
Software Requirements Specification

for

Farely

Version 1.0 approved

Prepared by

Chulpaibul Jiraporn
Leong Ko Rixie Tiffany
Nguyen Duy Khanh
Rachel Khan
Tran Anh Tai
Truong Quang Duc

Chill Team

18 / 03 / 2020

Table of Contents

1. Introduction	1
1.1. Purpose	1
1.2. Document Conventions	1
1.3. Intended Audience and Reading Suggestions	1
1.4. Product Scope	1
1.5. References	2
2. Overall Description	3
2.1. Product Perspective	3
2.2. Product Functions	3
2.3. User Classes and Characteristics	3
2.4. Operating Environment	3
2.5. Design and Implementation Constraints	3
2.6. User Documentation	4
2.7. Assumptions and Dependencies	4
3. External Interface Requirements	5
3.1. User Interfaces	5
3.2. Hardware Interfaces	5
3.3. Software Interfaces	5
3.4. Communications Interfaces	5
4. System Features	6
4.1. Accept Route Queries	6
4.2. Display Sorted Routes	7
4.3. Display Route Details	8
4.4. Display Air Quality and Weather Information	9
5. Other Nonfunctional Requirements	11
5.1. Performance Requirements	11
5.2. Safety Requirements	11
5.3. Security Requirements	11
5.4. Software Quality Attributes	11
5.5. Business Rules	11
Appendix A: Glossary	12
Appendix B: Analysis Models	14

Revision History

Name	Date	Reason For Changes	Version
Everyone	18/03/2020	Initialisation	1.0

1. Introduction

1.1 Purpose

The ZEA has launched Data-Driven Smart Nation Competition to elicit innovative applications that exploit the publicly available government data to change the ways we live, learn and work. As a participant of the competition, Team Chill has developed Farely. Farely is a **public transport Android application** that enables Singaporean commuters to find the best **routes** from a departure location to an arrival location, based on the **estimated price, estimated travel time or distance of the route**. This document will describe the complete features, functionalities and interface of our application.

1.2 Document Conventions

The format of the software requirement specification is simple. Bold font indicates topic title while the remainder of the document will be written using standard font Times New Roman, size 12. The document uses a multi-level numbered list to provide details for features and requirements.

1.3 Intended Audience and Reading Suggestions

The document's intended audience includes developers, testers and users.

- **Developers** who intend to use this document as a guideline for implementing the system are recommended to read the Overall Description, External Interfaces, System Features and Other Non Functional Requirements to understand the features and requirements of the system.
- **Testers** who wish to use the document as a guideline to formulate test cases are recommended to review System Features and Other Non Functional Requirements.
- **Users** who are interested in reading the software requirement specification document can review Overall Description and System Features to know more about the features the software provides.

1.4 Product Scope

The objectives of Farely are to find the most efficient route according to the user's preference and provide detailed description of the routes so as to minimize the cost of commuting in Singapore (fare price, distance or time).

1.5 References

Not applicable.

2. Overall Description

2.1 Product Perspective

Farely is a replacement for existing map apps. It is intended to be a customised app to Singaporean commuters as it provides estimated fare in addition to other route details.

2.2 Product Functions

There are 3 main product functions for this project:

1. The system shall accept route queries.
2. The system shall display sorted routes.
3. The system shall display route details.
4. The system shall display air quality and weather information.

2.3 User Classes and Characteristics

There are a total of 2 user classes that will be differentiated based on frequency of use for this project:

- Public transport commuters
 - Characteristic: Commuters, such as students and office employees, are our most important class of user as they can use our app on a daily basis to plan their journey efficiently.
- Tourists
 - Characteristic: Tourists may use our app as Farely is user friendly and provides a detailed step-by-step route direction and map along with checkpoints which helps them travel around Singapore.

2.4 Operating Environment

1. The system shall require a minimum of **Android versions 9.0**.
2. The Android phone must have enough storage space to install the Android app.
3. Google Map and GPS **must be** enabled to be able to use the Google Map.

2.5 Design and Implementation Constraints

1. **Performance:** The user query might sometimes take much time due to data.gov.sg API response time.

2. **Consistency:** The google API response for a specific origin and destination might be different which leads to the number of routes returned for the user might be different for each time query with similar origin, destination and concession type.
3. **Scalability:** The google map direction API uses a pay-as-you-go pricing scheme which may hinder scalability of the project.

2.6 User Documentation

- A demo will be presented to aid users in using our application.
- An API documentation will be provided for the Android app and the backend server to guide the developers in maintaining the application.

2.7 Assumptions and Dependencies

- All **API calls** are successful.
- All **fare and route** information can be obtained from publicly available APIs provided by the government or the education provider.
- The available APIs provide **sufficient information** on route details needed.
- Users have internet access.
- Users have at least **Android version 9.0** to be able to install and use the Android application.
- The application can make **use of Google API for free**.

3. External Interface Requirements

3.1 User Interfaces

- The system's user interface design shall follow Google's material design user interface specification.
- The Google Map should be able to be emerged in the application interface.
- The minimum screen size is 426 * 320 - which corresponds to the minimum Android phone size in the market today.

3.2 Hardware Interfaces

- Android Phone.
- Google Map and GPS should be enabled.
- The memory should be at least 5.1 Mb to be able to install the Android App.

3.3 Software Interfaces

- The system must be able to fetch API data from Data.gov.sg.
 - The system must be able to fetch API data from the dataset "Fares for Express Bus Services".
 - The system must be able to fetch API data from the dataset "Fares for Trunk Bus Services".
 - The system must be able to fetch API data from the dataset "Fares for Feeder Bus Services".
 - The system must be able to fetch API data from the dataset "Fares for MRT and LRT".
- The system must be able to fetch API data from the Land Transport Authority's DataMall.
 - The system must be able to fetch API data from the dataset "Bus Services".
- The system must be able to fetch API data from the Google Maps API.
 - The system must be able to fetch API data from Google Maps' Places API.
 - The system must be able to fetch API data from Google Maps' Geocoding API.
 - The system must be able to fetch API data from Google Maps' Directions API.
- The system must be able to fetch API data from the AirVisual API.

3.4 Communications Interfaces

- The system shall use the HTTPS protocol for the backend APIs.
- The communication protocol is encrypted using Secure Sockets Layer.

4. System Features

4.1 Accept Route Queries

4.1.1 Description

The system shall allow users to enter route queries. Route queries include the departure location, arrival location and fare type.

- Priority: High
- Benefit: 8
- Penalty: 8
- Cost: 5
- Risk: 2

4.1.2 Stimulus/Response Sequences

Stimulus: The user input the location name.

Response: The system displays candidate location name as the user types.

Stimulus: The user entered the departure and arrival location and selected the fare type and clicked the “search” button.

Response: The system displays a throbber animation while processing the query and displays a list of best routes after a few seconds.

Stimulus: The user taps on a route.

Response: The system displays a map which shows the direction and list of direction steps.

4.1.3 Functional Requirements

REQ 1: A user must be able to enter their departure location and arrival location.

REQ 1.1: The system must be able to accept the same types of location input formats for both departure location and arrival location.

REQ 1.1.1: The system must be able to accept a user’s current location as a location input.

REQ 1.1.2: The system must be able to accept a user’s plaintext search as a location input.

REQ 1.2: The system must be able to validate the location inputs for both departure location and arrival location.

REQ 1.2.1: If the system cannot interpret the user's departure location or arrival location, the system must display an error message.

REQ 2: The system must allow users to personalise their route queries according to their information.

REQ 2.1: A user must be able to select the fare type.

4.2 Display Sorted Routes

4.2.1 Description

The system shall allow users to view the sorted routes. The system shall find the best possible routes from a departure location to an arrival location. The routes will be sorted by estimated travel time, estimated distance and estimated fare. There are 3 tabs which correspond to 3 sorted routes criterias.

- Priority: High
- Benefit: 10
- Penalty: 10
- Cost: 8
- Risk: 5

4.2.2 Stimulus/Response Sequences

Stimulus: User clicked on the by Price tab

Response: List of available routes returned in sorted price order which route with cheaper price display first.

Stimulus: User clicked on the by Distance tab

Response: List of available routes returned in sorted distance order which route with shorter distance display first.

Stimulus: User clicked on the by Time tab

Response: List of available routes returned in sorted time order which route with shorter route time display first.

4.2.3 Functional Requirements

REQ 1: When a user sends the system a route query, the system must return the most efficient routes from the origin to the destination.

REQ 1.1: The system shall determine the most efficient routes from GoogleMap Directions API.

REQ 2: The system must sort the routes returned.

REQ 2.1: The system must be able to sort the routes by estimated travel time.

REQ 2.2: The system must be able to sort the routes by estimated price.

REQ 2.3: The system must be able to sort the routes by estimated price.

4.3 Display Route Details

4.3.1 Description

The system shall allow users to view the details of a route. This includes the directions of the route.

- Priority: Medium
- Benefit: 6
- Penalty: 2
- Cost: 2
- Risk: 2

4.3.2 Stimulus/Response Sequences

Stimulus: User clicked on each route item.

Response: The app will display the route details including the details polyline, checkpoints, travel modes to each checkpoints as well as the directions details.

Stimulus: Users try to zoom in / zoom out the detailed map.

Response: The map is zoomed in / zoomed out w.r.t the user actions.

4.3.3 Functional Requirements

REQ 1: The system must display the estimated travel time of the route.

REQ 1.1: The estimated travel time must be displayed in hours and minutes.

REQ 1.2: The estimated travel time must be displayed to the nearest minute.

REQ 2: The system must display the estimated total fare of the route

- REQ 2.1: The estimated total fare must be displayed in SGD.
- REQ 2.2: The estimated total fare must be displayed to 2 decimal places of accuracy.
- REQ 3: The system must display the distance of the route
- REQ 3.1: The distance must be displayed in km.
- REQ 3.2: The distance must be displayed to 2 decimal places of accuracy.
- REQ 4: The system must display the directions of the route.
- REQ 4.1: The system must display the information for each step in the route.
- REQ 4.1.1: If the step involves walking, the system must display the information about the departure and arrival locations in this step.
- REQ 4.1.1.1: The system must display the names of the departure location and arrival location in this step.
- REQ 4.1.2: If the step involves taking a bus or train, the system must display information about taking the bus or train.
- REQ 4.1.2.1: The system must display bus or train should be taken in this step.
- REQ 4.1.2.2: The system must display the names of the departure stop and arrival stop in this step.
- REQ 4.2: The system must display the steps in the order that they are to be taken.

4.4 Display Air Quality and Weather Information

4.4.1 Description

The system shall allow users to view real updates of air quality and weather conditions of selected locations.

- Priority: Low
- Benefit: 4
- Penalty: 1
- Cost: 1
- Risk: 1

4.4.2 Stimulus/Response Sequences

Stimulus: User entered location into either start location or destination.

Response: The app will air quality index on the button at the bottom right corner.

Stimulus: Users click on the air quality index button.

Response: The displays details of air quality, temperature and weather.

4.4.3 Functional Requirements

REQ 1: The system must display the air quality index.

REQ 2: The system must allow users to see details of the air quality, temperature and weather.

REQ 2.1: The air quality must be displayed in the air quality index.

REQ 2.2: The temperature must be displayed in degrees celsius.

REQ 2.3: The weather must be displayed as a description.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- When a user sends a route query to the system, the system must return the best routes within 20 seconds.
- The application must inform users if it is unable to return the routes.
- The application should be able to display the Air Quality Index for any place.

5.2 Safety Requirements

NA

5.3 Security Requirements

- The system shall use the HTTPS protocol for the backend APIs.

5.4 Software Quality Attributes

- Usability
 - 80% of first-time users must be able to enter a simple route query within 5 minutes of starting to use the system.
- Correctness
 - The system must estimate the travel time within 20 minutes of accuracy 80% of the time.
 - The system must estimate the total fare within 0.20 SGD of accuracy 80% of the time.

5.5 Business Rules

- The application only works for routes that start and end in Singapore.

Appendix A: Glossary

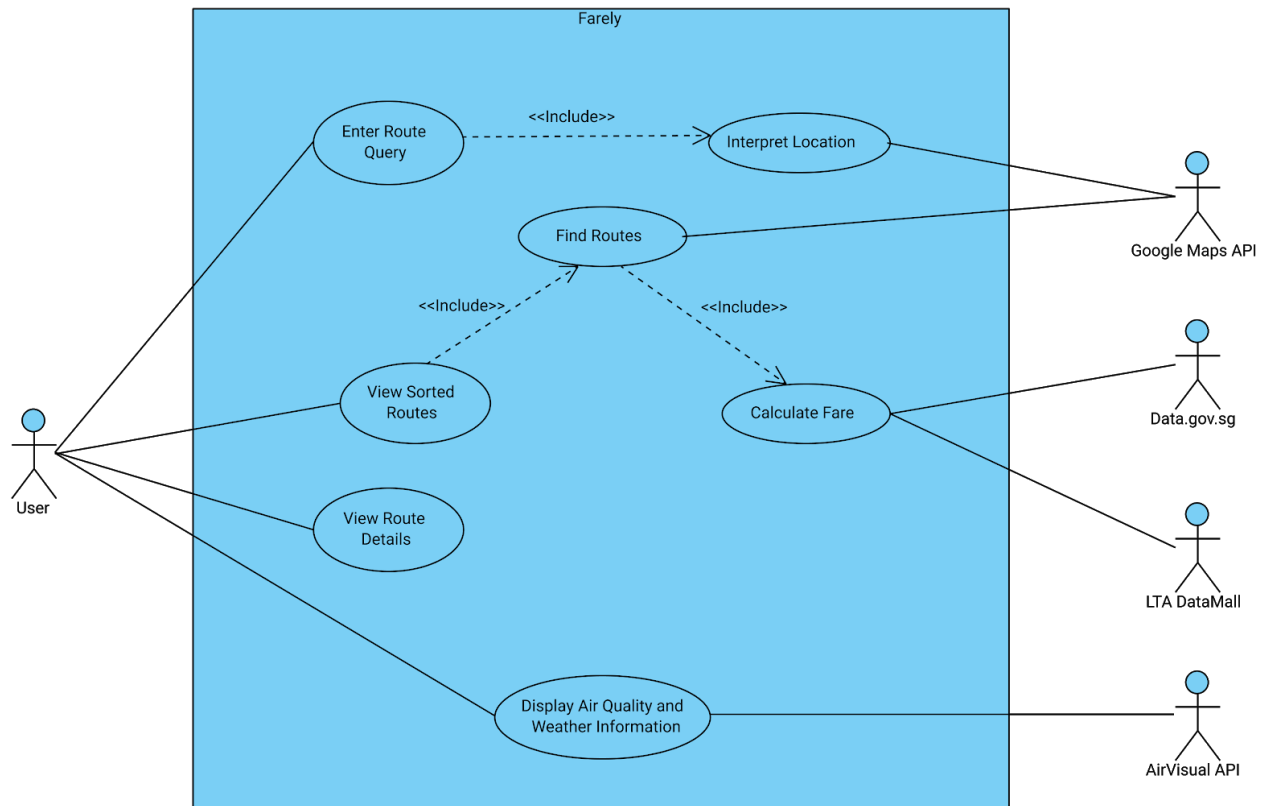
Term	Definition
Public Transport	Buses, trains, and other forms of transport that are available to the public, with charge set fares and run on fixed routes.
Route	The way taken in getting from an origin to a destination location. In this project, the routes must only include the usage of public transport and walking. The routes consist of multiple steps.
Route Query	The query sent to the system when a user requests for best routes from an origin to a destination location.
Departure Location	The location from which the user wants the route to start.
Arrival Location	The location to which the user wants the route to end at.
Step	This refers to a step in the directions of a route. A step consists of either taking a bus, taking a train or walking from one location to another.
Bus	A large motor vehicle carrying passengers by road, typically one serving the public on a fixed route and for a fare.
Train	A series of connected railway carriages or wagons moved by a locomotive or by integral motors. In this project, trains refer to the trains used in the MRT and LRT.
MRT	MRT stands for Mass Rapid Transit, which is a heavy rail rapid transit system that constitutes the bulk of the railway network in Singapore.
LRT	LRT stands for Light Rapid Transit, or Light Rail Transit, which is a series of localised automated guideway transit systems acting as feeder services to the heavy rail MRT.
Stop	This refers to a bus stop or train stop taken by a bus or train, respectively.
Fare Type	The type of pricing scheme used to calculate the price of the route. In this system, we will be considering the following 6 fare types: <ol style="list-style-type: none"> 1. Workfare transport concession card fare 2. Student card fare 3. Single trip 4. Senior citizen card fare 5. Persons with disabilities card fare 6. Adult card fare

Price	The total cost the user has to pay for public transport if they take the route.
Data.gov.sg	The Singapore government's one-stop portal to its publicly-available datasets from 70 public agencies.
DataMall	The Land Transport Authority's portal where it publishes a wide variety of land transport-related datasets for enterprises, third-party developers, researchers, and other members of the public to promote collaboration and co-creation of innovative and inclusive transport solutions.
Land Transport Authority (LTA)	A statutory board under the Ministry of Transport of Government of Singapore.
Places API	A service provided by Google Maps that converts a plaintext search to geographical locations (longitude and latitude).
Directions API	A service provided by Google Maps that calculates directions between locations.

Appendix B: Analysis Models

Use Case Model

Use Case Diagram



Use Case Description

Use Case ID:	1		
Use Case Name:	Enter Route Query		
Created By:	Leong Ko Rixie Tiffany	Last Updated By:	Leong Ko Rixie Tiffany
Date Created:	3 / 2 / 2020	Date Last Updated:	3 / 2 / 2020

Actor:	User (initiating actor)
Description:	The user enters a route query to find the best routes from a departure location to an arrival location.
Preconditions:	<ul style="list-style-type: none"> The user has launched the application.
Postconditions:	<ul style="list-style-type: none"> The system displays the fare type, departure location and arrival location entered by the user
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> The user enters their desired fare type. The system displays the entered fare type. The user enters their departure location using the included use case “Interpret Location”. The system displays the entered departure location. The user enters their arrival location using the included use case “Interpret Location”. The system displays the entered arrival location.
Alternative Flows:	<p>AF: The order of the user entering the required information does not matter. For instance, the user could select the fare type, then select the departure and arrival point.</p> <p>AF-S5: If the user wants to their current location for the departure location.</p> <ol style="list-style-type: none"> The user selects their current location. The system obtains the user’s current location through the GPS. The system displays and saves the user’s current location. The system returns to step 7.
Exceptions:	<p>EX1: If the user decides to use their current location, but the system cannot obtain the user’s current location.</p> <ol style="list-style-type: none"> The system prompts the user to enter their location using the included use case “Interpret Location” instead of using their current location.
Includes:	Interpret Location
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	2		
Use Case Name:	Interpret Location		
Created By:	Leong Ko Rixie Tiffany	Last Updated By:	Leong Ko Rixie Tiffany
Date Created:	2 / 2 / 2020	Date Last Updated:	18 / 3 / 2020

Actor:	User (initiating actor), Google Maps API
Description:	The user enters a location for either the departure location or arrival location that is interpreted using Google Maps API.
Preconditions:	<ul style="list-style-type: none"> The user has launched the application.
Postconditions:	<ul style="list-style-type: none"> The system saves the location.
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> The user taps the text input for the location input. The user enters the text of the location. The system sends a request to the Google Maps' Places API for autocomplete suggestions. If suggestions are found, the system displays the autocomplete suggestions. The system prompts the user to select one of the suggestions. If the user selects one of the suggestions, the system will display the location selected. The user presses enter. The system saves the location selected.
Alternative Flows:	<p>AF-S4: If Google Maps' Places API cannot find any suggestions.</p> <ol style="list-style-type: none"> The system does not display any suggestions. The system returns to step 4. <p>AF-S6: If the user doesn't select one of the candidate locations.</p> <ol style="list-style-type: none"> The user presses enter. The system sends a request to Google Maps' Places API to interpret the location. If the location can be found, the system displays the location and returns to step 8. If the location cannot be found, the system displays an error message and the system returns to step 2.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	3		
Use Case Name:	View Sorted Routes		
Created By:	Leong Ko Rixie Tiffany	Last Updated By:	Leong Ko Rixie Tiffany
Date Created:	3 / 2 / 2020	Date Last Updated:	18 / 3 / 2020

Actor:	User (initiating actor)
Description:	The user views the best routes from the departure location to the arrival location, sorted by price, time or distance.
Preconditions:	<ul style="list-style-type: none"> The user has successfully made a route query.
Postconditions:	<ul style="list-style-type: none"> The system displays a list of the best routes sorted by price or time. The system displays a message “No results found”.
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> The system gets the best routes, their estimated travel times and distance using the included use case “Find Routes” based on the route query. If the user selects the mode to sort by price, the system will sort the routes by their estimated price, from cheapest to most expensive. The system displays the list of routes with their respective directions, estimated price, estimated travel time and estimated distance.
Alternative Flows:	<p>AF-S3: If the user changes the mode to sort by time.</p> <ol style="list-style-type: none"> The system sorts the routes by their estimated travel time, from shortest travel time to longest travel time. The system returns to step 4. <p>AF-S3: If the user changes the mode to sort by distance.</p> <ol style="list-style-type: none"> The system sorts the routes by their estimated distance, from shortest distance to longest distance. The system returns to step 4.
Exceptions:	<p>EX1: If the fare is a null value.</p> <ol style="list-style-type: none"> The system displays that the total price is unavailable. If the mode is to sort by price, the system will place the route at the bottom of the list.
Includes:	Find Routes
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	4		
Use Case Name:	Find Routes		
Created By:	Tran Anh Tai	Last Updated By:	Tran Anh Tai
Date Created:	3 / 2 / 2020	Date Last Updated:	5 / 2 / 2020

Actor:	Google Maps API
Description:	The system returns the best routes from the departure location to the arrival location.
Preconditions:	<ul style="list-style-type: none"> The user has successfully made a route query.
Postconditions:	<ul style="list-style-type: none"> The system saves the best routes from the departure location to the arrival location The system displays an error message if no routes are found.
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> The system sends the route query to Google Maps' Directions API. Google Maps' Directions API returns all the available routes from departure location to the arrival location. The system calculates the estimated price of each route using the included use case "Calculate Fare". The system saves the best routes and their details.
Alternative Flows:	
Exceptions:	EX1: If no routes are found. <ol style="list-style-type: none"> The system will display an error message "No results found".
Includes:	Calculate Fare
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	5		
Use Case Name:	Calculate Fare		
Created By:	Chulpaibul Jiraporn	Last Updated By:	Chulpaibul Jiraporn
Date Created:	3 / 2 / 2020	Date Last Updated:	18 / 3 / 2020

Actor:	Land Transport Authority's DataMall API, Data.gov.sg
Description:	The system calculates the total fare for a route.
Preconditions:	<ul style="list-style-type: none"> • The user has successfully made a route query. • The system has found available routes from departure to arrival location using Google Map API.
Postconditions:	<ul style="list-style-type: none"> • The system returns the total price for the route. • The system returns a NaN value
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> 1. The system fetches bus, MRT and LRT fare data from data.gov.sg API. 2. The system fetches the bus service information from the Land Transport Authority's DataMall API. 3. The system calculates the total price using the data queried from Land Transport Authority's DataMall and Data.gov.sg API.
Alternative Flows:	
Exceptions:	EX1: If the system cannot fetch fare data from one of the API or both of the API. <ol style="list-style-type: none"> 1. The system returns a null value for the total price.
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	6		
Use Case Name:	View Route Details		
Created By:	Tran Anh Tai	Last Updated By:	Leong Ko Rixie Tiffany
Date Created:	3 / 2 / 2020	Date Last Updated:	4 / 2 / 2020

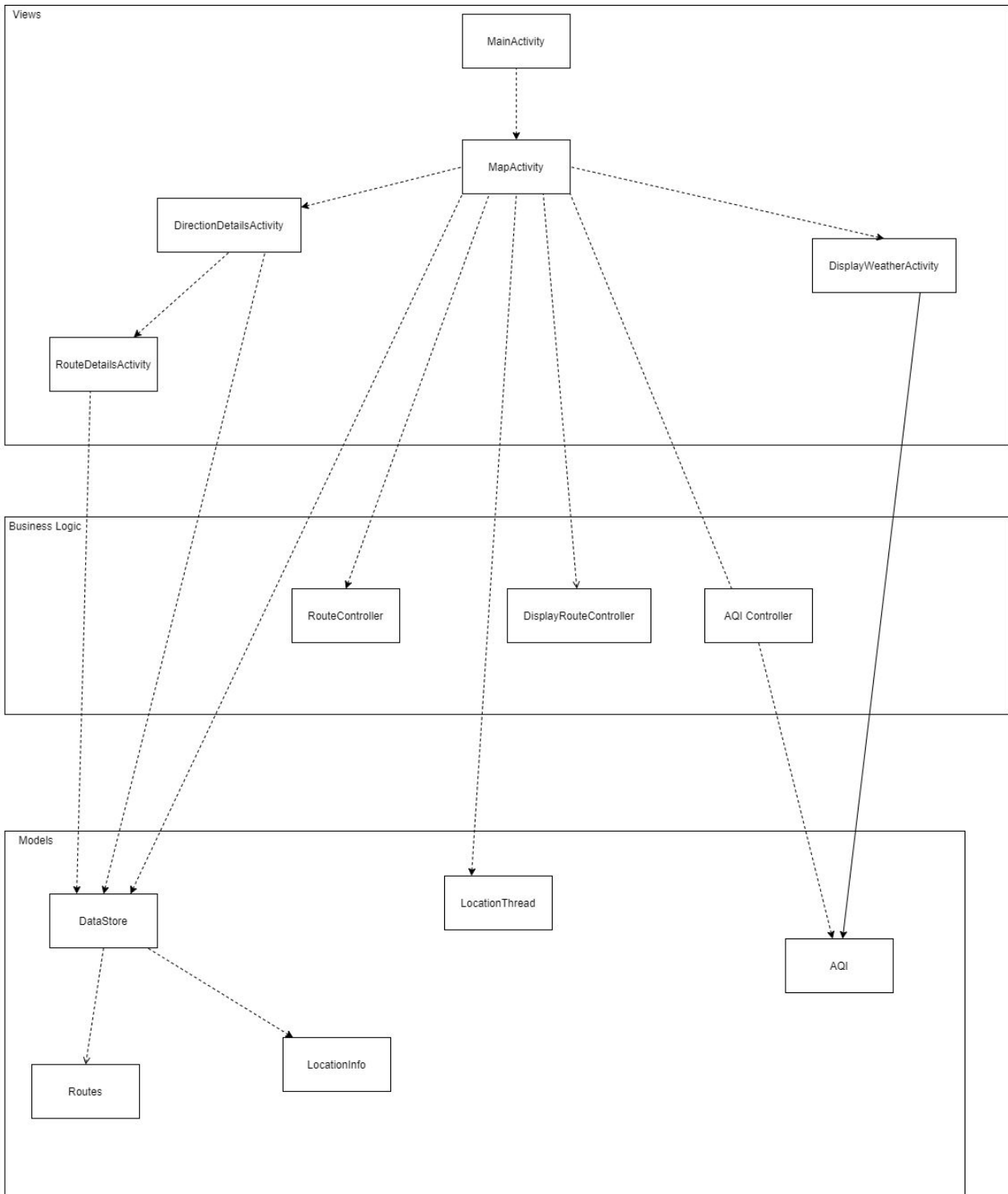
Actor:	User (initiating actor), Google Maps API
Description:	The system displays the details of a selected route from the departure location to the arrival location.
Preconditions:	<ul style="list-style-type: none"> • The user has successfully made a route query. • The system has found available routes from departure to arrival location using Google Map API.
Postconditions:	<ul style="list-style-type: none"> • The system displays the details of the route, including the directions of the route.
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects a route from the list of routes to view the details of. 2. The system retrieves the details of the route from the Google Maps' Directions API. 3. The system displays the details of the route, including the directions of the route. 4. For each step in the directions of the route, the system will display the details of the step. 5. If the step involves walking, the system will display the names of the departure location and arrival location of the step, and estimated time taken for the step. 6. If the step involves taking a bus or train, the system will display the names of the departure stop and arrival stop, the number of stops and estimated time taken for the step.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID:	7		
Use Case Name:	Display Air Quality and Weather Information		
Created By:	Chulpaibul Jiraporn	Last Updated By:	Chulpaibul Jiraporn
Date Created:	23/03/2020	Date Last Updated:	23/03/2020

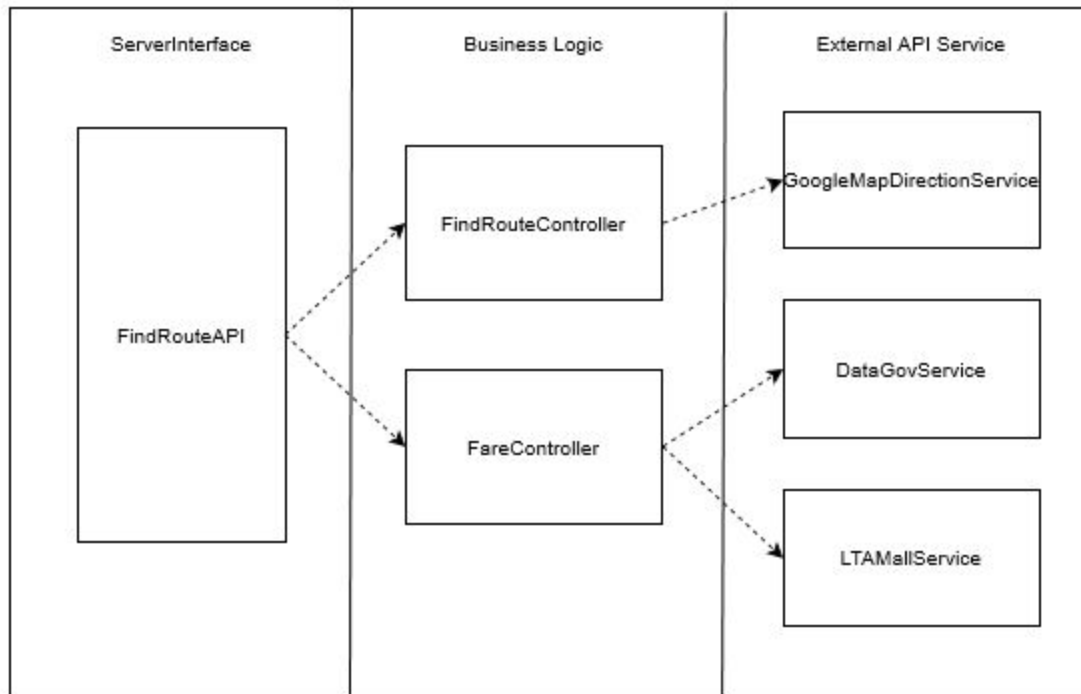
Actor:	User (initiating actor), AirVisual API
Description:	The system displays the information of the air quality and weather details
Preconditions:	<ul style="list-style-type: none"> • The user launched the application.
Postconditions:	<ul style="list-style-type: none"> • The system displays air quality and weather details.
Priority:	
Frequency of Use:	
Flow of Events:	<ol style="list-style-type: none"> 1. The system uses the current location of the map to retrieve air quality and weather information from AirVisual API. 2. The system displays the air quality index at the bottom right corner of the home page. 3. The user clicks the air quality button. 4. The system displays the details of air quality and weather.
Alternative Flows:	
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

System Architecture

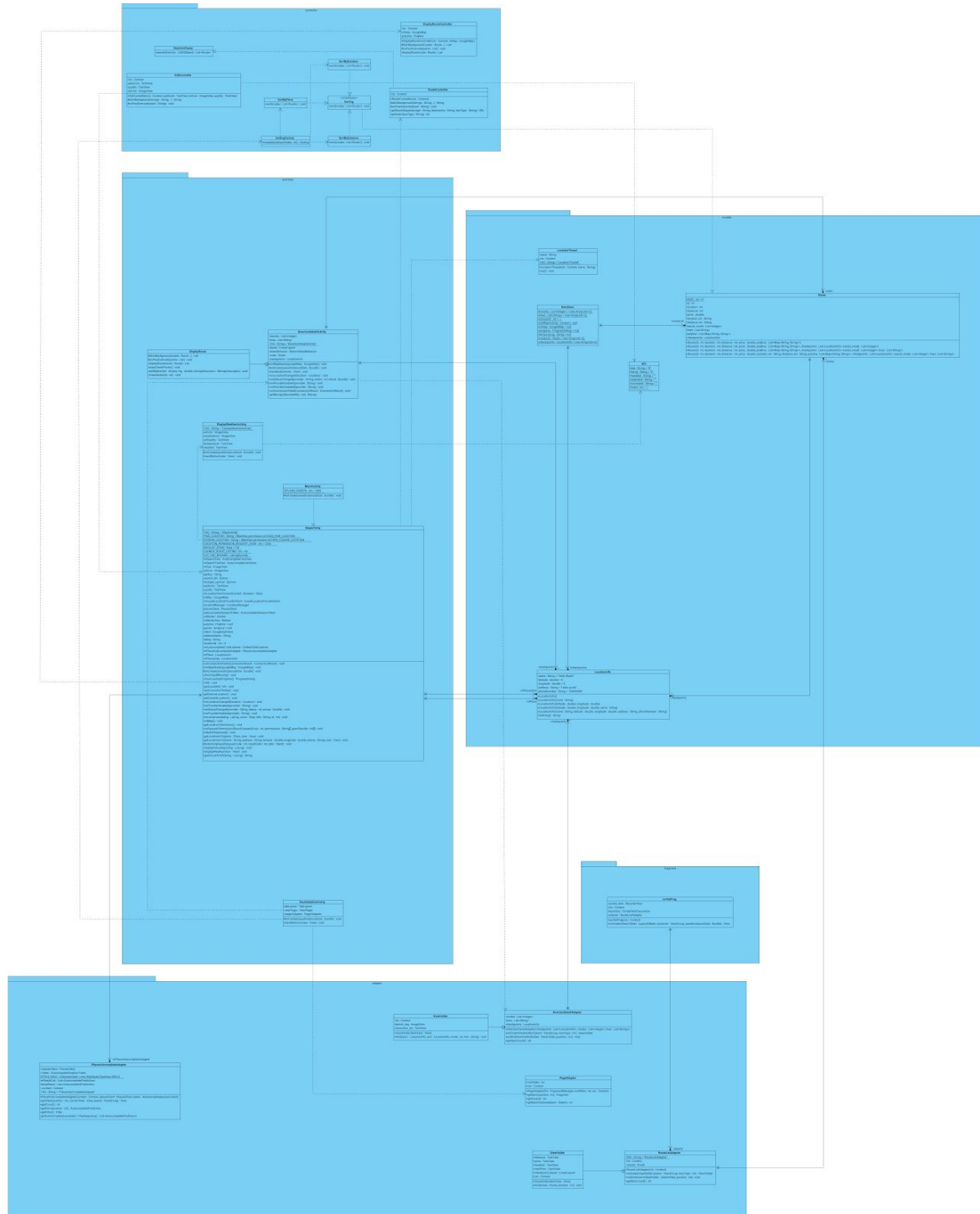
Frontend



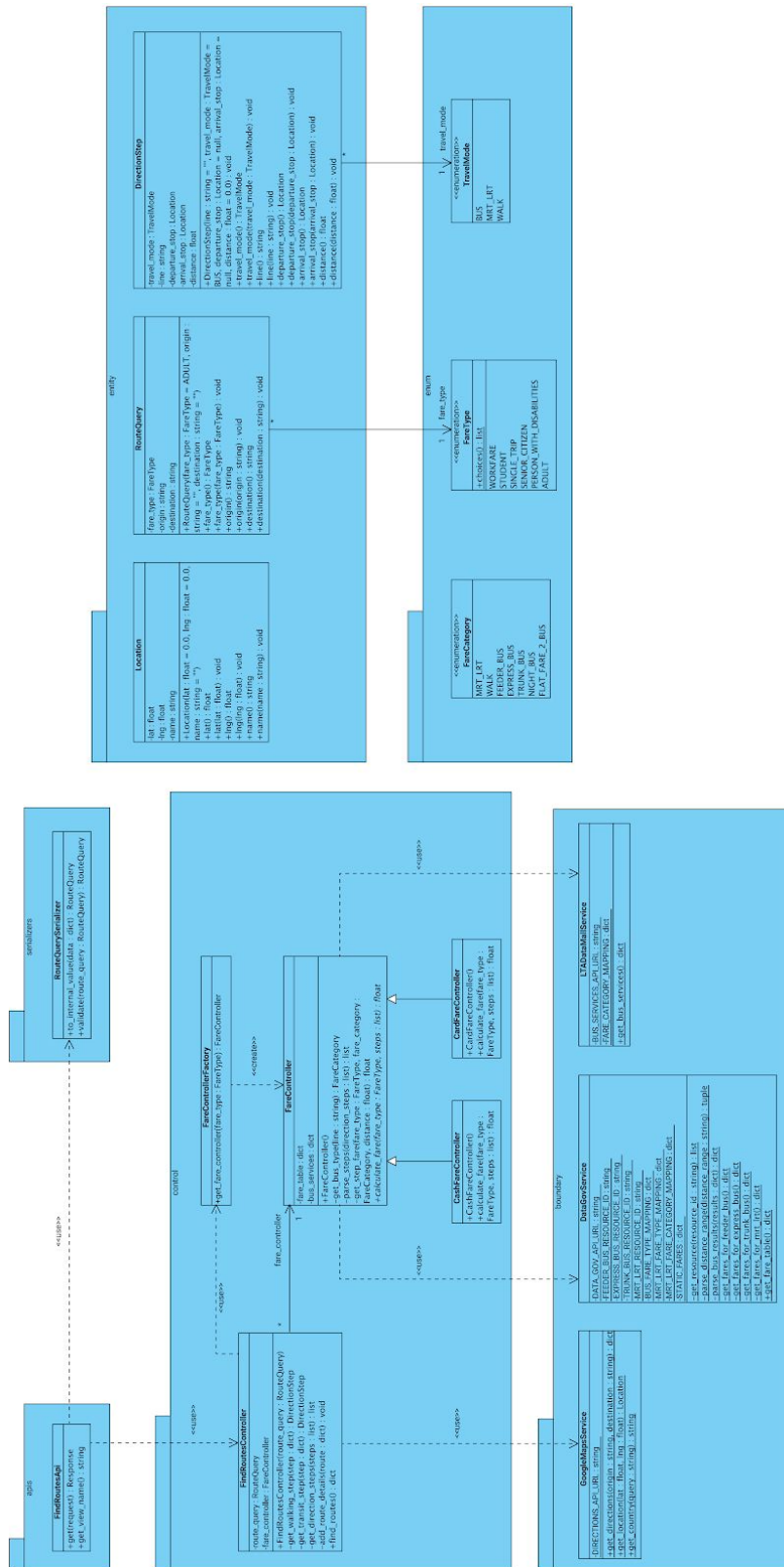
Backend



Frontend

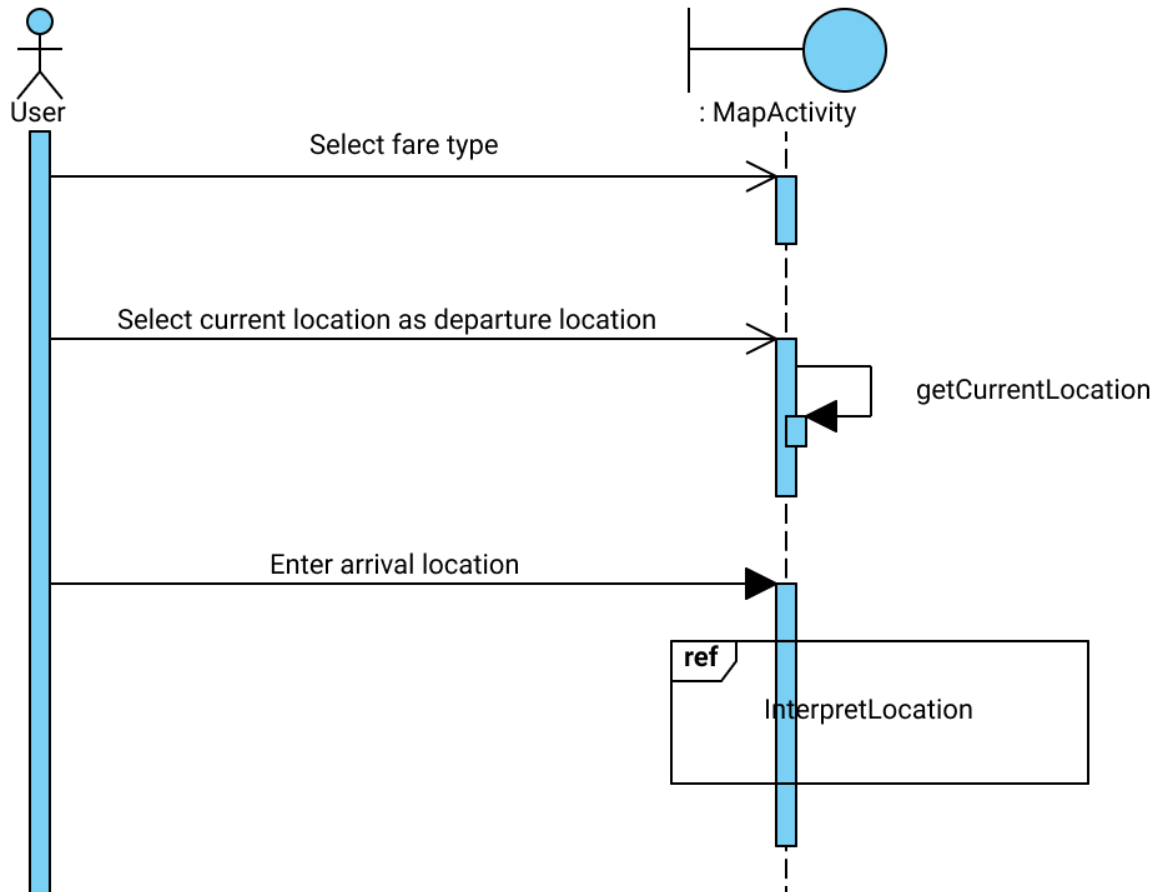


Backend

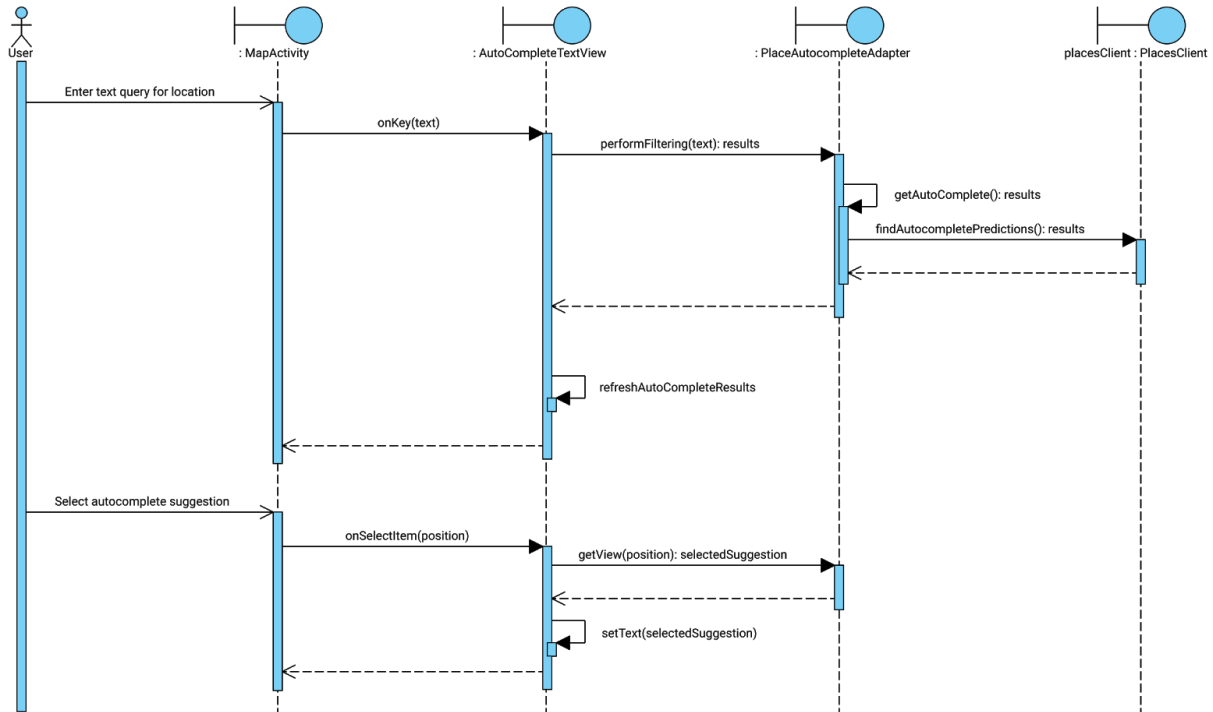


Sequence Diagrams

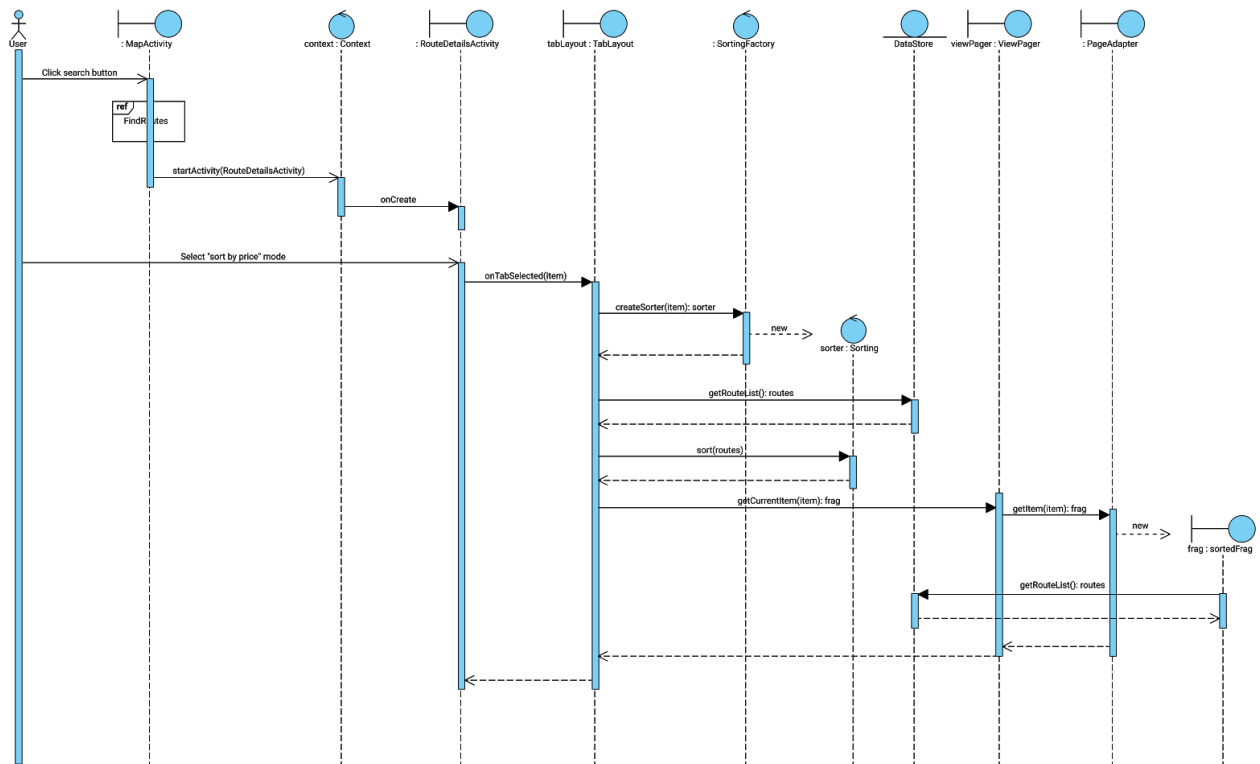
Enter Route Query



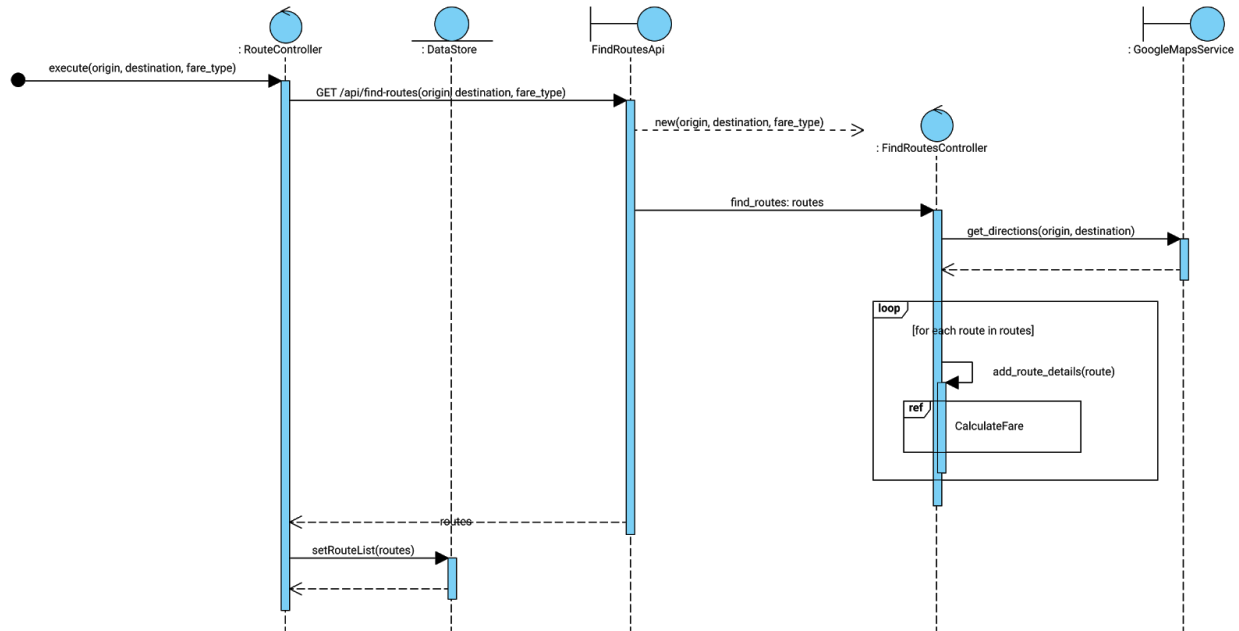
InterpretLocation

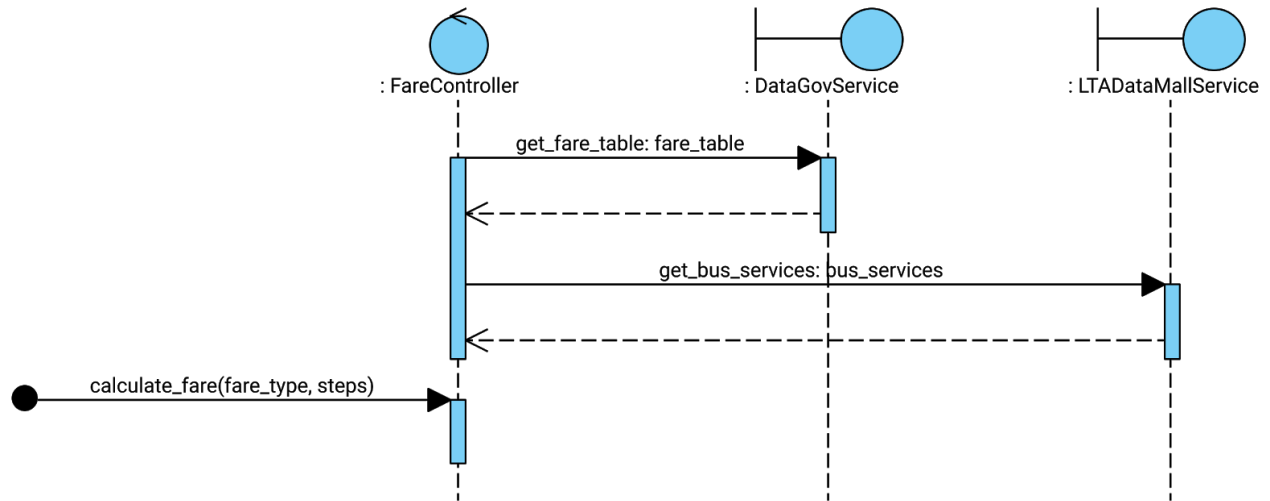


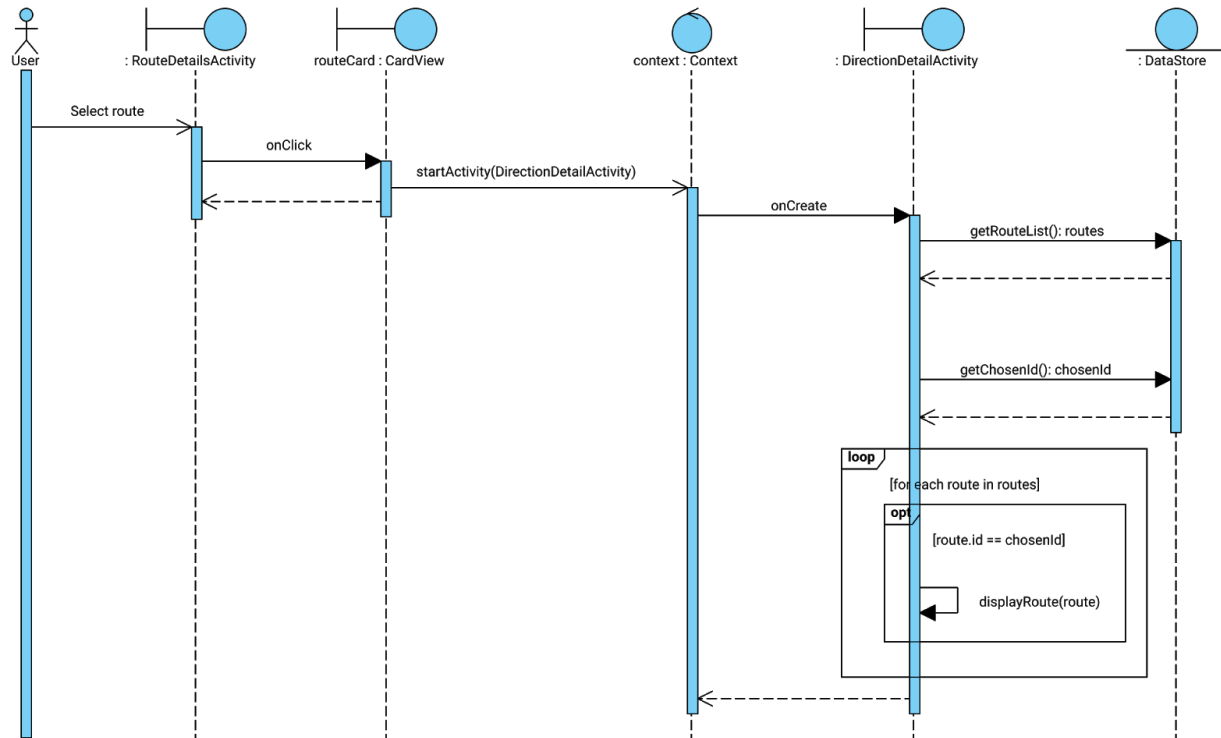
ViewSortedRoutes

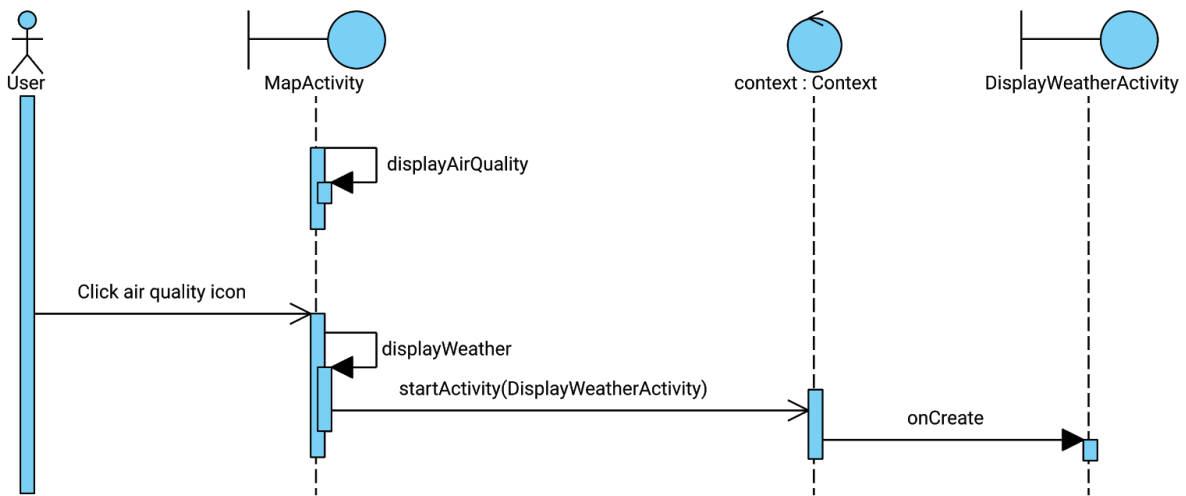


FindRoutes

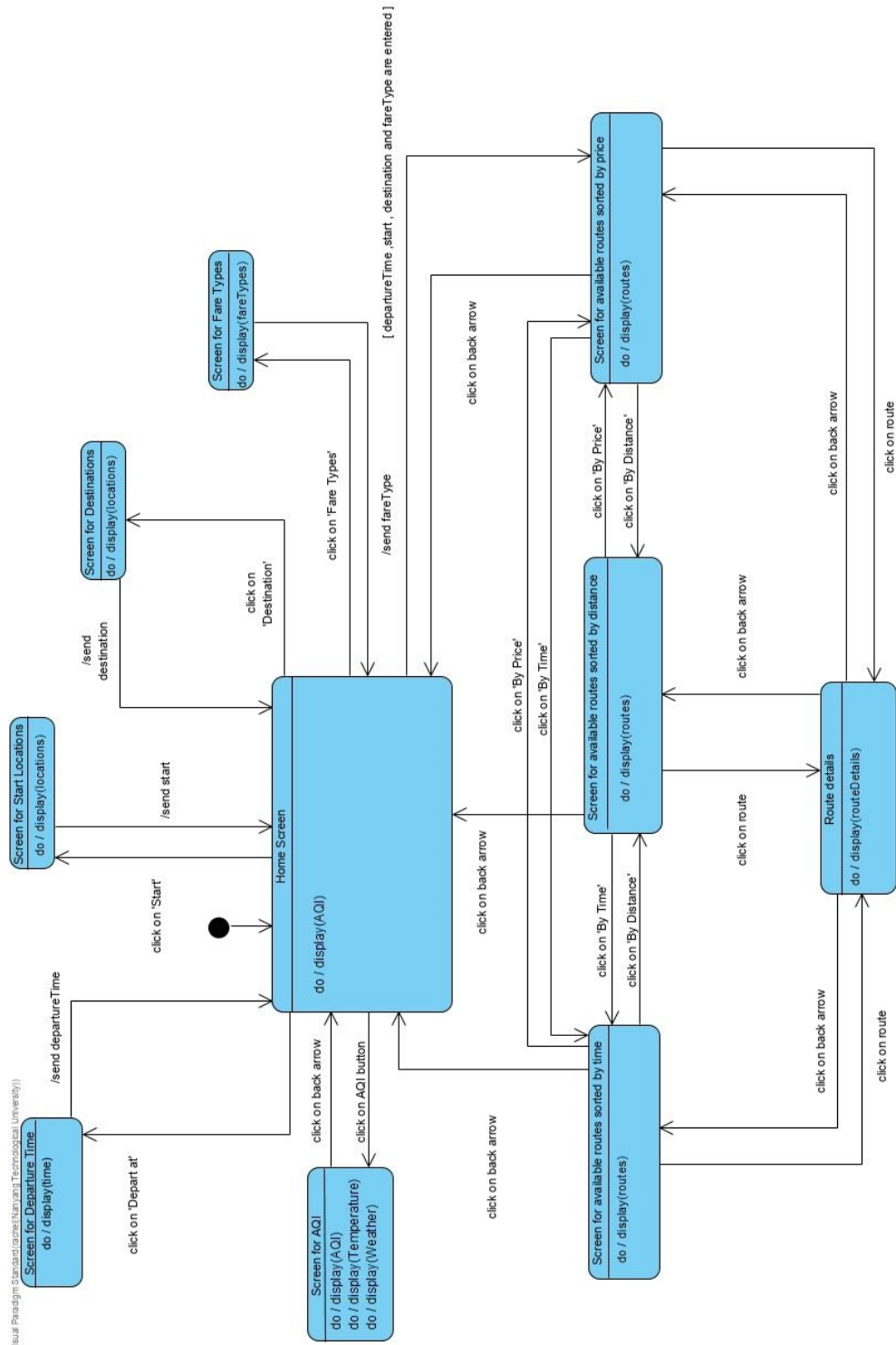


CalculateFare

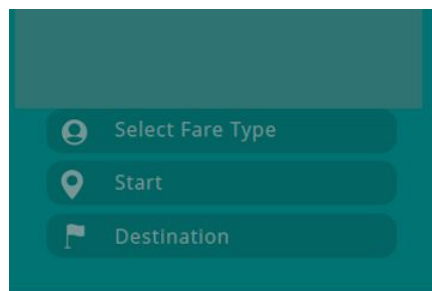
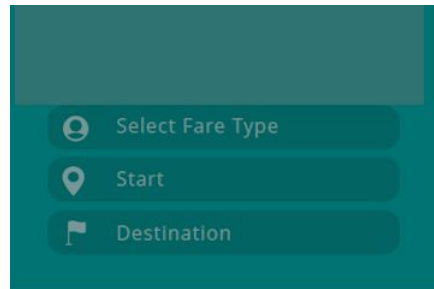
ViewRouteDetails

DisplayAirQualityAndWeatherInformation

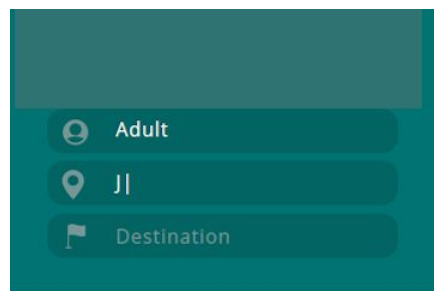
Dialog Map



UI Mockups



- ☒ Adult
- ☐ Student
- ☐ Senior Citizen
- ☐ Workfare Transport
- ☐ Person with Disabilities



- ☒ Current Location
- ☐ Junction 8
- ☐ Jurong Bird Park
- ☐ Jurong West
- ☐ Jurong East Mrt

