IT314 Project
Guesthouse booking system

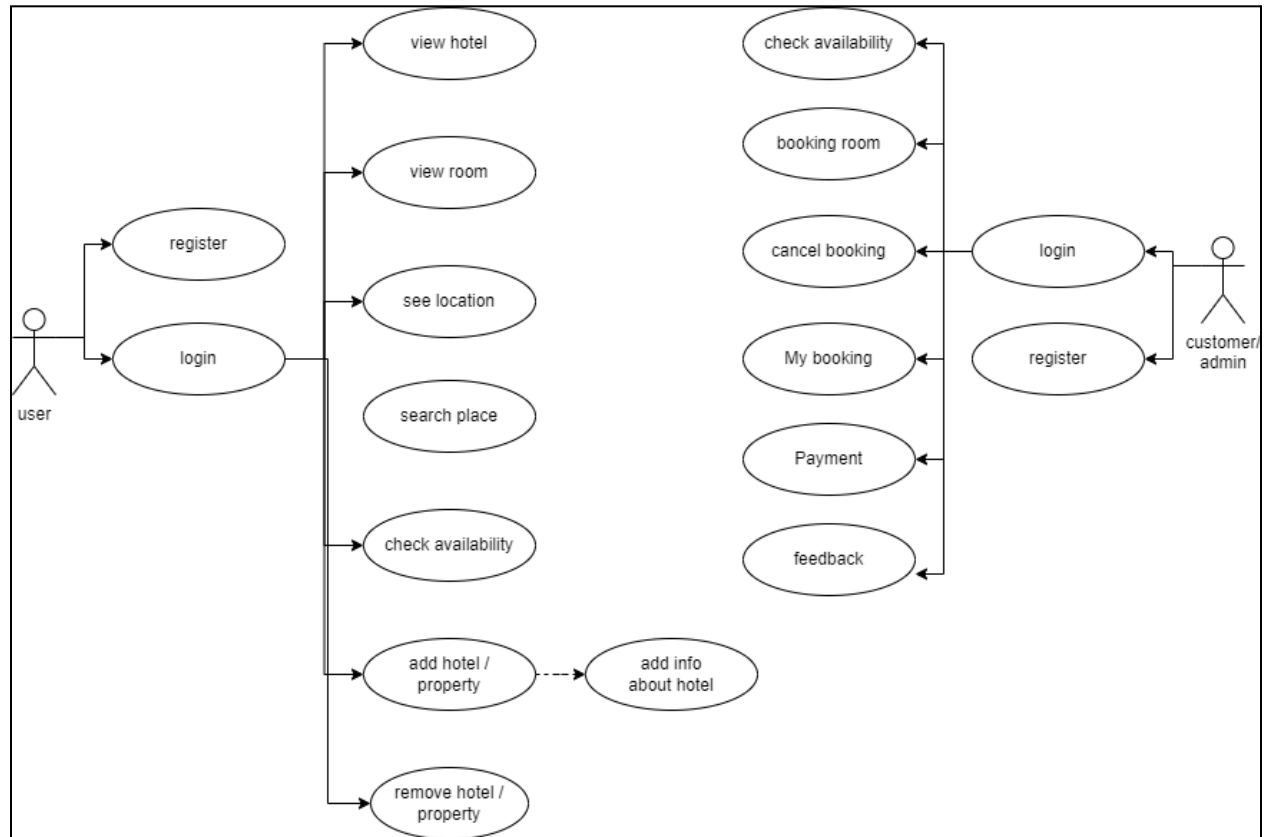<u>Software Requirements Specification (SRS):</u>

Introduction
Functional and Non-functional requirements

- **Functional requirements**:

1. **User Management:** Admins should be able to manage user accounts, reset passwords.

2. **Hotel/Guesthouse/Property Management:** Property owners should be able to register their property for booking/renting, with the admin able to verify documents and licenses.

3. **View hotel:** The system should filter the property/hotel list based on user filters.

4. **Room Management/service:** Admins should be able to add, edit, and delete rooms and vary room availability and pricing for different dates.

5. **Reservation Management:** Users should be able to apply for waiting lists, make bookings, verify documents, view, approve, and reject bookings, and receive email confirmations.

6. **Cancellation Management:** Cancellation should only be allowed if the user satisfies the hotel/guesthouse policy.

7. **Payment processing:** The system securely enters and stores payment information, and allows admins to view and track payments.

8. **User Support:** Users should be able to give feedback, and view and respond to customer support requests.

- **Non-functional requirements**:

1. **Performance:** The system should be quick, responsive, and able to handle high volumes of concurrent users and transactions.

2. **Scalability:** The system should be scalable and handle a large database of users and property details.

3. **Availability:** The system should have a high availability rate, minimize downtime, and have robust disaster recovery mechanisms.

4. **Security:** The system should have strong security measures to protect sensitive information and comply with data protection regulations.

5. **Usability:** The system should be user-friendly, intuitive, and accessible for those with disabilities.

6. **Interoperability:** The system should be compatible with a range of devices and browsers and integrate with other systems.

7. **Reliability:** The system should have a high level of reliability with minimal errors and bugs.

8. **Maintainability:** The system should be easily maintainable and have the capability to be improved over time.

Use Case Diagram:

Develop the use case textual description for each use case:

Use case	Apply Filter
Description	The system facilitates the customer to filter on the property like price and by Title.
Actors	Customer, The system
Pre Condition	The customer should have an active internet connection and he should give input in the filter bar.
Post Condition	The customer should be on the system page.
Flow	The customer will select the relevant filter options given below and then select apply filter.

Use case	Get Feedback
Description	It is the feature that will prompt the customer to give their valuable feedback at the time of checkout.
Actors	Customer, The system
Pre Condition	The customer should be on the system page
Post Condition	The system will ask for feedback in case of checkout.
Flow	The customer will get an option for feedback after checkout in the hostel.

Use case	Add Hotel Profile
Description	Adding a hotel profile is a feature of a guesthouse Booking system that allows owners or managers to provide important information about their property.
Actors	User, Hotel Owner
Pre Condition	for adding a hotel profile is having a guesthouse Booking system account and all necessary information about the guesthouse
Post Condition	for adding a hotel profile is that the guesthouse information is stored in the guesthouse Booking system, allowing customers to view and book the guesthouse online.

Flow	1.Log into the guesthouse Booking system 2.Navigate to My accommodation. 3.Fill in the necessary information about the guesthouse 4.Save the profile.
Alternative Flow	1.Log into the guesthouse Booking system 2.Navigate to My accommodation. 3.Encounter an error or missing information 4.Correct the error or provide the missing information 5.Save the profile

Use case	Cancellation
Description	Customers will be provided with the functionality to cancel the booking of rooms/reserve facilities.
Actors	Customer, System
Pre Condition	1.For cancellation a customer must login with the id with which he or she has booked a room. 2.Customers can cancel their booking a day before their booking date.
Post Condition	The system will remove your booking from the system .
Flow	1.Log into the guesthouse booking system. 2.Go to see my booking section and select the booking that he/she needs to cancel and click on cancel button.
Alternative Flow	If the customer fails to give correct login credentials then the system will prompt the customer with wrong credentials and ask to give correct credentials.

Use case	Make Payment
Description	The customer has to make the payment of their stay in the hotel. The bill is calculated according to their room type, services accessed, number of days of their stay, and other factors.
Actors	Customer, System
Pre Condition	1.The customer has selected a specific guest house or room to book and has proceeded to the checkout page. 2.The customer has selected a payment method and has entered the necessary payment information, such as credit card or bank

	account details. 3.The guesthouse Booking system has successfully connected with the selected payment gateway to process the payment.
Post Condition	1.The payment has been successfully processed, and the customer has received a payment confirmation. 2.The guest house reservation has been confirmed and added to the guest house Booking system. 3.The customer can access the reservation details and confirmation information in the guesthouse Booking system.
Flow	1. Customer click on the book button. 2. Ask for card details. 3. To confirm payment click on the pay button.

● Design document

Class diagram

user

Name : string

email : string

password : string

---

feedback()

register()

log In()

booking class

Name : string

phone : number

price : int

Place : place

user : user

CheckIn : Date

CheckOut :Date

Numofguest : int

---

booking()
cancel booking()
payment()

place

title : string

address : string

city : string

state : string

country : string

photos : string

description : string

perks : string

extraInfo : string

checkIn : string

checkOut : string

maxGuests : int

price : int

---

search()

edit place()

addPlace()

remove()

1   m   m   m