Tomas Nader w10172066

Thales Teixeira Rodrigues w10173761

CSC102 H003

Assignment # 4 – Hotel Management System

Program name: Prog4-Group6

**Objectives/ Program description**

The code displays a C++ program to execute a hotel management system. The system's primary goals include managing rooms and bookings and providing a menu-driven interface for users to interact with the system. The design aims to group functionality into Hotel, Room, Customer, and Booking classes. Users can view available rooms, book and unbook rooms, view booked rooms, and generate a guest summary.

Overall, the provided C++ program establishes a basic hotel management system with key functionalities encapsulated within distinct classes. The program's features cater to the essential aspects of hotel management, providing users with a comprehensive tool to handle room bookings and customer information.

**Sample Input**

1. Display Available Rooms

User selects option 1 from the menu.

Expected output: List of available rooms is displayed.

1. Book a Room

User selects option 2 from the menu.

Enter customers details.

1. Display Booked Rooms

User selects option 3 from the menu.

Expected output: List of booked rooms.

1. Display Guest Summary

User selects option 4 from the menu.

Enters customers name to get summary of guest.

1. Unbook a Room

User selects option 5 from the menu.

Unbook desired room providing the number of room.

1. Exit program.
2. Invalid Choice Handling:

User enters an invalid choice at the menu.

Expected Output: Error message prompts user to enter a valid choice.

1. User attempts to book a room that is already booked.
2. User tries to unbook a room that is not in the list of booked rooms.
3. User attempts to book a room that is not in the listed rooms.

**Sample Output**

1. Display Available Rooms:

Available Rooms:

Room number: 1

Room type: Single

Number of beds per room: 2

Amenities: Wi-Fi, TV, AC

Premium Room: No

Price $50 per night.

(other available rooms)….

2. Book a Room:

Enter customer name: John Doe

Enter customer age: 67

Enter customer phone number: 1234567890

Enter customer address: 123 Main St, Hattiesburg

Enter booking date (DDMMYYYY): 15112023

Enter room number: 1

Room 1 booked by John Doe.

3. Display Booked Rooms:

Room number: 1

Room type: Single

Number of beds per room: 2

Amenities: Wi-Fi, TV, AC

Premium Room: No

Price $50.0 per night.

Booked by: John Doe

4. Display Guest Summary:

Guest Summary:

Customer Name: John Doe

Customer Age: 67

Customer Phone Number: 1234567890

Customer Address: 123 Main St, Hattiesburg

Booking Date: 11/15/2023

5. Unbook a Room:

Room 1 unbooked successfully.

6. Exit program

Output: Exiting program.

7. Invalid Choice Handling:

Output: Invalid choice. Please try again.

8. User attempts to book a room that is already booked.

Output: Error. Room 1 is already booked.

9. User tries to unbook a room that is not in the list of booked rooms.

Output: Error. Room 1 not found in bookings.

10. User attempts to book a room that is not in the listed rooms.

Output:

Enter room number: 4234

Error. Room 4234 not found or not available for booking.

**Algorithm Design**

The overall algorithm design of the program encapsulated in main is the following:

Hotel Initialization: The Hotel constructor initializes the hotel with a specified number of single, double, and suite rooms.

Booking a Room: The bookRoom function searches for the specified room and books it for the provided customer if it is available.

Unbooking a Room: The unbookRoom function searches for the specified booking and unbooks the corresponding room if found.

Displaying Available Rooms: The roomsAvailable function iterates through the list of rooms and displays their information if they are not booked.

Displaying Booked Rooms: The roomsBooked function iterates through the list of bookings and displays the information of the booked rooms along with the booking customer.

Displaying Guest Summary: The guestSummary function takes a customer as input and displays their information and booking date.

**Obstacles in design** **/ development / coding of proposal**

* Error Handling: The code lacks comprehensive error handling, especially in the bookRoom and unbookRoom functions. Adding proper exception handling would make the system more robust.
* Code Duplication: Some code snippets for user input are duplicated in the main function. Extracting common functionality into separate functions could enhance code readability.
* Limited Validation: The code assumes valid inputs in many places. Adding input validation for user inputs, such as age, phone number, and room number, would improve the robustness of the system.
* Ensuring that each class interacts seamlessly, and that data is passed correctly can be a challenge.
* Error handling in Hotel class: The bookRoom function in the Hotel class throws a runtime error if an issue occurs. Implementing more specific error handling, such as using custom exception classes or providing detailed error messages, would enhance the robustness of the system.
* Another obstacle in the process of coding the program, was making sure all the classes have consistency and that they were able to work all together to achieve the final goal, which was to provide a user-friendly interface to manage a hotel.
* Furthermore, throughout the development of the program we had to make sure the functionality of the member functions worked in main.

**Further Enhancements**

User Interface Improvement: Enhance the user interface with clearer prompts and messages to guide users through the interaction.

Persistence: Implement a mechanism to persist data, such as room and booking information, between program executions.

Extended Functionality: Add features like searching for available rooms based on criteria, modifying bookings, and handling different room rates based on seasons.

Logging: Introduce a logging mechanism to record significant events or errors during program execution.

Graphical User Interface (GUI): Consider developing a graphical user interface for a more user-friendly experience.

Concurrency: Implement concurrent access control mechanisms if the system is to be used in a multi-user environment.

Database Integration: Integrate a database to store and retrieve hotel-related data for scalability and ease of management.

File reading: Read customer information from a file. Data entry can be simplified by allowing users to prepare customer information in a structured file format and then import it into the system. This reduces manual data entry errors and saves time.

Data Encryption: If dealing with sensitive customer information, consider implementing data encryption techniques to protect customer privacy.

Charges: Provide a mechanism for customers to make payments and to display room charges so the customer is able to purchase things from the room.

**Compilation screenshots**

1)

A screen shot of a computer

Description automatically generated

2)

A black screen with a black background

Description automatically generated

3)

A black screen with a black background

Description automatically generated

4)

A black background with white text

Description automatically generated

5)

A black screen with a black background

Description automatically generated

6)

A black screen with white text

Description automatically generated

7)

A black screen with a black border

Description automatically generated

8)

A black screen with a black background

Description automatically generated

9)

A screen shot of a computer

Description automatically generated

10)

A screenshot of a computer

Description automatically generated

**Conclusion**

With a well-organized C++ program, the hotel management system program offers a complete solution to effectively manage rooms, reservations and customer information. The four main classes that make up the system are Hotel, Room, Customer and Booking. Each class is intended to have a certain set of features. Through a menu-based interface, users can view which rooms are available, book and cancel them, and create guest reports.