

PROJECT

HOTEL RESERVATION SYSTEM



Students group

Leader : lama , id: 444444

Huda , id : 999999

Layan , id : 66677777

the division of work

lama : super class and all sub (Room , regularRoom , suite , and cover , UML of report)

huda : class Hotel and all its procedures

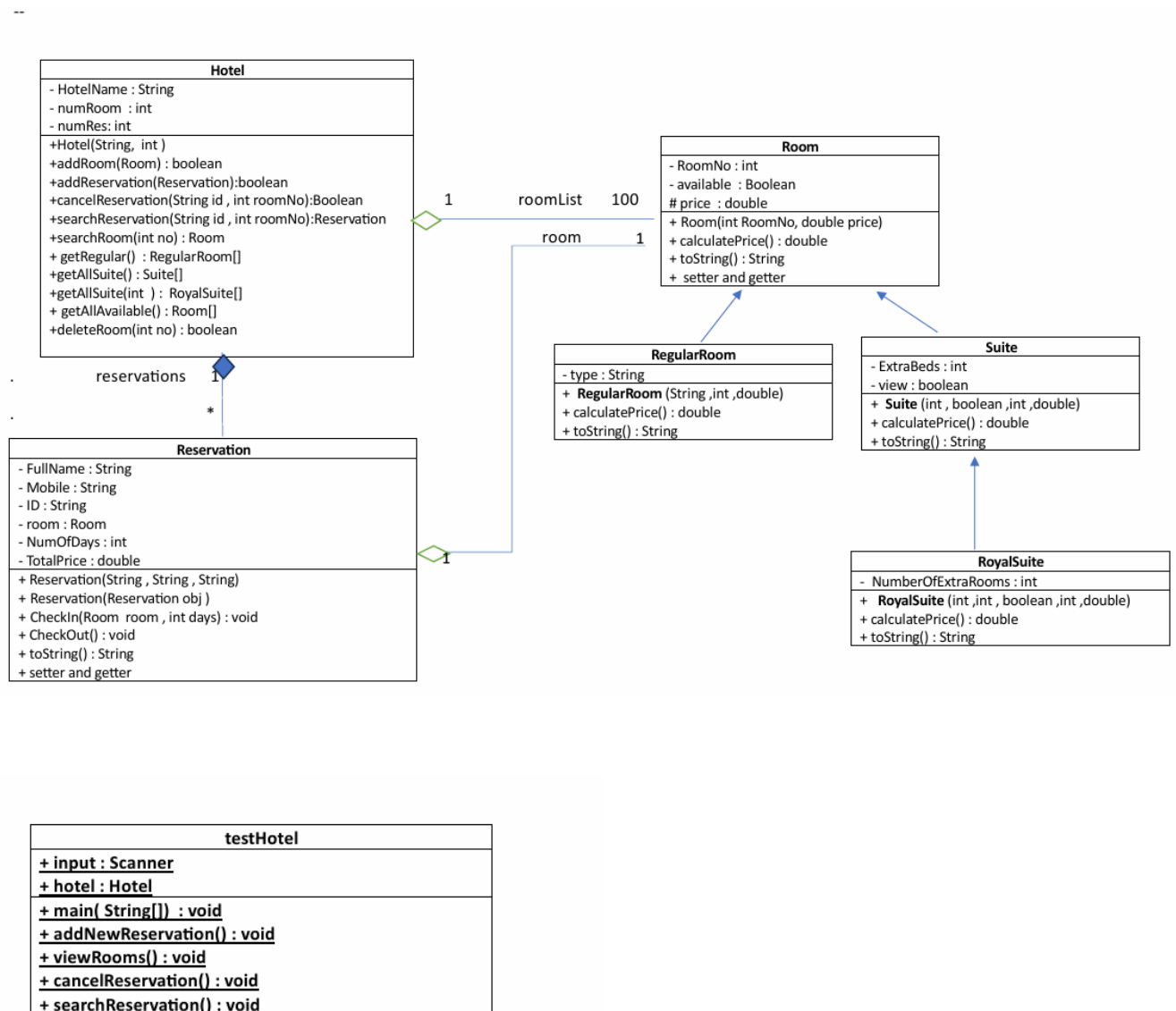
layan : test class and all its methods .

Introduction

The program is a hotel room and suite reservation system, which allows the user to browse rooms according to type or availability , add reservation , cancel and search about reservation by id of reservation and room number .

There are three types of rooms: regular rooms for a single or double, suites and royal suites, and each room has a price according to the existing standards.

UML



Description

Class Room :

1. Attributes :
 - RoomNo : an integer to represent the room number .
 - available : a Boolean to represent if room is available or already reserved by other person
 -
2. Methods :
 - Room(int RoomNo, double price) : Constructor to initialize the roomNo and price with the received parameters.
 - calculatePrice() : an abstract method to calculate the price for every room
 - toString() : to return all information of room

class RegularRoom :

that extends from room class

1. Attributes :
 - type : an string to represent the room type (single or double) .
2. Method :
 - RegularRoom(String type, int RoomNo, double price): Constructor to initialize all attributes
 - public double calculatePrice(): method that return double represent a price of room , price is calculate as follow :
 If type equal "double" than price = original price + 150 SR
 Otherwise price will remain as it is .

....

.....

....

Code :

```

public abstract class Room {
    private int RoomNo;
    private boolean available ;
    protected double price ;

    public Room(int RoomNo, double price) {
        this.RoomNo = RoomNo;
        this.price = price;
        available = true;
    }

    public abstract double calculatePrice();

    public int getRoomNo() {
        return RoomNo;
    }

    public void setRoomNo(int RoomNo) {
        this.RoomNo = RoomNo;
    }

    public boolean isAvailable() {
        return available;
    }

    public void setAvailable(boolean available) {
        this.available = available;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double price) {
        this.price = price;
    }

    public String toString() {
        return "Room No. : " + RoomNo + ", available=" + available ;
    }
}

=====

public class RegularRoom extends Room {
    private String type ; // single or double

```

```

public RegularRoom(String type, int RoomNo, double price) {
    super(RoomNo, price);
    this.type = type;
}

    public String toString() {
        return super.toString() + ", type of room : " + type ;
    }

    public double calculatePrice(){
        double total = 0 ;

        if(type.equalsIgnoreCase("double"))
            total = price + 150;
        else
            total = price;    // single

        return total;
    }
}

=====

public class Suite extends Room {
    private int ExtraBeds ;
    private boolean view ;

    public Suite(int ExtraBeds, boolean view, int RoomNo, double price) {
        super(RoomNo, price);
        this.ExtraBeds = ExtraBeds;
        this.view = view;
    }

    public String toString() {
        return super.toString() + " , No. of extra beds : " + ExtraBeds +
        ( view ? " , there is a nice view " : "");
    }

    public double calculatePrice(){
        double total = price ;
        total = total + ( 150 * ExtraBeds) ;

        if( view == true )
            total = total + 200 ;
        return total;
    }
}

=====

public class RoyalSuite extends Suite {
    private int NumberOfExtraRooms ;

    public RoyalSuite(int NumberOfExtraRooms, int ExtraBeds, boolean view, int

```

```

RoomNo, double price) {
    super(ExtraBeds, view, RoomNo, price);
    this.NumberOfExtraRooms = NumberOfExtraRooms;
}

public String toString() {
    return super.toString() + "\n number of Extra Room in royal suite : " +
    NumberOfExtraRooms ;
}

public double calculatePrice(){
    double total = super.calculatePrice();
    total = total + 350 * NumberOfExtraRooms ;

    return total;
}

public int getNumberOfExtraRooms() {
    return NumberOfExtraRooms;
}
}

=====

public class Reservation {
    private String FullName ;
    private String Mobile ;
    private String ID ;
    private Room room ;
    private int NumOfDays ;
    private double TotalPrice ;

    public Reservation(String FullName, String Mobile, String ID) {
        this.FullName = FullName;
        this.Mobile = Mobile;
        this.ID = ID;
    }

    public Reservation(Reservation obj ) {
        this.FullName = obj.FullName;
        this.Mobile = obj.Mobile;
        this.ID = obj.ID;
        this.NumOfDays = obj.NumOfDays ;
        this.room = obj.room ;
        this.TotalPrice = obj.TotalPrice ;
    }

    public void CheckIn(Room room , int days){
        NumOfDays = days ;
        this.room = room ;
        room.setAvailable(false);
        TotalPrice = NumOfDays * this.room.calculatePrice() ;
    }
}

```

```

public void CheckOut(){
    if( room.isAvailable()== true)
        System.out.println("this reservation is already checkOut befor.");
    else
        {
room.setAvailable(true);
        System.out.println("check out is done.");
        }
}

public String toString() {
return "FullName : " + FullName + ", Mobile=" + Mobile +
    "\n ID=" + ID + "\n room=" + room +
    "\n NumOfDays=" + NumOfDays +
    "\n TotalPrice : " + TotalPrice ;
}

public String getFullName() {
return FullName;
}

public void setFullName(String FullName) {
this.FullName = FullName;
}

public String getMobile() {
return Mobile;
}

public void setMobile(String Mobile) {
this.Mobile = Mobile;
}

public String getID() {
return ID;
}

public void setID(String ID) {
this.ID = ID;
}

    public Room getRoom() {
        return room;
    }

    public void setRoom(Room room) {
        this.room = room;
    }

}

=====

package testhotel;

```

```

public class Hotel {
private String HotelName ;
Room[]  roomList;
Reservation[]  reservations ;
int numRoom , numRes ;

public Hotel(String name ,  int maxReservation){
HotelName = name;
roomList = new Room[100] ;
numRoom = 0 ;
reservations  = new Reservation[maxReservation] ;
numRes = 0 ;
}

public boolean addRoom(Room room ){
if( numRoom < roomList.length ){  // aggregation
roomList[numRoom] = room ;
numRoom++;
return true;
}
return false;
}

public boolean deleteRoom(int no ){
for(int i = 0 ; i < numRoom ; i++)
if(roomList[i].getRoomNo() == no ){

if(roomList[i].isAvailable() == true ){
for( int j = i ; j < numRoom-1 ; j++)
roomList[j] = roomList[j+1] ;

numRoom--;
roomList[numRoom] = null;
return true;
}
else
System.out.println("this room is reserved already now by other person");
return false;
}

return false;
}

public boolean addReservation(Reservation res){
if( numRes < reservations.length ) {
reservations[numRes] = new Reservation(res) ;  //composition
numRes++;
return true;
}
}

```



```

else
return false;
}

public boolean cancelReservation(String id , int roomNo){
    for( int i = 0 ; i < numRes ; i++)    {
        if( reservations[i].getID().equals(id)  &&
            reservations[i].getRoom().getRoomNo() == roomNo )
        {
            reservations[i].CheckOut();
            reservations[i] = reservations[numRes-1];
            numRes--;
            reservations[numRes] = null;
            return true;
        }
    }

    return false;
}

public Reservation searchReservation(String id , int roomNo){
    for( int i = 0 ; i < numRes ; i++)
        if( reservations[i].getID().equals(id)  &&
            reservations[i].getRoom().getRoomNo() == roomNo )
            return reservations[i];

    return null;
}

public Room searchRoom(int no ){
    for(int i = 0 ;i < numRoom ; i++)
        if(roomList[i].getRoomNo() == no )
            return roomList[i] ;

    return null ;
}

public RegularRoom[] getRegular(){
    RegularRoom[] list = new RegularRoom[numRoom] ;
    int j = 0 ;
    for( int i = 0 ;i < numRoom ; i++)
        if( roomList[i] instanceof RegularRoom )
            list[j++] = (RegularRoom) roomList[i] ;

    return list ;
}

public Suite[] getAllSuite(){
    Suite[] list = new Suite[numRoom] ;
    int j = 0 ;
    for( int i = 0 ;i < numRoom ; i++)

```

```

if( roomList[i] instanceof Suite )
    list[j++] = (Suite) roomList[i] ;

return list ;
}

public RoyalSuite[] getAllSuite(int numberOfExtraRooms) // overload
{
    RoyalSuite[] list = new RoyalSuite[numRoom] ;
    int j = 0 ;
    for( int i = 0 ; i < numRoom ; i++)
        if( roomList[i] instanceof RoyalSuite &&
            ((RoyalSuite) roomList[i]).getNumberOfExtraRooms() >=
            numberOfExtraRooms )
            list[j++] = (RoyalSuite) roomList[i] ;

    return list ;
}

public Room[] getAllAvailable(){
    Room[] list = new Room[numRoom] ;
    int j = 0 ;
    for( int i = 0 ; i < numRoom ; i++)
        if( roomList[i].isAvailable() )
            list[j++] = roomList[i] ;

    return list ;
}

}

=====

package testhotel;

import java.util.Scanner;

public class TestHotel {
    static Scanner input = new Scanner(System.in) ;
    static Hotel hotel = new Hotel("Sama Hotel" , 1000) ;

    public static void main(String[] args) {

        RegularRoom reg1 = new RegularRoom("Single" , 101 , 350 ) ;
        RegularRoom reg2 = new RegularRoom("Single" , 102 , 300 ) ;
        RegularRoom reg3 = new RegularRoom("dobule" , 103 , 450 ) ;
        RegularRoom reg4 = new RegularRoom("double" , 104 , 450 ) ;

        //int ExtraBeds, boolean view, int RoomNo, double price) {
        Suite suit1 = new Suite( 2 , true , 205 , 900 ) ;
        Suite suit2 = new Suite( 0 , false , 206 , 800 ) ;
        Suite suit3 = new Suite( 1 , true , 207 , 1000 ) ;
    }
}

```

```
//(int NumberOfExtraRooms, int ExtraBeds, boolean view, int RoomNo, double
price) {
    RoyalSuite royall1 = new RoyalSuite(2 , 2 , true , 308 , 3500 );
    RoyalSuite royall2 = new RoyalSuite(3 , 0 , true , 309 , 5500 );

    hotel.addRoom(reg1) ;    hotel.addRoom(reg2) ;
    hotel.addRoom(reg3) ;    hotel.addRoom(reg4) ;

    hotel.addRoom(suit1) ;    hotel.addRoom(suit2) ;
    hotel.addRoom(suit3) ;

    hotel.addRoom(royall1) ;
    hotel.addRoom(royall2) ;

    //=====
    int choice ;
    do{
        System.out.println("***** Menue ***** ");
        System.out.println("1- add new reservation");
        System.out.println("2- view rooms");
        System.out.println("3- cancel reservation or checkOut reservation  ");
        System.out.println("4- search about reservation");
        System.out.println("5- delet room");
        System.out.println("6- Exit");
        choice = input.nextInt();
        switch( choice ){
            case 1 :
                addNewReservation();
                break;

            case 2 :
                viewRooms();
                break;

            case 3 :
                cancelReservation();
                break;

            case 4 :
                searchReservation();
                break;
            case 5 :
                System.out.println("Enter room No");
                int no = input.nextInt() ;
                if( hotel.deleteRoom(no)){
                    System.out.println("delete is done");
                }
                else
                    System.out.println("Can't Delete");
                break;
            case 6 :
```

```

System.out.println("**** good by ****");
break;
default :
System.out.println("invalid input ");
} // end switch
}while(choice != 6 ) ;

} // end main

//=====
public static void addNewReservation() {

System.out.println("Enter room No: ");
int no = input.nextInt();
Room roomObj = hotel.searchRoom(no) ;
if( roomObj == null || roomObj.isAvailable() == false){
System.out.println("this room is not available, try again");
return ;
}

System.out.println("Enter customer full name : ");
input.nextLine();
String name = input.nextLine();
System.out.println("Enter Mobile : ");
String mob = input.next();
System.out.println("Enter customer Id : ");
String id = input.next();
Reservation res = new Reservation(name , mob , id ) ;
System.out.println("Enter number of days that you want to make reservation
: ");
int days = input.nextInt();
res.CheckIn(roomObj, days);
hotel.addReservation(res) ;
System.out.println("reservation is added sucssfully , be happy with our
hotel");
System.out.println("*****");
System.out.println(res.toString() );
System.out.println("*****");

}

public static void viewRooms() {
System.out.println("enter your choice : ");
System.out.println("1- view Regular rooms");
System.out.println("2- view suite ");
System.out.println("3- view Royal Suite with Specific number of rooms ");
System.out.println("4- view all available rooms ");
System.out.println("5- return to main menu ");
int ch = input.nextInt();

```

```

switch(ch){
case 1 :
RegularRoom[] list = hotel.getRegular();
System.out.println("*****");
for( int i = 0 ;i < list.length ; i++ )
if( list[i] != null ){
System.out.println(list[i].toString());
System.out.println("*****");
}
break;
case 2 :
Suite[] list2 = hotel.getAllSuite();
System.out.println("*****");
for( int i = 0 ;i < list2.length ; i++ )
if( list2[i] != null && ! (list2[i] instanceof RoyalSuite ) ){
System.out.println(list2[i].toString());
System.out.println("*****");
}
break;
case 3 :
System.out.print("Enter number of extra rooms that you want : ");
int num = input.nextInt();
Suite[] list3 = hotel.getAllSuite(num);
System.out.println("*****");
for( int i = 0 ;i < list3.length ; i++ )
if( list3[i] != null ){
System.out.println(list3[i].toString());
System.out.println("*****");
}
break;
case 4 :
Room[] list4 = hotel.getAllAvailable();
System.out.println("*****");
for( int i = 0 ;i < list4.length ; i++ )
if( list4[i] != null ){
System.out.println(list4[i].getClass().getSimpleName());
System.out.println(list4[i].toString());
System.out.println("*****");
}
break;
case 5 :
System.out.println("*****");
break;
default :
System.out.println(" invalid input" ) ;
} // end switch

}

public static void cancelReservation(){
System.out.println("\n -----enter your choice : ");
System.out.println("1- Cancel reservation : ");

```

```

System.out.println("2- make checkOut for reservation ; ");
System.out.println("3- return to main menue ");
int ch = input.nextInt();
switch(ch) {
    case 1 :
        System.out.println("Enter customer id : ");
        String id = input.next();
        System.out.println("Enetr room No. ");
        int no = input.nextInt();

        if( hotel.cancelReservation(id, no))
            System.out.println("cancel done.");
        else
            System.out.println("sorry , can't cancel.");
        break ;

    case 2 :
        System.out.println("Enter customer id : ");
        id = input.next();
        System.out.println("Enetr room No. ");
        no = input.nextInt();
        Reservation res = hotel.searchReservation(id, no) ;
        if(res == null )
            System.out.println("can't found this reservation");
        else
            res.CheckOut();
        break;

    case 3 :
        System.out.println("-----");
        break;

    default :
        System.out.println("--- invalid input --");

}

}

public static void searchReservation() {
    System.out.println("Enter customer id : ");
    String id = input.next();
    System.out.println("Enetr room No. ");
    int no = input.nextInt();
    Reservation res = hotel.searchReservation(id, no) ;
    if(res == null )
        System.out.println("can't found this reservation");
    else
        System.out.println(res.toString());
    System.out.println("-----");
}

}

```

Screen shot of sample Run

```

run:
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
2
enter your choice :
1- view Regular rooms
2- view suite
3- view Royal Suite with Specific number of rooms
4- view all available rooms
5- return to main menue
3
Enter number of extra rooms that you want : 1
*****
Room No. : 308, available=true , No. of extra beds : 2 , there is a nice view
number of Extra Room in royal suite : 2
*****
Room No. : 309, available=true , No. of extra beds : 0 , there is a nice view
number of Extra Room in royal suite : 3
*****
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
2
enter your choice :

```

```

enter your choice :
1- view Regular rooms
2- view suite
3- view Royal Suite with Specific number of rooms
4- view all available rooms
5- return to main menue
1
*****
Room No. : 101, available=true, type of room : Single
*****
Room No. : 102, available=true, type of room : Single
*****
Room No. : 103, available=true, type of room : dobule
*****
Room No. : 104, available=true, type of room : double
*****
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
1
Enter room No:
102
Enter customer full name :
Omar Mohammad
Enter Mobile :
055566783
Enter customer Id :
111
Enter number of days that you want to make reservation :

```



```

Enter number of days that you want to make reservation :
2
reservation is added succssfully , be happy with our hotel
*****
FullName : Omar Mohammad, Mobile=055566783
ID=111
room=Room No. : 102, available=false, type of room : Single
NumOfDays=2
TotalPrice : 600.0
*****
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
4
Enter customer id :
111
Enetr room No.
101
can't found this reservation
-----
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
4
Enter customer id :
111

Enter customer id :
111
Enetr room No.
102
FullName : Omar Mohammad, Mobile=055566783
ID=111
room=Room No. : 102, available=false, type of room : Single
NumOfDays=2
TotalPrice : 600.0
-----
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
1
Enter room No:
102
this room is not available, try again
**** Menue ****
1- add new reservation
2- view rooms
3- cancel reservation or checkOut reservation
4- search about reservation
5- Exit
5
**** good by ****
BUILD SUCCESSFUL (total time: 1 minute 51 seconds)

```