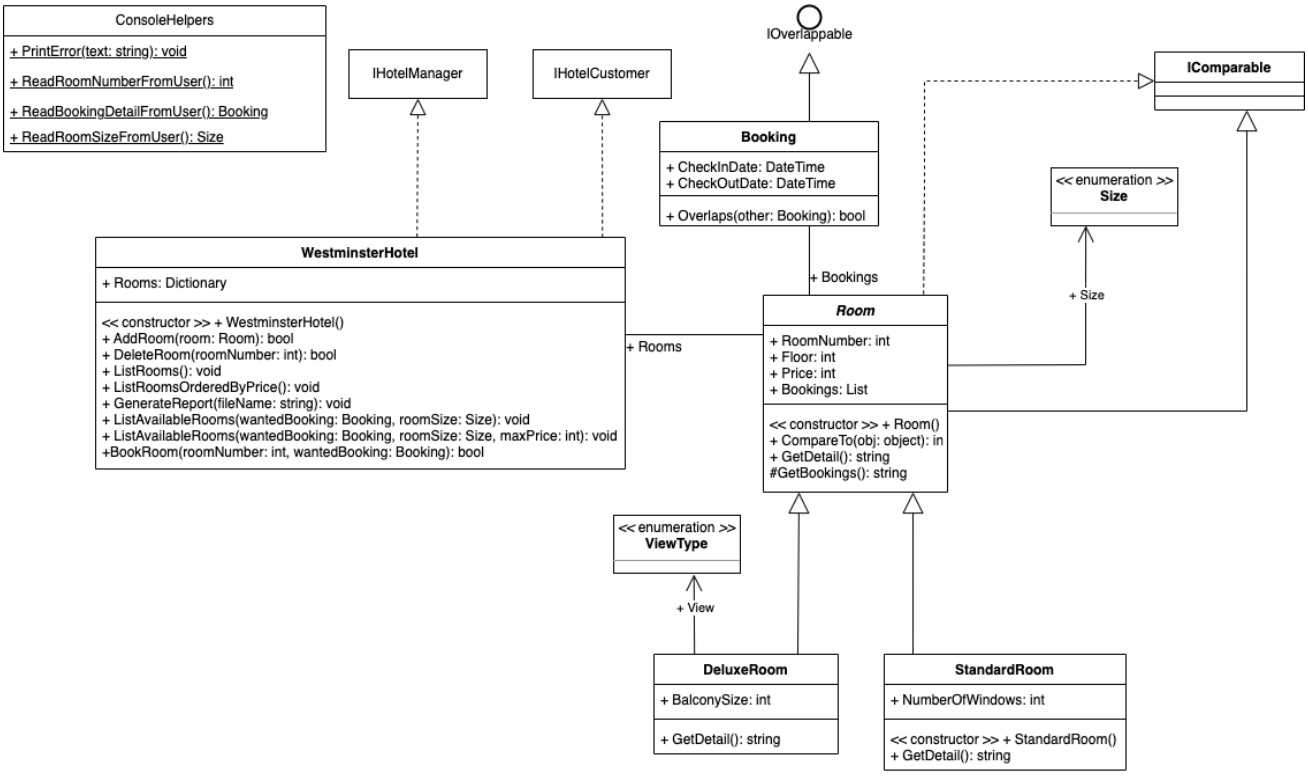
System Design (1):

The following table showcases a brief overview of the classes and methods used to develop the UML class diagram (shown further below along with an explanation of the functionalities using Object-Oriented principles) for the particular software system in question concerning 'Westminster Hotel'.

IHotelManager	IHotelManager interface having methods related to hotel manager.
IHotelCustomer	IHotelCustomer interface having methods related to hotel customers.
WestminsterHotel	WestminsterHotel Class having methods for Customer and Administrator by implementing IHotelCustomer and IHotelManager interface. This class also contains a dictionary of Rooms having booking.
Room	Abstract base class for Room attributes.
StandardRoom	Derived class from Room base class having additional attributes related to Standard Room.
DeluxeRoom	Derived class from Room base class having additional attributes related to Deluxe Room.
Booking	Booking class has a check-in and check-out attribute, also this class implements IOverlappable interface to check whether the booking overlaps with other booking or not.
ViewType	ViewType enum having attributes related to room views.
Size	Size enum having attributes related to room size.



This UML class diagram above illustrates the hotel booking management system's primary classes and their relationships. The Room class is an abstract class that represents a generic room in the hotel. It

contains information on the room number, floor, size, and nightly rate, as well as ways for checking availability and making reservations. The StandardRoom and DeluxeRoom classes inherit from the Room class and add features that are unique to each room type. The Booking class represents a reservation for a hotel room and includes characteristics for the room, the check-in date, and the check-out date. The IOverlappable interface specifies a single method for determining if two reservations overlap, whereas the IHotelManager and IHotelCustomer interfaces specify the activities that an administrator and a customer can perform on the hotel booking system, respectively. The WestminsterHotel class is the main class for the hotel booking system and implements the IHotelManager and IHotelCustomer interfaces. It saves the list of rooms and reservations in the necessary data structures and offers ways for dealing with them.

System Design (2): *982 words*

Room: This is an abstract class representing a standard hotel room. It possesses the following key qualities:

- room number - a positive integer indicating the unique number of the room's floor.
- size - a string indicating the room's dimensions (single, double, or triple)
- price - a float showing the nightly cost of a room

methods are also used to return a boolean indicating if the room is bookable for the specified dates (i.e., no existing bookings overlap with the given date range) and reserves a room for the specified dates and returns a Booking object reflecting the new reservation. This class is abstract because it is not intended to be created directly; rather, it serves as a foundation class for more particular room kinds.

StandardRoom and DeluxeRoom inherit from the Room class and add properties relevant to their respective room types. StandardRoom has a new NumberOfWindows attribute, while DeluxeRoom has BalconySize and view properties. These classes also contain a PrintDetails() function that publishes the room's characteristics, including the properties inherited from the Room class as well as the attributes unique to each type of room.

These classes inherit from the Room class and, when necessary, override or extend its methods to provide their own specialised behaviour. The GetDetails() method of the StandardRoom class may print the value of the NumberOfWindows property, but the GetDetails() method of the DeluxeRoom class may print the values of the BalconySize and View properties.

Booking: This class indicates a reservation for a hotel room. It possesses the following qualities:

- room: A pointer to the reservation's Room object
- checkIn: A DateTime object reflecting the booking's check-in date
- checkOut: A DateTime object providing the booking's check-out date

Moreover, it contains the following method of returning a boolean indicating if the current reservation overlaps with the specified reservation. This class maintains booking-specific information, such as the room being reserved and the dates of the reservation. The Overlaps() function is utilised to determine if the current booking overlaps with another booking, which is important for determining room availability.

IOverlappable: This interface offers a single method, Overlaps(Booking other), for determining if two reservations overlap. This allows other classes to independently implement the overlap checking capability. For instance, the Booking class may implement this function by comparing the check-in and check-out dates of the two reservations, but another class may employ a different strategy.

IHotelManager: This is an interface that specifies the administrative functions available for the hotel booking system. Methods being; AddRoom adds a new room to the booking system. This function returns

a boolean indicating whether the room was successfully added (i.e., the room was not a duplicate). DeleteRoom(int roomNumber): removes the specified room from the reservation system. This function returns a boolean indicating if the room was discovered and erased. ListRooms(): displays the information of any and all rooms in the booking system, including the attributes inherited from the Room class and the attributes particular to each room type. ListRoomsOrderedByPrice() is similar to ListRooms(), except the result is arranged in descending order by room price.

These methods enable an administrator to manage the hotel booking system's rooms, including adding new rooms, removing existing rooms, and viewing a list of all rooms. The ListRooms() function displays the information of all rooms, including any properties inherited from the Room class and any attributes particular to each room type (e.g., NumberOfWindows for StandardRoom, BalconySize and ViewType for DeluxeRoom). Comparable to ListRoomsOrderedByPrice(), ListRoomsOrderedByPrice() sorts the rooms by price in descending order prior to printing.

IHotelCustomer is an interface that specifies the hotel booking system actions that a customer can execute. It searches for available rooms for the specified date range and outputs the corresponding room information, and can reserve the room with the specified room number between the specified dates. This method returns a boolean indicating if the reservation was successful (i.e., whether the requested room was available).

WestminsterHotel is the fundamental class for the hotel reservation system. It includes the logic for these activities and implements the IHotelManager and IHotelCustomer interfaces. It possesses the following qualities of a list or dictionary of Room objects reflecting the hotel room reservations (ListRooms and AvailableRooms).

These methods are used to implement the IHotelManager interface; AddRoom after checking for duplicates, adds the specified room to the rooms list or dictionary. This function returns a boolean indicating whether the room was successfully added. DeleteRoom searches the rooms list or dictionary for the provided room number and, if found, deletes it. This function returns a boolean indicating if the room was discovered and destroyed. ListRooms invokes the functions on each room in the rooms dictionary or ListRoomsOrderedByPrice is similar to ListRooms, except that the rooms are sorted in decreasing price order prior to invoking the PrintDetails function on each room.

These methods are also used to implement the IHotelCustomer interface. It has the following IHotelCustomer implementation methods; via checking if a room is available using the Room class, and using DateTime for check in and DateTime for check out, the system searches the rooms list or dictionary for rooms that are available for the specified date range. Details on the available rooms are printed.
MakeBooking(int roomNumber, DateTime checkIn, DateTime checkOut): searches the rooms list or dictionary for the room with the specified room number and, if found, invokes the function to make the booking of the Room class to reserve the room for the specified dates. This method returns a boolean indicating if the reservation was successful.

This class keeps the list of rooms and reservations in the proper data structures and offers methods for dealing with them. The AddRoom(), DeleteRoom(), ListRooms(), and ListRoomsOrderedByPrice() methods enable an administrator to manage the rooms in a hotel booking system, whilst using the discussed methods to search for rooms and book said room enabling a customer (in this case, a guest), to search for and book hotel rooms.