BARRET HODGSON UNIVERSITY

Project Report

\*\*\*Hotel Management System\*\*\*

Members:

* Shafiq-Ur-Rehman
* Ali-Salman
* Abdur-Rasheed
* Faraz-Tariq
* Anzal-Abbas

In The Guidance Of:

* Sir Anjum
* Miss Unaiza Shahid

INTRODUCTION:

In the Software Requirements Specification SRS covers the Hotel Management System requirements needs. Hotel Management System is the main core appellation in this organization. The solution has to handle all of the major tasks of the hotel associated, operating, booking reservations, and Service guests, managing housekeeping, inventory management and including payroll management system. As well as responding effectively to problems and issues. The front desk staff will use this system The Software Requirements Specification SRS and the system which we are looking to develop has one goal is to make guests feel they are in an up to date and well run hotel to which they would like to return.

PROBLEM STATEMENT:

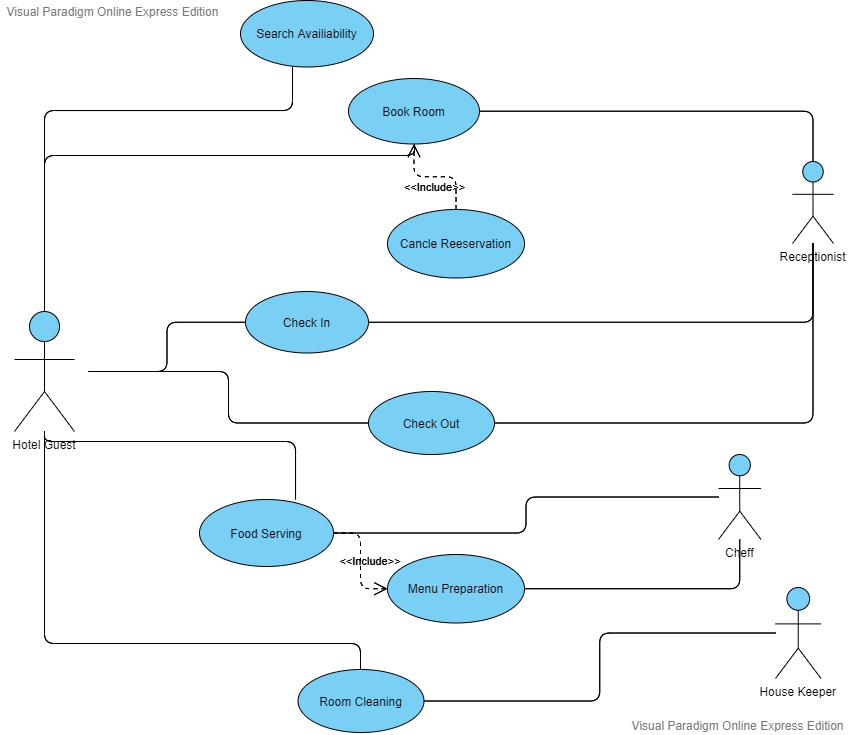
Hotel Management System Problem Statement,

A hotel system manages information about rooms, reservations, customers, and customer billing. A customer can make reservations, change, or cancel reservations through the hotel Program. When a customer makes reservations, he/she needs to check if a room the customer wants to reserve is available. If a room is available, the customer enters his/her information to the system and receives a confirmation number from the Program.

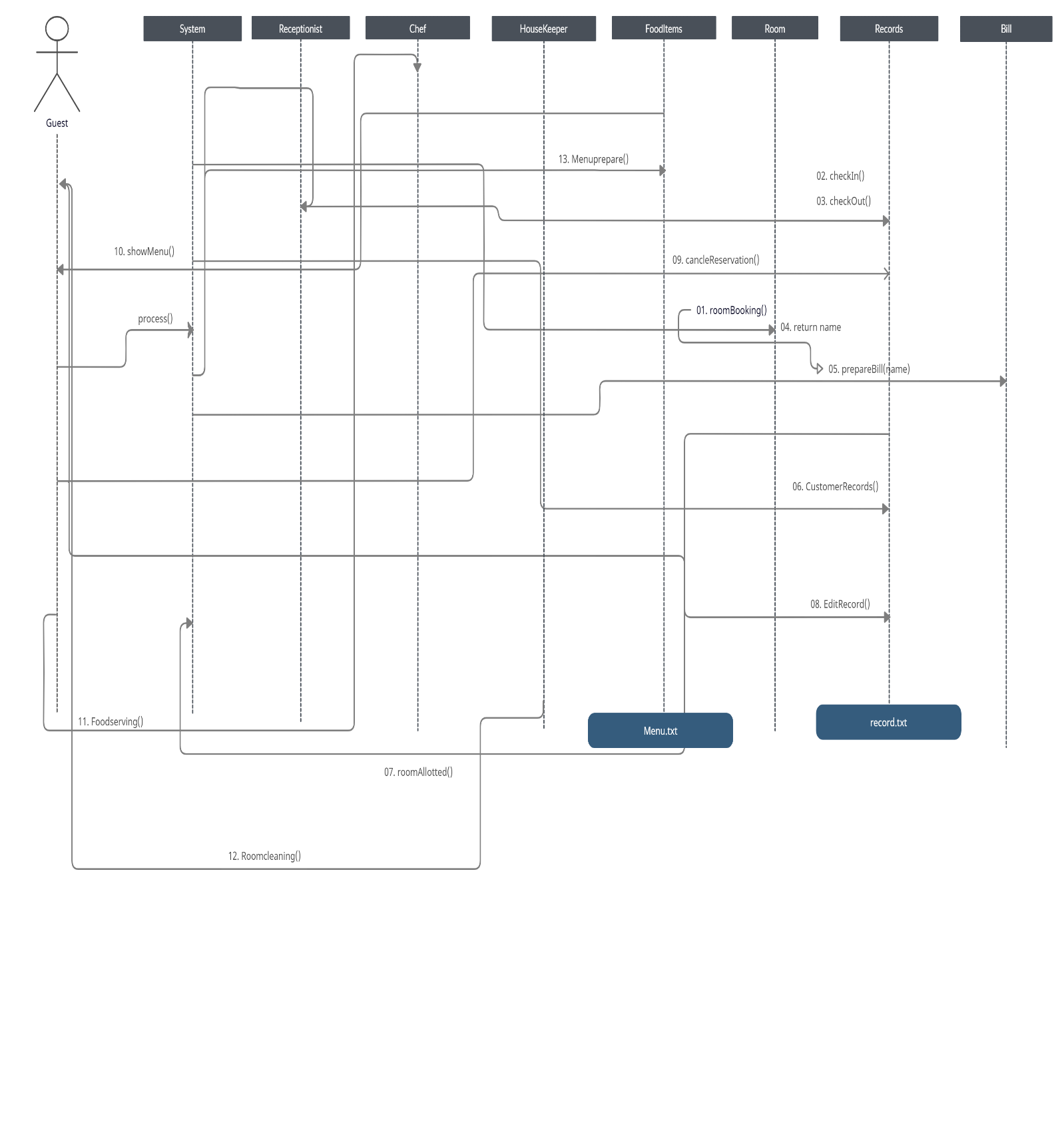
A desk clerk checks in a customer with only a prior reservation, change the checkout date, and check out the customer. A room is assigned to the customer at check-in time and a customer billing record is created at that time. The customer billing record is updated every night at 12. When a customer checks out, the desk clerk prints the bill. A customer can pay by cash, check, or credit card when he/she checks out.

* Develop the use case diagram showing actors and use cases for the hotel system.
* Describe Reserve Room use case using the template for use case description.

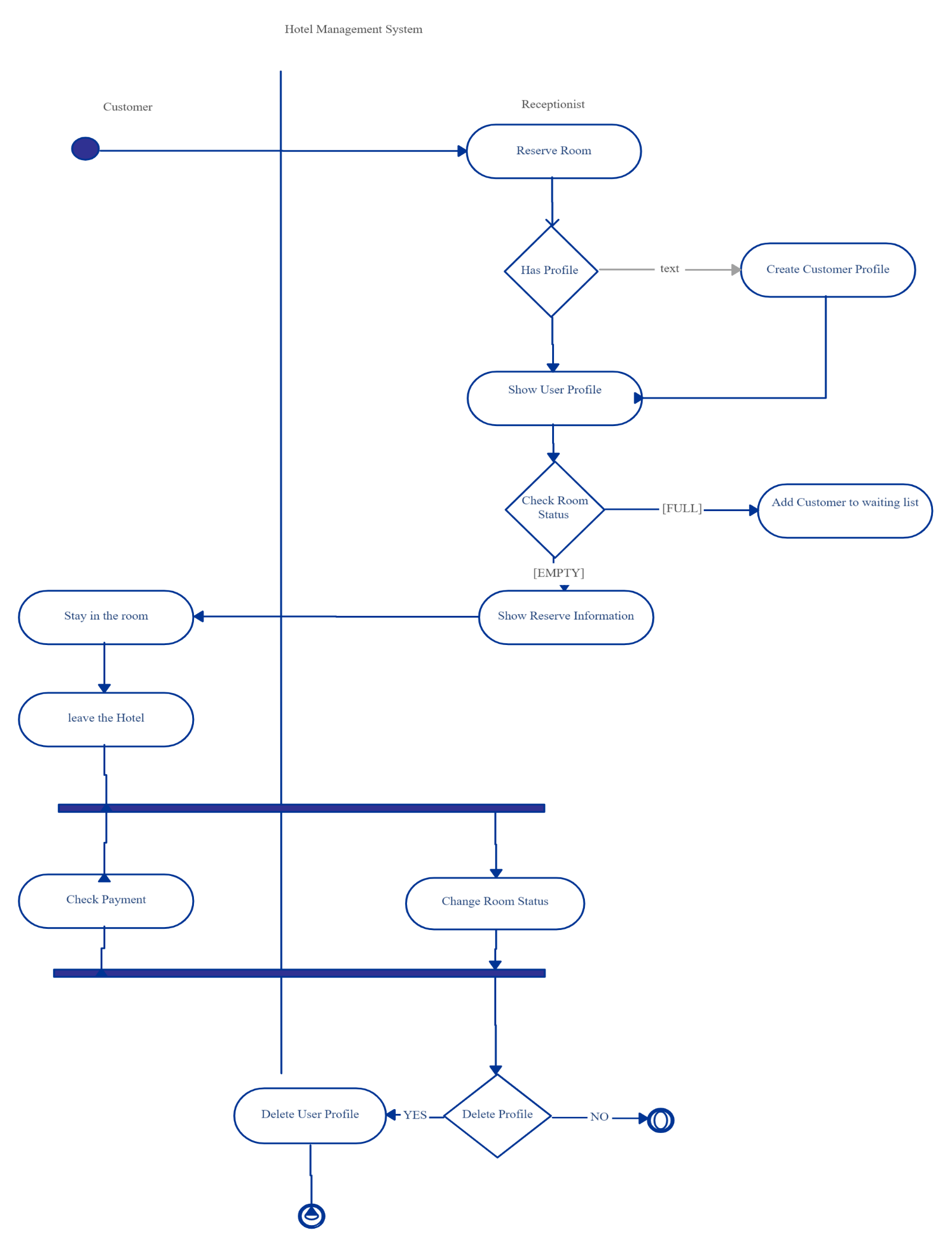
USE CASE DAIGRAM:



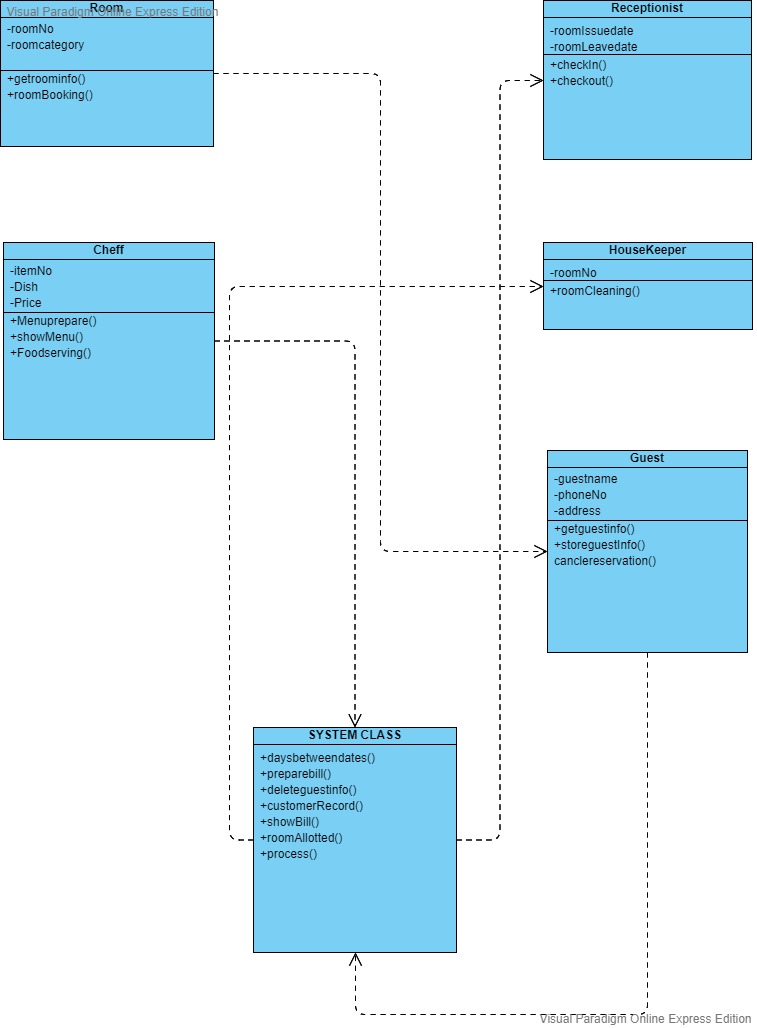
SEQUENCE DAIGRAM:



ACTIVITY DAIGRAM:



CLASS DAIGRAM:



COMPONENT DAIGRAM:

USER INTERFACE

HOTEL MANGEMENT SYSTEM

cancleReservation

deleteInfo

Record File

storeGuestInfo

getGuestInfo

GUEST

APPENDIC(--CODE--):

#include <iostream>

#include <windows.h>

#include <conio.h>

#include <stdlib.h>

#include <dirent.h>

#include <ctime>

#include <fstream>

#include <sstream>

#include <string>

#include <string.h>

using namespace std;

class Room

{

protected:

    string roomNo;

    string roomcategory;

public:

    void getRoomInfo()

    {

        system("cls");

        cout << "We have total number of room 50 including ordinary , luxuary and royal rooms" << endl;

        cout << "Enter Room NO: ";

        cin >> roomNo;

        cout << "Enter Room Category: ";

        cin >> roomcategory;

    }

    void roomBooking()

    {

    ROOM:

        getRoomInfo();

        // file handling

        string check1, check2, check3, check4, check5, check6, check7;

        ifstream checkInRecords;

        checkInRecords.open("Record.txt");

        checkInRecords >> check1;

        checkInRecords >> check2;

        checkInRecords >> check3;

        checkInRecords >> check4;

        checkInRecords >> check5;

        checkInRecords >> check6;

        checkInRecords >> check7;

        while (!checkInRecords.eof())

        {

            if (roomNo == check1)

            {

                cout << "This Room is Already Reserved , Try for another Room!!!....." << endl;

                cout << "Press Enter to Continue.....";

                getch();

                checkInRecords.close();

                goto ROOM;

            }

            checkInRecords >> check1;

            checkInRecords >> check2;

            checkInRecords >> check3;

            checkInRecords >> check4;

            checkInRecords >> check6;

            checkInRecords >> check7;

        }

        checkInRecords.close();

        ofstream roomRecord;

        roomRecord.open("Record.txt", ios ::app);

        roomRecord << roomNo;

        roomRecord << "\t\t\t";

        roomRecord << roomcategory;

        roomRecord << "\t\t\t\t";

        roomRecord.close();

    }

};

class Receptionist

{

protected:

    string roomIssueData;

    string roomLeaveData;

public:

    void checkIn()

    {

        // file pending

        time\_t t = time(NULL);

        tm \*tPtr = localtime(&t);

        roomIssueData = to\_string(tPtr->tm\_mday) + "/" + to\_string((tPtr->tm\_mon) + 1) + "/" + to\_string((tPtr->tm\_year) + 1900);

        ofstream IssueD;

        IssueD.open("Record.txt", ios ::app);

        IssueD << "\t\t\t";

        IssueD << roomIssueData;

        IssueD << endl;

        IssueD.close();

    }

    void checkout()

    {

        cout << "Enter Room Leave Data (\*GTM Format\*):";

        cin >> roomLeaveData;

        ofstream LeaveRoom;

        LeaveRoom.open("Record.txt", ios ::app);

        LeaveRoom << "\t";

        LeaveRoom << roomLeaveData;

        LeaveRoom.close();

        // file pending

    }

};

class HouseKeeper

{

protected:

    string roomNo;

public:

    void roomCleaning()

    {

        cout << "Enter Room number: ";

        cin >> roomNo;

        cout << "Processing: ";

        for (int i = 1; i <= 20; i++)

        {

            cout << "\xdb";

            Sleep(500);

        }

        cout << "\nYour Room Is Clean Now!";

    }

};

class Chef

{

protected:

    char itemNo[100];

    string Dish;

    string price;

public:

    void MenuPrepare()

    {

        ofstream prepare;

        prepare.open("Menu.txt");

        prepare << "ItemNO";

        prepare << "\t\t\t";

        prepare << "Dishes";

        prepare << "\t\t\t";

        prepare << "Prices";

        prepare << endl;

        cout << "Number's Of Dishes You want tu set? ";

        int noDishes;

        cin >> noDishes;

        for (int i = 1; i <= noDishes; i++)

        {

            prepare << i;

            prepare << "\t\t\t";

            cout << "Enter Dish name: ";

            cin >> Dish;

            cout << "Set Dish Price: ";

            cin >> price;

            prepare << Dish;

            prepare << "\t\t\t";

            prepare << price;

            prepare << endl;

        }

        prepare.close();

    }

    void showMenu()

    {

        //file handling

        string itemNo;

        ifstream myReadFile;

        myReadFile.open("Menu.txt");

        char output[200];

        if (myReadFile.is\_open())

        {

            int i = 0;

            myReadFile >> output;

            while (!myReadFile.eof())

            {

                if (i <= 2)

                {

                    cout << output << "\t\t\t";

                    myReadFile >> output;

                    i++;

                }

                if (i == 3)

                {

                    cout << endl;

                    i = 0;

                }

            }

        }

        myReadFile.close();

        cout << endl;

        cout << "Enter Item no to Order food!! >> ";

        cin >> itemNo;

    }

    void FoodServering()

    {

        cout << "Cooking in process: ";

        for (int i = 1; i <= 20; i++)

        {

            cout << "\xdb";

            Sleep(500);

        }

        cout << "\nServe!!";

    }

};

class HotelGuest : public Room

{

protected:

    string Guestname;

    string phoneNumber;

    string homeAddress;

public:

    void getGuestInfo()

    {

        cout << "Enter Your Name: ";

        cin >> Guestname;

        cout << "Enter Phone Number: ";

        cin >> phoneNumber;

        cout << "Enter Home address: ";

        cin >> homeAddress;

    }

    string storeGuestData()

    {

        getGuestInfo();

        //file handling

        ofstream guestInfo;

        guestInfo.open("Record.txt", ios ::app);

        guestInfo << Guestname;

        guestInfo << "\t\t\t";

        guestInfo << phoneNumber;

        guestInfo << "\t\t\t";

        guestInfo << homeAddress;

        guestInfo << "\t\t\t";

        guestInfo.close();

        return Guestname;

    }

    void cancleReservation()

    {

        // file pending

        string check, roomNo, roomcategory, Guestname, phoneNumber, homeAddress, roomLeaveData, roomIssueData;

        cout << "Enter Your Room No to Cancle Your Reservation! : ";

        cin >> check;

        ifstream DltRecord;

        DltRecord.open("Record.txt");

        ofstream NewRecord;

        NewRecord.open("Record1.txt", ios ::app);

        DltRecord >> roomNo;

        DltRecord >> roomcategory;

        DltRecord >> Guestname;

        DltRecord >> phoneNumber;

        DltRecord >> homeAddress;

        DltRecord >> roomLeaveData;

        DltRecord >> roomIssueData;

        while (!DltRecord.eof())

        {

            if (check != roomNo)

            {

                NewRecord << roomNo;

                NewRecord << "\t\t";

                NewRecord << roomcategory;

                NewRecord << "\t\t";

                NewRecord << Guestname;

                NewRecord << "\t\t";

                NewRecord << phoneNumber;

                NewRecord << "\t\t";

                NewRecord << homeAddress;

                NewRecord << "\t\t";

                NewRecord << roomLeaveData;

                NewRecord << "\t\t";

                NewRecord << roomIssueData;

                NewRecord << endl;

            }

            DltRecord >> roomNo;

            DltRecord >> roomcategory;

            DltRecord >> Guestname;

            DltRecord >> phoneNumber;

            DltRecord >> homeAddress;

            DltRecord >> roomLeaveData;

            DltRecord >> roomIssueData;

        }

        DltRecord.close();

        NewRecord.close();

        remove("Record.txt");

        rename("Record1.txt", "Record.txt");

    }

};

class System : public Receptionist, public HouseKeeper, public Chef, public HotelGuest

{

public:

    System()

    {

        ofstream hotelRecord;

        DIR \*directory;   // creating pointer of type dirent

        struct dirent \*x; // pointer represent directory stream

        string s = "Record.txt";

        bool result = false; //declaring string variable and assign it to false.

        if ((directory = opendir("./")) != NULL)

        { // check if directory  open

            while ((x = readdir(directory)) != NULL)

            {

                {

                    if (s == x->d\_name)

                    {

                        result = true; //if file found then  assign  result to false.

                        break; // break the loop if file found.

                    }

                }

            }

            closedir(directory); //close directory....

            if (result == false)

            {

                hotelRecord.open("Record.txt");

                hotelRecord << "Room-no";

                hotelRecord << "\t\t\t";

                hotelRecord << "Room-category";

                hotelRecord << "\t\t\t";

                hotelRecord << "Name";

                hotelRecord << "\t\t\t";

                hotelRecord << "Phone-No";

                hotelRecord << "\t\t\t";

                hotelRecord << "Addresss";

                hotelRecord << "\t\t\t";

                hotelRecord << "Leave-date";

                hotelRecord << "\t\t\t";

                hotelRecord << "Issue-date";

                hotelRecord << endl;

                hotelRecord.close();

            }

        }

    }

    int daysBetweenDates(string date1, string date2)

    {

        stringstream ss(date1 + "-" + date2);

        int year, month, day;

        char hyphen;

        // Parse the first date into seconds

        ss >> year >> hyphen >> month >> hyphen >> day;

        struct tm starttm = {0, 0, 0, day,

                             month - 1, year - 1900};

        time\_t start = mktime(&starttm);

        // Parse the second date into seconds

        ss >> hyphen >> year >> hyphen >> month >> hyphen >> day;

        struct tm endtm = {0, 0, 0, day,

                           month - 1, year - 1900};

        time\_t end = mktime(&endtm);

        // Find out the difference and divide it

        // by 86400 to get the number of days

        return abs(end - start) / 86400;

    }

    void prepareBill(string n)

    {

        string calcdate1, calcdate2, name, roomNo, roomcategory, Guestname, phoneNumber, homeAddress, roomLeaveData, roomIssueData;

        name = n;

        int days;

        ifstream getrecord;

        getrecord.open("Record.txt");

        getrecord >> roomNo;

        getrecord >> roomcategory;

        getrecord >> Guestname;

        getrecord >> phoneNumber;

        getrecord >> homeAddress;

        getrecord >> roomLeaveData;

        getrecord >> roomIssueData;

        while (!getrecord.eof())

        {

            if (name == Guestname)

            {

                ofstream guestbill;

                guestbill.open(name + "bill.txt");

                guestbill << "Name: " << name;

                guestbill << endl;

                guestbill << "Number: " << phoneNumber;

                guestbill << endl;

                guestbill << "Address: " << homeAddress;

                guestbill << endl;

                guestbill << "Issue-date: " << roomIssueData;

                guestbill << endl;

                guestbill << "Leave-date : " << roomLeaveData;

                string getmonth, getmonth2;

                getmonth = roomIssueData.substr(3, 4);

                getmonth = getmonth.substr(0, 2);

                getmonth2 = roomLeaveData.substr(3, 4);

                getmonth2 = getmonth2.substr(0, 2);

                calcdate1 = roomIssueData.substr(6, 9) + "-" + getmonth + "-" + roomIssueData.substr(0, 2);

                calcdate2 = roomLeaveData.substr(6, 9) + "-" + getmonth2 + "-" + roomLeaveData.substr(0, 2);

                days = daysBetweenDates(calcdate2, calcdate1);

                cout << endl;

                cout << endl;

                if (roomcategory == "ac" || roomcategory == "Ac" || roomcategory == "AC")

                {

                    guestbill << endl;

                    guestbill << "Amount: " << days \* 1200;

                }

                else

                {

                    guestbill << endl;

                    guestbill << "Amount: " << days \* 600;

                }

                guestbill.close();

            }

            getrecord >> roomNo;

            getrecord >> roomcategory;

            getrecord >> Guestname;

            getrecord >> phoneNumber;

            getrecord >> homeAddress;

            getrecord >> roomLeaveData;

            getrecord >> roomIssueData;

        }

        getrecord.close();

    }

    void deleteGuestInfo()

    {

        //file handling

        string check, roomNo, roomcategory, Guestname, phoneNumber, homeAddress, roomLeaveData, roomIssueData;

        cout << "Enter Your Room No to Edit Your Record..! : ";

        cin >> check;

        ifstream DltRecord;

        DltRecord.open("Record.txt");

        ofstream NewRecord;

        NewRecord.open("Record1.txt");

        DltRecord >> roomNo;

        cout << roomNo;

        DltRecord >> roomcategory;

        DltRecord >> Guestname;

        DltRecord >> phoneNumber;

        DltRecord >> homeAddress;

        DltRecord >> roomLeaveData;

        DltRecord >> roomIssueData;

        cout << "check" << endl;

        while (!DltRecord.eof())

        {

            cout << "check" << endl;

            if (check == roomNo)

            {

                cout << "Enter your New room no>> ";

                cin >> roomNo;

                cout << "Enter your Room category>> ";

                cin >> roomcategory;

                cout << "Enter your Name>> ";

                cin >> Guestname;

                cout << "Enter your phone number>> ";

                cin >> phoneNumber;

                cout << "Enter your home Address>> ";

                cin >> homeAddress;

                cout << "Enter Leave date>> ";

                cin >> roomLeaveData;

                cout << "Enter current date>> ";

                cin >> roomIssueData;

                NewRecord << roomNo;

                NewRecord << "\t\t";

                NewRecord << roomcategory;

                NewRecord << "\t\t";

                NewRecord << Guestname;

                NewRecord << "\t\t";

                NewRecord << phoneNumber;

                NewRecord << "\t\t";

                NewRecord << homeAddress;

                NewRecord << "\t\t";

                NewRecord << roomLeaveData;

                NewRecord << "\t\t";

                NewRecord << roomIssueData;

                NewRecord << endl;

            }

            else

            {

                NewRecord << roomNo;

                NewRecord << "\t\t";

                NewRecord << roomcategory;

                NewRecord << "\t\t";

                NewRecord << Guestname;

                NewRecord << "\t\t";

                NewRecord << phoneNumber;

                NewRecord << "\t\t";

                NewRecord << homeAddress;

                NewRecord << "\t\t";

                NewRecord << roomLeaveData;

                NewRecord << "\t\t";

                NewRecord << roomIssueData;

                NewRecord << endl;

            }

            DltRecord >> roomNo;

            DltRecord >> roomcategory;

            DltRecord >> Guestname;

            DltRecord >> phoneNumber;

            DltRecord >> homeAddress;

            DltRecord >> roomLeaveData;

            DltRecord >> roomIssueData;

        }

        DltRecord.close();

        NewRecord.close();

        remove("Record.txt");

        rename("Record1.txt", "Record.txt");

    }

    void customerRecord()

    {

        string check, roomNo, roomcategory, Guestname, phoneNumber, homeAddress, roomLeaveData, roomIssueData;

        ifstream showRecords;

        showRecords.open("Record.txt");

        cout << "Enter Room No. : ";

        cin >> check;

        while (!showRecords.eof())

        {

            showRecords >> roomNo;

            showRecords >> roomcategory;

            showRecords >> Guestname;

            showRecords >> phoneNumber;

            showRecords >> homeAddress;

            showRecords >> roomLeaveData;

            showRecords >> roomIssueData;

            if (roomNo == check)

            {

                system("cls");

                cout << "Room-no\t\t\tRoom-Category\t\t\tName\t\t\tphone-NO\t\t\tAddress\t\t\tLeave-Date\t\t\tIssue-Date" << endl;

                cout << roomNo << "\t\t\t" << roomcategory << "\t\t\t" << Guestname << "\t\t\t" << phoneNumber << "\t\t\t" << homeAddress << "\t\t\t" << roomLeaveData << "\t\t\t" << roomIssueData << endl;

                cout << endl;

                cout << endl;

                cout << "Press Enter to Continue......!";

                getch();

                break;

            }

        }

        if (roomNo != check)

        {

            cout << "Record is Not Found!!!!" << endl;

            cout << "Press Enter to Continue......!";

            getch();

        }

        showRecords.close();

    }

    void showBill()

    {

        cout << "Enter Guest Name: ";

        string name, gname[3], num[3], add[3], issD[3], LD[3], amn[3];

        cin >> name;

        ifstream Show;

        Show.open(name + "bill.txt");

        Show >> gname[0];

        Show >> gname[1];

        Show >> num[0];

        Show >> num[1];

        Show >> add[0];

        Show >> add[1];

        Show >> issD[0];

        Show >> issD[1];

        Show >> LD[0];

        Show >> LD[1];

        Show >> amn[0];

        Show >> amn[1];

        Show >> amn[2];

        Show.close();

        cout << gname[0];

        cout << " ";

        cout << gname[1];

        cout << endl;

        cout << num[0];

        cout << " ";

        cout << num[1];

        cout << endl;

        cout << add[0];

        cout << " ";

        cout << add[1];

        cout << endl;

        cout << issD[0];

        cout << " ";

        cout << issD[1];

        cout << endl;

        cout << LD[0];

        cout << " ";

        cout << LD[1];

        cout << endl;

        cout << amn[0];

        cout << " ";

        cout << endl;

        cout << amn[1];

        cout << " ";

        cout << amn[2];

        cout << endl;

    }

    void roomAllotted()

    {

        string roomNo, roomcategory, Guestname, phoneNumber, homeAddress, roomLeaveData, roomIssueData;

        ifstream showRecords;

        showRecords.open("Record.txt");

        system("cls");

        showRecords >> roomNo;

        showRecords >> roomcategory;

        showRecords >> Guestname;

        showRecords >> phoneNumber;

        showRecords >> homeAddress;

        showRecords >> roomLeaveData;

        showRecords >> roomIssueData;

        while (!showRecords.eof())

        {

            cout << roomNo << "\t\t\t" << roomcategory << "\t\t\t" << Guestname << "\t\t\t" << phoneNumber << "\t\t\t" << homeAddress << "\t\t\t" << roomLeaveData << "\t\t\t" << roomIssueData << endl;

            showRecords >> roomNo;

            showRecords >> roomcategory;

            showRecords >> Guestname;

            showRecords >> phoneNumber;

            showRecords >> homeAddress;

            showRecords >> roomLeaveData;

            showRecords >> roomIssueData;

        }

        cout << endl;

        cout << endl;

        cout << "Press Enter to Continue......!";

        getch();

        showRecords.close();

    }

    void process()

    {

        int loops = 0;

        int options;

        while (loops == 0)

        {

        Startup:

            string selectedItem, name;

            int price;

            cout << endl;

            cout << "1. Book A Room." << endl;

            cout << "2. Customer Records." << endl;

            cout << "3. Room Allotted." << endl;

            cout << "4. Edit Record." << endl;

            cout << "5. Guest Portal." << endl;

            cout << "6. chef Portal." << endl;

            cout << "7. billing." << endl;

            cout << "8. Exist." << endl;

            cout << endl;

            cout << "Enter your choice >> ";

            cin >> options;

            switch (options)

            {

            case 1:

                roomBooking();

                name = storeGuestData();

                checkout();

                checkIn();

                prepareBill(name);

                cout << "Press Enter to Continue.....";

                getch();

                system("cls");

                break;

            case 2:

                system("cls");

                customerRecord();

                system("cls");

                break;

            case 3:

                system("cls");

                roomAllotted();

                system("cls");

                break;

            case 4:

                system("cls");

                deleteGuestInfo();

                cout << "Press Enter to continue......";

                getch();

                system("cls");

                break;

            case 5:

                system("cls");

                int choice;

                cout << "1. Cancle Reservation" << endl;

                cout << "2. Food order" << endl;

                cout << "3. House Keeper" << endl;

                cout << "4. Exit";

                cout << endl;

                cout << endl;

                cout << "Enter your choice?  ";

                cin >> choice;

                switch (choice)

                {

                case 1:

                    system("cls");

                    cancleReservation();

                    system("cls");

                    break;

                case 2:

                    system("cls");

                    showMenu();

                    FoodServering();

                    cout << endl;

                    cout << "Press enter to continue.....";

                    getch();

                    system("cls");

                    break;

                case 3:

                    roomCleaning();

                    cout << "Press enter to continue.....";

                    getch();

                    system("cls");

                    break;

                default:

                    exit(0);

                    break;

                }

                break;

            case 6:

                system("cls");

                MenuPrepare();

                system("cls");

                cout << "Press enter to continue.....";

                getch();

                system("cls");

                break;

            case 7:

                system("cls");

                showBill();

                cout << "Press enter to continue.....";

                getch();

                system("cls");

                break;

            default:

                exit(0);

                break;

            }

        }

    }

};

int main()

{

    System s;

    cout << endl;

    cout << endl;

    cout << "\t\t\t\t\t\*\*\*\*\*Hotel Management System\*\*\*\*\*";

    cout << endl;

    s.process();

}

CONCLUSION & SUMARY:

Hotel management system now-a-day have the advantage of modernization. Computer have done the work more easy. Computer is playing a important role in management. Reports are made on daily basis for every customer check in or check out which can easily be seen by the management. Hotel management system has also primary purpose is to provide facilities to customers. A software for computers makes the things many times easy, these are made as user friendly and to keep a check and balance in hotel management and accounts as well. So, these things are important.