HOTEL MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

NAMAN L UPADHYAY(17BCE0980)

KUMAR HARSH (17BCE0993)

DATA STRUCTURES AND ALGORITHMS (CSE 2003)

VIT®
UNIVERSITY
(Estd. u/s 3 of UGC Act 1956)
VELLORE ▪ CHENNAI
www.vit.ac.in

APRIL, 2018

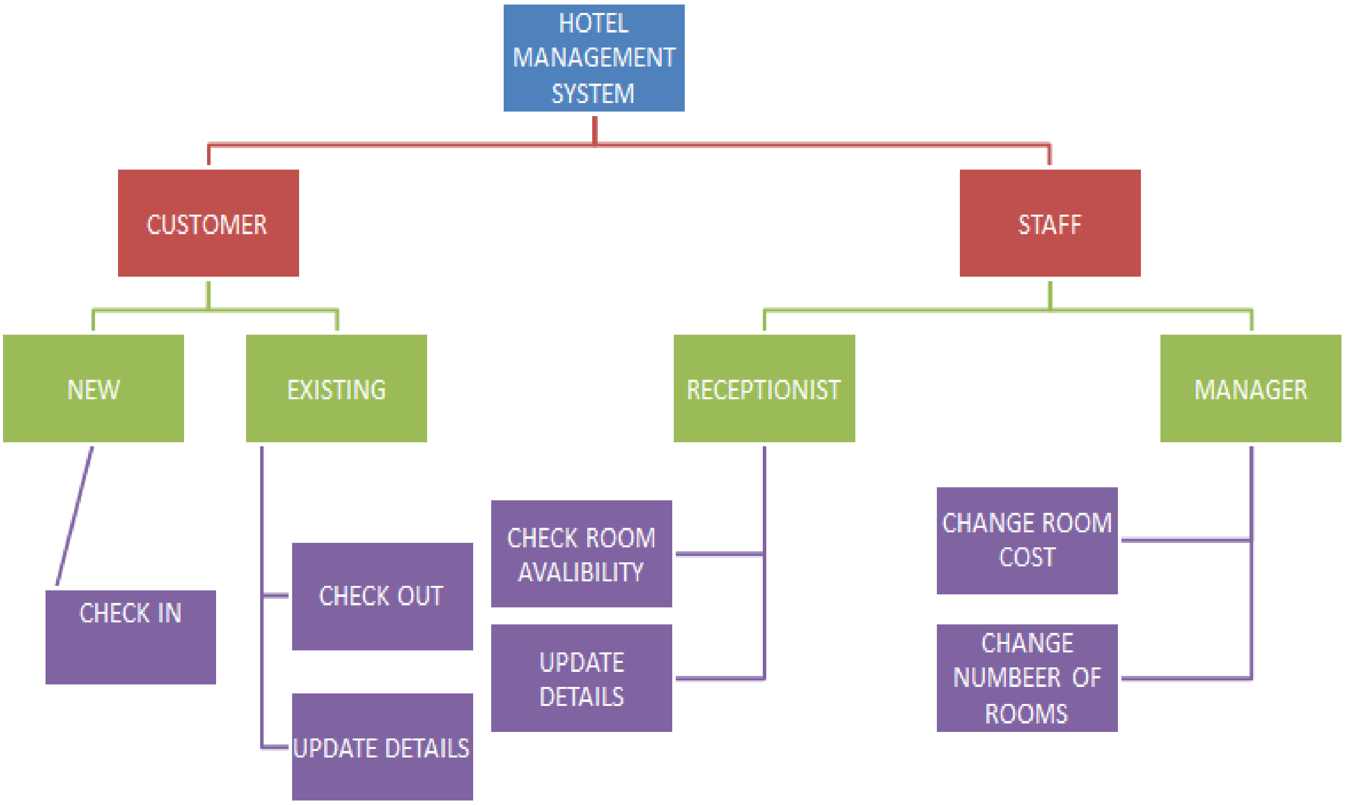
INTRODUCTION

Data Structures are the building blocks of every efficient program. There are several data structures widely used for problem solving, be it array, stack, queue, linked lists, etc. To create the backend working mechanism of the program, we have decided to use linked list. Our project provides a common platform for both customers as well as staff of the hotel. This makes the conduction of hotel management fluid and automated thus reducing manual work. Throughout the implementation of the project we have created every operation using linked list only to understand all the principal operations on linked lists like insertion of nodes at various position, deletion of nodes at different positions, searching elements in the list and updating of elements in the list. Our program serves as a common application to three types of operators: customer, receptionist and manager. We have provided ample amount of operation for each person to operate. The customers can check in, update their details; the receptionist can check room availability and the manager can change the cost and number of available rooms in the hotel. Our program is user-friendly and provides accurate and error free output.

PROBLEM STATEMENT

Before the execution of our project plan, we gained some information which could prove pivotal in the successful completion of our data structures project. The problem which is being analysed is HOTEL MANAGEMENT. The issues that are faced by every hotel management is maintenance of record. Many of the hotel in India are still using book method. So this program helps them to keep track of all the records and helps them in maintaining data.

PROPOSED METHODOLOGY/PROGRAM ARCHITECTURE



IMPLEMENTATION - I

- The user will have two options, either he can operate as a staff member or as a customer.
- If user chooses to be a customer, then he will have choice to be a new customer of existing customer.
- Being a new customer, he can check in by giving his personal details and choosing room type
- Being an existing customer, he can checkout or update his details.

IMPLEMENTATION - II

- If the user chooses to be a staff member, then he will have two choices, he can either be receptionist or Manager.

- If the user chooses to be the receptionist, he can check room availability as well as update the details of the customers. He can also see all the list of customers that are currently staying in the hotel.
- We also have a place for a manager who has all the features that receptionist has plus he will possess some extra features like he can increase or decrease the room price or change the residential structure by changing the number and type of rooms.

DSA CONCEPTS AND SOFTWARES USED

LINKED LIST:-

- Linked list acts as a pillar in the development of our program. We have used its unique advantages like dynamic size and ease of insertion and deletion to create a highly efficient program.
- Linked list doesn’t have pre-defined size and we have used it for our advantage to avoid wastage of memory.
- We have also learnt that insertion and deletion operations in link list are easy and more efficient than arrays and other data structures.

C++-LANGUAGE:-

- An important advantage of C++ is its ability to extend itself. A C++ program is basically a collection of functions that are supported by the C++ library this makes it easier for us to add our own functions to C++ library. Due to the availability of large number of functions, the programming task becomes simple.
- In C++ language, it is easier for us to think of a problem in terms of function modules or blocks. We can use the collection of these modules to make a complete program. This modular structure makes program debugging, testing and maintenance easier.

DYNAMIC MEMORY ALLOCATION AND REALLOCATION:-

- We have gained important knowledge about the dynamic memory allocation in C++-programming. In C++-programming, memory can be dynamically allocated using library functions- malloc from a memory called heap.
- In the same way, we can reuse the previously allocated memory by using library function- realloc or free.
- Through the implementation of the program, we have also learnt about structures to create a “user defined data type”.

PROGRAM:

```
#include<iostream>
#include<conio.h>
#include<stdlib.h>
using namespace std;
int mpass=2222, rpass=1111;
int oneroom=10, tworoom=10, threeroom=10, threebroom=10;
int oneroomo=0, tworoomo=0, threeroomo=0, threebroomo=0;
int outstat1=0,out=0,out1=0,out_1=0;
int onerp=5000, tworp=9000, threerp=12500, threebrp=15000;
struct node
{
    int room_no;
    int days;
    char name[200];
    int mobile;
    struct node*next;
};

struct node *head=NULL, *tail, *temp, *t, *a, *b, *p, *q;
int chnge_room_No(int cho)
{
    switch(cho){
        case 1:
            cout<<"Enter new number of One bedded room";
            cin>>oneroom;
            cout<<"Number of rooms have been updated!!";
            break;
        case 2:
            cout<<"Enter new number of Two bedded room";
            cin>>tworoom;
            cout<<"Number of rooms have been updated!!";
            break;
        case 3:
            cout<<"Enter new number of Three bedded room";
            cin>>threeroom;
            cout<<"Number of rooms have been updated!!";
            break;
        case 4:
            cout<<"Enter new number of Three bedded beach
view room";
            cin>>threebroom;
            cout<<"Number of rooms have been updated!!";
            break;
        default:
            cout<<"Wrong choice!!";
            break;
    }
}
void change_roomPrice(int chi)
{
    switch(chi){
        case 1:
            cout<<"Enter New price of one bedded room";
            cin>>onerp;
            cout<<"Price have been updated!!";
            break;
        case 2:
            cout<<"Enter New price of two bedded room";
            cin>>tworp;
            cout<<"Price have been updated!!";
            break;
        case 3:
            cout<<"Enter New price of one bedded room";
            cin>>threerp;
            cout<<"Price have been updated!!";
            break;
        case 4:
            cout<<"Enter New price of one bedded room";
            cin>>threebrp;
            cout<<"Price have been updated!!";
            break;
        default:
            cout<<"Wrong choice!!";
            break;
    }
}
int menu()
{
    int ret;
    cout<<"\nMain Menu";
    cout<<"\n1. CUSTOMER\n2. STAFF\n3. EXIT\n";
    cout<<"\nEnter your choice: ";
    cin>>ret;
    return ret;
}

int cust_menu()
{
    int ret;
    cout<<"\n  Customer Menu";
    cout<<"\n 1. New Customer \n 2. Existing customer \n
3. EXIT \n 4. Back to Main Menu\n";
    cout<<"\nEnter your choice: ";
    cin>>ret;
    return ret;
}

int staff_menu()
{
    int ret;
    cout<<"\n  Staff Menu";
    cout<<"\n 1. Manager  \n 2. Receptionist \n 3. EXIT \n
4. Back to Main Menu\n\n";
    cout<<"\nEnter your choice: ";
    cin>>ret;
    return ret;
}
```

```
int room_menu()

{

    int ret;

    cout<<"\n  Room Menu";

    cout<<"\n 1. One-Bedded Room - Rs."<<onerp<<" \n 2.
Two-Bedded Room - Rs."<<tworp<<"\n 3. Three-Bedded Room -
Rs."<<threerp<<"\n 4. Three-Bedded Beach View Room -
Rs."<<threebrp<<"\n  5.EXIT\n";

    cout<<"\nEnter your choice: ";

    cin>>ret;

    return ret;

}



int existingcustomer_menu()

{

    int ret;

    cout<<"\n  Existing Customer Menu";

    cout<<"\n 1. See your details \n 2. Update your
deatils \n 3. Checkout\n 4. Back to Main Menu\n";

    cout<<"\nEnter your choice: ";

    cin>>ret;

    return ret;

}


int manager_menu()

{

    int ret;
    cout<<"\n  Manager Menu";
    cout<<"\n 1. Room Avaibility \n 2. Room No. change \n
3. Room Price change \n 4. Back to Main Menu\n";
    cout<<"\nEnter your choice: ";
    cin>>ret;
    return ret;
}

int receptionist_menu()
{
    int ret;
    cout<<"\n  Receptionist Menu";
    cout<<"\n 1. Room Avaibility \n 2. Customer details
Update  \n 4. Back to Main Menu\n";
    cout<<"\nEnter your choice: ";
    cin>>ret;
    return ret;
}

int room_allotment()
{
    int outstat=0;
    cout<<"Select Room Type:";
    while(1&&(outstat!=1))
    {
        int rch=room_menu();
        switch(rch)
        {
            case 1: if (oneroom>0)
        {
            (temp->room_no)=(100+oneroom);
            oneroom--;
            outstat=1;
            break;
        }
        else
        {
            break;
        }
case 2: if (tworoom>0)
        {
            (temp->room_no)=(200+tworoom);
            tworoom--;
            outstat=1;
            break;
        }
        else
        {
            break;
        }
case 3: if (threeroom>0)
        {
            (temp->room_no)=(300+threeroom);
            threeroom--;
            outstat=1;
            break;
        }
```

```
                    else
                    {
                        break;
                    }
            case 4: if (threebroom>0)
                    {
                        (temp->room_no)=(400+threebroom);
                        threebroom--;
                        outstat=1;
                        break;
                    }
                    else
                    {
                        break;
                    }
            default: cout<<"Wrong Choice!!";
                        break;
        }

    }
}

void insert()
{
    temp=(struct node*)malloc(sizeof(struct node));
    temp->next=NULL;
    cout<<"Enter name:";
    int s=0;
    cin.getline(temp->name,200,'\t');
    int room_coice=room_allotment();
    cout<<"Enter no.of days:";
    cin>>temp->days;
    cout<<"Enter mobile no.:";
    cin>>temp->mobile;
    if(head==NULL)
    {
        head=tail=temp;
    }
    else
    {
        temp->next=head;
        head=temp;
    }
}


int search()
{
    int ele,i=0;
    cout<<"\nPlease enter your Mobile No.:";
    cin>>ele;
    t=head;
    while(t!=NULL)
    {
        if(t->mobile==ele)
            return i+1;
        t=t->next;
        i++;
    }
    return -1;

}

void disp(int pos)
{
        int tt=1;
        p=head;
        while(tt<pos)
        {
            p=p->next;
            tt++;
        }
            cout<<"\n\nSummary:";
            cout<<"\nRoom No.: ";
            cout<<p->room_no;
            cout<<"\nName: ";
            cout<<p->name;
            cout<<"\nNo of days: ";
            cout<<p->days;
            cout<<"\nMobile No.: ";
            cout<<p->mobile;

}
```

```
void update_no(int pos)
{
    int nno;
    int tt=1;
    p=head;
    while(tt<pos)
    {
        p=p->next;
        tt++;
    }
    cout<<"Enter new no.:";
    cin>>nno;
    p->mobile=nno;

    cout<<"\n\nSummary:";
        cout<<"\nRoom No.: ";
        cout<<p->room_no;
        cout<<"\nName: ";
        cout<<p->name;
        cout<<"\nNo of days: ";
        cout<<p->days;
        cout<<"\nMobile No.: ";
        cout<<p->mobile;

}

void room_out_chk(int no)
{
    no=(no/100);
    if (no==4)
        threebroomo++;
    else if (no==3)
        threeroomo++;
    else if (no==2)
        tworoomo++;
    else if (no==1)
        oneroomo++;
}

void calc_bill(int a, int b)
{
    int amount;
    switch(a)
    {
        case 1: amount=onerp* b;
                cout<<"\nYOUR TOTAL BILL:"<<amount;
                break;
        case 2: amount=tworp* b;
                cout<<"\nYOUR TOTAL BILL:"<<amount;
                break;
        case 3: amount=threerp* b;
                cout<<"\nYOUR TOTAL BILL:"<<amount;
                break;
        case 4: amount=threebrp* b;
                cout<<"\nYOUR TOTAL BILL:"<<amount;
                break;
    }
}

void checkout(int pos)
{
        int i=0;
        p=head;
        for(i=0;i<pos-1;i++)
            p=p->next;

        temp=p;
        cout<<"\n\nSummary:";
        cout<<"\nRoom No.: ";
        cout<<temp->room_no;
        cout<<"\nName: ";
        cout<<temp->name;
        cout<<"\nNo of days: ";
        cout<<temp->days;
        cout<<"\nMobile No.: ";
        cout<<temp->mobile;
        calc_bill(temp->room_no/100,temp->days);
        cout<<"\nCustomer checked out.";
        room_out_chk(temp->room_no);
        p=temp->next;
        free(temp);

}
```

```
int main()
{

    cout<<"\n\t\t\t****************************";
    cout<<"\n\t\t\t* HOTEL MANAGEMENT PROJECT *";
    cout<<"\n\t\t\t****************************";
    cout<<"\n\n\n";
    int ch;
    while(1)
    {
        ch=menu();
        switch(ch)
        {
            case 1: {
                    system("cls");
                    int chx;

                    do
                    {
                        chx=cust_menu();
                        switch(chx)
                        {
                            case 1:     system("cls");
                                        cout<<"New
Customer Checkin:\n\n ";
                                        insert();
                                        system("cls");
                                        cout<<"\nRoom
Allotment succesful!";

cout<<"\n\n\tSummary:";
                                        cout<<"\nRoom No.:
";
                                        cout<<head-
>room_no;
                                        cout<<"\nName: ";
                                        cout<<head->name;
                                        cout<<"\nNo of
days: ";
                                        cout<<head->days;
                                        cout<<"\nMobile
No.: ";
                                        cout<<head-
>mobile;
                                        break;


                            case 2:     system("cls");
                                        int che;

                                        do
                                        {

    che=existingcustomer_menu();
                                            switch(che)
                                            {
                                                case 1:
```

```
                                                                  {
    int pos=search();
                                                                  if
(pos==-1)
                                                                  {
    cout<<"Details not found\n";
                                                                  }
    else
                                                                  {
    disp(pos);
                                                                  }
    outstat1=1;
    break;
                                                                  }
                                                         case 2:
                                                                  {
    system("cls");
    cout<<"Update......";
    int pos=search();
                                                                  if
(pos==-1)
```

```
                                                        {

    cout<<"Details not found\n";
                                                        }

    else
                                                        {

    update_no(pos);
                                                        }



    break;
                                                        }
                                                case 3: {


    int pos=search();

    printf("%d", pos);
                                                        if
(pos==-1)
                                                        {

    cout<<"Details not found\n";
                                                        }

    else
                                                        {
```

```
    checkout(pos);
                                                        }


    outstat1=1;

    break;
                                                        }



                                            }
                                        }while(che!=4);
                                        break;

                              case 3:     {



                                        exit(0);
                                        break;


                                        }
                              case 4: {
                                        out=1;
                                        break;
                                    }


                              default: cout<<"Wrong choice";
break;
                          }
```

```
                    }while(chx!=4);

                    break;
                    }


            case 2: {
                    system("cls");
                    int chx;

                    do
                    {
                        chx=staff_menu();
                        switch(chx)
                        {
                            case 1: system("cls");
                                    cout<<"\nManager
Panel....... ";

                                    int chp,mpin;
                                    cout<<"\nEnter
password";

                                    cin>>mpin;
                                    if(mpin==mpass)
                                    {

                                    do
                                    {

    chp=manager_menu();
```

```
                                        switch(chp)
                                        {
                                            case 1: {

    cout<<"\nAvailabilty of Room";

    cout<<"\n\nOne-Bedded
Room:"<<oneroom+oneroomo<<"\ntwo-Bedded
Room:"<<tworoom+tworoomo<<"\nThree-Bedded
Room:"<<threeroom+threeroomo<<"\nThree-Bedded Beach View
Room:"<<threebroom+threebroomo;
                                                    break;
                                                    }
                                            case 2: {
                                                    int z;

    cout<<"\nEnter room type to be changed";

    cout<<"\n1.One Bedded\n2.Two Bedded\n3.Three
Bedded\n4.Three bedded beach view\n";

    scanf("%d",&z);

    chnge_room_No(z);
                                                    break;
                                                    }
                                            case 3: {
                                                    int q;

    cout<<"\nEnter room type whose price is to be
changed";
```

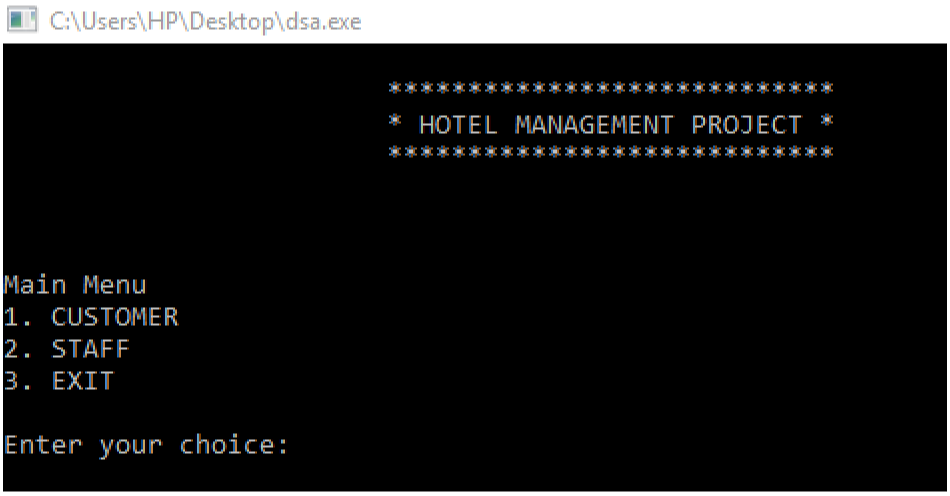
```
    cout<<"\n1.One Bedded\n2.Two Bedded\n3.Three
Bedded\n4.Three bedded beach view\n";

    cin>>q;

    change_roomPrice(q);
                                                    break;
                                                }
                                        }

                                    }while(chp!=4);


                                    break;
                                }
                                else
                                {cout<<"\n wrong pass!!";
                                break;
                                                   }

                            case 2: system("cls");
                                    cout<<"\nReceptionist
Panel....... ";

                                    int chn,rpin;
                                    cout<<"\nEnter
password";

                                    cin>>rpin;
                                    if(rpin==rpass)
                                    {
                                        do
                                        {

    chn=receptionist_menu();
                                            switch(chn)
                                            {
                                                case 1: {

    cout<<"\nAvailabilty of Room";

    cout<<"\n\nOne-Bedded
Room:"<<oneroom+oneroomo<<"\ntwo-Bedded
Room:"<<tworoom+tworoomo<<"\nThree-Bedded
Room:"<<threeroom+threeroomo<<"\nThree-Bedded Beach View
Room:"<<threebroom+threebroomo;
                                                        break;
                                                        }
                                                case 2: {

    system("cls");

    cout<<"Update......";

    int pos=search();
                                                            if
(pos==-1)
                                                            {

    cout<<"Details not found\n";
                                                            }
    else
                                                                       {

    update_no(pos);
                                                                       }



    break;
                                                                  }



                                                    }

                                               }while(chn!=4);



                                               break;
                                          }
                                          else
                                          {
                                               cout<<"\n Wrong
pass!!";
                                               break;
                                          }

                                     case 3:     exit(0);
```

```
                            }
                        }while(chx!=4);

                        break;
                        }
                case 3: exit(0);
                default: cout<<"Wrong choice";
            }
        }
        return 0;
}
```

OUTPUT/RESULTS FOR HOTEL MANAGEMENT SYSTEM

1. Main menu:-

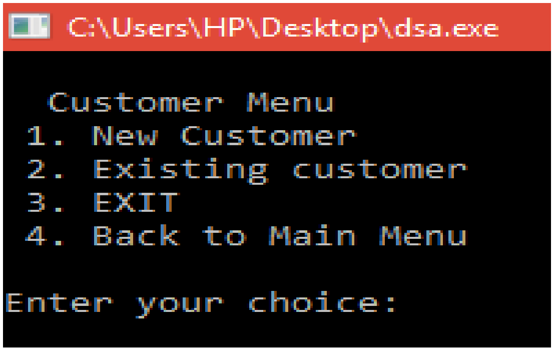
Main menu is the output of the function menu() which is called first in the main function. So, as the program is compiled, we see the main menu. In the main menu if user selects 1, he will be directed to the customer menu, to the staff menu if he chooses 2 and the program exits, if he chooses 3.


C:\Users\HP\Desktop\dsa.exe

```
                         ******************************
                         * HOTEL MANAGEMENT PROJECT *
                         ******************************



Main Menu
1. CUSTOMER
2. STAFF
3. EXIT

Enter your choice:
```

2. Customer menu:-

In the customer menu, user can be a new customer and check in or an existing customer and can update his details (mobile no.) or check out.

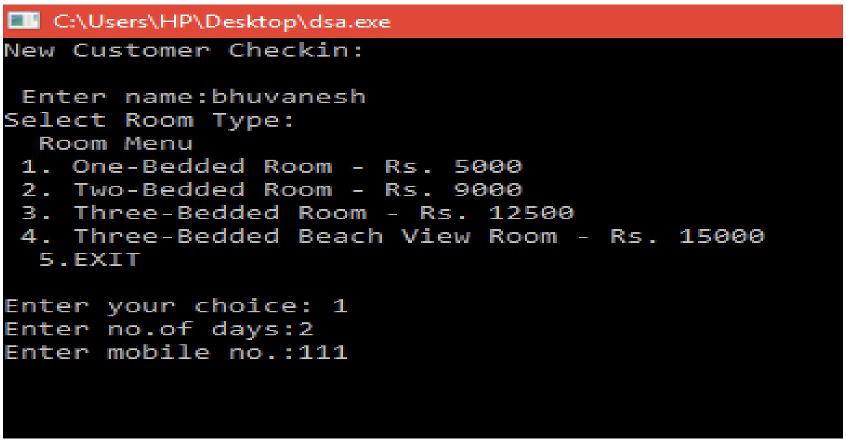


C:\Users\HP\Desktop\dsa.exe

```
  Customer Menu
 1. New Customer
 2. Existing customer
 3. EXIT
 4. Back to Main Menu

Enter your choice:
```

3. **CHECK IN:-**

   Check in function is actually insertion of a new node to the linked list at the front. This new node is actually an object of the structure defined in the program. Each node consists of three details-room type, room number ,number of days and mobile number of the customer**.**

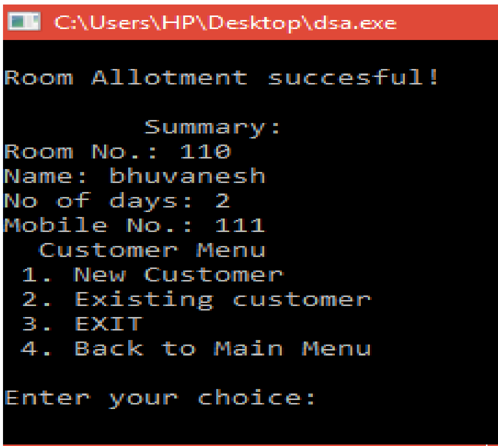


C:\Users\HP\Desktop\dsa.exe

```
New Customer Checkin:

 Enter name:bhuvanesh
Select Room Type:
  Room Menu
 1. One-Bedded Room - Rs. 5000
 2. Two-Bedded Room - Rs. 9000
 3. Three-Bedded Room - Rs. 12500
 4. Three-Bedded Beach View Room - Rs. 15000
  5.EXIT

Enter your choice: 1
Enter no.of days:2
Enter mobile no.:111
```

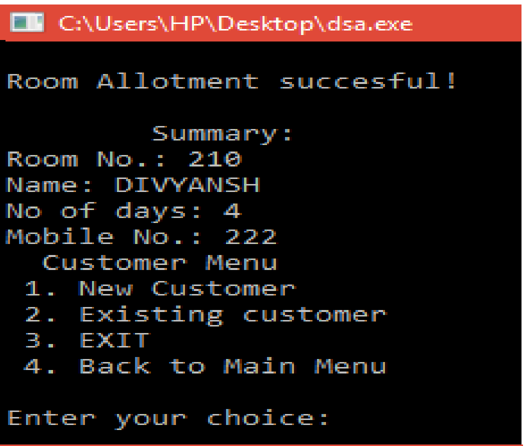
4. **ALLOTMENT :-**

   After the completion of check in, the newly created node is printed to show the summary of the current check in.



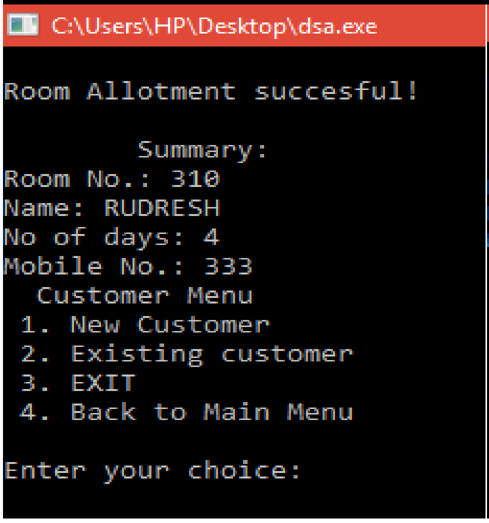
C:\Users\HP\Desktop\dsa.exe

```
Room Allotment succesful!

        Summary:
Room No.: 110
Name: bhuvanesh
No of days: 2
Mobile No.: 111
  Customer Menu
 1. New Customer
 2. Existing customer
 3. EXIT
 4. Back to Main Menu

Enter your choice:
```



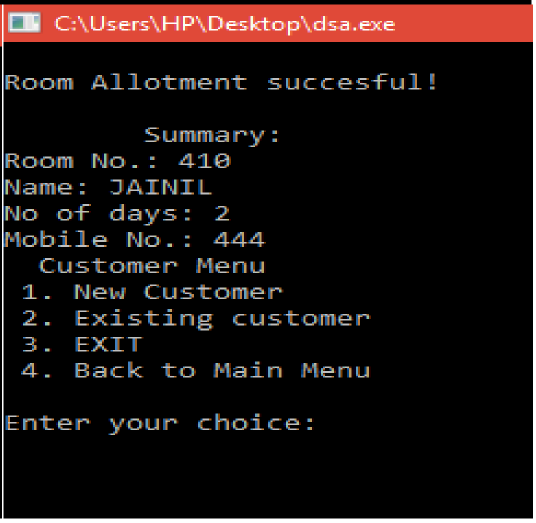
C:\Users\HP\Desktop\dsa.exe

```
Room Allotment succesful!

        Summary:
Room No.: 210
Name: DIVYANSH
No of days: 4
Mobile No.: 222
  Customer Menu
 1. New Customer
 2. Existing customer
 3. EXIT
 4. Back to Main Menu

Enter your choice:
```



C:\Users\HP\Desktop\dsa.exe

```
Room Allotment succesful!

        Summary:
Room No.: 310
Name: RUDRESH
No of days: 4
Mobile No.: 333
  Customer Menu
 1. New Customer
 2. Existing customer
 3. EXIT
 4. Back to Main Menu

Enter your choice:
```



C:\Users\HP\Desktop\dsa.exe

```
Room Allotment succesful!

        Summary:
Room No.: 410
Name: JAINIL
No of days: 2
Mobile No.: 444
  Customer Menu
 1. New Customer
 2. Existing customer
 3. EXIT
 4. Back to Main Menu

Enter your choice:
```

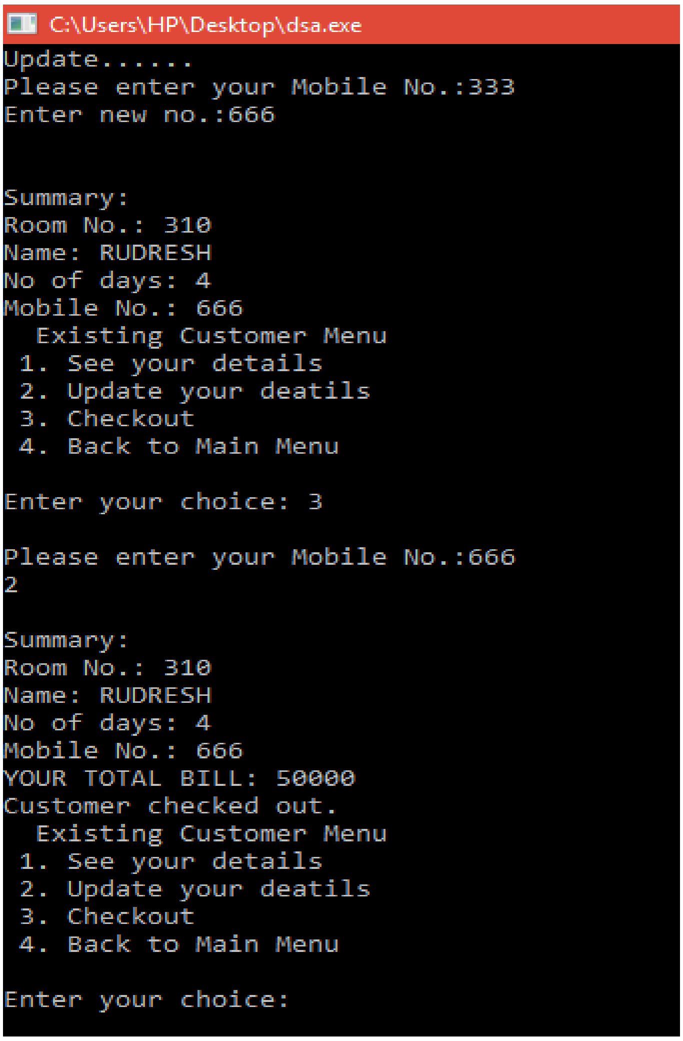
5. UPDATION AND CHECKOUT:-

The updation process is done in two steps:-

1. Traversal to the required node by searching the mobile no. to identify the node in the list
2. Updating the details of the node and printing the details of the node.

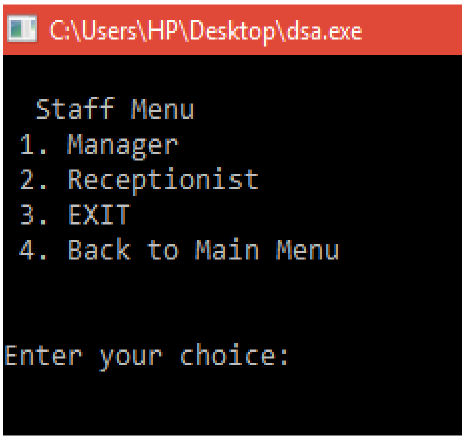
The checkout function is actually deletion of the node:-

1. First we traverse to the required node using mobile No.
2. Deletion of the node from the list
3. Generation of the bill by using the value of number of days in the node to be deleted and calculating bill.



```
Update......
Please enter your Mobile No.:333
Enter new no.:666


Summary:
Room No.: 310
Name: RUDRESH
No of days: 4
Mobile No.: 666
  Existing Customer Menu
 1. See your details
 2. Update your deatils
 3. Checkout
 4. Back to Main Menu

Enter your choice: 3

Please enter your Mobile No.:666
2

Summary:
Room No.: 310
Name: RUDRESH
No of days: 4
Mobile No.: 666
YOUR TOTAL BILL: 50000
Customer checked out.
  Existing Customer Menu
 1. See your details
 2. Update your deatils
 3. Checkout
 4. Back to Main Menu

Enter your choice:
```

6. **STAFF MENU:-**
   In the staff menu, choosing 1 will switch to manager menu and choosing 2 will switch to receptionist menu.

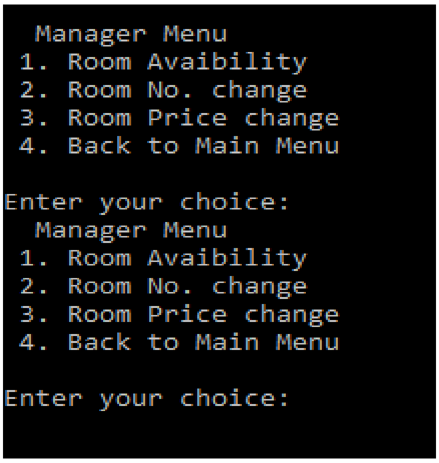


```
C:\Users\HP\Desktop\dsa.exe

  Staff Menu
 1. Manager
 2. Receptionist
 3. EXIT
 4. Back to Main Menu


Enter your choice:
```

7. **MANAGER MENU:-**

   In the manager menu, we can:-

   1. Check room availability- as a node is inserted, the roomtype of the node is read and the number of rooms of that room type is reduced by 1.On checkout, the number of rooms are increased according to the type of room checked out.
   2. Change the number of rooms-The number of rooms of a particular type can be changed by the manager.For this, we store room price as variable which can be changed by the manager and the newly updated number of rooms can be alloted using linked lists.
   3. Manager can also change the room price. Once the room price is changed,the bill of the customer is generated accordingly.



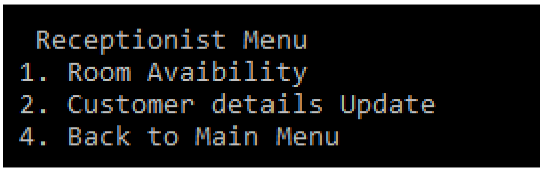
```
  Manager Menu
 1. Room Avaibility
 2. Room No. change
 3. Room Price change
 4. Back to Main Menu

Enter your choice:
  Manager Menu
 1. Room Avaibility
 2. Room No. change
 3. Room Price change
 4. Back to Main Menu

Enter your choice:
```

8.RECEPTIONIST MENU:-

In the receptionist menu, we can:-

1. Check room availability(also discussed under manager menu)

2. Update the details of the customers(also discussed under existing customers menu).



```
 Receptionist Menu
1. Room Avaibility
2. Customer details Update
4. Back to Main Menu
```

CONCLUSION

Computer software or just software is a general term used to describe a collection of computer programs, procedures and documentation that perform some tasks on a computer system. Software and applications are designed to enhance of our life. The basic purpose of shifting from a manual system to a computerised software system is to decrease manual labour, increase accuracy and to implement better security. Keeping these in mind we have, worked towards developing software or a program which enhances the functioning of a Hotel by supplementing software help. Our program will provide clarity, user-friendliness and accuracy to the users of this program.

Hence, our aim was not just to make a program as a part of our learning process but to also create something for the benefit of society as a whole and to do our part as much as we can.

REFERENCES

1. www.wikipedia.org
2. www.google.com
3. www.tutorialspoint.com
4. “INTRODUCTION TO ALGORITHMS BY THOMAS H. CORMEN-3RD EDITION”
5. “LET US C BY YASHWANT KANETKAR-13TH EDITION”
6. Learning C++ by SUMITA ARORA