



Al-Najah National University

Department of Computer Engineering

GRADUATION PROJECT II

HWBA PET



By:

MAJD HAMADNEH

MANAL KHALILI

SUPERVISED BY:

DR.HAYA SAMAANA

September 10, 2024

Abstract

TMy colleague and I wanted to create an app that contained all of these details in one location because of the recent surge in people's interest in pets, especially when we browse social media on a daily basis and see people asking for pet supplies, offering their pets for adoption, or inquiring about local veterinary clinics. One is to make sure that all of their needs are met and to assist these folks with their pets. The foundation of our idea is developing a mobile application for pet lovers and owners that offers them everything they need to raise their animals in one location, including supplies and services.

1. Easily ordered and purchased food and accessories for pet owners.
2. It will function as a forum where pet owners may communicate, post, and receive comments in order to assist one another in locating a lost animal.
3. Giving the user the names of veterinarians who work with veterinary clinics and assisting him in scheduling a visit to check on his pet's health.
4. Giving the user the names of cleaning, grooming, and other specialists connected to grooming clinics and assisting him in scheduling a visit to oversee his pet's care.
5. When the pet animal is on the road, make a reservation at a hotel for a certain number of days.

We noticed that there are a lot of applications in app stores for selling pet supplies and food, as well as for purchasing and adopting animals. Nevertheless, we were unable to locate any comparable applications in Palestine, nor any that offered all of these features at once. This project consists of a website for the administrator and a mobile application for users. The website may be connected to a business that provides services to pet owners. Numerous frameworks and tools were used in the project's construction.

I went with Spring Boot for the backend because of how easily and effectively it handles Java-based applications. I used Postman to test the API endpoints' functionality because it made it easy for me to validate requests and responses.

The database was managed by myself using XAMPP, a local server solution that is usually used in conjunction with PHP. In order to bind Java objects to database tables and carry out CRUD activities with ease, I created JPA (Java Persistence API).

I used RESTful APIs to connect my front-end Dart (Flutter) application to the back-end, enabling seamless communication between the mobile application and the server. This configuration made it possible for the Flutter application to dynamically retrieve and update data from the backend.

Table of Contents

1	Introduction	1
1.1	Problem	1
1.2	Objective	1
1.3	Scope of the Work	2
1.4	Importance	2
1.5	Report Organization	2
2	Constraints and Earlier work	3
2.1	Constraints and limitations	3
2.2	Standards	4
2.2.1	MVC (Model View Controller)	4
2.3	Earlier coursework	5
3	Literature review	6
4	Methodology	9
4.1	Tools, Programming Languages, APIs Technologies	9
4.2	client Side:	9
4.3	Website Side for admin:	11
4.4	Server Side:	11
4.5	IDEs and Code Editors	14
4.6	Database Structure	14
4.6.1	Database - Relational:	15
4.7	Features of The Application	15
4.8	Implementation	17
4.8.1	Mobile Application	17
4.8.2	Website:	32
4.9	Database and API Server	37
5	Conclusion and Results	43
6	Conclusion	44
7	References	46

Chapter 1

Introduction

1.1 Problem

People in Palestine are becoming more interested in and loving having pets that bring positive energy into their homes. However, one issue facing those considering pet ownership is the exploitation of these animals through inflated pricing, particularly when a new cat is born and its offspring are offered for prices that are deemed high for a Palestinian citizen. With this initiative, we have primarily attempted to address this issue.

There aren't many organizations or institutions that assist homeless pets on the streets, and nobody to turn to for advice on caring for or rearing animals, or on what supplies and attire the animal requires, or on dog food and cat.

1.2 Objective

Our primary objective in this project is to stop the exploitation of those who desire pet ownership by offering them the chance to adopt a dog or cat for free. The likelihood of getting a pet and experiencing something new rises with adoption. Since the pet's care, protection, food preservation, and prevention of vagrancy are of utmost importance, we will also assist the pet owner in providing food, clothing,

and other necessities for their companion through the project's buy area.

1.3 Scope of the Work

Our application primarily focuses on animals and things associated with them; users can showcase pets for adoption and take use of the services the program offers to other users.

1.4 Importance

The primary purpose of this program is to aid homeless animals by offering them for adoption. It also helps pet owners take care of their animals and locate everything they require in one convenient location, making it simple for them to access necessary services.

1.5 Report Organization

The structure of this report is as follows: Chapter 1 provides an overview of our project, while Chapter 2 provides an introduction and a summary of its significance and extent. Chapters 3 and 4 discuss the challenges and limitations we encountered while working on this project, while Chapter 5 details the approach we took to finish it. We went into great length about the ML model, the languages and tools that were used to develop it, and how we applied it in our application in Chapter 6. Chapter 7 presents the findings and a debate, while Chapter 8 concludes with a suggestion.

Chapter 2

Constraints and Earlier work

2.1 Constraints and limitations

Since the beginning of this project, we have faced numerous challenges and difficulties.

1. **Time constraint:** It was difficult because we were unfamiliar with some of the programming languages and frameworks we used. As this was our first time developing a full mobile application, we had to become familiar with new mobile frameworks like Flutter and technologies like Dart. Although the syntax is close to C, which made adaptation easier, mastering the backend framework proved to be a significant obstacle. We had to familiarize ourselves with the backend tool, Spring Boot. We used XAMPP for local server control and JPA (Java Persistence API) to communicate with the database. Because we had other responsibilities, it took more time and effort to understand and use these tools appropriately while working under pressure. because we had other things that needed our focus, each with their own projects. To get the app closer to our goal, we had to properly manage our time and put in more effort.
2. **Stable internet connection:** In order to communicate with the

Spring Boot backend and function properly, our application needs a steady online connection. By using RESTful APIs, this made it possible for the mobile app to interact with the backend and dynamically fetch or change data.

2.2 Standards

2.2.1 MVC (Model View Controller)

The shape in our system was displayed using the architectural pattern. To make tracking the flow easier, we can split the project into three components. These components are listed in the following order:

Model: This is a representation of the database we utilized, which was accessed through JPA (Java Persistence API) and locally maintained with XAMPP. To guarantee smooth data management and CRUD processes, the model responds to requests from the view and the controller and updates itself on a regular basis.

View: This is the mobile app's graphical user interface (GUI), created with Flutter to give users and administrators an easy-to-use interface for displaying and modifying data.

Controller: This is the backend server, built with Spring Boot, that controls communication between the model and the view and performs application logic. The controller responds to requests and updates data as necessary to guarantee effective coordination and data flow between the frontend and the backend.

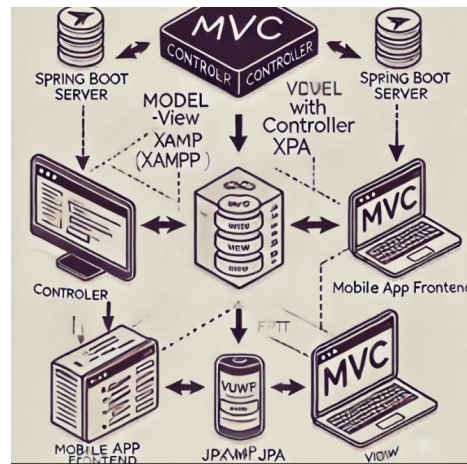


Figure 2.1: mvc

2.3 Earlier coursework

We studied computer engineering and took many courses covering topics like website creation, programming languages, and techniques. The basis for developing front-end and back-end applications and interacting with real databases was laid by taking courses in web programming, object-oriented programming, and databases. The Software Engineering course was also very helpful in helping us write our code in a way that adheres to advanced problem-solving and critical thinking concepts.

These classes accelerated the development process and made a substantial contribution to our project. We utilized Dart for the frontend, and because of its close resemblance to Java, the switch was not too difficult. We used Postman for API testing and Spring Boot, a Java-based framework, for the backend, , in addition to JPA, which controls database interactions via XAMPP. In order to fulfill the project's particular requirements, we added online tutorials in Flutter, Dart, and Spring Boot to our curriculum to further hone our skills.

Chapter 3

Literature review

There is a great deal of research on the relationship between pets and mental health in humans, and there are many different research topics that can be explored in this area. Possible study subjects include previously unexplored or little-studied subjects as well as revisiting previously studied subjects (Ryan & Ziebland, 2015). Humans and pets have relationships that span generations, social classes, nationalities, and cultures. It is an unforced, spontaneous human action. It is imperative that policy makers across the globe take into account specific evidence, especially in cases when evident human suffering is evident. There is a solid foundation for policy makers who are interested in improving mental health to reject the idea that dogs are just fluffy, furry, and insignificant. Individuals that are interested in using dogs to improve mental health may frequently only Instead of thinking about implementing costly or intricate new initiatives, we should identify obstacles and work to remove them (Pawson et al., 2005).

Starting with the value of pets for mental health, we will discuss the use of artificial intelligence and computer-based pet tracking and care applications, which are among the most important uses of realistic activities. Prior research has demonstrated that certain technology and applications can assist humans in providing for the well-being of their pets, such as virtual pets (Arnaldo et al., 2021). The pet water

dispenser, which can measure how much water a pet needs each day, and the food feeder, which is housed in a food storage container, were both equipped with Arduino systems (Chen & Elshakankiri, 2020). Additionally, use a pet litter box to gauge the amount of pet feces (Sangvanloy & Sookhanaphibarn, 2020; Wang, 2020). A smartphone app was created in the Visitors to the zoo can connect a QR tag to an animal cage to receive text or voice-activated information about the animal in three distinct languages: English, Arabic, and French. In addition, the app has the ability to play animal sounds in response to the QR code that is linked. Visitors may also purchase pets at the zoo and rate the app (Kadhim & Al-Qaraawi, 2020). (Tang et al., 2005) set up an additional application that employed a GPS sensor to follow the pet's movements throughout the house.

Another study uses specially made pet jackets connected to micro-controllers to detect physical touch in human-animal interaction systems with poultry pets, such as chickens, ducks, and so forth (Lee et al., 2006). Furthermore, a remote-activated smart pet door with a feces pad, food, and water dispenser was built and a mobile application for pet monitoring was proposed in the Philippines (Luayon et al., 2019). The pet door is intended for usage with cats and dogs. The creation of an animal adoption platform was the focus of another research project, which aims to improve public-animal welfare organization communication efficiency (Dsouza et al., 2022). The public can simply turn in stray animals or their own pets using this website. Additionally, this platform can be accessed for

the general public to adopt animals or pets from any group included on the created virtual platform (Dsouza et. Al., 2022). Additionally, good-hearted folks will have the opportunity to try something for poor animals. Individuals will have the opportunity to tell others about their experiences rescuing dogs, inspiring others to attempt the same (Dsouza et al., 2022). Another mobile application for tracking and

caring for pets was created and refined over several iterations by Shah et al. (2021). This app has a lot of features, like an integrated chat system for users to upload pictures of pets they wish to adopt, a pet food store with reasonably priced pet food, and a veterinarian section where users can get in touch with veterinarians for assistance with any medical issues pertaining to their pets. The following were the main goals this application addressed: The number of stray animals put to death will decline, people will become aware of the needy animals in their community, and compassionate individuals will have the chance to assist these creatures be able to share their stories of how they rescued pets and motivate others to do so (Shah et. Al., 2021).

Chapter 4

Methodology

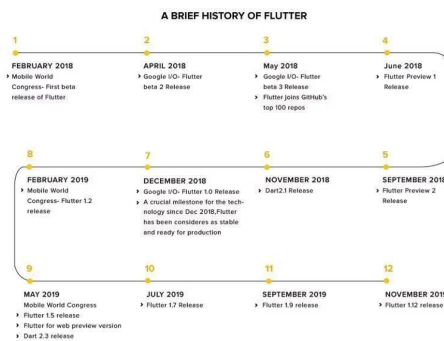
4.1 Tools, Programming Languages, APIs Technologies

We have a plethora of alternatives when it comes to developing applications, but after extensive study and reading, we also need to consider the essential and indispensable components of developing websites and mobile applications, like frameworks, programming languages, and tools. We will address all of these specifics and go into deeper detail. We choose to construct them using the subsequent framework.

4.2 client Side:

1. **Design:** In order for the user to grasp the application and easily navigate between the interfaces, we had to create a basic, user-friendly, and straightforward program. We also made the decision to design the basic application in a color that is visually pleasing. Before we had a clear vision of our application, we looked at a lot of designs and applications, including ones on Pinterest, Google Play, and the App Store. We then developed this design to our specifications and began developing the user interfaces.
2. **Frameworks:** To construct our application, we opt for the Flutter framework, which With only one code base, you can create

desktop (Linux, Mac, and Windows), web, and mobile (iOS and Android) applications with Flutter, an open-source user interface toolkit from Google. An application written in this language can provide users with a sophisticated, detailed, and engaging experience. You write the code once and run it everywhere, according to the single code rule! In this manner, you can simultaneously schedule and aim for the release of your software on all platforms. Time and money can be saved by launching the application onto the market as soon as possible.



Top 10 Motivators for Flutter App Development Teams to Consider:

- (a) Simple to learn.
- (b) Excellent Performance.
- (c) Reusable Devices .
- (d) Quick time to market.
- (e) Quick reload.
- (f) Perfect for creating an MVP.
- (g) Capability to produce excellent design.
- (h) Reduced development expenses.
- (i) Excellent community for documentation.

(j) Excellent work.

3. **Programming languages:** Flutter makes use of Google created the programming language Dart with online and mobile application developers in mind. It is intended for the development of iOS, Android, and web applications. Being able to function on all sophisticated web browsers, mobile devices, and even web servers is one of the language's objectives. It's clear that Dart will be used because the flutter framework will be utilized. Programming languages like Java, C, Swift, and Kotlin are very comparable to Dart. We observed that Dart had many similarities with Java, the most widely used language we know, which makes learning a new language simpler and less challenging.

4.3 Website Side for admin:

1. **Design:**First, we made a website that was only accessible by the administrator. On this website, they could add, remove, and see things like food and clothing, as well as conduct other functions like adding categories and viewing bookings for veterinary clinics.
2. **Programming languages:** Because it is lightweight, user-friendly, and enables the creation of natively developed desktop, web, and mobile applications from a single codebase, we chose to utilize the robust UI toolkit Flutter. The real-time hot reload feature of Flutter's Dart programming language increases development productivity and performance.

4.4 Server Side:

1. **Frameworks:**For the website and mobile app, we used the free and open-source Spring Boot framework on the server side. A Java-based framework called Spring Boot makes it easier to cre-

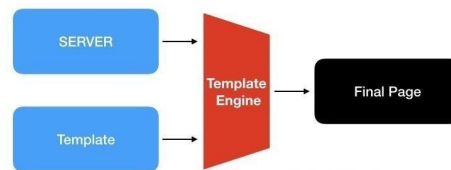
ate production-quality, standalone Spring apps. It enables the development of scalable, effective server-side applications and is intended to get you up and running with the least amount of configuration. Spring Boot has strong security and dependency management, supports RESTful APIs, and works effectively with a variety of databases. **This framework's advantages are:**

- (a) **Productivity:** :Quick development using pre-configured templates and embedded servers.
- (b) **Simplicity:**For developers who are familiar with Java, it is simple to learn and adapt.
- (c) **Speed:** Quick processing and starting of the application.
- (d) **Multi-platform:** Capable of being implemented on multiple platforms, such as cloud and containerized settings.
- (e) **Maintainability:**a well-organized code base with sustained scaling support.



We use the Spring Boot framework, which is a Java-based backend framework known for its simplicity, power, and efficiency. Spring Boot facilitates the creation of RESTful APIs, enabling smooth

communication between the server and client. It aids in the management of server routes and makes the development of web applications efficient and scalable. Additionally, it supports rendering dynamic content through APIs, which can be consumed by various frontend applications. We chose Spring Boot for its reliability, speed, and the wide range of tools it provides for Java-based applications.



plus the Firebase Database for conversation. A cloud-hosted database is known as a real-time database. Every linked client receives real-time data synchronization in JSON format. Using the JavaScript SDK and the Apple and Android platforms, you can construct cross-platform apps that share a real-time database instance and push updates to all of your clients automatically, therefore it's suitable for conversation.

- 2. Programming languages:** We utilized Java as the main language and the Spring Boot framework for the backend. Java is a flexible language that facilitates the development of scalable, effective back-end applications. Java is used to conduct server-side logic and API development using Spring Boot. We used Dart with Flutter for the front-end, which enables the development of stunning natively compiled mobile applications. When combined, Java and Dart offer a potent mix for effectively constructing full-stack apps.

4.5 IDEs and Code Editors

HWBA PET: Microsoft's Visual Studio Code Editor, which is cross-platform, free, offers rapid editing, and supports a multitude of programming languages, is what we used to design our application. Several built-in tools are included in it, such as a web designer, database schema designer, class designer, and support for Git.

1. **Version Control:** Since we were a team, we needed a mechanism to share work and integrate code. To this end, we set up two repositories on GitHub, one for the front end and one for the back end, and we also utilized Google Drive on occasion.
2. **API Endpoints Testing:** We need to test every endpoint before integrating our application with the client because the back-end server of our application includes a RESTful API. We made use of Postman for this. An excellent tool for analyzing RESTful APIs is Postman. We may use PUT, PATCH, DELETE, and several more request types with Postman. This utility contains almost all of the features that a developer may require.

4.6 Database Structure

The database is crucial in most applications, as it enables the reading, editing, and deletion of information, which adds significant value and ensures data backup. Our application is heavily reliant on the database for displaying and managing information. We used a relational database because it efficiently handles the relationships between different tables and integrates well with our API, providing the necessary structure and consistency for our data management needs.

4.6.1 Database - Relational:

Because relational databases offer strong support for managing structured data with established relationships, we chose them for our schema. Relational databases employ tables to store data in rows and columns, guaranteeing data integrity through associations and constraints, in contrast to NoSQL databases, which use flexible schemas and store data in documents that resemble JSON. Because of MySQL's robust capability for SQL queries and interoperability with Java-based applications, we used it in our project under XAMPP management. We efficiently carried out CRUD operations by mapping Java objects to database tables using JPA (Java Persistence API). With this configuration, data is arranged into tables with predefined schemas and intricate queries and relationships are handled via SQL, providing an orderly approach to data administration. The relational database methodology offers a reliable and dependable method of managing and querying data, which qualifies it for use in applications with intricate data linkages and structures.

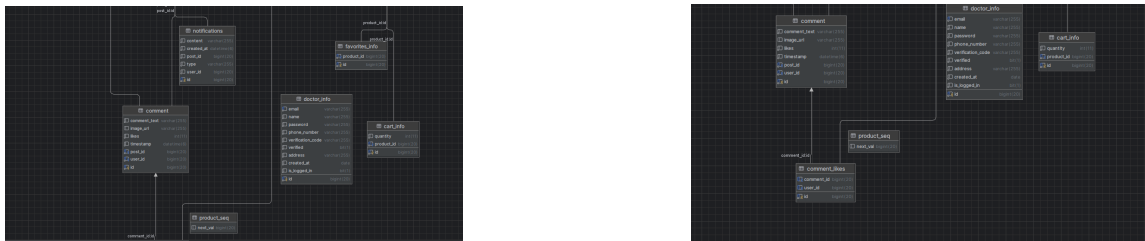


Figure 4.1: data base

4.7 Features of The Application

1. **Adoption**By making a post and engaging with other users directly, the user can adopt a dog, cat, or any other animal.
2. **Shop**This feature allows products that cater to the needs of animals, like food, toys, accessories, and other things, to be offered.

The user can view these products in the application by using the RESTful API. These products are shown with the option for the site administrator to add new products or categories.

3. **Veterinary Centers** Using this service, customers may schedule pet appointments by viewing a list of veterinarians who are linked with veterinary facilities. Users will be able to plan appointments and see which doctors are available. To avoid repeated reservations, the time slot will no longer be available once an appointment is set.

4. **Grooming Centers** With the use of this function, users may schedule appointments for a range of pet care treatments by viewing a list of experts connected to pet cosmetic clinics. To avoid repeated reservations, the time slot will no longer be available once an appointment is set.

All appointments are shown on a schedule that is accessible to the moderator. The moderator can examine and manage all reservations using this interface; they don't need to be accepted or rejected. The time window will close to other reservations once a reservation is confirmed. The comprehensive overview of all reservations offered by the moderator's dashboard guarantees effective management of the appointment scheduling procedure.

5. **Social Network** By downloading a post and seeking advice on how to care for and interact with pets, as well as a post to look for a lost pet, where a conversation occurs between the owner of the pet and the finder, we have created a platform for social communication among pet owners and made it easier for the owner to locate their animal companion. The chat's backend is powered by the Firebase database, while the front end is powered by Flutter.

6. **Hotel** With this function, the user can designate two dates to

reserve a spot for his dog or cat to be cared for in the event that he has to travel or is extremely busy. The user will see these dates in the application, and the website administrator will see the reservation dates.

4.8 Implementation

In this section, we'll dive into the details of each part of the system. All the tools, methods and libraries used are discussed in detail.

4.8.1 Mobile Application

1. **Welcome Screens:** These panels alert the user to the available features before he logs in or signs up; he sees the login/sign up screens after tapping "continue". Welcome screens contain liquid slider with images represents the main services in our app.

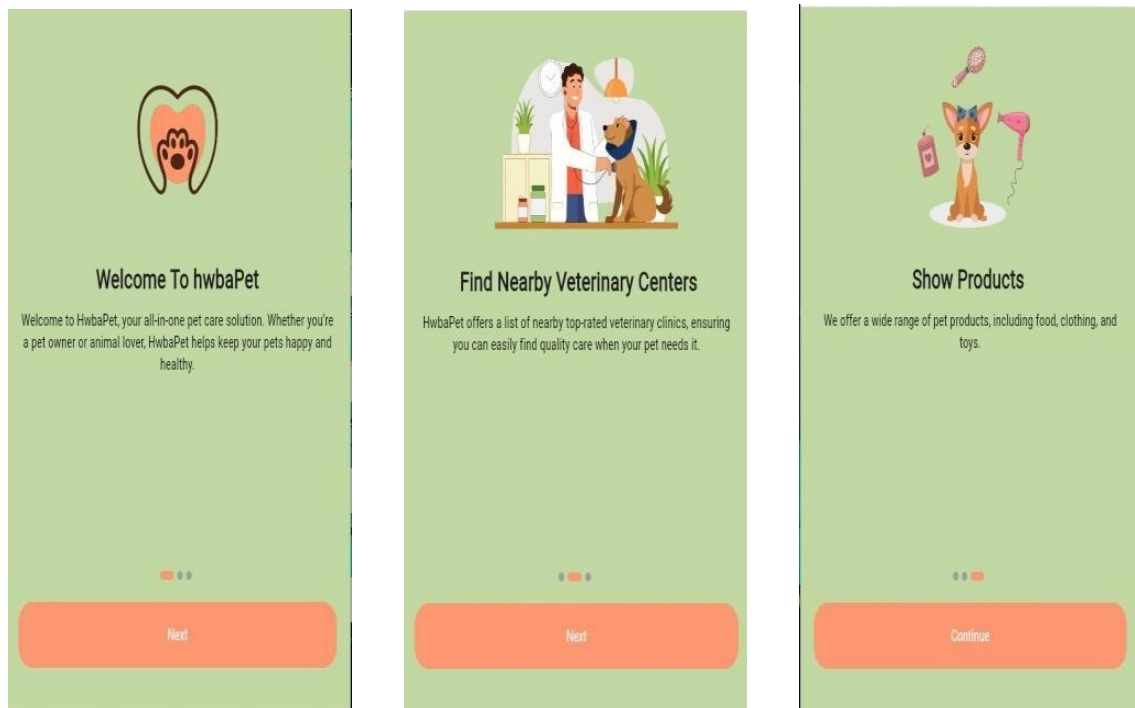


Figure 4.2: OnBording

2. **SignUp Screen:** If a user does not have an account, they can create a new one by providing the necessary information. The user is required to enter a valid email address so that the application can send a verification message to confirm their identity. Once verified, all their data will be stored in the MySQL database.

The sign up as doctor is different from the sign up as user and different from the sign up as hotel registration because the information is different.

The sign-up screen contains four text fields for the user's name, email, phone number, and password. When the user presses the sign-up button, their account will be created based on the user model in both Firebase and the MySQL database.

The account will only be created if the following conditions are met:

- (a) All text fields are filled.
- (b) The email is not already registered by another user and valid because There is a verification code to rely on.
- (c) The password is at least 6 characters long and contains both uppercase and lowercase letters.

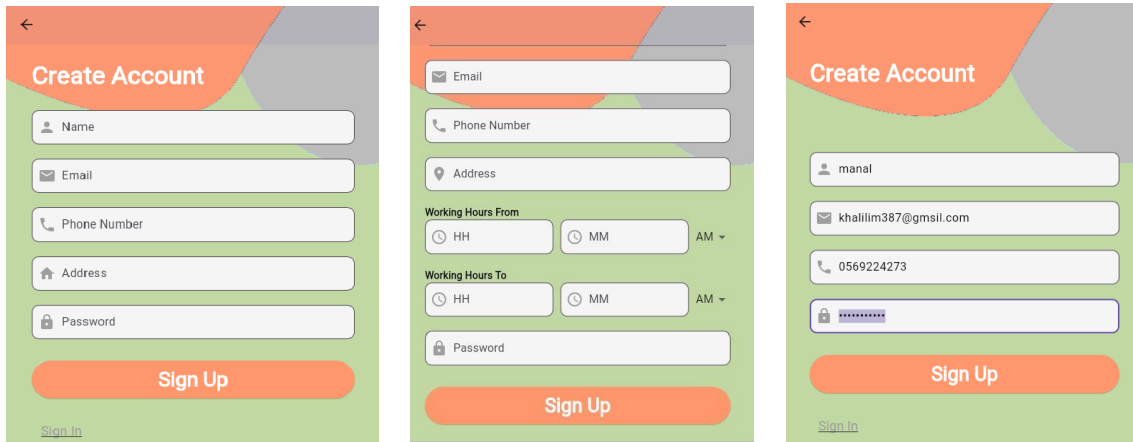


Figure 4.3: sign up

- Email verification Screen:** After clicking on sign up, the email will be verified. To confirm the registration process, verification will be done through email verification as shown:

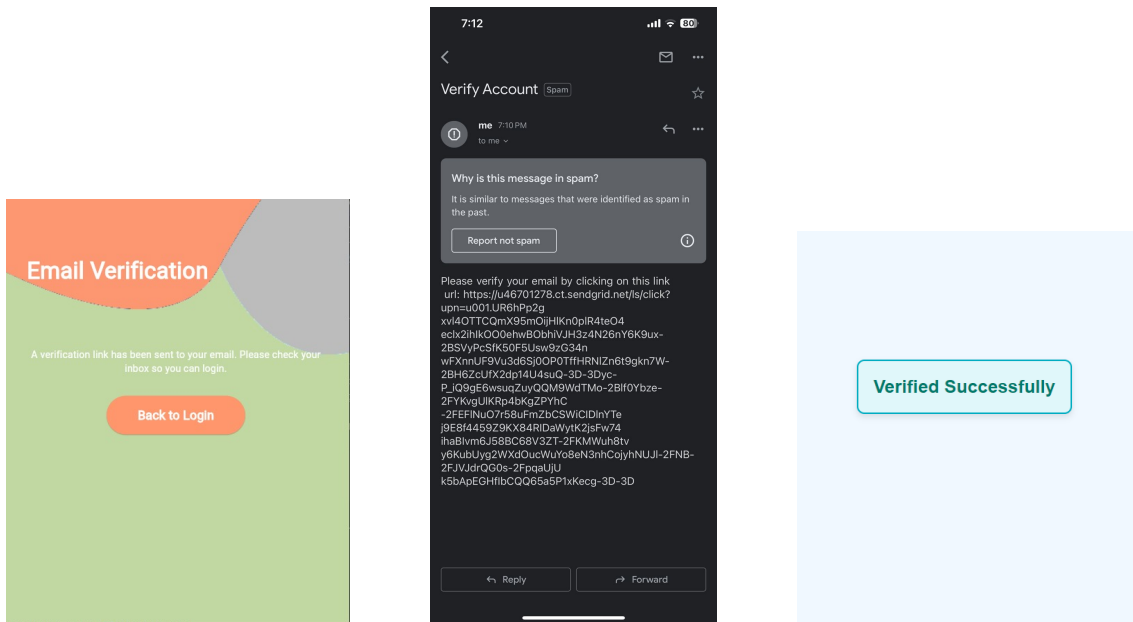


Figure 4.4: Email verification

4. **Forgot Password Screen:**In order to reset their password, users must enter a code that was sent to their registered email address as identification. This method improves security by restricting the ability to reset the password to the legitimate account owner. When changing passwords, this kind of email verification is frequently utilized for identity authentication, as the example below illustrates.

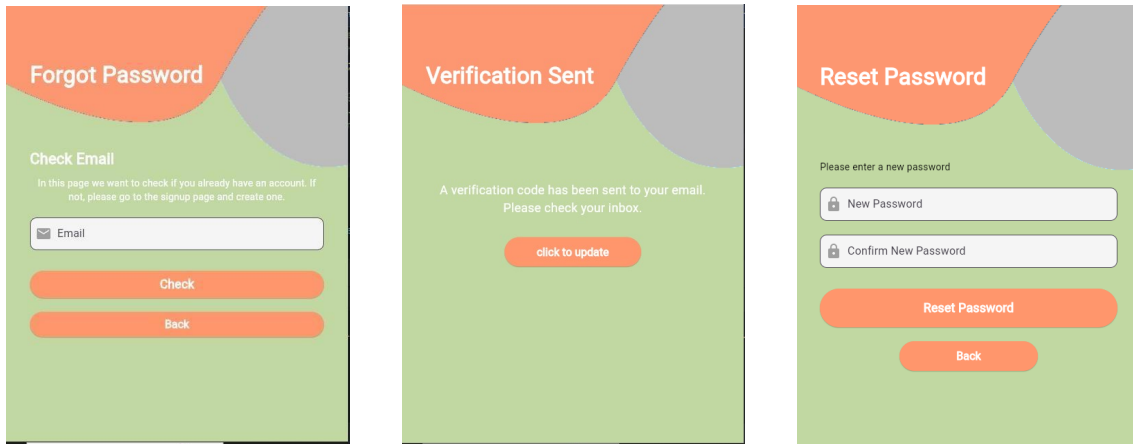


Figure 4.5: Forgot password

5. **Login Screen:**A user needs to have an account in order to log in. If not, they can make one by going to the registration page. The user might try to log in by entering their email address and password. The application will display a warning message if any required fields are left unfilled or if the information is inaccurate. Two text fields one for the user's email address and another for their password are present on the login screen. There's also a sign-in button that takes you to the home screen, but it only works when certain prerequisites are satisfied.
- (a) The two text fields are filled up.
 - (b) TBoth Firebase and DB have properly verified the email ad-

dress and password.

- (c) The password is the same as the one linked to the given email address.

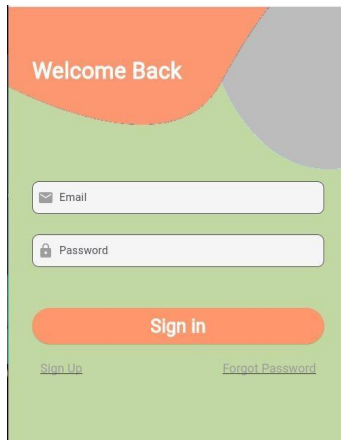


Figure 4.6: login

- 6. **Home Screen:** Home screen contains components represents all the services in the app as following: Slider: contains 4 slides represents all main pages, when slide tapped it will navigate to the corresponding page:

- (a) Slide 1: welcome Slide.
- (b) Slide 2: goes to Shop page.
- (c) Slide 3: goes to veterinary centers page.
- (d) Slide 4: goes to grooming centers page..

Button Row: contains two buttons one to go to grooming centers page and the other goes to hotel booking page. **Button bar(nab bar):** contains 5 options:

- (a) Home Screen.
- (b) Timeline Screen.

- (c) Shopping Cart Screen.
- (d) All Pets Screen. centers page.
- (e) Profile Screen.

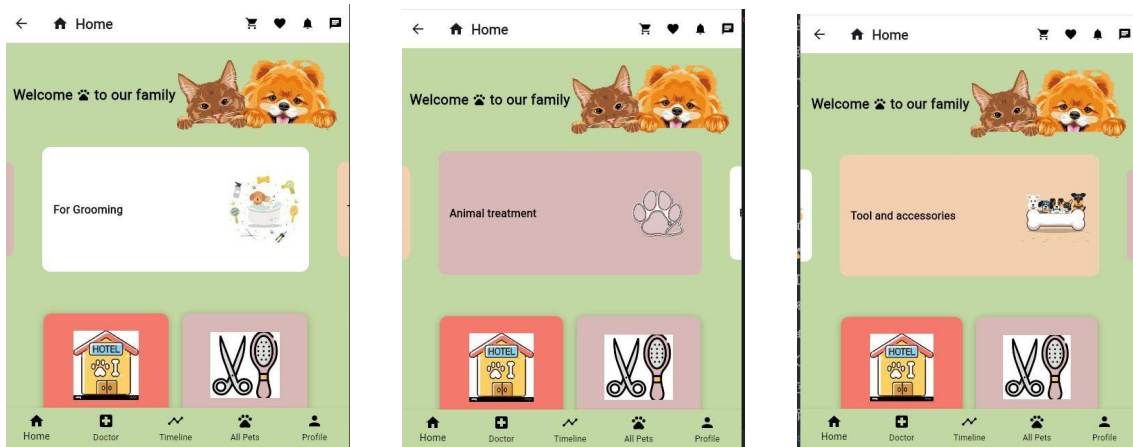


Figure 4.7: Home screen

7. **Shop Screen:** The shop screen displays every product as a card, allowing it to be sorted by category and to display every product. Additionally, a user can search for any product using its name, category, price, or rating. Any product can be deleted from the

cart or favorite icon, or added to the shopping cart or favorite items list for later review.

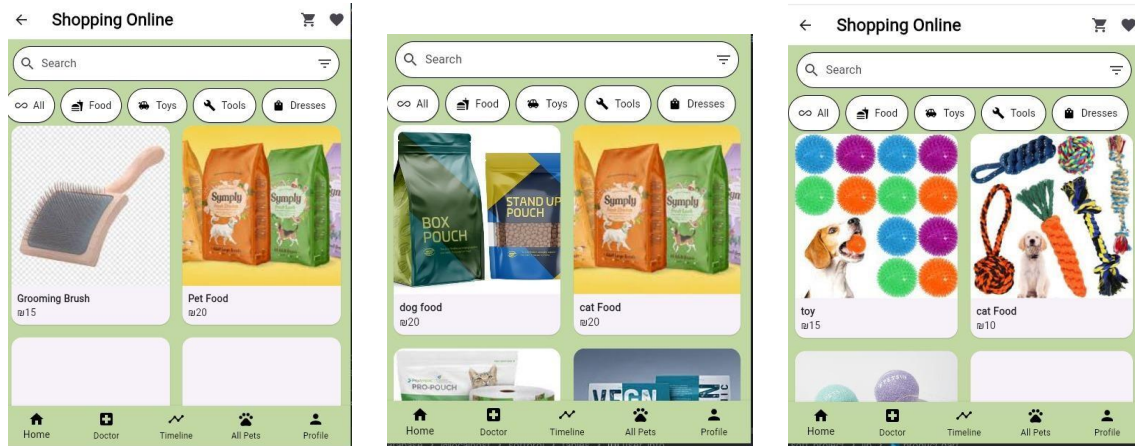


Figure 4.8: Shopping online

- 8. Shopping cart Screen:** The consumer can simply add any item to his shopping cart on this page and proceed to the checkout. The trolley will be empty when it starts. The product will then be added to the cart when we use the Product Details Screen and click the "add to cart" button. The total price of every item in the cart is determined and displayed on the cart page. We have

the option to add, remove, or alter the number of products in our cart.

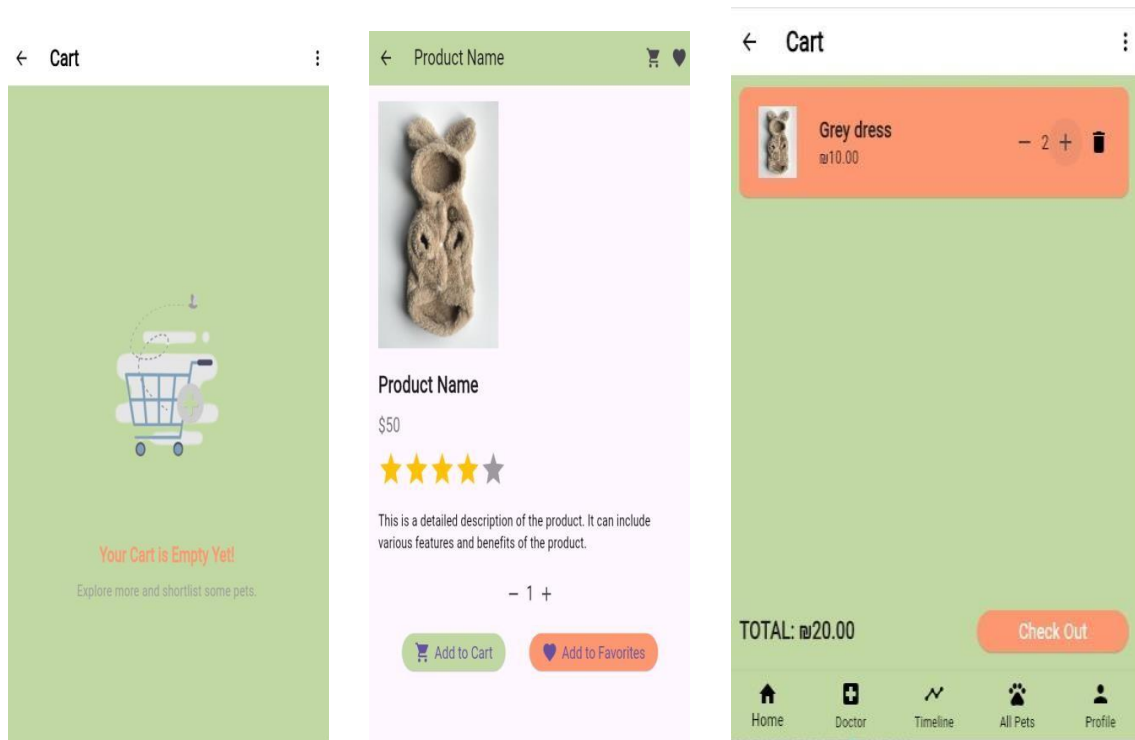


Figure 4.9: Shopping cart

9. **Place order Screen:** In order to finish the purchase procedure, the user must indicate here the payment type (cash or card) and the delivery mode (hand receipt or delivery to the lesson he already

added), as indicated below:

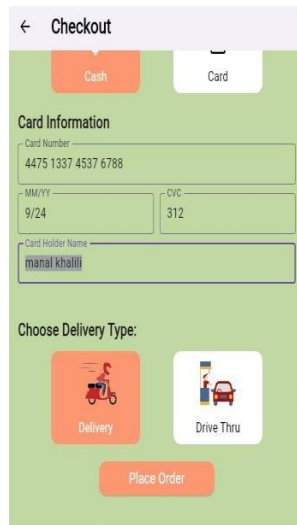


Figure 4.10: place order

10. **Add address Screen:** Within the app, the user can add delivery addresses manually by selecting their location and filling in for the location. When the user clicks the button, the address will be saved in their delivery address list.

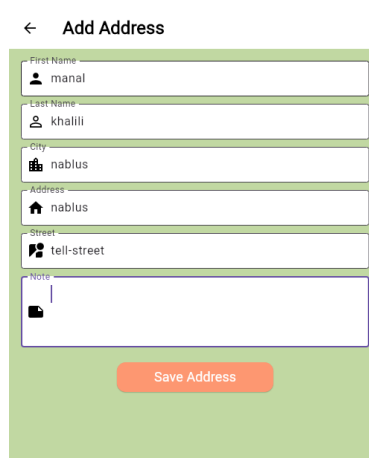


Figure 4.11: Add address

11. **Pending order Screen:**The order number will display within the user's card, and all orders that he has not yet received will appear here. The cost of the goods, the cost of delivery, the order's status, and the delivery method are all factors. Unless he receives the initial indication that the order is being created, as indicated below, the user has the option to remove the order.

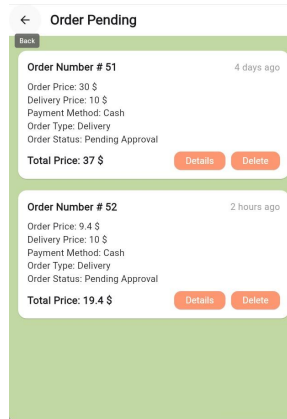


Figure 4.12: pending order

12. **Archive order Screen:**From me to pending orders The user has received the request and can now rate it, as indicated below, which is the difference.

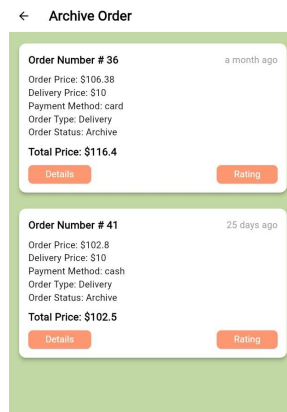


Figure 4.13: Archive order

13. **Order Details Screen:**The invoice for that order, which includes the products that were ordered, their number and price, and the total cost including delivery, will show up after clicking the details button on the previous page.

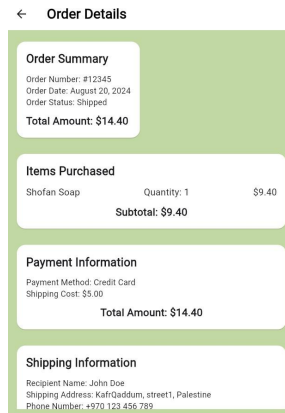


Figure 4.14: Order details

14. **Veterinary Centers Screen:**The primary interface is an image slider and a list of veterinary clinics displayed as vertically scrolling cards.

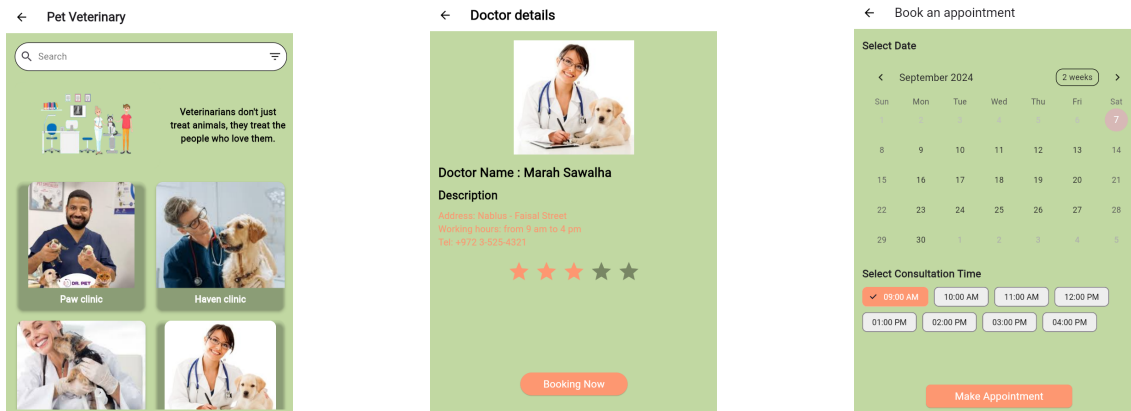


Figure 4.15: Veterinary Centers

selecting a center and clicking on it, we are taken to the information screen where we may schedule a visit to assess the health of our pet.

15. **Grooming Centers Screen:** Similar to veterinary clinics, the grooming center main page has a list of the centers as cards. You may select a center and click on its card to view its details, schedule an appointment, and read the comprehensive information.

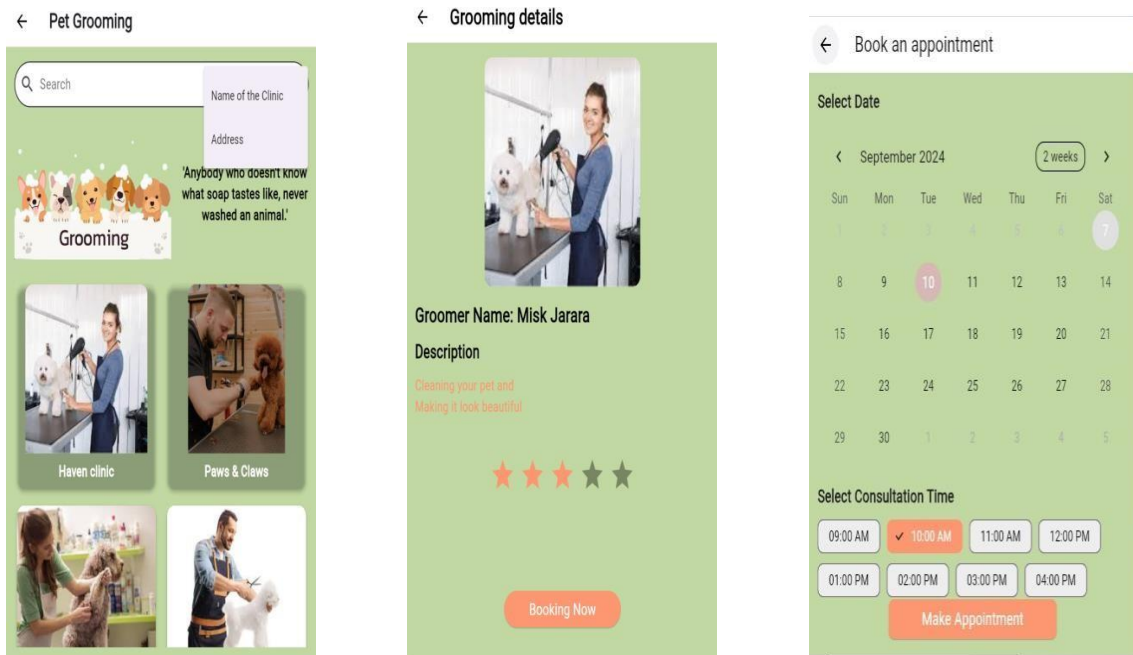


Figure 4.16: Grooming Centers

16. **Hotel Screen:** The "Calendar" button on the hotel's main screen must be pressed in order to bring up the calendar and select the two dates (beginning and ending) that indicate the days on which

your pet is permitted to stay.

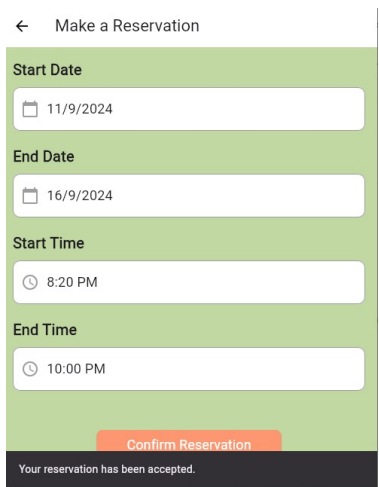
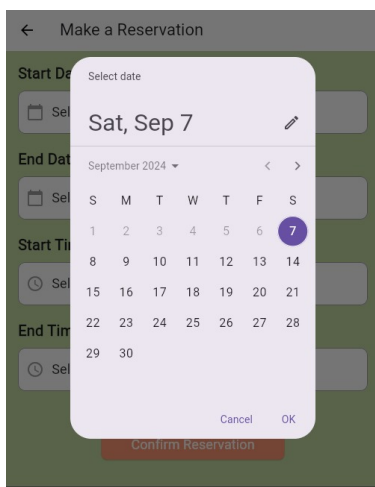
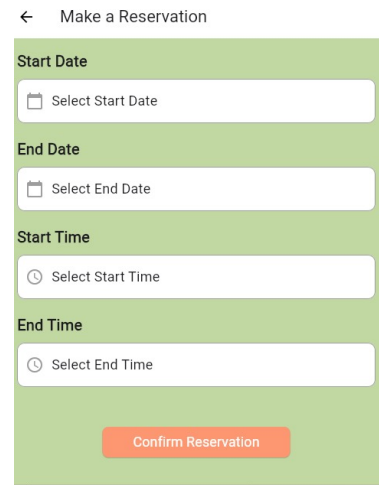
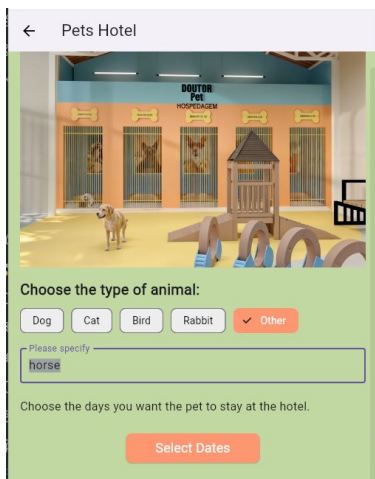


Figure 4.18: Hotel

17. **Social Network(Timeline):**displays all user posts shown as cards on a vertical scroll bar, with like and commenting options. By clicking on the post header, you can also access the post owner's profile and have a chat with them and then You can upload your own post by clicking the + icon at the top,We take the image from

gallery and the location will be detected using Geolocator.

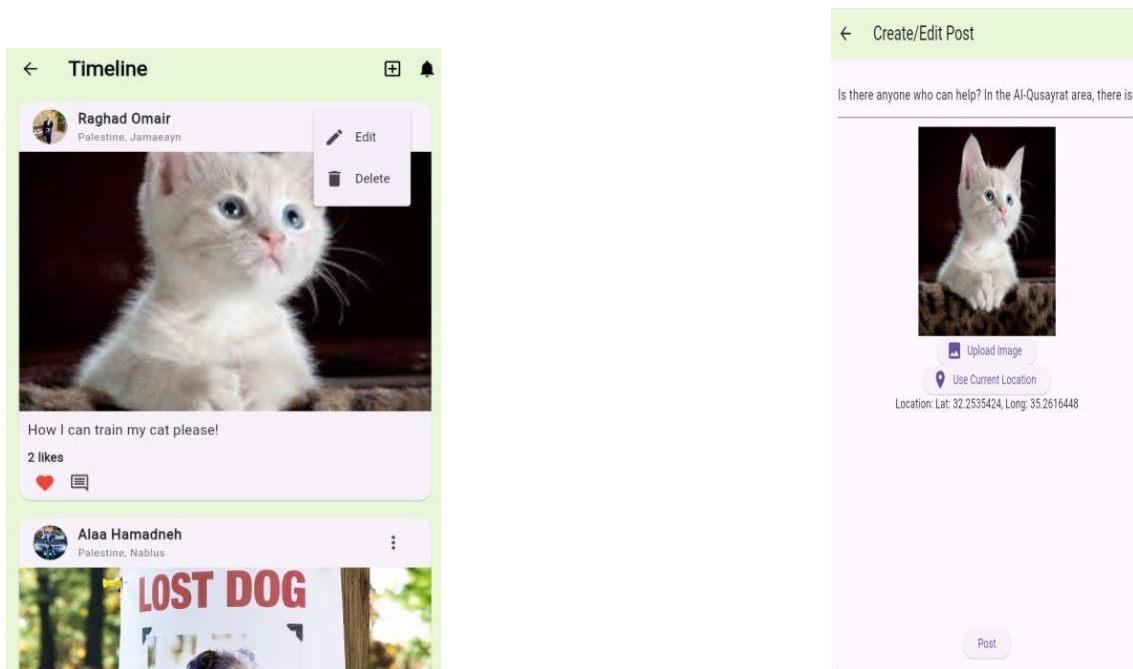


Figure 4.19: Time line

When you press on like or comment notification it will open the corresponding post and if you clicked on message icon it will open the chat screen.



Figure 4.20: Time line

18. Chat Screen:

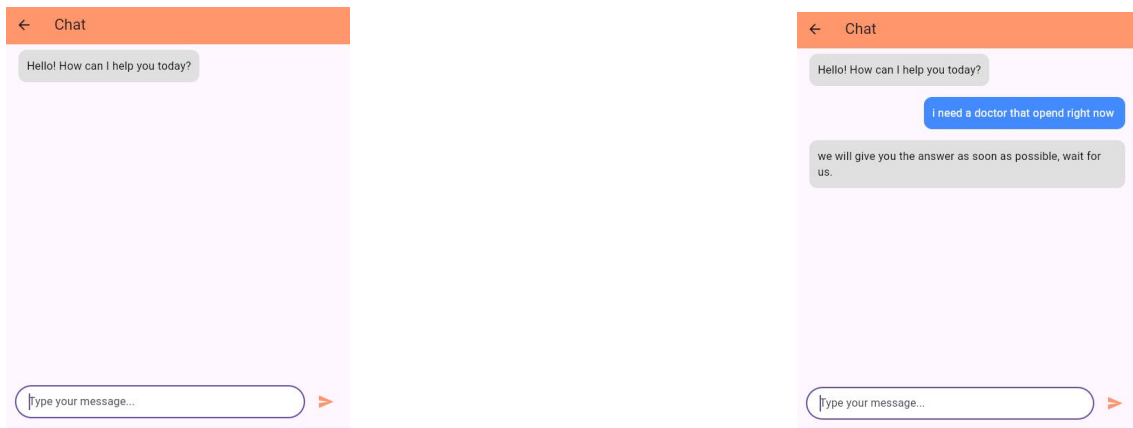


Figure 4.21: Chat

19. **Profile Screen:** In this screen we have user name and avatar and the posts that he shared, if the profile is open from profile's owner, he will see edit profile button but if the profile is opened from other users will see chat icon which navigate to the chat screen, we put the image before. Every doctor and every groomer has his own profile through which he can edit any information and view his dashboard.

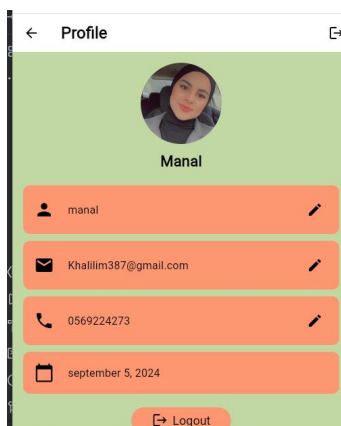


Figure 4.22: Profile

4.8.2 Website:

1. **Statistics:**In dashboard we access the data in four ways:
2. Count of new accounts per month or day or year.
3. Count of veterinary appointments per month or day or year.
4. Count of hotel bookings per month or day or year.
5. product sales profit per month or day or year. **Charts:**We have two charts:
6. First one for new account distribution over years from 2023-2024.
7. Second chart is about Veterinary appointments distribution per month.

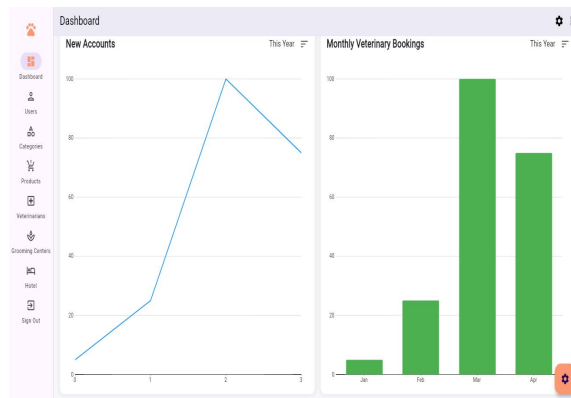
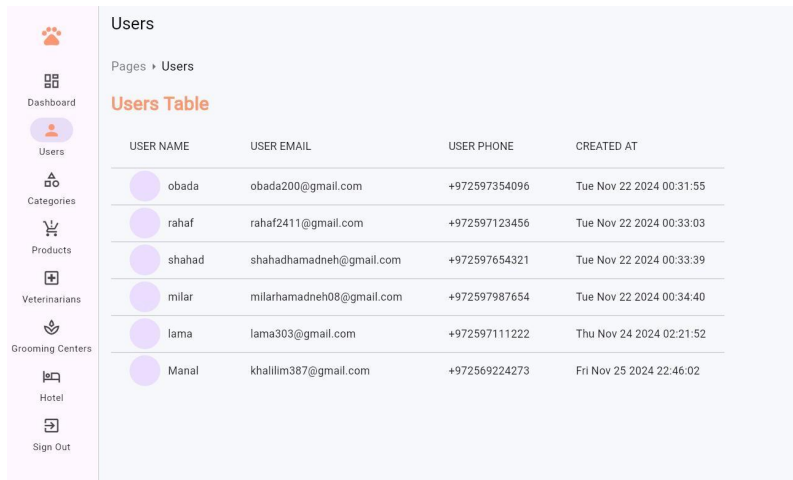


Figure 4.23: dashboard

8. **Tables:**We displayed a table including all of the pets in the database along with their names, images, and types. All of the significant database tables were shown on these screens, along with add, modify, and delete buttons.

Users Table:Table of users here is used for monitoring and statistics.

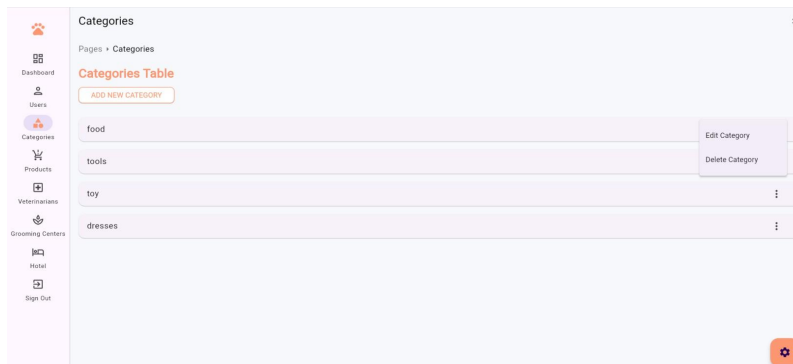


The screenshot shows a dashboard with a sidebar on the left containing navigation icons for Dashboard, Users, Categories, Products, Veterinarians, Grooming Centers, Hotel, and Sign Out. The main content area is titled 'Users' and contains a 'Users Table' with the following data:

USER NAME	USER EMAIL	USER PHONE	CREATED AT
obada	obada200@gmail.com	+972597354096	Tue Nov 22 2024 00:31:55
rahaf	rahaf2411@gmail.com	+972597123456	Tue Nov 22 2024 00:33:03
shahad	shahadhamadne@gmail.com	+972597654321	Tue Nov 22 2024 00:33:39
milar	milarhamadne08@gmail.com	+972597987654	Tue Nov 22 2024 00:34:40
lama	lama303@gmail.com	+972597111222	Thu Nov 24 2024 02:21:52
Manal	khalilim387@gmail.com	+972569224273	Fri Nov 25 2024 22:46:02

Figure 4.24: User table

9. Categories Table:



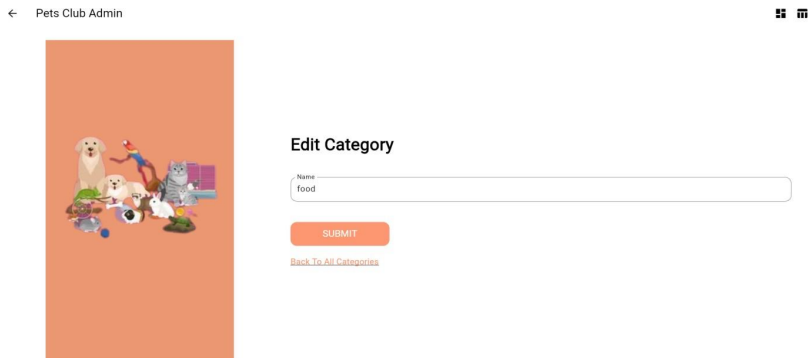
The screenshot shows a dashboard with a sidebar on the left containing navigation icons for Dashboard, Users, Categories, Products, Veterinarians, Grooming Centers, Hotel, and Sign Out. The main content area is titled 'Categories' and contains a 'Categories Table' with the following data:

Category Name	Actions
food	Edit Category
tools	Delete Category
toy	⋮
dresses	⋮

Figure 4.25: category

Edit Category:edit category name.

Delete Category:delete the category from database.



← Pets Club Admin

Edit Category

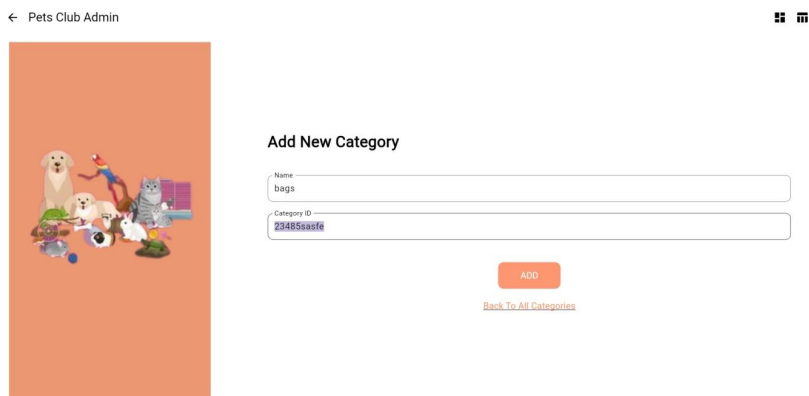
Name
food

SUBMIT

[Back To All Categories](#)

Figure 4.26: Edit category

when i click add new category



← Pets Club Admin

Add New Category

Name
bags

Category ID
23485aaafe

ADD

[Back To All Categories](#)

Figure 4.27: Add category

10. **Products:**Shows all product details like image, name, price, quantity, ... For this table we can perform add and delete actions.

PRODUCT NAME	PRODUCT ID	PRICE	QUANTITY	CATEGORY	DESCRIPTION	ACTIONS
Cutting nails	638282a735f69b10a4836393	5	1	clothes	Pet nail cleaning	⋮
Activity brain	638282a735f69b10a4836394	40	1	toys	Cat Activity Brain Mover	⋮
Delicious Food	638282a735f69b10a4836395	15	10	food	High-quality pet food	⋮
Stylish Dress	638282a735f69b10a4836396	25	5	dress	Beautiful dress for pets	⋮
Grooming Kit	638282a735f69b10a4836397	30	3	tools	Complete grooming kit for pets	⋮
Chew Toy	638282a735f69b10a4836398	10	20	toys	Durable chew toy for pets	⋮
Pet Carrier	638282a735f69b10a4836399	50	2	clothes	Comfortable pet carrier	⋮
Pet Bed	638282a735f69b10a4836400	60	4	clothes	Cozy pet bed for sleeping	⋮
Water Fountain	638282a735f69b10a4836401	35	6	tools	Automatic water fountain for pets	⋮
Pet Leash	638282a735f69b10a4836402	20	15	clothes	Strong and durable pet leash	⋮

Figure 4.28: Product

Add product:From the page we can add new product to the database.

← Pets Club Admin

Add New Product

Product ID:

Product Name:

Product Price:

Product Quantity: + -

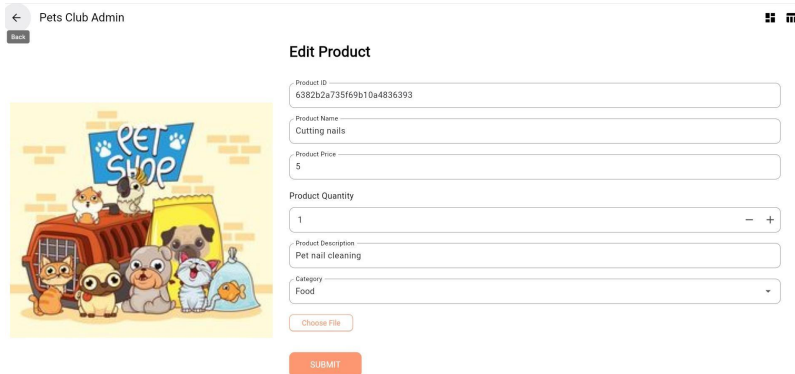
Product Description:

Category:

Please choose a picture of the product:

Figure 4.29: Add product

Edit product: I can edit any information related to the product



Edit Product

Product ID: 6382b2a735f69b10a4836393

Product Name: Cutting nails

Product Price: 5

Product Quantity: 1

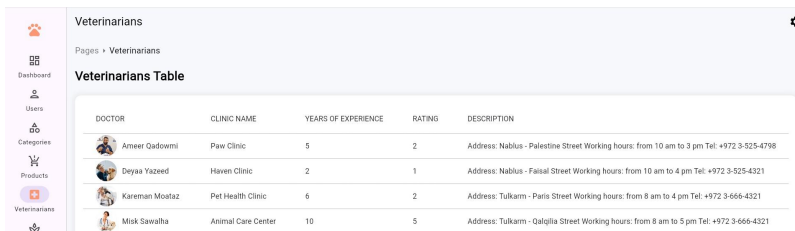
Product Description: Pet nail cleaning

Category: Food

SUBMIT

Figure 4.30: Edit product

- Veterinaries:** Here we display the list of veterinaries in the database and the details of it like doctor name, clinic name, experience, rating and description.







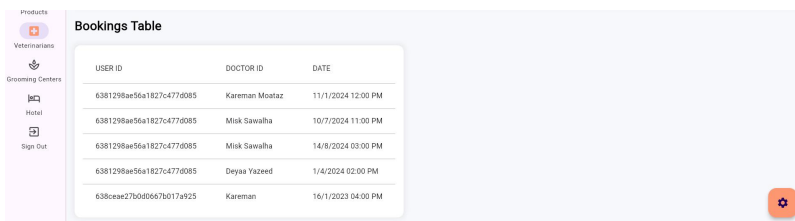
DOCTOR	CLINIC NAME	YEARS OF EXPERIENCE	RATING	DESCRIPTION
 Ameer Qadowmi	Paw Clinic	5	2	Address: Nablus - Palestine Street Working hours: from 10 am to 3 pm Tel: +972 3-525-4798
 Deyaa Yazeed	Haven Clinic	2	1	Address: Nablus - Faisal Street Working hours: from 10 am to 4 pm Tel: +972 3-525-4321
 Kareman Moataz	Pet Health Clinic	6	2	Address: Tulkarm - Paris Street Working hours: from 8 am to 4 pm Tel: +972 3-666-4321
 Misk Sawalha	Animal Care Center	10	5	Address: Tulkarm - Galgolia Street Working hours: from 8 am to 5 pm Tel: +972 3-666-4321

Figure 4.31: Veterinaries table

Booking Table:



USER ID	DOCTOR ID	DATE
6381298ae56a1827c477d085	Kareman Moataz	11/1/2024 12:00 PM
6381298ae56a1827c477d085	Misk Sawalha	10/7/2024 11:00 PM
6381298ae56a1827c477d085	Misk Sawalha	14/8/2024 03:00 PM
6381298ae56a1827c477d085	Deyaa Yazeed	1/4/2024 02:00 PM
638ceae27b0d06678017a925	Kareman	16/1/2023 04:00 PM

Figure 4.32: Booking table

12. **Grooming Centers Table & Grooming Centers Booking:** Here we display the list of grooming centers in the database and the details of it like specialist name, clinic name, rating and description.

The screenshot shows a web application interface with a sidebar on the left containing navigation options: Dashboard, Users, Categories, Products, Veterinarians, Grooming Centers, Hotel, and Sign Out. The main content area is titled 'Grooming Centers' and contains two tables.

Grooming Centers Table

PERSONNEL NAME	CLINIC NAME	RATING	DESCRIPTION
Sara Jameel	Haven clinic	5	We clean and dry pets
mohammed hamadneh	Paws & Claws	5	Cleaning your pets and making it look beautiful
Misk Jarara	Pet Paradise	3	Cleaning your pet and, Making it look beautiful
Ezz Berawi	Furry Friends	2	Cut and clean your pet's hair

Bookings Table

USER ID	DOCTOR ID	DATE
6381298ae56a1827c477d085	Ezz Berawi	17/8/2024 12:00 PM
63c818d93d58872a284513b4	Misk Jarara	20/7/2024 12:00 PM
63c855f33eb5aa303499a75f	mohammed hamadneh	21/6/2024 12:00 PM

Figure 4.33: Grooming table

13. **Hotels:** In this table we displaying all hotel bookings with the user ID and the start/last date of the booking.

The screenshot shows a web application interface with a sidebar on the left containing navigation options: Dashboard, Users, Categories, Products, Veterinarians, Grooming Centers, Hotel, and Sign Out. The main content area is titled 'Hotel Reservations' and contains one table.

Hotel Bookings Table

USER ID	ANIMAL TYPE	STARTING DATE	ENDING DATE
6381298ae56a1827c477d085	Dog	Tue Jan 03 2023 00:00:00 GMT+0200	Thu Jan 05 2023 00:00:00 GMT+0200
6381298ae56a1827c477d085	Cat	Thu Jan 19 2023 00:00:00 GMT+0200	Fri Jan 20 2023 00:00:00 GMT+0200
63c818d93d58872a284513b4	Bird	Mon Jan 23 2023 00:00:00 GMT+0200	Wed Jan 25 2023 00:00:00 GMT+0200
63c855f33eb5aa303499a75f	Rabbit	Tue Jan 17 2023 00:00:00 GMT+0200	Thu Jan 19 2023 00:00:00 GMT+0200
63c88ec33eb5aa303499a764	Other	Sun Jan 29 2023 00:00:00 GMT+0200	Wed Feb 01 2023 00:00:00 GMT+0200
63c786c43eb5aa303499a75e	Dog	Fri Feb 10 2023 00:00:00 GMT+0200	Mon Feb 13 2023 00:00:00 GMT+0200
63c8933ae33eb5aa303499a76d	Cat	Tue Feb 14 2023 00:00:00 GMT+0200	Wed Feb 15 2023 00:00:00 GMT+0200
63c7a1d93d58872a284513b4	Parrot	Thu Feb 16 2023 00:00:00 GMT+0200	Sat Feb 18 2023 00:00:00 GMT+0200
63c882e13eb5aa303499a771	Hamster	Mon Feb 20 2023 00:00:00 GMT+0200	Wed Feb 22 2023 00:00:00 GMT+0200
63c8d3f33eb5aa303499a773	Dog	Thu Feb 23 2023 00:00:00 GMT+0200	Sun Feb 26 2023 00:00:00 GMT+0200

Figure 4.34: hotel table

4.9 Database and API Server

Database Table			
API Request	Method Request	The Goal	Table fields
/signup	Post	To add new user	name, email, password
/Login	Post	Log the user	email, password
/userId	Patch	Update user data	Name, email, password
/data	Get	Authentication user	-
/login	Post	Log the Admin	username, password
Product/create	Post	insert product for shop online category	category,name, price,description, productImage, quantity,categoryId

Database Table

Product/show	Get	Get the stored products	-
/getProductFor Catergory	Get	Get the stored products by category	-
gromming/crea te	Post	Insert groom- ing people for grooming	grompersonId, clinicname, clinicgrom- Name,descripti on,image, rat- ing
gromming/get	Get	Get the grooming people for clinic	-
gromming/bok ing	post	Booking a day and time grompersonId	-
gromming/get Boking	Get	Get the user's appointments	-

Database Table			
Doctor/create	Post	Insert doctors' people for clinic	docId name clinicName, description image, exp rating
Doctor/get	Get	Get the doctor people for clinic	-
Doctor/boking	Post	Booking a day and time	docId, userId, dateTime sta- tus
Doctor/getBok Doctor/getid	Get	Get the user's appointments Get doctor id	-
category/creat	Post	Insert new category	name, category Id
category/show	Get	Get the categories	-

Database Table			
Cart/show/userId	Get	Get all order	-
/cart	Post	Insert new order to cart	Product Quantity user
bookhotel/add	Post	Insert New Appointment for hotel	User, last-date, firstdate
bookhotel/get	get	get all user's appointments	-
Adoption/create	Post	To insert adoption	adoptionId,pet Story,name,location,weight,petImage,age,color,gender, type
Adoption/get Adoption/update	Get Post	Get all adoptions Update adoption	- Id Status
reservati/show	get	to get all reservation	-

For the backend of my project, I chose Spring Boot due to its simplicity and efficiency in handling Java-based applications. To test the functionality of my API endpoints, I utilized Postman, which allowed me to verify the requests and responses effectively. For the database, I used XAMPP, which is a local server solution for managing databases, even though it is often associated with PHP. To interact with the database, I implemented JPA (Java Persistence API), which made it easier to map Java objects to the database tables and perform CRUD operations seamlessly.

Chapter 5

Conclusion and Results

We used two databases Spring Boot for the primary database and Firebase for the social network as well as a range of tools and languages, such as Flutter . We developed an intuitive application that allows pet owners to quickly access all of our features and locate everything they require in one location. An owner of a pet can purchase everything required for their upkeep, make reservations for a hotel, veterinary clinic, or grooming parlor, and ask other pet owners for assistance or guidance. We developed a user-friendly website that allows admins to quickly view and change data.

1. **Learning:** We've utilized new technologies that we've never used before, so it's obvious that learning new things involves thorough research and careful application, both of which cost time and effort. We decided on flutter as our platform and Spring Boot for back- and-end development and handling two distinct database schemas. The internet gave us access to an abundance of materials that helped us learn.
2. **Challenges:**
 - (a) Using new programming languages and frameworks made implementation challenging.
 - (b) Excessive editors and simulations that strained our laptops.
 - (c) Diverse features from various disciplines caused delays in our efforts.
 - (d) Difficult to operate with two databases

Chapter 6

Conclusion

1. **Summary:** We have created a cross-platform application that makes it easier for anyone who loves animals or who owns a pet to find everything they need in one place. The application offers services such as a section where users can purchase items and schedule appointments for veterinary or cosmetic clinics for their animal, adopt a pet, and finally a special section dedicated to social media, where users can communicate with each other, exchange messages, hold discussions, or look for a missing cat.
2. **Things we learned:**
 - (a) Programming for mobile devices with the Flutter framework, written in Dart.
 - (b) Developing a JavaScript-written, high-performance REST API with the spring boot framework.
 - (c) Using Postman to test API endpoints.
 - (d) Establish a conversation program.
 - (e) Using data from websites in our application, interacting with developer APIs, and working with servers and URLs with confidence
 - (f) Utilizing created APIs.
 - (g) Working with GitHub and Git; coordinating with other Flutter packages to complete our project.
3. **Recommendations** In computer science and engineering departments, mobile app development has grown in importance and merits further discussion. It will save time and effort for students to take a course that teaches the fundamentals of creating applications.

4. **Future Work:** There are many features we want to add to our software and many more we want to improve, will not stop there.
- (a) In addition to group chats, the chat system will be improved to accommodate voice messages, photos, and videos.
 - (b) For simpler and more precise delivery address input, use enhanced map integration and automated geolocation detection.
 - (c) By adding the narrative function and the ability to follow and unfollow, the app is becoming more similar to social media.
 - (d) Collaborate with local pet stores, clinics, and service providers to offer exclusive deals, promotions, and discounts to app users.

Chapter 7

References

1. Spring Boot Documentation Spring Framework Team. Spring Boot Documentation. Spring.io, <https://spring.io/projects/spring-boot> (Accessed September 7, 2024).
2. Postman API Testing Tool. Postman Inc. Postman Documentation. Postman.com, <https://www.postman.com/product/api-client/>(Accessed September 7, 2024).
3. XAMPP Documentation Apache Friends. XAMPP Documentation. Apache Friends, <https://www.apachefriends.org/index.html>(Accessed September 7, 2024).
4. Java Persistence API (JPA) Oracle. Java Persistence API (JPA) Documentation. Oracle.com, <https://docs.oracle.com/javaee/7/tutorial/persistence-intro.htm> (Accessed September 7, 2024).
5. Flutter Documentation Flutter.dev. Flutter Documentation.<https://flutter.dev/docs>(AccessedSeptember7, 2024).
6. RESTful API Design Documentation Fielding, Roy T. Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine, 2000. REST documentation can also be found at:<https://restfulapi.net/>.
7. Aqraldo, B. W., Jessen, Sentoman, Y., Markos, D., & Warnars, H. L. (2021). Detepet Mobile Application for Pet Tracking. 2021 International Conference on Emerging Smart Computing and Informatics (ESCI). Pune.
8. Dsouza, R., Vidhrani, V., Bhatade, S., Kadam, T., & Pednekar, P. (2022). DOG ADOPTION WEBSITE. International Research Journal of Modernization in Engineering Technology and Science, 2585-2590.

9. Shah, M., Shaikh, A., Shaikh, Z., & Shittal, A. (2021). Pet Adoption App. *International Research Journal of Engineering and Technology (IRJET)*, 2004-2006.
10. Ryan, S., & Ziebland, S. (2015). On interviewing people with pets: reflections from qualitative research on people with long-term conditions. *Sociology of Health & Illness*, 37(1), 67-80. doi:10.1111/1467-9566.12176
11. Pawson, R., Greenhalgh, T., Harvey, G., & Walshe, K (2005). Realist review—a new method of systematic review designed for complex policy interventions. *Journal of Health Services Research and Policy*, 10, 21-34. doi:10.1258/1355819054308530
12. Chen, Y., & Elshakankiri, M. (2020, April). Implementation of an IoT based pet care system. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)* (pp. 256-262). IEEE.
13. Sangvanloy, T., & Sookhanaphibarn, K. (2020, March). Automatic Pet Food Dispenser by using Internet of Things (IoT). In *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)* (pp. 132-135). IEEE.
14. Wang, R. (2020). Design of Mini Pets Feeding Intelligent Home System Based on IoT. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing* (pp. 31-40). Springer, Singapore.
15. Kadhim, W., Al-Qaraawi, S. M. (2020). Design and Implementation Of An Interactive System For Zoo Application Using Smart Mobile Phone And Qr Code. *International Journal of Research- Granthaalayah*, 8(4), 285-296.
16. Tang, Z., Hile, H., Bajracharya, S., & Jurdak, R. (2005). Pettracker-pet tracking system using motes. In *Proceedings of the Seventh International Conference on Ubiquitous Computing (UbiComp'05)*, Tokyo, Japan.
17. Lee, S. P., Cheok, A. D., James, T. K. S., Debra, G. P. L., Jie, C. W., Chuang, W., & Farbiz, F. (2006). A mobile pet wearable computer and mixed reality system for human–poultry interaction through the internet. *Personal and Ubiquitous Computing*, 10(5), 301-317.
18. Luayon, A. A. A., Tolentino, G. F. Z., Almazan, V. K. B., Pascual, P. E. S., & Samonte, M. J. C. (2019, January). PetCare: a smart pet care IoT mobile application. In *Proceedings of the 10th International Conference on E-Education, E- Business, E-Management, and E-Learning* (pp. 427- 431).

Chapter 8

Acknowledgements

First and foremost, we are incredibly grateful to God for his unwavering support during this project. Without it, we could not have finished it successfully.

I am grateful to my parents and other lovely families for their unwavering support and encouragement. Their efforts inspire me to go toward my goals. My parents in particular deserve special thanks for their assistance.

We appreciate our supervisor, Dr. Haya Samanna, for his insightful counsel, insightful remarks, and helpful critiques, all of which contributed to the enhancement of the project's characteristics.

Special thanks to all of the professors at An-Najah National University's Computer Engineering Department, with whom we have been studying for the past five years. Thank you for supporting our efforts and assisting us in bettering ourselves.

And many thanks to our friends and classmates for making university life easier and providing us with wonderful memories. This entire journey would not have been possible without you all.