

แอปพลิเคชันติดตามสัตว์เลี้ยง
Pet track application

| | |
|----------------------|-------------------------|
| นาย ชัยณรงค์ เดชสุภา | รหัสนักศึกษา 5914210004 |
| นางสาว สมฤดี พุ่มพวง | รหัสนักศึกษา 5914210008 |

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตาม
หลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชาเทคโนโลยีสารสนเทศ คณะวิทยาการและเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีมหานคร
ปีการศึกษา 2560

แอปพลิเคชันติดตามสัตว์เลี้ยง

Pet track application

นายชัยณรงค์ เดชสุภา รหัสนักศึกษา 5914210004

นางสาวสมฤดี พุ่มพวง รหัสนักศึกษา 5914210008

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาเทคโนโลยีสารสนเทศ คณะวิทยาการและเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีมหานคร

ปีการศึกษา 2560

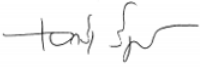


ITEC-2560-03-05


หัวข้อปริญญานิพนธ์ แอปพลิเคชันติดตามสัตว์เลี้ยง
Pet track application

ผู้ร่วมโครงการ นาย ชัยณรงค์ เดชสุภา รหัสนักศึกษา 5914210004
นางสาว สมฤดี พุ่มพวง รหัสนักศึกษา 5914210008


อาจารย์ที่ปรึกษา


.....
(ดร. เอกรัฐ รัชกาญจน์)

ปริญญานิพนธ์ฉบับนี้ ผ่านการพิจารณาจากคณะกรรมการ ดังมีรายชื่อต่อไปนี้


.....
(อาจารย์ พกษา ดวงมาสุข)


.....
(อาจารย์ เขาวริน สกุลวรากลาง)


.....
(ดร. สุรณพีร์ ภูมิวุฒิสาร)

บทคัดย่อ

วัตถุประสงค์ของปริญญานิพนธ์ฉบับนี้เป็นการศึกษาและออกแบบแอปพลิเคชันติดตามสัตว์เลี้ยงเป็นแอปพลิเคชันที่อำนวยความสะดวกในการติดต่อสื่อสารกันระหว่างผู้ใช้งาน (User) รูปแบบของแอปพลิเคชันมีการออกแบบเพื่อให้ใช้งานได้ง่าย เพียงแค่สมัครกรอกข้อมูลให้ครบถ้วนนำข้อมูลที่สมัครไปทำการล็อกอิน (Login) ผู้ใช้งานใส่อีเมล (Email) และพาสเวิร์ด (Password) เพื่อเข้าสู่หน้าผู้ใช้งาน จากนั้นจึงจะสามารถกรอกข้อมูลต่างๆที่มีในระบบ และมีหน้าตาแอปพลิเคชันที่เหมาะสมกับคนยุคนี้ นอกจากนี้ผู้งานแอปพลิเคชันสามารถดูตำแหน่งของสัตว์เลี้ยงที่มีอุปกรณ์ติดตาม (Tracking) เพราะมีการนำเทคโนโลยี GPS ในการบอกตำแหน่งของสัตว์เลี้ยงเช่น ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time) เก็บค่าลงในฐานข้อมูล (Firebase) ทำให้ผู้ใช้ (User) แอปพลิเคชันดูตำแหน่งของสัตว์เลี้ยงเพียงแค่เปิดดูแผนที่บนแอปพลิเคชันก็สามารถดูตำแหน่งของสัตว์เลี้ยงได้เลยและยังมีการแชร์ภาพกันระหว่างผู้ใช้แอปพลิเคชันทำให้ผู้ใช้งานสามารถแชร์ภาพสัตว์ของตนได้ และมีการแชร์ภาพของสัตว์เลี้ยงที่พบระหว่างผู้ใช้แอปพลิเคชันอีกด้วย

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี เนื่องจากได้รับความช่วยเหลือเป็นอย่างดีจากอาจารย์หลายท่านโดยเฉพาะความกรุณาจากอาจารย์ที่ปรึกษา ดร. เอกรัฐ รัชฎาภรณ์ ที่กรุณาสละเวลาให้คำปรึกษาและแนะนำต่างๆ ที่ช่วยให้ทางผู้จัดทำเห็นปัญหาและแนวทางใหม่ๆ ในการแก้ไขปัญหาของการทำโครงการในครั้งนี้ ทางผู้จัดทำรู้สึกเป็นเกียรติและซาบซึ้งในความกรุณาของท่านเป็นอย่างยิ่งจึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้ด้วย

สุดท้ายนี้ทางผู้จัดทำขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องในการทำโครงการแอปพลิเคชันติดตามสัตว์เลี้ยงและขอขอบคุณสมาชิกในกลุ่มที่ร่วมจัดทำโครงการในครั้งนี้ ทำให้สามารถทำโครงการนี้สำเร็จลุล่วงลงด้วยดี

นาย ชัยณรงค์ เดชสุภา

นางสาว สมฤดี พุ่มพวง

29 มิถุนายน 2561

สารบัญ

หน้า

| | |
|--|-----|
| บทคัดย่อ..... | I |
| กิตติกรรมประกาศ..... | II |
| สารบัญ..... | III |
| สารบัญรูป..... | V |
| สารบัญตาราง..... | X |
| บทที่ 1 บทนำ..... | 1 |
| 1.1ความเป็นมาของโครงการ..... | 1 |
| 1.2 วัตถุประสงค์ของโครงการ..... | 1 |
| 1.3 ขอบเขตของโครงการ..... | 1 |
| 1.4 ตารางแผนการดำเนินงาน..... | 3 |
| บทที่ 2 การวิเคราะห์และการออกแบบ..... | 5 |
| 2.1 พื้นฐานและทฤษฎีการใช้งาน..... | 5 |
| 2.2 ยูสเคสไดอะแกรม (Use Case Diagram)..... | 6 |
| 2.3 แผนภาพกิจกรรม (Activity Diagram)..... | 19 |
| 2.4 คลาสไดอะแกรม (Class Diagram)..... | 35 |
| 2.5 แผนภาพกิจกรรม (Sequence Diagram)..... | 41 |
| บทที่ 3 การออกแบบและการทดสอบระบบ..... | 50 |
| 3.1 หน้าจอหลักบนแอปพลิเคชัน..... | 50 |
| 3.2 หน้าจอ Login เข้าสู่ระบบ..... | 51 |
| 3.3 หน้าจอ Register สมัครสมาชิก..... | 52 |
| 3.4 หน้าจอหลักของแอปพลิเคชัน..... | 53 |
| 3.5 หน้าจอให้เลือกแชร์ภาพสัตว์เลี้ยงหายหรือพบเจอ..... | 54 |
| 3.6 หน้าจอการแชร์ภาพสัตว์เลี้ยงไม่พบบริเวณบ้านและหายออกจากบ้านเป็นเวลานาน..... | 55 |
| 3.7 หน้าจอการแชร์ภาพสัตว์เลี้ยงพบ..... | 56 |
| 3.8 หน้าจอการแก้ไขข้อมูล..... | 57 |
| 3.9 แสดงตำแหน่งของสัตว์เลี้ยง..... | 58 |
| 3.10 หน้าของผู้ดูแลระบบ..... | 59 |
| 3.11 หน้าจอการแสดงการแก้ไขข้อมูลของ Admin..... | 60 |
| 3.12 หน้า Create User and Admin..... | 61 |

สารบัญ (ต่อ)

| | หน้า |
|---|------|
| 3.13 ผลการทดลอง..... | 62 |
| 3.14 ผลลัพธ์การอ่านค่า GPS..... | 65 |
| บทที่ 4 สรุปและวิเคราะห์ผลการดำเนินงาน..... | 66 |
| 4.1 สรุปผลการดำเนินงานสรุปผลการดำเนินงาน..... | 66 |
| 4.2 วิเคราะห์ผลการดำเนินงาน..... | 69 |
| 4.3 แนวทางการพัฒนาในอนาคต..... | 74 |
| เอกสารอ้างอิง..... | 75 |

สารบัญรูป

หน้า

| | |
|--|----|
| รูปที่ 2.1 Use Case Diagram of tracker pet application on mobile..... | 6 |
| รูปที่ 2.2 Use Case Diagram of tracker pet application on Arduino..... | 14 |
| รูปที่ 2.3 Use Case Diagram of tracker pet application server..... | 16 |
| รูปที่ 2.4 Activity Diagram for register on Application mobile..... | 19 |
| รูปที่ 2.5 Activity Diagram for login on Application mobile..... | 21 |
| รูปที่ 2.6 Activity Diagram for Manage..... | 23 |
| รูปที่ 2.7 Activity Diagram for edit profile..... | 25 |
| รูปที่ 2.8 Activity Diagram for View position..... | 27 |
| รูปที่ 2.9 Activity Diagram for Share photo..... | 29 |
| รูปที่ 2.10 Activity Diagram Get position..... | 31 |
| รูปที่ 2.11 Activity Diagram Feed position..... | 33 |
| รูปที่ 2.12 Class Diagram pet track application on android mobile..... | 35 |
| รูปที่ 2.13 Class Diagram of Login..... | 36 |
| รูปที่ 2.14 Class Diagram of Register..... | 36 |
| รูปที่ 2.15 Class Diagram of Manage..... | 37 |
| รูปที่ 2.16 Class Diagram of Pet..... | 37 |
| รูปที่ 2.17 Class Diagram of Map..... | 38 |
| รูปที่ 2.18 Class Diagram of Google Map..... | 39 |
| รูปที่ 2.19 Class Diagram of User..... | 39 |
| รูปที่ 2.20 Sequence Diagram for Register | 41 |
| รูปที่ 2.21 Sequence Diagram for login..... | 43 |
| รูปที่ 2.22 Sequence Diagram for Manage..... | 44 |
| รูปที่ 2.23 Sequence Diagram for Edit profile..... | 46 |
| รูปที่ 2.24 Sequence Diagram for View position..... | 48 |
| รูปที่ 2.25 Sequence Diagram for Share photo..... | 49 |
| รูปที่ 3.1 หน้าจอแอปพลิเคชัน..... | 50 |
| รูปที่ 3.2 หน้าจอ Login เข้าสู่ระบบ..... | 51 |
| รูปที่ 3.3 หน้าจอสมัครสมาชิก(Register) | 52 |
| รูปที่ 3.4 หน้าจอหลักของแอปพลิเคชัน..... | 53 |

สารบัญรูป (ต่อ)

หน้า

| | |
|--|----|
| รูปที่ 3.5 หน้าจอการแชร์ภาพสัตว์เลี้ยงที่ไม่พบหรือหายออกไปเป็นเวลานาน และการแชร์ภาพสัตว์เลี้ยงพบจากผู้ใช้งานแอปพลิเคชันร่วมกัน..... | 54 |
| รูปที่ 3.6 หน้าจอการแชร์ภาพสัตว์เลี้ยงไม่พบบริเวณบ้านและหายออกจากบ้านเป็นเวลานาน..... | 55 |
| รูปที่ 3.7 หน้าจอการกรอกข้อมูลสัตว์เลี้ยงเพื่อทำแชร์ภาพสัตว์เลี้ยงที่พบ..... | 56 |
| รูปที่ 3.8 หน้าจอการแสดงการแก้ไขข้อมูล..... | 57 |
| รูปที่ 3.9 หน้าจอการแสดงตำแหน่งของสัตว์เลี้ยง..... | 58 |
| รูปที่ 3.10 หน้าของผู้ดูแลระบบ..... | 59 |
| รูปที่ 3.11 หน้าจอการแสดงการแก้ไขข้อมูลของ Admin..... | 60 |
| รูปที่ 3.12 หน้า Create User and Admin..... | 61 |
| รูปที่ 3.13 Source code read GPS..... | 62 |
| รูปที่ 3.14 ผลลัพธ์การอ่านค่า GPS..... | 65 |
| รูปที่ 3.15 ผลลัพธ์การอ่านค่าเมื่ออยู่ในที่ที่พบ..... | 65 |
| รูปที่ 4.1 หน้าแอปพลิเคชันของผู้ใช้งาน(User)..... | 69 |
| รูปที่ 4.2 หน้าจอหลักของแอปพลิเคชัน..... | 70 |
| รูปที่ 4.3 หน้าจอการเรียกดูข้อมูลการแชร์ภาพสัตว์เลี้ยงที่ไม่พบใน GPS และสัตว์เลี้ยงที่พบจากผู้ใช้งานแอปพลิเคชันร่วมกัน..... | 71 |
| รูปที่ 4.4 หน้าจอการแสดงการแชร์ภาพสัตว์เลี้ยงที่ไม่พบใน GPS จากผู้ใช้และผู้ใช้งานแอปพลิเคชันร่วมกัน..... | 72 |
| รูปที่ 4.5 หน้าจอการแสดงการแชร์ภาพสัตว์เลี้ยงที่พบจากผู้ใช้และผู้ใช้งานแอปพลิเคชัน ร่วมกัน..... | 73 |

สารบัญตาราง

| | หน้า |
|--|------|
| ตารางที่ 1.1 แผนการดำเนินงานโครงการ 1..... | 3 |
| ตารางที่ 1.2 แผนการดำเนินงานโครงการ 2..... | 4 |
| ตารางที่ 4.1 การใช้งาน Android Emulator..... | 67 |
| ตารางที่ 4.2 การทดสอบการทำงานของเซ็นเซอร์ GPS Module และ Node MCU Wi-Fi โดยทำการทดสอบการรับค่าแต่ละสถานที่..... | 68 |

บทที่1

บทนำ

1.1 ความเป็นมาของโครงการ

ในปัจจุบันทุกวันนี้เมื่อกล่าวถึงปัญหาสัตว์เลื้อยออกนอกพื้นที่บริเวณบ้านหรือไปยังสถานที่ต่างๆ จนทำให้เจ้าของออกตามหาโดยการตะโกนเรียกชื่อของสัตว์เลื้อย หรือถ้าออกนอกพื้นที่เป็นเวลานานส่วนใหญ่ก็จะพบวิธีการแก้ปัญหาโดยการใช้แผ่นกระดาษ A4 ปรี้นข้อความหรือเขียนข้อความตามหาสัตว์เลื้อยไปตามสถานที่ต่างๆ ที่เราเคยพบเห็นซึ่งก็อาจจะไม่แน่ว่ามีคนได้สังเกตป้ายดังกล่าวมากหรือน้อยเพียงใด หรือทำการโพสต์ตามโลก Social Network เพื่อขอความช่วยเหลือและรอการติดต่อกลับจากผู้พบเห็น

ดังนั้นทางคณะผู้จัดทำได้มองเห็นว่าสมาร์ตโฟน (Smartphone) ต่างหากที่เป็นสิ่งที่มีมนุษย์แทบจะพกติดตัวกันตลอดเวลา จึงได้คิดพัฒนาแอปพลิเคชันในลักษณะแอปพลิเคชันบนโทรศัพท์มือถือที่ทำงานร่วมกับบริการแผนที่ของ Google ในชื่อแอปพลิเคชันติดตามสัตว์เลื้อย โดยสามารถระบุตำแหน่งของสัตว์เลื้อยที่มีอุปกรณ์ (Tracking) ติดอยู่ได้เลย เพียงแค่เปิดแอปพลิเคชันและอุปกรณ์จะต้องมีการเชื่อมต่อกับอินเทอร์เน็ตอุปกรณ์ถึงจะส่งค่า ละติจูด (Latitude) ลองจิจูด (Longitude) เพื่อให้ผู้ใช้งาน (User) สามารถดูตำแหน่งของสัตว์เลื้อยบนสมาร์ตโฟน (Smartphone) ได้อย่างถูกต้อง

1.2 วัตถุประสงค์ของโครงการ

- 2.1 เพื่อลดปัญหาสัตว์เลื้อยสูญหายที่เกิดขึ้นบ่อยครั้งในชุมชน
- 2.2 เพื่อให้เจ้าของสัตว์เลื้อยรู้ตำแหน่งของสัตว์เลื้อยของตนเองได้ และช่วยให้สามารถตามหาสัตว์เลื้อยของตนเองได้ง่ายขึ้น
- 2.3 เพื่อเพิ่มวิธีการตามหาสัตว์เลื้อย และสะดวกกว่าการนำเอารูปภาพสัตว์เลื้อยไปติดไว้ตามสถานที่ต่าง ๆ

1.3 ขอบเขตของโครงการ

ขอบเขตโครงการ 1

- 3.1 Mobile Application สามารถทำงานได้บนระบบปฏิบัติการ Android เท่านั้น
- 3.2 ออกแบบระบบทั้งหมดให้รองรับโครงการ 2
- 3.3 ระบบสามารถอ่านค่าตำแหน่ง GPS Reader

ขอบเขตโครงการ 2

3.4 Mobile Application สามารถรองรับการ Login เข้าสู่ระบบได้โดยใช้

3.4.1 Username

3.4.2 Password

3.5 ระบบสามารถสมัครสมาชิกได้ โดยใช้ข้อมูลดังนี้

3.5.1 ชื่อ นามสกุล Username , Password อีเมล และ เบอร์โทรศัพท์

3.6 ระบบสามารถทำการแก้ไขข้อมูลสมาชิกได้ ผู้ใช้จะต้องทำการเข้าสู่ระบบแล้วเท่านั้น

3.7 ระบบสามารถจัดการข้อมูลพื้นฐานต่างๆ ได้โดยผู้ดูแลระบบ ดังนี้

3.7.1 สามารถแก้ไข เพิ่ม ลบ ข้อมูลของผู้ใช้ได้

3.7.2 สามารถแก้ไข เพิ่ม ลบ ข้อมูลของ Admin ได้

3.7.3 สามารถแก้ไข สิทธิ์ของผู้ใช้งาน Application ได้

3.7.4 ผู้ดูแลระบบสามารถจัดรายการข้อมูล และประเภทของสัตว์เลี้ยงได้

1) สุนัข

2) แมว

3.8 ระบบสามารถแชร์รูปสัตว์เลี้ยงได้ระหว่างผู้ใช้แอปพลิเคชันโดยต้องเลือกประเภทที่จะทำการแชร์ภาพ ประเภทมีดังนี้

3.8.1 สัตว์เลี้ยงสูญหาย

1) กรณีไม่พบสัตว์เลี้ยงในตำแหน่งตาม GPS

3.8.2 สัตว์เลี้ยงที่พบ

3.9 ข้อจำกัดของอุปกรณ์

3.9.1 อุปกรณ์ต้องทำการติดไว้กับตัวสัตว์เลี้ยงตลอดเวลาเมื่อต้องการติดตาม

3.9.2 เมื่ออุปกรณ์สูญหาย หรือถูกนำออกจากตัวของสัตว์เลี้ยงจะไม่สามารถทำการติดตามได้

1.4 ตารางแผนการดำเนินงานโครงการ 1

ตารางที่ 1.1 แผนการดำเนินงานโครงการ 1

| ลำดับ | แผนงานแต่ละสัปดาห์ | มกราคม | | | | กุมภาพันธ์ | | | | มีนาคม | | | | เมษายน | | | |
|-------|--|--------|---|---|---|------------|---|---|---|--------|---|---|---|--------|---|---|---|
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | เสนอหัวข้อโครงการและขอบเขตโครงการ | | | | | | | | | | | | | | | | |
| 2 | ศึกษาทฤษฎีหลักการและเทคโนโลยีที่เกี่ยวข้อง | | | | | | | | | | | | | | | | |
| 2.1 | ศึกษาการทำงานของโครงสร้าง GPSModule | | | | | | | | | | | | | | | | |
| 2.2 | ศึกษาการทำงานของโครงสร้าง Node MCU Wi-Fi | | | | | | | | | | | | | | | | |
| 2.3 | โปรแกรมย่อยที่ทดสอบ | | | | | | | | | | | | | | | | |
| 3 | ออกแบบ UML Diagram | | | | | | | | | | | | | | | | |
| 4 | ออกแบบหน้าจอ | | | | | | | | | | | | | | | | |
| 5 | ออกแบบภาพรวมของระบบ | | | | | | | | | | | | | | | | |
| 6 | ทดสอบการ Predict ข้อมูล | | | | | | | | | | | | | | | | |
| 7 | จัดทำเอกสารขึ้นสอบโครงการ | | | | | | | | | | | | | | | | |
| 8 | ขึ้นสอบโครงการ 1 | | | | | | | | | | | | | | | | |

1.5 แผนการดำเนินงานโครงการ 2

ตารางที่ 1.2 แผนการดำเนินงานโครงการ 2

| ลำดับ | แผนงานแต่ละสัปดาห์ | พฤษภาคม | | | | มิถุนายน | | | | กรกฎาคม | | | |
|-------|--------------------------------------|---------|---|---|---|----------|---|---|---|---------|---|---|---|
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | พัฒนาแอปพลิเคชัน | | | | | | | | | | | | |
| 2 | แก้ไขเอกสาร | | | | | | | | | | | | |
| 3 | พัฒนาแอปพลิเคชัน | | | | | | | | | | | | |
| 3.1 | จัดทำเอกสารและแก้ไขเอกสาร | | | | | | | | | | | | |
| 4 | แก้ไข ทดสอบ ใช้งานแอปพลิเคชัน | | | | | | | | | | | | |
| 4.1 | สภาพแวดล้อมที่ทดสอบตัวอุปกรณ์ | | | | | | | | | | | | |
| 5 | จัดทำเอกสารขึ้นสอบโครงการ 2 | | | | | | | | | | | | |
| 6 | สอบโครงการ 2 | | | | | | | | | | | | |
| 7 | ทำแอปพลิเคชันให้ตรงกับขอบเขตที่กำหนด | | | | | | | | | | | | |
| 8 | ส่งแอปพลิเคชันที่ทำการแก้ไข | | | | | | | | | | | | |
| 9 | ส่งเล่มดำ | | | | | | | | | | | | |

บทที่ 2

การวิเคราะห์และการออกแบบระบบ

2.1 พื้นฐานและทฤษฎีการใช้งาน

2.1.1 Unified Modeling Language (UML)

UML คือภาษาที่ใช้อธิบายแบบจำลองของระบบที่เราทำการสร้างหรือพัฒนาระบบนั้นๆและยังนำมาบ่งบอกถึง ลักษณะของระบบ รายละเอียดของระบบ การสร้างและออกแบบระบบUML เป็นภาษาสัญลักษณ์รูปภาพมาตรฐานใช้สำหรับในการสร้างแบบจำลองเชิงวัตถุเพื่อให้มองเห็นภาพได้ง่าย อีกทั้งUML เป็นภาษามาตรฐานที่คนส่วนมากให้การยอมรับและให้ความสำคัญเป็นอย่างยิ่งในการสร้างหรือออกแบบระบบใหม่ๆ ทั้งนี้ UML สามารถสร้างแบบพิมพ์เขียวให้แก่ระบบงานของเราได้อีกด้วย เราสามารถใช้UML สร้างมุมมองสำหรับรายละเอียดของระบบและยังสร้างเอกสารอ้างอิงให้กับระบบงานได้อีก เพราะ UML เป็นภาษาที่มีการใช้สัญลักษณ์รูปภาพและยังมีคำอธิบายในสัญลักษณ์รูปภาพนั้นๆ เพื่อให้คนที่ทำงานร่วมกันหรือกำลังพัฒนาระบบร่วมกันเกิดความเข้าใจในงานที่ทำร่วมกันอย่างชัดเจน และในบางครั้งมีคนส่วนใหญ่มักเข้าใจผิดหรือสับสนว่า UML นั้นคือการสร้างแผนภาพหรือเป็นเพียงการใช้สัญลักษณ์เพื่ออธิบายระบบงานเท่านั้น อันที่จริง UML คือสัญลักษณ์เพื่ออธิบายลักษณะของแบบจำลองข้อมูลในระบบที่เรากำลังจะสร้างหรือกำลังจะพัฒนาและยังเป็นแบบจำลองที่เอาไว้อธิบายแบบจำลองอื่นๆเพื่อให้เกิดความเข้าใจในกระบวนการทำงาน นั้นต้องมีความรู้และความเข้าใจในแนวความคิดเชิงวัตถุและระบบของตัวเองก่อน และยังต้องทำความเข้าใจในรายละเอียดที่เกี่ยวข้องอีกด้วย

ยูสเคสไดอะแกรม (Use Case Diagram) เป็นแผนภาพที่ใช้ในการแสดงความสัมพันธ์ระหว่างผู้ใช้งานระบบคือ (Actors) และหน้าที่การทำงาน (Use cases) ภายในระบบ รวมถึงการแสดงให้เห็นถึงบทบาทของผู้ใช้งานระบบแต่ละคน และยังใช้ในการสื่อสารระหว่างผู้พัฒนาระบบและผู้ใช้งานระบบอีกด้วย องค์ประกอบของ Use case diagram ประกอบไปด้วย Actor , Use Case , Relationship

คลาสไดอะแกรม (Class Diagram) ประกอบไปด้วย Class , Object , Relationships

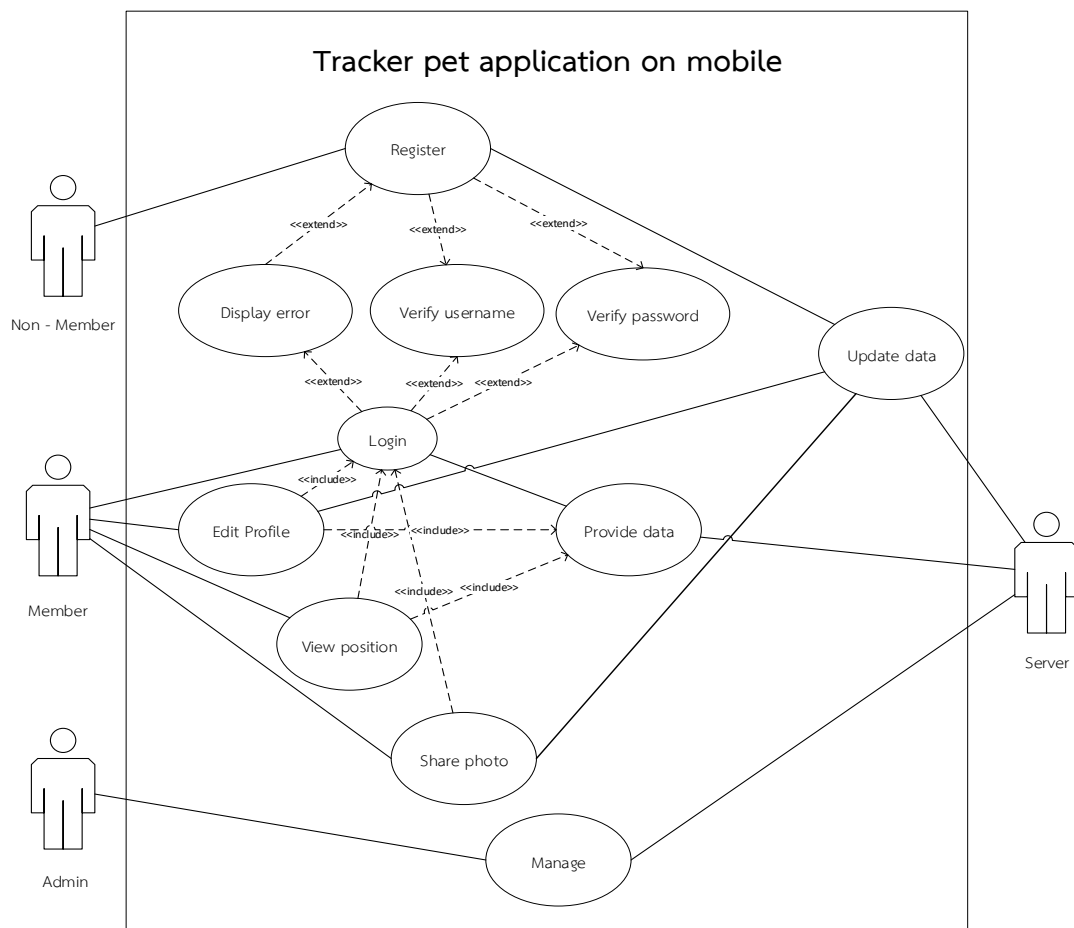
- 1) Class คือ วัตถุที่มีคุณสมบัติ (Attributes) และการกระทำ (Operation) ร่วมกัน
- 2) Object คือ ผลของการสืบทอด Class เป็นกลุ่มของคุณสมบัติที่บอกขอบเขตของวัตถุ โดยมีสถานะ (State) และพฤติกรรม (Behavior)
- 3) ความสัมพันธ์ระหว่างคลาส (Relationships) เช่น Dependency, Generalization, Association

ซีควเอนไดอะแกรม (Sequence Diagram) Sequence Diagram เป็นแผนภาพที่แสดงลำดับการกระทำระหว่างวัตถุ ณ เวลาต่าง ๆ แบบไดนามิก (Dynamic) โดยจะมีแสดงให้เห็นถึงกาติดต่อกันระหว่างวัตถุต่างๆว่ามีกระบวนการอย่างไร

2.2 ยูสเคสไดอะแกรม (Use Case Diagram)

เป็นแผนภาพที่ใช้แสดงว่าระบบมีการทำงานหรือมีหน้าที่อะไรบ้างเพื่อให้ได้งานตามความต้องการ โดยจะแสดงการทำงานของระบบที่ทำหน้าที่ให้ผลลัพธ์หรืองานตามที่คุณต้องการ

การออกแบบ Use Case ของแอปพลิเคชันติดตามสัตว์เลี้ยง (Pet track Application)



รูปที่ 2.1 Use Case Diagram of tracker pet application on mobile

คำอธิบาย Use Case Diagram ของ tracker pet application on mobile

Use Case ที่ 1 Register

1. Brief Description

ทำหน้าที่ในการสมัครสมาชิกของผู้ใช้งาน Application ประกอบด้วย ชื่อผู้ใช้งาน (Name) , email ,รหัสเข้าใช้งาน (Password), Confirm Password เบอร์โทรศัพท์ (Telephone)

2. Preconditions

- 2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ
3. Actor
 - 3.1 Non – Member คือผู้ใช้งาน (User) ที่กำลังทำการสมัครเข้าสู่ระบบ
 - 3.2 Server คือฐานข้อมูล (firebase)
4. Basic Flow of Event
 - 4.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าสมัครสมาชิก
 - 4.2 ผู้ใช้งาน (User) ต้องทำการกรอกข้อมูลพื้นฐานที่มีในระบบ
5. Alternative Flow
 - 5.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่อกับอินเทอร์เน็ตระบบจะไม่สามารถทำการสมัครสมาชิกได้ ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง
 - 5.2 หากทำการกรอกข้อมูลพื้นฐานไม่ครบถ้วนระบบจะแสดงข้อความแจ้งเตือนให้ผู้ใช้งาน (User) ทราบและทำการกรอกข้อมูลให้ครบถ้วนอีกครั้ง
6. Post conditions
 - 6.1 ระบบได้ทำการตรวจสอบสามารถทำการสมัครสมาชิกได้สำเร็จ

Use Case ที่ 2 Verify username

1. Brief Description

ทำหน้าที่ในการรับข้อมูลและ ชื่อผู้ใช้งาน (username) เพื่อตรวจสอบชื่อผู้ใช้งาน (User) ในการสมัครสมาชิกและเข้าสู่ระบบขอผู้ใช้งาน (User)
2. Preconditions
 - 2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ
3. Basic Flow of Event
 - 3.1 ผู้ใช้งาน (User) เข้าสู่หน้าสมัคร
 - 3.2 ผู้ใช้งาน (User) ต้องทำการกรอกข้อมูลให้ครบถ้วน
4. Alternative Flow
 - 4.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่อกับอินเทอร์เน็ตระบบจะไม่สามารถทำการสมัครสมาชิกได้ ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง
 - 4.2 หากทำการกรอกข้อมูลพื้นฐานไม่ครบถ้วนระบบจะแสดงข้อความแจ้งเตือนให้ผู้ใช้งาน (User) ทราบและทำการกรอกข้อมูลให้ครบถ้วนใหม่อีกครั้ง
5. Post conditions
 - 5.1 ระบบทำการตรวจสอบชื่อที่ผู้ใช้งาน (User) สามารถใช้งานได้ในระบบ

Use Case ที่ 3 Verify password

1. Brief Description

ทำหน้าที่ในการรับข้อมูลและตรวจสอบรหัสผ่าน (Password) ในการสมัครสมาชิก และเข้าสู่ระบบของผู้ใช้งาน (User)

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Basic Flow of Event

3.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าสมัครสมาชิก

3.2 ผู้ใช้งาน (User) ต้องทำการกรอกข้อมูลพื้นฐานให้ครบถ้วน

4. Alternative Flow

4.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่ออินเทอร์เน็ตระบบจะไม่สามารถทำการสมัครสมาชิกได้ ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) เชื่อมต่อใหม่อีกครั้ง

4.2 หากทำการกรอกข้อมูลพื้นฐานไม่ครบถ้วนระบบจะแสดงข้อความแจ้งเตือนให้ผู้ใช้งานทราบและทำการกรอกข้อมูลให้ครบถ้วน

5. Post conditions

5.1 ระบบทำการตรวจสอบรหัสผ่านถูกต้องสามารถใช้งานได้

Use Case ที่ 4 Display error

1. Brief Description

ทำหน้าที่ในการแจ้งเตือนข้อผิดพลาดในการกรอกข้อมูล มีการกรอกข้อมูลที่ไม่ถูกต้องลงในระบบ หรือกรอกข้อมูลไม่ครบตามที่ระบบต้องการ

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Basic Flow of Event

3.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าสมัครสมาชิก

3.2 ผู้ใช้งาน (User) ต้องทำการกรอกข้อมูลพื้นฐานให้ครบถ้วนประกอบไปด้วย ชื่อผู้ใช้งาน (Username), รหัสเข้าใช้งาน (Password), ชื่อและนามสกุล, email, เบอร์โทรศัพท์

4. Alternative Flow

4.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่ออินเทอร์เน็ตระบบจะไม่สามารถทำการสมัครสมาชิกได้ ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) เชื่อมต่อใหม่อีกครั้ง

4.2 หากทำการกรอกข้อมูลพื้นฐานไม่ครบถ้วนระบบจะแสดงข้อความแจ้งเตือนให้
ผู้ใช้งานทราบและทำการกรอกข้อมูลให้ครบถ้วน

5. Post conditions

5.1 ระบบทำการตรวจสอบการสมัครสมาชิกผิดพลาดมาสามารถสมัครสมาชิกได้

Use Case ที่ 5 Login

1. Brief Description

ทำหน้าที่ในการเข้าสู่ระบบเพื่อที่จะได้รับสิทธิ์ในการใช้งานแอปพลิเคชันในส่วน
ต่างๆ ประกอบไปด้วยอีเมลผู้ใช้งาน (Email) , รหัสเข้าใช้งาน(Password)

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตตลอดเวลา

3. Actor

3.1 Memberคือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลระบบ

3.2 Serverคือฐานข้อมูล (firebase)

4. Basic Flow of Event

4.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าล็อกอิน

4.2 ผู้ใช้งาน(User) ต้องทำการกรอกข้อมูลรหัสผ่านเพื่อเข้าสู่ระบบ โดยประกอบไป
ด้วยอีเมลผู้ใช้งาน (Email) , รหัสเข้าใช้งาน(Password)

5. Alternative Flow

5.1 เมื่อระบบไม่สามารถทำการเชื่อมต่ออินเทอร์เน็ตได้จะไม่สามารถทำการเข้าสู่
ระบบได้และให้ทำการเชื่อมต่อใหม่อีกครั้ง

5.2 หากทำการกรอกข้อมูลรหัสผ่านไม่ถูกต้องจะมีการแจ้งเตือนให้ผู้ใช้งานทราบ
แล้วทำการกรอกใหม่

6. Post conditions

6.1 ระบบทำการตรวจสอบสามารถล็อกอิน(Login) เข้าสู่ระบบแบบสมบูรณ์

Use Case ที่ 6 Edit Profile

1. Brief Description

สามารถทำการในการแก้ไขข้อมูลพื้นฐานของผู้ใช้งาน(User) ที่มีอยู่ในระบบแอปพลิเคชัน

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

3.1 Memberคือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลระบบ

3.2 Serverคือฐานข้อมูล (firebase)

4. Basic Flow of Event

4.1 ต้องทำการเข้าสู่ระบบ(Login) แอปพลิเคชันติดตามสัตว์เลี้ยงก่อน

4.2 เมื่อผู้ใช้งาน(User) ทำการเข้าสู่(Login) เรียบร้อย จึงจะสามารถจัดการแก้ไขข้อมูลของผู้ใช้ (User) ที่มีอยู่ในระบบได้

5. Alternative Flow

5.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่ออินเทอร์เน็ตระบบจะไม่สามารถทำการแก้ไขสมาชิกได้ ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง

5.2 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ใช้งานได้ทราบและทำการเข้าสู่ระบบใหม่อีกครั้ง

6. Post conditions

6.1 ระบบได้ทำการตรวจสอบการแก้ไขข้อมูลของผู้ใช้งาน (User) แก้ไขข้อมูลเรียบร้อยแล้วแสดงข้อมูลที่แก้ไข

Use Case ที่ 7 Share Photo

1. Brief Description

เป็นการแชร์รูปภาพของสัตว์เลี้ยงและใส่ข้อมูลของผู้ใช้งาน (User) บางส่วนลงไปใน
ในการแชร์รูปภาพด้วย

2. Actor

2.1 Memberคือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลระบบ

2.2 Serverคือฐานข้อมูล (firebase)

3. Preconditions

3.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

4. Basic Flow of Event

4.1 ต้องทำการเข้าสู่ระบบ (Login) แอปพลิเคชันติดตามสัตว์เลี้ยงก่อน

4.2 ผู้ใช้งาน (User) ทำการเข้าสู่หน้าแชร์รูปภาพ

5. Alternative Flow

5.1 เมื่อระบบไม่สามารถทำการเชื่อมต่ออินเทอร์เน็ตได้จะไม่สามารถทำการแชร์รูปภาพได้และให้ทำการเชื่อมต่อใหม่อีกครั้ง

5.1 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ใช้งานได้ทราบและทำการเข้าสู่ระบบใหม่

6. Post conditions

6.1 ระบบทำการตรวจสอบสามารถทำการแชร์รูปภาพเสร็จสิ้นแสดงรูปภาพที่แชร์

Use Case ที่ 8 View position

1. Brief Description

ทำหน้าที่ในการแสดงตำแหน่งของสัตว์เลี้ยงจากอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

3.1 Memberคือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลระบบ

3.3 Serverคือฐานข้อมูล (firebase)

4. Basic Flow of Event

4.1 ต้องทำการเข้าสู่ระบบ(Login) แอปพลิเคชันติดตามสัตว์เลี้ยงก่อน

4.2 ผู้ใช้งาน (User) ทำการเข้าสู่หน้าเว็บไซต์ตำแหน่งสัตว์เลี้ยง

5. Alternative Flow

5.1 เมื่อระบบไม่สามารถทำการเชื่อมต่ออินเทอร์เน็ตได้จะไม่สามารถทำการเช็คตำแหน่งของสัตว์เลี้ยงได้และให้ทำการเชื่อมต่อใหม่อีกครั้ง

5.2 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ใช้งานได้ทราบและทำการเข้าสู่ระบบใหม่

6. Post conditions

6.1 ระบบทำการแสดงตำแหน่งของสัตว์เลี้ยง

Use Case ที่ 9 Update data

1. Brief Description

ทำหน้าที่ในการอัปเดตข้อมูลของการสมัครสมาชิก อัปเดตการแก้ไขข้อมูลของผู้ใช้งาน (User) และอัปเดตการร้องขอของผู้ใช้งานในการเรียกดูตำแหน่งของสัตว์เลี้ยง

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

3.1 Memberคือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลระบบ

3.2 Non – Member คือผู้ใช้งาน (User) ที่กำลังทำการสมัครเข้าสู่ระบบ

3.3 Server คือฐานข้อมูล (firebase)

4. Basic Flow of Event

4.1 ผู้ใช้งาน (User) ทำการเข้าสู่หน้าแก้ไขข้อมูลส่วนตัว

4.2 ผู้ใช้งาน (User) ทำการเข้าสู่หน้าสมัครสมาชิก

4.3 ต้องทำการเข้าสู่ระบบ (Login) ก่อน

5. Alternative Flow

5.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่ออินเทอร์เน็ตระบบจะไม่สามารถทำการแก้ไขหรือจัดการข้อมูลได้ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง

5.2 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ใช้งานได้ทราบและให้ทำการเข้าสู่ระบบใหม่

6. Post conditions

6.1 แสดงค่าข้อมูลปัจจุบัน

Use Case ที่ 10 Provide data

1. Brief Description

ทำหน้าที่ในการส่งข้อมูลไปทำการแก้ไข หรือตรวจสอบการใช้สิทธิ์ต่างๆ ของผู้ใช้งาน (User) ตามข้อกำหนดที่มีอยู่ในระบบแอปพลิเคชันติดตามสัตว์เลี้ยง

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

3.1 Member คือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลในระบบได้

3.4 Server คือฐานข้อมูล (firebase)

4. Basic Flow of Event

4.1 ผู้ใช้งาน (User) ทำการเข้าสู่หน้าแก้ไขข้อมูลส่วนตัว

4.2 ต้องทำการเข้าสู่ระบบ (Login) ก่อนจึงจะสามารถ

5. Alternative Flow

5.1 เมื่อผู้ใช้งาน (User) ไม่ทำการเชื่อมต่ออินเทอร์เน็ตระบบจะไม่สามารถทำการแก้ไขหรือจัดการข้อมูลได้ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง

5.2 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ใช้งานได้ทราบและให้ทำการเข้าสู่ระบบใหม่อีกครั้ง

Use Case ที่ 11 Manage

1. Brief Description

ทำหน้าที่ในการตรวจสอบข้อมูลการสมัครสมาชิกของผู้ใช้งาน (User) สามารถทำการแก้ไขข้อมูลพื้นฐานต่างๆ ได้และจัดการข้อมูลประเภทของสัตว์เลี้ยงได้

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

3.1 Member คือผู้ดูแลระบบสามารถเข้ามาจัดการข้อมูลระบบ

3.5 Server คือฐานข้อมูล (firebase)

4. Basic Flow of Event

4.1 ผู้ดูแลระบบทำการเข้าสู่หน้าจัดการข้อมูล

4.2 ต้องทำการเข้าสู่ระบบ (Login) ก่อน

5. Alternative Flow

5.1 เมื่อระบบไม่สามารถทำการเชื่อมต่ออินเทอร์เน็ตได้จะไม่สามารถทำการแก้ไขหรือจัดการข้อมูลได้และให้ทำการเชื่อมต่อใหม่อีกครั้ง

5.2 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ดูแลระบบได้ทราบและให้ทำการเข้าสู่ระบบใหม่อีกครั้ง

6. Post conditions

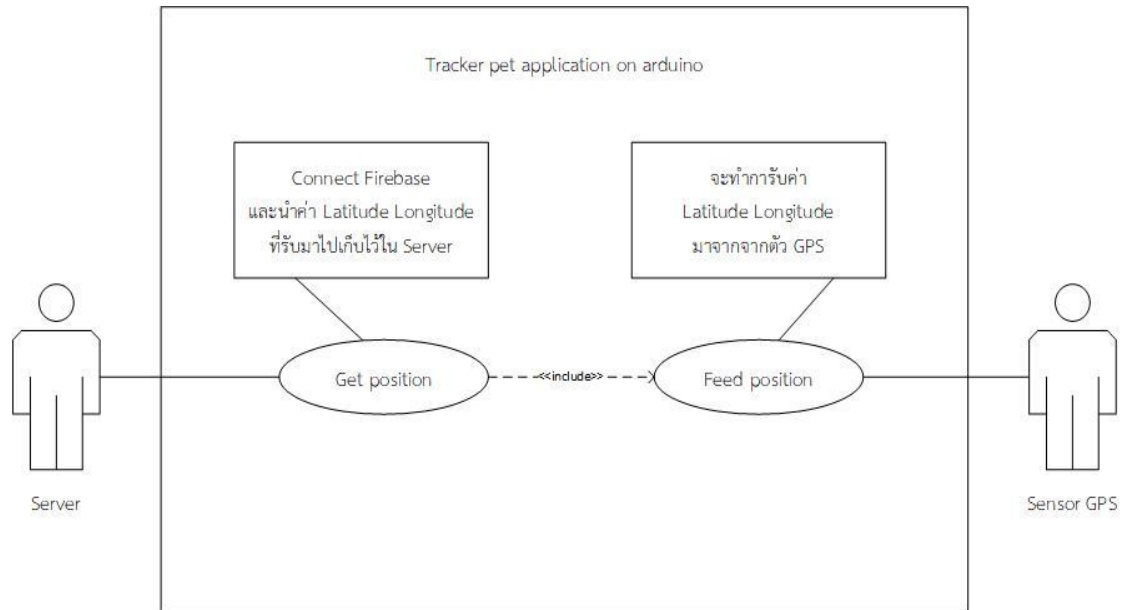
6.1 ระบบได้ทำการตรวจสอบสามารถทำการแก้ไขข้อมูลสมาชิกเสร็จสิ้น

6.2 ระบบได้ทำการตรวจสอบสามารถทำการลบข้อมูลสมาชิกเสร็จสิ้น

6.3 ระบบได้ทำการตรวจสอบสามารถทำการจัดการประเภทสัตว์เลี้ยงเสร็จสิ้น

6.4 แสดงค่าข้อมูลปัจจุบัน

Use Case Diagram Arduino



รูปที่ 2.2 Use Case Diagram of tracker pet application on Arduino

คำอธิบาย Use Case Diagram ของ tracker pet application on Arduino

Use case ที่ 1 Get position

1. Brief Description

ทำหน้าที่ในการดึงข้อมูลจากตำแหน่งที่อยู่ของเซ็นเซอร์ โดยมีการดึงเอาข้อมูล ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time) เก็บค่าลงในฐานข้อมูล (Firebase)

2. Preconditions

- 2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ
- 2.2 อุปกรณ์ต้องเปิดเพื่อตลอดเวลาเพื่อทำหน้าที่ส่งตำแหน่งที่อยู่ของสัตว์เลี้ยง

3. Actor

- 3.1 Server คือฐานข้อมูล (firebase)
- 3.2 Sensor คือตัวอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง

4. Basic Flow of Event

- 4.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าเว็บไซต์ตำแหน่งสัตว์เลี้ยง
- 4.2 ผู้ใช้งานต้องทำการเข้าสู่ระบบ (Login) ก่อน

5. Alternative Flow

5.1 เมื่อระบบไม่สามารถทำการเชื่อมต่ออินเทอร์เน็ตได้จะไม่สามารถทำการดูตำแหน่งสัตว์เลี้ยงได้และให้ทำการเชื่อมต่อใหม่อีกครั้ง

Use case ที่ 2 Feed position

1. Brief Description

ทำหน้าที่ในการบอกตำแหน่งของสัตว์เลี้ยง

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

2.2 อุปกรณ์ต้องเปิดเพื่อตลอดเวลาเพื่อทำหน้าที่ส่งตำแหน่งที่อยู่ของสัตว์เลี้ยง

3. Actor

3.1 Server คือฐานข้อมูล (firebase)

3.2 Sensor คือตัวอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง

4. Basic Flow of Event

4.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าเว็บไซต์ตำแหน่งสัตว์เลี้ยง

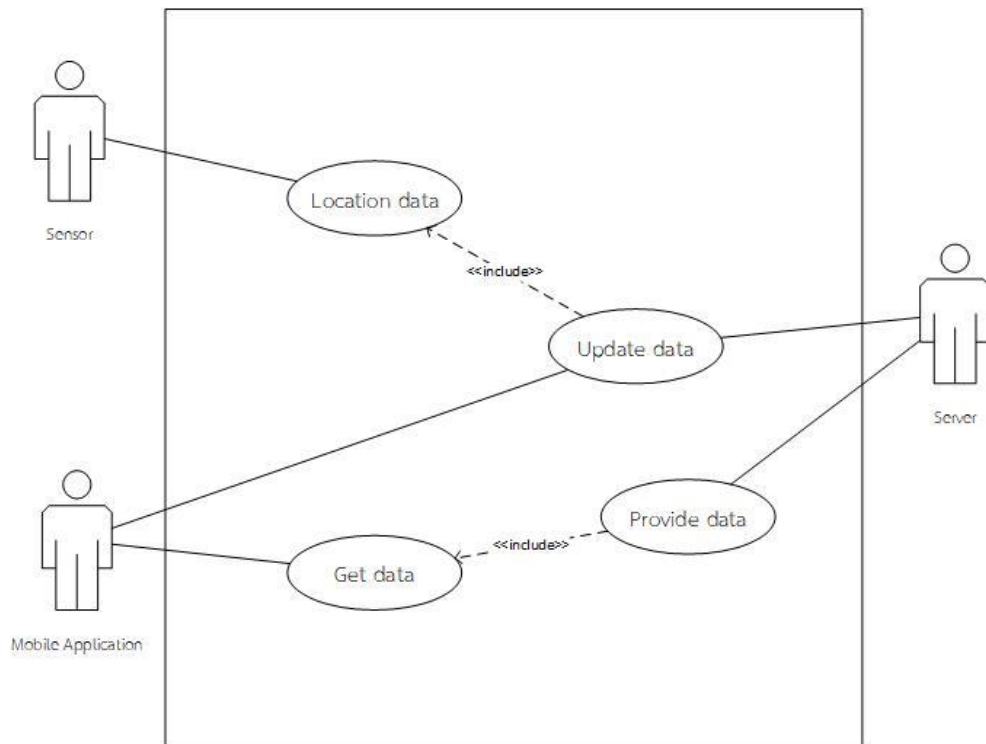
4.2 ผู้ใช้งานต้องทำการเข้าสู่ระบบ (Login) ก่อน

5. Alternative Flow

5.1 เมื่อระบบไม่สามารถทำการเชื่อมต่ออินเทอร์เน็ตได้จะไม่สามารถทำการดูตำแหน่งสัตว์เลี้ยงได้และให้ทำการเชื่อมต่อใหม่อีกครั้ง

5.2 หากทำการเข้าสู่ระบบไม่สำเร็จจะมีข้อความแจ้งเตือนให้ผู้ใช้งานได้ทราบและทำการเข้าสู่ระบบใหม่

Use case Diagram Server



รูปที่ 2.3 Use Case Diagram of tracker pet application server

คำอธิบาย Use Case Diagram ของ tracker pet application server

Use Case ที่ 1 Location data

1. Brief Description

ทำหน้าที่ในการรับข้อมูลตำแหน่งที่อยู่ Latitude Longitude จากตัวอุปกรณ์ และทำการส่งข้อมูลไปให้ Server เพื่อทำการอัปเดต

2. Actor

2.1 Non – Member คือผู้ใช้งาน (User) ที่กำลังทำการสมัครเข้าสู่ระบบ

2.2 Mobile Application คือตัวแอปพลิเคชันที่อยู่บนมือถือ

2.3 Sensor คือตัวอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง

3. Preconditions

3.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3.2 ต้องอยู่ในพื้นที่ที่ไม่ทึบจนเกินไปจึงจะสามารถรับค่าของตำแหน่ง GPS ได้

4. Basic Flow of Event

4.1 เมื่อผู้ใช้งาน (User) เข้าสู่หน้าสมัครสมาชิก

4.2 เมื่อผู้ใช้งาน (User) ทำการติดตั้งตัวอุปกรณ์เรียบร้อยแล้ว

5. Alternative Flow

5.1 เมื่อระบบไม่มีการเชื่อมต่อจะไม่สามารถทำการรับหรือเก็บข้อมูลได้

6. Post conditions

6.1 ระบบทำการอัปเดตข้อมูลเสร็จสมบูรณ์

6.2 ระบบทำการรับค่าข้อมูลตำแหน่งที่อยู่ของสัตว์เลี้ยงเสร็จสมบูรณ์

Use Case ที่ 2 Update data

1. Brief Description

ทำหน้าที่ในการรับและส่งข้อมูลหรืออัปเดตข้อมูลต่างๆ โดยผู้ใช้งานจะต้องทำการเข้าสู่ระบบ (Login) ก่อนในส่วนี้จึงจะสามารถทำงานได้

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

3.1 Non – Member คือผู้ใช้งาน (User) ที่กำลังทำการสมัครเข้าสู่ระบบ

3.2 Mobile Application คือตัวแอปพลิเคชันที่อยู่บนมือถือ

2.3 Sensor คือตัวอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง

4. Basic Flow of Event

4.1 ผู้ใช้งาน (User) เข้าสู่หน้าสมัครสมาชิก

4.2 ผู้ใช้งาน (User) ทำการอัปเดตหรือร้องขอข้อมูลจากฐานข้อมูล (firebase)

5. Alternative Flow

5.1 เมื่อระบบไม่มีการเชื่อมต่อจะไม่สามารถทำการเก็บข้อมูลและร้องขอข้อมูลได้

6. Post conditions

6.1 ระบบทำตรวจสอบและทำการอัปเดตข้อมูลเสร็จสิ้น

Use Case ที่ 3 Get data

1. Brief Description

ทำหน้าที่ในการรับข้อมูลตำแหน่งที่อยู่ Latitude Longitude จาก Server และทำการนำไปแสดงผลในฝั่งการทำงาน Application

2. Preconditions

2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ

3. Actor

- 3.1 Non – Member คือผู้ใช้งาน (User) ที่กำลังทำการสมัครเข้าสู่ระบบ
- 3.2 Mobile Application คือตัวแอปพลิเคชันที่อยู่บนมือถือ
- 3.3 Sensor คือตัวอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง
- 4. Basic Flow of Event
 - 4.1 เมื่อผู้ใช้งาน (User) ต้องการทำเข้าสู่ระบบ
 - 4.2 เมื่อผู้ใช้งาน (User) ทำการติดตั้งตัวอุปกรณ์เรียบร้อยแล้ว
- 5. Alternative Flow
 - 5.1 เมื่อระบบไม่มีการเชื่อมต่อจะไม่สามารถทำการรับและเก็บข้อมูลได้
- 6. Post conditions
 - 6.1 ระบบทำการตรวจสอบรับค่าข้อมูลเรียบร้อยแล้ว

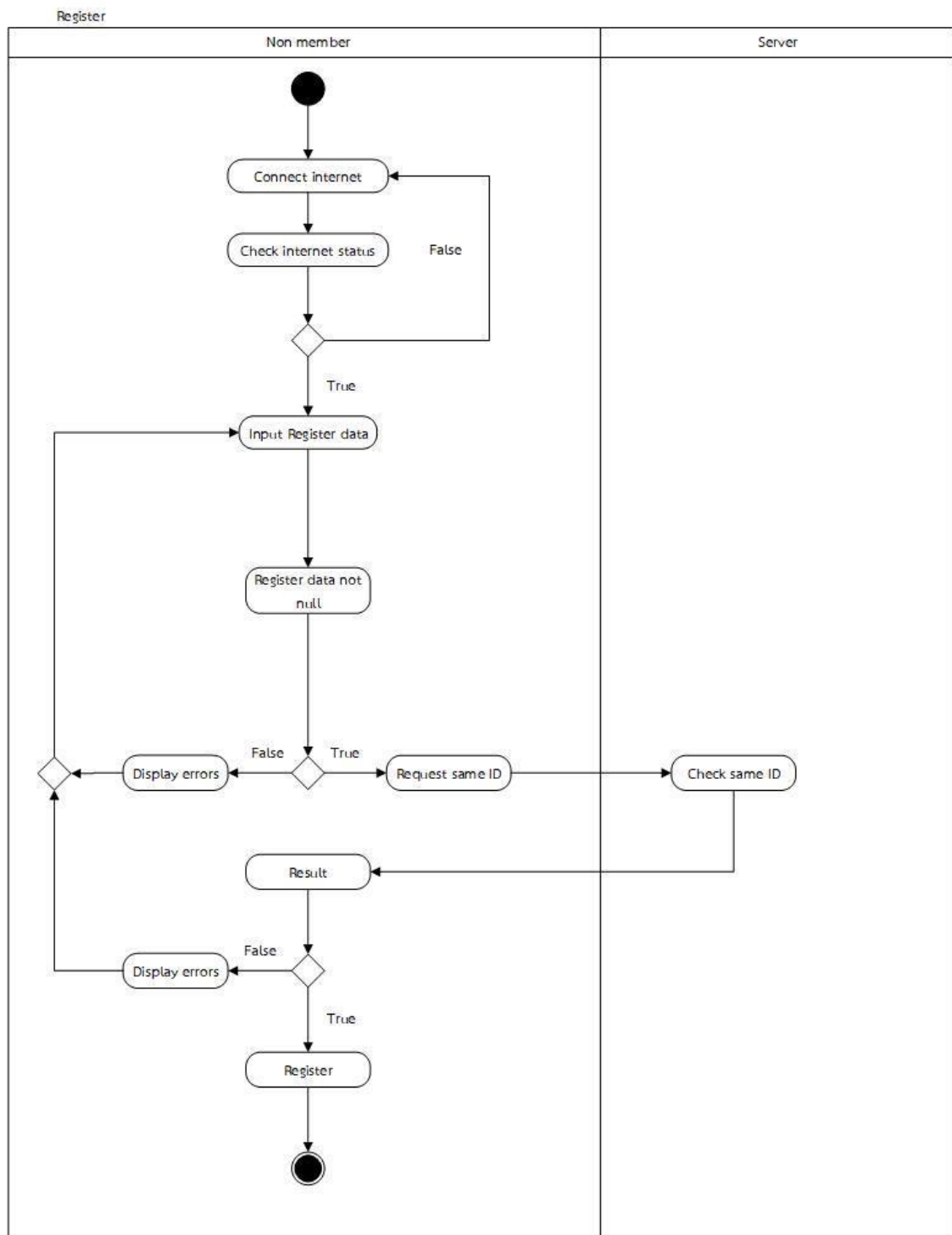
Use Case ที่ 4 Provide data

- 1. Brief Description

ทำหน้าที่ในการส่งข้อมูลค่าของตำแหน่งที่อยู่ใน Server ส่งไปให้ทางฝั่งของ Mobile Application เพื่อทำการแสดงผลตำแหน่งที่อยู่ของสัตว์เลี้ยง
- 2. Preconditions
 - 2.1 ต้องมีการเชื่อมต่อกับอินเทอร์เน็ตเสมอ
- 3. Actor
 - 3.1 Non – Member คือผู้ใช้งาน (User) ที่กำลังทำการสมัครเข้าสู่ระบบ
 - 3.2 Mobile Application คือตัวแอปพลิเคชันที่อยู่บนมือถือ
 - 3.3 Sensor คือตัวอุปกรณ์ (Tracking) ที่ติดอยู่กับสัตว์เลี้ยง
- 4. Basic Flow of Event
 - 4.1 เมื่อผู้ใช้งาน (User) ทำการเข้าสู่หน้าเรียกดูตำแหน่ง
 - 4.2 เมื่อผู้ใช้งาน (User) ทำการเข้าสู่ระบบ
- 5. Alternative Flow
 - 5.1 เมื่อระบบไม่มีการเชื่อมต่อจะไม่สามารถทำการรับข้อมูลของตำแหน่งจาก Server ได้ ระบบจะทำการแจ้งเตือนเพื่อทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง
- 6. Post conditions
 - 6.1 ระบบทำการตรวจสอบและทำการรับค่าข้อมูลเรียบร้อยแล้ว

2.3 แผนภาพกิจกรรม (Activity Diagram)

2.3.1 Activity Diagram for register

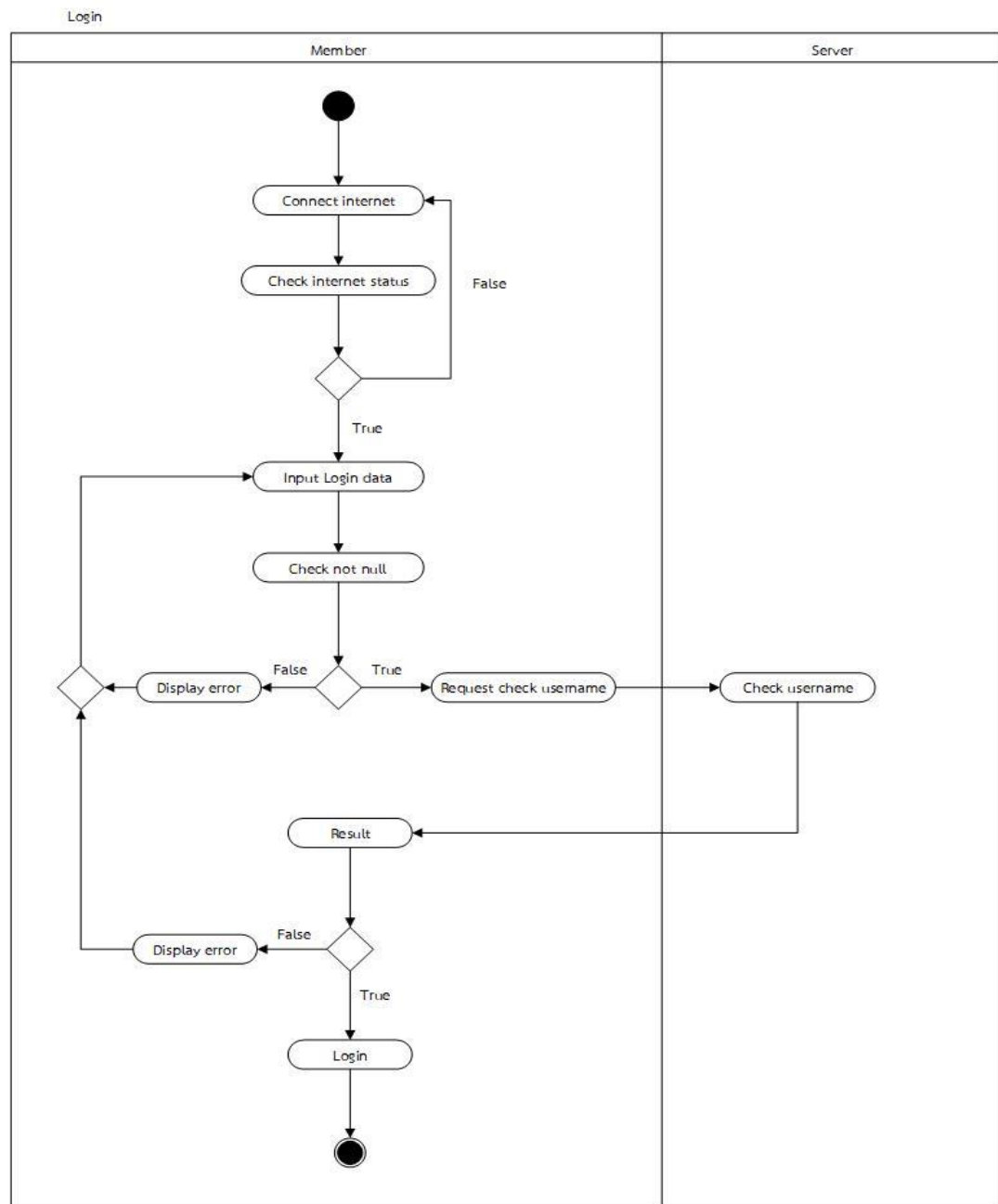


รูปที่ 2.4 Activity Diagram for register on Application mobile

คำอธิบาย Activity Diagram for register on Application mobile

1. Connect internet ทำการเชื่อมต่อกับอินเทอร์เน็ต
2. Check internet status ระบบจะทำการตรวจสอบว่ามีการเชื่อมต่อกับอินเทอร์เน็ตหรือไม่
 - 2.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงมีการเชื่อมต่ออินเทอร์เน็ตเรียบร้อยแล้วให้ทำการทำขั้นตอนต่อไปได้เลย
 - 2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงไม่มีการเชื่อมต่ออินเทอร์เน็ตให้ทำการเชื่อมต่อกับอินเทอร์เน็ตให้เรียบร้อยแล้วจึงจะสามารถทำขั้นตอนต่อไปได้
3. Input Register data ทำการกรอกข้อมูลพื้นฐานในระบบที่มีประกอบไปด้วย ชื่อผู้ใช้งาน (Name), email ,รหัสเข้าใช้งาน (Password), Confirm Password เบอร์โทรศัพท์ (Telephone)
4. Register data not null ระบบจะทำการตรวจสอบข้อมูลที่ผู้ใช้งานได้ทำการกรอกเข้ามา
 - 4.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงระบบได้ทำการตรวจสอบมีการกรอกข้อมูลครบถ้วนไม่มีค่าว่าง
 - 4.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Register data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วน
5. Request same ID ระบบจะทำการตรวจสอบข้อมูลที่ผู้ใช้งาน (User) ได้ทำการกรอกเข้ามา
6. Check same ID ระบบจะทำการตรวจสอบข้อมูลที่ผู้ใช้งาน (User) ที่ได้ทำการกรอกเข้ามามีการสมัครเข้ามาแล้วหรือไม่โดยจะตรวจสอบที่ Server เพื่อทำการเช็คในฐานข้อมูล (firebase) ว่าชื่อผู้ใช้งาน (User) และ อีเมล (Email) ซ้ำกับที่มีอยู่ในระบบหรือไม่
7. Result คือผลลัพธ์ที่ได้จากการตรวจสอบ
 - 7.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงระบบได้ทำการตรวจสอบมีการกรอกข้อมูลครบถ้วนไม่มีค่าว่างและไม่มีชื่อผู้ใช้งาน (User) และ อีเมล (Email) ซ้ำกับที่มีอยู่ในระบบ
 - 7.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Register data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วน
8. Register ทำการสมัครข้อมูลสำเร็จเรียบร้อยแล้ว

2.3.2 Activity Diagram for login



รูปที่ 2.5 Activity Diagram for login on Application mobile

คำอธิบาย Activity Diagram for login on Application mobile

1. Connect internet ทำการเชื่อมต่อกับอินเทอร์เน็ต
2. Check internet status ระบบจะทำการตรวจสอบว่ามีการเชื่อมต่อกับอินเทอร์เน็ตหรือไม่

2.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงมีการเชื่อมต่ออินเทอร์เน็ตเรียบร้อยแล้วให้ทำการทำขั้นตอนต่อไปได้เลย

2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงไม่มีการเชื่อมต่ออินเทอร์เน็ตให้ทำการเชื่อมต่อกับอินเทอร์เน็ตให้เรียบร้อยแล้วจึงจะสามารถทำขั้นตอนต่อไปได้

3. Input Login data ทำการกรอกข้อมูลพื้นฐานในระบบที่มีประกอบไปด้วย email และรหัสผ่านของผู้ใช้งาน (Password) เมื่อทำการกรอกข้อมูลครบถ้วนและถูกต้องระบบจะทำการตรวจสอบและดำเนินการให้ผู้ใช้ (User) เข้าสู่หน้าล็อกอิน (Login) เพื่อใช้งานแอปพลิเคชัน

4. Register not null ระบบทำการตรวจสอบข้อมูลที่ใช้ใช้งานได้ทำการกรอกเข้ามา

4.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงระบบได้ทำการตรวจสอบมีการกรอกข้อมูลครบถ้วนไม่มีค่าว่างหรือกรอกถูกต้องตามที่อยู่ในระบบ การเข้าสู่ระบบประกอบด้วยอีเมล (Email) และ Password

4.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Login data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วนและถูกต้อง

5. Request check username ระบบจะทำการตรวจสอบข้อมูลที่ใช้ใช้งาน (User) ว่าตรงกับที่มีในฐานข้อมูลหรือไม่ถ้ามีจะส่งให้ทำงานขั้นตอนต่อไป

6. Check username ระบบจะทำการตรวจสอบข้อมูลที่ใช้ใช้งาน (User) ที่ได้ทำการกรอกเข้ามามีการสมัครเข้ามาแล้วหรือไม่โดยจะตรวจสอบที่ Server เพื่อทำการเช็คในฐานข้อมูล (firebase) ว่าชื่อผู้ใช้ (User) มีอยู่ในระบบหรือไม่

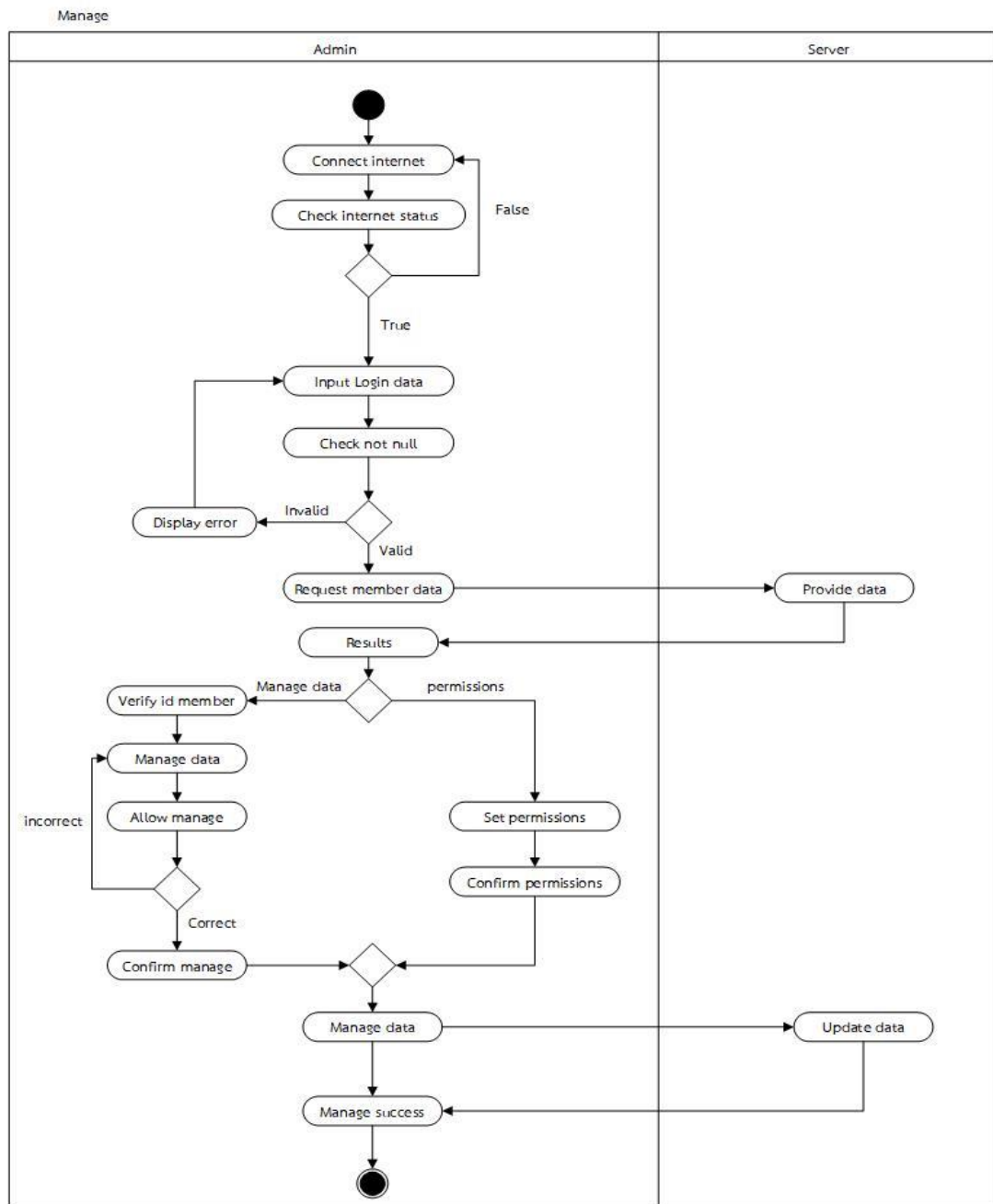
7. Result คือผลลัพธ์ที่ได้จากการตรวจสอบ

7.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงระบบได้ทำการตรวจสอบมีการกรอกข้อมูลครบถ้วนไม่มีค่าว่างและไม่มีชื่อผู้ใช้ (User) และ อีเมล (Email) สามารถเข้าสู่ระบบได้

7.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Login data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วน

8. Login ระบบทำการตรวจสอบผู้ใช้ (User) สามารถเข้าสู่ระบบเรียบร้อยแล้ว

2.3.3 Activity Diagram for Manage



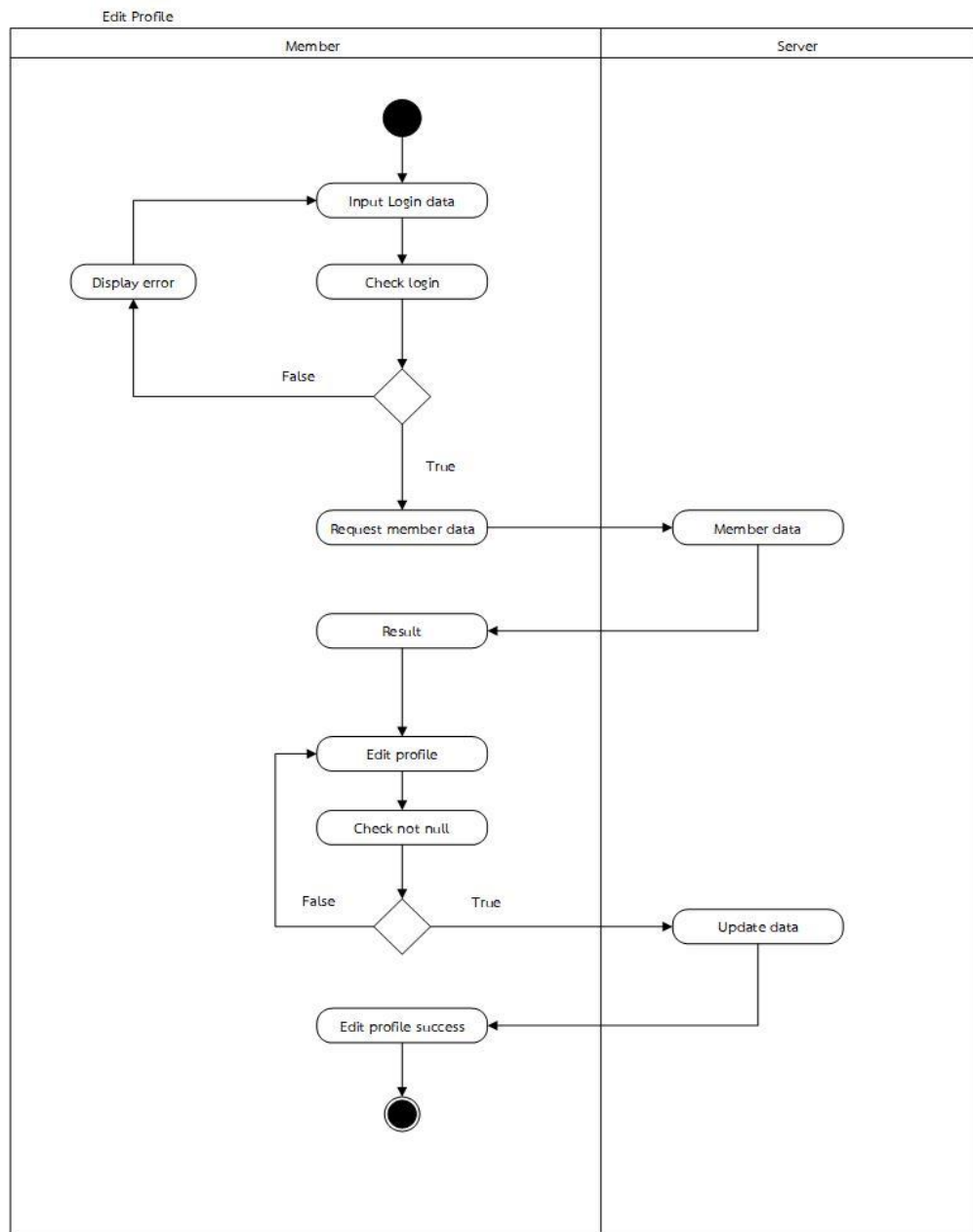
รูปที่ 2.6 Activity Diagram for Manage

คำอธิบาย Activity Diagram for Manage

1. Connect internet ทำการเชื่อมต่อกับอินเทอร์เน็ต
2. Check internet status ระบบจะทำการตรวจสอบว่ามีการเชื่อมต่อกับอินเทอร์เน็ตหรือไม่
 - 2.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงมีการเชื่อมต่ออินเทอร์เน็ตเรียบร้อยแล้วให้ทำการทำขั้นตอนต่อไปได้เลย

- 2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงไม่มีการเชื่อมต่ออินเทอร์เน็ตให้ทำการเชื่อมต่อกับอินเทอร์เน็ตให้เรียบร้อยจึงจะสามารถทำขั้นตอนต่อไปได้
3. Input Login data ทำการกรอกข้อมูลพื้นฐานในระบบที่มีประกอบไปด้วย email และรหัสเข้าใช้งาน (Password)
4. Register not nullระบบทำการตรวจสอบข้อมูลที่คุณดูแลระบบได้ทำการกรอกเข้ามา
- 4.1 Valid: ถ้าเงื่อนไขถูกต้อง ระบบจะทำการตรวจสอบมีการกรอกข้อมูลครบถ้วน ไม่มีค่าว่างหรือกรอกถูกต้องตามที่มีอยู่ในระบบ การเข้าสู่ระบบประกอบด้วยอีเมล (Email) และ Password
- 4.2 Invalid: ถ้าเงื่อนไขไม่เหมาะสมระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Login data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วนและถูกต้อง
5. Request check username ระบบจะทำการตรวจสอบข้อมูลที่คุณใช้งาน (User) ว่าตรงกับที่มีในฐานข้อมูลหรือไม่ถ้ามีจะส่งให้ทำงานขั้นตอนต่อไป
6. Provide dataระบบจะทำการให้ข้อมูลที่คุณดูแลระบบ
7. Result คือผลลัพธ์ที่ได้จากการตรวจสอบ
- 7.1 Manage data : การจัดการกับข้อมูล
- 7.1.1 Verify id member ระบบทำการตรวจสอบ ID ของคุณดูแลระบบ
- 7.1.2 Manage data จัดการกับข้อมูล
- 7.1.3 Allow manage อนุญาตให้คุณดูแลจัดการกับข้อมูล
- 7.1.3.1 Correct : เงื่อนไขที่คุณดูแลระบบสามารถแก้ไขข้อมูลได้
- 7.1.3.2 incorrect : เงื่อนไขไม่ถูกต้องระบบจะส่งค่าไปยัง Manage data เพื่อทำแก้ไขข้อมูลให้อีกครั้ง
- 7.1.4 Confirm manage ระบบทำการตรวจสอบคุณดูแลระบบสามารถแก้ไขข้อมูลได้
- 7.2 Permissions : สิทธิ์ของคุณดูแลระบบ
- 7.2.1 Set permissions กำหนดสิทธิ์ของคุณดูแลระบบ
- 7.2.2 Confirm permissions ยืนยันสิทธิ์ของคุณดูแลระบบ
8. Manage data คุณดูแลระบบการจัดการกับข้อมูลเรียบร้อยแล้วส่งค่าไปยัง Server เพื่อทำการอัปเดตข้อมูลลงในฐานข้อมูล (firebase)
9. Update data ทำการบันทึกข้อมูลลงในฐานข้อมูล (firebase)
10. Manage success คุณดูแลระบบการจัดการกับข้อมูลเสร็จสมบูรณ์

2.3.4 Activity Diagram for Edit profile



รูปที่ 2.7 Activity Diagram for edit profile

คำอธิบาย Activity Diagram for edit profile

1. Input Login data ทำการกรอกข้อมูลพื้นฐานในระบบที่มีประกอบไปด้วย email และรหัสเข้าใช้งาน (Password)

2. Check Login ระบบจะทำการตรวจสอบข้อมูลของผู้ใช้งาน (User) ที่ได้ทำการเข้าสู่ระบบ

2.1 True : ถ้าเงื่อนไขเป็นจริงหมายถึงระบบได้ทำการตรวจสอบแล้วให้ทำขั้นตอนต่อไปได้

2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Login data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วนและถูกต้อง

3. Request member data ระบบจะทำการตรวจสอบข้อมูลที่คุณดูและระบบว่าตรงกับที่มีในฐานข้อมูลหรือไม่ถ้ามีจะส่งให้ทำงานขั้นตอนต่อไป

4. Member data เป็นการทำงานฝั่ง Server ที่มีดึงข้อมูลจากฐานข้อมูลมาตรวจสอบ

5. Result คือผลลัพธ์ที่ได้จากการตรวจสอบข้อมูลเบื้องต้น

6. Edit profile ทำการแก้ไขข้อมูลที่มีอยู่ในระบบได้

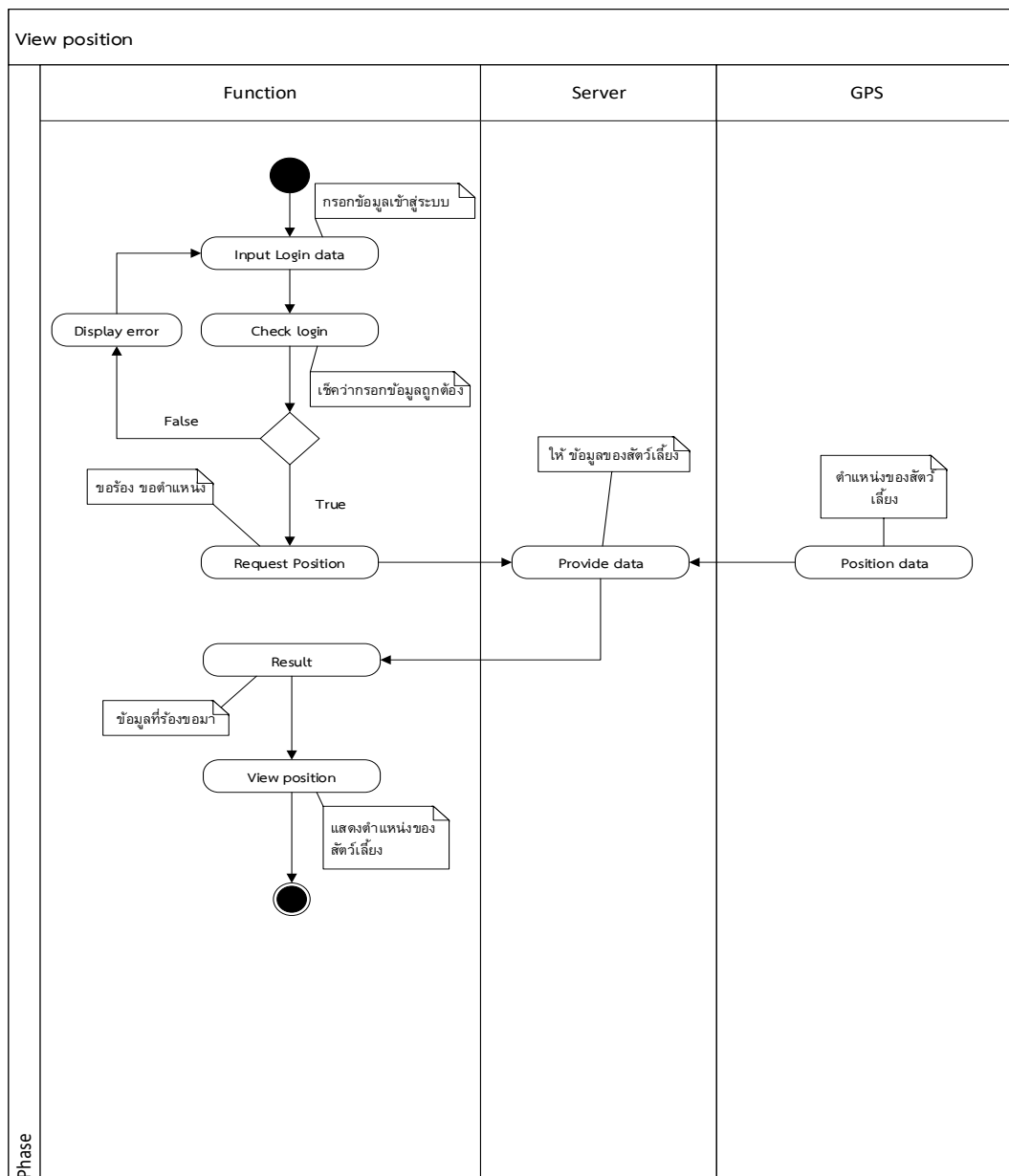
7. Check not null ระบบทำการตรวจสอบค่าข้อมูลว่าไม่มีค่าว่างและทำการกรอกข้อมูลครบถ้วนตามที่ระบบต้องการ ถ้าทำครบตามที่ระบุจึงจะสามารถทำงานในขั้นตอนต่อไปได้

7.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงระบบได้ทำการตรวจสอบมีการกรอกข้อมูลครบถ้วนไม่มีค่าว่างและทำการกรอกข้อมูลครบถ้วนจะถูกส่งค่าไปยัง Server เพื่อทำการบันทึกลงในฐานข้อมูล (firebase)

7.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Edit profile เพื่อทำการแก้ไขใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วน

8. Edit profile success ทำการแก้ไขข้อมูลเสร็จสมบูรณ์

2.3.5 Activity Diagram for View position



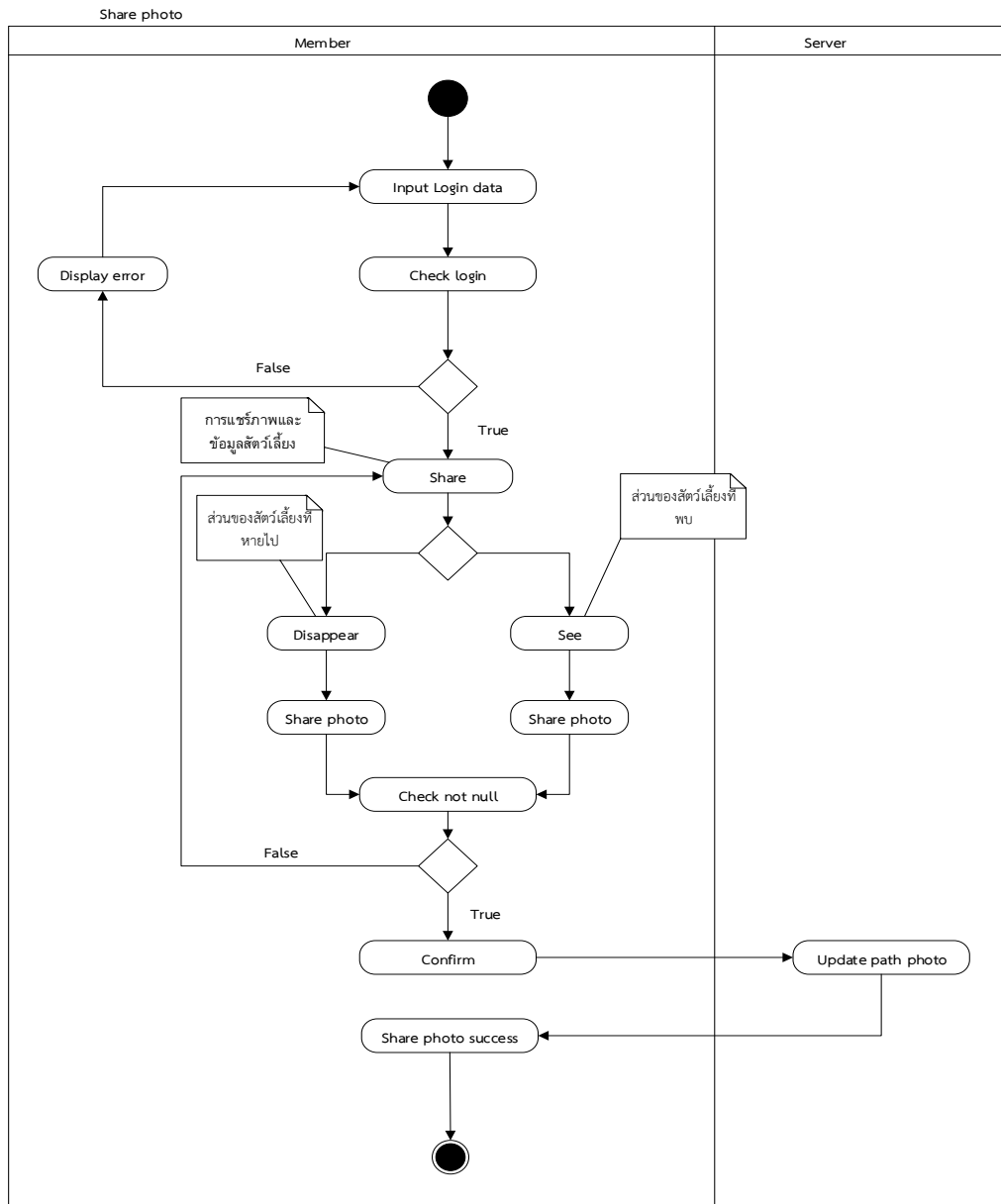
รูปที่ 2.8 Activity Diagram for View position

คำอธิบาย Activity Diagram for View position

1. Input Login data ทำการกรอกข้อมูลพื้นฐานเพื่อเข้าสู่ระบบประกอบไปด้วย email และรหัสเข้าใช้งาน (Password)
2. Check Login ระบบจะทำการตรวจสอบข้อมูลของผู้ใช้งาน (User) ที่ได้ทำการเข้าสู่ระบบ
 - 2.1 True : ถ้าเงื่อนไขเป็นจริงหมายถึงระบบได้ทำการตรวจสอบแล้วให้ทำขั้นตอนต่อไปได้
 - 2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Login data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วนและถูกต้อง

3. Request Position มีการร้องขอตำแหน่งที่อยู่ของสัตว์เลี้ยง ระบบจะทำการตรวจสอบการร้องขอค่าข้อมูล ตำแหน่งที่อยู่ของสัตว์เลี้ยงที่จากตัวอุปกรณ์ที่ติดอยู่กับสัตว์เลี้ยงในขณะนั้นมาแสดงบนแอปพลิเคชันติดตามสัตว์เลี้ยง
4. Provide data เป็นการทำงานฝั่ง Server ระบบจะทำการตรวจสอบและทำการให้ตำแหน่งที่อยู่ของสัตว์เลี้ยง
5. Position data เป็นการทำงานฝั่ง GPS หรือตัวอุปกรณ์ (Tracking) ที่ได้ทำการติดตั้งกับสัตว์เลี้ยงเรียบร้อยแล้ว Position data จะทำการส่งค่าข้อมูล ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time) ให้กับ Server เพื่อทำการเก็บค่าเหล่านี้ลงฐานข้อมูล (firebase)
6. Result คือผลรับที่ได้จากการร้องขอข้อมูลเบื้องต้นมา มีการส่ง ละติจูด (Latitude) ลองจิจูด (Longitude) จากตัวอุปกรณ์
7. View position ระบบทำการตรวจสอบและแสดงตำแหน่งที่อยู่ของสัตว์เลี้ยง

2.3.6 Activity Diagram for Share photo



รูปที่ 2.9 Activity Diagram for Share photo

คำอธิบาย Activity Diagram for Share photo

1. Input Login data ทำการกรอกข้อมูลพื้นฐานเพื่อเข้าสู่ระบบประกอบไปด้วย email และรหัสเข้าใช้งาน (Password)
2. Check Login ระบบจะทำการตรวจสอบข้อมูลที่ใช้ใช้งาน (User) ที่ได้ทำการเข้าสู่ระบบ
 - 2.1 True : ถ้าเงื่อนไขเป็นจริงหมายถึงระบบได้ทำการตรวจสอบแล้วให้ทำขั้นตอนต่อไปได้

2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะแสดงข้อความ Display error และส่งค่าไปยัง Input Login data ใหม่อีกครั้งเพื่อทำการกรอกข้อมูลให้ครบถ้วนและถูกต้อง

3.Shareผู้ใช้งาน (User) ทำการแชร์รูปภาพและข้อมูลของผู้ใช้งาน (User) โดยการแชร์มี 2 แบบ

3.1 Disappearเป็นในส่วนของการแชร์ภาพสัตว์เลี้ยงที่หาไม่พบและต้องการให้ผู้พบเห็นช่วยตามหาหรือติดต่อกลับมา

3.1.1 Share photoเป็นการแชร์ภาพสัตว์เลี้ยงที่หาไม่พบหรือออกบริเวณที่เจ้าของไม่พบเป็นเวลานาน

3.2. Seeเป็นในส่วนของการแชร์ภาพสัตว์เลี้ยงที่หาพบกันระหว่างผู้ใช้งาน (User) และต้องการติดต่อหรือประกาศให้เจ้าของสัตว์เลี้ยงทราบเพื่อทำการติดต่อกลับมา

3.2.1. Share photoเป็นการแชร์ภาพสัตว์เลี้ยงที่หาพบและติดการประกาศหรือต้องการให้เจ้าของสัตว์เลี้ยงติดต่อกลับมา

4. Check not nullระบบทำการตรวจสอบค่าข้อมูลว่ากรอกครบถ้วนและได้ทำการแชร์ภาพเรียบร้อยแล้ว

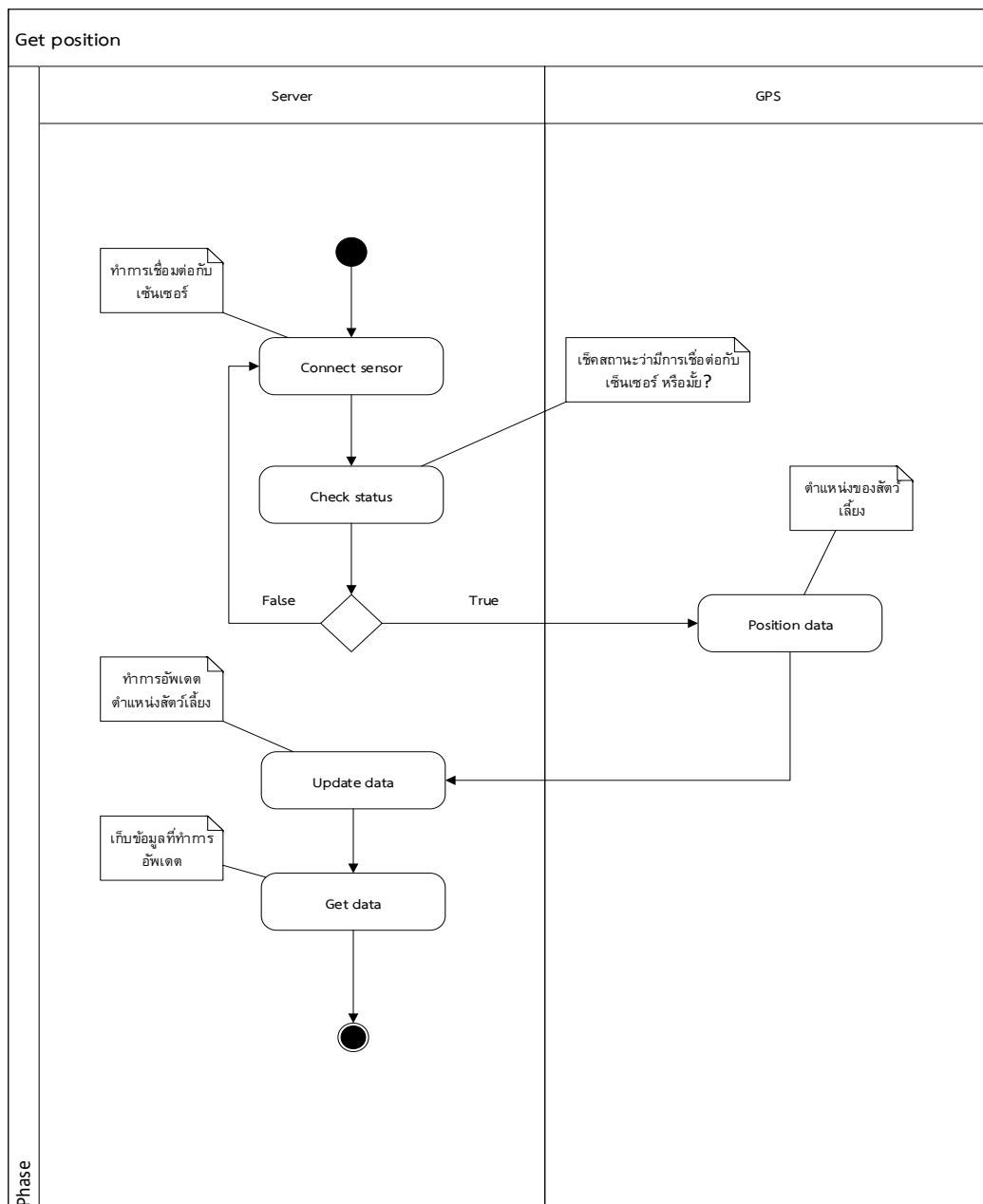
4.1 True : เงื่อนไขเป็นจริง หมายถึงไม่มีค่าว่างและทำการกรอกข้อมูลที่มีอยู่ในระบบครบถ้วน

4.2 False : เงื่อนไขเป็นเท็จ หมายถึงมีค่าว่างและทำการกรอกข้อมูลที่มีอยู่ในระบบไม่ครบถ้วนระบบจะส่งค่าไปยัง Share เพื่อให้ผู้ใช้งาน (User) ทำการกรอกข้อมูลใหม่อีกครั้ง

5. Confirm ผู้ใช้งาน (User) ทำการแชร์ภาพและข้อมูลทำการกดยืนยันเพื่อให้ข้อมูล Update

6. Update path photo เป็นการทำงานในฝั่งของ Server ระบบจะทำการเก็บภาพและข้อมูลที่ผู้ใช้งาน (User) ทำการ Update ไปยังฐานข้อมูล (firebase)

7. Share photo success ระบบทำการตรวจสอบผู้ใช้งาน (User) สามารถแชร์รูปภาพของสัตว์เลี้ยงได้เรียบร้อยแล้ว



รูปที่ 2.10 Activity Diagram Get position

คำอธิบาย Activity Diagram for Get position

1. Connect sensor ทำการเชื่อมต่อกับตัวอุปกรณ์โดยการเชื่อมต่อต้องมียินเทอร์เน็ตตลอดเวลา
2. Check status ระบบจะทำการตรวจสอบสถานะว่ามีการเชื่อมต่อกับตัวอุปกรณ์ (sensor) หรือไม่
 - 2.1 True : ถ้าเงื่อนไขเป็นจริง หมายถึงระบบได้ทำการตรวจสอบว่ามีการเชื่อมต่อกับตัวอุปกรณ์ (sensor) จริงให้ทำขั้นตอนต่อไป

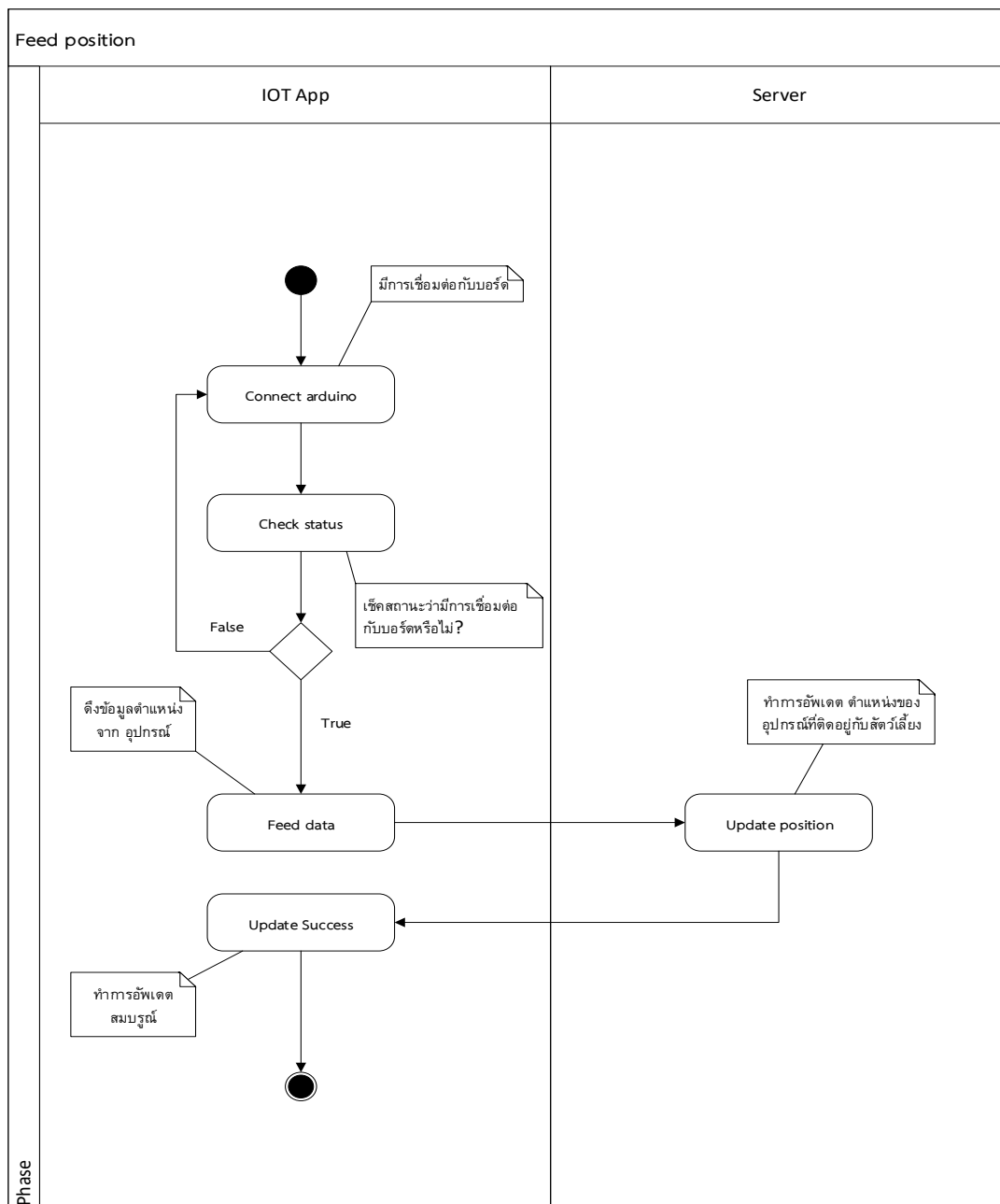
2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะส่งค่าไปยัง Connect sensor อีกครั้งเพื่อทำการเชื่อมต่อ

3. Position data เป็นการทำงานในฝั่งของ GPS โดยการทำงานของ Position data ทำหน้าที่ในการให้ข้อมูลตำแหน่งที่อยู่ของสัตว์เลี้ยงแก่ผู้ใช้งาน (User) ในระบบ

4. Update data ทำหน้าที่ในการ Update ตำแหน่งที่อยู่ของสัตว์เลี้ยงประกอบไปด้วย ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time) และเก็บค่าลงในฐานข้อมูล (Firebase) มีการ Update data แบบ Real-time บน Firebase และการอัปเดตข้อมูลแต่ละครั้งขึ้นอยู่กับความเร็วอินเทอร์เน็ตของผู้ใช้งาน (User) แต่ละบุคคล

5. Get data ทำหน้าที่รับข้อมูลที่ทำการอัปเดตลงลงไปเก็บไว้ในฐานข้อมูล (Firebase) ประกอบไปด้วย ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time) มีการรับค่าที่ผ่านการ Update data แบบ Real-time บน Firebase และการอัปเดตข้อมูลแต่ละครั้งขึ้นอยู่กับความเร็วอินเทอร์เน็ตของผู้ใช้งาน (User) แต่ละบุคคล

2.3.8 Activity Diagram for Feed position



รูปที่ 2.11 Activity Diagram Feed position

คำอธิบาย Activity Diagram for Get position

1. Connect arduino ทำการเชื่อมต่อกับตัวอุปกรณ์โดยการเชื่อมต่อต้องมีอินเทอร์เน็ตตลอดเวลา
2. Check status ระบบจะทำการตรวจสอบสถานะว่ามีการเชื่อมต่อกับตัวอุปกรณ์ (arduino) หรือไม่

2.1 True : ถ้าเงื่อนไขเป็นจริงหมายถึงระบบได้ทำการตรวจสอบว่ามีการเชื่อมต่อกับตัวอุปกรณ์ (arduino) จริงให้ทำขั้นตอนต่อไป

2.2 False : ถ้าเงื่อนไขเป็นเท็จ หมายถึงระบบจะส่งค่าไปยังConnect arduino อีกครั้งเพื่อทำการเชื่อมต่อ

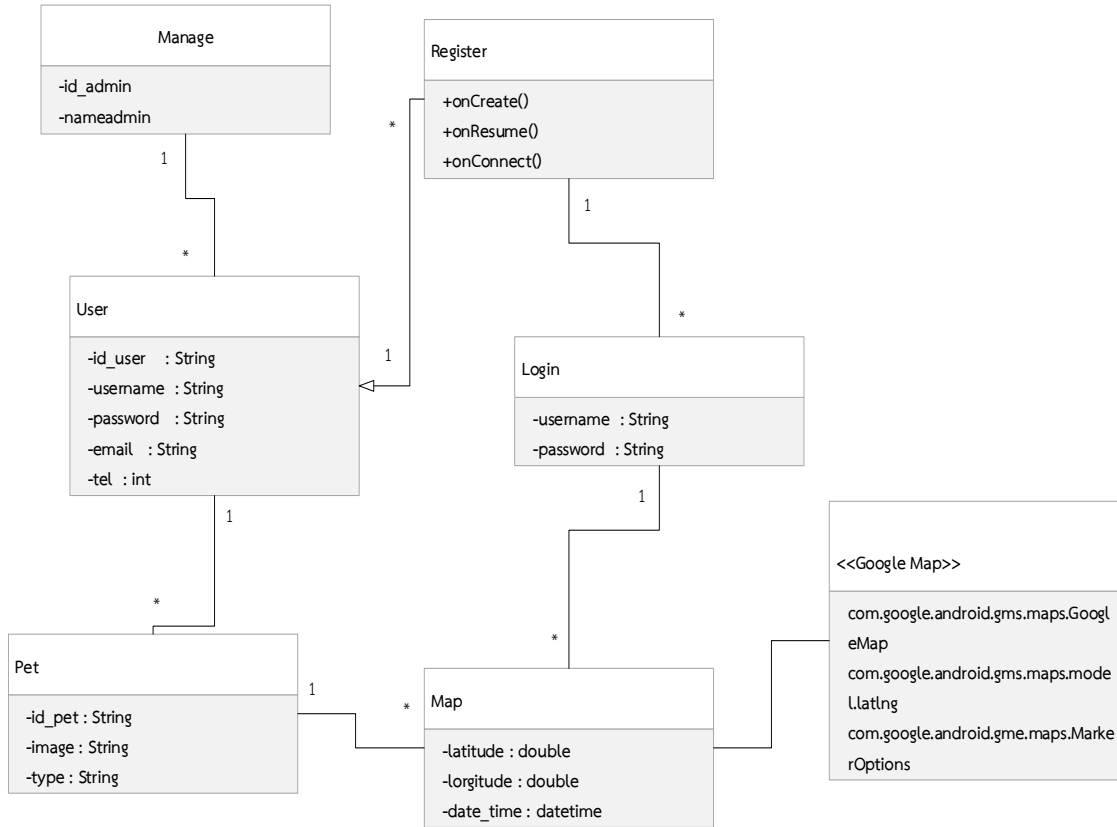
3. Feed dataเป็นการทำงานในฝั่งของ Serverโดยการทำงานของ Feed data ทำหน้าที่ในการดึงตำแหน่งของตัวอุปกรณ์ที่ติดอยู่กับสัตว์เลี้ยง

4. Update data ทำหน้าที่ในการ Update ตำแหน่งที่อยู่ของตัวอุปกรณ์ประกอบไปด้วย ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time) และเก็บค่าลงในฐานข้อมูล (Firebase)

5. Get data ทำหน้าที่เก็บข้อมูลที่ทำกรอัปเดตลงในฐานข้อมูล (Firebase) ประกอบไปด้วย ละติจูด (Latitude) ลองจิจูด (Longitude) วัน (Date) และเวลา (Time)

2.4 คลาสไดอะแกรม (Class Diagram)

แผนภาพที่แสดงคลาสและความสัมพันธ์ระหว่าง Class ของระบบที่สนใจ (Problem Domain)



รูปที่ 2.12 Class Diagram pet track application on android mobile

คำอธิบาย Class ของระบบการจัดเก็บข้อมูลของแอปพลิเคชันที่แสดงตำแหน่งและการทำงานของระบบแอปพลิเคชัน บนแอนดรอยด์รายละเอียดและความสัมพันธ์การทำงานของแต่ละ Class diagram มีดังต่อไปนี้

1. เส้น multiplicity เป็นเชื่อมความสัมพันธ์ของระบบที่แสดงจำนวนของสมาชิกที่อยู่ในความสัมพันธ์
2. Manage: Manage 1 คน สามารถแก้ไขข้อมูล User ได้หลายครั้ง
การแก้ไข User 1 ครั้ง ถูกกระทำโดย Manage 1 คน
3. Register: User 1 คน สามารถล็อกอิน (Login) ได้หลายครั้ง
การล็อกอิน (Login) 1 ครั้ง ถูกกระทำโดย User 1 คน
4. Login: การล็อกอิน (Login) 1 ครั้ง สามารถดูตำแหน่งสัตว์เลี้ยงได้หลายตำแหน่ง

ตำแหน่งสัตว์เลี้ยงหลายๆ ตำแหน่งสามารถดูได้จากการการล็อกอิน (Login) 1 ครั้ง

5. User: User 1 คน สามารถสมัครสมาชิกได้หลายครั้ง

การสมัครสมาชิกได้หลายครั้งถูกกระทำโดย User 1 คน

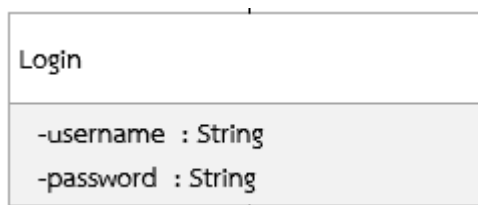
6. Pet: User 1 คน สามารถดูข้อมูลสัตว์เลี้ยงได้หลายครั้ง

ข้อมูลสัตว์เลี้ยงหลายๆ ข้อมูลถูกกระทำโดย User 1 คน

7. <<Google Map>> เป็นการเรียก Library จากข้างนอกเข้ามาใช้งานเพื่อแสดงตำแหน่งของสัตว์เลี้ยง

คำอธิบาย Class Diagram of pet track application on android mobile โดยมีคำอธิบายแต่ละ Class diagram มีดังต่อไปนี้

2.4.1 Class Login



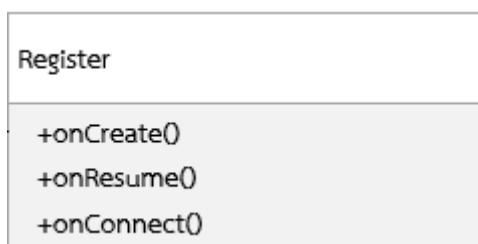
รูปที่ 2.13 Class Diagram of Login

Attribute ประกอบด้วย

(1) -Email สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บอีเมล (Email) ของผู้ใช้งาน (User) และมีการเก็บค่าข้อมูลเป็น String หมายถึงชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่

(2) -Password สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บรหัสผ่าน (Password) ใช้งาน (User) และมีการเก็บค่าข้อมูลเป็น String หมายถึงชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่

2.4.2 Class Register



รูปที่ 2.14 Class Diagram of Register

Attribute ประกอบด้วย

(1) +onCreate() สัญลักษณ์ (+) ใช้สำหรับเก็บข้อมูลข้อมูลที่สามารถมองเห็นได้จากภายนอกและ onCreateทำหน้าที่ในการสมัครสมาชิก

(2) +onResume() สัญลักษณ์ (+) ใช้สำหรับเก็บข้อมูลข้อมูลที่สามารถมองเห็นได้จากภายนอกและ onResume เป็นคำสั่งที่ใช้ในการสั่งให้เริ่มทำงานใหม่ๆในส่วนที่รันไม่ผ่าน

(3) +onConnect() สัญลักษณ์ (+) ใช้สำหรับเก็บข้อมูลข้อมูลที่สามารถมองเห็นได้จากภายนอกและ onConnect ทำหน้าที่เพื่อรองขอการเชื่อมต่อกับอินเทอร์เน็ต

2.4.3 Class Manage



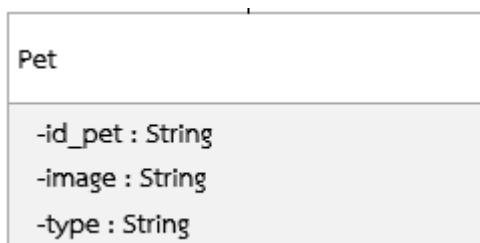
รูปที่ 2.15 Class Diagram of Manage

Attribute ประกอบด้วย

(1) -id_admin สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บID ของadmin

(2) -nameadmin สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บชื่อของผู้ใช้งาน (nameadmin) ของ admin

2.4.4 Class Pet



รูปที่ 2.16 Class Diagram of Pet

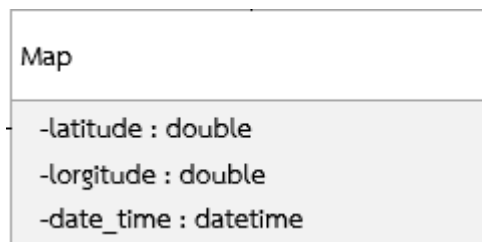
Attribute ประกอบด้วย

(1) -id_pet สัญลักษณ์ (+) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอกมีการเก็บค่า id_pet ทำหน้าที่เก็บค่า ID สัตว์เลี้ยงและมีการเก็บค่าข้อมูลเป็น String หมายถึงชุด (array) ของตัวอักษร (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่

(2) -image สัญลักษณ์ (+) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีการเก็บค่า image ทำหน้าที่เก็บค่ารูปภาพของสัตว์เลี้ยง และมีการเก็บค่าข้อมูลเป็นString หมายถึงชุด(array) ของตัวอักษร (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่

(3) -type() สัญลักษณ์ (+) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีการเก็บค่า type ทำหน้าที่เก็บค่าชนิดของสัตว์เลี้ยงและมีการเก็บค่าข้อมูลเป็น String หมายถึงชุด(array) ของตัวอักษร (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่

2.4.5 Class Map



รูปที่ 2.17 Class Diagram of Map

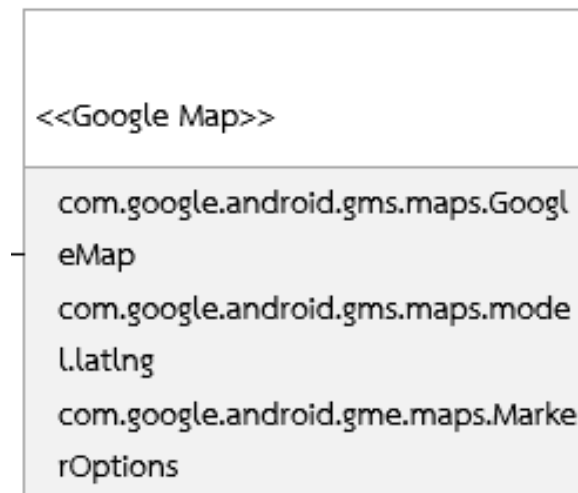
Attribute ประกอบด้วย

(1) -latitude สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บlatitudeหรือเดิมเรียกว่า เส้นรุ้ง แทนด้วยอักษรกรีก ϕ เป็นพิกัดที่ใช้บอกตำแหน่งบนพื้นโลก และแบ่งเขตสภาวะอากาศโดยวัดจากเส้นศูนย์สูตร พิกัดที่ใช้คู่กัน คือ ลองจิจูดเป็นการเก็บค่าตำแหน่งที่ตั้งของสัตว์และมีการเก็บค่าข้อมูลเป็นdoubleหมายถึงสามารถเก็บค่าเป็นจำนวนทศนิยมได้

(2) -longitude สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บ longitude หรือเดิมเรียกว่า เส้นแวง แทนด้วยอักษรกรีก λ เป็นพิกัดที่ใช้บอกตำแหน่งบนพื้นโลก โดยวัดไปทางตะวันออกหรือตะวันตกจากเส้นสมมติในแนวเหนือ-ใต้ที่เรียกว่าเส้นเมริเดียนแรก พิกัดที่ใช้คู่กัน คือ ละติจูดเป็นการเก็บค่าตำแหน่งที่ตั้งของสัตว์เลี้ยงและมีการเก็บค่าข้อมูลเป็นdoubleหมายถึงสามารถเก็บค่าเป็นจำนวนทศนิยมได้

(3) -date_time สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บวัน (date) และเวลา (time) เป็นการวันเวลาปัจจุบันของสัตว์และมีการเก็บค่าข้อมูลเป็นdoubleหมายถึงสามารถเก็บค่าเป็นจำนวนทศนิยมได้

2.4.6 Class Google Map



รูปที่ 2.18 Class Diagram of Google Map

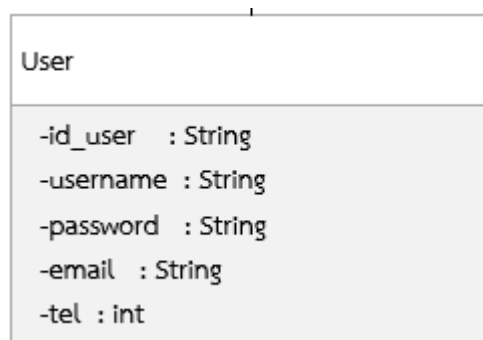
Class Diagram of Google Map เป็น Library ที่ถูกเรียกใช้งานเพื่อใช้ในการแสดงผลของ Map ประกอบด้วย

(1) com.google.android.gms.maps.GoogleMap คือคลาสหลักของ Google Maps Android API และจุดเริ่มต้นสำหรับวิธีการทั้งหมดที่เกี่ยวข้องกับแผนที่

(2) com.google.android.gms.maps.model.LatLng คือคลาสที่ไม่มีการเปลี่ยนรูปแสดง พิกัดละติจูดลองจิจูดเก็บเป็นองศาสรุปค่าข้อมูลคงที่จากส่วนที่มีการติดต่อ

(3) com.google.android.gms.maps.MarkerOptions คือ Marker (ปักหมุด) Location ลงใน Google Map เป็นวิธีการการระบุตำแหน่งหรือพิกัด Location ที่เราต้องการทำเป็นสัญลักษณ์ โดยตำแหน่งที่จะปักหมุด (Marker) นั้นเราจะอ้างจากค่า Latitude, Longitude

2.4.7 Class User



รูปที่ 2.19 Class Diagram of User

Attribute ประกอบด้วย

(1) -id_user สัญลักษณ์ (-) หรือเครื่องหมาย (Symbol) ตัวนี้เป็นเครื่องหมายใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก หมายความว่าบุคคลภายนอกไม่สามารถเปิดหรือดูค่าข้อมูลที่อยู่ในฐานข้อมูลได้นอกจากผู้ดูแลระบบ มีเก็บ ID ของผู้ใช้งาน (User)

ในส่วนของการเก็บ ID ของผู้ใช้งานจะถูกเก็บค่าเป็น String หมายถึงประเภทข้อมูลประเภทข้อความหรือการนำตัวอักษรหลายๆ ตัวมาต่อกันเรียกว่าชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่ โดยความยาวของ String นั้นสามารถเปลี่ยนแปลงได้ตามค่าที่กำหนดให้กับตัวแปร

(2) -Username สัญลักษณ์ (-) หรือเครื่องหมาย (Symbol) ตัวนี้เป็นเครื่องหมายใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บชื่อ (username) ผู้ใช้งานในระบบ ค่าข้อมูลผู้ใช้งานถูกเก็บค่าเป็น String หมายถึงประเภทข้อมูลประเภทข้อความหรือการนำตัวอักษรหลายๆ ตัวมาต่อกันเรียกว่าชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่ โดยความยาวของ String นั้นสามารถเปลี่ยนแปลงได้ตามค่าที่กำหนดให้กับตัวแปร

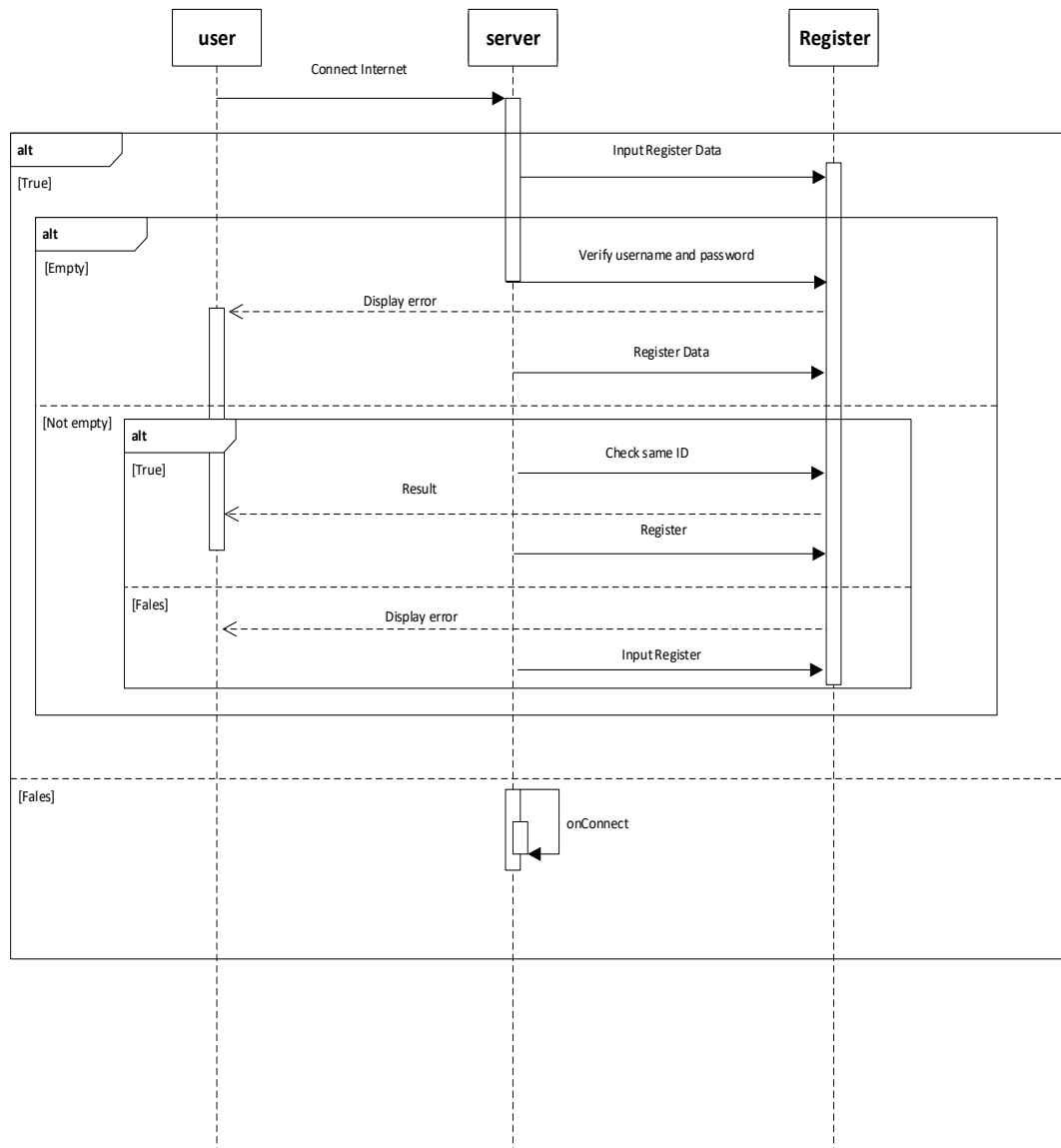
(3) -password สัญลักษณ์ (-) ใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก มีเก็บรหัสผ่านของผู้ใช้งาน และมีการเก็บค่าข้อมูลเป็น String หมายถึงชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่ โดยความยาวของ String นั้นสามารถเปลี่ยนแปลงได้ตามค่าที่กำหนดให้กับตัวแปร

(4) -Email สัญลักษณ์ (-) หรือเครื่องหมาย (Symbol) ตัวนี้เป็นเครื่องหมายใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก หมายความว่าบุคคลภายนอกไม่สามารถเปิดหรือดูค่าข้อมูลที่อยู่ในฐานข้อมูลได้นอกจากผู้ดูแลระบบ มีเก็บอีเมล (Email) ของผู้ใช้งานในส่วนของการเก็บค่าอีเมลของผู้ใช้งานจะถูกเก็บค่าเป็น String หมายถึงประเภทข้อมูลประเภทข้อความหรือการนำตัวอักษรหลายๆ ตัวมาต่อกันเรียกว่าชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่ โดยความยาวของ String นั้นสามารถเปลี่ยนแปลงได้ตามค่าที่กำหนดให้กับตัวแปร

(5) -tel สัญลักษณ์ (-) หรือเครื่องหมาย (Symbol) ตัวนี้เป็นเครื่องหมายใช้สำหรับเก็บข้อมูลข้อมูลที่ไม่สามารถมองเห็นได้จากภายนอก หมายความว่าบุคคลภายนอกไม่สามารถเปิดหรือดูค่าข้อมูลที่อยู่ในฐานข้อมูลได้นอกจากผู้ดูแลระบบ มีเก็บเบอร์โทรศัพท์ของผู้ใช้งานในส่วนของการเก็บค่าเบอร์โทรศัพท์ของผู้ใช้งานจะถูกเก็บค่าเป็น String หมายถึงประเภทข้อมูลประเภทข้อความหรือการนำตัวอักษรหลายๆ ตัวมาต่อกันเรียกว่าชุด (array) ของตัวอักขระ (character) ทั้งพิมพ์เล็กพิมพ์ใหญ่ โดยความยาวของ String นั้นสามารถเปลี่ยนแปลงได้ตามค่าที่กำหนดให้กับตัวแปร

2.5 แผนภาพกิจกรรม (Sequence Diagram)

2.5.1 Sequence Diagram for register



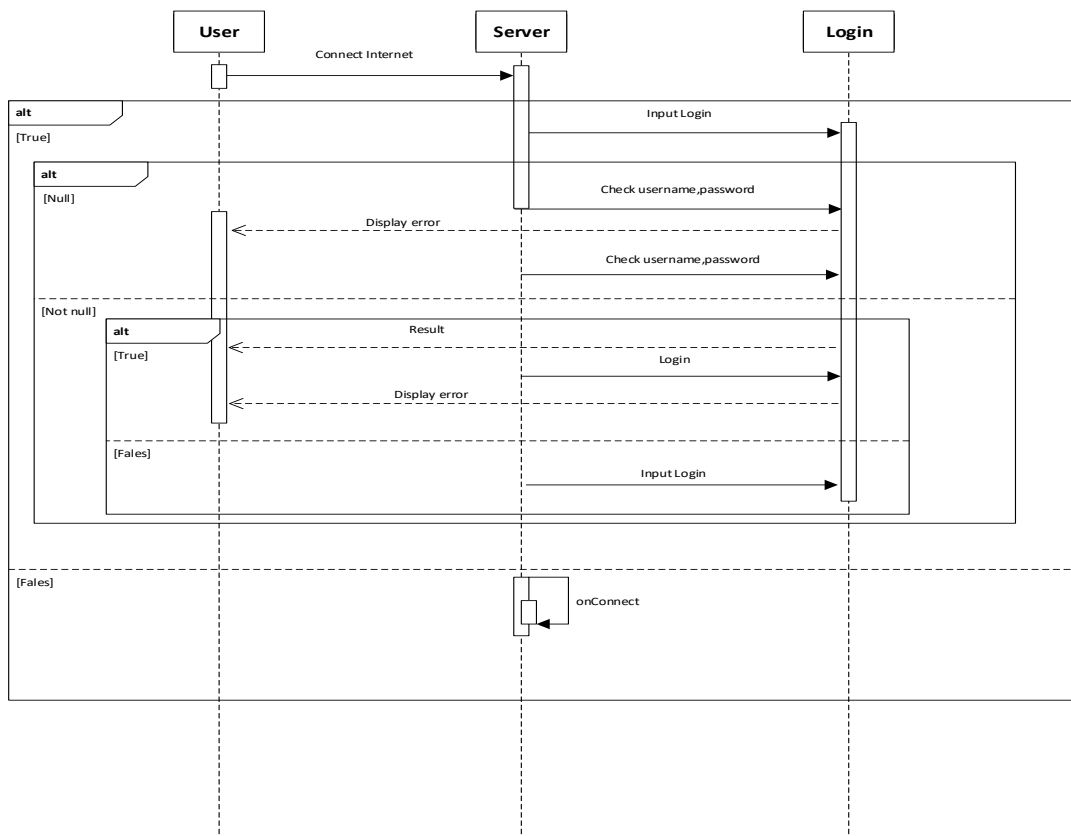
รูปที่ 2.20 Sequence Diagram for register

คำอธิบาย กระบวนการทำงานของ Sequence Diagram for register

- (1) User หมายถึงผู้ที่สมัครใช้งานในระบบ
- (2) Server หมายถึงการเชื่อมต่อกับฐานข้อมูล (firebase)
- (3) Register หมายถึงการสมัครสมาชิกเพื่อใช้งานในระบบโดยการสมัครต้องมีการกรอกข้อมูลให้ครบถ้วนประกอบไปด้วยชื่อใช้งาน (Name), อีเมล (Email), รหัสผ่านใช้งาน (Password), ทำการยืนยันพาสเวิร์ดอีกครั้ง (Confirm Password), เบอร์โทรศัพท์ (Telephone)

- (4) Connect Internet หมายถึงทำการเชื่อมต่อกับอินเทอร์เน็ต
- (5) Input Register Data หมายถึงทำการกรอกข้อมูลพื้นฐานในระบบที่มีประกอบไปด้วยชื่อผู้ใช้งานที่ทำการสมัคร (Name), email, เข้าใช้งาน (Password), Confirm Password เบอร์โทรศัพท์ (Telephone)
- (6) Verify username and password หมายถึงข้อมูลที่ร้องขอเพื่อตรวจสอบว่าเคยมีการสมัครก่อนหน้านี้หรือไม่และทำการยืนยันชื่อผู้ใช้และรหัสผ่าน
- (7) Check same ID หมายถึงทำการตรวจสอบว่ามีการซ้ำกันหรือไม่ เช่น อีเมลนี้เคยสมัครมาแล้ว และมีข้อมูลในระบบ
- (8) Result หมายถึงข้อมูลที่ร้องขอจาก User เพื่อให้ทำการกรอกข้อมูลใหม่อีกครั้ง
- (9) Input Register หมายถึงกรอกข้อมูลการสมัครอีกครั้ง
- (10) onConnect หมายถึงทำการเชื่อมต่อกรณีเป็น False ก็ทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง

2.5.2 Sequence Diagram for login



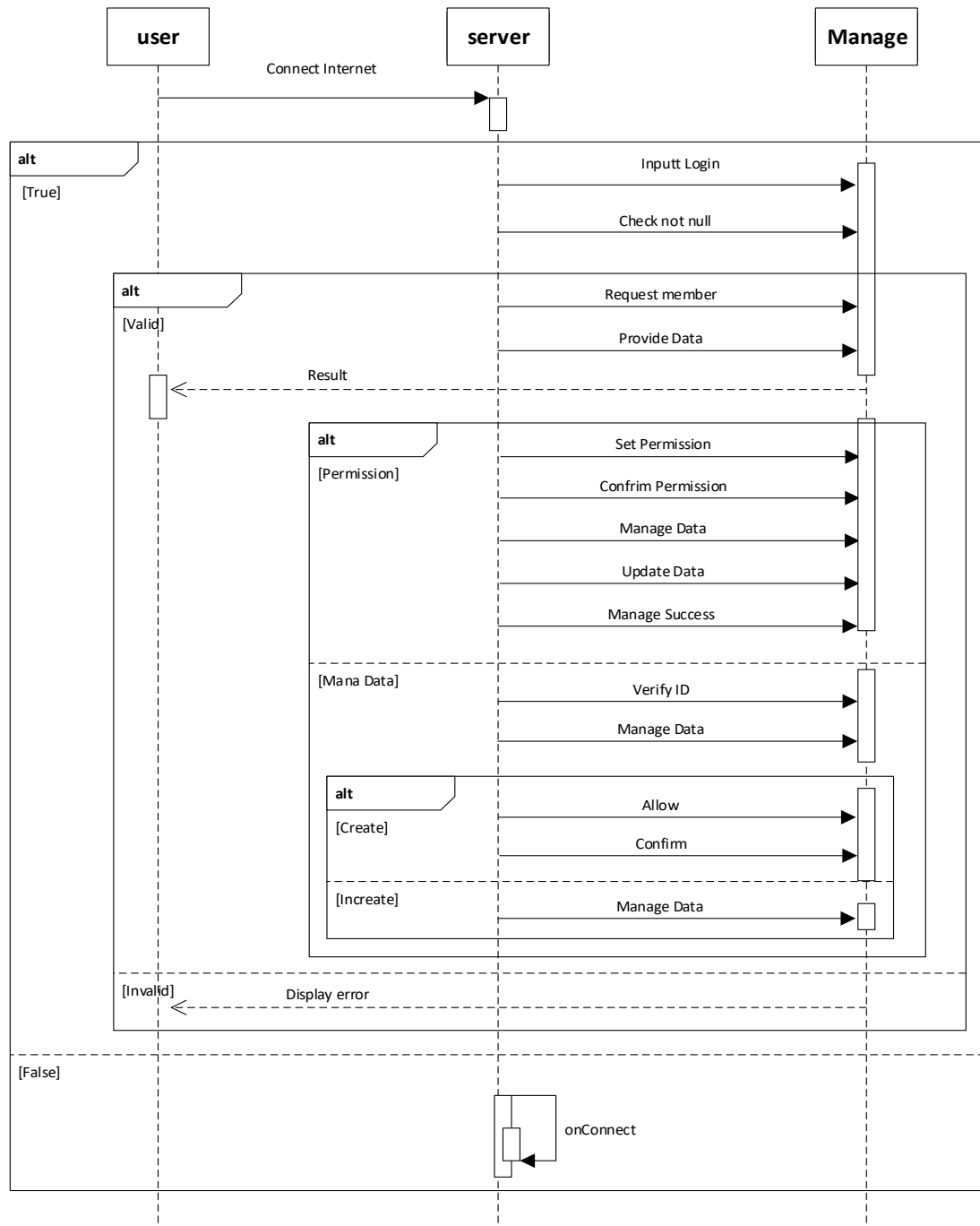
รูปที่ 2.21 Sequence Diagram for login

คำอธิบายกระบวนการทำงานของ Sequence Diagram for login

- (1) Server หมายถึงการเชื่อมต่อกับฐานข้อมูล (firebase)
- (2) Register หมายถึงการสมัครสมาชิกเข้าใช้งานระบบแอปพลิเคชันติดตามสัตว์เลี้ยง
- (3) Connect Internet หมายถึงทำการเชื่อมต่ออินเทอร์เน็ต
- (4) Check username, password หมายถึงระบบทำการตรวจสอบ ชื่อผู้ใช้ (User) และรหัสผ่าน (Password) ว่าตรงที่มีอยู่ในฐานข้อมูล (firebase) หรือไม่ ถ้าระบบทำการตรวจสอบแล้วไม่มีอยู่ในฐานข้อมูล (firebase) ระบบจะทำการแจ้ง Display error ให้ผู้ใช้งาน (User) ได้ทราบทันที
- (5) Result หมายถึงข้อมูลที่ร้องขอจากฐานข้อมูล (firebase)
- (6) Login หมายถึงการเข้าใช้งานแอปพลิเคชันติดตามสัตว์เลี้ยง
- (7) Display error หมายถึงระบบทำการตรวจสอบแล้วไม่ตรงตามเงื่อนไขที่ระบบกำหนด ระบบจะทำการแสดงข้อความ error เพื่อบอกให้ผู้ใช้งาน (User) ทราบว่าผู้ใช้งาน (User) ไม่สามารถทำการล็อกอิน (Login) เข้าสู่ระบบได้ให้ผู้ใช้งาน (User) ให้ทำการตรวจสอบอีเมล (Email) ที่กรอกเข้ามาหรือทำการตรวจสอบรหัสผ่าน (Password) ว่าถูกต้องหรือไม่อีกครั้งเพื่อทำการล็อกอิน (Login) เข้าสู่ระบบผู้ใช้งาน

- (8) Input Login หมายถึงใส่ข้อมูลที่ User ทำการ Register ผ่านแล้วโดยประกอบไปด้วย อีเมล(email), รหัสผ่าน(Password) จากนั้นจึงจะสามารถเข้าสู่ระบบได้
- (9) onConnect หมายถึงทำการเชื่อมต่อกรณีเป็น False ก็ทำการเชื่อมต่ออินเทอร์เน็ตใหม่

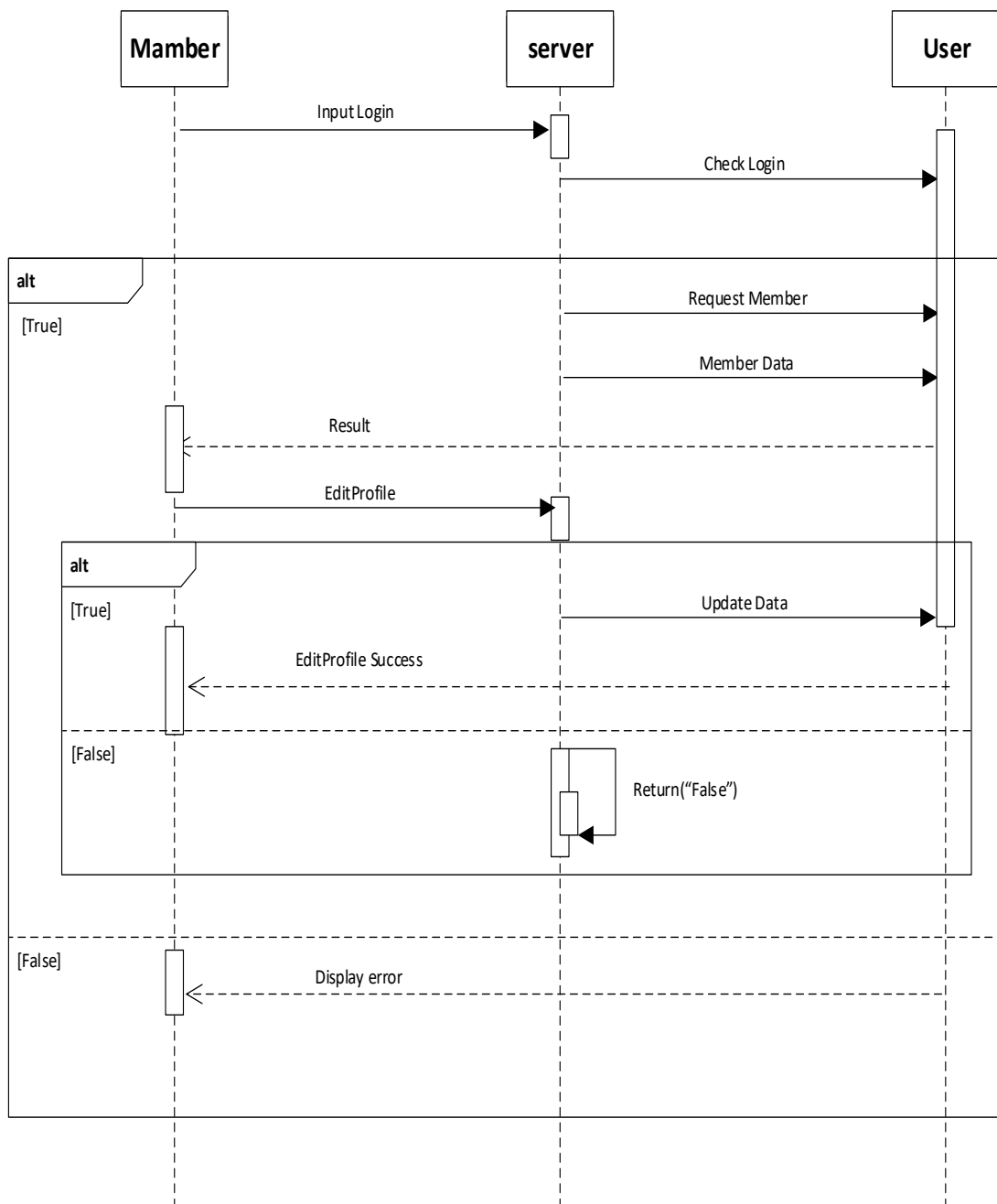
2.5.3 Sequence Diagram for Manage



รูปที่ 2.22 Sequence Diagram for Manage

คำอธิบาย กระบวนการทำงานของ Sequence Diagram for Manage

- (1) Manage หมายถึงการจัดการของแอดมิน (Admin)
- (2) Server หมายถึงการเชื่อมต่อกับฐานข้อมูล (firebase)
- (3) Connect Internet หมายถึงทำการเชื่อมต่ออินเทอร์เน็ต
- (4) Input Login หมายถึงกรอกข้อมูลเพื่อเข้าใช้งานของแอดมิน (Admin) ประกอบไปด้วย อีเมล (Email), รหัสผ่าน (Password) จึงจะเข้าหน้าแอดมิน (Admin) ได้
- (5) Check not null หมายถึงตรวจสอบว่ามีค่าที่ไม่ได้ทำการกรอกหรือไม่
- (6) Request member หมายถึงการร้องขอสิทธิการทำงานของแอดมิน (Admin)
- (7) Provide Data หมายถึงข้อมูลที่อยู่ใน firebase เอาออกมาให้ฝั่งผู้ใช้งาน (User)
- (8) Set Permission หมายถึงกำหนดค่าการเข้าถึงของผู้ใช้งาน (User) ในระบบแอปพลิเคชัน
ติดตามสัตว์เลี้ยงแอดมิน (Admin) สามารถกำหนดสถานะของผู้ใช้งานได้ว่าเป็นแอดมิน (Admin) หรือผู้ใช้งาน (User)
- (9) Confirm Permission หมายถึงทำการยืนยันข้อมูลที่แอดมิน (Admin) ได้ทำการแก้ไข
และอนุญาตการเข้าถึงข้อมูลนั้นๆ
- (10) Manage Data หมายถึงจัดการกับข้อมูลหรือแอดมิน (Admin) ได้ทำการแก้ไขข้อมูล
สำเร็จ
- (11) Update Data หมายถึงทำการอัปเดตข้อมูลที่แอดมิน (Admin) ได้ทำการแก้ไขหรือ
จัดการกับข้อมูลต่างๆ
- (12) Manage Success หมายถึงจัดการความสำเร็จ
- (13) Verify ID หมายถึงยืนยัน ID ที่แอดมิน (Admin) ทำการจัดการข้อมูลนั้นๆ
- (14) Allow หมายถึงการอนุญาตการเข้าถึงของข้อมูลที่แอดมิน (Admin) ได้ทำการกำหนด
- (15) Confirm หมายถึงทำการยืนยันข้อมูลทั้งหมดที่แอดมิน (Admin) ทำการจัดการอีกครั้ง
- (16) Display error หมายถึงทำการร้องขอใหม่อีกครั้ง
- (17) onConnect หมายถึงทำการเชื่อมต่อกรณีเป็น False ก็ทำการเชื่อมต่ออินเทอร์เน็ต
ใหม่

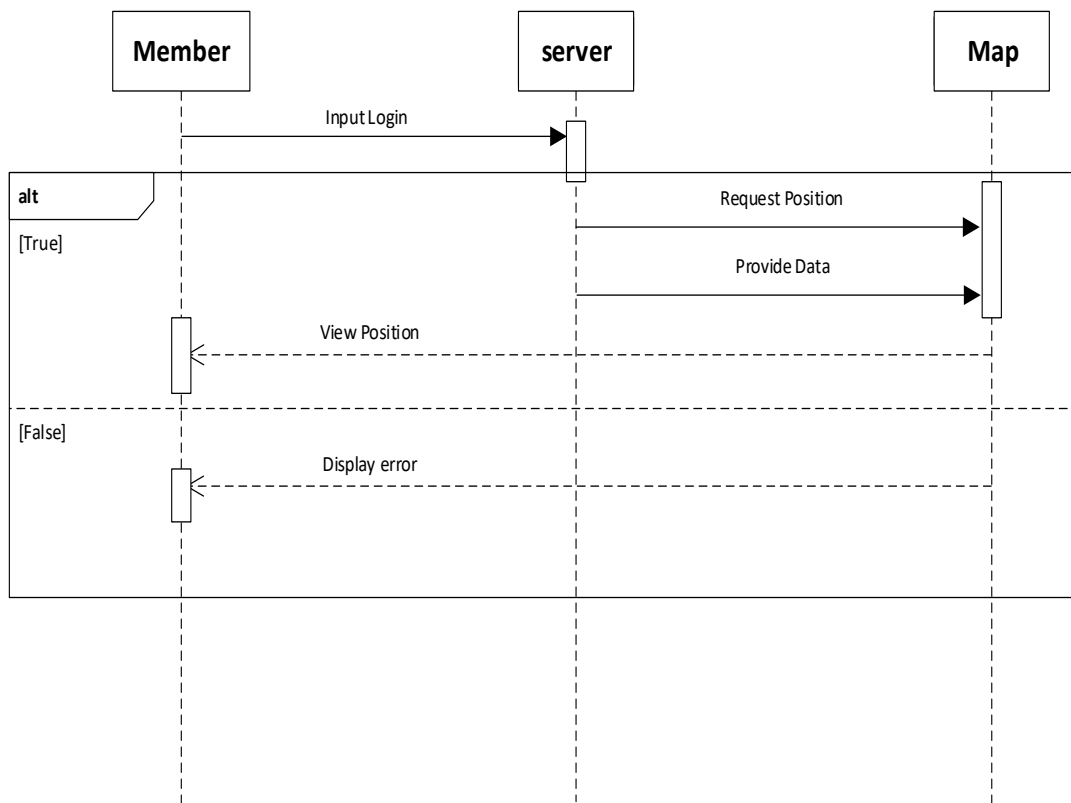


รูปที่ 2.23 Sequence Diagram for Edit profile

คำอธิบายกระบวนการทำงานของ Sequence Diagram for Edit profile

- (1) Server หมายถึงการเชื่อมต่อกับฐานข้อมูล (firebase)
- (2) User หมายถึงผู้ใช้งานในระบบ
- (3) Input Login หมายถึงกรอกข้อมูลผู้ใช้งาน ประกอบไปด้วย อีเมล (Email), รหัสผ่านของผู้ใช้งาน (Password)
- (4) Check Login หมายถึงตรวจสอบการ Login
- (5) Request Member หมายถึงขอข้อมูลสมาชิก
- (6) Member Data หมายถึงจัดการกับข้อมูล
- (7) Result หมายถึงข้อมูลที่ร้องขอจาก firebase
- (8) Edit Profile หมายถึงทำการแก้ไขโปรไฟล์และข้อมูลของผู้ใช้งาน
- (9) Update Data หมายถึงทำการอัปเดตข้อมูล
- (10) Edit Profile Success หมายถึงทำการแก้ไขโปรไฟล์หรือข้อมูลผู้ใช้งานเรียบร้อยแล้ว
- (11) Return ("False") หมายถึงทำการแก้ไขโปรไฟล์ใหม่อีกครั้ง
- (12) Display error หมายถึงทำการร้องขอใหม่อีกครั้ง

2.5.5 Sequence Diagram for View position

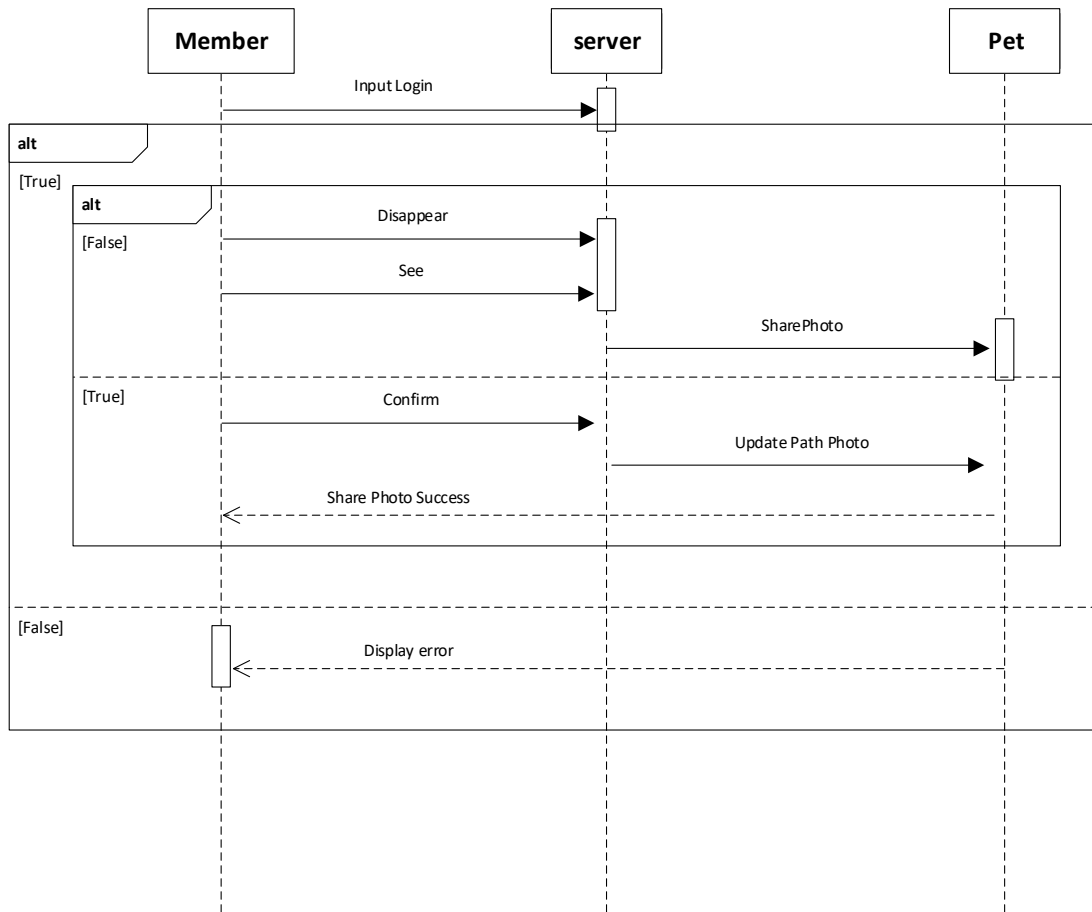


รูปที่ 2.24 Sequence Diagram for View position

คำอธิบาย กระบวนการทำงานของ Sequence Diagram for View position

- (1) Server หมายถึงการเชื่อมต่อกับฐานข้อมูล (firebase)
- (2) Map หมายถึงตำแหน่งที่สัตว์เลี้ยงอยู่
- (3) Input Login หมายถึงกรอกข้อมูลผู้ใช้
- (4) Request Position หมายถึงร้องขอตำแหน่งสัตว์เลี้ยง จาก server
- (5) Provide Data หมายถึงข้อมูลที่อยู่ใน database เอาออกมาให้ฝั่ง user
- (6) View Position หมายถึงแสดงที่อยู่ของสัตว์เลี้ยง
- (7) Display error หมายถึงทำการร้องขอใหม่อีกครั้ง

2.5.6 Activity Diagram for Share photo



รูปที่ 2.23 Sequence Diagram for Share photo

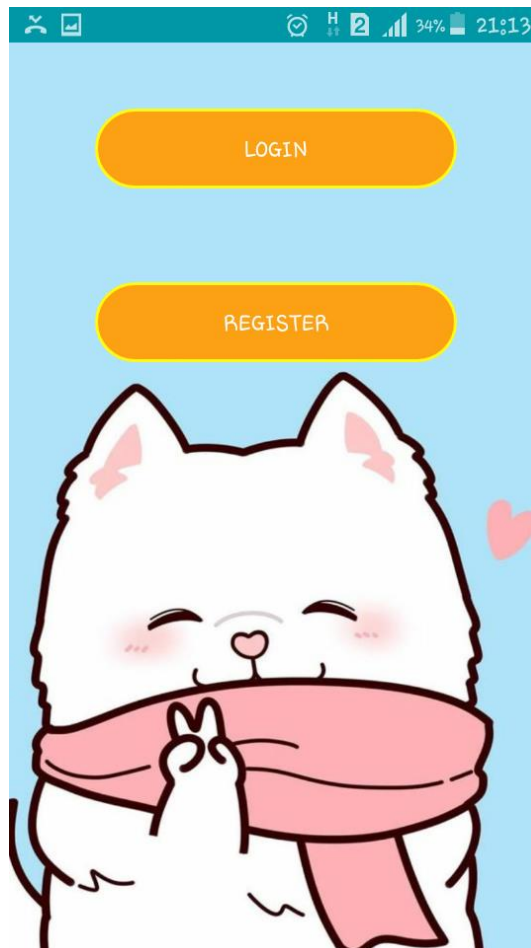
คำอธิบาย กระบวนการทำงานของ Sequence Diagram for Share photo

- (1) Server หมายถึงการเชื่อมต่อกับฐานข้อมูล (firebase)
- (2) Pet หมายถึงสัตว์เลี้ยง
- (3) Input Login หมายถึงทำการกรอกข้อมูลผู้ใช้
- (4) Disappear หมายถึงประเภทสัตว์เลี้ยงที่หาย
- (5) See หมายถึงสัตว์เลี้ยงที่พบ
- (6) Share Photo หมายถึงทำการแชร์ภาพ
- (7) Confirm หมายถึงยืนยัน
- (8) Update Path Photo หมายถึงทำการอัปเดตรูปภาพใหม่
- (9) Share Photo Success หมายถึงทำการแชร์ภาพสำเร็จ
- (10) Display error หมายถึงทำการร้องขอใหม่อีกครั้ง

บทที่ 3

การออกแบบและการทดสอบระบบ

3.1 หน้าจอหลักบนแอปพลิเคชัน



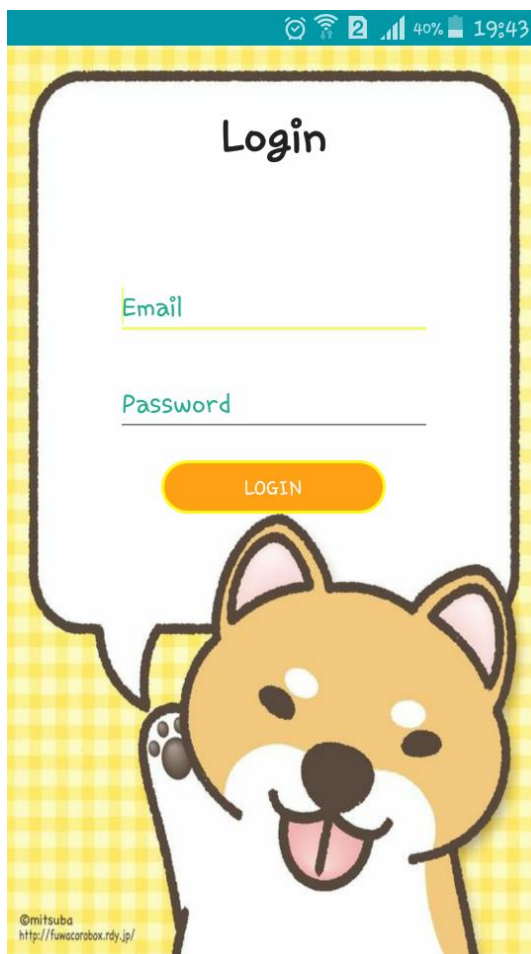
รูปที่ 3.1 หน้าจอหลักบนแอปพลิเคชัน

จากรูปที่ 3.1 เมื่อดาวน์โหลดแอปพลิเคชันมาสามารถทำการกดปุ่มได้ 2 ปุ่ม

ปุ่มที่ 1. Login สามารถเข้าใช้งานได้เลยหากมี ID User ในระบบ

ปุ่มที่ 2. Register การเข้าใช้งานต้องทำการสมัครเป็นสมาชิก (User) เพื่อให้สามารถเข้าใช้งานได้ถ้าหากยังไม่ทำการสมัครเป็นผู้ใช้งานแอปพลิเคชันสามารถกดที่ปุ่ม Register คือปุ่มที่ให้ผู้ใช้งานทำการสมัครสมาชิก (User) เพื่อที่จะสามารถเข้าใช้งานในระบบได้

3.2 หน้าจอ Login เข้าสู่ระบบ



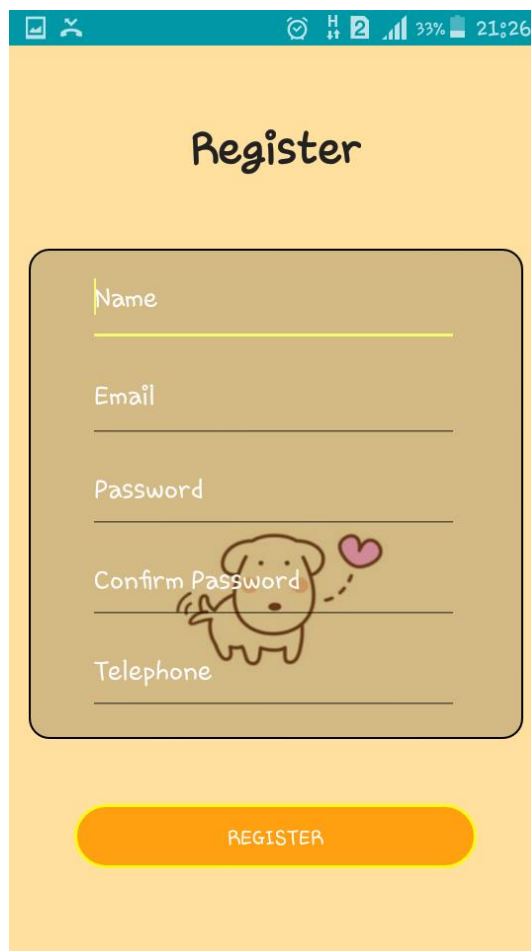
รูปที่ 3.2 หน้าจอ Login เข้าสู่ระบบ

จากรูปที่ 3.2 หน้าจอ Login เข้าสู่ระบบในส่วนนี้ผู้ใช้งานจะต้องทำการกรอกประกอบไปด้วย

1. อีเมล (Email) ที่ผู้ใช้งาน (User) ได้ทำการสมัครเพื่อเข้าใช้งาน
2. รหัสผ่าน (Password) ที่ผู้ใช้งาน (User) ได้ทำการสมัครเข้าใช้งาน

จากนั้นทำการกด Login เพื่อเข้าสู่ระบบการใช้งานหน้าต่อไป หากการเข้าสู่ระบบของผู้ใช้งาน (User) เกิดข้อผิดพลาดหรือไม่มีการเชื่อมต่ออินเทอร์เน็ต ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ได้ทราบว่าการเข้าสู่ระบบไม่สำเร็จ ระบบจะให้ทำการกรอกข้อมูลเพื่อเข้าสู่ระบบใหม่อีกครั้งหรือทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง เพื่อทำการเข้าไปยังหน้าผู้ใช้งาน (User) หลักของระบบแอปพลิเคชันติดตามสัตว์เลี้ยง

3.3 หน้าจอสมัครสมาชิก (Register)

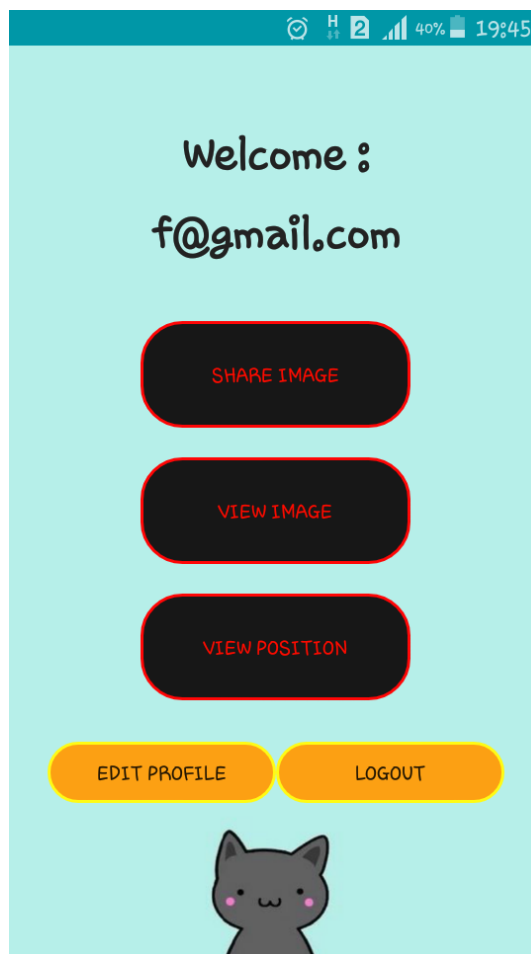


รูปที่ 3.3 หน้าจอสมัครสมาชิก (Register)

จากรูปที่ 3.3 หน้าจอสมัครสมาชิก (Register) ในส่วนนี้ผู้ใช้งาน (User) จะต้องทำการกรอกข้อมูลพื้นฐานเพื่อทำการสมัครสมาชิก (Register) โดยประกอบไปด้วย

1. ชื่อของผู้ใช้งาน (Name)
2. อีเมลของผู้ใช้งาน (Email)
3. รหัสผ่านของผู้ใช้งาน (Password)
4. ทำการยืนยันรหัสผ่านอีกครั้ง (Confirm Password) เพื่อทำการเช็คว่ารหัสผ่านที่กรอกมานั้นถูกต้อง
5. เบอร์โทรศัพท์ (Telephone) ใส่เบอร์โทรศัพท์ของผู้ใช้งานเพื่อเก็บเป็นข้อมูลพื้นฐาน
6. Register กดปุ่มนี้เพื่อยืนยันการกรอกข้อมูลที่สมัครเข้าสู่ระบบผู้ใช้งาน (User) หากกรอกข้อมูลของผู้ใช้ผิดพลาดหรือไม่มีการเชื่อมต่ออินเทอร์เน็ต ระบบจะทำการแจ้งเตือนให้ทราบว่า การสมัครเข้าสู่ผู้ใช้งาน (User) ระบบไม่สำเร็จ ให้ทำการกรอกข้อมูลสมัครใหม่อีกครั้ง

3.4 หน้าจอหลักของแอปพลิเคชัน



รูปที่ 3.4 หน้าจอหลักของแอปพลิเคชัน

จากรูปที่ 3.4 หน้าจอหลักของแอปพลิเคชัน คือหน้าที่ผู้ใช้งาน (User) แอปพลิเคชันสามารถเข้ามาใช้งานได้เมื่อผู้ใช้งาน (User) ทำการล็อกอิน (Login) เข้าสู่ระบบเรียบร้อยแล้วระบบก็จะทำการตรวจสอบว่าเป็นผู้ใช้งาน (User) จริงหรือไม่ หากบุคคลที่ทำการล็อกอิน (Login) เข้าสู่ระบบไม่ใช่ผู้ใช้งาน (User) แอปพลิเคชันจะไม่สามารถเข้ามาในหน้านี้ได้ ในหน้าจะมีปุ่มให้เลือก 5 ปุ่มประกอบไปด้วย

ปุ่มที่ 1 SHARE IMAGE คือการแชร์ภาพ สัตว์เลี้ยงที่หายและสัตว์เลี้ยงที่พบ

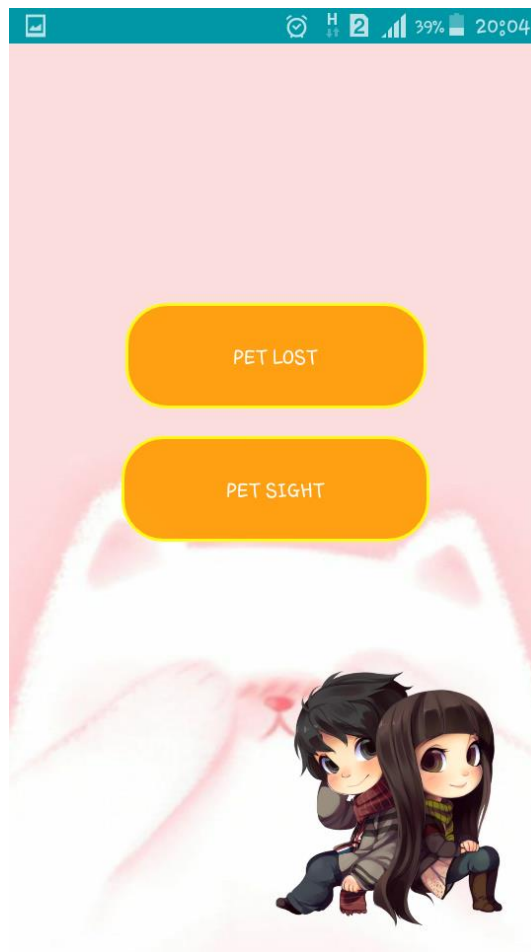
ปุ่มที่ 2 VIEW IMAGE คือการเรียกดูภาพและข้อมูลของผู้ที่ทำการแชร์ภาพ สัตว์เลี้ยงที่หายและสัตว์เลี้ยงที่พบ

ปุ่มที่ 3 VIEW POSITION คือ การเรียกดูตำแหน่งที่อยู่ของสัตว์เลี้ยง

ปุ่มที่ 4 EDIT PROFILE ทำการแก้ไขข้อมูลของผู้ใช้งาน (User)

ปุ่มที่ 5 LOGOUT ทำการออกจากระบบผู้ใช้งานแอปพลิเคชัน

3.5 หน้าจอให้เลือกแชร์ภาพสัตว์เลี้ยงหายหรือพบเจอ



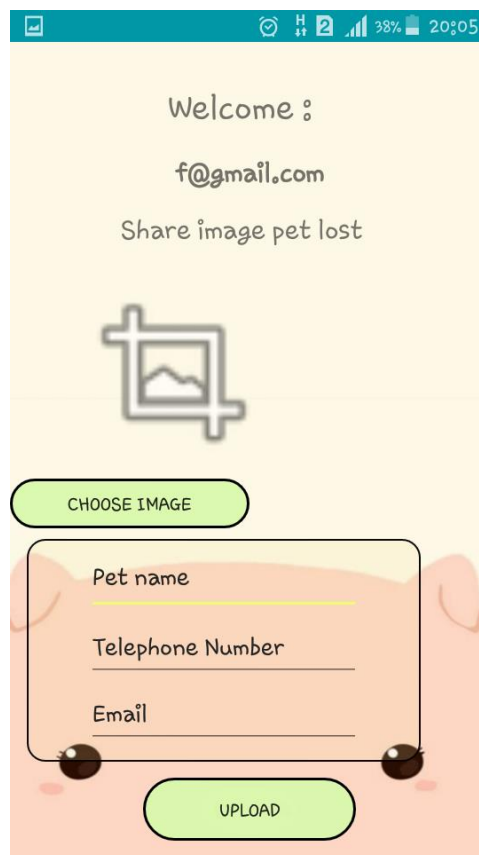
รูปที่ 3.5 หน้าจอการแชร์ภาพสัตว์เลี้ยงที่ไม่พบหรือหายออกไปเป็นเวลานาน
และการแชร์ภาพสัตว์เลี้ยงพบจากผู้ใช้งานแอปพลิเคชันร่วมกัน

จากรูปที่ 3.5 หน้าจอการแชร์ภาพสัตว์เลี้ยงที่ไม่พบหรือหายออกไปเป็นเวลานาน และการแชร์ภาพสัตว์เลี้ยงพบจากผู้ใช้งาน (User) แอปพลิเคชันร่วมกัน คือหน้าที่ผู้ใช้งานแอปพลิเคชัน สามารถแชร์รูปภาพสัตว์เลี้ยงที่ไม่หาไม่พบหรือแชร์ภาพสัตว์เลี้ยงที่พบได้เป็นการแชร์ภาพจาก ผู้ใช้งานร่วมกัน ประกอบไปด้วย 2 ปุ่ม

ปุ่มที่ 1. PET LOST แชร์ภาพสัตว์เลี้ยงที่ไม่พบบริเวณบ้านหรือหายออกไปเป็นเวลานาน เจ้าของจึงต้องการขอความร่วมมือหรือความช่วยเหลือจากผู้ใช้งาน (User) แอปพลิเคชันร่วมกัน จึง ทำการแชร์ภาพสัตว์เลี้ยงที่ตนรักลงบนแอปพลิเคชัน

ปุ่มที่ 2. PET SIGHTX เป็นการแชร์ภาพระหว่างผู้ใช้แอปพลิเคชัน เพื่อให้ผู้ที่เป็นเจ้าของของ สัตว์เลี้ยงทำการติดต่อกลับมายังผู้พบเจอสัตว์เลี้ยงที่ตนรักและต้องการให้มันกลับมา

3.6 หน้าจอการแชร์ภาพสัตว์เลี้ยงไม่พบบริเวณบ้านและหายออกจากบ้านเป็นเวลานาน

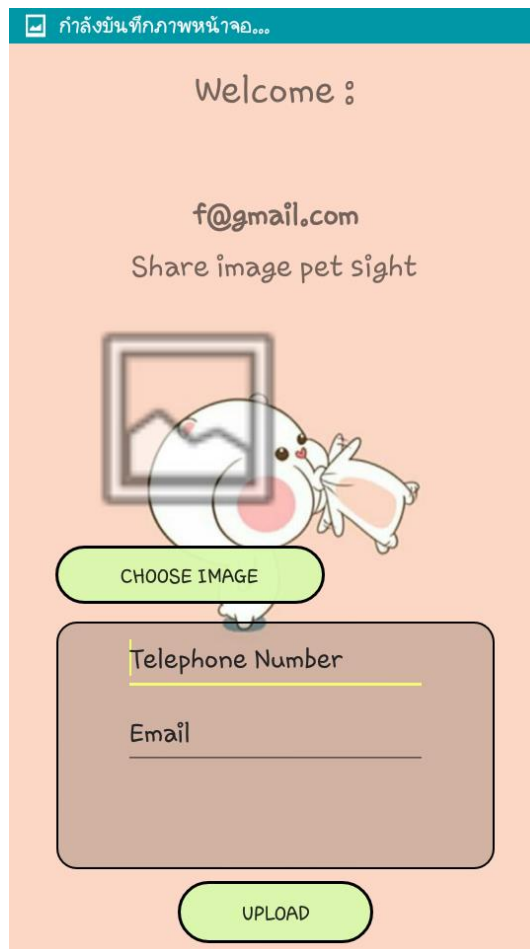


รูปที่ 3.6 หน้าจอการแชร์ภาพสัตว์เลี้ยงไม่พบบริเวณบ้านและหายออกจากบ้านเป็นเวลานาน

จากรูปที่ 3.6 หน้าจอการแชร์ภาพสัตว์เลี้ยงไม่พบบริเวณบ้านและหายออกจากบ้านเป็นเวลานาน คือ หน้าให้ผู้ใช้งานแอปพลิเคชัน สามารถทำการแชร์ภาพสัตว์เลี้ยงที่หายออกจากบ้านและต้องการตามหา โดยการกรอกข้อมูลประกอบไปด้วย

1. ปุ่ม CHOOSE IMAGE คือการเลือกรูปภาพสัตว์เลี้ยงที่ต้องการตามหาจากมือถือเพื่อทำการแชร์ภาพลงบนแอปพลิเคชันติดตามสัตว์เลี้ยง
2. Pet name ชื่อของสัตว์เลี้ยงโดยส่วนใหญ่สัตว์เลี้ยงมักจำชื่อของตัวเองได้ หากมีผู้พบเห็นอาจเป็นข้อมูลที่สำคัญเพื่อช่วยในการตามหาสัตว์เลี้ยงกลับคืนมา
3. Telephone Number ทำการกรอกเบอร์โทรศัพท์ของเจ้าของสัตว์เลี้ยงเพื่อรอการติดต่อกลับจากผู้พบเจอสัตว์เลี้ยงหรือติดต่อเพื่อให้ข้อมูลเพิ่มเติม เช่น ลักษณะของสัตว์เลี้ยง
4. Email กรอกอีเมล (Email) ผู้ใช้งาน (User) เพื่อรอการติดต่อกลับจากผู้พบเจอสัตว์เลี้ยงหรือติดต่อเพื่อให้ข้อมูลเพิ่มเติมเช่น ลักษณะของสัตว์เลี้ยง
5. ปุ่ม UPLOAD ทำการ update ข้อมูลที่กรอกมาทั้งหมดนี้ลงบนแอปพลิเคชันติดตามสัตว์เลี้ยง

3.7 หน้าจอการแชร์ภาพสัตว์เลี้ยงพบ

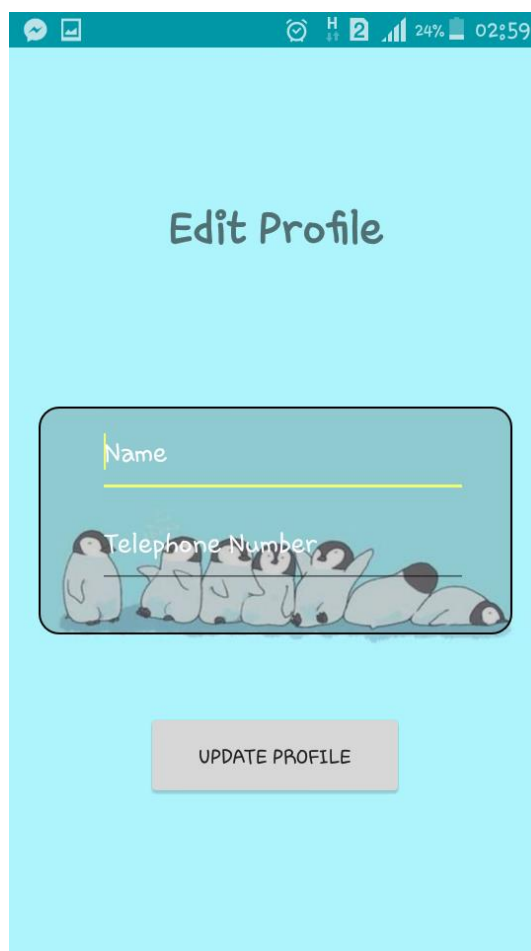


รูปที่ 3.7 หน้าจอการกรอกข้อมูลสัตว์เลี้ยงเพื่อทำแชร์ภาพสัตว์เลี้ยงที่พบ

จากรูปที่ 3.7 หน้าจอการกรอกข้อมูลสัตว์เลี้ยงเพื่อทำแชร์ภาพสัตว์เลี้ยงที่พบ คือหน้าที่ผู้ใช้งาน (User) สามารถทำการแชร์ภาพสัตว์เลี้ยงที่พบเจอตามสถานที่ต่างๆ และเป็นผู้ใช้งาน (User) แอปพลิเคชันร่วมกันเท่านั้นถึงจะเห็นภาพที่แชร์ร่วมกันได้ การกรอกข้อมูลประกอบไปด้วย

1. ปุ่ม CHOOSE IMAGE คือการเลือกรูปภาพสัตว์เลี้ยงที่ต้องการตามหาจากมือถือเพื่อทำการแชร์ภาพลงบนแอปพลิเคชันติดตามสัตว์เลี้ยง
2. Telephone Number ทำการกรอกเบอร์โทรศัพท์ของเจ้าของสัตว์เลี้ยงเพื่อรอการติดต่อกลับจากผู้พบเจอสัตว์เลี้ยงหรือติดต่อเพื่อให้ข้อมูลเพิ่มเติม เช่น ลักษณะของสัตว์เลี้ยง
3. Email กรอกอีเมล (Email) ผู้ใช้งาน (User) เพื่อรอการติดต่อกลับจากผู้พบเจอสัตว์เลี้ยงหรือติดต่อเพื่อให้ข้อมูลเพิ่มเติมเช่น ลักษณะของสัตว์เลี้ยง
4. ปุ่ม UPLOAD ทำการ update ข้อมูลที่กรอกมาทั้งหมดนี้ลงบนแอปพลิเคชันติดตามสัตว์เลี้ยง

3.8 หน้าจอการแก้ไขข้อมูล

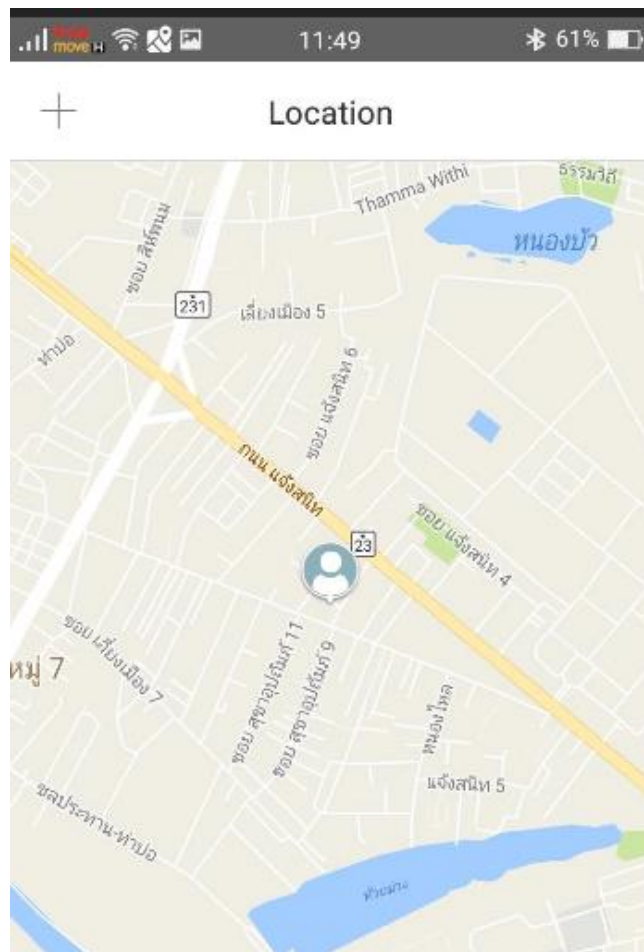


รูปที่ 3.8 หน้าจอการแสดงผลการแก้ไขข้อมูล

จากรูปที่ 3.8 หน้าจอการแสดงผลการแก้ไขข้อมูล คือหน้าที่ผู้ใช้งาน (User) แอปพลิเคชันสามารถทำแก้ไขข้อมูลของผู้ใช้งาน (User) โดยการกดที่ปุ่ม Edit Profile จึงจะสามารถทำการแก้ไขข้อมูลของผู้ใช้งาน (User) และทำการแก้ไขข้อมูลโดยระบบจะโชว์ข้อมูลเก่าที่ผู้ใช้งาน (User) ได้ทำการสมัครหรือได้ทำการกรอกไว้เบื้องต้นเพื่อให้ผู้ใช้งาน (User) ได้ทำการแก้ไข ประกอบไปด้วย

1. Name สามารถทำการแก้ไขชื่อของผู้ใช้งาน (User) ได้
2. Telephone Number ทำการแก้ไขเบอร์โทรศัพท์ของผู้ใช้งาน (User) ในกรณีผู้ใช้งาน (User) ทำการเปลี่ยนเบอร์โทรศัพท์
3. ปุ่ม UPLOAD ทำการ update ข้อมูลที่ผู้ใช้งานได้ทำการแก้ไขมาทั้งหมดนี้ลงในแอปพลิเคชันติดตามสัตว์เลี้ยง

3.9 แสดงตำแหน่งของสัตว์เลี้ยง

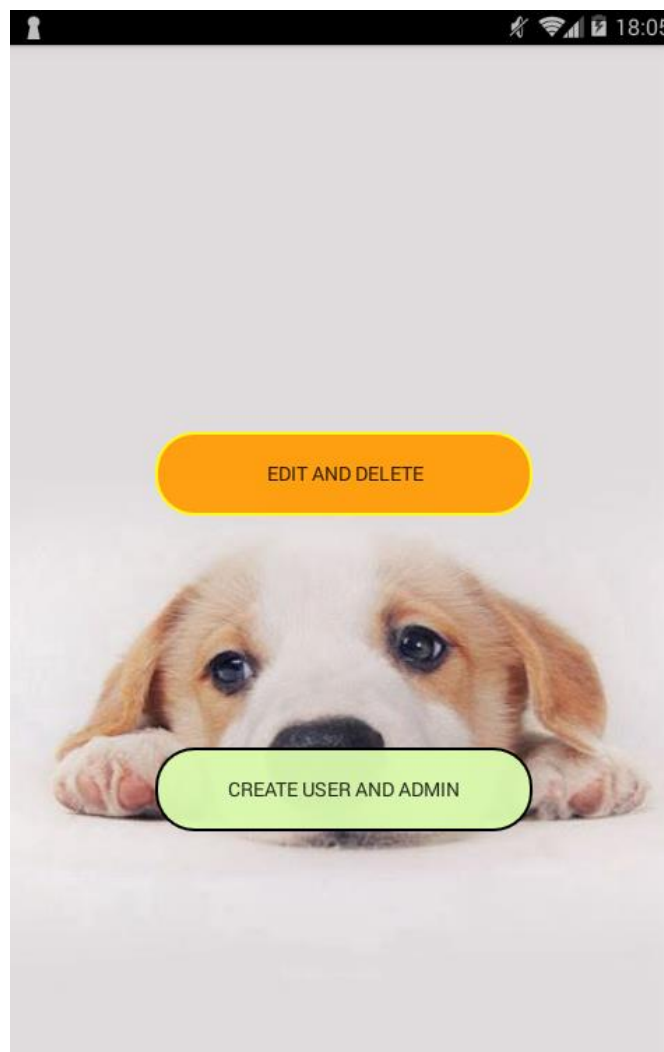


รูปที่ 3.9 หน้าจอการแสดงผลตำแหน่งของสัตว์เลี้ยง

จากรูปที่ 3.9 หน้าจอการแสดงผลตำแหน่งของสัตว์เลี้ยง คือหน้าที่ผู้ใช้งานแอปพลิเคชันสามารถทำการกดปุ่มที่ VIEW POSITION เพื่อทำการเรียกดูแผนที่ และตำแหน่งของสัตว์เลี้ยงจาก sensor โดย sensor จะทำการส่งตำแหน่งที่อยู่ของสัตว์เลี้ยงในปัจจุบันไปเก็บไว้ที่ firebase และ firebase จะทำการส่งค่าข้อมูล ประกอบไปด้วย

1. ละติจูด (Latitude) “เป็นพิกัดที่ใช้บอกตำแหน่งบนพื้นโลกและแบ่งเขตสภาวะอากาศโดยวัดจากเส้นศูนย์สูตร”
2. ลองจิจูด (Longitude) “เป็นพิกัดที่ใช้บอกตำแหน่งบนพื้นโลก โดยวัดไปทางตะวันออกหรือตะวันตกจากเส้นสมมติในแนวเหนือ-ใต้ ที่เรียกว่าเส้นเมริเดียนแรก”
3. วัน (Date) และเวลา (Time) ปัจจุบันไปเก็บไว้ที่ firebase

3.10 หน้าของผู้ดูแลระบบ



รูปที่ 3.10 หน้าของผู้ดูแลระบบ

จากรูปที่ 3.10 หน้าของผู้ดูแลระบบ Admin สามารถเลือกได้ 2 ปุ่ม

1. ปุ่ม EDIT AND DELETE คือปุ่มแก้ไขและลบข้อมูลของผู้ใช้งาน (User) คือหน้าที่ผู้ดูแลระบบสามารถจัดการกับข้อมูลของผู้ดูแลระบบและสามารถจัดการกับข้อมูลของผู้ใช้งาน (User) และยังสามารถลบข้อมูลของผู้ใช้งาน (User) ออกจากระบบได้
2. ปุ่ม Create User and Admin คือหน้าที่ผู้ดูแลระบบสามารถจัดการกับข้อมูลของผู้ดูแลระบบและสามารถจัดการกับข้อมูลของผู้ใช้งาน (User) ได้อีกด้วย ผู้ดูแลระบบยังกำหนดสิทธิผู้ใช้งานโดยการกำหนดสถานะผู้ใช้งานเพื่อเข้าถึงข้อมูลนั้นๆ

3.11 หน้าจอการแสดงผลการแก้ไขข้อมูลของ Admin

The screenshot shows an Android application interface. At the top, there's a status bar with a lock icon, signal strength, Wi-Fi, and the time 16:52. The main content area has a dark grey background. It displays two sets of user information. The first set is for 'ngz@hotmail.com' with Name: nueng, Telephone: 087778897, and Status: 1. The second set is for 'admin@hotmail.com' with Name: admin, Telephone: 087799782, and Status: 0. A white modal dialog box is overlaid on the screen, titled 'Updating ngz@hotmail.com'. It contains four input fields: 'nueng' (with a yellow underline), '087778897', '1', and an empty field. At the bottom of the dialog are two buttons: 'UPDATE' and 'DELETE'.

รูปที่ 3.11 หน้าจอการแสดงผลการแก้ไขข้อมูลของ Admin

จากรูปที่ 3.11 หน้าจอการแสดงผลการแก้ไขข้อมูลของ Admin คือหน้าที่ผู้ดูแลระบบสามารถจัดการกับข้อมูลของผู้ดูแลระบบ (Admin) สามารถ เพิ่ม ลบ แก้ไข และสามารถจัดการกับข้อมูลของผู้ใช้งาน (User) ได้อีกด้วย เช่น เพิ่ม ลบ แก้ไขได้อีกด้วยการแก้ไขประกอบไปด้วย

1. ชื่อของผู้ใช้งาน (User) และชื่อของผู้ดูแลระบบ (Admin)
2. เบอร์โทรศัพท์ของผู้ใช้งาน (User) และชื่อของผู้ดูแลระบบ (Admin)
3. ผู้ดูแลระบบสามารถกำหนดสิทธิหรือสถานะของผู้ใช้งาน (User)

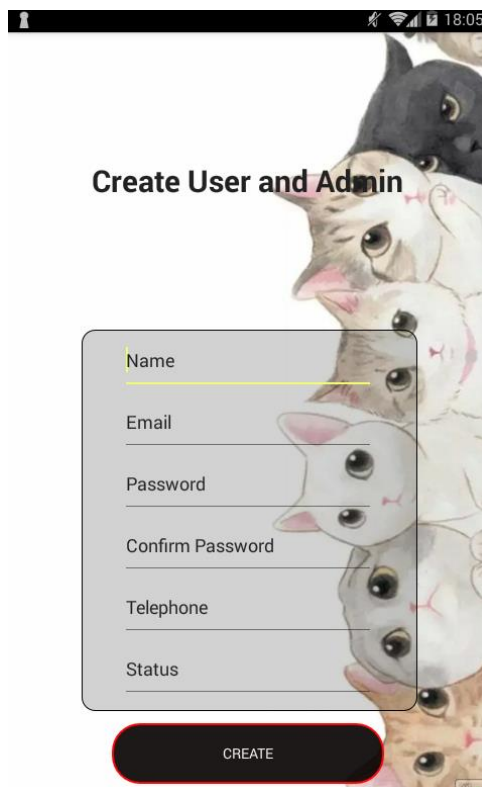
3.1 Status = 0 คือ สถานะเป็นผู้ดูแลระบบ (Admin)

3.2 Status = 1 คือ สถานะเป็นผู้ใช้งาน (User)

4. ปุ่ม UPLOAD ทำการ update ข้อมูลที่กรอกมาทั้งหมดนี้ลงบนแอปพลิเคชันติดตามสัตว์เลี้ยง

5. ปุ่ม DELETE ทำการลบข้อมูลของผู้ใช้งานในระบบ (User) และทำการลบข้อมูลของผู้ดูแลระบบ (Admin)

3.12 หน้า Create User and Admin



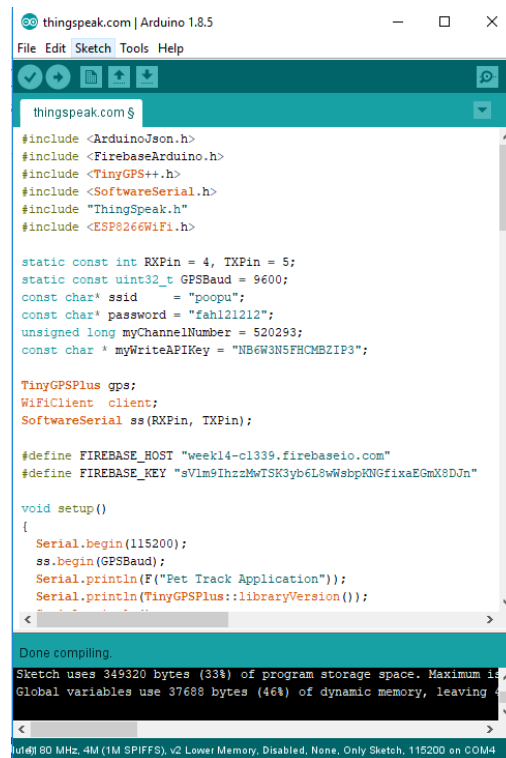
The screenshot shows a mobile application interface for creating a user or admin. The title 'Create User and Admin' is centered at the top. Below the title is a form with the following fields: Name, Email, Password, Confirm Password, Telephone, and Status. Each field has a corresponding input line. At the bottom of the form is a red button with the text 'CREATE'. The background of the form is a collage of various cute cat faces.

รูปที่ 3.12 หน้า Create User and Admin

จากรูปที่ 3.12 หน้า Create User and Admin คือหน้าที่ผู้ดูแลระบบสามารถจัดการกับข้อมูลของผู้ดูแลระบบ (Admin) และสามารถจัดการกับข้อมูลของผู้ใช้งาน (User) การ Create User and Admin ประกอบไปด้วย

1. Name ใส่ชื่อของผู้ใช้งาน (User) หรือชื่อที่ผู้ดูแลระบบต้องการเพิ่ม (Admin)
2. Email ใส่อีเมล (Email) ของผู้ใช้งานหรือใส่อีเมล (Email) ของผู้ดูแลระบบ (Admin)
3. Password ผู้ดูแลระบบ (Admin) ทำการตั้งรหัสผ่านให้กับผู้ใช้งาน (User) หรือตั้งรหัสผ่านให้กับผู้ดูแลระบบ (Admin)
4. Confirm Password ผู้ดูแลระบบ (Admin) ทำการยืนยันรหัสผ่านให้กับผู้ใช้งาน (User) อีกครั้ง หรือทำการยืนยันรหัสผ่านให้กับผู้ดูแลระบบ (Admin) อีกครั้ง
5. Telephone ผู้ดูแลระบบ (Admin) ทำการใส่เบอร์โทรศัพท์ของผู้ใช้งาน (User) หรือทำการใส่เบอร์โทรศัพท์ของผู้ดูแลระบบ (Admin)
6. Status ผู้ดูแลระบบสามารถกำหนดสิทธิหรือสถานะของผู้ใช้งาน (User)
 - 6.1 Status = 0 คือ สถานะเป็นผู้ดูแลระบบ (Admin)
 - 6.2 Status = 1 คือ สถานะเป็นผู้ใช้งาน (User)

3.13 ผลการทดลอง



```
thingspeak.com | Arduino 1.8.5
File Edit Sketch Tools Help

thingspeak.com $

#include <ArduinoJson.h>
#include <FirebaseArduino.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>

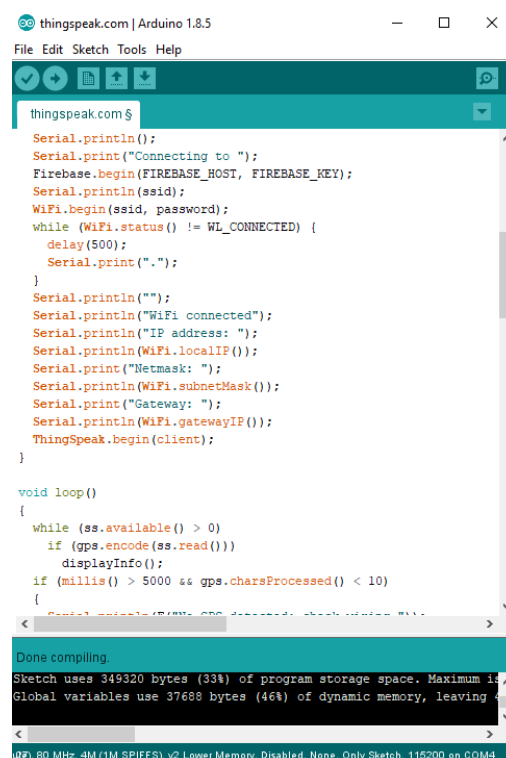
static const int RXPin = 4, TXPin = 5;
static const uint32_t GPSBaud = 9600;
const char* ssid = "poopu";
const char* password = "fah121212";
unsigned long myChannelNumber = 520293;
const char * myWriteAPIKey = "NB6W3N5FHCMBZIP3";

TinyGPSPlus gps;
WiFiClient client;
SoftwareSerial ss(RXPin, TXPin);

#define FIREBASE_HOST "week14-c1339.firebaseio.com"
#define FIREBASE_KEY "sVlm9Ihz2MwTSK3yb6L8wSbtpKNGfixaEGmX8DJn"

void setup()
{
  Serial.begin(115200);
  ss.begin(GPSBaud);
  Serial.println(F("Pet Track Application"));
  Serial.println(TinyGPSPlus::libraryVersion());
}

Done compiling.
Sketch uses 349320 bytes (33%) of program storage space. Maximum is 1024K.
Global variables use 37688 bytes (46%) of dynamic memory, leaving 42212 bytes free.
ESP8266 Pin 1: 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4
```



```
thingspeak.com | Arduino 1.8.5
File Edit Sketch Tools Help

thingspeak.com $

Serial.println();
Serial.print("Connecting to ");
Firebase.begin(FIREBASE_HOST, FIREBASE_KEY);
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
Serial.print("Netmask: ");
Serial.println(WiFi.subnetMask());
Serial.print("Gateway: ");
Serial.println(WiFi.gatewayIP());
ThingSpeak.begin(client);
}

void loop()
{
  while (ss.available() > 0)
  {
    if (gps.encode(ss.read()))
      displayInfo();
    if (millis() > 5000 && gps.charsProcessed() < 10)
    {
      Serial.println(F("TinyGPS++: GPS data received, sketch will not send it"));
    }
  }
}

Done compiling.
Sketch uses 349320 bytes (33%) of program storage space. Maximum is 1024K.
Global variables use 37688 bytes (46%) of dynamic memory, leaving 42212 bytes free.
ESP8266 Pin 1: 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4
```

รูปที่ 3.13 Source code read GPS



```
thingspeak.com | Arduino 1.8.5
File Edit Sketch Tools Help

thingspeak.com $
Serial.println(F("No GPS detected: check wiring."));
while (true);
}
void displayInfo()
{
  const String Name = "/yyyy";
  const String Name2 = "nueng";
  const String number = "/one";
  const String iot = "Iot";
  if (gps.location.isValid())
  {
    float latitude = (gps.location.lat());
    float longitude = (gps.location.lng());

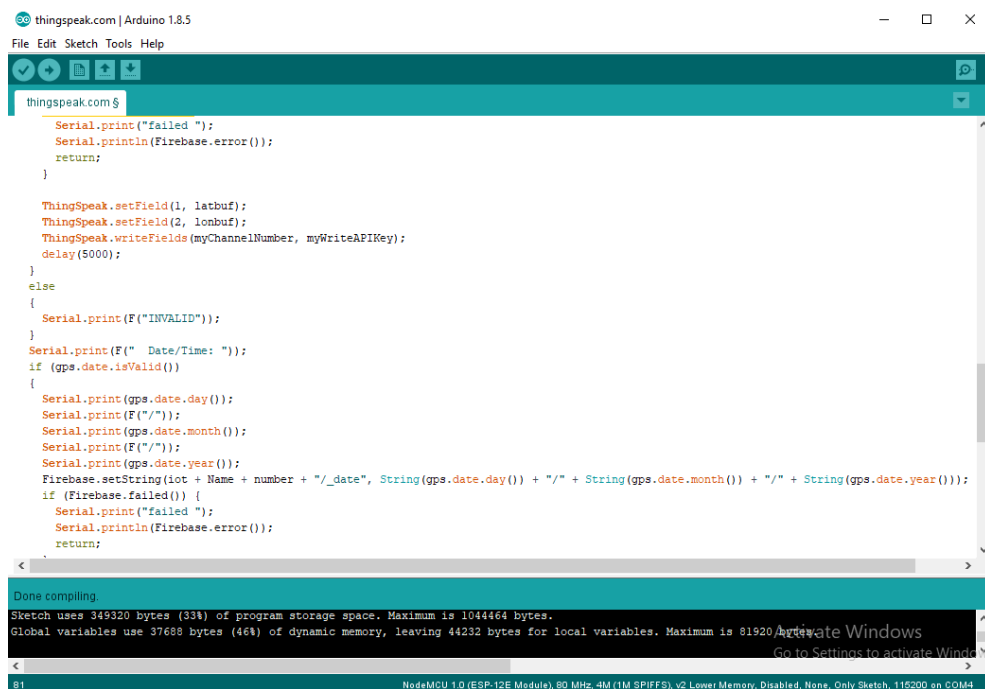
    String latbuf;
    latbuf += (String(latitude, 18));
    Serial.println(latbuf);

    String lonbuf;
    lonbuf += (String(longitude, 18));
    Serial.println(lonbuf);

    Firebase.setString(iot + Name + number + "_lat", latbuf);
    Firebase.setString(iot + Name + number + "_long", lonbuf);
    Firebase.setString(iot + Name + number + "_username", Name2);
    if (Firebase.failed()) {
      Serial.println(F("Failed "));
      Serial.println(Firebase.error());
      return;
    }

    ThingSpeak.setField(1, latbuf);
    ThingSpeak.setField(2, lonbuf);
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    delay(5000);
  }
  else
  {
    Serial.println(F("INVALID"));
  }
  Serial.print(F(" Date/Time: "));
  if (gps.date.isValid())
  {
    Serial.print(gps.date.day());
    Serial.print(F("/"));
    Serial.print(gps.date.month());
    Serial.print(F("/"));
    Serial.print(gps.date.year());
    Firebase.setString(iot + Name + number + "_date", String(gps.date.day()) + "/" + String(gps.date.month()) + "/" + String(gps.date.year()));
    if (Firebase.failed()) {
      Serial.println(F("Failed "));
      Serial.println(Firebase.error());
      return;
    }
  }
}

Done compiling.
Sketch uses 349320 bytes (33%) of program storage space. Maximum is 1044464 bytes.
Global variables use 37688 bytes (46%) of dynamic memory, leaving 44232 bytes for local variables. Maximum is 81920 bytes.
```



```
thingspeak.com | Arduino 1.8.5
File Edit Sketch Tools Help

thingspeak.com $
Serial.println(F("No GPS detected: check wiring."));
while (true);
}
void displayInfo()
{
  const String Name = "/yyyy";
  const String Name2 = "nueng";
  const String number = "/one";
  const String iot = "Iot";
  if (gps.location.isValid())
  {
    float latitude = (gps.location.lat());
    float longitude = (gps.location.lng());

    String latbuf;
    latbuf += (String(latitude, 18));
    Serial.println(latbuf);

    String lonbuf;
    lonbuf += (String(longitude, 18));
    Serial.println(lonbuf);

    Firebase.setString(iot + Name + number + "_lat", latbuf);
    Firebase.setString(iot + Name + number + "_long", lonbuf);
    Firebase.setString(iot + Name + number + "_username", Name2);
    if (Firebase.failed()) {
      Serial.println(F("Failed "));
      Serial.println(Firebase.error());
      return;
    }

    ThingSpeak.setField(1, latbuf);
    ThingSpeak.setField(2, lonbuf);
    ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
    delay(5000);
  }
  else
  {
    Serial.println(F("INVALID"));
  }
  Serial.print(F(" Date/Time: "));
  if (gps.date.isValid())
  {
    Serial.print(gps.date.day());
    Serial.print(F("/"));
    Serial.print(gps.date.month());
    Serial.print(F("/"));
    Serial.print(gps.date.year());
    Firebase.setString(iot + Name + number + "_date", String(gps.date.day()) + "/" + String(gps.date.month()) + "/" + String(gps.date.year()));
    if (Firebase.failed()) {
      Serial.println(F("Failed "));
      Serial.println(Firebase.error());
      return;
    }
  }
}

Done compiling.
Sketch uses 349320 bytes (33%) of program storage space. Maximum is 1044464 bytes.
Global variables use 37688 bytes (46%) of dynamic memory, leaving 44232 bytes for local variables. Maximum is 81920 bytes.
```

รูปที่ 3.13 Source code read GPS

```

thingspeak.com | Arduino 1.8.5
File Edit Sketch Tools Help

thingspeak.com $
// ...
return;
}
else
{
  Serial.print(F("INVALID"));
}

Serial.print(F(" "));
if (gps.time.isValid())
{
  if (gps.time.hour() < 10);
  Serial.print(gps.time.hour() + 7);
  Serial.print(F(":"));
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.print(gps.time.minute());
  Serial.print(F(":"));
  if (gps.time.second() < 10) Serial.print(F("0"));
  Serial.print(gps.time.second());
  Serial.print(F("."));
  if (gps.time.centisecond() < 10) Serial.print(F("0"));
  Serial.print(gps.time.centisecond());

  Firebase.setString(iot + Name + number + "/"_time", String(gps.time.hour() + 7) + ":" + String(gps.time.minute()) + ":" + String(gps.time.seco
  if (Firebase.failed()) {
    Serial.print("failed ");
    Serial.println(Firebase.error());
  }
}

Done compiling.
Sketch uses 349320 bytes (33%) of program storage space. Maximum is 1044464 bytes.
Global variables use 37688 bytes (46%) of dynamic memory, leaving 44232 bytes for local variables. Maximum is 61920 bytes.
NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4

```

```

thingspeak.com | Arduino 1.8.5
File Edit Sketch Tools Help

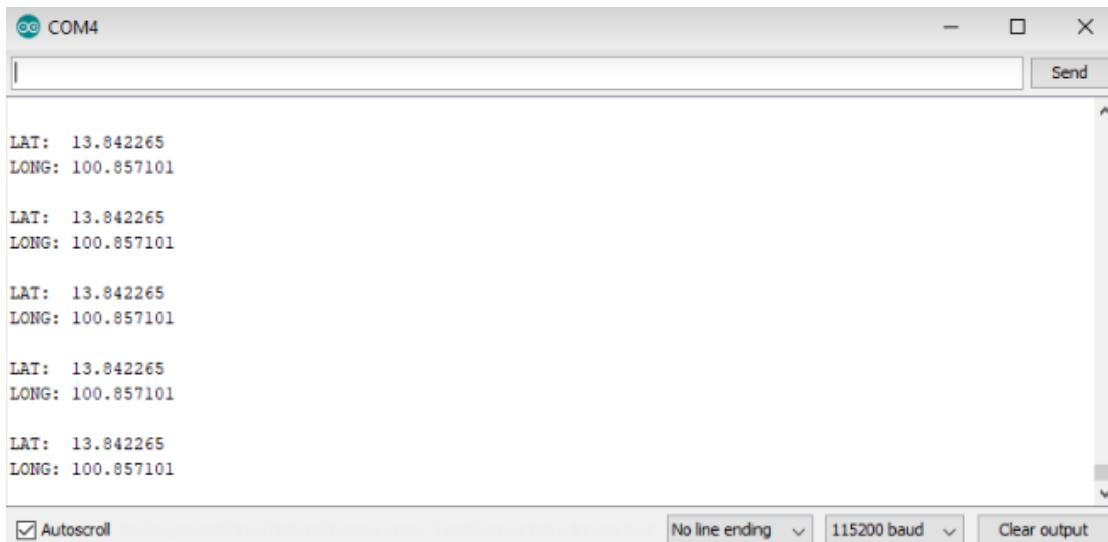
thingspeak.com $
// ...
if (gps.time.isValid())
{
  if (gps.time.hour() < 10);
  Serial.print(gps.time.hour() + 7);
  Serial.print(F(":"));
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.print(gps.time.minute());
  Serial.print(F(":"));
  if (gps.time.second() < 10) Serial.print(F("0"));
  Serial.print(gps.time.second());
  Serial.print(F("."));
  if (gps.time.centisecond() < 10) Serial.print(F("0"));
  Serial.print(gps.time.centisecond());

  Firebase.setString(iot + Name + number + "/"_time", String(gps.ti
  if (Firebase.failed()) {
    Serial.print("failed ");
    Serial.println(Firebase.error());
  }
  return;
}
else
{
  Serial.print(F("INVALID"));
}
Serial.println();
}

Done compiling.
Sketch uses 349320 bytes (33%) of program storage space. Maximum is 1044464 bytes.
Global variables use 37688 bytes (46%) of dynamic memory, leaving 44232 bytes for local variables. Maximum is 61920 bytes.
NodeMCU 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM4

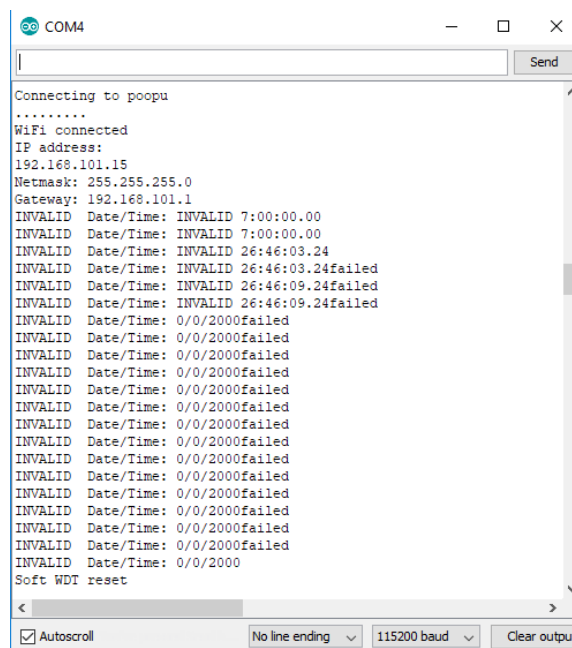
```

รูปที่ 3.13 Source code read GPS



รูปที่ 3.14 ผลลัพธ์การอ่านค่า GPS

จากรูปที่ 3.14 เป็นการแสดงผลการอ่านค่าเมื่ออยู่ในพื้นที่แจ้งมารับค่าตำแหน่งที่อยู่ของสัตว์เลี้ยงได้



รูปที่ 3.15 ผลลัพธ์การอ่านค่าเมื่ออยู่ในที่ทึบ

จากรูปที่ 3.15 เป็นการแสดงผลการอ่านค่าเมื่ออยู่ในที่ทึบจะไม่สามารถอ่านค่าที่อยู่ภายในตึกหน้าอาคารที่มีความหนาแน่น

บทที่ 4

สรุปและวิเคราะห์ผลการดำเนินงาน

4.1 สรุปผลการดำเนินงานสรุปผลการดำเนินงาน

4.1.1 เครื่องมือการใช้งานระบบและสภาพแวดล้อมที่ทดสอบระบบเครื่องมือ

4.1.1.1 สร้าง Mobile Application ใช้ Android Studio เวอร์ชัน 3.0

4.1.1.2 สร้างโปรแกรมสำหรับอ่านค่า GPS ใช้ Arduino IDE เวอร์ชัน 1.8.5

4.1.1.3 ใช้ Android SDK เวอร์ชัน 26.0.2

4.1.1.4 ใช้ Android Emulator ตัวจำลอง เวอร์ชัน 26.1.3 (7.1.1 Nougat – API 25)

4.1.1.5 ใช้มือถือ Android เวอร์ชัน 5.1.1 (มือถือรุ่น Galaxy Grand Prime)

4.1.1.6 ใช้ระบบปฏิบัติการ Windows 10

4.1.1.7 ใช้โปรแกรม Visio

4.1.1.8 ใช้ Blynk (ในการทดลองอ่านค่า GPS)

4.1.1.9 ใช้ Node MCU Wi-Fiเวอร์ชัน 1.0

4.1.1.10 ใช้ GPSModule

4.1.1.11 ใช้ Breadboard

4.1.1.12 ใช้ Power Supply 5V 2A

4.1.1.13 ใช้ Power Supply 12V 2A

4.1.2 ปัญหาในการดำเนินงาน

ตารางผลการทดสอบ แสดงความสามารถในการใช้งานแอปพลิเคชันติดตามสัตว์เลี้ยง บน API ของแอนดรอยด์ (Android) รุ่นต่างๆ โดยใช้ Android Emulator ตัวจำลองแอปโทรศัพท์บนมือถือ

ตารางที่ 4.1 การใช้งาน Android Emulator

หมายเหตุ : ✓ สามารถทำงานได้ ✕ ไม่สามารถทำงานได้

| รุ่น | ชื่อรุ่น | ระดับ API | การทำงาน |
|----------------|-------------|-----------|----------|
| Android 4.4w | KitKat Wear | API 20 | ✕ |
| Android 5.0 | Lollipop | API 21 | ✓ |
| Android 5.1 | Lollipop | API 22 | ✓ |
| Android 6.0 | Marshmallow | API 23 | ✓ |
| Android 7.0 | Nougat | API 24 | ✓ |
| Android 7.1.1 | Nougat | API 25 | ✓ |
| Android 8.0 | O | API 26 | ✕ |
| Android API 27 | API 27 | API 27 | ✕ |

จากตารางทดสอบดังตารางที่ 4.1 การใช้งาน Android Emulator นั้น Android ที่มีเวอร์ชันเก่ากว่ารุ่น Android 5.0 (Lollipop) ระดับ API 21 ไม่รองรับการทำงานแอปพลิเคชันติดตามสัตว์เลี้ยง และมาสามารถทำงานได้ในเวอร์ชันที่ใหม่กว่ารุ่น Android 7.1.1 (Nougat) ระดับ API 25

ตารางผลการทดสอบแสดงความสามารถในการใช้งานของเซ็นเซอร์ GPS Module และ Node MCU Wi-Fiว่ามีขีดจำกัดในการใช้งานของตัวอุปกรณ์หรือไม่

ตารางที่ 4.2 การทดสอบการทำงานของเซ็นเซอร์ GPS Module และ Node MCU Wi-Fi โดยทำการทดสอบการรับค่าแต่ละสถานที่

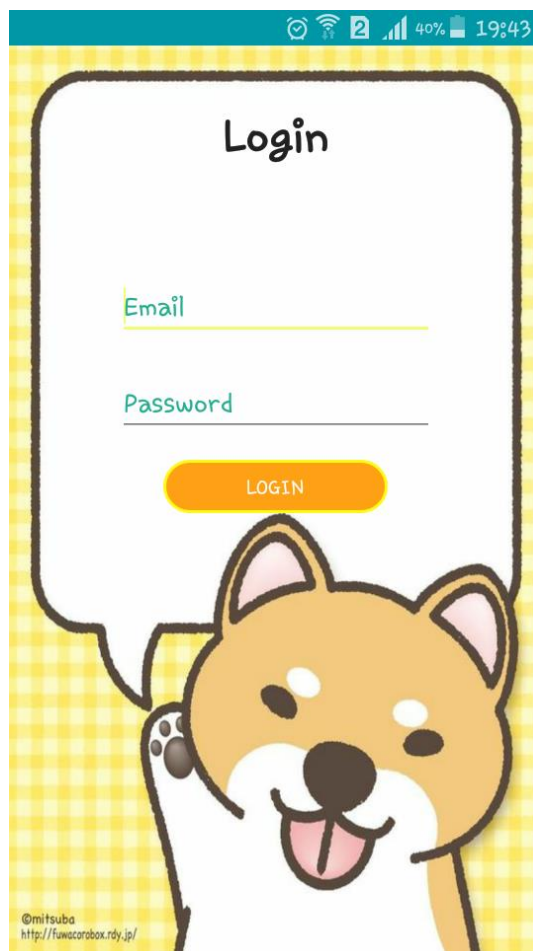
หมายเหตุ : เป็นได้แค่ 2 กรณีคือสามารถอ่านค่า GPS ได้ และ ไม่สามารถอ่านค่า GPS ได้

| | |
|---|--------------------------|
| ทำการทดสอบหน้า ที่หอพักของนักศึกษาที่ ความสูง 5 ชั้น | สามารถอ่านค่า GPS ได้ |
| ทำการทดสอบหน้า ที่ถนนที่มีรถวิ่งผ่าน | สามารถอ่านค่า GPS ได้ |
| ทำการทดสอบหน้า ล็อบบี้(lobby)ที่หอพักของ นักศึกษา | ไม่สามารถอ่านค่า GPS ได้ |
| ทำการทดสอบหน้าตึกคณะ IST ห้อง 106 | ไม่สามารถอ่านค่า GPS ได้ |
| ทำการทดสอบหน้าสนามกีฬา มหานคร | สามารถอ่านค่า GPS ได้ |

จากตารางทดสอบดังตารางที่ 4.2 การทดสอบการทำงานของเซ็นเซอร์ GPS Module และ Node MCU Wi-Fi โดยทำการทดสอบการรับค่าแต่ละสถานที่นั้นตัวอุปกรณ์ของเซ็นเซอร์ GPS Module และ Node MCU Wi-Fiว่ามีขีดจำกัดในการใช้งานหากอยู่ในบริเวณที่อับหรือมีความทึบเกินไปจะไม่สามารถอ่านค่า GPS ได้ และไม่มารดดูตำแหน่งของสัตว์เลี้ยงได้ แต่เมื่อนำอุปกรณ์ไปใช้ในที่โล่ง หรือที่แจ้งจะสามารถอ่านค่า GPS ได้ และสามารถระบุตำแหน่งของสัตว์เลี้ยงได้แม่นยำกว่า

4.2 วิเคราะห์ผลการดำเนินงาน

4.2.1 การเข้าสู่ระบบเพื่อใช้งาน

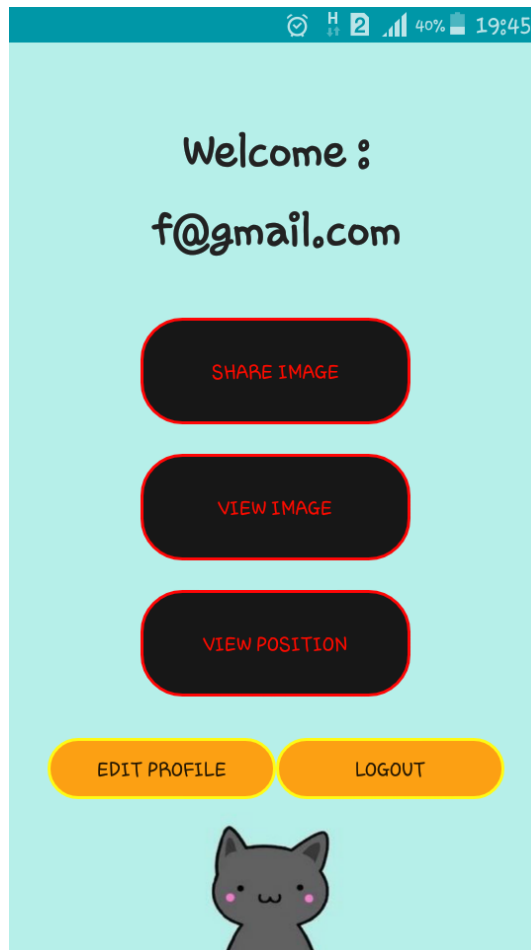


รูปที่ 4.1 หน้าแอปพลิเคชันของผู้ใช้งาน (User)

จากรูปที่ 4.1 เมื่อผู้ใช้งานทำการเปิดแอปพลิเคชันเพื่อเข้าใช้งานผู้ใช้งาน (User) จะต้องทำการยืนยันตัวตนก่อนเข้าใช้งานแอปพลิเคชันโดยประกอบไปด้วย

1. อีเมล (Email) ที่ผู้ใช้งาน (User) ได้ทำการสมัครเพื่อเข้าใช้งาน
2. รหัสผ่าน (Password) ที่ผู้ใช้งาน (User) ได้ทำการสมัครเข้าใช้งาน

จากนั้นทำการกด Login เพื่อเข้าสู่ระบบการใช้งานหน้าต่อไป หากการเข้าสู่ระบบของผู้ใช้งาน (User) เกิดข้อผิดพลาดหรือไม่มีการเชื่อมต่ออินเทอร์เน็ต ระบบจะทำการแจ้งเตือนให้ผู้ใช้งาน (User) ได้ทราบว่าการเข้าสู่ระบบไม่สำเร็จ ระบบจะให้ทำการกรอกข้อมูลเพื่อเข้าสู่ระบบใหม่อีกครั้งหรือทำการเชื่อมต่ออินเทอร์เน็ตใหม่อีกครั้ง เพื่อทำการเข้าไปยังหน้าผู้ใช้งาน (User) หลักของระบบแอปพลิเคชันติดตามสัตว์เลี้ยง



รูปที่ 4.2 หน้าจอหลักของแอปพลิเคชัน

จากรูปที่ 4.2 คือหน้าที่ผู้ใช้งานแอปพลิเคชันสามารถเข้าใช้งานได้เมื่อผู้ใช้งานทำการล็อกอิน (Login) เข้าใช้งาน ระบบก็จะทำการตรวจสอบว่าเป็นผู้ใช้งาน (User) จริงหรือไม่หากบุคคลที่ทำการล็อกอิน (Login) เข้าสู่ระบบไม่ใช่ผู้ใช้งาน (User) แอปพลิเคชันจะไม่สามารถเข้ามาในหน้านี้ได้ โดยในหน้านี้มีปุ่มให้เลือก 5 ปุ่มประกอบไปด้วย

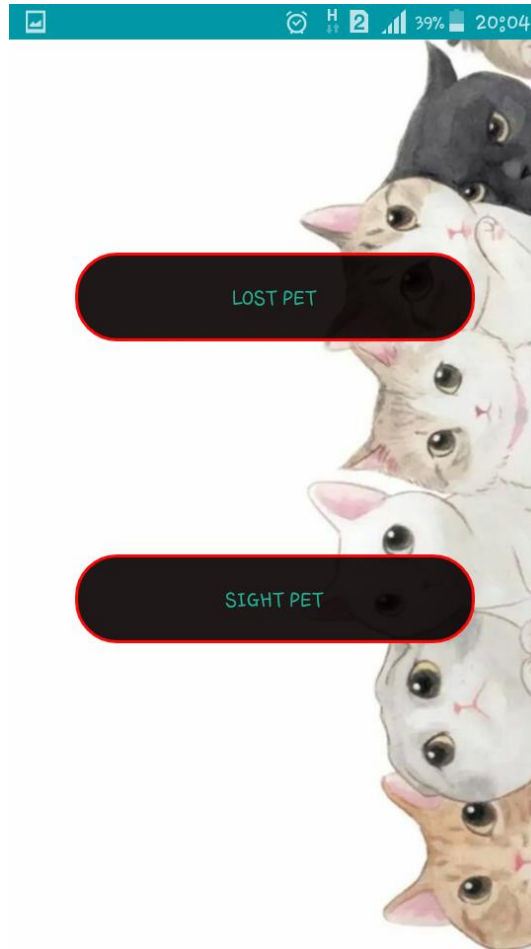
ปุ่มที่ 1 SHARE IMAGE คือการแชร์ภาพ สัตว์เลี้ยงที่หายและสัตว์เลี้ยงที่พบ

ปุ่มที่ 2 VIEW IMAGE คือการเรียกดูภาพและข้อมูลของผู้ที่ทำการแชร์ภาพ สัตว์เลี้ยงที่หายและสัตว์เลี้ยงที่พบ

ปุ่มที่ 3 VIEW POSITION คือ การเรียกดูตำแหน่งที่อยู่ของสัตว์เลี้ยง

ปุ่มที่ 4 EDIT PROFILE ทำการแก้ไขข้อมูลของผู้ใช้งาน (User)

ปุ่มที่ 5 LOGOUT ทำการออกจากระบบผู้ใช้งานแอปพลิเคชัน

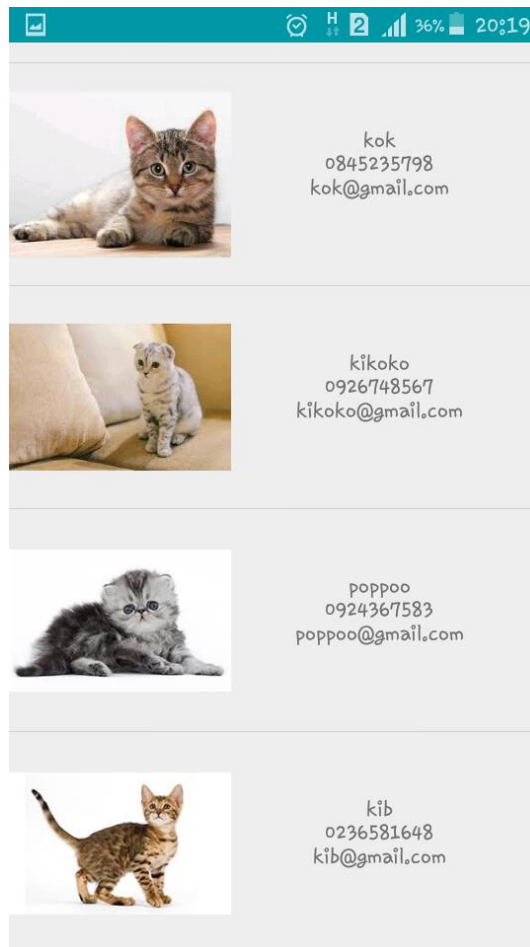


รูปที่ 4.3 หน้าจอการเรียกดูข้อมูลการแชร์ภาพสัตว์เลี้ยงที่ไม่พบใน GPS และสัตว์เลี้ยงที่พบจากผู้ใช้งานแอปพลิเคชันร่วมกัน

จากรูปที่ 4.3 คือหน้าที่ผู้ใช้งานแอปพลิเคชัน สามารถขอข้อมูลการแชร์ภาพสัตว์เลี้ยงที่เลี้ยงที่หายและสามารถขอข้อมูลการแชร์ภาพสัตว์เลี้ยงที่พบเจอตามสถานที่ต่างๆ ได้โดยการขอข้อมูลสัตว์เลี้ยงที่มีปุ่ม 2 ปุ่ม

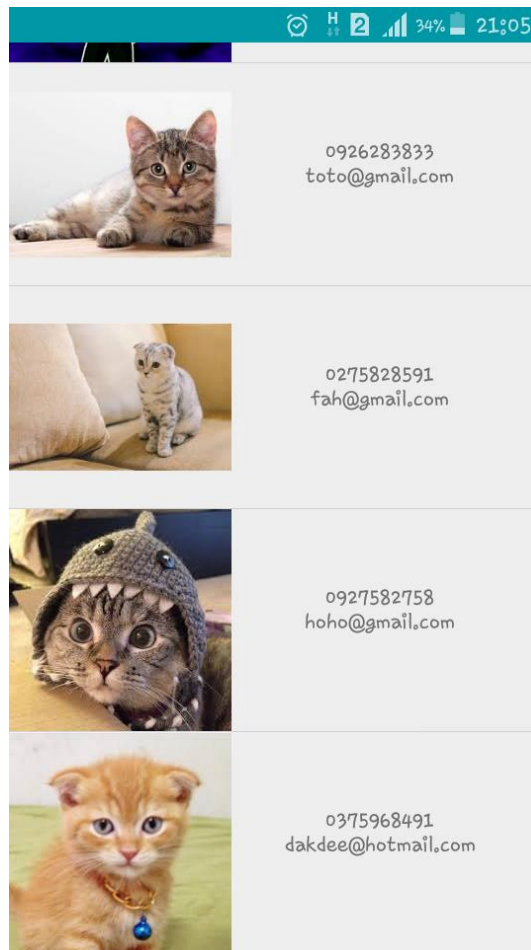
1. (LOST PET) สามารถขอข้อมูลการแชร์ภาพสัตว์เลี้ยงที่หายไปจากบริเวณบ้านหรือสัตว์เลี้ยงได้หายออกไปเป็นเวลานานเพื่อให้คนที่ใช้งานแอปพลิเคชันร่วมกันได้ทำการติดต่อกลับไปให้เจ้าของของสัตว์เลี้ยง

2. (SIGHT PET) สามารถขอข้อมูลการแชร์ภาพสัตว์เลี้ยงที่พบเจอตามสถานที่ต่าง ๆ เพื่อให้เจ้าของสัตว์เลี้ยงทำการติดต่อกลับไปให้ผู้พบเจอสัตว์เลี้ยง



รูปที่ 4.4 หน้าจอการแสดงผลการแชร์ภาพสัตว์เลี้ยงที่ไม่พบใน GPS
จากผู้ใช้และผู้ใช้งานแอปพลิเคชันร่วมกัน

จากรูปที่ 4.4 คือหน้าที่ผู้ใช้งานแอปพลิเคชัน ทำการกดปุ่ม (LOST PET) จากตัวอย่างในหน้า 71 สามารถขอข้อมูลการแชร์ภาพสัตว์เลี้ยงที่หายไปจากบริเวณบ้านหรือสัตว์เลี้ยงได้หายออกไปเป็นเวลานานเพื่อให้คนที่ใช้งานแอปพลิเคชันร่วมกันได้ทำการติดต่อกลับไปให้เจ้าของของสัตว์เลี้ยง เจ้าของสัตว์เลี้ยงจะทำการแชร์ภาพของสัตว์เลี้ยงตัวเองก็ต่อเมื่อ สัตว์เลี้ยงตัวนั้นได้หายออกจากบ้านเป็นเวลานาน และเจ้าของต้องการสัตว์เลี้ยงตัวนั้นกลับคือมาจึงได้ทำการแชร์ภาพ เพื่อประกาศขอความช่วยเหลือจากให้ใช้งานแอปพลิเคชันร่วมกันรับรู้รับทราบโดยทั่วกัน หากผู้ใช้งานแอปพลิเคชัน พบเจอสัตว์เลี้ยงตัวที่ได้ทำการแชร์ภาพบนแอปพลิเคชัน ก็จะสามารถติดต่อกลับมายังเจ้าของสัตว์เลี้ยงได้โดยตรง



รูปที่ 4.5 หน้าจอการแสดงผลการแชร์ภาพสัตว์เลี้ยงที่พบจากผู้ใช้และผู้ใช้งานแอปพลิเคชันร่วมกัน

จากรูปที่ 4.5 คือหน้าที่ผู้ใช้งานแอปพลิเคชัน สามารถดูข้อมูลการแชร์ภาพสัตว์เลี้ยงที่เลี้ยงที่หายเพื่อทำการติดต่อกลับไปเจ้าของของสัตว์เลี้ยง ในกรณีที่เป็นคนพบเจอสัตว์เลี้ยง ในหน้านี้จะมีข้อมูลของสัตว์เลี้ยงที่หายไปจาก GPS ได้ ก็ต่อเมื่อ ผู้ใช้งานไม่พบสัตว์เลี้ยงของตัวเองจึงทำการแชร์ภาพสัตว์เลี้ยงของตัวเองลงใน (PET LOST)

4.2.2 ข้อจำกัดของแอปพลิเคชันติดตามสัตว์เลี้ยง

4.2.2.1 การทำงานของตัวอุปกรณ์มีขีดจำกัด

4.2.2.1.1 การใช้งานของตัวอุปกรณ์ต้องแบตเตอรี่และแบตเตอรี่ก็มีขีดจำกัดในการใช้ ตามระยะเวลา หรือการใช้งานของผู้ใช้งานในระบบ

4.2.2.1.2 อุปกรณ์ต้องติดที่ตัวสัตว์เลี้ยงเสมอ

4.2.2.1.3 ตัวอุปกรณ์จะส่งค่า GPS ได้ก็ต้องเมื่อมีการเชื่อมต่อกับไวไฟ (Wi-Fi) เท่านั้น ถึงจะสามารถดูตำแหน่งของสัตว์เลี้ยงได้

4.3 แนวทางการพัฒนาในอนาคต

- 4.3.1 Mobile Application สามารถทำงานได้บนระบบปฏิบัติการ Android เท่านั้น ไม่สามารถทำงานบนระบบอื่นได้
- 4.3.2 ออกแบบระบบทั้งหมดให้รองรับการใช้งานจำนวนมาก
- 4.3.3 ระบบสามารถอ่านค่าตำแหน่ง GPS Reader
- 4.3.4 ระบบสามารถแชร์ภาพได้หลายส่วน
- 4.3.5 ควรรองรับการใช้งานของ ผู้ใช้งานให้มากกว่านี้
- 4.3.6 ผู้ใช้งาน สามารถ เพิ่มข้อมูลสัตว์เลี้ยงได้มากกว่า 5 ตัว
- 4.3.7 ควรเพิ่มช่องทางการค้นหาข้อมูลให้ง่ายกว่านี้ เช่น เพิ่มช่องค้นหา
- 4.3.8 ควรทำให้อุปกรณ์มีขนาดเล็กกว่านี้
- 4.3.9 ควรทำให้อุปกรณ์มีระยะเวลาในการใช้งานมากกว่านี้ (แบตเตอรี่)
- 4.3.10 ระบบสามารถแยกประเภทการแชร์ภาพสัตว์เลี้ยงได้
 - 4.3.10.1 แยกสีของสัตว์เลี้ยง
 - 4.3.10.2 แยกลักษณะของสัตว์เลี้ยง
 - 4.3.10.3 แยกพันธุ์ของสัตว์เลี้ยง
- 4.3.11 ระบบสามารถแยกประเภทการค้นกาสัตว์เลี้ยงได้
 - 4.3.11.1 แยกสีของสัตว์เลี้ยง
 - 4.3.11.2 แยกลักษณะของสัตว์เลี้ยง
 - 4.3.11.3 แยกพันธุ์ของสัตว์เลี้ยง

เอกสารอ้างอิง

- [1] งานวิจัยเรื่องโครงการระบบควบคุมไฟฟ้าไร้สายผ่านเว็บเบราว์เซอร์ งานวิจัยของนายประธาน เนียมน้อย นายจิตติ คงแก้วนายจตุรงค์ มะโน
- [2] จักรชัย พงษ์ศธร และณัฐนิชา วีระมงคลเลิศ. Android App Development ฉบับสมบูรณ์. พิมพ์ครั้งที่1. นนทบุรี: ไอดีซี พรีเมียร์, 2555
- [3] ปลื้มจรินทร์ อาสาทรงธรรม. (2546). Innovation หรือ นวัตกรรม คืออะไร? วารสารนักบริหาร, 23. จิรวัดน์ วงศ์ธงชัย. (2555). ปัจจัยด้านการรับรู้ที่มีผลต่อการยอมรับเทคโนโลยีบาร์โค้ดสองมิติของผู้ใช้งานกลุ่มเจนเอเรชั่นวาย. วิทยานิพนธ์ปริญญาโทบริหารธุรกิจ, มหาวิทยาลัยเทคโนโลยีสุรนารี.
- [4] นัยนา แก้วบ้านดอน. (2557). Development of Innovative Learning (iLearning) to Enhance Smart Teachers and Smart Student. International Conference Education and Leadership in Globalization: What does “think globally act locally” mean for education around the world? (ELGIC 2014).
- [5] สุวิชัย อินทรภิมย์. (2554). ระบบนำทางการท่องเที่ยวของจังหวัดสุราษฎร์ธานีด้วย กูเกิ้ลแมพ เอพีไอ บนมือถือระบบปฏิบัติการแอนดรอยด์. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ.
- [6] กอบเกียรติสระอุบล. การพัฒนา App Android . พิมพ์ครั้งที่1.กรุงเทพฯ: มีเดียเนทเวิร์ค, 2556
- [7] ล้วน สายยศและอังคณา สายยศ. (2543). เทคนิคการวิจัยทางการศึกษา. พิมพ์ครั้งที่3. กรุงเทพฯ: สุวีริยาสาส์น.