**HOTEL MANAGEMENT SYSTEM**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

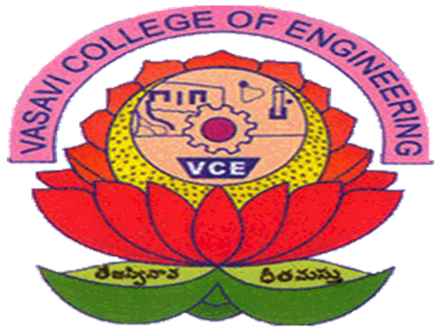
IN

**INFORMATION TECHNOLOGY**

By

**BANDARU NEHA 1602-19-737-144**

**CHOKKARAPU SINDHU 1602-19-737-174**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

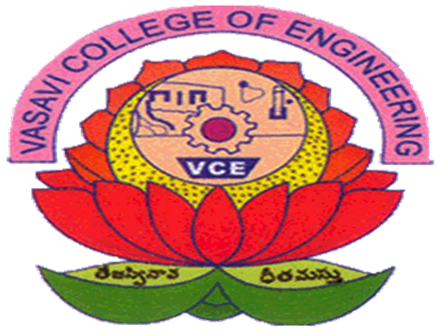
**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We, **BANDARU NEHA** and **SINDHU CHOKKARAPU**, bearing hall ticket numbers, **1602-19-737-144** and **1602-19-737-174** hereby declare that the project report entitled “HOTEL MANAGEMENT SYSTEM” is submitted in partial fulfilment of the requirement for the award of the degree of Bachelor of Engineering in Information Technology

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**NEHA BANDARU**

**1602-19-737-144**

**SINDHU CHOKKARAPU**

**1602-19-737-174**

**(Faculty In-Charge) (Head,Dept of IT)**

**ACKNOWLEDGEMENT**

We would like to thank Ms. Divya Linganeni for giving us an outstanding guidance and co-operation for designing this system. Without her kind patronage and guidance the project would not have taken shape. We would also like to express our gratitude and sincere regards for her kind approval of the project, time to time counselling and advice.

We would also like to thank our HOD for giving this opportunity to build up an application by the use of C Programming.

Next we would also to thank the entire faculty for their full co-operation.

Last but not the least, we would also like to thank our parents for all their support that continues to motivate us to work towards our betterment and created a environment during this pandemic situation where we could work and complete the project successfully.

So, with gratitude we acknowledge all those who guided and encouraged and served a beacon of light and crowned the effect with success.

**ABSTRACT**

The project HOTEL MANAGEMENT SYSTEM aims to build a system for managing the activities in a hotel. As customer and hotel are two related terms of business field, there should be proper way of management. The title of the project is related to hotel i.e., “Hotel Management System” deals with the basic records of customer details, room details, restaurant details, check-in and check-out dates etc., of the hotel, Our project is divided into various modules. It enables the admin to keep the records of the customers. This project is useful in automating the manual activities in hotels like services, booking, check-out, managing records of customers, etc. The manual way of doing these tasks is inefficient and ineffective. One of the objectives of this project is to eliminate paper work and reduce time and also to make the process effective. Only authorized users are allowed to enter through the system. So security is also maintained in the system. Thus, the Hotel Management System being one of the major necessities in today’s business field, this project has scope in hotels since the routine activities of managing departments in hotels become easy.

**CONTENTS**

1. **Introduction…………………………………………. 6**
2. **Technology……………………………………………8**
   1. Software requirements………………………….8
   2. Hardware requirements…………………………8
3. **Proposed work……………………………………….9**
   1. **DESIGN**
      1. Use case diagram and descriptions for all the use cases……………………………………...9
   2. **IMPLEMENTATION**
      1. Module wise code for the entire project…12
      2. Specific code is highlighted in bold…….40
      3. Github links……………………………...47
   3. **TESTING……………………………………………….48**
4. **Results…………………………………………………58**
5. **Additional knowledge gained………………………...74**
6. **Conclusion…………………………………………….75**
7. **Future work………………………………………….75**
8. **References…………………………………………….76**

**1. INTRODUCTION**

In today’s world the way of functioning and managing the system has been totally changed. There is a sudden and abrupt changes in the structure maintenance and modification, handling, levelling inside every system. As we know that, “necessity is the mother of invention”, in today’s challenging world, every system is developed and launched by the use of computer software and programming. Due to increasing hotels in the world, it is very enough to handle important files and data of the hotel efficiently and systematically. This project is designed to help hotel staff and the programming language used is C.

The main objective of our project “HOTEL MANAGEMENT SYSTEM” is to automate the process of day to day activities of hotel like room bookings, computing bills, providing food services to customers, check-in and check-outs. This project will automate the manual and legacy operations. The project will keep track of customer database, room details, check-in and vacating details. It is username and password protected as well. Only the authorized users are allowed to enter through the system. So security is also maintained in the system.

In our project, we provided an easy way of searching the customer in a particular room of a hotel by the provided room number. It also shows all the details of the existing customers in the hotel. Customers are provided with food services and the total bill is generated. It shows the status of rooms whether it is vacant or occupied by the customer. We also included reminder feature to show the check-out details of rooms on any provided date. When the check-out is done, the total bill customer need to pay is displayed. User also have the option to exit from the program.

Developing software on a topic like “Hotel Management System” has much scope. This project creates user friendly environment. It increases the efficiency and saves the time. The program saves the data in the form of binary files. This facilitates data hiding and also reduces loss of information. The entire information has maintained in the database or files and only authorized user can retrieve the necessary information which can be easily accessible from the file. Thus, the hotel management system being one of the major necessity in today’s business field, it can be effective to the hotel management and reduces the work load of the users.

**FEATURES OF THE PROJECT**

* **BOOK A ROOM**
* **VIEW ROOM RECORDS**
* **SEARCH CUSTOMER RECORD**
* **CHECK-OUT A ROOM**
* **FOOD SERVICES**

1. **ORDER**
2. **DISPLAY**
3. **EXIT**

* **REMINDER**
* **EXIT**

**2. TECHNOLOGY**

All computer software needs certain hardware components and also other software resources to be present, in order for computers to be used efficiently. These pre-requisites are known as System Requirements. System Requirements are of two types – Software Requirements and Hardware Requirements.

**SOFTWARE REQUIREMENTS**

Software requirements deal with defining the software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These preconditions are generally not included in the software package and need to be installed separately.

In order to use Hotel Management System, one should have the following software requirements:

* Operating system: windows 7 and above
* C compiler: GNU Compiler Collection (GCC).
* Editor: Any text editor that supports C language

**HARDWARE REQUIREMENTS**

Hardware requirements refer to the common set of requirements defined by any operating system or software application and are usually the physical computer resources. In this we look into the architecture, processing power, memory, secondary memory, display adapter and peripherals.

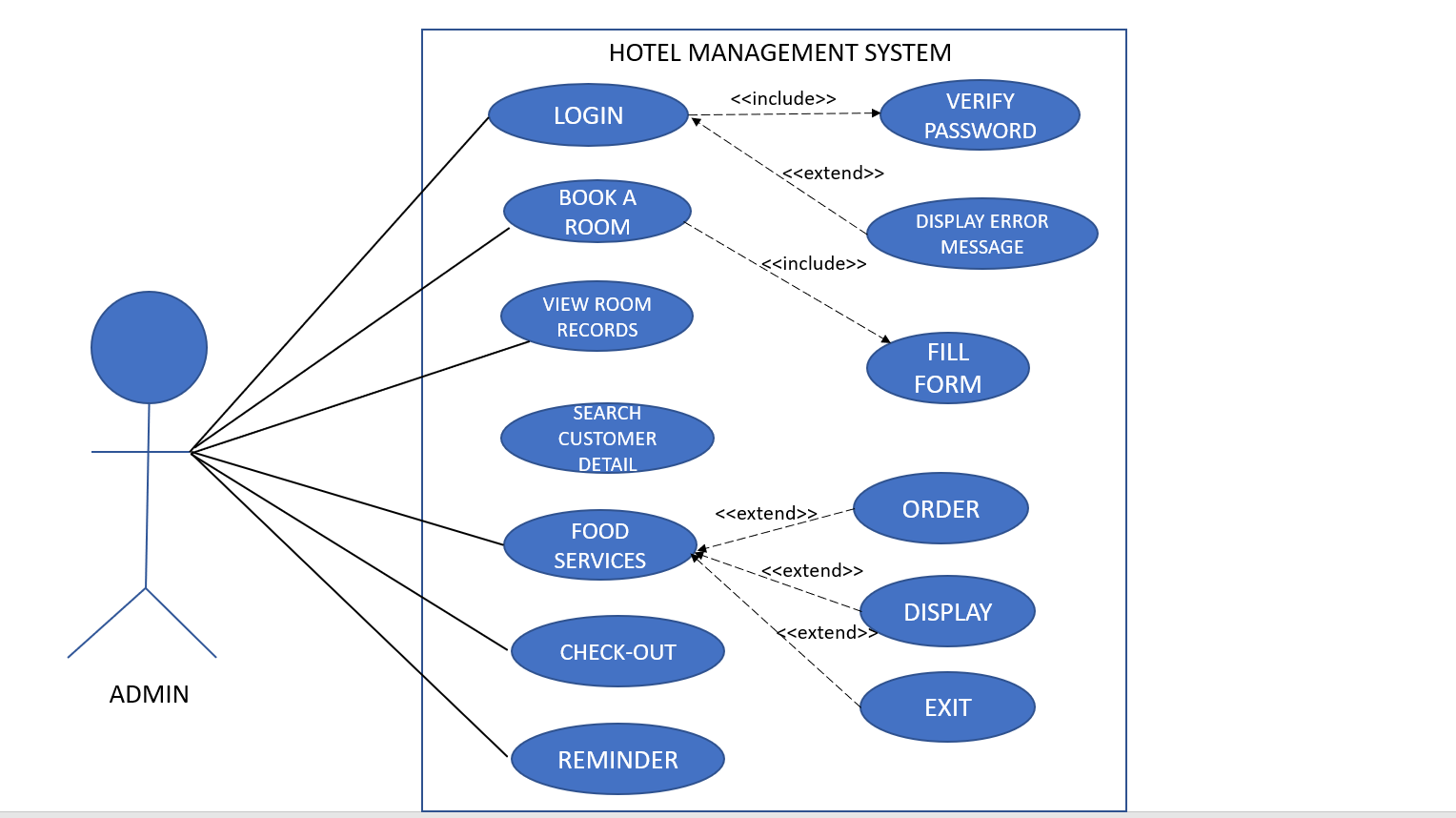
In order to use Hotel Management System, one should have the following hardware requirements:

* Processor: intel core I5 and above
* Memory: 4 GB RAM and above

**3. PROPOSED WORK**

**3.1) Design**

**USE CASE DIAGRAM**



**USE CASE DESCRIPTIONS**

* **LOGIN**

In the login form, the system prompts the user(admin) to enter the username and password. If the entered credentials are correct, the user is directed to the main menu else displays the error message.

* **BOOK A ROOM**

Here, customer will be asked to book a room and if that room is available then checking in procedure will be initiated and ask to fill the form with the necessary details of the customer like name, phone number, occupation, check-in and check-out dates for future references.

* **VIEW ROOM RECORDS**

In this option, it shows all the existing customer details in the hotel rooms.

* **SEARCH CUSTOMER RECORD**

Admin can search for a customer in any provided room number. If the room is booked, customer details are displayed else shows the message that the room is not booked.

* **CHECK-OUT A ROOM**

In this option it will first ask the admin for the room number it wants to vacate and then the customer details will be deleted from the database and the room will be available again for booking. The total bill for the vacated room to be paid by customer is displayed.

* **FOOD SERVICES**

1. **ORDER**

Ask the customer name and displays the food menu. After the food is ordered, total bill is generated**.**

1. **DISPLAY**

In this option, it displays the total number of food orders placed.

1. **EXIT**

It exits from the food services menu and goes back to main menu.

* **REMINDER**

In this option, it prompts the user to enter the date and if any room check-out date matches with the entered date, it displays those room numbers or shows the message that there is no notification.

* **EXIT**

The user can exit from the program and thank you message is displayed on the screen.

* 1. **Implementation**

**3.2.1) Module wise code for the entire project**

/\*

HOTEL MANAGEMENT SYSTEM IN C

SINDHU CHOKKARAPU 1602-19-737-174

NEHA BANDARU 1602-19-737-144

\*/

#include<stdio.h>

#include<string.h>

#include<time.h>

#include<conio.h>

#include<stdlib.h>

int Time() //to print current system time

{

time\_t currentTime;

time(&currentTime);

printf("Time: %s\n",ctime(&currentTime));

}

int book;

struct booking

{

char r\_number[5];

char name[50];

char occupation[100];

char age[5];

char phone\_number[10];

char period[2];

char in\_date[12];

char out\_date[12];

}s;

void login();

void logo();

void r\_booking();

void main\_menu();

void search\_room\_number();

void food\_services();

void items();

int display();

int order();

void view();

void check\_out();

void notify();

void check\_date(int d,int m,int y,struct booking r);

int main()

{

system("cls"); //to clear the screen

Time();

printf("\n\n\n\n\t\t=========================================================================\n\n\n\t\t\t\t\tTAJ HOTEL\n\n\n\t\t\t\t\t1602-19-737-174/144\n\n\n\t\t\t\t\tHotel Management System\n\n\n\t\t==========================================================================\n");

printf("\nPress Enter to continue...");

getch();

login();

}

void login() //login function

{

char u\_name[20],c=' ';

char password[10];

int a=0,i=0;

do

{

system("cls");

logo();

printf("\t\t\t\t\t\t\tLOGIN FORM\n\n");

printf("\n\t\t\t\t\t\t Enter User Name: ");

scanf("%s",&u\_name);

printf("\n\t\t\t\t\t\t Enter Password: ");

while(i<10)

{

password[i] = getch();

c = password[i];

if(c==13) break;

else printf("\*");

i++;

}

password[i] = '\0';

i=0;

if(strcmp(u\_name,"sindhu")==0 && strcmp(password,"2305")==0)

{

printf("\n\n\t\t\t\t\t\t\tWelcome!\n\n\t\t\t\t\t You have successfully logged In!!!\n\n\t\t\t\t\t\t\tHave a nice day...");

printf("\n\n\t\t\t\t\t\tEnter any key for services....");

getch();

main\_menu();

break;

}

else

{

printf("\n\n\t\t\t\t\tWrong Password!!!");

a++;

getch();

}

}while(a<=2);

if (a>2)

{

printf("\nSorry you have entered the wrong username and password for 3 times!!!");

getch();

}

system("cls");

}

void logo() //hotel logo

{

printf("\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t\t\t\t WELCOME TO TAJ HOTEL \n\n");

printf("\t\t\t\t\t SINDHU\_NEHA \n\n");

printf("\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

}

void main\_menu(){ //main menu function

system("cls");

Time();

getch();

int ch;

logo();

printf("\t\t\t\t\t\t1.Book a room\n\n");

printf("\t\t\t\t\t\t2.View Room records\n\n");

printf("\t\t\t\t\t\t3.Search Customer record\n\n");

printf("\t\t\t\t\t\t4.Check-Out a room\n\n");

printf("\t\t\t\t\t\t5.Food Services\n\n");

printf("\t\t\t\t\t\t6.Reminder\n\n");

printf("\t\t\t\t\t\t7.Exit");

printf("\n\n\t\t\t\t\t\t\tEnter Your Choice:");

scanf("%d",&ch);

switch(ch){

case 1:r\_booking();

break;

case 2:view();

break;

case 3:search\_room\_number();

break;

case 4:check\_out();

break;

case 5:food\_services();

break;

case 6:system("cls");

notify();

break;

case 7:

system("cls");

printf("\n\n\t\t\t\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\* THANK YOU \*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n\n\t\t\t\t\t \*\*\*\* VISIT AGAIN \*\*\*\*\t");

printf("\n\n\n");

exit(0);

break;

default:

system("cls");

printf("Incorrect Input");

printf("\n Press any key to go to main menu");

getch();

main\_menu();

}

}

void food\_services() //FOOD SERVICE

{

system("cls" );

Time();

getch();

logo();

int ser\_choice;

printf("\n\n\t\t\t\t\t\t 1. Order\n\n");

printf("\t\t\t\t\t\t 2.Display\n\n");

printf("\t\t\t\t\t\t 3 to Exit");

printf("\n\n\t\t\t\t\t\t Enter Your Choice:");

scanf("%d",&ser\_choice);

switch(ser\_choice)

{

case 1:order();

break;

case 2:display();

break;

case 3:main\_menu();

break;

}

}

struct order{

char name[50];

int number\_of\_food;

int cost;

}p[10000];

void items () //FOOD MENU

{

printf("\n\t\t\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("Item Code#\t\t Description\t\t\t\t\t\t Size \t\t\t\t\t Price(Rs)\n");

for(int h=0;h<145;h++)

printf("\*");

printf("\n[21]\t\t\t Burger \t\t\t\t\t\t 200/300gm\t\t\t\t\t 130/180\n");

printf("\n[22]\t\t\t Pizza \t\t\t\t\t\t 2/14/16inch\t\t\t\t 750/975/1250\n");

printf("\n[23]\t\t\t Fried Chicken \t\t\t\t\t 1/4/10 pcs\t\t\t\t 85/320/790\n");

printf("\n[24]\t\t\t Grilled Chicken \t\t\t\t\t quarter/half/full\t\t\t\t 70/140/250\n");

printf("\n[25]\t\t\t Noodles \t\t\t\t\t\t Half/Full \t\t\t\t 130/250\n");

printf("\n[26]\t\t\t Pasta \t\t\t\t\t\t Half/Full \t\t\t\t\t 120/240\n");

printf("\n[27]\t\t\t Oreo Shake \t\t\t\t\t\t 250mg \t\t\t\t 180\n");

printf("\n[28]\t\t\t Cold Coffee \t\t\t\t\t\t 250mg \t\t\t\t\t 65\n");

printf("\n[29]\t\t\t Cappuccino \t\t\t\t\t\t 250mg \t\t\t 70\n");

printf("\n[30]\t\t\t Coke \t\t\t\t\t \t 250/500mg \t\t\t\t 20/30\n\n");

}

int order\_count = 0;

int order () //ORDER FOOD

{

int n, in, qu;

char t;

printf ("please enter your name: ");

scanf("%c",&t); // statement to clear buffer

scanf("%[^\n]",p[order\_count].name);

items();

p[order\_count].cost = 0;

p[order\_count].number\_of\_food = 0;

level:

printf ("How many items you want to order? ");

scanf ("%d", &n);

getchar ();

while(n--){

printf ("\nEnter chosen item code\n");

scanf ("%d", &in);

getchar();

switch(in){

case 21 : {

printf ("Enter the size of burger: 1. 200gm 2. 300gm\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (130\*qu);

else p[order\_count].cost += (180\*qu);

break;

}

case 22 :{

printf ("Enter the size of Chicken Pizza: 1. 12inch 2. 14inch 3. 16inch\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (750\*qu);

else if (i==2) p[order\_count].cost += (975\*qu);

else p[order\_count].cost += (1250\*qu);

break;

}

case 23 :{

printf ("Enter the amount of chicken: 1. 2 pcs 2. 4 pcs 3. 10 pcs\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (85\*qu);

else if (i==2) p[order\_count].cost += (320\*qu);

else p[order\_count].cost += (790\*qu);

break;

}

case 24 :{

printf ("Enter the size of grilled chicken: 1. quarter 2. half 3. full\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (70\*qu);

else if (i==2) p[order\_count].cost += (140\*qu);

else p[order\_count].cost += (250\*qu);

break;

}

case 25 :{

printf ("Enter the size(bowl) of Noodles: 1. half 2. full\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (130\*qu);

else p[order\_count].cost += (250\*qu);

break;

}

case 26 :{

printf ("Enter the size(bowl) of pasta: 1. half 2. full\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (120\*qu);

else p[order\_count].cost += (240\*qu);

break;

}

case 27 : {

printf ("250 mg of Oreo Shake\n");

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

p[order\_count].cost += (180\*qu);

break;

}

case 28 : {

printf ("250 mg of Cold Coffee\n");

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

p[order\_count].cost += (65\*qu);

break;

}

case 29 : {

printf ("250 mg of Cappuccino\n");

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

p[order\_count].cost += (70\*qu);

break;

}

case 30 :{

printf ("Enter the size of Coke: 1. 250mg 2. 500mg\n");

int i;

scanf ("%d", &i);

getchar();

printf ("Please enter the quantity: ");

scanf ("%d", &qu);

getchar();

p[order\_count].number\_of\_food +=qu;

if (i==1) p[order\_count].cost += (20\*qu);

else p[order\_count].cost += (30\*qu);

break;

}

default : {

printf ("Invalid selection try again\n");

n++;

}

}

}

char temp;

printf ("Do you want to order anything else?(y/n)\n ");

scanf("%c", &temp);

if (temp == 'y' || temp == 'Y')

goto level;

if(temp=='n' || temp=='N')

{

printf ("\nNumber of food ordered %d\n", p[order\_count].number\_of\_food);

printf ("\nThank you %s. Your bill is %dRs.\nPlease wait while we prepare the food.\n\n", p[order\_count].name, p[order\_count].cost);

order\_count ++;

}

getch();

food\_services();

}

int display () //display orders function

{

printf ("Total order taken: %d\n", (order\_count));

getch();

food\_services();

}

void r\_booking() //ROOM BOOKING

{

system("cls" );

Time();

//getch();

logo();

printf("\n\t\t\t\t\t\t\tOur Hotel Rooms\n\n");

printf("\n\n\tRoom No\t\tCategory\t\tAvailable\t\tPeriod(x days)\t Arrival Date\t Exit Date\n\n");//Checking available or not

FILE \*fp;

char day[5];

char avail[10];

char in\_date[12],out\_date[12];

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"101")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t101\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"102")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t102\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"103")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t103\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"104")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t104\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"105")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t105\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"106")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t106\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"107")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t107\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"108")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t108\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"109")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t109\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

fp=fopen("detail.txt","r");

strcpy(day,"N/A");

strcpy(in\_date,"N/A");

strcpy(out\_date,"N/A");

strcpy(avail,"Yes");

while(fread(&s,sizeof(s),1,fp))

{

if(strcmp(s.r\_number,"110")==0)

{

strcpy(avail,"No");

strcpy(day,s.period);

strcpy(in\_date,s.in\_date);

strcpy(out\_date,s.out\_date);

}

}

fclose(fp);

printf("\n\t110\t\tDouble(1B)\t\t%s\t\t\t%s Days\t\t%s\t%s\n",avail,day,in\_date,out\_date);

printf("\n\n\t\t\t\t\t\tWhich room you want to book:");

scanf("%d",&book);

if(book>=101 && book<=110)

{

system("cls" );

Time();

getch();

logo();

time\_t t;

t = time(NULL);

struct tm tm = \*localtime(&t);

int d = tm.tm\_mday;

int m = tm.tm\_mon+1;

int y = tm.tm\_year+1900;

char date[10];

char temp;

printf("\n\t\t\t\t\t\tFill the form\n\n");

printf("\t\t\t\t");

for(int l=0;l<60;l++)

printf("\*");

fp=fopen("detail.txt","a");

struct booking s;

printf("\n\n\t\t\t\t\t\tReenter Room Number: ");

scanf("%s",&s.r\_number);

printf("\n\t\t\t\t\t\tEnter Name: ");

scanf("%c",&temp); // temp statement to clear buffer

scanf("%[^\n]",s.name);

printf("\n\t\t\t\t\t\tEnter Age: ");

scanf("%s",&s.age);

printf("\n\t\t\t\t\t\tEnter Phone Number: ");

scanf("%s",&s.phone\_number);

printf("\n\t\t\t\t\t\tEnter Occupation: ");

scanf("%s",&s.occupation);

printf("\n\t\t\t\t\t\tEnter Period(\'x\'days): ");

scanf("%s",&s.period);

printf("\n\t\t\t\t\t\tEnter Arrival Date(DD-MM-YY): ");

scanf("%s",&s.in\_date);

check\_date(d,m,y,s);

printf("\n\t\t\t\t\t\tEnter Exit Date(DD-MM-YY): ");

scanf("%s",&s.out\_date);

printf("\n\n\n\t\t\t\t\t\t This room has been booked for %s days",s.period);

fwrite(&s,sizeof(s),1,fp);

getch();

fclose(fp);

main\_menu();

}

else

{

printf("\n\t\t\t\t\tInvalid room no...");

getch();

main\_menu();

}

}

// 3.2.2) Specific logic

**void check\_date(int d,int m,int y,struct booking r)**

**//VALIDATE DATE FUNCTION (Checks whether the date is valid now or completed)**

**{**

**char da[12],mo[12],ye[12],res[12];**

**strcpy(da,strtok(r.in\_date,"-"));**

**strcpy(mo,strtok(NULL,"-"));**

**strcpy(ye,strtok(NULL,"-"));**

**int a = atoi(da);**

**int b = atoi(mo);**

**int c = atoi(ye);**

**if(c>y)**

**{**

**printf("");**

**}**

**else if(c==y)**

**{**

**if(b>m)**

**printf("");**

**else if(b==m)**

**{**

**if(a>=d)**

**printf("");**

**else**

**{**

**printf("\n\n\t\t\t\t\t\t\t\tEnter valid date!!!");**

**getch();**

**r\_booking();**

**}**

**}**

**else**

**{**

**printf("\n\n\t\t\t\t\t\t\t\tEnter valid date!!!");**

**getch();**

**r\_booking();**

**}**

**}**

**else**

**{**

**printf("\n\n\t\t\t\t\t\t\t\tEnter valid date!!!");**

**getch();**

**r\_booking();**

**}**

**}**

void view() //DISPLAY DETAILS OF CUSTOMERS

{

system("cls");

Time();

getch();

logo();

printf("\t\t\t\t\t\tCustomer Details\n\n");

FILE \*fp;

fp=fopen("detail.txt","r");

struct booking s;

printf("Room No\t Name\t\tAge\t Phone Number \t Occupation\t Period(days)\t Arrival Date\t Exit Date\n");

getch();

while (fread(&s,sizeof(s),1,fp))

{

printf("\n%s\t %s\t\t%s\t %s\t %s\t %s\t\t %s\t %s\n",s.r\_number,s.name,s.age,s.phone\_number,s.occupation,s.period,s.in\_date,s.out\_date);

}

fclose(fp);

getch();

main\_menu();

}

void search\_room\_number() //search customer record function

{

system("cls");

printf("\t\t\t\t\t\tCustomer Search Option\n\n");

printf("\t\t\t\t");

for(int k=0;k<60;k++)

printf("\*");

FILE \*fp;

int flag=1;

fp=fopen("detail.txt","r+");

struct booking s;

char se\_room[20];

if(fp==0)

exit(0);

fflush(stdin);

printf("\n\n\t\t\t\t\t\t Enter Room Number:");

scanf("%s",se\_room);

while(fread(&s,sizeof(s),1,fp))

{

if (strcmp(se\_room,s.r\_number) == 0)

{

flag=0;

printf("\n\n\n\tRoom Number\tName\tAge\tPhone Number\tOccupation\tPeriod(days)\t Arrival Date\t Exit date\n");

printf("\t");

for(int j=0;j<110;j++)

printf("\*");

printf("\n\t%s\t\t%s\t%s\t%s\t%s\t\t%s\t\t %s\t %s\n",s.r\_number,s.name,s.age,s.phone\_number,s.occupation,s.period,s.in\_date,s.out\_date);

break;

getch();

}

}

if(flag==1)

{

printf("\n\tRequested Room number is not booked!!!");

}

getch();

fclose(fp);

main\_menu();

}

void check\_out() //CHECK-OUT FUNCTION

{

int r,p,z,bill;

system("cls");

Time();

getch();

struct booking s;

logo();

FILE \*fpo;

FILE \*fpt;

printf("Enter the room no u want to check-out: ");

scanf("%d",&r);

fpo = fopen("detail.txt","r+");

fpt = fopen("Tempfile","w");

while(fread(&s,sizeof(s),1,fpo))

{

p = atoi(s.r\_number);

z=atoi(s.period);

bill = 1500\*z;

if(p!=r)

fwrite(&s,sizeof(s),1,fpt);

}

fclose(fpo);

fclose(fpt);

fpo= fopen("Detail.txt","w");

fpt = fopen("Tempfile","r");

while(fread(&s,sizeof(s),1,fpt))

fwrite(&s,sizeof(s),1,fpo);

printf("\n\t\tCheckout done...");

printf("\n\t\tRoom no %s bill is %d.\n",s.r\_number,bill);

printf("\n\t\tThe room %s is available again....",s.r\_number);

fclose(fpo);

fclose(fpt);

getch();

main\_menu();

}

**void notify() //Reminder function-shows notifications for check-out of rooms if any...**

**{**

**logo();**

**char today\_date[12];**

**printf("\n\n\n\t\t\t\tEnter Date(DD-MM-YYYY):");**

**fflush(stdin);**

**scanf("%s",today\_date);**

**int count=0;**

**FILE \*fp;**

**fp=fopen("detail.txt","r");**

**struct booking s;**

**while(fread(&s,sizeof(s),1,fp))**

**{**

**if (strcmp(s.out\_date,today\_date)==0)**

**{**

**printf("\n\t\t Room No:%s \tDate:%s\t Last Date Today",s.r\_number,s.out\_date);**

**count++;**

**}**

**}**

**fclose(fp);**

**if(count==0)**

**{**

**printf("\n\t\t\t\tNo Notification for today");**

**}**

**getch();**

**main\_menu();**

**}**

// END OF PROGRAM

// THANK YOU...

**3.2.3) Github links**

[**https://github.com/Neha-2001**](https://github.com/Neha-)

[**https://githum.com/Sindhu-2305**](https://githum.com/Sindhu-2305)

* 1. **TESTING – TEST CASES**
* LOGIN

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID**: TC01 | | | Use case ID  **UC01** |
| **Test case title**: Login(Admin) | | |
| **Test case description**: User has to enter username and password | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The system prompts the user to enter username and password. Password entered is not same as the hard-coded one. | An error message saying “Wrong Password” is displayed on the screen. | An error message saying “Wrong Password” is displayed on the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID**: TC02 | | | Use case ID  **UC01** |
| **Test case title**: Login(Admin) | | |
| **Test case description**: User has to enter username and password | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The system prompts the user to enter username and password. Password entered is same as the hard-coded one. | A message saying “Welcome! Have a nice day. Enter any key for services” is displayed on the screen. The user is directed to the main menu. | A message saying “Welcome! Have a nice day. Enter any key for services” is displayed on the screen. The user is directed to the main menu. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC03 | | | Use case ID  **UC02** |
| **Test case title** : Book a room(Admin) | | |
| **Test case description** : The admin books a room for the customer. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 1, it displays all the hotel rooms which are available and not. User has to enter the room number which he wants to book. Entered room number is invalid. | An error message saying “Invalid room no” is displayed on the screen. | An error message saying “Invalid room no” is displayed on the screen. | |

* BOOK A ROOM

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC04 | | | Use case ID  **UC02** |
| **Test case title** : Book a room(Admin) | | |
| **Test case description** : The admin books a room for the customer. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 1, it displays all the hotel rooms which are available and not.  User has to enter the room number which he wants to book. System prompts the user to fill a form of customer details. Entered date is past or completed. | An error message “Enter valid date” is displayed on the screen. | An error message “Enter valid date” is displayed on the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC05 | | | Use case ID  **UC02** |
| **Test case title** : Book a room(Admin) | | |
| **Test case description** : The admin books a room for the customer. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 1, it displays all the hotel rooms which are available and not.  User has to enter the room number which he wants to book. Entered room number is valid and entered details are valid. | A room will be booked for the customer. A message showing “The room has been booked for x(number) days.” | A room will be booked for the customer. A message showing “The room has been booked for x(number) days.” | |

* VIEW ROOM RECORDS

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC06 | | | Use case ID  **UC03** |
| **Test case title** : View room records(Admin) | | |
| **Test case description** : User can view the customer details. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 2. | The list of all the existing customers in the hotel with their details are displayed on the screen. | The list of all the existing customers in the hotel with their details are displayed on the screen. | |

* SEARCH CUSTOMER RECORD

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC07 | | | Use case ID  **UC04** |
| **Test case title** : Search Customer record(Admin) | | |
| **Test case description** : User can search for a customer in a particular room. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 3. The  system prompts the user to  enter for the room number to be searched. The entered room number is not booked or  invalid. | A message saying “Requested room number is not booked” is displayed on the screen. | A message saying “Requested room number is not booked” is displayed on the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC08 | | | Use case ID  **UC04** |
| **Test case title** : Search Customer record(Admin) | | |
| **Test case description** : User can search for a customer in a particular room. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 3. The  system prompts the user to  enter for the room number to be searched. The entered room number is booked. | It displays the customer  details of the requested room number on the screen. | It displays the customer  details of the requested room number on the screen. | |

* FOOD SERVICES

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC09 | | | Use case ID  **UC05** |
| **Test case title** : Food services(Admin) | | |
| **Test case description** : Customer is provided with the food services in the hotel | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 5. | The food services section with 3 options order, display and exit are displayed on the screen and prompts the user to enter his/her choice. | The food services section with 3 options order, display and exit are displayed on the screen and prompts the user to enter his/her choice. | |

* ORDER

|  |  |  |  |
| --- | --- | --- | --- |
| * TEST CASE TEMPLATE | | | |
| **Test case ID** : TC10 | | | Use case ID  **UC06** |
| **Test case title** : Order(Admin) | | |
| **Test case description** : The food order is placed for the customers. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 1 in the food services section. The system prompts the user to enter name and displays the food menu. The user enters the food item code to place the food order. The entered item code is invalid. | An error message saying “Invalid selection try again” is displayed on the screen. | An error message saying “Invalid selection try again” is displayed on the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC11 | | | Use case ID  **UC06** |
| **Test case title** : Order(Admin) | | |
| **Test case description** : The food order is placed for the customers. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 1 in the food services section. The system prompts the user to enter name and displays the food menu. The user enters the food item code to place the food order. The entered item code is valid and ask the user if he want to order anything(y/n). The user choice is y or Y. | The system prompts the user to enter item code for order. | The system prompts the user to enter item code for order. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC12 | | | Use case ID  **UC06** |
| **Test case title** : Order(Admin) | | |
| **Test case description** : The food order is placed for the customers. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 1 in the food services section. The system prompts the user to enter name and displays the food menu. The user enters the food item code to place the food order. The entered item code is valid and ask the user if he want to order anything(y/n). The user choice is n or N. | The number of ordered food and the total bill generated are displayed on the screen. A message “Thank you” and “Please wait while we prepare food” are displayed on the screen. | The number of ordered food and the total bill generated are displayed on the screen. A message “Thank you” and “Please wait while we prepare food” are displayed on the screen. | |

* DISPLAY

|  |  |  |  |
| --- | --- | --- | --- |
| * TEST CASE TEMPLATE | | | |
| **Test case ID** : TC13 | | | Use case ID  **UC07** |
| **Test case title** : Display(Admin) | | |
| **Test case description** : The user can see the total number of food orders taken by the hotel. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 2 in the food services section. | The system displays the total number of orders taken by the hotel. | The system displays the total number of orders taken by the hotel. | |

* EXIT

|  |  |  |  |
| --- | --- | --- | --- |
| * TEST CASE TEMPLATE | | | |
| **Test case ID** : TC14 | | | Use case ID  **UC08** |
| **Test case title** : Exit(Admin) | | |
| **Test case description** : The user can go back to the main menu from the food services section. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 3 in the food services section. | The system takes the user back to the main menu. | The system takes the user back to the main menu. | |

* REMINDER

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC15 | | | Use case ID  **UC09** |
| **Test case title** : Reminder(Admin) | | |
| **Test case description** : The user can be notified about the room check-out. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 6. The system prompts the user to enter the date on which the user wants to be notified for check-out of a room. The entered date matches with room number check-out date. | The room number which is matched and date are displayed on the screen. A message saying “Last date today” is displayed on the screen. | The room number which is matched and date are displayed on the screen. A message saying “Last date today” is displayed on the screen. | |

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC16 | | | Use case ID  **UC09** |
| **Test case title** : Reminder(Admin) | | |
| **Test case description** : The user can be notified about the room check-out. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 6. The system prompts the user to enter the date on which the user wants to be notified for check-out of a room. The entered date doesn’t match with any check-out date. | A message “No notification today” is displayed on the screen. | A message “No notification today” is displayed on the screen. | |

* CHECK-OUT

|  |  |  |  |
| --- | --- | --- | --- |
| * TEST CASE TEMPLATE | | | |
| **Test case ID** : TC17 | | | Use case ID  **UC10** |
| **Test case title** : Check-out a room(Admin) | | |
| **Test case description** : The customer check-out is done. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 4. The system prompts the user to enter the room number to checkout. | The room bill is generated and displayed on the screen. It shows a message “The room is available again” on the screen. | The room bill is generated and displayed on the screen. It shows a message “The room is available again” on the screen. | |

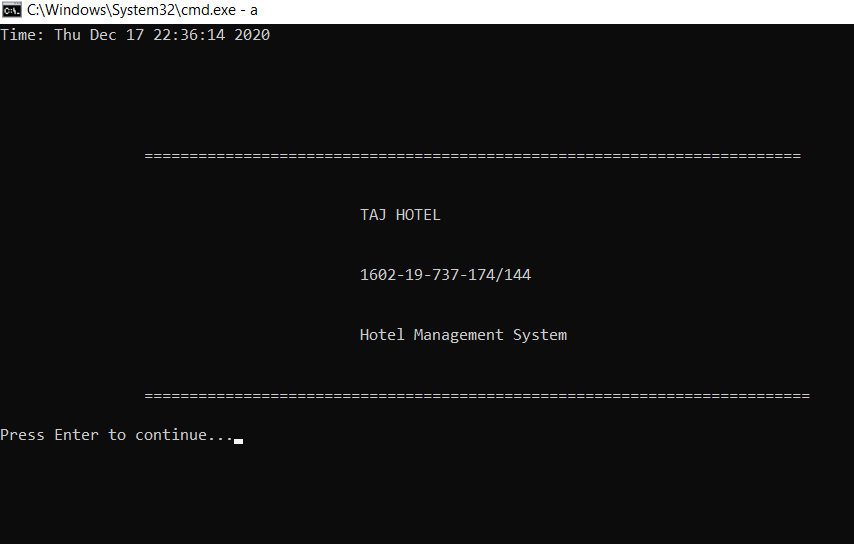
* EXIT

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE TEMPLATE | | | |
| **Test case ID** : TC18 | | | Use case ID :  **UC11** |
| **Test case title** : Exit(Admin) | | |
| **Test case description** : The user can exit from the program. | | |
| **Test steps** | **Expected result** | **Actual result** | |
| The user choice is 7. | The user exits the program and the message “THANK YOU.. VISIT AGAIN” is displayed on the screen. | The user exits the program and the message “THANK YOU.. VISIT AGAIN” is displayed on the screen. | |

**4.0)** **RESULTS**

**SCREENSHOTS OF EXECUTION**

**4.1) Start page with Hotel logo**



**4.2) Login**

After entering any key, the system takes the user to login page and prompts the user to enter the username and password.

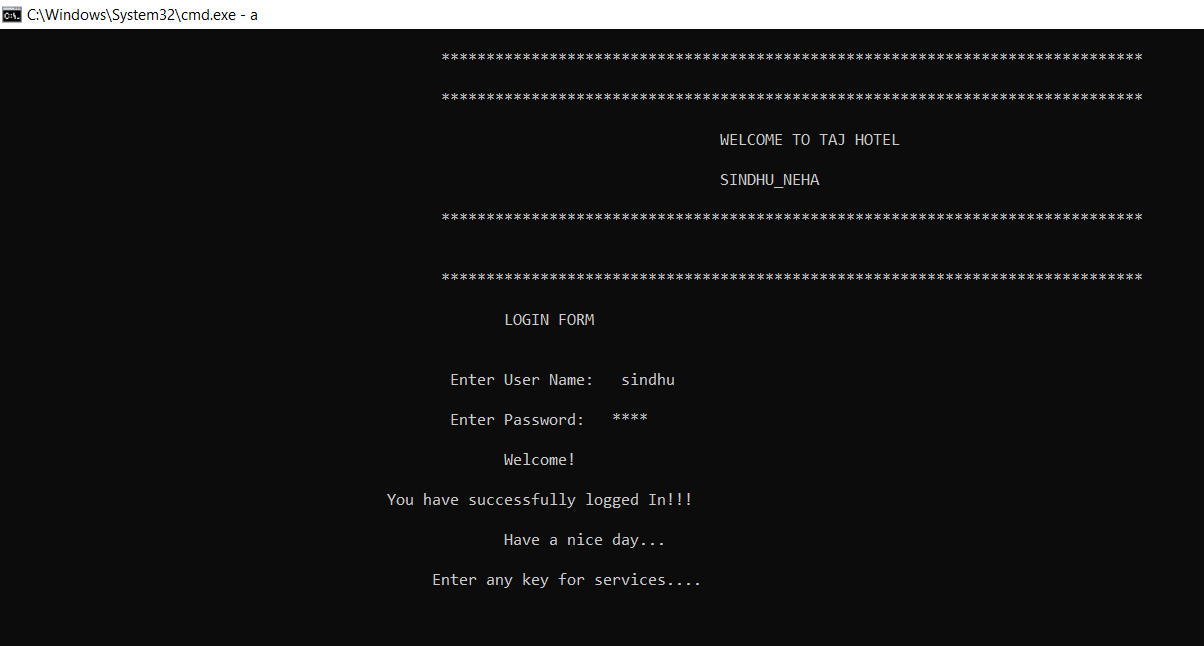
1. If the admin enters wrong username or password



1. If the admin enters the wrong username or password for 3 times

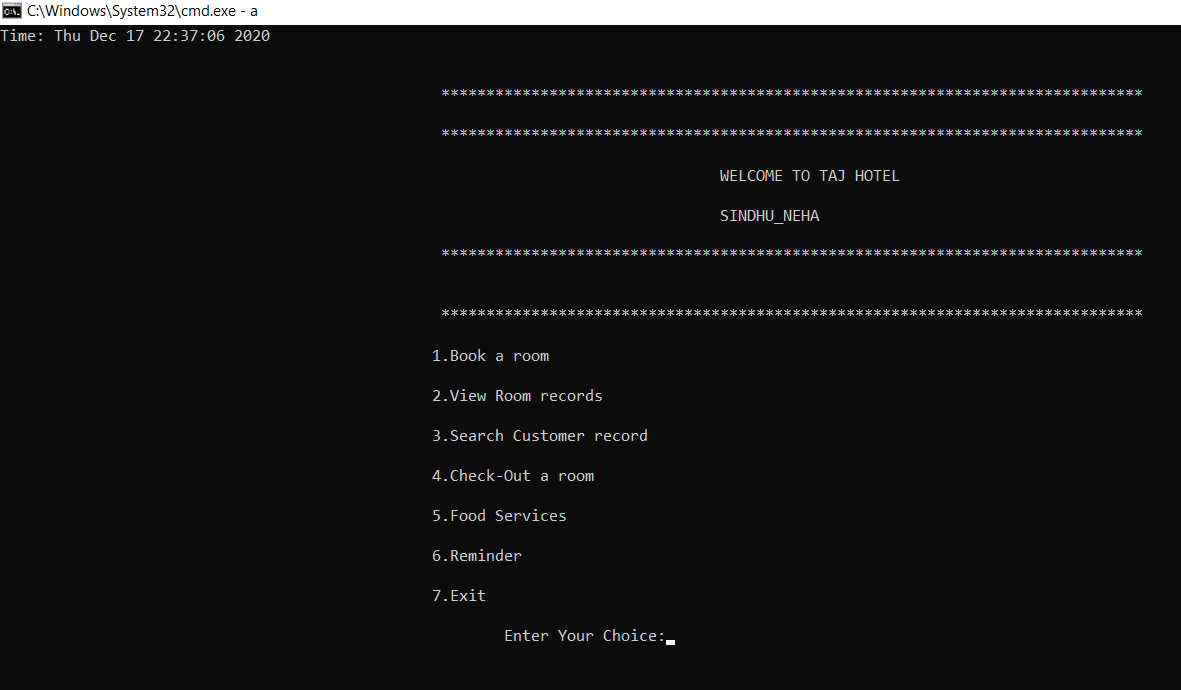


1. If the user enters correct username and password



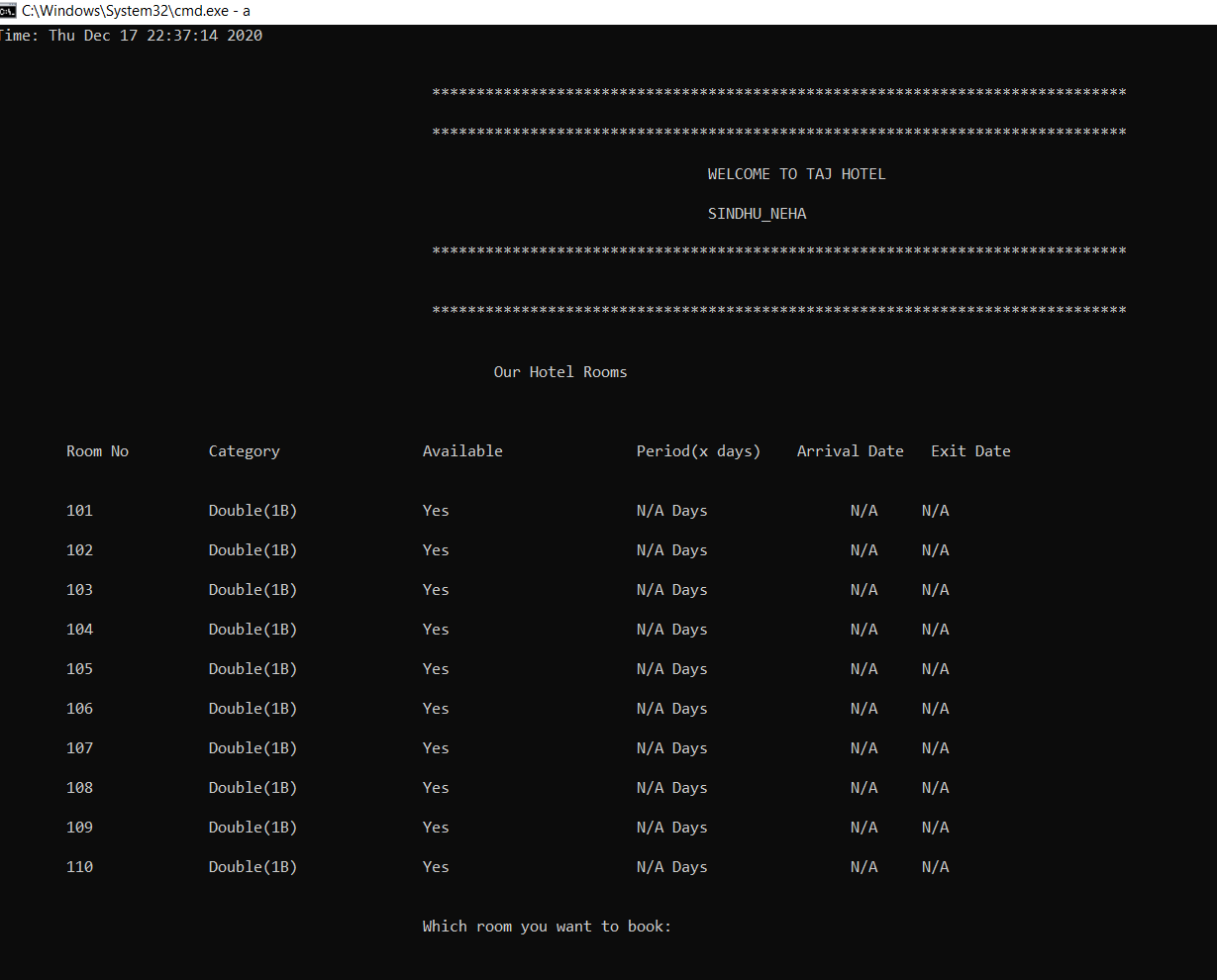
**4.3) Main menu**

After login, member should be directed to main menu. It shows all the features of the project including book a room, view room records, search a customer, check-out a room, food services reminder and exit.

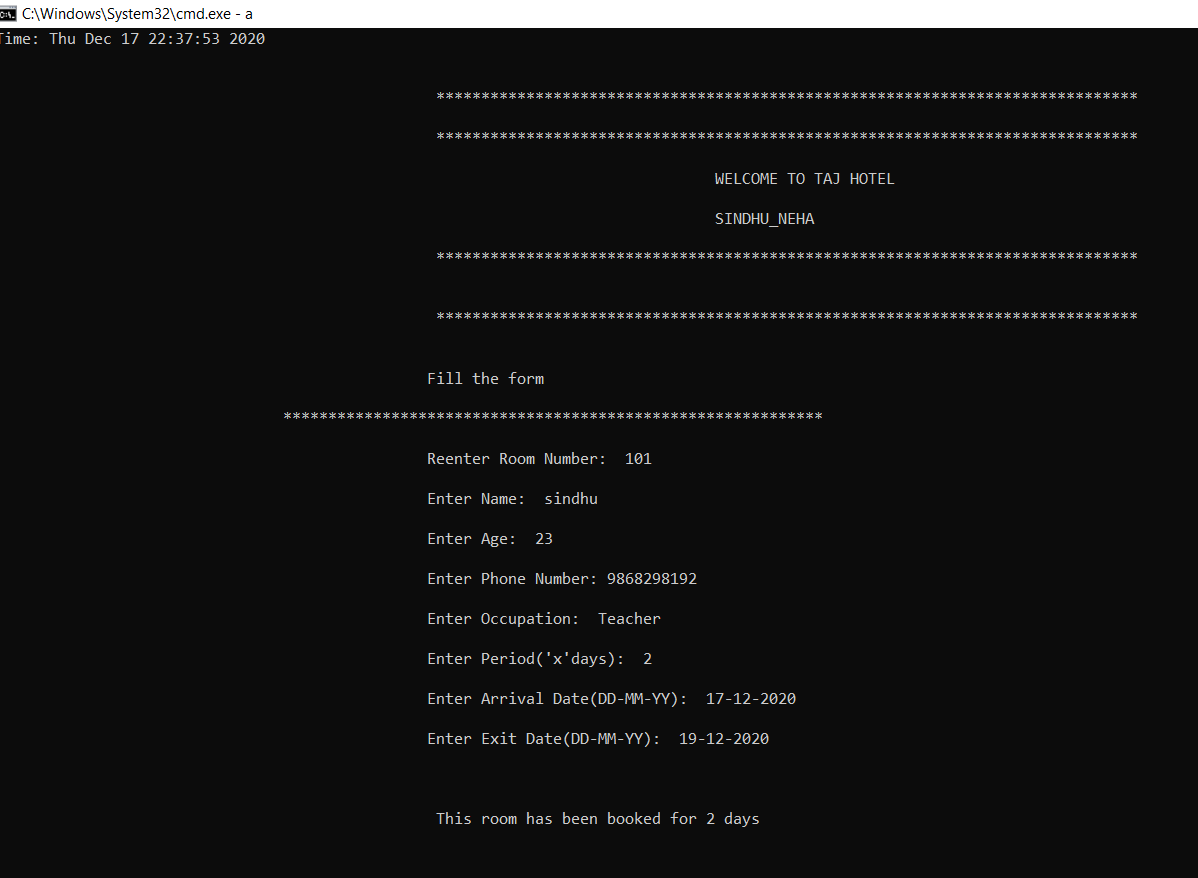


**4.4) Book a room**

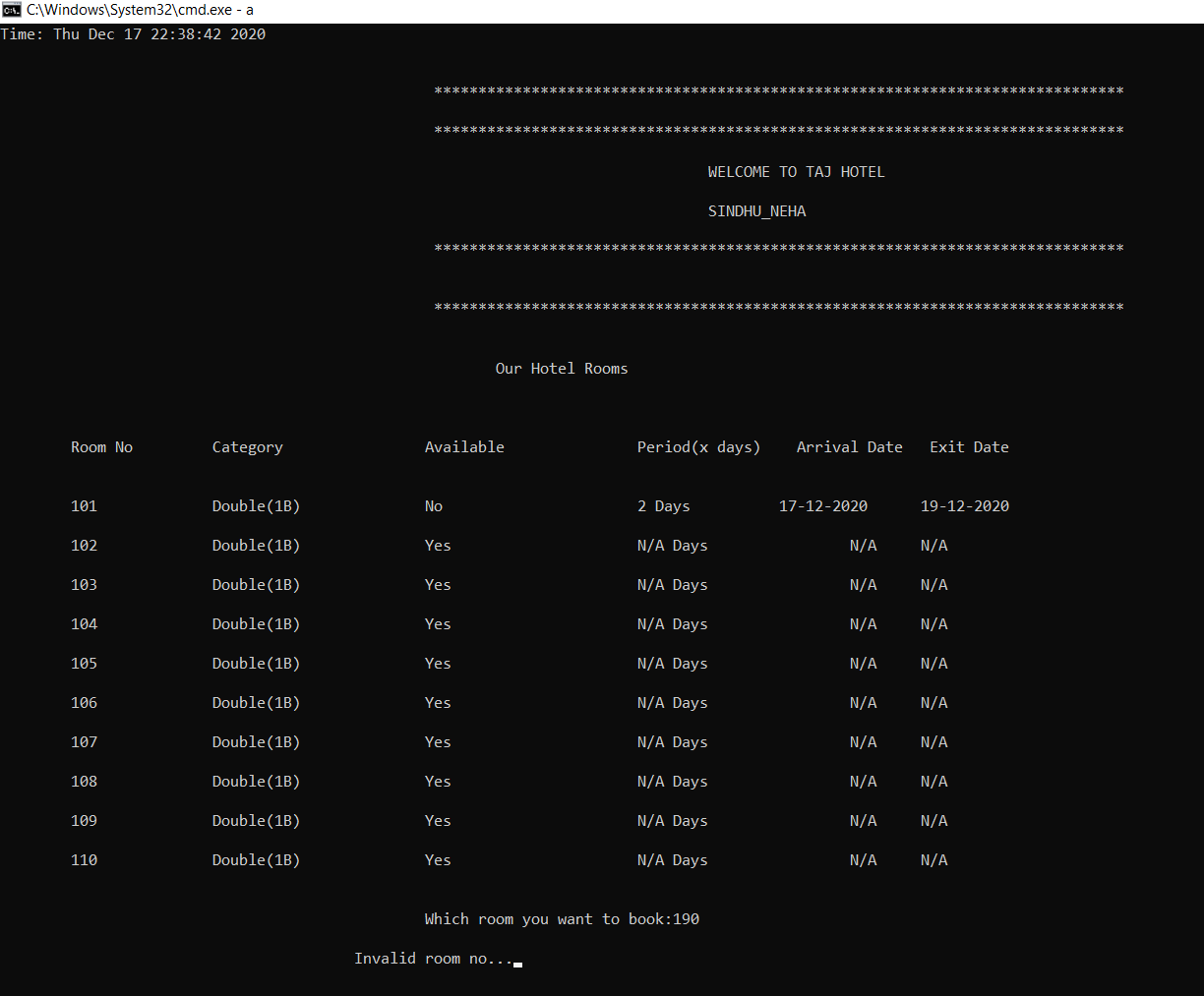
a) It shows all the available rooms in the hotel and ask for room number to book.



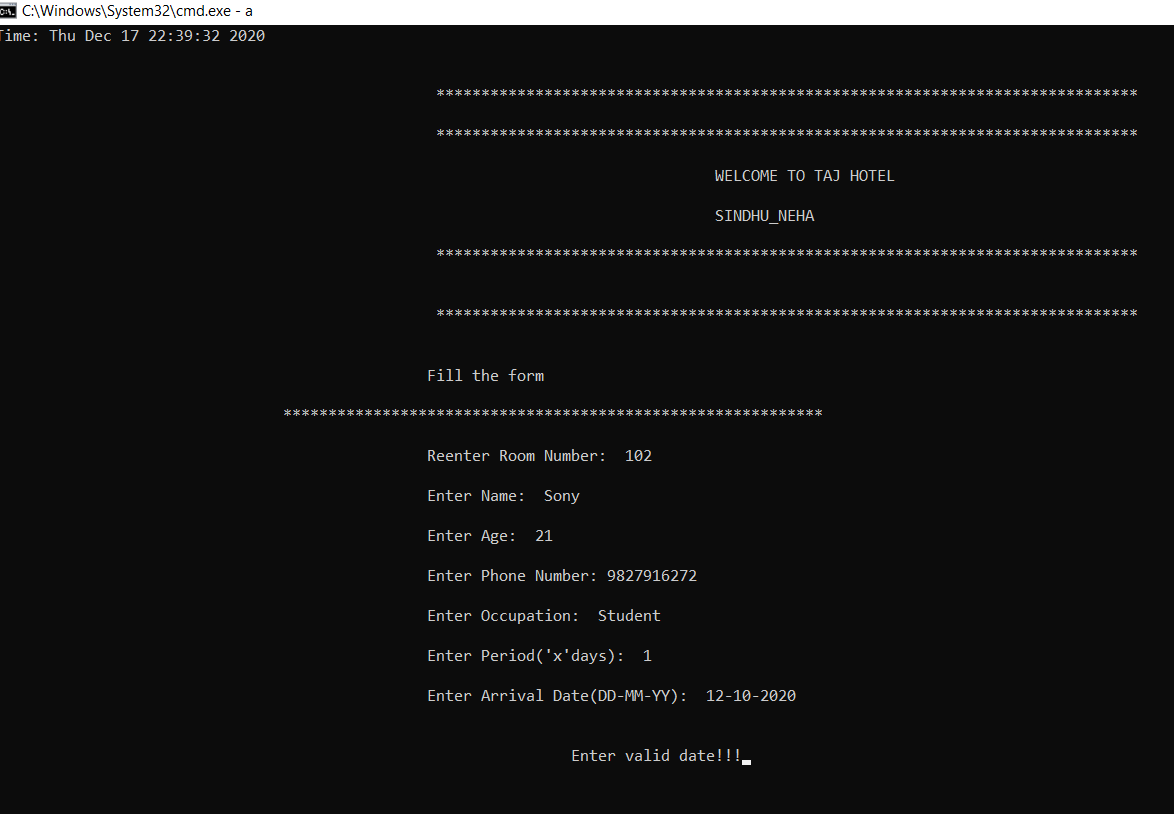
1. When the user enters valid room number, it asks to fill the customer details. If all the details are entered correctly, the room will be booked.



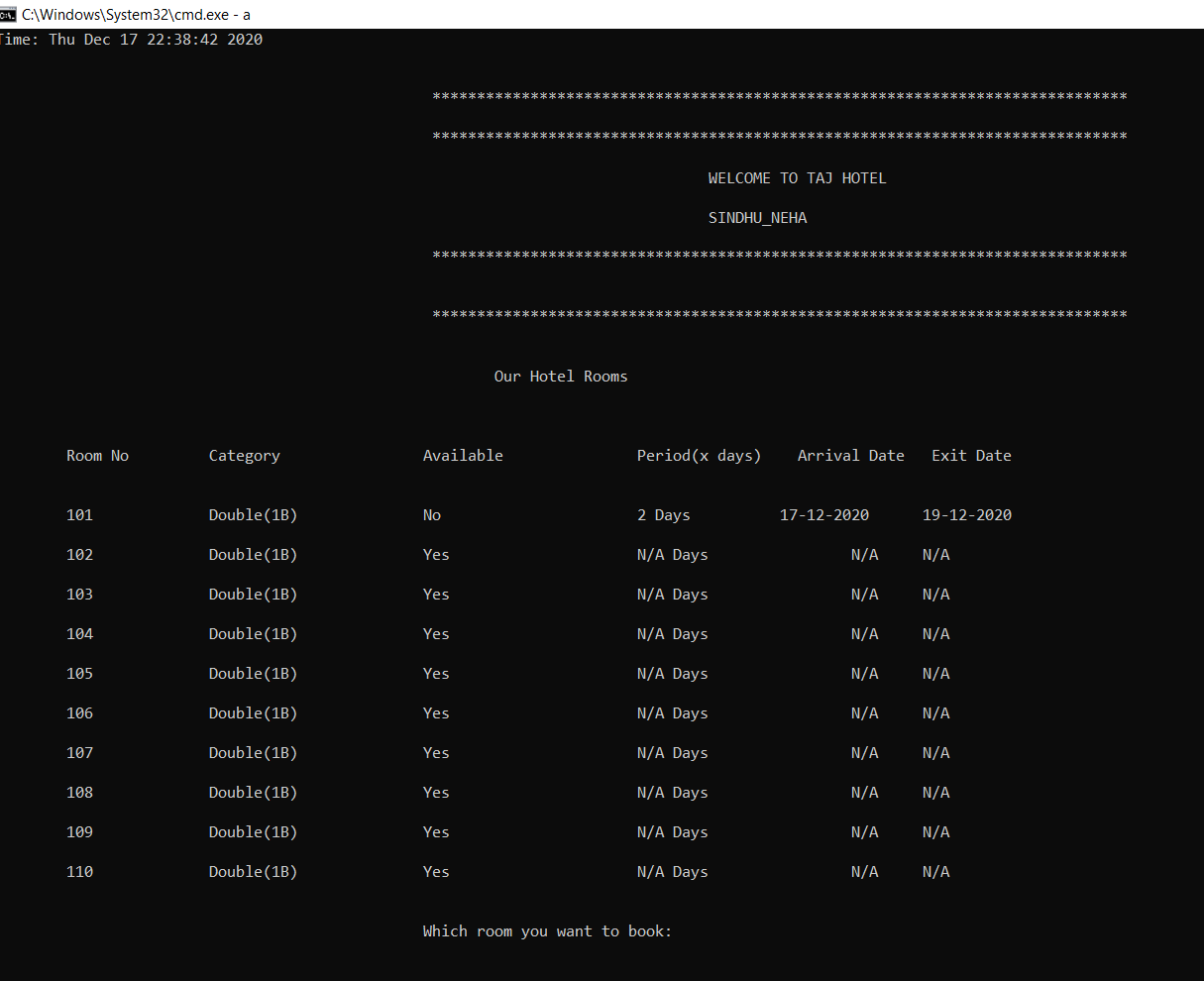
1. When the user enters invalid room number, it shows the message “Invalid room no..”



1. If the user insert a date that is past, it shows the message “Enter valid date..” message to notify the user about the mistake that has been.

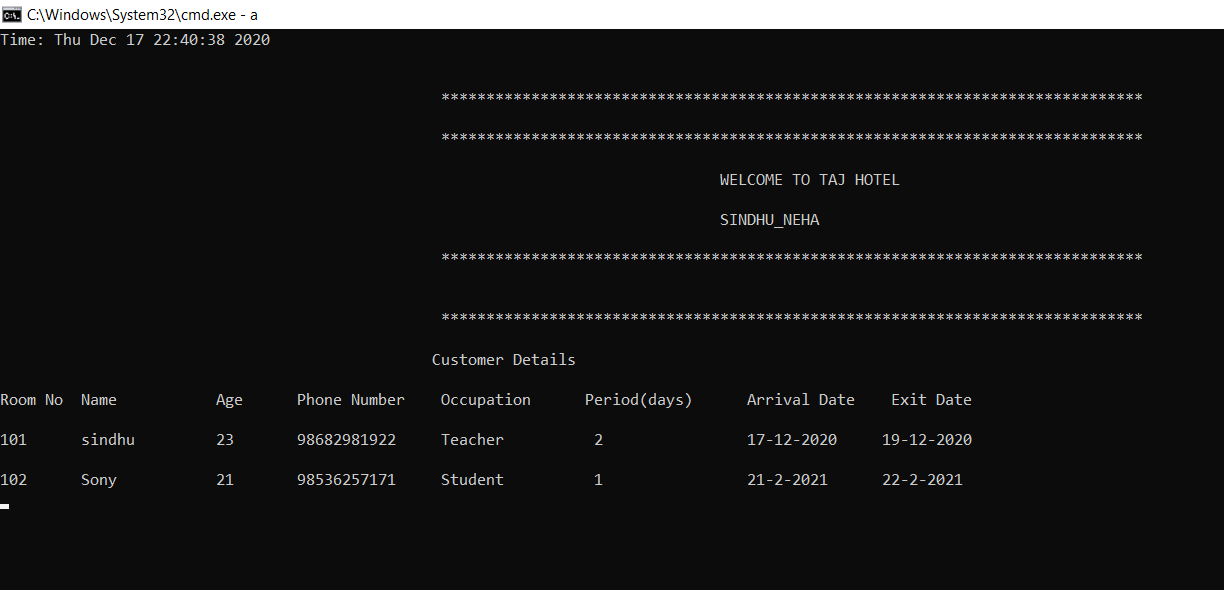


1. After booking a room, it shows that room is not available in hotel rooms



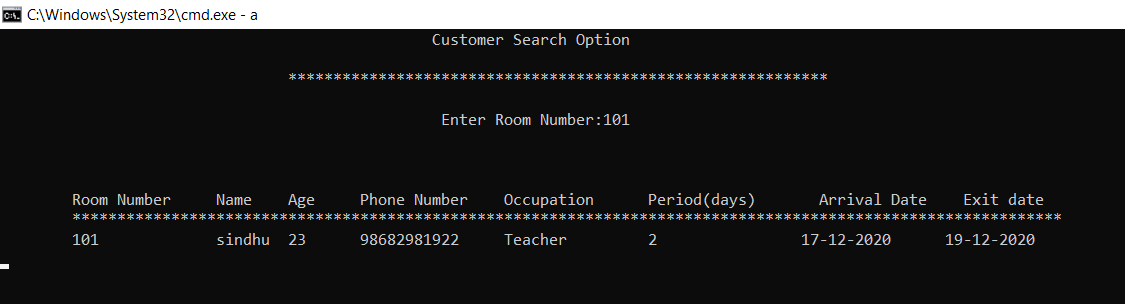
**4.5) View room records**

It shows all the existing customer details in the hotel

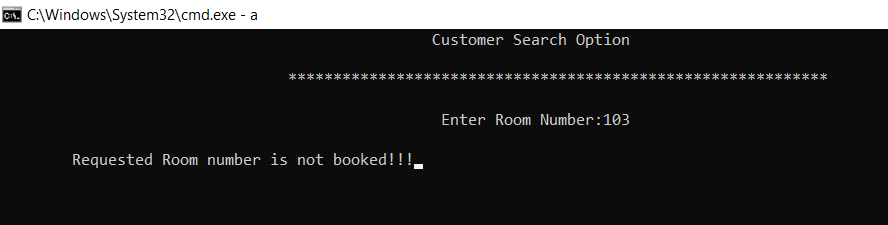


**4.6)** **Search Customer Record**

a) It asks the user to enter room number to be searched. If the room number is booked, it shows the customer details in that provided room.

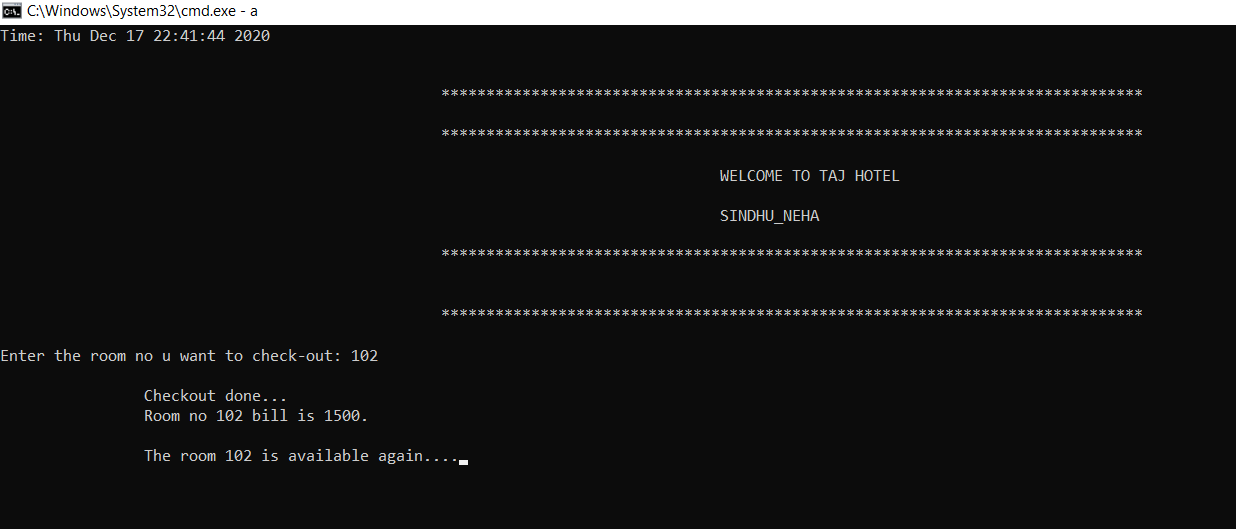


b) If the entered room number is not booked, it notifies the user that the searched room number is not booked.



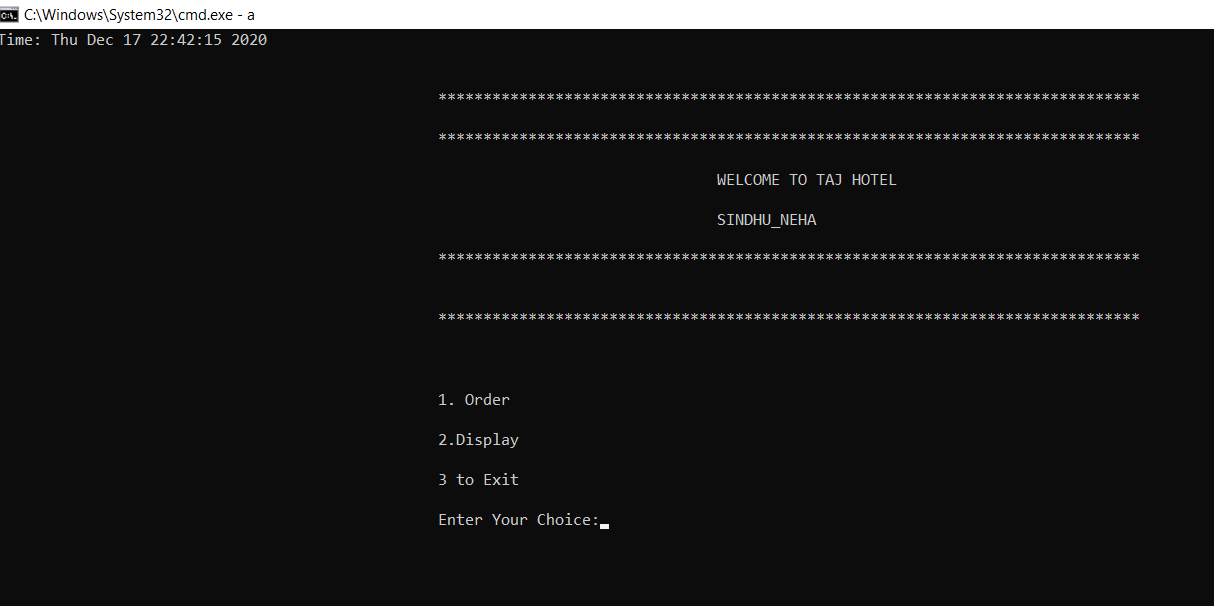
**4.7) Check-out a room**

If the user wants to check-out a room, he has to enter the room number to be check-out and the total bill is generated that is to be paid by the customer. After check-out is done, all the customer details of that room will be deleted in the database and shows the room is available again in the hotel rooms menu.

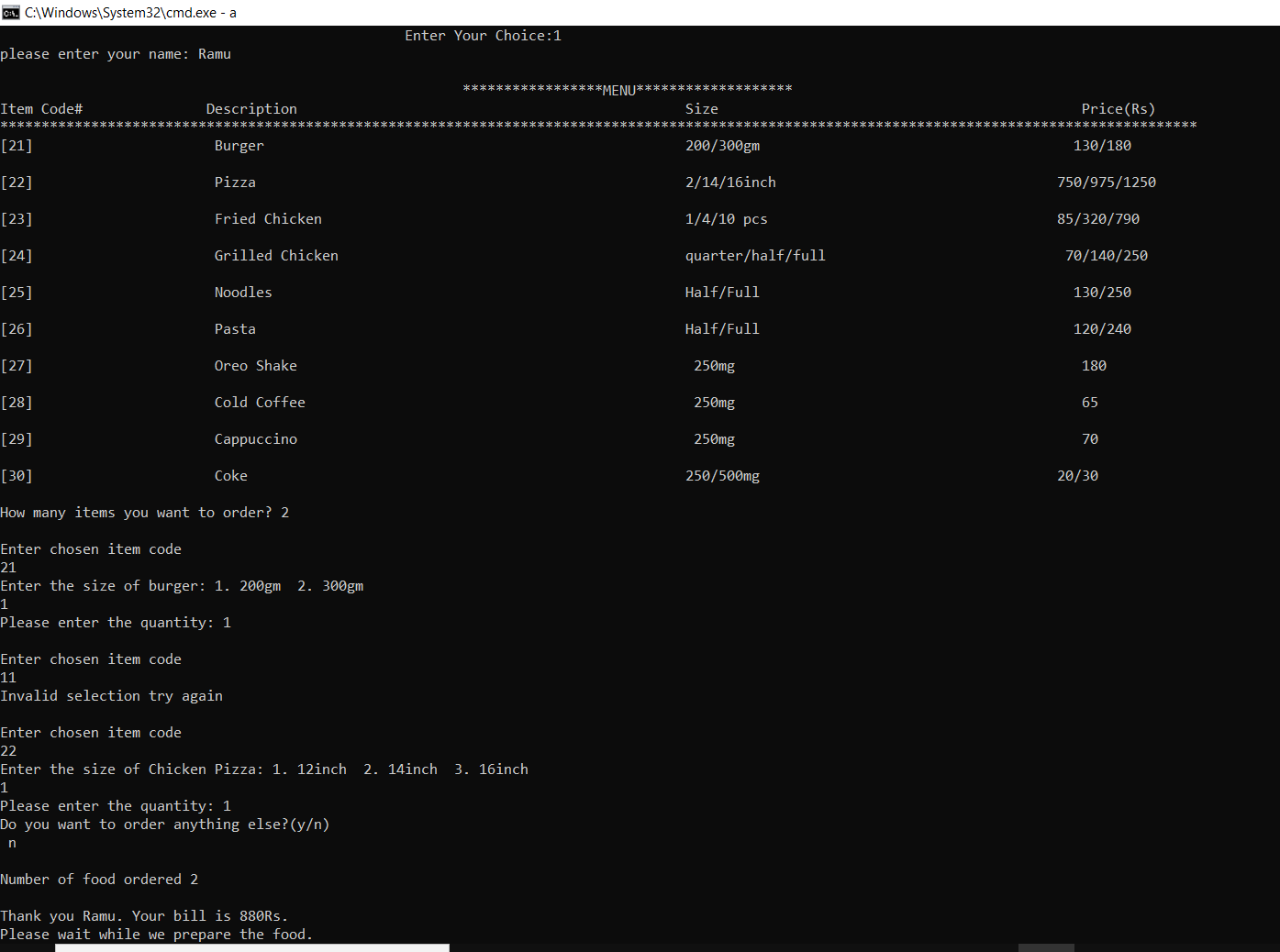


**4.8) Food services**

In the food services menu, there are 3 sections: Order, Display and Exit



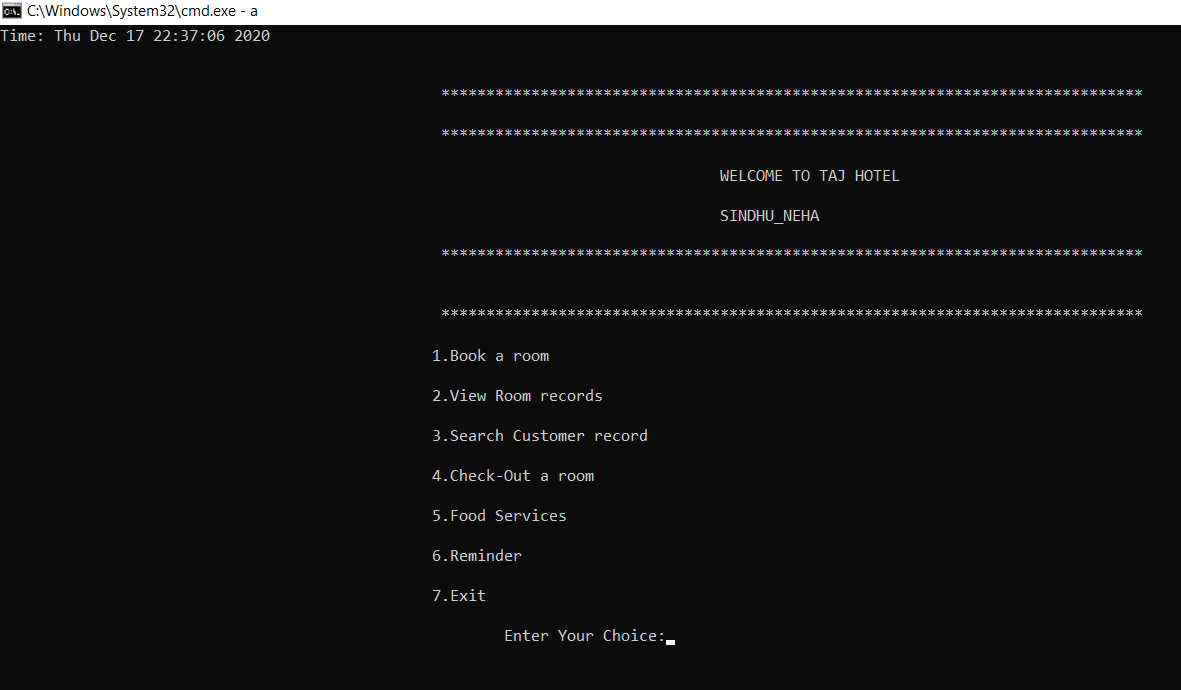
1. If the user choice is 1, it prompts for the customer name and displays the food menu. Then customer can place his food order by item codes. If the entered item code is invalid, it notifies the user that the item code is invalid. After the order is placed, total bill is generated to be paid by the customer.



1. If the user choice is 2, it displays the total number of food orders placed.

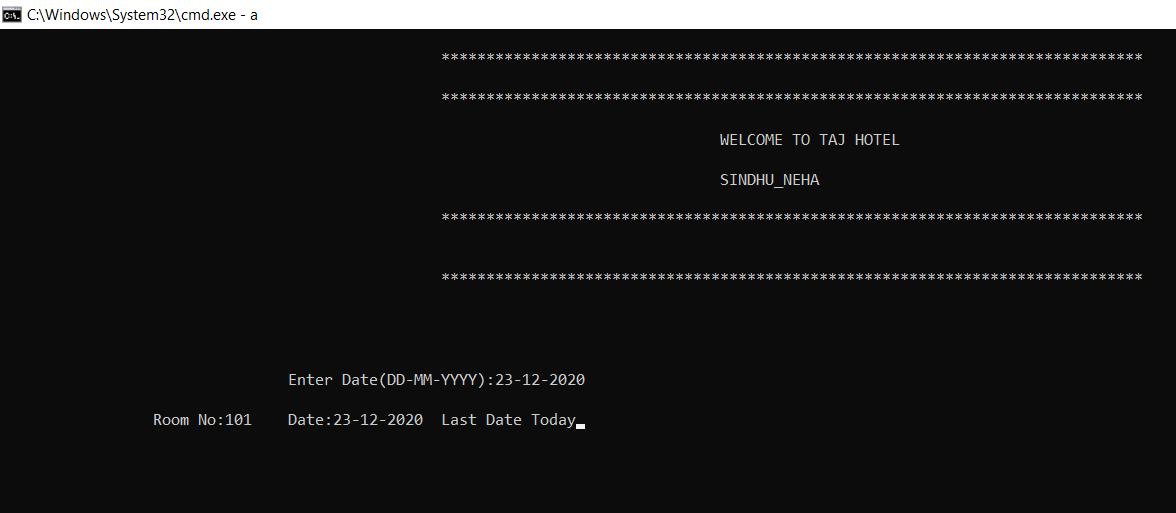


1. If the user choice is 3, it takes the user to the main menu from the food services menu.

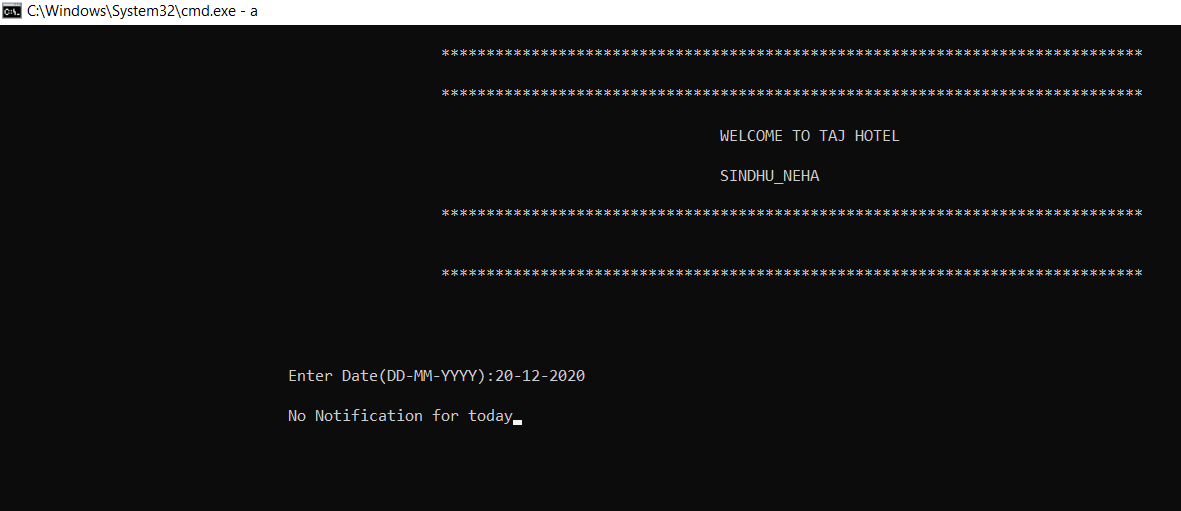


**4.9) Reminder**

1. It prompts the user to enter the date to check any room check-out is to be done on that day. If there is any room check-out date matches with the entered date, it displays all those room numbers to notify the user that those room numbers check-out should be done on that day.



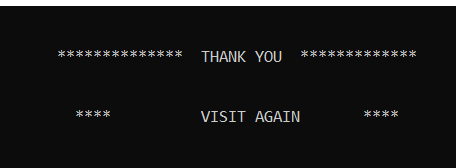
1. If any room check-out date doesn’t match with the entered date, It shows the message that is there is no notification today.



**4.10) Exit**

If the user wants to exit from the program, he can choose 7 in the main menu.

Thankyou note will be displayed on the screen.



**5.0) ADDITIONAL KNOWLEDGE GAINED**

We gained practical knowledge by doing this project. It helped us to grasp the C concepts in depth. It gave us an opportunity to apply our knowledge on C in innovative ways. It strengthened the understanding of our fundamentals through effective application of theoretical concepts. Our project mostly depends on file handling and we have gained lot of improvement in this part. We have learned various concepts like strtok, time functions, how to organize different modules in a single program and also the structure concepts.

**6.0) CONCLUSION**

We are living in a society where computer software has taken place over the paper and manual work. This project work is an attempt to develop a system that can be used for computerization of activities in the hotel. Since these activities are tedious process requiring lot of effort, more care has been for the system development. Our project doesn’t require any paper work and created in user friendly environment. Data security and reliability is also maintained. It increases efficiency and saves time. This represents a typical real world situation. In order to generate the final reports, all the functions have to be meticulously followed. This has been strengthened the understanding project. After the feasibility step, it is observed that this project is feasible and justified. As everything in the world have pros and cons. This applied on our project also. It helps the hotel staff to keep the track of their hotel rooms, customers database and their check-in and check-outs. We have also added an additional feature of notifying the admin about the hotel room check-outs and also included food services section. We have gained immense knowledge in coding and programming throughout this mini project. We have improved our collaborating skills and had fun while debugging and fixing errors. It was a great opportunity for both of us. The hotel management system project was implemented and executed successfully. All the features that are required for the HOTEL MANAGEMENT SYSTEM has been finished successfully.

**FUTURE ENHANCEMENT**

The proposed system is Hotel Management System. We can enhance this system by including more facilities like online booking, advance booking and cancellation, regular customer record, billing system, sending notifications to customers on their check-out dates. For security purpose, advanced encryption techniques can also be applied. Providing such features enable the users to include more comments into the system. Users can add extra enhancements in the system as per necessity in the future fulfillment of the requirement. Further requirements and improvements can easily be done since the coding is mainly structured or modular in nature. Changing the existing modules or adding new module can append improvements.

**7.0)** **REFERENCES**

We have referred to following sources for references during this project:

1. Computer Science: A structured Programming Approach Using C- Third Edition
2. [www.google.com](http://www.google.com)
3. [www.scribd.com](http://www.scribd.com)
4. [www.stackoverflow.com](http://www.stackoverflow.com)
5. [www.code-projects.org](http://www.code-projects.org)