# CAR RENTAL SYSTEM

A PROJECT REPORT

*Submitted by*

**Dhakad Rishabh (20220702033)**

**Umika Patel (20220702073)**

Under the guidance of

**Ms.Sangeeta Giri**

*In partial fulfillment of the award of the degree of*

**B.Tech( Computer Science and Engineering / Computer Science and Engineering with specialization in Artificial Intelligence and Machine Learning)**
**( Batch 2022-2026)**

Unitedworld Institute of Technology

KARNAVATI
UNIVERSITY

**APRIL 2024**

# 🚙CAR RENTAL SYSTEM JAVA OOPS PROJECT

**2 April 2024**

🚀 *Rent a Car: Experience the ease of renting cars through an interactive console.*
🔄 *Return a Car: Effortlessly return previously rented cars and update availability.*
👥 *Customer Management: Add new customers and maintain customer records.*
🚙 *Car Management: Manage cars, brands, models, and pricing details.*
📝 *Rental History: Keep track of rentals, customers, and rental durations.*

## INTRODUCTION

### 1. Project background and description

✓ *The above project represents a basic Java application for managing a car rental system. It encapsulates the core functionalities required for renting out cars, including maintaining a database of cars, customers, and rental transactions. The system is designed to handle operations such as adding cars and customers to the system, renting available cars, and returning them after the rental period ends.*

*This Java project simplifies car rental management by implementing classes for cars, customers, and rentals, and a CarRentalSystem class for interface. It allows users to select, return, and calculate rental costs using a per-day pricing model.*

### 2. Project scope

✓ *The scope of this project encompasses developing a foundational system for car rental operations, suitable for small to medium-sized rental services. It aims to demonstrate object-oriented programming principles, encapsulation of data, and interaction between different objects within a system. While the current implementation focuses on core functionalities through a console interface, the project lays the groundwork for further expansion, such as adding more complex pricing strategies, integrating a database for persistent storage, or evolving into a web-based application for wider accessibility.*

### 3. About Project

✓ *The provided Java project is a simple car rental system consisting of classes such as Car, Customer, Rental, and CarRentalSystem. The Car class represents a car with attributes like carId, brand, model, basePricePerDay, and availability status. Customer class holds information about the customer. Rental class manages the rental details of a car to a customer for a certain number of days. CarRentalSystem class orchestrates the operations like adding cars, customers, renting, and returning cars through a menu-driven interface. The main method initializes the system with some sample cars and launches the menu for user interaction.*

## 4. Project Flow Chart

- *Initialization: The program initializes a car rental system with empty lists for cars, customers, and rentals.*
- *Adding Cars and Customers: It allows adding cars and customers to the system with unique identifiers.*
- *Renting a Car: Customers can rent available cars by providing their name, selecting a car from the available list, and specifying the rental duration.*
- *Calculating Rental Cost: The system calculates the rental cost based on the selected car's base price per day and the rental duration.*
- *Confirmation: After confirming the rental, the system updates the car's availability status and records the rental information.*
- *Returning a Car: Customers can return rented cars by providing the car's ID. The system checks if the car is rented and updates its availability.*
- *Thank You Message: Upon exiting the program, a thank you message is displayed.*

## 5. Project Roles of Different Objects & Classes

*Car: Represents a vehicle available for rental. It stores details such as car ID, brand, model, base price per day, and availability status. Functionalities include calculating rental price, marking as rented, and returning the car to availability.*

*Customer: Represents an individual renting a car. It stores customer ID and name. No specific functionalities beyond basic data storage and retrieval.*

*Rental: Facilitates the connection between a rented car and a customer. It stores the rented car, customer, and rental duration. Allows retrieval of relevant information for rental transactions.*

*CarRentalSystem: Orchestrates the interaction between cars, customers, and rentals. It provides functionalities to add cars and customers, rent and return cars, and a menu system for user interaction. It manages the overall workflow of the car rental process.*

## 6. Responsibility of Specific Classes

- *Car Class: Defines attributes and methods for a car object, including car ID, brand, model, base price per day, and rental status.*

- *Customer Class: Represents a customer with attributes such as customer ID and name.*

- *Rental Class: Manages the rental process, associating a car with a customer and specifying the rental duration.*

- *CarRentalSystem Class: Orchestrates the car rental system, including functionalities like adding cars and customers, renting and returning cars, and displaying a menu for user interaction.*

- *Main Class: Entry point of the application, creates a CarRentalSystem instance and initializes it with sample cars, then launches the system menu for user interaction.*

## 7. Future Scope

✓ *This Car Rental System is a basic implementation that allows customers to rent and return cars. However, there are several areas where it can be expanded and improved for future development:*

*User Interface Enhancement: The current system operates via console input and output. Future iterations could incorporate a graphical user interface (GUI) to enhance user experience, making it more intuitive and visually appealing.*

*Database Integration: Integrate a database system (e.g., MySQL, MongoDB) to store information persistently. This would enable storing car, customer, and rental data, providing better management and scalability.*

*Advanced Booking System: Implement features for advanced booking, allowing customers to reserve cars for specific dates in advance. This could involve handling conflicts, managing deposits, and sending confirmation emails.*

*Payment Integration: Incorporate a payment gateway to facilitate online payments for rentals. This would require integrating with payment APIs (e.g., PayPal, Stripe) to handle transactions securely.*

*User Authentication and Authorization: Add user authentication and authorization mechanisms to secure access to the system. This ensures that only authorized users can rent or return cars and access specific functionalities*

*Reporting and Analytics: Develop reporting and analytics features to track rental trends, analyze revenue, and optimize inventory management. This could include generating reports on most rented cars, revenue per period, and customer preferences.*

*Maintenance and Service Tracking: Implement features to track car maintenance schedules and service history. This ensures that rented cars are in good condition and reduces the likelihood of breakdowns during rentals.*

*Localization and Internationalization: Support multiple languages and currencies to cater to a diverse customer base. This involves adapting the system to handle different languages, date formats, and currency conversions.*

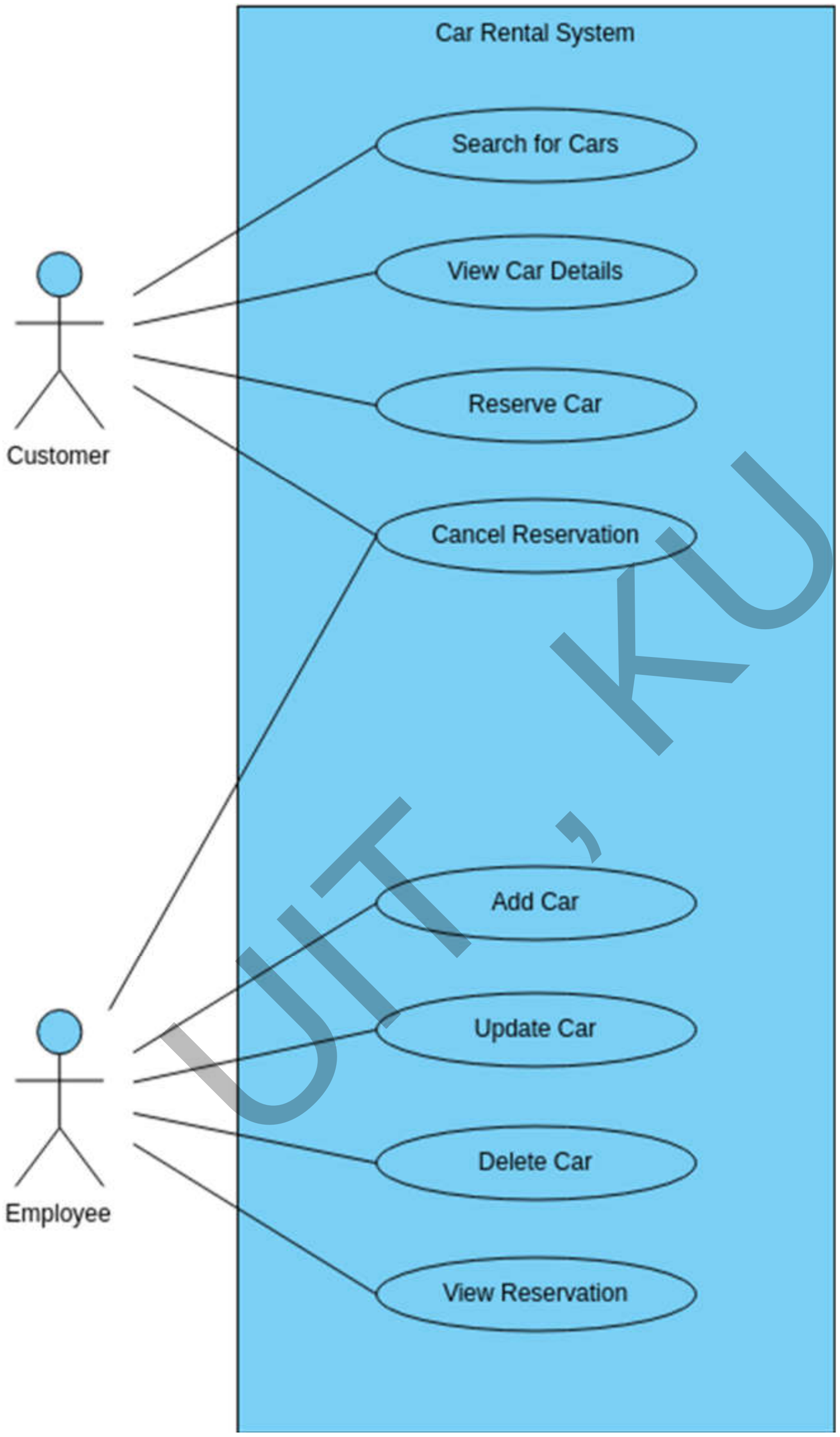
*Mobile Application: Develop a mobile application for the car rental system, enabling customers to browse available cars, make reservations, and manage rentals directly from their smartphones*

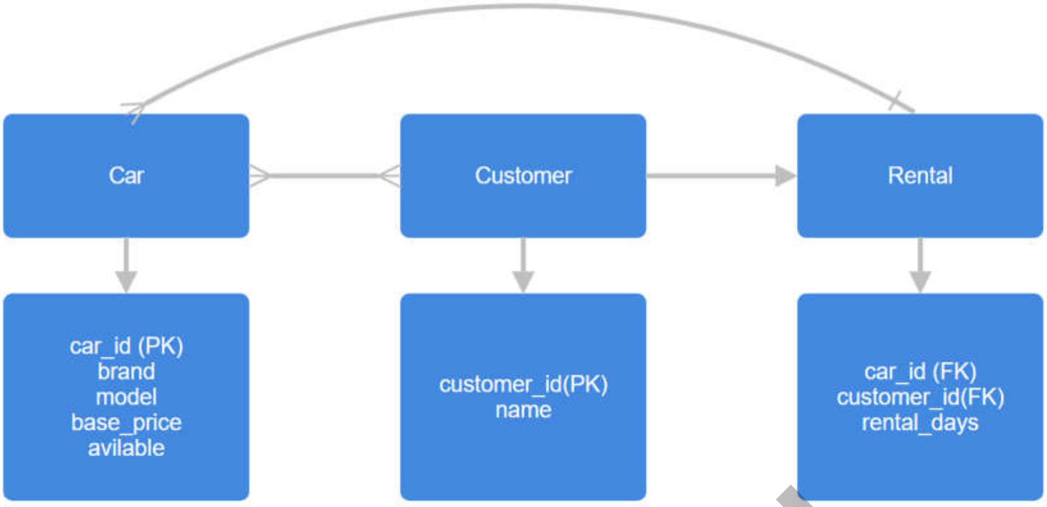
*Feedback and Review System: Incorporate a feedback and review system where customers can rate their rental experience and provide comments. This helps in maintaining service quality and addressing customer concerns.*

## 8. Hardware & Software Specification

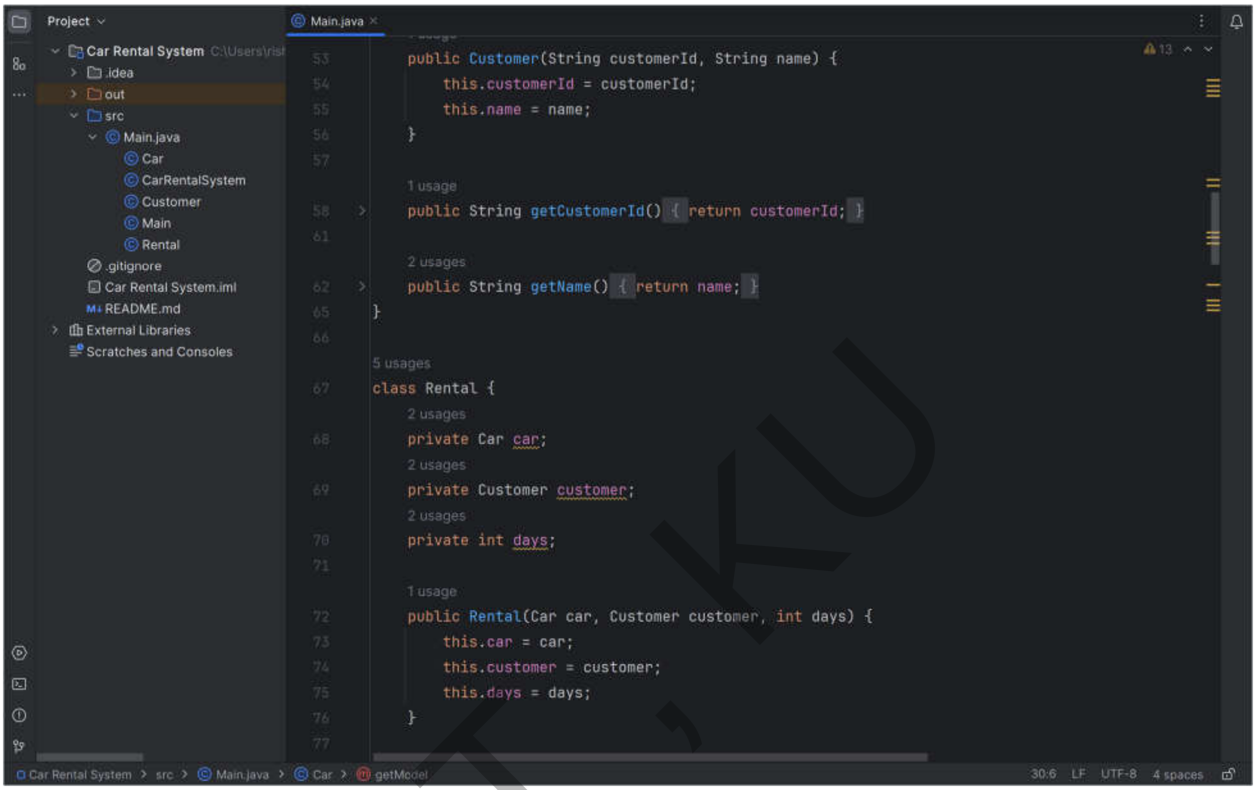
✓ *The provided Java project is a simple car rental system implemented using object-oriented programming principles. It consists of classes such as Car, Customer, Rental, and CarRentalSystem. The system allows users to rent and return cars, displaying rental information and calculating prices based on the number of rental days. It utilizes Java's standard libraries for input/output operations. The project can be run on any hardware capable of running Java applications, and it requires a Java Development Kit (JDK) installed on the system for compilation and execution.*

## 9. System Diagram & ER Diagram




Car
Customer
Rental
car_id (PK)
brand
model
base_price
avilable
customer_id(PK)
name
car_id (FK)
customer_id(FK)
rental_days

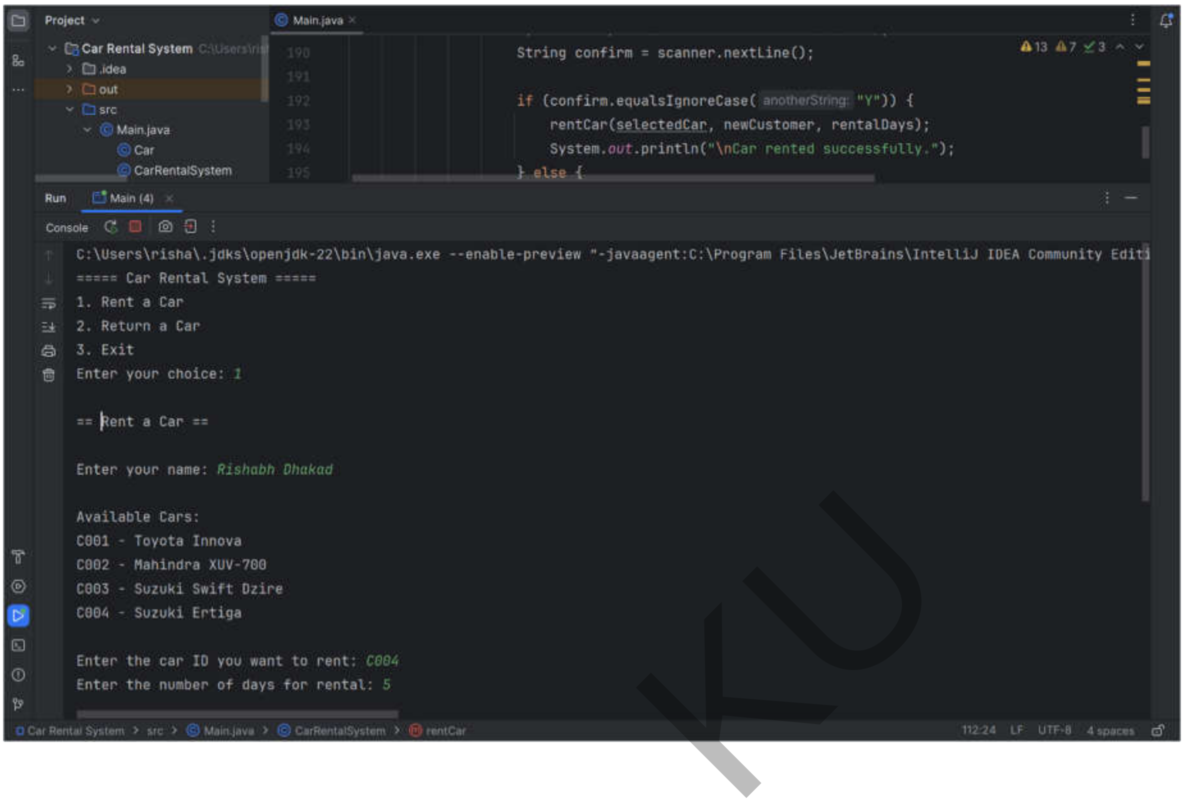
## 10. Code Representation



```
    public Customer(String customerId, String name) {
        this.customerId = customerId;
        this.name = name;
    }

    1 usage
    public String getCustomerId() { return customerId; }

    2 usages
    public String getName() { return name; }
}

5 usages
class Rental {
    2 usages
    private Car car;
    2 usages
    private Customer customer;
    2 usages
    private int days;

    1 usage
    public Rental(Car car, Customer customer, int days) {
        this.car = car;
        this.customer = customer;
        this.days = days;
    }
```

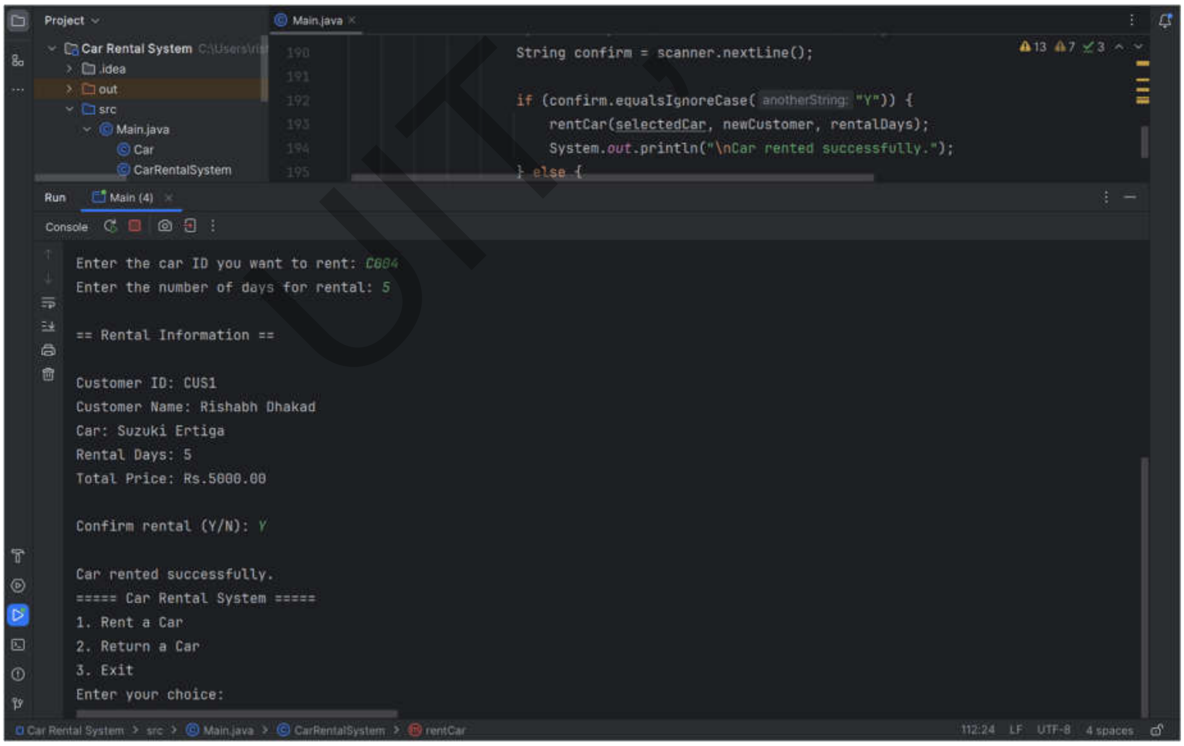
Github Repository Link for Code - https://github.com/StealthImmortal/Car-Rental-System-PT-Project

## 11. Sample Screenshots of Output



```java
String confirm = scanner.nextLine();

if (confirm.equalsIgnoreCase( anotherString: "Y")) {
    rentCar(selectedCar, newCustomer, rentalDays);
    System.out.println("\nCar rented successfully.");
} else {
```

```
C:\Users\risha\.jdks\openjdk-22\bin\java.exe --enable-preview "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editi
===== Car Rental System =====
1. Rent a Car
2. Return a Car
3. Exit
Enter your choice: 1

== Rent a Car ==

Enter your name: Rishabh Dhakad

Available Cars:
C001 - Toyota Innova
C002 - Mahindra XUV-700
C003 - Suzuki Swift Dzire
C004 - Suzuki Ertiga

Enter the car ID you want to rent: C004
Enter the number of days for rental: 5
```



```java
String confirm = scanner.nextLine();

if (confirm.equalsIgnoreCase( anotherString: "Y")) {
    rentCar(selectedCar, newCustomer, rentalDays);
    System.out.println("\nCar rented successfully.");
} else {
```

```
Enter the car ID you want to rent: C004
Enter the number of days for rental: 5

== Rental Information ==

Customer ID: CUS1
Customer Name: Rishabh Dhakad
Car: Suzuki Ertiga
Rental Days: 5
Total Price: Rs.5000.00

Confirm rental (Y/N): Y

Car rented successfully.
===== Car Rental System =====
1. Rent a Car
2. Return a Car
3. Exit
Enter your choice:
```

## 12. Conclusion

✓ *The provided Java code implements a simple car rental system. It allows users to rent and return cars, displaying rental information and calculating prices based on the number of rental days. The system keeps track of available cars, customers, and rentals. Users interact with the system through a menu-driven interface. Cars have unique identifiers, brands, models, and base prices per day. Customers have unique identifiers and names. Rentals associate a car with a customer and specify the number of rental days.*

ORACLE
Academy

# AWARD of FINAL EXAM COMPLETION

## Java Foundations

PRESENTED TO

Dhakad Rishabh Rajendra 20220702033

FOR SATISFACTORY COMPLETION OF COURSE FINAL EXAM

12th March 2024

**William McCabe**
**Vice President, Oracle Academy**

ORACLE
Academy

# AWARD of COURSE COMPLETION

## Java Foundations

PRESENTED TO

Umika Patel 20220702073

FOR SATISFACTORY COMPLETION OF ALL COURSEWORK

30th March 2024

**William McCabe**
**Vice President, Oracle Academy**