

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

SC2006 Software Engineering

Lab 2 Deliverables

Tutorial Group: Z58

The 7th

Name	Matriculation Number
Joanne Christina Salimin	U2120304J
Koh Hao Sheng	U2122672K
Lim Boon Hian	U2120791F
Lim Dong Wan	U2122455D
Loke Yong Jian	U2120476G
Ng Woon Yee	U2120622C
Tiwana Teg Singh	U2122816B

Table Of Contents

Project Mission Statement	3
Market Research	3
Target Audience	3
Functional Requirements	4
Non-Functional Requirements	12
Data Dictionary	13
Use Case Diagram	15
Use Case Descriptions	16
Initial UI Mockup Diagram	39
Class diagram of entity classes	42
Key boundary classes and control classes	43
Sequence diagrams of some use cases	44
Initial Dialog map	48

Project Mission Statement

The 7th team will develop an application that makes instant travel more accessible and stress-free for commuters and passengers within Singapore. The app does so by offering a comprehensive and practical aggregator platform to compare and select the most affordable and effective mode of transportation, according to the commuters' needs and wants. Our platform will compile the estimated travel time and fare data from multiple sources and transport modes, such as public transportation, ride-hailing and taxi services, as well as private vehicle rental firms to assist commuters in making informed choices and saving money and/or time on their journeys. In order to help Singapore achieve its vision of becoming a smart nation, we seek to streamline the transportation industry and encourage smart mobility by empowering regular commuters to openly compare the transportation options available to them.

Market Research

In Singapore, the car ownership rate is estimated to be around 12% according to The Straits Times. This means that a significant portion of the population is unable to reach their desired destinations quickly and efficiently. As a result, the revenue of ride-hailing services has seen a sharp increase, rising from SGD 1.05 billion in 2017 to an expected SGD 1.65 billion in 2027. This growth has attracted many companies, such as Grab, Gojek, Maxim, Ryde, etc., to compete for a share of the market in Singapore. We have noticed that the fares offered by these companies often vary due to differences in their pricing strategies.

Recently, car rental services have also entered the market and are gaining popularity. Like ride-hailing services, they have unique pricing strategies. With so many options available in Singapore, there is an opportunity to create an app that compares prices between ride-hailing, car rental, public transportation, and taxi services. It simplifies the process for customers by eliminating the need to check each app individually.

Target Audience

Our target audience consists of users who are looking for the most economical traveling options in Singapore. They may also be tech-savvy and prefer having all their ride-hailing needs fulfilled through a single platform.

Functional Requirements

1. Landing Page

1.1 When the User clicks on the application, an animation displaying the application's logo and name must be shown.

1.2 The application must check whether or not the User has given Global Positioning System (GPS) permission.

 1.2.1 If GPS permission is not yet given, User must be prompted to give GPS data permission.

 1.2.2 If the User refuses to give the GPS permission to the application once, the application must stop asking for it in the future.

 1.2.2.1 If the User does not provide GPS permission to the application, the application will not locate the User, instead it must display the zoomed-out of the Singapore map.

1.3 The application must display the option to change language settings to one of the four main native languages in Singapore: English, Chinese, Malay and Tamil.

 1.3.1 Once the language is chosen, all the text must be displayed in the chosen language.

1.4 The application must display three mutually exclusive options: "Register", "Sign In" and "Continue as Guest".

 1.4.1 If "Register" is chosen, User must be redirected to the Registration Page.

 1.4.2 If "Sign In" is chosen, User must be redirected to the Login Page.

 1.4.3 If "Continue as Guest" is chosen, User must be redirected to the Starting Location Input page.

2. Registration Page

2.1 User must input their email address into an input textbox to sign up for a new account.

 2.1.1 The User must click on the input text box.

 2.1.2 The application must check whether or not the User has input a valid email.

 2.1.2.1 If the email address is not valid, the textbox must be highlighted red, and the message "Invalid Email" must be displayed.

2.2 The User must input a 12-18 character-long alphanumeric password into an input textbox

 2.2.1 The User must click on the input text box.

 2.2.2 The application must check whether or not User has input a valid password

 2.2.2.1 If the password is not valid, the textbox must be highlighted red, and the message "Invalid Password" must be displayed.

2.3 The application must verify the authenticity of the User's inputted email through a verification email link sent to the User.

2.4 Once verification is complete, the user must be redirected to the Starting Location Input page.

3. Login Page

3.1 Users with an existing account must be able to login into the application with their respective email address and password.

- 3.1.1 The User must click on the input text box.
- 3.1.2 The User must input into the top input text box the email address used when registering for the account.
- 3.1.3 The User must input into the bottom input text box the password created when registering for the account.
- 3.2 The system must verify the correctness of the email and password entered by the User.
 - 3.2.1 If the login information cannot be ascertained, the system must print out an error message to the User, prompting them to reinput their login credentials.
 - 3.2.2 If the login information can be ascertained, the system must allow the User to log into the Application.
- 3.3 The system must also display a “Forgot Password” field to allow Users to retrieve their respective account login information.
- 3.4 The User must input their email address into an input textbox to receive an email from the system allowing them to reset their password.
- 3.5 The system must send the User an email to the Users that allows them to reset their password.

4. Starting Location Input

- 4.1 The User must enter both a starting location and destination address in order to query the application.
- 4.2 The User must be able to input a starting location, which can be either the User’s current physical location, based on the User’s phone GPS data, or another location chosen by the User.
 - 4.2.1 The User must click on the input text box.
 - 4.2.2 The User must be able to input the starting location in the input text box.
 - 4.2.3 The starting location must be in text format, and can include alphanumeric characters.
 - 4.2.4 The starting location must be able to be inputted by pinning the location on the map.
 - 4.2.5 The User must be able to remove their input by clicking a “X” button, which is located at the edge of the input text box.
- 4.3 Once the User inputs the starting location, the application must validate that the starting location is a valid address in Singapore.
 - 4.3.1 If the starting location is an invalid address in Singapore, the application must display the message “Invalid address! Please try again.”
 - 4.3.2 If the starting location is a valid address in Singapore, the application must allow the User to proceed entering a destination address.

5. Destination Input

- 5.1 The User must be able to input a destination address.
 - 5.1.1 The User must click on the input text box.
 - 5.1.2 The User must input the destination address in the input text box.
 - 5.1.3 The destination address must be in text format, and can include alphanumeric characters.
 - 5.1.4 The destination address must be inputted by pinning the location on the map

5.1.5 The User must be able to remove their input by clicking a “X” button, which is located at the edge of the input text box.

5.2 Once the User inputs the destination address, the application must validate that the destination address is a valid address in Singapore.

5.2.1 If the destination address is an invalid address in Singapore, display the message “Invalid address! Please try again.”

5.2.2 If the destination address is a valid address in Singapore, the application must allow the User to see a list of query results for the User’s requested trip, based on the User’s starting location and destination addresses.

6. Search Function

6.1 Once the user input both valid starting address and destination address, the application must display and sort the query results based on travel time or travel fare.

6.3.1 The application must first display and sort the query results based on travel fare, by default.

6.3.1.1 The User must be able to toggle between the 2 buttons - “Cheapest” and “Fastest” - to display the query result based on travel fare and travel time respectively.

6.3.2 The application must provide a filter button.

6.3.2.1 After the button is clicked, the application must display an extra query menu.

6.3.2.2 The user must be able to indicate the number of travelers in the query menu.

6.3.2.2.1 After the user indicates the number of travelers, the application must update the result list, such that all the results in the result list are updated based on the number of travelers.

6.3.2.4 The user must be able to indicate whether they prefer pets-friendly transportation mode.

6.3.2.4.1 If the user indicates they prefer pets-friendly transportation mode, the application must update the result list, such that all the results in the query results are pets-friendly.

6.3.2.4.2 If the user indicates they do not prefer pets-friendly transportation mode, the application must update the result list, such that all the results in the query results are either pets-friendly or not pets-friendly.

7. Travel Time and Fare Comparison from Public Transport (Bus/Train) / Ride-Hailing / Rental / taxi Options

7.1 The system must return a report after being queried.

7.2 If the User selects travel time as their choice, the report must be a result list of travel time of the respective transportation mode.

7.2.1 The result list must contain multiple result entries.

7.2.2 Every result entry must include one and only one transportation mode, which allows the User to travel from starting location to destination.

7.2.3 The result list must be sorted in the order of increasing travel time, with the first travel option at the top of the list having the shortest travel time involved.

7.2.4 Each result entry in the list must include the price of the corresponding transportation.

7.2.4.1 The price in the results must be a number, shown with two decimal points.

7.2.4.2 For public transport, the price returned must be calculated based on the travel distance between the starting point and the destination.

7.2.4.2.1 The system must decide the travel distance based on the recommended public transportation travel route by the Google Maps API.

7.2.4.3 For car-rental, the price returned must be calculated by multiplying estimated travel time with rental fee per unit time.

7.2.4.3.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.2.4.4 For ride-hailing, the price returned must be obtained from the ride-hailing companies' API.

7.2.4.5 For taxi, the price returned must be calculated by multiplying travel distance with taxi fare per distance.

7.2.4.5.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.2.5 Each result in the list must include the travel time of the corresponding transportation method.

7.2.5.1 The travel time in the results must be in 24 hour format.

7.2.5.2 For public transport, the time returned must be calculated based on the travel distance between the starting point and the destination.

7.2.5.2.1 The system must decide the travel distance based on the recommended public transportation travel route by the Google Maps API.

7.2.5.2.2 The result must consider the travel time of traveling to the bus stations and/or MRT stations.

7.2.5.3 For car-rental, the time returned must be calculated based on the travel distance.

7.2.5.3.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.2.5.3.2 The result must include the time taken to travel to the rental car place.

7.2.5.4 For ride-hailing, the time returned must be obtained from the ride-hailing companies' API.

7.2.5.5 For taxi, the time returned must be calculated based on the travel distance.

7.2.5.5.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.2.6 Each result in the list must include the company name, which provides the corresponding transportation method.

7.3 If the User selects travel fare as their choice, the report must be a result list of travel fare of the respective transportation mode.

7.3.1 Every result in the result list must include one and only one transportation mode, which allows the User to travel from starting location to destination.

7.3.2 The result list must be sorted in the order of increasing travel fare, with the first travel option at the top of the list having the lowest cost of travel involved.

7.3.3 Each result in the list must include the price of the corresponding transportation.

7.3.3.1 The price in the results must be a number, shown with two decimal points.

7.3.3.2 For public transport, the price returned must be calculated based on the travel distance between the starting point and the destination.

7.3.3.2.1 The system must decide the travel distance based on the recommended public transportation travel route by the Google Maps API.

7.3.3.3 For car-rental, the price returned must be calculated by multiplying estimated travel time with rental fee per unit time.

7.3.3.3.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.3.3.4 For ride-hailing, the price returned must be obtained from the ride-hailing companies' API.

7.3.3.5 For taxi, the price returned must be calculated by multiplying travel distance with taxi fare per distance.

7.3.3.5.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.3.4 Each result in the list must include the travel time of the corresponding transportation method.

7.3.4.1 The travel time in the results must be in 24 hour format.

7.3.4.2 For public transport, the time returned must be calculated based on the travel distance between the starting point and the destination.

7.3.4.2.1 The system must decide the travel distance based on the recommended public transportation travel route by the Google Maps API.

7.3.4.2.2 The result must consider the travel time of traveling to the bus stations and/or MRT stations.

7.3.4.3 For car-rental, the time returned must be calculated based on the travel distance.

7.3.4.3.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.3.4.3.2 The result must include the time taken to travel to the rental car place.

7.3.4.4 For ride-hailing, the time returned must be obtained from the ride-hailing companies' API.

7.3.4.5 For taxi, the time returned must be calculated based on the travel distance.

7.3.4.5.1 The system must decide the travel distance based on the recommended travel route by the Google Maps API.

7.3.5 Each result in the list must include the company name, which provides the corresponding transportation method.

7.4 Once the result list is returned, the User must be able to filter the results to cater to his viewing preference.

7.4.1 The User must be able to filter based on the number of passengers.

7.4.2 The User must be able to filter based on whether the mode of transport is pet-friendly.

8. Redirect to another application (Deep Linking)

8.1 The User will select their preferred mode of transportation based on the report received at the Sorting Travel Routes Page.

8.2 In the case where public transportation is selected, the User will be redirected to the Google Maps application.

8.2.1 If Google Maps is not installed, then the User will be redirected to the PlayStore page for Google Map installation

8.3 In the case where a car-rental is selected, the User will be redirected to the car rental's company application.

8.3.1 There are three different car-rental company available in the application: BlueSG, GetGo, Tribecar

8.3.2 In the case that the car rental company application is not yet installed, the User will be redirected to the PlayStore page for the chosen company's application installation

8.4 In the case where a ride-hailing is selected, the User will be redirected to the ride-hailing company application.

8.4.1 There are two different ride-hailing companies available in the application: Grab and Gojek.

8.4.2 In the case that the ride-hailing company application is not yet installed, the User will be redirected to the PlayStore page for the chosen company's application installation

8.5 In the case where a taxi is selected, the User will be redirected to the taxi company application.

8.5.1 There are three different taxi company available in the application: ComfortDelgro, SMRT Taxi, TransCab

8.5.2 In the case that the taxi's company application is not yet installed, the User will be redirected to the PlayStore page for the chosen company's application installation

9. History

9.1 The application must have a button to lead to a history page.

9.2 The user must be able to see their search history in the history page.

9.2.1 The search history must be sorted from latest to oldest.

9.2.2 The search history must include the starting location address, destination address, the date and the time that the user made the search query.

9.2.2.1 The starting location address must be in text format, and can include alphanumeric characters.

9.2.2.2 The destination address must be in text format, and can include alphanumeric characters.

9.2.2.3 The date displayed must be in DD/MM/YYYY format.

9.2.2.4 The search time displayed must be in 24 hour format.

10. Settings

- 10.1 The application must have a button to lead to a setting page.
- 10.2 The application must display the option to change language settings to one of the four main native languages in Singapore: English, Chinese, Malay and Tamil in the setting page.
 - 10.2.1 Once the language is chosen, all the text must be displayed in the chosen language.
- 10.3 The application must display the options to display distances using one of the two different units of length: US units or metric units in the setting page.
 - 10.3.1 If the User selects to display the distances using US units, the application must display the distances using US units.
 - 10.3.2 If the User selects to display the distances using metric units, the application must display the distances using metric units.
- 10.4 The application must display if the User is logged in the setting page.
 - 10.4.1 If the User is not logged in, the application must display two mutually exclusive options: “Register”, “Login”
 - 10.4.1.1 If “Register” is chosen, User must be redirected to the Registration Page
 - 10.4.1.2 If “Login” is chosen, User must be redirected to the Login Page
 - 10.4.2 If the User is logged in, the application must display three mutually exclusive options: “Change Email”, “Change Password”, “Logout”
 - 10.4.2.1 If “Change Email” is chosen, the User must be able to input their email address into an input text box to update their email.
 - 10.4.2.1.1 The application must check whether or not the User has input a valid email.
 - 10.4.2.1.1.1 If the email address is invalid, the textbox must be highlighted red, and the message “Invalid Email” must be displayed.
 - 10.4.2.1.1.2 If the email address is valid, the application must verify the authenticity of the User’s inputted email through a verification email link sent to the user.
 - 10.4.2.1.1.2.1 If the User authenticates the email address, the application must indicate to the User that the email address has been changed successfully.
 - 10.4.2.2 The User must input a 12-18 character-long alphanumeric password into an input textbox
 - 10.4.2.2.1 The User must be able to first key in the old password, followed by keying in twice the new password, in 3 different input boxes.
 - 10.4.2.2.2 The application must check whether or not the User has input a valid new password, when the User enters the new password for the first time.
 - 10.4.2.2.2.1 If the new password is invalid, the application must highlight the input box in red, and display the message “Invalid Password”.

10.4.2.2.2 If the new password is valid, the User must be able to re-enter the new password.

10.4.2.2.2.1 If the User has re-entered the new password incorrectly, the application must highlight the input box in red, and display the message “New password does not match”.

10.4.2.1.4 After the User key in both the new password and old password, the application must verify if the old password is correct.

10.4.2.1.4.1 If the old password is correct, the application must indicate to the User that the password has been changed successfully.

10.4.2.1.4.2 If the old password is incorrect, the application must indicate to the user that the old password is incorrect.

10.4.2.3 The application must provide a button for the User to logout from the account.

Non-Functional Requirements

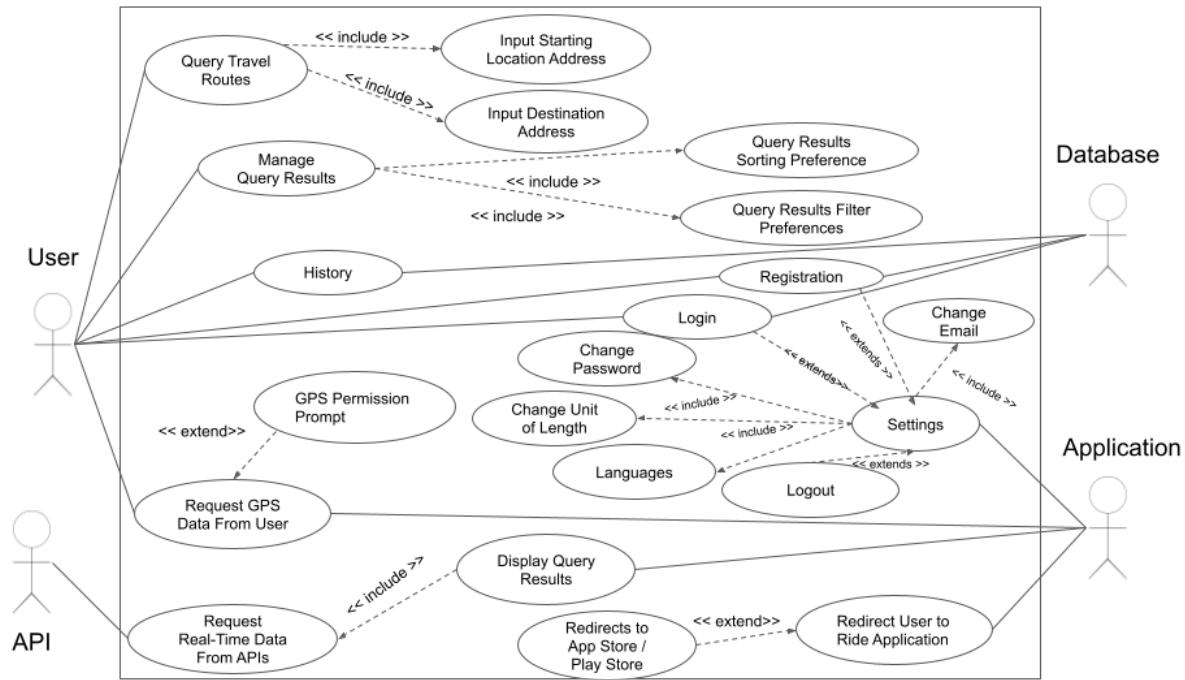
Usability	<p>User interface should be organized and simple for Users to navigate. There should be no visual clutters which makes it hard to locate the necessary buttons that the User is finding.</p> <p>Buttons should be easily recognizable</p> <ul style="list-style-type: none">- Address input text box should be accompanied with a map icon- Sorting button should have a sorting icon
Reliability	The app must be available for use 99.9% of the time, with no more than 1 hour of planned downtime per year.
Performance	<p>When a User requests a ride, the system must process the request and show the result within 30 seconds.</p> <p>The application should be able to redirect the User to the respective external application in 10 seconds.</p> <p>Location should be accurate with a maximum error of 50m.</p>
Supportability	The system must be able to support 500 Users simultaneously on the application at the same time
Scalability	More companies and other modes of travel can be integrated into the application, for future updates. In the case where the company has closed down, the application should update itself accordingly.

Data Dictionary

Term	Definition
User	An individual who uses the app to compare fares of different modes of transportation to reach their intended destination.
Starting Location	The point where the User wishes to begin his / her journey. It can either be his / her current physical location, as indicated by his / her phone GPS data, a location chosen by the User by pinpointing the location on the map or another location if he / she wants to plan in advance.
Destination	The intended location the User wishes to go. The User enters it when requesting a ride or seeking for public transport options, and it serves as the final stop. This data is used by the app to estimate the travel time and travel fare.
Travel Route	The path taken to get from the User's starting point to the intended destination. Upon gathering the User's input of the starting location and destination, the app will display a list of travel routes available. This varies between different travel routes and modes of transportation.
Travel Distance	The distance of a travel route from the User's starting location to the User's destination. This varies between different travel routes and modes of transportation.
Travel Time	The amount of time taken for the User to get from his / her starting point to his / her intended destination. This varies between different travel routes and modes of transportation. This data helps the User make an informed decision in choosing the best route.
Travel Fare	The cost incurred in the travel route between the User's starting point and intended destination. This varies between different travel routes and modes of transportation. This data helps the User to make an informed decision in choosing the best route.
Result List	Contains all the possible travel routes between the starting location and the destination.
Result Entry	One of the possible travel routes between the starting location and the destination found in the result list.

SortResult (Control Class)	This is one of the Key Control Class, which will sort the result list either by price or time.
FilterResult (Control Class)	This is one of the Key Control Class, which will filter the result list according to User's additional input. The User can filter the result list by total passenger number, or filter by whether the transportation mode is pets-friendly.

Use Case Diagram



Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Query Travel Routes		
Created By:	Koh Hao Sheng	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	User
Description:	User queries for possible travel routes.
Preconditions:	User must open the application.
Post-conditions:	Application requests for Global Positioning System (GPS) permission from User.
Flow of Events:	<ol style="list-style-type: none">Upon opening the application, it displays an animation containing the application's logo and name.Application prompts User to give GPS permission.Application displays a map and prompts User to enter a valid starting location address.Application prompts User to enter a valid destination address.
Alternative Flows:	-NIL-
Exceptions:	-NIL-
Assumptions:	-NIL-
Priority:	High

Use Case ID:	2		
Use Case Name:	Input Starting Location Address		
Created By:	Lim Dong Wan	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	23/2/2023

Actor:	User
Description:	User enters a starting location address.
Preconditions:	Application must request for GPS permission from User, and an input box must be available.
Post-conditions:	User enters a destination address.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the empty space in the input box. 2. User inputs a starting location address. 3. Application validates to see if the starting location is a valid address in Singapore. 4. If the starting location is valid, it displays another input box for the User to enter a destination address.
Alternative Flows:	2.AC.1 User exits the application by clicking on the “Back” button twice.
Exceptions:	2.EX.1 If the starting location address is invalid, it prompts the User to enter another address.
Assumptions:	-NIL-
Priority:	High

Use Case ID:	3		
Use Case Name:	Input Destination Address		
Created By:	Lim Dong Wan	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	User
Description:	User enters a destination address.
Preconditions:	User must enter a valid starting location address, and an input box must be available.
Post-conditions:	User decides how the query results should be sorted and filtered.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the empty space in the input box. 2. User inputs a destination address. 3. Application validates to see if the destination is a valid address in Singapore. 4. If the destination address is valid, it directs the User to the Query Results page.
Alternative Flows:	<p>3.AC.1 User clicks on the “Back” button. 3.AC.2 The application directs the User to the Input Starting Location Address page.</p>
Exceptions:	3.EX.1 If the destination address is invalid, it prompts the User to enter another address.
Assumptions:	-NIL-
Priority:	High

Use Case ID:	4		
Use Case Name:	Manage Query Results		
Created By:	Lim Boon Hian	Last Updated By:	Lim Dong Wan
Date Created:	11/2/2023	Date Last Updated:	19/2/2023

Actor:	User
Description:	User manages query results displayed in the application interface.
Preconditions:	User must enter a valid starting location address and destination address, and the application must redirect User to the Query Results page.
Post-conditions:	User decides how the query results should be sorted and filtered.
Flow of Events:	<ol style="list-style-type: none"> Upon entering a valid starting location address and destination address, the application redirects User to the Query Results page. Application requests data from various APIs, and displays a list of query results, sorted based on the cheapest travel fare by default. User decides whether to sort the results based on travel time or travel fare. User decides whether to filter the results based on the number of passengers traveling, or if the mode of transport is wheelchair-accessible or pet-friendly.
Alternative Flows:	-NIL-
Exceptions:	-NIL-
Assumptions:	-NIL-
Priority:	High

Use Case ID:	5		
Use Case Name:	Query Results Sorting Preference		
Created By:	Lim Boon Hian	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	User
Description:	User decides how the query results should be sorted.
Preconditions:	User must enter a valid starting location address and destination address, and the application must redirect User to the Query Results page.
Post-conditions:	Application requests data from various APIs to provide the cheapest or fastest route as the first option in the list of query results for the requested trip, based on the User's sorting preference.
Flow of Events:	<ol style="list-style-type: none"> 1. Application requests data from various APIs, and displays a list of query results, sorted based on the cheapest travel fare by default. 2. The User can toggle between the 2 buttons - “Cheapest” and “Fastest” - to display the query result based on travel fare and travel time respectively. <ol style="list-style-type: none"> a. If the button displays “Fastest” when the User clicks on it, the application sorts the query results based on the fastest travel time. b. If the button displays “Cheapest” when the User clicks on it, the application sorts the query results based on the cheapest travel fare.
Alternative Flows:	5.AC.1 User clicks on the “Filter” button, which enables the User to input extra filter requirements for the query results, so as to trim the list of query results.
Exceptions:	-NIL-
Assumptions:	The application can always find an available transportation mode, as long as both the starting location and destination addresses are valid Singapore addresses.
Priority:	High

Use Case ID:	6		
Use Case Name:	Query Results Filter Preference		
Created By:	Lim Boon Hian	Last Updated By:	Lim Dong Wan
Date Created:	11/2/2023	Date Last Updated:	19/2/2023

Actor:	User
Description:	User decides how the query results should be filtered.
Preconditions:	User must click on the “Filter” button.
Post-conditions:	Application requests data from various APIs, based on the extra filter requirements requested by the User, and updates the query results accordingly.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the “Filter” button. 2. Application displays an extra query menu as a pop-up. 3. User decides whether to indicate the number of passengers who will be traveling, or their preference for a pet-friendly or wheelchair-accessible transportation mode, in the query menu. <ol style="list-style-type: none"> a. If the User indicates the number of passengers, the application updates the query result list with travel options that are based on the indicated number of travelers. b. If the User indicates their preference for a pet-friendly transportation mode, the application updates the result list with pet-friendly transportation modes. c. If the User indicates their preference for a wheelchair-accessible transportation mode, the application updates the result list with wheelchair-accessible transportation modes.
Alternative Flows:	<p>6.AC.1 User drags and closes the query menu while it is active.</p> <p>6.AC.2 If the User does not indicate how many passengers will be traveling, the application updates the query result list, such that all the results are based on a single passenger traveling, by default.</p> <p>6.AC.3 If the User indicates no preference for pets-friendly transportation mode, the application updates the query result list, such that all the results are either pet-friendly or non-pet friendly transportation modes.</p> <p>6.AC.4 If the User indicates no preference for wheelchair-accessible transportation modes, the application updates the query result list, such that all the results are either wheelchair-accessible or non-wheelchair accessible transportation modes.</p>
Exceptions:	-NIL-

Assumptions:	-NIL-
Priority:	High

Use Case ID:	7		
Use Case Name:	Request GPS Data From User		
Created By:	Ng Woon Yee	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	User, Application
Description:	Application requests GPS access from the User in order to obtain the User's current location for the purpose of inputting a starting location address for a trip.
Preconditions:	User must open the application.
Post-conditions:	User enters a starting location address to start querying for a trip.
Flow of Events:	<ol style="list-style-type: none"> 1. User is currently in the Landing page of the application. 2. Application checks if the User has granted GPS access. 3. If the User has granted GPS access, the application retrieves the User's current location. 4. Application redirects the User to the Input Starting Location Address page.
Alternative Flows:	7.AC.1 If the User has not granted GPS access, the application raises an exception and prompts the User to give GPS permission to the application. If the User denies GPS permission, the application provides alternative means for the User to input address information.
Exceptions:	-NIL-
Assumptions:	User's phone has a GPS installed.
Priority:	High

Use Case ID:	8		
Use Case Name:	GPS Permission Prompt		
Created By:	Ng Woon Yee	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	User, Application
Description:	Application prompts the User to grant GPS permission, in the event that permission has not been granted.
Preconditions:	User has not granted GPS Permission to the application.
Post-conditions:	User enters a starting location address to start querying for a trip.
Flow of Events:	<ol style="list-style-type: none"> 1. Application displays a pop-up window to prompt the User to grant GPS access. 2. In the pop-up window, there is a link directing the User to the Setting App's Permission page. 3. User decides whether to grant GPS access. 4. If the User grants GPS access, the application retrieves the User's current location. 5. Application redirects the User to Input Starting Location Address Page, as granting GPS permission is optional for the User.
Alternative Flows:	8.AC.1 If the User denies GPS access, the application must allow the User to input address information through input box.
Exceptions:	-NIL-
Assumptions:	User's phone has a GPS installed.
Priority:	High

Use Case ID:	9		
Use Case Name:	Display Query Results		
Created By:	Loke Yong Jian	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	Application, API
Description:	Application displays a list of query results returned by the various APIs, based on the User's inputs of starting location and destination addresses.
Preconditions:	User must enter a valid starting location address and destination address.
Post-conditions:	Application displays a list of query results, sorted based on the cheapest travel fare by default.
Flow of Events:	<ol style="list-style-type: none"> 1. User enters valid starting location and destination addresses in the input box. 2. Application extracts the address information and query the result through the various APIs. 3. Application redirects the User to the Query Results page, which displays a list of query results based on the User's requested trip.
Alternative Flows:	-NIL-
Exceptions:	9.EX.1 If the connection between the application and API is not successful, the application prompts the User to try again later or choose another starting location and destination.
Assumptions:	-NIL-
Priority:	High

Use Case ID:	10		
Use Case Name:	Request Real-Time Data From APIs		
Created By:	Ng Woon Yee	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	Application, API
Description:	Application requests data from API to display a list of query results.
Preconditions:	User must input valid starting location and destination addresses and click the "Search" button.
Post-conditions:	Application displays a list of query results, sorted based on the cheapest travel fare by default.
Flow of Events:	<ol style="list-style-type: none"> 1. Application has the addresses of the starting location and destination inputted by the User. 2. Application sends a GET request to the API to obtain relevant information, including travel time, estimated price, and route details. 3. API responds to the application's request and sends back the relevant information. 4. Application redirects the User to the Query Results page, which displays a list of query results based on the User's requested trip.
Alternative Flows:	10.AC.1 In the case of no returned result, the application displays a "No available route right now, please try again later." message.
Exceptions:	-NIL-
Assumptions:	The Ride-Hailing and Google Maps APIs respond correctly and successfully to the application's requests.
Priority:	High

Use Case ID:	11		
Use Case Name:	Redirect User to Ride Application		
Created By:	Lim Boon Hian	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	Application
Description:	Application redirects the User to the corresponding ride application, based on the User's preferred transportation mode.
Preconditions:	User must select a transportation mode (result entry) from the query result list.
Post-conditions:	Application redirects the User to the said ride application.
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the preferred transportation mode from the result list. 2. Application invokes the corresponding ride application based on the transportation mode that the User has selected. <ol style="list-style-type: none"> a. If car-rental is selected, the application redirects the User to the car rental company's applications. b. If ride-hailing is selected, the application redirects the User to the ride-hailing company's applications. 3. The corresponding ride application will be opened for the User if the said application is installed in the User's device.
Alternative Flows:	-NIL-
Exceptions:	11.EX.1 If the corresponding ride application is not installed in the User's device, the application will lead the User to the device's application store (e.g. App Store, Play Store).
Assumptions:	User has the corresponding ride application installed on the User's device. User also chooses either car-rental or ride-hailing as their preferred transportation mode.
Priority:	Medium

Use Case ID:	12		
Use Case Name:	Redirect User to Application Store		
Created By:	Lim Boon Hian	Last Updated By:	Lim Dong Wan
Date Created:	8/2/2023	Date Last Updated:	19/2/2023

Actor:	Application
Description:	Application redirects the User to the corresponding application store, based on the device the User is using.
Preconditions:	Application must invoke the corresponding ride application based on the transportation mode the User has selected, but the said ride application is not installed on the User's device.
Post-conditions:	Application redirects the User to the said application store, prompting the User to install the said ride application.
Flow of Events:	<ol style="list-style-type: none"> 1. User selects the preferred transportation mode from the result list. 2. Application invokes the corresponding ride application based on the transportation mode that the User selected. <ol style="list-style-type: none"> a. If car-rental is selected, the application redirects the User to the car rental company's applications. b. If ride-hailing is selected, the application redirects the User to the ride-hailing company's applications. 3. Application detects that the corresponding ride application is not installed on the User's device. 4. Application redirects the User to the device's application store (e.g. App Store, Play Store).
Alternative Flows:	-NIL-
Exceptions:	-NIL-
Assumptions:	User does not have the corresponding ride application installed on the User's device.
Priority:	Medium

Use Case ID:	13		
Use Case Name:	Search History		
Created By:	Lim Dong Wan	Last Updated By:	Lim Dong Wan
Date Created:	11/2/2023	Date Last Updated:	19/2/2023

Actor:	User, Database
Description:	User views his / her past trip searches using the application.
Preconditions:	Application must have a “History” button available in the interface.
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the “History” button. 2. Application displays the User’s search history, sorted from latest to oldest. 3. Application displays information like starting location address, destination address, date and time that the User made the search query, in each entry of the search history.
Alternative Flows:	13.AC.1 Application displays an empty search history if the User has not made any search queries in the application yet.
Exceptions:	-NIL-
Assumptions:	-NIL-
Priority:	Medium

Use Case ID:	14		
Use Case Name:	Registration		
Created By:	Ng Woon Yee	Last Updated By:	Lim Dong Wan
Date Created:	10/2/2023	Date Last Updated:	19/2/2023

Actor:	User, Database
Description:	User registers for an account on the application to store their personal information.
Preconditions:	User must have a valid email address, and the "Register" button must be available in the Landing page.
Post-conditions:	User has a registered account on the application and proceeds to enter a starting location address to start querying for a trip.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the "Register" button in the Landing page and is redirected to the Registration page. 2. User provides a valid email address and creates a password with 12-18 characters, including alphanumeric characters. 3. User submits the registration information. 4. Application verifies the email address and password format inputted by the User. 5. If the information is valid, the application creates a User account. 6. Application notifies the User of successful registration. 7. Application redirects User to the Input Starting Location Address page.
Alternative Flows:	-NIL-
Exceptions:	<p>14.EX.1 If the email address is already registered, the application displays an error message and prompts the User to try again with a different email address.</p> <p>14.EX.2 If the password does not meet the required format, the application displays an error message and prompts the User to try again with a password that meets the requirements.</p>
Assumptions:	User has access to a valid email address.
Priority:	Medium

Use Case ID:	15		
Use Case Name:	Login		
Created By:	Ng Woon Yee	Last Updated By:	Lim Dong Wan
Date Created:	10/2/2023	Date Last Updated:	19/2/2023

Actor:	User, Database
Description:	User logs in to their account on the application.
Preconditions:	User must have a pre-existing account on the application, and the "Login" button must be available in the Landing page.
Post-conditions:	User has access to their personal information stored on the application, and proceeds to enter a starting location address to start querying for a trip.
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the "Login" button on the landing page. 2. User is redirected to the login page. 3. User must provide their registered email address and password. 4. User submits the login information. 5. Application verifies if the email address and password match a registered account. 6. If the information is valid, the application grants the User access to their account. 7. User is redirected to the Starting Location Input Page.
Alternative Flows:	-NIL-
Exceptions:	15.EX.1 If the email address and password do not match a registered account, the application displays an error message and prompts the User to try again.
Assumptions:	-NIL-
Priority:	Medium

Use Case ID:	16		
Use Case Name:	Settings		
Created By:	Lim Boon Hian	Last Updated By:	Lim Dong Wan
Date Created:	19/2/2023	Date Last Updated:	23/2/2023

Actor:	User
Description:	Application must have a “Settings” icon in the interface.
Preconditions:	-NIL-
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the Settings icon. 2. Application directs the User to Settings page. 3. The application must display the option to change language settings. 4. The application must display the options to display distances using different metric units. 5. If the User is not logged in, the application must display two mutually exclusive options: “Register”, “Login” <ol style="list-style-type: none"> a. If “Register” is chosen, User must be redirected to the Registration page b. If “Login” is chosen, User must be redirected to the Login page 6. If the User is logged in, the application must display three mutually exclusive options: “Change email”, “Change password”, “Logout”.
Alternative Flows:	16.AC.1 After the user clicks the return button, the user must be able to return to the previous page they are browsing.
Exceptions:	-NIL-
Assumptions:	Application must have a “Settings” icon in the interface, which redirects the User to the settings page.
Priority:	Medium

Use Case ID:	17		
Use Case Name:	Select Preferred Language		
Created By:	Lim Dong Wan	Last Updated By:	Lim Dong Wan
Date Created:	10/2/2023	Date Last Updated:	23/2/2023

Actor:	User
Description:	User selects the preferred language for the application.
Preconditions:	Application must have a “Settings” icon in the interface and a “Select Language” button in the Settings page.
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the Settings icon. 2. Application directs the User to Settings page. 3. User clicks on the “Select Language” button. 4. Application displays 4 different language buttons - “English”, “Chinese”, “Malay” and “Tamil”. 5. User decides and clicks on the desired language button. 6. Application displays all text in the language chosen by User.
Alternative Flows:	17.AC.1 If no language is chosen by the User, the application displays all text in English, by default.
Exceptions:	-NIL-
Assumptions:	User is able to read and understand at least 1 of the 4 languages provided by the application.
Priority:	Medium

Use Case ID:	18		
Use Case Name:	Change Email Address		
Created By:	Lim Dong Wan	Last Updated By:	Lim Dong Wan
Date Created:	20/2/2023	Date Last Updated:	23/2/2023

Actor:	User
Description:	User changes the email address used in registering for an account to another valid email address.
Preconditions:	Application must have a “Settings” icon in the interface and a “Change Email” button in the Settings page.
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the Settings icon. 2. Application directs the User to Settings page. 3. User clicks on the “Change Email” button. 4. Application directs the User to the Change Email page. 5. User enters into an empty input text box a different email address that is not currently used for the User account. 6. Application validates if the User has entered a valid email address. 7. If the new email address provided by the User is valid, the application verifies the authenticity of the email address by sending a verification email link to the User. 8. Once the User accepts the verification of the new email address, the application updates the User’s account profile with the new email address.
Alternative Flows:	18.AC.1 User clicks on the “Back” button. 18.AC.2 The application directs the User back to the Settings page.
Exceptions:	18.EX.1 If the new email address provided by the User is invalid, the application highlights the input text box in red and displays an “Invalid Email” message.
Assumptions:	User has a pre-existing account with the application, and has another valid and usable email address.
Priority:	Medium

Use Case ID:	19		
Use Case Name:	Change Password		
Created By:	Lim Dong Wan	Last Updated By:	Lim Dong Wan
Date Created:	20/2/2023	Date Last Updated:	23/2/2023

Actor:	User
Description:	User changes the password used in registering for an account to another password.
Preconditions:	Application must have a “Settings” icon in the interface and a “Change Password” button in the Settings page.
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the Settings icon. 2. Application directs the User to Settings page. 3. User clicks on the “Change Password” button. 4. Application directs the User to the Change Password page. 5. User enters into the first empty input text box the current old password used for the User account. 6. User enters into the second empty input text box a new password, which must be of 12-18 alphanumeric characters, for the account. 7. Application validates if the User has entered a valid new password. 8. If the User has provided a valid new password, the application prompts the User to re-enter the new password. 9. Application validates if the User has correctly re-entered the new password. 10. If the User has correctly re-entered the new password, the application validates if the User has entered the old password correctly. 11. If the User has entered the old password correctly, the application prompts the User that password has been changed successfully.
Alternative Flows:	<p>19.AC.1 User clicks on the “Back” button.</p> <p>19.AC.2 The application directs the User back to the Settings page.</p>
Exceptions:	<p>19.EX.1 If the User has provided an invalid new password, the application highlights the input text box in red and displays an “Invalid Password” message.</p> <p>19.EX.2: If the User has re-entered the new password incorrectly, the application highlights the input text box in red and displays a “New password does not match” message.</p> <p>19.EX.3 If the User has entered the old password incorrectly, the application prompts the User that the User has entered the old password incorrectly.</p>

Assumptions:	User has a pre-existing account with the application.
Priority:	Medium

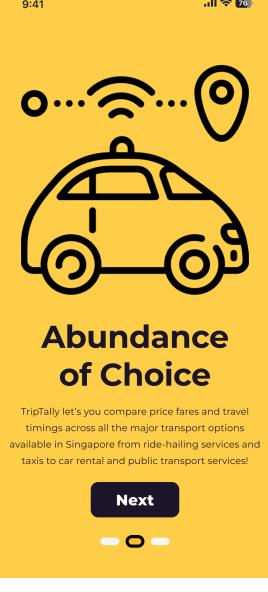
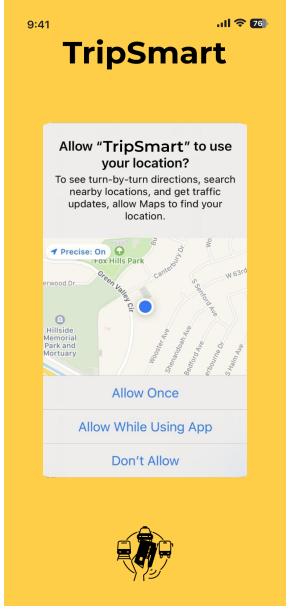
Use Case ID:	20		
Use Case Name:	Change Unit of Length		
Created By:	Ng Woon Yee	Last Updated By:	Ng Woon Yee
Date Created:	22/2/2023	Date Last Updated:	22/2/2023

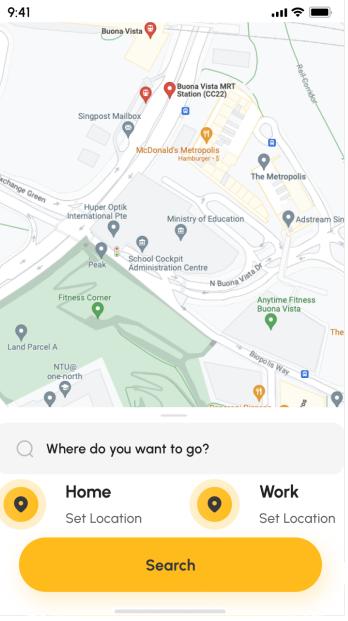
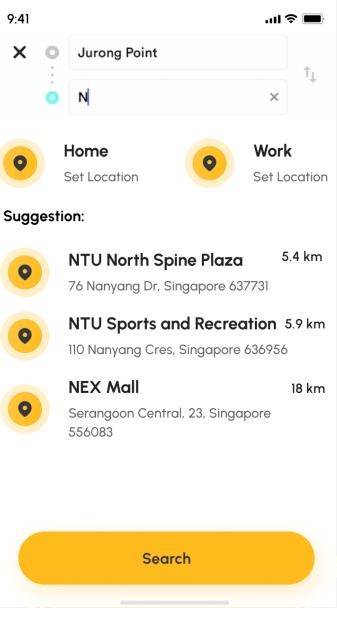
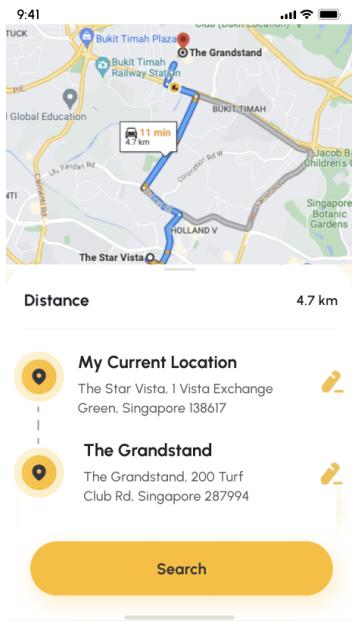
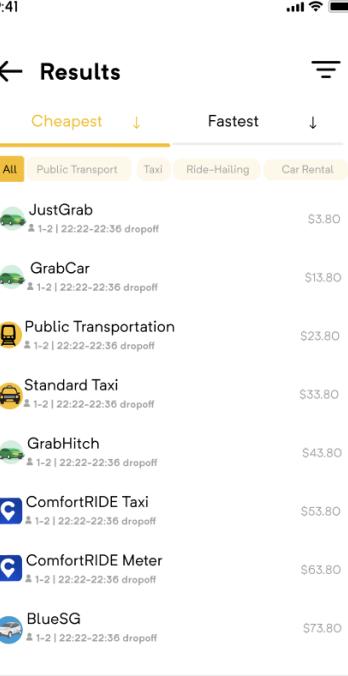
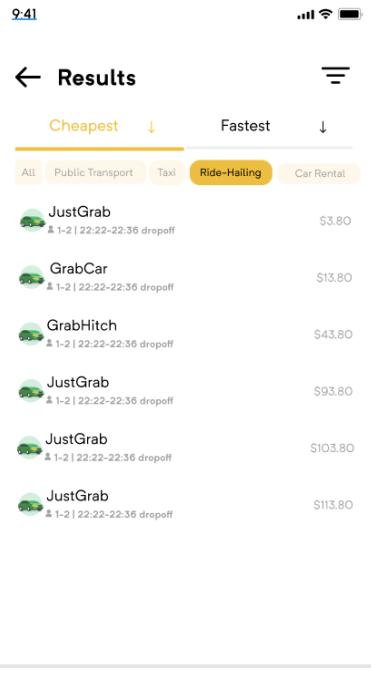
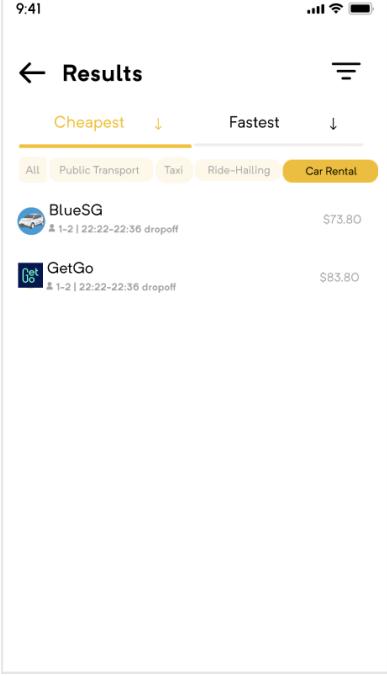
Actor:	User
Description:	User changes the unit of distance, either to US units or to metric units in the setting page.
Preconditions:	Application must have a “Settings” icon in the interface and a “Change Unit of Length” button in the Settings page.
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the Settings icon. 2. Application directs the User to Settings page. 3. User clicks on the “Change Unit of Length” button. 4. Application displays 2 different unit buttons - “US units” and “Metric units”. 5. User decides and clicks on the desired unit button. 6. Application displays distance of travel in the unit chosen by User.
Alternative Flows:	<p>20.AC.1 User clicks on the “Back” button. 20.AC.2 The application directs the User back to the Settings page.</p>
Exceptions:	-NIL-
Assumptions:	-NIL-
Priority:	Medium

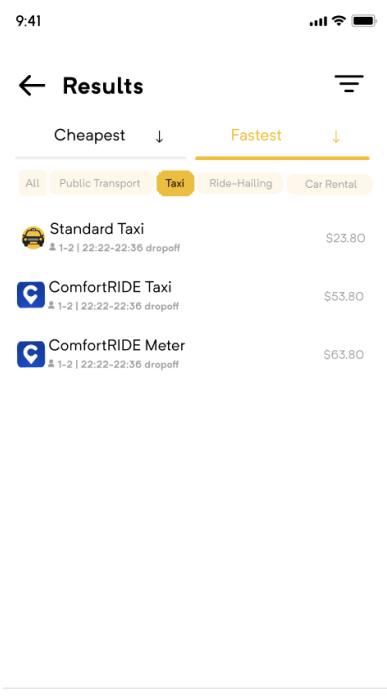
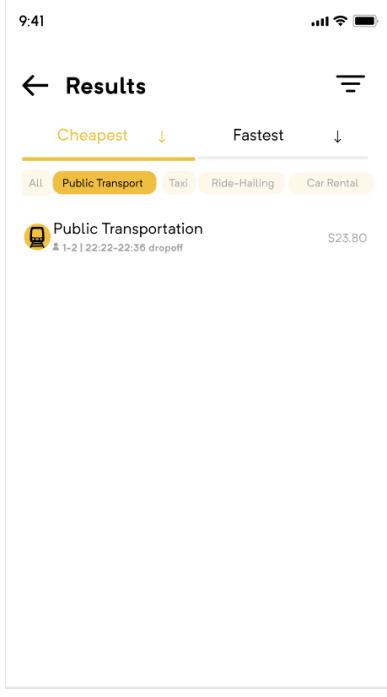
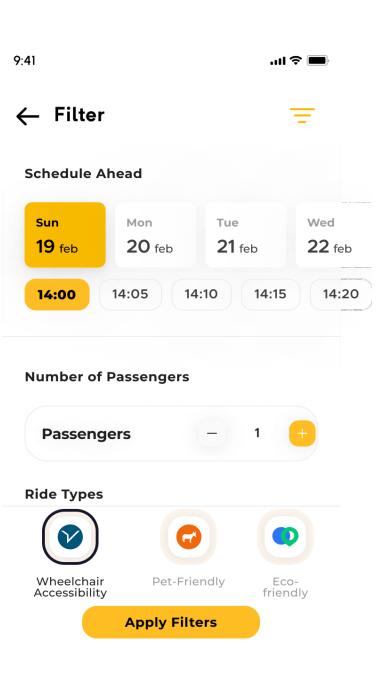
Use Case ID:	21		
Use Case Name:	Logout		
Created By:	Ng Woon Yee	Last Updated By:	Ng Woon Yee
Date Created:	22/2/2023	Date Last Updated:	22/2/2023

Actor:	User
Description:	User ends the session by logging out of their account.
Preconditions:	Users must have logged into their own account within the application.
Post-conditions:	-NIL-
Flow of Events:	<ol style="list-style-type: none"> 1. User clicks on the Settings icon. 2. Application directs the User to Settings page. 3. User clicks on the “Logout” button. 4. Application prompts binary choices questions(yes or cancel) to double confirm user’s intention to logout. 5. User logs out.
Alternative Flows:	21.AC.1 User clicks on the “Back” button. 21.AC.2 The application directs the User back to the Settings page.
Exceptions:	-NIL-
Assumptions:	User has a pre-existing account with the application.
Priority:	Medium

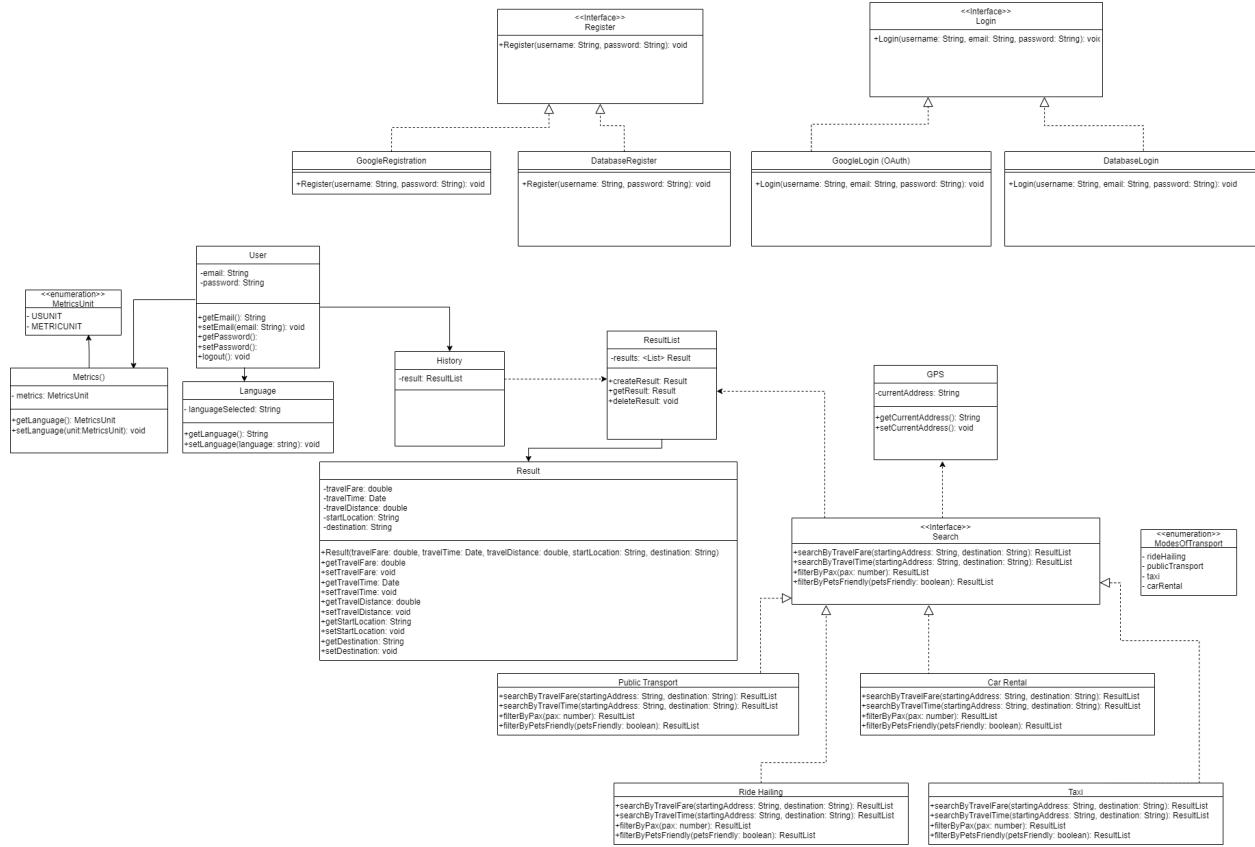
Initial UI Mockup Diagram

		
<p>Startup Page of the Application</p>	<p>Welcome Page of the Application (1)</p>	<p>Welcome Page of the Application (2)</p>
		
<p>Welcome Page of the Application (3)</p>	<p>Page where application requests User for GPS access</p>	<p>Landing page to allow User to continue as guest (Could look into expanding into a login/sign up page)</p>

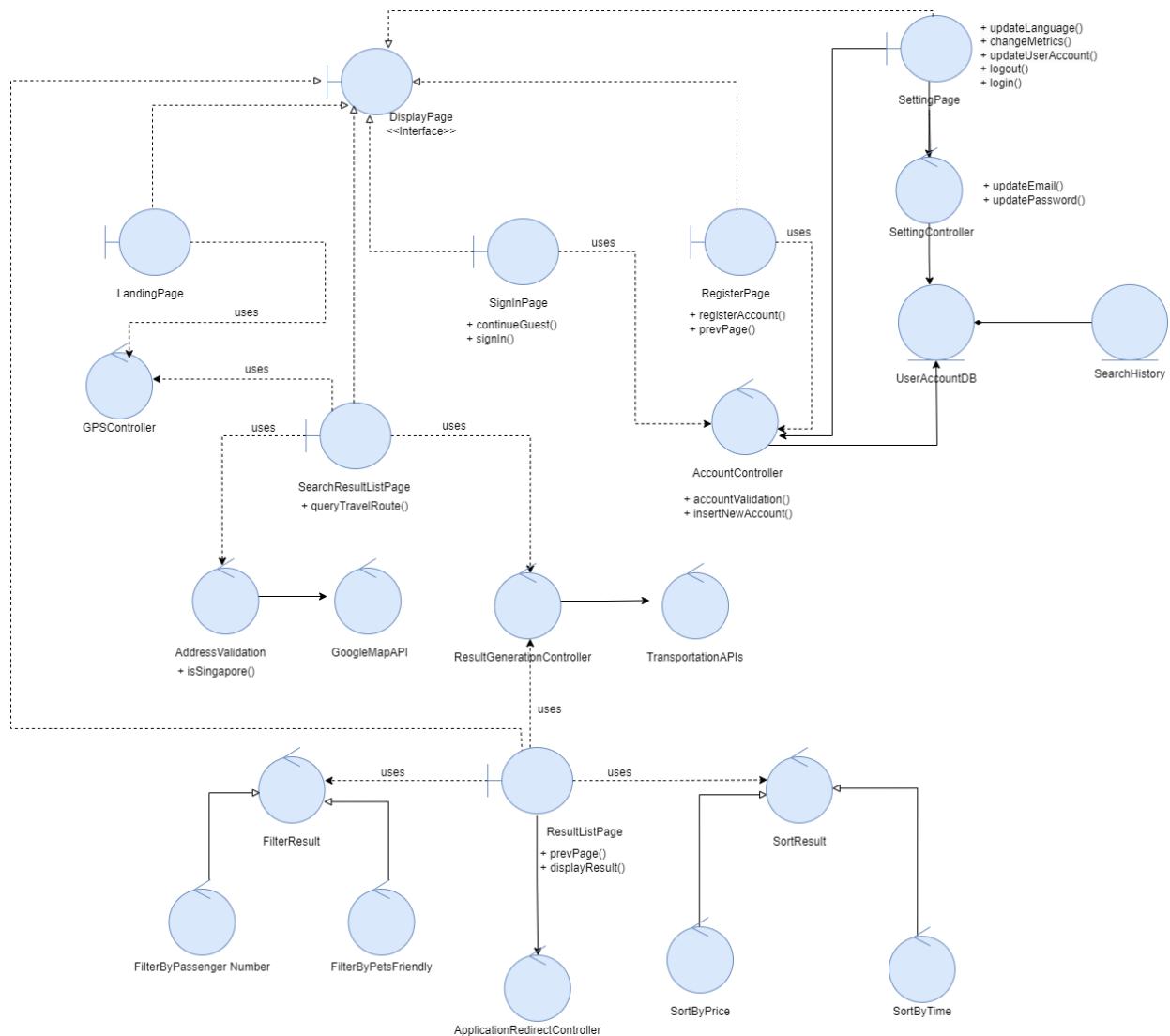
		
<p>Default page for User to type in the destination</p>	<p>Application showing destinations when locations are inputted</p>	<p>Application showing the distance between starting location and destination</p>
		
<p>Default Result Page</p> <ul style="list-style-type: none"> - All transportation mode will be shown 	<p>Result Page</p> <ul style="list-style-type: none"> - User choose to filter out the Ride-Hailing option 	<p>Result Page</p> <ul style="list-style-type: none"> - User choose to filter out the Car-Rental

<ul style="list-style-type: none"> - Order will shown with the lowest price first 	only	option only									
 <p>9:41</p> <p>← Results</p> <p>Cheapest ↓ Fastest ↓</p> <p>All Public Transport Taxi Ride-Hailing Car Rental</p> <p>Standard Taxi \$23.80 1-2 22:22-22:36 dropoff</p> <p>ComfortRIDE Taxi \$53.80 1-2 22:22-22:36 dropoff</p> <p>ComfortRIDE Meter \$63.80 1-2 22:22-22:36 dropoff</p>	 <p>9:41</p> <p>← Results</p> <p>Cheapest ↓ Fastest ↓</p> <p>All Public Transport Taxi Ride-Hailing Car Rental</p> <p>Public Transportation \$23.80 1-2 22:22-22:36 dropoff</p>	 <p>9:41</p> <p>← Filter</p> <p>Schedule Ahead</p> <table border="1"> <tr> <td>Sun 19 feb</td> <td>Mon 20 feb</td> <td>Tue 21 feb</td> <td>Wed 22 feb</td> </tr> <tr> <td>14:00</td> <td>14:05</td> <td>14:10</td> <td>14:15</td> <td>14:20</td> </tr> </table> <p>Number of Passengers</p> <p>Passengers - 1 +</p> <p>Ride Types</p> <p>Wheelchair Accessibility Pet-Friendly Eco-friendly</p> <p>Apply Filters</p>	Sun 19 feb	Mon 20 feb	Tue 21 feb	Wed 22 feb	14:00	14:05	14:10	14:15	14:20
Sun 19 feb	Mon 20 feb	Tue 21 feb	Wed 22 feb								
14:00	14:05	14:10	14:15	14:20							
<p>Result Page</p> <ul style="list-style-type: none"> - User choose to filter out the Taxi option only 	<p>Result Page</p> <ul style="list-style-type: none"> - User choose to filter out the Public Transport option only 	<p>Filter Page</p> <ul style="list-style-type: none"> - User can further filter their results according to their needs (i.e. schedule a commute for later or specifying the number of passengers in the trip etc.) 									

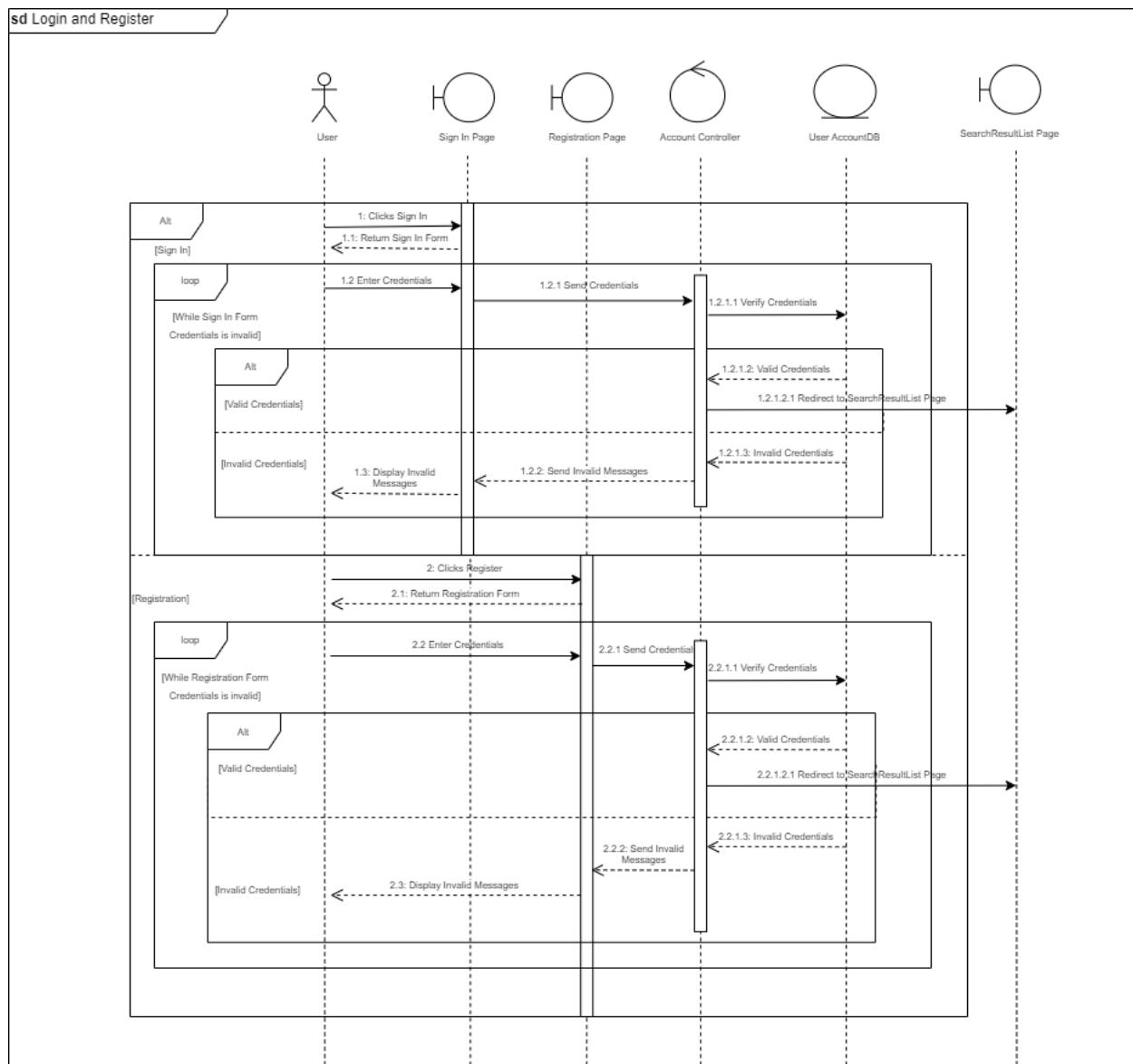
Class diagram of entity classes

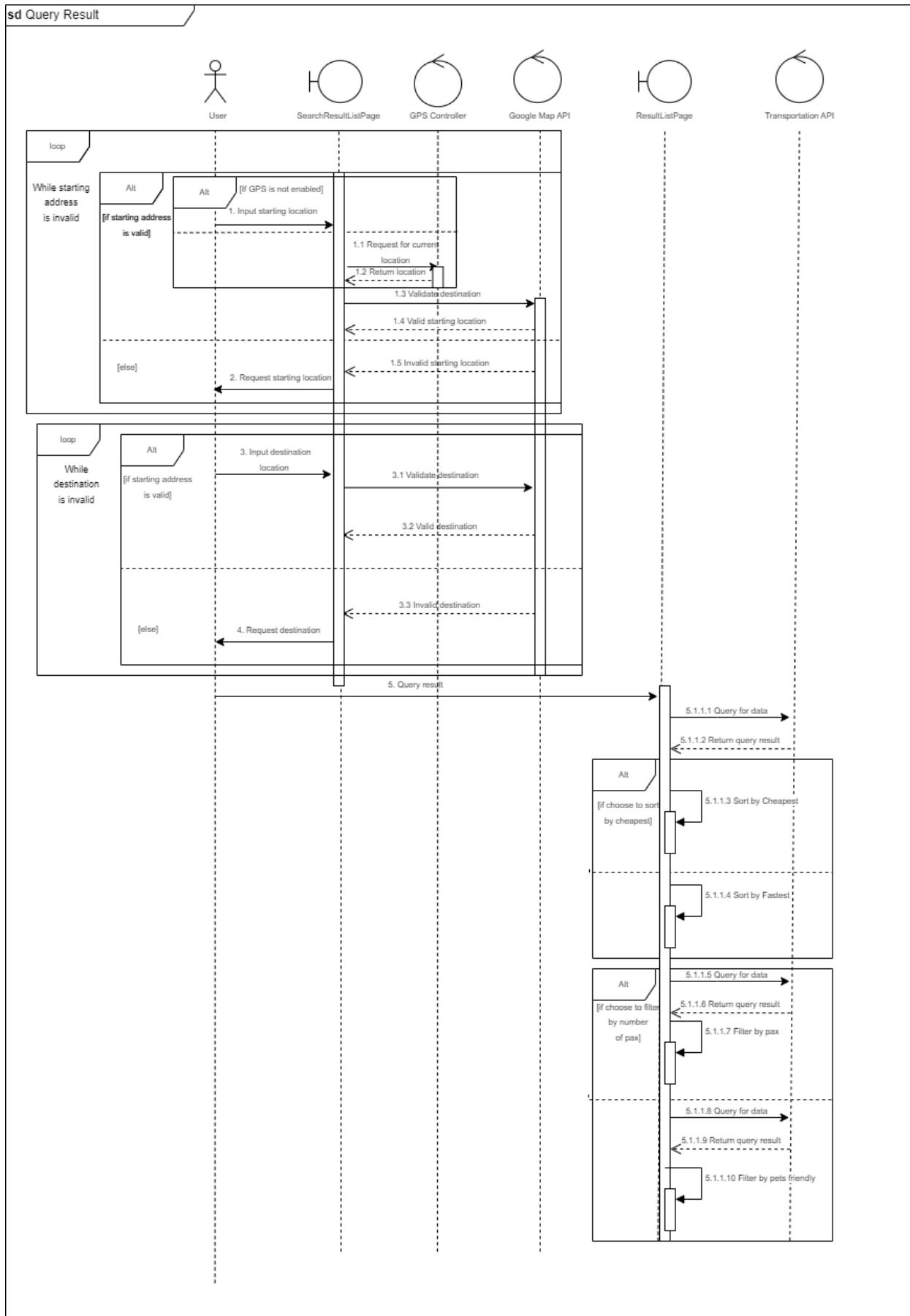


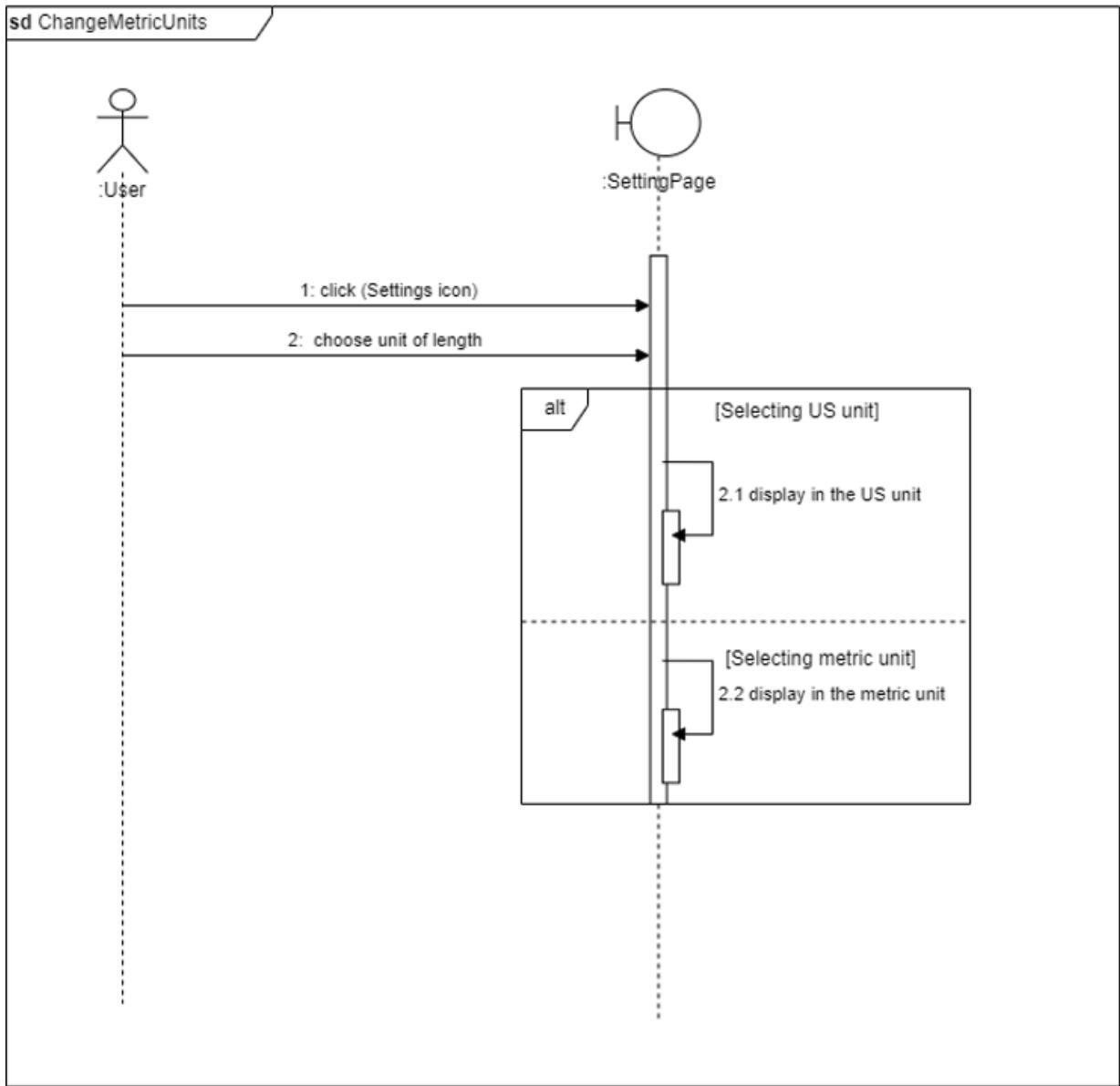
Key boundary classes and control classes

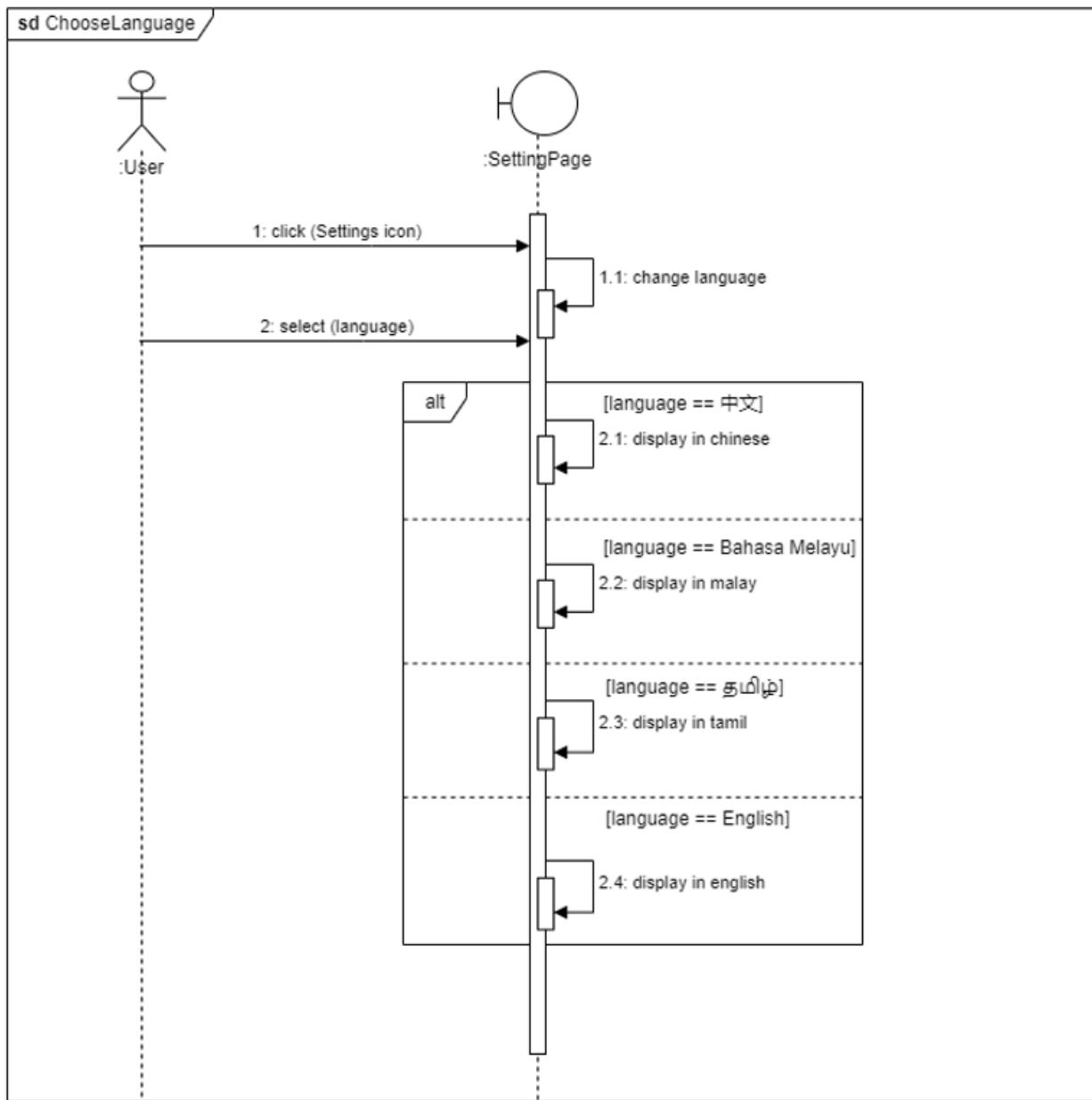


Sequence diagrams of some use cases









Initial Dialog map

