C++ Report

Hotel Booking System

Group No. : 7
Cohort : Sam Altman

Presented By: Pranav, Mahek, Pratik, Dhruv

Introduction to the Case Study

This case study focuses on the development of a Hotel Booking System using C++.
Managing room bookings manually in hotels can be time-consuming and prone to human error. The system simplifies the process by automating tasks such as room booking, checkout, and viewing currently occupied rooms.

The project is a console-based application designed for clarity, correctness, and ease of use. It demonstrates key concepts of Object-Oriented Programming (OOP) in C++, including classes, arrays, encapsulation, and functions.

Case Background (Abstract)

In many hotels, booking and checkout processes are handled manually using registers or spreadsheets. This method is inefficient and may lead to issues like double bookings, incorrect billing, or missing room availability data.

The objective of this project is to create a Hotel Booking System in C++ that automates the hotel's booking and checkout process.
It provides an interactive interface to manage room availability, calculate bills with taxes, and track occupied rooms efficiently.

Case Study Design

The system is designed using a menu-driven program structure that provides the following functionalities:

- Book Room
- Guest Checkout
- Show Occupied Rooms
- Exit

The program uses an object-oriented approach where all operations are encapsulated within a single class named **`Hotel`**.

Arrays are used to store room types, prices, and room availability details. The design ensures that the booking and checkout processes update availability data dynamically.

Methods & Algorithms / Technology Applied

The following methods and technologies are applied:

- **C++ Programming Language** for implementation
- **Class and Object Concepts** to encapsulate hotel data and operations
- **Array Data Structure** to store information about room types, prices, and availability
- **Conditional Statements** for input validation and logic flow
- **Loops** to handle repeated user interactions and display data
- **Menu-Driven Interface** using a `do-while` loop and `switch` statement
- **Basic Arithmetic Operations** for billing and tax calculations

These methods ensure smooth execution, accuracy in data management, and user-friendly interaction.

Case Study Implementation Details and Snapshots

The implementation of the Hotel Booking System consists of multiple functions within the class `Hotel`.
Each function performs a specific task related to booking management.

1. Class Declaration

The `Hotel` class contains arrays for room types, prices, availability, and occupancy. It also includes a tax rate for billing calculations.

```
class Hotel {
private:
    string type[3] = {"Single","Double","Deluxe"};
    int price[3] = {2000,4000,7000};
    int available[3] = {5,8,3};
    int occupied[3] = {0,0,0};
    int nights[3] = {0,0,0};
    double taxRate = 0.10;
public:
    void bookRoom();
    void guestCheckout();
    void showOccupied();
};
```

2. Book Room Function

This function allows users to select room type, number of rooms, and nights for their stay. It calculates the total bill with tax and updates room availability.

```
void bookRoom() {
    cout << "\n--- Available Rooms ---\n";
    for(int i=0;i<3;i++){
        cout << i+1 << ". " << type[i]
             << " | Rs." << price[i]
             << " | Available: " << available[i] << endl;
    }
    // Further logic for booking and bill calculation...
}
```

3. Guest Checkout Function

This function allows guests to check out of occupied rooms, freeing them up for future bookings.

```
void guestCheckout() {
    cout << "\n--- Guest Checkout ---\n";
    // Displays occupied rooms and updates availability
}
```

4. Show Occupied Rooms Function

Displays all currently occupied rooms with room type and number of nights booked.

```
void showOccupied() {
    cout << "\n--- Occupied Rooms ---\n";
    // Shows occupied room details
}
```

5. Main Function (Menu Driven Program)

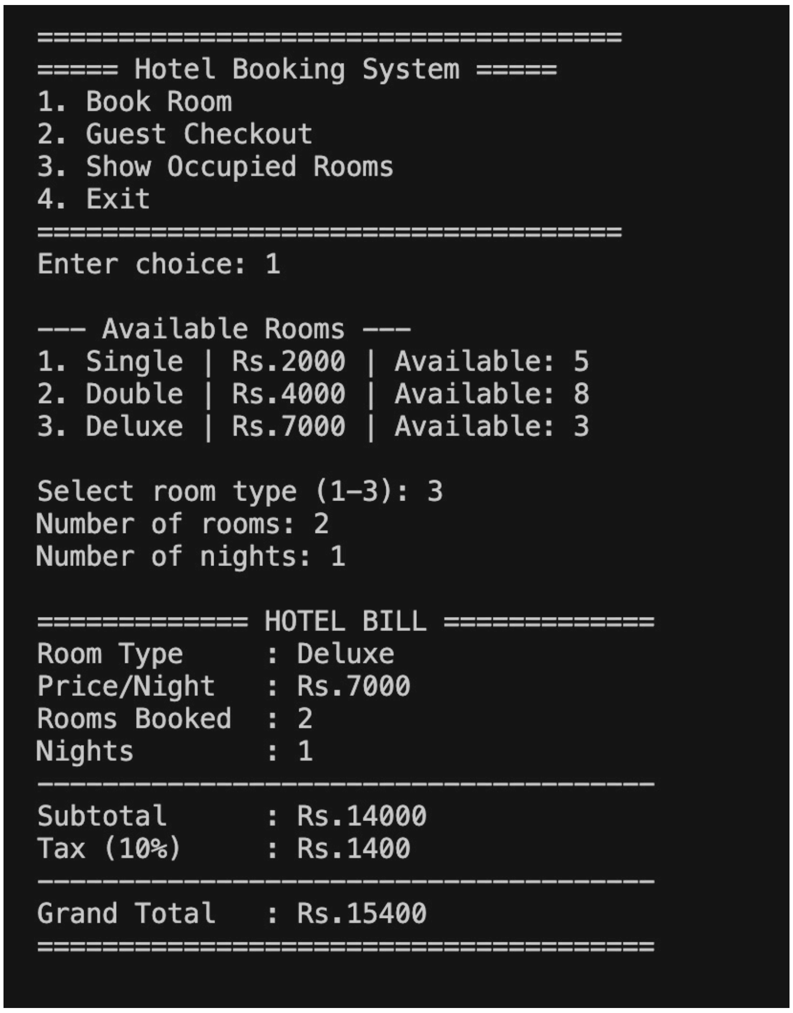
The main function repeatedly displays the menu until the user exits.

```
int main() {
    Hotel h;
    int choice;
    do {
        cout<<"\n====================================\n";
        cout << "===== Hotel Booking System =====\n";
        cout << "1. Book Room\n";
        cout << "2. Guest Checkout\n";
        cout << "3. Show Occupied Rooms\n";
        cout << "4. Exit\n";
        cout << "====================================\n";
        cout << "Enter choice: ";
        cin >> choice;

        switch(choice){
            case 1: h.bookRoom(); break;
            case 2: h.guestCheckout(); break;
            case 3: h.showOccupied(); break;
            case 4: cout << "Thank you!\n"; break;
            default: cout << "Invalid choice\n";
        }
    }while(choice != 4);
    return 0;
}
```

Snapshots of Output

1. **Booking Room:**



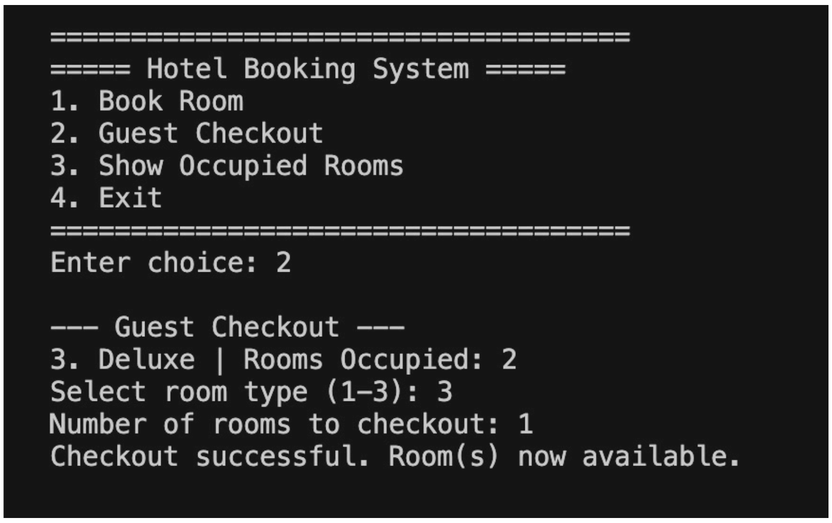
```
====================================
===== Hotel Booking System =====
1. Book Room
2. Guest Checkout
3. Show Occupied Rooms
4. Exit
====================================
Enter choice: 1

--- Available Rooms ---
1. Single | Rs.2000 | Available: 5
2. Double | Rs.4000 | Available: 8
3. Deluxe | Rs.7000 | Available: 3

Select room type (1-3): 3
Number of rooms: 2
Number of nights: 1

============= HOTEL BILL =============
Room Type     : Deluxe
Price/Night   : Rs.7000
Rooms Booked  : 2
Nights        : 1
--------------------------------------
Subtotal      : Rs.14000
Tax (10%)     : Rs.1400
--------------------------------------
Grand Total   : Rs.15400
======================================
```

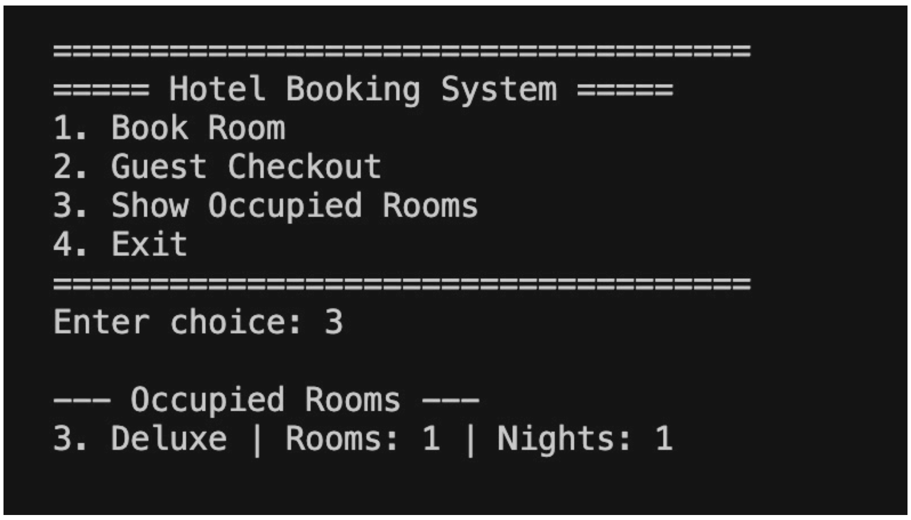
2. Guest Checkout:



```
====================================
===== Hotel Booking System =====
1. Book Room
2. Guest Checkout
3. Show Occupied Rooms
4. Exit
====================================
Enter choice: 2

--- Guest Checkout ---
3. Deluxe | Rooms Occupied: 2
Select room type (1-3): 3
Number of rooms to checkout: 1
Checkout successful. Room(s) now available.
```

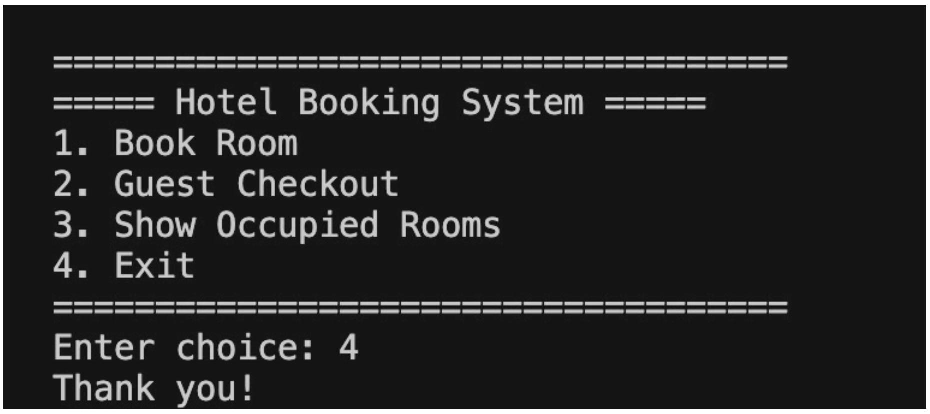
3. Show Occupied rooms



```
====================================
===== Hotel Booking System =====
1. Book Room
2. Guest Checkout
3. Show Occupied Rooms
4. Exit
====================================
Enter choice: 3

--- Occupied Rooms ---
3. Deluxe | Rooms: 1 | Nights: 1
```

4. Exit



```
====================================
===== Hotel Booking System =====
1. Book Room
2. Guest Checkout
3. Show Occupied Rooms
4. Exit
====================================
Enter choice: 4
Thank you!
```

Limitations & Future Improvements

The current system:

- Does not use file handling, so data is lost when the program ends.
- Handles only three room types with predefined quantities.

Future improvements may include:

- File handling for permanent record storage.
- Customer details (name, contact, address).
- Online booking functionality.
- Graphical interface using GUI libraries or frameworks.

Case Study Results and Conclusion

The **Hotel Booking System** successfully automates the process of room booking, checkout, and room status tracking.
It demonstrates the effective use of **C++ OOP principles** such as encapsulation and modular design.

The system is efficient, easy to use, and suitable for small hotels or academic learning purposes. This project showcases how programming concepts can be applied to solve real-world management problems.

References

1. Bjarne Stroustrup – *The C++ Programming Language*
2. E. Balagurusamy – *Object-Oriented Programming with C++*
3. cppreference.com – *C++ Standard Library Documentation*
4. W3Schools – *C++ Programming Guide*