****

**Cab Booking Management System**



**CBMS**

**Software Requirements Specification Version 0.2**

**Document Control:**

| **Project Revision History** | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  |  |  |  |  |
| **Date** | **Version** | **Author** | **Brief Description of Changes** | | | | **Approver Signature** | |
| 11/11/2022 | 0.1(Draft) | Group 07 | Basic design and documentation | | | |  | |
| 15/11/2022 | 0.2(Draft) | Group07 | Use-Case Diagram | | | |  | |

| **Team Members** |
| --- |

| **Employee ID:** | **Name** |
| --- | --- |
| 46281993 | Ekta Sharma |
| 46279703 | Talla Yamini Sai Padmavathi |
| 46279711 | Polasu Teja Sri |
| 46281987 | Annayasha Paul |
| 46281995 | Rashmi Kumari |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |

**Table of Contents**

**1.Introduction……………………………………………………………………………………05**

* 1. Purpose
  2. Scope
  3. Definitions, Acronyms, and Abbreviations
  4. References
  5. Overview

**2. Overall Design…………………………………………………………06**

2.1 Product Perspective

2.2. Assumptions and Dependencies

2.3. Operating Environment

2.4. Product Features

**3. Working……………………………………………………………….07**

3.1 Use case Diagram

**4. System Features………………………………………………............08**

4.1Functional Requirements

**5. External Interface Requirements……………………………...........10**

5.1User Interfaces

5.2 Hardware Interfaces

**6. Non-Functional Requirements………………………………............11**

6.1. Performance Requirements

6.2. Software Quality Attributes

6.3. Safety Requirements

**7. Operational Scenario………………………………………………..12**

**8. Demo…………………………………………………………………13**

**Cab Booking Management System**

1. **Introduction**
   1. **Purpose**

The purpose of the project and this documentation is to build a Cab Booking Management System application using C and its various supporting tools to book cab, calculate fare, fetch details of drivers and many more. It is inspired by the different apps present in the market such as Ola and Uber.

* 1. **Intended Audience**

This project is designed to provide users with a platform to login in the portal, book cabs, calculate fare, fetch details. This is designed according to user’s criteria and requirements.

* 1. **Project Scope**

The purpose of the application is to create a system which can be used to login into the portal according to their criteria and if the user is a cab driver he/she can update the car details and if he/she is a customer then he/she can book a cab, fetch car and drivers details and also make payments. The end goal is to make traveling of users smooth and the application easy to us.

* 1. **Reference**

<https://www.codespeedy.com/find-a-specific-file-in-a-directory-in-cpp/>

<https://www.javatpoint.com/file-handling-in-c>  for file handling concept

<https://www.javatpoint.com/linux-commands> Linux Command

**2. Overall Description**

**2.1 Product Perspective**

The product described in the SRS document is an automated and centralized alternative to traditional cab booking system. The product will replace the call and book features in traditional system with an online booking feature.

**2.2. Assumptions and Dependencies**

The following assumptions have been made in regards to the development of the Cab Booking System:

* Username is valid email addresses of respective user.
* C source code can be compiled on the machines.
* Administrator has the authority to add/delete employee accounts
* Administrator has the authority to delete client.

**2.3** **Operating Environment**

Operating Environment for the Cab Booking System is as follows:

* Operating System: Any UNIX Based OS
* Compiler: GCC or similar to compile source code written in C programming language.

**2.4. Product Features:**

The features are as follows:

* View Available Vehicles: The client must able to see all details about the available vehicles without any constraints.
* Calculate Fare: The client must be available to check the fare they should pay for the vehicles.
* Record maintenance: The system also must keep track the statistical reports of daily activities of the online booking.
* Discount Offer: The admin can create discount codes and the client can get discount on fares using the cod

**3.Working of Cab Booking System**

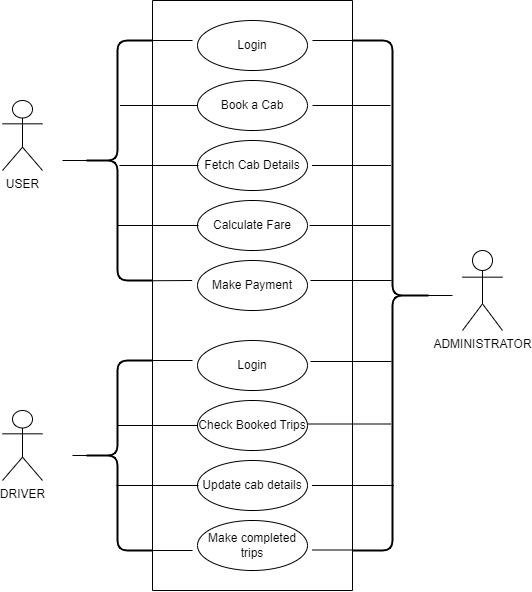
The Cab Booking System application is developed using the C programming language. This Cab Booking System application allows the user to view available cabs, register the cabsview profile and book cabs.

File handling and data structure concepts have been used for almost all functions in this application. This application is solved using several methods, like one can solve this program using user defined function concept, loop condition and conditional statements. The following steps are followed while implementing the given program :

* The input is entered i.e, the value of choice ( the menu number) selects the particular menu.
* Next it goes to a particular menu and then goes to the particular function.
* It prints the resultant value which is the output of the execution.
* The inputs written can also be stored in a common file format for storage purposes.
* The stored data can be retrieved for future modification or display purposes.

With the above mentioned steps we can insert, delete, retrieve or update a record in the Cab Booking Application.

**3.1** **Use case Diagram**



**Fig. 1.1**

**4.System Features**

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

**4.1**. **Functional Requirements**

Following is a list of functionalities of the system:

**1. Login Screen:**

1. Register new user
2. Register new cab driver
3. Login as user
4. Login as cab driver
5. Quit

For Login as User and Login as Cab Driver screen is entered, system authenticates it with entry in “user.txt” and “cabDrivers.txt” file.

If match is found then the screen (user or cab driver) is displayed.

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

**2. Login as User Screen:**

1. Schedule 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 “ScheduledTrips.txt” along with username and date.

1. Book Trip

All scheduled trips of user whose fare is provided by cab driver are displayed. User selects trip to book from the list. All details are stored in “BookedTrips.txt” along with username and date. Corresponding entry of trip is deleted from “ScheduledTrips.txt”.

1. Check Cab Driver Details

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

1. Check Cab Details

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

1. Check Bill

With this option user can check bill of the trip he/she booked. Bill Amount = Fare entered by cab driver \* number of seats required.

1. Make Payment

Bill is displayed to user for completed trip. Payment details such as mode of payment, credit card number, bill amount, status of payment will be stored in “payments.txt” file.

1. Quit: Quit from the screen.

**3. Login as Cab Driver Screen:**

1. Update Profile

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

1. Manage Car Details

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

1. Check Scheduled Trips

Display all scheduled trips whose date = today and time >= current time. Cab driver will select the trip to book and will provide fare. Cab drivers ID, Car ID and fare details will be updated in “BookedTrips.txt”.

1. Check Booked Trips

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

1. Mark Completed Trips

Booked trip will be marked as completed. The trip will be removed from BookedTrip.txt and will be added to CompletedTrip.txt along with user ID, Cab driver ID, Payment ID, amount etc.

**5.External Internal Requirements**

This application uses UNIX (CLI) to perform interaction between client and user.

Based on User choice the functions will perform tasks (Code reuse).

**5.1.** **User Interfaces**

* + GUL The application does not use Graphical User Interface
  + CLI: This application uses Command Line Interface to accept console commands by users choice and perform the needful functions.
  + MDI: This application will provide list or menu for user choice (Menu driven interface)

**5.2. Hardware Interfaces**

Hardware Requirements are as follows:

* 32 bit and 64bit Machine capable of running UNIX based operating system
* Storage space to save contact details filled by users (In the form of text file

**6. Non-Functional Requirements**

**6.1 Performance Requirements**

The Application is developed to run through CLI on UNIX based systems. As long as the machine can run the operating system along with the necessary dependencies without any flaws there are no additional requirements.

* 1. **Software Quality Attributes**

**Serviceability:**

* The system requires minimal amount of maintenance. The maintenance and upkeep can be performed by any person with a basic understanding and development experience in C.

**Reusability:**

* The application needs to be designed in a way such that the code can be easily reused and it can run in any UNIX like machine in the organization.

**Binary Compatibility**:

* This application should be compatible with any computer that has an UNIX based operating system.

**Portability:**

* The source code needs to be implemented in such a way that it is portable to any machine that can compile and run C programs.
  1. **Safety Attributes**
* Irrelevant contacts should not be displayed during searches to protect privacy.
* The implementation should not be susceptible to easy modification.

**7.Operational Scenarios**

1)Scenario 1: Login Portal

* User and cab driver should be able to login into the application.

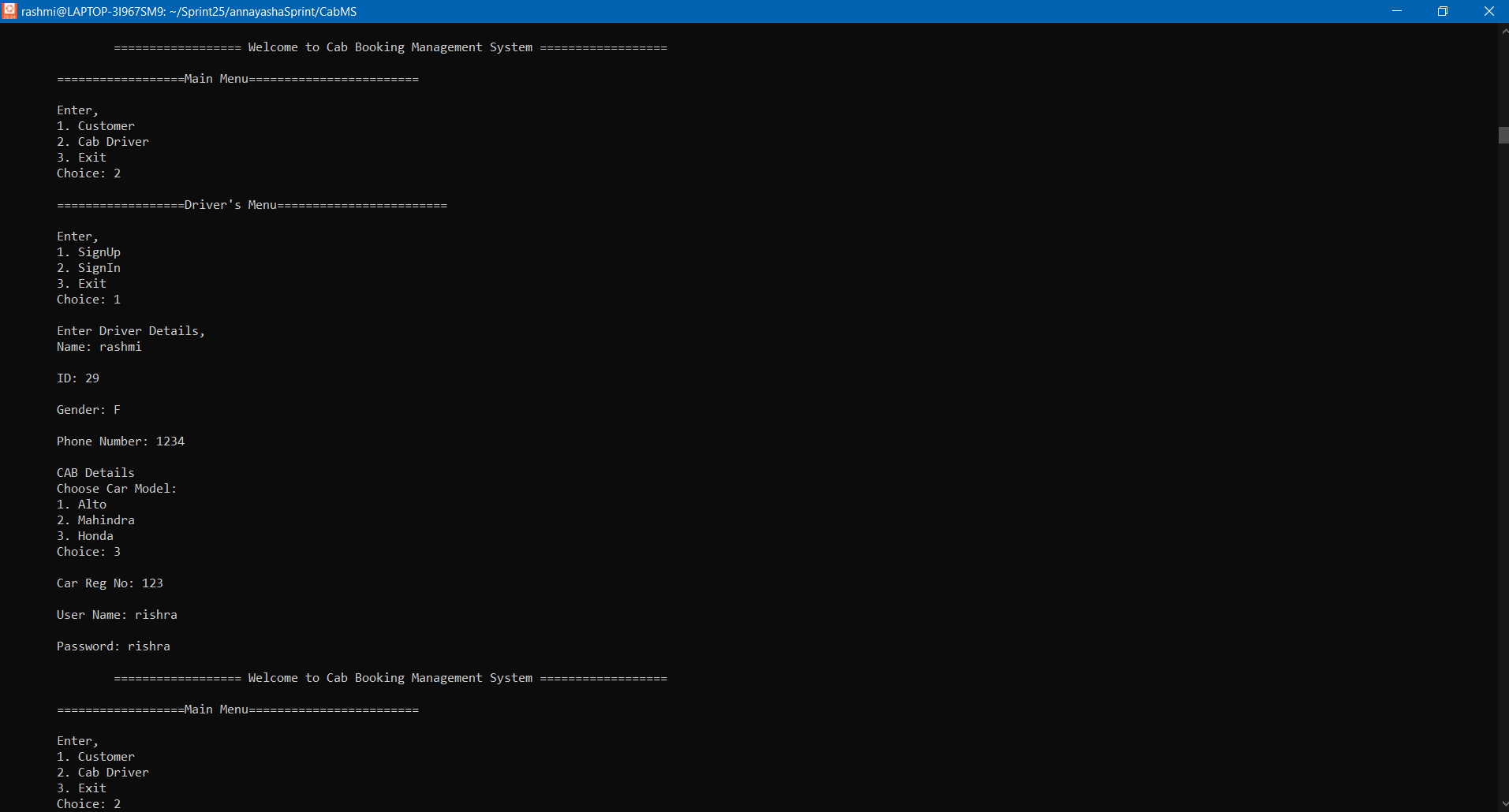
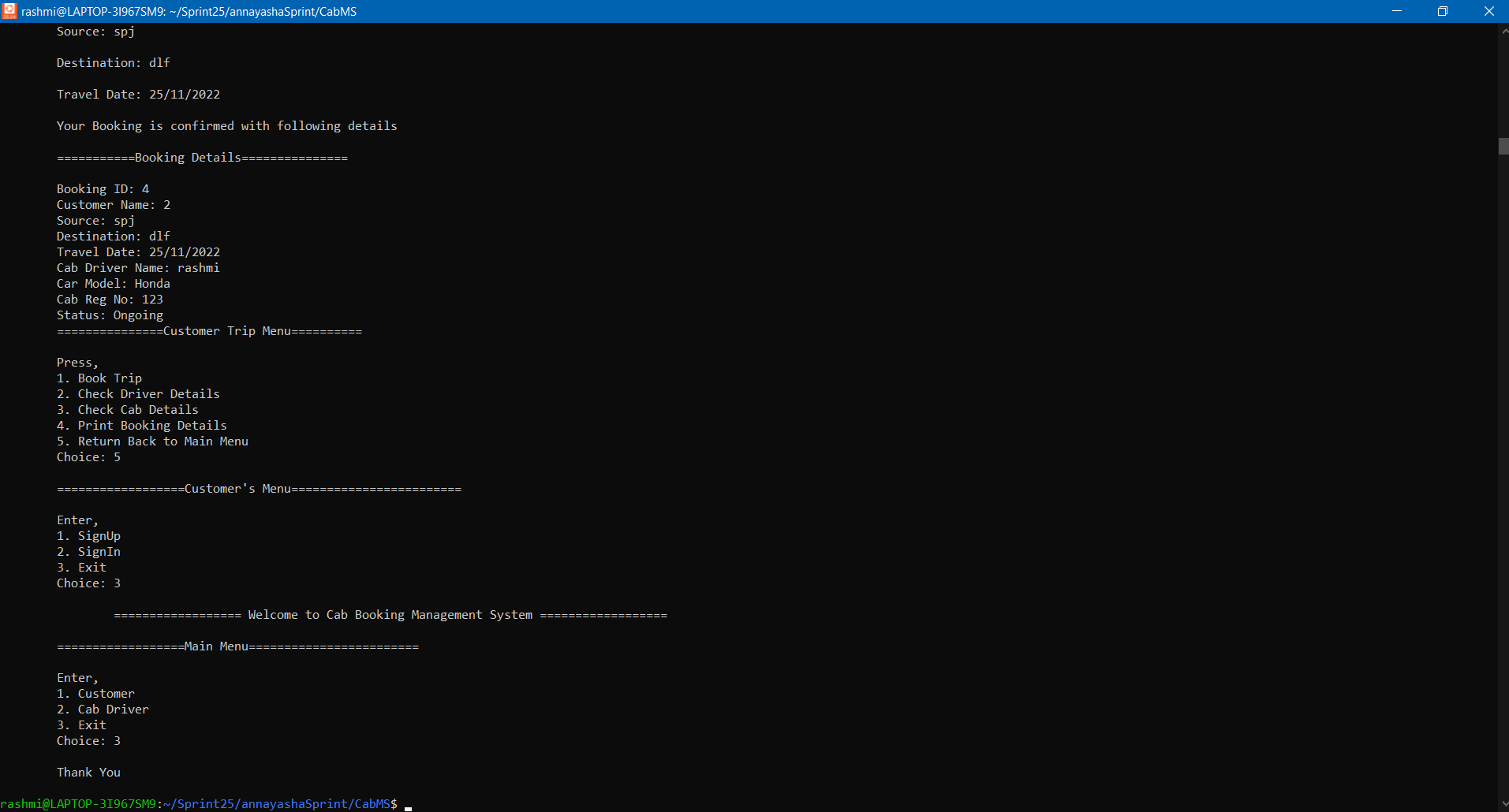
2)Scenario 2: If the User is a customer

* A >User should be able to schedule a trip.
* B >User should be able to select a trip and book a trip.
* C >User should be able to fetch cab & cab driver details after booking.
* D > User should be able to check bill details and make a payment.

3)Scenario 3: If the User is a cab driver

* A >Cab driver must be able to update his/her profile.
* B >Cab driver must be able to update car details.
* C >Cab driver must be able to check scheduled trips.
* D >Cab driver must be able to check booked trips and mark completed trips.

**8.Demo**



THANK YOU