# **Public Transportation App**

## **Description**

Designing an application that will ease one in his/her day to day travels. Simply by inputting the user’s current location, the desired destination, the current bus/mrt user is on, the application will inform the user on when to get off. This is to address the issue where people do not know when their stop is arriving and to lessen the confusion and stress when travelling to a new place.

**Application**: Mobile App

**Language:** Java

## **1. Documentation of functional and non-functional requirements**

We wish to target the following users with our innovative application :

* Students and Office workers
* Tourists
* Elderly

**Functional requirements:**

1. The user must have the following abilities using the application
   1. User shall check the various bus routes of each bus
   2. The user must be able to input the following details on the app
      1. User shall be able to enter the origin of the trip.
         1. The user shall enter input as an address.
         2. The user shall enter input the bus stop code too.
         3. The user shall enter input the MRT station code also.
         4. The user shall give location using the option of giving live location.
      2. User should be able to enter the destination of the trip.
         1. The user shall enter input as an address.
         2. The user shall enter input the bus stop code too.
         3. The user shall enter input the MRT station code also.
         4. The user shall give location using the option of giving live location.
      3. Users shall plan trips at the moment or shall plan a trip for the future.
   3. The user shall favourite trips to reuse in the future.
   4. The usrr shall remove trips from Route History
2. The system shall convert the user input into data.
   1. The system must parse a manual address into a geolocation using relevant API.
   2. If user requests live location, the system shall use location services to detect the location of the user.
   3. Date and time entered by the user shall be converted to the desired format for processing.
3. The system shall perform computations to calculate routes for the user.
   1. System shall retrieve information about bus stop location from LTA Datamall - Bus stops
   2. System shall retrieve information about MRT location from Data.gov.sg - [LTA MRT Station Exits](https://data.gov.sg/dataset/lta-mrt-station-exit) / [SDCP MRT Station Point](https://data.gov.sg/dataset/sdcp-mrt-station-point)
   3. System shall retrieve travel fare information from Data.gov.sg - [Fares for MRT and LRT](https://data.gov.sg/dataset/fare-structure-mrts-and-lrts) / [Fares for MRT and LRT](https://data.gov.sg/dataset/fare-structure-mrts-and-lrts) / [Fares for Feeder Bus Services](https://data.gov.sg/dataset/fare-for-feeder-bus-services) / [Fares for Express Bus Services](https://data.gov.sg/dataset/fare-for-express-bus-services)
   4. System shall retrieve information about bus route from LTA Datamall - Bus Routes
   5. System shall calculate list of available routes using Google Maps API.
   6. System shall provide the list of routes to the user
   7. System shall display the cost of each route
   8. System shall display the time taken for each route
   9. During the trip, the system shall have additional functionalities.
      1. System shall retrieve real-time information about bus arrival from LTA Datamall - Bus Arrival Data
      2. System shall be able to track a specific vessel in real time
      3. System shall display the vessel movement in real time
      4. System shall make suggestions if the stop is missed
      5. The system must alert the user when the next stop is the destination
   10. System must be able to store one or more “Favorite” routes
   11. This system shall work for Android devices.
   12. The system shall create a userID for every user when the user downloads the app.

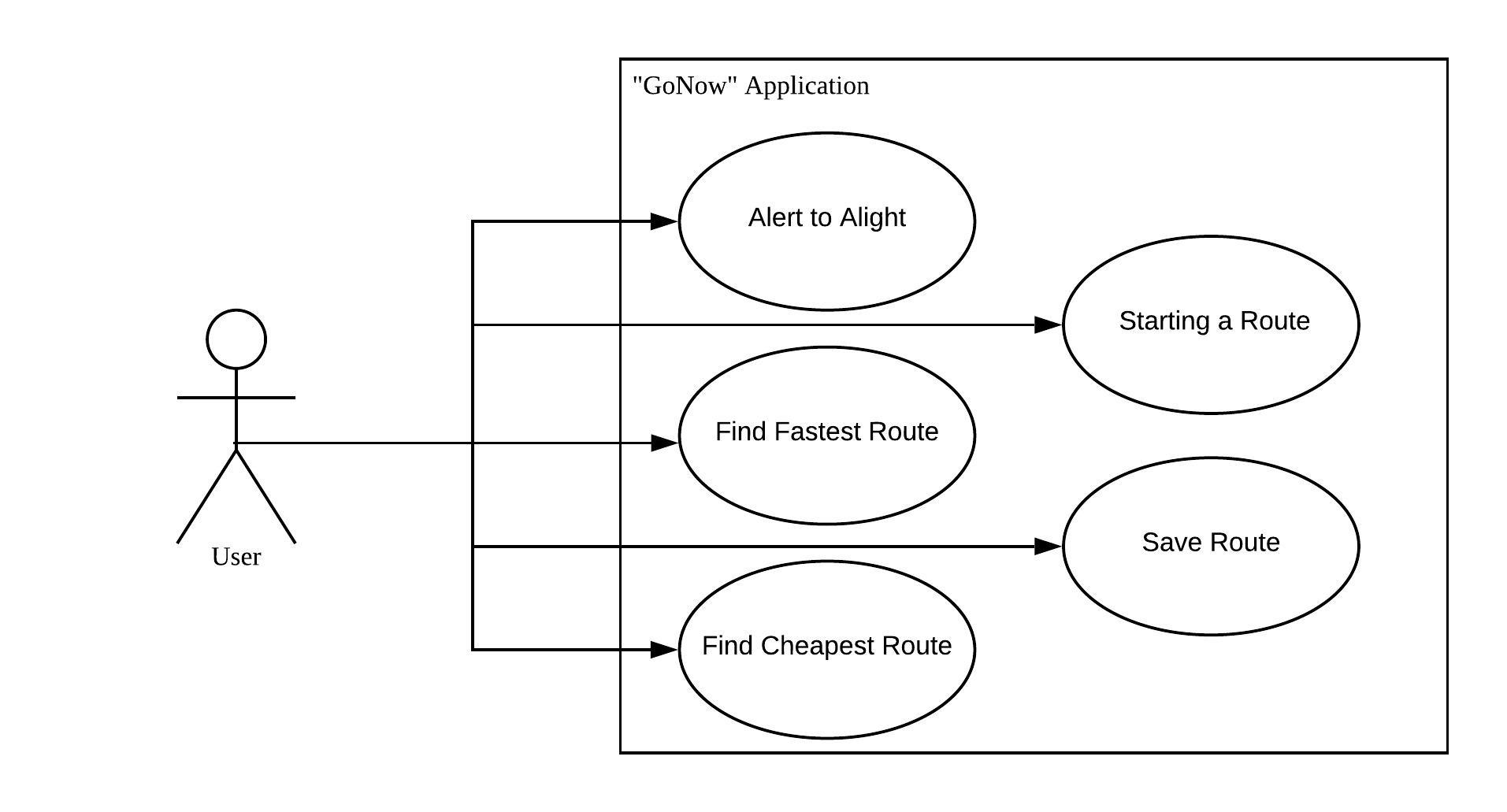
**Non-functional requirements:**

1. 99 % of users must be able to enter a simple query within 2 minutes of starting the system
2. After a system reboot, the restoration to full system functionality must not take longer than 3 minutes.
3. When the user is on the MRT/bus, the system shall to detect if the next stop is the destination within 1 minute.
4. System shall be able to calculate cheapest/ fastest route within 1 minute
5. Installation of the application must not take more than 5 minutes.
6. On missing the transport, the app should recalculate a new route within 2 minutes.
7. System response time must not exceed 1 minute.

## **2. Data dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| Term | Definition | Attributes | Relationship between terms |
| Bus | A large motor vehicle carrying passengers by road, serving the public on a fixed route for a fare. | Text | A type of transportation mode |
| Bus Service | The bus service number, e.g. 179. | Numerical | A bus can be operating in a specific service. |
| Bus Stop | A place where the bus regularly stops, usually marked by a sign. The bus stop can be represented by its name or code. | Text  Numerical | See definition |
| Bus Stop location | The location of a bus stop expressed in terms of coordinates. | Numerical | See definition |
| Mass Rapid Transport (MRT) | Fast moving commuter train that runs on a rail, bringing commuters from one station to another. | Text | A type of transportation mode |
| MRT Stop | A place where the MRT regularly stops. | Text  Numerical | See definition |
| MRT Stop location | The location of an MRT stop expressed in terms of coordinates. | Numerical | See definition |
| User | People in Singapore who are trying to make use of public transport to get from one place to another. Identified by a userID. | Numerical | The user interacts with the system by entering input or by performing actions. |
| System | The mobile application. | Text | Encompasses all other terms. |
| Input | Information of different forms. This information could for example the name of a bus stop, fare type or departure time. | Text  Numerical | Provided by the user and entered into the system. |
| Output | The result of the query created by the user from the database | List | A list of routes. |
| Origin / start station | The bus stop or MRT station from which the user would like to travel. | Text  Numerical | A specific bus stop or MRT stop. |
| Destination / end station | The bus stop or MRT station the user would like to travel to. | Text  Numerical | A specific bus stop or MRT stop. |
| Route | The route is a geographical description of how the user will get from the origin to the destination. | Numerical | Draws a connection between the origin and the destination. |
| Possible routes | The different routes a user can take to get from origin to destination. | List | A list of routes. |
| "Favorites" | An option available to users which enables them to store specific routes. | List | Saves specific routes. |
| Mode of transportation | The users will be able to travel by bus or MRT. As such, these are two types of modes of transportation. | Text | Represents bus or MRT. |
| Vessel | The specific mode of transportation the user is onboard, i.e. the specific bus or MRT the user is on expressed in coordinates. | Numerical | Represents bus or MRT. |
| Departure Time | The point in time when the vessel will leave from the origin. One or more alternatives will be given?? | Date/Time | See definition |
| Arrival Time | The point in time when the vessel will reach the destination. | Date/Time | See definition |
| User Location | The coordinates of the user .This location is used to track the users movements and alert when they are approaching the destination. | Numerical | Tied to a specific user. |
| Pre plan | Route with starting time at a time > 15 minutes | List | A type of route |
| Real time plan | Route with starting time < 15 minutes | List | A type of route |
| Distance | Distance travelled during trip. | Numerical | The distance between origin and destination. |
| Fare type | A category for the traveller to be put in, e.g. Student, Adult. | Text | Categorizes the user. |
| Fare | The price for a specific trip. | Numerical | The price for a specific mode of transportation, distance and fare type. |
| Price | The price the user will have to pay for their trip. | Numerical | The price is calculated from the mode of transportation, distance, fare and fare type. |
| Cheapest route | The route holding the cheapest price the user can pay for getting from origin to destination. | Numerical | Decided through a comparison of routes and their respective prices. |

## **3. Initial Use Case Model (consisting of Use Case diagram and Use Case descriptions)**

**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | ROUTE-1 | | |
| Use Case Name: | Finding the fastest route | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Displaying for the user, the fastest way to get from origin to destination by bus and/or MRT |
| Preconditions: |  |
| Postconditions: | Display the various routes and their timing needed |
| Priority: | High |
| Frequency of Use: | Once a week |
| Flow of Events: | 1. The user selects the option to “Plan My Trip”  2. The system prompts the user with options of “Bus” and “MRT”  3. The user checks the wanted options and selects the Arrow icon to move to the next step  4. The system asks the user for the starting point and the ending point  5. User selects the starting point fill-in box and enters in the starting address  6. System verifies that it is a valid address with Google Maps API  7. If system is verified by the system, the system prompts the user for the address of the destination.  8. User selects the ending point fill-in box and enters the ending destination  9. System verifies that it is a valid address with Google Maps API  10. If address is verified by the system, the system prompts  11. User selects the Arrow Icon to proceed  12. System asks the user for time and date and option of “Time of Arrival” or “Time of Departure”  13. User fills in time and date and selects the wanted option, followed by selecting the next option  14. The system calculates the fastest route with Google Maps API and lists possible routes sorted by time |
| Alternative Flows: | AFS-7 & 10:If address entered cannot be verified   1. The system displays the message “Location if not found. Please try again.” 2. System returns to previous step ( ie step 6 if AFS7, step 9 if AFS 10)   AFS5: If the user uses GPS to set starting location   1. User selects the “Your Location” option in the drop down menu 2. System converts live location to a Geolocation with the Google Maps API 3. System returns to step 8   AFS5 & 8: If the user uses a previous Favourite Location   1. User selects a previous Favourite Location option in the drop down menu 2. System retrieves the geolocation of the Favourite Location from memory 3. System returns to the next step |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | System would always be able to find a route from users origin to their destination |
| Notes and Issues: | If the user is planning a journey when MRT/bus services are not available (e.g outside of operating hours), the system will show the next possible route. |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | ROUTE-2 | | |
| Use Case Name: | Finding the cheapest route | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allowing the user to find the cheapest possible route |
| Preconditions: | This use case extends ROUTE-1 |
| Postconditions: |  |
| Priority: | High |
| Frequency of Use: | Once a week |
| Flow of Events: | 1. The user plans routes as per usual in ROUTE-1  2. Upon completing ROUTE-1, the user selects the option to sort route of lists by price  3. The system returns the list of possible routes sorted by price |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | ROUTE-3 | | |
| Use Case Name: | Save favourite routes | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Allowing the user to save routes for future use |
| Preconditions: | This use case extends either ROUTE-1 or ROUTE-2. |
| Postconditions: | Route is added to Favourite list |
| Priority: | Low |
| Frequency of Use: | Once a month |
| Flow of Events: | 1. The user views history of routes taken 2. The application shows a list of routes the user has taken sorted by time 3. The user can select a route previously taken to favourite 4. Application displays a message “Route added to favourites” |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | ROUTE-4 | | |
| Use Case Name: | Starting a route | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | User has chosen a route and wishes to take respective route with directions |
| Preconditions: | Extends Use case: Route 1/ Route 2.  User GPS is on. |
| Postconditions: |  |
| Priority: | High |
| Frequency of Use: | Daily |
| Flow of Events: | 1. The user selects their preferred route from list of routes calculated. 2. The systems get live location of user and displays a map showing where the user is and the route chosen. 3. System constantly updates the map shown based on changes in user location. 4. a) If the live location of user is near a bus stop and the chosen route includes taking a bus from that bus stop, the system would prompt for the next available bus in the LTA datamall API.   b) system displays the waiting time needed for the next bus  5. If the live location of user is within 5m of their destination, systems displays message “ You have arrived at your destination!”  6. Systems prompts if user wishes to save this route as one of their favourites route.  7. If user clicks on the option to save route, system would update its data and save this route to “Favourites Route” |
| Alternative Flows: |  |
| Exceptions: | EX1: User decides to cancel route   1. System displays “ You have chose to end the route” |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: | GPS tracking is fairly accurate |
| Notes and Issues: |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | ROUTE-5 | | |
| Use Case Name: | Alighting at the right stop | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actor: | User |
| Description: | Alerts user when the next stop is their intended destination |
| Preconditions: | The user has started a route |
| Postconditions: | User has alighted at the correct stop |
| Priority: | High |
| Frequency of Use: | Daily |
| Flow of Events: | 1. The application uses the user’s location to determine if the user is approaching the intended stop  2. When application determines the user is approaching the intended stop, the application issues a notification to the user  3. User acknowledges notification by selecting the “Yes” option  4. System provides instructions on the next part of the user’s route |
| Alternative Flows: | AF-S3: User does not acknowledge and misses the stop   1. Application notifies user “You have missed stop” 2. User selects option to reroute 3. User is brought back to Step 1   AF-S4: User has completed the route   1. Application notifies user “You have arrived at Your Destination!” |
| Exceptions: |  |
| Includes: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

## **4. UI Mockups**

***See attached files***