PetSpace

DESIGN DOCUMENT

4th Year Project 'PetSpace'
Student: Emma O'Connor
Student Number: C00237292
Supervisor: Paul Barry
Submission Date: 17/04/2023

Contents

Table of Figures

1. Introduction

The purpose of this design document is to outline the features of the "PetSpace" application. The document will provide information relating to the architecture of the application as well as the chosen design patterns that will be implemented.

System sequence diagrams will be included within the document to illustrate certain performances carried out between either the vet or the pet owner and the system itself.

A class diagram will also be included to demonstrate the structure of the system in detail.

A database layout will also be provided in order to depict the data being stored.

A prototype in the form of screen layouts will also be used in order to illustrate the system.

2. System Architecture
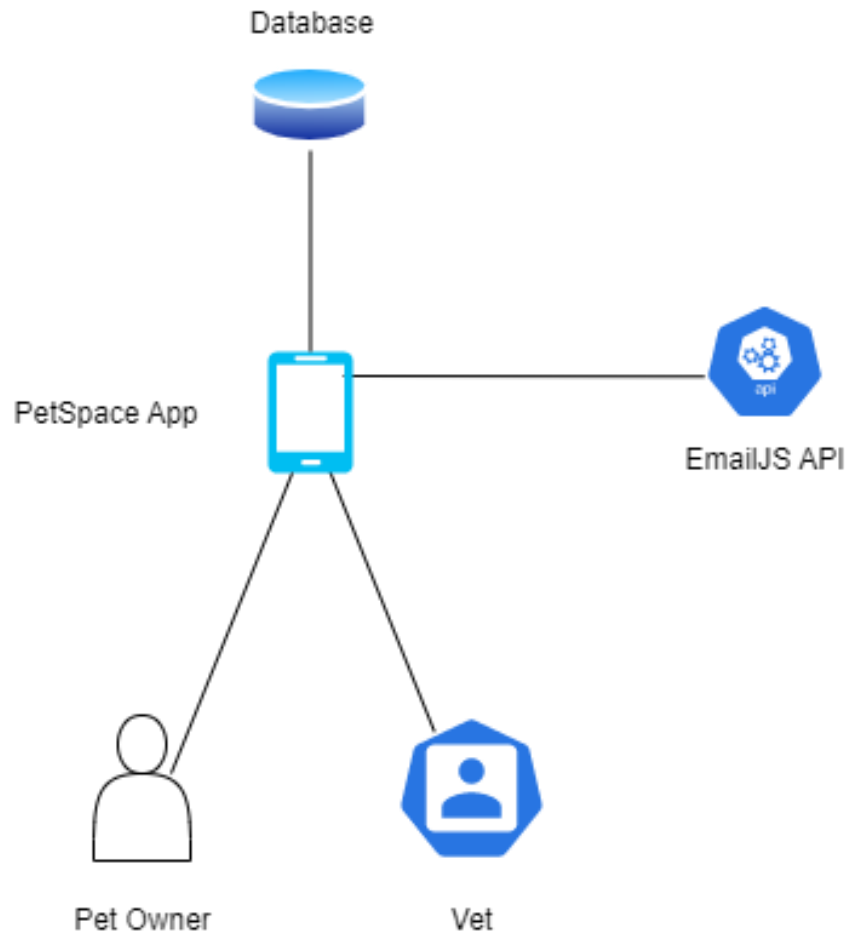
Figure 1 System Architecture

The PetSpace application will be created using React and EmailJS. EmailJS will serve as a provider to send booking information to both the pet owner and vet. It will also be used when a pet owner wishes to contact the vet personally. The front end will be written in JSX using Bootstrap. Node.JS runtime environment will be used for the backend to handle data communication. The database to store the information for the application will be the Firebase Realtime Database.

3. Tools and Technologies

3.1 Firebase

Firebase Realtime Database developed by Google will be the database for this project as it is cloud based non-Structured Query Language (NoSQL) and allows data to be stored/synced in Realtime. It also allows for the storing of images.

Firebase is optimised for offline user which is appealable if the vet or pet owner loses their connection, the information will not be lost as it will be stored locally on their device. When the connection is re-established, the data will be automatically synchronised.

Firebase can also provide functionalities e.g., crash reports and analytics. This allows developers to pinpoint their users' needs.

3.2 React

React, also known as React JS is a cross-platform Javascript library for building user interfaces and designed to work on both web and mobile platforms such as IOS and Android.

React Native is a framework used to create mobile applications for both IOS and Android. It is one of the most actively developed on platforms and the extensive array of libraries allow the developer to deploy their code as a web-application.

React also integrate the use of JavaScript into their development and is especially useful for applications which are both visually appealing and require input from the user. [1]

3.3 Node.js

Node.js is an open-source, cross-platform runtime environment which aids developers in building scalable applications using JavaScript. It is suitable for building high-performance applications that can handle a large number of concurrent connections. It is built on top of JavaScript v8 and enables developers to use JavaScript on both the client-side and server-side. [3]

3.4 EmailJS

EmailJS is a cloud-based service that allows for the sending of emails from the client-side JavaScript code. EmailJS integrate with email providers e.g. Gmail and Outlook and allow for the customisation of email templates with data from the application. The data from the PetSpace booking form will be used in order to generate these templates and emails will be sent to the pet owner and the vet. [4]

4. Design Pattern

Component-Driven Development (CDD) is a pattern that emphasises the use of reusable components as the building blocks of a user interface. The CDD pattern involves breaking down a user interface into small, self-contained components before creating the interface by merging these components together. It is a popular pattern amongst front-end frameworks/libraries like React, Angular and Vue making it easier to build and scale modular user interfaces (UI). [2]

The CDD pattern involves the following steps:

1. **Define Components:** This step identifies the different parts of the UI, defining them as separate and reusable components. Each component has a set of inputs /output properties allowing it to be used across multiple different contexts. E.g. components include the navigation bar, appointment form, signup/login form and contact form.

2. **Build Components:** Create components using front-end framework. Each component has a unique name and styled consistently with the user interface.

3. **Compose Components:** The components are used to build larger, complex user interfaces elements. Components are combined, data is passed between them and defines the structure/behaviour of the UI. For example, the login page could be composed of the login form and signup form components.

4. **Test Components:** Testing the components in different contexts to ensure that they work as expected. This could involve using a tool like Jest or Enzyme to write unit tests for each component.

5. **Document Components:** This step involves documenting each component e.g. its purpose, input and output properties and usage.

One potential disadvantage of the CDD pattern is that it can be time-consuming to set up and maintain a comprehensive library of reusable components. Creating and testing individual components, and then integrating them together to create larger user interface elements, requires a significant amount of upfront planning and effort. Additionally, ensuring that each component works correctly in different contexts, such as on different devices can be challenging and time-consuming. [2]

Overall, the CDD pattern can help PetSpace build a more flexible and scalable user interface by breaking it down into smaller, reusable components that can be composed together in different ways. By following this pattern, PetSpace can also improve its development speed, collaboration, and code quality.
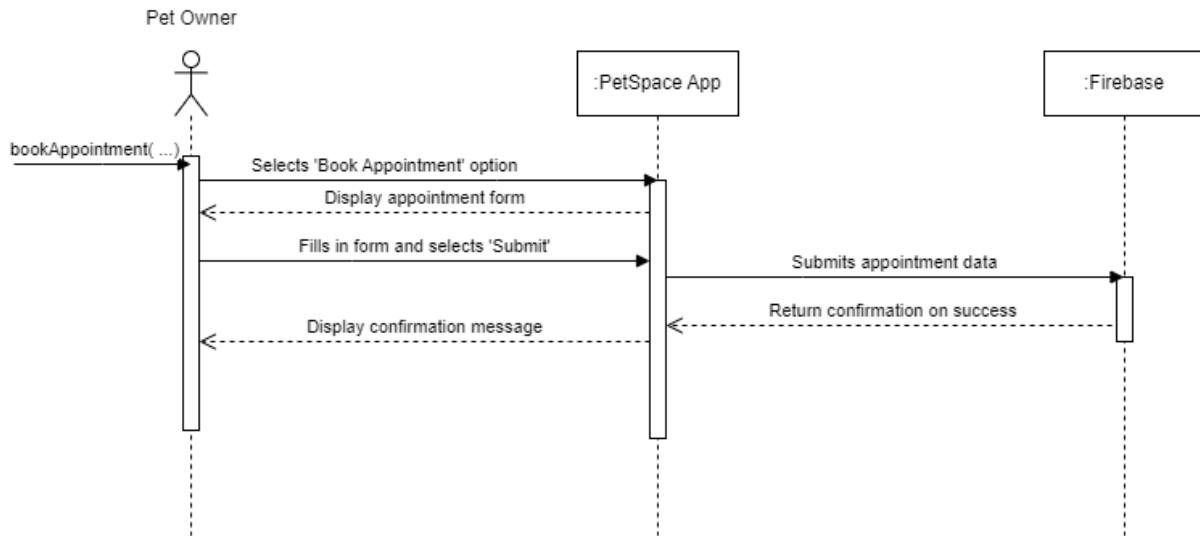
5. Sequence Diagrams

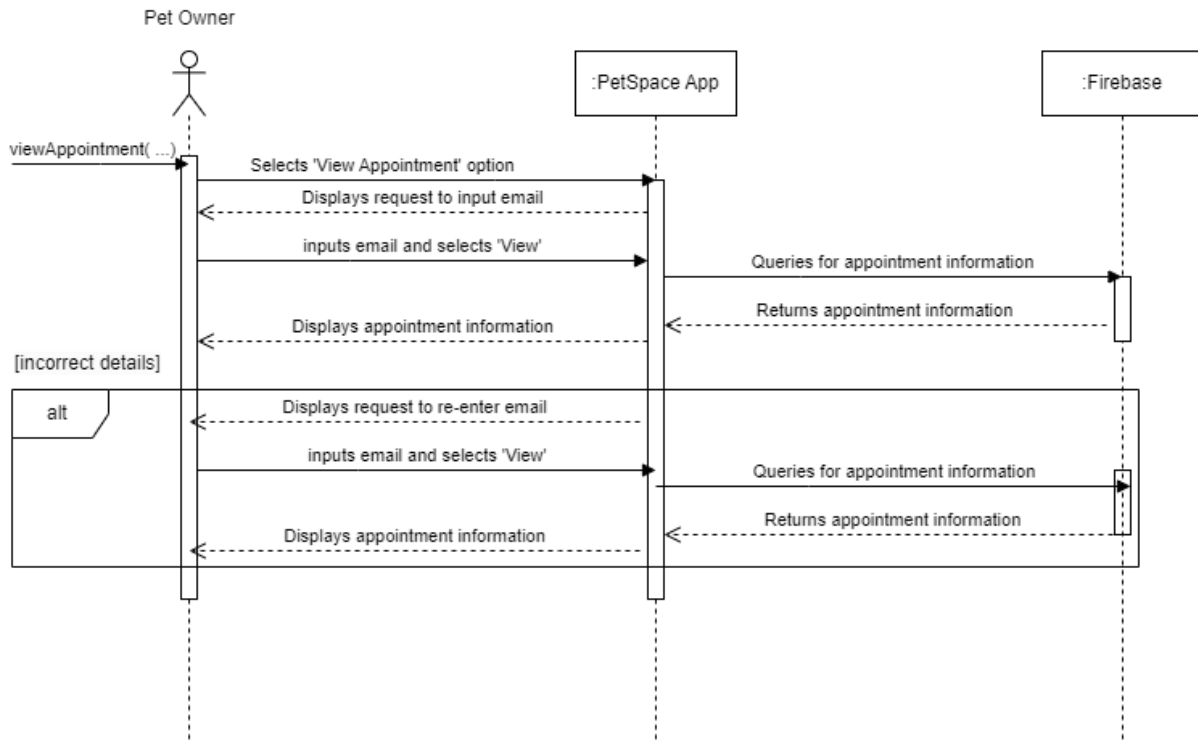*Figure 2 Sequence Diagram 1 - Create Appointment*

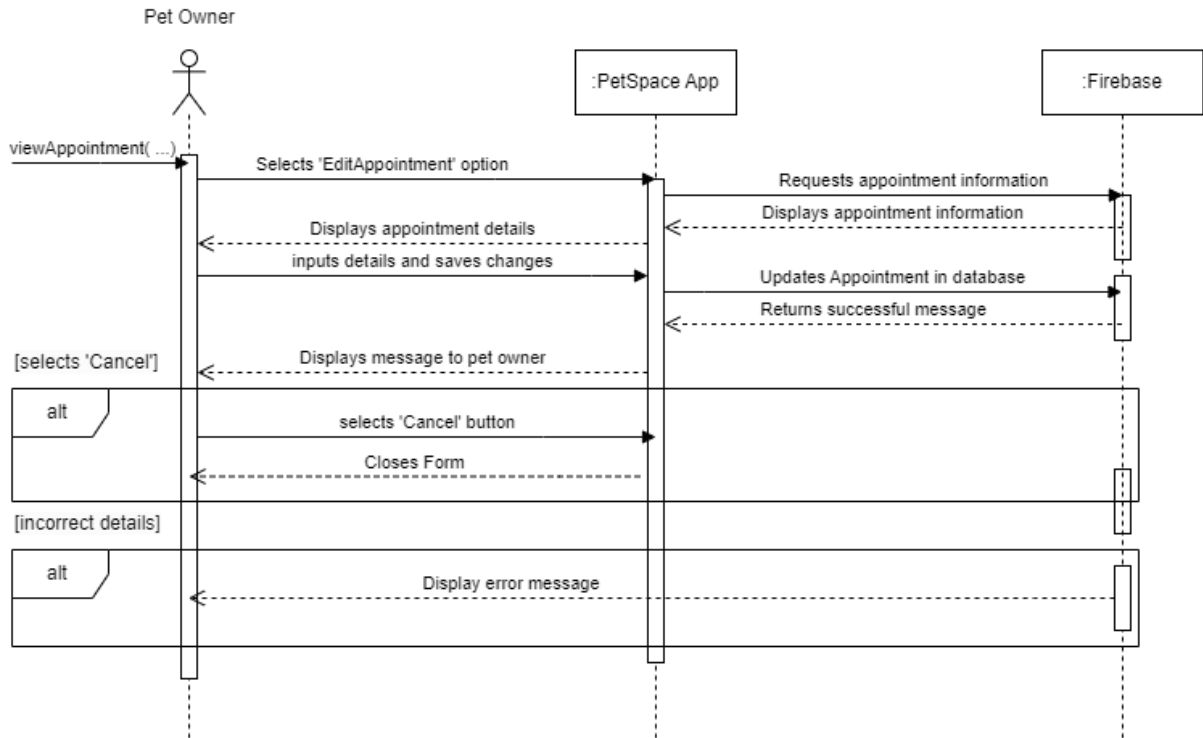*Figure 3 Sequence Diagram 2 - View Appointment*
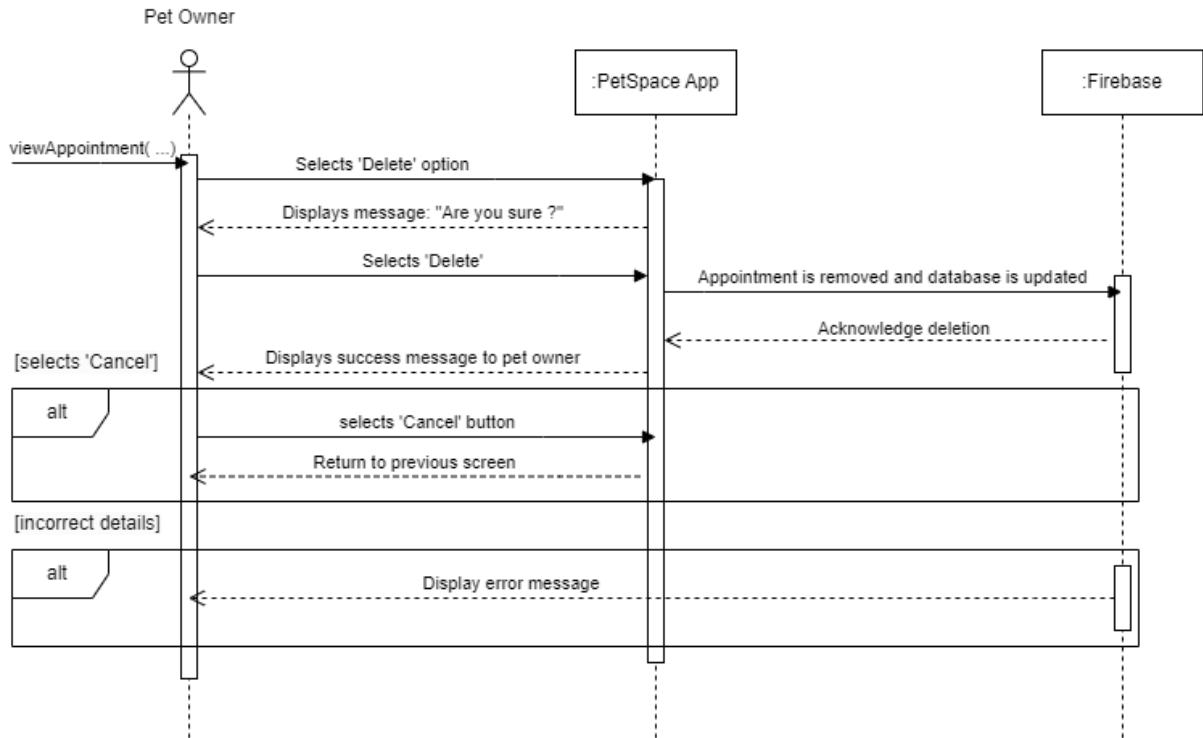
Figure 4 Sequence Diagram 3 - Edit Appointment

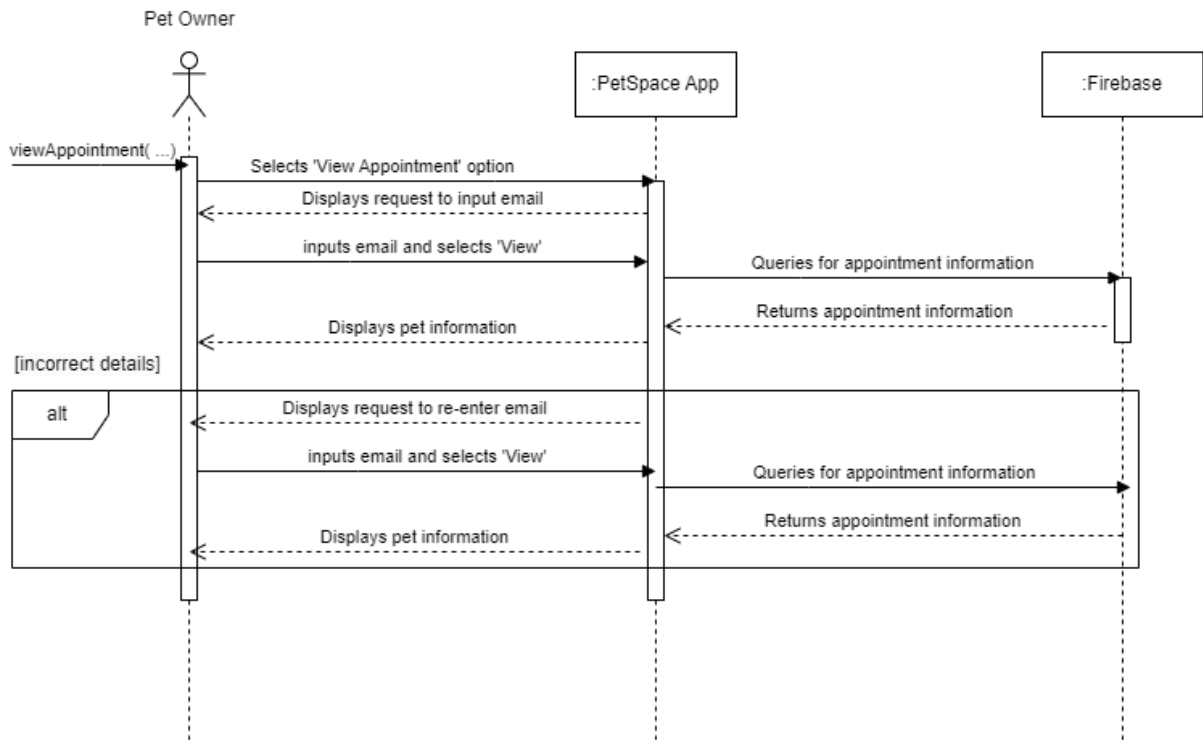*Figure 5 Sequence Diagram 4 - Delete Appointment*

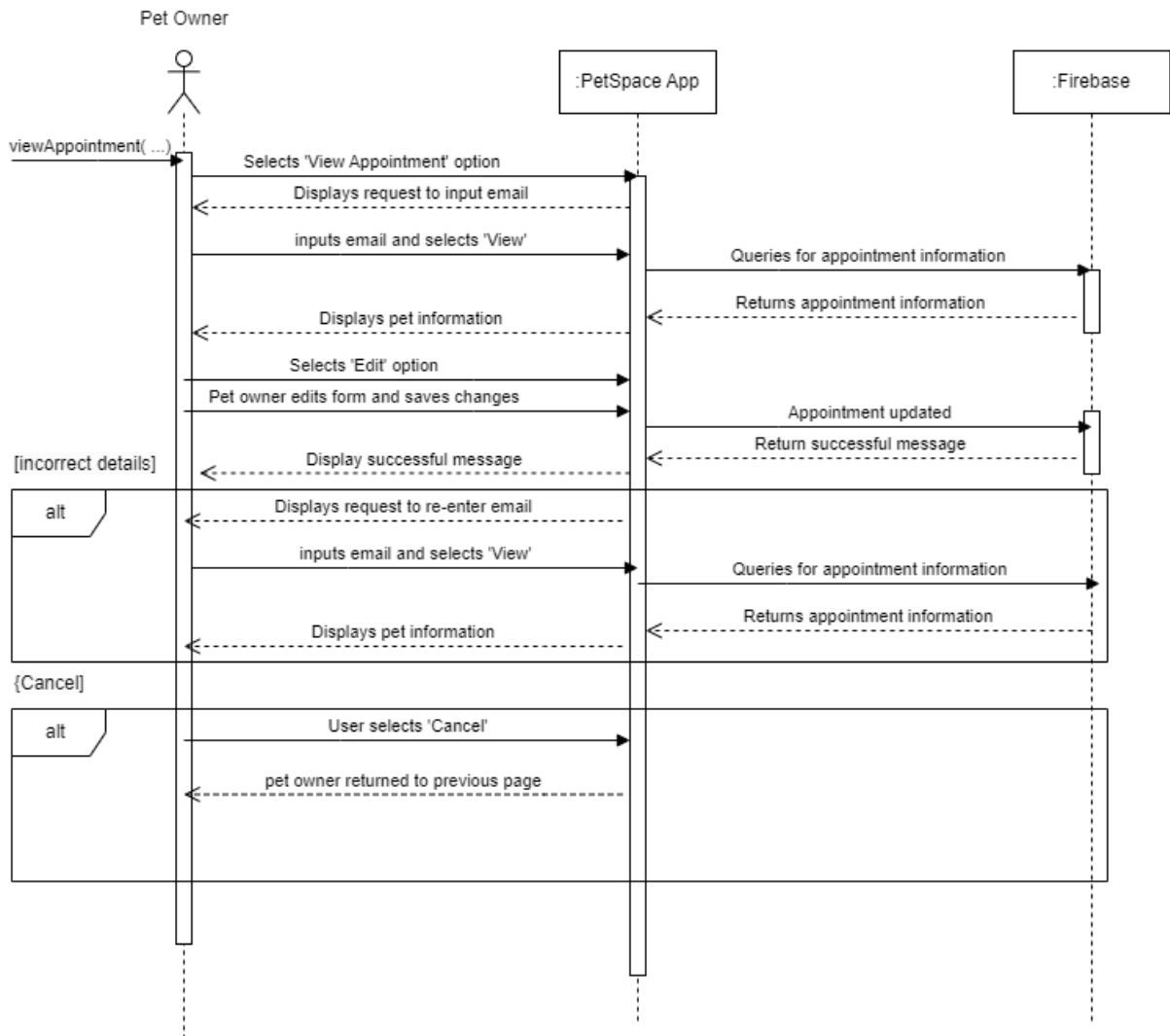*Figure 6 Sequence Diagram 5 - View Pet*
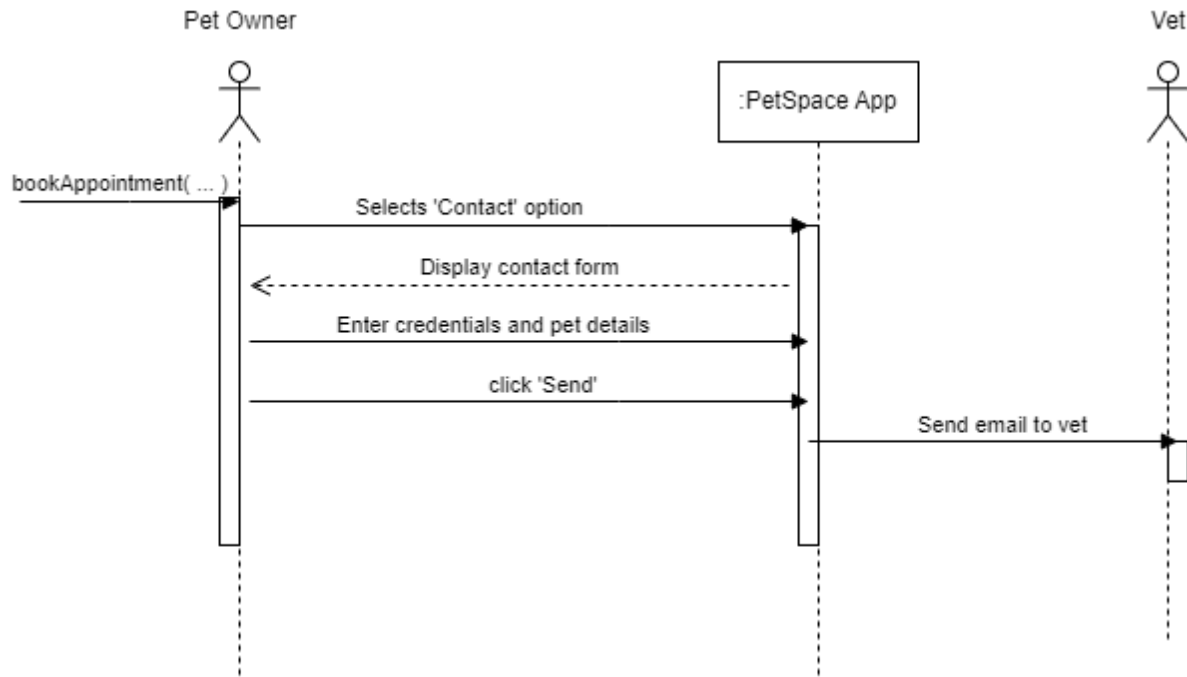
Figure 7 Sequence Diagram 6 - Edit Pet
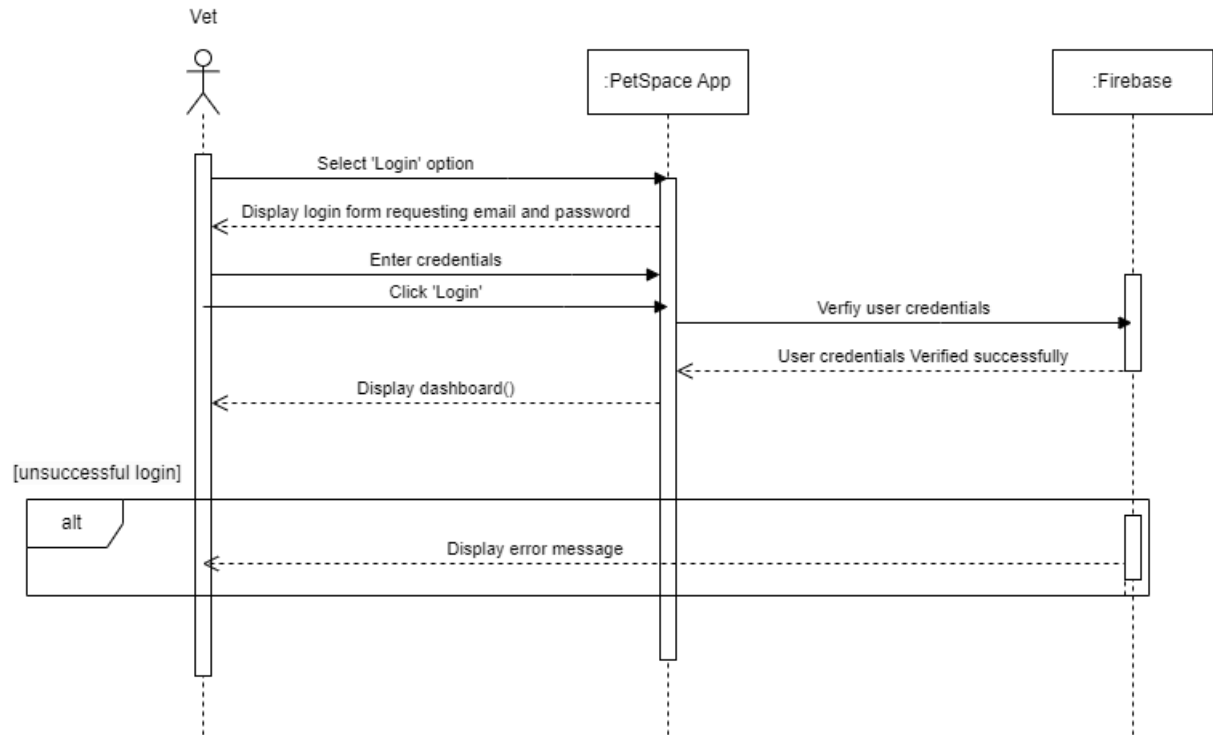
Figure 8 Sequence Diagram 7 - Contact

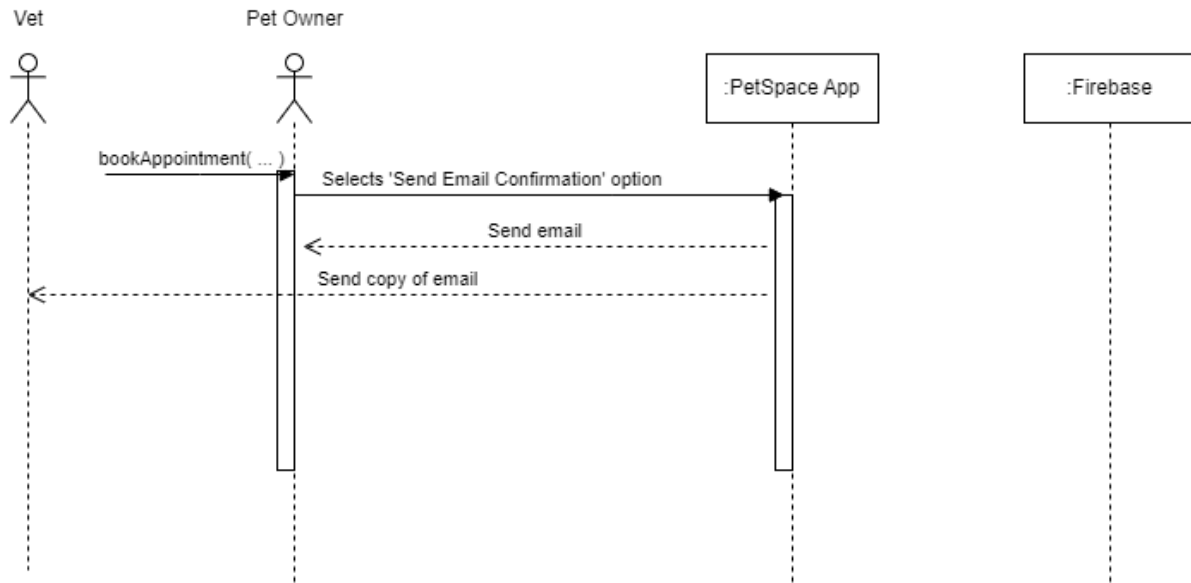*Figure 9 Sequence Diagram 8 - Login*

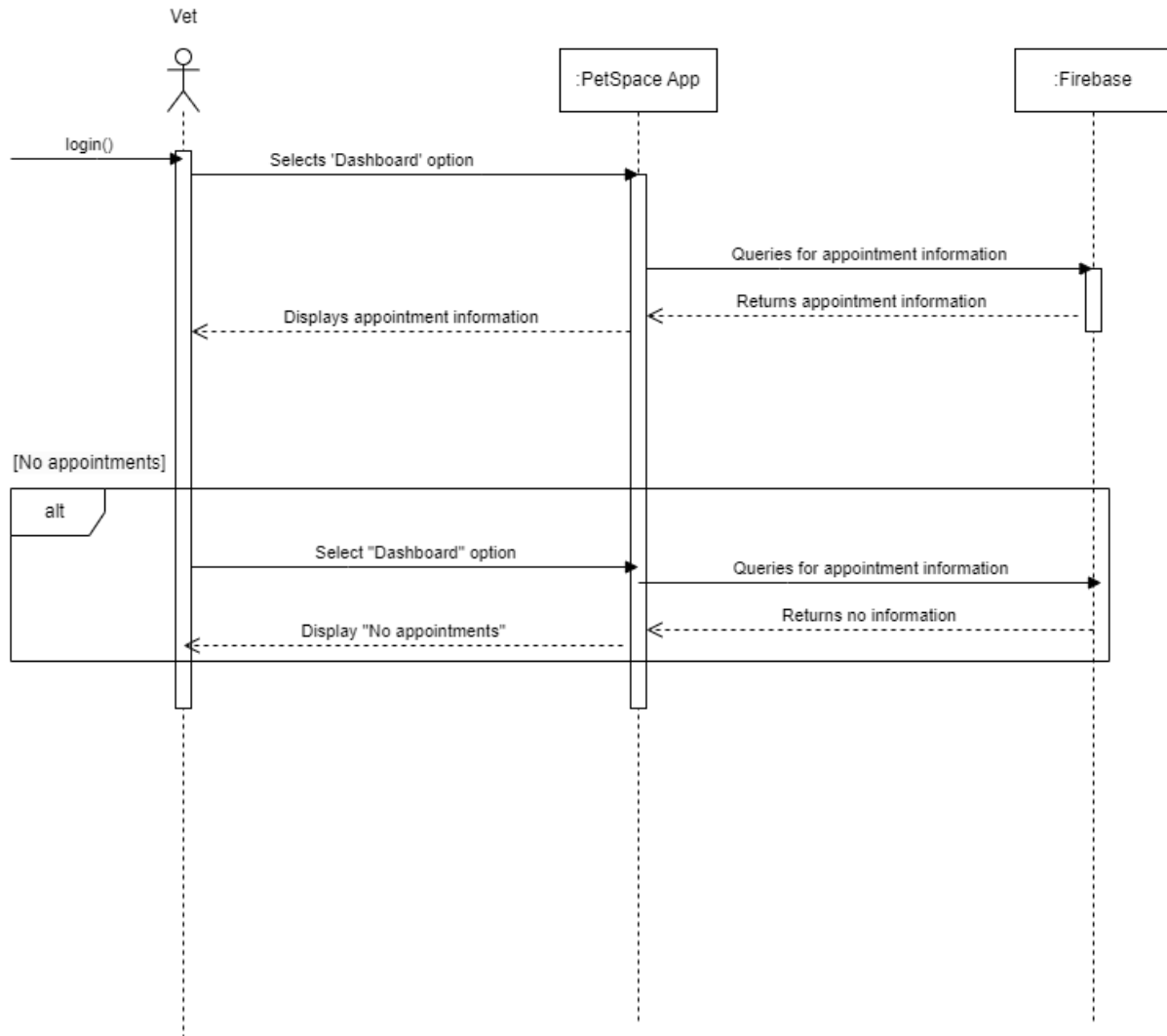*Figure 10 Sequence Diagram 9 - Email Appointment Confirmation*

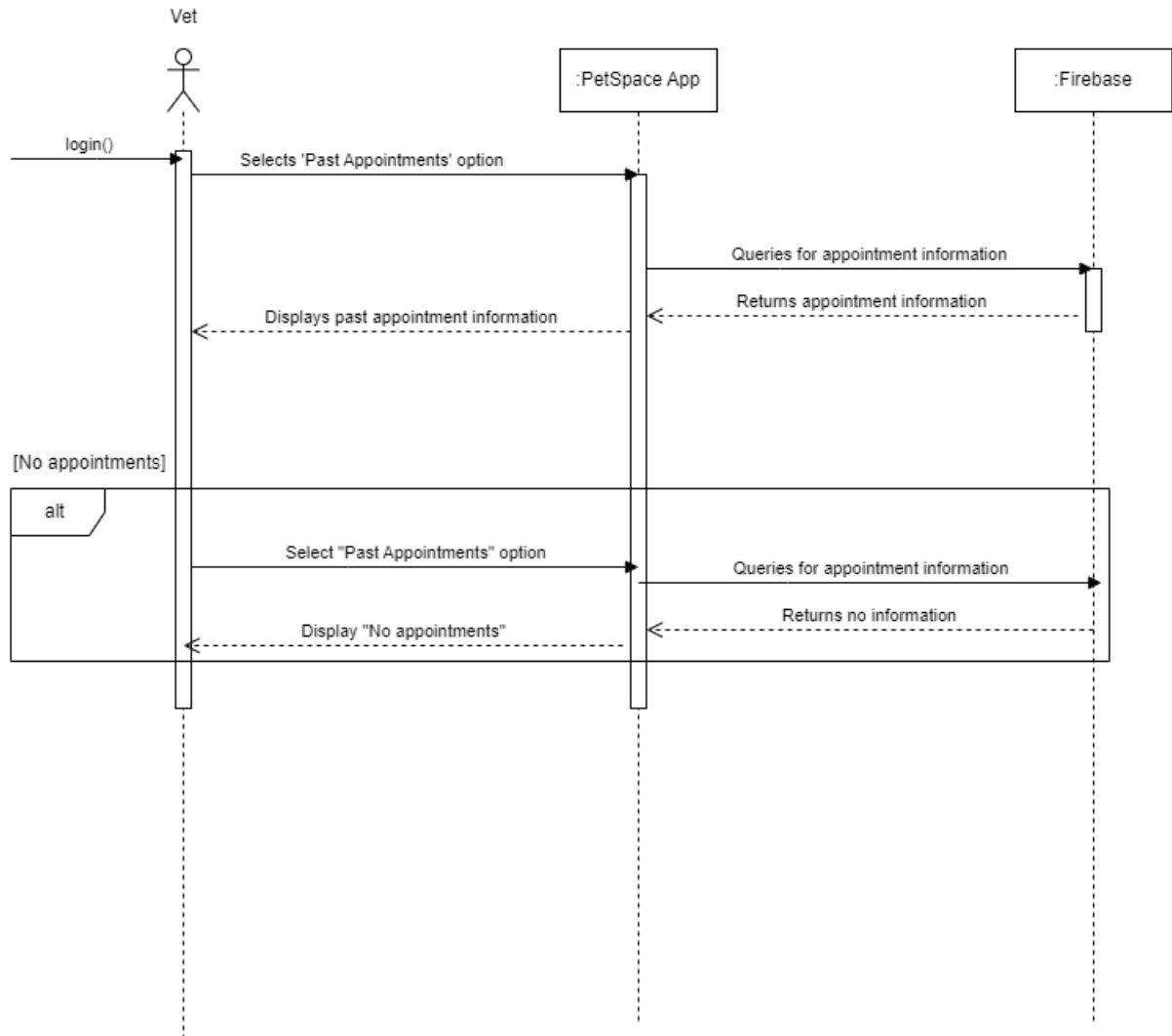*Figure 11 Sequence Diagram 10 - View Appointments*

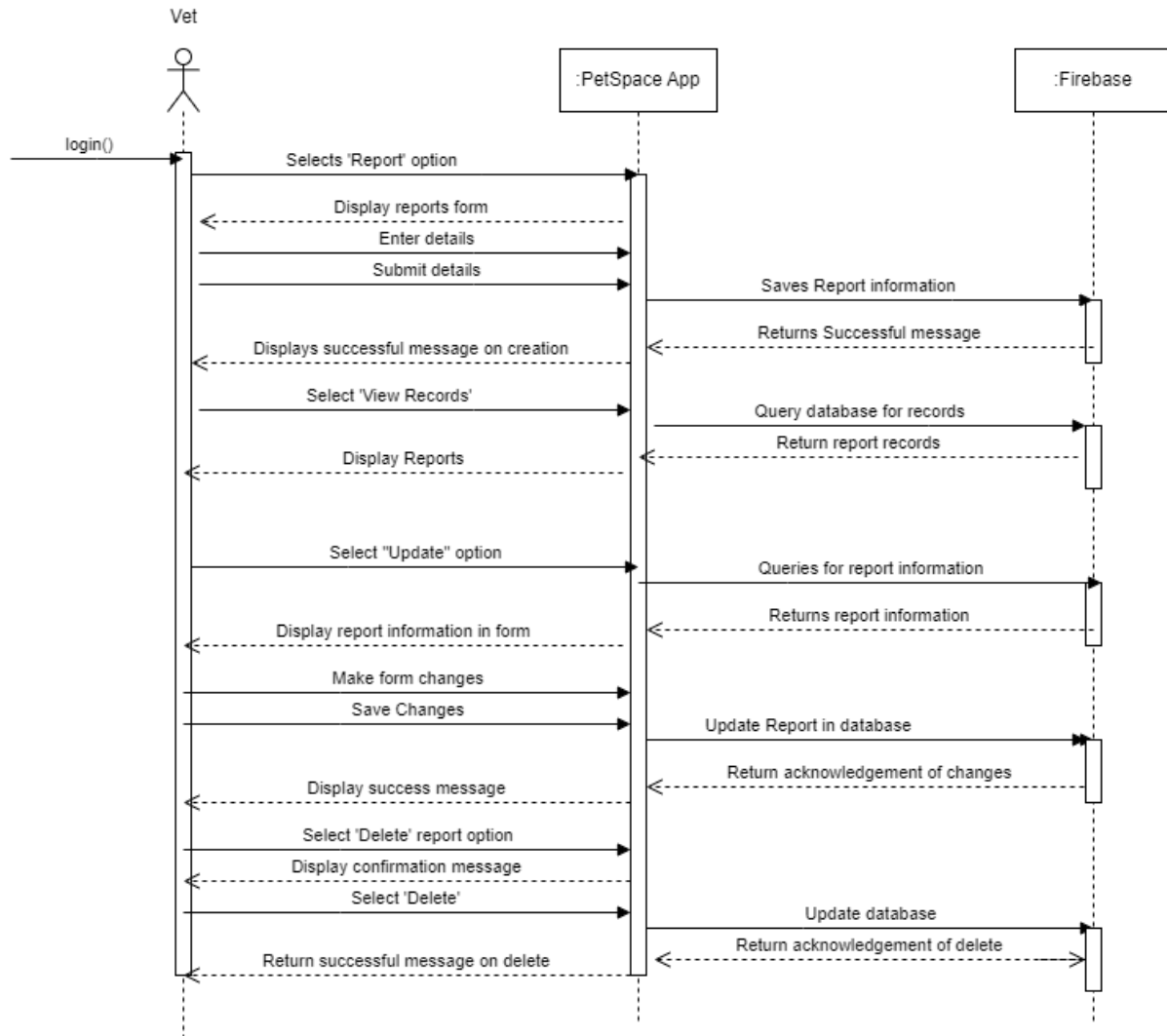*Figure 12 Sequence Diagram 11 – View Past Appointments*

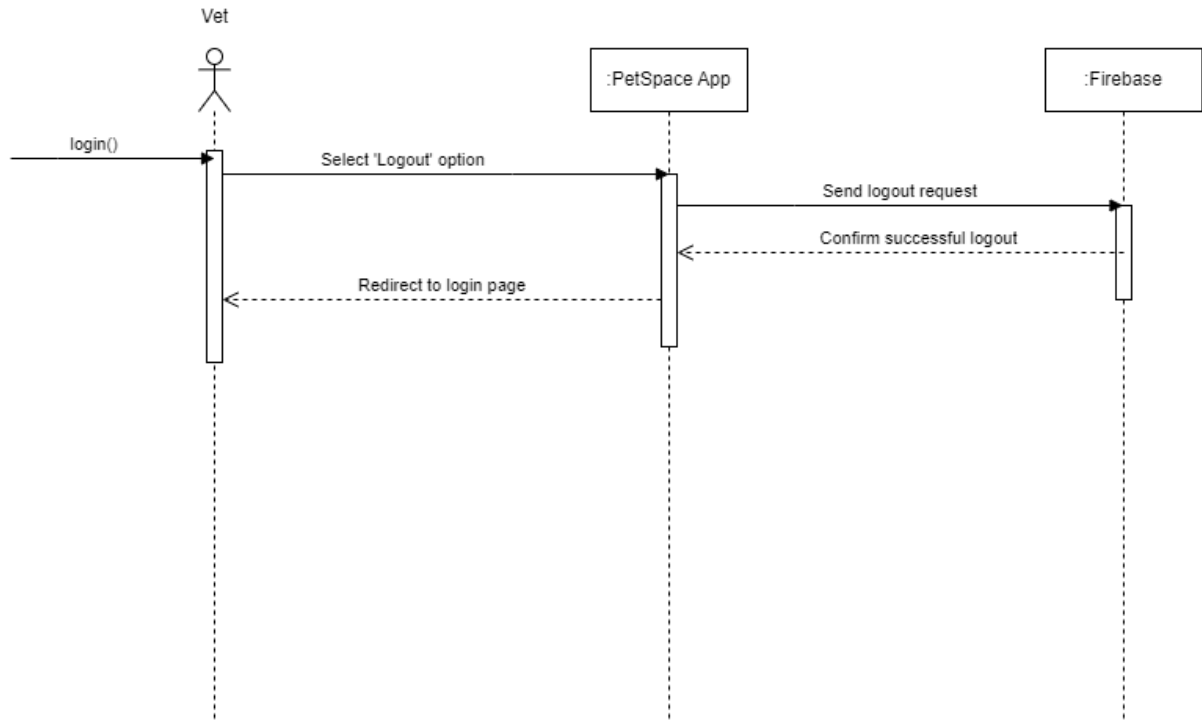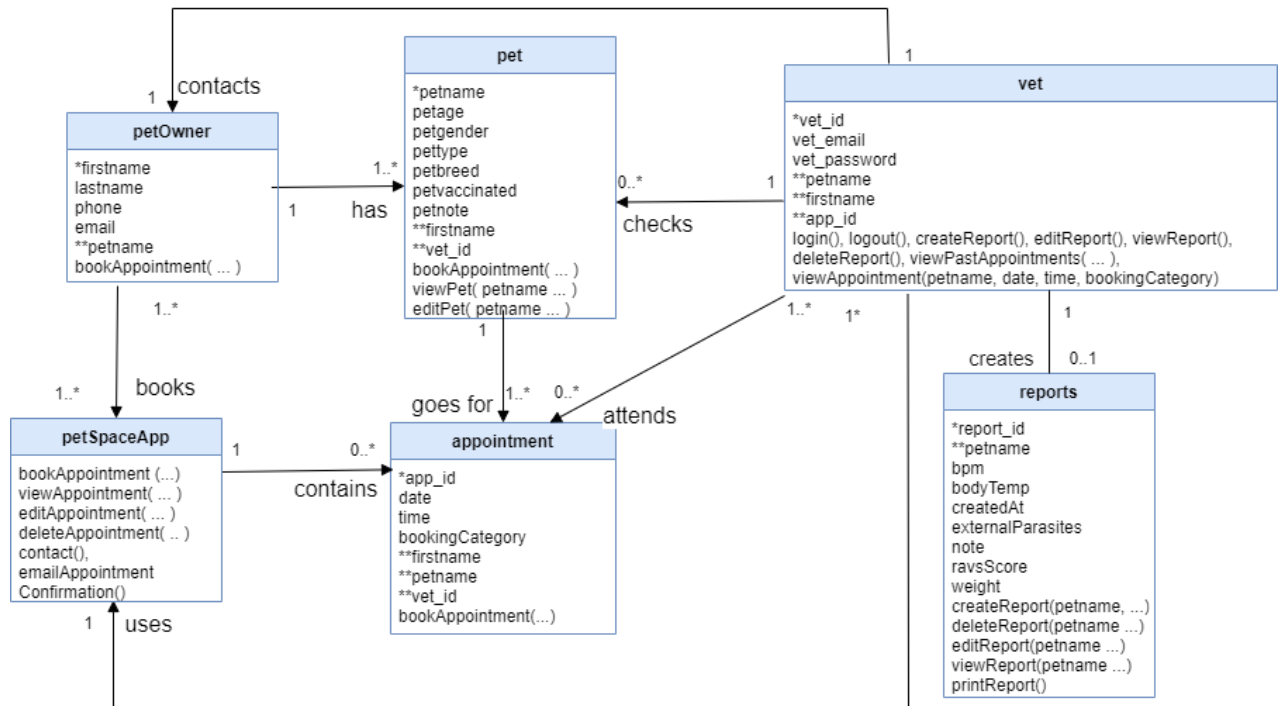*Figure 13 Sequence Diagram 12 - CRUD Report*

Figure 14 Sequence Diagram 13 - Logout

6. Class Diagram

* - Primary Key
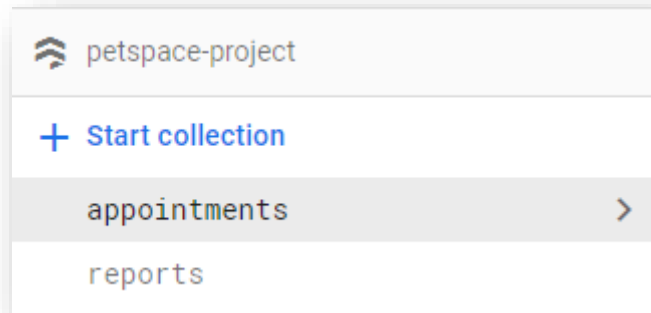*** - Foreign Key*

7. Database Layout
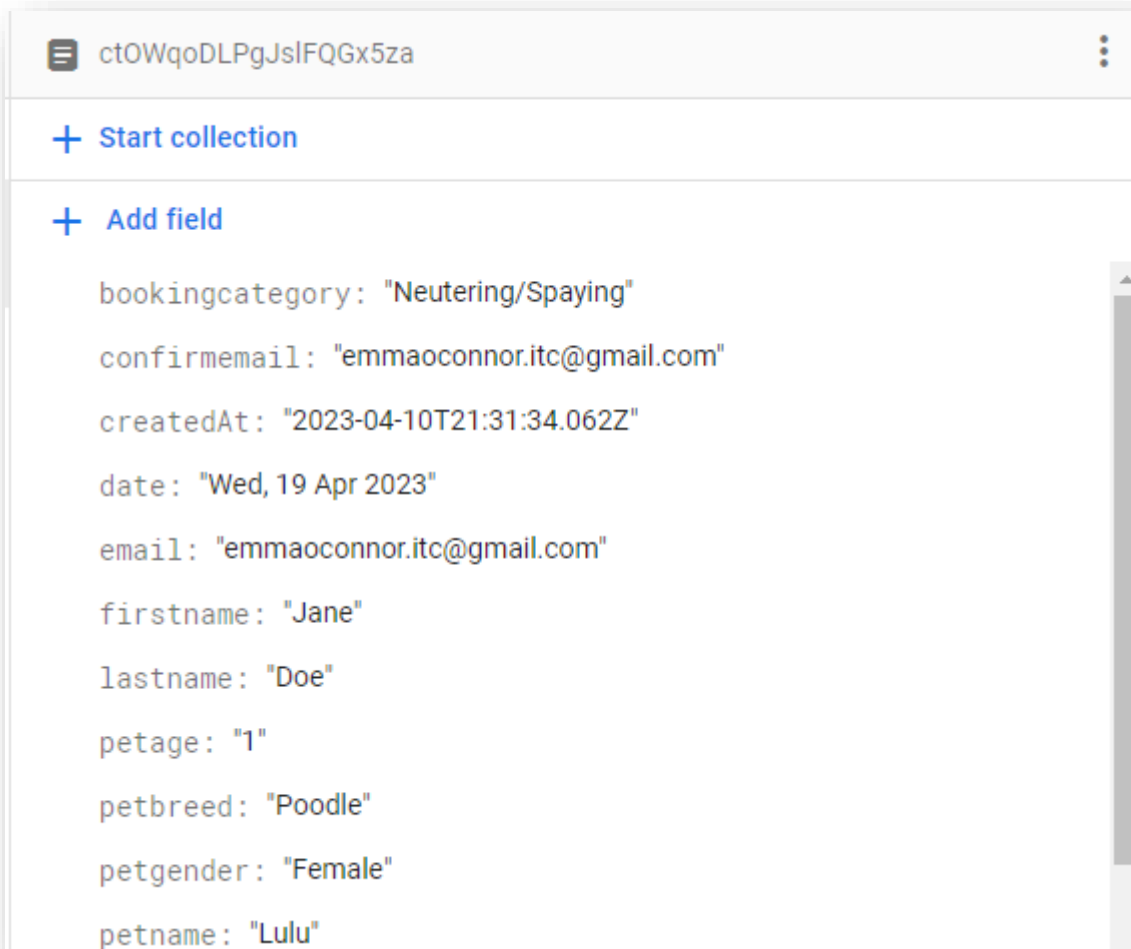

Figure 15 Database Layout – Appointments 1

Figure 16 Database layout - Appointments 2

Figure 17 Database Layout - Appointments 3

Report Database Layout:

Figure 18 Database Layout - Reports 1

8. UI/UX

The standard colours blue, white, and yellow are a good choice for veterinary UI/UX for several reasons:

Blue: Blue is often associated with trust, reliability, and professionalism. These are all important qualities for a veterinary practice such as PetSpace, as clients need to feel confident in their veterinarian's abilities and expertise. Blue also has a calming effect and can help reduce stress, which can be particularly important for pets and their owners who may be anxious about medical procedures.

White: White is often associated with cleanliness, purity, and sterility, which are all important qualities for a veterinary practice. White can also help create a sense of spaciousness and clarity, which can be particularly important for medical environments where details are important.

Yellow: Yellow is often associated with warmth, friendliness, and happiness. It can help create a welcoming and positive environment, which can be particularly important for veterinary practices where pet owners may be nervous or worried about their pets. Yellow can also help create a sense of optimism and hope, which can be important when dealing with medical issues.

Grey: Grey is also a commonly used colour in veterinary UI/UX design as it can complement the colour scheme. Grey is often associated with neutrality, balance and sophistication while also highlighting important information/elements.

Overall, the combination of blue, white yellow and grey can create a professional, clean, and welcoming environment for veterinary practices, while also helping to reduce stress and anxiety for both pets and their owners.

9. Prototype Screens

These illustrations are rough designs created for the UI of PetSpace. The final UI design will appear different when compared to the prototypes below.
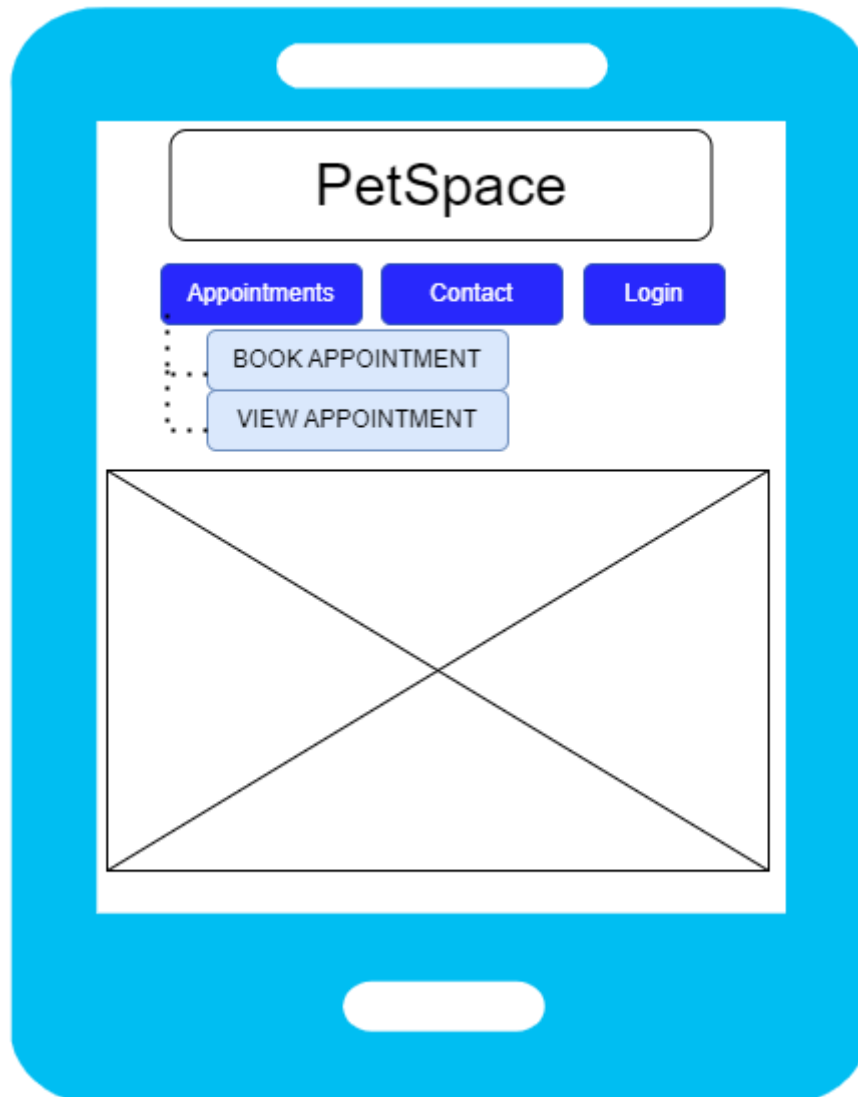
9.1 Landing Page

Figure 19 Landing Page

This is the Landing page for PetSpace that both the pet owner and vet can access. Only vet's however will be able to log into the system as their credentials and authorisation will be stored through firebase. From the 'Appointments' dropdown, the pet owner can choose to 'Book' and appointment or 'View' their booked appointment.

9.2 Book Appointment

Figure 20 Book Appointment Screen 1

The first page of the stepper pages is the 'Booking Details' page which includes a drop-down list of the type of appointment to be desired e.g. Check-up, Surgery, Follow-up Checkup, Boosters and Injections, Neutering/Spaying, Emergency, Teeth-Cleaning and Grooming. After choosing, the available dates for the next appointment are displayed below under 'Available Dates:' along with the times that are available for the pet owner to choose from under 'Available Times:'.
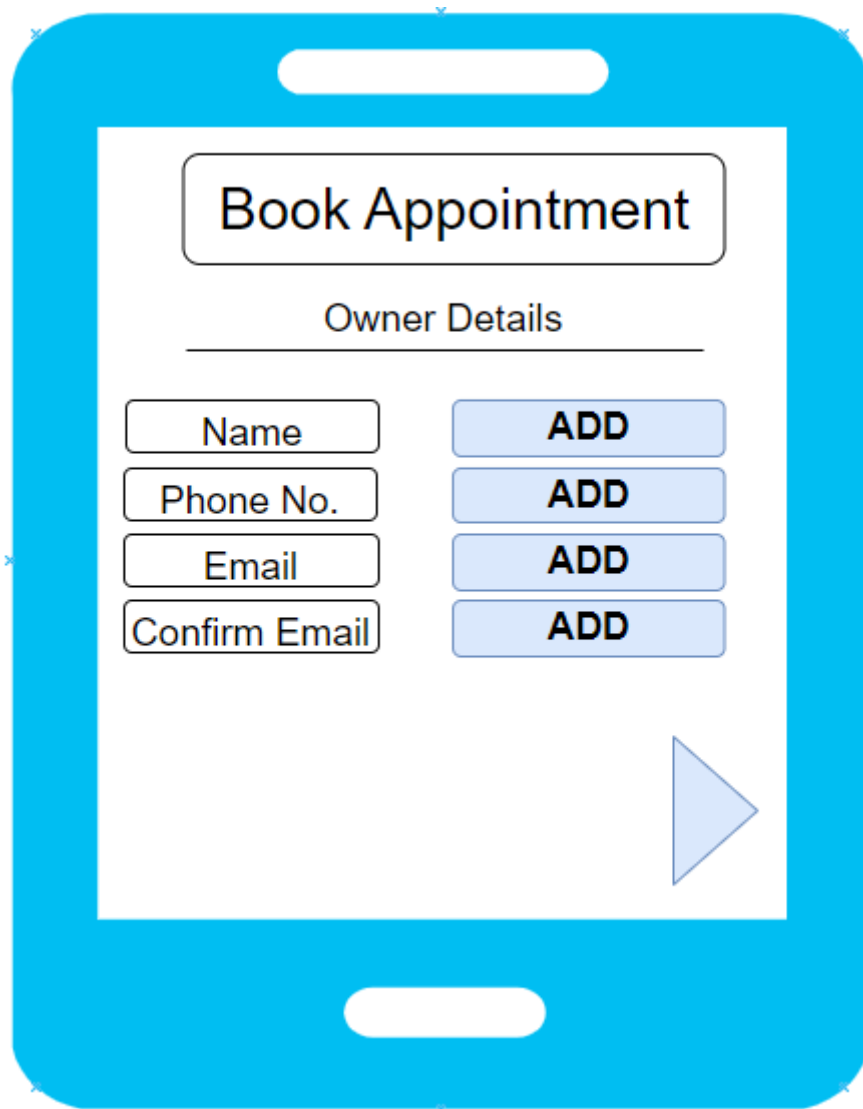
Figure 21 Book Appointment Screen 2

'Step 2' of the booking process involves the 'Pet Details' form page in which the pet owner is encouraged to enter details about the pet e.g. it's Name, Age, Gender, the type of Animal it is, Breed, is the animal Vaccinated? Along with a Note section to allow the pet owner to include any information the vet should be aware of or know. After the details have been recorded, the pet owner proceeds to 'step 3'.

Figure 22 Book Appointment Screen 3

This is the third screen of the booking appointment process for the pet owner. The screen displays the 'Owner Details' form in order to gather information from the pet owner. The purpose of this form is to connect the pet owner to the pet and to email a confirmation of the appointment to be sent to the pet owner. The pet owner clicks 'Next' to continue with the process.
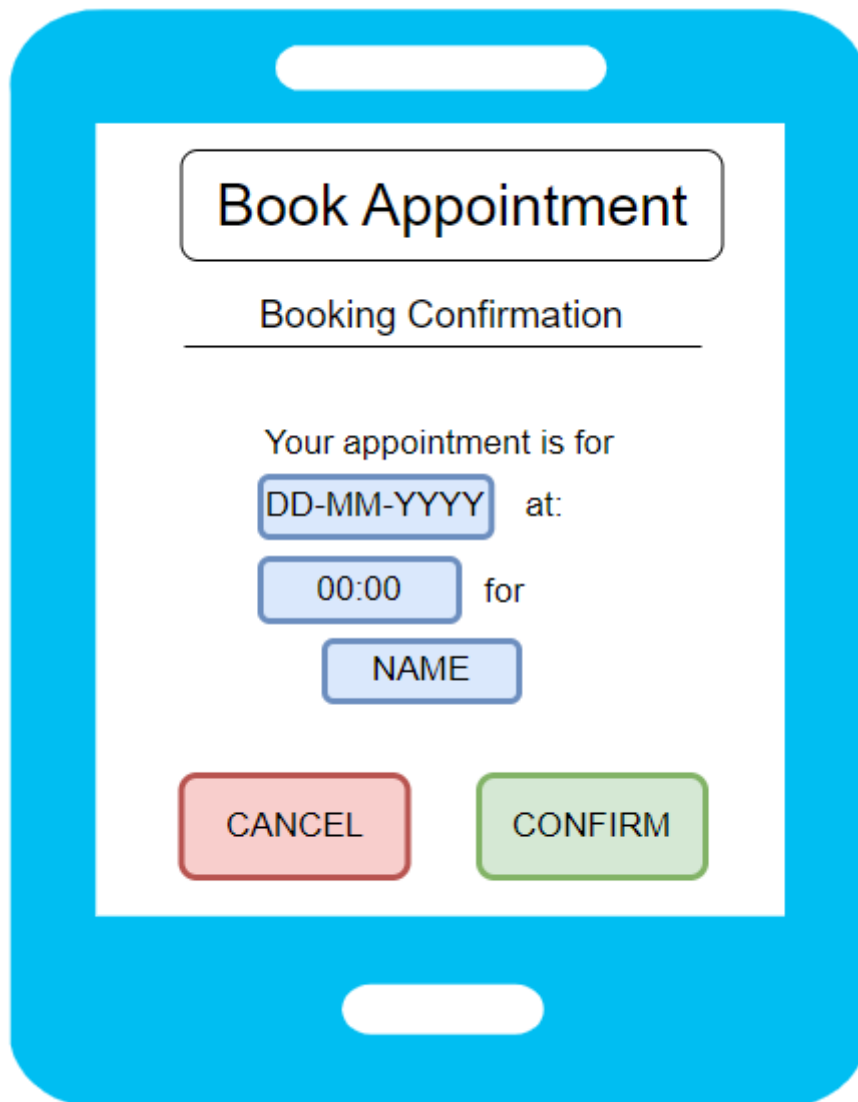
Figure 23 Book Appointment Screen 4

The 'Booking Confirmation' screen displays the booking type, date and time back to the pet owner where they can choose to either cancel or confirm the appointment. The 'Cancel' button will return the pet owner to the 'Home' screen whilst the 'Confirm' button will confirm the appointment.
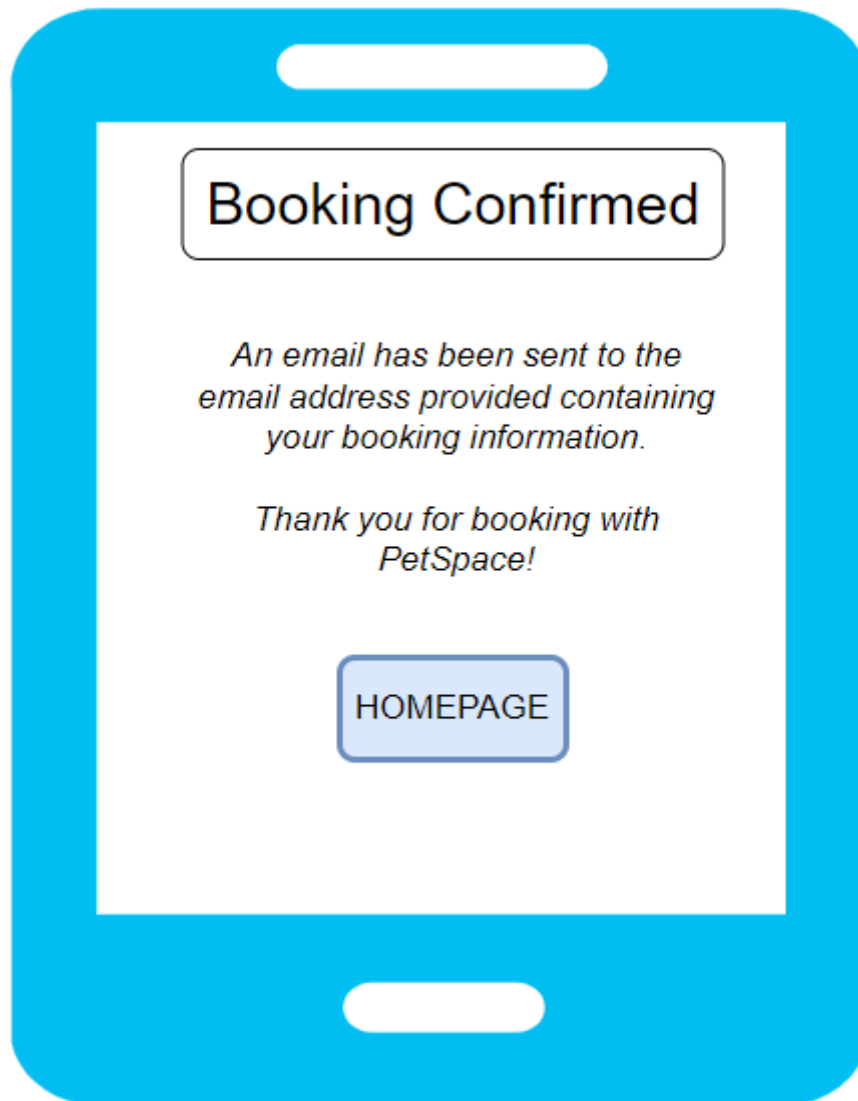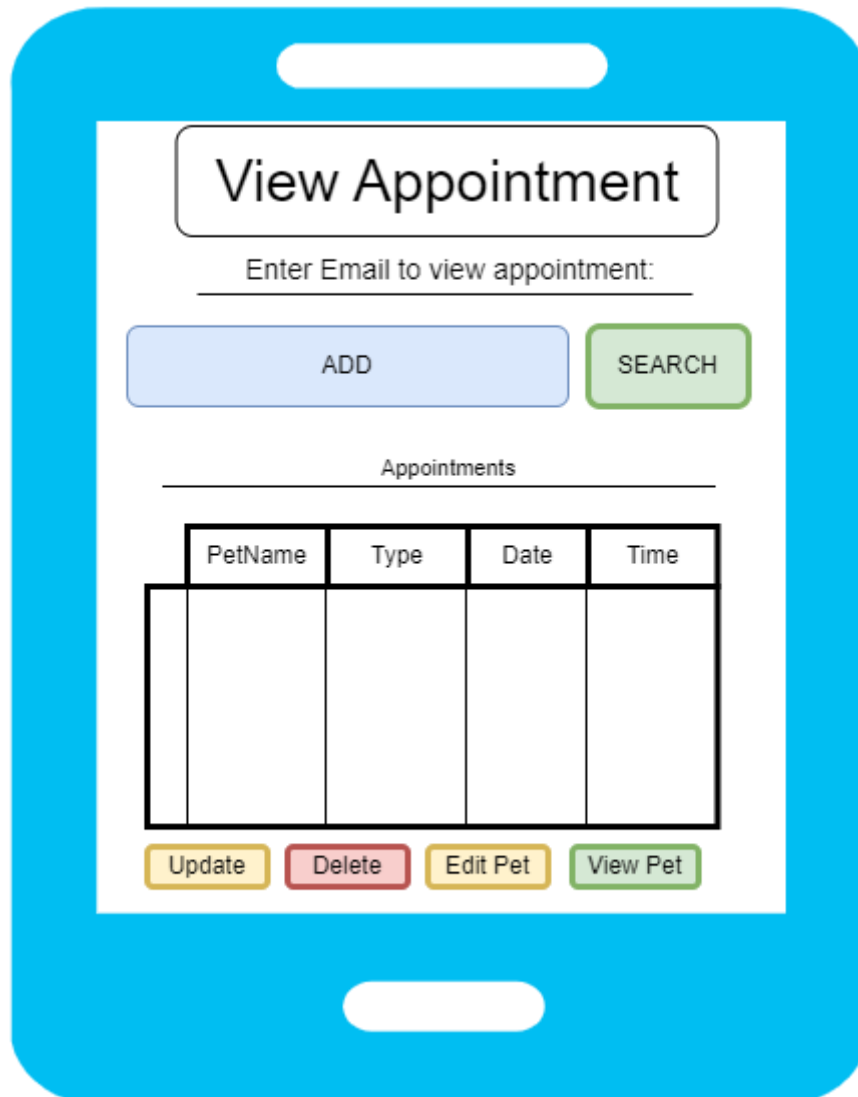
Figure 24 Book Appointment Screen 5

Once the appointment is confirmed, the 'Booking Confirmed' screen is shown and the information relating to the booking is stored in the database. An email confirmation is also sent to the pet owner about their booking along with a copy to the vet. The booking is added to the timesheet for the vet to access. The user clicks the 'Homepage' button at the bottom of the screen which will return them to the home screen of PetSpace.

9.3 View Appointment

Figure 25 View Appointments Screen 1

The 'View Appointments' screen requests the pet owner's email in which the booking was made in order to have full control of their booking. When it is submitted, A table of their booked appointments are shown below with the options to 'delete' and 'update' their appointment. There is also a function to view and edit their pet details.

9.4 Edit Appointment

Figure 26 Edit Appointment Screen 1

When the user chooses the 'update' option next to their appointment, the 'Edit Appointment' screen is loaded. Here, the pet owner can modify their details which will then be updated within the database for the vet when the pet owner clicks 'Update'.

9.5 View Pet

Figure 27 View Pet Screen

This screen is loaded when the pet owner clicks on the 'View Pet' option. This screen displays all of the important information about their pet for the appointment.

9.6 Edit Pet

Figure 28 Edit Pet Screen

This screen is loaded when the pet owner clicks on the 'Edit Pet' option. The screen displays an editable form of their pet's information. When the pet owner clicks on 'update', the database in turn is updated with the modified appointment.

9.7 Contact

Figure 29 Contact Screen

The Contact Screen displays a form that requests the pet owner's Name, Email and the Message they wish to send to the vet. Once the pet owner fills in the form and clicks 'Submit', an email is sent to the vet, notifying them of the email. This is achieved through the EmailJS API.

9.8 Signup

Figure 30 Sign-up Screen
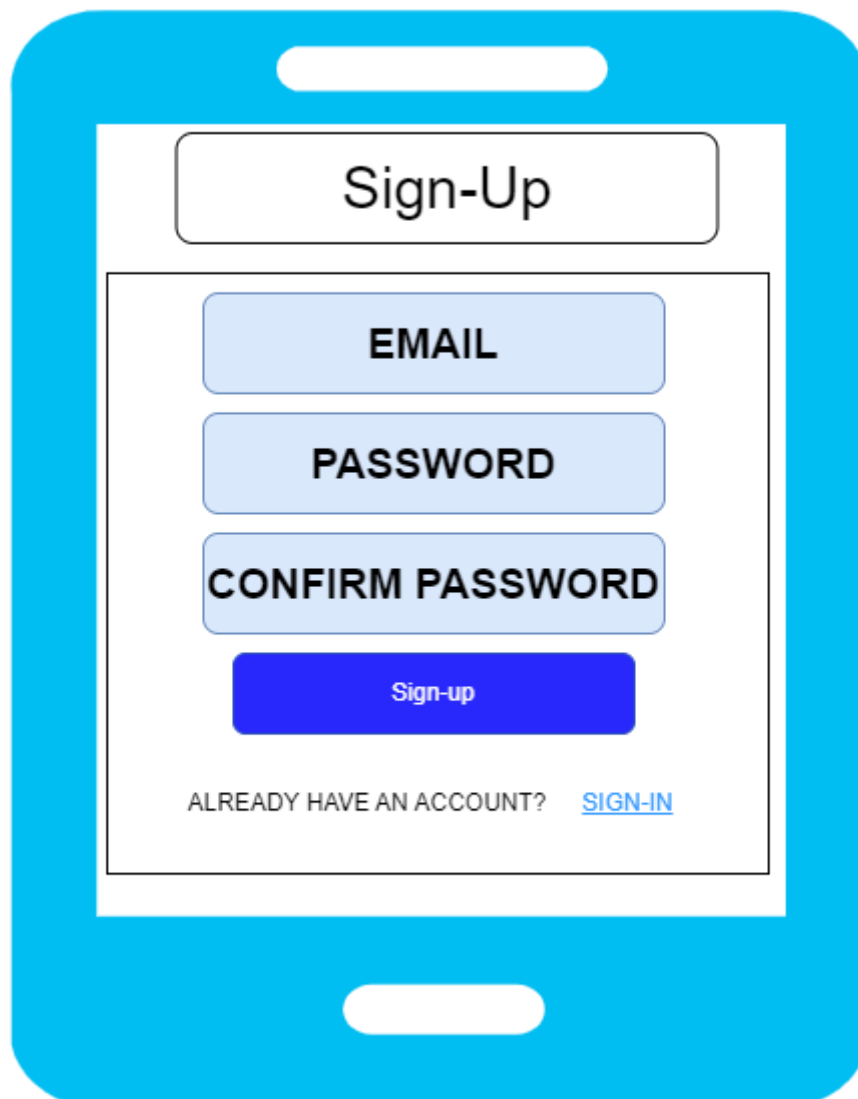
The Sign-up Screen is created as a GUI for new vet's who wish to work with PetSpace. It requires an email and password to sign-up. There is also an option to 'Sign-in' for pre-existing vets.

9.9 Login
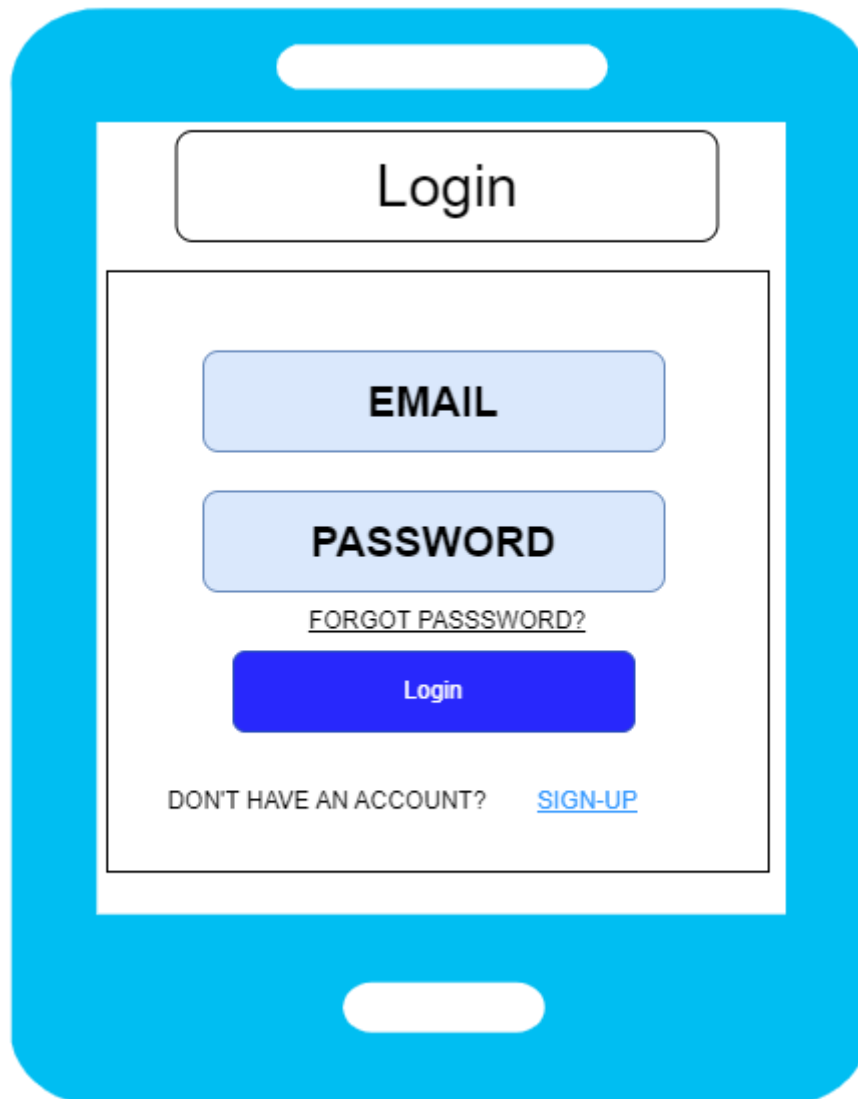
Figure 31 Login Screen

The Login screen is displayed for the vet to access their appointments. The authorisation and logged in data is stored through firebase using the authentication.

9.10 Reset Password
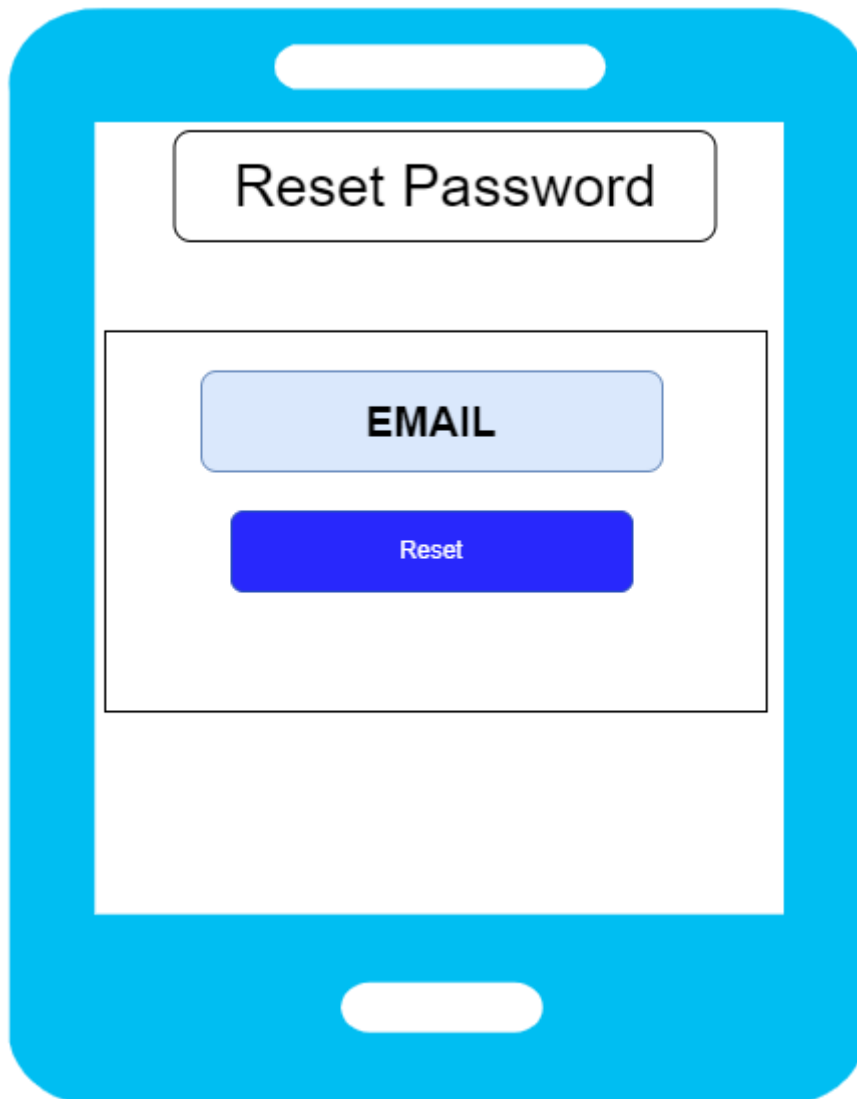
Figure 32 Reset Password Screen

This page is loaded when the user clicks on 'Forget Password?'. The Reset functionality is handled by Firebase and a reset link is sent to the email provided on the screen to change their password. This is better than a change password functionality because the link is securely generated by Firebase, the link is time-sensitive and can only be used once making it more difficult for attackers to gain access to the account.

9.11 Dashboard/View Appointment
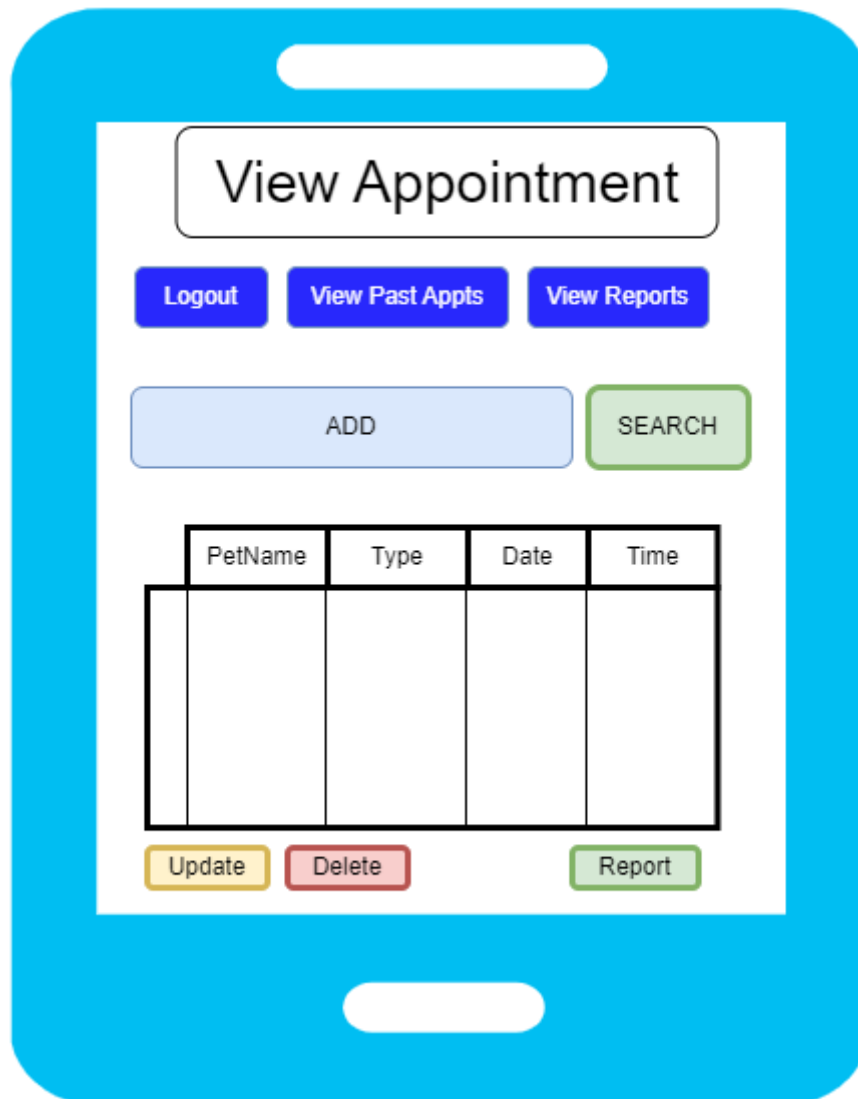
Figure 33 Dashboard/View Appointment Screen

Upon successful login, the 'Dashboard' is loaded for the Vet. The vet will have access to their upcoming appointments, past appointments and the ability to create reports for a specific pet that is stored in the database. They will also be able to delete these reports. There is also the functionality to search for a specific appointment by the animal's name. When a vet wishes to logout, they can do so by clicking the 'Logout' button.

9.12 Create Report

Figure 34 Create Report Screen

When a vet wishes to create a report for the specific pet, this screen displays a form requesting the animal's Body Temperature, BPM, Weight, if there are any External Parasites on the animal e.g. lice, a Ravs Score e.g. general health on a scale of critical to excellent and any extra information in the note section provided. The report is submitted and stored in the database.

9.13 Update Report

Figure 35 Update Report Screen

When A vet wishes to update a specific pet's report, they can simply click on the 'Update' functionality beside the pet's name and modify the details accordingly. They have the choice of whether to 'Cancel' their previous modifications which will not update the pet's record; or choose to 'Update' the report.

9.14 View Reports

Figure 36 View Reports Screen

When the vet wishes to view the reports, they can simply choose the 'View Reports' button which will lead them to this screen. Here they will be able to see all of the reports created for the pets. There is also an option to 'Print' these reports for physical records.

9.15 View Past Appointments

Figure 37 View Past Appointments Screen

This screen is loaded when the vet wishes to view their past booked appointments. A table of each appointment sorted in ascending order of 'Date' is displayed. The details include the pet name, the type of appointment booked along with the time scheduled for the appointment.

10. Conclusion

This document has successfully discussed the design features of the PetSpace application. The tools, Technologies and UI/UX along with the design pattern being used have also been discussed thoroughly. Sequence Diagrams and a class diagram have also been provided for visualisation of the interactions between the application. The layout of the database(s) have also been shown in order to depict how the data is stored. Lastly, many prototype screens, demonstrating the initial plans for the application have been provided.

10. References

[1] www.pluralsight.com. (2022). *Understanding React-Bootstrap and Making It Work for You*. [online] Available at: https://www.pluralsight.com/blog/software-development/react-bootstrap-explained#:~:text=React%2DBootstrap%20takes%20the%20CSS [Accessed 16 Feb. 2023].

[2] whatjackhasmade.co.uk. (n.d.). Jack Pritchard - Swansea - WhatJackHasMade. [online] Available at: https://whatjackhasmade.co.uk/component-driven-development/ [Accessed 17 Apr. 2023].

[3] Node.js. (2023). About. [online] Available at: https://nodejs.org/en/about.

[4] blog.openreplay.com. (2022). Sending emails from React with EmailJS. [online] Available at: https://blog.openreplay.com/sending-emails-from-react-with-emailjs/#:~:text=EmailJS%20is%20a%20JavaScript%20library.