**Car Rental System Documentation**

**Introduction:**

This Car Rental System is developed using Object-Oriented Programming (OOP) principles in Java. The system manages car rentals, customer information, and rental transactions. It consists of three main classes: `Car`, `Customer`, and `RentalAgency`. The project is designed to encapsulate data and behaviors for a modular and maintainable system.

**Classes**

***Car Class***

Attributes:

- licensePlate (String): The unique identifier for the car.

- model (String): The model of the car.

- isRented (boolean): Indicates whether the car is currently rented.

***Methods:***

- Car (String licensePlate, String model): Constructor to initialize a new car object.

- String getLicensePlate(): Returns the license plate of the car.

- String getModel(): Returns the model of the car.

- boolean isRented(): Returns whether the car is currently rented.

- void rentCar(): Marks the car as rented.

- void returnCar(): Marks the car as returned.

**Customer Class**

***Attributes:***

- id (String): The unique identifier for the customer.

- name (String): The name of the customer.

***Methods:***

- Customer (String id, String name): Constructor to initialize a new customer object.

- String getId(): Returns the ID of the customer.

- String getName(): Returns the name of the customer.

**RentalAgency Class**

***Attributes:***

- cars (ArrayList<Car>): A list of cars available in the rental agency.

- customers (ArrayList<Customer>): A list of customers registered with the rental agency.

***Methods:***

- RentalAgency(): Constructor to initialize a new rental agency.

- void addCar(Car car): Adds a new car to the rental agency.

- void addCustomer(Customer customer): Adds a new customer to the rental agency.

- Car findCarByLicensePlate(String licensePlate): Finds a car by its license plate.

- Customer findCustomerById(String id): Finds a customer by their ID.

- boolean rentCar(String licensePlate, String customerId): Rents a car to a customer if the car is available.

- boolean returnCar(String licensePlate): Returns a rented car.

**Usage Instructions**

1***. Initialize the Rental Agency:***

RentalAgency agency = new RentalAgency();

2. ***Add Cars to the Agency***:

Car car1 = new Car("123ABC", "Toyota");

Car car2 = new Car("456DEF", "Honda");

agency.addCar(car1);

agency.addCar(car2);

3***. Add Customers to the Agency:***

Customer customer1 = new Customer("001", "John Doe");

Customer customer2 = new Customer("002", "Jane Smith");

agency.addCustomer(customer1);

agency.addCustomer(customer2);

4. ***Rent a Car:***

boolean rentalSuccess = agency.rentCar("123ABC", "001");

if (rentalSuccess) {

System.out.println("Car rented successfully.");

} else {

System.out.println("Car rental failed.");

}

5***. Return a Car:***

boolean returnSuccess = agency.returnCar("123ABC");

if (returnSuccess) {

System.out.println("Car returned successfully.");

} else {

System.out.println("Car return failed.");

}

**Running Tests**

The project includes a suite of test cases to validate the functionality of the system. The tests are written using JUnit.

***Example Test Case***

***Test Case: Car Rental and Return***

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class RentalAgencyTest {

@Test

public void testCarRentalAndReturn() {

RentalAgency agency = new RentalAgency();

Car car = new Car("123ABC", "Toyota");

Customer customer = new Customer("001", "Abukar Hussein");

agency.addCar(car);

agency.addCustomer(customer);

assertTrue(agency.rentCar("123ABC", "001"));

assertTrue(car.isRented());

assertFalse(agency.rentCar("123ABC", "001"));

assertTrue(agency.returnCar("123ABC"));

assertFalse(car.isRented());

}

}