CBMS
0.2v

# Cab Booking Management System

The purpose of this document is to provide with a template for documenting CBMS.

**Document Control :**

# Project Revision History

| Date | Version | Author | Brief Description of Changes | Approver Signature |
|---|---|---|---|---|
| 15/11/2022 | 0.1v | Group 07 | Initial draft | |
| 19/11/2022 | 0.2v | Group 07 | Added ER diagram | |
| | | | | |

# 1. Introduction

This project is intended to implement Cab booking management system. Cab Booking System specializing in Hiring cabs to customers. It is an online system through which customers can view available cabs; register the cabs, view profile and book cabs. Cab booking service is a major transport service provided by the various transport operators in a particular city. Mostly peoples use cab service for their daily transportations need. The company must be a registered and fulfills all the requirements.

## 1.1. Intended Audience

This document could be shared or view across the following members the CG employees, BU SME's, internal SME's

| | |
|---|---|
| CG Employee | |
| BU | |

## 1.2. Acronyms/Abbreviations

| | |
|---|---|
| CBMS | Cab booking management system |
| | |
| | |
| | |

## 1.3. Project Purpose

The purpose of this high level design document is to add the necessary details to the CAB booking management system application using C and its various supporting tools to book CAB. It helps us to understand automatic streamlining of CABs for various routes generated by admin.

## 1.4. Key Project Objectives

Cab Booking Management System

## 1.5. Project Scope and Limitation

The scope of this project is to create application which will work on wide area. This will be an solution to people for their safe and easy journey to one place to another place. The user can book car as per their location and according to their choice. In this administrator he can add, remove the cars as well as manage

Driver's rides. It will save customers time and efforts to search vehicle on unknown place, if they travel from one place to another place.

- **In Scope**

The purpose of this application is to create system which can be used to login in to the portal according to their criteria if the user is a CAB driver he/she can update their details and they can get the details about the customer and if he/she is a customer they can get details of CAB driver and managing payments.

- **Out of scope**

It is not a real time project but it is a simulation.

## 1.6. Functional Overview

The scope of this project is to create application which will work on wide area. This will be an solution to people for their safe and easy journey to one place to another place. The user can book car as per their location and according to their choice. In this administrator he can add, remove the cars as well as manage Driver's rides. It will save customers time and efforts to search vehicle on unknown place, if they travel from one place to another place.

CAB management system includes

- 24/7 services are available.
- Driver and customer details.
- Dropping and picking up data.
- Booking system.
- Making payments.
- Tracking location data.
- Vehicle data.

## 1.7. Assumptions, Dependencies & Constraints

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.

## 1.8. Risks

NA. There are no risks for Cab booking management system.

# 2. Design Overview

The main design features include four major parts: the architecture, the user interface design, the files, process relation, and automation. In order to make these designs easier to understand, the design has been illustrated in attached diagrams ( Use Case, Data flow diagrams).

## 2.1. Design Objectives

The purpose of this application is to create system which can be used to login in to the portal according to their criteria if the user is a CAB driver he/she can update their details and they can get the details about the customer and if he/she is a customer they can get details of CAB driver and managing payments.

- **Recommended Architecture**

## 2.2. Architectural Strategies

- **Design Alternative**

The project uses a database to establish a connection between the user and server for cab management system.

- **Reuse of Existing Common Services/Utilities**

The project does not reuse any new common services or utilities.

- **Creation of New Common Services/Utilities**

The project does not create or use any new common services or utilities.

- **User Interface Paradigms**
  - Desktop or a Linux machine.
  - Command Line Interface (CLI).

- **System Interface Paradigms**
  - Operating system – Unix.
  - Linux Kernel version - 4.4.0-19041-Microsoft.
  - Bash shell: x86_64 GNU/Linux

- **Error Detection / Exceptional Handling**

  Should errors be encountered, an explanation will be displayed as to what went wrong. An error will be defined as anything that falls outside the normal and intended usage.

- **Memory Management**

NA

- **Performance**

Performance is going to be very important for this project. For everything to run smoothly for this project, The system will work on the customer's terminal and the performance depends upon the hardware component of the customer's system

- **Security**

The customer's terminal window shall never display a customer's password. It shall always be echoed with special characters representing typed characters. The system's back-end files shall never display a customer's password. The customer's password may be reset but never shown.

- **Concurrency and Synchronization**

NA

- **Housekeeping and Maintenance**

Very little maintenance should be required for this setup. An initial configuration will be the only system required interaction after system is put together. The only other user maintenance would be any changes to settings after setup, and any specified special cases where user settings or history need to be changed. Physical maintenance on the system's parts may be required, and would result in temporary loss of data or Internet. Upgrades of hardware and software should have little effect on this project, but may result in downtime.

# 3. System Architecture

System architecture is the conceptual model that defines the Structure, behavior, and more views of a system.

## 3.1. System Architecture Diagram. (Not Necessary)

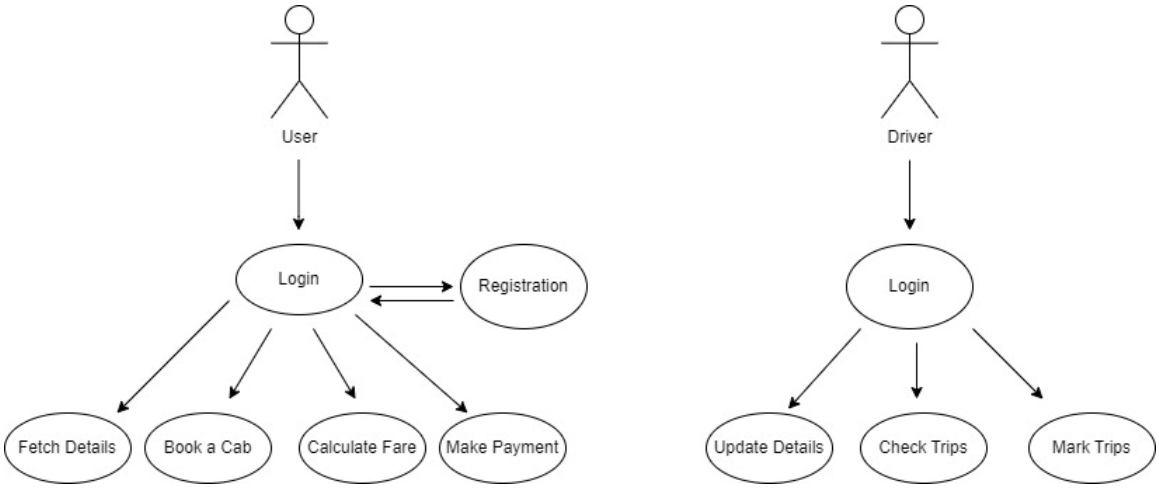


**Fig. 1.1**

## 3.2 System Use-Cases

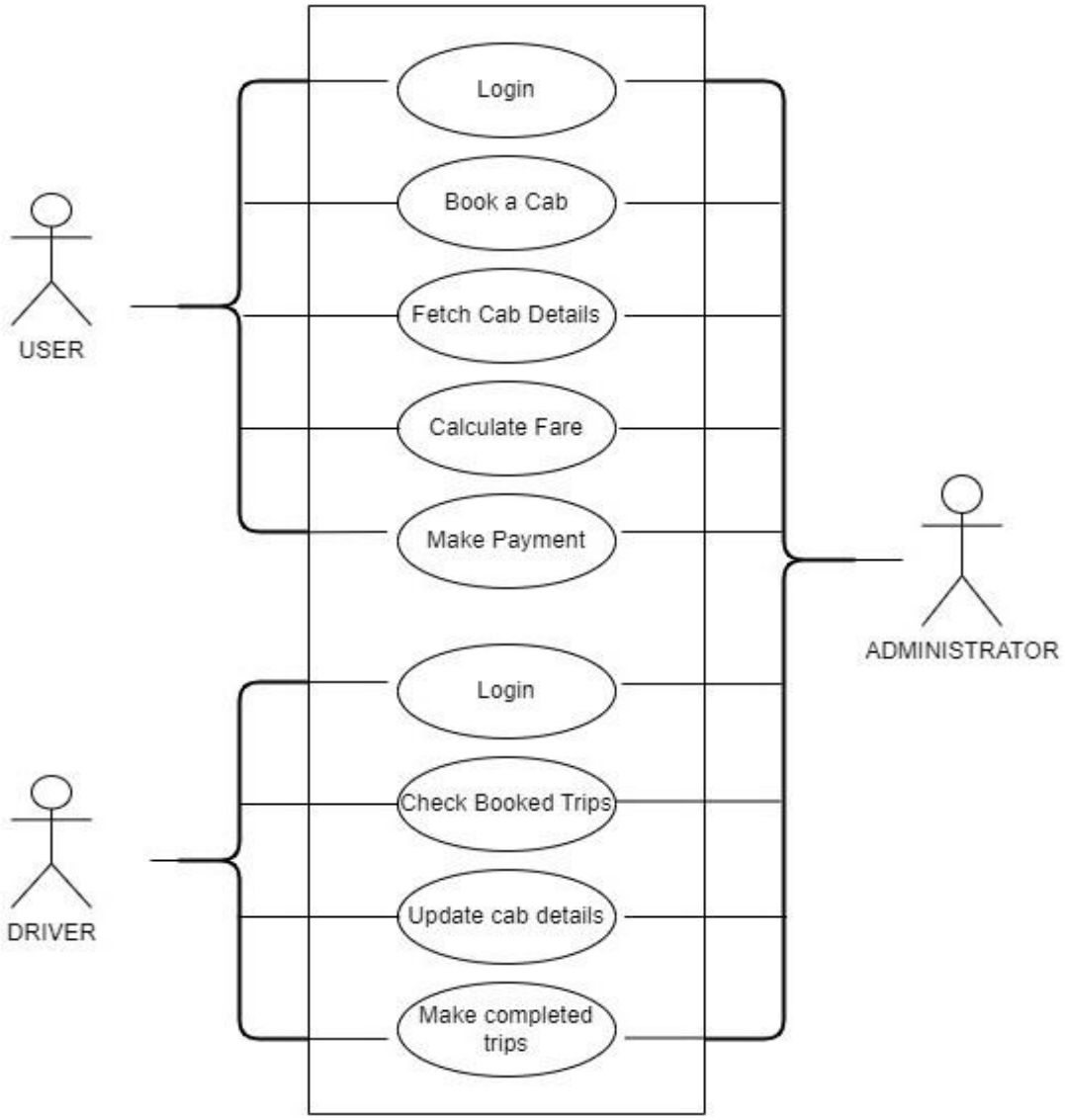


**Fig. 1.2**

## 3.3 Subsystem Architecture

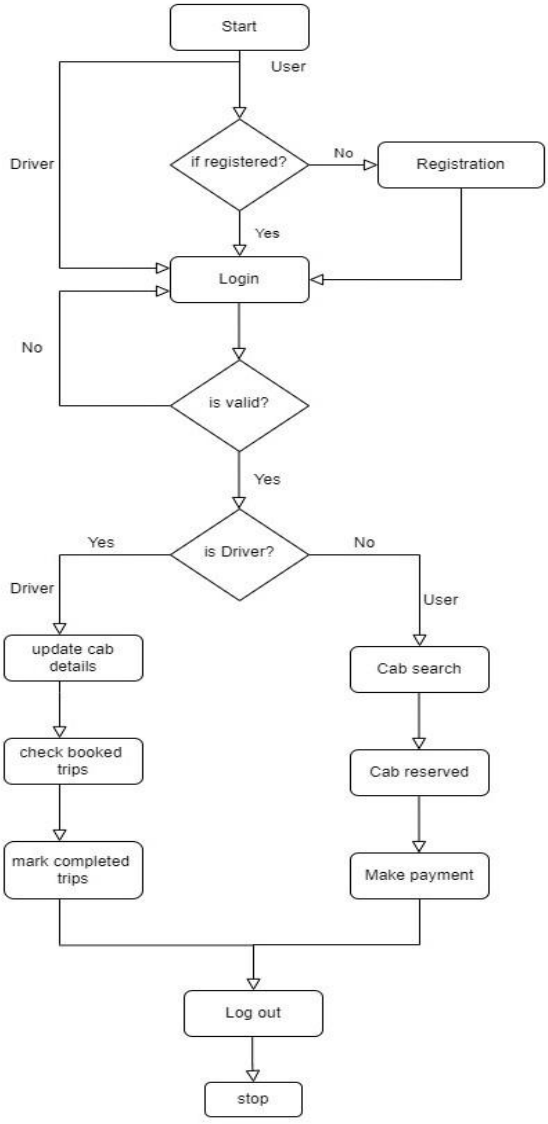


**Fig. 1.3**

## 3.4 System Interfaces

### 3.4.1 Internal Interfaces

The internal interfaces comprise interfaces through which the system interacts with the user through which it provides them services.

- Cloud System

- Internet

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).

### 3.4.2 External Interfaces

The external interface comprises interfaces through which the users interact with the system.

- Desktop or Linux Machine
- Internet

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).

## 3.5 Sequence Diagram

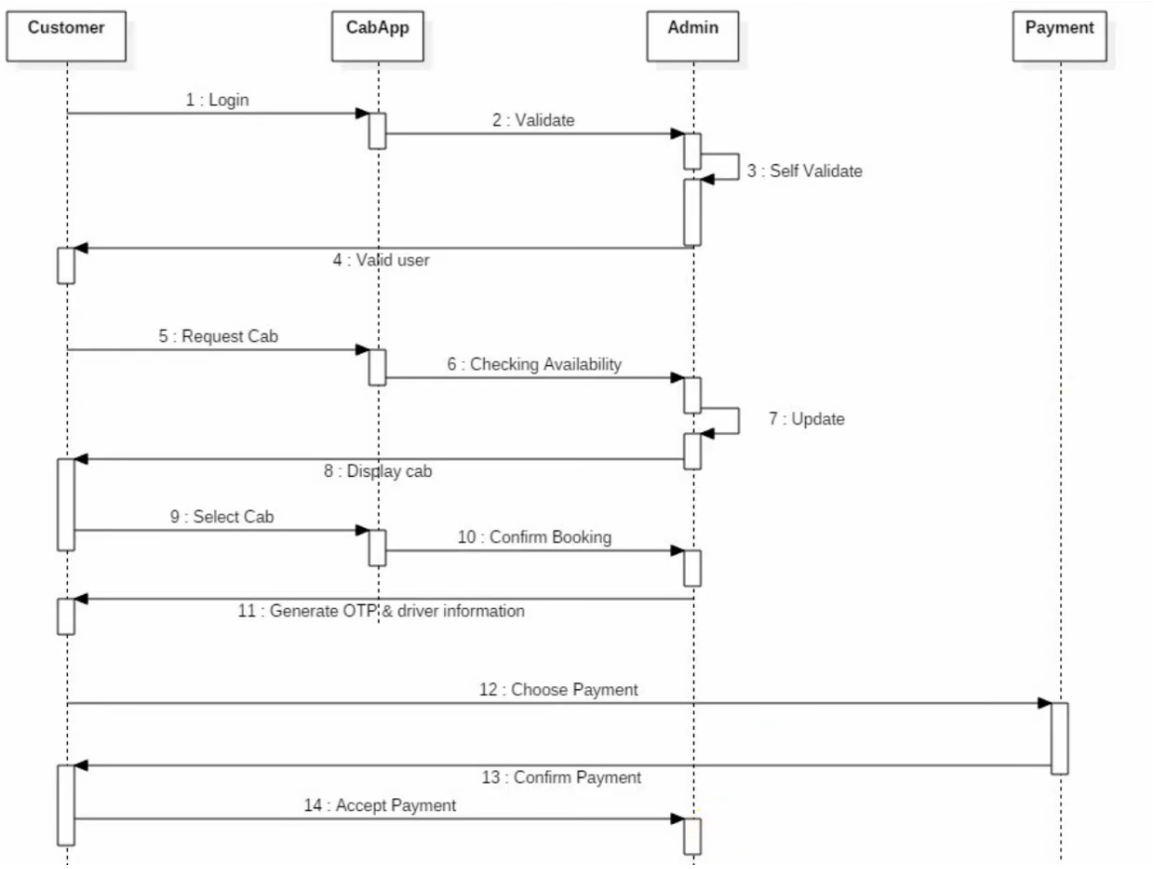


**Fig. 1.4**

| Actors | Description |
|---|---|
| **Administrator** | • When MyCab starts it displays Login Screen.<br>• Admin shows Register new user and Register new cab driver<br>• After that can choose Login as user and Login as cab driver<br>• Or can choose Quit to exit. |
| **User** | • Schedule Trip- MyTrip asks source and destination of the trip, number of seats required. All available car sizes and car models are displayed User selects car size and car model.<br>• Book Trip- All scheduled trips of user whose fare is provided by cab driver are displayed. User selects trip to book from the list.<br>• Cab driver details- With this option user can check cab driver details of the trip he/she booked.<br>• Cab details- With this option user can check cab details of the trip he/she booked.<br>• 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.<br>• 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. |
| **Cab Driver** | • cab driver ID.Update Profile- Details entered by cab drivers during registration can be modified using this option.<br>• Car Details- Cab driver will modify car details with this option. Details are stored in CabDetails.txt along with<br>• 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.<br>• Booked Trips- This option will display status of booked trip.<br>• Completed Trips- Booked trip will be marked as completed. The trip will be removed and will be added to along with user ID, Cab driver ID, Payment ID, |

# 4. Detailed System Design

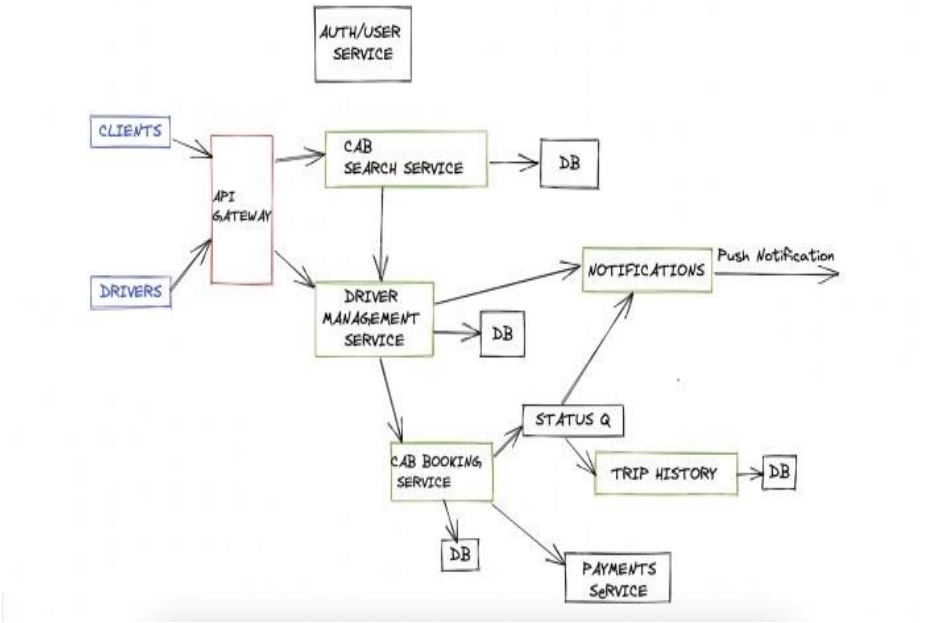


**Fig. 1.5**

## 4.1. Key Entities

- Admin
- User
- Cab Driver

## 4.2. Detailed-Level Database Design

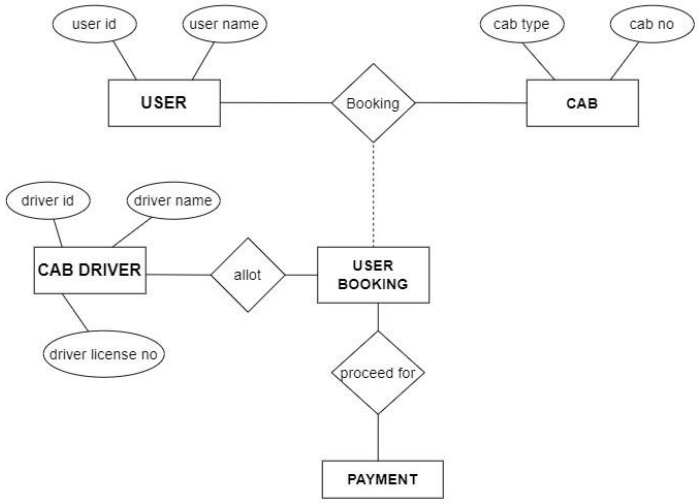


**Fig. 1.6**

- **Data Mapping Information**

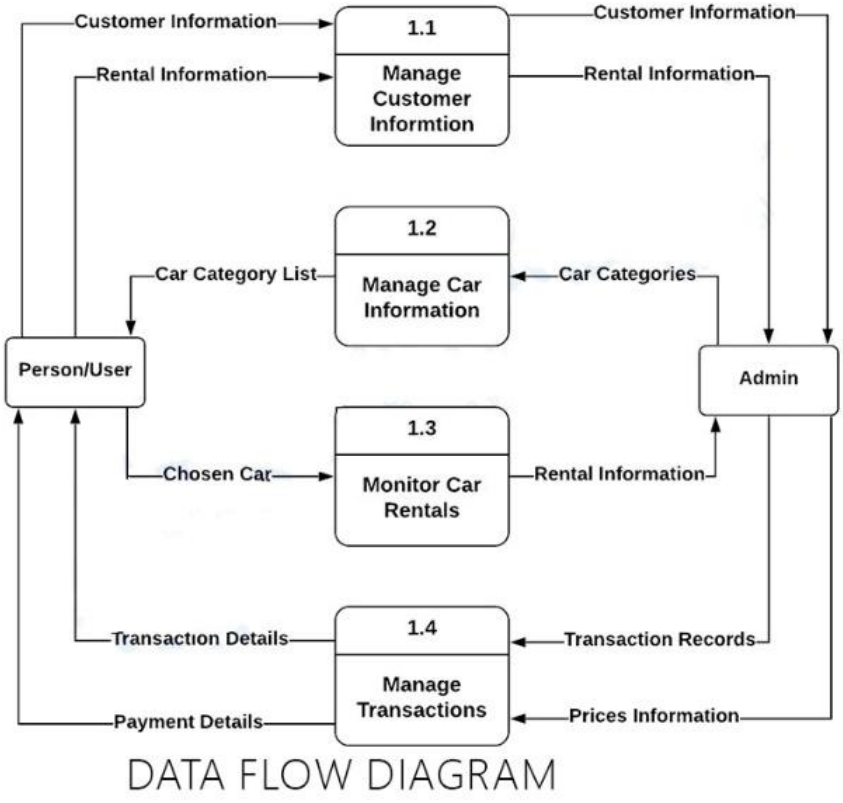


**Fig. 1.7**

- **Data Conversion**

N.A

## 4.3. Archival and retention requirements

NA

## 4.4. Disaster and Failure Recovery

NA

## 4.5. Business Process workflow

NA

## 4.6. Business Process Modeling and Management (as applicable)

NA

## 4.7. Business Logic

NA

## 4.8. Variables

UserName, UserPassword used for entering details of user.

## 4.9. Activity / Class Diagrams (as applicable)

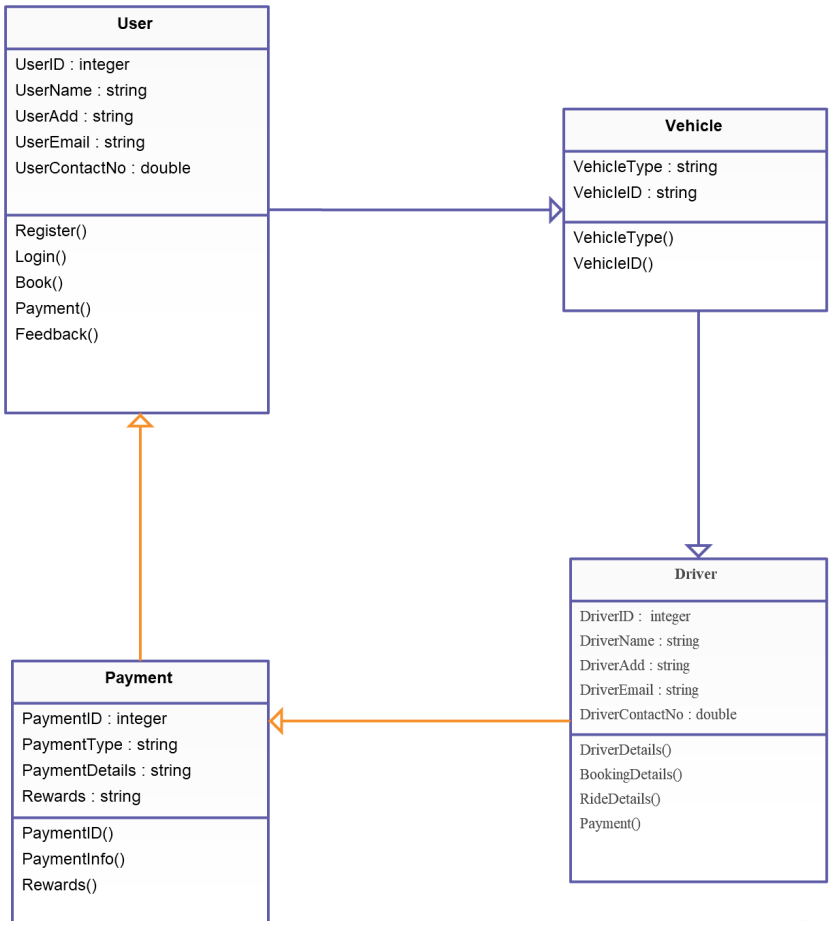


**Fig 1.8**

## 4.10. Data Migration

NA

- **Architectural Representation**

NA

- **Architectural Goals and Constraints**

NA

- **Logical View**

NA

- **Architecturally Significant Design Packages**

NA

- **Data model**

NA

- **Deployment View**

NA

# 5. Environment Description

## 5.1. Time Zone Support

It will support time zone as per Indian standard time (IST) in (GMT +5:30) and UST standard.

## 5.2. Language Support

C language and compilation using make file and gcc. The Linux commands to do that task we can specify the commands.

## 5.3. User Desktop Requirements

User desktop requires a Linux environment, Operating system of Linux Debian or Terminal x86_64 GNU/Linux and kernel version 4.4.0-19041-Microsoft #1237Microsoft and reliable

## 5.4. Server-Side Requirements

In server side,

- Disk space – Minimum 150GB
- Uninterrupted connectivity 24x7
- Monitor long running jobs, to reduce the server load

- **Deployment Considerations**

Deployment considerations are,

- 500Mhz Processor
- 120GB HDD CPU
- minimum 4GB RAM
- Network connectivity

- **Application Server Disk Space**

  Disk space – Minimum 150GB

- **Database Server Disk Space**

  NA

- **Integration Requirements**

  The PWD Displays the current working directory on the server for the logged in user

- **Jobs**

  NA

- **Network**

  NA

- **Others**

  NA

## 5.5. Configuration

NA

- **Operating System**
  - Operating system –Linux.
  - RAM - 4GB or more.
  - Processor - i3/i5.

- **Database**

  NA

- **Network**

  NA

- **Desktop**

  Minimum Windows 10, 4 gb Ram with i3 configuration is required.

## 6. References


1. Implementation of Appointment Management System for the Innovation Laboratory.
2. System Requirements Specification Document
3. Design and Implementation of a Cab Booking Management System, Department of Computer Science
4. Project Proposal Document.
5. System Specification Requirement.
6. High Level Design Document for CBMS


## 7. Appendix

- **https://www.geeksforgeeks.org/system-design-of-uber-app-uber-system-architecture/**

## Change Log

# CBMS Template Version Control (Maintained by QA)

| Date | Version | Author | Description |
|---|---|---|---|
| 15/11/2022 | 0.1v | Group07 | Initial Version |
| 19/11/2022 | 0.2v | Group 07 | Added ER Diagram |
| | | | |
| | | | |