RESCUE AND ADOPTION HOMELESS ANIMALS

REBAZ AWAT MUSTAFA

QAIWAN INTERNATIONAL UNIERSITY

PSZ 19:16 (Pind. 1/13)

UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : REBAZ AWAT MUSTAFA

Date of Birth : 11/03/2001

Title : RESCUE AND ADOPTION HOMLESS ANIMALS

Academic Session : 2023

I declare that this thesis is classified as:

□	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
□	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
☑	**OPEN ACCESS**	I agree that my thesis to be published as online open access (full text)

1. I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:
2. The thesis is the property of Universiti Teknologi Malaysia
3. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
4. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE OF STUDENT	**SIGNATURE OF SUPERVISOR**
Qu192SCSJ016	**MR. AKO A. JAFFER**
MATRIX NUMBER	**NAME OF SUPERVISOR**
Date: 23 MAY 2024	Date: 23 MAY 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 suffcient 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 A. Jaffer

Date : 23 MAY 2024

MOBILE APPLICATION FOR RESCUE AND ADOPTION HOMLESS ANIMALS

REBAZ AWAT MUSTAFA

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

School of Computing
Faculty of Engineering
Qaiwan International University

May 2024

DECLARATION

I declare that this thesis entitled *“On-Line Recognition of Developing Control Chart Patterns”* 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	:	REBAZ AWAT MUSTAFA
Date	:	17 MAY 2023

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

I am glad to convey my heartfelt gratitude to my supervisor Mr. Ako A. Jafar for his pushing support and assistance on the way to the completion of my project. He supported me with this wonderful opportunity to work on the project. The completion of the project would not have been possible without his help and insights.

ABSTRACT

The primary objective of this study is to investigate and address the issue of increasing homeless animal populations in Kurdistan, while also safeguarding them against various challenges. To accomplish this, a mobile application will be developed, specifically tailored for the region. Although such applications are already popular in Europe, Kurdistan currently lacks a similar solution. The motivation for this project arose from a distressing report issued by an Animal Protection Organization, which revealed the alarming death toll of 400 homeless animals within a single month. To tackle this pressing problem, the proposed application will offer a range of functionalities aimed at addressing the challenges associated with stray animals. Users will be able to utilize the app to notify shelters about the location of stray animals, thereby facilitating their rescue. Additionally, the application will connect interested individuals with the opportunity to adopt these animals, fostering a culture of compassion and responsible pet ownership. Moreover, the app will incorporate various other beneficial features to aid in the overall management of the issue. To ensure the effectiveness and relevance of the solution, extensive requirements gathering will be conducted in collaboration with animal shelters and relevant organizations. The insights and expertise of these stakeholders will serve as the foundation for the application's development, ensuring that it aligns with their expectations and addresses the prevailing problems in the most efficient manner.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

RAHA	-	Rescue and Adoption Homeless Animals
SRS	-	Software requirement Specification

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

In the past, the presence of dogs and cats on the streets of our city was quite rare, and animal shelters were effectively managing the situation by collecting and controlling their population, as well as keeping track of their whereabouts. However, there has been a noticeable increase in the number of stray animals recently, which has become a serious issue for our society. It is now common to see stray animals roaming the streets, causing various problems.

The growing population of dogs has been particularly bothersome for people, especially at night. Incidents of dog attacks, particularly targeting children, have also become more frequent. Unfortunately, these attacks have sometimes resulted in people taking drastic measures, including killing dogs in self-defense. Even cats have been affected, suffering from injuries while scavenging for food.

The rising number of stray animals has led to a general dislike and resentment towards them. Animal shelters are struggling to control the situation and prevent the further increase of these dogs. The overpopulation of stray animals has caused them to spread throughout the city. Additionally, there is currently no available app that allows us to report the location of stray animals and request assistance from shelters to retrieve them.

One crucial solution that needs to be emphasized is the need to reduce the number of homeless animals within the city. This can be achieved by establishing a connection between organizations, shelters, and individuals, encouraging collaboration to tackle this issue. The goal is to work together to decrease the stray animal population, prevent them from contracting diseases, and save them from starvation.

To address these challenges, a mobile application will be developed. The application will serve as a valuable tool to facilitate the swift gathering, feeding, and rescue of stray animals. Users will be able to provide information about the location of a stray animal, which will be transmitted to the nearest shelter capable of handling the situation, regardless of the animal's species. This feature will significantly enhance the effectiveness of rescuing these animals, thus preventing further spread within our cities.

By implementing these proposed measures, we can work towards minimizing the problems caused by stray animals, ensuring their well-being, and creating a harmonious environment for both humans and animals alike.

1.2 Problem Background

We are currently facing numerous challenges related to homeless animals. The existing methods employed by animal shelters and organizations to search for stray animals involve physically combing the streets within cities using vans or relying on phone calls reporting stray animal sightings. Unfortunately, we do not have an application specifically designed to aid in the rescue and control of stray animals.

When a pet goes missing, owners often have no choice but to search for their beloved companion themselves or contact an organization for assistance. This process can be time-consuming and there is no guarantee of success in locating the pet.

Another problem that shelters encounter is the continuous influx of new animals, resulting in a steady increase in their numbers. As shelters cannot accommodate all the animals, it becomes necessary for them to find suitable homes with pet lovers and individuals interested in adopting. However, the lack of awareness about adoption plans and processes at shelters poses a challenge. Many potential pet owners are unaware of whether a shelter has any pets available for adoption.

Moreover, shelters often face shortages of animal food due to the high intake of homeless animals. The majority of people are unfamiliar with the locations of animal shelters or organizations, making it difficult for them to provide support. Currently, shelters primarily rely on social media accounts as their means of communication.

To address these issues, it is imperative to develop a comprehensive mobile application that caters to the specific needs of homeless animals. This application would serve as a centralized platform to facilitate the rescue and control of stray animals. It could include features such as real-time reporting of stray animal sightings, a searchable database of lost and found pets, and information about adoption procedures and available pets at shelters.

Furthermore, the application should provide a channel for people to support shelters by donating funds, supplies, or volunteering their time. By creating an easily accessible and user-friendly application, we can bridge the gap between animal shelters, organizations, pet owners, and individuals who wish to contribute to the welfare of homeless animals.

1.3 Project Aim

the aim of the project is to develop a usable mobile application to rescue, protect, and control stray animals.

1.4 Project Objectives

The objectives of the project are:

(a) To study and investigate existing mobile applications for pet protection.

(b) To design and implement a mobile application for pet protection using Flutter.

(c) To test the developed application based on user acceptance test.

1.5 Project Scope

The main focus of this project is to address the various issues associated with homeless animals, as previously discussed, such as reducing the number of stray animals within cities, safeguarding them against diseases, and providing them with new homes. The proposed solution involves gathering all homeless animals and ensuring they find suitable shelters or homes.

To achieve these goals, a mobile application will be developed, offering several features that significantly improve the search and rescue process compared to traditional methods. By utilizing the app, users will have a more accessible and efficient way of reporting and locating stray animals. To ensure widespread usage, the application will be available on mobile devices.

The chosen methodology for this project is the Iterative Methodology, which will involve dividing the development process into three iterations. This approach allows for continuous feedback and improvement throughout the project's lifecycle. To gather the necessary information and requirements from stakeholders, the project team will collaborate with the PAK Organization (Protecting Animals in Kurdistan Organization). The aim is to incorporate expert insights and align the application with the expectations and needs of the stakeholders.

It is important to note that the application will not feature any advertising or social marketing elements. The primary focus is to provide a practical and efficient tool to address the problems related to homeless animals, without any distractions or commercial motives.

1.6 Project Importance

The proposed solution for addressing the issue of stray animals revolves around protecting and gathering these animals, ultimately providing them with suitable homes. The key feature of the solution is a tracking system that enables individuals to mark the location of stray animals on a map. This allows shelters and organizations to respond more promptly and efficiently to the calls for help.

The primary beneficiaries of this solution are the shelters and animal organizations themselves. The implementation of this system is expected to significantly streamline their work processes, making it easier and faster to address the needs of stray animals. By leveraging technology and the tracking capabilities of the proposed application, these entities can enhance their rescue efforts and provide timely assistance to animals in distress.

However, it is essential to consider potential risks and challenges that may arise during the development and implementation of the solution. One critical aspect is the usability of the system. It is crucial to ensure that the application is user-friendly and intuitive, so that both individuals reporting stray animals and shelter/organization personnel can easily navigate and utilize its features. Additionally, it is essential to gather comprehensive requirements from stakeholders to ensure that the solution meets their specific needs and expectations.

By addressing these potential risks and challenges, the proposed solution can offer an effective tool for protecting and rescuing stray animals, ultimately improving the overall welfare and well-being of these vulnerable creatures.

1.7 Report Organization

Chapter 1: Introduction: This chapter provides an overview of the project, highlighting the identified problems and the need for a solution. It aims to establish an understanding of the project's context and the potential impact of the proposed solutions. The objectives of the project are clearly stated to provide a roadmap for the subsequent chapters.

Chapter 2: Literature Review: In this chapter, a comprehensive review of the existing literature related to the project is presented. It includes case studies and examines similar systems or initiatives implemented in other contexts. The literature review serves as a foundation for understanding the current state of the field and identifying best practices that can be applied to the project.

Chapter 3: Methodology: The methodology chapter outlines the chosen approach for conducting the project. It provides a justification for selecting the specific methodology and explains how it aligns with the project's goals and requirements. The chapter also discusses any adaptations or modifications made to the methodology to suit the unique characteristics and needs of the project.

Chapter 4: Requirement Analysis and Design: This chapter focuses on the analysis and design phase of the project. It details the requirements gathering process, including stakeholder interviews and consultations. The chapter then proceeds to present the design of the system, utilizing various Unified Modeling Language (UML) diagrams to depict the system's structure, behavior, and interactions.

Chapter 5: Implementation and Testing: In this chapter, the design of the system is transformed into actual code. It outlines the implementation process and discusses any challenges or considerations encountered during development. The chapter also covers the testing phase, describing the various test methods employed, such as black-box and white-box testing, to ensure the quality and functionality of the application.

Chapter 6: Conclusion: The conclusion chapter provides a comprehensive summary of the project, highlighting its achievements, goals, and outcomes. It reflects on the overall success of the project, considering whether the objectives were met and the proposed solutions were effective. The chapter also offers suggestions for future improvements or enhancements to further advance the project and its impact.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The introduction chapter highlights the current lack of a dedicated mobile application for the stray animals domain, with animal shelters and organizations primarily relying on social media for their work. It addresses the misconception that social media alone is sufficient for their needs, emphasizing the need for a more efficient and effective solution. The chapter asserts that the limitations of relying solely on social media can be easily demonstrated, and the subsequent sections will present evidence to support this claim.

The main objectives of the project are outlined, which include reducing and protecting stray animals while establishing better control over their population. The proposed solution aims to address these objectives by taking into account the specific situation and user requirements within the stray animal domain.

The proposed application is characterized by its strengths and weaknesses. It is expected to provide various functionalities that will significantly improve the work of shelters in controlling and collecting stray animals. For example, the inclusion of an animal location marker feature will facilitate precise tracking and identification. However, it is important to acknowledge that the application will also have limitations and face challenges that need to be considered during its development and implementation.

The scope of the application is defined, with a focus on Sulaymaniyah City in Iraq. By narrowing the geographical range, the application can better cater to the specific needs and dynamics of that area, providing targeted support for managing stray animals.

By establishing the context and objectives of the project, the introduction chapter sets the stage for the subsequent sections, which will delve deeper into the issues surrounding the use of social media and present the proposed solution in more detail.

2.2 Case Study (If any)

The case study for the project is helps for making identifying essential factors, and gathering requirements of the project easier. The case study that we have is to capture all prospective, and understand more about project, by investigating all process of different animal organizations in Kurdistan

Case Study-1 Kurdistan Organization for Animal Rights Protection (KOARP): The Kurdistan Organization for Animal Rights Protection is an organization that ism volunteer, non-profit for protection animal protection Kurdistan Region. KOARP was founded in 2009 by Dr Suleiman Tamer, a veterinary surgeon that grew up in Kurdistan and. Dr Tamer funds the majority of KOARP's work himself.

In Kurdistan, there are no laws that now protect animals, and there is no infrastructure in place to safeguard, rescue, or treat stray animals. KOARP works with local communities to deliver necessary veterinary treatment and education while advocating for the legal protection of animals. We vigorously oppose the use of poisoned meat and other existing approaches to reducing stray animal populations in favours of more compassionate ones.

Case Study-2 War paws Iraq: War paws aim to alleviate the pain and suffering of animals in hostile environments and areas of civil conflict. they help the forgotten animals of war and they set up a found to help the animals rescue with the cost of running shelter and clink. war paws have partner with animal of Kurdistan set up the first ever animal shelter in Iraq. it has about 80 dogs in residence who need help. this organization ask people to donate 1 euro per, month to help animals.

Case Study-3: Protect Animals in Kurdistan: The organization that targeted to collected data form is PAK organization. This Organization founded in Sulaymaniyah/Iraq. Through their extensive programs, PAK is an organization that recently established, their goal is rescue stray animals from illness, starving, and protect them. they are active on social media; they share their work to warn society that leaving animal without help make their life endanger.

2.2.1 Company Organization Structure

PAK, the organization from which most of the data has been collected, operates as a volunteer-based group dedicated to helping and protecting stray animals. The organization primarily relies on social media as their means of communication and outreach.

The work of the organization is carried out through a two-fold approach. Firstly, they actively utilize social media platforms to raise awareness about stray animals, share information about their activities, and seek support from the public. Through their social media presence, they engage with individuals who are interested in their cause, encouraging them to contribute by means of donations or other forms of assistance.

Secondly, PAK has a team of experts and veterinarians who possess specialized knowledge and experience in dealing with animals. These individuals provide their expertise in handling various situations related to stray animals, such as rescuing injured animals, providing medical care, and addressing other urgent needs.

Despite their efforts, the organization acknowledges certain limitations. Due to the high number of stray animals, it becomes challenging for them to offer assistance to every animal in need. This limitation is often a result of insufficient resources and manpower. While they rely on public support, the scale of the issue can sometimes exceed their capacity to address it comprehensively.

By understanding the working process of PAK, it becomes evident that their operations heavily depend on social media platforms for communication and support. Additionally, the organization leverages the expertise of their team members, including veterinarians, to provide essential care and attention to stray animals. However, the limitations they face due to resource constraints indicate the need for a more comprehensive and efficient solution to tackle the challenges posed by the growing number of stray animals.

2.2.2 Manual Operation

In this project, extensive research has been conducted to investigate the various problems related to stray animals and to understand the underlying reasons contributing to the current situation. The purpose of this research was to identify the goals and prospective needs that should be addressed in order to effectively tackle the issue.

To gather the necessary information and requirements for the project, several organizations, including PAK, have been selected as primary sources. PAK, as a non-supported organization comprised of dedicated volunteers working towards helping and protecting stray animals, provides valuable insights and firsthand knowledge about the challenges faced in this domain. Through consultations and interactions with organizations like PAK, their expertise and experiences have been tapped into to gather essential information regarding the requirements and expectations from the proposed solution.

By collaborating with these organizations, the project ensures that the solution aligns with the real-world needs of the stakeholders involved in stray animal management. The input and perspectives gathered from organizations like PAK help in shaping the project's direction, allowing for a more comprehensive and effective approach to addressing the issues faced by stray animals.

The research conducted and the involvement of organizations like PAK contribute to a well-informed and data-driven project, ensuring that the proposed solution is tailored to the specific needs and challenges identified within the stray animals domain.

2.3 Current System Analysis

The information collected for analysis purposes forms the foundation for understanding the problem of stray animals. This information encompasses a range of data, including statistics, case studies, expert opinions, and stakeholder perspectives. By thoroughly analyzing this information, patterns and trends can be identified, contributing to a deeper understanding of the problem and its underlying causes.

Based on this analysis, several proposed solutions have been identified. These solutions aim to address the specific challenges and requirements of the stray animal domain. They are carefully selected by considering their feasibility, effectiveness, and potential impact on improving the situation.

The process of combining these proposed solutions involves assessing their compatibility and synergies. By considering the unique aspects and considerations of each solution, efforts are made to integrate them into a comprehensive and suitable approach. This approach aims to provide a holistic solution that takes into account the multifaceted nature of the stray animal problem.

By combining the proposed solutions, the aim is to create a cohesive and effective strategy that tackles the root causes of the problem, offers protection to stray animals, and supports the efforts of animal shelters and organizations. The goal is to develop a solution that is practical, sustainable, and capable of making a positive impact in addressing the challenges faced within the domain.

Through the integration of the collected information and the selection of suitable solutions, the project strives to provide a comprehensive and well-rounded approach to effectively address the problem of stray animals.

2.4 Comparison between existing systems

After gathering information, and researching on existing systems, a comparison table below has generated that shows how Competitors worked to make their work easier and solve the problem through software in the best way, path of competitors to solve the problem is different, but about quality all built for same domain. The local organization till current time they only use social media to do their work, so it means all competitors uses social media to solve the problem and make their work easer.

Table 2.1 Comparison between Existing System

Properties	***War paws Iraq***	***KOARP***	**PAK Social Media**
Search engine	Social Media	Social Media	Social Media
Favourite list	No	No	NO
Nearby animal for adoption	No	No	NO
Pre-request for-Adoption	No	NO	NO
Information about animals	YES	Yes	YES
Login	No private page	No private page	No private page
Register	Social Media	Social Media	Social Media
Guest mode	No private page	No private page	No private page

Locate nearby shelters	No	NO	NO
Donation	Phone contact	Phone contact	Phone contact
send stray animal location	Phone contact	Phone contact	Phone contact

2.5 Literature Review of Technology Used

Below table consist of all technologies that will be used for implementing this project:

- Vs Code: Visual Studio Code is a code editor, and support for development operations. This editor is used for designing and coding the project.

- Figma: a software for designing UX/UI for different systems and applications, this software is usable to use because, figma is a tool for teams and individuals to create and share high-quality work, where you can see work of many experts.

- EmailJS: EmailJS is a JavaScript library that used for sending emails through using only clint side technology.

- Flutter Framework: is one of the open-source frameworks supported by Google for building beautiful, compiled, and multi-platform applications by just one codebase. This framework helps to develop the application much easier and faster because flutter provides reactive programming language called Dart that makes development faster.

- Dart Language: Dart is a programming language for developing fast applications for multiplatform, and the goal of this programming language is to make flexible execution runtime platform app for development.

- Notion: is a space or a white board when can write , plan , design and create whatever you imagine, like creating a to do list for a project.

- Firebase: is a backend service (BaaS) app development platform that offers hosted backend services, including hosting for static files, a real-time database, cloud storage, authentication, crash reporting, machine learning, and remote setup. Flutter can be run on Firebase.

2.6 Chapter Summary

This chapter was about the literature review of the project, a brief information about targeted organization, manual operation of working of the domain. this chapter focuses on comparison between current systems that has similarity and constructed for the same domain for e problems regarding homeless animals.

CHAPTER 3

SYSTEM DEVELOPMENT METHODOLOGY

3.1 Introduction

The methodology chapter outlines the chosen approach for the project, which in this case is the Iterative Methodology. This section provides a detailed explanation of why this particular methodology was selected and how it will be applied throughout the project.

Iterative methodology is a process that breaks down the project or task into smaller, manageable iterations. Each iteration focuses on a specific aspect or goal and builds upon the progress made in the previous iteration. This approach allows for continuous improvement, adaptability, and flexibility in the project development process.

The chosen methodology is particularly suitable for software development projects, as it allows for frequent testing, feedback, and adjustments. By delivering a working product or solution at the end of each iteration, it ensures that progress is made and that customer or user feedback can be incorporated into subsequent iterations.

The iterative methodology also promotes early testing and idea validation, which helps to identify and address defects or issues in the project at an early stage. This helps to minimize wasted time and resources by catching and resolving problems sooner rather than later.

By adopting an iterative approach, the project aims to achieve incremental improvements and deliver a solution that is more aligned with the needs and expectations of the stakeholders. The flexibility provided by the iterative methodology allows for adjustments and refinements to be made throughout the project, ensuring that the final outcome is of high quality and meets the desired objectives.

Overall, the methodology chapter provides a clear justification for the selection of the iterative methodology and outlines how it will be applied to guide the project's development process. It highlights the benefits of this approach, such as frequent testing, early feedback, and continuous improvement, which are essential for the successful implementation of the project.

3.2 Methodology Choice and Justification

Reason of selecting this methodology is because, its more flexible, less costly change requirement, and easier to test and debug in small iterations. In this methodology requirement is easy to be defined clearly, and understand it easily. Most of the defects will be detected at early stages, Functions, and prototypes will be developed early in the project life cycle, using this methodology we will spend less time on documenting and more on designing, risks will be identified in the iteration also higher risks will be work on with early priority, finally Operating time reduced.

Choosing an iterative methodology for a project can be a good choice for a number of reasons:

- Flexibility: By breaking a project down into smaller, manageable chunks, the iterative approach allows for adjustments and changes to be made as needed.

- Continuous improvement: With each iteration, feedback and new information can be incorporated to improve the project.

- Early delivery: Iterative methodology allows for a working product or solution to be delivered early in the project, allowing for early feedback and validation.

- Cost-effective: With early delivery and continuous improvement, iterative methodology helps to minimize wasted time and resources by reducing the chances of errors and rework.

- Better risk management: Iterative methodology allows for early detection and correction of problems, reducing the risk of project failure.

3.3 Phases of the Chosen Methodology

Overall iterative methodology consists of 7 phases from requirement to till deployment and maintenance. At each phase the proposed solution improved and work on it before deploying it. And each phase is mentioned below:

1. Requirement gathering & analysis: In this phase, requirements are gathered from clints then the requirement will be checked and analysed whether requirements will fulfil or not. In this phase we will know if the budget based on the requirement will be enough or not, then go next phase.

2. Design: in design phase different type of software will be used for constructing various Diagram like, UML, activity, sequence and more.

3. Implementation: in this phase all requirements, and designs collected to implement it into coding language, in order to transform to computer language.

4. Testing: after coding and implementation phase testing phase will start, when all part of the implementation will be tested through different methods, like white box, and black box.

5. Deployment: after completing final version of the software, it will be deployed to the domain or to be used as the solution for the problem.

6. Review: This phase participates in review of the generated product to show its behaviour for viability of the product deployment.

7. Maintenance: After the software deployed in a functional environment, we face some issues, errors, and also sometimes updates required during the maintenance phase.

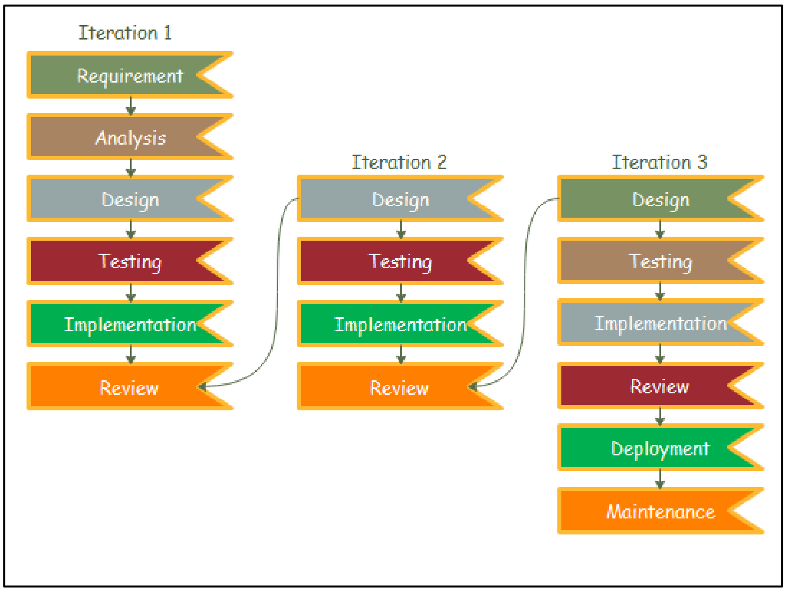


Figure 3.1 Example of Iterative Model

3.4 Technology Used Description

- Vs Code: Visual Studio Code is a code editor, and support for development operations. This editor is used for designing and coding the project.

- Figma: a software for designing UX/UI for different systems and applications, this software is usable to use because, figma is a tool for teams and individuals to create and share high-quality work, where you can see work of many experts.

- EmailJS: EmailJS is a JavaScript library that used for sending emails through using only clint side technology.

- Flutter Framework: is one of the open-source frameworks supported by Google for building beautiful, compiled, and multi-platform applications by just one codebase. This framework helps to develop the application much easier and faster because flutter provides reactive programming language called Dart that makes development faster.

- Dart Language: Dart is a programming language for developing fast applications for multiplatform, and the goal of this programming language is to make flexible execution runtime platform app for development.

- Notion: is a space or a white board when can write , plan , design and create whatever you imagine, like creating a to do list for a project.

- Firebase: is a backend service (BaaS) app development platform that offers hosted backend services, including hosting for static files, a real-time database, cloud storage, authentication, crash reporting, machine learning, and remote setup. Flutter can be run on Firebase.

3.5 System Requirement Analysis

This chapter is more about analysing system design of the project, and all design and analysis are created based on requirements which is essential point and starting point of the project. any information that will be gathered is used to describe result of requirement and analysis like process diagram, sequence diagram, activity diagram, and use case diagram. This chapter also include database design of the system which help us to know how well the structure of the database diagrams designs.

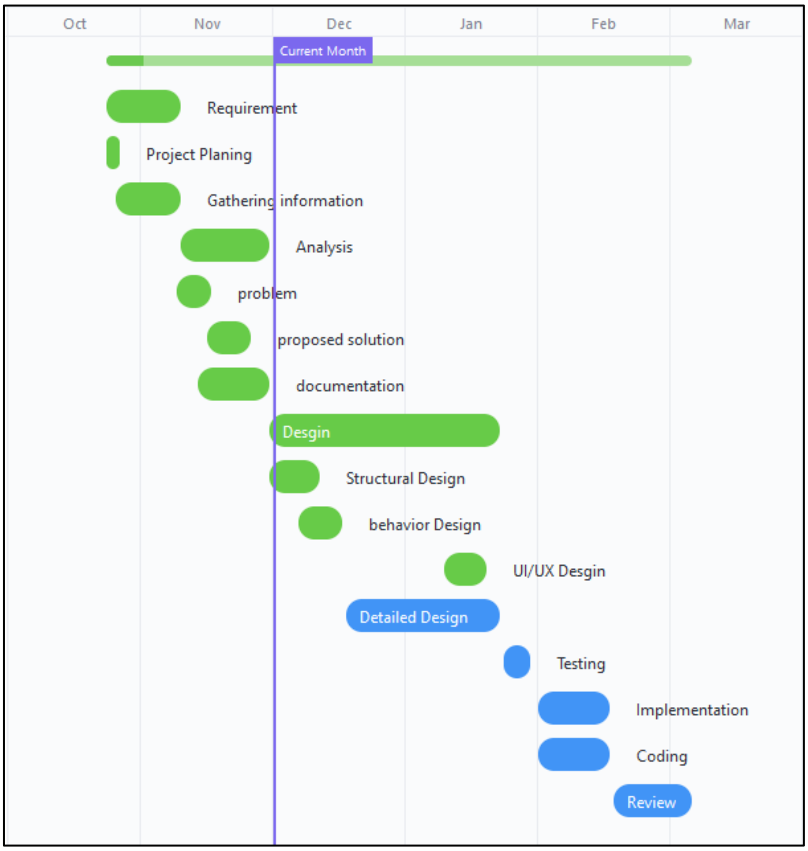


Figure 3.2 Example of Gannt Chart

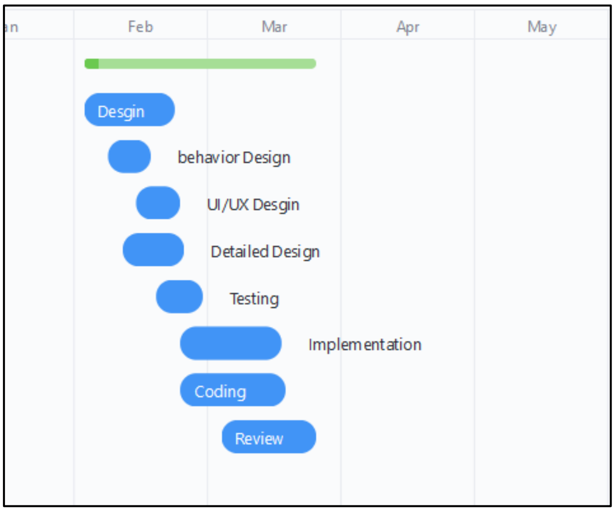


Figure 3.3 Example of Gannt Chart

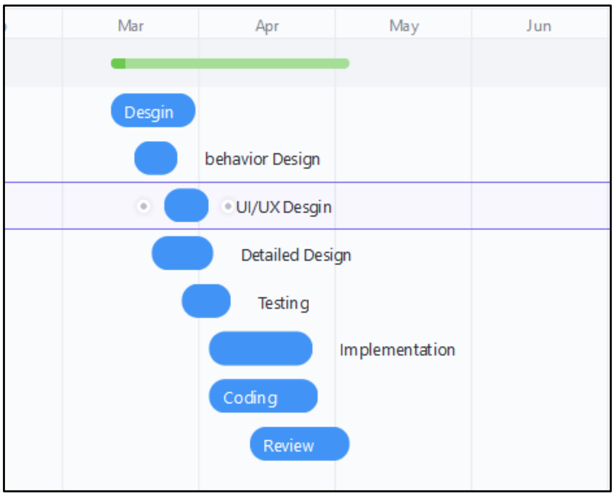


Figure 3.3 Example of Gannt Chart

3.6 Chapter Summary

This chapter was about iterative methodology and the reason of selected methodology. Every phase of the iterative methodology has been clarified and requirement analysis importance, reason, described.

CHAPTER 4

REQUIREMENT ANALYSIS AND DESIGN

4.1 Introduction

Requirement analysis plays a pivotal role in the system development process as it enables us to gain a comprehensive understanding of the problem at hand. Its primary purpose is to prevent errors and ensure the development of accurate software tailored to meet the client's needs. To achieve this, requirements are collected from the client through diligent information gathering. These requirements are then meticulously analyzed to ascertain the exact needs of the client and gain a thorough understanding of their desired outcome. Furthermore, this analysis aids in ensuring that the proposed solution aligns with the identified requirements. As a crucial aspect of this process, all the relevant actors involved are identified, and various software diagrams are employed to depict the different aspects and functionalities of the software in question.

4.2 Requirement Analysis

Once all the essential information has been gathered and various actors and stakeholders of the system have been identified, diverse types of diagrams are created to illustrate the system's structure and behavior. These diagrams are developed based on the available information, and they provide a visual representation of how the system is organized and how it operates. By employing these diagrams, a clear understanding of the system's structure and behavior can be obtained, facilitating effective communication among stakeholders and ensuring a cohesive development process

4.3 Project Design

Project design refers to the meticulous process of strategizing and outlining various components of a project, such as available resources, desired goals, objectives, and budgetary considerations. In the realm of project design, particular emphasis is placed on employing UML (Unified Modeling Language) class diagrams and use case diagrams. These visual tools effectively illustrate the interconnections among classes, showcasing their relationships, as well as presenting crucial attributes and operations associated with each class. By leveraging these diagrams, project designers can facilitate a comprehensive understanding of the project's structure and functionality.

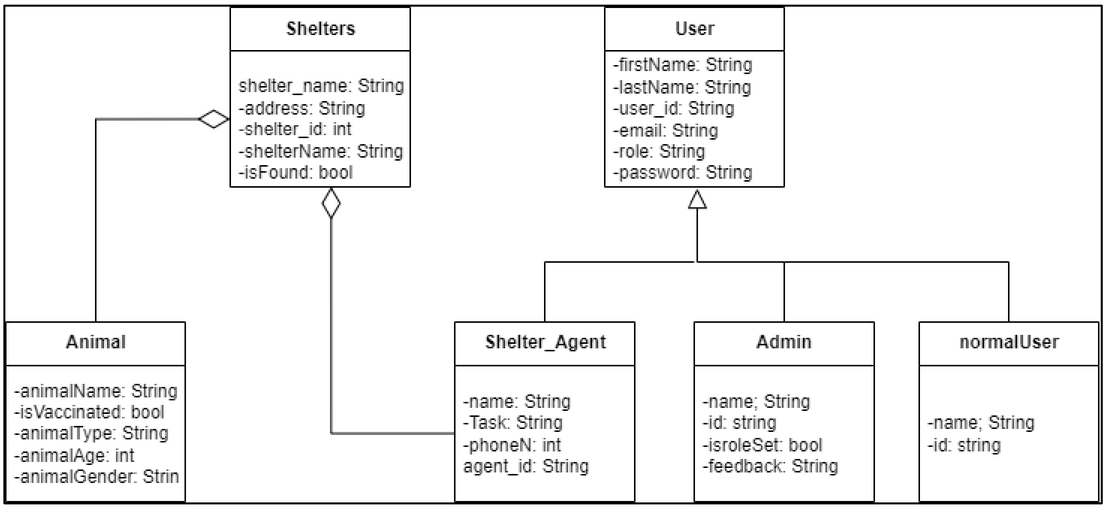


Figure 4.1 Example of Class Diagram

- Login: users access the app through this class by entering his/her username, email, and password.

- Register: to access some functionalities in the software users or guests have to register into the software by filling some information and creating account.

- User: this class contain users' information that registered to the app and store users' action and information.

- Admin: this contains admin information, and responsible for roles, adding organizations, and managing user data.

- Animal: this class include information of animals, and its type.

- Shelter: this class have lots of properties beside it stores all different type information about shelters, and animals, like list of vaccinated animals, list of shelter agents, having enough place for stray animals or not, or giving access to their employee account role of shelter agent.

- Shelter agent: consist of agent information and his/her tasks.

Relation between classes

- Register and User: a composition relation between them because of user class new users is register to the system.

- Login and User: login class cannot be existed without User class, because login inputs Is compared to users account information in User class, because variables connected to database and can recognize if the user already registered to the system or not.

- Admin and Login: association relationship admin access the app through login class.

- Admin and shelter: relation is aggregation admin register shelters, and give them permissions.

- Shelter and Animal class: the relation between them is aggregation, because shelters need animals to be called animal shelter, but Animals can exist without animal shelters and Shelter class stores all information about animals.

- Shelter and shelter Agent class: the relation is inheritance shelter Admin can provide tasks to the agents, and give their employees registration to the app as agent role.

4.4 Database Design

Database design is the process of creating a structured plan for how a database will be organized and used. This involves identifying the data that needs to be stored in the database, determining the relationships between different data elements, and deciding on the appropriate data types and structures to use.

4.5 Interface Design

Interface design, also referred to as user interface (UI) design, encompasses the process of crafting the visual components of a digital product, be it a website or a mobile application, with the aim of facilitating seamless and intuitive user interactions. It entails the creation of an optimal layout, a well-defined visual hierarchy, and an overall aesthetic appeal, in addition to designing the interactions and behaviors of the various elements within the user interface, and Sample UI:

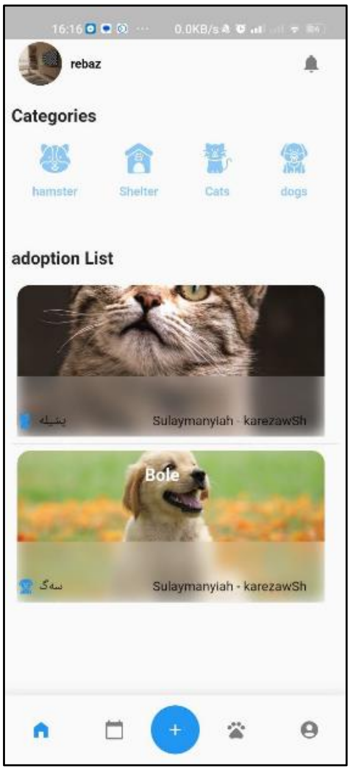
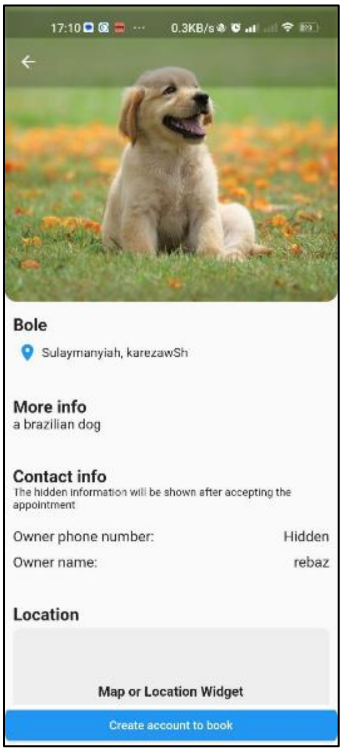


Figure 4.2 Example of Home UI



Bole

Sulaymanyiah, karezawSh

More info

a brazilian dog

Contact info

The hidden information will be shown after accepting the appointment

Owner phone number: Hidden

Owner name: rebaz

Location

Figure 4.3 Example animal description

Use Case Diagram: A graphical representation known as a use case diagram is used to determine a system's needs and how they relate to the users of the system. A use case diagram's key advantage is that it aids in clearly explaining a system's operation to various stakeholders, including consumers, project managers, and developers, and below figure is use case diagram of the system.

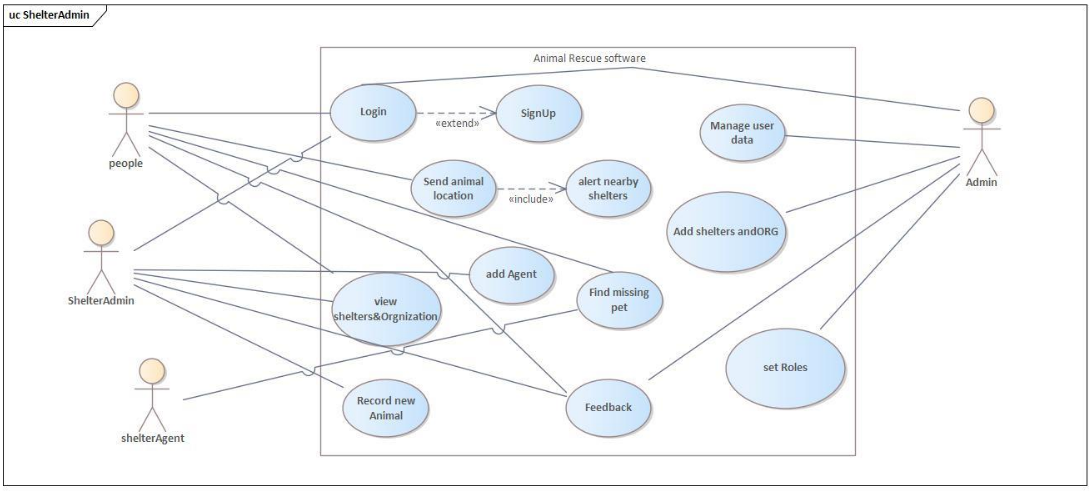


Figure 4.4 Example of Use Case diagram

Table 4.1 Actors and their Roles

No	Actor	Role
1	people	Normal users can register, mark animal location, and more
2	shelterAdmin	Can add new animal, receive user alerts for missing pets, and assign task, add map, and add shelter agents.
3	admin	Can edit user information, add shelters, view feedback.

Sequence diagram: a particular kind of graphic representation, sequence diagram is used to show the relationships and timing of interactions between elements or components in a system. Below sequence diagram shows how the system works, and what are processes. this diagram is responsible for adding animal that Animal Organization or animal Shelters want to share for adoption by people that like animals.

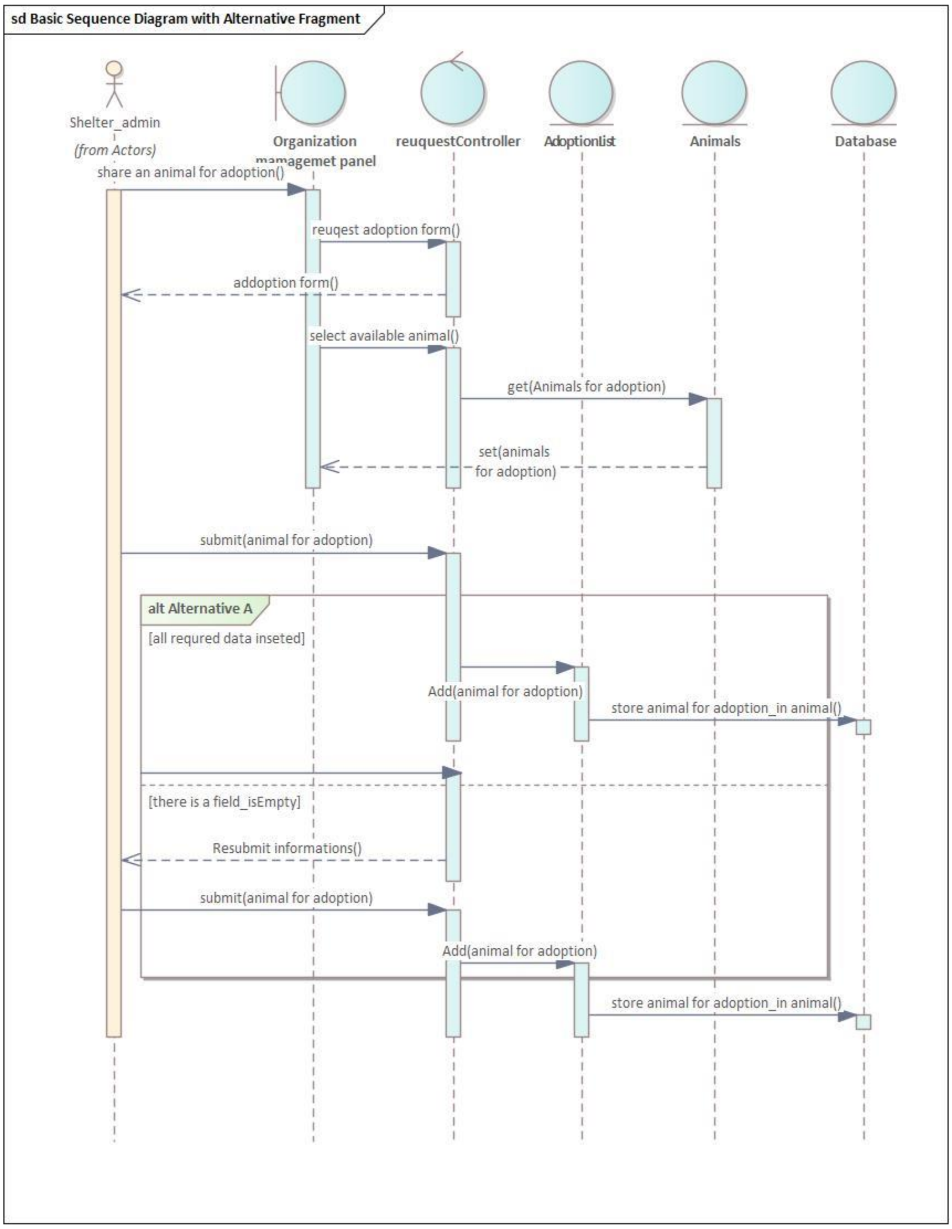

sd Basic Sequence Diagram with Alternative Fragment
Shelter_admin
(from Actors)
Organization mamagemet panel
reuquestController
Adoptionlist
Animals
Database
share an animal for adoption()
reuqest adoption form()
addoption form()
select available animal()
get(Animals for adoption)
set(animals for adoption)
submit(animal for adoption)
alt Alternative A
[all requred data inseted]
Add(animal for adoption)
store animal for adoption_in animal()
[there is a field_isEmpty]
Resubmit informations()
submit(animal for adoption)
Add(animal for adoption)
store animal for adoption_in animal()

Figure 4.5 Example of add animal for adoption

Sequence Diagram marking stray animals: This is the main process of the system, through this operation users can mark location of a stray animal on map and send it to organizations.

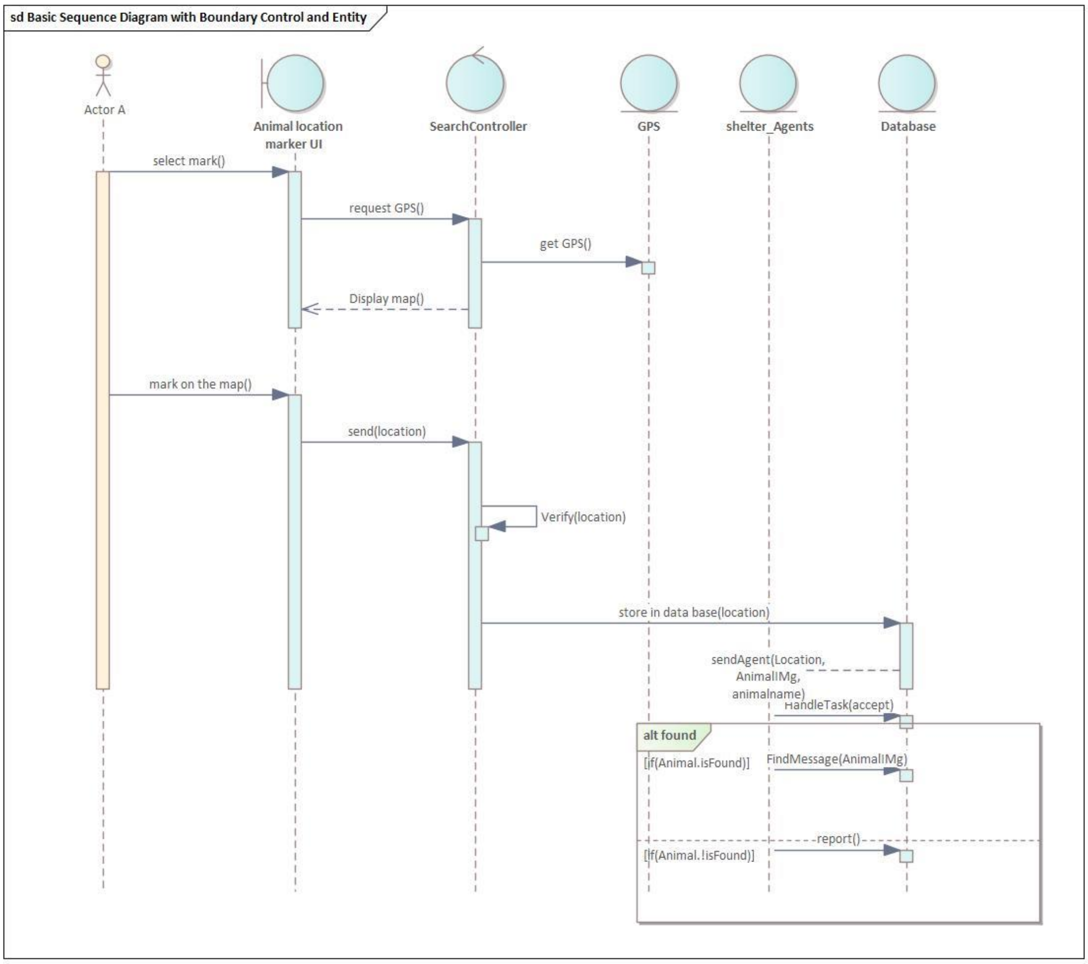


Figure 4.6 Example of Marking stray animal

Activity Diagram: A graphical depiction used in of software engineering is called an activity diagram. It is used to show how actions or activities move through a system. Below are examples of Activity diagrams of the proposed solution.

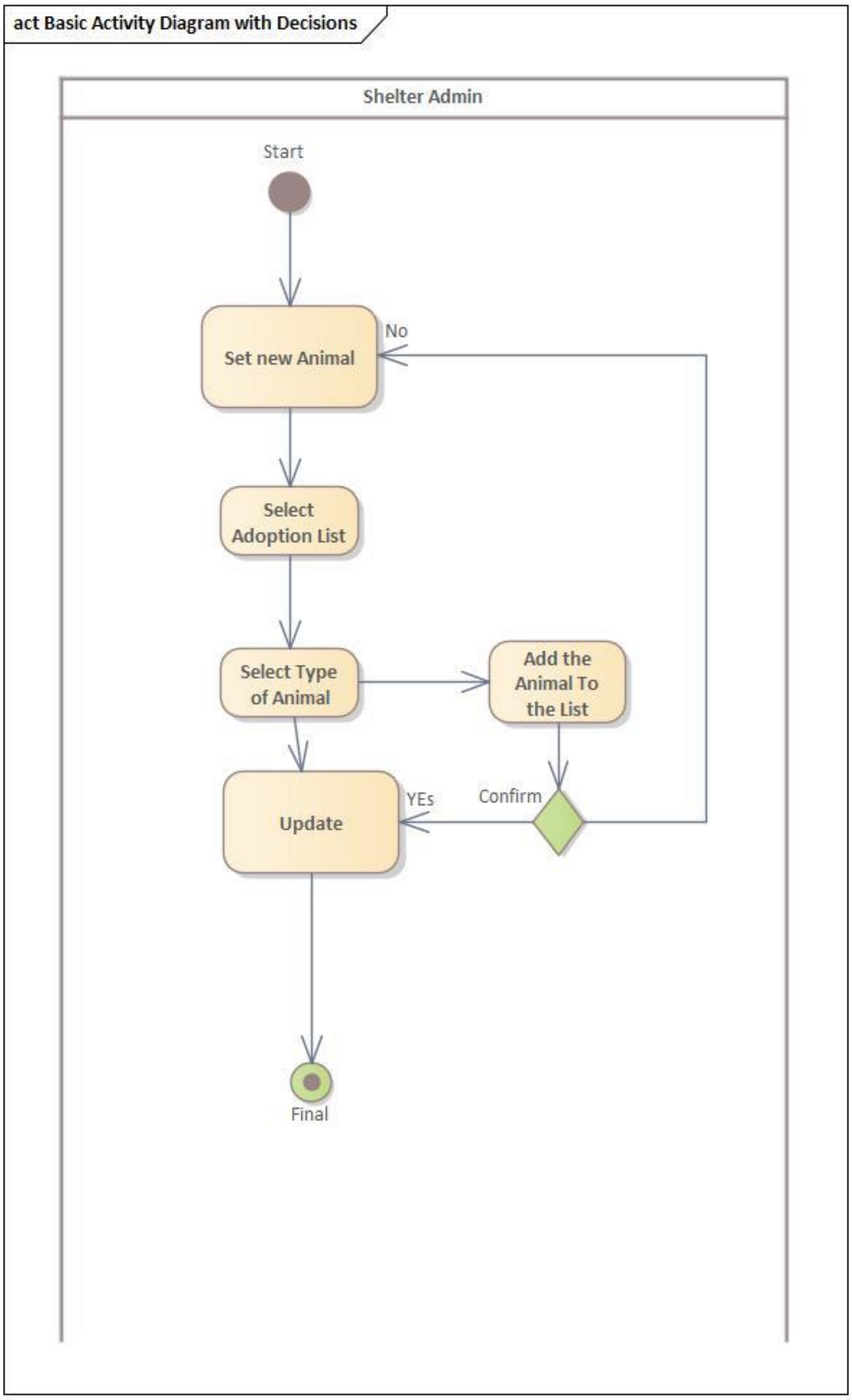


Figure 4.7 Example of Add animal activity diagram

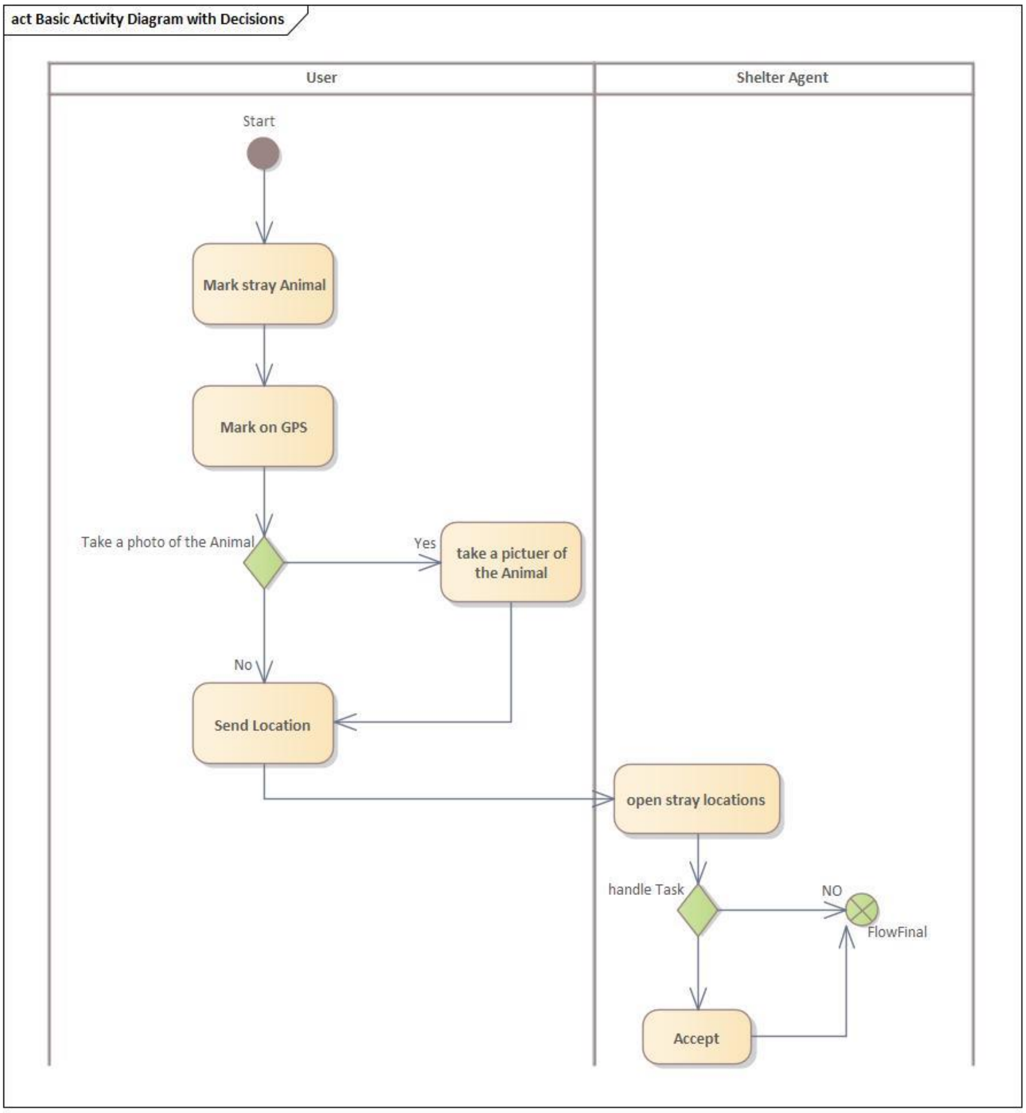


Figure 4.8 Example of Mark Stray animal

4.6 Chapter Summary

As a conclusion of the chapter, the requirement analysis and design for developing the application, in this chapter based on requirements different type of UML defined like sequence diagram for relations, and showing interaction, activity diagram for showing process of completing a task, and both class diagram, and use case diagram to show system functionality, and components.

CHAPTER 5

IMPLEMENTATION AND TESTING

5.1 Introduction

This chapter presents the testing phase of the mobile application for rescuing and adopting homeless animals. The implementation of the application is validated through various testing methods to ensure its functionality, reliability, and user acceptance. This chapter provides an overview of the testing approach and highlights the key testing activities.

5.2 Coding of System Main Functions

User interfaces are the part of an application that users interact with. We make user interfaces usable by following key principles, such as clarity, consistency, efficiency, and accessibility. We test our user interfaces to ensure that they are usable by following key steps, such as user testing, heuristic evaluation, and usability testing.

The code base of an application system is the foundation of its functionality. It includes the back-end and front-end development components that are necessary for the system to operate seamlessly. The code base is structured using best practices to ensure that it is maintainable, scalable, and secure.

5.3 Interfaces of System Main Functions

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

Here is a table that summarizes the key principles and steps for making user interfaces usable:

Table5.1 key principles

Principle	**Description**
Clarity	The user interface should be clear and easy to understand.
Consistency	The user interface should be consistent throughout the application.
Efficiency	The user interface should be efficient. Users should be able to complete their tasks quickly and easily.
Accessibility	The user interface should be accessible to all users, including users with disabilities.
User testing	We test our user interfaces with real users to get their feedback.
Heuristic evaluation	We use heuristic evaluation to identify potential usability problems with our user interfaces.
Usability testing	We use usability testing to measure the usability of our user interfaces.

5.4 Testing

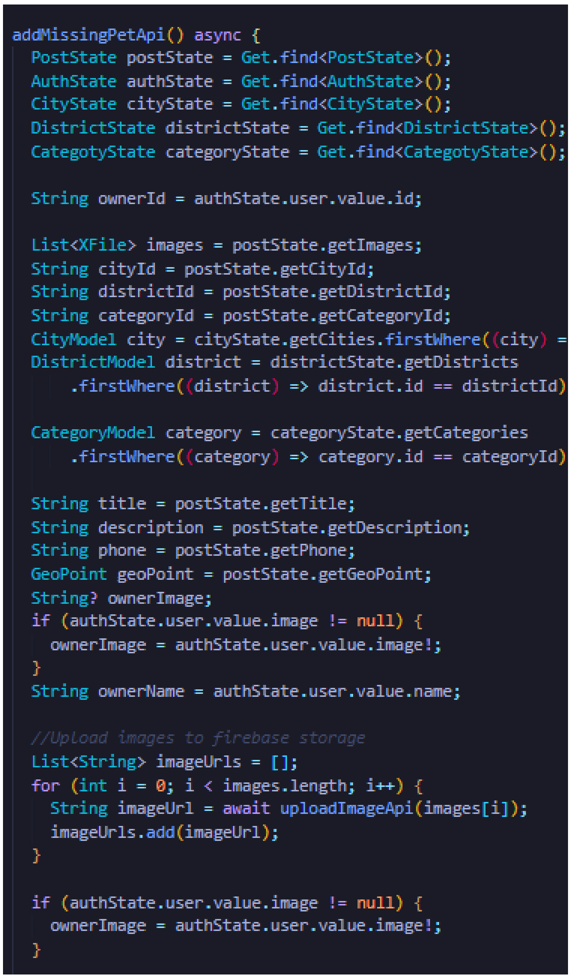
The code base of an application system is the foundation of its functionality. It includes the back-end and front-end development components that are necessary for the system to operate seamlessly. The code base is structured using best practices to ensure that it is maintainable, scalable, and secure.

Table5.2 Test cases

Test Case ID	Test Case Description	Test Steps	Expected Results	Actual Results
TC-1	Login	Open the animal adoption application., Enter the username and password., Click the 'Login' button.	The user should be logged in to the application.	The user was logged in to the application.
TC-2	Register	Open the animal adoption application., Click the 'Register' button., Enter the username, email address, password, and confirm password., Click the 'Register' button.	The user should be registered and logged in to the application.	The user was registered and logged in to the application.
TC-3	Add animal for adoption	Log in to the application., Click the 'Add Animal' button., Enter the animal's name, breed, age, gender, and description., Upload a photo of the animal., Click the 'Add Animal' button.	The animal should be added to the list of animals for adoption.	The animal was added to the list of animals for adoption.
TC-4	Add missing pet	Log in to the application., Click the 'Add Missing Pet' button., Enter the pet's name, breed, age, gender, and description., Upload a photo of the pet., Click the 'Add Missing Pet' button.	The missing pet should be added to the list of missing pets.	The missing pet was added to the list of missing pets.
TC-5	Report animal	Log in to the application., Click the 'Report Animal' button., Select the type of animal, the animal's location, and the animal's condition., Click the 'Report Animal' button.	The animal should be reported to the appropriate authorities.	The animal was reported to the appropriate authorities.
TC-6	Admin Login	Open the Admin Panel., Enter the Username and password, Click Login button	The admin should be logged in to the application.	The admin was logged in to the admin Panel
TC-7	Admin Add Category	Login to the Admin Panel., Click the Add Category button, Enter Category Name., Click add category Button	The category should be added to the list of categories	The category was added to the list of categories

5.4.1 Black box Testing

Black-box testing focused on examining the application's behavior from an end-user perspective without considering its internal implementation. For example, a test scenario involved creating a new user account and verifying that the registration process completed successfully, allowing the user to log in.



```
addMissingPetApi() async {
  PostState postState = Get.find<PostState>();
  AuthState authState = Get.find<AuthState>();
  CityState cityState = Get.find<CityState>();
  DistrictState districtState = Get.find<DistrictState>();
  CategotyState categoryState = Get.find<CategotyState>();

  String ownerId = authState.user.value.id;

  List<XFile> images = postState.getImages;
  String cityId = postState.getCityId;
  String districtId = postState.getDistrictId;
  String categoryId = postState.getCategoryId;
  CityModel city = cityState.getCities.firstWhere((city) =
  DistrictModel district = districtState.getDistricts
      .firstWhere((district) => district.id == districtId)

  CategoryModel category = categoryState.getCategories
      .firstWhere((category) => category.id == categoryId)

  String title = postState.getTitle;
  String description = postState.getDescription;
  String phone = postState.getPhone;
  GeoPoint geoPoint = postState.getGeoPoint;
  String? ownerImage;
  if (authState.user.value.image != null) {
    ownerImage = authState.user.value.image!;
  }
  String ownerName = authState.user.value.name;

  //Upload images to firebase storage
  List<String> imageUrls = [];
  for (int i = 0; i < images.length; i++) {
    String imageUrl = await uploadImageApi(images[i]);
    imageUrls.add(imageUrl);
  }

  if (authState.user.value.image != null) {
    ownerImage = authState.user.value.image!;
  }
```

Figure 5.1 Example Add missing pet function

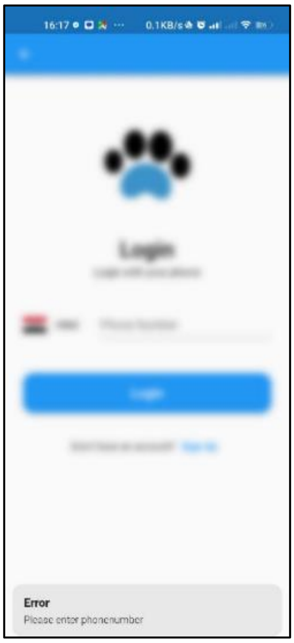
5.4.1.1 System Flow

The testing phase yielded positive results, with the application demonstrating satisfactory performance and meeting the defined acceptance criteria. Functional testing validated the correct behavior of the implemented features, while usability testing highlighted areas for improvement in the user experience.

Performance testing ensured that the application could handle the expected user load efficiently without any significant performance issues. Security testing revealed no major vulnerabilities, and appropriate security measures were implemented to protect user data.

5.4.1.2 Input Output Verification

Manual testing involved real-time interactions with the application to simulate user actions and validate the expected outcomes. For example, a tester manually navigated through the application, submitted an adoption request for a specific animal, and verified that the request was successfully recorded and communicated to the relevant shelter.

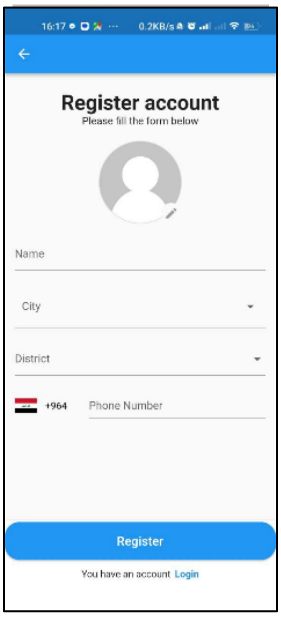


Error
Please enter phonenumber

Figure 5.2 Example of Login input test

Table 5.1 Example of login test case

Test Case Id	**Test Case Description**	**Test Steps**	**Expected Results**	**Actual Result**	**User Target**	**Test Input**	**Test Output**	**What Exist in test**
TC-1-1	Login with invalid credentials	1. add phone number 2. submit 3. verify OTP	The user should login using valid number	The user could not log in the application	Normal User	“0770 1311541”	‘login faild”	A user interface that show login component

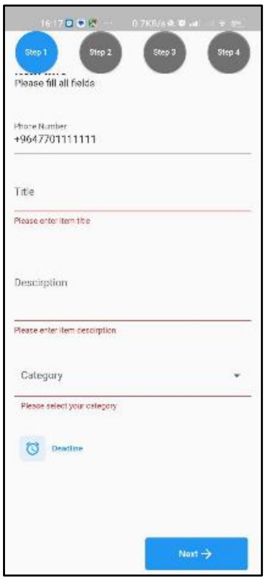

Register account

Please fill the form below

Figure 5.3 Example of Register

Table 5.1 Example of Register testcase

Test Case Id	**Test Case Description**	**Test Steps**	**Expected Results**	**Actual Result**	**User Target**	**Test Input**	**Test Output**	**What Exist in test**
TC-1-2	Register with valid credentials	1. add name,city, Phone,district 2. submit 3. verify OTP	The user should register fill all feilds	The user was Registered to the application	Normal User	“0770 1111111”	‘Registered successful”	User can Register using Ui



Please fill all fields

Phone Number
+9647701111111

Title
Please enter item title

Desciption
Please enter item description

Category
Please select your category

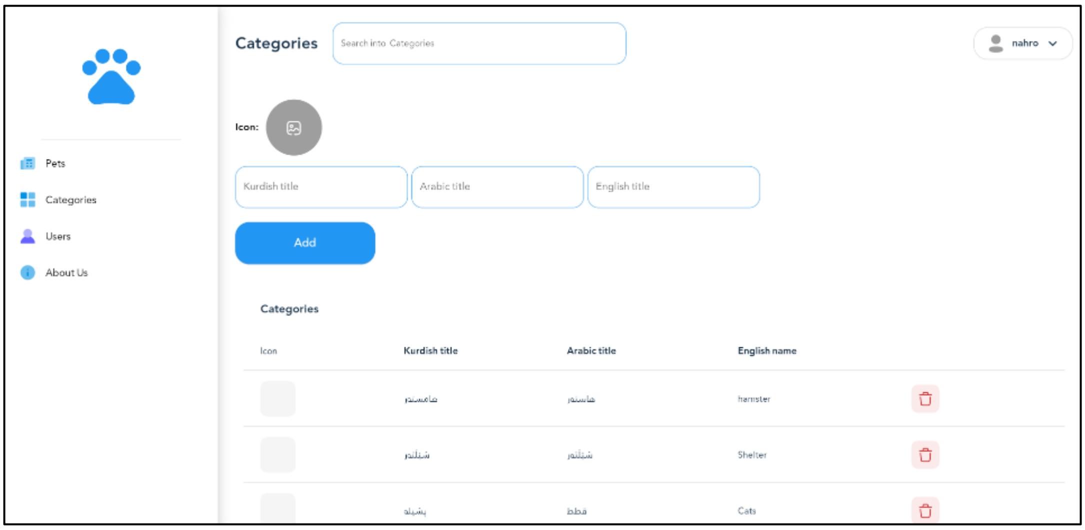
Figure 5.4 Example Add for adopotion

Table 5.1 Example of Add for adopotion

Test Case Id	**Test Case Description**	**Test Steps**	**Expected Results**	**Actual Result**	**User Target**	**Test Input**	**Test Output**	**What Exist in test**
TC-3-	Add for adoption with no data	1. add title,dus,category, ,time. 2. Next img. 3. submit	Add for adoptin by user	The user was submitted null	Normal User	No data	Fail no progress add data”	User can add adoption by Ui

Table 5.1 Example of category test case

Test Case Id	Test Case Description	Test Steps	Expected Results	Actual Result	User Target	Test Input	Test Output	What Exist in test
TC-5	Add category with valid field	1. add cat name in 3 lang. 2. submit	New category name dog	In home page new UI appeared	Shelter admin	Dog’ category	Dog category visible	Admin user interface for add ..



Categories

Icon	Kurdish title	Arabic title	English name	
	هامستەر	هامستر	hamster	
	شێلتەر	شێلتەر	Shelter	
	پشیلە	قطط	Cats	

Figure 5.5 Example Add category

Table 5.1 Example of admin login test case

Test Case Id	Test Case Description	Test Steps	Expected Results	Actual Result	User Target	Test Input	Test Output	What Exist in test
TC-5	Add category with valid field	1. add cat name in 3 lang. 2. submit	New category name dog	In home page new UI appeared	Shelter admin	Dog' category	Dog category visible	Admin user interface for add ..

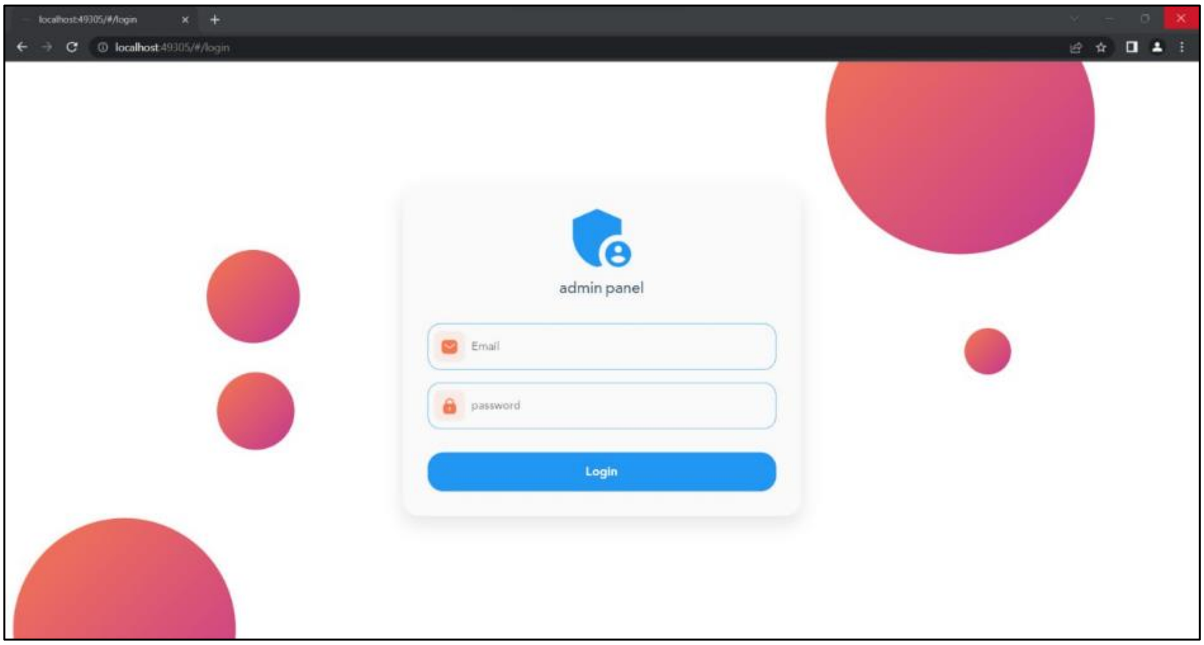
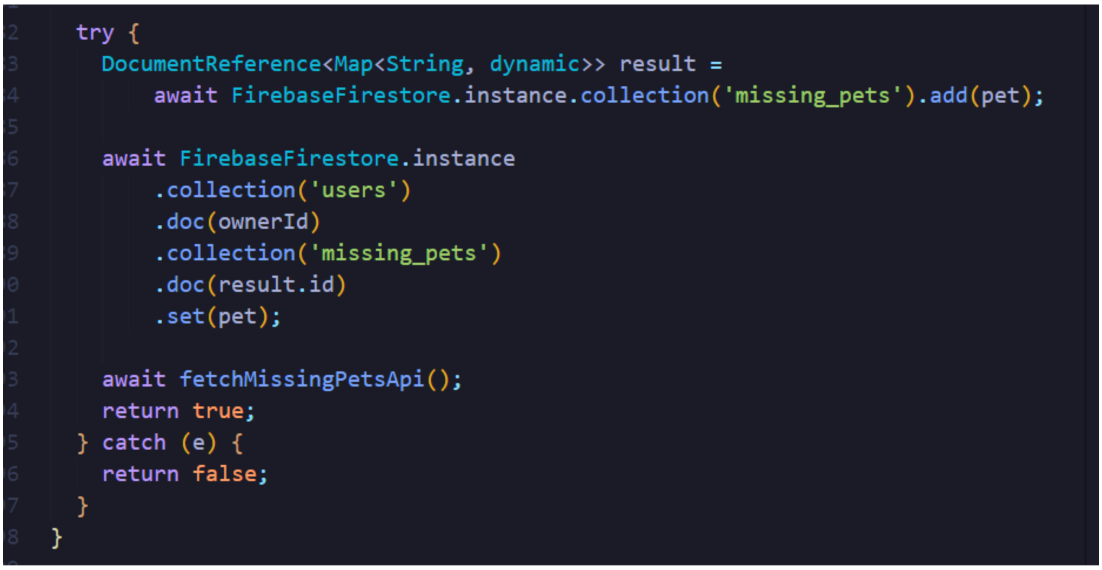


Figure 5.6 Example Add category

5.4.1.3 Error Messages

Video provides a powerful way to help you prove your point. When you click Online Video, you can paste in the embed code for the video you want to add. You can also type a keyword to search online for the video that best fits your document. To make your document look professionally produced, Word provides header, footer, cover page, and text box designs that complement each other. For example, you can add a matching cover page, header, and sidebar.

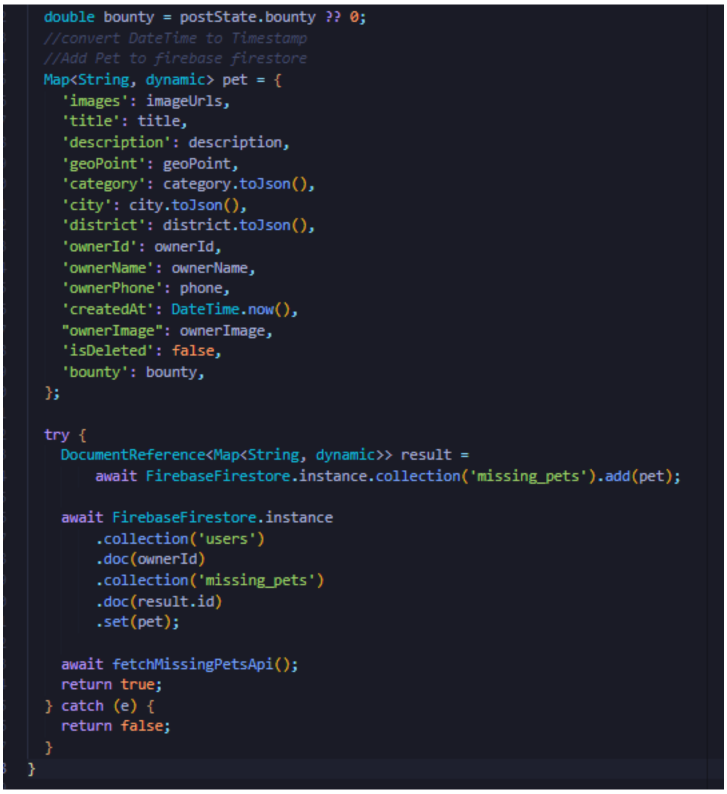


```
  try {
    DocumentReference<Map<String, dynamic>> result =
        await FirebaseFirestore.instance.collection('missing_pets').add(pet);

    await FirebaseFirestore.instance
        .collection('users')
        .doc(ownerId)
        .collection('missing_pets')
        .doc(result.id)
        .set(pet);

    await fetchMissingPetsApi();
    return true;
  } catch (e) {
    return false;
  }
}
```

Figure 5.7 Fetch function

5.4.2 White box Testing

White-box testing involved testing the internal structure and components of the application. Unit testing was performed to validate the correctness of individual modules and functionalities. For example, a unit test was conducted to ensure that the function responsible for adding a new animal for adoption to the database was working properly.



```
  double bounty = postState.bounty ?? 0;
  //convert DateTime to Timestamp
  //Add Pet to firebase firestore
  Map<String, dynamic> pet = {
    'images': imageUrls,
    'title': title,
    'description': description,
    'geoPoint': geoPoint,
    'category': category.toJson(),
    'city': city.toJson(),
    'district': district.toJson(),
    'ownerId': ownerId,
    'ownerName': ownerName,
    'ownerPhone': phone,
    'createdAt': DateTime.now(),
    "ownerImage": ownerImage,
    'isDeleted': false,
    'bounty': bounty,
  };

  try {
    DocumentReference<Map<String, dynamic>> result =
        await FirebaseFirestore.instance.collection('missing_pets').add(pet);

    await FirebaseFirestore.instance
        .collection('users')
        .doc(ownerId)
        .collection('missing_pets')
        .doc(result.id)
        .set(pet);

    await fetchMissingPetsApi();
    return true;
  } catch (e) {
    return false;
  }
}
```

Figure 5.8 Accuracy of data

5.4.3 User Testing

User acceptance testing involved real users and stakeholders assessing the application to determine its readiness for deployment. Users were provided with test scenarios and asked to perform various tasks within the application. For example, users were requested to register an account, add an animal for adoption, and submit an adoption request. Feedback and observations were collected to gauge user satisfaction, identify any remaining issues, and make necessary refinements.

We recently conducted a user acceptance testing (UAT) survey for our animal rescue application. The survey was designed to assess how comfortable users are with the application and to identify any potential usability issues.

The survey was administered to a sample of users who had recently used the application. The survey included questions about the following:

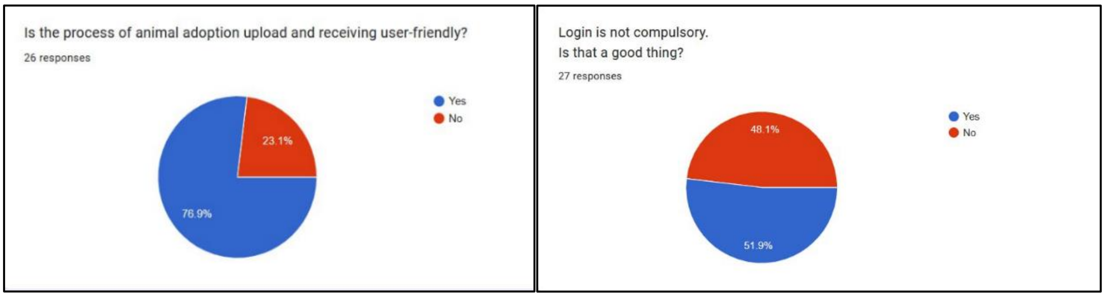


Figure 5.9 User survey

Usability testing aimed to assess the user experience and interface design of the application. For example, testers evaluated the ease of navigating through different screens, the intuitiveness of user interactions, and the clarity of information presentation. They provided feedback on areas that required improvement, such as streamlining the adoption process or enhancing the visibility of important features.

Performance testing evaluated the application's responsiveness, scalability, and resource usage under different load conditions. For example, performance tests were conducted to measure the application's response time when handling a large number of simultaneous adoption requests or loading animal listings with substantial data. Security testing aimed to identify vulnerabilities and ensure the application's data protection measures. For example, security tests were conducted to check if user data was properly encrypted during transmission and stored securely in the database. The application was also tested for potential vulnerabilities like login.

5.5 Chapter Summary

The comprehensive testing approach employed in the development of the mobile application for rescuing and adopting homeless animals ensured its functionality, reliability, and user acceptance. Identified issues were addressed promptly, and user feedback played a vital role in refining the application.

The successful completion of the testing phase paves the way for the subsequent chapters, which will focus on the system requirements, design aspects, and the overall conclusion of the project.

CHAPTER 6

CONCLUSION

6.1 Introduction

The proliferation of wild animals, particularly dogs and cats, roaming the streets in large numbers has posed significant challenges for communities in recent years. This issue, explored in the initial chapter, specifically affects people living in cities and presents a particular concern for women and children. Various problems associated with stray dogs are outlined in detail.

To address this problem, it is crucial for organizations and animal shelters to actively control and rescue these animals, mitigating the rapid increase in the population of stray dogs and cats within urban areas. By doing so, these organizations aim to save these animals from the risk of starvation and other perils they face. However, due to the absence of adequate facilities and tools to streamline their operations, animal organizations struggle to effectively control the number of stray animals. Their current methods primarily rely on receiving phone calls or deploying agents for manual searches, limiting their ability to address the issue comprehensively.

The chapters within the context can be summarized as follows:

Chapter 1: This chapter focuses on identifying the domain problem, emphasizing the significance of developing a proposed solution to address the issue at hand. It provides an overview of the problem and highlights the specific problems the proposed solution aims to solve.

Chapter 2: In this chapter, case studies are conducted to analyze the existing system and compare it with similar systems. The goal is to identify the missing features necessary for solving the current problem that are not currently provided.

Chapter 3: Building on the research and comparisons made in the previous chapter, this chapter delves into selecting the most suitable methodology for developing the proposed solution. It outlines the different phases required for implementation and provides a justification for the chosen methodology.

Chapter 4: The focus of this chapter lies in defining the requirements and creating diagrams that illustrate the structure and behavior of the proposed solution. It showcases the various functionalities of the system and provides an overview of how it operates.

6.2 Achievement of Project Objectives

Following the collection of requirements from stakeholders to gain a comprehensive understanding of the existing problems, the next step involves analysing these requirements and conducting a thorough comparison with existing systems. Through this process, the clarity of the intended solution becomes more apparent, specifically in addressing the issue of homeless animals.

To ensure effective development of the proposed solution, an architectural design pattern has been devised. This pattern serves as a blueprint for the overall structure and organization of the system. Additionally, a detailed design has been constructed to provide insights into the specific behaviours and structures that the proposed system will embody. This includes determining how the system should function and operate in order to effectively address the identified problems.

By engaging in the architectural design and detail design phases, the project gains a clear roadmap for the development process. The design patterns and specifications outline the necessary components, functionalities, and interactions required for the proposed system. This ensures that the system is developed in a manner that aligns with the identified problems and addresses the stakeholders' requirements effectively.

6.3 Suggestions for Future Improvement

After completing the Project Start-up Phase (PSM2) in Flutter, my focus will be on improving functionality and performance in the mobile application. I will strive to reach advanced levels of mobile app development by learning and mastering advanced techniques specific to Flutter.

Additionally, I will invest time and effort in analysing the project to provide valuable suggestions for organizational improvement. By leveraging my expertise and understanding of the project, I aim to contribute to the growth and efficiency of the organization.

In summary, my main objectives after PSM2 are to enhance functionality, optimize performance, investing in my project and offer suggestions for organizational.

REFERENCES

PETA UK. (2022) ‘Animal homelessness: Animals are not ours to abuse’, Retrieved January 14, 2023, from https://www.peta.org.uk/issues/animals-not-abuse/homelessness/

War Paws. (n.d.) ‘The shelter in Iraq’, Retrieved January 14, 2023, from https://warpaws.org/the-shelter-in-iraq/

KOARP. (n.d.) ‘About’, Retrieved January 14, 2023, from https://www.koarp.org/about

Pak Organization. (n.d.) ‘[Facebook page]’, Retrieved January 14, 2023, from https://www.facebook.com/PAK-Organization-1935362420075766/

Pedamkar, P. (2021) ‘Iterative methodology: Iterative methodology phases: Importance’, EDUCBA. Retrieved January 14, 2023, from https://www.educba.com/iterative-methodology/

Francino, Y. (2011) ‘What is iterative development? Definition from TechTarget’, Software Quality. Retrieved January 14, 2023, from https://www.techtarget.com/searchsoftwarequality/definition/iterative-development

Udoagwu, K. (2022) ‘How to carry out a requirements analysis’, Wrike. Retrieved January 14, 2023, from https://www.wrike.com/blog/how-carry-out-requirements-analysis/#What-is-a-requirements-analysis

Ryan, R. (n.d.) ‘A complete guide on pet apps mobile app development’, Retrieved May 20, 2021, from https://www.webiotic.com/a-complete-guide-on-pet-apps-mobile-app-development

Hamilton, T. (2023, May 6) ‘7 principles of software testing with examples’, Guru99. Retrieved from https://www.guru99.com/software-testing-seven-principles.html

Testmatick. (2016, January 21) ‘4 main principles of mobile testing as a service’, Retrieved from https://testmatick.com/4-main-principles-of-mobile-testing-as-a-service

Appendix A Software Requirement Specification

Software Requirements Specification

Rescue and Adoption Homeless Animals

Version 1.0

Printing Date

Department and Faculty

Computer Science-Software Engineering

Rebaz Awat Mustafa

REVISION PAGE

a. Overview

Describe the content of the current version.

b. Target Audience

State the targeted audience.

c. Project Team Members

List the team members and respective assigned module.

d. Version Control History

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**
<Current Version>			

1. INTRODUCTION

We have to avoid problems of dogs and cats, we could rarely see them on streets, and animal shelters were easily capable for collecting, controlling their growth and locating their locations lately number of stray anima increased in a way that Obviously we can see stray animals on the streets inside the city that made some serious issue for society.

Numbers of dogs bothered people a lot especially at night, also some events became common to happen which is attacking children by dogs, and as a result of this attack sometimes it leads to killing dogs by people, also Even cats get hurts because of stealing foods.

Having that number of stray animals made some people hate homeless animals. animal shelters not able to control catching them and preventing rising number of these dogs, because number of stray animals led to pet overpopulation, and this is the reason why they spread throughout the city and there is not an app that we can mark location of stray animals and send it to shelters to come for them.

One of the proposed solutions that is necessary to focus is trying to make process of reduce number of homeless animals inside the city by making a connection between organizations, shelters, and people to help each other for reducing number of stray animals, preventing them from getting animal diseases and dying from starving, based on a report In Aug/2022 the animal protection organization announced that more than 400 dogs died because of illnesses and hunger during that month in Sulaymaniyah City. For this project a mobile application will be build, and it should be useful to help gathering stray animals faster, feeding, and rescuing them.

1.1 Purpose

software requirements specification (SRS) document in this project describes what the software will do and how it will be expected to be developed. This document focused on the functionality of the product and its requirements to meet stakeholders need. Through this documentation we know that what are information need to be gathered about the product, and this is lead to make a better understanding about the software. Purpose of this Document is to show information about the product that is required before design phase like, getting information about goal of the product, audience, hardware requirements, software requirements, and functionalities of the product.

1.2 Scope

This project's scope is to overcome problems related to homeless animals mentioned above in detail like reducing stray animals inside cities, protecting them from diseases, and providing new home. And the solution proposed is to gather all homeless animals in shelters, then providing a home for them. The proposed application provides some features that should make an effective change at searching for stray animals much more accessible comparing to classic way through the app. To make the application be used by everyone anywhere, the application will be published as a mobile application. The methodologies for this project is agile Methodology, and the app will be done in 4 phases. My application won't include advertising, or any kind of social marketing..

1.3 Definitions, Acronyms and Abbreviation

SRS	-	Software Requirement Specification

1.4 References

Udoagwu, K. (2022) ‘How to carry out a requirements analysis’, Wrike. Retrieved January 14, 2023, from https://www.wrike.com/blog/how-carry-out-requirements-analysis/#What-is-a-requirements-analysis

Ryan, R. (n.d.) ‘A complete guide on pet apps mobile app development’, Retrieved May 20, 2021, from https://www.webiotic.com/a-complete-guide-on-pet-apps-mobile-app-development

1.5 Overview

This SRS document consists of three major parts, the introduction, overall description and specific requirements. The introduction describes the purpose of this SRS document and the scope of the SMART system. Next, the overall description section describes the overview of the entire system. This section describes the background of the requirements that aims to facilitate the understanding of the intended audience of this document. Lastly, the third section, specific requirements will describe the requirements specification and interfaces in more a detail level. This section aims to enable the designers and testers to understand and satisfy the requirements of the system.

2. OVERALL DESCRIPTION

Use Case Diagram: A graphical representation known as a use case diagram is used to determine a system's needs and how they relate to the users of the system.

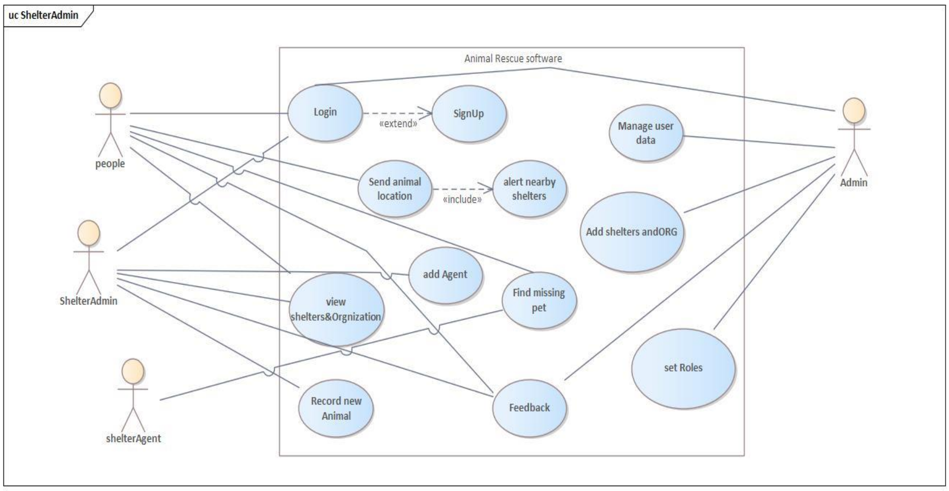


Figure A.1: Use Case Diagram of <Animal Rescue App>

2.1 Product Perspective

The perspective of the app is to help controlling, collecting, and rescuing animals easier, by providing some functionalities as showing in the above use case diagram.

A use case diagram's key advantage is that it aids in clearly explaining a system's operation to various stakeholders, including consumers, project managers, and developers, and below figure is use case diagram of the system

2.1.1 System Interfaces

For the software we have UI for 3 different type of user, a guess, normal users have account, UI for organizations, and shelters also the admin. Each has specific task and features but most functionalities are the same.

2.1.2 User Interfaces

The software user interface should provide what users need, and the software have to be usable in order different type of user can understand functionalities of the software. In user interface we provide registration, login , visit as guess, having lots of buttons not preferable by users so each user interface include some features and functionality that user do not feel about complexity of the UI.

2.1.3 Hardware Interfaces

The Rescue Animal software is mobile application, and the app Since the SMART system is a web based system, the system shall be able to send requests to the server whenever data is needed. Besides, the system shall be able to receive responses from the server.

2.1.4 Software Interfaces

Database management system (DBMS) is required to handle the data that is related to the system. MySQL is required for management (query, update, delete) of data that is involved with the SMART system. The following are the

The server needs to have a server operating system in order to handle requests from clients. The server operating system is an operating system that operates within the server side.

2.1.5 Communication Interfaces

The SMART system shall communicate with the server and external online banking service server through HTTPS protocol.

2.1.6 Memory

Maximum 50mb

2.1.7 Operations

(a) *Data inserting- Database;*

(b) *Data processing support functions;*

(c) *Dara Retrieving from database*

2.1.8 Site Adaptation Requirements

Adaption requirement of the app: it support android V10 and above Minimum 100mb of ram required, size of display not matter the app is responsive.

2.2 Product Functions

Based on the above use case we can see many functionalities in the app but, this part consist of the major functionalities in the software.

First functionality is marking stray animals. Through this method users can add location of the stray animal to the stray list and send it to nearby shelters.

Second functionality is adding animal for adoption this process will be handled by shelter admins, the process is for donating not needed animals in the shelters.

3rs and last important major functionality is adding agent by shelter-Organization admins, this to add their employee to the app with a different role comparing to normal users.

2.3 User Characteristics

User characteristics refer to the specific attributes or traits of the individuals who will be using the software. These characteristics can include things like their level of technical expertise, specific job roles or tasks that they will be performing with the software, and any physical or cognitive limitations that may impact their ability to use the software. In a software requirements specification (SRS), user characteristics are typically described in a section dedicated to user requirements. This section should provide a clear and detailed understanding of the user population, including information on their demographics, skills, and any other relevant information that may impact the design and development of the software.

2.4 Constraints

Some constraint of the app:
Onley 3 type of users are allowed in the system.
Only jpg, and jpeg images allowed to be uploaded.
Shelter admins can add shelter agents to their organization by inserting user email only.
Users cannot create Organization account only general admin can.

2.5 Assumption and Dependencies

There are several factors that effected requirement of the system. First rising number of stray animals made serious problem to people, second organizations complement, none of those organization is supported by government, and last factor is people no one care about problem of stray animals so it effected on the domain in massive way.

2.6 Apportioning of Requirements

There are some features that may delay to later version like adding shops of animals, adding veterinary sections.

3. SPECIFIC REQUIREMENTS

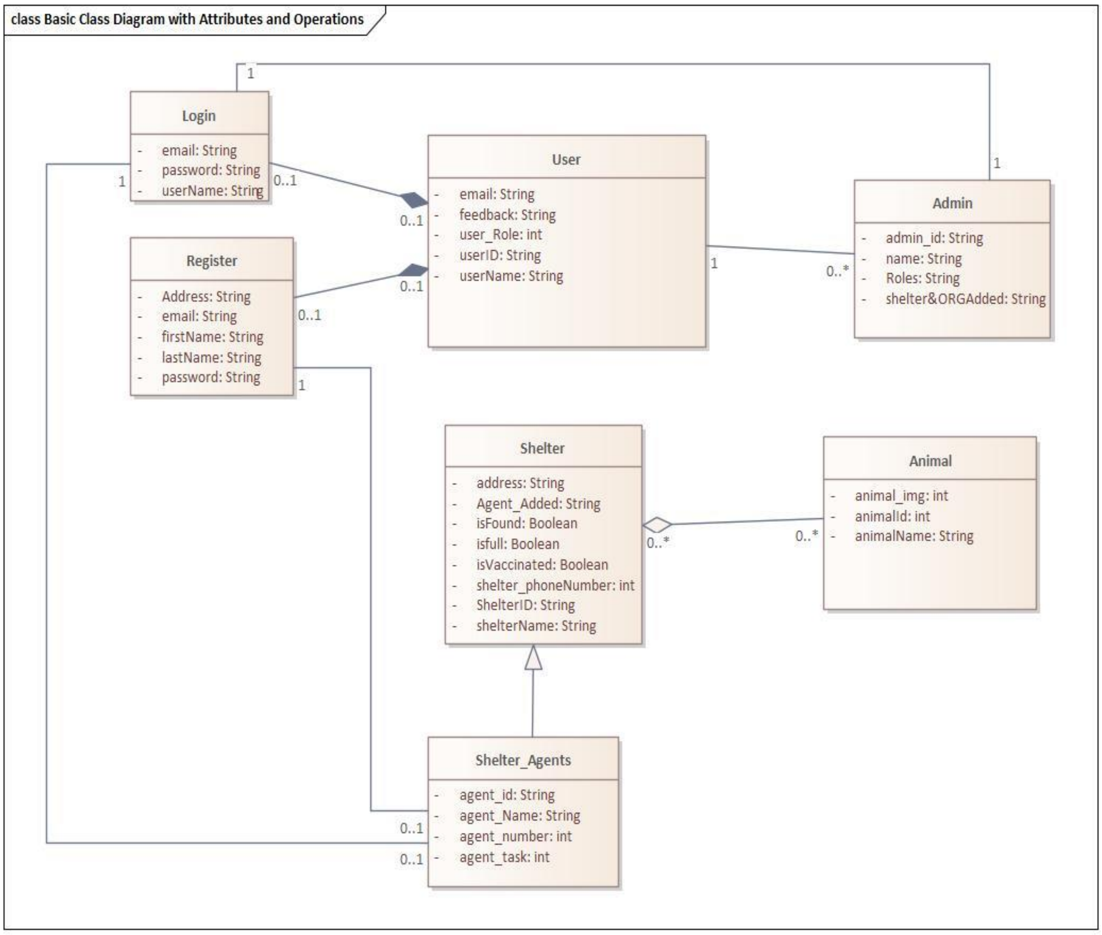


Figure A.2: Domain Model of <Rescue Animal>

3.1 System Features

3.1.1 Module <add animal for adoption >

State briefly the functional requirements (use cases) that are available in this module. Better to include the diagram of the specific module (or the example of Customer Support System – by subsystem, see example below) from the overall use case diagram in Figure 2.1.

3.1.1.1 UC001: Use Case <Add animal for adoption >

This use case is belonged to shelter admin and the functionalities is to add new animal for adoption that is not needed by the shelter, and the operations is just select the animal and add to adoption list.

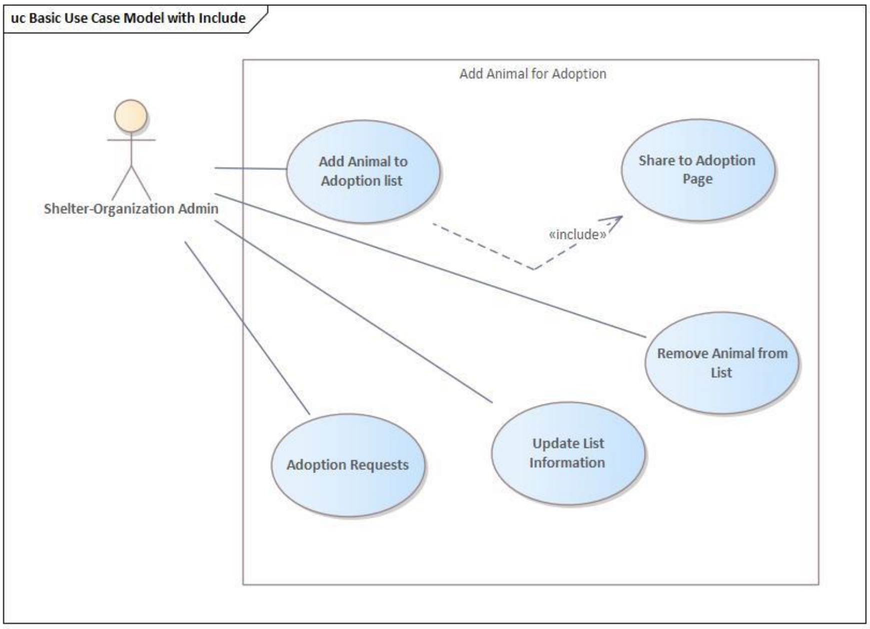


Figure A.3 Add animal for adoption

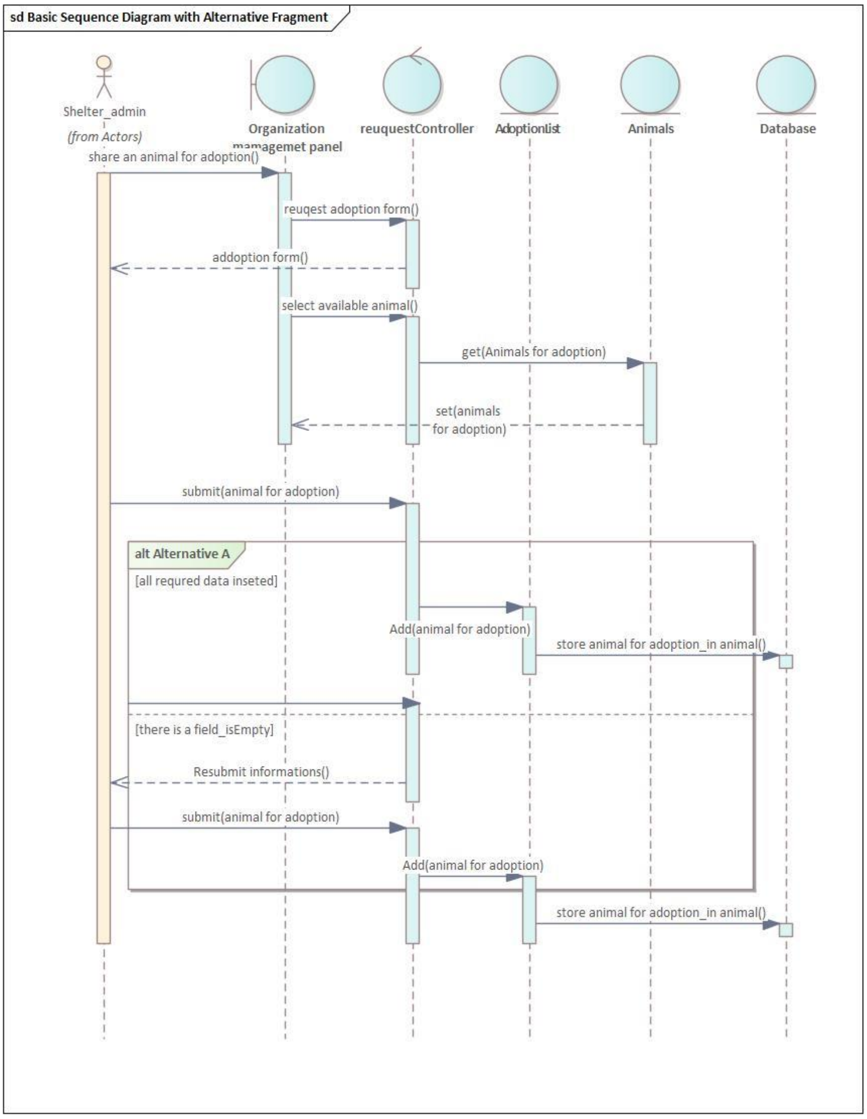

sd Basic Sequence Diagram with Alternative Fragment
Shelter_admin
(from Actors)
Organization mamagemet panel
reuquestController
AdoptionList
Animals
Database
share an animal for adoption()
reuqest adoption form()
addoption form()
select available animal()
get(Animals for adoption)
set(animals for adoption)
submit(animal for adoption)
alt Alternative A
[all requred data inseted]
Add(animal for adoption)
store animal for adoption_in animal()
[there is a field_isEmpty]
Resubmit informations()
submit(animal for adoption)
Add(animal for adoption)
store animal for adoption_in animal()

Figure A.4 System Sequence Diagram of <Add animal for adoption

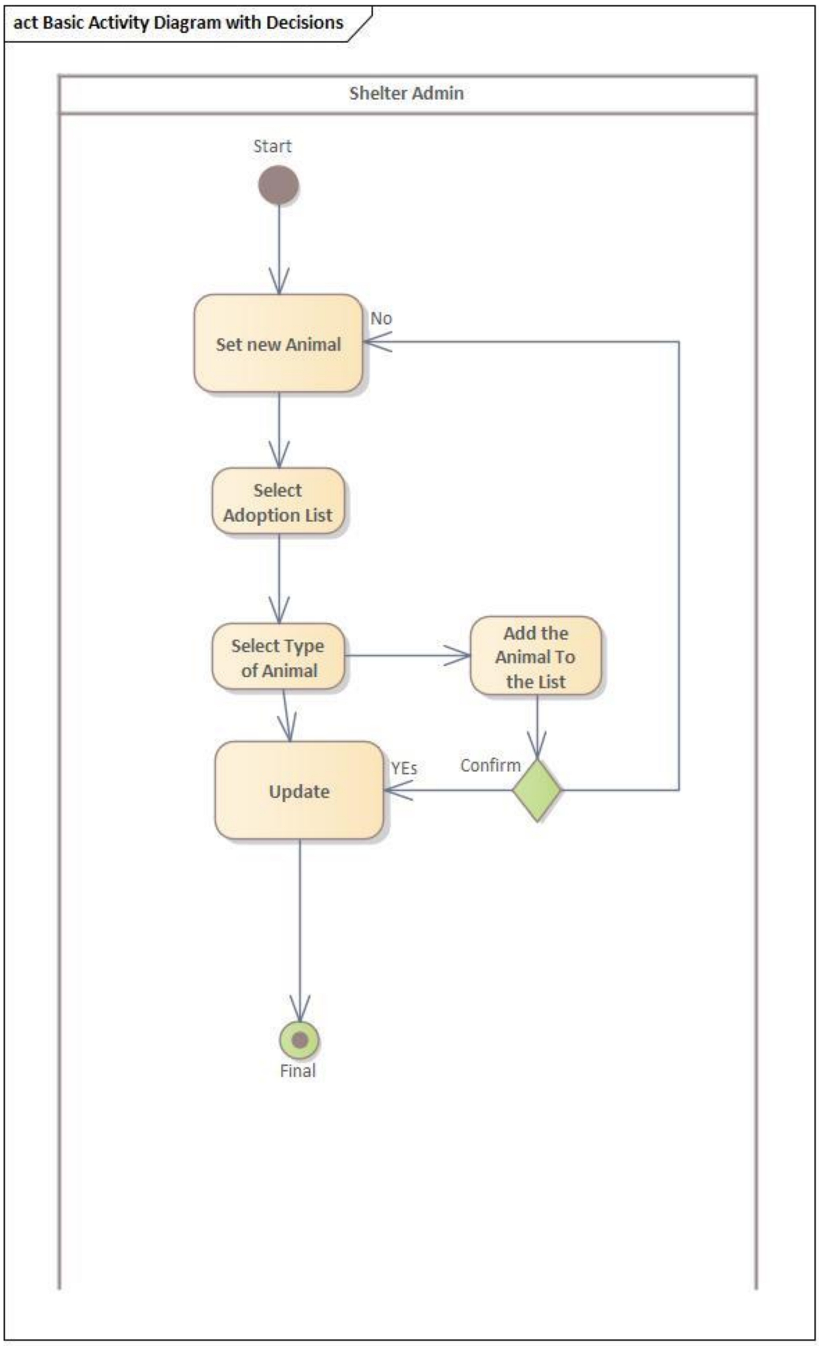


Figure A.5: Activity Diagram of <Add animal for adoption>

3.1.1.2 UC002: Use Case <Add Agent >

This use case is responsible to add new agent for the organization by the shelter admin by adding user account and fill requirement to set role of selected use to agent.

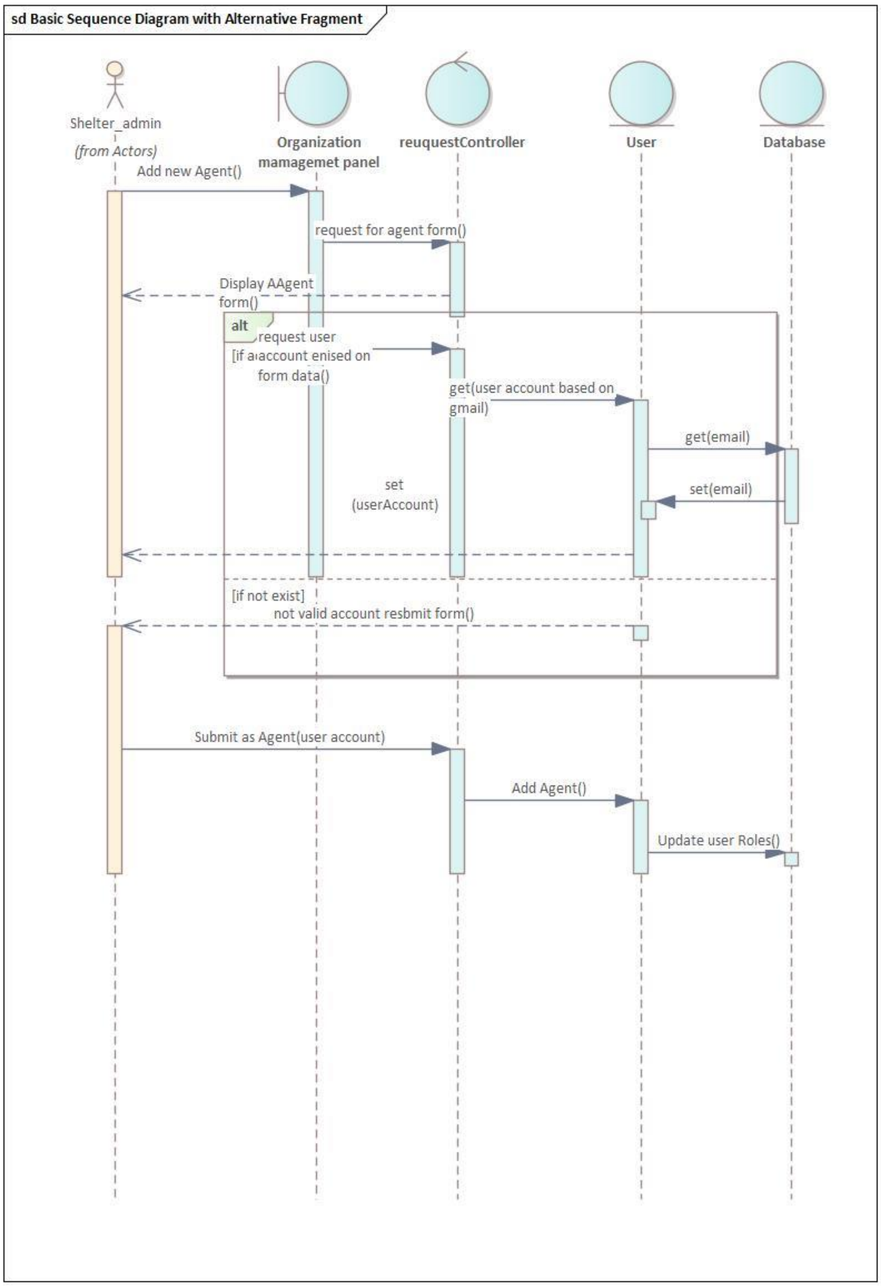


Figure A.6 Add new Agent Sequence Diagram

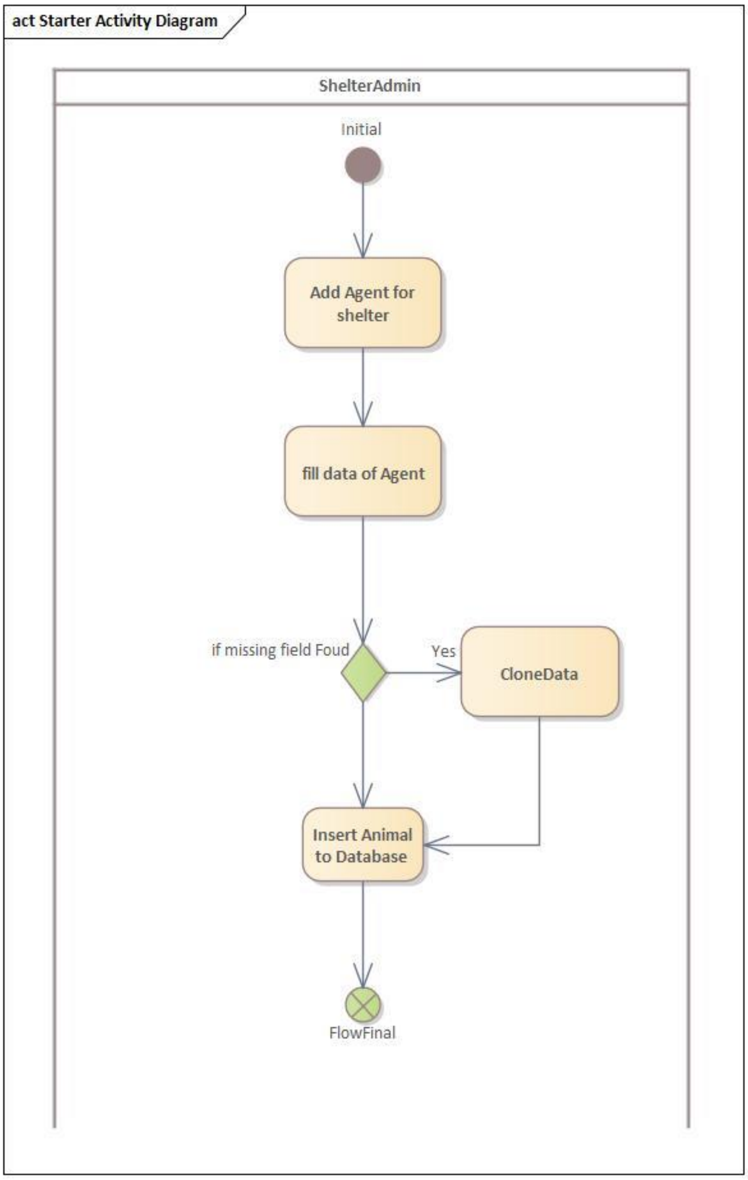


Figure A.7 Activity diagram for add new agent

3.1.1.3 UC003: Use Case <mark stray animal>

This use case represents the main process when a person sees a stray animal, and press mark. After applying operations location of the animal will be sent to nearby shelters.

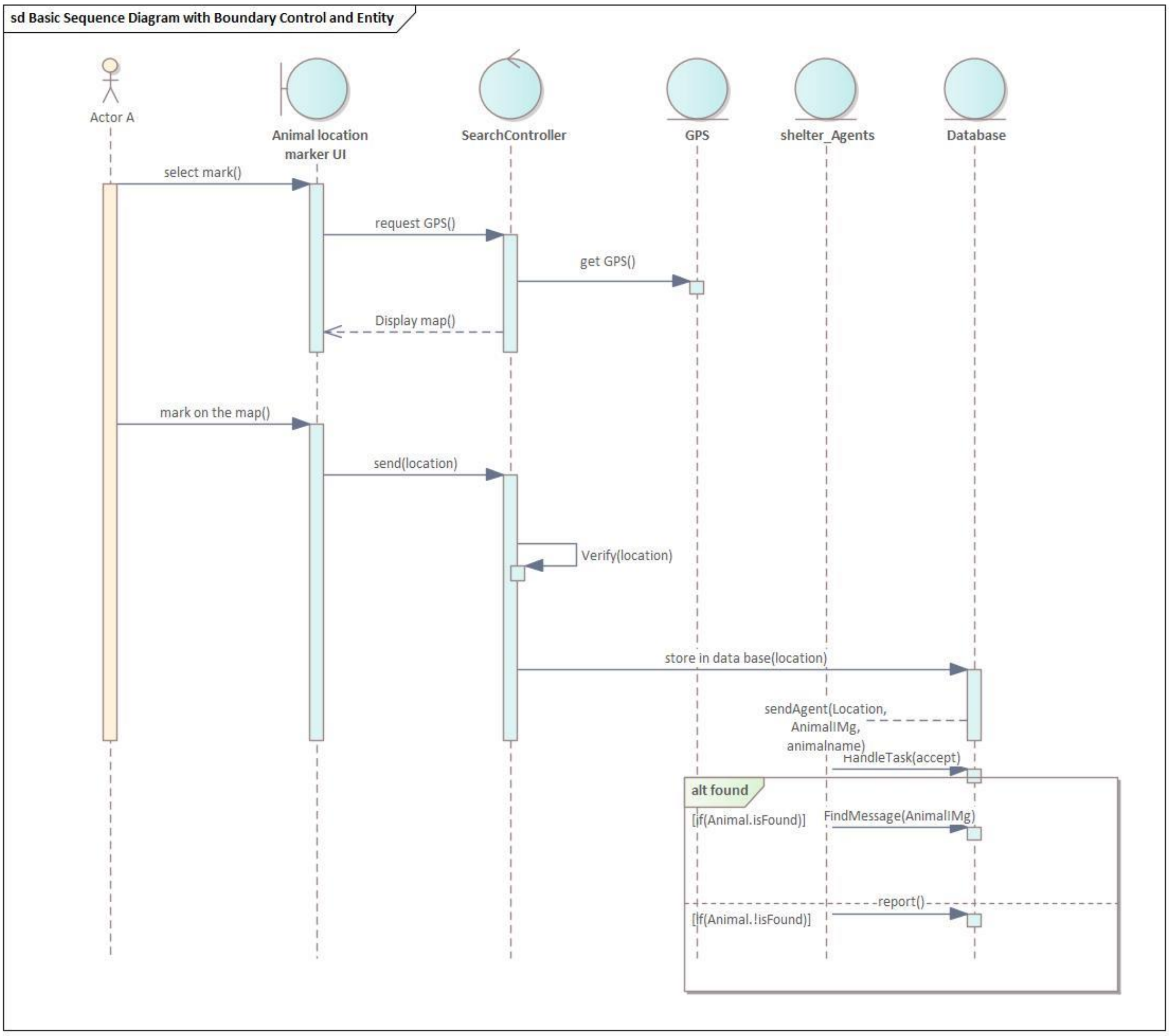


Figure A.8 Sequence Diagram for mark-stray animals

3.2 Performance Requirements

- The mobile application UI has to be responsive to all different type of android and IOS devices like mobile phone, and I pad.
- the app should log in within 5 second.
- any changes in the app have to be far away from ripple effect.
- The system should display the current and latest status of the application.
- The system should store the application form into the application form database after the applicant has chosen to submit it.

3.3 Design Constraints

The interface design should be simple and not complex in order to user understand. The app is connected to Firebase database through API. All functionalities, UI, and API will be written using dart language.

3.4 Software System Attributes

Indicate any specific attributes that the customers/users request such as system must be attractive and easy to use for any specific customers.

3.5 Other Requirements

State here other quality characteristics or non-functional requirements for either customers/users or developers such as adaptability, flexibility, interoperability, maintainability, portability, reliability, reusability and usability.

Appendix B Data Design

SCSJ3323: Software Design and Architecture

Software Design Document

Rescue and Adoption homeless Animals

Version 1.0

Printing Date

Department and Faculty

Prepared by: <Rebaz Awat Mustafa>

REVISION PAGE

e. Overview

The content is about developing a proposed app for solving problem regarding stray animals. Current version 1.0

f. Target Audience

Shelters, organizations,people

g. Project Team Members

Rebaz Awat

h. Version Control History

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**
<1.0>	**Rebaz Awat**		

1. INTRODUCTION

1.1 Purpose

software requirements specification (SDD) document in this project describes descrie architectural style and architectural design pattern, and design pattern. This document focused structure and behavior the product and its requirements to meet stakeholders need. Through this documentation we know that how to develop the software, and this is lead to make a better understanding about the software. Purpose of this Document is to show information about the product that is required before Construction phase like, getting information about goal of the product, audience, hardware requirements, software requirements, and functionalities of the product.

1.2 Scope

This project's scope is to overcome problems related to homeless animals mentioned above in detail like reducing stray animals inside cities, protecting them from diseases, and providing new home. And the solution proposed is to gather all homeless animals in shelters, then providing a home for them. The proposed application provides some features that should make an effective change at searching for stray animals much more accessible comparing to classic way through the app. To make the application be used by everyone anywhere, the application will be published as a mobile application. The methodologies for this project is agile Methodology, and the app will be done in 4 phases. My application won’t include advertising, or any kind of social marketing.

1.3 Overview

This SRS document consists of three major parts, the introduction, overall description and specific requirements. The introduction describes the purpose of this SRS document and the scope of the SMART system. Next, the overall description section describes the overview of the entire system. This section describes the background of the requirements that aims to facilitate the understanding of the intended audience of this document. Lastly, the third section, specific requirements will describe the requirements specification and interfaces in more a detail level. This section aims to enable the designers and testers to understand and satisfy the requirements of the system.

1.4 Architecture Style and Rationale

Architecture Style and Rationale Because of our project is mostly focus on interaction of the software with the system, so we focus more on user interfaces, and users really want to use a usable software in order to feel comfortable, because of this reason we selected MVC as a good pattern for our project. Below points are reasons that we get benefit from MVC pattern: It is much simpler to split and arrange the functionality of web applications into large-scale apps since the code is divided into three components. The key advantage of using the MVC pattern is that it makes it simpler to find certain code portions and quickly add new functionality. The MVC pattern offers the developer a guide on how to translate their ideas into code, which is useful during the early design phase of the program. Additionally, it's an excellent approach to lessen code duplication and make program maintenance easier. With MVC, new views may be added and updated without ever changing the overall architecture. This increases the scalability and flexibility of the application.

1.5 Architecture Model

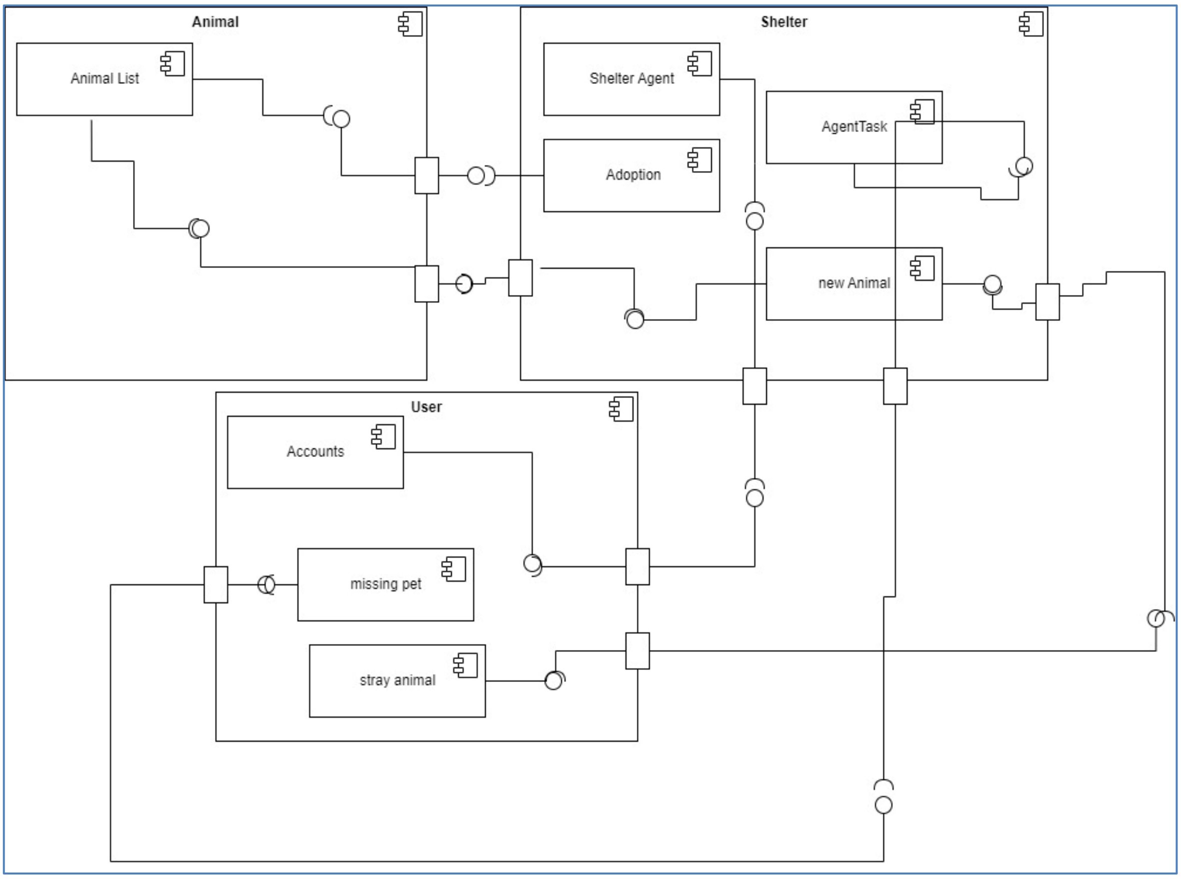


Figure B.1: Component Model of <Name of the System>

1.6 Use Case Diagram

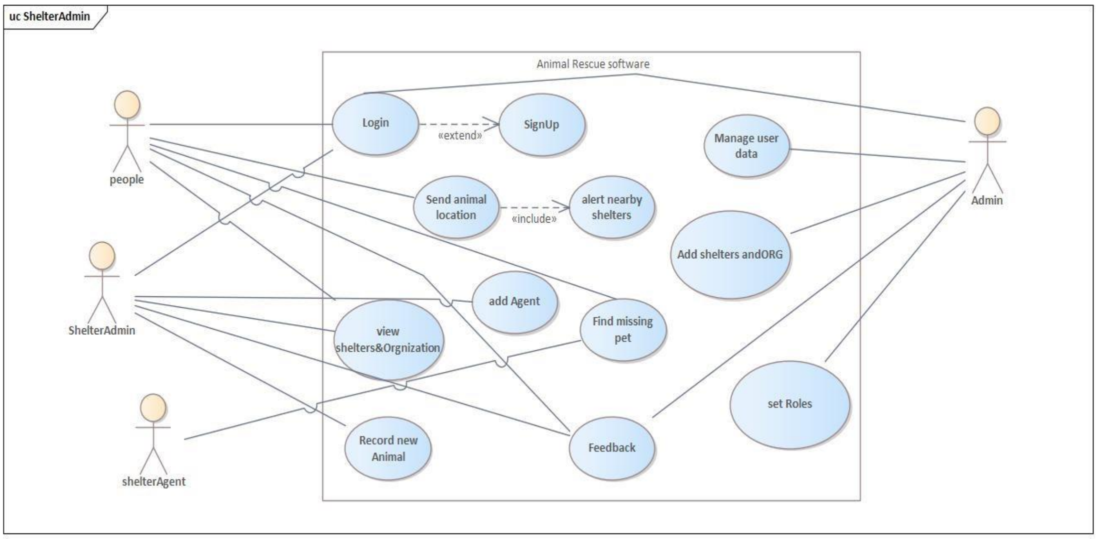


Figure B.2: Use Case Diagram of <Name of the System>

2. DETAILED DESCRIPTION OF COMPONENTS

2.1 Complete Package Diagram

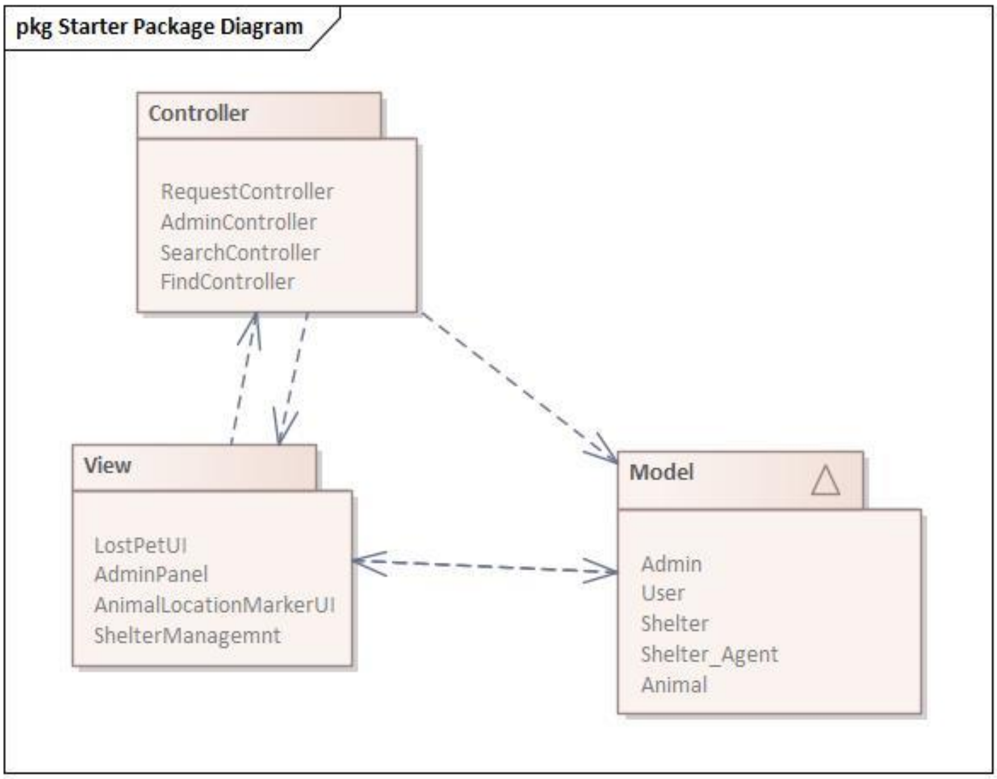


Figure B.3 : Subsystem of <Name of the System>

2.2 Component Model

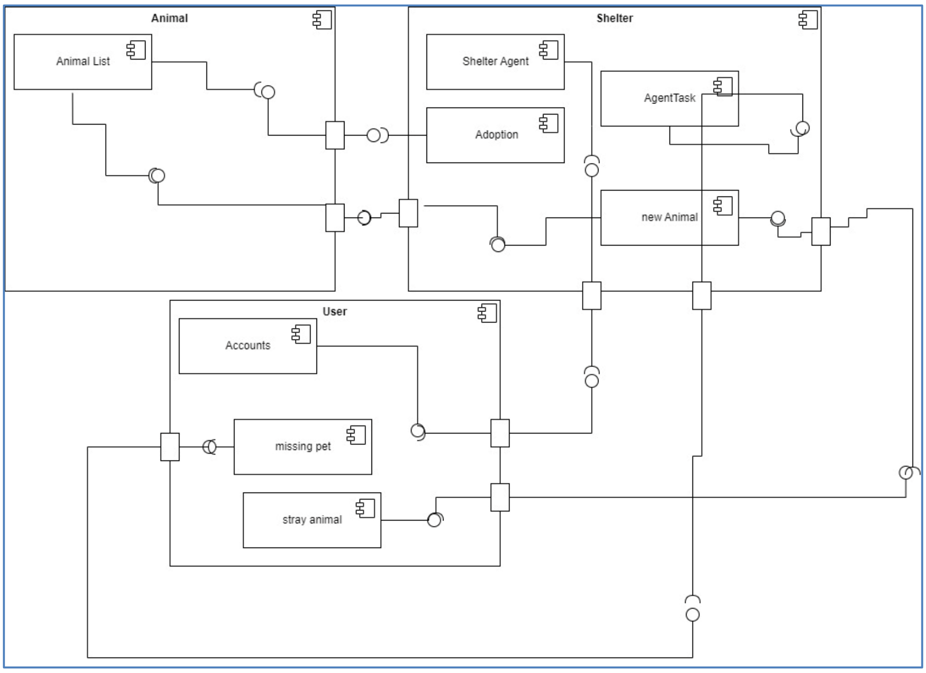


Figure B.4 : Component Diagram of <Name of the System>

2.3 Detailed Description

detailed description refers to a thorough explanation of the design and functionality of a software system or component. This can include information about the system's architecture, data structures, algorithms, interfaces, and other technical details. A detailed description is often used in documentation and technical specifications to provide a clear understanding of the software for developers, stakeholders, and other interested parties. It is also used to help ensure that the software is developed and tested in accordance with the requirements and specifications.

2.3.1.1 P001: Package <Shelter Admin Operation>

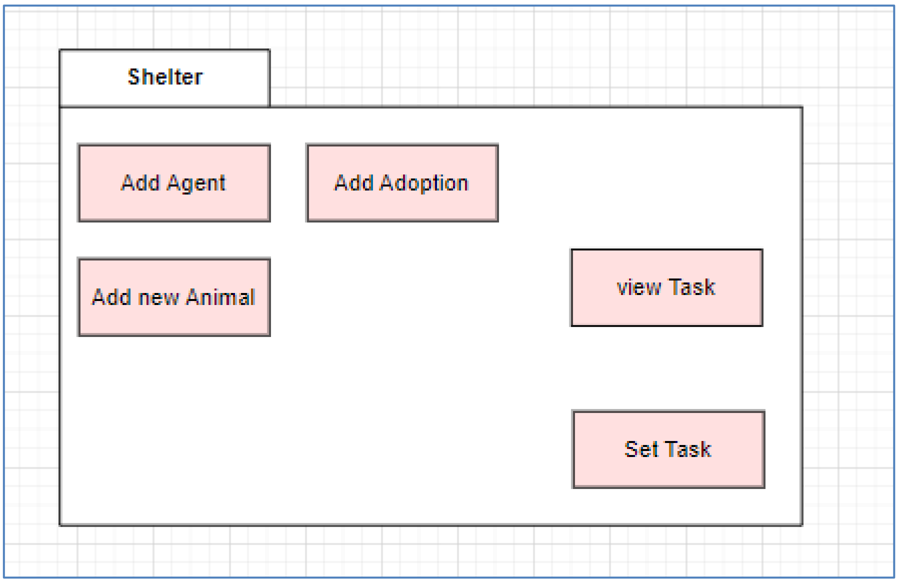


Figure B.5 Package diagram

2.3.1.2 Class Diagram

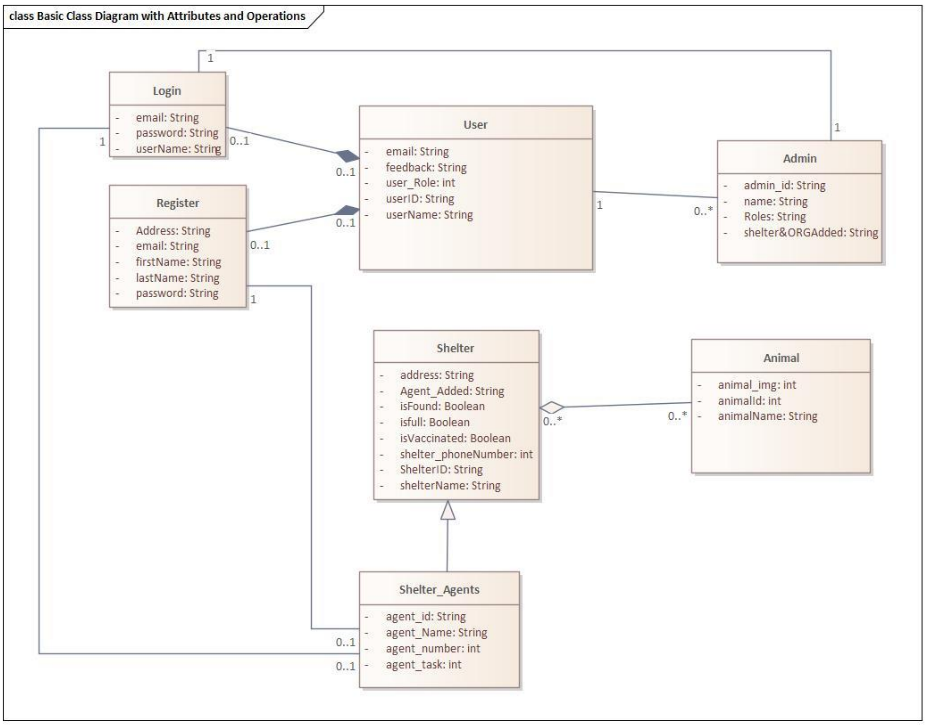


Figure B.6 Class diagram

2.3.1.3 Sequence Diagrams

1st Sequence Diagram- add animal for adoption this diagram is responsible for adding animal that Animal Organization or animal Shelters want to share for adoption by people that like animals.

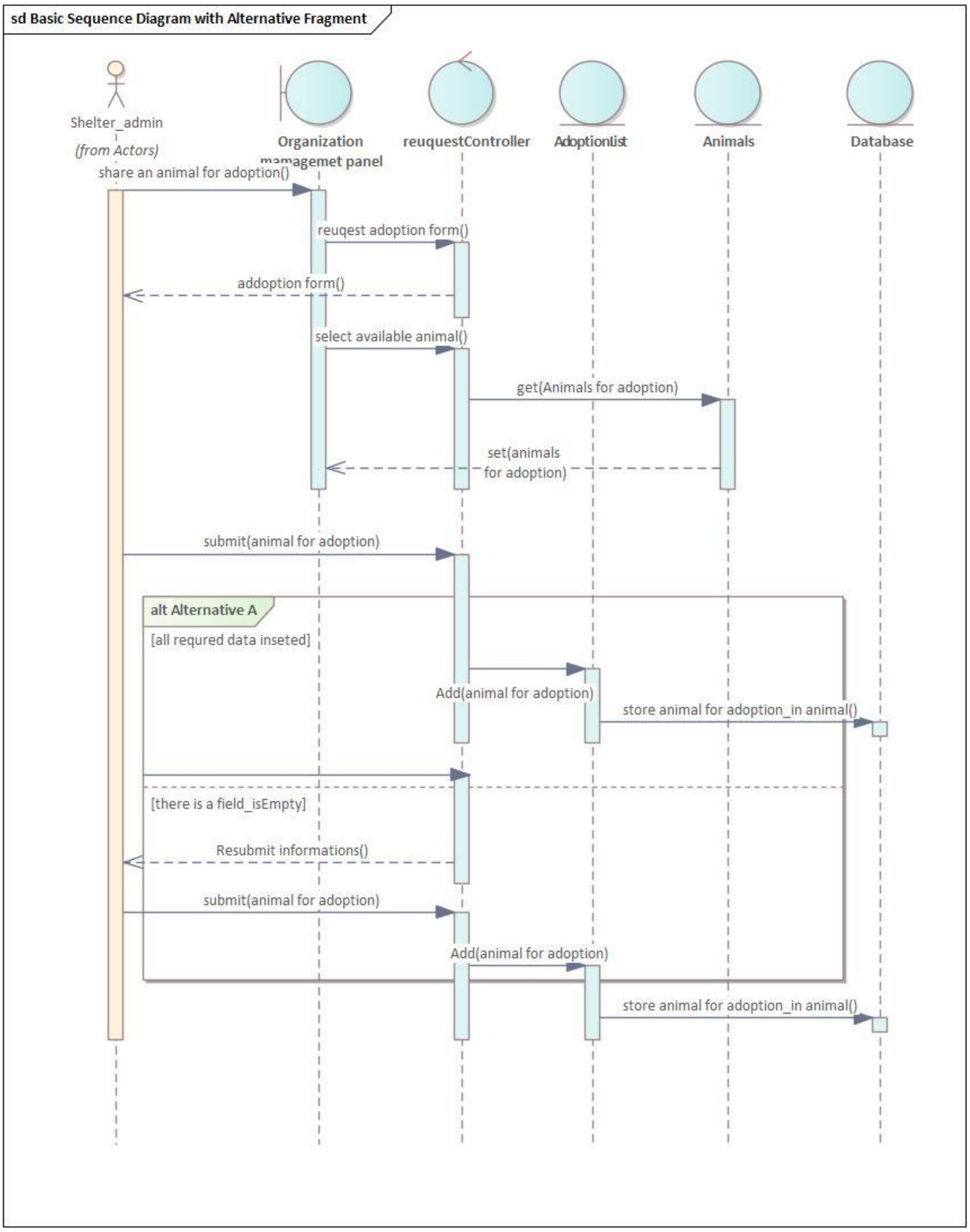

sd Basic Sequence Diagram with Alternative Fragment
Shelter_admin
(from Actors)
Organization mamagemet panel
reuquestController
Adoptionlist
Animals
Database
share an animal for adoption()
reuqest adoption form()
addoption form()
select available animal()
get(Animals for adoption)
set(animals for adoption)
submit(animal for adoption)
alt Alternative A
[all requred data inseted]
Add(animal for adoption)
store animal for adoption_in animal()
[there is a field_isEmpty]
Resubmit informations()
submit(animal for adoption)
Add(animal for adoption)
store animal for adoption_in animal()

Figure B.7 Sequence diagram add animal for adoption

2^{ed} Sequence Diagram marking stray animals this is the main process of the system, through this operation users can mark location of a stray animal on map and send it to organizations.

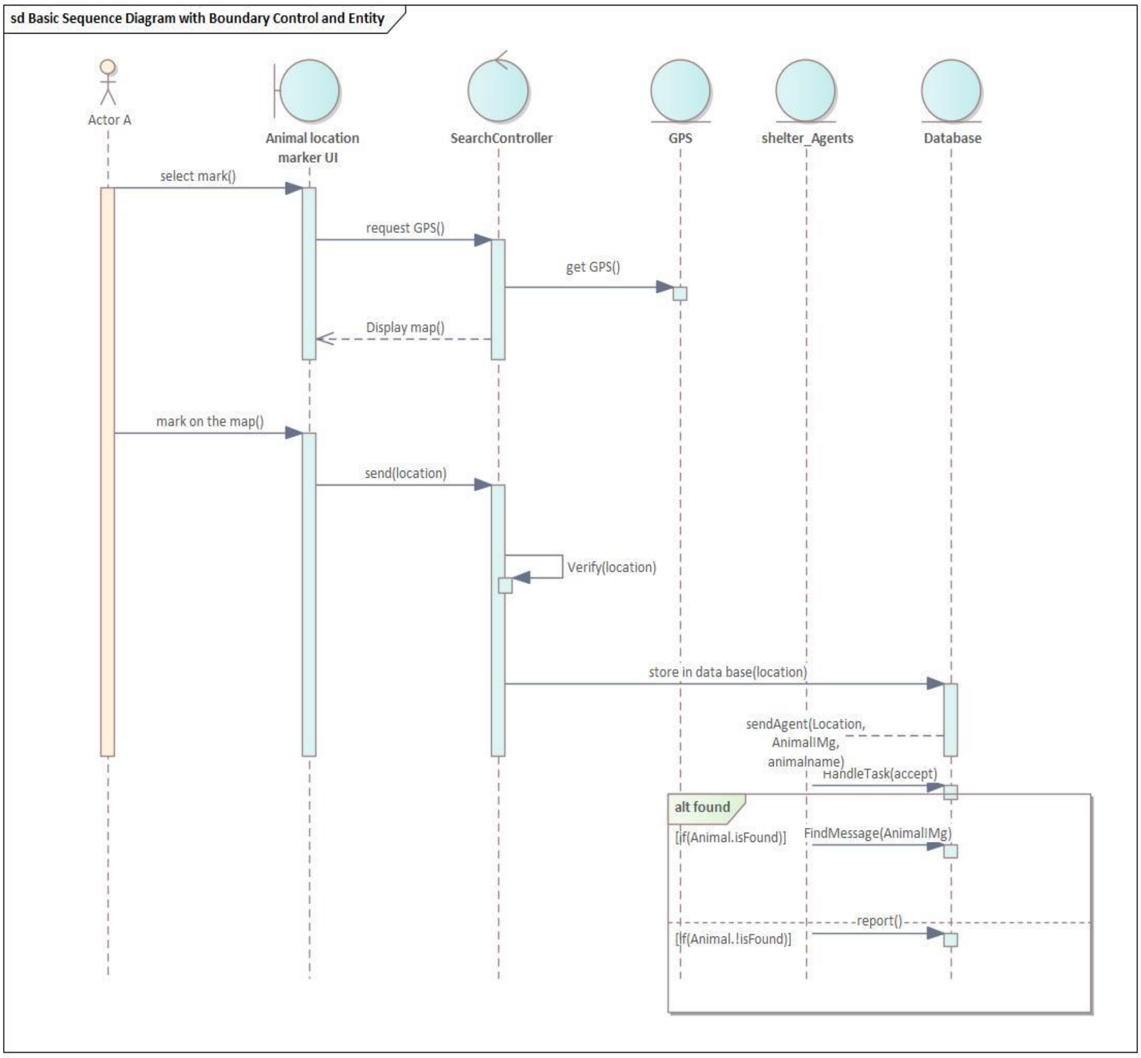


Figure B.8 Sequence diagram add animal for adoption

2.4 Data Design

Data Description:

Table B.1 Shelter data entity

Entity Name	**Shelter**

Method Name	**SelectAvailableAnimalForadoption()**
	get(Available_Animal_forAdoption)
	SubmitAnimalForAdoption()
	Set(animal for adoption)

Input	
Output	**String animalaName, animalID,….**
Algorithm	1. **Start** 2. **Request list available animals for adoption from data base …** 3. **database send the animal list to user** 4. **store selected animal for adoption in database 5. End**

Table B.2 Animal data entity

Entity Name	**Animal**
Method Name	**get(applicationID)**
Input	
Output	**String animalType, int animal id, animal name**
Algorithm	1. **Start** 2. **request for applicationID** 3. **system returns the value of variable ID 4. End**

Table B.3 Admin data entity

Entity Name	**Admin**

Method Name	**requestAccounts() getSelectedUsers()**
	AddRole()
	DisplayAccountInfo()

Input	**Search(AccountEmail)**
Output	
Algorithm	1. **Start** 2. **Request for all user accounts in database** 3. **Display all account info to the user** 4. **Get selected user account and redirect to admin** 5. **It will add the role after admin confirm 6. End**

Table B.4 user account data entity

Entity Name	**UserAccounts**

Method Name	**get(Accounts) SetA(Accounts)**
	Store(Roles)

Input	
Output	
Algorithm	1. **Start** 2. **Response for Admin request** 3. **Send all user accounts to admin 4. End**

Table B.5 Shelter agent data entity

Entity Name	**Shelter_Agent**

Method Name	**SendToAgent(Location,Animalimg,animalName)**
	handleTask(Accept)
	FindMessage(AnimalName)
	Report()

Input	
Output	**String Location, AnimalImg, animalName**
Algorithm	1. **Start** 2. **The agent receive location and detail about missing animal** 3. **The agent press yes for handling task** 4. **If the animal founded the agent warn owner, and shelter by find message.**
	5. End

Data Dictionary:

Table B.6 entities

Entity Name	**Description**
Shelter_Agent	**Include agent information with task handling for**
Admin	**Managing user account and setting roles**
User	**Contain user details and their roles**
Animal	**Contain animal information**
Shelter	**All info about shelters and events about animals and shelter agents**

References

[1]- A Complete Guide On Pet Apps Mobile App Development
https://www.webiotic.com/a-complete-guide-on-pet-apps-mobile-app-development/

[2]- Component Diagram tutorial
https://www.lucidchart.com/pages/uml-component-diagram

[3]- Architectural style vs Architectural patterns vs design
https://herbertograca.com/2017/07/28/architectural-styles-vs-architectural-patterns-vs-design-patterns/#:~:text=An%20Architectural%20Style%20is%20the,to%20solve%20a%20localised%20problem.

[4]- Too Many Animals, Too Few Good Homes
https://www.peta.org.uk/issues/animals-not-abuse/homelessness/

Appendix C

UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Software Testing Documentation

Project Title

Version 1.0

Printing Date

Department and Faculty

Prepared by:

Rebaz Awat Mustafa

REVISION PAGE

i. Overview

The content is about developing a proposed app for solving problem regarding stray animals. Current version 1.0

j. Target Audience

Shelters, organizations, people

k. Project Team Members

Shelters, organizations, people

l. Version Control History

Version	**Primary Author(s)**	**Description of Version**	**Date Completed**
<Current Version>	**Rebaz Awat**		

1. INTRODUCTION

1.1 Purpose

software requirements specification (SDD) document in this project describes describe architectural style and architectural design pattern, and design pattern. This document focused structure and behavior the product and its requirements to meet stakeholders need. Through this documentation we know that how to develop the software, and this is lead to make a better understanding about the software. Purpose of this Document is to show information about the product that is required before Construction phase like, getting information about goal of the product, audience, hardware requirements, software requirements, and functionalities of the product.

1.2 Scope

This project's scope is to overcome problems related to homeless animals mentioned above in detail like reducing stray animals inside cities, protecting them from diseases, and providing new home. And the solution proposed is to gather all homeless animals in shelters, then providing a home for them. The proposed application provides some features that should make an effective change at searching for stray animals much more accessible comparing to classic way through the app. To make the application be used by everyone anywhere, the application will be published as a mobile application. The methodologies for this project is agile Methodology, and the app will be done in 4 phases. My application won't include advertising, or any kind of social marketing.

1.3 Definitions, Acronyms and Abbreviation

SDT software Design Documentation

1.4 References

Udoagwu, K. (2022) ‘How to carry out a requirements analysis’, Wrike. Retrieved January 14, 2023, from https://www.wrike.com/blog/how-carry-out-requirements-analysis/#What-is-a-requirements-analysis

Ryan, R. (n.d.) ‘A complete guide on pet apps mobile app development’, Retrieved May 20, 2021, from https://www.webiotic.com/a-complete-guide-on-pet-apps-mobile-app-development

1.5 System Overview

Software design documentation is a detailed description of how a software system is planned, implemented, and organized. It serves as a blueprint for the development, maintenance, and evolution of a software system, and it is typically used by developers, project managers, and other stakeholders to understand the system's architecture, design decisions, and interfaces. This documentation includes things like class diagrams, flowcharts, pseudocode, and detailed explanations of the system's components and how they interact with one another. It is an important part of the software development process, as it helps ensure that the system is built in a clear and consistent manner, and that any changes or updates can be made in an organized and controlled way.

2. TEST CASES, DATA AND EXPECTED RESULTS

2.1 Test TC001 for Module <Add animal for adoption>: <Name of Use Case (UC001)>

Table C.1 UC001_01: e.g. Add animal (upload image)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_01_01	Image type	JPEG, JPG = ok	Added to animal card in adoption list	Pass
TC001_01_02	imageName	Maximum 200 character	Added to animal card in adoption list	Pass
TC001_01_03	Image size	Maximum 5mb	Added to animal card in adoption list	Pass
TC001_01_04	Image quality	Hd	Added to animal card in adoption list	pass

Table C.2 UC001_02: e.g. Add animal (name)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_02_01	“Bold”	ok	added	Pass
TC001_02_02	%%^	No symbols allowed	Rejected need to write again	Pass
TC001_02_03	10	Numbers alone not allowed	rejected	fail

Table C.3 UC001_03: e.g. Add animal (discerption)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC001_02_01	Text 200 word	Ok	added	Pass
TC001_02_02	شقاف	Not supported language	Rejected need to write again	Pass
TC001_02_03	300word text	No more than250 word allowed	rejected	Pass

2.2 Test TC002 for Module1: <Name of Use Case (Add Agent)>

Table C.4 UC002_02: e.g. Add role(user account)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC002_02_01	Set agent role	Dropdownlist not selected	error	Pass
TC002_02_02	Set agent role	ok	Rejected need to write again	Pass

2.3 Test TC003 for Module1: <Mark-stray Aimal (UC003)>

Table C.5 UC003_01: e.g. mark-stray animal (upload photo)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC003_01_01	Image type	JPEG, JPG = ok	Added to animal card in adoption list	Pass
TC003_01_02	imageName	Maximum 200 character	Added to animal card in adoption list	Pass
TC003_01_03	Image size	Maximum 5mb	Added to animal card in adoption list	Pass
TC003_01_04	Image quality	Hd	Added to animal card in adoption list	pass

Table C.6 UC003_01: e.g. mark-stray animal (send location)

Test Case ID	Input data	Expected result	Actual result	Pass / Fail
TC003_02_01	“address%sfeewaa23”	accepted	Location sent	Pass
TC003_02_02	“null address”	No places selected	Rejected need to mark again	Pass

Table C.7 Login Test case

Test Case Id	**Test Case Description**	**Test Steps**	**Expected Results**	**Actual Result**	**User Target**	**Test Input**	**Test Output**	**What Exist in test**
TC-1-1	Login with invalid credentials	1. add phone number 2. submit 3. verify OTP	The user should login using valid number	The user could not log in the application	Normal User	“07701311541”	‘login faild”	A user interface that show login component

Table C.8 Add for adoption

Test Case Id	**Test Case Description**	**Test Steps**	**Expected Results**	**Actual Result**	**User Target**	**Test Input**	**Test Output**	**What Exist in test**
TC-3-1	Add for adoption with no data	1. add title,dus,category, ,time. 2. Next img. 3. submit	Add for adoption by user	The user was submitted null	Normal User	No data	Fail no progress add data”	User can add adoption by Ui

Table C.9 Add Missing pet

Test Case Id	Test Case Description	Test Steps	Expected Results	Actual Result	User Target	Test Input	Test Output	What Exist in test
TC-4	Request help using add missing pet with valid data	1. add title,bounty,category, ,time. 2. Next img. 3. submit	Add for adopting by user	The user fill data: 'name" bool Type dog, Bounty 20$	Normal User	confirmed	Success the messing pet sent to shelter admin	User can add missing by Ui

Table C.10 Admin-shelter admin login

Test Case Id	Test Case Description	Test Steps	Expected Results	Actual Result	User Target	Test Input	Test Output	What Exist in test

TC-6	Login with valid credentials	1. add email 2. Add password 2. submit 3. login	The admin should login using valid email and pass	The user was logged in the web panel using email and password	Shelter admin	“example@gmail.com”	‘login successful”	Login Ui on web server

3. TEST APPROACH ANALYSIS

UC001: Add animal for adoption

animalimage

EP class 1 (valid): 1kb < animalimage< 5mb

EP class 2 (invalid): animalimage > 6mb

BVA values for animalimage: 1kb 5kb, 6mb

imageType

EP class 1 (valid): imageType =Jpeg

EP class 2 (valid): imageType =Jpg

EP class 3 (invalid): imageType =web

BVA values for imageType jpeg,jpg

Animal_Name

EP class 1 (valid): Animal_Name= “bold”

EP class 2 (invalid): Animal_Name =”23”

EP class 3 (invalid): Animal_Name =”%^$”

BVA values for Animal_Name bold, 23 %^$

Table C.11 APPENDIX C. TRACEABILITY MATRIX

Test Case ID	**Use Case ID/** **Sequence Diagram ID**	**Package ID**
TC001 for <Shelter> - TC001_01 - TC001_02	UC001 - SD001 - SD002	P001
TC002 for <shelter> - TC002_01 - TC002_02	UC002 - SD004 - SD005	P001
…		
TC003 for <User> - TC003_01 - TC003_02	UC003 - SD006 - SD007	P002
…		

QAIWAN INTERNATIONAL UNIVERSITY	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 Rebaz Awat Mustafa (name) hereby submit the final copies of my thesis for the degree of ☑ Bachelor / □ Master / □ Doctor of Philosophy

The title of the thesis is:

Rescu and Adoption of Homless Animals

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: Rebaz Awat

Department: Software engineering

Signature:

Date: 11/8/2024

ENDORSEMENT BY MAIN SUPERVISOR

Section 2 (To be completed by Main Supervisor)

Supervisor's Name: ... Mr. AKo ...

Department: ... S. E ...

I have examined the thesis of Mr. /Mrs. /Ms.: ... rebaz A wat ...

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

Dr.shihab 11/8/2024

(Main Supervisor's Signature & Stamp) (Date)

ENDORSEMENT BY Librarian

Section 3 (To be completed by Librarian)

Librarian's Name: D/kchwaz...othman ....

☑ Approved □ Not Approved

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

.... 11/8/2024

(Librarian's Signature & Stamp) (Date)

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: Rebaz Awat Mustafa

Signature of Student: ....

IC. No. / Passport No.: ....

Date: 11/8/2024