DOCUMENTATION

 # Car-Rental-System

 # Documentation

Car Class:

This class represents a car in the rental system.

Its associate properties are: make, model, registrationNumber, and isRented.

The Car class provides methods to rent and return the car, as well as getter and setter methods to access the car's details.

The toString() method is overridden to provide a string representation of the car object.

The CarTest class provides unit tests to ensure the functionality of the Car class.

Customer Class:

This class represents a customer in the rental system.

Its associate properties are: name and driverLicenseNumber.

The Customer class provides getter methods to access the customer's details.

The toString() method is overridden to give a string representation of the customer object.

The CustomerTest class provides unit tests to ensure the functionality of the Customer class.

RentalAgency Class:

This class shows the rental agency that aids in managing the car rental system.

It consists of two lists namely: cars and customers, which store the cars and customers.

The RentalAgency class provides methods to add cars and customers, find a car by its registration number, rent and return cars, and display the list of cars and customers.

The rentCar() and returnCar() methods deal with the rental and return operations.

The displayCars() and displayCustomers() methods gives a way to print the list of cars and customers.

The RentalAgencyTest class provides unit tests to ensure the functioning of the RentalAgency class.

Main Class:

This class shows Car Rental System usage by creating a RentalAgency instance, adding cars and customers, renting and returning cars, and displaying the updated state of the system.

The provided solution implements the basic functionality of a Car Rental System using OOP principles in Java. It includes the necessary classes, methods, and test cases to ensure the system's correct behavior. The Car, Customer, and RentalAgency classes work together to manage the rental operations, and the test cases validate the implementation.

Making classes for Car, Customer, and Rental Agency.

Car Class

```
public class Car {
    private String make;
    private String model;
    private String registrationNumber;
    private boolean isRented;
    public Car(String make, String model, String
registrationNumber) {
        this.make = make;
        this.model = model;
        this.registrationNumber = registrationNumber;
        this.isRented = false; // Initially, the car is not
rented
    }
public String getMake() {
        return make;
    }
 public String getModel() {
     return model;
    }
public String getRegistrationNumber() {
        return registrationNumber;
    }
 public boolean isRented() {
```

```
        return isRented;
    }
 public void rentCar() {
        if (!isRented) {
            isRented = true;
        }
    }
 public void returnCar() {
        if (isRented) {
            isRented = false;
        }
    }
 @Override
    public String toString() {
        return "Car{" +
                "make='" + make + '\'' +
                ", model='" + model + '\'' +
                ", registrationNumber='" + registrationNumber +
'\'' +
                ", isRented=" + isRented +
                '}';
    }
}
```

Test Case

CarTest,java

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CarTest {
@Test
    public void testRentCar() {
```

```
        Car car = new Car("Toyota", "Camry", "ABC123");
        assertFalse(car.isRented());
        car.rentCar();
        assertTrue(car.isRented());
    }
  @Test
    public void testReturnCar() {
        Car car = new Car("Toyota", "Camry", "ABC123");
        car.rentCar();
        assertTrue(car.isRented());
        car.returnCar();
        assertFalse(car.isRented());
  }
}
```

Representing customer class

```
public class Customer {
    private String name;
    private String driverLicenseNumber;
   public Customer(String name, String driverLicenseNumber)
  this.name = name;
        this.driverLicenseNumber = driverLicenseNumber;
    }
 public String getName() {
        return name;
    }
 public String getDriverLicenseNumber() {
        return driverLicenseNumber;
    }
 @Override
    public String toString() {
        return "Customer{" +
```

```
        "name='" + name + '\'' +
                ", driverLicenseNumber='" + driverLicenseNumber
+ '\'' +
                '}';
    }
}
```

Test Case

CustomerTest.java

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class CustomerTest {

    @Test
    public void testCustomerCreation() {
        Customer customer = new Customer("John Doe", "DL12345");
        assertEquals("John Doe", customer.getName());
        assertEquals("DL12345",
customer.getDriverLicenseNumber());
  }
}
```

Rental agency class

Particularly for managing rental transactions,list of customers and cars.

```
import java.util.ArrayList;
import java.util.List;
public class RentalAgency {
    private List<Car> cars;
    private List<Customer> customers;
 public RentalAgency() {
    cars = new ArrayList<>();
        customers = new ArrayList<>();
```

```
    }
  public void addCar(Car car) {
        cars.add(car);
    }

    public void addCustomer(Customer customer) {
        customers.add(customer);
    }
 public Car findCarByRegistrationNumber(String
registrationNumber) {
        for (Car car : cars) {
            if
(car.getRegistrationNumber().equals(registrationNumber)) {
                return car;
       }
        }
        return null;
    }
 public void rentCar(String registrationNumber) {
        Car car =
findCarByRegistrationNumber(registrationNumber);
        if (car != null && !car.isRented()) {
            car.rentCar();
            System.out.println("Car rented successfully.");
        } else {
            System.out.println("Car is not available for
rent.");
   }
  }
public void returnCar(String registrationNumber) {
        Car car =
findCarByRegistrationNumber(registrationNumber);
```

```
        if (car != null && car.isRented()) {
            car.returnCar();
            System.out.println("Car returned successfully.");
        } else {
            System.out.println("Car was not rented.");
        }
    }
public void displayCars() {
        for (Car car : cars) {
            System.out.println(car);
      }
    }

    public void displayCustomers() {
        for (Customer customer : customers) {
            System.out.println(customer);
        }
    }
}
```

Test Case

RentalAgencyTest.java

```
import org.junit.jupiter.api.Test;
import java.util.List;
import static org.junit.jupiter.api.Assertions.*;

public class RentalAgencyTest {

    @Test
    public void testAddCar() {
        RentalAgency rentalAgency = new RentalAgency();
        Car car = new Car("Toyota", "Camry", "ABC123");
```

```
        rentalAgency.addCar(car);
     List<Car> cars = rentalAgency.getCars();
        assertEquals(1, cars.size());
        assertEquals(car, cars.get(0));
    }

    @Test
    public void testAddCustomer() {
        RentalAgency rentalAgency = new RentalAgency();
        Customer customer = new Customer("John Doe", "DL12345");
        rentalAgency.addCustomer(customer);
        List<Customer> customers = rentalAgency.getCustomers();
        assertEquals(1, customers.size());
        assertEquals(customer, customers.get(0));
    }
}
```

Main Class

A demonstration of the functionalities of the car system

```
public class Main {
    public static void main(String[] args) {
        RentalAgency rentalAgency = new RentalAgency();
    // Add some cars to the agency
        rentalAgency.addCar(new Car("Toyota", "Camry",
"ABC123"));
        rentalAgency.addCar(new Car("Honda", "Civic",
"XYZ456"));
    // Add some customers to the agency
        rentalAgency.addCustomer(new Customer("John Doe",
"DL12345"));
        rentalAgency.addCustomer(new Customer("Jane Smith",
"DL67890"));
```

```
    // Display all cars
        System.out.println("Cars available in the agency:");
        rentalAgency.displayCars();
    // Display all customers
        System.out.println("\nCustomers registered in the
agency:");
        rentalAgency.displayCustomers();
    // Rent a car
        System.out.println("\nRenting a car:");
        rentalAgency.rentCar("ABC123");
    // Try to rent the same car again
        System.out.println("\nTrying to rent the same car
again:");
        rentalAgency.rentCar("ABC123");
    // Return the car
        System.out.println("\nReturning the car:");
        rentalAgency.returnCar("ABC123");
    // Try to return the car again
        System.out.println("\nTrying to return the same car
again:");
        rentalAgency.returnCar("ABC123");
    // Display all cars again to see the updated status
        System.out.println("\nUpdated list of cars:");
        rentalAgency.displayCars();
    }
}
```