**Specifications of the Reservation Class**

The reservation class is used to create a reservation record for a guest making reservation for one of the four available rooms at John & Jane bed & breakfast business.

Currently, the Reservation class makes these assumptions:

a)

All arguments input data values to any of the parameters of the Reservation public components are assumed to be valid. The Reservation class does not check for valid argument values, and it assumes they are valid. No testing for valid input data is required. This also means that all the arguments of any test method should contain valid values.

b)  
It is also assumed that the order of all inputted reservation dates is valid. That is reservation end date > reservation start date. The Reservation class does not check for valid order for reservation dates, and it assumes it is valid. No testing for valid order of reservation date is required. This also means that the order of reservation dates of any test method should be valid.

The reservation record includes these attributes:

a) a unique system-generated reservation ID based on Java universally unique identifier (UUID)data type  
b) system-generated date of when the reservation is made (current date) of Java date data type  
c) the guest ID of Java int data type  
d) the reserved room type. There are 3 room types which are “RoomWBath”, “RoomWView”, and “NormalRoom” and they are all of type String. There are 2 rooms of room type “NormalRoom”, all other room types have only 1 room   
e) reservation start date of type String of this pattern: MMM dd, yyyy

f) reservation end date of type String of this pattern: MMM dd, yyyy

The Reservation class provides getters and setters public methods for most of these attributes.

Additionally, the Reservation class provides two public methods as follows:

a) Once an instance of the Reservation class is created, the calculateReversationNumberOfDays()returns the number of days between the end and start dates of a reservation record including the start date and excluding the end date. For example, for a reservation start date of “Jan 02, 2025” and a reservation end date of “Jan 05, 2025”, this method will return 3.

b) Once an instance of the Reservation class is created, the calculateReservationBillAmount()returns the bill amount due from a guest for the reservation as a Java double value. The bill amount is simply the number of reservation days times the daily rate of the reserved room type. The daily rates per room types are as follows: “RoomWBath” -> [$200], “RoomWView” -> [$175], and “NormalRoom” -> [$125]

The complete specifications of all the public components of the Reservation class are:

Table

Description automatically generated