

---

# **Software Requirements Specification**

**for**

## **PathFinder**

**Prepared by**

- 1. Nafees Abdul Kader, U2222914B**
- 2. Lim Ricky, U2221808A**
- 3. Yong Wei Tuck, U2220228A**
- 4. Bryan Chia Song Nian, U2222849J**
- 5. Jordan Ng Teng Kiat, U2222317D**
- 6. Pushparajan Roshini, U2222546A**

**Nanyang Technological University, SCSB Team 5**

**21/4/2024**

# Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	2
1.4 Product Scope and Functions	2
1.5 References	2
1.6 Links	3
2. Overall Description	3
2.1 Product Perspective	3
2.2 User Class and Characteristics	3
2.3 Operating Environment	3
2.4 Design and Implementation Constraints	4
2.5 User Documentation	4
2.6 Assumptions and Dependencies	4
3. External Interface Requirements	5
3.1 User Interfaces	5
3.2 Hardware and Software Interfaces	10
4. System Features	11
4.1 Registration	11
4.2 Login	12
4.3 Forgot Password	13
4.4 Edit Profile	15
4.5 Input Destination Details	16
4.6 Prompt Location Permission	17
4.7 Detect Current Location	18
4.8 Display Trip Information	19
4.9 Get Route	20
4.10 Get Duration	21
4.11 Get Price	22
4.12 Confirm Trip	23
4.13 View Trip History	24
4.14 Edit Trip	25
4.15 Manage Wallet Balance	27
4.16 Set Expenditure Limit	28
4.17 Notify Low Balance	30
4.18 Logout	30

5. Other Nonfunctional Requirements	32
5.1 Performance Requirements	32
5.2 Safety Requirements	32
5.3 Security Requirements	32
5.4 Software Quality Attributes	32
6. Testing	32
6.1 Black Box Testing	32
6.2 White Box Testing	41
7. Appendix A: Data Dictionary	45
8. Appendix B: Analysis Models	46
9. Appendix C: Future Improvements	58

## Revision History

Name	Date	Reason For Changes	Version
Nafees Abdul Kader	27/02/24	Addition of a new use case - Logout	1.1
Pushparajan Roshini	27/02/24	Edited mostly all use cases according to new mock up flow	1.2
Bryan Chia	28/03/24	Addition of new use case - Forgot Password	1.3
Bryan Chia	03/04/24	Addition of black box testing	1.4
All	21/04/2024	Edited entire document according to new changes	1.5

# **1. Introduction**

## **1.1 Purpose**

Many Singaporeans nowadays prefer travelling via public transport or private hire rather than owning a new car. One reason could be due to the drastic increase in the Certificate of Entitlement (COE). Another reason is Singaporeans' proactive stance towards environmental preservation, leading to a reduction in the number of vehicles on the road. There could be a multitude of other factors contributing to this shift in transportation preferences. While people might find public transport or private hire to be an easier option, there are 2 factors that can make their lifestyle harder.

One of the factors include the rise in transportation fare for such services. The adult price fare for the bus services and Mass Rapid Transit (MRT) services are increasing yearly, while the fares for taxis have consistently been very high. These fares could get even worse during the peak hours. Another factor involves the time and distance of their travel. This is especially important for long commutes across Singapore, impacting individuals' transportation choices, particularly for school or work-related travel.

These 2 factors make planning travel in Singapore challenging. Some prefer taxis for shorter travel times but may face financial constraints if they were to travel daily. Some people may have a hard time monitoring their transportation expenditure. With all these different possibilities and scenarios, our website addresses these issues by providing trip tracking and planning services.

Our website provides users with various transportation options from current location to destination along with their associated costs and travel time. This enables users to plan their trips accordingly. For example, if time is limited, users can prioritise shorter routes, even if they entail higher costs. Whereas, if time is more flexible, users can opt for longer routes to save money.

Our website also stores all the trips that they have taken, which the user can access through the View Trip History. This feature, along with enables users to monitor their expenditure and travel time, facilitating better trip planning for future journeys.

In a nutshell, we want to provide Singaporeans a platform to better plan their trip, with the considerations of cost, time and duration, without interface, without any stress. The main goal is to push more residents to use public transport willingly, as they are more in control of their navigation as well as expenditure.

## **1.2 Document Conventions**

- Use of headings and subheadings to break the document into sections or subsections such as 1 for section head and 1.1 for subsections.
- Use of tables and diagrams (such as class diagram, use case description or sequence

diagram, etc) to present the content visually in helping the reader visualise the entire flow or features of the application.

- Use of consistent and understandable naming convention is used for the requirements of the project in ensuring that it is easily trackable.
- Use of bullet points to easily describe the details in point form for better understanding.

### 1.3 Intended Audience and Reading Suggestions

Our intended audience would be all the stakeholders who commute in Singapore. This document helps current developers create a logical development process to produce the web application.

Our documentation consists of the description and functionalities of the website, the different detailed procedures of our thought process in curating the features in diagrams and descriptions as well as the user interfaces. As such, this documentation is intended to be read in chronological sequence by all the stakeholders.

### 1.4 Product Scope and Functions

PathFinder provides users with a comprehensive trip planner service. It offers real-time data on available routing services based on the user's entered starting and ending address and includes GPS navigation to efficiently locate the user's current location. Key objectives of PathFinder include :

1. Displaying top 2 Transit and top 1 Taxi route within 1-km radius of the user's current location for convenient travel
  - a. Corresponding Duration
  - b. Corresponding Fare
  - c. Corresponding Trip Details
2. Providing clear, step-by-step instructions on each Transit route from start to end point
3. Summary of all the trips user has taken
  - a. Status to indicate if trip has been completed or is ongoing
4. Allowing user to set up monthly finds
  - a. Wallet automatically deducts fund as each trip is made
  - b. Visual indicator of funds left in account

### 1.5 References

1. One Map API Documentation  
<https://www.onemap.gov.sg/apidocs/apidocs/#search>
2. Land Transport Authority (LTA) Singapore Bus Timings and Taxi Availability Data, data retrieved from <https://datamall.lta.gov.sg/content/datamall/en.html>
3. Additional Taxi API Availabilities  
<https://api.data.gov.sg/v1/transport/taxi-availability>
4. Similar Reference for our trip planner  
<https://www.tripit.com/app/trips>

## **1.6 Links**

### **1.6.1 Github Repository**

Github Link : [https://github.com/scsbTeam5/Team-5\\_SWE.git](https://github.com/scsbTeam5/Team-5_SWE.git)

### **1.6.2 Demo Video**

Demo Video Link :

[https://entuedu-my.sharepoint.com/:v:/g/personal/limr0035\\_e\\_ntu\\_edu\\_sg/EYwAe2isrpBNmpkfUXGMxSYBQvLFLWMI-F8p61ORJHJdQg?nav=eyJyZWZlcnJhbEluZm8iOncicmVmZXJyYWxBcHAI0iJPbmVEcmI2ZUZvckJlc2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=geGTpy](https://entuedu-my.sharepoint.com/:v:/g/personal/limr0035_e_ntu_edu_sg/EYwAe2isrpBNmpkfUXGMxSYBQvLFLWMI-F8p61ORJHJdQg?nav=eyJyZWZlcnJhbEluZm8iOncicmVmZXJyYWxBcHAI0iJPbmVEcmI2ZUZvckJlc2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=geGTpy)

## **2. Overall Description**

### **2.1 Product Perspective**

PathFinder will be presented in the form of a web application. Therefore, internet connection or cellular data connection is definitely needed to access this website.

With an interface that is easy to manoeuvre with, PathFinder aims to provide users a platform to physically view all the possible ways to move from point A to point B, the different transportations that they can use, and the amount of money they would spend for each trip. The users can also monitor their transportation expenditure by viewing the trip history, allowing them to better plan their expenditure in the future and reflecting on where they cut if they over exceed their monthly budget.

### **2.2 User Class and Characteristics**

Users who commute and seek autonomy over their travel routes and expenses rely on PathFinder to effortlessly plan their journeys. By offering diverse transportation options, cost analysis, and trip history tracking, PathFinder empowers users to make informed decisions, ensuring efficient and budget-friendly commuting experiences.

## **2.3 Operating Environment**

PathFinder can be operated on all browsers. Any desktop or phone with a browser and with GPS activated, connected to the Internet could provide access to our web application. Additionally, the APIs we used from OneMap as well as LTA should be working.

## **2.4 Design and Implementation Constraints**

Loading and filtering a whole list of available taxi coordinates and fetching the nearest taxi hampers the search efficiency and slows down the display of taxi's data on the website. Consequently, users need to wait a short while for the application to be able to retrieve the accurate taxi details.

The second constraint is due to the substantial usage of the external API - OneMap API. Users need to type in their addresses or postal codes correctly according to OneMap.gov.sg's input expectations. If the address is keyed incorrectly, no routes or wrong routes will be fetched. Also, heavy reliance on external API can result in applications not being able to function properly once an API data provider experiences any technical issues.

## **2.5 User Documentation**

A simple user manual with screenshots is included in 3.1 in the form of UI Mockup. The video also guides users with the main features of PathFinder.

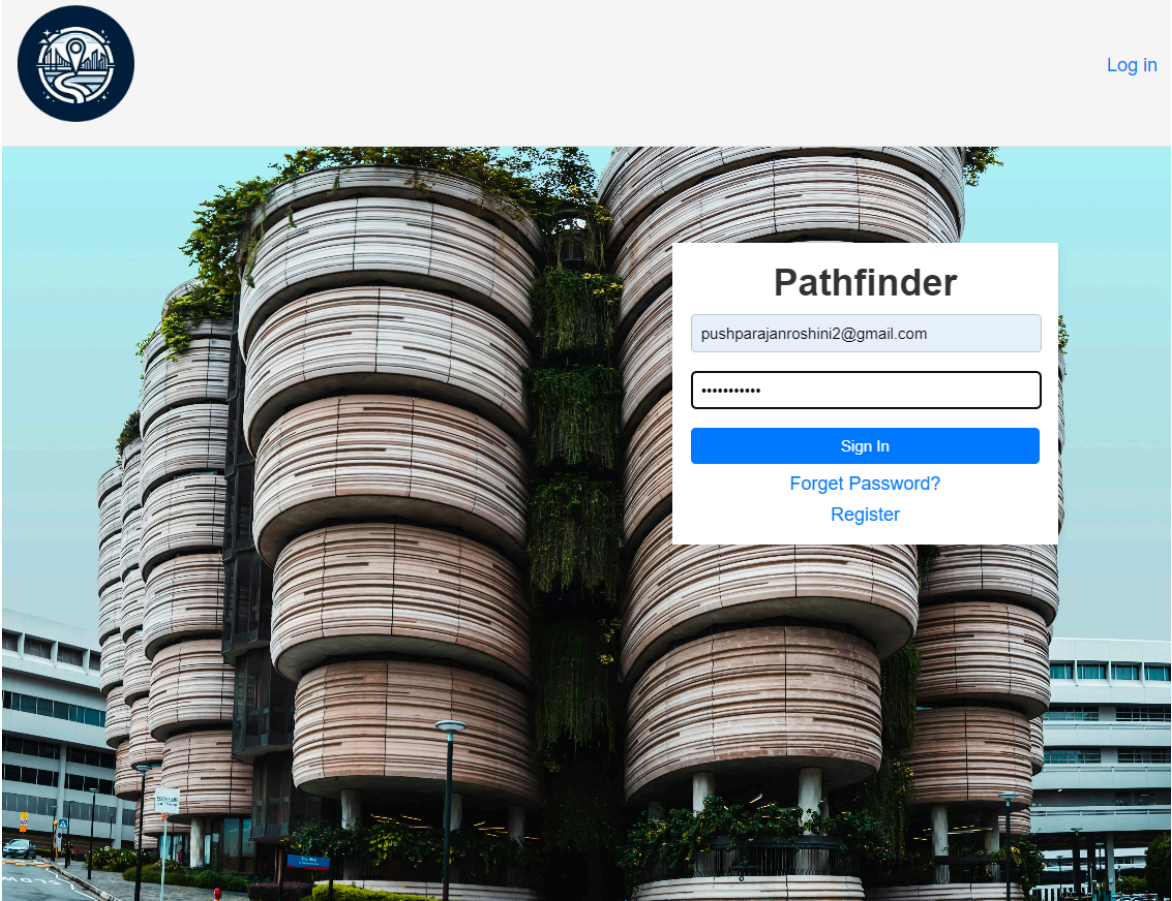
## **2.6 Assumptions and Dependencies**

It is assumed that users are inputting valid Singapore-based addresses in the input fields. It is also assumed the user has a stable internet connection as the application leverages on several APIs which rely on the internet. Navigation system would also be convenient for users if they had enabled GPS.

### 3. External Interface Requirements

#### 3.1 User Interfaces

##### 3.1.1 Login Page

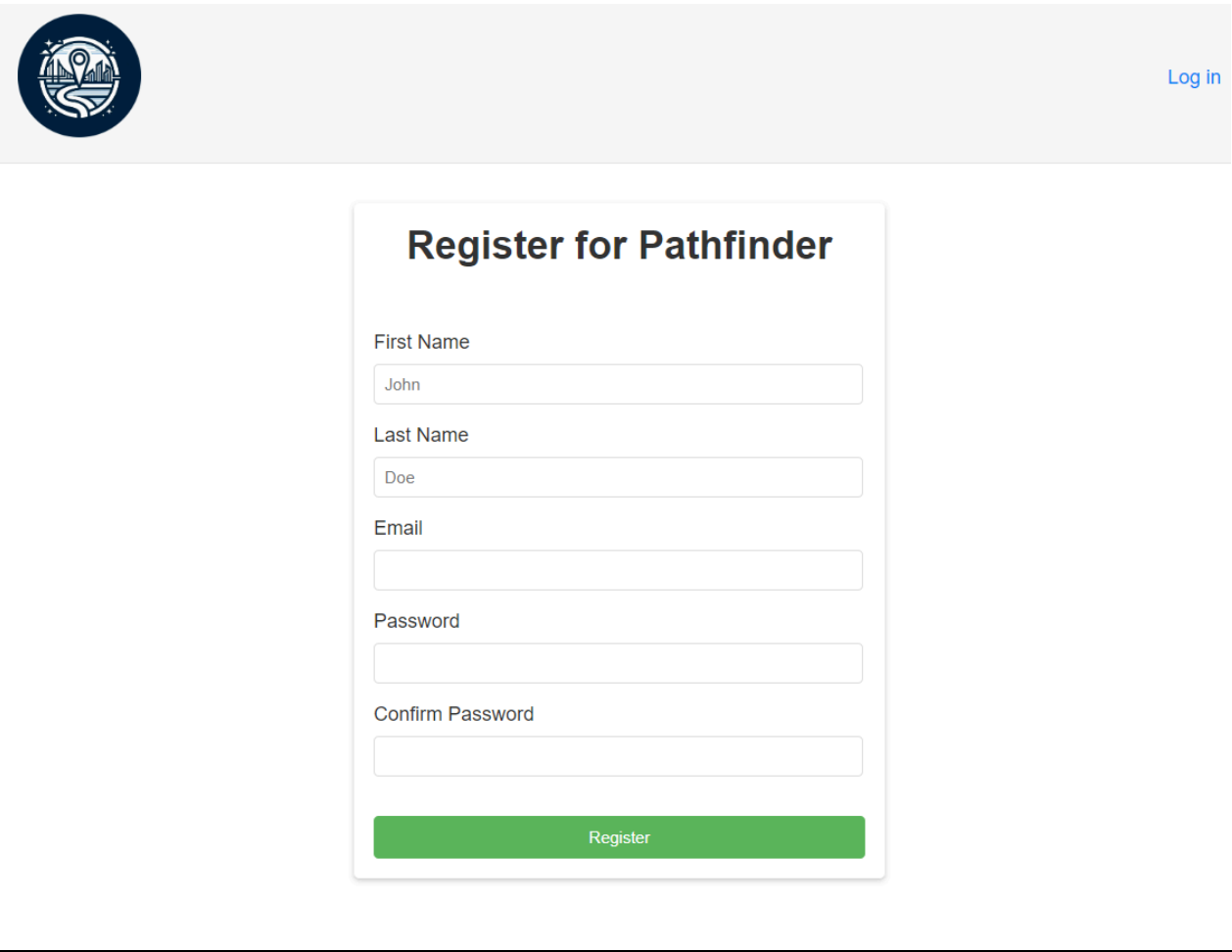


1. This is the homepage.

2. Users can choose to 'Log In', 'Register or change their password by clicking on 'Forgot Password'



### 3.1.2 Register Page



The image shows a web registration form titled "Register for Pathfinder". The form is centered on a light gray background. At the top left of the background is a circular logo with a stylized path and a location pin. At the top right is a "Log in" link. The form itself has a white background and a thin gray border. It contains the following fields: "First Name" (with "John" entered), "Last Name" (with "Doe" entered), "Email", "Password", and "Confirm Password". Each field is a white rectangular box with a light gray border. Below the "Confirm Password" field is a green "Register" button with white text. Below the form, there is a list of three requirements for the registration process.

**Register for Pathfinder**

First Name

Last Name

Email

Password


Confirm Password

[Log in](#)

**Register**

1. Users can create an account by filling up the respective details required by the website.
2. All fields need to be correctly inputted in order to Register.
3. Error-handling mechanisms are included when the user attempts to input incorrectly.

### 3.1.3 Forget Password Page

Log in


## Reset Password

Email:

Reset Password

1. Users can choose to reset password by typing in a valid registered email address
2. Reset password email gets sent to their registered email address inbox.

### 3.1.4 Add Trip Page

Add Trip Trip History Edit Profile Manage Wallet Balancelocalhost:3000 saysWelcome, Roshini

Start Address

SIMS VISTA MARKET AND FOOD CENTRE 49 SI

End Destination

Parkway Parade

Get routes

Fastest Route ☒ Cheapest Route ☐

#### Transit

Route 1

Duration (minutes): 26

Fare: \$1.30

Select

Leg 1:

Mode: WALK

Bus Number / MRT Line: N/A

From: Origin

To: BLK 49 MKT/FC

Leg 2:

Mode: BUS

Bus Number / MRT Line: 134

From: BLK 49 MKT/FC

Next Bus Arrival Time: 18:01:29

To: OPP PARKWAY PARADE

Leg 3:

Mode: WALK

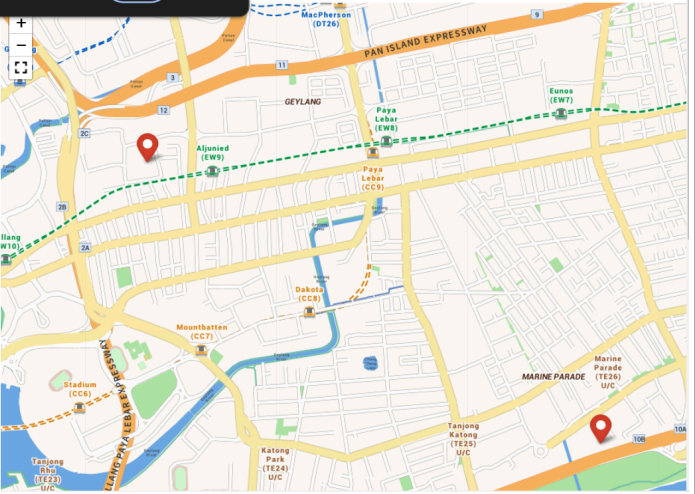
#### Taxi

Duration: 0 hours 9 minutes

Distance (km): 4.70

Fare: \$11.25

Select



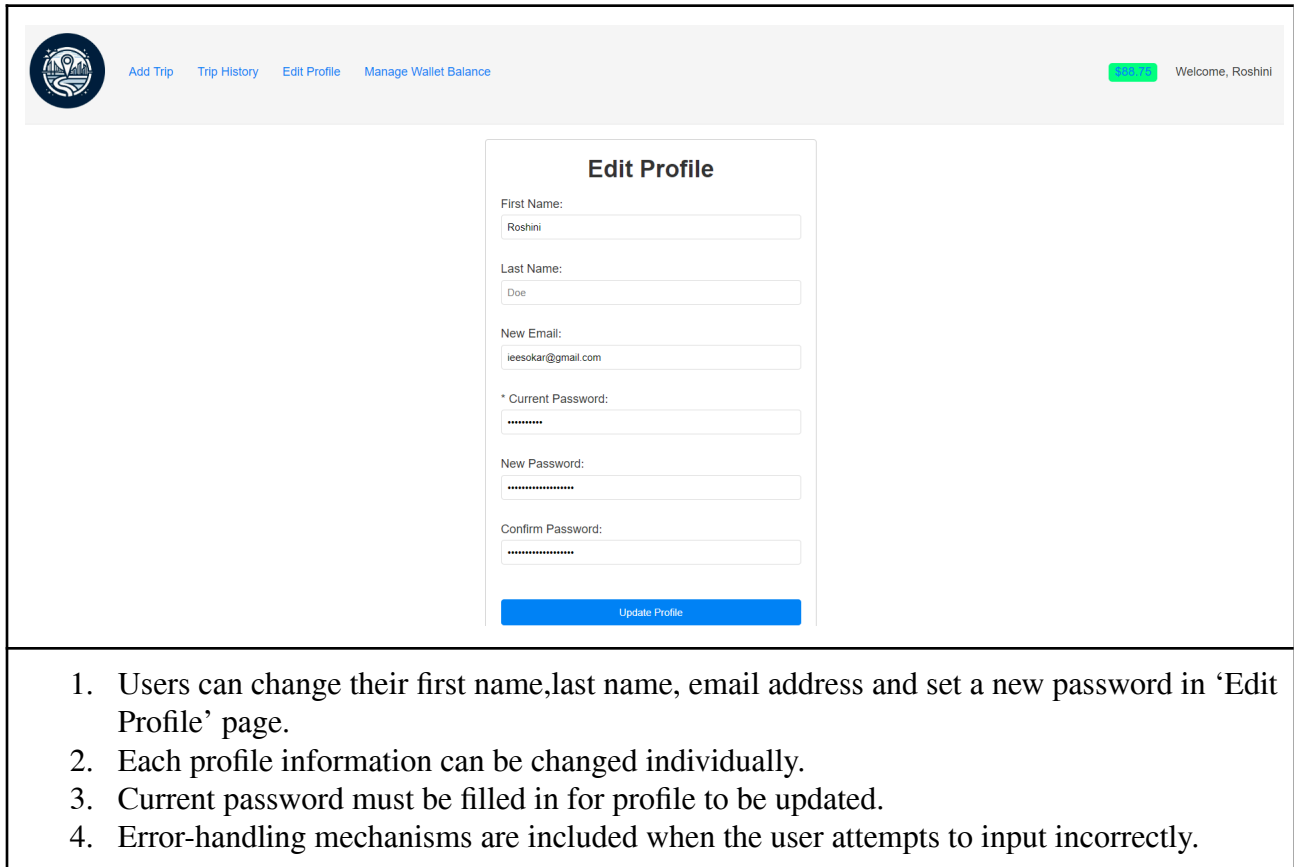
1. Users can enable GPS and have their current address to be autofilled or can choose to fill it up manually
2. After indicating the end destination, they obtain the possible routes they can choose (from transit and taxi), with filtering options of cheapest and fastest route.
3. Users select their preferred route and confirm their trip in the pop-up.

### 3.1.5 Trip History Page

The screenshot displays the 'Trip History' page of the PathFinder application. At the top, there is a navigation bar with a logo on the left and links for 'Add Trip', 'Trip History', 'Edit Profile', and 'Manage Wallet Balance' in the center. On the right, a green button labeled 'Log Out' and the text 'Welcome, Roshini' are visible. Below the navigation bar, a date selector shows 'Apr' with left and right arrows. The page features three summary statistics: 'Total Transit Fare: \$1.30', 'Total Taxi Fare: \$11.25', and 'Overall Total Fare: \$12.55'. The main content area lists two trips. The first trip is from 'SIMS VISTA MARKET AND FOOD CENTRE 49 SIMS PLACE' to 'Parkway Parade', with a taxi duration of '0 hours 9 minutes' and a fare of '\$11.25'. Its status is 'Ongoing', and there is an 'Update' button. The second trip is from '122 PAYA LEBAR WAY' to 'Parkway Parade', with a transit duration of '30 min' and a fare of '\$1.30'. Its status is 'Completed', and there is an 'Update' button. At the bottom of the trip list, there are three buttons: 'Ongoing', 'Completed', and 'Save'.

1. Once confirming a trip, users land on the 'Trip History' Page where they can see their ongoing and completed trips.
2. Trips are sorted according to months and their monthly funds are deducted according to the amount of money they spend on their transport.

### 3.1.6 Edit Profile Page



The screenshot displays the 'Edit Profile' page of the PathFinder application. The header includes a logo on the left, navigation links ('Add Trip', 'Trip History', 'Edit Profile', 'Manage Wallet Balance') in the center, and a user greeting ('Welcome, Roshini') on the right. The main content area contains a form titled 'Edit Profile' with the following fields:

- First Name: Roshni
- Last Name: Doe
- New Email: ieesokar@gmail.com
- \* Current Password: [masked]
- New Password: [masked]
- Confirm Password: [masked]

At the bottom of the form is a blue button labeled 'Update Profile'.

1. Users can change their first name, last name, email address and set a new password in 'Edit Profile' page.
2. Each profile information can be changed individually.
3. Current password must be filled in for profile to be updated.
4. Error-handling mechanisms are included when the user attempts to input incorrectly.

### 3.1.7 Manage Wallet Balance Page

The screenshot displays two states of the 'Manage Wallet' page. The top state shows the page with a green 'Set \$100' button and a 'Welcome, Roshini' message. The bottom state shows the page with a red '\$1.00' button and the same 'Welcome, Roshini' message. Both states feature a navigation bar with links: 'Add Trip', 'Trip History', 'Edit Profile', and 'Manage Wallet Balance'. The main content area includes a 'Set Monthly Funds' input field with '\$100', a checkbox for 'Alert When Fund Below 10%' which is checked, and an 'Update' button.

1. Users can set monthly funds in their wallet.
2. Wallet funds' indicator get marked in red once funds drop below the threshold of 10 percent (of their initial monthly funds) to warn users to be mindful of expenditure.

## 3.2 Hardware and Software Interfaces

Web application supports various device types, including desktop and mobile devices. PathFinder is built with Javascript, HTML and CSS in React.js framework and interfaces with other software components such as Firebase and Firestore API, OneMap API and LTA Datamall API. Firestore Realtime Database and Firebase Authentication API were used for data storage and retrieval of the trips made by the user, along with the route's duration and cost. User's wallet expenses were also stored and retrieved real-time to display on screen.

## 4. System Features

### 4.1 Registration

Use Case ID:	RG1		
Use Case Name:	Register		
Created By:	JORDAN NG TENG KIAT	Last Updated By:	NAFEES ABDUL KADER
Date Created:	12/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	Allows the user to register an account with an email address and password.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user does not have an existing account with the same email address.</li> <li>2. The user enters a valid email address and password.</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The user's account is stored in the Firebase database.</li> <li>2. The user is brought to the Home page.</li> </ol>
Priority:	High
Frequency of Use:	Once per user
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system prompts the user to enter their first and last name, email address, password and password confirmation.</li> <li>2. The user enters first and last name, email address, password and password confirmation.</li> <li>3. The user clicks the 'Register' button.</li> <li>4. The system checks for valid email addresses and passwords.</li> <li>5. The system stores details into the database.</li> <li>6. The system redirects user to Home page.</li> </ol>
Alternative Flows:	<p>AF-S4: If email address already exists within database</p> <ol style="list-style-type: none"> <li>1. The System displays "Email already exists. Please log in or use a different email address"</li> <li>2. System returns to step 1.</li> </ol> <p>AF-S4: If user's password does not meet the requirements</p> <ol style="list-style-type: none"> <li>1. System displays "Password must be at least 6 characters long and include at least one special character."</li> <li>2. System returns to step 1.</li> </ol> <p>AF-S4: If the the email address has no "@"</p> <ol style="list-style-type: none"> <li>1. The system displays the message "Please include an '@' in the email address."</li> <li>2. The system returns to step 1.</li> </ol>
Exceptions:	Nil
Includes:	Nil
Extends :	Login
Special Requirements:	Nil
Assumptions:	User is connected to the internet.
Notes and Issues:	Nil

Functional Requirements:

1. The user must be able to register for a new account.
  - 1.1. The system must display five text fields for the user to enter information.
    - 1.1.1. The first text field holds the user's first name.
    - 1.1.2. The second text field holds the user's last name.
    - 1.1.3. The third text field holds the user's email address.
    - 1.1.4. The fourth text field holds the user's password.
    - 1.1.5. The fifth text field holds the confirmation of the user's password.
  - 1.2. When the user enters information and clicks "Register", the system must validate the information entered.
    - 1.2.1. The system must check for a valid email address containing a single '@' symbol.
      - 1.2.1.1. If the email address does not contain a single '@', the system displays a message of "Please include an '@' in the email address."
    - 1.2.2. The system must check for a valid password following the requirements.
      - 1.2.2.1. The password must contain at least 6 characters.
      - 1.2.2.2. The password must contain at least 1 special character.
  - 1.3. When the system successfully validates the user's information, the system must store the user's account in the database.
  - 1.4. The system must bring the user to the Home page.

**4.2 Login**

Use Case ID:	LOG1		
Use Case Name:	Login		
Created By:	JORDAN NG TENG KIAT	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	12/02/2024	Date Last Updated:	27/02/2024

Actor:	User
Description:	Allows the user to login with the user's existing email address and corresponding password.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has entered a valid email address.</li> <li>2. The user's account exists in the system's database.</li> <li>3. The user has entered the correct password corresponding to the email address.</li> </ol>
Postconditions:	The user is brought to the home page.
Priority:	High
Frequency of Use:	Upon every log out.
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system prompts the user to enter an email address and password.</li> <li>2. The user enters email address and password</li> <li>3. The user clicks the 'Sign In' button.</li> <li>4. The system checks for a valid email address.</li> </ol>

	5. The system checks if the user's account exists in the system's database. 6. The system checks if the user has entered the correct password corresponding to the email address. 7. If the user's information is correct, the system will bring the user past the login page to the home page.
Alternative Flows:	AF-S4: If the user enters an invalid email address 1. The system displays the message "Invalid email address." 2. The system returns to step 1. AF-S5: If the system fails to find the user's account in the database 1. The system displays the message "Incorrect email or password." 2. The system returns to step 1. AF-S6: If the user entered an incorrect password corresponding to the email address 1. The system displays the message "Incorrect email or password." 2. The system returns to step 1.
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	The user is connected to the internet.
Notes and Issues:	Nil

### Functional Requirements:

1. The user must be able to login.
  - 1.1. The system must display two text fields for the user to enter information.
    - 1.1.1. The text field above holds the user's email address.
    - 1.1.2. The text field below holds the user's password.
  - 1.2. When the user enters information and clicks "Sign In", the system must validate the information entered.
    - 1.2.1. The system must check for a valid email address containing a single '@' symbol.
    - 1.2.2. The system must check if the user's account exists within the database using the email address.
    - 1.2.3. The system must check if the password is correct corresponding to the email address.
  - 1.3. When the system has successfully validated the user's information, the system must bring the user past the login page to the home page.

## 4.3 Forgot Password

Use Case ID:	FP1		
Use Case Name:	Forgot Password		
Created By:	BRYAN CHIA SONG NIAN	Last Updated By:	NAFEES ABDUL KADER
Date Created:	23/03/2024	Date Last Updated:	21/04/2024



Actor:	User
Description:	If the user forgets their password, they can enter their registered email address to trigger a reset.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user has clicked on “Forget Password?” on the login page</li> <li>2. The user has entered a valid email address.</li> <li>3. The user’s account exists in the system’s database.</li> </ol>
Postconditions:	Nil
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system prompts the user to enter an email address.</li> <li>2. The user enters an email address.</li> <li>3. The user clicks the ‘Reset Password’ button.</li> <li>4. The system checks for valid email addresses.</li> <li>5. The system checks if the user’s account exists in the system’s database.</li> <li>6. If the user’s email is valid and registered, the system sends an email containing a link to reset their password.</li> <li>7. The user follows the link to reset their password.</li> </ol>
Alternative Flows:	<p>AF-S4: If the user enters an invalid email address</p> <ol style="list-style-type: none"> <li>1. The system displays the message “Invalid email address.”.</li> <li>2. The system returns to step 1.</li> </ol> <p>AF-S4: If the user does not include a ‘@’ in the email address</p> <ol style="list-style-type: none"> <li>1. The system will show “Please include an ‘@’ in the email address”.</li> </ol>
1	Nil
Includes:	Nil
Extends:	Login
Special Requirements:	Nil
Assumptions:	The user is connected to the internet.
Notes and Issues:	Nil

### Functional Requirements:

1. The user must be able to reset their password after selecting “Forget Password?” on the login page.
  - 1.1. The system must display a text field for the user to enter information.
    - 1.1.1. The text field holds the user’s email address.
  - 1.2. When the user enters information and clicks “Reset Password”, the system must validate the information entered.
    - 1.2.1. The system must check for a valid email address containing a single ‘@’ symbol.

1.2.1.1. If the email address does not contain '@', it will mention "Please include an '@' in the email address".

1.2.2. The system must check if the user's account exists within the database using the email address.

1.3. When the system has successfully validated the user's information, the system must send a link for the user to reset their password

1.4. The user has to enter a new password meeting the requirements.

## 4.4 Edit Profile

Use Case ID:	EP1		
Use Case Name:	Edit Profile		
Created By:	JORDAN NG TENG KIAT	Last Updated By:	PUSHPARAJAN ROSHNI
Date Created:	12/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	User edits their own profile
Preconditions:	1. The user is logged in.
Postconditions:	1. The new profile changes are updated in the database. 2. The new profile is displayed
Priority:	Low
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user clicks on the Edit Profile tab.</li> <li>2. The system brings the user to the profile page.</li> <li>3. The system displays the user's profile with an "Update Profile" button beside each individual information.</li> <li>4. The user edits the desired information.</li> <li>5. The user clicks the "Save changes" button.</li> <li>6. The system checks if the new information is valid.</li> <li>7. The system saves and updates the new information of the user in the database.</li> </ol>
Alternative Flows:	AF-S5: If the user enters enters a new information that does not meet the requirements <ol style="list-style-type: none"> <li>1. The system displays the message "Error saving changes"</li> <li>2. The system returns to step 6.</li> </ol>
Exceptions:	Nil
Includes:	Login
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

### Functional Requirements:

1. The user must be able to edit their own profile
  - 1.1. The system must display the user profile.

- 1.1.1. The system must display a text box for users to update their first name.
  - 1.1.1.1. The current first name is automatically filled in.
- 1.1.2. The system must display a text box for users to update their last name.
- 1.1.3. The system must display a text box for users to update their new email address.
  - 1.1.3.1. The current email address is automatically filled in.
  - 1.1.3.2. The system must check that the new email address contains one “@” character.
  - 1.1.3.3. If the new email address is valid, the system must send a verification code to the new email address.
  - 1.1.3.4. The user must be able to enter the verification code received into the system.
  - 1.1.3.5. If the verification code is done successfully, the system must update the new email address on display and in the database.
- 1.1.4. The system must display a text box for users to type in their current password.
  - 1.1.4.1. This field is a must for users to fill up.
- 1.1.5. The system must display 2 text boxes for users to type in their new password.
  - 1.1.5.1. The first masked text box must be labelled as “New password”.
  - 1.1.5.2. The second masked text box must be labelled as “Confirm password”.
  - 1.1.5.3. The system must check that the password meets the requirements.
    - 1.1.5.3.1. The user’s new password must be at least 6 characters long and include at least one special character.
  - 1.1.5.4. The system must check that the new password matches the confirmed password.
  - 1.1.5.5. If the system successfully validates the new password, the system must update the new password in the database.
- 1.2. The system allows user to update any one of the above fields, and is not a must for users to fill up all the input boxes before clicking “Update Profile”

## 4.5 Input Destination Details

Use Case ID:	DD1		
Use Case Name:	Input Destination Details		
Created By:	YONG WEI TUCK	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	12/02/2024	Date Last Updated:	27/02/2024

Actor:	User
Description:	User enters the starting point and destination they want to go.
Preconditions:	1. The user is logged in. 2. The user clicks on “Add Trip” tab
Postconditions:	The system prompts user to give permission for location services.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The system goes to LP1 to prompt location permission

	2. User enters starting address and destination address 3. User clicks on Get Routes button
Alternative Flows:	AF-S2: If user allowed permission for location access, user starting address is fetched, and user only enters destination address
Exceptions:	Nil
Includes:	Login
Special Requirements:	Nil
Assumptions:	Assumed that addresses entered are in Singapore
Notes and Issues:	Nil

**Functional Requirements:**

1. Users must be able to input destination details after selecting “Add Trip” on home page
  - 1.1. User can input destination in the form of either:
    - 1.1.1. Address
    - 1.1.2. Name of place
    - 1.1.3. Postal Code
      - 1.1.3.1. System has default calendar inbuilt which presets the current date
2. System goes to LP1 to prompt for location permissions
  - 2.1. If user selects yes
    - 2.1.1. User’s current address is entered as starting address
  - 2.2. If user selects no
    - 2.2.1. System prompts user to enter a valid starting address in the form of either:
      - 2.2.1.1. Address
      - 2.2.1.2. Name of place
      - 2.2.1.3. Postal Code

**4.6 Prompt Location Permission**

Use Case ID:	LP1		
Use Case Name:	Prompt Location Permission		
Created By:	YONG WEI TUCK	Last Updated By:	YONG WEI TUCK
Date Created:	12/02/2024	Date Last Updated:	12/02/2024

Actor:	User
Description:	Prompts user to allow location permission
Preconditions:	1. The user is logged in.
Postconditions:	The system has permission to access user’s live location
Priority:	High
Frequency of Use:	High
Flow of Events:	1. System prompts user to allow permission for location access

Alternative Flows:	AF-S2: User does not want to allow permission for location access 1. System goes to DD1 for user to manually input starting address AF-S2: User allows permission for location access 1. System uses user's live location as starting point
Exceptions:	Nil
Includes:	Login
Extends:	Detect Current Location
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

**Functional Requirements:**

1. System prompts user to give permission for location access after clicking on "Add Trip" button on home page
  - 1.1. If user does not give permission, the system goes back to Input Destination Details (DD1) for user to manually input starting address
  - 1.2. If user gives permission, system uses user's live location as starting address

**4.7 Detect Current Location**

Use Case ID:	CL1		
Use Case Name:	Detect Current Location		
Created By:	YONG WEI TUCK	Last Updated By:	YONG WEI TUCK
Date Created:	12/02/2024	Date Last Updated:	12/02/2024

Actor:	User
Description:	Detect user's current location
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has allowed permission for location access (LP1)</li> </ol>
Postconditions:	The system gets user's current location information
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User allows permission for location access in LP1</li> <li>2. System gets user current location information</li> <li>3. User's current location information is used to be set as trip start location</li> </ol>
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

**Functional Requirements:**

1. The system must fetch user's current location information using GPS

2. The user's current location is used as trip start location during trip planning

## 4.8 Display Trip Information

Use Case ID:	DT1		
Use Case Name:	Display Trip Information		
Created By:	PUSHPARAJAN ROSHINI	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	13/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	Displays all the required details the user needs to know before confirming their trip. Transportation routes will be default ranked based on the shortest travel time.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has input their destination details</li> </ol>
Postconditions:	<p>The system has displayed the following for each mode of transportation (with a filter that user can choose) :</p> <ol style="list-style-type: none"> <li>1. Duration trip takes</li> <li>2. Fare</li> <li>3. Route of trip</li> </ol>
Priority:	Very High
Frequency of Use:	Very High
Flow of Events:	<ol style="list-style-type: none"> <li>1. User inputs destination details.</li> <li>2. Trip information is displayed on the screen to showcase details of each trip.</li> <li>3. User scrolls through the 2 options of transportation- transit and taxi to view the details under each option.</li> <li>4. User can choose to filter their trip options <ol style="list-style-type: none"> <li>a. "Shortest duration" and "Cheapest Route"</li> </ol> </li> </ol>
Alternative Flows:	-
Exceptions:	<p>EX 1 : API fails to retrieve some of the information</p> <ol style="list-style-type: none"> <li>1. An error message is shown to inform User that time/fare/routing services are unavailable</li> </ol>
Includes:	Confirm Trip, View Trip History, Get Duration, Get Route, Get Price
Special Requirements:	Nil
Assumptions:	<p>There can be trip options which include multiple modes of transportations. Hence,combined cost,duration and route will be displayed.</p> <p>All API used works and information retrieval is real time and accurate.</p> <p>User inputs a complete and correct address according to OneMap's routing website's expectations with correct formatting.</p>
Notes and Issues:	Nil

**Functional Requirements:**

1. The system must display all possible trip options available to transport user to their destination place
  - 1.1. Trips shall be ranked and filtered according to user's preference
    - 1.1.1. The system will display possible trip options in different modes of transportation methods.
    - 1.1.2. The system displays filtered trip options according to filter checked by user
      - 1.1.2.1. The system must display the filtering option "Shortest Duration", "Cheapest Route".
  - 1.2. Trips can be displayed as a solo mode of transport or a combination of several modes of transport
  - 1.3. The system must include more information about trip details for each trip option in display.
    - 1.3.1. The system must display total duration time for each trip option
    - 1.3.2. The system must display total fare cost for each trip option
    - 1.3.3. The system must display a detailed route for each trip option.

**4.9 Get Route**

Use Case ID:	GR1		
Use Case Name:	Get Route		
Created By:	PUSHPARAJAN ROSHINI	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	13/02/2024	Date Last Updated:	13/02/2024

Actor:	User, Taxi Availability API, MRT/Bus Arrival Timings API, Map API
Description:	Get Route fetches the correct route from the APIs and stores the route for each trip
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has input their destination details</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system has fetched the routes of each trip option under Display Trip Information</li> </ol>
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system checks for each possible route from User's start point to end point using APIs.</li> <li>2. The system fetches and displays routes for each possible mode of transport.</li> </ol>
Alternative Flows:	AF-S1: If no route is available : <ol style="list-style-type: none"> <li>1. The system shall display the message "There is no available route for this trip. Please try another destination or verify your destination details correctly."</li> <li>2. User will be redirected to input destination details' page.</li> </ol>

Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	API retrieval is accurate and timely done.
Notes and Issues:	Nil

#### Functional Requirements

1. The system must fetch each trip's respective route from APIs.
2. The system must store the routes each in the system and display in the Display Trip Information page.
  - 2.1. The routes must be accessible.

### 4.10 Get Duration

Use Case ID:	GD1		
Use Case Name:	Get Duration		
Created By:	PUSHPARAJAN ROSHINI	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	13/02/2024	Date Last Updated:	13/02/2024

Actor:	User, Taxi Availability API, MRT/Bus Arrival Timings API, Map API
Description:	Get Duration fetches the correct travel time from the APIs and stores the duration for each trip
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has input their destination details</li> </ol>
Postconditions:	The system has fetched the duration/travel time of each trip option under Display Trip Information.
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user inputs destination details,</li> <li>2. The system checks for estimated duration time from User's start point to end point using APIs.</li> <li>3. The system fetches and displays estimated duration for each possible mode of transport.</li> </ol>
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	API retrieval is accurate and timely done.
Notes and Issues:	Nil

#### Functional Requirements

1. The system must fetch each trip's respective estimated duration from APIs.
2. The system must store the travel times for each of the trip options in the system and display in the Display Trip Information page.



## 4.11 Get Price

Use Case ID:	GP1		
Use Case Name:	Get Price		
Created By:	NAFEES ABDUL KADER	Last Updated By:	NAFEES ABDUL KADER
Date Created:	13/02/2024	Date Last Updated:	13/02/2024

Actor:	User, Taxi Fare API, MRT/Bus Fare API
Description:	Get Price fetches price for each transport from the respective APIs and store it for each trip
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has input their destination details</li> </ol>
Postconditions:	<ol style="list-style-type: none"> <li>1. The system has fetched the price of each trip option under Display Trip Information</li> </ol>
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system retrieves the possible routes to destination from “Get Route”.</li> <li>2. The system fetches the estimated prices for each transportation used in each trip from “Get Route”</li> <li>3. The system stores the estimated prices for each transportation used in each trip in the system.</li> <li>4. The system displays the estimated prices for each mode of transport used for each trip</li> </ol>
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	<ol style="list-style-type: none"> <li>1. API retrieval is accurate and timely done.</li> <li>2. There is at least one route from the user’s current location to the destination</li> </ol>
Notes and Issues:	Nil

### Functional Requirements

1. The system must be able to get price from the APIs
  - 1.1. The system fetches the price for each mode of transportation from the respective API.
    - 1.1.1. The system fetches the taxi price from the Taxi Fare API, if the trip involves a taxi
    - 1.1.2. The system fetches the bus price from the MRT/Bus Fare API, if the trip involves a bus.
    - 1.1.3. The system fetches the MRT price from the MRT/Bus Fare API, if the trip involves MRT.
  - 1.2. The system stores prices for each transportation involved in each trip in the system
  - 1.3. The system displays the prices for each transportation involved in each trip in the Display Trip Information page.

## 4.12 Confirm Trip

Use Case ID:	CT1		
Use Case Name:	Confirm Trip		
Created By:	PUSHPARAJAN ROSHINI	Last Updated By:	JORDAN NG TENG KIAT
Date Created:	13/02/2024	Date Last Updated:	13/02/2024

Actor:	User
Description:	After User is aware of possible trip options, they choose their most desired trip by confirming the trip before proceeding ahead.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has inputted their destination details</li> <li>3. System has displayed all possible trip options for user to select from</li> </ol>
Postconditions:	The system updates the trip as “Ongoing” and adds the trip to the Trip History
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. With the list of trip options, the user selects their desired one and clicks “Select” button</li> <li>2. The system will prompt the user for confirmation on the selection of the trip.</li> <li>3. The system will then direct the user to the Trip History</li> <li>4. If the user’s ongoing trip ends, the system must update the ongoing trip as “completed” in the Trip History.</li> </ol>
Alternative Flows:	
Exceptions:	NIL
Includes:	Login
Special Requirements:	NIL
Assumptions:	<ol style="list-style-type: none"> <li>1. User must click on one of the trip options to confirm a trip.</li> <li>2. System is connected to the APIs before and during the trip</li> </ol>
Notes and Issues:	NIL

### Functional Requirements

1. The system must display a “Select” button for the user to click and confirm their trip option.
2. The system must ask User to confirm selection of trip
  - 2.1. The system must display a “Confirm Add Trip” button for User to click to confirm their selection.
  - 2.2. The system will automatically be directed to the Trip History
3. The system must update the user’s confirmed trip
  - 3.1. The system must save the user’s confirmed trip in the database as ongoing.
  - 3.2. The system must update the Trip History with the newly confirmed trip.

## 4.13 View Trip History

Use Case ID:	TH1		
Use Case Name:	View Trip History		
Created By:	JORDAN NG TENG KIAT	Last Updated By:	NAFEES ABDUL KADER
Date Created:	12/02/2024	Date Last Updated:	21/04/2024

Actor:	User
--------	------

Description:	Displays the past trips that the user has saved using “Add Trip”.
Preconditions:	1. The user is logged in. 2. The user has one or more completed trips.
Postconditions:	The system displays the past trips of the user.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The user clicks “View trip history”. 2. The system retrieves the user’s past trips by checking the database. 3. The system displays the user’s trip history. 4. The system displays the user’s monthly expenditure for each transport mode as well as total transport expenditure.
Alternative Flows:	AF-S1: If the user does not have any past trips yet 1. The system displays nothing 2. The system will display the monthly expenditure for each transport, for each month, \$0.
Exceptions:	Nil
Includes:	Edit Trip
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

Functional Requirements:

1. The user must be able to view trip history.
  - 1.1. The system must display the user’s ongoing trip at the top.
    - 1.1.1. The system must display the ongoing trip’s status as “Ongoing”.
  - 1.2. The system must display each past completed trips’ status as “Completed” and display information in reverse chronological order.
    - 1.2.1. The system must display the completed trip’s start location.
    - 1.2.2. The system must display the completed trip’s destination.
    - 1.2.3. The system must display the completed trip’s travel duration.
    - 1.2.4. The system must display the completed trip’s mode of transportation taken.
    - 1.2.5. The system must display the completed trip’s expenditure.
  - 1.3. The system must display the user’s monthly expenditure at the top of the page.
    - 1.3.1. The system must display expenditure for each transport method.
    - 1.3.2. The system must display expenditure for all the transports combined.
    - 1.3.3. The system must provide an option for users to switch between months to view previous months’ history.

**4.14 Edit Trip**

Use Case ID:	ET1		
Use Case Name:	Edit Trip		
Created By:	JORDAN NG TENG KIAT	Last Updated By:	NAFEES ABDUL KADER
Date Created:	13/02/2024	Date Last Updated:	21/04/2024

Actor:	User
--------	------

Description:	User is allowed to edit past completed trips in the Trip History page.
Preconditions:	<ol style="list-style-type: none"> <li>1. The user is logged in.</li> <li>2. The user has one or more completed trips.</li> <li>3. The user is in the Trip History page.</li> </ol>
Postconditions:	The system displays the past trips of the user.
Priority:	High
Frequency of Use:	Low
Flow of Events:	<ol style="list-style-type: none"> <li>1. The system prompts the user with 2 options - “Ongoing” or Completed” to edit status of the trip</li> <li>2. The user clicks on the “Save changes” button on the edited trip.</li> <li>3. The system updates the changes on the edited trip on display and in the database.</li> </ol>
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

**Functional Requirements:**

1. The system must allow the user to edit ongoing/past trips in the Trip History page.
  - 1.1. The system must display an option for editing the status of the trip : “Ongoing” or “Completed”.
    - 1.1.1. The system must display a “Save changes” button.
      - 1.1.1.1. If the user clicks on the “Save changes” button, the system must update the selected trip’s changes on display and in the database.

## 4.15 Manage Wallet Balance

Use Case ID:	WB1		
Use Case Name:	Manage Wallet Balance		
Created By:	NAFEES ABDUL KADER	Last Updated By:	JORDAN NG TENG KIAT
Date Created:	12/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	The user will be able to manage the notifications and the expenditure limit, whether the user wants to activate both of them or not
Preconditions:	1. The user is logged in.
Postconditions:	<ol style="list-style-type: none"> <li>1. If the user wants notifications, the notifications will be switched on</li> <li>2. If the user does not want notifications, the notifications will be switched off.</li> <li>3. If the user wants to set an expenditure limit, the user will activate the expenditure limit option.</li> <li>4. If the user does not want to set an expenditure limit, the use will not activate the expenditure limit option.</li> </ol>
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> <li>1. The user will click the Manage Wallet Balance tab</li> <li>2. If the user wants to set an expenditure limit, he/she will enter their limit in the Set Monthly Funds textbox and select "Update"</li> <li>3. If the user wants to turn on the notifications, the user will check the "Alert When Fund Below 10%" option</li> </ol>
Alternative Flows:	<p>AF-S2: If the user wants to edit their expenditure limit</p> <ol style="list-style-type: none"> <li>1. The user will enter their new limit into the Set Monthly Funds text box and click "update"</li> </ol> <p>AF-S3: If the user wants to enable or disable notifications without editing expenditure limit or 10% threshold</p> <ol style="list-style-type: none"> <li>1. The user will erase the number inside the Set Monthly Funds text box</li> <li>2. The user will uncheck the "Alert When Fund Below 10%" option.</li> </ol>
Exceptions:	Nil
Includes:	<ol style="list-style-type: none"> <li>1. Set Expenditure Limit</li> <li>2. Notify Low Balance</li> </ol>
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

**Functional Requirements:**

1. Users shall be able to manage their wallet balance.
  - 1.1. Users can access and click on the “Manage Wallet Balance” option found in the settings.
  - 1.2. The system will also display a text box labelled “Monthly Funds”
    - 1.2.1, The user can enter their expenditure limit and click on “update”
      - 1.2.1.1. The user can edit their expenditure limit by entering their new limit and clicking on “update”
  - 1.3. The system will display a “Alert When Fund Below 10%” option under “Manage Wallet Balance”.
    - 1.3.1. If the user wants to on the notifications, the user will check the “Alert When Fund Below 10%” option
    - 1.3.2. If the user wants to turn off the notifications, the user will uncheck the “Alert When Fund Below 10%” option.

**4.16 Set Expenditure Limit**

Use Case ID:	EL1		
Use Case Name:	Set Expenditure Limit		
Created By:	NAFEES ABDUL KADER	Last Updated By:	JORDAN NG TENG KIAT
Date Created:	12/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	This user case is used by the user to indicate how much money willing to spend money on transportation for a particular period of time (weekly/monthly)
Preconditions:	The user is logged in.
Postconditions:	The system stores the expenditure limit that the user has keyed in.
Priority:	Medium
Frequency of Use:	Medium, only when users intend on having an expenditure limit to have a restriction in how much they spend for transportation
Flow of Events:	<ol style="list-style-type: none"> <li>1. Users will click on the “Manage Wallet Balancet” button</li> <li>2. Users will then be redirected to a page with an empty field.</li> <li>3. Users can then key in their intended expenditure limit in the empty field, in numbers.</li> <li>4. The number keyed in can be a whole number. Eg. - 123 - 14578</li> <li>5. Users can only key in numbers up to 23 digits.</li> <li>6. Once the limit is keyed in, users will then press a “Update” button</li> <li>7. The system will update the database and show the newly set amount in the green box in the top right corner.</li> </ol>

	8. If the user wants to indicate a new expenditure limit, he/she can repeat steps 1 to 6.
Alternative Flows:	<p>AF-S3: If the user deletes the placeholder “100” in the textbox</p> <ol style="list-style-type: none"> <li>1. The system will indicate in “NaN”, to indicate that no amount is set</li> <li>2. The “Update” button will not affect the current expenditure limit as well as the 10% low balance threshold.</li> <li>3. The “Update” button will save the settings of the “Notify Low Balance” feature.</li> </ol> <p>AF-S5: If the user starts to key in numbers with more than 23 digits</p> <ol style="list-style-type: none"> <li>1. The system will disable the user from entering more numbers exceeding 23 digits.</li> </ol>
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	Nil
Notes and Issues:	Nil

### Functional Requirements:

1. Users shall be able to set an expenditure limit, by clicking on a “Manage Wallet Balance” option.
  - 1.1. The system will display an empty field to key in the expenditure limit.
    - 1.1.1. Users can only key in numbers in the empty field.
    - 1.1.2. The field only accepts up to 23 digits.
      - 1.1.2.1. If the user keys in number more than 23 digits, the system prevents user from keying more numbers.
    - 1.1.3. The amount keyed in could be in whole number.
    - 1.1.4. Space bar will be disabled.
    - 1.1.5. The field does not accept alphabets or characters.
      - 1.1.6.1. If an alphabet or character is keyed in, the system will indicate in “NaN” on the empty field, indicating that no value is set.
      - 1.1.6.2. If an alphabet or character is keyed in, the system will prevent non-integers from being entered.
      - 1.1.6.3. If the user clicks the “Update” button without setting any value in the “Set Monthly Funds” text box, the current expenditure limit and 10% low balance threshold will not be affected.
      - 1.1.6.4. If the user clicks the “Update” button without setting any value in the “Set Monthly Funds” text box, the “Alert When Fund Below 10%” setting will be saved in the database.
  - 1.2. The system will display a “Update” button.
    - 1.2.1. Once the user has keyed in the expenditure limit, the user can press “Confirm” to save the data.



#### 4.17 Notify Low Balance

Use Case ID:	RN1		
Use Case Name:	Notify Low Balance		
Created By:	NAFEES ABDUL KADER	Last Updated By:	JORDAN NG TENG KIAT
Date Created:	12/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	Displays on top of page when the amount of money stated is low, thereby advising the user to not spend too much on transport
Preconditions:	1. The user is logged in.
Postconditions:	The system highlights the balance red..
Priority:	High
Frequency of Use:	High
Flow of Events:	1. If the user has spent at least 90% of its expenditure limit, the system will highlight the balance in red, shown in the top right corner.
Alternative Flows:	-
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	-
Notes and Issues:	Nil

##### Functional Requirements:

1. Users can receive notifications when notifications are turned on.
  - 1.1. If the user has spent at least 90% of its expenditure limit, the system will highlight the balance in red, shown in the top right corner.
  - 1.2. The system will display the notifications on the top of the webpage

#### 4.18 Logout

Use Case ID:	LOG2		
Use Case Name:	Logout		
Created By:	NAFEES ABDUL KADER	Last Updated By:	JORDAN NG TENG KIAT
Date Created:	27/02/2024	Date Last Updated:	21/04/2024

Actor:	User
Description:	Allows the user to log out of their account after usage.
Preconditions:	1. The user is logged in
Postconditions:	1. The user is automatically redirected to the login page after 5 seconds

Priority:	High
Frequency of Use:	After every usage
Flow of Events:	<ol style="list-style-type: none"> <li>1. Users will click on the logout button on the top right hand of the home page under “Welcome, User”</li> <li>2. The system will indicate it has successfully logged out</li> <li>3. The system will have a 5-second countdown before redirecting to the login page.</li> <li>4. The system will direct the user to the login page.</li> </ol>
Alternative Flows:	AF-S3: If the user clicks on the Login button in the top right corner <ol style="list-style-type: none"> <li>1. The user will click on the Login button in the top right corner to skip the 5-second countdown and proceed to the Login page immediately.</li> </ol>
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	The user is at the home page of the website
Notes and Issues:	Nil

#### Functional Requirements:

1. The user must be able to logout.
  - 1.1. The system displays a “Logout” button at the top right section of the webpage.
  - 1.2. Users shall click on that “Logout” button.
  - 1.3. The system will successfully logout from the account
    - 1.3.1. While logging out, it will have a 5-second countdown.
    - 1.3.2. If the user clicks on the “Log in” button in the top right corner
      - 1.3.2.1. The system will immediately redirect the user to the Login page, skipping the 5-second countdown.
  - 1.4. The user will be directed to the login page.

## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

- 5.1.1 Users should receive route suggestions within 5 seconds of entering their search criteria.
- 5.1.2 The website should be able to handle high traffic volumes without performance degradation.
  - 5.1.2.1 The system should be able to handle a concurrent user load of at least 1000 users.
- 5.1.3 Route data, including prices and travel times, should be updated regularly to ensure accuracy.
  - 5.1.3.1 The estimated cost, duration calculation should not deviate by more than 10 percent from actual units for at least 95% of cases
- 3.1.4 The geolocation accuracy should be within a radius of 50m for 95% of user's requests.
  - 3.1.4.1 The geolocation data should be updated at least every 10s for real-time user tracking

### 5.2 Safety Requirements

- 5.2.1 Routes suggested to the users should not involve illegal or unsafe actions that may endanger them.

### 5.3 Security Requirements

- 5.3.1 Since any data keyed into the website is personal, privacy has to be maintained. Hence, the user's password should be encrypted.
- 5.3.2 User data, including search history and preferences, and financial information should be protected with encryption and secure storage practices.

### 5.4 Software Quality Attributes

- 5.4.1 **Usability:** The website should be user-friendly, intuitive, and easy to navigate, even for first-time users.
- 5.4.2 **Reliability:** The website should be highly available and minimise downtime to ensure consistent user experience.
- 5.4.3 **Testability:** The code should be well-structured and documented, facilitating efficient testing and bug fixing.
- 5.4.4 **Maintainability:** The code should be well-documented, modular, and easy to understand for future maintenance and updates.

## 6. Testing

### 6.1 Black Box Testing

#### 6.1.1 Registration

ID	Input	Expected	Actual
1	Register with a <b>valid</b> email address and password.	The system redirects users to the Home page.	The system redirects users to the Home page.
2	Register with the first name, last name, email address, password and/or confirm password field(s) <b>not populated</b> .	The system displays a message: "Please fill out this field." on unpopulated field(s).	The system displays a message: "Please fill out this field." on unpopulated field(s).
3	Register with an <b>invalid</b> email address format (no '@').	The system displays a message: "Please include an '@' in the email address. '...' is missing an '@'."	The system displays a message: "Please include an '@' in the email address. '...' is missing an '@'."
4	Register with an <b>invalid</b> email address format.(eg. without .com)	The system displays a message: "Invalid email. Please enter a valid email address."	The system displays a message: "Invalid email. Please enter a valid email address."
5	Register with an <b>existing</b> email address.	The system displays a message: "Email already exists. Please log in or use a different email address."	The system displays a message: "Email already exists. Please log in or use a different email address."
6	Register with a password <b>not meeting requirements</b> .	The system displays a message:	The system displays a message:

		“Password must be at least 6 characters long and include at least one special character.”	“Password must be at least 6 characters long and include at least one special character.”
7	Register with “password” and “confirm password” fields having <b>different inputs</b> .	The system displays a message: “Passwords do not match.”	The system displays a message: “Passwords do not match.”

Specific Cases					
Email Address					
1	johndoe@gmail.com	Approved	Approved		
2	johndoe123@gmail.com	Approved	Approved		
3	johndoe123@gmail	Rejected	Rejected		
4	johndoe	Rejected	Rejected		
Password					
1	P@ssword123	Approved	Approved		
3	P@ssword!sVeryStrong123	Approved	Approved		
2	password	Rejected	Rejected		
Combined					
	Email Address	Password	Confirm Password		
1	johndoe123@gmail.com	P@assword123	P@assword123	The system redirects users to the Home page.	The system redirects users to the Home page.
2	<b>johndoe123@gmail</b>	P@assword123	P@assword123	The system displays a message: “Invalid	The system displays a message: “Invalid

				email. Please enter a valid email address.”	email. Please enter a valid email address.”
3	johndoe123@gmail.com	<b>Password</b>	<b>Password</b>	The system displays a message: “Password must be at least 6 characters long and include at least one special character.”	The system displays a message: “Password must be at least 6 characters long and include at least one special character.”
4	<b>RegisteredEmail@gmail.com</b>	P@assword123	P@assword123	The system displays a message: “Email already exists. Please log in or use a different email address.”	The system displays a message: “Email already exists. Please log in or use a different email address.”
5	johndoe123@gmail.com	P@assword123	<b>password</b>	The system displays a message: “Passwords do not match.”	The system displays a message: “Passwords do not match.”
6	johndoe@gmail.com	P@assword123	<b>{Empty}</b>	The system displays a message: “Please fill out this field.” on unpopulated field(s).	The system displays a message: “Please fill out this field.” on unpopulated field(s).
7	<b>{Empty}</b>	P@assword123	P@assword123	The system displays a message: “Please fill out this field.” on unpopulated field(s).	The system displays a message: “Please fill out this field.” on unpopulated field(s).

## 6.1.2 Login

ID	Input	Expected	Actual
----	-------	----------	--------

1	Login with a <b>registered</b> email address and corresponding password.	The system redirects users to the Home page.	The system redirects users to the Home page.
2	Login with a registered email address but <b>incorrect</b> password.	The system displays a message: "Incorrect email or password."	The system displays a message: "Incorrect email or password."
3	Login with an <b>unregistered</b> email address.	The system displays a message: "Incorrect email or password."	The system displays a message: "Incorrect email or password."
4	Login with the email address field <b>not populated</b> .	The system displays a message: "Please fill out this field."	The system displays a message: "Please fill out this field."
5	Login with the password field <b>not populated</b> .	The system displays a message: "Please enter password."	The system displays a message: "Please enter password."

Specific Cases				
Combined				
	Email Address	Password		
1	johndoe123@gmail.com	P@assword123	The system redirects users to the Home page.	The system redirects users to the Home page.
2	johndoe123@gmail.com	<b>wrongPassword</b>	The system displays a message: "Incorrect email or password."	The system displays a message: "Incorrect email or password."
3	<b>unregisteredEmail@gmail.com</b>	P@assword123	The system displays a message:	The system displays a message:

			“Incorrect email or password.”	“Incorrect email or password.”
4	{Empty}	P@assword123	The system displays a message: “Please fill out this field.”	The system displays a message: “Please fill out this field.”
5	johndoe123@gmail.com	{Empty}	The system displays a message: “Please enter password.”	The system displays a message: “Please enter password.”

### 6.1.3 Forgot Password

ID	Input	Expected	Actual
1	Input an <b>email address</b> to trigger password reset.	The system displays a message: “Please check your inbox for the instructions to reset your password.”	The system displays a message: “Please check your inbox for the instructions to reset your password.”
2	Input an email address with <b>invalid format</b> to trigger password reset.	The system displays a message: “Please include an ‘@’ in the email address. ‘..’ is missing an ‘@’.”	The system displays a message: “Please include an ‘@’ in the email address. ‘..’ is missing an ‘@’.”
3	Trigger password reset with the email address field <b>not populated</b> .	The system displays a message: “Please fill out this field.”	The system displays a message: “Please fill out this field.”

### 6.1.4 Edit Profile

ID	Input	Expected	Actual
1	Update with <b>valid</b> current password, new password and confirm password.	The system redirects users to	The system redirects users to



		the Trip History page.	the Trip History page.
2	Update with valid current password and <u>new email address</u> .	The system displays a message “Please check your new email inbox to verify the new email.”	The system displays a message “Please check your new email inbox to verify the new email.”
3	Update with current password field <u>not populated</u> .	The system displays a message: “Please fill out this field.”	The system displays a message: “Please fill out this field.”
4	Update with <u>incorrect current password</u> .	The system displays a message “Current password invalid.”	The system displays a message “Current password invalid.”
5	Update with a password <u>not meeting requirements</u> .	The system displays a message: “Password must be at least 6 characters long and include at least one special character.”	The system displays a message: “Password must be at least 6 characters long and include at least one special character.”
6	Update with “password” and “confirm password” fields having <u>different inputs</u> .	The system displays a message: “Passwords do not match.”	The system displays a message: “Passwords do not match.”

Specific Cases						
Combined						
	New Email	Current Password	New Password	Confirm Password		

1	johndoe123@gmail.com	P@assword123	p4ssw0rd!	p4ssw0rd!	The system redirects users to the Trip History page.	The system redirects users to the Trip History page.
2	johndoe@gmail.com	P@assword123	{Empty}	{Empty}	The system displays a message “Please check your new email inbox to verify the new email.”	The system displays a message “Please check your new email inbox to verify the new email.”
3	johndoe123@gmail.com	{Empty}	p4ssw0rd!	p4ssw0rd!	The system displays a message: “Please fill out this field.”	The system displays a message: “Please fill out this field.”
4	johndoe123@gmail.com	P@assword123	password	password	The system displays a message: “Password must be at least 6 characters long and include at least one special character.”	The system displays a message: “Password must be at least 6 characters long and include at least one special character.”
5	johndoe123@gmail.com	P@assword123	p4ssw0rd!123	p4ssw0rd!	The system displays a message: “Passwords do not match.”	The system displays a message: “Passwords do not match.”

#### 6.1.5 Input Destination Details

ID	Input	Expected	Actual
1	Input a <b>valid</b> location address.	The system displays route options for both transit and taxi.	The system displays route options for both transit and taxi.
2	Input an <b>invalid</b> location address.	The system will not display any	The system will not display any

		information.	information.
3	Proceed with the address field <b>unpopulated</b> .	The system will not display any information.	The system will not display any information.

Specific Cases			
Location Address			
1	50 Nanyang Avenue	Approved	Approved
2	639798	Approved	Approved
3	{Empty}	Rejected	Rejected

#### 6.1.6 Edit Trip

ID	Input	Expected	Actual
1	Change the status of the trip from <b>“Ongoing”</b> to <b>“Completed”</b> , then click <b>“Save”</b> .	The status is changed to <b>“Completed”</b>	The status is changed to <b>“Completed”</b>
2	Change the status of the trip from <b>“Completed”</b> to <b>“Ongoing”</b> , then click <b>“Save”</b> .	The status is changed to <b>“Ongoing”</b>	The status is changed to <b>“Ongoing”</b>

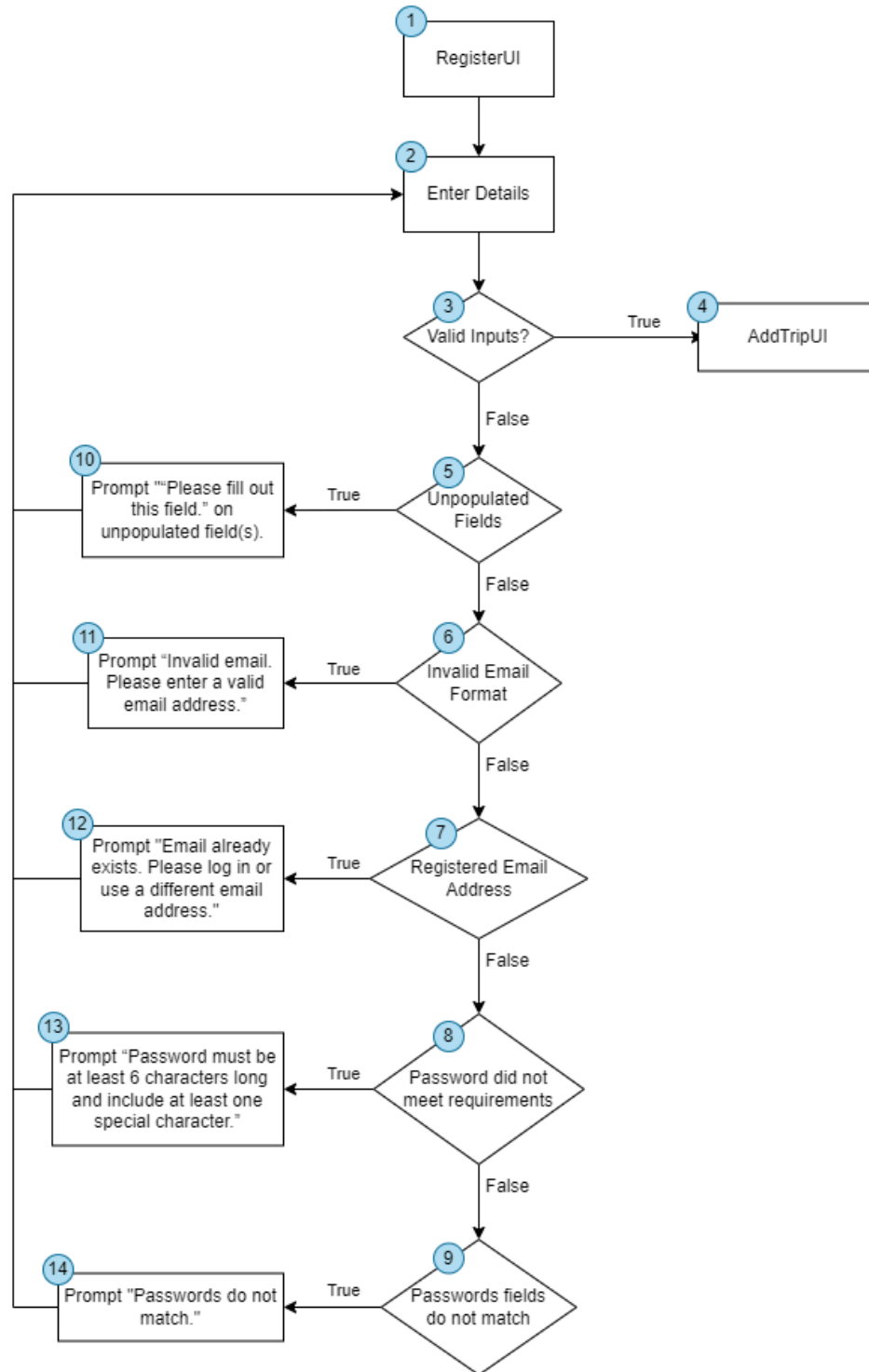
#### 6.1.7 Manage Wallet

ID	Input	Expected	Actual
1	Input an alphabet or a character in the empty field. Eg. input “abc” or “@#\$”	The empty field will display <b>“\$NaN”</b>	The empty field will display <b>“\$NaN”</b>
2	Input a number more than 23 digits in the empty field. Eg. 32453135675324678964321346	The system will stop accepting more numbers after 23 digits.	The system will stop accepting more numbers after 23 digits
3	Input a <b>valid</b> number that contains less than 23 digits and has no alphabets or special characters, then click	The system will update the Wallet	The system will update the Wallet

	"Update".	Balance and display it on the top right hand corner	Balance and display it on the top right hand corner
--	-----------	---	---

## 6.2 White Box Testing

### 6.2.1 Registration

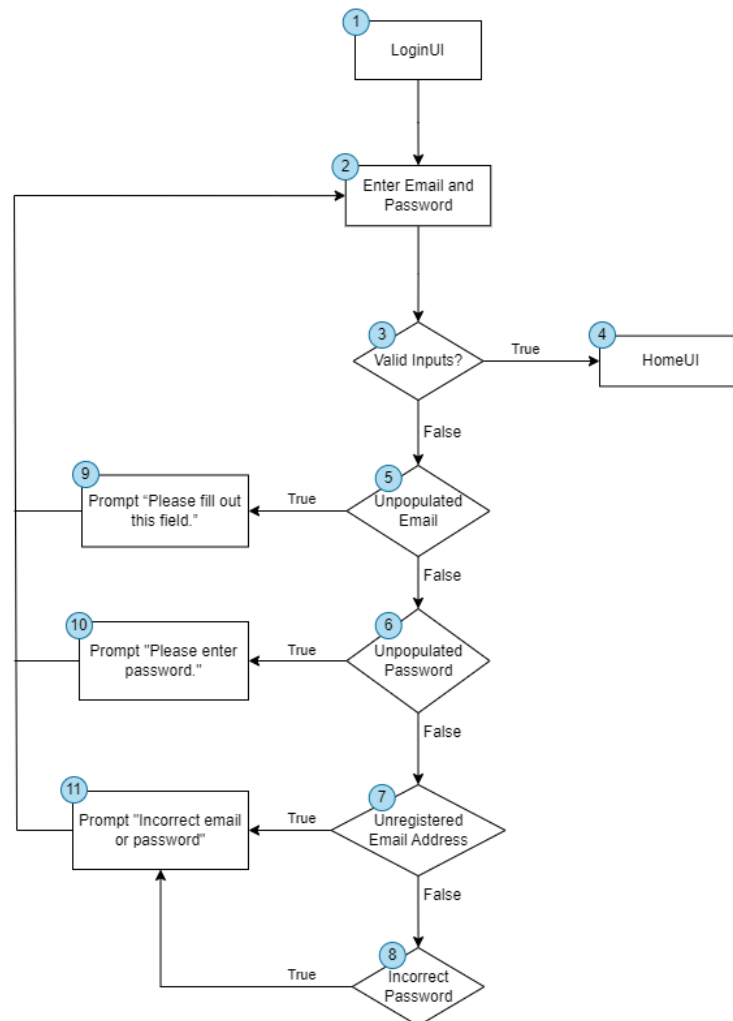


Number of edges = 18, Number of nodes = 14

Cyclomatic Complexity =  $18 - 14 + 2 = 6$

No.	Basis Path	Actual Execution Path
1	1, 2, 3, 4	1, 2, 3, 4
2	1, 2, 3, 5, 10, 2, 3, 4	1, 2, 3, 5, 10, 2, 3, 4
3	1, 2, 3, 5, 6, 11, 2, 3, 4	1, 2, 3, 5, 6, 11, 2, 3, 4
4	1, 2, 3, 5, 6, 7, 12, 2, 3, 4	1, 2, 3, 5, 6, 7, 12, 2, 3, 4
5	1, 2, 3, 5, 6, 7, 8, 13, 2, 3, 4	1, 2, 3, 5, 6, 7, 8, 13, 2, 3, 4
6	1, 2, 3, 5, 6, 7, 8, 9, 14, 2, 3, 4	1, 2, 3, 5, 6, 7, 8, 9, 14, 2, 3, 4

### 6.2.2 Login

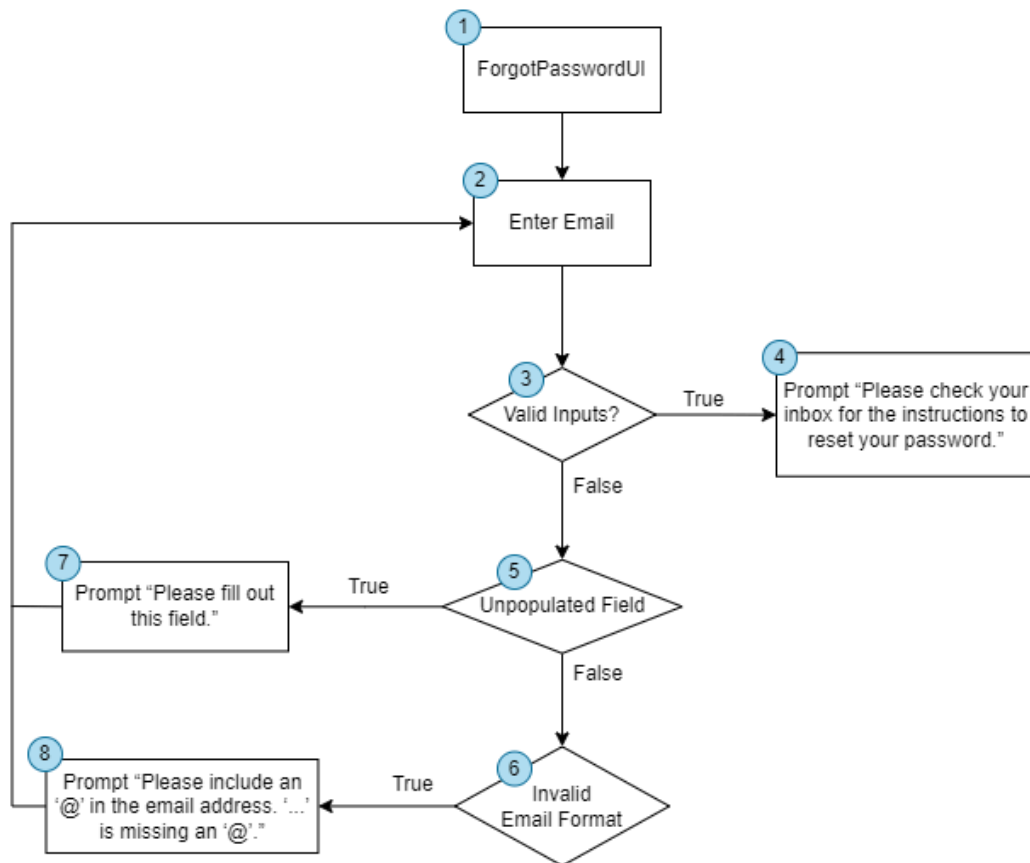


Number of edges = 14, Number of nodes = 11

Cyclomatic Complexity =  $14 - 11 + 2 = 5$

No.	Basis Path	Actual Execution Path
1	1, 2, 3, 4	1, 2, 3, 4
2	1, 2, 3, 5, 9, 2, 3, 4	1, 2, 3, 5, 9, 2, 3, 4
3	1, 2, 3, 5, 6, 10, 2, 3, 4	1, 2, 3, 5, 6, 10, 2, 3, 4
4	1, 2, 3, 5, 6, 7, 11, 2, 3, 4	1, 2, 3, 5, 6, 7, 11, 2, 3, 4
5	1, 2, 3, 5, 6, 7, 8, 11, 2, 3, 4	1, 2, 3, 5, 6, 7, 8, 11, 2, 3, 4

### 6.2.3 Forgot Password



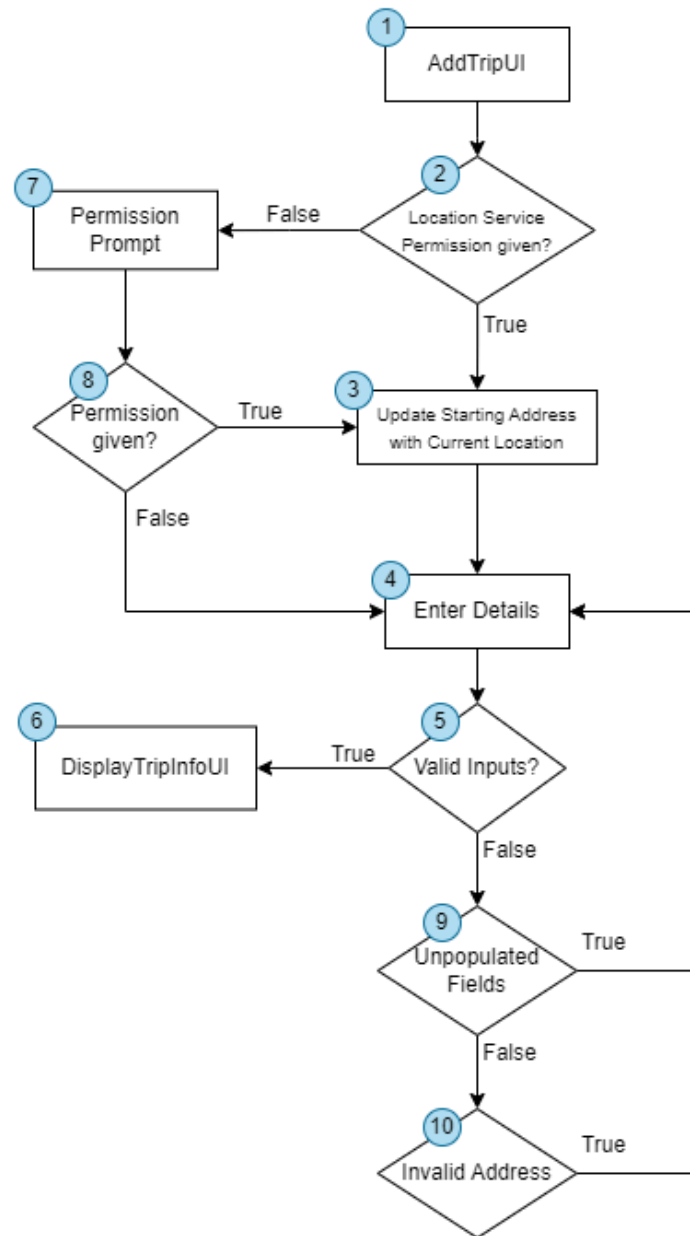
Number of edges = 9, Number of nodes = 8

Cyclomatic Complexity =  $9 - 8 + 2 = 3$

No.	Basis Path	Actual Execution Path
1	1, 2, 3, 4	1, 2, 3, 4
2	1, 2, 3, 5, 7, 2, 3, 4	1, 2, 3, 5, 7, 2, 3, 4

3	1, 2, 3, 5, 6, 8, 2, 3, 4	1, 2, 3, 5, 6, 8, 2, 3, 4
---	---------------------------	---------------------------

## 6.2.4 Get Route



Number of edges = 13, Number of nodes = 10

Cyclomatic Complexity =  $13 - 10 + 2 = 5$

No.	Basis Path	Actual Execution Path
1	1, 2, 3, 4, 5, 6	1, 2, 3, 4, 5, 6
2	1, 2, 7, 8, 3, 4, 5, 6	1, 2, 7, 8, 3, 4, 5, 6



3	1, 2, 7, 8, 4, 5, 6	1, 2, 7, 8, 4, 5, 6
4	1, 2, 3, 4, 5, 9, 4, 5, 6	1, 2, 3, 4, 5, 9, 4, 5, 6
5	1, 2, 3, 4, 5, 9, 10, 4, 5, 6	1, 2, 3, 4, 5, 9, 10, 4, 5, 6

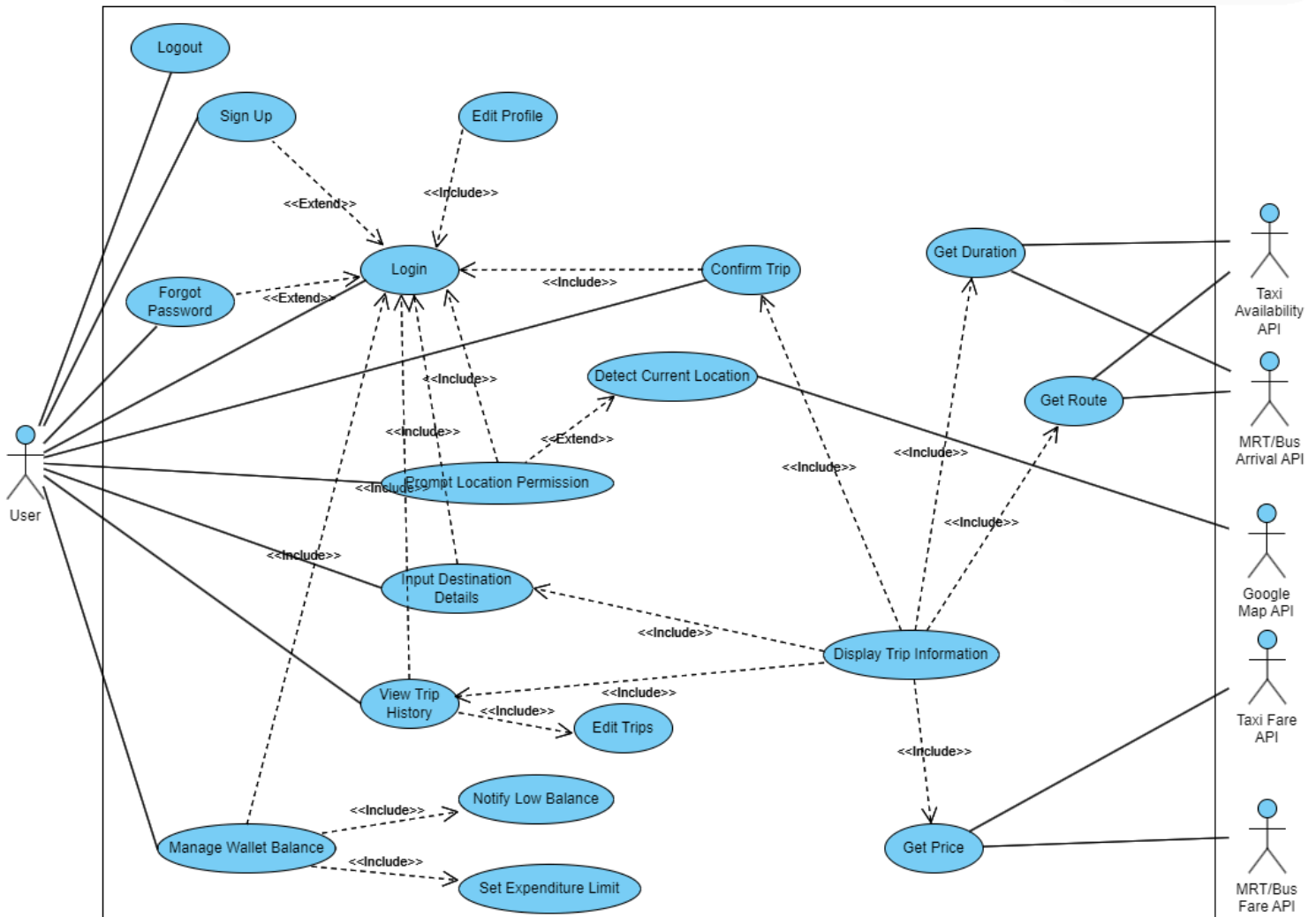
## 7. Appendix A: Data Dictionary

Term	Definition
Origin	The place where the user wishes to starts his journey
Destination	The place where the user intends to go
Transportation Mode	The different methods a user can utilise to get around the country. There are 3 modes of transport: MRT, BUS, TAXI.
Trip	A route connecting the origin from a destination. It can consist of the 3 different transportation modes. Routes will be shown to the user based on their preferences.
Cost	The total estimated cost incurred if the user follows the particular route.
Travel Time	The estimated time taken for the user to get from origin to destination if they follow the particular route.
Edit Trip	Users are given a choice if they wish to edit their trip status once they make changes they can save changes and complete their trip.
Manage Wallet Balance	Users wil key in a expenditure limit they want to spend on transport and this wallet balance depletes as they take on more trips. Once threshold is hit, notifications will be activated to inform user to spend money on transport carefully.

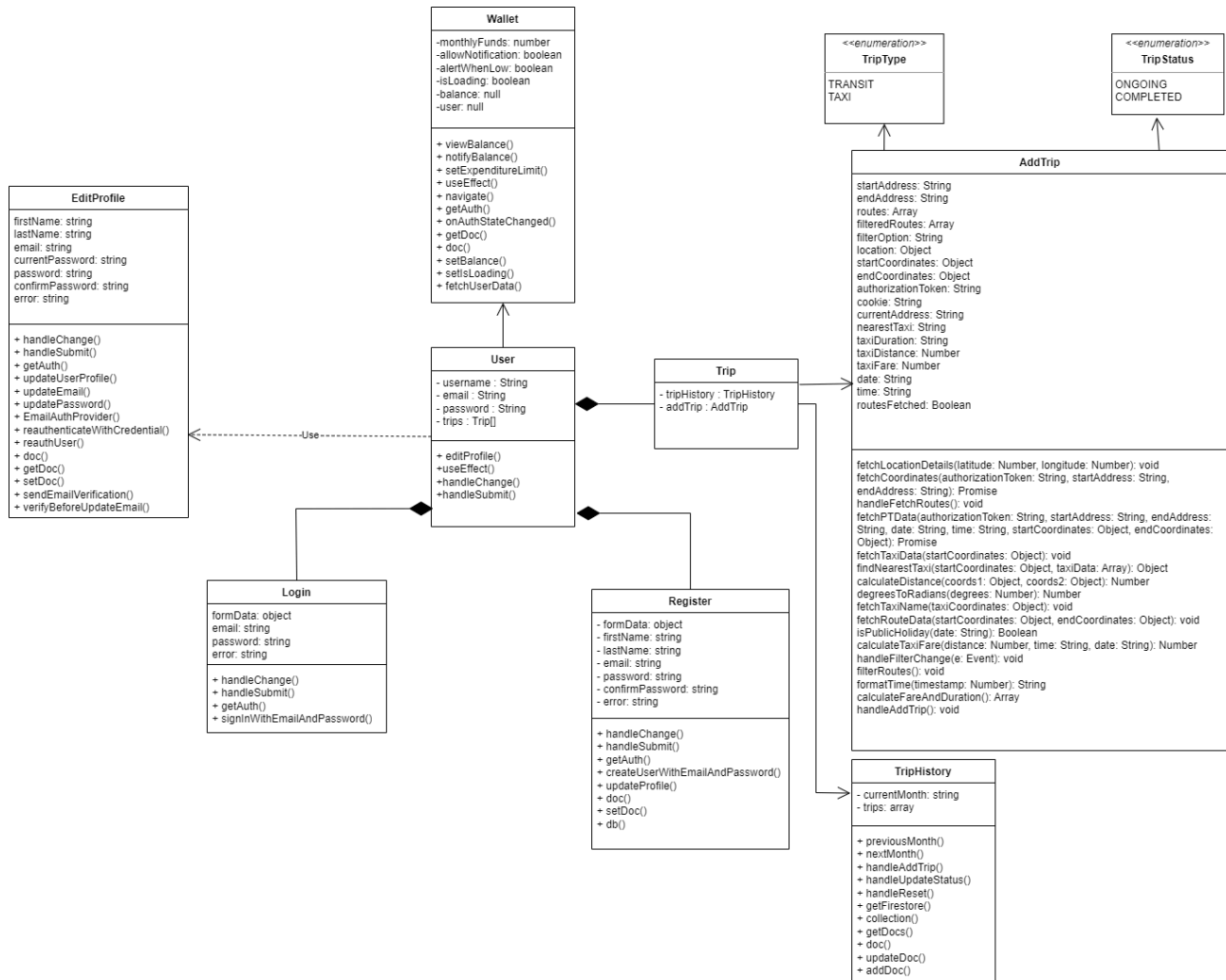
## 8. Appendix B: Analysis Models

Refer to the individual diagrams in the repository for a clearer view.

### Use Case Diagram

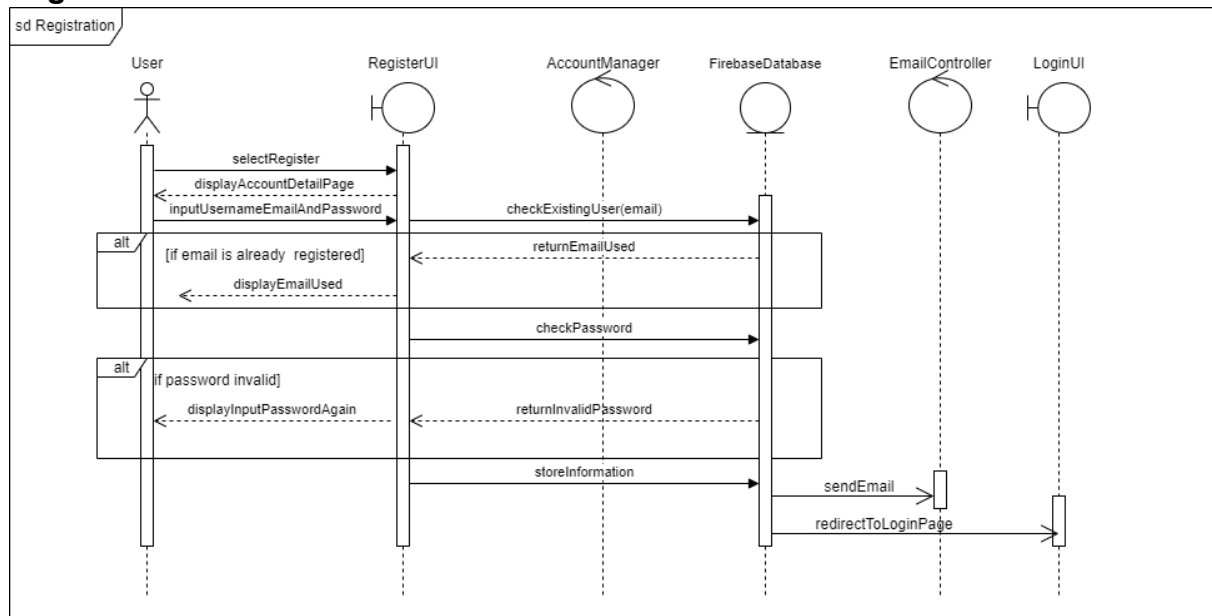


## Class Diagram

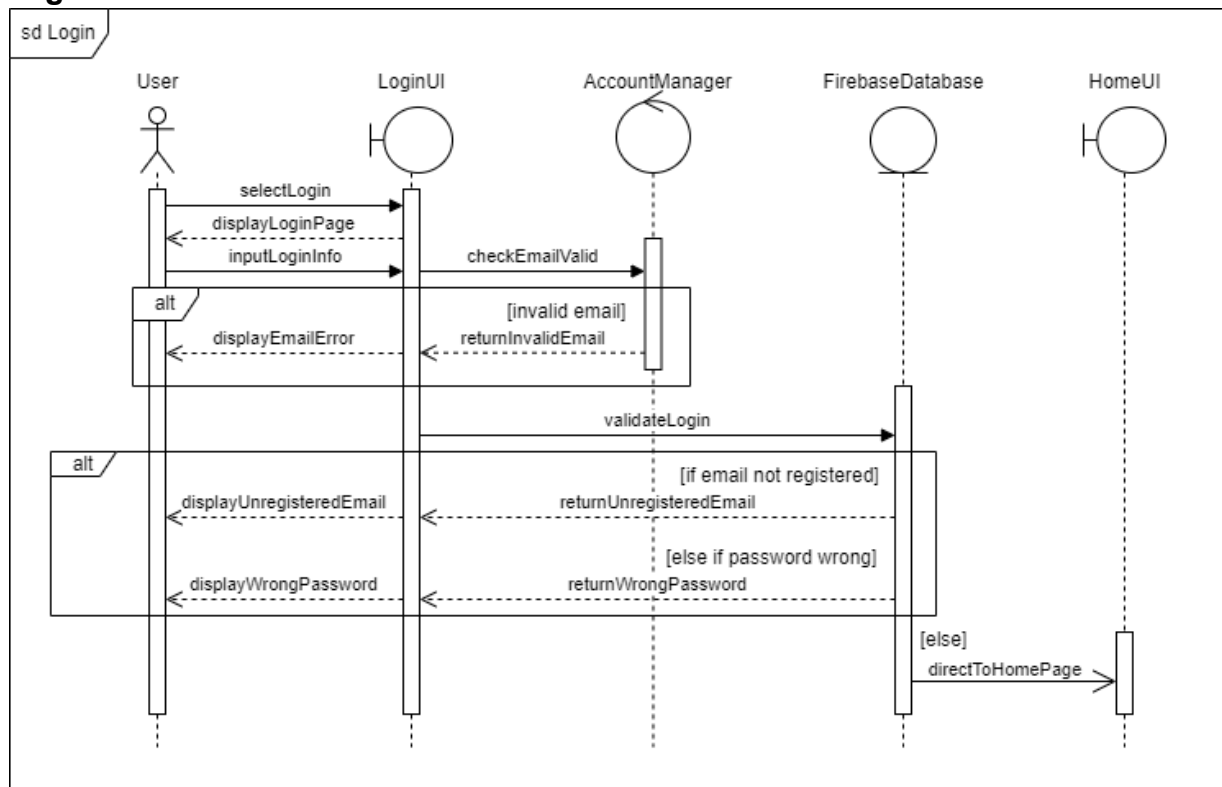


## Sequence Diagram

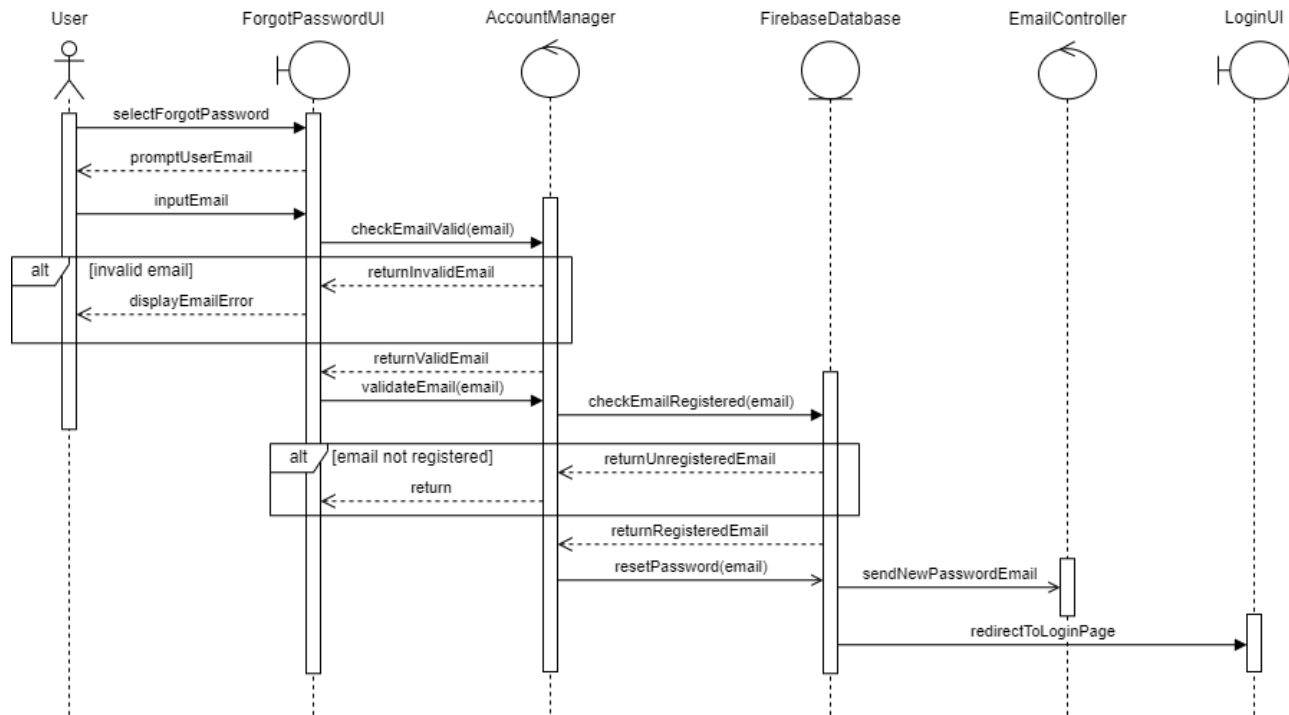
### Registration



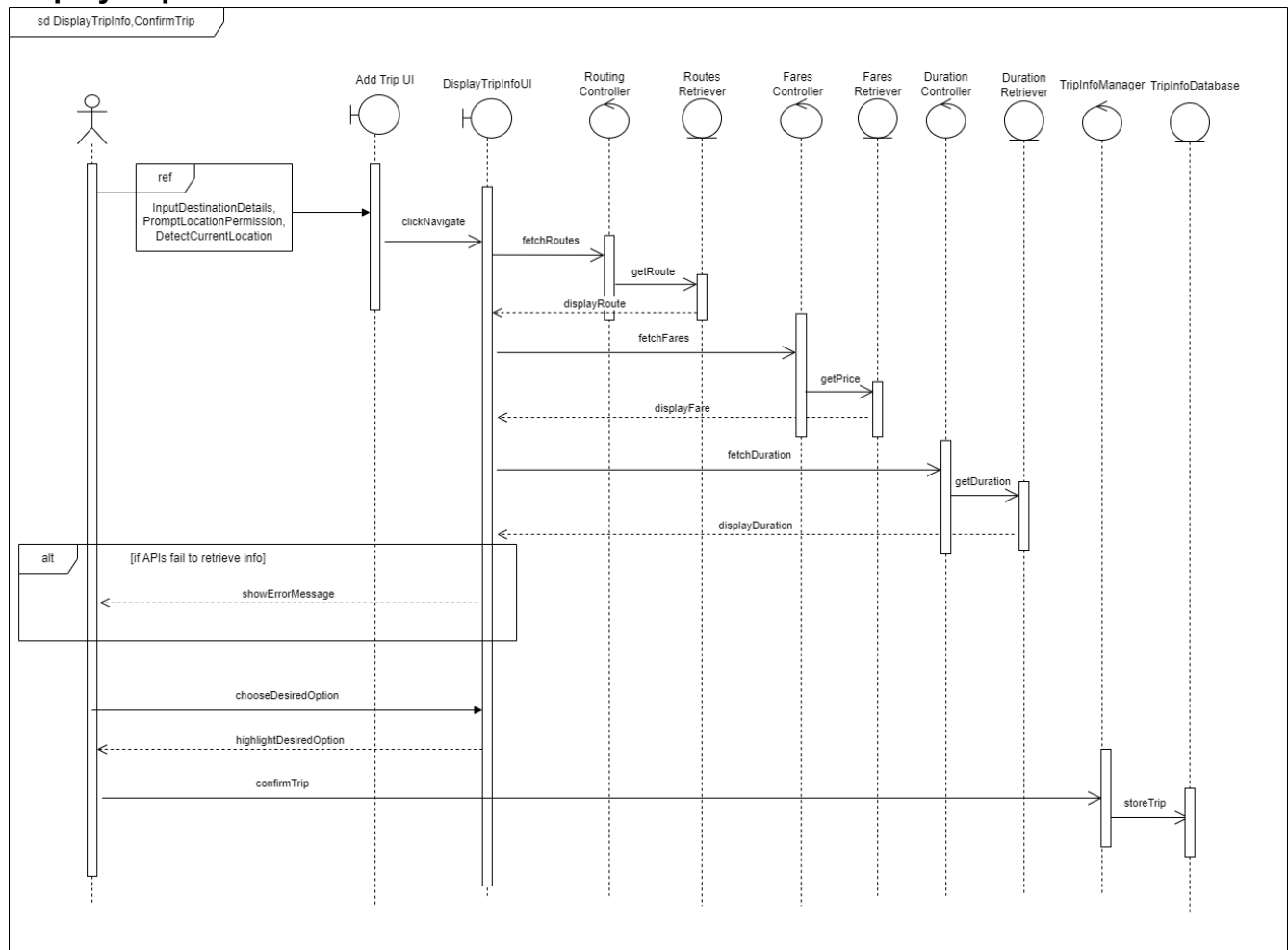
### Login



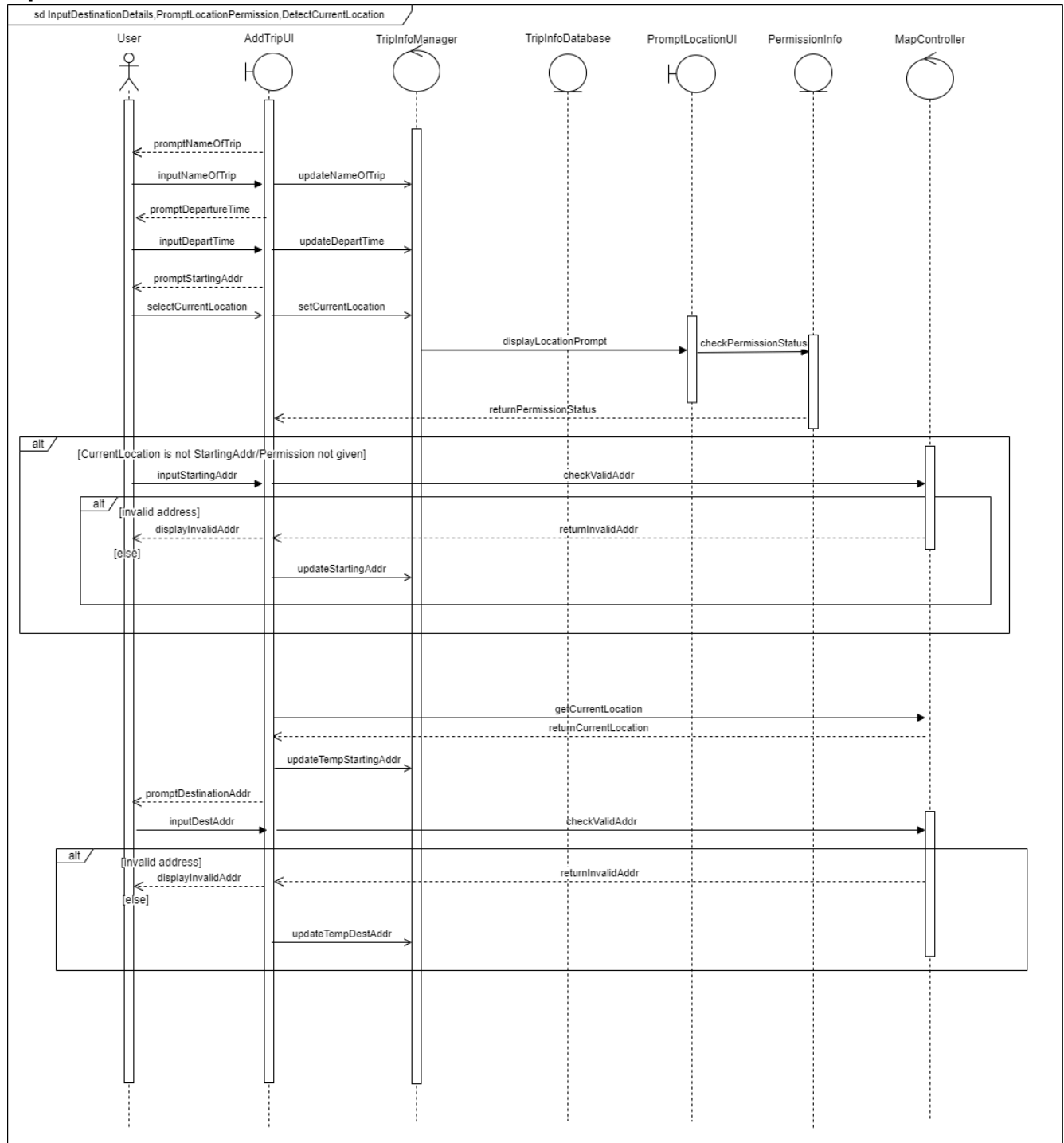
## Forget Password



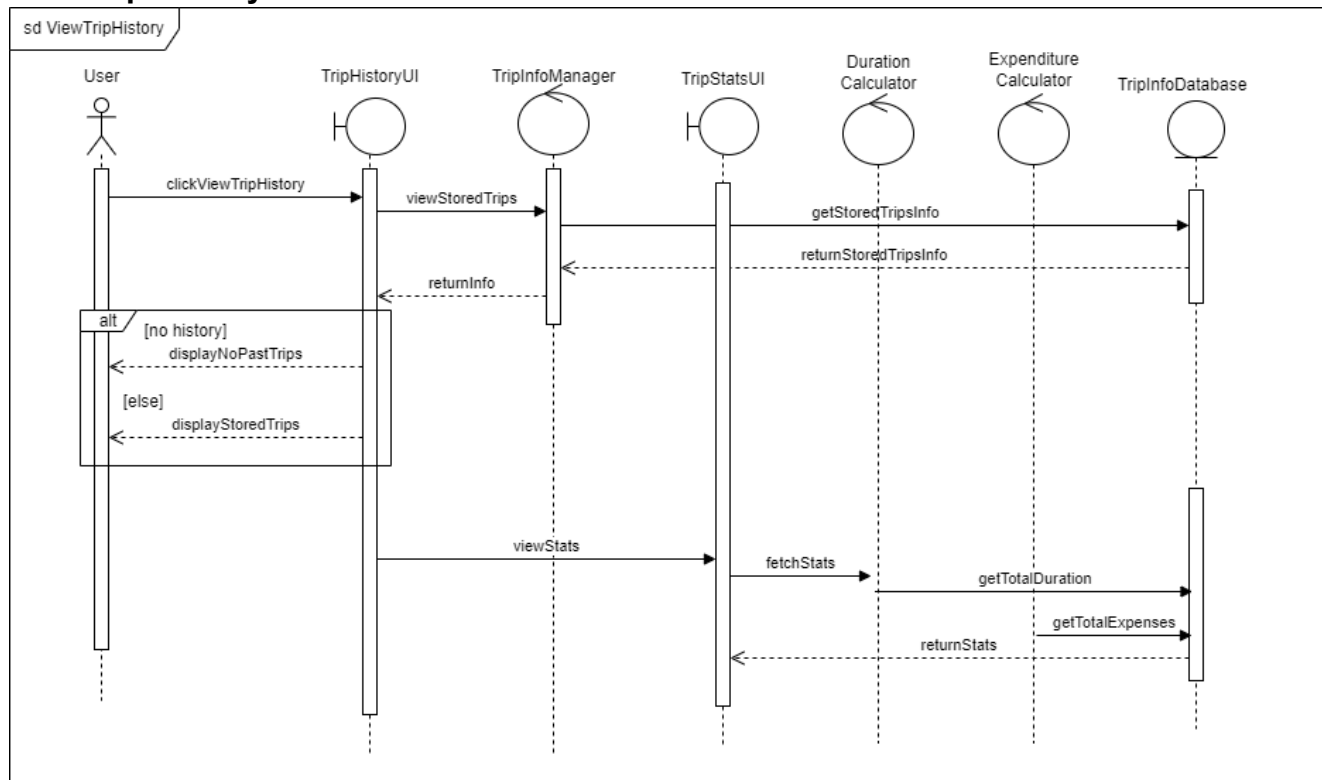
## Display Trip Info



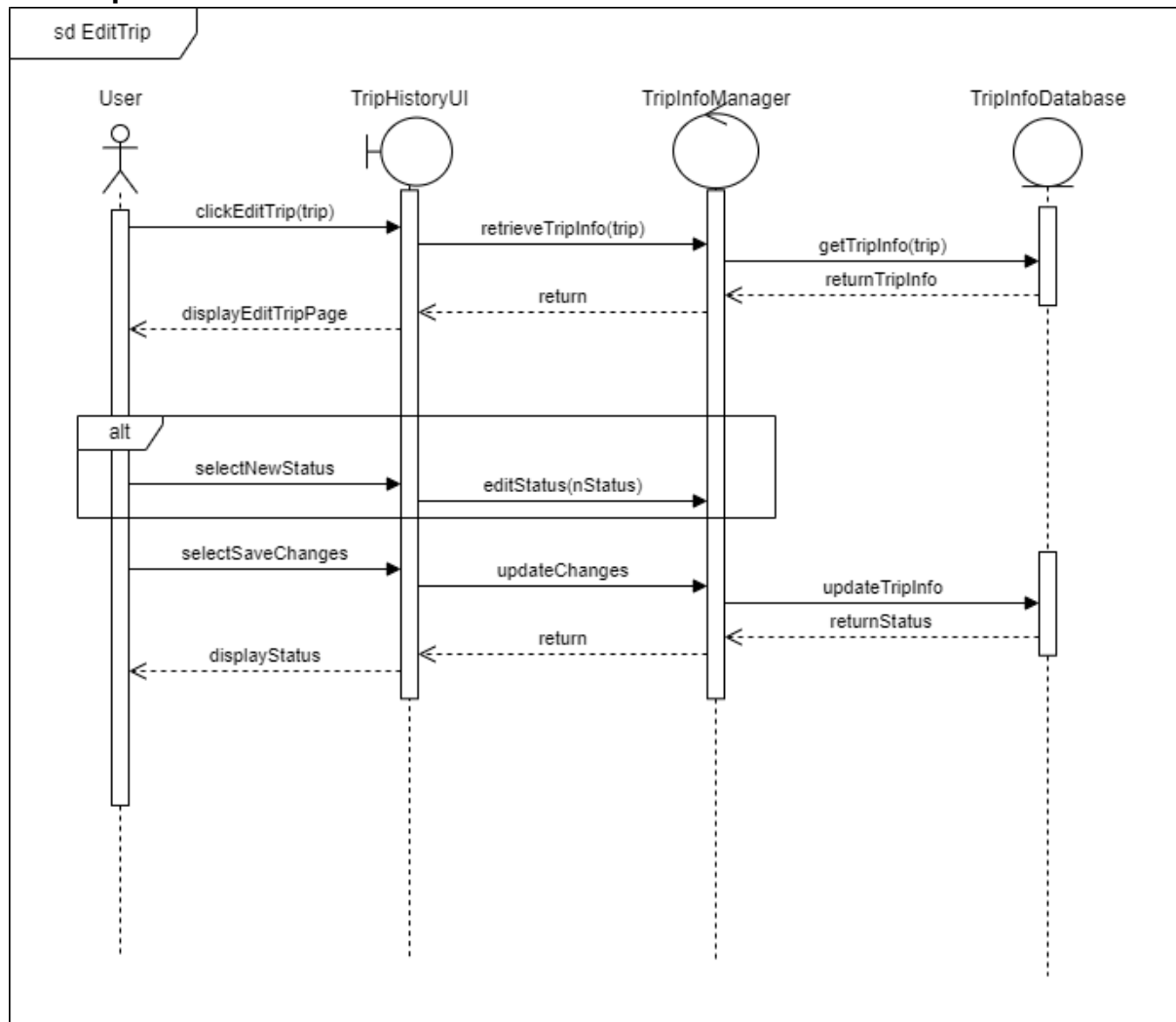
## Input Destination Details



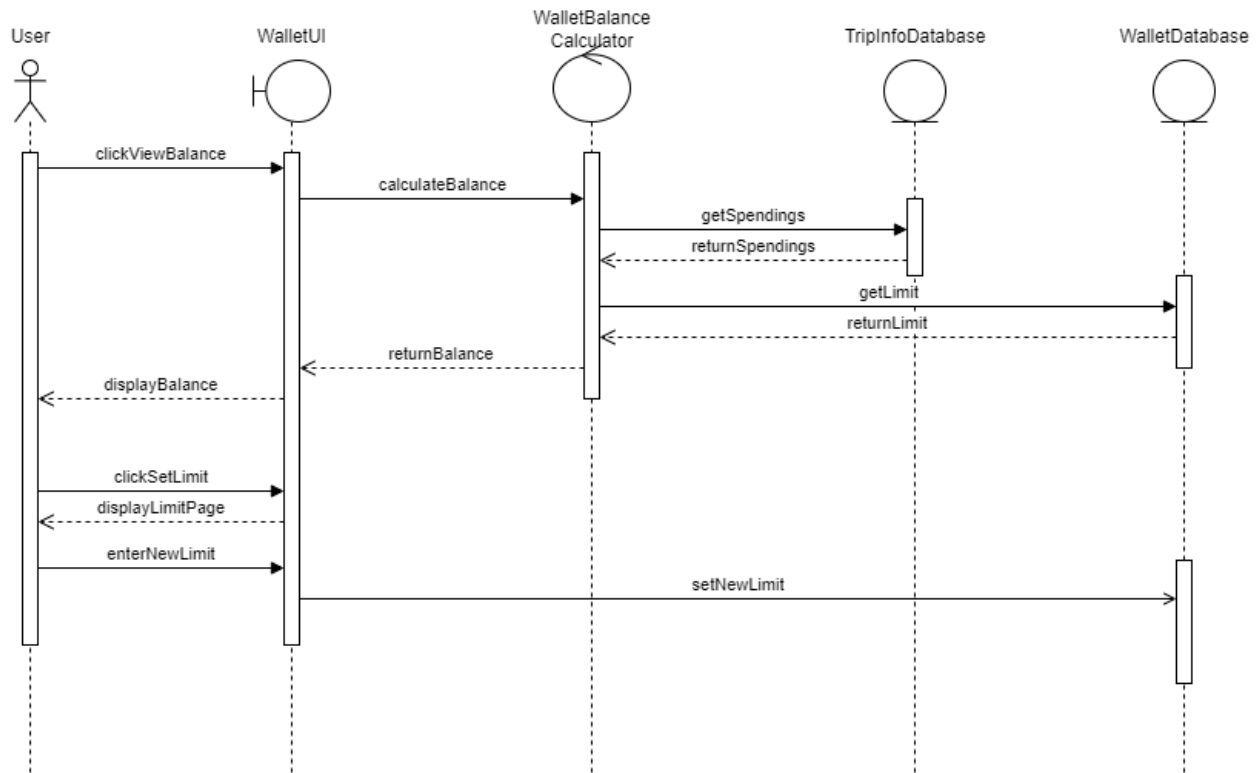
## View Trip History



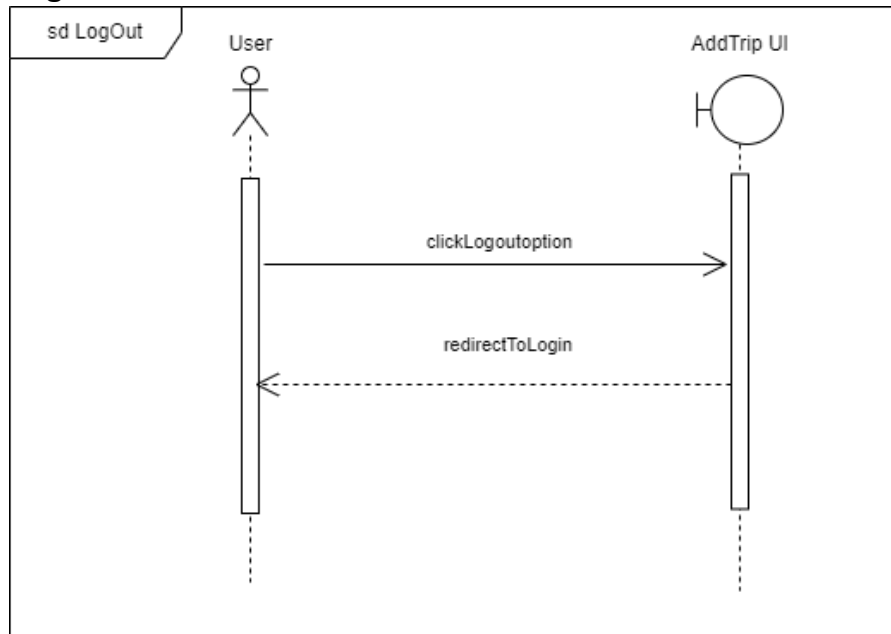


**Edit Trips**

## Manage Wallet Balance

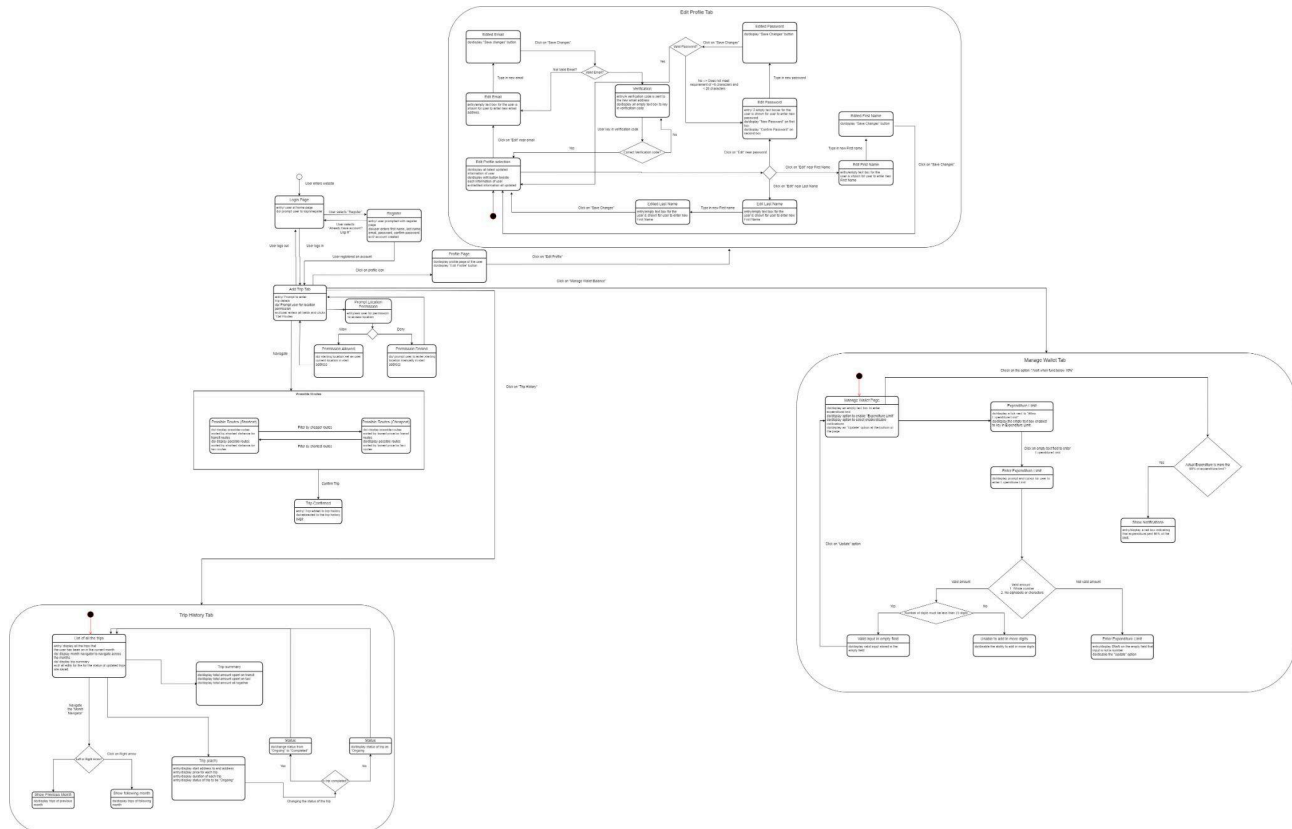


## Log out

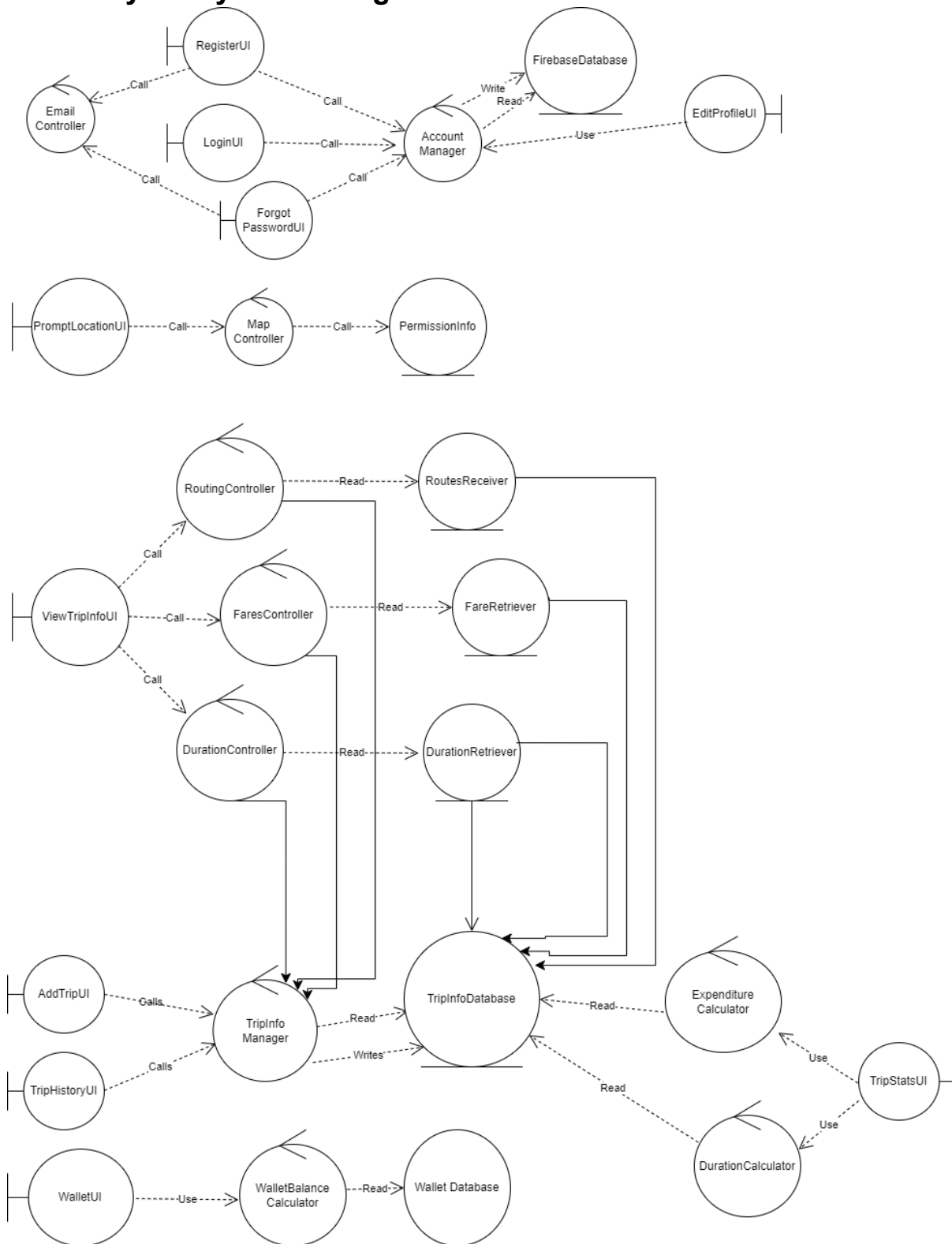


## Dialog Map

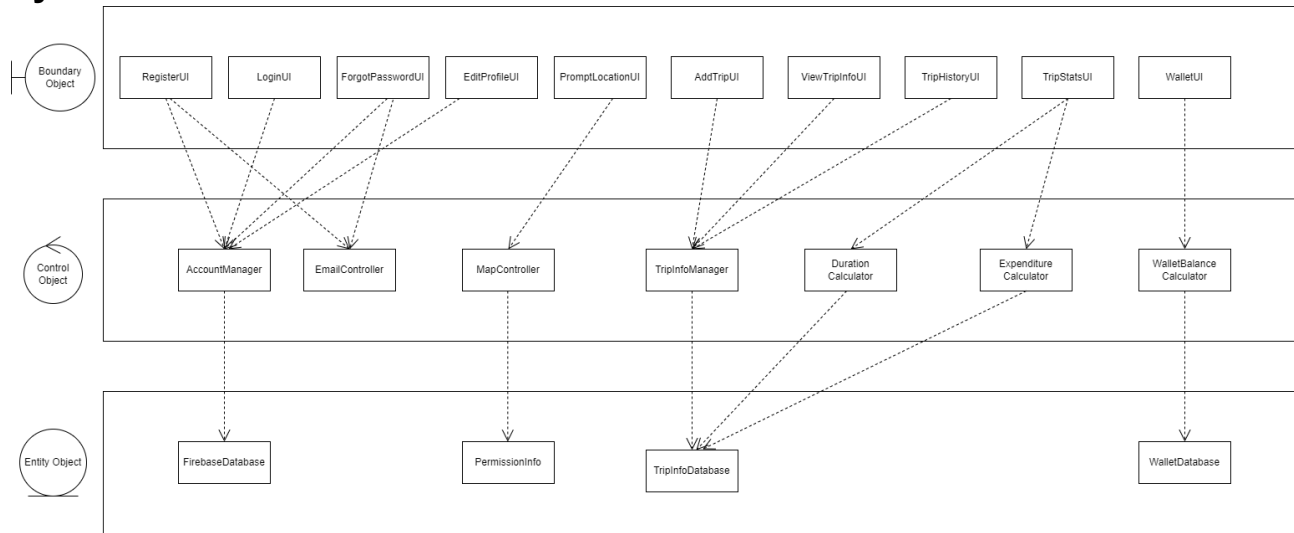
Refer to the repository for a clearer picture.



## Boundary Entity Class Diagram



## System Architecture



## **9. Appendix C: Future Improvements**

We aim to replace the OneMap API with a more stable map API (e.g., Google Maps) once we have secured sufficient funding. The OneMap API, while functional, exhibits occasional instability. Additionally, inaccurately inputted addresses may result in route retrieval failures due to the stringent formatting requirements of onemap.gov.sg's input features. We are committed to continuously enhancing our services and APIs to provide the most stable experience for our users.

We also plan to provide an option within our web application's "Manage Wallet Balance" feature, allowing users to convert their virtual currency into real funds through integration with accredited financial institutions. This enhancement will offer users greater flexibility.

PathFinder currently does not consider weather and terrain conditions when generating routes. However, it is part of our future plans to incorporate this feature, enabling users to make more informed decisions based on real-time conditions.