีการออกแบบแผนภาพการทำงานของผู้ใช้ระบบ (Use Case Diagram) และแผนภาพแสดงการปฏิสัมพันธ์ (Sequence Diagram) ของการส่งข้อความ ดังนี้

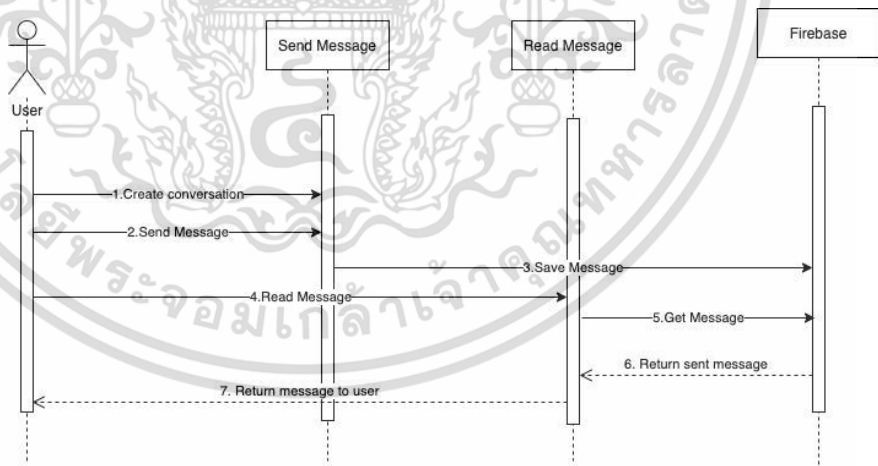
2.5.2.1 แผนภาพการทำงานของผู้ใช้ระบบ (Use Case Diagram)



รูปที่ 2.15 Use Case Diagramของการส่งข้อความ

จากรูปที่ 2.15 เป็น Use Case Diagram ของการส่งข้อความ โดยผู้ใช้งานจะสามารถส่งข้อความไปหาผู้ใช้รายอื่นได้ ต้องผ่านการเข้าสู่ระบบเพื่อยืนยันตัวตนก่อน แล้วหลังจากนั้นต้องสร้างการสนทนา หลังจากการสร้างการสนทนาเรียบร้อยแล้ว ผู้ใช้จะสามารถส่งข้อความตัวอักษร ข้อความรูปภาพ รวมไปถึงข้อความที่เป็นวิดีโอไปยังผู้ใช้รายอื่นได้

2.5.2.2 แผนภาพแสดงการปฏิสัมพันธ์ (Sequence Diagram)



รูปที่ 2.16 Sequence Diagram ของการส่งข้อความ

จากรูปที่ 2.16 เป็น Sequence Diagram ของผู้ใช้ระบบของการส่งข้อความ โดย เมื่อผู้ใช้ต้องการเริ่มการสนทนา ผู้ใช้ต้องสร้างการสนทนาใหม่ก่อน หลังจากนั้นจึงทำการส่งข้อความ ซึ่งข้อความที่ส่งไปจะถูกบันทึกไว้ในฐานข้อมูล หลังจากนั้นผู้ใช้สามารถอ่านข้อความได้โดยรับข้อความจาก Firebase

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อผู้ยืมเห็นหนังสือฉบับนี้ขอสงวนสิทธิ์ในการนำเอกสารนี้ไปใช้ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานวิจัย

ในการพัฒนาระบบการสนทนาของแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) เป็นการพัฒนาเพื่อเป็นทางเลือกให้ลูกค้าผู้ใช้บริการสามารถติดต่อสื่อสารระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้า ซึ่งมีวิธีการดำเนินงานดังนี้

3.1 การวางแผนและการวิเคราะห์ระบบ

การพัฒนาการสนทนาของแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ได้มีการวางแผนและวิเคราะห์ เพื่อพัฒนาระบบ โดยมีขั้นตอนดังนี้

3.1.1 ศึกษาความเป็นไปได้และเก็บรวบรวมข้อมูลที่เกี่ยวข้องกับการดำเนินงาน

- 1) ศึกษาความต้องการของผู้ใช้งาน เพื่อให้เข้าใจถึงจุดประสงค์ของการพัฒนาแอปพลิเคชัน เข้าใจความต้องการของผู้ใช้งาน รวมไปถึงวิธีการดำเนินงาน เพื่อนำข้อมูลที่ได้มาวางแผนการดำเนินงาน
- 2) เนื่องจากระบบที่จะพัฒนาเป็นระบบของแอปพลิเคชันบนระบบปฏิบัติการ iOS จึงจำเป็นต้องศึกษาข้อมูลเกี่ยวกับภาษา Swift เพื่อนำมาใช้ในการสร้างแอปพลิเคชันสำหรับ iOS, Mac, Apple TV และ Apple Watch ซึ่งภาษา Swift ถูกสร้างขึ้นเพื่อให้ใช้พัฒนาระบบ iOS โดยตรง
- 3) ศึกษาการใช้โปรแกรม Xcode ซึ่งเป็นเครื่องมือสำหรับนักพัฒนาโปรแกรมและแอปพลิเคชันบนแพลตฟอร์ม OS X และ iOS บนโทรศัพท์มือถือ เช่น แอปพลิเคชันบน iPhone

3.1.2 วิเคราะห์ความต้องการของระบบ (Requirement Collection and Analysis)

เมื่อได้ความต้องการของระบบมาแล้ว จึงได้มีการวิเคราะห์ความต้องการออกเป็น 2 ส่วน ดังนี้

1) Functional Requirement

Functional Requirement คือสิ่งที่ระบบควรที่จะทำ เพื่อตอบสนองความต้องการของผู้ใช้ ซึ่งระบบการสนทนามี Functional Requirement ดังนี้

- ผู้ใช้สามารถรับ - ส่งข้อความได้
- ผู้ใช้สามารถรับ - ส่งรูปภาพได้
- ผู้ใช้สามารถดูประวัติการสนทนาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมื่อคู่สนทนาได้เปิดอ่านข้อความแล้วให้ขึ้นสถานะ “อ่านแล้ว” ที่ข้อความ
- เมื่อมีข้อความใหม่จะมีการแจ้งเตือนผ่าน Notification
- ผู้ใช้สามารถกด Notification เพื่อไปยังหน้าจอสนทนาได้
- เมื่อมีข้อความใหม่และผู้ใช้งานยังไม่ได้ทำการเปิดอ่าน ปุ่มเข้าสู่ห้องสนทนาที่หน้ารายละเอียดคำสั่งซื้อ จะต้องแสดงจุดสีแดง
- หากมีข้อความที่ส่งไม่สำเร็จ จะทำการเก็บข้อความลง Local Database และเมื่ออินเทอร์เน็ตกลับมาใช้งานได้ จะทำการ ส่งข้อความนั้นใหม่อีกครั้ง

2) Non - Functional Requirement

เป็น Requirement ที่ใช้เพื่อเสริมให้ระบบสามารถทำงานได้สลับไหล และมีประสิทธิภาพมากขึ้น ซึ่งระบบการสนทนา มี Non - Functional Requirement ดังนี้

- เมื่อเวลาผ่านไป 24 ชั่วโมงหลังจากการส่งออเดอร์ ผู้ใช้จะไม่สามารถดูประวัติการสนทนา รับและส่งข้อความได้อีก

3.1.3 การวิเคราะห์ระบบงานเดิม

เนื่องจากระบบการสนทนาที่จะพัฒนาบนแอปพลิเคชันนั้น เป็นการพัฒนาระบบที่ทำงานอยู่บนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการที่มีอยู่ก่อนแล้ว ทำให้ผู้จัดทำจำเป็นต้องศึกษาเรียนรู้เกี่ยวกับกระบวนการทำงานของแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการ เพื่อให้สามารถเรียกใช้ฟังก์ชันการทำงาน รวมถึงเรียกใช้ข้อมูลที่เกี่ยวข้องและต่อยอดฟังก์ชันการทำงานที่มีอยู่ เพื่อพัฒนาระบบการสนทนาให้สมบูรณ์ โดยชุดคำสั่งที่ถูกพัฒนาต่อยอด และถูกเรียกใช้ในระบบการสนทนา มีดังนี้

3.1.3.1 ServiceAPI

เนื่องจากภายในแอปพลิเคชันได้มีการใช้ API ในการดึงข้อมูลจาก Server เพื่อนำข้อมูลมาแสดงผล ซึ่งการที่จะเรียก API ได้นั้น จะต้องเขียน Service และ Usecase เพื่อใช้ในการเรียก API ของระบบการทั้งหมด โดยใน Service จะประกอบไปด้วย 3 คลาส โดยชื่อคลาสจะเป็นชื่อที่ระบุว่าจะทำอะไร เช่น ระบบการสนทนา จะใช้ชื่อว่า Chat ก็จะประกอบไปด้วย 3 คลาส ดังนี้

- 1) Chat เป็นคลาสที่ใช้ในการระบุตัวแปรและข้อมูลที่ใช้ในการส่ง Request และ Response

- 2) ChatServiceAPI เป็นคลาสที่เก็บ Path URL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ChatService เป็นคลาสที่รวมกันระหว่างคลาส Chat กับคลาส ChatServiceAPI

ถัดมาก็จะเป็นในส่วนของ Usecase ซึ่ง Usecase จะนำไปใช้ใน ViewModel โดย Usecase จะประกอบไปด้วย 3 คลาส ดังนี้

- 1) ChatUseCases เป็นโปรโตคอลของ ChatUseCases ที่มีฟังก์ชันไว้สำหรับระบุ Request และ Response
- 2) ChatUseCasesModels เป็นคลาสสำหรับรับผล Response จากการต่อ API โดยจะรับมาจาก ChatService
- 3) DefaultContentUseCases เป็นคลาสรับรับการ Map ข้อมูลหลังจากที่ได้รับ Response โดยจะเป็นการระบุค่า Default ว่าเมื่อส่ง Request ไปแล้ว Response ที่ได้กลับมา success หรือ error

3.1.3.2 WebSocket

WebSocket เป็นคลาสกลางของแอปพลิเคชัน โดย WebSocketManager จะทำหน้าที่เกี่ยวกับการจัดการส่งและการรับ Event จากฝั่ง Back-End เพื่อใช้งานในการอัปเดตสถานะของคำสั่งซื้อ เช่น เมื่อลูกค้าผู้ใช้บริการได้ทำการสั่งซื้ออาหารแล้วทางผู้ประกอบการร้านค้าได้ทำการกดปฏิเสธคำสั่งซื้อ ตัวแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าจะส่ง Event เพื่ออัปเดตสถานการณ์สั่งซื้อให้ลูกค้าผู้ใช้บริการที่สั่งอาหารได้ทราบว่าทางผู้ประกอบการร้านค้าปฏิเสธคำสั่งซื้อ โดยคลาสดังกล่าวจะมีการเพิ่ม Event และ SocketURL เพื่อระบุที่อยู่ของ Socket ให้สอดคล้องกับที่ Back-End กำหนดขึ้นมา โดย Event ที่เพิ่มขึ้นมาจะใช้สำหรับการเข้าห้องสนทนา - ส่งข้อความและสถานะการอ่านข้อความ ซึ่ง Event เหล่านี้จะใช้สำหรับระบบการสนทนาโดยเฉพาะ

3.1.3.3 การ Upload Image

เนื่องจากการส่งรูปภาพจำเป็นต้องมีการอัปโหลดรูปภาพก่อน ซึ่งจะต้องอัปโหลดรูปขึ้นไปยัง CDN Service (Content Delivery Network Service) เพื่อให้แอปพลิเคชันในเครือข่ายสามารถแสดงผลภาพจาก URL ผ่านอินเทอร์เน็ตได้ โดยภายในแอปพลิเคชันจะใช้ในการอัปโหลดสลิปโอนเงิน หลังจากทำการสั่งซื้ออาหาร ซึ่งการอัปโหลดรูปภาพก็ใช้ API เช่นกัน ดังนั้นจึงสามารถเรียกใช้ได้ โดยการเรียกใช้จะเหมือนกับการเรียก API ของประวัติการสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การออกแบบ (Design)

ออกแบบระบบ (Design) เป็นขั้นตอนที่นำเอาปัญหาและความต้องการต่าง ๆ มาออกแบบ เพื่อใช้ในระบบงาน โดยแบ่งออกเป็นขั้นตอนย่อย ๆ ดังนี้

- 1) ออกแบบแผนภาพแสดงการทำงานของผู้ใช้ระบบ (Use Case Diagram) ใช้อธิบายความสามารถและผู้ที่เกี่ยวข้องทั้งหมดภายในระบบ
- 2) ออกแบบแผนภาพแสดงการปฏิสัมพันธ์ (Sequence Diagram) ใช้อธิบายลำดับเหตุการณ์ที่เกิดขึ้นระหว่างการใช้งานแอปพลิเคชัน
- 3) ออกแบบแผนภาพกิจกรรม (Activity Diagram) ใช้อธิบายกิจกรรมที่เกิดขึ้นในลักษณะกระแสการไหลของการทำงาน (Workflow)

3.1.5 การดำเนินงาน

การดำเนินงาน (Implementation) เป็นขั้นตอนในการเขียนโปรแกรม โดยมีวิธีการดำเนินงาน ดังนี้

- 1) ทำการเขียนโปรแกรมให้ตรงตาม Requirement
- 2) ทำการเขียนโปรแกรมให้ตรงตามที่ได้ออกแบบไว้ และพัฒนาระบบการสนทนาให้มีประสิทธิภาพ

3.1.6 การทดสอบระบบ

การทดสอบระบบ (Test) เป็นการทดสอบโปรแกรม โดยการทดสอบมีดังนี้

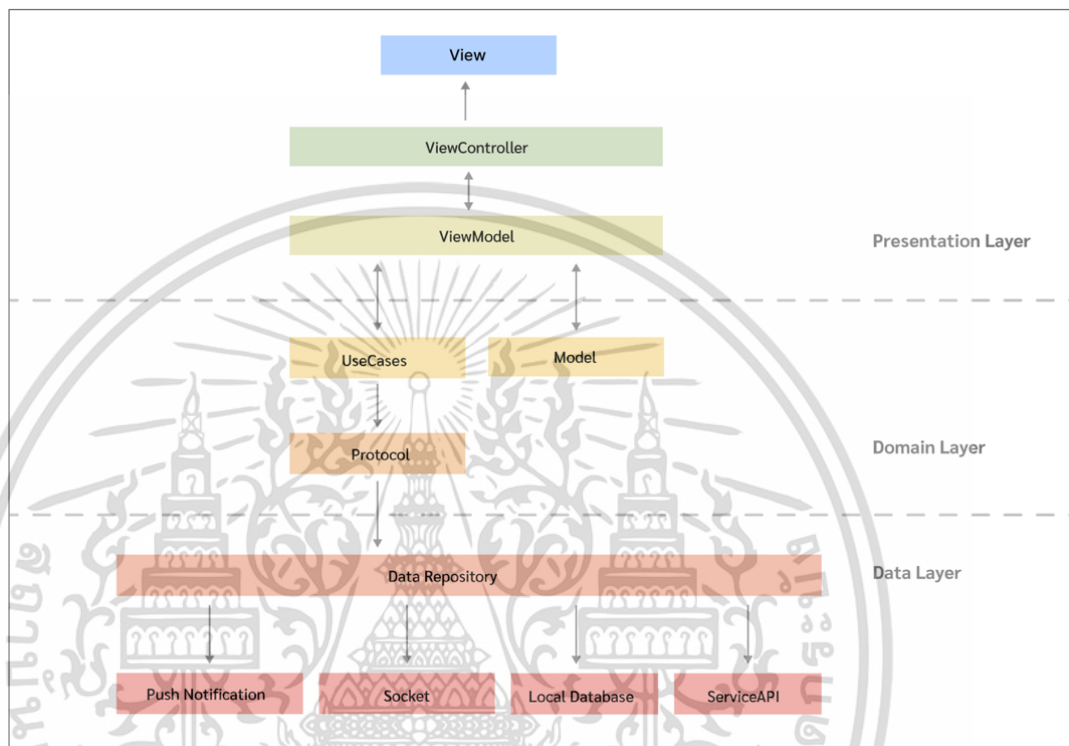
- 1) ทำการทดสอบการทำงานแต่ละฟังก์ชัน
- 2) ทำการทดสอบเคส Error ต่าง ๆ
- 3) ทำการทดสอบการส่งข้อความและการแสดงผลข้อความ
- 4) ทำการปรับปรุงแก้ไขข้อผิดพลาดที่พบเจอและพัฒนาโปรแกรมให้ดียิ่งขึ้น
- 5) หลังจากทำการตรวจสอบและไม่พบข้อบกพร่องใด ๆ ก็จะนำไปให้ทางทีม QA (Quality Assurance) ตรวจสอบและค้นหา Bug หรือ Defect หรือ Issue ให้ได้มากที่สุดเพื่อให้ Product มีคุณภาพ มากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ขั้นตอนการออกแบบระบบ

3.2.1 แผนภาพแสดงขั้นตอนการทำงานและส่วนต่าง ๆ ของระบบ

ระบบการสนทนานี้เป็นส่วนที่ทำต่อเนื่องมาจากการส่งรายการสั่งซื้อ ซึ่งในแอปพลิเคชันมีการแบ่งงานออกเป็นขั้นตอนการทำงาน โดยระบบการสนทนาจะมีขั้นตอนการทำงาน ดังนี้



รูปที่ 3.1 แผนภาพแสดงขั้นตอนการทำงานของระบบ

จากรูปที่ 3.1 เป็นแผนภาพที่แสดงขั้นตอนการทำงานของระบบ จะเห็นได้ว่าการทำงานออกเป็นชั้น ๆ ซึ่งจะอธิบายความสัมพันธ์ชั้นการทำงานและส่วนต่าง ๆ ของแต่ละชั้นทำงานประสานกัน โดยในแต่ละชั้นจะมีองค์ประกอบดังต่อไปนี้

3.2.1.1 Presentation Layer

Presentation Layer เป็นชั้นของส่วนที่ทำหน้าที่เกี่ยวกับการแสดงผล การตอบโต้กับผู้ใช้ ซึ่งจะประกอบไปด้วย

- 1) View มีหน้าที่ในการแสดงผลบนหน้าจอ ซึ่งในส่วนนี้จะใช้ในการแสดงผลส่วนประกอบต่าง ๆ ของหน้าห้องสนทนา
- 2) ViewController มีหน้าที่ควบคุมการแสดงผล และจะมีการรับและส่งข้อมูลไปมากับ ViewModel และถ้าหากจะต้องทำการเปลี่ยนแปลงการแสดงผลบนหน้าจอ ViewController จะไปเรียก View เพื่อให้ทำการแสดงผล โดยในระบบการสนทนา ViewController จะทำการเรียกหน้า View เพื่อให้ทำการแสดงผลชื่อร้านค้า หรือกรณีที่รายการ

สั่งซื้อเกิน 24 ชั่วโมง ก็จะเรียก View ให้ทำการแสดงผลหน้า Empty State รวมไปถึงมีหน้าที่ในการจัดการการแสดงผลข้อความทั้งหมด

- 3) ViewModel มีหน้าที่ทำในส่วนของ Logic ต่าง ๆ และเป็นตัวกลางที่มีการเชื่อมต่อกับชั้น Domain Layer เพื่อทำการรับข้อมูลที่มาจกชั้น Domain Layer มาใช้ในการประมวลผลแล้วทำการส่งต่อข้อมูลเพื่อแสดงผล เช่น ViewModel จะนำข้อมูลรายละเอียดคำสั่งซื้อมาประมวลผลว่าการสั่งซื้อมีระยะเวลาเกิน 24 ชั่วโมงหรือไม่ และมีการรับข้อมูลที่มาจก Usecases เช่น ข้อมูลจก API มาใช้ในแบ่งข้อมูลข้อความออกตามวันที่ที่ส่งข้อความ

3.2.1.2 Domain Layer

Domain Layer เป็นชั้นที่ของส่วนที่ทำหน้าที่เกี่ยวกับการจัดการข้อมูล และเป็นตัวกลางของข้อมูล ซึ่งจะประกอบไปด้วย

- 1) Model มีหน้าที่ในการเก็บข้อมูลที่มีการ Input และ Output ใน ViewModel
- 2) Usecases มีหน้าที่รวมข้อมูลที่ได้รับที่มาจาก API ซึ่ง ViewModel ก็จะเป็นตัวเรียก Usecases เพื่อนำข้อมูลไปใช้งานต่อ
- 3) Protocol มีหน้าที่ในการกำหนดทิศทางและหน้าที่การทำงานของแต่ละคลาส

3.2.1.3 Data Layer

Data Layer เป็นส่วนของข้อมูลดิบ (Raw Data) หรือข้อมูลที่ไม่มีการเปลี่ยนแปลง

- 3.2.1.3.1 Data Repository มีหน้าที่ในการเก็บข้อมูลต่าง ๆ ที่มาจากหลังบ้าน และ API โดย Data Repository จะทำการเรียกข้อมูลจากส่วนต่าง ๆ ดังนี้

- 1) Push Notification มีหน้าที่ในการแสดงแจ้งเตือนทั้งเสียงและแสดงผลข้อมูลบนหน้าจอ ซึ่งจะใช้ในการแจ้งเตือนข้อความที่ผู้ประกอบการร้านค้าได้ทำการส่งข้อความมายังลูกค้าผู้ใช้บริการ
- 2) Socket มีหน้าที่ในการรับและส่ง Event ต่าง ๆ พร้อมข้อมูลที่ใช้ในการติดต่อสื่อสาร ซึ่ง Socket จะใช้ในการรับและส่งข้อความ และการทำงานอื่น ๆ ของระบบการสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

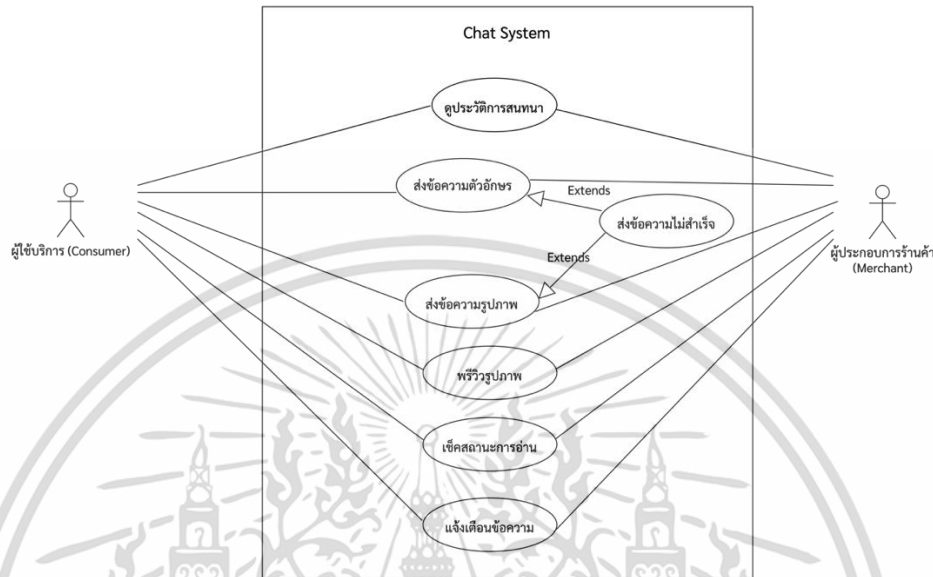
- 3) Local Database เป็นฐานข้อมูลที่มีหน้าที่ในการเก็บข้อมูลลงบนแหล่งเก็บข้อมูลภายในของอุปกรณ์นั้น ๆ ซึ่งในระบบการสนทนาจะใช้ในการเก็บข้อมูลข้อความที่ส่งไม่สำเร็จเนื่องจากอินเทอร์เน็ตมีปัญหา และเมื่ออินเทอร์เน็ตกลับมาใช้งานได้ ผู้ใช้งานจะสามารถเห็นข้อความดังกล่าวบนหน้าจอสนทนา
- 4) ServiceAPI เป็นที่เก็บข้อมูลที่ใช้ในการส่ง Request และ Response ที่จะได้รับกลับมาจาก API และมีการเก็บ Path ที่ใช้ในการยิง API โดยในระบบการสนทนาจะใช้ API ในการขอข้อมูลประวัติการสนทนาและอัปโหลดรูปภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 แผนภาพแสดงการทำงานของผู้ใช้ระบบ (Use Case Diagram)

ระบบการสนทนาเป็นระบบที่ใช้สำหรับสนทจาระหว่างลูกค้าผู้ใช้บริการ กับผู้ประกอบการร้านค้าซึ่งลูกค้าผู้ใช้บริการสามารถที่จะส่งข้อความต่าง ๆ ไปยังผู้ประกอบการร้านค้าได้ โดยการทำงานของระบบ มีแนวทางดังนี้



รูปที่ 3.2 Use Case Diagram

จาก Use Case Diagram รูปที่ 3.2 ระบบการสนทนาจะเริ่มจากเมื่อลูกค้าผู้ใช้บริการมีการสั่งอาหารเรียบร้อยแล้ว ลูกค้าผู้ใช้บริการสามารถเข้าไปยังห้องสนทนา โดยเมื่อเข้าไปในห้องสนทนาแล้ว ลูกค้าผู้ใช้บริการจะเห็นประวัติการสนทจาระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้า โดยประวัติการสนทนาจะขึ้นอยู่กับเลขที่รายการสั่งซื้อ จากนั้นลูกค้าผู้ใช้บริการสามารถที่จะส่งข้อความได้ ทั้งข้อความที่เป็นตัวอักษรและข้อความที่เป็นรูปภาพ เมื่อข้อความถูกส่งสำเร็จ ทางฝั่งของผู้ประกอบการร้านค้าจะได้รับข้อความนั้น ๆ กรณีที่ข้อความที่ลูกค้าผู้ใช้บริการทำการส่งไปยังผู้ประกอบการร้านค้าและข้อความที่ลูกค้าผู้ใช้บริการได้รับเป็นข้อความรูปภาพ ลูกค้าผู้ใช้บริการสามารถที่จะกดที่รูปภาพเพื่อดูรูปได้แบบเต็มจอ ในกรณีที่ลูกค้าผู้ใช้บริการทำการส่งข้อความที่ผู้ประกอบการร้านค้า แต่ทางฝั่งลูกค้าผู้ใช้บริการเกิดปัญหาด้านอินเทอร์เน็ต ทำให้ข้อความส่งไม่สำเร็จ ข้อความนั้นจะถูกเก็บลงบน Local Database และเมื่ออินเทอร์เน็ตกลับมาใช้งานได้ ข้อความที่ส่งไม่สำเร็จจะทำการส่งใหม่อัตโนมัติ อีกทั้งลูกค้าผู้ใช้บริการสามารถตรวจสอบสถานะการอ่านของข้อความที่ได้ทำการส่งไปยังผู้ประกอบการร้านค้าได้ โดยถ้าผู้ประกอบการร้านค้าได้ทำการเปิดอ่านข้อความแล้ว ข้อความทางฝั่งลูกค้าผู้ใช้บริการจะมีสถานะขึ้นให้รู้ว่าข้อความมีสถานะว่าอ่านแล้ว และเมื่อลูกค้าผู้ใช้บริการได้รับข้อความใหม่จากทางผู้ประกอบการร้านค้า ลูกค้าผู้ใช้บริการจะมีการแจ้งเตือนแบบ Push Notification เพื่อแจ้งเตือนให้ลูกค้าผู้ใช้บริการทราบว่าข้อความใหม่เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 Use Case Description

Use Case Description เป็นการอธิบายรายละเอียดของ Use Case ซึ่งการอธิบายรายละเอียดของ Use Case ระบบการสนทนา สามารถอธิบายได้ดังนี้

ตารางที่ 3.1 Use Case Description ของส่งข้อความตัวอักษร

Name	ส่งข้อความตัวอักษร
Description	ลูกค้าผู้ใช้บริการสามารถส่งข้อความที่เป็นตัวอักษร รวมไปถึงอิโมจิไปยังผู้ประกอบการร้านค้าได้
Input	ข้อความตัวอักษร และอิโมจิ
Output	ข้อความตัวอักษร และอิโมจิ เวลาที่ลูกค้าผู้ใช้บริการทำการส่งข้อความ และสถานะการอ่าน
Condition	ข้อความตัวอักษรที่ลูกค้าผู้ใช้บริการทำการส่งไปต้องไม่เกิน 128 ตัวอักษร

ตารางที่ 3.2 Use Case Description ของส่งข้อความรูปภาพ

Name	ส่งข้อความรูปภาพ
Description	ลูกค้าผู้ใช้บริการสามารถส่งข้อความที่เป็นรูปภาพไปยังผู้ประกอบการร้านค้าได้
Input	รูปภาพ
Output	รูปภาพ เวลาที่ลูกค้าผู้ใช้บริการทำการส่งข้อความ และสถานะการอ่าน
Condition	ต้องมีการเชื่อมต่ออินเทอร์เน็ต

ตารางที่ 3.3 Use Case Description ของพรีวิวรูปภาพ

Name	พรีวิวรูปภาพ
Description	ลูกค้าผู้ใช้บริการสามารถกดที่รูปภาพเพื่อดูรูปภาพแบบเต็มจอได้
Input	รูปภาพ
Output	รูปภาพขนาดขยายเต็มจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 Use Case Description ของดูประวัติการสนทนา

Name	ดูประวัติการสนทนา
Description	ลูกค้าผู้ใช้บริการสามารถดูประวัติการสนทนายาระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้า โดยประวัติการสนทนาจะเป็นประวัติการสนทนาของห้องสนทนานั้น ๆ
Input	เข้าไปยังหน้าห้องสนทนา
Output	ประวัติการสนทนา
Condition	จะต้องเข้าไปยังห้องสนทนาห้องนั้นก่อน และเวลาที่ทำการสั่งซื้อต้องไม่เกิน 24 ชั่วโมง

ตารางที่ 3.5 Use Case Description ของเช็คสถานะการอ่าน

Name	เช็คสถานะการอ่าน
Description	ลูกค้าผู้ใช้บริการสามารถตรวจสอบสถานะการอ่านของข้อความที่ได้ทำการส่งไปยังผู้ประกอบการร้านค้าได้ โดยถ้าผู้ประกอบการร้านค้าได้ทำการเปิดอ่านข้อความแล้ว ข้อความทางฝั่งลูกค้าผู้ใช้บริการจะมีสถานะขึ้นให้รู้ว่าข้อความมีสถานะว่าอ่านแล้ว
Input	
Output	สถานะการอ่านของข้อความ โดยจะขึ้นข้อความว่า “อ่านแล้ว”
Condition	กรณีที่ข้อความล่าสุดขึ้นสถานะว่าอ่านแล้ว ข้อความก่อนหน้าทั้งหมดจะต้องมีสถานะว่าอ่านแล้วเช่นเดียวกัน

ตารางที่ 3.6 Use Case Description ของแจ้งเตือนข้อความ

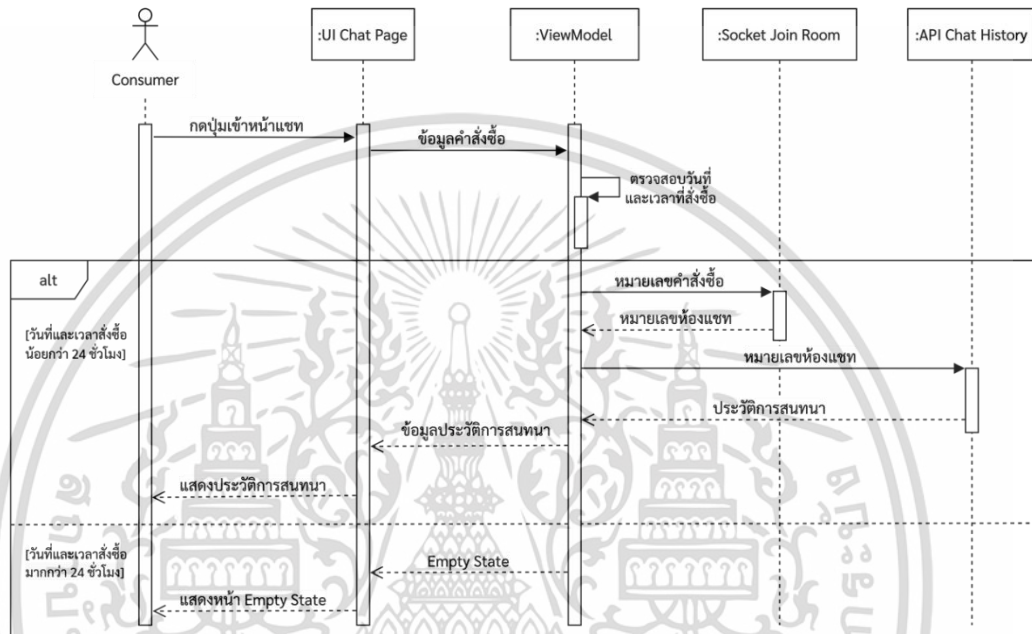
Name	แจ้งเตือนข้อความ
Description	ลูกค้าผู้ใช้บริการทำการส่งข้อความไปยังผู้ประกอบการร้านค้าสำเร็จ ทางผู้ประกอบการร้านค้าจะได้รับการแจ้งเตือนข้อความ (Notification) และถ้าผู้ประกอบการร้านค้าส่งข้อความมายังลูกค้าผู้ใช้บริการ ทางฝั่งลูกค้าผู้ใช้บริการก็จะได้รับการแจ้งเตือนข้อความ (Notification) โดยลูกค้าผู้ใช้บริการ สามารถกด Notification เพื่อไปยังห้องสนทนานั้นได้
Input	ผู้ประกอบการร้านค้าทำการส่งข้อความมายังลูกค้าผู้ใช้บริการ
Output	แสดงผลแจ้งเตือนข้อความว่าได้รับข้อความใหม่จากทางผู้ประกอบการร้านค้าที่สนทนาอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แผนภาพแสดงถึงการปฏิสัมพันธ์ (Sequence Diagram)

Sequence Diagram เป็นแผนภาพที่จะแสดงให้เห็นถึงการทำงานที่เชื่อมต่อกันระหว่างอ็อบเจ็กต์ของคลาส โดยในแผนภาพจะแสดงถึงลำดับเหตุการณ์ที่เกิดขึ้นระหว่างการใช้งานแอปพลิเคชัน ซึ่งสามารถเขียน Sequence Diagram ได้ดังนี้

3.2.3.1 Sequence Diagram ของการดูประวัติการสนทนา

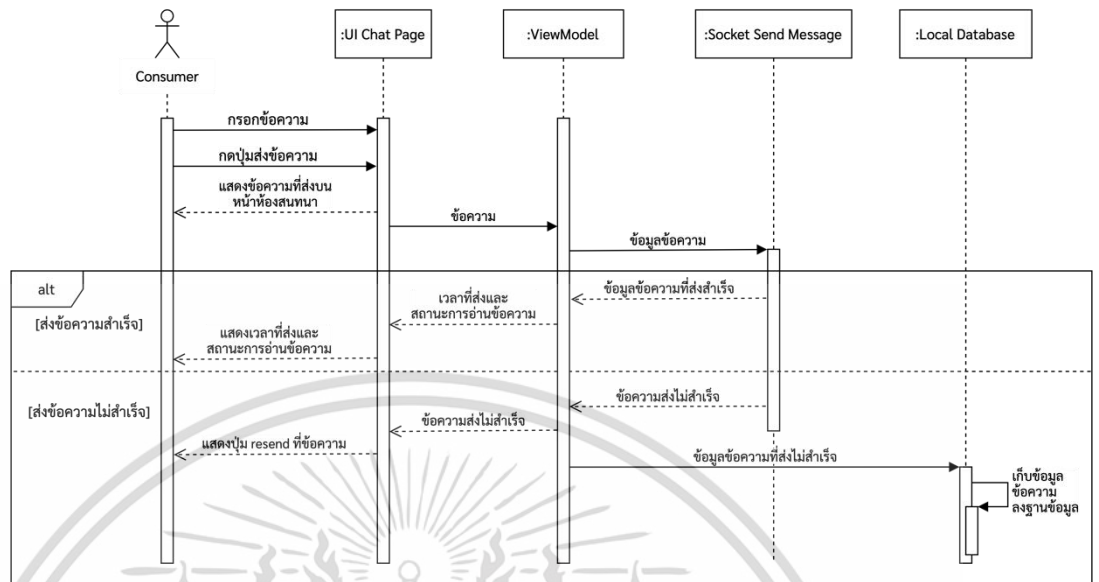


รูปที่ 3.3 Sequence Diagram ของการดูประวัติการสนทนา

จากรูปที่ 3.3 เป็น Sequence Diagram ของการดูประวัติการสนทนา โดยเริ่มต้นจากลูกค้าผู้ใช้บริการกดปุ่มเพื่อไปยังหน้าห้องสนทนาผ่านทางหน้ารายละเอียดคำสั่งซื้อ ซึ่งเมื่อเข้าไปยังหน้าห้องสนทนาแล้ว จะมีการส่งข้อมูลคำสั่งซื้อไปที่ ViewModel เพื่อทำการตรวจสอบวันที่และเวลาที่ทำการสั่งซื้อ หากวันที่และเวลาที่ทำการสั่งซื้อมีระยะเวลาไม่เกิน 24 ชั่วโมง ViewModel จะส่งหมายเลขคำสั่งซื้อไปยัง Socket Join Room และทาง Socket จะมีการคืนค่าหมายเลขห้องสนทนากลับมา เพื่อนำหมายเลขห้องสนทนาไปใช้ในการขอประวัติการสนทนาจาก API และ API จะส่งประวัติการสนทนากลับมายัง ViewModel และ ViewModel จะส่งข้อมูลประวัติการสนทนาไปยังหน้าห้องสนทนา เพื่อแสดงประวัติการสนทนาทั้งหมดให้ลูกค้าผู้ใช้บริการ แต่ถ้าวันที่และเวลาที่ทำการสั่งซื้อมีระยะเวลาเกิน 24 ชั่วโมง ViewModel จะคืนค่าเพื่อให้หน้าห้องสนทนาแสดงหน้า Empty State โดยหน้า Empty State นี้ ลูกค้าผู้ใช้บริการจะไม่สามารถดูประวัติการสนทนาและส่งข้อความได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2 Sequence Diagram ของการส่งข้อความตัวอักษร

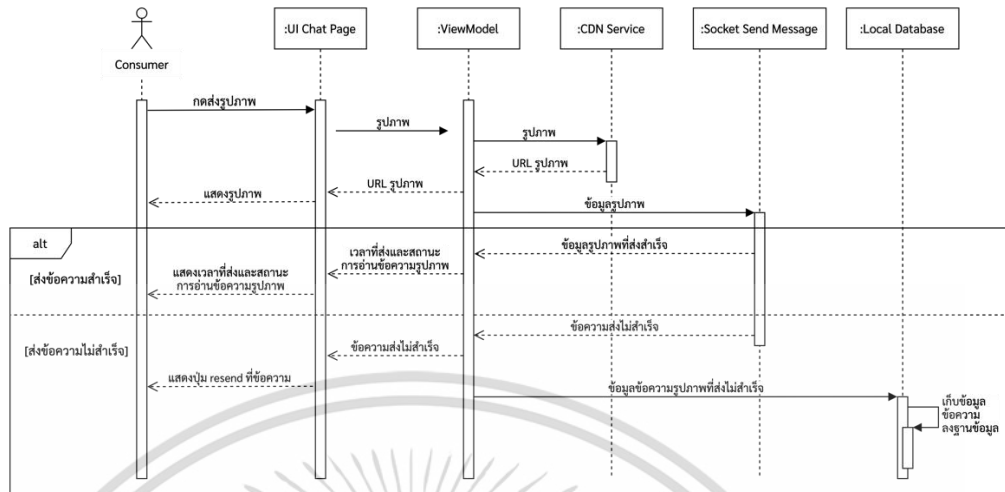


รูปที่ 3.4 Sequence Diagram ของการส่งข้อความตัวอักษร

จากรูปที่ 3.4 เป็น Sequence Diagram ของการส่งข้อความ โดยเริ่มจากเมื่อลูกค้าผู้ใช้บริการอยู่ในห้องสนทนาแล้ว ลูกค้าผู้ใช้บริการสามารถกรอกข้อความและกดปุ่มส่งเพื่อส่งข้อความที่หน้าห้องสนทนา และหน้าห้องสนทนาจะแสดงข้อความที่ได้ทำการส่ง และได้มีการส่งข้อความไปยัง ViewModel เพื่อให้ ViewModel ทำการส่งข้อมูลข้อความไปยังผู้ประกอบการร้านค้า โดยผ่านทาง Socket Send Message ถ้าหากว่าข้อความที่ส่งไปถูกส่งสำเร็จ ทาง Socket จะส่งข้อมูลข้อความที่ส่งสำเร็จกลับมายัง ViewModel และหลังจากนั้น ViewModel จะส่งเวลาที่ส่งข้อความและสถานะการอ่านข้อความไปยังหน้าห้องสนทนา เพื่อแสดงผลเวลาและสถานะการอ่านข้อความให้กับลูกค้าผู้ใช้บริการ แต่ในกรณีที่อินเทอร์เน็ตมีปัญหา ข้อความจะส่งไปยังผู้ประกอบการร้านค้าไม่สำเร็จ ซึ่งในกรณีนี้ Socket จะคืนค่ากลับมายัง ViewModel ว่าข้อความส่งไม่สำเร็จ และ ViewModel จะส่งค่ากลับไปยังหน้าห้องสนทนาเพื่อให้หน้าห้องสนทนาแสดงปุ่ม Resend ที่ข้อความ ซึ่งปุ่ม Resend จะทำให้ลูกค้าผู้ใช้บริการรู้ว่าข้อความดังกล่าวส่งไปยังผู้ประกอบการร้านค้าไม่สำเร็จ และ ViewModel จะทำการเก็บข้อความที่ส่งไม่สำเร็จนั้นลงใน Local Database

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.3 Sequence Diagram ของการส่งข้อความรูปภาพ

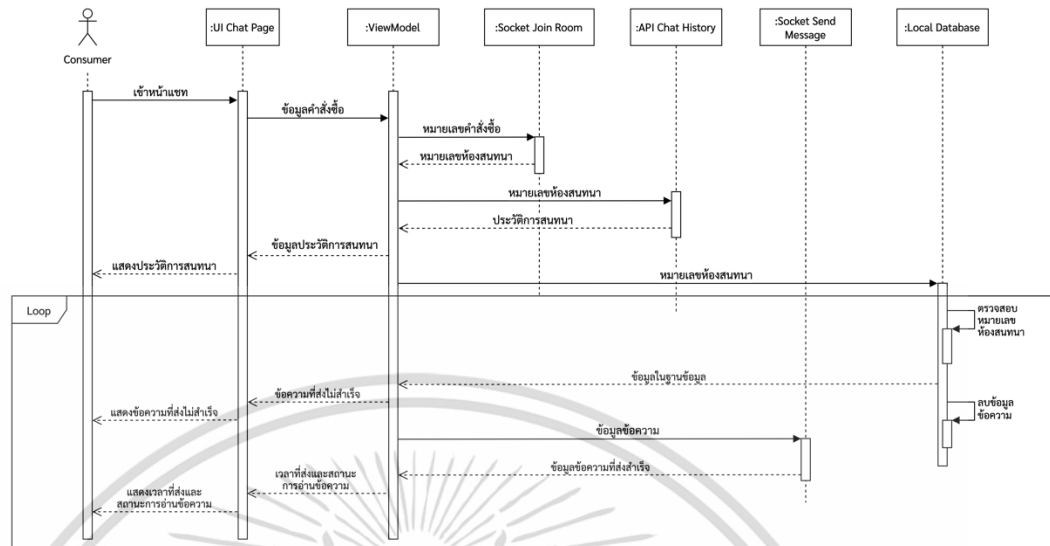


รูปที่ 3.5 Sequence Diagram ของการส่งข้อความรูปภาพ

จากรูปที่ 3.5 เป็น Sequence Diagram ของการส่งข้อความรูปภาพ โดยเริ่มจากเมื่อลูกค้าผู้ใช้บริการอยู่ในห้องสนทนา และลูกค้าผู้ใช้บริการต้องการที่จะส่งข้อความที่เป็นรูปภาพ โดยลูกค้าผู้ใช้บริการสามารถส่งข้อความที่เป็นรูปภาพได้ จากการเลือกรูปภาพในคลังรูปภาพ หรือลูกค้าผู้ใช้บริการสามารถถ่ายรูปเพื่อส่งไปยังผู้ประกอบการร้านค้าได้ โดยเมื่อลูกค้าผู้ใช้บริการกดปุ่มยืนยันเพื่อส่งรูปภาพแล้ว จะส่งรูปภาพไปยัง ViewModel เพื่อนำรูปภาพนั้นไปอัปโหลดบน CDN Service เพื่อให้รูปภาพสามารถที่จะแสดงผ่านอินเทอร์เน็ตได้ เมื่ออัปโหลดรูปภาพเสร็จเรียบร้อยแล้ว CDN Service จะส่ง URL ของรูปภาพกลับมาที่ ViewModel เพื่อให้ ViewModel ส่ง URL ของรูปภาพไปแสดงภาพที่หน้าห้องสนทนาของลูกค้าผู้ใช้บริการ หลังจากนั้น ViewModel จะทำการส่งข้อมูลข้อความรูปภาพไปยังผู้ประกอบการร้านค้า โดยผ่านทาง Socket Send Message ถ้าหากว่าข้อความรูปภาพที่ส่งไปถูกส่งสำเร็จ ทาง Socket จะส่งข้อมูลข้อความรูปภาพที่ส่งสำเร็จกลับมายัง ViewModel และหลังจากนั้น ViewModel จะส่งเวลาที่ส่งข้อความและสถานะการอ่านข้อความไปยังหน้าห้องสนทนา เพื่อแสดงผลเวลาและสถานะการอ่านข้อความให้กับลูกค้าผู้ใช้บริการ แต่ในกรณีที่อินเทอร์เน็ตมีปัญหา ข้อความรูปภาพจะส่งไปยังผู้ประกอบการร้านค้าไม่สำเร็จ ซึ่งในกรณีนี้ Socket จะคืนค่ากลับมายัง ViewModel ว่าข้อความส่งไม่สำเร็จ และ ViewModel จะส่งค่ากลับไปยังหน้าห้องสนทนา เพื่อให้หน้าห้องสนทนาแสดงปุ่ม Resend ที่ข้อความ ซึ่งปุ่ม Resend จะทำให้ลูกค้าผู้ใช้บริการรู้ว่าข้อความดังกล่าวส่งไปยังผู้ประกอบการร้านค้าไม่สำเร็จ และ ViewModel จะทำการเก็บข้อความที่ส่งไม่สำเร็จนั้นลงใน Local Database

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานโดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีก Local Database ส่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.4 Sequence Diagram ของการส่งข้อความที่ส่งไม่สำเร็จ



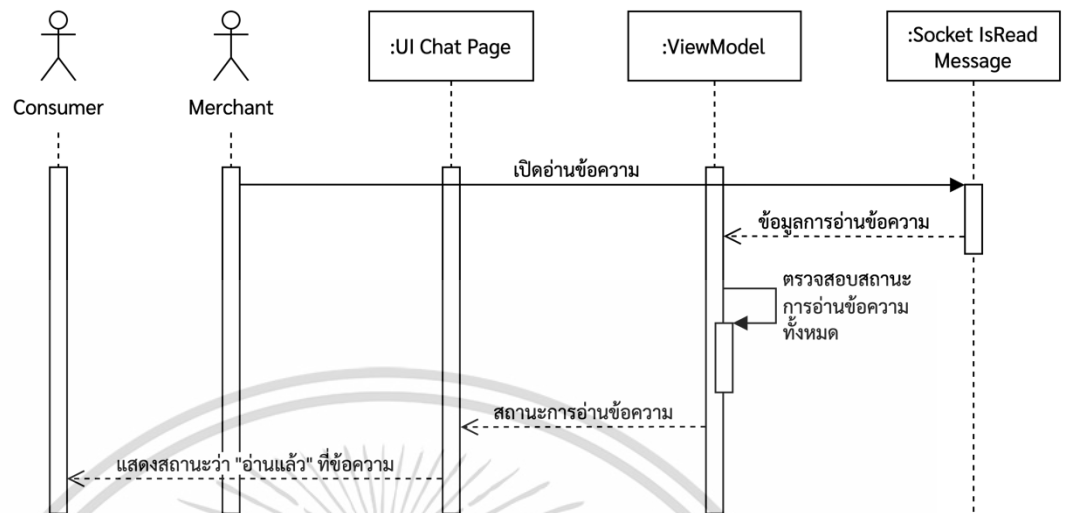
รูปที่ 3.6 Sequence Diagram ของการส่งข้อความที่ส่งไม่สำเร็จ

จากรูปที่ 3.6 เป็น Sequence Diagram ของการส่งข้อความที่ส่งไม่สำเร็จ ซึ่งในกรณีนี้จะเกิดได้จากก่อนหน้านั้นทางลูกค้าผู้ใช้บริการได้ทำการส่งข้อความไปแล้ว แต่อินเทอร์เน็ตเกิดมีปัญหาระหว่างการส่ง ทำให้ข้อความส่งไปยังผู้ประกอบการร้านค้าไม่สำเร็จ ซึ่งเมื่ออินเทอร์เน็ตกลับมาใช้งานได้ ลูกค้าผู้ใช้บริการจะเริ่มทำการเข้าห้องสนทนาใหม่ และจะมีการส่งข้อมูลคำสั่งซื้อไปที่ ViewModel เพื่อนำข้อมูลคำสั่งซื้อไปเชื่อมต่อกับ Socket Join Room และ Socket จะมีการคืนค่าหมายเลขห้องสนทนากลับมา ซึ่งหมายเลขห้องสนทนาที่ได้จะเป็นหมายเลขเดิม และจะนำหมายเลขห้องสนทนาไปใช้ในการขอข้อมูลประวัติการสนทนาของห้องสนทนานั้น ๆ เมื่อทางลูกค้าผู้ใช้บริการได้รับประวัติการสนทนาทั้งหมดแล้ว ViewModel จะนำหมายเลขห้องสนทนาไปตรวจสอบกับข้อมูลใน Local Database ว่ามีข้อความที่ส่งไม่สำเร็จหรือไม่ ซึ่งถ้าหากมีข้อมูลใน Local Database และหมายเลขห้องสนทนาตรงกับข้อมูลใน Local Database ข้อมูลใน Local Database จะถูกส่งไปยัง ViewModel และข้อมูลนั้นจะถูกลบออกจาก Local Database จากนั้นใน ViewModel จะส่งข้อความที่ส่งไม่สำเร็จไปแสดงบนหน้าห้องสนทนา จากนั้น ViewModel จะทำการส่งข้อมูลข้อความที่ส่งไม่สำเร็จไปยังผู้ประกอบการร้านค้าใหม่อีกครั้ง โดยผ่านทาง Socket Send Message ถ้าหากว่าข้อความที่ส่งไปถูกส่งสำเร็จ ทาง Socket จะส่งข้อมูลข้อความที่ส่งสำเร็จกลับมายัง ViewModel และหลังจากนั้น ViewModel จะส่งเวลาที่ส่งข้อความและสถานะการ

อ่านข้อความไปยังหน้าห้องสนทนา เพื่อแสดงผลเวลาและสถานะการอ่านข้อความให้กับลูกค้าผู้ใช้บริการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งทางผู้ให้บริการต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

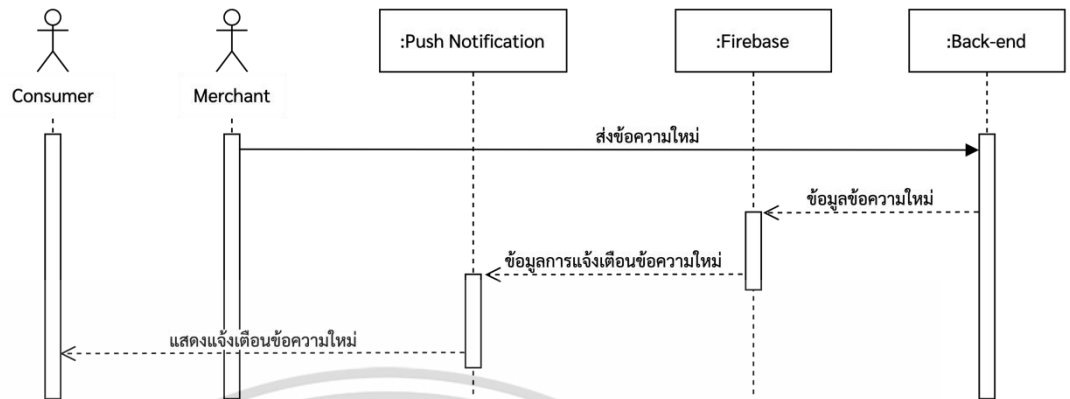
3.2.3.5 Sequence Diagram ของการเช็คสถานะการอ่าน



รูปที่ 3.7 Sequence Diagram ของการเช็คสถานะการอ่าน

จากรูปที่ 3.7 เป็น Sequence Diagram ของการเช็คสถานะการอ่านของข้อความ โดยการที่ข้อความบนหน้าห้องสนทนาของลูกค้าผู้ใช้บริการจะมีสถานะขึ้นว่า “อ่านแล้ว” จะเกิดขึ้นได้จากลูกค้าผู้ใช้บริการได้ทำการส่งข้อความไปยังทางผู้ประกอบการร้านค้า แล้วทางผู้ประกอบการร้านค้าได้ทำการเปิดอ่านข้อความ ผู้ประกอบการร้านค้าจะมีการส่งข้อมูลไปที่ Socket IsRead Message ซึ่งเป็น Socket ที่ส่งข้อมูลเกี่ยวกับสถานะการอ่านข้อความ จากนั้น ViewModel จะรับข้อมูลการอ่านข้อความมาจาก Socket ซึ่งภายในข้อมูลการอ่านข้อความจะมีข้อมูลหมายเลขห้องสนทนา สถานะการอ่านและผู้ที่อ่านข้อความ โดย ViewModel จะนำข้อมูลเหล่านี้มาทำการตรวจสอบว่า หมายเลขห้องสนทนาตรงกันหรือไม่ ซึ่งหากตรวจสอบแล้วว่าหมายเลขห้องตรงกัน และผู้ส่งคือผู้ประกอบการร้านค้า ส่วนของ ViewModel จะทำการเปลี่ยนสถานะการอ่าน และแสดงสถานะการอ่านที่ข้อความทั้งหมดที่ลูกค้าผู้ใช้บริการเป็นผู้ส่ง โดยจะแสดงข้อความว่า “อ่านแล้ว” ด้านบนของเวลาที่ทำการส่งของข้อความนั้น ๆ

3.2.3.6 Sequence Diagram ของการแจ้งเตือนข้อความ



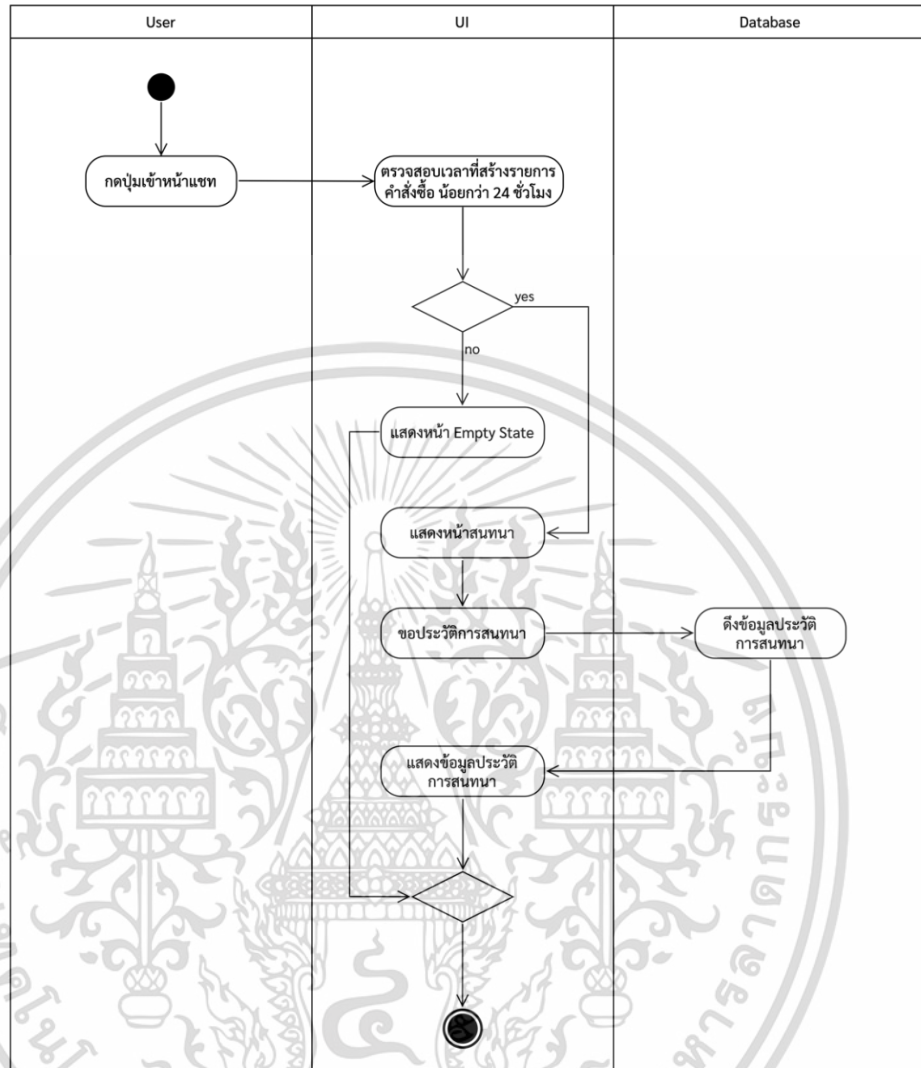
รูปที่ 3.8 Sequence Diagram ของการแจ้งเตือนข้อความ

จากรูปที่ 3.8 เป็น Sequence Diagram ของการแจ้งเตือนข้อความ โดยลูกค้าผู้ใช้บริการจะได้รับการแจ้งเตือนจากการที่ผู้ประกอบการร้านค้าเป็นผู้ส่งข้อความ ซึ่งผู้ประกอบการร้านค้าจะมีการส่งข้อความไปยังหลังบ้าน (Back-end) และหลังบ้าน (Back-end) จะทำการส่งข้อมูลข้อความไปที่ Firebase และ Firebase ส่งข้อมูลการแจ้งเตือนข้อความไปให้ Push Notification จัดการข้อมูลในการแจ้งเตือนข้อความ โดยภายใน Push Notification จะประกอบไปด้วยชื่อของร้านค้าที่เป็นผู้ส่งข้อความ และข้อความที่ผู้ประกอบการร้านค้าได้ส่งมา เพื่อแสดงแจ้งเตือนและข้อความใหม่ที่ผู้ประกอบการร้านค้าส่งให้กับลูกค้าผู้ใช้บริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 แผนภาพกิจกรรม (Activity Diagram)

3.2.4.1 Activity Diagram ของการดูประวัติการสนทนา



รูปที่ 3.9 Activity Diagram ของการดูประวัติการสนทนา

จากรูปที่ 3.9 เป็นแผนภาพกิจกรรมของการดูประวัติการสนทนา โดยการที่ลูกค้าผู้ใช้งานจะสามารถดูประวัติการสนทนาได้ ผู้ใช้งานจะต้องทำการกดปุ่มเข้าหน้าห้องสนทนาจากหน้ารายการคำสั่งซื้อ ซึ่งเมื่อกดปุ่มเพื่อเข้าหน้าห้องสนทนาแล้วระบบจะทำการตรวจสอบวันที่และเวลาที่ทำการสั่งซื้อ ถ้าหากวันที่และเวลาที่ทำการสั่งซื้อมีระยะเวลาที่เกิน 24 ชั่วโมงแล้ว ที่หน้าจอก็จะแสดงหน้าที่เป็น Empty State ซึ่งในหน้าที่นี้ผู้ใช้งานจะไม่สามารถที่จะดูประวัติการสนทนาและทำการสนทนาได้ แต่ถ้าหากวันที่และเวลาที่ทำการสั่งซื้อมีระยะเวลาไม่เกิน 24 ชั่วโมงระบบจะแสดงหน้าห้องสนทนาขึ้นมา หลังจากนั้นระบบจะทำการขอประวัติการสนทนาจาก Database และแสดงผลข้อมูลประวัติการสนทนาที่หน้าห้องสนทนา

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามนำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการวิจัยและการอภิปรายผล

จากการดำเนินงานการพัฒนากระบวนการสนทนาของแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ซึ่งระบบการสนทนาได้ถูกพัฒนาขึ้นตามที่ได้มีการออกแบบไว้ในบทที่ 3 และระบบนี้จะใช้บนระบบปฏิบัติการ iOS โดยใช้โปรแกรม Xcode และภาษา Swift ในการพัฒนา หลังจากทีพัฒนาระบบเรียบร้อยแล้ว จะต้องมีการทดสอบระบบด้วย Unit Test และ QA (Quality Assurance) โดยระบบการสนทนาที่ ลูกค้าผู้ใช้บริการสามารถสนทนากับผู้ประกอบการร้านค้าได้ ภายใน 24 ชั่วโมงที่ทำการสั่งซื้อ ซึ่งลูกค้าผู้ใช้บริการสามารถส่งข้อความตัวอักษร อีโมจิ รูปภาพไปยังผู้ประกอบการร้านค้าได้ อีกทั้งหากทางผู้ประกอบการร้านค้าได้ทำการเปิดอ่านข้อความแล้วจะมีสถานะขึ้นที่ข้อความว่า “อ่านแล้ว” เพื่อให้ลูกค้าผู้ใช้บริการทราบว่าได้มีการเปิดอ่านข้อความแล้ว และถ้าผู้ประกอบการร้านค้าได้ทำการส่งข้อความมายังลูกค้าผู้ใช้บริการ จะมีการแจ้งเตือนทั้งแบบ Push Notification และแจ้งเตือนที่หน้ารายละเอียดคำสั่งซื้อ ว่ามีข้อความใหม่เข้ามาและยังไม่ได้ทำการเปิดอ่าน ซึ่งแต่ละหน้าได้จะได้ผลลัพธ์ ดังนี้

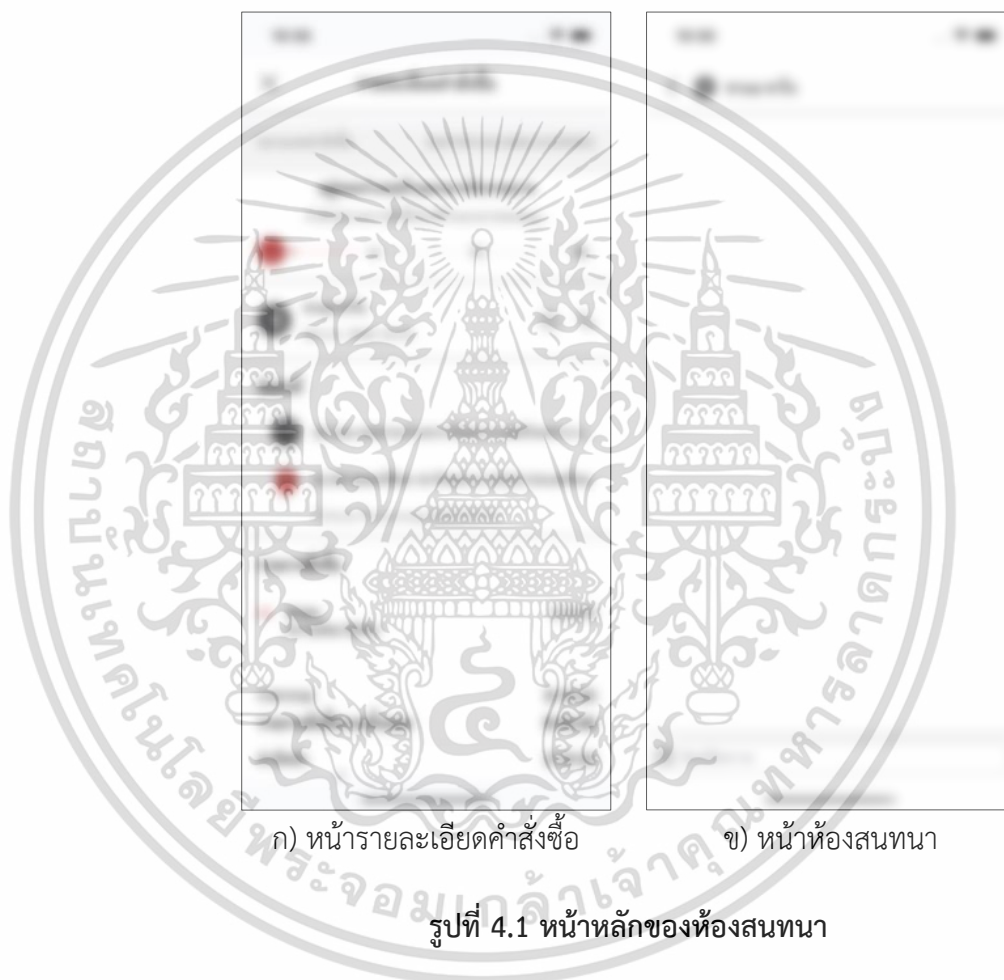
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 หน้าหลักของห้องสนทนา

4.1.1 หน้าหลักของห้องสนทนา

- 1) กรณีที่วันที่และเวลาที่ทำการสั่งซื้อ มีระยะเวลาน้อยกว่า 24 ชั่วโมง

เมื่อลูกค้าผู้ใช้บริการได้กดเข้าห้องสนทนาจากหน้ารายละเอียดคำสั่งซื้อ หากวันที่และเวลาที่ทำรายการสั่งซื้อ มีระยะเวลาน้อยกว่า 24 ชั่วโมง จะแสดงหน้าห้องสนทนา เพื่อให้ลูกค้าผู้ใช้บริการสามารถส่งและรับข้อความได้ ดังรูปที่ 4.1



ก) หน้ารายละเอียดคำสั่งซื้อ

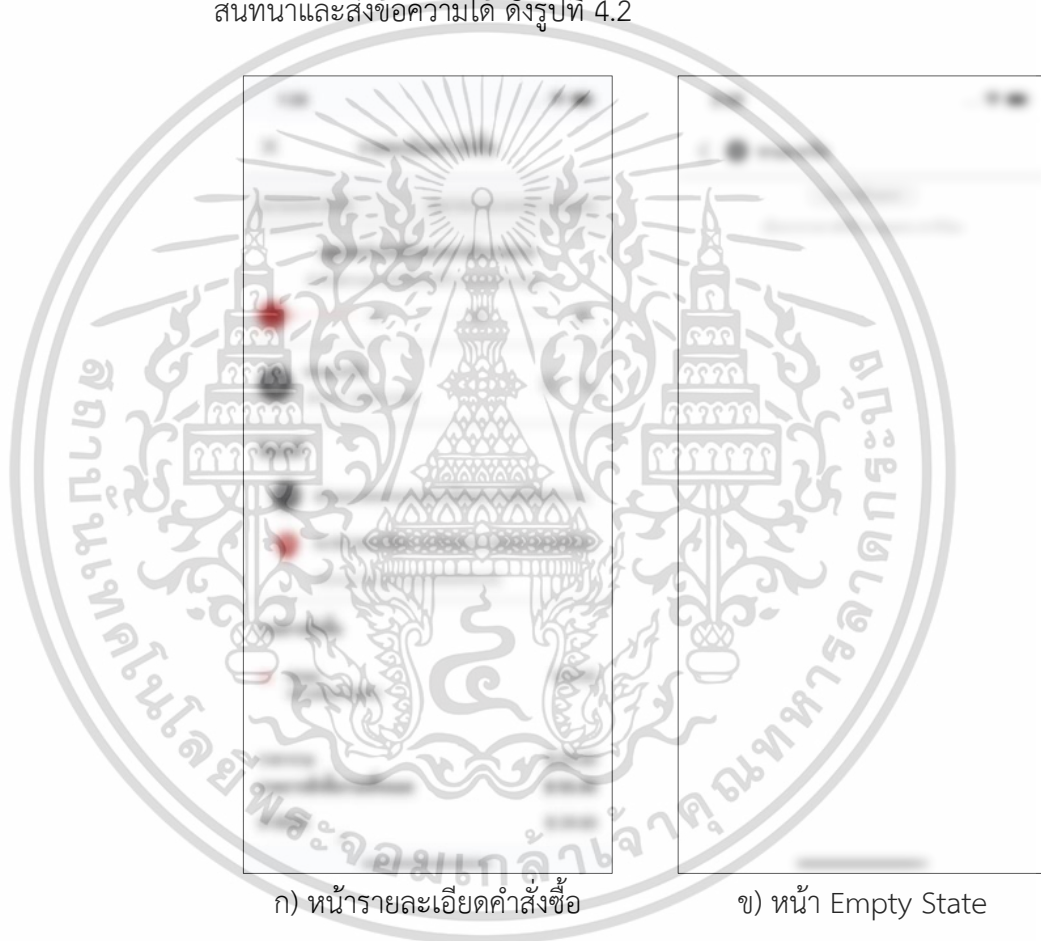
ข) หน้าห้องสนทนา

รูปที่ 4.1 หน้าหลักของห้องสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) กรณีที่วันที่และเวลาทำการสั่งซื้อ มีระยะเวลามากกว่า 24 ชั่วโมง (Empty State)

เมื่อลูกค้าใช้บริการได้กดเข้าห้องสนทนาจากหน้ารายละเอียดคำสั่งซื้อ หากวันที่และเวลาทำการสั่งซื้อ มีระยะเวลามากกว่า 24 ชั่วโมง จะแสดงหน้า Empty State เนื่องจากการสนทนาเป็นแบบชั่วคราว ดังนั้นเมื่อเวลาผ่านไปเกิน 24 ชั่วโมง คำสั่งซื้อได้สิ้นสุดลงเป็นที่เรียบร้อยแล้ว การสนทาระหว่างลูกค้าใช้บริการกับผู้ประกอบการร้านค้าได้สิ้นสุดลงเช่นกัน ทางระบบจึงได้ทำการปิดห้องสนทนา ทำให้ลูกค้าใช้บริการจะไม่สามารถดูประวัติการสนทนาและสั่งซื้อความได้ ดังรูปที่ 4.2



รูปที่ 4.2 หน้า Empty State

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การส่งข้อความตัวอักษรและแสดงข้อความในหน้าห้องสนทนา

1) การส่งข้อความตัวอักษรในห้องสนทนา

ลูกค้าผู้ใช้บริการสามารถพิมพ์เพื่อส่งข้อความได้ทั้งข้อความที่เป็นตัวอักษรและอีโมจิ ซึ่งการส่งข้อความในแต่ละครั้ง ข้อความจะต้องมีความยาวไม่เกิน 128 ตัวอักษร และลูกค้าผู้ใช้บริการสามารถส่งข้อความได้โดยการกดที่ปุ่มส่งด้านขวามือ ซึ่งปุ่มส่งจะแสดงก็ต่อเมื่อมีการข้อความอยู่ที่ช่องที่ให้พิมพ์ข้อความ ดังรูปที่ 4.3



รูปที่ 4.3 การส่งข้อความตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การแสดงข้อความตัวอักษรในห้องสนทนา

ลูกค้าผู้ใช้บริการจะเห็นวันที่ เดือน ปีที่ส่งข้อความ และเห็นข้อความทั้งข้อความที่ผู้ประกอบการร้านค้าเป็นผู้ส่ง และข้อความที่ลูกค้าผู้ใช้บริการส่งไปหาผู้ประกอบการร้านค้า ซึ่งถ้าหากข้อความที่ลูกค้าผู้ใช้บริการส่งสำเร็จ จะแสดงเวลาที่ส่งข้อความอยู่ด้านข้างของข้อความ ดังรูปที่ 4.4

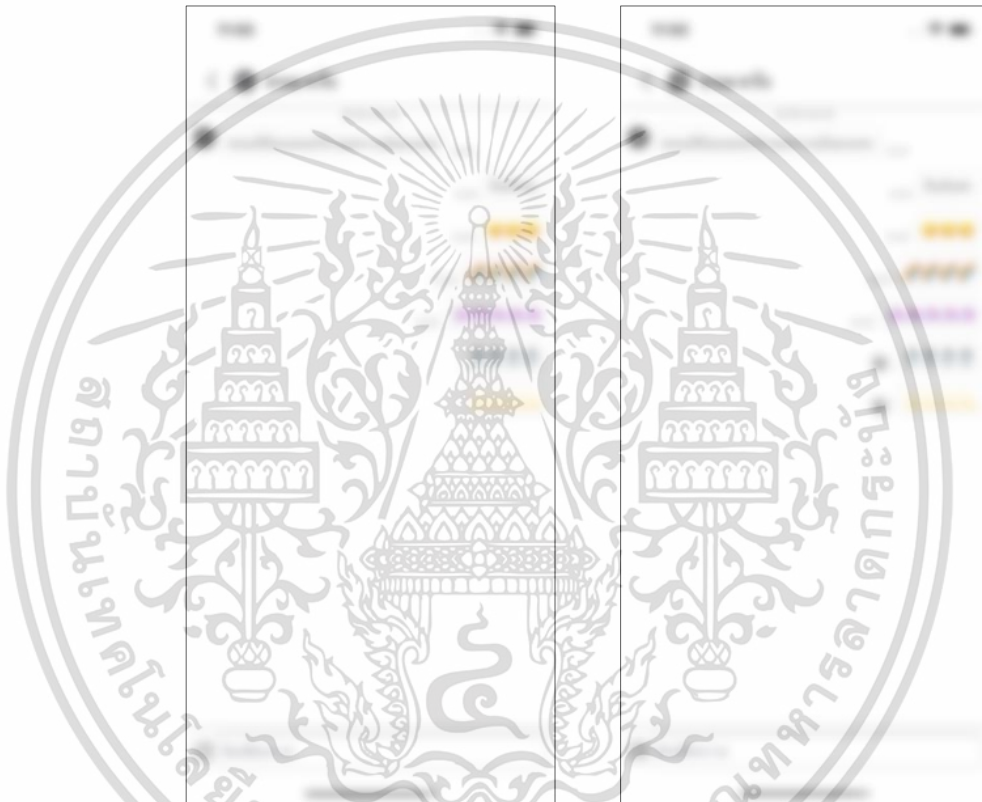


รูปที่ 4.4 แสดงข้อความตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การแสดงข้อความตัวอักษรที่ส่งไม่สำเร็จ

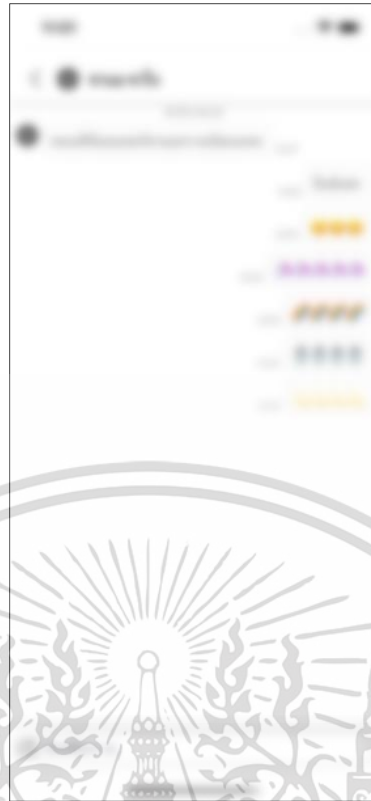
การแสดงข้อความตัวอักษรที่ส่งไม่สำเร็จ เกิดจากเมื่อลูกค้าผู้ใช้บริการทำการส่งข้อความ แล้วอินเทอร์เน็ตมีปัญหา ในช่วงแรกข้อความที่ส่งไปจะยังไม่แสดงเวลา แต่เมื่อเวลาผ่านไป จะแสดงปุ่ม Resend เพื่อแสดงให้ลูกค้าผู้ใช้บริการทราบว่าข้อความส่งไปหาผู้ประกอบการร้านค้าไม่สำเร็จ ดังรูปที่ 4.5 และเมื่ออินเทอร์เน็ตกลับมาใช้งานได้ ข้อความจะถูกส่งใหม่โดยอัตโนมัติ และจะแสดงเวลาที่ทำการส่งข้อความใหม่ ดังรูปที่ 4.6



ก) รอข้อความตอบกลับจาก server ข) แสดงปุ่ม resend ที่ข้อความ

รูปที่ 4.5 แสดงข้อความตัวอักษรที่ส่งไม่สำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 เมื่ออินเทอร์เน็ตกลับมาใช้งานได้ จะทำการส่งข้อความใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การส่งข้อความรูปภาพและแสดงข้อความในหน้าห้องสนทนา

1) การส่งข้อความรูปภาพในห้องสนทนา

ลูกค้าผู้ใช้บริการสามารถส่งข้อความที่เป็นรูปภาพได้ โดยสามารถเลือกส่งรูปภาพได้จากการถ่ายภาพหรือรูปจากคลังรูปภาพ และเมื่อเลือกรูปเสร็จแล้วรูปดังกล่าวจะถูกแสดงไปยังหน้าห้องสนทนา ดังรูปที่ 4.7



ก) เลือกรูปภาพจากกล้องหรือคลังรูปภาพ



ข) แสดงรูปภาพในคลังรูปภาพ



ค) ปรับขนาดรูปภาพที่ต้องการส่ง
และยืนยันการส่ง



ง) ส่งข้อความรูปภาพสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.7 การส่งข้อความรูปภาพ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) แสดงพรีวิวรูปภาพ

ลูกค้าผู้ใช้บริการสามารถดูรูปภาพขนาดเต็มจอได้จากการกดไปที่รูปภาพ
แล้วที่หน้าจอจะมีการแสดงรูปภาพนั้นแบบเต็มจอ ดังรูปที่ 4.8



รูปที่ 4.8 แสดงพรีวิวรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 แสดงสถานะการอ่านข้อความ

เมื่อผู้ประกอบการร้านค้าได้ทำการเปิดอ่านข้อความ ที่หน้าห้องสนทนาจะแสดงสถานะการอ่านข้อความว่า “อ่านแล้ว” ในทุกข้อความที่ถูกค้าผู้ใช้บริการเป็นผู้ส่งข้อความ ดังรูปที่ 4.9



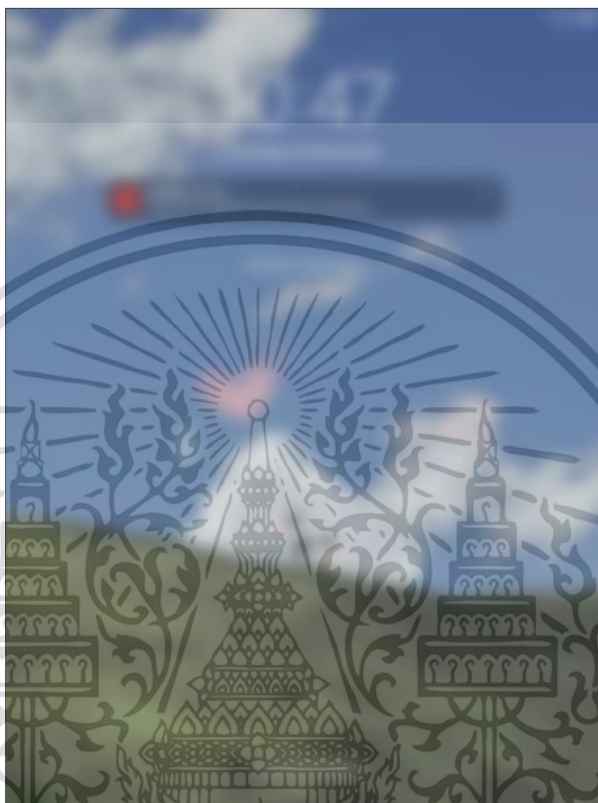
รูปที่ 4.9 แสดงสถานะการอ่านข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 การแจ้งเตือนข้อความใหม่

1) การแจ้งเตือนผ่าน Notification

เมื่อมีข้อความใหม่เข้ามา จะมีการแจ้งเตือนแบบ Push Notification เพื่อให้ลูกค้าผู้ใช้บริการทราบว่าข้อความใหม่เข้ามา ดังรูปที่ 4.10

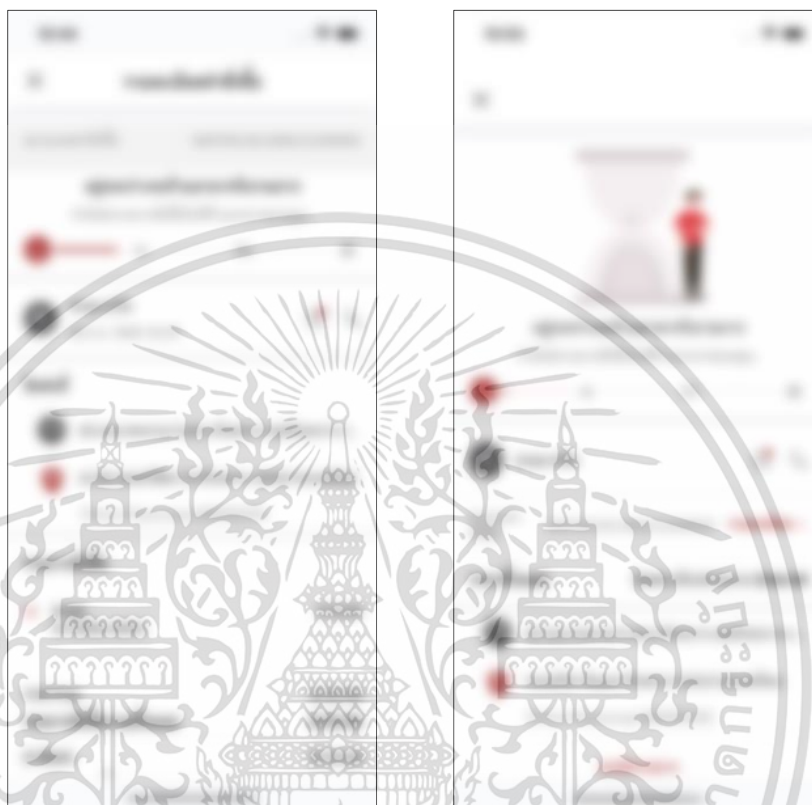


รูปที่ 4.10 การแจ้งเตือนผ่าน Notification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) แสดงแจ้งเตือนว่ามีข้อความที่ยังไม่ได้เปิดอ่าน

กรณีที่มีข้อความใหม่เข้ามา แล้วลูกค้าผู้ใช้บริการยังไม่ได้ทำการเปิดอ่าน จะมีจุดสีแดงขึ้นที่ปุ่มกดเข้าห้องแชท ทั้งในหน้ารายละเอียดคำสั่งซื้อและหน้าติดตามคำสั่งซื้อ ดังรูปที่ 4.11



ก) หน้ารายละเอียดคำสั่งซื้อ

ข) หน้าติดตามคำสั่งซื้อ

รูปที่ 4.11 แสดงแจ้งเตือนว่ามีข้อความที่ยังไม่ได้เปิดอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบระบบ

หลังจากที่ได้พัฒนาระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการ ในขั้นตอนถัดไป จะเป็นการทำการทดสอบระบบ เพื่อตรวจสอบว่าการทำงานของชุดคำสั่งต่าง ๆ หรือ Logic การทำงานนั้นทำงานถูกต้องหรือไม่ โดยการทดสอบการทำงานจะทดสอบด้วยการเขียน Unit Test ซึ่ง Unit Test จะเป็นการเขียนชุดคำสั่งสำหรับการทดสอบอัตโนมัติกับคลาสที่ต้องการทดสอบ และชุดคำสั่งที่ต้องการทดสอบจะส่งข้อมูลเข้าไปยังคลาสนั้นเพื่อทำการทดสอบ และนำผลลัพธ์ที่ได้มาเปรียบเทียบกับผลลัพธ์นั้นตรงกับที่ได้คาดหวังไว้หรือไม่ โดยในการทดสอบ Logic การทำงาน จะเขียนการทดสอบกับคลาส ViewModel เมื่อทดสอบในกรณีต่าง ๆ แล้ว จะได้ผลดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 Test Case การดูประวัติการสนทนา

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ดูประวัติการสนทนา	กดเข้าห้องสนทนาและมีข้อมูลประวัติการสนทนา	1) เข้าห้องสนทนา 2) มีข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 200	1) วันที่และเวลาที่ทำส่งชื่อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	มีข้อมูลประวัติการสนทนาในอาเรย์ chatMessage	Pass
ดูประวัติการสนทนา	กดเข้าห้องสนทนาและมีข้อมูลวันที่ของประวัติการสนทนา โดยวันที่ของข้อมูลประวัติการสนทนาเป็นวันที่เดียวกันทั้งหมด	1) เข้าห้องสนทนา 2) มีข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 200	1) วันที่และเวลาที่ทำส่งชื่อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	มีข้อมูลวันที่ของประวัติการสนทนาในอาเรย์ chatDate	Pass
ดูประวัติการสนทนา	กดเข้าห้องสนทนาและไม่มีข้อมูลประวัติการสนทนา	1) เข้าห้องสนทนา 2) ไม่มีข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 200	1) วันที่และเวลาที่ทำส่งชื่อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	ไม่มีข้อมูลวันที่ในข้อมูลประวัติการสนทนาในอาเรย์ chatMessage	Pass

ตารางที่ 4.1 Test Case การดูประวัติการสนทนา (ต่อ)

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ดูประวัติการสนทนา	กดเข้าห้องสนทนาและไม่มีวันที่และข้อมูลประวัติการสนทนา	1) เข้าห้องสนทนา 2) ไม่มีข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 200	1) วันที่และเวลาที่ทำส่งข้อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	ไม่มีข้อมูลวันที่ของประวัติการสนทนาในอาเรย์	Pass
ดูประวัติการสนทนา	กดเข้าห้องสนทนาและมีวันที่ที่ทำการสนทนาเป็นคนละวันกัน	1) เข้าห้องสนทนา 2) ไม่มีข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 200	1) วันที่และเวลาที่ทำส่งข้อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	มีข้อมูลวันที่ในอาเรย์ chatDate โดยในอาเรย์จะมีจำนวน section ที่มากกว่า 1 section	Pass
ดูประวัติการสนทนา	กดเข้าห้องสนทนาและมีข้อมูลประวัติการสนทนาโดยวันที่ที่ทำการสนทนาเป็นคนละวันกัน	1) เข้าห้องสนทนา 2) ไม่มีข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 200	1) วันที่และเวลาที่ทำส่งข้อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	มีข้อมูลข้อมูลประวัติการสนทนาในอาเรย์ chatMessage โดยในอาเรย์จะมีจำนวน section ที่มากกว่า 1 section	Pass

ตารางที่ 4.1 Test Case การดูประวัติการสนทนา (ต่อ)

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ดูประวัติการสนทนา	กดเข้าห้องสนทนาแล้วเกิดปัญหาที่ Server	1) เข้าห้องสนทนา 2) ไม่พบข้อมูลประวัติการสนทนา โดยการ Mock status ของ API ให้เป็น 500	1) วันที่และเวลาที่ทำสั่งซื้อมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	ไม่พบข้อมูลประวัติการสนทนา	Pass
ดูประวัติการสนทนา	กดเข้าห้องสนทนาแล้วแสดงหน้า Empty State	1) ตรวจสอบวันที่และเวลาที่ทำสั่งซื้อ 2) แสดงหน้า Empty State	1) วันที่และเวลาที่ทำสั่งซื้อมีระยะเวลาเกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254	แสดงหน้า Empty State	Pass

ตารางที่ 4.2 Test Case การกดส่งข้อความ

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
กดส่งข้อความ	กรณีที่ไม่มีการประวัติการสนทนา และเมื่อมีการส่งข้อความ จะมีข้อความในอาร์เรย์ chatMessage	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อความที่จะส่ง โดยให้มีค่าเป็น String	มีข้อมูลข้อความในอาร์เรย์ chatMessage	Pass
กดส่งข้อความ	กรณีที่ไม่มีการประวัติการสนทนา และเมื่อมีการส่งข้อความ ข้อความจะมีสถานะการส่งเป็น waiting เพื่อรอแสดงเวลาในการส่งข้อความ	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อความที่จะส่ง โดยให้มีค่าเป็น String	สถานะการส่งข้อความมีค่าเท่ากับ "waiting"	Pass

ตารางที่ 4.3 Test Case การรับข้อความ

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
รับข้อความ	กรณีที่ไม่มีประวัติการสนทนา และเมื่อผู้ประกอบการร้านค้ามีการส่งข้อความ จะมีข้อมูลข้อความในอาเรย์ chatMessage	1) เข้าห้องสนทนา 2) รับข้อความจากผู้ประกอบการร้านค้า	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อมูลข้อความของผู้ประกอบการร้านค้า	มีข้อมูลข้อความในอาเรย์ chatMessage	Pass
รับข้อความ	กรณีที่ไม่มีประวัติการสนทนา และเมื่อผู้ประกอบการร้านค้ามีการส่งข้อความ จะมีข้อมูลวันที่ในอาเรย์ chatDate	1) เข้าห้องสนทนา 2) รับข้อความของผู้ประกอบการร้านค้าจาก Socket	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อมูลข้อความของผู้ประกอบการร้านค้า	มีข้อมูลข้อความในอาเรย์ chatMessage	Pass

ตารางที่ 4.4 Test Case การส่งข้อความสำเร็จ

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ส่งข้อความสำเร็จ	กรณีที่ไม่มีประวัติการสนทนา และเมื่อมีการส่งข้อความ และข้อความส่งสำเร็จ ข้อความจะมีสถานะการส่งเป็น success	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ 3) รับข้อความที่ลูกค้า ผู้ให้บริการเป็นผู้ส่งจาก Socket	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อมูลข้อความของลูกค้า ผู้ให้บริการที่เป็นผู้ส่ง	สถานะการส่งข้อความมีค่าเท่ากับ "success"	Pass
ส่งข้อความสำเร็จ	กรณีที่ไม่มีประวัติการสนทนา และเมื่อมีการส่งข้อความ และข้อความส่งสำเร็จ ข้อความจะได้รับและแสดงเวลาที่ทำการส่งข้อความ	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ 3) รับข้อความที่ลูกค้า ผู้ให้บริการเป็นผู้ส่งจาก Socket	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อมูลข้อความของลูกค้า ผู้ให้บริการที่เป็นผู้ส่ง	ข้อมูลเวลา (timeStamp) ที่ ข้อมูลข้อความในอาเรย์ chatMessage จะไม่ค่าไม่เท่ากับ ""	Pass

ตารางที่ 4.4 Test Case การส่งข้อความสำเร็จ (ต่อ)

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ส่งข้อความสำเร็จ	กรณีที่มีประวัติการสนทนา และเมื่อส่งข้อความสำเร็จ แล้วข้อความที่ส่งเป็นคนละวันกับข้อมูลในประวัติการสนทนา ในอาเรย์ของ chatDate จะมีจำนวน section ที่มากกว่า 1 section	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ 3) รับข้อความที่ลูกค้า ผู้ใช้บริการเป็นผู้ส่งจาก Socket	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อมูลประวัติข้อความโดยวันที่ และเวลาของข้อความเป็นวันก่อนหน้า	อาเรย์ของ chatDate จะมีจำนวน section ที่มากกว่า 1 section	Pass
ส่งข้อความสำเร็จ	กรณีที่มีประวัติการสนทนา และเมื่อส่งข้อความสำเร็จ แล้วข้อความที่ส่งเป็นคนละวันกับข้อมูลในประวัติการสนทนา ในอาเรย์ของ chatMessage จะมีจำนวน section ที่มากกว่า 1 section	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ 3) รับข้อความที่ลูกค้า ผู้ใช้บริการเป็นผู้ส่งจาก Socket	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ข้อมูลประวัติข้อความโดยวันที่ และเวลาของข้อความเป็นวันก่อนหน้า	อาเรย์ของ chatMessage จะมีจำนวน section มากกว่า 1 section	Pass

ตารางที่ 4.5 Test Case การส่งข้อความไม่สำเร็จ

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ส่งข้อความไม่สำเร็จ	กรณีที่ไม่มีประวัติการสนทนา และเมื่อมีการส่งข้อความ และอินเทอร์เน็ตมีปัญหา ข้อความจะส่งไม่สำเร็จโดย ข้อความจะมีสถานะการส่ง เป็น fail	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ไม่ได้รับข้อมูลข้อความของลูกค้า ผู้ให้บริการที่เป็นผู้ส่งจาก Socket	สถานะการส่งข้อความมีค่า เท่ากับ “fail”	Pass
ส่งข้อความไม่สำเร็จ	กรณีที่ไม่มีประวัติการสนทนา และเมื่อมีการส่งข้อความ และอินเทอร์เน็ตมีปัญหา ข้อความจะส่งไม่สำเร็จ และ ข้อความจะถูกเก็บลงบน Local Database	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ 3) เมื่อข้อความส่งไม่สำเร็จ ข้อความจะถูกเก็บลงบน Local Database	1) วันที่และเวลาที่ทำส่งข้อความ ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ไม่ได้รับข้อมูลข้อความของลูกค้า ผู้ให้บริการที่เป็นผู้ส่งจาก Socket 4) เก็บข้อมูลข้อความลงใน Local Database	มีข้อมูลใน Local Database	Pass

ตารางที่ 4.6 Test Case การส่งข้อความใหม่อีกครั้ง

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ส่งข้อความใหม่อีกครั้ง (Resend)	กรณีที่ไม่มีประวัติการสนทนา และมีข้อความที่ส่งไม่สำเร็จ อยู่ เมื่ออินเทอร์เน็ตกลับมาใช้ได้ จะมีการส่งข้อความนั้นใหม่ จะทำการลบข้อมูลข้อความนั้นออกจาก Local Database	<ol style="list-style-type: none"> 1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่งข้อความ 3) เมื่อข้อความส่งไม่สำเร็จ ข้อความจะถูกเก็บลงบน Local Database 4) เข้าห้องสนทนาใหม่ 5) ส่งข้อความที่อยู่ใน Local Database ใหม่ 6) ลบข้อมูลข้อความที่ส่งใหม่ ออกจาก Local Database 	<ol style="list-style-type: none"> 1) วันที่และเวลาที่ทำส่งข้อความมีระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ไม่ได้รับข้อมูลข้อความของลูกค้า ผู้ใช้บริการที่เป็นผู้ส่งจาก Socket 4) เก็บข้อมูลข้อความลงใน Local Database 	สถานะการส่งข้อความที่ทำการส่งใหม่มีค่าเท่ากับ "success"	Pass

ตารางที่ 4.6 Test Case การส่งข้อความใหม่อีกครั้ง (ต่อ)

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
ส่งข้อความใหม่อีกครั้ง (Resend)	กรณีที่ไม่มีประวัติการสนทนา และมีข้อความที่ส่งไม่สำเร็จ อยู่ เมื่ออินเทอร์เน็ตกลับมา ใช้ได้ จะมีการส่งข้อความนั้น ใหม่ และเมื่อส่งสำเร็จ ข้อความจะมีสถานะการส่ง เป็น success	1) เข้าห้องสนทนา 2) พิมพ์ข้อความแล้วกดส่ง ข้อความ 3) เมื่อข้อความส่งไม่สำเร็จ ข้อความจะถูกเก็บลงบน Local Database 4) เข้าห้องสนทนาใหม่ 5) ส่งข้อความที่อยู่ใน Local Database ใหม่	1) วันที่และเวลาที่ทำส่งข้อความมี ระยะเวลาไม่เกิน 24 ชั่วโมง 2) หมายเลขห้องสนทนา: 254 3) ไม่ได้รับข้อมูลข้อความของลูกค้า ผู้ใช้บริการที่เป็นผู้ส่งจาก Socket 4) เก็บข้อมูลข้อความลงใน Local Database	ไม่มีข้อมูลใน Local Database	Pass

ตารางที่ 4.7 Test Case การแสดงสถานะการอ่านข้อความ

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status (Pass/Fail)
แสดงสถานะการอ่านข้อความ	เมื่อผู้ประกอบการร้านค้าได้ทำการอ่านข้อความที่ลูกค้าผู้ใช้บริการเป็นผู้ส่ง จะเปลี่ยนสถานะการอ่านข้อความจาก false เป็น true	1) มีประวัติข้อความการสนทนาในอาเรย์ chatMessage 2) รับข้อมูลสถานะการอ่านของผู้ประกอบการร้านค้า	1) ข้อมูลข้อความในอาเรย์ chatMessage มีทั้งข้อความที่ผู้ส่งเป็นผู้ประกอบการร้านค้าและลูกค้าผู้ใช้บริการ 2) ข้อมูลการอ่านข้อความของผู้ประกอบการร้านค้า	สถานะการอ่านข้อความที่ลูกค้าผู้ใช้บริการเป็นผู้ส่งจะมีค่าเป็น true	Pass
แสดงสถานะการอ่านข้อความ	เมื่อลูกค้าผู้ใช้บริการได้ทำการอ่านข้อความที่ลูกค้าผู้ใช้บริการได้ทำการส่ง จะไม่เปลี่ยนสถานะการอ่านข้อความ	1) มีประวัติข้อความการสนทนาในอาเรย์ chatMessage 2) รับข้อมูลสถานะการอ่านของผู้ประกอบการร้านค้า	1) ข้อมูลข้อความในอาเรย์ chatMessage มีทั้งข้อความที่ผู้ส่งเป็นผู้ประกอบการร้านค้าและลูกค้าผู้ใช้บริการ 2) ข้อมูลการอ่านข้อความของลูกค้าผู้ใช้บริการ	สถานะการอ่านข้อความที่ลูกค้าผู้ใช้บริการเป็นผู้ส่งจะมีค่าเป็น false	Pass

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการที่ผู้จัดทำได้มีโอกาสร่วมโครงการสหกิจศึกษากับบริษัทแห่งหนึ่ง ผู้จัดทำได้รับมอบหมายให้พัฒนาระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเช่น เพื่อเป็นช่องทางเลือกในการติดต่อสื่อสารระหว่างลูกค้าผู้ใช้บริการกับผู้ประกอบการร้านค้าบนแพลตฟอร์มคลาวด์คิทเช่น ซึ่งระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเช่นนั้นเป็นส่วนหนึ่งของแพลตฟอร์มจึงจำเป็นต้องทำงานร่วมกันกับอีกสองส่วนที่สำคัญ คือ ระบบการสนทนาที่ทำงานบนระบบหลังบ้าน และระบบการสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า ซึ่งเมื่อพัฒนาระบบการสนทนาแล้ว ได้มีการนำไปใช้งานจริงกับแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการเป็นที่เรียบร้อยแล้ว โดยลูกค้าผู้ใช้บริการสามารถใช้ระบบการสนทนาได้เมื่อทำการอัปเดตแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการให้เป็นเวอร์ชันล่าสุดบนอุปกรณ์ระบบปฏิบัติการ iOS

5.2 ข้อเสนอแนะ

ในการพัฒนาระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเช่น ผู้จัดทำได้พบเจอปัญหาต่าง ๆ และมีข้อเสนอแนะเกี่ยวกับการพัฒนาระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเช่น เพื่อนำมาปรับปรุงและพัฒนาในอนาคต

- 1) เนื่องจากผู้จัดทำยังขาดความรู้และประสบการณ์ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ iOS ทำให้มีการเขียนพวก Logic การทำงาน หรือคำสั่งการทำงานบางคำสั่งไปอยู่ใน ViewController แทนที่ควรจะอยู่ใน ViewModel ทำให้ชุดคำสั่งการทำงานอยู่กระจายตามไฟล์ต่าง ๆ ดังนั้นจึงควรที่จะต้องปรับปรุงและแก้ไข เพื่อให้ชุดคำสั่งการทำงานอยู่ที่เดียวกัน และจะทำให้ง่ายต่อการ Maintenance
- 2) เนื่องจากผู้จัดทำยังขาดความรู้และประสบการณ์ในการทำการทดสอบแอปพลิเคชัน ทำให้ต้องใช้เวลาในการศึกษาค่อนข้างนาน และการเขียนการทดสอบ Unit Test นั้น จำเป็นต้องทดสอบการทำงานในส่วนของ ViewModel จึงต้องมีการย้ายส่วนของ Logic ไปไว้ใน ViewModel เพื่อให้สามารถทำการทดสอบและง่ายต่อการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) เนื่องจากในระบบการสนทนาบนแอปพลิเคชันสำหรับลูกค้าผู้ใช้บริการนั้น ลูกค้าผู้ใช้บริการสามารถส่งข้อความตัวอักษรได้ แต่ถ้าหากว่ามีตัวเลือกข้อความที่ลูกค้าผู้ใช้บริการมักพิมพ์บ่อย ๆ มาเป็นตัวเลือกแทนการพิมพ์ข้อความ จะสามารถช่วยให้ลูกค้าผู้ใช้บริการประหยัดเวลาและสะดวกต่อการใช้งานมากยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Abhilash Mathur. **MVVM in iOS Swift**. [ออนไลน์]. แหล่งที่มา : <https://medium.com/@abhilash.mathur1891/mvvm-in-ios-swift-aa1448a66fb4>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [2] Aj. Thiti Theerathean. **Introduction to Swift**. [ออนไลน์]. แหล่งที่มา : <https://ajthiti.gitbook.io/swift/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [3] Apple Inc. **Swift ภาษาอันทรงพลังที่เปิดกว้างให้ทุกคนสามารถสร้างแอปอันน่าทึ่ง**. [ออนไลน์]. แหล่งที่มา : <https://www.apple.com/th/swift/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [4] benznest. **[iOS] การติดตั้งและใช้งาน Cocoapods ใน Xcode**. [ออนไลน์]. แหล่งที่มา : <https://benzneststudios.com/blog/ios/how-to-install-cocoapod-on-xcode/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [5] Chuong Tran. **IOS CHAT APPLICATION [dissertation]**. Vaasa, Finland. Vaasa University of Applied Sciences; 2021
- [6] Farhana Mustafa. **Using the MVVM Architectural Design Pattern in iOS**. [ออนไลน์]. แหล่งที่มา : <https://blog.devgenius.io/using-the-mvvm-architectural-design-pattern-in-ios-c70e16352be5>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [7] **Figma คืออะไร? ทำไมถึงเป็น Tool มาแรงที่สุดแห่งปี**. [ออนไลน์]. แหล่งที่มา : <https://blog.skooldio.com/figma-ui-design-tool>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [8] **Git version control UI SourceTree**. [ออนไลน์]. แหล่งที่มา : <https://onlymycode.blogspot.com/2017/05/git-version-control-ui-sourcetree.html>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [9] Jedsada Saengow. **[Firebase] คืออะไร มาดูวิธีสร้าง Project และทำความเข้าใจกับ Firebase**. [ออนไลน์]. แหล่งที่มา : <https://medium.com/jed-ng/firebase-คืออะไร-มาดูวิธีสร้าง-project-และทำความเข้าใจกับ-firebase-d48bfac67b14>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [10] jake-kim. **[iOS - swift] Taste of MVVM concept applied with clean architecture**. [ออนไลน์]. แหล่งที่มา : <https://ios-development.tistory.com/555>. สืบค้นเมื่อ 15 ธันวาคม 2565
- [11] Jirawatee. **รู้จัก Firebase Cloud Messaging (FCM) ตั้งแต่ Zero จนเป็น Hero**. [ออนไลน์]. แหล่งที่มา : <https://medium.com/firebasethailand/รู้จัก-firebase-cloud-messaging-fcm-ตั้งแต่-zero-จนเป็น-hero-fb7900af92cd>. สืบค้นเมื่อ 5 ตุลาคม 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [12] K.Sarit. **วิธีติดตั้ง Pod สำหรับ Project ของคุณบน Xcode.** [ออนไลน์]. แหล่งที่มา : <https://medium.com/@anutakoon/วิธีติดตั้ง-pod-สำหรับ-project-ของคุณบน-xcode-386b7896de0d>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [13] **Module คืออะไร?** [ออนไลน์]. แหล่งที่มา : <https://www.ineedtoknow.org/module-คืออะไร/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [14] Naruapon Suwanwijit. **ภาษา Swift.** [ออนไลน์]. แหล่งที่มา : <https://medium.com/@naruapon/ภาษา-swift-73a77c62110>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [15] N. Piyush, T. Choudhury and P. Kumar, "Conversational commerce a new era of e-business," 2016 International Conference System Modeling & Advancement in Research Trends (SMART), 2016, pp. 322-327, doi: 10.1109/SYSMART.2016.7894543.
- [16] Ong Komsit Chusangthong. **เริ่มเขียน MVVM กันเถอะ [iOS].** [ออนไลน์]. แหล่งที่มา : <https://medium.com/tech-at-tdg/เริ่มเขียน-mvvm-กันเถอะ-ios-2bc8d1292b01>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [17] Pé Pratchaya. **มาเขียน UnitTest แบบ BDD ด้วย Quick/Nimble กันดีกว่า [iOS].** [ออนไลน์]. แหล่งที่มา : <https://blog.appsynth.net/มาเขียน-unittest-แบบ-bdd-ด้วย-quick-nimble-กันดีกว่า-ios-7884f790da9>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [18] Prakai. **เข้าใจกระบวนการคิดและทำงานแบบ ‘Agile – Scrum’ ทำไมถึงสำคัญเวลาประชุม ‘ก้าวเล็กแต่บรรลุเป้าหมายใหญ่เร็ว’.** [ออนไลน์]. แหล่งที่มา : <https://www.marketingoops.com/data/how-to-effectively-plan-sprints-for-agile-and-scrum-marketing-teams/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [19] Surachet Y. **[iOS] มาเริ่มเขียน Clean Swift กันเถอะ.** [ออนไลน์]. แหล่งที่มา : <https://medium.com/@surachet.y/ios-มาเริ่มเขียน-clean-swift-กันเถอะ-e69f9ce567e1>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [20] Teerajiraphatchandej. **รู้จักกับ Realm: ฐานข้อมูลบนแอปทางเลือกใหม่ ที่เร็วกว่า SQLite.** [ออนไลน์]. แหล่งที่มา : <https://nextflow.in.th/2016/realm-mobile-database-intro-thai/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [21] Ton Attapong. **Swift รู้จักกับ Life cycle ใน iOS.** [ออนไลน์]. แหล่งที่มา : <https://medium.com/seekster-development/swift-รู้จักกับ-life-cycle-ใน-ios-5940a7561bb0>. สืบค้นเมื่อ 5 ตุลาคม 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [22] Traitanit Huangsrri. **มีอะไรใหม่เกี่ยวกับ Testing จากงาน WWDC 2019.** [ออนไลน์]. แหล่งที่มา : <https://medium.com/linedevth/มีอะไรใหม่เกี่ยวกับ-testing-จากงาน-wwdc-2019-25c7d409ccfb>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [23] TUXSA. **Scrum & Sprint กุญแจสำคัญของการทำ Agile.** [ออนไลน์]. แหล่งที่มา : <https://tuxsablog.skilllane.com/data-savvy/scrum-and-sprint/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [24] Waerohanee Wachayee. **Postman คืออะไร.** [ออนไลน์]. แหล่งที่มา : <https://medium.com/@waerohaneewaehayee/postman-คืออะไร-c6461461cc55>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [25] **โมดูล (Module).** [ออนไลน์]. แหล่งที่มา : http://www.168training.com/e-learning_new/python/lesson2/more/page1.php. สืบค้นเมื่อ 5 ตุลาคม 2565



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

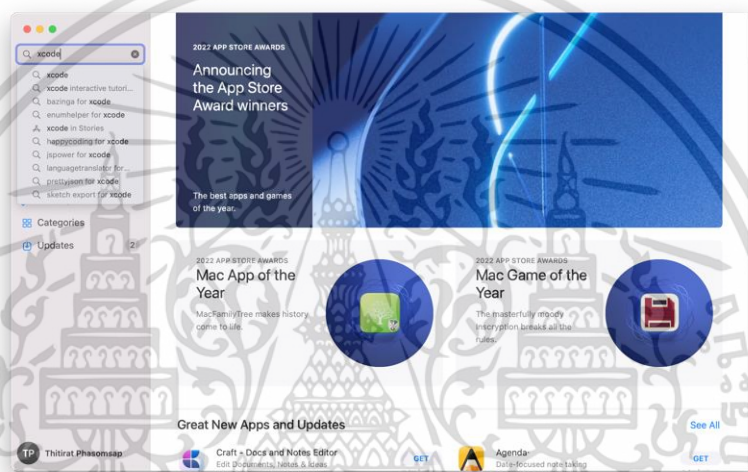
ภาคผนวก ก

การติดตั้งโปรแกรม Xcode

ก.1 ดาวนโหลดและติดตั้งโปรแกรม Xcode

ก.1.1 ขั้นตอนการดาวน์โหลดและติดตั้งโปรแกรม Xcode

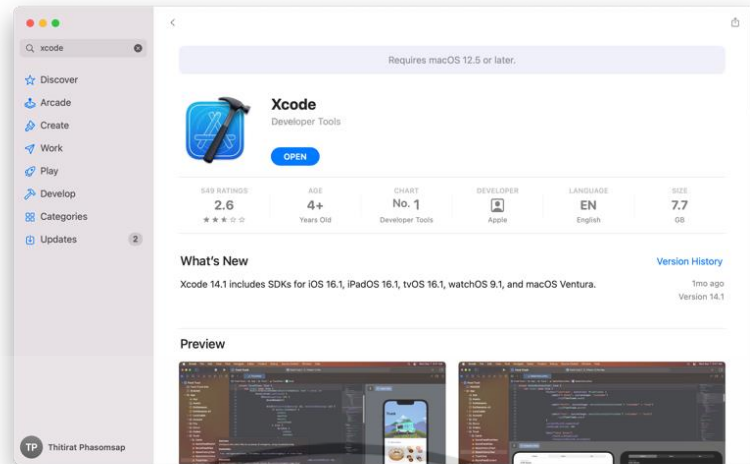
- 1) เริ่มจากอัปเดต macOS เป็น version ล่าสุด
- 2) เข้าไป App Store ของ MacBook และค้นหาโปรแกรม Xcode ดังรูปที่ ก.1



รูปที่ ก.1 หน้าผลการค้นหาโปรแกรม Xcode ใน App Store

- 3) กด install โปรแกรม Xcode
- 4) เมื่อดาวน์โหลดเสร็จเรียบร้อยแล้ว ให้กดปุ่ม Open เพื่อทำการเปิด App Xcode ดังรูปที่ ก.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 หน้า App Store แสดงปุ่ม Open เพื่อเปิด App

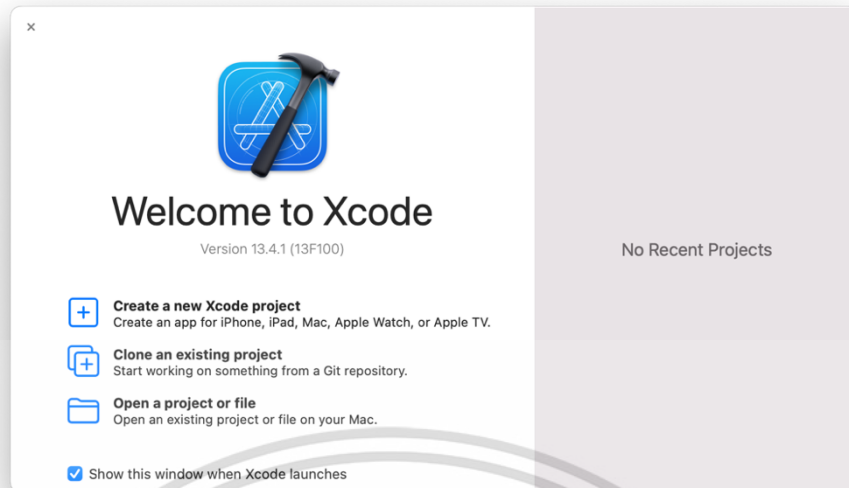
- 5) เมื่อเปิดโปรแกรมแล้ว ให้เลือก Agree เพื่อยอมรับ license ดังรูปที่ ก.3



รูปที่ ก.3 หน้าจอแสดงข้อความขออนุญาตก่อนติดตั้งแอปพลิเคชัน

- 6) กรอกรหัสประจำเครื่องเพื่อเริ่มทำการติดตั้ง
- 7) เมื่อติดตั้งสำเร็จ จะขึ้นหน้า Xcode ที่ได้ทำการติดตั้ง และหลังจากนี้จะสามารถใช้โปรแกรม Xcode ได้ ดังรูปที่ ก.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 หน้าต่างโปรแกรม Xcode

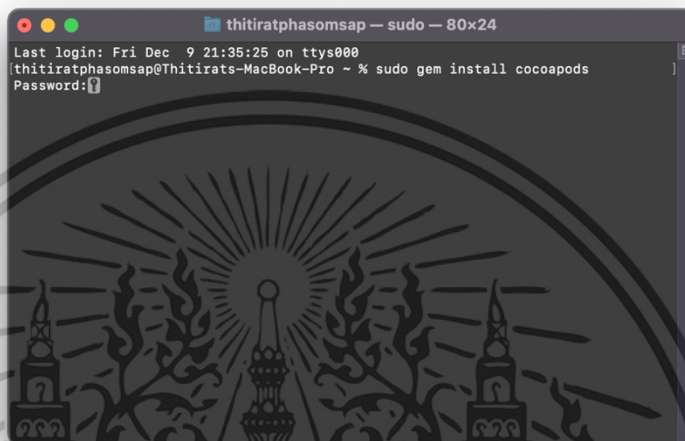
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การติดตั้ง Cocoapods

ข.1 การติดตั้ง Cocoapods

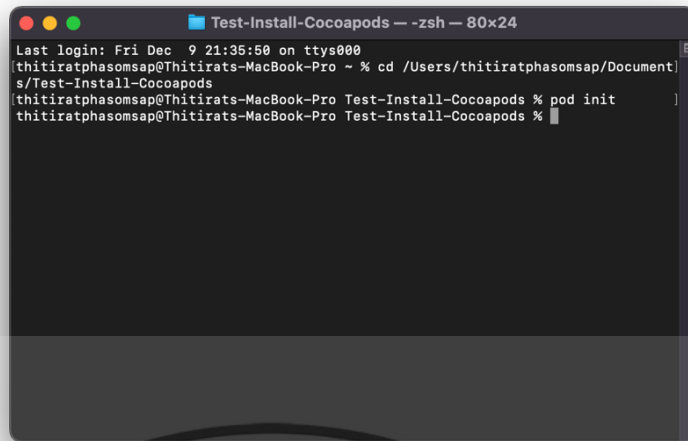
- 1) เปิด Terminal แล้วพิมพ์คำสั่ง `sudo gem install cocoapods` ลงใน Terminal เพื่อติดตั้ง Cocoapods ลงในเครื่อง ดังรูปที่ ข.1



รูปที่ ข.1 ติดตั้ง Cocoapods ลงในเครื่อง

- 2) หลังจากนั้นจะขึ้นให้กรอก password เสร็จแล้วก็รอการติดตั้ง
- 3) เมื่อติดตั้งเรียบร้อยแล้วให้เขาไปที่ path ของโปรเจค Xcode ที่ต้องการจะติดตั้ง pod
- 4) หลังจากนั้น พิมพ์คำสั่ง `pod init` ลงไปใน Terminal เพื่อสร้าง Podfile ไว้สำหรับติดตั้ง Library ต่าง ๆ ดังรูปที่ ข.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



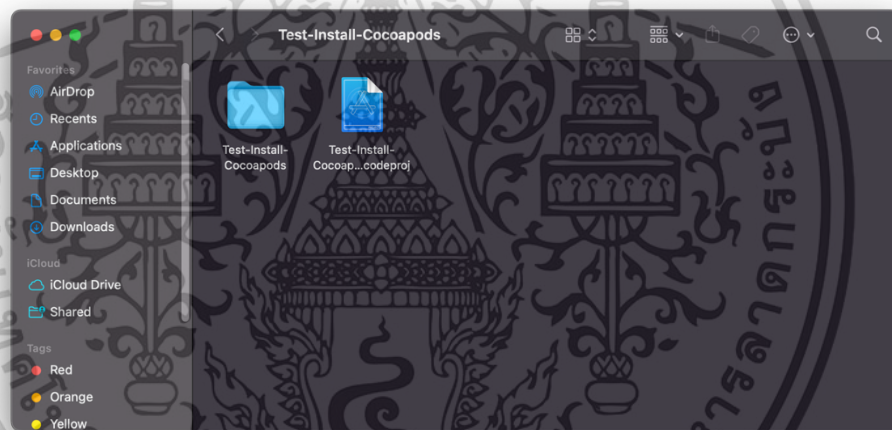
```

Test-Install-Cocoapods --zsh-- 80x24
Last login: Fri Dec 9 21:35:50 on ttys000
[thitiratphasomsap@Thitirats-MacBook-Pro ~ % cd /Users/thitiratphasomsap/Documents/Test-Install-Cocoapods
[thitiratphasomsap@Thitirats-MacBook-Pro Test-Install-Cocoapods % pod init
thitiratphasomsap@Thitirats-MacBook-Pro Test-Install-Cocoapods %

```

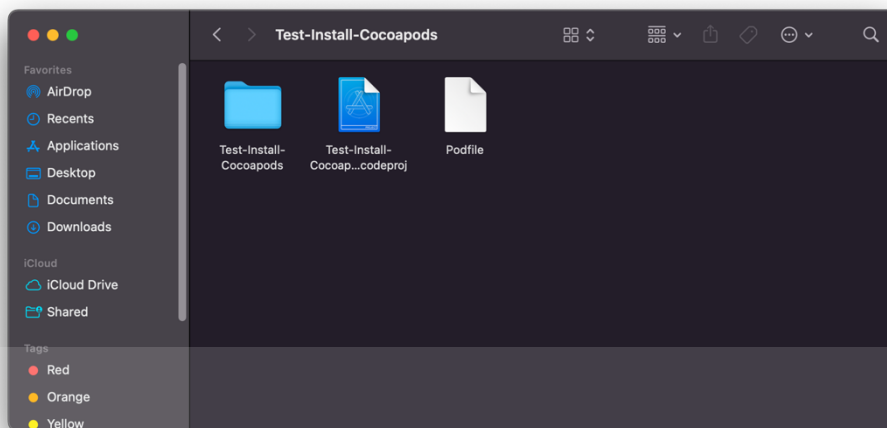
รูปที่ ข.2 สร้าง Podfile

5) เมื่อติดตั้งเสร็จแล้ว ที่โปรเจกจะมีไฟล์ Podfile เพิ่มขึ้นมา ดังรูปที่ ข.3



ก) ไฟล์โปรเจกก่อนติดตั้ง Cocoapods

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ข) ไฟล์โปรเจกต์หลังก่อนติดตั้ง Cocoapods

รูปที่ ข.3 ไฟล์โปรเจกต์หลังสร้าง Podfile



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6) จากนั้นพิมพ์คำสั่ง `pod install` ลงบน Terminal เพื่อติดตั้ง Cocoapods ดังรูปที่ ข.4

```

Last login: Fri Dec 9 21:53:04 on ttys000
thitiratphasomsap@Thitirats-MacBook-Pro ~ % cd /Users/thitiratphasomsap/Documents/Test-Install-Cocoapods
thitiratphasomsap@Thitirats-MacBook-Pro Test-Install-Cocoapods % pod init
thitiratphasomsap@Thitirats-MacBook-Pro Test-Install-Cocoapods % pod install

Analyzing dependencies
Downloading dependencies
Generating Pods project
Integrating client project

[[!]] Please close any current Xcode sessions and use 'Test-Install-Cocoapods.xcworkspace' for this project from now on.
Pod installation complete! There are 0 dependencies from the Podfile and 0 total pods installed.

[[!]] The Podfile does not contain any dependencies.

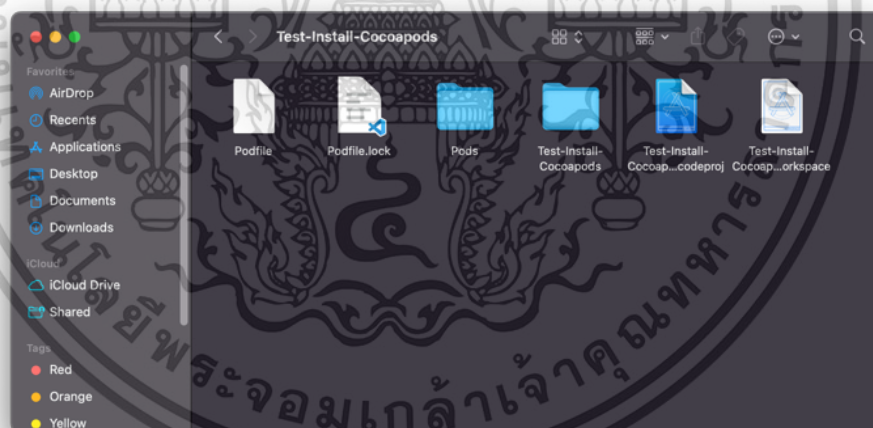
[[!]] Automatically assigning platform 'iOS' with version '15.5' on target 'Test-Install-Cocoapods' because no platform was specified. Please specify a platform for this target in your Podfile. See 'https://guides.cocoapods.org/syntax/podfile.html#platform'.

[[!]] Your project does not explicitly specify the CocoaPods master specs repo. Since CDN is now used as the default, you may safely remove it from your repos directory via 'pod repo remove master'. To suppress this warning please add 'warn_for_unused_master_specs_repo => false' to your Podfile.
thitiratphasomsap@Thitirats-MacBook-Pro Test-Install-Cocoapods %

```

รูปที่ ข.4 ติดตั้ง Cocoapods

- 7) หลังจากทำการ `install` เรียบร้อยแล้ว จะได้ไฟล์นามสกุล `.xcworkspace` เพิ่มขึ้นมาซึ่งโดยปกติแล้วก่อนที่จะติดตั้ง Cocoapods จะมีแค่ไฟล์ `.xcodeproj` เมื่อติดตั้ง Cocoapods แล้วจะย้ายไปทำงานในไฟล์ `.xcworkspace` แทน ดังรูปที่ ข.5

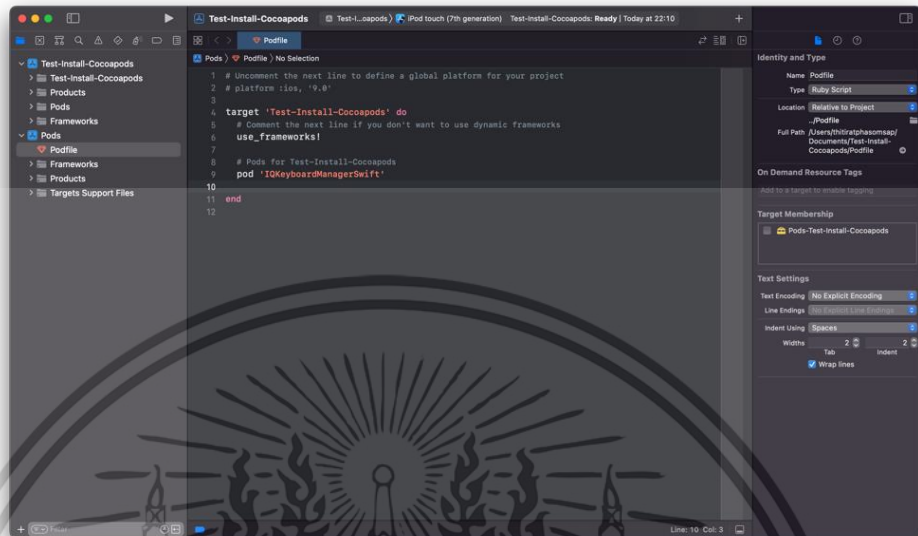


รูปที่ ข.5 ไฟล์ทั้งหมดในโปรเจก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.2 การติดตั้ง Library ใน cocoapods

- 1) เปิดไฟล์โปรเจกต์นามสกุล .xcworkspace ขึ้นมา ซึ่งเมื่อเข้ามาแล้วจะเจอไฟล์ Podfile
- 2) พิมพ์คำสั่งเพื่อติดตั้ง Library ที่ต้องการลงใน podfile ดังรูป ข.6



รูปที่ ข.6 พิมพ์ Library ที่ต้องการติดตั้ง

- 3) หลังจากนั้นให้เปิด Terminal โดย path ยังเป็น path ของไฟล์โปรเจกต์ แล้วพิมพ์คำสั่ง pod install ดังรูป ข.7



รูปที่ ข.7 ติดตั้ง Library ลงโปรเจกต์

- 4) เมื่อ Install เรียบร้อยแล้ว ก็สามารถใช้ Library ในโปรเจกต์ได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา

วันที่ 9 เดือน กุมภาพันธ์ พ.ศ. 2566

ข้าพเจ้า นางสาว ฐิติรัตน์ ผสมทรัพย์ รหัสประจำตัว 62050147

นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์ ภาควิชา วิทยาการคอมพิวเตอร์

ขอรับรองว่าสหกิจศึกษา เรื่อง

ชื่อภาษาไทย การพัฒนาฟีเจอร์การสนทนาสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเชน

ชื่อภาษาอังกฤษ THE DEVELOPMENT OF CHAT FEATURE FOR CUSTOMERS ON CLOUD KITCHEN PLATFORM

ปีการศึกษา 2565

เป็นผลงานวิจัยที่มีได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มสหกิจศึกษาระดับสมบูรณ์แล้ว

โปรแกรมอักขราวิสุทธิ์ 0.25 % หรือโปรแกรม Turnitin %

ลงชื่อ ฐิติรัตน์

(ฐิติรัตน์ ผสมทรัพย์)

นักศึกษา

ข้าพเจ้า ผศ.ดร. อินทรพร อรัณยะนาค อาจารย์ที่ปรึกษาสหกิจศึกษาได้ตรวจสอบสหกิจศึกษาของนักศึกษา

ข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ อินทรพร อรัณยะนาค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า อาจารย์ที่ปรึกษา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำรับรองเล่มสหกิจศึกษาโดยสถานประกอบการ

วันที่ 10 เดือน มีนาคม พ.ศ. 2566

ข้าพเจ้า นายทรงวุฒิ ชัยประเสริฐศรี ตำแหน่ง Head of Mobile Developer เป็นตัวแทนของสถานประกอบการ บริษัท ฟูลทิม จำกัด ขอรับรองว่า ทางสถานประกอบการได้ตรวจสอบเล่มสหกิจศึกษา เรื่อง การพัฒนาฟีเจอร์การสนทนาสำหรับลูกค้าผู้ใช้บริการบนแพลตฟอร์มคลาวด์คิทเช่น ของนักศึกษาชื่อ นางสาวฐิติรัตน์ ผสมทรัพย์ ซึ่งเป็นนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เรียบร้อยแล้ว และไม่มีส่วนหนึ่งส่วนใดในเล่มสหกิจศึกษาที่มีข้อมูลอ่อนไหว และ/หรือ ข้อมูลอันเป็นความลับอันจะก่อให้เกิดความเสียหายต่อสถานประกอบการ รวมทั้งอนุญาตให้สามารถเผยแพร่ต่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ **ทรงวุฒิ ชัยประเสริฐศรี**

(ทรงวุฒิ ชัยประเสริฐศรี)

ตัวแทนสถานประกอบการ

ข้าพเจ้า ผศ.ดร.อินทราพร อรัญยะนาค อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบเล่มสหกิจศึกษาแล้ว และรับทราบว่าสถานประกอบการดำเนินการตรวจสอบเล่มสหกิจศึกษาแล้ว จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ **อินทราพร อรัญยะนาค**

(อินทราพร อรัญยะนาค)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้