

**Assignment 2: Royal Stay Hotel Management System**

Maryam A. Nasib

Interdisciplinary Studies, Zayed University

ICS220 - 21383 Programming Fundamentals

Dr. Andrew Leonce

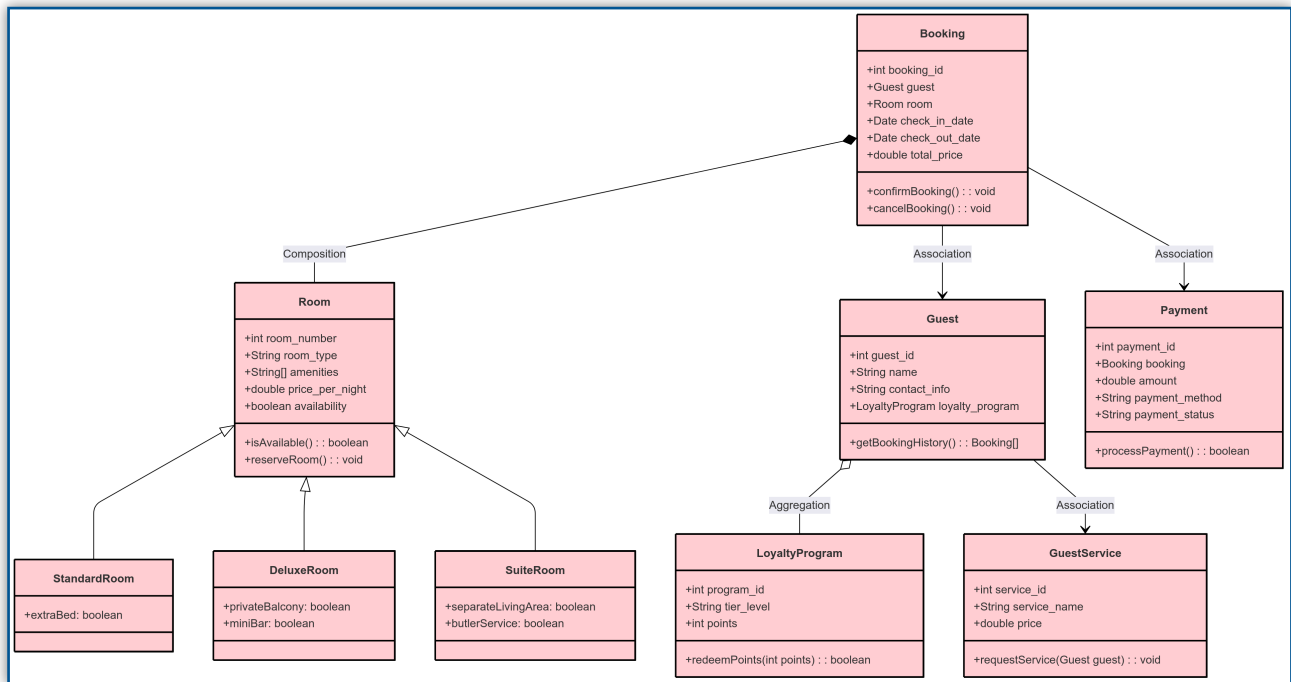
March 28th, 2025

## Royal Stay Hotel Management System:

### A. Design UML Class Diagram

The Royal Stay Hotel Management System is designed to effectively manage hotel bookings, guest management, transactions, and loyalty status. The class diagram below illustrates the key concepts and their relationships.

#### • UML Class Diagram



#### • Classes identifies along with key assumptions:

1. **Room:** Identifies a hotel room and is in charge of storing essential data such as room number, type, services, price per night, and availability status.
2. **Guest:** Identifies a hotel guest and holds all of their personal data and documentation, including contact information and loyalty points.
3. **Booking:** Handles the booking procedure and connects guests with hotel rooms. It covers important booking information like check-in/check-out dates and booking status.
4. **Payment:** In charge of handling payments for booked accommodations. This class keeps records of payment elements like the amount, payment type, and transaction status.
5. **LoyaltyProgram:** Handles every guest's loyalty points, boosting client retention through prizes for return visits. It maintains track of the points acquired by guests and performs the redemption procedure.
6. **GuestService:** In charge of organizing and tracking any extra services ordered by guests throughout their stay. This can consist of services such as room service, spa reservations, additional towels, and any other hotel services.

#### • Class: Room

##### • Attributes:

1. **room\_number:** The distinctive room identifier for each guest.
2. **room\_type:** Indicates the type of room (for example, single, double, or suite).

3. **amenities:** Room facilities available, including Wi-Fi and the minibar, etc.
4. **price\_per\_night:** the price of the room per night, required for invoicing.
5. **availability:** A boolean that indicates whether or not the room can be booked or already booked.

- **Responsibilities:**

1. The Room class checks if the room is available for booking.
2. It includes ways for reserving the room, which indicates it as being unavailable after a guest has reserved it.
3. It serves to guarantee that the system accurately maintains room status (available or booked), avoiding overbooking or cancellations.

- **Relationship Types:**

1. Composition: The link between Booking and Room, a booking is not possible without the existence of a room.
2. Aggregation: The relationship among a guest and a loyalty program, a guest can take part in the loyalty program, yet the program cannot be fundamentally dependent on them.
3. Association: The relationship between a guest and booking, a guest may have several reservations, but a booking is strictly associated with one guest at a given time.
4. Inheritance: The Room class is the base, with StandardRoom, DeluxeRoom, and SuiteRoom inheriting shared attributes and adding specialized features.

- **Modularity:**

1. Each class is responsible for a specific part of the hotel management system, making it easy to extend or modify one part without affecting others.
2. The Payment, Booking, and LoyaltyProgram classes are interconnected, reflecting a modular approach that models real-world dependencies.

- **Class: Booking**

- **Attributes:**

1. **booking\_id:** The distinctive booking identifier for each booking.
2. **guest:** The guest who made the reservation
3. **room:** The room associated with the booking
4. **check\_in\_date:** The date when the guest have checked in.
5. **check\_out\_date:** The date when the guest have checked out.
6. **total\_price:** The total price calculated for the guests stay.

- **Responsibilities:**

1. The link of a guest to their specific room for a certain amount of time span.
2. Makes sure that the room is available before confirm the booking.
3. Manages cancellations and updates room appropriately.

- **Relationship Types:**

1. Composition: The link between Booking and Room, a Booking is not possible without the existence of a Room.

2. Association: The relationship between a `Guest` and `Booking`, a `Guest` may have several reservations, but a `Booking` is strictly associated with one `Guest` at a given time.

- **Modularity:**

1. Each class is responsible for a specific part of the hotel management system, making it easy to extend or modify one part without affecting others.

- **Class: Guest**

- **Attributes:**

1. `guest_id`: The distinctive identifier for each guest.
2. `name`: Guests full name.
3. `contact_info`: Guests information like their phone number and or email address.
4. `loyalty_program`: (optional) Guests membership in rewards program.

- **Responsibilities:**

1. Stores the guests details and contact information.
2. The ability to check the guests past bookings and current reservations.
3. Handles the partaking in the loyalty program.

- **Relationship Types:**

1. Aggregation: The relationship among a guest and a loyalty program, a guest can take part in the loyalty program, yet the program cannot be fundamentally dependent on them.
2. Association: The relationship between a guest and booking, a guest may have several reservations, but a booking is strictly associated with one guest at a given time.

- **Modularity:**

1. Keeping the guests information separate, making it easy to update details without affecting booking or payment processing.

- **Class: Payment**

- **Attributes:**

1. `payment_id`: The distinctive room identifier for each payment.
2. `booking`: The booking associated to the payment.
3. `amount`: The total amount paid.
4. `payment_method`: The way the guests paid for their stay (e.g., credit card, cash)
5. `payment_status`: Indicates if the payment is completed or pending.

- **Responsibilities:**

1. Processes the payments made for conformed bookings.
2. Makes sure that the payments made are linked to the correct bookings.
3. Keeps track of payment status to prevent unpaid bookings.

- **Relationship Types:**

1. Association: A payment is associated with a booking, ensuring each booking has a payment record.

- **Modularity:**

1. Maintains the financial transactions separate from booking management.
2. Makes sure that payment security and easy assessment of financial records.

- **Class: LoyaltyProgram**

- **Attributes:**

1. `program_id`: The distinctive identifier for the loyalty program.
2. `tier_level`: The membership tier (e.g. Silver, Gold, Platinum).
3. `points`: Overall collected reward points.

- **Responsibilities:**

1. Keeps track of the guests engagement in loyalty rewards.
2. Handles point collection and redemption.

- **Relationship Types:**

1. Aggregation: The relationship among a guest and a loyalty program, a guest can take part in the loyalty program, yet the program cannot be fundamentally dependent on them. So the program exist independently.

- **Modularity:**

1. Keeps rewards separate from fundamental hotel functionalities.
2. Easily adaptable to introduce new tiers or rewards.

- **Class: GuestService**

- **Attributes:**

1. `service_id`: The distinctive identifier for the service.
2. `service_name`: The name of the service. (e.g, room cleaning, spa).
3. `price`: The cost of the service.

- **Responsibilities:**

1. Handles additional services that are offered to guests.
2. Allows guests to request and pay for extra services.

- **Relationship Types:**

1. Association: A guest may request several services, but a service may serve multiple guests.

- **Modularity:**

1. Keeps guests services separate from booking and payment.
2. Allows for easy introduction of new services.