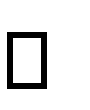
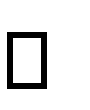
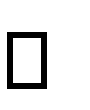
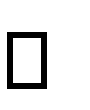
Case Study: MyCab – Cab Booking System

* 1. **INSTRUCTIONS**

 Create a directory by your name in drive. Under that create a subdirectory called **MiniProject** and Store your Project there.

Use C or C++. You can refer to your course material.



The code modules in the mini project should follow all the coding standards.

Number of participants for this project = 5 to 6

**PROBLEM STATEMENT**

* 1. **OBJECTIVE**

To create an online cab booking system like Uber where users can book a cab for their destination. The application is to be developed as Desktop Application. There are 2 entities User and Cab driver.

* 1. **ABSTRACT OF THE PROJECT**
     1. User and cab driver should be able to login into the application.
     2. User should be able to Book a trip.
     3. User should be able to details of the booking confirmation along with cab driver details after booking.
     4. Cab driver must be able to update his/her profile.
     5. Cab driver must be able to update car details.
     6. Cab driver must be able to check booked trips and mark completed trips.
  2. **FUNCTIONAL COMPONENTS OF THE PROJECT**

Following is a list of functionalities of the system. Wherever, the description of functionality is not adequate; you can make appropriate assumptions and proceed.

When MyCab starts it displays Following Screen -

Login Screen

* + 1. Register new user
    2. Register new cab driver
    3. Login as user
    4. Login as cab driver

1. Quit

System should maintain comma separated **users.txt** where each line stores username, password and all other information about registered user.

System should maintain comma separated **cabDrivers**.txt where each line stores username,

password and all other information about registered cab drivers.

Whenever “Login as user” is entered, system authenticates it with entry in “users.txt” file.

* + If match is found then “User Screen” is displayed.
  + If match is not found then message “Invalid User or password” is displayed and system exits.

Whenever “Login as cab driver” is entered, system authenticates it with entry in “cabDrivers .txt” file.

* + If match is found and “Cab Driver Screen” is displayed.
  + If match is not found then message “Invalid Cab Driver or password” is displayed and system exits.

1. When user login MyCab displays “User Screen”

User Screen

* 1. Book Trip
  2. Check Cab Driver Details
  3. Check Cab Details
  4. Print Booking Details

0. Quit

Enter your option : <option> option = 1 (Book Trip)

MyTrip asks source and destination of the trip, number of seats required

All availble car sizes and car models are displayed User selects car size and car model

All these details are stored in “BookTrips.txt” along with username and date along with the cab driver details/id

This is comma seperated file.

BookID, Booking Date, UserID, SRC, DEST, DriverID

if booking is successful user should be displayed with following data

1001| 23-11-22 | 222 | Delhi | Mumbai | 201 | XYZdriver | CABNo |

option = 2 (Check cab driver details)

With this option user can check cab driver details of the trip he/she booked. option = 3 (Check cab details)

With this option user can check cab details of the trip he/she booked.

option = 4 (Print Booking)

1001| 23-11-22 | 222 | Delhi | Mumbai | 201 | XYZdriver | CABNo |

1. When cab driver logins in, MyCab displays “CabDriver Screen”

CabDriver Screen

* 1. Update Profile
  2. Manage Car Details
  3. Check Booked Trips
  4. Mark Completed Trips

1. Quit

Enter your option : <option> option = 1 (Update Profile)

Details entered by cab drivers during registration can be modified using this option. “cabDrivers.txt” will be updated with new details.

option = 2 (Manage Car Details)

Cab driver will modify car details with this option. Details are stored in CabDetails.txt along with cab driver ID.

option = 4 (Check Booked Trips)

This option will display status of booked trip. If it is taken by user then driver is proceed to user location.

option = 5 (Mark Completed Trips)

Booked trip will be marked as completed. The trip will be removed from BookedTrip.txt

# Assumptions:

**Note:**

If you are using C

1. Use Linked Lists to read data from corresponding text files at the beginning of program.
2. Updates on data during program execution should be done in these Linked Lists.
3. When user quits the program all Linked Lists data to be writting in corresponding text files so that updated data is available in next program execution.

If you are using C++ Arrays of objects can be used instead of Linked List.

# Set Up Checklist for Mini Project Software Requirement:

# Minimum System / Hardware Requirements: