Information Systems and Data Modeling
IT1090

Group Assignment

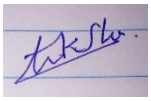
Title: (Online Customer Support System)	
Batch Number: (4.2)	Group Number: (MLB_04.02_07)

Declaration:

We hold a copy of this assignment that we can produce if the original is lost or damaged.

We hereby certify that no part of this assignment has been copied from any other group's work or from any other source. No part of this assignment has been written / produced for our group by another person except where such collaboration has been authorized by the subject lecturer/tutor concerned.

Group Members:

Student ID	Name	Signature
IT23631410	B G S H Dilshara	
IT23588714	K H S Dinsara	
IT23775442	T G C Theekshana	
IT23656574	G H C R Saumya	
IT23637832	D.D. Samaranayaka	

Submitted on: 06/10/2024

Contents

1 Introduction

We looked at how the internet links businesses and consumers while researching luxury automobile solution systems. We also looked at how a website may increase a client's accessibility to a service. Consequently, we devised a remedy for the motor solutions sector by building a website where consumers may peruse, purchase, and obtain car solutions online. By registering, customers may also use our website to provide suggestions for improvements, get customer support, and ask questions about our products or services.

A variety of technologies must be explored and understood to develop a luxury motor solutions website. Programming languages (such as HTML, CSS, and JavaScript) and relational databases were among them (Such as My SQL).

A database has to have broad data management capabilities. Because it is simpler to manage a computer-based stored data system than it is to manage written or typed materials, databases are necessary. This saves time and effort for both customers and administrators. There's also no chance of data loss because it's saved in a database. The only people who can control the system are its users. Gathering requirements and doing requirement analysis are essential while developing a database. Next, it will be possible to ascertain the data needs and the functional and nonfunctional standards.

2 Hypothetical Scenario

A guest lands on the website based on curiosity and goes through the long list of luxury cars on display. The guest navigates models such as the BMW X5, but since they are still a guest, they are confined to limited features. Pleased with the variety on display, he signs up to receive special offers and have a personalized session.

Now, they are a user who logs in to get further access to more vehicle details like financing options and the real status of availability. Intending to buy a BMW X5, he contacts online chat support inquiring about options on warranty and time of delivery. The response from the chat assistant is immediate, full of details, and thus very helpful in his making a decision.

Slightly apprehensive, he gets hold of the online call assistance to speak with a person personally. The customer speaks to a customer care representative, who advances the last steps in buying and ascertains the vehicle's availability at the dealer. The user, confident now, proceeds with the purchase.

In the background, the admin monitors the website's functionality by ensuring seamless operations through account handling and securing the platform. After sales, a marketing manager can identify the user as a high-value customer and will send customized offers for subsequent purchases of products and related maintenance services to keep the loyalty going in the long run.

3 Requirement Analysis

3.1 Main Requirements

3.1.1 Functional Requirements

Main Functions of the website and Events that take place between the users and the system is described by the Functional requirements. Six users are using this Online Luxury Motor Solution system. Namely: Guest, Registered User, Online call assistant, Admin, Chat assistant and the Marketing manager. They access this system in different ways where it is related to them.

1. Guest and Registered user (They can access the front-end of the system)

User requirements for Guest-

- View a list of available vehicles for sale or rent, their details, and specifications.
- Search for vehicles based on particular criteria such as make, model, price, location, etc.
- View partial reviews and ratings of the vehicle or services.
- View information about available services such as maintenance, repair, insurance, etc.
- Ability to send in a question/inquiry via contact forms without registration.
- Specials: Any promotions or special deals regarding vehicles or services.
- The system will pop up to create an account or log in for further actions or access to special features.
- System term and condition views, privacy policies.
- Can browse all vehicles with detailed descriptions, high-quality images, and videos.
- Can book or reserve vehicles for purchase, rental, or test drives.
- Can make purchases, submit payments, or apply for financing if applicable.
- Can schedule service appointments for repairs, maintenance, or inspections.

System requirements for Guest –

- The guest user should have privileges to view the sale, rental, or other services like maintenance and insurance of the vehicles.
- Enable the guest user to view promotions, discounts, or special offers concerning vehicles and services.
- Mailing: Engage with guests about navigating your facility through automated or live chat.
- There should also be pages for "About Us," "Terms of Service," "Privacy Policy, and FAQs accessible to every guest.

System requirements for Registered user –

- Sign-up of new users with basic information.
- After every login, redirect him directly to the personal dashboard showing
- The registered users can directly buy vehicles from the website by choosing the financing options.
- Succesfully paid the system generates order confirmation and sends it via email and also on the dashboard.
- Registered users should be able to submit reviews, rating of vehicles, dealers, or services they have used.
- Registered users shall be able to send and receive messages with any of the dealers, sellers, or service providers on the website.
- Send automated notifications and alerts of purchase confirmation, delivery updates, and service reminders.

2. Online call assistant and chat assistant (They can access the back-end of the system)

User requirements-

- Can allow call center assistants to access/view customer information, which includes personal details, history of orders, service appointments, and enquiries.
- Assistants could respond to customer queries directly over live chat or via email through the backend system.
- It should be able to register customer complaints and forward those complaints to the relevant departments for proper action.
- Provide assistants with access to current promotional offers and loyalty programs they can, where appropriate, apply to customer accounts.

System requirements-

- User-friendly backend dashboard that displays an overview of the customer accounts, tasks in progress, open inquiries, and escalated cases.
- They can also send and receive the customer's e-mails within the back-end system and manage them, with access to e-mail templates.
- Allow assistants to search for particular documentation or support materials by keyword or tag.
- Backup,The back end shall provide the assistants with the facility to create, assign, and track cases with regard to customer complaints or issues.

3. Admin (They can access the back-end of the system)

User requirements-

- Admin can perform the creation, update, and delete of user accounts.
- Admin manages the service packages.
- Admin manages payment gateways: configuration, addition of new gateways, setting of commission rates, and updating payment rules.
- Admin can manage system security settings, such as password policies and multi-factor authentication.
- Admins should access real-time monitoring tools to check the system performance.
- The provision of tools for administrators to manage Data Subject Requests and verify that the system is compliant with applicable data protection laws.

System requirements-

- Admins should be able to export data.
- Admins can exercise full control over user accounts: create, edit, suspend, or delete an account.
- Admin logins should be supported to add an extra layer of security.
- Performing the backup internally at regular intervals and securely storing the backup information.
- Systemize the logging of system errors that admins can review and debug.

4. Marketing manager (They can access the back-end of the system)

User requirements-

- It should grant a Marketing Manager the ability to create, edit, and then launch marketing campaigns regarding promotions and discounts on special offers for vehicle sales and services.
- Marketers should be in a position to send messages only to the selected customer segments based on their interest or past interaction.
- Content management: It involves uploading and management of all types of banners, promotional images, marketing text used in different campaigns along the frontend of the system.
- There is a need to provide tools to the marketing manager so that he can monitor in real time the performance of the marketing campaigns.

System requirements-

- Provide intuitive interfaces to create, manage, and track the results of marketing campaigns.
- enable marketing managers to create their ad hoc segments through filters such as geographic location, purchase history, vehicle preference, or service history.
- We want to provide them with a facility for creating lead capture forms and workflows that can possibly automate follow-up messages and nurturing campaigns.

3.1.1 Non-Functional Requirements

Simply said, quality qualities are non-functional needs. The system's attributes that aren't directly related to a certain function are explained. More importance could be placed on non-functional criteria than on functional ones. The system can be worthless if these goals are not met.

Speed

- The system needs to operate quickly.
- The system can access more users at same time without any failures.

Availability

- The system should be available in 24/7.

User friendly

- Even people with less IT knowledge should be able to utilize the system.

Reliability

- The system must have ability to detect the invalid user credentials.

Security

- It should be able to block unauthorized access, misuse, forgery, and keep user data secured.
- Also, nobody can access the system using any other's user ID and password by providing a unique user ID and password.

Scalability

- The system should be able to handle a higher workload on-demand.

Performance

- Admin can add, edit, remove, update properties
- Any number of users can be able to access the system at the same time and the response of the system regarding to the user requests will be very high.

3.2 Data Requirements

- Customer
    - CustomerID
    - Address
    - LicenseNumber
    - PhoneNumber
    - DOB
    - Email
    - First_name
    - Last_name

- Vehicle
    - Fuel_type
    - VehicleID
    - Model
    - Status
    - Price
    - Make
    - Year

- Salesperson
    - Email
    - SalespersonID
    - Phonenumber
    - Hiredate
    - SalesTarget
    - Department
    - First_name
    - Last_name

- Sale
    - SaleID
    - SaleDate
    - VehicleID
    - SalespersonID
    - PaymentMethod
    - SalePrice
    - CustomerID

- Service
    - ServiceDate
    - Description
    - Service_type
    - MechanicID
    - ServiceID
    - VehicleID
    - Cost

- Inventory
    - InventoryID
    - VehicleID
    - Location
    - StockStatus
    - LastUpdated
    - Quantity

- Supplier
    - SupplierID
    - Email
    - Rating
    - SupplierType
    - Name
    - Adress

- Order
    - OrderID
    - SupplierID
    - VehicleID
    - OrderStatus
    - DeliveryDate
    - OrderDate
    - TotalCost
    - Quantity

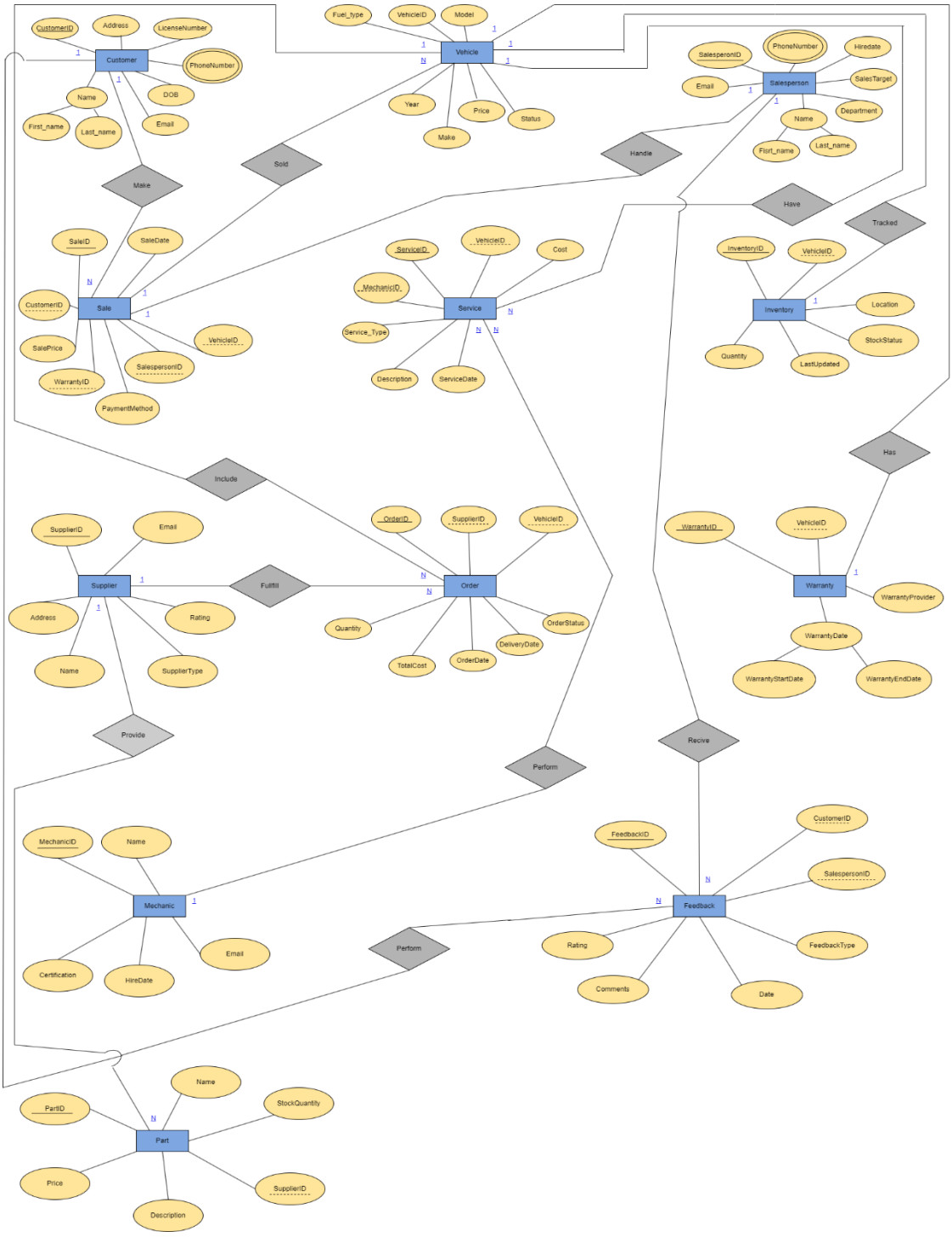
- Warranty
    - WarrantyID
    - VehicleID
    - WarrantyProvider
    - WarrantyStartDate
    - WarrantyEndDate

- Mechanic
    - MechanicID
    - Name
    - Email
    - HireDate
    - Certification

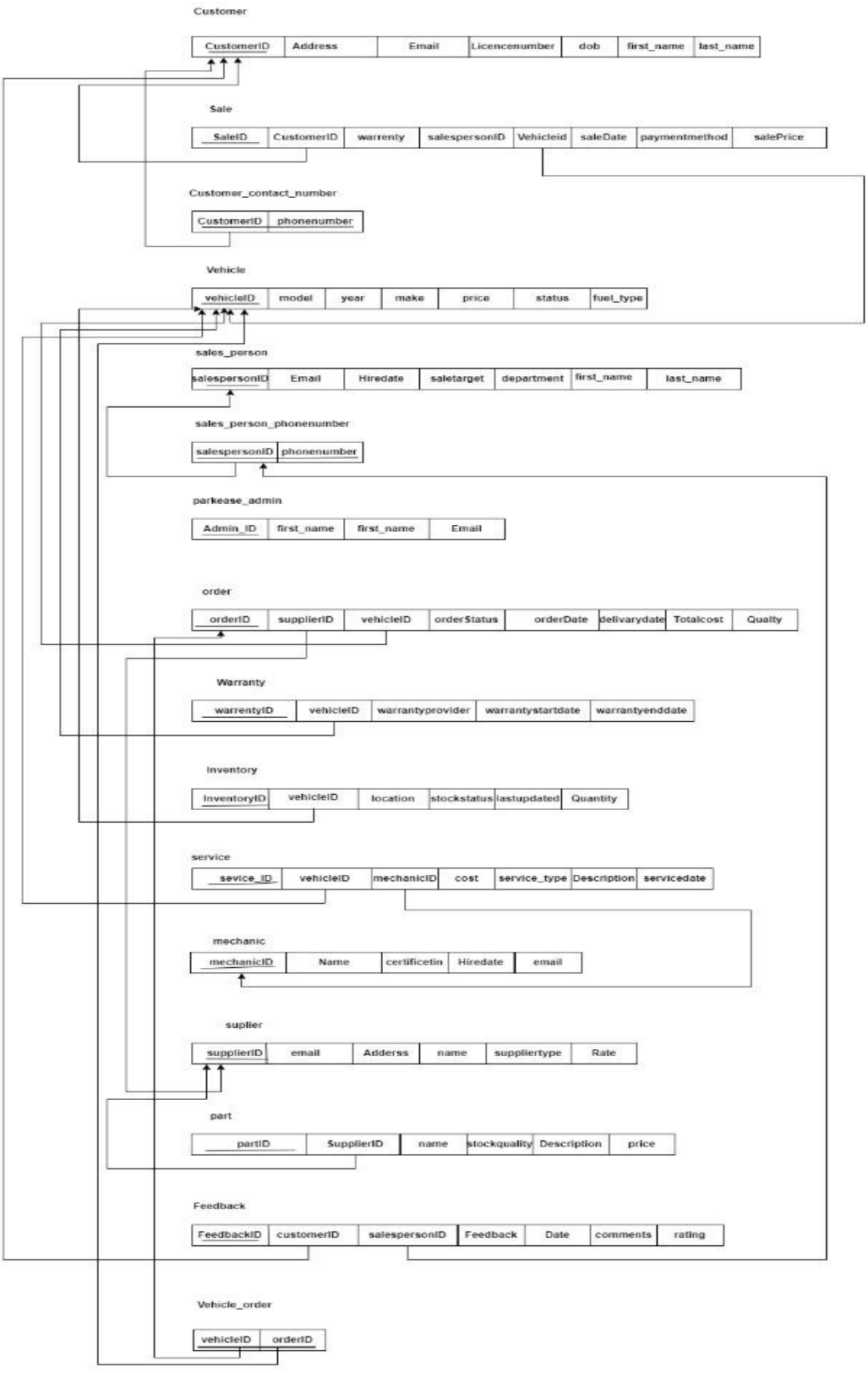
- Feedback
    - FeedbackID
    - CutomerID
    - SalespersonID
    - FeedbackType
    - Date
    - Comments
    - Rating

- Part
    - Name
    - StockQuantity
    - SupplierID
    - Description
    - Price
    - PartID

4 Entity Relationship (ER Diagram)



5 Relational Schema



6 SQL Queries

6.1 Data Base Create

```
Create Customer Table
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    Address VARCHAR(255),
    Email VARCHAR(255),
    Licencenumber VARCHAR(50),
    dob DATE,
    first_name VARCHAR(100),
    last_name VARCHAR(100)
);

Create Customer Contact Number Table
CREATE TABLE Customer_contact_number (
    CustomerID INT,
    phonenumber VARCHAR(20),
    PRIMARY KEY (CustomerID, phonenumber),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

 Create Vehicle Table
CREATE TABLE Vehicle (
    vehicleID INT PRIMARY KEY,
    model VARCHAR(100),
    year INT,
    make VARCHAR(100),
    price DECIMAL(10, 2),
    status VARCHAR(50),
    fuel_type VARCHAR(50)
);

Create Sales Person Table
CREATE TABLE Sales_person (
    salespersonID INT PRIMARY KEY,
    Email VARCHAR(255),
    Hiredate DATE,
    saletarget INT,
    department VARCHAR(100),
    first_name VARCHAR(100),
    last_name VARCHAR(100)
);

Create Sales Person Phone Number Table
CREATE TABLE Sales_person_phonenumber (
    salespersonID INT,
    phonenumber VARCHAR(20),
    PRIMARY KEY (salespersonID, phonenumber),
    FOREIGN KEY (salespersonID) REFERENCES Sales_person(salespersonID)
);
```

```sql
Create Mechanic Table
CREATE TABLE Mechanic (
    mechanicID INT PRIMARY KEY,
    Name VARCHAR(100),
    certificatcin VARCHAR(100),
    Hiredate DATE,
    email VARCHAR(255)
);

Create Sale Table
CREATE TABLE Sale (
    SaleID INT PRIMARY KEY,
    CustomerID INT,
    warranty VARCHAR(100),
    vehicleID INT,
    saleDate DATE,
    paymentmethod VARCHAR(50),
    salePrice DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (salespersonID) REFERENCES Sales_person(salespersonID),
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID)
);

Create Warranty Table
CREATE TABLE Warranty (
    warrantyID INT PRIMARY KEY,
    vehicleID INT,
    warrantyprovider VARCHAR(100),
    warrantystartdate DATE,
    warrantyenddate DATE,
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID)
);

 Create Inventory Table
CREATE TABLE Inventory (
    InventoryID INT PRIMARY KEY,
    vehicleID INT,
    location VARCHAR(100),
    stockstatus VARCHAR(50),
    lastupdated DATE,
    Quantity INT,
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID)
);

Create Service Table
CREATE TABLE Service (
    service_ID INT PRIMARY KEY,
    vehicleID INT,
    mechanicID INT,
    cost DECIMAL(10, 2),
    service_type VARCHAR(100),
    Description TEXT,
    servicedate DATE,
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID),
    FOREIGN KEY (mechanicID) REFERENCES Mechanic(mechanicID)
);
```

```sql
Create Supplier Table
CREATE TABLE Supplier (
    supplierID INT PRIMARY KEY,
    email VARCHAR(255),
    Address VARCHAR(255),
    name VARCHAR(100),
    suppliertype VARCHAR(100),
    Rate DECIMAL(10, 2)
);

Create Part Table
CREATE TABLE Part (
    partID INT PRIMARY KEY,
    SupplierID INT,
    name VARCHAR(100),
    stockquality VARCHAR(50),
    Description TEXT,
    price DECIMAL(10, 2),
    FOREIGN KEY (SupplierID) REFERENCES Supplier(supplierID)
);

Create VOrder Table
CREATE TABLE VOrder (
    orderID INT PRIMARY KEY,
    supplierID INT,
    vehicleID INT,
    orderStatus VARCHAR(50),
    orderDate DATE,
    deliverydate DATE,
    Totalcost DECIMAL(10, 2),
    Quality VARCHAR(50),
    FOREIGN KEY (supplierID) REFERENCES Supplier(supplierID),
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID)
);

Create Feedback Table
CREATE TABLE Feedback (
    FeedbackID INT PRIMARY KEY,
    customerID INT,
    salespersonID INT,
    Feedback TEXT,
    Date DATE,
    comments TEXT,
    rating INT,
    FOREIGN KEY (customerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (salespersonID) REFERENCES Sales_person(salespersonID)
);

Create Vehicle Order Table
CREATE TABLE Vehicle_order (
    vehicleID INT,
    orderID INT,
    PRIMARY KEY (vehicleID, orderID),
    FOREIGN KEY (vehicleID) REFERENCES Vehicle(vehicleID),
    FOREIGN KEY (orderID) REFERENCES VOrder(orderID)
);
```

Inserting values for tables

```
INSERT INTO Customer (CustomerID, Address, Email, Licencenumber, dob, first_name,
last_name)
VALUES
(1, '123 Rajagirya', 'senash@gmail.com', 'AA-4545', '2003-01-01', 'Senash', 'Dilshara'),
(2, '85/1 Malbe ', 'chesmisubaaa@gmail.com', 'BB-2525', '2005-02-02', 'Chesmi',
'Kothalawala'),
(3, '85/1 Bambalapitiya', 'chamodi@gmail.com', 'LK-639', '1998-03-03', 'Chamodi',
'Mendis'),
(4, '65/8 Kumaragewatta', 'sinethbanda@gmail.com', 'UH-45', '1999-04-04', 'Theekshana',
'Kota'),
(5, '50/A Mathara', 'theekshana@gmail.com', 'CK-2525', '2000-05-05', 'Shameen',
'Saumya');
```

```
INSERT INTO Customer_contact_number (CustomerID, phonenumber)
VALUES
(1, '076369565'),
(1, '075245558'),
(2, '075585555'),
(3, '071545654'),
(4, '0');
```

```
INSERT INTO Vehicle (vehicleID, model, year, make, price, status, fuel_type)
VALUES
(1, 'Model S', 2020, 'Tesla', 100000.00, 'Available', 'Electric'),
(2, 'Civic', 2019, 'Honda', 400000.00, 'Sold', 'Gasoline'),
(3, 'Accord', 2021, 'Honda', 500000.00, 'Available', 'Gasoline'),
(4, 'Camry', 2022, 'Toyota', 800000.00, 'Available', 'Gasoline'),
(5, 'Mustang', 2020, 'Ford', 700000.00, 'Available', 'Gasoline');
```

```
INSERT INTO Mechanic (mechanicID, Name, certificatcin, Hiredate, email)
VALUES
(1, 'Senash Dilshara', 'MEC123', '2015-06-15', 'senash@gmail.com'),
(2, 'Theekshana Gamage', 'MEC456', '2018-08-20', 'theekshana@gmail.com'),
(3, 'Sineth Dinsara', 'MEC789', '2019-11-05', 'sineth@gmail.com'),
(4, 'Chamodi Dinusha', 'MEC234', '2020-01-10', 'chamodi@gmail.com'),
(5, 'Chesmi Kothalawa', 'MEC678', '2021-03-30', 'chesmi@gmail.com');
```

```
INSERT INTO Sales_person (salespersonID, Email, Hiredate, saletarget, department,
first_name, last_name)
VALUES
(1, 'malpala@example.com', '2019-01-01', 100000, 'Sales', 'Malpala', 'Kumara'),
(2, 'samaranayaka@example.com', '2020-01-01', 120000, 'Sales', 'Samaranayaka',
'Kumarahami'),
(3, 'gamapala@gmail.com', '2021-01-01', 80000, 'Sales', 'Gamapala', 'Malpala'),
(4, 'kumara@gmail.com', '2022-01-01', 90000, 'Sales', 'Kumara', 'Hami'),
(5, 'saumya@gmail.com', '2023-01-01', 70000, 'Sales', 'Saumya', 'Marathunga');

INSERT INTO Sales_person_phonenumber (salespersonID, phonenumber)
```

```
VALUES
(1, '07856565'),
(1, '07255896'),
(2, '07625848'),
(3, '07896556'),
(4, '07542256');

INSERT INTO Sale (SaleID, CustomerID, warranty, salespersonID, Vehicleid, saleDate,
paymentmethod, salePrice)
VALUES
(1, 1, '2 Years', 1, 1, '2024-03-01', 'Credit Card', 78599.00),
(2, 2, '3 Years', 2, 2, '2024-04-05', 'Cash', 22100.00),
(3, 3, '1 Year', 3, 3, '2024-05-10', 'Credit Card', 26520.00),
(4, 4, '5 Years', 1, 4, '2024-06-15', 'Debit Card', 30450.00),
(5, 5, '2 Years', 2, 5, '2024-07-20', 'Cash', 45650.00);

INSERT INTO Warranty (warrantyID, vehicleID, warrantyprovider, warrantystartdate,
warrantyenddate)
VALUES
(1, 1, 'Tesla', '2024-01-01', '2026-01-01'),
(2, 2, 'Honda', '2024-01-05', '2027-01-05'),
(3, 3, 'Honda', '2024-01-10', '2025-01-10'),
(4, 4, 'Toyota', '2024-01-15', '2029-01-15'),
(5, 5, 'Ford', '2024-01-20', '2026-01-20');

INSERT INTO Inventory (InventoryID, vehicleID, location, stockstatus, lastupdated,
Quantity)
VALUES
(1, 1, 'Main Lot', 'In Stock', '2024-01-01', 10),
(2, 2, 'Main Lot', 'Sold', '2024-01-01', 0),
(3, 3, 'Main Lot', 'In Stock', '2024-01-01', 5),
(4, 4, 'Main Lot', 'In Stock', '2024-01-01', 8),
(5, 5, 'Main Lot', 'In Stock', '2024-01-01', 12);

INSERT INTO Supplier (supplierID, email, Address, name, suppliertype, Rate)
VALUES
(1, 'supplier1@example.com', '101 Supplier St', 'Supplier One', 'Parts', 500.00),
(2, 'supplier2@example.com', '102 Supplier St', 'Supplier Two', 'Parts', 750.00),
(3, 'supplier3@example.com', '103 Supplier St', 'Supplier Three', 'Parts', 600.00),
(4, 'supplier4@example.com', '104 Supplier St', 'Supplier Four', 'Parts', 550.00),
(5, 'supplier5@example.com', '105 Supplier St', 'Supplier Five', 'Parts', 700.00);


INSERT INTO Part (partID, SupplierID, name, stockquality, Description, price)
VALUES
(1, 1, 'Brake Pads', 'High', 'Quality brake pads for vehicles', 1500.00),
(2, 1, 'Oil Filter', 'Medium', 'Oil filter for various models', 2500.00),
(3, 2, 'Spark Plugs', 'High', 'High-quality spark plugs', 300.00),
(4, 2, 'Air Filter', 'Medium', 'Air filter for various models', 2000.00),
(5, 3, 'Battery', 'High', 'Car battery for multiple models', 30000.00);

INSERT INTO VOrder (orderID, supplierID, vehicleID, orderStatus, orderDate, deliverydate,
Totalcost, Quality)
VALUES
(1, 1, 1, 'Delivered', '2024-01-01', '2024-01-05', 80999.99, 'Good'),
(2, 2, 2, 'Pending', '2024-01-05', NULL, 25000.00, 'Fair'),
(3, 3, 3, 'Delivered', '2024-01-10', '2024-01-15', 26500.00, 'Good'),
(4, 1, 4, 'Pending', '2024-01-15', NULL, 30500.00, 'Good'),
```

```
(5, 2, 5, 'Delivered', '2024-01-20', '2024-01-25', 55000.00, 'Excellent');


INSERT INTO Feedback (FeedbackID, customerID, salespersonID, Feedback, Date, comments,
rating)
VALUES
(1, 1, 1, 'Excellent service!', '2024-09-02', 'Very satisfied with the purchase.', 10),
(2, 2, 2, 'Good experience', '2024-08-06', 'Satisfactory service overall.', 5);
```

7 Performance Requirement

The Performance Requirements are crucial to the system's success. They are listed below,

- At least 500–1000 concurrent users must be able to use the platform at once
- The system's performance criteria include speed and usability.
- All pertinent modules should get real-time inventory updates for cars, parts, and accessories.
- System must be active 24 hours, 365 days for a Registered user to access the system without any inconvenience.
- A Registered User can access the system numerous times by entering his/ her login credentials.
- Registered user can edit or delete his/her account details.
- System loads within a minimum time.
- Administrator can add or remove shared experiences and feedbacks, Approve Reservation Requests and cancel reservation requests, manage User accounts, edit/ update vehicle details.
- Website functionality may be expanded by developers.
- A developer might upgrade the system.

- For problems to be fixed, developers must find defects and faults.
- Create a user-friendly user interface.
- The website must always be accessible to users through any kind of browser or device.

8 Security Requirements

- Restricted functions ought to be inaccessible to unauthorized users.
- Every piece of data in the system ought to be backed up in the database.
- A strong password that contains capital, lowercase, digits, and special characters is required for the user account.
- There should only be one user account per email address.
- A redundant server is required for database server maintenance.
- The system's data is only accessible and modifiable by administrators.