Future work possible :-

1. Option to calculate the fare according to the time of booking cab.
2. The availability status of the driver is changed only in the code and not in the database whereas ideally it should change in both places.
3. Option to login any driver rather than only the one which is booked by the customer can be provided.
4. Option to check notification once driver is logged in. Also only the selected driver must get the notification and from there he can reject or accept the upcoming ride.
5. No option to change the driver’s available status in the database i.e. once the driver is marked unavailable in database there is no way to make the driver available again.
6. Option to add new driver.
7. Option to give rating for the selected driver.
8. Option to add new landmarks rather than the one mentioned in the database.
9. Option to share the cab.
10. Option to select the number of seats in the car.
11. Option to give feedback.
12. Can be linked with google maps to show the fastest route.
13. Fare hike in case of bad weather by linking with weather app.
14. Password can be made safer by ensuring certain conditions are met like it must contain a capital letter and some number.
15. Option to send OTP to confirm the authenticity of the user.
16. Option for various payment options like netbanking, paytm, GPay, cash etc. by linking with appropriate apps.
17. Display the drivers details like his mobile number, languages spoken by him etc. to the user once the ride has been confirmed by the driver.
18. Display continuously the location of driver rather than just the ETA.
19. After the journey is over the current location of the driver should be changed.

Journey:

Function countLineFast -> This method is used to count the number of lines in the given text file.

Variables :-

Scanner sc = Scanner sc is used to open city.txt and driverData.txt .

Scanner lg = Scanner lg is used to open customerLoginData.txt file and DriverLoginData.txt file .

Scanner in= To input from the console.

Files used :-

City.txt = Information about the dimensions of the city and various landmarks available in the city.

Driver.txt= Contains the name of drivers, their ratings, their current locations, availability status and speed for each driver respectively.

CustomerLoginData.txt = Contains name of the customer and their passwords.

DriverLoginData.txt= Name of the drivers and their passwords.

Working of code

Login:-

On running the code a choice is given to the user to either login or create a new account. If the user chose to create new account than it user is asked about username. The username entered should be unique i.e. it shouldn’t be in the database already. If it is in the database then it will keep on asking to enter any other username or enter ‘quit’ to terminate the program. After entering a unique username the user is asked for a password and after that both these details are stored in the database and the user is moved to the login page where he should reenter these details for login.

On login screen the user have to enter the correct username(i.e. it should be present in the database) and password to login successfully. If the username enters incorrect username then code will keep on asking for a valid username (or quit to terminate). After entering correct username there are 3 chances to enter the correct password. If not done in 3 chances the program will terminate.

List<landmarks> landmarks stores all the available landmarks from the the city.txt file.

After this there is a check if all the landmarks entered in the city.txt are within the coordinates of the city or not. If they are not then the program terminates with a message showing that landmarks can not be out of city , please check your input.

Customer journey details:-

After logging in successfully the customer is asked about the pickup location. The entered pickup location must be an available landmark, if not the program will keep on asking about the user to enter valid location. Similarly destination is asked and it should be valid as well as separate from the pickup location for obvious reasons.

Available Driver details :-

After entering the pickup location and destination a list of available drivers along with their ETA’s and ratings is displayed. Also the fare of the journey is displayed and the user has to enter the index of the driver that they want to chose for the drive. Note that the fare is same for all drivers since it depends on the distance between the pickup location and destination and not between the driver and pickup location. The user has to enter the index of the driver chosen by them and if entered incorrectly it will keep on asking to enter correct index. After this the control is shifted to the chosen driver.

Login for driver :-

The chosen driver is asked to enter their password. Again 3 tries are given to enter correct password. If not entered correctly in 3 chances a message showing that driver is unable to login is printed and the control is given back to the customer and the same list of drivers is shown again showing the available drivers and chose some other driver again.

If the password is entered correctly then the driver is logged in the app and the journey details are shown to him with the pickup location, destination and fare provided. The driver has option to either accept the ride or reject it. If the ride is rejected the control is again given back to customer and the message is shown about rejection of ride and the user is asked to chose another driver.

If the driver accepts the ride then continuous ETA is displayed on the screen according to condition that one second in reality means 30 seconds in the app. After the ETA is 0 a message is shown that the ride has reached and after 3 seconds(just for reference) a message is shown that the ride has ended.