COMSATS University Islamabad Abbottabad Campus

Assignment: 1.

Subject: Software Design and Architecture.

Section: 5A.

Submitted By:

HAMZA GUL(FA22-BSE-086)

Topic: Report on car rental booking system.

Submitted to: Sir Mukhtiar Zamin.

Submission date: 8th November 2024.

Project: Car Rental System

1. Introduction

The Car Rental Booking System is designed to streamline the car rental process by providing users with a simple, intuitive interface for browsing, booking, and managing car rentals. This system is particularly aimed at simplifying the rental experience for customers and optimizing the operation and resource allocation for car rental agencies. The architecture incorporates modular design principles and scalable technology to ensure both functionality and future adaptability.

2. Objectives

Customer Convenience: Allow customers to browse available cars, make reservations, and manage bookings easily.

Operational Efficiency: Improve the process of managing reservations, tracking vehicle status, and updating rental records.

Scalability: Build a system that can grow to accommodate additional locations, vehicles, and users over time.

Data Security and Reliability: Implement secure data storage and ensure that sensitive customer data is protected.

3. System Architecture

The Car Rental Booking System uses a layered architecture with a focus on separation of concerns to promote scalability and maintainability.

- ➢ **Presentation Layer:** The frontend interface, developed as a web application, allows users to search for cars, view car details, and manage reservations. This layer interacts with users and communicates with the backend to display real-time information.
- ➢ **Business Logic Layer:** Contains the core logic for rental operations, including vehicle availability checks, rental pricing calculations, and booking validations. This layer enforces business rules and handles the flow of data between the frontend and the database.

➢ **Data Access Layer:** Manages interactions with the database, including CRUD operations for cars, customers, bookings, and payment records. This layer provides a clean separation between business logic and data management.

➢ **Database Layer:** A relational database stores data about vehicles, customers, bookings, and payment history. Tables are designed to accommodate various relationships, such as one-to-many relationships between customers and bookings.

4. Key Functional Modules

User Management: Allows new users to register and existing users to log in securely. User data is stored and managed securely to maintain privacy.

Vehicle Management: Manages the inventory of vehicles available for rent, including their availability, types, rental rates, and maintenance records.

Booking Management: Facilitates car reservations and manages details such as booking dates, customer details, and rental duration. The module ensures that only available cars can be booked and avoids scheduling conflicts.

Payment Processing: Integrates with third-party payment gateways to handle transactions securely. This module logs payment details and ensures successful transaction processing.

Admin Panel: Provides tools for administrators to manage vehicles, review booking history, and update the rental fleet information.
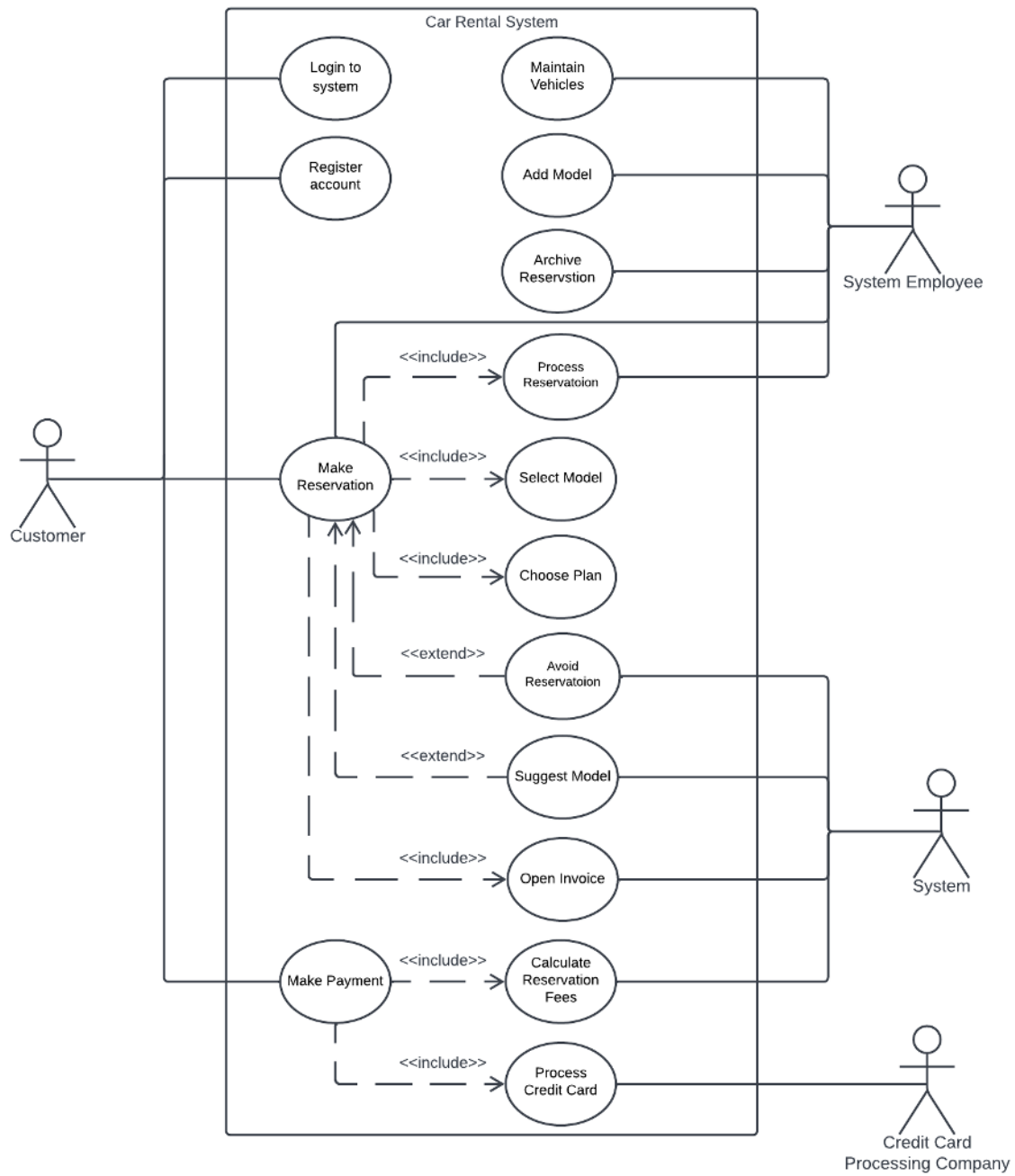
5. Design Patterns

The Car Rental Booking System applies several key design patterns to optimize modularity and flexibility:

Model-View-Controller (MVC): Separates user interface, business logic, and data, promoting maintainability.

Factory Pattern: Used in the Vehicle Management module to instantiate various car types (e.g., SUV, sedan) based on customer needs.

Observer Pattern: Applied to notify users of status changes, such as car availability and booking confirmation.

Usecase Diagram:

Car Rental System

- Login to system
- Register account
- Maintain Vehicles
- Add Model
- Archive Reservstion
- Make Reservation
- Process Reservatoion — <<include>>
- Select Model — <<include>>
- Choose Plan — <<include>>
- Avoid Reservatoion — <<extend>>
- Suggest Model — <<extend>>
- Open Invoice — <<include>>
- Make Payment
- Calculate Reservation Fees — <<include>>
- Process Credit Card — <<include>>

Actors: Customer, System Employee, System, Credit Card Processing Company

Fully Dressed Usecase Diagram:

Use Case ID: UC-001

Use Case Name: Login

Primary Actor: Customer

Preconditions:

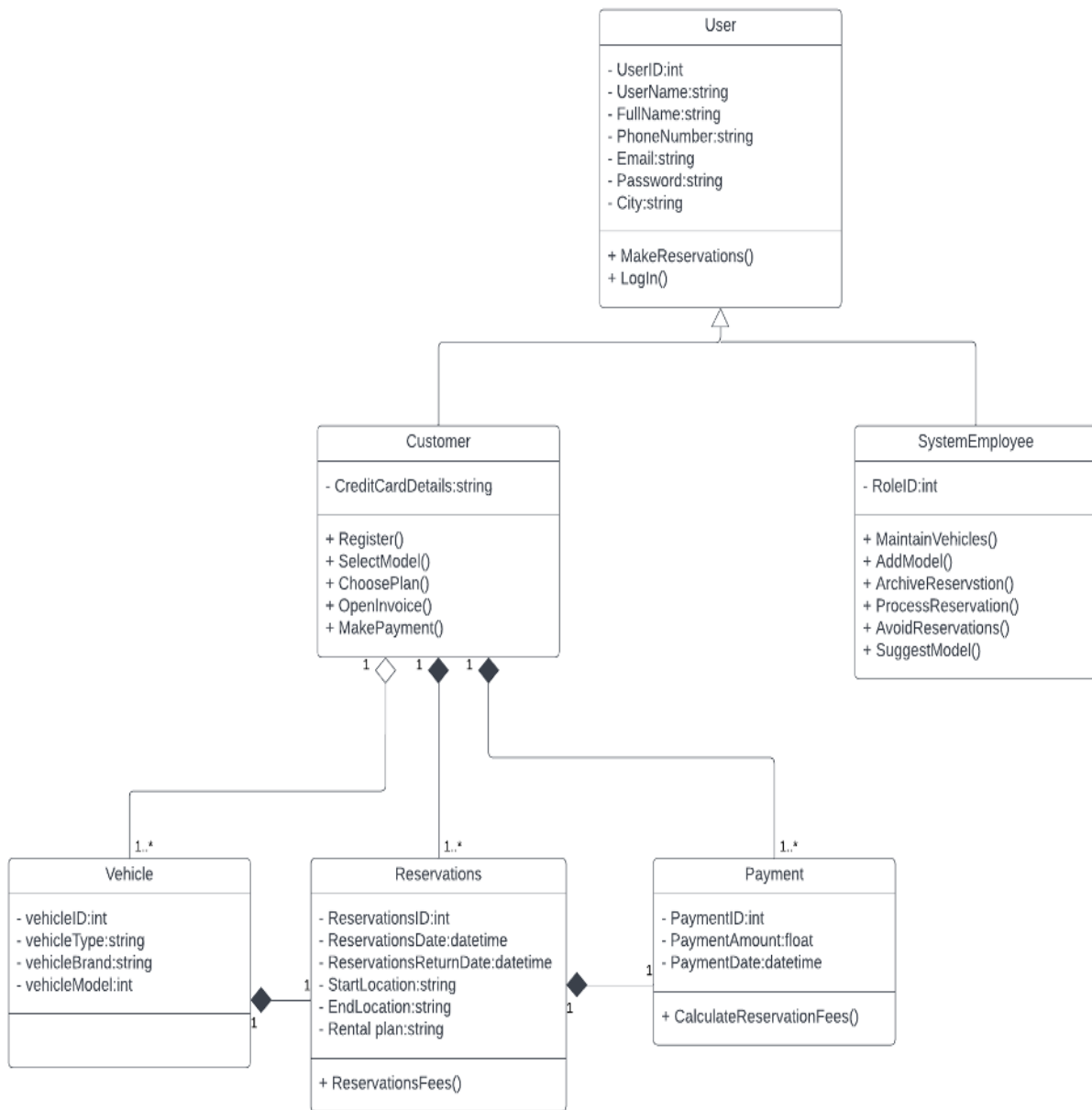- Customer has a valid **username** and **password**.

Postconditions:

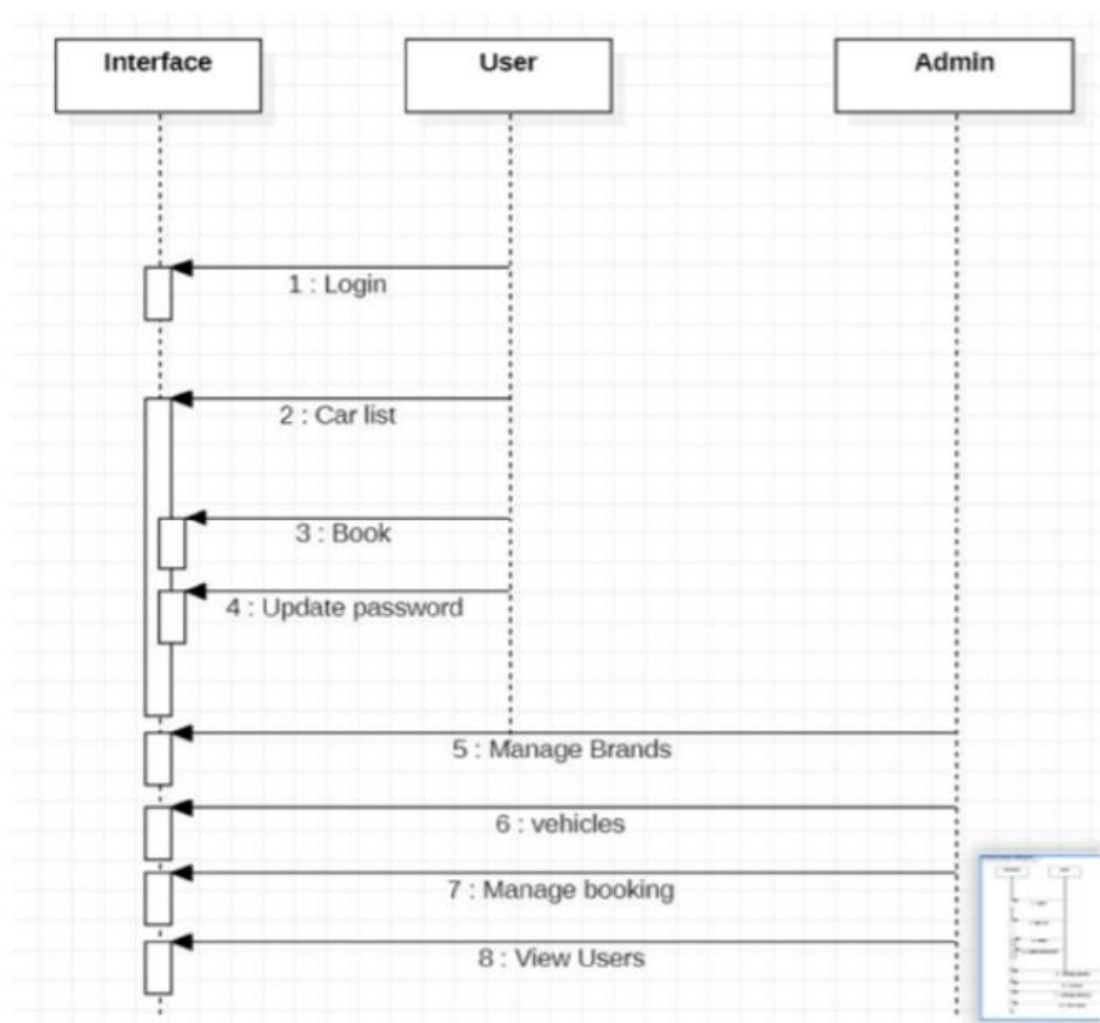- Customer is logged in and redirected to the dashboard.

Main Success Scenario:

1. Customer enters **username** and **password**.
2. System verifies the credentials.
3. If valid, the customer is logged in and redirected to the dashboard.
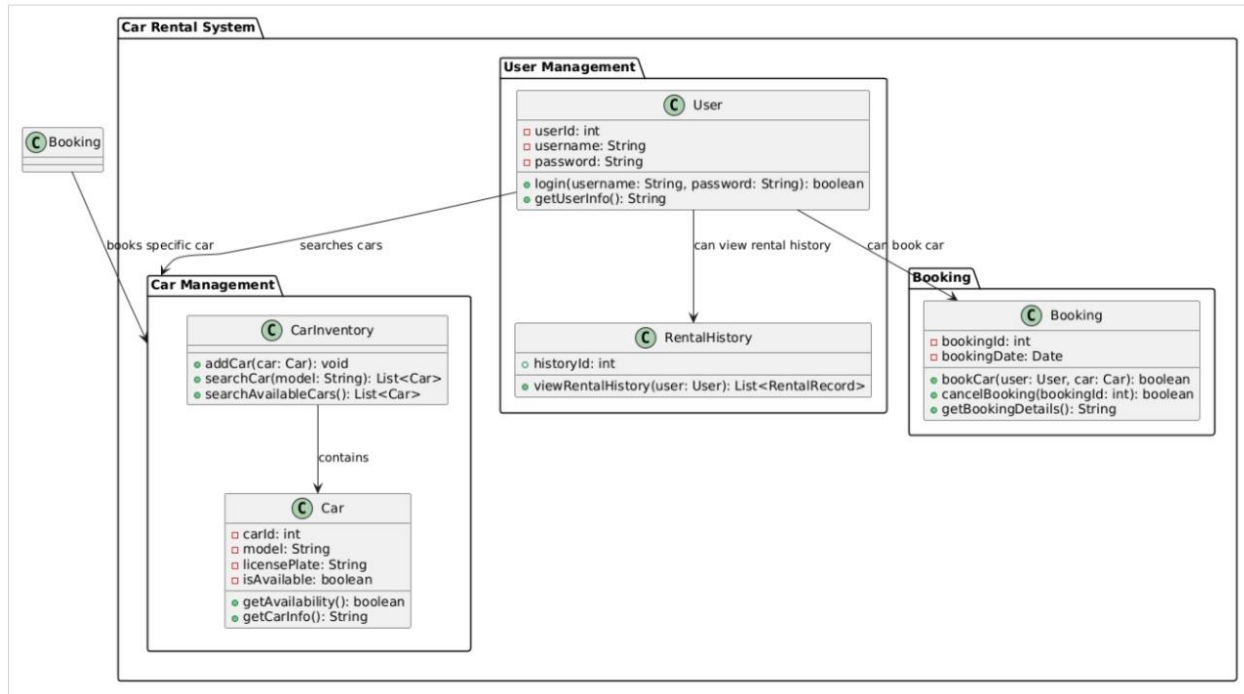4. If invalid, the system displays an error message: "Invalid username or password."

Class Diagram

User

- UserID:int
- UserName:string
- FullName:string
- PhoneNumber:string
- Email:string
- Password:string
- City:string

+ MakeReservations()
+ LogIn()

Customer

- CreditCardDetails:string

+ Register()
+ SelectModel()
+ ChoosePlan()
+ OpenInvoice()
+ MakePayment()

SystemEmployee

- RoleID:int

+ MaintainVehicles()
+ AddModel()
+ ArchiveReservstion()
+ ProcessReservation()
+ AvoidReservations()
+ SuggestModel()

Vehicle

- vehicleID:int
- vehicleType:string
- vehicleBrand:string
- vehicleModel:int

Reservations

- ReservationsID:int
- ReservationsDate:datetime
- ReservationsReturnDate:datetime
- StartLocation:string
- EndLocation:string
- Rental plan:string

+ ReservationsFees()

Payment

- PaymentID:int
- PaymentAmount:float
- PaymentDate:datetime

+ CalculateReservationFees()

Sequence Diagram

Package Diagram

Conclusion

The Car Rental Booking System is designed to be a robust, user-friendly solution for both customers and rental agencies. Its modular architecture allows easy maintenance and scaling, supporting future enhancements as the business expands. By leveraging well-known design patterns and following best practices in software architecture, this system provides a reliable platform that meets both user demands and business requirements.