Driver App

*Undertaken By:*

**ALI SHAN**

Reg. No. CIIT/FA18-bSE-007/Vhr

**HASSAN SHAKEEL**

Reg. No. CIIT/FA18-BSE-061/Vhr

*Supervised By:*

**MR. NASHIT ALI**



A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELORS IN COMPUTER SCIENCE / SOFTWARE ENGINEERING

DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVESITY ISLAMABAD, VEHARI CAMPUS

VEHARI – PAKISTAN

SESSION 2018-2022

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of BS (SE) “Project title” was developed by

**ALI SHAN (CIIT/FA18-BS/SE/ 007)** and

**HASSAN SHAKEEL (CIIT/FA18-BS/SE -061)**

under the supervision of “MR. NASHIT ALI” and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Sciences.

---------------------------------------

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Head of Department**

**(Department of Computer Science)**

**DEDICATION**

To my loving parents and teachers who always encouraged me to be the soul of my periphery and taught me to work hard for the things that I aspire to achieve. My family members and sincere friends also encouraged me during the challenges of graduate school and life. I truly grateful for having such type of family members and sincere friends in my life.

**ACKNOWLEDGEMENT**

All praise to ALLAH ALMIGHTY alone, the Omnipotent, the most compassionate. His prophet “HAZRAT MUHAMMAD” (peace be upon him), the most perfect and exalted among and of ever born on the surface of the earth, who is forever torch of guideless and knowledge for human.

We feel great pleasure in expressing our heartiest gratitude to our honorable HOD **Dr. Malik Muhammad Ali Shahid** and Supervisor **Mr. Nashit Ali** (Lecturer of Compute Science Department) COMSATS University Islamabad, Vehari Campus for kind behavior, valuable suggestions, worth and keen supervision, scholarly criticisms, sympathetic attitude towards completion of this dissertation report. His kind behavior and attitude during period of project work is unforgettable. We express our sincere thanks to all respectable teachers and faculty members of computer science department of COMSATS University Islamabad, Vehari Campus.

We feel great happiness in expressing our thanks to our family for their love and support. Prayers of our family are a treasure for our life.

This dissertation report is a proof of sincerity of those who helped us during this project work. So, we submit our earnest thanks again to all of them for their encouragement and moral support. Finally, we pray for health happiness and prosperity of all the participants.

**Project Team Members**

Ali Shan

Hassan Shakeel

**PROJECT BRIEF**

PROJECT NAME DRIVER APP

ORGANIZATION NAME PRIVATE/SELF BASED

OBJECTIVE WORLD-WIDE USERS OF AN APP

UNDERTAKEN BY ALI SHAN AND HASSAN SHAKEEL

SUPERVISED BY MR. NASHID ALI

DESIGNATION

CS DEPARMENT

COMSATS UNIVESITY ISLAMABAD, VEHARI CAMPUS

Started On 15/12/2021

Completed On 15/1/2022

COMPUTER USED CORE I5 AND CORE I7

SOURCE LANGUAGE DART \*/

OPERATING SYSTEM WINDOWS

TOOLS USED ANDROID STUDIO, FIREBASE

**ABSTRACT**

This app lets users with vehicles help other users who have the closest current location and destination by giving them a lift. A user who has a vehicle and is about to start a journey first marks its status as “I can share a ride” and then enters its destination in the app before starting the journey. A person who wants to travel but is not having conveyance requests the app for a lift by providing his/her destination. The app calculates three things:

1. Requesting user’s current distance from the current distances of all the users who have marked their status as “I can share a ride”.
2. The distance of destination of the requesting user from the destinations of all the users who have marked their status as “I can share a ride”.
3. Finally, the app adds up the current location distance and destination distance for each user

whose status is “I can share a ride” from the requesting user.

The user with the smallest sum is the selected user. A message appears on the selected user’s smartphone that someone wants a ride. The location of the requesting user is also shared with the selected user on the map so that the selected user may pick the requesting user from its current location. After the journey ends, a user who has set its status to “I can share a ride” will mark its status as “I cannot share a ride” so that the app knows that he/she is no longer offering a ride.

# Contents

Contents vi

1Introduction 1

1.1 System Introduction 1

1.2 Background of the System 1

1.3 Objectives of the System 2

1.4 Significance of the System 2

2Overall Description 3

2.1 Product Perspective 3

2.2 Product Scope 3

2.3 Product Functionality 3

2.4 Users and Characteristics 4

2.5 Operating Environment 4

3Specific Requirements 5

3.1 Functional Requirements 5

3.2 Behaviour Requirements 5

3.3 External Interface Requirements 10

4Other Non-functional Requirements 15

4.1 Performance Requirements 15

4.2 Safety and Security Requirements 15

4.3 Software Quality Attributes 15

5Design Description 16

5.1 Composite Viewpoint 16

5.2 Logical Viewpoint 17

5.3 Information Viewpoint 18

5.4 Interaction Viewpoint 19

# 1. Introduction

## There are many frameworks available to develop an Android mobile application. We use to code in Dart language by using Flutter and Android studio. Driver App is intended to design to overcome the problem of finding the cab for his all work in an effective manner. This project describes the development of an online Driver app for Users. The proposed system intends to ease the selection of driver by providing all information in one app where a user can easily search for a driver and can communicate directly. User can also see his history, select favorite driver, select rideshare option, select schedule booking, and bargain his fare with drives and also view notifications. This is making information retrieval and management faster, easier, and more efficient as compared to the current system. Android based Driver app is an important mean of achieving. In comparison to the old system, you want to visit a physically and bargaining with drivers and consume time. At the present situation, User use a manual system to look for Drivers without any knowledge or information about the Drivers, besides that, they also do not know is any drivers available for ride or not before visiting the Stand. With Driver app, user can check with system the availability of drivers without meeting them physically. Driver can also accept or reject them on basis of availability. Driver App is developed using Flutter.

## System Introduction

Driver App will be an android application which will works like Careem, Uber and Dida (App in China). This application will provide services to passengers and drivers. Driver App connects passengers and drivers using GPS (Google Map API (Application Program Interface)) system. Everyone who has any type of vehicle can register on this application. This app will support all type of vehicles. This application enables passengers to order any kind of vehicle with their Smartphone. The main objective of this application is that drivers online register their vehicles by providing their essential information and then check their nearest pick and drop.

The project has two parts, the first one is an application run on android device and the second one is server on Firebase database. Android section will have two applications. One is master refers to driver and other one is client refers to passengers. If the driver accepts booking, then the server will send order confirmation to the passenger application. This application is managing passengers’ booking in quick and easy way with shortest time possible. With one click on the button you can order a vehicle if you are a passenger. With another click you accept or decline requests if you are a driver. No party to be added to control this work. Some types of vehicles still don’t have any online ride service. But with this application, any type of vehicle can register. This app will support all type of vehicles. This app will run in all cities. All vehicle drivers can register and earn money using this app without any limitation. This service will not have any central office.

## Background of the System

Our project is related to Mobile Application in which we develop an application using Flutter and Android Studio with the help of Dart language. Many similar systems exist before this project. The work was done manually. We add the fare bargaining and also rideshare new features and due to ridesharing the fare/rent of users must divide.

## Objectives of the System

To ensure durable drivers, customer loyalty, and recognition/satisfaction. Maintain its recognition throughout the world. Rise the number of drivers to reduce the estimated waiting time. Avoid being replaced by self-driving cars.

## Significance of the System

## User use a manual system to look for Drivers without any knowledge or information about the Drivers, besides that, they also do not know is any drivers available for ride or not before visiting the Stand. With Driver app, user can check with system the availability of drivers without meeting them physically. Driver can also accept or reject them on basis of availability. This app is basically open-source online ride service so this app provides opportunity to drivers to register their Vehicles. With this app, drivers can register any type of vehicle and then start finding the nearest pick and drop. This app will support all type of vehicles. With this app, drivers who have loss in their business resolve their issues. This app has no central management office so this app will run in all cities. This application also provides the text to speech service for driver when it delivers a notification that there exists a request from passenger. Advancement to introduce open-source app will increase its market value.

# 2. Overall Description

## 2.1 Product Perspective

The project has two parts, the first one is an application run on android device and the second one is server on Firebase database. Android section will have two applications. One is master refers to driver and other one is client refers to passengers. If the driver accepts booking, then the server will send order confirmation to the passenger application. This application is managing passengers’ booking in quick and easy way with shortest time possible. With one click on the button you can order a vehicle if you are a passenger. With another click you accept or decline requests if you are a driver. No party to be added to control this work. Some types of vehicles still don’t have any online ride service. But with this application, any type of vehicle can register. This app will support all type of vehicles. This app will run in all cities. All vehicle drivers can register and earn money using this app without any limitation. This service will not have any central office.

## 2.2 Product Scope

## This project can be implemented in the form of mobile application to reduce the cost of hardware. (Uber and Careem is an example of it). This app is basically open-source online ride service so this app provides opportunity to drivers to register their Vehicles. With this app, drivers can register any type of vehicle and then start finding the nearest pick and drop. This app will support all type of vehicles. With this app, drivers who have loss in their business resolve their issues. This app has no central management office so this app will run in all cities. This application also provides the text to speech service for driver when it delivers a notification that there exists a request from passenger. Advancement to introduce open-source app will increase its market value.

## 2.3 Product Functionality

**Users**

All Users should Interact with the application Be able to login to their accounts using their email/phone and password provided.

Be able to order cab.

Be able to cancel cab.

Be able to bargaining the fare with drivers.

Be able to rideshare.

Be able to select payment methods.

Be able to logout from their profiles.

**Admin**

Admin should Be able to add, delete and modify database

Be able to add and remove users and set authority

Be able to monitor the system

Admin-Managers Be able to view all drivers

Be able to view the list of customers

Be able to view all the customer request

Be able to view all workers available

Be able to create, edit and delete information

Be able to search for information

Be able to notice the editing’s and changes made by the users.

## 2.4 Users and Characteristics

**Users**

All Users should Interact with the application Be able to login to their accounts using their email/phone and password provided.

Be able to order cab.

Be able to cancel cab.

Be able to bargaining the fare with drivers.

Be able to rideshare.

Be able to select payment methods.

Be able to logout from their profiles.

**Admin**

Admin should Be able to add, delete and modify database

Be able to add and remove users and set authority

Be able to monitor the system

Admin-Managers Be able to view all drivers

Be able to view the list of customers

Be able to view all the customer request

Be able to view all workers available

Be able to create, edit and delete information

Be able to search for information

Be able to notice the editing’s and changes made by the users.

**Driver**

Driver should Be able to accept, declaimed order.

Be able to add and remove users and set authority

Be able to monitor the system

Be able to view the list of customers

Be able to view all the customer request

Be able to view all workers available

## 2.5 Operating Environment

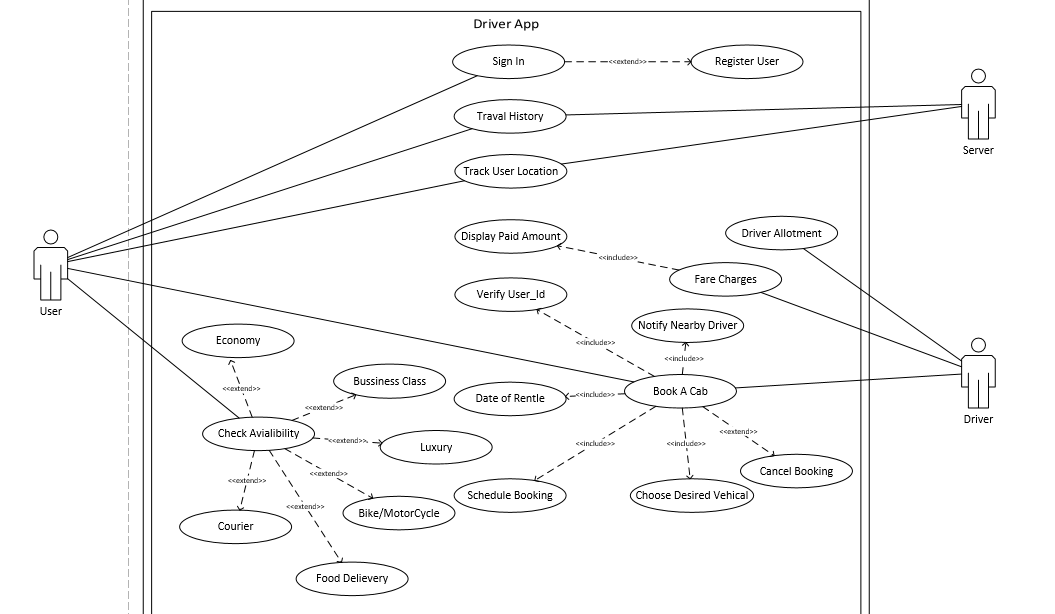
This application would run on Android OS. This application would run smoothly on almost all Android devices starting from Android 5.0.

# 3. Specific Requirements

## 3.1 Functional Requirements

* The Application must have a logo at the start of application.
* Every online booking needs to be associated with an account.
* One account cannot be associated with multiple users.
* Search results should enable users to find the most recent and relevant booking rides options.
* System should enable users to book / pay for their rides only in cash or credit card.
* The application shall keep track of all processes and changes happening to the data between login and logout times of the users.
* The application should enable the users to logout after using the application when the user clicks on the logout button.

### 3.1.1 Use Case Diagram



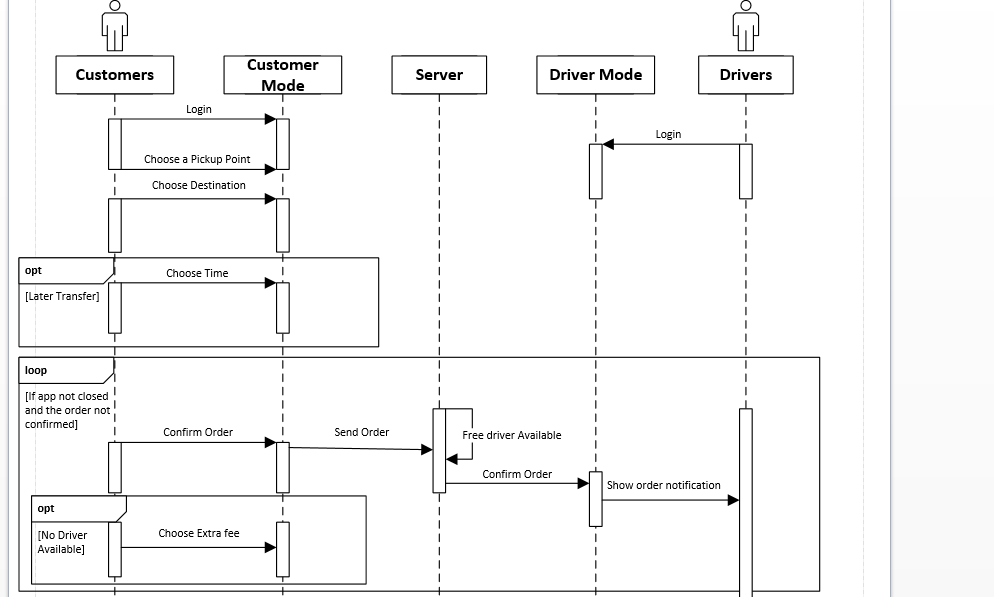
### 3.1.2 Use Case Descriptions

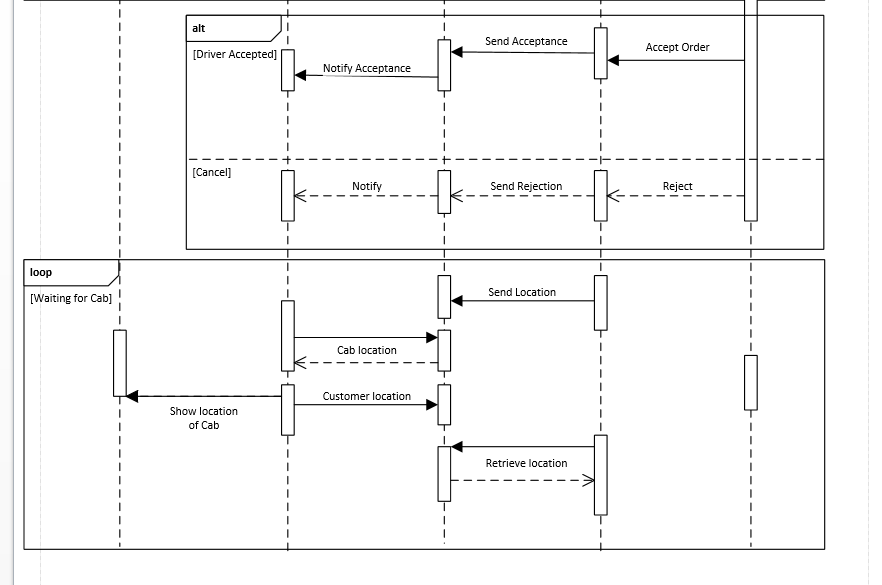
User: users can login through the phone no. and book a ride through mobile.

Driver: Seen the request of users and goes to the location .

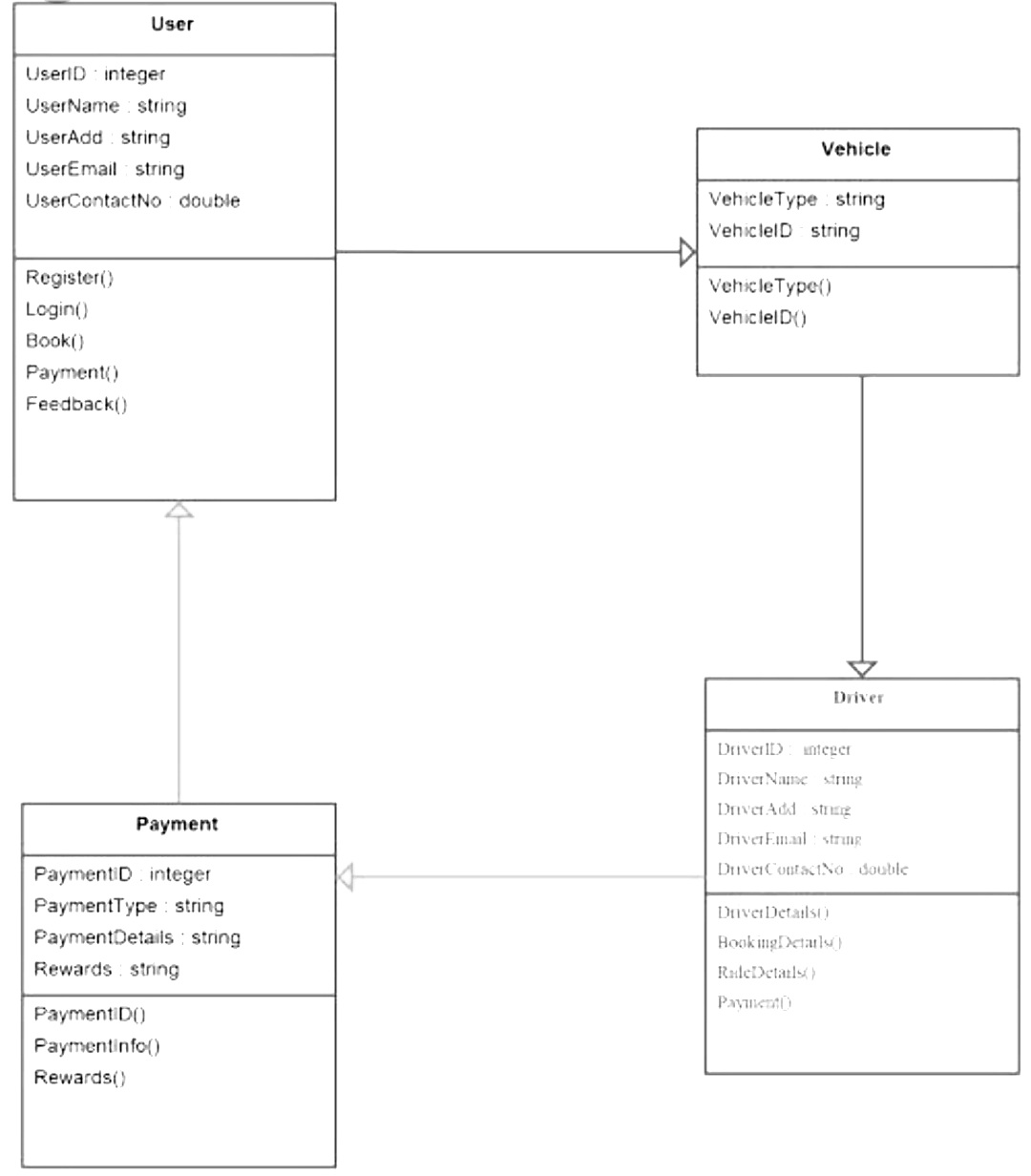
Server: Saved the record of User, Driver, and Travel History.

### 3.1.3 System Sequence Diagram / System Collaboration Diagram [Optional for BSCS Students ~ Mandatory for BSSE Students]



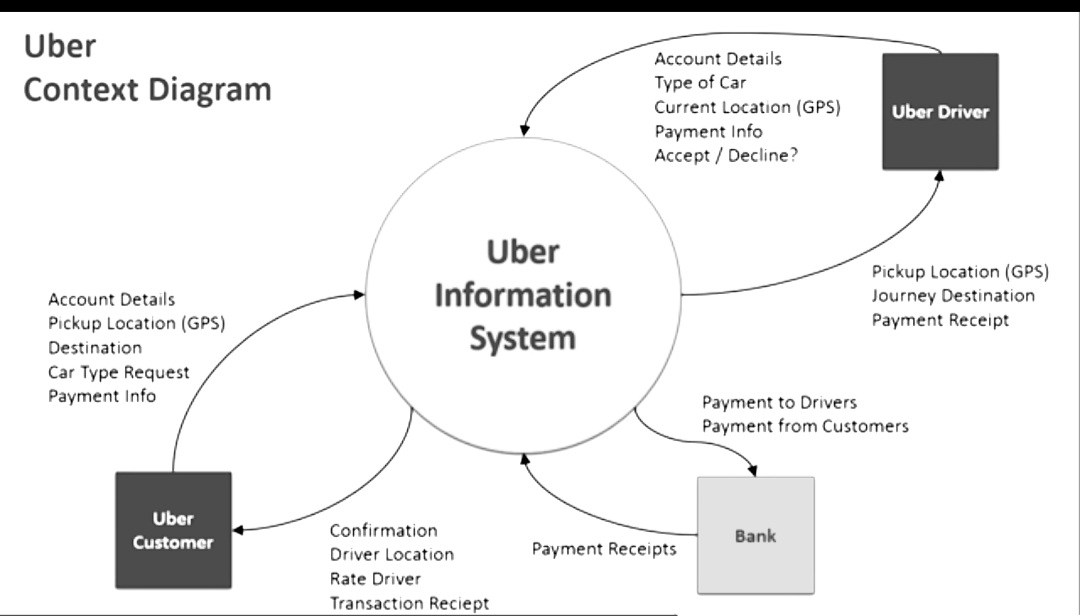


### 3.1.4 Class Diagram [Optional for BSCS Students ~ Mandatory for BSSE Students]

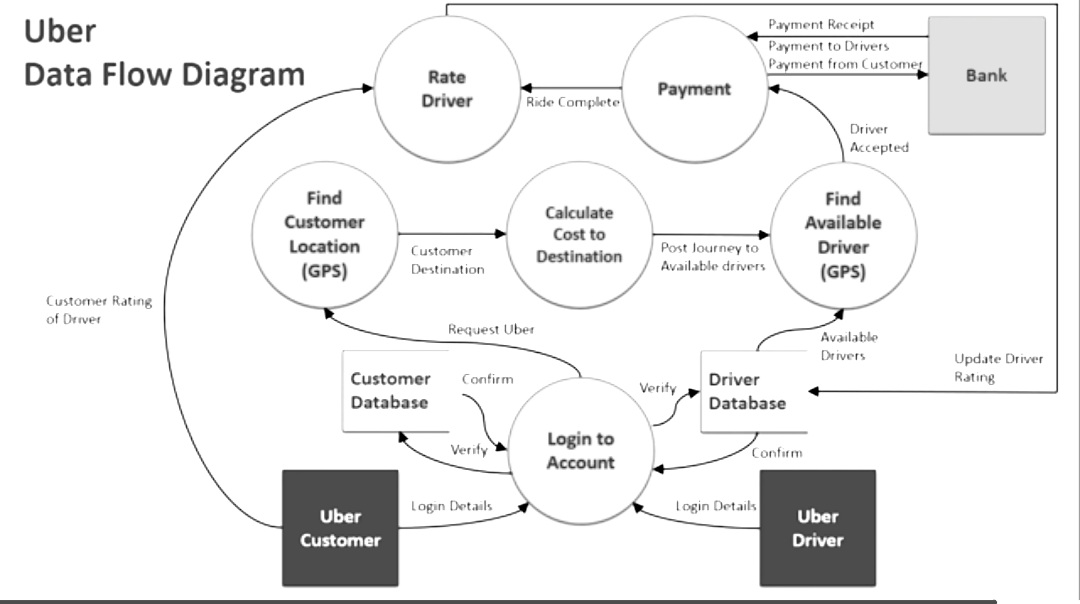


### 3.1.5 Data Flow Diagram

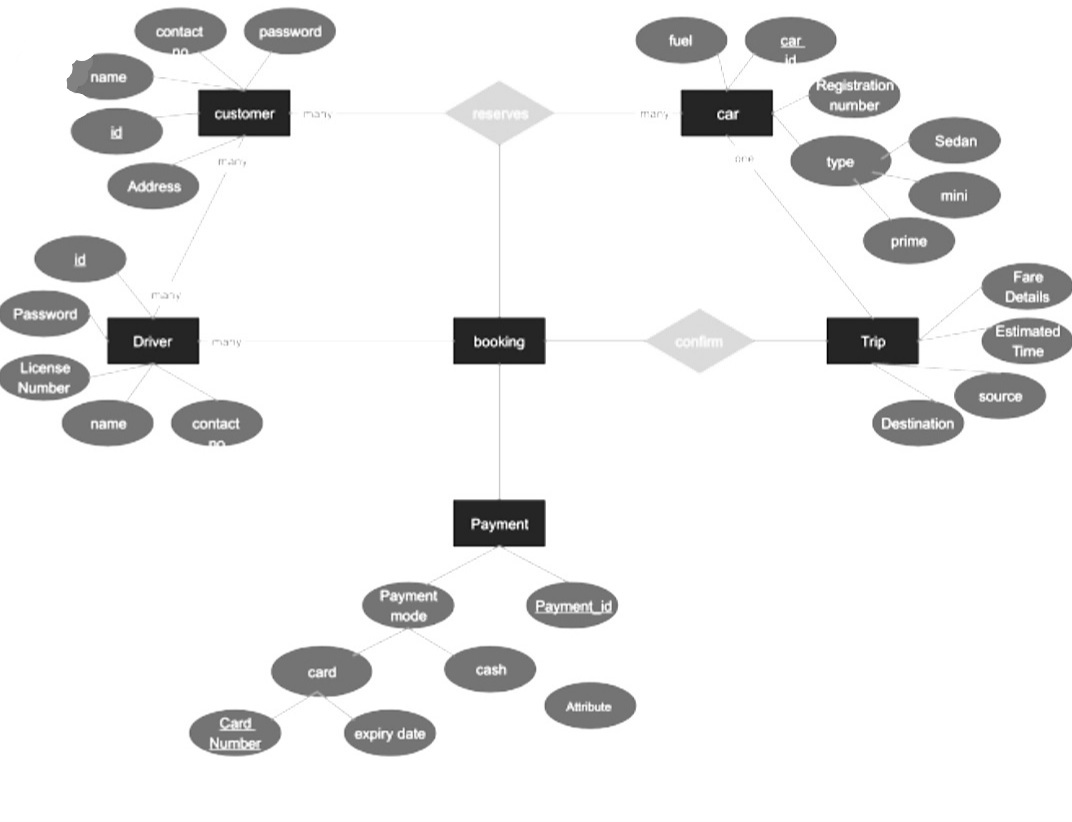
DFD Context (0-level) Diagram



DFD Level-1 Diagram

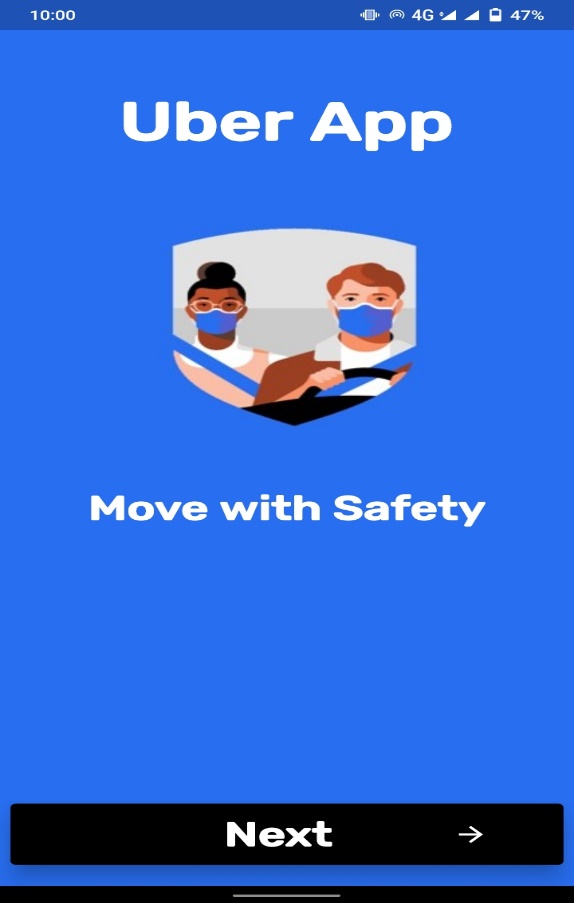


### 3.1.6 Database Design or ERD

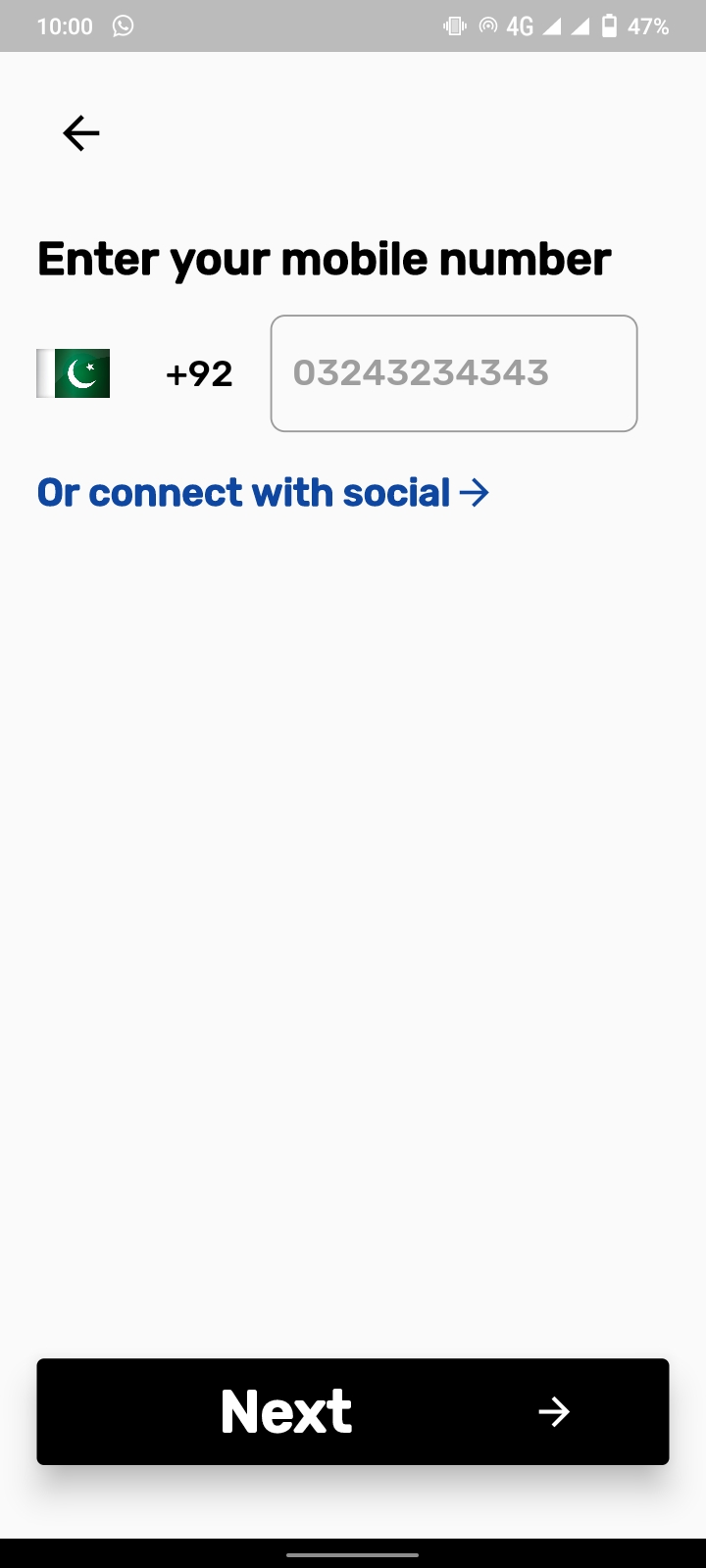


## 3.2 External Interface Requirements

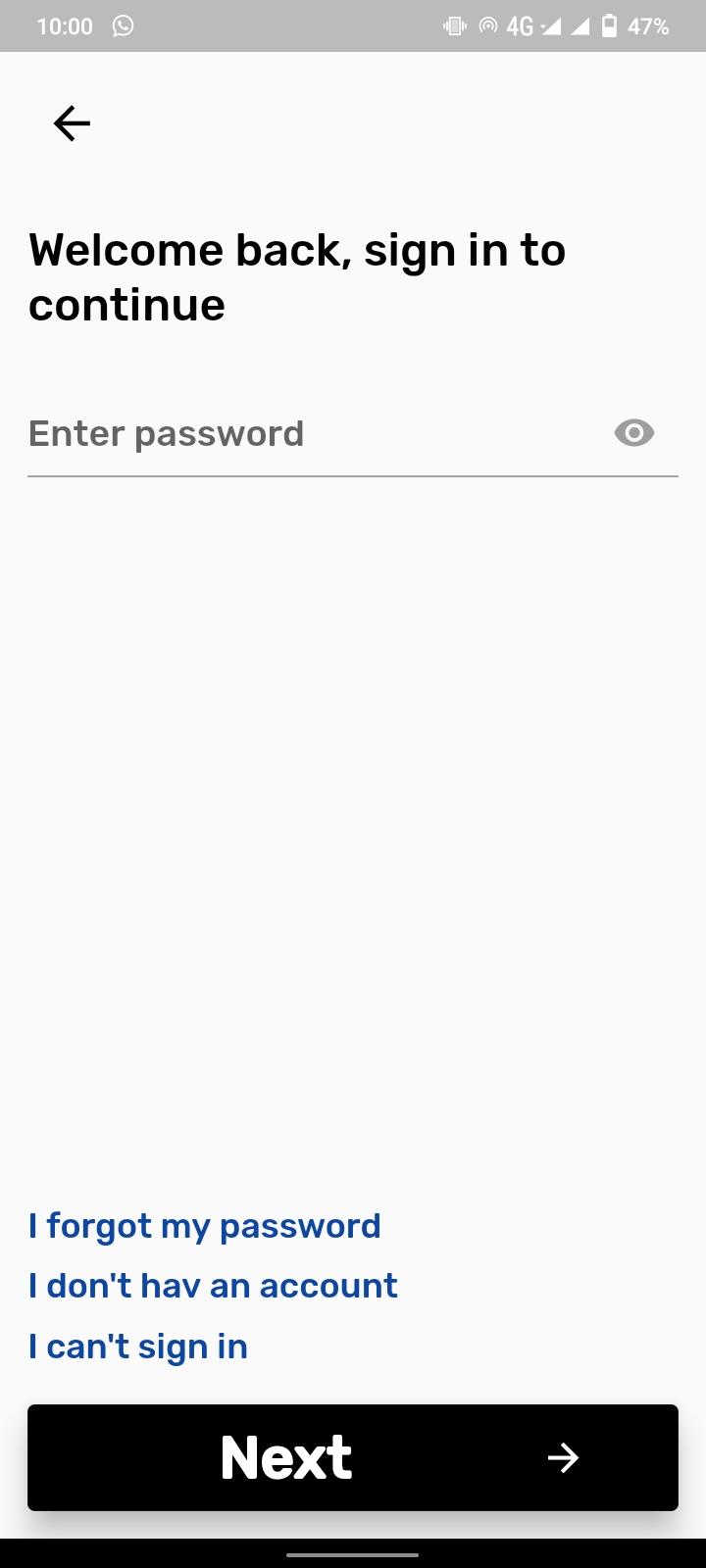
### 3.2.1 User Interfaces

**Splash Screen:**

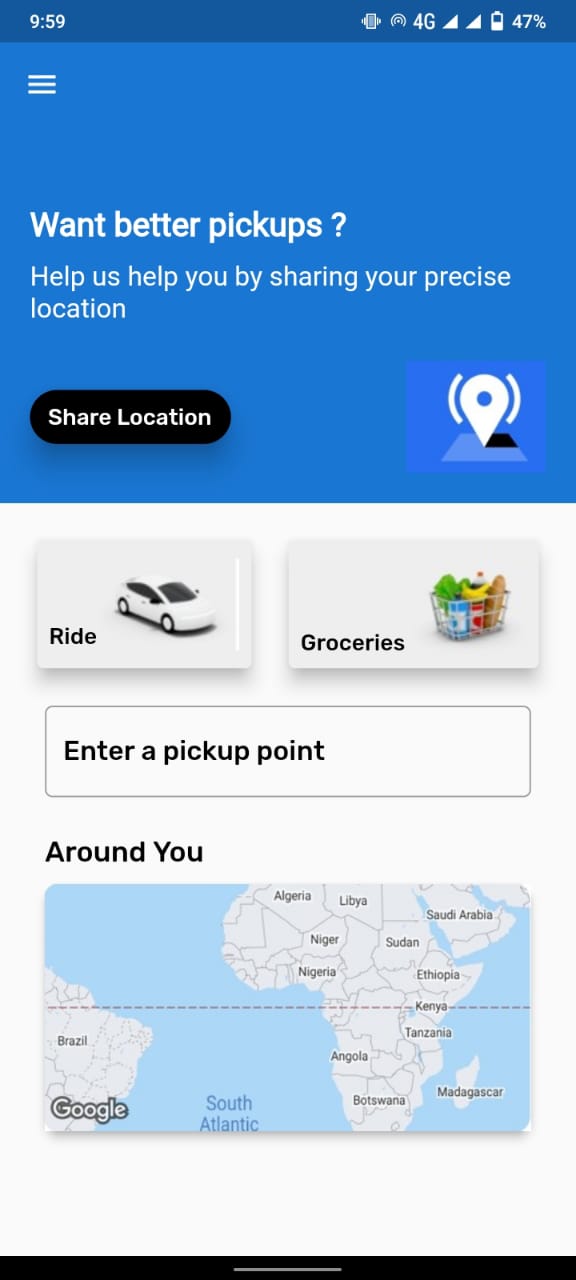
**Login Screen:**



**Password Screen:**



**Main Screen:**



### 3.2.3 Software Interfaces

We shall be using following APIs and Plugins:

* country\_state\_city\_picker: ^1.2.8
* datetime\_picker\_formfield: ^2.0.0
* splash\_screen\_view: ^1.0.3
* animations: ^1.1.0
* flutter\_stripe: ^2.0.1
* image\_picker: ^0.8.4+4
* firebase\_storage: ^10.2.3
* http: ^0.13.4
* shared\_preferences: ^2.0.8
* provider: ^6.0.1
* url\_launcher: ^6.0.17
* email\_auth: ^1.0.0
* cloud\_firestore: ^2.5.4
* firebase\_core: ^1.8.0
* firebase\_auth: ^3.1.4
* flutter\_local\_notifications: ^6.1.0
* crypt: ^3.0.1
* flutter\_markdown: ^0.6.9
* cupertino\_icons: ^1.0.2

# 4. Non-Functional Requirements

## 4.1 Performance Requirements

**Usability**

The Application should be easy to use by every user. In order to accomplish this objective, the system should have a simple and well-designed interface.

**Accessibility**

The Application should be made accessible to the people who live everywhere in the world. anyone can use the system regardless of the location and can get the information they acquire.

**Performance**

The performance of the application should be fast and efficient in adding information of drivers. The system should be available for user in real time and always up to date.

**Speed**

The application response time is a significant requirement because the action cannot be postponed or delayed. The application should be fast enough to satisfy the user’s needs and should not waste their time.

**EFFICIENCY**

Efficiency of any system is concerned with the minimum processing time as well as the optimal use of system resources in designing the proposed systems. Our android application will be efficient in using processing resources. It can be efficiently run on all android devices.

**Availability**

The Application should operate 24 hours a day.

**Friendly GUI’s**

The users of this application have different types of people and different levels of technical skills; therefore, the application should be understandable by all the users. Consequently, the Application should provide an easy to use, friendly Graphical User Interface (GUI).

## 4.2 Safety and Security Requirements

**Security**

Data inserted by user is secured and saved by this application, and will be redundant, in order to perform the exact action in specified situation.

**Maintainability**

The application should be maintained in order to perform the best of its ability.

## 4.3 Software Quality Attributes

**Extensibility** **and Stability**

The application should be flexible enough to allow improvements for the future and should be able to adapt any additional future change in activities; the application components can be modified for more changes and features allow the addition of new features without disturbing the main functionalities of the application.

**Disaster Recovery**

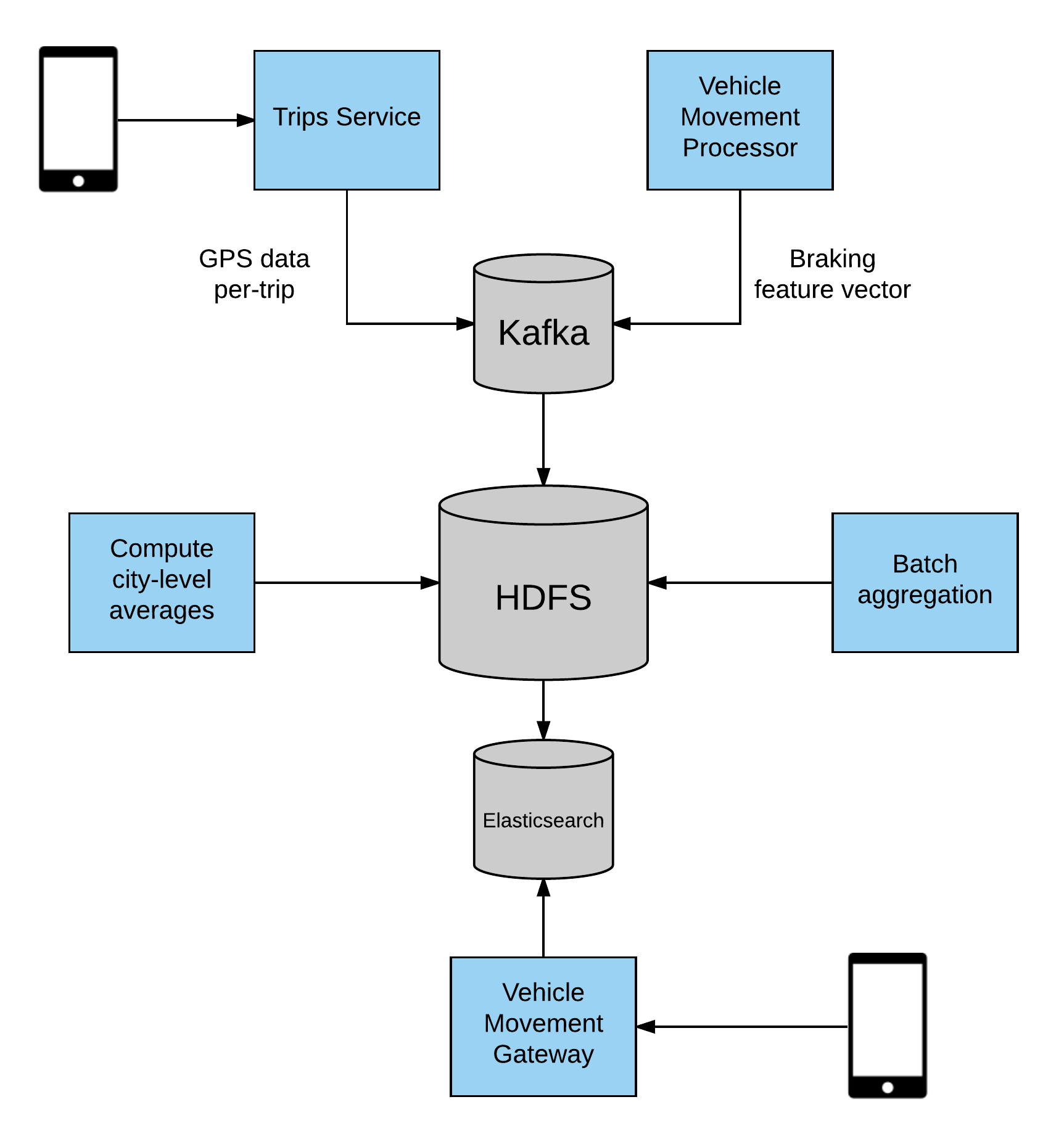
The application should be able to recover from an unsuitable problem .and should back up data.

**Error Handling**

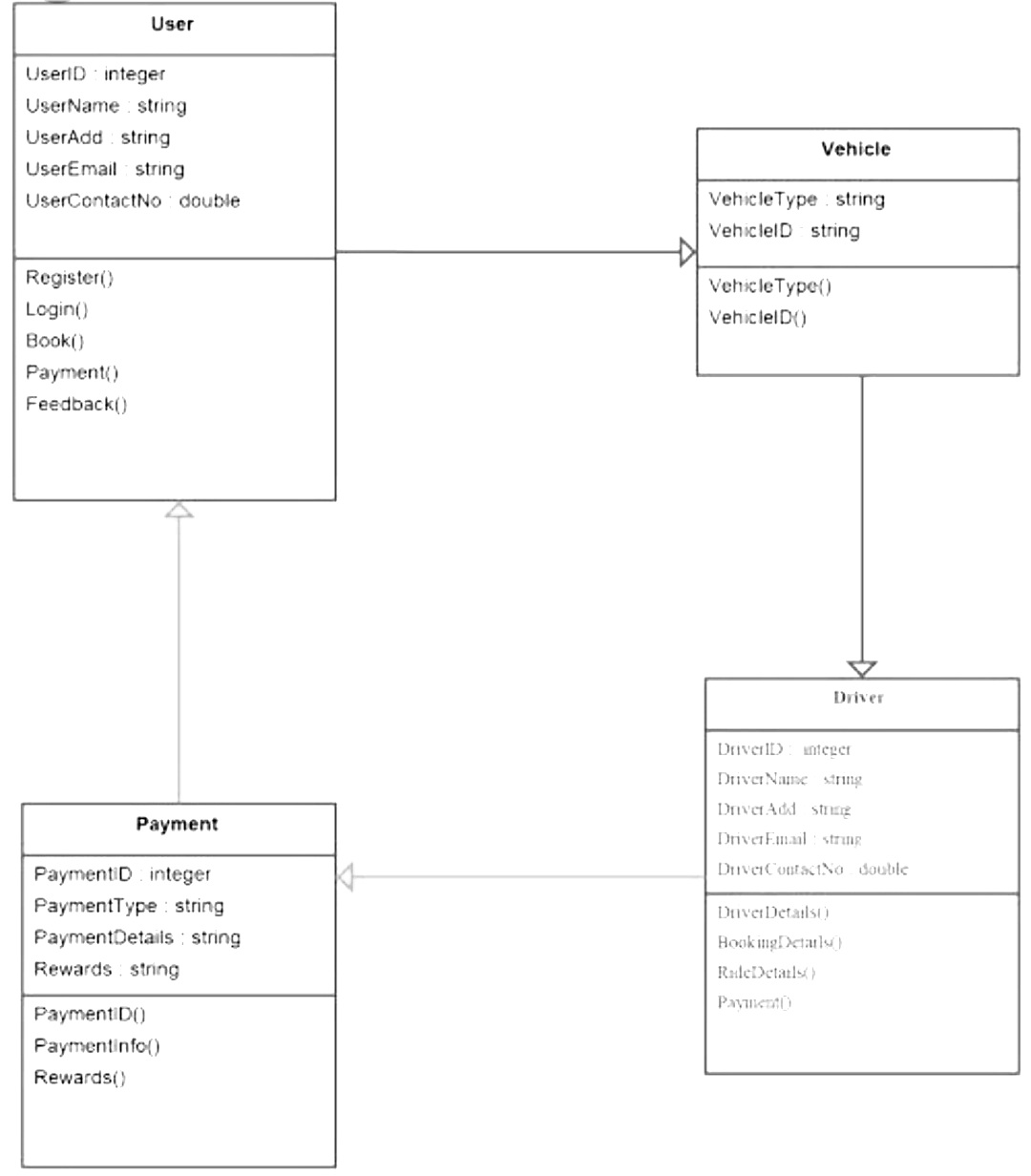
The application should be able to handle unexpected errors quickly and easily. “1. Any transaction will not take more than 10 seconds, etc.

# 5. Design Description

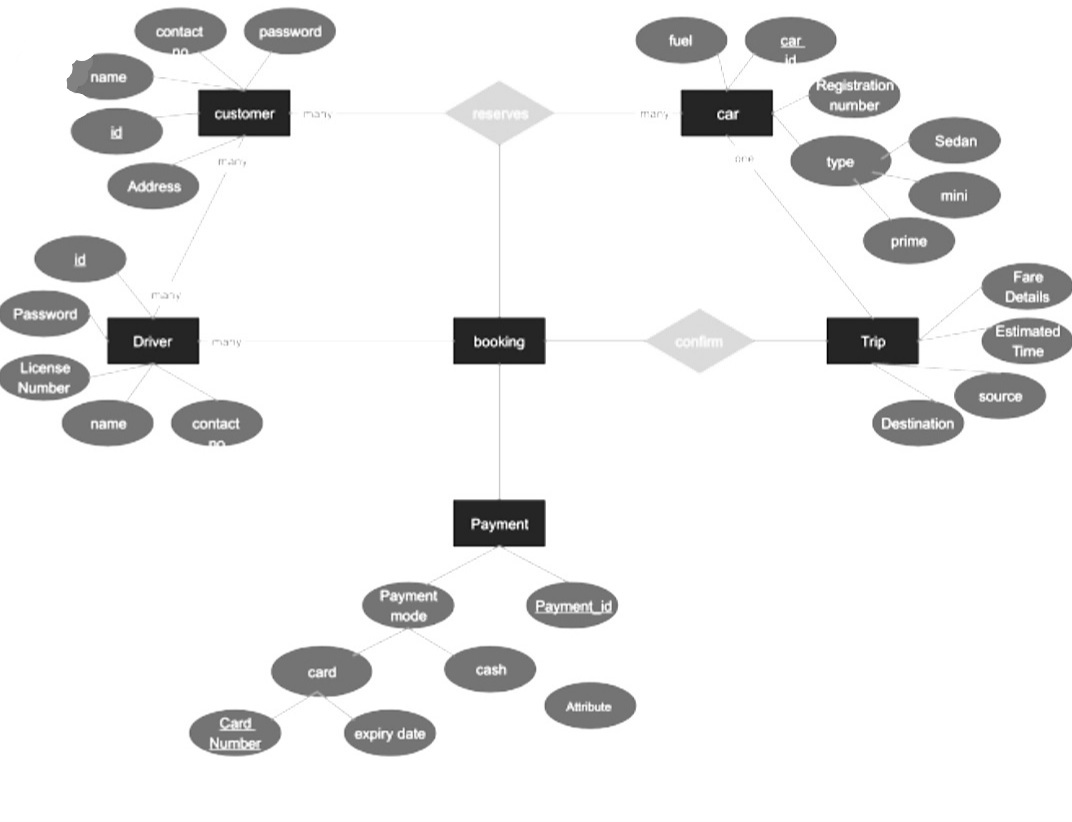
## 5.1 Composite Viewpoint



## 5.2 Logical Viewpoint



## 5.3 Information Viewpoint



## 

## 5.4 Interaction Viewpoint

