

การพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง
Developing an Application for Finding Homes for Pets

นายธัช	แก้วกุลธรรม	6104800014
นายพิภพ	ชูเกียรติพัลลภ	6104800040

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์
มหาวิทยาลัยสยาม
ปีการศึกษา 2564

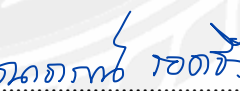
หัวข้อปริญญานิพนธ์	การพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง		
	Developing an Application for Finding Homes for Pets		
หน่วยกิตของปริญญานิพนธ์	3 หน่วยกิต		
รายชื่อคณะผู้จัดทำ	นายธัช	แก้วกุลธรรม	6104800014
	นายพิภพ	ชูเกียรติพัลลภ	6104800023
อาจารย์ที่ปรึกษา	อาจารย์ชานาภรณ์	รอดชีวิต	
ระดับการศึกษา	วิทยาศาสตรบัณฑิต		
ภาควิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2564		

อนุมัติให้ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์

คณะกรรมการสอบปริญญานิพนธ์


.....ประธานกรรมการ
(พล.อ.ท.ยศ.ดร. พาหรัณ สงวนโกศล)


.....กรรมการ
(อาจารย์เอก บำรุงศรี)


.....อาจารย์ที่ปรึกษา
(อาจารย์ชานาภรณ์ รอดชีวิต)

หัวข้อปริญญาโท	การพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง		
หน่วยกิตของปริญญาโท	3 หน่วยกิต		
รายชื่อคณะผู้จัดทำ	นายธัช	แก้วกุลธรรม	6104800014
	นายพิภพ	ชูเกียรติพัลลภ	6104800023
อาจารย์ที่ปรึกษา	อาจารย์ธนาภรณ์	รอดชีวิต	
ระดับการศึกษา	วิทยาศาสตรบัณฑิต		
ภาควิชา	วิทยาการคอมพิวเตอร์		
ปีการศึกษา	2564		

บทคัดย่อ

วัตถุประสงค์ในการจัดทำปริญญาโทนี้เพื่อพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง เพื่อเป็นช่องทางหนึ่งในการช่วยลดปัญหาสัตว์เลี้ยงจรจัดที่ไร้บ้าน เป็นการสร้างพื้นที่สำหรับการร่วมมือ โดยมีผู้เลี้ยงสัตว์ที่ไม่สามารถที่จะเลี้ยงสัตว์ต่อได้ หรือพบเจอสัตว์เลี้ยงที่ต่าง ๆ แต่ไม่สามารถเลี้ยงเองได้ แต่อยากช่วยเหลือให้สัตว์เลี้ยงจรจัดให้มีเจ้าของดูแลและมีบ้านอยู่อาศัย ดังนั้นทางคณะผู้จัดทำจึงได้พัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยงทำงานได้ทั้งบนระบบปฏิบัติการแอนดรอยด์และไอโอเอส โดยผู้ใช้สามารถโพสต์ข้อมูลของสัตว์ไม่ว่าจะเป็นสัตว์เลี้ยงจรจัดหรือมีเจ้าของ คลิกถูกใจโพสต์ที่สนใจ และสามารถรับดูแลสัตว์เลี้ยงที่ผู้ใช้ต้องการรับไปดูแลได้โดยมีระบบแชทที่ให้ผู้ใช้งานพูดคุยกับเจ้าของโพสต์ก่อน เจ้าของโพสต์จะสามารถอนุมัติหรือปฏิเสธผู้ที่มาขอรับดูแลได้ โดยทางระบบจะมีระบบแจ้งเตือน 7 วันเพื่อให้ผู้รับเลี้ยงส่งความคืบหน้าสัตว์เลี้ยงที่รับไปเลี้ยงด้วยเพื่อป้องกันการนำสัตว์ไปทารุณกรรม และสามารถดูข้อมูลส่วนตัวของตนเองได้ ในการออกแบบส่วนติดต่อกับผู้ใช้ได้ออกแบบตามหลักการของ UX และ UI และการพัฒนาระบบได้ประยุกต์ใช้เทคโนโลยีครอสแพลตฟอร์ม และระบบจัดการฐานข้อมูลแบบ NoSQL เครื่องมือที่ใช้ในการพัฒนา ได้แก่ โปรแกรม Visual Studio Code Android Studio Flutter Framework และเขียนชุดคำสั่งด้วยภาษา Dart จัดการฐานข้อมูลด้วย Firebase โดยระบบจะสามารถช่วยเป็นสื่อกลางให้สัตว์เลี้ยงจรจัด หรือผู้ที่เลี้ยงสัตว์เลี้ยงไม่ไหวได้มีโอกาสหาที่อยู่อาศัยใหม่ได้

คำสำคัญ: แอปพลิเคชัน, สื่อกลาง, สัตว์เลี้ยง, สัตว์จรจัด

Project title	Developing an Application for Finding Homes for Pets		
Project credits	3 Units		
By	Mr. Thanat	Kaewkultham	6104800014
	Mt. Pipob	Chukiatpunlob	6104800023
Advisor	Miss Thanaporn	Rodcheewit	
Degree	Bachelor of Science		
Major	Computer Science		
Faculty	Science		
Academic year	2021		

Abstract

The development of a program to help pets find homes was the goal of this thesis and to serve as a vehicle for collaborating with herders that are unable to continue rearing the animals, and assisting in the reduction of homeless and stray animals, or people that encounter stray animals in many locations but are unable to raise their own. They also would like to assist stray animals in finding owners who will care for them and provide them with a place to live. The developers created a pet adoption software that can be used on both iOS and Android devices. Regardless whether an animal is owned or not, users can share information about it like articles that they find interesting. It has a chat feature that enables people to speak with the post's owner before adopting the animals. Owners of posts will have the option to accept or reject moderators. To prevent animal cruelty, the system sends a 7-day message asking the adopter to report progress on the adopted animal, and users may access their own personal data. Cross-platform technology was used in the system development, and the user interface was created using the principles of UX and UI. A NoSQL database management system was also used. Virtual Studio Code, Android Studio, the Flutter Framework, and Dart language are examples of development tools, and utilized Firebase to manage databases. The system can be useful for stray animals as a conduit, or those who are unable to care for pets might try to locate a new home.

Keywords: applications, intermediaries, pets, stray animals

Approved by

.....

Approved by

.....

กิตติกรรมประกาศ (Acknowledgment)

การจัดทำวิทยานิพนธ์ฉบับนี้สำเร็จได้นั้น คณะผู้จัดทำได้รับความกรุณาจาก อาจารย์ผู้สอนทุกท่านที่ให้ข้อมูลต่าง ๆ ส่งผลให้คณะผู้จัดทำได้รับความรู้และประสบการณ์ต่าง ๆ ที่มีค่ามากมายสำหรับวิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีจากความร่วมมือและสนับสนุนจากหลายฝ่ายดังนี้

1. อาจารย์ ชนาภรณ์ รอดชีวิต อาจารย์ที่ปรึกษา

คณะผู้จัดทำใคร่ขอขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ ที่ได้ให้คำแนะนำสำคัญเพื่อให้การสอบวิทยานิพนธ์ฉบับนี้ และผู้มีส่วนร่วมทุกท่าน รวมทั้งผู้ที่ไม่ได้กล่าวนาม ที่มีส่วนร่วมในการให้ข้อมูลให้ความช่วยเหลือ และเป็นທີ່ปรึกษาให้คำแนะนำต่าง ๆ จนทำให้งานทุกอย่างประสบความสำเร็จไปด้วยดี และทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ซึ่งคณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

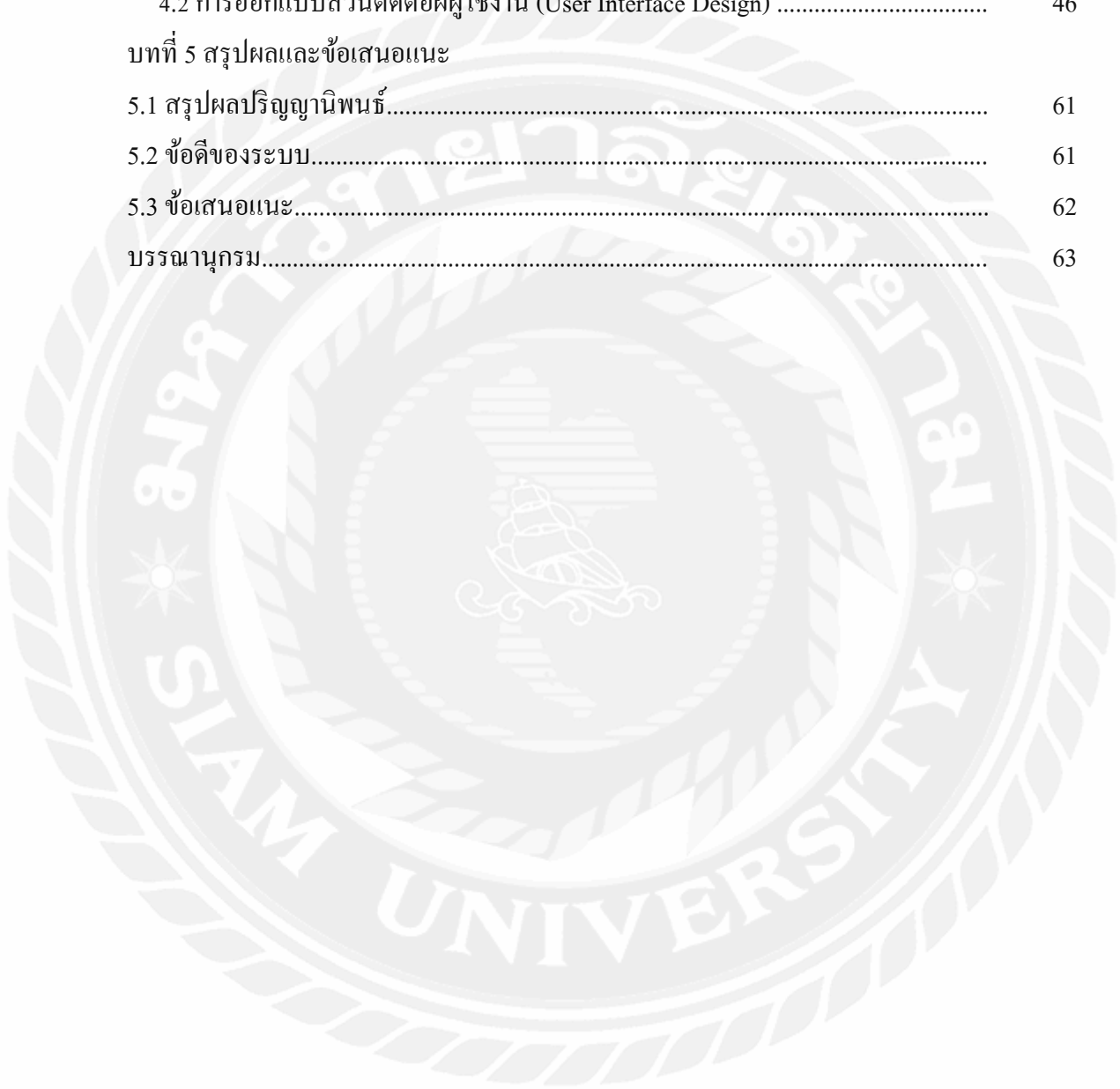
นาย ธนัช แก้วกุลธรรม
นาย พิภพ ชูเกียรติพัลลภ

สารบัญ

	หน้า
บทคัดย่อ.....	ก
Abstract.....	ข
กิตติกรรมประกาศ.....	ค
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์.....	1
1.3 ขอบเขตของปริญญาานิพนธ์.....	1
1.4 ประโยชน์ที่ได้รับ.....	2
1.5 ขั้นตอนและวิธีดำเนินงานปริญญาานิพนธ์.....	3
1.6 แผนและระยะเวลาในการดำเนินปริญญาานิพนธ์.....	5
1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนาระบบ.....	5
1.8 อุปกรณ์และเครื่องมือที่รองรับ.....	6
บทที่ 2 การทบทวนเอกสารและวรรณกรรมที่เกี่ยวข้อง	
2.1 Flutter.....	7
2.2 Dart.....	8
2.3 Visual Studio.....	9
2.4 Android Studio.....	10
2.5 Figma.....	11
2.6 Postman.....	12
2.7 Firebase.....	13
2.8 Node.js.....	14
บทที่ 3 การวิเคราะห์และออกแบบระบบ	
3.1 รายละเอียดของปริญญาานิพนธ์.....	15
3.2 แสดงฟังก์ชันการทำงานของระบบด้วย Use Case Diagram.....	16
3.3 คำอธิบายรายละเอียดของยูสเคส.....	17
3.4 Sequence Diagram.....	26
3.5 Class Diagram.....	33
3.6 โครงสร้างของฐานข้อมูลใน NoSQL Database.....	35

สารบัญ (ต่อ)

	หน้า
บทที่ 4 การออกแบบทางกายภาพ	
4.1 โครงสร้างของแอปพลิเคชัน (Application Structure Map)	44
4.2 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface Design)	46
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 สรุปผลปริญญานิพนธ์.....	61
5.2 ข้อดีของระบบ.....	61
5.3 ข้อเสนอแนะ.....	62
บรรณานุกรม.....	63



สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนและระยะเวลาในการดำเนินงาน.....	5
ตารางที่ 3.1 แสดงรายละเอียดของ Use case: Login Facebook.....	17
ตารางที่ 3.2 แสดงรายละเอียดของ Use case: Login Google.....	18
ตารางที่ 3.3 แสดงรายละเอียดของ Use case: Register.....	19
ตารางที่ 3.4 แสดงรายละเอียดของ Use case: View All Pet.....	19
ตารางที่ 3.5 แสดงรายละเอียดของ Use case: View Detail Pet.....	20
ตารางที่ 3.6 แสดงรายละเอียดของ Use case: Favorite Pet.....	20
ตารางที่ 3.7 แสดงรายละเอียดของ Use case: Location Pet.....	21
ตารางที่ 3.8 แสดงรายละเอียดของ Use case: Post Pet.....	21
ตารางที่ 3.9 แสดงรายละเอียดของ Use case: Adopt Pet.....	22
ตารางที่ 3.10 แสดงรายละเอียดของ Use case: Chat Contact.....	22
ตารางที่ 3.11 แสดงรายละเอียดของ Use case: Approve.....	23
ตารางที่ 3.12 แสดงรายละเอียดของ Use case: Profile.....	23
ตารางที่ 3.13 แสดงรายละเอียดของ Use case: Edit Profile.....	24
ตารางที่ 3.14 แสดงรายละเอียดของ Use case: Logout.....	24
ตารางที่ 3.15 แสดงรายละเอียดของ Use case: Background Notification.....	25
ตารางที่ 3.16 รายละเอียดของ โครงสร้างของข้อมูล: Users.....	41
ตารางที่ 3.17 รายละเอียดของ โครงสร้างของข้อมูล: Pets.....	42
ตารางที่ 3.18 รายละเอียดของ โครงสร้างของข้อมูล: Adopt.....	43
ตารางที่ 3.19 รายละเอียดของ โครงสร้างของข้อมูล: Chat.....	43
ตารางที่ 3.20 รายละเอียดของ โครงสร้างของข้อมูล: Notifications.....	43
ตารางที่ 4.1 คำอธิบายรายละเอียดของ โครงสร้างแอปพลิเคชัน.....	45

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 สัญลักษณ์ Flutter.....	7
รูปที่ 2.2 ตัวอย่างหน้าจอการพัฒนาด้วย Flutter Framework.....	8
รูปที่ 2.3 สัญลักษณ์ภาษา Dart.....	8
รูปที่ 2.4 ตัวอย่างหน้าจอการพัฒนาด้วยภาษา Dart.....	9
รูปที่ 2.5 สัญลักษณ์ Visual Studio Code.....	9
รูปที่ 2.6 โปรแกรม Visual Studio Code.....	10
รูปที่ 2.7 สัญลักษณ์ Android Studio.....	10
รูปที่ 2.8 ตัวอย่างหน้าจอ Emulator.....	11
รูปที่ 2.9 สัญลักษณ์ Figma.....	11
รูปที่ 2.10 ตัวอย่างหน้าจอ UX/ UI ที่ออกแบบด้วย Figma.....	12
รูปที่ 2.11 สัญลักษณ์ Postman.....	12
รูปที่ 2.12 ตัวอย่างหน้าจอการนำ Postman มาใช้งาน.....	12
รูปที่ 2.13 สัญลักษณ์ Firebase.....	13
รูปที่ 2.14 ตัวอย่างหน้าจอ Firebase ที่ใช้ในการพัฒนาแอปพลิเคชัน.....	13
รูปที่ 2.15 สัญลักษณ์ Node.js.....	14
รูปที่ 2.16 ตัวอย่าง Node.js.....	14
รูปที่ 3.1 Use Case Diagram ของ แอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง.....	16
รูปที่ 3.2 Sequence diagram : Login.....	26
รูปที่ 3.3 Sequence diagram : Register.....	27
รูปที่ 3.4 Sequence diagram : View All Pet.....	28
รูปที่ 3.5 Sequence diagram : View Detail Pet.....	29
รูปที่ 3.6 Sequence diagram : Favorite Pet.....	30
รูปที่ 3.7 Sequence diagram : Location Pet.....	31
รูปที่ 3.8 Sequence diagram : Post Pet.....	32
รูปที่ 3.9 Sequence diagram : Adopt Pet.....	33
รูปที่ 3.10 Sequence diagram : Chat Contact.....	34
รูปที่ 3.11 Sequence diagram : Approve.....	35
รูปที่ 3.12 Sequence diagram : Profile.....	36

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 3.13 Sequence diagram : Edit Profile.....	37
รูปที่ 3.14 Sequence diagram : Logout.....	38
รูปที่ 3.15 Sequence diagram : Background Notification.....	39
รูปที่ 3.16 Class Diagram ของแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง.....	40
รูปที่ 4.1 โครงสร้างของแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง.....	44
รูปที่ 4.2 หน้าเข้าสู่ระบบ (Login).....	47
รูปที่ 4.3 หน้าลงทะเบียน (Register).....	48
รูปที่ 4.4 หน้าแรก (Home).....	49
รูปที่ 4.5 หน้ารายละเอียดของสัตว์เลี้ยง (Detail).....	50
รูปที่ 4.6 หน้าสัตว์เลี้ยงที่ถูกใจ (Favorite).....	51
รูปที่ 4.7 หน้ารายการสัตว์เลี้ยงที่เลือก (Listchat).....	52
รูปที่ 4.8 หน้าสนทนา (Chat).....	53
รูปที่ 4.9 หน้ารายการสัตว์เลี้ยงที่โพสต์หาบ้าน (Mypost).....	54
รูปที่ 4.10 หน้าโพสต์หาบ้านให้สัตว์เลี้ยง (Post).....	55
รูปที่ 4.11 หน้าติดต่อ (Contact).....	56
รูปที่ 4.12 หน้าการตั้งค่า Setting.....	57
รูปที่ 4.13 หน้าโปรไฟล์ (Profile).....	58
รูปที่ 4.14 หน้าแก้ไขโปรไฟล์ (Edit Profile).....	59
รูปที่ 4.15 การแจ้งเตือนของแอปพลิเคชัน (Alert Application).....	60

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันจะเห็นได้ว่ามีสัตว์จรจัดเป็นจำนวนมากโดยเฉพาะสุนัขและแมว ซึ่งอาจจะมาจากความไม่รับผิดชอบของเจ้าของและไม่มีมาตรการควบคุมการขยายพันธุ์จากภาครัฐ นอกจากนี้ในสถานการณ์ของปัจจุบันจาก Covid-19 และน้ำท่วมในหลายพื้นที่ ทำให้ความรับผิดชอบชีวิตอีกหนึ่งชีวิตเป็นเรื่องที่ลำบากมากขึ้น จากข้อมูลข้างต้นการแก้ไขปัญหานี้ไม่ใช่ปัญหาที่จะแก้ไขได้โดยคนใดคนหนึ่งหรือกลุ่มใดกลุ่มหนึ่ง แต่ต้องเป็นการร่วมมือของเราทุกคน โดยเบื้องต้นทางคณะผู้จัดทำได้สังเกตเห็นแนวทางที่สามารถสร้างพื้นที่สำหรับการร่วมมือโดยสมัครใจ โดยมีผู้เลี้ยงสัตว์ที่ไม่สามารถที่จะเลี้ยงสัตว์ต่อไปได้ หรือพบเจอสัตว์จรจัดที่ต่างๆ แต่ไม่สามารถเลี้ยงเองได้แต่อยากช่วยเหลือให้สัตว์จรจัดให้มีเจ้าของดูแลและมีบ้านอยู่อาศัย จะสามารถเพิ่มช่องทางการติดต่อกับผู้ที่มีความสามารถจะเลี้ยงดูแลสัตว์จรจัดเหล่านี้ได้ ด้วยปัญหาเหล่านี้

จากปัญหาดังกล่าวทางคณะผู้จัดทำจึงได้ศึกษาและพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง เพื่อลดปัญหาสัตว์จรจัด และช่วยเหลือผู้ที่ไม่สามารถเลี้ยงสัตว์เลี้ยงต่อไปได้ด้วยเนื่องจากปัญหาต่างๆ ทางคณะผู้จัดทำจึงได้คาดหวังว่าแอปพลิเคชันนี้จะช่วยลดปัญหาสัตว์จรจัดข้างต้นได้ไม่มากก็น้อย โดยทางคณะผู้จัดทำได้ทำการพัฒนาแอปพลิเคชันเป็นแบบครอสแพลตฟอร์ม (Cross Platform Application) สามารถทำงานได้ทั้งบนระบบปฏิบัติการแอนดรอยด์ (Android) และไอโอเอส (iOS) พัฒนาด้วยโปรแกรมแอนดรอยด์สตูดิโอ ด้วยภาษาคาร์ท โดยผู้ใช้สามารถโพสต์ข้อมูลของสัตว์ไม่ว่าจะเป็นสัตว์จรจัดหรือมีเจ้าของ และสามารถรับดูแลสัตว์ที่ผู้ใช้ต้องการรับไปดูแลได้ โดยมีระบบแชทที่ใช้พูดคุยกับเจ้าของโพสต์ก่อน โดยเจ้าของโพสต์จะสามารถอนุมัติหรือปฏิเสธผู้ที่มาขอรับดูแลได้เพื่อเป็นการป้องกันเบื้องต้นในการนำสัตว์ไปทารุณกรรม

1.2 วัตถุประสงค์ของปริญญาานิพนธ์

เพื่อพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง

1.3 ขอบเขตปริญญาานิพนธ์

- 1.3.1 สถาปัตยกรรมที่ใช้ในการพัฒนาโครงการเป็นโมบายแอปพลิเคชันแบบครอสแพลตฟอร์ม (Cross-Platform Mobile Application)
- 1.3.2 สถาปัตยกรรมฐานข้อมูลเป็น NoSQL Database
- 1.3.3 การเริ่มต้นใช้งานแอปพลิเคชันโดยการล็อกเข้าสู่ระบบ

- 1.3.3.1 สามารถล็อกอินผ่าน Google, Facebook
- 1.3.4 ฟังก์ชันของระบบหาบ้านให้สัตว์เลี้ยงโดยสามารถเลือกหมวดหมู่ของสัตว์เลี้ยงที่ต้องการรับเลี้ยง
 - 1.3.4.1 สามารถหาสัตว์เลี้ยงตามที่คุณต้องการ ได้จากโพสต์ของผู้ใช้งานคนอื่น
 - 1.3.4.2 สามารถนำโพสต์มาหาบ้านให้สัตว์เลี้ยงได้
 - 1.3.4.3 สามารถดูโพสต์ใหม่ๆ ได้โดยไม่มีการแยกประเภทสัตว์เลี้ยง
 - 1.3.4.4 สามารถคลิกถูกใจโพสต์ที่สนใจเพื่อจะมาดูย้อนหลังได้ง่ายขึ้น
 - 1.3.4.5 สามารถดูรายละเอียดโพสต์เพิ่มได้จากการคลิกเข้าไปที่โพสต์ที่สนใจ
 - 1.3.4.6 สามารถส่งแบบฟอร์มความต้องการรับเลี้ยงสัตว์ไปหาเจ้าของโพสต์ได้
 - 1.3.4.7 เจ้าของโพสต์มีสิทธิ์ในการตัดสินใจเลือกผู้ที่ต้องการนำสัตว์เลี้ยงไปเลี้ยงได้
 - 1.3.4.8 สามารถแชทคุยกับผู้ใช้โพสต์ได้
- 1.3.5 เกี่ยวกับ Profile ของผู้ใช้งาน
 - 1.3.5.1 สามารถแก้ไข Profile ได้
 - 1.3.5.2 สามารถดูโพสต์ที่ผู้ใช้งานเลือก Favorite ได้จากหน้าถูกใจใช่เลย
 - 1.3.5.3 สามารถดูโพสต์ที่โพสต์ไปได้จากหน้าโพสต์ของฉัน
 - 1.3.5.4 สามารถดูผู้ใช้งานที่สนใจจะรับสัตว์เลี้ยงไปเลี้ยงได้จากหน้ารายชื่อผู้ติดต่อขอเลี้ยง
- 1.3.6 ขอบเขตของข้อมูล
 - 1.3.6.1 สามารถเก็บข้อมูลของผู้ใช้
 - 1.3.6.2 สามารถเก็บข้อมูลของสัตว์เลี้ยง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 1.4.1 ได้แอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยงจัดที่ตอบสนองต่อผู้ที่ต้องการจะรับเลี้ยงสัตว์หรือผู้ที่ไม่สามารถเลี้ยงสัตว์ต่อได้
- 1.4.2 ช่วยให้ผู้ให้บริการในการหาบ้านให้สัตว์เลี้ยงสามารถใช้บริการได้ง่าย และสามารถทำงานได้อย่างมีประสิทธิภาพ
- 1.4.3 ได้ช่วยเหลือสัตว์เลี้ยงจัดให้มีที่อยู่อาศัย
- 1.4.4 ลดภาระให้กับวัด และสถานดูแลสัตว์เลี้ยงจัดในการดูแลสัตว์เลี้ยงจัด

1.5 ขั้นตอนและวิธีดำเนินงานปริญญานิพนธ์

1.5.1 รวบรวมความต้องการและศึกษาข้อมูล (Detailed Study)

คณะผู้จัดทำได้ทำการรวบรวมและศึกษาปัญหาจากสภาพแวดล้อมปัจจุบัน คณะผู้จัดทำ ได้มองเห็นถึงปัญหาส่วนตัวที่จัดที่ไม่ได้รับการดูแลหรือผู้ที่เมตตาส่วนตัวจัดแต่ไม่สามารถดูแลส่วนตัวจัดได้ เพื่อให้เป็นช่องทางหนึ่งให้ผู้คนได้มองเห็นจึงนำมาวิเคราะห์ฟังก์ชันการทำงานของระบบ รวมถึงศึกษาเทคโนโลยีที่นำมาใช้ในการพัฒนาระบบ

1.5.2 วิเคราะห์ระบบงาน (System Analysis)

นำข้อมูลต่าง ๆ และขอบเขตที่ได้จากการรวบรวมข้อมูลนำมาวิเคราะห์ และวางแผนปฏิบัติงานเพื่อทำการพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง เพื่อให้ตอบสนองความต้องการผู้ใช้อย่างสมบูรณ์ โดยวิเคราะห์จากความต้องการ และขอบเขตการทำงานของแอปพลิเคชัน จนได้ฟังก์ชันการทำงานของระบบ เพื่อนำไปพัฒนาให้มีประสิทธิภาพมากขึ้นเมื่อพัฒนาเป็นแอปพลิเคชัน สำหรับอุปกรณ์สมาร์ตโฟนต่าง ๆ โดยนำเสนอด้วยภาพต่อไปนี้ Use Case Diagram , Sequence Diagram , Class Diagram และ โครงสร้างของฐานข้อมูล

1.5.3 ออกแบบระบบงาน (System Design)

ในขั้นตอนนี้จะทำการออกแบบแอปพลิเคชันเพื่อให้สามารถนำไปใช้งานได้จริง และเป็นแนวทางในการพัฒนาระบบต่อไป

1.5.3.1 ออกแบบสถาปัตยกรรม (Architecture Design) สถาปัตยกรรมที่ใช้ในการพัฒนา โครงงานเป็นแบบไคลเอนท์/ เซิร์ฟเวอร์

1.5.3.2 ออกแบบฐานข้อมูล (Database) โดยใช้ รูปแบบการเก็บของฐานข้อมูล โดยใช้รูปแบบ NoSQL ผ่าน Firebase เพื่อจัดเก็บข้อมูลของผู้ใช้งาน และข้อมูลของสัตว์เลี้ยงที่ต้องการหาบ้าน

1.5.3.3 ออกแบบหน้าจอส่วนติดต่อผู้ใช้ (User Interface Design)

ออกแบบส่วนติดต่อกับผู้ใช้หรือหน้าจอ ให้สามารถกรอกข้อมูลส่งข้อมูล ไปยังระบบฐานข้อมูลได้ถูกต้อง ครบถ้วน มีการทำงานที่ชัดเจน ไม่ซับซ้อน เพื่ออำนวยความสะดวกของผู้ใช้งาน ผู้ใช้งานโดยใช้ Flutter ในการพัฒนาสำหรับระบบปฏิบัติการแอนดรอยด์ และ ไอโอเอส สามารถปรับแต่ง UI (User Interface) ที่มีความยืดหยุ่น แยกการออกแบบเพื่อนำไปที่ประสบการณ์ของผู้ใช้งาน UX (User Experience)

1.5.3.4 กำหนดเครื่องมือในการพัฒนาระบบ

1.5.3.4.1 โปรแกรม Visual Studio Code นำมาใช้ในการรัน Code และทดสอบระบบของแอปพลิเคชัน

1.5.3.4.2 โปรแกรม Android Studio นำมาใช้ในการสร้าง Emulator สำหรับแสดงและทดสอบระบบของแอปพลิเคชัน

1.5.3.4.3 โปรแกรม Postman นำมาใช้ทดสอบ api

1.5.3.4.4 โปรแกรม Node.js นำมาใช้ในฝั่งของการพัฒนาส่วน Backend

1.5.3.4.5 โปรแกรม Figma นำมาใช้ออกแบบ UX/ UI หน้าตาของแอปพลิเคชัน

1.5.3.4.6 โปรแกรม Firebase นำมาใช้ในการบริหารจัดการเก็บข้อมูลต่าง ๆ รวมถึงเป็นโฮสต์ของ api

1.5.3.4.7 การใช้ Flutter Framework นำมาใช้ในการพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยงเพื่อการตกแต่งความสวยงามทั้งหมด ทำให้การพัฒนาแอปพลิเคชันดูสวยงาม และเป็นสัดส่วนมากขึ้น

1.5.3.4.8 การใช้ Dart Programming Language นำไปใช้ร่วมกับ Flutter ในการช่วยสร้าง UI ของแอปพลิเคชัน

1.5.4 พัฒนาระบบ (System Development)

เป็นขั้นตอนในการพัฒนาระบบเป็นการนำข้อมูลทั้งหมดที่ได้จากการวิเคราะห์และออกแบบระบบไว้ มาพัฒนาและเขียนชุดคำสั่ง โดยใช้โปรแกรม Visual Studio Code ในการพัฒนาแอปพลิเคชันด้วยภาษา Dart และใช้ Firebase ในการบริหารจัดการฐานข้อมูลที่มีความยืดหยุ่น และเน้นความเร็วในการใช้งาน

1.5.5 ทดสอบระบบ (System Testing)

คณะผู้จัดทำได้ทำการทดสอบและพัฒนาระบบไปพร้อมๆ กัน โดยใช้ Emulator ของ Android Studio ในการทดสอบการแสดงผลของหน้าจอต่าง ๆ เมื่อตรวจสอบความผิดพลาดในการทำงานของแต่ละหน้าจอและการแสดงผล รวมทั้งตรวจสอบข้อมูลต่างๆ ว่ามีความผิดพลาดในการทำงานในขั้นตอนใดบ้าง ถ้าพบข้อผิดพลาดจะทำการแก้ไขให้ถูกต้อง และทำการทดสอบอีกครั้งหลังจากทำการแก้ไขเสร็จเรียบร้อยแล้ว

1.5.6 จัดทำเอกสารประกอบปฏิญานิพนธ์ (Documentation)

เป็นการจัดทำเอกสารแนวทางประกอบปฏิญานิพนธ์ วิธีการและขั้นตอนการดำเนินปฏิญานิพนธ์ เพื่อนำเสนอรายงานต่ออาจารย์ที่ปรึกษา และคณะกรรมการสอบปฏิญานิพนธ์ เพื่อขอรับคำแนะนำ และนำไปใช้เป็นเอกสารอ้างอิงในอนาคต

1.6 ขั้นตอนและระยะเวลาในการดำเนินงานปฏิญานิพนธ์

ตารางที่ 1.1 ขั้นตอนและระยะเวลาในการดำเนินงานปฏิญานิพนธ์

ขั้นตอนการดำเนินงาน	2565		
	พ.ค.	มิ.ย.	ก.ค.
1. รวบรวมความต้องการ	↔		
2. วิเคราะห์ระบบ	↔↔		
3. ออกแบบระบบ	↔↔		
4. พัฒนาระบบ		↔↔↔↔	
5. ทดสอบระบบ		↔↔↔↔	
6. จัดทำเอกสาร			↔↔

1.7 อุปกรณ์และเครื่องมือที่ใช้ในการพัฒนา

1.7.1 ฮาร์ดแวร์ (Hardware)

1.7.1.1 เครื่องคอมพิวเตอร์ โน้ตบุ๊ก Asus รุ่น TUF Gaming

- AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz
- NVIDIA GeForce RTX 2060
- 16 GB
- 1 TB SSD PCIe M.2

1.7.1.2 เครื่องคอมพิวเตอร์

- CPU - Intel(R) Core(TM) i9-9900K CPU - @ 3.60GHz 8Cores
- AMD Radeon RX 6700 XT
- 16 GB
- 10000 TB SSD PCIe M.2

1.7.2 ซอฟต์แวร์ (Software)

1.7.2.1 โปรแกรม Virtual Studio Code

1.7.2.2 โปรแกรม Android Studio

1.7.2.3 โปรแกรม Postman

1.7.2.4 โปรแกรม Node.js

1.7.2.5 โปรแกรม Figma

1.7.2.6 โปรแกรม Firebase

1.7.2.7 เว็บเบราว์เซอร์ Google Chrome

1.8 อุปกรณ์แลคเครื่องมือที่รองรับระบบ

1.8.1 ฮาร์ดแวร์ (Hardware)

1.8.1.1 Emulator Pixel 4 XL API 31

1.8.2 ซอฟต์แวร์ (Software)

1.8.2.1 Android Version 10 – 12

1.8.2.2 Sdk Versoin 21 - 30

บทที่ 2

การทบทวนวรรณกรรมที่เกี่ยวข้อง

การพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง คณะผู้จัดทำได้ทำการการศึกษาข้อมูล แนวคิด และทฤษฎีที่เกี่ยวข้อง เพื่อเป็นกรอบแนวทางในการพัฒนาระบบ ซึ่งสามารถแบ่งเป็นหัวข้อ โดยมีรายละเอียดดังต่อไปนี้

2.1 ฟลัตเตอร์ เฟรมเวิร์ค (Flutter Framework)¹

Flutter เป็น open-source ในการพัฒนา SDK สร้างขึ้นโดย Google เพื่อใช้ในการพัฒนา โปรแกรมประยุกต์สำหรับ Android และ iOS โดย Flutter จะเป็นแบบ hybrid-native สามารถใช้งานได้ทั้งบน Android และ iOS ประสิทธิภาพใกล้เคียงกับ native app ภาษาที่ใช้ใน Flutter ก็คือภาษา Dart โดย Flutter จะมี Component พื้นฐานให้เพื่อที่จะได้ใกล้เคียงกับ design guideline สำหรับ iOS นั้นก็มี Cupertino เป็น Component พื้นฐานเหมือนกัน เพื่อให้ใกล้เคียงกับ Native element มากที่สุด โดยคณะผู้จัดทำได้นำ flutter มาใช้ในการพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง เพื่อการตกแต่งความสวยงามทั้งหมด ทำให้การพัฒนาแอปพลิเคชันดูสวยงาม และเป็นสัดส่วนมากขึ้น

2.1.1 ข้อดีของการใช้ Flutter ในการพัฒนาแอปพลิเคชัน

- 2.1.1.1 ประสิทธิภาพที่ดีเนื่องจากภาษา Dart ลักษณะการ compile เป็น machine code
- 2.1.1.2 หน้าตาของแอปพลิเคชันที่พัฒนาจะเหมือนกัน ถึงแม้ว่าเวอร์ชันของ OS จะต่างกัน แต่ก็อาจไม่ได้เหมือนกันทุกอย่าง แต่ก็สามารถ design แยกแต่ละ Platform ได้โดยดูจากตัวแปร Platform ได้เช่นเดียวกัน



รูปที่ 2.1 สัญลักษณ์ Flutter

¹ <https://www.mindphp.com/บทความ/241-mobile-application/5554-flutter-framework.html>

```

import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:project/screens/chat_screen.dart';
import 'package:project/screens/contact_screen.dart';
import 'package:project/screens/detail_screen.dart';
import 'package:project/screens/home_screen.dart';
import 'package:project/screens/home_screen.dart';
import 'package:project/screens/login_screen.dart';
import 'package:project/screens/register_screen.dart';
import 'package:project/screens/reset_password_screen.dart';
import 'package:project/screens/setting_screen.dart';
import 'package:project/screens/signin_screen.dart';
import 'package:project/screens/signup_screen.dart';
import 'package:project/screens/welcome_screen.dart';

Future _firebaseMessagingBackgroundHandler(FirebaseMessaging messaging) async {
  print('Background message: ${messaging.data}');
}

Future main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(primarySwatch: Colors.teal),
      initialRoute: '/',
    );
  }
}

```

รูปที่ 2.2 ตัวอย่างหน้าจอการพัฒนาด้วย Flutter Framework

2.2 ภาษาดาร์ต (Dart Programming Language)²

Dart เป็นภาษาโปรแกรมที่เอาไว้สำหรับพัฒนาแอปพลิเคชันบนแพลตฟอร์มหลากหลาย โดยได้ทั้ง mobile, desktop, server และที่ web สิ่งที่เป็นที่นิยมที่สุดที่ทำให้คนสนใจมาเรียนภาษา Dart คือเพื่อที่จะเอาไปใช้ร่วมกับ Flutter ที่เป็นเครื่องมือช่วยสร้าง UI ของ Google ซึ่งใช้ได้ทั้งกับ Android และ iOS หรือจะเป็นใน Desktop กับ Web

ภาษา Dart ถูกสร้างโดย Google และปล่อยให้ใช้งานแบบ open source ทำให้สามารถนำไปใช้งานได้ฟรี ๆ และการที่ Dart ถูกออกแบบมาให้ใช้งานได้ง่ายและมีประสิทธิภาพแบบภาษาเชิงวัตถุอื่น ๆ อย่าง Java C# C++ จึงเป็นตัวเลือกภาษาที่น่าสนใจในการศึกษาเป็นภาษาแรกอีกด้วย โดยคณะผู้จัดทำได้นำภาษาดาร์ตไปใช้ร่วมกับ Flutter ในการช่วยสร้าง UI ของแอปพลิเคชัน



Dart

รูปที่ 2.3 สัญลักษณ์ภาษา Dart

² <https://www.borntodev.com/2020/04/11/เรียน-dart-แบบก้าวกระโดด/>

```

1 import 'package:firebase_messaging/firebase_messaging.dart';
2 import 'package:flutter/material.dart';
3 import 'package:firebase_core/firebase_core.dart';
4 import 'package:project/screens/chat_screen.dart';
5 import 'package:project/screens/contact_screen.dart';
6 import 'package:project/screens/detail_screen.dart';
7 import 'package:project/screens/feature_list_screen.dart';
8 import 'package:project/screens/home_screen.dart';
9 import 'package:project/screens/login_screen.dart';
10 import 'package:project/screens/news_screen.dart';
11 import 'package:project/screens/register_screen.dart';
12 import 'package:project/screens/setting_screen.dart';
13 import 'package:project/screens/sign_in_screen.dart';
14 import 'package:project/screens/sign_out_screen.dart';
15 import 'package:project/screens/sign_up_screen.dart';
16 import 'package:project/screens/sign_up_screen.dart';
17 import 'package:project/screens/sign_up_screen.dart';
18 import 'package:project/screens/sign_up_screen.dart';
19
20 Future _sendMessage(BuildContext context) async {
21   print('background message: $message-${DateTime.now()}');
22 }
23
24 Future main() async {
25   WidgetsFlutterBinding.ensureInitialized();
26   await Firebase.initializeApp();
27   FirebaseMessaging.onBackgroundMessage(_sendMessage);
28   runApp(const MyApp());
29 }
30
31 class MyApp extends StatelessWidget {
32   const MyApp({Key? key}) : super(key: key);
33
34   @override
35   Widget build(BuildContext context) {
36     return MaterialApp(
37       title: 'Flutter Demo',
38       theme: ThemeData(primarySwatch: Colors.teal),
39       initialRoute: '/',
40     );
41   }
42 }
43
44 void main() {
45   runApp(MyApp());
46 }

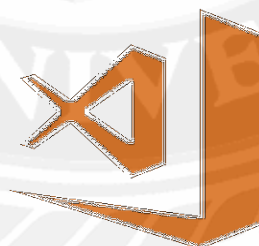
```

รูปที่ 2.4 ตัวอย่างหน้าจอการพัฒนาด้วยภาษา Dart

2.3 วิชวลสตูดิโอโค้ด (Visual Studio Code)³

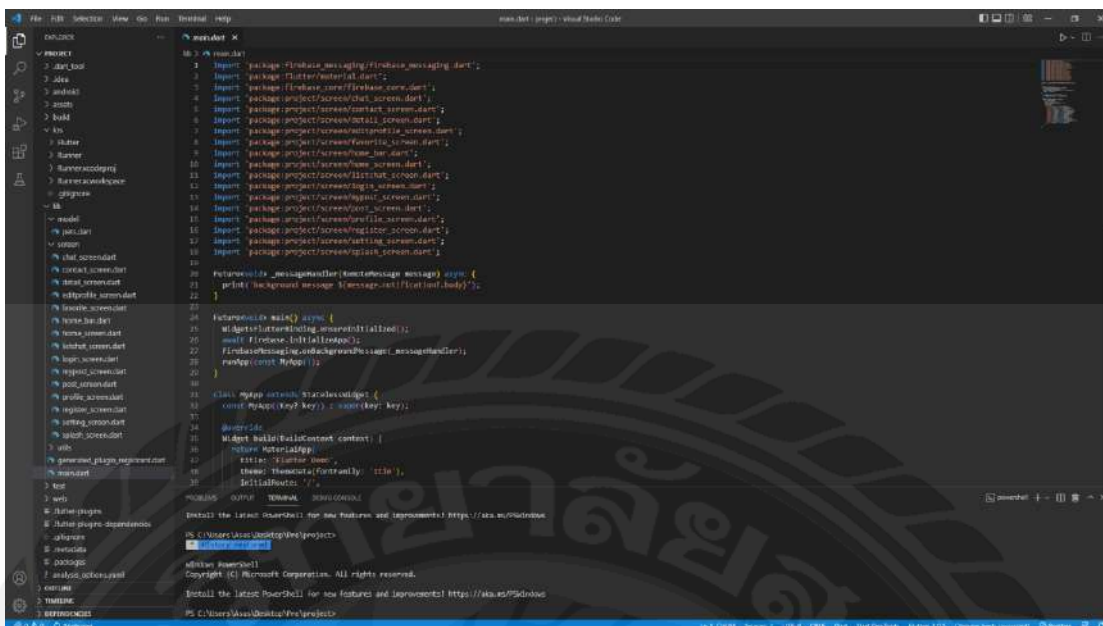
Visual Studio Code หรือ VSCode เป็นโปรแกรม Code Editor ที่ใช้ในการแก้ไขและปรับแต่งโค้ด จากค่ายไมโครซอฟท์ สามารถนำมาใช้งานได้แบบฟรี

ซึ่ง Visual Studio Code นั้น เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานข้ามแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux สนับสนุนทั้งภาษา JavaScript, TypeScript และ Node.js สามารถเชื่อมต่อกับ Git ได้ นำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือส่วนขยายต่าง ๆ ให้เลือกใช้อย่างมากมาย ไม่ว่าจะเป็น 1.การเปิดใช้งานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python, PHP หรือ Go 2.Themes 3.Debugger 4.Commands เป็นต้น โดยคณะผู้จัดทำได้นำ Visual Studio Code มาใช้ในการรัน code และทดสอบระบบของแอปพลิเคชัน



รูปที่ 2.5 สัญลักษณ์ Visual Studio Code

³ <https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>



รูปที่ 2.6 โปรแกรม Visual Studio Code

2.4 แอนดรอยด์สตูดิโอ (Android Studio)⁴

Android Studio เป็น IDE Tool จาก Google ใ้ใช้สำหรับการพัฒนา Android โดยพัฒนาจากแนวคิดพื้นฐานมาจาก IntelliJ IDEA คล้าย ๆ กับการทำงานของ Eclipse และ Android ADT Plugin โดยวัตถุประสงค์ของ Android Studio คือต้องการพัฒนาเครื่องมือ IDE ที่สามารถพัฒนา App บน Android ให้มีประสิทธิภาพมากขึ้น ทั้งด้านการออกแบบ GUI ที่ช่วยให้สามารถ Preview ตัว App มุมมองที่แตกต่างกันบน Smart Phone แต่ละรุ่น สามารถแสดงผลบางอย่างได้ทันทีโดยไม่ต้องทำการรัน App บน Emulator รวมทั้งยังแก้ไขปรับปรุงในเรื่องของความเร็วของ Emulator ที่ยังเจอปัญหากันอยู่ในปัจจุบัน โดยคณะผู้จัดทำได้นำแอนดรอยด์สตูดิโอมาใช้ในการสร้าง Emulator สำหรับแสดงและทดสอบระบบของแอปพลิเคชัน



รูปที่ 2.7 สัญลักษณ์ Android Studio

⁴ <https://medium.com/@palmz/เริ่มต้นสร้าง-android-application-พื้นฐานด้วย-android-studio-lab-3sb04-3fda43b07a1>



รูปที่ 2.8 ตัวอย่างหน้าจอ Emulator

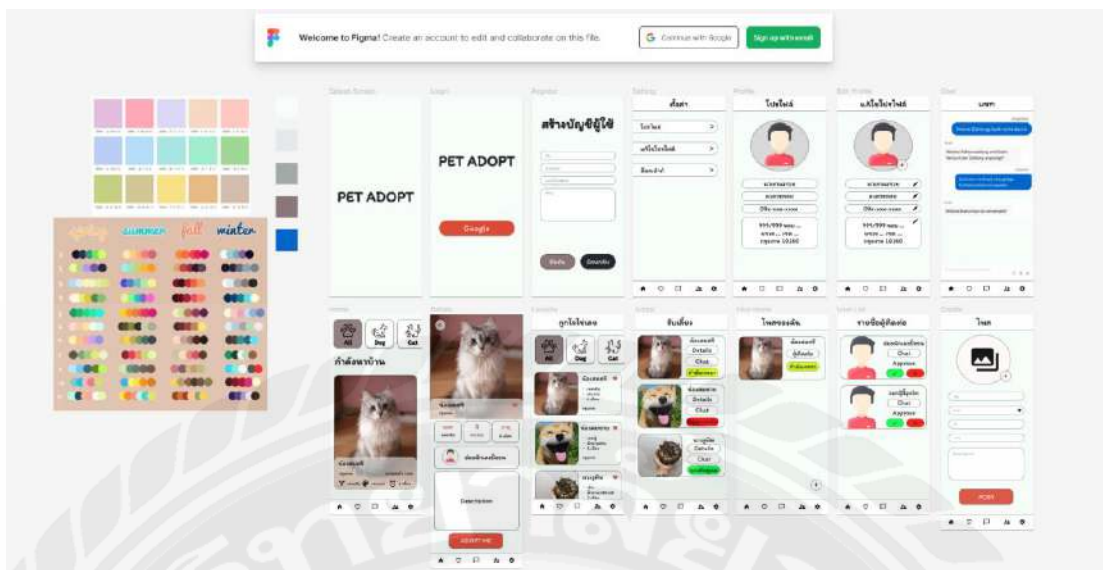
2.5 ฟิกมาร์ (Figma)⁵

FIGMA คือเครื่องมือออกแบบเว็บไซท์ หรือ แอปพลิเคชันต่าง ๆ ที่เกิดมาเพื่อช่วยนักออกแบบ UX/ UI โดยสามารถใช้งานได้ผ่านทาง web browser ไม่จำเป็นต้อง install (แต่ก็มีแอปฯ ให้เราสามารถ install ลงเครื่องด้วย) ทำให้สะดวกในการใช้งานมาก โดยตัวเครื่องมือออกแบบมาให้เหมาะกับคนที่จำเป็นจะต้องทำโปรเจกต์ร่วมกันกับทีม หรือต้องการหาคำตอบเวลาคุยโปรเจกต์กับลูกค้าและลูกค้าเปิดไฟล์งานไม่ได้ อีกทั้งตัวเครื่องมือยังมีฟีเจอร์ที่น่าสนใจ ช่วยให้สามารถทำงานได้ง่ายขึ้นอีกด้วย โดยคณะผู้จัดทำได้นำฟิกมาร์มาออกแบบ UX/ UI หน้าตาของแอปพลิเคชัน



รูปที่ 2.9 สัญลักษณ์ Figma

⁵ <https://dev.classmethod.jp/articles/whats-about-figma/>



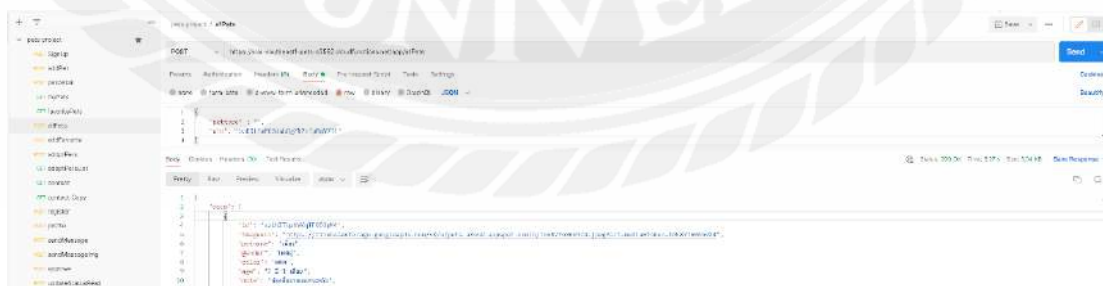
รูปที่ 2.10 ตัวอย่างหน้าจอ UX/ UI ที่ออกแบบด้วย Figma

2.6 โปสต์แมน (Postman)⁶

Postman คือเครื่องมือสำหรับการพัฒนาและทดสอบ API service ซึ่งเป็นที่นิยมอย่างกว้างขวาง เพราะการใช้งานที่ง่ายดาย มี UI สวยงามเข้าใจง่าย ๆ และสามารถใช้ได้ฟรี ฟังก์ชันทั่วไปที่คนใช้งานจะเป็นการทดสอบส่ง request เพื่อดู response ที่ได้กลับมาเป็นข้อมูล แอดชัน ๆ ต่าง ๆ โดยคณะผู้จัดทำได้ใช้โปสต์แมนในการทดสอบ api



รูปที่ 2.11 สัญลักษณ์ Postman



รูปที่ 2.12 ตัวอย่างหน้าจอการนำ Postman มาใช้งาน

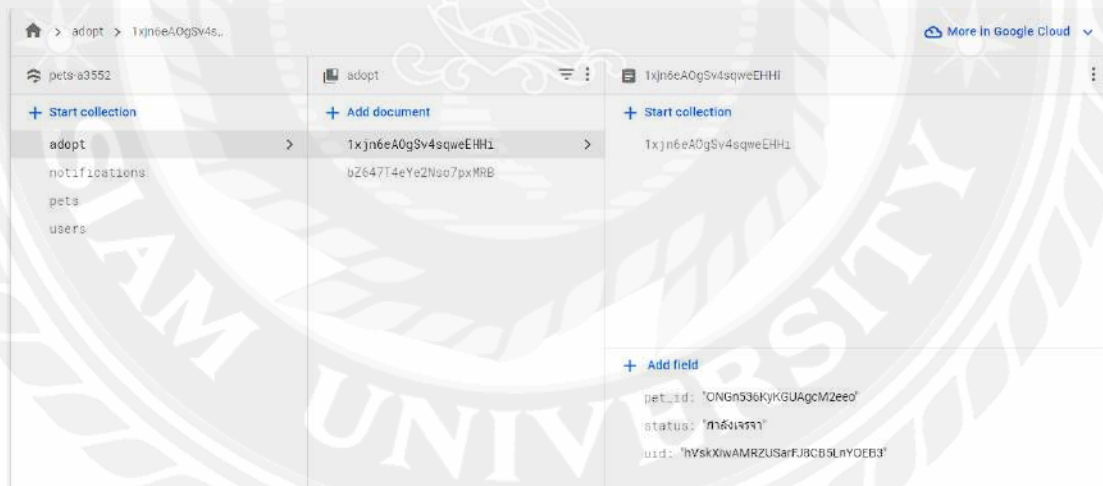
⁶ <https://www.mindphp.com/บทความ/microsoft/237-free-software/8446-postman.html>

2.7 ไฟร์เบส (Firebase)⁷

Firebase (ไฟร์เบส) คือ บริการ backend และ แพลตฟอร์ม ครอบงำสำหรับนักพัฒนาแอปพลิเคชัน และโปรแกรมประยุกต์บนเว็บแพลตฟอร์มที่มีเครื่องมือและโครงสร้างพื้นฐานที่ได้รับการออกแบบมาเพื่อช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่มีคุณภาพสูง Firebase (ไฟร์เบส) ถูกสร้างขึ้นจากคุณสมบัติเสริมว่านักพัฒนาสามารถผสมและจับคู่เพื่อให้พอดีกับความต้องการของบริษัท ก่อตั้งขึ้นในปี 2011 โดยแอนดรูว์ลีและเจมส์ เทมปลิน เริ่มต้น Firebase (ไฟร์เบส) เป็นฐานข้อมูลเรียลไทม์ซึ่งมี API ที่ช่วยให้นักพัฒนาในการจัดเก็บและซิงค์ข้อมูล Google Firebase 2.0 ก่อให้เกิดได้ซึ่งกิจการ Firebase (ไฟร์เบส) และมีการพัฒนาให้สามารถ จากบริการ backend เก็บข้อมูลอย่างเดียว มาเป็น แพลตฟอร์ม ครอบงำสำหรับนักพัฒนาแอป (รองรับ IOS, Android, Web) รองรับบริการแทบทุกอย่างที่นักพัฒนาแอปพลิเคชันต้องใช้งาน โดยคณะผู้จัดทำได้นำไฟร์เบสมาใช้ในการบริหารจัดการเก็บข้อมูลต่าง ๆ รวมถึงเป็น โฮสต์ของ api



รูปที่ 2.13 สัญลักษณ์ Firebase



รูปที่ 2.14 ตัวอย่างหน้าจอ Firebase ที่ใช้ในการพัฒนาแอปพลิเคชัน

⁷ <https://www.mindphp.com/คู่มือ/73-คืออะไร/3921-what-is-firebase-backend.html>

บทที่ 3

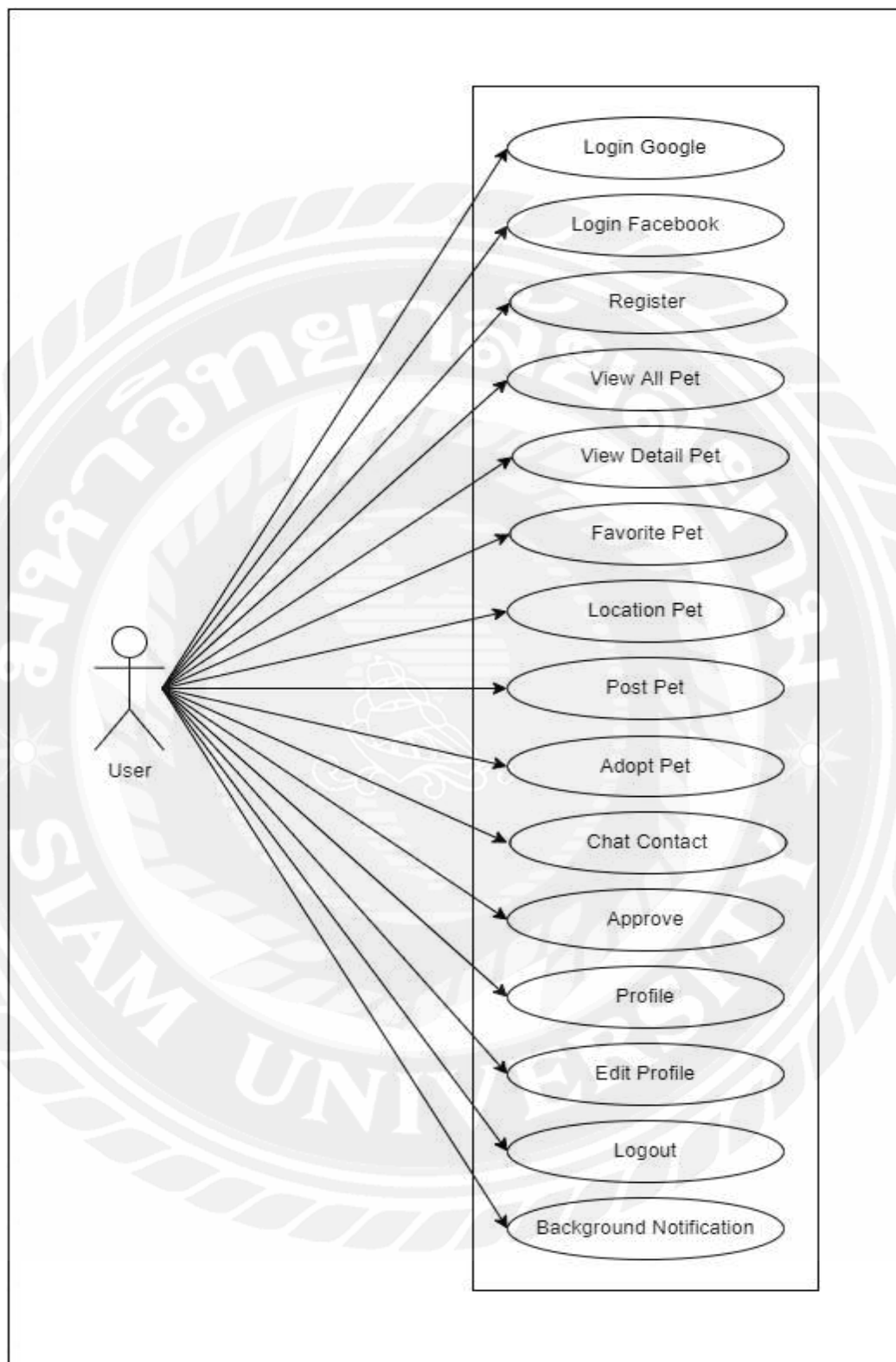
การวิเคราะห์ระบบ

3.1 รายละเอียดของปฏิญานินพนธ์

แอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง เป็นแอปพลิเคชันที่เหมาะสมสำหรับทุกเพศทุกวัยที่ต้องการหาบ้านให้สัตว์เลี้ยง หรือต้องการที่จะเลี้ยงสัตว์ ซึ่งสามารถใช้งานได้ในระบบปฏิบัติการ แอนดรอยด์ และ ไอโอเอส พัฒนาขึ้นมาเพื่อตอบสนองความต้องการของผู้ใช้งาน โดยผู้ใช้งานเป็นได้ทั้งผู้ที่ต้องการส่งต่อสัตว์เลี้ยงหรือพบเจอสัตว์จรจัดหรือผู้ที่ต้องการหาสัตว์เลี้ยงไปดูแล ก็สามารถใช้แอปพลิเคชันนี้เป็นสื่อกลางในการค้นหาและติดต่อ เพื่อส่งต่อหรือรับสัตว์เลี้ยงไปดูแล ในระบบของแอปพลิเคชันนี้ผู้ใช้งานจะสามารถที่จะโพสต์หาบ้านให้สัตว์เลี้ยงและสัตว์จรจัดได้หรือจะรับสัตว์เลี้ยงจากโพสต์ของผู้ใช้ได้โดยจะมีไลเคนของสัตว์เลี้ยงนั้น ๆ อยู่ภายในโพสต์และยังสามารถติดต่อกับผู้โพสต์ได้โดยผ่านระบบแชท

ดังนั้นในการพัฒนาระบบ จะประกอบด้วยผู้ใช้งานที่เป็นได้ทั้งผู้หาบ้านให้สัตว์เลี้ยงหรือผู้ที่ต้องการรับเลี้ยงสัตว์ โดยผู้ใช้งานจะต้องล็อกอินเพื่อเข้าใช้งานผ่านกูเกิ้ล (Google) หรือ เฟสบุ๊ค (Facebook) ในการเข้าสู่ระบบครั้งแรกจะต้องกรอกข้อมูลชื่อ นามสกุล เบอร์โทรศัพท์ และที่อยู่เพื่อเก็บลงฐานข้อมูลไว้ใช้สำหรับแจ้งเตือนความคืบหน้าของการรับเลี้ยงสัตว์ แอปพลิเคชันสามารถให้ผู้ใช้โพสต์หาบ้านให้สัตว์เลี้ยงได้ และถ้าผู้ใช้ชอบหรือถูกใจสัตว์ตัวไหนก็สามารถคลิก favorite ไว้ดูความเคลื่อนไหวของสัตว์ตัวนั้นได้ แต่ถ้าผู้ใช้อยากรับสัตว์ไปเลี้ยงก็สามารถคลิกปุ่ม Adopt จากโพสต์ของผู้ใช้งานอื่นได้ หลังจากนั้นสัตว์เลี้ยงจะไปอยู่ในหน้า listchat ของผู้คลิก Adopt และสามารถคุยกับเจ้าของโพสต์ผ่านแชทได้เลย แอปพลิเคชันพัฒนาโดยใช้ Flutter ในการสร้าง UI ข้อมูลของผู้ใช้งานและข้อมูลต่างๆ ภายในแอปพลิเคชันจะถูกจัดเก็บลงในฐานข้อมูล firebase และเขียนชุดคำสั่งด้วยภาษา Dart เครื่องมือที่ใช้ในการพัฒนา คือ Visual Studio Code

3.2 แสดงฟังก์ชันการทำงานของระบบ (Use Case Diagram)



รูปที่ 3.1 Use Case Diagram ของ แอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง

3.3 คำอธิบายรายละเอียดของยูสเคส (Use Case Description)

ตารางที่ 3.1 แสดงรายละเอียดของ Use case: Login Facebook

Use Case Name	Login Facebook
Use Case ID	UC1
Brief Description	สำหรับการยืนยันตัวตนของผู้ใช้ โดยใช้อีเมลและรหัสผ่านของ โ Facebook
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานจะต้องมีบัญชี (account) ของ Facebook ก่อนถึงสามารถเข้าใช้ งานได้
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้เข้าแอปพลิเคชัน 2. ระบบจะแสดงปุ่มสำหรับเข้าสู่ระบบ 3. ผู้ใช้ป้อนข้อมูลอีเมลและรหัสผ่านของบัญชี Facebook <ol style="list-style-type: none"> 3.1 มีบัญชีอยู่ในระบบ <ol style="list-style-type: none"> 3.1.1 ระบบจะไปที่หน้า Home 3.2 ยังไม่มีบัญชีอยู่ในระบบ <ol style="list-style-type: none"> 3.2.1 ระบบจะนำไปหน้า Register เพื่อกรอกข้อมูล
Post Condition	เมื่อเข้าสู่ระบบได้สำเร็จ จะสามารถเข้าไปยังหน้า Home
Alternative Flows	-

ตารางที่ 3.2 แสดงรายละเอียดของ Use case: Login Google

Use Case Name	Login Google
Use Case ID	UC2
Brief Description	สำหรับการยืนยันตัวตนของผู้ใช้ โดยใช้อีเมลและรหัสผ่านของ Google
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องมีบัญชี (Account) ของ Google ก่อน
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้เข้าแอปพลิเคชัน 2. ระบบจะแสดงปุ่มสำหรับเข้าสู่ระบบ 3. ผู้ใช้ป้อนข้อมูลอีเมลและรหัสผ่านของ account google <ol style="list-style-type: none"> 3.1 มีบัญชีอยู่ในระบบ <ol style="list-style-type: none"> 3.1.1 ระบบจะไปที่หน้า Home 3.2 ยังไม่มีบัญชีอยู่ในระบบ <ol style="list-style-type: none"> 3.2.1 ระบบจะนำไปหน้า Register เพื่อกรอกข้อมูล
Post Condition	เมื่อเข้าสู่ระบบได้สำเร็จ จะไปยังหน้า Home
Alternative Flows	-

ตารางที่ 3.3 แสดงรายละเอียดของ Use case: Register

Use Case Name	Register
Use Case ID	UC3
Brief Description	สำหรับเก็บข้อมูลส่วนตัวของผู้ใช้
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องทำการเข้าสู่ระบบ โดยไม่มีบัญชีอยู่ในระบบ
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเกิดขึ้นก็ต่อเมื่อผู้ใช้งานล็อกอินผ่าน Facebook หรือ google โดยใช้บัญชีที่ไม่มีอยู่ในระบบ 2. จะแสดง Field ให้กรอกข้อมูลส่วนตัว 3. จะมีปุ่มยืนยันกับย้อนกลับ <ol style="list-style-type: none"> 3.1 คลิกยืนยันระบบจะนำไปหน้า Home 3.2 คลิกย้อนกลับระบบจะทำการ Logout กลับไปสู่หน้า Login
Post Condition	เมื่อผู้ใช้งานกรอกข้อมูลสำเร็จ จะ ไปยังหน้า Home (หน้าแรก)
Alternative Flows	-

ตารางที่ 3.4 แสดงรายละเอียดของ Use case: View All Pet

Use Case Name	View All Pet
Use Case ID	UC4
Brief Description	หน้าโฮม ผู้ใช้จะสามารถเลือกดูสัตว์เลี้ยงที่หาบ้านได้หลากหลาย
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องทำการเข้าสู่ระบบ โดยมีบัญชีอยู่ในระบบ
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเกิดขึ้นก็ต่อเมื่อผู้ใช้งานล็อกอินผ่าน Facebook หรือ Google โดยใช้บัญชีที่มีอยู่ในระบบ 2. ผู้ใช้งานสามารถเลือกหมวดหมู่ของสัตว์เลี้ยงได้
Post Condition	ระบบจะ ไปยังหน้าแสดงสัตว์เลี้ยงของหมวดนั้นๆ
Alternative Flows	-

ตารางที่ 3.5 แสดงรายละเอียดของ Use case: View Detail Pet

Use Case Name	View Detail Pet
Use Case ID	UC5
Brief Description	หน้าแสดงรายละเอียดของสัตว์เลี้ยง
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องทำการคลิกเลือกสัตว์เลี้ยงที่มีความสนใจรับเลี้ยง
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้งานคลิกเลือกสัตว์เลี้ยงที่สนใจ 2. สามารถดูรายละเอียดต่างๆ ของสัตว์เลี้ยงที่สนใจ
Post Condition	-
Alternative Flows	-

ตารางที่ 3.6 แสดงรายละเอียดของ Use case: Favorite Pet

Use Case Name	Favorite Pet
Use Case ID	UC6
Brief Description	หน้าแสดงสัตว์เลี้ยงที่ผู้ใช้งานได้คลิกถูกใจไว้
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องคลิกถูกใจสัตว์ที่ชอบก่อน
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้งานคลิกถูกใจสัตว์จากหน้ารายละเอียด (Detail) 2. แสดงสัตว์เลี้ยงที่เราคลิกถูกใจและสามารถคลิกไปดูรายละเอียด (Detail) ของสัตว์เลี้ยงตัวนั้นได้ 3. แสดงสถานะของสัตว์เลี้ยงที่ผู้ใช้งานคลิกถูกใจ
Post Condition	-
Alternative Flows	-

ตารางที่ 3.7 แสดงรายละเอียดของ Use case: Location Pet

Use Case Name	Location Pet
Use Case ID	UC7
Brief Description	สถานที่อยู่ของสัตว์เลี้ยง
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานสามารถคลิกดูที่อยู่ของสัตว์เลี้ยงได้จากหน้ารายละเอียด (Detail) ของสัตว์เลี้ยงที่สนใจ
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้คลิกเลือกสัตว์เลี้ยงที่สนใจ 2. หน้าจอจะแสดงหมุดที่ตั้งอยู่ด้านขวาล่างเมื่อคลิกระบบจะนำไปการนำทางโดย Google Map
Post Condition	ระบบจะนำไปยังหมุดที่ปักไว้ใน Google Map
Alternative Flows	-

ตารางที่ 3.8 แสดงรายละเอียดของ Use case: Post Pet

Use Case Name	Post Pet
Use Case ID	UC8
Brief Description	หน้าแสดงโพสต์ที่ผู้ใช้โพสต์ข้อมูลหาบ้านให้กับสัตว์เลี้ยง
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องล็อกอินเข้าสู่ระบบผ่านบัญชี Facebook หรือ บัญชี Google โดยใช้บัญชีที่มีอยู่ในระบบ
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเกิดขึ้นก็ต่อเมื่อผู้ใช้คลิกปุ่มบวกจากหน้าโพสต์ 2. จะมี Field ให้กรอกข้อมูลและรูปของสัตว์เลี้ยงที่ผู้ใช้งานต้องการโพสต์ 3. คลิกปุ่มโพสต์เพื่อบันทึกข้อมูลลงระบบ
Post Condition	ระบบจะพาไปหน้าโพสต์ที่ผู้ใช้ได้โพสต์สัตว์เลี้ยงไว้
Alternative Flows	-

ตารางที่ 3.9 แสดงรายละเอียดของ Use case: Adopt Pet

Use Case Name	Adopt Pet
Use Case ID	UC9
Brief Description	ปุ่มสำหรับผู้สนใจรับสัตว์เลี้ยงไปดูแล
Primary Actors	User
Secondary Actors	-
Preconditions	คลิกได้จากหน้ารายละเอียด (Detail) ของสัตว์เลี้ยงที่สนใจ
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้คลิกเลือกสัตว์เลี้ยงที่สนใจ 2. หลังจากผู้ใช้ดูรายละเอียดของสัตว์เรียบร้อยแล้ว ถ้าผู้ใช้สนใจรับเลี้ยงสามารถคลิกปุ่ม Adopt เพื่อไปพูดคุยกับเจ้าของโพสต์ได้
Post Condition	ระบบจะพาไปที่หน้า Home
Alternative Flows	-

ตารางที่ 3.10 แสดงรายละเอียดของ Use case: Chat Contact

Use Case Name	Chat Contact
Use Case ID	UC10
Brief Description	ผู้ใช้สามารถพูดคุย, เกรงใจ, ติดต่อ, ต่อรอง
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้ต้องคลิกปุ่ม Adopt สัตว์เลี้ยงหรือมีผู้มาคลิก Adopt สัตว์เลี้ยงที่ผู้ใช้โพสต์หาบ้านไว้
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อคลิกปุ่มแชท 2. ส่งข้อความและรูปภาพกันได้ระหว่างเจ้าของโพสต์และผู้รับดูแลสัตว์เลี้ยง
Post Condition	-
Alternative Flows	-

ตารางที่ 3.11 แสดงรายละเอียดของ Use case: Approve

Use Case Name	Approve
Use Case ID	UC11
Brief Description	สำหรับการอนุมัติผู้ที่สนใจรับเลี้ยงสัตว์เลี้ยง
Primary Actors	User
Secondary Actors	-
Preconditions	ต้องมีผู้สนใจมาคลิก Adopt สัตว์เลี้ยงที่โพสต์ก่อน
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อคลิกปุ่มผู้ติดต่อของโพสต์ 2. เมื่อพูดคุยกันเสร็จแล้วจะสามารถเลือกที่จะยอมรับหรือปฏิเสธผู้ขอรับดูแลคนนั้นได้
Post Condition	-
Alternative Flows	-

ตารางที่ 3.12 แสดงรายละเอียดของ Use case: Profile

Use Case Name	Profile
Use Case ID	UC12
Brief Description	แสดงข้อมูลส่วนตัว
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องทำการเข้าสู่ระบบ (Login) จากบัญชีที่กรอกลงทะเบียน (Register) แล้ว
Main Flow	1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้คลิกปุ่ม Profile 2. สามารถดูรายละเอียดข้อมูลส่วนตัวของตนเองได้
Post Condition	-
Alternative Flows	-

ตารางที่ 3.13 แสดงรายละเอียดของ Use case: Edit Profile

Use Case Name	Edit Profile
Use Case ID	UC13
Brief Description	แก้ไขข้อมูลส่วนตัว
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องเข้าสู่ระบบ Login จากบัญชีที่กรอกลงทะเบียน (Register) เรียบร้อยแล้ว
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้เข้า Profile และคลิกแก้ไข 2. จะมี Field ให้กรอกข้อมูลและรูปของผู้ใช้ 3. คลิกยืนยันหรือย้อนกลับได้
Post Condition	ระบบจะนำไปสู่หน้า Profile
Alternative Flows	-

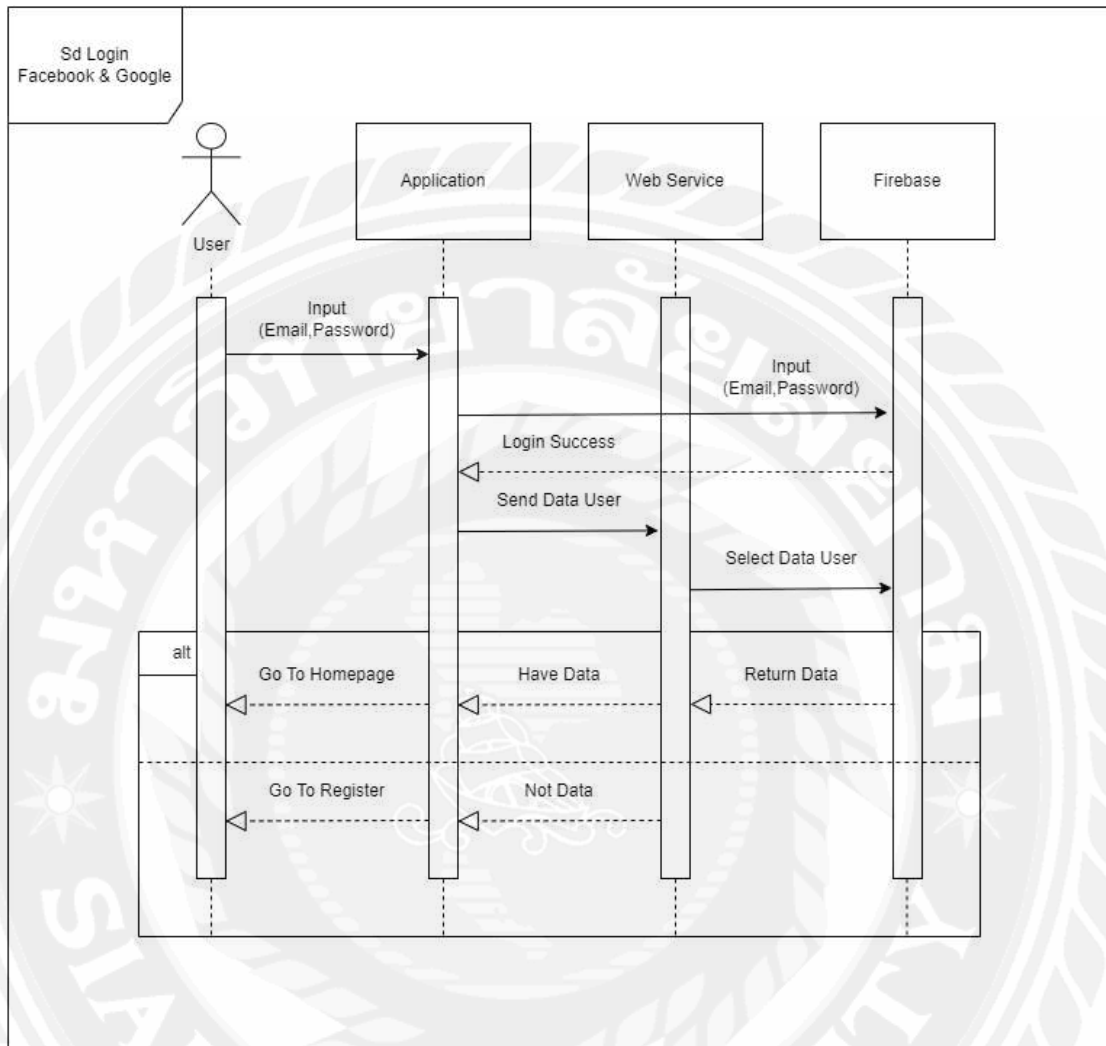
ตารางที่ 3.14 แสดงรายละเอียดของ Use case: Logout

Use Case Name	Logout
Use Case ID	UC14
Brief Description	ออกจากระบบ
Primary Actors	User
Secondary Actors	-
Preconditions	ผู้ใช้งานต้องล็อกอินผ่านบัญชี Facebook หรือ Google โดยใช้บัญชีที่มีอยู่ในระบบ
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้คลิกปุ่ม Logout 2. ออกจากระบบ
Post Condition	ระบบจะนำไปสู่หน้าเข้าสู่ระบบ (Login)
Alternative Flows	-

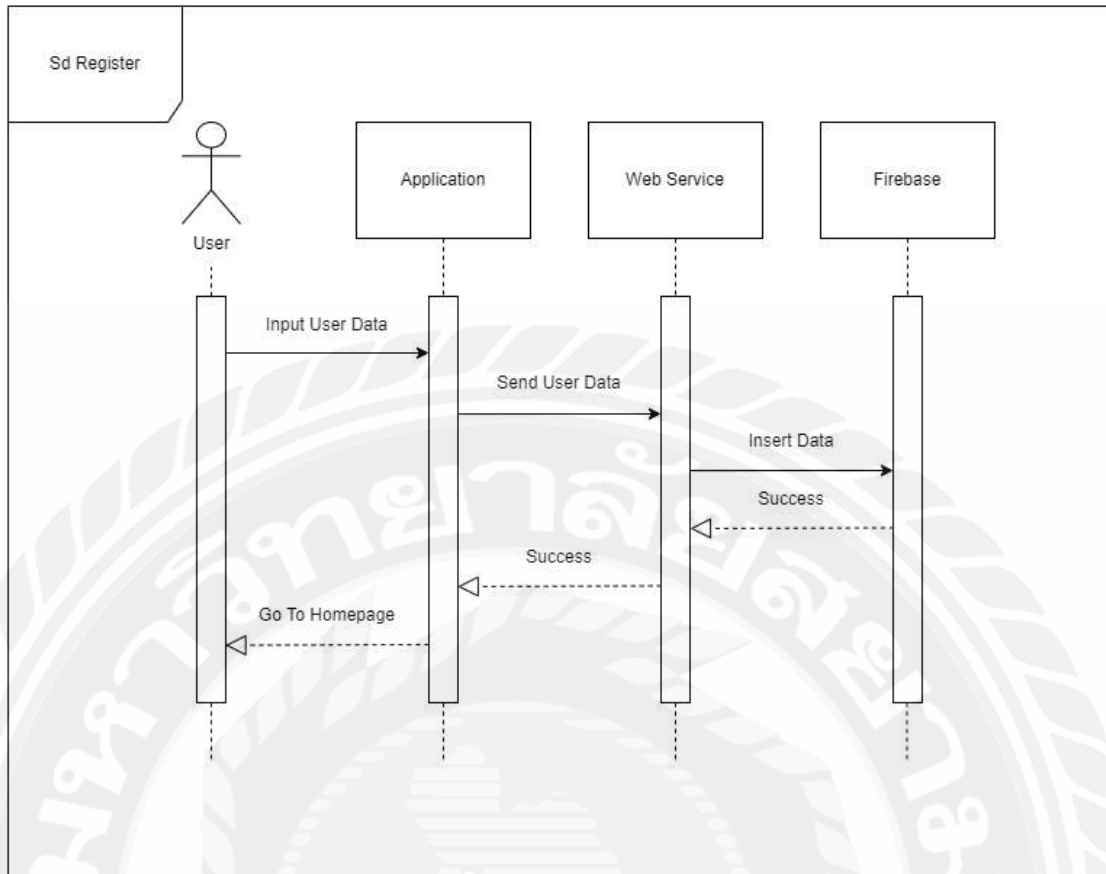
ตารางที่ 3.15 แสดงรายละเอียดของ Use case: Background Notification

Use Case Name	Background Notification
Use Case ID	UC15
Brief Description	การแจ้งเตือน
Primary Actors	User
Secondary Actors	-
Preconditions	หลังจากผู้ใช้งานมีการเปลี่ยนสถานะของสัตว์เลี้ยง
Main Flow	<ol style="list-style-type: none"> 1. ยูสเคสจะเริ่มขึ้นเมื่อผู้ใช้งานทำการ Adopt Approve 2. ระบบจะทำการแจ้งเตือนไปที่โทรศัพท์ของเจ้าของโพสต์หรือผู้ใช้ที่ได้รับการดูแล
Post Condition	-
Alternative Flows	-

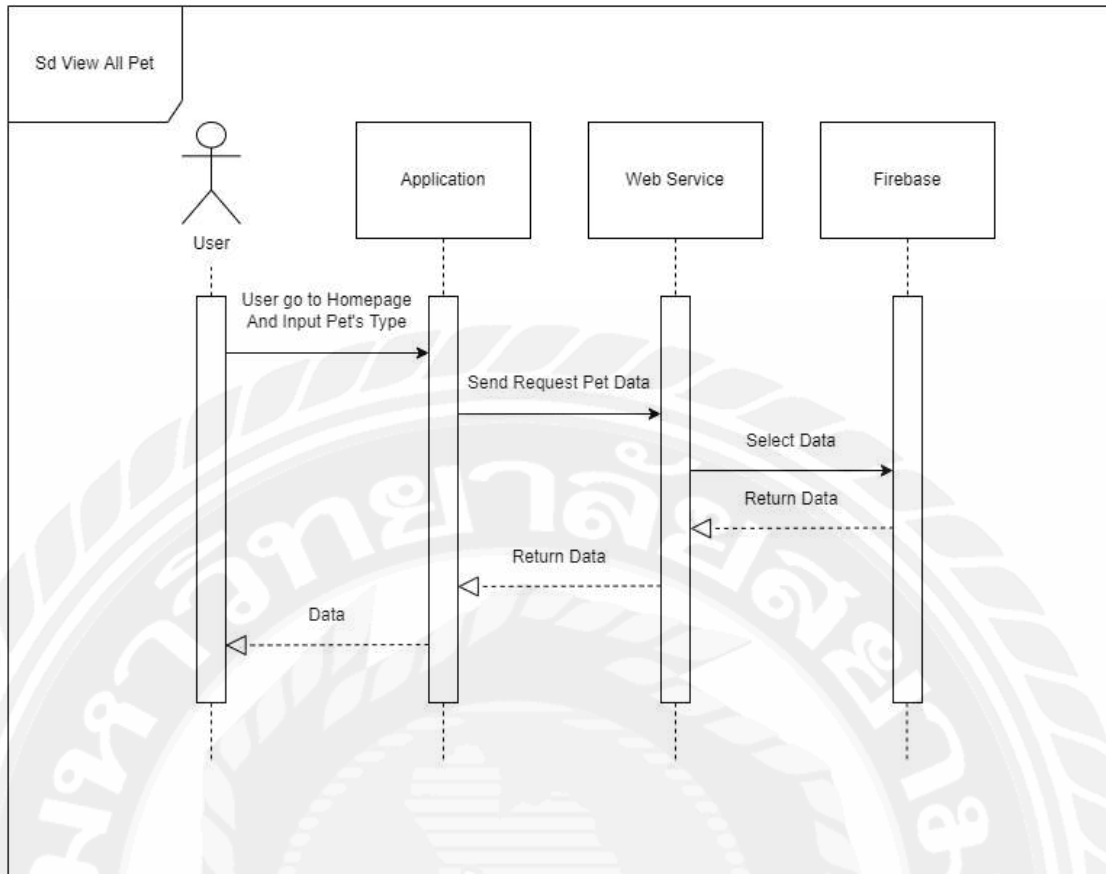
3.4 แสดงขั้นตอนการทำงานของฟังก์ชัน (Sequence Diagram)



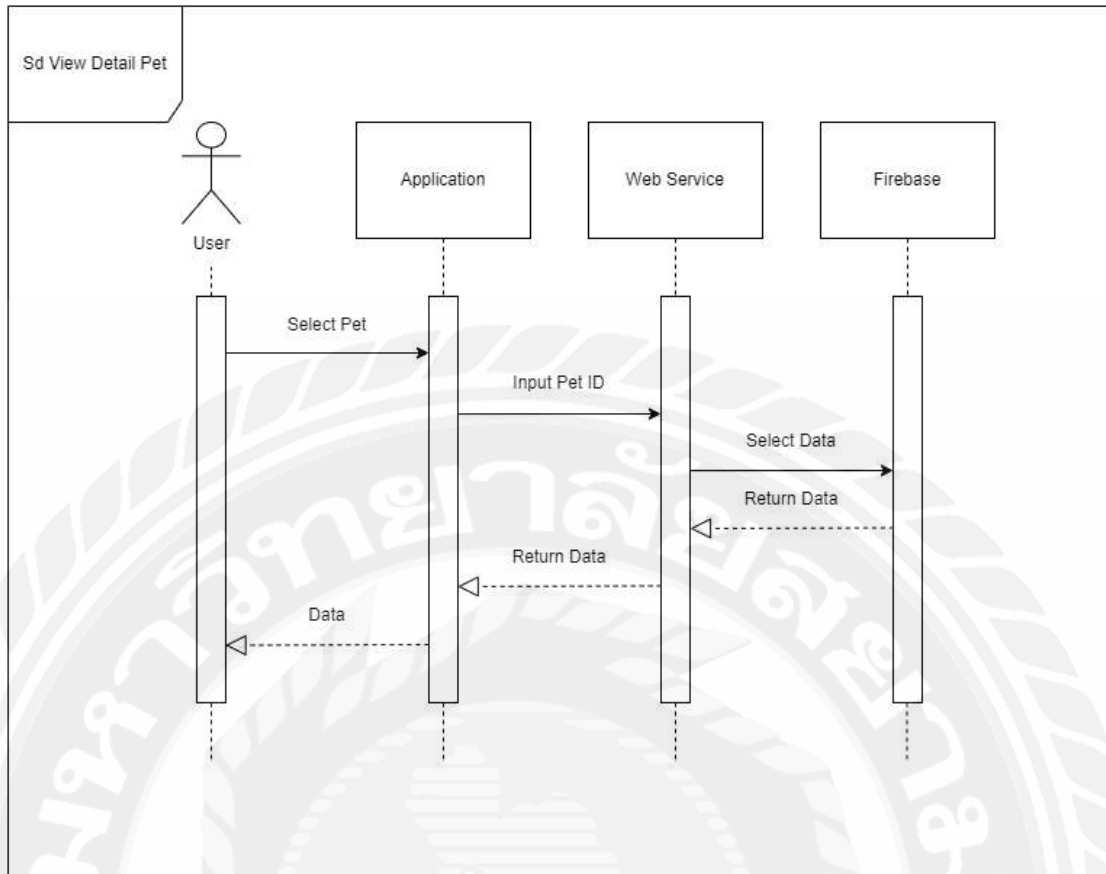
รูปที่ 3.2 Sequence diagram : Login



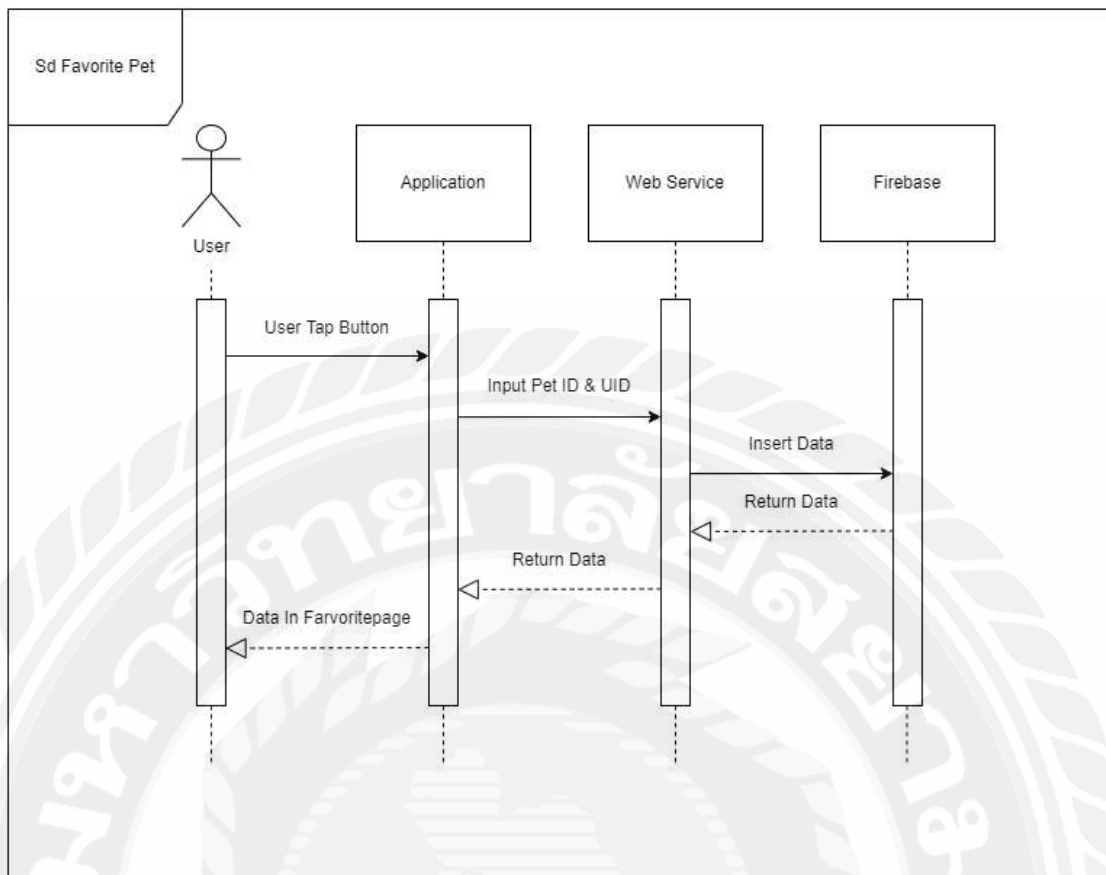
รูปที่ 3.3 Sequence diagram : Register



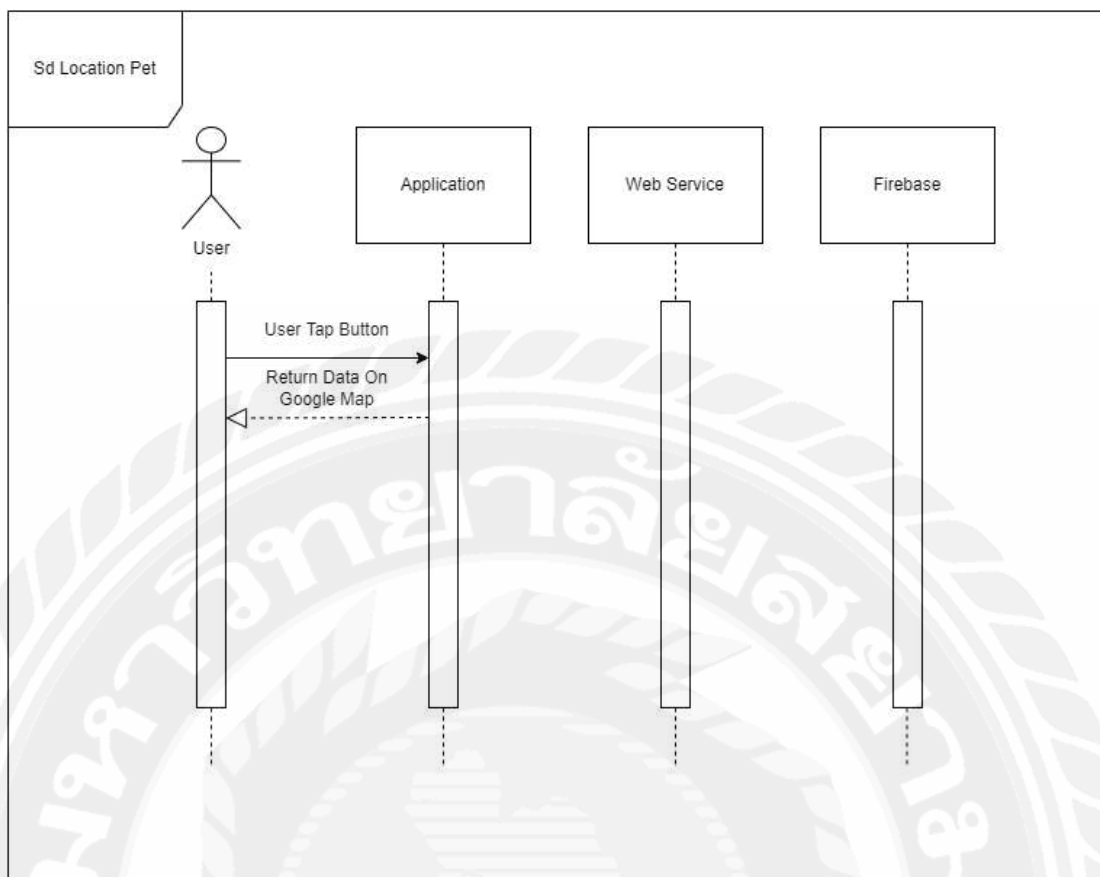
รูปที่ 3.4 Sequence diagram : View All Pet



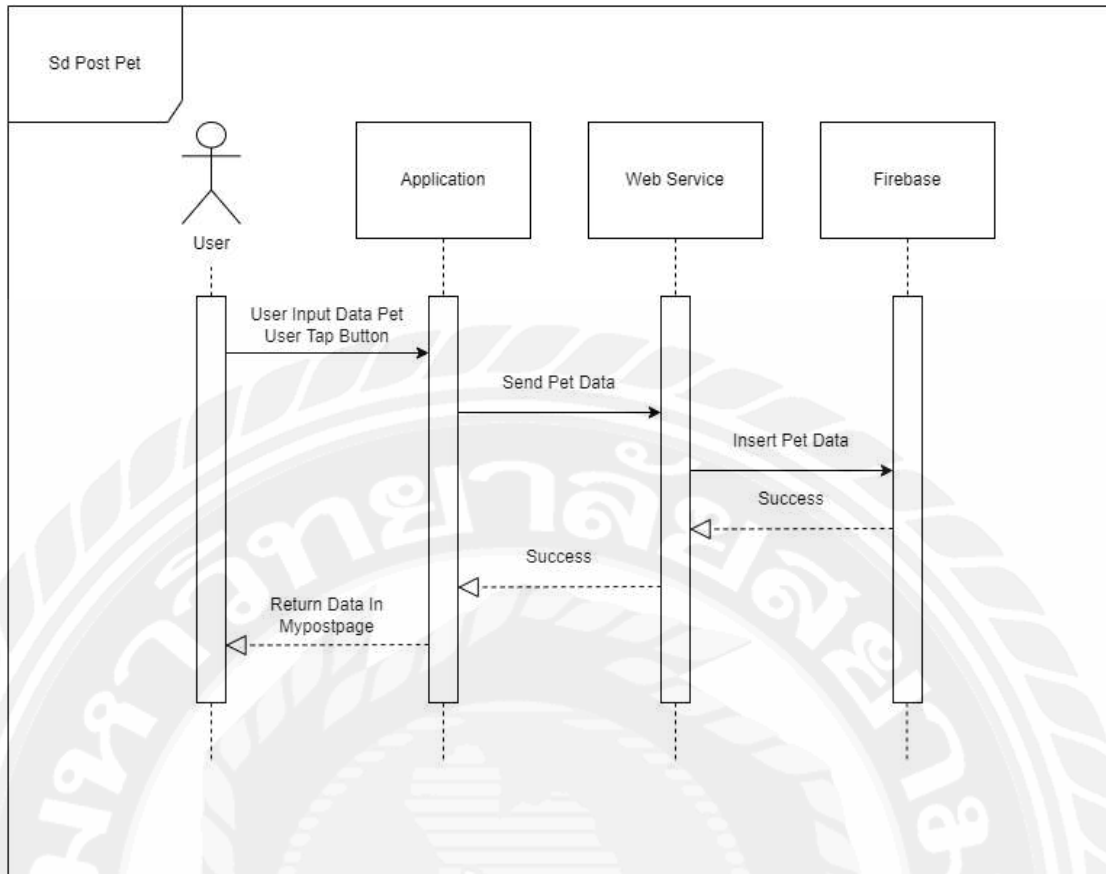
รูปที่ 3.5 Sequence diagram : View Detail Pet



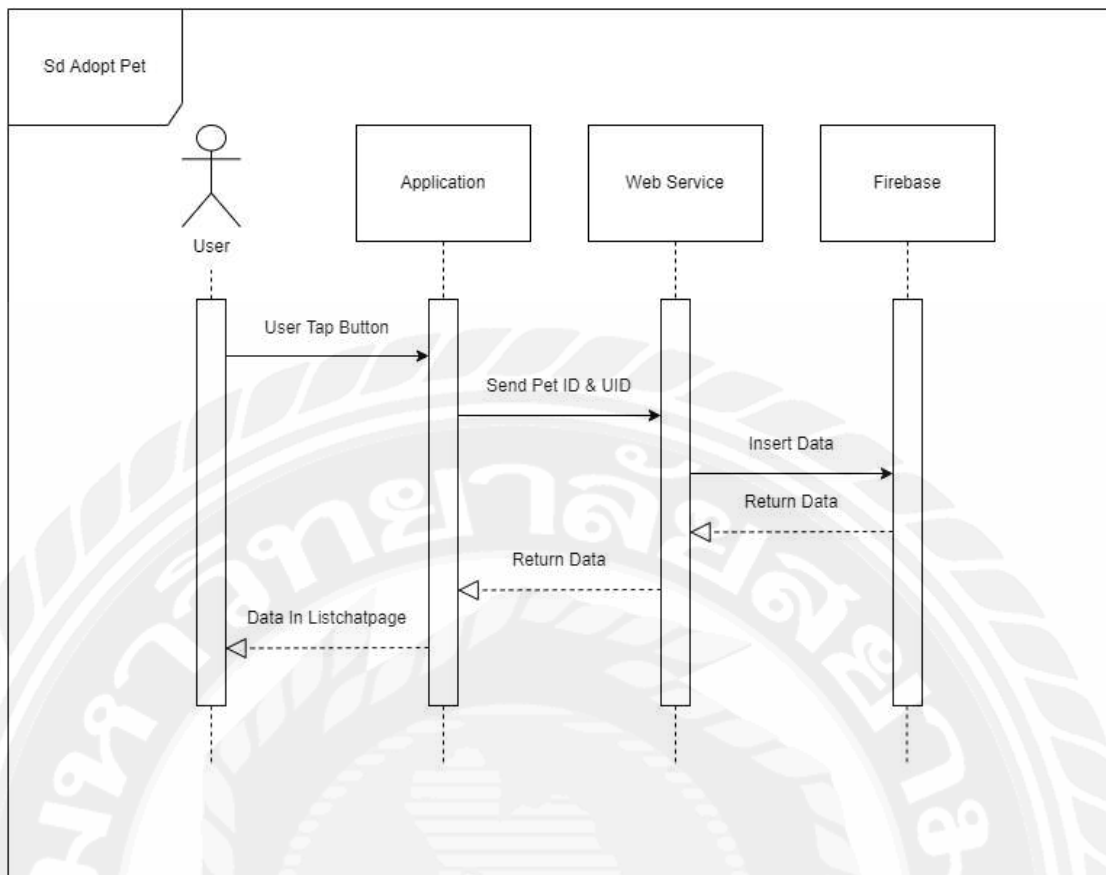
รูปที่ 3.6 Sequence diagram : Favorite Pet



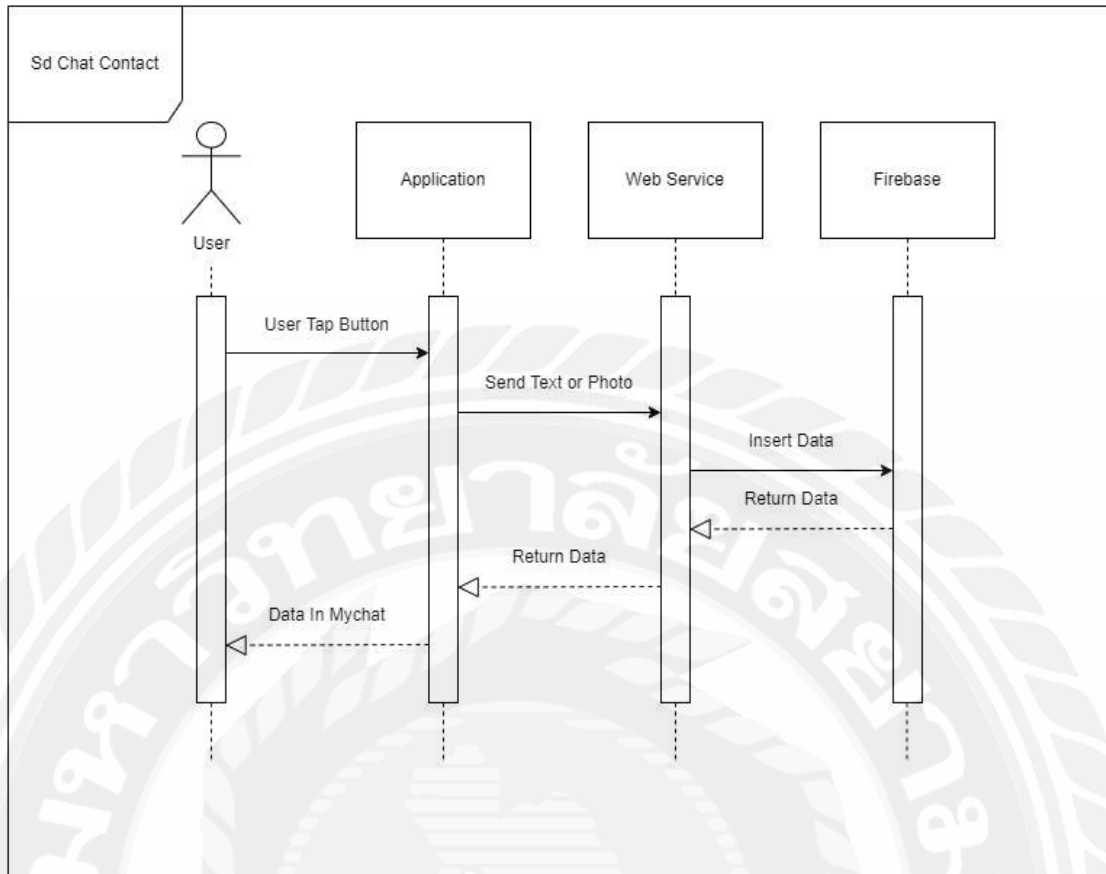
รูปที่ 3.7 Sequence diagram : Location Pet



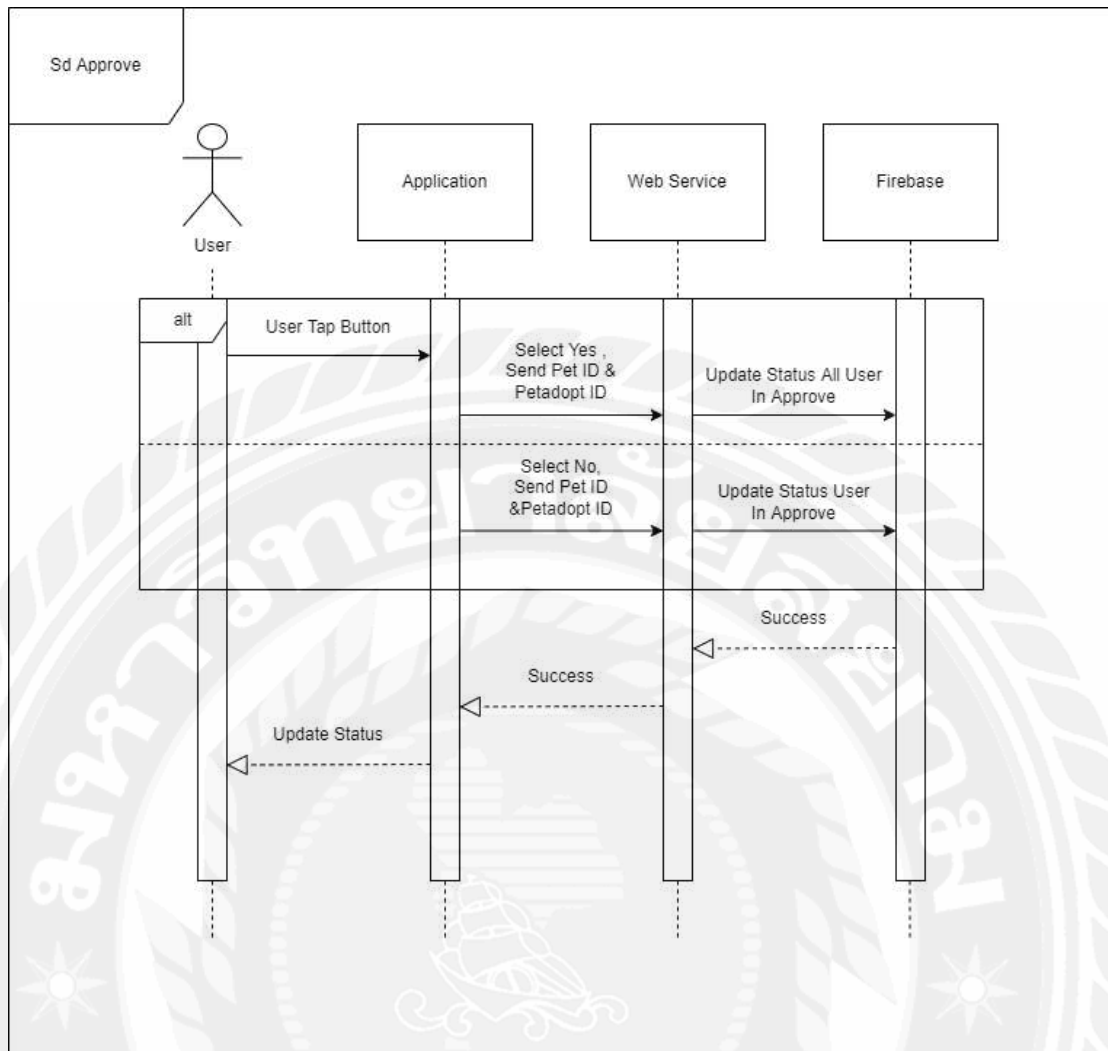
รูปที่ 3.8 Sequence diagram : Post Pet



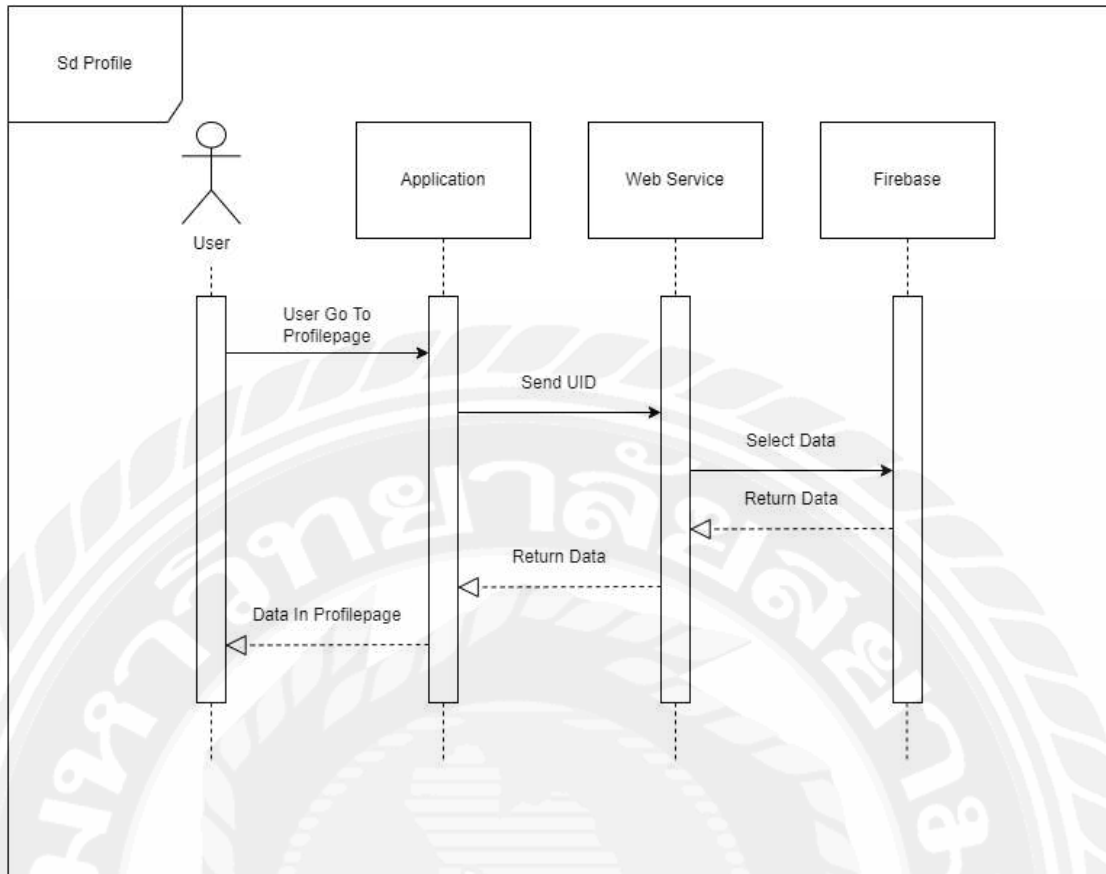
รูปที่ 3.9 Sequence diagram : Adopt Pet



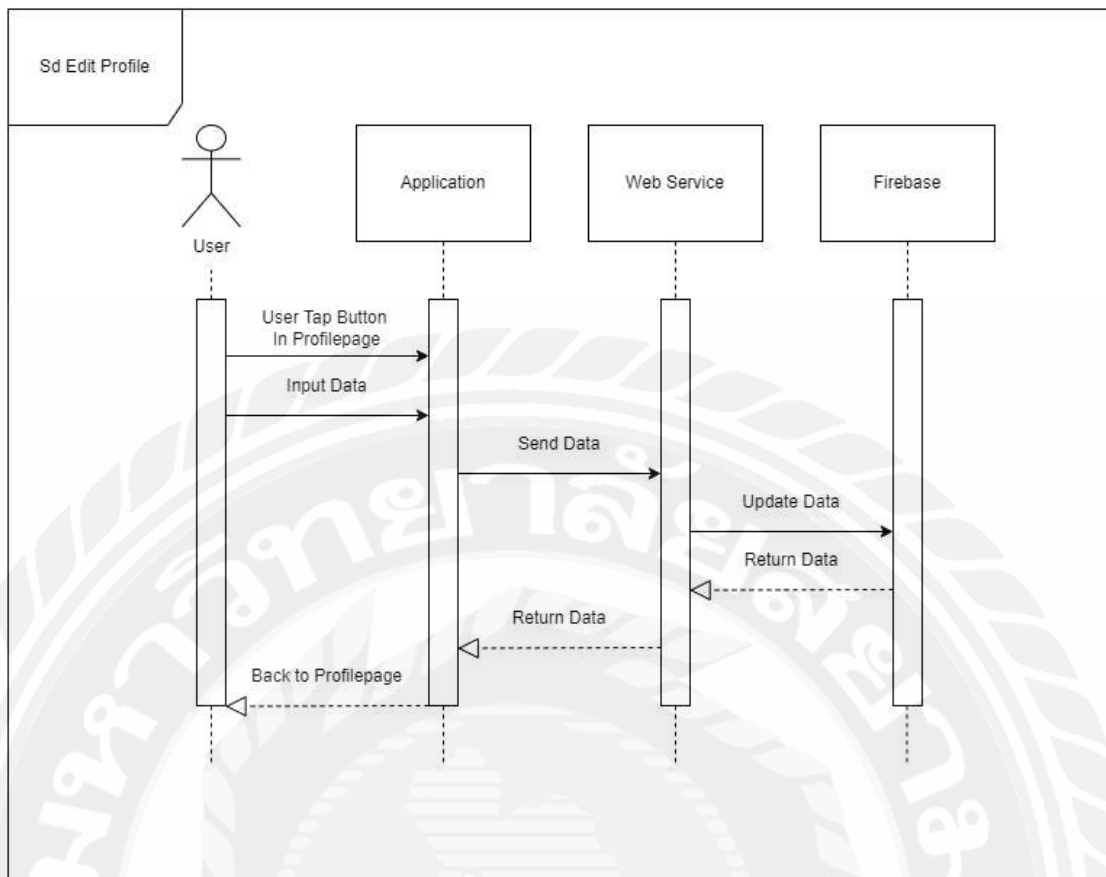
รูปที่ 3.10 Sequence diagram : Chat Contact



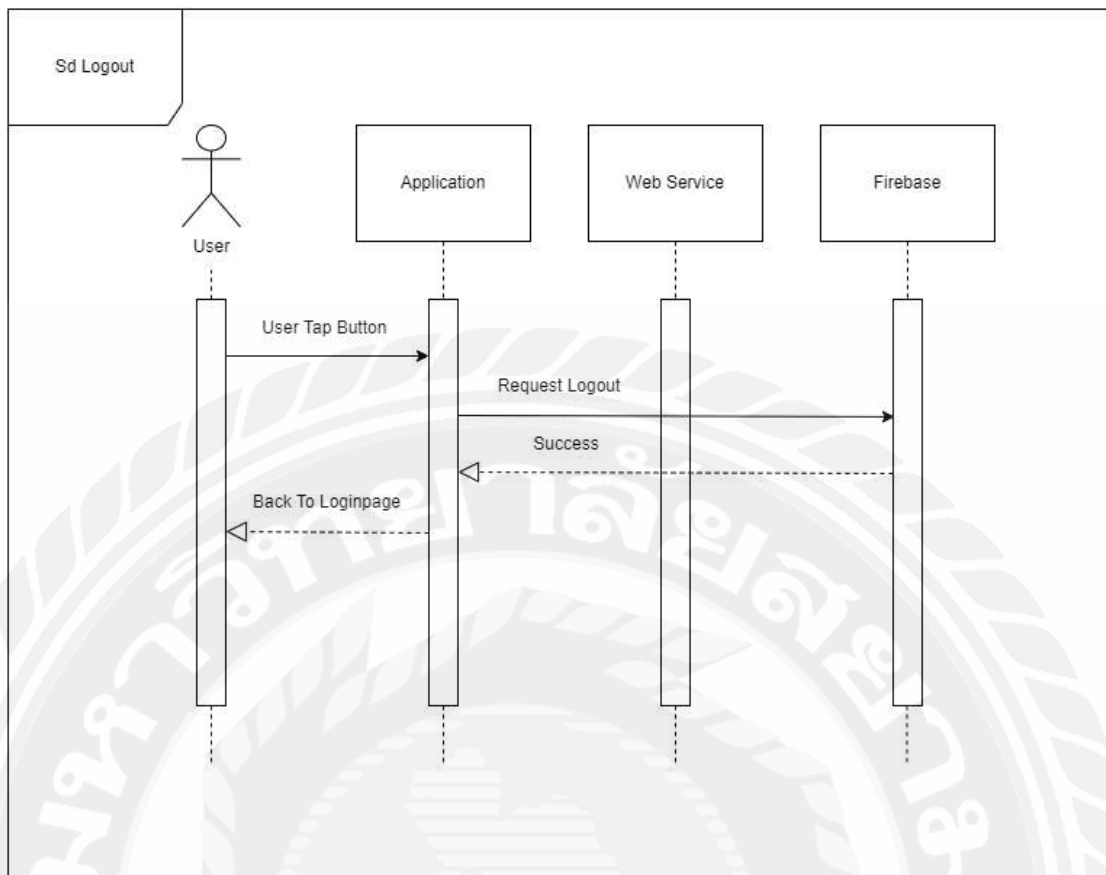
รูปที่ 3.11 Sequence diagram : Approve



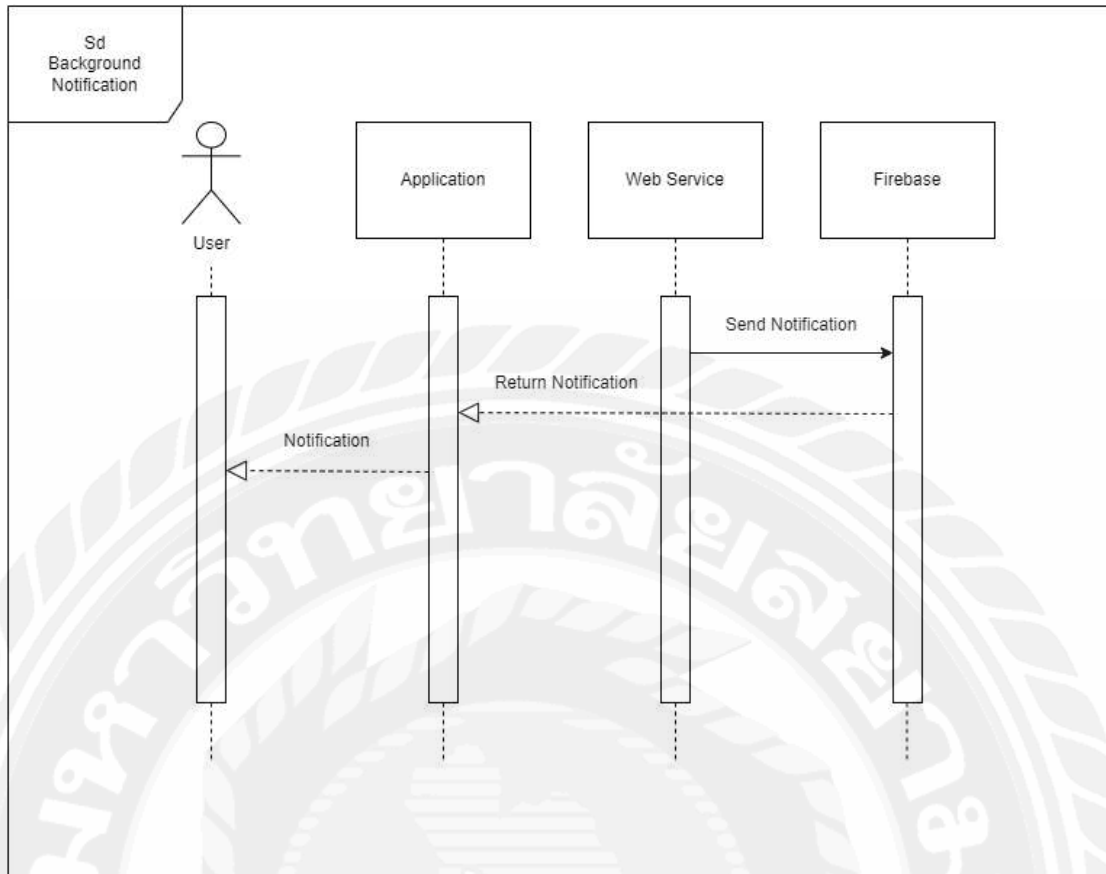
รูปที่ 3.12 Sequence diagram : Profile



รูปที่ 3.13 Sequence diagram : Edit Profile

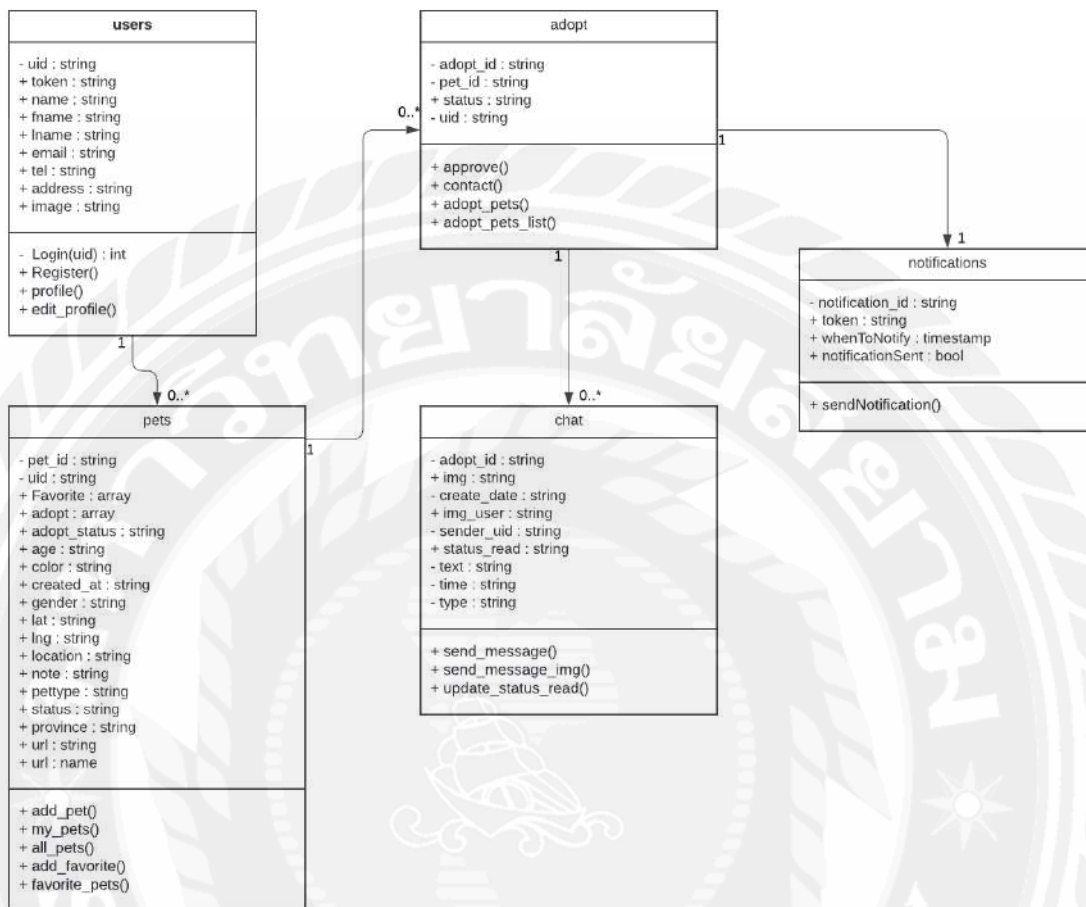


รูปที่ 3.14 Sequence diagram : Logout



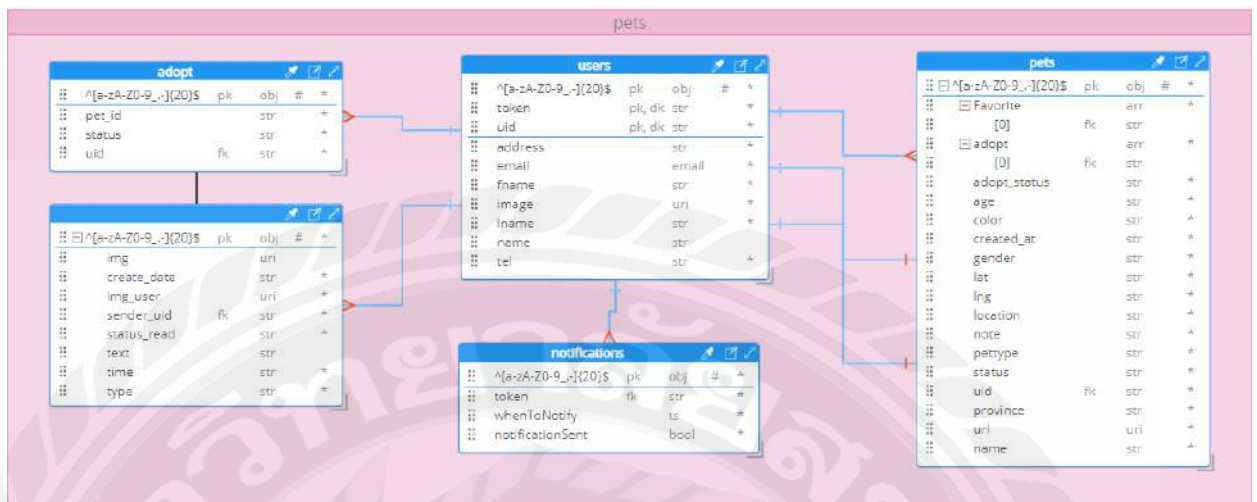
รูปที่ 3.15 Sequence diagram : Background Notification

3.1 แผนภาพแสดงองค์ประกอบคลาส (Class Diagram)



รูปที่ 3.16 Class Diagram ของแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง

3.6 โครงสร้างของข้อมูลใน NoSQL Database



รูปที่ 3.17 โครงสร้างของข้อมูล (NoSQL Structure)

ตารางที่ 3.16 users

name	Description	Data Type
uid	id ของ user ที่ได้จาก auth	String
token	token ของโทรศัพท์	String
name	ชื่อที่ได้จากการ login	String
fname	ชื่อ	String
lname	นามสกุล	String
email	จดหมายอิเล็กทรอนิกส์	String
tel	เบอร์โทรศัพท์	String
image	รูปภาพของ user	String

ตารางที่ 3.17 pets

name	Description	Data Type
Favorite	เก็บ uid ของ user ที่คลิกถูกใจ	array
adopt	เก็บ uid ของ user ที่คลิกขอรับเลี้ยง	array
adopt_status	สถานะของการรับเลี้ยง	String
age	อายุสัตว์เลี้ยง	String
color	สีสัตว์เลี้ยง	String
created_at	วันที่สร้างโพสต์	String
gender	เพศสัตว์เลี้ยง	String
lat	ละติจูด พิกัดทางภูมิศาสตร์	String
lng	ลองจิจูด พิกัดทางภูมิศาสตร์	String
location	ที่อยู่ของสัตว์เลี้ยง	String
name	ชื่อสัตว์เลี้ยง	String
note	รายละเอียดเพิ่มเติม	String
pettype	ประเภทสัตว์เลี้ยง	String
province	จังหวัด	String
status	สถานะกำหนด ว่ามีเจ้าของหรือ จรจัด	String
uid	uid ของผู้โพสต์	String
url	รูปภาพสัตว์เลี้ยง	String

ตารางที่ 3.18 adopt

name	Description	Data Type
pet_id	id ของสัตว์เลี้ยง	String
status	สถานะของการได้รับดูแล	String
uid	id user ที่ขอรับเลี้ยง	String

ตารางที่ 3.19 chat

name	Description	Data Type
create_date	วันที่สร้าง	String
img_user	รูปภาพ user	String
sender_uid	uid ของผู้ส่งข้อความ	String
status_read	สถานะการอ่านข้อความ	String
text	ข้อความ	String
time	เวลาที่ส่งข้อความ	String
type	ประเภทข้อความ	String
img	รูปภาพ	String

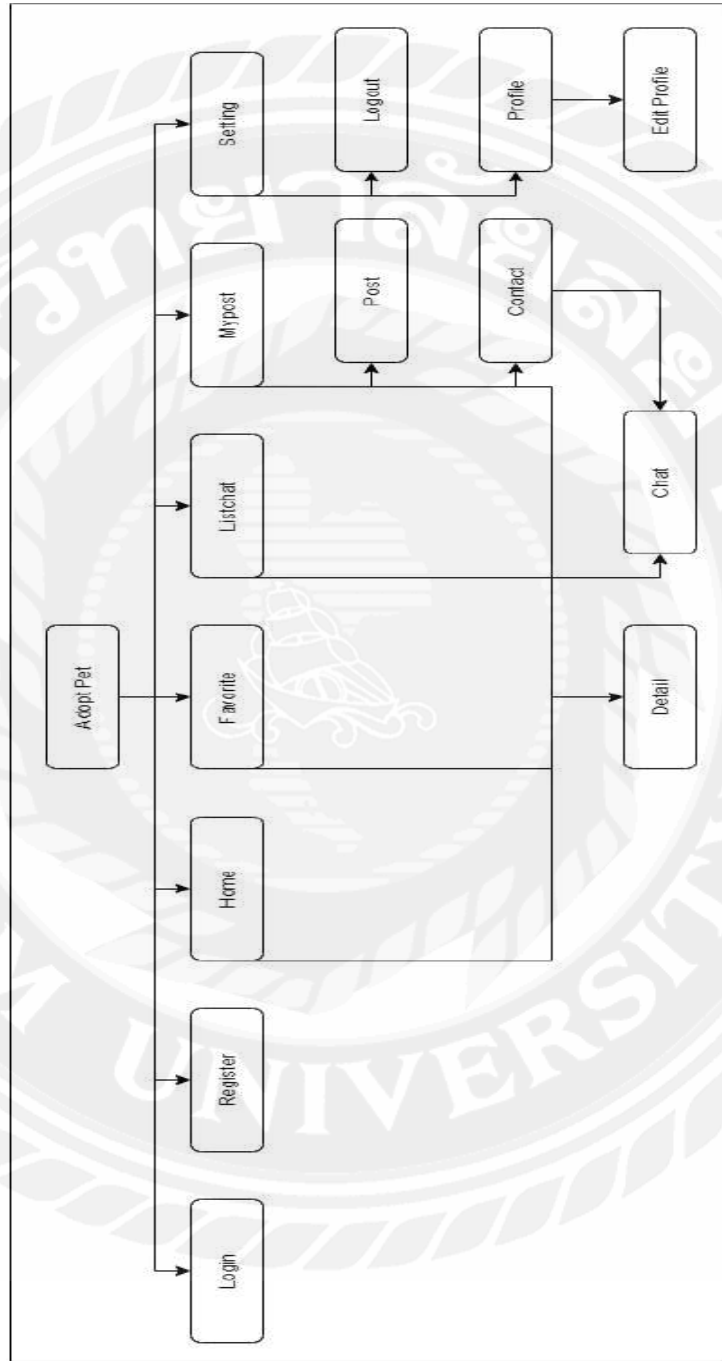
ตารางที่ 3.20 notifications

name	Description	Data Type
notificationSent	สถานะของการส่งแจ้งเตือน	boolean
token	id ของโทรศัพท์	String
whenToNotify	วันเวลาที่ต้องการส่งแจ้งเตือน	timestamp

บทที่ 4

การออกแบบทางกายภาพ

4.1 โครงสร้างของแอปพลิเคชัน (Application Structure Map)



รูปที่ 4.1 โครงสร้างของแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง

ตารางที่ 4.1 คำอธิบายรายละเอียดของโครงสร้างของแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง

หน้าจอ	คำอธิบายรายละเอียด
Login	การเข้าสู่ระบบโดยผ่านบัญชี Facebook หรือ Google
Register	การลงทะเบียนเพื่อใช้งานครั้งแรก
Home	เป็นหน้าหลักของระบบ โดยจะแสดงสิ่งที่ผู้ใช้โพสต์เป็นหมวดหมู่ มีทั้งหมด 4 หมวด <ol style="list-style-type: none"> 1. All Pet 2. Dog 3. Cat 4. Other
Detail	เป็นหน้าแสดงรายละเอียดของสัตว์เลี้ยงตัวนั้นๆ โดยจะแสดงดังนี้ <ul style="list-style-type: none"> - ชื่อ - จังหวัด - สถานะ - เพศ - สี - อายุ - ผู้โพสต์ - คำอธิบายเพิ่มเติม - สถานที่ <p>และสามารถคลิก Favorite สัตว์เลี้ยงที่ถูกใจได้จากหน้านี้</p>
Favorite	เป็นหน้าแสดงสัตว์เลี้ยงที่ผู้ใช้ถูกใจและสามารถคลิกดูรายละเอียดได้
Listchat	เป็นหน้าแสดงสัตว์เลี้ยงที่ผู้เข้ามาเพื่อพูดคุยกับเจ้าของโพสต์
Chat	เป็นหน้าแสดงการพูดคุยระหว่างผู้โพสต์กับผู้ต้องการรับเลี้ยง
Mypost	เป็นหน้าแสดงโพสต์ที่ผู้ใช้เป็นผู้โพสต์
Post	เป็นหน้าใส่รายละเอียดของสัตว์เลี้ยงที่ผู้ใช้ต้องการโพสต์ โดยต้องกรอกข้อมูลดังนี้ <ul style="list-style-type: none"> - รูป - สถานะ - ประเภทสัตว์ - ชื่อ - เพศ

	<ul style="list-style-type: none"> - ลี - วันเกิด - ที่อยู่ - จังหวัด - คำอธิบายเพิ่มเติม
Contact	เป็นหน้าแสดงผู้ใช้ที่ต้องการรับเลี้ยงสัตว์ของผู้โพสต์
Setting	เป็นหน้าการตั้งค่าจะมีส่วนของ Profile และ Logout
Profile	เป็นหน้าแสดงข้อมูลของผู้ใช้
Edit Profile	<p>เป็นการจัดการแก้ไขข้อมูลของผู้ใช้ โดยข้อมูลที่สามารถแก้ไขได้มีดังนี้</p> <ul style="list-style-type: none"> - รูป - ชื่อ - นามสกุล - เบอร์โทรศัพท์ - ที่อยู่

4.2 การออกแบบส่วนติดต่อผู้ใช้งาน (User Interface Design)

แอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยง ทำหน้าที่เป็นสื่อกลางสำหรับกลุ่มคนที่ต้องการหาบ้านให้สัตว์เลี้ยงหรือต้องการรับเลี้ยง ในการออกแบบส่วนติดต่อของผู้ใช้นั้นจะยึดหลัก UX/ UI เป็นหลัก โดยคำนึงถึงความพอใจและการใช้งานที่ง่ายต่อผู้ใช้ด้วยการออกแบบที่เรียบง่าย โดยผู้ใช้สามารถใช้ระบบงานได้ด้วยตนเอง ดังนั้นรูปแบบ ตัวอักษรการจัดวางตำแหน่งออบเจกต์ต่างๆ จะเป็นที่คุ้นเคยของผู้ใช้อยู่แล้ว โดยมีรายละเอียดดังนี้

PET ADOPT

Facebook

Google

รูปที่ 4.2 หน้าเข้าสู่ระบบ (Login)

จากรูปที่ 4.2 แสดงหน้าเข้าสู่ระบบ (Login) เป็นส่วนของหน้าแรกเมื่อผู้ใช้เข้าใช้งานแอปพลิเคชัน ในหน้าจอนี้จะมีปุ่มให้คลิกเลือกอินอยู่สองปุ่ม คือ สามารถเลือกอินด้วยบัญชี Facebook และบัญชี Google ผู้ใช้สามารถเลือกอินได้ตามต้องการ

สร้างบัญชีผู้ใช้



ชื่อ

นามสกุล

เบอร์โทรศัพท์

ที่อยู่

ยืนยัน

ย้อนกลับ

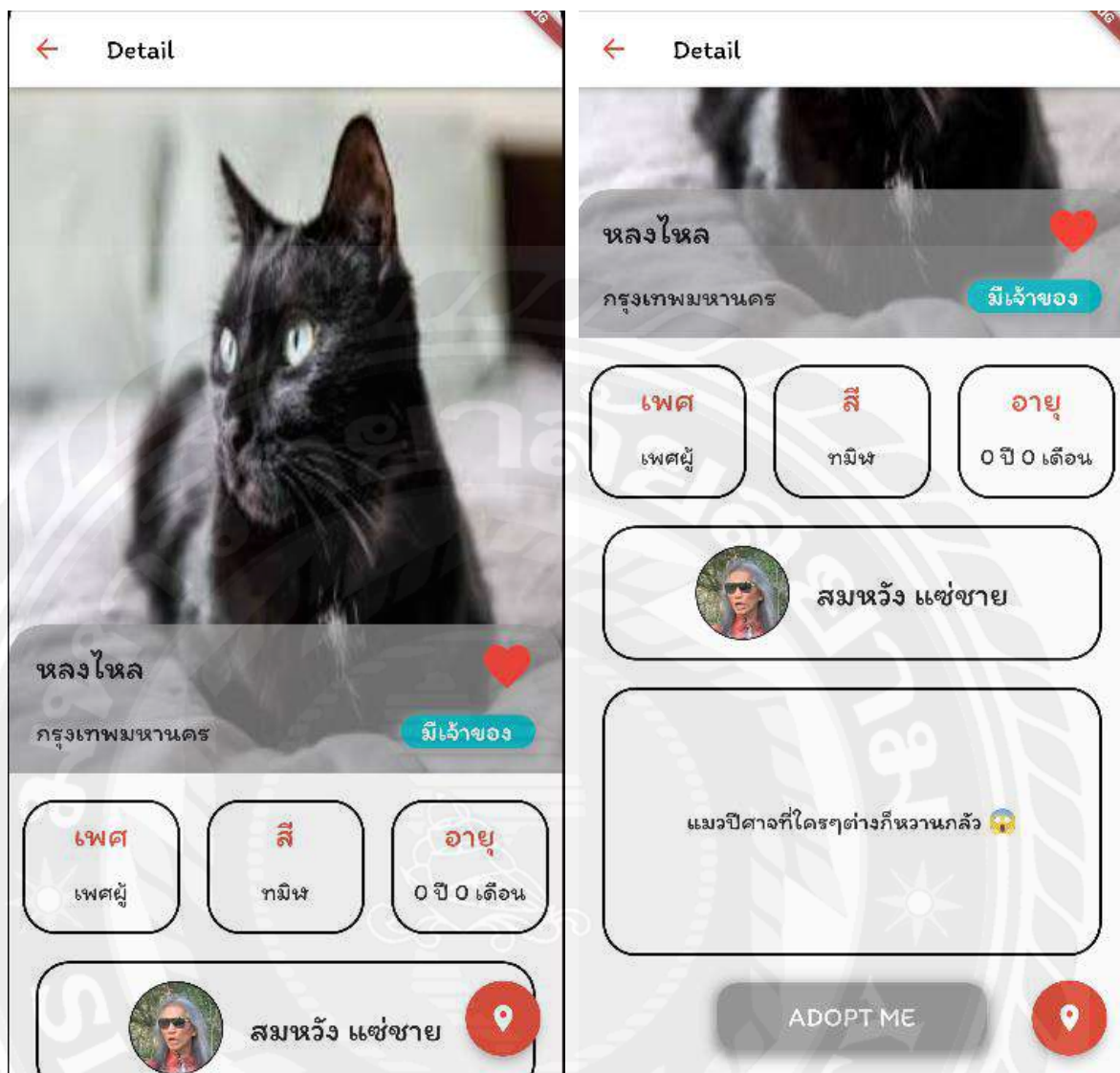
รูปที่ 4.3 หน้าลงทะเบียน (Register)

จากรูปที่ 4.3 แสดงหน้าสำหรับลงทะเบียน (Register) เป็นหน้าใส่รายละเอียดของผู้ใช้งาน โดยจะมาที่หน้านี้เฉพาะที่ผู้ใช้งานที่เข้าใช้งานครั้งแรกเท่านั้น เมื่อผู้ใช้กรอกรายละเอียดของผู้ใช้เสร็จสิ้น ในการเข้าระบบครั้งต่อไปผู้ใช้จะไม่ต้องกรอกข้อมูลอีก โดยผู้ใช้ต้องกรอกข้อมูล ชื่อ, นามสกุล, เบอร์โทรศัพท์ และที่อยู่



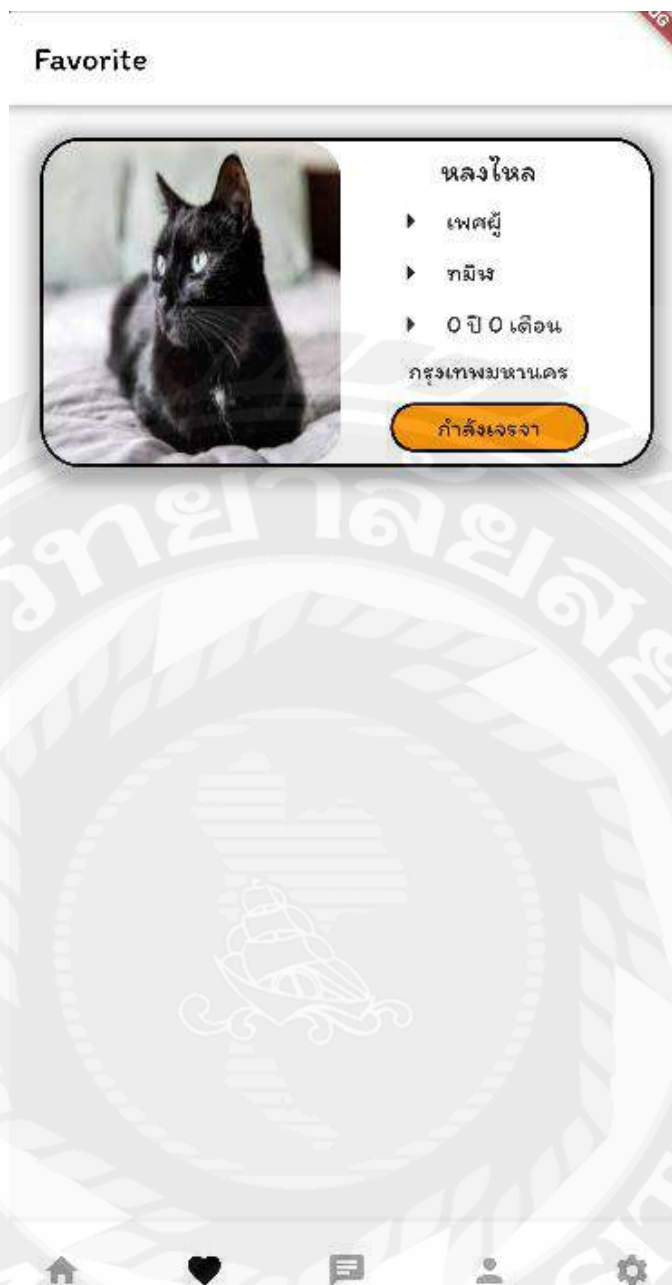
รูปที่ 4.4 หน้าแรก (Home)

จากรูปที่ 4.4 แสดงหน้าแรก (Home) โดยจะแสดงฟีดการอัปเดตสัตว์เลี้ยงที่ผู้ใช้งานนำมาโพสต์หาบ้าน โดยจะแบ่งเป็น 4 ส่วนคือ 1.All 2.Dog 3.Cat 4.Other และยังสามารถคลิกที่รูปเพื่อเข้าไปดูรายละเอียดของสัตว์เลี้ยงตัวนั้น ๆ ได้ โดยหน้าแรกจะบอกรายละเอียดของสัตว์เลี้ยงไว้บางส่วน ได้แก่ ชื่อสัตว์เลี้ยง, จังหวัดที่สัตว์เลี้ยงอยู่, เพศ, สี, อายุ และจำนวนผู้ที่สนใจจะรับเลี้ยงสัตว์เลี้ยงตัวนั้นๆ



รูปที่ 4.5 หน้ารายละเอียดของสัตว์เลี้ยง (Detail)

จากรูปที่ 4.5 แสดงหน้ารายละเอียดของสัตว์เลี้ยง (Detail) โดยจะแสดงรายละเอียดของสัตว์เลี้ยงนั้น ๆ โดยจะมีรูปของสัตว์ ชื่อของสัตว์ จังหวัดที่อยู่ สถานะของสัตว์ เพศ สี อายุ ผู้โพสต์ และคำอธิบายเพิ่มเติมเกี่ยวกับสัตว์เลี้ยงตัวนั้น และหน้านี้ยังสามารถคลิกดู Location ที่สัตว์เลี้ยงตัวนั้นอยู่ได้ ถ้าผู้ใช้ชื่นชอบก็สามารถคลิกหัวใจไว้เพื่อเก็บไว้ดูย้อนหลังได้ที่หน้า Favorite แต่ถ้าอยากรับเลี้ยงก็สามารถคลิกเลือก Adopt Me เพื่อพูดคุยกับเจ้าของโพสต์ได้



รูปที่ 4.6 หน้าสัตว์เลี้ยงที่ถูกใจ (Favorite)

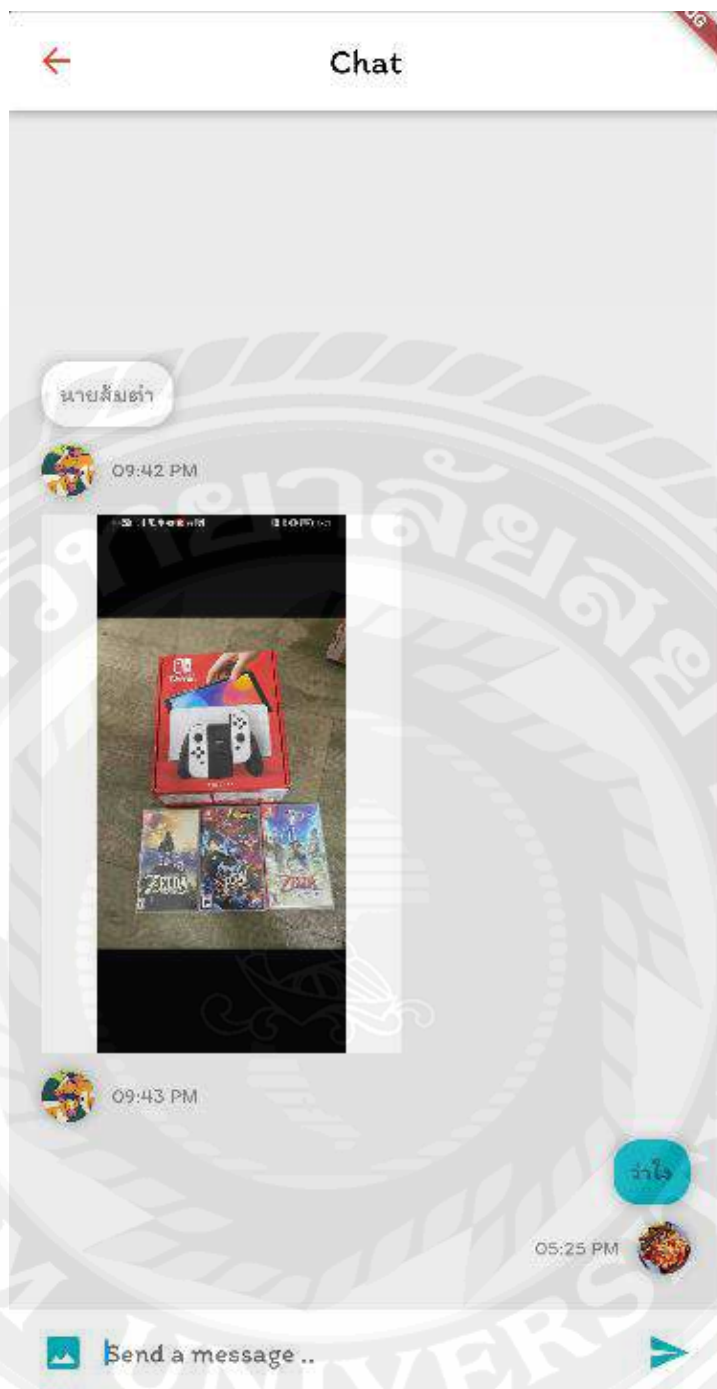
จากรูปที่ 4.6 แสดงหน้าถูกใจ (Favorite) จะแสดงสัตว์เลี้ยงที่ผู้ใช้ถูกใจโดยจะมีรายละเอียดบางส่วน และยังมีสถานะของสัตว์เลี้ยงตัวนั้นว่ามีผู้ใช้อื่นรับเลี้ยงไปแล้วหรือยัง หรือพูดคุยกับเจ้าของโพสต์อยู่ และยังสามารถคลิกเข้าไปดูหน้ารายละเอียด (Detail) ของสัตว์เลี้ยงได้อีกด้วย

ListChat



รูปที่ 4.7 หน้ารายการสัตว์เลี้ยงที่เลือก (Listchat)

จากรูปที่ 4.7 แสดงหน้า Listchat จะแสดงสัตว์เลี้ยงที่ผู้ใช้นั้นคลิก Adopt Me มาแล้วจะมีปุ่มแชทเพื่อให้ผู้ใช้ได้ติดต่อกับเจ้าของโพสต์ และยังสามารถคลิกเข้าไปดูรายละเอียด และแสดงสถานะของสัตว์เลี้ยงอีกด้วย



รูปที่ 4.8 หน้าสนทนา (Chat)

จากรูป 4.8 แสดงหน้าการสนทนาระหว่างผู้โพสต์กับผู้ที่จะรับเลี้ยง โดยสามารถส่งได้ทั้งข้อความและรูป โดยจะแสดงเวลาที่ผู้ใช้ส่งข้อความและรูปหากัน และแสดงสถานะการอ่านว่าอีกฝ่ายอ่านแล้วหรือยังไม่ได้อ่านเช่นกัน



รูปที่ 4.9 หน้ารายการสัตว์เลี้ยงที่โพสต์หาบ้าน (Mypost)

จากรูปที่ 4.9 แสดงหน้า Mypost โดยจะแสดงโพสต์ที่ผู้ใช้โพสต์หาบ้านให้สัตว์เลี้ยง สามารถดูรายละเอียดกับสถานะได้ และยังมีปุ่มผู้ติดต่อเพื่อให้ผู้ใช้รู้ว่ามิใช่คนไหนคลิก Adopt สัตว์ที่ผู้ใช้โพสต์ และยังสามารถคลิกปุ่มบวกที่มุมขวาล่างเพื่อไปสร้างโพสต์หาบ้านให้สัตว์เลี้ยง ของผู้ใช้ได้

← Post

สถานะ

ประเภทสัตว์

ชื่อ

เพศ

สี

วันเกิด

ที่อยู่

จังหวัด

Description

รูปที่ 4.10 หน้าโพสต์หาบ้านให้สัตว์เลี้ยง (Post)

จากรูปที่ 4.10 แสดงหน้าสำหรับโพสต์สัตว์เลี้ยงที่ต้องการหาบ้าน โดยจะให้ผู้ผู้ใช้ได้รายละเอียดของสัตว์ที่ผู้ใช้ต้องการจะโพสต์หาบ้านให้ ประกอบด้วย รูป, สถานะ, ประเภทของสัตว์เลี้ยง, ชื่อ, เพศ, สี, วันเกิด, ที่อยู่, จังหวัด และรายละเอียด

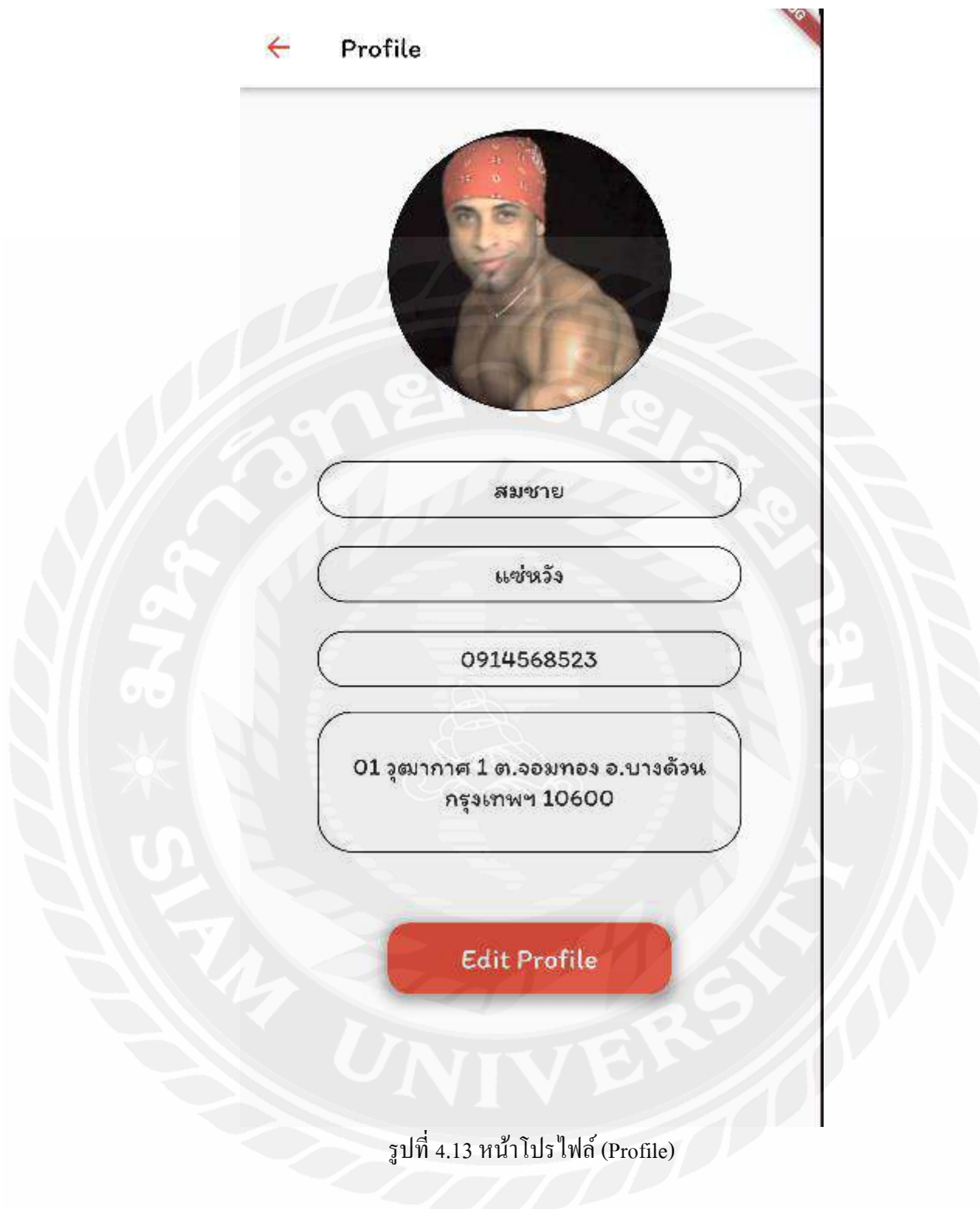


รูปที่ 4.11 หน้าติดต่อ (Contact)

จากรูปที่ 4.11 แสดงหน้าผู้ติดต่อ โดยจะแสดงผู้ใช้ที่ต้องการรับเลี้ยงสัตว์ของผู้โพสต์ซึ่งผู้โพสต์จะสามารถเหตุผลคุยติดต่อกับผู้ใช้ที่ต้องการรับเลี้ยงสัตว์ได้และยังให้การตัดสินใจได้ว่าผู้ใช้นี้สมควรได้รับสัตว์ไปเลี้ยงไหม



จากรูปที่ 4.12 แสดงหน้าการตั้งค่า โดยจะแสดงปุ่มเข้าไปหน้าโปรไฟล์ของผู้ใช้และปุ่ม Logout เอาไว้สำหรับผู้ใช้อยากจะเปลี่ยนไอดีหรือเข้ารหัสผิด

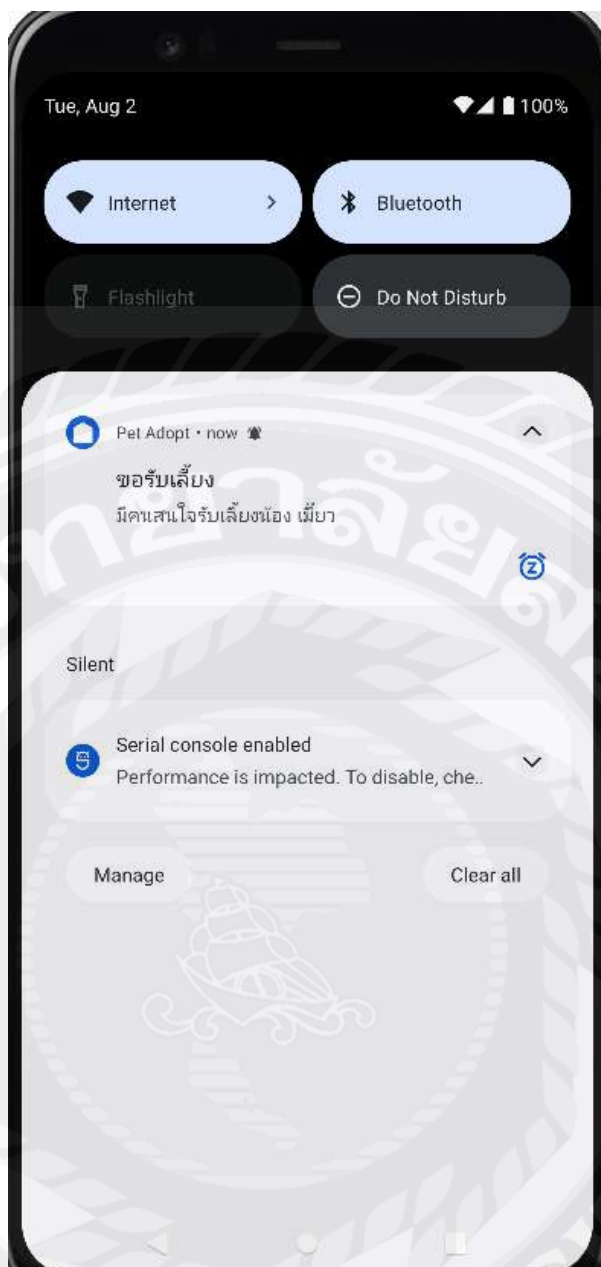


รูปที่ 4.13 หน้าโปรไฟล์ (Profile)

จากรูปที่ 4.13 แสดงหน้าข้อมูลผู้ใช้โดยจะแสดงข้อมูลส่วนตัวของผู้ใช้ซึ่งสามารถแก้ไขได้ด้วยการคลิกปุ่ม Edit Profile

รูปที่ 4.14 หน้าแก้ไขโปรไฟล์ (Edit Profile)

จากรูปที่ 4.14 แสดงหน้าแก้ไขข้อมูลส่วนตัว ผู้ใช้สามารถแก้ไขรูปภาพหรือข้อมูลต่างๆ ของผู้ใช้ได้ เมื่อผู้ใช้กรอกข้อมูลเรียบร้อยแล้วให้คลิกปุ่ม Save ระบบพาผู้ใช้กลับไปหน้า Profile พร้อมกับข้อมูลที่ผู้ใช้แก้ไขไว้แล้ว



รูปที่ 4.15 การแจ้งเตือนของแอปพลิเคชัน (Alert Application)

จากรูปที่ 4.15 แสดงการแจ้งเตือนหลังจากที่ผู้ใช้ไม่ได้อยู่ในแอปพลิเคชัน โดยจะแจ้งเตือนตอนที่ผู้มาขอรับเลี้ยงสัตว์ของผู้ใช้ หรือแจ้งเตือนไปยังผู้ที่ผู้โพสต์ให้สิทธิ์รับเลี้ยงผู้นั้น และแจ้งเตือนความคืบหน้าหลังจากผู้รับเลี้ยงสัตว์ไป 7 วันให้ส่งความคืบหน้าของสัตว์เลี้ยงตัวนั้นๆ

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลปริญญานิพนธ์

การพัฒนาแอปพลิเคชันสำหรับหาบ้านให้สัตว์เลี้ยงได้พัฒนาเสร็จสิ้นตามวัตถุประสงค์ที่ตั้งไว้ โดยผู้ใช้งานสามารถนำสัตว์เลี้ยงหรือสัตว์เลี้ยงจรจัดที่ต้องการจะหาบ้านให้สัตว์เลี้ยง มาโพสต์ในแอปพลิเคชันได้โดยการโพสต์รูป ข้อมูลรายละเอียดต่าง ๆ ได้ผ่านแอปพลิเคชัน และผู้ใช้งานยังสามารถหาสัตว์เลี้ยงไปรับเลี้ยงได้ผ่านทางหน้าโฮมโดยจะมีสัตว์เลี้ยงต่าง ๆ ให้ผู้ใช้งานได้เลือกตามที่มีผู้ใช้งานมาโพสต์ โดยสามารถดูรายละเอียดของสัตว์เลี้ยง ที่อยู่ของสัตว์เลี้ยง และสามารถคลิกดูใจไว้เพื่อดูภายหลังได้ หลังจากผู้ใช้ต้องการรับเลี้ยงสัตว์เลี้ยงผู้ใช้สามารถคลิกกดปุ่มเพื่อให้ผู้โพสต์ทราบได้ว่ามีผู้อยากได้รับสัตว์เลี้ยงไปเลี้ยง โดยมีระบบแชทไว้สำหรับให้ผู้โพสต์กับผู้ใช้พูดคุยกัน และผู้โพสต์ยังสามารถเลือกได้ว่าจะให้ผู้รับเลี้ยงคนไหนได้รับไปเลี้ยง เมื่อตกลงพูดคุยกันสำเร็จก็สามารถรับสัตว์เลี้ยงไปได้เลย แล้วทางระบบจะมีแจ้งเตือน 7 วัน เพื่อให้ผู้รับเลี้ยงส่งความคืบหน้าของสัตว์เลี้ยงที่รับไปเลี้ยงด้วยเพื่อป้องกันการนำสัตว์เลี้ยงไปทารุณกรรม

5.2 ข้อดีของระบบ

- 5.2.1 ช่วยให้ผู้เลี้ยงจรจัดมีโอกาสมีที่อยู่อาศัย
- 5.2.2 ช่วยให้องค์กรอิสระสามารถติดตามสัตว์เลี้ยงจรจัดจากผู้ใช้งานบนแอปพลิเคชันได้
- 5.2.3 ช่วยให้ผู้เลี้ยงสัตว์เลี้ยงต่อไม่ไหวได้มีโอกาสหาที่อยู่ใหม่ให้สัตว์เลี้ยงเหล่านั้น
- 5.2.4 ช่วยเป็นสื่อกลางระหว่างผู้ใช้
- 5.2.5 ช่วยลดโรคติดต่อของสัตว์เลี้ยงจรจัด
- 5.2.6 ช่วยให้ผู้เลี้ยงที่อยากเลี้ยงสัตว์เลี้ยงแต่ไม่มีเงินทุนพอที่จะซื้อได้
- 5.2.7 ช่วยให้ผู้เลี้ยงหาสัตว์เลี้ยงที่ต้องการได้
- 5.2.8 ช่วยให้ผู้เลี้ยงดูรายละเอียดสัตว์เลี้ยงก่อนที่จะรับเลี้ยงได้
- 5.2.9 ช่วยให้ผู้เลี้ยงติดต่อกันระหว่างผู้ใช้ได้
- 5.2.10 ช่วยให้ผู้โพสต์หาบ้านให้สัตว์เลี้ยงมีสิทธิ์ในการตัดสินใจในการรับผู้ใช้ที่ต้องการรับไปเลี้ยงดู

5.3 ข้อเสนอแนะ

เพื่อให้ระบบมีประสิทธิภาพมากยิ่งขึ้น ควรพัฒนาฟังก์ชันเพิ่มเติม ดังนี้

- 5.3.1 ให้มีการแจ้งเตือนแบบขณะอยู่ในแอปพลิเคชัน
- 5.3.2 ให้สามารถโพสต์รูปสัตว์เลี้ยงได้หลายๆ รูปใน 1 โพสต์
- 5.3.3 เพิ่มระบบยืนยันตัวตนด้วยเอกสารราชการ
- 5.3.4 เพิ่มในส่วนของการคัดกรองจากหน่วยงานที่มีหน้าที่ในการคัดกรอง ก่อนที่จะนำสัตว์ไปเลี้ยงเพื่อป้องกันการนำสัตว์ไปทารุณกรรม
- 5.3.5 เพิ่มในส่วน of ข้อมูลสัตว์เลี้ยงที่ไม่ทราบที่มา เช่น วันเกิด



บรรณานุกรม

คลาสเมธอด. (2565). ฟิกมาร์ (*Figma*). เข้าถึงได้จาก

<https://dev.classmethod.jp/articles/whats-about-figma/>

บอร์นทูเดฟ. (2563). ภาษาคาร์ท (*Dart Programming Language*). เข้าถึงได้จาก

[มีเดียม. \(2561\). แอนดรอยด์สตูดิโอ \(*Android Studio*\). เข้าถึงได้จาก](https://www.borntodev.com/2020/04/11/เขียน-dart-แบบก้าวกระโดด/เขียนซี. (2560). มายด์</p></div><div data-bbox=)

<https://medium.com/@palmz/เริ่มต้นสร้าง-android-application-พื้นฐานด้วย-android-studio-lab-3sb04-3fda43b07a1>

มายด์พีเอชพี. (2561). ฟลัตเตอร์ เฟรมเวิร์ค (*Flutter Framework*). เข้าถึงได้จาก

<https://www.mindphp.com/บทความ/241-mobile-application/5554-flutter-framework.html>

มายด์พีเอชพี. (2560). วิวสทูดิโอโค้ด (*Visual Studio Code*). เข้าถึงได้จาก

<https://www.mindphp.com/บทความ/microsoft/4829-visual-studio-code.html>

มายด์พีเอชพี. (2565). โพสต์แมน (*Postman*). เข้าถึงได้จาก

<https://www.mindphp.com/บทความ/microsoft/237-free-software/8446-postman.html>

มายด์พีเอชพี. (2563). ไฟร์เบส (*Firebase*). เข้าถึงได้จาก

<https://www.mindphp.com/คู่มือ/73-คืออะไร/3921-what-is-firebase-backend.html>

มายด์พีเอชพี. (2565). โนดส์ทีทเจเอส (*Node.js*). เข้าถึงได้จาก

<https://www.mindphp.com/developer/dev-node-js/8542-1-step-website-nodejs.html>