Car Pooling Project – Documentation

Project Rules & Regulation (SDLC)

Requirement Analysis

- Number of users: Guest, Rider, Driver, Admin
- Budget: Academic Project

Step 1: How Many Users Work on Website

- Guest (Visitor) → Website
- Rider (Passenger) → Website
- Driver → Website
- Admin → Admin Panel

Step 2: Define Work of Each User

Guest (Visitor):

- Visit website
- Search rides (without booking)
- Contact Us

Rider (Passenger)

- Sign-up / Login → users.tbl
- Manage Profile → users.tbl
- Search Ride → rides.tbl
- Book Seat → bookings.tbl
- Manage Bookings → bookings.tbl
- Provide Feedback / Rating → feedbacks.tbl
- Contact Us → contact_us.tbl
- Logout → users.tbl

Driver

- Sign-up / Login → users.tbl

- Manage Profile → users.tbl
- Add Vehicle → vehicles.tbl
- Offer Ride → rides.tbl
- Manage Rides → rides.tbl
- Confirm/Reject Bookings → bookings.tbl
- Contact Us → contact_us.tbl
- Logout → users.tbl

Admin

- Login → admin.tbl
- Dashboard (overview of system) → admin.tbl
- Manage Users (Drivers, Riders) → users.tbl
- Manage Vehicles → vehicles.tbl
- Manage Rides → rides.tbl
- Manage Bookings → bookings.tbl
- View Reports → bookings.tbl, rides.tbl, users.tbl
- Manage Feedback → feedbacks.tbl
- Logout → admin.tbl

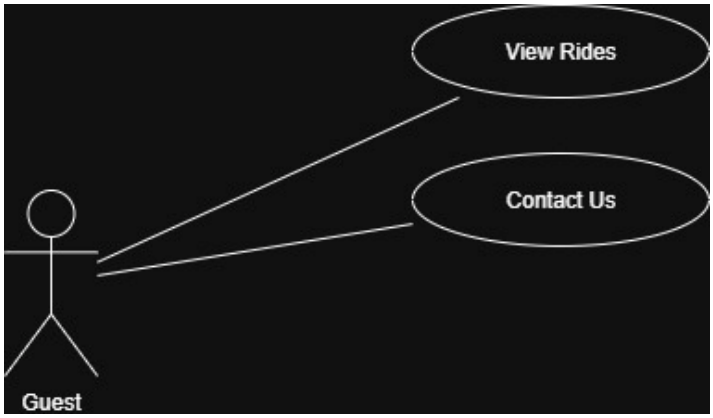
Step 3: Define Panel by User

- Website: Guest / Rider / Driver
- Admin Panel: Admin

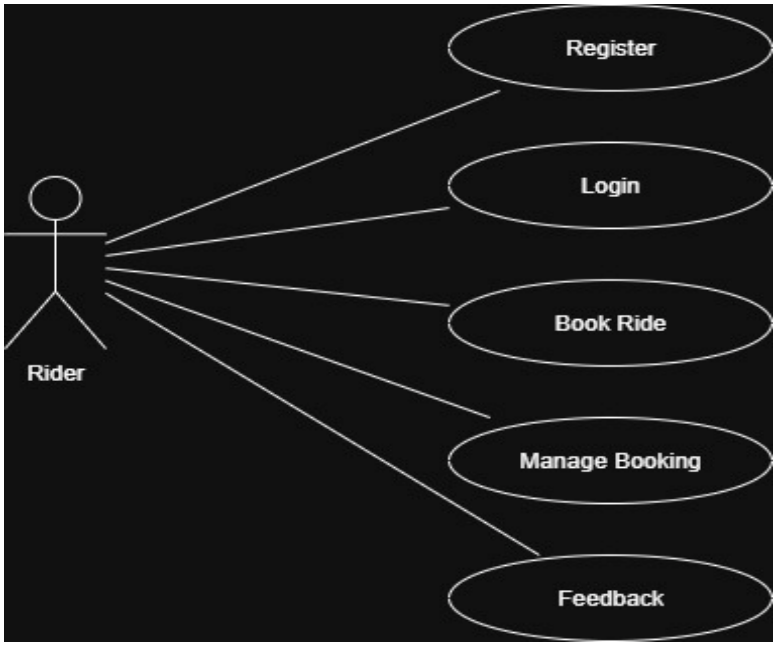
Step 4: Define Some Diagrams

Use Case Diagram (Text Explanation):

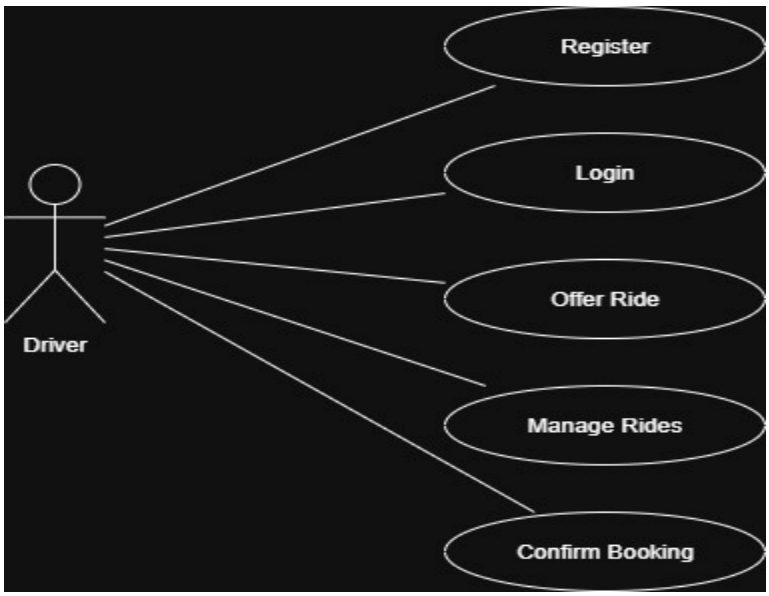
- Guest → View Rides, Contact Us



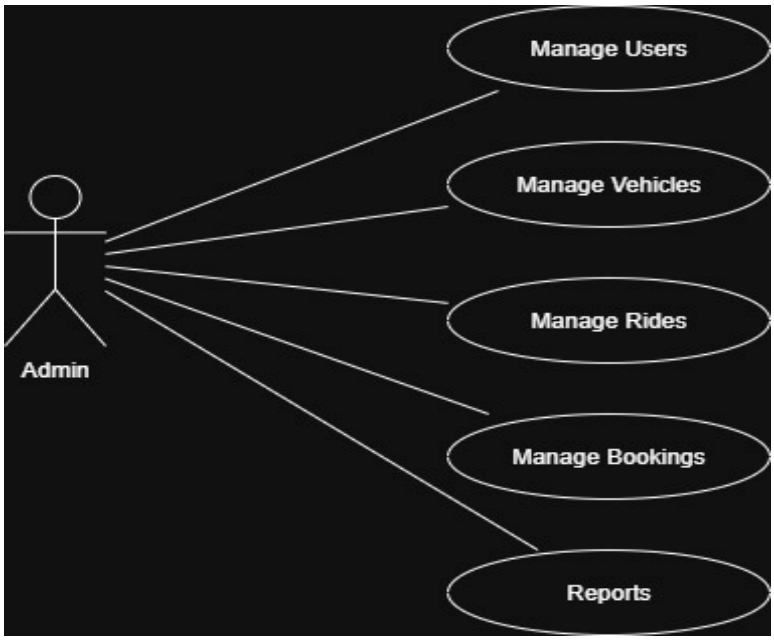
- Rider → Register, Login, Book Ride, Manage Booking, Feedback



- Driver → Register, Login, Offer Ride, Manage Rides, Confirm Booking

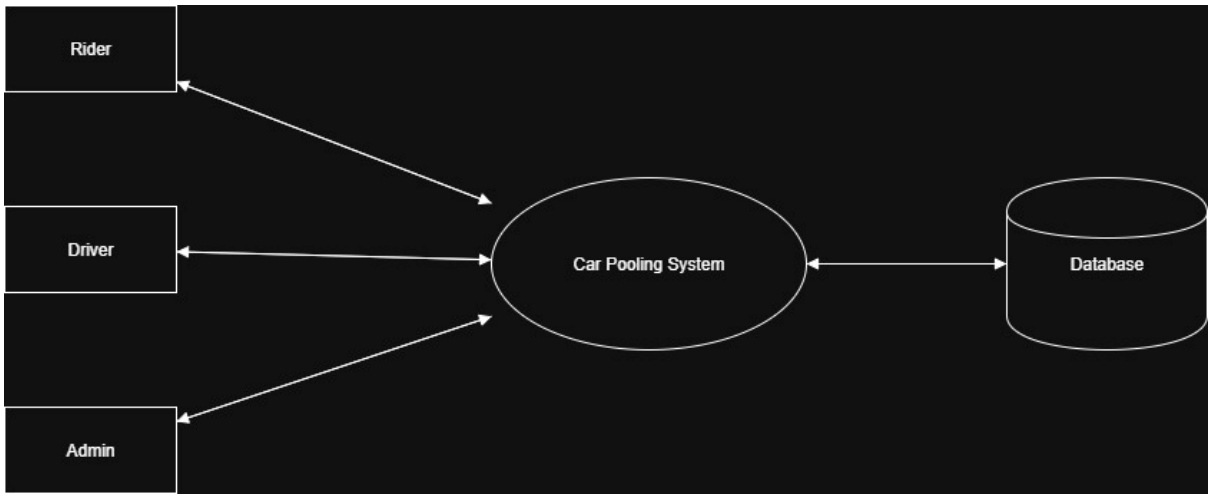


- Admin → Manage Users, Vehicles, Rides, Bookings, Reports

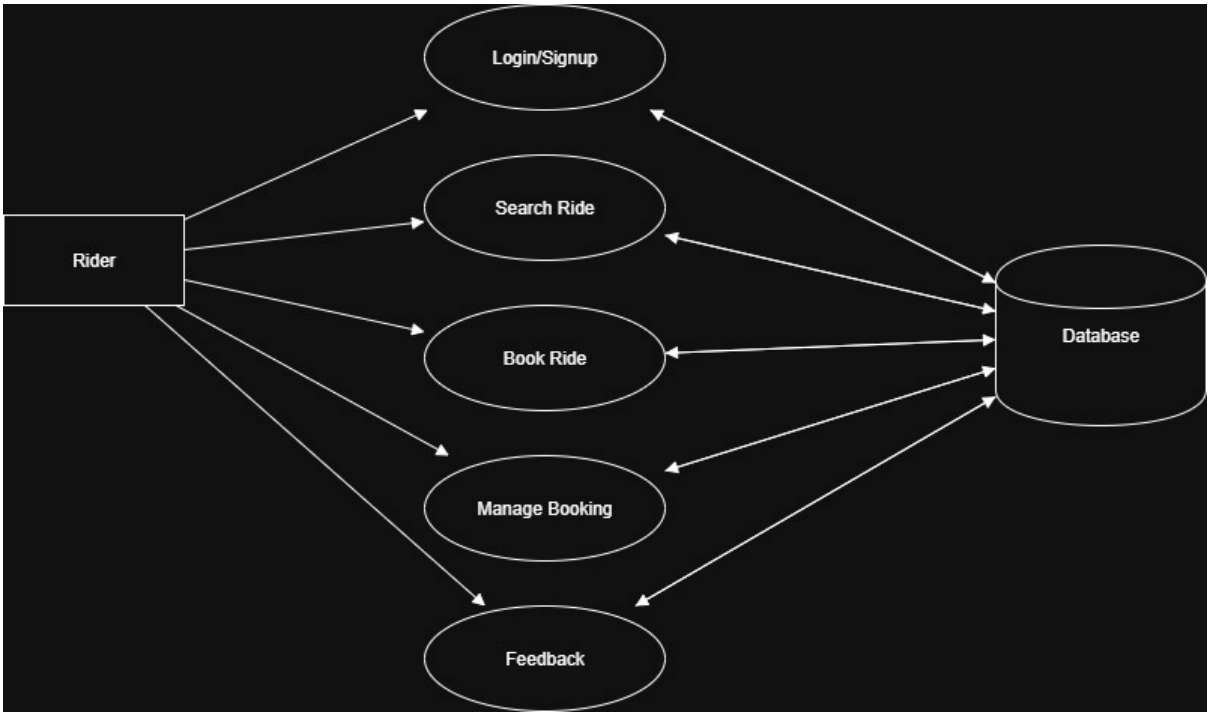


DFD – Data Flow Diagram (Explanation):

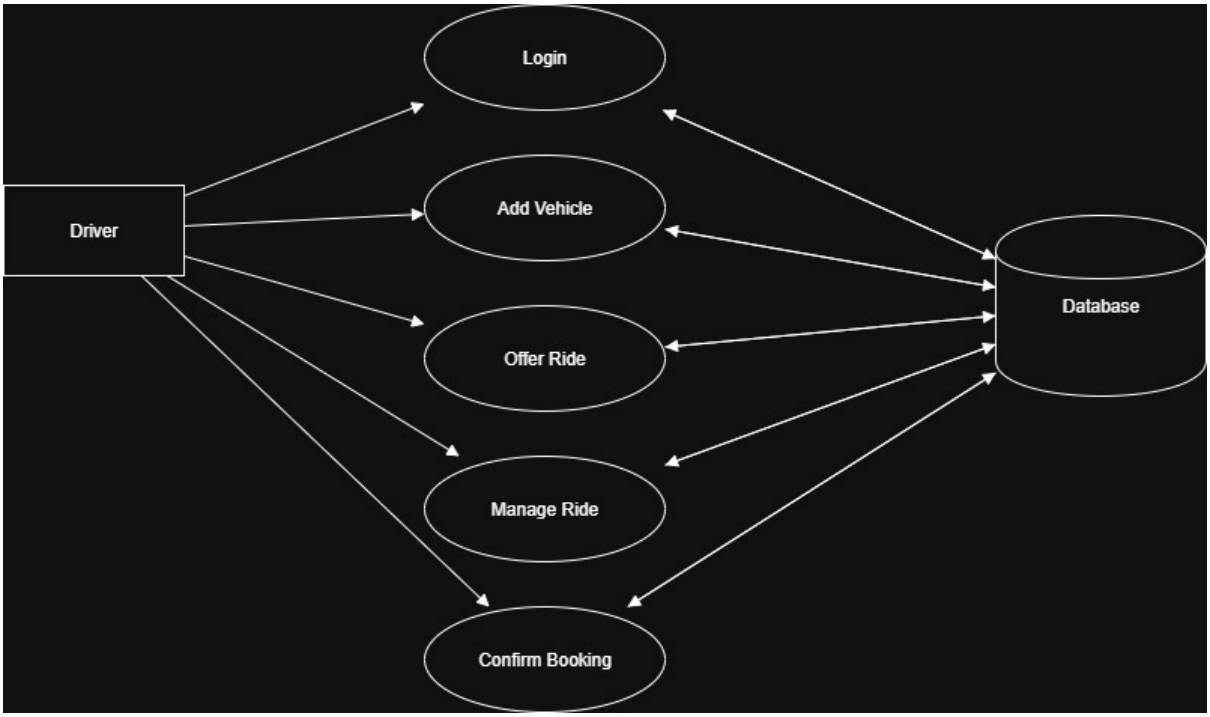
- **Level 0 (Context Diagram):** Users (Driver, Rider, Admin) interact with Car Pooling System → Database



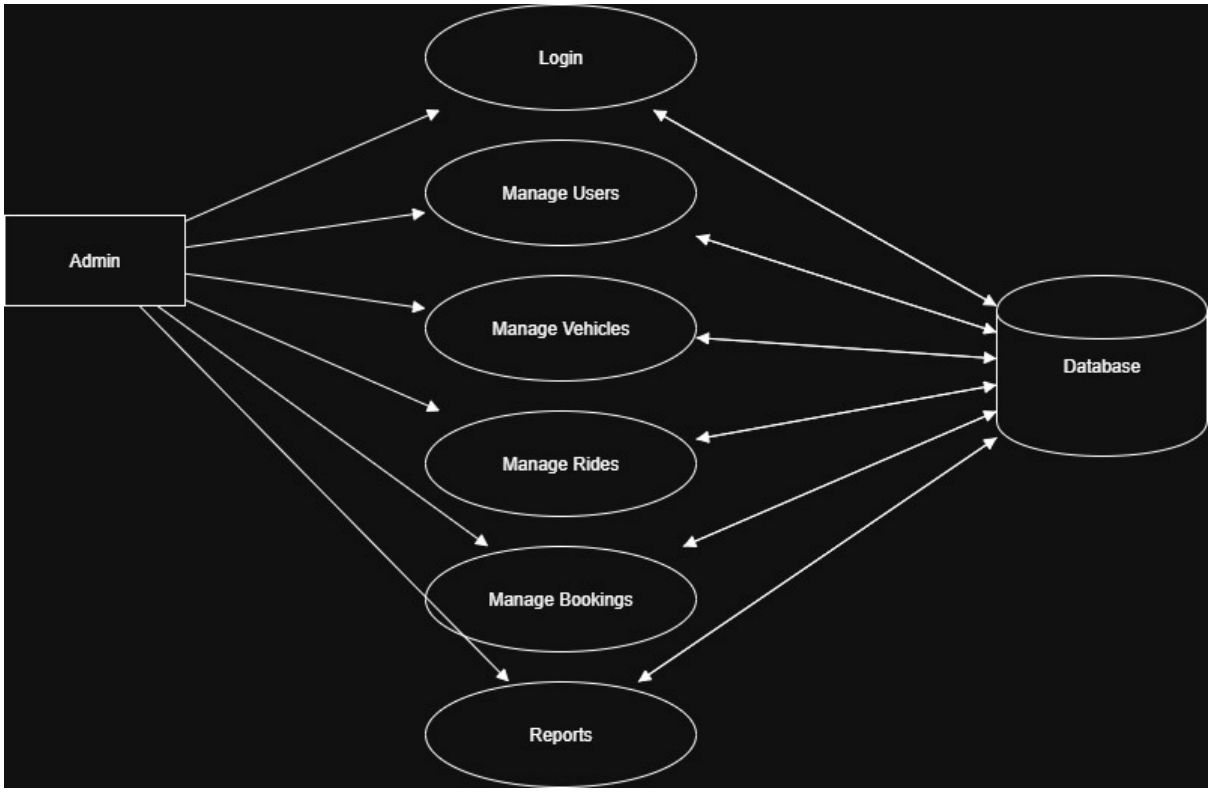
- **Level 1 (Rider side):** Rider → (Login/Signup, Search Ride, Book Ride, Manage Booking, Feedback) → Database



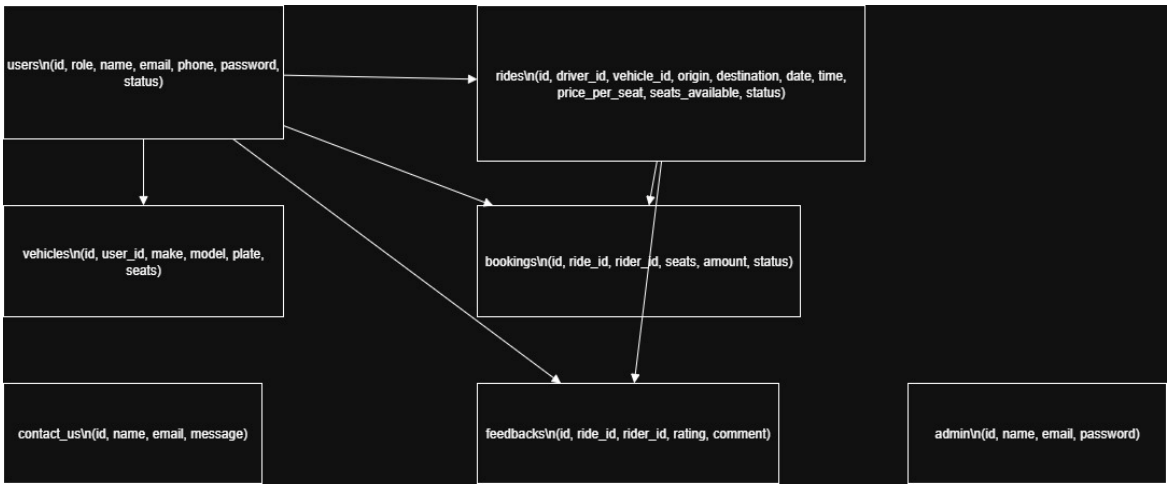
- **Level 1 (Driver side):** Driver → (Login, Add Vehicle, Offer Ride, Manage Ride, Confirm Booking) → Database



- **Level 1 (Admin side):** Admin → (Login, Manage Users, Vehicles, Rides, Bookings, Reports) → Database



- ER Diagram:



Step 5: Define Total No. of Tables List

- users.tbl
- vehicles.tbl
- rides.tbl
- admin.tbl
- bookings.tbl

- feedbacks.tbl
- contact_us.tbl

users.tbl

- id (PK, Auto)
- role (enum: guest, rider, driver, admin)
- name
- email
- phone
- password
- status (active/inactive)

vehicles.tbl

- id (PK, Auto Increment)
- user_id (FK → users.id)
- make
- model
- plate
- seats

rides.tbl

- id (PK, Auto Increment)
- driver_id (FK → users.id)
- vehicle_id (FK → vehicles.id)
- origin
- destination
- date
- time
- price_per_seat
- seats_available
- status (active/completed/cancelled)

bookings.tbl

- id (PK, Auto Increment)
- ride_id (FK → rides.id)
- rider_id (FK → users.id)
- seats
- status (pending/confirmed/cancelled)
- amount

feedbacks.tbl

- id (PK, Auto Increment)
- ride_id (FK → rides.id)
- rider_id (FK → users.id)
- rating (1–5)
- comment

contact_us.tbl

- id (PK, Auto Increment)
- name
- email
- message

admin.tbl

- id (PK, Auto Increment)
- name
- email
- password

Step 6: Define Each Table Column / Data Dictionary

Database & table create / website designing

```
CREATE DATABASE carpooling;

CREATE TABLE users( id INT PRIMARY KEY AUTO_INCREMENT,role
ENUM('rider','driver','admin') DEFAULT 'rider', name VARCHAR(255), email
VARCHAR(255) UNIQUE, phone VARCHAR(20), password VARCHAR(255), status
TINYINT DEFAULT 1);

CREATE TABLE admin(  id INT PRIMARY KEY AUTO_INCREMENT,  name
VARCHAR(255),  email VARCHAR(255),  password VARCHAR(255));

CREATE TABLE vehicles(  id INT PRIMARY KEY AUTO_INCREMENT,  user_id INT,
make VARCHAR(255), model VARCHAR(255), plate VARCHAR(50),seats INT,
FOREIGN KEY(user_id) REFERENCES users(id));

CREATE TABLE rides(id INT PRIMARY KEY AUTO_INCREMENT,driver_id INT,
vehicle_id INT, origin VARCHAR(255), destination VARCHAR(255), date DATE,
time TIME, price_per_seat INT, seats_available INT, status
ENUM('active','completed','cancelled') DEFAULT 'active', FOREIGN KEY(driver_id)
REFERENCES users(id), FOREIGN KEY(vehicle_id) REFERENCES vehicles(id));

CREATE TABLE bookings(id INT PRIMARY KEY AUTO_INCREMENT,ride_id INT,
rider_id INT,seats INT,amount INT,status ENUM('pending','confirmed','cancelled')
DEFAULT 'pending',FOREIGN KEY(ride_id) REFERENCES rides(id),FOREIGN
KEY(rider_id) REFERENCES users(id));

CREATE TABLE feedbacks(id INT PRIMARY KEY AUTO_INCREMENT,ride_id INT,
rider_id INT, rating INT, comment VARCHAR(255), FOREIGN KEY(ride_id)
REFERENCES rides(id),FOREIGN KEY(rider_id) REFERENCES users(id));

CREATE TABLE contact_us(id INT PRIMARY KEY AUTO_INCREMENT,name
VARCHAR(255), email VARCHAR(255),message VARCHAR(500));
```