



Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Summer, Year: 2022), B.Sc. in CSE (Day)

Course Title: Data Structure Lab
Course Code: CSE 106 Section: 213DD

Lab Project Name: Hotel Management System

Student Details

	Name	ID
1.	Mostofa Rafid	213902097

Submission Date: 11/09/2022

Course Teacher's Name: Md. Sultanul Islam Ovi

[For Teachers use only: **Don't Write Anything inside this box]**

Lab Project Status

Marks:

Signature:

Comments:

Date:

Table of Contents

Chapter 1	3
Introduction.....	3
1.1 Introduction.....	3
1.2 Design Goals/Objective	4
Chapter 2	5
Implementation of the Project	5
2.1 Algorithms	5
2.2 Code	5
2.3 Screenshots	14
Chapter 3	18
Conclusion	18
3.1 Learning Outcome	18
3.2 Future Scope	18

Chapter 1

Introduction

1.1 Introduction

My project title is Hotel Management System. I have tried my best to make the complicated process of Hotel Management System as simple as possible using Data Structure & Modular technique & Menu oriented interface. I have tried to design the Project in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error. This project can be used by Administrator. Administrator can maintain daily updates in the hotel records.

The main aim of the entire activity is to automate the process of day-to-day activities of Hotel like Room activities, Admission of a New Customer, assign a room according to customer's demand and compute the bill etc.

In this project, only main activities can be performed of a Hotel Management System, but utmost care has been taken to make the system efficient and user friendly.

“HOTEL Management System” has been designed to computerized the following functions that are performed by the system:

- Admission of New customer
- Search any room's information
- Modification to room details
- Statement of Customer Details

1.2 Design Goals/Objective

A computer-based management system is designed to handle all the primary information required to calculate monthly statements. This project intends to introduce more user friendliness in the various activities such as record update, maintenance, and searching.

The searching of record has been made quite simple with binary search so we can find information faster. As all the details of the customer can be obtained by simply keying the room number of that customer. Similarly, record maintenance and update can also be accomplished by using the identification of the customer with all the details being automatically generated.

The main purpose of my project is performing each Employee's activity in computerized way rather than manually which is time consuming.

Activities of Hotels like:

- Room activities,
- Admission of a New Customer,
- Assign a room according to customer's demand,
- Compute the bill etc.
- List of Regular customers.

Chapter 2

Implementation of the Project

2.1 Algorithms

- Input customer's data using linked list
- Bubble sort
- Binary search
- View all data
- Update data
- Delete data

2.2 Code

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<conio.h>

struct hoteln
{
    int room;
    char name[50];
    char phone[20];
    int days;
    int bill;
    struct hoteln *next;
}* head;

struct hoteln* midl(struct hoteln* first, struct hoteln* last)
{
    if(first==NULL)
        return NULL;
```

```

struct hoteln* slow=first;
struct hoteln* fast=first->next;

while(fast!=last)
{
    fast=fast->next;
    if(fast!=last)
    {
        slow=slow->next;
        fast=fast->next;
    }
}
return slow;
}
struct hoteln* binary(struct hoteln *head, int room)
{
    struct hoteln* first=head;
    struct hoteln* last=NULL;

    do
    {
        struct hoteln* mid=midl(first, last);

        if(mid==NULL)
            return NULL;

        if(mid->room==room)
            return mid;

        else if(mid->room<room)
            first=mid->next;
    }
}

```

```

        else
            last = mid;
    }
    while (last==NULL||last!=first);
    return NULL;
}

void insert(int room, char* name, char* phone, int days, int bill)
{
    struct hoteln * hotel=(struct hoteln *) malloc(sizeof(struct hoteln));
    hotel->room=room;
    strcpy(hotel->name, name);
    strcpy(hotel->phone, phone);
    hotel->days=days;
    hotel->bill=bill;
    hotel->next=NULL;

    if(head==NULL)
    {
        head=hotel;
    }
    else
    {
        hotel->next=head;
        head=hotel;
    }
}

void search(int room)
{
    struct hoteln *temp=head;

    if(binary(head,room)==NULL)

```

```

{
    printf("\t\t\t\t\tNo information found!!!\n");
}
else
{
    printf("\t\t\t\t\tRoom Number: %d\n", temp->room);
    printf("\t\t\t\t\tName: %s\n", temp->name);
    printf("\t\t\t\t\tPhone: %s\n", temp->phone);
    printf("\t\t\t\t\tDays: %d\n", temp->days);
    printf("\t\t\t\t\tBill($): %d\n\n", temp->bill);
    return;
}
}

int display()
{
    int count=0;
    struct hoteln * temp = head;
    while(temp!=NULL)
    {
        printf("\t\t\t\t\tRoom Number: %d\n", temp->room);
        printf("\t\t\t\t\tName: %s\n", temp->name);
        printf("\t\t\t\t\tPhone: %s\n", temp->phone);
        printf("\t\t\t\t\tDays: %d\n", temp->days);
        printf("\t\t\t\t\tBill($): %d\n\n", temp->bill);
        temp = temp->next;
        count++;
    }
    return count;
}

void swap(struct hoteln *a, struct hoteln *b)
{
    int temp=a->room;

```



```

a->room=b->room;
b->room=temp;
}
void sorting(int count)
{
    int swapped, i;
    struct hoteln *ptr1;
    struct hoteln *lptr = NULL;
    if (head==NULL)
        return;
    do
    {
        swapped = 0;
        ptr1 = head;
        while(ptr1->next!=lptr)
        {
            if(ptr1->room>ptr1->next->room)
            {
                swap(ptr1, ptr1->next);
                swapped=1;
            }
            ptr1=ptr1->next;
        }
        lptr=ptr1;
    }
    while(swapped);
    display();
}

void update(int room)
{
    struct hoteln *temp=head;
    while(temp!=NULL)

```

```

{
    if(temp->room==room)
    {
        printf("\n\t\t\t\t\tEnter Days: ");
        scanf("%d", &temp->days);
        printf("\n\t\t\t\t\tEnter bill($): ");
        scanf("%d", &temp->bill);
        printf("\n\t\t\t\t\tUpdate Successful!!!\n");
        return;
    }
    temp=temp->next;
}
printf("\n\t\t\t\t\tNo information found!!!\n");
}

void delete(int room)
{
    struct hoteln *temp1=head;
    struct hoteln *temp2=head;
    while(temp1!=NULL)
    {
        if(temp1->room==room)
        {
            if(temp1==temp2)
            {
                head=head->next;
                free(temp1);
            }
            else
            {
                temp2->next=temp1->next;
                free(temp1);
            }
        }
        printf("\n\t\t\t\t\tDeletion successful!!!\n");
    }
}

```

```

        return;
    }
    temp2=temp1;
    temp1=temp1->next;
}
printf("\n\t\t\t\t\tNo information found!!!\n");
}

int main()
{
    head=NULL;
    int ch, kop, room, days, bill;
    char name[50], phone[20];
    system("cls");
    while(1)
    {
        system("cls");
        printf("\t\t-----\n");
        printf("\t\t***** HOTEL MANAGEMNET SYSTEM
*****\n");

        printf("\t\t-----\n");
        printf("\n\t\t\t\t\t1. Book a room");
        printf("\n\t\t\t\t\t2. Search");
        printf("\n\t\t\t\t\t3. View");
        printf("\n\t\t\t\t\t4. Update");
        printf("\n\t\t\t\t\t5. Delete");
        printf("\n\t\t\t\t\t6. exit");
        printf("\n\n\t\t\t\t\tEnter choice(1-6): ");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1:
                system("cls");

```

```
printf("\n\t\t\t\t\tBOOKING");  
printf("\n\t\t\t\t\tt-----\n");  
printf("\n\t\t\t\t\tEnter room number: ");  
scanf("%d", &room);  
printf("\t\t\t\t\tEnter name: ");  
fflush(stdin);  
gets(name);  
printf("\t\t\t\t\tEnter mobile number: ");  
fflush(stdin);  
gets(phone);  
printf("\t\t\t\t\tEnter Days: ");  
scanf("%d", &days);  
printf("\t\t\t\t\tEnter bill($): ");  
scanf("%d", &bill);  
insert(room, name, phone, days, bill);  
  
break;  
case 2:  
system("cls");  
printf("\n\t\t\t\t\tSEARCH");  
printf("\n\t\t\t\t\tt-----\n");  
printf("\n\t\t\t\t\tEnter the room number: ");  
scanf("%d", &room);  
search(room);  
  
break;  
case 3:  
system("cls");  
printf("\n\t\t\t\t\tVIEW");  
printf("\n\t\t\t\t\tt-----\n");  
sorting(kop);  
  
break;  
case 4:
```

```
printf("\n\t\t\t\t\t-----\n");  
printf("\n\t\t\t\t\t\t\tEnter the room number: ");  
scanf("%d", &room);  
update(room);  
  
break;  
  
case 5:  
  
system("cls");  
printf("\n\t\t\t\t\t\t\tDELETE");  
printf("\n\t\t\t\t\t\t\t-----\n");  
printf("\n\t\t\t\t\t\t\tEnter the room number: ");  
scanf("%d", &room);  
  
delete(room);  
  
break;  
  
case 6:  
  
exit(1);  
  
}  
  
printf("\n\n\t\t\t\t\tContinue to Again!!!");  
getch();  
  
}
```

2.3 Screenshots

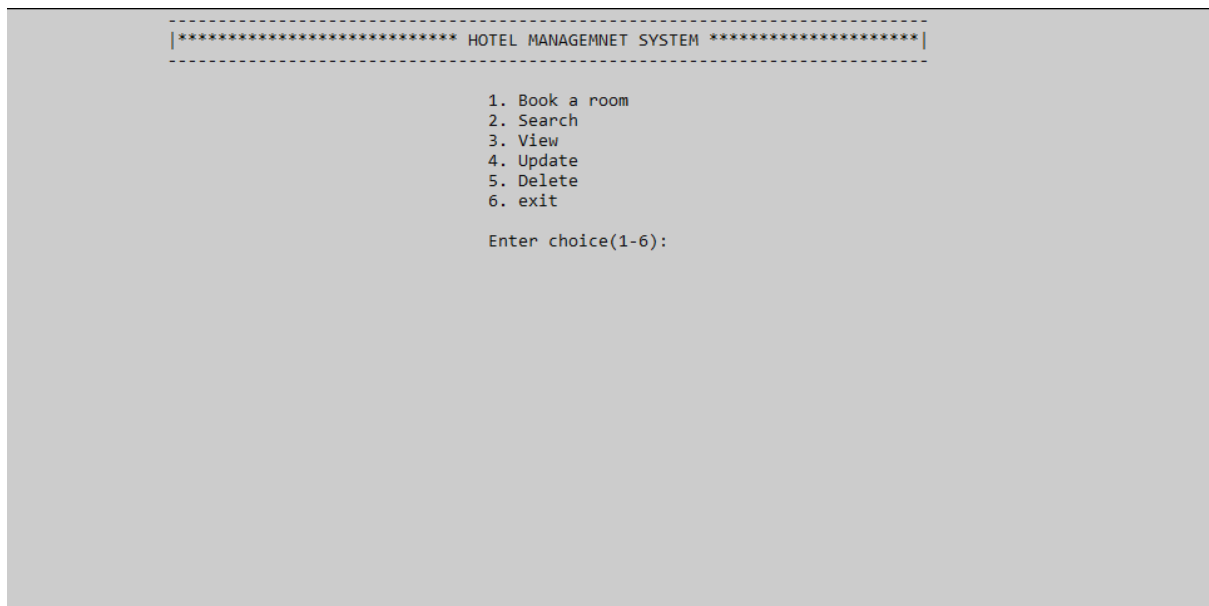


Figure 1: Main Menu interface



Figure 2: Book a Room

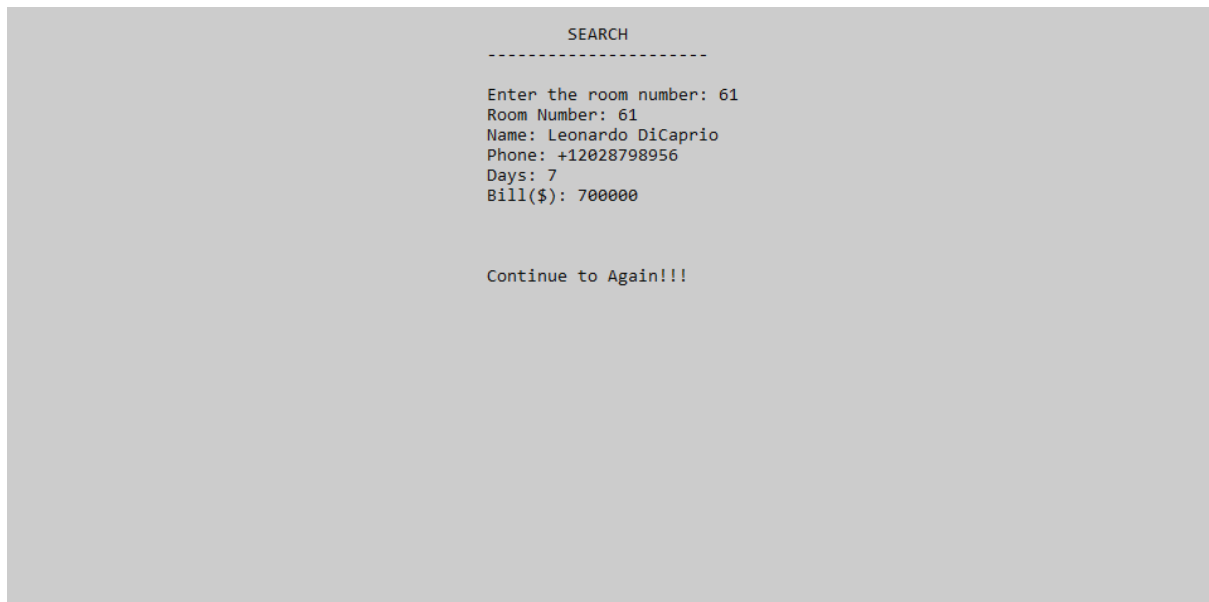


Figure 3: Search

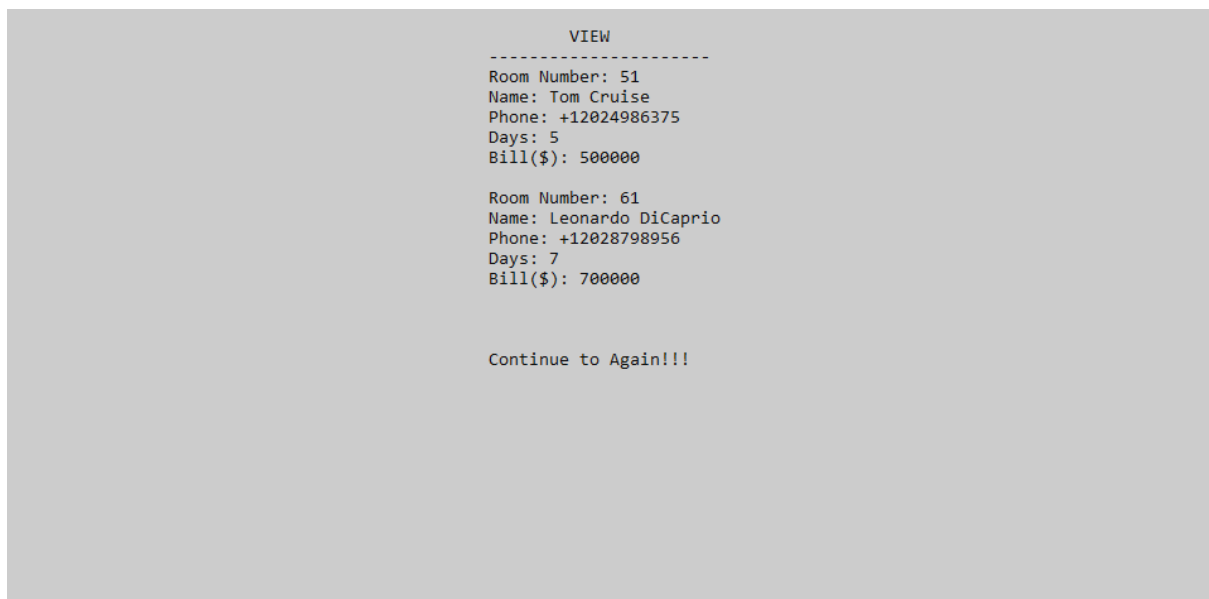


Figure 4: View

```

                                UPDATE
                                -----
Enter the room number: 51
Enter Days: 8
Enter bill($): 800000
Update Successful!!!

Continue to Again!!!_

```

```

                                VIEW
                                -----
Room Number: 51
Name: Tom Cruise
Phone: +12024986375
Days: 8
Bill($): 800000

Room Number: 61
Name: Leonardo DiCaprio
Phone: +12028798956
Days: 7
Bill($): 700000

Continue to Again!!!_

```

Figure 5: Update


```

                                DELETE
                                -----
Enter the room number: 51
Deletion successful!!!

Continue to Again!!!

```

```

                                VIEW
                                -----
Room Number: 61
Name: Leonardo DiCaprio
Phone: +12028798956
Days: 7
Bill($): 700000

Continue to Again!!!

```

Figure 6: Delete

Chapter 3

Conclusion

3.1 Learning Outcome

I have designed the given proposed system to automate the process of Hotel Management. This project is useful for the authorities which keep track of all the users registered in a particular state. The authority can add hotel room details, availability of rooms, booking etc.

In this project, I have implemented data structure algorithms in C. I added binary search to fetch information which has less time complexity, and bubble sort to see the room information in sorted list.

3.2 Future Scope

- This project can be used in hotel.
- More function can be added to this project if any hotel want.