UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

"Hotel Room Booking System"

Submitted by:

K Virupakshi	**PES1UG21CS264**
LSS Praneeth Kumar	**PES1UG21CS305**
Mani Shankar M	**PES1UG21CS307**
Jyothiraditya D	**PES1UG21CS921**

6th Semester E Section

Prof. Bhargavi Mokashi
Assistant Professor

January - May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
100ft Ring Road, Bengaluru – 560 085, Karnataka, India

Table of Contents

INTRODUCTION

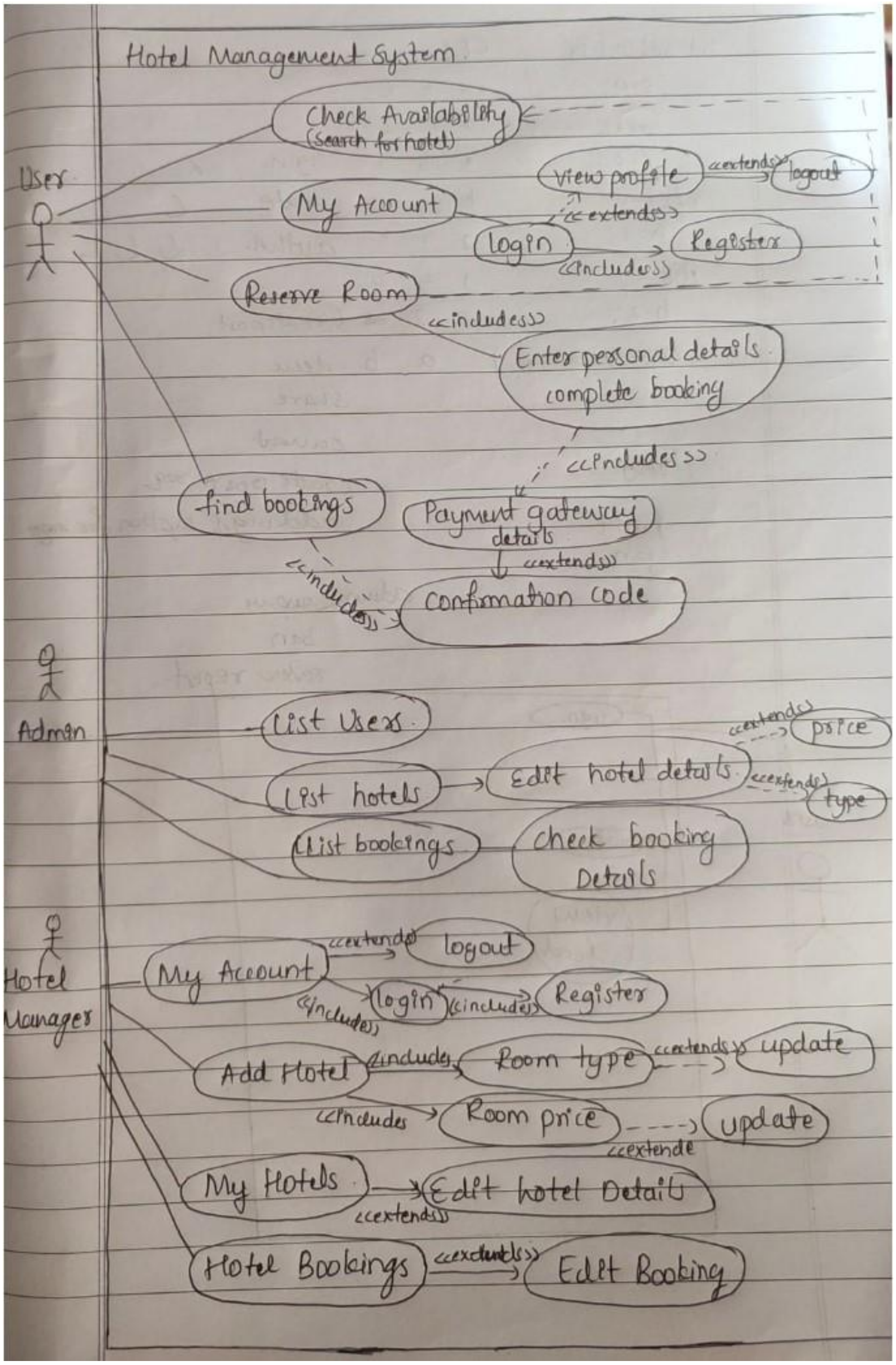
HotelBookingApp is a straightforward web application aimed at streamlining the hotel reservation process for customers and hotel managers alike. Developed using Java and Spring Boot in the backend, and Thymeleaf in the frontend, the system follows the Model-View-Controller (MVC) architectural pattern.

- User Registration & Management: Users can register, login, and manage their profiles. Various data validations (e.g. strong password requirement) have been implemented.
- Hotel Management: Hotel managers can add/edit hotels, specifying details (e.g. name, address, room counts, prices) in a single interface.
- Hotel Search: Enables customers to search for available hotels based on location and check-in/check-out dates.
- Hotel Listing: Displays a list of available hotels with relevant details such as name, available room counts, and prices.
- Hotel Details: Provides in-depth information on hotels, including name, address, room availability, pricing, and an interactive map leveraging the Nominatim geocoding API and Leaflet library.
- Room Booking: Customers can select the desired number of rooms and get redirected to payment for finalizing the reservation.
- Payment Processing: Secure credit card payment with validations like Luhn checks and custom validators for expiry dates and CVV. (No third-party payment gateways are implemented.)
- Booking Management: Customers and hotel managers can view their bookings through the dashboard.
- Admin Panel: Allows administrators to manage users, hotels, rooms, and bookings.

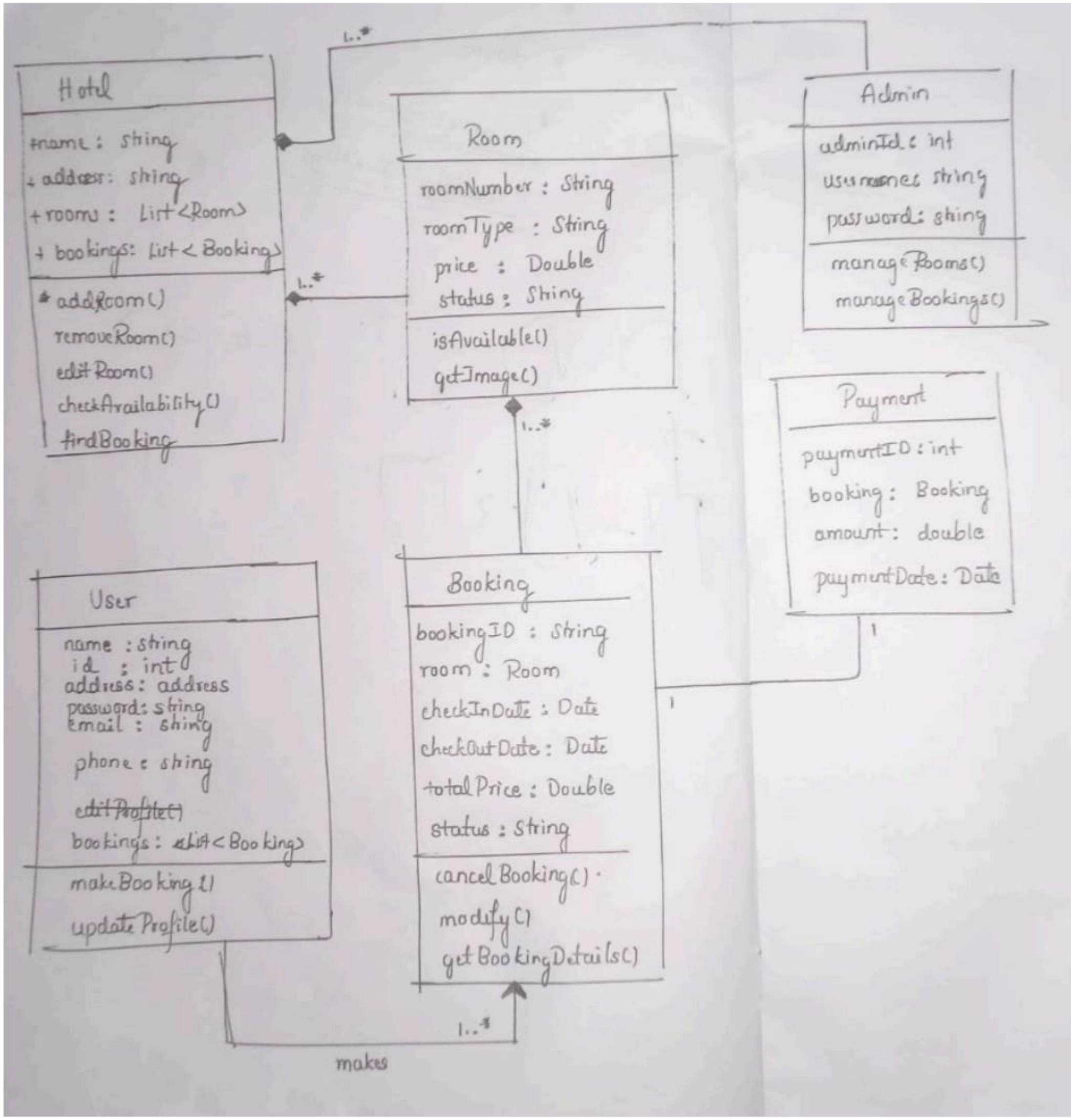
- Responsive Design: The app is optimized for various devices including desktops, tablets, and smartphones.

UML DIAGRAMS

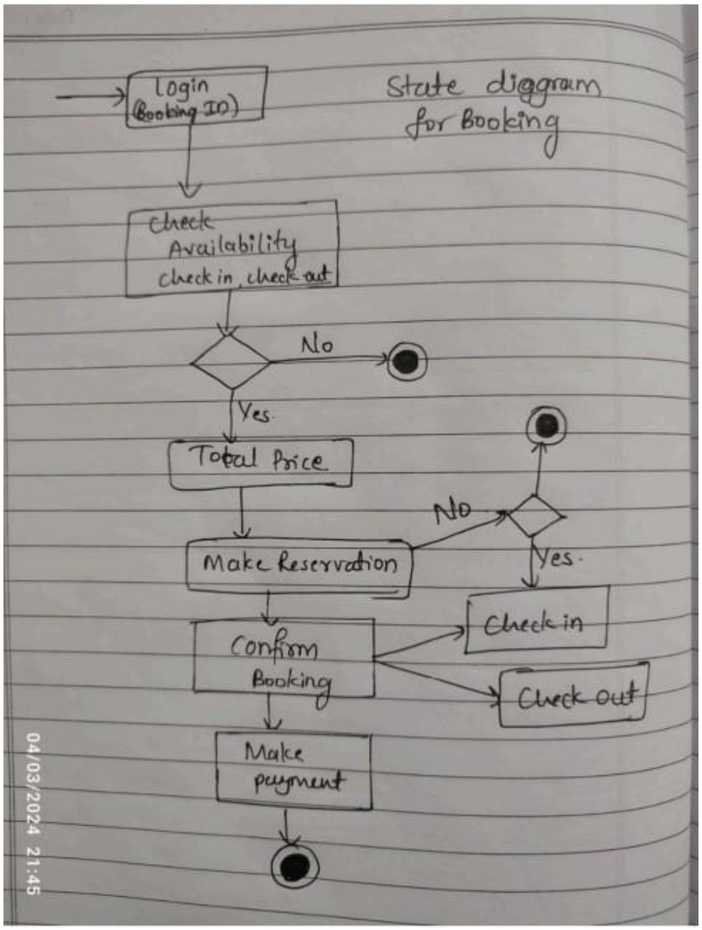
Use case Diagram

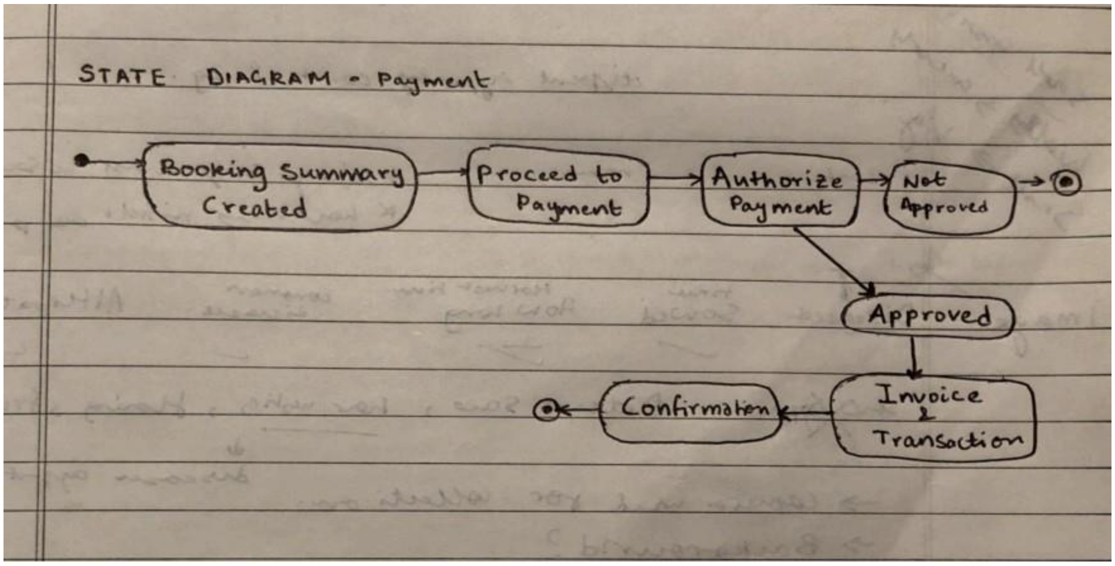

Hotel Management System
User
Check Availability (Search for hotel)
My Account
view profile
<<extends>>
logout
<<extends>>
login
<<includes>>
Register
Reserve Room
<<includes>>
Enter personal details complete booking
<<includes>>
find bookings
Payment gateway details
<<extends>>
<<includes>>
confirmation code
Admin
List Users.
<<extends>>
price
List hotels
Edit hotel details.
<<extends>>
type
List bookings
check booking Details
Hotel Manager
My Account
<<extends>>
logout
<<includes>>
login
<<includes>>
Register
Add Hotel
<<includes>>
Room type
<<extends>>
update
<<includes
Room price
<<extende
update
My Hotels
<<extends>>
Edit hotel Details
Hotel Bookings
<<extends>>
Edit Booking

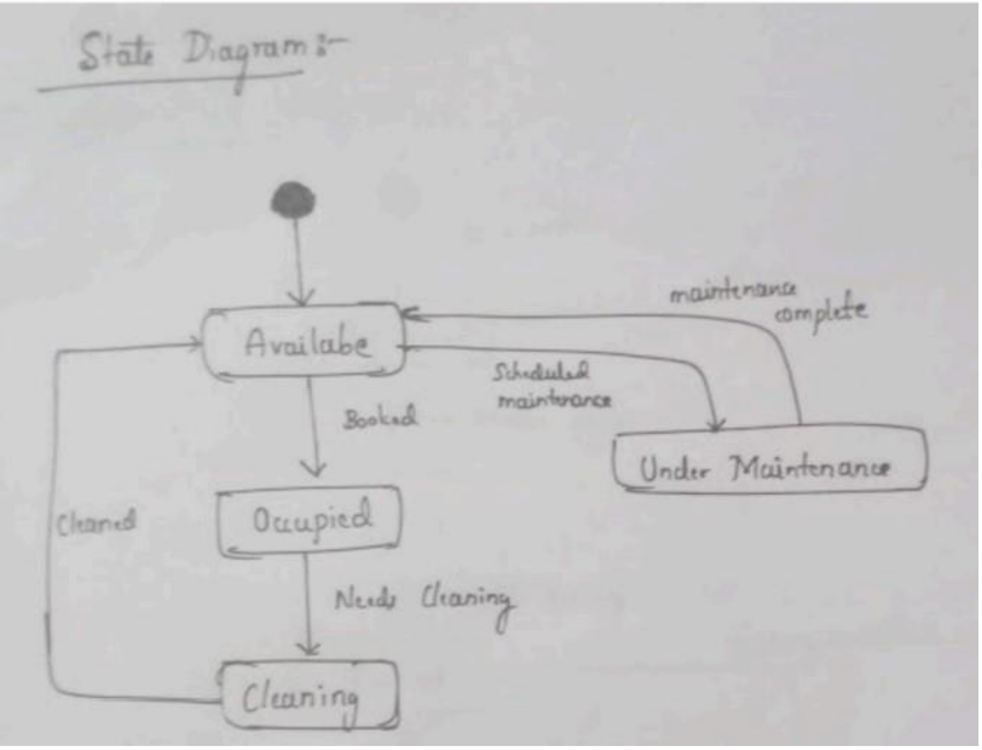
Class Diagram



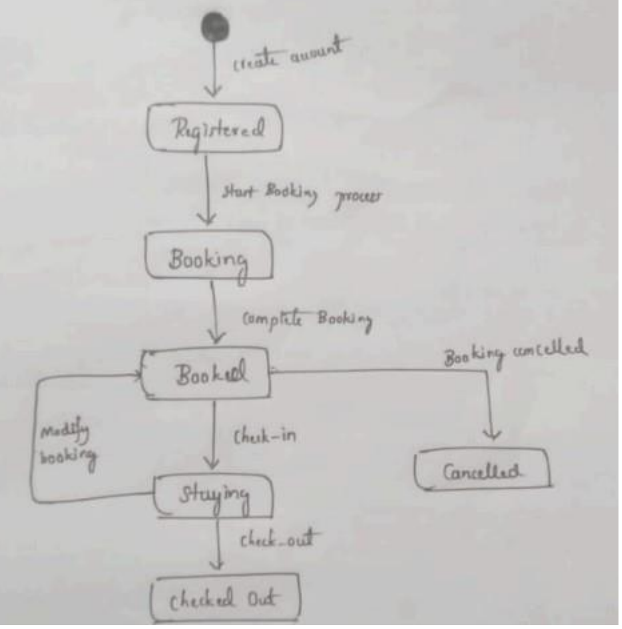
State Diagram



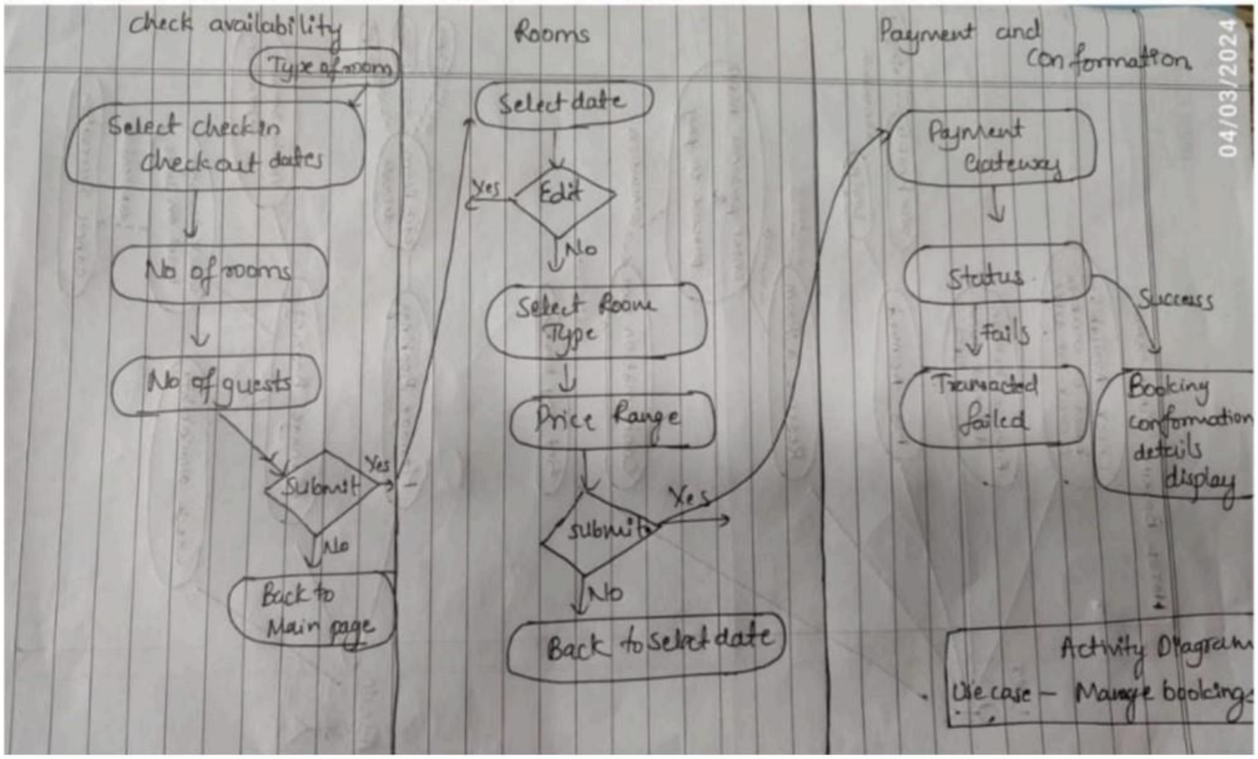

STATE DIAGRAM - Payment
Booking Summary Created
Proceed to Payment
Authorize Payment
Not Approved
Approved
Invoice & Transaction
Confirmation


State Diagram :-
Availabe
Booked
Occupied
Needs Cleaning
Cleaning
Cleaned
Scheduled maintenance
Under Maintenance
maintenance complete

Guest State Diagram :-

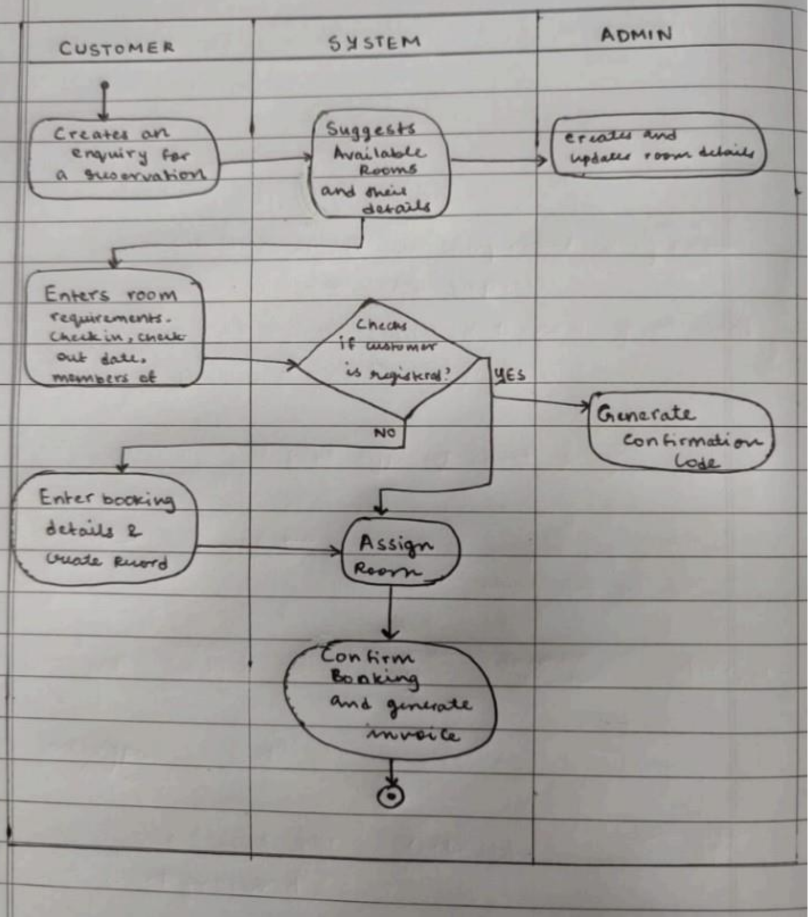


Activity Diagram

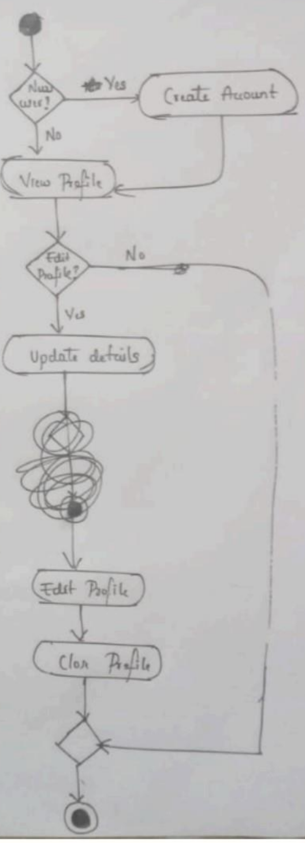


ACTIVITY DIAGRAM

USE CASE - Reserve a Room.

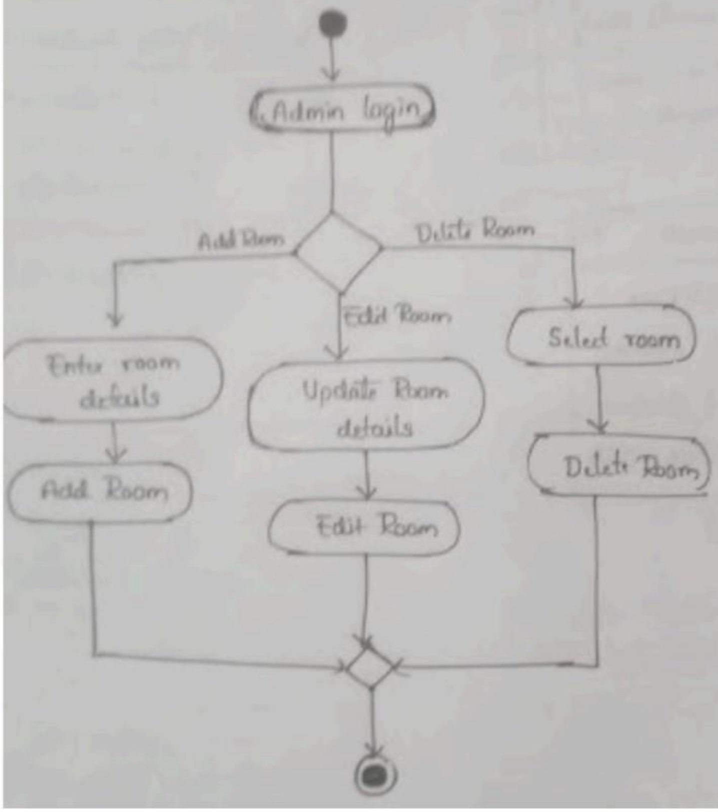

CUSTOMER
SYSTEM
ADMIN
Creates an enquiry for a reservation
Suggests Available Rooms and their details
creates and updates room details
Enters room requirements. Check in, check out date, members of
Checks if customer is registered?
YES
NO
Generate Confirmation Code
Enter booking details & create Record
Assign Room
Confirm Booking and generate invoice

User profile management:



Activity Diagram :-

Room Management :-



DESIGN PRINCIPLES

Single Responsibility Principle (SRP)

The Single Responsibility Principle (SRP) is applied to ensure that each class within the system has only one responsibility or reason to change. This principle is crucial for maintaining modularity, scalability, and testability in the codebase. By adhering to SRP, each class in the system is focused on a specific aspect of user registration or management, making the codebase easier to understand, maintain, and extend.
SRP ensures that each class is focused on a single aspect of user registration or management, leading to a more modular, maintainable, and scalable codebase.

CardExpiryValidator class: This class is responsible for validating the format and validity of a credit card expiry date. It does this by implementing the ConstraintValidator interface and providing the logic for checking if the provided expiry date is in the correct format and is not in the past.

CardExpiry annotation: This annotation marks a field as requiring validation for a credit card expiry date. It is responsible for associating the validation logic (defined in CardExpiryValidator) with the annotated field.

Interface Segregation Principle

The decision to apply the Interface Segregation Principle to the BookingRepository and CustomerRepository was made to ensure that these interfaces are tailored precisely to the specific needs of their clients. By breaking down these interfaces into smaller, focused ones, unnecessary dependencies are avoided, and clients are only required to implement the methods relevant to their use cases. This promotes better code organization and reduces the risk of unintended side effects when changes are made.

DESIGN PATTERNS

Factory Method Pattern

The Factory Method Pattern is utilized to abstract the creation of different types of users (e.g., hotel managers, customers, administrators) based on predefined roles or criteria. This pattern provides a flexible and scalable approach to user creation by encapsulating the creation logic within factory classes. Each subclass of the factory is responsible for creating a specific type of user, promoting code reuse and maintainability. By employing the Factory Method Pattern, the system can accommodate future changes or additions to user roles without requiring significant modifications to the existing codebase.

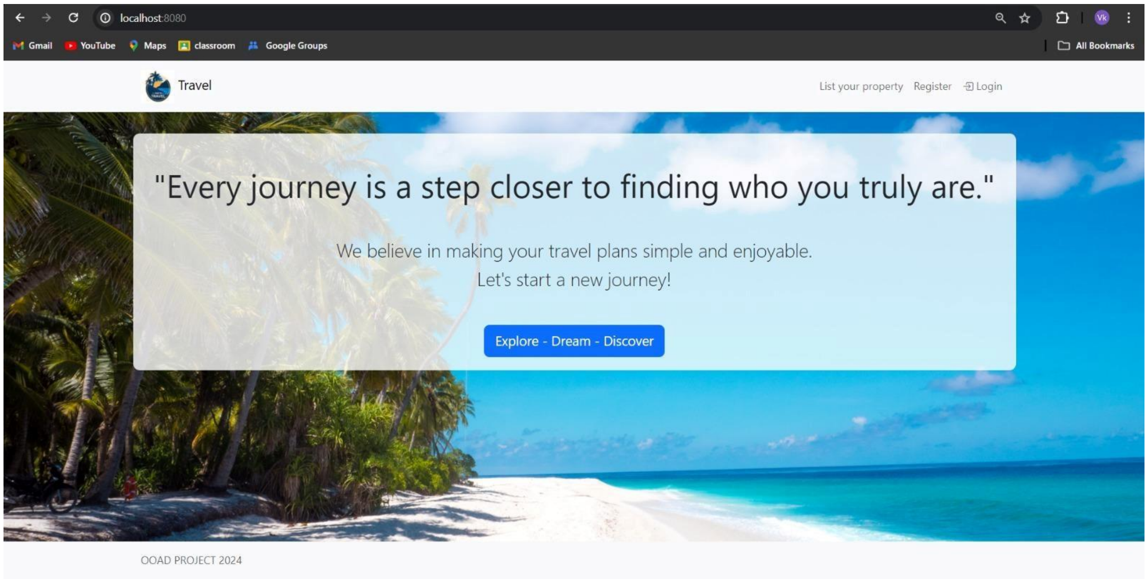
The Factory Method Pattern facilitates flexible and scalable user creation by encapsulating the creation logic within factory classes. This promotes code reuse and maintainability, allowing the system to accommodate changes or additions to user roles with minimal impact on existing code.

Decorator Design Patterns:

Validity: The isValid method within CardExpiryValidator is responsible for determining the validity of the provided credit card expiry date. It performs validation by checking if the date is in the correct format and if it is not in the past. This method encapsulates the validation logic.

Decorator Design Pattern: The CardExpiryValidator class implements the ConstraintValidator interface, which is a part of the decorator design pattern. This interface allows CardExpiryValidator to wrap the validation logic and apply it to fields annotated with CardExpiry. By implementing this interface, CardExpiryValidator can be seamlessly integrated into the validation framework of the project without directly modifying the core validation logic.

OUTPUT SCREENSHOTS

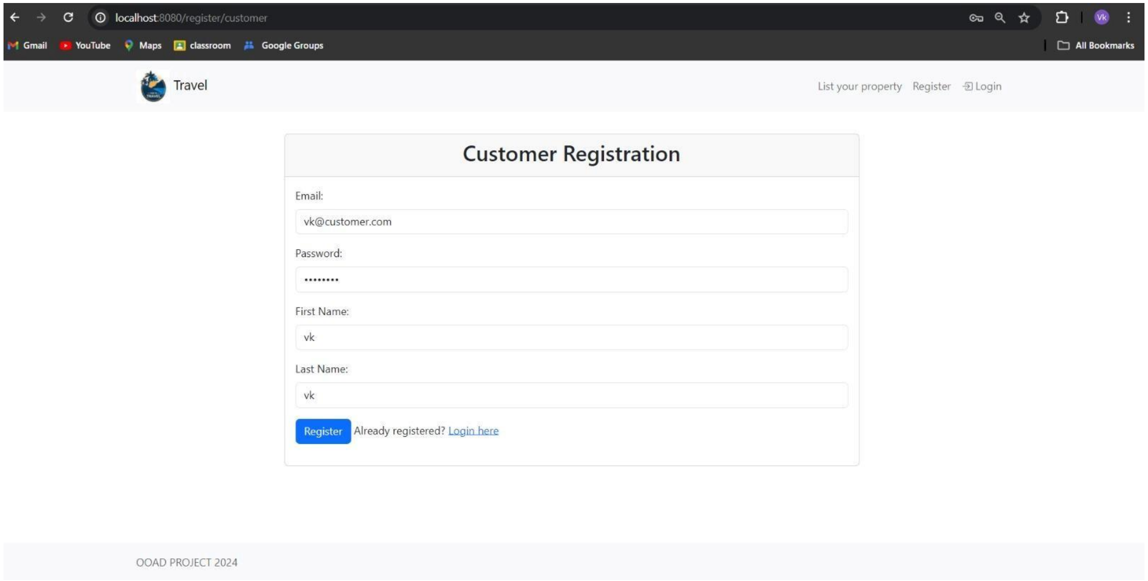


"Every journey is a step closer to finding who you truly are."

We believe in making your travel plans simple and enjoyable.
Let's start a new journey!

OOAD PROJECT 2024

Customer page



Customer Registration

Email:
vk@customer.com

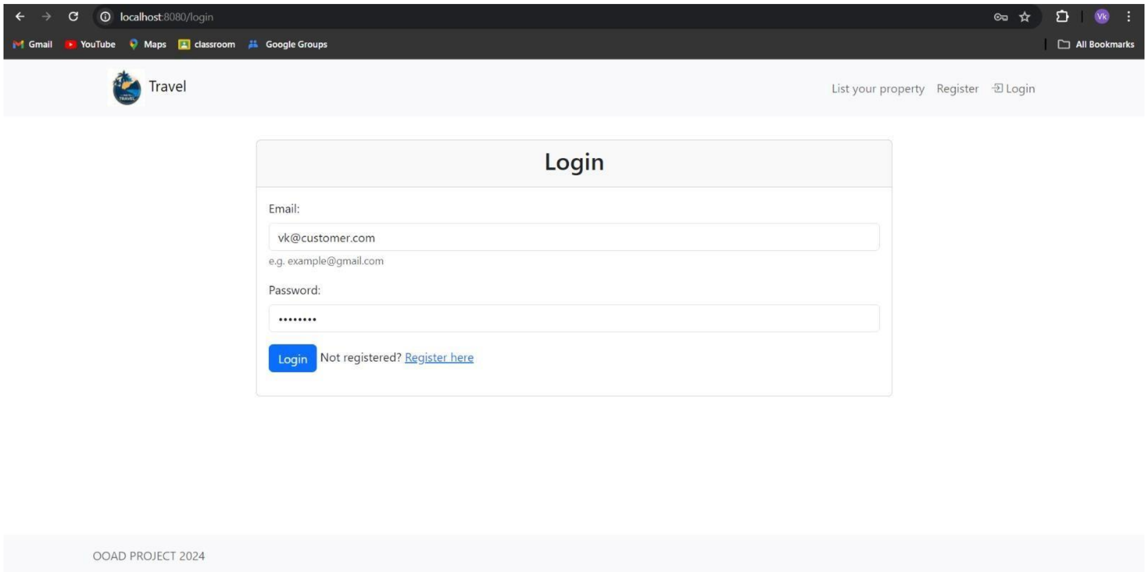
Password:
•••••••••

First Name:
vk

Last Name:
vk

Already registered? Login here

OOAD PROJECT 2024



Login

Email:

vk@customer.com

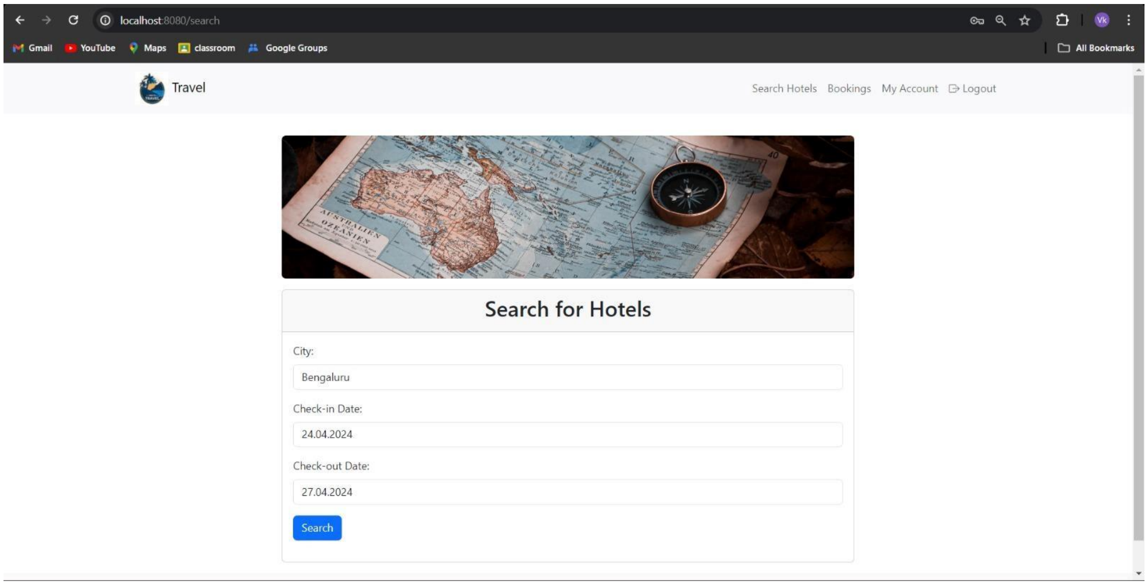
e.g. example@gmail.com

Password:

••••••••

Login Not registered? Register here

OOAD PROJECT 2024

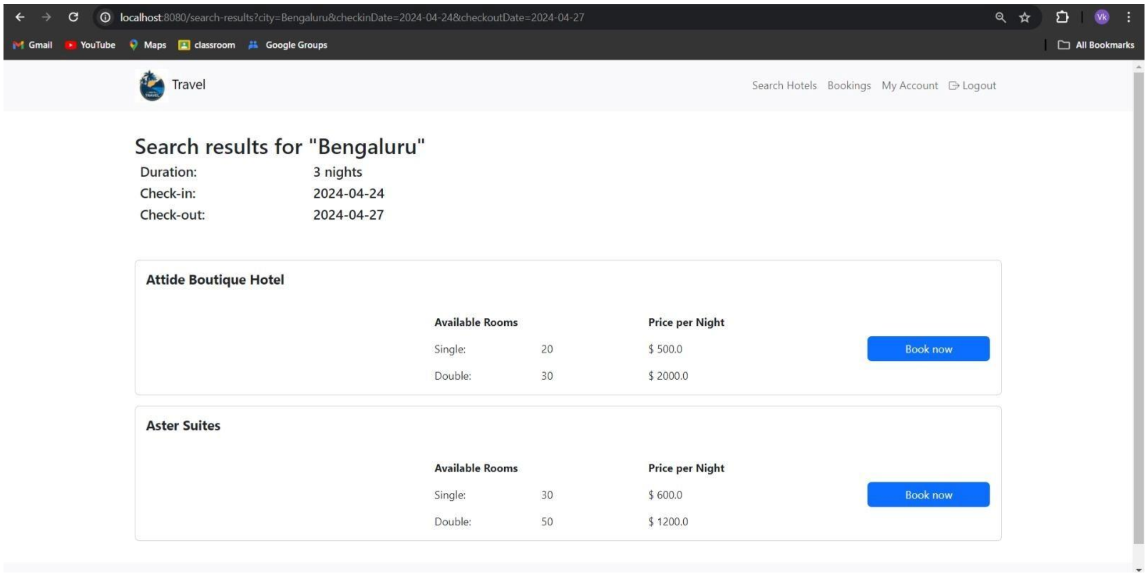


Search for Hotels

City:
Bengaluru

Check-in Date:
24.04.2024

Check-out Date:
27.04.2024



Search results for "Bengaluru"

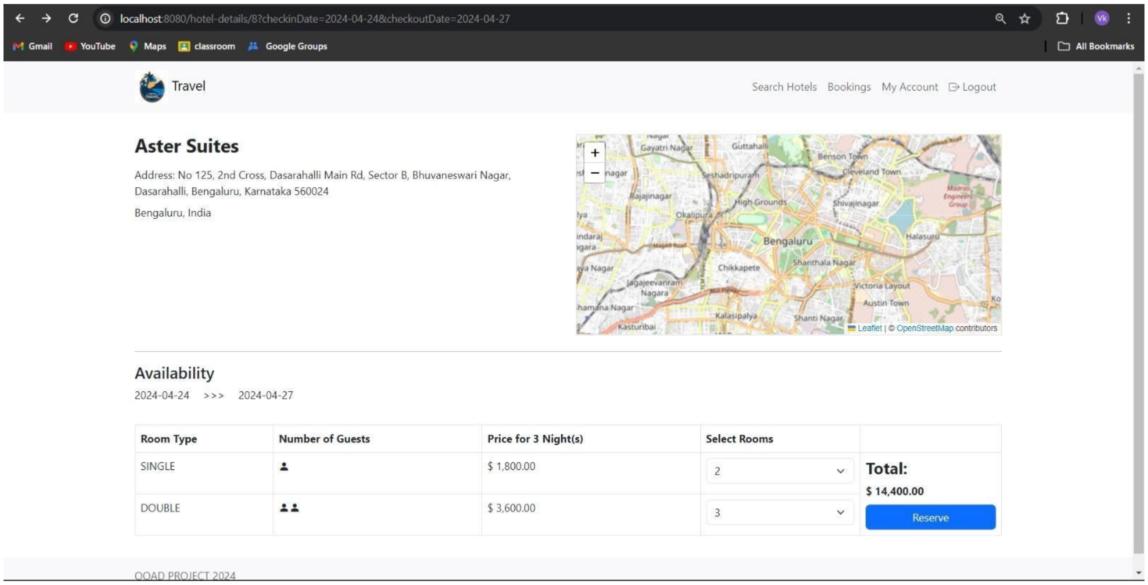
Duration:	3 nights
Check-in:	2024-04-24
Check-out:	2024-04-27

Attide Boutique Hotel

Available Rooms		Price per Night	
Single:	20	$ 500.0	Book now
Double:	30	$ 2000.0	

Aster Suites

Available Rooms		Price per Night	
Single:	30	$ 600.0	Book now
Double:	50	$ 1200.0	


localhost:8080/hotel-details/8?checkinDate=2024-04-24&checkoutDate=2024-04-27
Gmail
YouTube
Maps
classroom
Google Groups
All Bookmarks
Travel
Search Hotels
Bookings
My Account
Logout
Leaflet | © OpenStreetMap contributors

Aster Suites

Address: No 125, 2nd Cross, Dasarahalli Main Rd, Sector B, Bhuvaneswari Nagar, Dasarahalli, Bengaluru, Karnataka 560024

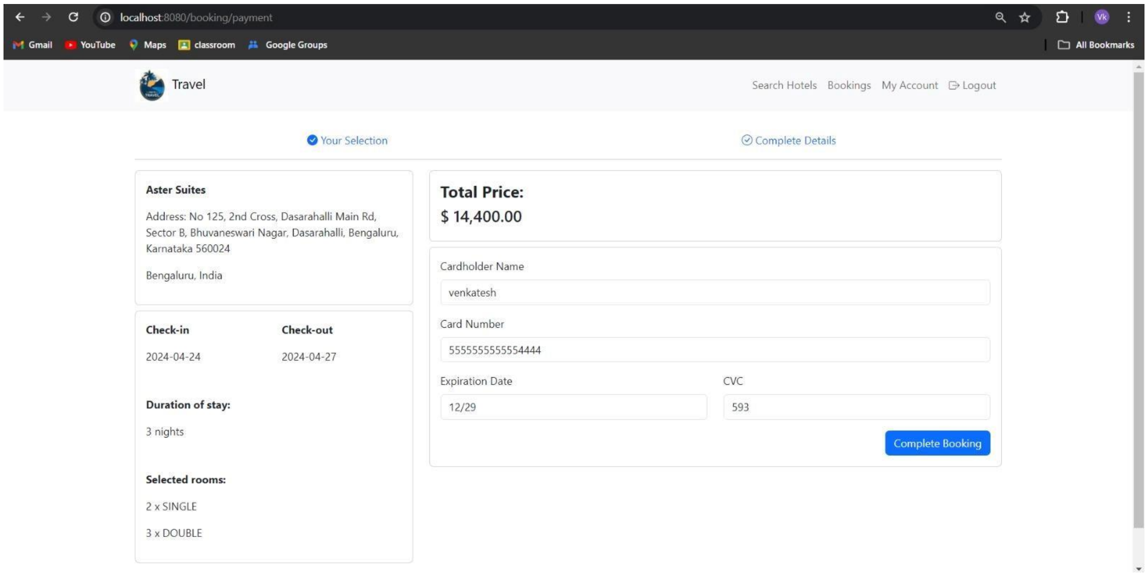
Bengaluru, India

Availability

2024-04-24 >>> 2024-04-27

Room Type	Number of Guests	Price for 3 Night(s)	Select Rooms	
SINGLE	👤	$ 1,800.00	2	**Total:** $ 14,400.00 Reserve
DOUBLE	👤👤	$ 3,600.00	3	

OOAD PROJECT 2024



Aster Suites

Address: No 125, 2nd Cross, Dasarahalli Main Rd, Sector B, Bhuvaneswari Nagar, Dasarahalli, Bengaluru, Karnataka 560024

Bengaluru, India

Check-in	Check-out
2024-04-24	2024-04-27

Duration of stay:

3 nights

Selected rooms:

2 x SINGLE

3 x DOUBLE

Total Price:
$ 14,400.00

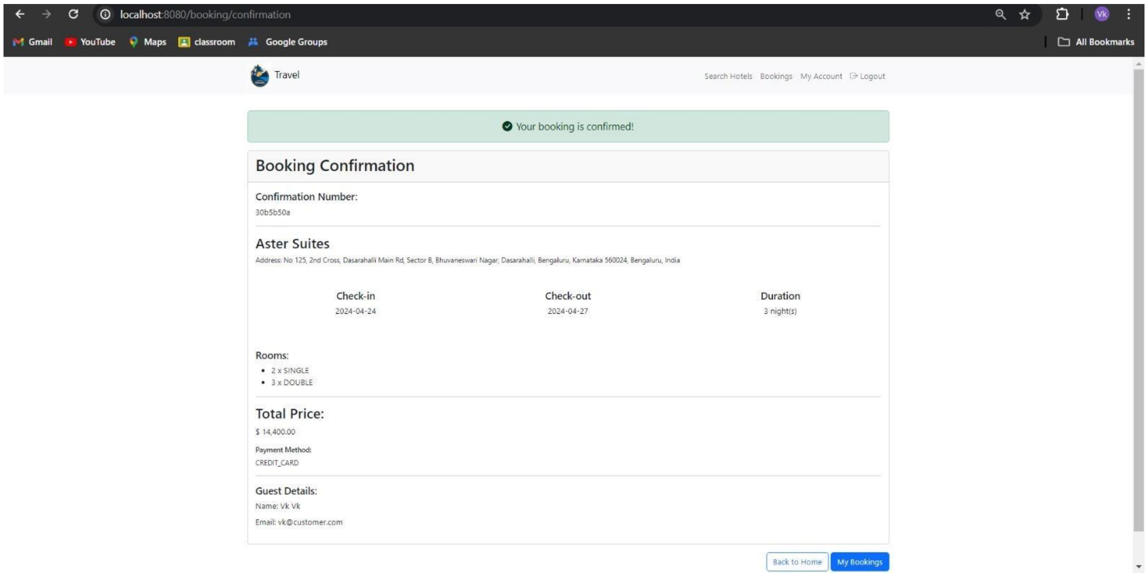
Cardholder Name: venkatesh

Card Number: 5555555555554444

Expiration Date: 12/29

CVC: 593

Complete Booking



✓ Your booking is confirmed!

Booking Confirmation

Confirmation Number:

30b5b50a

Aster Suites

Address: No 125, 2nd Cross, Dasarahalli Main Rd, Sector B, Bhuvaneswari Nagar, Dasarahalli, Bengaluru, Karnataka 560024, Bengaluru, India

Check-in	Check-out	Duration
2024-04-24	2024-04-27	3 night(s)

Rooms:

- 2 x SINGLE
- 3 x DOUBLE

Total Price:

$ 14,400.00

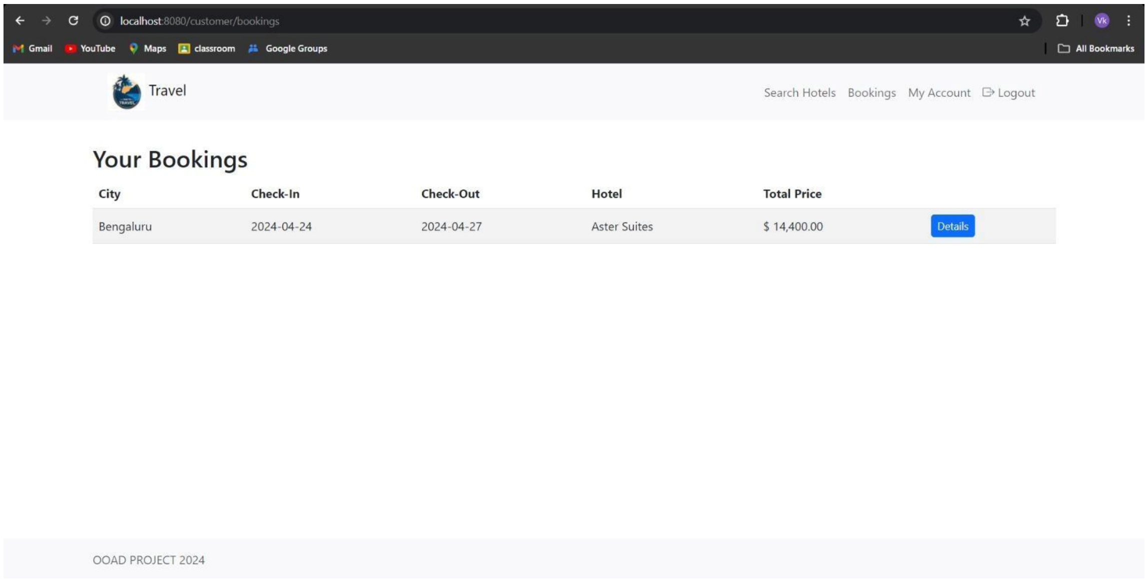
Payment Method:

CREDIT_CARD

Guest Details:

Name: Vk Vk

Email: vk@customer.com

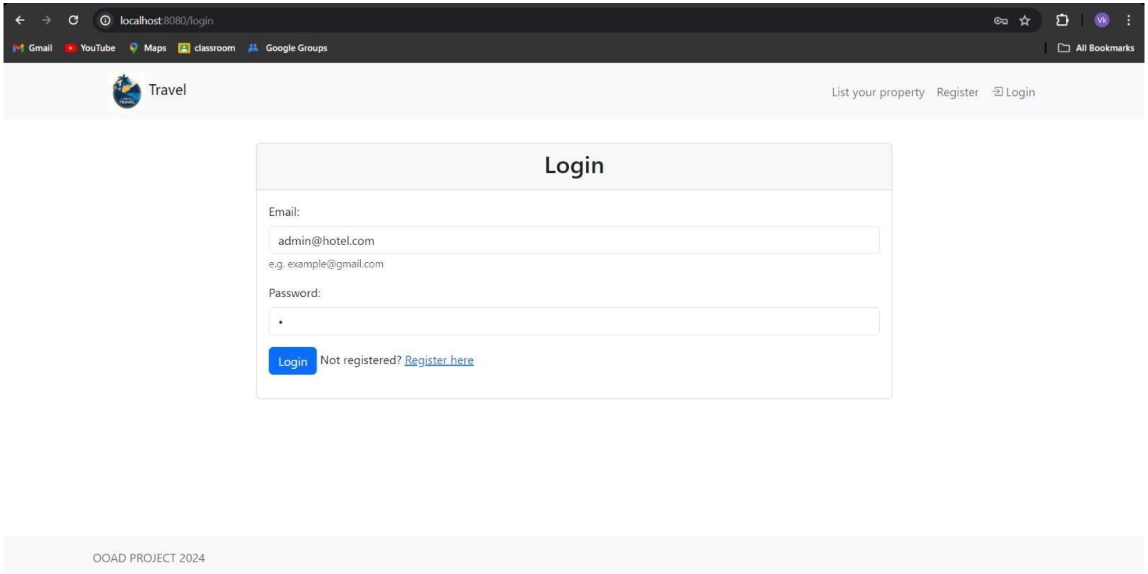


Your Bookings

City	Check-In	Check-Out	Hotel	Total Price	
Bengaluru	2024-04-24	2024-04-27	Aster Suites	$ 14,400.00	Details

OOAD PROJECT 2024

Admin page



localhost:8080/login

Login

Email:

admin@hotel.com

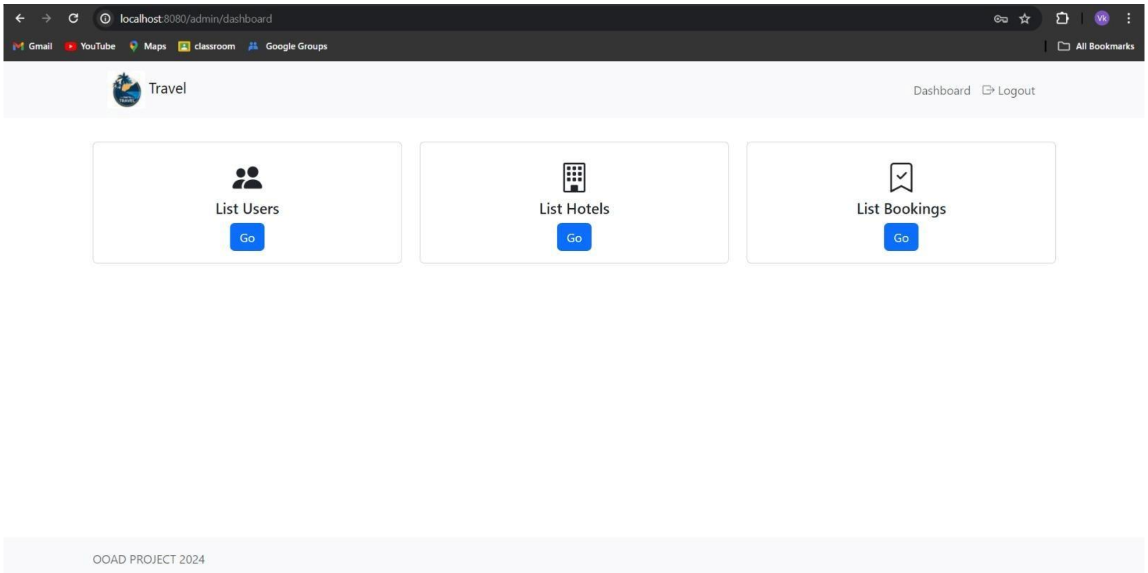
e.g. example@gmail.com

Password:

•

Login Not registered? Register here

OOAD PROJECT 2024



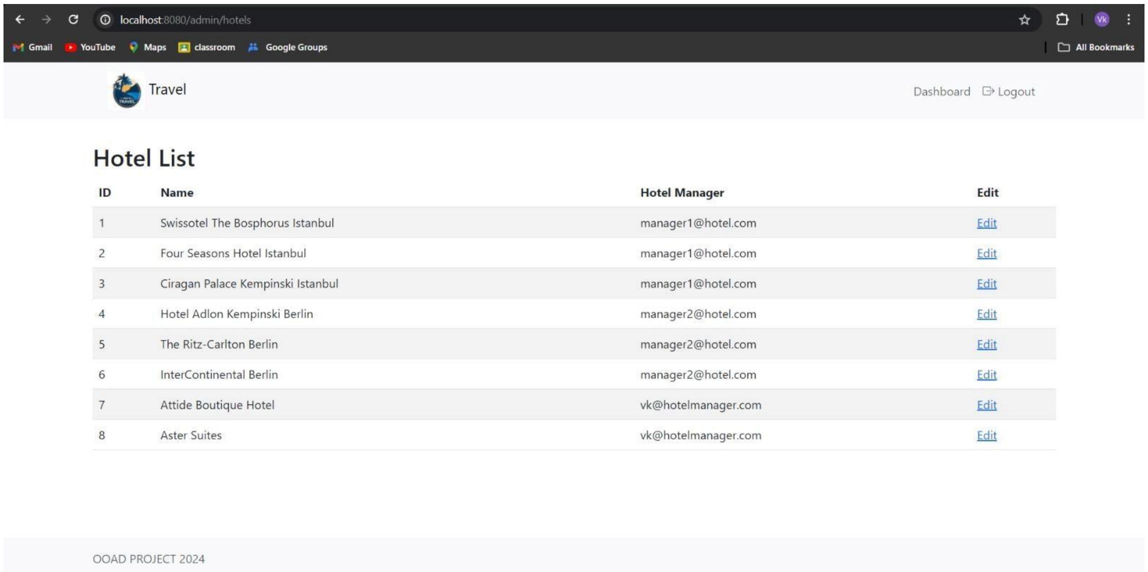
localhost:8080/admin/dashboard

List Users Go

List Hotels Go

List Bookings Go

OOAD PROJECT 2024

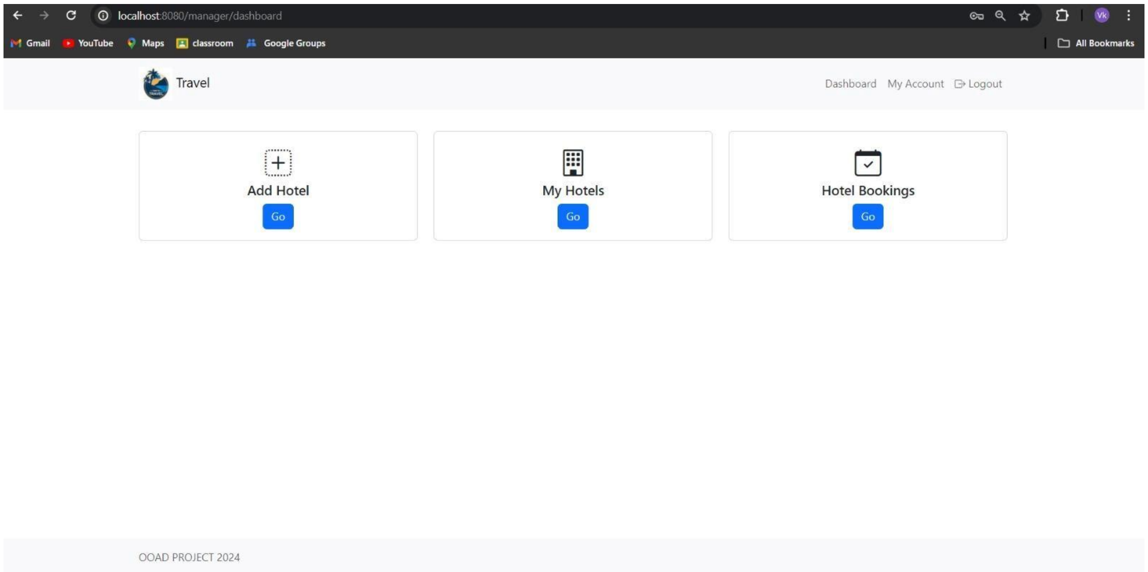


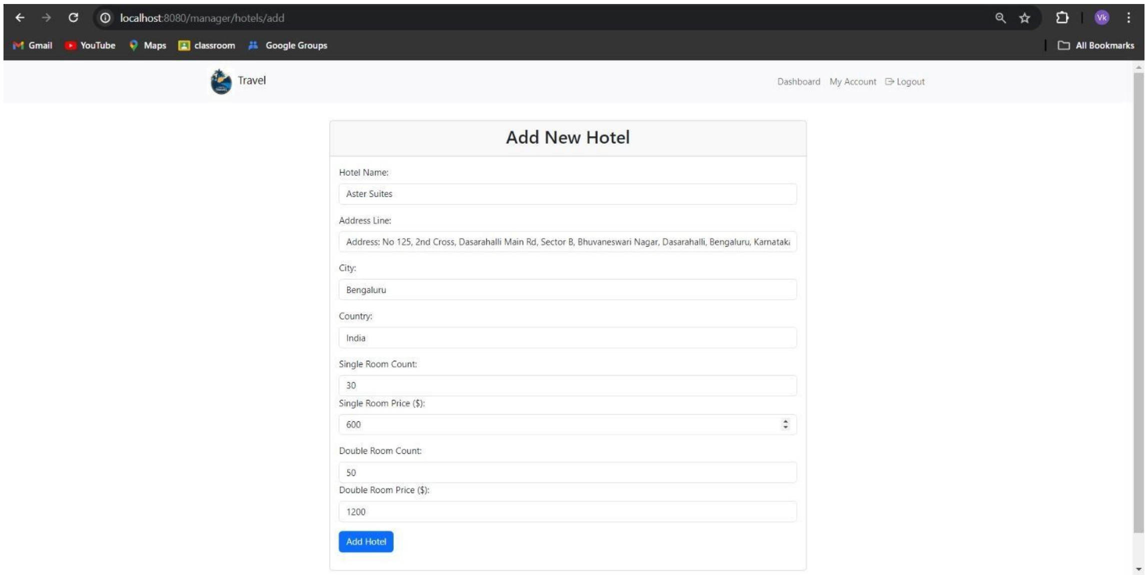
Hotel List

ID	Name	Hotel Manager	Edit
1	Swissotel The Bosphorus Istanbul	manager1@hotel.com	Edit
2	Four Seasons Hotel Istanbul	manager1@hotel.com	Edit
3	Ciragan Palace Kempinski Istanbul	manager1@hotel.com	Edit
4	Hotel Adlon Kempinski Berlin	manager2@hotel.com	Edit
5	The Ritz-Carlton Berlin	manager2@hotel.com	Edit
6	InterContinental Berlin	manager2@hotel.com	Edit
7	Attide Boutique Hotel	vk@hotelmanager.com	Edit
8	Aster Suites	vk@hotelmanager.com	Edit

OOAD PROJECT 2024

HOTEL MANAGER PAGE





Add New Hotel

Hotel Name:
Aster Suites

Address Line:
Address: No 125, 2nd Cross, Dasarahalli Main Rd, Sector B, Bhuvaneswari Nagar, Dasarahalli, Bengaluru, Karnataka

City:
Bengaluru

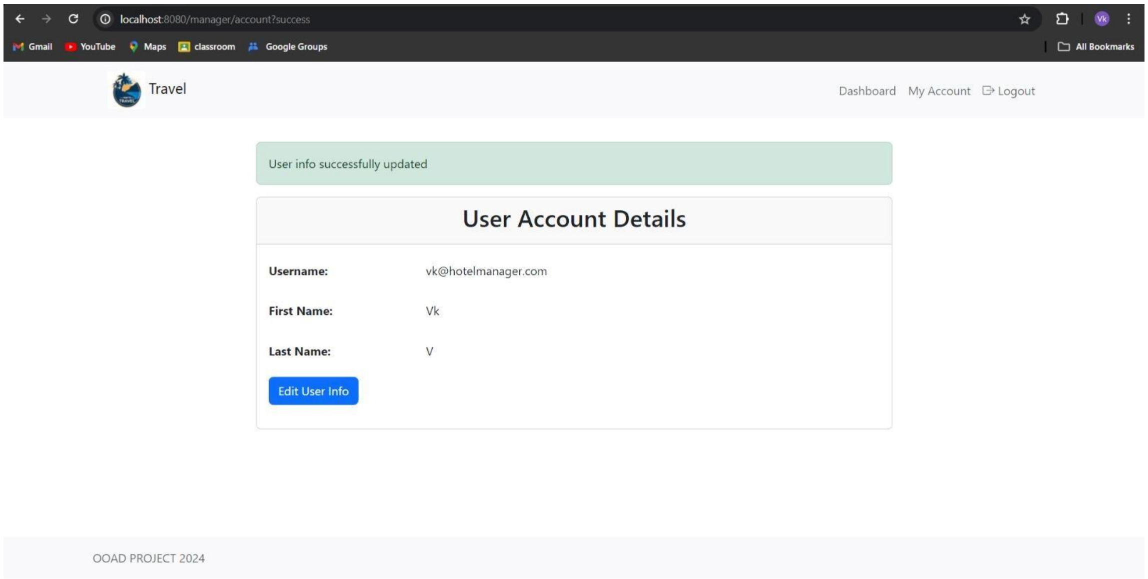
Country:
India

Single Room Count:
30

Single Room Price ($):
600

Double Room Count:
50

Double Room Price ($):
1200



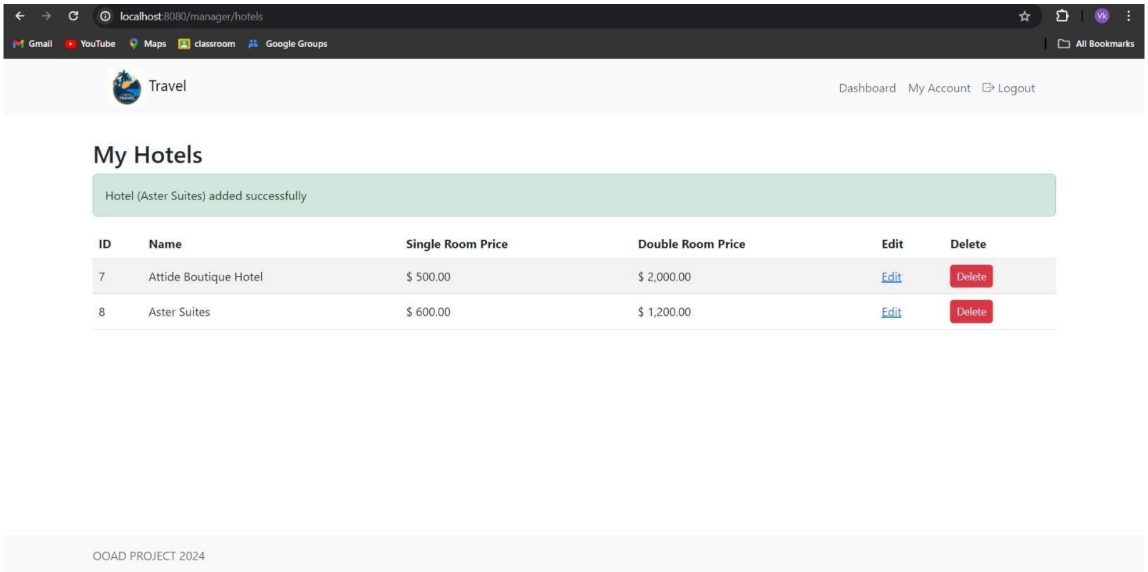
User info successfully updated

User Account Details

Username: vk@hotelmanager.com

First Name: Vk

Last Name: V

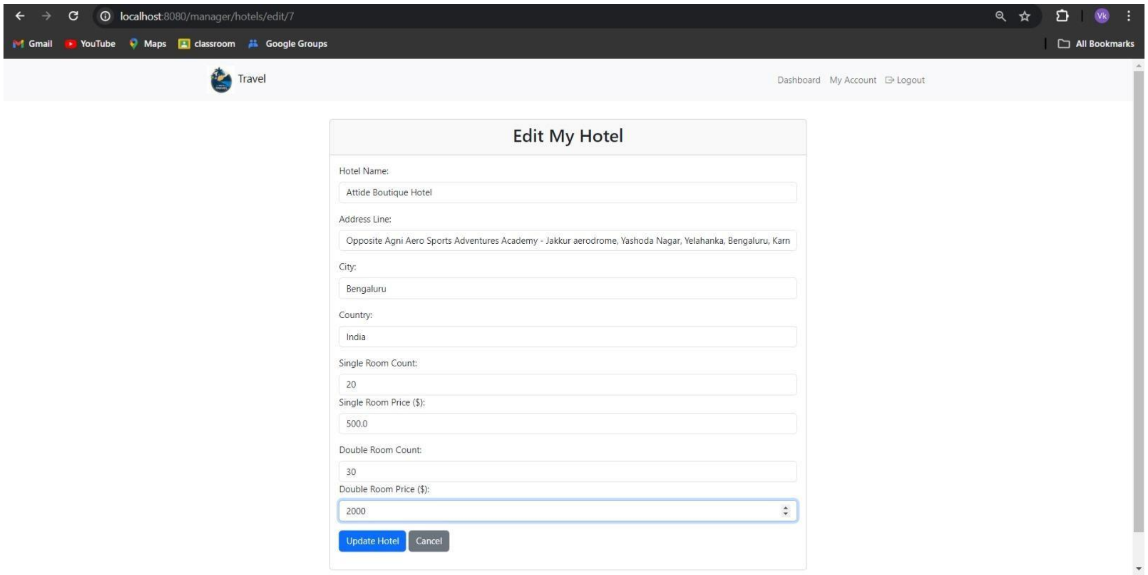


My Hotels

Hotel (Aster Suites) added successfully

ID	Name	Single Room Price	Double Room Price	Edit	Delete
7	Attide Boutique Hotel	$ 500.00	$ 2,000.00	Edit	Delete
8	Aster Suites	$ 600.00	$ 1,200.00	Edit	Delete

OOAD PROJECT 2024



Edit My Hotel

Hotel Name: Attide Boutique Hotel

Address Line: Opposite Agni Aero Sports Adventures Academy - Jakkur aerodrome, Yashoda Nagar, Yelahanka, Bengaluru, Karn

City: Bengaluru

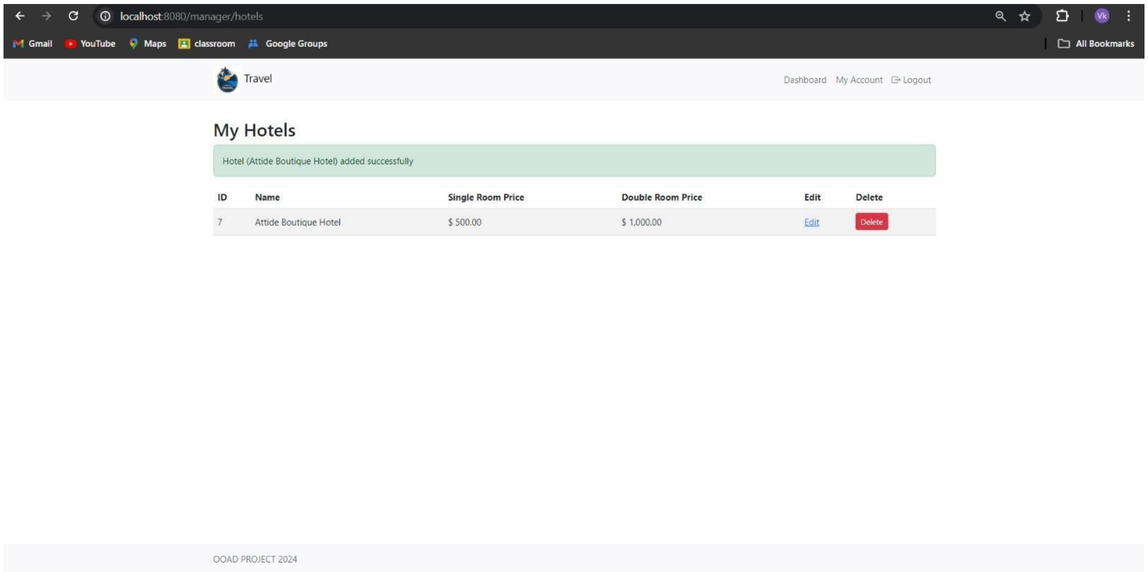
Country: India

Single Room Count: 20

Single Room Price ($): 500.0

Double Room Count: 30

Double Room Price ($): 2000



My Hotels

Hotel (Attide Boutique Hotel) added successfully

ID	Name	Single Room Price	Double Room Price	Edit	Delete
7	Attide Boutique Hotel	$ 500.00	$ 1,000.00	Edit	Delete

OOAD PROJECT 2024

WORK RESPONSIBILITY

K Virupakshi - User Registration
Registration Management
Payment

LSS Praneeth Kumar- Hotel Search &
Listing Booking
Management

Mani Shankar M - Hotel Management
Admin Panel.

Jyothiradithya D -Room Booking
Payment Processing

Github Repository Link:

https://github.com/DJ654312/HotelManagementSystem