

## UE21CS351A: Database Management System

### PROJECT REPORT

#### Car Rental Management System

Shriansh - PES1UG21CS584, Shubha - PES1UG21CS589

#### Table of Contents:

##### Contents

Introduction: .....	1
Purpose of the project: .....	1
Scope of the Project: .....	2
Project Description .....	3
Project Overview: .....	3
Major Project Functionalities: .....	3
E-R Diagram: .....	4
Relational Schema: .....	5
Front-End and Output: .....	6

#### Introduction:

##### Purpose of the project:

The primary purpose of the Car Rental System is to provide a seamless and user-friendly interface to facilitate customers in reserving cars for their needs. With the increasing demand for self-driven car experiences and the convenience of not owning a car in urban environments, a robust car rental system is crucial. This system aims to simplify the process of car rentals, track availability of cars, calculate costs based on selected categories, monitor bookings, and handle the billing process effectively. The project

focuses on ensuring the storage, retrieval, modification, and deletion of data related to car rentals is smooth, accurate, and efficient.

## Scope of the Project:

The Car Rental System is envisioned to cater to a broad range of users, both members and non-members. The scope includes:

- a) **Customer Management:** Registration of new members, storage of crucial customer details like driving license, contact details, and membership ID. This also caters to non-members, ensuring they can avail the services with ease.
- b) **Car Inventory Management:** Track all cars in the system, their current status (available or booked), and manage the details of each car, including its make, model, mileage, and registration number.
- c) **Car Category Management:** Define and manage different car categories, setting the number of persons it can accommodate, luggage capacity, pricing details, and late fee structures.
- d) **Location Management:** Manage different pickup and drop-off locations, providing customers the flexibility to choose based on their convenience.
- e) **Booking and Reservation Management:** Handle all bookings, track their current status, duration of booking, and any additional details like rental insurance or discount codes.
- f) **Billing and Payment Management:** Generate bills post car return, detailing all charges, taxes, and any late fees incurred.

The system is designed keeping in mind the ease of use for customers and effective management for administrators. It aims to revolutionize the car rental experience by integrating technology effectively.

# Project Description

## Project Overview:

The Car Rental System is an integrated platform designed to centralize and streamline the process of renting cars to customers. In the modern urban landscape where the costs of car ownership are escalating and the convenience of cab services is evident, there's a growing demand for self-drive rental options. Catering to this demand, our system provides customers with an intuitive interface to browse available cars, select their desired models based on their requirements, and complete the reservation process. Simultaneously, the system aids admins in managing the cars, tracking bookings, and ensuring efficient billing.

## Major Project Functionalities:

### 1. User Registration & Authentication:

- Allows new users to register either as a member or a non-member.
- Secure login and authentication for returning users.
- Profile management with options to update personal details and view booking history.
- Entities Involved: Customer
- Inputs Required: Name, Email, Password, Driving License Number, Phone Number, Address.

### 2. Car Browsing & Selection:

- Display available cars with necessary details like make, model, mileage, and registration number.
- Filter options based on car categories, locations, or other specifications.
- Check car availability in real-time.

### 3. Booking & Reservation:

- Select pickup and drop off locations.
- Choose the duration for the rental.
- Add on options like rental insurance or apply discount codes.
- Confirmation of booking with a unique booking ID for tracking.

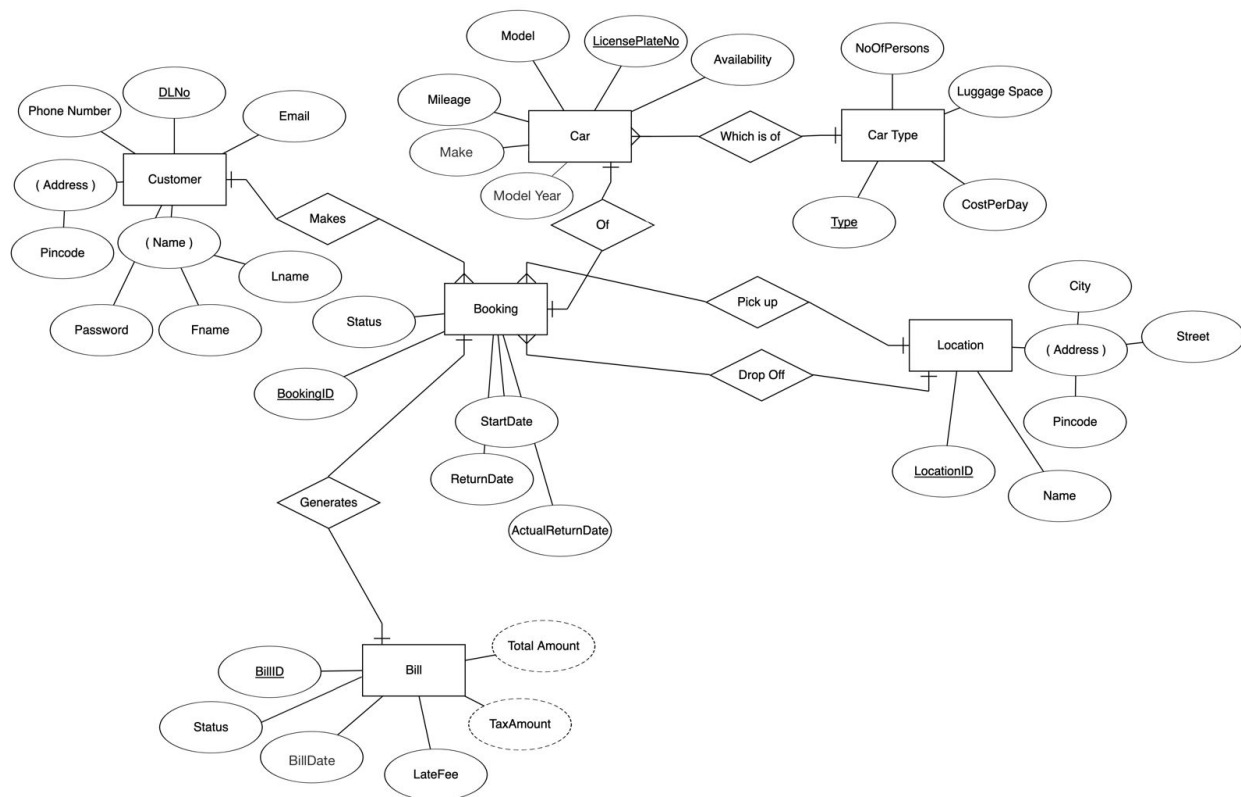
#### 4. Dynamic Pricing & Billing:

- Pricing calculations based on selected car category, rental duration, and any additional features.
- Realtime billing estimate.
- Generate detailed bills post car return with all charges, taxes, and late fees, if applicable.

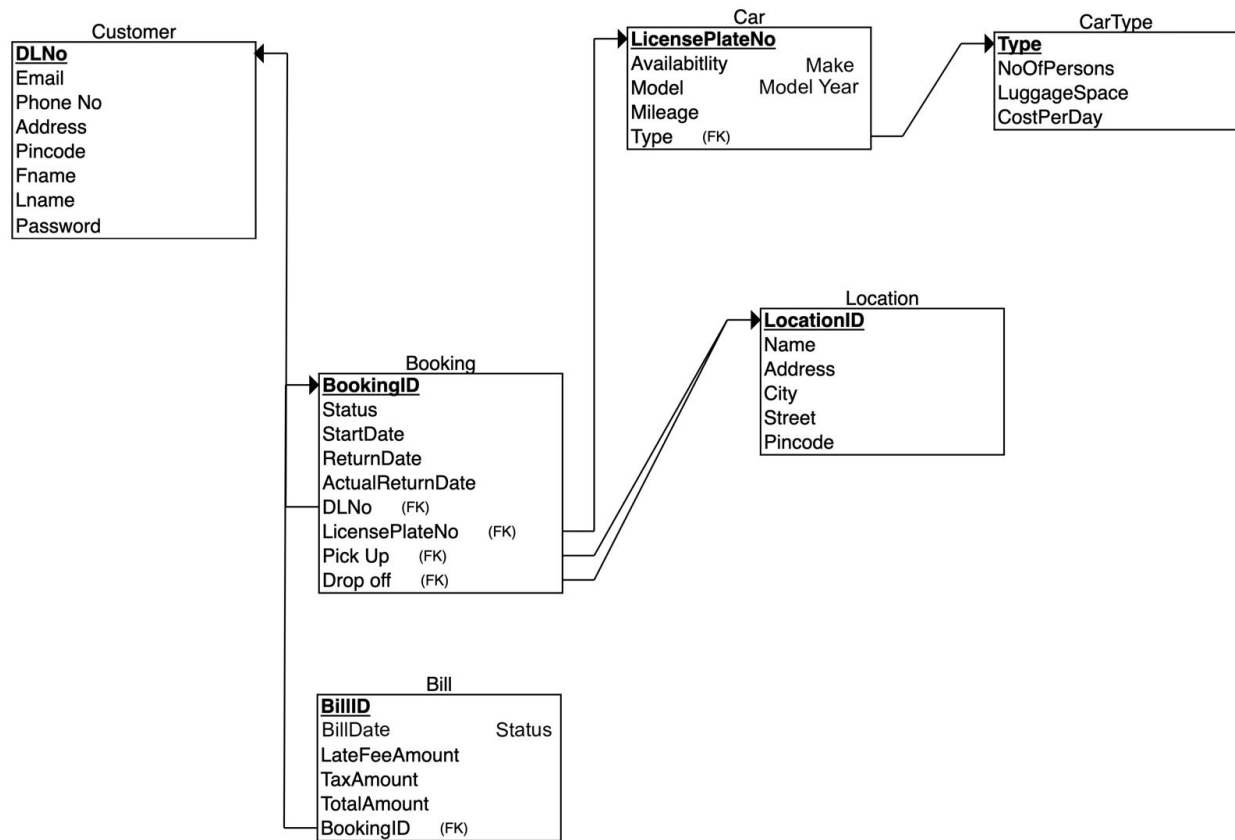
#### 5. Location Management:

- Display available pickup and drop off locations.
- Option for users to suggest or request new locations.

### E-R Diagram:



# Relational Schema:



Front-end and Output:

**Rent a car** is within  
your finger tips.

Register

Login

**Rent a car** with us.

Login

Username

shriansh

Password

...

Log in

**Rent a car** is within  
your finger tips.

Car Type:

ANY

Pick-up Location:

Pick up

Search Cars

Confused about what car to get? [Click here for more info](#)

Car Type:

STANDARD

Pick-up Location:

Dallas

Search Cars

Confused about what car to get? [Click here for more info](#)

## Car Category Information

Category Name	No. of Luggage	No. of Persons	Cost per Day	Late Fee per Day
COMPACT	3	5	Rs. 32.0	Rs. 0.96
ECONOMY	2	5	Rs. 30.0	Rs. 0.9
FULL SIZE	4	5	Rs. 40.0	Rs. 1.2
FULL SIZE SUV	2	8	Rs. 60.0	Rs. 1.8
LUXURY CAR	5	5	Rs. 75.0	Rs. 2.25
MID SIZE	3	5	Rs. 35.0	Rs. 1.05

## Available Cars

Registration Number	Model	Company	Year	Mileage	Car Type	Location	
VBN6283	TAURUS	FORD	2015	17500	STANDARD	DALLAS LOVE FIELD AIRPORT	<a href="#">Book</a>



## Booking Details for Car VBN6283

Registration Number:	VBN6283
Model:	TAURUS
Company:	FORD
Year:	2015
Mileage:	17500
Car Type:	STANDARD
Luggage Space:	3
Seating Capacity:	5
Pick up location:	DALLAS LOVE FIELD AIRPORT

Address:	Herb Kelleher Way Dallas Texas 75235
Cost Per Day:	38.0
Late Fee Per Day:	1.14

Start Date:

End Date:

01/05/2018



01/06/2018



Find Cost

<b>Address:</b>	Herb Kelleher Way Dallas Texas 75235
<b>Cost Per Day:</b>	38.0
<b>Late Fee Per Day:</b>	1.14

Start Date:

End Date:



From 2018-05-01 to 2018-06-01

<b>Total Cost:</b>	1178.0
--------------------	--------

Find Cost

Book

## Booking Confirmed

**Booking ID** B1024  
**Car Type and Model** STANDARD, FORD  
**From** 2018-05-01 00:00:00  
**To** 2018-06-01 00:00:00  
**Amount** 1178.0  
**Late fee per day** 1.14  
**Drop Location** DALLAS LOVE FIELD AIRPORT, Herb Kelleher Way Dallas Texas 75235  
**Pickup Location** ('D', 'H')  
**Registration Number** VBN6283

## Booking Details

Your bookings

Booking ID	Car Model	Registration Number	Amount	Booked Start Date	Booked Return Date	Pickup Location	Drop Location	Actual return date (if any)	Action
B1004	ESCAPE	WLZ8955	48.0	2023-10-24	2023-11-25	NEWARK LIBERTY INTL AIRPORT	NEWARK LIBERTY INTL AIRPORT	<input type="text" value="dd/mm/yyyy"/>	<a href="#">Generate Bill</a>
B1024	TAURUS	VBN6283	1178.0	2018-05-01	2018-06-01	DALLAS LOVE FIELD AIRPORT	DALLAS LOVE FIELD AIRPORT	<input type="text" value="dd/mm/yyyy"/>	<a href="#">Generate Bill</a>

## Payment Info

Bill ID	Booking ID	Cost	Booked Return	Actual Return - if any	Late Fee	Tax	Final Amount	Pay Now
BL018	B1024	1178.0	2018-06-01	2023-12-01	0.0	212.04	1390.04	<a href="#">Pay Now</a>

## Booking Details

Your bookings

Bill ID	Cost	Late Fee	Tax	Total Amount	Date (of return)	Status	Action
BL017	48.0	0.0	8.64	56.64	2023-11-25	Unpaid	<a href="#">Pay Now</a>
BL016	280.0	0.0	50.4	330.4	2023-12-20	Unpaid	<a href="#">Pay Now</a>
BL018	1178.0	0.0	212.04	1390.04	2018-06-01	Unpaid	<a href="#">Pay Now</a>

2018

Bill ID	Cost	Late Fee	Tax	Total Amount	Date (of return)	Status	Action
BL018	1178.0	0.0	212.04	1390.04	2018-06-01	Unpaid	<a href="#">Pay Now</a>

Bill ID	Cost	Late Fee	Tax	Total Amount	Date (of return)	Status	Action
BL017	48.0	0.0	8.64	56.64	2023-11-25	Paid	Already Paid!
BL016	280.0	0.0	50.4	330.4	2023-12-20	Unpaid	<a href="#">Pay Now</a>
BL018	1178.0	0.0	212.04	1390.04	2018-06-01	Unpaid	<a href="#">Pay Now</a>

```
if request.method == 'POST':
    # collect car type, pick up location, drop off location, pick up date, drop
    car_type = request.form['car_type']
    pick_up = request.form ['pick_up']

    cars = find_available_cars(car_type, pick_up)

def find_available_cars(car_type, pick_up):
    if pick_up and car_type:
        cursor.execute('SELECT location_id FROM location_details WHERE city = %s', (pick_up,))
        pick_up = cursor.fetchall()

        cars = []

        for centre in pick_up:
            loc = centre[0]
            # get all cars that satisfy the constraints
            cursor.execute('SELECT * FROM car WHERE category_name = %s AND location_id = %s AND availability_flag="Y"', (car_type, loc))
            cars.extend(cursor.fetchall())
    elif pick_up:
        cursor.execute('SELECT location_id FROM location_details WHERE city = %s', (pick_up,))
        pick_up = cursor.fetchall()

        cars = []

        for centre in pick_up:
            loc = centre[0]
            # get all cars that satisfy the constraints
            cursor.execute('SELECT * FROM car WHERE location_id = %s AND availability_flag="Y"', (loc,))
            cars.extend(cursor.fetchall())
    elif car_type:
        cursor.execute('SELECT * FROM car WHERE category_name = %s AND availability_flag="Y"', (car_type,))
        cars = cursor.fetchall()
    else:
        cursor.execute('SELECT * FROM car WHERE availability_flag="Y"')
        cars = cursor.fetchall()

    # replace the location_id with the location name
    locations = []
    for i in cars:
        cursor.execute('SELECT * FROM location_details WHERE location_id = %s', (i[-2],))
        locations.append(cursor.fetchone()[1])
```

```
def bill_year():
    year = request.form["year"]
    if year == '':
        return redirect(url_for('bills'))
    data = show_bills(email_id=session.get('email_id'), year=year)
    return render_template('bill_year.html', data=data, year=year)

def show_bills(email_id, year=None):
    if year:
        query = """
        SELECT * FROM billing_details bill
        JOIN booking_details book ON bill.booking_id = book.booking_id
        JOIN customer_details cust ON book.DL_NUMBER = cust.DL_NUMBER
        WHERE cust.email_id = %s AND YEAR(book.RET_DT_TIME) = %s"""
        cursor.execute(query, (email_id, year))
    else:
        query = """
        SELECT * FROM billing_details bill
        JOIN booking_details book ON bill.booking_id = book.booking_id
        JOIN customer_details cust ON book.DL_NUMBER = cust.DL_NUMBER
        WHERE cust.email_id = %s"""
        cursor.execute(query, (email_id,))
    data = cursor.fetchall()
    return data
```

```

> Execute
DELIMITER //

> Execute
CREATE PROCEDURE CALCULATE_LATE_FEE_AND_TAX(
  IN actualReturnDate DATE,
  IN bookingId CHAR(5) -- Adjust the VARCHAR length according to your schema
)
BEGIN
  DECLARE lateFeePerDay DECIMAL(10,2);
  DECLARE DayDifference DECIMAL(10,2);
  DECLARE totalLateFee DECIMAL(10,2);
  DECLARE totalTax DECIMAL(10,2);
  DECLARE regNo CHAR(7);
  DECLARE amt FLOAT(10,2);
  DECLARE ReturnDate DATE;

  SELECT REGISTRATION_NUMBER FROM BOOKING_DETAILS WHERE BOOKING_ID = bookingId INTO regNo;
  SELECT RET_DT_TIME FROM BOOKING_DETAILS WHERE BOOKING_ID = bookingId INTO ReturnDate;
  -- UPDATE BOOKING_DETAILS SET ACT_RET_DT_TIME = actualReturnDate WHERE BOOKING_ID = bookingId;

  SELECT CC.LATE_FEE_PER_DAY INTO lateFeePerDay
  FROM CAR_CATEGORY CC
  INNER JOIN CAR C ON CC.CATEGORY_NAME = C.CATEGORY_NAME
  WHERE C.REGISTRATION_NUMBER = regNo;

  SELECT AMOUNT FROM BOOKING_DETAILS WHERE BOOKING_ID = bookingId INTO amt;

  IF actualReturnDate > ReturnDate THEN
    SET dayDifference = DATEDIFF(actualReturnDate, ReturnDate);
    SET totalLateFee = dayDifference * lateFeePerDay;
  ELSE
    SET totalLateFee = 0;
  END IF;

  SET totalTax = (amt + totalLateFee) * 0.18;

  UPDATE BOOKING_DETAILS SET LATE_FEE = totalLateFee, TAX_AMOUNT = totalTax,
  ACT_RET_DT_TIME = actualReturnDate, BOOKING_STATUS = 'R' WHERE BOOKING_ID = bookingId;

END //

```

```
DELIMITER //
```

▷ Execute

```
CREATE TRIGGER GENERATE BILLING
AFTER UPDATE ON BOOKING_DETAILS
FOR EACH ROW
BEGIN
    -- DECLARE next_id VARCHAR(255);
    DECLARE newBillId VARCHAR(255); -- Adjust the size as per your column definition
    DECLARE totalLateFee DECIMAL(10,2);
    DECLARE totalTax DECIMAL(10,2);
    DECLARE totalAmount DECIMAL(10,2);

    -- SELECT CONCAT('BL', LPAD(IFNULL(CONVERT(SUBSTRING(MAX(BILL_ID), 3), UNSIGNED INTEGER), 0) + 1, 3, '0')) INTO newBillId
    -- FROM BILLING_DETAILS;

    -- Calculate the next lexicographically available ID for table2
    SELECT CONCAT('BL', LPAD(CAST(SUBSTRING(MAX(BILL_ID), 3) + 1 AS UNSIGNED), 3, '0'))
    INTO newBillId
    FROM BILLING_DETAILS;

    SET totalAmount = NEW.AMOUNT + NEW.LATE_FEE + NEW.TAX_AMOUNT;

    -- If no rows are present in table2, set the initial value
    IF newBillId IS NULL THEN
        SET newBillId = 'BL001';
    END IF;

    -- Insert a new row into table2 with the calculated ID
    -- INSERT INTO table2 (id, other_column)
    -- VALUES (next_id, NEW.other_column); -- Adjust 'other_column' based on your schema

    INSERT INTO BILLING_DETAILS (BILL_ID, BILL_DATE, BILL_STATUS, TOTAL_AMOUNT, TAX_AMOUNT, BOOKING_ID, LATE_FEE)
    VALUES (newBillId, CURDATE(), 'D', totalAmount, NEW.TAX_AMOUNT, NEW.BOOKING_ID, NEW.LATE_FEE);

END;
//
```