



A REAL TIME MOBILE  
APP FOR SHARING CARS

# Table of Contents

1. Introduction
2. Overall Description
3. Functional Requirement
4. Non-Functional Requirement
5. Analysis Design Document
6. System Design
7. Screenshot Samples
8. Let's Talk



# About Mobile App For Sharing Cars

## Introduction

It is an application intended to better utilize the vacant seat in the passenger's cars. It will help reduce pollution and the cost of travel. This application is GPS enabled.





# Scope Of The Application

Carpooling is one of the latest technologies, making travelling convenient and efficient for the commoner. It is also known as car-sharing, in which one can travel to their destination while sharing a vehicle and the expenses incurred. Hence fuel costs, tolls and the stress of driving will be reduced when more people travel together in one vehicle. It also helps to reduce traffic congestion and other poisonous gases in the air. It can help to save a lot of space in the parking lot.





# Scope Of The Application

Using the carpooling system is an intelligent decision during high fuel prices and high pollution periods. In our application, we will make an Android-based application that will allow passengers to collaborate with other like-minded people and plan out their journey using the easy UI of the app after signing in to it. Pre-registration ensures security, as only identified people get into the vehicle to establish trust. People will also be able to share expenses and not have to worry about reaching late while making new connections.



# Possible Features

- 1. Logging in
- 2. Personal Profile
- 3. Add vehicle details
- 4. Verifying Vehicle details
- 5. Add Licence details
- 6. Verifying details
- 7. Temperature & Vaccination status
- 8. Offer Ride
- 9. Take ride
- 10. View active rides
- 11. Accepting or Reject requests
- 12. Cancel Request
- 13. Cancel Ride
- 14. Modify ride
- 15. Real-time tracking for mobility updates
- 16. Getting the quickest and shortest route
- 17. Navigation
- 18. Pricing estimation
- 19. Messaging
- 20. Notifications
- 21. Billing section
- 22. Split-the-bill option
- 23. Multiple payment channel integrations
- 24. Feedback section
- 25. Report
- 26. Support feature



# Requirements Elicitation

## Use Case Approach

Use case approach is being used as a requirement elicitation technique. The use cases describe the ‘what’ of a system and not ‘how’. Hence, they only give a functional view of the system.

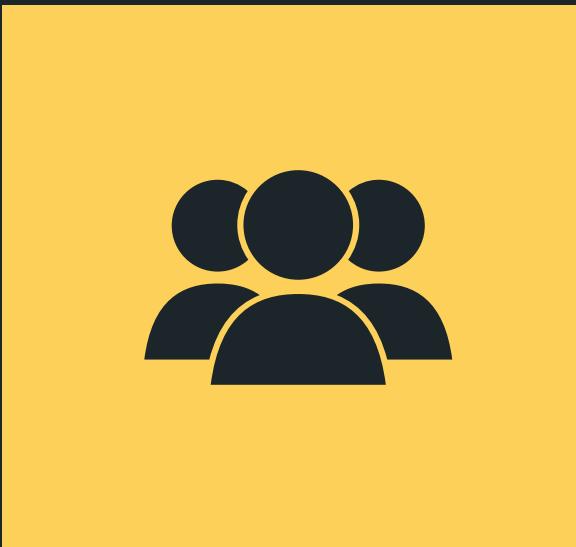
The components of the use case design include three crucial things:

1. Actor: It is the external agent that lies outside the system but interacts with it somehow. An actor, maybe a person, a machine etc. In our project actors are:
  - a. User offering ride
  - b. User taking ride
  - c. Admin
2. Use Cases: They describe the sequence of interactions between actors and the system. They capture who(actors) do what(interaction) with the system.
3. Use Case Diagram: A use case diagram graphically represents what happens when an actor interacts with a system.

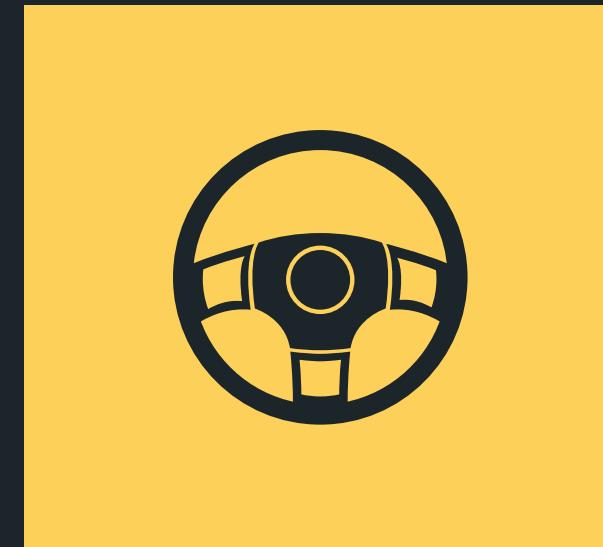


# Benefits

## What We'll Offer



Community



Quality Driving



Convenience



Efficiency



# Scrum Model

Scrum is an **agile development methodology** used in the development of Software based on an iterative and incremental processes



## Reasons For Choosing

### DIVIDE AND CONQUER

divided the car-pooling project into multiple iterations to focus on the quality

### ADAPTABILITY

Teams can easily make changes in the plan as per the feedback of the users

### LOW RISK

The application will be developed in sprints reducing the risk of getting off track

### CROSSFUNCTIONAL TEAMS

In every iteration different teams can work on different areas



# USERS & STAKEHOLDERS

USERS

CARPOOLERS

RIDERS

ADMIN

STAKEHOLDERS

CUSTOMERS

DEVELOPMENT  
TEAM

PROJECT  
MANAGERS

DESIGNERS

ADMIN

GOVERNMENT

# Functional Requirements

As a user, I want to register myself, so that I can log in to the application

As a user offering a ride, I want to add the car, so that I can offer the ride

As an admin, I want to accept or reject the cars that the user adds so that I can authenticate that the information is genuine

As a user offering a ride, I want to offer a ride by filling in the necessary details, so that anybody who wants to share the ride can request me

As a user offering a ride, I want to accept or reject users' taking a ride, so that I can choose according to my choice

As a user taking a ride, I want to view the offered rides, so that I can select the ride according to my requirements

# Functional Requirements

As a user, I want to reset the password, so that I can login again in case I forgot the password

As an admin, I want to view all rejected cars, so that I can view that information in future.

As a user, I want the application to verify users' login credentials, so that no user can access the application without registration.

As a user taking a ride, I want to request a ride to a user available, so that I can take a ride with that user.

As an admin, I want to view all accepted cars, so that I can view that information in future.

As a user, I want to edit my profile details, so that I can update the details whenever required.

# Non - Functional Requirements

As a user, I want to be able to run the application on all versions of android so that I can use it on any android device

As a user, I want the application to be available 99.99% of the time so that I can access it whenever required

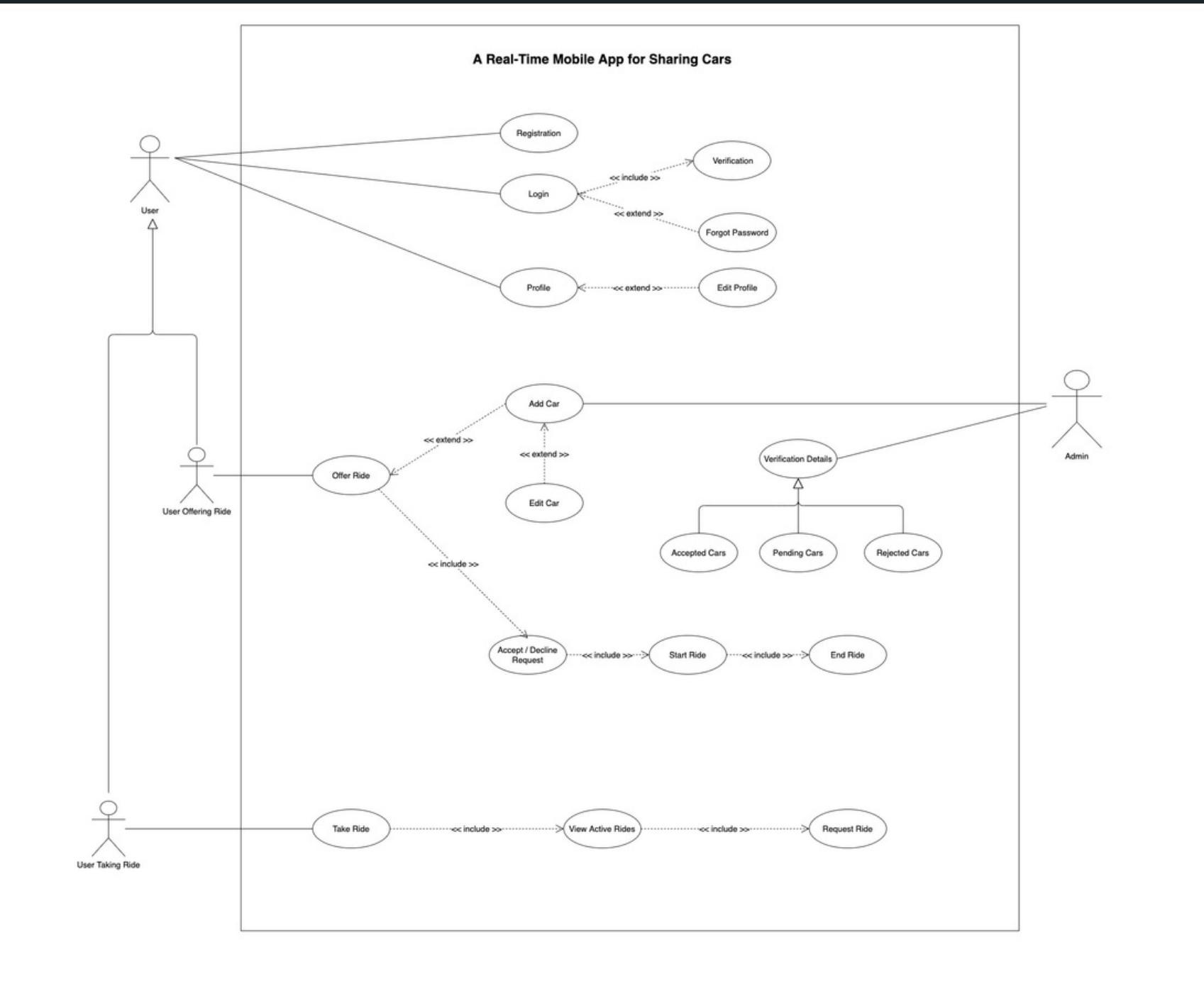
As a user, I want the application to respond quickly so that I can have a pleasant and seamless experience

As a user, I want the application not to share any of my details with any third party without my permission so that I can trust and use the application without any fear

As a user, I want an easy user interface so that it is easy to navigate in the application

As an application user, I want the number of users to be scalable in the database so that the application works smoothly

# Use Case Diagram

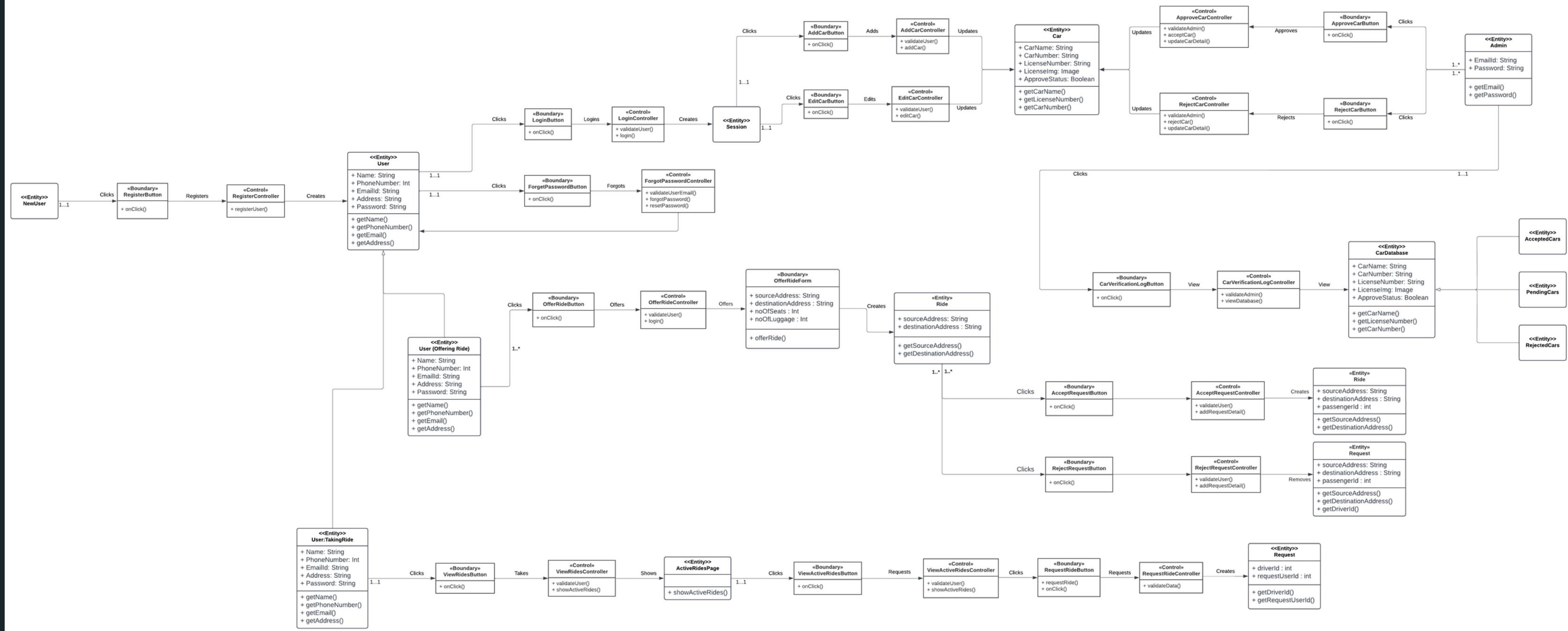


# Use Case Description

- **Registration :-** Users have to register by providing necessary details.
- **Login :-** After registration, users need to log in to access the functionalities of the application.
- **Offer Ride :-** The user will select this feature to offer the ride to another user.
- **Add Car :-** The user offering a ride will be required to add the car and licence details which further will be verified by the admin.
- **Accept / Decline Request :-** The user can accept or decline the request of the users who are requesting to share the ride.
- **Take Ride :-** In order to take a ride, the user needs to click on the take a ride button and provide the necessary details.
- **Request Ride :-** In order to take a ride, the user needs to request another available user.



# Complete Analysis Class Diagram

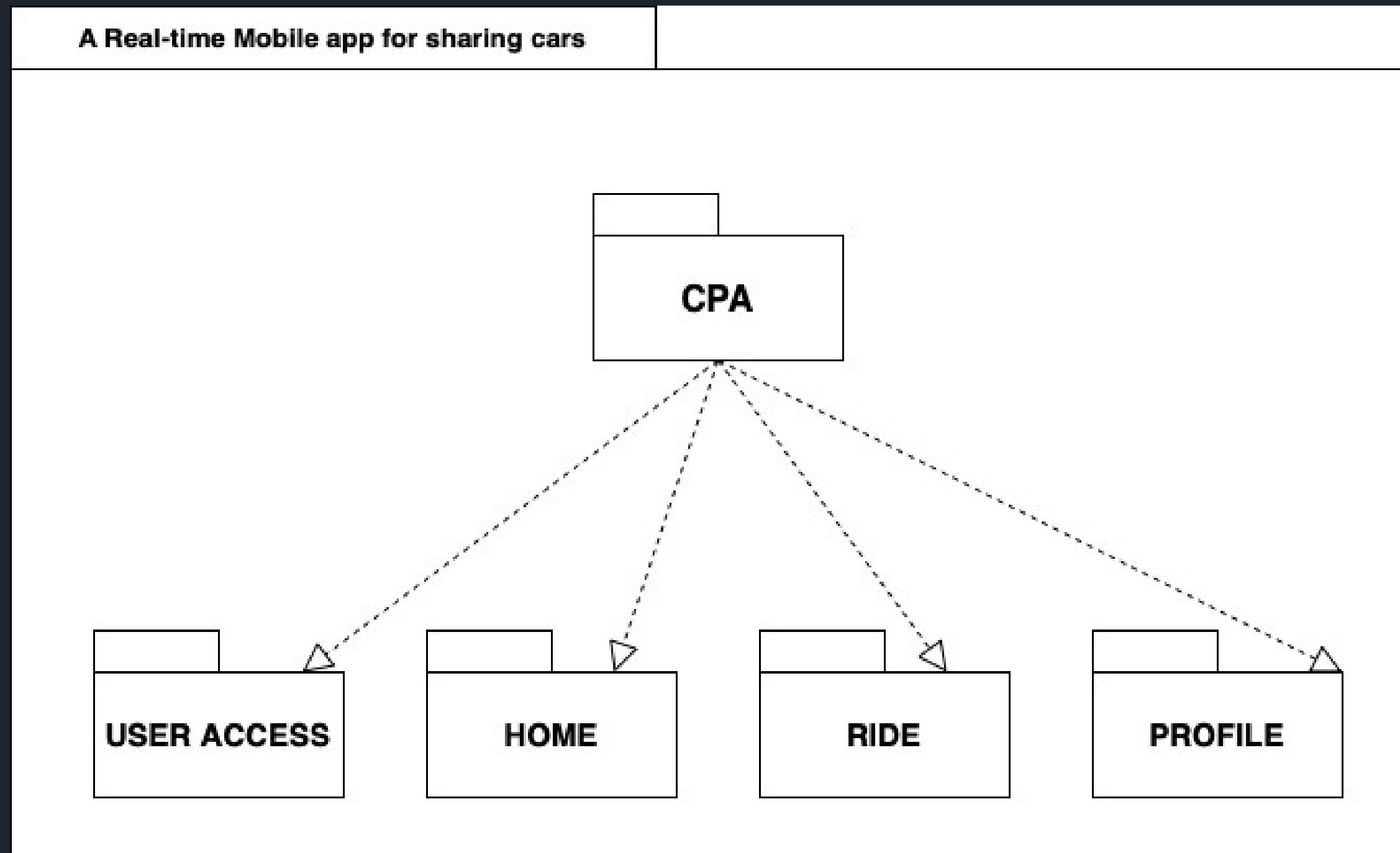


# Analysis Class Diagram

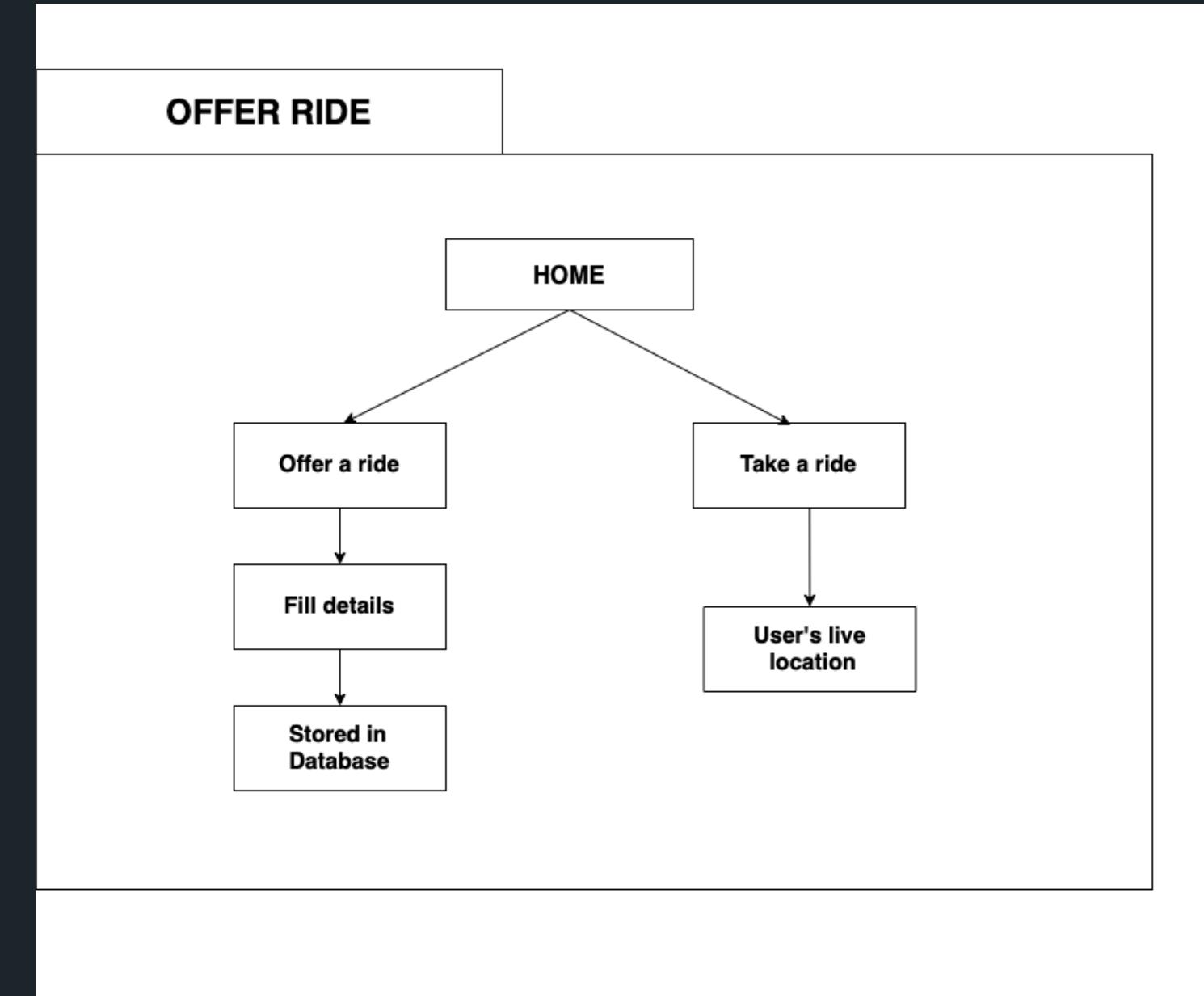
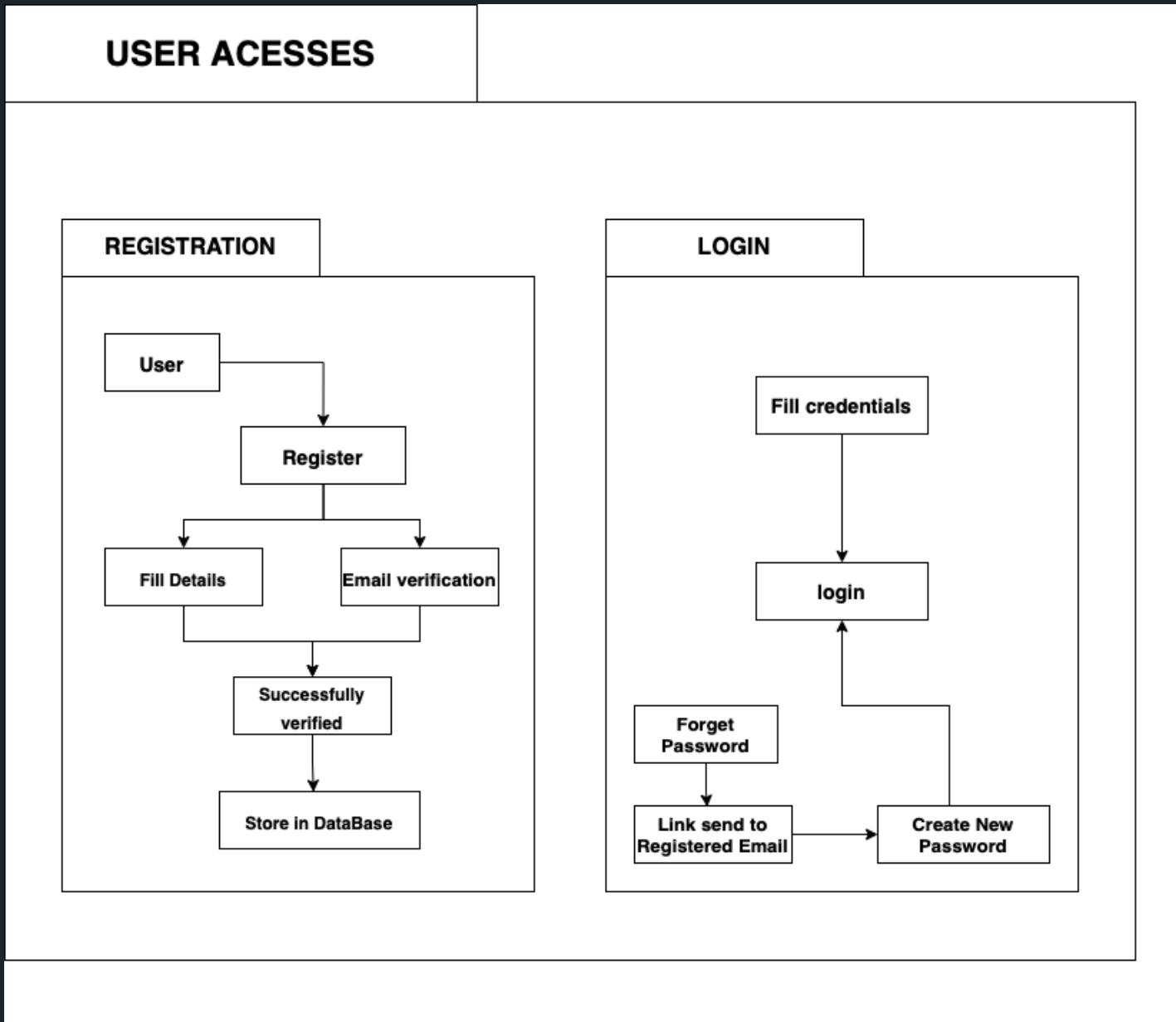


Presented by Group No - 7

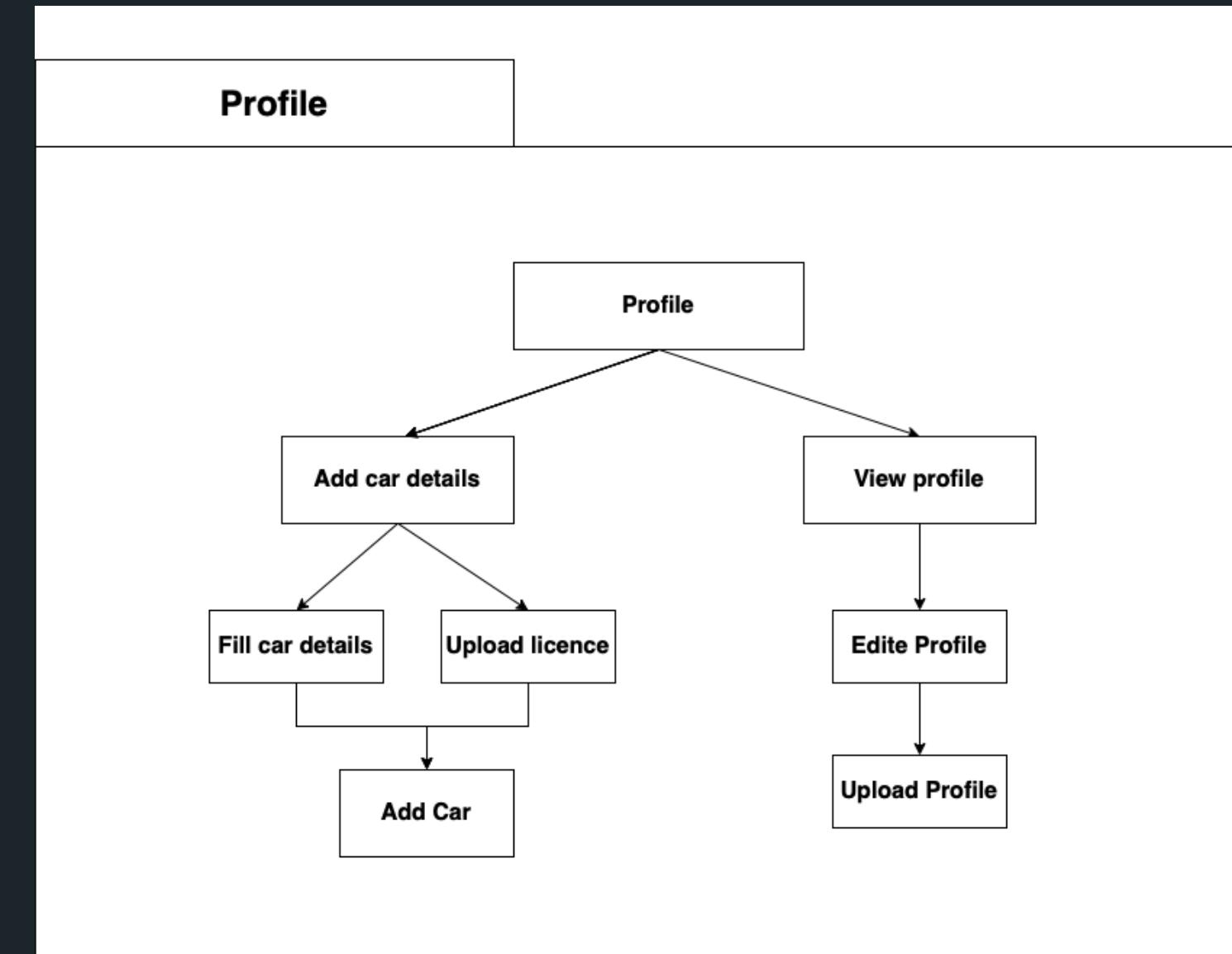
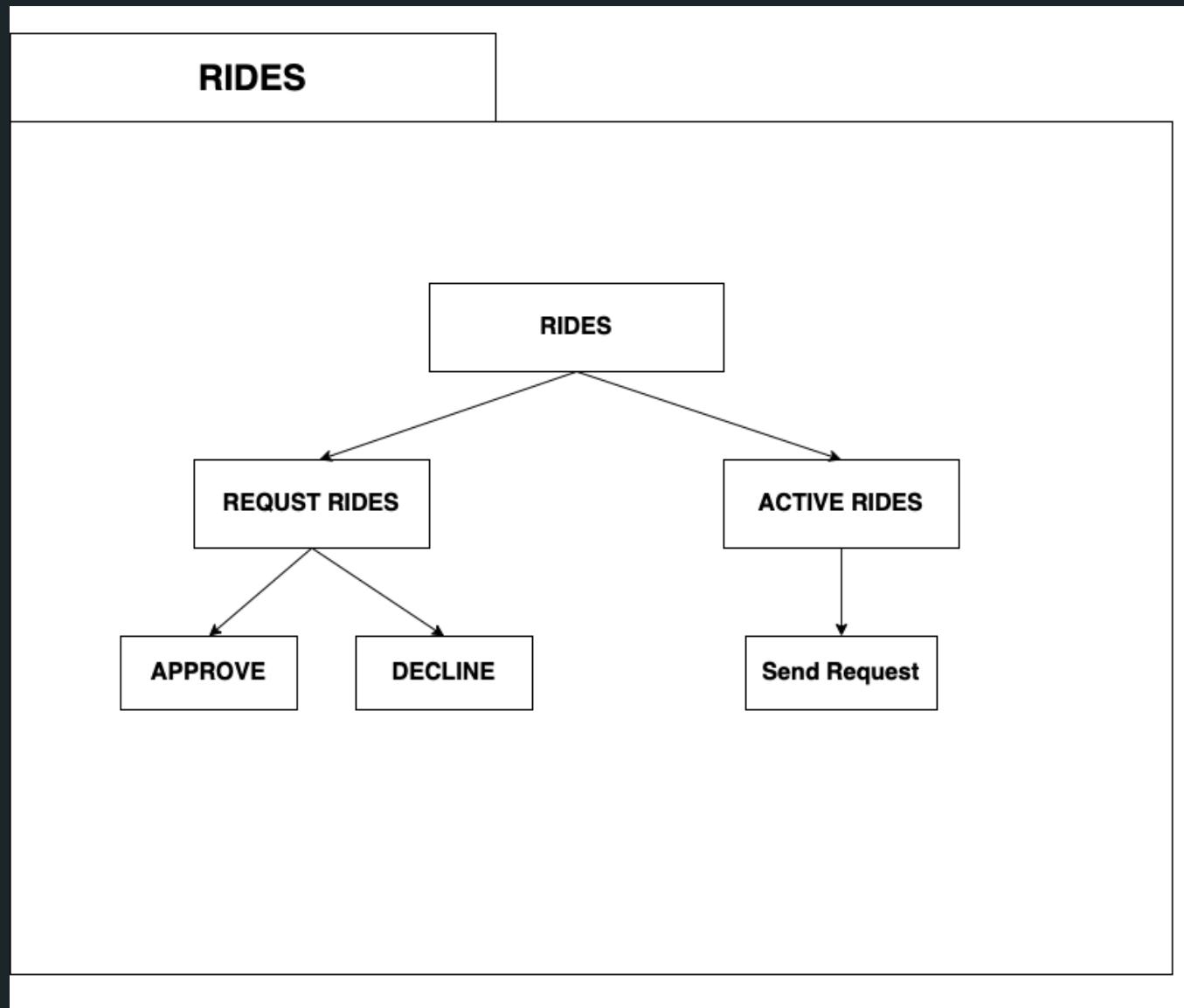
# System Design



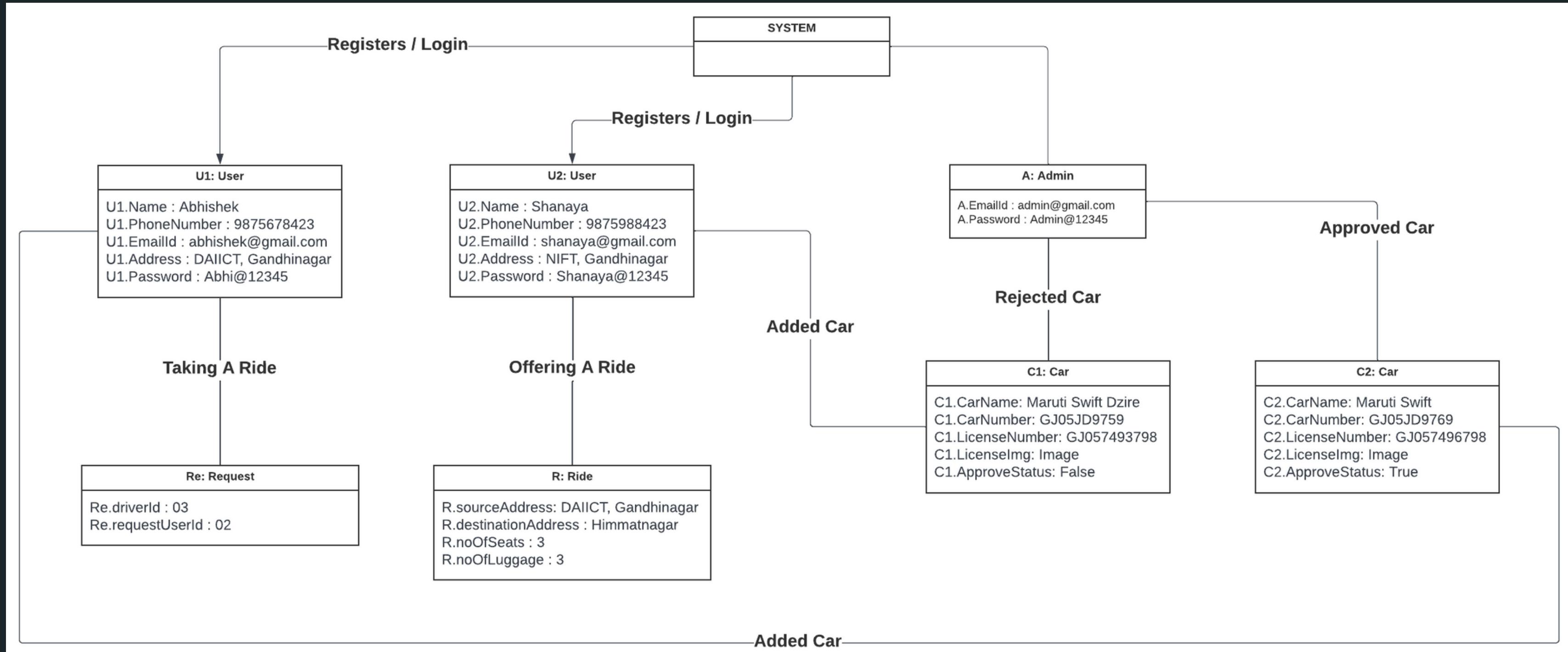
# Sub System Design



# Sub System Design

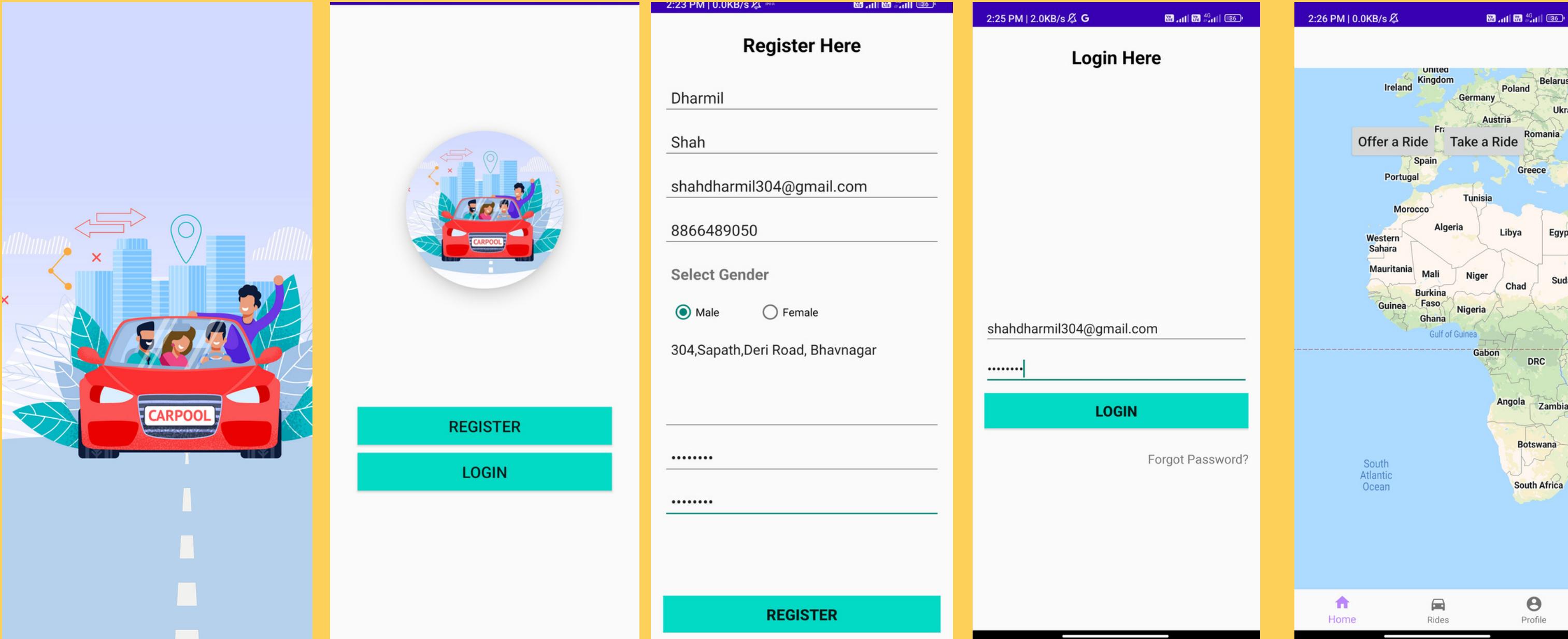


# Object Design



# App Prototype

## Screenshot Samples

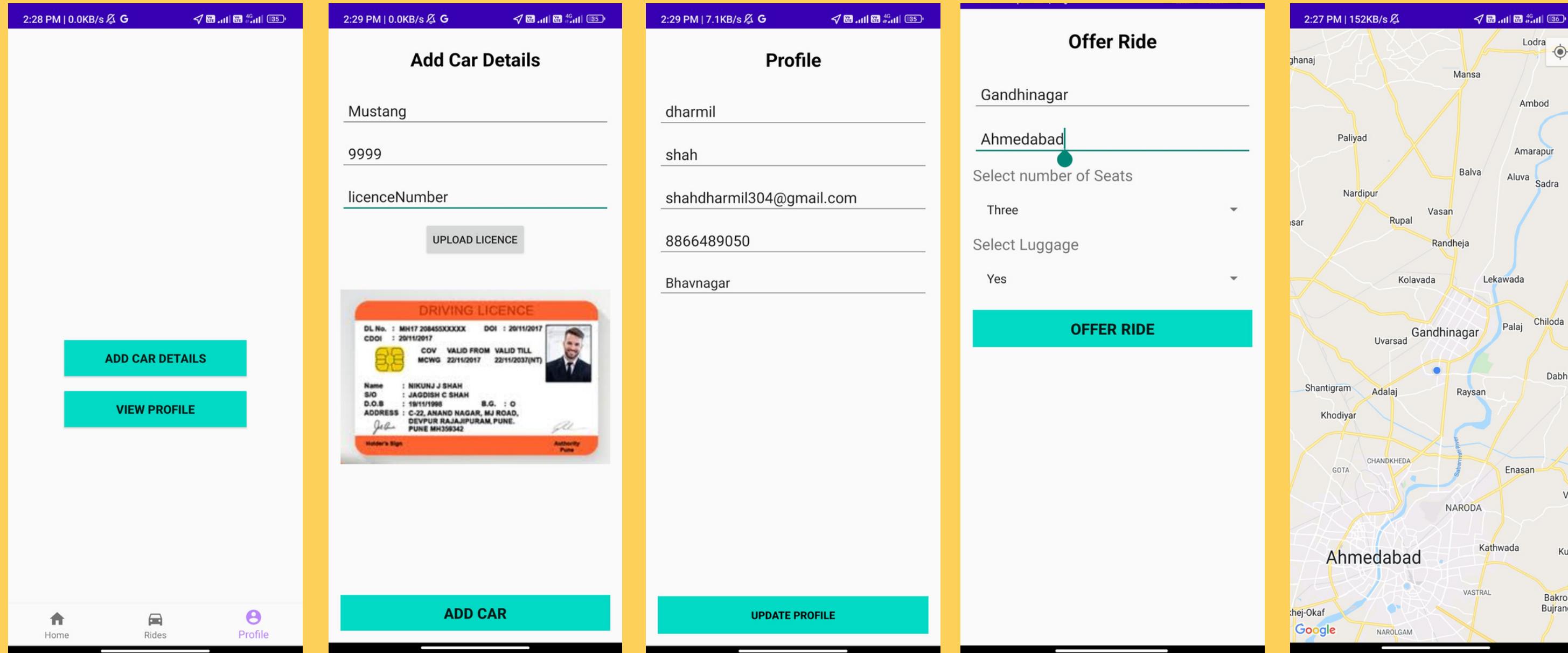


Presented by Group No - 7



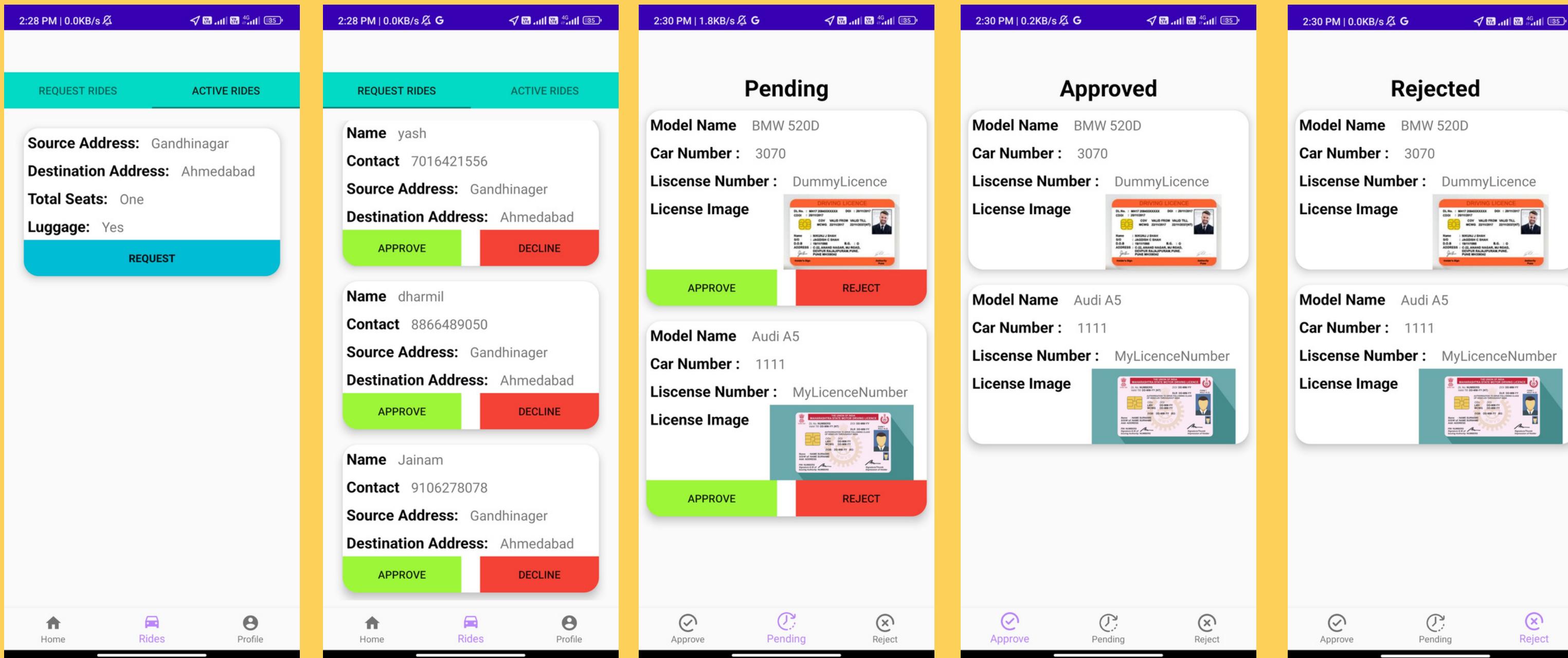
# App Prototype

## Screenshot Samples



# App Prototype

## Screenshot Samples



# Let's Talk

We're open for questions  
and comments.

DEVANSH DALMIA	202112002
JAINAM SHAH	202112016
ABHISHEK ANEJA	202112020
YASH SOLANKI	202112028
KARTIK TRIVEDI	202112056
DHARMIL SHAH	202112109
DARSHI DESAI	202112113
KHUSHI SHAH	202112123
NISHA SHAH	202112125
SHREYA ACHARYA	202112127

