

FurRescue: A Mobile Application for Pet and Stray Animal Locator with Geo-Fencing and AI Breed Detection

Yong Huei Jean¹, Norfaradilla Wahid^{1*}

¹ *Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA*

*Corresponding Author: faradila@uthm.edu.my

DOI: <https://doi.org/10.30880/aitcs.2025.06.01.001>

Article Info

Received: 13 June 2024

Accepted: 8 May 2025

Available online: 30 June 2025

Keywords

Strays, Adopt, Rescue, Flutter, Dart, TensorFlow 2.0, Geofencing, Mobile Application, Pets

Abstract

Stray dogs and cats in Malaysia are increasingly problematic due to unspayed animals and inconsistent feeding practices. Local councils' spay-and-return efforts are undermined by individuals feeding strays without spaying them. This complicates locating strays for adoption or reuniting lost pets with their owners, while social media posts often fail to gain sufficient visibility. The FurRescue app, developed with Visual Studio Code, Flutter, Dart, TensorFlow 2.0, and Firebase, addresses these issues by enabling users to list strays, lost, and found pets. Features include dog breed detection, geofencing, a community forum, and a chatroom for direct communication. User acceptance testing showed the app's functionalities are effective and its UI/UX design well-received.

1. Introduction

Stray dogs and cats are a prevalent issue in Malaysia, with their numbers steadily increasing yearly. According to Subang Jaya City Council, there were 1000 cases, while the Sabak Bernam District Council recorded 3000 public complaints about stray dogs in 2022 [1]. According to The Star, an English-language newspaper in Malaysia, on June 12, 2023, the Selangor Local Government, Public Transport, and New Village Development Committee chairman Ng Sze Han revealed that the Selangor government plans to ban pet stores from selling cats and dogs [2]. Since the Selangor government is promoting cat and dog adoption and is planning to ban pet stores from selling cats and dogs with the hope of reducing the number of stray cats and dogs, stray dogs and cats will soon be high in demand. An application is proposed to provide a platform to adopt stray dogs and cats and also to find lost pets. A few objectives have been set as follows to achieve the aim:

- i. To analyse and design a specific platform that enables the user to access the pets or stray animals that are available for adoption
- ii. To develop an efficient and valuable mobile platform for users to get information on lost and found pets
- iii. To implement testing on the functionality for the developed application on the target users

This project involves building a cross-platform application for Android and IOS devices targeted at Malaysians. Users can adopt stray animals and get or put information about lost and found pets at their fingertips. Users no longer need to access social media like Facebook to look for posts. Users will be able to get notified from the proposed application whenever they have entered an area that consists of strays or lost or found pets. The photo of the dog that uploaded by the users will be classify to identify the breed using the TensorFlow Lite model. Furthermore, the proposed application uses geofencing technology which get the current location of the user and notify the user using push notification when the created strays, lost pet, or found pet listing is within their geofence boundary.

The application speeds up the process of adopting nearby dogs and cats by looking at the map without asking for the location. It puts a wealth of information, resources, and support at users' fingertips, all accessible through

one convenient online platform. Offering this consolidated approach enhances the user experience and eases the process of lost and found pets.

The paper is organized as follows: Section 1 discusses the introduction to the study, Section 2 will present the related work, followed by Section 3 which explains the methodology or framework that used in the entire project. Lastly, Section 4 which comprises the implementation of the proposed modules and analysing the results by performing the testing.

2. Related Work

This chapter explains the literature review that has been conducted for this project. The discussion will start with an overview of stray dogs and cats, followed by the technology used in developing the application, which are TensorFlow 2.0 with Keras, geofencing, Dart language, Flutter framework, and Firebase. The comparison of the existing systems with the proposed application will also be discussed.

2.1 Existing Approach to Adopt Strays

People who find stray dogs and cats will choose to post on social media as there is no specific platform for them to help stray dogs and cats get adopted. The user will take photos of the strays and post them on social media, such as Facebook, Instagram, and WhatsApp. In some cases, people may prefer to share the information within WhatsApp community groups to ask their neighbour if they are interested especially if the strays are found in their vicinity. Similarly, pet owners who lost their pets or people who found the lost pets will do similar actions.

2.2 Technology Used

This section aims to discuss the technologies used in developing this application. This application is a mobile application focused on multi-platform such as IOS and Android, and therefore, the Flutter framework and Dart language are used in developing this application. Firebase is used as a database in the development process of this application. Geofencing is a technology that relies on GPS and other location-based technologies, and it is used in the project to show the nearby lost and found pets and strays. Additionally, the model is trained using TensorFlow 2.0 with Keras to classify the dog breed.

2.2.1 TensorFlow 2.0

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications [3]. TensorFlow version 2.0 is used in this project to train the model for image classification, specifically for predicting dog breeds. The dataset used for training is provided by TensorFlow and is known as the Stanford Dogs dataset which consists of 120 breeds of dogs. The model is subsequently converted to TensorFlow Lite (tflite) format before being imported into the mobile application. The TensorFlow Lite model reduces the overall size of the application and ensures that the model can run smoothly on the user's device.

2.2.2 Geofencing

Geofencing is a technology that builds on top of geolocation technology and uses it to establish virtual boundaries around particular locations [4]. The event triggered on a smartphone every time the user entered the virtual border around the geofenced area. This technology does not continuously track a user's exact geographic location at regular intervals based on the app's settings. The user's location is only disclosed when they enter or exit the predefined area. This protects the user's privacy as it does not expose the user's location intrusively.

2.2.3 Dart Language

Dart is an open-source general-purpose programming language developed and maintained by Google, and it has been proven that Dart has the ability to develop robust web applications [5]. It supports application development on both the client and server side. With Dart, the Android application, IOS application, web application, and IoT (Internet of Things) can be easily developed using the Flutter framework.

2.2.4 Flutter Framework

Flutter is an open-source SDK that released in 2016 by Google and is able to build stable and high-performance cross-platform mobile applications, for example, iOS and Android operating systems. Flutter also enables developers to create web and desktop applications. Flutter can be used for both front-end and back-end

development. The programming language used by Flutter is Dart. The application that uses Flutter in the development process is faster and easier than the traditional ways. Using the Flutter framework makes it easy for the application to create interactive and pretty user interfaces. This is due to the automatic reconstruction of the widget tree offered by the Flutter framework. This allows the developer to see the changes instantly without reloading the application. This means that the developer is able to write the code and see the design reflected in real time.

2.2.5 Firebase

Firebase is a set of backend cloud computing services and application development platforms provided by Google. Firebase is categorized as a NoSQL database program that stores data in JSON-like documents [6]. Firebase not only offers a real-time database but, at the same time, offers authentication services as well. Firebase also provides cloud storage for developers to store large files without hesitation. Developers do not need to purchase a server as the Firebase operates on a serverless architecture. Furthermore, Firebase is more suitable in handling real-time data like chat application.

2.3 Study on Similar Systems

A comparison between the three existing applications and the proposed application is made. The existing applications that are compared with the proposed application are Missing Pets – Find Lost Pet [6], PeTrace – Lost Pets [7] and PetFinder.my [8].

The Missing Pets is an application that helps the pet's owner to create post that related their lost pet and the pet that they found without any owner. Posting a lost pet or found pet with a photo and location does not charge the user, but the application offers a few extra functions that can be purchased, such as pinning the post created by the users on the main page. PeTrace – Lost Pets, an application that allows the user to create posts of lost and found pets and, at the same time, allows users to create posts for the adoption of pets. All lost, found and adopted pets are geolocated on the map according to the filter applied. PetFinder.my is an application that allows the user to adopt the pets, report lost and found pets. User will be able to comment below the post to ask for further information.

The comparison is made based on their operating system, user experience (UX), user interface design (UI), and functions. The comparison of the existing applications and the proposed application is shown in Table 1. The main difference between the proposed application and the existing system is the proposed application utilises geofencing technology, which can create push notifications to notify users if they have entered a specific location within strays or lost or found pets. The proposed application also uses the TensorFlow 2.0 with Keras to train the dog breed dataset to classify the dog breed. The chatroom offers the chance to get further information on the listings, unlike other similar systems, which need to provide additional contact methods if want to get additional information. The forum is proposed to enable the users to inquire about strays and pets based on topic categories.

Table 1 Comparison Table Between Existing Systems and Proposed System

Category	Features/System	Missing Pets – Find Lost Pet	PeTrace – Lost Pets	PetFinder.my	Proposed Applicatoin
Operating system	Android-based	Yes	Yes	Yes	Yes
	iOS-based	Yes	Yes	Yes	Yes
User experience (UX)	Simple interface	Yes	Yes	Yes	Yes
	Easy to navigate	No	Yes	Yes	Yes
User interface (UI)	Invisible design	Yes	No	No	Yes
	Connection between the stray helper	No	No	No	Yes
	A high number of advertisements	No	Yes	No	No

Table 1 Comparison Table Between Existing Systems and Proposed System (cont.)

Category	Features/System	Missing Pets – Find Lost Pet	PeTrace – Lost Pets	PetFinder.my	Proposed Application
Functions	Get notified when animals are within the specific virtual boundaries of the user	No	No	No	Yes
	Dog breed detection	No	No	No	Yes
	Forum	No	No	No	Yes
	Chatroom	No	No	No	Yes
	Create a lost pet listing	Yes	Yes	Yes	Yes
	Create a found pet listing	Yes	Yes	Yes	Yes
	Create strays listing	No	No	No	Yes
Create report	No	No	No	Yes	

3. Methodology/Framework

Methodology is a research method comprising practices, strategies, techniques, processes, rules and procedures [9]. To achieve the objectives and the scope of the project, the methodology of the Prototyping Model is chosen. The planning phase, analysis phase, hardware and software requirements, design phase, implementation phase, and system development workflow will be discussed in this section.

3.1 Prototype Model

The Prototyping Model is one of the most popularly used Software Development Life Cycle (SDLC) Models. This model is essential for unclear requirements early in the development process. The model helps clarify the user requirements by developing the prototype of the end product. The prototype of the end product is then tested and refined repeatedly as per user feedback to achieve the objectives and scopes of the project. This model involves the users and the stakeholders in the development process, which provides immediate feedback. This model was selected because it allows the developer to identify errors during development. The system development prototyping model is shown in Figure 1.

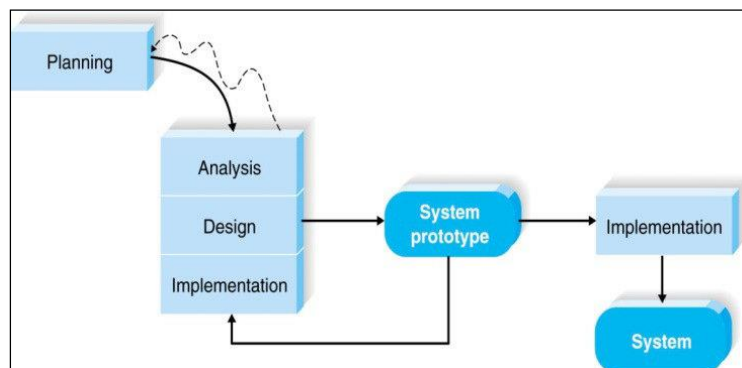


Fig. 1 Prototyping Model [10]

3.1.1 Planning Phase

In this phase, the overall scope, objectives, and requirements of the projects are outlined. The phase determines the fundamentals of the project. In the planning phase, the development framework, technologies used, the project plan, and the project schedule are identified. To ensure the works are organised and well-managed, the Gantt Chart is drawn and is shown in Appendix A. Using the Gantt Chart, the important tasks will be prioritised and ensure the project can be finished within the scheduled time and on the right track.

3.1.2 Analysis Phase

In this phase, the requirements, including the functional requirements and non-functional requirements, are defined. All the collected information is analysed, and the project objectives have been converted into the specified system functions for the new proposed application. The study of similar systems in the market is conducted to have comparison between the proposed application and similar systems. This is to let the developer identify the advantages and the limitations of the functionalities in the existing application in the market, avoid them, and propose modules with functionalities that do not exist in the existing application. The proposed application will use Object-Oriented Analysis and Design (OOAD) to design and implement software systems. OOAD is a software engineering methodology that uses object-oriented concepts to design and implement the functionalities of the software [11]. Unified Modelling Language (UML) diagrams are drawn to represent the components and the interactions between the functionalities in the system. The UML diagrams, including the use case diagram, sequence diagram, and class diagram, are attached in Appendix B, C, and D respectively.

3.1.2.1 Functional Requirement

Functional requirement analysis is a crucial phase in the software development life cycle (SDLC). The functional requirement explains the behaviour of the system under various conditions. The functional requirement includes the process of the system that must be executed and the information it must contain. Table 2 shows the functional requirements for the proposed application.

Table 2 *Functional requirements for the proposed application*

Functions	Functionalities
Register	<ul style="list-style-type: none"> • The application should allow users to register an account. • The application should not allow an existing email and username to register a new account. • The application should only allow verified registered users to redirect to the login page. • The application should allow the users to sign up using Google Sign-up.
Login	<ul style="list-style-type: none"> • The application should allow users to input an email address or username and password to log in. • The application should allow the users to sign in using Google Sign-in. • The application should allow users to reset their password if they have forgotten their password
Logout	<ul style="list-style-type: none"> • The application should allow users to log out.
Delete user account	<ul style="list-style-type: none"> • The function should allow the user to delete their account.
Profile update	<ul style="list-style-type: none"> • The function should allow the user to delete their account.
Conversation	<ul style="list-style-type: none"> • The application should allow users to chat with another user.
Forum	<ul style="list-style-type: none"> • The application should allow the users to create, read, delete, and update the post.
Listing	<ul style="list-style-type: none"> • The function should allow users to create lost or found pets and strays listing. • The functions should allow users to remove the listing if the case is solved.
Favourite listing	<ul style="list-style-type: none"> • The function should allow users to save their favourite listing in the 'Favourite' tab for future use.
Geofencing	<ul style="list-style-type: none"> • The function should update the location of the user in real-time. • The function should allow users to view the map and adjust the radius of their geofence accordingly. • The function should allow the application to send the push notification to users when a geofence is triggered.
Dog breed classification	<ul style="list-style-type: none"> • The function should allow the user to upload the dog image and get the result of the predicted dog breed. • The function should have displayed the probability of the predicted dog breed.

Table 2 *Functional requirements for the proposed application (cont.)*

Functions	Functionalities
Report	<ul style="list-style-type: none"> The function should allow users to write reports about the strays in their community and send them to the Majlis Perbandaran. The function should be able to retrieve the current location of the user and send the report to the email of the nearest Majlis Perbandaran.

3.1.2.2 Non-Functional Requirement

Non-functional requirement analysis in software development defines the criteria that describe the performance and limitations of the system that are related to the user's experience. Table 3 shows the non-functional requirements for the proposed application.

Table 3 *Non-functional requirements for the proposed application*

Functions	Functionalities
Operational	<ul style="list-style-type: none"> The application should work with internet services. The database should be updated in real-time.
Performance	<ul style="list-style-type: none"> The application should be available to be used anytime and anywhere.
Security	<ul style="list-style-type: none"> The application should only allow verified users with the correct email and password to log in to the application.

3.1.3 Design Phase

To achieve the objectives of the project, the design of the application is important as it produces the functional requirements of the proposed application. The wireframe of the user interface will be designed before prototyping, allowing the designer to draft the overall design concept rapidly and provide the basic structure of the user interface. This gives a visualisation of the layout of the component of an application for the designer and the stakeholder, hence avoiding time wasted in thinking about the layout of the user interface during the prototyping. The wireframes for the user interface are attached in Appendix G.

3.1.4 Implementation Phase

The implementation phase starts after the prototype has successfully met all the user requirements. The code is written using the Visual Studio Code as the integrated development environment (IDE). The mobile application is developed using Flutter as a UI framework and Dart as a programming language. The dog breed classification model is trained using TensorFlow 2.0. and converted into TensorFlow Lite (tflite) format before being imported into the mobile application. The Firebase is connected to the mobile application. Upon sign-up, the email verification will be sent to the user to verify their email and registration. The user interfaces are drawn and the prototype of the application is developed.

3.1.5 System Development Workflow

There are 4 development phases in the prototype model, which are planning, analysis, design, and implementation. The phase will have several activities that need to be conducted throughout the project development, and the output of the activities will be tabulated. Table 4 shows the system development workflow of the proposed application.

Table 4 *System Development Workflow of the Proposed Application*

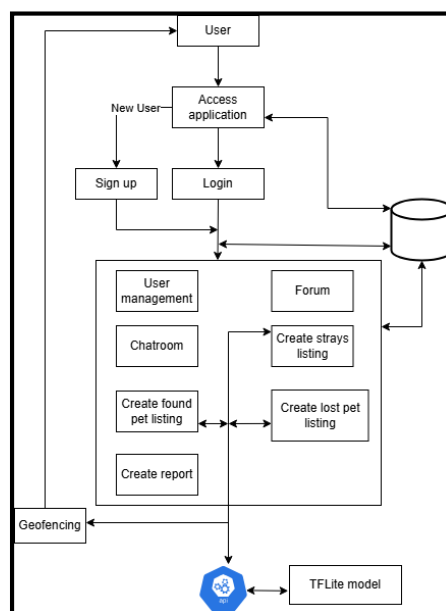
Phase	Activity	Output
Planning	<ul style="list-style-type: none"> Identify problem statements, objectives, project scope, expected result, and project significance Schedule the project timeline 	<ul style="list-style-type: none"> Project proposal Gantt Chart Introduction Literature review

Table 4 System Development Workflow of the Proposed Application (cont.)

Phase	Activity	Output
Analysis	<ul style="list-style-type: none"> - The existing mobile applications in the market are observed - The features of the mobile applications and the proposed mobile application are observed and recorded - Identify the hardware and software requirement - Determine the interaction of components of the application - Determine the view of data - Determine the relationship between the classes 	<ul style="list-style-type: none"> - Functional and non-functional requirements - Software and hardware requirement - Use case diagram - Sequence diagram - Activity diagram - Class diagram
Design	<ul style="list-style-type: none"> - Visualise the basic structure and layout of the user interface of the application - Design database 	<ul style="list-style-type: none"> - Wireframes of the application - Prototype of the application - Database design
Implementation	<ul style="list-style-type: none"> - Develop the proposed application based on the functional requirement and non-functional requirements - Connect the application with the database 	<ul style="list-style-type: none"> - Proposed system

3.2 System Architecture

The system architecture diagram provides a high-level overview of how the components in the system interact with each other and work together. The system architecture is said to be the definition of a software system. The system architecture diagram helps the reader easy to understand how the system works using a diagram. Figure 2 shows the system architecture of the proposed application.

**Fig. 2** System architecture of the proposed application

3.2.1 Unified Modelling Language (UML)

Unified Modelling Language (UML) is a visual language to capture software designs and patterns. The UML is a diagram used to describe, visualise, build, and record software system components [12]. The UML diagrams consist of four types, which are use case diagram, sequence diagram, activity diagram, and class diagram. The UML of the project will be drawn and described in this section.

3.2.2 Use Case Diagram

The functional requirements of the proposed system are visually presented using a use case diagram. The use case diagram is developed as part of the analysis to provides a visual description of the actors and the use case involved in the system. The arrows show the interaction between the actors and the use case. The user, administrator, GPS system, and the API are the actors in this project. The use case diagram is attached in Appendix A. The use case diagram consists of 17 main use cases, which are sign up, log in, log out, delete the account, update profile, create a post, read the post, update post, delete a post, create lost pet listing, create found pet listing, create strays listing, delete listing, view listing, conversation, create a report, and upload a picture for dog breed classification. These use cases are the functional requirements of the proposed application. The actors in this application are the user and the GPS system.

3.2.3 Class Diagram

The class diagram is one of the types of UML and is to be applied during the project phases such as analysis, design and implementation phases. The class diagram also shows the overview of the software. The class diagram consists of the name of the class, attribute, and method. The name of the class must be unique.

3.2.4 User Interface Design

User Interface (UI) design is important in software development. User interface significantly influences the user experience when using the application, and a bad user interface will not attract users to continue using the application. A good user interface ensures the user can navigate to the desired page and understand the features provided in the application easily.

3.3 TensorFlow Lite model

The dataset that is used to train the dog breed classification model is called 'Stanford Dog' [13]. This dataset is provided by the TensorFlow. The dataset is loaded, as shown in Figure 3. The dataset consists of 20580 images, while the first 3000 are split for training and the last 20 % is split for testing in this project. TensorFlow 2.0 is used together with Keras, which is a high-level API for building and training deep learning models. The pre-trained model that is used in this project is called the MobileNetV2 model, which is loaded from TensorFlow Hub. After the training of the model, the model is converted to TensorFlow Lite, which is much smaller and crucial for mobile devices with limited storage. The class labels of the dataset are extracted and saved to a file called labels.txt, which will be imported into the Flutter mobile application, as shown in Figure 5.

```

1 import tensorflow as tf
2 import tensorflow_hub as hub
3 import tensorflow_datasets as tfds
4 from tensorflow.keras import layers, models

1 # load dataset
2 (train_dataset, test_dataset), dataset_info = tfds.load(
3     'stanford_dogs',
4     split=['train[:3000]', 'test[80:]'],
5     shuffle_files=True,
6     as_supervised=True,
7     with_info=True
8 )

```

Fig. 3 Load dataset from the TensorFlow

```

1 URL = "https://tfhub.dev/google/tf2-preview/mobilenet_v2/classification/4"
2 base_model = hub.KerasLayer(URL, input_shape=(IMG_SIZE, IMG_SIZE, 3), trainable=False)

```

Fig. 4 Load dataset from the TensorFlow

```

ds_info = tfds.builder('stanford_dogs').info
labels = ds_info.features['label'].names

with open('labels.txt', 'w') as f:
    for label in labels:
        f.write(f"{label}\n")

```

Fig. 5 Saved labels to labels.txt file

4.0 Result

The implementation and testing will be discussed in this section. The application has been tested by conducting user acceptance testing to ensure that the developed application is well accepted by the target users and meets the objectives of this project.

4.1 Implementation

The implementation and testing will be discussed in this section. The application has been tested by conducting user acceptance testing to ensure that the developed application is well accepted by the target users and meets the objectives of this project. This application includes seven modules: module user management, the module of communication between the users, the module of strays, the module of lost pets, the module of found pets, the module of forum, the module of chatroom and the module of creating the report. The application is developed using the Flutter UI framework, TensorFlow 2.0, Dart language, and Firebase as the database. Figure 6 shows the login interface with code segment. If the email is not verified yet, the user will be redirect to email verification screen. Figure 7 shows the sign-up interface. Users can use Google Sign-In to sign in or sign up for their account. Figure 8 shows the listing interface with geofencing features when viewing the listing from map. Figure 9 shows the stray listing interface with favourite icon on-pressed and the favourite tab. Figure 10 shows the forum interface and the chatroom interface. Figure 11 shows the interface for creating the report and the interface of showing the nearest Majlis Perbandaran after selecting the location. Figure 12 shows the dark mode interface for creating a listing and the dog breed detection with result after uploading the dog image. Figure 13 shows the code segment of performing inference using TFLite model.

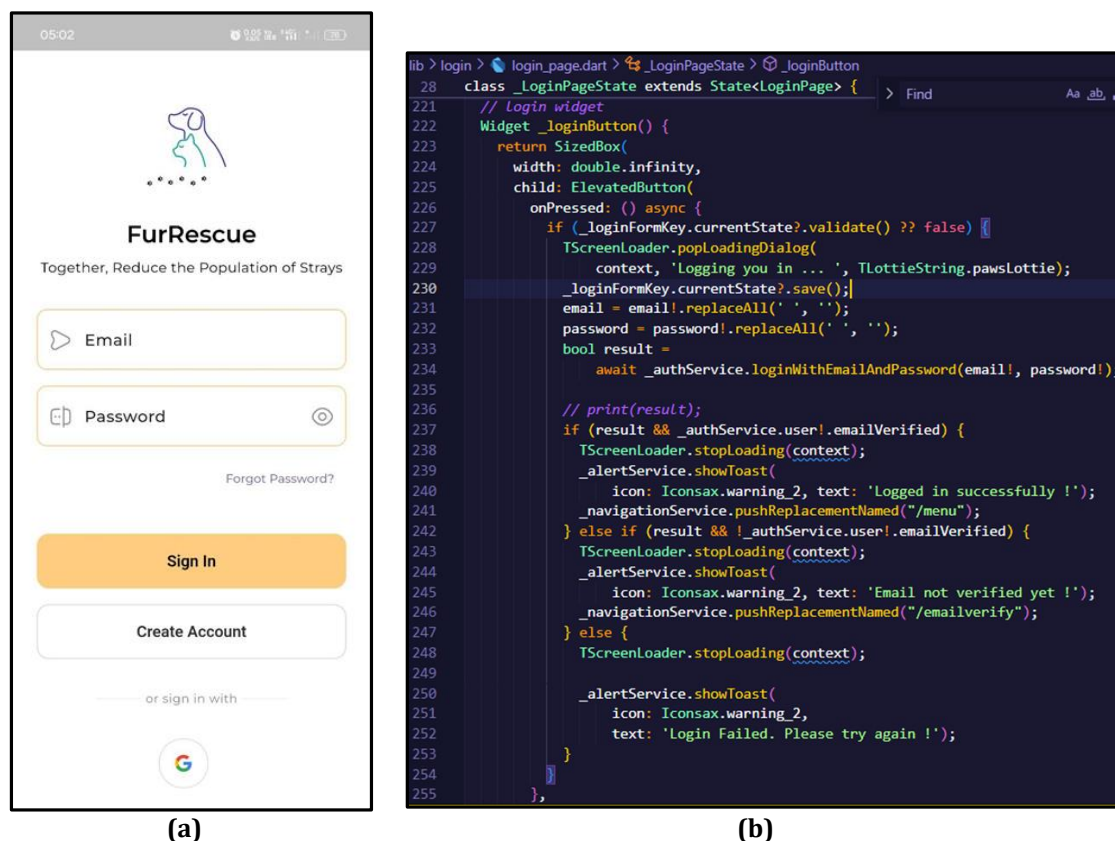


Fig. 6 (a) User sign-in screen (b) Code segment for user sign-in

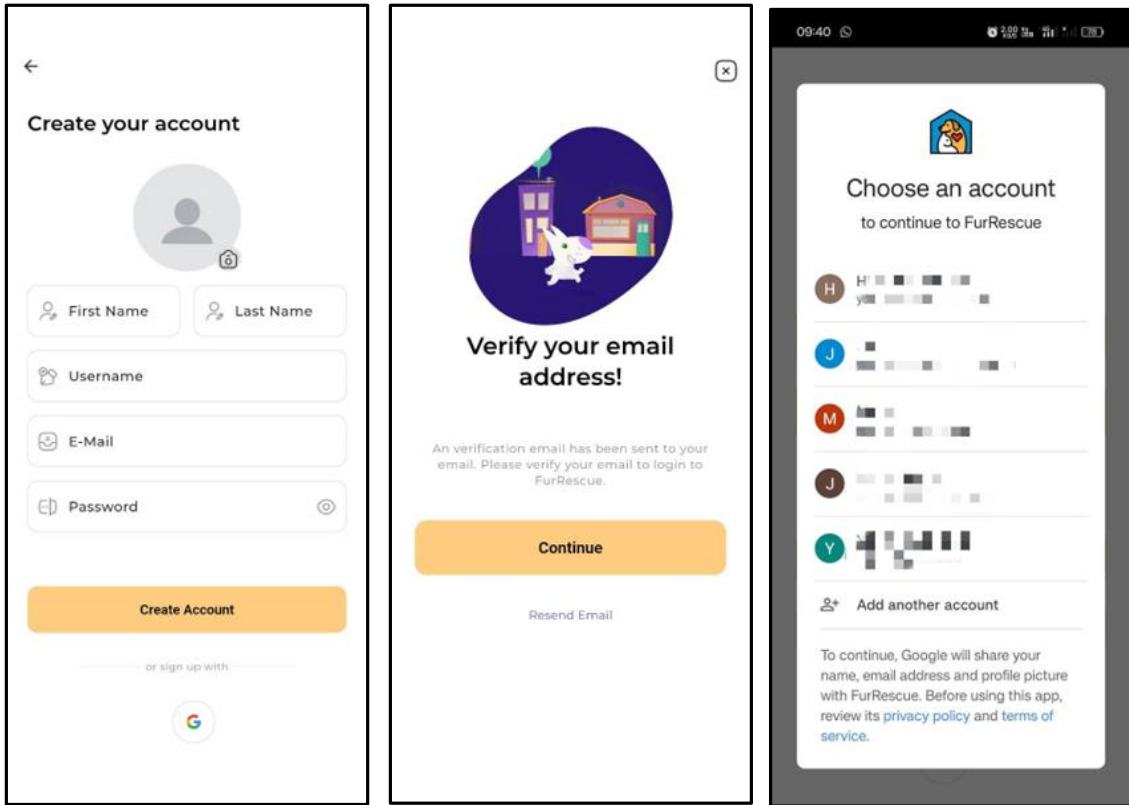


Fig. 7 (a) User sign-up screen (b) Email verification screen (c) Google Sign-in function

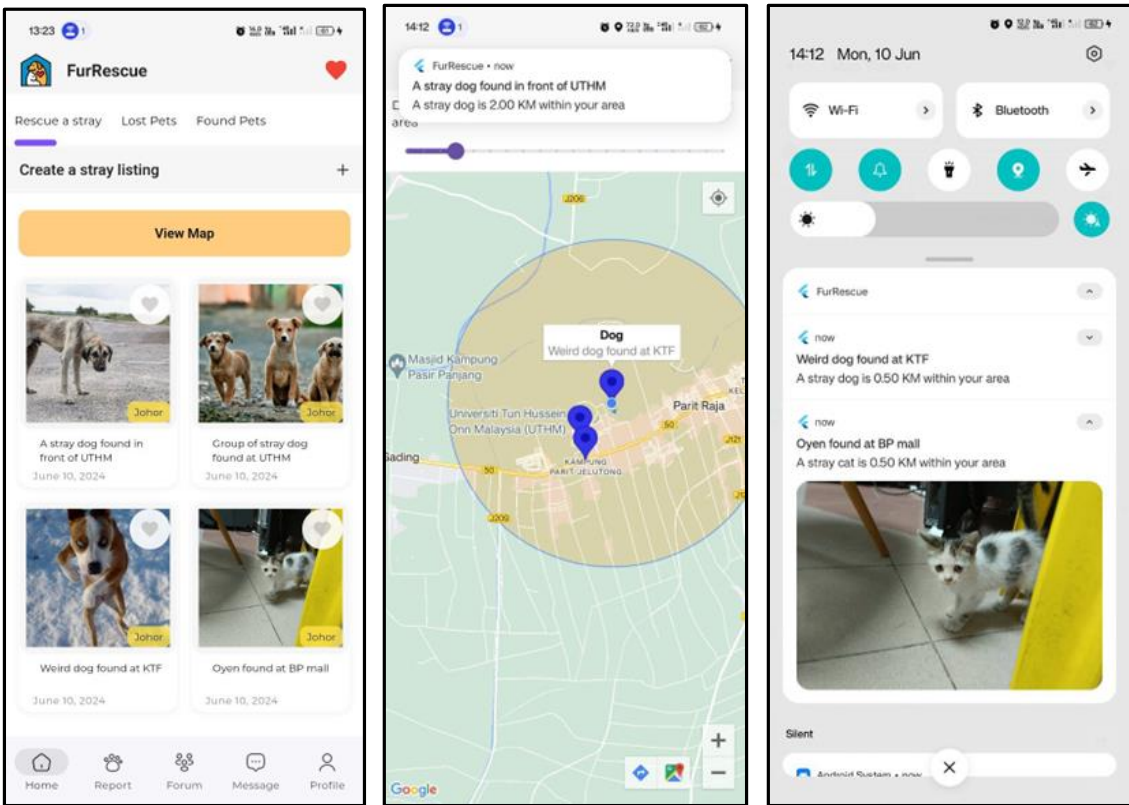


Fig. 8 (a) Listing interface (b) Geofencing feature on map interface (c) Notification triggered

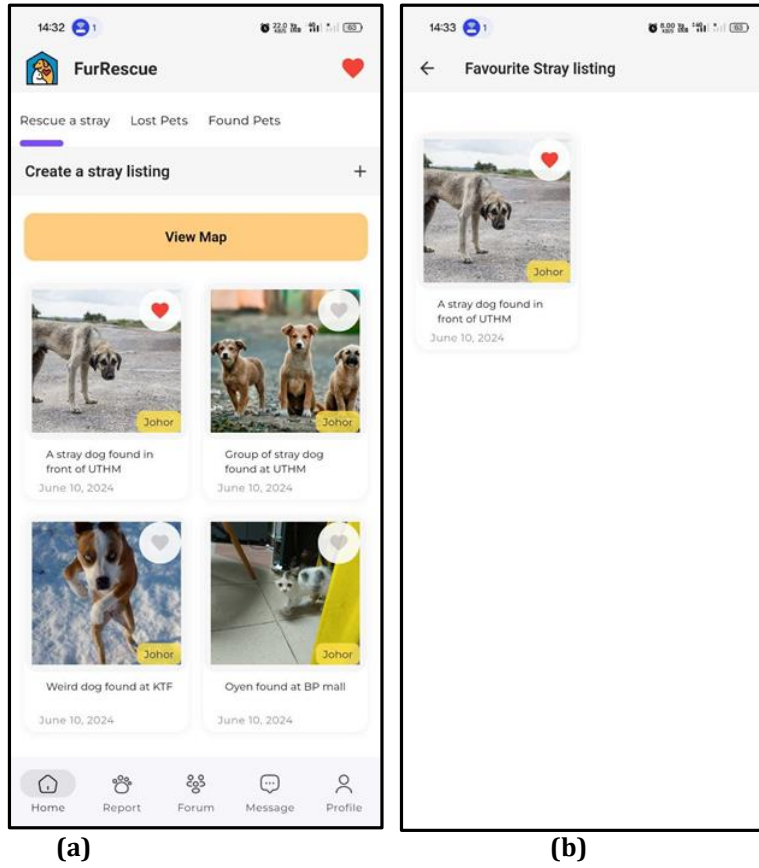


Fig. 9 (a) Favourite icon on listing interface (b) Favourite tab interface

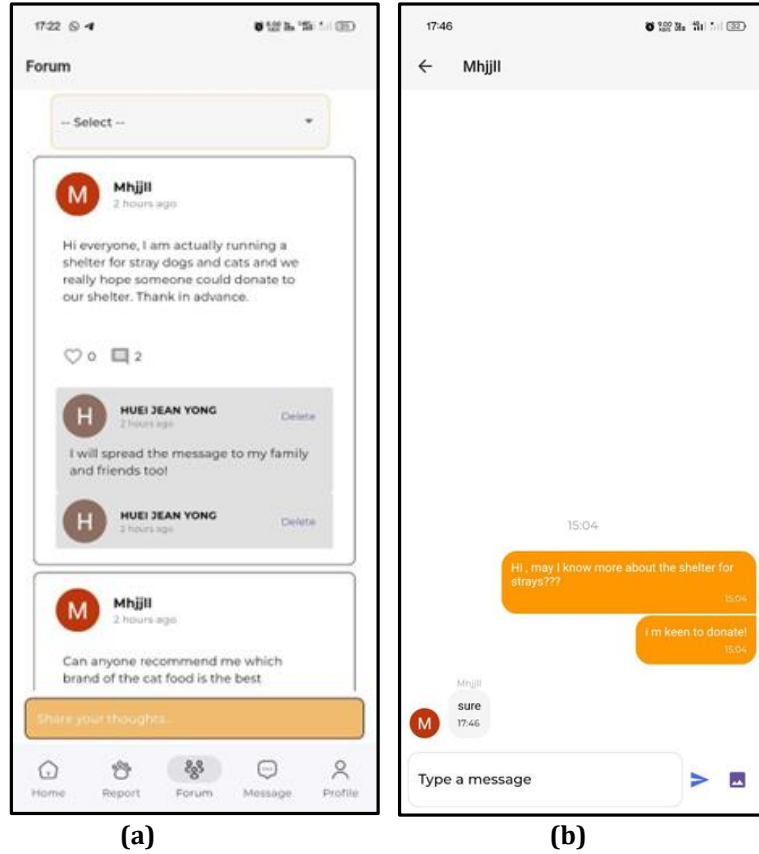


Fig. 10 (a) Forum interface (b) Chatroom interface

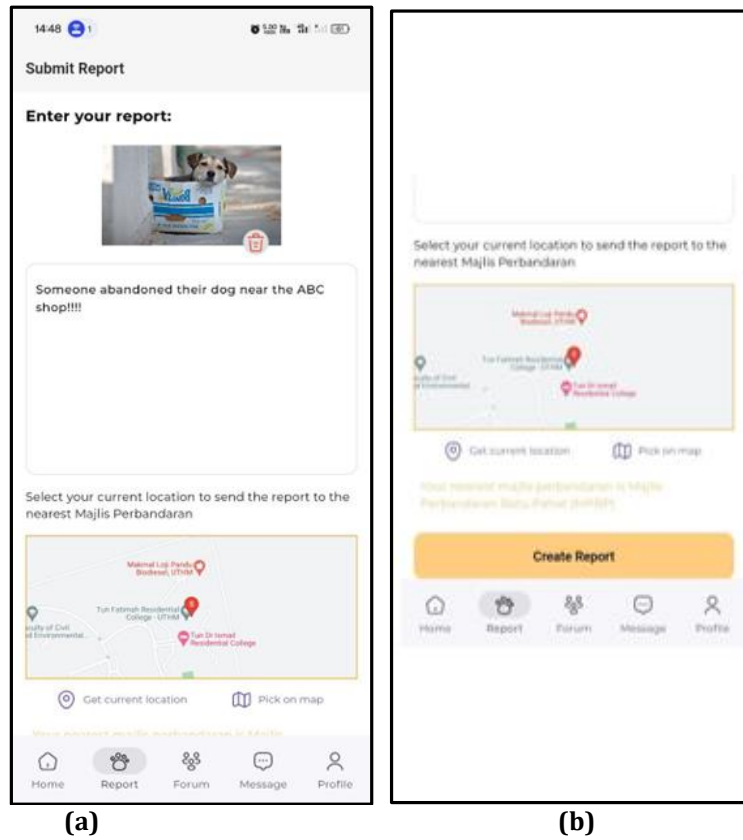


Fig. 11 (a) Interface for creating a report (b) Nearest Majlis Perbandaran is shown when the user input their location

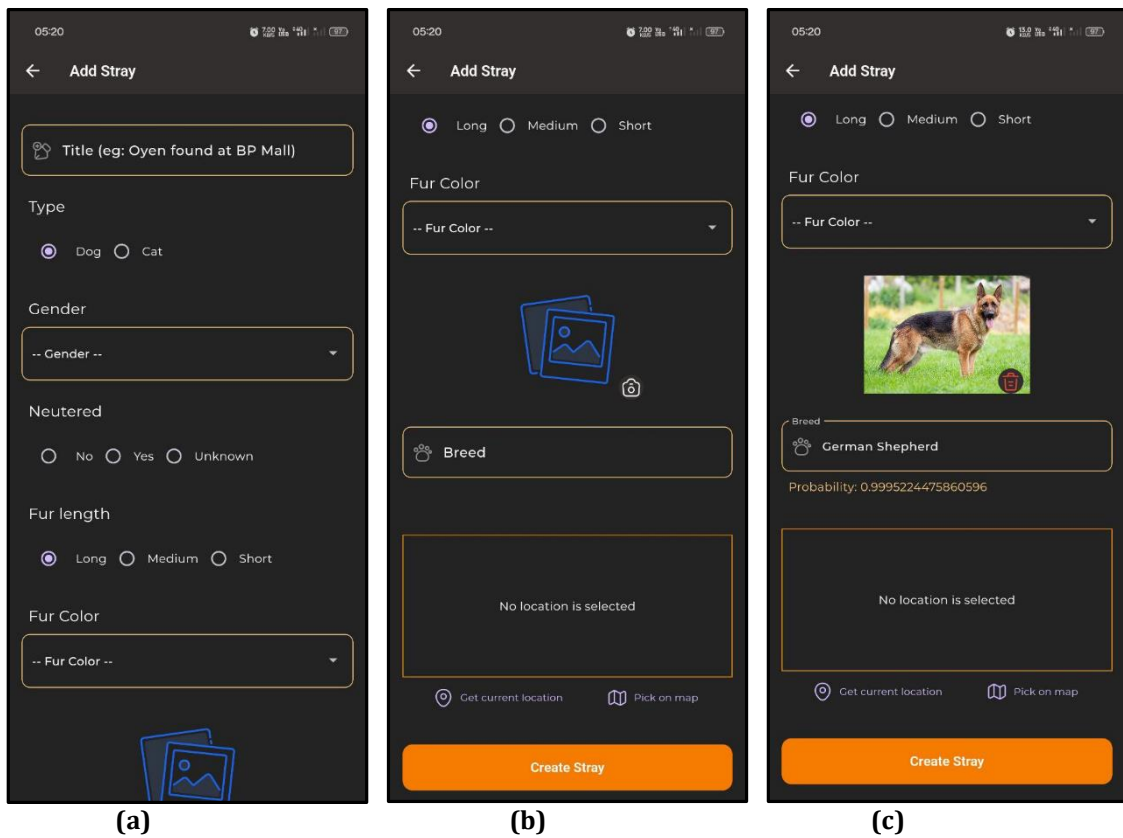


Fig. 12 (a) Interface for creating a listing in dark mode (b) Interface for creating a listing in dark mode (c) Dog breed classification result and its probability

```

Future<void> runInference() async {
  if (_labels == null || selectedImage == null || _selectedType != dog) {
    setState() {
      _tips = catTips;
    });
    return;
  }

  try {
    Uint8List inputBytes = await preprocessImage(selectedImage!);
    var input = inputBytes.buffer.asFloat32List().reshape([1, 224, 224, 3]);
    var outputBuffer = List<double>.filled(120, 0).reshape([1, 120]);

    _interpreter.run(input, outputBuffer);

    List<double> output = outputBuffer[0];
    debugPrint('Raw output: $output');

    double maxProb = output.reduce((curr, next) => curr > next ? curr : next);
    int highestProbIndex = output.indexOf(maxProb);
    String classificationResult = _labels[highestProbIndex];

    setState() {
      _probability = maxProb;
      _breedController.text = classificationResult;
      _tips = 'Probability: $_probability';
    });
  } catch (e) {
    debugPrint('Error during inference: $e');
  }
}

```

Fig. 13 Code segment for performing inference using a TFLite model

4.2 Testing

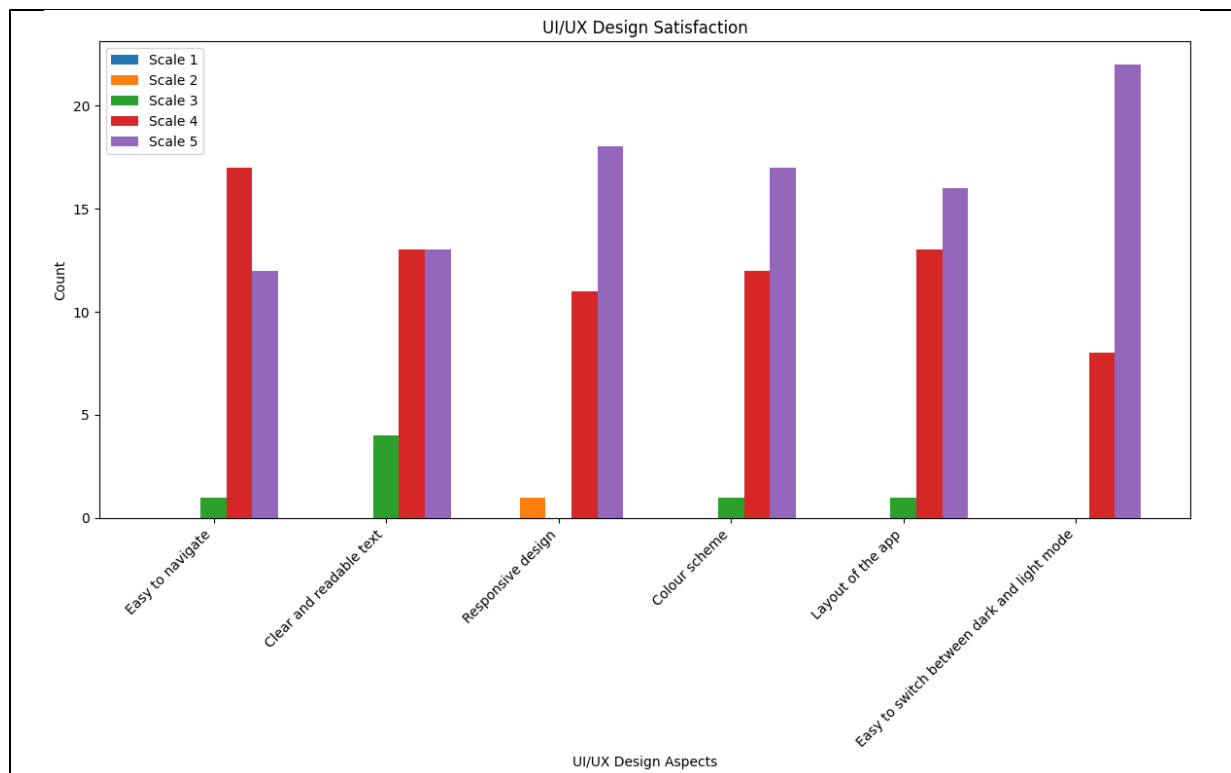
The test plan and user acceptance testing are carried out for all the features and modules provided in the FurRescue application to test and ensure each of the modules can work appropriately and ensure the satisfaction of the user. Table 5 shows the list of test cases with results, while Table 6 shows the user acceptance testing result. Table 7 shows the correspondence between the response options and the numerical values.

Table 5 List of test cases with results

No.	Test Case	Expected Outcome	Actual Outcome	Status
LOGIN TEST CASE				
1.	Login to the app using the registered email and password or Google Sign-in	Login successfully and will be redirected to the home page if the email is verified with an alert message.	As expected	PASS
2.	Login with Google Sign-In	Login successfully and will be redirected to the home page with an alert message.	As expected	PASS
3.	Login with invalid credential information	Login failed and will be redirected to the login page with an error message.	As expected	PASS
REGISTRATION TEST CASE				
1.	Register the app with all the validated data	Register successfully and will be redirected to the email verification screen with an alert message shown.	As expected	PASS
2.	Register the app with invalidated data	Registration failed, and an error message will be shown.	As expected	PASS
3.	Register the app with Google Sign-In	Register successfully and redirected to the home page.	As expected	PASS

Table 5 List of test cases with results (cont.)

No.	Test Case	Expected Outcome	Actual Outcome	Status
VIEW LISTING WITH GEOFENCE FEATURE TEST CASE				
1.	View the map	Listing that within the user's location boundary will be shown.	As expected	PASS
2.	Action trigger	Push notification when the listing is within the geofence boundary.	As expected	PASS
COMMUNICATION				
1.	Send message	The message will be sent to the user in real-time.	As expected	PASS
CREATE REPORT				
1.	Click Create Report button after filling in all the required field	The default email application will be opened, and the report will be sent to the email address of the nearest Majlis Perbandaran.	As expected	PASS
2.	Retrieve location	The nearest Majlis Perbandaran message will be shown.	As expected	PASS
FORUM				
1.	Select category	The post will be filtered and shown according to the category.	As expected	PASS
2.	Create, Edit, and delete post	The post will be created, edited, or deleted	As expected	PASS
3.	Likes or comments the post	The post will show the number of likes and comments	As expected	PASS

**Fig. 14** User acceptance of application UI/UX design

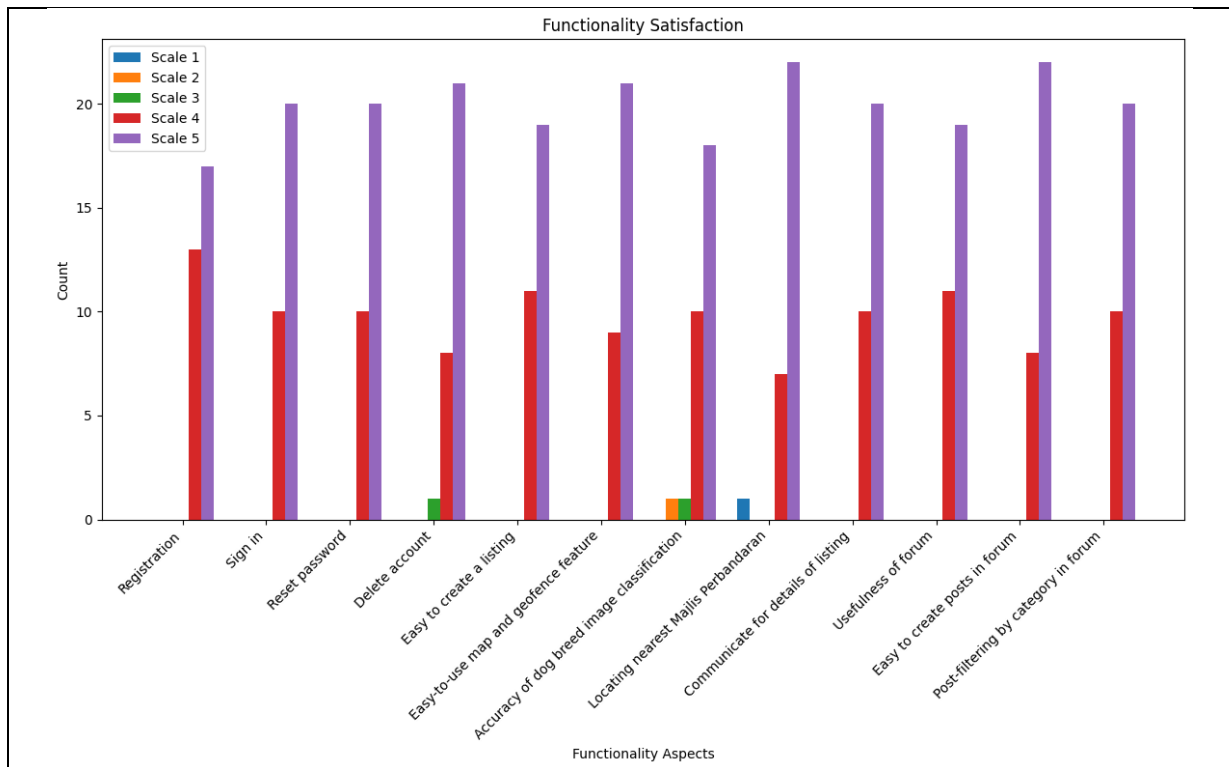


Fig. 15 User acceptance of application functionality

Table 5 shows that the application has passed all the test plans. From Figure 13, most respondents found the app's navigation to be very easy or easy, indicating a high level of user-friendliness. Feedback suggests that the text on the app is generally clear and readable. Users also agree that the app is very responsive or responsive on different devices. The colour scheme and layout of the application have also received a large majority of respondents expressing satisfaction with the design. Users agreed that the switchable between Dark Mode and Light Mode is rather very easy or easy. In terms of functionality as shown in Figure 15, most respondents agree that the application has met their expectations by giving a high scale when answering the questionnaire.

5.0 Conclusion

The FurRescue application has been proven that it has successfully met the project objectives after the testing phase was carried out. Users can effortlessly create listings anytime and anywhere, facilitating swift response to animal rescue needs. Leveraging the geofencing feature, users receive timely notifications when listings are within their designated geographic areas, streamlining the adoption process. By enabling users to showcase strays with location details, the application contributes to mitigating the stray population's impact. Although the application has been successfully developed, and the majority function of the application accomplished its objectives, the application still has several limitations. The FurRescue application does not allow users to search the location using the search bar. This means that the user can only use the application to retrieve their location or by picking the location on the map itself. The application also lacks a push notification specifically for the chat function. The user will not be able to receive the notification for the new incoming chat. The application also not be able to receive the notification if the listing is within their geofence boundary when they are not using the application. Besides that, in the forum, user not able to reply to specific comment on the post. The user is also not able to add more than one images when creating the listing. In viewing the details of the listing, user not able to zoom in the image of the listing. By identifying the limitation, the application's functionalities can be improved and ensure it remains effective and user-friendly.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

*The authors confirm contribution to the paper as follows: **study conception and design:** Yong Huei Jean, Norfaradilla Binti Wahid; **data collection:** Yong Huei Jean, Norfaradilla Binti Wahid; **analysis and interpretation of results:** Yong Huei Jean, Norfaradilla Binti Wahid; **draft manuscript preparation:** Yong Huei Jean, Norfaradilla Binti Wahid. All authors reviewed the results and approved the final version of the manuscript.*

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

- [1] S. S. Priya, "Over 1,000 dengue cases a week cause for concern in Subang Jaya," *The Star*, Feb. 24, 2023. [Online]. Available: <https://www.thestar.com.my/metro/metro-news/2023/02/24/over-1000-dengue-cases-a-week-cause-for-concern-in-subang-jaya>. (accessed: Jun. 12, 2023).
- [2] S.S. Priya and A. Leong, "Selangor plans to ban pet stores from selling cats and dogs soon," *The Star*, Jun. 12, 2023. [Online]. Available: <https://www.thestar.com.my/metro/metro-news/2023/06/12/selangor-plans-to-ban-pet-stores-from-selling-cats-and-dogs-soon>. (accessed: Jun 12, 2023).
- [3] T. Developers, *TensorFlow, Zenzo*, 2022.
- [4] Y. Shevchenko and U. Reips, "Geofencing in location-based behavioral research: Methodology, challenges, and implementation," *Behavior Research Methods*, Aug. 2023, <https://doi.org/10.3758/s13428-023-02213-2>
- [5] A. Tashildar, N. Shah, R. Gala, T. Giri, and P. Chavhan, "Application development using Flutter," *Int. Res. J. Modern. Eng. Technol. Sci.*, vol. 2, no. 8, pp. 1262-1266, 2020
- [6] I. K. G. Sudiarta, I. N. E. Indrayana, I. W. Suasnawa, S. A. Asri, and P. W. Sunu, "Data structure comparison between MySQL Relational Database and Firebase Database NOSQL on mobile based tourist tracking application," *Journal of Physics*, vol. 1569, p. 032092, Jul. 2020, <https://doi.org/10.1088/1742-6596/1569/3/032092>
- [7] "Missing Pets - Find Lost Pet - Apps on Google Play," *play.google.com*. <https://play.google.com/store/apps/details?id=com.pet.missingpets> (accessed Dec. 27, 2023).
- [8] "PeTrace - Lost Pets - Apps on Google Play," *play.google.com*. https://play.google.com/store/apps/details?id=com.pettrace.pettrace.pettrace&hl=en_US (accessed Dec. 27, 2023).
- [9] "PetFinder.my - Apps on Google Play," *play.google.com*. <https://play.google.com/store/apps/details?id=com.petfinder.andykoh&hl=en&gl=US> (accessed Dec. 27, 2023).
- [10] A. Cote, "Project Management Methodologies and Frameworks (A guide for 2024)," *Paymo*, Dec. 06, 2023. <https://www.paymoapp.com/blog/project-management-methodologies/>
- [11] H. Aman, R. Miftah, N. Murli, A. Mustapha, and M. M. Zainuddin, "Convoy Marshal: Group-Based Navigation Mobile Application," *Advanced Science Letters*, vol. 24, no. 3, pp. 1660-1665, Mar. 2018, doi: <https://doi.org/10.1166/asl.2018.11132>
- [12] GeeksforGeeks, "Object oriented analysis and design," *GeeksforGeeks*, Apr. 11, 2023. <https://www.geeksforgeeks.org/object-oriented-analysis-and-design/>
- [13] TensorFlow, "stanford_dogs," *TensorFlow Datasets*, 2024. [Online]. Available: https://www.tensorflow.org/datasets/catalog/stanford_dogs. (Accessed: Jun. 12, 2024).

Appendix A: Gantt Chart

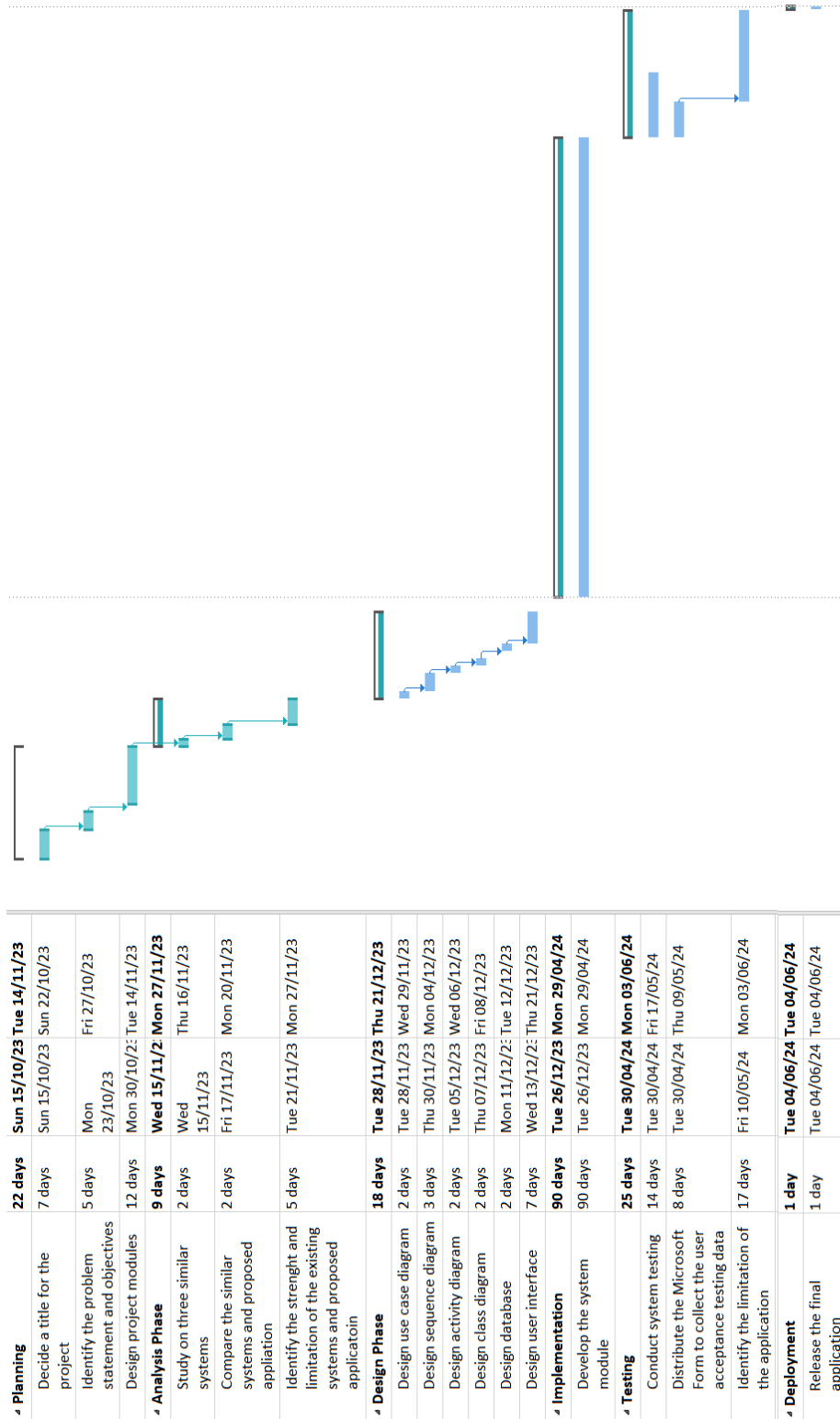


Fig. A.1 Gantt Chart

Appendix B: Use Case Diagram

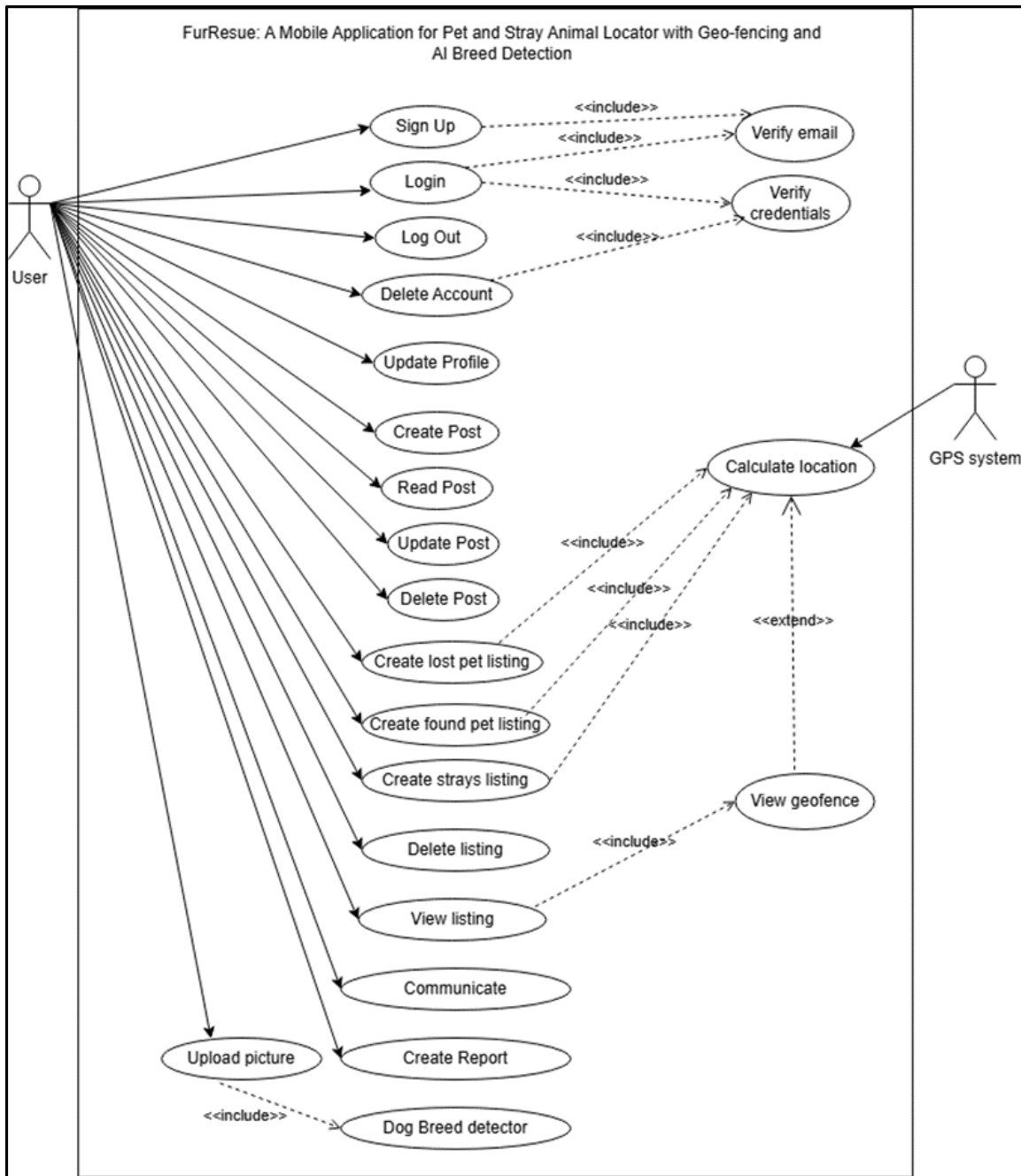


Fig. B.1 Use Case Diagram

Appendix C: Sequence Diagram

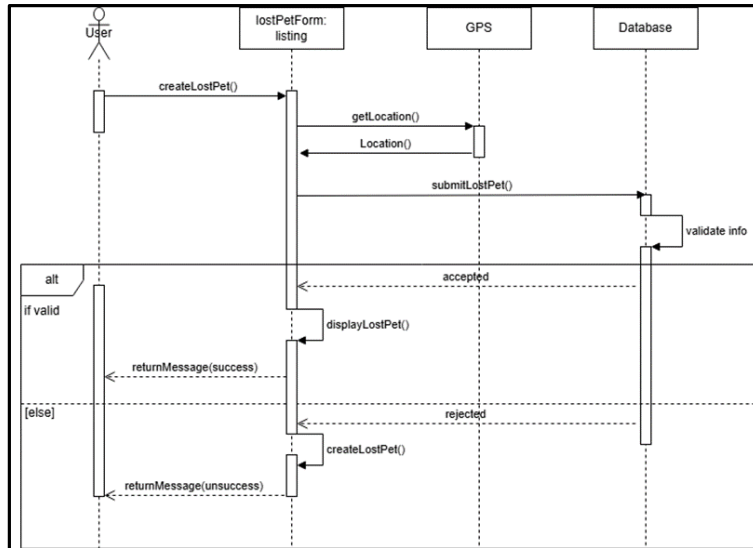


Fig. C.2 Sequence diagram for creating lost pet listing

Appendix D: Class Diagram

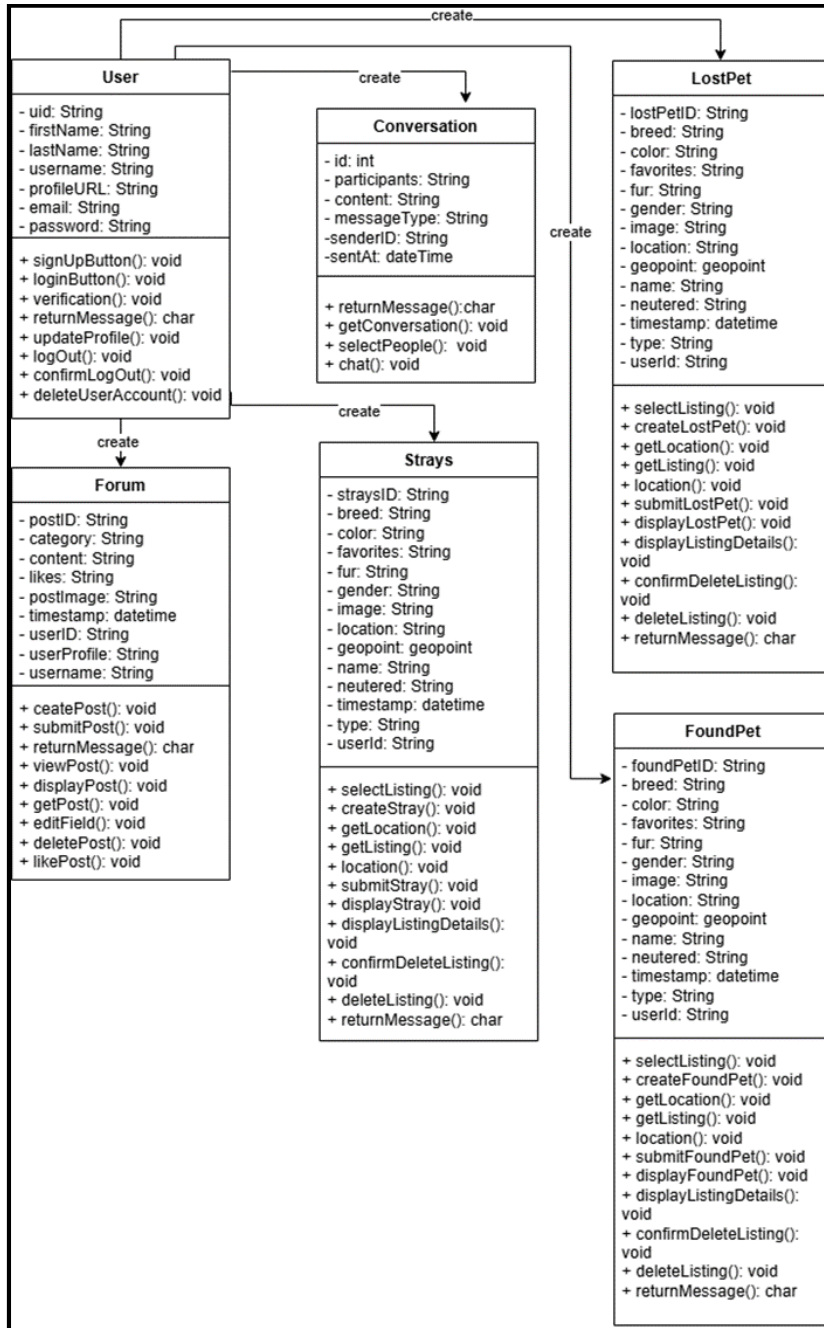


Fig. D. 1 Class Diagram