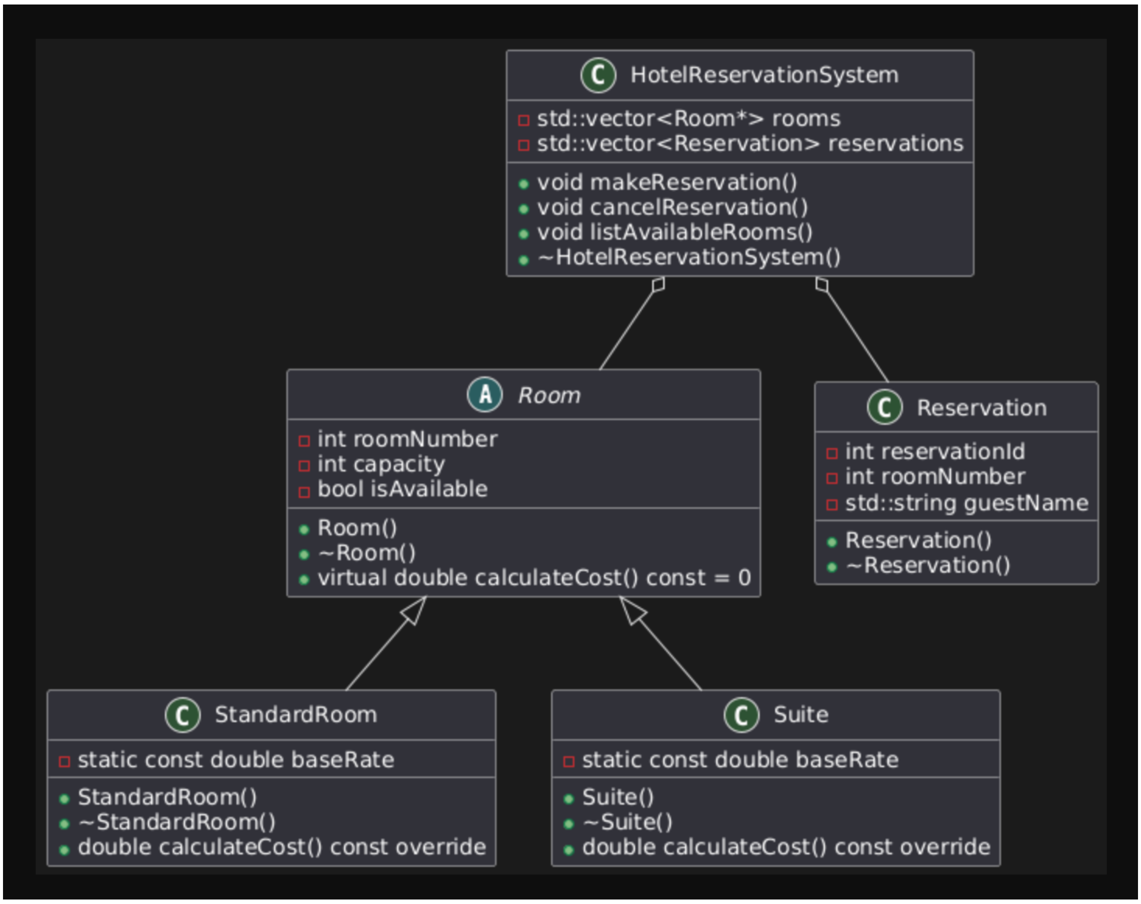
UML Diagram Explanation: Hotel Reservation System



The UML diagram represents the architecture of a **Hotel Reservation System**, including its main classes and their relationships.

1. **Class HotelReservationSystem**

- This class is the central part of the system.
- **Attributes:**
  - std::vector<Room*> rooms: A collection of pointers to Room objects.
  - std::vector<Reservation> reservations: A collection of Reservation objects.
- **Methods:**

- void makeReservation(): Allows users to make a reservation.
- void cancelReservation(): Allows users to cancel an existing reservation.
- void listAvailableRooms(): Lists all rooms that are available for reservation.
- ~HotelReservationSystem(): Destructor to clean up resources.

2. **Class Room (Abstract)**

- This is an abstract base class that defines the general attributes and behaviors of a room.
- **Attributes:**
  - int roomNumber: Unique identifier for the room.
  - int capacity: Maximum number of guests the room can accommodate.
  - bool isAvailable: Indicates whether the room is available for booking.
- **Methods:**
  - Room(): Constructor to initialize the room.
  - ~Room(): Virtual destructor.
  - virtual double calculateCost() const = 0: Pure virtual method that calculates the cost of the room. It is overridden in derived classes.

3. **Class StandardRoom (Inherits from Room)**

- Represents standard rooms in the hotel.
- **Attributes:**
  - static const double baseRate: The base cost of a standard room.
- **Methods:**
  - StandardRoom(): Constructor.
  - ~StandardRoom(): Destructor.
  - double calculateCost() const override: Overrides the base class method to calculate the cost based on baseRate.

4. **Class Suite (Inherits from Room)**

- Represents suites in the hotel.

- **Attributes:**
  - static const double baseRate: The base cost of a suite.
- **Methods:**
  - Suite(): Constructor.
  - ~Suite(): Destructor.
  - double calculateCost() const override: Overrides the base class method to calculate the cost based on baseRate.

5. **Class Reservation**

- Represents a reservation made by a guest.
- **Attributes:**
  - int reservationId: Unique identifier for the reservation.
  - int roomNumber: The room associated with the reservation.
  - std::string guestName: The name of the guest who made the reservation.
- **Methods:**
  - Reservation(): Constructor to create a new reservation.
  - ~Reservation(): Destructor.

Relationships:

- **Inheritance:**
  - StandardRoom and Suite inherit from the abstract class Room.
- **Association:**
  - HotelReservationSystem manages a collection of Room objects (via pointers) and Reservation objects.

This UML diagram provides a clear representation of the system's classes, their attributes, methods, and relationships, offering a structured view of how the system operates.