UBER-TAXI BOOKING SERVICE

HELPER DOCUMENT

This document contains:

Hardik Jain- 2020A7PS0102P

Sarvesh Gupta- 2020A7PS0093P

UBER-TAXI BOOKING SERVICE

A Brief Description: -

The following application facilitates a taxi booking service. A customer opens the application, chooses to enter the Customer Interface, following which he has to select the option to book a cab. Here, he is presented with a list of Landmarks, where he has to select his pickup location followed by the drop-off location. The Application presents the user with a list of available cabs. In the application, the city, drivers, and landmarks in the city have been initialized in the program. He selects his preferred cab. Then the driver logs into the driver interface and confirms to start the journey. The journey is then tracked and the application gives a real time feedback of ETA. After completion the user has the option to rate his experience.

Overview on the various classes and methods within them:

Class: Location

Variables:

1. int x_coordinate,
2. int y_coordinate,
3. int diff_x,
4. int diff_y

Methods:

1. Distance(int,int): Taking the x and y coordinates of the destination as parameters, this method is used to calculate the distance between the current location and the destination.

2. setNewLocation(int,int): Uses the parameters to set the coordinates of the new location.

<u>Class: Landmark</u>

<u>Variables:</u>

1. Int id
2. String Name:

<u>Methods:</u>

1. get_x
2. get_y: These both are the getter for landmark information
3. display: Displays the various details associated with the Landmark
4. ETA: Calculates the estimated time of arrival. Here we have assumed that it takes 1 min to change a single coordinate by 1 unit.

<u>Class: Cab</u>

<u>Variables:</u>

total_cabs

unique_req

<u>Methods:</u>

1. Change_state(int) : This changes the state of a cab with its id of the parameter when a customer

books it and updates the details in the array containing all the info about the cabs.

2. getState(int): This method is used to return the availability of the cab whose id we input as the parameter to the method.

Class: Booking

Methods:

1. available_cabs(): Displays the Ids of the available cabs at a particular instance of time and instantly stores and then displays and then clears the arraylist so as to avoid repeated additions.
2. set_pickup: Accepts an input for the Pickup Landmark id
3. set_drop: Accepts an input for the Dropoff Landmark id
4. choose_cab: The customer chooses the desired cab and then the state of the cab is changed(occupied)
5. startJourney-instantiates the start of the journey and calls the underlying methods

Class: Customer

Methods:

1. book_cab: This method calls the various methods of set_pickup,set_drop, choose_cab that are required to book a cab.

Class: Driver

Method:

1. confirm_booking: Prints the confirmation on booking the cab

2. face: Prints a message that the user in in the driver's interface

3.pathCalculator(int,int): calculate and displays the distance between the two landmarks.

4.pathCalculator(int): Calculates and shows the distance between the customer and the Driver.

Class: Uber

Methods:

Main: Main calls methods from various classes.

Future work possible along the same lines:

UBER-TAXI BOOKING SERVICE

In the given application, we have treated the city as a grid of 25 X 25. This implication makes the code a bit unrealistic. Incorporating the application with the Google Maps server to find nearest cabs, least ETA could further enhance the functionality.

Incorporating the factors like Time of the Day and Congestion could be incorporated for the Fare Calculation