# Software Requirements and Design Document

# For

# Group <20>

Version 1.0

**Authors**:
Brady Henderson
Sebastian Metellus
Chelsea Mensah
Ryan Fontaine
Nevin Ramjitsingh

## 1. Overview (5 points)

This web application can be used to create personal schedules to help pet owners plan out all their pet's needs. This includes when they want to go on walks, when to feed their pet, or even to meet up with other friends. This system works by emailing the user via a provided email address and a specified time and frequency. The user can input dates and times on a calendar interface to schedule events in which they can choose to be notified for.
The application is built using React JS, HTML, and CSS. The HTML and CSS code are used for rendering the web pages the user can interact with. The React JS code is implemented for dynamic functionality, such as entering user registration or log on information via communicating with the database. We are using Firebase to handle user authentication and to handle the user's data as well. We use the Cloud Firestore where we have a user's collection populated with unique user documents where each user has their separate data (like the dogs and events array). We are using the FullCalendar API to display the events on a calendar for the user, and we are using FullCalendar methods to edit add and delete events as well.

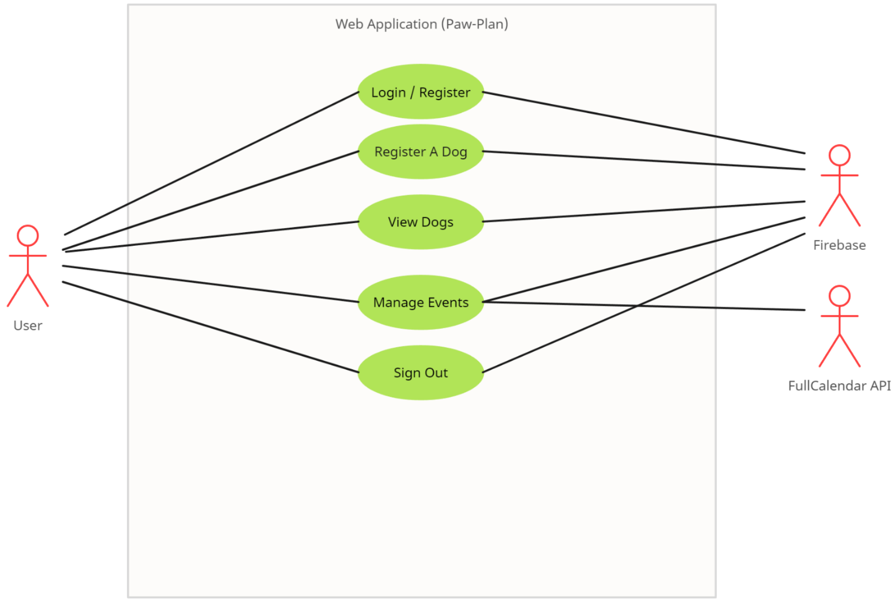
## 2. Functional Requirements (10 points)

1. The home page must be rendered when visiting the default URL.: High
2. The buttons on the Nav Bar must route the user to the corresponding page: High
3. The user must be able to use the login prompt to authenticate themselves on the Login page: High
4. The user must be able to create a new account via the registration prompt on the Registration page: High
5. The about page must show a general description of the product: Low
6. An authenticated user must be able to add their pet's information on the Dashboard page: Medium
7. An authenticated user must be able to add important dates on the Calendar page: High
8. An authenticated user must be able to edit personal information on the settings page: High

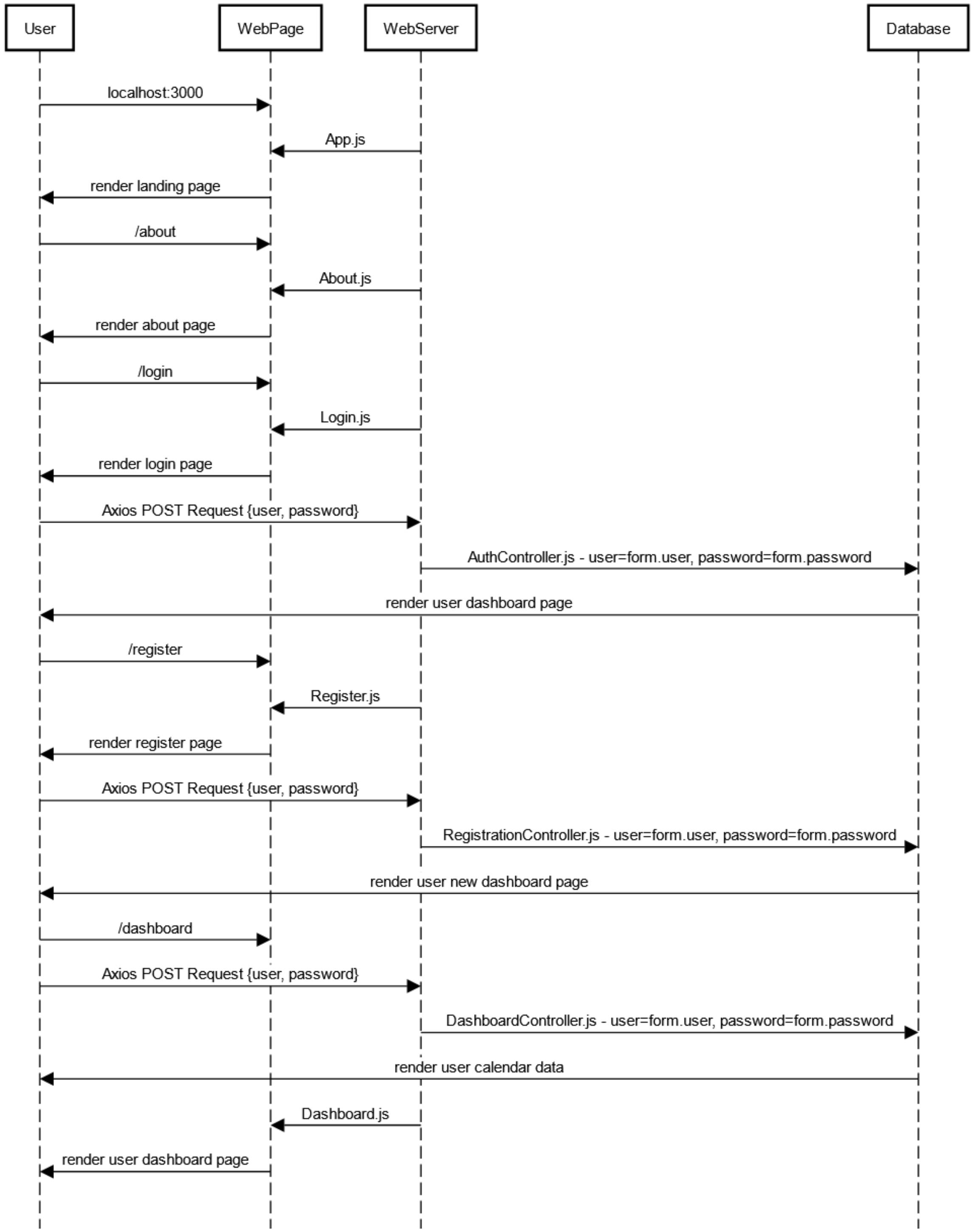
## 3. Non-functional Requirements (10 points)

1. The login script should not be vulnerable to SQL injection.
2. The database should not be able to be deleted by users.
3. Users should be able to log out and log back in and have the same personal experience.
4. Users should not be able to view other users' information.
5. Pages should render consistently across different screen sizes.

## 4. Use Case Diagram (10 points)



## 5. Class Diagram and/or Sequence Diagrams (15 points)



Paw Plan

User
WebPage
WebServer
Database
localhost:3000
App.js
render landing page
/about
About.js
render about page
/login
Login.js
render login page
Axios POST Request {user, password}
AuthController.js - user=form.user, password=form.password
render user dashboard page
/register
Register.js
render register page
Axios POST Request {user, password}
RegistrationController.js - user=form.user, password=form.password
render user new dashboard page
/dashboard
Axios POST Request {user, password}
DashboardController.js - user=form.user, password=form.password
render user calendar data
Dashboard.js
render user dashboard page

## 6. Operating Environment (5 points)

Paw Plan will be able to operate on Web browsers such as Safari, Chrome.To store the login information and pet profiles our software must be compatible with Firebase. Our software must also be compatible with the API's that we are using so far that the full calendar API.

## 7. Assumptions and Dependencies (5 points)

- The web application assumes the user has an internet connection.
- The web application assumes the user has an email address.
- The web application assumes that the user has a dog.
- The web application assumes that the FullCallender API is working and doesn't lose support
- The web application assumes that the Firebase is working and doesn't lose support.
- The web application assumes that all imports used throughout the application is working and doesn't lose support