IT2080 – ITP
Year 2, Semester 2, 2024
Assignment 2

Easy Rent Vehicle Rental And Hiring Management System

Group Details

Batch: 4.1

Group Number: ITP24_B1_04

	Student Registration Number	Student Name
1	IT22580344	Jayasekara M.P.S.S
2	IT22286246	Chandrasekara C.M.P.V
3	IT22137128	Wijewantha H.D.C
4	IT22230638	Nathasha K.D.S
5	IT22101242	Tharuka M.S
6	IT22124876	Silva M.W.P.R.K
7	IT22271150	Abeysekara.W.R.G.M
8	IT22281128	Jayasundara J.M.T.P

Contents

Progress

Function	Completed User Stories	Incomplete User Stories
Customer Management System **IT22137128**	- As a new user, I want to register easily on the platform. - As a renter, I want to view vehicle ads within the system. - As a renter, I want to search for vehicles. - As a renter, I want to report vehicles if necessary. - As a renter, I want to add vehicles to my favourite list and manage it. - As a user, I want to provide feedback after service. - As a user, I want to submit complaints at any time	As a user, I want the option to delete my account if needed.
Booking and Payment Management System **IT22580344**	- As a Renter, I want to create, edit, delete booking requests for selected vehicle. - As a Renter, I want to see my booking history. - As a Renter, I want to see estimate price for my trip - As a Renter, I want to search my booking history. - As a Vehicle Owner, I want to check booking request for all my vehicles. - As a Vehicle Owner, I want to search booking requests. - As a Vehicle Owner, I want to approve or reject booking requests. - As a Vehicle Owner, I want to provide a rejected message for renter when rejecting booking request. - As a Renter, I can check my booking request approved or rejected.	- As a renter, I want to do a payment - As a renter, I want to see my payment history - As a Owner, I want to add additional cost to estimated bill - As a Owner, I want to check my earning History

Vehicle Management System **IT22230638**	• As the vehicle manager, I want to add vehicles to the system approved by the maintenance manager. • As the vehicle manager, I want to add vehicles to the system updated by the maintenance manager. • As the vehicle manager, I want to delete, update vehicles added to the system as required. • As the vehicle manager, I want to generate the rental reports.	• As the vehicle manager, I want to investigate the complaints made by the renters regarding the vehicles or the system and give answers to those complaints.
Driver Management System **IT22124876**	• As a driver, I can register with the system. • As a driver, I can watch the registered drivers. Also, can add new drivers. • As a driver, I'm able to get the booking details when an owner assigns a driver for a hire. • As a driver manager, I would like to know the customer feedback and if there are any mistakes or faults of drivers, I may ask them to correct the mistakes.	• As a driver, I need to get my monthly report which includes about the dates and locations I have visited and the salary. • As a driver, I want send a reject message if I couldn't be able to attend the assigned hire.
Vehicle Owner Management System **IT22101242**	• As an unregistered vehicle owner, I want to create a profile and register in the system • As a vehicle owner, I want to be able to update and delete profile information • As a vehicle owner, I want to add my vehicles details • As a vehicle owner, I want to update and delete my vehicle details as needed	• As a vehicle owner, I want to be able to add driver details and update them • As a vehicle owner, I want to receive notifications about bookings and other updates • As a vehicle owner, I want to check earnings

| HR Management System **IT22286246** | • As a HR manager, I want to be able to log in to the system.
• As a HR manager, I want to add employees so I can access to the system.
• As a HR manager, I want to add new employees, so I can fill the form and add employees.
• As a HR manager, I want to update and delete employees, so I can update and delete employees using employees' details table.
• As a HR manager, I want to review employee leaves, so I can access to the employee leave page.
• As a HR manager, I want to allocate salary for employees, So I can generate salary using calculate salary page.
• As an employee, I want to request a leave, so I can fill the leave form and request a leave.
• As an employee, I want to request a leave, so I can fill the leave form and request a leave.
• As an employee, I want to add daily attendance, so I can fill the attendance form. | • As a HR manager, I want to accept or reject leave request.
• As a HR manager, I want to generate report |

Repository Link Of the Project

https://github.com/Sudila-Jayasekara/EasyRent

ER Diagram
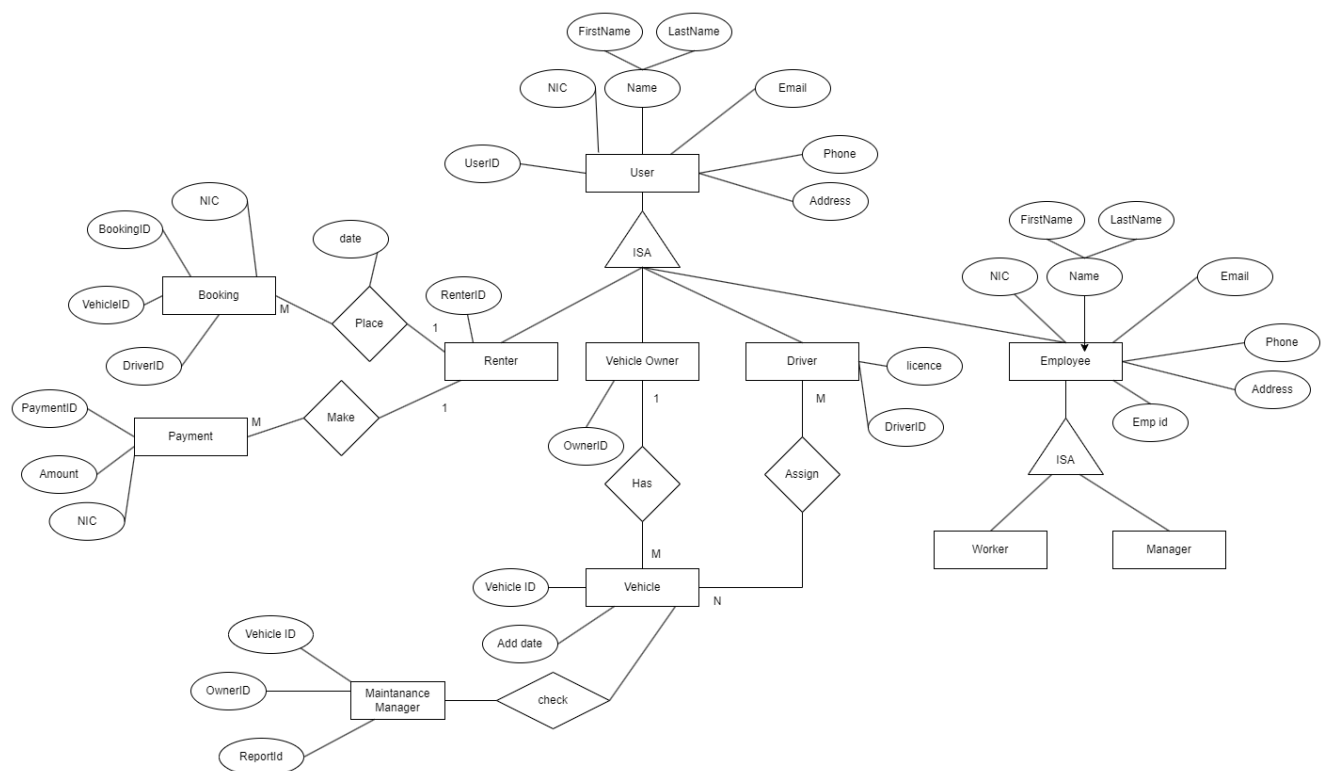
High-level System Design
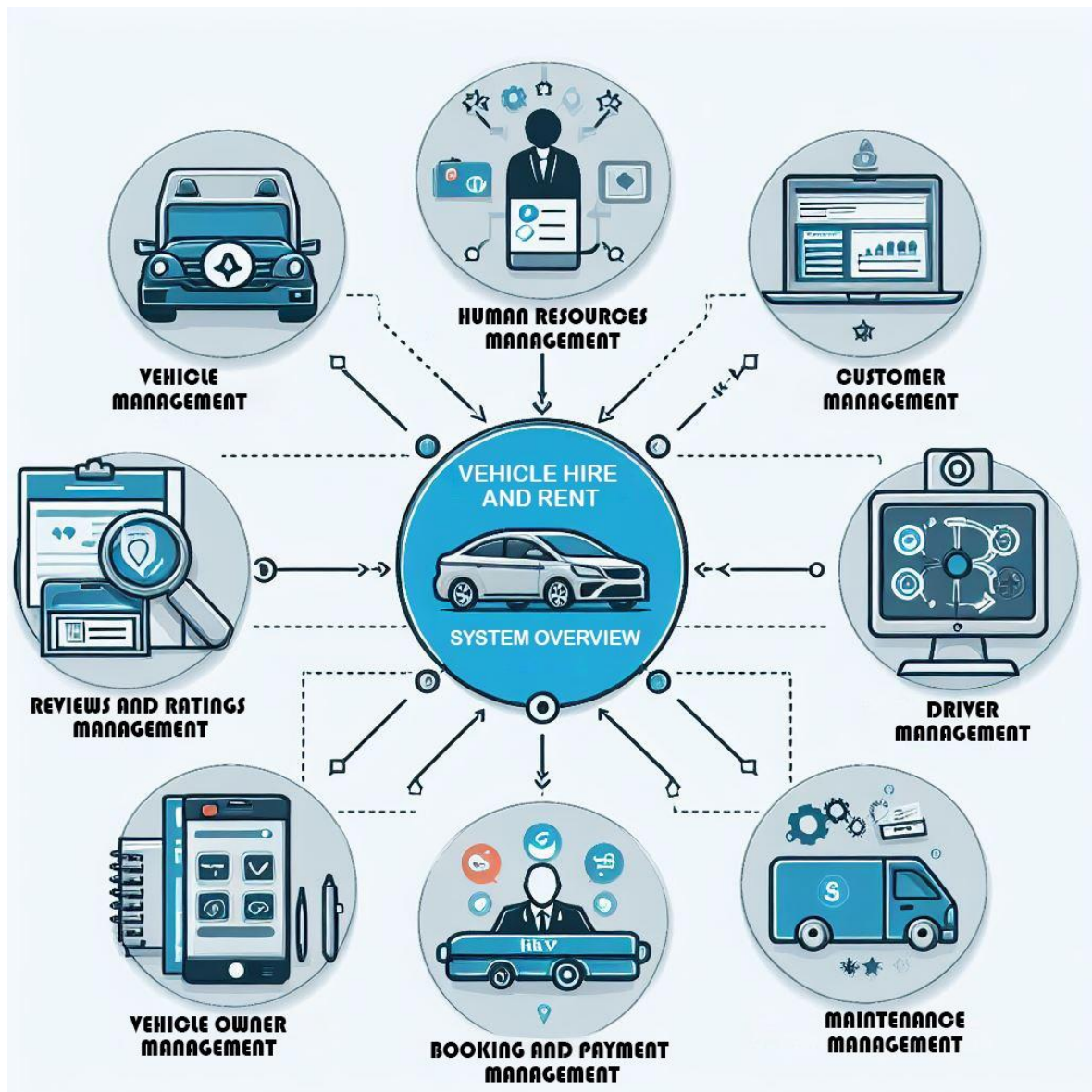
Normalized Schema
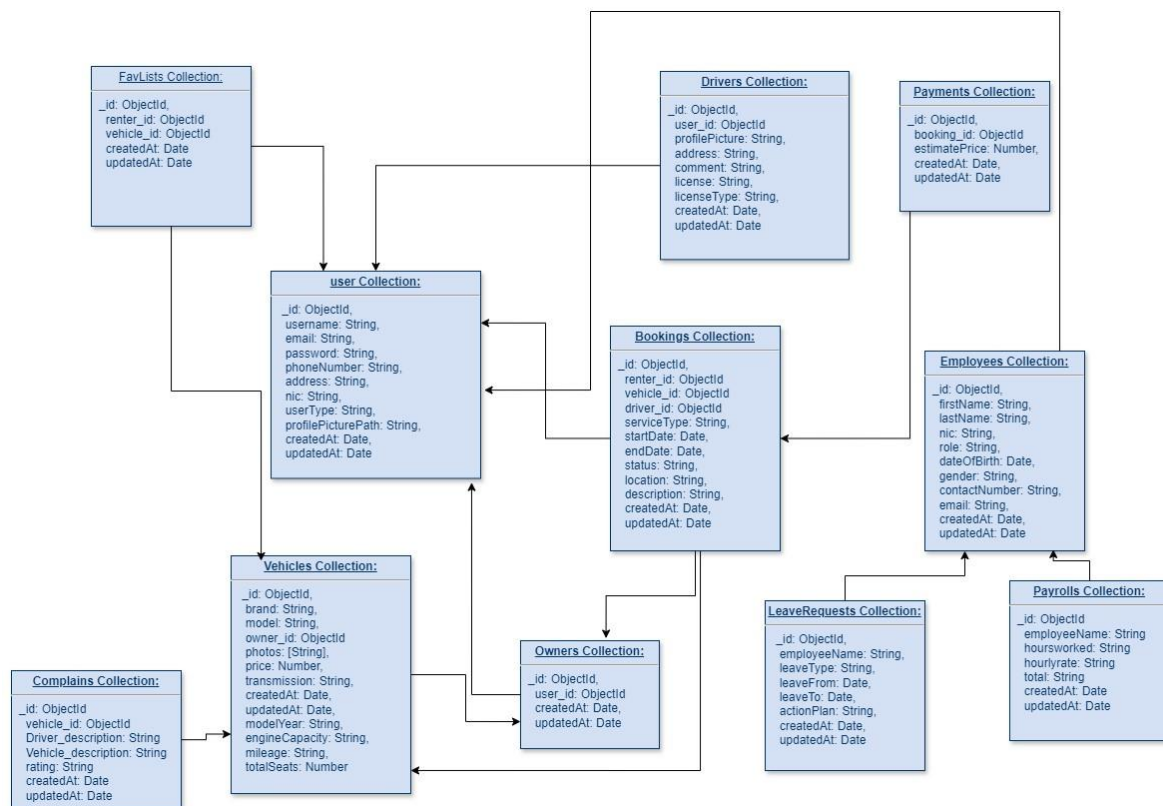
Test Case Design

Test Function: Report Generation
Test Case Designed By: It22137128
Test Priority: High
Pre-condition: Renter must be logged into the system.
Test Steps:
- Renter navigates to the "Report Generation" section.
- Selects the desired parameters for the report, such as rental duration, vehicle type, location, etc.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	rental duration, vehicle type, location, etc	A detailed report should be generated, including information such as rental history,	The report is generated successfully and displays relevant data	pass	The renter successfully generates a report based on selected parameters, providing

		expenses, vehicle usage statistics, etc.			valuable insights into their rental activities

Test Function: Search Vehicles Ads and Visibility
Test Case Designed By: It22137128
Test Priority: High
Pre-condition: Renter must be logged into the system.
Test Steps:
- Renter accesses the "Search Vehicles Ads" feature.
- Enters search criteria such as vehicle type, location, availability dates, etc.
- Initiates the search.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	vehicle type, location, availability dates, etc	The system should display relevant vehicle ads that match the search criteria. The ads should include detailed information about the vehicles and their availability	vehicle ads matching the search criteria are displayed, with accurate information regarding availability and features	pass	The renter successfully searches for vehicle ads and finds the desired vehicles with accurate visibility and availability information

Test Function: Search Booking history
Test Case Designed By: IT22580344
Test Priority: High
Pre-condition: Renter must be logged into the system.
Test Steps:
- Vehicle Renter navigates to the "Booking" section.
- Enters search criteria such as vehicle type, location, availability dates, etc.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Vehicle brand, Vehicle model,	The system should only display the relevant	System display only relevant bookings the	pass	The renter successfully searches for bookings

	location, dates	bookings that matching to search criteria	matching to search criteria		

Test Function: Generate Payment Report
Test Case Designed By: IT22580344
Test Priority: High
Pre-condition: Owner must be logged into the system.
Test Steps:
- Vehicle Owner navigates to the "Earning" section.
- Select a week, month or year for generate reports

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Select week, month or year for generate report	System should generate a report for all earning for entered period of time	System displays generated report for owner earnings for selected time period	pass	The owner successfully generated report for earning

Test Function: Adding Approved Vehicles to the system
Test Case Designed By: IT22230638
Test Priority: High
Pre-condition: Vehicle Manager must be logged into the system.
Test Steps:
- Vehicle Manager navigates to the "Approved Vehicles" section in the system.
- Check the approved vehicle forms.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Approved vehicles' forms	Approved vehicles successfully added to the system for rental.	Approved vehicles successfully added to the system for rental.	pass	The vehicle manager successfully add approved vehicle to the system after receiving approval form the maintenance manager.

Test Function: Investigating Renter Complaints
Test Case Designed By: IT22230638

Test Priority: High
Pre-condition: Vehicle Manager must be logged into the system.
Test Steps:
- Vehicle Manager navigates to the "Reviews and Ratings" section in the system.
- Access the list of Reviews submitted by renters.
- Reviews and ratings for vehicles and the system are inspected thoroughly.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Reviews and ratings provided by renters	Effectively investigate and respond appropriately to renter complaints by thoroughly analysing reviews and ratings provided by renters.	Responding appropriately to address any identified issues or concerns.	pass	The vehicle manager effectively investigating renter complaints and responding appropriately to address any identified issues or concerns.

Test Function: Report Generation
Test Case Designed By: IT22271150
Test Priority: High
Pre-condition: Renter must be logged into the system.
Test Steps:
- Renter navigates to the "Reviews and rating page" section.
- Select the form to add reviews and ratings.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Vehicle id. Rating through stars. Review for vehicle. Review for Driver.	A detailed report should be generated, including information such as Reviews for driver, vehicle separately and rating through	The report is generated successfully and displays relevant data	pass	The renter Can add reviews effectively and other renters can see all the reviews and rating in booking page.

		starts to trip.			

Test Function: Response for driver assigning for a hire.
Test Case Designed By: IT22124876
Test Priority: high
Pre-condition: Owner should assign a driver for the hire.
Test Steps:

- Driver navigates to the "View bookings" page.
- Driver checks for the dates and the locations where the drivers have been assigned.
- Response whether he can participate to the hire or not.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Dates that the renters have done a booking.	Owners checked the dates and assign a suitable driver for the hire.	Assigning is accurately displays at the relevant driver's profile.	pass	Renters can make a booking for a vehicle. So the vehicle owners can check the dates and assign a relevant driver to the booking. Drivers can watch the dates they have been assigned and response whether they can attend or not.

Test Function: Adding vehicles details to the system
Test Case Designed By: IT22101242
Test Priority: High
Pre-condition: User has necessary permissions to add vehicle details
Test Steps:

- Enter valid vehicle details into the respective fields
- Check for a confirmation message indicating successful addition of vehicle details

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Vehicle details: brand, model, model year, engine capacity, total seats, vehicle id, start date, transmission, price, mileage	The owner adds the vehicle details and confirms successful addition.	Vehicle details are added, and the system displays a success message.	pass	Verify that the owner can successfully add vehicle details

Test Function: Adding driver details to the system
Test Case Designed By: IT22101242
Test Priority: High
Pre-condition: The vehicle owner has necessary permissions to add driver details
Test Steps:

- Enter valid driver details in the form
- Check if the added driver details are visible in the system's database

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Form including driver's name, email, contact no, NIC and other details	Driver details are successfully added to the system	Driver details added successfully	pass	Verify that the system can add driver details with valid inputs

Test Function: Add new employees to the system.
Test Case Designed By: IT22286246
Test Priority: High
Pre-condition: HR manager must be logged into the system.

Test Steps:
- Enter employee details by filling form.
- Click the register button.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	First name, Last name, NIC, Role, Date of birth, select gender, Contact number, Email	Create new employee in database. Display registration successful message.	Add new employee to database and display "Employee data send successfully" message.	pass	Add a new employee to the system.

Test Function: Leave request management.
Test Case Designed By: IT22286246
Test Priority: High
Pre-condition: Employee must be registered into system.
Test Steps:
- Employees navigate to "Leave Management" section in the system.
- Fill the leave request form.
- Click "Submit Request" button.

Test ID	Test Inputs	Expected Output	Actual Output	Result	Description
	Employee name, leave type, leave from, Leave to, Action plan	Apply leave request. Display request successful message.	Add new request in database.	pass	Add a new request in the database and HR manager can review those requests.

Individual Contribution

Renter Management System
IT22137128

Completed Features	Incomplete Features
- Develop a user registration form with necessary fields. - Implement validation for email and password. - Create a delete account functionality with a confirmation prompt. - Design and implement a vehicle ad display page. - Develop search functionality with filters for vehicle type, location, etc. - Develop functionality to add/remove vehicles from the favourites list	- Manage User Profile - Implement saving of search results for easy access

Pseudocode for Favourite List Component

```
Start:
    Fetch Watchlist Data
    If Watchlist Data Exists:
        Fetch Vehicle Details
        Display Vehicle Cards
        Ask User if They Like a Vehicle
        If User Likes a Vehicle:
            Remove Vehicle from Watchlist
        Else:
            Add Vehicle to Watchlist
        Update Watchlist State
    Else:
        Display No Data Message
    Repeat from Fetch Watchlist Data
```
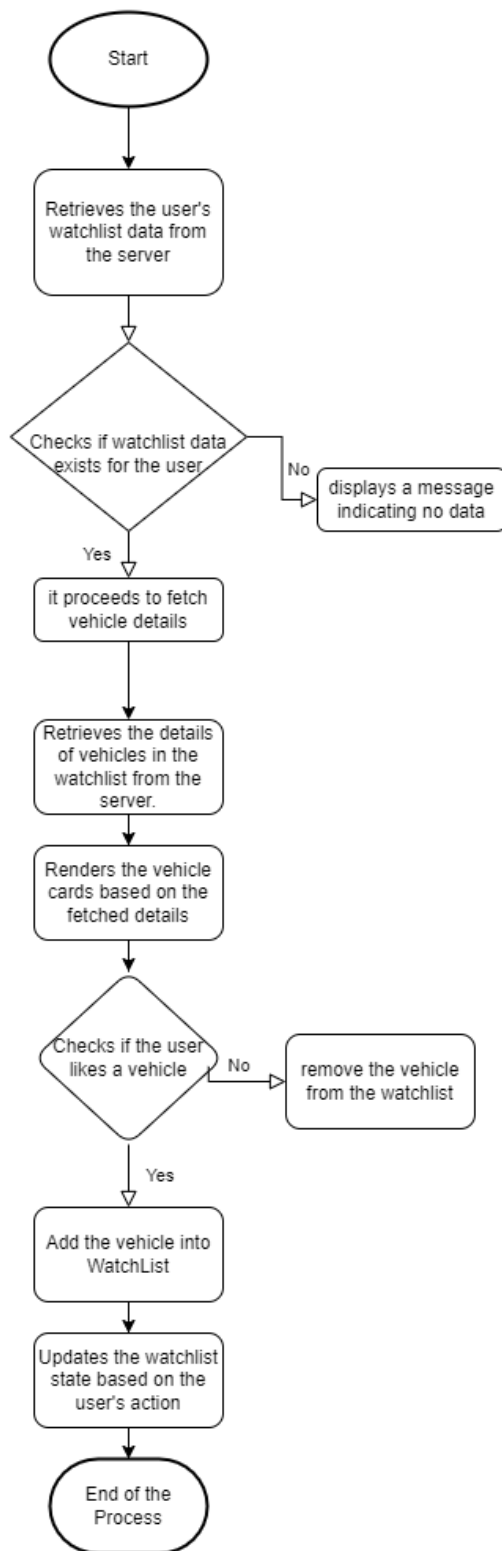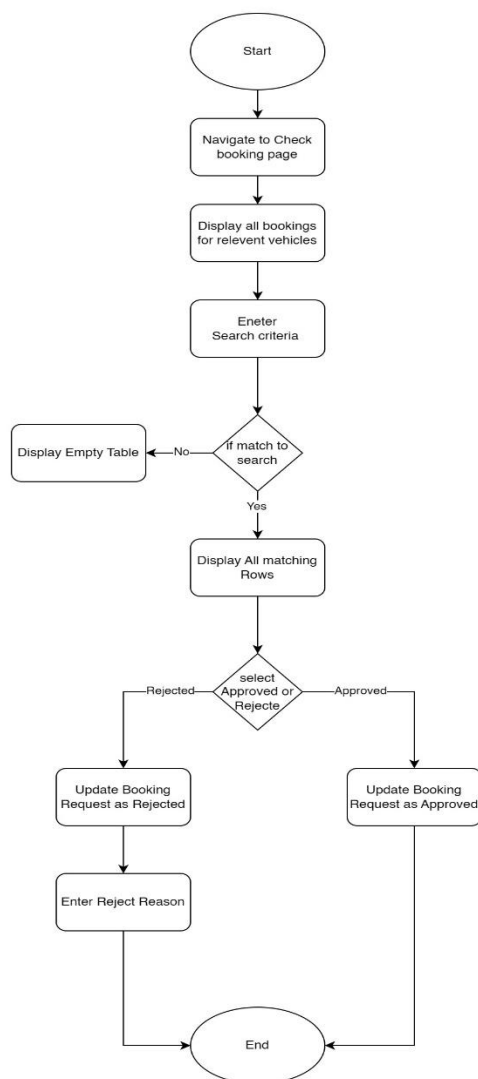
Flow-chart

```
                    ┌─────────┐
                    │  Start  │
                    └─────────┘
                         │
                         ▼
              ┌────────────────────┐
              │ Retrieves the user's│
              │ watchlist data from │
              │    the server       │
              └────────────────────┘
                         │
                         ▼
                    ◇─────────◇
              Checks if watchlist data ──No──► ┌──────────────────┐
                exists for the user            │ displays a message│
                    ◇─────────◇                │ indicating no data│
                         │                     └──────────────────┘
                        Yes
                         ▼
              ┌────────────────────┐
              │ it proceeds to fetch│
              │   vehicle details   │
              └────────────────────┘
                         │
                         ▼
              ┌────────────────────┐
              │ Retrieves the details│
              │  of vehicles in the │
              │  watchlist from the │
              │       server.       │
              └────────────────────┘
                         │
                         ▼
              ┌────────────────────┐
              │ Renders the vehicle │
              │ cards based on the  │
              │  fetched details    │
              └────────────────────┘
                         │
                         ▼
                    ◇─────────◇
              Checks if the user ──No──► ┌──────────────────┐
               likes a vehicle           │ remove the vehicle│
                    ◇─────────◇          │ from the watchlist│
                         │               └──────────────────┘
                        Yes
                         ▼
              ┌────────────────────┐
              │ Add the vehicle into│
              │     WatchList       │
              └────────────────────┘
                         │
                         ▼
              ┌────────────────────┐
              │ Updates the watchlist│
              │ state based on the  │
              │   user's action     │
              └────────────────────┘
                         │
                         ▼
                    ┌─────────┐
                    │ End of the│
                    │  Process  │
                    └─────────┘
```

Booking and Payment Management System
IT22580344

Completed Features	Incomplete Features
• Develop Booking form with necessary fields for create bookings. • Develop Booking history page for see all relevant bookings for renter. • Add update and delete function for booking. • Develop Check booking page that Owner can sell all relevant bookings for their vehicles. • Develop functionality to approve or reject bookings with a message. • Generate Estimate bill for bookings	• Generate total bill with add additional cost to estimated bill • Implement payment history page. • Implement Earning page.

Flow-chart

Vehicle Management System
IT22230638

Completed Features	Incomplete Features
• Develop the vehicle additions to the system as approved by the maintenance manager. • Develop the vehicle additions to the system updated by the maintenance manager. • Develop delete and update vehicles added to the system as required. • Generate rental reports.	• Investigate complaints made by renters regarding vehicles or the system and provide answers.

Flow-chart
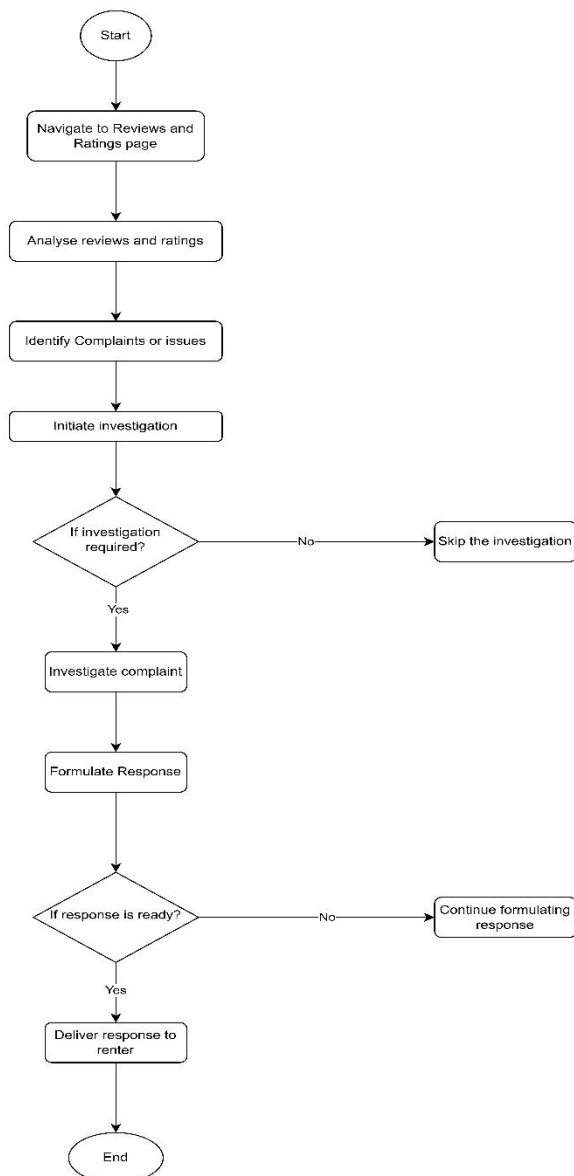
Pseudocode

```
BEGIN ComplaintInvestigationProcess:

    DECLARE complaints AS List

FUNCTION retrieveRenterComplaints () RETURNS List:

    END FUNCTION

    FUNCTION investigateComplaint(complaint) RETURNS InvestigationResult:

        DECLARE investigationResult AS InvestigationResult

RETURN investigationResult

    END FUNCTION

FUNCTION formulateResponse(investigationResult) RETURNS Response:

        DECLARE response AS Response

 RETURN response

    END FUNCTION

FUNCTION deliverResponse (complaint, response):

        DISPLAY "Response delivered to the renter for complaint: " + complaint

    END FUNCTION

 complaints = retrieveRenterComplaints ()

    FOR EACH complaint IN complaints:

        investigationResult = investigateComplaint(complaint)

        response = formulateResponse(investigationResult)

        deliverResponse(complaint, response)

    END FOR
```
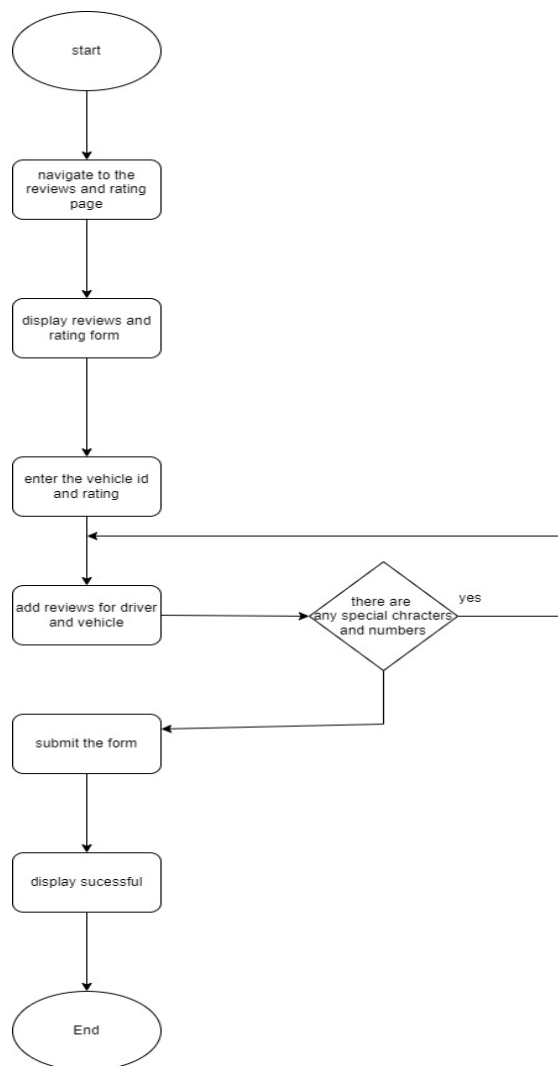
Reviews and Ratings Management System
IT22271150

Completed Features	Incomplete Features
• Develop a user reviews and rating form with necessary fields. • Implement validation for reviews and vehicle id • Create a delete reviews and rating functionality with a confirmation prompt. • Develop edit,display,create, functionalities.	• Display reviews and ratings in vehicle page • Report generate to specific vehicle id

Flow-chart

Driver Management System

IT22124876

Completed Features	IncompleteFeatures
• Implemented registration functionality allowing drivers to create accounts by providing necessary details. • Implemented a feature that allows drivers to view the list of registered drivers. • Implemented functionality where drivers receive booking details.	• Design a user interface for drivers where they can request their monthly report. • Develop a user interface for drivers to view their assigned hires and select the ones they need to reject.

Pseudocode

Start

Receive Request for Driver Assignment

Check Driver Availability

If driver is available:

Assign Driver to Job

Update Driver's Status

Update Job Status

Send Confirmation to User

If driver is not available:

Check Backup Driver Availability

If backup driver is available:

Assign Backup Driver to Job

Update Backup Driver's Status

Update Job Status

Send Confirmation to User

If backup driver is not available:

Send Unavailability Notification to User

End

HR Management System
IT22286246

Completed Features	Incomplete Features
• Develop the employee registration and update and delete employee details. • Develop the salary allocation part. • Develop delete and update salary allocate details. • Develop the leave request management part. • Develop the HR manager review leave details.	• There is still some work to be done on the part of approving leave requests. • Generate report in system

Flow Chart
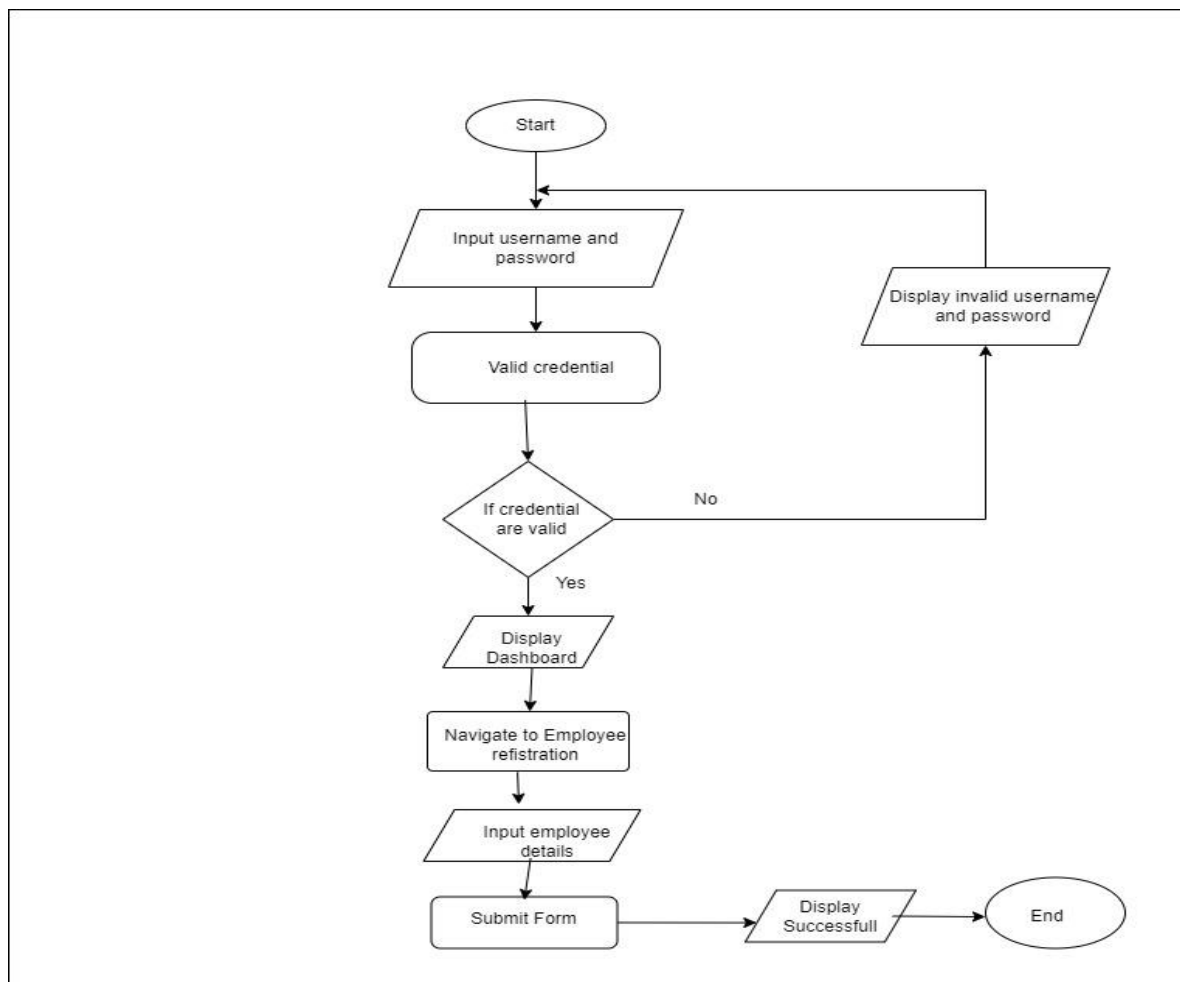
Pseudocodes

BEGIN EmployeeRegistrationProcess:

    DISPLAY Registration From.

    DECLARE firstName, lastName, NIC, role, dateOfBirth, gender, contactNumber, email AS STRING

    DISPLAY "Enter First Name:"

    READ firstName

    DISPLAY "Enter Last Name:"

    READ lastName

    DISPLAY "Enter NIC (National Identity Card Number):"

    READ NIC

    DISPLAY "Enter Role:"

    READ role

    DISPLAY "Enter Date of Birth (YYYY-MM-DD):"

    READ dateOfBirth

    DISPLAY "Select Gender (Male/Female/Other):"

    READ gender

    DISPLAY "Enter Contact Number:"

    READ contactNumber

    DISPLAY "Enter Email:"

    READ email

    IF validateEmployeeDetails(firstName, lastName, NIC, role, dateOfBirth, gender, contactNumber, email) THEN

        CALL saveEmployeeData(firstName, lastName, NIC, role, dateOfBirth, gender, contactNumber, email)

        DISPLAY "Registration successful.!"

    ELSE:

   DISPLAY "Invalid details."

END.

Vehicle Owner Management System
IT22101242

Completed Features	Incomplete Features
- Implemented owner authentication to securely register owner and create their profiles in the system - Implemented a profile page where registered owners can view and update, delete their profile information - Implemented a page where vehicle owners can add their vehicle details and added validation to ensure that required fields are filled out correctly - Provided a page where vehicle owners can view a list of their vehicles and perform actions such as editing and deleting - Implemented user interface elements for vehicle owners to add driver details	- Implement to update and delete options for driver details - Provide dashboard for vehicle owners to customize their notification preferences - Design a user-friendly dashboard to view owner's earnings

Flow-chart
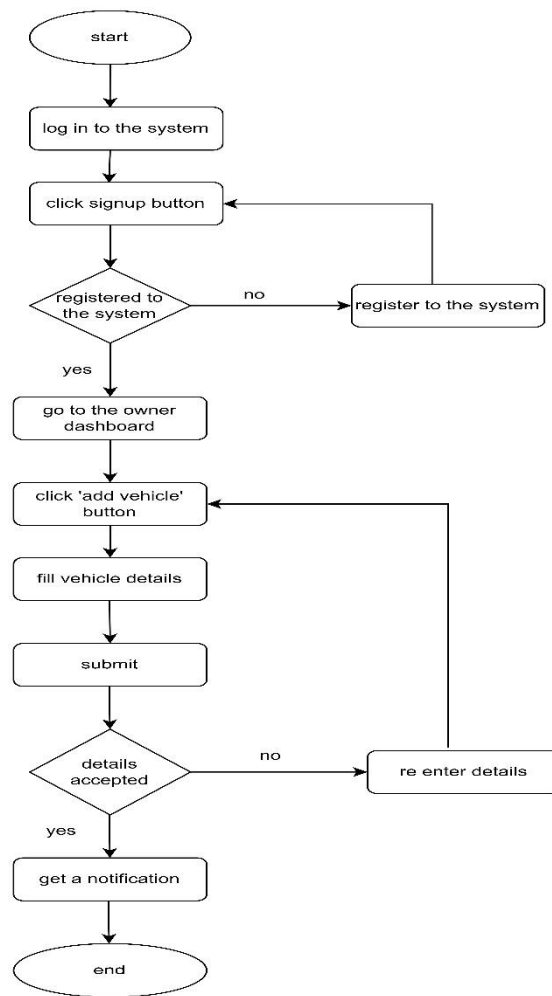
Pseudocode

BEGIN VehicleRegistrationProcess:

    DISPLAY Registration Form for Vehicle Details.

    DECLARE brand, model, modelYear, engineCapacity, totalSeats, vehicleID, startDate, transmission, price, mileage AS STRING

    DISPLAY "Enter Brand:"

    READ brand

    DISPLAY "Enter Model:"

    READ model

    DISPLAY "Enter Model Year:"

    READ modelYear

```
        DISPLAY "Enter Engine Capacity:"

        READ engineCapacity

        DISPLAY "Enter Total Seats:"

        READ totalSeats

        DISPLAY "Enter Vehicle ID:"

        READ vehicleID

        DISPLAY "Enter Start Date (YYYY-MM-DD):"

        READ startDate

        DISPLAY "Enter Transmission Type:"

        READ transmission

        DISPLAY "Enter Price:"

        READ price

        DISPLAY "Enter Mileage:"

        READ mileage

IF validateVehicleDetails(brand, model, modelYear, engineCapacity, totalSeats, vehicleID, startDate,
transmission, price, mileage) THEN

 CALL saveVehicleData(brand, model, modelYear, engineCapacity, totalSeats, vehicleID, startDate,
transmission, price, mileage)

      DISPLAY "Vehicle details added successfully.!"

   ELSE:

      DISPLAY "Invalid details."

END.
```