|  |  |
| --- | --- |
| PET STATE DOCUMENTATION | Abstract  This document contains the system proposal ,System Requirements Specification ,Software Detailed Design ,Test Plan and the User Manual for the Pet State android application.  Group Three  **COMP 390 GROUP PROJECT** |

**Group Three**

**S13/09396/17 Hilda Chari Mwashumbe**

**S13/03140/18 Joy Mary Jepkosgei Kirongo**

**S13/03134/18 Anthony Mumo Mutua**

**S13/03129/18 Alvin Alwena Masambu**

**S13/03122/18 Duncan Ndegwa**

**S13/03130/18 Ian Ndathi Mburu**

Table of Contents

[PROPOSAL 3](#_Toc100416422)

[Introduction 3](#_Toc100416423)

[Objectives: 3](#_Toc100416424)

[Problem statement: 4](#_Toc100416425)

[Importance of the system: 4](#_Toc100416426)

[Expected results: 4](#_Toc100416427)

[Challenges: 5](#_Toc100416428)

[Work Schedule 5](#_Toc100416429)

[Future improvements: 5](#_Toc100416430)

[SOFTWARE REQUIREMENTS SPECIFICATION 7](#_Toc100416431)

Introduction…………………………………………………………………………………………………………………………………..9

Purpose………………………………………………………………………………………………………………………………………9

Intended Audience……………………………………………………………………………………………………………………10

Scope………………………………………………………………………..………………………………………………………………10

Overall Description………………………………………………………………………..…………………………………………10

Assumptions and Constraints…………………………………………………………………………………………………..10

System Features ………………………………………………………………………………………………………………………….10

User requirements………………………………………………………………………..…………………………………………11

System requirements………………………………………………………………………..…………………………………….11

Interface Requirements………………………………………………………………………..…………………………………11

Functional requirements………………………………………………………………………..………………………………..11

Non -Functional requirements………………………………………………………………………..……………………….12

UML Diagrams

Use Case diagram………………………………………………………………………..………………………………………… 13

Use Case Description………………………………………………………………………..……………………………………13

Activity Diagram………………………………………………………………………..……………………………………..……23

SOFTWARE DETAIL DESIGN……………………………………………………………………………………………..…………….25

System Architecture…………………………………………………………………………………………………….…………….25

Data Flow Diagram……………………………………………………………………………………………………………….26

Package Diagram…………………………………………………………………………………………………………………..27

Database Design……………………………………………………………………………………………………………………28

Human Interface Design………………………………………………………………………….………………………………28

Detailed Design…………………………………………………………………………………………………………………………...31

Software Detailed Design………………………………………………………………………………………………….…….31

Security Detailed Design………………………………………………………………………………………………………………31

TEST PLAN………………………………………………………………………………………………………………………………………36

Test Cases……………………………………………………………………………………………………………………………………44

USER MANUAL………………………………………………………………………………………………………….…………………..48

# SYSTEM PROPOSAL

Introduction**:**

Pet keeping can be described as symbiotic relationship since they are kept for pleasure that they can give its owner and this pleasure appears to be mutual. Keeping pets has been practiced from prehistoric times to present and as pets are found in early every culture and society, pet keeping apparently satisfies a deep universal human need.

In a study published on 7th Feb 2019 by Emmah N. Kwebu,460 of 1213 of the study households owned dogs and 526 of those without dogs had previously owned dogs. Since many households own pets, our projects aims to assist pet owners in providing quality and effective care to the pets.

## Objectives:

1. To provide pet owners with relevant information to give quality care to their pets.
2. To provide a link between veterinary doctors and pet owners on the platform.
3. To generate income by deducting ten percent of the consultation fee charged.
4. To provide 24hr day service.
5. To save on time since the pet owner does not need to visit the veterinary doctor physically every time the pet is sick.

## Problem statement:

In recent years there has been an increase in pet keeping, many of the pet owners have no professional information on how to provide quality care to the pets and so they are lack easy access to proper medical attention.

Also pet owners have had the burden of always needing to physically contact the veterinary for diagnosis and treatment of their pets.

## Importance of the system:

1. The system will provide necessary information on giving quality care for pets e.g. immunization, nutrition and treatment.
2. The system will provide a link between pet owners and vet doctors which will ensure professional medical attendance to the pets.
3. The system will ensure all time access to pet info on demand.
4. The system will reduce the costs incurred by the pet owners to travel for physical consultations.
5. The system will give the vet doctor the power to diagnose and treat pets remotely.

## Expected results:

1. Reduced mortality rates in common household pets.
2. Better relationship between owner and their pets since they can take care of their pets better.
3. Effective health monitoring.

## Challenges:

1. Competitions from existing methods of pet care e.g. Other online platforms.
2. Payment methods require a subscription to be used in the application.

Work Schedule**:**

This project is estimated to take a duration of 10 weeks.

## Future improvements:

1. Incorporate Artificial Intelligence to provide enhanced diagnostics and possible vulnerabilities.
2. Include other payment methods.

# **SOFTWARE REQUIREMENT SPECIFICATION**

## 

## Introduction

Most local veterinarians rely on hard copy methods of data collection, management and storage in terms of their client’s medical reports. These methods are inefficient at best. “Pet State” seeks to provide a solution to this by automating these tasks through our mobile application that functions as an interface for the collection and management of pet medical data.

Most owners lack enough information on how to care for their pets. “Pet state” seeks to provide an ever present guide to the pet owners on how to do this.

This document details the plan for the development of “Pet State.” It is intended for developers, designers, and testers working on “Pet State” as well as potential project investors. It also provides access to a vet incase of a sick pet.

This document will include a summary of:

* how the system will function
* the scope of the project from the development viewpoint
* the technology used to develop the project, and
* the Overall Description.
  1. **Purpose**

The purpose of the system is to provide an interface for easy collection, storage and retrieval of the medical information of pets .

To enable pet owners to access pet care information and services remotely.

To provide access to quality pet care information and a link to vets for the pet owners.

* 1. **Intended audience**

The system is for use by veterinary doctors and pet owners. It is useful for pet monitoring as well as keeping track of their own clients.

It is also used by pet owners to gain easy access to vets and quality health care for their pets.

* 1. **Scope**

For the system’s optimal performance, data collection ranges from pet details to individual pet traits including details of the owners of the pets. This data is stored and can be viewed by the vet and pet owners on demand.

* 1. **Overall description**

The primary goal of the project is to construct an alpha-quality version of an application that can be transitioned for future development. The aim is to create an application that is indispensable for both vets and pet owners.

* 1. **Assumptions and constraints**

The end of the semester imposes a hard deadline for completing the project because of this emphasis will be placed on constructing a system that includes a large, but not necessarily fully detailed feature set (breadth instead of depth).

Additionally our staffing is not negotiable, limiting the flexibility of the team skill set.

It is assumed that if more than half of the project is completed by half the total time the project will be successful before the deadline

## System Features

## User requirements

The application should provide an interface for the vet to add clients, manage existing clients and their pets. It should store and provide access to the pet’s medical records. It should also allow the pet owner to add their pet, view information for quality pet care and fill a form on the symptoms of an ill pet and send it to vet.

## System requirements

Android 4.4

80MB RAM

16GB STORAGE

## Interface requirements

Database should interact with the android system to view the individual pet history, registered clients and pets.

* 1. **Functional requirements.**

The system shall allow a user(whether vet or pet owner) to create an account on the platform.

It shall allow user authentication for security purposes for both the vet and pet owner.

It shall allow the pet owner to view, add and modify their pet details

It shall allow the vet to add new clients.

It shall allow the vet to view the client’s details, who they are and where they are from.

It shall allow the vet to view pet details, that is, age, type, breed and the medical records of the pet’s previous visits.

It shall allow the vet to create a new record of the current visit.

It shall allow the user to change or reset password.

It shall allow the pet owner to share records with other vets.

* 1. **Non-functional requirements**

The application should be easy to learn.

The application shall be available to the vet 24 hours.

The application shall provide security features to ensure privacy of the client and pet data.

It shall have a quick response time.

1. **UML Diagrams**
   1. **Use-Case Diagram**

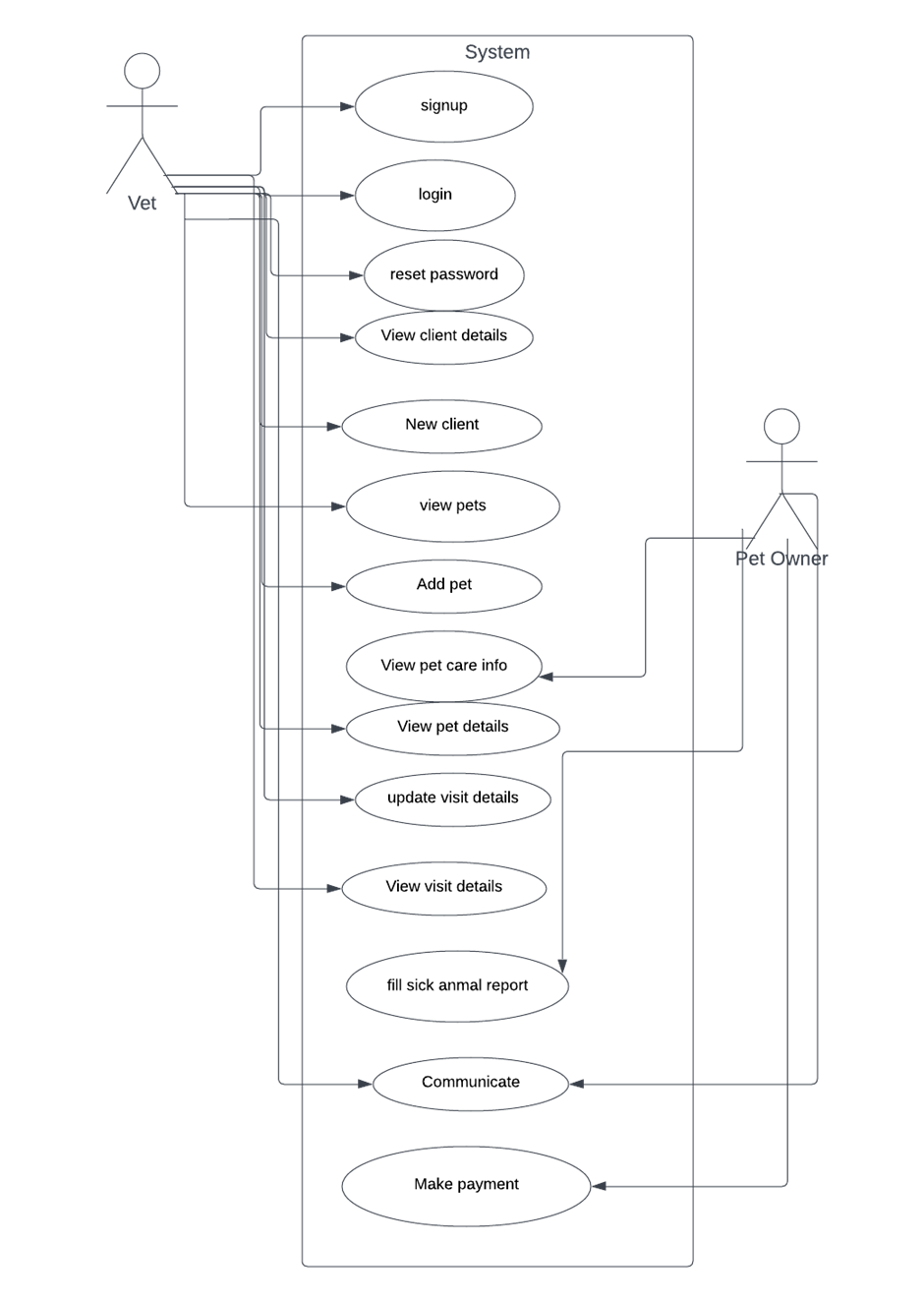


Figure 1(Use Case Diagram for "Pet State")

* 1. **Use-Case Descriptions**

|  |
| --- |
| Sign In **ID:** UC-1 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet gains access into the system by authenticating themselves |
| **Trigger:** Opening the mobile application |
| **Type:** External |
| **Pre-Conditions:**   1. The vet has a valid account 2. The vet has installed the application in their smart phone. |
| **Normal Course:**   1. The vet enters their registered email address 2. The vet enters their associated password 3. The vet clicks on the sign in button |
| **Alternative Course:**   1. The vet enters their registered email address 2. The vet enters their associated password 3. The vet clicks on the sign in button 4. The system invalidates the credentials (email or password) 5. The system gives the user two more chances to authenticate 6. If the credentials are invalid after the third trial, the system bans the user from accessing the system |
| **Post-Conditions:**   1. The vet gains access into the system and is able to view clients and their details |

|  |
| --- |
| Sign Up **ID:** UC-2 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes a vet’s initial registration into the system |
| **Trigger:** Clicks the sign up button |
| **Type:** External |
| **Pre-Conditions:**   1. The app is installed into a mobile device 2. The user has access to the internet |
| **Normal Course:**   1. The vet clicks on the sign up button 2. The vet enters the name, contact details, physical address, creates a password in respective form textboxes 3. The vet clicks on the sign up button 4. The system sends a verification code to the registered email address 5. The vet enters the code 6. The system verifies the vet and issues an account |
| **Alternative Course:**   1. The vet clicks on the sign up button 2. The vet enters the name, contact details, physical address, creates a password in respective form textboxes 3. The vet clicks on the sign up button 4. On discovering empty textboxes, the system alerts the user to fill in the required fields 5. The system sends a verification code to the registered email address 6. The vet enters the code 7. The system verifies the vet and issues an account |
| **Post-Conditions**   1. The user acquires a valid account and is able to log into the system application |

|  |
| --- |
| Reset Password **ID:** UC-3 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet can change their password in the case that they forgot their initial password |
| **Trigger:** The vet clicks on the forgot password button |
| **Type:** External |
| **Pre-Conditions:**   1. The vet has a valid account |
| **Normal Course:**   1. The vet clicks on the Forgot Password button 2. The vet provides their registered email address into a textbox 3. The vet clicks on the Submit button 4. The system sends a verification code to the provided email address 5. The vet provides the code into textboxes 6. The app validates the provided code and activates password change textboxes 7. The vet enters a new password into a textbox and re-enters the password into a confirm password textbox 8. The vet clicks on the submit button 9. The system confirms that the password has been changed successfully and opens the sign in page |
| **Alternative Course:**   1. The vet clicks on the Forgot Password button 2. The vet provides their registered email address into a textbox 3. The vet clicks on the Submit button 4. The system sends a verification code to the provided email address 5. The vet provides the code into textboxes 6. The code fails to verify 7. The system asks the user to re-enter the verification code OR to click on the Resend button 8. The app validates the provided code and activates password change textboxes 9. The vet enters a new password into a textbox and re-enters the password into a confirm password textbox 10. The vet clicks on the submit button 11. The system confirms that the password has been changed successfully and opens the sign in page   **Alternative Course 2:**   1. The vet clicks on the Forgot Password button 2. The vet provides their registered email address into a textbox 3. The vet clicks on the Submit button 4. The system sends a verification code to the provided email address 5. The vet provides the code into textboxes 6. The app validates the provided code and activates password change textboxes 7. The vet enters a new password into a textbox and re-enters the password into a confirm password textbox 8. The passwords on the two textboxes do not match 9. The system asks the user to re-enter the passwords 10. The vet clicks on the submit button 11. The system confirms that the password has been changed successfully and opens the sign in page |
| **Post-Conditions**   1. The vet signs into the application 2. The password is updated |

|  |
| --- |
| New Client **ID:** UC-4 **Priority:** High |
|  |
| **Actor: Vet** |
| **Description:** This use-case describes how a vet can enter a new client into the application system |
| **Trigger:** Clicking on the New Client button |
| **Type:** External |
| **Pre-Conditions:**   1. The vet successfully logs into the system 2. The client does not exist in the system |
| **Normal Course:**   1. The vet clicks on the New Client button 2. The system provides a page with textboxes for filling client details 3. The vet uploads the client’s image in the avatar field 4. The vet enters the customer’s Full name, Email address, Phone number, and Physical address in the respective field textboxes 5. The vet clicks on the Save button |
| **Alternative Course:**   1. The vet clicks on the New Client button 2. The system provides a page with textboxes for filling client details 3. The vet uploads the client’s image in the avatar field 4. The vet enters the customer’s Full name, Email address, Phone number, and Physical address in the respective field textboxes 5. The vet clicks on the Save button 6. One or more required textboxes are unfilled, the system alerts the vet to fill in the empty textboxes 7. The vet clicks on the Save button |
| **Post-Conditions**   1. The client details are stored into the system |

|  |
| --- |
| View Clients & Details **ID:** UC-5 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet can view a list of clients and their details as registered in the application system |
| **Trigger:** Successful log in to the system |
| **Type:** External |
| **Pre-Conditions:**   1. The vet has a valid account |
| **Normal Course:**   1. The vet successfully logs into the system 2. The system presents the homepage |
| **Alternative Course:**   1. There are no registered clients in the system 2. The vet adds new clients into the system |
| **Post-Conditions**   1. The vet can access client details |

|  |
| --- |
| View Pets **ID:** UC-6 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet can view a list of registered pets owned by a particular client. |
| **Trigger:** The vet clicks on a particular client. |
| **Type:** External |
| **Pre-Conditions:**   1. The client whose pets are being viewed exists, that is, is registered. |
| **Normal Course:**   1. The vet clicks on a client. 2. The list of pets owned by that client is displayed. |
| **Alternative Course:**   1. If no registered pets exist, the vet adds a new pet. |
| **Post-Conditions:**  None |

|  |
| --- |
| Add Pet **ID:** UC-7 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet registers a pet to a registered client |
| **Trigger:** A click on the Add Pet button |
| **Type:** External |
| **Pre-Conditions:**   1. The client is registered |
| **Normal Course:**   1. The vet clicks on the Add Pet button 2. The vet uploads an image of the pet in the avatar field 3. The vet enters the Name, Kind, Breed, Color, Year of Birth in the respective textbox fields 4. The vet clicks on the Save button |
| **Alternative Course:**   1. The vet clicks on the Add Pet button 2. The vet uploads an image of the pet in the avatar field 3. The vet enters the Name, Kind, Breed, Color, Year of Birth in the respective textbox fields 4. The vet clicks on the Save button 5. One or more required fields are unfilled and the system alerts the vet to fill in the required fields 6. The vet clicks on the Save button |
| **Post-Conditions**   1. The pet is registered to their respective owner(client) |

|  |
| --- |
| View Pet Details **ID:** UC-8 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet can view basic details about a pet and its visit details |
| **Trigger:** Click on a particular pet |
| **Type:** Externa**l** |
| **Pre-Conditions:** The pet is registered |
| **Normal Course:**   1. The vet clicks on the target pet 2. The system provides a page with the pet’s details and visit history |
| **Alternative Course:**   1. None |
| **Post-Conditions**   1. The vet is able to view visit details 2. The vet is able to add visit details |

|  |
| --- |
| Update Visit Details **ID:** UC-9 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet modifies previously recorded visit details. |
| **Trigger:** A click on the update button on the visit details page. |
| **Type:** External |
| **Pre-Conditions:**   1. The visit details for already exists for them to be updated. |
| **Normal Course:**   1. The vet clicks on the update button. 2. A page with the previously inserted data is shown. 3. The vet is able to make changes to the data. 4. The vet clicks on the save button. 5. The information is saved with a footnote of the date it was updated. |
| **Alternative Course:**   1. The vet clicks on the update button. 2. A page with the previously inserted data is shown. 3. The vet is able to make changes to the data. 4. The vet clicks the cancel button. 5. The updates are not saved and the details remain as they were. |
| **Post-Conditions:**   1. The changes are saved and information stored is updated. |

|  |
| --- |
| Enter Visit Details **ID:** UC-10 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet can insert the diagnosis and prescription of a pet for a particular visit. |
| **Trigger:** Clicking on the add visit button for the specific pet. |
| **Type:** External |
| **Pre-Conditions:**   1. The pet must have been previously registered and it’s details exists. |
| **Normal Course:**   1. The vet clicks on the add visit button. 2. The system provides a page with basic pet details such as the breed and pet name and the date pre-filled. 3. The vet enters the temperature, weight, diagnosis and prescription in their respective textboxes. 4. The vet clicks on the save button to save the information. |
| **Alternative Course:**   1. The vet clicks on the add visit button. 2. The system provides a page with basic pet details such as the breed and pet name and the date pre-filled. 3. The vet enters the temperature, weight, diagnosis and prescription in their respective textboxes. 4. The vet clicks on the save button to save the information. 5. One or more of the textboxes are left unfilled, the system alerts the vet to fill in the empty textboxes. 6. The vet fills each textbox. 7. The vet clicks on the save button to save the information. |
| **Post-Conditions:**   1. The visit details are saved. 2. A new entry for past visits is added. |
| View Visit Details **ID:** UC-11 **Priority:** High |
|  |
| **Actor:** Vet |
| **Description:** This use-case describes how a vet can access details of a pet’s past visits. |
| **Trigger:** Clicking a particular visit from the list of visits made by a pet. |
| **Type:** External |
| **Pre-Conditions:**   1. The pet must have been previously registered. |
| **Normal Course:**   1. The vet clicks on the target pet. 2. The vet clicks on the specific date they want to view. 3. The details of the visit that had previously been recorded are displayed. |
| **Alternative Course:**   1. If no visit details exists, a new visit is added instead. |
| **Post-Conditions:**  None |

|  |
| --- |
| View Pet Care Info **ID:** UC-11 **Priority:** High |
|  |
| **Actor:** Pet Owner |
| **Description:** This use-case describes how a pet owner can view vaccination and nutritional information on pets |
| **Trigger:** Click on the left hand side floating action button |
| **Type:** External |
| **Pre-Conditions:**   1. The pet owner is logged in |
| **Normal Course:**   1. The pet owner clicks on the floating action button 2. The system presents the information on the screen |
| **Post-Conditions**   1. The pet owner can access pet care information |

|  |
| --- |
| Fill sick animal report **ID:** UC-12 **Priority:** High |
|  |
| **Actor:** Pet Owner |
| **Description:** This use-case describes how a pet owner can record the symptoms of their sick animal |
| **Trigger:** Clicking on the add report floating action button |
| **Type:** External |
| **Pre-Conditions:**   1. The pet owner is logged in |
| **Normal Course:**   1. The pet owner clicks on the add report floating action button 2. The system presents the blank report 3. The pet owner fills in the details i.e. the symptoms the pet is experiencing on the respective textboxes 4. The pet owner clicks save and the details are saved |
| **Post-Conditions**   1. The system now has a new record of a sick animal report |

|  |
| --- |
| Communicate **ID:** UC-13 **Priority:** High |
|  |
| **Actor:** Vet, Pet Owner |
| **Description:** This use-case describes how a vet and a pet owner can open the chat and communicate with each other through the system |
| **Trigger:** Clicking on the chat floating action button |
| **Type:** External |
| **Pre-Conditions:**   1. The pet owner has paid for the service |
| **Normal Course:**   1. The pet owner makes the payment and is successfully directed to the chat while the vet clicks on the chat floating action button 2. Both actors are able to send and receive messages to and from each other in real time |
| **Alternative Course:**   1. The pet owner does not pay for the service 2. The pet owner is not allowed into the chat |
| **Post-Conditions**   1. The pet owner and vet are able to communicate |

|  |
| --- |
| Make Payment **ID:** UC-13 **Priority:** High |
|  |
| **Actor:** Pet Owner |
| **Description:** This use-case describes how a pet owner can pay for services in the system |
| **Trigger:** Clicking on the chat floating action button |
| **Type:** External |
| **Pre-Conditions:**   1. The pet owner is logged in |
| **Normal Course:**   1. The pet owner clicks on the chat floating action button 2. The system presents a prompt for them to enter their phone number 3. The pet owner receives an STK Push and makes the payment 4. The payment is validated |
| **Post-Conditions**   1. The pet owner is allowed to join the chat with the vet |

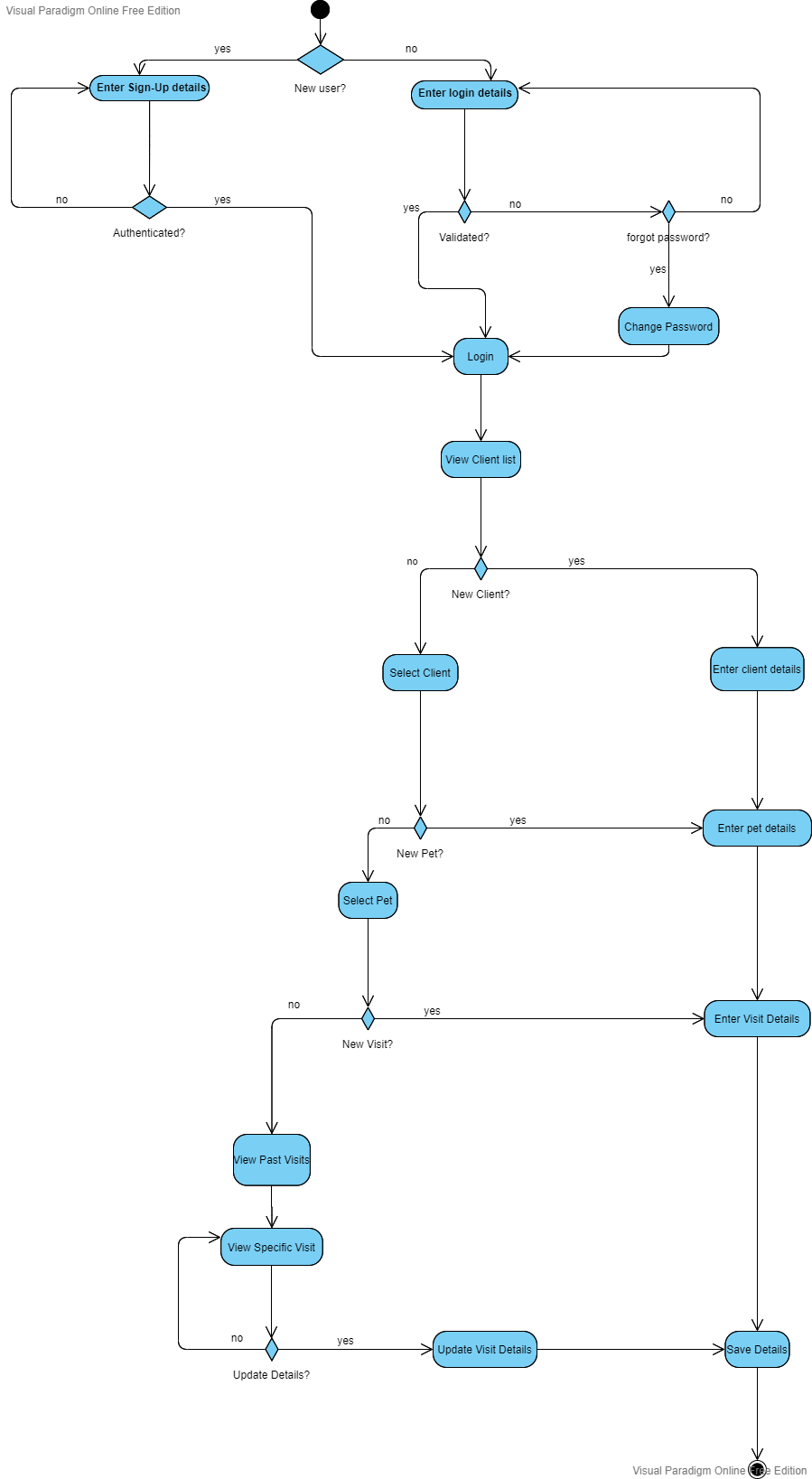
* 1. **Activity Diagram**

Figure 2(Activity Diagram for "Pet State")

# **SOFTWARE DESIGN DOCUMENT**

## Introduction

Many vets have difficulty accessing some of their repetitive tasks, particularly recording data, because they must rely on hard copies. A system which is often fragmented, unreliable and inefficiently operated. Lack of proper framework is the leading obstacle to meeting mobility needs of the people who need the services the most. The purpose of this deployment is to replicate and advance the success by providing an efficient filing system to leverage other activities inclusive of decision making. Goals are to use service technology integration to:

* Increase mobility and accessibility.
* Achieve more efficient use of service, that is do more with less.

**Constraints**

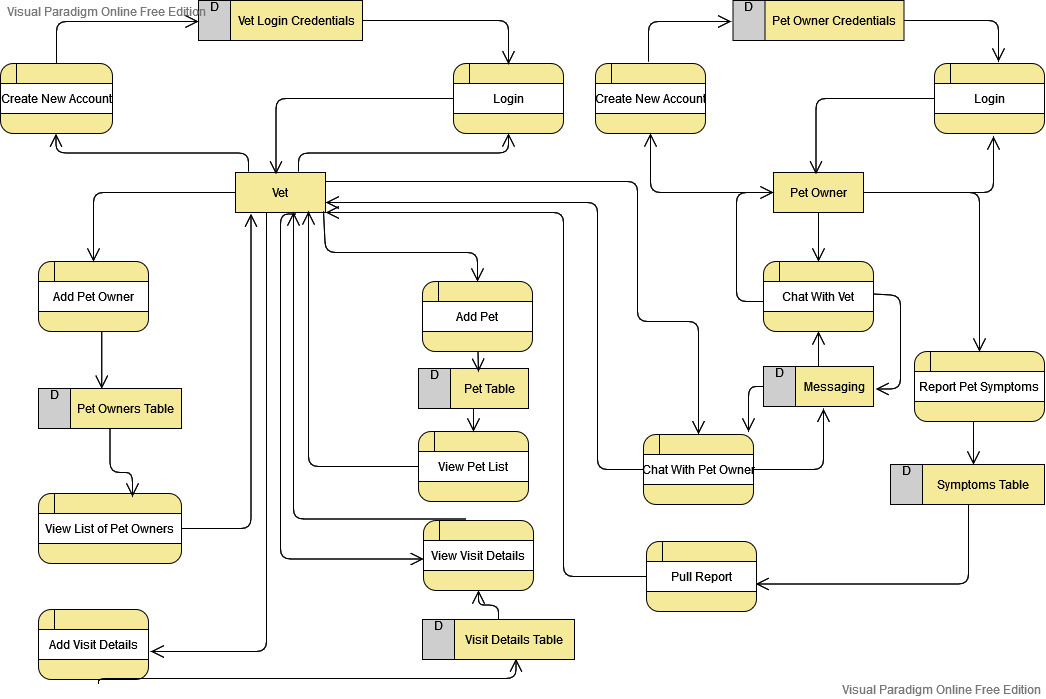
Timeline-the fixed timeline of ten weeks could lead to burn out and hamper with creativity, this will have an effect on the design process.

Device specific-The pet state application will only be available to devices running on android operating system with at least 80MB RAM and 16GB storage.

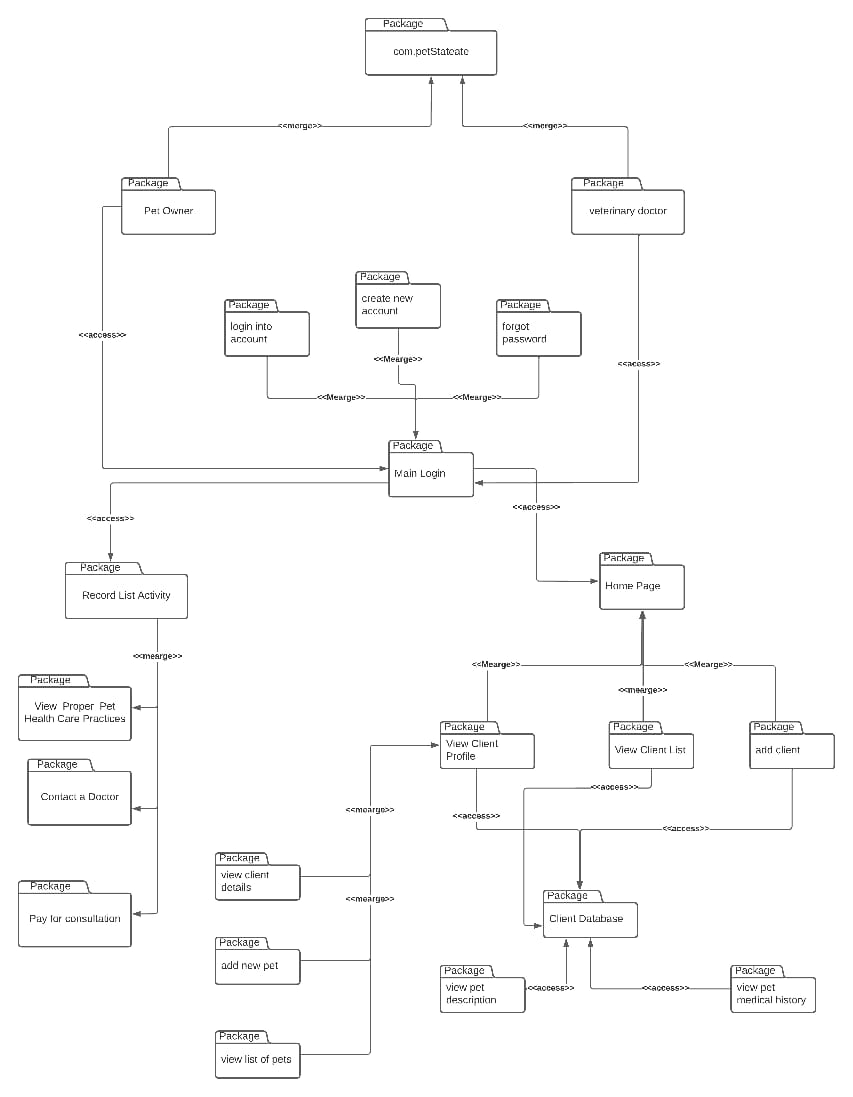
Cost-during design some of the design tools require premium membership so as to be applied on the project. Premium membership incurs additional cost.

**Architecture of the system**

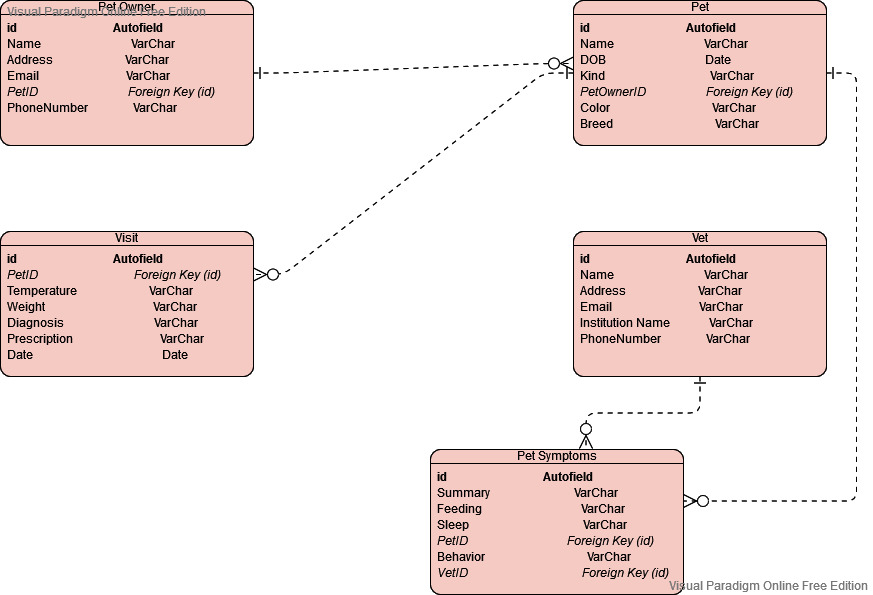
Data Flow Diagram.



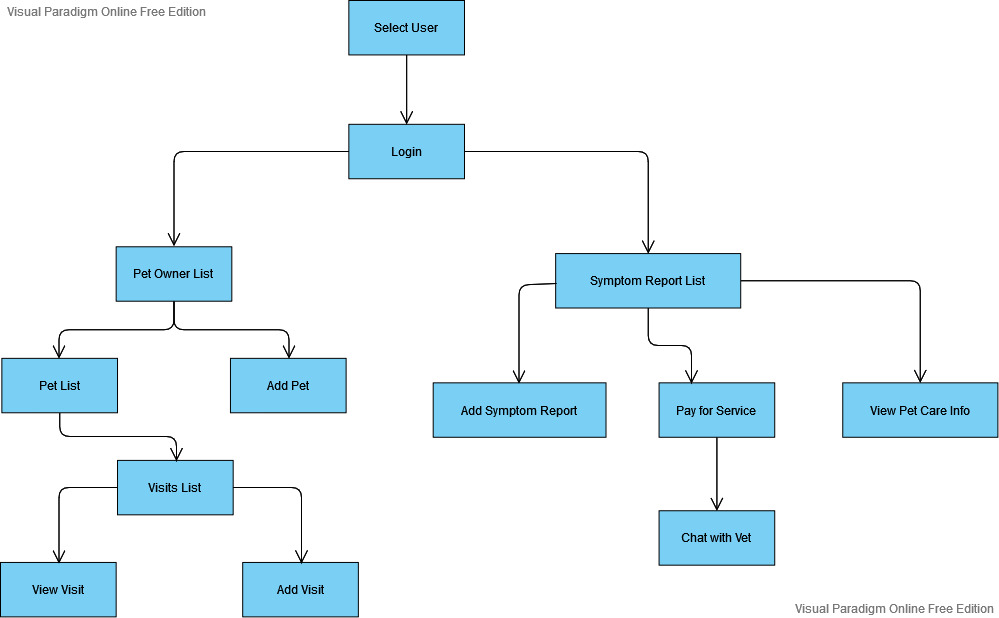
PACKAGE DIAGRAM



Database Design

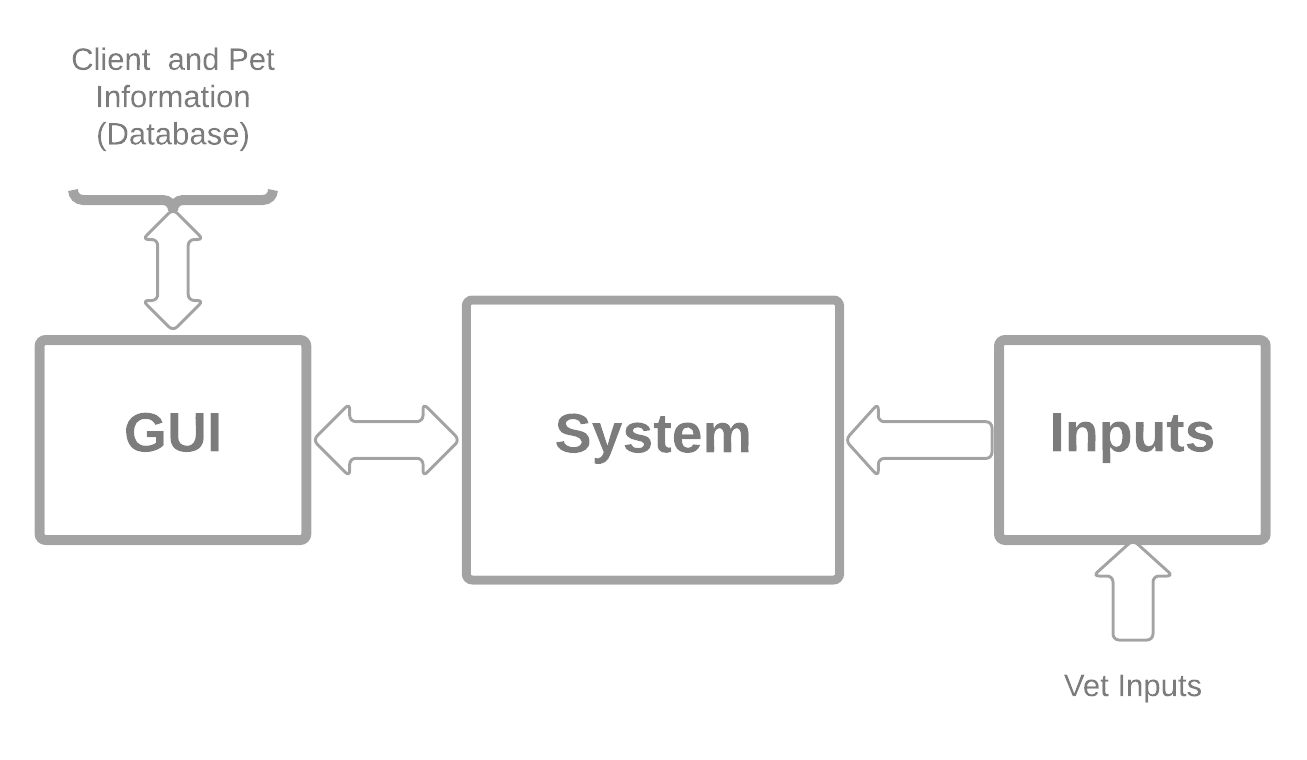


Human machine interface



**Human Machine Interface**

1. **Inputs** – The system accepts data inputs entered by the Vet. The inputs include:
2. Client Information – an avatar to identify the client, full name of the client, email address and phone number and their physical address
3. Pet Information – an avatar to identify the pet, the name of the pet, breed, clolor, kind of the pet and its date of birth. The system also accepts visit information about a pet, which includes, the measured weight of the pet during visitation, its temperature at the time of diagnosis as well as diagnosis details and prescription.
4. **Outputs** – the system generates reports on the pet information stored in the system database. The reports are generated on the last day of each month. The reports include:
5. List of pets visiting each month
6. Number of pets from a specific geographical region
7. Graphs comparing the kinds of pets seen by the vet in a period of one month



**Detailed design**

Kotlin is the primary language that will be used in designing since:

1. It is a modern statically typed programming language used by android developers that helps boost productivity, developer satisfaction and code safety.
2. The IDE (android studio) fully supports Kotlin
3. Kotlin is a cross-platform, general-purpose programming language with type inference. It is designed to interoperate fully with Java, and the JVM version of Kotlin's standard library depends on the Java Class Library, but type inference allows its syntax to be more concise.

Minimum API level 19 (kitkat 4.4) and above allows a 99.7% audience. The target is API level 29 along with build tools, revision 28.0.3.

The build tool component is updated by Android Gradle Plugin which is integrated into the IDE.

Gradle is the main building tool that compiles the code since it automatically resolves external dependencies.

**Software Detailed Design**

Among the exiting login providers we will be using Google’s firebase which is a cloud platform.

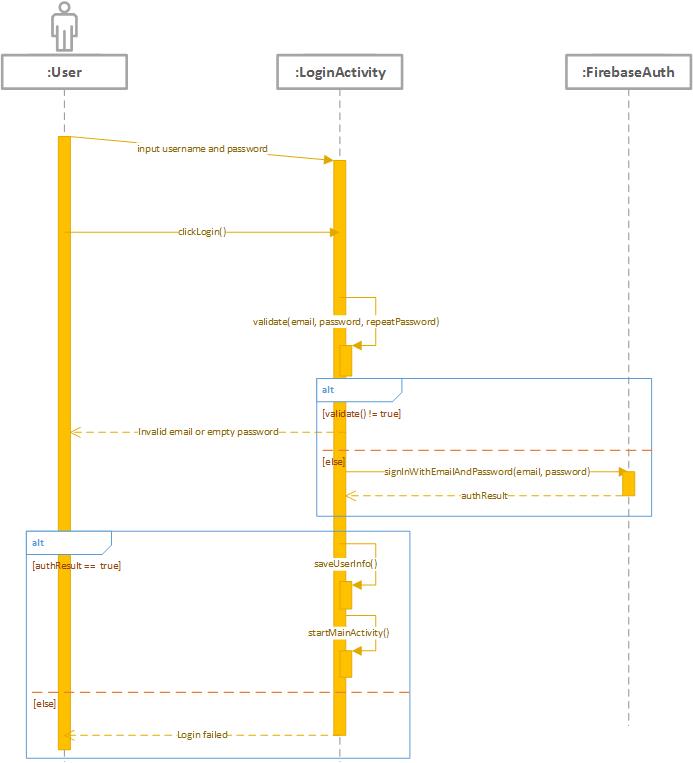
Reasons as to why we will be using firebase include:

1. It is easily integrated with android, IOS, unity and c++
2. It is easy to monitor performance and stability
3. Has a fully managed backend infrastructure

**Application activities**

1. Login activity
2. Sign up activity
3. Main activity
4. Fragment home
5. Fragment analysis
6. Fragment profile
7. Fragment database
8. Settings activity
9. **Login activity**

The user logs in after the credentials are authenticated. Authentication data is handled by firebase backend infrastructure.



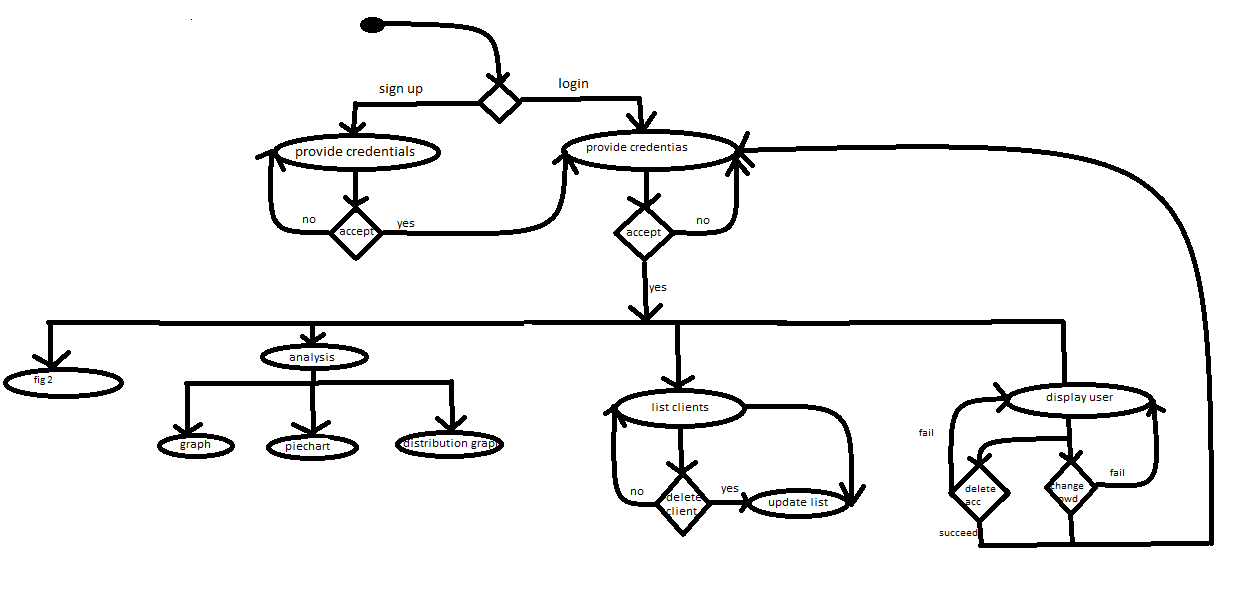
1. **Signup activity**

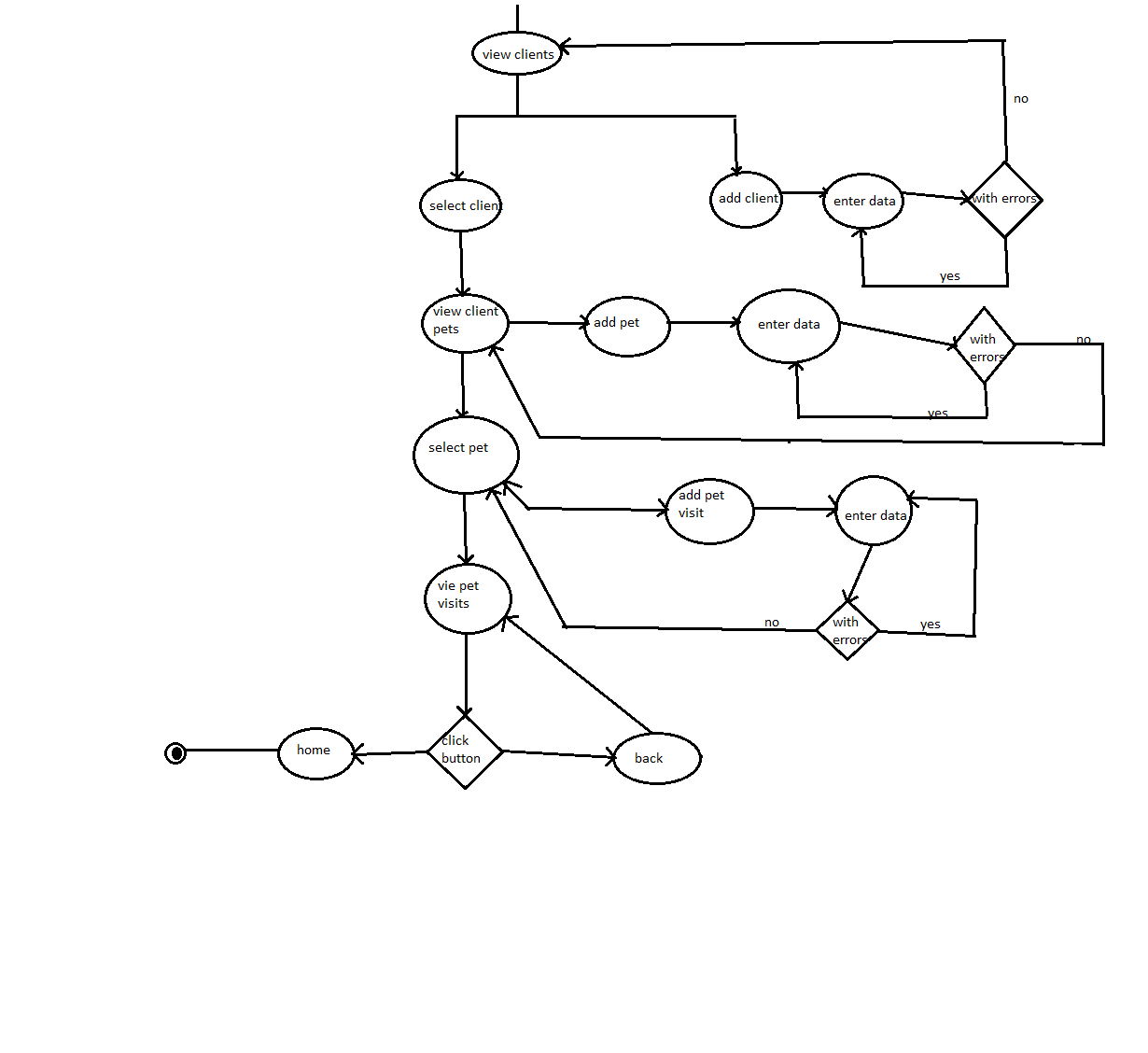
This happens when a user does not have an account. Users provide their signup credentials which are stored by firebase. The signup sequence is similar to the login sequence except that the credentials are stored as the first time data and therefore no wrong credentials exception.

1. **Main activity**

The main activity is tabbed with four fragments. The fragments are: home, profile, analysis and database.

Activity diagram



**Fig 2**

**Security Detailed Design**

Security is achieved through caching of user credentials in the mobile secure key store.

Authentication by firebase framework is achieved through secure communication to the server over an encrypted channel. The application is signed with key where signature is in SHA-1 format.

Once the communication channel is established and identity of the application is verified, the data transmission is initiated.

Authorization is maintained during account creation. Each account is only allowed to read and write every database in their domain.

Execution privileges are only allowed for the developer.

Auditing is supported by verbosely logging the trail to storage/emulated/[user ID].

Intrusion Detection System is powered by firebases server against various attacks. Once an attacker tries and is detected the fingerprint is blacklisted and cannot connect again.

**Performance**

The heap size is limited due to the little dependencies and a small file size. The application utilizes a RAM capacity of approximately 80 mbs.

Though a storage as a service is implemented, availability is guaranteed since the connection is not mandatory and data will be synchronized only when connection is available else cached locally in internal storage.

In the application manifest by setting allowbackup=TRUE allows back up of the application state alongside system backup.

In the event of a re-install the data is restored and user can continue working with previous clients easily.

**Communication**

For internal I/O functions an interface class exists that spawns across the entire Application domain.

By inheriting from the class R (a superclass) data from every interface can be easily accessed.

UI components before compilation are coded/structured under the XML format. The resource file tells the system where and how the window manager positions the user interface on the screen with varying display sizes.

The device modem provides a channel for communication between server and the application.

Therefore a catastrophic failure would occur if the device modem would be faulty or unavailable. A connection is required to provide authentication and fetch data in-case of a re-installation of the application.

**TESTPLAN**

Introduction

1.1. General information

This document describes the methods and procedures that will be used by the group in the functional testing process of the PetState mobile application.

It is meant to be used as a manual during testing works. It describes the procedure of the testing process. The test plan is intended for the group 3 members.

The objective of the testing activities is to check functions and features of the Pet State application

1.2. Purpose

This Test Plan document for the Pet State project supports the following objectives:

● Identify existing project information and software components to be tested.

● Recommendation and description of the testing strategies to be employed.

● Identify required resources and provide a test effort estimate

● List the test project deliverable elements.

The results of test execution will be noted and addressed by the respective group member developing the noted area.

2. Scope of project

2.1. Scope of Pet State mobile application test plan.

The following components and functions would be tested:

1. Create an account using email.

2. Create an account using Social media accounts.

3. Login.

4. Password recovery

5. View client details

6. Modify client details

7. Modify pet records.

8.Vet consultation payment

3. Work plan

The parties agreed to follow the next work plan:

1. Test plan preparation

2. Test plan approval

3. Functional testing and bugs reporting

4. Daily reports preparation

5. Final report preparation

4. Test Plan and Strategy

4.1. Functional testing

The objective of functional testing is to make sure that the whole software product works according to the requirements, and no significant errors appear in the application.

Functional testing is the most substantial part of software testing. It involves checking different aspects of the system. A software product must pass all the planned tests. Only in this case its quality can be assured.

Test Objective

Ensure proper target-of-test functionality

Technique

Execute each use case, use-case flow, or function, using valid and invalid data, to verify the following:

● The expected results occur when valid data is used.

● The appropriate error or warning messages are displayed when invalid data is used.

● Each rule is properly applied

Completion Criteria

● All the planned tests are performed.

● There are no show-stopping errors.

● All the errors of high priority and severity are fixed.

● The test results are evaluated, discussed and approved.

Special

Considerations:

None.

4.2. Test Procedure

Test procedure assumes the next points:

● Reporting of found software bugs.

Various aspects of the application will be checked by executing of different testing types such as:

● Functional Testing

● UI Testing

● Usability Testing

● Compatibility Testing (devices)

It also will be checked how the software product is run on the browsers and devices that are supposed to support it, how it starts and stops, and how much time it needs to launch.

4.3. Bug Reports

Bug reports are created in order to provide the development team and the project managers with exhaustive information about the discovered defects. They must be helpful in determining causes of the errors and correcting them.

Defect Severity can be classified into four categories:

● Critical (blocker) defects are the failure of the complete software system or of a critical subsystem, and no work or testing can be carried out after the occurrence of the defect. It also applies to data loss failures and with processes that leave inconsistent data stored in the database.

● Major defects (and crashes) are those which also cause failure of the entire or part of the system, but there are some processing alternatives which allow further operation of the system. It also applies to the system crashing, or aborting, during normal operation of a non-critical flow.

● Minor defects do not result in failure but cause the system to show incorrect, incomplete, or inconsistent results.

● Trivial defects are small errors that do not affect the functionality: typos, grammar mistakes, wrong terminology, etc.

The information that is indicated in each bug report:

● the software product name;

● version number of the software product

Each report provides the next information about the defect:

● summary, which is short description of the problem;

● location of the defect in the software product;

● steps to reproduce the error;

● frequency of the defect occurrence;

● severity of the defect;

● additional information about the defect in the form of attached screenshots or video records.

5. Resources

5.1. The list of the devices

Name of the device

OS

iPhone devices

All supported OS

Android devices

All supported OS

6. The criteria of quality

The product should operate in accordance with the requirements and the functional specification. The product should not contain critical and blocking defects in the final version of the project.

7. Test Team Expectations

The test team must be provided with valid, updated documents during the whole testing process.

All show-stopping errors must be corrected as soon as possible.

The developers should correct all the errors in the software modules before releasing the application.

8. Responsibilities of Test Team Members

Project Manager

● Managing the whole testing process.

● Providing all the needed resources for the testing activities.

● Analyzing the tasks and distributing them between team members.

● Creating test documentation, including test cases, test plans, etc.

● Proposing best practices and tools for a project.

9. Deliverables

● Test Plan.

● Bug reports and reports regarding the testing progress.

**PETSTATE PROJECT TESTCASES**

**Registration page**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case number** | **Feature** | **Test case description** | **Test steps** | **Test data** | **Expected results** |
| TC 01 | User interface | Check all the textboxes, radio buttons, buttons and dropdowns | 1. Click on buttons and dropdown menus | NA | UI should be perfect |
| TC 02 | Required fields | Check required fields by not filling data | 1. Do not enter any values in the fields 2. Click on register | No data | Should show mandatory symbol on mandatory fields |
| TC 03 | Required fields | Check user should Register by filling all the required fields | 1. Enter valid values in the required fields. 2. Click the Register button | Name  Institution name  Phone number  Email  Address  Password | Users should be registered successfully. |
| TC 04 | Email validation | Check the Email text field | 1. Enter valid Emails 2. Click on the Register Button | [Pet@gmail.com](mailto:Pet@gmail.com)  Vet@gmail.com | Valid emails |
| TC 05 | Password Validation | Check the password when passing valid data | 1. Enter value in alphanumeric which is in between 6-32 2. Click on Register button | 123456 | It should not show any validation message |
| TC 06 | User type | Select which type of a user are you | 1. Click on the drop down menu 2. Select if signing up as a vet or a pet owner | NA | A type of user should be selected |

**Login page**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case number** | **Feature** | **Test case description** | **Test steps** | **Test data** | **Expected results** |
| TC 01 | User interface | Check all the textboxes, radio buttons, buttons and dropdowns | Click on radio buttons, buttons and dropdowns | NA | UI should be perfect  Text boxes and buttons should align |
| TC 02 | Required fields | Check required fields by not filling data | 1. Do not enter any values in the fields 2. Click on login | No data | Should show mandatory symbol on mandatory fields |
| TC 03 | Required fields | Check user should login by filling all the required fields | 1. Enter valid values in the required fields. 2. Click the login button | NA | Users should be logged in successfully. |
| TC 04 | User Login | Check When passing a correct email and invalid password | 1. Enter valid email 2. Enter incorrect password 3. Click on Login Button | [pet@gmail.com](mailto:pet@gmail.com)  pass123467 | User should not log in and should show proper error message |
| TC 05 | User Login | Check when pass correct email and password | 1. Enter valid username 2. Enter valid password 3. Click on Login Button | [pet@gmail..com](mailto:pet@gmail..com)  123456 | User should login |
| TC 06 | Email validation | Check the Email text field | 1. Enter valid Email 2. Click on the Register Button | [Pet@gmail.com](mailto:Pet@gmail.com)  Vet@gmail.com | Valid emails |
| TC 07 | Password Validation | Check the password when passing valid data | 1. Enter value in alphanumeric which is in between 6-32 2. Click on Register button | 123456 | It should not show any validation message |
| TC 08 | User login | Check if the password is entered in encrypted | 1. Enter valid email 2. Enter password 3. Click on Login Button | [pet@gmail.com](mailto:pet@gmail.com)  123456 | Password is entered in encrypted form |
| TC 09 | Signup Option for new users | Check whether the signup link for the new user is working | Click Signup link | NA | Clicking signup link takes the user to signup page successfully |
| TC 10 | Forgot Password | Verify Forgot Password sends a forgot password link. | 1. Click on the Forgot Password link. 2. Enter Email and click on the send code button 3. Now go to mail | NA | User should get the forgot password link on his/her email id. |
| TC 11 | User type | Select which type of a user are you | 1. Click on the drop down menu 2. Select if logging in as a vet or a pet owner | NA | A type of user should be selected |

**Add new client**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case number** | **Feature** | **Test case description** | **Test steps** | **Test data** | **Expected results** |
| TC 01 | Adding a new client | Check all the fields are filled | 1. Enter the client details 2. click on the save button | John Doe  0712345678  [pet@gmail.com](mailto:pet@gmail.com)  50309 | The data should be saved successfully |
| TC 02 | Required fields | Check required fields by not filling data | 1. Do not enter any values in the fields 2. Click on save | No data | Should show mandatory symbol on mandatory fields |

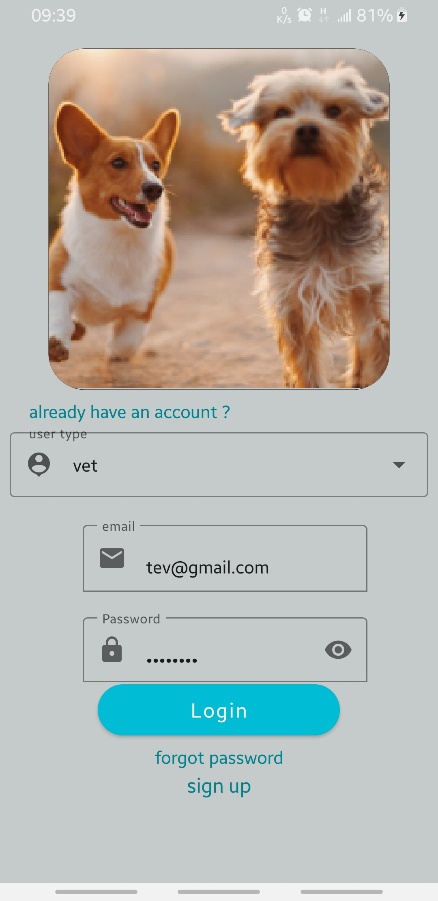
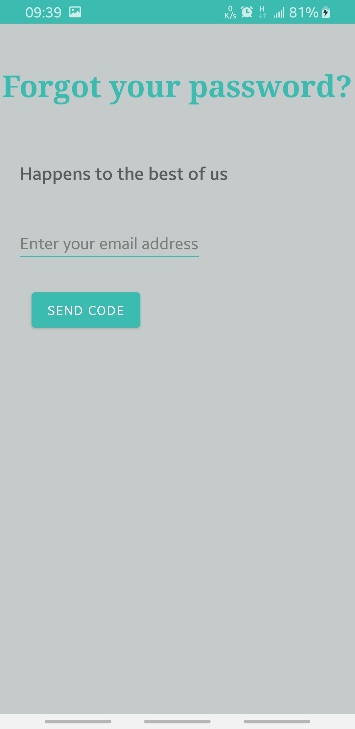
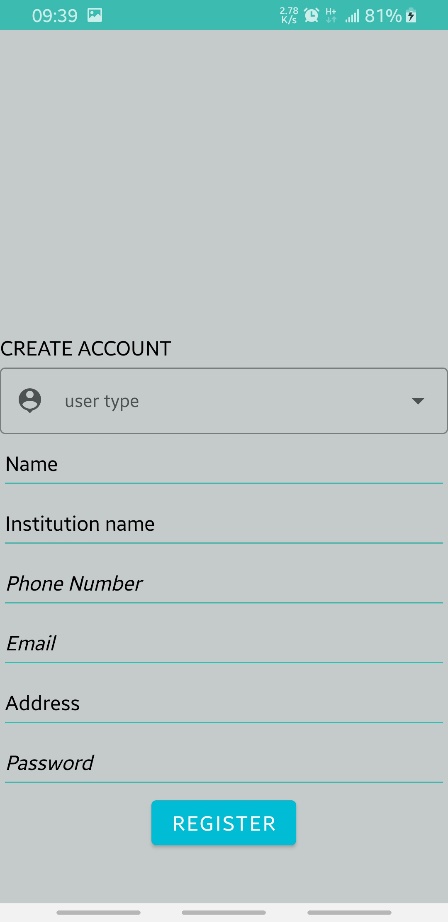
**Add a new pet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case number** | **Feature** | **Test case description** | **Test steps** | **Test data** | **Expected results** |
| TC 01 | Required fields | Check required fields by not filling data | 1. Do not enter any values in the fields 2. Click on save | No data | Should show mandatory symbol on mandatory fields |
| TC 02 | Required fields | Check user should save by filling all the required fields | 1. Enter valid values in the required fields. 2. Click the save button | Kelly  12 weeks  Dog  Chihuahua  white | Details should be saved successfully. |

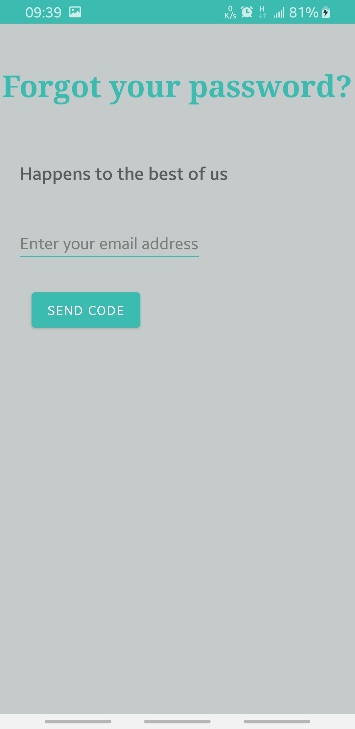
USER MANUAL

This document is a comprehensive guide on how to navigate through PetState.

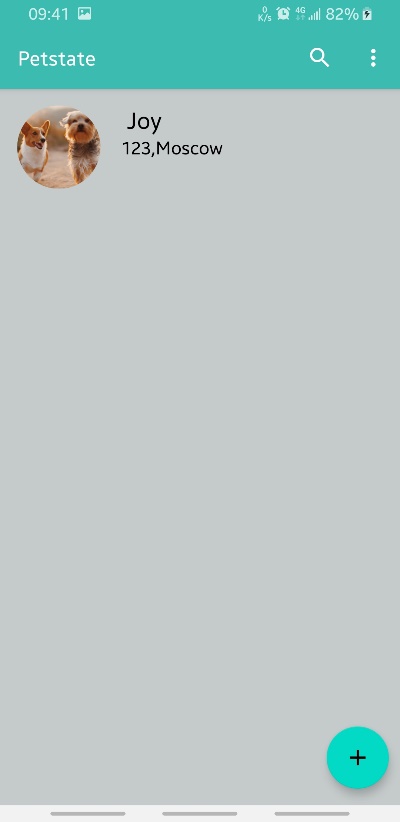
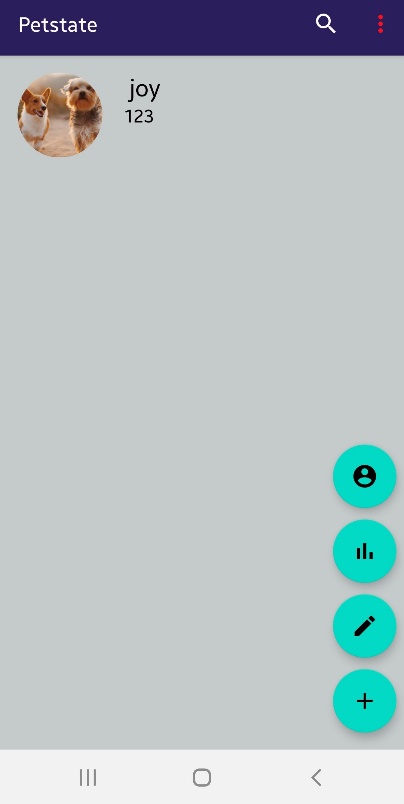
On opening the app, you’ll be directed to the login window where you have the option of entering your login credentials then entering the system upon clicking the login button, you can click sign up to create an account or click on forgot password in order to reset your account password.

Below are screenshots of the login, signup and forgot password windows.  

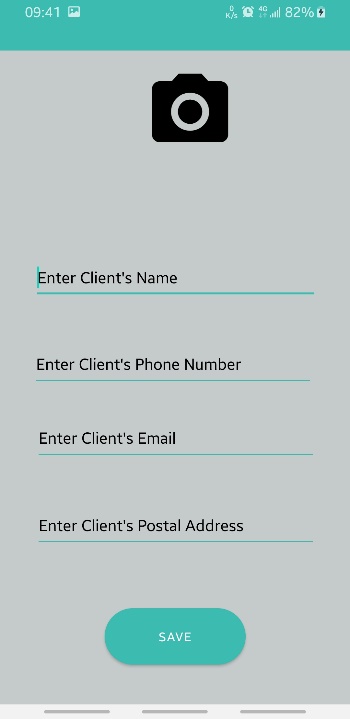
In the case of forgotten password, click on the forgot password button enter your email address and click send code



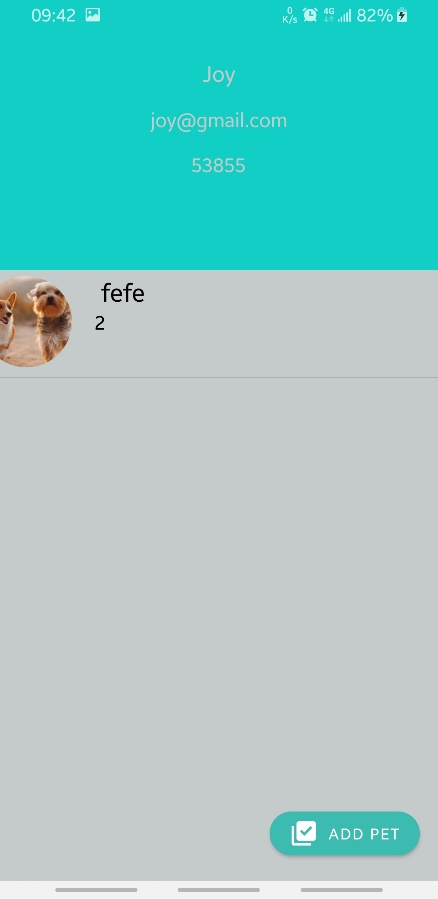
As a **vet**, to be able to add a pet to the database, you have to add a client first. Click on the floating action plus button that appears on the bottom of the screen and three other buttons appear.

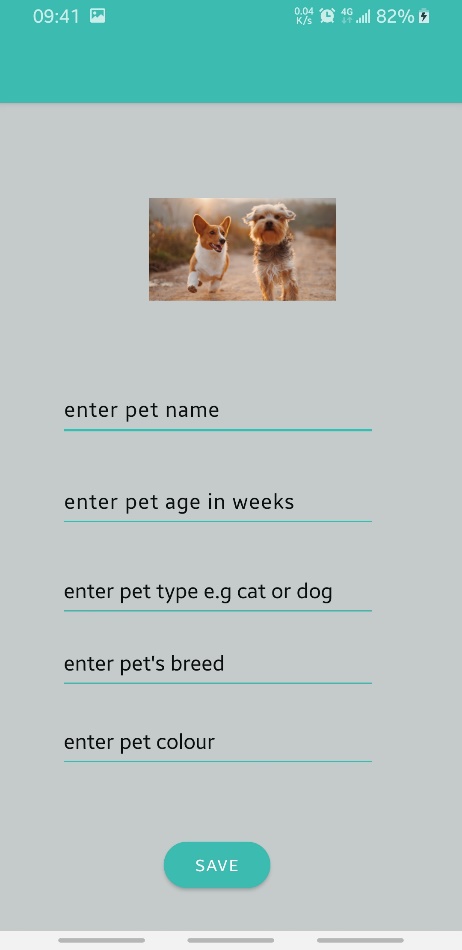
If you click on the button with a user icon, (The top most button) this activity allows you to add a new client to your records. All details must be filled for the data to be saved.



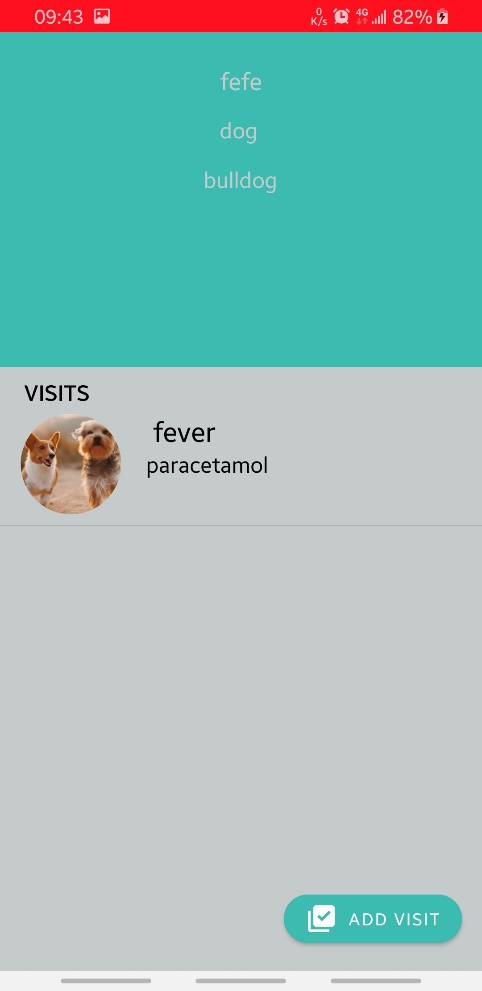
If you click on the client you create (in this case: joy), you will be able to see the client’s list of pets as well as information you had earlier saved about your client.



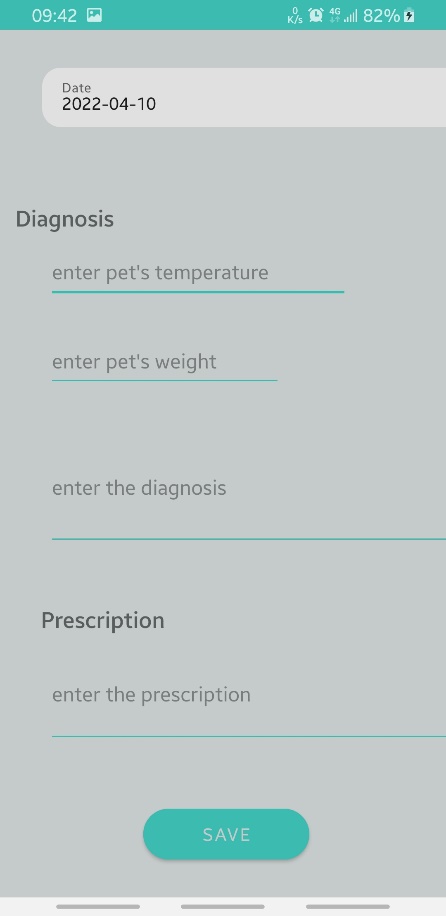
Clicking on the add pet button above will enable you to add a new pet into the system through the following window



On clicking an individual pet, (in this case: fefe), you will be able to see the past visits the pet has made as well as details of the pet you had previously added.

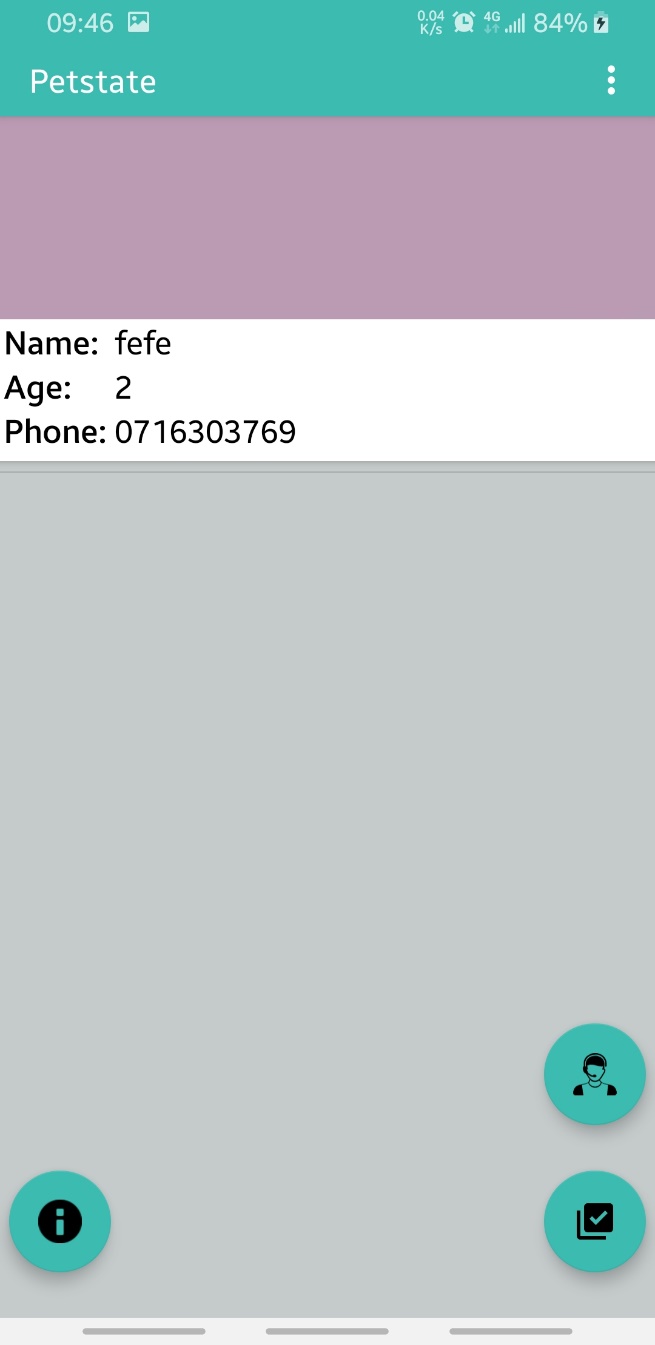


On clicking the add visit button above, you will be able to add a new visit for the same pet through the window shown below.



Once the visit is saved, you can click on it to see exactly what you had saved.

As a **pet owner**, on gaining access to the system, you’ll be able to see the list of past sick animal reports you had filled as well as three buttons as shown below



On clicking the information button (the left most button) you’ll be able to see information charts for vaccination and nutrition of the pet as shown below.



If instead you click on the bottom right button of the pet owner’s home screen, you’ll be able to fill in the sick animal report as shown below

