Write a project report in a Word file “Project\_Yourinitials.docx”, which contains:

1. the problem statement (5%)

A local hotel has asked us to create a system to book their guests into their rooms as part of their modernization strategy. This requires that we are able to book guests, check room availability, and process proper receipts for payment.

1. hierarchy chart, flowchart, and pseudocode of your design (15%)

PSUEDOCODE///////////////////////////////////////////////////////////////////////////////////////////

Start

// Ayden Bishop, Zach Tarske, and Joshua Bockerstette

void userSignIn()

void userSignUp()

void Room\_Type()

int Receipt(int, int, int, int)

bool getOccupied()

void pickRoom(int)

const int MAX\_ROOMS = 10

const int ROOM\_TYPES = 3

int floor1[MAX\_ROOMS][ROOM\_TYPES]

int floor2[MAX\_ROOMS][ROOM\_TYPES]

int floor3[MAX\_ROOMS][ROOM\_TYPES]

bool occupied[MAX\_ROOMS]

const int A\_PRICE = 80

const int B\_PRICE = 150

const int C\_PRICE = 175

Main ()

Declare char choice

Declare char again

do {

Display "------------------------------" \t

Display setw(16) << "Menu:" \t \t

Display "a) User Sign In" \t

Display "b) User Sign Up" \t

Display "------------------------------" \t

Input choice

}

while (choice != 'a' && choice != 'b' && choice != 'c')

{

"Error: Please choose either 'a' or 'b'"

Input choice

}

switch (tolower(choice)) {

case 'a':

userSignIn()

break

case 'b':

userSignUp()

break

}

Display "(y/n)"

Input again

} while(tolower(again) == 'y')

return

}

Void Pick\_Dates ()

{

int Checkin\_Date

int Checkout\_Date

Display "Input check in date"

Input Checkin\_Date

while(Checkin\_Date < 1 && Checkin\_Date > 30)

{

Display "Error: The date has to be in the range of 1-30"

Input Checkin\_Date

}

Display "Input check out date"

Input Checkout\_Date

while(Checkout\_Date <= Checkin\_Date && Checkout\_Date > 30)

{

Display "Error: The date has to be in the range of " << (Checkin\_Date + 1) << "-30"

Input Checkout\_Date

}

}

END

void signup ()

declare username

declare password

display "Welcome to Signup for Hotel Overlook"

display " "

display " "

display "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

display "Please enter your wanted username"

input username

display "Please enter your wanted password"

input password

OPENFILE userinfo.txt

INSERT +username

INSERT +password

CLOSEFILE

Display "Welcome User"

END

Void Room\_Type ()

Declare int price;

Declare bool occupied;

Display “Choose Room Type (1, 2, 3): “

Display “1) Single (1 Bed 1 Bath)” \t “2) Double (2 Bed 1 Bath)” \t “3) Suite (3 Bed 2 Bath)” \t

Switch (room\_type)

a. ‘1’

Display “You Picked the room type: Single Room (1 Bed 1 Bath)”

price = a\_price

occupied = getOccupied()

b. ‘2’

Display “You Picked the room type: Double (2 Bed 1 Bath)”

price = b\_price

occupied = getOccupied()

c. ‘3’

Display “You Picked the room type: Suite (3 Bed 2 Bath)”

price = c\_price

occupied = getOccupied()

Display "There are "+occupied " rooms available for your dates"

pickRoom(occupied)

END

int Receipt (price,total,checkin\_Date,checkout\_Date,username)

Display "--- Receipt ---"

Display "User:"+username

Display "Check In Date: " << "12/"<<checkin\_Date<<"/2024" \n

Display "Check Out Date: " << "12/"<<checkout\_Date<<"/2024" \n

Display "Room type price: $" + price

Display "Total nights: " + (checkout\_Date - checkin\_Date)

Display "Total amount due: $" + total

END

bool getOccupied()

Bool occupied

for (int i = 0; i < MAX\_ROOMS; i++)

{

Occupied = random() % 2

floor1[][]

floor2[][]

floor3[][]

}

}

void PickRoom()

declare choice

for (int row = 0; row <= MAX\_ROOMS; row++)

{

Display “Rooms: “ << hotel[][] << “ “

for (int column =

END

void signin ()

declare bool ValidUser

display "Please enter your username"

input username

display "Please enter your password"

input password

OPENFILE userinfo.txt

set VaildUser = Compare userinfo.txt to +username && +password

If VaildUser == true

Display "Welcome User"

END

void guest ()

declare username

declare password

Display "Welcome guest, no sign-in required"

END

int RandomNumber()

int occupied = random() %2

return occupied

END

1. source code (code 40%, programing writing style/documentation 5%)

#include <iostream>

#include <fstream>

#include <string>

#include <iomanip>

#include <cstdlib>

#include <ctime>

using namespace std;

const int A\_PRICE = 80;

const int B\_PRICE = 150;

const int C\_PRICE = 175;

const int MAX\_ROOMS = 10;

int hotel[3][MAX\_ROOMS]; // 3 types of rooms, 10 rooms each

int occupied[MAX\_ROOMS];  // Track the status of each room (0 = Available, 1 = Taken)

// Function declarations for the sign-up and sign-in features

bool validateUser(const string &filename, const string &username, const string &password);

bool isUsernameTaken(const string &filename, const string &username);

void signUp(const string &filename);

void signIn(const string &filename);

// Function for receipt

void Receipt(int price, int Checkin\_Date, int Checkout\_Date);

// Function for picking room number

void pickRoom(int occupied[], int MAX\_ROOMS);

// Function to initialize rooms randomly (Available/Taken)

void getOccupied();

int main() {

    const string filename = "Userinfo.txt";  // For sign-in/sign-up functionality

    char choice;

    char again;

    double total = 0;

    double price = 0;

    int Checkin\_Date;

    int Checkout\_Date;

    do {

        // Menu for sign in and sign up

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

        cout << setw(16) << "Menu:" << endl << endl;

        cout << "a) User Sign In" << endl;

        cout << "b) User Sign Up" << endl;

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

        cin >> choice;

        while(choice != 'a' && choice != 'b') {

            cout << "Error: Please choose either 'a' or 'b'" << endl;

            cin >> choice;

        }

        switch (tolower(choice)) {

            case 'a':

                signIn(filename);  // Handle sign-in

                break;

            case 'b':

                signUp(filename);  // Handle sign-up

                break;

        }

        // Getting check-in and check-out dates

        cout << "Input check-in date (1-30): ";

        cin >> Checkin\_Date;

        while (Checkin\_Date < 1 || Checkin\_Date > 30) {

            cout << "Error: The date has to be in the range of 1-30" << endl;

            cin >> Checkin\_Date;

        }

        cout << "Input check-out date (greater than check-in date, 1-30): ";

        cin >> Checkout\_Date;

        while (Checkout\_Date <= Checkin\_Date || Checkout\_Date > 30) {

            cout << "Error: The date has to be in the range of " << (Checkin\_Date + 1) << "-30" << endl;

            cin >> Checkout\_Date;

        }

        // Call Room Type and pick Room

        int room\_type;

        cout << "Choose room type:" << endl;

        cout << "1) Single (1 Bed 1 Bath)" << endl;

        cout << "2) Double (2 Beds 1 Bath)" << endl;

        cout << "3) Suite (3 Beds 2 Baths)" << endl;

        cin >> room\_type;

        while (room\_type != 1 && room\_type != 2 && room\_type != 3) {

            cout << "Error: Choose between the numbers 1, 2, or 3" << endl;

            cin >> room\_type;

        }

        switch(room\_type) {

            case 1:

                cout << "You picked the room type: Single (1 Bed 1 Bath)" << endl;

                price = A\_PRICE;

                break;

            case 2:

                cout << "You picked the room type: Double (2 Beds 1 Bath)" << endl;

                price = B\_PRICE;

                break;

            case 3:

                cout << "You picked the room type: Suite (3 Beds 2 Baths)" << endl;

                price = C\_PRICE;

                break;

        }

        // Randomly set room availability

        getOccupied();

        // Show room availability for the selected room type

        cout << "\nRoom Availability: " << endl;

        cout << "-----------------------------------" << endl;

        // Displaying room numbers with fixed width for alignment

        for (int i = 0; i < MAX\_ROOMS; i++) {

            cout << setw(7) << (i + 101);  // Set width to 7 for room numbers

        }

        cout << endl;

        // Displaying room availability status (Taken/Available) with fixed width for alignment

        for (int i = 0; i < MAX\_ROOMS; i++) {

            if (occupied[i] == 1)

                cout << setw(10) << "Taken";  // Set width to 10 for "Taken"

            else

                cout << setw(10) << "Available";  // Set width to 10 for "Available"

        }

        cout << endl << endl;

        // Allow the user to pick an available room

        int room\_number;

        cout << "Choose a room number from the available ones: ";

        cin >> room\_number;

        while (room\_number < 101 || room\_number > 110 || occupied[room\_number - 101] == 1) {

            cout << "Error: Please choose an available room number between 101 and 110." << endl;

            cin >> room\_number;

        }

        // Calculate subtotal

        double subtotal = price \* (Checkout\_Date - Checkin\_Date);

        // Calculate total with tax (8.25%)

        double total = subtotal \* 1.0825;

        // Call Receipt function to display details

        Receipt(price, Checkin\_Date, Checkout\_Date);

        cout << "(y/n) Do you want to continue? ";

        cin >> again;

    } while(tolower(again) == 'y');

    return 0;

}

// Function to print the receipt

void Receipt(int price, int Checkin\_Date, int Checkout\_Date) {

    double subtotal = price \* (Checkout\_Date - Checkin\_Date);

    double total = subtotal \* 1.0825;

    cout << "\n--- Receipt ---" << endl;

    cout << "Room type price: $" << price << endl;

    cout << "Total nights: " << (Checkout\_Date - Checkin\_Date) << endl;

    cout << "Subtotal: $" << fixed << setprecision(2) << subtotal << endl;

    cout << "Tax (8.25%): $" << fixed << setprecision(2) << subtotal \* 0.0825 << endl;

    cout << "Total amount due: $" << fixed << setprecision(2) << total << endl;

}

// Function to initialize room status randomly (Available/Taken)

void getOccupied() {

    srand(time(0));  // Seed for random number generation

    for (int i = 0; i < MAX\_ROOMS; i++) {

        occupied[i] = rand() % 2;  // Randomly assign each room as either Taken (1) or Available (0)

    }

}

// Function to validate user during sign-in

bool validateUser(const string &filename, const string &username, const string &password) {

    ifstream file(filename);

    if (!file.is\_open()) {

        cout << "Error opening file." << endl;

        return false;

    }

    string fileUsername, filePassword;

    while (file >> fileUsername >> filePassword) {

        if (fileUsername == username && filePassword == password) {

            return true;

        }

    }

    return false;

}

// Function to check if username is already taken during sign-up

bool isUsernameTaken(const string &filename, const string &username) {

    ifstream file(filename);

    if (!file.is\_open()) {

        cout << "Error opening file." << endl;

        return false;

    }

    string fileUsername, filePassword;

    while (file >> fileUsername >> filePassword) {

        if (fileUsername == username) {

            return true;

        }

    }

    return false;

}

// Function to sign up a new user

void signUp(const string &filename) {

    string username, password;

    cout << "Enter a new username: ";

    cin >> username;

    if (isUsernameTaken(filename, username)) {

        cout << "Username already exists. Please try again." << endl;

        return;

    }

    cout << "Enter a new password: ";

    cin >> password;

    ofstream file(filename, ios::app);

    if (!file.is\_open()) {

        cout << "Error opening file." << endl;

        return;

    }

    file << username << " " << password << endl;

    file.close();

    cout << "Sign up successful!" << endl;

}

// Function to sign in an existing user

void signIn(const string &filename) {

    string username, password;

    cout << "Enter your username: ";

    cin >> username;

    cout << "Enter your password: ";

    cin >> password;

    if (validateUser(filename, username, password)) {

        cout << "Sign in successful!" << endl;

    } else {

        cout << "Invalid username or password. Please try again." << endl;

    }

}

1. test plan (5 or more test cases) (10%)
2. summary (like/dislike/challenges/future improvements) (5%)