

**School of Computer Science and Engineering
SC2006 Software Engineering
Lab Group A26**

**Software Requirement Specification
For
TrainTogether**

Version 5.3 approved

Prepared By:

Lee Yen Foong Ernest (U2122511A)

Low Zhan Long (U2120919K)

Solomon Duke Tneo Yruan Rui (U2122205J)

Tey Li Zhang Edmund (U2121492J)

Wan Kai Jie (U2120427H)

13th April 2023

Table of Contents

1.0 Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience	2
1.4 Product Scope	2
2.0 Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	4
2.3 User Classes and Characteristics	5
2.4 Operating Environment	5
2.5 Design and Implementation Constraints	5
2.6 User Documentation	5
2.7 Assumptions and Dependencies	6
3.0 External Interface Requirements	7
3.2 Hardware Interfaces	17
3.3 Software Interfaces	18
3.4 Communications Interfaces	19
4.0 Functional Requirements	20
5.0 Other Nonfunctional Requirements	22
5.1 Performance Requirements	22
5.2 Reliability Requirements	22
5.3 Safety Requirements	23
5.4 Security Requirements	23
6.0 Data Dictionary	24
7.0 Other Requirements	25
7.1 Installation	25
7.2 Running the App	25
8.0 Appendix	26
8.1 Use Case Diagram	26
8.2 Use Case Description	27
8.3 Class Diagram	54
8.3.1 View and Routing Manager Class Diagram	55
8.3.2 Observer and Controllers Class Diagram	56
8.3.3 Models Class Diagram	57
8.4 Dialog Map	58
8.5 System Architecture	63
8.6 Sequence Diagram	64
8.7 Unit Testing	88
8.7.1 Black Box Testing	88
8.7.2 White Box Testing	96

1.0 Introduction

1.1 Purpose

The purpose of this SRS is to provide a detailed description of the requirements for the development of TrainTogether, a web application. The SRS outlines the functional and non-functional requirements of the web application including, use case models, class diagrams, sequence diagram, state machine diagram, user interface design, system architecture and other requirements. This document will also serve as a reference for the development team, stakeholders and clients, ensuring everyone involved in this project is clear and understands its objectives.

1.2 Document Conventions

The following describes the standard and guidelines used in creating the TrainTogether Software Requirements Specification (SRS) document:

1.2.1 Document Format

The SRS document will be created in Microsoft Word and saved as a .pdf file.

1.2.2 Document Header

The SRS Document header will include: the document title (TrainTogether) and the document version number.

1.2.3 Document Footer

The SRS Document Footer will include the page number and date of the document.

1.2.4 Document Headings

The SRS Document Headings will be formatted at Times New Roman, font size 18, bold.

1.2.5 Document Subheadings

The SRS Document Subheadings will be formatted at Times New Romans, font size 14, bold.

1.2.6 Document Font

The font used throughout the SRS Document will be Times New Roman, size 12.

1.2.7 Document List

For lists, bullet points will be used, while numberings will be used for steps and procedures.

1.2.8 Document Tables

Tables will be used in this SRS Document to present data and information in a structured format.

1.2.9 References

All references to external documents will be cited in the document and included in the reference list at the end of the document.

1.3 Intended Audience

This SRS document is intended for all stakeholders involved in the making and Implementation of the TrainTogether software system, including:

1.3.1 Software Developers

To understand the constraints and technical details of the system

1.3.2 Project Managers

To ensure the project is on track with time, within budget, and according to client's needs

1.3.3 Quality Control Testers

To create test cases to ensure the system meets the functional, non-functional and specific requirements.

1.3.4 End Users

End users, including customers and backend users, will use this SRS as a reference to understand how the TrainTogether software works, its features and their responsibilities when using the software.

1.4 Product Scope

TrainTogether is a web application and the ultimate fitness corner companion designed to help users achieve their fitness goals. TrainTogether can be used as a platform for users to find the nearest fitness corner and access to a wide range of pre-designed workout programs that are tailored to users' fitness level, goals and the facilities at the fitness corner. At the same time, users can also use this web application to find restaurants that suit their fitness goals.

2.0 Overall Description

2.1 Product Perspective

TrainTogether is a web application that can be used across all browsers including but not limited to Google Chrome, Mozilla Firefox, Safari and Microsoft Edge. TrainTogether aims to provide a platform for users to search for nearby fitness corners and customise an exercise plan based on its location and facilities.

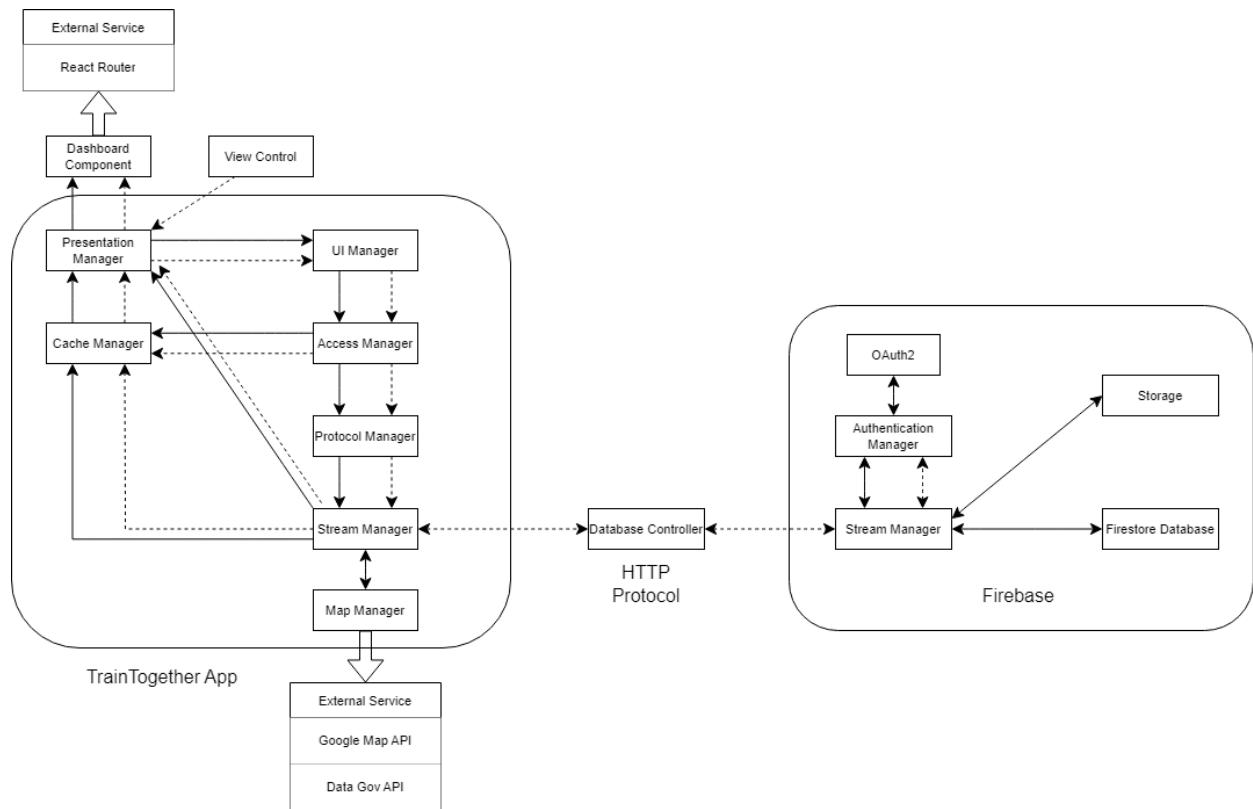


Diagram 1: System Client-Server Architecture Diagram of the TrainTogether Web Application

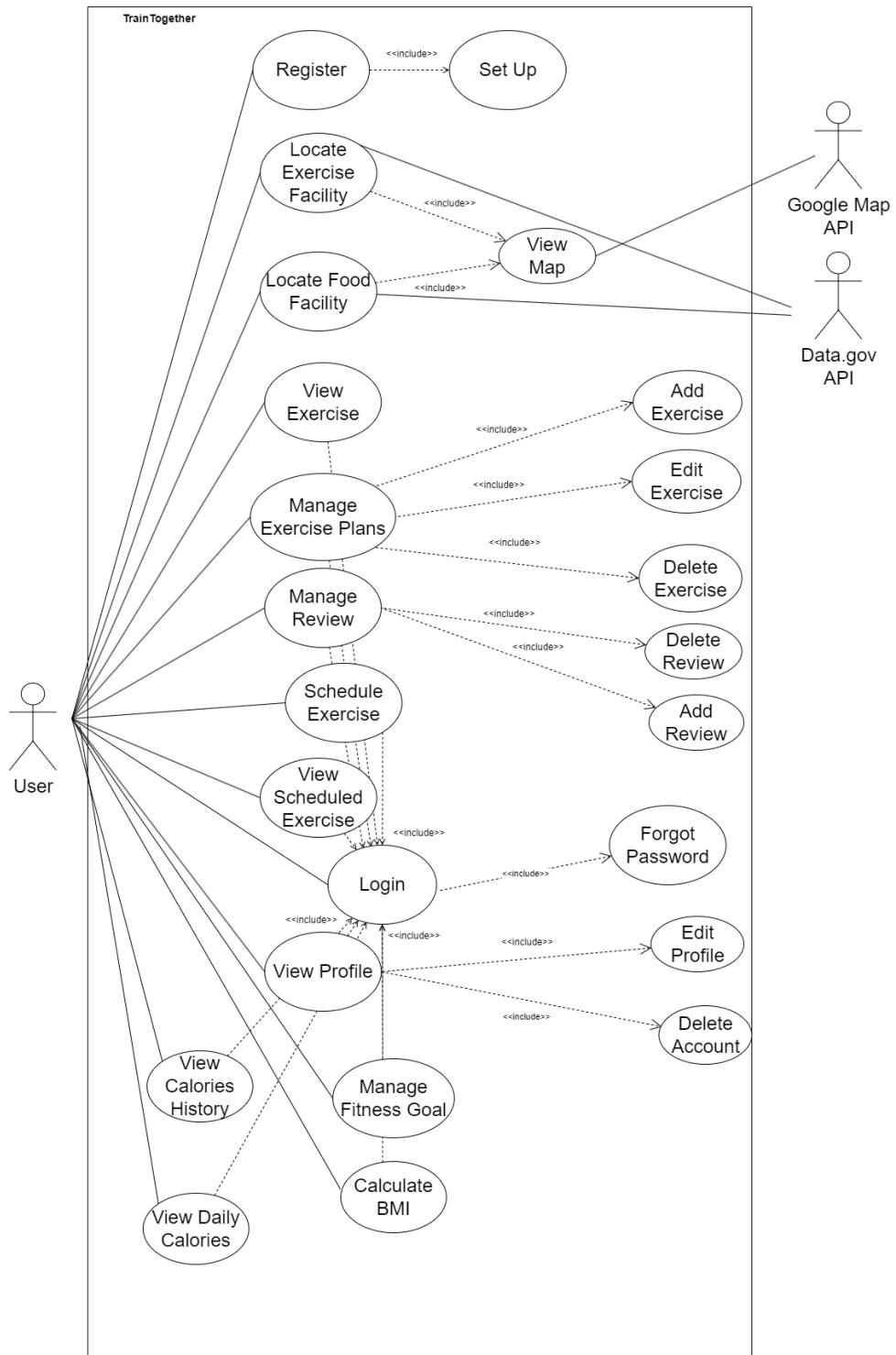
The TrainTogether application is a new, self-contained product which makes use of certain existing systems and Application Programming Interfaces (APIs) like Google Maps API and Data.gov API.

Shown on the left box above are all the systems in place to help the application run and interact with external services and systems.

The right hand side box shows the firebase systems which will assist our TrainTogether app with the fetching and storing of user information and transactions.

The database controller in the middle of the 2 boxes will act as the link between our application client and server database, with the use of HTTP Protocol.

2.2 Product Functions



The TrainTogether must minimally be able to:

- Register users with a new account
- Allow users to login to their account with a unique identifier for account and password
- Allow users to locate exercise facilities (fitness corners) and food facilities
- Manage and edit exercise plans to suit their own needs
- Schedule exercises

- Edit Profile
- Manage and edit fitness goals based on their current fitness levels
- Review and rate exercises
- View Calories Burnt

2.3 User Classes and Characteristics

There is only 1 main user class for TrainTogether, that is users who are interested in using our TrainTogether to create a workout plan at their nearby exercise facilities. The characteristics of our users are as followed :

- Users who are interested in fitness and exercise.
- They may have different levels of fitness experience, from beginners to advanced.
- They may have different fitness goals, such as weight loss, muscle building, or general fitness.
- They may have different preferences for workout types, durations, and intensities.
- They may want to track their progress and see improvements over time
- They may be looking for efficient and effective workouts that can be easily incorporated into their daily routine.
- They may expect features such as quick and easy workout creation, saved workout templates, and personalised recommendations

2.4 Operating Environment

TrainTogether currently works on any web browser including Google Chrome, Mozilla Firefox as well as Microsoft Edge. It will also be using Firebase to store users' data. Access to external APIs such as Data.gov API and Google Maps API is also required to provide additional information for our TrainTogether to work properly.

2.5 Design and Implementation Constraints

TrainTogether currently only works on web browsers and it's only available in English. At the same time, the usage of Firebase has a few limitations including limited querying capabilities, data migration problems and it doesn't work in countries that don't allow Google. Nonetheless, TrainTogether can be converted easily from web browsing to mobile app since React supports such conversion.

The use of data.gov API may not be that updated hence the food facilities and exercise facilities may not be the most time-accurate information.

2.6 User Documentation

A guide for how to use the TrainTogether Application will be made public for users to download. With the guide provided, it will increase the usability from the first interaction with the user as it is clear and easy to understand.

2.7 Assumptions and Dependencies

- 2.7.1 Availability of Internet Connection: It is assumed that users of the website application will have a stable and reliable internet connection to access the application and its features.
- 2.7.2 Standard Web Browsers: It is assumed that users will access the website application using standard web browsers, such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge, and the application will be designed and tested to be compatible with these browsers.
- 2.7.3 Proper Input from Users: It is assumed that users will provide accurate and valid input while using the website application, such as selecting appropriate exercises and following recommended guidelines for creating their own workout plans.
- 2.7.4 Authentication and Authorization: It is assumed that proper authentication and authorization mechanisms will be in place to ensure that only authorised users are able to access and modify their own workout plans.
- 2.7.5 Third-Party APIs: The website application may have dependencies on third-party APIs for features such as exercise facilities data. Any changes or updates to these third-party APIs may impact the functionality of the website application.
- 2.7.6 External Services: The website application may rely on external services, such as cloud storage, email notifications, or payment gateways, which may have their own dependencies and limitations that can affect the overall performance and availability of the application.

3.0 External Interface Requirements

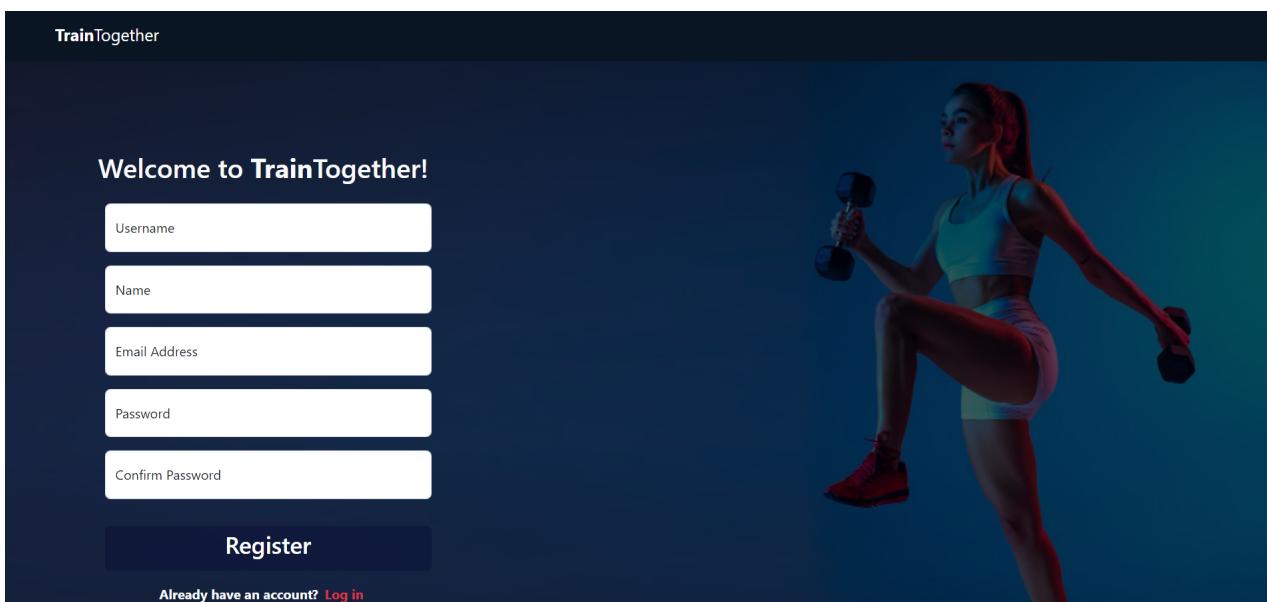
3.1 User Interfaces

Users can access the various user interfaces in TrainTogether as it is highly interlinked. Users will traverse through our applications based on their actions and interactions within our applications.

3.1.1 Register Page 1

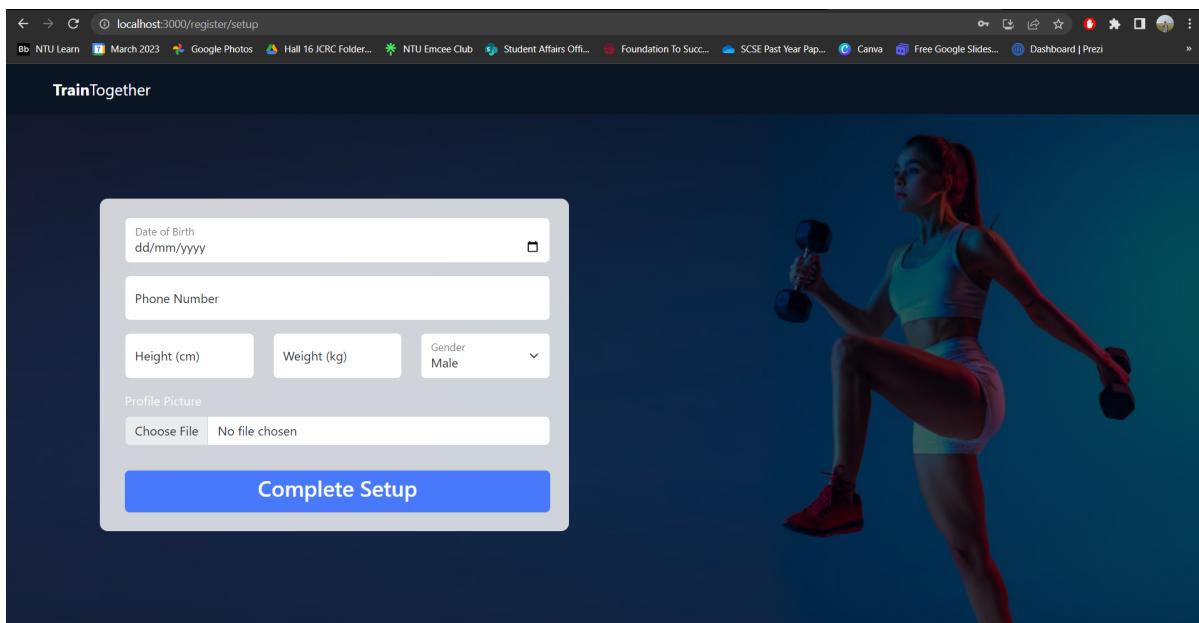
The first page of our register page requires our users to enter their username, name, email address, password and confirm password. Upon selecting the register, users will be directed to Register Page 2. The username must be at least 6 characters long while the password must contain at least 8 characters, consisting of one number, one Uppercase, one Lowercase and one number and a special character.

Alternatively, if the user has an account, the user can choose to log in by clicking on the red “Log In” button and TrainTogether will direct them to login page.



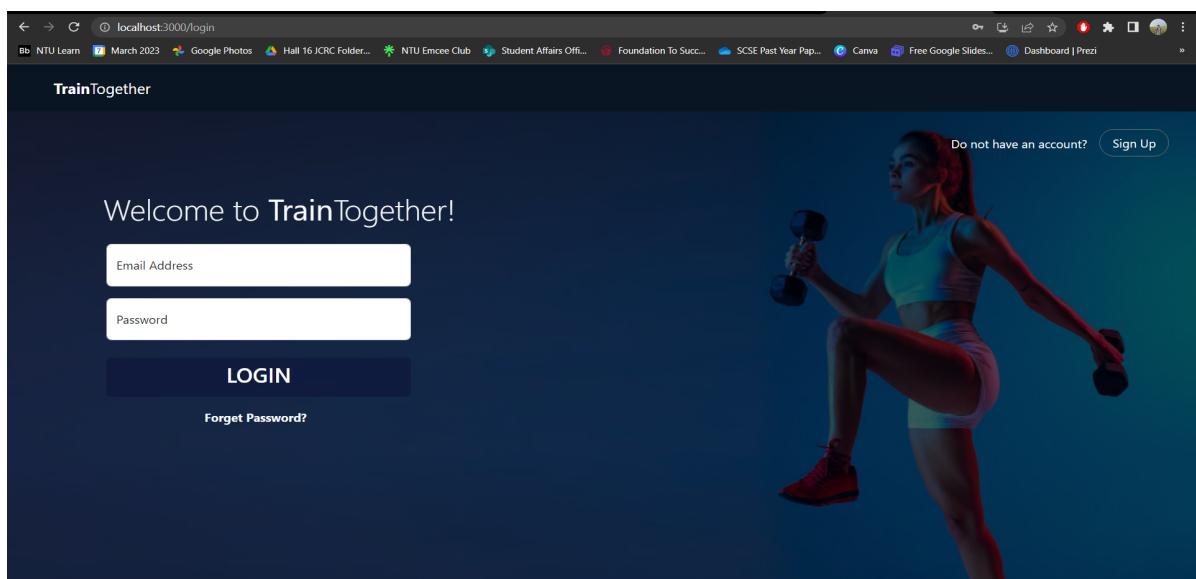
3.1.2 Register Page 2

The second page of the register page requires users to enter their date of birth with DD/MM/YYYY format, their phone number, height in cm, weight in kg, gender and a picture for their profile picture. After including all of their details, users can select Complete Setup and will be brought to our main landing page which is the dashboard.



3.2.3 Login Page

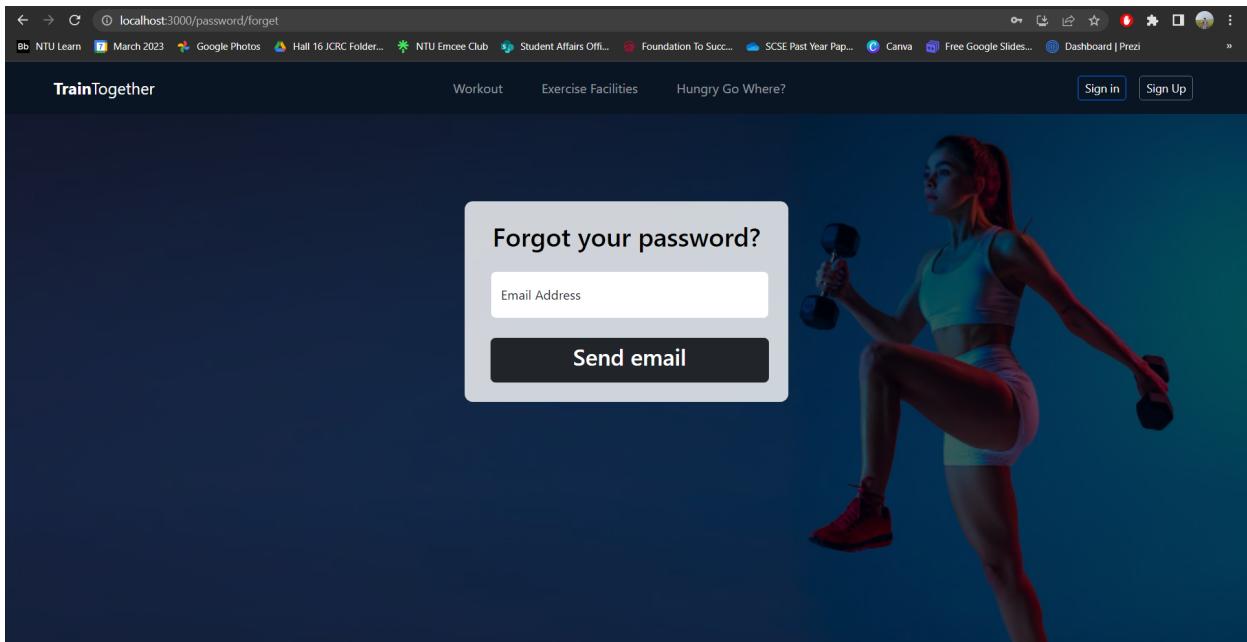
The login page requires users to input their email address and password. Once the user has entered their email address and password, upon selecting login, TrainTogether will check if the email address and password provided is accurate, and direct the user to our main landing page, which is the dashboard. In this page, users who do not have an account can select sign up so that they will be directed to Register Page 1. Users who selected Forgot Password will be directed to the Forgot Password page.



Software Requirements Specification for TrainTogether v5.3

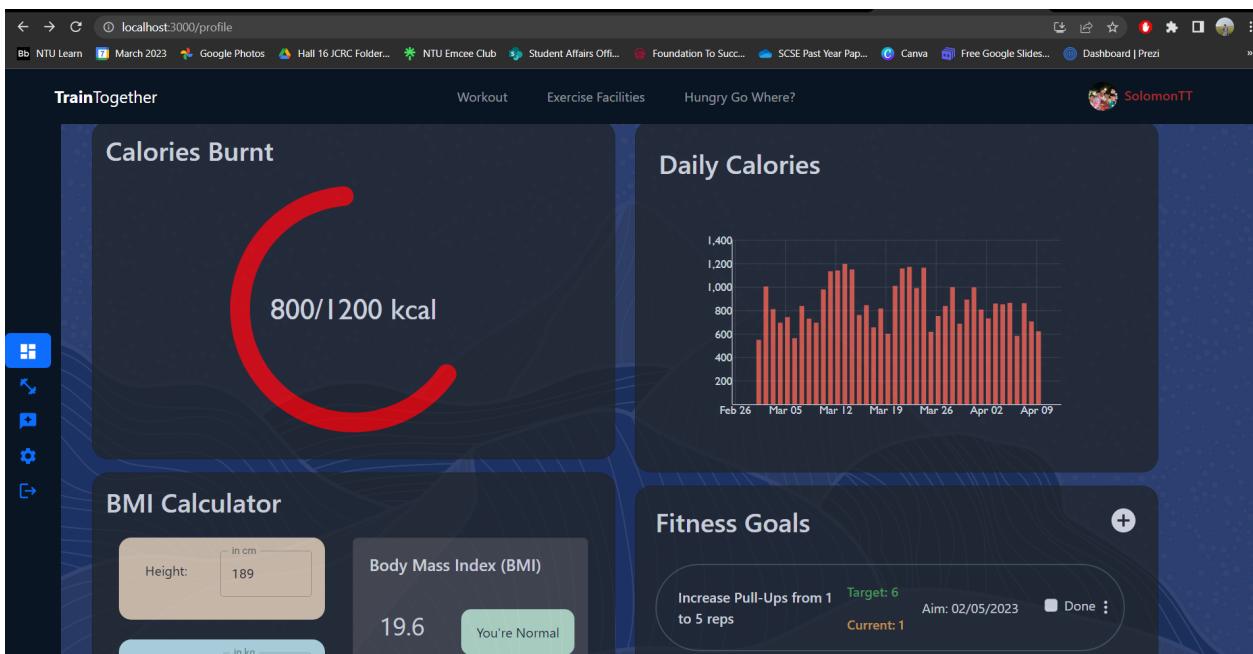
3.1.4 Forgot Password Page

At the Forgot Password Page, users are required to input their email address so that a link will be sent to the user to change his/her password.



3.1.5 Dashboard Page

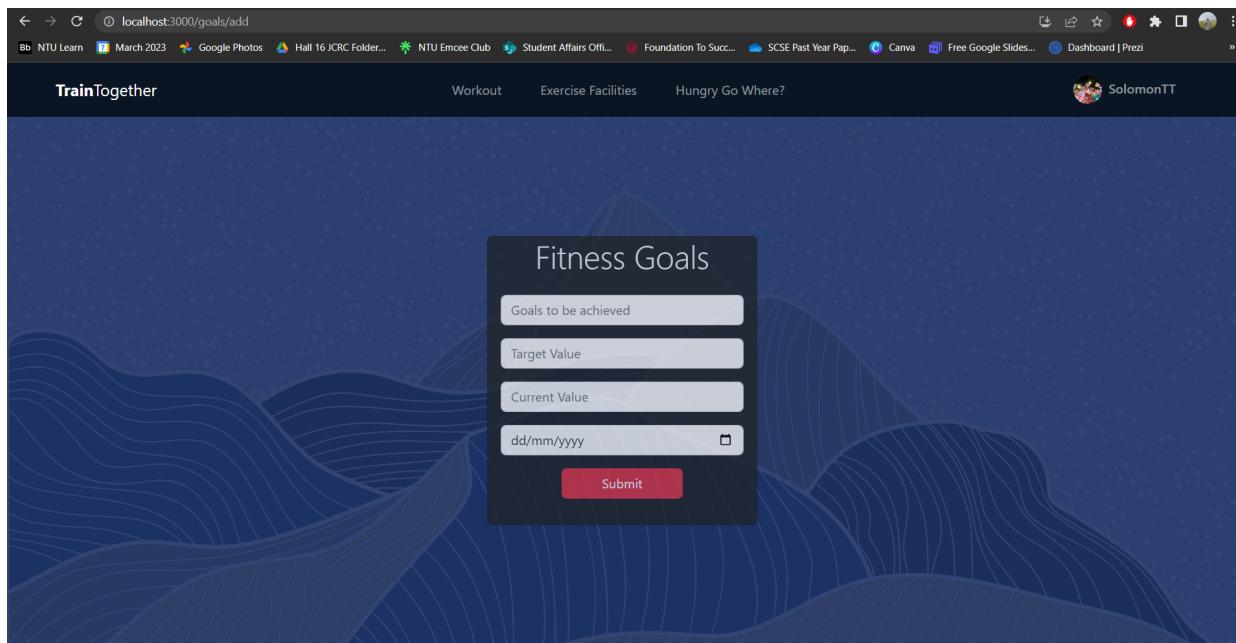
The Dashboard Page displays the calories burnt by the user for the day, located under “calories burnt”, a bar chart showing the calories burnt on a daily basis, located under “daily calories”, a BMI Calculator and Fitness Goals. Users can input their height and weight to calculate their Body Mass Index. Besides, users will be directed to the Fitness Goals page upon selecting the “plus icon” located beside Fitness Goals.



Software Requirements Specification for TrainTogether v5.3

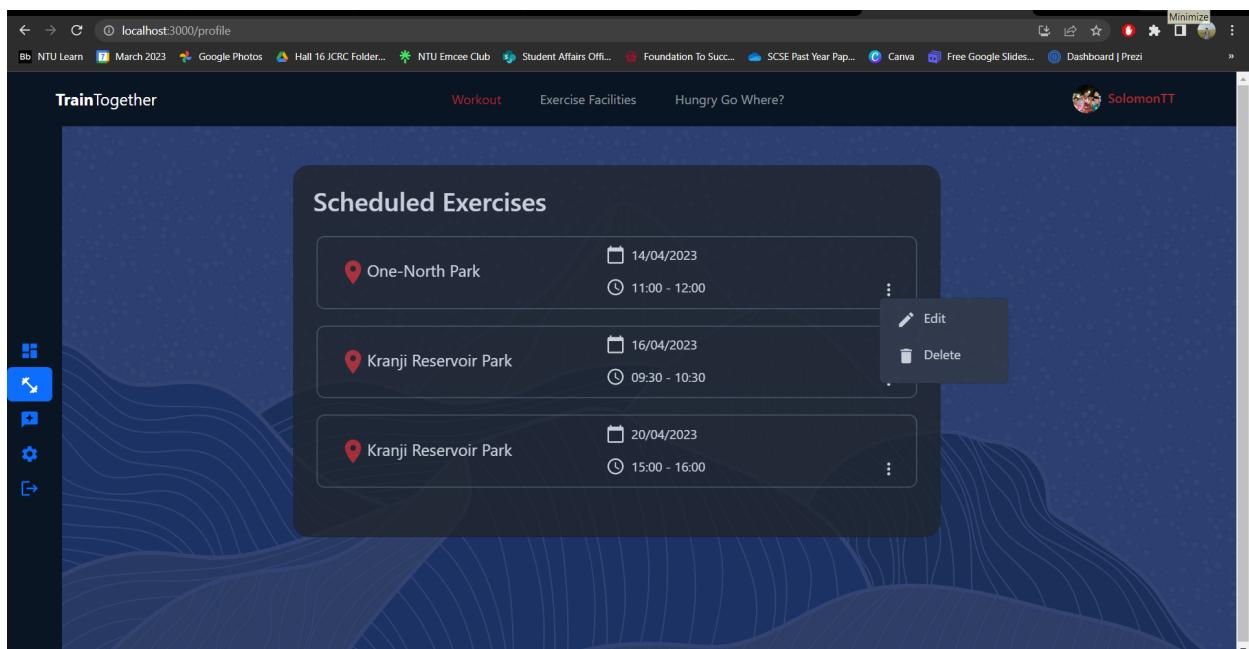
3.1.6 Fitness Goals Page

At the Fitness Goal page, user has to name the goal that user desires to achieve, eg: Push up, Target value and current value in numeric, and a deadline for the goal. Once the user has entered the required details, the user can select submit.



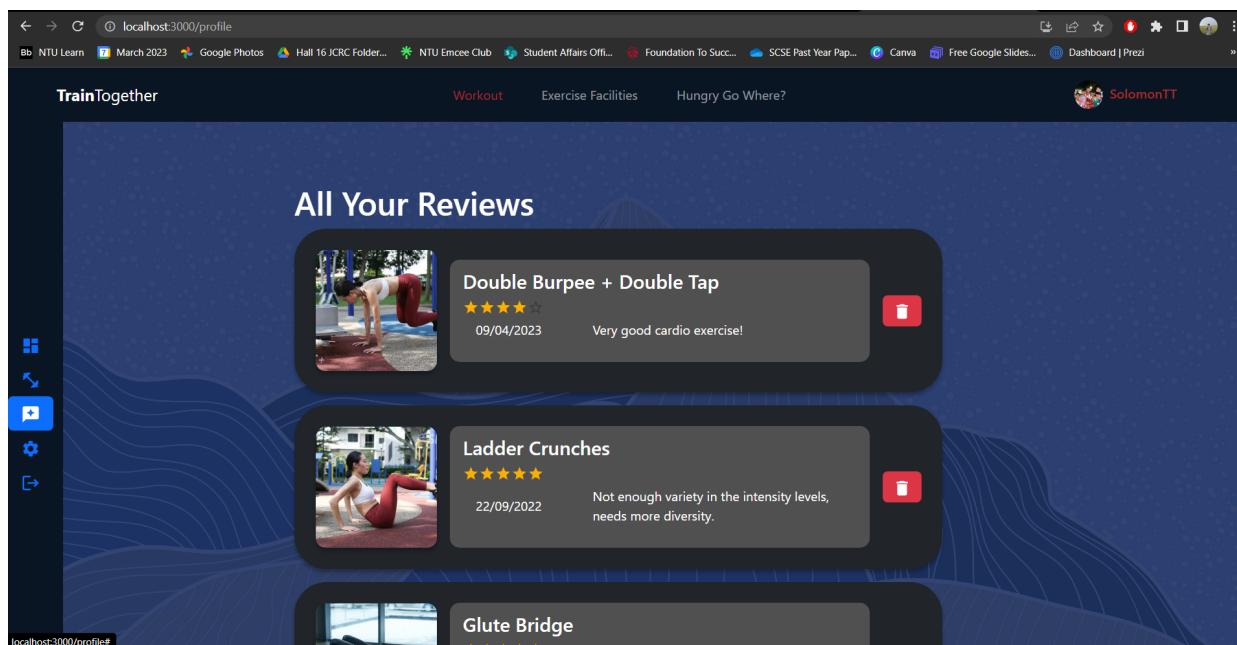
3.1.7 User Schedule Exercises Page

Selecting the dumbbell icon on the left sidebar of the dashboard page will direct users to the User Scheduled Exercises Page. In the User Schedule Exercises Page, users can see all of the scheduled exercise plans. Each scheduled exercise plan will show the name of the exercise plan, date and time. The three dots provide users the option to edit or delete the scheduled exercise plan.



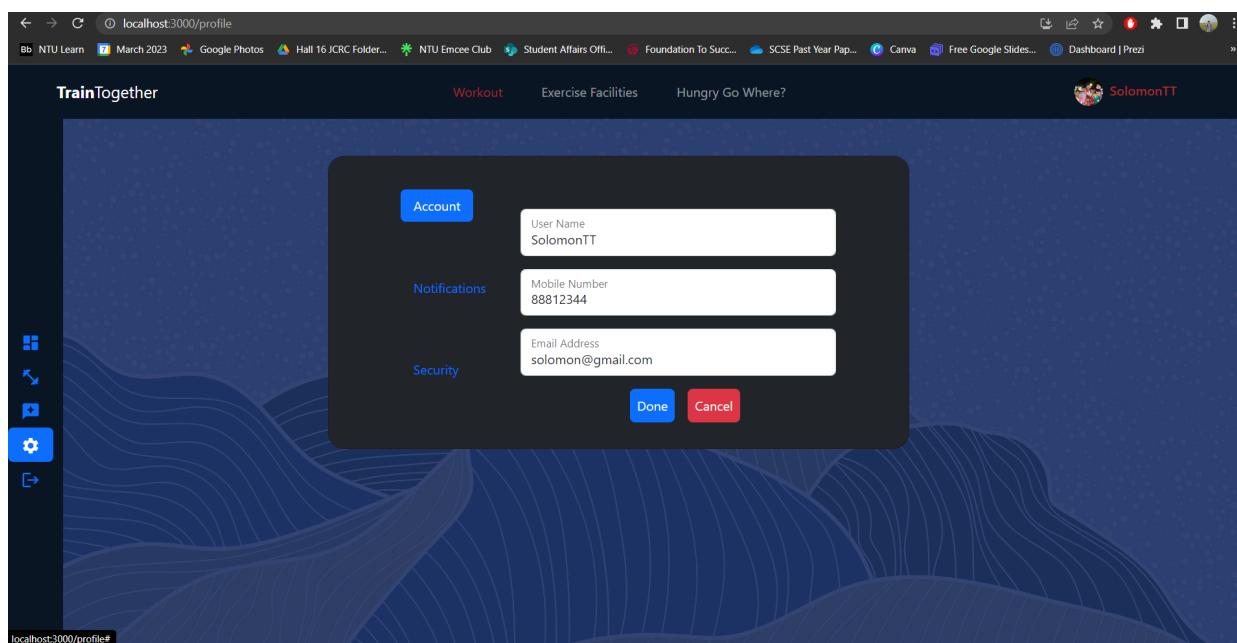
3.1.8 User Review Page

Selecting the chat box icon on the left sidebar of the dashboard will direct users to the user review page. Here, users can see all of the user's past reviews. Users can delete their reviews by selecting the “rubbish icon”.



3.1.9.1 Settings - Account Page

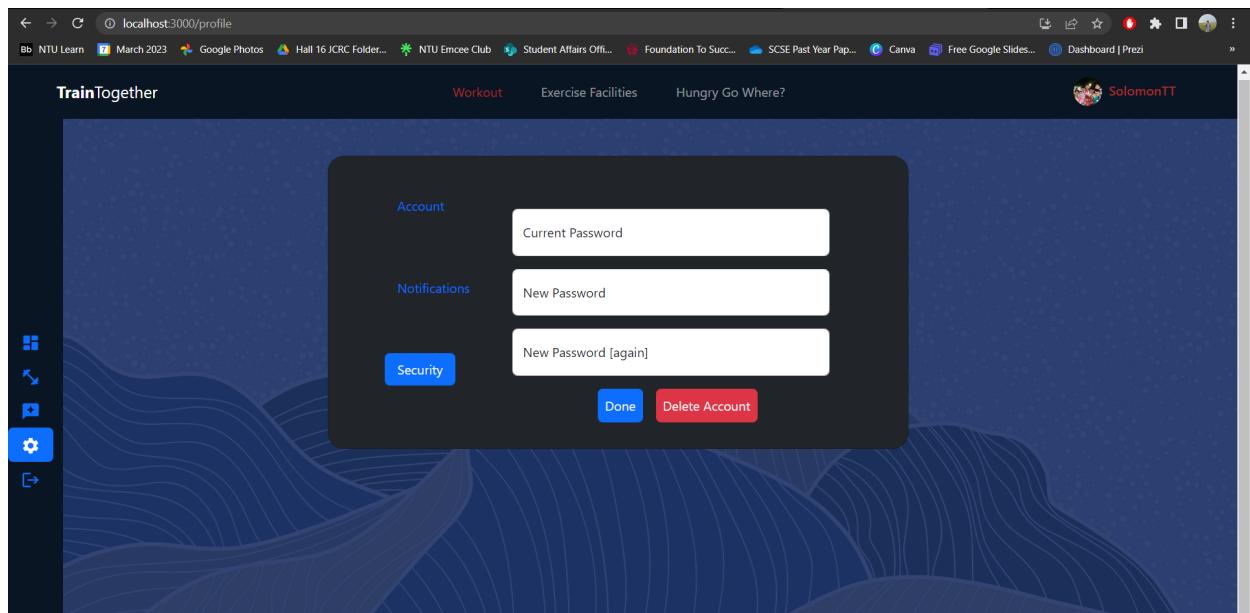
Users will be directed to the settings page upon selecting the gear icon on the left sidebar of the dashboard. Users can choose to rename their username, update mobile number and email address at this page. Once the user is done with the editing, the user just has to click Done.



Software Requirements Specification for TrainTogether v5.3

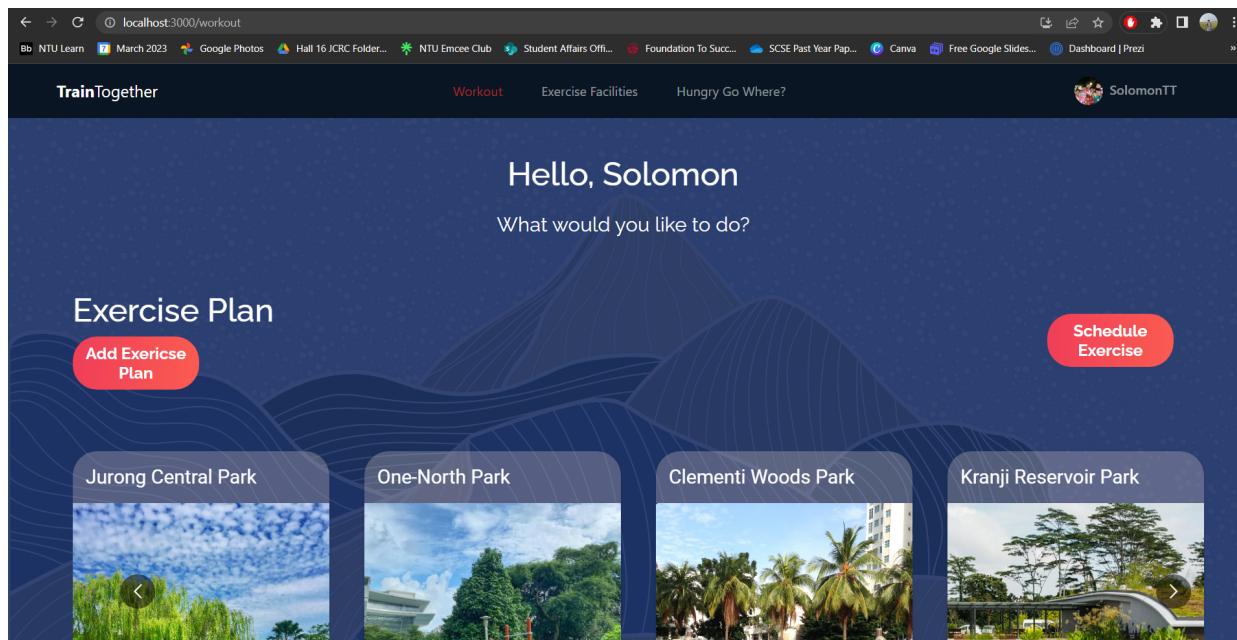
3.1.9.2 Settings - Security Page

Users have to enter their current password before entering their new password. At this page, users can choose to delete their account by selecting delete account.



3.1.10.1 Workout Page - Exercise Plans

Users will be directed to the Workout Page upon selecting Workout at the top tab. At this page, there is a “Add Exercise Plan” option or “Schedule Exercise” option. Users will be directed to the add exercise plan page upon clicking on the “Add Exercise Plan” icon whereas users will be directed to the Schedule Exercise page upon clicking on the “Schedule Exercise” icon. The list of Exercise plans will also be shown here in a carousel manner.



3.1.10.2 Workout Page - Exercise

Users will be directed to the exercise portion when they scroll down in the Workout Page. The list of exercises will also be shown here in a carousel manner, sorted according to exercise groups such as arms, back, cardio and core.

The screenshot shows the 'Workout' section of the TrainTogether website. At the top, there's a banner for 'JURONG CENTRAL PARK' with a photo of people exercising. Below it is a navigation bar with links like 'Workout', 'Exercise Facilities', and 'Hungry Go Where?'. On the right, there's a user profile for 'SolomonTT'. The main content area is titled 'Exercises' and features a horizontal carousel. Each card in the carousel represents a different exercise group: 'Arms' (a woman doing push-ups), 'Back' (a man doing pull-ups), 'Cardio' (a woman doing lunges), and 'Core' (a woman doing sit-ups). Each card has a small left and right arrow icon for navigating through more options.

3.1.11 - Exercise Plan Page

When a user selects an exercise plan, they will be directed to the exercise plans page. Here, the type of exercises tagged to the exercise plan, including its reps, sets, rests and calories burned will be shown here.

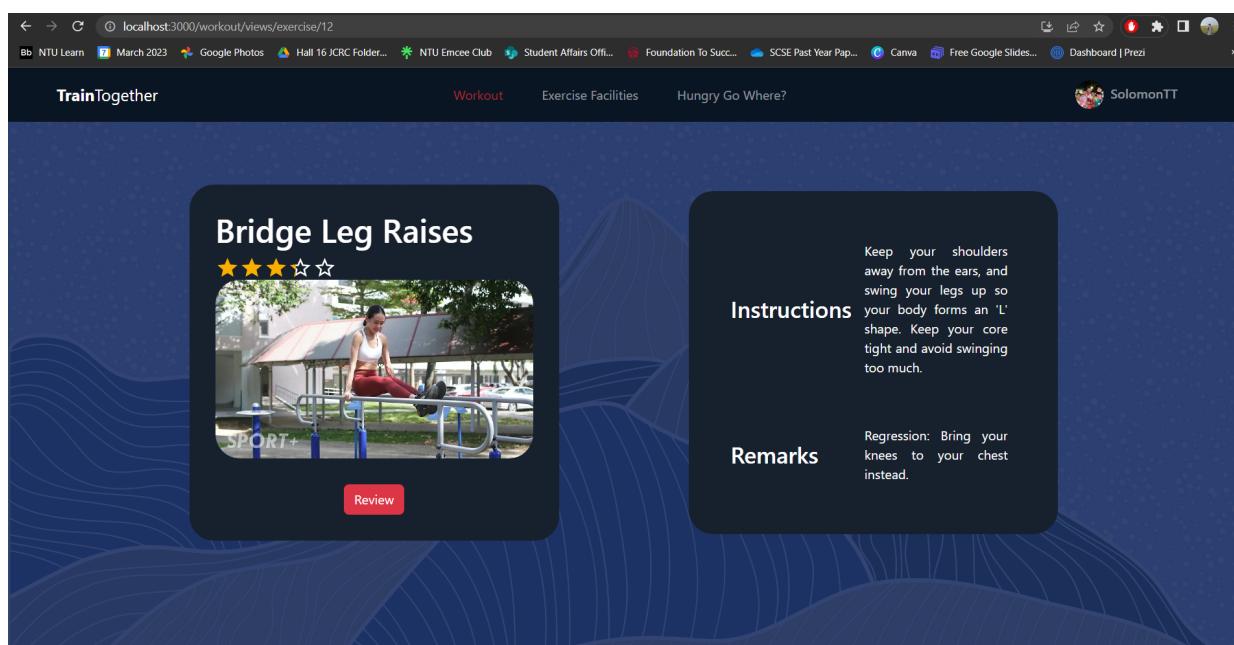
The screenshot shows the 'Exercise Plans' section for 'Jurong Central Park' on the TrainTogether website. The top navigation bar includes 'Workout', 'Exercise Facilities', and 'Hungry Go Where?'. The main content area is titled 'Exercise Plans' and 'Jurong Central Park'. Below this is a table showing a list of exercises with their details:

No.	Exercise	Reps	Sets	Rest (seconds)	Calories Burned
1	Pull-Up	15	3	120	33.75
2	Inclined Pull-ups	25	3	120	56.25
3	Dead Bug	50	3	120	75
4	Ladder Crunches	25	3	120	56.25
5	Hanging Ab Crunches	10	3	60	22.5
6	High Knees + Step Up	10	3	60	30

At the bottom of the table, it says 'Total Calories Burned' followed by '273.75'.

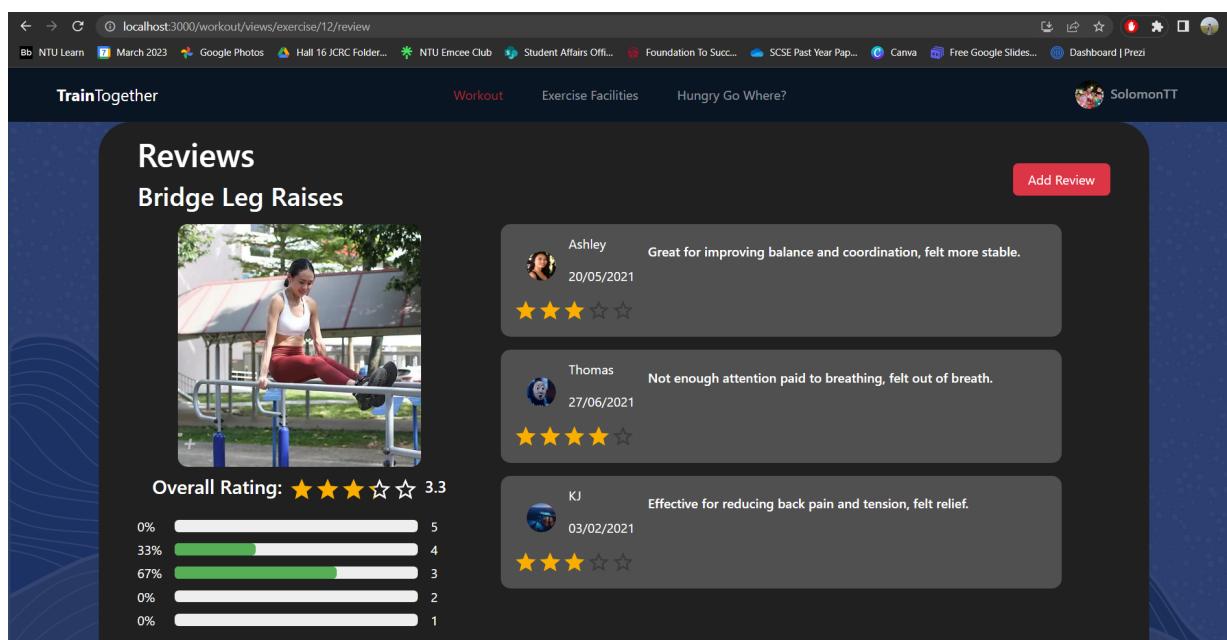
3.1.12 - Exercise Page

When a user selects an exercise from the exercise carousel, they will be directed to the exercise page. Here, the instructions and remarks of the exercise will be shown here. The average rating of the exercise out of 5 stars will also be shown here. There is also a review button that will direct users to the review page upon clicking.



3.1.13 - Review Page

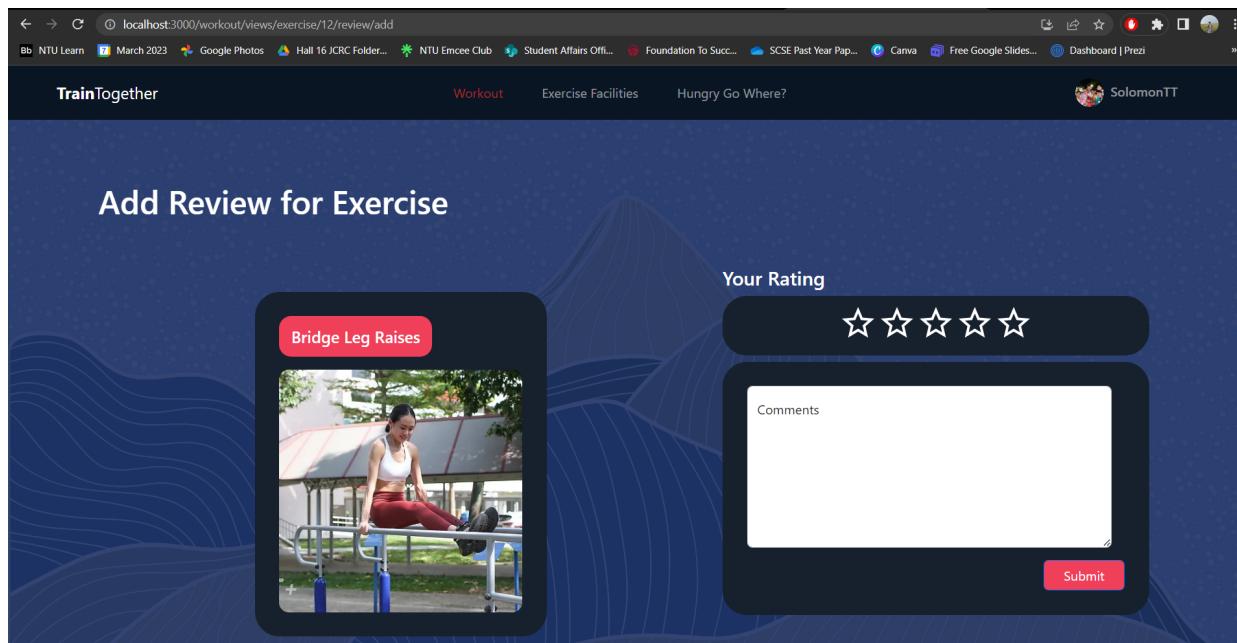
At the review page, users can see ratings and reviews made by other users. Users will be directed to the Add Review Page upon selecting the “Add Review” button.



Software Requirements Specification for TrainTogether v5.3

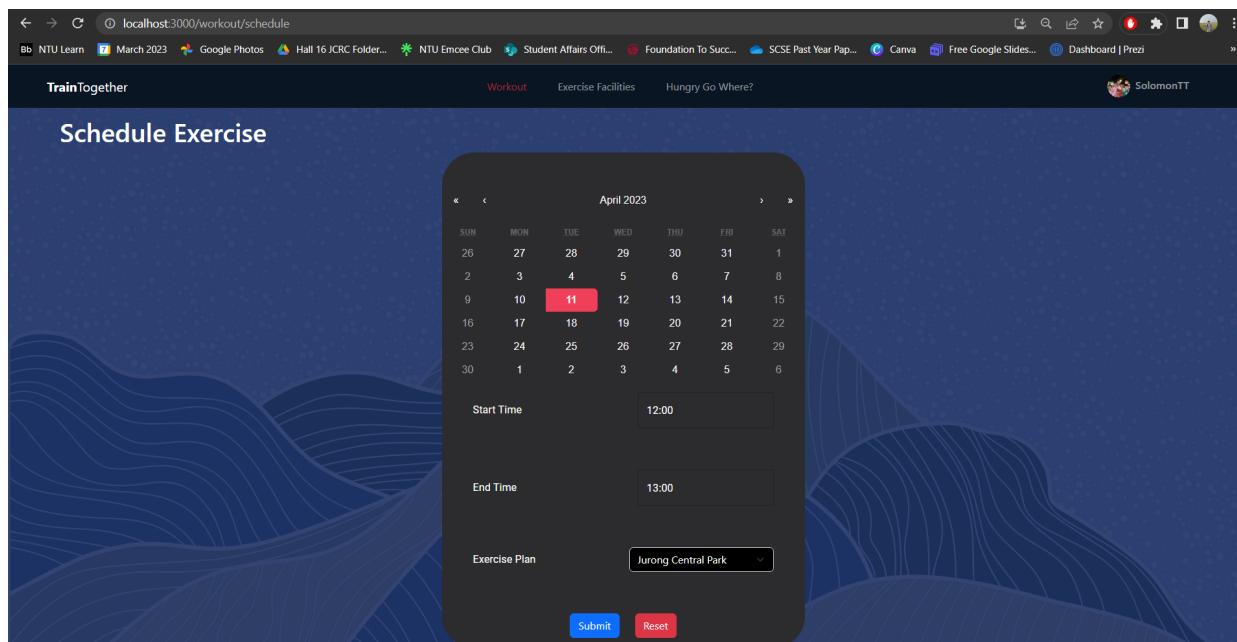
3.1.14 - Add Review Page

At the Add Review Page, users are to give their ratings for the exercise using 5 stars. Users can also include comments and reviews for the exercise. Once the user is done, the user can select submit and their reviews will be stored in the database.



3.1.15 - Schedule Exercise

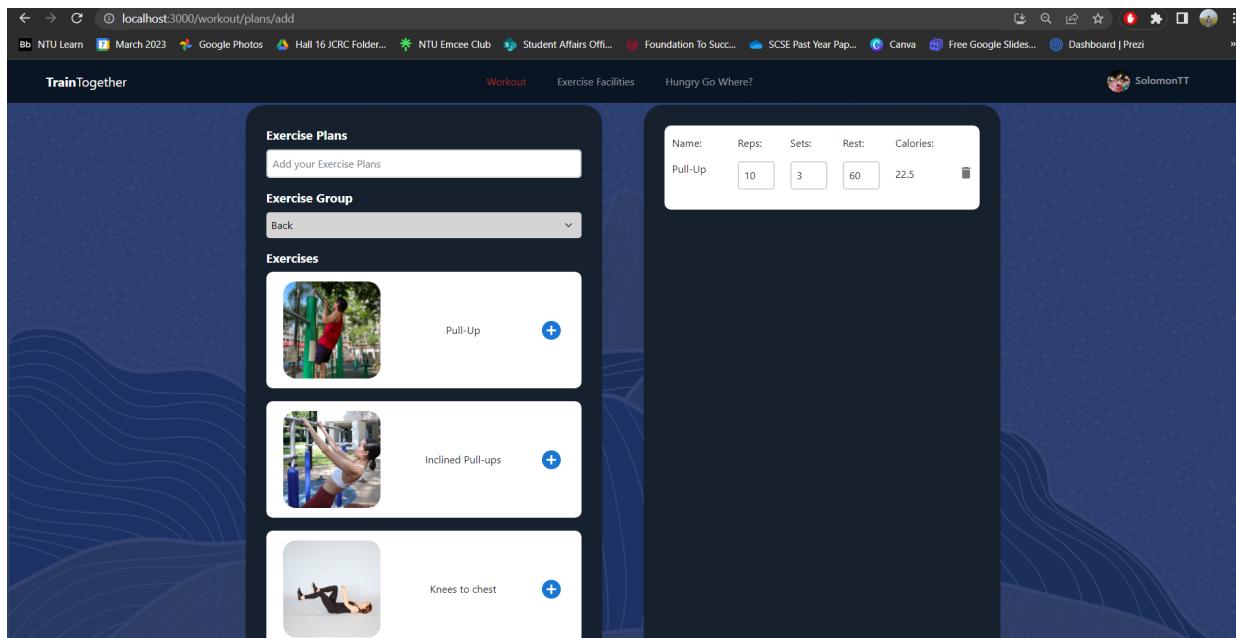
The schedule exercise page consists of a calendar, start time, end time input bar and a list of exercise plans. Users can schedule their exercise plans on this page. Users must include the date, start and end time of the exercise plan before clicking the submit button.



Software Requirements Specification for TrainTogether v5.3

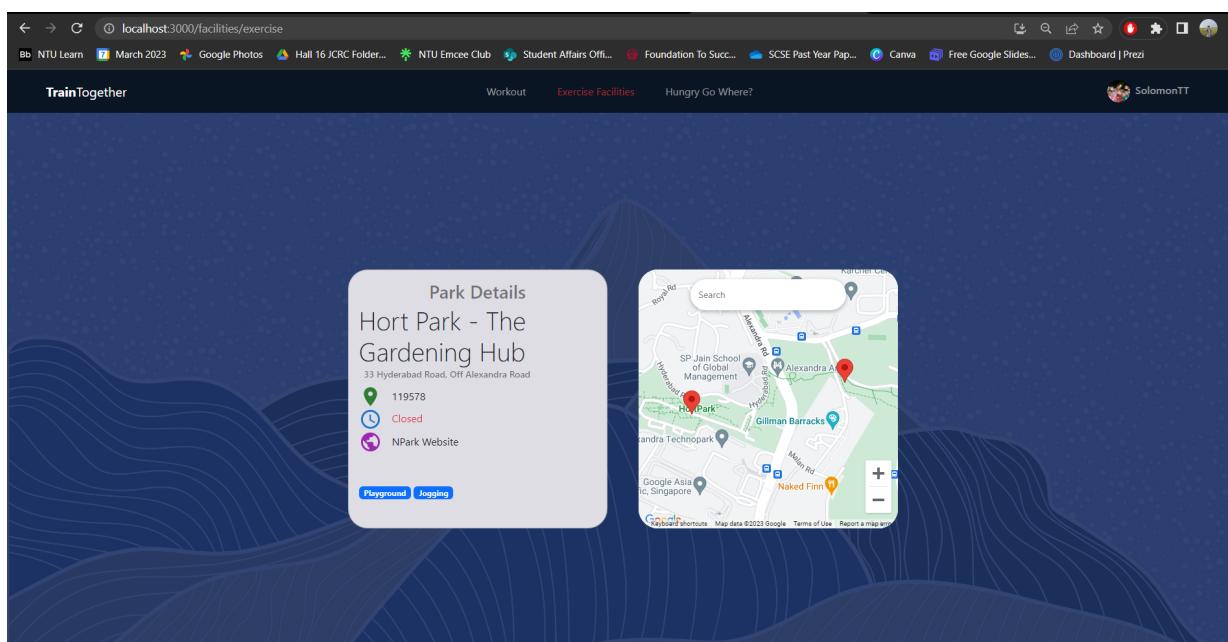
3.1.16 - Add Exercise Plan

Users can create their own customised set of exercise plans on this page. To do so, users have to name the exercise plan, add exercises from the list of exercises sorted based on their exercise group.



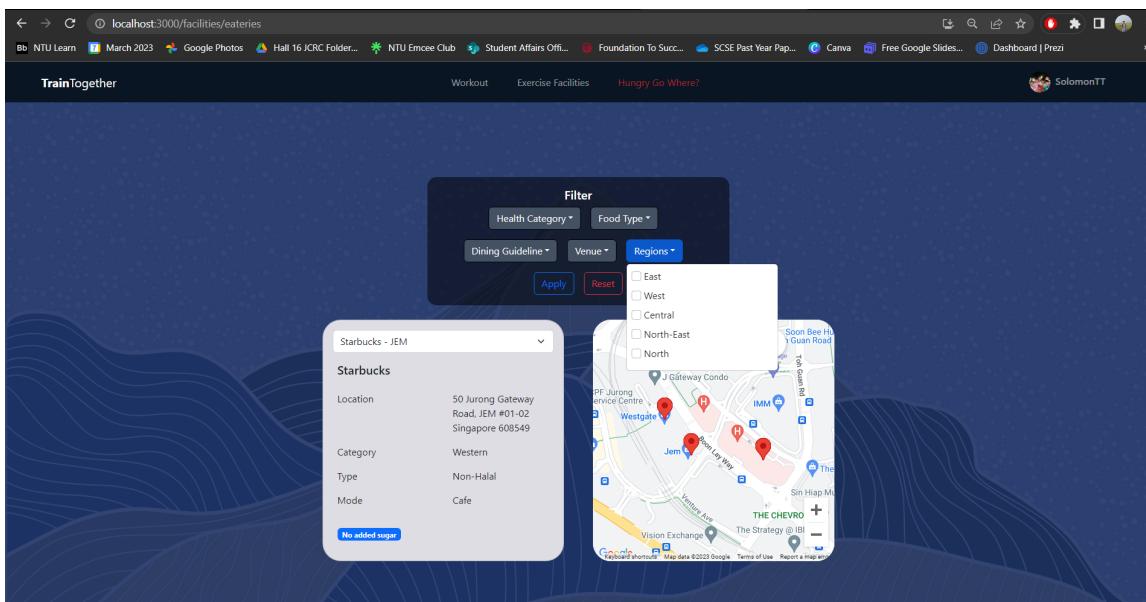
3.1.17 - Exercise Facilities

The exercise facilities page is TrainTogether's main landing page. It consists of a Google Map and the details of the fitness park. Users can make use of the search bar in the Google Map to search for the nearest fitness corner. Under the park details, the address, postal code, opening hours and website of the park will be displayed here.



3.1.18 Hungry Go Where

The Hungry Go Where page consists of a Google Map and the details of the food facility. Users can make use of the search bar in the Google Map to search for the nearest healthy food facility. At the same time, users can choose to filter the food facility based on its Health Category, Food Type, Dining Guidelines, Venue and Regions. Under the food facilities details, the location address, category, type and mode will be displayed.



3.2 Hardware Interfaces

TrainTogether is a web-based application that can be accessed using the web browser by desktop, laptops, tablets and phones.

To use the location-based feature of TrainTogether, the device must have access to GPS sensors and grant access to TrainTogether to use it. This allows TrainTogether to communicate with the GPS sensor in the device to provide the user's current location on Google Maps.

TrainTogether also must communicate with network components, like routers and servers to establish a connection to the internet to receive and transmit data. TrainTogether uses standard internet protocols, such as HTTP, TCP/IP and HTTPS to communicate with the network components.

3.3 Software Interfaces

TrainTogether is built using React and Firebase. React is a JavaScript library for building user interfaces. Firebase provides the backend for the application, which includes user authentication, storage, and real-time database. TrainTogether also integrates with external APIs such as Google Maps API and Data Gov API.

For React, we have the following third-party dependencies and their version numbers:

Dependency	Version	Description
@emotion/react	^11.10.6	CSS-in-JS library for React
@emotion/styled	^11.10.6	CSS-in-JS library for React
@mui/icons-material	^5.11.11	Material icons for MUI
@mui/material	^5.11.12	React components for MUI
@mui/x-date-pickers	^6.0.0	Date picker components for MUI
@react-google-maps/api	^2.18.1	React components for Google Maps API
@syncfusion/ej2-react-linear-gauge	^20.4.48	React components for Syncfusion's linear gauge
@testing-library/jest-dom	^5.16.5	Custom Jest matchers for testing DOM nodes
@testing-library/react	^13.4.0	Testing utilities for React
@testing-library/user-event	^13.5.0	Testing utilities for user events in React components
bootstrap	^5.2.3	CSS and JavaScript framework for building responsive, mobile-first websites
date-fns	^2.29.3	Date utility library for JavaScript
dayjs	^1.11.7	Date utility library for JavaScript
firebase	^9.18.0	Firebase SDK for building web and mobile applications
formik	^2.2.9	Library for building forms in React
prop-types	^15.8.1	Runtime type checking for React props
react	^18.2.0	JavaScript library for building user interfaces
react-bootstrap	^2.7.0	Bootstrap components built with React
react-calendar	^4.0.0	Calendar component for React
react-dom	^18.2.0	Entry point for React DOM rendering

react-hot-toast	^2.4.0	Toast notification library for React
react-multi-carousel	^2.8.2	Carousel component for React
react-router-bootstrap	^0.26.2	React Router components for Bootstrap
react-router-dom	^6.8.0	Declarative routing for React
react-scripts	^5.0.1	Configuration and scripts for Create React App
victory	^36.6.8	Charting library for React
web-vitals	^2.1.4	Library for measuring web vital metrics
yup	^0.32.11	Library for schema validation

The list of dependencies could also be found in package.json.

3.4 Communications Interfaces

TrainTogether requires an internet connection and must be able to communicate with components such as routers and servers to connect to the Firebase backend. The application communicates with Firebase server via REST APIs over HTTPs protocols. Data transfer rate will be determined by the user's internet speed. The Firebase server provides data synchronisation, ensuring that the app always displays the most up-to-date information. Communication security is also provided by Firebase, which uses SSL/TLS encryption to protect data communication.

TrainTogether also requires communication with external APIs such as Google Maps API and Data.gov API.

4.0 Functional Requirements

- 4.1 Users must be able to register for a new account.
 - 4.1.1 Users must provide the following details: username, name, email, mobile number, password, height, weight and date of birth.
 - 4.1.1.1 TrainTogether must prompt the user to change username if the username has already been taken.
 - 4.1.1.2 TrainTogether must prompt the user to change username if the username does not contain at least 6 characters.
 - 4.1.1.3 TrainTogether must prompt the user to change password if the password didn't adhere to the system's requirement: at least 8 characters, at least 1 special character and letters in mixed cases.
 - 4.1.1.4 TrainTogether must prompt the user to change the email if the email does not contain the "@" character.
 - 4.1.1.5 Users may choose to upload a picture to be used as their profile picture.
 - 4.1.1.6 TrainTogether must create a new account for the user if all the requirements are fulfilled.

- 4.2 Users must be able to login with their valid email and password.
 - 4.2.1 TrainTogether must be able to validate the user's username and Password.
 - 4.2.2 TrainTogether must direct the user to the main landing page after validating the user's credentials.

- 4.3 Users must be able to query for exercise facilities based on the location.
 - 4.3.1 Users must be able to view a map of nearby exercise facilities such as fitness corners.
 - 4.3.2 Users must be able to view the details of the exercise facilities.
 - 4.3.2.1 The details of the exercise facilities must include details such as its address, postal code, opening hours and website.

- 4.4 Users must be able to query for food facilities based on the location.
 - 4.4.1 Users must be able to filter the food facilities based on health category
 - 4.4.1.1 The health category includes lower in sugar, higher in wholegrains, lower in calories, no added sugar, source of dietary fibre, eat 2+2 servings of fruits, vegetables daily.
 - 4.4.2 Users must be able to filter food facilities by food type.
 - 4.4.2.1 The food type must include drinks, western, desserts, snack and kuehs, bubble tea, Chinese, fast food, Malay, Thai, cut fruits, Japanese, Korean, local flavours, Fusion, Indian.
 - 4.4.3 Users must be able to filter food facilities by dining guideline, which is either Halal or non-halal.

- 4.4.4 Users must be able to filter food facilities by Venue
 - 4.4.4.1 The venue includes hawker centre, bakery, cafe, kiosk, restaurant, fast food, coffee shop, dessert, F&B on digital platform(s).
- 4.4.5 Users must be able to filter food facilities by regions
 - 4.4.5.1 The regions must include East, West, Central, North-East, North
- 4.5 Users must be able to browse exercises.
 - 4.5.1 Users must be able to view detailed instructions on how to perform the exercise.
 - 4.5.2 Users must be able to view the remarks, average ratings for each exercise.
- 4.6 Users must be able to manage their exercise plan.
 - 4.6.1 Users must be able to create new exercise plans.
 - 4.6.2 Users must be able to add new exercises to their exercise plan.
 - 4.6.3 Users must be able to remove exercises from their exercise plan.
 - 4.6.4 Users must be able to delete their exercise plans.
- 4.7 Users must be able to manage reviews for each exercise.
 - 4.7.1 Users must be able to add reviews by leaving a comment for the exercise.
 - 4.7.1.1 Users must be able to rate the exercise using out of 5 stars.
 - 4.7.2 Users must be able to view all reviews posted by other users of the exercise, including their own.
 - 4.7.3 Users must be able to delete their own reviews.
- 4.8 Users must be able to schedule their exercise.
 - 4.8.1 Users must be able to schedule an exercise plan by selecting a time and date for that particular exercise.
 - 4.8.2 Users must be able to view all of their scheduled exercises.
 - 4.8.3 Users must be able to remove scheduled exercises.
- 4.9 Users must be able to edit their profile.
 - 4.9.1 Users must be able to change their username, mobile number and email address.
 - 4.9.2 Users must be able to change their notification choice.
 - 4.9.2.1 Users must be able to change their notification choice for exercise reminders
 - 4.9.2.2 Users must be able to change their notification choice for email notification
 - 4.9.3 Users must be able to update their password.
 - 4.9.3.1 TrainTogether must prompt the user to enter the user's current password before updating the password.
 - 4.9.3.2 Users must enter a password with at least 8 characters, 1 special character, 1 uppercase, 1 lowercase and 1 number.

- 4.9.3.3 Users must be able to delete their account.
- 4.10 Users must be able to manage their fitness goals.
 - 4.10.1 Users must be able to add their fitness goals.
 - 4.10.1.1 Users must clearly specify the goal to be achieved.
 - 4.10.1.2 TrainTogether must prompt the user to enter a numeric value for their fitness goals.
 - 4.10.1.3 Users must be able to select a date as a deadline for that particular fitness goal.
 - 4.10.2 Users must be able to edit fitness goals.
 - 4.10.3 Users must be able to delete fitness goals.
- 4.11 Users must be able to calculate their Body Mass Index.
 - 4.11.1 TrainTogether must prompt the user to enter the user's height and weight to calculate the user's body mass index.
- 4.12 Users must be able to view their past calories burnt in a barchart format.
- 4.13 Users will be able to view the amount of calories burnt for the day.

5.0 Other Nonfunctional Requirements

5.1 Performance Requirements

- 5.1.1 TrainTogether must be compatible with the latest version of major web browsers (Chrome, Firefox, Safari, Edge)
- 5.1.2 The design must have clear navigation toolbars and consistent designs for easy navigability.
- 5.1.3 Response time of TrainTogether should be no longer than 3 seconds after user's input.
- 5.1.4 TrainTogether must be able to handle an increase in traffic by at least 100% without performance degradation.
- 5.1.5 TrainTogether must be able to store at least 1 million records.

5.2 Reliability Requirements

- 5.2.1 TrainTogether must ensure that errors should not crash the application.
- 5.2.2 TrainTogether must restart within 1 minute if it crashed.

5.3 Safety Requirements

- 5.3.1 TrainTogether must have clear and updated documentation for developers to understand and improve the app.
- 5.3.2 TrainTogether needs to have error prevention functions.
 - 5.3.2.1 TrainTogether must route users to another page when an error occurs with clear concise error description.
 - 5.3.2.2 Error Pages should appear when users navigate into an invalid page.

5.4 Security Requirements

- 5.4.1 TrainTogether must ensure that every user is valid and has the correct permission to perform actions.
 - 5.4.1.1 TrainTogether must have OAuth2 Authentication verification before allowing the user access to certain actions

6.0 Data Dictionary

Term	Definition
Exercise	A physical activity that the user engages in to improve their fitness and health.
Exercise Plan	A collection of exercises that a user has created or added to their workout routine, including details on the repetitions, rest time, and sets for each exercise.
Exercise Facility	A fitness corner where users can engage in physical activity.
Scheduled Exercise	A time and exercise facility where the user schedule his exercise plan
Fitness Goal	A target that the user sets for themselves, such as a certain number of calories burned or a specific exercise to be completed.
Location	The location of the user and exercise facilities, as provided by the device's GPS or manually entered by the user.
Reviews	Reviews that are left by users, which can be both qualitative and quantitative, pertaining to the suggested exercises.
Food Facility	An eatery which is part of Health Promotion Board's list of healthy eateries

7.0 Other Requirements

7.1 Installation

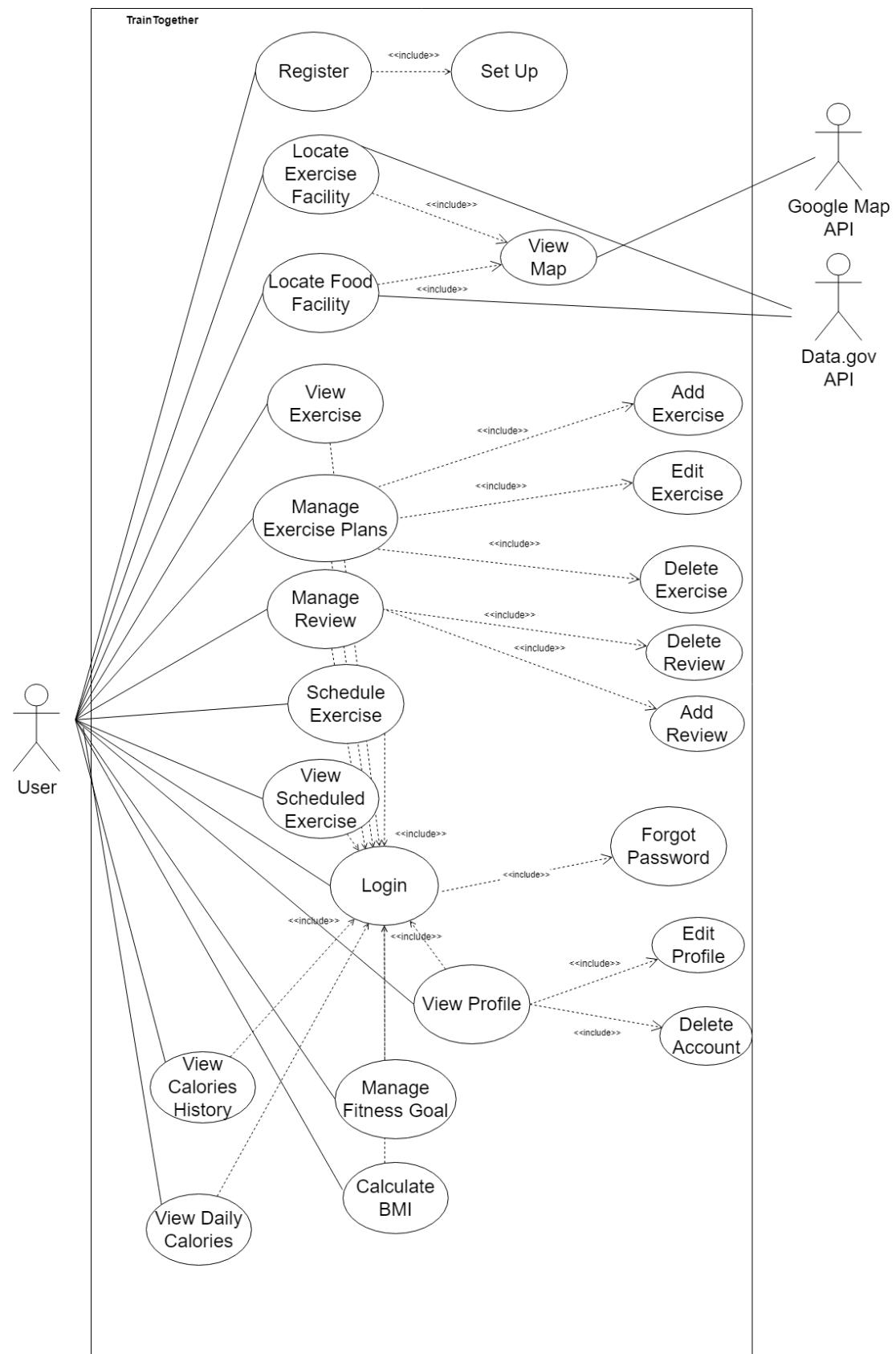
- 7.1.1 To install and run the app, you will need to have Node.js and npm installed on your computer. Node.js is a runtime environment that allows you to run JavaScript code outside of a browser. npm is a package manager that lets you install and manage dependencies for your project.
- 7.1.2 To install Node.js and npm, you can follow the instructions on their official website: <https://nodejs.org/en/>
- 7.1.3 Once you have Node.js and npm installed, you need to install the node modules that are required for the app. These are the libraries and packages that provide various functionalities and features for the app.
- 7.1.4 To install the node modules, you need to navigate to the project folder in your terminal and run the following command: `npm install`

7.2 Running the App

- 7.2.1 To run the app, you need to use the following command in your terminal: `npm start`
- 7.2.2 This will start a local server and open the app in your default browser. You can also access the app by visiting `http://localhost:3000` in your browser.

8.0 Appendix

8.1 Use Case Diagram



8.2 Use Case Description

Use Case ID:	001		
Use Case Name:	Register		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Tey Li Zhang Edmund
Date Created:	18 February 2023	Date Last Updated:	6 April 2023

Actor:	User
Description:	This use case allows users to create an account in the system to access the application's functions. The user has to enter a unique email address for registration and cannot use an email that has already been registered.
Preconditions:	1. User must be in the Register page
Postconditions:	1. System creates an account 2. System displays the profile page
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs their username, name, email address, password and re-enter their password into the form 2. User submits the form 3. System validates the username, name, email address, password and the re-entered password 4. System creates account 5. System triggers Set Up use case 6. System displays profile page
Alternative Flows:	<ol style="list-style-type: none"> 1. AF-S1: User already has account and clicks on login <ol style="list-style-type: none"> 1.1. System triggers Login use case
Exceptions:	001.EX.1 User provides incomplete information during registration <ol style="list-style-type: none"> 1. System will display "Missing information. Please enter the required fields." 2. System prompts the User to enter their details 3. Use case resumes at main flow step 1 001.EX.2 User provides an email address that has been used <ol style="list-style-type: none"> 1. System will display "Email has been used. Please register with another email or login with an existing account." 2. System prompts the User to enter their details 3. Use case resumes at main flow step 1
Includes:	Set Up, Register with Google Account
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • User enters in a valid phone number and email address
Notes and Issues:	NIL

Use Case ID:	002		
Use Case Name:	Set Up		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Tey Li Zhang Edmund
Date Created:	6 April 2023	Date Last Updated:	6 April 2023

Actor:	User
Description:	This use case allows users to fill in additional information for their account.
Preconditions:	<ol style="list-style-type: none"> 1. User must have already registered for an account 2. User must not have set up their account
Postconditions:	<ol style="list-style-type: none"> 1. System insert additional information into the User's account
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. System displays a form which contains Date of Birth, Phone Number, Height in cm, Weight in kg, dropdown menu for gender and file input for Profile Picture. 2. User inputs Date of Birth, Phone Number, Height, Weight, select their gender and upload their Profile Picture. 3. User submits the form. 4. System validates the form. 5. System insert additional information in the form to the User's account
Alternative Flows:	NIL
Exceptions:	<p>002.EX.1 User submitted invalid inputs in the form</p> <ol style="list-style-type: none"> 1. System displays error message for the invalid input 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 1
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • User entered their actual Phone Number, Date of Birth, Weight, Height, Gender
Notes and Issues:	NIL

Use Case ID:	003		
Use Case Name:	Login		
Created By:	Low Zhan Long	Last Updated By:	Tey Li Zhang Edmund
Date Created:	2 February 2023	Date Last Updated:	6 April 2023

Actor:	User
Description:	This use case allows the user to login into the system to access the relevant functions of the application. To login to the system, all users have to enter their unique email address and their password.
Preconditions:	<ol style="list-style-type: none"> 1. User does not have an existing session cookie 2. User has a valid account
Postconditions:	<ol style="list-style-type: none"> 1. System displays the profile page
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs his email into the System 2. User inputs his password into the System 3. User submits email and password to the System 4. System verifies the email and password 5. System displays the user's profile page
Alternative Flows:	<ol style="list-style-type: none"> 1. AF-S1: User forgets password and choose Forget Password <ol style="list-style-type: none"> 1.1. System trigger Forget Password use case
Exceptions:	003.EX.1 User enters Invalid email or password <ol style="list-style-type: none"> 1. System displays “Invalid email and/or password” message 2. System prompts for email and password 3. Use case resumes at main flow step 1.
Includes:	Forget Password
Extends:	NIL
Special Requirements:	<ul style="list-style-type: none"> • System should hash and salt the password entered by the User to verify if the hashed passwords match
Assumptions:	<ul style="list-style-type: none"> • Users should not be able to brute force and try out all the emails and passwords to gain entry into the system
Notes and Issues:	NIL

Use Case ID:	004		
Use Case Name:	Forget Password		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Tey Li Zhang Edmund
Date Created:	11 April 2023	Date Last Updated:	11 April 2023

Actor:	User
Description:	This use case allows the user to reset their password. A password reset link will be sent to the account that the user registered with.
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account registered 2. User is in Forgot Password Page
Postconditions:	<ol style="list-style-type: none"> 1. Password reset link is sent to User 2. System update password of User
Priority:	Medium
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. System displays Forgot Password 2. User enters their email address 3. User submits their email address 4. System validates the email address 5. System verifies the email address 6. System send reset link to the email address 7. System displays “A password reset link has been sent to your email.” 8. User click on reset link in email 9. User is redirected back to the TrainTogether 10. User enters their new password and re-enters their password for confirmation 11. User submit password 12. System validates the password 13. System updates the password for the User 14. System directs user to log in page
Alternative Flows:	NIL
Exceptions:	<p>004.EX.1 User submitted invalid email format</p> <ol style="list-style-type: none"> 1. System displays “Invalid Email Format” 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 1 <p>004.EX.2 User submitted email that is not registered</p> <ol style="list-style-type: none"> 1. System displays “Email is not registered to an account” 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 1 <p>004.EX.3 User submitted password that does not contain at least one Uppercase, Lowercase, one number and special character</p> <ol style="list-style-type: none"> 1. System displays “Password must contain at least one Uppercase, Lowercase, one number and special character” 2. System prompts User to enter their details again 3. Use case resumes at the main flow step 9
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	005		
Use Case Name:	Locate Exercise Facility		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Tey Li Zhang Edmund
Date Created:	6 April 2023	Date Last Updated:	6 April 2023

Actor:	User, Google Map, Data Gov API
Description:	This use case allows users to locate exercise facilities in Singapore and receive information about the name, location, opening hours and website of the exercise facility.
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must have been logged in 3. User must be in the Exercise Facility Tab
Postconditions:	NIL
Priority:	High
Frequency of Use:	5/session
Flow of Events:	<ol style="list-style-type: none"> 1. System triggers the View Map use case 2. User selects an Exercise Facility for more information 3. System query Data Gov API to get the Exercise Facility name, location, opening hours and website. 4. System displays the queried data
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	View Map
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	006		
Use Case Name:	Locate Food Facility		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Solomon Tneo
Date Created:	5 February 2023	Date Last Updated:	11 April 2023

Actor:	User, Google Map, Data Gov API
Description:	This use case allows users to locate healthy food facilities in Singapore and receive information about the store name, food type, dining guidelines, venue type and health category
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must be logged in 3. User must be in the Hungry Go Where tab
Postconditions:	<ol style="list-style-type: none"> 1. System shows details of the food store
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on HungryGoWhere in the Header 2. System triggers the View Map use case 3. User can filter Food Facility from the food type, dining guidelines, venue type and health category 4. User selects a location for more information 5. System query Data Gov API to get the Food Facility name, location, opening hours and website 6. System displays the store name, food type, dining guidelines, venue type and health category
Alternative Flows:	NIL
Exceptions:	<p>006.EX.1 User does not enter any food requirements 1. System will display all available food stores</p>
Includes:	NIL
Extends:	View Map
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	<ul style="list-style-type: none"> • There are 5 filters for users to use to filter food stores, including, health category, food type, dining guideline, venue and regions. • For health category, the available options are: lower in sugar, higher in wholegrains, lower in calories, no added sugar, source of dietary fibre, eat 2+2 serving of fruits, vegetables daily • For food type, the available options are: drinks, bread, cakes, cookies, pastries, Western, desserts, snacks and kuih, bubble tea, Chinese, fast food, Malay, Thai, cut fruit, Japanese, Korean, local flavours, Fusion, Indian • For dining guideline, the available options are Halal and Non-Halal • For venue, the available options are: hawker centre, bakery, cafe, kiosk, restaurant, fast food, coffee shop, dessert, f&b on digital platforms • For regions, the available options are: North, East, Central, North-East, West • User can click on 'Reset' to reset the filter

Use Case ID:	007		
Use Case Name:	View Map		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Solomon Tneo
Date Created:	5 February 2023	Date Last Updated:	11 April 2023

Actor:	User, Google
Description:	Able to navigate, scroll and zoom on location of exercise facility or food facility
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must have been logged in 3. User must be in the Exercise Facility Tab or Hungry Go Where Tab
Postconditions:	<ol style="list-style-type: none"> 1. System shows Map
Priority:	High
Frequency of Use:	5/session
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on Exercise facilities tab 2. System displays a map of the user's current location 3. System request Google for the Google Maps 4. User zoom in or out using punch gestures or using their mouse gestures 5. System updates map based on gestures 6. User scroll around the map by dragging it with their mouse 7. System updates map based on gesture 8. User can query for a specific location on the map by typing into the search bar 9. System updates map based on query.
Alternative Flows:	<ol style="list-style-type: none"> 1. AF-S1: User clicks on Hungry Go Where tab <ol style="list-style-type: none"> 1.1. System displays a map of the user's current location 1.2. System request Google for the Google Maps 1.3. User zoom in or out using punch gestures or using their mouse gestures 1.4. System updates map based on gestures 1.5. User scroll around the map by dragging it with their mouse 1.6. System updates map based on gestures 1.7. User can query for a specific location on the map by typing into the search bar 1.8. System updates map based on query.
Exceptions:	007.EX.1 User query a location that does not exist on the map <ol style="list-style-type: none"> 1. System will display "Location does not exist" 2. Use case resumes at the main flow at step 4
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	008		
Use Case Name:	View Exercises		
Created By:	Low Zhan Long	Last Updated By:	Solomon Tneo
Date Created:	2 February 2023	Date Last Updated:	11 April 2023

Actor:	User
Description:	Able to view exercises' instructions and remarks
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must login 3. User must have rights, privilege and access
Postconditions:	1. System displays exercise's instructions and remarks
Priority:	Medium
Frequency of Use:	1/Session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. User clicks on Workout 3. System displays Workout page 4. User able to view Exercises 5. User selects an exercise 6. System displays exercise's instructions and remarks
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	<ul style="list-style-type: none"> ● System is to sort exercises according to body parts, which are: Arms, Back, Cardio, Core and Legs

Use Case ID:	009		
Use Case Name:	Manage Exercise Plan		
Created By:	Low Zhan Long	Last Updated By:	Solomon Tneo
Date Created:	2 February 2023	Date Last Updated:	13 April 2023

Actor:	User
Description:	This use case allows users to manage their exercise plans. Users are able to add, edit and delete their exercise plan.
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must login 3. User must have rights, privilege and access
Postconditions:	<ol style="list-style-type: none"> 1. System displays exercise plans that user desires
Priority:	Medium
Frequency of Use:	1/Session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. User clicks on Workout 3. System displays Workout page 4. System displays list of exercise plans and add exercise plan button 5. User selects an exercise plan to view more information 6. System displays the name of the exercise plan and the list of exercises tagged in the exercise plan 7. System displays edit exercise plan button and delete exercise plan button 8. User can select to add, edit and delete exercise plan
Alternative Flows:	<ol style="list-style-type: none"> 1. AF-S7: User selects add exercise plan <ol style="list-style-type: none"> 1.1. System displays add exercise plan page 1.2. User enters name of exercise plan 1.3. User uploads a picture for the exercise plan 1.4. User selects on exercises to be added 1.5. System triggers Add Exercise use case 1.6. User clicks “Save”. 1.7. System adds the new exercise plan to the database 1.8. Use case resumes at main flow step 3 2. AF-S7: User selects delete exercise plan <ol style="list-style-type: none"> 2.1. System triggers Delete Exercise use case 2.2. System deletes the exercise plan from the database 2.3. Use case resumes at main flow step 3 3. AF-S7: User selects edit exercise plan <ol style="list-style-type: none"> 3.1. System displays edit exercise plan page 3.2. User enters new name of exercise plan 3.3. User uploads a new picture for the exercise plan 3.4. System triggers Edit Exercise use case 3.5. User selects submit 3.6. System saves exercise plans detail and updates database 3.7. Use case resumes at main flow at step 3
Exceptions:	009.EX.1. Exercise Plans have no exercises <ul style="list-style-type: none"> • System will display “no exercise” message 009.EX.2. Users did not create any exercise plans <ul style="list-style-type: none"> • System will display list of default exercise plans
Includes:	Login, Add Exercise, Edit Exercise, Delete Exercise

Software Requirements Specification for TrainTogether v5.3

Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	<ul style="list-style-type: none">• System has a list of default exercise plans for each fitness corners shown on the map

Use Case ID:	010		
Use Case Name:	Add Exercise		
Created By:	Low Zhan Long	Last Updated By:	Tey Li Zhang Edmund
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	Able to add exercise into an exercise plan
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must login 3. User must have rights, privilege and access 4. User must have created at least 1 exercise plan
Postconditions:	<ol style="list-style-type: none"> 1. Exercise is added into the exercise plan
Priority:	Medium
Frequency of Use:	2/Session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. User clicks on Workout 3. System displays Workout page 4. User clicks on add exercise plan 5. User enters name of exercise plan 6. User chooses an exercise group 7. System displays list of exercises tagged under the chosen exercise group 8. System selects the “plus icon” beside the exercise name 9. User edits the exercise’s reps and sets 10. User selects save 11. System adds the new exercise plan into the list of exercise plans as well as into the database and displays added exercises in the exercise plan.
Alternative Flows:	<ol style="list-style-type: none"> 1. AF-S3 Users selects edit exercise plan <ol style="list-style-type: none"> 1.1. System displays list of exercises tagged under the chosen exercise group 1.2. User adds exercise to the exercise plan 1.3. User edits the exercise’s reps and sets 1.4. User selects submit 1.5. System adds new exercise plan to the database 1.6. System displays added exercises in the exercise plan.
Exceptions:	008.EX.1. Users did not create any exercise plans <ul style="list-style-type: none"> • User is unable to add exercise
Includes:	Login, Manage Exercise Plan
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • User’s desired exercise is available in TrainTogether App • User is able to carry out the exercise add at user’s desired location
Notes and Issues:	<ul style="list-style-type: none"> • User does not have rights to add exercise for default exercise plans

Use Case ID:	011		
Use Case Name:	Edit Exercise		
Created By:	Low Zhan Long	Last Updated By:	Tey Li Zhang Edmund
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	Able to edit exercise in an exercise plan
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must login 3. User must have rights, privilege and access 4. User must have created at least 1 exercise plan 5. User must have added at least 1 exercise
Postconditions:	<ol style="list-style-type: none"> 1. Exercise edited in the exercise plan
Priority:	Medium
Frequency of Use:	0-1/Session
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on Workout 2. System displays Workout page 3. User clicks on an exercise plan 4. System displays list of exercises tagged under the chosen exercise plan 5. User selects edit exercise beside the exercise name 6. System displays list of exercises tagged under the chosen exercise group 7. User selects the “plus icon” beside the exercise name 8. User edits the exercise’s reps and sets 9. User selects submit 10. System updates the exercise plan in the database 11. System displays updated exercises in the exercise plan.
Alternative Flows:	NIL
Exceptions:	<p>009.EX.1. Users did not create any exercise plans</p> <ul style="list-style-type: none"> • User is unable to edit exercise <p>009 EX 2. Users did not add exercise in exercise plans</p> <ul style="list-style-type: none"> • User is unable to edit exercise
Includes:	Login
Extends:	Manage Exercise Plan
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • User’s desired exercise is available in TrainTogether App • User is able to carry out the exercise add at user’s desired location
Notes and Issues:	<ul style="list-style-type: none"> • User does not have rights to edit exercise for default exercise plans

Use Case ID:	012		
Use Case Name:	Delete Exercise		
Created By:	Low Zhan Long	Last Updated By:	Solomon Tneo
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	Able to delete exercise from an exercise plan
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must login 3. User must have rights, privilege and access 4. User must have created at least 1 exercise plan 5. User must have added at least 1 exercise
Postconditions:	<ol style="list-style-type: none"> 1. Exercise deleted in the exercise plan
Priority:	Medium
Frequency of Use:	0-1/Session
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on Workout 2. System displays Workout page 3. User clicks on an exercise plan 4. System displays list of exercises tagged under the chosen exercise plan 5. User selects the edit exercise besides the exercise name 6. System displays list of exercises tagged under the chosen exercise group 7. User selects the delete exercise plan button beside the exercise name 8. System remove exercise from the exercise plan 9. User selects submit 10. System displays updated exercise plans.
Alternative Flows:	NIL
Exceptions:	<p>010.EX.1. Users did not create any exercise plans</p> <ul style="list-style-type: none"> • User is unable to delete exercise <p>010 EX 2. Users did not add exercise in exercise plans</p> <ul style="list-style-type: none"> • User is unable to delete exercise
Includes:	Login
Extends:	Manage Exercise Plan
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	<ul style="list-style-type: none"> • User does not have rights to delete exercise for default exercise plans

Use Case ID:	013		
Use Case Name:	Manage Review		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	13 April 2023

Actor:	User
Description:	Able to Manage Review about Exercise
Preconditions:	1. User must have selected the Exercise Group
Postconditions:	NIL
Priority:	Low
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. System displays a list of Exercises under the Exercise Group 3. User selects an Exercise 4. System displays details about the selected Exercise 5. User clicks to 'Review' to view the reviews of the selected Exercise 6. System displays the list of reviews tagged to the particular Exercise
Alternative Flows:	<ol style="list-style-type: none"> 1. AF: User view reviews from dashboard <ol style="list-style-type: none"> 2. User clicks on reviews from dashboard 3. System displays all the reviews from all exercises 2. AF-S5: User selects add review <ol style="list-style-type: none"> 2. System displays add review page 3. System trigger Add Review use case 4. System adds the new review to the database 5. Use case resumes at main flow step 5 3. AF: Users selects delete review <ol style="list-style-type: none"> 2. User clicks on Profile 3. User clicks on Reviews component 4. System triggers Delete Review use case 5. Systems deletes the review from the database 6. Use case resumes at alternative flow step 3
Exceptions:	NIL
Includes:	Login, Add Review, Delete Review
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	014		
Use Case Name:	Add Review		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	13 April 2023

Actor:	User
Description:	Able to Add Review about Exercise
Preconditions:	<ol style="list-style-type: none"> 1. User must have selected the Exercise 2. User must be logged in
Postconditions:	<ol style="list-style-type: none"> 1. The new review is added to database
Priority:	Low
Frequency of Use:	1/Exercise Plan
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on Workout 2. System displays Workout page 3. User selects an Exercise Group 4. User select an Exercise 5. User clicks view Review on selected Exercise 6. User clicks add Review on selected Exercise 7. User input details of Review Details 8. User clicks Submit 9. System generates a Review for Exercise
Alternative Flows:	NIL
Exceptions:	<p>014.EX.1 User provides does not give a rating when submitting a review</p> <ul style="list-style-type: none"> • System will display an error message and prompts the user to give a rating
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

Use Case ID:	015		
Use Case Name:	Delete Review		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	Able to Delete Review about Exercise
Preconditions:	<ol style="list-style-type: none"> 1. User must have done a review about an exercise before 2. User must be logged in
Postconditions:	<ol style="list-style-type: none"> 1. The particular review is deleted from database
Priority:	Low
Frequency of Use:	1/Review
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on Profile 2. User clicks on Reviews component 3. User selects the Review 4. User clicks the delete button to delete the review 5. System deletes Review
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> User is sure of the choice to delete the review
Notes and Issues:	<ul style="list-style-type: none"> There is no editing of reviews available, hence a user is required to delete the review before adding a new one if there are any edits the user wants to make to a review

Use Case ID:	016		
Use Case Name:	Schedule Exercise		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	Able to schedule an exercise plan
Preconditions:	<ol style="list-style-type: none"> There must be at least one exercise plan available to be scheduled for
Postconditions:	<ol style="list-style-type: none"> Newly scheduled exercise plan is added to database
Priority:	Medium
Frequency of Use:	1/Exercise Plan
Flow of Events:	<ol style="list-style-type: none"> User logs in using the included use case Login User clicks onto the Schedule Exercise button User input schedule details User clicks Submit System updates newly scheduled exercise
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> User only schedules an exercise plan that is on a future date and time
Notes and Issues:	<ul style="list-style-type: none"> User is able to reset the Exercise Plan to the default one when inputting schedule details

Use Case ID:	017		
Use Case Name:	View Scheduled Exercise		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	Able to view Scheduled Exercise from profile
Preconditions:	1. User must have an account
Postconditions:	NIL
Priority:	Medium
Frequency of Use:	1/Session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. User clicks Profile 3. User selects Scheduled Exercise component 4. System displays all Scheduled Exercises belonging to the user
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	<ul style="list-style-type: none"> • Scheduled Exercises are ordered according to date from the closest to the furthest date, in the Scheduled Exercise component page

Use Case ID:	018		
Use Case Name:	View Profile		
Created By:	Lee Yen Foong Ernest	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	This use case allows the user to view his account from the settings function
Preconditions:	1. User must have an account
Postconditions:	1. User is able to view his account 2. User is unable to login into the same account
Priority:	Medium
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. User clicks Profile 3. System displays User dashboard 4. User clicks on the Settings button 5. User clicks on Account button 6. System displays basic profile details 7. If user selects edit profile, System triggers the Edit Profile use case 8. If user selects delete account, System triggers the Delete Profile use case
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login, Edit Profile, Delete Account
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	<ul style="list-style-type: none"> • User is able to view other types of profile details in the dashboard page, which is demonstrated in other use cases

Use Case ID:	019		
Use Case Name:	Edit Profile		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	This use case allows the user to edit and update his own profile information
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must be logged in
Postconditions:	<ol style="list-style-type: none"> 1. New Details have been updated
Priority:	Medium
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks onto Profile 2. System displays User dashboard 3. User clicks on Account Settings 4. User edits information about Account 5. User clicks Done Button 6. System update Profile
Alternative Flows:	NIL
Exceptions:	<p>019.EX.1 User enters invalid username, mobile number or email address</p> <ul style="list-style-type: none"> • System will display an error message and prompt user to re-enter their particulars again
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • User should enter in a valid username, mobile number and email address
Notes and Issues:	<ul style="list-style-type: none"> • User is also able to switch to Notifications segment to update notification settings • User is also able to switch to Security segment to change their password

Use Case ID:	020		
Use Case Name:	Delete Account		
Created By:	Low Zhan Long	Last Updated By:	Wan Kai Jie
Date Created:	2 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	This use case allows the user to delete his own account through the Account settings function
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must be logged in
Postconditions:	<ol style="list-style-type: none"> 1. User's account has been deleted 2. User is unable to login into the same account 3. User will be redirected back to the login page
Priority:	Medium
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks onto Profile 2. System displays User dashboard 3. User clicks on Account Settings 4. User clicks on Security segment 5. User clicks on Delete Account Button 6. System sends warning message to user 7. User clicks on Confirm Delete Account Button 8. System notifies user through email that account have been deleted 9. System logs User out of account and remove account
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	NIL
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ul style="list-style-type: none"> • All relevant databases will also delete details related to the User • User is sure of the choice to delete their account, and is not accidental
Notes and Issues:	NIL

Use Case ID:	021		
Use Case Name:	Manage Fitness Goals		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Lee Yen Foong Ernest
Date Created:	5 February 2023	Date Last Updated:	13 April 2023

Actor:	User
Description:	This use case allows the user to view all the fitness goals that he has created. User is able to add, edit and delete his fitness goals.
Preconditions:	<ol style="list-style-type: none"> 1. User must have an account 2. User must have at least one fitness goal when editing or deleting
Postconditions:	<ol style="list-style-type: none"> 1. User is able to view his updated fitness goals
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘Manage Fitness Goal Page’ component 5. System retrieves User’s fitness goals 6. System displays user’s fitness goals in ‘Fitness Goal’ component
Alternative Flows:	<ol style="list-style-type: none"> 1. AF-S7: User decides to add a fitness goal <ol style="list-style-type: none"> 1.1. User clicks on ‘+’ button to add a new fitness goal 1.2. System displays ‘Add Fitness Goal Page’ 1.3. User enter details of fitness goal 1.4. User clicks Submit button 1.5. System generates the updated fitness goals list 1.6. User clicks “Save”. 1.7. System adds the new fitness goal to the database 2. AF-S7: Users decides to edit fitness goal <ol style="list-style-type: none"> 2.1. User clicks on ‘Edit’ button to edit a fitness goal 2.2. System retrieves user’s fitness goals 2.3. System displays user’s fitness goals in ‘Edit Fitness Goal Page’ 2.4. User changes details of the fitness goal and submit 2.5. System updates the fitness goal and displays updated fitness goals list in ‘Fitness Goal’ component in Dashboard Page 3. AF-S7: Users decides to delete fitness goal <ol style="list-style-type: none"> 3.1. User clicks on ‘Delete’ button to delete a fitness goal 3.2. System deletes the fitness goal 3.3. System generates the updated fitness goals list
Exceptions:	<p>021.EX.1 User provides incomplete details when adding fitness goal</p> <ul style="list-style-type: none"> • System will display an error message and prompts the user to enter the details correctly <p>021. EX.2 User provides incorrect details when adding fitness goal</p> <ul style="list-style-type: none"> • System will display an error message and prompts the user to enter the details correctly

	021. EX.1 User provides incorrect details when editing fitness goal <ul style="list-style-type: none">● System will display an error message and prompts the user to enter the details correctly
Includes:	Login
Extends	NIL
Special Requirements:	NIL
Assumptions:	User enters all details of fitness goals correctly <ol style="list-style-type: none">1. Target value and current value must be an integer2. Target date cannot be in the past3. Target date cannot be more than 100 years in the future
Notes and Issues:	NIL

Use Case ID:	022		
Use Case Name:	Calculate BMI		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Lee Yen Foong Ernest
Date Created:	5 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	This use case allows the user to calculate BMI using a height and weight value
Preconditions:	1. User must have an account
Postconditions:	1. User is able to view his BMI index value
Priority:	Medium
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘BMI Calculator’ component 5. System retrieves User’s height and weight and calculates BMI index 6. System displays BMI index value in ‘BMI Calculator’ component
Alternative Flows:	<ol style="list-style-type: none"> 1. AF6: User can input custom height and weight value 7. User inputs a different height or weight value 8. System displays new BMI index value in ‘BMI Calculator’ component
Exceptions:	NIL
Includes:	Login
Extends:	NIL
Special Requirements:	NIL
Assumptions:	<ol style="list-style-type: none"> 1. User must input a Height and Weight value that is greater than 0
Notes and Issues:	NIL

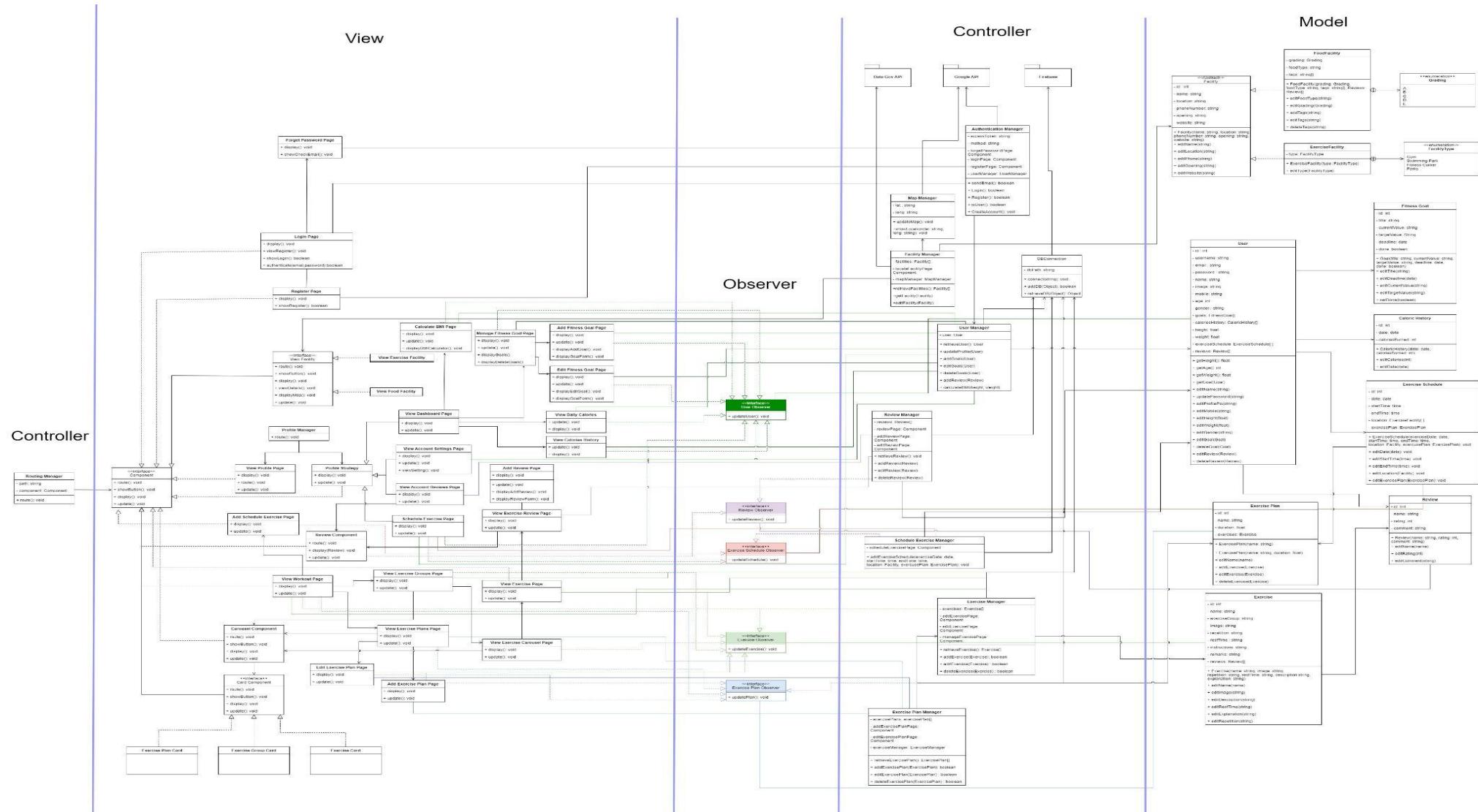
Use Case ID:	023		
Use Case Name:	View Calories History		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Lee Yen Foong Ernest
Date Created:	5 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	This use case allows the user to be able to view his calories burnt history
Preconditions:	1. User must have an account
Postconditions:	1. User can see his calories burnt in the past
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘Daily Calories’ component 5. System retrieves User’s calories history 6. System displays calories history in ‘View Calories History’ component
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

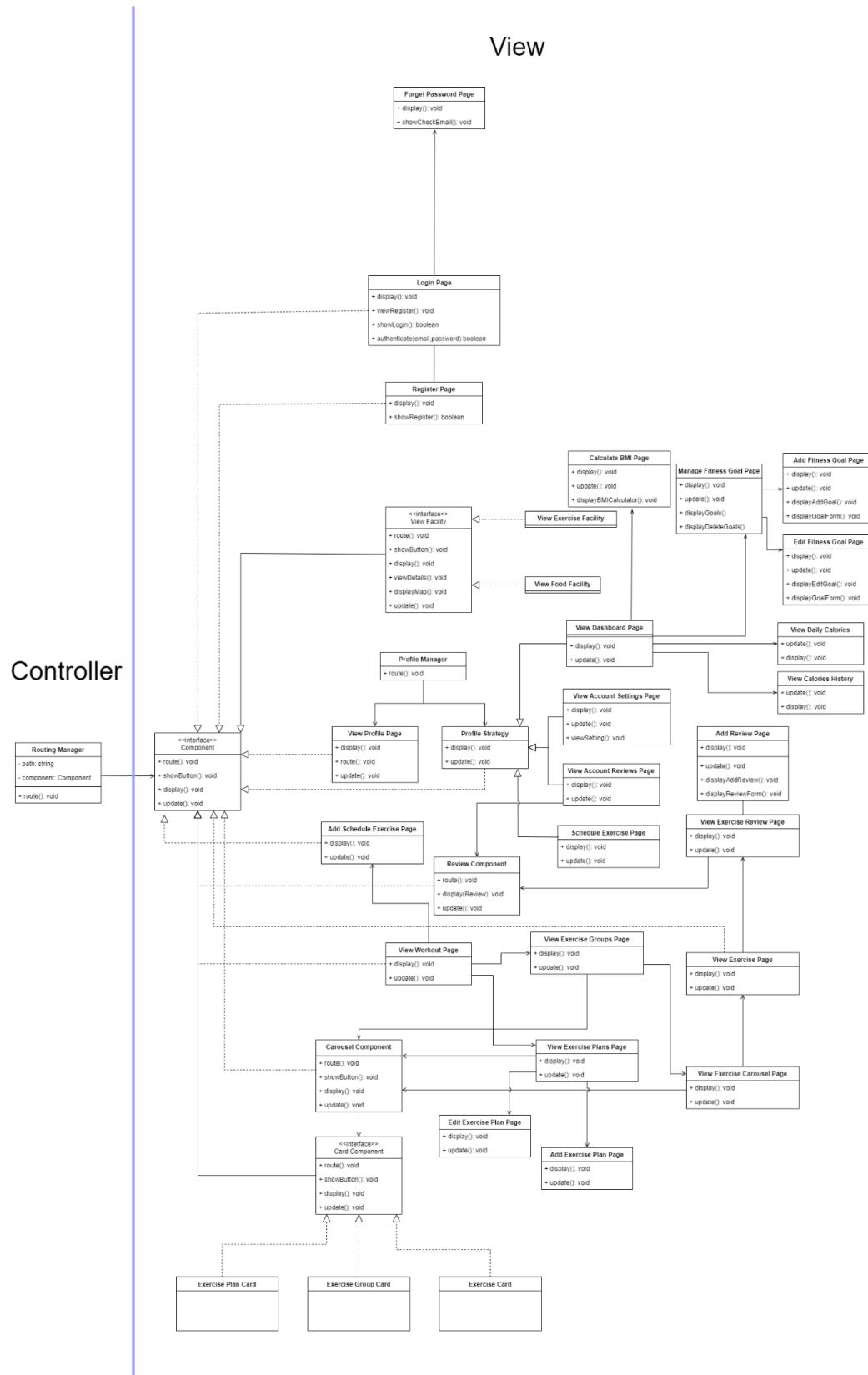
Use Case ID:	024		
Use Case Name:	View Daily Calories		
Created By:	Tey Li Zhang Edmund	Last Updated By:	Lee Yen Foong Ernest
Date Created:	5 February 2023	Date Last Updated:	12 April 2023

Actor:	User
Description:	This use case allows the user to be able to view his daily calories burnt
Preconditions:	1. User must have an account
Postconditions:	1. User can see his current calories burnt for the day
Priority:	High
Frequency of Use:	1/session
Flow of Events:	<ol style="list-style-type: none"> 1. User logs in using the included use case Login 2. System displays Profile Page 3. Profile Manager displays Dashboard Page 4. System renders ‘View Daily Calories’ component 5. System retrieves User’s calories burnt today 6. System displays calories burnt in ‘View Daily Calories’ component
Alternative Flows:	NIL
Exceptions:	NIL
Includes:	Login
Extends:	NIL
Special Requirements:	NIL
Assumptions:	NIL
Notes and Issues:	NIL

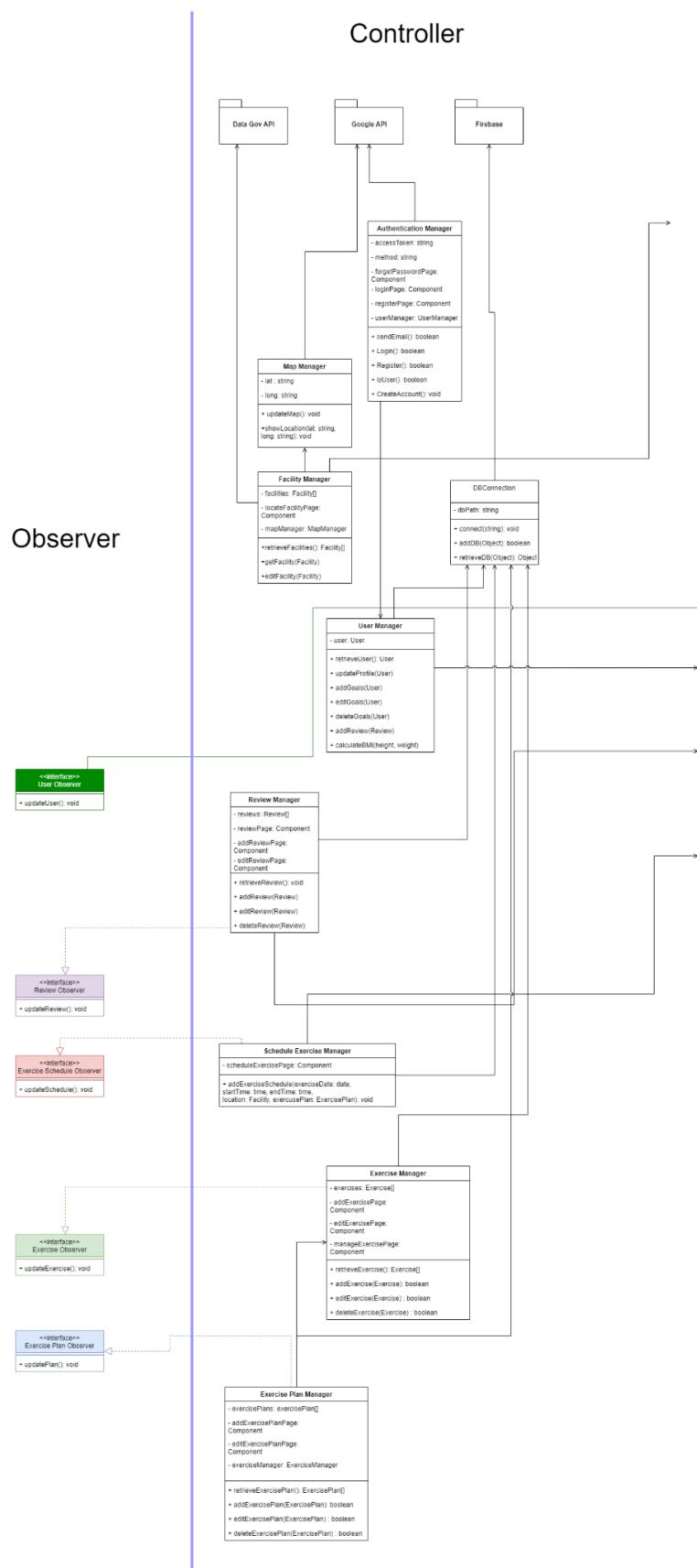
8.3 Class Diagram



8.3.1 View and Routing Manager Class Diagram

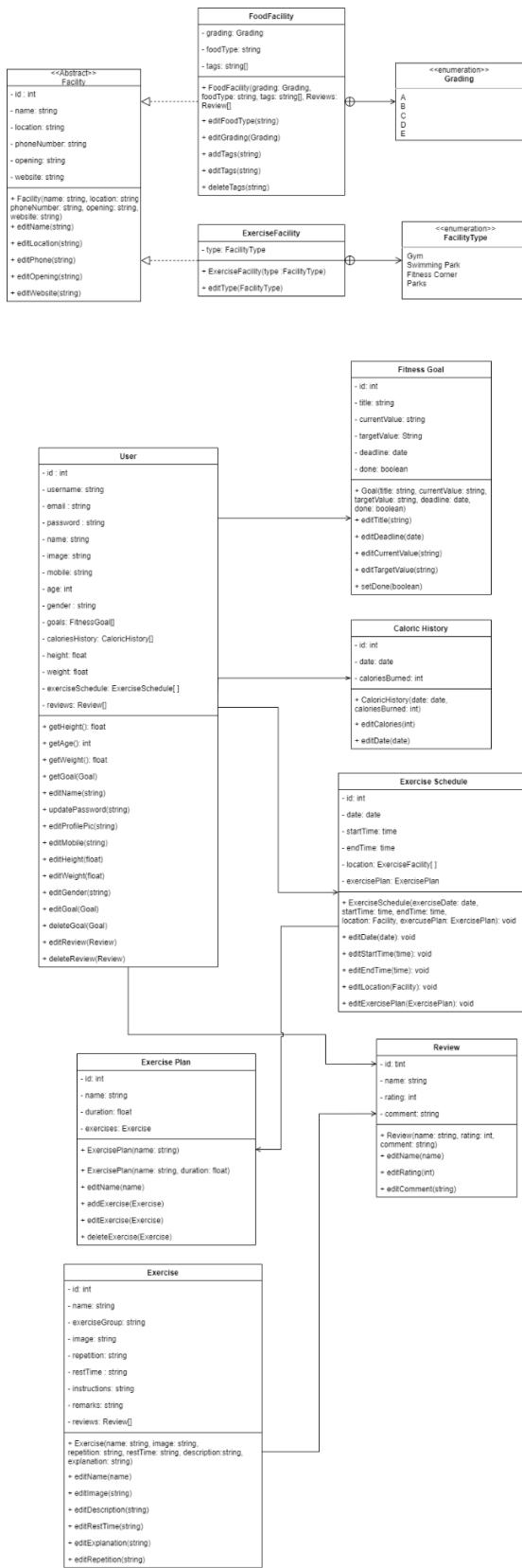


8.3.2 Observer and Controllers Class Diagram



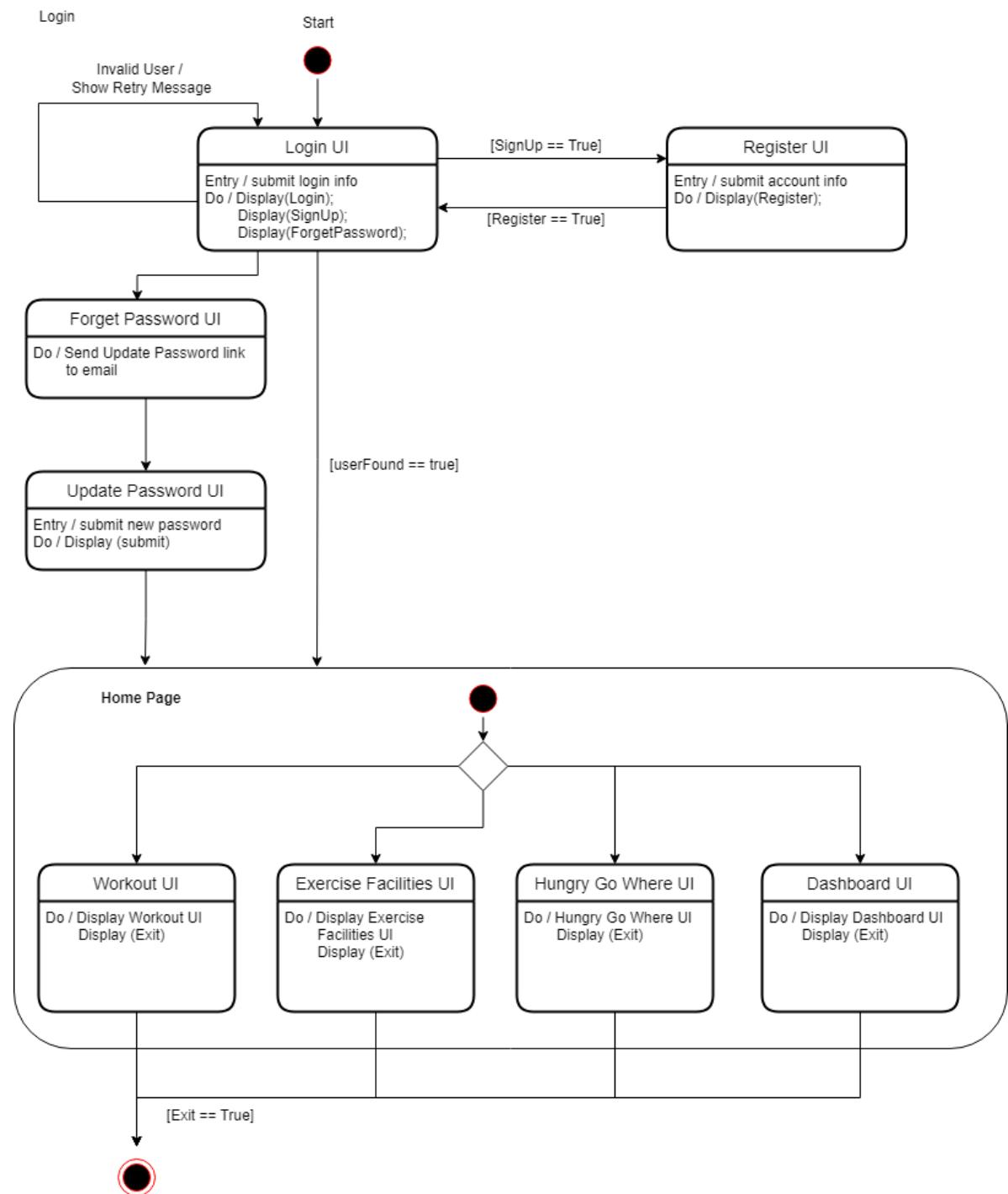
8.3.3 Models Class Diagram

Model

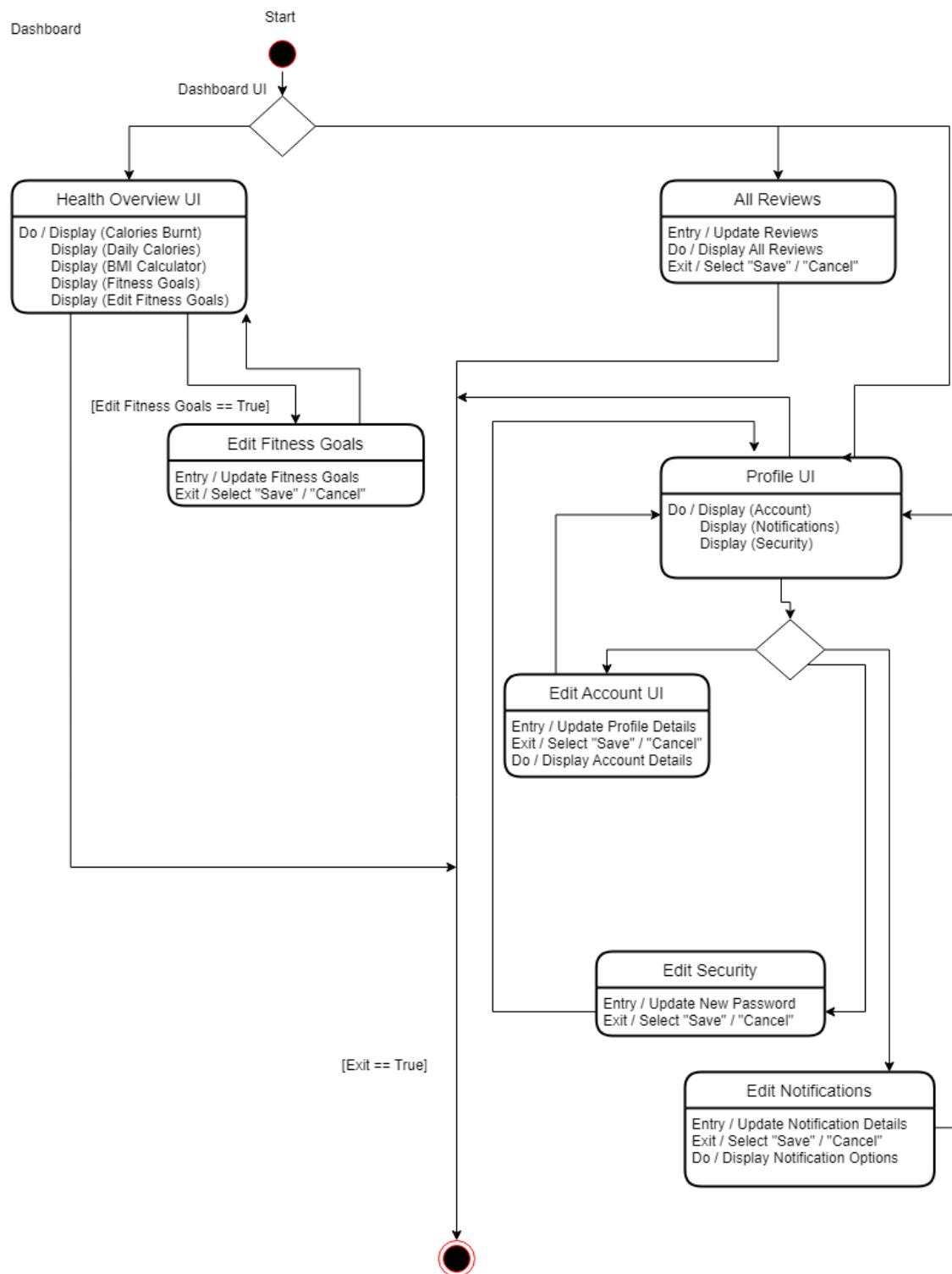


8.4 Dialog Map

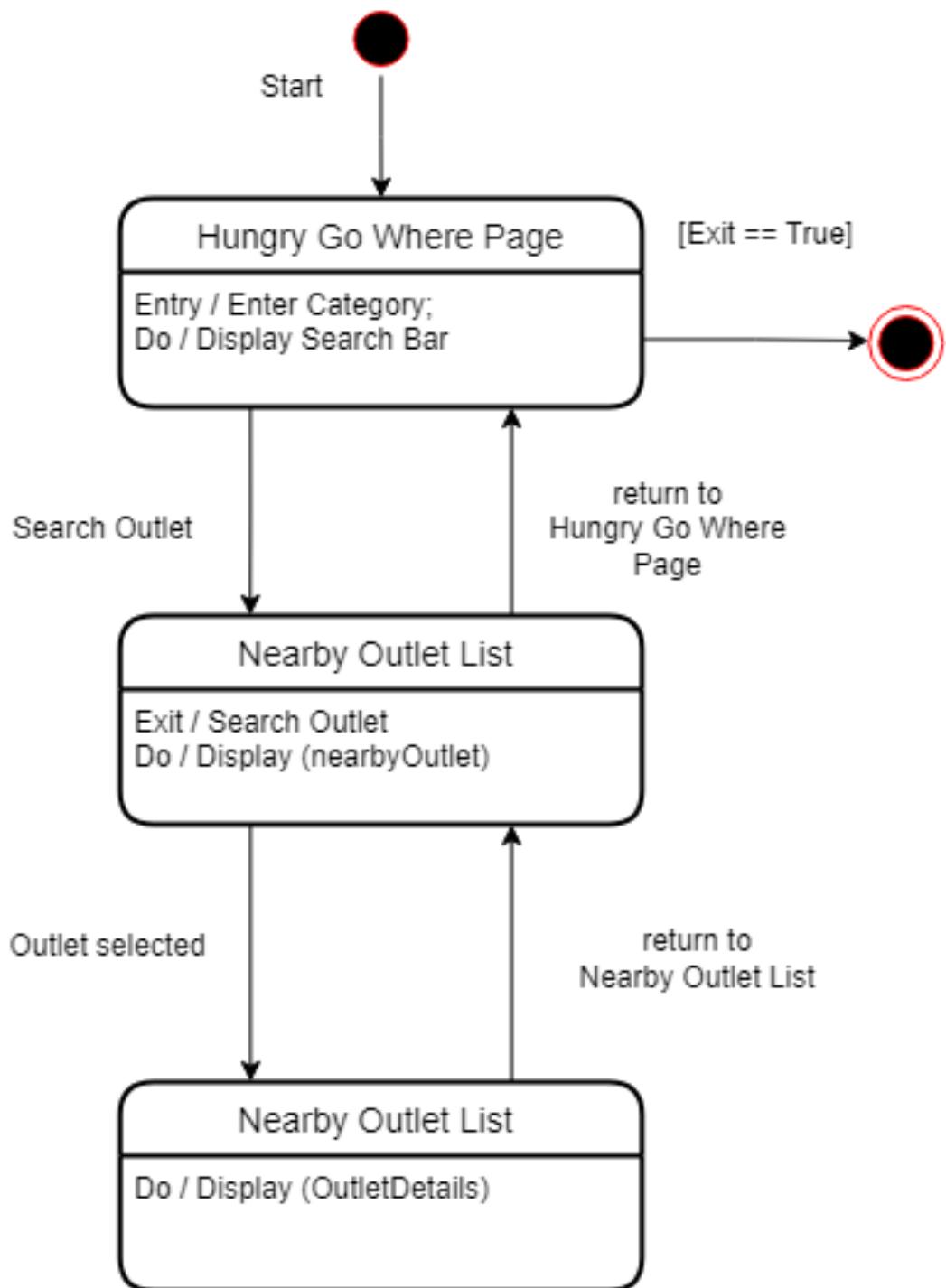
8.4.1 Login Page



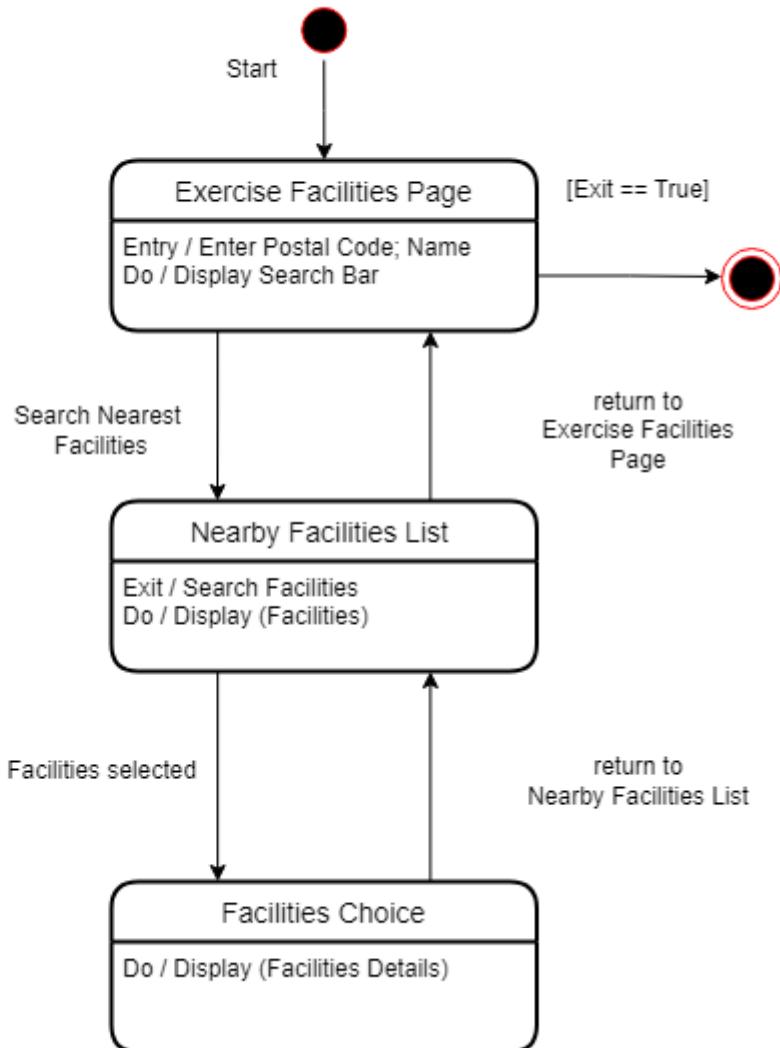
8.4.2 Dashboard Page



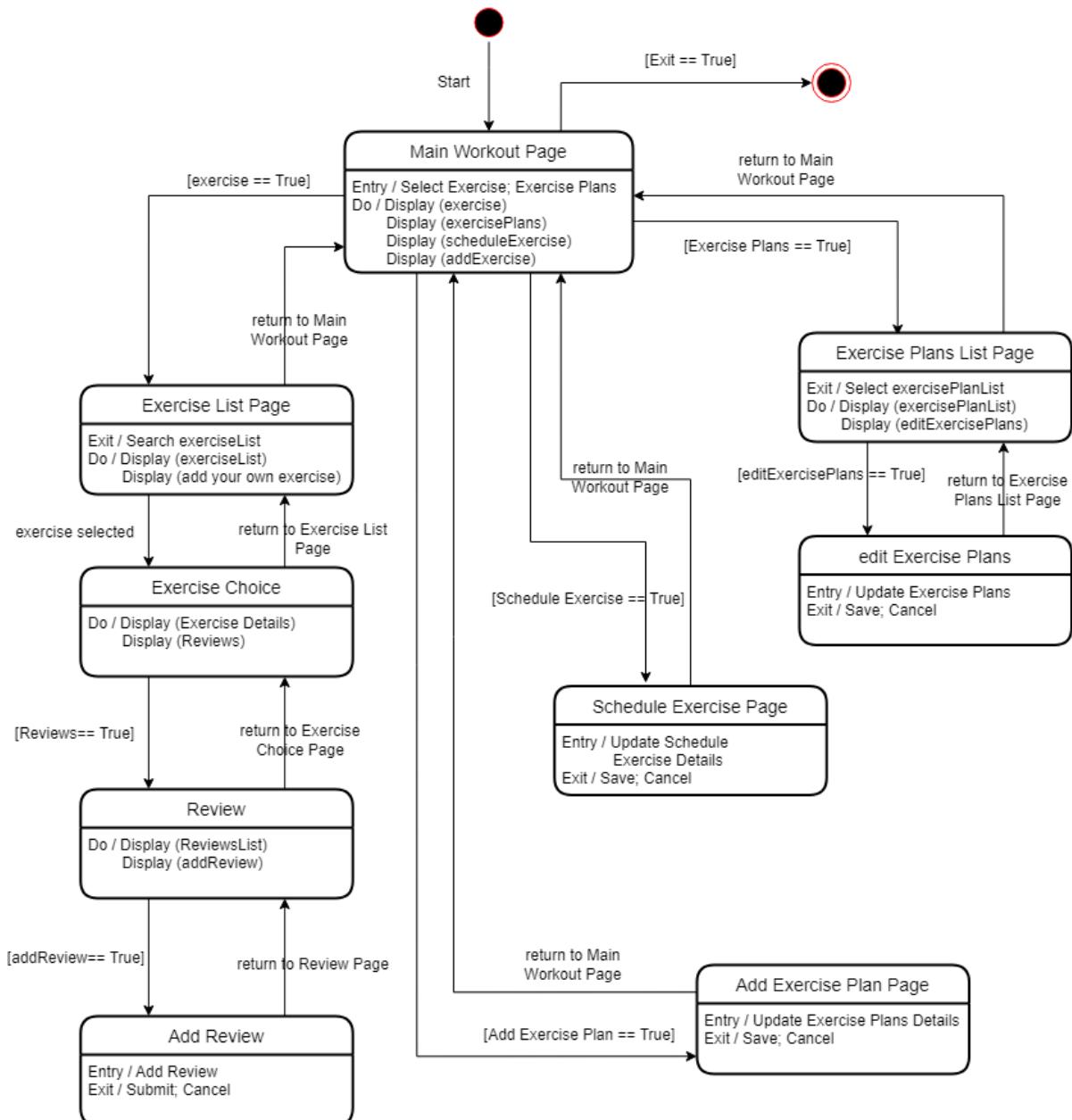
8.4.3 Hungry Go Where Page



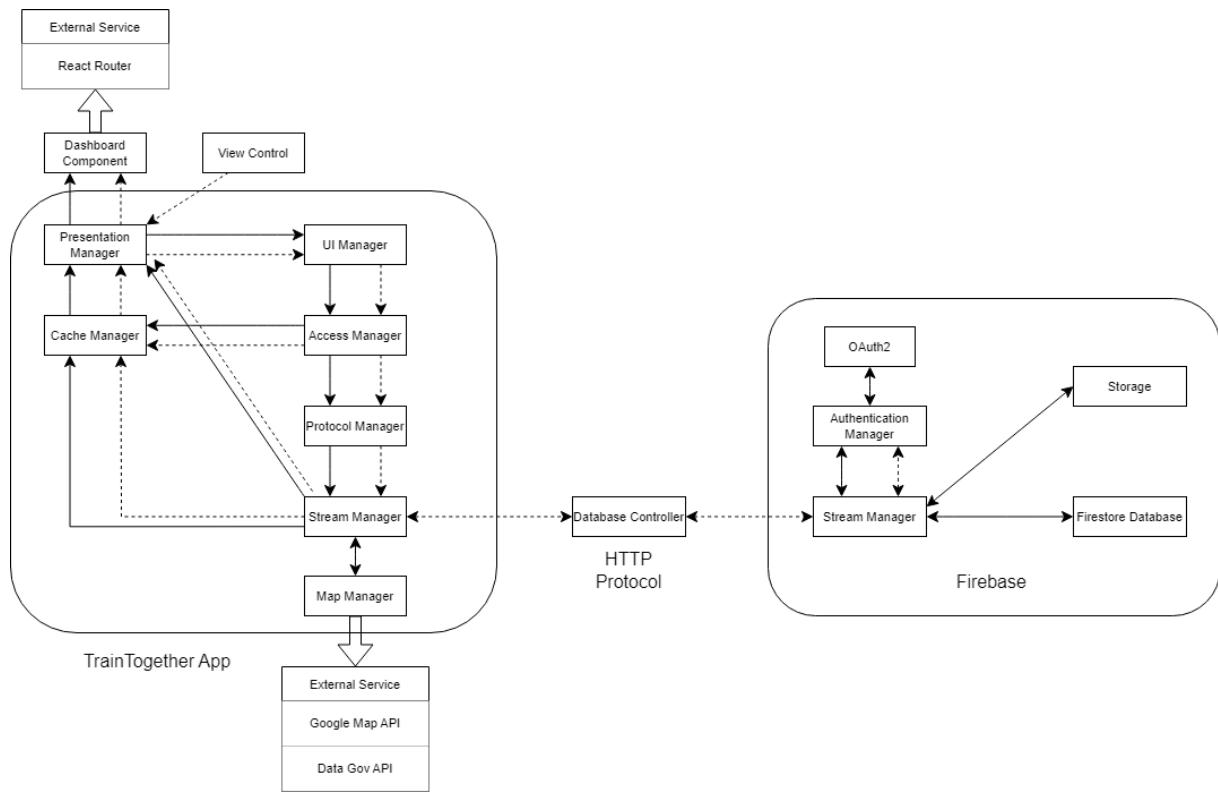
8.4.4 Exercise Facilities Page



8.4.5 Workout Page

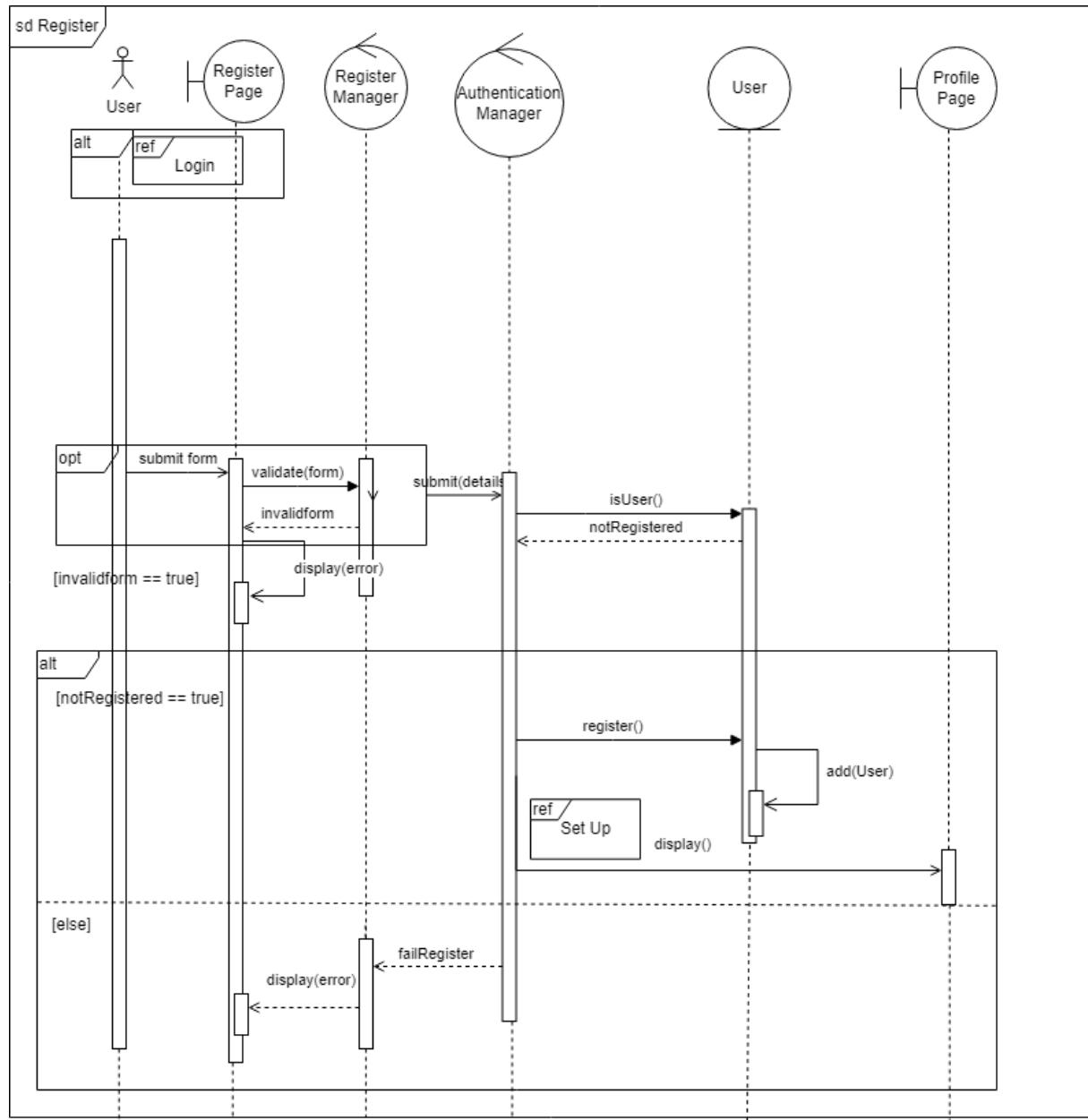


8.5 System Architecture

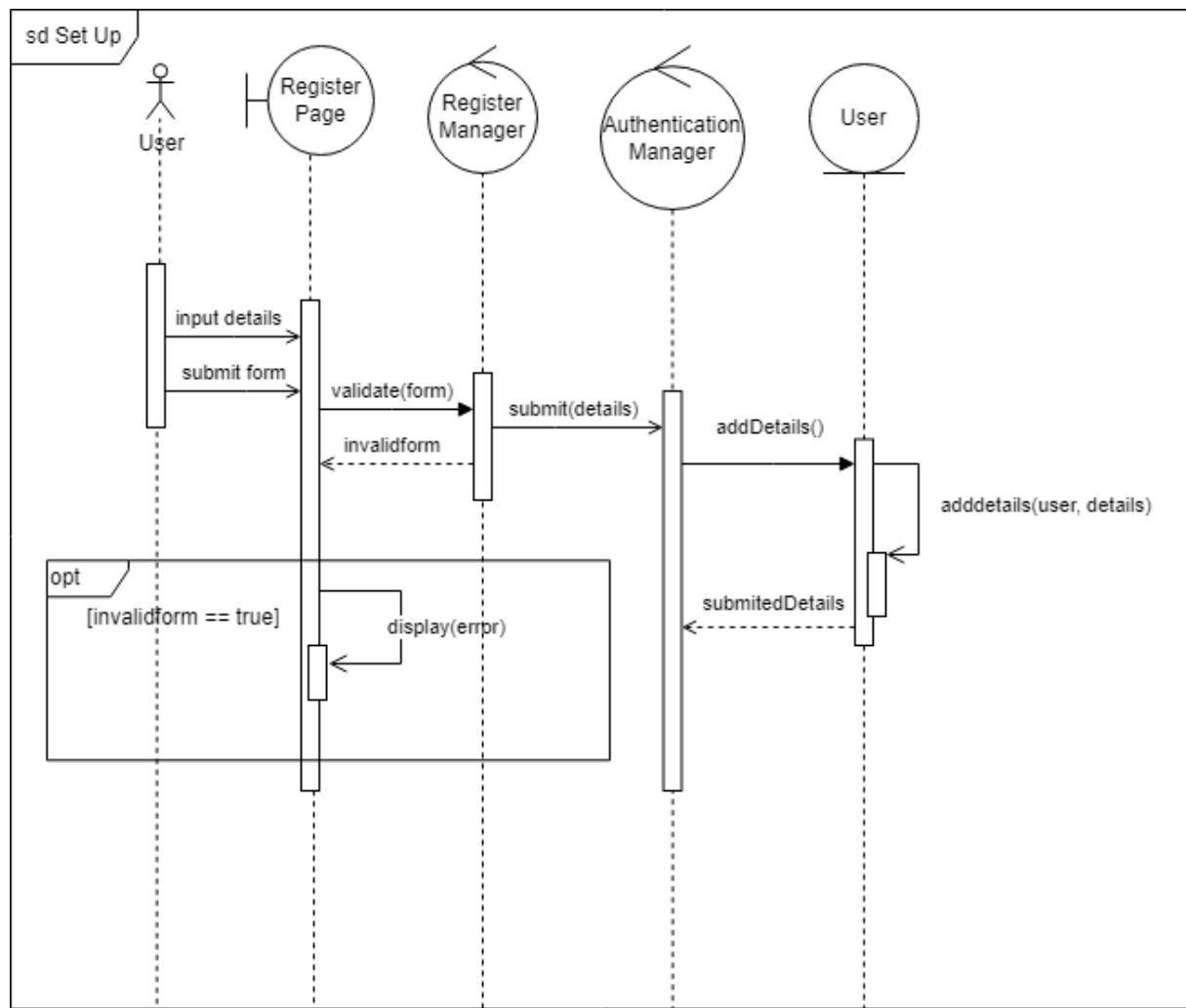


8.6 Sequence Diagram

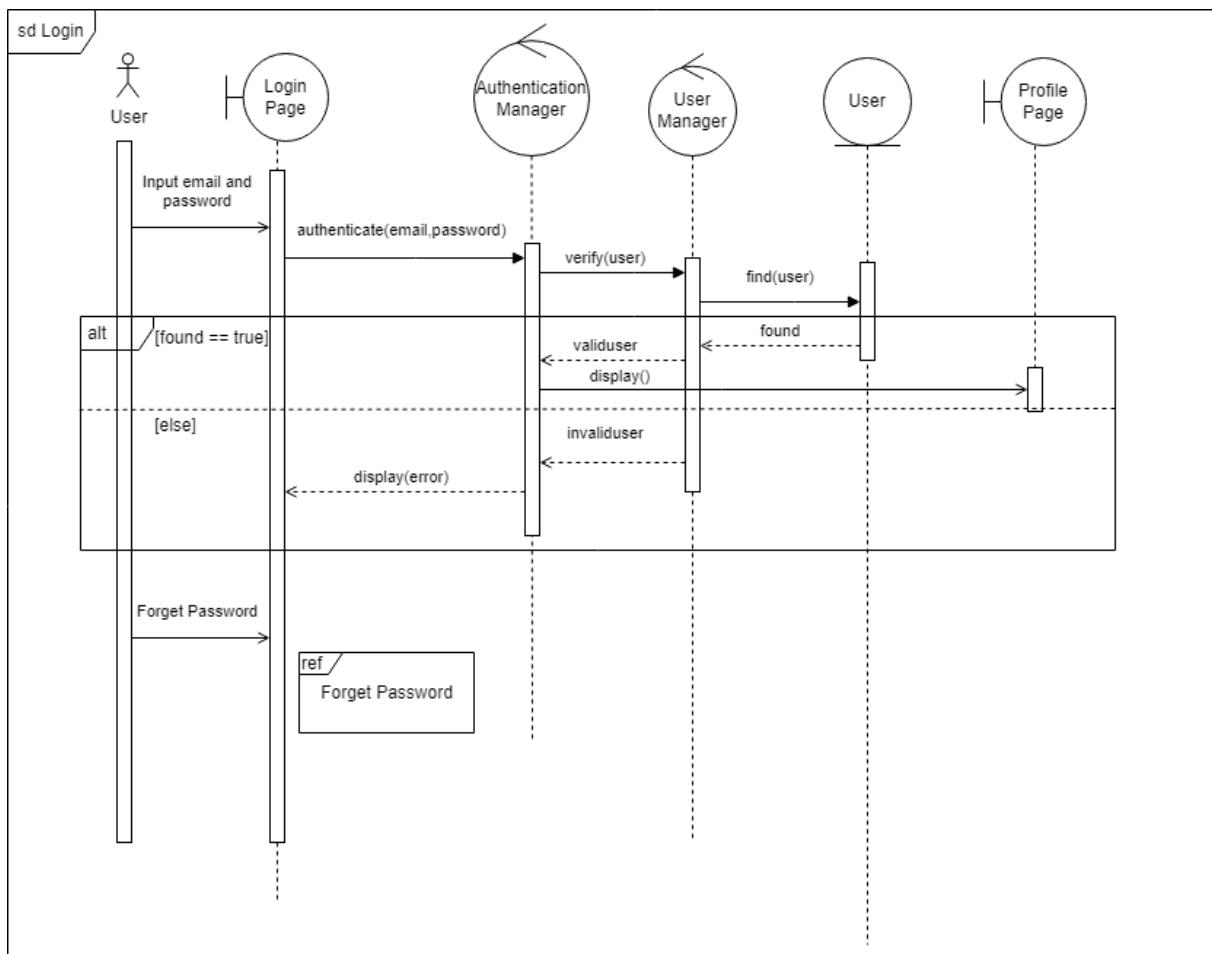
8.6.1 Register



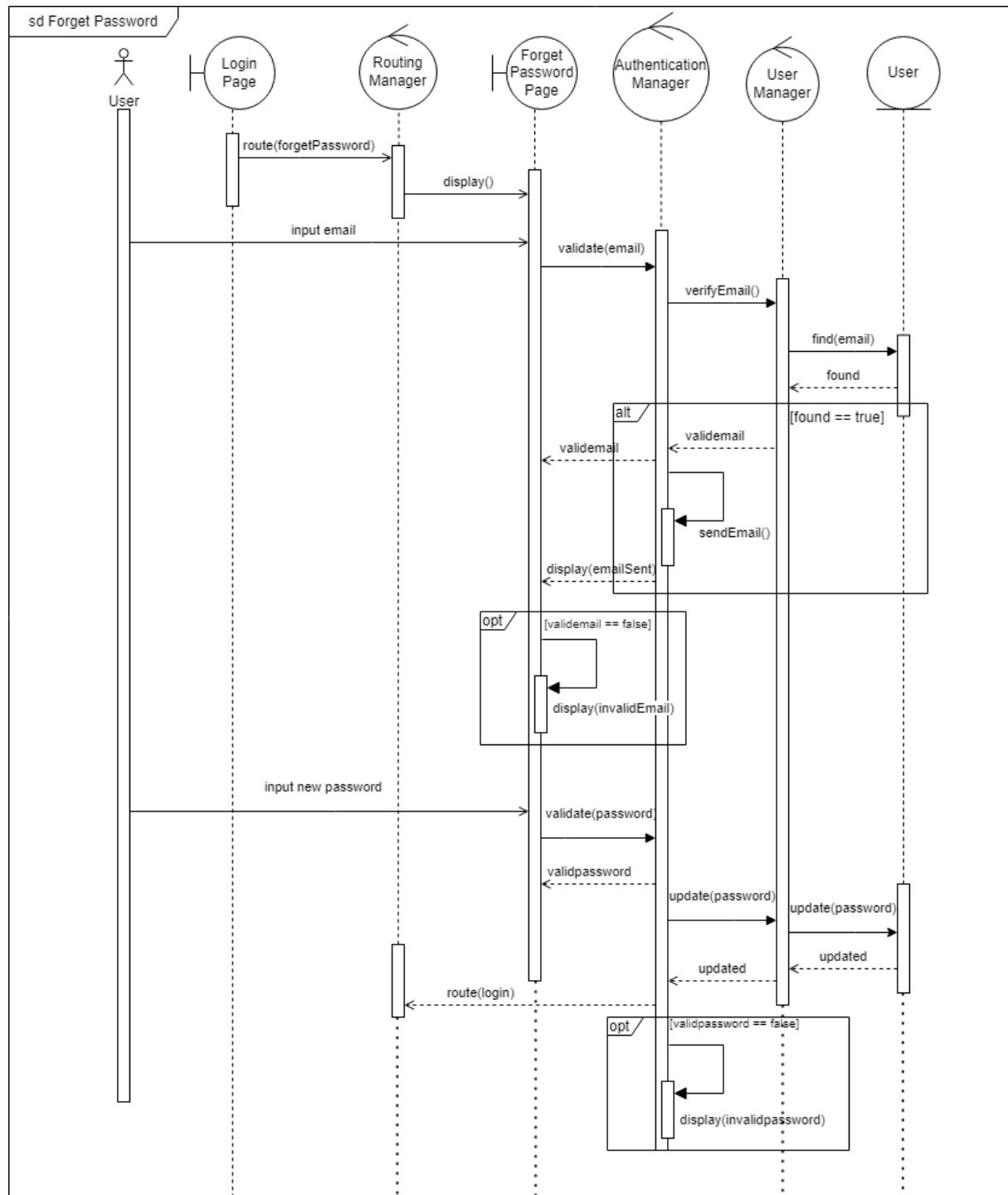
8.6.2 Set Up



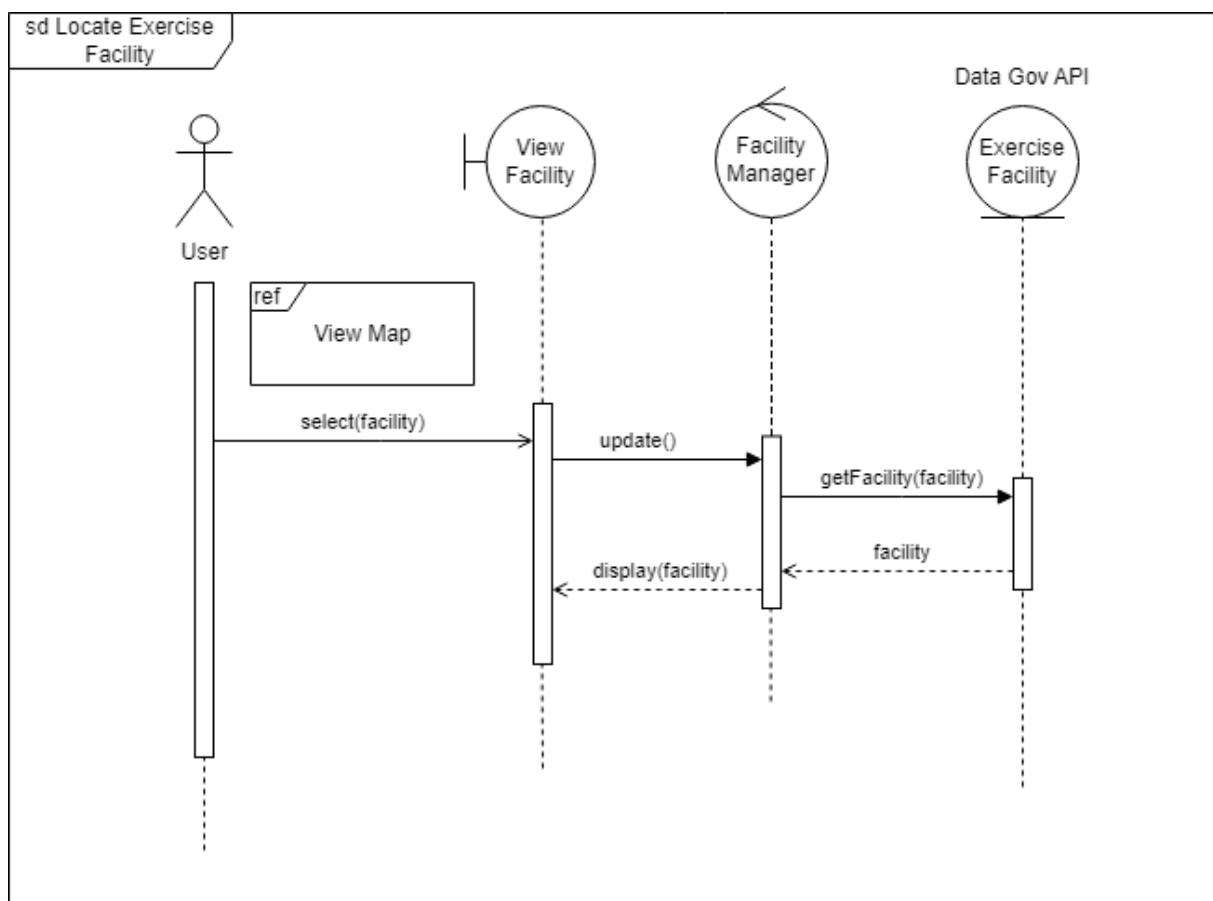
8.6.3 Login



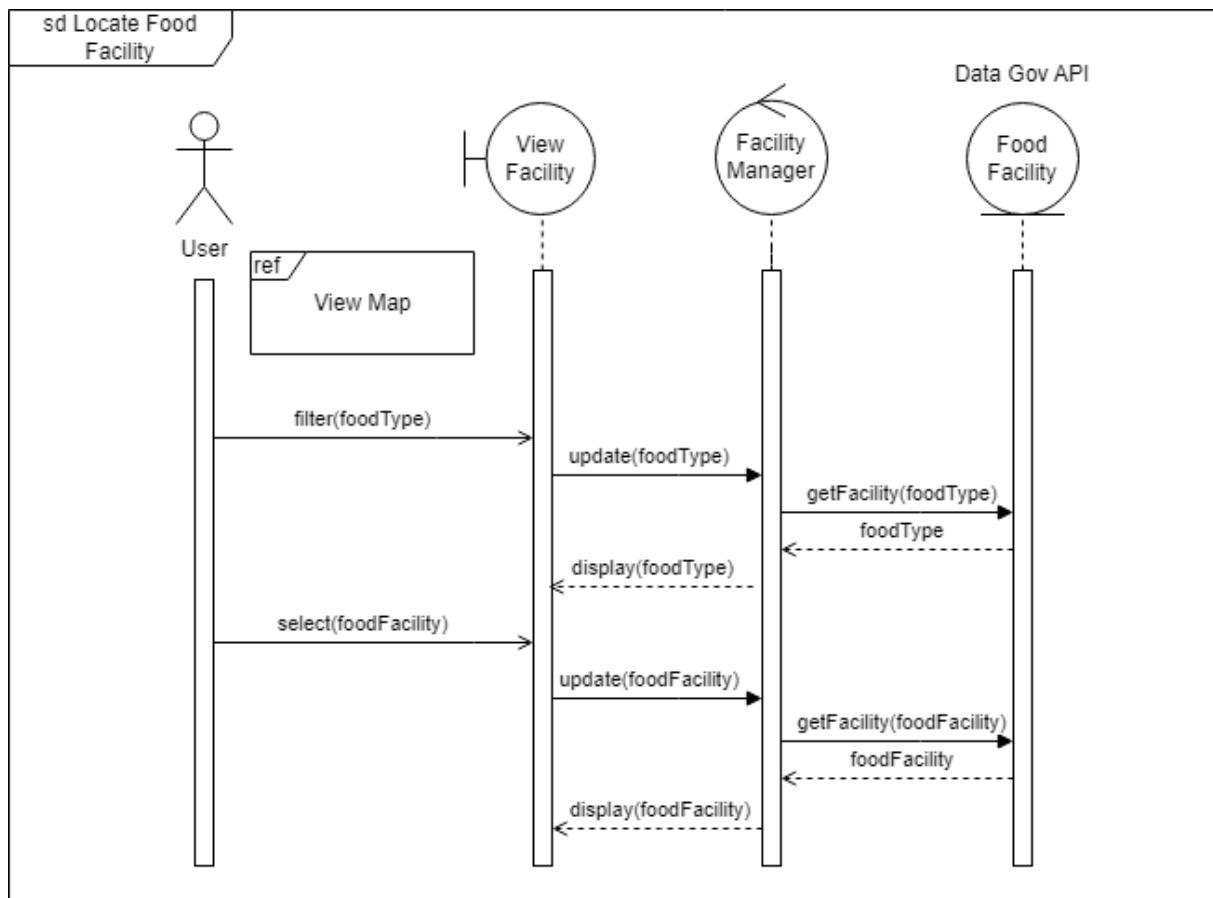
8.6.4 Forgot Password



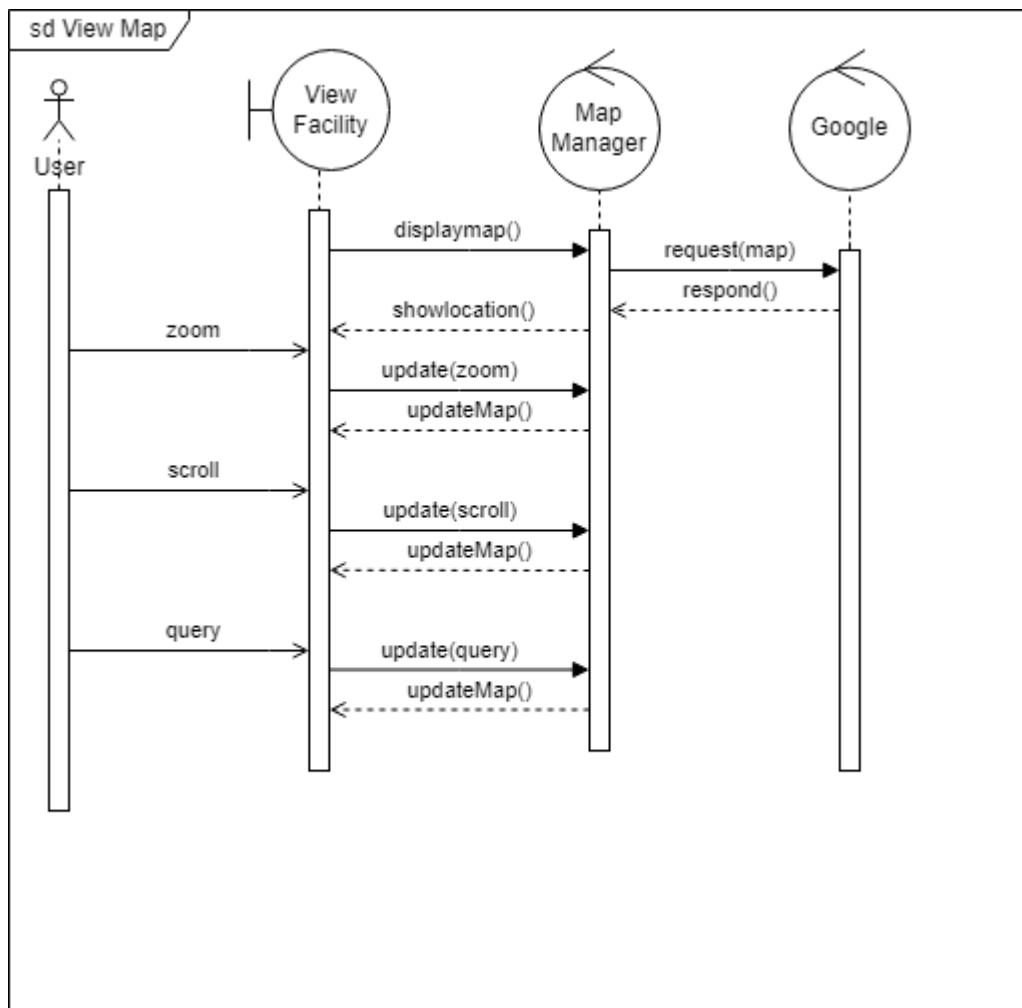
8.6.5 Locate Exercise Facilities



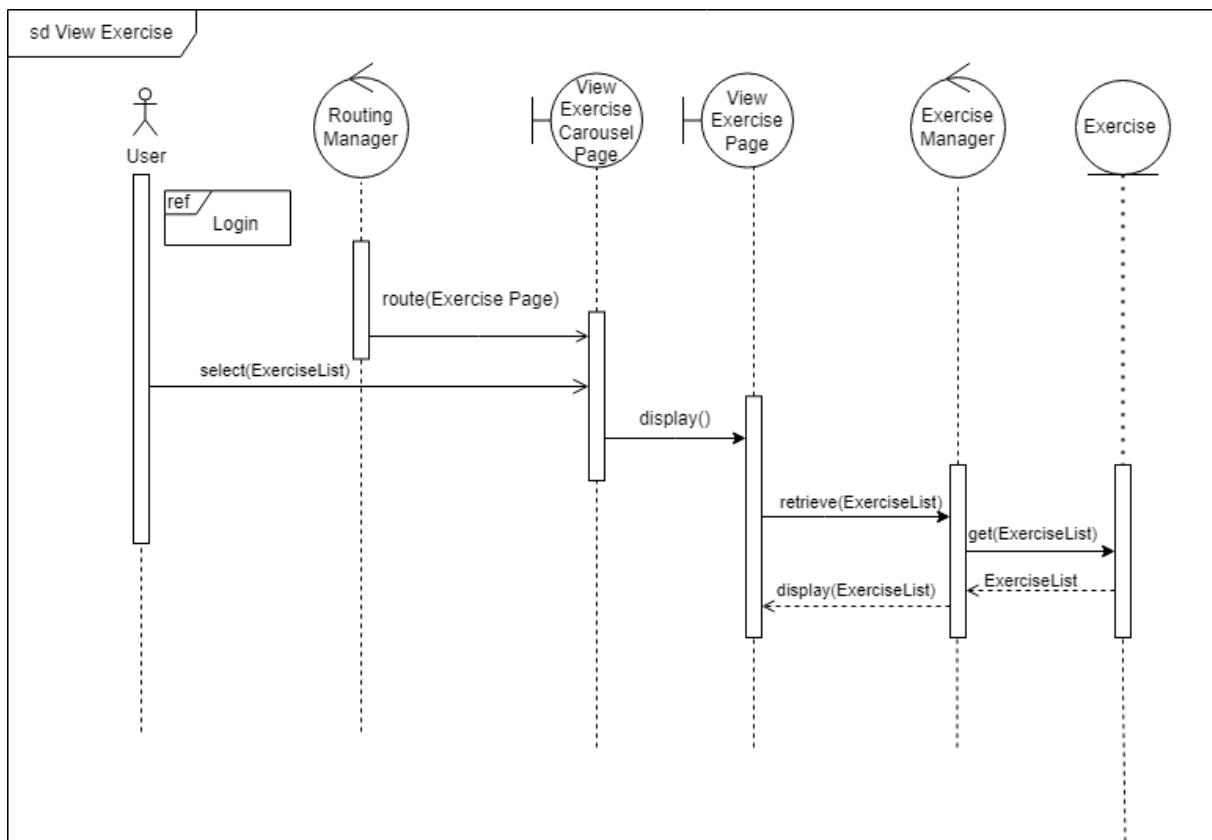
8.6.6 Locate Food Facilities



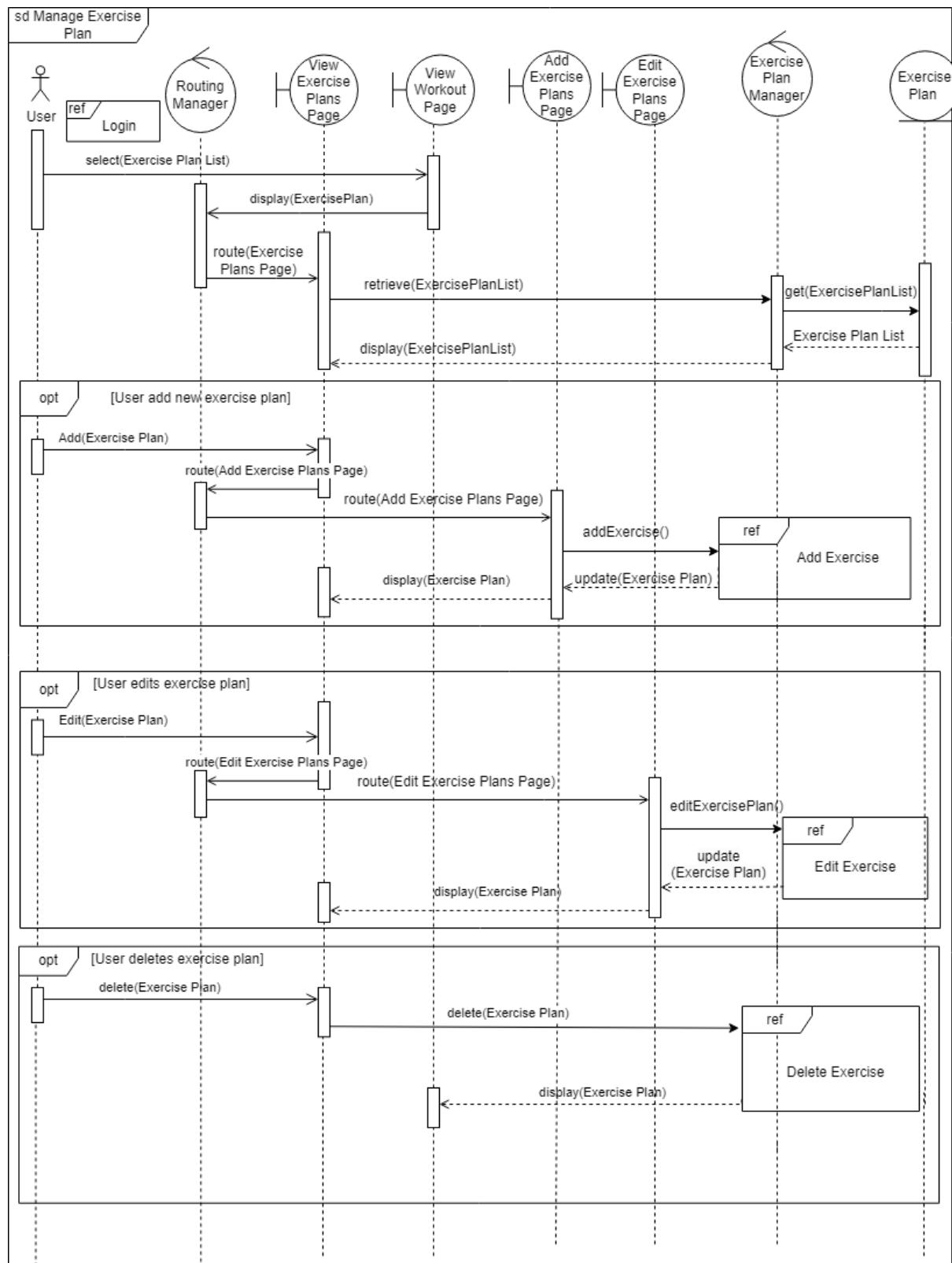
8.6.7 View Map



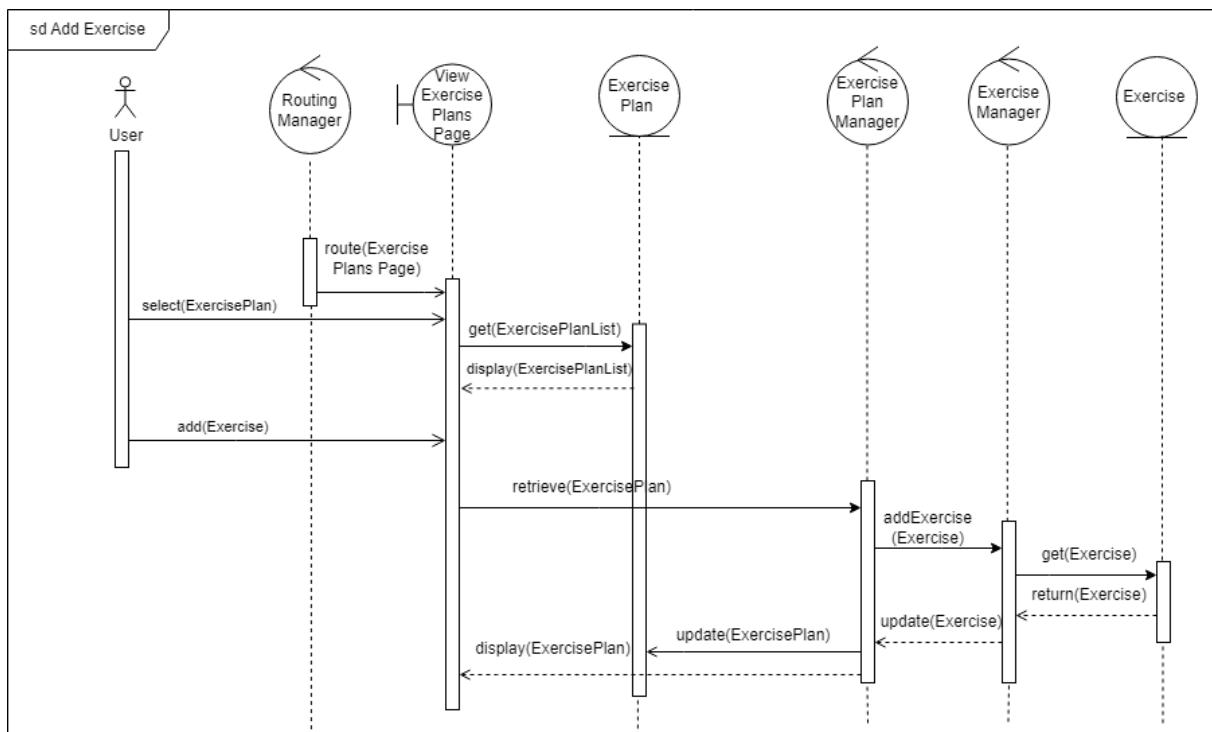
8.6.8 View Exercise



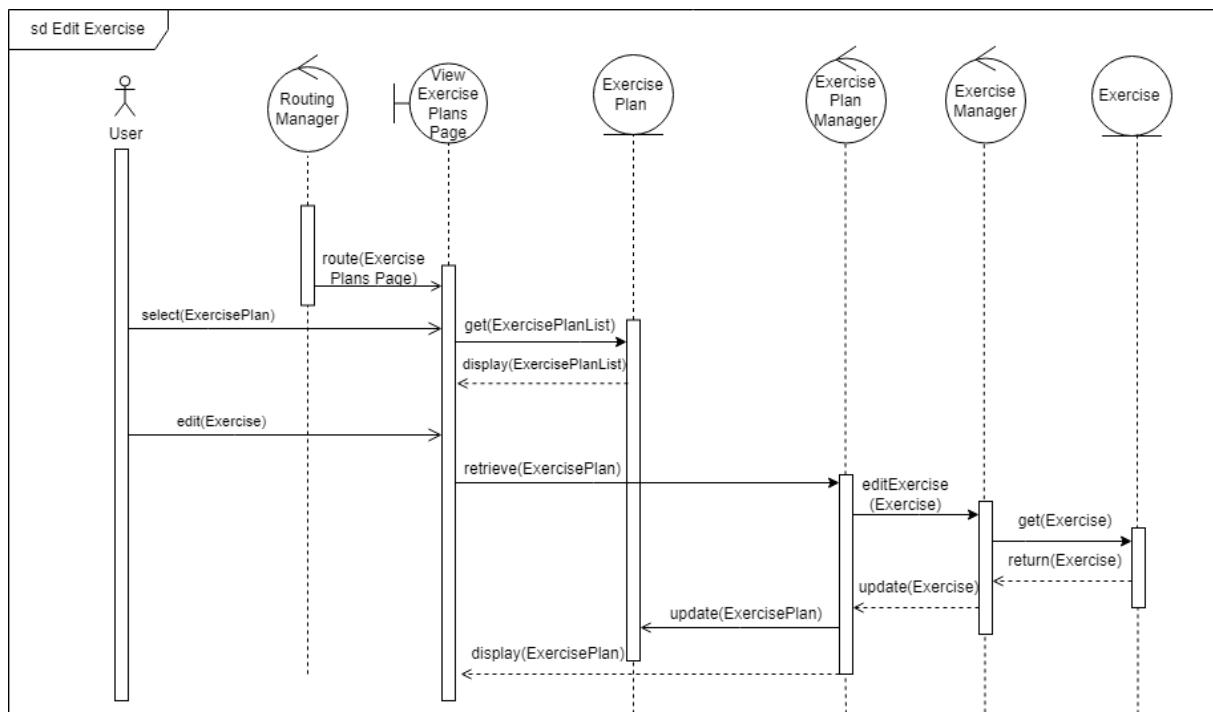
8.6.9 Manage Exercise Plan



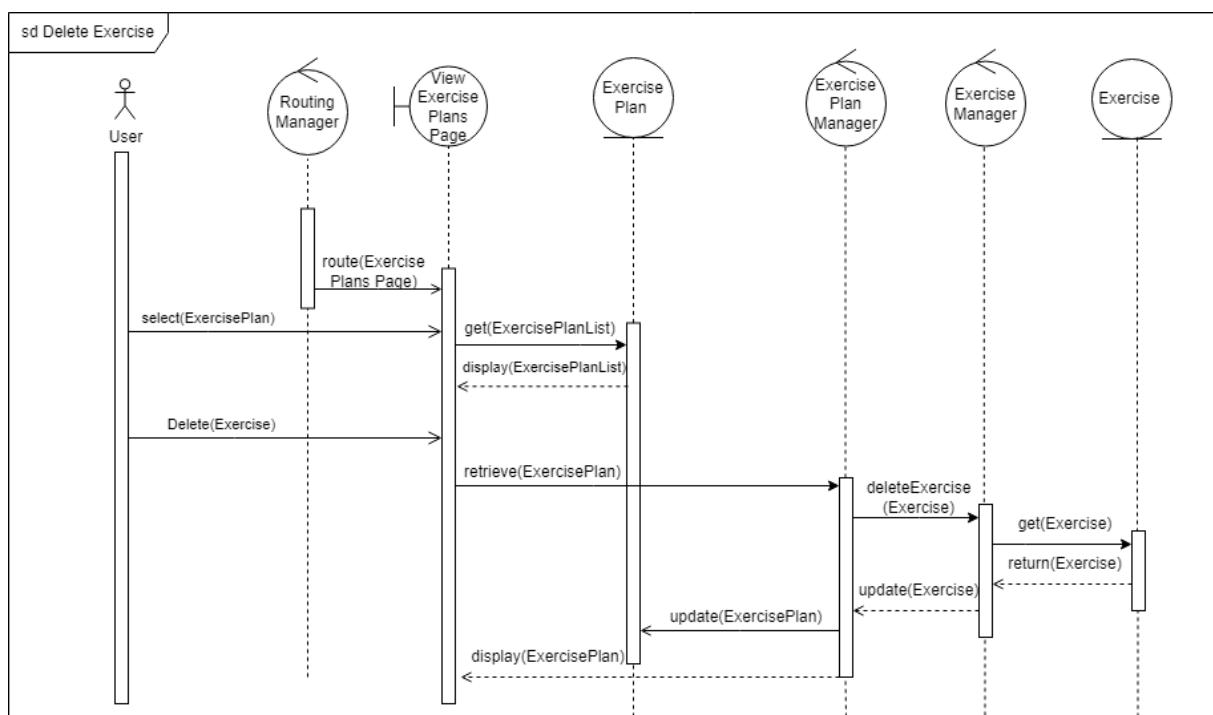
8.6.10 Add Exercise



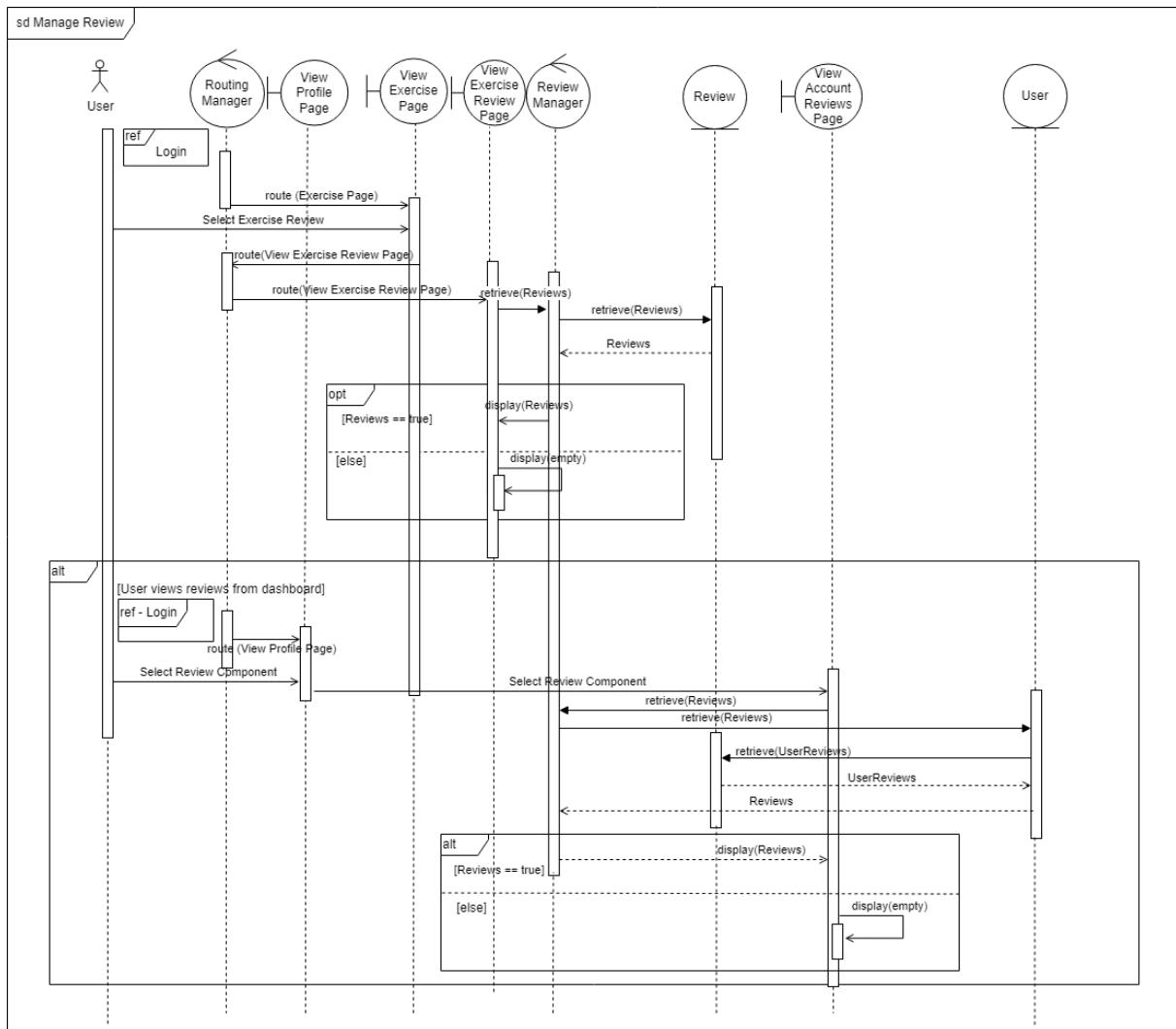
8.6.11 Edit Exercise



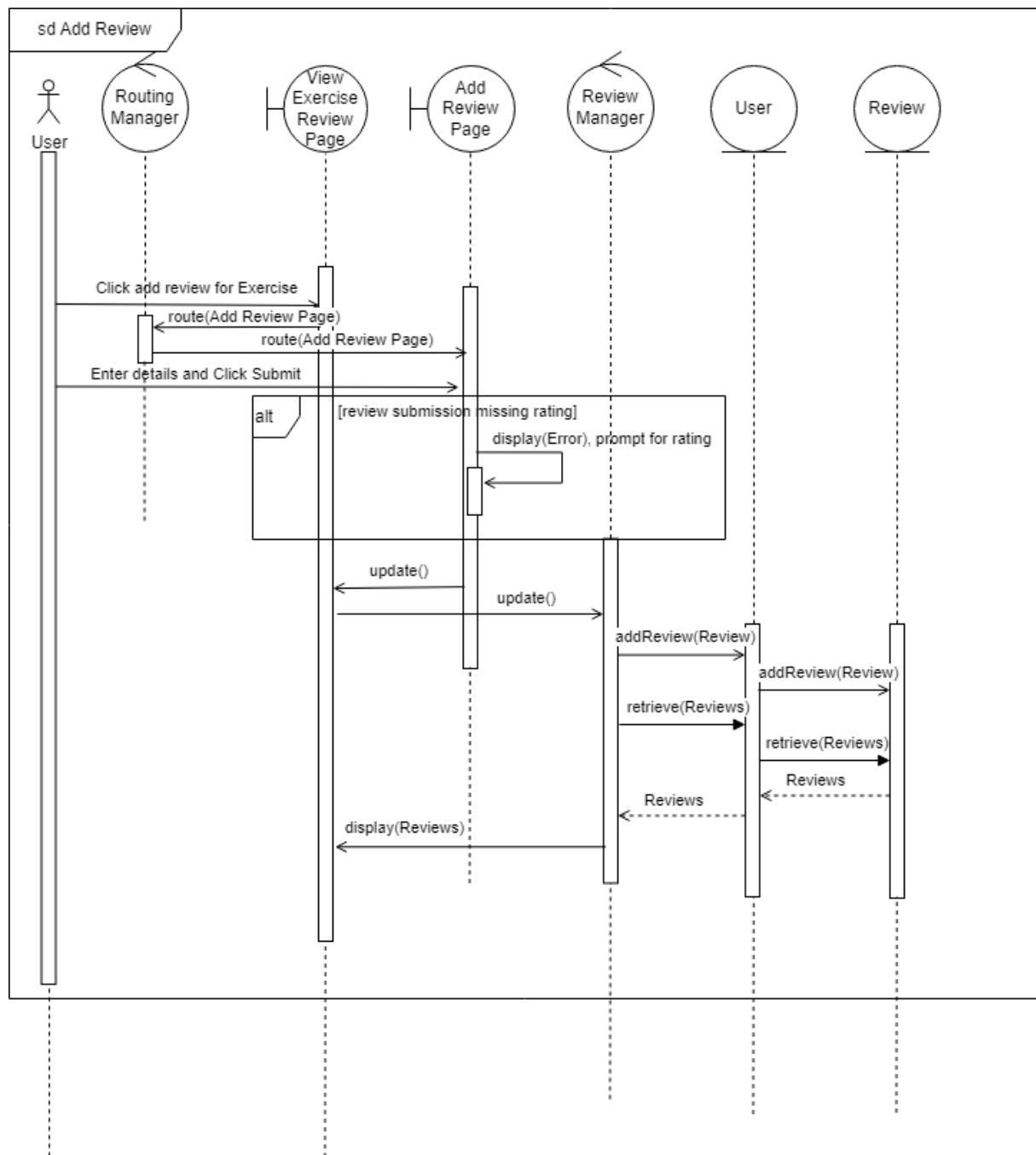
8.6.12 Delete Exercise



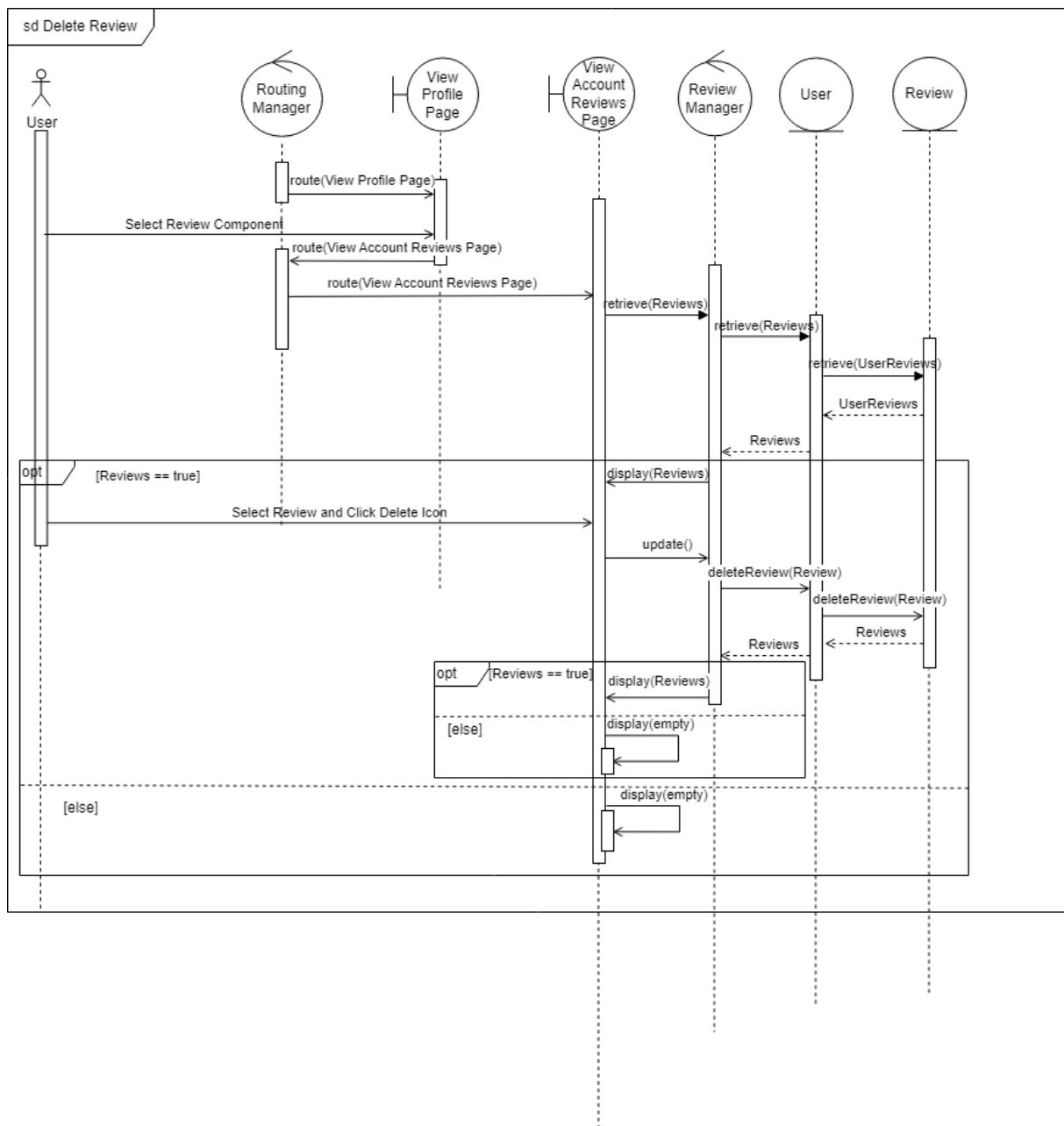
8.6.13 Manage Review



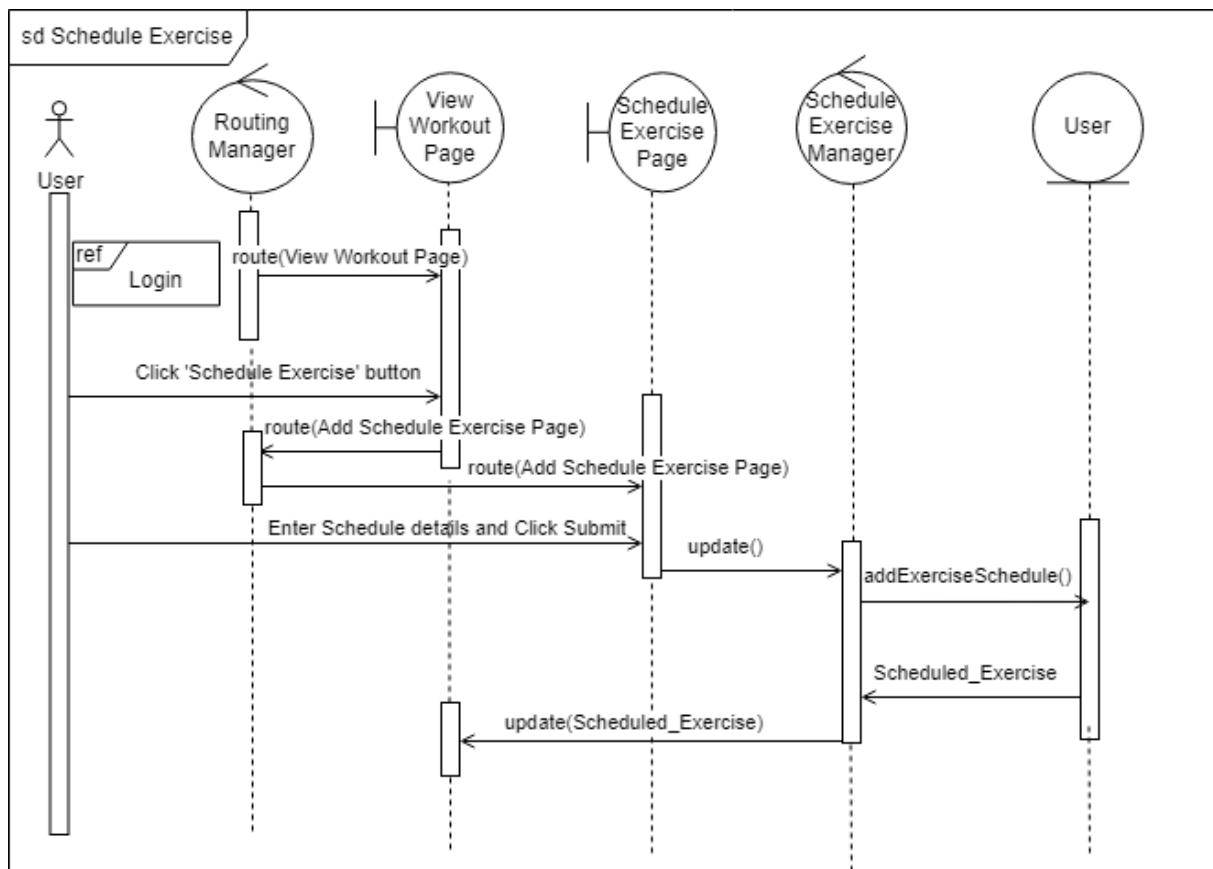
8.6.14 Add Review



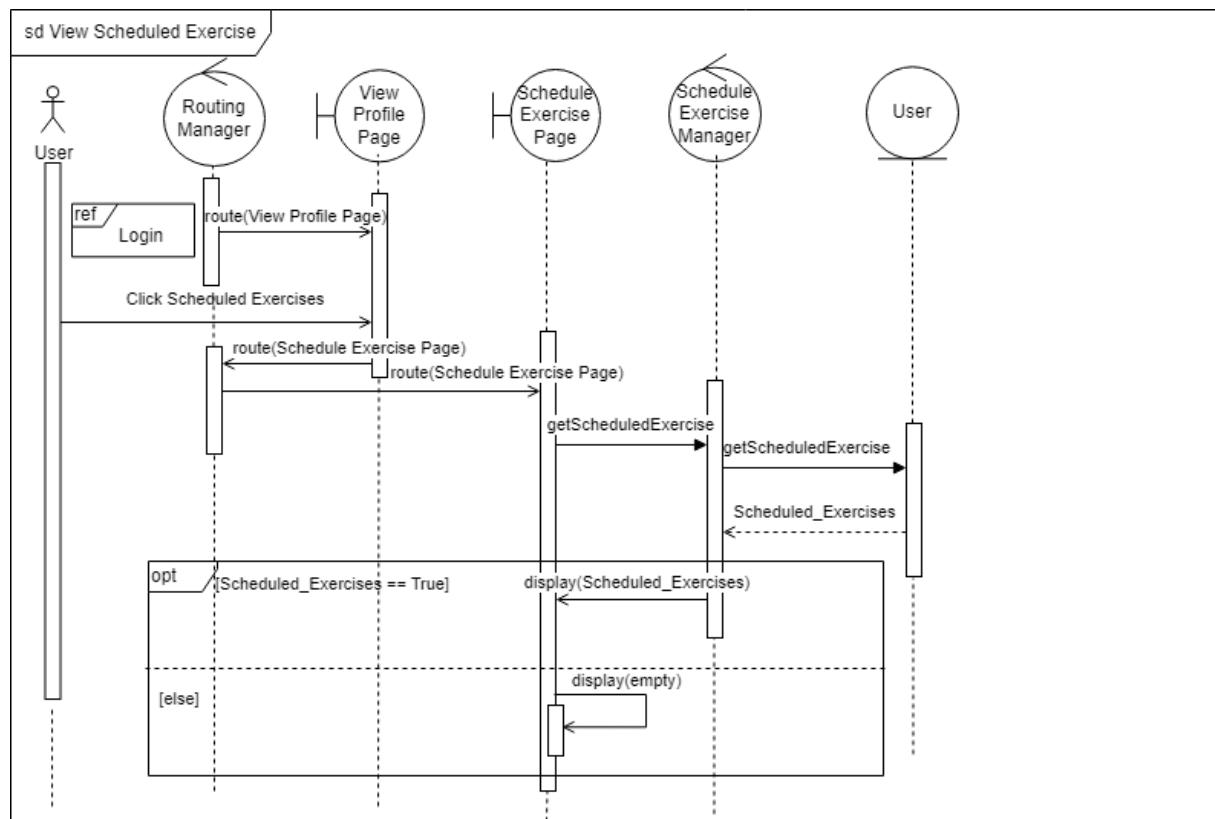
8.6.15 Delete Review



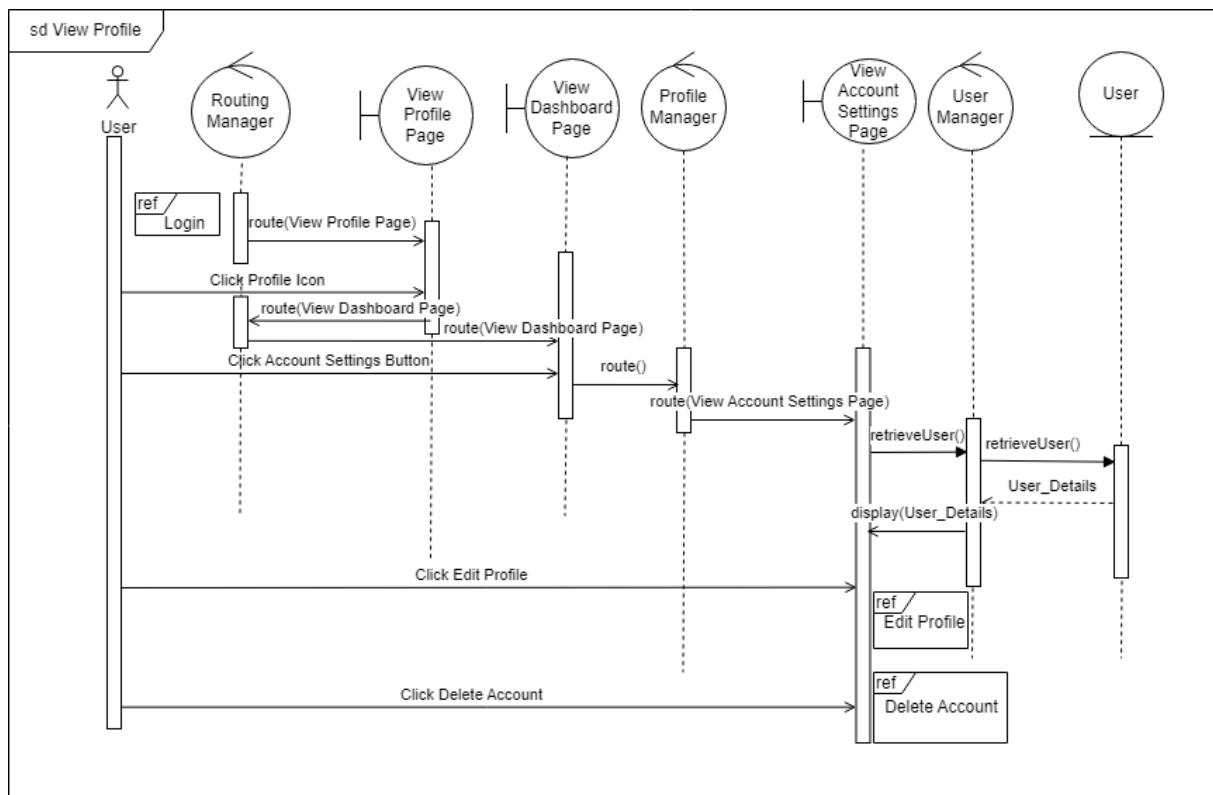
8.6.16 Schedule Exercise



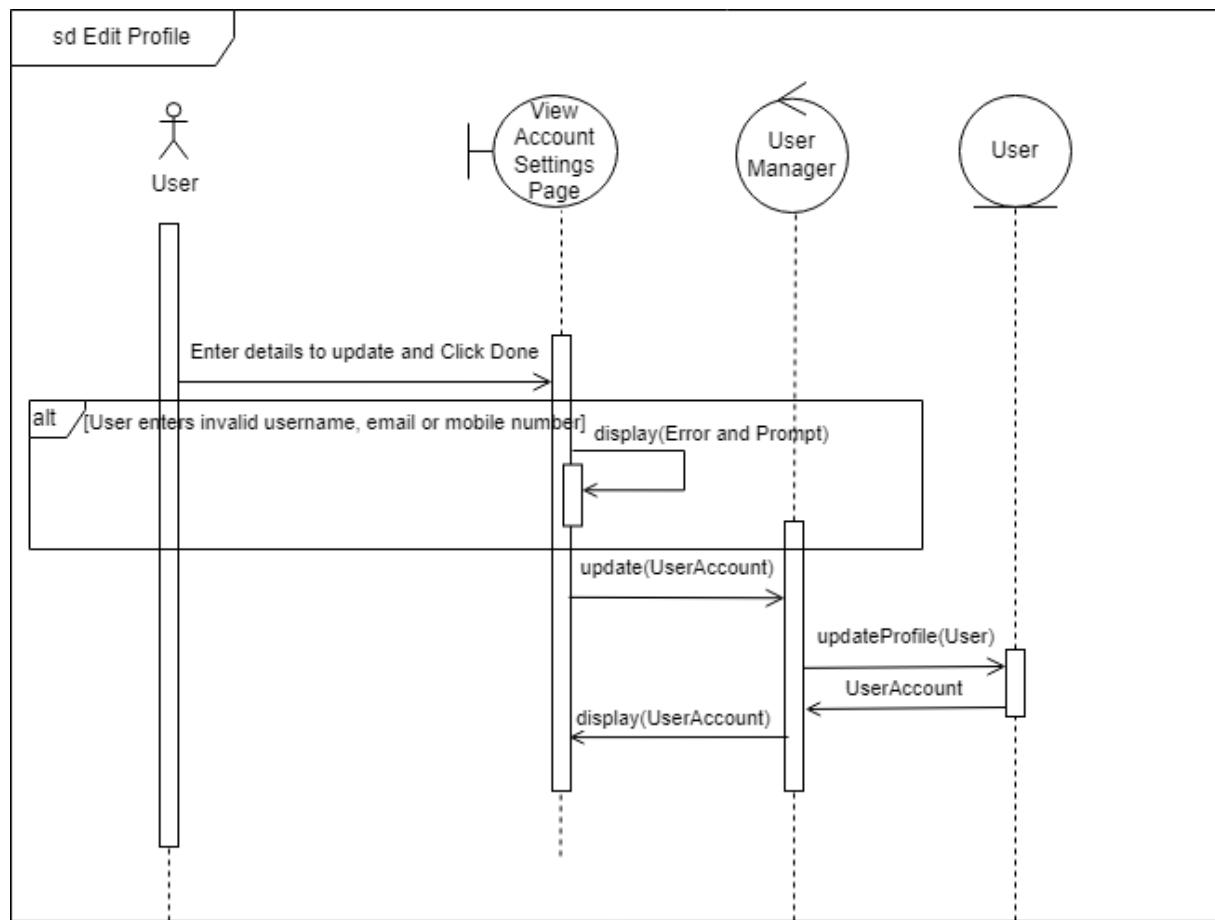
8.6.17 View Schedule Exercise



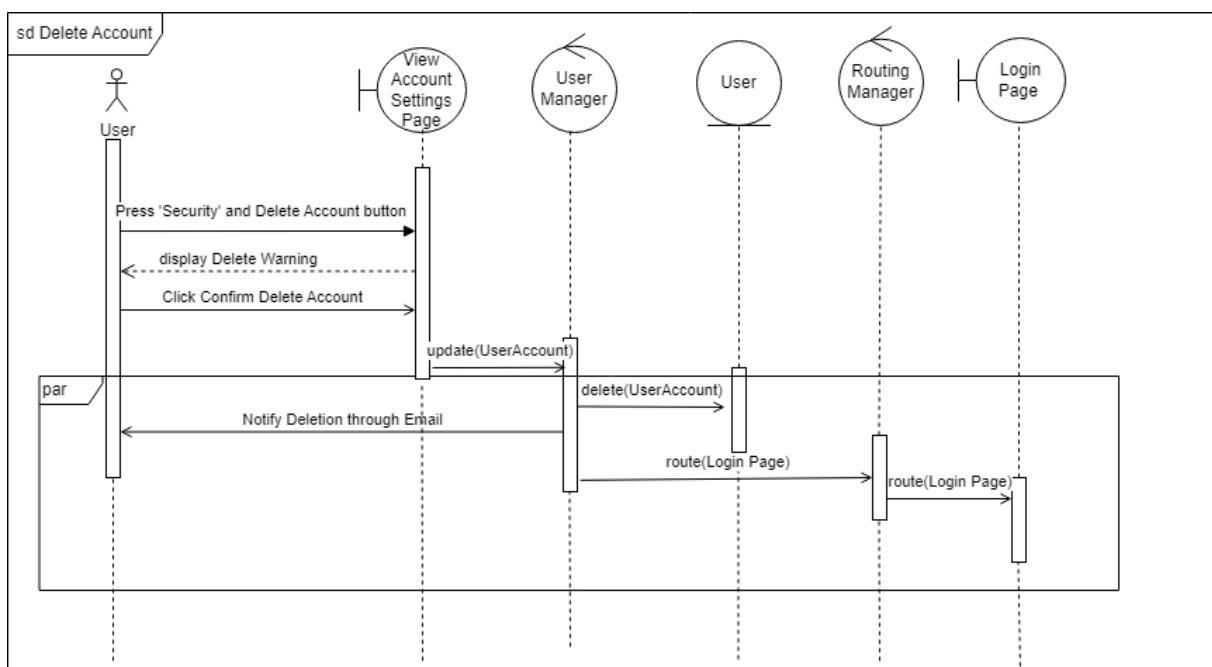
8.6.18 View Profile



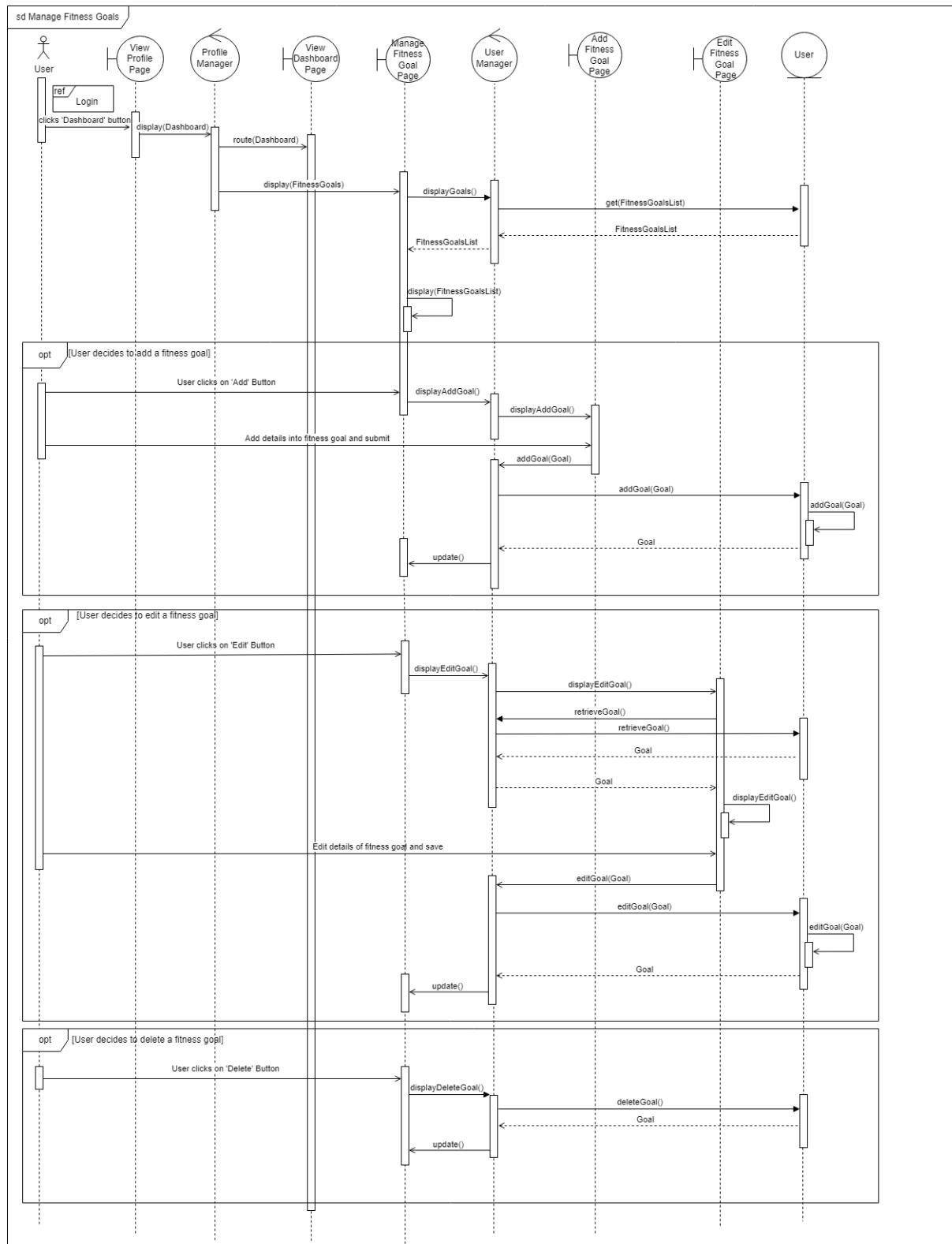
8.6.19 Edit Profile



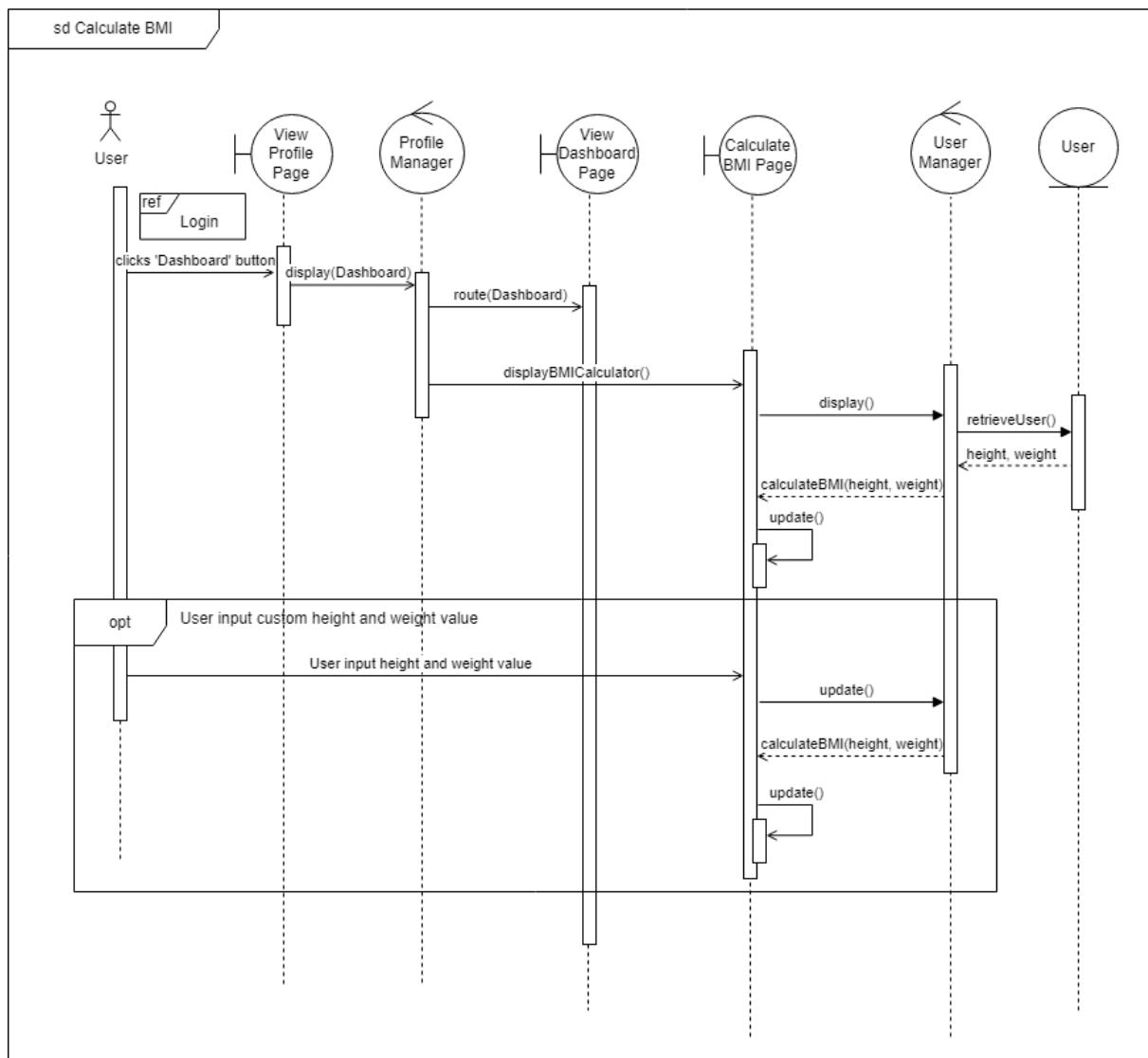
8.6.20 Delete Account



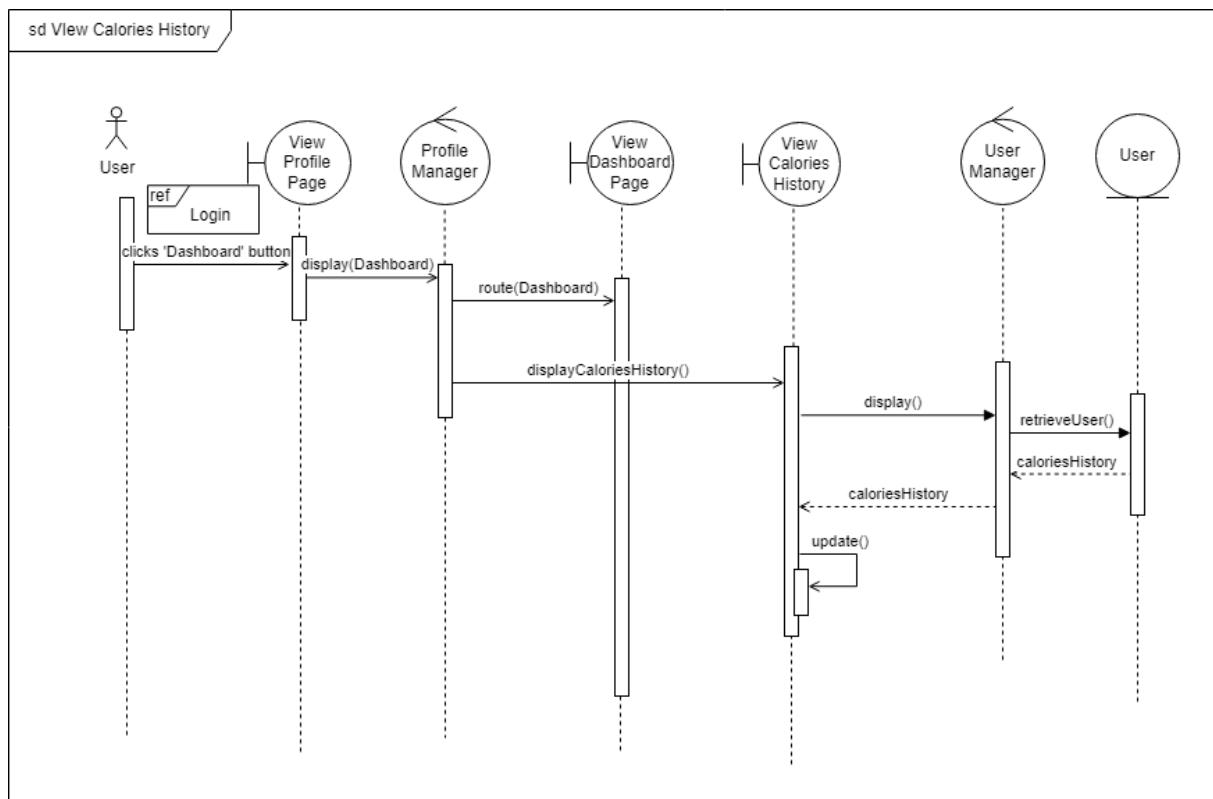
8.6.21 Manage Fitness Goals



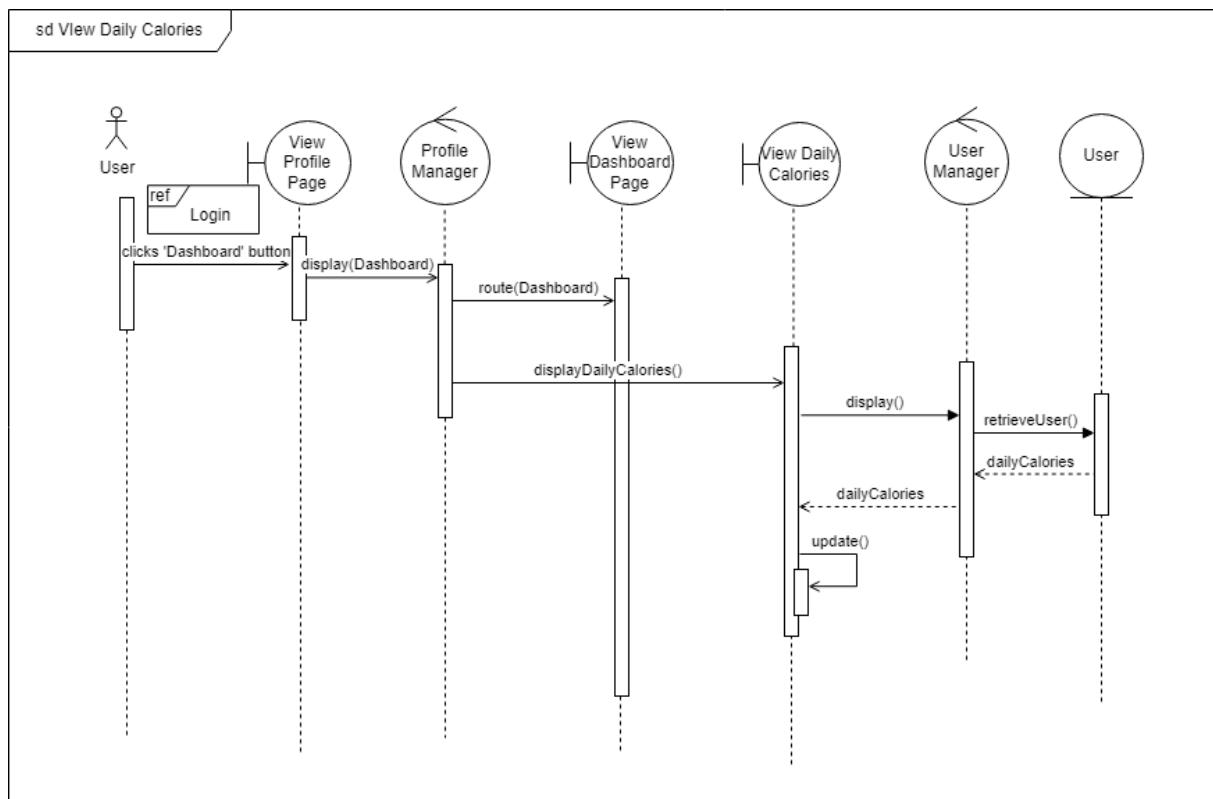
8.6.22 Calculate BMI



8.6.23 View Calories History



8.6.24 View Daily Calories



8.7 Unit Testing

8.7.1 Black Box Testing

For Black Box Testing we used both equivalence class testing and boundary value testing for the registering and setting up of users.

1. Register and Set Up User

Equivalence Classes and Boundary Values

Username

Specification	No spacing Minimum 6 characters long Must not be null
Valid Equivalence Classes	edmund562
Invalid Equivalence Classes	edmun (5 characters) edmund 562 (spacing)

Name

Specification	Must not be null
Valid Equivalence Classes	Ernest
Invalid Equivalence Classes	Null input

Email Address

Specification	Requires '@' in email Requires '.com' in email Cannot have been used Cannot be null
Valid Equivalence Classes	user@gmail.com
Invalid Equivalence Classes	usergmail.com (No @ symbol), user@gmai (No .com) user@gmail.com (used email)

Password

Specification	Minimum 8 characters
----------------------	----------------------

	One uppercase One special characters One lowercase One numeric character Must not be null
Valid Equivalence Classes	P@ssw0rd
Invalid Equivalence Classes	Null input P@ssw0r (7 characters) p@ssw0rd (No uppercase) Passw0rd (No special case) P@ssword (No numeric) P@SSW0RD (No lowercase)

Re-enter Password

Specification	Same input as password Must not be null
Valid Equivalence Classes	Same input as password
Invalid Equivalence Classes	Null input Different input from password

Date of Birth

Specification	Cannot be less than born 10 years ago (As of the current year) Cannot be more than 100 years ago Must not be null
Valid Equivalence Classes	01 Jan 1923 3 Apr 2013
Invalid Equivalence Classes	31 Dec 1899 4 April 2014 Null input

Height

Specification	No negative numbers Minimum of 90 cm Maximum of 220 cm
----------------------	--

	Must not be null
Valid Equivalence Classes	90, 220
Invalid Equivalence Classes	-20 89.9 220.1

Weight

Specification	No negative numbers Minimum of 20 kg Maximum of 400 kg Must not be null
Valid Equivalence Classes	20 400
Invalid Equivalence Classes	-1 19.9 400.9

Gender

Specification	Form control - only 3 acceptable values Male, Female, Others
Valid Equivalence Classes	Male Female Others
Invalid Equivalence Classes	-

Phone Number

Specification	Phone number must be 8 digits Number must begin with 6, 8 or 9 Must not be null
Valid Equivalence Classes	61234567 80001234 91234000

Invalid Equivalence Classes	9234 78911234 Null input
------------------------------------	--------------------------------

Profile Picture

Specification	File must be a picture format Allow for empty profile page which will use the default picture
Valid Equivalence Classes	Empty file .jpg file .png file .gif file
Invalid Equivalence Classes	.txt file

Username	Name	Email Address	Password	Re-enter Password	Date of Birth	Height	Weight	Gender	Phone Number	Profile Picture	Expected Result	Test Result
edmund562	Ernest	user@gmail.com	P@ssw0rd	P@ssw0rd	3 April 2013	90	20	Male	80001234	empty file	Account created	Account created
edmun	Ernest	edmund@gmail.com	P@ssw0rd	Password@123	4 Sep 1968	100	20.1	Female	80001234	jpg file	Username is too short - a minimum of 6 characters.	Username is too short - a minimum of 6 characters.
edmund 562	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	14 Apr 2000	110	21	Others	80001234	png file	Username must not contain spaces	Username must not contain spaces
Null input	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	10 Feb 2001	120	200	Male	61234567	empty file	No username provided!	No username provided!
ernestiscute	Null input	edmund@gmail.com	P@ssw0rd	P@ssw0rd	19 Sep 1989	130	100	Female	61234567	jpg file	No name provided!	No name provided!
solo588	Ernest	usergmail.com	P@ssw0rd	P@ssw0rd	21 Jan 2002	140	50.5	Others	61234567	png file	Invalid email address	Invalid email address
blackknight	Ernest	user@gmail	P@ssw0rd	P@ssw0rd	1 Dec 1938	150	60	Male	61234567	empty file	Invalid email address	Invalid email address
yellowriver	Ernest	Null input	P@ssw0rd	P@ssw0rd	31 Jan 1948	160	70	Female	61234567	jpg file	No password provided!	No password provided!
yellowriver	Ernest	user@gmail.com	P@ssw0rd	P@ssw0rd	31 Jan 1948	160	70	Female	61234567	jpg file	Email has been used. Please register with another email or login with existing account.	Email has been used. Please register with another email or login with existing account.
yell0wpanther	Ernest	edmund@gmai.c om	Null input	P@ssw0rd	10 Aug 1900	170	80	Others	61234567	png file	No password provided!	No password provided!

Software Requirements Specification for TrainTogether v5.3

cptUSA	Ernest	edmund@gmail.com	P@ssw0r	P@ssw0r	8 Aug 1950	180	90.6	Male	61234567	empty file	Password is too short - a minimum of 8 characters.	Password is too short - a minimum of 8 characters.
KJisCool	Ernest	edmund@gmail.com	p@ssw0rd	p@ssw0rd	6 Jun 1948	190	100.6	Female	91234000	jpg file	Password must contain at least one Uppercase, Lowercase, one number and special character	Password must contain at least one Uppercase, Lowercase, one number and special character
ThomasTrain	Ernest	edmund@gmail.com	Passw0rd	Passw0rd	15 Nov 1967	200	110.1	Others	91234000	png file	Password must contain at least one Uppercase, Lowercase, one number and special character	Password must contain at least one Uppercase, Lowercase, one number and special character
WhinnyBear	Ernest	edmund@gmail.com	P@ssword	P@ssword	2 Aug 2000	210	120	Male	91234000	empty file	Password must contain at least one Uppercase, Lowercase, one number and special character	Password must contain at least one Uppercase, Lowercase, one number and special character
HelloThere	Ernest	edmund@gmail.com	P@SSW0RD	P@SSW0RD	5 May 2000	215	330.4	Female	91234000	jpg file	Password must contain at least one Uppercase, Lowercase, one number and special character	Password must contain at least one Uppercase, Lowercase, one number and special character
JackTrades	Ernest	edmund@gmail.com	P@ssw0rd	p@ssw0rd	3 Feb 1984	216.5	140.9	Others	91234000	png file	Passwords must match	Passwords must match
JackTrades	Ernest	edmund@gmail.com	P@ssw0rd	Null input	3 Feb 1984	216.5	140.9	Others	91234000	png file	No password provided!	No password provided!

BlueBalls	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	31 Dec 1899	115	150	Male	91234000	empty file	You cannot choose a date before this!	You cannot choose a date before this!
BlueWaffle	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	4 April 2014	195	160	Female	91234000	jpg file	You cannot be younger than 10 years old!	You cannot be younger than 10 years old!
BlueWaffle	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	Null input	195	160	Female	91234000	jpg file	Please enter your date of birth	Please enter your date of birth
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	9 Mar 2009	-20	170	Others	91234000	png file	Must be at least 90cm	Must be at least 90cm
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	7 Oct 1995	89.9	180	Male	91234000	empty file	Must be at least 90cm	Must be at least 90cm
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	7 Jul 1974	220.1	190.1	Female	91234000	jpg file	Cannot be more than 220 cm	Cannot be more than 220 cm
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	7 Jul 1974	Null input	190.1	Female	91234000	jpg file	Please enter your height in cm	Please enter your height in cm
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	1 Dec 1957	201	19.9	Others	91234000	png file	Must be at least 20 kg	Must be at least 20 kg
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	141	-1	Male	91234000	Empty file	Must be at least 20 kg	Must be at least 20 kg
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	185.6	400.9	Female	91234000	jpg file	Cannot be more than 400 kg	Cannot be more than 400 kg
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	185.6	Null input	Female	91234000	jpg file	Please enter your weight in kg	Please enter your weight in kg
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	180	80	Female	9234	jpg file	Phone number is not valid	Phone number is not valid
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	180	80	Female	78911234	jpg file	Phone number is not valid	Phone number is not valid

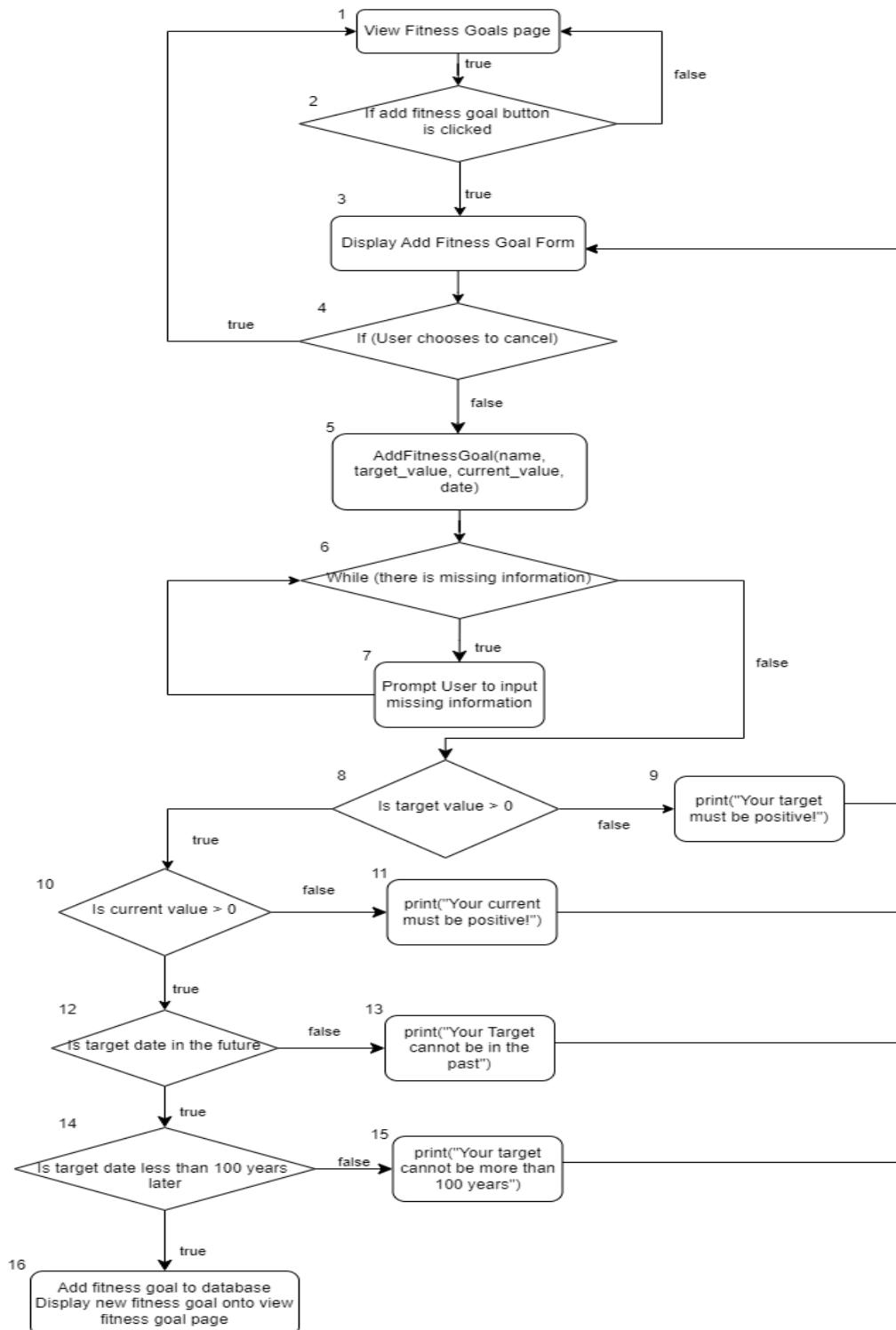
Software Requirements Specification for TrainTogether v5.3

BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	180	80	Female	Null input	jpg file	Please enter your mobile number	Please enter your mobile number
BlueBird	Ernest	edmund@gmail.com	P@ssw0rd	P@ssw0rd	29 Jan 1978	190	220	Others	91234000	txt file	An error have occurred. Please follow the direction Only accept image files with extension of GIF, JPG, PNG	An error have occurred. Please follow the direction Only accept image files with extension of GIF, JPG, PNG

8.7.2 White Box Testing

For White Box Testing we use basic path testing for two test cases: Adding Fitness Goals and Registering Users.

1. Adding Fitness Goals



Cyclomatic Complexity = $|edges| - |nodes| + 2 = 22 - 16 + 2 = 8$

Cyclomatic Complexity = $|decision point| + 1 = 7 + 1 = 8$

Basic Paths

Path 1 (baseline): 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16

Path 2: 1, 2, 1

Path 3: 1, 2, 3, 4, 1

Path 4: 1, 2, 3, 4, 5, 6, 7, 6, 8, 10, 12, 14, 16

Path 5: 1, 2, 3, 4, 5, 6, 8, 9, 3, 4, 5, 6, 8, 10, 12, 14, 16

Path 6: 1, 2, 3, 4, 5, 6, 8, 10, 11, 3, 4, 5, 6, 8, 10, 12, 14, 16

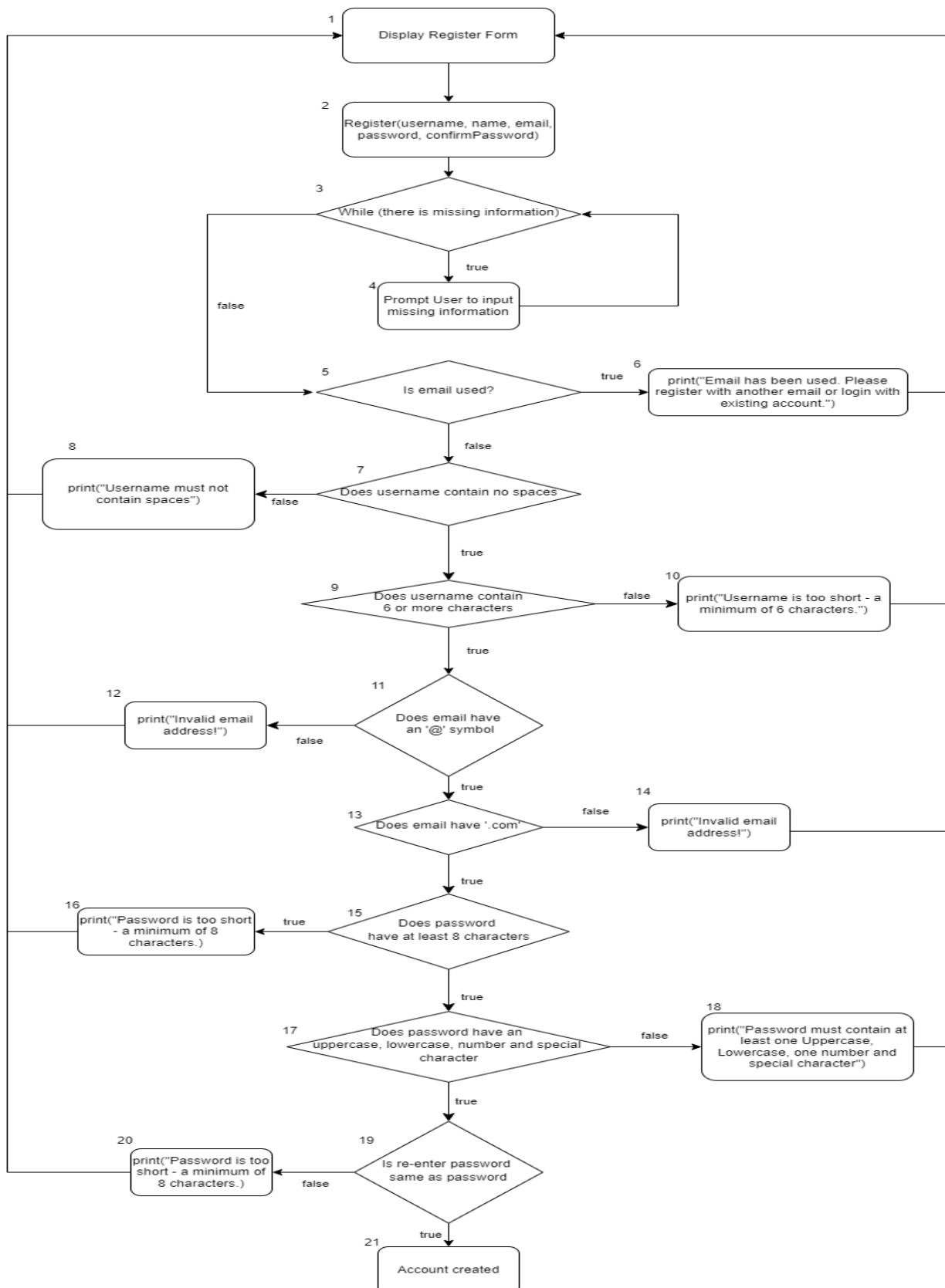
Path 7: 1, 2, 3, 4, 5, 6, 8, 10, 12, 13, 3, 4, 5, 6, 8, 10, 12, 14, 16

Path 8: 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 15, 3, 4, 5, 6, 8, 10, 12, 14, 16

Test cases

Path	Clicked	Cancel	Missing Information	Target Value	Current Value	Target Date	Expected Result	Test Result
1	True	False	False	30	20	3 Aug 2023	Fitness Goal Created	Fitness Goal Created
2	False	False	-	-	-	-	Redirected to Fitness Goal Page	Redirected to Fitness Goal Page
3	True	True	-	-	-	-	Redirected to Fitness Goal Page	Redirected to Fitness Goal Page
4	True	False	True	30	NULL	3 Aug 2023	You must have a current value!	You must have a current value!
5	True	False	False	-1	20	3 Aug 2023	Your target must be positive	Your target must be positive
6	True	False	False	30	-1	3 Aug 2023	Your current must be positive!	Your current must be positive!
7	True	False	False	30	20	3 Aug 2023	Your Target cannot be in the past	Your Target cannot be in the past
8	True	False	False	30	20	3 Aug 2223	Your target cannot be more than 100 years	Your target cannot be more than 100 years

1. Register users



Cyclomatic Complexity = |edges| - |nodes| + 2 = 29 - 21 + 2 = 10

Cyclomatic Complexity = |decision point| + 1 = 9 + 1 = 10

Basic Paths

Path 1 (baseline): 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 2: 1, 2, 3, 4, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 3: 1, 2, 3, 5, 6, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 4: 1, 2, 3, 5, 7, 8, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 5: 1, 2, 3, 5, 7, 9, 10, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 6: 1, 2, 3, 5, 7, 9, 11, 12, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 7: 1, 2, 3, 5, 7, 9, 11, 13, 14, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 8: 1, 2, 3, 5, 7, 9, 11, 13, 15, 16, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 9: 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 18, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Path 10: 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 20, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21

Test cases

Path	Username	Name	Email	Password	Re-enter password	Date of Birth	Expected Result	Test Result
1	edmund562	Ernest	user@gmail.com	P@ssw0rd	P@ssw0rd	3 April 2013	Account Created	Account Created
2	edmund562	Null input	user@abc.com	P@ssw0rd	P@ssw0rd	3 April 2013	No name provided!	No name provided!
3	edmund562	Ernest	user@gmail.com	P@ssw0rd	P@ssw0rd	3 April 2013	Email has been used. Please register with another email or login with existing account.	Email has been used. Please register with another email or login with existing account.
4	edmund 562	Ernest	user@abc.com	P@ssw0rd	P@ssw0rd	3 April 2013	Username must not contain spaces	Username must not contain spaces
5	edmun	Ernest	user@abc.com	P@ssw0rd	P@ssw0rd	3 April 2013	Username is too short - a minimum of 6 characters.	Username is too short - a minimum of 6 characters.

Software Requirements Specification for TrainTogether v5.3

6	edmund562	Ernest	usergmail.com	P@ssw0rd	P@ssw0rd	3 April 2013	Invalid email address	Invalid email address
7	edmund562	Ernest	user@abc	P@ssw0rd	P@ssw0rd	3 April 2013	Invalid email address	Invalid email address
8	edmund562	Ernest	user@abc.com	P@ssw0r	P@ssw0r	3 April 2013	Password is too short - a minimum of 8 characters.	Password is too short - a minimum of 8 characters.
9	edmund562	Ernest	user@abc.com	password	P@ssw0r	3 April 2013	Password must contain at least one Uppercase, Lowercase, one number and special character	Password must contain at least one Uppercase, Lowercase, one number and special character
10	edmund562	Ernest	user@abc.com	P@ssw0rd	Passw0rd	3 April 2013	Passwords must match	Passwords must match