ANIMAL ADOPTION MOBILE APPLICATION

KARO MAHMOOD AHMAD

QAIWAN INTERNATIONAL UNIVERSITY

UNIVERSITI TEKNOLOGI MALAYSIA
DECLARATION OF project report

Author's full name	:	KARO ARAZ MOHAMMED			
Student's Matric No.	:	QU180040	Academic Session	:	2023-2024
Date of Birth	:	10/6/1999	UTM Email	:	Kmqu180040@uniq.edu.iq
Project Report Title	:	ANIMAL ADOPTION MOBILE APPLICATION			

I declare that this thesis is classified as:

☒ **OPEN ACCESS** — I agree that my report to be published as a hard copy or made available through online open access.

☐ **RESTRICTED** — Contains restricted information as specified by the organization/institution where research was done. *(The library will block access for up to three (3) years)*

☐ **CONFIDENTIAL** — Contains confidential information as specified in the Official Secret Act 1972)

(If none of the options are selected, the first option will be chosen by default)

I acknowledged the intellectual property in the project report belongs to Universiti Teknologi Malaysia, and I agree to allow this to be placed in the library under the following terms :

1. This is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of only.
3. The Library of Universiti Teknologi Malaysia is allowed to make copies of this project report for academic exchange.

Signature of Student:
Signature :

Full Name KARO ARAZ MOHAMMED
Date : 10 JANUARY 2024

Approved by Supervisor(s)

Signature of Supervisor :

Full Name of Supervisor
Mr. AKO Abubakr
Date : 10 JANUARY 2024

NOTES : If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction

"I hereby declare that we have read this thesis and in my opinion this thesis is sufficient in term of scope and quality for the award of the degree of Bachelor of Computer Science (Software engineering)"

Signature : 

Name of Supervisor :Mr. AKO Abubakr

Date :10 JANUARY 2024

ANIMAL ADOPTION MOBILE APPLICATION

KARO MAHMOOD AHMAD

A thesis submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Computer Science (Software engineering)

Faculty of Computing
Qaiwan international university

FEBRUARY 2024

DECLARATION

I declare that this thesis entitled *"ANIMAL ADOPTION MOBILE APPLICATION"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : 

Name :Karo Mahmood Ahmad

Date :10 JANUARY 2024

DEDICATION

I dedicate this final year project on the "Animal Adoption Mobile Application" to all the furry friends who fill our lives with joy and companionship. May this endeavour contribute in some small way to improving the lives of animals in need and fostering meaningful connections between pets and their future forever homes. Additionally, this project is dedicated to the countless individuals and organizations working tirelessly in the field of animal welfare. Your dedication and compassion inspire us to create technology that makes a positive impact on the lives of our four-legged friends.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to everyone who contributed to the realization of the "Animal Adoption Mobile Application" project. Special thanks to my project supervisor for their guidance, expertise, and unwavering support throughout the development process. I extend my appreciation to the members of my project team for their collaborative efforts and commitment. Each team member played a vital role in bringing this vision to life, and I am thankful for their dedication and hard work. I also want to acknowledge the invaluable feedback and insights provided by my peers and mentors during the project review sessions. Your constructive criticism and suggestions greatly enriched the project. Lastly, my sincere thanks go to my family and friends for their encouragement and understanding during the demanding phases of this project. Your support has been a constant source of motivation. This project would not have been possible without the collective effort and support of everyone involved, and for that, I am truly grateful.

ABSTRACT

The Animal Adoption Application is a project that aims to tackle the problem of stray animals and offers a platform for people who cannot or do not want to buy pets. The project seeks to develop an application that simplifies the process of adopting homeless animals by enabling users to search and request animals for adoption. The challenge at hand is that there are individuals who desire to adopt pets but face financial constraints preventing individuals from making a purchase. Simultaneously, people are willing to donate pets but lack a suitable platform to connect with potential adopters. This application seeks to address this dual issue by providing a solution that caters to both those seeking affordable adoption options and those looking to contribute by donating pets. This project aims to offer a digital solution that provides a lifeline to animals living on the streets or in shelters, while also catering to people who are unable to buy animals due to financial constraints or personal preferences. The application enables users to search for animals according to the individual's preferences and access detailed profiles, comprising photos and descriptions. Adoption requests can be submitted through the application by adopters. The application encourages adoption and educates users on responsible pet ownership. The project focuses on utilizing technology to tackle a significant social problem and generate a beneficial outcome for animals and society. This project aims to improve the well-being of homeless animals and facilitate the adoption process for interested individuals. The aim is to promote animal adoption and facilitate the process, ultimately leading to more animals finding caring homes.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS.

AOT	-	Ahead-of-time
API	-	Application Programming Interface
Baas	-	backend-as-a-service
CRNK	-	Cat Rescue in Kurdistan
ERD	-	Entity Relationship Diagram
HTTP	-	Hypertext Transfer Protocol
IDE	-	integrated development environment
KRG	-	Kurdistan Regional Government
PAWS	-	Protection of Animals Welfare in Kurdistan.
SDLC	-	Software Development Life Cycle
KOARP	-	Kentucky Organization for Animal Rights and Protection
UI	-	User Interface

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Animal Adoption Application is a senior project that aims to address the issue of homeless animals and provide a platform for individuals who cannot or will not purchase pets. With so many animals living on the streets and in shelters, this application connects potential adopters with animals in need of homes. The Animal Adoption Application's objective is to create an accessible and user-friendly platform where individuals can search for adoptable animals and submit adoption requests. This application seeks to connect homeless animals with compassionate individuals who are willing to provide a loving home for the animals. Individuals who are unable to purchase animals due to financial constraints or personal preferences can now explore the option of adoption using this application. The application allows users to search for animals based on criteria including species, breed, age, and location. Each user can view in-depth profiles of the animals, which include photographs, descriptions, and background and personality information. After locating an animal of interest, users can submit an adoption request via the application. The adoption request will be forwarded to the appropriate animal shelter or rescue organization, which will evaluate the applicant's suitability based on certain criteria, including their living situation, lifestyle, and ability to care for the animal.

1.2 Problem Background

Due to the fact that there are so many dogs and cats on every street, a lot of difficulties has been facing Kurdistan regarding animals, one reason being that the animals are hangers who may attack a child or another person, some people are terrified of the animals. Many people would like to keep pets at home, but due to

financial constraints or inability to afford them, they must resort to searching for animals, contacting animal finders, and facilitating the delivery of animals to those experiencing difficulties in caring for them. Through this application, users can either apply to adopt a pet or sell their animals by connecting with animal shelters and potential adopters. By providing information, resources, and fostering a community of pet owners, such an application can contribute to promoting responsible pet ownership and enhancing the well-being of animals.

With a user-friendly interface, potential adopters can easily search for available animals, view their profiles, and connect with relevant adoption agencies or organizations. Social media platforms may not have robust mechanisms for verifying the legitimacy of adoption posts or ensuring the safetyof animals being adopted. This lack of verification can lead to potential scams, misinformation, or inadequate checks on adopters, putting both the animals and adopters at risk. Providing support for animal shelters: The application can provide support for animal shelters by facilitating donations, volunteer work, and other formsof support. This can help animal shelters continue to provide care and support for animals in need. (Kanagaratham et al., 2017).

1.3 Project Aim

The aim of this project is to develop an efficient and user-friendly mobile application for animal adoption.

1.4 Project Objectives

i. To study and analyse the existing web or mobile application for animal adoption in Kurdistan Regional Government (KRG).

ii. To design and develop user mobile applications using flutter.

iii. To test and evaluate the proposed application based on the user acceptance test.

1.5 Project Scope

i. Development of a dynamic application using Flutter and Dart.

ii. Seamless integration with Firebase for streamlined data management. 3- Facilitation of animal adoption and welfare.

iii. Catering to two user categories: regular users and veterinaryprofessionals.

iv. Effortless navigation for users to discover adoptable animals and those inneed.

v. Active engagement features such as browsing available animals, sharing details about animals requiring aid, and providing feedback on veterinarycare.

vi. Addressing the issue of stray animals through a centralized database.

1.6 Project Importance

This system was chosen because there are many problems in the city with animals in the street. Some people are afraid of them because they are hangers who may attack a child or reduce Stray Population: The app can help connect pet owners who are unable to care for their animals with individuals willing to adopt them, reducing the number of stray animals on the streets. Improved Community Relations: The app can help improve community relations by providing a platform for animal lovers and veterinary shops to connect. It can also help educate the public on the proper care and treatment of animals, improving the relationship between humans and animals. In summary, the mobile application for animals that connects lost pets with their owners and helps connect animal lovers with veterinary shops can have several important benefits for animal welfare, safety, community relations, and awareness. It is an important project that can make a positive impact on the lives of animals.

1.7 Report Organization

Chapter 1 (Introduction): This chapter introduces the importance of animal adoption applications and the current issues surrounding the current approaches tohandling animal adoption. Moreover, the objective, scope and the importance of the proposed system are clarified. This chapter is about an overview of the project, and problems, working to know what would be effect of the proposed solutions and clarify objectives.

Chapter 2 (Literature Review): the chapter is about literature review of the project, case studies, and comparison between current systems.

Chapter 3 (Methodology): the chapter is clarifying methodology that has been chosen for the project, and a justification about why this methodology is chosen.

Chapter 4 (Requirement Analysis and Design): this chapter focus on design part of the system through different UML diagrams.

Chapter 5 (Implementation and Testing): the chapter is about converting the design into code, and testing the app after coding, through Test methods, like black, and white box.

Chapter 6 (Conclusion): is a conclusion about the project, and what are the achievements, goals, and future improvement suggestions.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter, a literature review is conducted to evaluate and compare existing projects in the field of animal adoption. The current project aims to develop an application for finding veterinarians and communicating with them. After surveying the literature, it was found that there are several existing adoption systems, each with its own advantages and disadvantages.

2.2 Case Study

2.2.1 Animal Adoption Application vs. Get Your Pet

The Animal Adoption Application sets itself apart by offering a broader accessibility and inclusivity for individuals interested in adopting pets, not just those who are looking to adopt from current pet owners. Unlike Get Your Pet, which primarily facilitates pet adoption directly from the previous owners, the Animal Adoption Application provides a comprehensive platform that connects potential adopters with a wide range of animals from various shelters, offering detailed profiles including photographs, descriptions, and personality information. This ensures a wider selection and a more informed adoption process. (Doland, 2024).

2.2.2 Animal Adoption Application vs. Arlington Pets

While Arlington Pets is a commendable platform for adopters within the Arlington area, the Animal Adoption Application offers a more versatile search functionality, including the ability to search based on species, breed, age, and location nationwide. This wider geographical scope makes the Animal Adoption Application a more inclusive option for users across different regions, expanding the opportunities for homeless animals to find a loving home beyond local boundaries. (Author, 2016).

2.2.3 Animal Adoption Application vs. Miwuki Pet Shelter

Miwuki Pet Shelter, with its international network of shelters, offers a broad platform for pet adoption. However, the Animal Adoption Application enhances user experience by streamlining the adoption request process directly through the application, forwarding requests to shelters or rescue organizations. This direct approach simplifies the process for users and shelters alike, potentially speeding up the adoption process and improving the match quality between pets and adopters through a more detailed evaluation of applicant suitability. (Team & Team, 2022).

2.2.4 Comparative Table

Table 2-1 Case study comparison

Feature	Animal Adoption Application	Get YourPet	Arlington Pets	Miwuki Pet Shelter
Geographical Scope	Nationwide	Direct from owners	Arlington area	International
Search Functionality	Advanced (species, breed,age, location)	Limited	Localized	Broad
User Accessibility	High (focused on financial constraints/personal preferences)	Medium	Local	High
Adoption Process	Streamlined through app	Direct owner contact	Direct shelter contact	Direct shelter contact
Pet Profiles	In-depth (photos, descriptions, background)	Detailed	Detailed	Detailed
Target Audience	Individuals unable to purchase pets	General adopters	Arlington residents	International adopters

2.3 Current System Analysis

The Kurdistan Organization for Animal Rights Protection (f) is a local, non-governmentalgroup that works to improve the lives of animals and ensure their safety across Kurdistan. Dr.Sulaiman Tamer, moved by experiences with animals seen on a daily basis in native Kurdistan, launched KOARP in 2009. Dr. Tamer's own contributions aren't the only thing keeping KOARP afloat; the organization also receives funding from grants and donations. Kurdistan does not haveany legislation in

place to safeguard animals, and there is no system in place to help domesticated or wild animals in need. The Kentucky Organization for Animal Rights and Protection (KOARP) campaigns for animal rights legislation and partners with communities to provide lifesaving veterinary services and education. Advocacy for more compassionate alternatives to the present ways of controlling stray animal numbers, such as the distribution of poisoned meat, is actively pursued (Kurdistan Organisation for Animal Rights Protection | KOARP, n.d.).

2.4 Comparison between exist

When comparing PAWS of Kurdistan and Cat Rescue in Kurdistan (CRNK), it can be seen that both organizations are dedicated to advancing animal welfare, particularly in the Kurdistan region.

2.4.1 Cat Rescue in Kurdistan – CRNK

Cat Rescue in Kurdistan (CRNK) is a fictional cat rescue organisation dedicated to the welfare and protection of cats in the Kurdistan region. CRNK focuses on rescuing stray and abandoned cats, providing them with necessary medical care, shelter, and finding them loving homes. CRNK was established with the aim of addressing the challenges faced by the growing population of stray cats in Kurdistan. The organisation's dedicated volunteers work tirelessly to rescue cats in need, provide veterinary treatment including spaying/neutering, vaccinations, and medical care for injured or sick cats. CRNK also emphasises the importance of public awareness and education regarding responsible cat ownership. They strive to educate the local community about the significance of proper cat care, sterilisation, and the prevention of abandonment.

2.4.2 PAWS of Kurdistan

PAWS of Kurdistan is a fictional animal welfare organization based in the Kurdistan region. The acronym "PAWS" stands for "Protection of Animals Welfare in Kurdistan." The organization is dedicated to promoting and safeguarding the welfare of animals, including both domestic and wild animals, in Kurdistan. PAWS of Kurdistan focus on various aspects of animalwelfare, including rescue, rehabilitation, advocacy, and education. Here are some key areas of their work: Animal Rescue: PAWS of Kurdistan work tirelessly to rescue and provide care for animals in need. They respond to reports of injured, abused, or abandoned animals and strive to ensure their safety and well-being. (Pawsofkurdistan | Pawsofkurdistaninternational, n.d.).

i. Information Accessibility: A website provides a centralized platform where detailed information about the organization's adoption process, requirements, available animals, and theirprofiles can be easily accessed. This helps potential adopters make informed decisions and find the right companion for their home.

ii. Communication and Contact: A website typically includes contact information, such as phonenumbers or email addresses, allowing interested individuals to reach out to the organization withinquiries or adoption inquiries. It streamlines communication and ensures prompt responses to potential adopters.

iii. Collaborating with Existing Animal Welfare Organizations: Partnering with established animal welfare organizations that have websites or adoption platforms can help expand the reach and provide a more comprehensive adoption process.

iv. Community Outreach and Events: Conducting adoption events and outreach programs in local communities can increase awareness and provide direct opportunities for potential adopters to interact with animals in need of homes.

2.5 Proposed System

The reporting feature can be very helpful in reuniting lost animals with their owners. It can provide a quick and easy way for people to report a lost or found animal and potentially increase the chances of finding the animal's owner. Additionally, having a central platform for lost and found animal reports can help reduce the time and effort required for pet owners and animal shelters to connect with each other.

The veterinary shop feature can also be very beneficial for pet owners. Finding a reliable and trustworthy veterinary shop can be challenging, especially in a new area. Having a platformthat allows users to search for veterinary shops in their vicinity, check their availability and services, and read reviews from other pet owners can help make the process much easier.

Table 2-2 System proposal

	PAWS	CRNK	KOARP
User friendly	x	√	√
search	x	x	x
Tracking	x	x	x
Contact	√	√	√
Profile	x	x	x

This table illustrates various websites and social media platforms that encounter several issues. For instance, they lack user-friendliness and fail to provide platforms for showcasing animals online. While they maintain social media accounts, the absence of animal search and tracking functionalities is notable. Should any issues with animals arise, individuals may contact or message them through their social media channels. The second and third platforms possess websites but lack animal search and tracking capabilities, thereby limiting users' abilities to find animals.

Furthermore, these platforms primarily assist animals found on the streets, neglecting the search, retrieval, and presentation of animals to potential adopters lacking online presence.

In contrast, a mobile application for animal adoption offers robust search functionality, real-time tracking, streamlined contact options, an intuitive user interface, and comprehensive animal profiles. These features significantly enhance accessibility, user experience, communication, and the adoption process, thereby increasing the likelihood of successful and meaningful animal adoptions.

2.6 Literature Review of Technology Used

The system is a Mobile-based application. It is an Android application. For the application, Visual Studio is used for Flutter and Dart language coding, and for the database, XAMP program is utilized with Firebase.

Swift (iOS): Swift is the primary programming language for developing native iOS applications. It is powerful, intuitive, and has a modern syntax. It is recommended for iOS app development if targeting Apple devices.

2.6.1 Swift Frontend

Swift UI is a modern and declarative framework introduced by Apple for building user interfaces across all Apple platforms, including iOS, macOS, watchOS, and tvOS. It offers a wide range of built-in UI components, layout options, animations, and powerful tools for creating interactive and visually appealing interfaces. SwiftUI is gaining popularity due to its simplicity, code reusability, and the ability to preview and iterate designs quickly. It is recommended for new projects or applications that target the latest Apple platforms. (Artemov, 2024).

2.6.2 Swift Backend

Vapor is a widely used server-side Swift framework for building web applications and APIs. It provides a robust set of tools, APIs, and middleware for handling routing, data management, authentication, and other backend tasks. Vapor leverages Swift's type safety and performance to deliver efficient and scalable server-side solutions. It has a vibrant community, good documentation, and strong support for asynchronous programming paradigms. Vapor is a popular choice for building backend services and can be used in conjunction with other Swift frameworks like SwiftUI for full-stack Swift development.

Kotlin (Android): JetBrains developed Kotlin, a modern programming language. It is the preferred language for Android app development as it is fully compatible with Java and offers enhanced features and better safety.

2.6.3 Kotlin Frontend

Kotlin/JS: Kotlin can be transpired to JavaScript using the Kotlin/JS compiler, enabling the writing of Kotlin code that runs in web browsers. Kotlin/JS can be utilized with popular frontend frameworks like React or Vue.js. (Kotlin Programming Language, n.d.).

2.6.4 Ktor Backend

Ktor: Ktor is a lightweight, asynchronous web framework for building backend applications in Kotlin. It provides a simple and intuitive API for handling routing, HTTP requests, authentication, and more.

React Native: React Native is a cross-platform framework developed by Facebook. It allows developers to build mobile applications using JavaScript and

offers a "write once, run anywhere" approach. React Native is suitable for developing apps that can run on both iOS and Android platforms.

2.6.5 React Native Frontend

JavaScript (ES6+): React Native utilizes JavaScript as the primary language for writing frontend code. With React Native, native mobile apps can be built using JavaScript and the React library. (React Native · Learn Once, Write Anywhere, n.d.).

2.6.6 Node.js Backend

Node.js: Node.js is a JavaScript runtime that enables server-side JavaScript execution. It is commonly used as a backend technology in the React Native ecosystem. With Node.js, y build server-side logic can be built and handle backend operations such as API integration, data processing, and database interactions.

Flutter: Flutter is an open-source UI framework developed by Google. It enables developers to build native-like mobile applications for iOS and Android using a single codebase written in the Dart programming language. Flutter provides a rich set of pre-designed UI components and offers excellent performance.

2.6.7 Dart Frontend

Dart: Dart is the main language used for writing frontend code in Flutter. It provides a modern and concise syntax, along with a rich set of libraries and frameworks specific to Flutter development. With Dart, the UI components can be built, handle user interactions, and manage application state in Flutter.

2.6.8 Firebase Backend

Firebase is a comprehensive backend-as-a-service (BaaS) platform provided by Google. It offers a range of cloud-based services such as authentication, real-time database, cloud storage, cloud functions, and more. Firebase provides easy integration with Flutter and simplifies backend development by offering pre-built features and infrastructure.

This below table showing the comparing of five key functionalities for mobile application development using Swift, Kotlin, React Native, and Flutter.

Table 2-3 Comparison of Existing technologies

Functionality	Swift	Kotlin	React Native	Flutter
Native UI	Yes	Yes	Yes	Yes
Code Sharing	Limited (iOS, macOS)	Limited (iOS, Android)	High (iOS, Android)	High (iOS, Android)
Performance	High	High	Moderate	High
Ecosystem	Rich	Growing	Large and Mature	Growing
Third-party APIs	Extensive	Good	Moderate	Growing

To figure out which one is the best when it comes to creating mobile applications, The advantages and the advantages of each needs to be studied:

Swift: Swift has a rich ecosystem and extensive third-party APIs, making it a powerfullanguage for iOS development. However, it is limited to iOS and macOS platforms and does not have native support for Android.

Kotlin: Kotlin offers good performance and access to a wide range of third-party APIs.However, its code sharing capabilities are limited, making it challenging to share codebetween iOS and Android platforms.

React Native: React Native provides a high level of functionality and access to a large ecosystem. However, it may have limited access to certain native APIs, requiring additional customization for platform-specific features.

Flutter: Flutter has a strong ecosystem and extensive third-party APIs. It allows for high code sharing between iOS and Android platforms, making it an good choice.

2.7 Chapter Summary

In this chapter, the literature review related to the current system has been dissuaded, including both the positive and negative sides. Additionally, the other articles and systems that are associated with our system has also been discussed. Also, further discussion was made on the technologies that are going to be used in the system, as well as those used in other systems. In summary, the current system has been explored, other related articles and systems, and the technology that could be used in the system created. This information has implications for the system's development and implementation.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

By defining actions to be performed and techniques to apply to the development process and product, it is possible to improve control and management of the system development process and simplify and standardize it. Iimproving the quality, organization, and usability of your system can be achieved through the implementation of a methodology. The purpose of selecting the methodologies and the best place for animals will be discussed in this chapter. The discussion also includes the tools and technologies that are used to analyse the system.

3.2 Methodology Choice and Justification

Software companies and developers utilize a wide range of development methodologies for different projects. The selection of an appropriate methodology is crucial, as it aims to facilitate a seamless development process and effectively address the project requirements. While each methodology has its own set of pros and cons, choosing the right one plays a pivotal role in successfully implementing a software project.

3.3 Agile Methodology

Software development is done using a cycle called the Software Development Life Cycle, or SDLC. Planning, requirement gathering, design development, testing,

and deployment are the other four phases. The software development life cycle has several facets and paradigms (waterfall, iterative, agile, etc.).

A sequential development process known as "waterfall" begins withrequirements, moves through design, implementation, testing, and deployment. This methodology requires that each step be finished before going on to the next. For instance, the needs of an individual must be created before implementing them, and developing something else concurrently is not feasible during this process.

Software development using the agile process is iterative and incremental, with the requirement for flexibility to adapt to customer needs. It helped with time management, iterative development, and adaptable planning. Throughout all phases of the developmental life cycle, interactions are encouraged and prescribed by this theoretical framework. Agile allows for regular testing while also giving end users, stakeholders, and the business the chance to offer input on their work-in-progress. Agile development helps you to remain adaptable and make quick adjustments when business demands and user needs change often.

In comparison to the other techniques, agile methodology has been picked since it allows for more dependability. For instance, using the waterfall methodology, modifications to the system must wait until all stages have been completed before they can be made. In agile, there are sprints in which the person will create the project. For instance, if the agile life cycle has six stages, sprints would be created for each of them. In each sprint, there is a time limit during which the person must complete a work in order to receive the product for the final time. Feedback is received for each sprint, depending on the feedback, changes in your project can easily be made.

3.4 Advantages and Disadvantages of agile

This table showing advantages and disadvantages of agile

Table 3-1 Agile Advantages and Disadvantages

Advantages	Disadvantages
Flexibility: Agile allows for frequent iterations and welcomes changes throughout the development process, making it well-suited for projects with evolving requirements.	Lack of Predictability: The adaptabilityof Agile can make it challenging to predict project timelines, especially when requirements are subject to frequent changes.
Customer Involvement: Agile methodologies prioritize customer collaboration and feedback, ensuring that the final product meets their needsand expectations.	Dependency on Team Communication: Effective communication is vital in Agile, and a lack of it can hinder progress and coordination among teammembers.
Rapid Delivery: By breaking the project into smaller, manageable increments called sprints, Agile enables quicker delivery of working software.	Resource Requirements: Agile methodologies often require dedicated team members and adequate resourcesto support frequent iterations and deliverables.
Enhanced Quality: Regular testing and continuous integration in Agile help identify and address issues early, resulting in higher overall software quality.	Scope Creep: Without proper control and management, Agile projects may be susceptible to scope creep, where new requirements are continuously added, potentially impacting project timelines and budgets.

3.5 System Requirement Analysis

The agile methodology has six phases, as shown in Figure 3.1, the phases are requirement (plan), design, develop, test, evaluate, and feedback (meet)

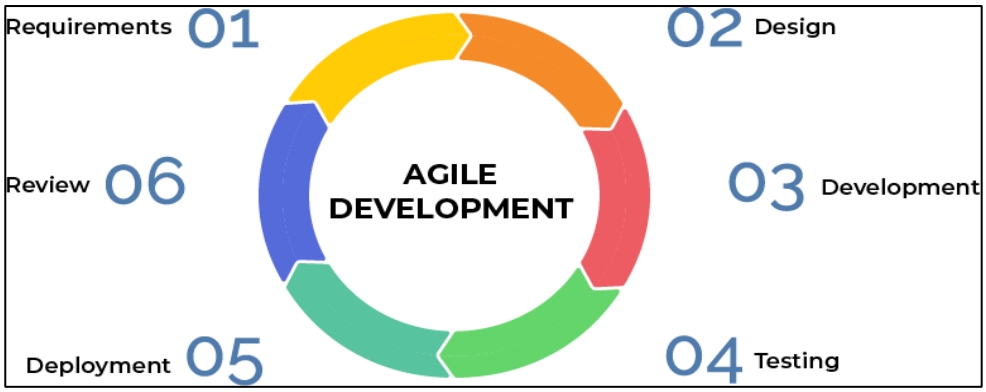

Figure 3-1 Agile Development

3.5.1 Requirement (plan)

As part of this phase, the developer collects the requirements from the stakeholders during a meeting during which the stakeholders discuss all the requirements that need to be incorporated into the system. Stakeholders will prioritise some functions that must be included in the project during this phase.

3.5.2 Design

It is the second phase of the project where they will receive the requirements, discuss them, manage them, prioritize them, and then discuss the tools that will be used and the programming languages that will be used.

3.5.3 **Development**

When the developers agree on the design phase plan, they will make sprints for improving the current system's design and separate the sprints. To finish the development phase on time, they will create a sprint duration.

3.5.4 **Testing**

Having your quality assurance team test the software product before releasing it is essential to ensuring that it operates properly and fulfills its intended function. The testing process may also help with the resolution of any significant user experience or security issues. This stage is completed in test-driven development before the product is made accessible to users, or it even starts before coding.

3.5.5 **Evaluate**

The developer will do some evaluations prior to putting the results of the working iteration into production once all steps have been completed and testing is complete.

3.5.6 **Feedback (meeting)**

At the final stage, when the system has been put into production, developers will meet with stakeholders and present the system to them. Any feedback provided will be taken into consideration, and necessary changes will be made. If no feedback is received, the system will be published for users.

3.6 Technology used in the developing system

In this article, the technologies that were utilized will be discussed to create the Best Place for Animals application and their advantages for the system.

3.6.1 Visual studio Code

Microsoft's integrated development environment (IDE), known as Microsoft Visual Studio, is used to construct a range of software, including websites, web applications, online services, and mobile apps. Complementary tools, compilers, and other features are included in Visual Studio to aid in the software development process. It supports a variety of coding languages, and for this project, Visual Studio is utilized to implement all of the code in the IDE.

3.6.2 Flutter

Flutter also offers a flexible and powerful state management system that enables developers to manage the application state in a scalable and efficient way. Additionally, Flutter provides a range of tools for debugging and testing applications, and supports both unit and integration testing. Overall, Flutter is a powerful and popular framework for developing mobile applications that offers a fast, efficient, and flexible approach to cross-platform development. (Flutter - Build Apps for Any Screen, n.d.).

3.6.3 Dart

Dart also supports a number of features that make it easy to write clean and maintainable code, such as optional named and positional parameters, a concise syntax for defining classes and functions, and support for extension methods. The language

also provides a number of tools and libraries that make it easy to build and deploy applications, including package management tools, an ahead-of-time (AOT) compiler, and a virtual machine for running Dart code. Overall, Dart is a modern and powerful programming language that offers a range of features and tools for building high-performance, scalable, and maintainable applications for web, mobile, desktop, and server environments. (Dart Programming Language, n.d.).

3.6.4 **Firebase**

Using Firebase as the backend for a Flutter application brings numerous benefits. Firebase offers a real-time NoSQL database, enabling instant data synchronization across clients. This is particularly valuable for applications requiring real-time updates and collaborative features. Firebase also provides seamless authentication and user management services, simplifying the process of registering and authorizing users. Additionally, Cloud Firestore, Firebase's scalable cloud-based database, offers powerful querying capabilities and offline support. Cloud Functions allows the person to extend their application's functionality with serverless computing, triggered by database changes or HTTP requests. Firebase's Cloud Storage securely stores and serves user-generated content, while analytics and performance monitoring tools offer valuable insights. Firebase Hosting simplifies deployment, and Firebase App Distribution facilitates app testing. Integration with other Firebase services further enhances the capabilities of your Flutter application. Overall, Firebase's features and services streamline backend development, enabling developers to focus on creating exceptional user experiences.

3.7 Hardware and Software Requirement Analysis

The system requirements for every system are hardware and software. Hardware devices are those physical devices that give input and output based on the memory and processor devices in them. Software is a set of instructions for a computer to do specific tasks, such as coding and building a system.

3.7.1 **Hardware Requirement for Samsung**

The hardware requirements for an Samsung, as outlined in Table 3.2, set the minimum specifications for optimal performance and functionality.

Table 3-2 Samsung Hardware

Hardware	Minimum Specification
Processor	Snapdragon 675
Random Access Memory	4 GB
Hard Drive	128 GB
Operating System Architecture	64 Bit
Display	5.7″ HD1080p
Version	Android 9.0

3.7.2 **Hardware Requirement for iPhone**

The hardware requirements for an iPhone, as outlined in Table 3.3, set the minimum specifications for optimal performance and functionality.

Table 3-3 iPhone Hardware

Hardware	Minimum Specification
Processor	A13
Random Access Memory	4 GB
Hard Drive	128 GB
Operating System Architecture	64 Bit
Display	4.7″ HD 1080p

Version	IOS 15

3.8 Chapter Summary

As a conclusion, this chapter explains the methodology used to develop Best Place for Animals. Each phase of the methodology was explained and well understood, as was the justification for the system. The relationship between the requirement phase and the designing phase and how the design phase works for other faces Hardware and software requirements and perspectives are clearly explained for the technologies used in the system.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

This chapter will cover the study and design of the ideal animal habitat. The use case diagram, activity diagram, and sequence diagram will be used to explain the roles ofeach user in the program. The UML class diagram and database architecture will be usedto illustrate how the system interacts with one another. The high-resolution prototype of the user interface will also be included in the chapter.

4.2 Requirement Analysis

The requirement analysis will explain the application's functionality as well as the customers, veterinarians, and admin users. The administrator is in charge of managing theusers, and clients may view the list of veterinarians, get their contact information, and consult with them. The doctors will respond to the patients' questions.

4.3 Use Case Diagram

Use case diagram is showing the interaction between the actors which are the users and the system. Use case Diagram specifying the requirements. This Use Case Diagram isfor best place for animals it shows the interaction between the Actors and the application.

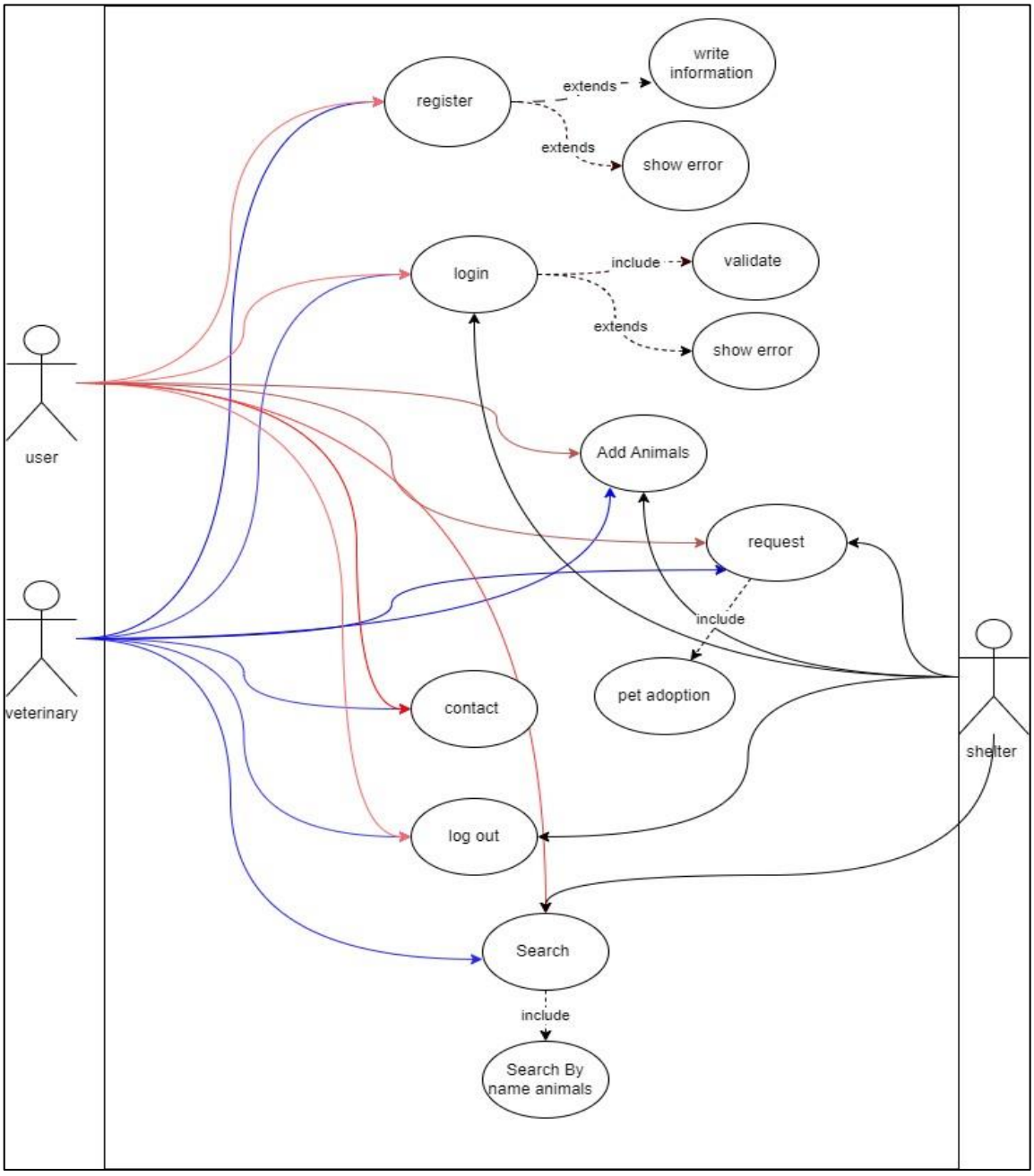

Figure 4-1 Use Case Diagram

Based on the use case diagram shown in Figure 4.1, customers are permitted to register, login to the application, and subsequently view veterinary doctors. Additionally, they can conduct searches, which encompass main types of pets. Customers also have thecapability to contact doctors. On the other hand, doctors are permitted to register, login tothe system, and communicate with their customers.

4.4 Database Design

The Entity Relationship Diagram (ERD) represents the relationships between entity sets stored in a database. The provided (Figure 4.2) illustrates the ERD designed for an animal adoption system, typically used with MySQL databases. However, the application in question utilizes Firebase as its database, which may introduce certain differences in database structure and functionality. Despite these differences, the figure serves as a visual tool to explain the database design and relationships within the animaladoption application. It provides a clear overview of how entities are connected and organized within the system.

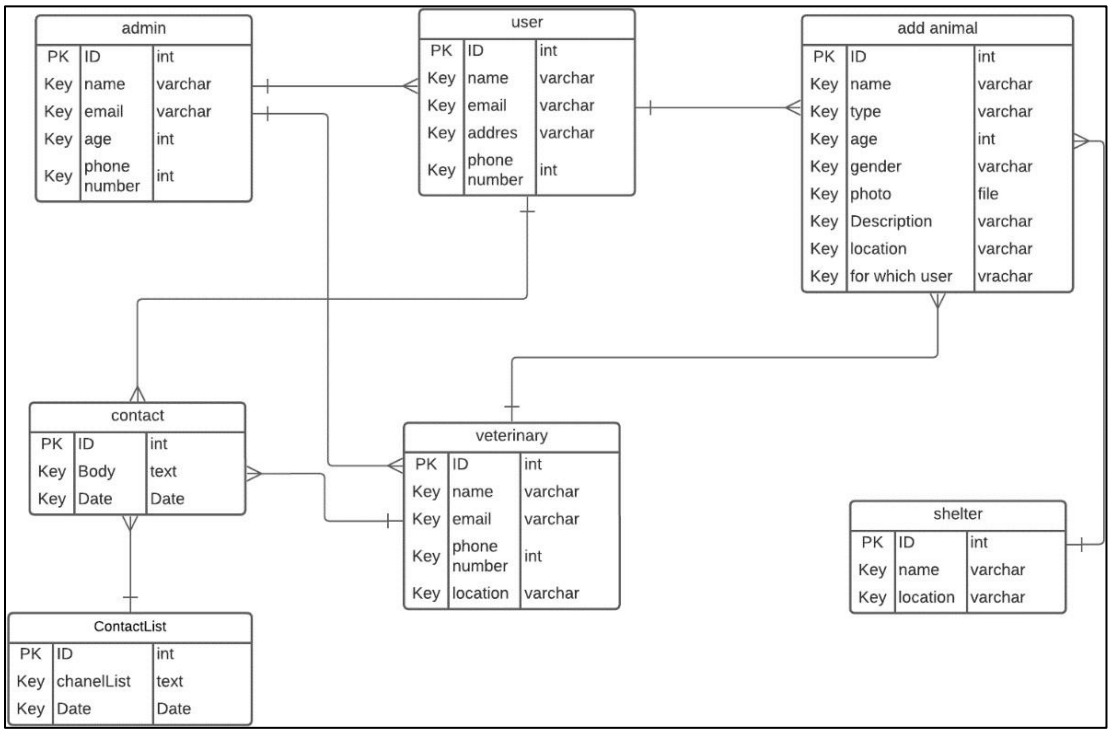

Figure 4-2 ERD designed

4.5 Class Diagram

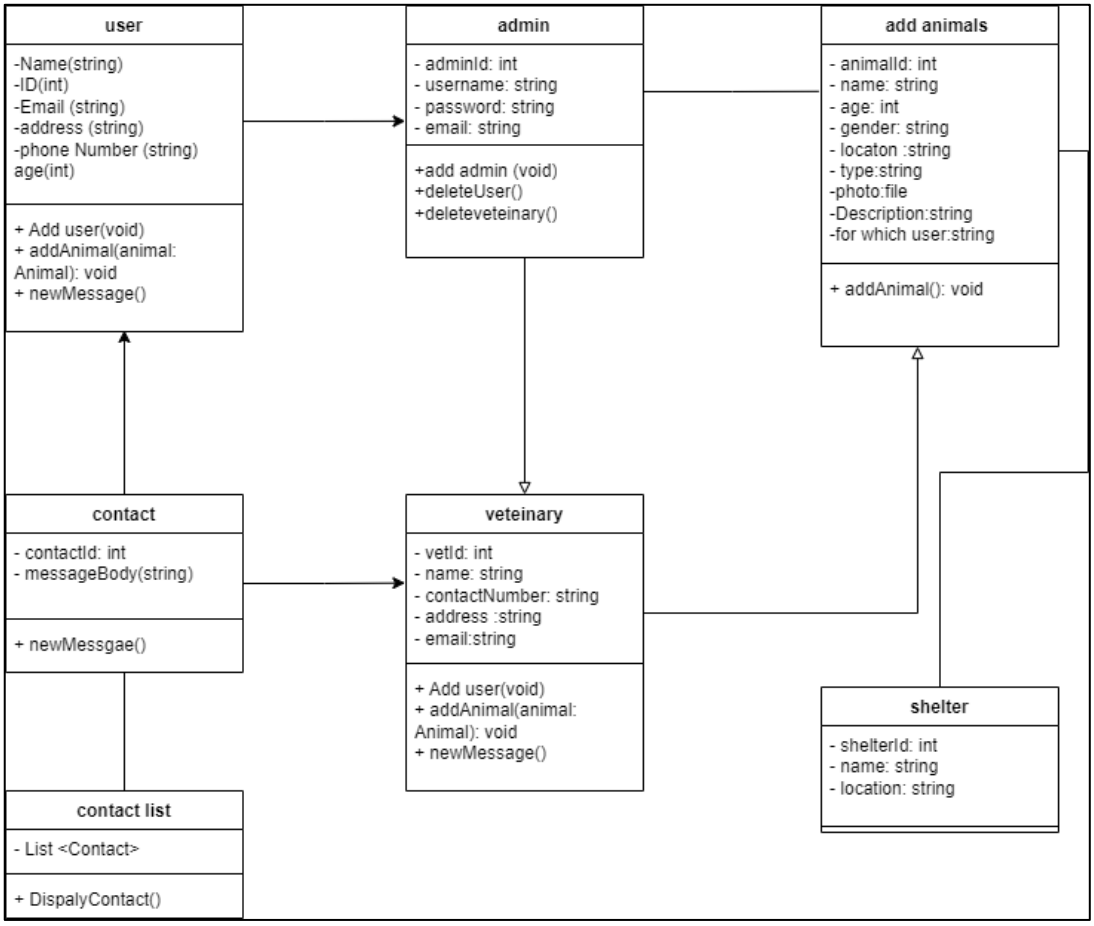

Figure 4-3 Interface

4.6 Interface Design

These shown figures are the design of the interface for the animal's adoption:

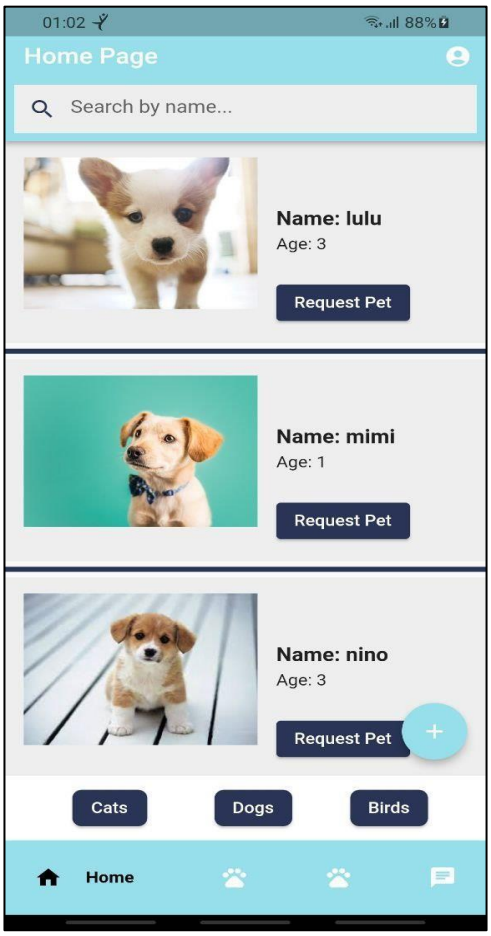

Figure 4-4 Interface Home Page

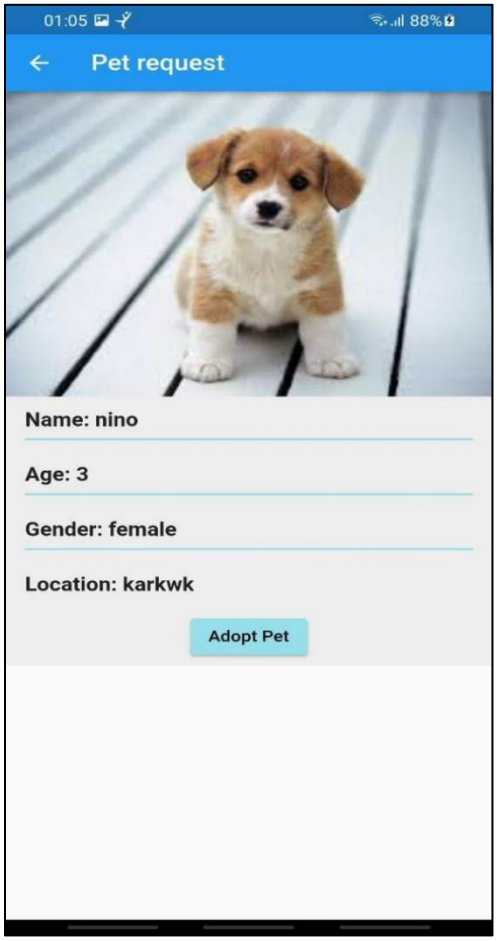

Figure 4-5 Interface Pet Request

In Figure 4.4, can observe the interface of the home page of the application. On this page, you will find listings of pets available for donation, and you have the option torequest a pet.

In Figure 4.5 displays a page containing detailed information about the pets you're interested in. If you wish to adopt a pet, simply click on "adopt pet" to send a request for donation.

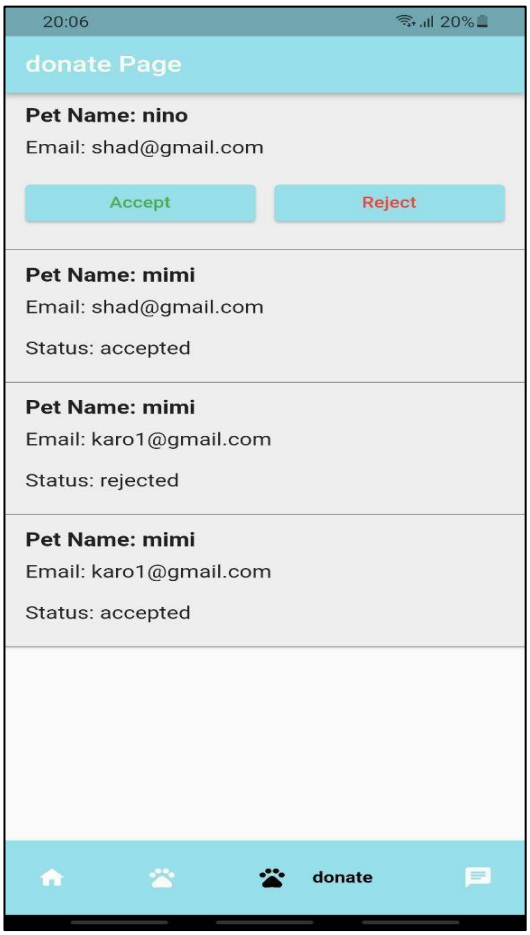

Figure 4-6 Interface Donate Page

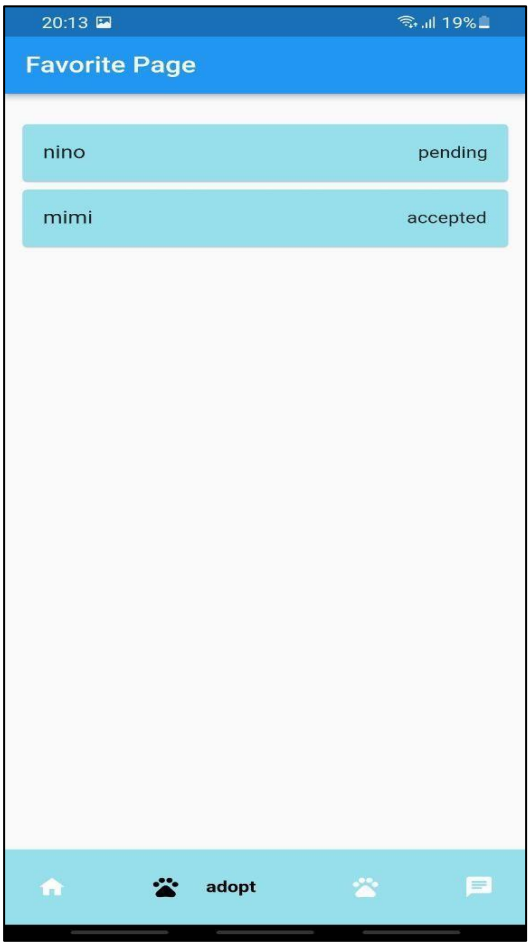

Figure 4-7 Interface Requested Page

In Figure 4.6, on this particular page, you can view the donations made for you to add apet to this application. The owner of the pet has the authority to either accept or reject the request.

In Figure 4.7, on this page, you can view the pets you've requested.

4.7 Achievements

The Animal Adoption Application, developed as a senior project, represents an innovative solution to the issue of homeless animals. The primary goal was to establish a user-friendly platform connecting animals in need with individuals seeking to adopt. The application not only facilitated the adoption process but also aimed to raise awareness about responsible pet ownership. Achievements include the creation of a meaningful impact on pet overpopulation, diminishing shelter populations, and promoting adoption over purchase. The application's educational component ensured adopted animals received proper care. Overall, the project contributed to the well-being of animals and the community, fostering a culture of responsible pet ownership and societal awareness.

4.8 Chapter Summary

In the chapter describes the System analysis and design for the Find my doctor. Many diagrams created to explain the analysis and design in many ways. Use case, activity diagram, Class diagram and Entity Relationship Diagram (ERD)has been created which is useful for the next chapters in PSM2 for implementations for the application.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

The focus of this chapter will centre on the implementation and testing of the project. As the project encompasses key functions and features, there will be detailed explanations and examples of the code for each system. The stages involved in this phase include coding and testing.

5.2 System's Primary Function

The primary function of the system revolves around pet adoption through a mobile application. Users are able to add pets for adoption and submit requests for adopting pets. Following a request, the owner of the pet can either accept or reject the adoption request. Additionally, the system facilitates communication through a chat feature, allowing users to engage in conversations with both other users and veterinary professionals.

5.2.1 Home Page

The home page serves as the central hub, showcasing pet listings anthor by users. It prominently displays pets categorized under different categories such as dogs, cats, and birds. Users can navigate through these categories to explore available pets. The homepage also features a search functionality, enabling users to find specific pets

based on type or name. For those who own pets and wish to contribute to the application, there is an option to add their pets directly through the home page.

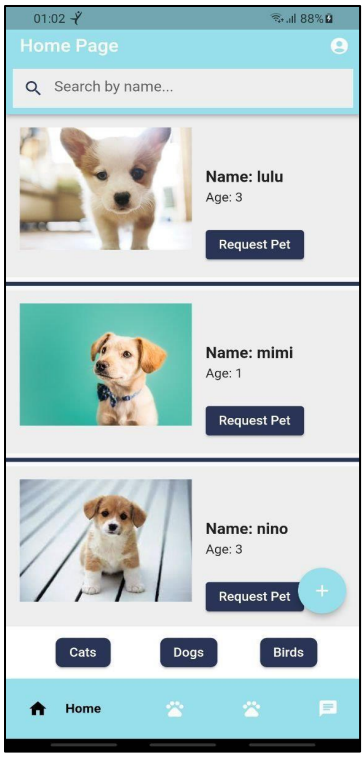

Figure 5-2 Interface

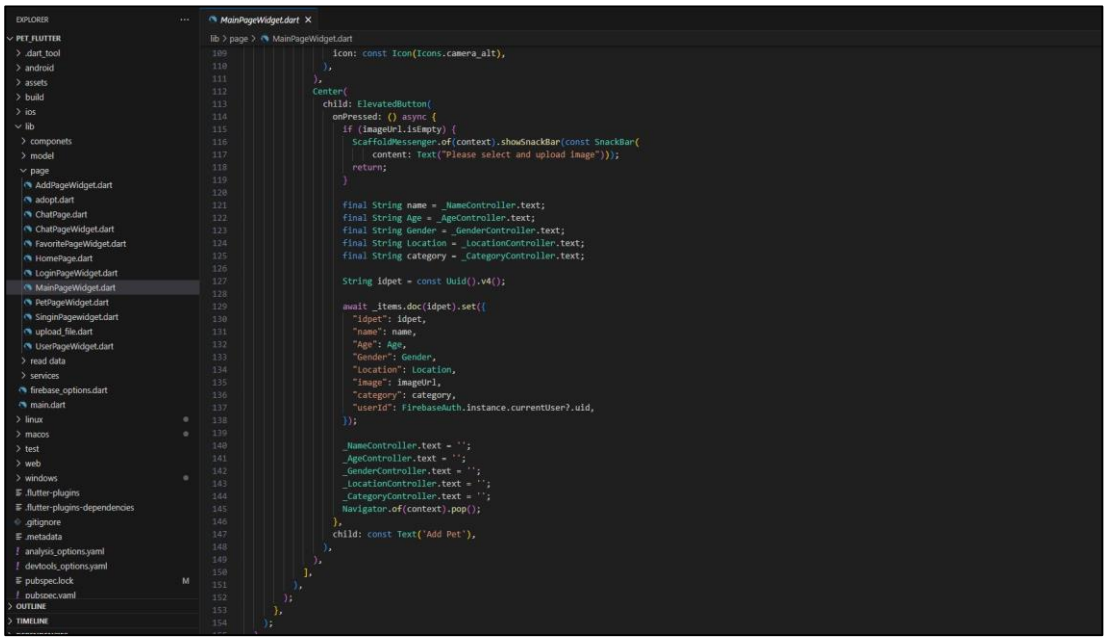

Figure 5-1 code part

5.2.2 Donate Page

The donation page is designed to display requests for the adoption of pets that users have listed in the application. On this page, users can review and manage incoming adoption requests for their listed pets. The options to either accept or reject each request are provided, giving users control over the adoption process for their pets.

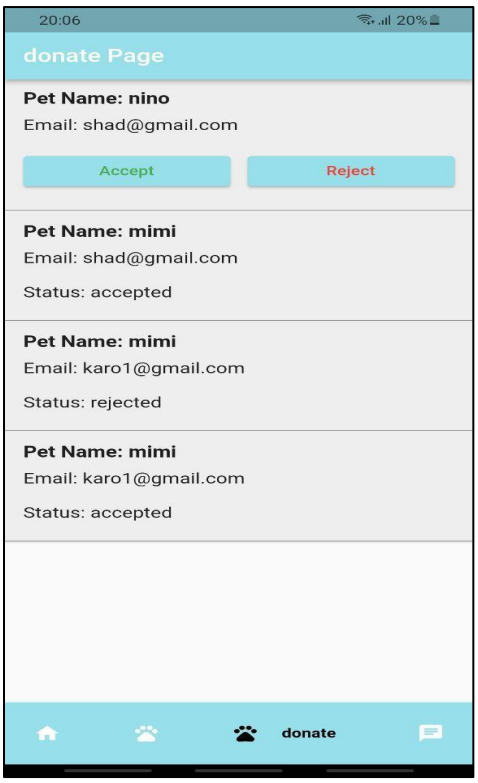

Figure 5-3 Interface

5.2.3 Adopt Page

Every request has a status of either pending or comes with an accept or reject option. This page offers a simple and well-organized interface for users to handle and reply to adoption requests for the pets they have listed. Its purpose is to expedite the

adoption process. The ability to accept or reject requests offers users a simple way to decide on the adoption of their pets in an informed manner.

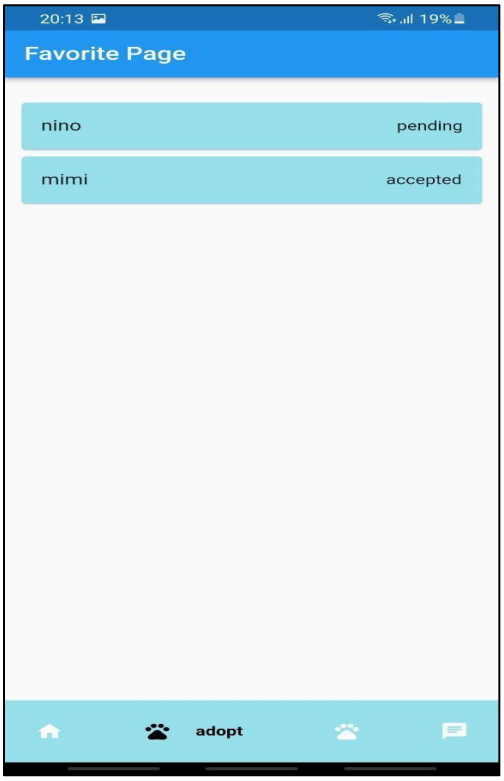

Figure 5-4 Interface

5.2.4 Chat Page

The purpose of the chat page is to enable two different user types to communicate with each other: general users and veterinary professionals. With the help of this feature, users can converse with each other according to their roles. The page facilitates easy communication by enabling message exchanges between regular users and veterinary professionals. In order to meet the specific needs and facilitate interactions between these two user categories within the application.

Figure 5-5 Interface

Let's outline two separate testing approaches: one involving the creation of a use case diagram before the application development, and the other involving questions for users of the application.

The diagram those Test Case is before the create the application the and testing aboutUser Registration:

Ensures the use case diagram outlines scenarios related to user registration, covering successful and unsuccessful registrations, as well as password recovery.

Adding a Pet:

Examines the use case for adding a pet to verify its comprehensiveness, encompassing all necessary steps, including variations like adding multiple pets.

Adoption Process:

Reviews the use case diagram for the pet adoption process, checking for scenarios related to viewing available pets, submitting adoption requests, and owner responses.

Search Functionality:

Examines the use case for searching for pets based on type or name, ensuring it covers various search scenarios and filters.

5.3 User Acceptant Test

This project has undergone evaluation and testing through a user acceptance test questionnaire. The application was distributed to multiple users within our university, and the questionnaire was diligently distributed for their feedback.

The result:

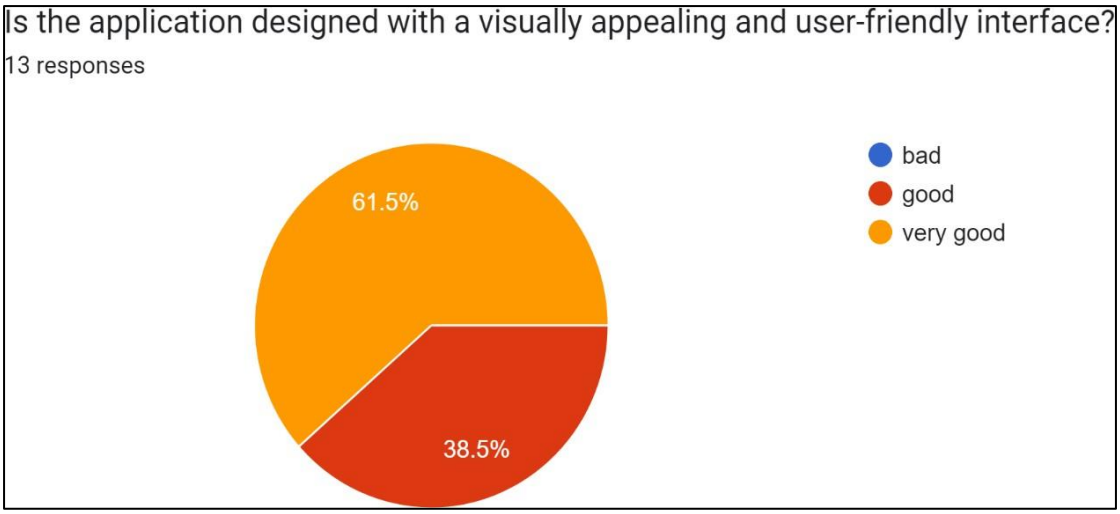

Figure 5-6 Question-1

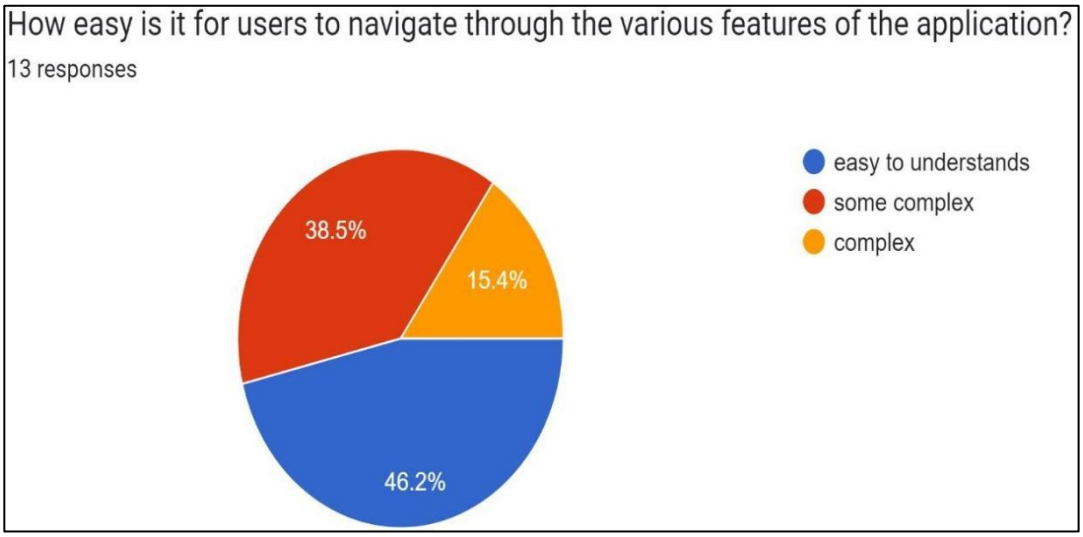

Figure 5-7 Question-2

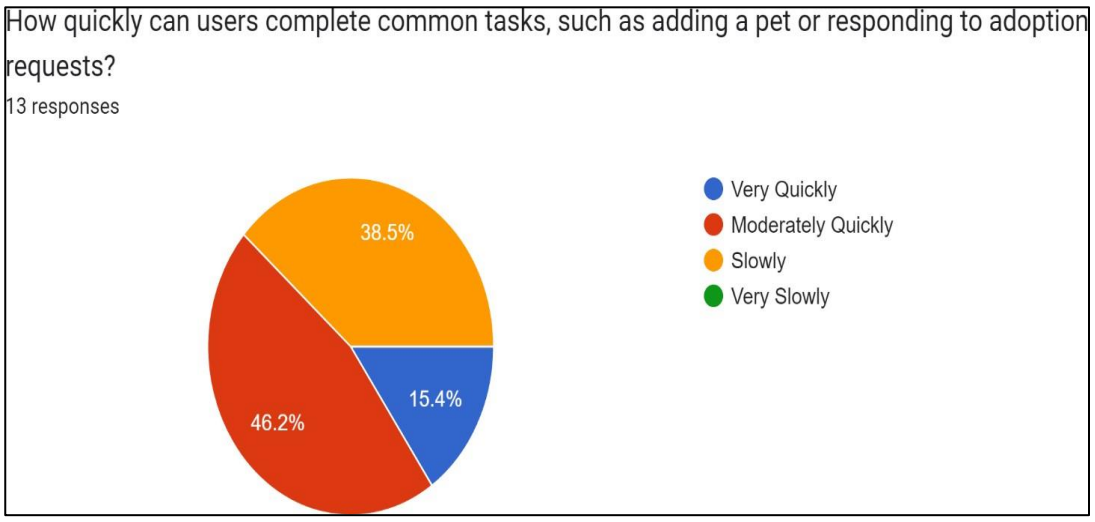

Figure 5-8 Question-3

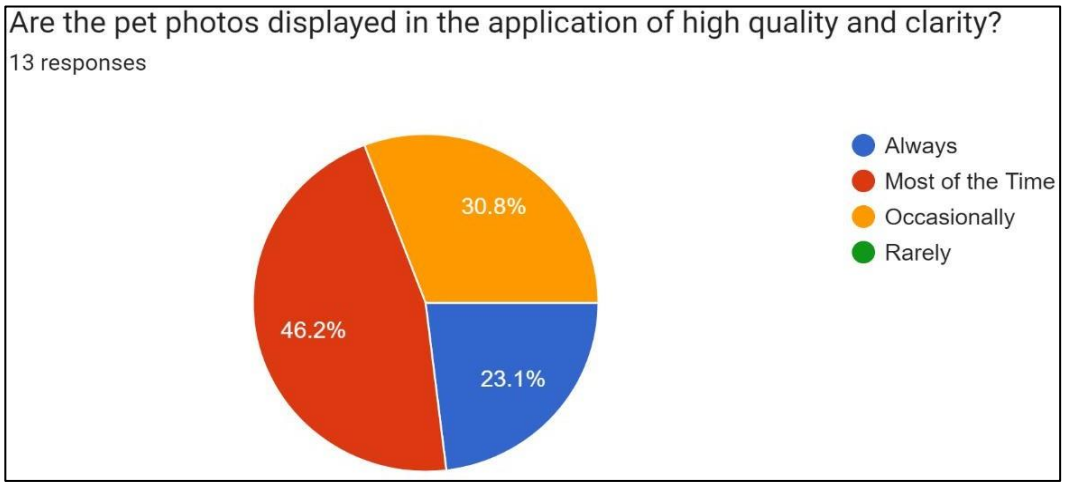

Figure 5-9 Question-4

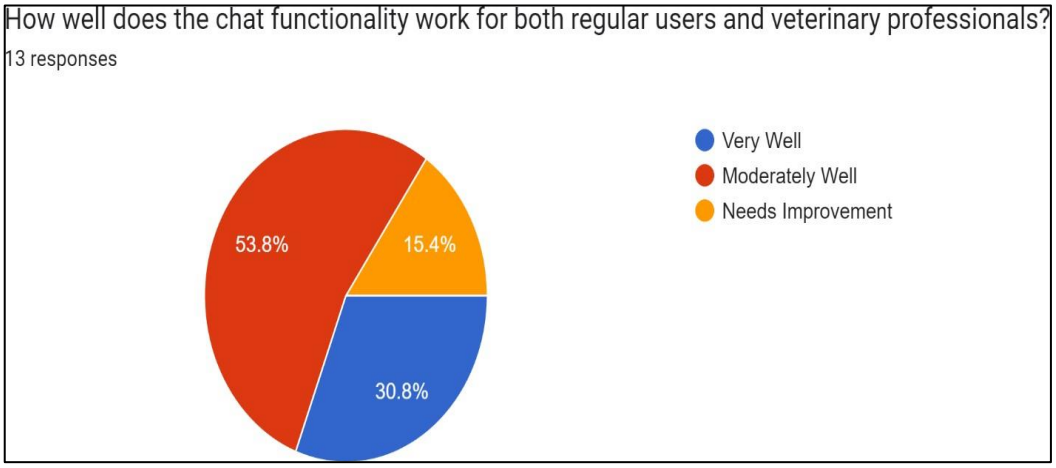

Figure 5-10 Question-5

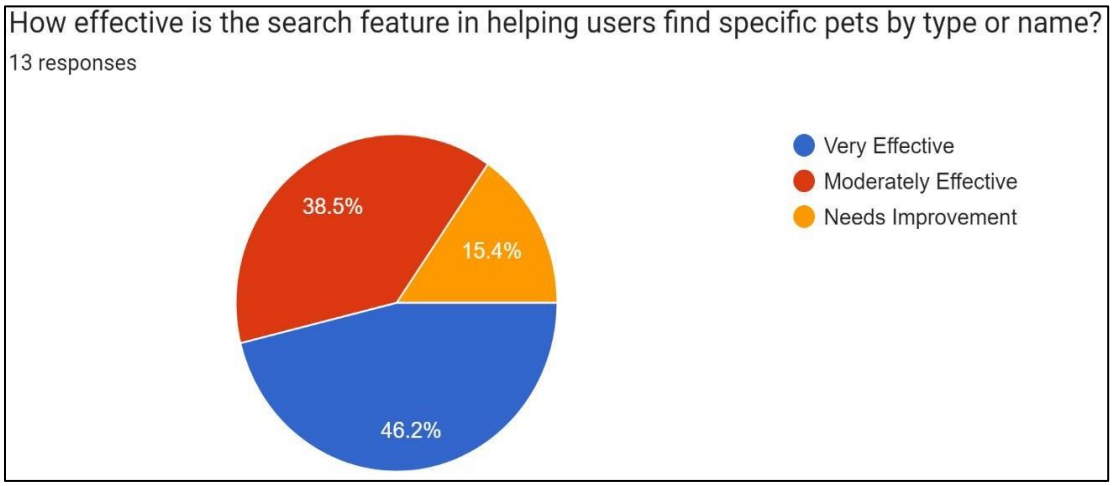

Figure 5-11 Question-6

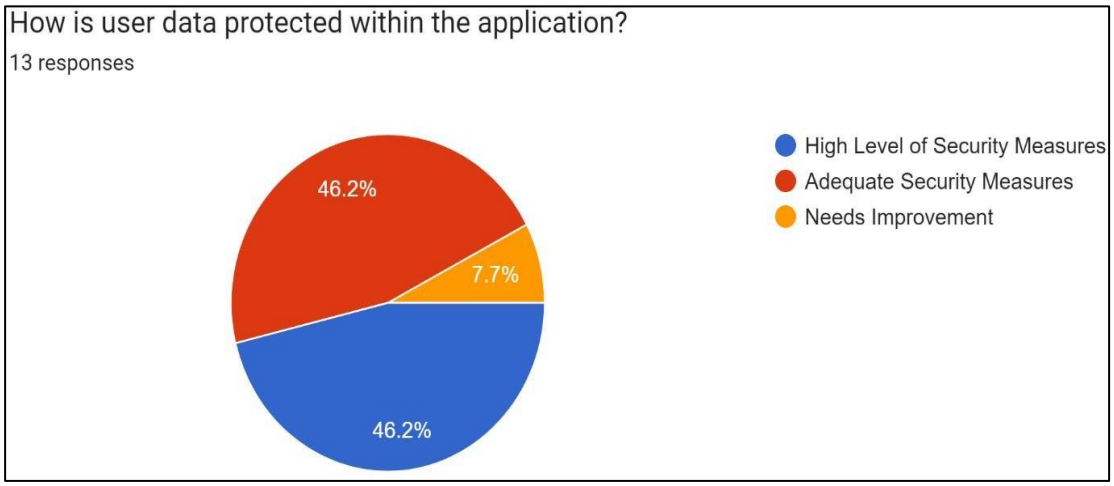

Figure 5-12 Question-7

CHAPTER 6

CONCLUSION

6.1 Introduction

In retrospect, the Animal Adoption Application represented an innovative solution to the pervasive issue of homeless animals, tailored to individuals unable or unwilling to purchase pets. The primary goal of this senior project was to establish a conduit between animals in need and compassionate individuals seeking to offer them loving homes, achieved through the development of a user-friendly platform facilitating the search and request process for animal adoption. The creation of this application aimed to spotlight the predicament of homeless animals, advocating adoption as a responsible and compassionate alternative. Its functionality empowered users to make a meaningful impact by seamlessly searching for and requesting adoptable animals, thereby extending a lifeline to those living on the streets or languishing in shelters. Through this initiative, the intent was to contribute to the well-being of animals and the broader community, addressing pet overpopulation, diminishing shelter populations, and alleviating the suffering of these animals by endorsing adoption over purchase.

Beyond merely connecting potential adopters with animals in need, the Animal Adoption Application underscored the significance of responsible pet ownership. By furnishing pet care resources and information, the application aspired to ensure that adopted animals received the love, care, and attention they rightfully deserved in their new homes.]

6.2 Summary of chapters:

Chapter 1 (Introduction): This chapter introduced the significance of animal adoption applications and delineated the prevailing issues surrounding extant approaches to handling animal adoption. Furthermore, it elucidated the project's objectives, scope, and importance, providing an overarching view of the project and its potential impact.

Chapter 2 (Literature Review): The chapter delved into a comprehensive review of relevant literature, encompassing case studies and comparisons with existing systems.

Chapter 3 (Methodology): This chapter clarified the chosen methodology for the project and provided a rationale for the selection, outlining the approach taken to achieve project goals.

Chapter 4 (Requirement Analysis and Design): This chapter focused on the design aspects of the system, elucidating various UML diagrams to depict the architectural and detailed design components.

Chapter 5 (Conclusion): Served as a reflective conclusion on the project, summarizing achievements, goals met, and offering suggestions for future improvements.

6.3 Achievement of Project Objectives

Requirements were meticulously gathered from stakeholders to pinpoint issues related to homeless animals. Following an analysis and comparison of existing systems, the viability of the proposed solution became evident. An architectural design pattern and detailed design were formulated to provide guidance for the system's development.

6.4 Suggestions for Future Improvement

As the project transitioned into PSM2, focus shifted towards the remaining aspects of detail design, construction design, and the implementation of the proposed solution into a fully functional and practical mobile application. Emphasis was placed on enhancing user experience by designing a flexible UX/UI, ensuring users felt at ease when interacting with the application.

REFERENCES

Apple Inc. (n.d.). *Swift - Apple Developer*. https://developer.apple.com/swift/.

Artemov, M. (2024, January 10). On our journey from UIKit to SwiftUI - SwissBorg Engineering - Medium. *Medium*. https://medium.com/swissborg-engineering/on-our-journey-from-uikit-to-swiftui-c6b49e26f91b.

Author, A. S. (2016, December 28). *Find your next pet with these great pet adoption apps*. AppAdvice. https://appadvice.com/apps/best-pet-adoption-apps#google_vignette.

Bradley, J., & Rajendran, S. (2021). Increasing adoption rates at animal shelters: a two-phase approach to predict length of stay and optimal shelter allocation. *BMC Veterinary Research*, *17*(1). https://doi.org/10.1186/s12917-020-02728-2.

Dart programming language. (n.d.). Dart. https://dart.dev/.

Doland, J. (2024, January 9). *The 8 best pet adoption Websites in 2024*. PetMag. https://petmag.com/best-pet-adoption-websites/.

Kotlin Programming Language. (n.d.). Kotlin. https://kotlinlang.org/.

Pawsofkurdistan | Pawsofkurdistaninternational. (n.d.). Pawsofkurdistan. https://www.pawsofkurdistaninternational.com.

React Native · Learn once, write anywhere. (n.d.). https://reactnative.dev/.

Team, F., & Team, F. (2022, May 29). *11 Best pet adoption apps in USA for Android & iOS | Freeappsforme - Free apps for Android and iOS*. Freeappsforme - Free Apps for Android and iOS | Cool Apps to Download. https://freeappsforme.com/pet-adoption-apps-usa/.

Appendix A Software Requirement Specifications

SRSs, or Software Requirements Specifications, detail software system requirements and specifications. It clarifies what the software should do and behave for stakeholders and the development team. The SRS covers features, functionality, user interfaces, performance expectations, and system constraints, ensuring everyone understands. It guides software development and ensures the final product meets goals and user expectations.

(a) 1.1 Purpose

SRSs define and document software system requirements and specifications. It helps developers and stakeholders agree on the software's goals and features. The SRS prevents misunderstandings and scope creep and improves software development life cycle communication. Its main objective is to guide the development process and ensure the software product meets user expectations.

(b) 1.2 Scope

This project is for the university community. Its goal is to help people stay up-to-date on everything that happens on campus, such as changes to the campus landscape, new club activities, and other things.

(c) 1.3 Definitions, Acronyms and Abbreviation

AA: Animals AdoptionUC: Use Case

SRS: Software Requirement Specification

Definitions of all terms, acronyms and abbreviation used are to be defined here.

(d) 1.4 Overview

The next chapter, "Overall Description," covers productive viewpoints, system features, user characteristics, and constraints. Chapter 3, Specific Requirements, discusses user, hardware, and software interfaces. System features discuss use cases, requirements, and activity diagrams.

2. OVERALL DESCRIPTION

Use case diagram is showing the interaction between the actors which are the users and the system. Use case Diagram specifying the requirements. This Use Case Diagram is for best place for animals it shows the interaction between the Actors and the application.

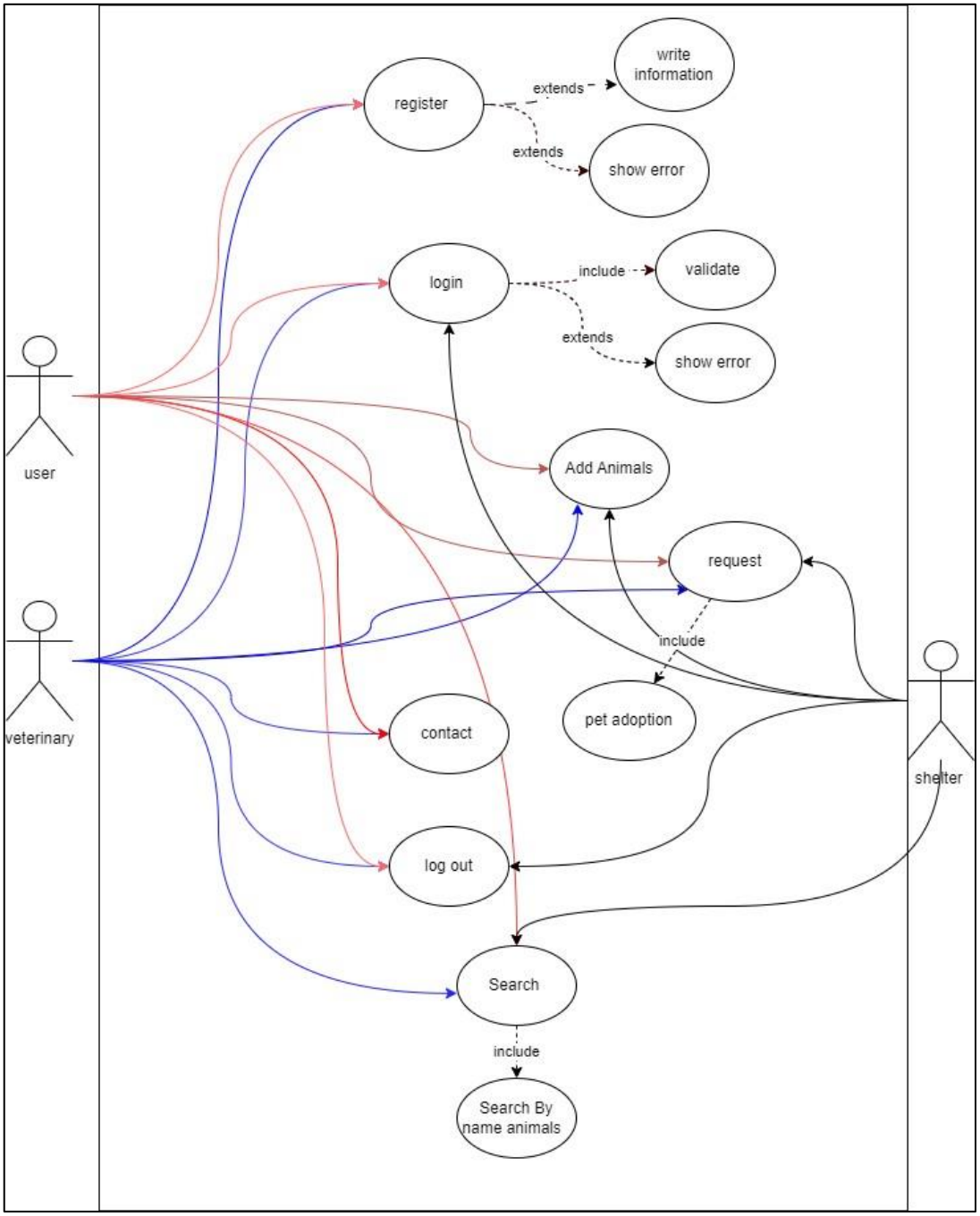

Figure 6-1 use case diagram for animal adoption

(e) 2.1 Product Perspective

As the QAEP is a web-based application, a web browser such as Google Chrome, Safari, etc. is required. The web server is linked to the MySQL databaseserver. The system retrieves the information from the database and displays it on the website.

In this system has 4 users:

1- user

2- Shelter

3- Veterinary

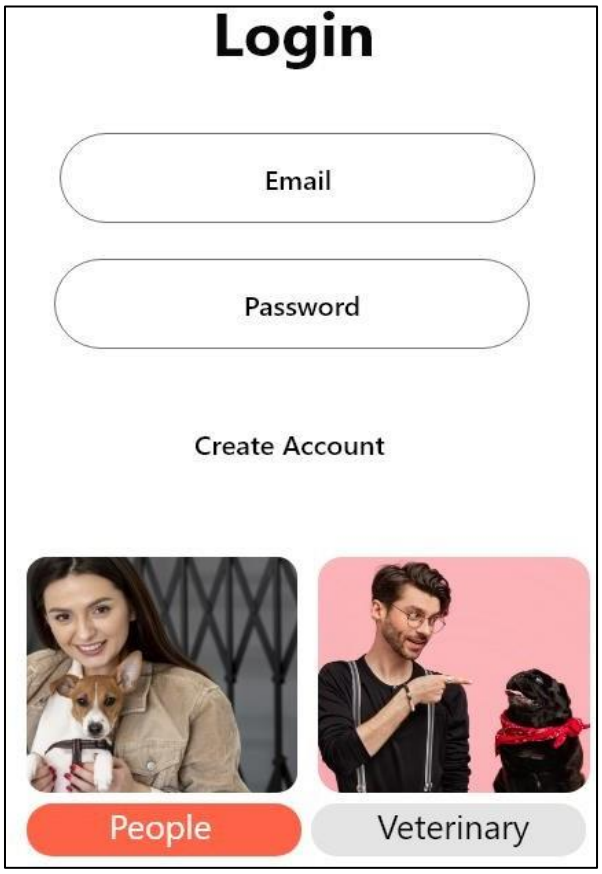

Figure 6-2 Login Page

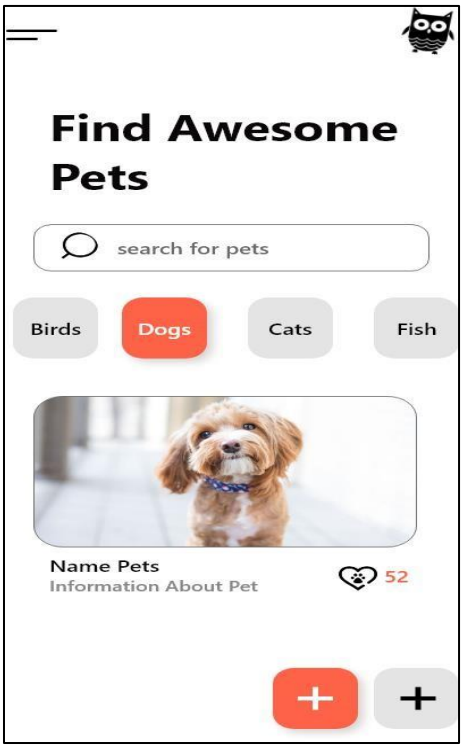

Figure 6-3 Home page

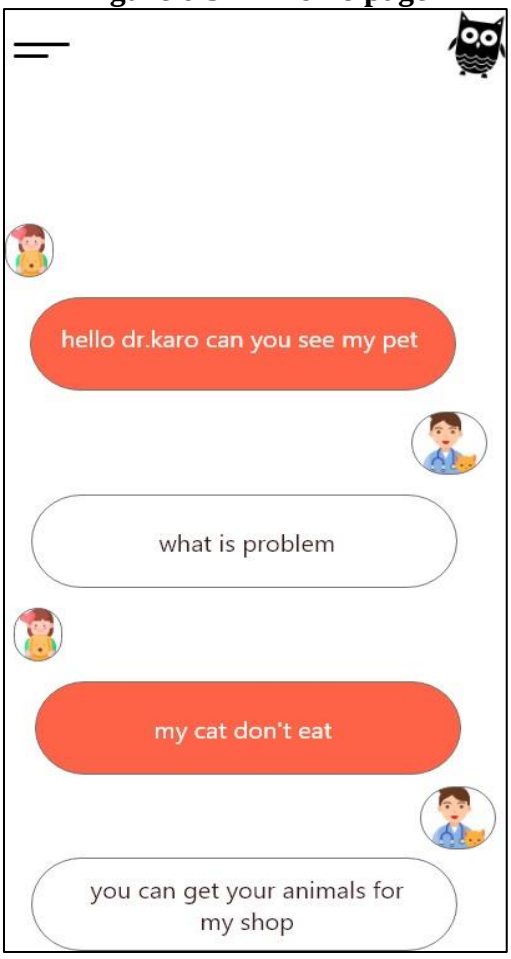

Figure 6-4 Chat page

Figure 6-5 Request pet

2.1.3 Hardware Interfaces

Table 6-1 Hardware Interfaces

Hardware	Minimum Specification
Processor	A13
Random Access Memory	4 GB
Hard Drive	128 GB
Operating System Architecture	64 Bit
Display	4.7″ HD 1080p
Version	IOS 15

2.1.4 Software Interfaces

Table 6-2 Software Interfaces

Hardware	**Specification**
Operating System	Windows 10
Integrated Development Environment	Visual Studio Code, android studio
Database Management System	Xampp
Web Browser	Any
Designing tools	Lucid Charts
High Fidelity Prototype	Adobe XD

2.1.5 Communication Interfaces

All web browsers will work with this system.

2.1. 6 Memory

The primary memories should have at least 2GB of RAM, and the secondary memories should have at least 32GB and, ideally, 64GB.

2.2 Product Functions

The following statement describes the system's intended application and the means by which its components will interact in light of this animal's adoption

2.3 User Characteristics

The website's user interface is too easy to use. All the features that the system offers should be described in an easily navigable interface.

2.4 Constraints

1. Performance: The performance of the website must be fast.
2. Usability The system must have a friendly-user interface so that everyone understands easily.
3. Availability: the system must be available at all times for every user.

2.5 Assumption and Dependencies

Existence of suitable animal shelters or adoption centers: It is assumed that facilities or organizations that provide animal adoption services already exist. These shelters should be equipped with the necessary infrastructure and resources to house and care for animals until they are adopted.

Veterinary care and health evaluations: It is assumed that animals available for adoption have received the necessary veterinary care and health evaluations. This includes vaccinations, spaying or neutering, and any other necessary medical treatments to ensure the animals' health and adoption readiness.

3. SPECIFIC REQUIREMENTS

Figure 3.1: Domain Model of <Animals Adoption >

3.1 External Interface Requirements

3.1.1 User Interfaces

Provide the details for Section 2.1.2.

3.1.2 Hardware Interfaces

Provide the details for Section 2.1.3.

3.1.3 Software Interfaces

Provide the details for Section 2.1.4.

3.1.4 Communication Interfaces

Provide the details for Section 2.1.5.

3.2 System Features

3.2.1 Module <animals adoption >

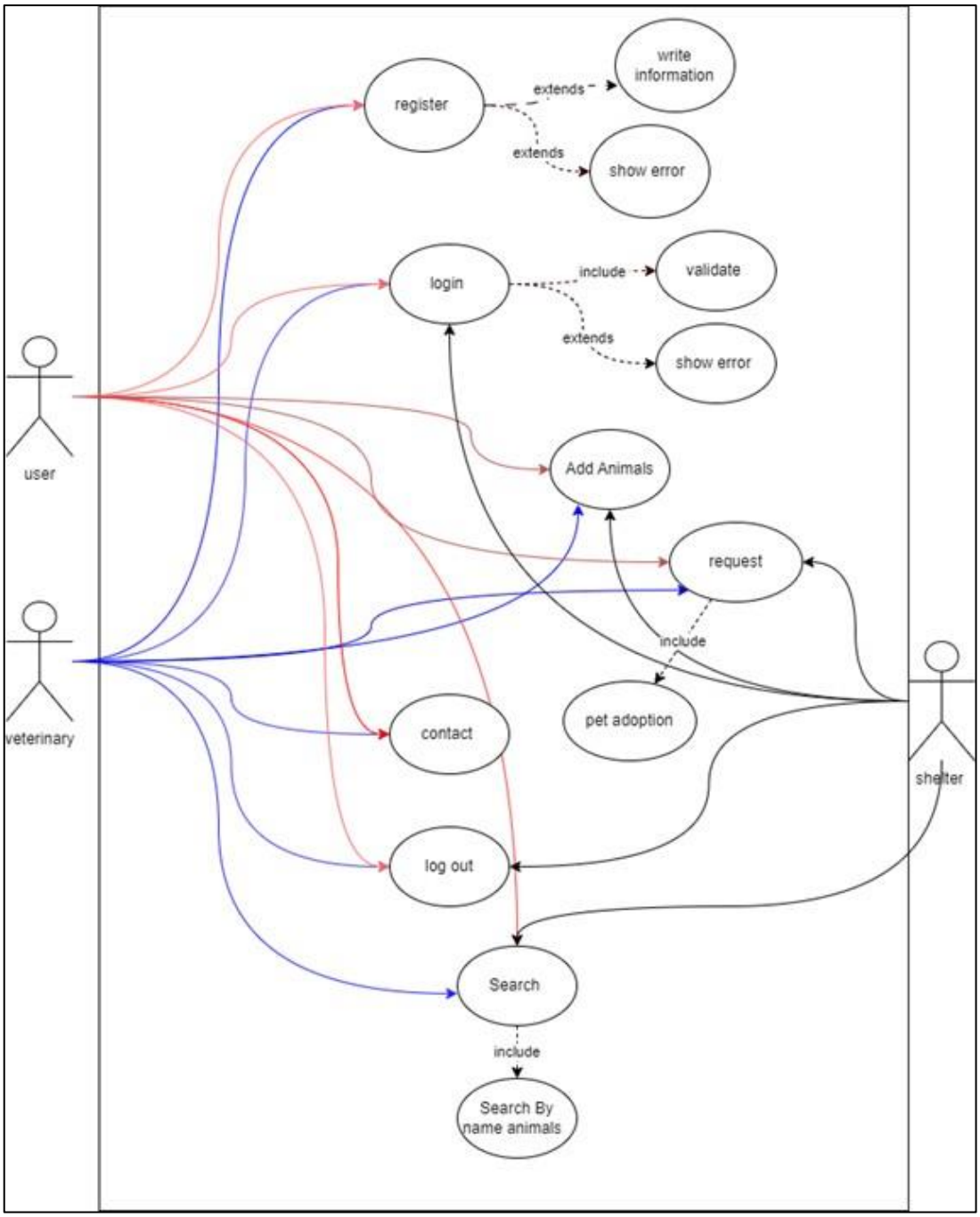

Figure 6-6 Use Case Diagram of Animals Adoption

3.2.1.1 UC001: Use Case < Register >

Table 6-3 Use Case Register

Use Case ID:	UC-01		
Use Case Name:	Register		
Created By:	Karo Mahmood	**Last Updated By:**	Karo Mahmood
Date Created:	14/5/2023	**Last Revision Date:**	15/6/2023
Actors:	User, veterinary		
Description:	The use case talks about how the user register to the application.		
Pre-conditions:	The user must have the access to the internet. The user must access the application.		
Flow of events:	The user opens the application. The user clicks on the Register button. The user will fill up the form and they will send the registration form. Then the click on register button. The system will send the inputs to the data base. If registration is successful, the user will be informed.		

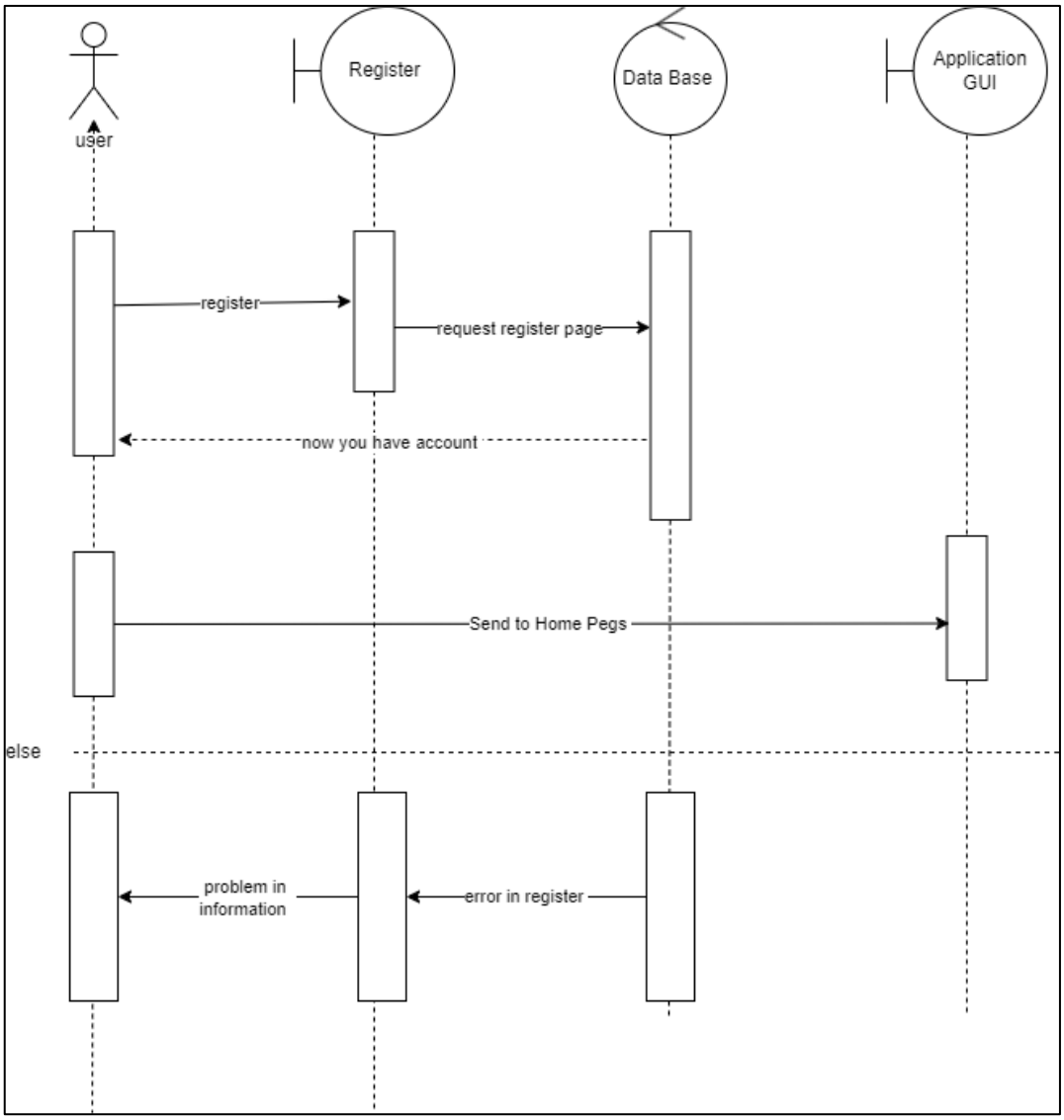

Figure 6-7 System Sequence Diagram of Register

3.2.1.2 UC002: Use Case <login>

Table 6-4 Use Case login

Use Case ID:	UC-02		
Use Case Name:	Login		
Created By:	Karo Mahmood	**Last Updated By:**	Karo Mahmood
Date Created:	14/5/2023	**Last Revision Date:**	15/6/2023
Actors:	User, veterinary, shelter		
Description:	The use case talks about how the user, veterinary, shelter Login to the application.		
Pre-conditions:	1. The user and user must have the access to the internet. The user and user must access the application. The user must get register.		
Flow of events:	The user open the application. The user Enter the Username and password. The system validates the username and the password. (b) The username and password will be verified by the system. The system will show the application home page if the login and password are valid.		

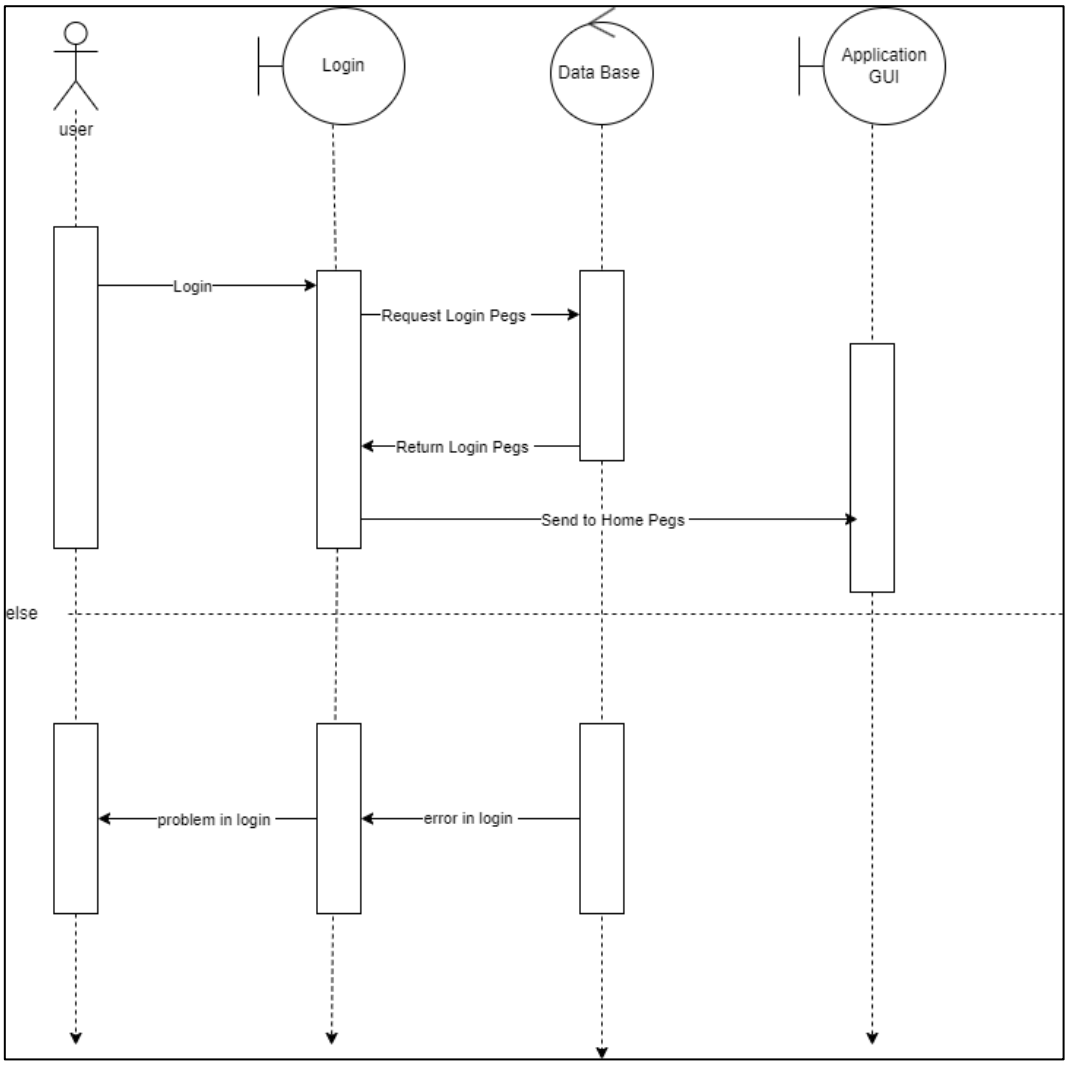

Figure 6-9 _ System sequence diagram of Login

3.2.1.3 UC003: Use Case <search >

Table 6-5 Use Case Search

Use Case ID:	UC-03		
Use Case Name:	Search		
Created By:	Karo Mahmood	**Last Updated By:**	Karo Mahmood
Date Created:	14/5/2023	**Last Revision Date:**	15/6/2023
Actors:	Student		
Description:	The use case talks about how the users Search in the application.		
Pre-conditions:	The user must have the access to the internet. The user must access the application. The User must be logged in to the application.		
Flow of events:	The users must login to the application. The users will write down the name. The system will show the all the information by the name.		

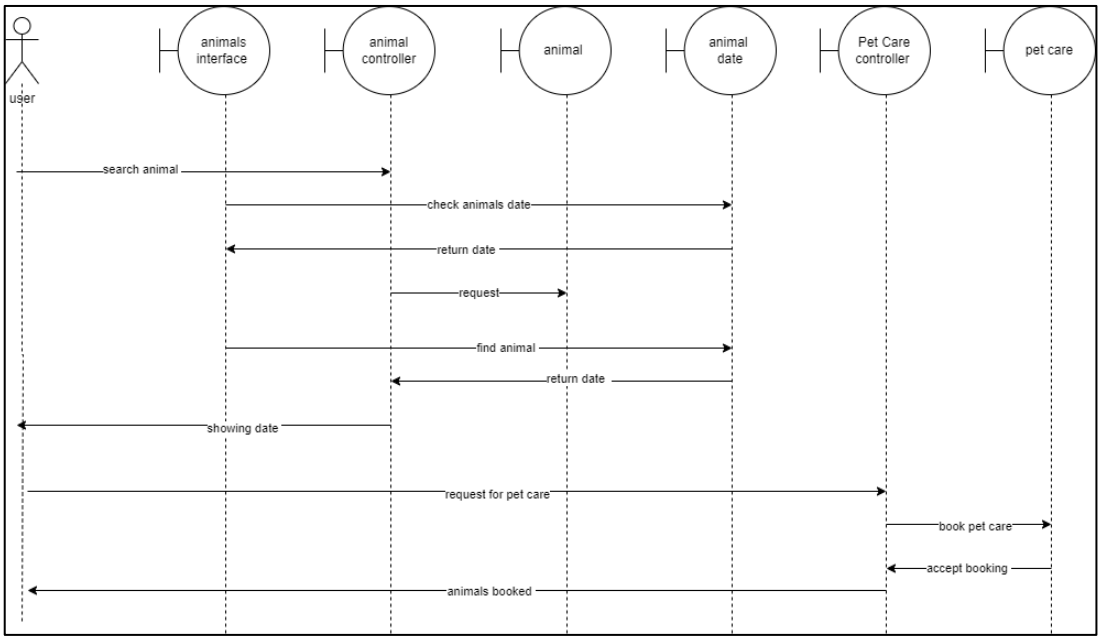

Figure 6-11 System sequences diagram for search

3.2.1.4 UC003: Use Case <Contact>

Table 6-6 Use Case Contact

Use Case ID:	UC-05		
Use Case Name:	Contact us		
Created By:	Karo Mahmood	**Last Updated By:**	Karo Mahmood
Date Created:	14/5/2023	**Last Revision Date:**	15/6/2023
Actors:	User, veterinary		
Description:	The use case talks about how the users can contact the veterinary.		
Pre-conditions:	The user must have the access to the internet. The user must access the application. The User must be logged in to the application. The user clicks on contact page The user enter the send messages.		
Flow of events:	The users must login to the application. The users will click on the contact us page.		

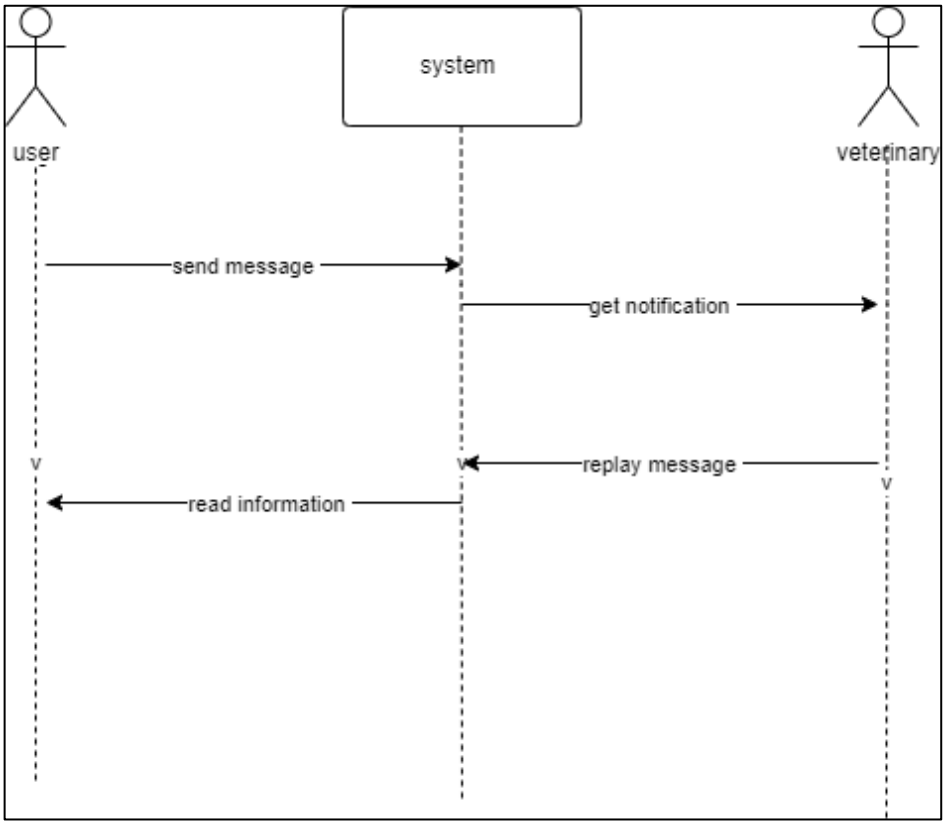

Figure 6-12 Sequence Diagram of Contact

3.2.1.5 UC003: Use Case <logout>

Table 6-7 Use Case logout

Use Case ID:	UC-05		
Use Case Name:	Logout		
Created By:	Karo Mahmood	**Last Updated By:**	Karo Mahmood
Date Created:	14/5/2023	**Last Revision Date:**	15/6/2023
Actors:	User, veterinary		
Description:	The use case talks about how the users can contact the veterinary.		
Pre-conditions:	The user must have the access to the internet. The user must access the application. The User must be logged in to the application. The user clicks on logout(c)		
Flow of events:	The users must login to the application. The users will click on the logout.		

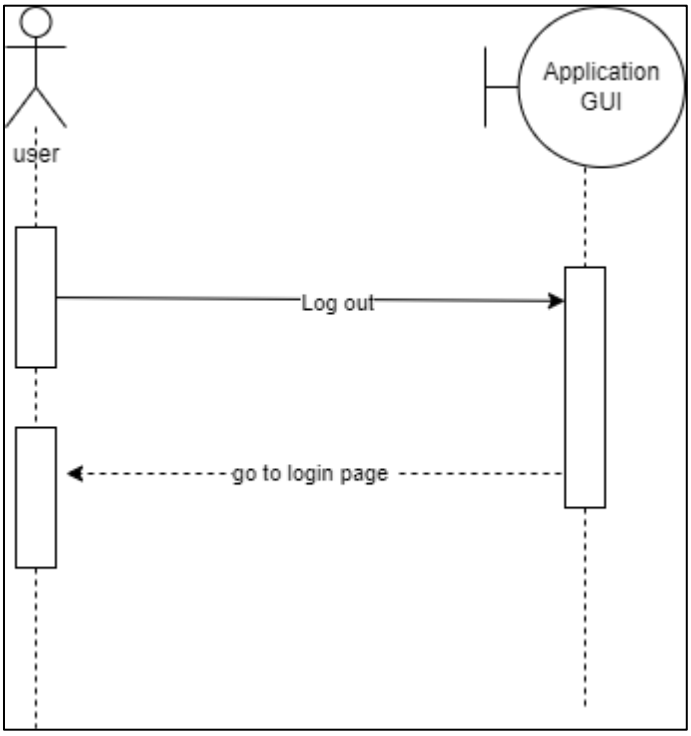

Figure 6-13 System sequence diagram of Log out

3.3 Performance Requirements

Software systems must meet performance requirements. Requirements include response time, throughput, scalability, resource usage, availability, reliability, and security. They ensure the system performs efficiently, reliably, and securely, satisfying users and meeting their needs.

3.4 Design Constraints

To create an application, the system makes use of many pieces of software. A listof the system's
software is provided below:

1. The system has a user-friendly interface because to the integration of flutter,Dart.
2. The data is stored in a firebase database, which was created for the system.

3.5 Software System Attributes

Availability: The system will work without internet interruption.Usability: The system navigates quickly and easily.
MySQL services secure database user data.

Appendix B Software Design Document

4. INTRODUCTION

A Software Design Document, or SDD, is a document that describes the architecture and design of a software system. It provides a detailed description of the software's structure, organization, and implementation. The SDD assists the development team in comprehending the system's design principles, components, and interactions.

(f) 4.1 Purpose

SDDs describe a software system's design and architecture. Developers, stakeholders, and future maintainers can reference it for software structure, components, and interactions. The SDD clarifies system design, implementation, and technical considerations. It helps developers communicate, maintain consistency, and implement the software system according to design. The SDD helps maintain and improve software by revealing its structure and design rationale.

(g) 4.2 Scope

An SDD typically covers architectural overview, detailed component design, data design, interface design, algorithmic design, and other design aspects. It covers software system design and technical details.

(h) 4.3 Definitions, Acronyms and Abbreviation

SDD: Software Design Description

(i) 4.4 Overview

System analysis is used to discuss system architecture. The system's data design, which specifies the data model and entity types. This document includes an ERD and data dictionary. and the system's main interfaces, which improve application visibility.

5. SYSTEM ARCHITECTURAL DESIGN

5.1 Architecture Model

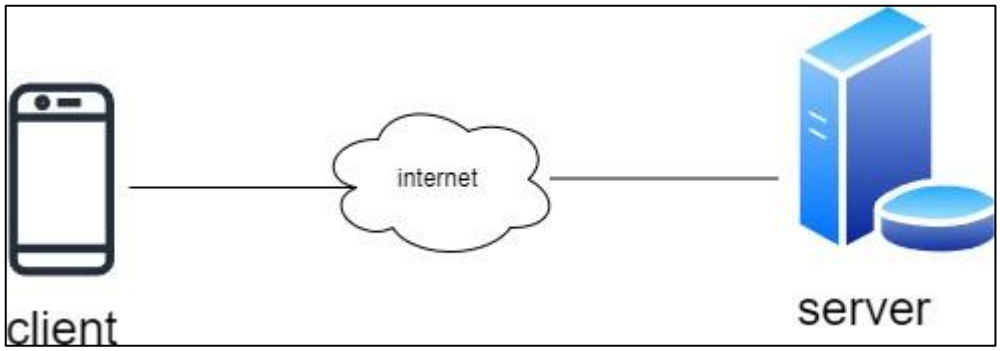

Figure 6-14 Component Model of System Architecture

6. DETAILED DESCRIPTION OF COMPONENTS

6.1 Complete Package Diagram

6.2 Detailed Description

(j)

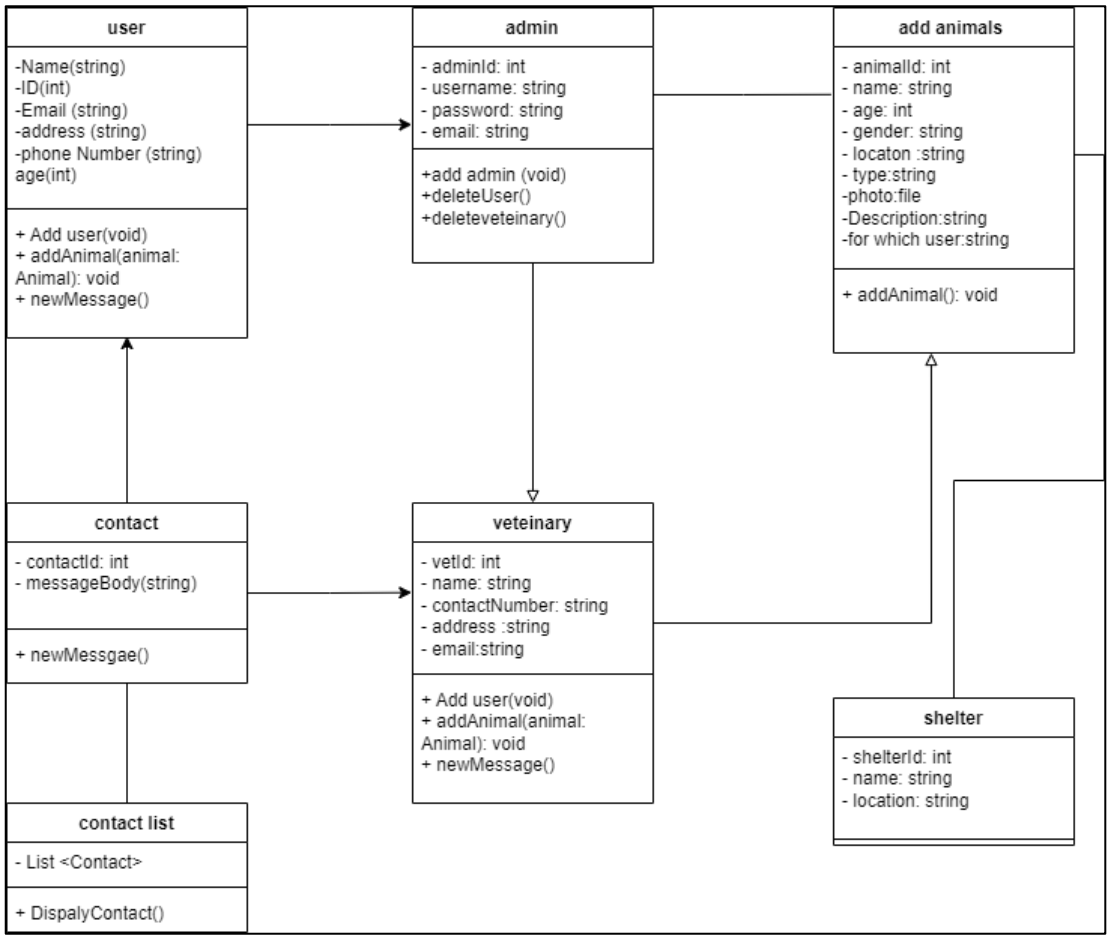

Figure 6-15 Class Diagram

6.2.1.1 Sequence Diagrams

a) SD001: Sequence diagram for Request

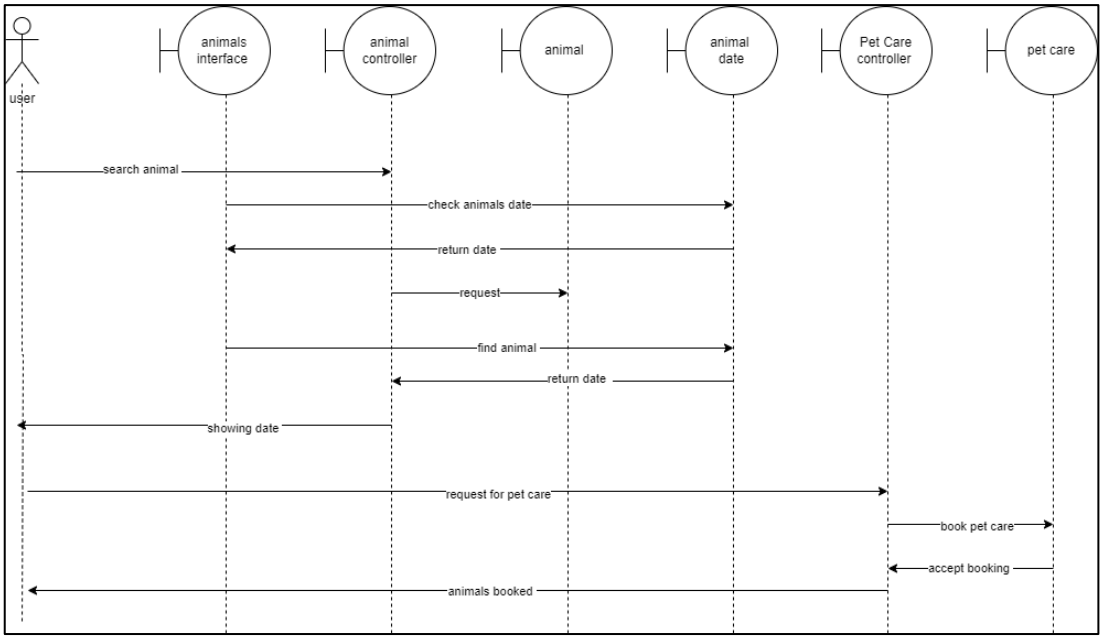

Figure 6-16 Sequence diagram for Request

b) SD002: Sequence diagram for Add Animal

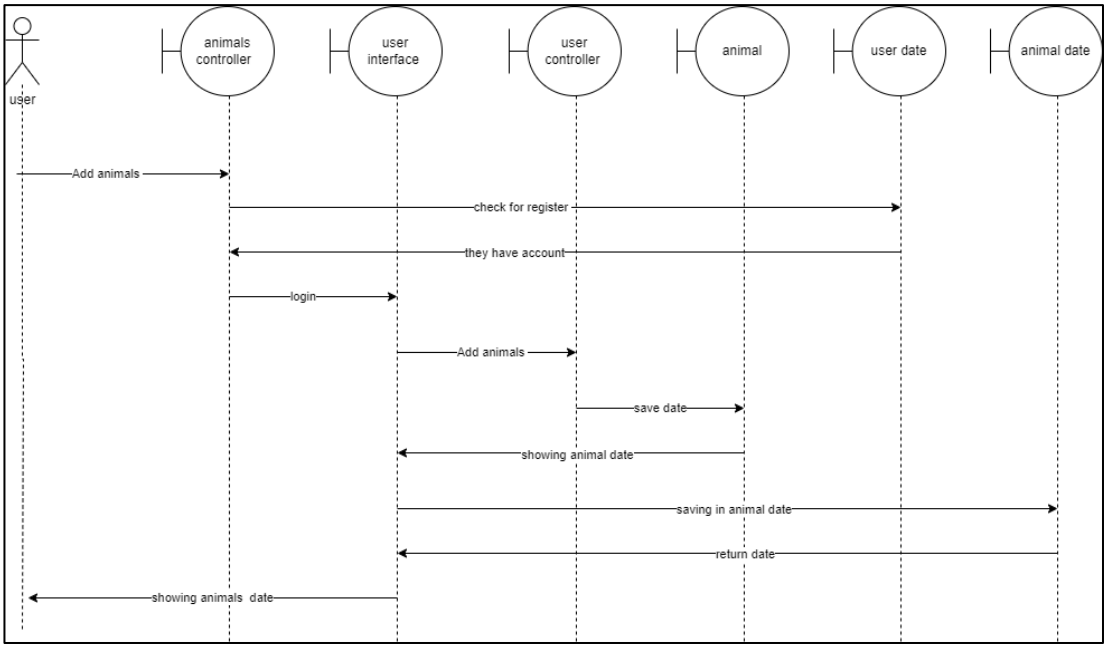

Figure 6-17 Sequence diagram for Add Animal

7. DATA DESIGN

7.1 Data Description

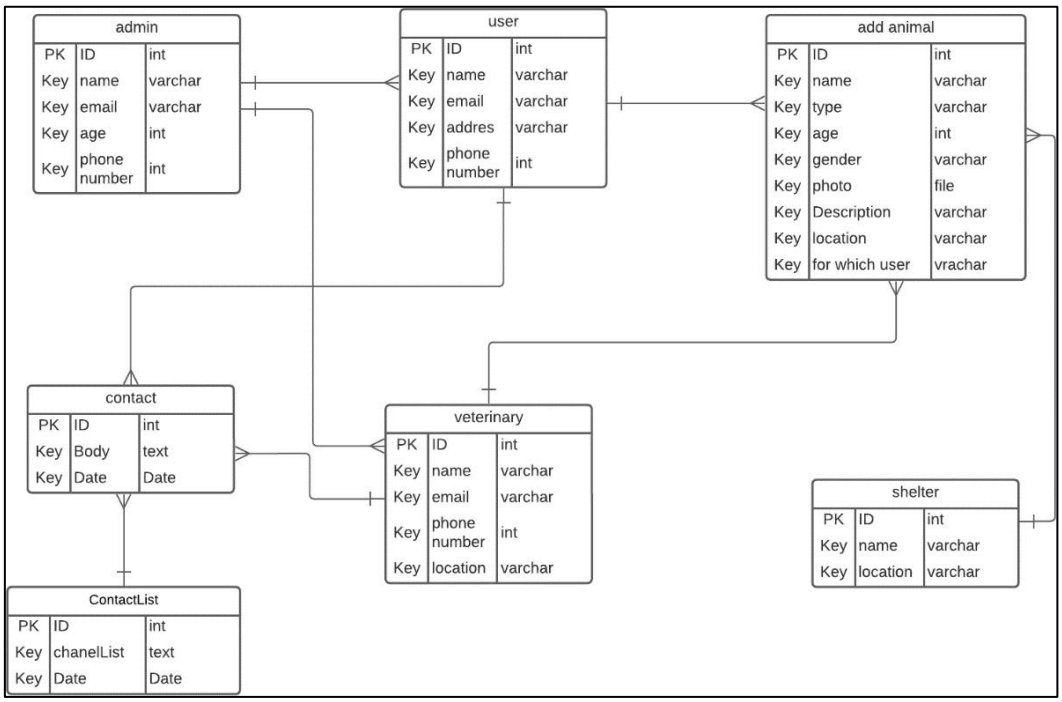

Figure 6-18 Data Dictionary

8. Table 4.1 (user Table)

Table 6-8 user Table

Attribute	Type	PK/FK
ID	Int	PK
Name	Varchar	
Email	Text	
Phone No	Text	
Age	Int	

9. Table 4.2 (veterinary Table)

Table 6-9 veterinary Table

Attribute	Type	PK/FK
ID	Int	PK
Name	Varchar	
Email	Varchar	
Phone No	Text	
Location	Varchar	
Age	Int	

10. Table 4.3 (add animal Table)

Table 6-10 Add Animal Table

Attribute	Type	PK/FK	Length	Null
ID	int	PK	11	
Name	Varchar		250	
Location	Varchar		250	
Phone No	Varchar		250	
Description	Text		350	
Age	int		11	
photo	file			
type	Varchar		250	
Gender	Varchar		250	

11. Table 4.4 (Contact List Table)

Table 6-11 Contact List Table

Attribute	Type	PK/FK	Length	Null
ID	int	PK	11	
Contact List	Text		350	
Date	Date			

12. Table 4.5 (Contact Table)

Table 6-12 Contact Table

Attribute	Type	PK/FK	Length	Null
Contact ID	int	PK	11	
Body	Text		350	
Date	Date			

13. Table 4.6 (Shelter)

Table 6-13 Shelter

Attribute	Type	PK/FK	Length	Null
Contact ID	int	PK	11	
Name	Varchar		250	
Location	Varchar		250	

14. Table 4.7 (Admin)

Table 6-14 Admin

Attribute	Type	PK/FK	Length	Null
ID	Int	PK	11	
Name	Varchar		250	
Email	Text		350	
Phone No	Text		350	
Age	Int		11	

15. USER INTERFACE DESIGN

15.1 Screen Images

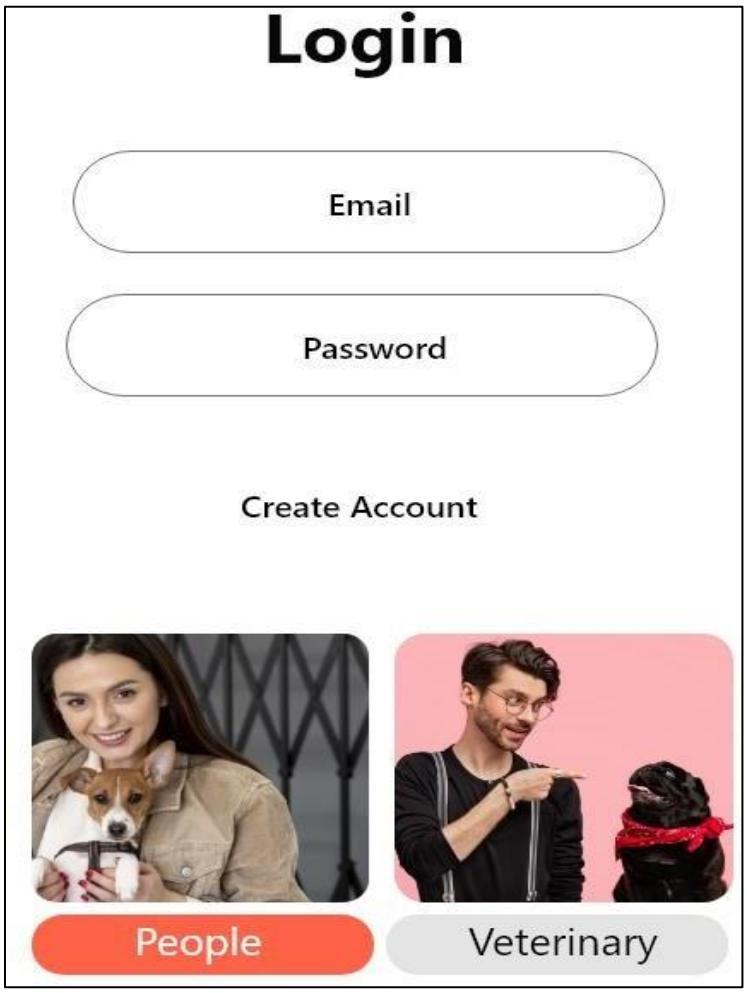

Figure 6-19 Login Page

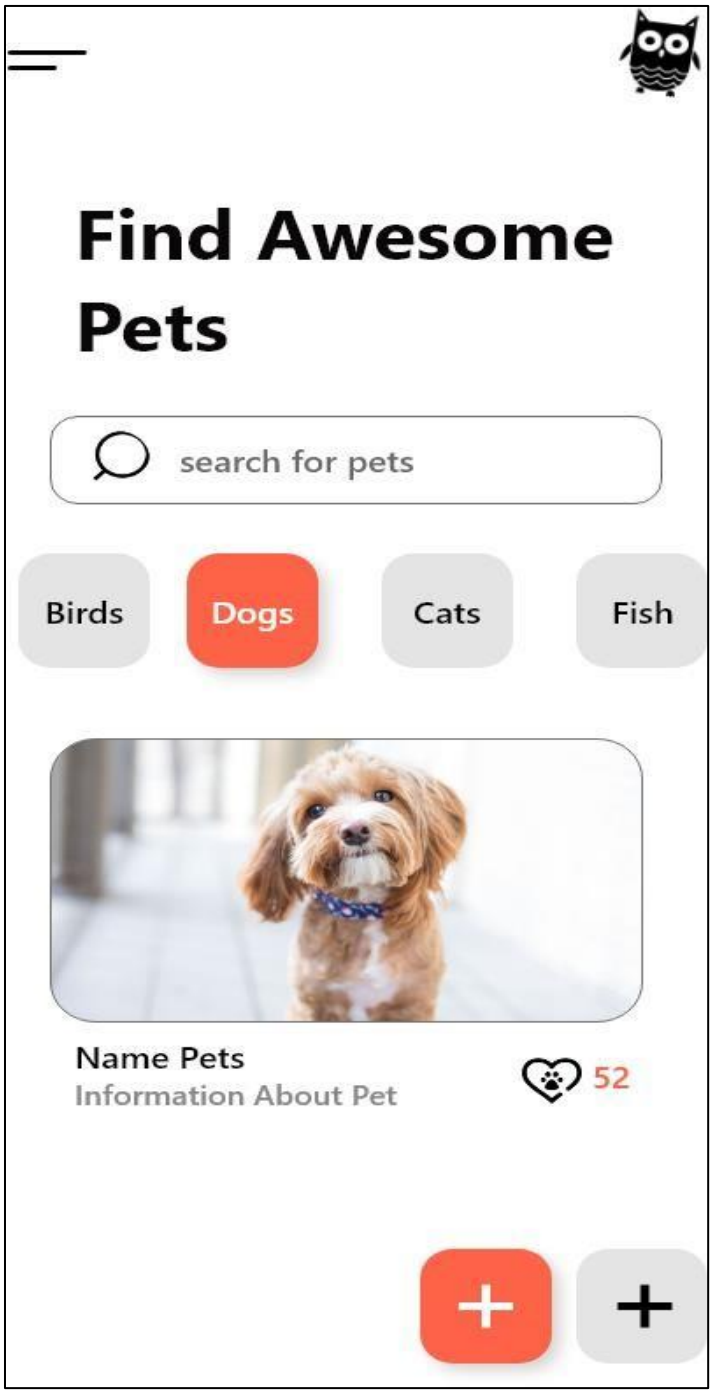

Figure 6-20 Home page

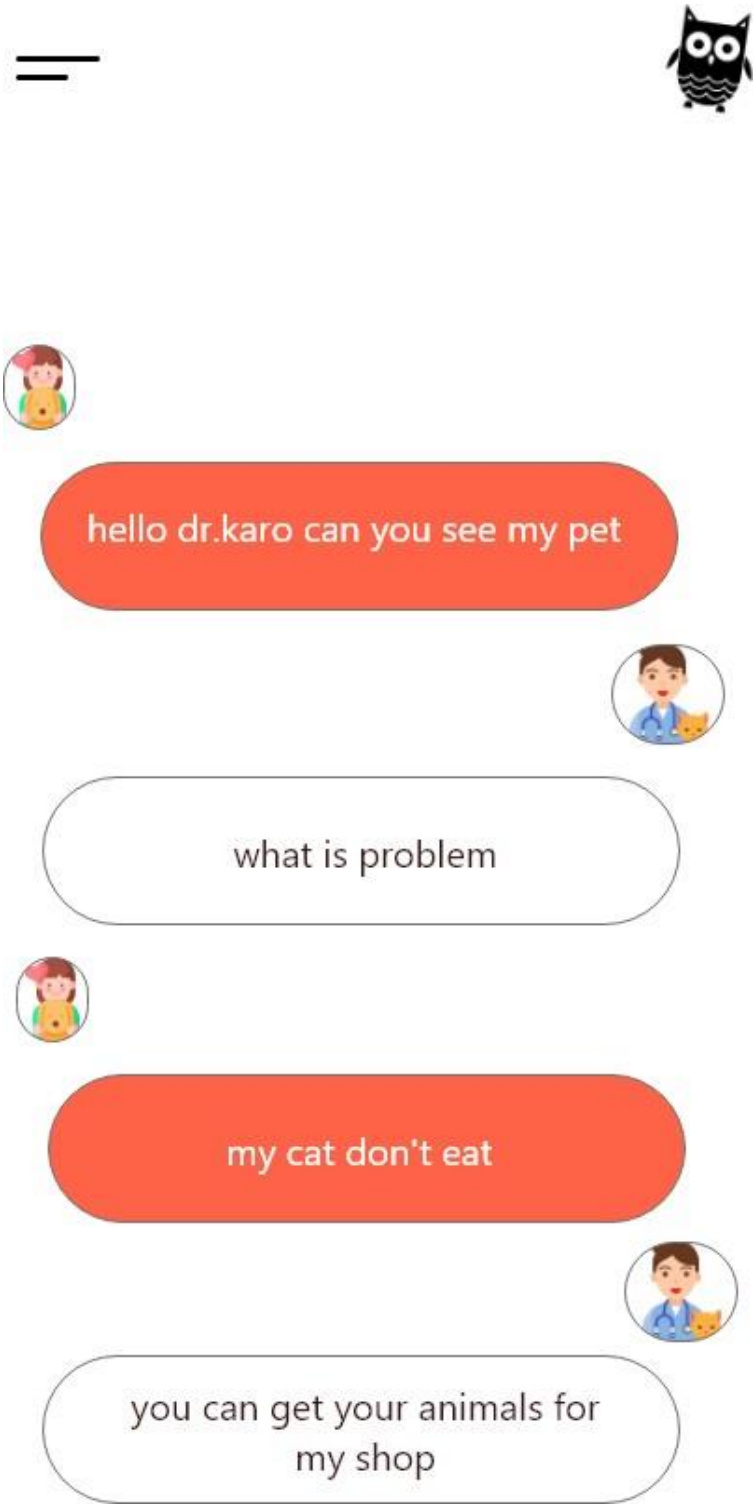

Figure 6-21 Chat Page

Appendix C Software Test Description

16. INTRODUCTION

(k) 17.1 Purpose

System Test Design (STD) is a document that describes the test design and strategies for system-level testing of a software system. An STD is intended to provide a lucid and structured method for testing the functionality, performance, and dependability of software.

(l) 17.2 Scope

The STD scope guides the testing efforts and ensures that the document contains enough information for system-level testing. It clarifies which testing activities and components are covered by the document and manages expectations about detail and coverage.

(m) 17.3 Definitions, Acronyms and Abbreviation

STD: Software Testing Document

(n) 17.4 System Overview

An STD, or System Test Design, provides an overview of the testing methodology and strategies for system-level software testing. It provides a high-level overview of the testing process and serves as a guide for the testing team.

17. TEST CASES, DATA AND EXPECTED RESULTS

17.1 Test TC001 for Module <Name of Module1>: <Name of Use Case (UC001)>

This test contains the following test cases:UC001_01: e.g. Login (username)

Table 6-15 Test case 001

Test Case ID	Input data	Expectedresult	Actual result	Pass / Fail
TC001_01_01	Correct username and password, then click Loginbutton	Successful and redirect to Homepage page (user, admin, veterinary, shelter)	Successful and redirect to Homepage page (user, admin, veterinary, shelter)	Pass
TC001_01_02	Incorrect username or password, then click Login button	Unsuccessful Login and error message display	Unsuccessful Login and error message display	Pass
TC001_01_03	System displays please fill out this field	System displays please fill out this field	System displays please fill out this field	Pass

UC001_02: e.g. Login (password)

Table 6-16 Login (password)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_02_01	5	Password is too short, try again	try again	Fail
TC001_02_02	6	Password OK	login	Pass
TC001_02_03	10	Password OK	login	Pass
TC001_02_04	11	Password is too long, try again	login	Pass
TC001_02_05	ab12!@	Password OK	login	Pass
TC001_02_06	abc123	Password missing symbol	try again	Fail
TC001_02_07	abc!@#	Password missing number	try again	Fail
TC001_02_08	123!@#	Password missing character	try again	Fail
TC001_02_09	abc123	Password missing symbol	try again	Fail
TC001_02_10	abcdef	Password missing numbers and symbol	try again	Fail
TC001_02_11	123456	Password missing characters and symbol	try again	Fail
TC001_02_12	!@#$%^	Password missing numbers and characters	try again	Fail
TC001_02_13	(empty)	Password other than character, number and symbol	try again	Fail

17.2 Test TC002 for Module1: <login>

Table 6-17 Test Case for Login

Use Case Name	Login
Use Case ID	uc01
Description	The use case talks about how the user Login to the application.
Pre-Condition	The user must have an account on the application.
Date	11- 6 - 2023
Tester:	Karo Mahmood

17.3 Test TC003 for Module2: < Search >

Table 6-18 Test Case for Search

Use Case Name	Search
Use Case ID	uc02
Description	The use case talks about how the users Search in the application.
Pre-Condition	(d) The user must have the access to the internet.
Date	11- 5 - 2023
Tester:	Karo Mahmood

QAIWAN INTERNATIONAL UNIVERSITY	Form No.:LIB003 Edition: 01 Effective Date: 01/ 06/ 2024 Page (s): 03
FINAL THESIS SUBMISSION FORM	

Section 1 (To be completed by Student)
(Please tick (√) where applicable)

Library
Qaiwan International University

Submission of Final Copies of Thesis

I Kara Mahmud Ahmad (name) hereby submit:
the final copies of my thesis for the degree of [✓] Bachelor / [ ] Master / [ ] Doctor of Philosophy

The title of the thesis is:

Animal Adoption Mobile Application

Hence, I declare that:

[✓] i) My thesis has been reviewed by my Main Supervisor / HoD / Dean / Faculty

[✓] ii)I had submitted the CD of my softcopy of the thesis (the contents are similar with the hardcopy of the thesis) in a single PDF file.

[✓] iii)I had submitted the Degree Scroll Information Form.

Student Name: Kara Mahmud Ahmad

Department: Software Engineering

Signature: [signature]

Date: 10/7/2024

1

ENDORSEMENT BY MAIN SUPERVISOR

Section 2 (To be completed by Main Supervisor)

Supervisor's Name: Ako Abubakar

Department: S.E

I have examined the thesis of Mr. /Mrs. /Ms.: Kawa Muhammed

I hereby confirm that all corrections and amendments made to the thesis have been rectified by the candidate.

(Main Supervisor's Signature & Stamp)

30/7/2024
(Date)

ENDORSEMENT BY Librarian

Section 3 (To be completed by Librarian)

Librarian's Name: Dlkhwaz Othman

☑ Approved ☐ Not Approved

I hereby confirm that all corrections and amendments made to the thesis have been rectified by the candidate.

(Librarian's Signature & Stamp)

30/7/2024
(Date)

2

STUDENT'S DECLARATION

Section 4 (To be completed by Student)

Student Thesis's Copyright and University's Intellectual Property

"The copyright to a thesis belongs to the student. However, as a condition of being awarded the degree, the student hereby grants to the University, a free, ongoing, non-exclusive right to use the relevant work and/or thesis for the University's teaching, research and promotional purposes as well as free and the non- exclusive right to retain, re-produce, display and distribute a limited number of copies of the thesis, together with the right to require its publication for further research and archival use."

I declare that the contents presented in this thesis are my own which was done at Qaiwan International University unless stated otherwise. The thesis has not been previously submitted for any other degree.

I also declare that my thesis has been reviewed by the Main Supervisor / HoD / Dean / Faculty the comments are as stated in Section 2 (page 2).

Name of Student: Kara Mahmood Ahmad

Signature of Student: .....

IC. No. / Passport No.: B0811364 2

Date: 30/7/2024

3