Case Study - Car Rental System

by

Dinesh S

Project Overview :

The Car Rental System project is a menu-driven application with key functionalities like Customer Management, Car Management, Lease Management, and Payment Handling. It involves maintaining SQL-based tables for vehicles, customers, leases, and payments, with robust data handling using object-oriented principles. Custom exceptions manage errors, ensuring user-friendly interactions. Database utilities streamline connections, while comprehensive unit tests validate system reliability.

Schema Designing :

Created tables for vehicle, customer, payment, lease and inserted sample values.

```
mysql> use carrentdb;
Database changed
mysql> show tables;
+---------------------+
| Tables_in_carrentdb |
+---------------------+
| customer            |
| lease               |
| payment             |
| vehicle             |
+---------------------+
4 rows in set (0.03 sec)
```

```
mysql> select*from customer;
+------------+-----------+----------+-----------------------+-------------+
| customerID | firstName | lastName | email                 | phoneNumber |
+------------+-----------+----------+-----------------------+-------------+
|          1 | John      | Doe      | john.doe@example.com  | 1234567890  |
|          2 | Jane      | Smith    | jane.smith@example.com | 0987654321  |
+------------+-----------+----------+-----------------------+-------------+
2 rows in set (0.03 sec)

mysql> select*from lease;
+---------+-----------+------------+------------+------------+-------+
| leaseID | vehicleID | customerID | startDate  | endDate    | type  |
+---------+-----------+------------+------------+------------+-------+
|      51 |         2 |          2 | 2024-12-01 | 2024-12-10 | Daily |
+---------+-----------+------------+------------+------------+-------+
1 row in set (0.01 sec)

mysql> select*from vehicle;
+-----------+--------+-------+------+-----------+-----------+-------------------+----------------+
| vehicleID | make   | model | year | dailyRate | status    | passengerCapacity | engineCapacity |
+-----------+--------+-------+------+-----------+-----------+-------------------+----------------+
|         1 | Toyota | Camry | 2020 |    100.00 | available |                 5 |           2.50 |
|         2 | Honda  | Civic | 2021 |     80.00 | available |                 5 |           1.80 |
+-----------+--------+-------+------+-----------+-----------+-------------------+----------------+
2 rows in set (0.02 sec)

mysql> select*from payment;
+-----------+---------+-------------+----------+
| paymentID | leaseID | paymentDate | amount   |
+-----------+---------+-------------+----------+
|        52 |      51 | 2024-11-23  | 10000.00 |
+-----------+---------+-------------+----------+
1 row in set (0.01 sec)
```

Project Directory Structure :

1) entity:
   Created entity package to these classes and added getter, setter methods
   → Customer Class
   → Lease Class
   → Payment Class
   → Vehicle Class
2) dao:
   Created Interface called ICarLeaseRepository and Implementation file called CarLeaseRepositoryImpl to implement the interface methods.
   → ICarLeaseRepository class
   → CarLeaseRepositoryImpl class
3) Exception:
   Created exceptions for,
   → CarNotFoundException Class
   → CustomerNotFoundException Class
   → LeaseNotFoundException Class

4) Util:

   Created Util package to Connect to the database.

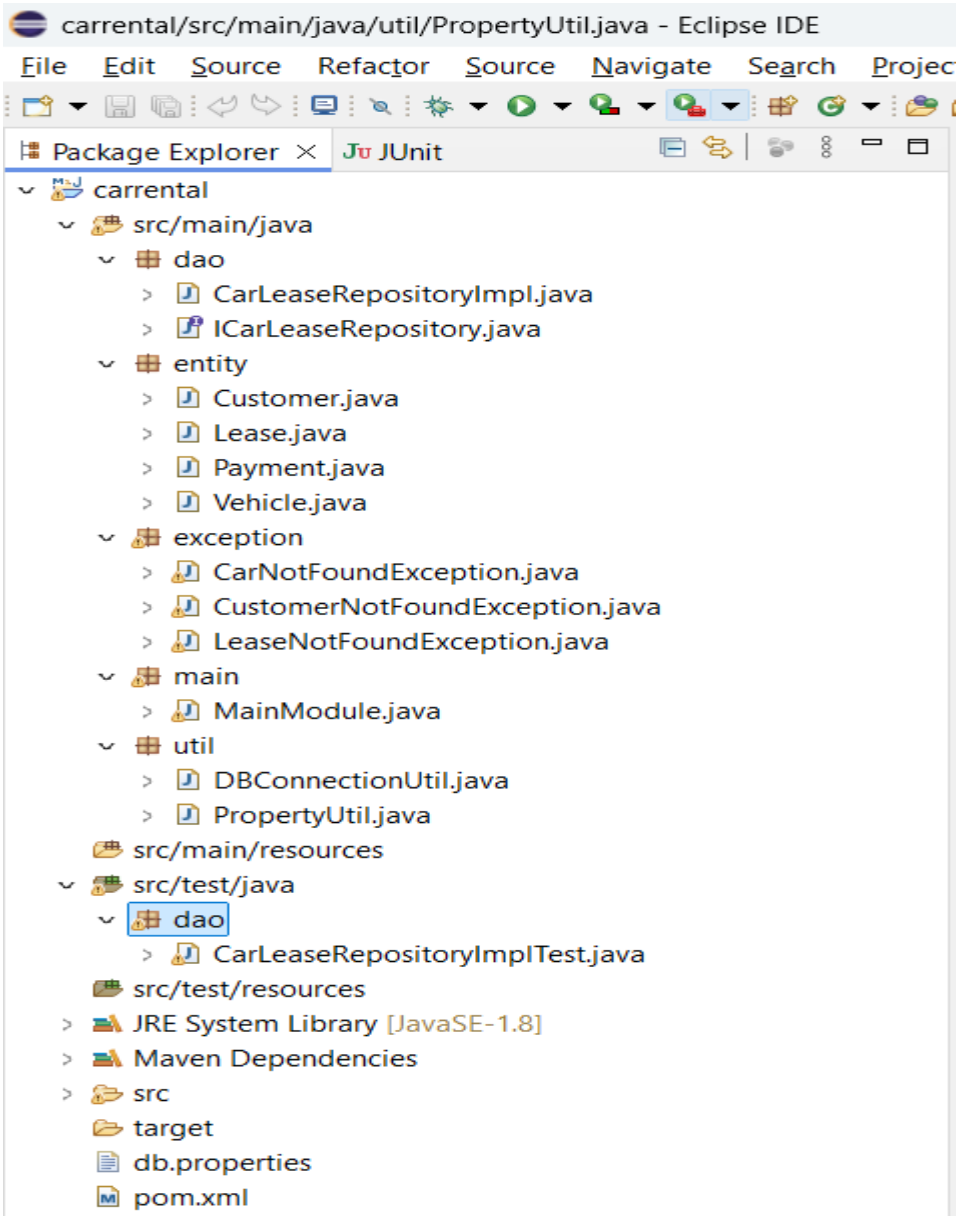
   → DBConnectionUtil class

   → PropertyUtil class

5) Main:

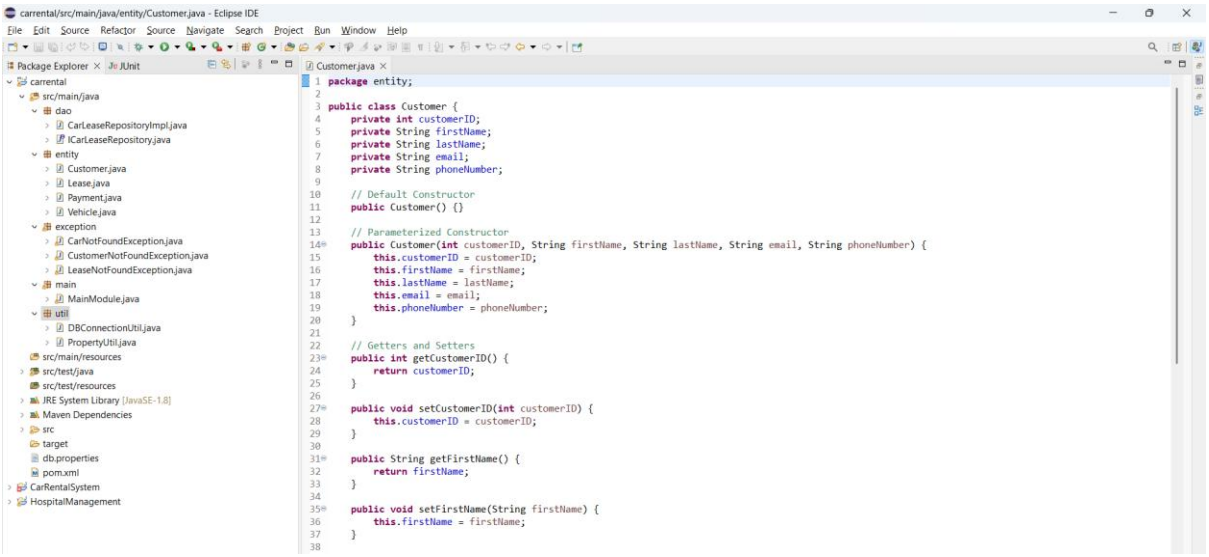
   Main class to trigger all the methods and get input from user.

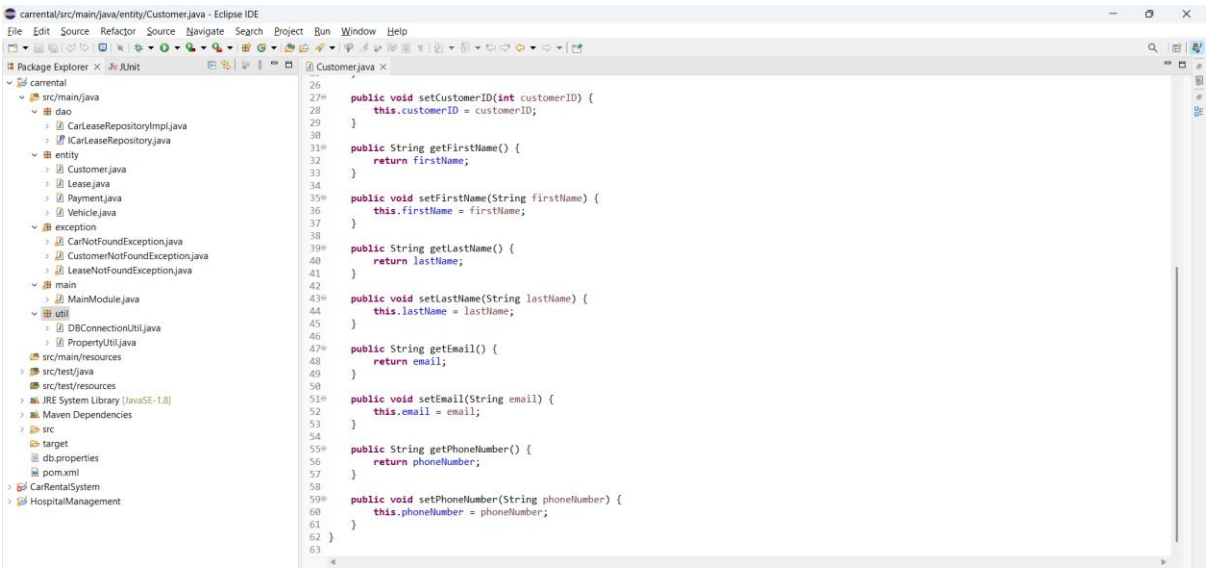
   → Main class



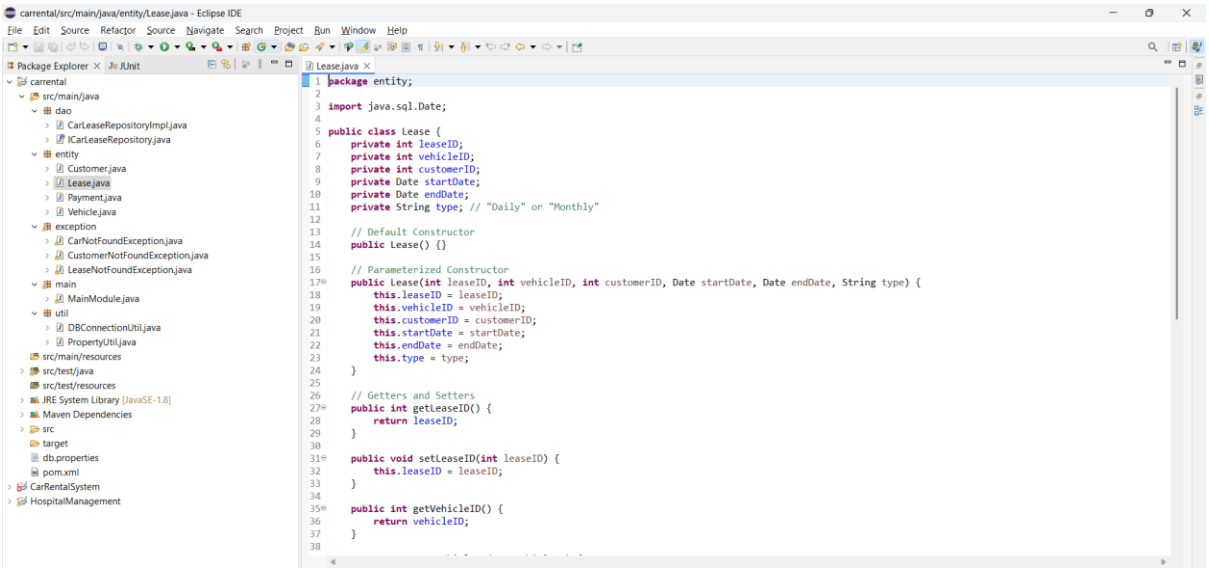
entity Package:

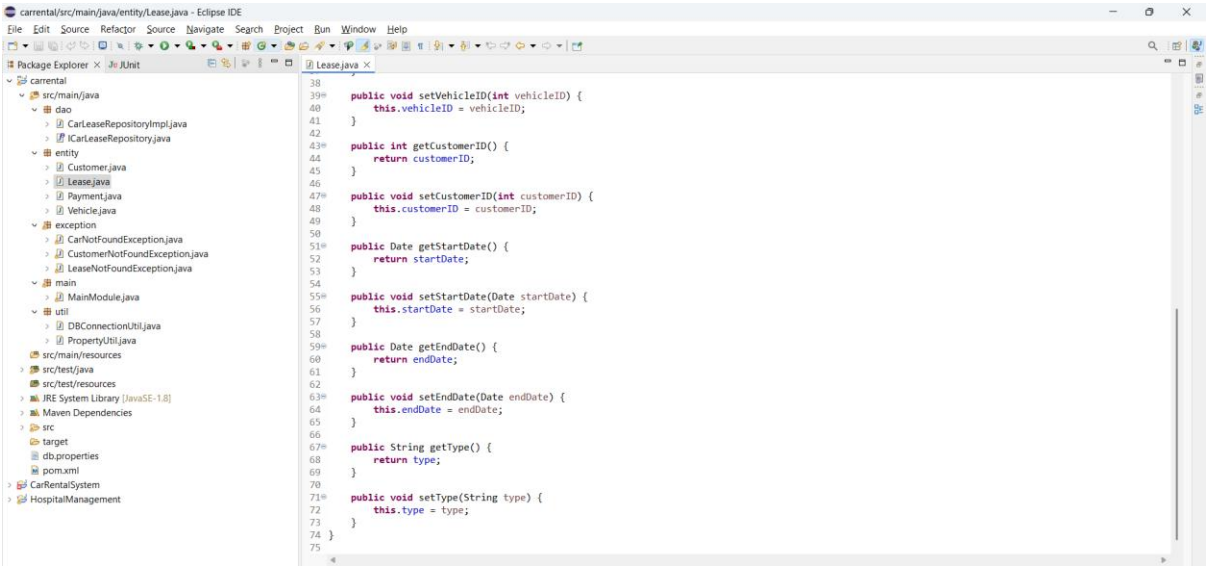
Customer.java :



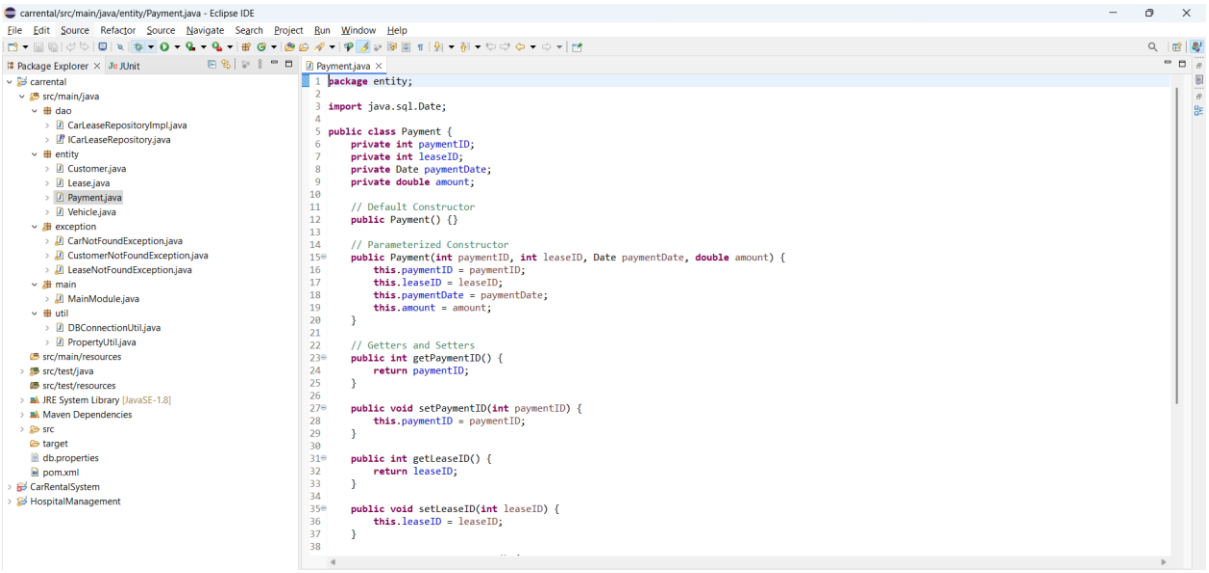


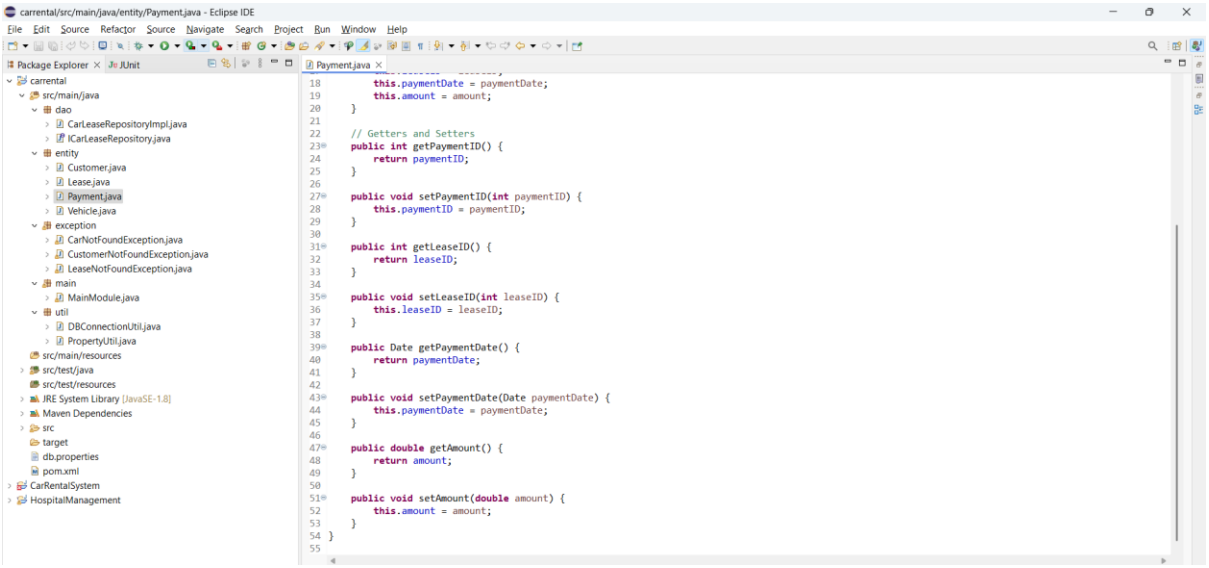
Lease.java:


carrental/src/main/java/entity/Lease.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
Lease.java
package entity;
import java.sql.Date;
public class Lease {
private int leaseID;
private int vehicleID;
private int customerID;
private Date startDate;
private Date endDate;
private String type; // "Daily" or "Monthly"
// Default Constructor
public Lease() {}
// Parameterized Constructor
public Lease(int leaseID, int vehicleID, int customerID, Date startDate, Date endDate, String type) {
this.leaseID = leaseID;
this.vehicleID = vehicleID;
this.customerID = customerID;
this.startDate = startDate;
this.endDate = endDate;
this.type = type;
}
// Getters and Setters
public int getLeaseID() {
return leaseID;
}
public void setLeaseID(int leaseID) {
this.leaseID = leaseID;
}
public int getVehicleID() {
return vehicleID;
}

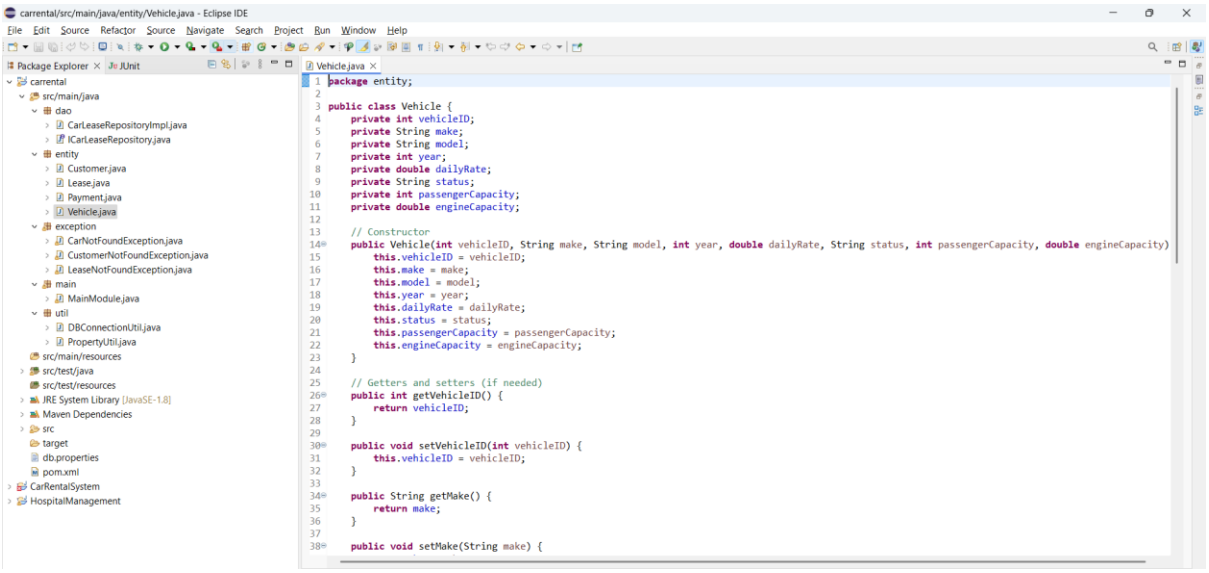

carrental/src/main/java/entity/Lease.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
Lease.java
public void setVehicleID(int vehicleID) {
this.vehicleID = vehicleID;
}
public int getCustomerID() {
return customerID;
}
public void setCustomerID(int customerID) {
this.customerID = customerID;
}
public Date getStartDate() {
return startDate;
}
public void setStartDate(Date startDate) {
this.startDate = startDate;
}
public Date getEndDate() {
return endDate;
}
public void setEndDate(Date endDate) {
this.endDate = endDate;
}
public String getType() {
return type;
}
public void setType(String type) {
this.type = type;
}
}

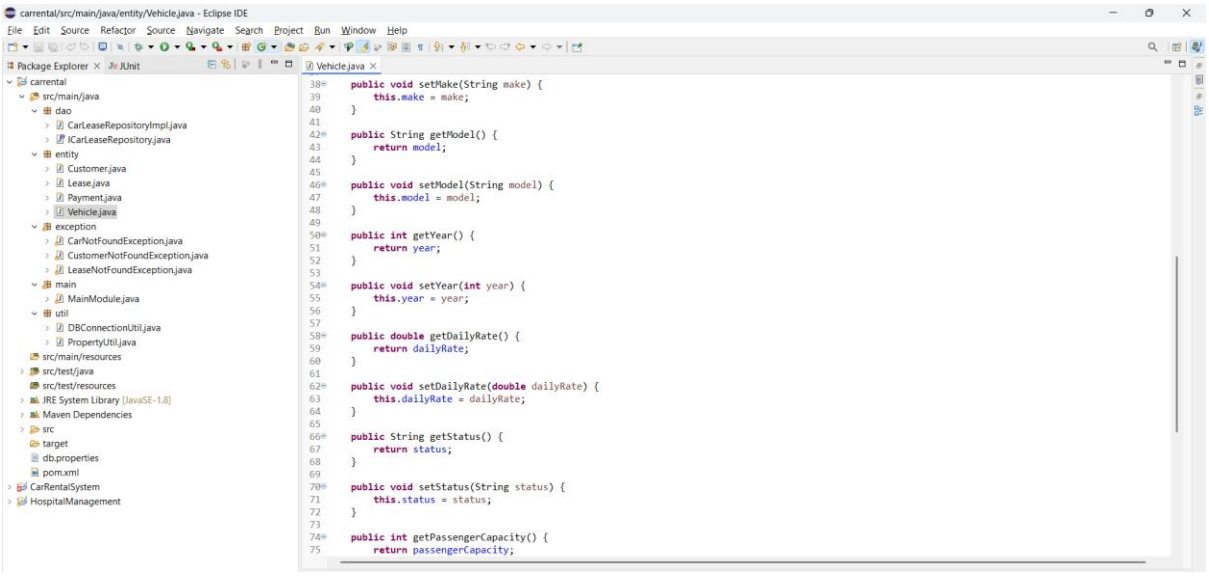
Payment.java:

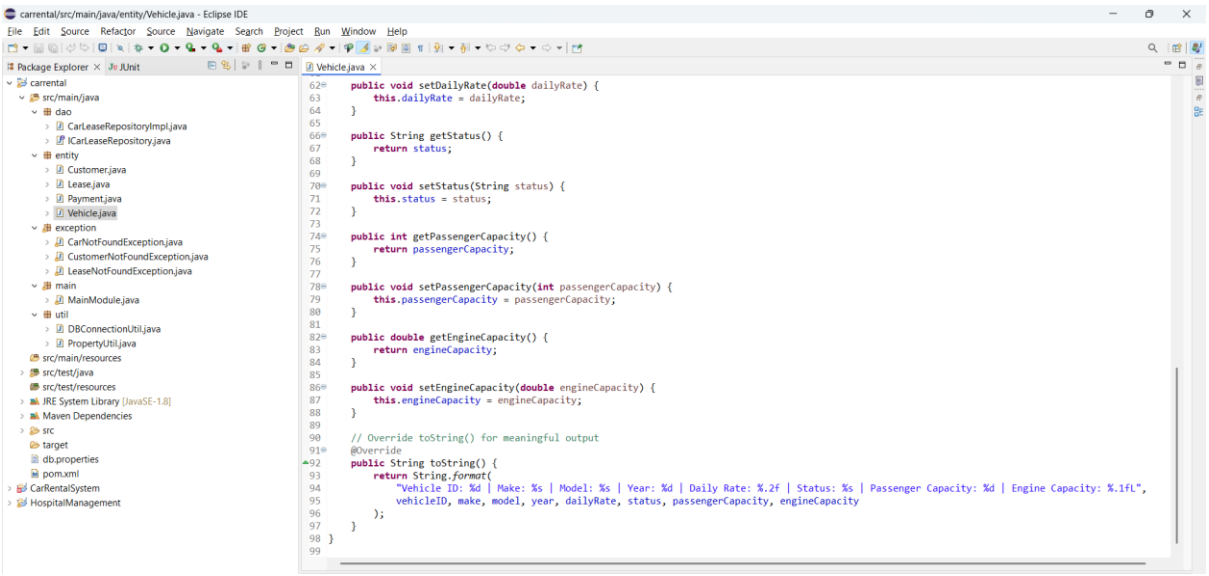




Vehicle.java:

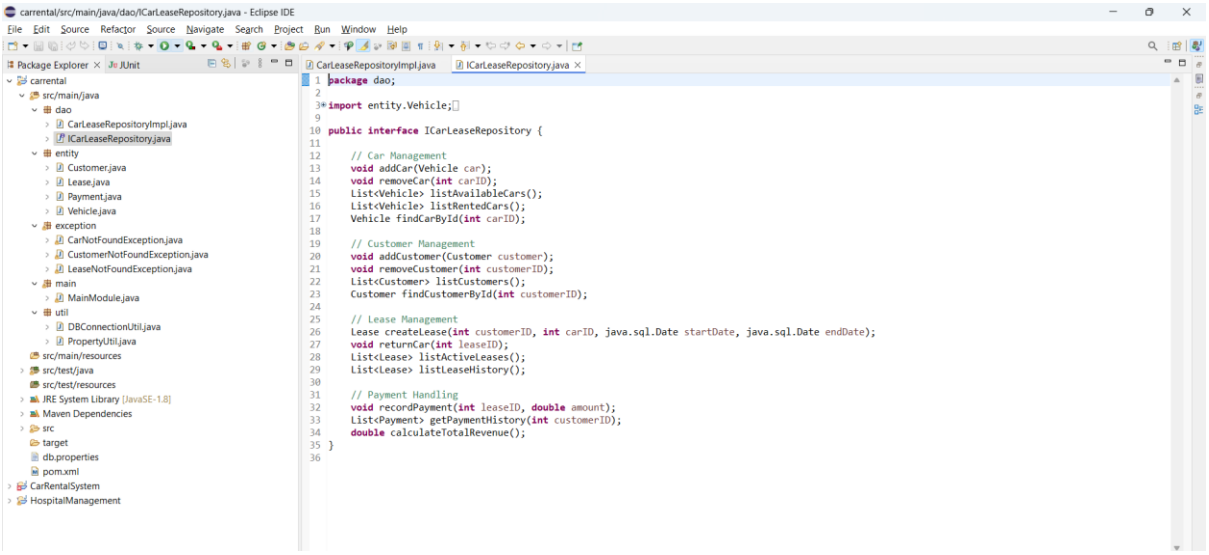






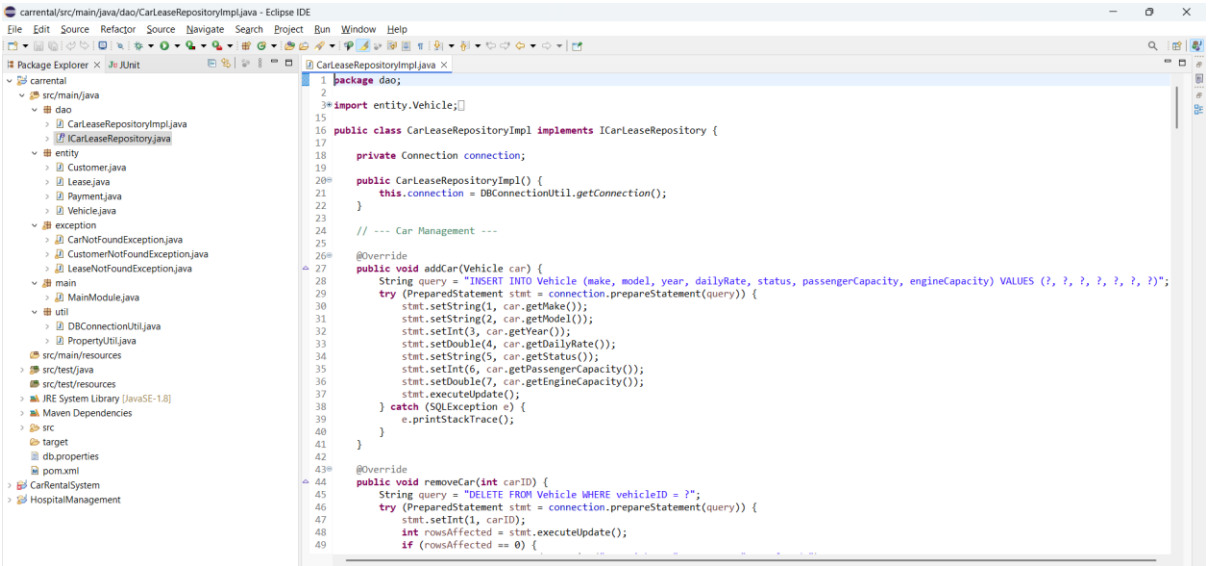
dao Package:

ICarLeaseRepository.java

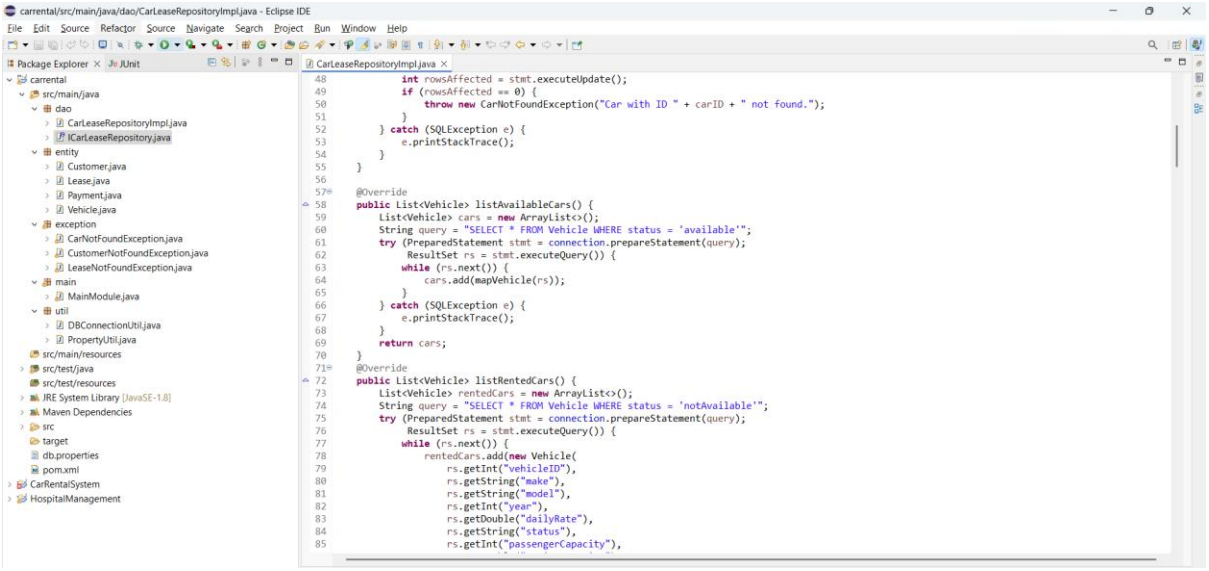


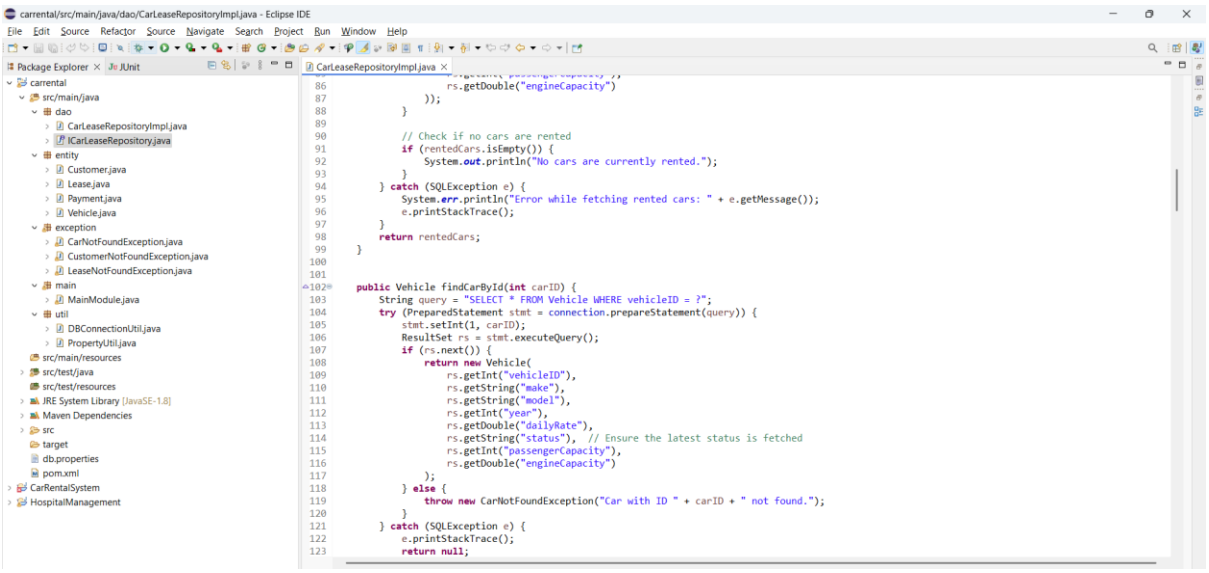
```
package dao;

import entity.Vehicle;

public interface ICarLeaseRepository {

    // Car Management
    void addCar(Vehicle car);
    void removeCar(int carID);
    List<Vehicle> listAvailableCars();
    List<Vehicle> listRentedCars();
    Vehicle findCarById(int carID);

    // Customer Management
    void addCustomer(Customer customer);
    void removeCustomer(int customerID);
    List<Customer> listCustomers();
    Customer findCustomerById(int customerID);

    // Lease Management
    Lease createLease(int customerID, int carID, java.sql.Date startDate, java.sql.Date endDate);
    void returnCar(int leaseID);
    List<Lease> listActiveLeases();
    List<Lease> listLeaseHistory();

    // Payment Handling
    void recordPayment(int leaseID, double amount);
    List<Payment> getPaymentHistory(int customerID);
    double calculateTotalRevenue();
}
```

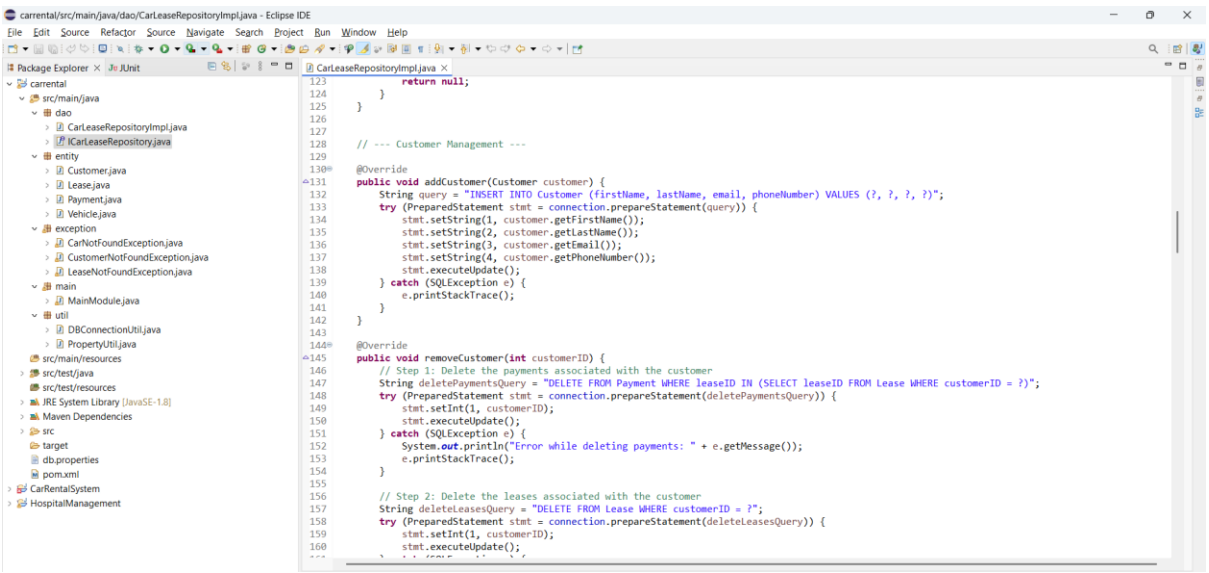
CarLeaseRepositoryImpl.java

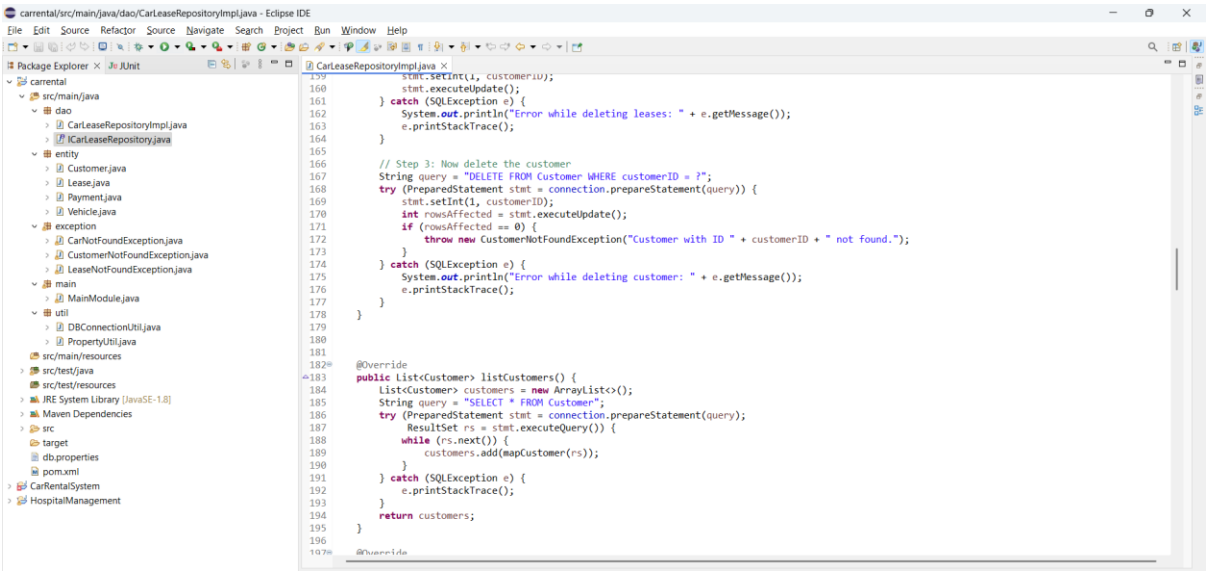


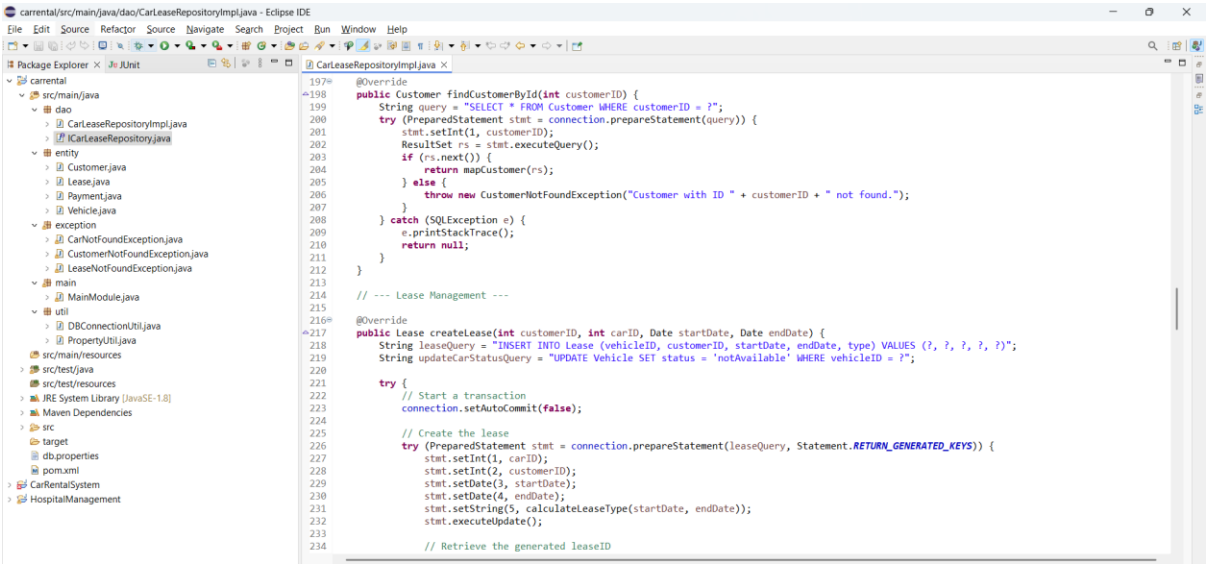
```
package dao;

import entity.Vehicle;

public class CarLeaseRepositoryImpl implements ICarLeaseRepository {

    private Connection connection;

    public CarLeaseRepositoryImpl() {
        this.connection = DBConnectionUtil.getConnection();
    }

    // --- Car Management ---

    @Override
    public void addCar(Vehicle car) {
        String query = "INSERT INTO Vehicle (make, model, year, dailyRate, status, passengerCapacity, engineCapacity) VALUES (?, ?, ?, ?, ?, ?, ?)";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            stmt.setString(1, car.getMake());
            stmt.setString(2, car.getModel());
            stmt.setInt(3, car.getYear());
            stmt.setDouble(4, car.getDailyRate());
            stmt.setString(5, car.getStatus());
            stmt.setInt(6, car.getPassengerCapacity());
            stmt.setDouble(7, car.getEngineCapacity());
            stmt.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void removeCar(int carID) {
        String query = "DELETE FROM Vehicle WHERE vehicleID = ?";
        try (PreparedStatement stmt = connection.prepareStatement(query)) {
            stmt.setInt(1, carID);
            int rowsAffected = stmt.executeUpdate();
            if (rowsAffected == 0) {
```

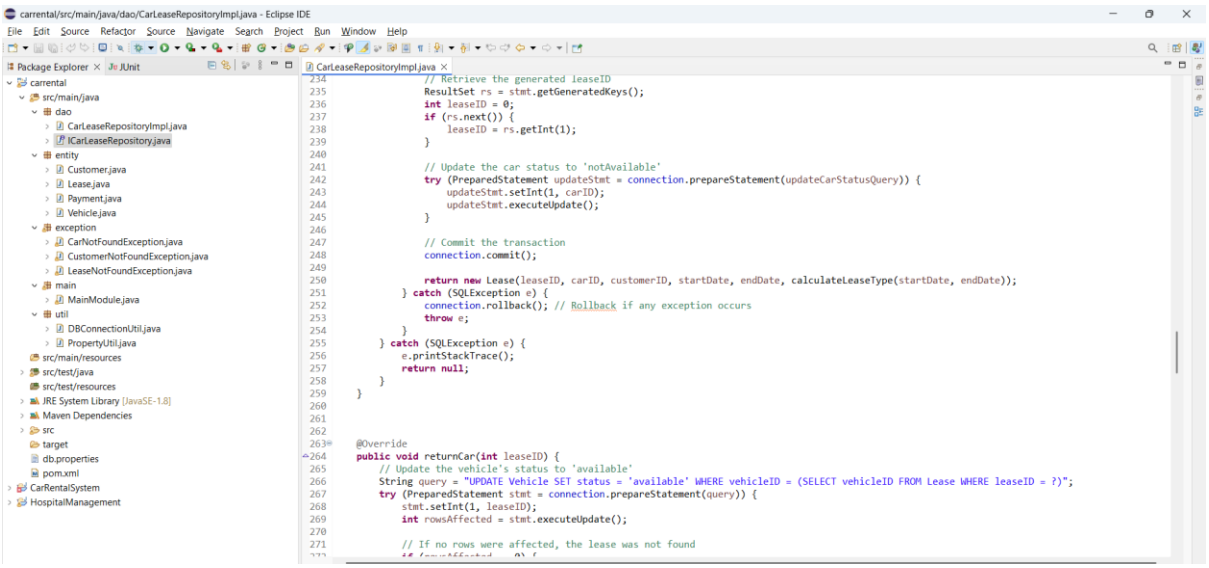

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
int rowsAffected = stmt.executeUpdate();
if (rowsAffected == 0) {
throw new CarNotFoundException("Car with ID " + carID + " not found.");
}
} catch (SQLException e) {
e.printStackTrace();
}
}
@Override
public List<Vehicle> listAvailableCars() {
List<Vehicle> cars = new ArrayList<>();
String query = "SELECT * FROM Vehicle WHERE status = 'available'";
try (PreparedStatement stmt = connection.prepareStatement(query);
ResultSet rs = stmt.executeQuery()) {
while (rs.next()) {
cars.add(mapVehicle(rs));
}
} catch (SQLException e) {
e.printStackTrace();
}
return cars;
}
@Override
public List<Vehicle> listRentedCars() {
List<Vehicle> rentedCars = new ArrayList<>();
String query = "SELECT * FROM Vehicle WHERE status = 'notAvailable'";
try (PreparedStatement stmt = connection.prepareStatement(query);
ResultSet rs = stmt.executeQuery()) {
while (rs.next()) {
rentedCars.add(new Vehicle(
rs.getInt("vehicleID"),
rs.getString("make"),
rs.getString("model"),
rs.getInt("year"),
rs.getDouble("dailyRate"),
rs.getString("status"),
rs.getInt("passengerCapacity"),

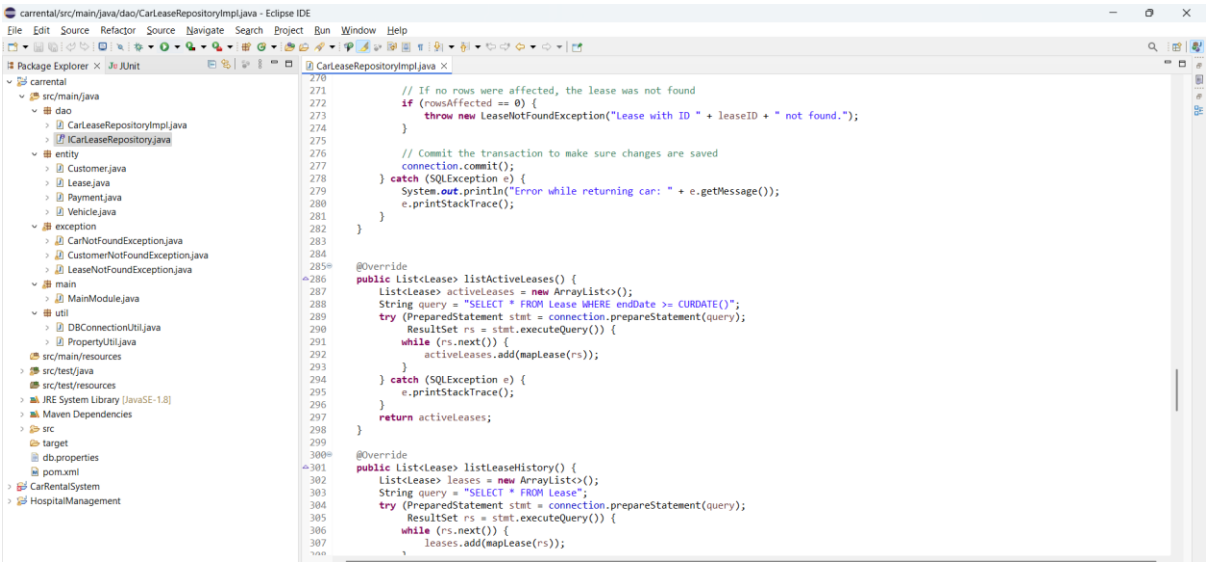

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
rs.getDouble("engineCapacity")
));
}
// Check if no cars are rented
if (rentedCars.isEmpty()) {
System.out.println("No cars are currently rented.");
}
} catch (SQLException e) {
System.err.println("Error while fetching rented cars: " + e.getMessage());
e.printStackTrace();
}
return rentedCars;
}
public Vehicle findCarById(int carID) {
String query = "SELECT * FROM Vehicle WHERE vehicleID = ?";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setInt(1, carID);
ResultSet rs = stmt.executeQuery();
if (rs.next()) {
return new Vehicle(
rs.getInt("vehicleID"),
rs.getString("make"),
rs.getString("model"),
rs.getInt("year"),
rs.getDouble("dailyRate"),
rs.getString("status"), // Ensure the latest status is fetched
rs.getInt("passengerCapacity"),
rs.getDouble("engineCapacity")
);
} else {
throw new CarNotFoundException("Car with ID " + carID + " not found.");
}
} catch (SQLException e) {
e.printStackTrace();
return null;

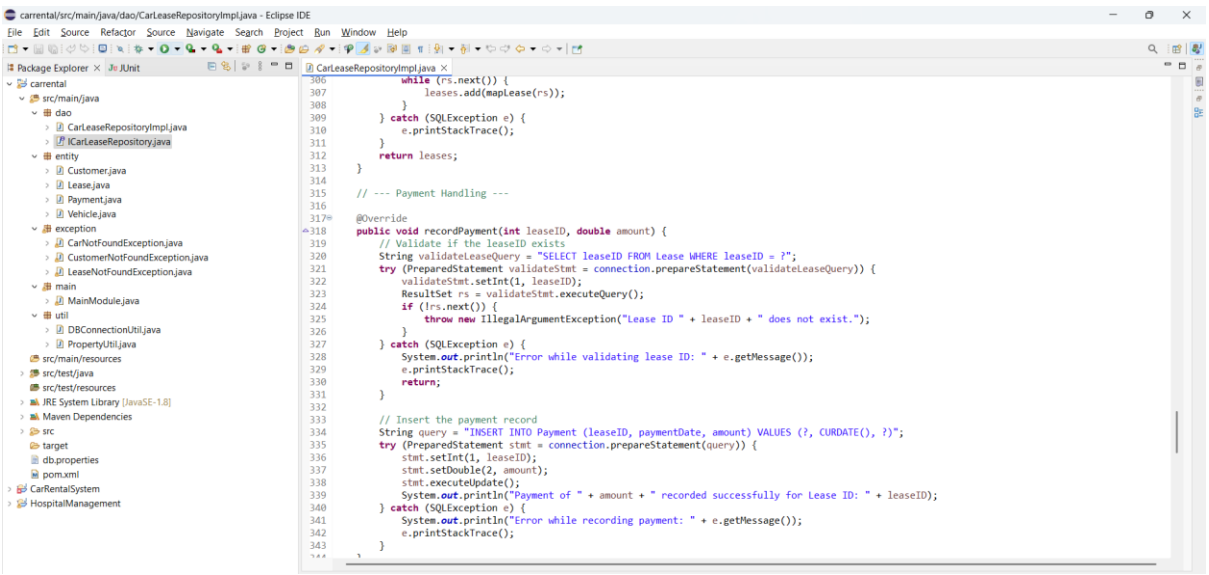

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
return null;
}
}
// --- Customer Management ---
@Override
public void addCustomer(Customer customer) {
String query = "INSERT INTO Customer (firstName, lastName, email, phoneNumber) VALUES (?, ?, ?, ?)";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setString(1, customer.getFirstName());
stmt.setString(2, customer.getLastName());
stmt.setString(3, customer.getEmail());
stmt.setString(4, customer.getPhoneNumber());
stmt.executeUpdate();
} catch (SQLException e) {
e.printStackTrace();
}
}
@Override
public void removeCustomer(int customerID) {
// Step 1: Delete the payments associated with the customer
String deletePaymentsQuery = "DELETE FROM Payment WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customerID = ?)";
try (PreparedStatement stmt = connection.prepareStatement(deletePaymentsQuery)) {
stmt.setInt(1, customerID);
stmt.executeUpdate();
} catch (SQLException e) {
System.out.println("Error while deleting payments: " + e.getMessage());
e.printStackTrace();
}
// Step 2: Delete the leases associated with the customer
String deleteLeasesQuery = "DELETE FROM Lease WHERE customerID = ?";
try (PreparedStatement stmt = connection.prepareStatement(deleteLeasesQuery)) {
stmt.setInt(1, customerID);
stmt.executeUpdate();

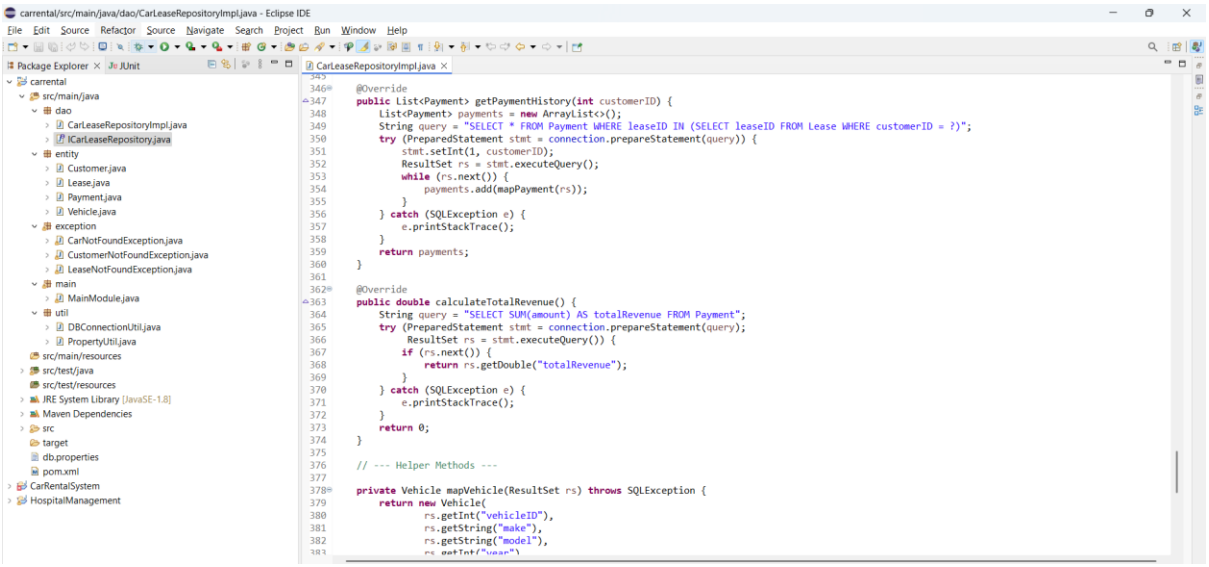

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
stmt.setInt(1, customerID);
stmt.executeUpdate();
} catch (SQLException e) {
System.out.println("Error while deleting leases: " + e.getMessage());
e.printStackTrace();
}
// Step 3: Now delete the customer
String query = "DELETE FROM Customer WHERE customerID = ?";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setInt(1, customerID);
int rowsAffected = stmt.executeUpdate();
if (rowsAffected == 0) {
throw new CustomerNotFoundException("Customer with ID " + customerID + " not found.");
}
} catch (SQLException e) {
System.out.println("Error while deleting customer: " + e.getMessage());
e.printStackTrace();
}
}
@Override
public List<Customer> listCustomers() {
List<Customer> customers = new ArrayList<>();
String query = "SELECT * FROM Customer";
try (PreparedStatement stmt = connection.prepareStatement(query);
ResultSet rs = stmt.executeQuery()) {
while (rs.next()) {
customers.add(mapCustomer(rs));
}
} catch (SQLException e) {
e.printStackTrace();
}
return customers;
}
@Override

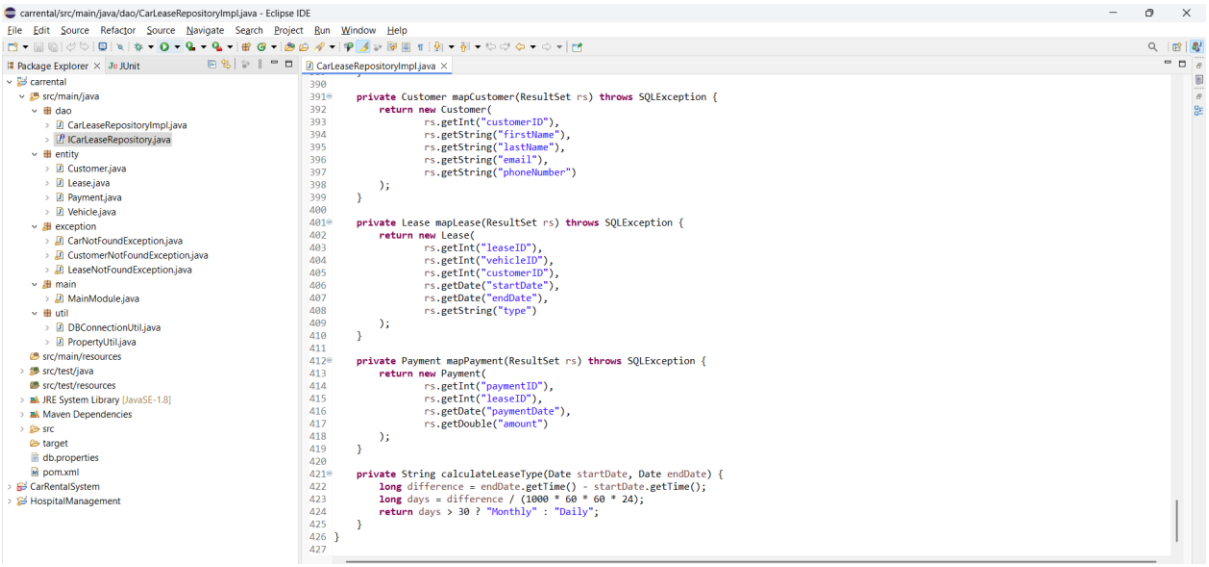

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
@Override
public Customer findCustomerById(int customerID) {
String query = "SELECT * FROM Customer WHERE customerID = ?";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setInt(1, customerID);
ResultSet rs = stmt.executeQuery();
if (rs.next()) {
return mapCustomer(rs);
} else {
throw new CustomerNotFoundException("Customer with ID " + customerID + " not found.");
}
} catch (SQLException e) {
e.printStackTrace();
return null;
}
}
// --- Lease Management ---
@Override
public Lease createLease(int customerID, int carID, Date startDate, Date endDate) {
String leaseQuery = "INSERT INTO Lease (vehicleID, customerID, startDate, endDate, type) VALUES (?, ?, ?, ?, ?)";
String updateCarStatusQuery = "UPDATE Vehicle SET status = 'notAvailable' WHERE vehicleID = ?";
try {
// Start a transaction
connection.setAutoCommit(false);
// Create the lease
try (PreparedStatement stmt = connection.prepareStatement(leaseQuery, Statement.RETURN_GENERATED_KEYS)) {
stmt.setInt(1, carID);
stmt.setInt(2, customerID);
stmt.setDate(3, startDate);
stmt.setDate(4, endDate);
stmt.setString(5, calculateLeaseType(startDate, endDate));
stmt.executeUpdate();
// Retrieve the generated leaseID


carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
// Retrieve the generated leaseID
ResultSet rs = stmt.getGeneratedKeys();
int leaseID = 0;
if (rs.next()) {
leaseID = rs.getInt(1);
}
// Update the car status to 'notAvailable'
try (PreparedStatement updateStmt = connection.prepareStatement(updateCarStatusQuery)) {
updateStmt.setInt(1, carID);
updateStmt.executeUpdate();
}
// Commit the transaction
connection.commit();
return new Lease(leaseID, carID, customerID, startDate, endDate, calculateLeaseType(startDate, endDate));
} catch (SQLException e) {
connection.rollback(); // Rollback if any exception occurs
throw e;
}
} catch (SQLException e) {
e.printStackTrace();
return null;
}
}
@Override
public void returnCar(int leaseID) {
// Update the vehicle's status to 'available'
String query = "UPDATE Vehicle SET status = 'available' WHERE vehicleID = (SELECT vehicleID FROM Lease WHERE leaseID = ?)";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setInt(1, leaseID);
int rowsAffected = stmt.executeUpdate();
// If no rows were affected, the lease was not found


carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
// If no rows were affected, the lease was not found
if (rowsAffected == 0) {
throw new LeaseNotFoundException("Lease with ID " + leaseID + " not found.");
}
// Commit the transaction to make sure changes are saved
connection.commit();
} catch (SQLException e) {
System.out.println("Error while returning car: " + e.getMessage());
e.printStackTrace();
}
}
@Override
public List<Lease> listActiveLeases() {
List<Lease> activeLeases = new ArrayList<>();
String query = "SELECT * FROM Lease WHERE endDate >= CURDATE()";
try (PreparedStatement stmt = connection.prepareStatement(query);
ResultSet rs = stmt.executeQuery()) {
while (rs.next()) {
activeLeases.add(mapLease(rs));
}
} catch (SQLException e) {
e.printStackTrace();
}
return activeLeases;
}
@Override
public List<Lease> listLeaseHistory() {
List<Lease> leases = new ArrayList<>();
String query = "SELECT * FROM Lease";
try (PreparedStatement stmt = connection.prepareStatement(query);
ResultSet rs = stmt.executeQuery()) {
while (rs.next()) {
leases.add(mapLease(rs));

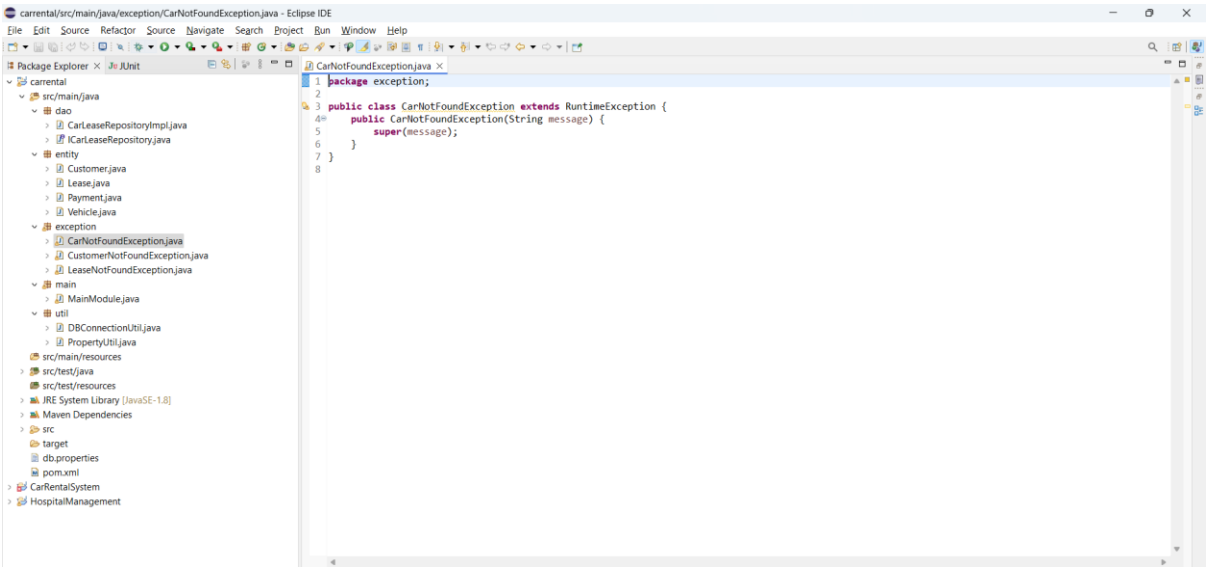

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
while (rs.next()) {
leases.add(mapLease(rs));
}
} catch (SQLException e) {
e.printStackTrace();
}
return leases;
}
// --- Payment Handling ---
@Override
public void recordPayment(int leaseID, double amount) {
// Validate if the leaseID exists
String validateLeaseQuery = "SELECT leaseID FROM Lease WHERE leaseID = ?";
try (PreparedStatement validateStmt = connection.prepareStatement(validateLeaseQuery)) {
validateStmt.setInt(1, leaseID);
ResultSet rs = validateStmt.executeQuery();
if (!rs.next()) {
throw new IllegalArgumentException("Lease ID " + leaseID + " does not exist.");
}
} catch (SQLException e) {
System.out.println("Error while validating lease ID: " + e.getMessage());
e.printStackTrace();
return;
}
// Insert the payment record
String query = "INSERT INTO Payment (leaseID, paymentDate, amount) VALUES (?, CURDATE(), ?)";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setInt(1, leaseID);
stmt.setDouble(2, amount);
stmt.executeUpdate();
System.out.println("Payment of " + amount + " recorded successfully for Lease ID: " + leaseID);
} catch (SQLException e) {
System.out.println("Error while recording payment: " + e.getMessage());
e.printStackTrace();
}


carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
@Override
public List<Payment> getPaymentHistory(int customerID) {
List<Payment> payments = new ArrayList<>();
String query = "SELECT * FROM Payment WHERE leaseID IN (SELECT leaseID FROM Lease WHERE customerID = ?)";
try (PreparedStatement stmt = connection.prepareStatement(query)) {
stmt.setInt(1, customerID);
ResultSet rs = stmt.executeQuery();
while (rs.next()) {
payments.add(mapPayment(rs));
}
} catch (SQLException e) {
e.printStackTrace();
}
return payments;
}
@Override
public double calculateTotalRevenue() {
String query = "SELECT SUM(amount) AS totalRevenue FROM Payment";
try (PreparedStatement stmt = connection.prepareStatement(query);
ResultSet rs = stmt.executeQuery()) {
if (rs.next()) {
return rs.getDouble("totalRevenue");
}
} catch (SQLException e) {
e.printStackTrace();
}
return 0;
}
// --- Helper Methods ---
private Vehicle mapVehicle(ResultSet rs) throws SQLException {
return new Vehicle(
rs.getInt("vehicleID"),
rs.getString("make"),
rs.getString("model"),

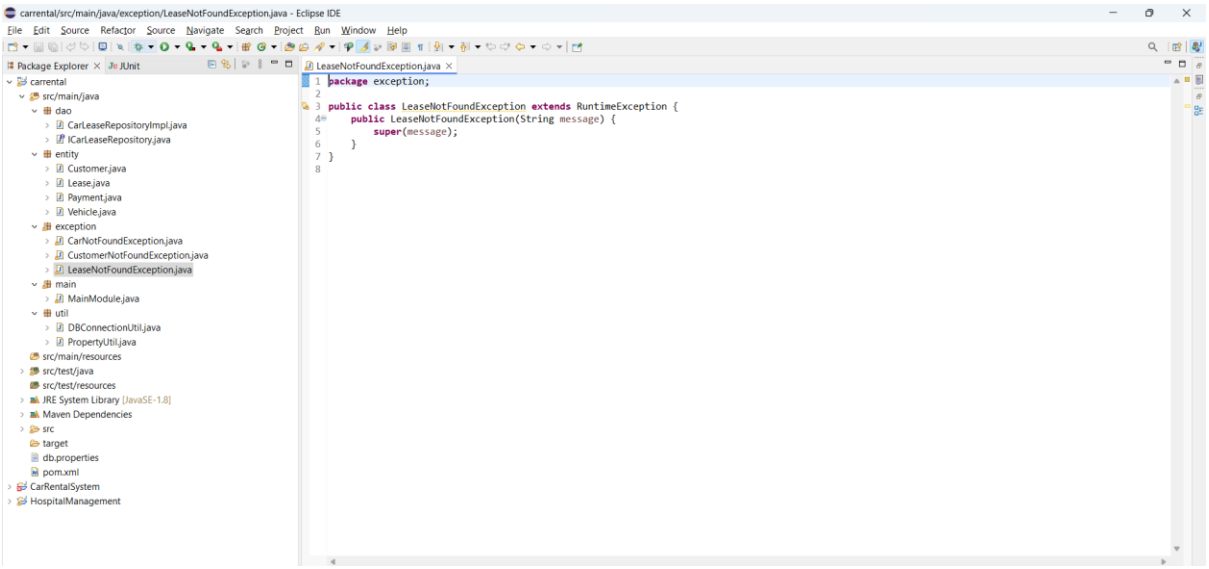

carrental/src/main/java/dao/CarLeaseRepositoryImpl.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
CarLeaseRepositoryImpl.java
private Customer mapCustomer(ResultSet rs) throws SQLException {
return new Customer(
rs.getInt("customerID"),
rs.getString("firstName"),
rs.getString("lastName"),
rs.getString("email"),
rs.getString("phoneNumber")
);
}
private Lease mapLease(ResultSet rs) throws SQLException {
return new Lease(
rs.getInt("leaseID"),
rs.getInt("vehicleID"),
rs.getInt("customerID"),
rs.getDate("startDate"),
rs.getDate("endDate"),
rs.getString("type")
);
}
private Payment mapPayment(ResultSet rs) throws SQLException {
return new Payment(
rs.getInt("paymentID"),
rs.getInt("leaseID"),
rs.getDate("paymentDate"),
rs.getDouble("amount")
);
}
private String calculateLeaseType(Date startDate, Date endDate) {
long difference = endDate.getTime() - startDate.getTime();
long days = difference / (1000 * 60 * 60 * 24);
return days > 30 ? "Monthly" : "Daily";
}
}

exception Package :

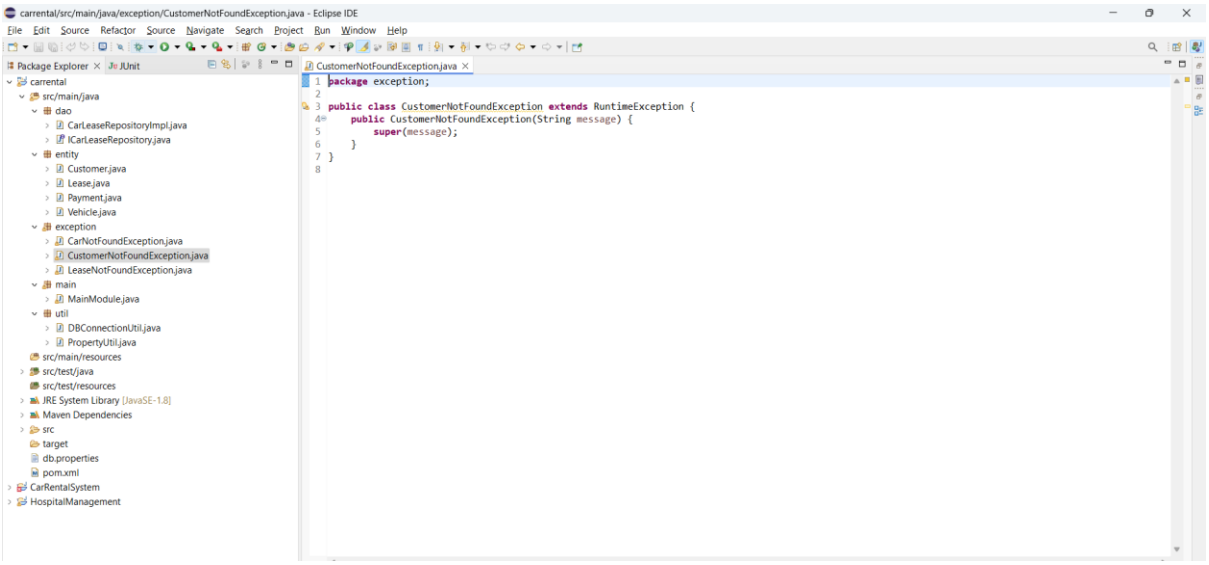
CarNotFoundException.java:



LeaseNotFoundException.java

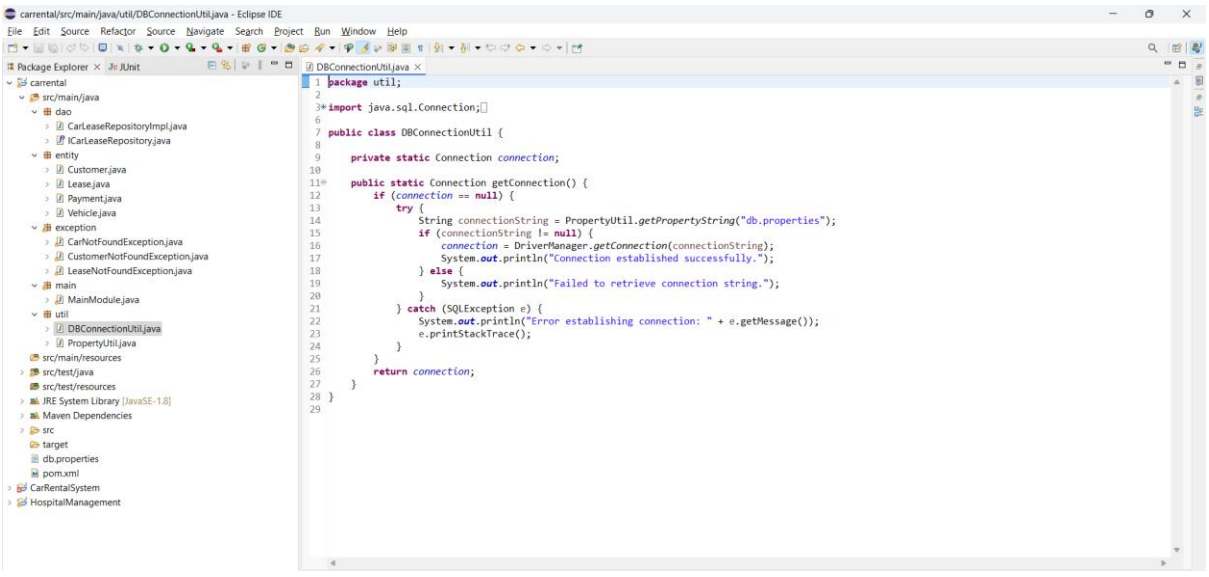


CustomerNotFoundException.java

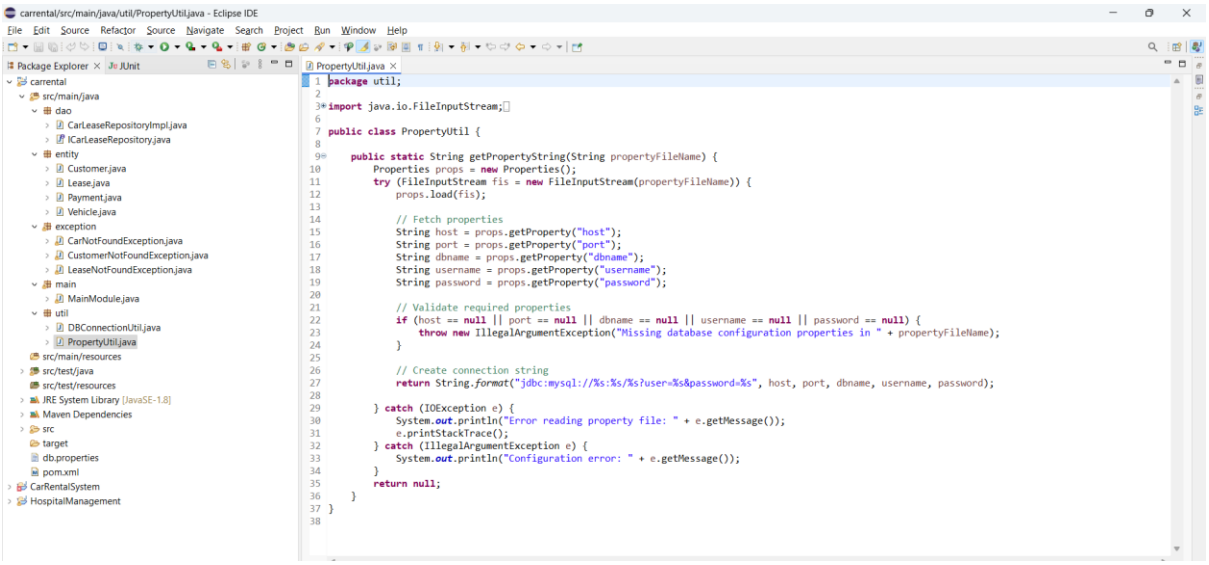


util Package:

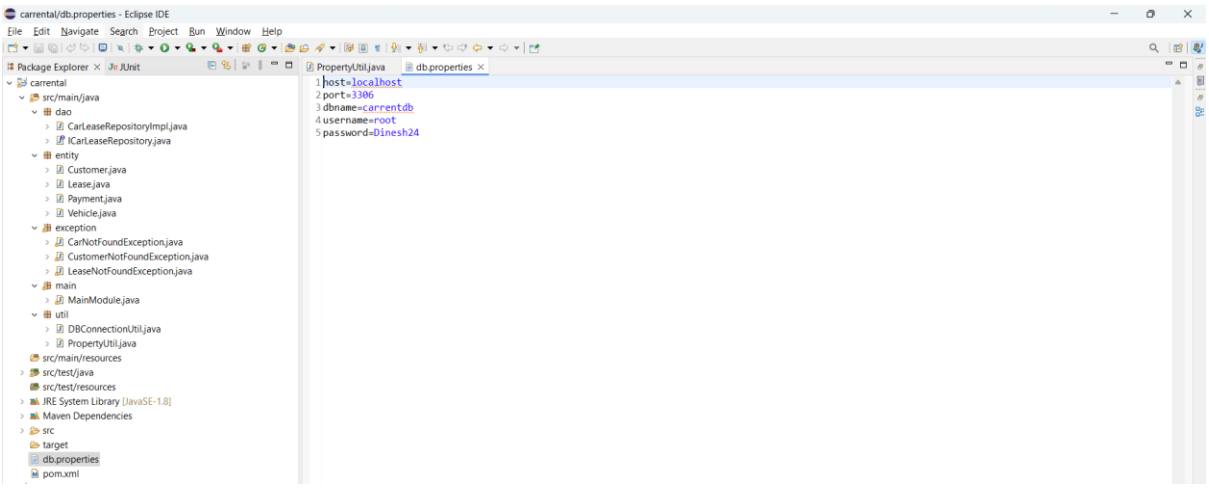
DBConnectionUtil.java:



PropertyUtil.java:

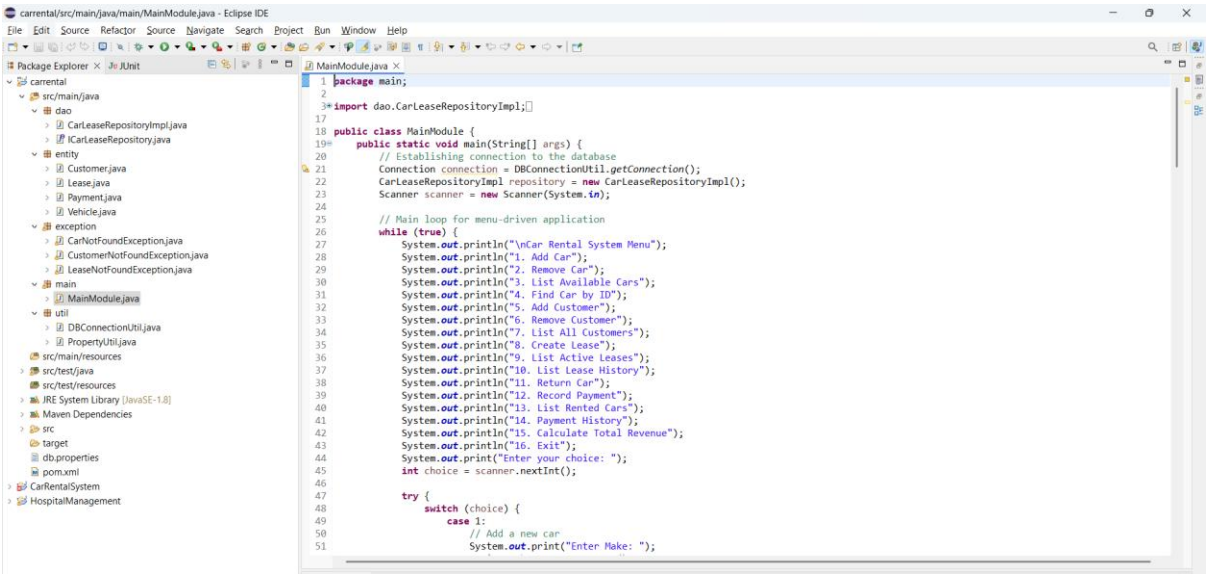


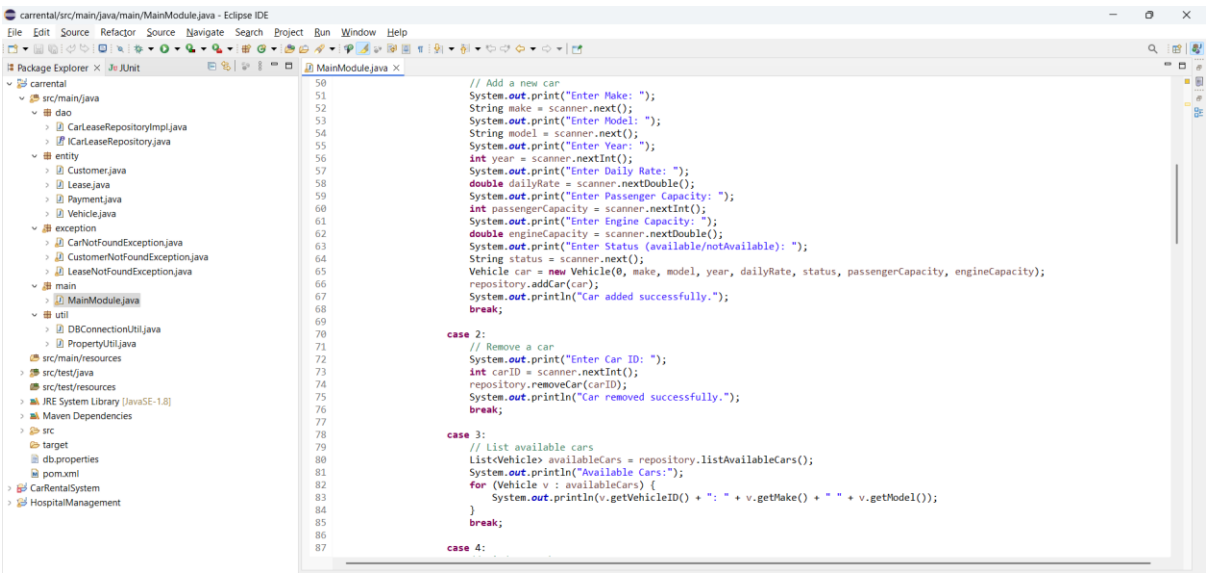
db.properties (file in root folder)

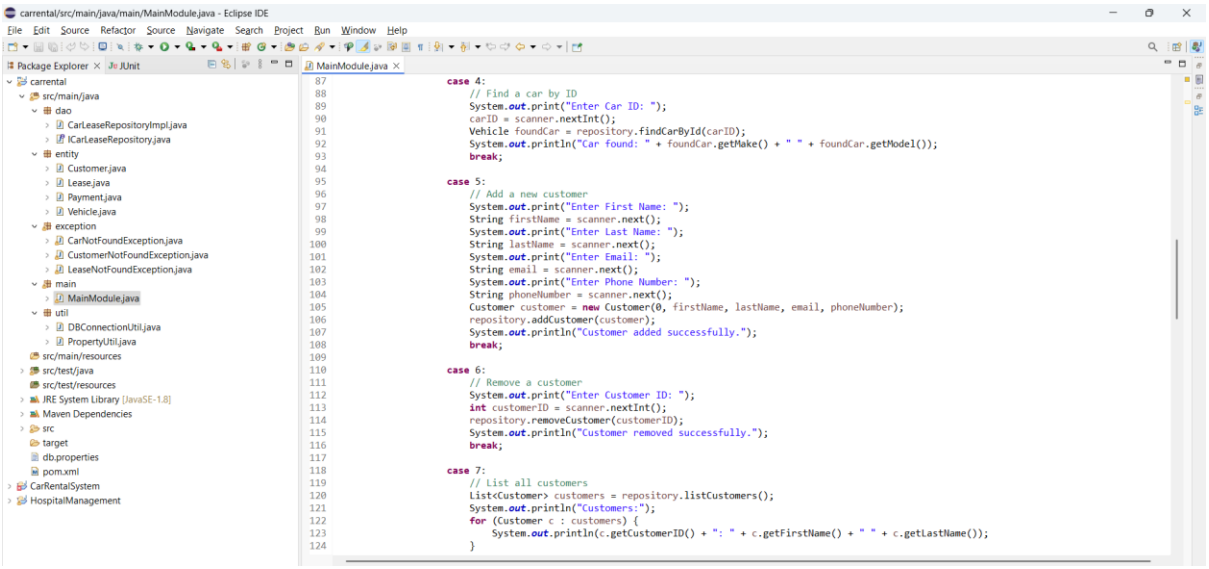


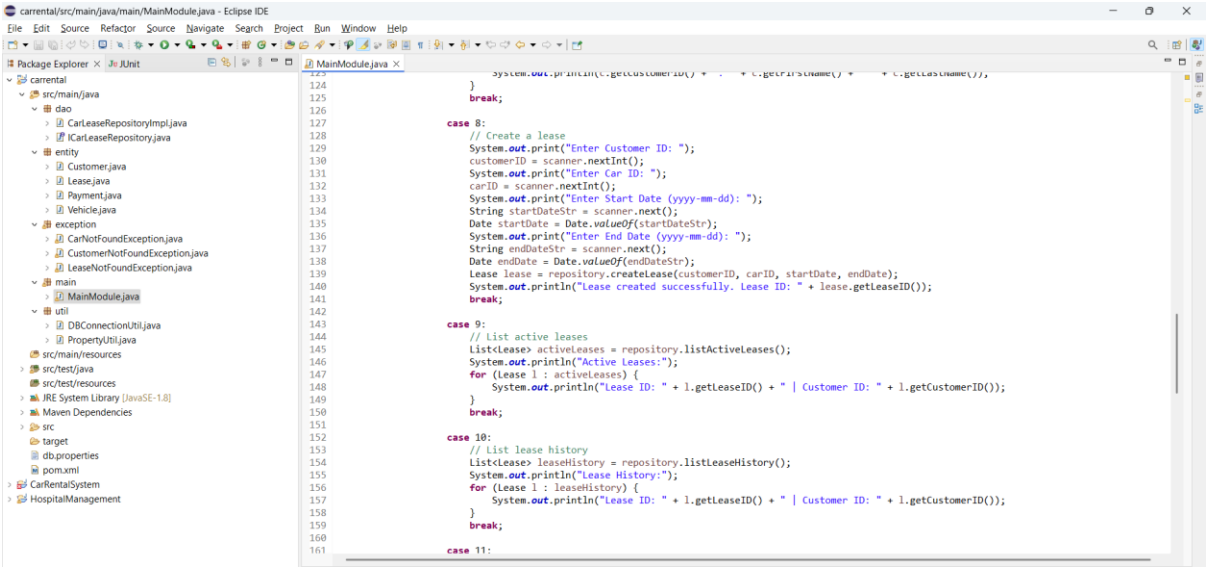
main Package:

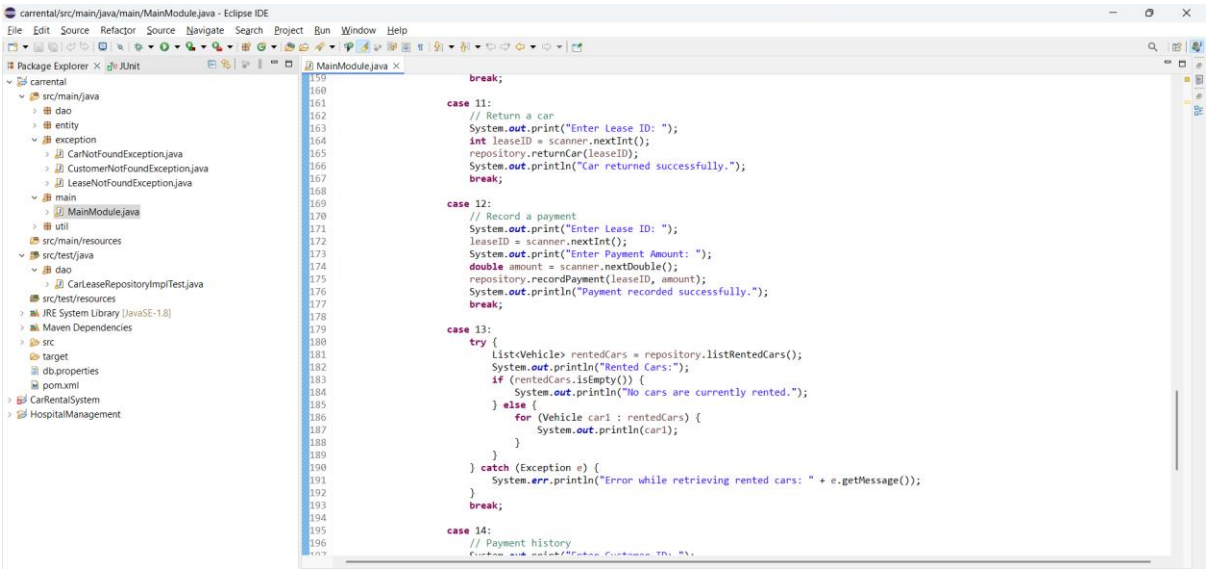
MainModule.java:

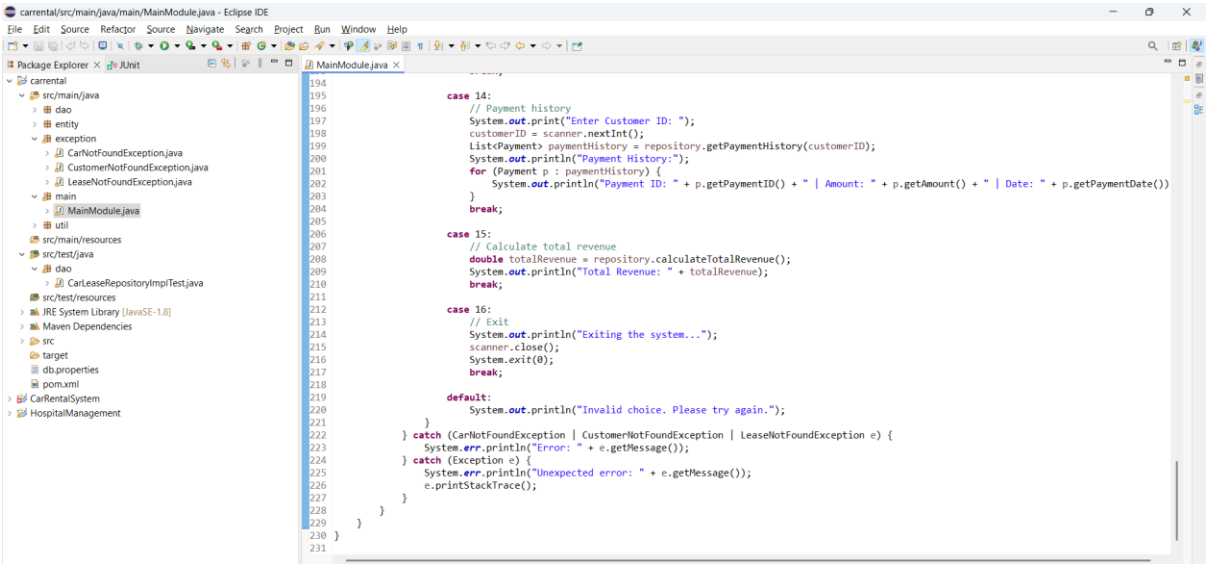





carrental/src/main/java/main/MainModule.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
MainModule.java
case 4:
// Find a car by ID
System.out.print("Enter Car ID: ");
carID = scanner.nextInt();
Vehicle foundCar = repository.findCarById(carID);
System.out.println("Car found: " + foundCar.getMake() + " " + foundCar.getModel());
break;
case 5:
// Add a new customer
System.out.print("Enter First Name: ");
String firstName = scanner.next();
System.out.print("Enter Last Name: ");
String lastName = scanner.next();
System.out.print("Enter Email: ");
String email = scanner.next();
System.out.print("Enter Phone Number: ");
String phoneNumber = scanner.next();
Customer customer = new Customer(0, firstName, lastName, email, phoneNumber);
repository.addCustomer(customer);
System.out.println("Customer added successfully.");
break;
case 6:
// Remove a customer
System.out.print("Enter Customer ID: ");
int customerID = scanner.nextInt();
repository.removeCustomer(customerID);
System.out.println("Customer removed successfully.");
break;
case 7:
// List all customers
List<Customer> customers = repository.listCustomers();
System.out.println("Customers:");
for (Customer c : customers) {
System.out.println(c.getCustomerID() + ": " + c.getFirstName() + " " + c.getLastName());
}

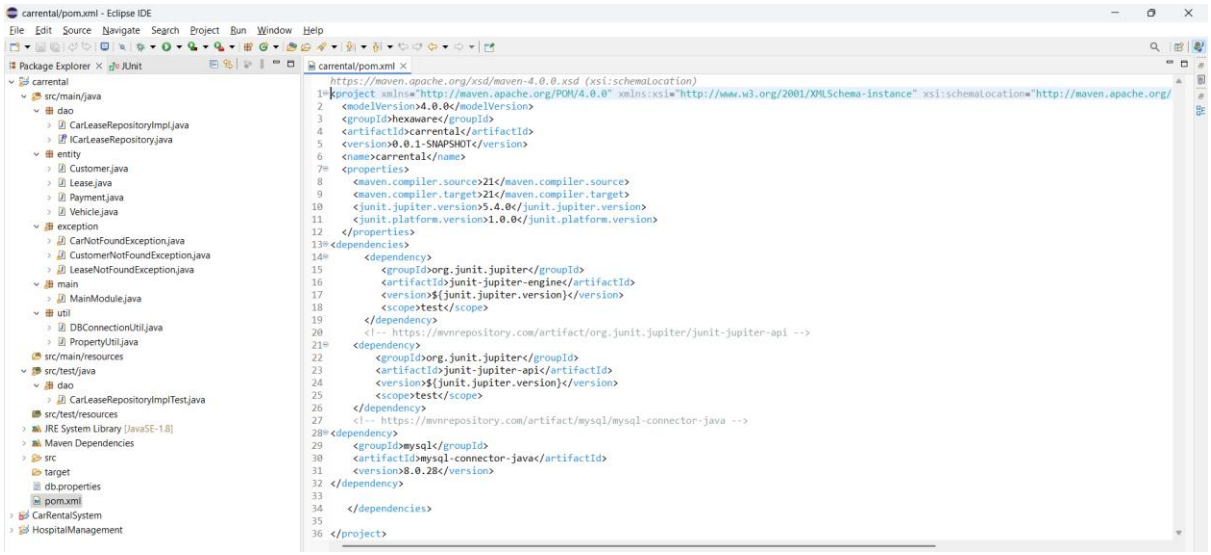

carrental/src/main/java/main/MainModule.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
CarLeaseRepositoryImpl.java
ICarLeaseRepository.java
entity
Customer.java
Lease.java
Payment.java
Vehicle.java
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
DBConnectionUtil.java
PropertyUtil.java
src/main/resources
src/test/java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
MainModule.java
}
break;
case 8:
// Create a lease
System.out.print("Enter Customer ID: ");
customerID = scanner.nextInt();
System.out.print("Enter Car ID: ");
carID = scanner.nextInt();
System.out.print("Enter Start Date (yyyy-mm-dd): ");
String startDateStr = scanner.next();
Date startDate = Date.valueOf(startDateStr);
System.out.print("Enter End Date (yyyy-mm-dd): ");
String endDateStr = scanner.next();
Date endDate = Date.valueOf(endDateStr);
Lease lease = repository.createLease(customerID, carID, startDate, endDate);
System.out.println("Lease created successfully. Lease ID: " + lease.getLeaseID());
break;
case 9:
// List active leases
List<Lease> activeLeases = repository.listActiveLeases();
System.out.println("Active Leases:");
for (Lease l : activeLeases) {
System.out.println("Lease ID: " + l.getLeaseID() + " | Customer ID: " + l.getCustomerID());
}
break;
case 10:
// List lease history
List<Lease> leaseHistory = repository.listLeaseHistory();
System.out.println("Lease History:");
for (Lease l : leaseHistory) {
System.out.println("Lease ID: " + l.getLeaseID() + " | Customer ID: " + l.getCustomerID());
}
break;
case 11:


carrental/src/main/java/main/MainModule.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
entity
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
src/main/resources
src/test/java
dao
CarLeaseRepositoryImplTest.java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
MainModule.java
break;
case 11:
// Return a car
System.out.print("Enter Lease ID: ");
int leaseID = scanner.nextInt();
repository.returnCar(leaseID);
System.out.println("Car returned successfully.");
break;
case 12:
// Record a payment
System.out.print("Enter Lease ID: ");
leaseID = scanner.nextInt();
System.out.print("Enter Payment Amount: ");
double amount = scanner.nextDouble();
repository.recordPayment(leaseID, amount);
System.out.println("Payment recorded successfully.");
break;
case 13:
try {
List<Vehicle> rentedCars = repository.listRentedCars();
System.out.println("Rented Cars:");
if (rentedCars.isEmpty()) {
System.out.println("No cars are currently rented.");
} else {
for (Vehicle car1 : rentedCars) {
System.out.println(car1);
}
}
} catch (Exception e) {
System.err.println("Error while retrieving rented cars: " + e.getMessage());
}
break;
case 14:
// Payment history


carrental/src/main/java/main/MainModule.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Package Explorer JUnit
carrental
src/main/java
dao
entity
exception
CarNotFoundException.java
CustomerNotFoundException.java
LeaseNotFoundException.java
main
MainModule.java
util
src/main/resources
src/test/java
dao
CarLeaseRepositoryImplTest.java
src/test/resources
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
db.properties
pom.xml
CarRentalSystem
HospitalManagement
MainModule.java
case 14:
// Payment history
System.out.print("Enter Customer ID: ");
customerID = scanner.nextInt();
List<Payment> paymentHistory = repository.getPaymentHistory(customerID);
System.out.println("Payment History:");
for (Payment p : paymentHistory) {
System.out.println("Payment ID: " + p.getPaymentID() + " | Amount: " + p.getAmount() + " | Date: " + p.getPaymentDate())
}
break;
case 15:
// Calculate total revenue
double totalRevenue = repository.calculateTotalRevenue();
System.out.println("Total Revenue: " + totalRevenue);
break;
case 16:
// Exit
System.out.println("Exiting the system...");
scanner.close();
System.exit(0);
break;
default:
System.out.println("Invalid choice. Please try again.");
}
} catch (CarNotFoundException | CustomerNotFoundException | LeaseNotFoundException e) {
System.err.println("Error: " + e.getMessage());
} catch (Exception e) {
System.err.println("Unexpected error: " + e.getMessage());
e.printStackTrace();
}
}
}
}

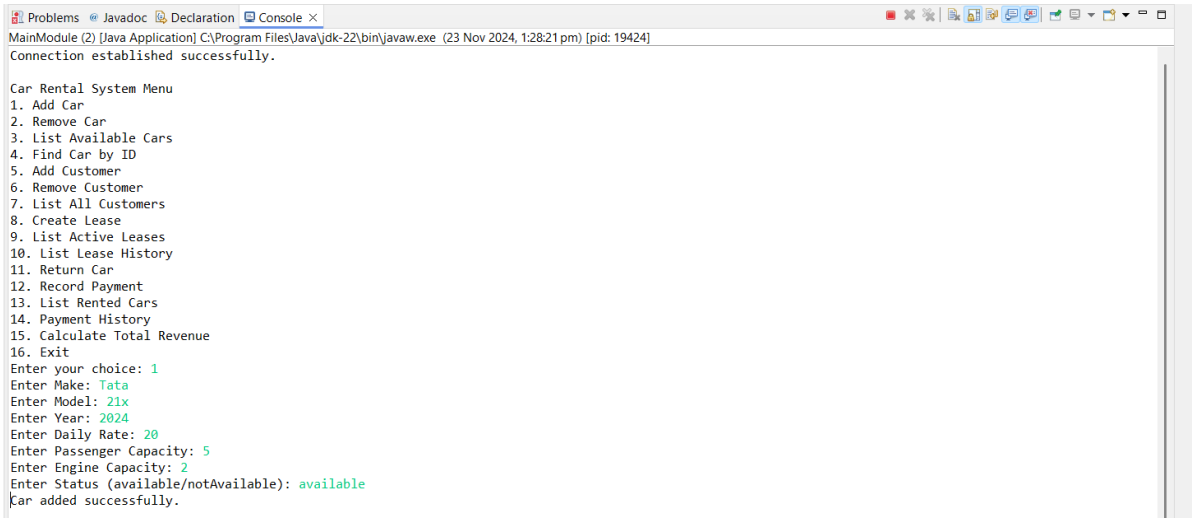
Pom.xml file for dependencies:

Pom.xml:

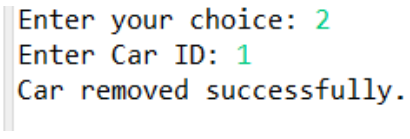


Output (By running MainModule.java):

1)



2)



3)

```
Enter your choice: 3
Available Cars:
2: Honda Civic
9: Tata 21x
```

4)

```
Enter your choice: 4
Enter Car ID: 2
Car found: Honda Civic
```

5)

```
Enter your choice: 5
Enter First Name: Dinesh
Enter Last Name: Saravanan
Enter Email: saravana1974dp@gmail.com
Enter Phone Number: 6369505351
Customer added successfully.
```

6)

```
Enter your choice: 6
Enter Customer ID: 1
Customer removed successfully.
```

7)

```
Enter your choice: 7
Customers:
2: Jane Smith
9: Dinesh Saravanan
```

8)

```
Enter your choice: 8
Enter Customer ID: 9
Enter Car ID: 2
Enter Start Date (yyyy-mm-dd): 2024-12-25
Enter End Date (yyyy-mm-dd): 2024-12-30
Lease created successfully. Lease ID: 56
```

9)

```
Enter your choice: 9
Active Leases:
Lease ID: 55 | Customer ID: 2
Lease ID: 56 | Customer ID: 9
```

10)

```
Enter your choice: 10
Lease History:
Lease ID: 55 | Customer ID: 2
Lease ID: 56 | Customer ID: 9
```

11)

```
Enter your choice: 11
Enter Lease ID: 56
Car returned successfully.
```

12)

```
Enter your choice: 12
Enter Lease ID: 56
Enter Payment Amount: 500
Payment of 500.0 recorded successfully for Lease ID: 56
Payment recorded successfully.
```

13)

```
Enter your choice: 13
No cars are currently rented.
Rented Cars:
No cars are currently rented.
```

14)

```
Enter your choice: 14
Enter Customer ID: 9
Payment History:
Payment ID: 58 | Amount: 500.0 | Date: 2024-11-23
```

15)

```
Enter your choice: 15
Total Revenue: 500.0
```

Exception Handling:

Throwing Exception where Customer Id , Car Id , Lease Id are invalid.

1)Customer Id Invalid-

```
Enter your choice: 6
Enter Customer ID: 6669
Error: Customer with ID 6669 not found.
```

2)Lease Id Invalid-

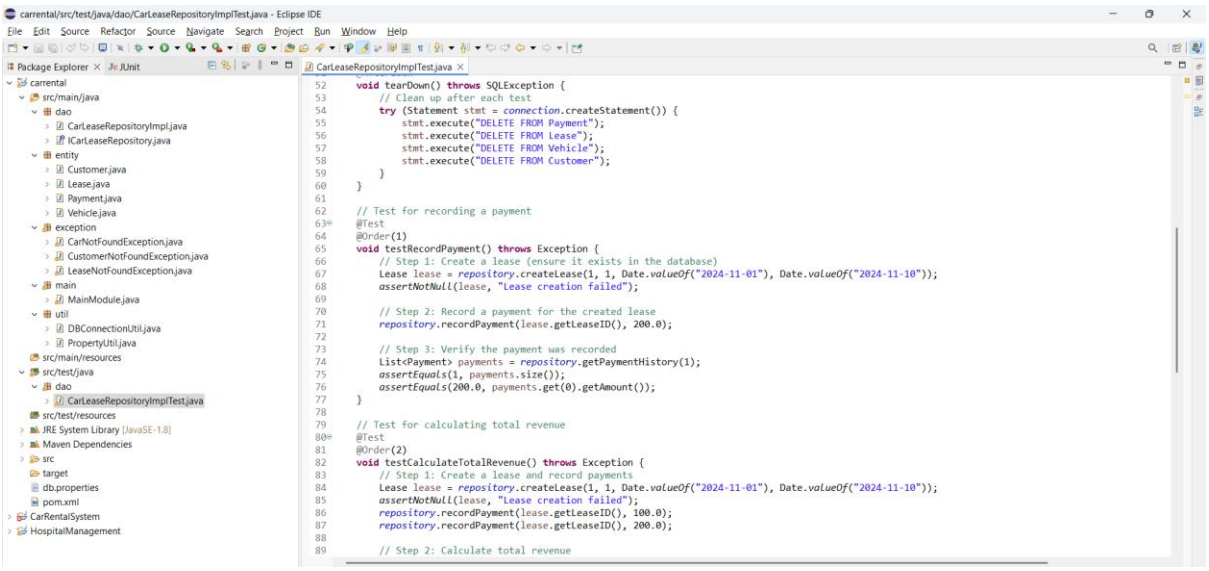
```
Enter your choice: 11
Enter Lease ID: 10
Error: Lease with ID 10 not found.
```

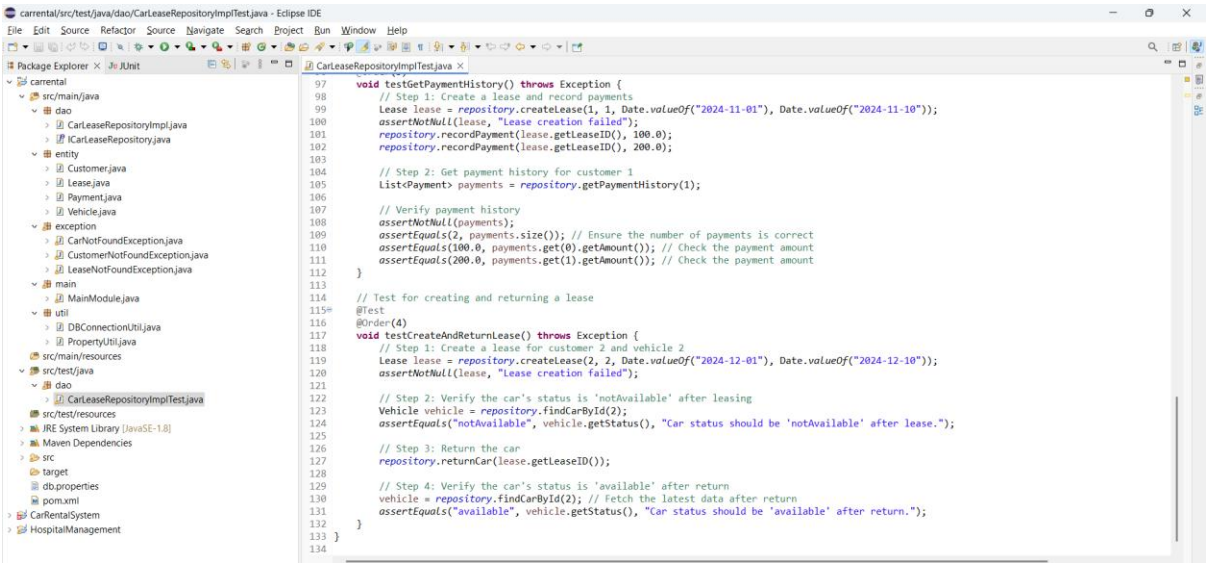
3)Car Id Invalid-

```
Enter your choice: 4
Enter Car ID: 555
Error: Car with ID 555 not found.
```

Junit test case for CarLeaseRepositoryImpl:

CarLeaseRepositoryImplTest.java:





Test Case 4/4 Passed:

Test Cases:

1) testRecordPayment() - Test for recording a payment.
2) testCalculateTotalRevenue() - Test for calculating total revenue.
3) testGetPaymentHistory() - Test for getting payment history.
4) testCreateAndReturnLease() - Test for creating and returning a lease.