

IT314 Project

Guesthouse booking system

GROUP: 16**Lab-06**

Members

Student ID	NAME
202001126	PATEL AYUSH MUKESHBHAI
202001129	DEEP JAIMESH SHAH
202001138	GANDHI SOMIN
202001146	MAKADIYA MUDIT BHAVESHBHAI
202001156	KHAMAR AYUSH KINCHITKUMAR
202001159	RADADIYA DARPAN VITTHALBHAI
202001161	GAJERA AYUSH KAMLESHBHAI
202001165	PATEL HARSH BHARATBHAI
202001170	RAVAL KEVIN KIRITKUMAR
202001175	ANKIT LOCHAN

Domain Analysis Models

Introduction:

- The domain of a guesthouse booking system would be in the hospitality industry, specifically in the sub-domain of lodging and accommodations. The motivation behind developing a guesthouse booking system is to simplify the process of booking guesthouse accommodations for both guests and owners.

General Information About the Domain:

- Guests usually search for available accommodations, make reservations, and pay for their stay wherever they are on any trip. While guest house owners usually have to manage reservations, payments, and room availability in order to carry out their business. Guesthouse booking system helps by automating the booking process which can help increase efficiency, reduce errors, and improve the overall guest experience. It also allows owners to better manage their business, streamline operations, and maximize revenue by easily tracking room occupancy and rates. Additionally, a booking system can help enhance security and privacy by securely storing guest information and processing payments in a safe and reliable manner.
- Such a system will have users from all over the world, leading to a high possibility of large amounts of requests hitting the website simultaneously. So the website will need efficient scalability to handle such high load levels.

USERS:

- The end-user interacting with the front-end and database, responding to backend calls, is a user involved in system interaction. The end-user requires no training to use the system for the first time. When API calls to the backend are made, the backend system will make the appropriate database calls.

Existing Software:

- The features of existing guesthouse booking systems can vary depending on the specific platform and its target market. However, some common features of guesthouse booking systems include:
 - **Room availability and booking:** This feature allows guests to search for available rooms and make a reservation.
 - **Online payment processing:** Most guesthouse booking systems allow guests to make payments online using a credit card, PayPal, or other payment gateways.
 - **Room inventory management:** This feature allows guesthouse owners to manage their room inventory, update room availability, and block rooms for maintenance or repairs.
 - **Guest communication:** This feature allows guesthouse owners to communicate with guests before, during, and after their stay, through email, SMS, or in-app messaging.
 - **Review and rating management:** This feature allows guests to leave reviews and ratings of their stay, which can help improve the reputation of the guesthouse and attract more bookings.
 - **Integration with other systems:** Many guesthouse booking systems integrate with other hospitality systems such as property management software, customer relationship management tools, and marketing automation platforms.

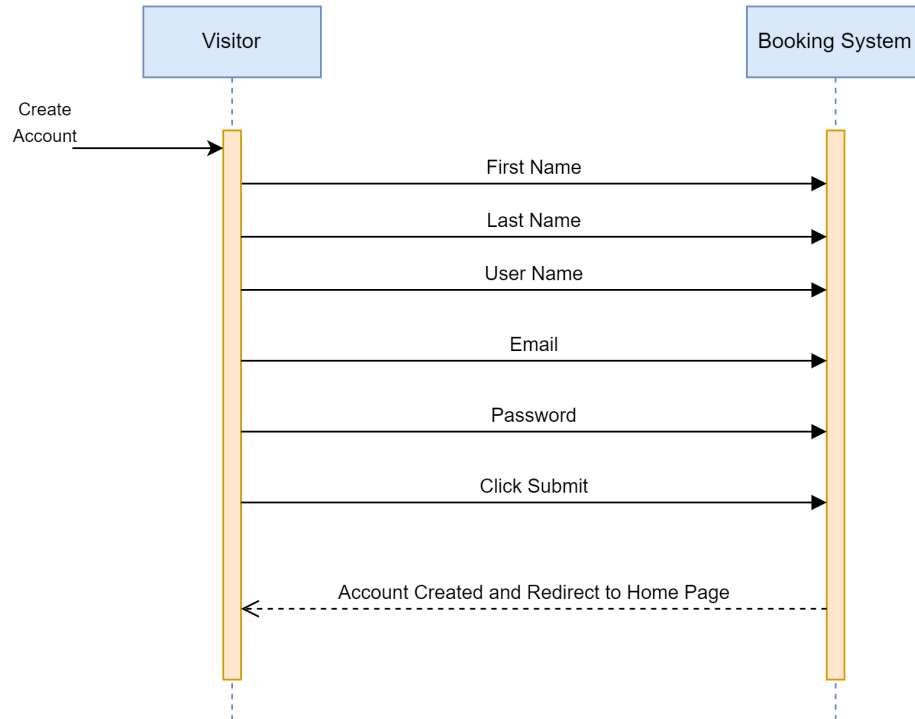
Identify boundary, entity, control object

- **Entities:**
 - **Customers:** Individuals who want to book a room in the guesthouse.
 - **Rooms:** The physical spaces available for guests to book and stay in.
 - **Staff/Manager:** The employees responsible for managing the guest house, including front desk staff, housekeeping, and maintenance personnel.
- **Processes:**
 - **Reservations:** Booking a particular room for a guest based on the availability and guest's preferences
 - **Room allocation:** Once a guest has made a booking request, the system will allocate an available room based on the guest's preferences and room availability.

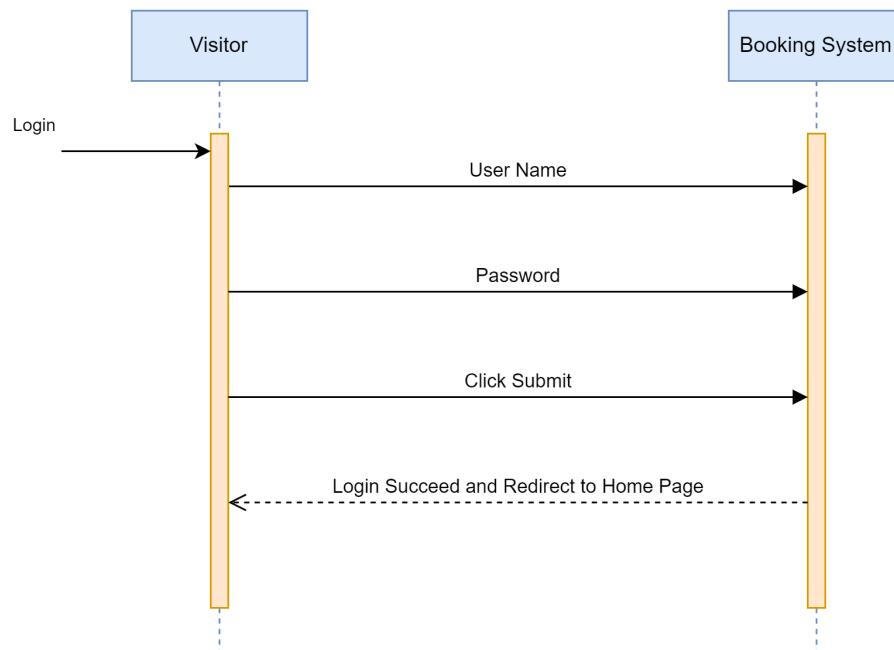
- **Payment:** The methods and systems used to collect payment from guests, including credit cards, cash, and online payment platforms.
 - **Availability:** The number of rooms that are available for booking on a particular date and time.
 - **Check-in and check-out:** The check-in process involves verifying the guest's identity and confirming the details of the reservation. The system will update the room status to occupied once the guest has checked in. When the guest is ready to leave, they will complete the check-out process, which involves settling any outstanding balances, returning the room key, and confirming that the room has been vacated. The system will update the room status to available once the guest has checked out.
 - **Reviews:** Feedback provided by guests about their stay at the guesthouse, which can be used to improve the quality of service and attract new customers.
- **Relationships:**
 - **Guest and booking:** A guest can make multiple bookings, but each booking must be associated with a unique guest. This relationship helps ensure that guest information is accurately tracked and that bookings are properly attributed to the right person.
 - **Room and booking:** Each booking must be associated with a specific room in the guesthouse. This relationship helps ensure that the guesthouse knows which rooms are occupied and which are available.
 - **Room and availability:** The availability of a room must be tracked in the system to prevent overbooking. This relationship helps ensure that guests can only book rooms that are available.
 - **Guest and payment:** Each booking must be associated with a payment made by the guest. This relationship helps ensure that the guesthouse is paid for its services.
 - **Staff and booking:** Staff members must be able to view and manage bookings in the system. This relationship helps ensure that staff can easily access and update booking information as needed.

Sequence Diagrams:

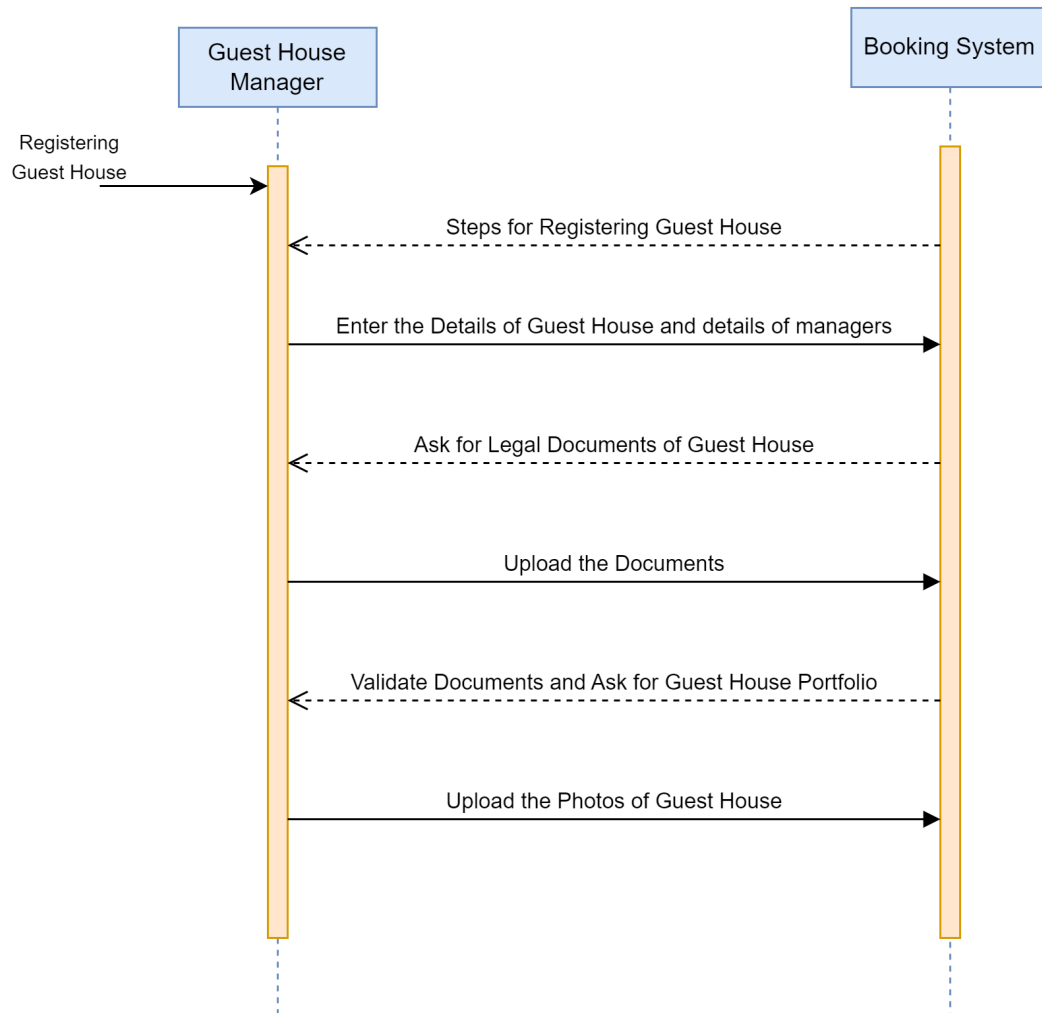
- **Creating Account for Random Visitor**



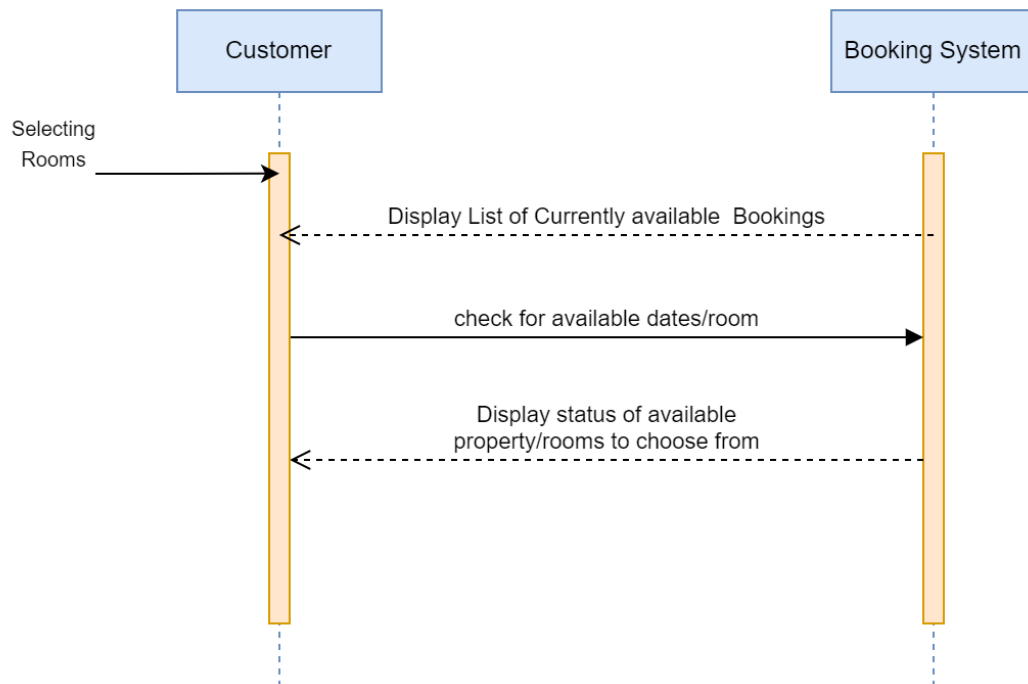
- **Login Process for Customer**



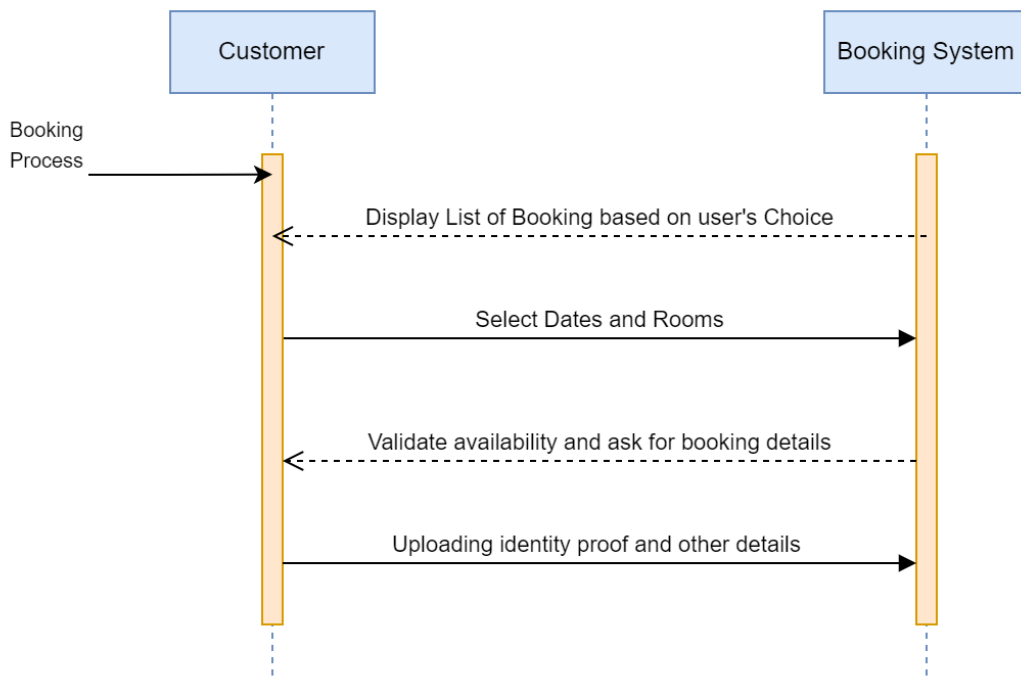
- **Registering Guest House**



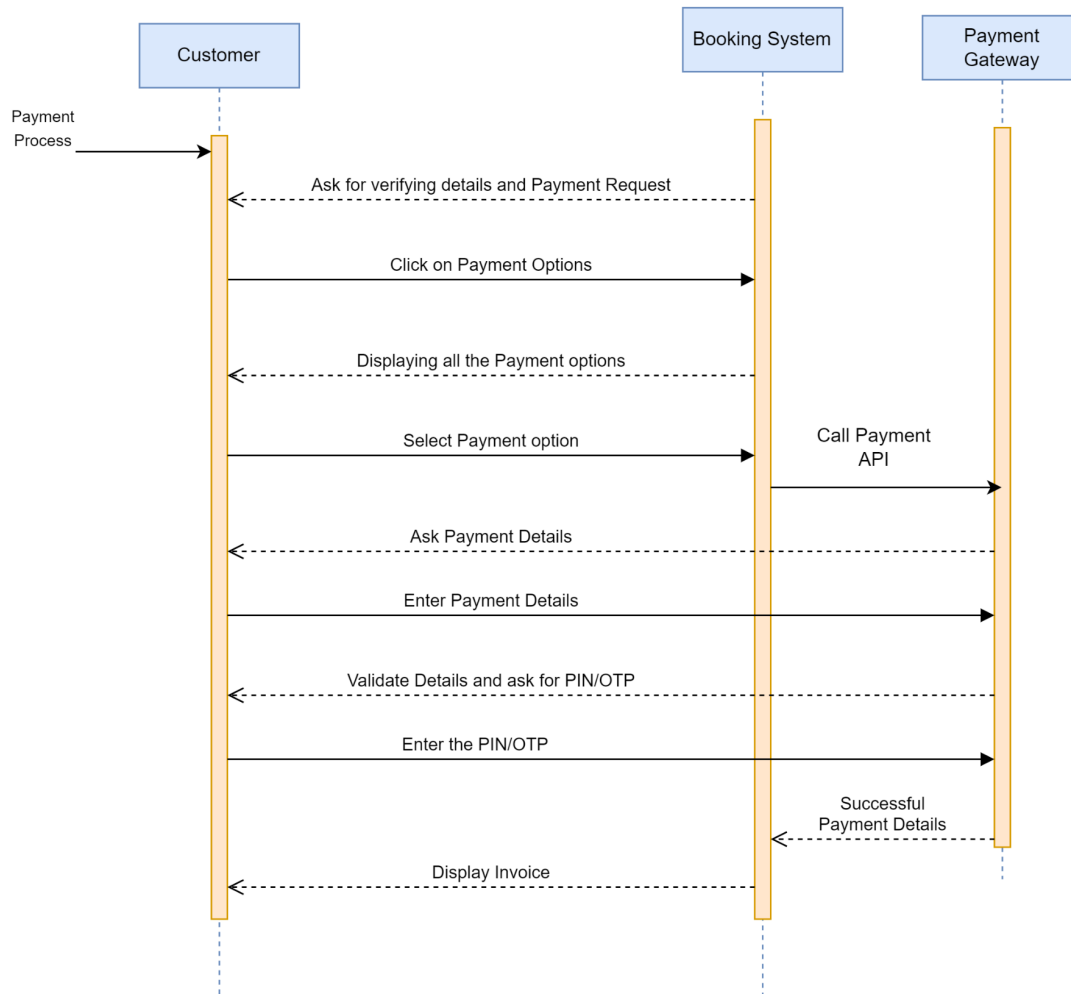
- Looking for Available Bookings



- Booking Rooms/Property



• Booking's Payment Process



Class Diagram:

• Classes of Hotel Booking System Class Diagram:

- **Payments Class:** Manage all the operations of Payments
- **Booking Class:** Manage all the operations of Booking
- **Customers Class:** Manage all the operations of Customers
- **Hotel Class:** Manage all the operations of Hotel
- **Rooms Class:** Manage all the operations of Rooms
- **Hotel Registration class:** Manage all the states and files regarding hotel registration
- **Feedback class:** store data regarding each feedback.

● **Classes and their methods of Hotel Booking System Class**

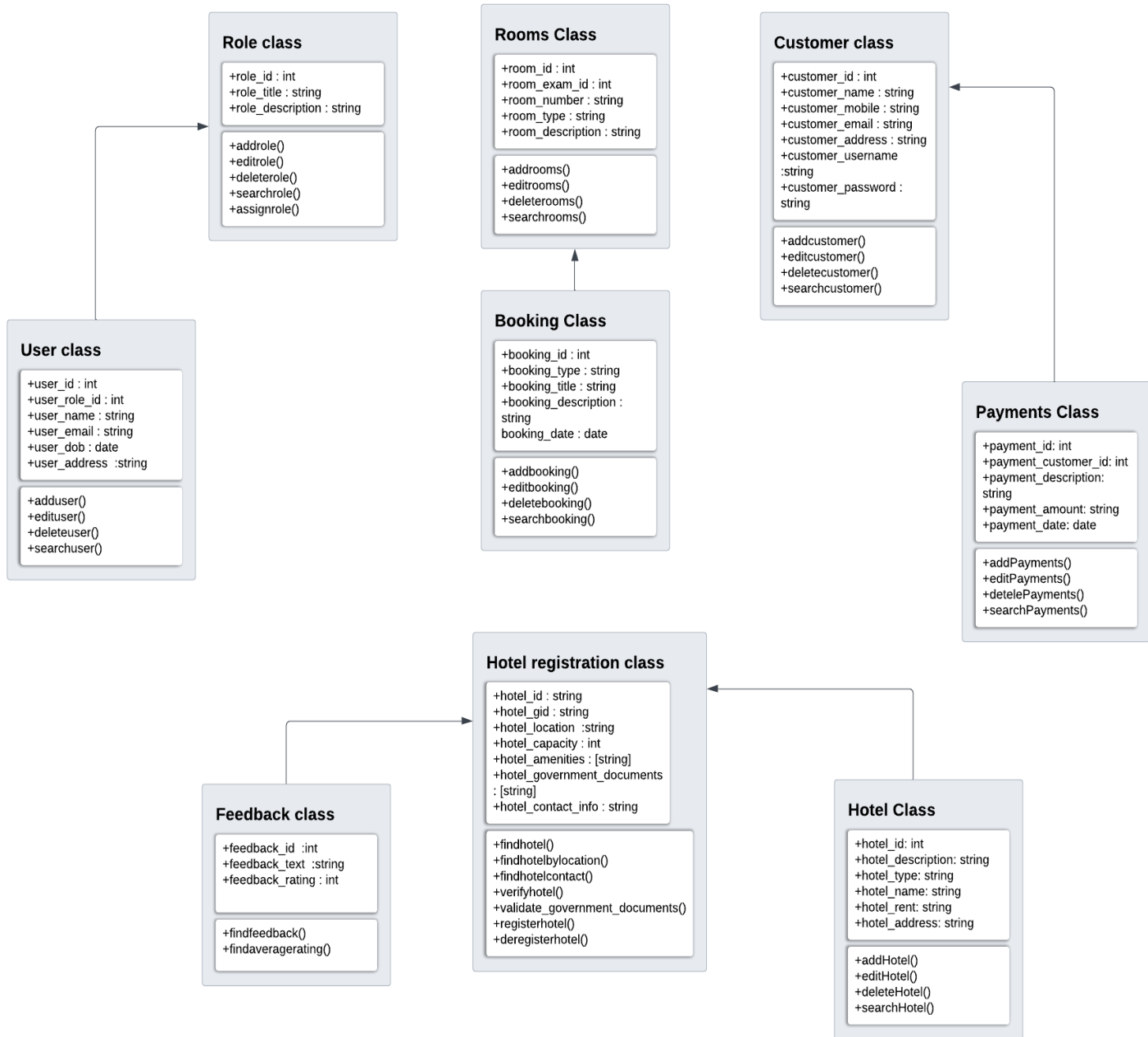
Diagram:

- **Payments Methods:** add payments(), edit payments(), delete payments(), updatePayments(), savePayments(), searchPayments()
- **Booking Methods:** addBooking(), editBooking(), deleteBooking(), updateBooking(), saveBooking(), searchBooking()
- **Customers Methods:** addCustomers(), editCustomers(), deleteCustomers(), updateCustomers(), saveCustomers(), searchCustomers()
- **Hotel Methods:** addHotel(), editHotel(), deleteHotel(), updateHotel(), saveHotel(), searchHotel()
- **Rooms Methods:** addRooms(), editRooms(), deleteRooms(), updateRooms(), saveRooms(), searchRooms()
- **Hotel Registration methods:** findhotel(), findhotelbylocation(), findhotelcontact(), verifyhotel(), validate_government_documents(), registerhotel(), deregisterhotel()
- **Feedback methods:** findfeedback(), findaverageating()

● **Classes and their attributes of Hotel Booking System Class**

Diagram:

- **Payments Attributes:** payment_id, payment_customer_id, payment date, payment_ amount, payment_description
- **Booking Attributes:** booking_id, booking title, booking_type, booking date, booking description
- **Customers Attributes:** customer_id, customer_name, customer_ mobile, customer_email, customer_username, customer_password, customer address
- **Hotel Attributes:** hotel_id, hotel_name, hotel_type, hotel_ rent, hotel_ description, hotel_address
- **Rooms Attributes:** room_id, room_ hotel_ id, room_number, room category, room_type, room_description
- **Feedback attributes:** feedback_id :int, feedback_text :string, feedback_rating : int.
- **Hotel registration attributes:** hotel_id : string, hotel_gid : string, hotel_location :string, hotel_capacity : int, hotel_amenities : [string], hotel_government_documents : [string], hotel_contact_info : string



Identify the design goals:

The design goals for a guesthouse booking system :

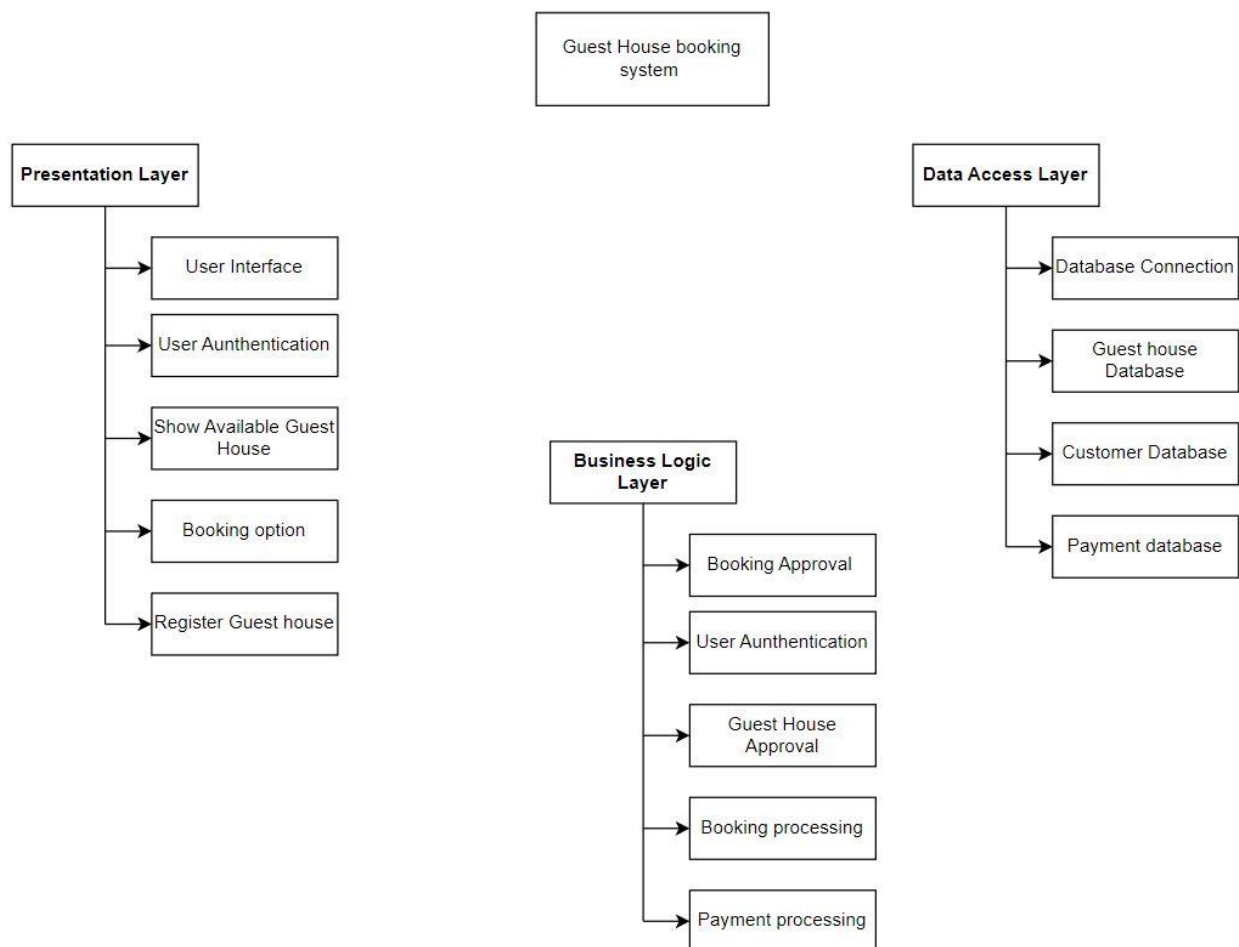
- **Ease of use:** The system should be easy to use for guests who are booking rooms, as well as for staff and business owners who are managing bookings, reservations and registering their hotels .
- **Efficiency:** In order to make the procedure easy and hassle-free for both guests and staff, the system's ability to handle bookings, reservations, and payments must be effective.
- **Reliability:** In terms of processing data securely, ensuring the accuracy of booking information and avoiding errors or conflicts the system should be reliable.
- **Scalability:** The system should be able to handle a large amount of reservations and visitors and scale up or down in response to demand.
- **Flexibility:** The system should be flexible enough to handle various booking scenarios (such as single or multiple rooms, various room kinds, etc.) and provide customizing booking options.
- **Integration:** The system should be able to integrate with other systems such as payment gateways, accounting software.
- **Accessibility:** All users should be able to access the system, regardless of their level of technical expertise or physical ability and accessibility standards should be followed.

Create a high level system design

Choose Architecture:

- This application will use a client-server architecture. We will use a 3 layer application architecture that consists of a presentation tier, an application tier and a data tier. The data tier stores information, the application tier handles logic, and the presentation tier is a graphical user interface (GUI).

- **Presentation Layer:** This layer is distributed to a computing device using a web browser or a web-based application and is constructed with ReactJS. Application programme interface (API) calls are the primary communication between the presentation tier and the other levels.
- **Business Logic Layer:** Also called the application layer, is written in Node.js and contains the business logic that supports the application's core functions. This choice is because it allows the app to be used by many users and provides flexibility for future development.
- **Data Access Layer:** A database and software for controlling read and write access to a database make up the data access layer. We will be using MongoDB for this layer.



Identify Subsystems:

- A layered architecture technique can be used to create the architecture of a guesthouse reservation system, with each subsystem standing in for a different layer. Flexible and scalable communication across the layers is possible because of clearly defined interfaces and APIs.

The high-level system design for a guesthouse booking system can be divided into four subsystems, each responsible for a specific set of functions as shown below:

