# Hotel Management System

Version 1.0 (2017-2018)

# Computer Science Project

Developed By

Ritesh Kumar,

Ritik Keswani

Delhi Public School, R.K.Puram

www.dpsrkp.net

Index

| Sno | Description | Pageno |
|---|---|---|
| 1 | Certificate | 3 |
| 2 | Acknowledgement & References | 4 |
| 3 | Introduction | 5-7 |
| 4 | Source Code | 8-20 |
| 5 | Output Screen | 21-24 |
| 6 | Hardware & Software requirement | 25 |

# Certificate

This is to certify that **Hotel Management System** ,Computer Science project is developed by **Ritesh Kumar,Ritik Keswani** under my supervision in the computer lab of Delhi Public School, R.K.Puram in the session 2017-2018. The work done by them is original.

Mukesh Kumar
Computer Science Teacher
Date: ____________

# Acknowledgement

I would like to express my sincere gratitude to my computer teacher Mr. Mukesh Kumar for his vital support, guidance and encouragement without which this project would not come forth from  my side. Who helped me completing the project by giving ideas,thoughts and made this project easy and accurate.
I wish to thank my parents for their undivided support and interest who inspired me and encouraged me to go my own way, without which I would be unable to complete my project.

# Reference

1.www.google.com/C++ project
2.www.wikepedia.com/c++ projects
3.Classnotes.

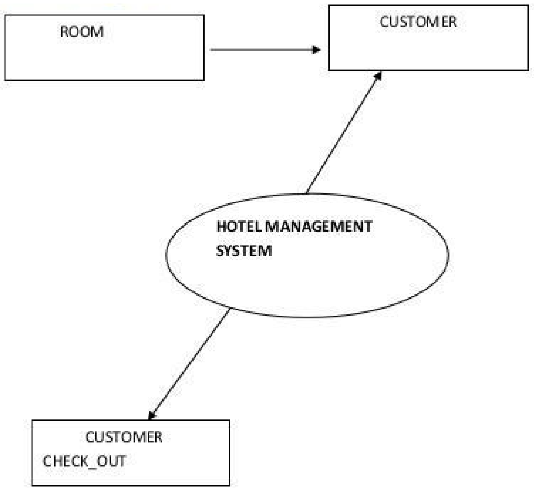
# Introduction

A typical luxury Hotel requires a management system to control its various operations such as maintaining account of all the people in its domain of services,attending to various needs of customers and also achieving increased efficiency in the all working of the hotel itself.

**Purpose Of The System:**

The Hotel Management System aims to make a simpler a staff's attraction with the various modules of the Hotel and ease the process of acquiring information and providing services.

**Context Diagram:**



The Hotel Management System is based on the concept of ***<u>OOPS</u>*** , so the user data and other confidential information remains safe.

**Design Description:**

The program is based on a class as described below:

Class:ROOM

It has following data members in private:

(i) **Rno** of type int for Room No.

(ii)**Type** of type char for  Room Type (SB DB PR SU).

(iii)**Tariff** of type int for Room Tariff.

(iv)**Status** of type int  to Check The Vacancies.

(v)**Name** of type char which has a capacity of 30 characters for Customer Name.
(vi)**Email**of type char which has a capacity of 25 characters for Customer Email Adress.
(vii)**Mobile** of type char  for Customer’s 10 digit  Mobile No..
(viii) **SSno** of type char  for Customer’s 12 digit  Mobile No..
(ix) **NoD** of type int for  Number of Days stayed.
And it has following member function in public visibility mode:
**NewRoomAdd()**-Function designed for  getting values of room no, room type, room  tariff and also it assigns status as 0 which in turn means that room is vacant and then it returns these values to AddRoom Function.

**CheckinDetails()-** Function designed for getting values of Customer’s Details at the time of Check-in.

**CheckoutDetails()**-Function gets called at the time of check-out when Checkout() function is called.  It displays all the information of the customer and assigns room status i.e. Status variable back to 0 indicating the room is vacant again.

**retRno()**-Rno being in private visibility mode cannot  directly accessed from the object so,using this function object can indirectly access  Rno.

**retType()-**Type being in private visibility mode cannot  directly accessed from the object so,using this function object can indirectly access  Type.
**Tariff()**-It returns room tariff for 1 night, which is then used to calculate the total bill cost.

**retName()-**Name being in private visibility mode cannot  directly accessed from the object so, using this function object can indirectly access  Name.

**retEID()**-Email being in private visibility mode cannot  directly accessed from the object so, using this function object can indirectly access  Email.

**retMobile()**-Mobile being in private visibility mode cannot  directly accessed from the object so, using this function object can indirectly access  Mobile.

**retSSno()**-SSno being in private visibility mode cannot directly accessed from the object so, using this function object can indirectly access SSno.

**retStatus()**-It returns Status value which is used by various fuctions such as Checkin().

**retNoD()**-It returns the No of Days since customers check-in.
The first screen of the program is a menu which has following options:

1.Include a new room-This function is made for admin. This options calls a function named AddRoom(). It is used in case a new vacantroom is constructed.

2.Search for a room and check it's status- This option calls SearchRoom() function for its working which takes a room no from the user and checks that room is occupied or vacant, in case the room is occupied it shows the data of the customer staying in that room.

3.Checkout-This options is used at the time of customer's checkout it calls checkout function for its working which makes the room status as vacant.

4.ModifyRoom Details- This option is used in case the customer already staying in the room has changed his details like mobile no .,Email ID etc.This option calls ModifyRoom() function for its working

5.Allotment of New Room-This option is used at the time of Check-in It calls a function Checkin() for its working.The desired room for its entered by the customer and it provides a list of empty room of that type and assigns the room according to user's choice.

6.To Get the List of Reserved Rooms-This option is for admin in case to know how many rooms are occupied along with their respective room number. It calls Report() for its working.

7.To View Transactions History- This function reads an another file named "TRANSACT.DAT" and shows information of all transactions done till now.

# Source Code

```
/*
Project Title:Hotel Management
Version :1.0
Developed By :RiteshKumar,RitikKeswani
School :DPS RK Puram
*/
#include<fstream.h>

#include<conio.h>

#include<string.h>

#include<stdio.h>

#include<ctype.h>


void Checkout();

void AddRoom();

void Checkin();

void DisplayRoom();

void ModifyRoom();

void SearchRoom();

void DeleteRoom();

void Report();

void TransactHis();

void VacantList();


class ROOM

{


int Rno;                        //Room No.

char Type[4];                   //Room Type SB DB PR SU
```

```
float Tariff;                    //Tariff

int Status;                     //Variable to Check The Vacancies


char Name[30];                   //Customer Name

char Email[25];                 //Customer Email Adress

char Mobile[11];                  //Customer Mobile No.

char SSno[13];                    //Social Security No.

long int NoD;                     //Number of Days


public:

void NewRoomAdd()

  {

            cout<<"Room No          :";cin>>Rno;    //Room No.

            cout<<"Type [SB DB PR SU]:";gets(Type);  //Room Type

            cout<<"Tariff                    :";cin>>Tariff; //Tariff

            Status=0;                   //Variable to Check The Vacancies

  }

void CheckinDetails()

  {

            cout<<"Name       :";gets(Name);

            cout<<"Email      :";gets(Email);

            cout<<"Mobile     :";gets(Mobile);

            cout<<"Aadhar  No. :";gets(SSno);

            cout<<"No Of Days  :";cin>>NoD;

             Status=1;

  }

void CheckOutDetails()
```

```
  {
              cout<<"Name      :"<<Name<<endl;
              cout<<"Email     :"<<Email<<endl;
              cout<<"Mobile    :"<<Mobile<<endl;
              cout<<"Aadhar No.  :"<<SSno<<endl;
              Status=0;
  }
void DisplayDetails()
  {
              cout<<"Customer Name               :"<<Name<<endl;
              cout<<"Customer Email ID           :"<<Email<<endl;
              cout<<"Customer Mobile NO.          :"<<Mobile<<endl;
              cout<<"Customer Aadhar No.          :"<<SSno<<endl;
              cout<<"--------------------------------------------"<<endl;
  }

void OutputCustomer()
  {
              if (Status==1)
              {
              cout<<"Customer Name               :"<<Name<<endl;
              cout<<"Customer Email ID           :"<<Email<<endl;
              cout<<"Customer Mobile NO.          :"<<Mobile<<endl;
              cout<<"Customer Aadhar No.          :"<<SSno<<endl;
              cout<<"Status                     :Occupied"<<endl;
              }
  }

int retRno(){return Rno;}
char* retType(){return Type;}
float retTariff(){return Tariff;}
char* retName(){return Name;}
char* retEID(){return Email;}
char* retMobile(){return Mobile;}
char* retSSno(){return SSno;}
int retStatus(){return Status;}
long int retNoD(){return NoD;}
  };

void main()
{ clrscr();
   int a;char Pass[20];int choice2,choice3,choice4;
        do
        { clrscr();
        cout<<"       ------------WELCOME TO OBSIDIAN NIMBUS RESORT------------     "<<endl;
        cout<<"1.Admin"<<endl;
        cout<<"2.User"<<endl;
        cout<<"3.Reports"<<endl;
        cout<<"4.Exit"<<endl;
        cin>>a;
        if(a==1)
        {
                cout<<"Enter Password:"<<endl;
```

```
          gets(Pass);

          if(strcmp(Pass,"password")==0)

          {

           clrscr();

          cout<<"1:Include A New Room" <<endl;

          cout<<"2.:Modify Room Details"<<endl;

          cout<<"3:Transaction History"<<endl;

          cin>>choice2;


          switch(choice2)

          {

                  case 1:AddRoom();break;

                  case 2:ModifyRoom();break;

                  case 3:TransactHis();break;

                  default:cout<<"Wrong Keyword"<<endl;

          }

     }

else

cout<<"Wrong Password"<<endl;

     }


     if(a==2)

     {    clrscr();

          cout<<"1:Checkin"<<endl;

          cout<<"2:Check Status"<<endl;

          cout<<"3:Checkout"<<endl;

          cin>>choice3;
```

```
        switch(choice3)
         {
                case 1:Checkin();break;
                case 2:SearchRoom();break;
                case 3:Checkout();break;
         }
         }
        else if(a==3)
         {     clrscr();
                cout<<"1:List Of Occupied Rooms"<<endl;
                cout<<"2:List Of Vacant Rooms"<<endl;
                cin>>choice4;
                switch(choice4)
                {
                        case 1:Report();break;
                        case 2:VacantList();break;
                }
         }
        else
                cout<<"Thank You"<<endl;

 }
while (a!=4);

getch();
}
```

```
void AddRoom()
{        clrscr();
         fstream fil;
         fil.open("ROOM.DAT",ios::binary|ios::out);
         ROOM R1;char Choice;
         do
         {
                  cout<<"<<Enter Room Details>>"<<endl;
                  R1.NewRoomAdd();
                  fil.write((char*)&R1,sizeof(R1));
                  cout<<"Enter Y To Include More New Rooms Or Enter N To GO Back"<<endl;
                  cin>>Choice;
                   Choice=toupper(Choice);
         }
         while(Choice=='Y');
         fil.close();
         getch();
         }

void Checkin()
 {
         clrscr();
         fstream fil,fil2;
         fil.open("ROOM.DAT",ios::binary|ios::in|ios::out);
         fil2.open("TRANSACT.DAT",ios::binary|ios::app);
             ROOM R1;   char TypeR[4],Response;      int Chin=0;
         cout<<"Type of room require [SB DB PR SU]:";gets(TypeR);
```

```
          while(Chin==0 &&fil.read((char*)&R1,sizeof(R1)))
    {
          if(strcmpi(R1.retType(),TypeR)==0 && R1.retStatus()==0)
      {
                    cout<<"Room Available:"<<R1.retRno()<<endl;
                    cout<<"Checkin (Y/N)?";cin>>Response;
                      Response=toupper(Response);
                    if (Response=='Y')
                      {
                              R1.CheckinDetails();
                              fil.seekp(fil.tellg()-sizeof(ROOM));
                              fil.write((char*)&R1,sizeof(ROOM));
                              fil2.write((char*)&R1,sizeof(ROOM));
                                Chin++;
                      }
            }
   }
if (Chin==0)
          cout<<"Room not availble!!!"<<endl;


fil.close();
getch();
 }




void SearchRoom()
{         clrscr();
```

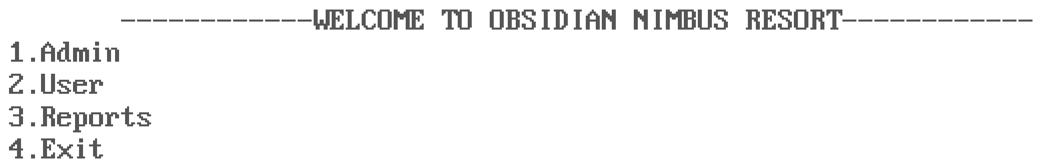
```
          fstream fil;

          fil.open("ROOM.DAT",ios::binary|ios::in);

          ROOM R1;

          int flag=0,Search;

          cout<<"Enter The Room No. to be Searched"<<endl;

          cin>>Search;

          while(fil.read((char*)&R1,sizeof(R1))&&flag==0)

   {

          if(R1.retRno()==Search)

       {

          R1.OutputCustomer();

          if(R1.retStatus()==0)

          cout<<"Room Is Vacant!"<<endl;

          flag++;

       }

   }

          if(flag==0)

                    cout<<"Room Not Found!"<<endl;

fil.close();

getch();

 }




void ModifyRoom()

 {

          clrscr();
          char A;

          fstream fil;

          fil.open("ROOM.DAT",ios::binary|ios::in|ios::out);

            ROOM R1,R2;

          int Rsc,flag=0;

          cout<<"Enter The Room No. to be Modified"<<endl;

          cin>>Rsc;

          while(fil.read((char*)&R1,sizeof(R1)))

                    if(Rsc==R1.retRno())

    {

                              R1.CheckinDetails();

                              int current=fil.tellg();

                              fil.seekp(current-sizeof(R1));

                              fil.write((char*)&R1,sizeof(R1));

                              flag++;

    }

if(flag==0)

                    cout<<"Room Not Found"<<endl;

fil.close();

getch();

 }

void Checkout()

{

          clrscr();

          int Check;

          cout<<"Enter The Room No For Checkout:";

          cin>>Check;
```

```
        fstream fil,fil2;

        ROOM R1; int flag=0;

        fil.open("ROOM.DAT",ios::in|ios::binary);

        fil2.open("TEMP.DAT",ios::binary|ios::out);

        while(fil.read((char*)&R1,sizeof(R1)))

         {

                if(R1.retRno()==Check&&R1.retStatus()==1)

           {

                R1.CheckOutDetails();

                fil2.write((char*)&R1,sizeof(ROOM));

                float Cost=R1.retTariff()*R1.retNoD();

                cout<<"Your Bill is Rs.:"<<Cost<<endl;

                flag++;

         }

                else

                        fil2.write((char*)&R1,sizeof(R1));

         }

        if(flag==0)

                cout<<"The room is Vacant"<<endl;

fil.close();

fil2.close();

remove("ROOM.DAT");

rename("TEMP.DAT","ROOM.DAT");

getch();

}


void Report()
```

```
{
          clrscr();
                    fstreamfil;
                    ROOM R1;int i=0;int flag=0;
                    fil.open("ROOM.DAT",ios::in|ios::binary);

                    while(fil.read((char*)&R1,sizeof(R1)))
          {
                              if(R1.retStatus()==1)
                    {
                                        cout<<"Following Rooms Are Reserved"<<endl;
                                        cout<<++i<<"."<<R1.retRno()<<"-"<<R1.retType()<<endl;
                                        flag++;
                    }
                    if(flag==0)
                    {
                              clrscr();
                              cout<<"No Room Reserved"<<endl;
                    }
    }
fil.close();
getch();
}
void TransactHis()
{
          clrscr();
          fstream fil;
```

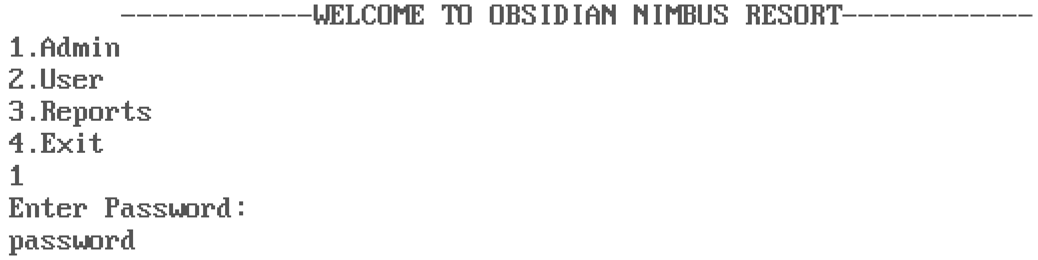
```
{

        ROOM R1;

        fil.open("TRANSACT.DAT",ios::binary|ios::in);

        cout<<"Details Of All Transactions Done Till Now Are :"<<endl;

        while(fil.read((char*)&R1,sizeof(ROOM)))

         {


                R1.DisplayDetails();

         }

fil.close();

getch();

}

voidVacantList()

{  clrscr();

        fstream fil;

        ROOM R1;int i=1;int flag=0;

        fil.open("ROOM.DAT",ios::in|ios::binary);

        cout<<"Following Rooms Are Vacant:"<<endl;

        while(fil.read((char*)&R1,sizeof(R1)))

        if(R1.retStatus()==0)

    {

                cout<<i<<"."<<R1.retRno()<<"-"<<R1.retType()<<endl;

                i++;

                flag++;

    }

        if(flag==0)

   {

                cout<<"None Of The Rooms Are Vacant:"<<endl;
    }
fil.close();
getch();
  }
```

OUTPUT SCREEN
1.



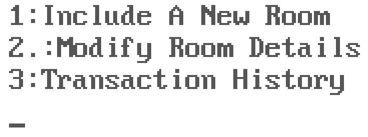
```
       ------------WELCOME TO OBSIDIAN NIMBUS RESORT------------
1.Admin
2.User
3.Reports
4.Exit
```

2.



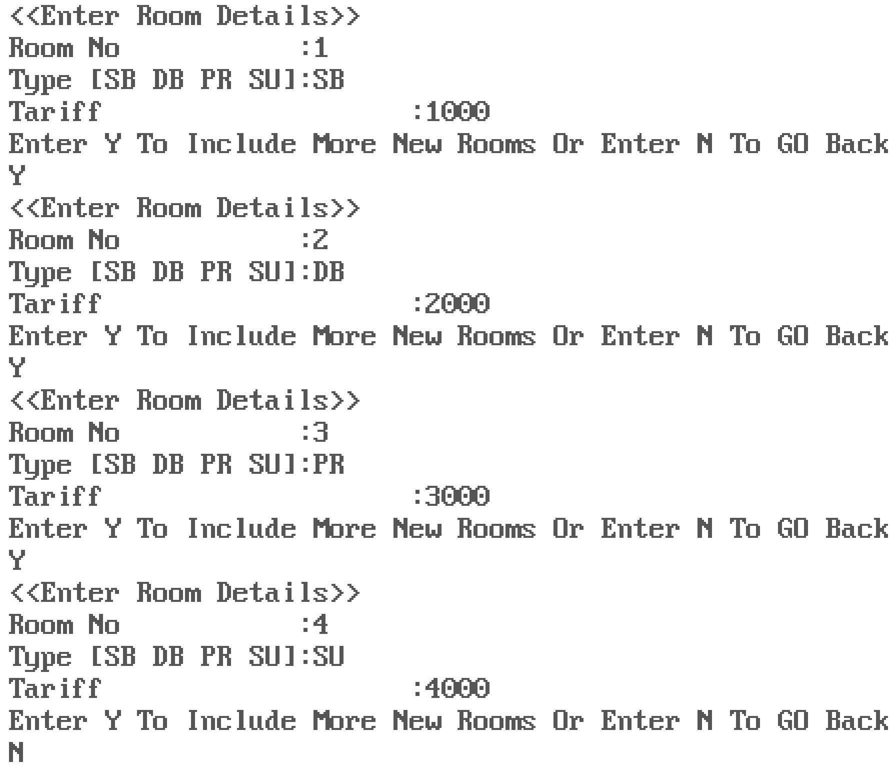
```
       ------------WELCOME TO OBSIDIAN NIMBUS RESORT------------
1.Admin
2.User
3.Reports
4.Exit
1
Enter Password:
password
```

3.



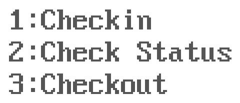
```
1:Include A New Room
2.:Modify Room Details
3:Transaction History
_
```

4.



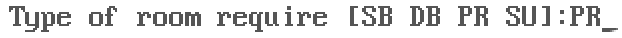
```
<<Enter Room Details>>
Room No           :1
Type [SB DB PR SU]:SB
Tariff                   :1000
Enter Y To Include More New Rooms Or Enter N To GO Back
Y
<<Enter Room Details>>
Room No           :2
Type [SB DB PR SU]:DB
Tariff                   :2000
Enter Y To Include More New Rooms Or Enter N To GO Back
Y
<<Enter Room Details>>
Room No           :3
Type [SB DB PR SU]:PR
Tariff                   :3000
Enter Y To Include More New Rooms Or Enter N To GO Back
Y
<<Enter Room Details>>
Room No           :4
Type [SB DB PR SU]:SU
Tariff                   :4000
Enter Y To Include More New Rooms Or Enter N To GO Back
N
```

5.



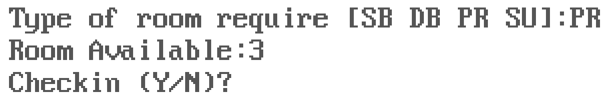
```
1:Checkin
2:Check Status
3:Checkout
```

6.



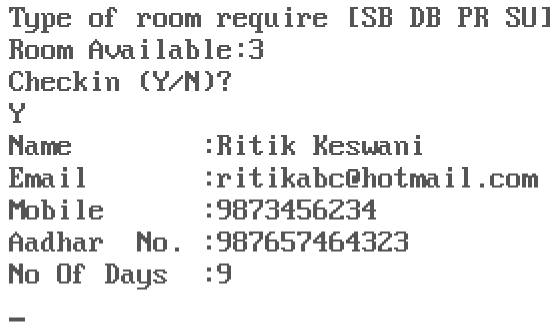
```
Type of room require [SB DB PR SU]:PR_
```

7.



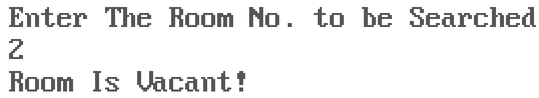
```
Type of room require [SB DB PR SU]:PR
Room Available:3
Checkin (Y/N)?
```

8.



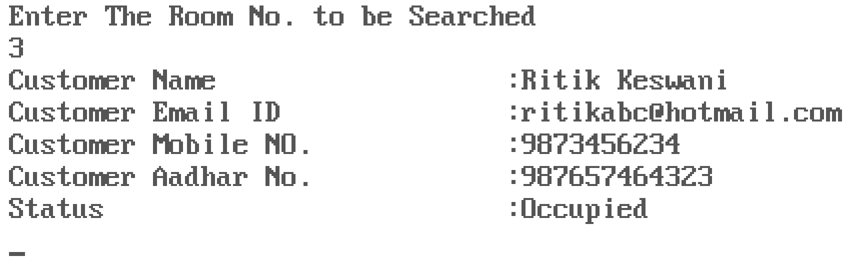
```
Type of room require [SB DB PR SU]:PR
Room Available:3
Checkin (Y/N)?
Y
Name         :Ritik Keswani
Email        :ritikabc@hotmail.com
Mobile       :9873456234
Aadhar  No.  :987657464323
No Of Days   :9
_
```

9.



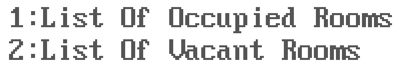
```
Enter The Room No. to be Searched
2
Room Is Vacant!
```

10.



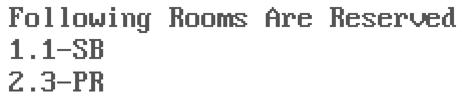
```
Enter The Room No. to be Searched
3
Customer Name                  :Ritik Keswani
Customer Email ID              :ritikabc@hotmail.com
Customer Mobile NO.            :9873456234
Customer Aadhar No.            :987657464323
Status                         :Occupied
_
```

11.



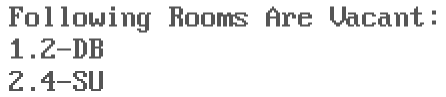
```
1:List Of Occupied Rooms
2:List Of Vacant Rooms
```

12.



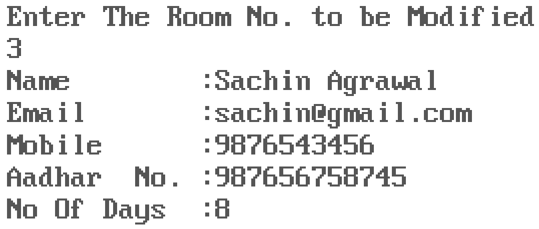
```
Following Rooms Are Reserved
1.1-SB
2.3-PR
```

13.



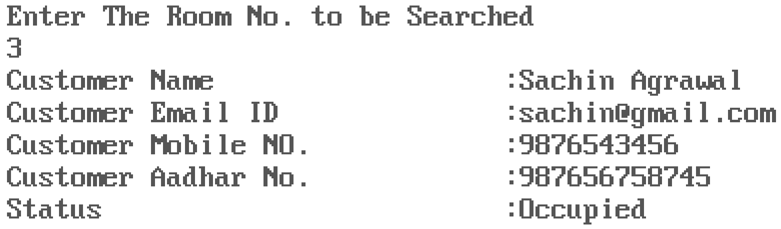
```
Following Rooms Are Vacant:
1.2-DB
2.4-SU
```

14.



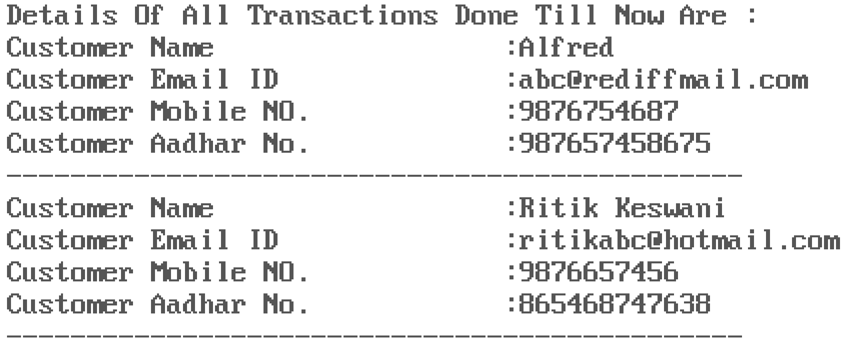
```
Enter The Room No. to be Modified
3
Name         :Sachin Agrawal
Email        :sachin@gmail.com
Mobile       :9876543456
Aadhar  No. :987656758745
No Of Days  :8
```

15.



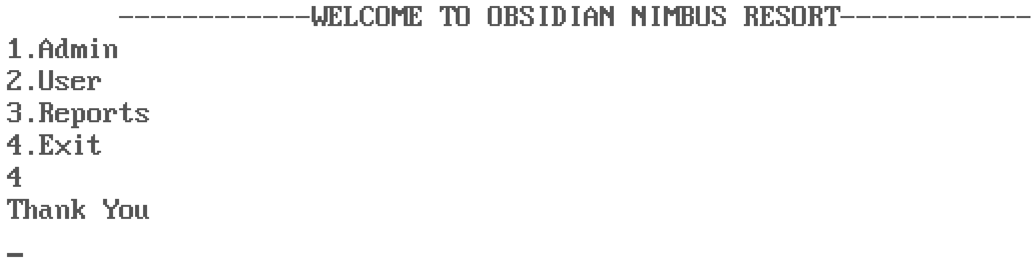
```
Enter The Room No. to be Searched
3
Customer Name                    :Sachin Agrawal
Customer Email ID                :sachin@gmail.com
Customer Mobile NO.              :9876543456
Customer Aadhar No.              :987656758745
Status                           :Occupied
```

**16.**



```
Details Of All Transactions Done Till Now Are :
Customer Name                    :Alfred
Customer Email ID                :abc@rediffmail.com
Customer Mobile NO.              :9876754687
Customer Aadhar No.              :987657458675
----------------------------------------------
Customer Name                    :Ritik Keswani
Customer Email ID                :ritikabc@hotmail.com
Customer Mobile NO.              :9876657456
Customer Aadhar No.              :865468747638
----------------------------------------------
```

17.



```
       ------------WELCOME TO OBSIDIAN NIMBUS RESORT------------
1.Admin
2.User
3.Reports
4.Exit
4
Thank You

_
```

## Hardware & Software Requirement

### Hardware Requirement

Intel core/i3/i5/i7 or any equivalent

With at least 256 MB RAM

2 MB free space on Hard Disk

Color Monitor/LCD

### Operating System & Compiler

MS Windows

Turbo C++ 3.1 Compiler