**Hotel Booking System**

**Objective:**  
Develop a hotel booking system that enables users to book rooms, check availability, and manage reservations. The system should accommodate different room types and rates while ensuring a robust and flexible architecture through object-oriented design principles.

**Key Concepts and Requirements:**

**1. Class Hierarchy**

**Hotel Class:**

* Represents the entire hotel.
* Contains attributes like hotel name, location, a collection of rooms, and reservations.
* Manages the overall operations of room booking and availability.

**Room Class:**

* Represents individual rooms within the hotel.
* Attributes include room number, type (e.g., single, double, suite), rate per night, and availability status.
* Methods should include checking availability, updating availability, and getting room details.

**Reservation Class:**

* Represents a reservation made by a guest.
* Attributes include reservation ID, guest information, room details, check-in and check-out dates, and total cost.
* Methods should include creating a reservation, modifying a reservation, and canceling a reservation.

**Guest Class:**

* Represents a guest making a reservation.
* Attributes include guest name, contact information, and a collection of reservations made by the guest.
* Methods should include viewing past and current reservations and updating guest details.

**2. Aggregation**

**Hotel-Room Relationship:**

* A hotel contains multiple rooms. This aggregation relationship should allow for the easy addition or removal of rooms from the hotel.
* Reservation-Room Relationship:
* Each reservation is associated with a specific room or set of rooms. Ensure that rooms are correctly linked to reservations to maintain data integrity.

**3. Encapsulation**

**Room Availability:**

Room availability should be encapsulated within the Room class. Implement methods to check and update availability that are only accessible through the Room class, ensuring that the availability status cannot be manipulated directly.

**Reservation Management:**

Encapsulate the logic for handling reservations within the Reservation class. This includes the creation, modification, and cancellation of reservations, ensuring that the business logic is centralized and protected.

**4. Polymorphism**

* Handling Different Room Types:
* Implement polymorphism to handle different types of rooms (e.g., SingleRoom, DoubleRoom, Suite). Each room type should inherit from the Room base class but may override specific methods or attributes (e.g., calculateRate() method) to reflect the unique characteristics and pricing structures of different room types.

**Functional Requirements:**

**Room Management:**

* Ability to add, view, and remove rooms in the hotel.
* Different room types should be created, each with its own pricing structure and amenities.

**Booking Process:**

* Users should be able to search for available rooms based on specific dates.
* The system should allow users to make a reservation by selecting available rooms.
* The booking should reflect the total cost based on the room type and duration of stay.

**Reservation Management:**

* Users should be able to view their reservations, modify the dates, or cancel a reservation.
* The system should handle conflicts (e.g., double booking of the same room) gracefully.

**Guest Management:**

* Ability to create and manage guest profiles.
* Guests should have access to their past and current reservations and should be able to update their personal information.

**Reporting:**

The system should provide basic reporting, such as a list of all reservations, occupancy rates, and revenue generated by room type.