**HOTEL BOOKING MANAGEMENT SYSTEM**

### **PREPARED FOR**

JEE Cloud Training Program

Capgemini Technology Services

### **PREPARED BY**

Avantika Singh

Saurabh Potdar

Bhrigu Garg

**Introduction:**

The Hotel Booking Management System is a Java-based solution for automating major operations of hotel room booking. It contains user-friendly features for customers to search hotel rooms based on check-in date, check-out date and availability of rooms. The user can then choose a room type and initiate his booking. All the bookings are approved by the admin prior to confirming the booking status. The admin can add, update or delete a hotel from the system. He can also add, delete or update a room from the system.

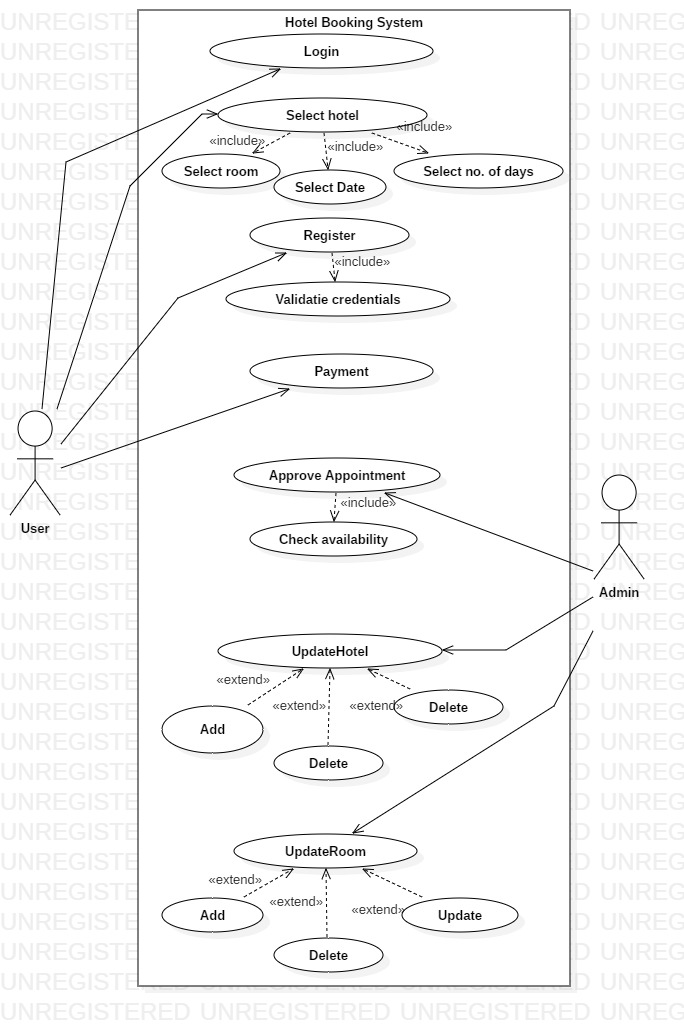
Following is the functionality provided by the system:

There are two categories of people who would access the system: customer and administrator. Each of these would have some exclusive privileges.

1. The customer can:
   1. Create his user account.
   2. Login into the application.
   3. Check for available rooms in a hotel.
   4. Make a booking.
   5. Cancel a booking.
2. The administrator can:
   1. Login into the application.
   2. Add a hotel or a room in a hotel.
   3. Update a hotel or a room in a hotel.
   4. Delete a hotel or a room in a hotel.

**Use Case view:**

The following diagram represents the interaction of the customer and the administrator with the system. It also shows the various use cases in which both types of users are involved. It provides a high-level view of the working of the system. It is also providing a requirement analysis for the system.



**Class and Methods Description:**

We have made a few assumptions with respect to the application, which are:

1. Admin and User are different classes, and Admin class extends User class.
2. Every needs to enter a Unique login ID which is of length 10.
3. The User Password is String of length 10.

The application will consist of the following classes, the utility of which have been described below:

1. **User:** This class stores the details of user:

Attributes:

userId: String

userName: String

userEmail: String

userDob: String

userPhone: String

Methods: -

*logIn()*: User will login if credentials are correct.

*logOut()*: Removes all the session related information of a user.

makeBooking(Hotel,Date,Room): Will return the booking ID.

*getBookingId(User user):* Will return the booking ID as Long.

The following methods will only be allowed for user.

1. **Administrator**. :- This class extends user class.

Attributes:

*addHotel(Hotel):* Adds a hotel.

*removeHotel(Hotel):* Removes a hotel.

*updateHotel(Hotel):* Updates a hotel.

*addRoom(Room):* Adds a Room.

*updateRoom(Room):* Updates room

*deleteRoom(Room):* Deletes room

1. **Booking Management**: This class stores the details of a booking made by a particular userId.

Attributes:

bookingId: String

bookingStatus: String

bookingDate: DateTime

checkIn: DateTime

checkOut: DateTime

bookingCost: Double

hotelList: List

Methods:

getAdminApproval(): Boolean:-

Asks admin to approve the booking and returns a boolean value.

*viewBooking(): List<User>* :-

Returns a list of all the users who made the booking.

generateReport(): List:-

Generates a report which contains the list of all the bookings made in every hotel.

1. **Hotel**: This class stores information about the hotels.

Attributes:

hotelId: String

hotelName: String

hotelAddress: String

roomList: Hashmap

1. **Room**: This class stores all the details of a room.

Attributes:

roomId: String

roomType: String

roomRent: Double

roomNumber: String

roomDescription: String

1. **Payment:** This class stores all payment details of the customer.

Attributes:

user: User

paymentStatus: enum

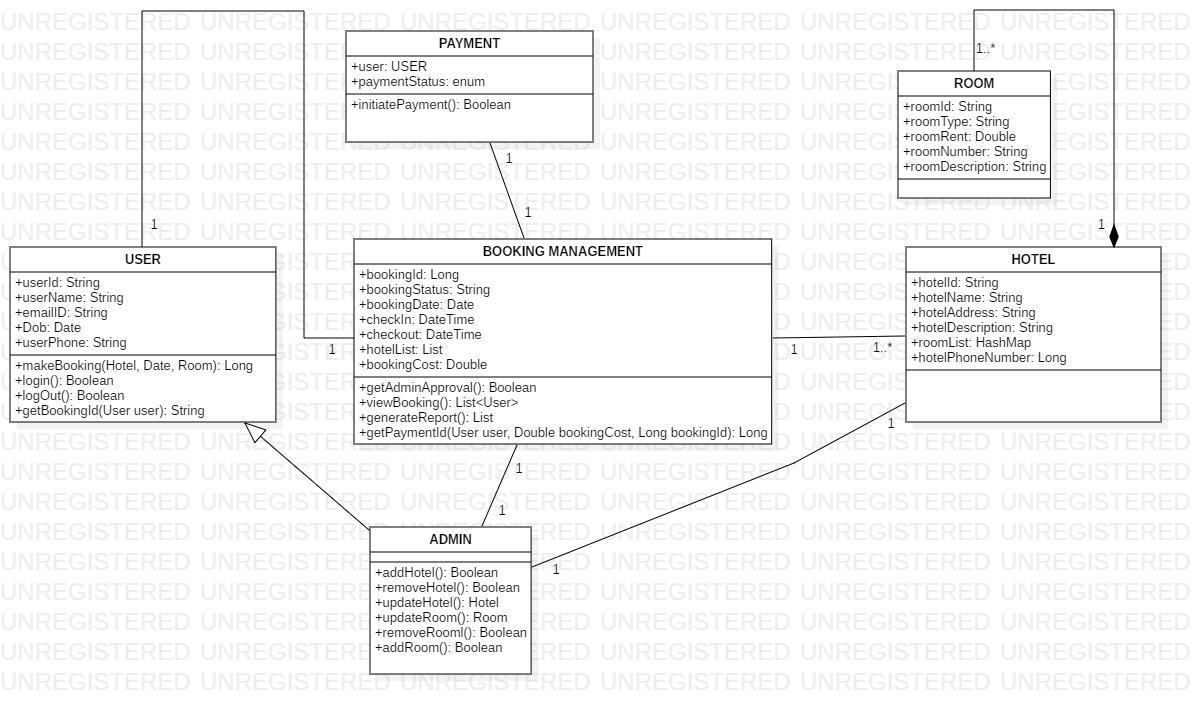
Methods:

initiatePayment(): Boolean

Intiates the payment process of the customer.

**Class Design View:**

The following diagram shows the attributes and methods of each class. Also, it represents the constraints imposed on the system. It represents the static view of the application and can be directly mapped to an Object-Oriented Programming language.



**Flow:**

1. Admin class extends user class and has special methods of its own.
2. Booking Management class has list of hotels. Hotels has a list of rooms(hashmap).
3. User selects room and after admin gives permission for booking
4. After Admin authorization and successful payment, room is booked.

**Validations:**

* The checkout date should be greater than the checkin date.
* The checkin date should be greater thang the booking date.
* The userId and userName should not be null.
* The user must be at least 18 year old.
* The Hotel address must contain a pincode and must have a phone number.
* The booking date must be greater than the current date.