SC2006 Lab 1: Group

Team Members:

Mishra Pradyumn, U2123912E (Team Lead)

Kway Yi Shen, U2121308J (Vice Team Lead)

Veeraraghavan Srivathsan Nithyasri, U2120134J

Muhammad Irfan Bin Ameer Hamzah, U2122409L

Functions:

1. Login
2. Sign up
3. Favourite Locations
4. Display Fares
5. Check Fares
6. Determine Route

Functional Requirements:

1. The System must contain a login function
   1. User can key in a username and password
      1. Username and password must be text of length > 0 and < 512
      2. Password must contain at least 1 lowercase alphabet and 1 number
      3. Username and Password combination must be in the Registered User Database
         1. If not, the System must prompt the user to try again.
         2. If the user enters their username or password wrong three times, the System prompts the user to try again after 1 minute

1. The System must contain a sign up function
   1. User must key in a username and password
      1. Username and password must be in text of length > 0 and < 512
      2. Password must contain at least 1 lowercase alphabet and 1 number
      3. Username must not be in registered user database
      4. System must add in the username and password into Registered User Database

1. The system must be able to edit a User’s favourite location
   1. The User must be able to add selected locations
      1. The locations must be in Singapore
   2. The User must be able to delete selected locations
      1. The locations must be in Singapore
   3. The User can only have a maximum of 10 favorited locations

1. The System must display the total fare price when called
   1. Fare price must display the split of Public Transport and Private Transport
      1. Public Transport fare will be shown as a split of MRT fare and Bus fare
      2. Fare Prices must be displayed in SGD ($)
      3. Fare Prices must be displayed with 2 decimal points
      4. Fare Prices must be colour coded into different colours (red for MRT, Green for Bus, Blue for Private Transport)

1. The system must be able to determine a 2 distinct routes,  given a start and an end destination
   1. The public transport route with price and the private transport route with price is calculated
      1. The start and end destinations must be valid locations in Singapore
      2. The system must be able to check the fare of a given route in real time
      3. The system must be able to calculate prices dynamically from the API used
      4. The start and end destinations must not be the same

1. The final output shows the user the fare and distance of both Public and Private transport
   1. The user can further choose one of 2 options - Cheapest Price or Shortest Route
      1. Depending on the user’s preference, the Verdict is displayed

Non Functional

1. The Web Application buttons must be responsive within a 5 second time limit
2. The User must choose two unique points in the map as the to and from locations
3. The User must be able to login within 5 seconds
4. The price given must always be more than zero.
5. The detailed roadmap of Singapore should be displayed.

Assumptions

1. The User is based in Singapore
2. The User has a pre-defined start and end location
3. The User has a working device and internet connection

Data Dictionary:

|  |  |
| --- | --- |
| Bus Fare | Price to get from point A to point B on the bus |
| Taxi Fare | Price to get from point A to B on the Comfort Delgro Taxi |
| Bus Duration | Time it takes to get from point A to point B on the bus |
| Taxi Duration | Time it takes to get from point A to point B on the Comfort Delgro Taxi |
| Traffic | Flow of vehicles on the road |
| User | A person who will be using this specific application |
| Destination | The specific name of a place, where the user wants to go |
| Route | A Path from one location to another location |
| Verdict | The final output of which is the best option of transport |
| SGD | The official currency of Singapore |

Use case diagram:

Diagram

Description automatically generated

Flow of events:

1. Users will key in their username and password.
2. Registered User Database will verify details
3. User will be logged in
4. Users will key in their Username, Password and Confirm Password
5. Registered User Database will update with Username and Password
6. User will be logged in
7. User will save or unsave a specific location
8. Registered User Database will saved the new list of saved locations for the User
9. User will press the “Fares” option
10. System will use Fare API to get the updated fares for bus/train and taxi
11. System will display a User-Friendly table of current fare prices
12. User will choose origin and destination
13. System uses goThere maps API to determine the routes for public transport and taxi
14. System will call Fare API to get the current fares for taxi and public transport
15. System will calculate the fares for each route using current fares
16. System will display routes and fares for each route

Use Case Template

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 1 | | |
| Use Case Name: | Login | | |
| Created By: | Yi Shen | Last Updated By: | Yi Shen |
| Date Created: | 26/1/2023 | Date Last Updated: | 26/1/2023 |

|  |  |
| --- | --- |
| Actor: | User, Registered User Database |
| Description: | Allows users to login to the application. |
| Preconditions: | User has an account |
| Postconditions: | User will be displayed with our Verdict |
| Priority: | 1 |
| Frequency of Use: | Always |
| Flow of Events: | 1. Users will key in their username and password. 2. Registered User Database will verify details 3. User will be logged in |
| Alternative Flows: | NIL |
| Exceptions: | EX1: If the user has keyed in the wrong username and password   1. Registered User Database will reject details 2. User will be prompted to try again   EX2: If the user has keyed in the wrong username and password three times   1. Registered User Database will reject details 2. User will be shown and error message |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

Use Case 2

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 2 | | |
| Use Case Name: | Sign up | | |
| Created By: | Yi Shen | Last Updated By: | Yi Shen |
| Date Created: | 26/1/2023 | Date Last Updated: | 26/1/2023 |

|  |  |
| --- | --- |
| Actor: | User, Registered User Database |
| Description: | Allows users to sign up in the application |
| Preconditions: | NIL |
| Postconditions: | 1. Registered User Database will be updated with the new User’s account 2. User will be logged in |
| Priority: | 1 |
| Frequency of Use: | Once per User |
| Flow of Events: | 1. Users will key in their Username, Password and Confirm Password 2. Registered User Database will update with Username and Password 3. User will be logged in |
| Alternative Flows: | NIL |
| Exceptions: | EX1:If User keys in an existing username   1. Registered User Database will reject Username 2. User will receive an error message “Username Taken!” 3. User will be prompted to try again   EX2: If Confirm Password is different from Password   1. User will receive error message “Please try again” |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

Use Case 3

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 3 | | |
| Use Case Name: | Manage Favourite Locations | | |
| Created By: | Yi Shen | Last Updated By: | Yi Shen |
| Date Created: | 26/1/2023 | Date Last Updated: | 26/1/2023 |

|  |  |
| --- | --- |
| Actor: | User, Registered User Database |
| Description: | Allows users favourite locations |
| Preconditions: | Users must be logged in |
| Postconditions: | Update list of favourited locations will be saved into Registered User Database |
| Priority: | 1 |
| Frequency of Use: | Seldom |
| Flow of Events: | 1. User will save or unsave a specific location 2. Registered User Database will saved the new list of saved locations for the User |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

Use Case 4

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 4 | | |
| Use Case Name: | Display Fares | | |
| Created By: | Yi Shen | Last Updated By: | Yi Shen |
| Date Created: | 26/1/2023 | Date Last Updated: | 26/1/2023 |

|  |  |
| --- | --- |
| Actor: | User, Fare API |
| Description: | Allows users to see the current fares for the various modes of transport |
| Preconditions: | Users must be logged in |
| Postconditions: | A table with various fares will be displayed to the User |
| Priority: | 1 |
| Frequency of Use: | Seldom |
| Flow of Events: | 1. User will press the “Fares” option 2. System will use Fare API to get the updated fares for bus/train and taxi 3. System will display a User-Friendly table of current fare prices |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |

Use Case 5

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 5 | | |
| Use Case Name: | Determine Route | | |
| Created By: | Yi Shen | Last Updated By: | Yi Shen |
| Date Created: | 26/1/2023 | Date Last Updated: | 26/1/2023 |

|  |  |
| --- | --- |
| Actor: | User,goThere maps API |
| Description: | Determine routes from origin to destination either by public transport or taxi |
| Preconditions: | 1. Users must be logged in. |
| Postconditions: | Routes for public transport and taxi will be determined |
| Priority: | 1 |
| Frequency of Use: | Always |
| Flow of Events: | 1. User will choose origin and destination 2. System uses goThere maps API to determine the routes for public transport and taxi |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | NIL |
| Special Requirements: | NIL |
| Assumptions: | 1. The locations are located in Singapore |
| Notes and Issues: | NIL |

Use Case 6

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: | 6 | | |
| Use Case Name: | Check Fares | | |
| Created By: | Yi Shen | Last Updated By: | Yi Shen |
| Date Created: | 26/1/2023 | Date Last Updated: | 26/1/2023 |

|  |  |
| --- | --- |
| Actor: | Fare API, goThere maps API |
| Description: | Determine routes from origin to destination either by public transport or taxi |
| Preconditions: | 1. Users must be logged in.  2. Routes for public transport and taxi will be determined |
| Postconditions: | Routes and fares of public transport and taxi will be displayed |
| Priority: | 1 |
| Frequency of Use: | Always |
| Flow of Events: | 1. System will call Fare API to get the current fares for taxi and public transport 2. System will calculate the fares for each route using current fares 3. System will display routes and fares for each route |
| Alternative Flows: | NIL |
| Exceptions: | NIL |
| Includes: | Determine Routes |
| Special Requirements: | NIL |
| Assumptions: | NIL |
| Notes and Issues: | NIL |