CSCI5708 – MOBILE COMPUTING

Project Design

MACS - GROUP 5

Aditya Mahale – B00867619
Khushalkumar Maganbhai Gondaliya – B00912092
Kishan Mahendrabhai Savaliya - B00896729
Navya Jayapal -B00886554
Vatsal Yadav – B00893030

1. Abstract

The loyalty of dogs toward humans makes them our best friends. Stray dogs in many countries are neglected. There is no organization in these countries to take care of them. We are developing an app that enables humans to help dogs who are in grave danger and need a caring family. Our idea is to create an app that provides a centralized platform for connecting animal shelters and adopters. It will have an interface to connect registered users with veterinarians associated with a shelter. The app will also provide an interface for shelter users to add new pets and accept donations.

2. Introduction

An app or a website for adopting strays is quite common in countries such as USA and Canada. However, stray animals in India do not get the same privilege as their counterparts in North America. Especially during COVID times, in India, stray dogs were not fed due to which many of them starved to death. Another instance is animal cruelty, where a pack of wild dogs was acting as a nuisance because they were feeding on an endangered bird in a rural region in Rajasthan. They were instructed to be killed by the local governing body without any remorse.

A couple of instances from first-hand experience back in India: A lady in Delhi personally rescues stray dogs and puppies who are abandoned in a garbage dumpster. She bears the cost of any veterinary treatment of the dogs and brings them to her home. Her house is full of happy dogs grateful for her empathy for them. Her only means of knowing where to find abandoned dogs is through word of mouth and a few connections she made on WhatsApp. There is no systematic way for her to find out all the dogs who need rescuing. Also, there's a limit to the number of dogs she can bring home. Another instance is of a man who fed stray dogs in Pune during COVID times. However, he could only do this in his local region, and he had no means to broadcast this information to other places where help was needed. Studies suggest that stray dogs who live in the cities of India are often overwhelmed due to extreme noise and pollution. They deserve a lot better than this. These reasons motivated us to create an app that addresses all these issues.

One of the local websites for adoption is Lily Pad. The website offers services for the adoption of cats. Another service is to provide cat supplies such as food, toys, grooming products, etc. The idea is very similar to ours, but the services are geographically restricted to Canada. In India, the is no such website for connecting animal shelters and adopters. Another example of the app is 911 rescue for animals called "Let it Wag". The app allows people to call and rescue strays who are hurt or in a bad condition. This is a wonderful app that cares for and provides immediate rescue to animals in need. Our idea is an extension of this app which allows users to adopt the dog to be a part of their family.

3. User Persona and Target Audience

Stray dogs are considered a nuisance in most developing and underdeveloped countries. They do not have the necessary facilities to accommodate and care for them. One of the reasons for this is the lack of awareness among people of how to treat animals. Due to all these reasons, our target audience is majorly in these countries.

1. Pet Adopter User Persona

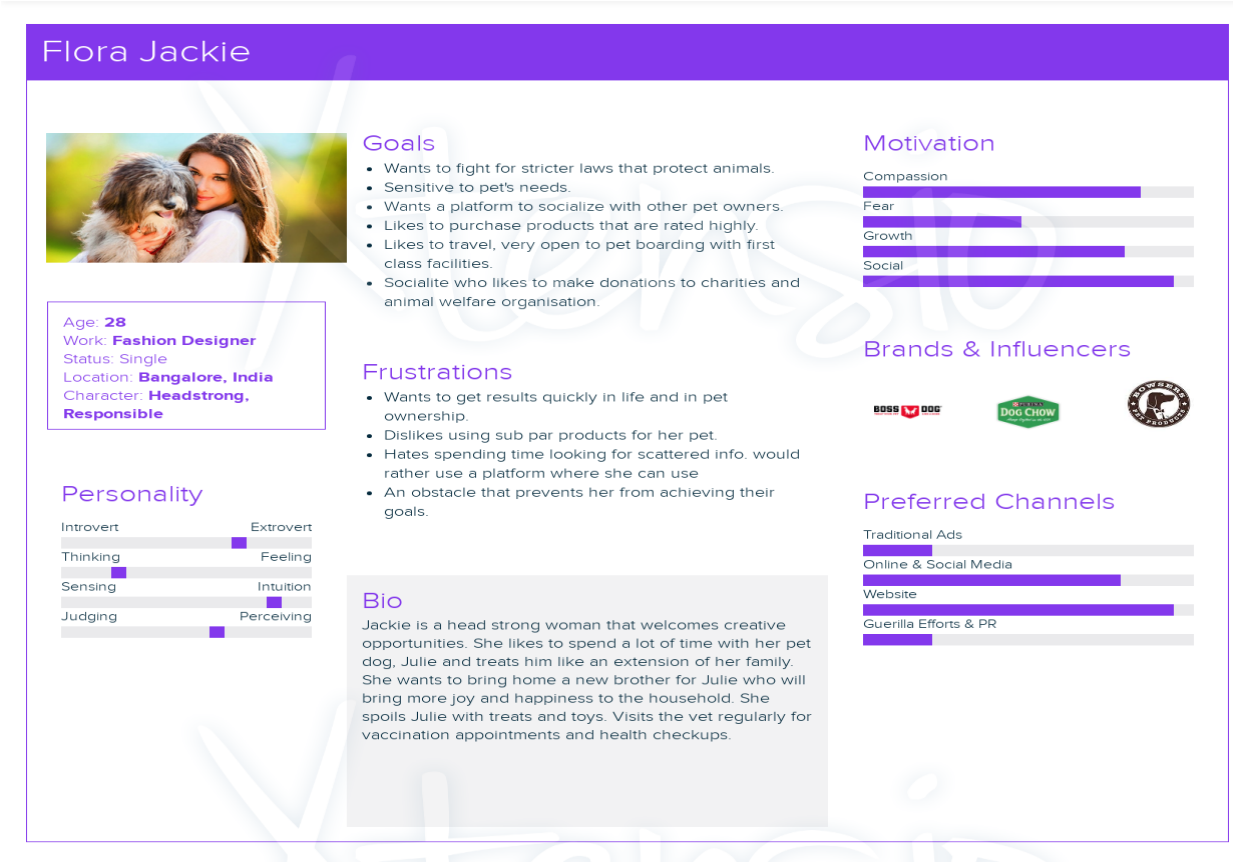

Figure 1 Adopter User Persona [1]

2. Shelter Employee User Persona

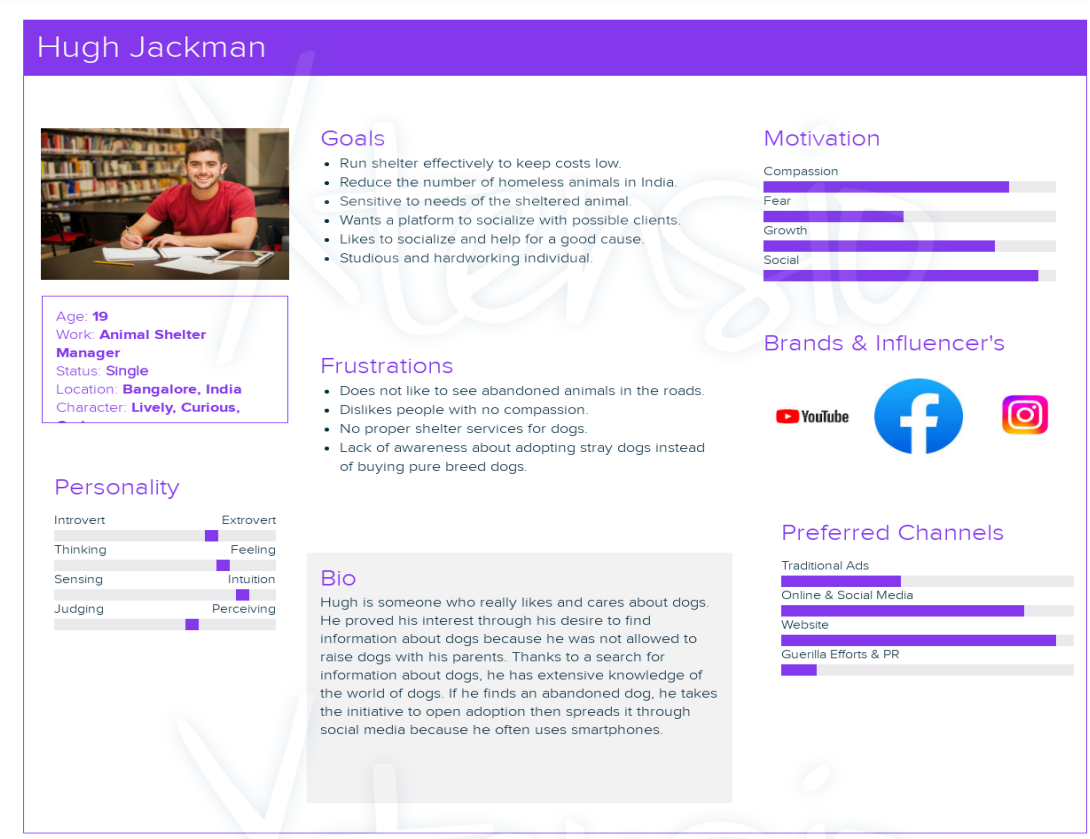

Figure 2 Adopter User Persona [1]

4. Data Model

4.1 ER Diagram

Figure 2 shows the ER diagram of the application. This explains how we will store the data. The **Shelter** entity has a field such as its own unique id, name, and address which is a composite attribute made of street, city, and zip code. Using zip code, we can get information about the location which is shown here as derived attributes, latitude, and longitude.

Shelters can have many pets which can be identified by their unique Ids. Users also have unique identifiers and other details as shown in the diagram. Users can donate to the shelter and it will be recorded by a unique id, userid who donated, sheltereId to indentify shelter to which the donation has been made.

The veterinarian will have a separate table. All shelters have veterinarians and users can book their appointments. In the appointment table, userid, slotId, doctorId will be stored. After a successful booking of an appointment, on the application, we will disable the slot to prevent conflict in bookings.

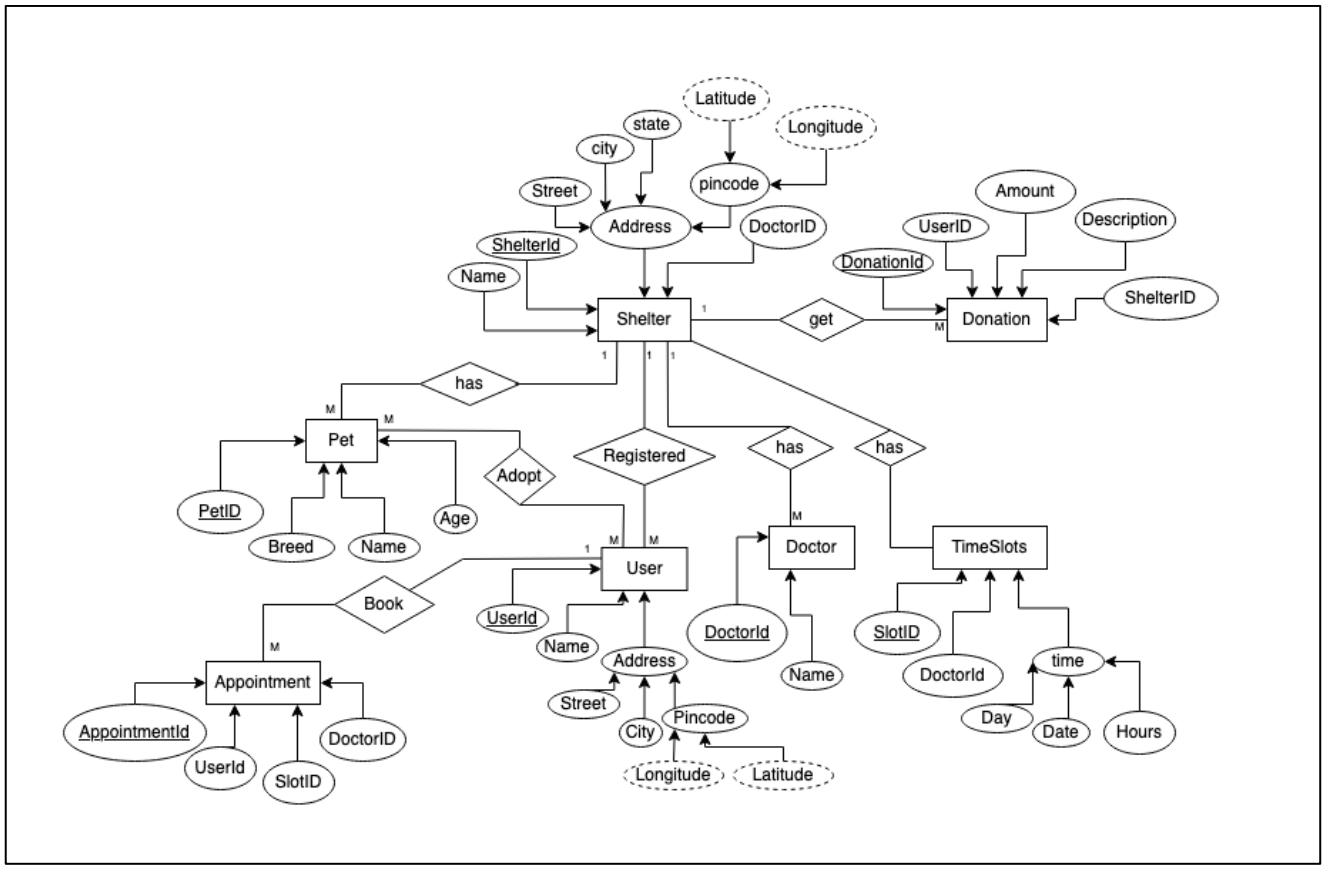

Figure 2 ER Diagram of the application [2]

5. View

5.1 Augmented Sitemap

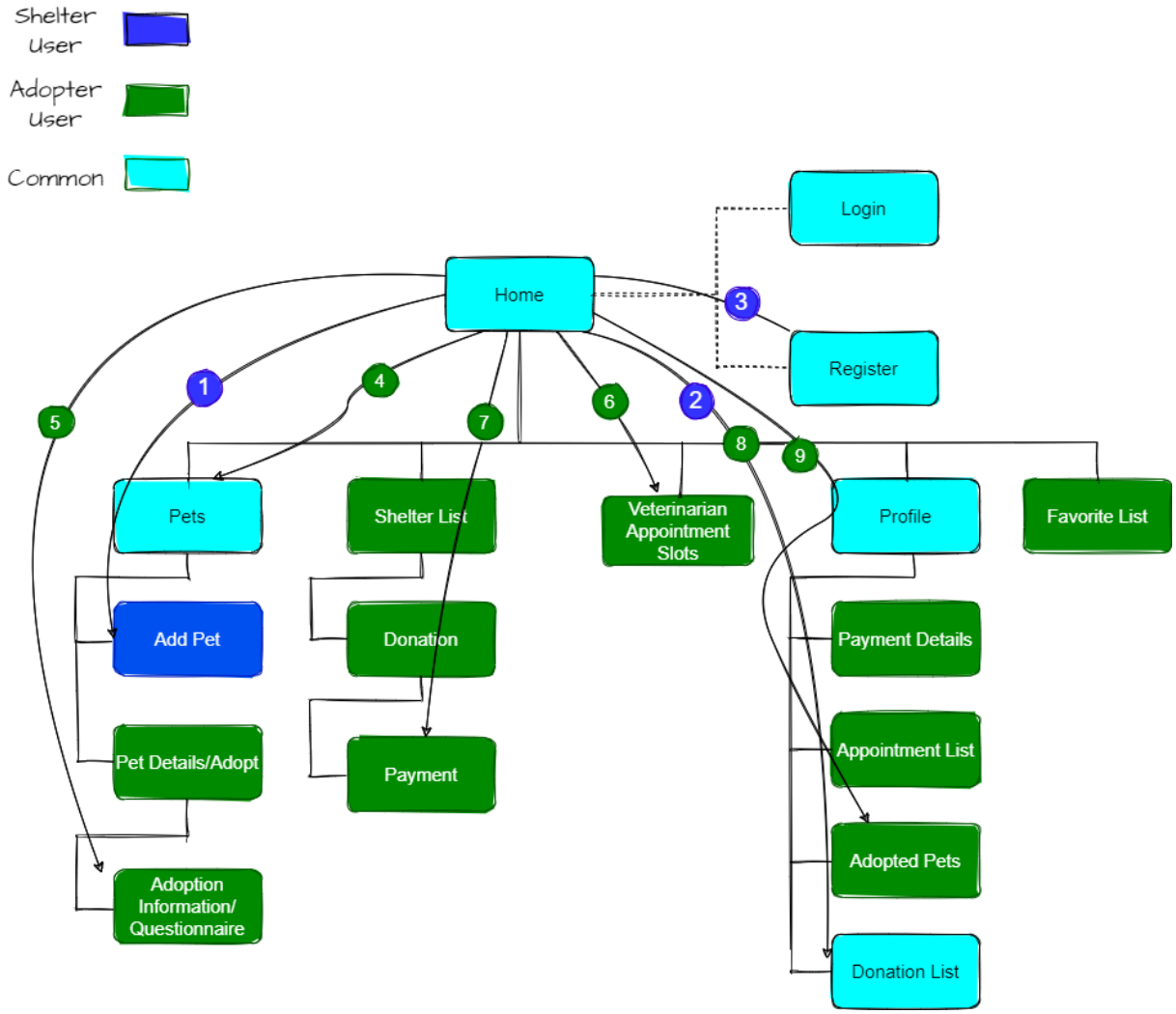

Figure 3 Augmented Sitemap

Use Case 1: Add Pet

1. An authenticated shelter user opens the app.
2. The system displays home page.
3. The user clicks on the pet's icon.
4. The system displays pets listing page with "Add Pet" FAB.
5. The user clicks on the "Add Pet" FAB.
6. The system opens the "Add Pet" form.
7. The user can click on upload image to add image of the dog.
8. The user can click on add text to add description of the dog.
9. The user clicks on submit to add the dog to the dog listings.
10. The system adds the pet to the list.

Use Case 2: View Donations

1. An authenticated shelter user opens the app.
2. The system displays home page.
3. The user clicks on profile icon.
4. The system displays list of profile options.
5. The user clicks on the “Donations List”.
6. The system displays all the received donations.

Use Case 3: Add Veterinarian

1. An unauthenticated shelter user opens the app.
2. The system displays home page with register button.
3. The user clicks on the “register” button.
4. The system shows shelter information form along with fields for adding veterinarian details associated with that shelter.
5. The add adds veterinarian information and clicks on register.
6. The system successfully adds the veterinarian information for that shelter.

Use Case 4: Searching a pet

1. An authenticated user opens the app
2. The system displays home page with list of pets.
3. The user clicks on the search icon and enters the breed name
4. The user checks the pet listings.
5. The user can click on any pet listing.
6. The system shows photos of the pet and information about it.

Use Case 5: Adopting a dog

1. The user opens the app
2. The system displays home page with list of pets.
3. The user checks the dog listing.
4. The user can click on any dog listing.
5. The app shows photos of the dog and information about it.
6. If user wants to adopt the dog, he/she will be redirected to the adoption information page
7. The user will be given a form to fill up with their information which will be validated by the dog shelter team.
8. The user will be contacted by the team via email and a link will be sent to continue the adoption process.
9. Once the user has guaranteed the team that they will adopt the dog, notification with details about the date and time of adoption will be sent to the user.

Use Case 6: Booking a veterinarian appointment

1. An authenticated user opens the app.
2. User clicks on the veterinarian listing option.
3. App displays the veterinarian listings.
4. User clicks on one of the listed veterinarians.
5. System displays available time slots.

6. User chooses one of the available time slots.
7. System displays confirm booking prompt.
8. User clicks on confirm.
9. System displays a successfully booked message.

Use Case 7: Donate money to shelters

1. The user opens the app
2. The app shows a list of shelters that accept donations.
3. The user selects the Shelter to which the user wants to donate.
4. The app shows the page with a text box to enter the donation amount the user wants to make.
5. The user enters the amount and clicks on the payment button.
6. The app shows the payment page with payment options.
7. The user can make a payment using PayPal.
8. The user enters PayPal details to make a payment.
9. Paypal APIs will verify details, and payment will be processed.
10. Money will be credited to Shelter's bank account.

Use Case 8: View all donations

1. An authenticated user opens the app.
2. The system displays home page.
3. The user clicks on profile icon.
4. The system displays list of profile options.
5. The user clicks on the “Donations List”.
6. The system displays all the given donations.

Use Case 9: Viewing adopted pets

1. An authenticated user opens the app.
2. The system displays home page.
3. The user clicks on profile icon.
4. The system displays list of profile options.
5. The user clicks on the “Adopted Pets”.
6. The system displays all the adopted pets.

5.2 Wireframes

1. Sign up as a user

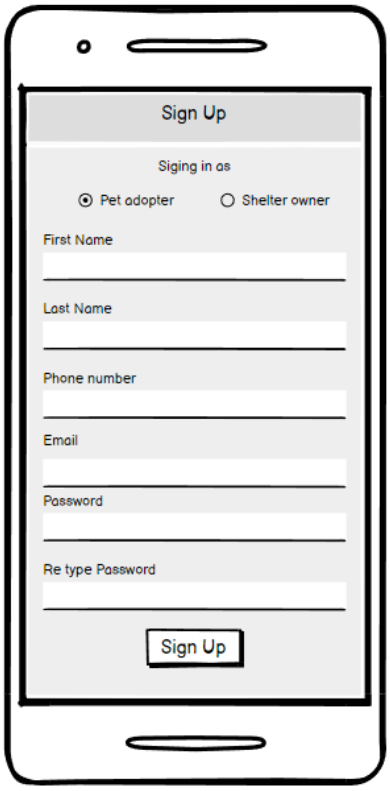

Figure 4 Wireframe for the sign up as a user

2. Login page

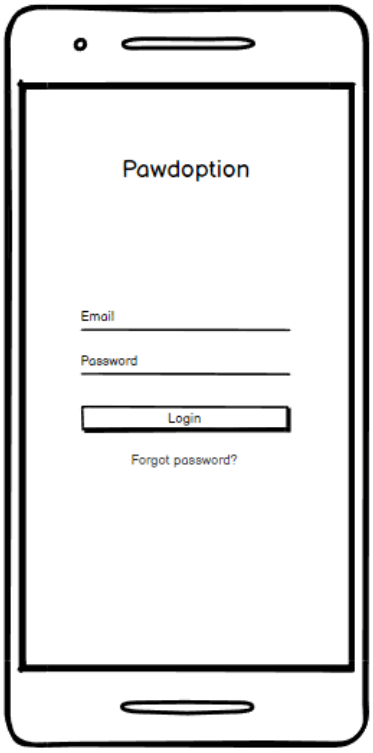

Figure 5 Wireframe for the login page

3. Sign up as a shelter

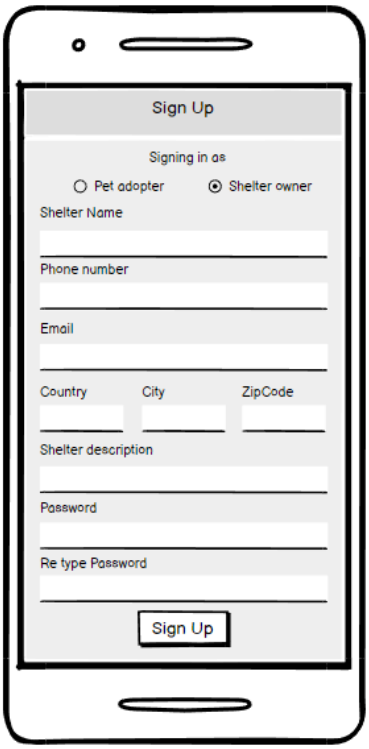

Figure 6 Wireframe for the sign up as a shelter

4. To check the list of donations received for a shelter

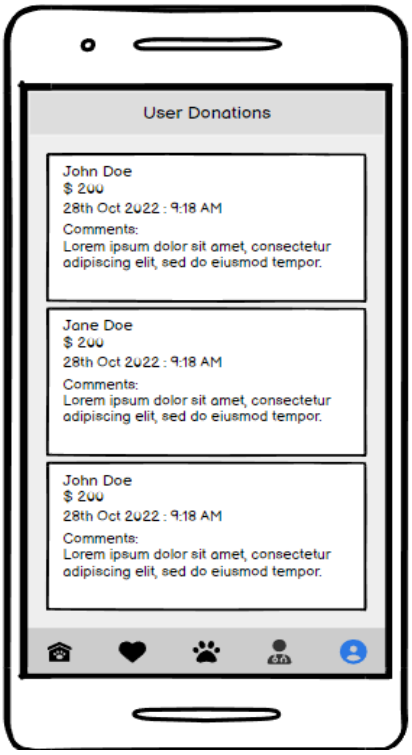

Figure 7 Wireframe for a shelter to see list of donations received

5. Add pet posting

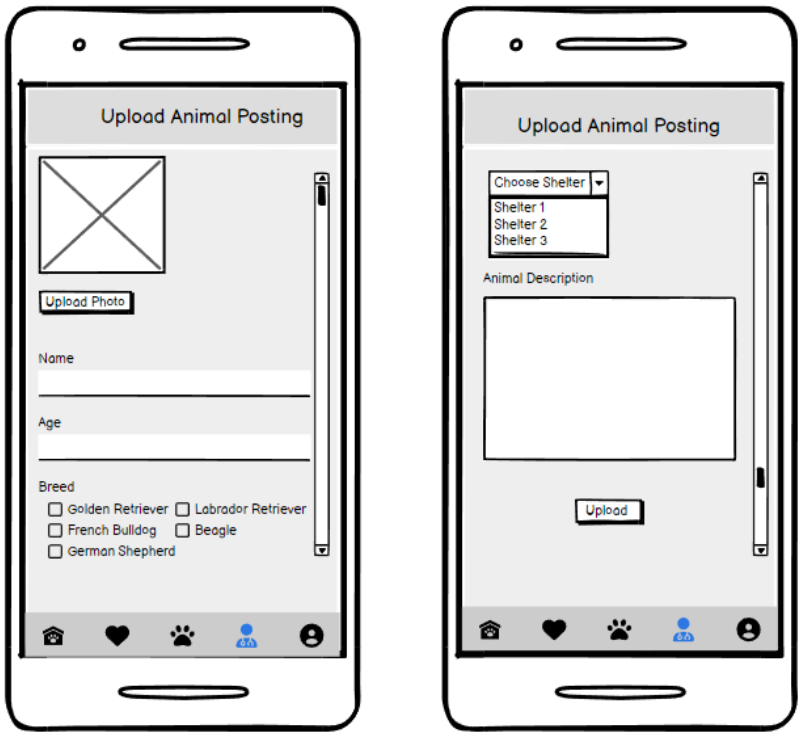

Figure 8 Wireframe to add the pet posting by the shelter

6. List of Veterinary Doctors (shelter view)

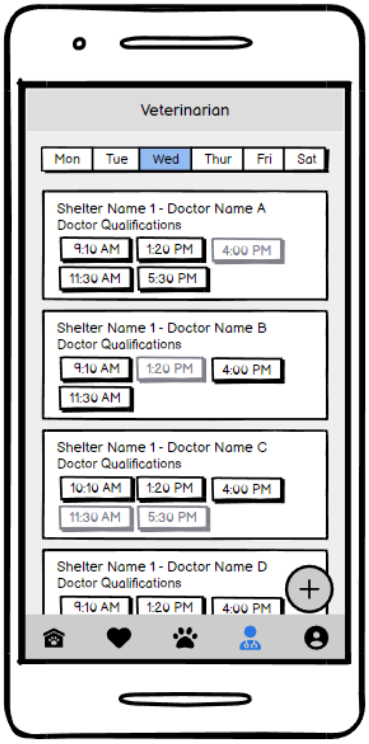

Figure 9 Wireframe for the list of appointments available for a shelter

7. Veterinary Form

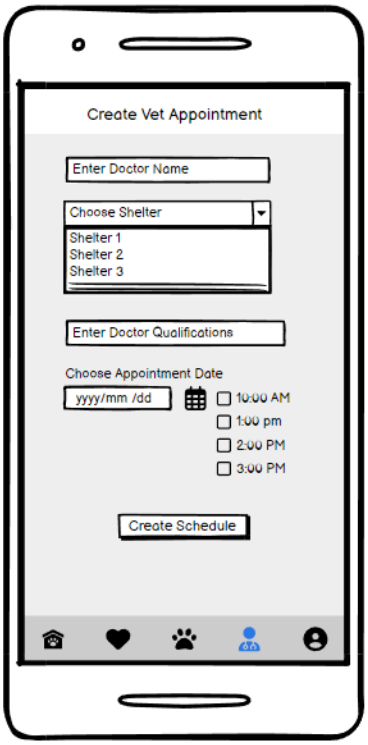

Figure 10 Veterinary form

8. View veterinarian appointments

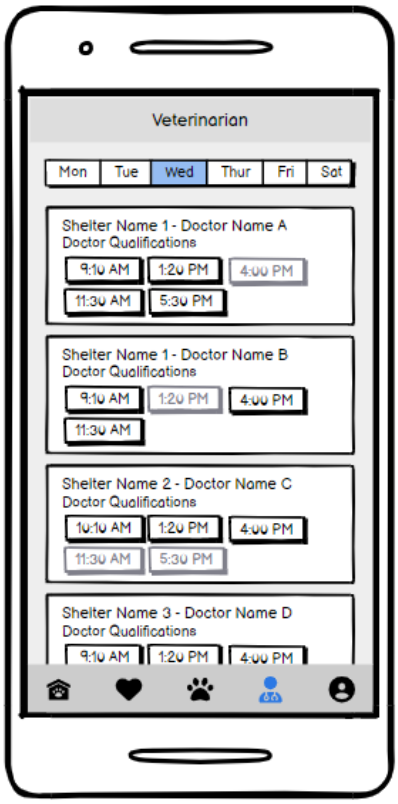

Figure 11 Wireframe for viewing list of available veterinarian appointments using Balsamiq [BAL]

9. Book veterinarian appointment

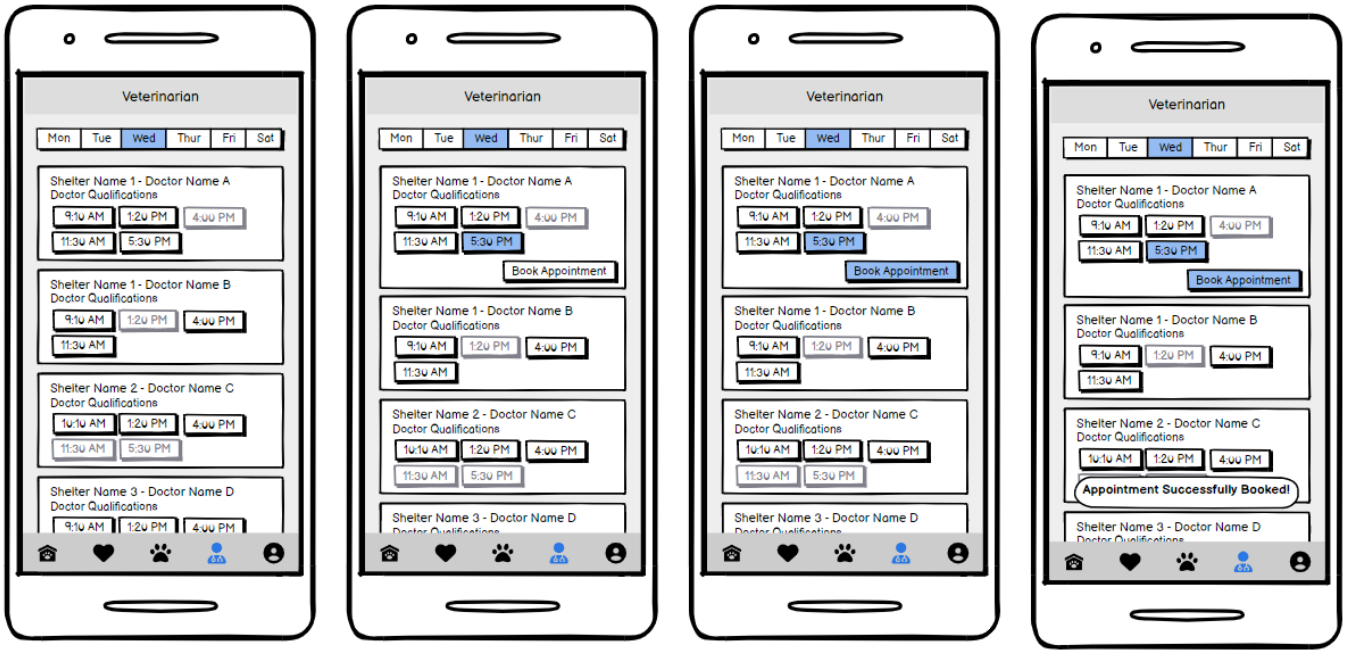

Figure 12 Wireframe for booking a veterinarian appointment using Balsamiq

10. View nearby shelters

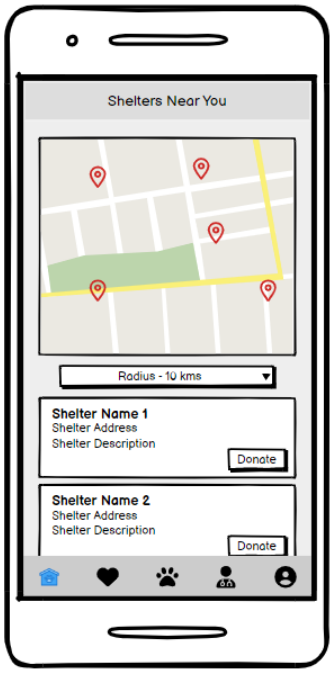

Figure 13 Wireframe for viewing list nearby shelters using Balsamiq

11. Donate to a shelter

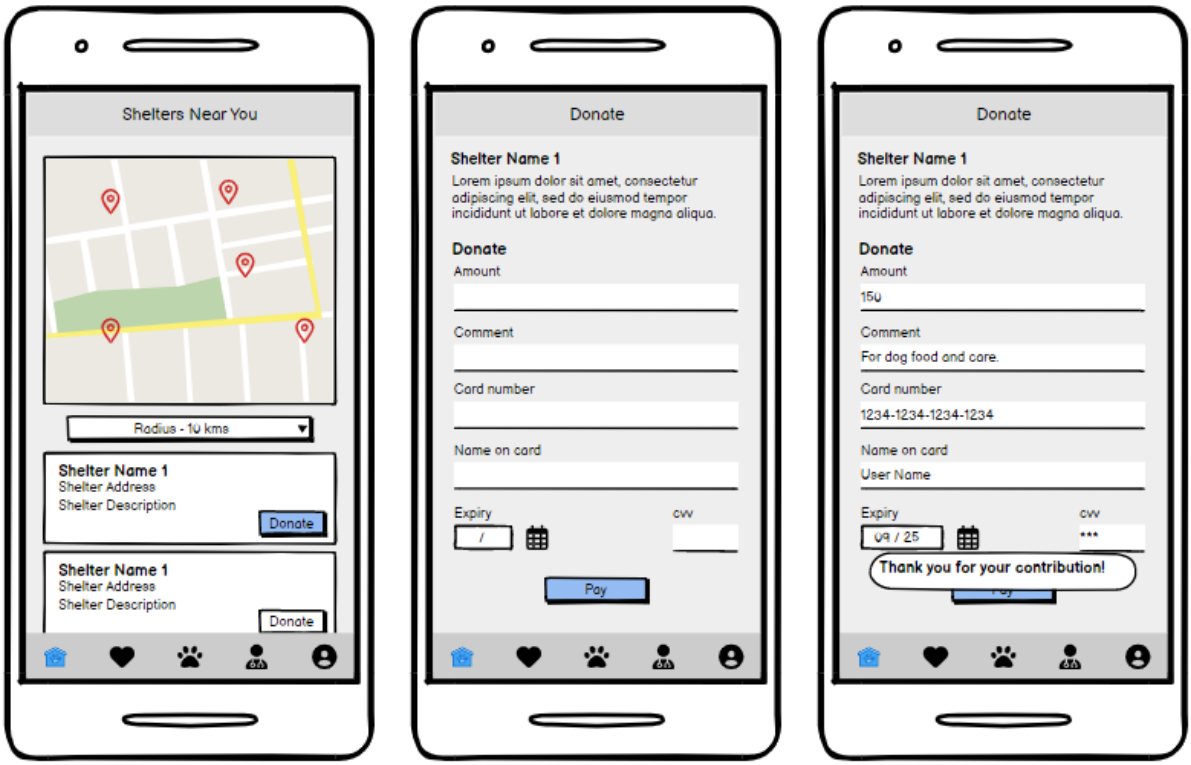

Figure 14 Wireframe for donating to a shelter using Balsamiq

12. View list of pets

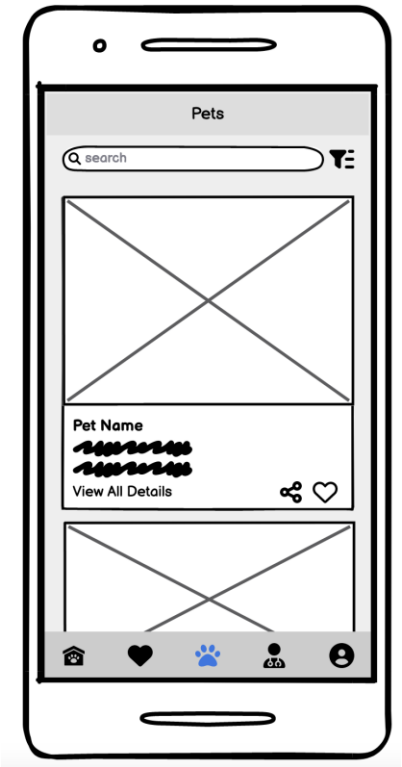

Figure 15 Wireframe for the list of pets

13. Share pets with friends

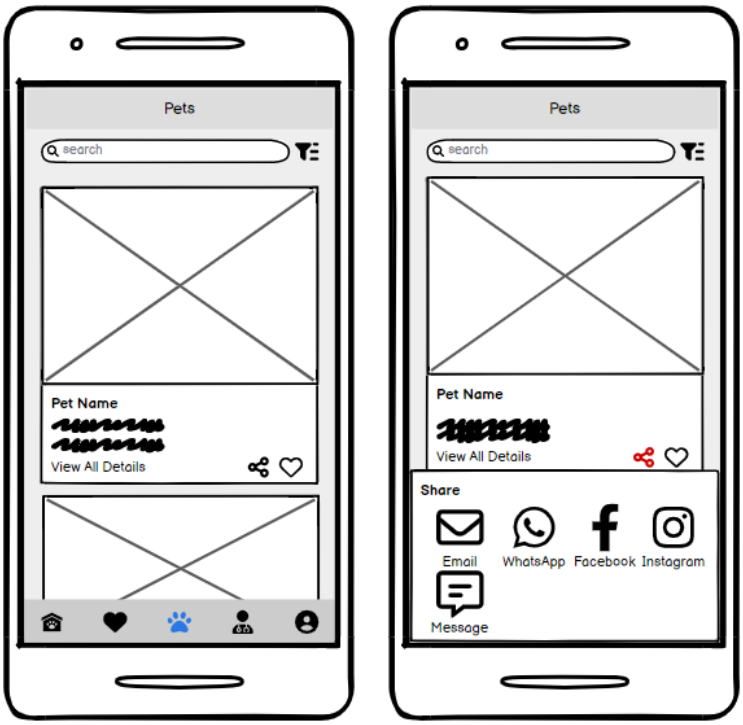

Figure 16 Wireframe for sharing pets using Balsamiq

14. Add pet to favorites

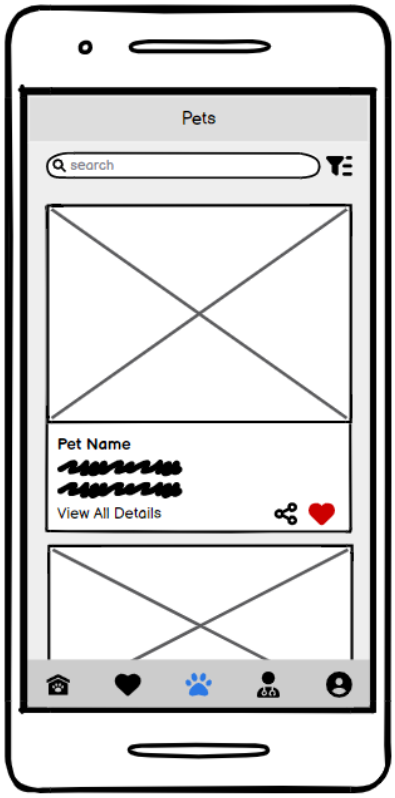

Figure 17 Wireframe for adding a pet to favourites using Balsamiq

15. Search for a Pet

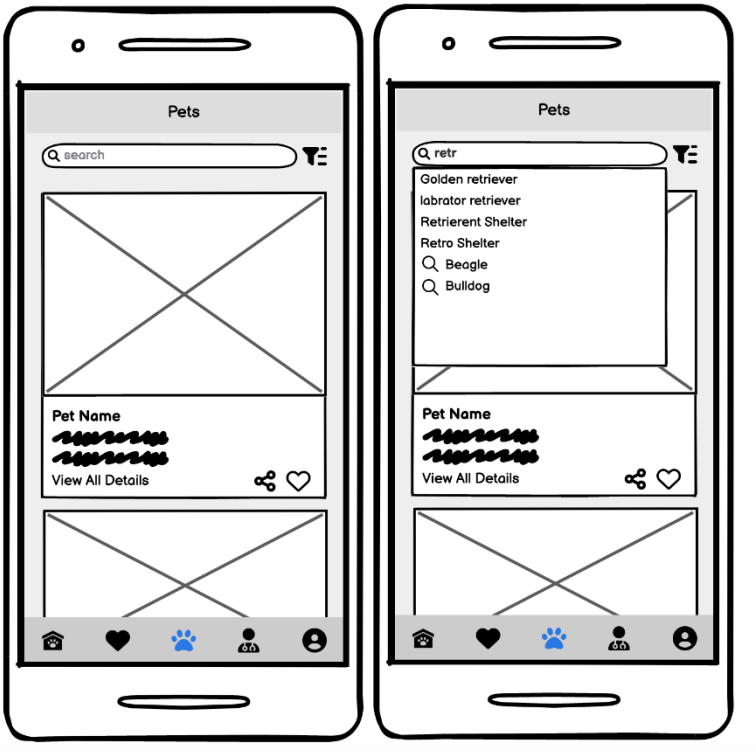

Figure 18 Wireframe to search for a pet in search bar

16. View information about the pet

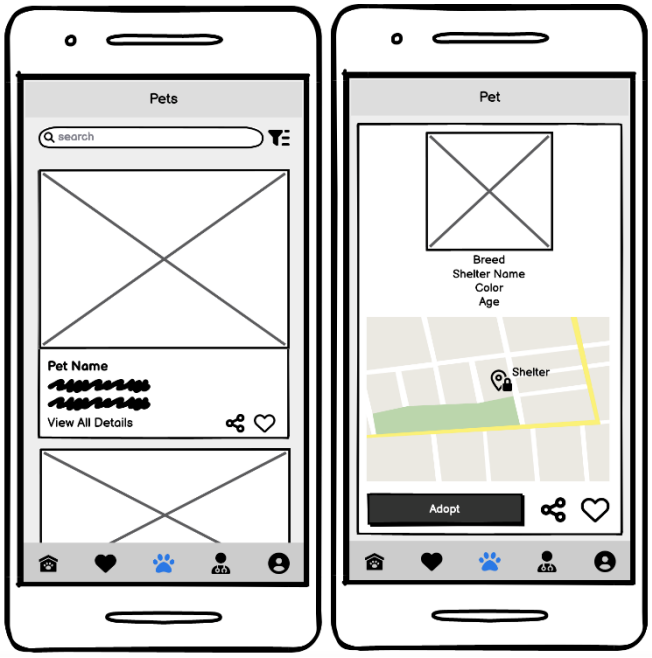

Figure 19 Wireframe to view all the information about the pet

17. Adopt a pet

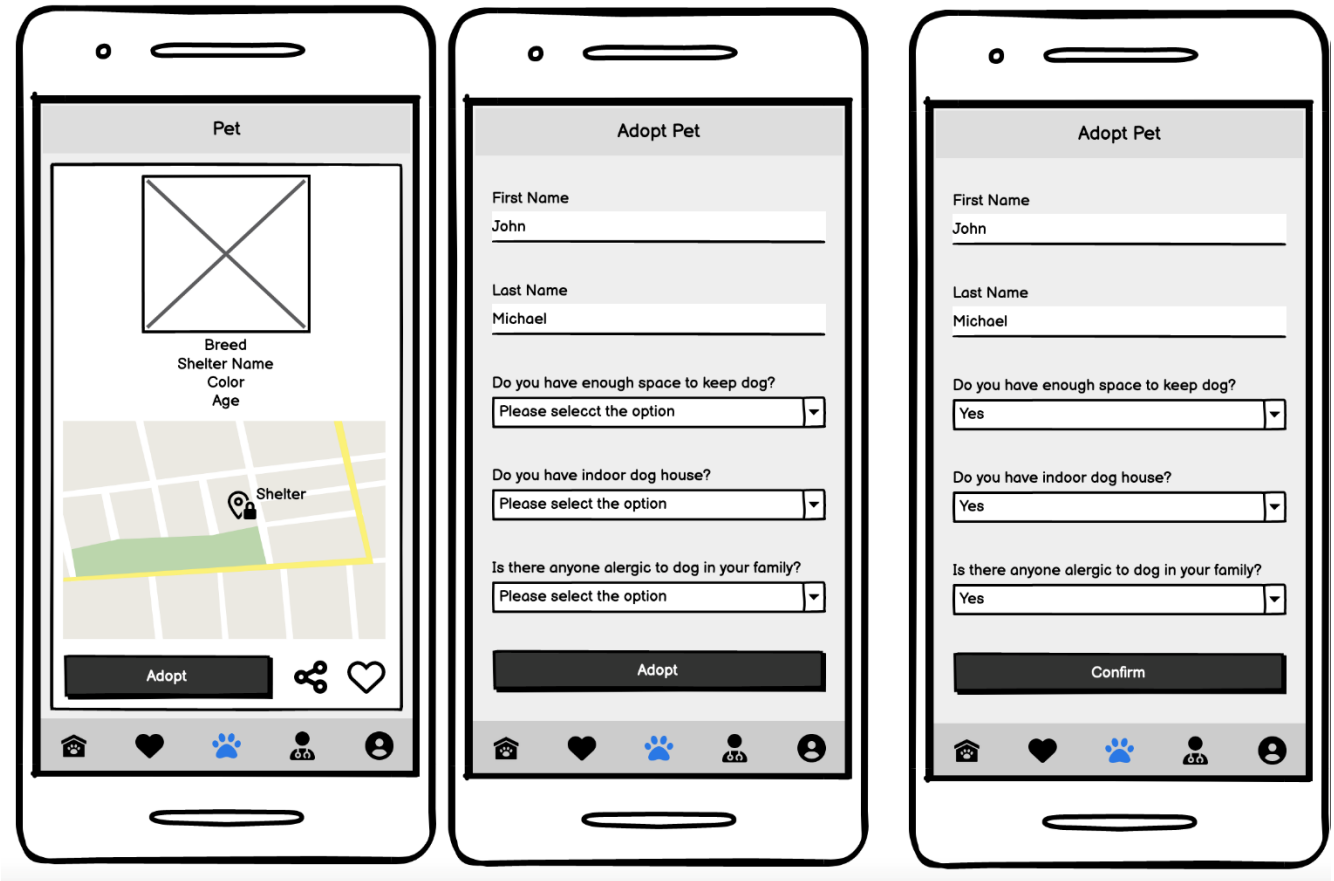

Figure 20 Wireframe to adopt a pet

18. User profile page

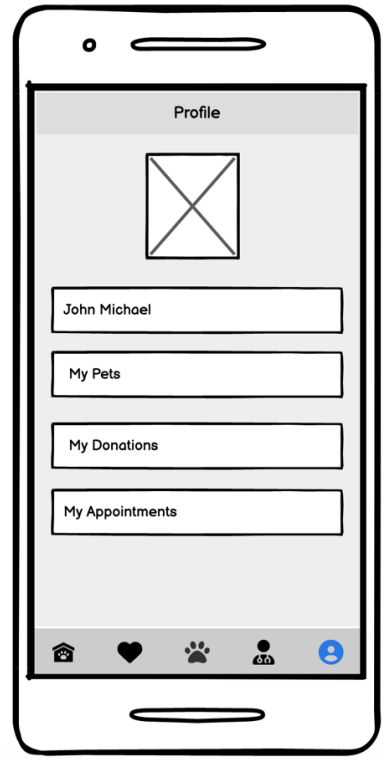

Figure 21 Wireframe for the user profile page

19. View my adopted pets

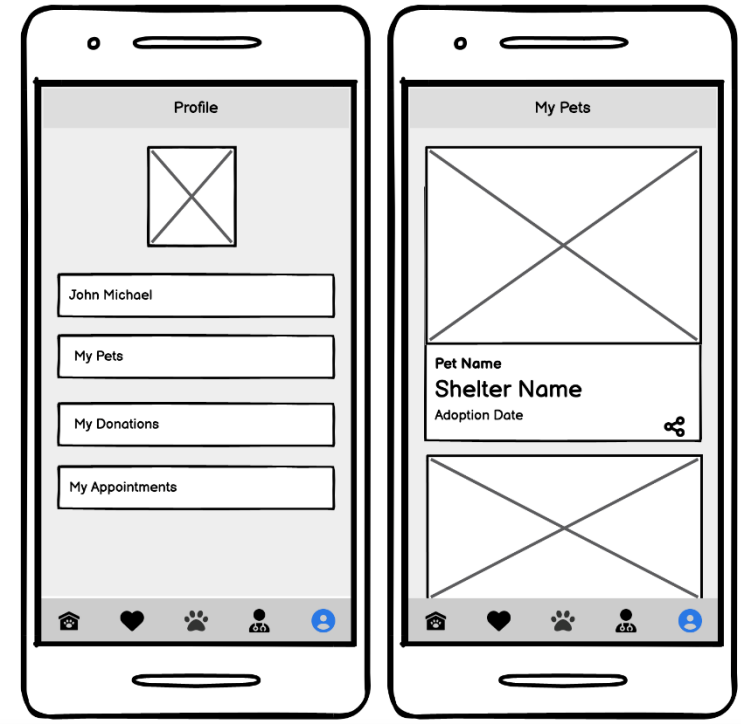

Figure 22 Wireframe to see the list of adopted pets by user

20. Donations made by the user

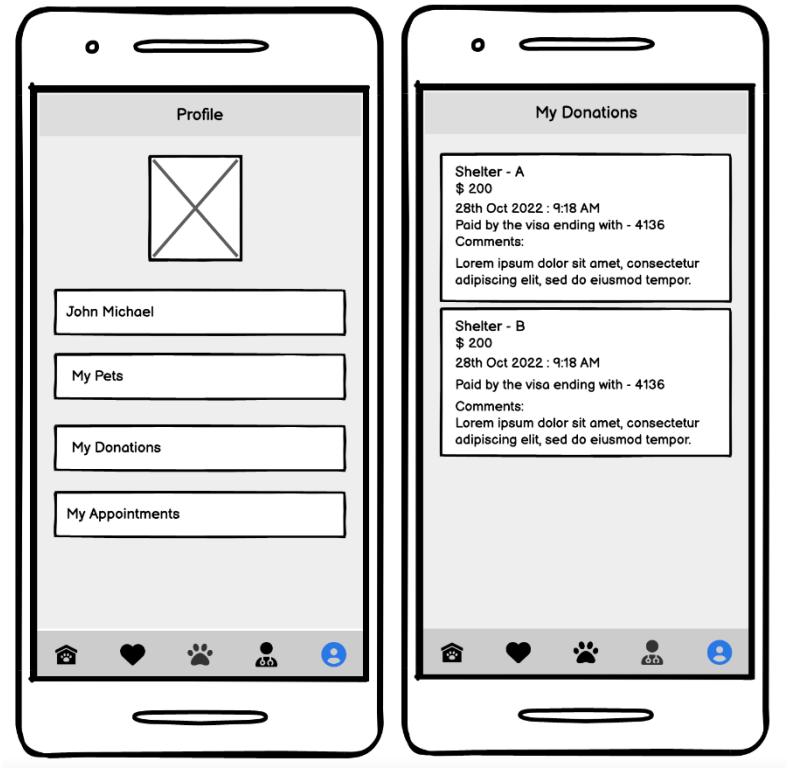

Figure 23 Wireframe to see list of donations made by the user

21. Appointments made by the user

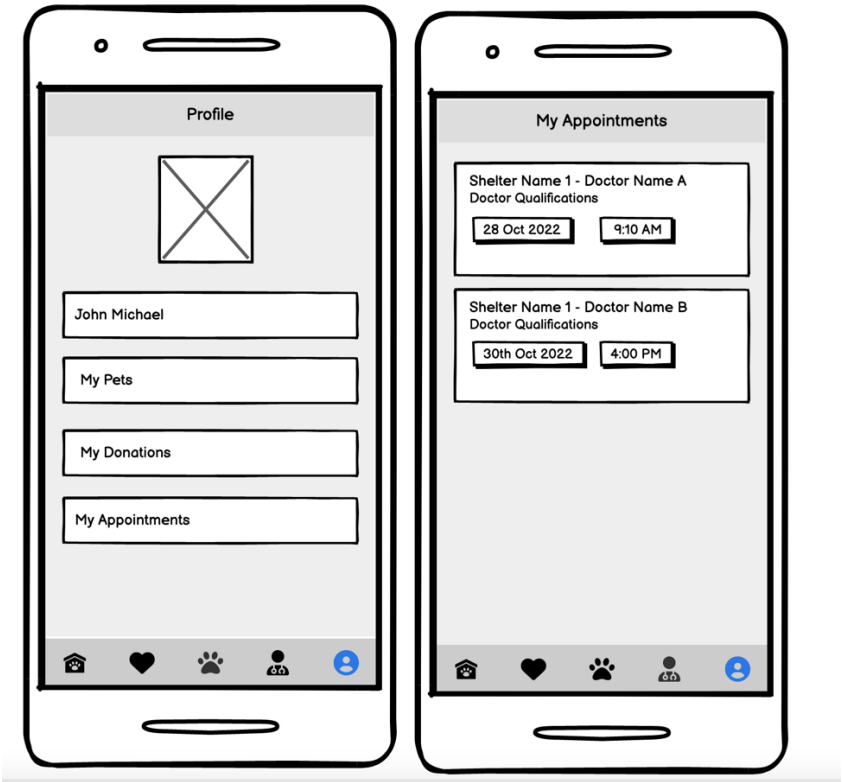

Figure 24 Wireframe to see list of appointments made by user

22. Filter Pet and Search bar for the filter section

Figure 25 Wireframe for the filter pet and search bar in filter section

6. Controller

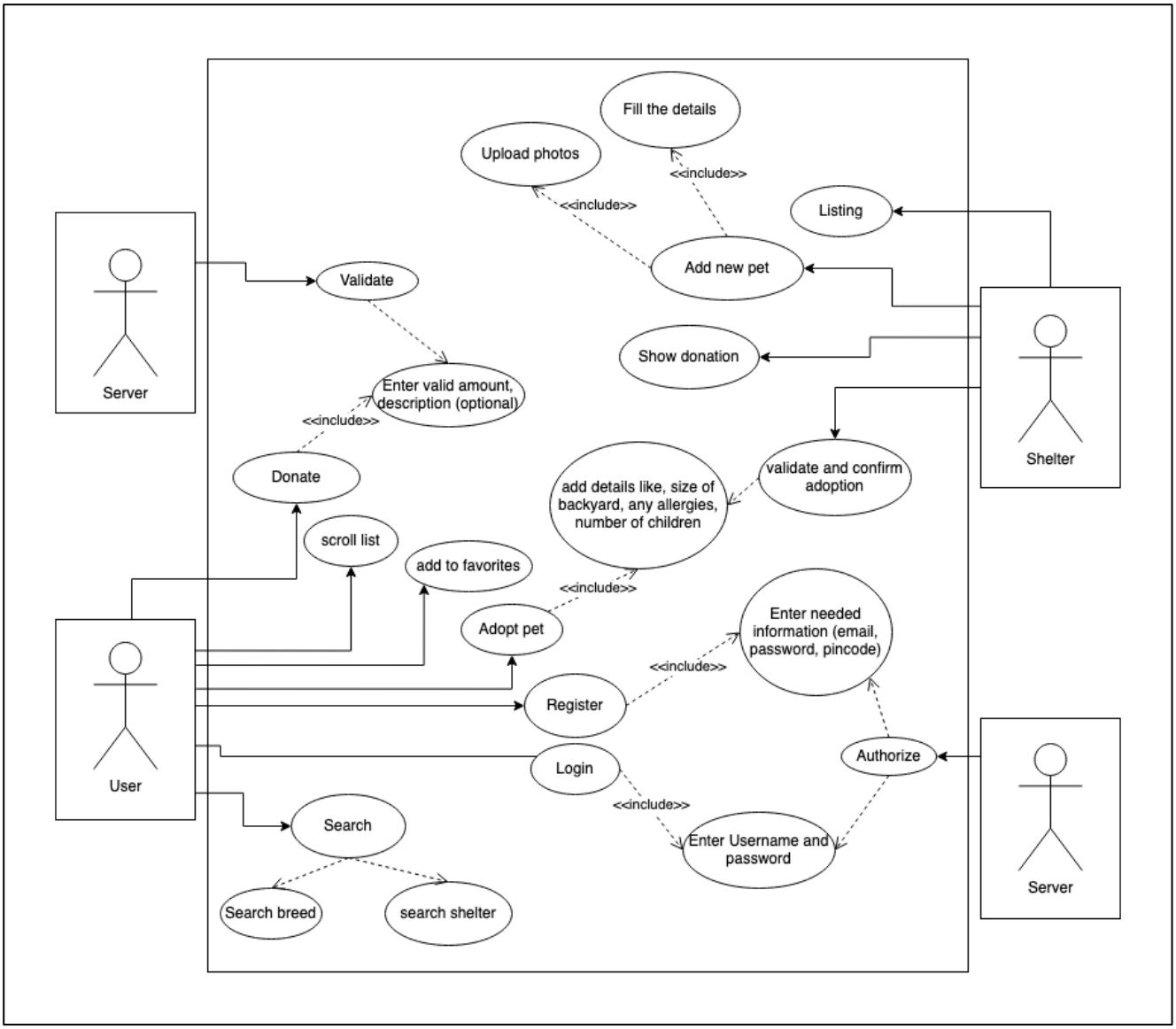

Figure 26 UML Diagram

7. Risk Analysis

We have designed a form to get user details during adoption and whether they are eligible for the adoption. In this form, we are not doing any background checks, so there is a possibility that the adopter may enter the wrong detail. The same can happen from the shelter side; they put incorrect information in the dog listing.

We will provide the donation feature, where users can select the shelter they want to donate to and enter the donation amount they want to make. Then they can pay the amount. But, after the payment, we have no plan to implement the feature of providing the tax exemption receipt to donors who can get the tax credit. Here, donators are donating money; hence they will expect to receive the receipt, which they can use during tax filing to get the credit.

There is one more development risk related to backend services and databases, which can affect the cost too. We have the feature of appointment booking, so there will be an entry in the database for every appointment. Hence, as time passes, the size of the appointment table will keep increasing and will increase data storage costs. As a solution, we will have to implement a function that runs at a specified period and deletes unnecessary entries from the database.

8. Timeline

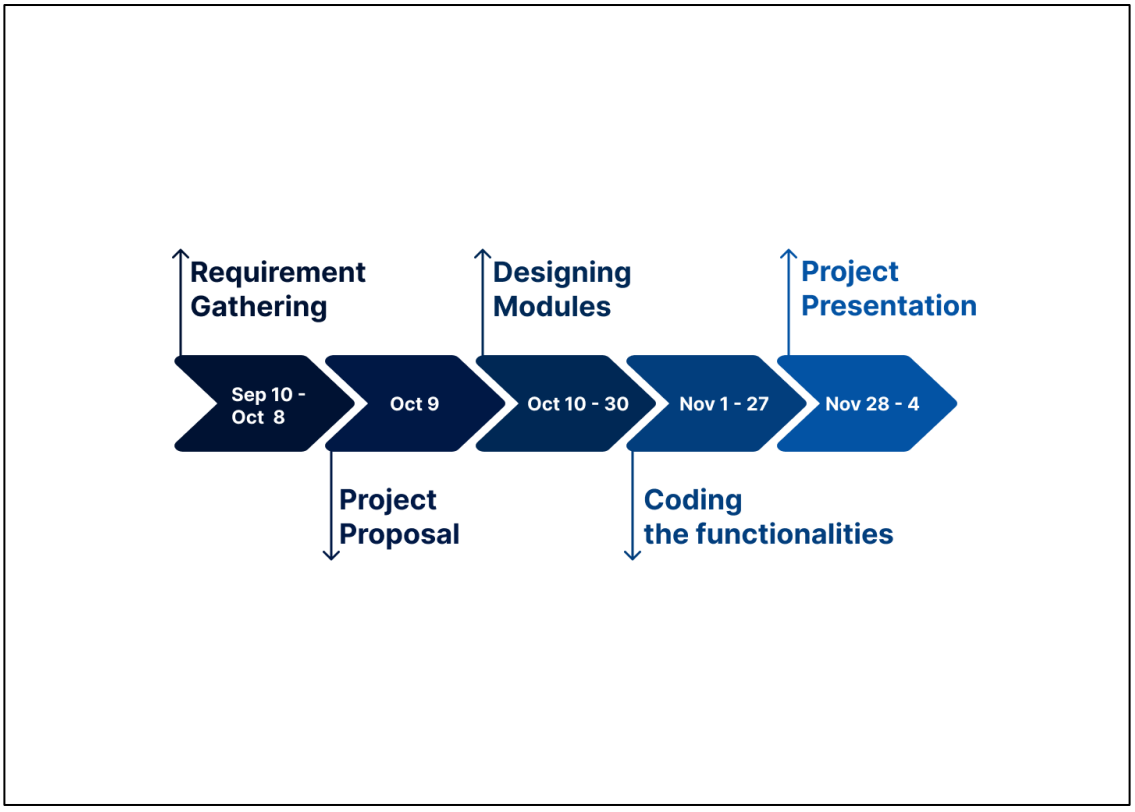

Figure 27 Estimated Timeline of the project completion [3]

9. Libraries

1. CardView – CardView is an Android Jetpack components library that helps in creating Material Design card with custom shadows, corners, and elevation [4]. CardView will help us in showing list items across the application.
2. RecyclerView – RecyclerView is an Android Jetpack library that allows efficient display of large datasets in lists [5]. We will use RecyclerView to show lists and grids in our application.
3. Glide – Glide is an open source image loading framework for Android that helps in media decoding, caching, and loading [6]. We will use Glide to show images in our application.
4. CameraX - CameraX is a Jetpack library for easy camera app development and integration [7]. We will use CameraX for clicking pet photos for uploading to create a new pet for adoption.
5. Material – Material is a library for Android Design to create Material Design components [8]. We will use Material library for UI development of the application.
6. Retrofit - A type-safe HTTP client for Android and Java that helps in consuming RESTful services [9]. We will use Retrofit for consuming external APIs.
7. Gson - Gson is a Java library that can be used to convert Java Objects into their JSON representation and vice versa [10]. We will use Gson for consuming JSON.

10. High-fidelity prototypes

1. View Pets list

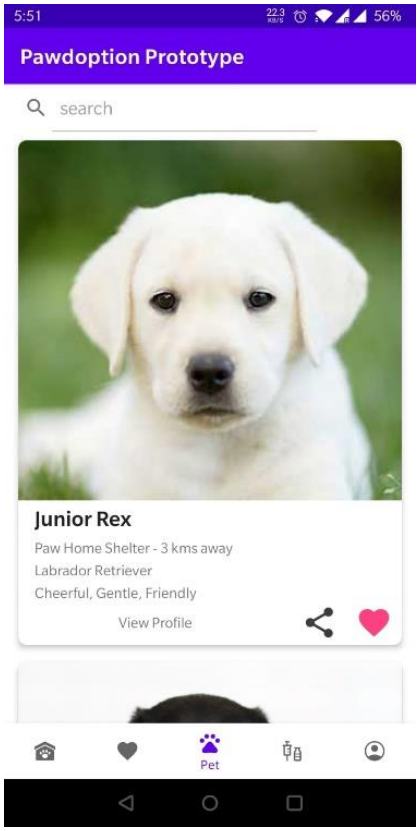

Figure 28 High-fidelity prototypes of View pet list screen created using Android Studio [11]

2. Share Pet

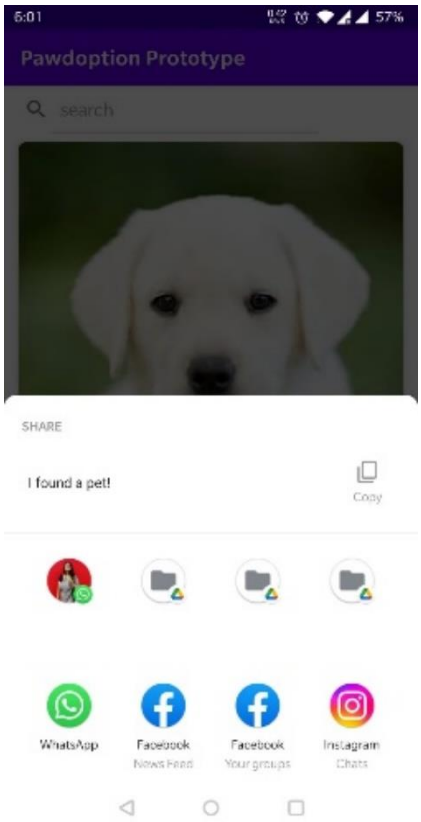

Figure 29 High-fidelity prototypes of share pet view created using Android Studio [11]

11. References

[1] "Create powerful business content together with xtensio," *Create powerful business content together with Xtensio*, 02-Oct-2022. [Online]. Available: https://xtensio.com/. [Accessed: 30-Oct-2022].

[2] "Security-first diagramming for teams.," *Diagram Software and Flowchart Maker*. [Online]. Available: https://www.diagrams.net/. [Accessed: 30-Oct-2022].

[3] "The Collaborative Interface Design Tool.," *Figma*. [Online]. Available: https://www.figma.com/. [Accessed: 30-Oct-2022].

[4] "Cardview : Android developers," *Android Developers*. [Online]. Available: https://developer.android.com/jetpack/androidx/releases/cardview. [Accessed: 30-Oct-2022].

[5] "Create dynamic lists with RecyclerView : android developers," *Android Developers*. [Online]. Available: https://developer.android.com/develop/ui/views/layout/recyclerview. [Accessed: 30-Oct-2022].

[6] Bumptech, "Bumptech/glide: An image loading and caching library for Android focused on smooth scrolling," *GitHub*. [Online]. Available: https://github.com/bumptech/glide. [Accessed: 30-Oct-2022].

[7] "Camerax Overview : android developers," *Android Developers*. [Online]. Available: https://developer.android.com/training/camerax. [Accessed: 30-Oct-2022].

[8] "Material design," *Material Design*. [Online]. Available: https://m3.material.io/. [Accessed: 30-Oct-2022].

[9] "Retrofit," *Square*. [Online]. Available: https://square.github.io/retrofit/. [Accessed: 30-Oct-2022].

[10] Google, "Google/gson: A java serialization/deserialization library to convert java objects into JSON and back," *GitHub*. [Online]. Available: https://github.com/google/gson. [Accessed: 30-Oct-2022].

[11] "Over 300+ dog breeds with pictures and prices," *Dogbreedslist.info*. [Online]. Available: https://www.dogbreedslist.info/. [Accessed: 30-Oct-2022].