

SC2006 – Software Engineering

Software Requirements Specification

for

TaxiNow

Prepared by:

Ronald Teo Boon Keat	U2220156E
Toh Si En Ernneth	U2221199F
Vithun Natesh Balakathiravan	U2221650C
Soo Qi Yang	U2221761L
Tan Jing Jie	U2221344L
Sushmita Ramaratnam	U2222958B

Group TREE

14 April 2024

TABLES OF CONTENT

1. Introduction.....	4
1.1 Purpose.....	4
1.2 Intended Audience and Reading Suggestions.....	4
1.3 Product Scope.....	4
2. Overall Description.....	5
2.1 Product Perspective.....	5
2.1.1 Use Case Diagram.....	5
2.2 Product Functions.....	6
2.2.1 Class Diagram.....	7
2.2.2 Dialog Map.....	8
2.2.3 System Architecture.....	10
2.3 User Classes and Characteristics.....	11
2.4 Operating Environment.....	11
2.5 Design and Implementation Constraints.....	11
2.6 Assumptions and Dependencies.....	11
3. External Interface Requirements.....	12
3.1 User Interfaces.....	12
4. System Features.....	15
4.1 Functional Requirements.....	15
4.2 Use Case Description.....	17
4.3 Sequence Diagrams.....	29
4.3.1 Setting User Location On Map.....	29
4.3.2 Map Interactions.....	30
4.3.3 Generating Recommendation.....	31
5. Other Non-Functional Requirement.....	32
5.1 Performance Requirements.....	32
5.1.1 Maintainability and Platform.....	32
5.1.2 Performance.....	32
5.2 Security Requirements.....	33
5.2.1 User Permissions Security.....	33
5.2.2 Data Minimisation.....	33
5.3 Software Quality Attributes.....	33
5.3.1 Reliability.....	33
5.3.2 Usability.....	33
6. Testing.....	34
6.1 Blackbox Testing.....	34
6.1.1 Converting Street Name to Coordinates.....	34
6.1.2 Get Taxi Coordinates Around User.....	35

6.2 Whitebox Testing.....	37
6.2.1 Getting Recommended Locations.....	37
6.2.2 Getting Coordinates of Taxis within 5km radius of a given latitude and longitude.	39
6.2.3 State transition of Default, GPS and Manual Location.....	41

1. Introduction

1.1 Purpose

TaxiNow is a platform-independent mobile application that provides everyone with real-time traffic information about the availability of taxis in the vicinity of the user, in Singapore. TaxiNow includes all taxi companies in Singapore, which do not include private-hire companies. Moving forward with technological advancements and Singapore's transition towards a smart nation, TaxiNow utilises APIs and a range of public data to deliver an innovative, interactive and interesting application that will undeniably improve the productivity and quality of life of people residing in and visiting Singapore.

1.2 Intended Audience and Reading Suggestions

This document is intended for developers, project managers, marketing staff, testers, and documentation writers. While the app was being developed, developers and documentation writers constantly updated the information here such that future developers can understand and expand on the preexisting code. This documentation was also used by project managers to monitor the status of the app development, marketing staff to find unique features that are advertisable, and testers to test all the different features of the software.

This SRS contains functional and non-functional requirements of our application, a use case diagram with use case descriptions and sequence diagrams for each use case, the class diagram, a dialog map, system architecture, and our testing cases and results. Readers should follow the sequence of the document and read it in its entirety at least once before focusing on the respective sections that interest them.

1.3 Product Scope

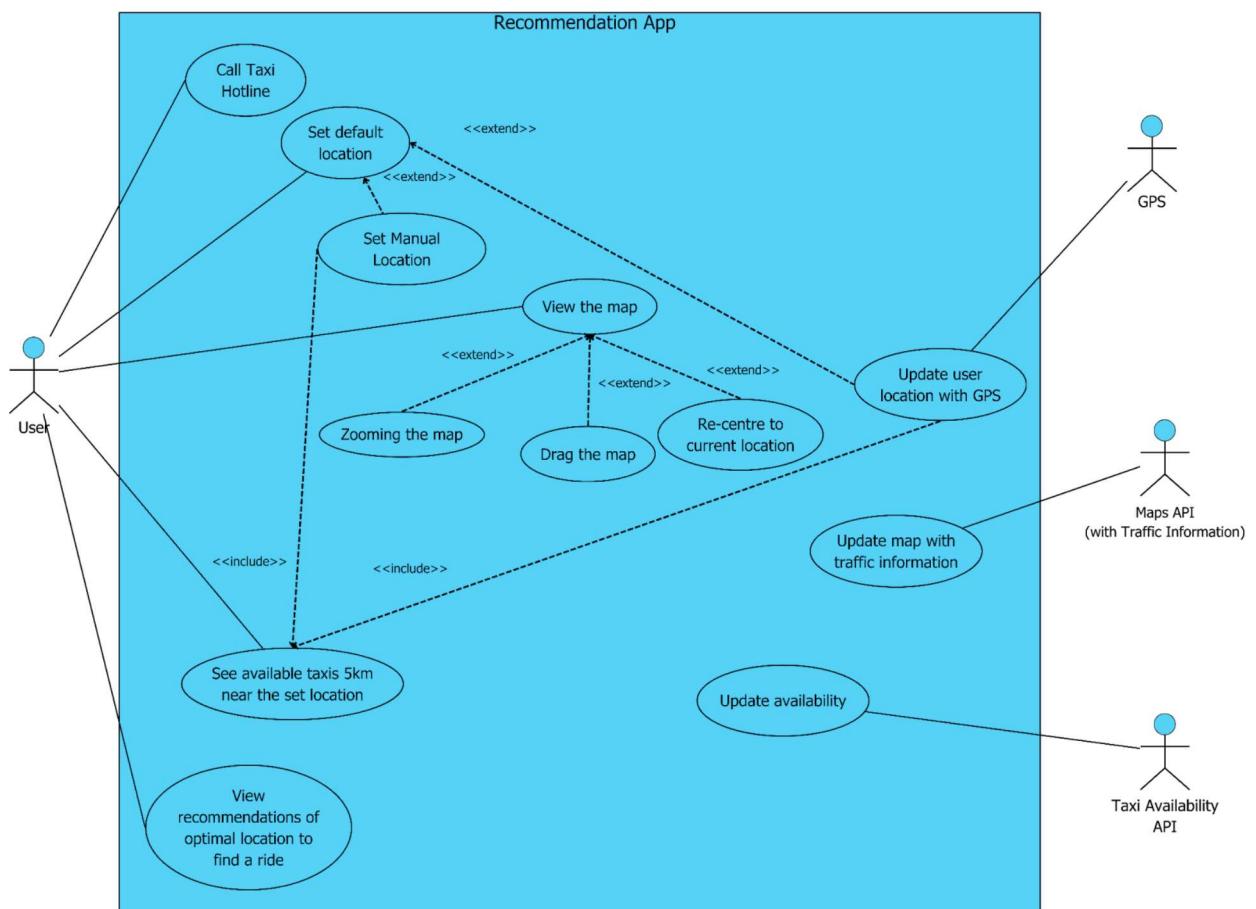
Private-hire vehicles often come at a high cost and users often take a long time finding a ride to their destination. Taxis are more economical, and are often used as a simpler and faster alternative to private-hire applications, without the need to register for an account or key in sensitive credit card details. Taxis are usually the prime choice for tourists visiting Singapore for a short term due to its simplicity. However, where do we go to ensure we can successfully flag a taxi? This is where TaxiNow steps in to solve all uncertainties, providing everyone with accurate real-time information of the current traffic conditions and available taxis within the near vicinity of them, providing them with the optimal and recommended location to flag for a taxi. With TaxiNow, everyone would be able to find the best location to flag a taxi, for higher cost-savings and time-savings without the need to register for an account.

2. Overall Description

2.1 Product Perspective

TaxiNow fuses the features of taxi availability, a map interface, and a location recommendation feature that recommends users the optimal location to hire a taxi to get a cheaper fare amidst high demand situations or in cases of unable to hail a ride as there are no taxis nearby.

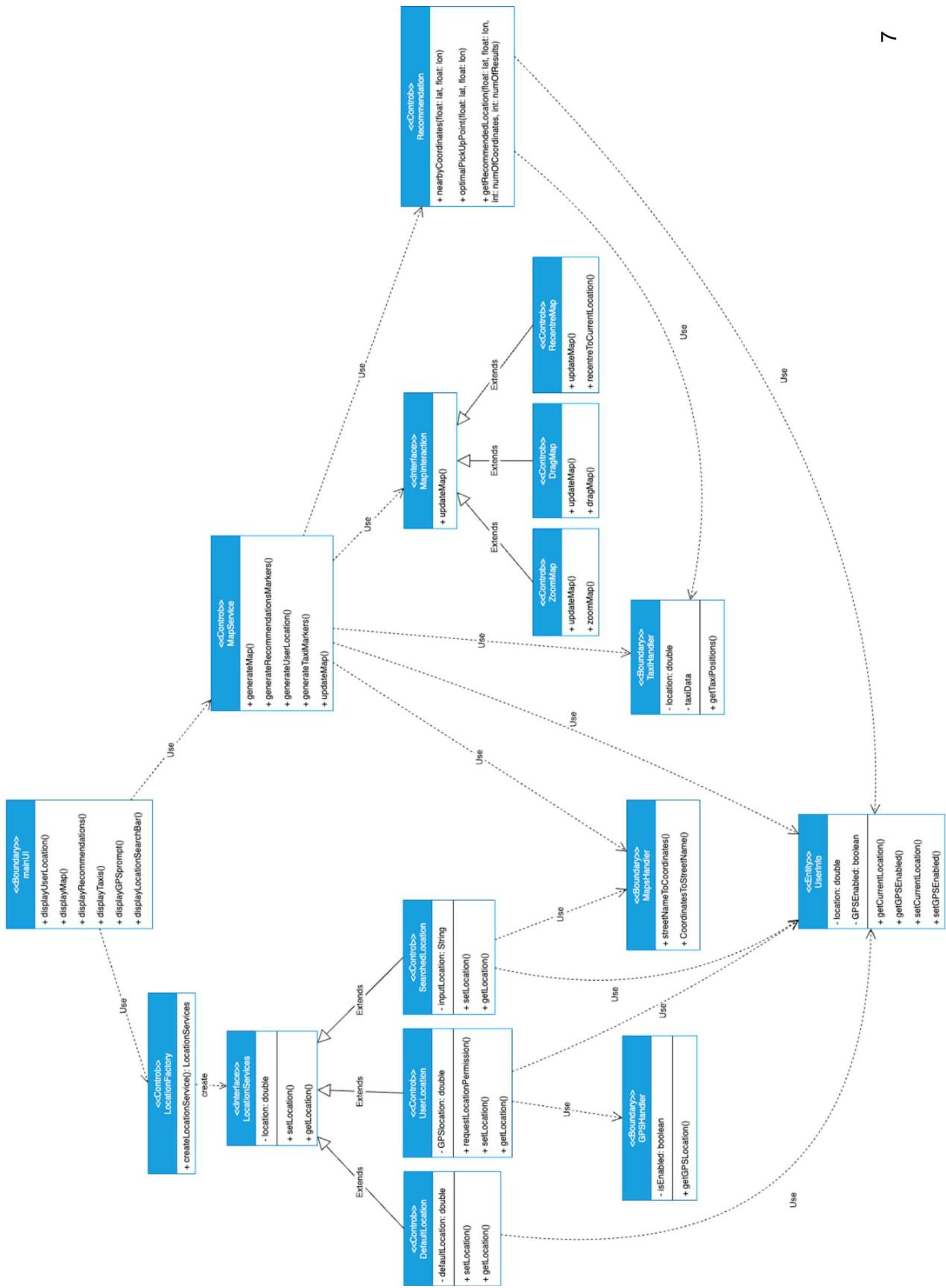
2.1.1 Use Case Diagram



2.2 Product Functions

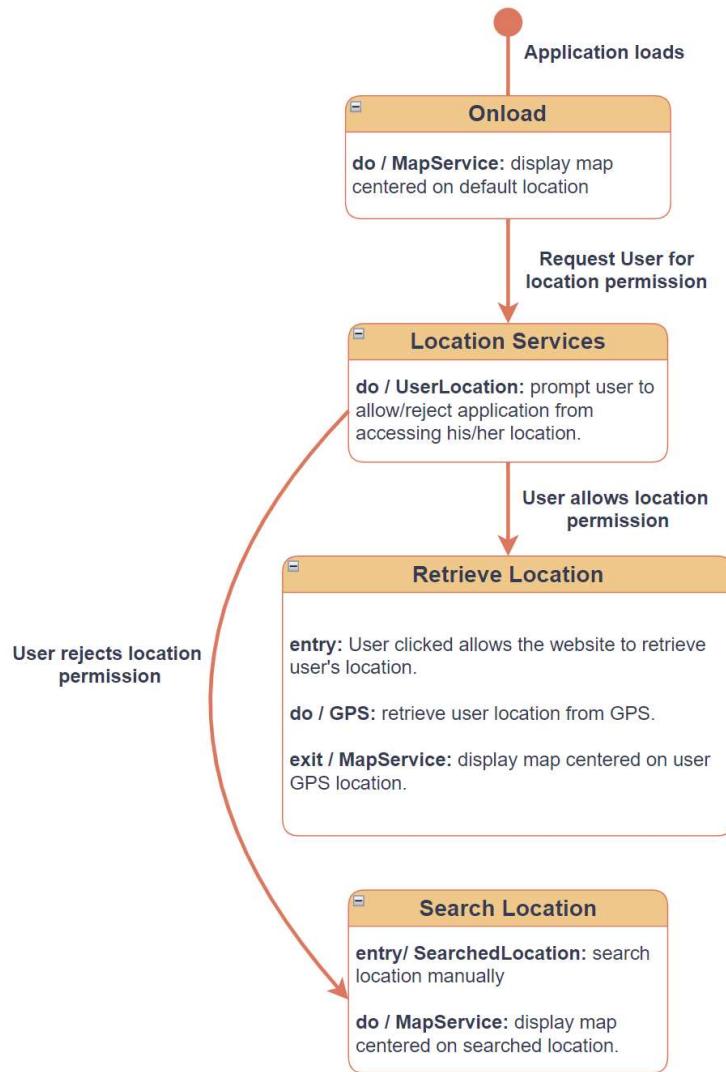
(Refer to the next page for class diagram)

2.2.1 Class Diagram

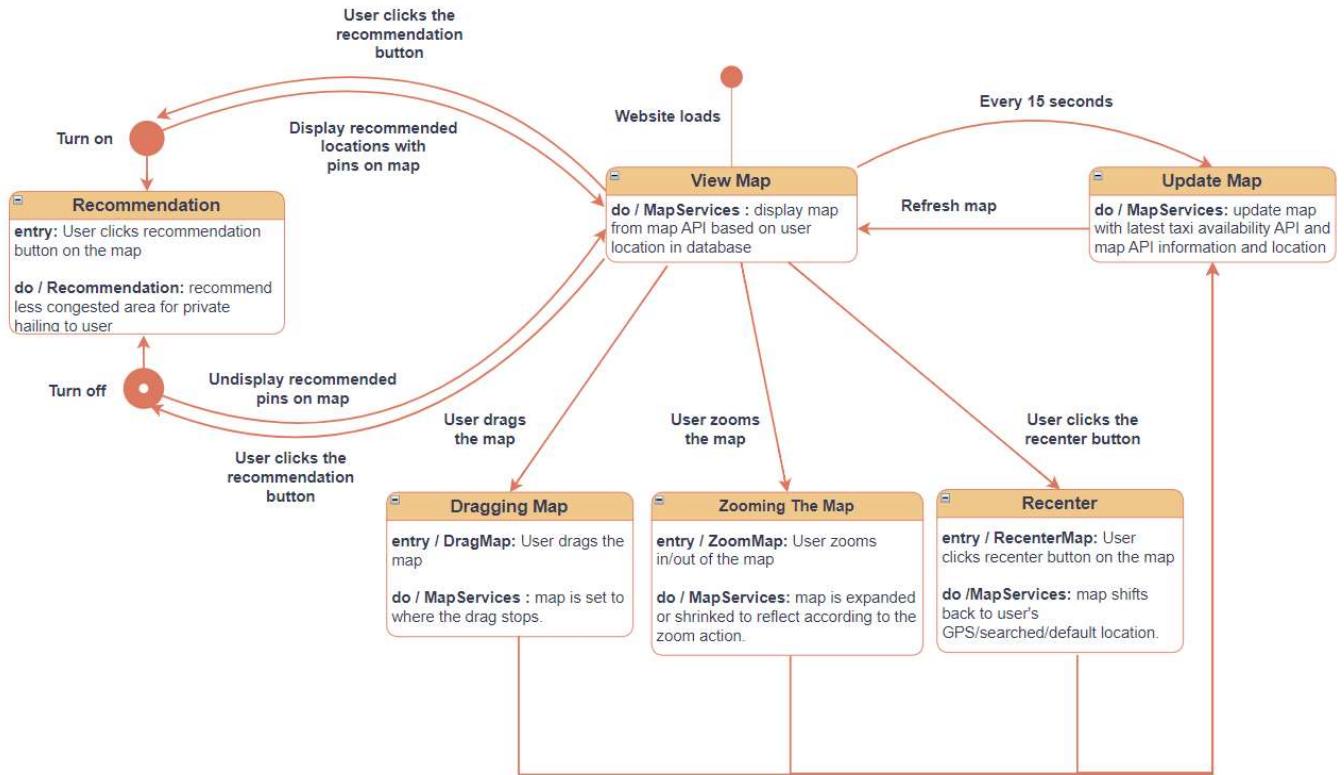


2.2.2 Dialog Map

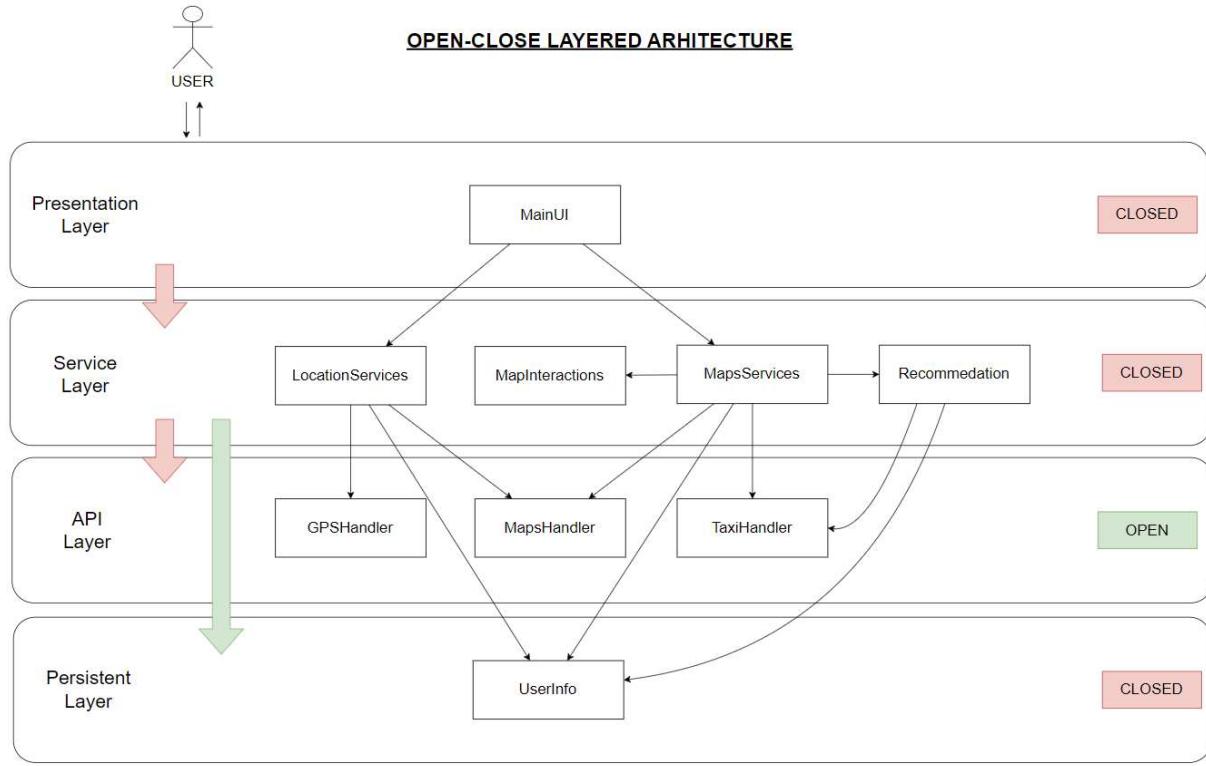
DIALOG MAP FOR SETTING LOCATION



DIALOG MAP FOR VIEWING MAP



2.2.3 System Architecture



2.3 User Classes and Characteristics

The majority of TaxiNow users are expected to be either checking where the taxis are located at or wanting to get a recommended location for hailing a cab. Users will use this application as they experience high demand pricing on private hire applications but do not know where to go to get a cheaper ride. This often results in users waiting endlessly or trying their luck by walking aimlessly to somewhere further in hope of reduction in high demand pricing. Otherwise, users could use the application to look up on the taxi guides and hotlines to make bookings.

2.4 Operating Environment

TaxiNow operates on both IOS (IOS 15 and above) and Android (OS 7 and above) devices. It is recommended that the device has location services functionality.

2.5 Design and Implementation Constraints

The software should be developed using proper code naming conventions in Javascript using React Native. The app should be able to run on both Android and IOS. The backend is created using Go programming language and it should be run on a server. The APIs utilised are:

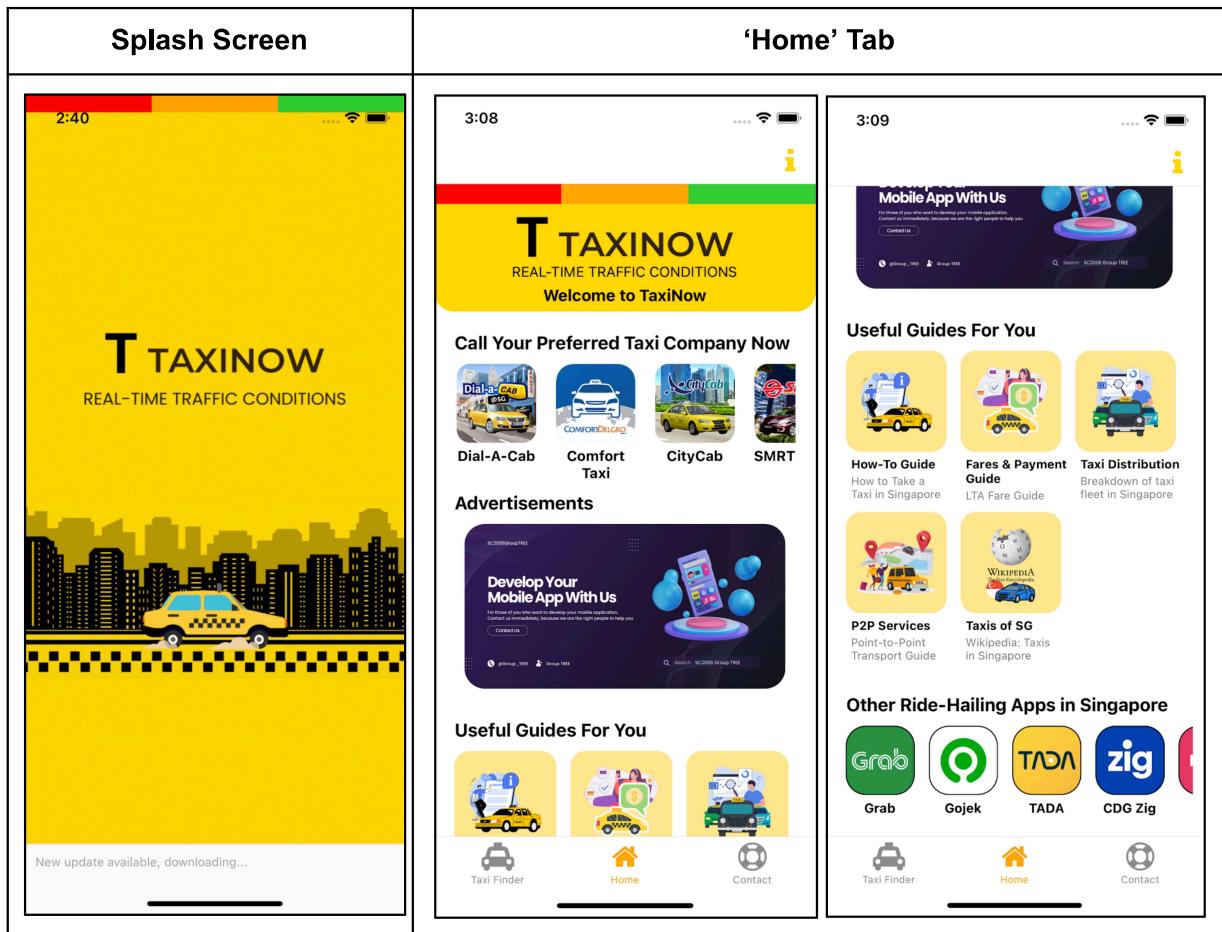
- Google API
 - Geolocation
 - Places
 - Routes
 - Maps
- Data Gov API
 - Taxi availability

2.6 Assumptions and Dependencies

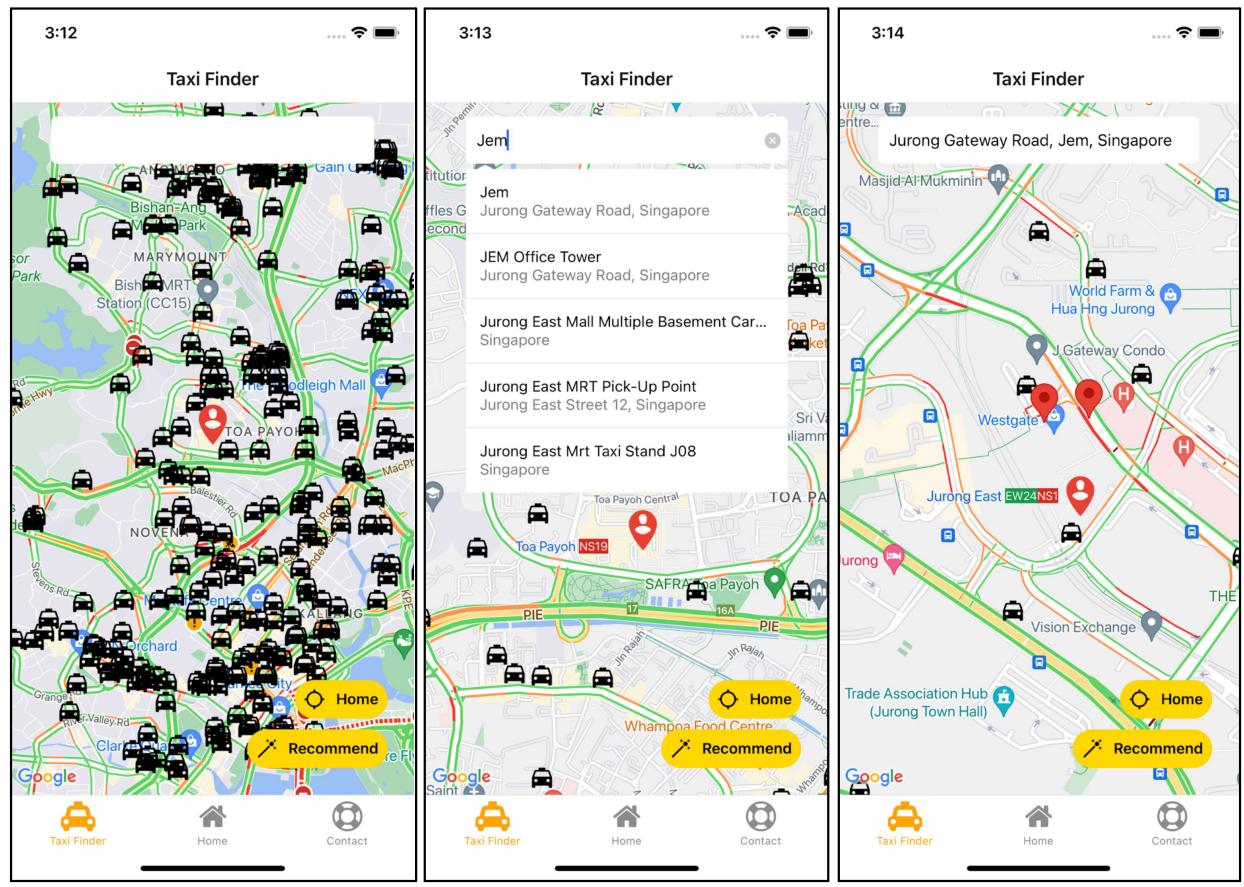
TaxiNow depends on third-party APIs which include Google Maps API and Data Gov API. The application may not work as intended if the application fails to connect to the API. It is assumed that the user has an internet connection when using the application.

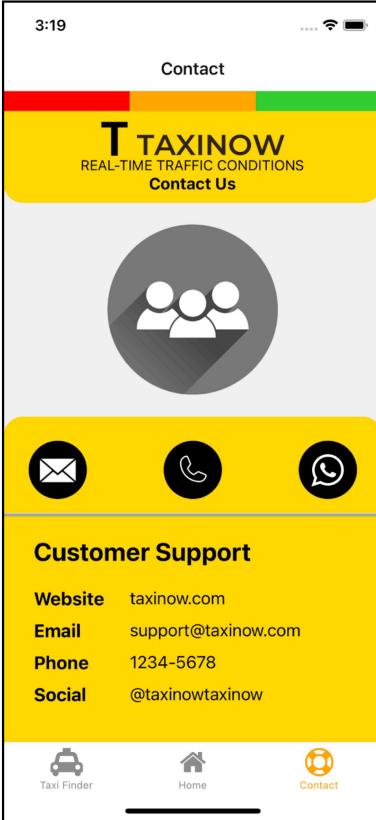
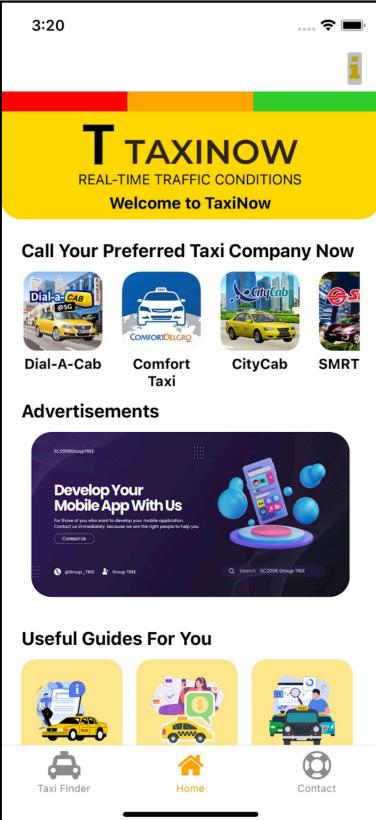
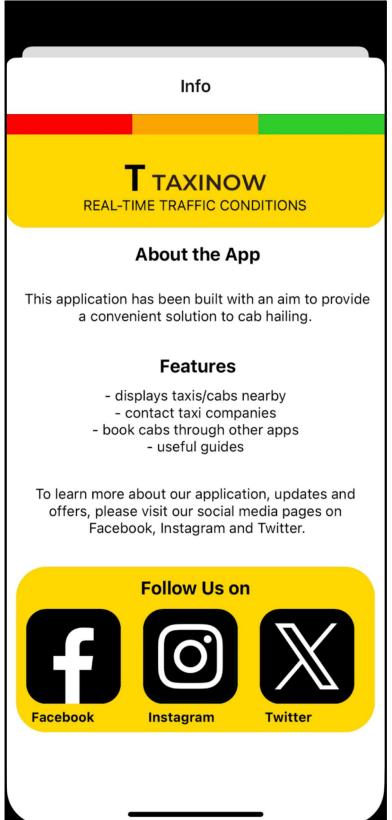
3. External Interface Requirements

3.1 User Interfaces



'Taxi Finder' Tab



'Contact' Tab	Clicking 'i' icon in 'Home' Tab → 'Info' Page	
 <p>Contact</p> <p>TAXINOW REAL-TIME TRAFFIC CONDITIONS Contact Us</p> <p>Customer Support</p> <p>Website taxinow.com Email support@taxinow.com Phone 1234-5678 Social @taxinowtaxinow</p> <p>Taxi Finder Home Contact</p>	 <p>TAXINOW REAL-TIME TRAFFIC CONDITIONS Welcome to TaxiNow</p> <p>Call Your Preferred Taxi Company Now</p> <p>Dial-A-Cab Comfort Taxi CityCab SMRT</p> <p>Advertisements</p> <p>Useful Guides For You</p> <p>Taxi Finder Home Contact</p>	 <p>Info</p> <p>TAXINOW REAL-TIME TRAFFIC CONDITIONS</p> <p>About the App</p> <p>This application has been built with an aim to provide a convenient solution to cab hailing.</p> <p>Features</p> <ul style="list-style-type: none"> - displays taxis/cabs nearby - contact taxi companies - book cabs through other apps - useful guides <p>To learn more about our application, updates and offers, please visit our social media pages on Facebook, Instagram and Twitter.</p> <p>Follow Us on</p> <p>Facebook Instagram Twitter</p> <p>Taxi Finder Home Contact</p>

4. System Features

4.1 Functional Requirements

Functional Requirements:

1. The user must be able to see a map on the mobile application on the Taxi Finder page.
 - 1.1. The user should be able to see all roads that can be seen on Google Maps within Singapore.
 - 1.1.1. The application must query the Google Maps API to retrieve the road data within Singapore.
 - 1.2. The user should be able to see the traffic conditions of all roads within a 5km radius from his position.
 - 1.2.1. The traffic conditions displayed must be the same as shown on the traffic condition API response.
 - 1.2.1.1. The mobile application must query the time traffic condition API to retrieve the traffic conditions.
 - 1.2.1.2. The traffic conditions must be updated to match the traffic condition API every 15 seconds.
 - 1.2.2. The traffic condition of all roads visible should be indicated with an appropriate colour.
 - 1.2.2.1. The colour should be green if there is no congestion.
 - 1.2.2.2. The colour should be orange if there is slight congestion.
 - 1.2.2.3. The colour should be red if there is heavy congestion.
 2. The user must be able to indicate his current location.
 - 2.1. The user's current location should be represented as a red pin on the map.
 - 2.1.1. The user's current location should be in the centre of the displayed map when the user loads into the application in the search bar.
 - 2.1.2. The map must show traffic conditions as indicated by 1.2.2. within 5km of the user's current location.
 - 2.2. Upon loading into the mobile application, the application should request a user's permission to retrieve their current location.
 - 2.2.1. If successful, the user's current location must be displayed correctly on the map.
 - 2.2.2. If unsuccessful, the user's current location must be defaulted to NTU North Spine bus stop.
 - 2.3. The user should be able to manually select a different location on the map.
 - 2.3.1. The user should be able to drag around the map to different areas in Singapore.
 - 2.3.2. The user should be able to return to automatic location detection mode by pressing a button that is labelled as "Home".
 3. The user must be able to see all available taxis within a 5km radius on the map from his position.
 - 3.1. The web application must query the live taxi API to retrieve live taxi position information.
 - 3.2. Each taxi should be represented by a black car symbol on the map.

- 3.3. The position of all the taxis that are within a 5km radius of the user should be updated every 10 seconds.
 - 3.4. Each taxi position should match the coordinates provided by the live taxi API response.
4. The mobile application should provide recommendations to the user on where to find a ride.
- 4.1. There should be a recommendation button “Recommend” on the bottom right of the application on the Taxi Finder page.
 - 4.1.1. The mobile application should respond with an appropriate suggestion within 3 seconds of the user pressing the button.
 - 4.1.1.1. If the current location is the most optimal, a red pop-up should appear, informing the user that the current location is the most optimal.
 - 4.1.1.2. If not, the mobile application should recommend areas indicated by a red pin near wherever the traffic congestion colours are indicated as the lowest.
 - 4.1.1.2.1. These locations should be indicated with pins on the map of the mobile application.
5. Users must be able to book cabs on the Home Page.
- 5.1 The user must be able to click on each taxi company icon to call cab companies.
 - 5.1.1 When the User clicks on the cab company's icon, the company's number must be displayed along with an option to dial that number. - 5.2 The user must be able to visit other cab applications.
 - 5.2.1 When the user clicks on the cab application's icon, it should direct them to the chosen cab company's app or website.
 - 5.2.1.1 If the user has not downloaded that Application yet, the URL should direct them to the App Store on iPhone or Play Store on Android.
6. Users must be able to view advertisements on the Home Page.
7. Users are offered Customer Support in the Contact Page.
- 7.1 When the user clicks on the Email icon, the app should direct them to sending an email to support@taxinow.com via their mail app.
 - 7.2 When the user clicks on the Phone icon, the app should direct them to the number 12345678.
 - 7.3 When the user clicks on the Whatsapp icon, the app should direct them to start a Whatsapp chat with 12345678.
8. The user must be able to view information upon clicking on the ‘i’ icon on the top right corner of the Home Page.
- 8.1 The user must be able to view information about the application.
 - 8.2 The user must be able to visit the Social Media pages of the application.
 - 8.2.1 When the user clicks on the Facebook icon, the app directs them to Facebook.
 - 8.2.2 When the user clicks on the Instagram icon, the app directs them to Instagram.
 - 8.2.3 When the user clicks on the Twitter icon, the app directs them to Twitter (X).

4.2 Use Case Description

Use Case ID:	001		
Use Case Name:	Set default location		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	11/2/2024	Date Last Updated:	26/3/2024

Actor:	User
Description:	Sets the user's location to be at a default location and reflects it on the map.
Preconditions:	1. The user must have just loaded into the application or never allows location services.
Postconditions:	1. The application stores the user's location to be at a default location in the system. 2. The application shows the user's location to be at the default location.
Priority:	-
Frequency of Use:	1 - 3 Times per lifetime of the app being run.
Flow of Events:	1. The application sets and updates the location of the user in the system to be in NTU North Spine. 2. The application shows the user's updated location on the map.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	002		
Use Case Name:	Set Manual Location		
Created By:	Tan Jing Jie	Last Updated By:	Soo Qi Yang
Date Created:	6/2/2024	Date Last Updated:	11/2/2024

Actor:	User
Description:	Sets the user's location to be at a user-defined location and reflects it on the map. Taxi information is updated on the map.
Preconditions:	1. This use case extends the SetDefaultLocation use case. It is initiated by the system if the user desires to set a manual location.
Postconditions:	1. The application stores the user's location to be at his selected location in the system. 2. The application shows the user's location to be at his selected location.
Priority:	-
Frequency of Use:	1-5 Times per lifetime of the web app being run.
Flow of Events:	<ol style="list-style-type: none"> 1. User keys in a specific location that he/she wants to manually set his/her current location to, into a search bar. 2. Application will send the name of the user specified location to the map API. 3. The application will request for the coordinates of the location from the map API. 4. The user presses the return key on the search bar. 5. The application stores and fixes the user's current location to be at the specified coordinates in the system. 6. The application shows the user's updated location on the map. 7. The application displays the location of available taxis that it had retrieved and stored from the UpdateAvailability use case, using the included "See available taxis 5km near the set location use" case.
Alternative Flows:	-
Exceptions:	<p>EX1: The user tries to set his manual location as a place that is not in Singapore.</p> <ol style="list-style-type: none"> 1. The application displays an error message "Error. Cannot set location outside of Singapore". <p>EX2: The map API does not respond</p> <ol style="list-style-type: none"> 1. The application shows a map API connection failure message that informs the user to try again by refreshing the page.
Includes:	- See available taxis 5km near the set location
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	003		
Use Case Name:	Update user location with GPS		
Created By:	Tan Jing Jie	Last Updated By:	Soo Qi Yang
Date Created:	6/2/2024	Date Last Updated:	11/2/2024

Actor:	User, GPS
Description:	Detects the user's real time current location and reflects it on the map. Taxi information is updated on the map.
Preconditions:	<ol style="list-style-type: none"> This use case extends the SetDefaultLocation use case. It is initiated by the system whenever the user has enabled location services and does not have a manual location set.
Postconditions:	<ol style="list-style-type: none"> The application stores the user's location to be at his current location in the system. The application shows the user's location to be at his current location.
Priority:	-
Frequency of Use:	Once every 15 seconds.
Flow of Events:	<ol style="list-style-type: none"> The user has been on the application for 15 seconds continuously. The application obtains the user's current location from the GPS. The application will store the user's current location to be at the specified coordinates in the system. The application will show the user's updated current location on the map. The application displays the location of available taxis that it had retrieved and stored from the UpdateAvailability use case using the included "See available taxis 5km near the set location use" case.
Alternative Flows:	-
Exceptions:	<p>EX1: The GPS does not respond</p> <ol style="list-style-type: none"> The application shows a GPS failure message that informs the user to check their permissions and try again by refreshing the page, or to set their location manually.
Includes:	- See available taxis 5km near the set location
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	004		
Use Case Name:	View the map		
Created By:	Tan Jing Jie	Last Updated By:	Soo Qi Yang
Date Created:	6/2/2024	Date Last Updated:	11/2/2024

Actor:	User
Description:	Map, traffic data retrieved from the maps API and traffic availability API is displayed on the website. The map dynamically updates every 5 seconds to show any updated information that the application may have received from the APIs.
Preconditions:	1. The application has been loaded
Postconditions:	1. Users can see the map, and if within his position range, the corresponding latest traffic conditions and taxis.
Priority:	-
Frequency of Use:	Approximately 12 times a minute.
Flow of Events:	<ol style="list-style-type: none"> 1. The user loads the website, or has been on the website for at least 5 seconds continuously. 2. The application displays the corresponding traffic conditions that it had retrieved and stored from the UpdateMapWithTrafficInformation use case.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	005		
Use Case Name:	Zooming the map		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	11/2/2024	Date Last Updated:	26/3/2024

Actor:	User
Description:	The user can magnify and minimise the size of the map according to their personal preference.
Preconditions:	<ol style="list-style-type: none"> This use case extends the ViewTheMap use case. It is initiated by the system whenever the user issues a zoom command on the application.
Postconditions:	<ol style="list-style-type: none"> The user is able to view the map at his preferred zoom size.
Priority:	-
Frequency of Use:	Approximately 7 times a minute.
Flow of Events:	<ol style="list-style-type: none"> The user zooms in or out of the map by pinching in or out of the map.. The application updates its zoom percentage internally. The application displays the map data at the corresponding zoom percentage to the user. Sends an alert to the application to notify the application that the map has been zoomed, so it can seek to update the traffic information within 5 seconds.
Alternative Flows:	-
Exceptions:	<p>EX1: The user attempts to zoom in beyond 500%.</p> <ol style="list-style-type: none"> The application's zoom percentage remains capped at 500% and does not increase further.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	006		
Use Case Name:	Drag the map		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	11/2/2024	Date Last Updated:	11/2/2024

Actor:	User
Description:	The user can drag the map to view places other than his current set location.
Preconditions:	<ul style="list-style-type: none"> 1. This use case extends the ViewTheMap use case. It is initiated by the system whenever the user clicks, holds and attempts to drag the map on the application.
Postconditions:	<ul style="list-style-type: none"> 1. The user is able to view the desired location of the map.
Priority:	-
Frequency of Use:	Approximately 15 times a minute.
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks, holds and drags the map 2. The application identifies the distance moved by the mouse, then according to the new centre location and zoom level, it requests the relevant map information from the Maps API. 3. The application displays the map data obtained from the Maps API. 4. Sends an alert to the application to notify the application that the map has been dragged, so it can seek to update the traffic information within 5 seconds.
Alternative Flows:	-
Exceptions:	<p>EX1: The map API does not respond</p> <ol style="list-style-type: none"> 1. The application shows a map API connection failure message that informs the user to try again by refreshing the page.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	007		
Use Case Name:	Re-centre to current location		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	11/2/2024	Date Last Updated:	11/2/2024

Actor:	User
Description:	The user can move the map back to be centred on his current location.
Preconditions:	<ul style="list-style-type: none"> 1. This use case extends the ViewTheMap use case. It is initiated by the system whenever the user clicks the re-centre button.
Postconditions:	<ul style="list-style-type: none"> 1. The user is able to view the map with his current set location centralised.
Priority:	-
Frequency of Use:	Approximately 3 times a minute.
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks on the re-centre button. 2. The user's current location is obtained from the system. 3. The application displays the appropriate map information and location, centred around the user's current location according to the zoom size stored.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	008		
Use Case Name:	See available taxis 5km near the set location		
Created By:	Tan Jing Jie	Last Updated By:	Soo Qi Yang
Date Created:	6/2/2024	Date Last Updated:	11/2/2024

Actor:	User
Description:	Available taxis within 5km proximity of user location or specified location is shown.
Preconditions:	1. The user must have loaded the UI of the application.
Postconditions:	1. The map displays the location of all nearby available taxis on the map.
Priority:	-
Frequency of Use:	Once every 10 seconds
Flow of Events:	<ol style="list-style-type: none"> 1. The user changes his location. 2. The application receives a notification that will initiate the UpdateAvailability use case. 3. The user's current coordinates are obtained from the system. 4. From the application's storage of the taxi locations, which are updated by the UpdateAvailability use case, the application only displays those taxis that are within 5km radius of the user's current location.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	009		
Use Case Name:	View recommendations of optimal location to find a ride		
Created By:	Tan Jing Jie	Last Updated By:	Soo Qi Yang
Date Created:	6/2/2024	Date Last Updated:	11/2/2024

Actor:	User
Description:	Recommend locations indicated with pins on the map for user to hail a ride within 0.2km radius around user.
Preconditions:	1. User must have his location set already.
Postconditions:	2. The application displays the recommended optimal locations to find a ride.
Priority:	-
Frequency of Use:	Once a minute.
Flow of Events:	<ol style="list-style-type: none"> 1. The user presses the “Get recommendation” button. 2. The application obtain user’s location from the system. 3. The application calculates the optimal location coordinates using the information about the nearby taxis and traffic conditions that would be continuously updated by “Update map with traffic information” and “Update Availability” use cases. 4. The application displays the optimal location to the user as pins on the map.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	010		
Use Case Name:	Update map with map and traffic information		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	11/2/2024	Date Last Updated:	11/2/2024

Actor:	Maps API
Description:	Updates the application with map and traffic information
Preconditions:	-
Postconditions:	<ul style="list-style-type: none"> 1. The application has the latest information on the map and the traffic conditions.
Priority:	-
Frequency of Use:	Approximately 6 times a minute.
Flow of Events:	<ol style="list-style-type: none"> 1. The application has run for 15 seconds continuously, or the system receives a notification to update the traffic information. (Possibly from Drag the Map or Zooming the map use cases) 2. The application requests information on traffic congestion from the Maps API. 3. The application requests map information from the Maps API. 4. The application stores this information that can be used by other use cases, such as View the map.
Alternative Flows:	-
Exceptions:	<p>EX1: The application has not been able to obtain and update its traffic congestion information or Map information from the Maps API.</p> <ol style="list-style-type: none"> 1. The application shows a Maps API connection failure message that informs the user to try again by refreshing the page.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	011		
Use Case Name:	Update availability		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	11/2/2024	Date Last Updated:	11/2/2024

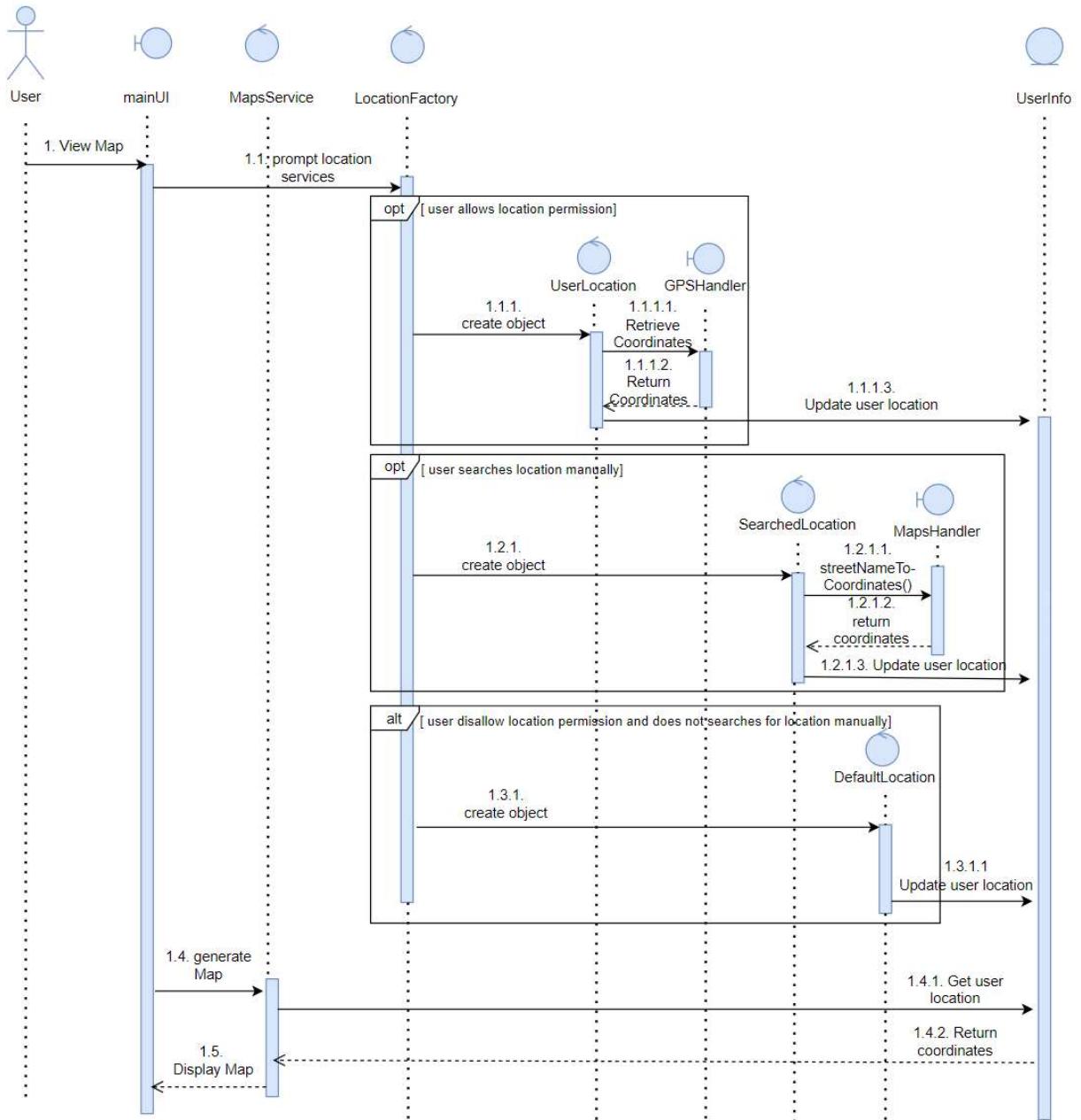
Actor:	Taxi API
Description:	Updates the application with taxi availability information
Preconditions:	-
Postconditions:	<ul style="list-style-type: none"> 1. The application has the latest information on all the available taxis in Singapore.
Priority:	-
Frequency of Use:	Approximately 6 times a minute.
Flow of Events:	<ul style="list-style-type: none"> 1. The application has run for 15 seconds continuously. 2. The application requests information on live taxi coordinates from the Taxi Availability API. 3. The application stores this information that can be used by other use cases, such as View the map.
Alternative Flows:	-
Exceptions:	<p>EX1: The application has not been able to obtain and update its Taxi coordinates information from the Taxi Availability API.</p> <ul style="list-style-type: none"> 1. The application shows a Taxi Availability API connection failure message that informs the user to try again by refreshing the page.
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	012		
Use Case Name:	Call Taxi Hotline		
Created By:	Soo Qi Yang	Last Updated By:	Soo Qi Yang
Date Created:	4/4/2024	Date Last Updated:	4/4/2024

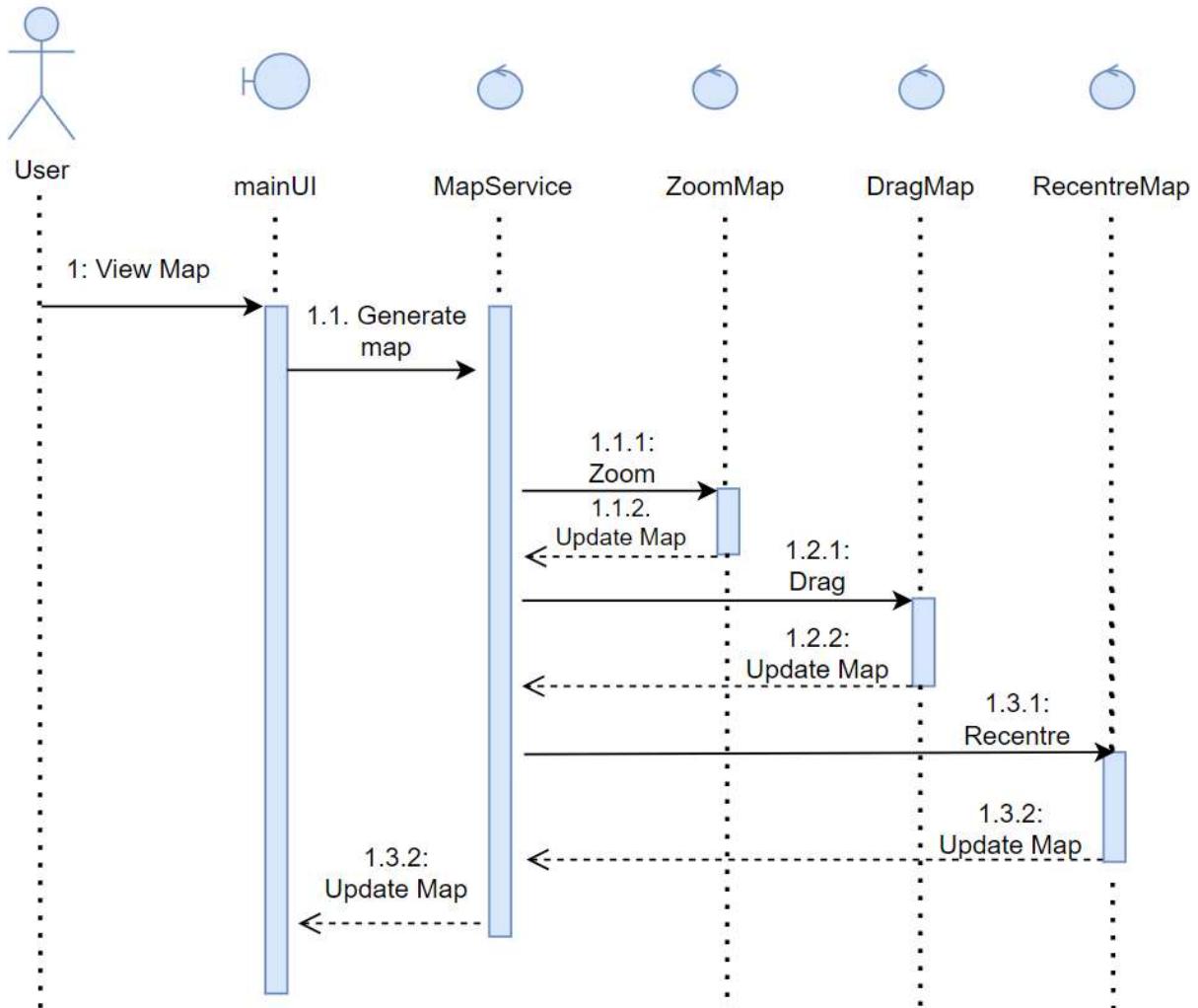
Actor:	User
Description:	Allows the user to obtain the hotline number of the relevant taxi brand they want to call easily.
Preconditions:	-
Postconditions:	The user will be on a screen, that is able to call the taxi hotline with a single click of the phone button.
Priority:	-
Frequency of Use:	Approximately 1 times every 5 minutes.
Flow of Events:	<ol style="list-style-type: none"> 1. The user scrolls on the app to find the brand of taxi they want to call 2. The user presses the button that has the name of the taxi brand they wish to call 3. The user is brought to a system phone or calling application of choice, with the appropriate hotline number already keyed in.
Alternative Flows:	-
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

4.3 Sequence Diagrams

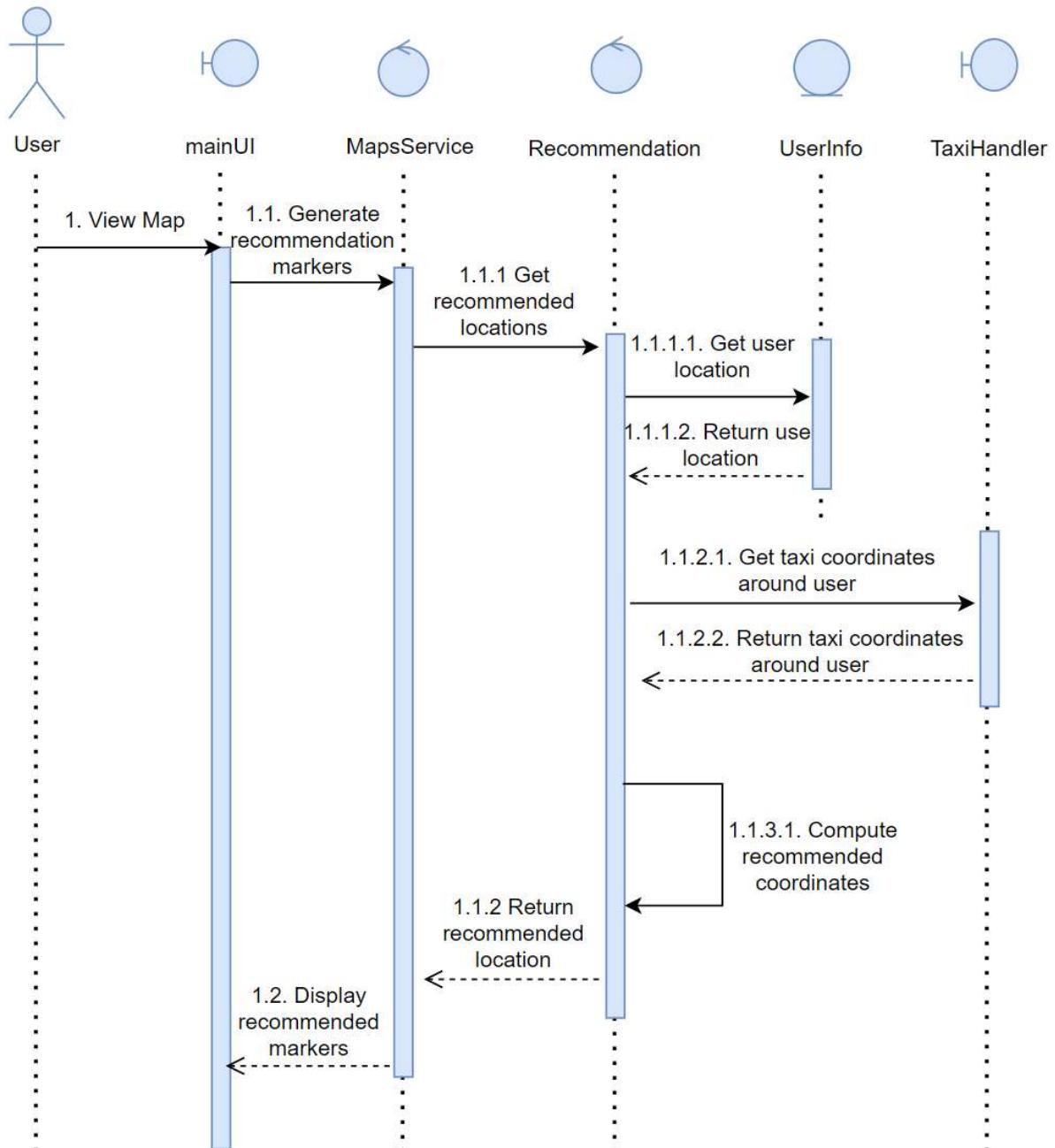
4.3.1 Setting User Location On Map



4.3.2 Map Interactions



4.3.3 Generating Recommendation



5. Other Non-Functional Requirement

5.1 Performance Requirements

5.1.1 Maintainability and Platform

The system must run on the IOS (IOS 15 and above) and Android (OS 7 and above) mobile operating system.

5.1.2 Performance

- The system should ask for the user's location services permission within 3 seconds of starting the app.
- The system should load the map within 4 seconds after the app is loaded.
- The system should load the road conditions on the map within 3 seconds after the map is loaded.
- The system should load the nearby taxis for the user to see within 2 seconds after the map is loaded.
- The system should calculate and display the recommendation within 6 seconds of the user pressing the recommendation button.
- The system should re-centre the map back to the user's current location within 3 seconds of pressing the Home button.
- If the user has enabled live location services, the interval between updates of the user's live position (displayed by the user's position pin on the map) should not be more than 20 seconds.
- After a location is searched and the return key is pressed on the manual location search bar, the system should not take more than 3 seconds to move the user's location pin to the new location and display the map centralised on it.
- After the user's pin has been moved to a new location (either by search functionality or location services), the system should display all available taxis in a 5km radius, within 2 seconds.
- Upon dragging the map to a new place, the new portion of the map should be rendered on the screen within 2 seconds.
- The system should load and display the appropriate tab within 2 seconds of clicking onto it.
- On the IOS mobile app, upon pressing a taxi company icon in the "Home" tab, the system should display the option to call the taxi company within 2 seconds.
- On the Android mobile app, upon pressing a taxi company icon in the "Home" tab, the system should display the option for the user to select the app that he wants to use to call the taxi company, within 2 seconds.

5.2 Security Requirements

5.2.1 User Permissions Security

- The system must never track the user's location unless the user gives it permission to.

5.2.2 Data Minimisation

- The system should not collect any unnecessary information from the user, other than the ones that are relevant, in this case, only their current location (if permitted).
- The system must not store or collect data of the user's location history. Such information should only be kept as long as needed and discarded afterwards.

5.3 Software Quality Attributes

5.3.1 Reliability

- The system must be able to show the taxi information at least 95% of the time.
- The system must be able to provide a recommendation at least 95% of the time.

5.3.2 Usability

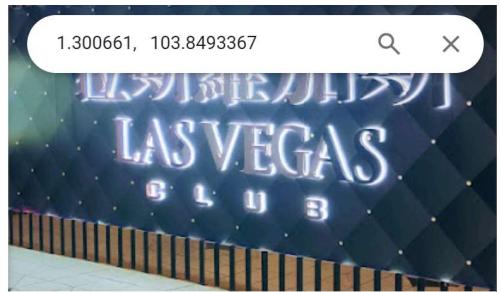
- 9 out of 10 new users should be able to get the system to calculate and display a recommended spot to hire a taxi within 1 minute of loading the app.
- All interactive buttons should have accompanying tooltips or text to briefly explain its functionality.

6. Testing

6.1 Blackbox Testing

6.1.1 Converting Street Name to Coordinates

Black Box testing is done on our streetNameToCoordinates function that converts street names to coordinates. For valid equivalence class, we identified it to be any address and/or postal code in Singapore. For invalid equivalence class, we identified it to be any location outside of Singapore.

Street Name	Expected Output	Actual Output				
JEM (valid equivalence class)	1.332647, 103.7433111	<p>Request URL <code>http://34.143.206.119/maps/sntocoords?streetName=JEM</code></p> <p>Server response</p> <table><tr><td>Code</td><td>Details</td></tr><tr><td>200</td><td>Response body</td></tr></table> <pre>[1.332647, 103.7433111]</pre>	Code	Details	200	Response body
Code	Details					
200	Response body					
Las Vegas (invalid equivalence class)	It should not return Las Vegas in America but rather a location in Singapore that is related to the name “Las Vegas”	<p>Request URL <code>http://34.143.206.119/maps/sntocoords?streetName=Las%20Vegas</code></p> <p>Server response</p> <table><tr><td>Code</td><td>Details</td></tr><tr><td>200</td><td>Response body</td></tr></table> <pre>[1.300661, 103.8493367]</pre>  <p>Club Duke 名爵 4.1 ★★★★☆ (20) 卡拉OK厅</p>	Code	Details	200	Response body
Code	Details					
200	Response body					

6.1.2 Get Taxi Coordinates Around User

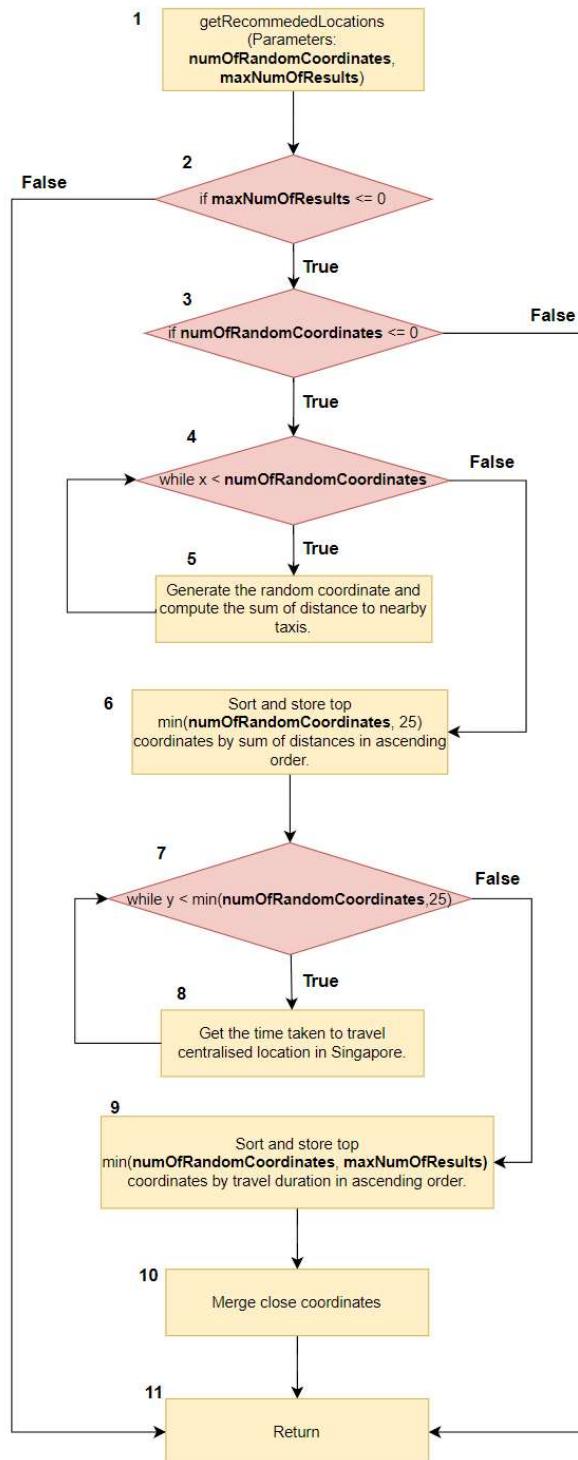
Black Box testing is done on our getNearbyTaxiLocations function that retrieves all the coordinates of taxis within 5km proximity of the user. For a valid equivalence class, we identified it to be any coordinates within Singapore. For invalid equivalence class, we identified it to be any coordinates outside of Singapore.

Latitude	Longitude	Expected Output	Actual Output		
1.34705327277948 97 (valid input)	103.680810966265 13 (valid input)	A list with at least 1 taxi coordinates.	<p>Request URL <code>http://34.143.206.119/taxi?latitude=1.34705327277948</code></p> <p>Server response</p> <table> <tr> <td>Code</td> <td>Details</td> </tr> </table> <p>200</p> <p>Response body</p> <pre>[{ "latitude": 1.33371, "longitude": 103.64354 }, { "latitude": 1.31216, "longitude": 103.6623 }, { "latitude": 1.32573, "longitude": 103.67934 }, { "latitude": 1.3407992, "longitude": 103.68582863 }, { "latitude": 1.34858215, "longitude": 103.68666165 }, { "latitude": 1.34, "longitude": 103.68848 }, { "latitude": 1.3409945, "longitude": 103.6888915 }]</pre>	Code	Details
Code	Details				
36.1822086964353 9 (invalid input - outside of Singapore)	103.680810966265 13 (valid input)	null	<p>Request URL <code>http://34.143.206.119/taxi?latitude=36.1822086964353</code></p> <p>Server response</p> <table> <tr> <td>Code</td> <td>Details</td> </tr> </table> <p>200</p> <p>Response body</p> <pre>null</pre>	Code	Details
Code	Details				

1.34705327277948	-115.165432103308	null	Request URL <code>http://34.143.206.119/taxi?lat=1.34705327277948&lon=-115.165432103308</code>				
97 (valid input)	42 (invalid input - outside of Singapore)		Server response <table><thead><tr><th>Code</th><th>Details</th></tr></thead><tbody><tr><td>200</td><td>Response body <code>null</code></td></tr></tbody></table>	Code	Details	200	Response body <code>null</code>
Code	Details						
200	Response body <code>null</code>						

6.2 Whitebox Testing

6.2.1 Getting Recommended Locations



Basis Paths:

- I. 1, 2 ,3, 4, 5, 4, 6 ,7, 8, 7, 9, 10, 11 (Baseline path - “typical path”)
- II. 1, 2, 3, 4, 5, 4, 6, 7, 9, 10, 11
- III. 1, 2, 3, 4, 6, 7, 8, 7, 9, 10, 11
- IV. 1, 2, 11
- V. 1, 2, 3, 11

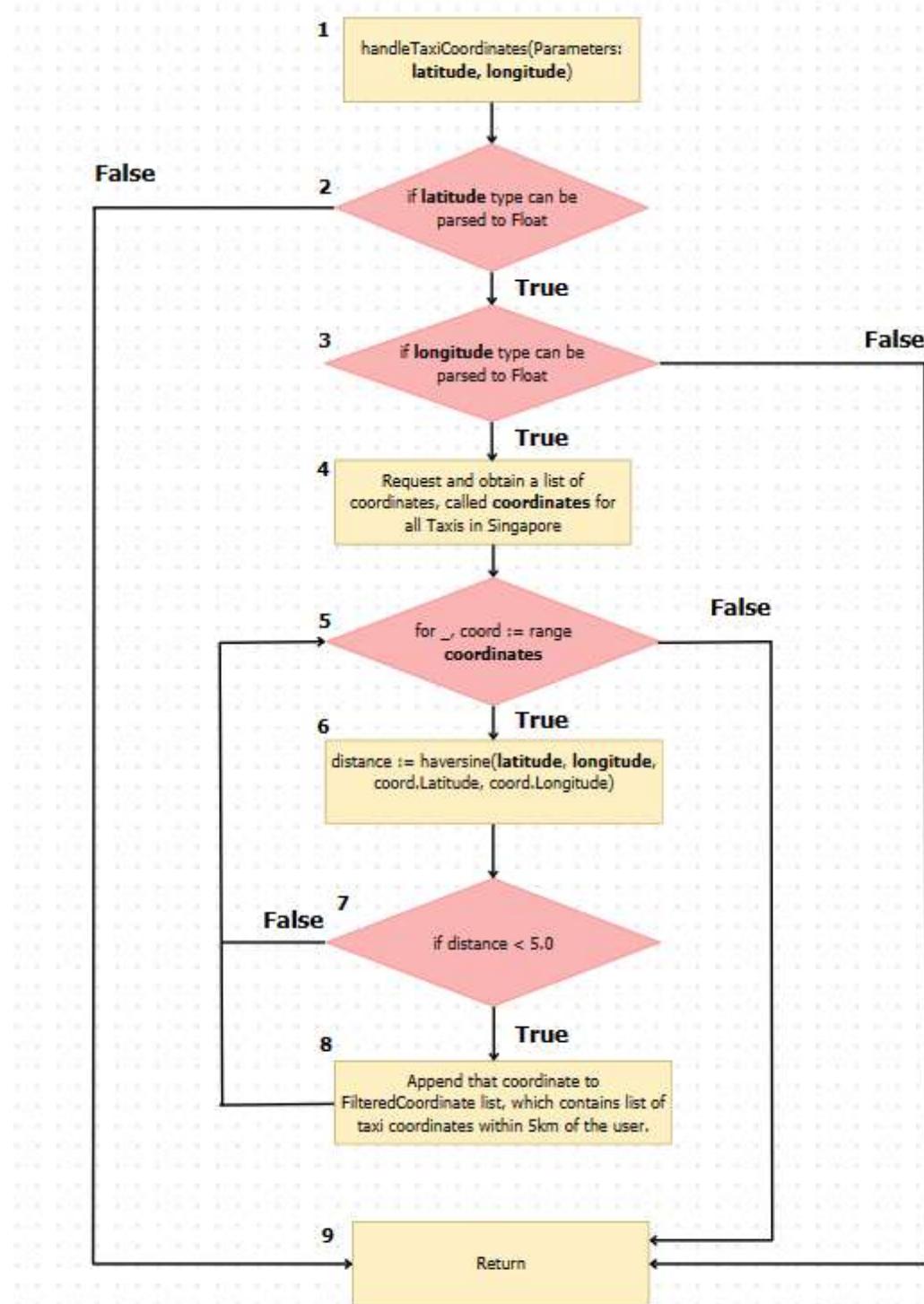
Test Cases:

- I. numOfRandomCoordinates = 1, maxNumOfResults = 1
- II. Infeasible basis path (the while loop is guaranteed to execute if node 2 & 3 is true)
- III. Infeasible basis path (the while loop is guaranteed to execute if node 2 & 3 is true)
- IV. numOfRandomCoordinates = 1, maxNumOfResults = 0
- V. numOfRandomCoordinates = 0, maxNumOfResults = 1

Real Execution Path:

- I. 1, 2 ,3, 4, 5, 4, 6 ,7, 8, 7, 9, 10, 11
- II. -
- III. -
- IV. 1, 2, 11
- V. 1, 2, 3, 11

6.2.2 Getting Coordinates of Taxis within 5km radius of a given latitude and longitude



Basis Paths:

- I. 1, 2, 3, 4, 5, 6, 7, 5 6, 7, 8, 5, 9 (Baseline path - “typical path”)
- II. 1, 2, 3, 4, 5, 9
- III. 1, 2, 3, 4, 5, 6, 7, 8, 5, 9
- IV. 1, 2, 3, 4, 5, 6, 7, 5, 9
- V. 1, 2, 9
- VI. 1, 2, 3, 9

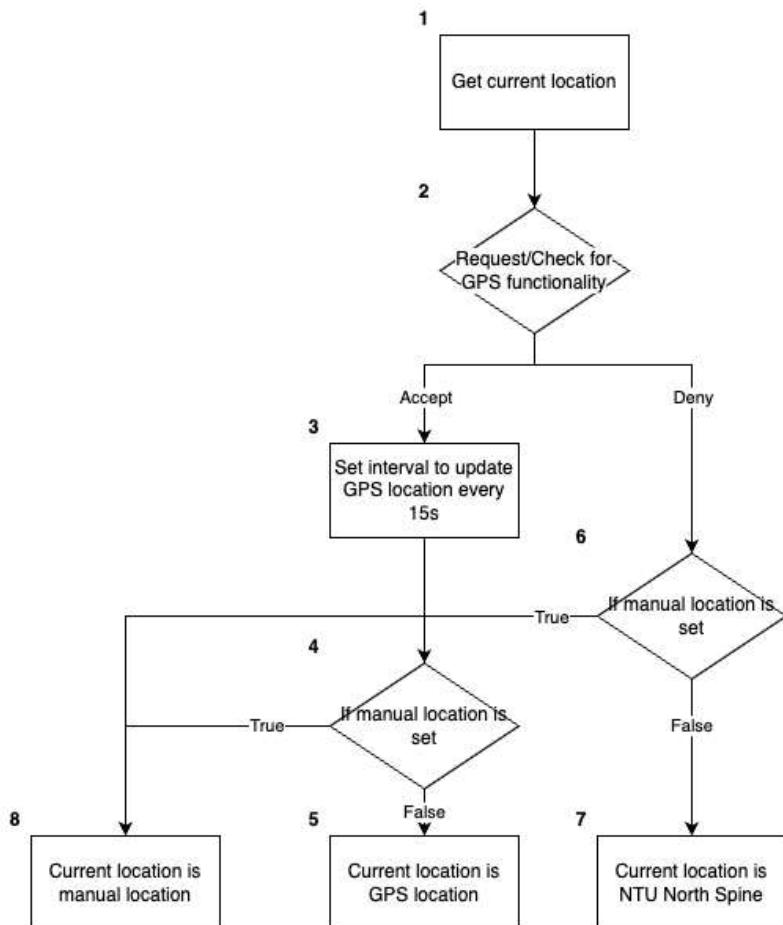
Test Cases:

- I. **latitude** = 1.3332, **longitude**= 103.74338 (Jurong East Mall in Singapore)
- II. **Infeasible** basis path (there are always taxis in singapore, hence the initial coordinates list will never be empty)
- III. **Infeasible** basis path (taxis are too widely spread over Singapore. There is no input latitude/longitude combination such that every taxi will be within 5km of it)
- IV. **latitude** = 34.052235, **longitude** = -118.243683 (Los Angeles, USA. Should not have any taxis in Singapore within 5km radius of it)
- V. **latitude** = “THISISASTRINGNOTAFLOAT” , **longitude** = 103.74338
- VI. **latitude** = 1.3332 , **longitude** = “THISISASTRINGNOTAFLOAT”

Real Execution Path:

- I. 1, 2, 3, 4, 5, 6, 7, 5 6, 7, 8, 5, 9
- II. -
- III. -
- IV. 1, 2, 3, 4, 5, 6, 7, 5, 9
- V. 1, 2, 9
- VI. 1, 2, 3, 9

6.2.3 State transition of Default, GPS and Manual Location



Basic Paths:

- I. 1, 2, 6, 7 (Baseline Path)
- II. 1, 2, 3, 4, 5
- III. 1, 2, 6, 8
- IV. 1, 2, 3, 4, 8

Test Case:

- I. Disabled location service (Location = NTU North Spine Coordinates)
- II. Enabled location service (Location = NTU SCSE Building)
- III. Disabled location service, typed JEM manually (Location = JEM Building)
- IV. Enabled location service, typed JEM manually (Location = JEM Building)

Real Execution Paths:

- I. No location service, no manual location, current location = NTU North Spine
- II. With location service, no manual location, current location = GPS Location
- III. No location service, with manual location, current location = manual location
- IV. With location service, with manual location, current location = manual location