Basics of Programming 2

Student Name: Akhrikhar Yahya

Neptun Code: K20UXP

Lab Group ID: CS16B

Lab Instructor: Zsolt Kornél Dobos

Project Title: Hotel Management System

Type of Project: Final Project

Submission Date: 19-05-2024

# Introduction

The Hotel Management System project aims to develop a software solution for managing hotel operations efficiently. The system allows customers to book rooms and check in , managers to oversee reservations. This documentation provides a comprehensive overview of the program's functionality, implementation details, and usage instructions.

# Program Interface

To run the program, compile the provided C++ source code files (hotel.h, hotel.cpp, projectmain.cpp) using a C++ compiler such as g++ or Visual Studio. Execute the compiled binary file in the terminal or command prompt. The program interface includes menu-driven interactions where users can select options using numeric inputs.

To terminate the program, users can select the appropriate exit option from the menu. (Or click ctrl + c)

# Program Execution

## User Roles:

**Customer**: Can book rooms and check in.

**Manager**: Can manage reservations, view booked rooms, cancel reservations, and view checked-in rooms.

## Menu Options:

**Book a room**: Customers and managers can book available rooms.

**Check room availability**: Customers can view available rooms.

**Check-in:** Customers can check in to their reserved rooms.

**Cancel Room Reservation**: Managers can cancel room reservations.

**View checked-in rooms list**: Managers can view a list of currently checked-in rooms.

**Exit**: Terminate the program.

## Input and Output:

The program accepts user input through the terminal for menu selections, room bookings, check-ins, etc. It outputs relevant information and prompts for user interactions. Additionally, the program writes reservation details to text files (file.txt, checkedIn.txt, receipt.txt) for data persistence.

If the customer wants to book a room then first he must login as customer and then choose the first option which is choice 1 – book room and then he will have to choose the type of room, deluxe or standard and then the message "room booked successfully "will be printed.

The information of the room will be also printed in the receipt.txt, info such as room type, room number and price.

## Project Overview

The Hotel Management System is designed to streamline the management of hotel room reservations and customer check-ins. It supports functionality for both customers and managers, providing capabilities such as booking rooms, viewing room availability, checking in customers, and cancelling room reservations.

## Project Structure

The project consists of the following files:

**hotel.h:** Header file containing class definitions and function prototypes.

**hotel.cpp**: Implementation file containing the function definitions for the classes declared in hotel.h.

**projectmain.cpp**: Main program file that drives the application by initializing the hotel and invoking the login process.

## hotel.h

This file declares the classes and functions used in the hotel management system.

Classes

- **User**

Base class for Customer and Manager.

Attributes: username, password.

Methods:

User(const string& uname, const string& pwd): Constructor.

bool authenticate(const string& uname, const string& pwd) const: Authenticates the user.

string getUsername() const: Returns the username.

- **Customer**

Inherits from User.

Constructor: Customer(const string& uname, const string& pwd).

- **Manager**

Inherits from User.

Constructor: Manager(const string& uname, const string& pwd).

- **Hotel**

Attributes: hotelname, hoteladdress, hotelPhonenumber.

Methods:

Hotel(string name, string address, string num): Constructor.

void hotelinfoprint() const: Prints hotel information.

- **RoomReservation**

Attributes: standardRooms, deluxeRooms, bookedRooms, checkedInGuests.

Methods:

RoomReservation(int stdRooms, int dlxRooms): Constructor.

bool bookRoom(string roomType): Books a room.

bool checkInCustomer(const string& customerName, const string& roomNumber, const string& checkInDate, const string& checkOutDate): Checks in a customer.

void writeReceipt(const string& customerName, const string& roomNumber, const string& checkInDate, const string& checkOutDate, const string& roomType): Writes a receipt.

string promptRoomType(): Prompts the user for room type.

void printCheckedInRooms() const: Prints all checked-in rooms.

- **Functions**

bool isStringUnique(const string& filename, const string& newString): Checks if a string is unique in a file.

void viewrooms(): Displays all reserved rooms.

void CancelRoomReservation(): Cancels a room reservation.

void displayMenu(): Displays the customer menu.

void displayMenuManager(): Displays the manager menu.

void login(): Handles user login and menu navigation.

## hotel.cpp

This file contains the implementations of the methods declared in hotel.h.

**Detailed Functionality**

**User Authentication**

The system differentiates between Customer and Manager based on their credentials. The login function handles the authentication process and directs users to the appropriate menu.

**Room Booking**

Users can book rooms by specifying the room type (standard or deluxe). The RoomReservation class manages room availability and booking processes. It ensures rooms are uniquely booked and updates the booking information in a file.

**Check-In Process**

Customers can check in by providing their room number and check-in/out dates. The system verifies the room number against booked rooms and updates the check-in status, maintaining a record of checked-in guests.

**Room Cancellation**

Managers can cancel room reservations, making the rooms available for booking again. The CancelRoomReservation function handles this by removing the room from the booked list and updating the reservation file.

**Receipt Generation**

The system generates receipts for room bookings and check-ins, storing them in a file for record-keeping.

**View Rooms**

The system provides functionality to view all reserved and checked-in rooms. This helps both customers and managers keep track of room statuses.

## projectmain.cpp

This file contains the main function, which initializes a hotel object and starts the login process.

## Algorithms:

User Authentication: Compares input username and password with stored credentials to authenticate users.

Room Booking: Checks room availability and books a room if available.

Check-in: Registers a customer as checked-in to a reserved room.

Receipt Generation: Writes reservation details to a receipt file.

File Operations: Read/write operations to manage room reservations and receipts.

### Data Structures:

Vectors: Used to store booked room numbers.

Unordered Map: Stores checked-in guest information with room numbers as keys and customer names as values.

Strings: Store hotel information, usernames, passwords, room numbers, and dates.

File Streams: Used for reading from and writing to text files for data persistence.

## Testing and Verification

1. Book room

Testing the book room choice.

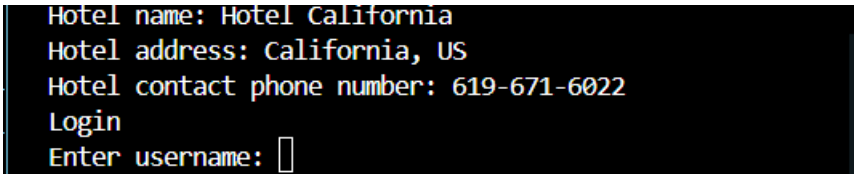
First the customer will have to login.

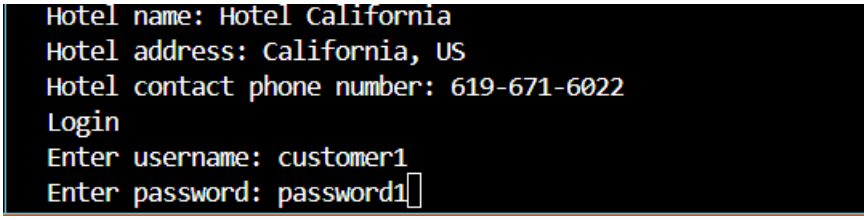
There are 3 customers: customer1 , customer2 and customer3

The password of each customer is password1 , password2 and password3 .

We will first login as customer1 .

In the user name we will type customer1 and the password will be password1 .




After that we will get the customer menu :

```
Hotel Management System Menu
1. Book a room
2. Check room availability
3. Check-in
4. Exit
Enter your choice:
```

We will choose the first choice: Book a room.

And after that we will have to choose the room type:

```
Hotel Management System Menu
1. Book a room
2. Check room availability
3. Check-in
4. Exit
Enter your choice: 1
Option 1 selected: Book a room
Please choose the room type (standard/deluxe):
```

Let choose deluxe.

```
Option 1 selected: Book a room
Please choose the room type (standard/deluxe): deluxe
Room deluxe3 booked successfully.
Hotel Management System Menu
1. Book a room
2. Check room availability
3. Check-in
4. Exit
Enter your choice:
```

After choosing the room type the message of deluxe3 booked successfully will be printed.

And the room deluxe3 will be written in the file.txt as a booked room and the customer will receive a receipt of his booking in the receipt file.

project hotel ori > file.txt

```
1  standard1
2  deluxe1
3  deluxe2
4  deluxe3
5
```

```
24  Room Reservation Receipt
25  Room Type: deluxe
26  Room Number: deluxe3
27  Total Amount: $55.00
28  ------------------------
29
```

2. Check in

Let check in the same room , room deluxe3 .

We have to chose the number 3 as our choice in the customer menu and then choose the deluxe3 as the room we want to check in at . after that the user have to choose the check in date and check out date .

```
Enter your choice: 3
Option 3 selected: Check-in
Enter room number: deluxe3
Enter check-in date (YYYY-MM-DD): 2024-06-01
Enter check-out date (YYYY-MM-DD): 2024-06-03
Customer checked in successfully.
Hotel Management System Menu
1. Book a room
2. Check room availability
3. Check-in
4. Exit
Enter your choice:
```

The message customer checked in successfully will be printed and the details of the room and the customer name and check in date and check out date will be written in the receipt file .

```
29   Room Reservation Receipt
30   Customer Name: customer1
31   Room Type: deluxe
32   Room Number: deluxe3
33   Check-In Date: 2024-06-01
34   Check-Out Date: 2024-06-03
35   Total Amount: $55.00
36   ------------------------
```

3. Cancel room reservation:

If the manager wants to cancel the room reservation, first the manager have to login to the interface as a manager and the password is admin and then manager needs to choose choice number 3 and choose the room to be cancelled , suppose it room deluxe3 after that the message “Room reservation cancelled successfully and the reserved room lists will be printed for the manager .

```
Option 3 selected: Cancel Room Reservation
Which room do you want to make available to book?
deluxe3
Room reservation canceled successfully.

Reserved rooms now:
standard1
deluxe1
deluxe2
Hotel Management Manager System Menu
1. Book a room
2. list of the reserved rooms
3. Cancel Room Reservation
4. Exit
5. View checked-in rooms list
Enter your choice:
```

## Improvements and Extensions

Areas for improvement and future extensions include:

Enhancing user interface and error messages.

Implementing advanced reservation features like room preferences and special requests.

Integrating with a database for scalable data management.

Adding support for online bookings and payment processing.

## Difficulties Encountered

Challenges encountered during the project include:

Designing an efficient reservation system with proper data handling.

Managing user authentication and access control securely.

Implementing file operations for data persistence.

## Conclusion

The Hotel Management System project provides a robust solution for managing hotel operations effectively. By addressing user requirements and incorporating feedback, the program delivers a user-friendly experience for customers and managers alike.

## References :

[1] C++ Documentation. https://en.cppreference.com/

[2] Stack Overflow. https://stackoverflow.com/

[3] GeeksforGeeks. https://www.geeksforgeeks.org/