PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Object Oriented Analysis and Design using Java (UE20CS352)

Mini Project

PROJECT TITLE:

HOTEL MANAGEMENT SYSTEM USING JAVA

TEAM MEMBERS:

1. **RITVIK W – PES2UG20CS275**
2. **ROBIN ROY– PES2UG20CS279**
3. **RONIT B – PES2UG20CS284**
4. **SAIEESH S RAO – PES2UG20CS298**

Submitted to:

Prof. Saranya Devi

Problem statement:

A hotel management system using Java is a software solution designed to automate various operations and processes in a hotel, such as room reservations, check-ins, check-outs, billing, inventory management and food ordering system.

The system will have different modules such as check-in, check-out, food ordering, and billing, all of which will be accessible through a user-friendly interface. It will also be designed to work with different types of users, such as managers, receptionists, and customers. The admin role can view prices of rooms/dishes and edit the values or append new data into the database.

Some of the key features in the hotel management system are:

Reservation module: This module will allow users to make room reservations online or over the phone. It will also enable users to check room availability, view room rates, and select room types. This module will enable users to check guests in, assign rooms, and record guest details such as name, address, and contact information.

Check-out module: This module will enable users to check guests out, generate invoices, and process payments. It will also allow users to update room availability status and track room occupancy rates.

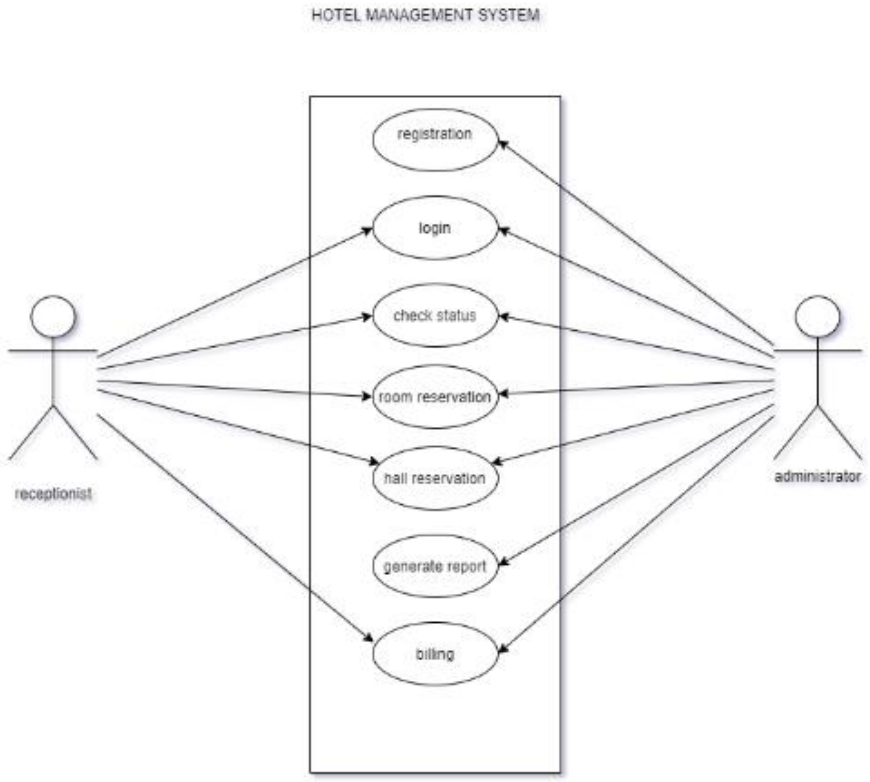
Restaurant module: This module will allow users to manage guest requests for room service, housekeeping, and maintenance. It will also enable users to update room status and inventory levels.

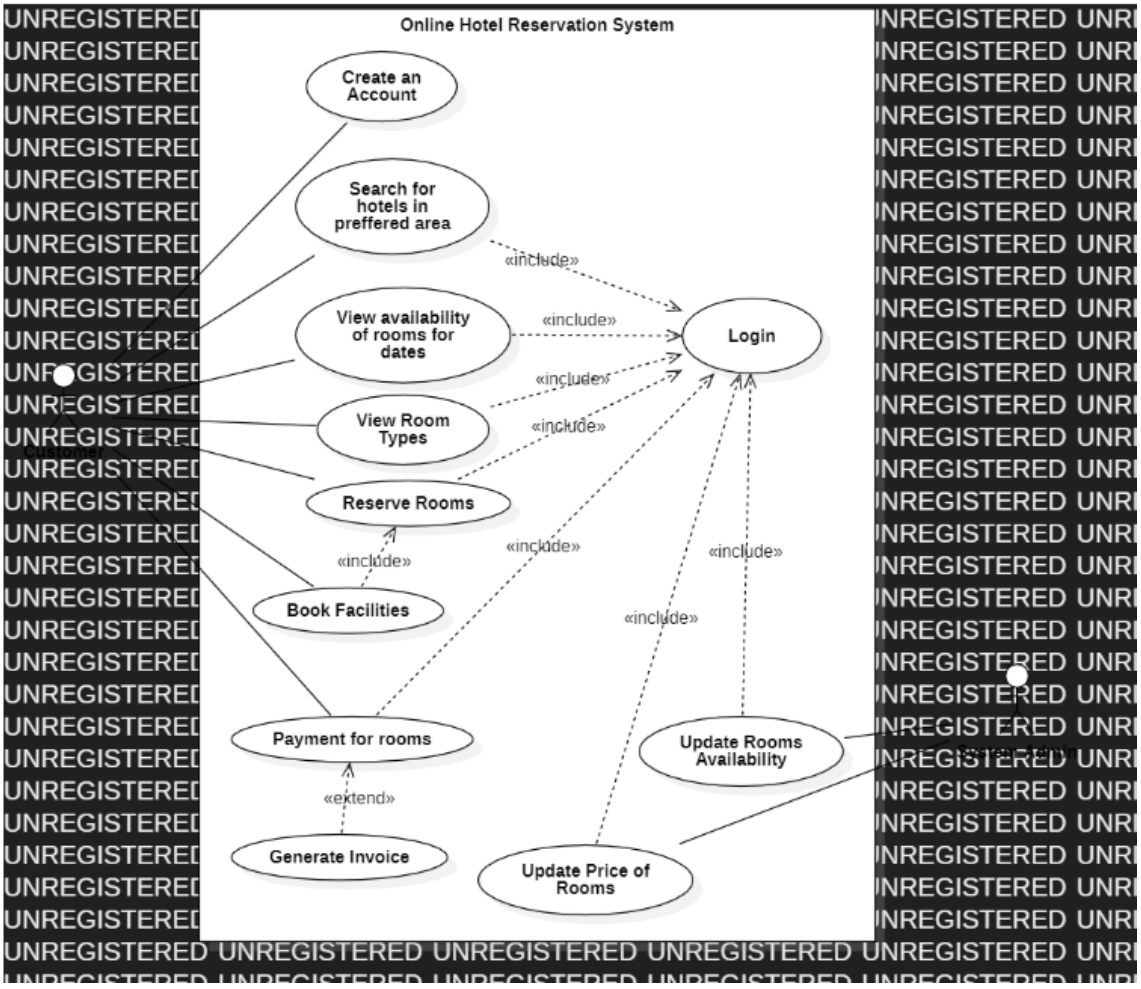
Billing module: This module will enable users to generate invoices, process payments, and track revenue. It will also allow users to generate financial reports, such as profit and loss statements and occupancy rates.

The software can also manage dishes, manage rooms and also has a login module.

Overall, the e-commerce application using Java project is a robust and scalable solution for building an online marketplace. It leverages Java's powerful features and offers a user-friendly interface for both buyers and sellers.

Models:


HOTEL MANAGEMENT SYSTEM
registration
login
check status
room reservation
hall reservation
generate report
billing
receptionist
administrator


Online Hotel Reservation System
Create an Account
Search for hotels in preffered area
View availability of rooms for dates
View Room Types
Reserve Rooms
Book Facilities
Payment for rooms
Generate Invoice
Login
Update Rooms Availability
Update Price of Rooms
«include»
«extend»

Design Patterns:

1. **Model-View-Controller (MVC):** The repository follows the Model-View-Controller pattern. The model layer contains the data access objects (DAOs) and entity classes, which represent the data and provide a layer of abstraction. The view layer contains the user interface (UI) classes, which interact with the user and display information. The controller layer contains the controller classes, which act as intermediaries between the model and view layers and handle user input.

2. **Factory Method Pattern:** The system uses the Factory Method pattern in the 'RoomFactory' class, which creates instances of different types of rooms based on the input provided.

Design Principles:

1. Single Responsibility Principle (SRP): The System follows the Single Responsibility Principle, as each class has only one responsibility. For example, the Guest class is responsible for representing a hotel guest, the Room class is responsible for representing a hotel room, and the Hotel class is responsible for managing the hotel's operations.

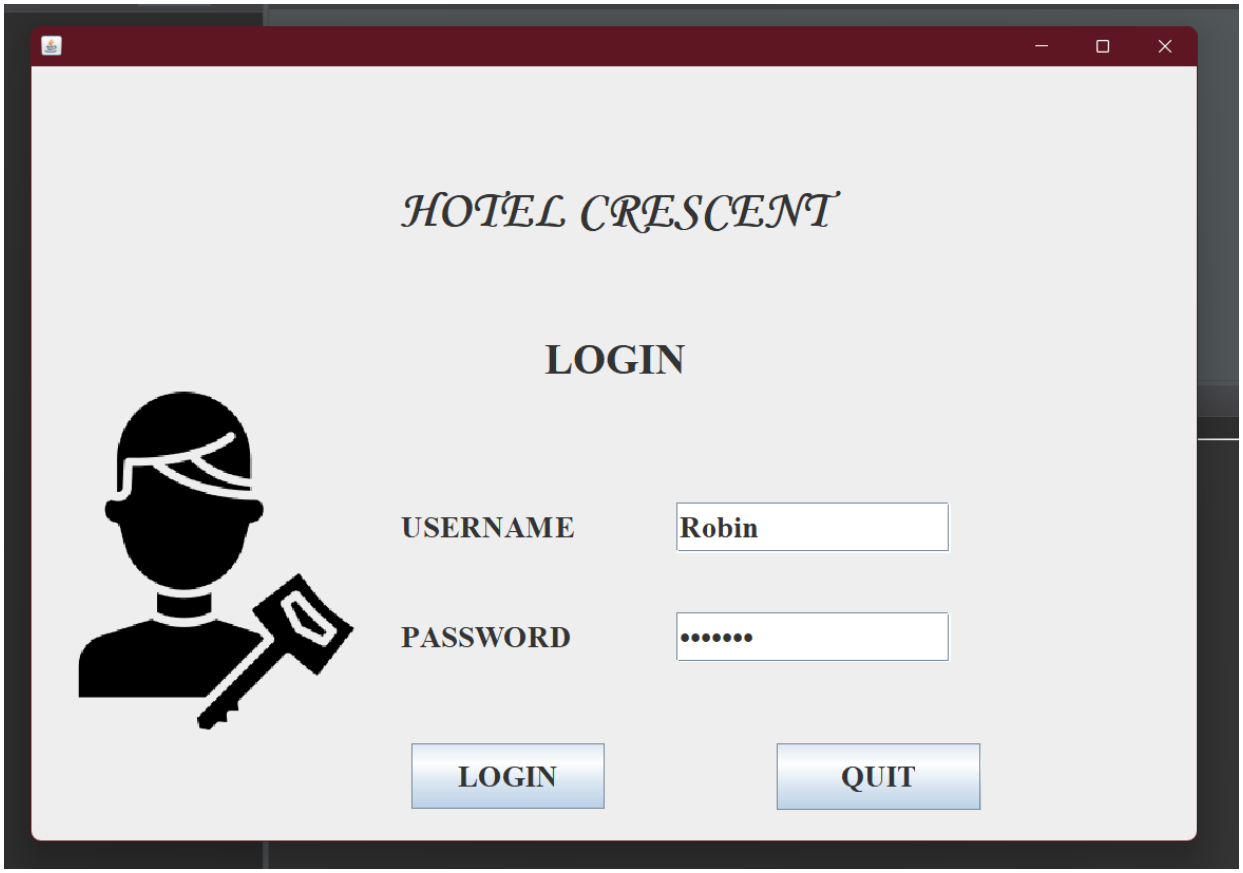
2. Open-Closed Principle (OCP): The system follows the Open-Closed Principle, as it allows for extension without modification. For example, new types of rooms can be added to the Room class without modifying the existing code.

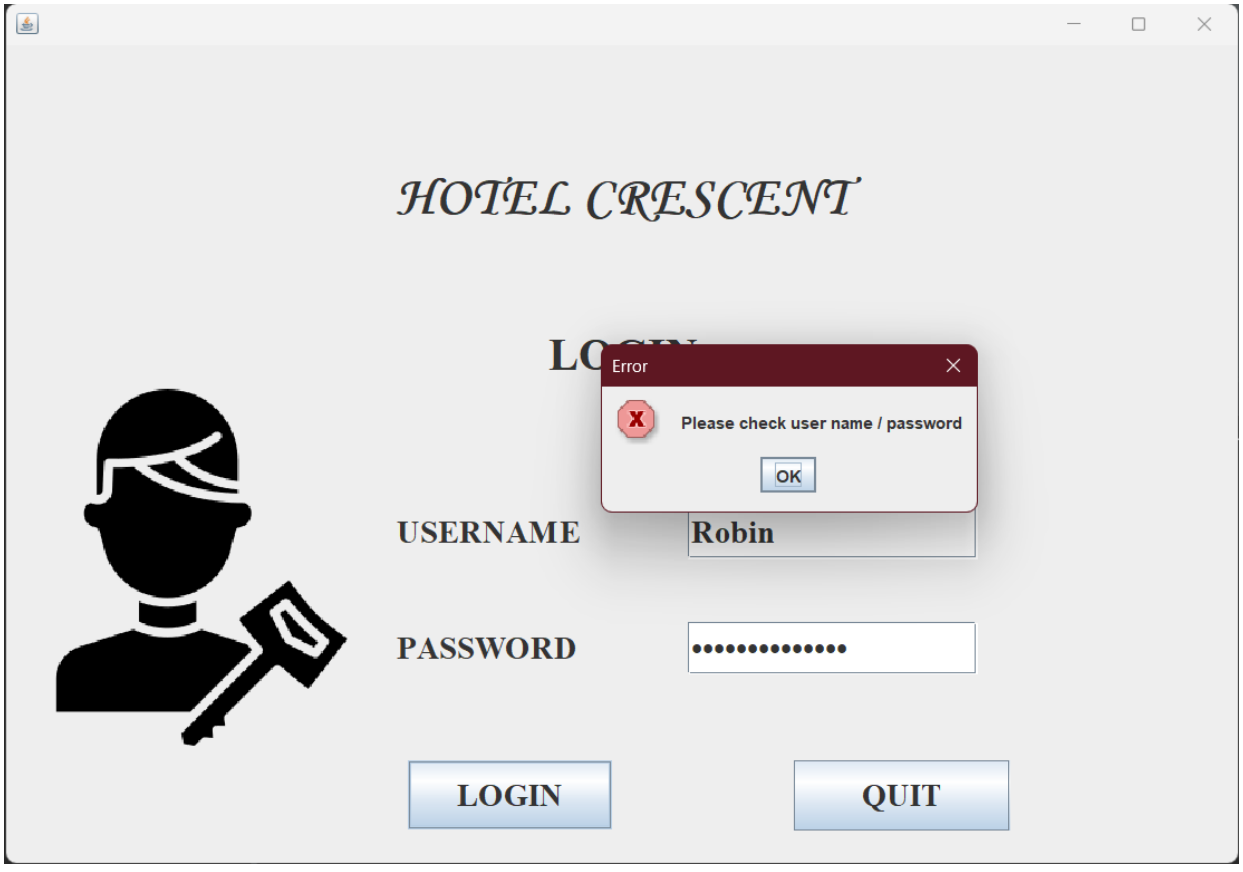
3. Liskov Substitution Principle (LSP): The system follows the Liskov Substitution Principle, as it allows for subclasses to be substituted for their superclasses without affecting the program's behavior.

4. Interface Segregation Principle (ISP): The system follows the Interface Segregation Principle, as each interface defines only the methods that are necessary for its specific purpose.

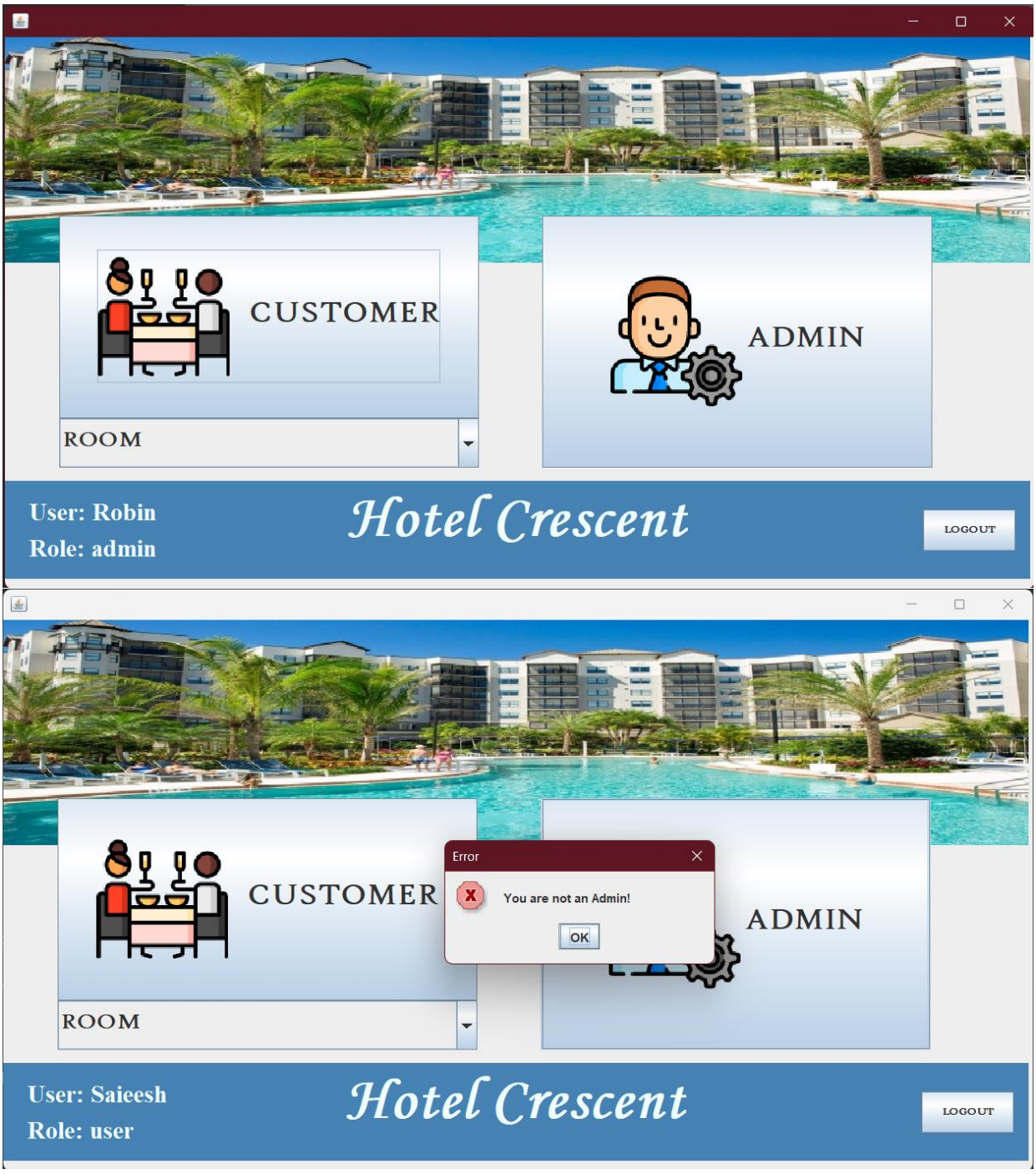
Screenshots:

1. User Login

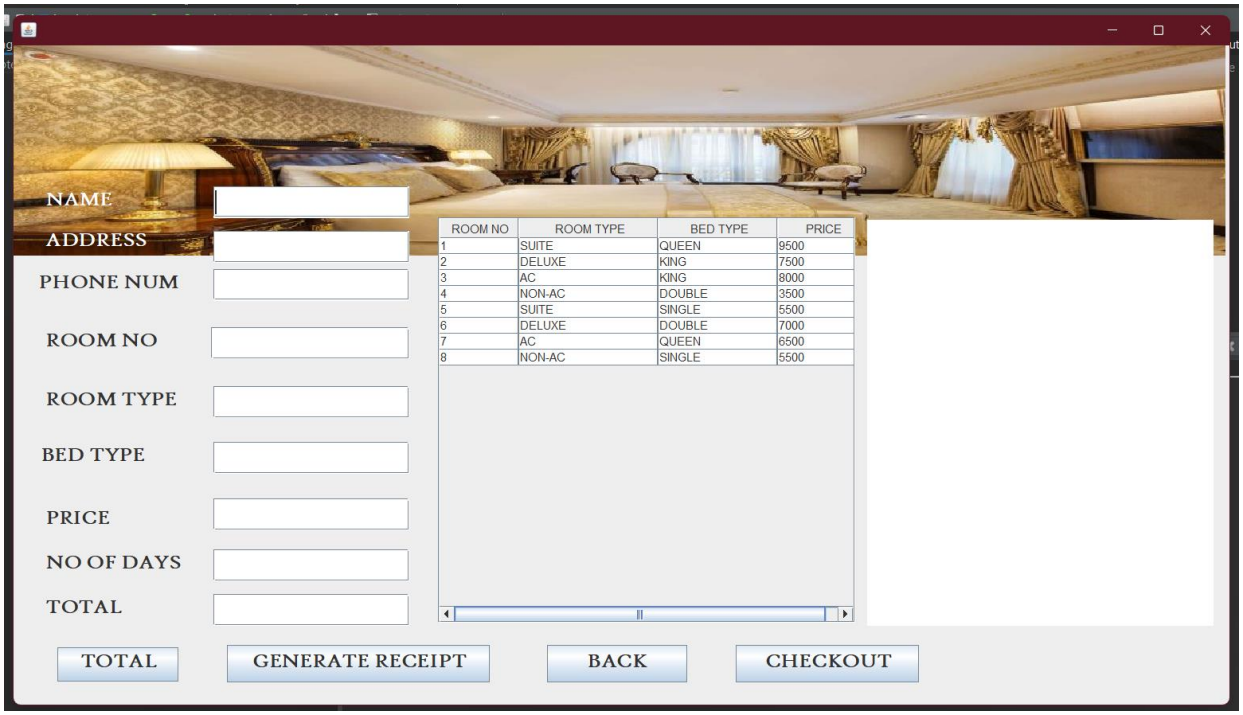


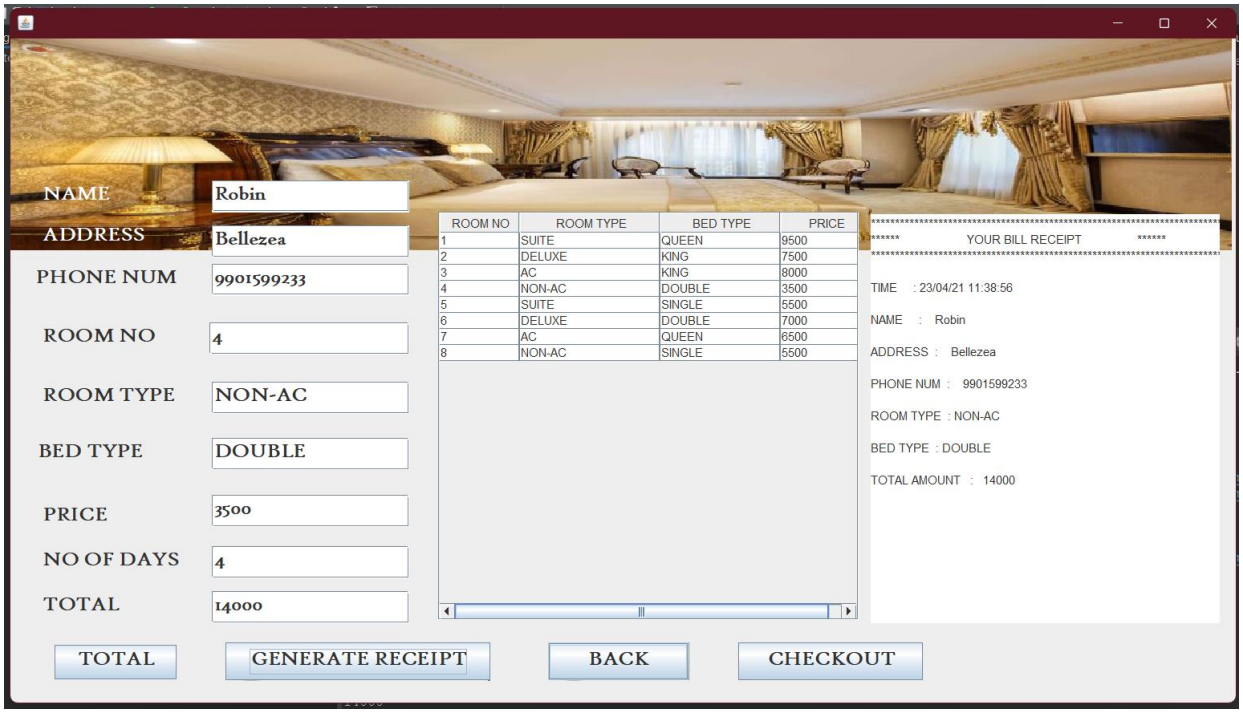


2. UI Page

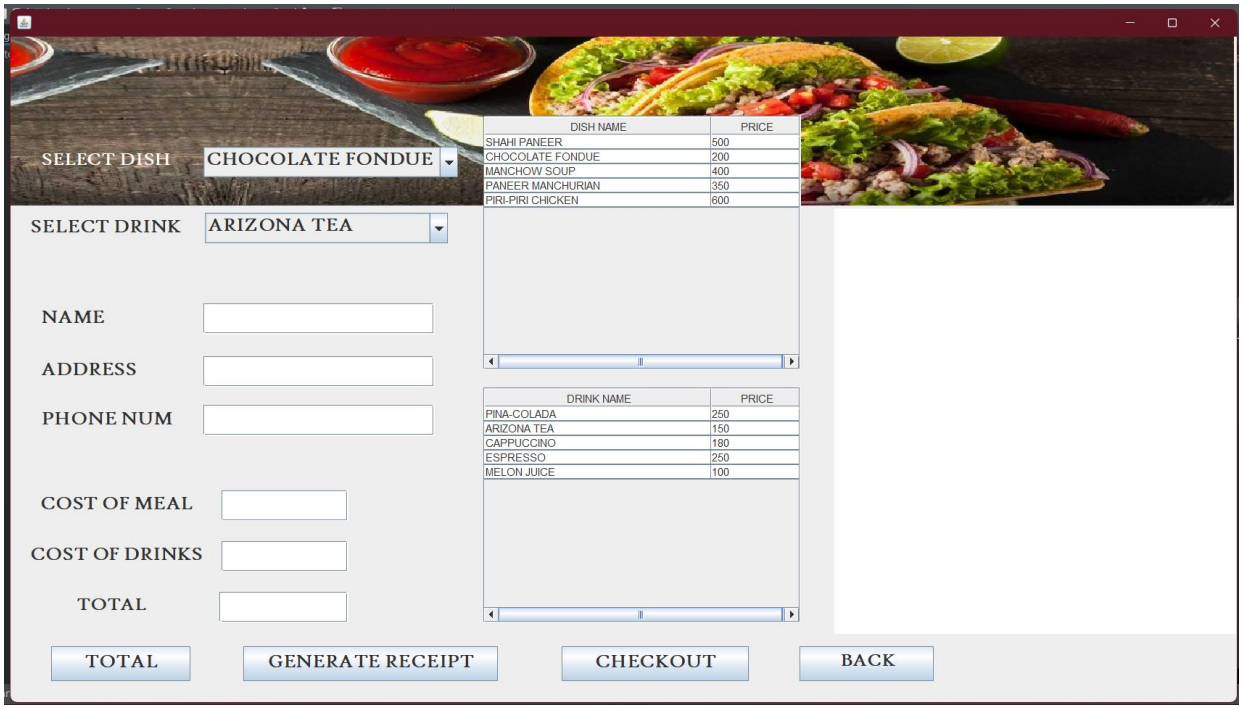

CUSTOMER
ADMIN
ROOM
User: Robin
Role: admin
Hotel Crescent
LOGOUT
CUSTOMER
Error
You are not an Admin!
OK
ADMIN
ROOM
User: Saieesh
Role: user
Hotel Crescent
LOGOUT

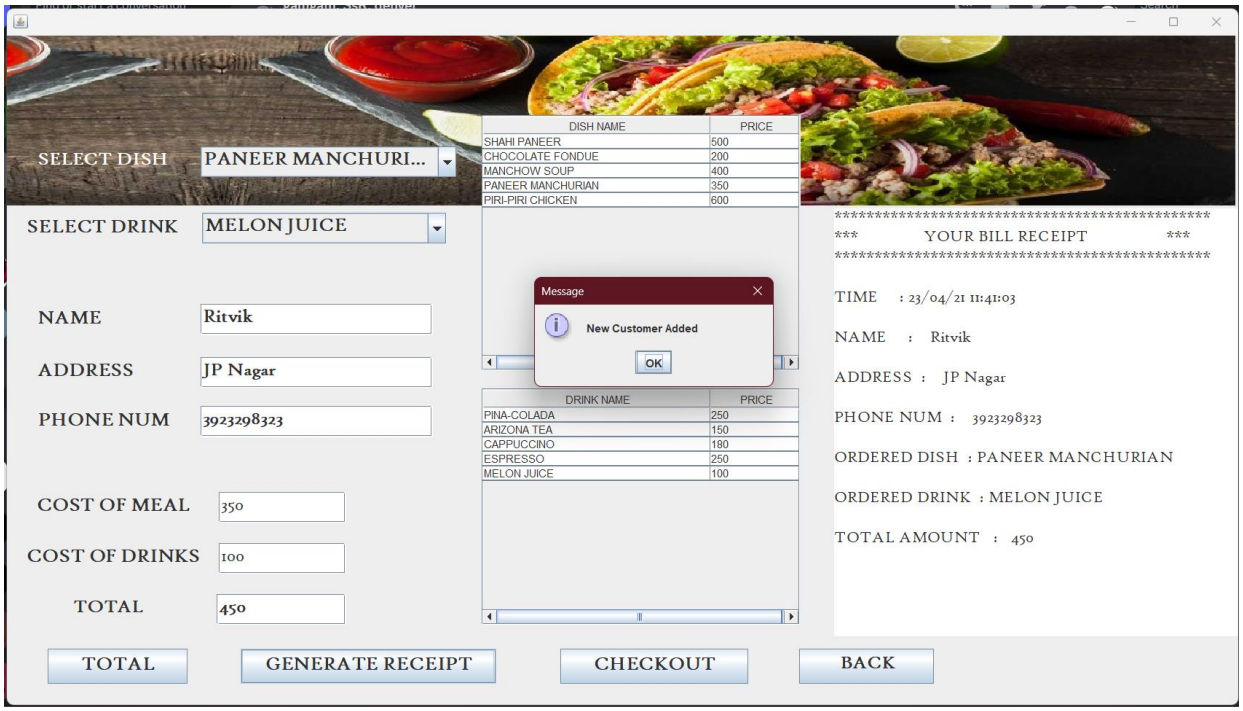
3. Room Booking



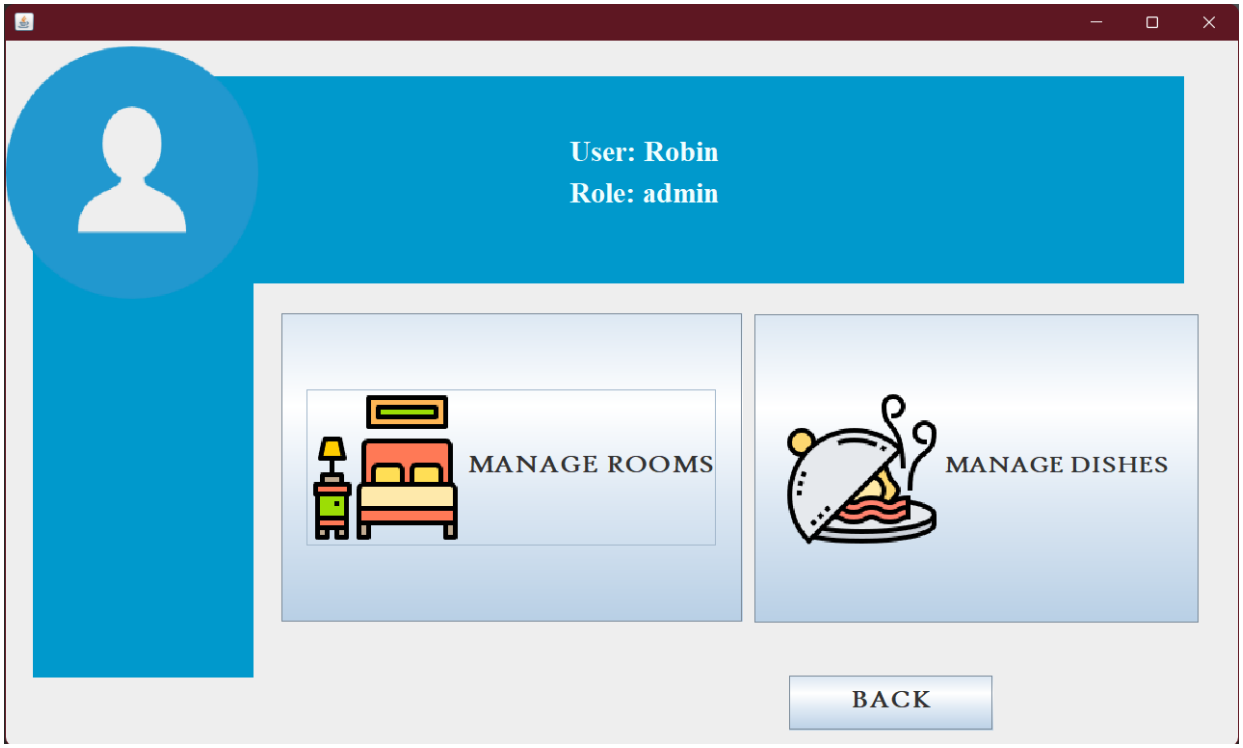


4. Food ordering

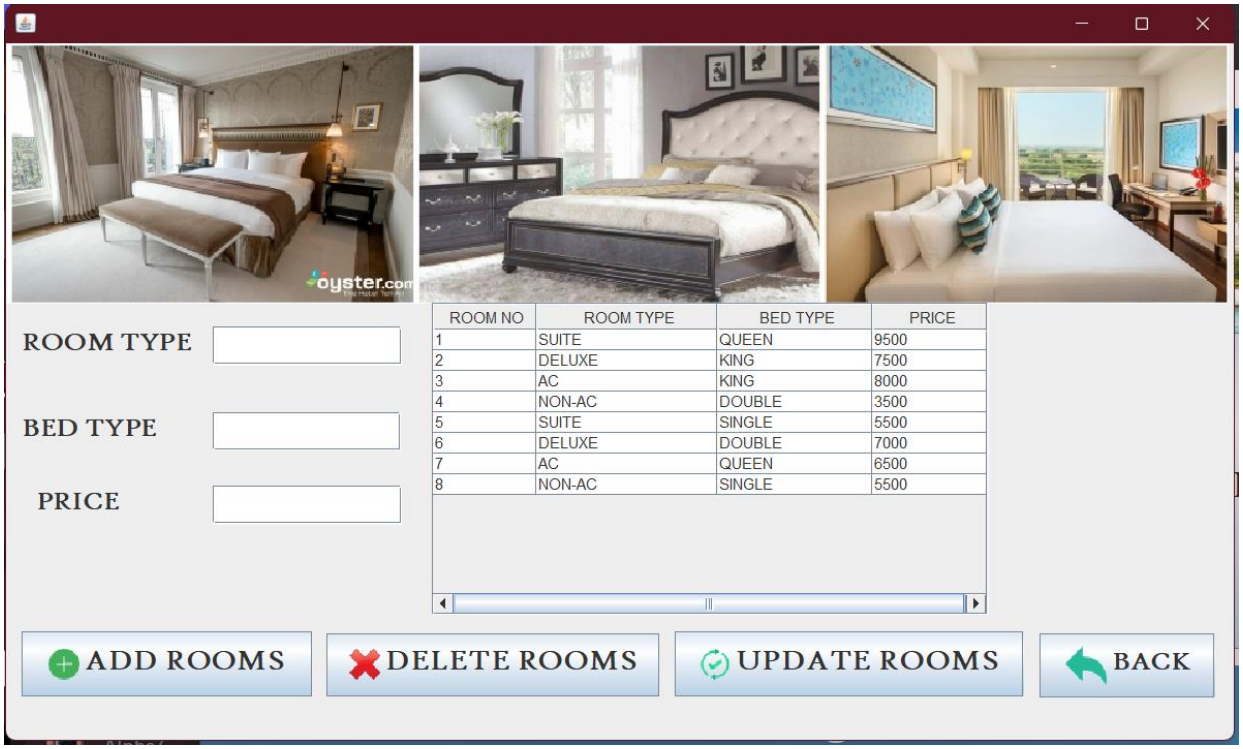




5. Admin interface



6. Admin editing room details



7. Admin editing food details

Github link:

https://github.com/Obsarian/OOAD-Project_Hotel-Management-System