
Software Requirements Specification

for

PathFinder

Version 1.0 approved

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

27/2/2024

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Product Scope	2
2. System Features	2
2.1 Registration	2
2.2 Login	3
2.3 Edit Profile	5
2.4 Input Destination Details	6
2.5 Prompt Location Permission	7
2.6 Detect Current Location	8
2.7 Display Trip Information	9
2.8 Get Route	10
2.9 Get Duration	11
2.10 Get Price	12
2.11 Confirm Trip	13
2.12 View Trip History	14
2.13 Edit Trip	15
2.14 Manage Wallet Balance	16
2.15 Set Expenditure Limit	17
2.16 Notify Low Balance	19
2.17 Logout	20
3. Other Nonfunctional Requirements	21
3.1 Performance Requirements	21
3.2 Safety Requirements	21
3.3 Security Requirements	21
3.4 Software Quality Attributes	22
4. Other Requirements	22

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

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.

1.2 Intended Audience and Reading Suggestions

Our intended audience would be all the stakeholders in Singapore. This includes the users of this website, the developers of this website, the documentation writers, the product testers, project managers as well as the marketing and sales team.

Our documentation consists of the description about the website, the different use cases and its descriptions, the functional and non-functional requirements as well as the use case diagram. As such, this documentation is intended to be read in chronological sequence by all the stakeholders.

1.3 Product Scope

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

With an interface that is easy to manoeuvre with, this product 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.

2. System Features

2.1 Registration

Use Case ID:	RG1		
Use Case Name:	Register		
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 register an account with an email address and password.
Preconditions:	<ol style="list-style-type: none"> 1. The user does not have an existing account with the same email address. 2. The user enters a valid email address and password.
Postconditions:	<ol style="list-style-type: none"> 1. The user's account is stored in the database. 2. The user is brought to the login page.
Priority:	High
Frequency of Use:	Once per user
Flow of Events:	<ol style="list-style-type: none"> 1. The system prompts the user to enter an email address and password. 2. The user enters email address and password 3. The user clicks the 'Register' button. 4. The system checks for valid email addresses and passwords. 5. The system prompts for user's first name and last name. 6. The system stores details into the database.
Alternative Flows:	<p>AF-S4: If email address already exists within database</p> <ol style="list-style-type: none"> 1. The System displays "Account already registered. Please login or register with another email address." 2. System returns to step 1. <p>AF-S4: If user's password does not meet the requirements</p> <ol style="list-style-type: none"> 1. System displays "Password does not meet the requirements". 2. System returns to step 1.

	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.
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 four text fields for the user to enter information.
 - 1.1.1. The first text field holds the user’s first name for username.
 - 1.1.2. The second text field holds the user’s email address.
 - 1.1.3. The third text field holds 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.
 - 1.2.2. The system must check for a valid password following the requirements.
 - 1.2.2.1. The password must contain at least 8 characters.
 - 1.2.2.2. The password must contain at most 20 characters.
 - 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 login page.

2.2 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"> 1. The user has entered a valid email address. 2. The user’s account exists in the system’s database. 3. The user has entered the correct password corresponding to the email address.
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"> 1. The system prompts the user to enter an email address and password. 2. The user enters email address and password 3. The user clicks the ‘Sign in’ button.

	<ol style="list-style-type: none"> 4. The system checks for valid email addresses and passwords. 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:	<p>AF-S4: If the user enters an invalid email address</p> <ol style="list-style-type: none"> 3. The system displays the message "Invalid email address." 4. The system returns to step 1. <p>AF-S5: If the system fails to find the user's account in the database</p> <ol style="list-style-type: none"> 1. The system displays the message "Account does not exist." 2. The system returns to step 1. <p>AF-S6: If the user entered an incorrect password corresponding to the email address</p> <ol style="list-style-type: none"> 1. The system displays the message "Incorrect 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.

2.3 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:	27/02/2024
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:	1. The user clicks on Manage Profile tab. 2. The system brings the user to the profile page. 3. The user clicks on "Edit profile" button. 4. The system brings the user to the "Edit profile" page. 5. The system displays the user's profile with "Edit" button beside each individual information. 6. The user edits the desired information. 7. The user clicks the "Save changes" button. 8. The system checks if the new information is valid. 9. The system saves and updates the new information of the user in the database.		
Alternative Flows:	AF-S7: If the user enters enters a new information that does not meet the requirements 1. The system displays the message "Error saving changes" 2. The system returns to step 6.		
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 the user's first name with an "Edit" button beside.
 - 1.1.1.1. If the user clicks on the "Edit" button, the system must display a new text box for the user to enter the new first name.
 - 1.1.1.2. If the user clicks on the "Save changes" button, the system must update the new first name on display and in the database.
 - 1.1.2. The system must display the user's email address with an "Edit" button beside.
 - 1.1.2.1. If the user clicks on the "Edit" button, the system must display a new empty text box for the user to enter the new email address.
 - 1.1.2.2. If the user clicks on the "Save changes" button, the system must check that the email address contains one "@" character.
 - 1.1.2.3. If the new email address is valid, the system must send a verification code to the new email address.
 - 1.1.2.4. The user must be able to enter the verification code received into the system.
 - 1.1.2.5. If the verification code is done successfully, the system must update the new email address on display and in the database.
 - 1.1.3. The system must display the user's password address with an "Edit" button beside.

- 1.1.3.1. If the user clicks on the “Edit” button, the system must display two new empty masked text box for the user to enter the new password.
 - 1.1.3.1.1. The first masked text box must be labelled as “New password”.
 - 1.1.3.1.2. The second masked text box must be labelled as “Confirm password”.
- 1.1.3.2. If the user clicks on the “Save changes” button, the system must check that the password meets the requirements.
 - 1.1.3.2.1. The user’s new password must contain at least 8 characters.
 - 1.1.3.2.2. The user’s new password must contain at most 20 characters.
- 1.1.3.3. If the system successfully validates the new password, the system must update the new password in the database.

2.4 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 and the arrival time
Preconditions:	<ol style="list-style-type: none"> 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:	<ol style="list-style-type: none"> 1. The system asks user to name their trip. 2. The system asks for user’s departure time and date. 3. The system asks if user is using current location as starting point 4. If user chooses yes, system goes to LP1 to prompt location permission if not allowed yet, followed by CL1 to access user live location
Flows:	AF-S3: If user chooses no <ol style="list-style-type: none"> 1. System prompts user to enter valid starting address AF-S3: If the user inputs an invalid address <ol style="list-style-type: none"> 1. The system displays the message “Invalid address entered, please try again” 2. User is prompted to enter an address AF-S4: If user rejects permission to access location <ol style="list-style-type: none"> 1. System returns back to DD1 for user to manually input starting 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. System must prompt user to input their name of the trip.
 - 1.2. System must prompt user to input destination in the form of either:
 - 1.2.1. Address
 - 1.2.2. Name of place
 - 1.2.3. Postal Code
 - 1.3. System must inform user if an invalid address is entered
2. System asks for user’s departure time and date.
 - 2.1. System has default calender inbuilt which presets the current date of trip
 - 2.1.1. User can choose to toggle the calender to set an upcoming date.
 - 2.2. System has default clock inbuilt which presets the current time of trip.
 - 2.2.1. User can choose to toggle the clock to set preferred timing.
3. System asks if user is using current location as starting position
 - 3.1. If user selects yes
 - 3.1.1. System goes to LP1 to prompt location access permission. User is required to allow permission.
 - 3.1.2. If user declines location access permission, system returns to DD1 for user to enter starting address manually
 - 3.2. If user selects no
 - 3.2.1. System prompts user to enter a valid starting address in the form of either:
 - 3.2.1.1. Address
 - 3.2.1.2. Name of place
 - 3.2.1.3. Postal Code
 - 3.2.2. System must inform user if an invalid address is entered

2.5 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

2.6 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:	1. The user is logged in. 2. The user has allowed permission for location access (LP1)
Postconditions:	The system gets user's current location information
Priority:	High
Frequency of Use:	High
Flow of Events:	1. User allows permission for location access in LP1 2. System gets user current location information 3. User's current location information is used to be set as trip start location
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

2.7 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:	27/02/2024

Actor:	User
Description:	Trip Information displays all the required details 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"> 1. The user is logged in. 2. The user has input their destination details
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"> 1. Duration trip takes 2. Fare 3. Route of trip
Priority:	Very High
Frequency of Use:	Very High
Flow of Events:	<ol style="list-style-type: none"> 1. User inputs destination details. 2. Trip information is displayed on the screen immediately to showcase details of each trip. 3. User scrolls through different options of transportation and clicks dropdown to view the details under each option. 4. User can choose to filter their trip options <ol style="list-style-type: none"> a. For example, "Fewer Transfers" will filter and only display trip options with less than 2 modes of transport b. "Shortest duration" and "Cheapest Route"
Alternative Flows:	<p>AF-S3: If User decides to not proceed with any of the trips</p> <ol style="list-style-type: none"> 1. User presses back and trip information is no longer shown on screen 2. User will be redirected back to inputting their destination details page
Exceptions:	<p>EX 1 : API fails to retrieve some of the information</p> <ol style="list-style-type: none"> 1. An error message is shown to inform User that time/fare/routing services are unavailable
Includes:	Confirm Trip, View Trip History, Get Duration, Get Route, Get Price
Special Requirements:	Nil

Assumptions:	There can be trip options which include a multiple modes of transportations. Hence,combined cost,duration and route will be displayed. All API used works and information retrieval is real time and accurate.
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 filtering option "Shortest Duration", "Cheapest Route" and "Fewer Transfers".
 - 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 a dropdown for each trip option in the display to show more information about trip details
 - 1.3.1. The system must display estimated time of arrival for each trip option
 - 1.3.2. The system must display total estimated duration time for each trip option
 - 1.3.3. The system must display total estimated fare cost for each trip option
 - 1.3.4. The system must display total estimated route for each trip option.

2.8 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, Google 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"> 1. The user is logged in. 2. The user has input their destination details
Postconditions:	<ol style="list-style-type: none"> 1. The system has fetched the routes of each trip option under Display Trip Information
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The system checks for each possible route from User's start point to end point using APIs.

	2. The system fetches and displays routes under each dropdown menu for each possible mode of transport.
Alternative Flows:	AF-S1: If no route is available : 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.” 2. User will be redirected to input destination details’ page.
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.

2.9 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, Google 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"> 1. The user is logged in. 2. The user has input their destination details
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"> 1. The user inputs destination details, 2. The system checks for estimated duration time from User’s start point to end point using APIs. 3. The system fetches and displays estimated duration under each dropdown menu for each possible mode of transport.

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.

2.10 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"> 1. The user is logged in. 2. The user has input their destination details
Postconditions:	<ol style="list-style-type: none"> 1. The system has fetched the price of each trip option under Display Trip Information
Priority:	Medium
Frequency of Use:	Medium
Flow of Events:	<ol style="list-style-type: none"> 1. The system retrieves the possible routes to destination from "Get Route". 2. The system fetches the estimated prices for each transportation used in each trip from "Get Route" 3. The system stores the estimated prices for each transportation used in each trip in the system. 4. The system displays the estimated prices under each dropdown menu for each mode of transport used for each trip
Alternative Flows:	Nil
Exceptions:	Nil
Includes:	Nil
Special Requirements:	Nil
Assumptions:	<ol style="list-style-type: none"> 1. API retrieval is accurate and timely done. 2. There is at least one route from the user's current location to the destination

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.

2.11 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"> 1. The user is logged in. 2. The user has inputted their destination details 3. System has displayed all possible trip options for user to select from
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"> 1. With the list of trip options, the user selects their desired one and clicks “Confirm Trip” button 2. If the user’s ongoing trip ends, the system must update the ongoing trip as “completed” in the Trip History.
Alternative Flows:	AF-S1 : If User chooses to cancel or change their trip halfway <ol style="list-style-type: none"> 1. Current trip is cancelled. 2. User gets redirected to display page of all possible trip options.
Exceptions:	NIL
Includes:	Login
Special Requirements:	NIL
Assumptions:	<ol style="list-style-type: none"> 1. User must click on one of trip options to confirm a trip. 2. System is connected to the APIs before and during the trip
Notes and Issues:	NIL

Functional Requirements

1. The system must display a “Confirm Trip” button for the user to click and confirm their trip option.
2. The system must update the user’s confirmed trip
 - 2.1. The system must save the user’s confirmed trip in the database as ongoing.
 - 2.2. The system must update the Trip History with the newly confirmed trip.

2.12 View Trip History

Use Case ID:	TH1		
Use Case Name:	View Trip History		
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:	Displays the past trips that the user has saved using “Add Trip”.
Preconditions:	<ol style="list-style-type: none"> 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:	<ol style="list-style-type: none"> 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. 5. The system displays the user’s monthly total duration spent on transport.
Alternative Flows:	AF-S1: If the user does not have any past trips yet <ol style="list-style-type: none"> 1. The system displays the message “No past trips available”.
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 date.
 - 1.2.2. The system must display the completed trip’s start location.
 - 1.2.3. The system must display the completed trip’s destination.
 - 1.2.4. The system must display the completed trip’s travel duration.
 - 1.2.5. The system must display the completed trip’s mode of transportation taken.
 - 1.2.6. The system must display the completed trip’s distance travelled
 - 1.2.7. 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 option for user to switch between months to view previous months' history.
- 1.4. The system must display the user's total monthly duration spent on transport at the top of the page below expenditure.
 - 1.4.1. The system must provide option for user to switch between months to view previous months' history.

2.13 Edit Trip

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

Actor:	User
Description:	User is allowed to edit past completed trips in the Trip History page.
Preconditions:	<ol style="list-style-type: none"> 1. The user is logged in. 2. The user has one or more completed trips. 3. The user is in the Trip History page.
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"> 1. The user clicks on the "Edit trip" button on the desired trip. 2. The system prompts the user with various options to edit the trip. 3. The user clicks on the "Save changes" button on the edited trip. 4. The system updates the changes on the edited trip on display and in database.
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 trips in the Trip History page.
 - 1.1. The system must display an "Edit" button beside each trip.
 - 1.1.1. If the user clicks on the "Edit" button, the system must bring the user to the Edit page of the selected trip.
 - 1.1.2. The system must display an option for editing the following trip factors:

- 1.1.2.1. Start time
- 1.1.2.2. End time
- 1.1.2.3. Duration
- 1.1.2.4. Price
- 1.1.2.5. Status
- 1.1.3. The system must display a “Save changes” button.
 - 1.1.3.1. If the user clicks on the “Save changes” button, the system must update the selected trip’s changes on display and in database.

2.14 Manage Wallet Balance

Use Case ID:	WB1		
Use Case Name:	Manage Wallet Balance		
Created By:	NAFEES ABDUL KADER	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	12/02/2024	Date Last Updated:	12/02/2024

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:	1. If the user wants notifications, the notifications will be switched on 2. If the user does not want notifications, the notifications will be switched off. 3. If the user wants to set an expenditure limit, the user will activate the expenditure limit option. 4. If the user does not want to set an expenditure limit, the use will not activate the expenditure limit option.
Priority:	High
Frequency of Use:	High
Flow of Events:	1. The user will click the Manage Profile tab 2. The user will click on the “Manage Wallet Balance” option. 3. The user will then see one of the options to allow notifications. 4. If the user wants to on the notifications, the user will press on the “Allow notifications” option 5. The user will also see the options to activate the expenditure limit. 6. If the user wants to set an expenditure limit, he/she will press on the “Allow expenditure limit”
Alternative Flows:	AF-S4: If the user does not want to have notifications 1. The user will press on the “Do not allow notifications” option.

	AF-S5: If the user does not want to set an expenditure limit 1. The user will press on the “Do not allow expenditure limit”.
Exceptions:	Nil
Includes:	1. Set Expenditure Limit 2. Notify Low Balance
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 display a “Notifications” option under “Manage Wallet Balance”.
 - 1.2.1. The system will prompt the user to select between 2 different options under “Notifications”.
 - 1.2.1.1. If the user wants to on the notifications, the user will click on “Allow notifications” option
 - 1.2.1.2. If the user wants to off the notifications, the user will click on “Do not allow notifications” option.
 - 1.3. The system will also display an “Allow expenditure limit” option under “Manage Wallet Balance”.
 - 1.3.1. The system will prompt the user to select between 2 different options under “Allow expenditure limit”.
 - 1.3.1.1. If the user wants to set an expenditure limit, the user will click on “Allow expenditure limit”.
 - 1.3.1.2. If the user does not want to set an expenditure limit, the user will click on “Do not allow expenditure limit”.

2.15 Set Expenditure Limit

Use Case ID:	EL1		
Use Case Name:	Set Expenditure Limit		
Created By:	NAFEES ABDUL KADER	Last Updated By:	NAFEES ABDUL KADER
Date Created:	12/02/2024	Date Last Updated:	12/02/2024

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:	2. The user is logged in.
Postconditions:	The system stores the expenditure limit that the user has keyed in. For eg.

	"\$100 is keyed as transportation expenditure limit by the user"
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"> 1. Users will click on the "Set Expenditure Limit" button 2. Users will then be redirected to a page with an empty field. 3. Users can then key in their intended expenditure limit in the empty field, in numbers. 4. The number keyed in can be a whole number or a decimal number. 5. If the number is a decimal number, the user can only key in up to 2 decimal places. 6. For both whole and decimal numbers, the system allows a total of 9 numbers to be keyed in at a time. <p>Eg.</p> <ul style="list-style-type: none"> - 123 - 14578 - 158943.77 <ol style="list-style-type: none"> 7. Spaces are disabled in this option 8. Once the limit is keyed in, users will then press a "Confirm" button 9. A pop up message will then show to indicate that the expenditure limit is saved in the user's account. 10. If the user wants to indicate a new expenditure limit, he/she can repeat steps 1 to 8.
Alternative Flows:	<p>AF-S3: If the user does not key in numbers, but alphabets and other characters</p> <ol style="list-style-type: none"> 1. The system will indicate in red near the empty field that the user can only key in numbers and nothing else 2. The "Confirm" button will be in grey colour to prevent the user from saving the expenditure limit as alphabets and characters <p>AF-S5: If the user has a decimal number with more than 2 decimal places</p> <ol style="list-style-type: none"> 1. The system will round the number to 2 decimal places once the user has clicked the "Confirm button". <p>AF-S6: If the user starts to key in numbers with more than 9 digits</p> <ol style="list-style-type: none"> 1. The system will disable the user from entering more numbers exceeding 9 digits.
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 "Set Expenditure Limit" 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 9 digits.
- 1.1.3. The amount keyed in could be in whole number.
- 1.1.4. The amount keyed in could be in decimal.
 - 1.1.4.1. User can only key in the decimal up to 2 decimal places
- 1.1.3.2. If the user keys in a decimal with more than 2 decimal places, the system will automatically round the number to 2 decimal places.
- 1.1.5. Space bar will be disabled.
- 1.1.6. 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 red near the field to only key in numbers and nothing else
 - 1.1.6.1. If an alphabet or character is keyed in, the system will disable the “Confirm” button to prevent the user from saving that data.
- 1.2. The system will display a “Confirm” button.
 - 1.2.1. Once the user has keyed in the expenditure limit, the user can press “Confirm” to save the data.

2.16 Notify Low Balance

Use Case ID:	RN1		
Use Case Name:	Notify Low Balance		
Created By:	NAFEES ABDUL KADER	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	12/02/2024	Date Last Updated:	27/02/2024

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 displays a notification on the webapp.
Priority:	High
Frequency of Use:	High
Flow of Events:	<ol style="list-style-type: none"> 1. If the user has spent at least 90% of its expenditure limit, the system displays a notification that states “Low amount left for expenditure” 2. The system will show the notification on the top of the webpage.
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. The system sends the user through notifications when they are about to reach their transportation expenditure limit.

1.1.1. The notification received will be “Low amount left for expenditure.”

1.2. The system will display the notifications on the top of the webpage

2.17 Logout

Use Case ID:	LOG2		
Use Case Name:	Logout		
Created By:	NAFEES ABDUL KADER	Last Updated By:	PUSHPARAJAN ROSHINI
Date Created:	27/02/2024	Date Last Updated:	27/02/2024

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 brought back to the login page
Priority:	High
Frequency of Use:	After every usage
Flow of Events:	1. Users will click on the logout button on the top right hand of the home page (Add Trip Tab) 2. The system will direct the user to the login page.
Alternative Flows:	-
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 user will be directed to the login page.

3. Other Nonfunctional Requirements

3.1 Performance Requirements

- 3.1.1 Users should receive route suggestions within 5 seconds of entering their search criteria.
- 3.1.2 The website should be able to handle high traffic volumes without performance degradation.
 - 3.1.2.1 The system should be able to handle a concurrent user load of at least 1000 users.
- 3.1.3 Route data, including prices and travel times, should be updated regularly to ensure accuracy.

3.1.3.1 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 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

3.2 Safety Requirements

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

3.3 Security Requirements

3.3.1 Since any data keyed into the website is personal, privacy has to be maintained. Hence, the user's password should be encrypted.

3.3.2 User data, including search history and preferences, and financial information should be protected with encryption and secure storage practices.

3.4 Software Quality Attributes

3.4.1 **Usability:** The website should be user-friendly, intuitive, and easy to navigate, even for first-time users.

3.4.2 **Reliability:** The website should be highly available and minimise downtime to ensure consistent user experience.

3.4.3 **Testability:** The code should be well-structured and documented, facilitating efficient testing and bug fixing.

3.4.4 **Maintainability:** The code should be well-documented, modular, and easy to understand for future maintenance and updates.

4. Other Requirements

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 4 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 details such as cost and duration of their ongoing trip, and once they make changes they can save changes and complete their trip.
Manage Wallet Balance	Users will 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.

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

1. USE CASE DIAGRAM

