ระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้

Online Social System for Short-Distance Shopping Services

นาย วรายุส สายสี 6304800003

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยสยาม

ปีการศึกษา 2566

หัวข้อปริญญานิพนธ์	ระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ Online Social System for Short-Distance Shopping Services
หน่วยกิตของปริญญานิพนธ์	3 หน่วยกิต
รายชื่อผู้จัดทำ	นายวรายุส สายสี 6304800003
อาจารย์ที่ปรึกษา	อาจารย์จรรยา แหยมเจริญ
ระดับการศึกษา	ปริญญาตรี
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2566

อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
(พล.อ.ท.ผศ.ดร. พาห์รณ สงวนโภคัย)

.....กรรมการ
(อาจารย์ธนาภรณ์ รอดชีวิต)

.....อาจารย์ที่ปรึกษา
(อาจารย์จรรยา แหยมเจริญ)

หัวข้อปริญญานิพนธ์	ระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้
หน่วยกิตของปริญญานิพนธ์	3 หน่วยกิต
รายชื่อผู้จัดทำ	นายวรายุส สายสี 6304800003
อาจารย์ที่ปรึกษา	อาจารย์จรรยา แหยมเจริญ
ระดับการศึกษา	ปริญญาตรี
ภาควิชา	วิทยาการคอมพิวเตอร์
ปีการศึกษา	2566

บทคัดย่อ

การพัฒนาระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ มีวัตถุประสงค์เพื่อสร้างแพลตฟอร์มสำหรับผู้ใช้ที่ต้องการความสะดวกในการซื้อสินค้าหรือบริการโดยไม่ต้องพึ่งพาแพลตฟอร์มร้านค้าออนไลน์ ที่มีค่าบริการค่อนข้างแพง และเป็นช่องทางในการเพิ่มยอดขายให้กับร้านทั่วไปและร้านค้าในชุมชนที่ไม่ได้เข้าร่วมกับแพลตฟอร์มร้านค้าออนไลน์ในปัจจุบัน โดยระบบไม่มีการคิดค่าใช้บริการเพิ่มเติมจากร้านค้า ผู้ใช้ของระบบสามารถโพสต์ฝากซื้อสินค้าผ่านและสามารถรับฝากซื้อสินค้าตามที่มีผู้โพสต์ได้โดยจะได้รับค่าจ้างเป็นรายได้เสริม ระบบนี้ช่วยให้ผู้ใช้สามารถประหยัดเวลาและเพิ่มความสะดวกในการซื้อสินค้า โดยพัฒนาเป็นโมบายแอปพลิเคชัน พัฒนาด้วย Flutter สำหรับสร้าง Native Interface บนระบบปฏิบัติการแอนดรอยด์ ข้อมูลของระบบจัดเก็บด้วย Firebase

คำสำคัญ: สื่อสังคมออนไลน์, การบริการฝากซื้อสินค้าหรือบริการ, โมบายแอปพลิเคชัน

Project Title : Online Social System for Short-Distance Shopping Services

Credits : 3 Units

By : Mr. Warayut Saisi 6304800003

Advisor : Miss Janya Yamcharoen

Degree : Bachelor of Science

Major : Computer Science

Faculty : Science

Semester/Academic year: 3/2023

Abstract

The development of the Online Social System for Short-Distance Shopping Services aims to create a platform for users seeking convenience in purchasing goods or services without relying on expensive online shopping platforms. This system serves as a channel to increase sales for general and community stores that are not currently part of online shopping platforms. The system does not charge additional service fees from the stores. Users can post purchase requests and accept tasks posted by others, earning supplementary income. This system enhances user convenience and saves time by facilitating purchases. The platform has developed as a mobile application using Flutter to create a native interface for the Android operating system. The system's data is stored using Firebase.

Keywords: social media online, shopping procurement services, mobile application

.....

(Project Advisor)

Approved by

.....

กิตติกรรมประกาศ

(Acknowledgment)

การจัดทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี ผู้จัดทำได้รับความกรุณาจากอาจารย์ผู้สอนทุกท่านที่ได้รับความกรุณาชี้แนะและช่วยเหลือในทุกๆ ด้าน เพื่อจัดทำปริญญานิพนธ์นี้ ส่งผลให้ผู้จัดทำได้รับความรู้และประสบการณ์ต่างๆ ที่มีค่ามากมาย สำหรับปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนดังนี้

1. อาจารย์ จรรยา แหยมเจริญ อาจารย์ที่ปรึกษา

ผู้จัดทำขอขอบคุณคณะกรรมการสอบปริญญานิพนธ์ ที่ได้ให้คำแนะนำสำคัญในการสอบปริญญานิพนธ์ฉบับนี้ และผู้ที่มีส่วนร่วมทุกท่านรวมทั้งบุคคลอื่นๆ ที่ไม่ได้กล่าวนามไว้ ที่มีส่วนร่วมในการให้ข้อมูล และให้ความช่วยเหลือในด้านต่างๆ และเป็นที่ปรึกษาให้คำแนะนำต่างๆ จนทำให้งานทุกอย่างประสบความสำเร็จไปได้ด้วยดี และจัดทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ซึ่งผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

ผู้จัดทำ

นายวรายุส สายสี

สารบัญ

หน้า

บทคัดย่อ..... ก

Abstract..... ข

กิตติกรรมประกาศ..... ค

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์..... 2

1.3 ขอบเขตปริญญานิพนธ์..... 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ..... 3

1.5 ขั้นตอนและวิธีการดำเนินงานปริญญานิพนธ์..... 3

1.6 แผนและระยะเวลาในการดำเนินงานปริญญานิพนธ์..... 5

1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนา..... 5

1.8 อุปกรณ์และเครื่องมือที่รองรับระบบ..... 6

บทที่ 2 การทบทวนวรรณกรรมที่เกี่ยวข้อง

2.1 ภาษา Dart……..... 7

2.2 Flutter Framework…….……... 8

2.3 Android Studio……...…… 9

2.4 Firebase..... 10

บทที่ 3 การวิเคราะห์ระบบ

3.1 รายละเอียดของปริญญานิพนธ์..... 12

3.2 แสดงฟังก์ชันการทำงานของระบบด้วย Use Case Diagram..... 13

3.3 คำอธิบายรายละเอียดของยูสเคส (Use Case Description)..... 14

3.4 แสดงขั้นตอนการทำงานของฟังก์ชันด้วย Sequence Diagram..... 20

3.5 แผนภาพแสดงองค์ประกอบของคลาส (Class Diagram)..... 24

3.6 รายละเอียดของคลาส..... 25

3.7 โครงสร้างของข้อมูล (Data Structure)..... 30

3.8 รายละเอียดของโครงสร้างข้อมูล (Data Dictionary)……..... 31

บทที่ 4 การออกแบบทางกายภาพ

4.1 โครงสร้างของแอปพลิเคชัน (Application Structure Map)..... 33

4.3 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)..... 36

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลและข้อเสนอแนะ

5.1 สรุปผลปริญญานิพนธ์..... 49

5.2 ข้อดีของระบบ..... 49

5.3 ข้อเสนอแนะ..... 49

บรรณานุกรม..... 51

สารบัญตาราง

หน้า

ตารางที่ 1.1 ระยะเวลาในการดำเนินงานปริญญานิพนธ์..... 5
ตารางที่ 3.1 แสดงรายละเอียดของ Use case Login..... 14
ตารางที่ 3.2 แสดงรายละเอียดของ Use case Register..... 15
ตารางที่ 3.3 แสดงรายละเอียดของ Use case EditProfile..... 16
ตารางที่ 3.4 แสดงรายละเอียดของ Use case CreatePost…..... 17
ตารางที่ 3.5 แสดงรายละเอียดของ Use case DepositCoin..... 18
ตารางที่ 3.6 แสดงรายละเอียดของ Use case WithdrawCoin..... 18
ตารางที่ 3.7 แสดงรายละเอียดของ Use case Chat……..... 19
ตารางที่ 3.8 รายละเอียดของ Class Diagram Detail : Profile…..... 25
ตารางที่ 3.9 รายละเอียดของ Class Diagram Detail : Register…..... 25
ตารางที่ 3.10 รายละเอียดของ Class Diagram Detail : Login…….…..... 25
ตารางที่ 3.11 รายละเอียดของ Class Diagram Detail : Feed…….….... 26
ตารางที่ 3.12 รายละเอียดของ Class Diagram Detail : PostList……..... 26
ตารางที่ 3.13 รายละเอียดของ Class Diagram Detail : CreatePost…..... 27
ตารางที่ 3.14 รายละเอียดของ Class Diagram Detail : PostDetail……..... 28
ตารางที่ 3.15 รายละเอียดของ Class Diagram Detail : TakeDeal…….…..... 28
ตารางที่ 3.16 รายละเอียดของ Class Diagram Detail : Chat…..….…..... 28
ตารางที่ 3.17 รายละเอียดของ Class Diagram Detail : PushNotificationService.....,..... 29
ตารางที่ 3.18 รายละเอียดของโครงสร้างข้อมูล : Posts……..... 31
ตารางที่ 3.19 รายละเอียดของโครงสร้างข้อมูล : Users……..... 32
ตารางที่ 3.20 รายละเอียดของโครงสร้างข้อมูล : Chat Rooms..... 32
ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Deal..... 34

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 ตัวอย่างการสร้างคลาสด้วยถายาดาร์ต..... 7
รูปที่ 2.2 ตัวอย่างแอปพลิเคชันที่พัฒนาด้วยฟลัตเตอร์..... 8
รูปที่ 2.3 โปรแกรม Android Studio..... 9
รูปที่ 2.4 หน้าจอ Firebase แสดงการวิเคราะห์แอปพลิเคชัน..... 11
รูปที่ 3.1 Use Case Diagram ของระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้…..... 13
รูปที่ 3.2 Sequence diagram : Login..... 20
รูปที่ 3.3 Sequence diagram : Register..... 20
รูปที่ 3.4 Sequence diagram : Edit Profile..... 21
รูปที่ 3.5 Sequence diagram : Create Post……..... 21
รูปที่ 3.6 Sequence diagram : Deposit Coin……..... 22
รูปที่ 3.7 Sequence diagram : Withdraw Coin …..... 22
รูปที่ 3.8 Sequence diagram : Chat……..... 23
รูปที่ 3.9 Class Diagram ของระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ 24
รูปที่ 3.10 โครงสร้างของข้อมูล (Data Structure)..... 30
รูปที่ 4.1 โครงสร้างของแอปพลิเคชัน Deal..... 33
รูปที่ 4.2 ไอคอนของแอปพลิเคชัน Deal..... 33
รูปที่ 4.3 หน้าเข้าสู่ระบบ (Login)..... 33
รูปที่ 4.4 หน้าลงทะเบียนเข้าใช้งานแอปพลิเคชัน Deal (Register..... 37
รูปที่ 4.5 หน้า Feed Post..... 38
รูปที่ 4.6 หน้า Create Post..... 39
รูปที่ 4.7 หน้า Post Detail..... 40
รูปที่ 4.8 หน้า MyDeal…..... 41
รูปที่ 4.9 หน้า Post Detail หลังจากมีผู้ตอบตกลงรับฝากซื้อสินค้า……..... 42
รูปที่ 4.10 หน้า Chat..... 43
รูปที่ 4.11 หน้า Wallet..... 44
รูปที่ 4.12 หน้า Deposit Coin..... 45
รูปที่ 4.13 หน้า Withdraw Coin..... 46
รูปที่ 4.14 หน้า Profile……..... 47
รูปที่ 4.15 หน้า Eitdit Profile……..... 48

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

แรงจูงใจในการพัฒนาระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ เนื่องจากถ้าเราต้องการซื้อสินค้าหรือบริการที่อยู่ใกล้เคียง เราจะต้องเดินทางไปซื้อด้วยตนเอง แต่ถ้าไม่มีเวลาว่างที่จะเดินทางไปซื้อสินค้าด้วยตนเอง อาจจะต้องพึ่งพาบริการจัดส่งสินค้า อย่างไรก็ตามบริการดังกล่าวอาจไม่สามารถใช้งานได้เนื่องจากร้านค้าที่ต้องการซื้อไม่ได้อยู่ในแอปพลิเคชันบริการจัดส่งสินค้านั้น ประกอบกับในปัจจุบันประชาชนส่วนใหญ่มีโทรศัพท์มือถือ (Mobile Phone) เป็นอุปกรณ์ประจำตัวสำหรับใช้ในการดำเนินชีวิตประจำวัน อาทิเช่น ใช้เพื่อการสื่อสาร ทำธุรกรรมทางการเงิน ติดตามข่าวสาร เข้าสู่สังคมออนไลน์ และบริการอื่นๆ ดังนั้นผู้จัดทำจึงได้มีแนวคิดพัฒนาระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ ที่ใช้งานผ่านแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ โดยสามารถฝากซื้อสินค้าจากร้านค้าที่อยู่ในระยะรัศมี 3 กิโลเมตร กับผู้ใช้ของระบบคนอื่น นั่นหมายความว่าผู้ใช้ของระบบเป็นได้ทั้งผู้ฝากซื้อและผู้รับฝากซื้อ โดยผู้รับฝากซื้อจะได้รับค่าจ้างเป็นค่าตอบแทน

นอกจากนี้เพื่อเป็นการสนับสนุนร้านค้าขนาดเล็ก ร้านค้าชุมชน ให้มียอดขายเพิ่มขึ้นโดยไม่ถูกหักค่านายหน้าจากแพลตฟอร์มร้านค้าออนไลน์ ซึ่งจำนวนร้านค้าที่ไม่ได้พึงพาแพลตฟอร์มร้านค้าออนไลน์มีอีกเป็นจำนวนมาก และยังเป็นช่องทางในการสร้างรายได้เสริมให้กับคนทั่วไปแค่รับฝากซื้อของตามที่มีผู้ใช้ประกาศว่าจ้างก็จะได้ค่าจ้างทันที

ระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ พัฒนาเป็นโมบายแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ พัฒนาด้วย Flutter และภาษา Dart จัดเก็บข้อมูลไว้บน Firebase กลุ่มผู้ใช้ที่ลงทะเบียนบนแอปพลิเคชันจะเป็นได้ทั้งผู้ฝากซื้อสินค้า และผู้รับจ้างซื้อสินค้า โดยฟังก์ชันการทำงานประกอบด้วย 6 ฟังก์ชันหลัก ประกอบด้วย 1) ฟังก์ชันการลงทะเบียนเข้าใช้งานและการยืนยันตัวตน ฟังก์ชันการ 2) โพสต์แจ้งหาผู้รับฝากซื้อสินค้าที่ต้องการ 3) ฟังก์ชันสนทนาระหว่างผู้ฝากซื้อสินค้าและผู้รับฝากซื้อสินค้า (Chat) 4) ฟังก์ชันการคำนวณระยะทาง 5) ฟังก์ชันการแจ้งเตือน (Notification) และ 6) ฟังก์ชนการชำระเงินผ่านกระเป๋าเงินอิเล็กทรอนิกส์ (E-Wallet)

1.2 วัตถุประสงค์ของปริญญานิพนธ์

เพื่อพัฒนาระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ สำหรับเป็นอีกหนึ่งช่องทางในการเพิ่มยอดขายให้กับร้านค้า และสร้างรายได้เสริมให้กับบุคคลทั่วไป

1.3 ขอบเขตของปริญญานิพนธ์

ฟังก์ชันการทำงานของระบบ ประกอบด้วย

1.3.1.1 ลงทะเบียนเพื่อใช้งานระบบ โดยใช้บริการจาก Firebase Authentication

1.3.1.2 การยืนยันตัวตนเพื่อเข้าใช้งานระบบ (Authentication) ผ่านอีเมลที่ได้ให้ไว้ในขั้นตอนการลงทะเบียนบนแอปพลิเคชัน

1.3.1.3 การแก้ไขรายละเอียดของผู้ใช้

1.3.1.3.1 ผู้ใช้สามารถแก้ไขรูปโปรไฟล์ได้

1.3.1.3.2 ผู้ใช้สามารถแก้ไขข้อมูลประวัติส่วนตัวและเพศได้

1.3.1.4 การสร้างโพสต์สำหรับฝากซื้อสินค้าและรับซื้อสินค้า

1.3.1.4.1 สามารถเลือกที่จะเป็นผู้ฝากซื้อสินค้า หรือ เป็นผู้รับฝากซื้อสินค้าได้

1.3.1.4.2 สามารถโพสต์ฝากซื้อสินค้าได้ โดยกรอกข้อมูลรายละเอียดสินค้าและร้านค้าที่ต้องการได้

1.3.1.4.3 สามารถเพิ่มรูปภาพที่เกี่ยวข้องได้

1.3.1.4.4 สามารถตอบรับโพสต์ที่ต้องการหาผู้รับฝากซื้อสินค้าได้

1.3.1.5 การส่งข้อความ (Chat)

1.3.2.3.1 โพสต์ฝากซื้อสินค้าที่ได้รับตอบรับแล้ว ผู้ใช้ที่เป็นผู้ฝากซื้อและผู้ใช้ที่เป็นผู้รับฝากซื้อสินค้าสามารถสนทนาผ่านแอปพลิเคชันได้แบบเรียลไทม์

1.3.2.3.2 ผู้ใช้สามารถดูสถานะของโพสต์ได้ ว่าถูกอ่านหรือยัง

1.3.1.6 ระบบการคำนวณระยะทาง

1.3.1.6.1 ระบบจะคำนวณหาระยะทางระหว่างผู้ใช้ในระบบโดยผู้ใช้ที่อยู่ในระยะรัศมี 3 กิโลเมตร จะได้รับการแจ้งเตือนถ้ามีโพสต์ต้องการฝากซื้อสินค้า โดยโพสต์ที่ห่างกันน้อยที่สุดจะถูกแสดงขึ้นมาก่อน

1.3.1.7 การแจ้งเตือน (Notification)

1.3.1.7.1 ผู้ใช้จะได้รับการแจ้งเตือนเมื่อสถานะของโพสต์มีการเปลี่ยนแปลง

1.3.1.7.2 ผู้ใช้จะได้รับการแจ้งเตือนเมื่อมีข้อความส่งถึงผู้ใช้

1.3.1.7.3 ผู้ใช้จะได้รับการแจ้งเตือนเมื่อมีผู้ใช้ที่โพสต์ฝากซื้อสินค้าและอยู่ในระยะที่ใกล้กัน

1.3.1.6.4 ผู้ใช้จะได้รับการแจ้งเตือนเมื่อมีโพสต์ใหม่เกิดขึ้นในระยะใกล้กับผู้ใช้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 ช่วยลดเวลาที่ต้องใช้ในการเดินทางไปซื้อสินค้า ทำให้ผู้ใช้งานมีเวลามากขึ้นสำหรับทำกิจกรรมอื่นๆ

1.4.2 เป็นช่องทางสร้างรายได้ให้แก่ผู้ใช้งานระบบ

1.4.3 เพิ่มช่องทางในการขายสินค้าและบริการให้กับร้านค้าทั่วไปและร้านค้าในชุมชน โดยไม่ต้องเสียค่านายหน้า

1.4.4 ผู้บริโภคซื้อสินค้าในราคาที่เป็นธรรม ไม่มีการบวกเพิ่มค่าบริการต่างๆ

1.5 ขั้นตอนและวิธีการดำเนินงานปริญญานิพนธ์

1.5.1 การรวบรวมความต้องการและการศึกษาข้อมูล (Detailed Study and Requirement Collection)

รวบรวมจากความต้องการของผู้จัดทำ และทำการสัมภาษณ์กลุ่มผู้ใช้งานที่มีความต้องการคล้ายๆ กัน รวมถึงกลุ่มผู้ใช้กลุ่มอื่นๆ เพื่อเก็บข้อมูลเกี่ยวกับความต้องการในการฝากซื้อสินค้าและบริการ รวมถึงปัญหาที่พบจากการเดินทางไปซื้อสินค้าด้วยตนเอง ศึกษาแอปพลิเคชันบริการจัดส่งสินค้าที่มีอยู่ในท้องตลาดเพื่อศึกษาแนวทางการทำงานและฟังก์ชันการทำงานต่าง ๆ ที่เป็นที่นิยม รวมถึงศึกษาระบบใกล้เคียงและรวบรวมข้อมูล เพื่อนำไปทำการวิเคราะห์ต่อไป

1.5.2 การวิเคราะห์ระบบ (System Analysis)

นำข้อมูลและความต้องการที่ได้รวบรวมมาทำการวิเคราะห์ระบบงานใหม่และวางแผนการปฏิบัติงาน เพื่อกำหนดขอบเขตของระบบและฟังก์ชันการทำงาน โดยนำเสนอผ่านแผนภาพไดอะแกรมต่างๆ ได้แก่ Use Case Diagram แสดงภาพรวมฟังก์ชันการทำงานทั้งหมด Sequence Diagram แสดงขั้นตอนการทำงานของฟังก์ชัน Class Diagram แสดงองค์ประกอบคลาส และ Data Structure Tree แสดงโครงสร้างของฐานข้อมูล NoSQL

1.5.3 การออกแบบระบบ (System Design)

ออกแบบระบบตามที่ได้วิเคราะห์ไว้ในขั้นตอนก่อนหน้านี้ โดย

1.5.3.1 ออกแบบโครงสร้างของข้อมูล (Data Structure Design)

เพื่อให้ง่ายต่อการแลกเปลี่ยนข้อมูลข้ามระบบหรือข้ามแพลตฟอร์ม ได้ทำการออกแบบโครงสร้างข้อมูลตามหลักการของ NoSQL Database ที่ทำงานบนคลาวด์ โดยเลือกใช้ Firebase

1.5.3.2 ออกแบบส่วนติดต่อกับผู้ใช้งาน (User Interface Design)

การออกแบบหน้าจอที่ใช้ในการติดต่อกับผู้ใช้ User Interface (UI) มุ่งเน้นออกแบบให้ผู้ใช้ใช้งานง่ายและสามารถใช้งานได้จริง โดยคำนึงถึง User Experience (UX) เป็นหลักเพื่อให้ผู้ใช้ใช้งานได้ง่าย สามารถใช้งานได้ด้วยตนเอง

1.5.3.3 ออกแบบอัลกอริทึมในการคำนวณระยะห่างระหว่างผู้ใช้ จะแสดงผลให้แก่ผู้ใช้ในบริเวณใกล้เคียงกัน โดยการคำนวณจากตำแหน่งปัจจุบันของผู้ใช้

1.5.4 การพัฒนาระบบ (System Development)

นำข้อมูลที่ได้ทำการวิเคราะห์และออกแบบไว้มาเขียนเป็นชุดคำสั่งด้วย ภาษา Dart และ Flutter Framework เขียนชุดคำสั่งด้วยโปรแกรม Microsoft Visual Studio Code และเรียกใช้ API ต่างๆ ของ Flutter ทั้งในการสร้างส่วนติดต่อกับผู้ใช้ หรือ UI และการสร้างฟังก์ชันต่างๆ รวมถึงการติดต่อกับ Firebase

1.5.5 การทดสอบระบบ (System Testing)

ในการทดสอบระบบผู้จัดทำได้ทำการทดสอบพร้อมกับขั้นตอนของการเขียนชุดคำสั่ง โดยทำการทดสอบการทำงานฟังก์ชันย่อย (Unit Testing) โดยถ้าฟังก์ชันไม่สามารถทำงานได้ตามที่กำหนดจะทำการปรับแก้ทันที และทดสอบการทำงานร่วมกันระหว่างฟังก์ชันแต่ละฟังก์ชัน (Integration Testing) ว่าสามารถทำงานร่วมกัน สามารถส่งข้อความ (Message) ถึงกันได้ถูกต้องหรือไม่ ทำการคำนวณระยะห่างได้ถูกต้องหรือไม่ ถ้าพบข้อผิดพลาดจะทำการปรับแก้ทันที และได้ทำการทดสอบระบบทั้งหมด (System Testing) เมื่อพัฒนาทุกฟังก์ชันเสร็จแล้ว เพื่อทดสอบการทำงานของระบบทั้งหมดว่าสามารถทำงานร่วมกันและเป็นไปตามหน้าที่ที่กำหนดหรือไม่

1.5.6 การจัดทำเอกสารประกอบปริญญานิพนธ์ (Documentation)

เป็นการจัดทำเอกสารเพื่อนำเสนอรายละเอียดของปริญญานิพนธ์สำหรับเป็นเอกสารอ้างอิง และแนวทางในการศึกษาต่อไป

1.6 แผนและระยะเวลาในการดำเนินงานปริญญานิพนธ์

ตารางที่ 1.1 ระยะเวลาในการดำเนินงานปริญญานิพนธ์

กิจกรรม	2566					2567				
	ส.ค	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.
1. รวบรวมความต้องการและศึกษาข้อมูล		←	→							
2. วิเคราะห์ระบบ			←	→						
3. ออกแบบระบบ					←	→	→	→	→	
4. พัฒนาระบบ					←	→	→	→	→	
5. ทดสอบระบบ								←	→	
6. จัดทำเอกสารประกอบปริญญานิพนธ์							←	→	→	→

1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนา

1.7.1 ฮาร์ดแวร์ (Hardware)

1.7.1.1 เครื่องคอมพิวเตอร์ โน้ตบุ๊ค Lenovo IdeaPad Gaming 3

1.7.1.2 โทรศัพท์มือถือ Android

1.7.2 ซอฟต์แวร์ (Software)

1.7.2.1 ระบบปฏิบัติการ Microsoft Windows 11

1.7.2.2 โปรแกรม Visual Studio Code

1.7.2.3 Flutter Framework

1.7.2.4 Firebase API

1.7.2.5 Version Control

1.8 อุปกรณ์และเครื่องมือที่รองรับระบบ

1.8.1 ฮาร์ดแวร์ (Hardware)

1.8.1.1 โทรศัพท์มือถือหรือแท็บเล็ตที่สามารถเชื่อมต่อเครือข่ายอินเทอร์เน็ตได้

1.8.2 ซอฟต์แวร์ (Software)

1.8.2.1 ระบบปฏิบัติการ Android

บทที่ 2
การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง

ในการจัดทำปริญญานิพนธ์นี้ ผู้จัดทำได้ทำการศึกษาค้นคว้าแนวคิด ทฤษฎีเทคโนโลยี และเครื่องมือต่างๆ เพื่อนำมาประยุกต์ใช้ในการพัฒนาปริญญานิพนธ์ ดังต่อไปนี้

2.1 ภาษา Dart[1]

ดาร์ต (Dart) เป็นภาษาโปรแกรมที่ออกแบบโดย Lars Bak และ Kasper Lund และพัฒนาโดยกูเกิล สามารถใช้ในการพัฒนาโปรแกรมประยุกต์บนเว็บ และ แอปพลิเคชันที่ทำงานบนอุปกรณ์โทรศัพท์เคลื่อนถือหรือมือถือ ตลอดจนแอปพลิเคชันที่ทำงานบนฝั่งของเซิร์ฟเวอร์ และแอปพลิเคชันที่ทำงานบนอุปกรณ์คอมพิวเตอร์เดสก์ท็อป ดาร์ตเป็นภาษาเชิงวัตถุอิงคลาส (Object Oriented Language) โครงสร้างของภาษา DART คล้ายกับ C/C++ และ Java โดยที่จะมีความเป็นภาษาแบบ Structure Programming แต่ก็ยังมีความสามารถแบบภาษาประเภท Object Oriented Programming ด้วย ภาษาดาร์ตถูกใช้งานอย่างแพร่หลายในการพัฒนาแอปพลิเคชันต่างๆ โดยเฉพาะอย่างยิ่งในการสร้างแอปพลิเคชันที่รันบนระบบปฏิบัติการแอนดรอยด์ด้วย Flutter framework ที่ใช้ดาร์ตเป็นภาษาหลัก และภาษาดาร์ตเหมาะสมกับการพัฒนาแอปพลิเคชันที่มีขนาดใหญ่เนื่องจากภาษาดาร์ตเป็นภาษาที่มีความยืดหยุ่นสูง และมีความสามารถในการจัดการกับการพัฒนาขนาดใหญ่ได้อย่างมีประสิทธิภาพ

```dart
class People{
    String sayHi() => "Hi!";
}

void main(){
    People p = People();
    String Function() f = p.sayHi;
    print(f());      // output: Hi!
}
```

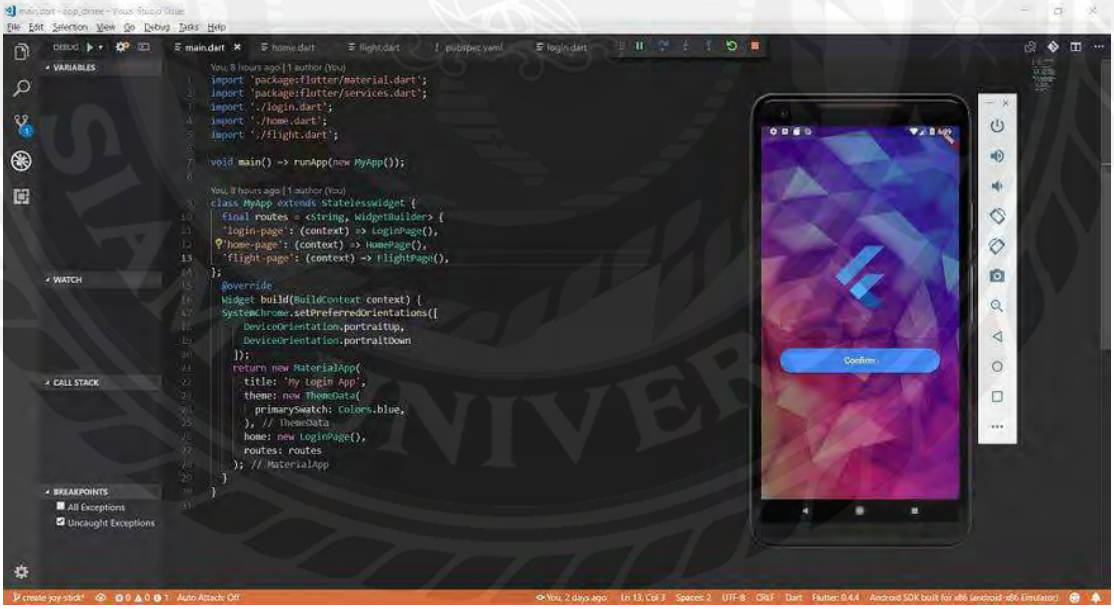
รูปที่ 2.1 ตัวอย่างการสร้างคลาสด้วยภาษาดาร์ต

[1] https://th.wikipedia.org/wiki/ภาษาดาร์ต

2.2 **Flutter Framework**[2]

ฟลัตเตอร์ (Flutter) เป็นเฟรมเวิร์ก (Framework) ส่วนติดต่อผู้ใช้ที่โอเพนซอร์ส พัฒนาโดยกูเกิล สามารถใช้พัฒนาแอปพลิเคชันได้แบบข้ามแพลตฟอร์ม (Cross-platform) จากฐานชุดคำสั่งเดี่ยว (Single Codebase) ไปทำงานบนเว็บเบราว์เซอร์, แอนดรอยด์, ไอโอเอส, กูเกิล ฟิวเชีย, เว็บเบราว์เซอร์, ลินุกซ์, แมคโอเอส (รวมไปถึง แอปเปิลซิลิคอน) และ ไมโครซอฟท์ วินโดวส์ได้ โดยฟลัตเตอร์มีจุดเด่น ดังนี้

- ระบบ Hot Reload โดยเมื่อมีการทดสอบ, การสร้าง, การเพิ่มฟีเจอร์หรือการกระทำต่าง ๆ กับส่วนติดต่อกับผู้ใช้ (User Interface) หรือ UI จะต้องมีการรีโหลด (Reload) เพื่อให้หน้า UI update ซึ่งระบบ Hot Reload จะเข้ามาช่วยในส่วนของการ reload โดยจุดเด่นของระบบนี้คือการย่นระยะเวลาที่ใช้ในการ reload ให้เหลือเพียงเสี้ยววินาทีเท่านั้น ทำให้การพัฒนา UI ของ application มีความรวดเร็วขึ้นอย่างมาก
- Build-In ที่ช่วยในการออกแบบ UI ให้มีความสวยงามยิ่งขึ้นอย่าง Material Design และ Cupertino (iOS-flavor) Framework ที่ช่วยให้การทำ animation ต่าง ๆ หรือ gesture ของ UI เป็นเรื่องง่ายยิ่งขึ้น
- สามารถใช้งานร่วมกับ IDE ที่กำลังเป็นที่นิยมอยู่ในปัจจุบันอย่าง VS Code และ Android Studio ได้อีกด้วย



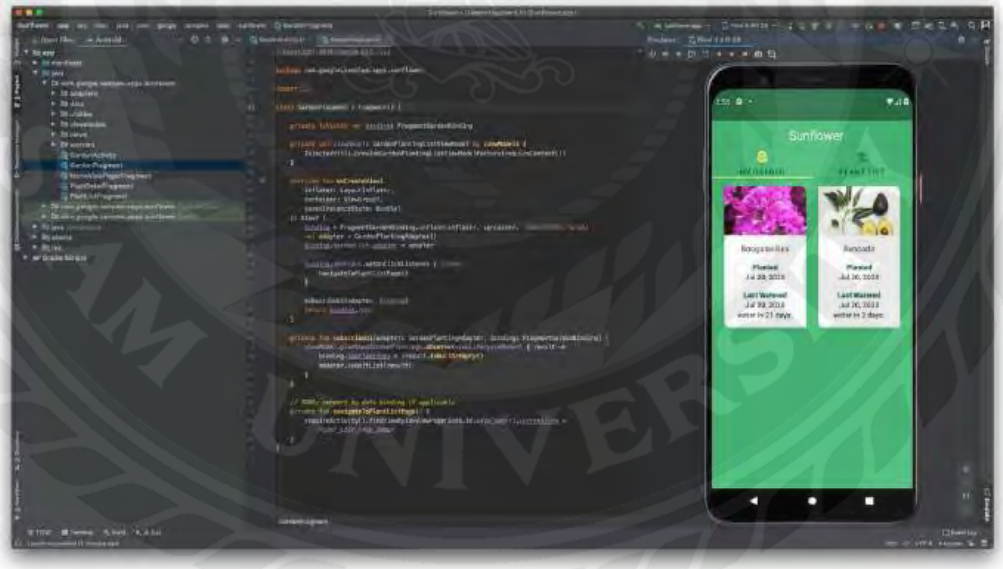
รูปที่ 2.2 ตัวอย่างแอปพลิเคชันที่พัฒนาด้วยฟลัตเตอร์

[2] https://th.wikipedia.org/wiki/ฟลัตเตอร์

2.3 **Android Studio**[3]

Android Studio เป็นเครื่องมือในการพัฒนาโปรแกรมชนิด IDE (Integrated Development Environment) ของกูเกิลสำหรับใช้ในการพัฒนาแอปพลิเคชันบนระบบปฏิบัติการแอนดรอยด์ โดยพัฒนาจากแนวคิดพื้นฐานมาจาก IntelliJ IDEA คล้ายๆ กับการทำงานของ Eclipse และ Android ADT Plugin โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนาแอปพลิเคชันที่ทำงานบนระบบปฏิบัติการแอนดรอยด์ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบส่วนติดต่อกับผู้ใช้ (GUI) ที่ช่วยให้สามารถแสดงมุมมองของแอปพลิเคชัน (Preview) ที่มีมุมมองที่แตกต่างกันบนสมาร์ทโฟนแต่ล่ะรุ่น สามารถแสดงผลบางอย่างได้ทันทีโดยไม่ต้องทำการรันแอปพลิเคชันบนเครื่องจำลอง (Emulator) รวมทั้งแก้ไขปรับปรุงในเรื่องของความเร็วของ เครื่องจำลองที่ยังคงพบปัญหากันอยู่ในปัจจุบัน

Android SDK เป็นเครื่องมือสำหรับพัฒนาโปรแกรมหรือแอปพิเคชันบนระบบปฏิบัติการแอนดรอยด์ โดยในชุด SDK นั้นจะมีโปรแกรมและไลบรารี่ต่างๆ ที่จำเป็นต่อการพัฒนาแอปพลิเคชั่นบนแอนดรอยด์ อย่างเช่น เครื่องจำลอง (Emulator) ซึ่งทำให้ผู้ใช้สามารถสร้างแอปพลิเคชั่นและนำมาทดลองรันบนเครื่องจำลองก่อน โดยมีสภาวะแวดล้อมเหมือนอุปกรณ์มือถือที่รันระบบปฏิบัติการแอนดรอยด์จริง



รูปที่ 2.3 โปรแกรม Android Studio

[3] https://medium.com/@boyreallife/android-studio-lab-3sb04-4202ceff7167

2.4 **Firebase**[4]

Firebase คือชุดเครื่องมือและบริการที่ครอบคลุมจากกูเกิล ซึ่งนำเสนอเป็นแพลตฟอร์ม Backend-as-a-Service (BaaS) ช่วยให้นักพัฒนาสร้าง เปิดใช้ และขยายทั้งแอปพลิเคชันมือถือและเว็บได้อย่างง่ายดาย มีฐานข้อมูลเรียลไทม์ การพิสูจน์ตัวตน พื้นที่เก็บข้อมูล โฮสติ้ง และฟีเจอร์อื่นๆ และจัดการทั้งหมดได้จากแพลตฟอร์มเดียว

หัวใจหลักคือ Firebase มีฐานข้อมูลแบบเรียลไทม์ที่ซิงค์ (Sync) ข้อมูลระหว่างอุปกรณ์ที่เชื่อมต่อทั้งหมดแบบเรียลไทม์ ฐานข้อมูลใช้โมเดลข้อมูลเชิงเอกสาร NoSQL ซึ่งช่วยให้นักพัฒนาสามารถจัดเก็บข้อมูลในลักษณะที่ยืดหยุ่นและปรับขนาดได้ ข้อมูลถูกจัดเก็บในรูปแบบ JSON Firebase ยังให้บริการการตรวจสอบสิทธิ์ที่มีประสิทธิภาพ ช่วยให้นักพัฒนาสามารถใช้การตรวจสอบผู้ใช้ที่ปลอดภัยในแอปพลิเคชัน]ของตนได้สะดวกและง่าย

อกจากฐานข้อมูลแบบเรียลไทม์และบริการตรวจสอบสิทธิ์แล้ว Firebase ยังมีพื้นที่เก็บข้อมูลบนคลาวด์ที่ช่วยให้นักพัฒนาสามารถจัดเก็บไฟล์ขนาดใหญ่ เช่น รูปภาพและวิดีโอ ที่เก็บข้อมูลถูกรวมเข้ากับฐานข้อมูลแบบเรียลไทม์ มอบประสบการณ์ที่ราบรื่นในการจัดเก็บและเรียกใช้ข้อมูล

Firebase มีโซลูชันการโฮสต์แบบสแตติก ซึ่งช่วยให้นักพัฒนาสามารถโฮสต์เนื้อหาเว็บของตนได้โดยตรงจากแพลตฟอร์ม Firebase เช่น ไฟล์ HTML, CSS และ JavaScript ซึ่งทำให้การปรับใช้และการปรับขนาดเว็บแอปพลิเคชันทำได้ง่ายโดยไม่จำเป็นต้องตั้งค่าและกำหนดค่าฝั่งเซิร์ฟเวอร์ที่ซับซ้อน

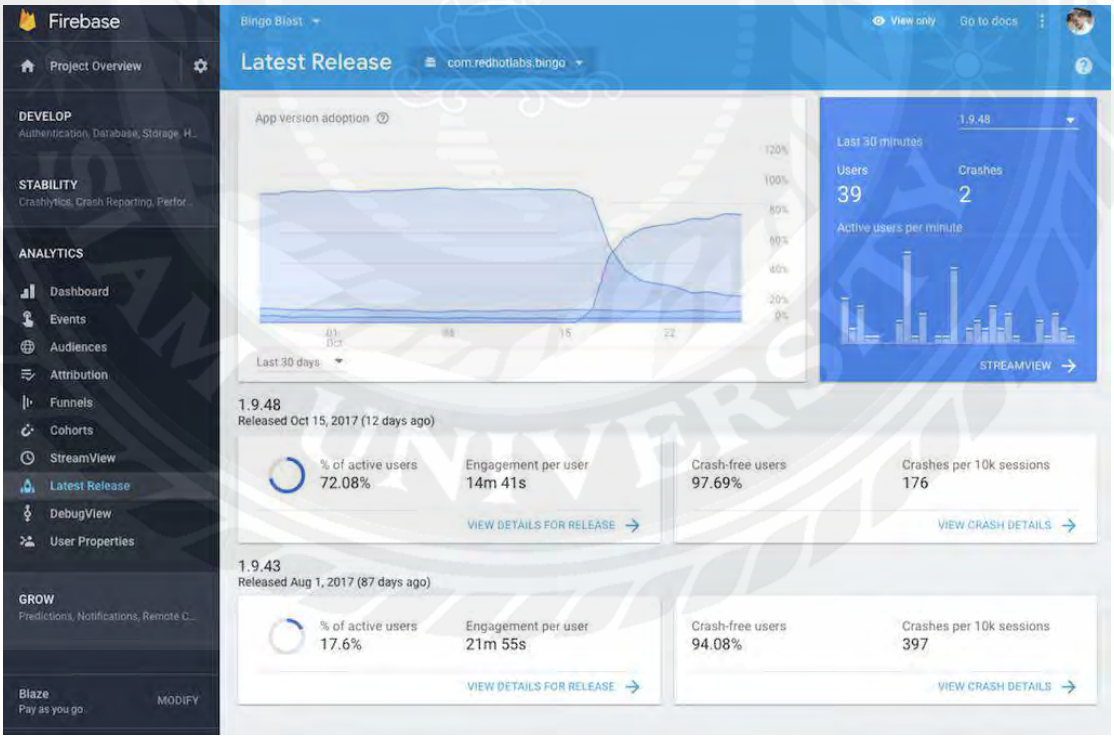
Firebase มีชุดเครื่องมือสำหรับวิเคราะห์การใช้งานและประสิทธิภาพของแอปพลิเคชัน เช่น การรายงานข้อขัดข้อง การตรวจสอบประสิทธิภาพ และการกำหนดค่าระยะไกล เครื่องมือเหล่านี้ให้ข้อมูลเชิงลึกที่มีคุณค่าเกี่ยวกับการใช้แอปพลิเคชัน ช่วยให้นักพัฒนาสามารถระบุและแก้ไขปัญหาด้านประสิทธิภาพและปรับปรุงประสบการณ์ผู้ใช้โดยรวม

Firebase สามารถใช้เพื่อพัฒนาแอปพลิเคชันได้หลากหลาย อาทิเช่น

- โซเชียลมีเดีย : ฐานข้อมูลตามเวลาจริงของ Firebase และบริการตรวจสอบสิทธิ์ทำให้เป็นแพลตฟอร์มที่เหมาะสำหรับการ สร้างแอปโซเชียลมีเดีย ซึ่งผู้ใช้สามารถเชื่อมต่อและแชร์เนื้อหาได้แบบเรียลไทม์
- อีคอมเมิร์ซ : Firebase ให้บริการต่างๆ ที่จำเป็นสำหรับการ สร้างแอปอีคอมเมิร์ซ เช่น การซิงโครไนซ์ข้อมูลแบบเรียลไทม์ การยืนยันตัวตนผู้ใช้ และที่เก็บข้อมูลบนคลาวด์สำหรับรูปภาพผลิตภัณฑ์

[4] https://appmaster.io/th/blog/firebase-khuue-aair

- เกม : ฐานข้อมูลเรียลไทม์และเครื่องมือวิเคราะห์ของ Firebase ทำให้เป็นแพลตฟอร์มที่เหมาะสำหรับการ สร้างแอปเกม ซึ่งผู้ใช้สามารถแข่งขันและทำงานร่วมกันได้แบบเรียลไทม์
- ส่งข้อความ : ฐานข้อมูลเรียลไทม์และบริการตรวจสอบสิทธิ์ของ Firebase ทำให้เป็นแพลตฟอร์มที่เหมาะสำหรับการ สร้างแอพส่งข้อความ ที่ผู้ใช้สามารถสื่อสารและแชร์เนื้อหาแบบเรียลไทม์
- ตามตำแหน่ง : Firebase ให้บริการซิงโครไนซ์ข้อมูลแบบเรียลไทม์และบริการพื้นที่เก็บข้อมูลบนคลาวด์ ทำให้เป็นแพลตฟอร์มที่เหมาะสำหรับการสร้างแอพตามตำแหน่ง ซึ่งผู้ใช้สามารถแชร์ตำแหน่งและค้นหาจุดสนใจในบริเวณใกล้เคียง
- การทำงานร่วมกัน : ฐานข้อมูลแบบเรียลไทม์ของ Firebase และบริการตรวจสอบสิทธิ์ทำให้เป็นแพลตฟอร์มที่เหมาะสำหรับการสร้างแอปการทำงานร่วมกัน ซึ่งผู้ใช้สามารถทำงานร่วมกันในโครงการและงานต่างๆ ได้แบบเรียลไทม์
- ยูทิลิตี้ : Firebase ให้บริการที่หลากหลายซึ่งจำเป็นสำหรับการสร้างแอพยูทิลิตี้ เช่น การซิงโครไนซ์ข้อมูลแบบเรียลไทม์ การตรวจสอบสิทธิ์ผู้ใช้ และที่เก็บข้อมูลบนคลาวด์สำหรับจัดเก็บข้อมูลผู้ใช้



รูปที่ 2.4 หน้าจอ Firebase แสดงการวิเคราะห์แอปพลิเคชัน

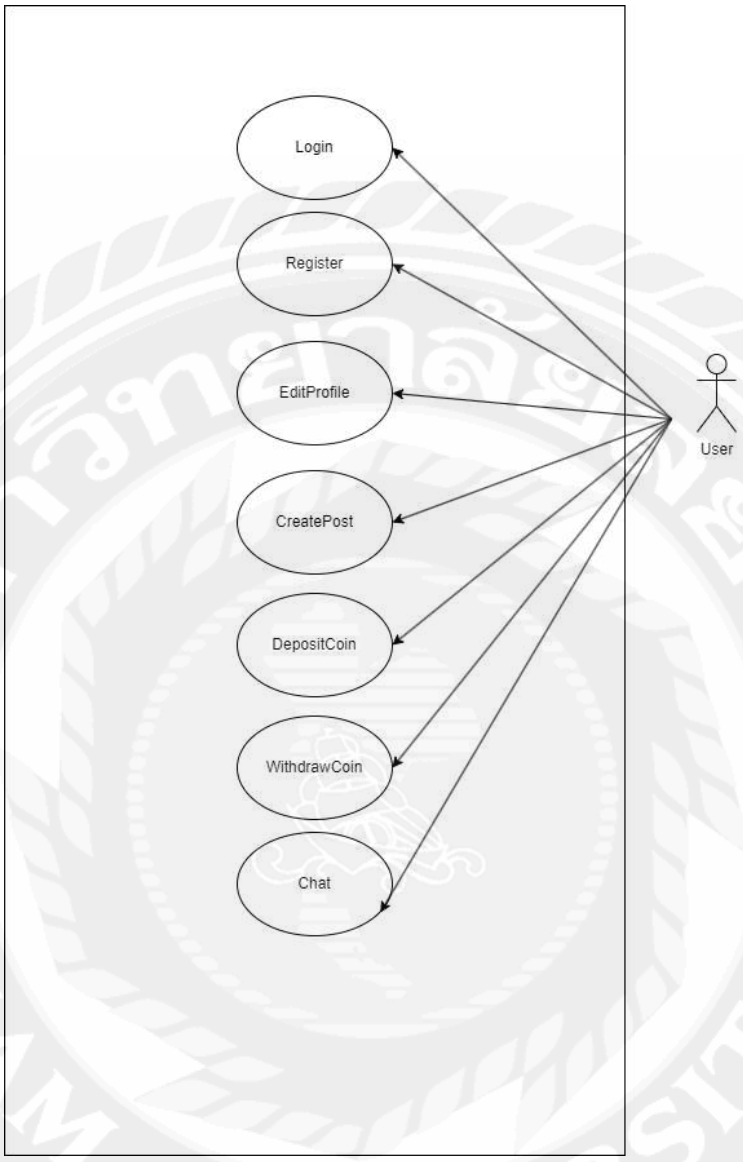
บทที่ 3
การวิเคราะห์ระบบ

3.1 รายละเอียดของปริญญานิพนธ์

ผู้จัดทำได้พัฒนาระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้เป็นโมบายแอปพลิเคชัน (Mobile Application) โดยใช้ชื่อแอปพลิเคชันว่า “Deal” เป็นแอปพลิเคชันสำหรับรับ-ฝากซื้อสินค้าจากคนที่อยู่ใกล้เคียงกับร้านค้าหรือสินค้าที่ต้องการ เพื่อเป็นทางเลือกสำหรับผู้ที่ไม่มีเวลาในการซื้อสินค้าที่อยู่ใกล้เคียงด้วยตนเองได้ โดยแอปพลิเคชันนี้ช่วยให้ผู้ซื้อสามารถประหยัดเวลาในการเดินทางไปซื้อด้วยตนเอง เป็นช่องทางเพิ่มยอดขายให้กับร้านค้าทั่วไปและร้านค้าชุมชน และยังสร้างรายได้ให้กับผู้ที่รับฝากซื้อสินค้า โดยจะได้เป็นค่าจ้างตามตกลง

ดังนั้นในการพัฒนาระบบ จะประกอบด้วย ผู้ใช้งานจะต้องลงทะเบียนเพื่อเข้าใช้งานระบบเสียก่อนเป็นลำดับแรกเมื่อเข้าสู่แอปพลิเคชันในครั้งแรก โดยผู้ใช้สามารถลงทะเบียนผ่านอีเมล (Email Address) เมื่อผู้ใช้เข้าสู่ระบบเรียบร้อยแล้ว ระบบจะทำงานเก็บข้อมูลที่อยู่ (Location) ของผู้ใช้เพื่อทำงานคำนวณระยะห่างและแสดงโพสต์ที่ระยะใกล้เคียงกับผู้ใช้งาน คนอื่นๆ ผู้ใช้ของระบบสามารถเป็นได้ทั้งผู้ฝากซื้อสินค้าและผู้รับฝากซื้อสินค้า เมื่อผู้ใช้ทำการลงทะเบียนเข้าใช้งานระบบจะทำการสร้างกระเป๋าเงินอิเล็กทรอนิกส์ (E-Wallet) ให้สำหรับใช้เป็นช่องทางในการชำระค่าสินค้าและบริการและสำหรับรับค่าจ้างบริการฝากซื้อสินค้า พัฒนาแอปพลิเคชันพัฒนาโดยใช้ Flutter Framework ในการสร้าง Native Interface สำหรับแอนดรอยด์ ข้อมูลของผู้ใช้งานจะถูกจัดเก็บลงใน Firebase และเขียนชุดคำสั่งด้วยภาษา Dart เครื่องมือที่ใช้เขียนชุดคำสั่งคือ Microsoft Visual Studio Code ผู้จัดทำยังได้พัฒนาระบบการแจ้งเตือน (Notification System) เพื่อให้ผู้ใช้ได้รับการแจ้งเตือนเมื่อมีผู้โพสต์ใหม่ในระยะใกล้เคียง หรือ มีการตอบกลับจากโพสต์ของตัวผู้ใช้และเพิ่มระบบพูดคุย (Chat) เฉพาะโพสต์นั้นๆ เพื่อให้ผู้ฝากซื้อสินค้าและผู้รับฝากซื้อสินค้าสามารถติดต่อสื่อสารกันได้โดยตรงเกี่ยวกับรายละเอียดของการซื้อสินค้านั้นๆ

3.2 แสดงฟังก์ชันการทำงานของระบบด้วย Use Case Diagram



รูปที่ 3.1 Use Case Diagram ของระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้

3.3 คำอธิบายรายละเอียดของยูสเคส (Use Case Description)

ตารางที่ 3.1 แสดงรายละเอียดของ Use Case Login

Use Case Name	LOGIN
Use Case ID	UC1
Brief Description	สำหรับยืนยันตัวตน โดยใช้อีเมลและรหัสผ่าน เพื่อเข้าใช้งานแอปพลิเคชัน Deal
Primary Actors	User
Secondary Actors	-
Preconditions	จะต้องทำการลงทะเบียนก่อนเข้าใช้งานระบบ
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้เข้าสู่ระบบแอปพลิเคชัน 2. ระบบแสดง Field สำหรับป้อนข้อมูลเพื่อเข้าสู่ระบบ โดยใช้อีเมลและรหัสผ่าน 2.1 ถ้าผู้ใช้ป้อนข้อมูล อีเมลผู้ใช้ และ รหัสผ่าน ถูกต้อง 2.1.1 ระบบจะแสดงหน้าหลักของแอปพลิเคชัน 2.2 ถ้าผู้ใช้ป้อนข้อมูล อีเมลผู้ใช้ และ รหัสผ่าน ไม่ถูกต้อง 3.2.1 ระบบจะแสดง ข้อความ ไม่สามารถเข้าสู่ระบบได้ 3.2.2 ผู้ใช้ต้องป้อนข้อมูล เพื่อเข้าสู่ระบบอีกครั้ง
Post Condition	เมื่อเข้าสู่ระบบสำเร็จ จะสามารถใช้งานฟังก์ชันต่างๆ ได้
Alternative Flows	-

ตารางที่ 3.2 แสดงรายละเอียดของ Use Case Register

Use Case Name	REGISTER
Use Case ID	UC2
Brief Description	สำหรับผู้ใช้รายใหม่ลงทะเบียนข้าใช้งานแอปพลิเคชัน Deal
Primary Actors	User
Secondary Actors	-
Preconditions	-
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้ทำการลงทะเบียนข้อมูลผู้ใช้ 2. ผู้ใช้สามารถลงทะเบียนผู้ใช้ผ่านทางอีเมล ในการลงทะเบียนใช้งานแอปพลิเคชัน 3. ผู้ใช้งานป้อนข้อมูลสำหรับลงทะเบียนผู้ใช้งาน ดังนี้ ชื่อผู้ใช้งาน, อีเมล, รหัสผ่าน 3.1 ถ้าผู้ใช้งานป้อนข้อมูลครบตามที่กำหนด 3.1.1 ระบบจะส่งลิงก์ยืนยันตัวตนไปที่อีเมลของผู้ใช้ 4. ผู้ใช้งานทำการลงทะเบียนข้อมูลผู้ใช้งาน จัดเก็บข้อมูลใน Firebase ที่มีชื่อว่า users
Post Condition	เมื่อลงทะเบียนสำเร็จระบบจะเข้าสู่หน้า Login เพื่อให้ผู้ใช้ยืนยันตัวตนอีกครั้ง
Alternative Flows	-

ตารางที่ 3.3 แสดงรายละเอียดของ Use Case EditProfile

Use Case Name	EDITPROFILE
Use Case ID	UC3
Brief Description	สำหรับจัดการข้อมูลส่วนตัวของผู้ใช้งาน
Primary Actors	User
Secondary Actors	-
Preconditions	-
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้งานทำการคลิกไอคอนแก้ไขโปรไฟล์ที่หน้า Profile 2. ผู้ใช้งานแก้ไขข้อมูลผู้ใช้งาน ดังนี้ รูปภาพผู้ใช้งาน เพศ และประวัติส่วนตัว (bio) ผู้ใช้ 3. ผู้ใช้งานจัดการข้อมูลสำเร็จ โดยนำข้อมูลที่ผู้ใช้ทำการเปลี่ยนแปลง ไปทำการปรับปรุงแก้ไขข้อมูลใน Firebase ที่มีชื่อว่า users
Post Condition	-
Alternative Flows	-

ตารางที่ 3.4 แสดงรายละเอียดของ Use Case CreatePost

Use Case Name	CREATEPOST
Use Case ID	UC4
Brief Description	สำหรับสร้างโพสต์เพื่อฝากซื้อสินค้าหรือรับฝากซื้อสินค้า
Primary Actors	User
Secondary Actors	-
Preconditions	-
Main Flow	1. ยูสเคสจะเริ่มขึ้นผู้ใช้ทำการคลิก Create Post ในหน้าหลัก 2. ผู้ใช้งานป้อนข้อมูลรายละเอียดโพสต์ดังนี้ ประเภทโพสต์ หัวข้อโพสต์ เนื้อหาโพสต์ สถานที่ ราคา และรูปภาพ 3. ระบบจัดเก็บข้อมูลใน Firebase ที่มีชื่อว่า posts
Post Condition	จะมีการแจ้งเตือนไปยังผู้ใช้คนอื่นที่อยู่ในระยะห่างจากผู้ใช้ 3 กิโลเมตร และเมื่อผู้ใช้คลิกที่แจ้งเตือนโพสต์จะสามารถเข้าไปอ่านโพสต์นั้นได้
Alternative Flows	-

ตารางที่ 3.5 แสดงรายละเอียดของ Use Case DepositCoin

Use Case Name	DEPOSITCOIN
Use Case ID	UC5
Brief Description	-
Primary Actors	User
Secondary Actors	-
Preconditions	
Main Flow	1. ยูสเคสจะเริ่มขึ้นผู้ใช้คลิก Deposit ที่หน้า Wallet 2. ผู้ใช้งานป้อนข้อมูลรายละเอียดบัตรดังนี้ เลขที่บัตรเครดิต วันหมดอายุ เลข CVV ชื่อบนหน้าบัตร และจำนวนเงินที่ต้องการฝากเข้าแอปพลิเคชัน 3. ระบบจะทำการเพิ่มจำนวน Coin ใน Firebase ของ users
Post Condition	ผู้ใช้สามารถนำเงินใน Wallet ไปใช้ในการชำระค่าสินค้าและค่าบริการจ้างฝากซื้อสินค้าได้
Alternative Flows	-

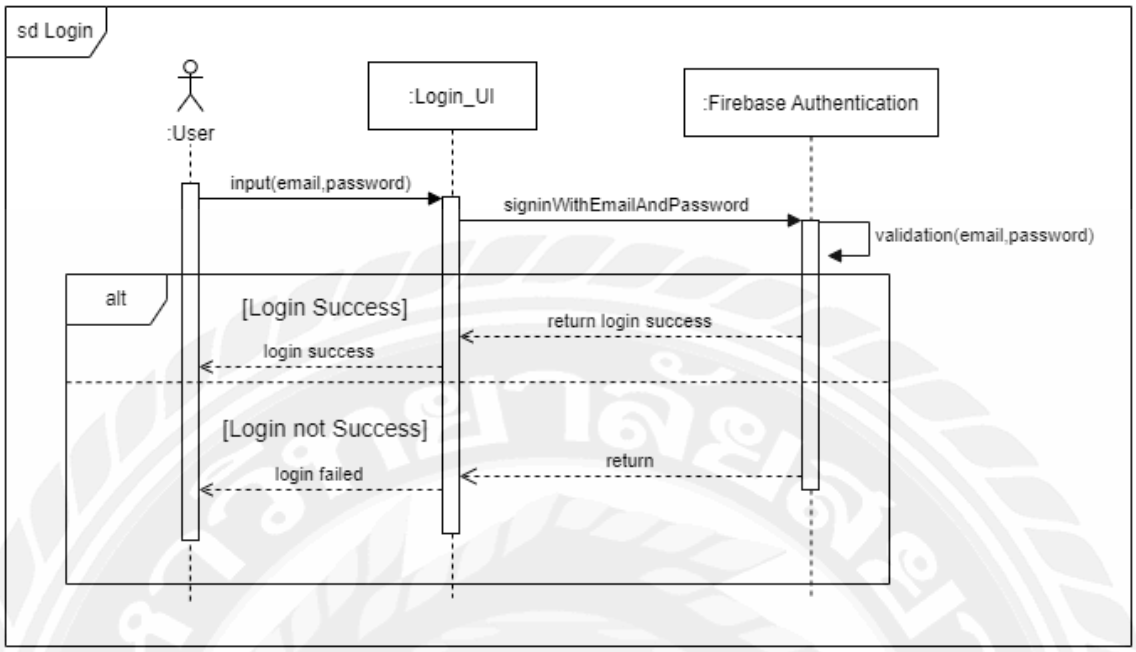
ตารางที่ 3.6 แสดงรายละเอียดของ Use Case WithdrawCoin

Use Case Name	WIDTHDRAWCOIN
Use Case ID	UC6
Brief Description	-
Primary Actors	Users
Secondary Actors	-
Preconditions	-
Main Flow	1. ยูสเคสจะเริ่มขึ้นผู้ใช้คลิก Withdraw ที่หน้า Wallet 2. ผู้ใช้งานป้อนข้อมูลรายละเอียดบัตรดังนี้ เลขที่บัตร วันหมดอายุ เลข CVV ชื่อบนหน้าบัตร จำนวนเงินที่ต้องการถอน 3. ระบบจะทำการถอนจำนวน Coin ใน Firebase ของ users
Post Condition	จำนวนเงินที่ถอนจะถูกโอนไปยัง Wallet ของผู้ที่รับฝากซื้อสินค้า
Alternative Flows	-

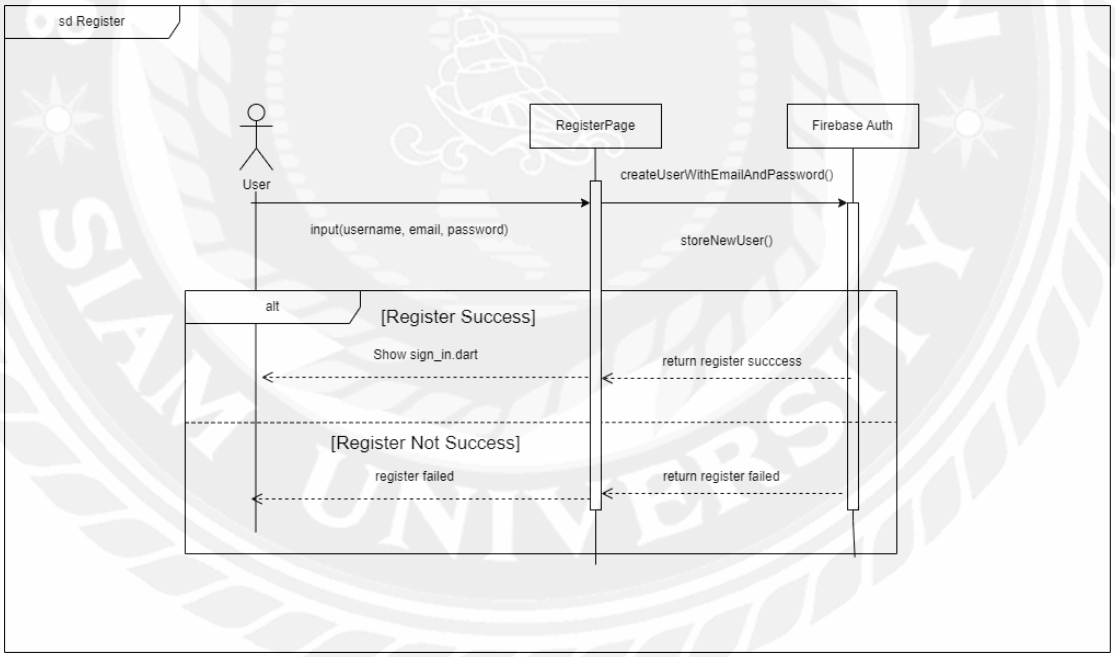
ตารางที่ 3.7 แสดงรายละเอียดของ Use Case Chat

Use Case Name	CHAT
Use Case ID	UC7
Brief Description	User
Primary Actors	-
Secondary Actors	-
Preconditions	-
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้คลิกที่หน้ารายละเอียดโพสต์หลังจากได้ทำการตกลงโพสต์แล้ว 2. ผู้ใช้จะสามารถพูดคุยรายละเอียดของโพสต์กับฝ่ายตรงข้าม
Post Condition	จะมีการแจ้งเตือนไปยังผู้รับโพสต์
Alternative Flows	-

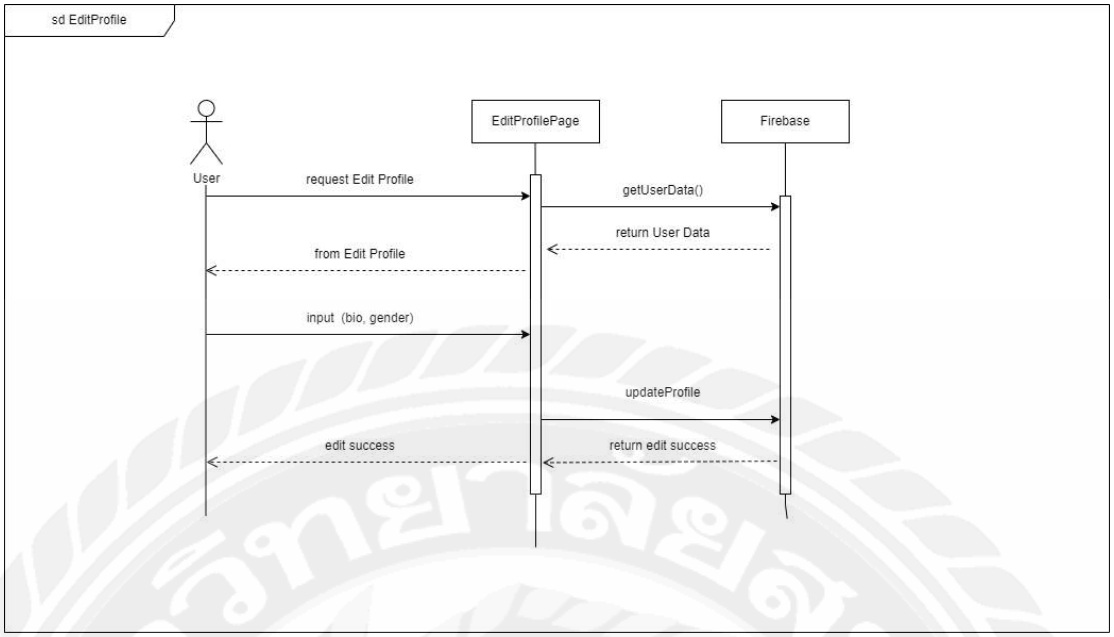
3.4 แสดงขั้นตอนการทำงานของฟังก์ชันด้วย Sequence Diagram



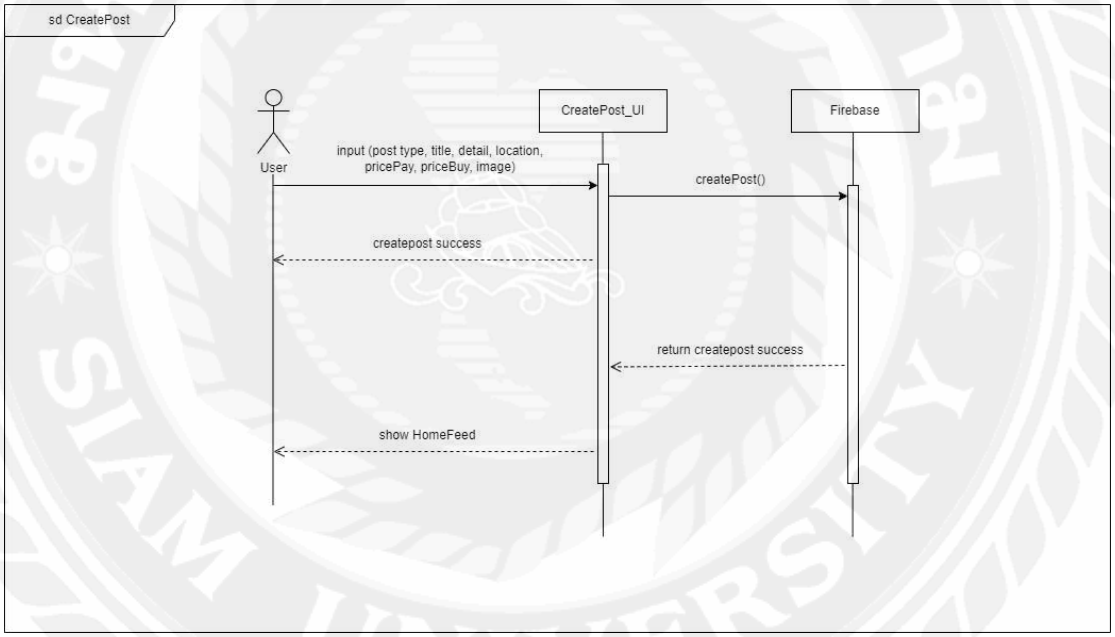
รูปที่ 3.2 Sequence Diagram : Login



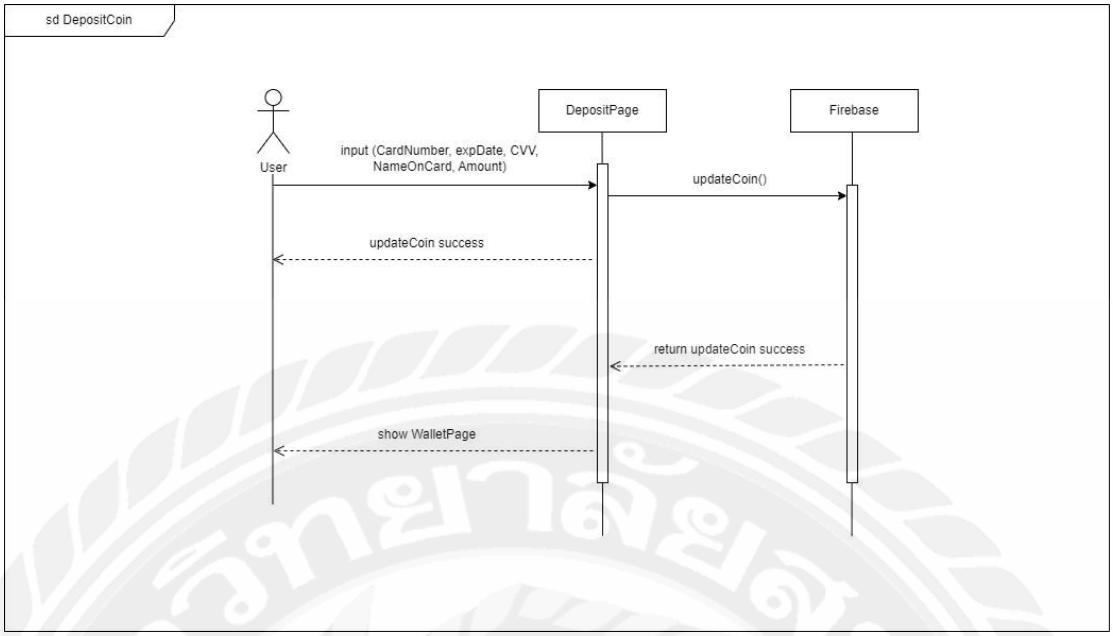
รูปที่ 3.3 Sequence Diagram : Register



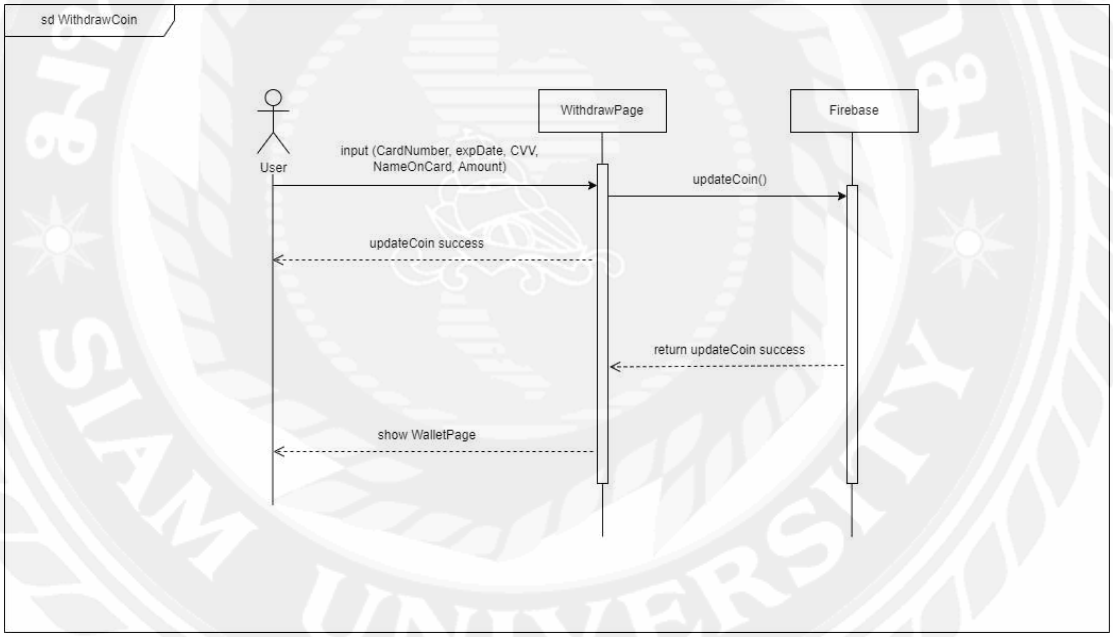
รูปที่ 3.4 Sequence Diagram : Edit Profile



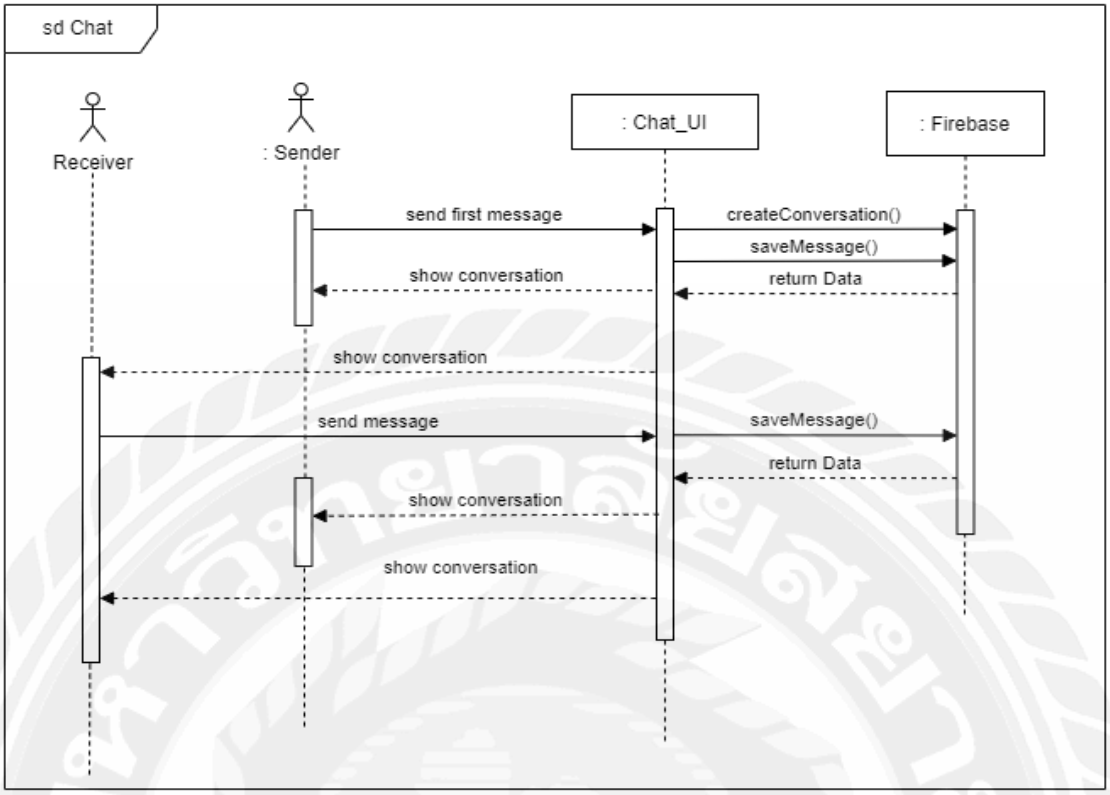
รูปที่ 3.5 Sequence Diagram : CreatePost



รูปที่ 3.6 Sequence Diagram : DepositCoin

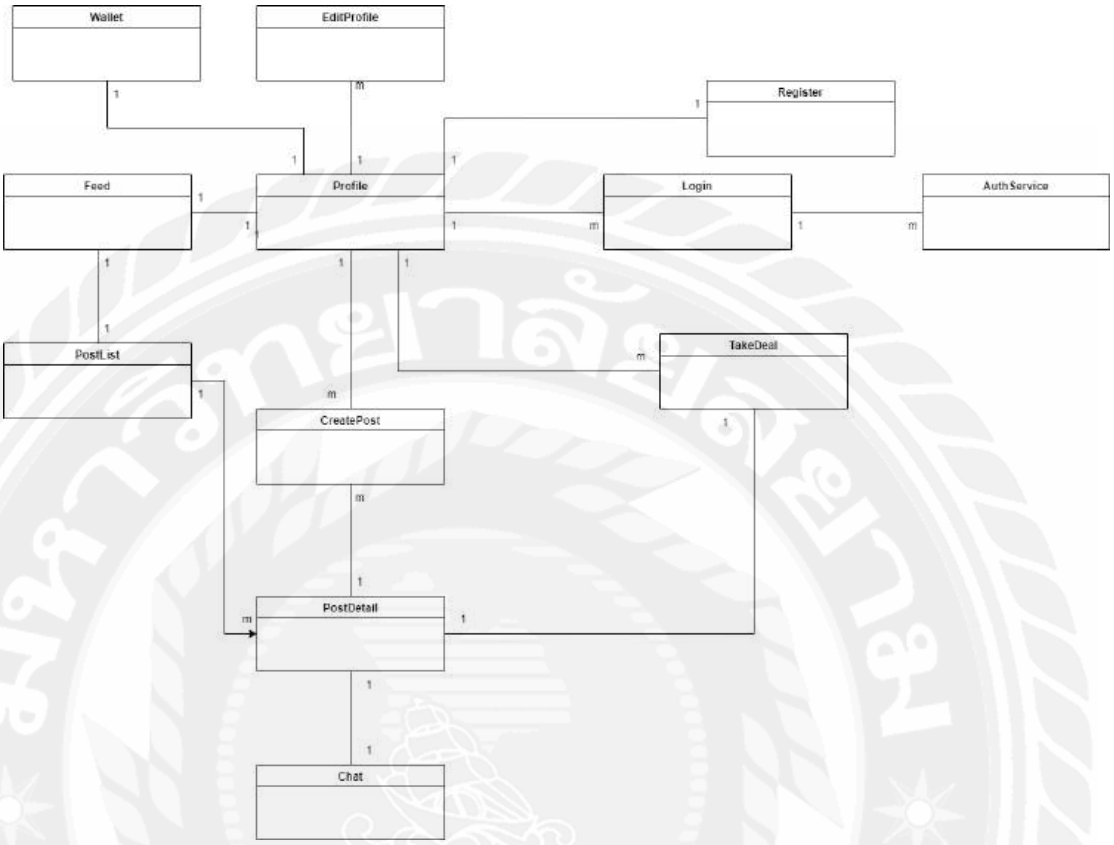


รูปที่ 3.7 Sequence Diagram : WithdrawCoin



รูปที่ 3.8 Sequence Diagram : Chat

3.5 แผนภาพแสดงองค์ประกอบของคลาส (Class Diagram)



รูปที่ 3.9 Class Diagram ของระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้

3.6 รายละเอียดของคลาส

ตารางที่ 3.8 รายละเอียดของคลาส Profile

Profile
- UserModel: object - PostModel: object - currentLongtitude: double - currentLatitude: double - uid: String
+ getLocation() + getPostbyId()

ตารางที่ 3.9 รายละเอียดของคลาส Register

Register
- email: String - password: String - uid: String - coin: Double - username: String - UserModel: object
+ handleEmailRegister()

ตารางที่ 3.10 รายละเอียดของคลาส Login

Login
- email - password - uid
+ handleSignIn() + checkUser()

ตารางที่ 3.11 รายละเอียดของคลาส Feed

Feed
- PostModel: object - uid: String - currentLongtitude: double - currentLatitude: double - FindJobClick: bool - HireJobClick: bool
+ getLocation() + getPostData() + getPostByType()

ตารางที่ 3.12 รายละเอียดของคลาส PostList

PostList
- uid: String - PostModel: List - currentLongtitude: double - currentLatitude: double
+ getLocation() + getPostData()

ตารางที่ 3.13 รายละเอียดของคลาส CreatePost

CreatePost
-uid: String -pid: String -title: String -detail: String -location: String -postdate: Timestamp -postImage: String -postby: String -latitude: double -longitude: double -isTake: bool -takeby: String -profileImage: String -isFindJob: bool -pricePay: double -priceBuy: double -totalPrice: double -isGave: bool -isReceived: bool
+getLocation() +createPost() +countUserWithin3KmOfPost() +sendNotificationCreatePost()

ตารางที่ 3.14 รายละเอียดของคลาส PostDeail

PostDeail
- uid: String - PostModel: object - currentLongtitude: double - currentLatitude: double - isowner: bool
+getLocation() +getPostDetail()

ตารางที่ 3.15 รายละเอียดของคลาส TakeDeal

TakeDeal
- uid: String - pid: String - uidPostby: String - PostModel: object
+ takePost() + sendNotificationToUser()

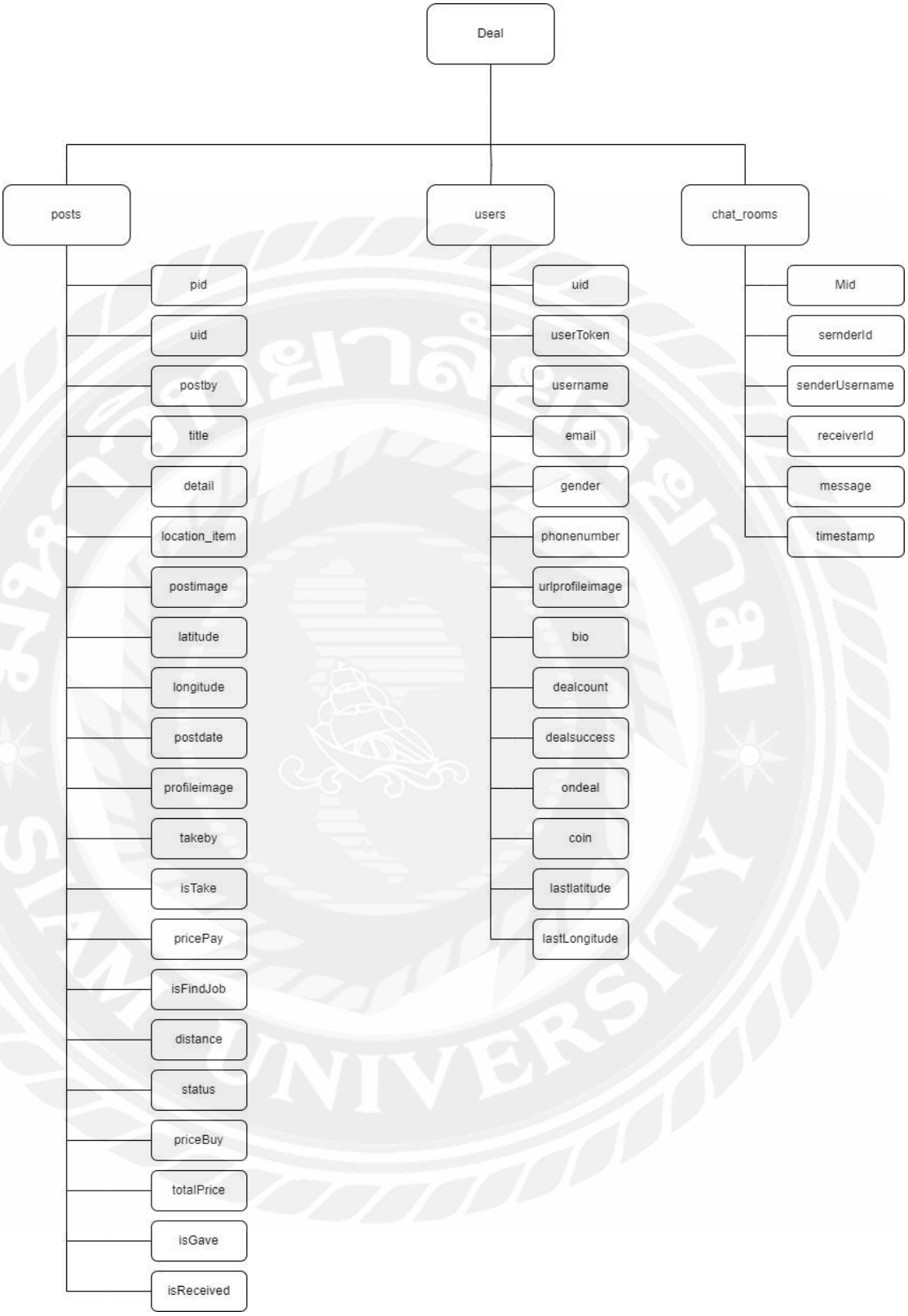
ตารางที่ 3.16 รายละเอียดของคลาส Chat

Chat
-Mid: String -senderId: String -receiverId: String -message: String -timestamp: Timestamp
+getMessages() +sendMessage()

ตารางที่ 3.17 คลาส PushNotificationService

PushNotificationService
- deviceToken: String - messageTitle: String - messageBody: String - endpointFirebaseCloudMessaging: String - message: object
+ sendClickNotificationToUser() + sendMessageNotificationToUser() + sendNotificationCreatePost()

3.7 โครงสร้างของข้อมูล (Data Structure)



รูปที่ 3.10 โครงสร้างของข้อมูล (NoSQL Database Structure)

3.8 รายละเอียดของโครงสร้างข้อมูล (Data Dictionary)

ตารางที่ 3.18 รายละเอียดของข้อมูล posts

Element Name	Description	Data Type
pid	รหัสเฉพาะของโพสต์	String
uid	รหัสเฉพาะของผู้โพสต์	String
postby	ชื่อผู้โพสต์	String
title	หัวข้อโพสต์	String
detail	รายละเอียดโพสต์	String
location_item	ที่อยู่	String
postimage	ที่อยู่ของรูปภาพโพสต์	String
latitude	ที่อยู่ของผู้โพสต์	String
longitude	ที่อยู่ของผู้โพสต์	String
postdate	วันที่สร้างโพสต์	Timestamp
profileImage	ที่อยู่รูปประจำตัวผู้โพสต์	String
takeby	รหัสเฉพาะของผู้รับข้อเสนอโพสต์	String
isTake	สถานะของโพสต์	bool
pricePay	จำนวนเงินค่าจ้าง	double
isFindJob	ประเภทของโพสต์	bool
distance	ระยะห่างระหว่างผู้ใช้กับเจ้าของโพสต์	double
status	ความคืบหน้าของโพสต์	String
priceBuy	ราคาสินค้าที่ต้องการซื้อ	double
totalPrice	ราคารวมค่าจ้างและราคาสินค้า	double
isGave	เช็คสถานะส่งของ	bool
isReceived	เช็คสถานะได้รับของ	bool

ตารางที่ 3.19 รายละเอียดของข้อมูล users

Element Name	Description	Data Type
uid	รหัสเฉพาะของผู้ใช้	String
userToken	รหัสเครื่องของผู้ใช้	String
email	อีเมลผู้ใช้	String
gender	เพศ	Array
phonenumber	หมายเลขโทรศัพท์	String
urlprofileimage	ที่อยู่รูปภาพประจำตัวผู้ใช้	String
bio	ประวัติ	String
dealcount	จำนวนการดีล	int
dealsuccess	จำนวนดีลที่สำเร็จ	int
ondeal	จำนวนที่อยู่ระหว่างดีล	int
coin	จำนวนเหรียญ	double
lastLatitude	ที่อยู่ล่าสุด	double
lastLongitude	ที่อยู่ล่าสุด	double

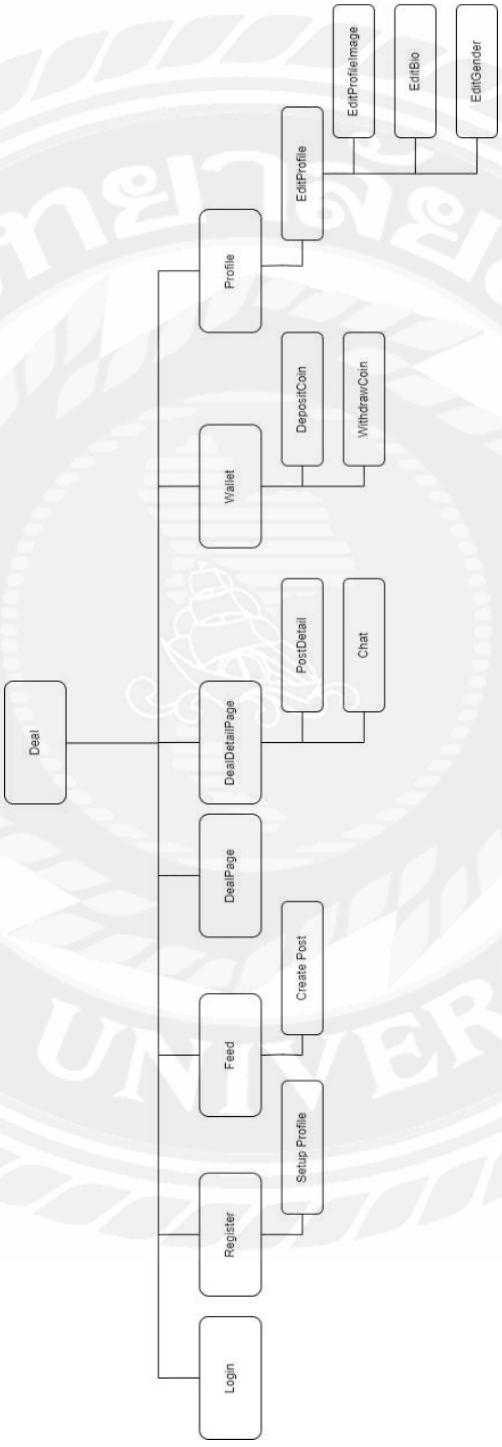
ตารางที่ 3.20 รายละเอียดของข้อมูล Chat_rooms

Element Name	Description	Data Type
Mid	รหัสเฉพาะของโพสต์	String
senderId	รหัสเฉพาะของผู้ส่ง	String
senderUsername	ชื่อผู้ใช้ของผู้ส่ง	String
receiverId	รหัสเฉพาะผู้รับ	String
message	ข้อความ	String
timestamp	เวลาส่งข้อความ	Timestamp

บทที่ 4

การออกแบบทางกายภาพ

4.1 โครงสร้างของแอปพลิเคชัน (Application Structure Map)



รูปที่ 4.1 โครงสร้างของแอปพลิเคชัน Deal

ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Deal

หน้าจอ	คำอธิบายรายละเอียด
Login (Sign_in.dart)	การเข้าสู่ระบบเพื่อยืนยันตัวตน
Register (register.dart)	การสมัครใช้งานแอปพลิเคชัน
Setup Profile	การสมัครใช้งานแอปพลิเคชัน จะต้องกรอกข้อมูล - ชื่อผู้ใช้ - อีเมล - รหัสผ่าน
Feed (Feed_pagfe.dart)	แสดงโพสต์ทั้งหมดที่ห่างจากตัวผู้ใช้ในระยะใกล้
CreatePost (createPost_page.dart)	การสร้างโพสต์ โดยต้องระบุข้อมูลดังนี้ - ประเภทโพสต์ - หัวข้อโพสต์ - รายละเอียดโพส - ที่อยู่ - ราคาจ้าง - ราคาสินค้า - รูปภาพ
DealPage (deal_page.dart)	แสดงโพสต์ทั้งหมดที่ผู้ใช้เป็นผู้ว่าจ้างหรือรับจ้าง
postDetailPage (postDetail_page.dart)	แสดงรายละเอียดของโพสต์ หากผู้ใช้เป็นผู้ว่าจ้างหรือรับจ้าง สามารถพูดคุยได้โดยคลิกที่ปุ่ม Chat
Wallet (WalletPage.dart)	แสดงชื่อผู้ใช้ และ จำนวน Coin ของผู้ใช้
DepositCoin (DepositPage.dart)	การเพิ่ม Coin ต้องระบุข้อมูลดังนี้ - เลขบัตร - วันหมดอายุบัตร - เลขหลังบัตร (CVV) - ชื่อบนบัตร - จำนวนเงิน
WithdrawCoin (Withdraw.dart)	การถอนเงิน ต้องระบุข้อมูลดังนี้ - เลขบัตร - วันหมดอายุบัตร

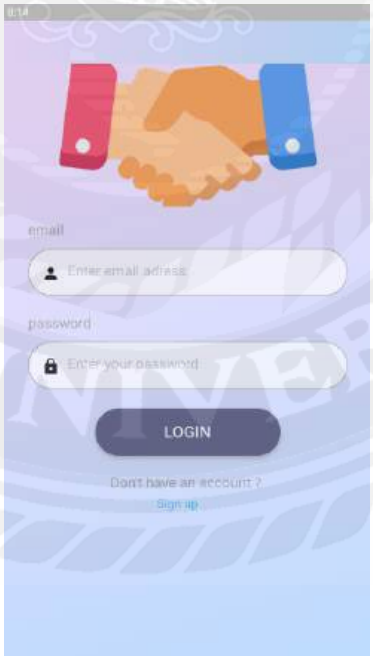
ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างแอปพลิเคชัน Deal (ต่อ)

หน้าจอ	คำอธิบายรายละเอียด
	- เลขหลังบัตร (CVV) - ชื่อบนบัตร - จำนวนเงิน
Profile (profile.dart)	แสดงข้อมูลของผู้ใช้ และ แสดงโพสต์ของผู้ใช้
EditProfile (editprofile_page.dart)	แสดงข้อมูลผู้ใช้
EditProfileImage (editprofile_image.dart)	แก้ไขรูปประจำตัวผู้ใช้
EditBio (editbio_page.dart)	แก้ไขประวัติผู้ใช้
EditGender (editgender_page.dart)	แก้ไขเพศผู้ใช้
Chat (chat_page.dart)	ระบบการส่งข้อความถึงผู้ใช้คนอื่น โดยจะแสดงผลข้อความที่ส่ง และข้อความที่ผู้ใช้ตอบกลับ เป็นการแสดงผลแบบเรียลไทม์ (Real Time)
Settings (settingscreen.dart)	แสดงรายการตั้งค่าสำหรับผู้ใช้โดยมีการตั้งค่าดังนี้ - การตั้งค่าธีมสีในระบบ

4.2 การออกแบบส่วนติดต่อกับผู้ใช้ (User Interface Design)

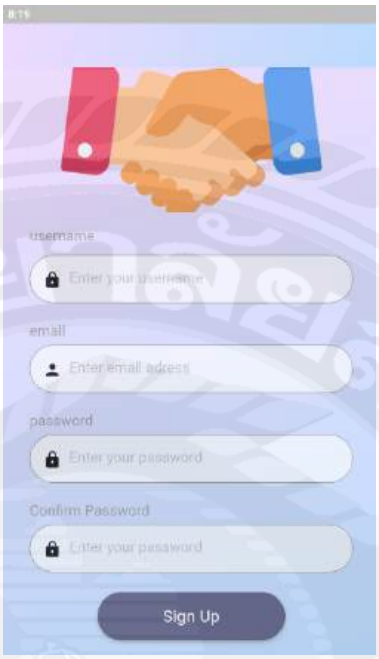
แอปพลิเคชัน Deal ทำหน้าที่นำผู้ที่ต้องการฝากซื้อสินค้าและผู้รับฝากซื้อสินค้ามาพบกัน ในการออกแบบส่วนติดต่อกับผู้ใช้ (User Interface หรือ UI) นั้นจะยึดหลักการของ UX/UI เป็นหลัก โดยคำนึงถึงความพึงพอใจและการใช้งานที่ง่ายต่อผู้ใช้ ด้วยการออกแบบที่เรียบง่าย ผู้ใช้สามารถเรียนรู้การใช้งานระบบได้ด้วยตนเอง การจัดวางตัวอักษร และการจัดวางตำแหน่งไอคอน ให้ผู้ใช้ทำความเข้าใจง่ายที่สุด โดยมีรายละเอียดดังนี้

รูปที่ 4.2 ไอคอนของแอปพลิเคชัน Deal



รูปที่ 4.3 หน้าเข้าสู่ระบบ (Login)

จากรูปที่ 4.3 หน้าเข้าสู่ระบบ (Login) ในหน้าจอนี้จะแสดงโลโก้ของแอปพลิเคชัน และช่องทางการเข้าสู่ระบบเพื่อใช้งานแอปพลิเคชัน ถ้าเป็นผู้ใช้ที่เคยลงทะเบียนในระบบแล้ว จะทำการเข้าสู่ระบบให้โดยอัตโนมัติ

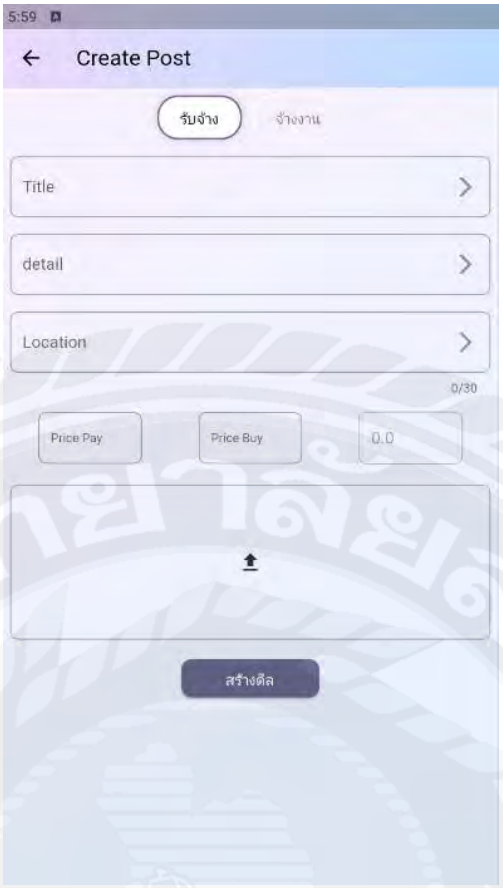


รูปที่ 4.4 หน้าลงทะเบียนเข้าใช้งานแอปพลิเคชัน Deal (Register)

จากรูปที่ 4.4 หน้าลงทะเบียนเข้าใช้งานสำหรับผู้ใช้รายใหม่ เพื่อเป็นข้อมูลของผู้ใช้สำหรับยืนยันตัวตนของผู้ใช้ในการเข้าใช้งานระบบ โดยจะมีช่องสำหรับกรอกข้อมูลผู้ใช้ ดังนี้ ชื่อผู้ใช้ อีเมล และปุ่ม Sign Up สำหรับการเข้าสู่ขั้นตอนการ Login ถัดไป

รูปที่ 4.5 หน้า Feed Post

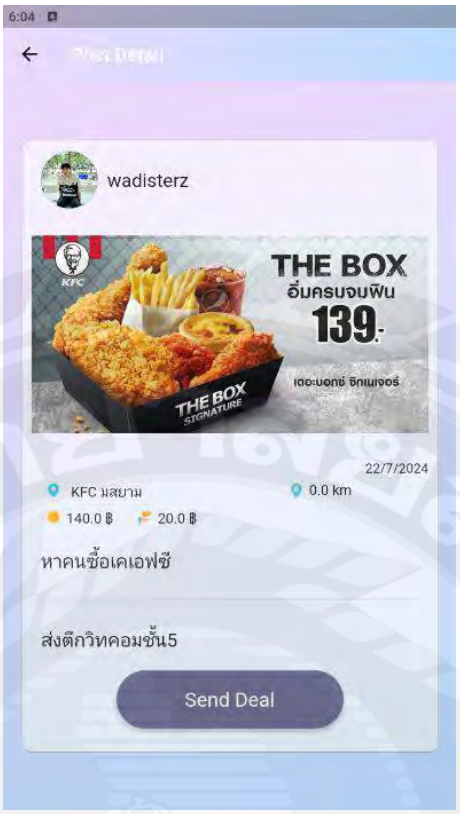
จากรูปที่ 4.5 หน้า Feed Post เป็นหน้าที่แสดงข้อมูลของโพสต์ในระยะใกล้ โดยจะเป็นโพสต์ฝากซื้อสินค้าหรือรับฝากซื้อสินค้าที่อยู่ในระยะ 3 กิโลเมตรจากตำแหน่งปัจจุบันของผู้ใช้ โดยจะแสดงรายละเอียดของผู้โพสต์ ระยะห่างระหว่างผู้โพสต์กับผู้ใช้ รูปภาพของผู้โพสต์ วันเวลาโพสต์ จำนวนค่าจ้าง และมีปุ่ม Create Post ไว้สำหรับสร้างโพสต์ใหม่ได้



รูปที่ 4.6 หน้า Create Post

จากรูปที่ 4.6 หน้า Create Post นี้เป็นส่วนหนึ่งของแอปพลิเคชันที่ใช้ในการสร้างโพสต์ใหม่ โดยมีส่วนประกอบต่างๆ ดังนี้:

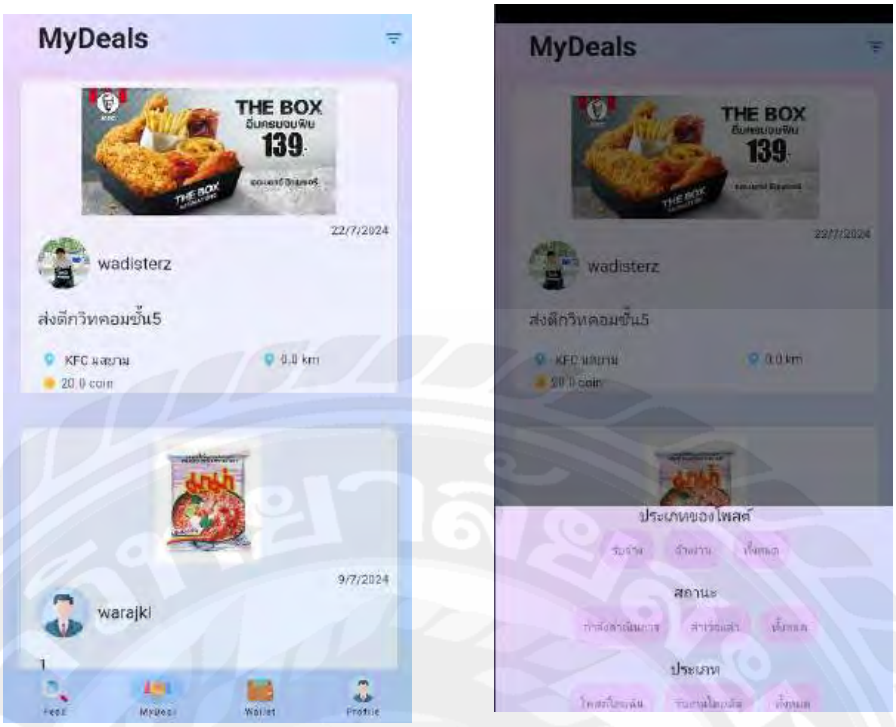
- ปุ่มสลับระหว่าง "รับจ้าง" และ "จ้างงาน": อยู่ด้านบนของหน้าจอ ช่วยให้ผู้ใช้เลือกประเภทของ โพสต์ว่าจะเป็นการรับจ้างหรือการจ้างงาน
- Title: ช่องสำหรับกรอกชื่อหรือหัวข้อของโพสต์
- Detail: ช่องสำหรับกรอกรายละเอียดของโพสต์
- Location: ช่องสำหรับกรอกสถานที่ที่เกี่ยวข้องกับโพสต์
- Price Pay: ช่องสำหรับกรอกราคาที่ผู้จ้างงานต้องการจ่าย
- Price Buy: ช่องสำหรับกรอกราคาที่ผู้รับจ้างต้องการได้รับ
- 0.0: เป็นการแสดงผลราคาที่คิดรวม
- ปุ่มอัพโหลดไฟล์ (รูปไอคอนลูกศรชี้ขึ้น): สำหรับอัพโหลดไฟล์หรือรูปภาพที่เกี่ยวข้องกับโพสต์ ปุ่มสร้างโพสต์ (สร้างดีล): ปุ่มสำหรับสร้างโพสต์หลังจากกรอกข้อมูลทั้งหมดเสร็จสิ้น



รูปที่ 4.7 หน้า Post Detail

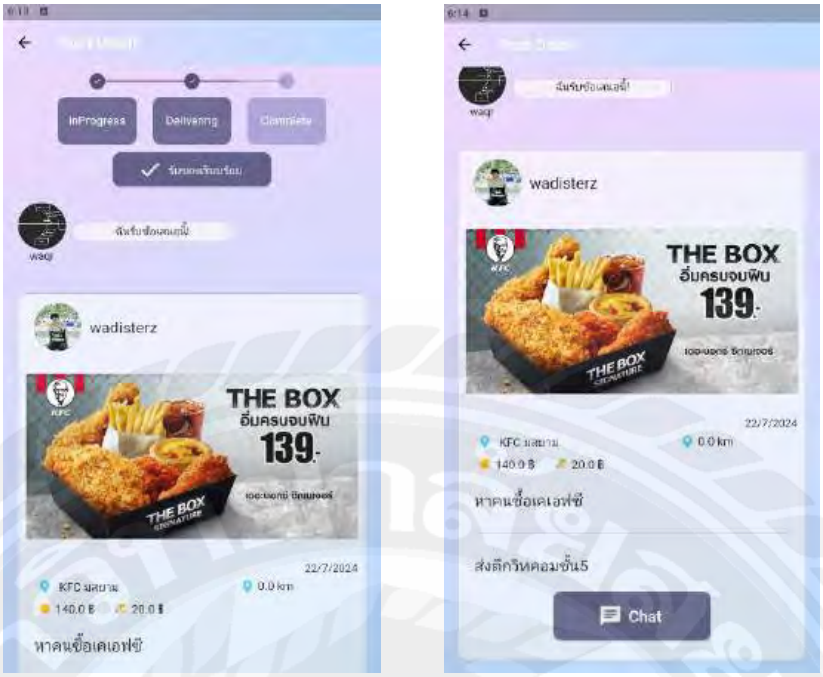
จากรูปที่ 4.7 หน้าจอ Post Detail นี้เป็นการแสดงรายละเอียดของโพสต์ที่ถูกสร้างขึ้น โดยมีส่วนประกอบต่างๆ ดังนี้:

- โปรไฟล์ผู้โพสต์: ประกอบด้วยรูปโปรไฟล์และชื่อผู้โพสต์ (ในกรณีนี้คือ wadisterz)
- รูปภาพประกอบโพสต์: รูปภาพของสินค้าหรือบริการที่เกี่ยวข้องกับโพสต์ (ในที่นี้คือภาพจาก KFC ที่แสดงถึง The Box ราคา 139 บาท)
- สถานที่: ระบุที่ตั้งของสถานที่เกี่ยวข้องกับโพสต์ (ในกรณีนี้คือ KFC สยาม)
- ราคา: แสดงราคาสินค้าหรือบริการ (140 บาท) และค่าบริการเพิ่มเติม (20 บาท)
- หัวข้อโพสต์ : แสดงหัวข้อโพสต์
- รายละเอียดโพสต์ : แสดงรายละเอียดเพิ่มเติมของโพสต์
- ระยะทาง: แสดงระยะทางจากตำแหน่งปัจจุบัน (0.0 กม.)
- วันที่: วันที่โพสต์ถูกสร้างขึ้น (22/7/2024)
- ปุ่ม Send Deal: ปุ่มสำหรับตกลงรับงานตามรายละเอียดของโพสต์



รูปที่ 4.8 หน้า MyDeal

จากรูปที่ 4.8 หน้าจอ MyDeals นี้แสดงรายการดีลที่ผู้ใช้เกี่ยวข้อง ด้านขวาบนสามารถกรองการแสดงผลของรายการดีลได้

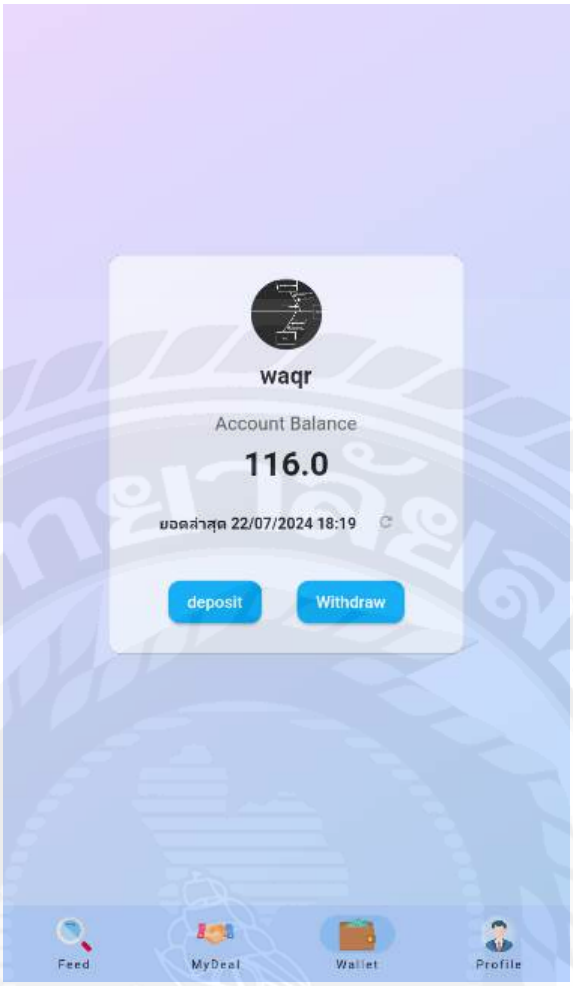


รูปที่ 4.9 หน้า Post Detail หลังจากมีผู้ตอบตกลงรับฝากซื้อสินค้า

จากรูปที่ 4.10 หน้า Post Detail หลังจากมีผู้ตกลงรับฝากซื้อสินค้าแล้ว ด้านบนจะแสดง Timeline ของโพสต์หากอีกฝ่ายทำการกดปุ่ม Timeline จะเกิดการอัปเดต ด้านล่างผู้ใช้สามารถพูดคุยรายละเอียดโพสต์กับอีกฝ่ายได้โดยการคลิกที่ปุ่ม Chat

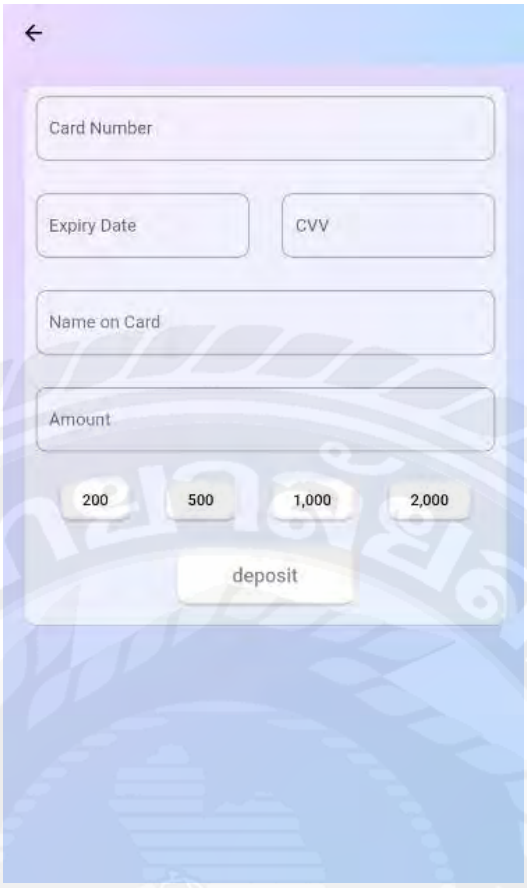
รูปที่ 4.10 หน้า Chat

จากรูปที่ 4.10 หน้า Chat เป็นหน้าจอสำหรับให้ผู้ใช้ส่งข้อความถึงอีกฝ่ายโดยฝั่งด้านซ้ายจะแสดงข้อความของผู้ใช้คนที่กำลังคุยด้วย และฝั่งขวาเป็นข้อความที่ผู้ใช้เป็นคนพิมพ์ข้อความส่งไป เมื่อพิมพ์ข้อความในช่องกรอกข้อความเสร็จแล้ว ต้องทำการกดปุ่ม Send ที่อยู่ทางด้านล่างขวาเพื่อเป็นการยืนยันการส่งข้อความ



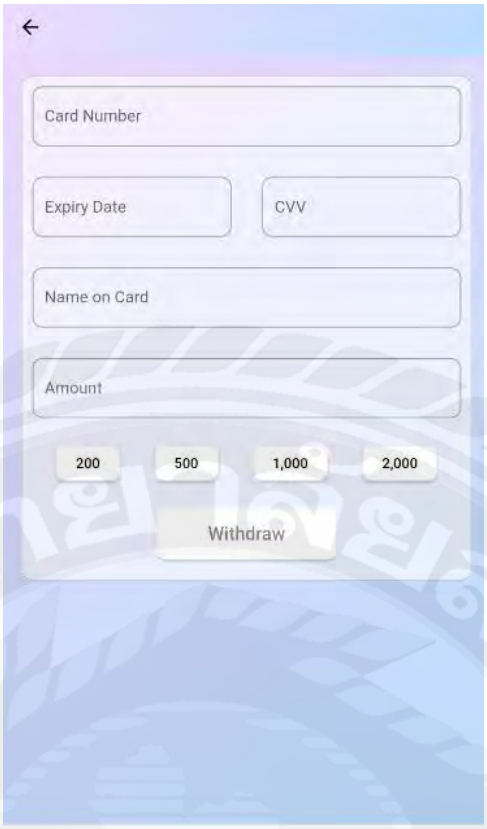
รูปที่ 4.11 หน้า Wallet

จากรูปที่ 4.11 หน้า Wallet แสดงจำนวนยอด Coin ในบัญชีของผู้ใช้



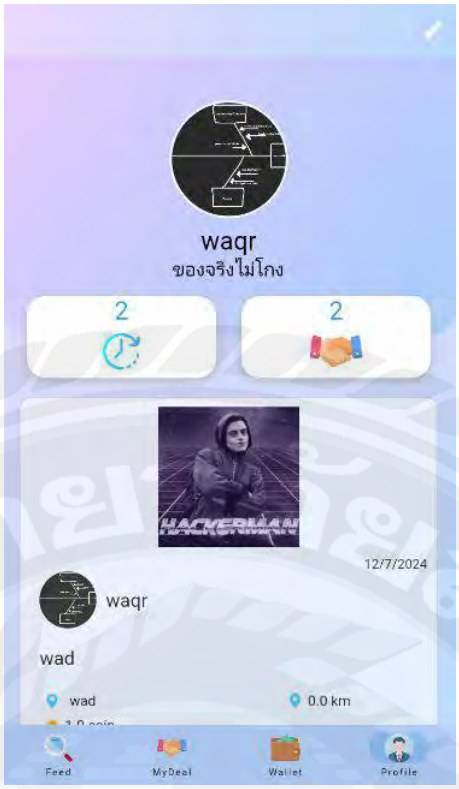
รูปที่ 4.12 หน้า Deposit Coin

จากรูปที่ 4.12 หน้า Deposit Coin เป็นหน้าสำหรับฝากเงินเข้าสู่ระบบ โดยให้ผู้ใช้กรอกข้อมูลรายละเอียดบัตร และ จำนวนเงินที่ต้องการฝากเข้าสู่ระบบ



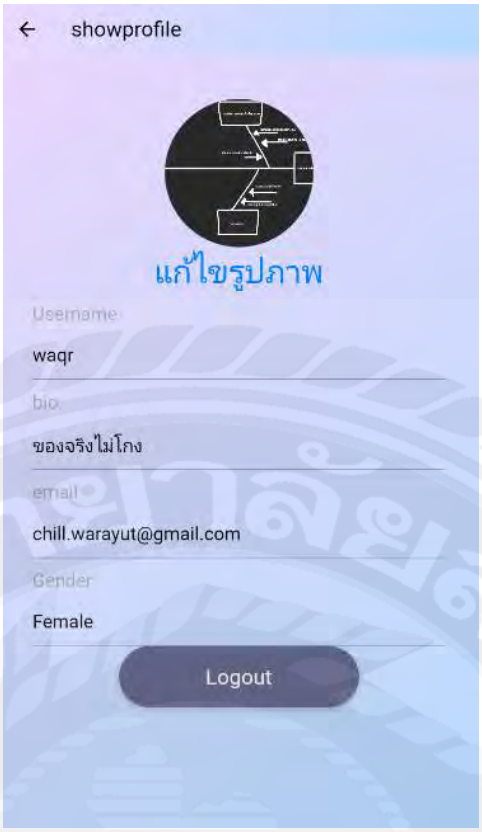
รูปที่ 4.13 หน้า Withdraw Coin

จากรูปที่ 4.13 หน้า Chat เป็นหน้าสำหรับถอนเงินออกจากระบบ โดยให้ผู้ใช้กรอกข้อมูลรายละเอียดบัตร และ จำนวนเงินที่ต้องการถอน



รูปที่ 4.14 หน้า Profile

จากรูปที่ 4.14 หน้า Profile แสดงข้อมูลผู้ใช้ และ โพสต์ที่ผู้ใช้เคยโพสต์ รวมทั้งประวัติการดีล สำเร็จ และ จำนวนที่อยู่ระหว่างดีล โดยขวาบนสามารถแก้ไขโปรไฟล์ได้



รูปที่ 4.15 หน้า Edit Profile

จากรูปที่ 4.15 หน้า Edit Profile เป็นหน้าจอสำหรับให้ผู้ใช้แก้ไขข้อมูลส่วนตัว โดยจะสามารถแก้ไข รูปภาพ, bio, Gender ได้ และ ยังเป็นหน้าสำหรับให้ผู้ใช้ออกจากระบบโดยคลิกที่ปุ่ม Logout

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลปริญญานิพนธ์

การพัฒนาระบบสังคมออนไลน์สำหรับการรับฝากซื้อสินค้าในระยะใกล้ หรือแอปพลิเคชัน Deal มีวัตถุประสงค์เพื่อสร้างแพลตฟอร์มที่ช่วยให้ผู้ใช้สามารถฝากซื้อสินค้าจากคนที่อยู่ใกล้เคียงได้อย่างสะดวกและรวดเร็ว แอปพลิเคชันนี้ถูกออกแบบมาเพื่อตอบสนองความต้องการของผู้ที่ไม่มีเวลาเดินทางไปซื้อสินค้าด้วยตนเอง โดยผู้ใช้สามารถลงทะเบียนผ่านอีเมลและเข้าสู่ระบบได้อย่างง่ายดาย ข้อมูลที่อยู่ของผู้ใช้ถูกเก็บรักษาใน Firebase เพื่อใช้ในการคำนวณระยะห่างและแสดงโพสต์ที่ใกล้เคียง การใช้ Flutter Framework ในการสร้าง Native Interface สำหรับแอนดรอยด์ช่วยให้แอปพลิเคชันมีประสิทธิภาพสูงและใช้งานได้ง่าย ระบบการแจ้งเตือนช่วยให้ผู้ใช้ได้รับข้อมูลทันทีเมื่อมีผู้โพสต์ใหม่หรือเมื่อมีการตอบกลับจากโพสต์ของตนเอง เพิ่มระบบแชทเฉพาะโพสต์นั้นๆ เพื่อให้ผู้ซื้อและผู้ขายสามารถติดต่อสื่อสารกันได้โดยตรงเกี่ยวกับรายละเอียดของสินค้าและการจัดส่ง

5.2 ข้อดีของระบบ

5.2.1 ผู้ใช้สามารถฝากซื้อสินค้าและบริการจากผู้ที่อยู่ใกล้เคียงได้โดยไม่ต้องเดินทางเอง ลดเวลาในการเดินทางและการหาสินค้า

5.2.2 การแจ้งเตือนแบบเรียลไทม์ช่วยให้ผู้ใช้ไม่พลาดข้อมูลสำคัญเกี่ยวกับโพสต์ใหม่และการตอบกลับ

5.2.3 ระบบแชทเฉพาะโพสต์ช่วยให้ผู้ซื้อและผู้ขายสามารถสื่อสารกันได้โดยตรง เพิ่มความชัดเจนในการตกลงรายละเอียดของสินค้าและการจัดส่ง·

5.2.4 การพัฒนาด้วยเทคโนโลยีที่ทันสมัยทำให้แอปพลิเคชันสามารถปรับปรุงและเพิ่มฟีเจอร์ใหม่ได้ง่ายตามความต้องการของผู้ใช้ในอนาคต

5.3 ข้อเสนอแนะ

เพื่อให้ระบบมีประสิทธิภาพมากยิ่งขึ้น ควรมีการพัฒนาฟังชันการทำงานเพิ่มเติม ดังนี้

5.3.1 ควรเพิ่มระบบรีวิวและการให้คะแนนสำหรับผู้ว่าจ้างและผู้รับจ้าง เพื่อให้ผู้ใช้สามารถดูความคิดเห็นและคะแนนของผู้ว่าจ้างหรือผู้รับจ้างอื่นๆ ซึ่งจะช่วยเพิ่มความน่าเชื่อถือและความมั่นใจในการทำธุรกรรม

5.3.2 ควรพัฒนาแอปพลิเคชันให้สามารถใช้งานได้บนระบบ iOS เพื่อขยายฐานผู้ใช้และเพิ่มความสะดวกในการเข้าถึงแอปพลิเคชัน

5.3.3 เพิ่มตัวเลือกการแจ้งเตือนสำหรับประเภทสินค้าที่ผู้ใช้สนใจเป็นพิเศษ เพื่อให้ผู้ใช้ได้รับการแจ้งเตือนเฉพาะสินค้าที่ตรงกับความต้องการ

5.3.4 ปรับปรุงระบบค้นหาให้มีความแม่นยำและรวดเร็วยิ่งขึ้น โดยการเพิ่มฟิลเตอร์และตัวเลือกการค้นหาที่หลากหลาย เช่น การค้นหาตามหมวดหมู่ ราคา หรือระยะทาง

5.3.5 พัฒนาระบบยืนยันตัวตนผู้ใช้ที่มีความปลอดภัยสูงขึ้น เช่น การยืนยันตัวตนด้วยบัตรประชาชน หรือการใช้เทคโนโลยีการตรวจสอบใบหน้า เพื่อเพิ่มความปลอดภัยและลดความเสี่ยงจากการฉ้อโกง

บรรณานุกรม

ทีเอมีโม. (2563, 4 มกราคม). *Dart 101: ทำความรู้จักภาษา Dart ฉบับโปรแกรมเมอร์*. [เว็บบล็อก]. https://www.tamemo.com/post/172/dart-101-intro/

มีเดียม. (2561, 22 มิถุนายน). *มาทำความรู้จักกับ Flutter กันเถอะ*. [เว็บบล็อก]. https://medium.com/@hizokaz/มาทำความรู้จักกับ-flutter-กันเถอะ-4dca2ad634bd

มีเดียม. (2561, 20 กุมภาพันธ์). *สรุปพื้นฐานภาษา Dart ก่อนเขียน Flutter*. [เว็บบล็อก]. https://medium.com/@codingthailand/สรุปพื้นฐานภาษา-dart-ก่อนเขียน-flutter-db80b637e207

มีเดียม. (2561, 20 กุมภาพันธ์). *Android Studio (Lab 3SB04)*. [เว็บบล็อก]. https://medium.com/@boyreallife/android-studio-lab-3sb04-4202ceff7167

วิกิพีเดีย. (น.ป.ป). *ฟลัตเตอร์*. วันที่สืบค้น 3 มีนาคม 2567, จาก https://th.wikipedia.org/ฟลัตเตอร์

วิกิพีเดีย. (น.ป.ป). *ภาษาดาร์ต*. วันที่สืบค้น 3 มีนาคม 2567, จาก https://th.wikipedia.org/ภาษาดาร์ต

เอดับเบิลยูเอส. (2566). *Flutter คืออะไร*. https://aws.amazon.com/th/what-is/flutter/

แอปมาสเตอร์. (2563, 25 กุมภาพันธ์). *Firebase คืออะไร*. [เว็บบล็อก]. https://appmaster.io/th/blog/firebase-khuue-aair