4. System Features

Please refer to the Use Case Description document for the Description, Priority, Stimulus and Response Sequences of each system feature. This document only consists of their respective Functional Requirements.

4.1 Register (RR01)

Functional Requirements:

1. The user must be able to register for a new account via our registration system.
    1.1. The system must display 3 text fields for the user to enter registration information.
        1.1.1. The system must display one text field for Email Address
        1.1.2. The system must display one text field for Password
        1.1.3. The system must display one text field for Confirm Password
    1.2. The system must display a “Register” button.
    1.3. When the user clicks “Register”, the system must validate that all required fields are not left empty.
    1.4. When the user clicks “Register”, the system must validate that the Email Address entered is a valid email
    1.5. When the user clicks “Register”, the system must validate that the password entered must meet the following requirements:
        1.5.1. A minimum of 6 characters in length.
        1.5.2. Contain at least one (1) character from three (3) of the following categories: Uppercase letter (A-Z) Lowercase letter (a-z) Digit (0-9) Special character (~`!@#$%^&*()+=_-{}[]\|:;”"?/<>,.).

4.2 Login (LL01)

Functional Requirements:

1. The user must be able to log into the system.
    1.1. The system must display 2 text fields for the user to enter his login information.
        1.1.1. The system must display one text field for Email Address.
        1.1.2. The system must display one text field for Password.
    1.2. The system must display a “Login” button.
    1.3. When the user clicks “Login”, the system must validate that all fields are not left empty.
    1.4. When the user clicks “Login”, the system must verify the information obtained from the text fields.
        1.4.1. The email must be found in the database of the system.
        1.4.2. The password entered must match the password of the user.
    1.5. When the user clicks “Login” and information is verified to be a valid account, the system must log the user into the home screen of the system.

4.3 Add Trip (AT01)

Functional Requirements:

1. The user must be able to add a trip.
    1.1. The system must display 3 text fields with autocomplete for the user to enter trip information.
        1.1.1. The system must display one text field with autocomplete for Start Point.
        1.1.2. The system must display one text field with autocomplete for End Point.
        1.1.3. The system must display one text field with autocomplete for Trip Name.
    1.2. The system must display an “Add Trip” button.
    1.3. The system must display a “Cancel” button.
    1.4. When the user clicks “Add Trip”, the system must save the trip information to the database.

4.4 View Saved Trips (VT01)

Functional Requirements:

1. The system allows the users to view previously saved trips
    1.1. The system must display a list of all the previously saved trips by the user
    1.2. The system must allow the user to click on one of the listed trips to Execute Trip (Use Case ID: ET01)
    1.3. The system must refresh the page when any changes occur

4.5 Delete Trip (DT01)

Functional Requirements:

1. The user must be able to delete a previously saved trip from the system
    1.1. The system must display a list of previously saved trips
    1.2. The user must be given the option to select one or multiple trips from this list to delete
    1.3. The system must display a confirmation message asking the user to confirm if they want to go forward with deleting the selected trips
        1.3.1. The message must clearly state the implications of deleting a previously saved trip
    1.4. The user must be given a choice to either confirm the deletion or cancel it
    1.5. After the confirmation from the user, the deletion must be reflected in the system

4.6 Edit Trip Name (ET01)

Functional Requirements:

1. The user must be able to edit the name of a previously saved trip
    1.1. The system must display a list of previously saved trips
    1.2. The user must be given the option to select one trip from this list to edit
    1.3. The system must display a text field for the user to enter the new name of the trip
        1.3.1. The text field must allow the user to input 64 characters

1.4. The system must give the user a choice to confirm the change or cancel editing the trip
    1.4.1. The system must not allow the user to confirm the change if the text field is empty or if the new name entered is the same as the current name of the trip
1.5. After the confirmation from the user, the change must be reflected in the system

4.7 Star Trip (ST01)

Functional Requirements:

1. The user must be able to 'Star' a previously saved trip on the system
    1.1. The system must display a list of previously saved trips
    1.2. The user must be given the option to select one trip from this list to star
    1.3. The system must move the starred trip to the top of the list of previously saved trips
    1.4. The system must show an indication (either via a Star icon or a Pin icon beside the trip) that the trip has been starred.

4.8 Execute Trip (ET02)

Functional Requirements:

1. The user must be able to run an instance of a saved trip.
    1.1. The user must be able to select a mode of transport
        1.1.1. There should be a button for Private Car/Motorbike
        1.1.2. There should be a button for Public Transport
        1.1.3. There should be a button for Private Hire Taxi
        1.1.4. There should be a button for Bicycle
        1.1.5. There should be a button for Walking
    1.2. The system must display the route of the trip with the selected mode of transport
        1.2.1. The system must retrieve the route from Use Case GR1 (Get Route)
        1.2.2. The system must display the interactive map retrieved from the Google Maps API
        1.2.3. The system must display the route on top of the interactive map
        1.2.4. The user must be able to zoom in and out using the pan and pinch gestures on screen
    1.3. The system must display the price for the selected mode of transport
        1.3.1. The system must retrieve the price from Use Case GP1 (Get Price)
        1.3.2. The system must display the estimated price beside the selected mode of transport
        1.3.3. The price must be displayed in S$
        1.3.4. The price must be rounded to 2 decimal places
    1.4. The system must save an instance of the trip to the database
        1.4.1. The system must save the saved trip ID
        1.4.2. The system must save the mode of transport
        1.4.3. The system must save the time of execution
        1.4.4. The system must save the estimated price
        1.4.5. The system must save the duration of the trip
        1.4.6. The system must save the distance covered

4.9 Get Route (GR01)

Functional Requirements:

1. The system must get retrieve the route for a given starting point and destination point
    1.1. The system must query the Google Maps API to retrieve the route for a given starting point and destination point
    1.2. The system must also fetch the directional instructions form the Google Maps API
    1.3. The system must display pins on the map indicating points of change in mode of transport (walking, bus, train) when showing a public transport route.

4.10 Get Price (GP01)

Functional Requirements:

1. The system must retrieve the price for a given route and mode of transportation
    1.1. The system must make an API call to the appropriate API to get the price of a route
        1.1.1. The system must contact the OneMap API when using Public Transport
        1.1.2. The system must contact the FareFinder API when using private hire taxi
        1.1.3. The system must contact the TollGuru API when using car
        1.1.4. No API calls to be made for other modes of transport

4.11 View Trip History (VH01)

Functional Requirements:

1. The system must query historical trip information from the database
    1.1. The data retrieved need to be sorted by time of execution in descending order
2. The system need to render the historical trip information on Trip History page
    2.1. The system must refresh the page when any changes occur

4.12 Edit Past Trip Price (EP01)

Functional Requirements:

1. The system must store edited information as session storage
2. The system must query the database according to the index of trip
3. The system must update the index using session storage data
4. The system must dynamically render the updated price on the trip information page after confirmation of edit by the user

4.13 View Stats (VS01)

Functional Requirements:

1. The system must query the database for all trip statistics
    1.1. The system must have calculated all the statistics based on trip history
    1.2. The system must have saved all calculated statistics into database
2. The system must render the statistics in the form of percentages/charts
    2.1. The rendered analytics dashboard must be updated whenever a trip is executed

4.14 Interactive Data Visualisation (DV01)

Functional Requirements:

1. The system must show an interactive pie chart where each slice can be clicked by the user
    1.1. The proportion of trips taken by Car is a slice
    1.2. The proportion of trips taken by Taxi is a slice
    1.3. The proportion of trips taken by Public Transport is a slice
    1.4. The proportion of trips taken by Walking is a slice
    1.5. The proportion of trips taken by Cycling is a slice
2. The system must display in a tabular form the data of all trips taken by the chosen mode of transport
    2.1. The system must display the executed trip ID
    2.2. The system must display the distance covered in that trip
    2.3. The system must display the time taken for that trip
    2.4. The system must display the cost of the trip

4.15 Static Data Visualisation (DV02)

Functional Requirements:

1. The system must show a bar graph of the cost per trip
    1.1. The y-axis is the cost in SGD
    1.2. The x-axis the recent trip number
    1.3. If there are more than 10 trips in the system, the 10 most recent trips must be used
2. The system must show a line graph of the time taken per trip
    2.1. The y-axis is the time taken in minutes
    2.2. The x-axis the recent trip number
    2.3. If there are more than 10 trips in the system, the 10 most recent trips must be used

4.16 Settings (SE01)

Functional Requirements:

1. The system must show some user preference options
    1.1. The system must show an option to edit account details
    1.2. The system must show an option to switch display themes
    1.3. The system must show an option to logout
2. The system must display the current user's name

3. The system must display the current app version

4.17 Edit Account Details (AD01)

Functional Requirements:

1. The system must show a few options to edit account details
    1.1. The system must show an option to change display name
        1.1.1. The system must allow the user to cancel the name change
    1.2. The system must show an option to change user email
        1.2.1. The system must ensure that the new email address is valid
        1.2.2. The system must ensure that the new email address is not already registered
        1.2.3. The system must allow the user to cancel the email change

4.18 Set Theme (SE02)

Functional Requirements:

1. The system must switch the current UI theme
    1.1. If the current UI Theme is light, the UI theme is changed to dark
    1.2. If the current UI Theme is dark, the UI theme is changed to light
    1.3. The default UI theme is light
2. The system must ensure that all UI elements abide to the new theme
    2.1. All UI elements must check the theme before they are rendered
    2.2. The change in theme should trigger a re-render in all the elements currently being displayed