

การพัฒนาฟีเจอร์การสนทนาบนแอปพลิเคชันสำหรับ
ผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิทเชน

THE DEVELOPMENT OF CHAT FEATURES
ON MOBILE APPLICATION FOR MERCHANT PARTNERS
OF CLOUD KITCHEN PLATFORM



สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาวิทยาศาสตรบัณฑิต (สาขาวิชาวิทยาการคอมพิวเตอร์)
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ปีการศึกษา 2565
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DEVELOPMENT OF CHAT FEATURES
ON MOBILE APPLICATION FOR MERCHANT PARTNERS
OF CLOUD KITCHEN PLATFORM



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

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาตของเอกสารทุกครั้งที่มีการนำไปใช้

ACADEMIC YEAR 2022

หัวข้อสหกิจศึกษา

การพัฒนาฟีเจอร์การสนทนาบนแอปพลิเคชัน
สำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิทเชิน
THE DEVELOPMENT OF CHAT FEATURES
ON MOBILE APPLICATION FOR MERCHANT PARTNERS
OF CLOUD KITCHEN PLATFORM

ชื่อนักศึกษา

นายสรายุทธ วจิตตาภิบาล รหัสนักศึกษา 62050234

ปริญญา

วิทยาศาสตรบัณฑิต (สาขาวิชาวิทยาการคอมพิวเตอร์)

ภาควิชา

วิทยาการคอมพิวเตอร์

ปีการศึกษา

2565

อาจารย์ที่ปรึกษา

ผศ.ดร.อินทราพร อรัณยะนาค

คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.) อนุมัติ
ให้สหกิจศึกษานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต (สาขาวิชา
วิทยาการคอมพิวเตอร์) ประจำปีการศึกษา 2565

คณะกรรมการสอบ	ลายมือชื่อ
ผศ.ดร.อินทราพร อรัณยะนาค กรรมการและอาจารย์ที่ปรึกษา	อินทราพร อรัณยะนาค
ผศ.ดร.นวลสวาท หิรัญสกุลวงศ์ กรรมการ	นวลสวาท หิรัญสกุลวงศ์

ลิขสิทธิ์ของคณะวิทยาศาสตร์
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อสหกิจศึกษา	การพัฒนาฟีเจอร์การสนทนาบนแอปพลิเคชัน สำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิทเชน
ชื่อนักศึกษา	นายสรณฤกษ์ จงจิตตาภิบาล รหัสนักศึกษา 62050234
ปริญญา	วิทยาศาสตร์บัณฑิต (สาขาวิชาวิทยาการคอมพิวเตอร์)
ภาควิชา	วิทยาการคอมพิวเตอร์
คณะ	วิทยาศาสตร์
มหาวิทยาลัย	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง (สจล.)
ปีการศึกษา	2565
อาจารย์ที่ปรึกษา	ผศ.ดร.อินทราพร อรัณยະนาค

บทคัดย่อ

โครงการสหกิจศึกษานี้ มีวัตถุประสงค์เพื่อร่วมพัฒนาแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ซึ่งเป็นแพลตฟอร์มสำหรับการสั่งซื้อและการจัดส่งอาหาร มีเป้าหมายเพื่อให้บริการจัดส่งอาหารให้แก่ลูกค้าที่ไม่สะดวกเดินทางไปทานอาหารที่ร้าน และเป็นการเพิ่มโอกาสทางการค้าแก่ผู้ประกอบการให้มีช่องทางการขายสินค้าได้มากขึ้น จากการร่วมพัฒนา ผู้จัดทำได้ทำการศึกษา ค้นคว้า พัฒนา และทดสอบฟีเจอร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า (Chat Features for Merchant Application) ที่ทำงานอยู่บนระบบปฏิบัติการแอนดรอยด์ (Android) ซึ่งเป็นส่วนหนึ่งของระบบการสนทนา (Chat System) บนแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) เพื่อสร้างช่องทางการติดต่อสื่อสารระหว่างผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มกับลูกค้าที่เข้ามาสั่งซื้ออาหารที่สะดวก ผ่านการรับส่งข้อความอักษรและข้อความรูปภาพ เป็นการสนับสนุนผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มให้สามารถบริการลูกค้าที่เข้ามาสั่งซื้ออาหารได้ดียิ่งขึ้น

คำสำคัญ : แพลตฟอร์มคลาวด์คิทเชน, ฟีเจอร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า, ระบบการสนทนา, ระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	THE DEVELOPMENT OF CHAT FEATURES ON MOBILE APPLICATION FOR MERCHANT PARTNERS OF CLOUD KITCHEN PLATFORM
Students	Mr.Sranrork Jongiitapibal Student ID 62050234
Degree	Bachelor of Science (Computer Science)
Department	Computer Science
Faculty	Science
University	King Mongkut's Institute of Technology Ladkrabang (KMITL)
Academic Year	2022
Advisor	Asst.Prof.Dr.Inthraporn Aranyanak

Abstract

The purpose of this cooperative education is to contribute to the development of a cloud kitchen platform for food ordering and delivery. The goal is to offer food delivery services to customers who are unable to dine in restaurants and to expand business opportunities for entrepreneurs by providing additional sales channels. This study provides a chat feature for the merchant application on Android operating system. This chat feature serves as a communication channel between entrepreneurs and customers, allowing for better customer service through text and picture messaging.

Keywords : Cloud kitchen platform, Chat features for the merchant application, Chat system, Android operating system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำสหกิจศึกษาครั้งนี้ บรรลุผลสำเร็จลุล่วงไปได้ด้วยดี โดยได้รับความกรุณา ความช่วยเหลือและการสนับสนุน จากบุคลากรหลายท่าน ผู้จัดทำขอขอบพระคุณบุคคลต่าง ๆ ที่ได้ให้ความช่วยเหลือและเสียสละเวลาในการให้คำแนะนำปรึกษาโดยตลอด อันได้แก่

ครอบครัวอันได้แก่ บิดา มารดา ที่คอยสนับสนุนในด้านการศึกษาและตลอดจนปัจจัยด้านอื่น ๆ ทั้งการดูแลร่างกาย การเป็นกำลังจิตใจที่สำคัญ

ผศ.ดร.อินทราพร อรัณยธาดา และ ผศ.ดร.นวลสวาท หิรัญสกุลวงศ์ ที่ได้สละเวลาให้คำแนะนำ ปรึกษา ให้ข้อเสนอแนะและวิธีแก้ไขปัญหาต่าง ๆ ที่เกิดขึ้นตลอดในการจัดทำสหกิจศึกษานับนี้ อีกทั้งยังคอยติดตามความคืบหน้าของการดำเนินงานของสหกิจศึกษาครั้งนี้ และช่วยตรวจสอบแก้ไข เพื่อให้บรรลุผลสำเร็จไปได้ด้วยดี

บริษัท ฟูลทิม จำกัด และพี่เลี้ยงระหว่างที่ได้สหกิจศึกษาที่ได้มอบโอกาสและประสบการณ์อันดี ในการเป็นนักศึกษาฝึกงานในตำแหน่งผู้พัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ รวมทั้งให้ความรู้ในการทำงานภายในองค์กร ความรู้ทางด้านวิทยาการคอมพิวเตอร์ และชีวิตในการทำงาน ตลอดจนการดำเนินสหกิจศึกษา เพื่อให้สามารถนำไปใช้ได้จริง มีประสิทธิภาพ และทำให้ผลงานที่เกิดขึ้น สามารถปฏิบัติงานได้อย่างราบรื่น

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

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

สราญฤกษ์ จงจิตตาภิบาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
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 ระบบปฏิบัติการแอนดรอยด์ (Android operating system).....	3
2.2 การพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์.....	4
2.3 การพัฒนาแอปพลิเคชันตามแนวทางการพัฒนาแบบ MVM.....	5
2.4 รูปแบบโครงสร้างของโปรเจกต์แบบ Architecture Components.....	8
2.5 พื้นฐานการสร้างส่วนต่อประสานผู้ใช้บนแอปพลิเคชันแอนดรอยด์.....	9
2.6 การแสดงผลรายการและการทำงานของอะแดปเตอร์ (Adapter).....	10
2.7 การสื่อสารแลกเปลี่ยนข้อมูลผ่าน RESTful Web service.....	11
2.7.1 HTTP Request.....	11
2.7.2 HTTP Response.....	11
2.8 การสื่อสารแลกเปลี่ยนข้อมูลผ่าน Socket.IO.....	12
2.9 การทดสอบระบบสำหรับแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์.....	13
2.9.1 การทดสอบระบบแบบอัตโนมัติสำหรับแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์..	13
2.9.2 การเขียนการทดสอบสำหรับการทดสอบอัตโนมัติแบบ Unit Test.....	14
2.10 เครื่องมือที่ใช้ในการพัฒนา.....	15
2.10.1 Android Studio.....	15
2.10.2 Kotlin for Android.....	15
2.10.3 Sourcetree.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุขัดแย้งและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

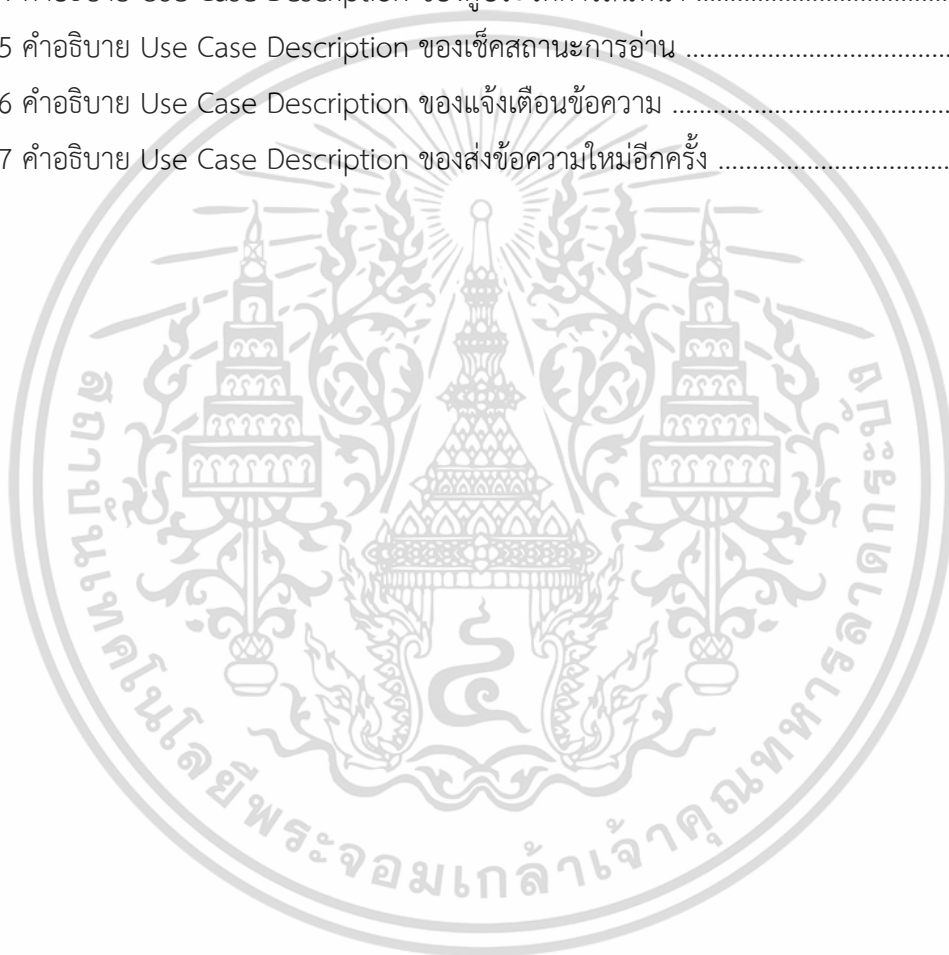
สารบัญ (ต่อ)

	หน้า
2.10.4 Firebase.....	17
2.10.5 Postman.....	17
2.10.6 Room Persistence Library.....	18
2.10.6 Figma	18
2.11 กระบวนการพัฒนาซอฟต์แวร์แบบไจล์ (Agile).....	18
บทที่ 3 วิธีการดำเนินงานวิจัย	20
3.1 การวางแผนและการวิเคราะห์ระบบ.....	20
3.1.1 วิเคราะห์ความต้องการของผู้ใช้และความต้องการของระบบ	20
3.1.2 วิเคราะห์ระบบงานเดิม	21
3.2 การการออกแบบระบบ	23
3.2.1 แผนภาพการแสดงผลการทำงานของผู้ใช้ระบบ (Use Case Diagram).....	23
3.2.2 แผนภาพที่แสดงขั้นตอนการทำงานและส่วนต่าง ๆ ของแอปพลิเคชัน.....	27
3.2.3 แผนภาพที่แสดงถึงการปฏิสัมพันธ์ (Sequence Diagram)	30
บทที่ 4 ผลการพัฒนาและการอภิปรายผล	40
4.1 ส่วนต่อประสานผู้ใช้.....	41
4.1.1 ภายในห้องสนทนา.....	41
4.1.2 หน้ารายละเอียดคำสั่งซื้อ.....	53
4.1.3 การแจ้งเตือน	54
4.2 การทดสอบระบบ	56
4.2.1 การทดสอบ ViewModel.....	57
4.2.2 การทดสอบ Adapter.....	60
บทที่ 5 สรุปผลการพัฒนาและข้อเสนอแนะ.....	65
5.1 สรุปผลการพัฒนา	65
5.2 ข้อเสนอแนะ	65
เอกสารอ้างอิง	67
ภาคผนวก.....	68
ภาคผนวก ก.....	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางเปรียบเทียบไวยากรณ์พื้นฐานระหว่างภาษา Java และภาษา Kotlin	16
3.1 คำอธิบาย Use Case Description ของส่งข้อความตัวอักษร	13
3.2 คำอธิบาย Use Case Description ของส่งข้อความรูปภาพ	24
3.3 คำอธิบาย Use Case Description ของพรีวิรูรูปภาพ	25
3.4 คำอธิบาย Use Case Description ของดูประวัติการสนทนา	25
3.5 คำอธิบาย Use Case Description ของเช็คสถานะการอ่าน	25
3.6 คำอธิบาย Use Case Description ของแจ้งเตือนข้อความ	26
3.7 คำอธิบาย Use Case Description ของส่งข้อความใหม่อีกครั้ง	26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ภาพแสดงการกระจายตัวของเวอร์ชันระบบปฏิบัติการแอนดรอยด์	3
2.2 ภาพแสดงโปรแกรมจำลองอุปกรณ์แอนดรอยด์	4
2.3 แผนภาพแสดงแนวคิดของ MVC	5
2.4 แผนภาพแสดงรูปแบบการพัฒนาแอปพลิเคชันแบบ MVC	6
2.5 แผนภาพแสดงแนวคิดของ MVVM	7
2.6 แผนภาพแสดงรูปแบบการพัฒนาแอปพลิเคชันแบบ MVVM	7
2.7 แผนภาพแสดงการแบ่งชั้นของส่วนต่าง ๆ ตามแนวคิด Architecture Components	8
2.8 แผนภาพการทำงานของส่วนต่อประสานผู้ใช้	9
2.9 แผนภาพการทำงานของลิสต์อะแดปเตอร์ (List Adapter)	10
2.10 แผนภาพพีระมิดของการทดสอบระบบ	11
2.11 ภาพการพัฒนาแอปพลิเคชันบน Android Studio	15
3.1 แผนภาพโครงสร้างของพื้นฐานของแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า	21
3.2 แผนภาพการแสดงผลการทำงานของผู้ใช้ระบบการสนทนา	23
3.3 แผนภาพโครงสร้างของพีเจอร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า	27
3.4 แผนภาพ Sequence Diagram ของการดูประวัติการสนทนา	30
3.5 แผนภาพ Sequence Diagram ของการโหลดประวัติการสนทนาใหม่อีกครั้ง	31
3.6 แผนภาพ Sequence Diagram ของการโหลดประวัติการสนทนาเพิ่ม	32
3.7 แผนภาพ Sequence Diagram ของการส่งข้อความอักษร	33
3.8 แผนภาพ Sequence Diagram ของการส่งข้อความรูปภาพ	34
3.9 แผนภาพ Sequence Diagram ของการส่งข้อความอักษรใหม่อีกครั้ง	35
3.10 แผนภาพ Sequence Diagram ของการส่งข้อความรูปภาพใหม่อีกครั้ง	36
3.11 แผนภาพ Sequence Diagram ของการระบุสถานะการอ่าน	37
3.12 แผนภาพ Sequence Diagram ของการแจ้งเตือนข้อความใหม่	38
3.13 แผนภาพ Sequence Diagram ของการแสดงผลปุ่มเข้าสู่ห้องสนทนาพร้อมด้วยจุดสีแดง	39
4.1 ห้องสนทนา กรณีคำสั่งซื้ออยู่ภายในระยะเวลา 24 ชั่วโมง	41
4.2 ห้องสนทนา กรณีคำสั่งซื้อที่มีระยะเวลามากกว่า 24 ชั่วโมง	42
4.3 ห้องสนทนา เมื่อสัญญาณอินเทอร์เน็ตขาดข้อง	43
4.4 การพิมพ์ข้อความ	44
4.5 การส่งข้อความ	45

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกา
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.6 การแสดงสถานะการอ่านของข้อความอักษร	46
4.7 ข้อความอักษรที่ถูกส่งไม่สำเร็จ	47
4.8 การเลือกรูปภาพ	48
4.9 การส่งข้อความรูปภาพ	49
4.10 การแสดงสถานะการอ่านของข้อความรูปภาพ	50
4.11 การแสดงผลรูปภาพเต็มจอ	51
4.12 ข้อความรูปภาพที่ถูกส่งไม่สำเร็จ	52
4.13 ข้อความรูปภาพที่ถูกส่งไม่สำเร็จ	53
4.14 การแจ้งเตือนข้อความใหม่	54
4.15 การจัดกลุ่มของการแจ้งเตือน	55
ก.1 หน้าเว็บไซต์สำหรับการดาวน์โหลด JetBrains Toolbox	68
ก.2 หน้าต่างเริ่มต้นการติดตั้งโปรแกรม JetBrains Toolbox	69
ก.3 หน้าต่างเมื่อการติดตั้งโปรแกรม JetBrains Toolbox	69
ก.4 หน้าต่างโปรแกรม JetBrains Toolbox	70
ก.5 Android Studio ที่ยังไม่ถูกติดตั้งในหน้าต่างโปรแกรม JetBrains Toolbox	70
ก.6 Android Studio ที่ถูกติดตั้งแล้วในหน้าต่างโปรแกรม JetBrains Toolbox	71
ก.7 หน้าต่างการเริ่มต้นการทำงานของ Android Studio	71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

บริษัทหนึ่งได้จัดทำแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ซึ่งเป็นแพลตฟอร์มสำหรับการสั่งซื้อและการจัดส่งอาหาร มีเป้าหมายเพื่อให้บริการจัดส่งอาหารให้แก่ลูกค้าที่ไม่สะดวกเดินทางไปทานอาหารที่ร้าน และเป็นการเพิ่มโอกาสทางการค้าแก่ผู้ประกอบการให้มีช่องทางการขายสินค้าได้มากขึ้น ในส่วนของบริการจัดส่งอาหารนั้น ทางแพลตฟอร์มมีทางเลือกให้แก่ผู้ประกอบการร้านค้าสามารถเลือกใช้บริการผู้ขนส่งอาหารจากทางแพลตฟอร์มหรือเลือกใช้บริการขนส่งอาหารของตนเองได้

โดยแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) เป็นแพลตฟอร์มที่รองรับการใช้งานผ่านแอปพลิเคชันทั้งหมด 3 แอปพลิเคชัน ได้แก่ 1) แอปพลิเคชันสำหรับผู้ให้บริการ (Consumer Application) ซึ่งมีไว้สำหรับลูกค้าผู้ให้บริการที่ต้องการสั่งอาหารจากร้านอาหารต่างๆ 2) แอปพลิเคชันสำหรับร้านค้า (Merchant Application) สร้างขึ้นเพื่อต้องการให้ร้านค้าที่เข้าร่วมแพลตฟอร์มสามารถสร้างหน้าร้านและเมนูต่าง ๆ ให้ลูกค้าได้เลือกซื้อ และ 3) แอปพลิเคชันสำหรับผู้จัดส่ง (Rider Application) เพื่อใช้สำหรับติดต่อรับข้อมูลที่อยู่ในการจัดส่งอาหาร

ทั้งนี้แพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ยังไม่สมบูรณ์ดีนักและยังต้องพัฒนาอีกหนึ่งสำคัญ คือ ระบบการสนทนาระหว่างผู้ประกอบการร้านค้า และลูกค้าผู้ให้บริการ เพื่อใช้ในการติดต่อสื่อสารรวมถึงการแจ้งรายละเอียดเกี่ยวกับรายการอาหารที่ได้ทำการสั่งซื้อ ดังนั้นจึงต้องมีการพัฒนาฟีเจอร์การสนทนาทั้งบนแอปพลิเคชันสำหรับลูกค้าผู้ให้บริการและแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า

ผู้จัดทำได้เล็งเห็นถึงความสำคัญ และประโยชน์ที่จะได้รับจากการพัฒนาระบบดังกล่าว จึงได้จัดทำฟีเจอร์การสนทนา (Chat Features) ในส่วนของระบบที่รองรับสำหรับแอปพลิเคชันของร้านค้า (Merchant Application) เพื่อให้ร้านค้าที่เข้าร่วมแพลตฟอร์มสามารถติดต่อกับลูกค้าผู้ให้บริการที่เข้ามาสั่งซื้ออาหารได้ ผ่านการรับส่งข้อความและรูปภาพ ซึ่งจะเป็นเครื่องมือที่ผู้ประกอบการร้านใช้ในการยกระดับการบริการให้ดียิ่งขึ้น

1.2 วัตถุประสงค์ของงานวิจัย

- 1) เพื่อพัฒนาคุณลักษณะการสนทนา (Chat Features) บนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า (Merchant Application)
- 2) เพื่อเพิ่มช่องทางสื่อสารให้ผู้ประกอบการร้านค้าใช้ติดต่อกับลูกค้าที่เข้ามาสั่งซื้ออาหารผ่านการรับส่งข้อความและรูปภาพ
- 3) เพื่อให้ผู้ประกอบการร้านค้าสามารถให้บริการหรือปรุงอาหารตามรายละเอียดต่างๆ ที่ได้รับจากลูกค้าผู้ใช้บริการ

1.3 ขอบเขตของงานวิจัย

- 1) พัฒนาฟีเจอร์การสนทนา สำหรับผู้ประกอบการร้านค้าเท่านั้น
- 2) พัฒนาฟีเจอร์ของแอปพลิเคชันบนมือถือ (Mobile Application) ที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ (Android)
- 3) แอปพลิเคชันบนมือถือ (Mobile Application) ทำงานผ่านเครือข่ายอินเทอร์เน็ต
- 4) ออกแบบและพัฒนาการทดสอบการทำงานของแอปพลิเคชันแบบอัตโนมัติ (Automate Test) ด้วยวิธีการยูนิทเทส (Unit Test)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1) แพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) แห่งนี้ได้รับคุณลักษณะการสนทนา (Chat Features)
- 2) ผู้ประกอบการร้านค้าสามารถติดต่อสื่อสารกับลูกค้าที่เข้ามาสั่งซื้ออาหารได้อย่างสะดวก
- 3) ผู้ประกอบการร้านค้าสามารถปรุงและส่งมอบอาหาร หรือให้บริการอื่นๆ ตามที่ลูกค้าผู้สั่งซื้ออาหารแจ้งรายละเอียดไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

การจัดทำสหกิจเล่มนี้เป็นการพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์สำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิทเซ็น ซึ่งจำเป็นต้องอาศัยความรู้และการเข้าใจทฤษฎีต่าง ๆ เพื่อใช้ในกระบวนการพัฒนาซึ่งได้มาจากการศึกษา ทบทวนเอกสาร รายงานการวิจัย และบทความ ในหัวข้อที่เกี่ยวข้องโดยมีรายละเอียดตามหัวข้อต่อไปนี้

2.1 ระบบปฏิบัติการแอนดรอยด์ (Android operating system)

แอนดรอยด์ (Android) คือระบบปฏิบัติการแบบเปิดเผยซอร์ฟแวร์ต้นฉบับ (Open Source) โดยบริษัท กูเกิล (Google Inc.) เป็นระบบปฏิบัติการที่ใช้งานได้บนอุปกรณ์ที่หลากหลาย สามารถทำงานบนอุปกรณ์ที่มีขนาดหน้าจอและความละเอียดที่ต่างกันได้ โดยทั่วไปแอนดรอยด์ (Android) มักเป็นที่รู้จักในฐานะระบบปฏิบัติการบนสมาร์ตโฟน ซึ่งทำให้ผู้บริโภคที่ต้องการเลือกซื้อโทรศัพท์มือถือได้มีตัวเลือกหลากหลายราคา แต่นอกจากนั้นแล้วแอนดรอยด์ยังนิยมใช้บนอุปกรณ์ต่าง ๆ ที่ไม่ใช่สมาร์ตโฟน อาทิเช่น แท็บเล็ต สมาร์ตทีวี สมาร์ตวอช หรือแม้แต่กระทั่งบนเครื่องชำระเงินและตู้จำหน่ายสินค้าอัตโนมัติ เป็นต้น ซึ่งในปัจจุบันระบบปฏิบัติการแอนดรอยด์ถูกพัฒนามาแล้วทั้งหมดหลายเวอร์ชัน โดยมีการกระจายตัวของเวอร์ชันที่ใช้งานดังรูปที่ 2.1 ต่อไปนี้

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 - Ice Cream Sandwich	16	99.9%
4.2 - Jelly Bean	17	99.7%
4.3 - Jelly Bean	18	99.7%
4.4 - KitKat	19	98.8%
5.0 - Lollipop	21	98.4%
5.1 - Lollipop	22	96.2%
6.0 - Marshmallow	23	92.7%
7.0 - Nougat	24	90.4%
7.1 - Nougat	25	88.2%
8.0 - Oreo	26	85.2%
8.1 - Oreo	27	77.3%
9.0 - Pie	28	62.8%
10 - Q	29	40.5%
11 - R	30	13.5%
12 - S	31	

รูปที่ 2.1 ภาพแสดงการกระจายตัวของเวอร์ชันระบบปฏิบัติการแอนดรอยด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

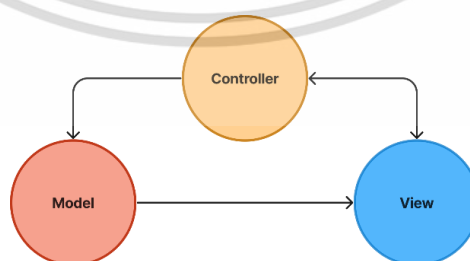
3) Resource ไฟล์นามสกุล .xml เป็นไฟล์เก็บทรัพยากรต่างๆ ที่จะถูกเรียกใช้โดย Java Class หรือ Kotlin Class เช่น โครงสร้างของส่วนต่อประสานผู้ใช้ รูปภาพ ค่าสี รูปแบบตัวอักษร รวมถึงการกำหนดนำทางไปยังหน้า Activity หรือหน้า Fragment ต่าง ๆ ภายในแอปพลิเคชัน

2.3 การพัฒนาแอปพลิเคชันตามแนวทางการพัฒนาแบบ MVVM

ปัญหาที่มักพบได้บ่อยในการพัฒนาโปรแกรมคือเมื่อโปรแกรมมีความซับซ้อนและมีชุดคำสั่งจำนวนมากขึ้นทำให้ยากต่อการดูแล สมาชิกในทีมเกิดความลำบากในการทำงานร่วมกัน และในกรณีที่โปรแกรมต้องการพัฒนาให้มีฟังก์ชันในการทำงานเพิ่มขึ้นก็ทำได้ยาก เนื่องจากชุดคำสั่งจากส่วนต่างๆ อาจทำงานซ้ำซ้อนและสร้างความสับสน เพราะแต่ละคนมีรูปแบบการเขียนโค้ดไม่เหมือนกันและตัวโปรแกรมไม่ได้ถูกพัฒนาไปในทิศทางเดียวกัน

จากปัญหาที่กล่าวไปข้างต้นจึงเกิดเป็นสิ่งที่เรียกว่า Design patterns กล่าวคือรูปแบบในการเขียนโปรแกรมที่ถูกพัฒนาขึ้นมาเพื่อลดความซับซ้อนและเพิ่มความสะดวกในการเขียนโปรแกรม ทำให้ผู้พัฒนารวมถึงทีมงานสามารถเข้าใจสิ่งที่โปรแกรมต้องทำได้ง่ายขึ้น และสามารถไปพัฒนาโปรแกรมต่อไปในแนวทางเดียวกันได้ ในปัจจุบันมีรูปแบบและแนวทางในการพัฒนาต่าง ๆ ถูกคิดค้นออกแบบเพื่อใช้ในงานที่แตกต่างกันออกไป

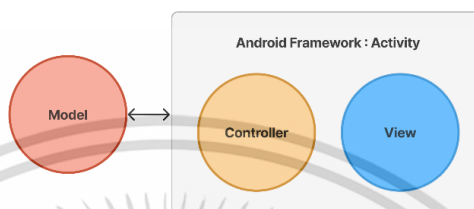
โดยรูปแบบที่เป็นที่นิยมและมักถูกแนะนำให้ใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์คือ การพัฒนาแบบ MVVM (ย่อมาจาก Model-View-ViewModel) ซึ่งเป็นรูปแบบที่ต่อยอดมาจาก MVC (Model-View-Controller) แต่รูปแบบ MVVM จะมีองค์ประกอบที่แตกต่างออกไปเล็กน้อยซึ่งจะไปด้วยสามส่วน ได้แก่ Model View ViewModel โดยส่วนทั้งสามแยกหน้าที่การทำงานอย่างชัดเจนลดความซ้ำซ้อนและความสับสน ต่างจากการพัฒนาแบบเดิมอย่าง MVC ที่เป็นการแยกส่วน Model และ View ออกจากกัน โดยมี Controller เป็นตัวกลาง ซึ่ง Controller จะจัดการข้อมูลใน Model ตามการกระทำที่ได้ที่จาก View แต่ตัว View ยังรับรู้ถึงการเปลี่ยนแปลงของข้อมูลผ่าน Model ได้โดยตรง ดังที่ปรากฏในรูปที่ 2.3



รูปที่ 2.3 แผนภาพแสดงแนวคิดของ MVC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อประยุกต์ใช้รูปแบบ MVC ในการพัฒนาแอปพลิเคชันบนแอนดรอยด์ ส่วน View และ ส่วน Controller จะทำงานเกี่ยวเนื่องกันบนคลาส Activity หรือ Fragment ของแอนดรอยด์ ดังรูปที่ 2.4 ส่งผลให้คลาส Activity ทำหน้าที่ติดต่อควบคุมส่วนต่อประสานผู้ใช้งานและการประมวลผล Business Logic ไปด้วย ทำให้โค้ดในคลาสดังกล่าวไม่ได้แยกการทำงานออกจากกันได้อย่างชัดเจน ทั้งยากต่อการดูแลและทำให้เกิดปัญหาในการต่อยอดฟังก์ชันการทำงาน



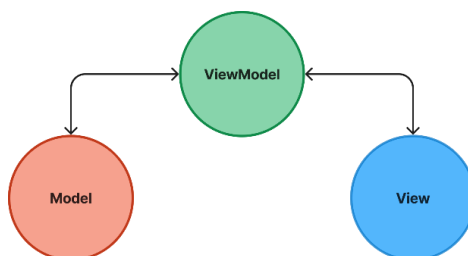
รูปที่ 2.4 แผนภาพแสดงรูปแบบการพัฒนาแอปพลิเคชันแบบ MVC

ถึงแม้รูปแบบการพัฒนาแบบ MVC จะสามารถเข้าใจการทำงานได้ไม่ยาก แต่ก็ไม่เหมาะกับโปรเจกต์ที่มีขนาดใหญ่ หรือโปรเจกต์ที่มีแนวโน้มที่จะเติบโตและขยายขนาดออกไปอีก ทั้งยังนำโค้ดจากส่วนใดส่วนหนึ่งไปใช้ได้อย่างยากเพราะว่า Model View และ Controller ต้องเชื่อมถึงกันเสมอ นอกจากนี้การพัฒนาแบบ MVC ยังไม่เอื้ออำนวยต่อการทดสอบระบบแบบอัตโนมัติอีกด้วย

รูปแบบการพัฒนาแบบ MVVM จึงเข้ามาแก้ไขปัญหาในการจัดการโค้ดให้ลดความซ้ำซ้อนและช่วยให้สามารถที่จะขยายตัวได้อย่างอิสระ โดยมีของแต่ละแนวคิดแบ่งการทำงานส่วนให้เป็นไปตามหน้าที่เท่านั้น ซึ่งจะแยกคลาสออกอย่างชัดเจนโดยสามารถจำแนกการทำงานของแต่ละส่วนได้ดังนี้

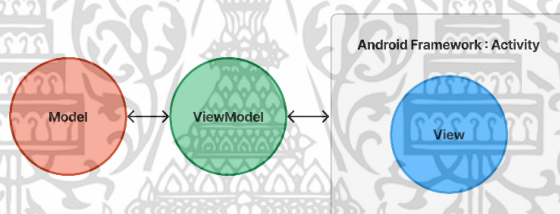
- 1) Model คือส่วนของข้อมูล เป็นส่วนที่เก็บข้อมูลของแอปพลิเคชันไว้ มีหน้าที่เกี่ยวกับตรรกะการจัดการข้อมูลต่างๆ
- 2) View คือส่วนที่ทำหน้าที่ต่อประสานผู้ใช้ ทำหน้าที่ควบคุมการแสดงผลและจัดการตรรกะที่ใช้ปฏิสัมพันธ์ตอบโต้กับผู้ใช้ เช่น กำหนดรูปลักษณ์ของสิ่งต่างๆ บนหน้าจอ หน้าตาของฟิเคชัน การเปลี่ยนหน้า การรับค่าจากผู้ใช้ การจัดการ Android life cycle
- 3) ViewModel คือตัวกลางระหว่างส่วน Model และส่วน View โดยมีหน้าที่ช่วยจัดการตรรกะต่างๆ และการรับส่งข้อมูลระหว่างส่วน Model และส่วน View และมีหน้าที่สำคัญคือการประมวลผลด้าน Business Logic อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แผนภาพแสดงแนวคิดของ MVVM

ในรูปแบบการพัฒนาแบบ MVVM นั้น ส่วน Model และส่วน View จะถูกแยกออกจากกันอย่างสิ้นเชิงตามที่ปรากฏในรูปที่ 2.5 และ 2.6 โดยมี ViewModel เป็นตัวกลาง กล่าวคือ Model และส่วน View จะไม่อ้างอิงถึงกันเลยและจะทำการรับส่งข้อมูลต่าง ๆ ผ่านตัว ViewModel เท่านั้น การแยกส่วนกันอย่างสิ้นเชิงเช่นนี้จะทำให้โค้ดส่วนต่าง ๆ เป็นอิสระต่อกัน แต่ละส่วนสามารถถูกนำไปใช้ซ้ำได้ง่าย ทั้งยังสามารถเพิ่มหรือต่อขยายส่วนหนึ่ง ๆ โดยที่ไม่กระทบส่วน ๆ อื่น



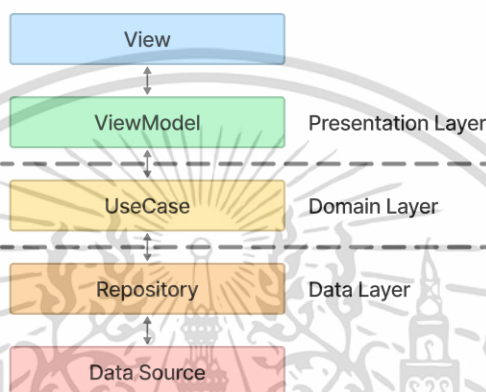
รูปที่ 2.6 แผนภาพแสดงรูปแบบการพัฒนาแอปพลิเคชันแบบ MVVM

และเมื่อส่วนต่าง ๆ เป็นอิสระต่อกันจึงทำให้สามารถทดสอบฟังก์ชันการทำงานของส่วนหนึ่ง ๆ แบบอัตโนมัติได้อีกด้วย นักพัฒนาสามารถเขียนชุดคำสั่งให้ทำการทดสอบแบบอัตโนมัติกับฟังก์ชันการประมวลผล Business logic ชุดคำสั่งดังกล่าวจะทำการส่งข้อมูลเข้าไปยังส่วนของ ViewModel โดยตรง และเปรียบเทียบผลลัพธ์ที่ได้ว่าเป็นไปคาดหวังหรือไม่ โดยไม่จำเป็นต้องรับข้อมูลผ่าน View การทำการทดสอบแบบอัตโนมัติเช่นนี้ช่วยให้สามารถประหยัดเวลาที่ใช้ในการทดสอบการทำงานของฟังก์ชันในกรณีต่างๆ ได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 รูปแบบโครงสร้างของโปรเจ็คแบบ Architecture Components

ในปี 2017 Google ได้เปิดตัวชุดพัฒนาที่มีชื่อเรียกว่า Architecture Components เพื่อเป็นแนวทางแนะนำสำหรับการกำหนดรูปแบบโครงสร้างของโปรเจ็คเพื่อให้สามารถดูแลและขยายขนาดออกไปได้ง่าย เป็นกำหนดรูปแบบในการทำงานร่วมกันระหว่างโค้ดแต่ละส่วน โดยมีแนวคิดสำคัญคือการแยกชั้น Layer การทำงานของแอปพลิเคชันให้โค้ดแต่ละส่วนแยกชั้นออกจากกันอย่างชัดเจน และแต่ละชั้นจะอ้างอิงถึงชั้นที่ติดกับตนเองเท่านั้น ดังที่ปรากฏในรูปที่ 2.7



รูปที่ 2.7 แผนภาพแสดงการแบ่งชั้นของส่วนต่าง ๆ ตามแนวคิด Architecture Components

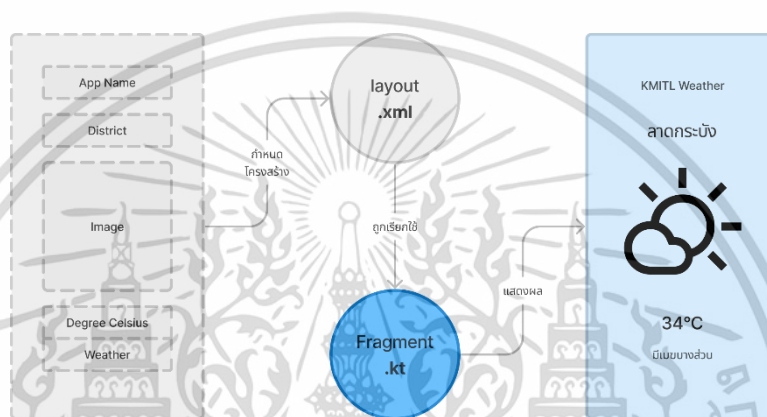
ตามแนวคิด Architecture Components โค้ดส่วนต่าง ๆ จะแบ่งออกเป็นทั้งหมด 3 ชั้นและแต่ละชั้นจะมีส่วนการทำงานดังต่อไปนี้

- 1) Presentation Layer เป็นชั้นบนสุดที่จะติดต่อกับผู้ใช้ รับข้อมูลจากชั้น Domain Layer หรือชั้น Data Layer ขึ้นมาเพื่อแสดงผลแก่ผู้ใช้ อีกทั้งทำหน้าที่ประมวล View Logic และ Business Logic รวมถึงส่งข้อมูลที่ประมวลแล้วส่งกลับไปยังชั้น Domain Layer
- 2) Domain Layer เป็นชั้นที่จะทำหน้าที่เกี่ยวกับ Data Logic โดยจะทำการจัดรูปแบบข้อมูลที่ได้จากชั้น Data Layer ให้อยู่ในรูปแบบที่สามารถใช้แสดงผลได้ก่อนจะส่งไปยังชั้น Presentation Layer และในทางกลับกันก็ต้องจัดการแปลงรูปข้อมูลที่ได้รับจากผู้ใช้ผ่านทาง Presentation Layer ให้สามารถส่งกลับไปยัง Data Layer ได้อีกด้วย
- 3) Data Layer เป็นชั้นที่ของข้อมูล มีหน้าที่จัดหาข้อมูลจากแหล่งข้อมูลต่างๆ ทั้งจาก Local Storage ภายในอุปกรณ์ และจาก Network / Service เพื่อส่งไปแสดงผลบนชั้น Presentation Layer รวมถึงส่งข้อมูลที่ได้จากชั้น Presentation Layer ไปยัง Local Storage หรือ Service ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 พื้นฐานการสร้างส่วนต่อประสานผู้ใช้บนแอปพลิเคชันแอนดรอยด์

สำหรับส่วนต่อประสานผู้ใช้ (User Interface) ของแอปพลิเคชันแอนดรอยด์นั้นจะประกอบขึ้นจากสองส่วน โดยส่วนแรกจะเป็นชุดคำสั่งที่ใช้กำหนดโครงสร้างของเลย์เอาต์ (Layout) ซึ่งจะถูกบรรจุอยู่ในไฟล์ประเภท .xml และส่วนที่สองคือชุดคำสั่งที่ถูกบรรจุอยู่ใน Kotlin Class ที่จะนำหน้าที่เรียกใช้เลย์เอาต์ (Layout) จากไฟล์ .xml มาแสดงผล รวมถึงกำหนดพฤติกรรมการแสดงผลและตรรกะต่าง ๆ ในการตอบโต้กับผู้ใช้ที่จะมีกับเลย์เอาต์ (Layout) ดังกล่าว ตามในรูปที่ 2.8



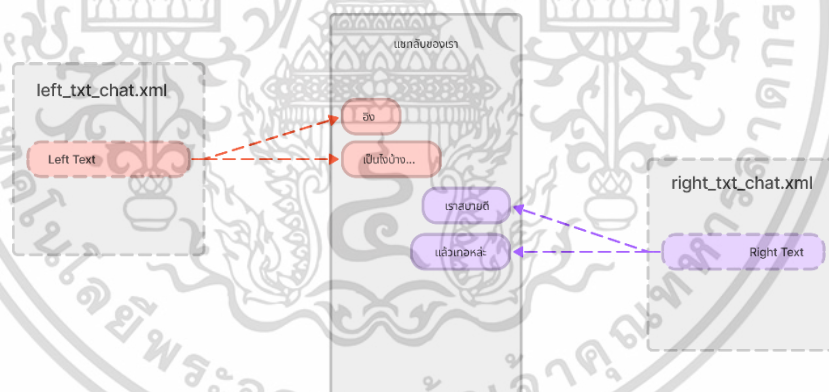
รูปที่ 2.8 แผนภาพการทำงานของส่วนต่อประสานผู้ใช้

โดยในหนึ่งหน้าการแสดงผล Kotlin Class สามารถเรียกใช้ layout.xml หลาย ๆ ไฟล์เพื่อช่วยกันกำหนดโครงสร้างส่วนต่าง ๆ ได้ และในหน้าการแสดงผลหนึ่ง ๆ ก็สามารถมี Kotlin Class หลาย ๆ คลาสช่วยกันกำหนดพฤติกรรมการแสดงผลได้ในส่วนที่ต่างกันได้ ซึ่งเป็นประโยชน์มากในการแบ่งหน้าที่ของแต่ละคลาสให้ชัดเจน สามารถแก้ไขเพิ่มเติมโครงสร้างเลย์เอาต์ (Layout) หรือชุดคำสั่งในส่วนใดส่วนหนึ่งของหน้าจอได้โดยที่ส่วนอื่น ๆ ไม่ได้รับผลกระทบ

2.6 การแสดงผลรายการและการทำงานของอะแดปเตอร์ (Adapter)

อะแดปเตอร์ (Adapter) คือ Kotlin Class ที่ทำงานร่วมกับส่วนต่อประสานอย่างหน้า Activity หรือหน้า Fragment ทำหน้าที่เป็น Controller ของส่วน View ซึ่งมีหน้าที่จัดการแสดงผลรายการของข้อมูลที่เป็นเรียงแถวต่อกัน กล่าวคือมีหน้าที่ในการนำเลย์เอาต์ (Layout) หลาย ๆ ตัวมาจัดเรียงบนหน้าจอและกำหนดพฤติกรรมของการแสดงผลของรายการเหล่านั้น โดยเลย์เอาต์ (Layout) ที่ถูกนำมาจัดเรียงบนหน้านั้นสามารถมีลักษณะแตกต่างกันตามไฟล์ .xml ที่จะถูกเรียกใช้ ซึ่ง Adapter จะเรียกใช้ไฟล์ .xml ไตมาแสดงผลก็ขึ้นอยู่กับข้อมูลที่ Adapter ได้รับว่าข้อมูลนั้นตรงตามเงื่อนไขใดที่ถูกกำหนดไว้

เพื่อให้สามารถเข้าใจการทำงานของ Adapter ได้มากยิ่งขึ้น จึงจะขอยกตัวอย่างการทำงานในการจัดเรียงประวัติการสนทนา โดยจะเป็นการสนทาระหว่างเราและคู่สนทนาเพียงสองคน ซึ่งประวัติการสนทนานั้นก็คือการที่กล่องข้อความหลาย ๆ กล่อง ถูกจัดเรียงจากบนลงล่างตามเวลา แต่จะมีเงื่อนไขสำคัญคือ โดยหากเป็นข้อความที่ถูกส่งจากเรา กล่องข้อความที่ปรากฏบนหน้าจอจะต้องอยู่ชิดทางขวา ในกลับกันหากเป็นข้อความที่ถูกส่งจากคู่สนทนา กล่องข้อความที่ปรากฏบนหน้าจอจะต้องอยู่ชิดทางซ้าย



รูปที่ 2.9 แผนภาพการทำงานของลิสต์อะแดปเตอร์ (List Adapter)

จากเงื่อนไขที่ถูกกำหนดไว้ข้างต้น Adapter จะทำการอ่านข้อมูลประวัติการสนทนาและเปรียบเทียบลักษณะของข้อมูลว่าข้อความไหนใครเป็นผู้ส่ง จากนั้นจึงเรียกใช้ไฟล์ .xml ที่ถูกเตรียมไว้มาแสดงผลกล่องข้อความตามเงื่อนไข ทำให้ได้ผลลัพธ์การจัดเรียงประวัติการสนทนาบนหน้าจออย่างที่เราเห็นกันทั่วไปในแอปพลิเคชันที่รับส่งข้อความการสนทนา ดังในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การสื่อสารแลกเปลี่ยนข้อมูลผ่าน RESTful Web service

RESTful หรือ REST Representation state transfer คือ Web Service ชนิดหนึ่งที่ใช้สื่อสารกันบนอินเทอร์เน็ตผ่านทางโพรโทคอล HTTP หรือ HTTPS การทำงานของ RESTful Web Service จะอาศัย URL ของคำสั่ง HTTP Request เพื่อค้นหาและประมวลผลแล้วตอบกลับ HTTP Response ไปในรูปแบบ XML หรือ JSON

2.7.1 HTTP Request

สำหรับคำสั่ง HTTP Request ที่ Client ส่งไปยัง Server เพื่อขอให้ประมวลผลและส่งข้อมูลตอบกลับมาจะประกอบด้วย

- 1) URL คือส่วนที่ระบุอยู่ของ Server ที่ Client ต้องการจะขอข้อมูล
- 2) HTTP Request Headers คือส่วนที่ระบุข้อมูลและกฎต่าง ๆ ในการเชื่อมต่อ
- 3) HTTP Method เป็นส่วนที่กำหนดประเภทของคำสั่ง Request จาก Client เป็นส่วนที่บอกว่า Client ต้องการทำอะไรกับข้อมูล เพื่อที่ Server จะได้ทำงานได้ตรงตามที่ต้องการ ซึ่งตอนออกแบบ RESTful Web service จะต้องกำหนดว่าการเข้าถึงข้อมูลแต่ละแบบจะให้ใช้ Method ใด โดยประเภทของคำสั่ง Request ที่ใช้กันทั่วไปจะมีอยู่ด้วยกัน 3 เมธอด ได้แก่ GET ใช้สำหรับขอข้อมูลจาก Server, PUT ใช้สำหรับคำขอที่ต้องการอัปเดตข้อมูล และ POST ใช้สำหรับคำขอที่ต้องการสร้างข้อมูลใหม่
- 4) HTTP Request Body คือส่วนที่ใช้ระบุรายละเอียดของคำสั่งขอข้อมูลนั้น ๆ สำหรับเมธอด GET มักเป็นรูปแบบของ Query String สำหรับใช้ขอข้อมูลที่ต้องการ ส่วนเมธอด PUT หรือ POST จะเป็นข้อมูลที่ Client ต้องการจะส่งให้ Server

2.7.2 HTTP Response

ส่วน HTTP Response ที่ Server ตอบกลับมาให้กับ Client ตามที่ได้ขอ Request มาก ซึ่งจะประกอบไปด้วยส่วนต่าง ๆ ดังนี้

- 1) HTTP Response Header คือส่วนระบุข้อมูลรายละเอียดที่อธิบายถึงคุณลักษณะของ Response ว่าข้อมูลมีขนาดไหน เป็นข้อมูลประเภทอะไร รวมถึง HTTP Status Code ที่เป็นชุดตัวเลขที่ใช้บ่งบอกสถานะของการทำงานว่า Request ที่ Client ส่งมานั้นทำงานสำเร็จหรือไม่อย่างไร โดย Status ที่มักพบเห็นได้บ่อยๆ เช่น 200 Ok มีความหมายว่า Request ดังกล่าวทำงานสำเร็จและข้อมูลถูกส่งกลับมาถึง Client อย่างถูกต้อง และ 500 Internal Server Error ซึ่งหมายความว่า เกิดข้อผิดพลาดที่ Server ทำให้ไม่สามารถส่งข้อมูลกลับมาตาม Request ที่ Client ขอมมาได้

- 2) HTTP Response Body คือส่วนที่บรรจุข้อมูลตามที่ Client ได้ Request ขอมมา โดยมักอยู่ในรูปของ XML หรือ JSON ซึ่งมีขนาดเล็กและ Extract ออกมาใช้งานได้ง่าย

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาดเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การสื่อสารแลกเปลี่ยนข้อมูลผ่าน Socket.IO

Socket.IO เป็นไลบรารีที่ใช้งานสำหรับการสื่อสารแลกเปลี่ยนข้อมูลบนอินเทอร์เน็ตผ่านโพรโทคอล WebSocket ซึ่งจะรับประกันความถูกต้องและลำดับของข้อมูลที่ถูกส่ง ทั้งยังเป็นการสื่อสารแบบสองทิศทาง กล่าวคือสามารถเป็นผู้รับและผู้ส่งได้ในเวลาเดียวกัน โดย Socket.IO มีหลักการทำงานไม่ซับซ้อน ส่งหรือรับข้อมูลผ่านอีเวนต์ (Event) โดยทั้งฝั่ง Server จะเป็นผู้กำหนด Socket Address ในรูปแบบของ IP Address ที่ตามด้วย Port Number จากนั้น Client จึงจะสามารถเชื่อมต่อเข้ามาผ่าน Socket Address ดังกล่าว ตามตัวอย่างด้านล่าง

```
socketAddress = http://www.example.com/chat:3000
```

กิจกรรมการรับส่งข้อมูลทั้งหมดผ่าน Socket.IO จะถูกทำบนอีเวนต์ Event ต่างๆ ซึ่งอีเวนต์คือชื่อของกิจกรรมที่ถูกกำหนดไว้ ทำให้ทั้งตัว Server และ Client ทราบดีว่าเมื่อเกิดกิจกรรมดังกล่าวขึ้นจะต้องดำเนินการอย่างไรต่อไป โดยจะใช้คำสั่ง Emit เพื่อทำการส่งอีเวนต์และข้อมูลไปยังฝั่งตรงข้าม และจะใช้คำสั่ง On ในการรอรับข้อมูลจากอีเวนต์นั้น ดังตัวอย่างชุดคำสั่งต่อไปนี้

กำหนดให้ “MESSAGE_FROM_CLIENT” เป็นชื่อของอีเวนต์ที่ Client จะทำการส่งข้อความไปยัง Server โดย Emit อีเวนต์ดังกล่าวไปพร้อมกับข้อความที่ Client ต้องการส่ง

```
clientSocket.emit("MESSAGE_FROM_CLIENT", "Hello, server! I am client.")
```

เมื่อ Server ได้รับอีเวนต์ “MESSAGE_FROM_CLIENT” และข้อความจาก Client ผ่านคำสั่ง On ให้ Server ทำการแสดงผลข้อความดังกล่าว

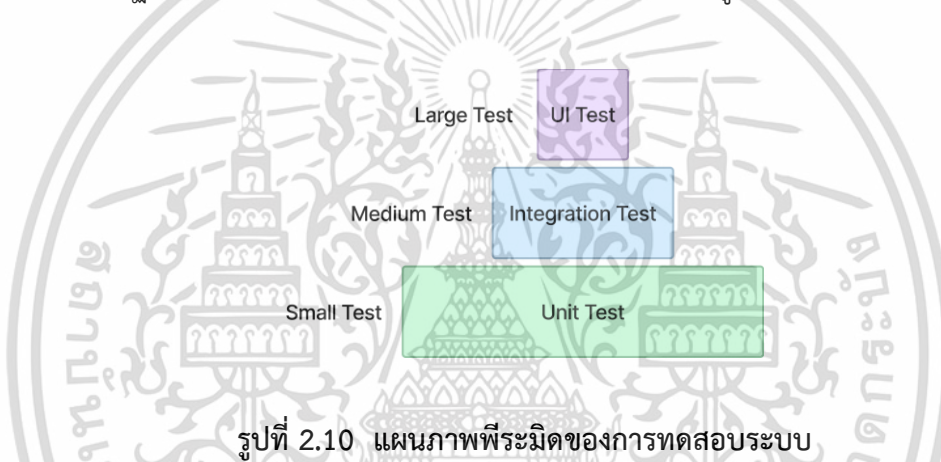
```
serverSocket.on("MESSAGE_FROM_CLIENT") { message ->
  print("Message form client: $message")
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 การทดสอบระบบสำหรับแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

2.9.1 การทดสอบระบบแบบอัตโนมัติสำหรับแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์

ในกระบวนการพัฒนาแอปพลิเคชันหนึ่ง ๆ นั้นจะต้องมีการทดสอบการทำงานของแอปพลิเคชันอยู่เรื่อย ๆ เพื่อให้การทำงานของแอปพลิเคชันมีข้อผิดพลาดน้อยที่สุด ซึ่งการทำการทดสอบแบบอัตโนมัติจะช่วยให้ผู้พัฒนาประหยัดทั้งเวลาในการทดสอบได้มาก นอกจากนี้ยังสามารถใช้ทดสอบการทำงานในกรณีต่าง ๆ ที่อาจจะไม่สะดวกในการทดสอบด้วยมืออีกด้วย โดยการทำการทดสอบแบบทดสอบแบบอัตโนมัติสำหรับแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์นั้นจะแบ่งออกได้เป็น 3 ระดับดังรูปที่ 2.10



รูปที่ 2.10 แผนภาพพีระมิดของการทดสอบระบบ

1) Small Test มักที่รู้จักในชื่อ Unit Test เป็นการทดสอบการทำงานของชุดคำสั่งในคลาสใดคลาสหนึ่ง เพื่อให้มั่นใจว่าชุดคำสั่งหรือฟังก์ชันต่าง ๆ สามารถทำงานได้ถูกต้องตามหน้าที่ของคลาสนั้นๆ โดยจะเป็นการเขียนชุดคำสั่งขึ้นมาชุดหนึ่งสำหรับการทดสอบแบบอัตโนมัติกับคลาสนั้นๆ ชุดคำสั่งดังกล่าวจะทำการส่งข้อมูลเข้าไปยังคลาสโดยตรง และเปรียบเทียบผลลัพธ์ที่ได้จากคลาสนั้นว่าเป็นไปคาดหวังหรือไม่

2) Medium Test หรือที่เรียกว่า Integration Test เป็นการทดสอบการทำงานของโค้ดหลายส่วนร่วมกัน เพื่อให้มั่นใจว่าชุดคำสั่งที่อยู่ต่างคลาสนั้น สามารถทำงานร่วมกันได้อย่างถูกต้อง เช่นการทำงานร่วมกันระหว่างส่วน View กับ ViewModel หรือการทำงานร่วมกันระหว่างส่วนต่าง ๆ ใน Data Layer โดยจะเป็นการเขียนชุดคำสั่งขึ้นมาชุดหนึ่งสำหรับการทดสอบแบบอัตโนมัติกับคลาสนั้นๆ ชุดคำสั่งดังกล่าวจะทำการป้อนข้อมูลเข้าไปยังคลาสใดคลาสหนึ่ง และรอดูผลลัพธ์ที่ได้จากกระบวนการทำงานของคลาสนั้นๆ ว่าเป็นไปตามที่คาดหวังหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Large Test หรือ UI Test เป็นการทดสอบการทำงานของชุดคำสั่งทั้งหมด โดยจำลองสถานการณ์ต่าง ๆ ให้เหมือนกับการใช้งานจริงของผู้ใช้งาน เพื่อให้มั่นใจว่าแอปพลิเคชันของเราสามารถทำงานได้ตามที่คาดหวังไว้

ซึ่งโดยพื้นฐานของการเขียนการทดสอบแล้ว นักพัฒนาควรเน้นไปที่เขียน Small Test เป็นหลัก และทำการทดสอบ Medium Test และ Large Test ตามกันไป เพราะหากแต่ละคลาสสามารถทำหน้าที่ของตนเองได้สมบูรณ์แล้ว การทำงานในภาพรวมก็จะสมบูรณ์ไปด้วย

2.9.2 การเขียนการทดสอบสำหรับการทดสอบอัตโนมัติแบบ Unit Test

การทดสอบการทดสอบอัตโนมัติแบบ Unit Test เป็นการทดสอบการทำงานของคลาสใดคลาสหนึ่งหรือส่วนใดส่วนหนึ่งของแอปพลิเคชันเพื่อการันตรีว่าการทำงานของส่วนนั้น ๆ หรือคลาสนั้น ๆ หน้าที่ของตนได้อย่างถูกต้อง ซึ่งในการทำทดสอบอัตโนมัติแบบ Unit Test ทุกกรณีนั้น ผู้พัฒนาจะต้องกำหนดสิ่งต่อไปนี้

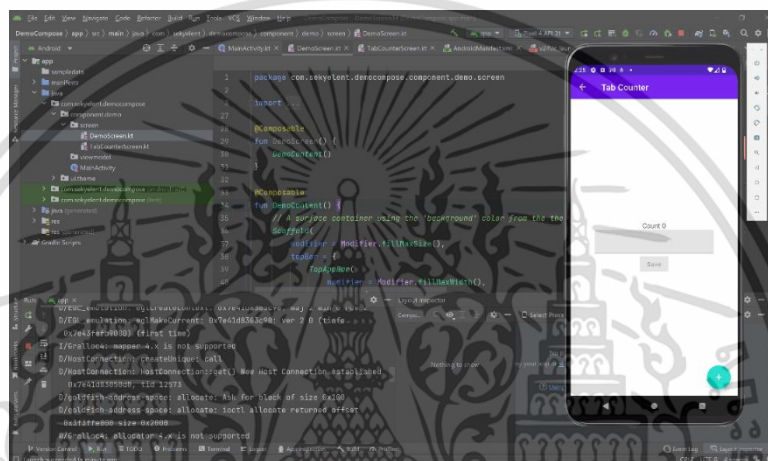
- 1) ส่วนการทำงาน (Unit) หรือคลาส (Class) ที่ต้องการทำการทดสอบ
- 2) ขั้นตอนการทดสอบ (Test Step)
- 3) ข้อมูลที่ใช้ในการทดสอบ (Test Data)
- 4) ผลลัพธ์ที่คาดหวัง (Expected Output)

เมื่อทำการทดสอบระบบอัตโนมัติแบบ Unit Test ส่วนการทำงาน (Unit) หรือคลาส (Class) ที่ต้องการทำการทดสอบจะถูกเรียกใช้และทำงานตามขั้นตอนการทดสอบ (Test Step) ซึ่งการทำงานฟังก์ชันต่าง ๆ จะรับข้อมูลที่ใช้ในการทดสอบ (Test Data) เพื่อทำการประมวลผลออกมาเป็นผลลัพธ์ที่ได้ (Actual Output) โดยผลลัพธ์ที่ได้ (Actual Output) ดังกล่าวนั้นจะถูกนำมาเปรียบเทียบกับผลลัพธ์ที่คาดหวัง (Expected Output) หากผลลัพธ์ที่ได้ (Actual Output) เป็นไปตามผลลัพธ์ที่คาดหวัง (Expected Output) ก็ถือว่าการทดสอบอัตโนมัติในกรณีนั้น ๆ ผ่านการทดสอบ ในทางกลับกัน หากผลลัพธ์ที่ได้ (Actual Output) เป็นไม่ไปตามผลลัพธ์ที่คาดหวัง (Expected Output) ถือว่าการทดสอบอัตโนมัติในกรณีนั้น ๆ ไม่ผ่านการทดสอบ

2.10 เครื่องมือที่ใช้ในการพัฒนา

2.10.1 Android Studio

Android Studio เป็น IDE Tool ที่ Google Inc. พัฒนาขึ้นเพื่อให้เป็นเครื่องมือหลักสำหรับการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการ Android โดยเฉพาะ ซึ่งกระบวนการที่สำคัญในการพัฒนาแอปพลิเคชันทั้งหมดสามารถถูกทำบน Android Studio ได้ทั้งหมด ตั้งแต่การเริ่มต้นสร้างโปรเจก การเขียนชุดคำสั่ง การออกแบบส่วนต่อประสานผู้ใช้ จนกระทั่งการทดสอบแบบอัตโนมัติ ซึ่ง Android Studio มีรูปลักษณะการใช้งานตามรูปที่ 2.11



รูปที่ 2.11 ภาพการพัฒนาแอปพลิเคชันบน Android Studio

2.10.2 Kotlin for Android

Kotlin เป็นภาษาโปรแกรมมิ่งยุคใหม่ที่มีความสามารถหลากหลาย สามารถใช้พัฒนา Web Application Development, Backend Development และ Mobile Application Development ทั้งบนระบบปฏิบัติการแอนดรอยด์ (Android) และระบบปฏิบัติการไอโอเอส (iOS) โดยเฉพาะบนระบบปฏิบัติการแอนดรอยด์ (Android) นั้น อาจกล่าวได้ว่าการเข้ามาถึงของ Kotlin ถือว่าเป็นการปฏิวัติวงการการพัฒนาแอปพลิเคชันบนแอนดรอยด์ (Android Application) เลยก็ว่าได้ เพราะว่าเมื่อเทียบกับ Java ที่เป็นแต่เดิมเป็นภาษาหลักในการพัฒนาแอปพลิเคชันบนแอนดรอยด์ (Android Application) แล้ว Kotlin เป็นภาษายุคใหม่ที่ไวยากรณ์มีความกระชับ อ่านง่าย เข้าใจได้ง่ายกว่ามาก ทั้งยังมีคุณสมบัติต่าง ๆ ที่ทันสมัย ลบข้อด้อยต่าง ๆ ที่เคยมีในภาษา Java

โดยจะยกตัวอย่างเปรียบเทียบไวยากรณ์พื้นฐานของทั้งสองภาษาผ่านชุดคำสั่งที่ใช้ประกาศคลาสนักฟุตบอล (People class) ที่จะมีแอททริบิวต์ (Attribute) 1 ตัวคือ ชื่อ (name) และชุดคำสั่งสร้างออบเจกต์นักกีฬา (People object) จากคลาสข้างต้น

ตารางที่ 2.1 ตารางเปรียบเทียบไวยากรณ์พื้นฐานระหว่างภาษา Java และภาษา Kotlin

ชุดคำสั่งภาษา Java	ชุดคำสั่งภาษา Kotlin
<pre>public class People { private String name; public Student(String name) { this.name = name; } public String getName() { return name; } public void setName(String name) { this.name = name; } public int getId() { return id; } public void setId(Int id) { this.id = id; } }</pre>	<pre>data class People (val id: Int, val name: String)</pre>
<pre>public void creatAndPrintPeople () { String name = "สรายุทธชัย" int id = 62050234 Student st = new Student(id, name) System.out.printf("Student: %d %s", st.getId, st.getName()) }</pre>	<pre>fun creatAndPrintStudent() { val name = "สรายุทธชัย" val id = 62050234 val st = Student(id, name) print("st: \${ppl.id} \${ st.name}") }</pre>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.3 Sourcetree

Sourcetree คือซอฟต์แวร์ Git Client เป็นเครื่องมือที่ช่วยจัดการเวอร์ชันของโค้ดให้กับนักพัฒนาและทีม โดยจะเก็บประวัติว่าไฟล์แต่ละไฟล์ถูกสร้าง ลบ แก้ไขโดยใคร เมื่อไหร่ และอย่างไรเอาไว้ทั้งหมด ทำให้นักพัฒนาสามารถและทีมติดตามการเปลี่ยนแปลงของโค้ดได้ตลอด ช่วยให้นักพัฒนาหลาย ๆ คนในทีมสามารถทำงานในแอปพลิเคชันเดียวได้อย่างต่อเนื่องเป็นเนื้อเดียวกัน Sourcetree นั้นเป็น Git GUI Client ที่จะมีหน้าต่างรูปลักษณะที่เข้าใจง่าย ใช้งานสะดวกกว่า Git Client แบบดั้งเดิมที่จะต้องเข้าใช้งานผ่าน CMD หรือ Terminal ที่จะต้องพิมพ์คำสั่งต่าง ๆ ด้วยตนเอง Sourcetree มีแผนผังแสดงประวัติที่ชัดเจนและแสดงรายละเอียดการเปลี่ยนแปลงของโค้ดในแต่ละช่วงเวลา หรือหากจากจะใช้งานคำสั่ง Git ระดับสูงก็สามารถเรียกใช้ Terminal ผ่าน Sourcetree ได้เช่นกัน

2.10.4 Firebase

Firebase คือแพลตฟอร์มที่ให้บริการเกี่ยวกับระบบหลังบ้าน (Back-end as a Service) กล่าวคือเป็นบริการระบบหลังบ้านที่มีเครื่องมือต่าง ๆ ให้พร้อมใช้งาน ซึ่ง Firebase นั้นนิยมใช้กับ Mobile Application ซึ่ง Firebase จะมีทั้งเครื่องมือระบบหลังบ้านที่สามารถใช้งานได้ฟรีไม่มีค่าใช้จ่าย และเครื่องมือที่ค่าใช้จ่ายสำหรับการขยายกำลังประมวลผลเพื่อใช้งานกับข้อมูลจำนวนมาก

ในแง่ของการพัฒนาระบบการสนทนานั้น จะใช้เครื่องมือตัวหนึ่งของ Firebase ที่ใช้สำหรับการแจ้งเตือนข้อความใหม่ เครื่องมื่อดังกล่าวคือ Firebase Cloud Message (FCM) เป็นบริการส่งข้อความแจ้งเตือนแบบข้ามแพลตฟอร์ม (Cross platform messaging) ซึ่งจะช่วยให้ Mobile Application ทั้งระบบปฏิบัติการไอโอเอส (iOS) และระบบปฏิบัติการแอนดรอยด์ (Android) สามารถรับการแจ้งเตือนได้

2.10.5 Postman

Postman เป็นเครื่องมือสำหรับพัฒนา API หรือนักพัฒนาทั่วไปที่ต้องการใช้งาน API ในการพัฒนาเว็บไซต์ (Website) หรือแอปพลิเคชัน (Application) ของตน Postman นั้นเป็น API Testing Tool กล่าวคือ เป็นเครื่องมือที่ใช้ในการทดสอบการทำงานของ API โดยจะส่ง Request ไปเพื่อตรวจสอบ Response ต่าง ๆ ที่จะได้รับกลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.6 Room Persistence Library

Room เป็นฐานข้อมูล SQLite ที่ใช้สำหรับเก็บข้อมูลบนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ (Android App) เป็นไลบรารี (Library) ที่ช่วยให้สามารถเรียกใช้การทำงานต่าง ๆ ที่เกี่ยวข้องกับฐานข้อมูลให้สะดวกมากขึ้น โดย Room จะสร้างฐานข้อมูล Room Database เตรียมไว้เพื่อรอเก็บข้อมูลตามที่ต้องการ ตัวแอปพลิเคชัน (Application) ไม่จำเป็นต้องติดต่อกับฐานข้อมูลโดยตรง แต่จะติดต่อผ่านคลาส Data Access Object หรือ DAO แทน ซึ่ง DAO นั้นจะเป็นคลาสที่มีฟังก์ชันที่บรรจุคำสั่ง SQL ต่าง ๆ ไว้ให้คลาสอื่น ๆ เรียกใช้งานเพื่อจัดการกับข้อมูลต่าง ๆ ในฐานข้อมูล

2.10.6 Figma

Figma เป็นเครื่องมือสำหรับการออกแบบที่สามารถใช้ออกแบบได้ตั้งแต่เว็บไซต์ (Website) แอปพลิเคชัน (Application) โลโก้ (Logo) หรือ อาร์ตเวิร์ก (Artwork) ต่าง ๆ โดย Figma นั้นมีจุดเด่นสำคัญคือการทำงานร่วมกันเป็นทีม ทำให้ทั้งนักออกแบบ UX/UI สามารถส่งต่องานให้กับนักพัฒนาได้ง่าย ทางนักพัฒนาเองก็สามารถ Export อาร์ตเวิร์ก (Artwork) ต่าง ๆ มาใช้งานได้ทันที

2.11 กระบวนการพัฒนาซอฟต์แวร์แบบอไจล์ (Agile)

กระบวนการพัฒนาซอฟต์แวร์แบบอไจล์ (Agile Software Development Methodology) เป็นแนวทางและวิธีการการพัฒนาซอฟต์แวร์ที่ใช้กันอย่างแพร่หลายในปัจจุบัน มีแนวคิดสำคัญคือไม่เคร่งในระเบียบวิธีมากเกินไป ไม่เน้นการจัดทำเอกสาร แต่เน้นผลผลิตหรือตัวซอฟต์แวร์เป็นหลัก และให้ความสำคัญกับการสื่อสารในการทำงานร่วมกัน โดยจะแบ่งแผนงานและเวลาการพัฒนาออกเป็นช่วง ๆ แต่ละช่วงมีระยะเวลาไม่เกิน 1-4 สัปดาห์ เพื่อให้ยืดหยุ่น พร้อมทั้งรับความเปลี่ยนแปลง ลดความเสี่ยงในการพัฒนา เมื่อมีการเปลี่ยนแปลงหรืออุปสรรคเกิดขึ้นจะสามารถปรับแผนการพัฒนาได้อย่างทันท่วงที

โดยทั่วไปแล้ว Agile ไม่มีระเบียบวิธีตายตัว แล้วแต่ว่าจะนำไปปรับไปใช้ให้เหมาะกับองค์กรของตน Agile อาจจะเป็นส่วนขยายของกระบวนการพัฒนาซอฟต์แวร์อื่น ๆ ที่มีอยู่เดิมให้มีความยืดหยุ่นและประสิทธิภาพมากขึ้น ทั้งนี้อไจล์ Agile มีหลักการและแนวทางสำคัญ 4 ข้อ ดังต่อไปนี้

1) เน้นให้ความสำคัญกับทีมเวิร์ค สร้างสิ่งแวดล้อมให้สนับสนุนการสื่อสาร จัดให้สมาชิกในทีมมาจากหลายสายงาน เพื่อช่วยกันสร้างความเข้าใจในรายละเอียดของงานในแง่มุมต่าง ๆ

2) เน้นการสร้างซอฟต์แวร์ที่ใช้งานได้จริงมากกว่าการยึดถือเอกสารหรือรายงาน แบ่งโครงการเป็นงานย่อย ๆ เพื่อกำหนดเป้าหมายที่ชัดเจนในระยะเวลานั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เน้นการทำงานร่วมระหว่างนักพัฒนาและลูกค้าหรือผู้ใช้ตลอดโครงการ เนื่องจากฟีดแบค Feedback ของผู้ใช้คือสิ่งเดียวที่จะบอกได้ว่าซอฟต์แวร์นั้นถูกพัฒนามาในแนวทางที่ถูกต้อง

4) เน้นความยืดหยุ่น พร้อมรับการเปลี่ยนแปลงได้ตลอดเวลา เรียนรู้จากข้อผิดพลาดและผลสำเร็จ เพื่อปรับปรุงการทำงานช่วยให้โครงการสามารถดำเนินต่อไปได้อย่างราบรื่น

ซึ่งแนวคิดแบบ Agile มักจะมาคู่กับกรอบการทำงาน (Framework) แบบ Scrum ซึ่งเป็นวิธีการที่ให้ทุกคนในทีมมาช่วยกันรุมงาน โดยรอบเวลาในการทำงานแบบ Scrum จะเป็นระยะเวลาสั้นๆ จะแบ่งเป็น sprint โดยความยาวของ Sprint อยู่ตั้งแต่ระหว่าง 1 – 4 สัปดาห์ เมื่อครบกำหนด Sprint ทีมงานจะต้องเสร็จงานตามที่ได้วางแผนไว้โดยไม่จำเป็นต้องรอให้งานเสร็จทั้งหมด สามารถนำบางส่วนที่ใช้งานก่อนได้ให้ลูกค้าหรือผู้ใช้งานแล้วค่อยทยอยส่วนอื่น ๆ เพิ่มเติมในภายหลัง ลักษณะแบบนี้เรียกว่า MVP หรือ Minimum Viable Product ถือเป็นเครื่องมือสำคัญในการประเมินตลาดว่าสินค้านั้นจะรุ่งหรือจะร่วงได้ดีทีเดียว ซึ่งในแต่ละ Sprint จะมีการทำกิจกรรมของ Scrum เพื่อให้การดำเนินงานเป็นขั้นตอนชัดเจน วัดผลได้ และลดการประชุมที่ไม่จำเป็น ซึ่ง Scrum จะประกอบไปด้วยกิจกรรมต่าง ๆ ดังนี้

1) Sprint Goal คือ การระบุวัตถุประสงค์ของ Sprint เช่น การลงมือพัฒนาฟีเจอร์นี้ต้องถูกดำเนินการไปกี่ส่วนกี่เปอร์เซ็นต์ หรือต้องการให้ได้ข้อสรุปความต้องการของผู้ใช้ สิ่งสำคัญคือการทำให้ทีมเห็นเป้าหมายตรงกัน เพื่อให้รู้ว่าต้องทำงานชิ้นนี้ไปเพื่ออะไร ทีมสามารถพัฒนาชิ้นงานไปในทิศทางเดียวกัน และวัดผลได้ชัดเจน

2) Sprint Planning คือ กำหนดกิจกรรมหรืองานที่ต้องทำภายในระยะเวลาที่กำหนดสั้น ๆ ไม่เกิน 2 - 4 สัปดาห์ โดยแจกแจงงานใหญ่ให้เป็งานย่อย ๆ เพื่อให้รู้ว่าต้องทำงานอะไรบ้าง เป้าหมายก็คือการทำงานให้เสร็จภายในเวลาอันรวดเร็ว เน้นใช้งานได้จริงเป็นหลัก

3) Sprint Review คือการตรวจสอบผลลัพธ์ของ Sprint โดยจะมีการวัดผล ตรวจสอบ วิเคราะห์และรับ Feedback เพื่อนำไปปรับปรุงแก้ไขเพิ่มเติม ส่วนมากจะใช้เวลาไม่เกิน 2 ชั่วโมง สำหรับ Sprint 2 สัปดาห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีการดำเนินงานวิจัย

ในการพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์สำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มตลาดดิจิทัลเขื่อนนั้น เป็นการพัฒนาช่องทางการสื่อสารให้ผู้ประกอบการร้านค้าสามารถรับส่งข้อความและรูปภาพกับลูกค้าผู้ใช้บริการที่เข้ามาสั่งซื้ออาหารได้โดยสะดวก ซึ่งผู้จัดทำมีวิธีการดำเนินงานดังนี้

3.1 การวางแผนและการวิเคราะห์ระบบ

การพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์สำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มตลาดดิจิทัลเขื่อน มีการวางแผนและการวิเคราะห์ระบบในแง่มุมต่าง ๆ เพื่อให้เตรียมพร้อมในการพัฒนาพีเจอาร์ดังกล่าว ซึ่งมีขั้นตอนต่อไปนี้

3.1.1 วิเคราะห์ความต้องการของผู้ใช้และความต้องการของระบบ

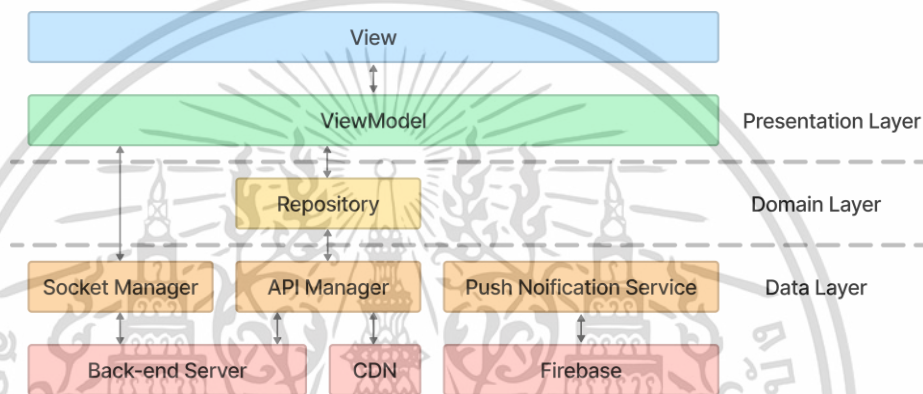
จากการศึกษาพฤติกรรมการใช้งานแอปพลิเคชันที่ใช้สำหรับการสั่งซื้ออาหารที่มีพีเจอาร์ในส่งข้อความสนทนาและข้อกำหนดต่าง ๆ ของระบบแพลตฟอร์มตลาดดิจิทัลเขื่อน สามารถสรุปเป็นความต้องการ 11 ข้อดังต่อไปนี้

- 1) ผู้ใช้สามารถรับ-ส่งข้อความได้
- 2) ผู้ใช้สามารถรับ-ส่งรูปภาพได้
- 3) ผู้ใช้สามารถพรีวิวรูปภาพได้
- 4) ผู้ใช้สามารถดูประวัติการสนทนา
- 5) เมื่อคู่สนทนาได้เปิดอ่านข้อความแล้วให้ขึ้นสถานะ "อ่านแล้ว"
- 6) เมื่อมีข้อความใหม่จะมีการแจ้งเตือนผ่าน Push Notification
- 7) ผู้ใช้สามารถกด Push Notification เพื่อไปยังหน้าแชทได้
- 8) Push Notification ถูกจัดกลุ่มไว้ตามรายการการสั่งซื้อ
- 9) เมื่อมีข้อความใหม่และผู้ใช้ยังไม่ได้ทำการเปิดอ่าน ปุ่มเข้าสู่ห้องสนทนาที่หน้ารายละเอียดคำสั่งซื้อจะแสดงจุดสีแดง
- 10) หากมีข้อความที่ส่งไม่สำเร็จ จะทำการเก็บข้อความลง local database ไว้ และเมื่ออินเทอร์เน็ตกลับมาใช้งานได้ ผู้ใช้งานสามารถกดส่งข้อความนั้นใหม่อีกครั้ง
- 11) เมื่อเวลาผ่านไป 24 ชั่วโมงหลังจากการทำรายการสั่งซื้อผู้ใช้จะไม่สามารถดู

เอกสารนี้เป็นเอกสารต้นฉบับและรับส่งข้อความได้อีก เช่นนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 วิเคราะห์ระบบงานเดิม

การพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันนั้นเป็นการพัฒนาพีเจอาร์ใหม่ที่ทำงานอยู่บนพื้นฐานของแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่มีอยู่เดิมดังที่ปรากฏอยู่ในรูปที่ 3.1 ผู้จัดทำจึงจำเป็นต้องศึกษากระบวนการทำงานและส่วนต่าง ๆ ของแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า เพื่อให้สามารถเรียกใช้ฟังก์ชันการทำงานรวมถึงข้อมูลที่เกี่ยวข้องมาใช้งาน และต่อยอดฟังก์ชันการทำงานที่มีอยู่เดิมให้พัฒนาเป็นพีเจอาร์การสนทนาที่สมบูรณ์ได้ โดยจะมีชุดคำสั่งต่าง ๆ ที่ถูกพัฒนาต่อยอดและถูกเรียกใช้เพื่อสนับสนุนการทำงานของพีเจอาร์การสนทนาดังนี้



รูปที่ 3.1 แผนภาพโครงสร้างของพื้นฐานของแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า

3.1.2.1 API Manager

API Manager เป็นคลาสที่ทำหน้าที่เกี่ยวกับการจัดการสร้างและการส่ง Request การรับ Response และข้อมูลจากฝั่ง Back-end หรือ Service อื่น ๆ ใช้ในการขอข้อมูลทั่ว ๆ ไปอื่น ๆ ที่ใช้อยู่งานอยู่เดิมในแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า อย่างเช่น การขอข้อมูลประวัติการสั่งซื้อมาแสดงผล

โดยคลาสดังกล่าวถูกทำซ้ำขึ้นมาใหม่และแก้ชุดคำสั่งให้สามารถส่ง Request และรับ Response ที่มีคุณลักษณะสอดคล้องกับข้อกำหนดที่ใช้ในการแลกเปลี่ยนข้อมูลผ่านช่องทางใหม่ที่ฝั่ง Back-end เตรียมไว้ใช้สำหรับพีเจอาร์การสนทนาโดยเฉพาะ

3.1.2.2 Socket Manager

Socket Manager เป็นคลาสที่หน้าเกี่ยวกับการจัดการส่งและการรับ Event จากฝั่ง Back-end เพื่อใช้งานในการอัปเดตสถานะของคำสั่งซื้อ เช่น หากร้านค้ากดรับคำสั่งซื้อและเริ่มปรุงอาหาร ตัวแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าจะส่ง Event เพื่ออัปเดตให้ลูกค้าผู้ใช้บริการที่เข้ามาสั่งอาหารได้ทราบว่าทางร้านได้รับคำสั่งซื้อและเริ่มปรุงอาหารแล้ว

โดยคลาสดังกล่าวถูกทำซ้ำขึ้นมาใหม่และแก้ไขชุดคำสั่งให้กำหนดชื่อของ Event และที่อยู่ Socket Address ให้สอดคล้องกับที่ฝั่ง Back-end กำหนดขึ้นมาใหม่สำหรับไว้ใช้รับส่งข้อความและรูปภาพสำหรับฟีเจอร์การสนทนาโดยเฉพาะ

3.1.2.3 Upload ViewModel

เป็นคลาส ViewModel ที่มีหน้าหลักคือการอัปโหลดรูปขึ้นไปยัง CDN Service (Content Delivery Network Service) เพื่อให้แอปพลิเคชันในเครือข่ายสามารถแสดงผลภาพจาก URL ผ่านอินเทอร์เน็ตได้ ซึ่งแต่เดิมใช้ในการตั้งภาพโลโก้ร้านค้า และการตั้งรูปภาพให้แก่เมนูอาหาร

โดยคลาส Upload ViewModel นั้นไม่ได้ถูกทำซ้ำหรือแก้ไขเพิ่มเติม แต่ถูกเรียกใช้ในการอัปโหลดรูปขึ้นไปยังเครือข่ายก่อนจะทำการส่ง URL ของรูปภาพดังกล่าวไปแสดงผลในห้องสนทนาผ่าน Socket

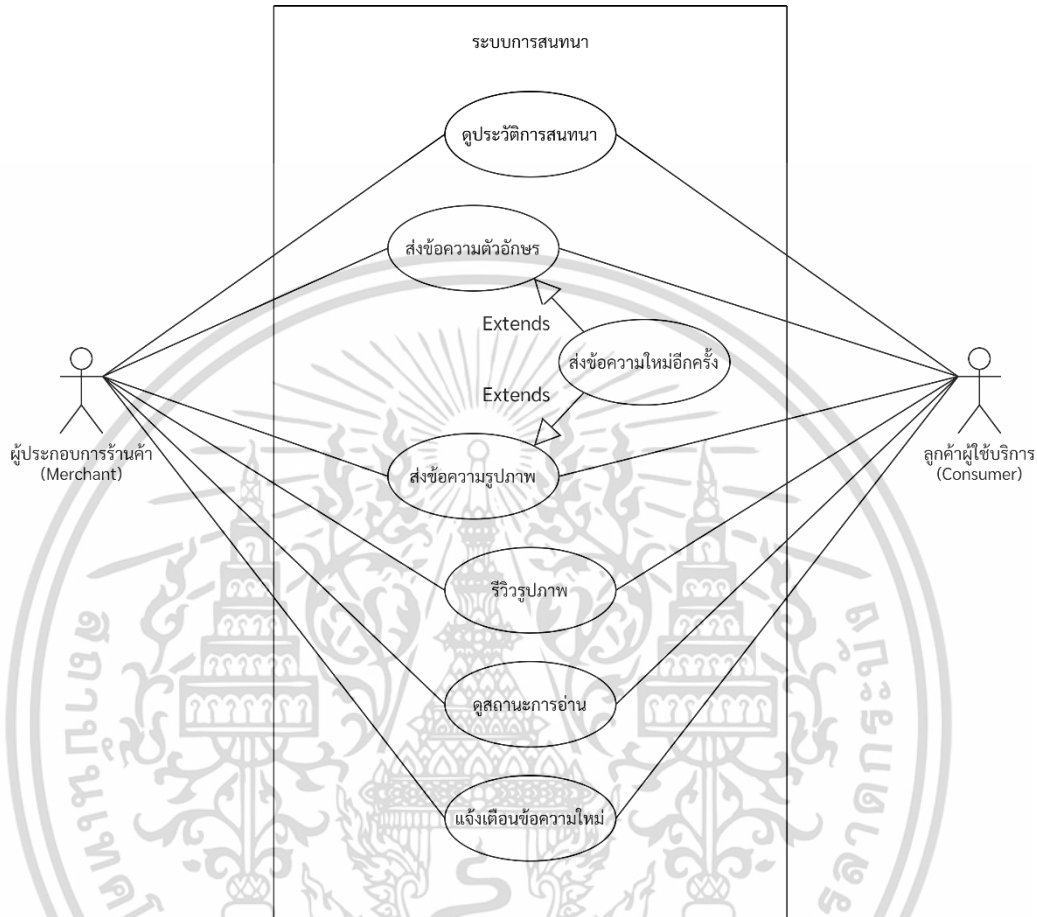
3.1.2.4 Push Notification Service

เป็นคลาสที่ใช้รองรับข้อมูลการแจ้งเตือนจาก Firebase Cloud Message (FCM) และทำการแจ้งเตือนทั้งเสียงแจ้งเตือนและแสดงข้อมูลบนหน้าจอ ซึ่งแต่เดิมใช้ในการแจ้งเตือนสถานะต่าง ๆ ของคำสั่งซื้อ เช่น แจ้งเตือนให้ผู้ประกอบการร้านค้าทราบว่าคำสั่งซื้อถูกยกเลิกโดยลูกค้า

โดยคลาสดังกล่าวถูกแก้ไขเพิ่มเติมชุดคำสั่งให้สามารถรับข้อมูลการแจ้งเตือนข้อความใหม่และสามารถจัดกลุ่มข้อความแจ้งเตือนข้อความใหม่ตามคำสั่งซื้อ พร้อมทั้งแสดงผลข้อความการแจ้งเตือนที่จะประกอบไปด้วยชื่อของลูกค้าผู้ใช้บริการและข้อความใหม่ที่ลูกค้าผู้ใช้บริการส่งมา

3.2 การการออกแบบระบบ

3.2.1 แผนภาพการแสดงผลการทำงานของผู้ใช้ระบบ (Use Case Diagram)



รูปที่ 3.2 แผนภาพการแสดงผลการทำงานของผู้ใช้ระบบการสนทนา

จากการทำงานของระบบที่ปรากฏในรูปที่ 3.2 นั้น ระบบการสนทนาจะเริ่มขึ้นเมื่อผู้ประกอบการร้านค้าได้รับคำสั่งซื้อจากลูกค้าผู้ใช้บริการ ผู้ประกอบการร้านค้าสามารถเข้าไปยังห้องสนทนาได้ โดยแต่ละคำสั่งซื้อจะมีห้องสนทนาของเป็นตัวเอง ซึ่งเมื่อเข้าไปในห้องสนทนาแล้วร้านค้าผู้ใช้บริการจะเห็นประวัติการสนทนาระหว่างผู้ประกอบการร้านค้าและลูกค้าผู้ใช้บริการได้ จากนั้นผู้ประกอบการร้านค้าจะสามารถส่งข้อความได้ทั้งข้อความตัวอักษรและข้อความรูปภาพ

เมื่อข้อความถูกส่งสำเร็จ ลูกค้าผู้ใช้บริการจะได้รับข้อความนั้น ๆ หากข้อความดังกล่าวเป็นข้อความรูปภาพ ข้อความดังกล่าวจะปรากฏเป็นรูปภาพบนห้องสนทนาซึ่งทั้งผู้ประกอบการร้านค้าและลูกค้าผู้ใช้บริการสามารถกดที่รูปภาพเพื่อดูรูปแบบเต็มจอได้ ในไม่ช้ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ผู้ประกอบการร้านค้าทำการส่งข้อความสัญญาณอินเทอร์เน็ตขัดข้อง ทำให้ส่งข้อความไม่สำเร็จ ข้อความดังกล่าวจะถูกเก็บลงในฐานข้อมูลและเมื่อสัญญาณอินเทอร์เน็ตกลับมาใช้งานได้ ผู้ประกอบการร้านค้าสามารถกดส่งข้อความได้ใหม่อีกครั้ง

ผู้ประกอบการร้านค้ายังสามารถทราบได้ว่าข้อความที่ส่งไปนั้นถูกอ่านโดยลูกค้า ผู้ใช้บริการหรือยัง โดยเมื่อลูกค้าผู้ให้บริการได้ทำการเปิดอ่านข้อความจากร้านค้า ผู้ประกอบการแล้ว ข้อความดังกล่าวจะปรากฏอักษรบ่งบอกสถานะว่าข้อความนั้นถูกเปิดอ่านแล้ว

นอกจากนี้ เมื่อผู้ประกอบการร้านค้าได้รับข้อความใหม่จากลูกค้าผู้ให้บริการ ร้านค้า จะได้รับการแจ้งเตือนแบบ Push Notification เพื่อแจ้งให้ผู้ประกอบการร้านค้าทราบว่า มีข้อความใหม่ลูกค้าผู้ให้บริการส่งเข้ามา

ตารางที่ 3.1 คำอธิบาย Use Case Description ของส่งข้อความตัวอักษร

Name	ส่งข้อความตัวอักษร
Description	ผู้ประกอบการร้านค้าสามารถส่งข้อความที่เป็นตัวอักษร และอีโมจิ (Emoji) ไปยังลูกค้าผู้ให้บริการได้
Input	ข้อความตัวอักษร และอีโมจิ (Emoji)
Output	เวลาที่ข้อความถูกส่งได้สำเร็จ
Condition	ข้อความตัวอักษรที่ลูกค้าผู้ให้บริการทำการส่งไปต้องไม่เกิน 128 ตัวอักษร และจะต้องมีการเชื่อมต่ออินเทอร์เน็ต

ตารางที่ 3.2 คำอธิบาย Use Case Description ของส่งข้อความรูปภาพ

Name	ส่งข้อความรูปภาพ
Description	ผู้ประกอบการร้านค้าสามารถส่งข้อความที่เป็นรูปภาพไปยังลูกค้าผู้ให้บริการได้
Input	รูปภาพ
Output	เวลาที่ลูกค้าผู้ให้บริการทำการส่งข้อความ
Condition	ส่งรูปภาพได้ครั้งละหนึ่งรูปภาพและจะต้องมีการเชื่อมต่ออินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 คำอธิบาย Use Case Description ของพรีวิรูรูปภาพ

Name	พรีวิรูรูปภาพ
Description	ผู้ประกอบการร้านค้าสามารถกดที่ข้อความรูปภาพเพื่อดูรูปภาพแบบเต็มจอได้
Input	
Output	รูปภาพขนาดขยายเต็มจอ

ตารางที่ 3.4 คำอธิบาย Use Case Description ของดูประวัติการสนทนา

Name	ดูประวัติการสนทนา
Description	ผู้ประกอบการร้านค้าสามารถดูประวัติการสนทนายระหว่างผู้ประกอบการร้านค้ากับผู้ใช้บริการ โดยประวัติการสนทนาจะเป็นประวัติการสนทนาของห้องแชทนั้นๆ
Input	
Output	ประวัติการสนทนา
Condition	ผู้ประกอบการร้านค้าจะสามารถดูประวัติการสนทนาได้เฉพาะเวลาที่ทำการสั่งซื้อต้องไม่เกิน 24 ชั่วโมง

ตารางที่ 3.5 คำอธิบาย Use Case Description ของเช็คสถานะการอ่าน

Name	เช็คสถานะการอ่าน
Description	ผู้ประกอบการร้านค้าสามารถตรวจสอบสถานะการอ่านของข้อความที่ได้ทำการส่งไปยังลูกค้าผู้ใช้บริการได้ โดยหากลูกค้าผู้ใช้บริการได้ทำการเข้าสู่ห้องสนทนาและเปิดอ่านข้อความแล้ว ข้อความที่ถูกส่งจากผู้ประกอบการร้านค้าจะมีสถานะขึ้นให้รู้ว่าข้อความดังกล่าวถูกอ่านโดยลูกค้าผู้ใช้บริการแล้ว
Input	
Output	สถานะการอ่านของข้อความ
Condition	กรณีที่ข้อความล่าสุดขึ้นสถานะว่าอ่านแล้ว ข้อความก่อนหน้าทั้งหมดจะต้องมีสถานะว่าอ่านแล้วเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 คำอธิบาย Use Case Description ของแจ้งเตือนข้อความ

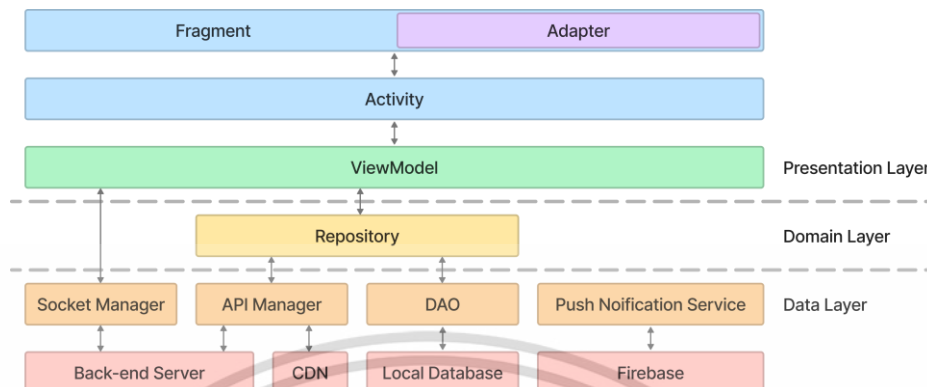
Name	แจ้งเตือนข้อความ
Description	ร้านค้าผู้ประกอบการทำการส่งข้อความไปยังร้านค้าสำเร็จ ทางร้านค้าจะได้รับการแจ้งเตือนข้อความ (notification) และถ้าร้านค้าส่งข้อความมายังลูกค้าผู้ใช้บริการ ทางฝั่งลูกค้าผู้ใช้บริการก็จะได้รับการแจ้งเตือนข้อความ (notification) โดยลูกค้าผู้ใช้บริการสามารถกด notification เพื่อไปยังห้องแชทนั้นได้
Input	
Output	แสดงผลแจ้งเตือนข้อความว่าได้รับข้อความใหม่จากทางฝั่งร้านค้า

ตารางที่ 3.7 คำอธิบาย Use Case Description ของส่งข้อความใหม่อีกครั้ง

Name	ส่งข้อความใหม่อีกครั้ง
Description	ลูกค้าผู้ใช้บริการทำการส่งข้อความไปยังร้านค้าสำเร็จ ทางร้านค้าจะได้รับการแจ้งเตือนข้อความ (notification) และถ้าร้านค้าส่งข้อความมายังลูกค้าผู้ใช้บริการ ทางฝั่งลูกค้าผู้ใช้บริการก็จะได้รับการแจ้งเตือนข้อความ (notification) โดยลูกค้าผู้ใช้บริการสามารถกด notification เพื่อไปยังห้องแชทนั้นได้
Input	
Output	แสดงผลแจ้งเตือนข้อความว่าได้รับข้อความใหม่จากทางฝั่งร้านค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 แผนภาพที่แสดงชั้นการทำงานและส่วนต่าง ๆ ของแอปพลิเคชัน



รูปที่ 3.3 แผนภาพโครงสร้างของพีเจอาร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า

แผนภาพที่ปรากฏในรูปที่ 3.3 นั้นแสดงถึงโครงสร้างของแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิดเงินตามแนวคิดของ Architecture Components โดยอธิบายความสัมพันธ์ชั้นการทำงานและส่วนต่าง ๆ ของแต่ละชั้นที่ทำงานสอดประสานกัน โดยในแต่ละชั้นจะมีองค์ประกอบดังต่อไปนี้

3.2.2.1 Data Layer

ชั้นของส่วนที่เป็น Data Source หรือแหล่งของข้อมูล

3.2.2.1.1 Back-end Server

เป็นระบบหลังบ้านของแพลตฟอร์มคลาวด์คิดเงิน มีหน้าประประมวลผลการทำงานต่าง ๆ รวมถึงเป็นแหล่งของข้อมูลสำคัญที่จะรับส่งข้อมูลต่าง ๆ ที่จำเป็นให้แก่แอปพลิเคชันในแพลตฟอร์ม ทั้งยังเป็นตัวกลางที่ใช้ในการรับส่งข้อความของพีเจอาร์การสนทนาอีกด้วย

3.2.2.1.2 CDN Service (Content Delivery Network Service)

เป็นบริการที่ใช้ฝากรูปภาพ โดยสามารถอัปโหลดรูปภาพและเพื่อให้แอปพลิเคชันของแพลตฟอร์มคลาวด์คิดเงิน สามารถแสดงผลรูปภาพจาก URL ผ่านอินเทอร์เน็ตได้ เป็นแหล่งข้อมูลที่ใช้ในการแสดงผลข้อความที่เป็นรูปภาพ

3.2.2.1.3 Socket Manager

เป็นคลาส Kotlin ที่ทำหน้าที่ในการรับและส่ง Event พร้อมด้วยข้อมูลเพื่อใช้ในการติดต่อกับ Back-end Server มีบทบาทสำคัญในการรับส่งข้อความและการทำงานอื่น ๆ ของพีเจอาร์การสนทนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1.4 API Manager

เป็นคลาส Kotlin ที่ทำหน้าที่ในการส่ง Request และรับ Response เพื่อขอข้อมูลที่ต้องการจาก Back-end และบริการ Service อื่นๆ มีบทบาทสำคัญในการขอข้อมูลประวัติการสนทนาของพีเจอาร์การสนทนา

3.2.2.1.5 Local Database

เป็นฐานข้อมูลที่เก็บข้อมูลประเภท SQLite โดยฐานข้อมูลดังกล่าวเป็นแหล่งเก็บข้อมูลภายในของอุปกรณ์ ใช้เก็บข้อมูลข้อความที่ถูกส่งไม่สำเร็จในกรณีที่สัญญาณอินเทอร์เน็ตขัดข้อง เพื่อให้ผู้ใช้สามารถส่งข้อความดังกล่าวอีกครั้งเมื่อสัญญาณอินเทอร์เน็ตกลับมาใช้งานได้อีกครั้ง

3.2.2.1.6 DAO (Data Access Object)

เป็นคลาส Kotlin ที่ทำหน้าที่ในการกำหนดคำสั่งจัดการข้อมูลภายในฐานข้อมูล Local Database เป็นคลาสที่มีฟังก์ชันที่บรรจุคำสั่ง SQL ต่าง ๆ ไว้ให้คลาสอื่น ๆ เรียกใช้งานเพื่อจัดการกับข้อมูลต่าง ๆ ในฐานข้อมูล

3.2.2.1.7 Firebase

เป็นแพลตฟอร์มที่ให้บริการเกี่ยวกับระบบหลังบ้าน (Back-end as a Service) ที่จะทำการส่งข้อความแจ้งเตือนแบบข้ามแพลตฟอร์ม (Cross platform messaging) กล่าวคือเมื่อลูกค้าผู้ใช้บริการทำการส่งข้อความมายังร้านค้าผู้ประกอบการ Firebase จะทำการส่งข้อมูลของการแจ้งเตือนมายังแอปพลิเคชันสำหรับร้านค้าผู้ประกอบการเพื่อทำการแจ้งเตือนต่อไป

3.2.2.1.8 Push Notification Service

เป็นคลาสที่ใช้รองรับข้อมูลการแจ้งเตือนจาก Firebase และทำการแจ้งเตือนทั้งเสียงแจ้งเตือนและแสดงข้อมูลบนหน้าจอ ใช้ในการแจ้งเตือนสถานะต่าง ๆ ของคำสั่งซื้อ รวมถึงรับข้อมูลการแจ้งเตือนข้อความใหม่และสามารถจัดกลุ่มข้อการแจ้งเตือนข้อความใหม่ตามคำสั่งซื้อ พร้อมทั้งแสดงผลข้อความการแจ้งเตือนที่จะประกอบไปด้วยชื่อของลูกค้าผู้ใช้บริการและข้อความใหม่ที่ลูกค้าผู้ใช้บริการส่งมา

3.2.2.2 Domain Layer

ชั้นของส่วนที่ทำหน้าที่เกี่ยวกับตรรกะการจัดการข้อมูล

3.2.2.2.1 Repository

เป็นคลาส Kotlin ที่ทำหน้าที่เป็นศูนย์รวมข้อมูลต่าง ๆ จากแหล่งข้อมูลทั้ง API Manager และ Local Database โดยในแง่ของการแลกเปลี่ยนข้อมูลผ่าน API Manager นั้น Repository จะทำแปลงข้อมูลที่ได้จาก Response เป็นข้อมูลที่สามารถใช้งานได้ ก่อนจะส่งไปยัง ViewModel ต่อไป ส่วนในแง่ของ Local

Database เมื่อ ViewModel ต้องการเก็บข้อมูลใด ๆ ก็สามารถส่งต่อให้ Repository จัดการต่อด้วยการแปลงข้อมูลดังกล่าวเป็นสามารถบรรจุลงในฐานข้อมูล SQL ได้ก่อนทำการบันทึกข้อมูลดังกล่าวต่อไป

3.2.2.3 Presentation Layer

ชั้นของส่วนที่ทำหน้าที่เกี่ยวกับตรรกะการตอบโต้กับผู้ใช้และ Business Logic

3.2.2.3.1 ViewModel

เป็นตัวกลางระหว่างส่วนของข้อมูลและส่วนต่อประสาน ทำหน้าที่รับส่งข้อมูลระหว่างส่วน มีบทบาทในการนำข้อความอักษร อีโมจิ และรูปภาพ ที่ได้รับจากผู้ใช้ผ่านส่วน View สร้างเป็นข้อมูลลงข้อความที่สามารถแสดงผลเป็นกล่องข้อความในห้องสนทนาได้ ก่อนจะส่งไปยัง Socket Manager เพื่อทำการส่งข้อความดังกล่าวต่อไป นอกจากนี้ยังหน้าที่จัดตรรกะอื่น ๆ เช่น ในกรณีเมื่อเวลาผ่านไป 24 ชั่วโมงหลังจากการทำรายการสั่งซื้อผู้ใช้จะไม่สามารถดูประวัติการสนทนาและรับส่งข้อความได้อีก จึงต้องเปรียบเทียบเวลาที่คำสั่งซื้อถูกสร้างกับเวลาปัจจุบันเพื่อให้สามารถระบุได้ว่าผู้ใช้งานกำลังเข้าสู่ห้องสนทนาของคำสั่งซื้อที่เกิน 24 ชั่วโมงไปแล้วหรือไม่

3.2.2.3.2 Activity

เป็นส่วนหลักของ View ซึ่งจะเป็นตัวกำหนดวงจรชีวิตของห้องสนทนา หากถูกเปิดอยู่หน้า Activity ก็หมายความว่าผู้ใช้งานกำลังเปิดห้องสนทนาอยู่ Socket ก็จะทำหน้าที่เชื่อมต่อเพื่อทำการรับส่งข้อความ ในทางกลับกันหากหน้า Activity ถูกปิด Socket จะทำการปิดการเชื่อมต่อเพื่อหยุดรับส่งข้อความ

3.2.2.3.3 Fragment

เป็นส่วนของ View ที่ทำงานอยู่บน Activity อีกชั้นหนึ่ง ทำหน้าที่แสดงผลหน้าจอห้องสนทนา ตอบโต้กับผู้ใช้เมื่อกดปุ่มต่างๆ รวมถึงการรับข้อมูลจากผู้ใช้ ทั้งข้อความอักษร อีโมจิ และรูปภาพ เพื่อส่งข้อมูลดังกล่าวไปเป็นข้อความ

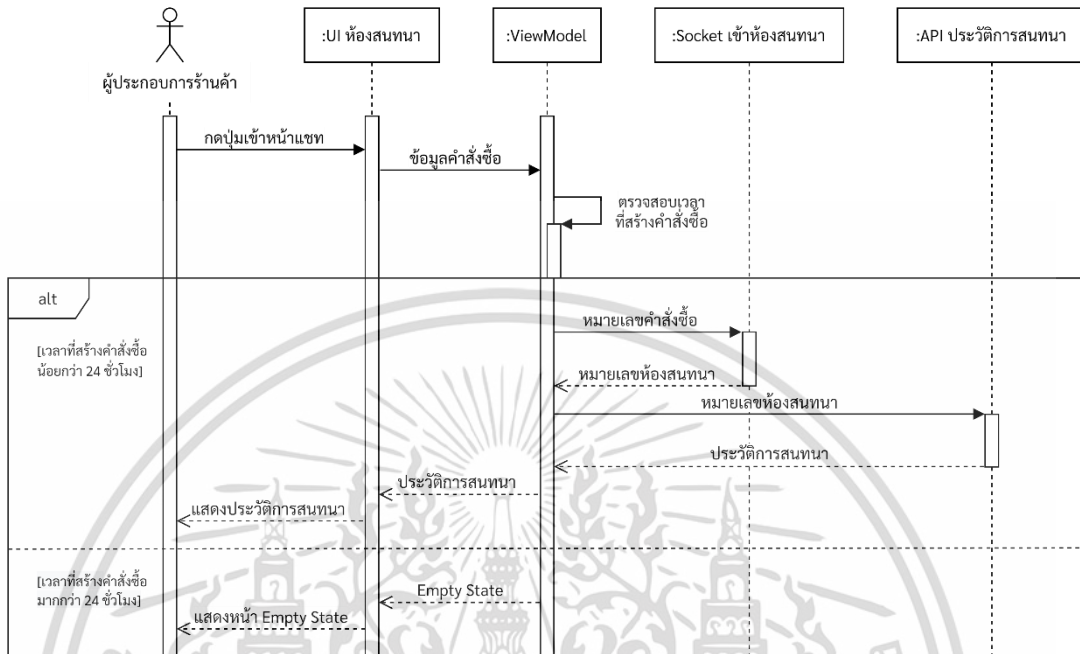
3.2.2.3.4 Adapter

เป็นคลาสที่มีหน้าที่รับประวัติการสนทนามาทำการแสดงผลเป็นรายการของกล่องข้อความที่เรียงต่อ ๆ กันจากบนลงล่างตามเวลา ซึ่ง Adapter จะเป็นผู้กำหนดว่าข้อความใดควรแสดงผลแบบใด หากเป็นข้อความที่ถูกส่งจากเรา กล่องข้อความที่ปรากฏบนหน้าจอจะต้องอยู่ชิดทางขวา ในกลับกันหากเป็นข้อความที่ถูกส่งจากคู่สนทนา กล่องข้อความที่ปรากฏบนหน้าจอจะต้องอยู่ชิดทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 แผนภาพที่แสดงถึงการปฏิสัมพันธ์ (Sequence Diagram)

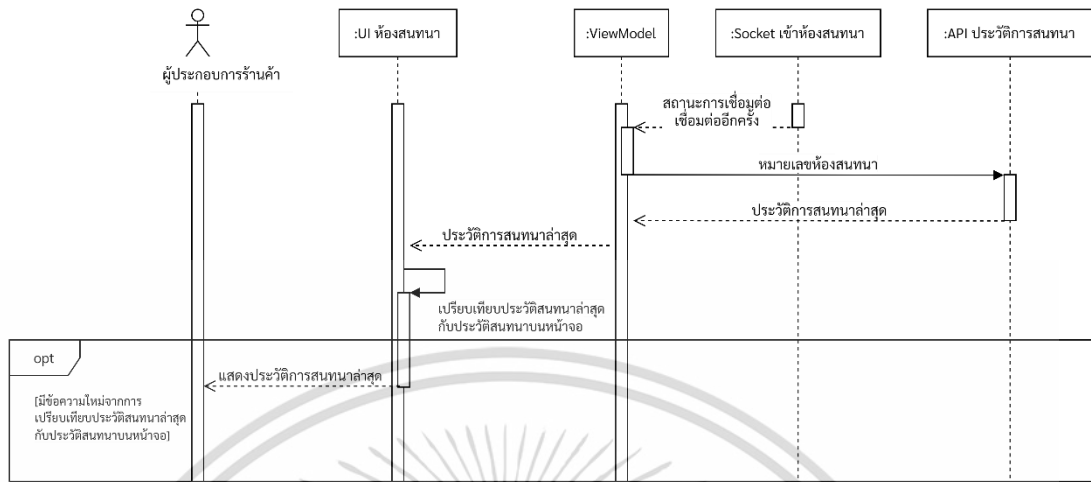
1) Sequence Diagram ของการดูประวัติการสนทนา



รูปที่ 3.4 แผนภาพ Sequence Diagram ของการดูประวัติการสนทนา

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.4 การทำงานของการดูประวัติการสนทนาจะเริ่มต้นจากผู้ประกอบการร้านค้า (Merchant) กดปุ่มเข้าสู่ห้องสนทนาผ่านทางหน้ารายละเอียดคำสั่งซื้อ เมื่อเข้าไปยังหน้าห้องแชทแล้วจะมีการส่งข้อมูลของคำสั่งซื้อไปยัง ViewModel เพื่อทำการตรวจสอบวันเวลาที่ทำการสั่งซื้อ หากวันเวลาที่ทำการสั่งซื้อยังมีระยะเวลาไม่เกิน 24 ชั่วโมง ViewModel จะทำการส่งหมายเลขคำสั่งซื้อดังกล่าวไปยังระบบหลังบ้านผ่าน Socket เพื่อขอหมายเลขห้องสนทนากลับมาเพื่อใช้ในการขอประวัติการสนทนาจาก API อีกต่อหนึ่ง และเมื่อ API ได้ทำการข้อมูลส่งประวัติกลับมา ประวัติการสนทนาดังกล่าวจะถูกนำมาแสดงผลผ่านห้องสนทนา แต่ถ้าวันเวลาที่ทำการสั่งซื้อมีระยะเวลาเกิน 24 ชั่วโมง ViewModel คืนค่าเพื่อห้องสนทนาแสดงหน้า Empty state ซึ่งผู้ประกอบการร้านค้า (Merchant) จะไม่สามารถดูประวัติการสนทนาและส่งข้อความได้

2) Sequence Diagram ของการโหลดประวัติการสนทนาใหม่อีกครั้ง

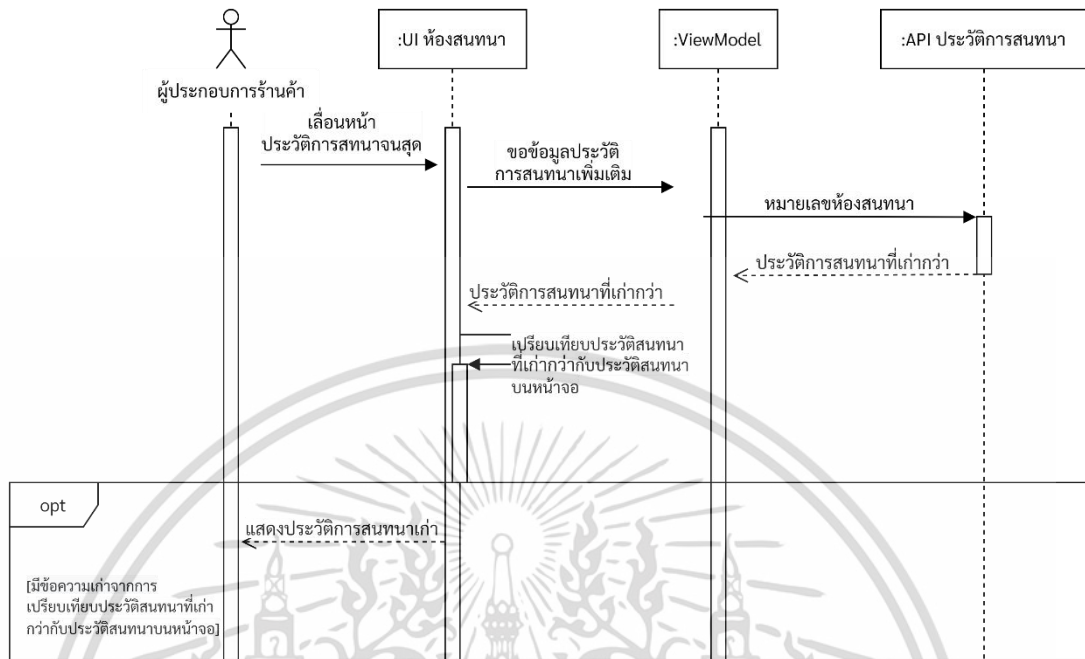


รูปที่ 3.5 แผนภาพ Sequence Diagram ของการโหลดประวัติการสนทนาใหม่อีกครั้ง

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.5 การทำงานของการโหลดประวัติการสนทนาใหม่อีกครั้งจะเริ่มต้นหากสัญญาณอินเทอร์เน็ตเกิดขัดข้องในขณะที่ผู้ประกอบการร้านค้าอยู่ภายในห้องสนทนา เมื่อสัญญาณอินเทอร์เน็ตเน็ตกลับมาใช้งานได้ และ Socket ได้ทำการเชื่อมต่อกับระบบหลังบ้าน Socket คืบค่าสถานการณ์เชื่อมต่อให้แก่ ViewModel ทราบว่าการเชื่อมต่อกลับมาแล้ว ViewModel ก็จะทำ การขอประวัติการสนทนาอีกจากระบบหลังอีกครั้งผ่าน API ก่อนจะส่งต่อประวัติการสนทนาที่ถูกขอใหม่ไปยังส่วน View เพื่อทำการเปรียบเทียบประวัติการสนทนาที่มีอยู่บนหน้าจอกับประวัติการสนทนาที่เพิ่งถูกขอใหม่ หากพบข้อความใหม่จากการเปรียบเทียบประวัติการสนทนาทั้งสอง ส่วน View จะนำข้อความใหม่ดังกล่าวมาแสดงผลบนหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Sequence Diagram ของการโหลดประวัติการสนทนาเพิ่ม

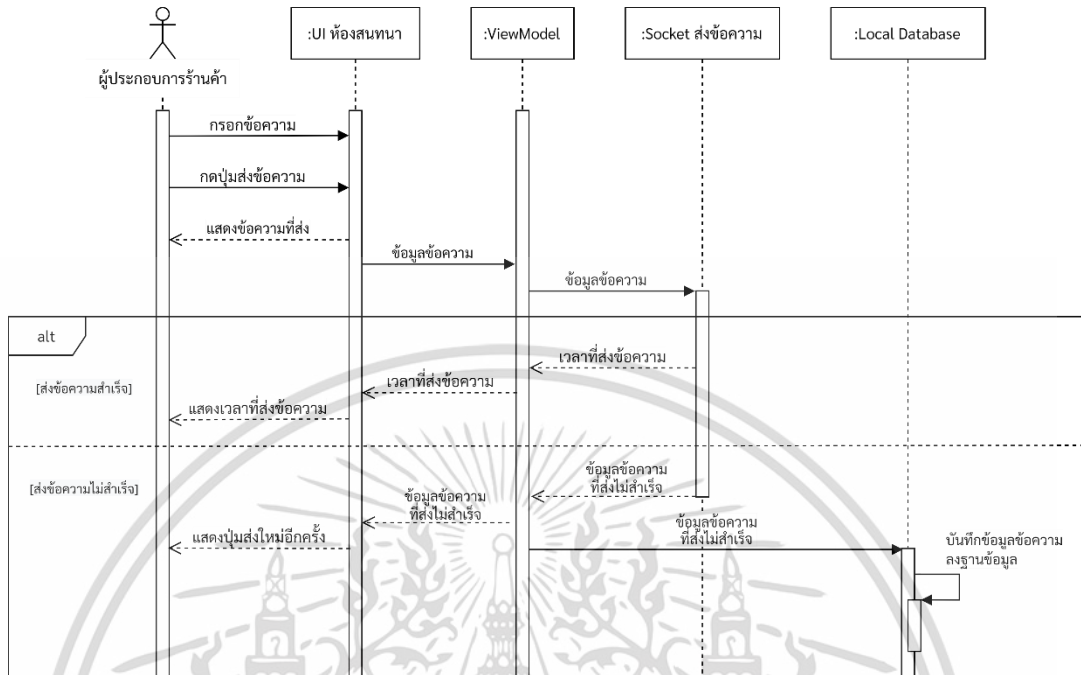


รูปที่ 3.6 แผนภาพ Sequence Diagram ของการโหลดประวัติการสนทนาเพิ่ม

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.6 การทำงานของการโหลดประวัติการสนทนาเพิ่มจะเริ่มต้นเมื่อผู้ประกอบการร้านค้าเข้าสู่ห้องสนทนาจะปรากฏเป็นข้อความเรียงต่อกันจากบนลงล่างตามเวลา ซึ่งการโหลดประวัติการสนทนาแสดงผลในครั้งแรกนั้นจะปรากฏข้อความล่าสุดไม่เกิน 20 ข้อความ และหากประวัติการสนทนาของห้องสนทนานั้นมีมากกว่า 20 ข้อความ ข้อความอื่น ๆ ที่เกินกว่าจะ 20 ข้อความแรกจะถูกโหลดมาแสดงผลเมื่อผู้ประกอบการร้านค้าเลื่อนไปยังส่วนบนสุดของ 20 ข้อความแรก โดย ViewModel ก็จะมีการขอประวัติการสนทนาเพิ่มจากระบบหลังอีกครั้งผ่าน API ครั้งละ 20 ข้อความ ก่อนจะส่งต่อประวัติการสนทนาเพิ่มไปยังส่วน View เพื่อทำการเปรียบเทียบประวัติการสนทนาที่มีอยู่บนหน้าจอกับประวัติการสนทนาเพิ่ม หากพบข้อความเก่ากว่า 20 ข้อความล่าสุดจากการเปรียบเทียบประวัติการสนทนาทั้งสอง ส่วน View จะนำข้อความเก่าดังกล่าวมาแสดงผลบนต่อด้านบนของ 20 ข้อความแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) Sequence Diagram ของการส่งข้อความอักษร

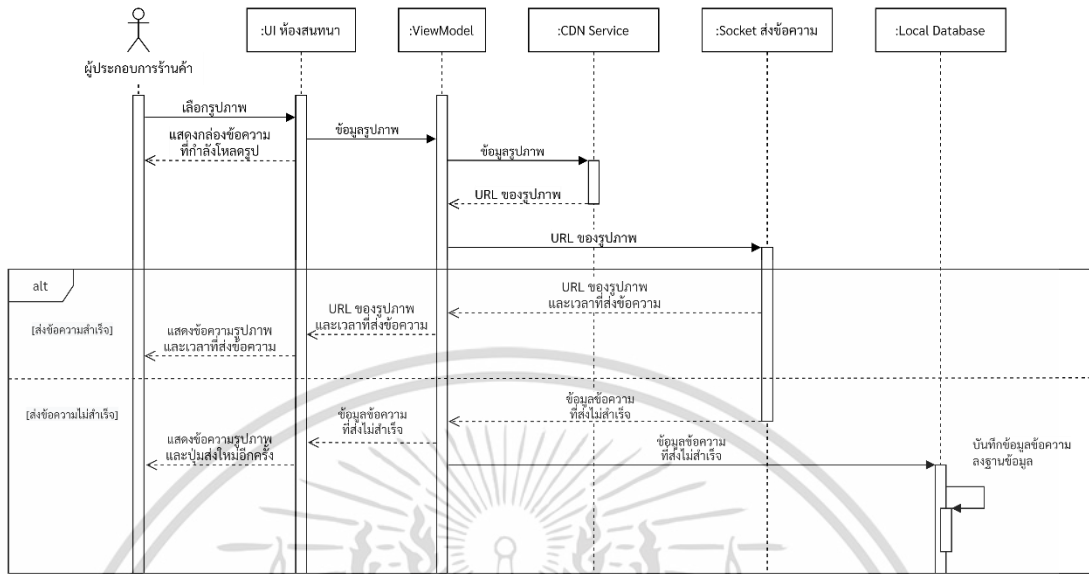


รูปที่ 3.7 แผนภาพ Sequence Diagram ของการส่งข้อความอักษร

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.7 การทำงานของการไหลตประวัติการสนทนาเพิ่มจะเริ่มต้นเมื่อผู้ประกอบการร้านค้า อยู่ในห้องสนทนาแล้ว ผู้ประกอบการร้านค้าสามารถกรอกข้อความและกดปุ่มเพื่อส่งข้อความได้ ห้องสนทนาจะทำการแสดงข้อความและส่งต่อข้อความดังกล่าวไปยัง ViewModel จากนั้น ViewModel จะทำการส่งข้อมูลข้อความดังกล่าวลูกค้าผู้ใช้บริการผ่านทาง Socket และเมื่อข้อความถูกส่งได้สำเร็จ Socket จะคืนค่าเวลาที่ข้อความถูกส่งได้สำเร็จกลับมาเพื่อให้แสดงผลแก่ผู้ประกอบการร้านค้าว่าข้อความดังกล่าวถูกส่งสำเร็จแล้ว แต่ถ้าหากในขณะที่ส่งข้อความนั้นสัญญาณอินเทอร์เน็ตเกิดขัดข้องและจนทำให้ส่งข้อความไม่สำเร็จ Socket จะคืนค่ากลับมาให้ทราบว่า การส่งข้อความล้มเหลว ทางห้องสนทนาจะแสดงผลปุ่มส่งข้อความใหม่อีกครั้งเพื่อผู้ประกอบการร้านค้าทราบว่าข้อความดังกล่าวถูกส่งไม่สำเร็จ และหลังจากนั้น ViewModel จะส่งข้อความดังกล่าวไปบันทึกลงใน Local Database เพื่อใช้ในการส่งข้อความใหม่อีกครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Sequence Diagram ของการส่งข้อความรูปภาพ

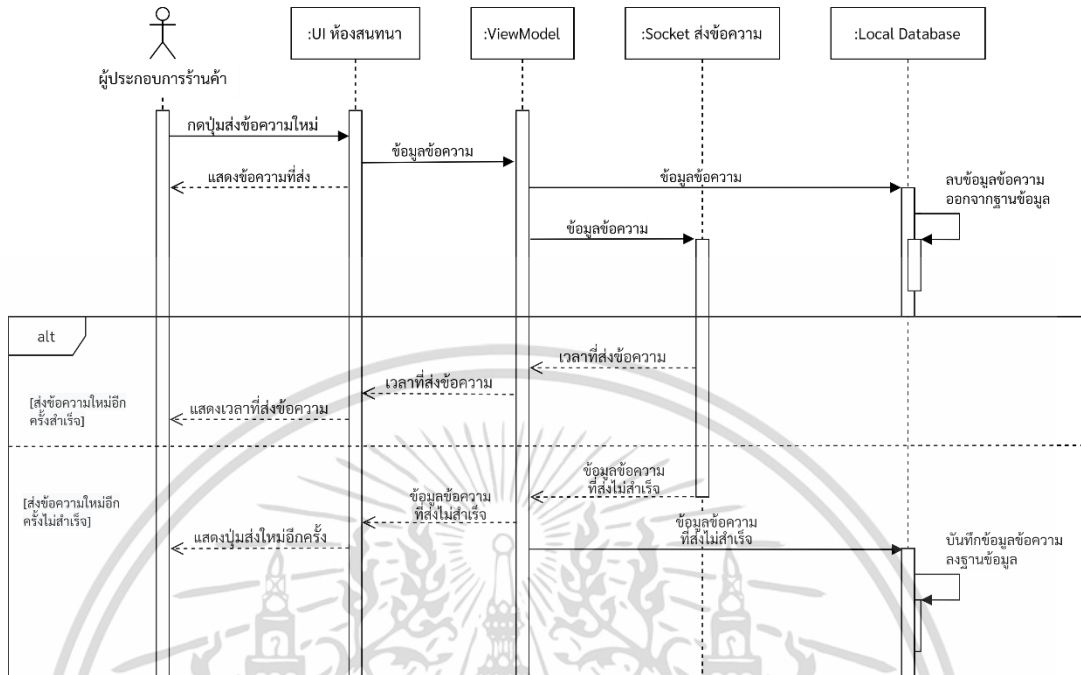


รูปที่ 3.8 แผนภาพ Sequence Diagram ของการส่งข้อความรูปภาพ

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.8 การทำงานของการส่งข้อความรูปภาพจะเริ่มขึ้นเมื่อผู้ประกอบการร้านค้าอยู่ในห้องสนทนาแล้วผู้ประกอบการร้านค้าสามารถกดปุ่มส่งรูปภาพเพื่อเลือกรูปภาพจากในอุปกรณ์และทำการส่งรูปภาพ ห้องสนทนาจะทำการแสดงกล่องข้อความที่กำลังโหลดและส่งต่อรูปภาพไปยัง ViewModel ซึ่ง ViewModel จะทำการอัปเดตรูปภาพดังกล่าวขึ้นไปฝากไว้ที่ CDN Service โดยเมื่อทำการอัปเดตสำเร็จ CDN Service จะคืนค่า URL ที่ใช้แสดงผลภาพดังกล่าว จากนั้น ViewModel จะทำการส่งข้อมูลข้อความรูปภาพพร้อมด้วย URL ให้ลูกค้าผู้ใช้บริการผ่านทาง Socket และเมื่อข้อความรูปภาพถูกส่งได้สำเร็จ Socket จะคืนค่าเวลาที่ข้อความรูปภาพถูกส่งได้สำเร็จกลับมาเพื่อให้แสดงผลแก่ผู้ประกอบการร้านค้าว่าข้อความรูปภาพดังกล่าวถูกส่งสำเร็จแล้ว แต่ถ้าหากในขณะที่ส่งข้อความรูปภาพนั้นสัญญาณอินเทอร์เน็ตเกิดขัดข้องและจนทำให้ส่งข้อความรูปภาพไม่สำเร็จ Socket จะคืนค่ากลับมาให้ทราบว่าการส่งข้อความรูปภาพล้มเหลว ทางห้องสนทนาจะแสดงผลปุ่มส่งข้อความใหม่อีกครั้งเพื่อผู้ประกอบการร้านค้าทราบว่าข้อความรูปภาพดังกล่าวถูกส่งไม่สำเร็จ และหลังจากนั้น ViewModel จะส่งข้อความดังกล่าวไปบันทึกลงใน Local Database เพื่อใช้ในการส่งข้อความใหม่อีกครั้งต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) Sequence Diagram ของการส่งข้อความอักษรใหม่อีกครั้ง

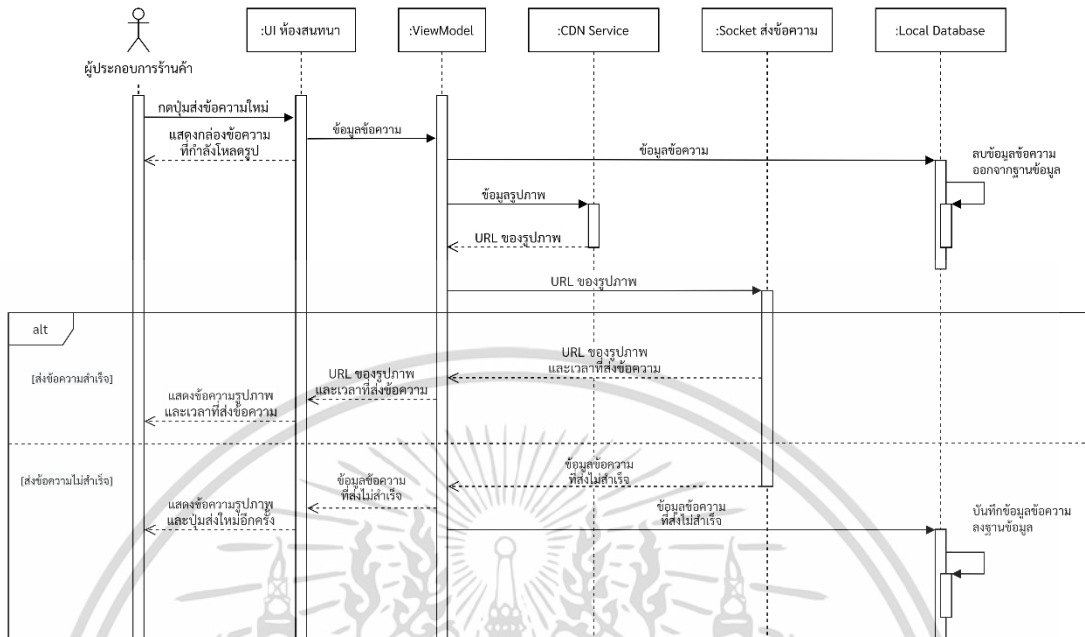


รูปที่ 3.9 แผนภาพ Sequence Diagram ของการส่งข้อความอักษรใหม่อีกครั้ง

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.9 การทำงานของการส่งข้อความอักษรใหม่อีกครั้งจะเริ่มต้นเมื่อมีข้อความที่ถูกส่งไม่สำเร็จ ข้อความดังกล่าวจะปรากฏในห้องสนทนาพร้อมกับปุ่มส่งข้อความใหม่อีกครั้ง เมื่อผู้ประกอบการร้านค้าทำการกดปุ่มส่งข้อความใหม่อีกครั้ง ViewModel จะทำการลบข้อมูลของข้อความดังกล่าวออกจาก Local Database และทำการส่งข้อความใหม่อีกครั้งผ่าน Socket เมื่อข้อความที่ส่งใหม่อีกครั้งถูกส่งไปได้สำเร็จ Socket จะคืนค่าเวลาที่ข้อความถูกส่งได้สำเร็จกลับมาเพื่อให้แสดงผลแก่ผู้ประกอบการร้านค้าว่าข้อความดังกล่าวถูกส่งสำเร็จ แต่ถ้าหากการส่งข้อความยังคงล้มเหลวอีก Socket จะคืนค่ากลับมาให้ทราบว่าการส่งข้อความล้มเหลว ปุ่มส่งข้อความใหม่อีกครั้งด้านข้างข้อความจะกลับมาปรากฏขึ้นใหม่อีกครั้ง และหลังจากนั้น ViewModel จะส่งข้อมูลของข้อความที่ส่งไม่สำเร็จดังกล่าวไปบันทึกลงใน Local Database เพื่อใช้ในการส่งข้อความใหม่อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

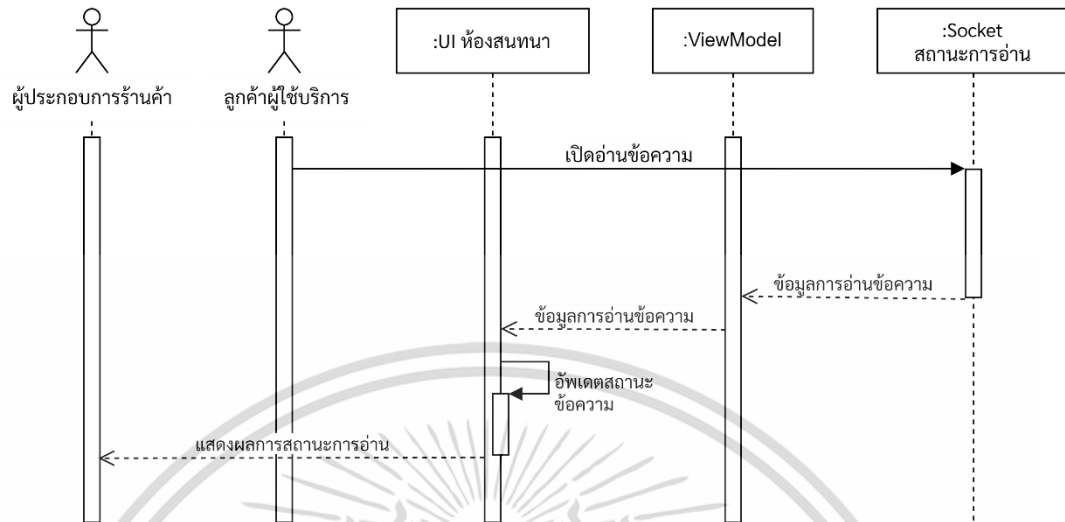
7) Sequence Diagram ของการส่งข้อความรูปภาพใหม่อีกครั้ง



รูปที่ 3.10 แผนภาพ Sequence Diagram ของการส่งข้อความรูปภาพใหม่อีกครั้ง

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.10 การทำงานของการส่งข้อความรูปภาพใหม่อีกครั้งจะเริ่มต้นเมื่อมีข้อความรูปภาพที่ถูกส่งไม่สำเร็จ ข้อความดังกล่าวจะปรากฏในห้องสนทนาพร้อมกับปุ่มส่งข้อความใหม่อีกครั้ง เมื่อผู้ประกอบการร้านค้าทำการกดปุ่มส่งข้อความใหม่อีกครั้ง ViewModel จะสืบข้อมูลของข้อความรูปภาพดังกล่าวออกจาก Local Database และ ViewModel จะทำการอัปโหลดรูปภาพดังกล่าวขึ้นไปฝากไว้ที่ CDN Service ใหม่อีกครั้ง โดยเมื่อทำการอัปโหลดสำเร็จ CDN Service จะคืนค่า URL ที่ใช้แสดงผลภาพดังกล่าว จากนั้น ViewModel จะทำการส่งข้อมูลข้อความรูปภาพพร้อมด้วย URL ให้ลูกค้าผู้ใช้บริการผ่านทาง Socket เมื่อข้อความรูปภาพที่ส่งใหม่อีกครั้งถูกส่งไปได้สำเร็จ Socket จะคืนค่าเวลาที่ข้อความถูกส่งได้สำเร็จกลับมาเพื่อให้แสดงผลแก่ผู้ประกอบการร้านค้าว่าข้อความรูปภาพดังกล่าวถูกส่งสำเร็จ แต่ถ้าหากการส่งข้อความรูปภาพยังคงล้มเหลวอีก Socket จะคืนค่ากลับมาให้ทราบว่า การส่งข้อความรูปภาพล้มเหลว ปุ่มส่งข้อความใหม่อีกครั้งด้านข้างข้อความรูปภาพจะกลับมาปรากฏขึ้นใหม่อีกครั้ง และหลังจากนั้น ViewModel จะส่งข้อมูลข้อความรูปภาพที่ส่งไม่สำเร็จดังกล่าวไปบันทึกลงใน Local Database เพื่อใช้ในการส่งข้อความใหม่อีกครั้ง

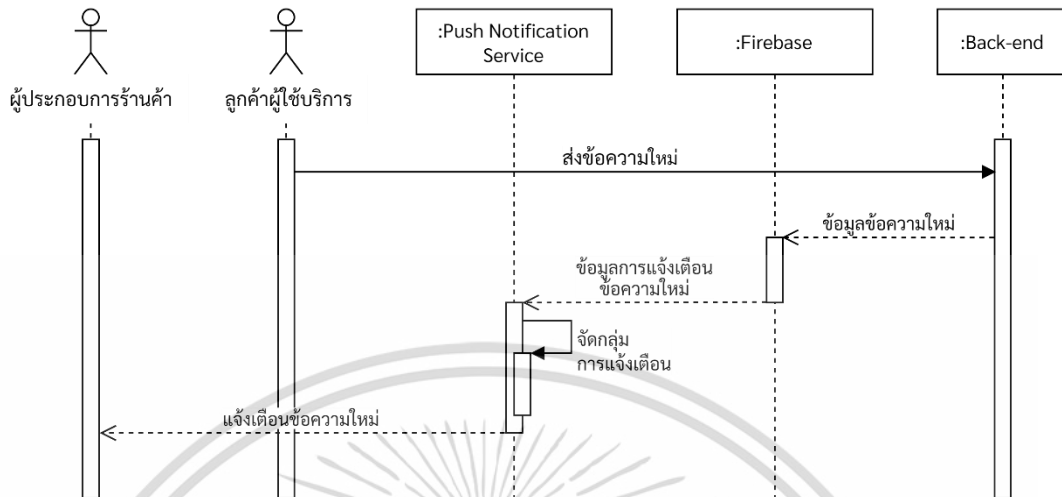
8) Sequence Diagram ของการระบุสถานะการอ่าน



รูปที่ 3.11 แผนภาพ Sequence Diagram ของการระบุสถานะการอ่าน

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.11 การทำงานของการระบุสถานะการอ่านจะเริ่มต้นหากผู้ประกอบการร้านค้าได้ส่งข้อความใด ๆ เข้าสู่ห้องสนทนา ข้อความนั้นในห้องสนทนาปรากฏจะเพียงตัวข้อความและเวลาที่ข้อความถูกส่งสำเร็จเท่านั้น จะยังไม่ปรากฏข้อความที่บ่งบอกสถานะการอ่าน แต่เมื่อลูกค้าผู้ใช้บริการได้อ่านข้อความดังกล่าวด้วยการเข้าสู่ห้องสนทนา ทางแอปพลิเคชันของลูกค้าผู้ใช้บริการจะส่งข้อมูลการอ่านข้อความกลับมาผ่าน Socket และ ViewModel จะส่งต่อข้อมูลการอ่านไปยังหน้าห้องสนทนาเพื่ออัปเดตข้อมูลของข้อความบนหน้าจอ เพื่อให้ปรากฏข้อความด้านข้างบ่งบอกว่าข้อความที่ถูกส่งจากผู้ประกอบการร้านค้านั้นถูกอ่านแล้วโดยลูกค้าผู้ใช้บริการ

9) Sequence Diagram ของการแจ้งเตือนข้อความใหม่

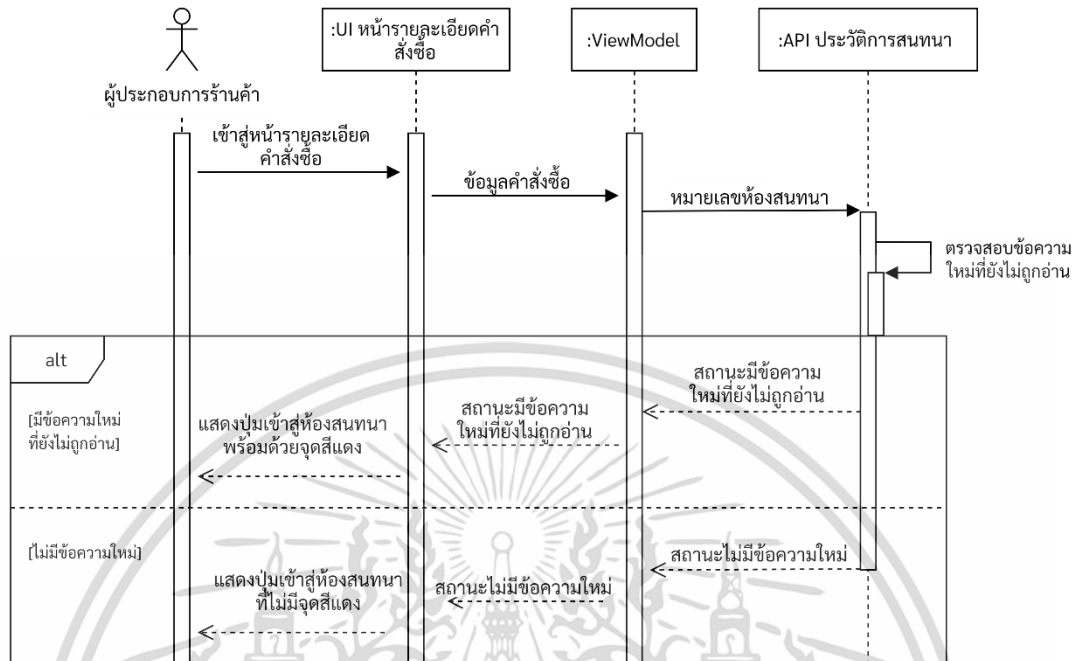


รูปที่ 3.12 แผนภาพ Sequence Diagram ของการแจ้งเตือนข้อความใหม่

ผู้ประกอบการร้านค้าจะได้รับการแจ้งเตือนข้อความใหม่จากแอปพลิเคชัน เมื่อลูกค้าผู้ใช้บริการได้มีการส่งข้อความใด ๆ มายังผู้ประกอบการร้านค้า ข้อความจะถูกส่งเข้าระบบหลังบ้าน (Back-end) ผ่านทาง Socket ซึ่งข้อมูลข้อความดังกล่าวจะถูกส่งต่อไปยัง Firebase เพื่อทำการส่งข้อมูลไปแจ้งเตือนผู้ประกอบการร้านค้า โดยในแอปพลิเคชันของผู้ประกอบการร้านค้าจะมี Push Notification Service ที่คอยรอรับข้อมูลข้อความใหม่จาก Firebase และเมื่อมีข้อมูลจาก Firebase ถูกส่งเข้ามา Push Notification Service จะทำการส่งเสียงแจ้งเตือนและปรากฏการแจ้งเตือนข้อความใหม่บนหน้าจอ ดังที่ปรากฏตาม Sequence Diagram ในรูปที่ 3.12 ซึ่งการแจ้งเตือนจะประกอบไปด้วยชื่อของลูกค้าผู้ใช้บริการที่เป็นผู้ส่งข้อความ และข้อความใหม่ที่ผู้ประกอบการลูกค้าผู้ใช้บริการได้ส่งมา นอกจากนี้การแจ้งเตือนข้อความจะถูกจับกลุ่มตามรายการคำสั่งซื้อ กล่าวคือข้อความใหม่จากหลายคำสั่งซื้อ ข้อความดังกล่าวจะไม่ปะปนกันแต่จะเรียงเกาะกลุ่มกันตามคำสั่งซื้อนั้น ๆ เพื่อลดความสับสนในการอ่านการแจ้งเตือนเมื่อมีข้อความจากหลายคำสั่งซื้อถูกส่งเข้ามาพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) Sequence Diagram ของการแสดงผลปุ่มเข้าสู่ห้องสนทนาพร้อมด้วยจุดสีแดง



รูปที่ 3.13 แผนภาพ Sequence Diagram ของการแสดงผลปุ่มเข้าสู่ห้องสนทนาพร้อมด้วยจุดสีแดง

จากแผนภาพ Sequence Diagram ที่ปรากฏในรูปที่ 3.13 การทำงานของการแสดงผลปุ่มเข้าสู่ห้องสนทนาพร้อมด้วยจุดสีแดงจะเริ่มต้นเมื่อผู้ประกอบการร้านค้าเข้าสู่หน้ารายละเอียดคำสั่งซื้อเพื่อตรวจสอบรายการอาหาร แอปพลิเคชันจะทำการส่งข้อมูลของคำสั่งไปยังระบบหลังบ้านเพื่อทำการตรวจสอบว่าห้องสนทนาของคำสั่งซื้อที่ถูกเปิดแสดงรายละเอียดในปัจจุบันนั้นมีข้อความจากลูกค้าผู้ใช้บริการที่ยังไม่ถูกอ่านโดยผู้ประกอบการร้านค้าหรือไม่ หากมีข้อความใหม่จากลูกค้าผู้ใช้บริการส่งเข้ามาแล้วยังไม่ถูกอ่านโดยผู้ประกอบการร้านค้า ระบบหลังบ้านจะส่งข้อมูลสถานะว่าข้อความใหม่และยังไม่ถูกเปิดอ่านกลับมายังแอปพลิเคชัน เพื่อให้แสดงผลปุ่มเข้าสู่ห้องสนทนาที่จะปรากฏจุดสีแดงอยู่มุมด้านบนขวาของปุ่ม เป็นการทำสัญลักษณ์แจ้งเตือนผู้ประกอบการร้านค้าให้ทราบว่ามีความใหม่จากลูกค้าผู้ใช้บริการส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการพัฒนาและการอภิปรายผล

จากการดำเนินงานการพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิทเชน (Cloud Kitchen Platform) ที่ได้ถูกพัฒนาขึ้นตามที่ได้มีการออกแบบไว้ในบทที่ 3 โดยการพัฒนาพีเจอาร์การสนทนาดังกล่าวจะถูกพัฒนาสำหรับแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ (Android Application) ซึ่งจะดำเนินการพัฒนารวมถึงทำการทดสอบระบบด้วยชุดคำสั่งต่าง ๆ ขึ้นที่เขียนขึ้นด้วยภาษา Kotlin บนโปรแกรม Android Studio



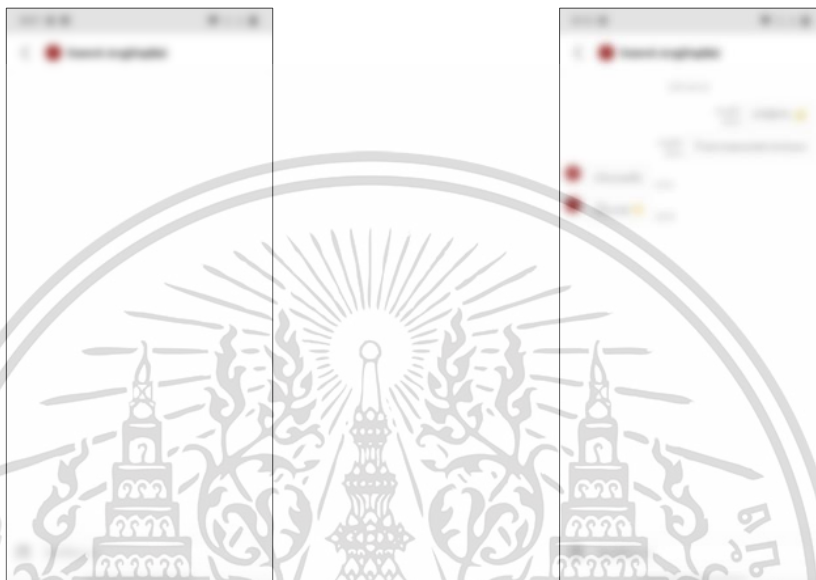
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ส่วนต่อประสานผู้ใช้

4.1.1 ภายในห้องสนทนา

4.1.1.1 การแสดงประวัติการสนทนา

- 1) ห้องสนทนา กรณีคำสั่งซื้ออยู่ภายในระยะเวลา 24 ชั่วโมง



ก) ห้องสนทนาว่างเปล่า

ข) ห้องสนทนามีการสนทนา

รูปที่ 4.1 ห้องสนทนา กรณีคำสั่งซื้ออยู่ภายในระยะเวลา 24 ชั่วโมง

เนื่องด้วยห้องสนทนานั้นเป็นห้องสนทนาชั่วคราวที่มีอายุการใช้งานเพียง 24 ชั่วโมง และห้องสนทนาจะถูกสร้างขึ้นตามคำสั่งซื้อ กล่าวคือคำสั่งซื้อแต่ละรายการจะมีห้องสนทนาเป็นของตนเอง หากคำสั่งซื้อใด ๆ อยู่ภายในระยะเวลา 24 ชั่วโมง เมื่อผู้ประกอบการร้านค้ากดปุ่มเข้าสู่ห้องสนทนาผ่านหน้ารายละเอียดคำสั่งซื้อของคำสั่งซื้อนั้น ๆ ผู้ประกอบการร้านค้าจะเข้าสู่ห้องสนทนาและสามารถทำการรับส่งข้อความได้ ซึ่งโดยเริ่มต้นแล้วห้องสนทนาที่ยังไม่มีการรับและส่งข้อความใด ๆ จะแสดงผลเป็นห้องสนทนาที่ว่างเปล่าดังรูปที่ 4.1 ก. และเมื่อการสนทนาได้รับและส่งข้อความเกิดขึ้นจะแสดงผลประวัติการสนทนาเป็นกล่องข้อความที่เรียงกันจากบนล่างตามเวลาที่ข้อความถูกส่ง โดยข้อความที่ถูกส่งจากลูกค้าผู้ใช้บริการจะอยู่ชิดทางซ้ายของห้องสนทนา ส่วนข้อความที่ถูกส่งจากผู้ประกอบการร้านค้าจะอยู่ชิดทางขวาของห้องสนทนาดังที่ปรากฏใน รูปที่ 4.1 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ห้องสนทนา กรณีคำสั่งซื้อมีระยะเวลามากกว่า 24 ชั่วโมง

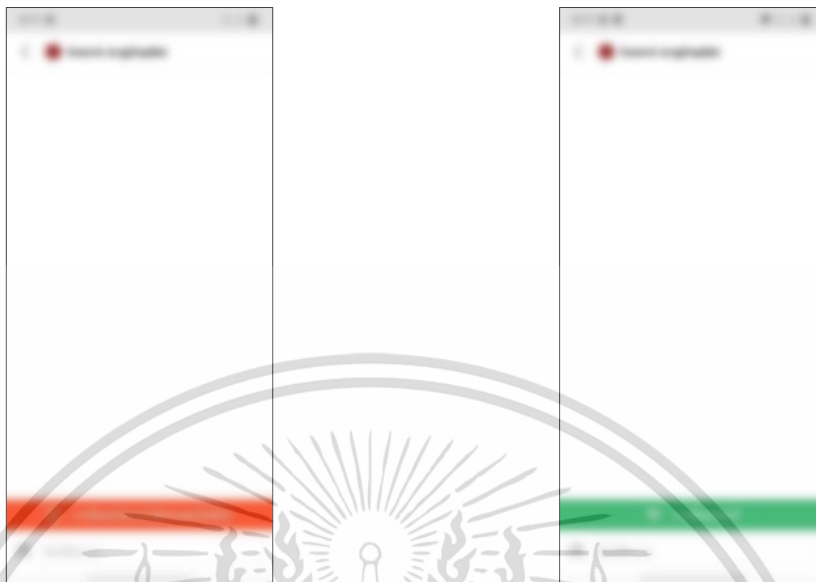


รูปที่ 4.2 ห้องสนทนา กรณีคำสั่งซื้อมีระยะเวลามากกว่า 24 ชั่วโมง

หากคำสั่งซื้อมีระยะเวลามากกว่า 24 ชั่วโมง เมื่อผู้ประกอบการร้านค้ากดปุ่มเข้าสู่ห้องสนทนาผ่านหน้ารายละเอียดคำสั่งซื้อ จะปรากฏหน้า Empty State ดังที่ปรากฏในรูปที่ 4.2 เนื่องด้วยห้องสนทนานั้นเป็นห้องสนทนาชั่วคราวที่มีอายุการใช้งานเพียง 24 ชั่วโมง และห้องสนทนาจะถูกสร้างขึ้นตามคำสั่งซื้อ กล่าวคือ คำสั่งซื้อแต่ละรายการจะมีห้องสนทนาเป็นของตนเอง ข้อความในห้องสนทนาหนึ่ง จะไม่ปะปนกับห้องสนทนาอื่น ๆ แม้จะเป็นการสนทนากับลูกค้าผู้ใช้บริการและผู้ประกอบการร้านค้าคนเดียวกัน เช่น หากลูกค้าคนหนึ่งทำการสั่งซื้ออาหารกับผู้ประกอบการร้านค้าคนเดิม 2 ครั้ง ก็จะเป็นการทำรายการคำสั่งซื้อ 2 ครั้งและจะเกิดเป็นห้องสนทนา 2 ห้องที่ไม่เกี่ยวข้องกัน เมื่อผ่านไป คำสั่งซื้อสิ้นสุดลง และทางผู้ประกอบการร้านค้ากับลูกค้าผู้ใช้บริการได้สิ้นสุดการสนทนาที่เกี่ยวข้องกับคำสั่งซื้อนั้น ๆ แล้ว ทางระบบการสนทนาจะทำการปิดห้องสนทนา ทำให้ผู้ประกอบการร้านค้าจะไม่สามารถดูประวัติการส่งสนทนา รวมถึงไม่สามารถรับและส่งข้อความได้อีกต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.1 การแสดงสถานะสัญญาณอินเทอร์เน็ตขัดข้อง



ก) สัญญาณอินเทอร์เน็ตขัดข้อง

ข) สัญญาณอินเทอร์เน็ตกลับมาใช้งานได้

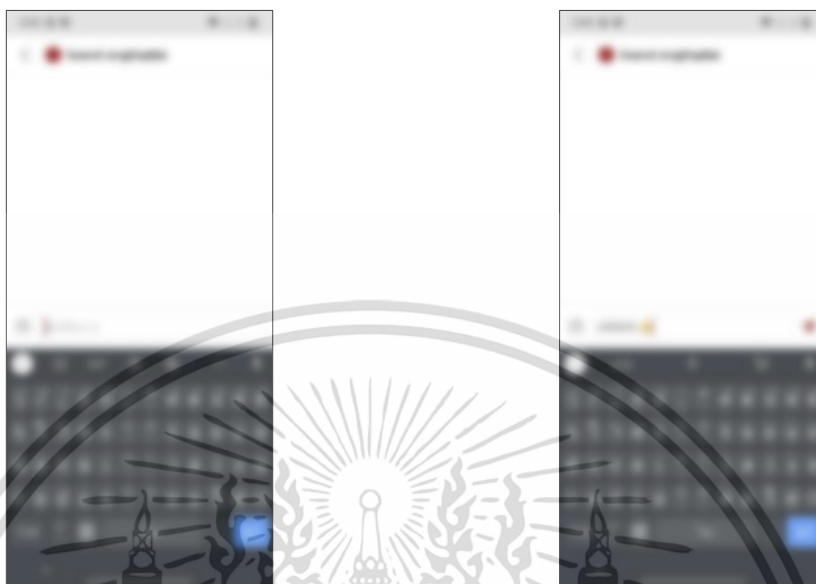
รูปที่ 4.3 ห้องสนทนา เมื่อสัญญาณอินเทอร์เน็ตขัดข้อง

เมื่อสัญญาณอินเทอร์เน็ตเกิดขัดข้องในขณะที่ร้านค้าผู้ให้บริการกำลังอยู่ในห้องสนทนา แอปสถานะสีแดงที่ระบุข้อความบ่งบอกถึงสถานะสัญญาณอินเทอร์เน็ตขัดข้องปรากฏขึ้นให้ผู้ประกอบการทราบตามที่แสดงอยู่ในรูปที่ 4.3 ก. และเมื่อสัญญาณอินเทอร์เน็ตกลับมาใช้งานได้อีกครั้ง แอปสถานะสีแดงจะเปลี่ยนเป็นสีเขียวและระบุข้อความบ่งบอกให้ผู้ประกอบการร้านค้าทราบว่าอินเทอร์เน็ตกลับมาใช้งานได้แล้ว ดังที่ปรากฏในรูปที่ 4.3 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2 การรับและส่งข้อความอักษร

1) การพิมพ์ข้อความอักษร



ก) ข้อความว่างเปล่า

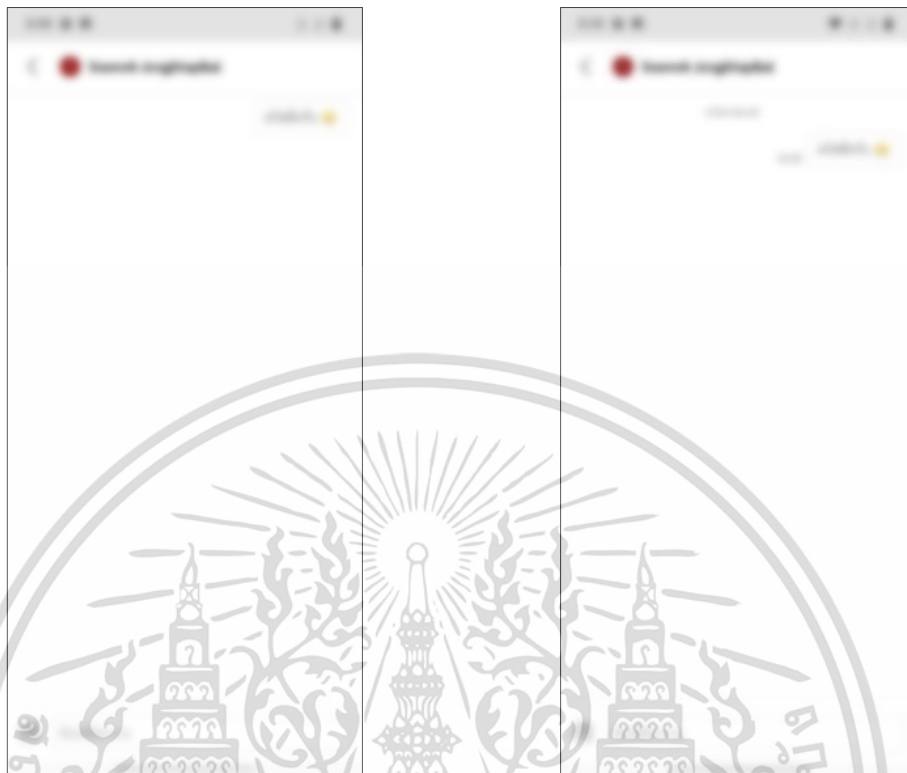
ข) ข้อความถูกพิมพ์

รูปที่ 4.4 การพิมพ์ข้อความ

ผู้ประกอบการร้านค้าสามารถพิมพ์ข้อความได้ทั้งที่เป็นข้อความตัวอักษรและอีโมจิลงในช่องรับข้อความ ซึ่งในแต่ละข้อความนั้นจะมีความยาวได้ไม่เกิน 128 ตัวอักษร และผู้ประกอบการร้านค้าสามารถส่งข้อความที่พิมพ์ไว้ในช่องรับข้อความได้โดยการกดปุ่มส่งข้อความด้านขวามือดังที่ปรากฏในรูปที่ 4.4 ข. ซึ่งปุ่มส่งข้อความดังกล่าวจะไม่ปรากฏขึ้นหากยังไม่มีข้อความใด ๆ ถูกพิมพ์ลงไปช่องรับข้อความตามรูปที่ 4.4 ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การส่งข้อความอักษร



ก) ข้อความอักษรกำลังถูกส่ง

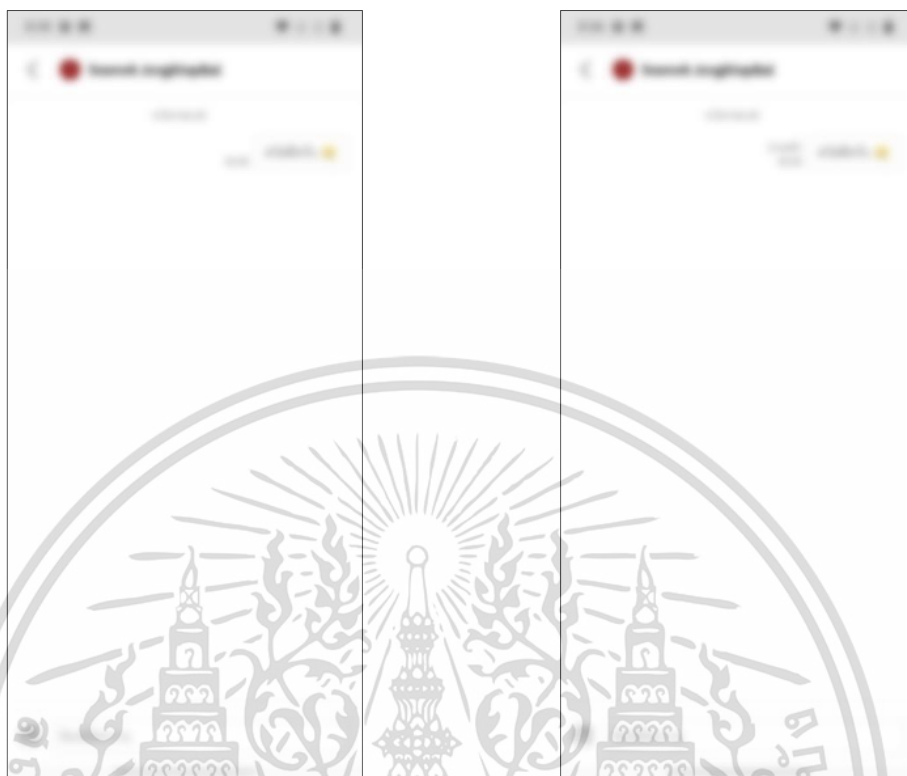
ข) ข้อความอักษรถูกส่งสำเร็จ

รูปที่ 4.5 การส่งข้อความ

เมื่อผู้ประกอบการร้านค้าพิมพ์ข้อความอักษรในช่องรับข้อความและกดปุ่มส่งข้อความ ข้อความอักษรที่อยู่ในช่องรับข้อความจะถูกส่งยังห้องสนทนาและจะแสดงเป็นกล่องข้อความอยู่ทางขวาของหน้าจอโดยจะยังไม่ปรากฏเวลาที่ทำการส่งข้อความได้สำเร็จ ดังรูปที่ 4.5 ก. และเมื่อข้อความดังกล่าวถูกส่งได้สำเร็จจะปรากฏเวลาที่ทำการส่งข้อความได้สำเร็จข้างของกล่องข้อความตามรูปที่ 4.5 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การแสดงสถานะการอ่านของข้อความอักษร



ก) ข้อความอักษรที่ยังไม่ถูกอ่าน

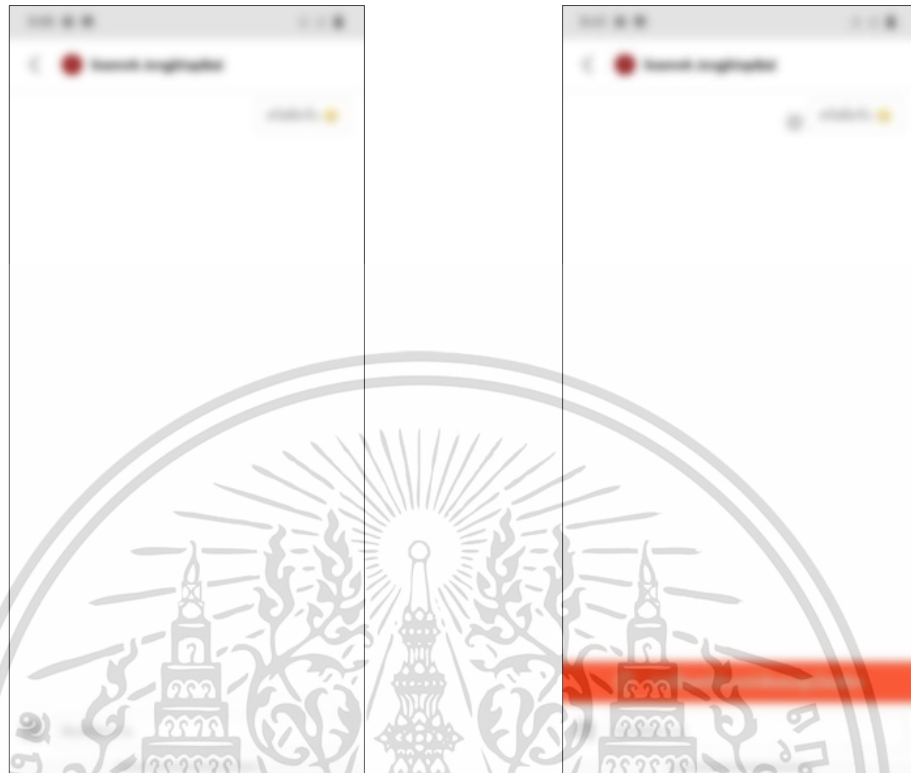
ข) ข้อความอักษรที่ถูกอ่านแล้ว

รูปที่ 4.6 การแสดงสถานะการอ่านของข้อความอักษร

เมื่อผู้ประกอบการร้านค้าส่งข้อความอักษรได้สำเร็จ จะปรากฏกล่องข้อความที่ระบุเวลาที่ทำการส่งข้อความดังกล่าวสำเร็จ แต่จะยังไม่ปรากฏข้อความสถานะระบุการอ่าน ตามรูปที่ 4.6 ก. แต่เมื่อลูกค้าผู้ใช้บริการเข้าสู่ห้องสนทนาและทำการอ่านข้อความดังกล่าวแล้ว ข้อความรูปภาพที่ถูกส่งโดยผู้ประกอบการร้านค้าทั้งหมดจะปรากฏข้อความสถานะระบุการอ่านด้านข้างกล่องข้อความเป็นการแจ้งให้ผู้ประกอบการร้านค้าทราบว่าลูกค้าได้ทำการเปิดอ่านข้อความแล้ว ดังรูปที่ 4.6 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การแสดงปุ่มส่งข้อความอักษรใหม่อีกครั้ง



ก) ข้อความอักษรกำลังถูกส่ง

ข) ข้อความอักษรถูกส่งไม่สำเร็จ

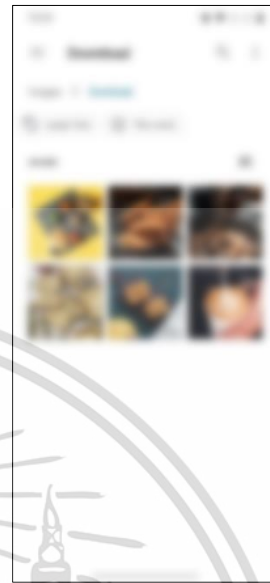
รูปที่ 4.7 ข้อความอักษรที่ถูกส่งไม่สำเร็จ

หากสัญญาณอินเทอร์เน็ตเกิดขัดข้องในขณะที่ผู้ประกอบการร้านค้ากำลังทำการส่งข้อความอักษร ในช่วงขณะหนึ่ง กล่องข้อความดังกล่าวจะยังไม่ปรากฏเวลาที่ทำการส่งข้อความได้สำเร็จตามรูปที่ 4.7 ก. แต่เมื่อเวลาผ่านไปปุ่มส่งข้อความใหม่อีกครั้งจะปรากฏขึ้นด้านข้างกล่องข้อความที่ส่งไม่สำเร็จ เพื่อเป็นการแจ้งให้ผู้ประกอบการร้านค้าทราบว่าข้อความดังกล่าวนั้นถูกส่งไม่สำเร็จและพร้อมจะถูกส่งใหม่อีกครั้งเมื่อผู้ประกอบการร้านค้าปุ่มส่งข้อความใหม่อีกครั้ง ดังรูปที่ 4.7 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

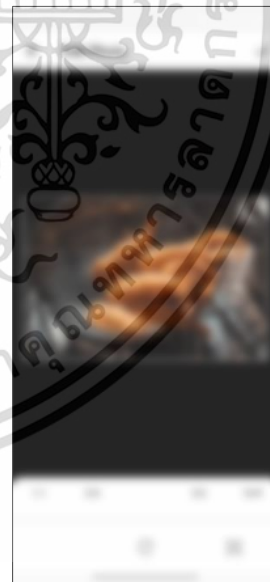
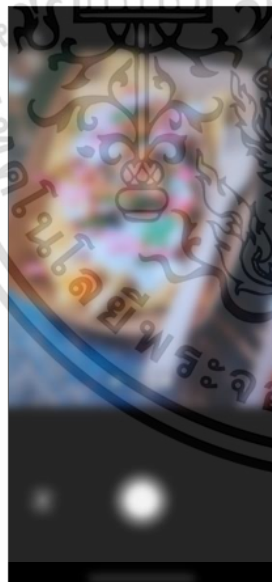
4.1.1.3 การรับและส่งข้อความรูปภาพ

1) การเลือกรูปภาพ



ก) การเลือกรูปจากอุปกรณ์
หรือถ่ายรูปผ่านกล้อง

ข) การเลือกรูปจากอุปกรณ์



ค) การถ่ายรูปผ่านกล้อง

ง) การปรับแต่งรูป

รูปที่ 4.8 การเลือกรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับร้านค้าผู้ใช้บริการสามารถส่งข้อความรูปได้ เมื่อใดที่ปุ่มส่งรูปภาพจะไม่ว่ากรณีใดๆ ทั้งสิ้น อีก ป้ารากฏตัวเลือกให้ด้วยการเลือกรูปภาพจากอุปกรณ์หรือการถ่ายรูปผ่านกล้องดังใน

รูป 4.8 ก. หากเลือกการเลือกรูปภาพจากอุปกรณ์ จะสามารถเลือกรูปภาพต่าง ๆ จากในอุปกรณ์ได้ตามรูปที่ 4.8 ข. แต่ถ้าหากเลือกการถ่ายรูปผ่านกล้องจะสามารถถ่ายรูปใหม่ได้ตามรูปที่ 4.8 ค. และเมื่อเลือกรูปหรือถ่ายรูปใหม่เรียบร้อยแล้วสามารถดูรูปภาพดังกล่าวแบบเต็มจอและสามารถครอบตัดรูปภาพได้ก่อนทำการส่ง ดังรูป 4.8 ง.

2) การส่งข้อความรูปภาพ



ก) ข้อความรูปภาพกำลังถูกส่ง

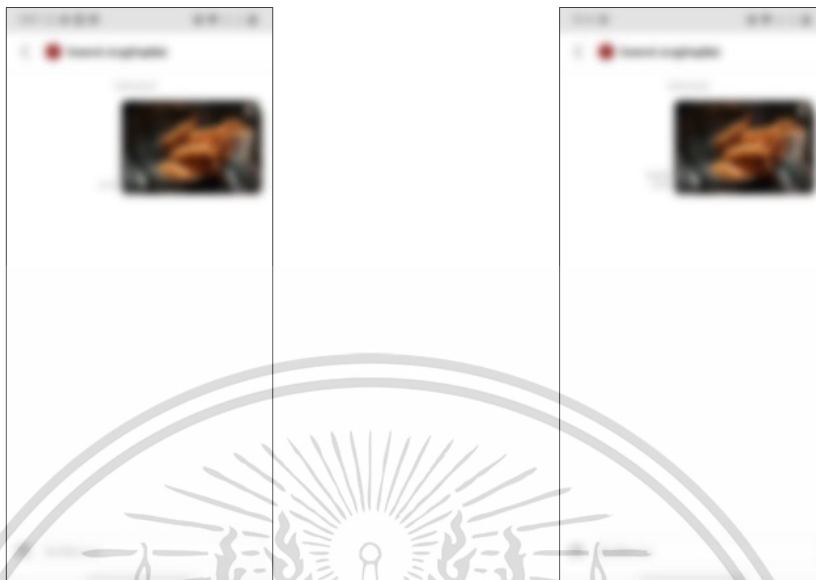
ข) ข้อความรูปภาพถูกส่งสำเร็จ

รูปที่ 4.9 การส่งข้อความรูปภาพ

เมื่อผู้ประกอบการร้านค้าเลือกรูปภาพจากอุปกรณ์หรือถ่ายภาพและทำการส่งรูปภาพ รูปภาพดังกล่าวจะถูกส่งยังห้องสนทนาและจะแสดงเป็นกล่องข้อความอยู่ทางขวาของหน้าจอโดยจะยังไม่ปรากฏเวลาที่ทำการส่งข้อความได้สำเร็จ ดังรูปที่ 4.9 ก. และเมื่อข้อความรูปภาพดังกล่าวถูกส่งได้สำเร็จจะปรากฏเวลาที่ทำการส่งข้อความได้สำเร็จดังข้างของกล่องข้อความตามรูปที่ 4.9 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) การแสดงสถานะการอ่านของข้อความรูปภาพ



ก) ข้อความรูปภาพที่ยังไม่ถูกอ่าน

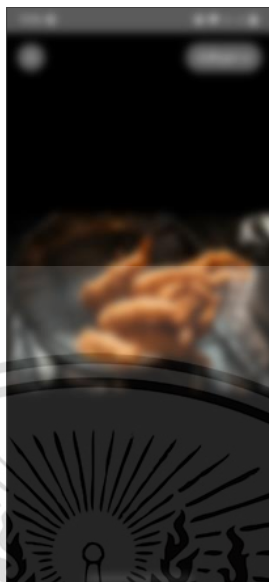
ข) ข้อความรูปภาพที่ถูกอ่านแล้ว

รูปที่ 4.10 การแสดงสถานะการอ่านของข้อความรูปภาพ

เมื่อผู้ประกอบการร้านค้าส่งข้อความรูปภาพได้สำเร็จ จะปรากฏกล่องข้อความที่ระบุเวลาที่ทำการส่งข้อความดังกล่าวสำเร็จ แต่จะยังไม่ปรากฏข้อความสถานะการอ่าน ตามรูปที่ 4.10 ก. แต่เมื่อลูกค้าผู้ใช้บริการเข้าสู่ห้องสนทนา และทำการอ่านข้อความรูปภาพดังกล่าวแล้ว ข้อความที่ถูกส่งโดยผู้ประกอบการร้านค้าทั้งหมดจะปรากฏข้อความสถานะการอ่านด้านข้างกล่องข้อความเป็นการแจ้งให้ผู้ประกอบการร้านค้าทราบว่าลูกค้าได้ทำการเปิดอ่านข้อความแล้ว ดังที่ปรากฏในรูปที่ 4.10 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การแสดงผลรูปภาพเต็มจอ

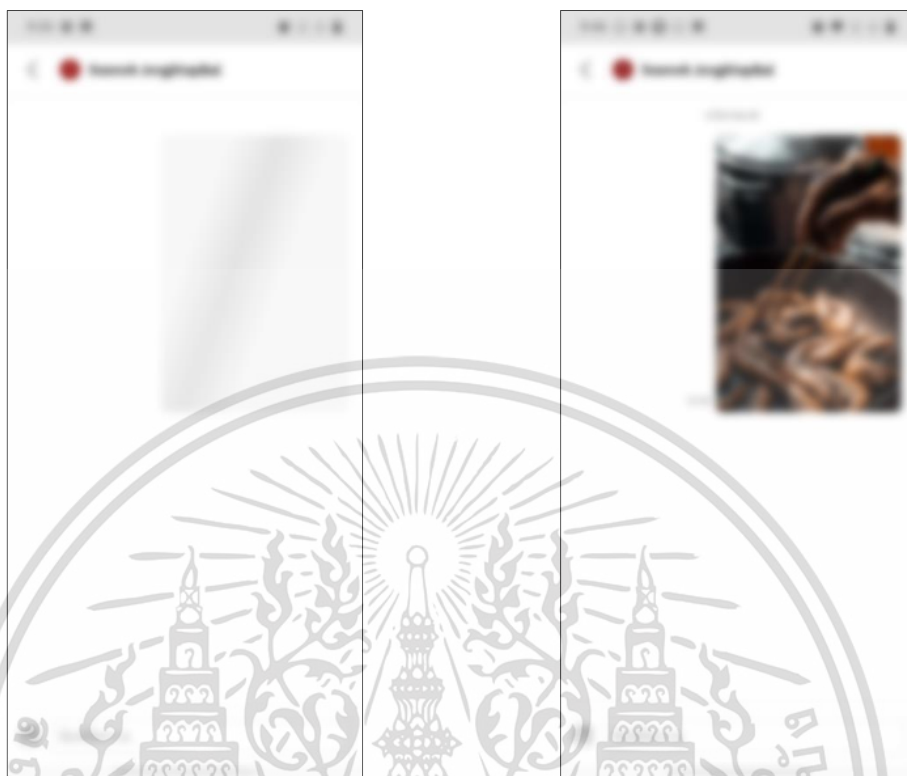


รูปที่ 4.11 การแสดงผลรูปภาพเต็มจอ

เมื่อข้อความรูปภาพถูกส่งได้สำเร็จทั้งที่ถูกส่งจากผู้ประกอบการร้านค้า และลูกค้าผู้ใช้บริการปรากฏอยู่ในห้องสนทนา ผู้ประกอบการร้านค้าสามารถทำการกดที่ข้อความรูปภาพ เพื่อให้รูปภาพนั้น ๆ สามารถแสดงผลแบบเต็มหน้าจอได้ โดยทางมุมซ้ายบนของหน้าจอเป็นปุ่มกดเพื่อกลับไปยังหน้าห้องสนทนา หรือจะกดปุ่มทางขวาบนเพื่อทำการบันทึกภาพดังกล่าวลงในอุปกรณ์ ดังที่ปรากฏในรูป 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การแสดงปุ่มส่งข้อความรูปภาพใหม่อีกครั้ง



ก) ข้อความรูปภาพกำลังถูกส่ง

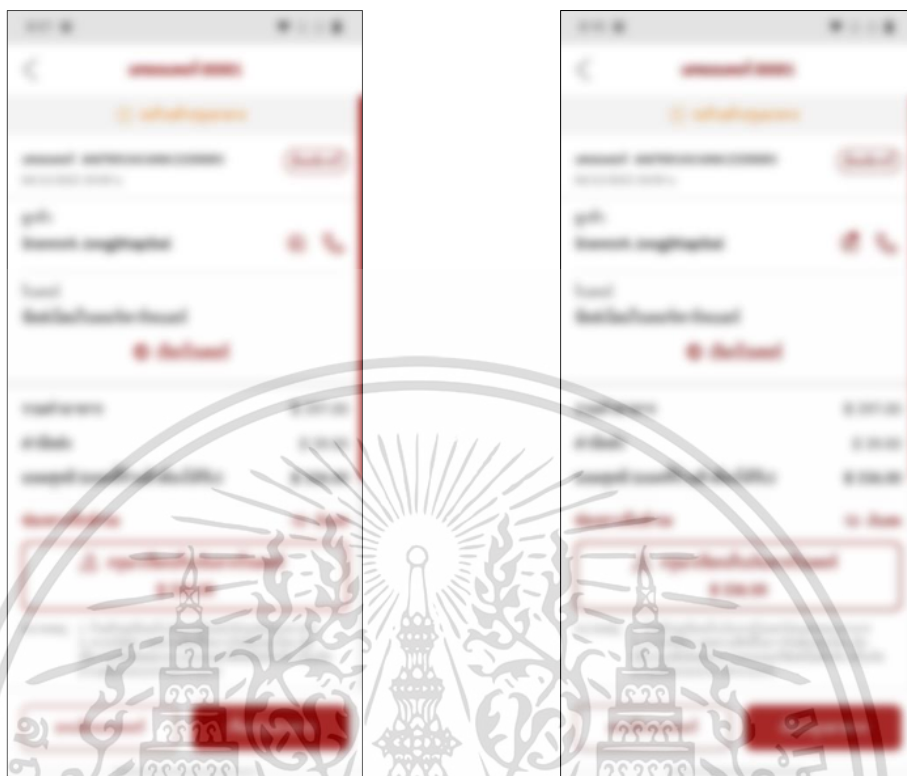
ข) ข้อความรูปภาพถูกส่งไม่สำเร็จ

รูปที่ 4.12 ข้อความรูปภาพที่ถูกส่งไม่สำเร็จ

หากสัญญาณอินเทอร์เน็ตเกิดขัดข้องในขณะที่ผู้ประกอบการร้านค้ากำลังทำการส่งข้อความรูปภาพ ในช่วงหนึ่ง กล้องข้อความรูปภาพดังกล่าวจะยังไม่ปรากฏเวลาที่ทำการส่งข้อความได้สำเร็จตามรูปที่ 4.12 ก. แต่เมื่อเวลาผ่านไปปุ่มส่งข้อความใหม่อีกครั้งจะปรากฏขึ้นด้านข้างกล้องข้อความที่ส่งไม่สำเร็จ เพื่อเป็นการแจ้งให้ผู้ประกอบการร้านค้าทราบว่าข้อความรูปภาพดังกล่าวนั้นถูกส่งไม่สำเร็จและพร้อมจะถูกส่งใหม่อีกครั้งเมื่อผู้ประกอบการร้านค้าปุ่มส่งข้อความใหม่อีกครั้ง ดังที่ปรากฏในรูปที่ 4.12 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 หน้ารายละเอียดคำสั่งซื้อ



ก) ปุ่มเข้าสู่ห้องสนทนา

ข) ปุ่มเข้าสู่ห้องสนทนาพร้อมด้วยจุดสีแดง

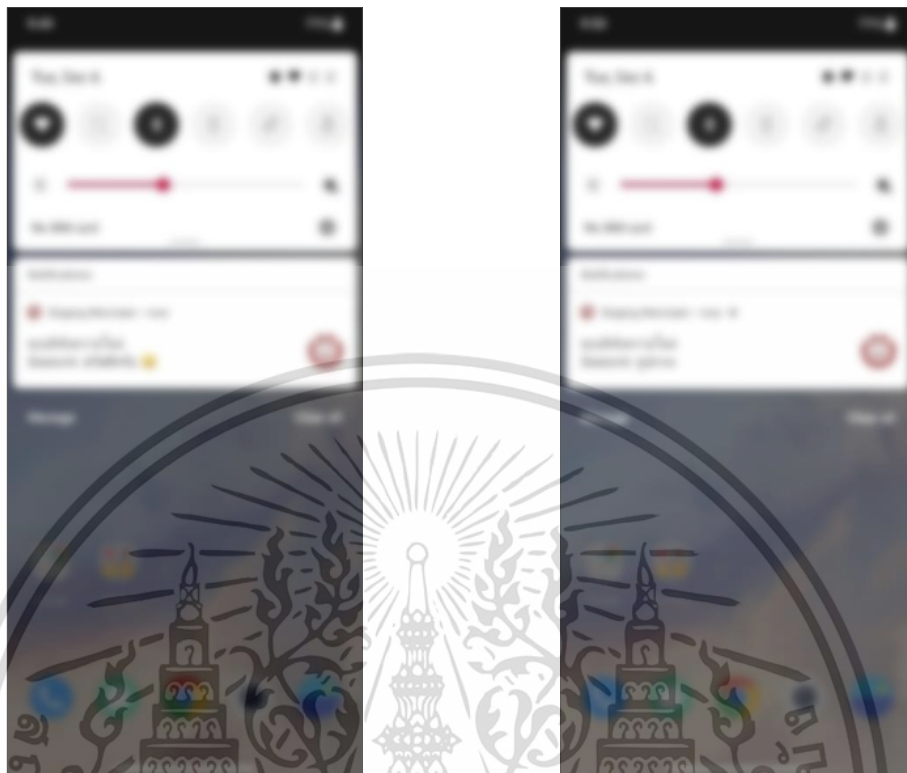
รูปที่ 4.13 ข้อความรูปภาพที่ถูกส่งไม่สำเร็จ

เมื่อผู้ประกอบการร้านค้าเข้าสู่หน้ารายละเอียดคำสั่งซื้อจะปรากฏรายละเอียดต่างๆ ของคำสั่งซื้อ รวมถึงช่องทางการติดต่อกับลูกค้าใช้บริการทำการสั่งซื้อทั้งการโทรด้วยเบอร์มือถือและการรับส่งข้อความผ่านระบบสนทนาของทางแพลตฟอร์มเอง โดยหากไม่มีข้อความที่ถูกส่งมาจากลูกค้าผู้ใช้บริการและยังไม่ถูกเปิดอ่านโดยผู้ประกอบการร้านค้า ปุ่มเข้าสู่ห้องสนทนาจะไม่ปรากฏจุดสีแดงตามรูปที่ 4.13 ก. แต่หากมีข้อความที่ถูกส่งมาจากลูกค้าผู้ใช้บริการและยังไม่ถูกเปิดอ่านโดยผู้ประกอบการร้านค้า ปุ่มเข้าสู่ห้องสนทนาดังกล่าวจะปรากฏจุดสีแดงอยู่บริเวณมุมขวาบนของปุ่มเพื่อเป็นการแจ้งให้ผู้ประกอบการร้านค้าทราบว่ามีความข้อความที่ถูกส่งมาจากลูกค้าผู้ใช้บริการ รูปที่ 4.13 ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การแจ้งเตือน

1) แจ้งเตือนข้อความใหม่



ก) การแจ้งเตือนข้อความอักษร

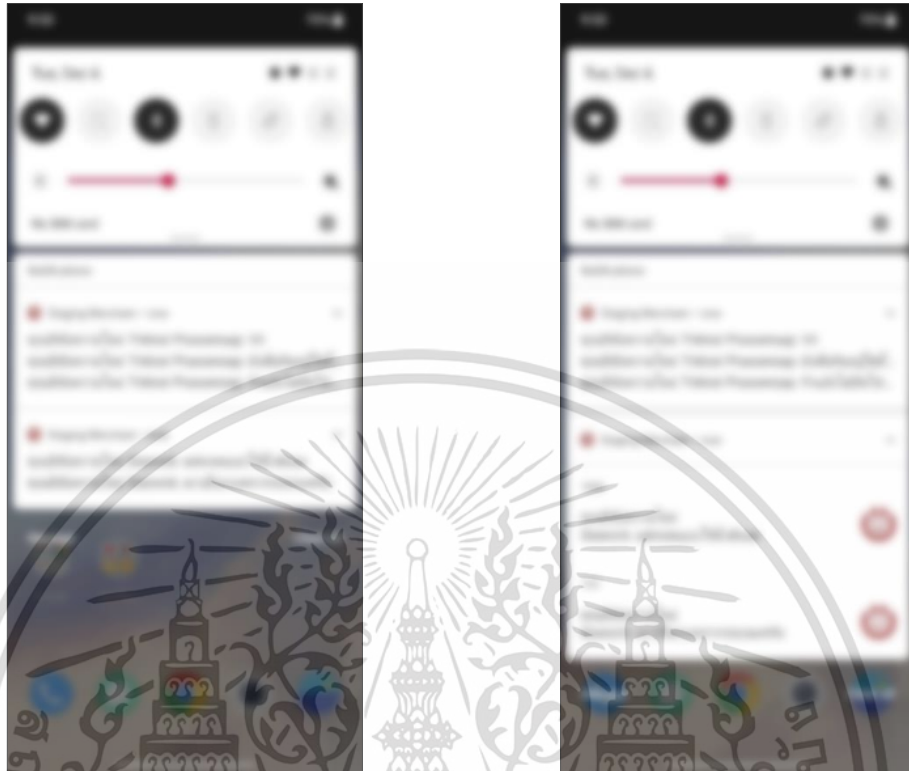
ข) การแจ้งเตือนข้อความรูปภาพ

รูปที่ 4.14 การแจ้งเตือนข้อความใหม่

เมื่อลูกค้าผู้ใช้บริการทำการส่งข้อความมายังผู้ประกอบการร้านค้า ผู้ประกอบการร้านค้าจะได้รับการแจ้งเตือน Push Notification โดยหากข้อความที่ถูกส่งมาเป็นข้อความอักษร การแจ้งเตือนจะแสดงผลชื่อของลูกค้าผู้ใช้บริการที่ส่งข้อความมาตามด้วยเนื้อหาของข้อความตามรูปที่ 4.14 ก. แต่หากข้อความที่ถูกส่งมาเป็นข้อความรูปภาพ การแจ้งเตือนจะแสดงผลชื่อของลูกค้าผู้ใช้บริการที่ส่งข้อความมาตามด้วยข้อความระบุว่าข้อความที่ถูกส่งมาใหม่นั้นเป็นข้อความรูปภาพ ดังที่ปรากฏในรูปที่ 4.14 ข. ซึ่งผู้ประกอบการร้านค้าสามารถกดที่การแจ้งเตือนทั้งสองแบบเพื่อเข้าสู่ห้องสนทนาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การจัดกลุ่มของการแจ้งเตือน



ก) กลุ่มของการแจ้งเตือน

ข) กลุ่มของการแจ้งเตือนที่ถูกวางออก

รูปที่ 4.15 การจัดกลุ่มของการแจ้งเตือน

เมื่อลูกค้าผู้ใช้บริการทำการสั่งซื้อความหลาย ๆ ข้อความมายังผู้ประกอบการร้านค้า ผู้ประกอบการร้านค้าจะได้รับการแจ้งเตือน Push Notification ที่ถูกจัดกลุ่มเอาไว้ตามรายการของคำสั่งซื้อ กล่าวคือข้อความจากคำสั่งซื้อหนึ่ง ๆ จะถูกจัดกลุ่มเอาไว้อยู่ด้วยกันและไม่ปะปนกับข้อความอื่น ๆ ดังที่ปรากฏในรูปที่ 4.15 ก. และในแต่ละกลุ่มข้อความนั้นสามารถกดเพื่อที่สามารถกดการแจ้งเตือนเพื่อดูการแจ้งเตือนแต่ละอันได้ ซึ่งผู้ประกอบการร้านค้าสามารถกดที่การแจ้งเตือนเพื่อเข้าสู่ห้องสนทนาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบระบบ

หลังจากการลงมือพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้า ขั้นตอนถัดไปเป็นการทำการทดสอบระบบเพื่อให้มั่นใจว่าการทำงานของชุดคำสั่งต่างๆนั้นถูกทำการทดสอบด้วย Unit Test ที่จะเป็นการเขียนชุดคำสั่งสำหรับการทดสอบแบบอัตโนมัติกับคลาสใดคลาสหนึ่ง ชุดคำสั่งดังกล่าวจะทำการส่งข้อมูลเข้าไปยังคลาสนั้น ๆ โดยตรง และเปรียบเทียบผลลัพธ์ที่ได้จากคลาสนั้นว่าเป็นไปตามคาดหวังหรือไม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การทดสอบ ViewModel

การทดสอบการทำงานของชุดคำสั่ง ๆ ภายในคลาส ViewModel ซึ่งเป็นคลาสสำคัญที่ทำหน้าที่เกี่ยวกับ Business Logic โดยจะมีการทดสอบการทำงานในกรณีต่าง ๆ ดังต่อไปนี้

ตารางที่ 4.1 Test Case ของ ViewModel

	Test Case Objective	Test Step	Test Data	Expected Result	Status
การโหลดประวัติ การสนทนา	การโหลดประวัติการสนทนาครั้งแรกโดยไม่เกิดข้อผิดพลาด	1) ตั้งค่าหมายเลขห้องสนทนาเป็น 234 2) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนา	1) หมายเลขห้องสนทนา 234	มีข้อมูลประวัติการสนทนาในอาเรย์	ผ่าน
การโหลดประวัติ การสนทนา	การโหลดประวัติการสนทนาเพิ่มอีกครั้งโดยไม่เกิดข้อผิดพลาด	1) ตั้งค่าหมายเลขห้องสนทนาเป็น 234 2) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนา 3) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนาเพิ่มอีกครั้ง	1) หมายเลขห้องสนทนา 234	มีข้อมูลประวัติการสนทนาเก่ากว่าในอาเรย์	ผ่าน
การโหลดประวัติ การสนทนา	การโหลดประวัติการสนทนาล่าสุดอีกครั้งโดยไม่เกิดข้อผิดพลาด	1) ตั้งค่าหมายเลขห้องสนทนาเป็น 234 2) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนา 3) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนาล่าสุดอีกครั้ง	1) หมายเลขห้องสนทนา 234	มีข้อมูลประวัติการสนทนาล่าสุดในอาเรย์	ผ่าน

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การดักข้อผิดพลาดจากการโหลดประวัติการสนทนา	การดักข้อผิดพลาดที่เกิดจากการโหลดประวัติการสนทนาครั้งแรก	1) ตั้งค่าหมายเลขห้องสนทนาเป็น -234 2) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนา	1) หมายเลขห้องสนทนา -234	ค่า Error ที่ระบุสาเหตุของความผิดพลาดมาจากการโหลดประวัติการสนทนา	ผ่าน
การดักข้อผิดพลาดจากการโหลดประวัติการสนทนา	การดักข้อผิดพลาดที่เกิดจากการโหลดประวัติการสนทนาที่เพิ่มอีกครั้ง	1) ตั้งค่าหมายเลขห้องสนทนาเป็น 234 2) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนา 3) ตั้งค่าหมายเลขห้องสนทนาเป็น -234 4) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนาเพิ่มอีกครั้ง	1) หมายเลขห้องสนทนา 234 2) หมายเลขห้องสนทนา -234	ค่า Error ที่ระบุสาเหตุของความผิดพลาดมาจากการโหลดประวัติการสนทนาที่เก่ากว่าข้อความปัจจุบัน	ผ่าน
การดักข้อผิดพลาดจากการโหลดประวัติการสนทนา	การดักข้อผิดพลาดที่เกิดจากการโหลดประวัติการสนทนาล่าสุดอีกครั้ง	1) ตั้งค่าหมายเลขห้องสนทนาเป็น 234 2) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนา 3) ตั้งค่าหมายเลขห้องสนทนาเป็น -234 4) เรียกฟังก์ชันที่ทำการขอข้อมูลประวัติการสนทนาล่าสุดอีกครั้ง	1) หมายเลขห้องสนทนา 234 2) หมายเลขห้องสนทนา -234	ค่า Error ที่ระบุสาเหตุของความผิดพลาดมาจากการโหลดประวัติการสนทนาล่าสุดอีกครั้ง	ผ่าน

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การแสดงผลหน้าประวัติการสนทนาเมื่อคำสั่งซื้อมีระยะเวลาอยู่ภายใน 24 ชั่วโมง	การแสดงผลหน้าประวัติการสนทนา เมื่อคำสั่งซื้อมีระยะเวลา 1 นาที	1) เรียกฟังก์ชันที่ทำการเปรียบเทียบเวลาปัจจุบันและเวลาที่สร้างคำสั่งซื้อว่ามากหรือน้อยกว่า 24 ชั่วโมง	1) เวลาที่สร้างคำสั่งซื้อที่มีระยะเวลา 1 นาที “2022-12-08T10:39:18.845Z”	คืนค่า True เพื่อทำการแสดงผลหน้าประวัติการสนทนาต่อไป	ผ่าน
การแสดงผลหน้าประวัติการสนทนาเมื่อคำสั่งซื้อมีระยะเวลาอยู่ภายใน 24 ชั่วโมง	การแสดงผลหน้า Empty State เมื่อคำสั่งซื้อมีระยะเวลา 23 ชั่วโมง 59 นาที	1) เรียกฟังก์ชันที่ทำการเปรียบเทียบเวลาปัจจุบันและเวลาที่สร้างคำสั่งซื้อว่ามากหรือน้อยกว่า 24 ชั่วโมง	1) เวลาที่สร้างคำสั่งซื้อที่มีระยะเวลา 23 ชั่วโมง 59 นาที “2022-12-07T10:41:18.803Z”	คืนค่า True เพื่อทำการแสดงผลหน้าประวัติการสนทนาต่อไป	ผ่าน
การแสดงผลหน้า Empty State เมื่อคำสั่งซื้อเกิน 24 ชั่วโมง	การแสดงผลหน้า Empty State เมื่อคำสั่งซื้อมีระยะเวลา 24 ชั่วโมง	1) เรียกฟังก์ชันที่ทำการเปรียบเทียบเวลาปัจจุบันและเวลาที่สร้างคำสั่งซื้อว่ามากหรือน้อยกว่า 24 ชั่วโมง	1) เวลาที่สร้างคำสั่งซื้อที่มีระยะเวลา 24 ชั่วโมง “2022-12-07T10:40:18.844Z”	คืนค่า True เพื่อทำการแสดงผลหน้าประวัติการสนทนาต่อไป	ผ่าน
การแสดงผลหน้า Empty State เมื่อคำสั่งซื้อเกิน 24 ชั่วโมง	การแสดงผลหน้า Empty State เมื่อคำสั่งซื้อมีระยะเวลา 24 ชั่วโมง 1 นาที	1) เรียกฟังก์ชันที่ทำการเปรียบเทียบเวลาปัจจุบันและเวลาที่สร้างคำสั่งซื้อว่ามากหรือน้อยกว่า 24 ชั่วโมง	1) เวลาที่สร้างคำสั่งซื้อที่มีระยะเวลา 24 ชั่วโมง 1 นาที “2022-12-07T10:39:18.845Z”	คืนค่า False เพื่อทำการแสดงผลหน้า Empty State ต่อไป	ผ่าน

4.2.2 การทดสอบ Adapter

การทดสอบการทำงานของชุดคำสั่ง ๆ ภายในคลาส Adapter ซึ่งเป็นคลาสสำคัญที่ทำหน้าที่เกี่ยวกับ View Logic โดยจะมีการทดสอบการทำงานในกรณีต่าง ๆ ดังต่อไปนี้

ตารางที่ 4.2 Test Case ของ Adapter

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การนำข้อมูลประวัติสนทนา มาแสดงผล	การนำข้อมูลประวัติสนทนา มาแสดงผล โดยข้อความที่นำมาแสดงผลถูกส่งในวันเดียวกัน	1) บรรจุข้อมูลข้อความที่ถูกส่งในวันเดียวกัน 2 ข้อความลงในอาเรย์ 2) เรียกใช้ฟังก์ชันที่อาเรย์ที่บรรจุข้อมูลข้อความไปแสดงผลข้อมูลประวัติการสนทนา	1) ข้อมูลข้อความที่ถูกส่งในวันเดียวกัน 2 ข้อความ	1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความทั้งสองพร้อมด้วยส่วนระบุวันที่ส่งข้อความ 1 ตัว ด้านบนข้อความทั้งสอง	ผ่าน
การนำข้อมูลประวัติสนทนา มาแสดงผล	การนำข้อมูลประวัติสนทนา มาแสดงผล โดยข้อความที่นำมาแสดงผลถูกส่งในคนละวัน	1) บรรจุข้อมูลข้อความที่ถูกส่งในวันแรก และข้อมูลข้อความที่ถูกส่งในวันถัดมาลงในอาเรย์ 2) เรียกใช้ฟังก์ชันที่อาเรย์ที่บรรจุข้อมูลข้อความไปแสดงผลข้อมูลประวัติการสนทนา	1) ข้อมูลข้อความที่ถูกส่งในวันแรก 2) ข้อมูลข้อความที่ถูกส่งในวันถัดมา	1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความทั้งสองพร้อมส่วนระบุวันที่ส่งข้อความ 1 ตัว ด้านบนข้อความที่ถูกส่งในวันแรก และส่วนระบุวันที่ส่งข้อความ 1 ซึ่งขึ้นกลางข้อความทั้งสอง	ผ่าน

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การนำข้อความที่ส่งจากลูกค้าผู้ให้บริการมาแสดงผล	การนำข้อความที่ส่งจากลูกค้าผู้ให้บริการมาแสดงผลในห้องสนทนา โดยข้อความที่นำมาแสดงผลถูกส่งในวันเดียวกัน	1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความจากลูกค้าผู้ให้บริการมาแสดงผลบนห้องสนทนา	1) ข้อมูลข้อความจากลูกค้าผู้ให้บริการที่ถูกส่งในวันเดียวกัน 2 ข้อความ	1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความทั้งสองจากลูกค้าผู้ให้บริการพร้อมด้วยส่วนระบุวันที่ส่งข้อความ 1 ตัวด้านบนข้อความทั้งสอง	ผ่าน
การนำข้อความที่ส่งจากลูกค้าผู้ให้บริการมาแสดงผล	การนำข้อความที่ส่งจากลูกค้าผู้ให้บริการมาแสดงผลในห้องสนทนา โดยข้อความที่นำมาแสดงผลถูกส่งในคนละวัน	1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความจากลูกค้าผู้ให้บริการมาแสดงผลบนห้องสนทนา	1) ข้อมูลข้อความที่ถูกส่งในวันแรก 2) ข้อมูลข้อความที่ถูกส่งในวันถัดมา	1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความทั้งสองจากลูกค้าผู้ให้บริการพร้อมส่วนระบุวันที่ส่งข้อความ 1 ตัวด้านบนข้อความที่ถูกส่งในวันแรก และส่วนระบุวันที่ส่งข้อความ 1 ซึ่งขึ้นกลางข้อความทั้งสอง	ผ่าน

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การนำข้อความที่ส่งจาก ผู้ประกอบการร้านค้ามา แสดงผล	การนำข้อความที่ส่งจาก ผู้ประกอบการร้านค้ามาแสดงผล ในห้องสนทนา โดยข้อความที่ นำมาแสดงผลถูกส่งในวันเดียวกัน	1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความ ที่ยังไม่ระบุเวลาการส่งที่สำเร็จจาก ผู้ประกอบการร้านค้าจำนวน 2 ข้อความ มาแสดงผลบนห้องสนทนา 2) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความ ที่ระบุเวลาการส่งที่สำเร็จแล้วจาก ผู้ประกอบการร้านค้าจำนวน 2 ข้อความ มาแสดงผลบนห้องสนทนา	1) ข้อมูลข้อความจาก ผู้ประกอบการร้านค้าที่ถูกส่ง ในวันเดียวกัน 2 ข้อความที่ยัง ไม่ระบุเวลาที่ส่งสำเร็จ 2) ข้อมูลข้อความเดิมจาก ผู้ประกอบการร้านค้าที่ถูกส่ง ในวันเดียวกัน 2 ข้อความแต่ ถูกระบุเวลาที่ทำการส่งที่ สำเร็จแล้ว	1) ข้อมูลประวัติสนทนาที่ ประกอบด้วยข้อความทั้งสองจาก ผู้ประกอบการร้านค้าพร้อมด้วยส่วน ระบุวันที่ส่งข้อความ 1 ตัวด้านบน ข้อความทั้งสอง	ผ่าน
การนำข้อความที่ส่งจาก ผู้ประกอบการร้านค้ามา แสดงผล	การนำข้อความที่ส่งจาก ผู้ประกอบการร้านค้ามาแสดงผล ในห้องสนทนา โดยข้อความที่ นำมาแสดงผลถูกส่งในคนละวัน	1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความ ที่ยังไม่ระบุเวลาการส่งที่สำเร็จจาก ผู้ประกอบการร้านค้าจำนวน 2 ข้อความ มาแสดงผลบนห้องสนทนา 2) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความ ที่ระบุเวลาการส่งที่สำเร็จแล้วจาก ผู้ประกอบการร้านค้าจำนวน 2 ข้อความ มาแสดงผลบนห้องสนทนา	1) ข้อมูลข้อความจาก ผู้ประกอบการร้านค้าที่ถูกส่ง ในคนละวัน 2 ข้อความที่ยังไม่ ระบุเวลาที่ส่งสำเร็จ 2) ข้อมูลข้อความเดิมจาก ผู้ประกอบการร้านค้าที่ถูกส่ง ในในคนละวัน 2 ข้อความแต่ ถูกระบุเวลาที่ทำการส่งที่ สำเร็จแล้ว	1) ข้อมูลประวัติสนทนาที่ ประกอบด้วยข้อความทั้งสองจาก ผู้ประกอบการร้านค้าพร้อมส่วนระบุ วันที่ส่งข้อความ 1 ตัวด้านบน ข้อความที่ถูกส่งในวันแรก และส่วน ระบุวันที่ส่งข้อความ 1 ซึ่งขึ้นกลาง ข้อความทั้งสอง	ผ่าน

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การระบุสถานะการอ่านของข้อความที่ส่งจากผู้ประกอบการร้านค้า	การอัปเดตสถานะการอ่านของข้อความที่ส่งจากผู้ประกอบการร้านค้า	<p>1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความที่ยังไม่ระบุเวลาการส่งที่สำเร็จจากผู้ประกอบการร้านค้าจำนวน 2 ข้อความ มาแสดงผลบนห้องสนทนา</p> <p>2) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความที่ระบุเวลาการส่งที่สำเร็จแล้วจากผู้ประกอบการร้านค้าจำนวน 2 ข้อความ มาแสดงผลบนห้องสนทนา</p> <p>3) เรียกใช้ฟังก์ชันที่ใช้ในอัปเดตสถานะการอ่านข้อความที่ส่งจากผู้ประกอบการร้านค้าทั้งหมด</p>	<p>1) ข้อมูลข้อความจากผู้ประกอบการร้านค้าที่ถูกส่งในคนละวัน 2 ข้อความที่ยังไม่ระบุเวลาที่ส่งสำเร็จ</p> <p>2) ข้อมูลข้อความเดิมจากผู้ประกอบการร้านค้าที่ถูกส่งในคนละวัน 2 ข้อความแต่ถูกระบุเวลาที่ทำการส่งที่สำเร็จแล้ว</p>	<p>1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความทั้งสองจากผู้ประกอบการร้านค้าที่ถูกอ่านแล้วพร้อมส่วนระบุวันที่ส่งข้อความ 1 ตัว ด้านบนข้อความที่ถูกส่งในวันแรก และส่วนระบุวันที่ส่งข้อความ 1 ซึ่งขึ้นกลางข้อความทั้งสอง</p>	ผ่าน

Test Scenario	Test Case Objective	Test Step	Test Data	Expected Result	Status
การแสดงผลปุ่มส่งข้อความใหม่อีกครั้ง	การแสดงผลปุ่มส่งใหม่อีกครั้งเมื่อข้อความของผู้ประกอบการร้านค้าส่งไม่สำเร็จ	<p>1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความที่ยังไม่ระบุเวลาการส่งที่สำเร็จจากผู้ประกอบการร้านค้า ข้อความมาแสดงผลบนห้องสนทนา</p> <p>2) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความจากผู้ประกอบการร้านค้าที่ส่งไม่สำเร็จเพื่อนำมาแสดงผลปุ่มส่งข้อความใหม่อีกครั้ง</p>	<p>1) ข้อความที่ยังไม่ระบุเวลาการส่งที่สำเร็จจากผู้ประกอบการร้านค้า</p> <p>2) ข้อมูลข้อความเดิมจากผู้ประกอบการร้านค้าที่ถูกระบุสถานะว่าข้อความดังกล่าวส่งไม่สำเร็จ</p>	1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความจากผู้ประกอบการร้านค้าที่ถูกระบุสถานะว่าข้อความดังกล่าวส่งไม่สำเร็จ	ผ่าน
การแสดงผลข้อความที่ถูกส่งใหม่อีกครั้ง	การแสดงผลข้อความที่ถูกส่งใหม่อีกครั้ง เมื่อข้อความที่ส่งไม่สำเร็จถูกดปุ่มส่งใหม่อีกครั้ง	<p>1) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความที่ยังไม่ระบุเวลาการส่งที่สำเร็จจากผู้ประกอบการร้านค้า ข้อความมาแสดงผลบนห้องสนทนา</p> <p>2) เรียกใช้ฟังก์ชันที่ใช้ในการรับข้อความจากผู้ประกอบการร้านค้าที่ส่งไม่สำเร็จเพื่อนำมาแสดงผลปุ่มส่งข้อความใหม่อีกครั้ง</p> <p>3) เรียกใช้ฟังก์ชันที่ใช้ส่งข้อความใหม่อีกครั้ง และลบข้อความเมื่อข้อความที่ส่งไม่สำเร็จ</p>	<p>1) ข้อความที่ยังไม่ระบุเวลาการส่งที่สำเร็จจากผู้ประกอบการร้านค้า</p> <p>2) ข้อมูลข้อความเดิมจากผู้ประกอบการร้านค้าที่ถูกระบุสถานะว่าข้อความดังกล่าวส่งไม่สำเร็จ</p>	1) ข้อมูลประวัติสนทนาที่ประกอบด้วยข้อความจากผู้ประกอบการร้านค้าที่ถูกส่งใหม่อีกครั้ง	ผ่าน

บทที่ 5

สรุปผลการพัฒนาและข้อเสนอแนะ

5.1 สรุปผลการพัฒนา

จากการที่ผู้จัดทำได้มีโอกาสร่วมโครงการสหกิจศึกษากับบริษัทแห่งหนึ่ง ผู้จัดทำได้ทำการพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิดิเชิน ซึ่งในระหว่างการพัฒนาผู้จัดทำได้ออกแบบและพัฒนาการทดสอบการทำงานของพีเจอาร์ดังกล่าวแบบอัตโนมัติ ช่วยให้ผู้จัดทำสามารถตรวจพบและแก้ไขจุดบกพร่องที่เกิดขึ้นในระหว่างการพัฒนาได้ อันเป็นผลทำให้พีเจอาร์การสนทนาสามารถทำงานได้อย่างสมบูรณ์ โดยผู้ประกอบการร้านค้าสามารถทำการสนทนากับลูกค้าผู้ใช้บริการที่เข้ามาสั่งซื้ออาหารได้ ผ่านการรับส่งข้อความอักษรและข้อความรูปภาพ ซึ่งจะช่วยให้ผู้ประกอบการร้านค้าสามารถส่งมอบอาหารหรือบริการอื่น ๆ ให้แก่ลูกค้าผู้ใช้บริการได้ดีมากยิ่งขึ้น โดยพีเจอาร์การสนทนาดังกล่าวนี้ได้ถูกเพิ่มให้แก่แอปพลิเคชันสำหรับผู้ประกอบการร้านค้าเป็นที่เรียบร้อยแล้ว ผู้ประกอบการร้านค้าสามารถใช้งานพีเจอาร์การสนทนาได้ทันทีเมื่ออัปเดตแอปพลิเคชันของทางแพลตฟอร์มคลาวด์คิดิเชินให้เป็นเวอร์ชันล่าสุด

ถึงแม้พีเจอาร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิดิเชินนั้นมีการทำงานที่เรียบง่ายและตอบสนองความต้องการพื้นฐานในการสนทนาได้ แต่ก็ยังมีข้อจำกัดในการทำงานอยู่ ทางผู้ประกอบการร้านค้าจะสามารถส่งข้อความอักษรและรูปภาพก็ข้อความก็ได้ ไม่มีการจำกัดจำนวนข้อความที่จะถูกส่ง แต่กลับมีข้อจำกัดด้านปริมาณของข้อมูลที่สามารถถูกส่งไปได้ในแต่ละข้อความ โดยข้อความตัวอักษรนั้นจะส่งได้ครั้งละ 128 ตัวอักษร และข้อความรูปภาพที่จะส่งได้ครั้งละ 1 รูปภาพเท่านั้น นอกจากนี้ยังขาดซึ่งฟังก์ชันการทำงานอื่น ๆ ที่จะช่วยอำนวยความสะดวกสบายในใช้งานอีกด้วย

5.2 ข้อเสนอแนะ

ในการพัฒนาพีเจอาร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิดิเชิน ทางผู้จัดทำได้พบข้อสังเกตทำในระหว่างการพัฒนาจึงเกิดข้อเสนอแนะที่จะนำไปปรับปรุงและพัฒนาต่อไปในอนาคต

1) เนื่องจากผู้จัดทำยังขาดการประสบการณ์และความรู้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ ส่งผลให้พีเจอาร์การสนทนาถูกพัฒนาด้วยชุดคำสั่งและรูปแบบการทำงานพื้นฐานทั่วไป ทำให้ยังมีช่องว่างในการพัฒนาซึ่งสามารถปรับปรุงประสิทธิภาพการทำงานด้วยชุดคำสั่งขั้นสูงและรูปแบบการทำงานที่เหมาะสมกับการพัฒนาพีเจอาร์การสนทนามากกว่าและให้ประสิทธิภาพการทำงานที่ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เนื่องจากผู้จัดทำจัดยังขาดการประสบการณ์และความรู้ในการทำการทดสอบแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ ส่งผลให้ไม่มีการทำการทดสอบอัตโนมัติแบบ Unit Test กับส่วนการทำงานต่าง ๆ ในระหว่างที่ทำการพัฒนาพีเจอาร์การสนทนา ท้ายที่สุดเมื่อการพัฒนาพีเจอาร์การสนทนาใกล้สิ้นสุดลงแล้วยังคงเหลือจุดบกพร่องจำนวนมาก ทำให้ผู้จัดทำต้องรีบทำการแก้ไขจุดบกพร่องเหล่านั้นด้วยความเร่งรีบ ซึ่งการทดสอบอัตโนมัติแบบ Unit Test เป็นระยะ ๆ ระหว่างการพัฒนาจะช่วยให้ นักพัฒนาพบจุดบกพร่องได้แต่เนิ่น ๆ และทำการแก้ไขได้ก่อนที่จุดบกพร่องนั้นจะก่อความยุ่งยากในการพัฒนาซอฟต์แวร์

3) เนื่องจากรูปแบบการสนทนาของผู้ประกอบการร้านค้าที่ทำการสนทนากับลูกค้า ผู้ใช้บริการที่เข้ามาสั่งซื้ออาหารนั้นมักจะเป็นการตอบคำถามเกี่ยวกับอาหารหรือผลิตภัณฑ์ ซึ่งอาจเป็นบทสนทนาหรือข้อความที่ทางผู้ประกอบการร้านค้าต้องตอบลูกค้าผู้ให้บริการอยู่บ่อย ๆ และอาจเป็นข้อความที่จะมีความยาว ไม่สะดวกที่จะถูกพิมพ์ซ้ำใหม่บ่อย ๆ หากภายในห้องสนทนาผู้ประกอบการร้านค้าสามารถกำหนดของชุดของข้อความที่มักต้องตอบลูกค้าบ่อย ๆ เตรียมไว้ให้สามารถกดส่งข้อความดังกล่าวได้ทันทีโดยไม่ต้องพิมพ์ใหม่ จะช่วยให้ผู้ประกอบการร้านค้าประหยัดเวลาและสะดวกต่อการใช้งานมากยิ่งขึ้น

4) เนื่องจากการสนทนาของผู้ประกอบการร้านค้าที่ทำการสนทนากับลูกค้าผู้ให้บริการที่เข้ามาสั่งซื้ออาหารนั้นอาจจะต้องมีการส่งรูปภาพมากกว่าหนึ่งรูปภาพ ซึ่งถึงแม้พีเจอาร์การสนทนาจะไม่จำกัดจำนวนในการข้อความรูปภาพ แต่ก็สามารถส่งได้ครั้งละ 1 รูปภาพเท่านั้น หากผู้ประกอบการร้านค้าต้องการส่งรูปภาพจำนวนหนึ่ง จะต้องทำการเลือกส่งรูปทีละภาพ และทำเช่นนี้ไปจนครบทุกรูปภาพ ซึ่งอาจจะเป็นการยุ่งยาก หากสามารถเลือกรูปที่จะส่งในแต่ละครั้งได้หลาย ๆ รูปภาพ จะช่วยให้ผู้ประกอบการร้านค้าประหยัดเวลาและสะดวกต่อการใช้งานมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] สมเกียรติ กิจวงศ์วัฒน์. มาเขียน Unit Test บน Android Studio กัน. [Online]. แหล่งที่มา : <https://akexorcist.dev/unit-test-in-android-studio/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [2] สมเกียรติ กิจวงศ์วัฒน์. เรื่องที่ควรรู้เกี่ยวกับการเขียนเทสบนแอนดรอยด์. [Online]. แหล่งที่มา : <https://akexorcist.dev/all-about-write-a-test-in-android/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [3] สมเกียรติ กิจวงศ์วัฒน์. ทำไม App Architecture บนแอนดรอยด์ในยุคนี้ถึงนิยมใช้ Architecture Components?. [Online]. แหล่งที่มา : <https://akexorcist.dev/why-architecture-components-is-recommended-app-architecture-for-android/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [4] Git version control UI SourceTree. [Online]. แหล่งที่มา : <https://onlymycode.blogspot.com/2017/05/git-version-control-ui-sourcetree.html/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [5] Jedsada Saengow. [Firebase] คืออะไร มาดูวิธีสร้าง Project และทำความเข้าใจกับ Firebase. [Online]. แหล่งที่มา : <https://medium.com/jed-ng/firebase-คืออะไร-มาดูวิธีสร้าง-project-และทำความเข้าใจกับ-firebase-d48bfac67b14>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [6] Jirawatee. รู้จัก Firebase Cloud Messaging (FCM) ตั้งแต่ Zero จนเป็น Hero. [Online]. แหล่งที่มา : <https://medium.com/firebasethailand/รู้จัก-firebase-cloud-messaging-fcm-ตั้งแต่-zero-จนเป็น-hero-fb7900af92cd>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [7] Kotlin vs Java. [Online]. แหล่งที่มา : <https://mediaan.com/mediaan-blog/kotlin-vs-java/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [8] Nutti Saelor. MVC MVP MVVM คืออะไร และต่างกันอย่างไร. [Online]. แหล่งที่มา : <https://medium.com/@leelorz6/mvc-mvp-mvvm-คืออะไรและแตกต่างกันอย่างไร-ca16a19631dc/>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [9] Patchara Boonmathanaruk. Figma คืออะไร? ทำไมถึงเป็น Tool มาแรงที่สุดแห่งปี. [Online]. แหล่งที่มา : <https://blog.skooldio.com/figma-ui-design-tool>. สืบค้นเมื่อ 5 ตุลาคม 2565
- [10] Woranat Prathummanee. ทำไมภาษา Kotlin ถึงเป็นภาษาหลักในการพัฒนา Android? และ Kotlin จะมาแทนที่ Java หรือไม่?. [Online]. แหล่งที่มา : <https://blog.skooldio.com/kotlin-vs-java-android/>. สืบค้นเมื่อ 5 ตุลาคม 2565

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การติดตั้งโปรแกรม Android Studio

ก.1 ดาวน์โหลดและติดตั้งโปรแกรม Android Studio

ก.1.1 ขั้นตอนการดาวน์โหลดและติดตั้งโปรแกรม JetBrains Toolbox App

- 1) ในปัจจุบันนี้ Android Studio สามารถถูกดาวน์โหลดและติดตั้งผ่านโปรแกรม JetBrains Toolbox ได้ ซึ่งการติดตั้ง Android Studio นั้นมีข้อดีคือสามารถให้ตัวโปรแกรมทำการอัปเดตเป็นแบบอัตโนมัติได้ โดยเริ่มต้นด้วยการไปยังเว็บไซต์อย่างเป็นทางการสำหรับการดาวน์โหลด JetBrains Toolbox App <https://www.jetbrains.com/toolbox-app/> ดังรูปที่ ก.1

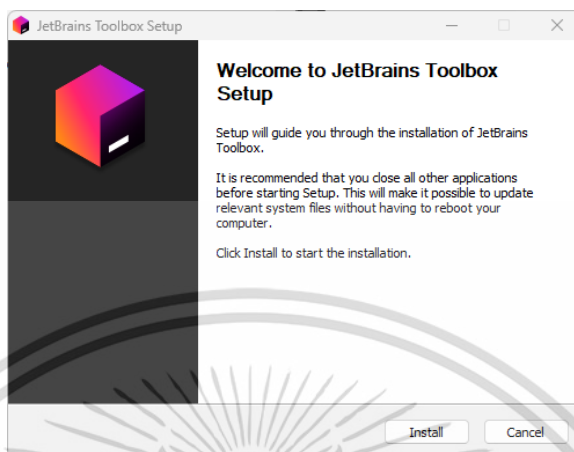


รูปที่ ก.1 หน้าเว็บไซต์สำหรับการดาวน์โหลด JetBrains Toolbox

- 2) กวดาวน์โหลดโปรแกรม JetBrains Toolbox

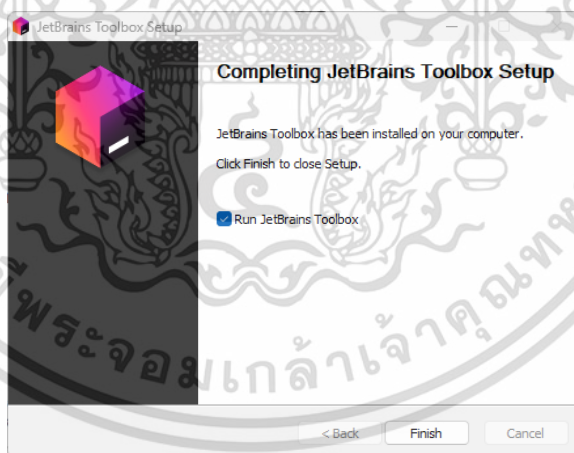
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 3) เมื่อเปิดไฟล์โปรแกรมตัวติดตั้งโปรแกรมขึ้นมา จะแสดงหน้าต่างให้กด Install เพื่อทำการติดตั้งต่อไป ดังในรูป ก.2



รูปที่ ก.2 หน้าต่างเริ่มต้นการติดตั้งโปรแกรม JetBrains Toolbox

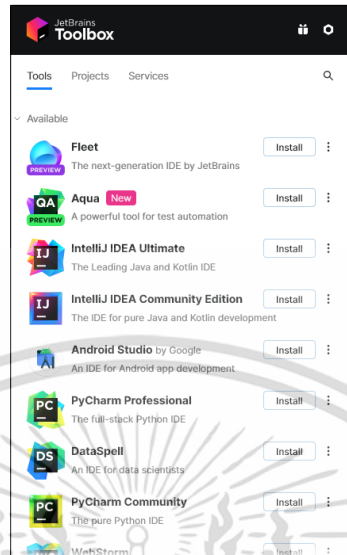
- 4) เมื่อติดตั้งสำเร็จจะปรากฏปุ่ม Finish เพื่อปิดหน้าต่างการติดตั้งและเริ่มต้นการทำงานของโปรแกรม JetBrains Toolbox ดังในรูปที่ ก.3



รูปที่ ก.3 หน้าต่างเมื่อการติดตั้งโปรแกรม JetBrains Toolbox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

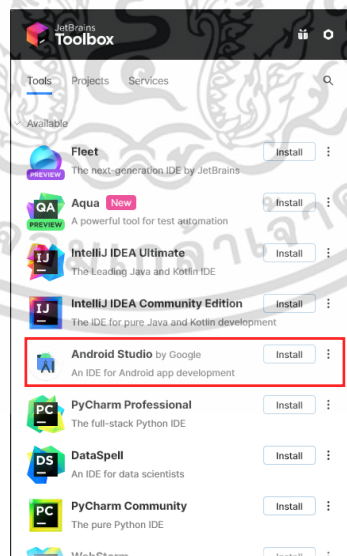
5) เมื่อโปรแกรมต้นทำงานจะปรากฏเป็นหน้าต่างดังรูปที่ ก.4



รูปที่ ก.4 หน้าต่างโปรแกรม JetBrains Toolbox

ก.1.2 ขั้นตอนการดาวน์โหลดและติดตั้งโปรแกรม Android Studio

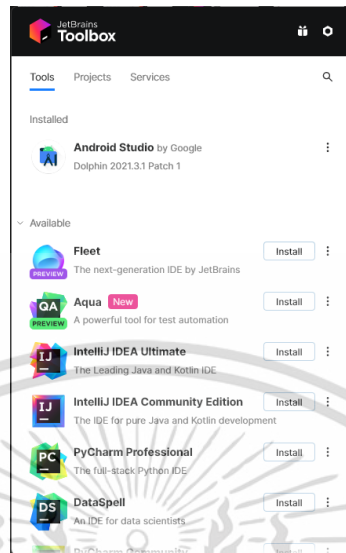
1) เมื่อโปรแกรมต้นทำงาน JetBrains Toolbox จะปรากฏเป็นรายการของโปรแกรมสำหรับพัฒนาโปรแกรมต่าง ๆ หนึ่งในนั้นคือ Android Studio



รูปที่ ก.5 Android Studio ที่ยังไม่ถูกติดตั้งในหน้าต่างโปรแกรม JetBrains Toolbox

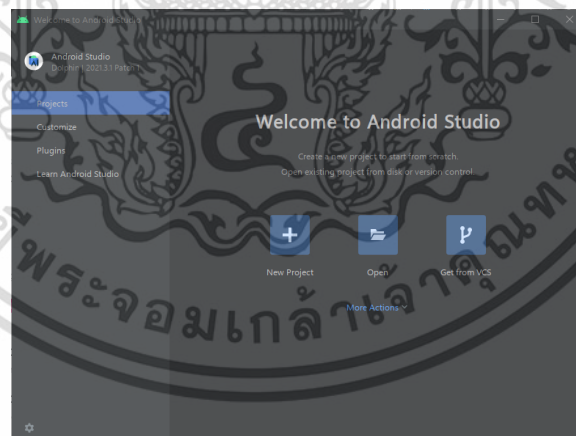
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ให้ทำการกดปุ่ม Install เพื่อเริ่มต้นการติดตั้ง Android Studio



รูปที่ ก.6 Android Studio ที่ถูกติดตั้งแล้วในหน้าต่างโปรแกรม JetBrains Toolbox

3) เมื่อการติดตั้งเสร็จสิ้นจึงจะสามารถเปิดโปรแกรม Android Studio ได้
ดังที่ปรากฏเป็นหน้าต่างเริ่มการทำงานของ Android Studio ในรูปที่ ก.7



รูปที่ ก.7 หน้าต่างการเริ่มการทำงานของ Android Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



งานทะเบียนคณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

คำรับรองเล่มโครงการพิเศษ/ปัญหาพิเศษ/สหกิจศึกษา

วันที่ 17 เดือน กุมภาพันธ์ พ.ศ. 2566

ข้าพเจ้า นายสรานุกฤษ จงจิตตาทิบาล รหัสประจำตัว 62050234 นักศึกษาหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์ ขอรับรองว่าสหกิจศึกษา เรื่อง ชื่อภาษาไทย การพัฒนาฟีเจอร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มคลาวด์คิทเชน ชื่อภาษาอังกฤษ The Development Of Chat Features On Mobile Application For Merchant Partners Of Cloud Kitchen Platform ปีการศึกษา 2565 เป็นผลงานวิจัยที่มีได้คัดลอกหรือละเมิดลิขสิทธิ์ของผู้อื่นและได้ผ่านการตรวจสอบความซ้ำซ้อนเรียบร้อยแล้ว และได้แนบเอกสารการตรวจสอบการลอกเลียนงานวรรณกรรมที่ตรวจสอบจากเล่มสหกิจศึกษานับสมบูรณ์แล้ว โปรแกรมอักขรวิสุทธิ 0.96%

ลงชื่อ สรานุกฤษ จงจิตตาทิบาล

(นายสรานุกฤษ จงจิตตาทิบาล)

นักศึกษา

ข้าพเจ้า ผศ.ดร.อินทรภาพ อรัณยณะนาค อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบสหกิจศึกษาของนักศึกษาข้างต้น แล้ว ขอรับรองว่าเป็นผลงานวิจัยของนักศึกษาจริงและมีเนื้อหาสมบูรณ์จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ อินทรภาพ อรัณยณะนาค

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ 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 การพัฒนาพีเจอร์การสนทนาบนแอปพลิเคชันสำหรับผู้ประกอบการร้านค้าที่เข้าร่วมแพลตฟอร์มตลาดดิจิทัลของนักศึกษาชื่อ นายสรายุทธ วจิตตาทิบาล ซึ่งเป็นนักศึกษาภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เรียบร้อยแล้ว และไม่มีส่วนหนึ่งส่วนใดในเล่มสหกิจศึกษานี้ที่มีข้อมูลอ่อนไหว และ/หรือ ข้อมูลอันเป็นความลับอันจะก่อให้เกิดความเสียหายต่อสถานประกอบการ รวมทั้งอนุญาตให้สามารถเผยแพร่ต่อสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังได้ จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ **ทรงวุฒิ ชัยประเสริฐศรี**

(ทรงวุฒิ ชัยประเสริฐศรี)

ตัวแทนสถานประกอบการ

ข้าพเจ้า ผศ.ดร.อินทราพร อรัณยะนาค อาจารย์ที่ปรึกษาสหกิจศึกษา ได้ตรวจสอบเล่มสหกิจศึกษาแล้ว และรับทราบว่าสถานประกอบการดำเนินการตรวจสอบเล่มสหกิจศึกษาแล้ว จึงลงชื่อไว้เป็นหลักฐาน

ลงชื่อ **อินทราพร อรัณยะนาค**

(อินทราพร อรัณยะนาค)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้