Software Requirements Specification

for

<PetBuddy>

Version 1.0 approved

Prepared by <Long Ming Wei, Koh Jia Xin, Krystal Pek Ke Yun, Juannisa Putri Sunarya, Chan Zi Jian>

<SC2006 Group 59>

<16/11/2024>

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

This document specifies the software requirements for "PetBuddy" Version 1.0. It details the platform's functionalities, features, and limitations, offering a clear and organised framework for development. The scope of this Software Requirements Specification (SRS) encompasses the entire digital platform, including subsystems for managing pet owners and caretakers, coordinating bookings, and handling scheduling logistics. This SRS is crucial for directing the development, testing, and maintenance of the platform, ensuring the successful achievement of project objectives.

1.2 Document Conventions

To ensure consistency, these key conventions are followed:

- Font style: ‘Times New Roman’ for headers, ‘Arial’ for body text
- Font size: 11 for ‘Arial’, 12 for ‘Times New Romans
- Requirement prioritisation: To ensure the effective allocation of resources and focus on the most critical features, requirements in PetBuddy are prioritised as follows:
    - **High Priority**: Essential features that are critical for the system to function and meet core user needs.
    - **Medium Priority**: Important features that enhance the system but are not immediately critical.
    - **Low Priority**: Optional features that are desirable but can be deferred if resources are constrained.

1.3 Intended Audience and Reading Suggestions

The intended audience is mainly for future software developers making updates or fixing the app. Readers are encouraged to view the broad product scope and functions, use case diagram, class diagram, and state flow diagram and more to know the brief overview of the app’s features and flow, then look through the individual sequential diagrams and use case descriptions for individual pages and modules that are relevant to the portion of the app they are tasked to work on.

Additionally, other departments like marketing and HR can also look through just the broad product functions and perspective to promote the app better and provide better customer support or to hire the suitable people with the right skill sets to work on various features of the app.

1.4 Product Scope

PetBuddy is a digital platform designed to connect pet owners with caretakers who meet their scheduling needs. Its primary goal is to simplify the process of finding reliable pet care, while helping caretakers manage bookings efficiently. The platform allows pet owners to browse caretaker profiles, make bookings, and ensure clear communication, while caretakers can manage their schedules with ease. This solution supports the business strategy of offering a seamless, user friendly experience in the pet care sector, ultimately contributing to customer satisfaction and service excellence.

1.5 References

Eventbrite API Reference: https://www.eventbrite.com/platform/api

How to set up frontend with React and Next.js: https://nextjs.org/docs/app

How to set up a backend database with SQL and FastAPI: https://fastapi.tiangolo.com/tutorial/sql-databases/#create-database-tables-on-startup

2. Overall Description

2.1 Product Perspective

PetBuddy is a new product designed to connect pet owners and potential caretakers. It is not part of any product family nor any existing systems. The platform functions independently and also integrates with existing technologies via APIs. A system architecture diagram in Figure 1 is provided to illustrate the interconnections and external interfaces.

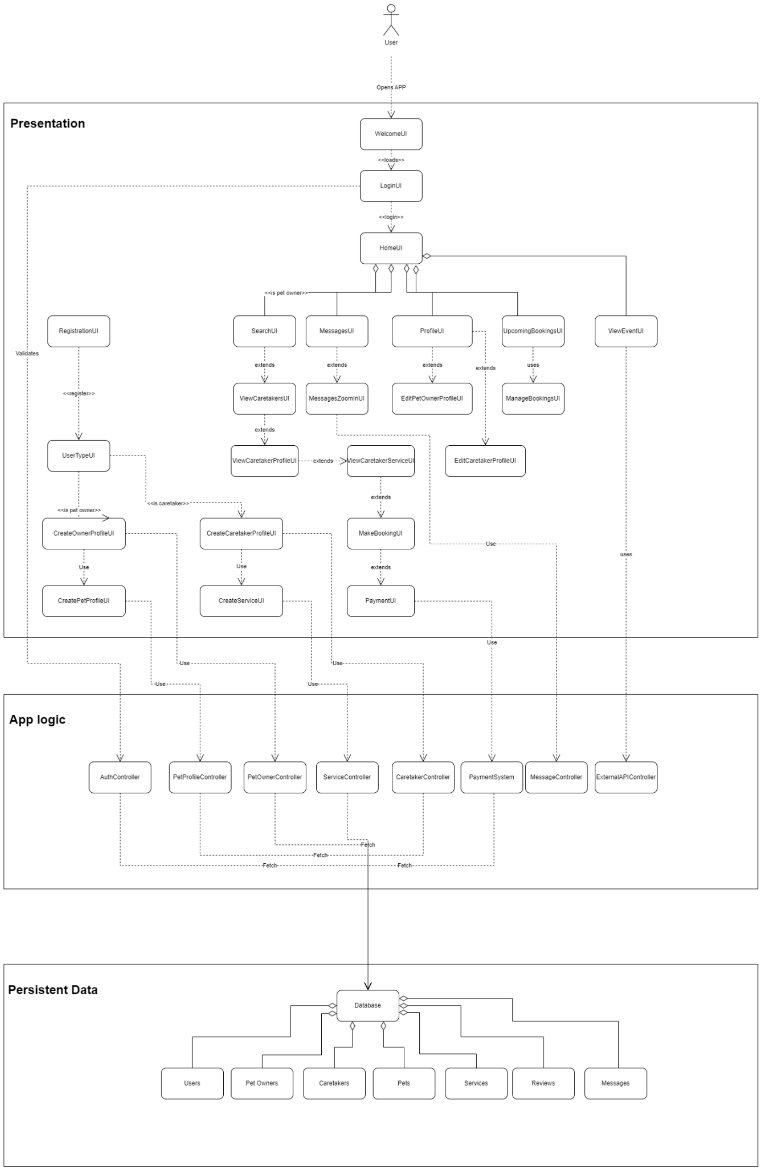


Figure 1: System Architecture

2.2 Product Functions

2.2.1 Use Case Diagram

The use case diagram in Figure 2 illustrates the key functionalities provided by PetBuddy. The system is designed for two primary user types and integrates with three APIs to deliver its features.

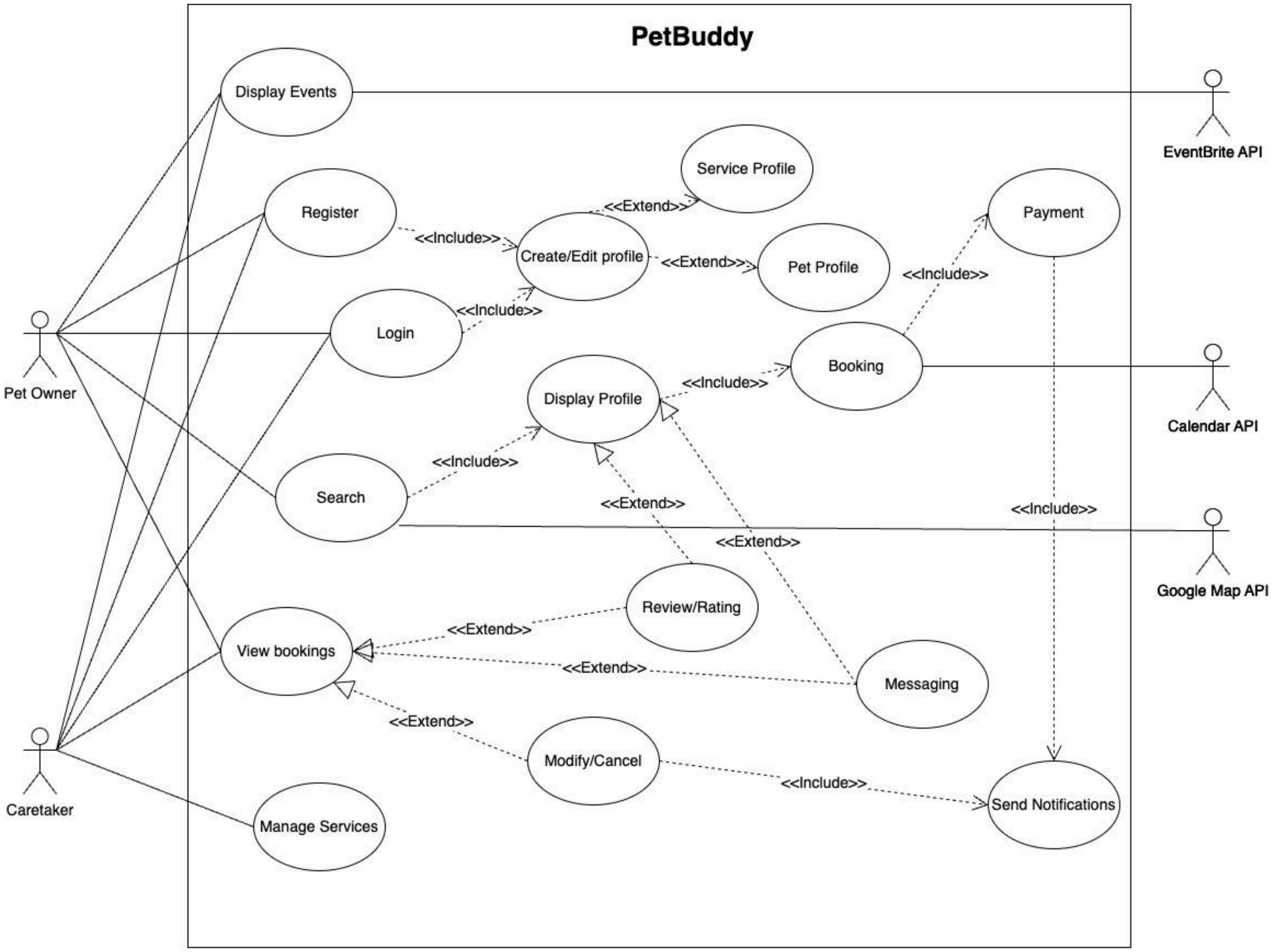


Figure 2: Use Case Diagram

2.2.2 Major Product Functions

PetBuddy allow users to perform the following functions:

1. **Authentication**
    - Signup
    - Login
    - Create Pet Owner/Caretaker Profile

2. **Pet Owner Functions**
    - Edit Pet Owner Profile
    - Create/Edit Pet Profile
    - Search for Caretakers
    - Filter Caretakers using Maps
    - Create/Edit/Delete Booking
    - Create Payment

3. **Caretaker Functions**
   - Edit Caretaker Profile
   - Create/Edit Service Profile

4. **Notifications**
   - Systemwide alerts and userspecific notifications

5. **Messaging**
   - Realtime Messaging between Pet Owners and Caretakers

6. **Other Functions**
   - Change Password
   - View Booking History
   - View Payment History
   - View Upcoming Bookings

2.2.3 Class Diagram

The Class Diagram (Figure 3) illustrates the interactions between various classes and outlines the key functionalities that PetBuddy must support.

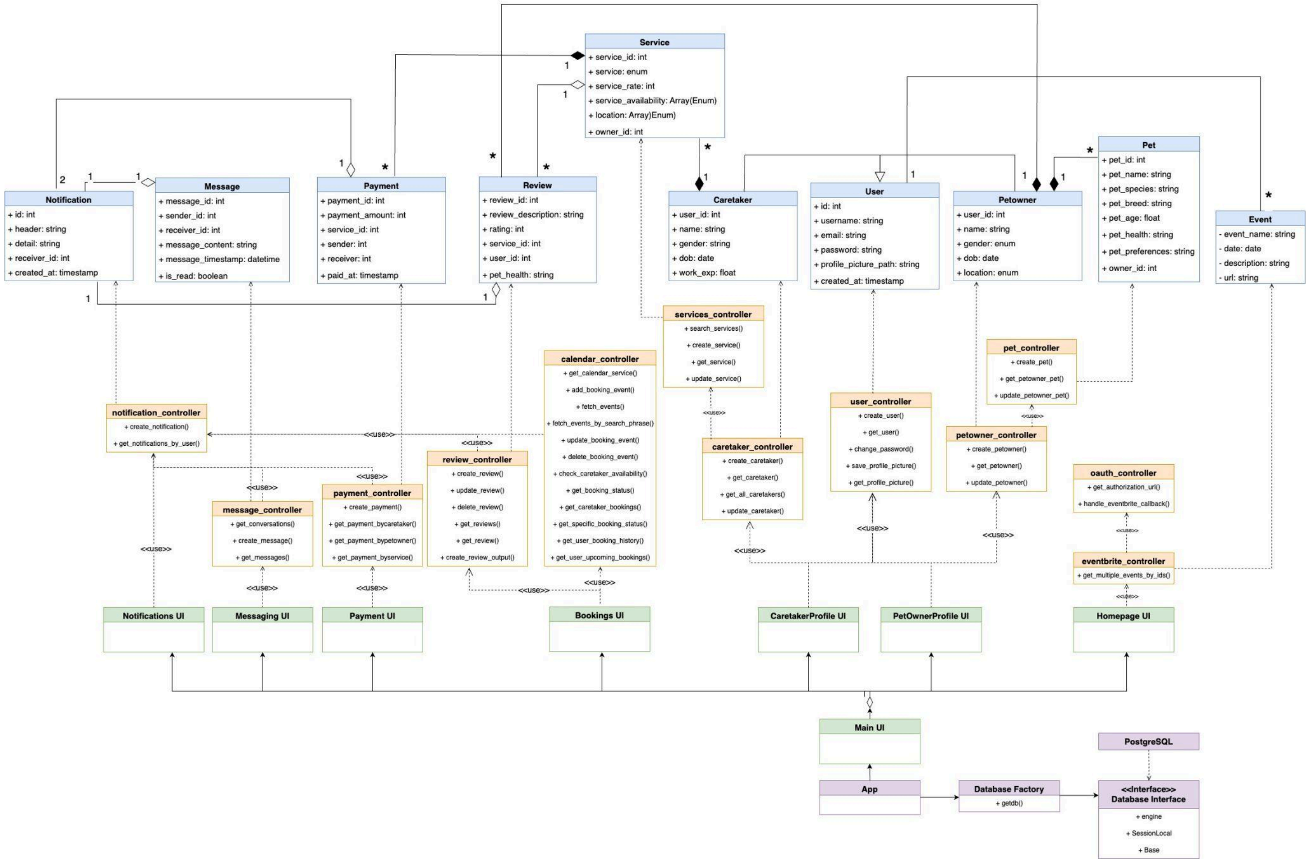


Figure 3: Class Diagram

2.3 User Classes and Characteristics

Users are classified into:

- Pet Owners: Individuals who need care services for their pets. They seek available caretakers to ensure their pets receive consistent care.
- Caretakers: Service providers who offer pet care services. They require an efficient system to manage multiple bookings and maintain clear communication with pet owners.

Both types of users are essential for PetBuddy's ecosystem to function effectively. For example, pet owners submit care requests based on their scheduling needs, while caretakers manage these requests and provide the necessary pet care services, creating a seamless connection between those seeking and providing pet care services in Singapore.

2.4 Operating Environment

The PetBuddy application will be webbased, designed for optimal use on both mobile and desktop devices (to be viewed in Light Mode). It will support the latest versions of Chrome, Safari and Edge browsers. It will also integrate seamlessly with external APIs, including Eventbrite, Google Calendar and Google Map, as well as utilise SQL-based database storage for data management, specifically PostgreSQL. These integrations and hosting solutions are chosen to ensure smooth functionality and adaptability across various devices and platforms.

2.5 Design and Implementation Constraints

2.5.1 Frontend Constraints

2.5.1.1 React Components

While React's component-based approach offers flexibility and reusability, it can also lead to challenges in managing state and data flow across complex applications. Developers need to carefully design component hierarchies and organise relationships between certain states, props, components and pages well to avoid complexity and maintainability issues.

2.5.1.2 External APIs

The use of external APIs (Eventbrite, Google Calendar, Google Maps) means the performance of the app depends on the availability and performance of these services which can be out of the developers' control. Developers need to handle potential errors and inconsistencies in API responses and ensure that the integration is seamless and user friendly.

2.5.2 Backend Constraints

The backend server is developed using FastAPI (in python), and uses PostgreSQL as the database.

2.5.2.1 Database Management

Since PostgreSQL is used for the app's backend, the creation and management of data structures, such as tables and columns, must be manually coded. Developers are responsible for writing the queries to retrieve and modify the data.

2.5.2.2 Structured vs. Unstructured Data

PostgreSQL is optimised for structured data, which may present challenges when dealing with unstructured data types (e.g. audio or video). Should the platform require future features such as messaging attachments or embedded media in pet owner/caretaker profiles, additional solutions may be needed to handle these unstructured data types effectively.

2.5.3 General Constraints

2.5.3.1 Hosting

- PetBuddy will be developed and run in a local environment, utilising local servers and databases for testing and development.
- This choice introduces limitations regarding scalability and data storage as the platform will not be making use of cloud for flexibility. As the platform grows, careful management of resources and server capacity will be essential to accommodate increasing user traffic and data without the flexibility of cloud infrastructure. Hence the need to ensure that the platform is optimised for performance even with these constraints.

2.5.3.2 Integration with External APIs

- Eventbrite API Integration: The platform will integrate with the Eventbrite API to provide users with event management functionalities. When users click on an event to view more details or make a booking, they will be redirected to the corresponding Eventbrite page. This integration must be seamless to ensure a smooth transition from the PetBuddy platform to Eventbrite, maintaining a cohesive user experience throughout. The platform should handle this redirection without delays and ensure that booking details and event information are easily accessible to users.

- Google Calendar API Integration: The PetBuddy platform will integrate with the Google Calendar API to allow users to manage their appointments with caretakers directly within the platform. Users will be able to view, schedule, and sync appointments with their Google Calendar. Clicking on specific event details will redirect users to their Google Calendar for further scheduling or updates. The integration must ensure accurate synchronisation of appointment data between the PetBuddy platform and Google Calendar, maintaining consistency and reliability in scheduling across both systems.

- Google Maps API Integration: The platform will use the Google Maps API to display an interactive map directly within the PetBuddy webpage. Caretakers' locations will be marked on the map, allowing users to visually find nearby caretakers. Users will be able to zoom in, zoom out, and click on specific caretaker markers to view more details. The integration must be responsive across both desktop and mobile devices, ensuring a smooth and intuitive user experience. Additionally, the API integration should comply with Google's usage policies, including proper security measures for API key management and staying within usage limits.

2.5.3.3 Security Considerations

- The platform will implement secure user authentication and password hashing using bcrypt. Sensitive user data will be handled according to best practices in security to prevent unauthorised access, where account passwords will be encrypted and stored in the database which is in compliance with local data protection regulations to prevent unauthorised access.

2.6 User Documentation

PetBuddy will provide a complete set of user documentation to ensure developers have a clear understanding and can easily use the platform. The documentation will include the following components:

2.6.1 Read me files

2.6.1.1 Main Repository

A README file for the main repository will offer an overview of the project, covering the architecture, setup instructions for the development environment, deployment guidelines, and a highlevel summary of the repository's contents.

2.6.1.2 Frontend Repository

The frontend repository will have its own README file detailing the setup process, the structure of the frontend application, guidelines for using React.js, TypeScript, and TailwindCSS within the project, and instructions for running and building the application.

2.6.1.3 Backend Repository

The backend repository will include a README file with instructions for setting up the FastAPI server, configuring Python environments, managing dependencies, and details on SQL database integration.

2.6.2 Code comments

Presence of comments for code statements and function blocks to explain its purpose.

2.7 Assumptions and Dependencies

2.7.1 Assumptions

Given functionalities like viewing events from Eventbrite and messaging, it is assumed that the users are always connected to the internet.

2.7.2 Dependencies

2.7.2.1 Frontend Libraries

- React.js: Core framework for building the user interface
- Next.js: For routing and server side rendering
- TypeScript: For type safety
- Axios: For making HTTP requests to the backend
- TailwindCSS: For styling
- reactdatepicker: For the date and time selection interface
- momenttimezone: For handling timezones and date/time formatting across different timezones
- @vis.gl/reactgooglemaps: React wrapper for Google Maps API v3
- @googlemaps/markerclusterer: For grouping markers together on the map

2.7.2.2 Backend Libraries

- fastapi: Main web framework for your API
- uvicorn: Backend server to run FastAPI

- sqlalchemy: For database ORM
- pythonjose: For JWT token handling
- bcrypt: For password hashing
- pydantic: For data validation
- pythonmultipart: For handling form data
- pytz: For timezone handling
- websockets: For WebSocket protocol implementation
- httpx: To provide a more reliable http

2.7.2.3 External services

APIs used in PetBuddy:

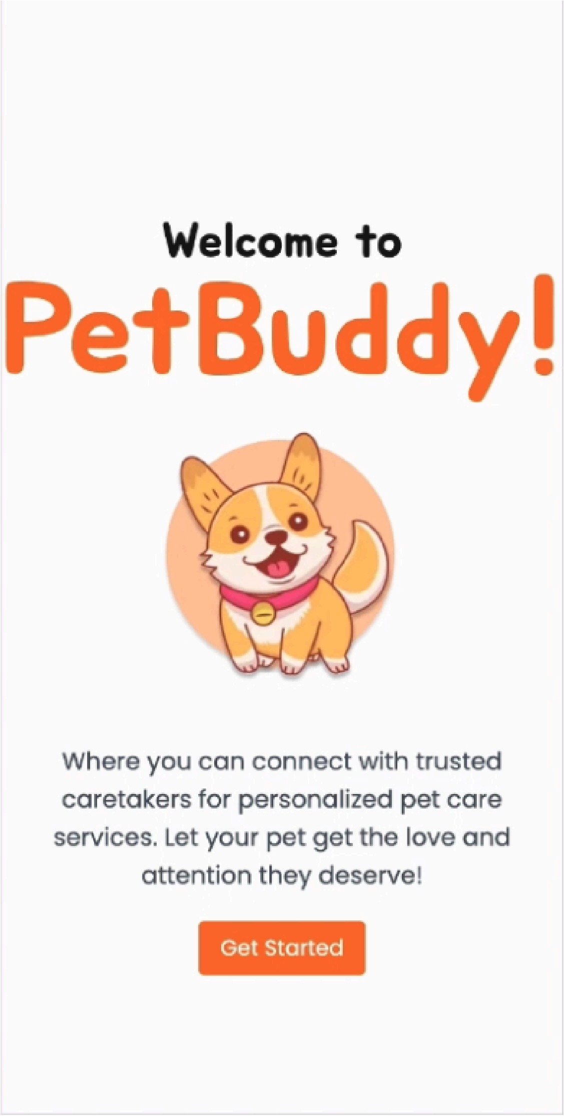
- EventBrite API: To retrieve events for pets
- Google Calendar API: To facilitate bookings
- Google Maps API: To display the caretaker locations

3. External Interface Requirements

3.1 User Interfaces

3.1.1 App Displays

3.1.1.1 Welcome page

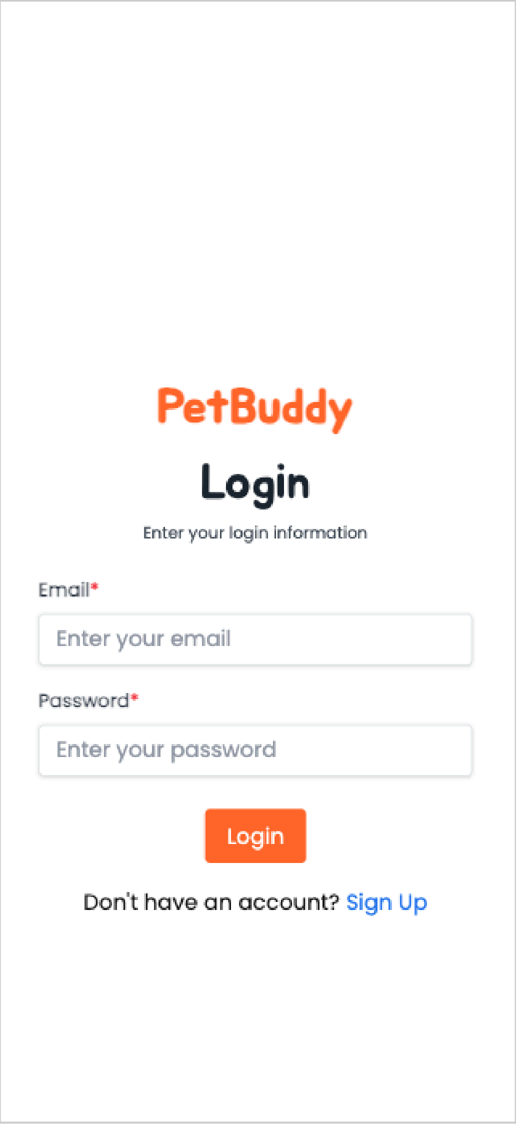


Welcome to

PetBuddy!

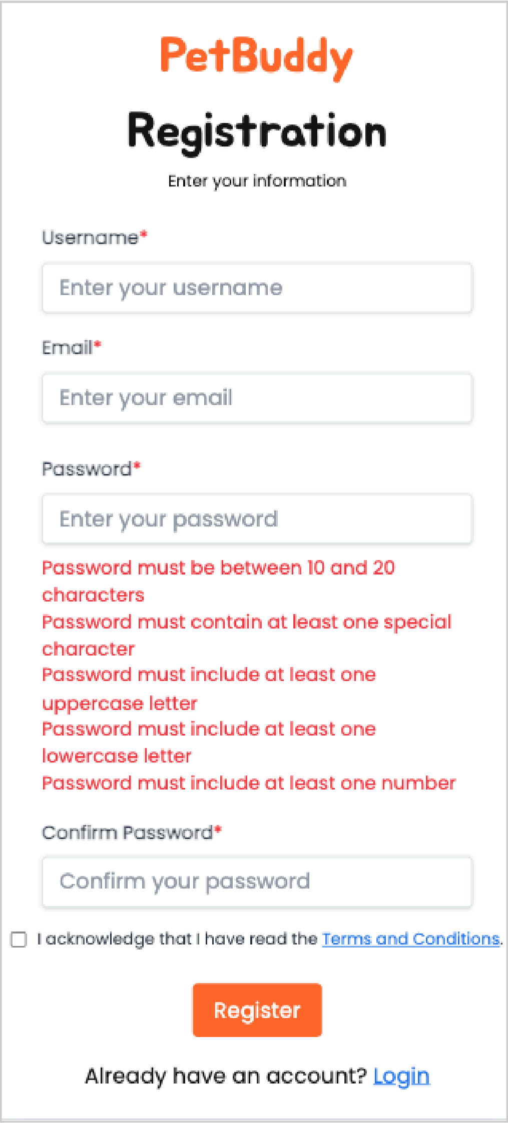
Where you can connect with trusted caretakers for personalized pet care services. Let your pet get the love and attention they deserve!

3.1.1.2 Login + Sign Up



Enter your login information

Don't have an account? Sign Up



Enter your information

Password must be between 10 and 20 characters

Password must contain at least one special character

Password must include at least one uppercase letter

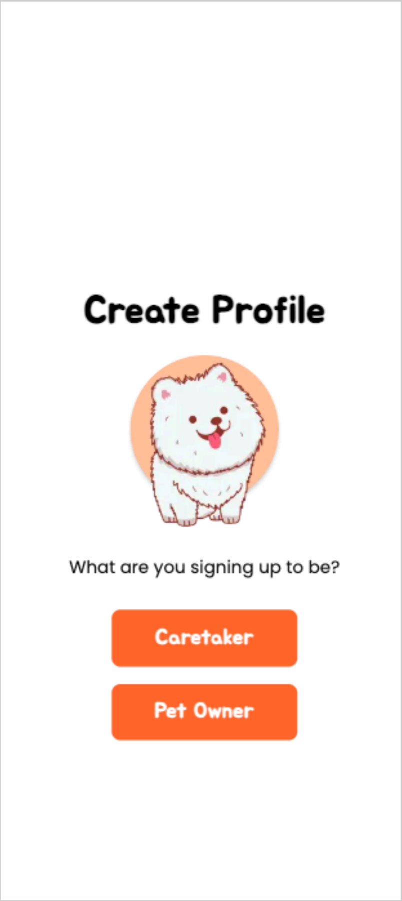
Password must include at least one lowercase letter

Password must include at least one number

□ I acknowledge that I have read the Terms and Conditions.

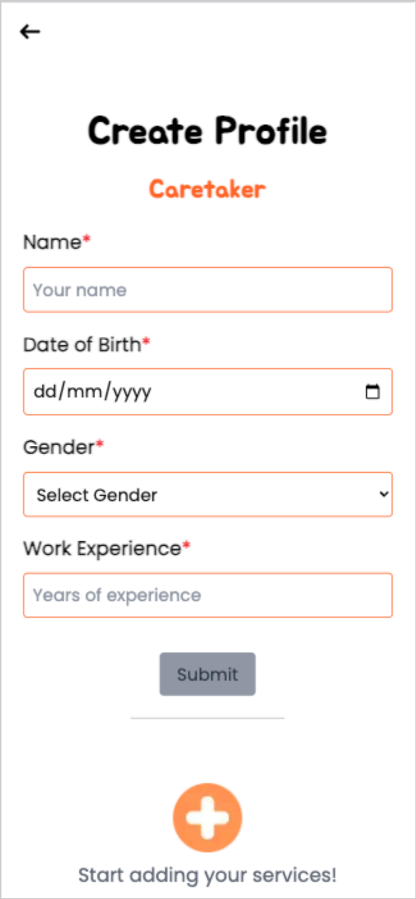
Already have an account? Login

3.1.1.3 Profile Creation page (Caretaker + Pet Owner)



Create Profile

What are you signing up to be?



Create Profile

Caretaker

Name*

Your name

Date of Birth*

dd/mm/yyyy

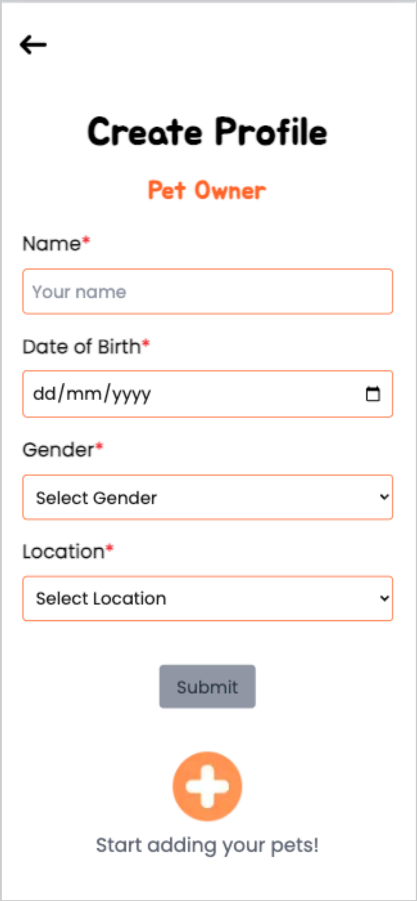
Gender*

Select Gender

Work Experience*

Years of experience

Start adding your services!



Create Profile

Pet Owner

Name*

Your name

Date of Birth*

dd/mm/yyyy

Gender*

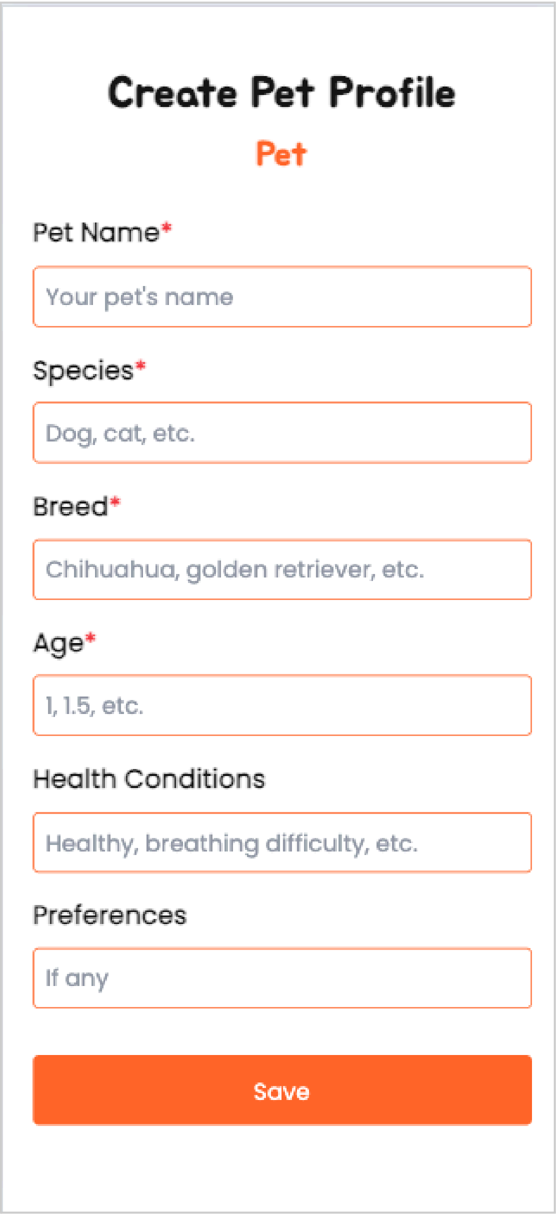
Select Gender

Location*

Select Location

Start adding your pets!

3.1.1.4 Service Creation Page (Caretaker) + Pet Profile Creation Page (Pet Owner)



Create Pet Profile

Pet

Pet Name*

Your pet's name

Species*

Dog, cat, etc.

Breed*

Chihuahua, golden retriever, etc.

Age*

1, 1.5, etc.

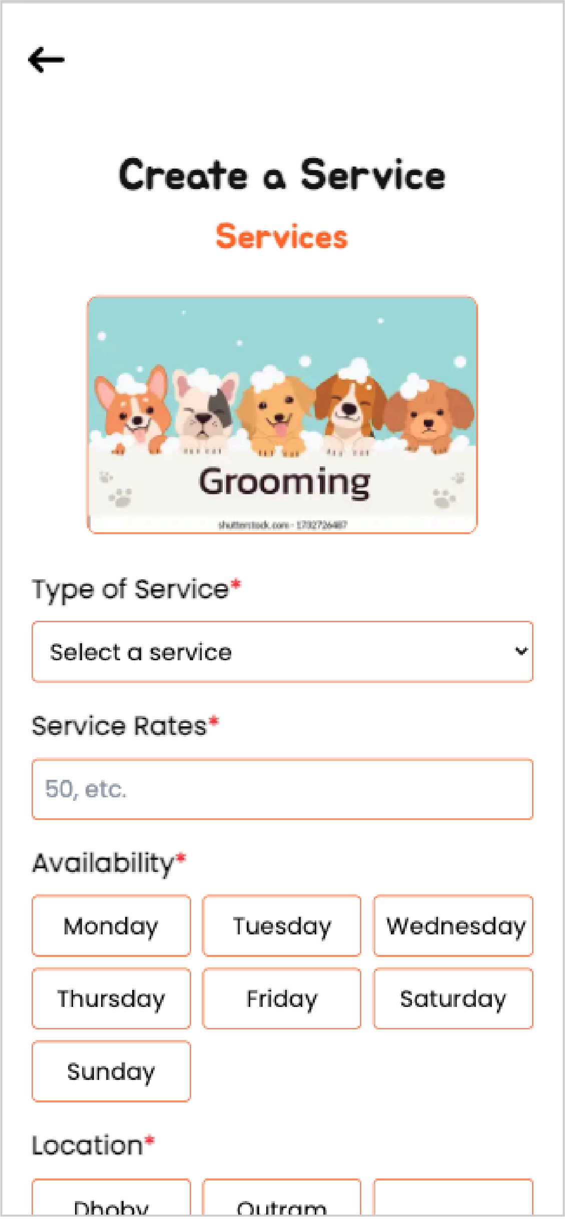
Health Conditions

Healthy, breathing difficulty, etc.

Preferences

If any

Save



←

Create a Service

Services

Grooming

shutterstock.com · 1732726487

Type of Service*

Select a service

Service Rates*

50, etc.

Availability*

Monday Tuesday Wednesday

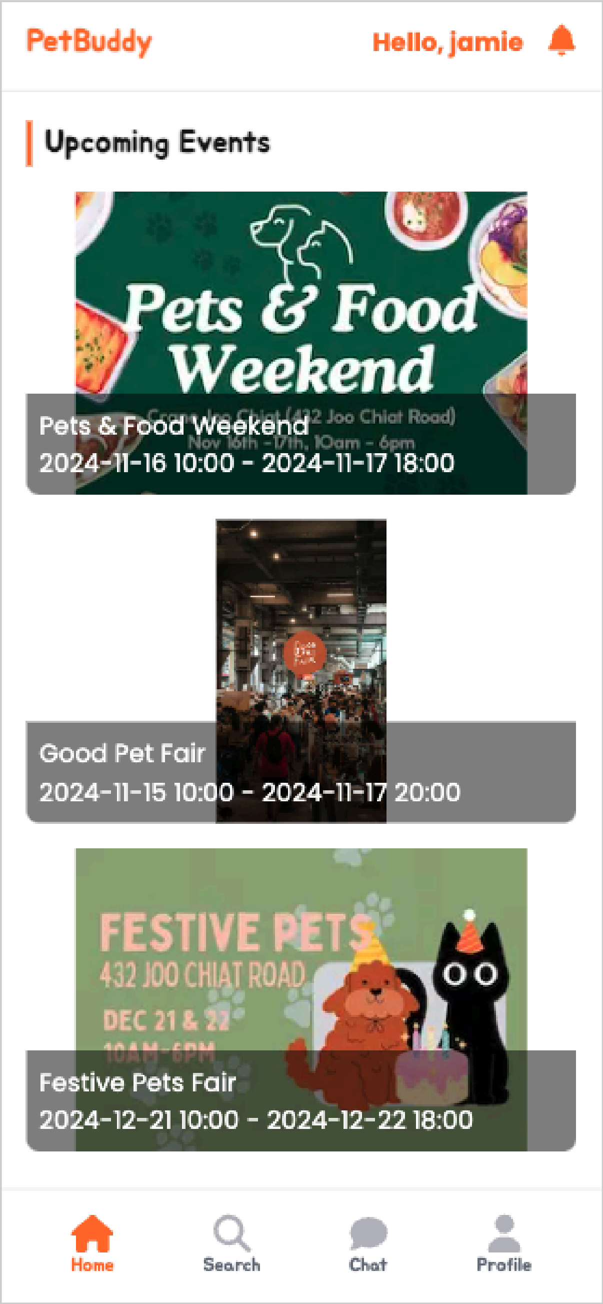
Thursday Friday Saturday

Sunday

Location*

Dhoby Outram

3.1.1.5 Homepage



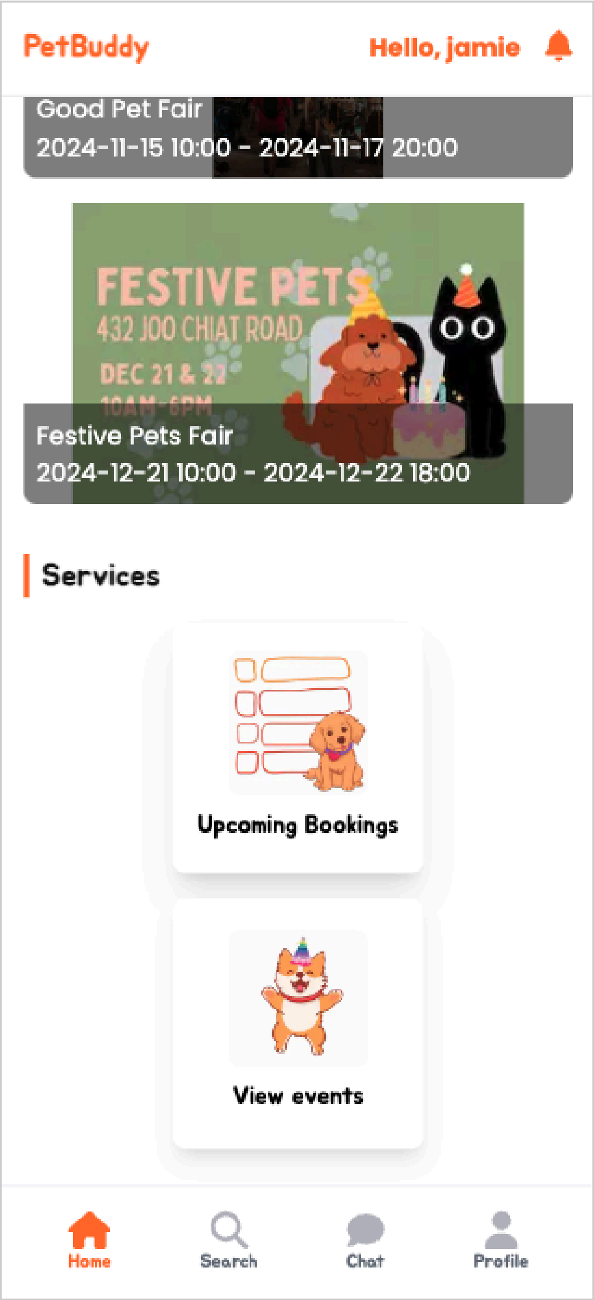
PetBuddy Hello, jamie

Upcoming Events

Pets & Food Weekend
2024-11-16 10:00 - 2024-11-17 18:00

Good Pet Fair
2024-11-15 10:00 - 2024-11-17 20:00

Festive Pets Fair
2024-12-21 10:00 - 2024-12-22 18:00



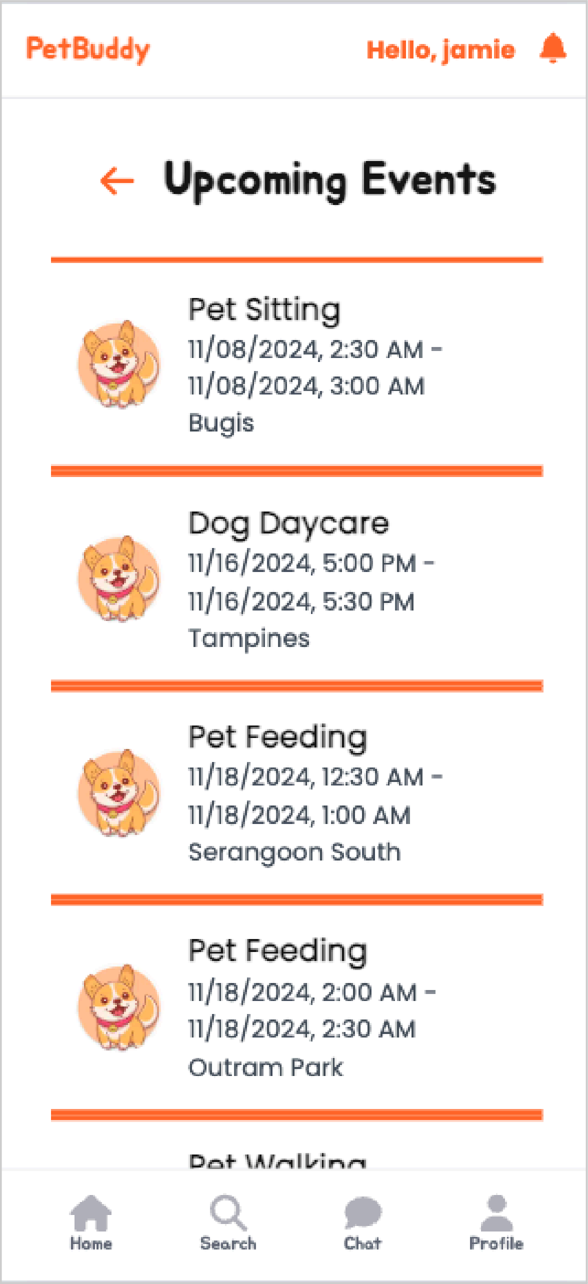
PetBuddy Hello, jamie

Good Pet Fair
2024-11-15 10:00 - 2024-11-17 20:00

Festive Pets Fair
2024-12-21 10:00 - 2024-12-22 18:00

Services

3.1.1.6 Upcoming Bookings + View Events



Upcoming Events

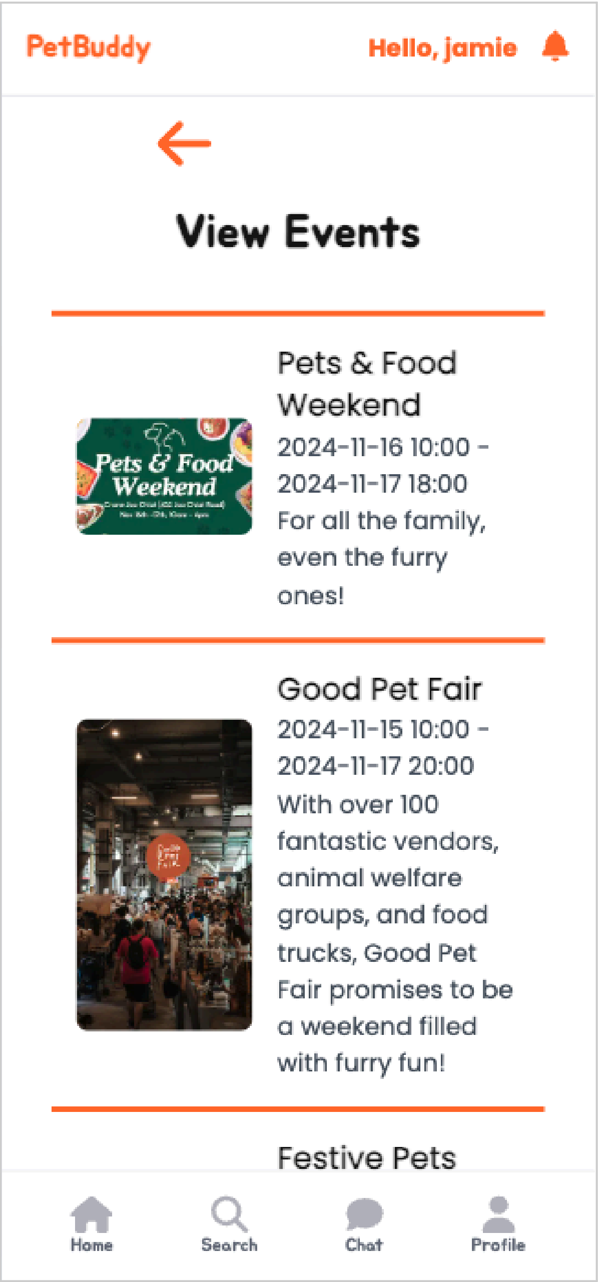
Pet Sitting
11/08/2024, 2:30 AM -
11/08/2024, 3:00 AM
Bugis

Dog Daycare
11/16/2024, 5:00 PM -
11/16/2024, 5:30 PM
Tampines

Pet Feeding
11/18/2024, 12:30 AM -
11/18/2024, 1:00 AM
Serangoon South

Pet Feeding
11/18/2024, 2:00 AM -
11/18/2024, 2:30 AM
Outram Park

Pet Walking



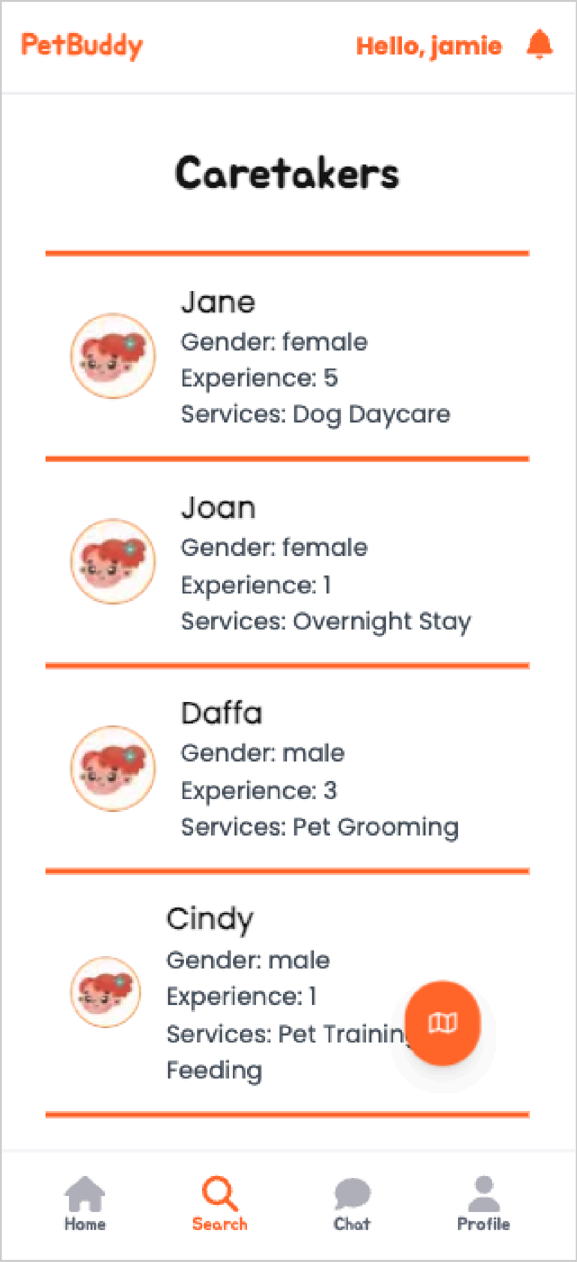
View Events

Pets & Food Weekend
2024-11-16 10:00 -
2024-11-17 18:00
For all the family, even the furry ones!

Good Pet Fair
2024-11-15 10:00 -
2024-11-17 20:00
With over 100 fantastic vendors, animal welfare groups, and food trucks, Good Pet Fair promises to be a weekend filled with furry fun!

Festive Pets

3.1.1.7 Search + Map to Find Caretakers



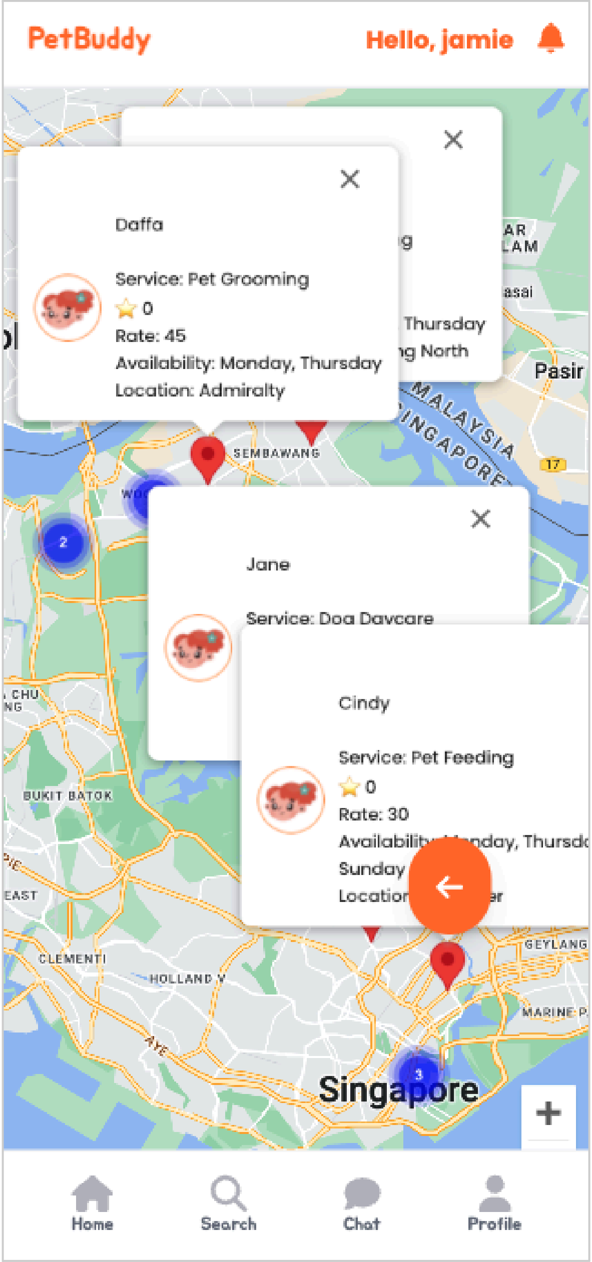
Caretakers

Jane
Gender: female
Experience: 5
Services: Dog Daycare

Joan
Gender: female
Experience: 1
Services: Overnight Stay

Daffa
Gender: male
Experience: 3
Services: Pet Grooming

Cindy
Gender: male
Experience: 1
Services: Pet Trainin
Feeding



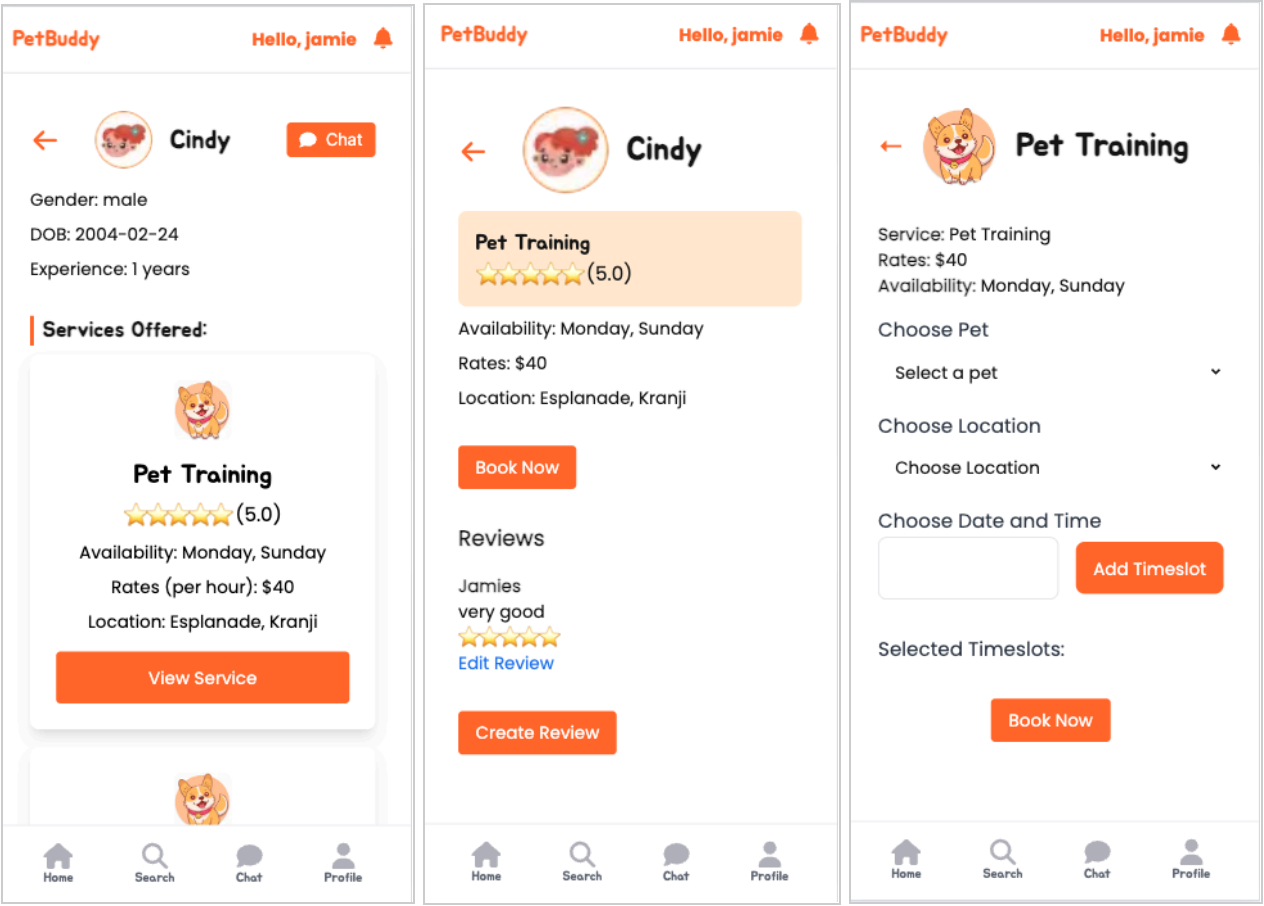
Daffa
Service: Pet Grooming
⭐ 0
Rate: 45
Availability: Monday, Thursday
Location: Admiralty

g
Thursday
g North

Jane
Service: Doa Davcare

Cindy
Service: Pet Feeding
⭐ 0
Rate: 30
Availability: Monday, Thursda
Sunday
Locatior er

3.1.1.8 Caretaker Profile + Services + Booking Page



Gender: male

DOB: 2004-02-24

Experience: 1 years

Services Offered:

Pet Training

☆☆☆☆☆ (5.0)

Availability: Monday, Sunday

Rates (per hour): $40

Location: Esplanade, Kranji

Pet Training

☆☆☆☆☆ (5.0)

Availability: Monday, Sunday

Rates: $40

Location: Esplanade, Kranji

Reviews

Jamies

very good

☆☆☆☆☆

Service: Pet Training

Rates: $40

Availability: Monday, Sunday

Choose Pet

Select a pet

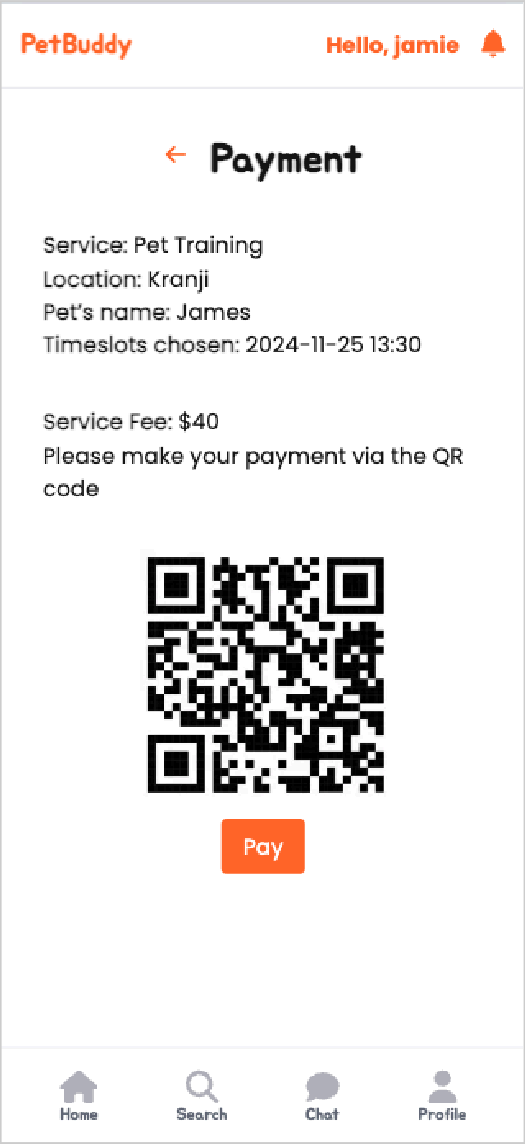
Choose Location

Choose Location

Choose Date and Time

Selected Timeslots:

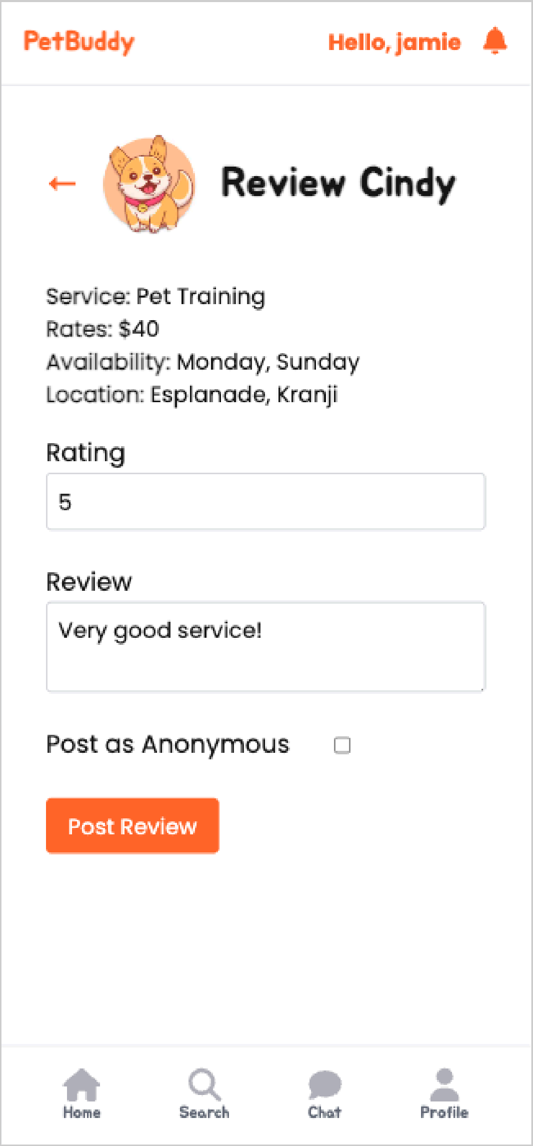
3.1.1.9 Payment Page + Review



Payment

Service: Pet Training
Location: Kranji
Pet's name: James
Timeslots chosen: 2024-11-25 13:30

Service Fee: $40
Please make your payment via the QR code



Review Cindy

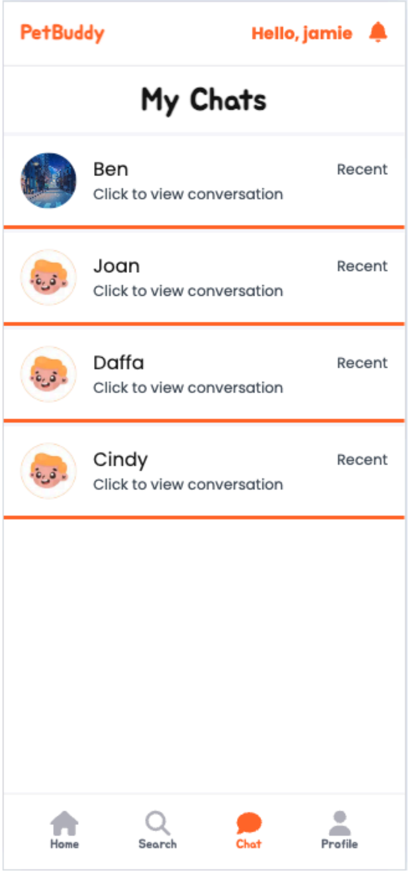
Service: Pet Training
Rates: $40
Availability: Monday, Sunday
Location: Esplanade, Kranji

Rating
5

Review
Very good service!

Post as Anonymous □

3.1.1.10 Messages + Chat Room + Notification



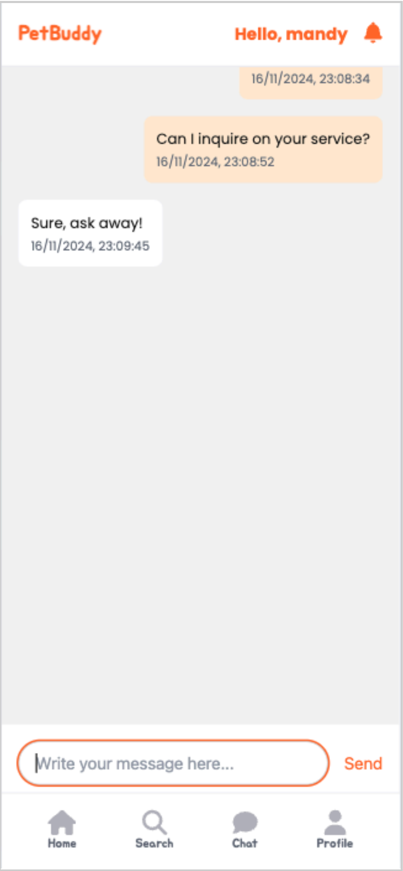
My Chats

Ben Recent
Click to view conversation

Joan Recent
Click to view conversation

Daffa Recent
Click to view conversation

Cindy Recent
Click to view conversation

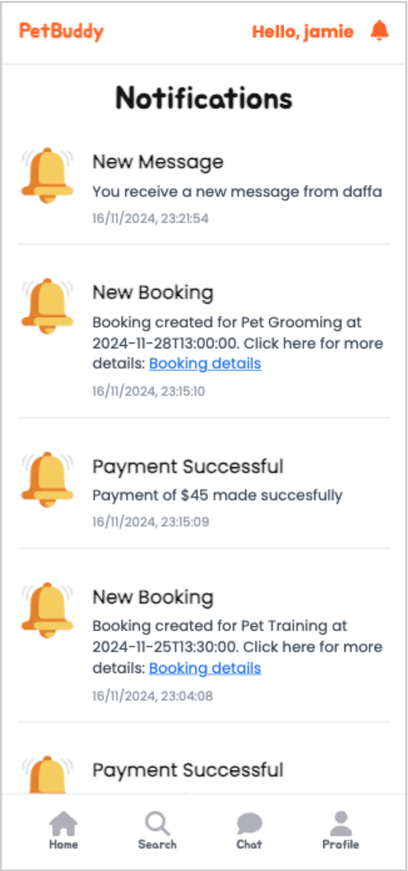


16/11/2024, 23:08:34

Can I inquire on your service?
16/11/2024, 23:08:52

Sure, ask away!
16/11/2024, 23:09:45

Write your message here...



Notifications

New Message
You receive a new message from daffa
16/11/2024, 23:21:54

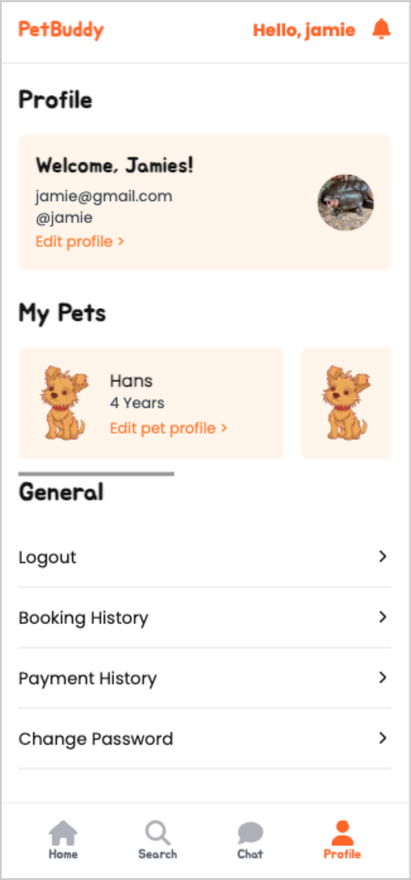
New Booking
Booking created for Pet Grooming at 2024-11-28T13:00:00. Click here for more details: Booking details
16/11/2024, 23:15:10

Payment Successful
Payment of $45 made succesfully
16/11/2024, 23:15:09

New Booking
Booking created for Pet Training at 2024-11-25T13:30:00. Click here for more details: Booking details
16/11/2024, 23:04:08

Payment Successful

3.1.1.11 Profile (Caretaker + Pet Owner) + Edit Profile



Profile

Welcome, Jamies!
jamie@gmail.com
@jamie
Edit profile >

My Pets

Hans
4 Years
Edit pet profile >

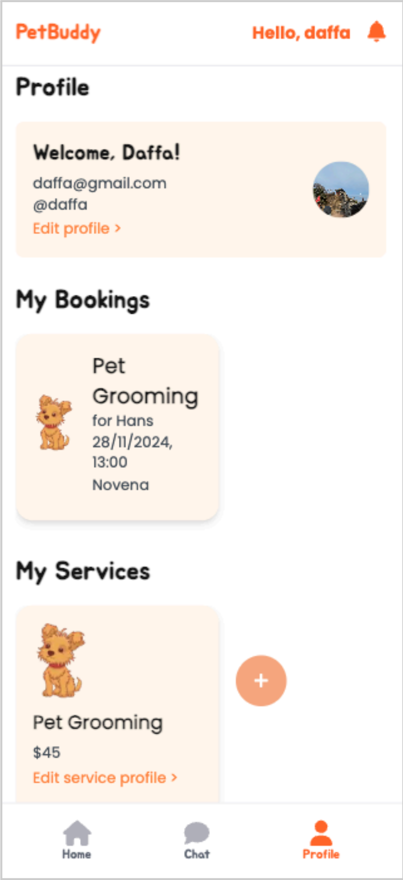
General

Logout >

Booking History >

Payment History >

Change Password >



Profile

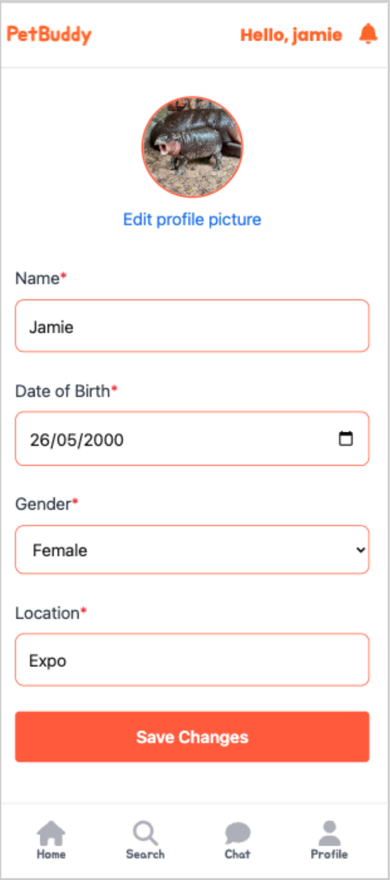
Welcome, Daffa!
daffa@gmail.com
@daffa
Edit profile >

My Bookings

Pet Grooming
for Hans
28/11/2024,
13:00
Novena

My Services

Pet Grooming
$45
Edit service profile >



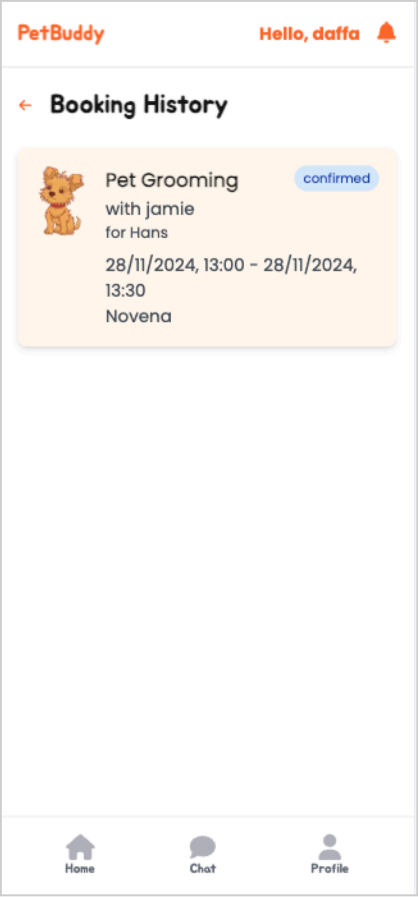
Name* Jamie

Date of Birth* 26/05/2000

Gender* Female

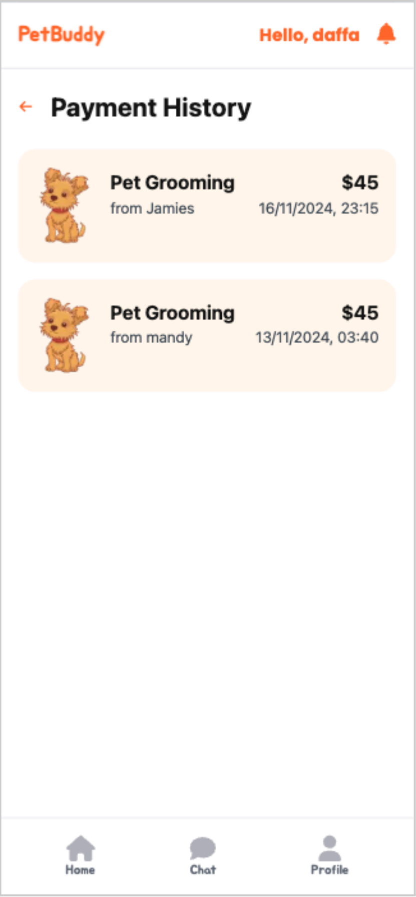
Location* Expo

3.1.1.12Booking History + Payment History + Change Password



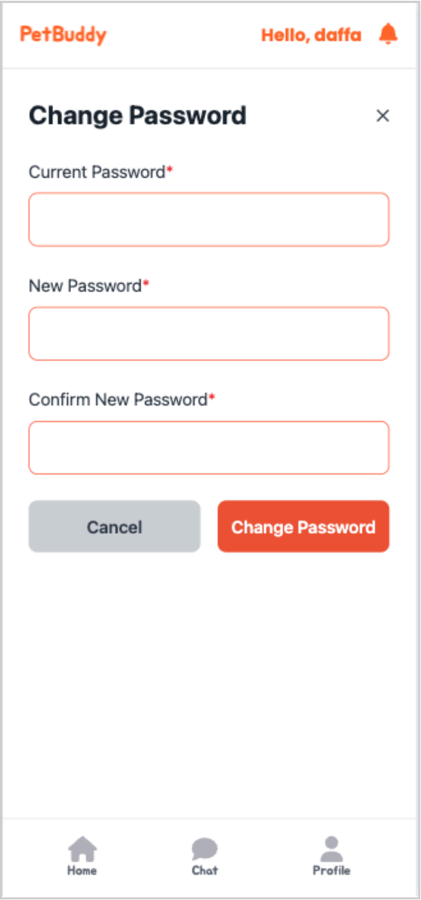
← Booking History

Pet Grooming confirmed
with jamie
for Hans
28/11/2024, 13:00 - 28/11/2024, 13:30
Novena



← Payment History

Pet Grooming	$45
from Jamies	16/11/2024, 23:15
Pet Grooming	$45
from mandy	13/11/2024, 03:40



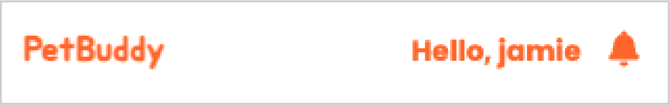
Change Password

Current Password*

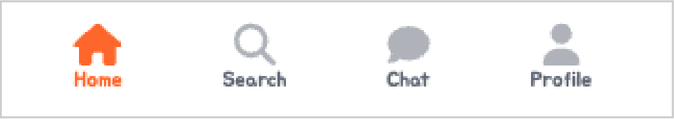
New Password*

Confirm New Password*

3.1.2 Components



Header



Menu Bar

3.2 Hardware Interfaces

The app supports both mobile and PC devices that have internet access as the web based app uses dynamic layouts. Older devices that use older operating systems are compatible too as long as device and web browser performance is decent.

3.3 Software Interfaces

PostgreSQL is used for storing the backend databases, and has several tables, namely; *caretakers, messages, notifications, payment, petowners, pets, reviews, services and users*. These tables are relational as they have foreign keys to link to each other, such as *petowners* and *caretakers* sharing user_id with *users* to identify user profile types. It can be shared across the entire app and its userbase. Controllers and routers are used to allow the app to view, process and edit data in the appropriate data tables in the SQL server.

External APIs like Google Maps, Google Calendar and Eventbrite are also used to facilitate visualisation of caretaker locations, managing bookings and displaying pet related events respectively, with Google Calendar in particular having its own SQL router to update bookings in the appropriate users' calendars. The frontend can also display pages dynamically based on the most updated data thanks to React's dynamic component based architecture and Next.js' dynamic routing.

3.4 Communications Interfaces

PetBuddy's communication infrastructure is built on multiple protocols and frameworks. They are:

- The system uses **HTTP/HTTPS** protocol for RESTful API communication, implemented using FastAPI for the backend and Next.js for the frontend. All sensitive data transmission is secured through HTTPS encryption in production, with authentication handled via JWT (JSON Web Tokens) and password securely hashed by using bcrypt.

- For realtime communication in the messaging feature, the system implements **WebSocket** protocol. This enables instant message delivery and realtime updates between users. The WebSocket implementation is handled through FastAPI's Websocket support, managing connections, user presence and message delivery.

- The system integrated with **Google Calendar API** for booking management. This integration uses HTTPS protocol and OAuth 2.0 for authentication, allowing the system to create, manage and synchronise booking events.

- Error handling follows a standardised approach across the application. All errors return appropriate HTTP status codes along with descriptive error messages.

- The application implements comprehensive security measures. All private endpoints require authentication to ensure only authorised users can access them. Although the system includes a CORS configuration, it currently allows requests from all origins () to facilitate broad accessibility. This configuration ensures compatibility with various clients but could be refined in future iterations to restrict access to specific trusted domains for enhanced security.

4. System Features

Please refer to the Use Case Diagram in 2.2.1 for an overview of the use cases and its function.

4.1 Authentication

4.1.1 Description and Priority

The **SignUp** feature allows new users, including Pet Owners and Caretakers, to create accounts on the PetBuddy platform. This feature ensures users have personalised access to the application's functionalities by securely storing their credentials and validating input criteria.

Priority: High

- Benefit: 9 (essential for onboarding new users and expanding the user base).
- Penalty: 8 (without it, users cannot access the platform's features).
- Cost: 5 (moderate implementation complexity due to validation and database interaction).
- Risk: 6 (potential risks include handling sensitive user data and enforcing strong password policies).

4.1.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The user enters their username, email, password, confirm password and checks terms and condition box.
Response: If password meets requirement, on screen text will turn green.

b. Stimulus: The user clicks the Register button.
Response:
    - The system validates the inputs and checks if such data exists in the database.
    - If valid, the user is redirected to the login screen.
    - If invalid, an error message is displayed: "Failed to register. Please try again"

Alternative Flow:

- AF1:
    - Stimulus: The user clicks the Login hyperlink.
      Response: The system redirects the user to the Login page.

Exception Flow:

- Stimulus: User left username or email or password or confirm password or terms and condition fields blank:
  Response: An error message prompts the user to complete the missing fields.
- Stimulus: User entered an email that has already been registered,
  Response: An error message is displayed:"Email already used"
- Stimulus: User entered a username that is not unique,
  Response: An error message is displayed:"Username already exists"
- Stimulus: User entered confirm password and it does not match password,
  Response: An error message is displayed: "Passwords do not match"

4.1.3 Functional Requirements

REQ1:The system must have a registration process to create accounts for new users

- REQ1.1: The system must require users to enter a unique username
    - REQ1.1.1: If the username is already taken, app must display error message
- REQ1.2: The system must require users to enter a unique email
    - REQ1.2.1: If email is already used, app must display error message
    - REQ1.2.2: If email is invalid, app must display error message
- REQ1.3: The system must require users to create a password for their account
    - REQ1.3.1: The system must ensure password meets requirements
        - REQ1.3.1.1: Password must be at least 10 characters long
        - REQ1.3.1.2: Password must contain at least 1 special character
        - REQ1.3.1.3: Password must include at least 1 uppercase character
        - REQ1.3.1.4: Password must include at least 1 lowercase character
        - REQ1.3.1.5: Password must include at least 1 number
        - REQ1.3.1.6: Password must not exceed 20 characters long
        - REQ1.3.1.7: If password do not meet any of the requirements, an error message must pop up below to indicate which requirements are not fulfilled
        - REQ1.3.1.8: Password must be masked with dots by default
    - REQ1.3.2: The system should require users to reenter their password for confirmation immediately after the initial entry
        - REQ1.3.2.1: Password must be masked with dots
        - REQ1.3.2.2: If the password entered in this step does not match the password entered previously, an error message must pop up to indicate that the passwords do not match

4.2 Login

4.2.1 Description and Priority

The **Login** feature allows users, including Pet Owners and Caretakers, to authenticate themselves using their registered accounts. This feature is critical for securing access to application functionalities.

Priority: High

- Benefit: 9 (ensures secure access)
- Penalty: 8 (without it, users cannot access most features)
- Cost: 4 (implementation is straightforward with existing libraries)
- Risk: 5 (potential risks with handling user credentials securely

4.2.2 Stimulus/Response Sequences

Main Flow:

c. Stimulus: The user enters their username or email.
   Response: The system validates the input format (e.g., email structure).

d. Stimulus: The user enters their password (which is masked by default)
   Response: The system ensures the password field is populated correctly.

e. Stimulus: The user clicks the Login button.
   Response:
   - The system validates the credentials against the database.
   - If valid, the user is logged in and redirected to the dashboard.
   - If invalid, an error message is displayed: “Login failed. Username and password do not match.”

Alternative Flow:

- AF1:
  - Stimulus: The user clicks the Sign Up hyperlink.
    Response: The system redirects the user to the Register page.

Exception Flow:

- Stimulus: User left username or password fields blank:
  Response: An error message prompts the user to complete the missing fields.

4.2.3 Functional Requirements

REQ2.1: The system must validate user credentials against the database to ensure they match an existing account.
REQ2.2: The system must provide an error message when login fails, with the text: "Invalid credentials or server error"
REQ2.3: The password input must be masked by default
REQ2.4: The system must redirect users to the homepage upon successful login.
REQ2.5: The system must support login using an email address.

REQ2.6: The system must redirect the user to the Register page if the “Sign Up” hyperlink is clicked.
REQ2.7: The system must display a prompt if the username or password field is left blank.
REQ2.8: The system must support retry attempts if login fails.

4.3 Create User Profile

4.3.1 Description and Priority

The Create User Profile feature is a critical part of onboarding users after they have created an account. After a successful registration and login, the system checks if a user profile exists. If a profile is missing, the user is redirected to the Create User Profile page, where they must specify their role. This choice is permanent and cannot be altered later.

- Pet Owners will need to provide their name, date of birth, gender, and location.
- Caretakers will need to provide their name, date of birth, gender, and work experience.

This step is vital for tailoring the user experience based on the selected role and is necessary for features such as service bookings, profile management, and personalised interactions with the system.

Priority: High

- Benefit: 9 (Key feature for the app's functionality, enabling access to rolespecific features like bookings).
- Penalty: 8 (Without profiles, users cannot use many core features).
- Cost: 6 (Involves effort for frontend design, input validation, and backend data handling).
- Risk: 6 (Possible challenges with invalid data, user experience issues during profile creation).

4.3.2 Stimulus/Response Sequences

Main flow:

a. Stimulus: After completing the account creation process and logging in, the user attempts to access the app.
   Response: The system checks if a profile exists for the user.
   - If no profile exists, the user is redirected to the Create User Profile page.

b. Stimulus: The user is prompted to select their role (Pet Owner or Caretaker).
   Response: The system loads the appropriate profile creation form based on the user's selection.

   i. Stimulus (Pet Owner): The user enters their name, date of birth, gender, and location, then submits the form.
      Response:
      - The system validates the inputs for completeness and correctness.
      - If valid, the profile is saved in the database, and the user is redirected to the dashboard or homepage.

   ii. Stimulus (Caretaker): The user enters their name, date of birth, gender, and work experience, then submits the form.
      Response:
      - The system validates the inputs for completeness and correctness.
      - If valid, the profile is saved, and the user is redirected to the dashboard or homepage.

Exception Flow:

- Stimulus: The user provides incomplete or invalid data (e.g., missing fields, incorrect formats like an invalid date of birth).
  Response: The system displays an error message prompting the user to correct the input before proceeding with submission.
- Stimulus: The user attempts to select a future date for Date of Birth.
  Response: The system disables the selection of future dates in the date picker, ensuring the user cannot select a future date.

4.3.3 Functional Requirements

REQ3.1: The system must check if a user profile exists after login. If no profile exists, the system must redirect the user to the Create User Profile page.
REQ3.2: The system must present users with an option to choose between creating a Pet Owner or Caretaker profile. This selection must be permanent.

REQ3.3.1: For Pet Owners, the system must provide the following fields for profile creation:

- Name:
  - field must accept a nonempty string and validate that the name is within a reasonable length (At least 3 characters and no more than 30 characters)
- Date of Birth:
  - field must allow the user to input their birth date in a valid format (YYYYMMDD) and validate that the date is in the past and that the user meets the minimum age requirement (16 years old).
- Gender:
  - field must allow the user to select an option such as "Male", "Female", or "Prefer not to say"
- Location:
  - field must allow the user to select their location from a list of Singapore areas (Eg. Boonlay, Pioneer, Expo etc)
- Profile picture:
  - field must be optional
  - field must accept any png or jpg file

REQ3.3.2 For Caretakers, the system must provide the following fields for profile creation:

- Name:
  - field must accept a nonempty string and validate that the name is within a reasonable length (At least 3 characters and no more than 30 characters)
- Date of Birth:
  - field must allow the user to input their birth date in a valid format (YYYYMMDD) and validate that the date is in the past and that the user meets the minimum age requirement (16 years old).
- Gender:
  - field must allow the user to select an option such as "Male", "Female", or "Prefer not to say"
- Work Experience:
  - field must allow users to input a number and validate that the number of years the user input for work experience is less than the user's age which will be derived from the date of birth input
- Profile picture:
  - field must be optional
  - field must accept any png or jpg file

REQ3.4: The system must validate all user inputs for completeness and correct formatting. If any field is missing or contains incorrect data (e.g., invalid date format or empty required field), the system must prompt the user with an error message.

REQ3.5: After successful submission, the system must store the user profile in the database, ensuring data persistence.

REQ3.6: Once the profile is created, the user's role (Pet Owner or Caretaker) cannot be changed. The system must prevent role changes after submission.

4.4 Edit User Profile

4.4.1 Description and Priority

The Edit User Profile feature allows users to update their profile details after they have logged in. Users can modify their name, date of birth, gender, profile picture and location (for Pet Owners) or work experience (for Caretakers). This feature ensures that users can keep their profiles up to date and make necessary changes to their personal information.

Priority: Medium

4.4.2 Stimulus/Response Sequences

Normal Flow:

a. Stimulus: The user logs into the system and navigates to the Profile page.
   Response: The Profile page is displayed with the current profile information.

b. Stimulus: The user selects the Edit Profile button.
   Response: The system displays the current profile information in editable fields.

c. Stimulus: The user updates their profile information (e.g., name, date of birth, gender, location for Pet Owners; name, date of birth, gender, work experience for Caretakers).
   Response: The system validates the inputs and checks for required fields and correct formatting.

d. Stimulus: The user submits the updated profile information.
   Response: The system saves the updated profile information in the database if validation is successful.

e. Stimulus: The user successfully submits the updated profile information.
   Response: The system saves the updated profile, displays a success message, and returns to the Profile page.

Alternative Flows:

- ○ Stimulus: The user selects the Cancel button during the editing process.
  Response: The system prompts the user with a confirmation message to cancel the editing, and if confirmed, returns to the Profile page without saving changes.

Exceptions:

- Stimulus: The user enters invalid or incomplete data, such as missing required fields or incorrectly formatted information.
  Response: The system displays an error message prompting the user to correct the information (e.g., missing fields, invalid date format, etc.).
- Stimulus: The user attempts to select a future date for Date of Birth.
  Response: The system disables the selection of future dates in the date picker, ensuring the user cannot select a future date

4.4.3 Functional Requirements

REQ4.1: The system must allow the user to access the Edit Profile page from their profile section after logging in.

REQ4.2: The system must display the current profile information (name, date of birth, gender, location for Pet Owners; name, date of birth, gender, work experience for Caretakers) on the Edit Profile page for the user to update.

REQ4.3.1: For Pet Owners, the system must provide the following fields for profile creation:

- Name:
  - field must accept a nonempty string and validate that the name is within a reasonable length (At least 3 characters and no more than 30 characters)
- Date of Birth:
  - field must allow the user to input their birth date in a valid format (YYYYMMDD) and validate that the date is in the past and that the user meets the minimum age requirement (16 years old).
- Gender:
  - field must allow the user to select an option such as "Male", "Female", or "Prefer not to say"
- Location:
  - field must allow the user to select their location from a list of Singapore areas (Eg. Boonlay, Pioneer, Expo etc)
- Profile picture:
  - field must be optional
  - field must accept any png or jpg file

REQ4.3.2 For Caretakers, the system must provide the following fields for profile creation:

- Name:
  - field must accept a nonempty string and validate that the name is within a reasonable length (At least 3 characters and no more than 30 characters)
- Date of Birth:
  - field must allow the user to input their birth date in a valid format (YYYYMMDD) and validate that the date is in the past and that the user meets the minimum age requirement (16 years old).
- Gender:
  - field must allow the user to select an option such as "Male", "Female", or "Prefer not to say"
- Work Experience:
  - field must allow users to input a number and validate that the number of years the user input for work experience is less than the user's age which will be derived from the date of birth input
- Profile picture:
  - field must be optional
  - field must accept any png or jpg file

REQ4.4: After successful validation, the system must update the user profile in the database with the new information and store it persistently.

REQ4.5: The system must display a success message upon successful profile update, confirming that the profile has been successfully edited.

REQ4.6: The system must allow the user to cancel the edit process and return to the previous page without saving any changes. The user must be prompted with a confirmation message before cancelling to ensure they do not accidentally lose changes.

REQ4.7: The system must maintain the user’s role (Pet Owner or Caretaker) during the editing process. The role should not be editable after the initial profile creation.

4.5 Create Pet Profile

4.5.1 Description and Priority

After pet owners create their profiles, they can start creating the profile for their pets. This feature allows pet owners to create and manage profiles for their pets after setting up their own user profile. This feature is essential for enabling pet-related services and maintaining accurate pet information for caretakers. Pet owners will need to provide: the pet's name, species (dog, cat, etc.) Breed, age, health conditions, and their preferences, if any.

Priority: High

- Benefit: 9 (Essential for pet care services, enables caretakers to understand pet needs)
- Penalty: 8 (Without pet profiles, owners cannot book services or receive proper care for their pets)
- Cost: 7 (Requires robust frontend forms, image handling, and extensive data validation)
- Risk: 6 (Potential issues with incomplete data)

4.5.2 Stimulus/Response Sequences

Main flow:

a. Stimulus: After creating a pet owner user's profile, pet owner selects the "Add New Pet" button under the profile form.
   Response: System displays the pet profile creation form.
b. Stimulus: Pet owner enters their pet's information (name, species, breed, etc.)
   Response: The system validates the inputs for completeness and correctness. If valid, the pet profile is saved in the database, and the user is redirected to the dashboard or homepage.

Exception flow:

a. Stimulus: The pet owner provides incomplete or invalid data (e.g. missing fields, incorrect formats like an invalid date of birth).
   Response: The system displays an error message prompting the user to correct the input before proceeding with submission.

4.5.3 Functional Requirements

REQ5.1: The system must verify that the user is logged in and is a pet owner before creating a pet

REQ5.2: The system must provide the following fields for profile creation:

- Name:
  - Field must accept a nonempty string and validate that the name is within a reasonable length
- Pet species:
  - Field must accept a nonempty string
- Pet breed:
  - Field must accept a nonempty string
- Pet age:
  - Field must accept a nonempty number in incrementals of 0.5

    - The number inputted must be in number of years
- Pet health status:
    - Field must be optional
- Pet preferences:
    - Field must be optional

REQ5.3: The system must make the correct updates in all data fields of the pet according to what the user inputted in the create page and store it persistently as a new row

REQ5.4: The system must display a success message upon successful pet profile creation, confirming that the profile has been successfully created.

REQ5.5: The system must allow the pet owner to cancel the create process and return to the previous page without saving any changes.

4.6 Edit Pet Profile

4.6.1 Description and Priority

The Edit Pet Profile feature allows pet owners to update their pet profile details after they have logged in. Pet owners can modify their pet's name, species (dog, cat, etc.), breed, age, health conditions, and their preferences. This feature ensures that users can keep their pet's profile up to date and make necessary changes to their pet's information.

Priority: Medium

4.6.2 Stimulus/Response Requirement

Main Flow:

a. Stimulus: The pet owner logs into the system and navigates to the Profile page.
   Response: The Profile page is displayed with the current profile and pet profile information.
b. Stimulus: The pet owner selects the Edit Pet Profile button.
   Response: The system displays the current pet profile information in editable fields.
c. Stimulus: The pet owner updates their pet profile information (e.g., their pet's name, species, breed, age, health conditions, and their preferences).
   Response: The system validates the inputs and checks for required fields and correct formatting.
d. Stimulus: The pet owner submits the updated pet profile information.
   Response: The system saves the updated pet profile information in the database if validation is successful.
e. Stimulus: The pet owner successfully submits the updated pet profile information.
   Response: The system saves the updated pet profile, displays a success message, and returns to the Pet Owner Profile page.

Exception Flow:

a. Stimulus: The user selects the Cancel button during the editing process.
   Response: The system returns to the Pet Owner Profile page without saving changes.
b. Stimulus: The user enters invalid or incomplete data, such as missing required fields or incorrectly formatted information.
   Response: The system displays an error message prompting the user to correct the information (e.g., missing fields, invalid date format, etc.)

4.6.3 Functional Requirements

REQ6.1: The system must allow the pet owner to access the Edit Pet Profile page from their profile section after logging in.

REQ6.2: The system must display the current pet profile information (pet's name, species, breed, age, health conditions, and preferences) on the Edit Pet Profile page for the pet owner to update.

REQ6.3: The system must provide the following fields for profile creation:

- Name:
    - Field must accept a nonempty string and validate that the name is within a reasonable length

- Pet species:
    - Field must accept a nonempty string
- Pet breed:
    - Field must accept a nonempty string
- Pet age:
    - Field must accept a nonempty number in incrementals of 0.5
    - The number inputted must be in number of years
- Pet health status:
    - Field must be optional
- Pet preferences:
    - Field must be optional

REQ6.4: After successful validation, the system must update the pet profile in the database with the new information and store it persistently.

REQ6.5: The system must display a success message upon successful pet profile update, confirming that the profile has been successfully edited.

REQ6.6: The system must allow the pet owner to cancel the edit process and return to the previous page without saving any changes.

REQ6.7: The system must maintain the user's role (Pet Owner) and the current pet's profile during the editing process.

4.7 Create Service

4.7.1 Description and Priority

The Create Service feature enables caretakers to set up and manage their offered pet care services. This feature is crucial for establishing their business presence on the platform and connecting with pet owners. Caretakers will need to provide the type of service they offer, service rates, availability, and location.

Priority: High

- Benefit: 9 (Critical for enabling caretakers to offer services and generate income)
- Penalty: 8 (Without service profiles, caretakers cannot receive bookings or earn through the platform)
- Cost: 7 (Requires complex scheduling system, pricing calculations, and location services)
- Risk: 6 (Challenges with scheduling conflicts, pricing accuracy, location validation)

4.7.2 Stimulus/Response Requirement

Main Flow:

a. Stimulus: After creating a caretaker user's profile, caretaker selects the "Add New Service" button under the profile form.
   Response: System displays the service creation form.
b. Stimulus: Caretaker selects service type.
   Response: System loads a drop down menu with the full list of service types. Caretaker is to choose one service type from this drop down menu.
c. Stimulus: Caretaker enters the desired service rates.
   Response: System validates pricing format.
d. Stimulus: Caretaker selects the days where they are available to offer their service.
   Response: System will turn the colours of the chosen days to orange.
e. Stimulus: Caretaker selects the locations of their service
   Response: System will turn the colours of the chosen days to orange.
f. Stimulus: Caretaker submits service profile
   Response: System validates all required fields. If valid, the service is saved in the database, and the user is redirected to the dashboard or homepage.

Exception Flow:

a. Stimulus: The caretaker provides incomplete or invalid data (e.g. missing fields, incorrect formats like an invalid date of birth).
   Response: The system displays an error message prompting the user to correct the input before proceeding with submission.

4.7.3 Functional Requirements

REQ7.1: The system must verify that the user is logged in and is a caretaker before creating a service

REQ7.2: The system must provide the following fields for service creation:

- Type of service:
  - There must be a drop down menu with the complete list of services available.
- Service rates:

  - The field must accept a nonempty number with increments of 5.
- Availability:
  - The field must display the days in a week, from Monday to Sunday.
  - Users must be able to select multiple days.
- Location:
  - The field must display all the locations available in Singapore.
  - Users must be able to select multiple locations.

REQ7.3: The system must make the correct updates in all data fields of the pet according to what the user inputted in the create page and store it persistently as a new row

REQ7.4: The system must display a success message upon successful service creation, confirming that the service has been successfully created.

REQ7.5: The system must allow the caretaker to cancel the create process and return to the previous page without saving any changes.

4.8 Edit Service

4.8.1 Description and Priority

The Edit Service feature allows caretakers to update and modify their existing service offerings, including the type of service, rates, availability, and location. This functionality ensures that caretakers can keep their service information current, helping pet owners access accurate and relevant details when booking services.

Priority: Medium

4.8.2 Stimulus/Response Requirement

Main Flow:

a. Stimulus: Caretaker navigates to their profile and selects an existing service to edit.
   Response: System loads a prefilled form with the current details of the selected service.
b. Stimulus: Caretaker updates service fields (e.g., modifies rates, changes availability, or updates locations).
   Response: System validates the updated information for correctness and completeness.
c. Stimulus: Caretaker submits the updated service details.
   Response: System saves the changes to the database and redirects the user to the dashboard with a success message.

Exception Flow:

a. Stimulus: Caretaker submits the form with incomplete or invalid data (e.g., missing required fields or invalid pricing format).
   Response: System displays an error message highlighting the problematic fields and prompts the caretaker to correct the input.
b. Stimulus: Caretaker cancels the edit process before submission.
   Response: System discards unsaved changes and returns the user to the previous page without updating the service.

4.8.3 Functional Requirements

REQ8.1: The system must verify that the user is logged in and is a caretaker before allowing the editing of a service.
REQ8.2: The system must load the current service details in an editable form, including:

- Type of service: Display the current selection with a dropdown menu for modification.
- Service rates: Allow the user to update the pricing, validated to accept a nonempty number in increments of 5.
- Availability: Allow updates to the days selected, displaying a weekly view (Monday to Sunday) with multiple day selection enabled.
- Location: Display the current locations selected, allowing users to modify or add multiple locations from a predefined list.

REQ8.3: The system must validate all updated fields and save changes persistently in the database upon submission.

REQ8.4: The system must display a confirmation message to the user upon successful update of the service.
REQ8.5: The system must allow the caretaker to cancel the edit process and return to the previous page without saving any changes.
REQ8.6: The system must ensure that updates to availability or location fields do not conflict with other services provided by the caretaker.

4.9 Search caretakers

4.9.1 Description and Priority

Pet owners can search for caretakers based on location and service type. This feature includes a map integration, where caretakers are displayed on a map based on their location and the services they offer. The user can view caretaker profiles and make a booking or further interactions from the profile page. This feature is of High priority as it directly influences the user's ability to find and engage caretakers efficiently.

Priority: High

- Benefit: 9 (Essential for the user to quickly find caretakers based on location and service type)
- Cost: 6 (Google Maps API integration incurs costs, and database management requires resources)
- Risk: 4 (Risks related to API integration and map loading failures)
- Penalty: 3 (Temporary inconvenience if caretakers or maps do not load correctly, but not critical to overall app function)

4.9.2 Stimulus/Response Sequences

a. Stimulus: The user navigates to the "Caretakers" section in the app.
Response: The app loads the caretaker search interface, showing a list of available caretakers.

b. Stimulus: The user selects the map icon at the bottom right of the search caretakers UI page.
Response: The app displays caretakers on a map, based on the user's location and the services they provide.

c. Stimulus: The user selects a caretaker from the list or map.
Response: The app displays the selected caretaker's full profile.

d. Stimulus: The user selects a caretaker’s profile to view more details.
Response: The app opens the profile page, showing the caretaker's full profile and services.

4.9.3 Functional Requirements

REQ9.1: The system must allow users to search for caretakers based on location
REQ9.2: The system must display a list of caretakers with relevant details such as name, services offered, and location.
REQ9.3: The system must integrate with the Google Maps API to display caretakers’ locations on a map when the map icon is selected.
REQ9.4: The system must allow users to view full profiles of caretakers after selecting them from the list or map.

REQ9.5: The system must display an error message if no caretakers are found that match the user’s search criteria.
REQ9.6: The system must handle errors with the map display and prompt the user to try again if the map fails to load.

4.10 Display Profile

4.10.1 Description and Priority

Pet owners can view their own profiles as well as the profiles of caretakers after performing a search. The profile includes details such as the caretaker's introduction, service details, and ratings. Pet owners will also be able to view their own profile, which includes their personal details and any pets they own. However, caretakers can only view their own profile, which includes their personal and service details but not pet owner profiles.

Priority: High

- Benefit: 9 (Enhances the user experience by providing critical information that helps pet owners make informed decisions about caretakers and also allows pet owners to manage their profiles)
- Cost: 5 (Involves managing and displaying dynamic profile data, but no external integrations like APIs)
- Risk: 4 (Low risk since the feature primarily involves displaying stored profile data)
- Penalty: 4 (If the profile is not displayed correctly, it could frustrate users and affect their decisionmaking process)

4.10.2 Stimulus/Response Sequences

Petowner:

a. Stimulus: The pet owner logs into their account and navigates to their profile page
Response: The app displays the pet owner's profile, including their personal details and pet information (if available).

b. Stimulus: The petowner user selects a caretaker profile after performing a search.
Response: The app displays the selected caretaker’s profile with relevant details like introduction, service details, and ratings.

Caretaker:

c. Stimulus: The caretaker logs into their account and navigates to the profile page
Response: The app displays the caretaker's personal details, service information, and user ratings/reviews.

4.10.3 Functional Requirements

REQ10.1: The system must display the selected caretaker’s profile when a pet owner selects it from the search results.
REQ10.2: The system must allow caretakers to view their own profile, displaying their personal details, list of upcoming bookings, list of services they provide, and ratings and reviews when the caretaker navigates to the profile page.
REQ10.3: The system must allow petowners to view their own profile, including their personal details and any pets they own when the petowner navigates to the profile page
REQ10.4: The system must ensure only pet owners can view caretaker profiles.
REQ10.5: The system must display accurate and updated information for both pet owner and caretaker profiles.
REQ10.6: If the profile data is missing or incomplete, the system should display a default message or placeholder indicating that certain information is unavailable.

4.11 Booking Services

4.11.1 Description and Priority

The Booking Services feature allows pet owners to book appointments with caretakers based on their availability. The pet owner can select the caretaker, the pet for the service, and a suitable time slot. Payment is made and both parties receive notifications of the confirmed booking, with the booking details updated in the calendar.

Priority: High

- Benefit: 9 (The feature is crucial for enabling pet owners to book caretakers, which is a core functionality of the platform)
- Penalty: 9 (Failure to provide a working booking system would severely impact user experience and could lead to loss of users)
- Cost: 6 (The cost involves integrating payment systems and calendar APIs, which is moderately expensive but manageable)
- Risk: 7 (There is a moderate risk of issues like doublebooking or incorrect availability, but these can be mitigated with proper validation)

4.11.2 Stimulus/Response Sequences

Main flow:

a. Stimulus: The pet owner selects a caretaker from the list of available caretakers.
   Response: The system displays the available time slots based on the caretaker's calendar and availability.

b. Stimulus: The pet owner selects a pet and a time slot.
   Response: The system confirms the selected time slot availability.

c. Stimulus: The pet owner confirms the booking and proceeds to payment.
   Response: The system processes the payment and updates the calendar with the booking details.

d. Stimulus: The payment is successfully processed.
   Response: Both the pet owner and caretaker receive notifications, and the booking details are updated in the calendar.

Alternative Flows

- AF11:
  - If the caretaker is unavailable at the selected time, the system will notify the pet owner and prompt them to choose another time slot.

Exception Flow:

- Stimulus: The pet owner attempts to select a time slot that is no longer available.
  Response: The system will grey out the unavailable time slot, preventing the pet owner from selecting it. No error message will be displayed, as the slot is visually unavailable. The pet owner will be prompted to choose from the available time slots that are still active.

4.11.3 Functional Requirements

REQ11.1: The system must allow pet owners to select a caretaker and view the available services.

REQ11.2: The system must show available time slots for a selected caretaker.
REQ11.3: The system must ensure that caretakers' time slots are updated to reflect their availability (to prevent overlapping bookings).
REQ11.4: The system must allow the pet owner to select a date and time slot from the available options.
REQ11.5: The system must enable the pet owner to confirm the booking once the date and time slot are selected.
REQ11.6: The system must allow the pet owner to make a payment for the booked service.
REQ11.7: The system must notify both the pet owner and caretaker once the booking is confirmed.
REQ11.8: The system must create a calendar event for both the pet owner and the caretaker with the booking details.
REQ11.9: The system must ensure that once a time slot is booked, it is no longer available for other pet owners to select.
REQ11.10: The system must prevent pet owners from booking caretakers during their unavailable times.

4.12 Edit Booking

4.12.1 Description and Priority

The Edit Booking feature allows pet owners to modify their upcoming bookings by selecting a new date and time slot. This action is available from their home page under "Upcoming Bookings." After selecting the booking, the pet owner can edit the booking details, reselect the time slot, and confirm the changes. Notifications are sent to both the pet owner and caretaker to confirm the edit.

Priority: High

- Benefit: 9 (Allows flexibility for users to adjust bookings, enhancing user satisfaction and experience)
- Penalty: 9 (Failure to provide editing functionality could lead to frustrated users and negative reviews)
- Cost: 6 (Moderate cost for implementing UI changes and integrating backend for updates)
- Risk: 6 (Moderate risk of miscommunication or scheduling conflicts, mitigated by time slot validation)

4.12.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The pet owner navigates to the home page and clicks on "Upcoming Bookings."
Response: The system displays a list of the pet owner's upcoming bookings.

b. Stimulus: The pet owner selects the booking they want to edit.
Response: The system displays the booking details and options to either edit or cancel the booking.

c. Stimulus: The pet owner selects "Edit."
Response: The system prompts the pet owner to reselect the date and time slot for the booking.

d. Stimulus: The pet owner confirms the new date and time.
Response: The system updates the booking with the new details and sends notifications to both the pet owner and caretaker.

Alternative Flow:

- AF12: If no available time slots match the pet owner's selected date, the system will notify them and prompt them to select a different time.

Exception Flow:

- Stimulus: The pet owner attempts to edit a booking but selects an unavailable time slot.
Response: The system will grey out the unavailable time slot and prompt the pet owner to choose another.

4.12.3 Functional Requirements

REQ12.1: The system must allow the pet owner to view their upcoming bookings on the home page.

REQ12.2: The system must display the booking details when the pet owner selects an upcoming booking.
REQ12.3: The system must provide the pet owner with options to either edit or cancel the booking.
REQ12.4: The system must prompt the pet owner to reselect the date and time when editing a booking.
REQ12.5: The system must validate the availability of the selected time slot during the editing process.
REQ12.6: The system must update the booking details after the pet owner confirms the new date and time.
REQ12.7: The system must notify both the pet owner and caretaker after the booking has been edited.
REQ12.8: The system must ensure that the updated booking details are reflected in the calendar for both the pet owner and the caretaker.
REQ12.9: The system must update the booking history and list of upcoming bookings for both caretaker and petowner where the cancelled booking is no longer shown

4.13 Cancel Booking

4.13.1 Description and Priority

The Cancel Booking feature allows pet owners to cancel their upcoming bookings directly from the home page. After selecting the booking they wish to cancel, the system immediately processes the cancellation and notifies both the pet owner and caretaker. This action does not require reselecting a time slot but confirms the cancellation.

Priority: High

- Benefit: 9 (Provides flexibility to pet owners and ensures bookings are accurately managed)
- Penalty: 9 (Failure to allow cancellations can lead to confusion and frustration for users)
- Cost: 5 (Low cost as this feature only requires backend support for updating bookings)
- Risk: 5 (Low risk if the cancellation process is well-defined, but there may be confusion if cancellations aren't communicated promptly)

4.13.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The pet owner navigates to the home page and clicks on "Upcoming Bookings."
Response: The system displays the pet owner's upcoming bookings.

b. Stimulus: The pet owner selects the booking they want to cancel.
Response: The system displays the booking details and options to either edit or cancel the booking.

c. Stimulus: The pet owner selects "Cancel."
Response: The system processes the cancellation and updates the status of the booking.

d. Stimulus: The cancellation is processed.
Response: The system sends notifications to both the pet owner and caretaker confirming the cancellation.

Alternative Flow:

- AF12: If the pet owner attempts to cancel a booking that has already been completed or passed, the system will not provide a cancel option.

4.13.3 Functional Requirements

REQ13.1: The system must allow the pet owner to view their upcoming bookings on the home page.
REQ13.2: The system must provide the pet owner with the option to cancel a booking from the booking details page.
REQ13.3: The system must immediately process the cancellation once the pet owner confirms the action.
REQ13.4: The system must notify both the pet owner and caretaker upon successful cancellation of the booking.
REQ13.5: The system must update the calendar for both the pet owner and caretaker after the cancellation is processed.
REQ13.6: The system must prevent cancellation of bookings that are completed or past the scheduled time.

REQ13.7: The system must update the booking history and list of upcoming bookings for both caretaker and petowner where the cancelled booking is no longer shown

4.14 Create Payment

4.14.1 Description and Priority

The Create Payment feature enables pet owners to make secure payments when booking a caretaker's service. This feature is essential for completing transactions on the platform and ensuring a smooth booking process. Pet owners will need to provide payment details to finalise their bookings.

Priority: High

- Benefit: 9 (Essential for completing bookings and facilitating caretaker compensation)
- Penalty: 8 (Without a payment system, bookings cannot be finalised, and the platform cannot function effectively)
- Cost: 7 (Requires integration with secure payment gateways and robust validation mechanisms)
- Risk: 6 (Potential challenges with transaction failures or payment gateway issues)

4.14.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: Pet owner selects the "Book Now" button for a specific caretaker service.
Response: The system displays the booking details, including the total amount to be paid.

b. Stimulus: Pet owner selects “Pay”
Response: The system processes the payment and confirms if the payment is successful.

c. Stimulus: Payment is successfully processed.
Response: The system displays a success message and notifies both the caretaker and pet owner.

4.14.3 Functional Requirements

REQ14.1: The system must allow pet owners to initiate a payment process only after booking a service.
REQ14.2: The system must display a payment form with booking details (e.g., service type, caretaker name, and total amount).
REQ14.3: The system must display a success message to the user upon successful payment.
REQ14.4: The system must notify both the caretaker and the pet owner of the completed payment via the notifications.
REQ14.5: The system must allow pet owners to cancel the payment process before submission, returning to the booking page without processing any transactions.

4.15 Notifications

4.15.1 Description and Priority

The notifications feature enables users to receive timely alerts for important events within the PetBuddy app. Notifications will be sent in real time to both pet owners and caretakers based on specific triggers such as when a review is submitted, a payment is made, or a message is received. This feature ensures that users are kept informed about their interactions and transactions on the platform, enhancing communication and user engagement.

Priority: High

- Benefit: 9 (Keeps users informed and engaged, improves communication and accountability between pet owners and caretakers)
- Penalty: 8 (Without notifications, users may miss critical updates, leading to delayed actions and confusion)
- Cost: 7 (Requires integration with a notification system and backend support for realtime messaging)
- Risk: 6 (Potential issues with notification delivery due to network or system failures)

4.15.2 Stimulus/Responses Sequences

Main flow:

a. Stimulus: A pet owner submits a review for a service provided by a caretaker.
   Response: The caretaker receives a notification indicating that a review has been posted on their service.

b. Stimulus: A pet owner completes a payment for a service.
   Response: Both the pet owner and the caretaker receive a notification confirming the successful payment.

c. Stimulus: A pet owner sends a message to a caretaker.
   Response: The caretaker receives a notification that a new message has been sent.

d. Stimulus: A caretaker sends a message to a pet owner.
   Response: The pet owner receives a notification that a new message has been sent.

4.15.3 Functional Requirements

REQ15.1: The system must send a notification to the caretaker when a pet owner submits a review on their service.
REQ15.2: The system must send a notification to both the pet owner and the caretaker when a payment for a service is completed.
REQ15.3: The system must send a notification to the recipient when a new message is sent to them.
REQ15.4: The system must ensure that notifications are delivered in real time or as soon as possible to the recipient.
REQ15.5: The system must allow users to configure their notification preferences (e.g., enable/disable, sound, frequency).
REQ15.6: The system must ensure that notifications include relevant information, such as the sender's name, service details, or payment amount.

REQ15.7: The system must allow for message-based notifications to be displayed in the app and sent via email (or other communication channels, if required).

4.16 Messaging

4.16.1 Description and Priority

The messaging feature allows pet owners and pet caretakers to communicate in real time through the platform. This feature is essential for fostering effective communication regarding pet services, booking details, and any updates or special requests. Pet owners and caretakers will be able to send and receive messages related to pet care, service requests, and any concerns that may arise during service.

Priority: High

- Benefit: 9 (Improves communication, enhances user experience, ensures clarity on services)
- Penalty: 7 (Without messaging, there is limited interaction between pet owners and caretakers, leading to confusion or delayed services)
- Cost: 6 (Requires a websocket integration, frontend interface, and data synchronisation)
- Risk: 5 (Possible delays or message delivery failures due to network or system issues)

4.16.2 Stimulus/Response Sequences

Main flow:

a. Stimulus: A pet owner clicks on the "Messages" tab from the caretaker profile.
   Response: The system displays a list of ongoing conversations with pet caretakers.

b. Stimulus: Pet owner selects a conversation with a specific caretaker.
   Response: The system displays the conversation history and an input field for the user to type a new message.

c. Stimulus: Pet owner types a message and presses "Send"
   Response: The system sends the message to the caretaker and updates the conversation history with the new message.

Exception flow:

a. Stimulus: Pet owner tries to send an empty message.
   Response: The system displays an error message stating that the message cannot be empty.

b. Stimulus: Pet owner tries to send a message without being logged in.
   Response: The system redirects the user to the login page with a message to sign in first.

4.16.3 Functional Requirements

REQ16.1: The system must ensure that messages are sent and received in real time.
REQ16.2: The system must allow users to initiate and respond to messages from pet owners and caretakers.
REQ16.3: The system must store message histories for each conversation, ensuring users can view previous messages.
REQ16.4: The system must notify users when a new message has been received from the other party.
REQ16.5: The system must allow users to send text messages, but also support media files such as images or videos in future releases.

4.17 Change Password

4.17.1 Description and Priority

The Change Password feature allows users to update their account password for security purposes. This feature is essential for maintaining account security and follows standard password management practices. Users must provide their current password for verification before setting a new password. The required information is as follows: current password, new password, confirm new password.

Priority: High
- Benefit: 8 (Essential for account security and user control)
- Penalty: 8 (Without this feature, users cannot maintain secure access to their accounts)
- Cost: 5 (Relatively straightforward implementation with existing authentication system)
- Risk: 7 (Security concerns, password validation, potential for account lockouts)

4.17.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: User selects "Change Password" from their Profile page, under General.
   Response: System displays the change password form with three fields: current password, new password, confirm new password.
b. Stimulus: User enters current password
   Response: System validates the field is not empty, and the input matches the current password.
c. Stimulus: User enters new password
   Response: System ensures that the new password meets the requirement
d. Stimulus: User confirm new password
   Response: System checks if it matches the new password that was previously inputted. The system shows an error message if the password does not match.
e. Stimulus: User submits change password form
   Response: System validates all fields: verifies current password is correct, confirms new password meets all requirements, updates password in database. The system will show a success message and redirect the user to the Profile page.

Exception Flow:

a. Stimulus: User enters incorrect current password.
   Response: The system displays an error message.
b. Stimulus: New password doesn't meet requirements.
   Response: The system displays an error message.
c. Stimulus: Confirm new password doesn't match new password
   Response: The system displays an error message and prevents change of password form submission.

4.17.3 Functional Requirements

REQ17.3.1: The system must ensure password meets requirements

REQ17.3.1.1: Password must be at least 10 characters long

REQ17.3.1.2: Password must contain at least 1 special character

REQ17.3.1.3: Password must include at least 1 uppercase character

REQ17.3.1.4: Password must include at least 1 lowercase character

REQ17.3.1.5: Password must include at least 1 number

REQ17.3.1.6: Password must not exceed 20 characters long

REQ17.3.1.7: If password do not meet any of the requirements, an error message must pop up below to indicate which requirements are not fulfilled

REQ17.3.1.8: Password must be masked with dots by default

REQ17.3.2: If the password entered in the confirm new password field does not match the password entered previously, an error message must pop up to indicate that the passwords do not match

4.18 View Booking History

4.18.1 Description and Priority

The View Booking History feature allows users (pet owners and caretakers) to review details of their past bookings. This feature provides transparency and recordkeeping for users, enabling them to track completed transactions, review past services, and verify payment or service details.

Priority: Medium

- Benefit: 7 (Enhances user experience by providing access to historical data)
- Penalty: 6 (Without this feature, users may face inconvenience tracking previous services)
- Cost: 5 (Requires data storage and retrieval functions)
- Risk: 4 (Potential issues with data accuracy or incomplete records)

4.18.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The user selects the "Booking History" option from the profile page.
   Response: The system retrieves and displays a list of past bookings sorted by date, showing essential details such as service type, date, caretaker/pet owner information, and booking status.

Exception Flow:

a. Stimulus: The user has no previous bookings.
   Response: The system displays a message indicating that there are no bookings to show.

b. Stimulus: System fails to retrieve booking history as user is not logged in.
   Response: Redirect user to login page

4.18.3 Functional Requirements

REQ18.1: The system must verify the user is logged in before displaying the booking history.
REQ18.2: The system must retrieve and display a list of completed bookings with key details, including service type, caretaker/pet owner name, date, and payment status.
REQ18.3: The system must handle cases where no booking history is available and display an appropriate message.
REQ18.4: The system must handle retrieval errors and notify the user with an error message if bookings cannot be fetched.

4.19 View Payment History

4.19.1 Description and Priority

The View Payment History feature allows users (pet owners and caretakers) to access records of all their past payments. This feature ensures financial transparency and helps users track their spending or earnings on the platform. Users can view details such as the date of payment, amount, and service associated with the payment.

Priority: Medium

- Benefit: 8 (Provides users with an overview of their financial transactions for better management and accountability)
- Penalty: 7 (Without this feature, users may face difficulties tracking their financial activities)
- Cost: 6 (Requires secure storage and retrieval of payment data)
- Risk: 5 (Potential issues with incomplete or inaccurate payment records)

4.19.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The user selects the "Payment History" option from the dashboard or menu.
   Response: The system retrieves and displays a list of past payments sorted by date, showing essential details such as payment amount, service name, caretaker/pet owner name, and payment status.

b. Stimulus: The user clicks on a specific payment entry.
   Response: The system displays detailed information about the selected payment, including service details and payment date.

Exception Flow:

a. Stimulus: The user has no payment history.
   Response: The system displays a message indicating that there are no payment records available.

4.19.3 Functional Requirements

REQ19.1: The system must verify that the user is logged in before displaying payment history.
REQ19.2: The system must retrieve and display a list of payments with details, including payment amount, service name, caretaker/pet owner name, and date.
REQ19.3: The system must allow users to click on a payment entry to view detailed information about the selected transaction.
REQ19.4: The system must handle cases where no payment history is available and display an appropriate message.
REQ19.5: The system must handle retrieval errors gracefully and notify the user with an error message if payments cannot be fetched.
REQ19.6: The system must ensure all payment data displayed is accurate and uptodate.

4.20 View Upcoming Bookings

4.20.1 Description and Priority

The View Upcoming Bookings feature enables users (pet owners and caretakers) to view a list of their scheduled services. This feature ensures users can easily manage their schedules, prepare for upcoming appointments, and avoid conflicts. Users can see details such as the date, time, service type, location, and associated pet/caretaker.

Priority: High

- Benefit: 9 (Crucial for scheduling and maintaining a seamless user experience)
- Penalty: 8 (Without this feature, users may struggle to keep track of their commitments, leading to missed or doublebooked appointments)
- Cost: 7 (Requires realtime integration with the booking system and calendar APIs)
- Risk: 6 (Potential issues with data accuracy or synchronisation errors)

4.20.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The user selects the "Upcoming Bookings" option from the homepage.
   Response: The system retrieves and displays a list of all upcoming bookings sorted by date and time, including details such as service type, date, time, location, and pet/caretaker name.

b. Stimulus: The user clicks on a specific booking entry.
c. Response: The system displays detailed information about the selected booking, including the service description, associated user, and any special instructions or notes. Pet owners can also edit/cancel booking

Exception Flow:

a. Stimulus: The user has no upcoming bookings.
   Response: The system displays a message indicating that there are no upcoming bookings.

4.20.3 Functional Requirements

REQ20.1: The system must verify that the user is logged in before displaying upcoming bookings.
REQ20.2: The system must retrieve and display a list of upcoming bookings with details, including service type, date, time, location, and associated pet/caretaker.
REQ20.3: The system must allow users to click on a booking entry to view detailed information about the selected booking.
REQ20.4: The system must handle cases where no upcoming bookings are available and display an appropriate message.
REQ20.5: The system must handle retrieval errors and notify the user with an error message if bookings cannot be fetched.
REQ20.6: The system must ensure all booking data displayed is accurate and uptodate.

4.21 Review/Rating

4.21.1 Description and Priority

The Review/Rating feature allows pet owners to submit and amend reviews and ratings for caretaker services. Pet owners can rate the caretaker service on a scale of 1 to 5 stars and provide textual feedback on their experience. Reviews can be submitted anonymously, and both pet owners and caretakers can view the ratings. Caretakers are notified when a new review is posted.

Priority: Medium

- Benefit: 7 (This feature encourages user feedback and trust-building between pet owners and caretakers, improving service quality)
- Penalty: 5 (Failure to allow reviews may diminish the credibility and transparency of the platform)
- Cost: 4 (The cost is relatively low, as it mainly involves adding functionality for reviews, ratings, and notifications)
- Risk: 6 (There is a moderate risk of users submitting invalid reviews, such as duplicates or inaccurate feedback, which could negatively affect the platform's reputation)|

4.21.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The pet owner navigates to the "Booking History" section to find a specific booking
   Response: The system displays an option to leave a rating and review for the service.

b. Stimulus: The pet owner selects the option to submit a rating and review.
   Response: The system presents a rating scale (1-5 stars) and a text field for the review

c. Stimulus: The pet owner enters a rating and review and submits it.
   Response: The system verifies that the pet owner has received the service and hasn’t posted a review for the same service before. The system then submits the review and notifies the caretaker.

d. Stimulus: The pet owner wants to amend their review or rating.
   Response: The system allows the pet owner to edit the review text or rating.

Alternative Flows:

- AF-21: If the pet owner selects the option to leave a review anonymously:
  - Stimulus: The user selects to leave the review anonymously.
  - Response: The system verifies the user's identity but displays the review with no identifiable information ("Anonymous" name)

Exception Flow:

- Stimulus: The pet owner attempts to submit more than one review for the same service.
  Response: The system displays an error message stating that multiple reviews for the same service are not allowed.
- Stimulus: The pet owner tries to submit a review without having received the service.
  Response: The system prevents the submission and displays an error message saying the review cannot be posted.

4.21.3 Functional Requirements

REQ21.1: The system must allow pet owners to leave a rating (1-5 stars) and a review text for the services they received.
REQ21.2: The system must prevent users from submitting more than one review for the same service.
REQ21.3: The system must provide a 500-character limit for the review text.
REQ21.4: The system must allow pet owners to submit reviews anonymously if selected.
REQ21.5: The system must send notifications to caretakers when a new review or rating is posted.
REQ21.6: The system must allow pet owners to edit their previously submitted review and rating for a specific service.
REQ21.7: The system must ensure that the review and rating are viewable by both the pet owner and caretaker.
REQ21.8: The system must display the review without identifiable information if submitted anonymously (blank profile picture, “Anonymous” username).

4.22 View Events

4.22.1 Description and Priority

The View Upcoming Events feature allows users (pet owners and caretakers) to view only upcoming events related to pet care services, sourced directly from Eventbrite. This can include pet care workshops, adoption events, or any community events that are listed on Eventbrite. Users can see event details like the event name, date, time, location, and a brief description. This feature will help users stay informed about pet-related events in Singapore.

Priority: Medium

- Benefit: 8 (Provides users with valuable community engagement opportunities that could enhance their pet care experience)
- Penalty: 6 (Without this feature, users might miss out on events related to pet care, reducing platform engagement)
- Cost: 7 (Requires integration with the Eventbrite API for fetching and displaying event details)
- Risk: 5 (Possible issues with data synchronisation or API failures when retrieving event data)

4.22.2 Stimulus/Response Sequences

Main Flow:

a. Stimulus: The user navigate to the homepage
   Response: The system fetches upcoming events related to pet care from Eventbrite and displays them in a list, showing event title, date, time, location, and a brief description.

b. Stimulus: The user clicks on an event from the list.
   Response: The system redirects the user to the Eventbrite registration page for the selected event where they can sign up

Exception Flow:

a. Stimulus: There are no upcoming events found for pet care services on Eventbrite.
   Response: The system informs the user that no upcoming events are available at the moment.

4.22.3 Functional Requirements

REQ22.1: The system must ensure that the user is logged in before displaying upcoming events.

REQ22.2: The system must retrieve and display a list of only upcoming events related to pet care from Eventbrite. The event list should include event title, date, time, location, and description.

REQ22.3: The system must allow users to click on an event entry to view full event details, including a link to Eventbrite for registration.

REQ22.4: The system must redirect users to Eventbrite's registration page when they choose to register for an event.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should be able to handle up to 5,000 users simultaneously with minimal latency.
- The system should be able to respond to any requests requested by the user within 5 seconds.
- Upon running the app, the app must be fully functional within 5 seconds.

5.2 Safety Requirements

- The system shall not require or share any personal contact information (e.g., phone numbers, home addresses) between pet owners and caretakers in the chat function. All communication shall be conducted within the app to ensure the privacy and safety of both parties.
- The system shall require all pet care service providers (e.g., sitters, walkers) to register with a valid email before they can be listed as available for services. This ensures that only registered users can offer services, helping to reduce the risk of unqualified or potentially harmful individuals providing care.

5.3 Security Requirements

- The system shall provide clear and accessible Terms of Service and Privacy Policy to all users. The system shall also obtain explicit consent from users before collecting or processing their personal data, ensuring transparency and compliance with privacy regulations.
- The system shall implement regular backups for critical user data (e.g., pet owner details, service booking information) to ensure data integrity and availability in the event of system failures or disasters.
- The system shall automatically expire user sessions after 30 minutes of inactivity to prevent unauthorised access and enhance the security of user accounts.

5.4 Software Quality Attributes

- The system should be online and available for usage at all times of the day, except for scheduled maintenance timings.
- The system should not lose any data after restart or update.
- The app should provide informative and user friendly error messages, guiding users on how to resolve issues.
- Users should be able to see the full content regardless of what mobile phones they are using.
- Users must not spend more than 10 minutes to create an account.

5.5 Business Rules

5.5.1 Review and Rating System

- Reviews must follow the community guidelines, and inappropriate, abusive, or offensive content will be subject to removal or moderation.
- Ratings for a caretaker's service will be averaged to provide an overall rating that is visible to all users.

5.5.2 User Registration and Authentication

- All users (pet owners and caretakers) must register with a valid email address to access the system's full features.
- The system will require users to set up a strong password that meets security standards for account protection.

5.5.3 Service Availability and Scheduling

- Caregivers must manually set their availability for specific services, and this availability must be updated in real time to reflect any changes.
- Pet owners will only be able to book services within the available time slots indicated by the caregiver.

5.5.4 Messaging

- Users must refrain from sending inappropriate messages such as vulgar language or providing links to a scam website
- Users who partake in such behaviour will be subject to removal or moderation

6. Other Requirements

6.1 Internationalisation Requirements

- Initially, PetBuddy will support only the English language. Future releases will consider multilanguage support, including local languages like Mandarin, Malay or Tamil, to accommodate a wider range of users in Singapore.
- Date and time representations will be based on the Singapore timezone (GMT+8), but future releases may support international timezones as the user base expands globally.

6.2 Legal Requirements

- The system must adhere to the guidelines of the PDPA to ensure the privacy and protection of personal data, especially for pet owners and caretakers.
- The platform must ensure that pet care service providers understand and comply with local laws regarding pet welfare and safety during services. Although the system does not currently validate qualifications, the platform will operate with the intent of upholding relevant pet welfare standards.

6.3 Reuse Objectives

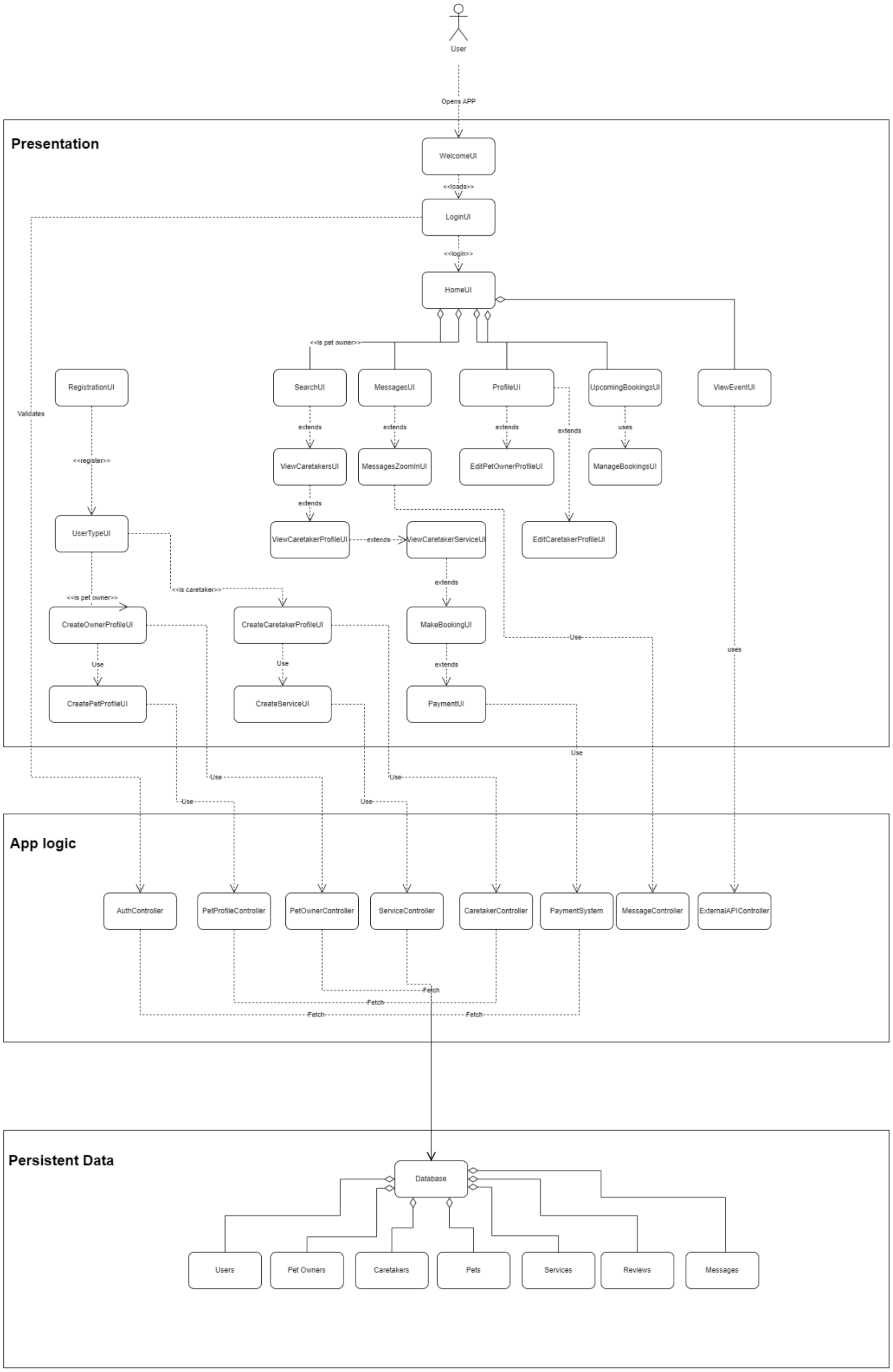
- Components developed for the PetBuddy platform must be modular and reusable within the project and for potential future projects. This includes service-related APIs and user management components.
- All APIs must follow industry standards such as RESTful API principles to ensure ease of integration and future scalability, as well as potential integration with third party platforms.

Appendix A: Glossary

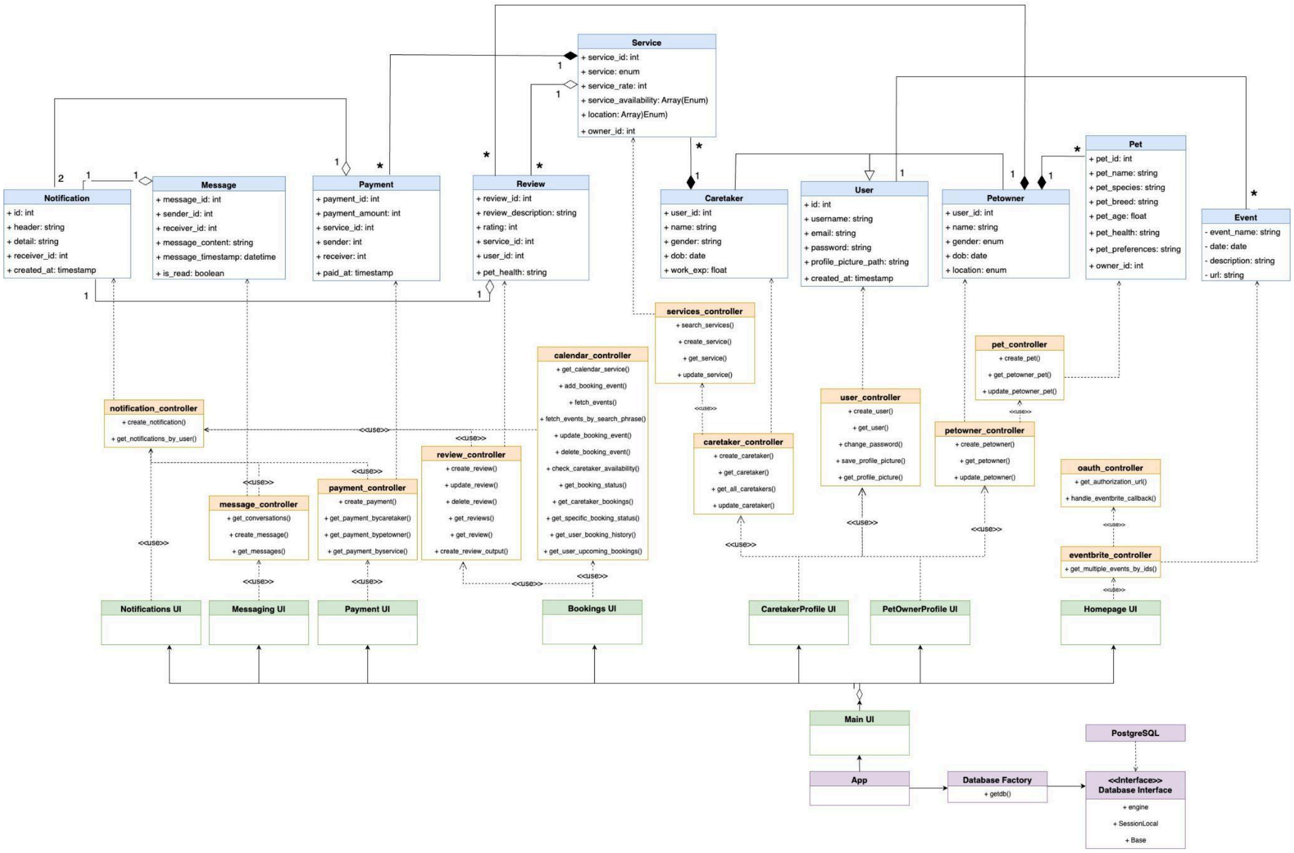
Acronym/Abbreviation	**Meaning**
API	Application Programming Interface - A set of protocols, tools, and definitions that allow different software applications to communicate with each other. It specifies how software components should interact, enabling developers to access specific functionality or data in a system, service, or platform without needing to understand its internal workings.
PDPA	Personal Data Protection Act - Singapore's law that regulates the collection, use, and disclosure of personal data. It aims to protect individuals' privacy while allowing organisations to use personal data responsibly.
SQL	Structured Query Language - SQL is a standard programming language used to manage and manipulate relational databases. It allows users to query, insert, update, and delete data, as well as define and manage database structures
FastAPI	FastAPI: - A modern, high performance web framework for building APIs with Python 3.7+ based on standard Python type hints. It is designed to create RESTful APIs quickly and efficiently, offering automatic validation of request data, high speed performance, and easy integration with other tools like OAuth and databases.
React	React - A popular JavaScript library for building user interfaces, primarily for singlepage applications. It allows developers to create reusable UI components and efficiently update the user interface in response to data changes. React is known for its fast rendering using a virtual DOM and is widely used for building dynamic, interactive web applications.
Frontend	Frontend - The part of a web application or website that users interact with directly. It includes everything users see and experience, such as the layout, design, and interactive elements.

Backend	Backend - The server-side part of a web application or website that manages data, business logic, and database interactions. It is responsible for processing requests from the frontend, handling data storage, and ensuring the application functions correctly.
SRS	Software Requirements Specification - A document that describes the functional and nonfunctional requirements for a software application. It outlines the system's intended purpose, features, performance standards, and design constraints. The SRS serves as a blueprint for developers and stakeholders to ensure that the software meets the intended goals and user needs.

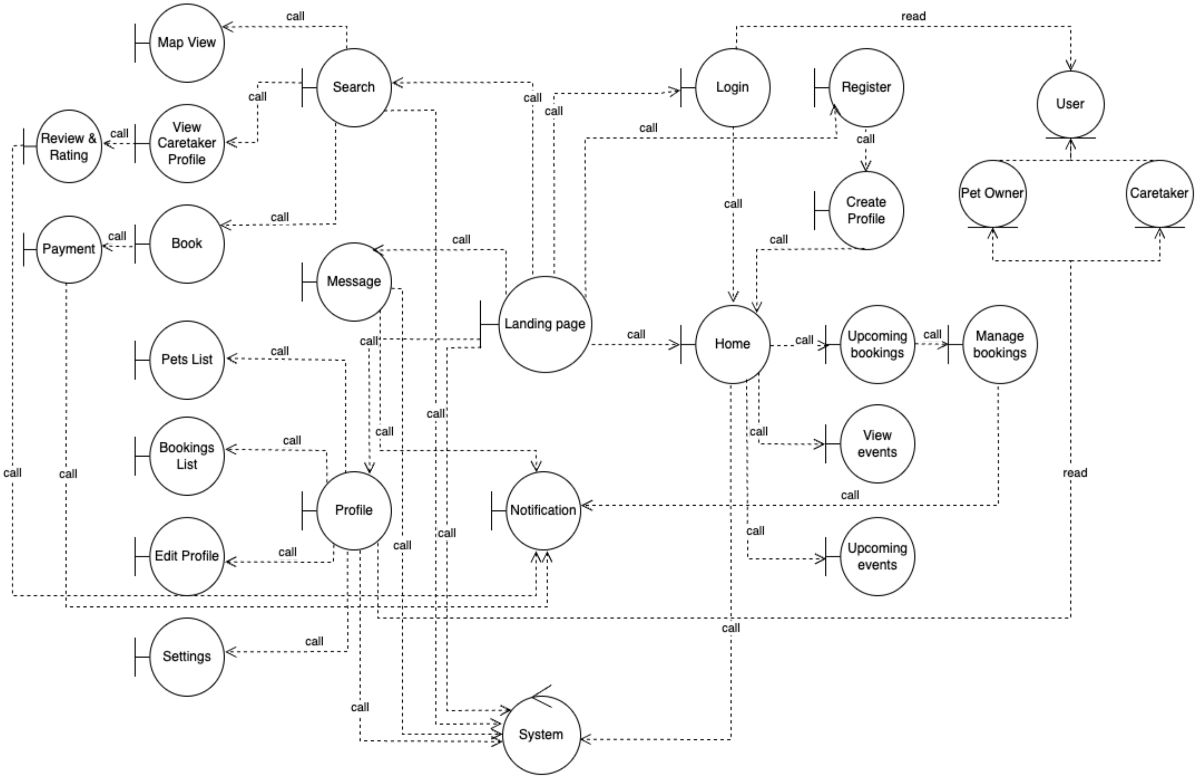
Appendix B: Analysis Models



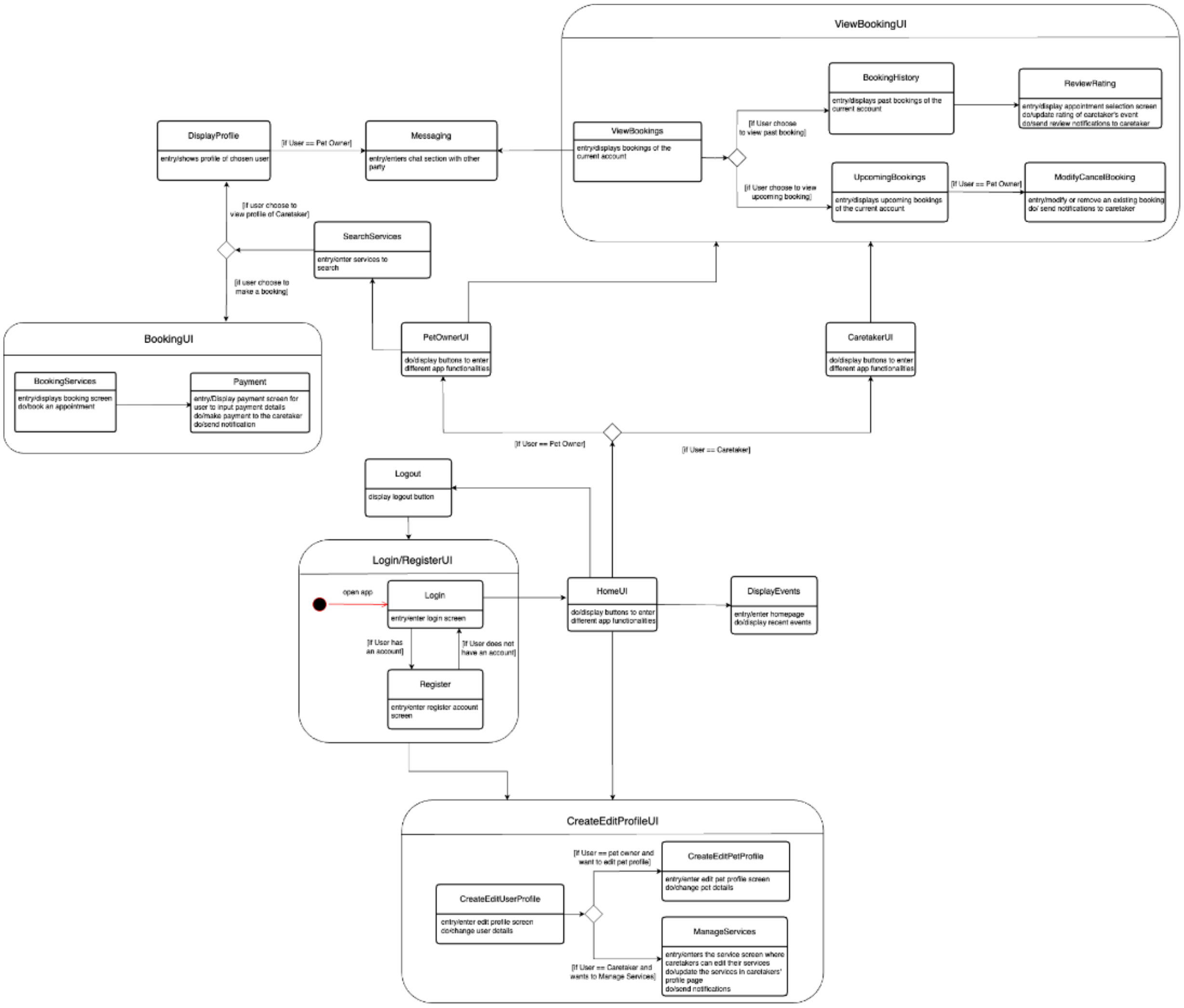
System Architecture



Class Diagram



Key Boundary Class Diagram


ViewBookingUI
ViewBookings
entry/displays bookings of the current account
[If User choose to view past booking]
[if User choose to view upcoming booking]
BookingHistory
entry/displays past bookings of the current account
ReviewRating
entry/display appointment selection screen
do/update rating of caretaker's event
do/send review notifications to caretaker
UpcomingBookings
entry/displays upcoming bookings of the current account
[If User == Pet Owner]
ModifyCancelBooking
entry/modify or remove an existing booking
do/ send notifications to caretaker
DisplayProfile
entry/shows profile of chosen user
[If User == Pet Owner]
Messaging
entry/enters chat section with other party
[If user choose to view profile of Caretaker]
SearchServices
entry/enter services to search
[if user choose to make a booking]
BookingUI
BookingServices
entry/displays booking screen
do/book an appointment
Payment
entry/Display payment screen for user to input payment details
do/make payment to the caretaker
do/send notification
PetOwnerUI
do/display buttons to enter different app functionalities
CaretakerUI
do/display buttons to enter different app functionalities
[If User == Pet Owner]
[If User == Caretaker]
Logout
display logout button
Login/RegisterUI
open app
Login
entry/enter login screen
[If User has an account]
[If User does not have an account]
Register
entry/enter register account screen
HomeUI
do/display buttons to enter different app functionalities
DisplayEvents
entry/enter homepage
do/display recent events
CreateEditProfileUI
CreateEditUserProfile
entry/enter edit profile screen
do/change user details
[If User == pet owner and want to edit pet profile]
CreateEditPetProfile
entry/enter edit pet profile screen
do/change pet details
[If User == Caretaker and wants to Manage Services]
ManageServices
entry/enters the service screen where caretakers can edit their services
do/update the services in caretakers' profile page
do/send notifications

Dialog Map

Appendix C: To Be Determined List

There are no TBD items for this version of Software Requirement Specification

Source: http://www.frontiernet.net/~kwiegers/process_assets/srs_template.doc