Balasubramanyam - 12117

Hotel Room Reservation System

WHAT:	**HOW:**
1) Question: Do you want the guest to register in your webpage? Ans: yes, guests want to register the page. 2) Question: Do you want the guest to check in and check out the room ? Ans: yes, guest want to do check in and check out process. 3) Question: Do you want the guest to reserve the room? Ans: Yes, guests want to reserve the room. 4) Question: Do to you want the guest to add/update/delete/view their details? Ans: Yes, guests want to do this operation. 5) Question: How do guests pay rent for the rooms? Ans: yes, guest want to pay the rent for rooms, by payment method. 6) Question: Guests able to give reviews ? Ans: Yes, guests make reviews.	1)Question: Method1: Through the mobile number. Method2: Through the email-id. Method3: Through the both number or Email-id. 2)Question: Method1: By note the time and date of arrival of guest Method2: By note the time and date of relieving the guest from room Method3: By note the both time and date of Arrival and Relieving the room details 3)Question: Method1: By Showing the picture of the room and give details to reserve. Method2: By giving the dimension about the room to the guest, to do reserve. Method3: By giving the room name like single bed and double bed room to guest, to do reserve. 4)Question: Method1: Guest able to add/update details details. Method2: Guest able to delete/ view details Method3: Guest able to add/update/delete/view their details. 5)Question: Method1: Through the bank transfer Method2: Through the upi, cards method Method3: Through the both bank and card Methods. 6)Question: Method1: Review through the Text message. Method2: Review through the Audio. Method3: Through both Text and Audio.

WHY:	**WHY NOT:**
1)Question: Method3: Through the both number or email-id. (Through the both number and email-id user can easily register and login into the page) 2)Question: Method3: By note the both time and date of Arrival and Relieving the room details. (By note the both arrival and relieving only we can easily check in and check out the guest) 3)Question: Method1: By Showing the picture of the room and give details to reserve (By showing the picture of room only guest easily understand about the room and cleanliness, facility.) 4)Question: Method3: Guest able to add/update/delete/view their details. (If giving an access to guest can to do all operation and it is user friendly for the guest) 5)Question: Method3: Through the both bank and card Methods. (By many payment methods to avoid the transaction failure) 6)Question: Method3: Through the both Text and Audio (By giving the review in both text and audio other user can easily understand about the hotel facilities)	1)Question: Some of them are not having the email-id, if provide the mobile number easily to register and login. 2)Question: By note the both Time and Date of Arrival and Relieving is important for an Check in and check out for guests. 3)Question: By giving the dimension and room name guest cannot the easily understand about the room condition, by through the picture it is Easy. 4)Question: If restrict the some operation to the user it make difficult to the user to make some operation on the page. 5)Question: By making the many payment method it is very easier for the guest to pay and avoid the Failures. 6)Question: By giving the review in both text and audio other user can easily understand about the hotel facilities

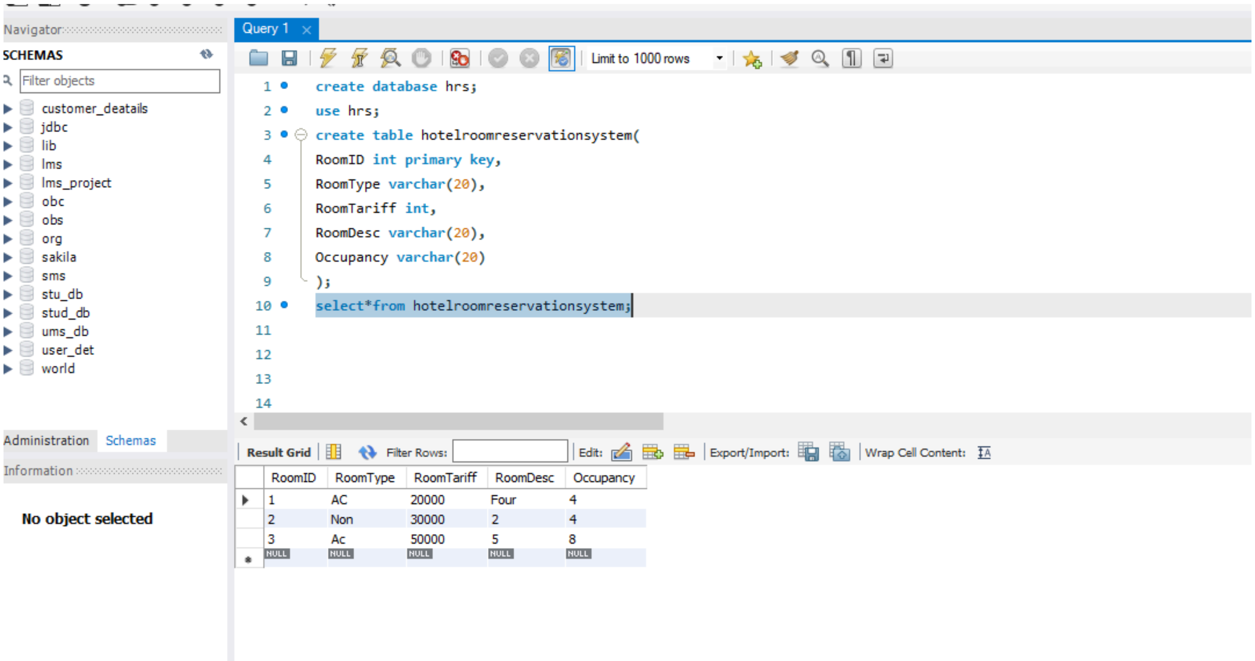
Hotel room reservation system

Property	Type
RoomID	Int
RoomType	String
RoomTariff	int
RoomDesc	String
Occupancy	String

Note:

1. Define a console based application for the given scenario using Java.
2. Use MySQL as your database.
3. Perform all the CRUD operations using Java DAO pattern.
4. Make the application as user interactive.
5. Maintain the Code Quality and the Coding Standard .

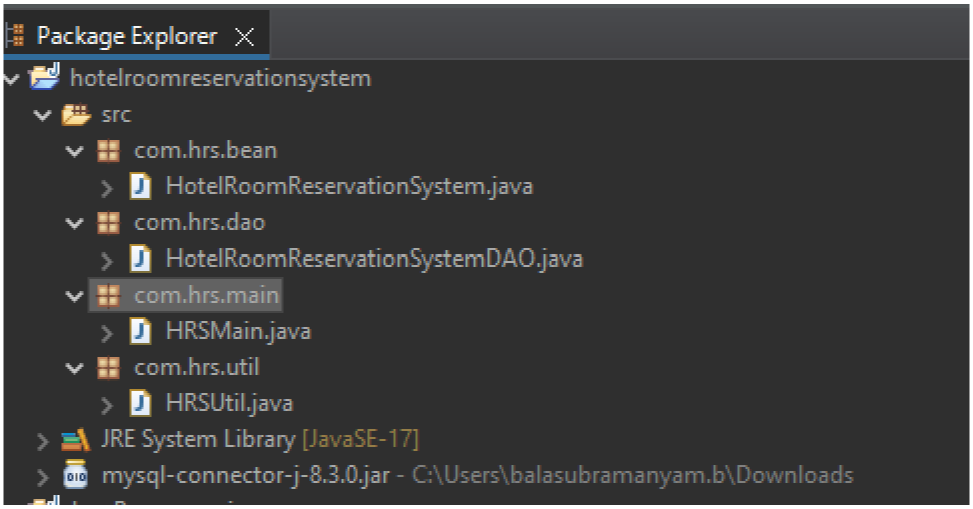
DataBase:



```sql
create database hrs;
use hrs;
create table hotelroomreservationsystem(
RoomID int primary key,
RoomType varchar(20),
RoomTariff int,
RoomDesc varchar(20),
Occupancy varchar(20)
);
select*from hotelroomreservationsystem;
```

RoomID	RoomType	RoomTariff	RoomDesc	Occupancy
1	AC	20000	Four	4
2	Non	30000	2	4
3	Ac	50000	5	8
NULL	NULL	NULL	NULL	NULL

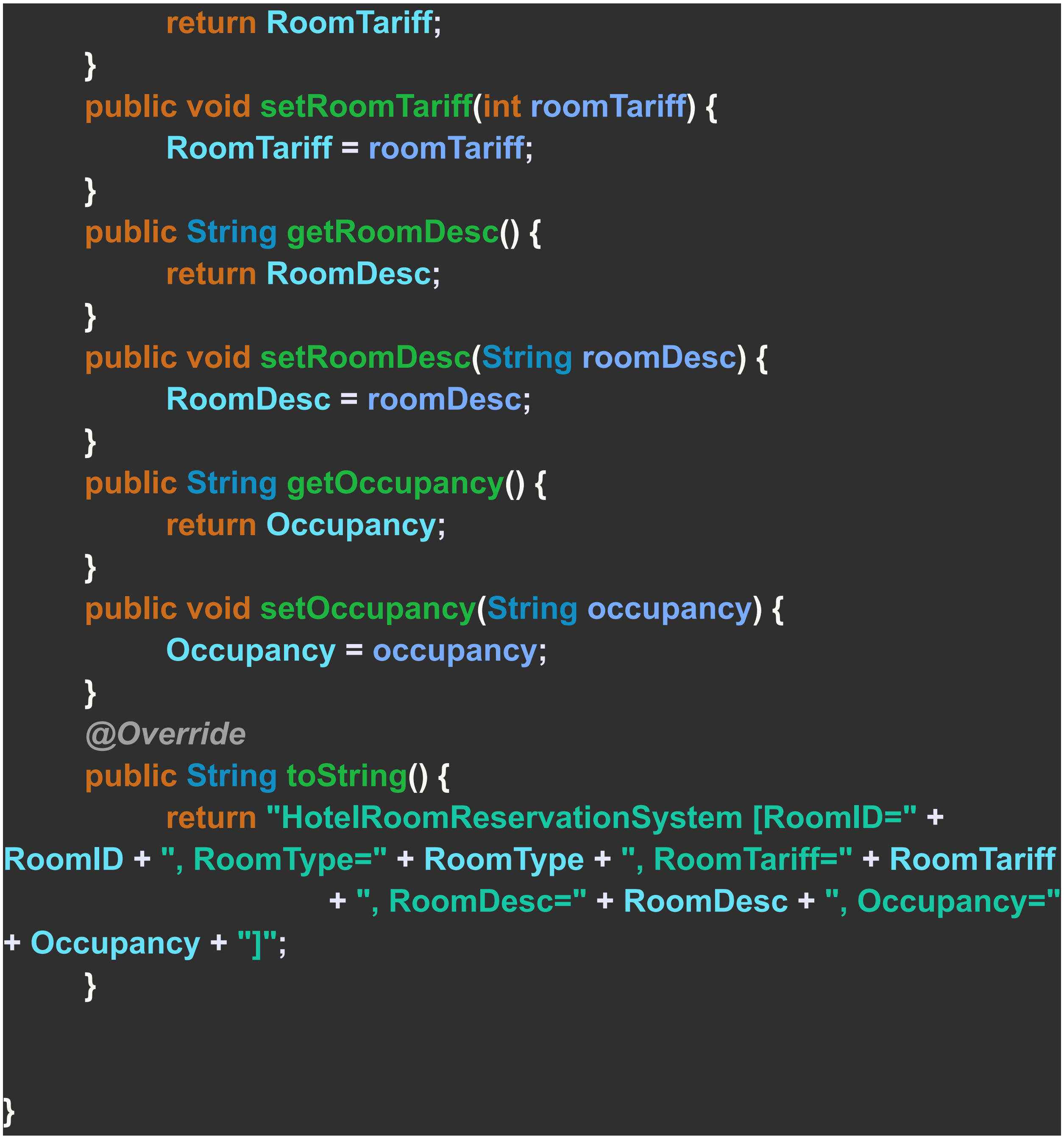
Applications:



```
hotelroomreservationsystem
  src
    com.hrs.bean
      HotelRoomReservationSystem.java
    com.hrs.dao
      HotelRoomReservationSystemDAO.java
    com.hrs.main
      HRSMain.java
    com.hrs.util
      HRSUtil.java
  JRE System Library [JavaSE-17]
  mysql-connector-j-8.3.0.jar - C:\Users\balasubramanyam.b\Downloads
```

Com.hrs.bean:

```java
package com.hrs.bean;
public class HotelRoomReservationSystem {
        private int RoomID;
        private String RoomType;
        private int RoomTariff;
        private String RoomDesc;
        private String Occupancy;
        public HotelRoomReservationSystem() {
                super();
        }
        public HotelRoomReservationSystem(int roomID, String roomType, int roomTariff, String roomDesc, String occupancy) {
                super();
                RoomID = roomID;
                RoomType = roomType;
                RoomTariff = roomTariff;
                RoomDesc = roomDesc;
                Occupancy = occupancy;
        }
        public int getRoomID() {
                return RoomID;
        }
        public void setRoomID(int roomID) {
                RoomID = roomID;
        }
        public String getRoomType() {
                return RoomType;
        }
        public void setRoomType(String roomType) {
                RoomType = roomType;
        }
        public int getRoomTariff() {
```



```
		return RoomTariff;
	}
	public void setRoomTariff(int roomTariff) {
		RoomTariff = roomTariff;
	}
	public String getRoomDesc() {
		return RoomDesc;
	}
	public void setRoomDesc(String roomDesc) {
		RoomDesc = roomDesc;
	}
	public String getOccupancy() {
		return Occupancy;
	}
	public void setOccupancy(String occupancy) {
		Occupancy = occupancy;
	}
	@Override
	public String toString() {
		return "HotelRoomReservationSystem [RoomID=" +
RoomID + ", RoomType=" + RoomType + ", RoomTariff=" + RoomTariff
				+ ", RoomDesc=" + RoomDesc + ", Occupancy="
+ Occupancy + "]";
	}


}
```

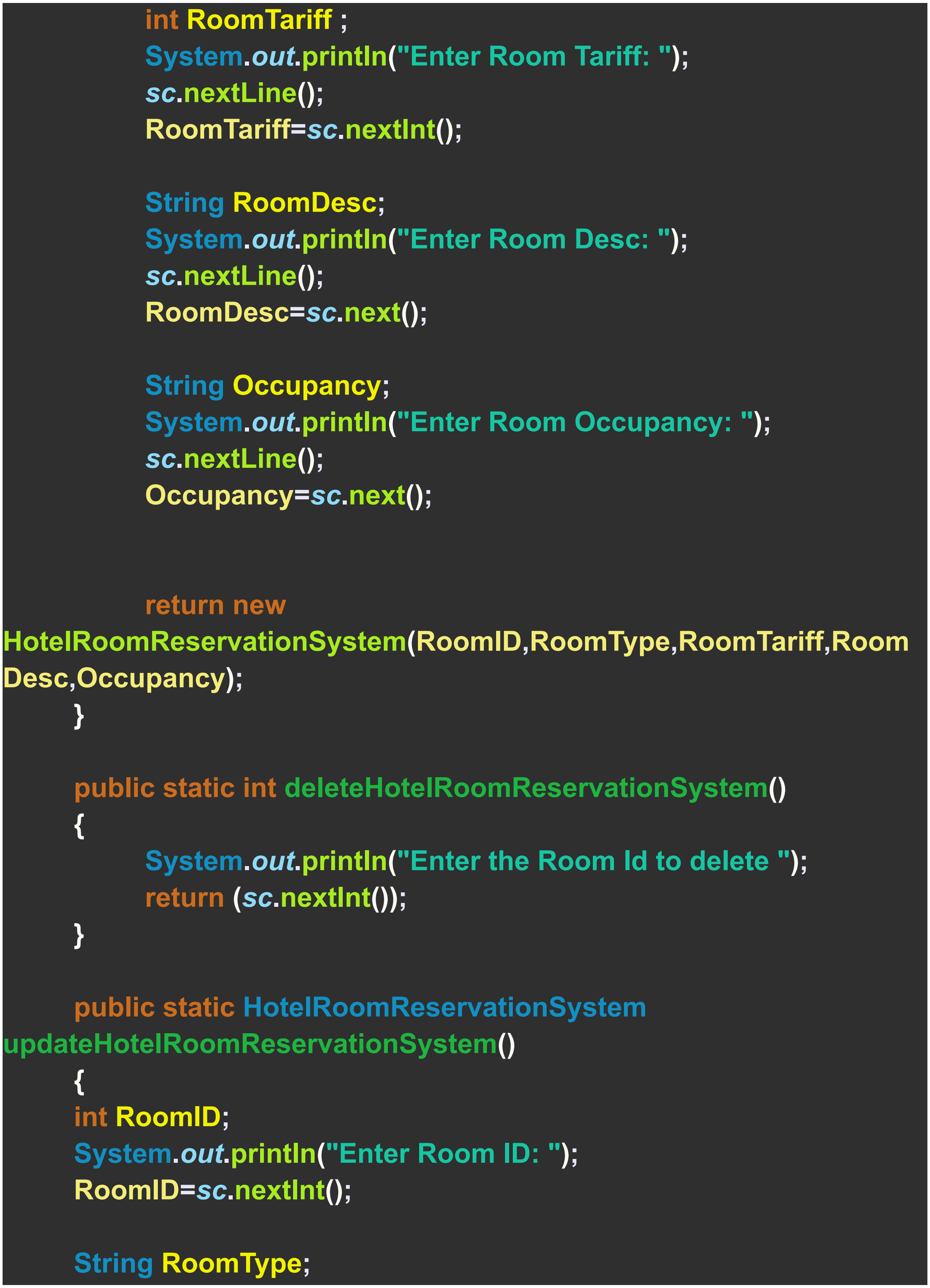
Com.hrs.dao :

```
package com.hrs.dao;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import com.hrs.bean.HotelRoomReservationSystem;
import com.hrs.util.HRSUtil;
public class HotelRoomReservationSystemDAO {
                public int
insertHotelRoomReservationSystem(HotelRoomReservationSystem
hrs)
                {
                        int n=0;
                        try {
                                Connection con=HRSUtil.getDBConnection();
                                String sql="insert into
HotelRoomReservationSystem values(?,?,?,?,?)";
                                PreparedStatement
ps=con.prepareStatement(sql);
                                ps.setInt(1, hrs.getRoomID());
                                ps.setString(2,hrs.getRoomType() );
                                ps.setInt(3, hrs.getRoomTariff());
                                ps.setString(4,hrs.getRoomDesc() );
                                ps.setString(5,hrs.getOccupancy() );

                                n=ps.executeUpdate();

                        }
                        catch(Exception e)
                {
                        System.out.println(e);
                }
          return n;
        }


                public int deleteHotelRoomReservationSystem(int RoomID)
                {
                        int n=0;
                        try {
                                Connection con=HRSUtil.getDBConnection();
                                String sql="delete from 
HotelRoomReservationSystem where RoomID=?";
                                PreparedStatement 
ps=con.prepareStatement(sql);
                                ps.setInt(1, RoomID);
                                n=ps.executeUpdate();

                        }
                        catch(Exception e)
                        {
                                System.out.println(e);
                        }
          return n;
        }

         public int 
updateHotelRoomReservationSystem(HotelRoomReservationSystem 
hrs)
                {
                        int n=0;
                        try {
                                Connection con=HRSUtil.getDBConnection();
                String sql="update hotelroomreservationsystem
set RoomType=?,RoomTariff=?,RoomDesc=?,Occupancy=? where
RoomID=?";
                PreparedStatement
ps=con.prepareStatement(sql);

                ps.setString(1,hrs.getRoomType() );
                ps.setInt(2, hrs.getRoomTariff());
                ps.setString(3,hrs.getRoomDesc() );
                ps.setString(4,hrs.getOccupancy() );
                ps.setInt(5, hrs.getRoomID());
                n=ps.executeUpdate();

            }
            catch(Exception e)
            {
                System.out.println(e);
            }
     return n;
    }

     public HotelRoomReservationSystem
findHotelRoomReservationSystem(int RoomID) {
        HotelRoomReservationSystem
hotelRoomReservationSystem = null;
            try {
                Connection con=HRSUtil.getDBConnection();
                String sql="select * from
HotelRoomReservationSystem where RoomID=?";
                PreparedStatement
ps=con.prepareStatement(sql);
                ps.setInt(1, RoomID);
                ResultSet rs=ps.executeQuery();
                if(rs.next()) {
```

```java
					hotelRoomReservationSystem = new HotelRoomReservationSystem(rs.getInt(1),rs.getString(2),rs.getInt(3),rs.getString(4),rs.getString(5));
					}
			} catch (SQLException e) {
				System.out.println(e);
			}
			return hotelRoomReservationSystem;
		}

	public HotelRoomReservationSystem[] findAllHotelRoomReservationSystem() {
			List<HotelRoomReservationSystem> hotelRoomReservationSystem = new ArrayList<>();
			try {
				Connection con=HRSUtil.getDBConnection();
				String sql="select * from HotelRoomReservationSystem";
				PreparedStatement ps=con.prepareStatement(sql);
				ResultSet rs=ps.executeQuery();
				while(rs.next()) {
					hotelRoomReservationSystem.add(new HotelRoomReservationSystem(rs.getInt(1),rs.getString(2),rs.getInt(3),rs.getString(4),rs.getString(5)));
				}
			} catch (SQLException e) {
				System.out.println(e);
			}
			return hotelRoomReservationSystem.toArray(new HotelRoomReservationSystem[hotelRoomReservationSystem.size()]);
	}
	}
```

Com.hrs.main :

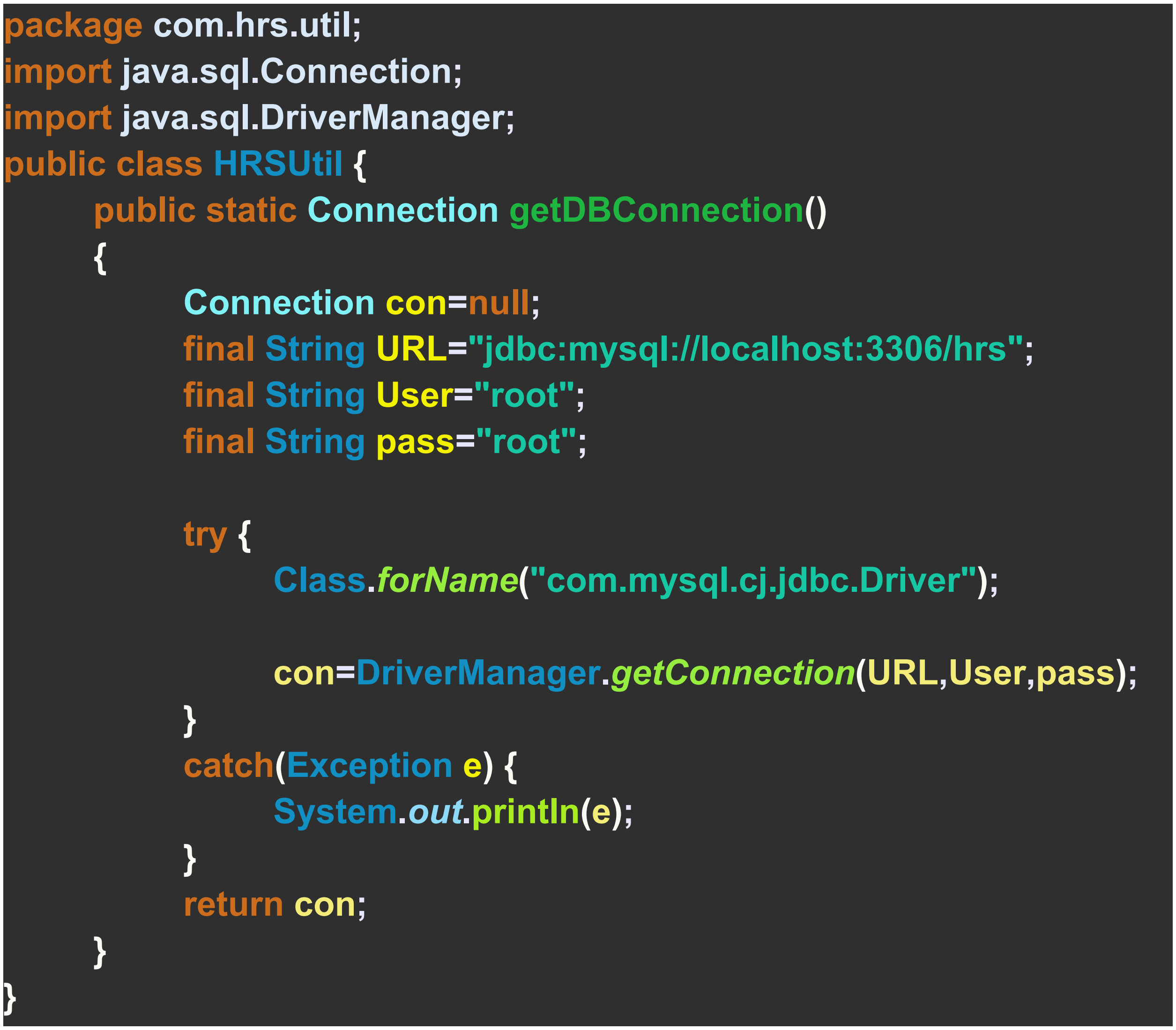
```
package com.hrs.main;
import java.util.Scanner;
import com.hrs.bean.HotelRoomReservationSystem;
import com.hrs.dao.HotelRoomReservationSystemDAO;
public class HRSMain {
static Scanner sc=new Scanner(System.in);

	public static int displaymenu()
	{
		System.out.println("1.Insert  2.Delete  3.Update   4.Find 5.FindAll  6.Exit");
		System.out.println("Enter your Choice");
		int choice=sc.nextInt();
		return choice;
	}

	public static HotelRoomReservationSystem insertHotelRoomReservationSystem()
	{
		int RoomID;
		System.out.println("Enter Room ID: ");
		RoomID=sc.nextInt();

		String RoomType;
		System.out.println("Enter Room Type: ");
		sc.nextLine();
		RoomType=sc.next();
```



```
        int RoomTariff ;
        System.out.println("Enter Room Tariff: ");
        sc.nextLine();
        RoomTariff=sc.nextInt();

        String RoomDesc;
        System.out.println("Enter Room Desc: ");
        sc.nextLine();
        RoomDesc=sc.next();

        String Occupancy;
        System.out.println("Enter Room Occupancy: ");
        sc.nextLine();
        Occupancy=sc.next();


        return new HotelRoomReservationSystem(RoomID,RoomType,RoomTariff,RoomDesc,Occupancy);
    }

    public static int deleteHotelRoomReservationSystem()
    {
        System.out.println("Enter the Room Id to delete ");
        return (sc.nextInt());
    }

    public static HotelRoomReservationSystem updateHotelRoomReservationSystem()
    {
    int RoomID;
    System.out.println("Enter Room ID: ");
    RoomID=sc.nextInt();

    String RoomType;
    System.out.println("Enter Room Type: ");
    sc.nextLine();
    RoomType=sc.next();

    int RoomTariff ;
    System.out.println("Enter Room Tariff: ");
    sc.nextLine();
    RoomTariff=sc.nextInt();

    String RoomDesc;
    System.out.println("Enter Room Desc: ");
    sc.nextLine();
    RoomDesc=sc.next();

    String Occupancy;
    System.out.println("Enter Room Occupancy: ");
    sc.nextLine();
    Occupancy=sc.next();

    return new
HotelRoomReservationSystem(RoomID,RoomType,RoomTariff,Room
Desc,Occupancy);
}

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        String msg=" ";
        int n;
        HotelRoomReservationSystemDAO dao=new
HotelRoomReservationSystemDAO();
        do {
            switch(displaymenu())
            {
            case 1:
                HotelRoomReservationSystem
obs1=insertHotelRoomReservationSystem();

n=dao.insertHotelRoomReservationSystem(obs1);
                if(n==1)
                    System.out.println("Record Inserted
Successfully");
                else
                    System.out.println("Record Insertion
Failure");
                break;

            case 2:
                int
RoomID=deleteHotelRoomReservationSystem();

n=dao.deleteHotelRoomReservationSystem(RoomID);
                if(n==1)
                    System.out.println("Record Deleted
Successfully");
                else
                    System.out.println("Record Deletion
Failure");
                break;


            case 3:
                HotelRoomReservationSystem
obs2=updateHotelRoomReservationSystem();

n=dao.updateHotelRoomReservationSystem(obs2);
                if(n==1)
                    System.out.println("Record Updated
Successfully");
                else
```

```
                                        System.out.println("Record Updation 
Failure");
                                break;

                        case 4:
                                System.out.println("Enter the Room ID to find ");
                                int RoomID1=sc.nextInt();
                                HotelRoomReservationSystem 
obs3=dao.findHotelRoomReservationSystem(RoomID1);
                                if(obs3!=null)
                                        System.out.println(obs3);
                                else
                                        System.out.println("Record Not Found");
                                break;

                        case 5:
                                HotelRoomReservationSystem[] 
obs4=dao.findAllHotelRoomReservationSystem();
                                if(obs4!=null)
                                {
                                        for(HotelRoomReservationSystem s:obs4)
                                                System.out.println(s);
                                }
                                else
                                        System.out.println("No Records Found");
                                break;
                        }

                        System.out.println("Do you wish to 
continue(yes/no)");
                  msg=sc.next();

                } while(msg.equals("yes"));
        }
}
```

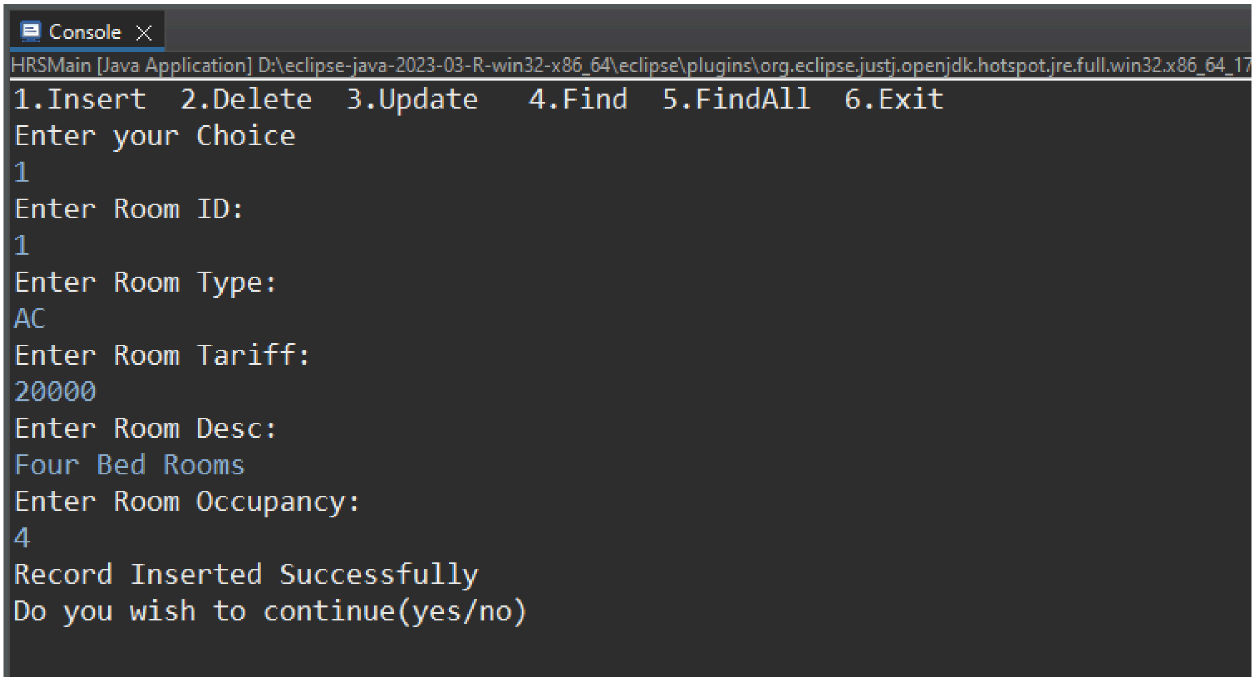
Com.hrs.Util :



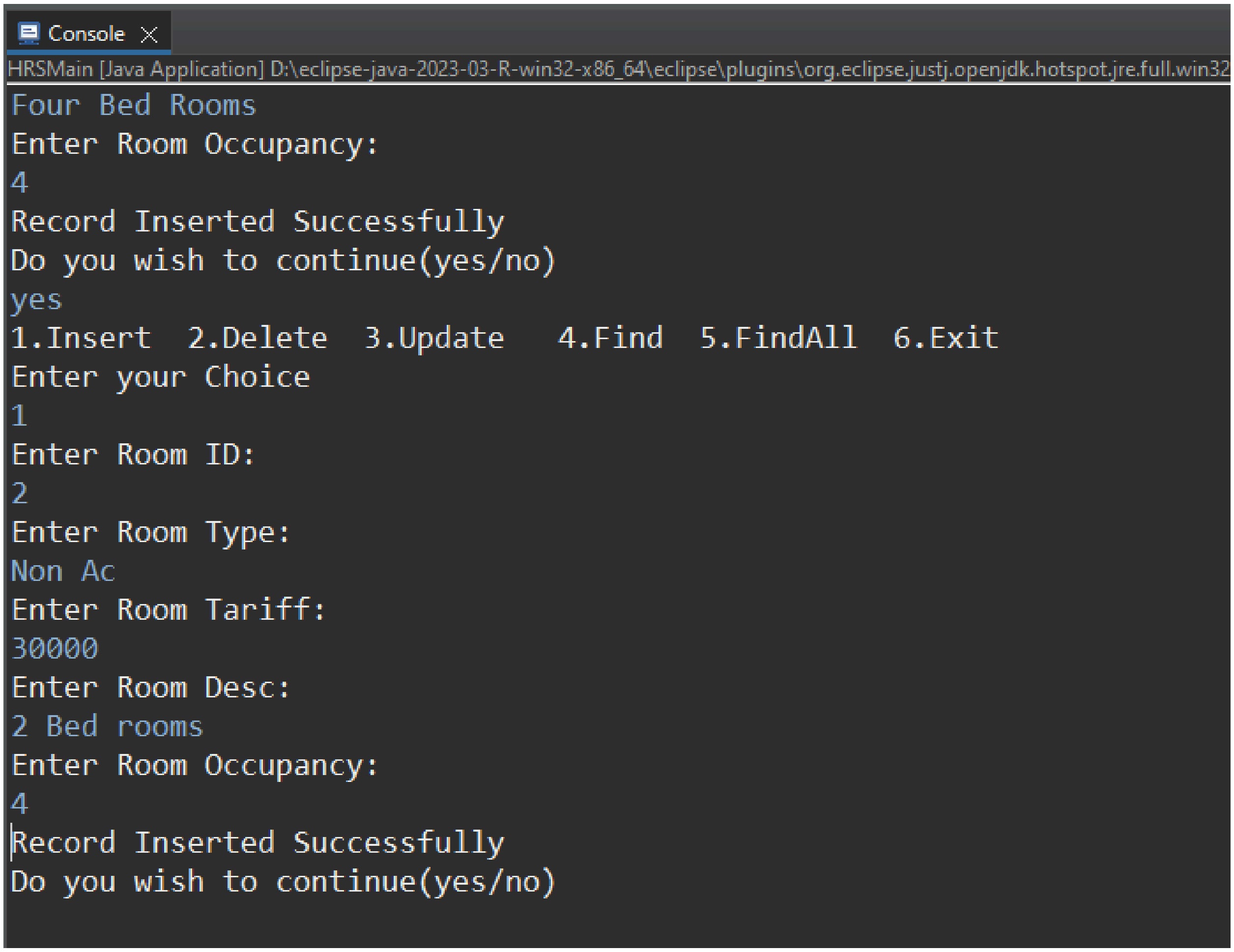
```
package com.hrs.util;
import java.sql.Connection;
import java.sql.DriverManager;
public class HRSUtil {
	public static Connection getDBConnection()
	{
		Connection con=null;
		final String URL="jdbc:mysql://localhost:3306/hrs";
		final String User="root";
		final String pass="root";

		try {
			Class.forName("com.mysql.cj.jdbc.Driver");

			con=DriverManager.getConnection(URL,User,pass);
		}
		catch(Exception e) {
			System.out.println(e);
		}
		return con;
	}
}
```

Crud Operations:

Insertion:

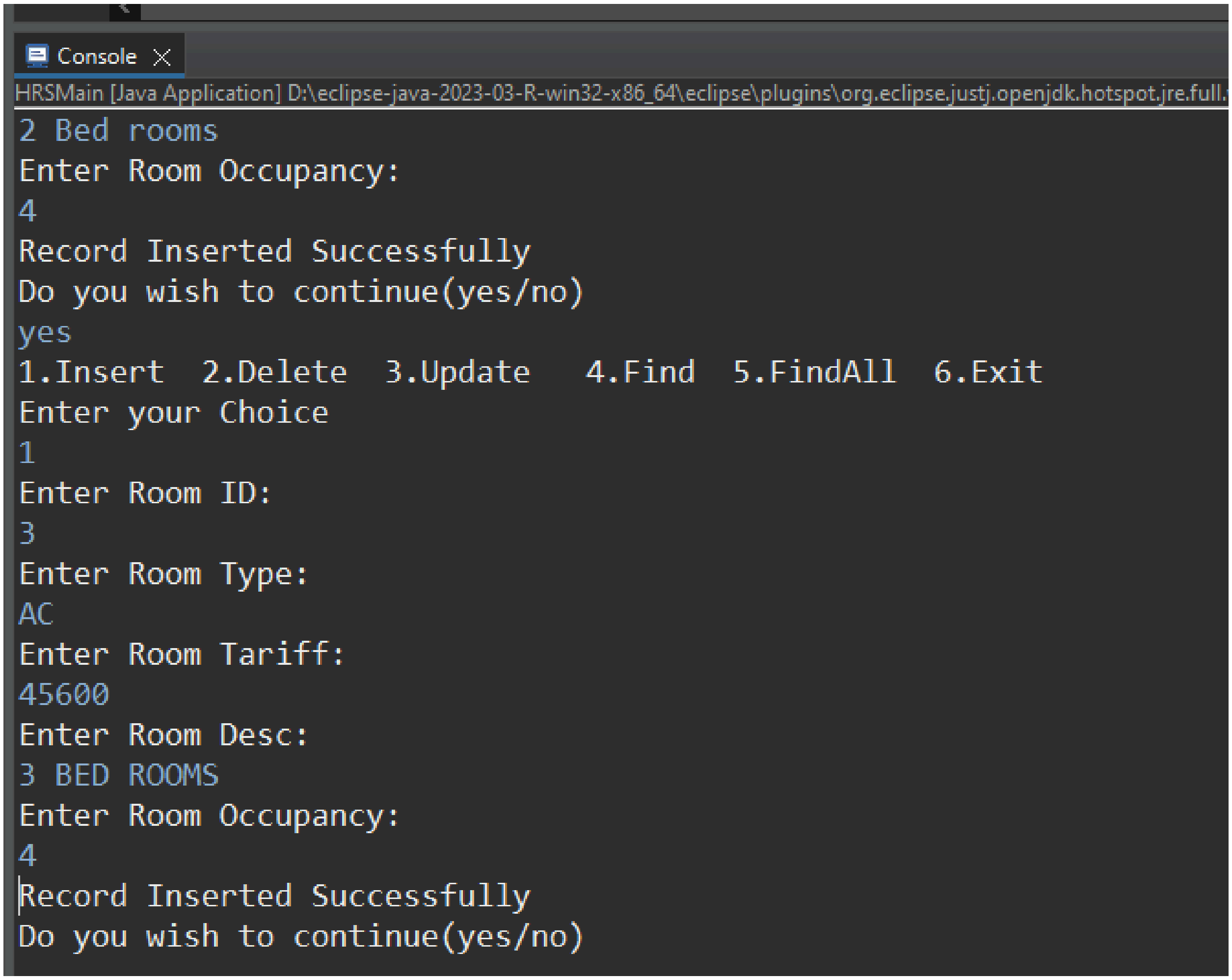


```
HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
1
Enter Room ID:
1
Enter Room Type:
AC
Enter Room Tariff:
20000
Enter Room Desc:
Four Bed Rooms
Enter Room Occupancy:
4
Record Inserted Successfully
Do you wish to continue(yes/no)
```

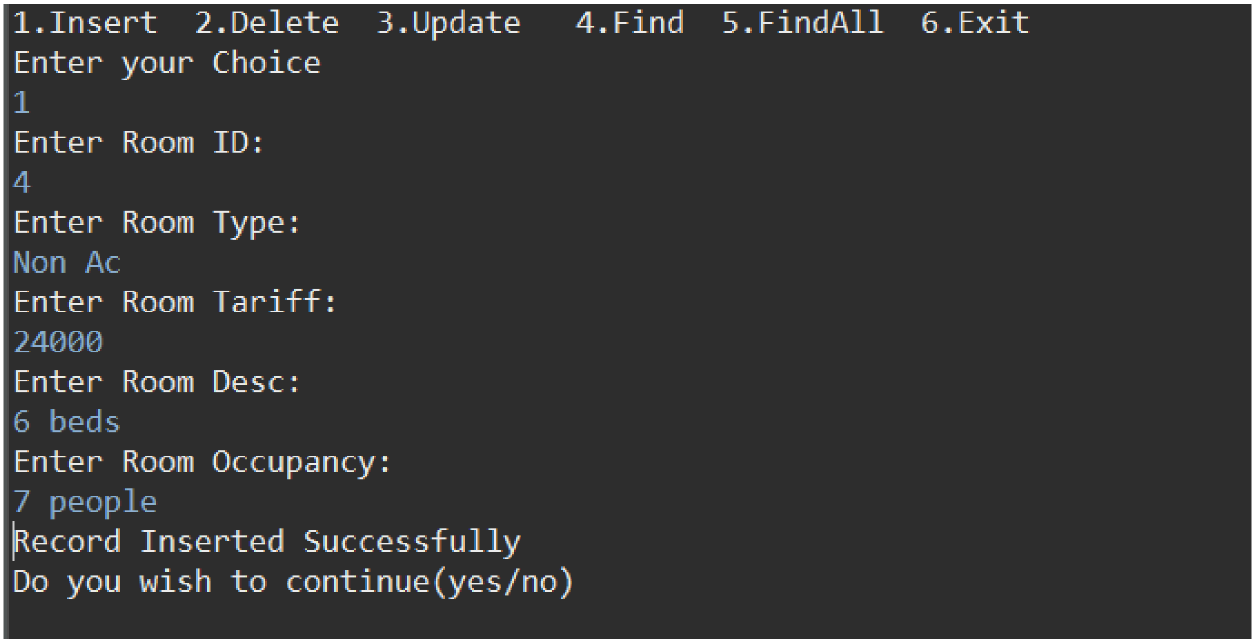


Console
HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32

```
Four Bed Rooms
Enter Room Occupancy:
4
Record Inserted Successfully
Do you wish to continue(yes/no)
yes
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
1
Enter Room ID:
2
Enter Room Type:
Non Ac
Enter Room Tariff:
30000
Enter Room Desc:
2 Bed rooms
Enter Room Occupancy:
4
Record Inserted Successfully
Do you wish to continue(yes/no)
```

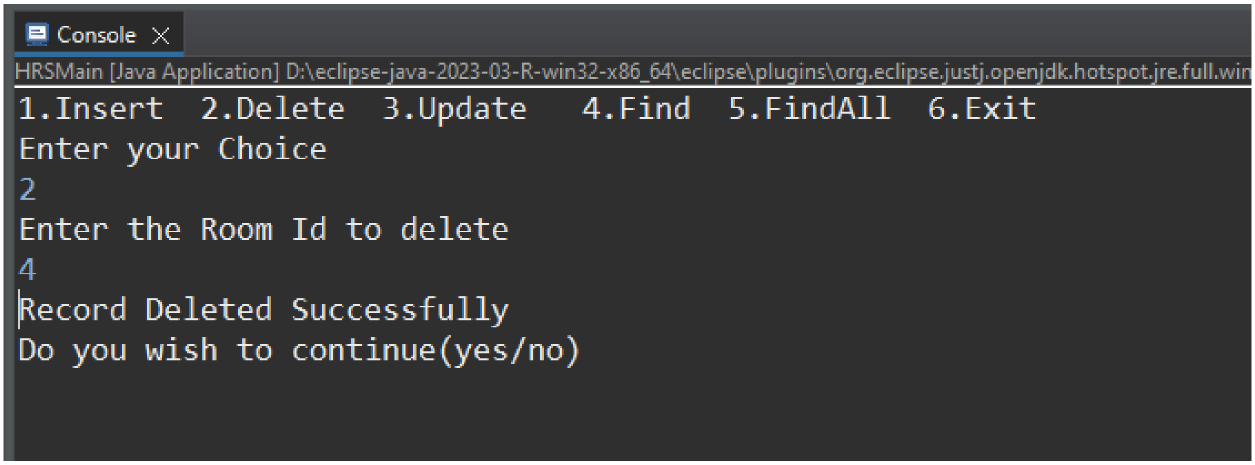


```
HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.
2 Bed rooms
Enter Room Occupancy:
4
Record Inserted Successfully
Do you wish to continue(yes/no)
yes
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
1
Enter Room ID:
3
Enter Room Type:
AC
Enter Room Tariff:
45600
Enter Room Desc:
3 BED ROOMS
Enter Room Occupancy:
4
Record Inserted Successfully
Do you wish to continue(yes/no)
```



```
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
1
Enter Room ID:
4
Enter Room Type:
Non Ac
Enter Room Tariff:
24000
Enter Room Desc:
6 beds
Enter Room Occupancy:
7 people
Record Inserted Successfully
Do you wish to continue(yes/no)
```

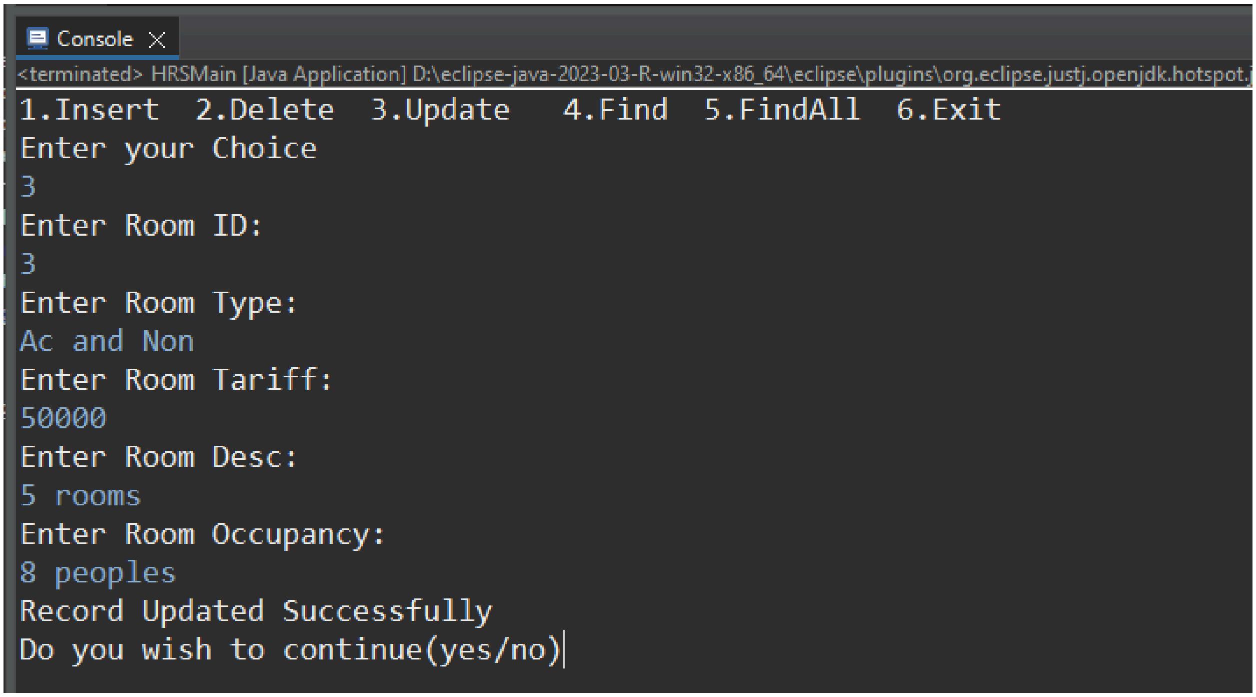
Deletion :



Console

```
HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
2
Enter the Room Id to delete
4
Record Deleted Successfully
Do you wish to continue(yes/no)
```

Updation:

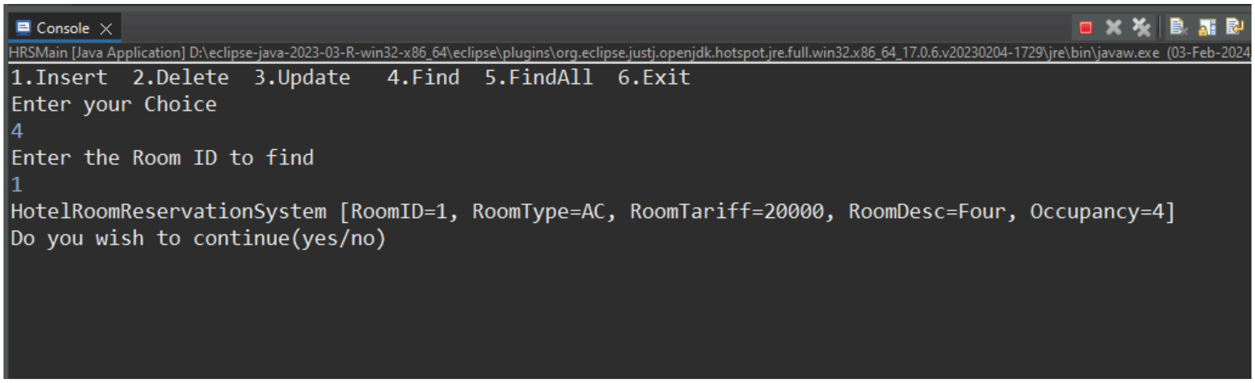


Console

```
<terminated> HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.j
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
3
Enter Room ID:
3
Enter Room Type:
Ac and Non
Enter Room Tariff:
50000
Enter Room Desc:
5 rooms
Enter Room Occupancy:
8 peoples
Record Updated Successfully
Do you wish to continue(yes/no)
```

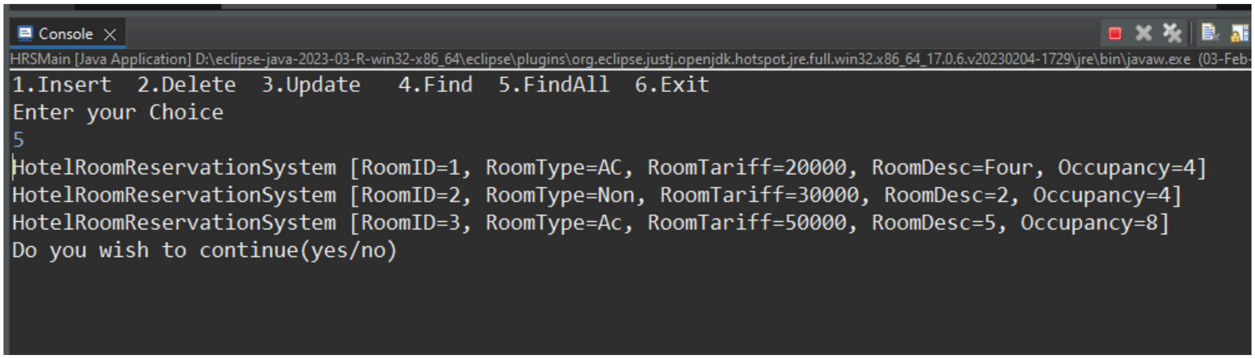
Read :

Find



```
HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe  (03-Feb-2024
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
4
Enter the Room ID to find
1
HotelRoomReservationSystem [RoomID=1, RoomType=AC, RoomTariff=20000, RoomDesc=Four, Occupancy=4]
Do you wish to continue(yes/no)
```

Find All



```
HRSMain [Java Application] D:\eclipse-java-2023-03-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.6.v20230204-1729\jre\bin\javaw.exe  (03-Feb-
1.Insert  2.Delete  3.Update   4.Find  5.FindAll  6.Exit
Enter your Choice
5
HotelRoomReservationSystem [RoomID=1, RoomType=AC, RoomTariff=20000, RoomDesc=Four, Occupancy=4]
HotelRoomReservationSystem [RoomID=2, RoomType=Non, RoomTariff=30000, RoomDesc=2, Occupancy=4]
HotelRoomReservationSystem [RoomID=3, RoomType=Ac, RoomTariff=50000, RoomDesc=5, Occupancy=8]
Do you wish to continue(yes/no)
```