**HOTEL MANAGEMENT SYSTEM**



**--- PREPARED BY ---**

**Naqeeb Masood**

**Muhammad Anas**

**Mohammad Abdullah Sohail**

**--INTRODUCTION--**

This hotel management system is developed for hotels those are using a manual system to handle hotel processes. Our project has management system that can store the details of staff and managers. It also has food serving system and room management system. The program cannot be executed without the authorized person’s Password. He can further change the password if he wants. Manager can access to all system functionalities with limited restrictions. The main aim of the entire activity is to automate the process of day to day activities of Hotel.

**PURPOSE OF ONLINE SYSTEM**

The Hotel Management System aims to make simpler a staff’s interaction with the various modules of the Hotel and ease the process of acquiring information and providing services.. It will also allow cutting the operational costs of the hotel.

**PROJECT AIMS AND OBJECTIVES**

The project aims and objectives that will be achieved after completion of this project are as follows:

* Store the details of managers and staff.
* Online food ordering system with additional option of canceling specific item.
* Room booking system with additional option of updating details of room through Room number.

**ADVANTAGES**

The Advantages are in the following:

* Save time on admin task.
* Develop strong relationships with your guests.
* Increase your online visibility.
* Increase bookings.

**DISVANTAGES**

There is few Disadvantages of Hotel management system which is in the following:

* If there is a power failure while making a reservation, all the information has to be given again
* If there is a power cut while there is a check-in or check-out, the guest need to wait for full server start up

**FUNCTIONALITY**

* **USER LOGIN:**

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system .The user id and password will be verified and if invalid id is there user is allowed to not enter the system. User id is provided when they register. The system must only allow user with valid id and password to enter the system.

* **REGISTER STAFF DETAILS:**
* This feature can be used only by Administrator to edit Staff and Manager Details.
* System must be able to delete information if information is wrong. File handling is used to save all the record i.e. Staff member name, cnic number, Age , Address etc.
* **ADDING/MODIFYING DETAILS:**
* This feature allows Admin to edit the details of Staff and managers.
* In Food Serving System it helps the customer to edit the order without canceling the whole order.

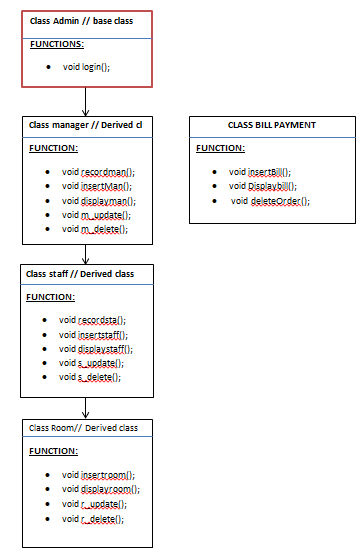
**SEARCH SYSTEM:**

* In Room Management System it allows the admin or Manager to edit the details of a user with the help of Room Number

**PROJECT SPECIFICATION:**

* **Classes:**

We used multilevel inheritance in which, a derived class will inherit a base class and as well as the derived class also act as the base class to other class. In this situation, each derived class inherits all the characteristics of its base classes.



* **PROJECT SOURCE CODE:**

**MANAGER.H**

#ifndef MANAGER\_H

#define MANAGER\_H

#include <fstream>

using namespace std;

class managerClass

{

  private:

    string mname,mage,madress,mcnic;

  public:

   void recordman();                                         //RECORD      for      Management System;

   void insertMan( fstream& managerDetail);                  // Insert manager Details

   void displayman(fstream& managerDetail);                  //Display Manager Details

   void m\_update(fstream& managerDetail);                    //Manager update

   void m\_delete(fstream& managerDetail);                    //manager delete

};

#endif

**MANAGER.CPP:**

#include<iostream>

#include<string>

#include<windows.h>

#include "manager.h"

#include<fstream>

using namespace std;

void managerClass::recordman()

{

     system("cls");

     cout<<"\n\n\n\t\t\t\tManagement Record";

     cout<<"\n\n Press 1 for Insert Manager Record";

     cout<<"\n Press 2 for Update Manager Detail";

     cout<<"\n Press 3 for Delete Some Manager Detail";

     cout<<"\n Press 4 for Display Manager Record";

     cout<<"\n Press 5 for Exit on Management System ";

     cout<<"\n Press 6 for Exit on Main Dashboard ";

     cout<<"\n\n Select Your Choice : ";

}

 void managerClass::insertMan(fstream& managerDetail)  // Insert manager Details

 {

     system("cls");

     managerDetail.open("Manager\_Details.txt", ios::app | ios::out);

     cout<<"\n\t\t\tInsert Manager Details\n";

     cin.ignore();

     cout<<"\n\t Manager Name  : ";

     getline(cin,mname);

     cout<<"\n\t Manager Age   : ";

     getline(cin,mage);

     cout<<"\n\tManager Adress : ";

     getline(cin,madress);

     cout<<"\n\t Manager CNIC  : ";

     getline(cin,mcnic);

        //Store Manager Detail

     managerDetail << " " << mname << " " << mage << " " << madress << " " << mcnic << "\n";

     managerDetail.close();

   }

void managerClass::displayman(fstream& managerDetail)

{

     system("cls");

     int totalm = 0;

     managerDetail.open("Manager\_Details.txt", ios::in | ios::out);

    if(!managerDetail)

    {

         cout << "\n\t\t\tNo Manager Detail File Is Present...";

    }else

    {

         managerDetail >> mname >> mage >> madress >> mcnic ;

        while (!managerDetail.eof())

        {

             cout<<"\n\t\t\t---------------------------";

             cout<<"\n\t\t\t  Display Manager Details";

             cout<<"\n\t\t\t---------------------------";

             cout << "\n\n\t\t   Manager Name : " << mname;

             cout << "\n\t\t    Manager Age : " << mage;

             cout << "\n\t\t Manager Address: " << madress;

             cout << "\n\t\t   Manager Cnic :" << mcnic;

             managerDetail >> mname >> mage >> madress >> mcnic ;

             totalm++;

        }

        if (totalm == 0)

        {

             cout << "\n\t\t\tNo Manager Data Is Present...";

        }

    }

     managerDetail.close();

}

void managerClass::m\_update(fstream& managerDetail)                                           //Manager update

{

    system("cls");

    fstream  ma;

    string name;

    int foundm = 0;

    managerDetail.open("Manager\_Details.txt", ios::in);

    if (!managerDetail)

    {

         cout << "\n\t\t\tNo Manager Detail File Is Present...";

    }

    else

    {

         cin.ignore();

         cout << "\nEnter Name of Manager for update: ";

         getline(cin, name);

         ma.open("record.txt", ios::app | ios::out);

         managerDetail >> mname >> mage >> madress>> mcnic ;

        while (!managerDetail.eof())

        {

            if (name != mname){

                ma << " " << mname << " " << mage << " " << madress << " " << mcnic << "\n";

             }

            else

            {

                cout<<"\n\n\t\t\t\tInsert Record"<<endl;

                cin.ignore();

                cout<<"\n\t Manager Name   : ";

                getline(cin, mname);

                cout<<"\n\t  Manager Age   : ";

                getline(cin, mage);

                cout<<"\n\t Manager Address : ";

                getline(cin, madress);

                cout<<"\n\t  Manager Cnic  : ";

                getline(cin, mcnic);

                ma << " " << mname << "  " << mage << " " << madress << " " << mcnic << "\n";

                foundm++;

            }

            managerDetail >> mname >> mage >> madress >> mcnic;

        }

            if (foundm == 0)

            {

            cout << "\n\t\t\t Manager "<<name <<" Not Found....";

            }

        }

        ma.close();

        managerDetail.close();

        remove("Manager\_Details.txt");

        rename("record.txt", "Manager\_Details.txt");

}

void managerClass::m\_delete(fstream& managerDetail)                                          //manager delete

{

     system("cls");

     fstream  ma;

     int foundm = 0;

     string name;

     managerDetail.open("Manager\_Details.txt", ios::in);

    if (!managerDetail)

    {

         cout << "\n\t\t\tNo Manager Detail File Is Present...";

         managerDetail.close();

    }

    else

    {

         cin.ignore();

         cout << "\nEnter Name of Manager For Delete Data: ";

         getline(cin,name);

         ma.open("record.txt", ios::app | ios::out);

         managerDetail >> mname >> mage >> madress>>mcnic;

        while (!managerDetail.eof())

        {

            if (name != mname)

            {

                ma << " " << mname << " " << mage << " " << madress << " " << mcnic << "\n";

            }

            else

            {

                foundm++;

                cout << "\n\t\t\tSuccessfully "<<name<< " name of Data Deleted";

            }

            managerDetail >> mname >> mage >> madress>>mcnic;

        }

        if (foundm == 0)

        {

            cout << "\n\t\t\t Manager "<< name <<" Not Found....";

        }

        ma.close();

        managerDetail.close();

        remove("Manager\_Details.txt");

        rename("record.txt", "Manager\_Details.txt");

    }

}

**STAFF.H:**

#ifndef STAFF\_H

#define STAFF\_H

#include<fstream>

#include "manager.h"

using namespace std;

class staffClass : public managerClass{

  private:

   string sname,sage,sadress,scnic;

  public:

   void recordsta();                                   //RECORD      for      Staff System;

   void insertstaff(fstream& staffDetail);             // Insert Staff Details

   void displaystaff(fstream& staffDetail);            //Display Staff Detail

   void s\_update(fstream& staffDetail);                 //Staff update

   void s\_delete(fstream& staffDetail);                 //Staff delete

};

#endif

**STAFF.CPP:**

#include<iostream>

#include<string>

#include<windows.h>

#include "manager.h"

#include "staff.h"

#include<fstream>

using namespace std;

  void staffClass::recordsta()                                  //RECORD      for      Staff System;

  {

     system("cls");

     cout<<"\n\n\n\t\t\t\tStaff Record";

     cout<<"\n\n Press 1 for Insert Staff Record";

     cout<<"\n Press 2 for Update Staff Record";

     cout<<"\n Press 3 for Delete Some Staff Detail";

     cout<<"\n Press 4 for Display Staff Record";

     cout<<"\n Press 5 for Exit on Management System ";

     cout<<"\n Press 6 for Exit on Main Dashboard ";

     Sleep(900);

     cout<<"\n\n Select Your Choice : ";

   }

  void staffClass::insertstaff(fstream& staffDetail)                    // Insert Staff Details

  {

     system("cls");

     staffDetail.open("Staff\_Details.txt", ios::app | ios::out);

     cout<<"\n\t\t\tInsert Staff Details\n";

     cin.ignore();

     cout<<"\n\t  Staff Name  : ";

     getline(cin,sname);

     cout<<"\n\t  Staff Age  : ";

     getline(cin,sage);

     cout<<"\n\tStaff Adress : ";

     getline(cin,sadress);

     cout<<"\n\t Staff CNIC  : ";

     getline(cin,scnic);

    // Store Staff Detail

     staffDetail << " " << sname << " " << sage << " " << sadress << " " << scnic << "\n";

     staffDetail.close();

   }

  void staffClass::displaystaff(fstream& staffDetail)              //Display Staff Details

    {

     system("cls");

     int totals = 0;

     staffDetail.open("Staff\_Details.txt", ios::in);

    if(!staffDetail)

    {

        cout << "\n\t\t\tNo Staff Detail File Is Present...";

    }

    else

    {

        staffDetail >> sname >> sage >> sadress >> scnic ;

        while (!staffDetail.eof())

        {

             cout<<"\n\t\t\t---------------------------";

             cout<<"\n\t\t\t  Display Staff Details";

             cout<<"\n\t\t\t---------------------------";

             cout<<"\n\n\t\t Staff Name : " << sname;

             cout<<"\n\t\t Staff Age : " << sage;

             cout<<"\n\t\t Staff Address: " << sadress;

             cout<<"\n\t\t Staff Cnic :" << scnic;

             staffDetail >> sname >> sage >> sadress >> scnic ;

             totals++;

        }

        if(totals == 0)

        {

             cout << "\n\t\t\tNo Staff Data Is Present...";

        }

    }

     staffDetail.close();

   }

void staffClass::s\_update(fstream& staffDetail)             //Staff update

{

 system("cls");

    fstream  sa;

    string name;

    int foundss = 0;

    staffDetail.open("Staff\_Details.txt", ios::in);

    if (!staffDetail)

    {

        cout << "\n\t\t\tNo Staff Detail File Is Present...";

    }

    else

    {

        cin.ignore();

        cout << "\nEnter Name of Staff for update: ";

        getline(cin, name);

        sa.open("record.txt", ios::app | ios::out);

        staffDetail >> sname >> sage >> sadress >> scnic ;

        while (!staffDetail.eof())

        {

            if (name != sname){

                sa << " " << sname << " " << sage << " " << sadress << " " << scnic << "\n";

             }

            else

            {

                cout<<"\n\n\t\t\t\tInsert Record"<<endl;

                cin.ignore();

                cout<<"\n\t Staff Name   : ";

                getline(cin, sname);

                cout<<"\n\t  Staff Age   : ";

                getline(cin, sage);

                cout<<"\n\t Staff Address : ";

                getline(cin, sadress);

                cout<<"\n\t  Staff Cnic  : ";

                getline(cin, scnic);

                sa << " " << sname << " " << sage << " " << sadress << " " << scnic << "\n";

                foundss++;

            }

            staffDetail >> sname >> sage >> sadress >> scnic ;

        }

            if (foundss == 0)

            {

            cout << "\n\t\t\t Staff "<<name <<" Not Found....";

            }

        sa.close();

        staffDetail.close();

        remove("Staff\_Details.txt");

        rename("record.txt", "Staff\_Details.txt");

    }

}

void staffClass::s\_delete(fstream& staffDetail)        //Staff delete

{

 system("cls");

    fstream  sa;

    int founds = 0;

    string name;

    staffDetail.open("Staff\_Details.txt", ios::in);

    if (!staffDetail)

    {

        cout << "\n\t\t\tNo Staff Detail File Is Present...";

        staffDetail.close();

    }

    else

    {

        cin.ignore();

        cout << "\nEnter Name of Staff For Delete Data: ";

         getline(cin,name);

        sa.open("record.txt", ios::app | ios::out);

           staffDetail >> sname >> sage >> sadress >> scnic ;

        while (!staffDetail.eof())

        {

            if (name != sname)

            {

                sa << " " << sname << " " << sage << " " << sadress << " " << scnic << "\n";

            }

            else

            {

                founds++;

                cout << "\n\t\t\tSuccessfully "<<name<< " name of Data Deleted";

            }

           staffDetail >> sname >> sage >> sadress >> scnic ;

        }

        if (founds == 0)

        {

            cout << "\n\t\t\t Staff "<<name <<" Not Found....";

        }

        sa.close();

        staffDetail.close();

        remove("Staff\_Details.txt");

        rename("record.txt", "Staff\_Details.txt");

    }

}

**ROOM.H:**

#ifndef ROOM\_H

#define ROOM\_H

#include<fstream>

#include "staff.h"

using namespace std;

struct rom;

class roomClass: public staffClass{

private:

  string rnum,rname,rage,rcnic;

public:

   void insertroom(fstream& roomDetaill);              // Insert room Details

   void displayroom(fstream& roomDetaill);             //Display Room Details

   void r\_update(fstream& roomDetaill);                //Staff update

   void r\_delete(fstream& roomDetaill);                //room delete

};

#endif

**ROOM.CPP:**

#include<iostream>

#include<string>

#include<windows.h>

#include "manager.h"

#include "staff.h"

#include "room.h"

#include<fstream>

using namespace std;

 void roomClass::insertroom(fstream& roomDetaill)         // Insert room Details

 {

     system("cls");

     Sleep(800);

     roomDetaill.open("Room\_Details.txt", ios::app | ios::out);

     cout<<"\n\t\t\tInsert Customer Details\n";

     cin.ignore();

     cout<<"\n\t Room Number  : ";

     getline(cin,rnum);

     cout<<"\n\tCustomer Name : ";

     getline(cin,rname);

     cout<<"\n\tCustomer Age  : ";

     getline(cin,rage);

     cout<<"\n\tCustomer CNIC : ";

     getline(cin,rcnic);

     roomDetaill << " " << rnum << " " << rname << " " << rage << " " << rcnic << "\n";

     roomDetaill.close();

 }

 void roomClass::displayroom(fstream& roomDetaill)               //Display Room Details

{

    system("cls");

    int totalr = 0;

    roomDetaill.open("Room\_Details.txt", ios::in);

    if (!roomDetaill)

    {

        cout << "\n\t\t\tNo Room Detail File Is Present...";

    }

    else

    {

        roomDetaill >> rnum >> rname >> rage >> rcnic ;

        while (!roomDetaill.eof())

        {

            cout<<"\n\t\t\t---------------------------";

            cout<<"\n\t\t\t  Display Room Details";

            cout<<"\n\t\t\t---------------------------";

            cout << "\n\n\t\t Room Number : " << rnum;

            cout << "\n\t\t Customer Name : " << rname;

            cout << "\n\t\t Customer Age  : " << rage;

            cout << "\n\t\t Customer Cnic :"  << rcnic;

        roomDetaill >> rnum >> rname >> rage >> rcnic ;

           totalr++;

        }

        if (totalr == 0)

        {

            cout << "\n\t\t\tNo Room Data Is Present...";

        }

    }

    roomDetaill.close();

}

 void roomClass::r\_update(fstream& roomDetaill)                                //Staff update

{

system("cls");

    fstream  ra;

    string num;

    int foundr = 0;

    roomDetaill.open("Room\_Details.txt", ios::in);

    if (!roomDetaill)

    {

        cout << "\n\t\t\tNo Room Detail File Is Present...";

    }

    else

    {

        cin.ignore();

        cout << "\nEnter Room Number of Customer for update: ";

        getline(cin, num);

        ra.open("record.txt", ios::app | ios::out);

        roomDetaill >> rnum >> rname >> rage >> rcnic ;

        while (!roomDetaill.eof())

        {

            if (num != rnum){

                ra <<" "<< rnum <<" "<< rname <<" "<< rage <<" "<< rcnic <<"\n";

             }

            else

            {

                cout<<"\n\n\t\t\t\tInsert Record"<<endl;

                cin.ignore();

                cout<<"\n\t Room Number   : ";

                getline(cin, rnum);

                cout<<"\n\t  Customer Name   : ";

                getline(cin, rname);

                cout<<"\n\t Customer Age : ";

                getline(cin, rage);

                cout<<"\n\t  Customer Cnic  : ";

                getline(cin, rcnic);

                ra << " " << rnum << " " << rname << " " << rage << " " << rcnic << "\n";

                foundr++;

            }

             roomDetaill >> rnum >> rname >> rage >> rcnic ;

        }

            if (foundr == 0)

            {

             cout << "\n\t\t\t "<<num<<" Room Number Not Found....";

            }

         ra.close();

         roomDetaill.close();

         remove("Room\_Details.txt");

         rename("record.txt", "Room\_Details.txt");

    }

    }

 void roomClass::r\_delete(fstream& roomDetaill)              //room delete

{

     system("cls");

     fstream  ra;

     int founds = 0;

     string roomno;

     roomDetaill.open("Room\_Details.txt", ios::in);

    if(!roomDetaill)

    {

        cout <<"\n\t\t\tNo Room Detail File Is Present...";

        roomDetaill.close();

    }

    else

    {

        cin.ignore();

        cout << "\nEnter Room Number of Customer for Delete: ";

         getline(cin,roomno);

        ra.open("record.txt", ios::app | ios::out);

        roomDetaill >> rnum >> rname >> rage >> rcnic ;

        while (!roomDetaill.eof())

        {

          if (roomno != rnum)

          {

            ra << " " << rnum << " " << rname << " " << rage << " " << rcnic << "\n";

          }else

          {

            founds++;

            cout << "\n\t\t\tSuccessfully "<<roomno<< " Room Number of Data Deleted";

          }

        roomDetaill >> rnum >> rname >> rage >> rcnic ;

        }

        if(founds == 0)

        {

            cout<<"\n\t\t\t Customer "<<roomno <<" Room Number Not Found....";

        }

         ra.close();

         roomDetaill.close();

         remove("Room\_Details.txt");

         rename("record.txt", "Room\_Details.txt");

    }

}

**BILL.H:**

#ifndef BILL\_H

#define BILL\_H

#include<fstream>

using namespace std;

class billPayment{                                                   //Display bill Payment

   private:

    string z;

     char b;

    public :

     void insertBill();

     void Displaybill();

     void deleteOrder();

};

#endif

**BILL.CPP:**

#include<iostream>

#include<string>

#include<windows.h>

#include"manager.h"

#include"staff.h"

#include"room.h"

#include"bill.h"

#include<fstream>

using namespace std;

    static int amount , count , ch\_K\_H , ch\_K\_F , s\_k\_f , s\_k\_h , fishf, fishh;

     void billPayment::insertBill(){

      while(true){

       g:

       system("cls");

       cout<<"\n\t\t\t\tOrder Your Food\n";

       cout<<"\n\tPress 1 For Chicken karhai full   800 PKR";

       cout<<"\n\tPress 2 For Chicken karhai half   400 PKR";

       cout<<"\n\tPress 3 For Sekh kabab 12 pcs     300 PKR";

       cout<<"\n\tPress 4 For Sekh kabab 6 pcs      150 PKR";

       cout<<"\n\tPress 5 For Fish 1 kg             900 PKR";

       cout<<"\n\tPress 6 For Fish 1/2 kg           450 PKR";

       cout<<"\n\tPress 7 For Enough Order\n";

       cin>>b;

       if(b=='1')

         {

          if(count<=50){

          amount = amount + 800 ;

          count = count + 1;

          ch\_K\_F=ch\_K\_F+1;

          cout<<"\nOrder Successfull";

          Sleep(1000);

          }else

          {

          cout<<"No More Full Chicken Karhai Available!"<<endl;

          Sleep(1000);

          }

          }else if(b=='2')

          {

          if(count<=50)

          {

          amount = amount + 400 ;

          count = count + 1;

          ch\_K\_H = ch\_K\_H+1;

          cout<<"\nOrder Successfull";

          Sleep(1000);

          }else

          {

          cout<<"No more Half Chicken Karhai Available!"<<endl;

          Sleep(1000);

          }

          }else if(b=='3')

          {

          if(count<=50){

          amount = amount + 300 ;

          count = count + 1;

          s\_k\_f=s\_k\_f+1;

          cout<<"\nOrder Successfull";

          Sleep(1000);

          }else

          {

          cout<<"No more Full Plate Seekh Kabab Available !"<<endl;

          Sleep(1000);

          }

          }else if(b=='4')

          {

          if(count<=50)

          {

          amount = amount + 150 ;

          count = count + 1;

          s\_k\_h=s\_k\_h+1;

          cout<<"\nOrder Successfull";

          Sleep(1000);

          }else

          {

          cout<<"No more Half Plate Seekh Kabab Available !"<<endl;

          Sleep(1000);

          }

          }else if(b=='5')

          {

          if(count<=50)

          {

          amount = amount + 900 ;

          count = count + 1;

          fishf=fishf+1;

          cout<<"\nOrder Successfull";

          Sleep(1000);

          }else

          {

          cout<<"No more 1KG Fish Available !"<<endl;

          Sleep(1000);

          }

          }else if(b=='6')

          {

          if(count<=50)

          {

          amount = amount + 450 ;

          count = count + 1;

          fishh=fishh+1;

          cout<<"\nOrder Successfull";

          Sleep(1000);

          }else

          {

          cout<<"No more 1/2 KG Fish Available !"<<endl;

          Sleep(1000);

          }

          }else if(b=='7')

          {

             goto t;

          }else

          {

             cout<<"\n\tPress Correct Number\n";

             Sleep(3000);

             goto g;

          }

          }

          t:

          cin.ignore();

      }

     void billPayment::Displaybill(){

         system("cls");

         if((ch\_K\_F || ch\_K\_H || s\_k\_f || s\_k\_h || fishf || fishh || amount) > 0){

             cout<<"\n\t\t\t\tDisplay Payment\n";

             Sleep(800);

             cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

             cout<<"Number of Chicken karhai full = "<<ch\_K\_F<<endl;

             cout<<"Number of Chicken karhai half = "<<ch\_K\_H<<endl;

             cout<<"Number of Sekh kabab 12 pcs   = "<<s\_k\_f<<endl;

             cout<<"Number of Sekh kabab 6 pcs    = "<<s\_k\_h<<endl;

             cout<<"Number of Fish 1 kg           = "<<fishf<<endl;

             cout<<"Number of Fish 1/2 kg         = "<<fishh<<endl;

             cout<<"Total amount                  = "<<amount<<endl;

             cout<<"\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"<<endl;

          }else{

             cout<<"\n\tPayment Bill is Empty!";

          }

     }

     void billPayment::deleteOrder(){

    while(true){

          g:

         system("cls");

       cout<<"\n\t\t\tYour pending Orders\n";

       cout<<"\n\tChicken karhai Full :"<<ch\_K\_F;

       cout<<"\n\tChicken karhai Half :"<<ch\_K\_H;

       cout<<"\n\t Sekh kabab 12 pcs  :"<<s\_k\_f;

       cout<<"\n\t  Sekh kabab 6 pcs  :"<<s\_k\_h;

       cout<<"\n\t    Fish 1 kg       :"<<fishf;

       cout<<"\n\t   Fish 1/2 kg      :"<<fishh;

       cout<<"\n\t   Total amount     :"<<amount;

       cout<<"\n\n\t\t\t\tDelete Your Food\n";

       cout<<"\n\tPress 1 For Chicken karhai full ";

       cout<<"\n\tPress 2 For Chicken karhai half ";

       cout<<"\n\tPress 3 For Sekh kabab 12 pcs   ";

       cout<<"\n\tPress 4 For Sekh kabab 6 pcs    ";

       cout<<"\n\tPress 5 For Fish 1 kg           ";

       cout<<"\n\tPress 6 For Fish 1/2 kg         ";

       cout<<"\n\tPress 7 For Enough Delete Order\n";

       cin>>b;

       if(b=='1'){

          if(ch\_K\_F>0){

          amount = amount - 800 ;

          count = count - 1;

          ch\_K\_F=ch\_K\_F-1;

          cout<<"\nOrder Cancel Successfull";

          Sleep(1000);

          }else{

          cout<<"No Pending Full Chicken Karhai Available!"<<endl;

          cout<<"\n\n\tPress any key for Continue\n";

          cin>>z;

          }

          }else if(b=='2'){

          if(ch\_K\_H>0){

          amount = amount - 400 ;

          count = count - 1;

          ch\_K\_H = ch\_K\_H-1;

          cout<<"\nOrder Cancel Successfull";

          Sleep(1000);

          }else{

          cout<<"No Pending Half Chicken Karhai Available!"<<endl;

          cout<<"\n\n\tPress any key for Continue\n";

          cin>>z;

          }

          }else if(b=='3'){

          if(s\_k\_f>0){

          amount = amount - 300 ;

          count = count - 1;

          s\_k\_f=s\_k\_f-1;

          cout<<"\nOrder Cancel Successfull";

          Sleep(1000);

          }else{

          cout<<"No Pending Full Plate Seekh Kabab Available !"<<endl;

          cout<<"\n\n\tPress any key for Continue\n";

          cin>>z;

          }

          }else if(b=='4'){

          if(s\_k\_h>0){

          amount = amount - 150 ;

          count = count - 1;

          s\_k\_h=s\_k\_h-1;

          cout<<"\nOrder Cancel Successfull";

          Sleep(1000);

          }else{

          cout<<"No Pending Half Plate Seekh Kabab Available !"<<endl;

          cout<<"\n\n\tPress any key for Continue\n";

          cin>>z;

          }

          }else if(b=='5'){

          if(fishf>0){

          amount = amount - 900 ;

          count = count - 1;

          fishf=fishf-1;

          cout<<"\nOrder Cancel Successfull";

          Sleep(1000);

          }else{

          cout<<"No Pending 1KG Fish Available !"<<endl;

          cout<<"\n\n\tPress any key for Continue\n";

          cin>>z;

          }

          }else if(b=='6'){

          if(fishh>0){

          amount = amount - 450 ;

          count = count - 1;

          fishh=fishh-1;

          cout<<"\nOrder Cancel Successfull";

          Sleep(1000);

          }else{

          cout<<"No Pending 1/2 KG Fish Available !"<<endl;

          cout<<"\n\n\tPress any key for Continue\n";

          cin>>z;

          }

          }else if(b=='7'){

             goto t;

          }else{

             cout<<"\n\tPress Correct Number\n";

             Sleep(3000);

             goto g;

          }

          }

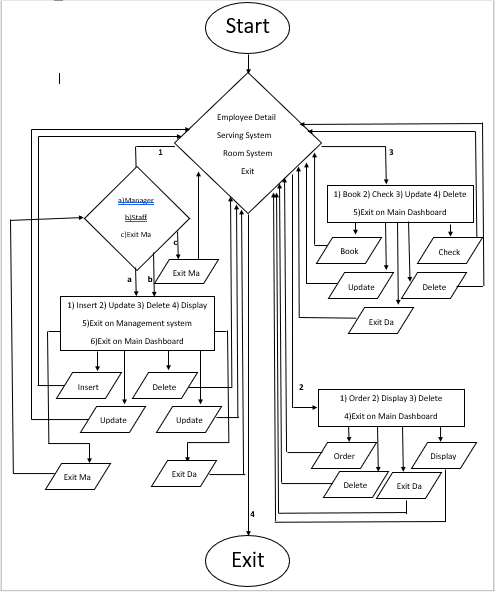
          system("cls");

          t:

          cin.ignore();

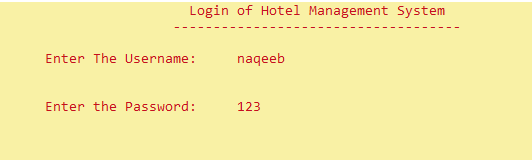
       }

* **Flow Chart:**

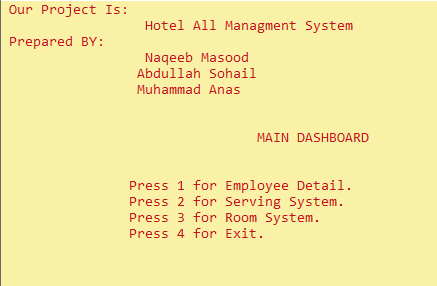


* **Screen Shots:**

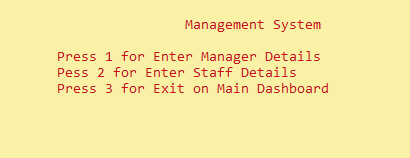
**LoginPage**



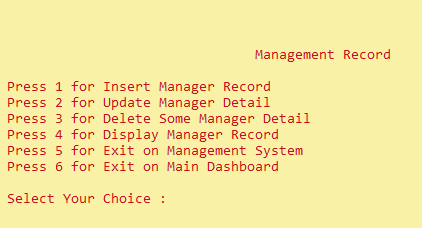
**MainDashBoard**



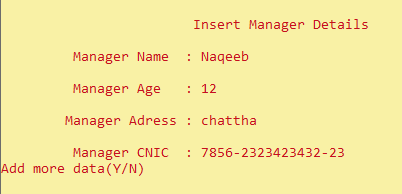
**Employee Detail**



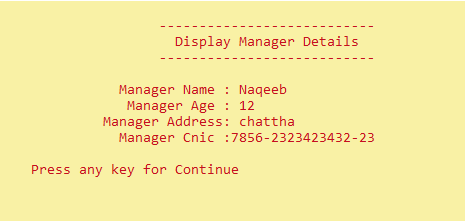
**Manager Detail**



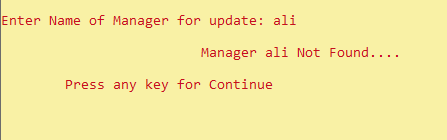
**Insert Manager Record**

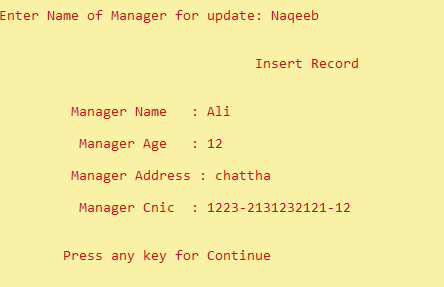


**Display Manager Detail**

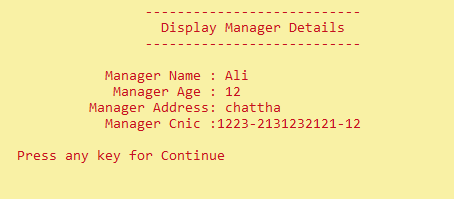


**Update Manager Detail**

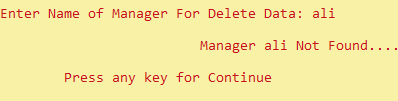


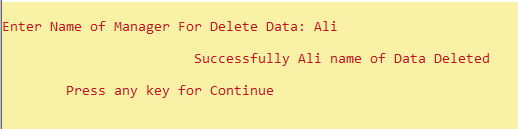


**Again Display After Update**

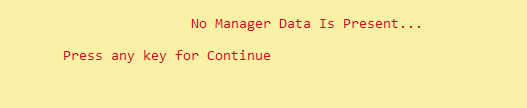


**Delete Manager Detail**

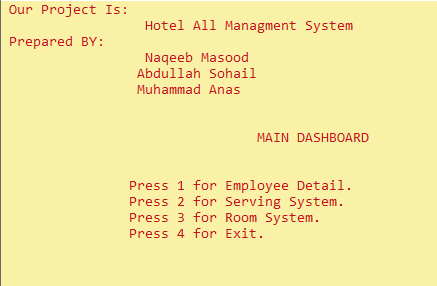




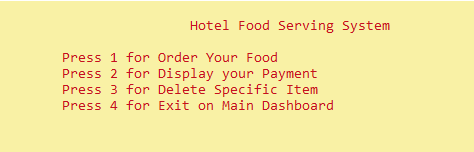
**Again Display After Update**



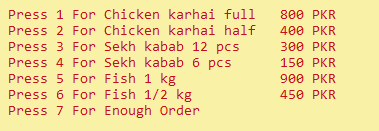
**MainDashBoard**



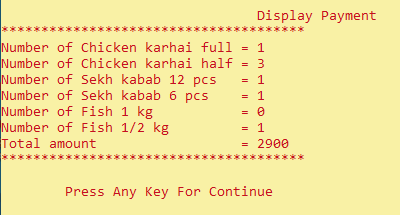
**Serving System**



**Order Your Food**



**Display Your Payment**



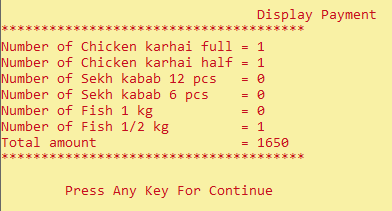
**Delete Specific Item**



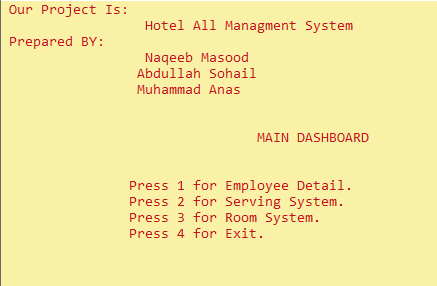
**After Delete Specific Order**



**Display After Update**



**MainDashBoard**



**Room System**

