Hotel Management System

Submitted by:

- Muhammad Usman
- Abdul Samad
- Momna Nawaz

Instructor:

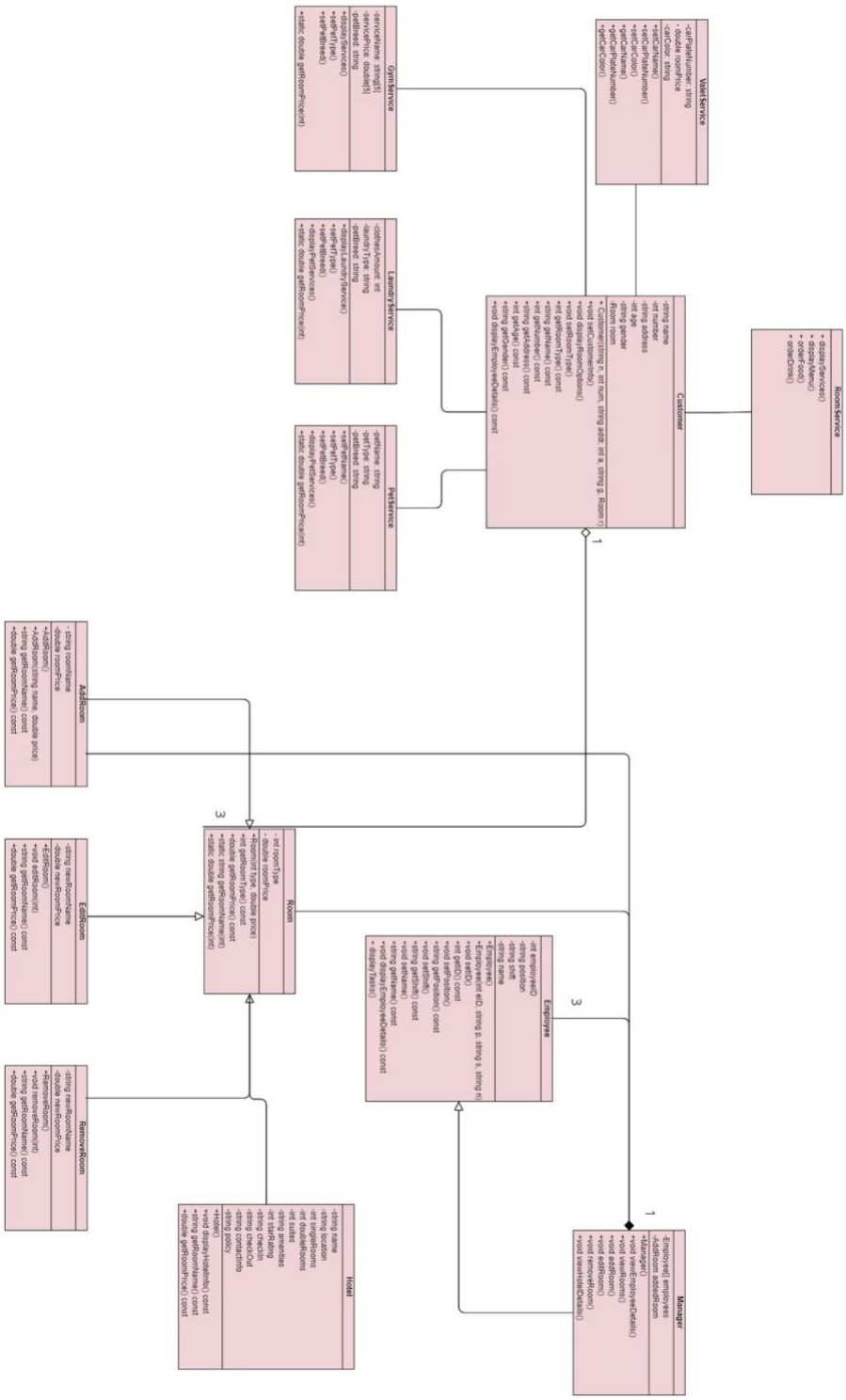
Sir Sagheer Ahmad

Subject:

Object Oriented Programming

CONTENTS

UML Diagram


ValetService
-carPlateNumber: string
- double roomPrice
-carColor: string
+setCarName()
+setCarPlateNumber()
+setCarColor()
+getCarName()
+getCarPlateNumber()
+getCarColor()
GymService
-serviceName: string[5]
-servicePrice: double[5]
-petBreed: string
+displayGymServices()
+setPetType()
+setPetBreed()
+static double getRoomPrice(int)
LaundryService
-clothesAmount: int
-laundryType: string
-petBreed: string
+displayLaundryServices()
+setPetType()
+setPetBreed()
+displayPetServices()
+static double getRoomPrice(int)
PetService
-petName: string
-petType: string
-petBreed: string
+displayPetServices()
+setPetType()
+setPetBreed()
+displayPetServices()
+static double getRoomPrice(int)
Customer
-string name
-int number
-string address
-string age
-string gender
-Room room
+ Customer(string n, int num, string addr, int a, string g, Room r)
+void setCustomerInfo()
+void displayCustomerDetails()
+int getRoomType() const
+string getName() const
+int getNumber() const
+string getAddress() const
+string getAge() const
+string getGender() const
+void displayEmployeeDetails() const
RoomService
+ displayServices()
+ display(Menu)
+ orderFood()
+ orderDrink()
AddRoom
-string roomName
-double roomPrice
+AddRoom()
+AddRoom(string name, double price)
+string getRoomName() const
+double getRoomPrice() const
Room
- int roomType
-Room(int type, double price)
+Room()
+int getRoomType() const
+double getRoomPrice() const
+static string getRoomName(int)
+static double getRoomPrice(int)
EditRoom
-string newRoomName
-double newRoomPrice
+EditRoom()
+void editRoom(int)
+string getRoomName() const
+double getRoomPrice() const
RemoveRoom
-string nameRoomName
-double newRoomPrice
+RemoveRoom()
+void removeRoom(int)
+string getRoomName() const
+double getRoomPrice() const
Employee
-int employeeID
-string position
-string shift
-string name
+Employee(int eID, string p, string s, string n)
+void setID()
+int getID() const
+void setPosition()
+string getPosition() const
+void setShift()
+string getShift() const
+void setName()
+string getName() const
+void displayEmployeeDetails() const
+double calculateTotalSalary()
Manager
-Employee[] employees
-AddRoom addedRoom
+Manager()
+void viewEmployee()
+void addRoom()
+void removeRoom()
+void viewHotelDetails()
Hotel
-string name
-string location
-int singleRooms
-int doubleRooms
-int suites
-string amenities
-int starRating
-string checkIn
-string checkOut
-string cancellationPolicy
-string policy
+Hotel()
+void displayHotelInfo() const
+string getRoomName() const
+double getRoomPrice() const
1
3
3
1

Abstract:

This report provides a detailed overview of a Hotel Management System program written in C++. The program facilitates the management of hotel operations by supporting functionalities for managers, employees, and customers. The primary focus is on room management, employee clock-in processes, customer reservations, and additional hotel services. The program utilizes object-oriented programming principles such as inheritance, composition, polymorphism, and aggregation to achieve a modular and flexible design.

Introduction:

The Hotel Management System is designed to streamline hotel operations, making it easier for managers to handle room details, employees to manage their shifts, and customers to make reservations and provide feedback. The program employs a menu-driven interface, allowing users to navigate through various options based on their roles within the hotel system.

Idea:

The core idea behind the Hotel Management System is to create a comprehensive software solution that addresses the various needs of hotel management. By integrating functionalities for different user roles, managers, employees, and customers, the system ensures that all aspects of hotel operations are covered.

Stake Holders:

- Hotel owners and management who will benefit from the streamlined operations and improved efficiency.
- Hotel staff who will find it easier to manage their tasks and shifts.
- Hotel customers who will experience a more convenient and responsive service.

Classes:

1. **Manager:** Handles room management, employee details, and hotel details.
   - viewEmployeeDetails(): Displays details of all employees.
   - viewRooms(): Lists all rooms and their statuses.
   - addRoom(): Adds a new room to the hotel.
   - editRoom(): Edits the details of an existing room.
   - removeRoom(): Removes a room from the hotel.
   - viewHotelDetails(): Displays overall hotel details.

2. **Employee:** Manages employee information and tasks.
   - setName(): Sets the employee's name.
   - setID(): Sets the employee's ID.
   - setPosition(): Sets the employee's position.
   - setShift(): Sets the employee's shift.
   - getName(): Returns the employee's name.
   - getID(): Returns the employee's ID.

- getPosition(): Returns the employee's position.
- getShift(): Returns the employee's shift.
- displayEmployeeDetails(): Displays the employee's details.
- displayTasks(): Displays tasks assigned to the employee.

3. **Customer:** Manages customer reservations and feedback.
   - setCustomerInfo(): Sets the customer's information.
   - displayRoomOptions(): Displays available room options.
   - setRoomType(): Sets the type of room for the customer.
   - getName(): Returns the customer's name.
   - getNumber(): Returns the customer's contact number.
   - getAddress(): Returns the customer's address.
   - getAge(): Returns the customer's age.
   - getGender(): Returns the customer's gender.

4. **Room:** Represents hotel rooms and manages room details.

   - Static functions: getRoomPrice(): Returns the price of a room type.
   - getRoomName(): Returns the name of a room type.

5. **Service:** Provides a base for various hotel services.

- virtual void displayServices() = 0: Pure virtual function to display available services.
- RoomService, ValetService, PetService, LaundryService, GymService (Derived Classes): Represent specific hotel services.

- Override displayServices(): Provides specific implementations for displaying available services.

Functions:

1. Main Menu Functions:
Display options for manager, employee, customer, and exit. Manage input and call appropriate functions based on user choice.

2. Manager Functions:

- viewEmployeeDetails(): Displays details of all employees.
- viewRooms(): Lists all rooms and their statuses.
- addRoom(): Adds a new room to the hotel.
- editRoom(): Edits the details of an existing room.
- removeRoom(): Removes a room from the hotel.
- viewHotelDetails(): Displays overall hotel details.

3. Employee Functions:

- setName(), setID(), setPosition(), setShift(): Set various employee details.
- getName(), getID(), getPosition(), getShift(): Get various employee details.
- displayEmployeeDetails(): Displays the employee's details.
- displayTasks(): Displays tasks assigned to the employee.

4. Customer Functions:

- setCustomerInfo(): Sets the customer's information.
- displayRoomOptions(): Displays available room options.
- setRoomType(): Sets the type of room for the customer.
- getName(), getNumber(), getAddress(), getAge(), getGender(): Get various customer details.
- Handle additional service requests and payments.

Key Object Oriented Programming Concepts:

1. Inheritance:
Classes such as RoomService, ValetService, PetService, LaundryService, and GymService inherit from the base class Service. This allows them to override the displayServices() function and provide specialized implementations.

2. Composition:
The Manager class uses composition to include three Employee objects, directly embedding them within its structure.

3. Aggregation:
The Customer class aggregates Room objects, maintaining a reference to one or more rooms that the customer interacts with.

4. **Polymorphism:**
   The program employs polymorphism through the use of the virtual function displayServices() in the Service class. This allows derived classes to provide their own implementations, enabling dynamic behavior based on the actual object type at runtime.

Conclusion:

The Hotel Management System program effectively demonstrates the use of object-oriented programming principles to create a flexible and maintainable software solution for hotel management. By leveraging inheritance, composition, aggregation, and polymorphism, the program provides a comprehensive tool for managing hotel operations, catering to the needs of managers, employees, and customers alike. The modular design ensures that the system can be easily extended and maintained, making it a robust solution for the domain of hotel management.

CODE:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

class Employee
{
private:
    int employeeID;
    string position;
    string shift;
    string name;

public:
    Employee() : employeeID(0), position(""), shift(""), name("") {}
    Employee(int eID, string p, string s, string n)
        : employeeID(eID), position(p), shift(s), name(n) {}

    void setID()
    {
        cout << "Enter your ID: ";
        cin >> employeeID;
    }

    int getID() const
    {
        return employeeID;
    }

    void setPosition()
    {
        cout << "Enter your position: ";
        cin >> position;
    }

    string getPosition() const
    {
        return position;
    }

    void setShift()
    {
        cin.ignore();
        cout << "Enter your shift (Day or Night): ";
        getline(cin, shift);
    }

    string getShift() const
    {
        return shift;
```

```
}

void setName()
{
    cout << "Enter your name: ";
    cin.ignore();
    getline(cin, name);
}

string getName() const
{
    return name;
}

void displayEmployeeDetails() const
{
    cout << "Name: " << name << ", ID: " << employeeID << endl;
}

void displayTasks() const
{
    int employeePosition;
    cout << "Select your position: " << endl;
    cout << "1. Receptionist " << endl;
    cout << "2. Security " << endl;
    cout << "3. Janitor " << endl;
    cin >> employeePosition;

    if (employeePosition == 1)
    {
        cout << "List of tasks to complete: " << endl;
        cout << "1. Guest Check-In/Check-Out " << endl;
        cout << "2. Reservation Management " << endl;
        cout << "3. Billing and Payments " << endl;
        cout << "4. Information Provision " << endl;
        cout << "5. Customer Service " << endl;
        cout << "6. Problem Solving " << endl;
    }

    else if (employeePosition == 2)
    {
        cout << "List of tasks to complete: " << endl;
        cout << "1. Patrolling and Surveillance" << endl;
        cout << "2. Access Control" << endl;
        cout << "3. Incident Response" << endl;
        cout << "4. Guest Safety" << endl;
        cout << "5. Report Writing" << endl;
        cout << "6. Emergency Coordination" << endl;
    }

    else if (employeePosition == 3)

    {
```

```
            cout << "List of tasks to complete: " << endl;
            cout << "1. Cleaning and Maintenance" << endl;
            cout << "2. Trash Removal " << endl;
            cout << "3. Monitoring Supplies " << endl;
            cout << "4. Reporting Maintenance Issues " << endl;
            cout << "5. Assisting Housekeeping " << endl;
            cout << "6. Compliance with Safety Standards " << endl;
        }

        else
        {
            cout << "Please select the correct option" << endl;
        }
    }
};


class Room
{
private:

    int roomType;
    double roomPrice;

public:
    Room(int type, double price) : roomType(type), roomPrice(price) {}

    int getRoomType() const
    {
        return roomType;
    }

    double getRoomPrice() const
    {
        return roomPrice;
    }

    static string getRoomName(int roomType)
    {
        switch (roomType)
        {
        case 1: return "Standard Single Room";
        case 2: return "Economy Single Room";
        case 3: return "Cozy Retreat";
        case 4: return "Standard Double Room";
        case 5: return "Deluxe Double Room";
        case 6: return "Spacious Duo";
        case 7: return "Executive Suite";
        case 8: return "Presidential Suite";
        case 9: return "Luxury Suite";
        default: return "Unknown Room Type";
        }
    }
```

```
    static double getRoomPrice(int roomType)
    {
        switch (roomType)
        {
        case 1: return 600;
        case 2: return 700;
        case 3: return 800;
        case 4: return 990;
        case 5: return 1100;
        case 6: return 1200;
        case 7: return 1700;
        case 8: return 2000;
        case 9: return 3000;
        default: return 0;
        }
    }
};


class AddRoom : public Room
{
private:
    string roomName;
    double roomPrice;

public:
    AddRoom() : Room(0, 0), roomName("Default Room"), roomPrice(0) {}
    AddRoom(string name, double price)
        : Room(0, 0), roomName(name), roomPrice(price) {}

    string getRoomName() const
    {
        return roomName;
    }

    double getRoomPrice() const
    {
        return roomPrice;
    }
};


class EditRoom : public Room
{
private:
    string newRoomName;
    double newRoomPrice;

public:
    EditRoom() : Room(0, 0), newRoomName(""), newRoomPrice(0) {}

    void editRoom(int roomType)
```

```
    {
        cout << "Enter new room name: ";
        cin.ignore();
        getline(cin, newRoomName);

        cout << "Enter new room price: ";
        cin >> newRoomPrice;

        cout << "Room edited successfully. New details:" << endl;
        cout << "Room Name: " << newRoomName << endl;
        cout << "Room Price: " << newRoomPrice << endl;
    }
};


class RemoveRoom : public Room
{
public:
    RemoveRoom() : Room(0, 0) {}

    void removeRoom(int roomType)
    {
        cout << "Room removed: " << Room::getRoomName(roomType) << endl;
    }
};


class Hotel
{
private:
    string name;
    string location;
    int singleRooms;
    int doubleRooms;
    int suites;
    string amenities;
    int starRating;
    string checkIn;
    string checkOut;
    string contactInfo;
    string policy;

public:
    Hotel()
    {
        name = "Grand Vista Hotel";
        location = "F-6 Markaz, Islamabad";
        singleRooms = 40;
        doubleRooms = 50;
        suites = 60;
        amenities = "Outdoor Pool, Fitness Center, On-site Restaurant, Free
Wi-Fi, Business Center";
        starRating = 4;
```

```
        checkIn = "3:00 PM";
        checkOut = "11:00 AM";
        contactInfo = "Phone: +1 555-123-4567, Email: contact@domain.com";
        policy = "Pet Policy: Pets are not allowed. Cancellation Policy: Free
cancellation up to 48 hours before check-in.";
    }

    void displayHotelInfo() const
    {
        cout << "Hotel Information:" << endl;
        cout << "Name: " << name << endl;
        cout << "Location: " << location << endl;
        cout << "Single Rooms: " << singleRooms << endl;
        cout << "Double Rooms: " << doubleRooms << endl;
        cout << "Suites: " << suites << endl;
        cout << "Amenities: " << amenities << endl;
        cout << "Star Rating: " << starRating << endl;
        cout << "Check-in Time: " << checkIn << endl;
        cout << "Check-out Time: " << checkOut << endl;
        cout << "Contact Information: " << contactInfo << endl;
        cout << "Policies: " << policy << endl;
    }
};

class Manager : public Employee
{
private:
    Employee employees[3];
    AddRoom addedRoom;

public:
    Manager()
    {
        employees[0] = Employee(1, "Manager", "Day", "Hammad Ashraf");
        employees[1] = Employee(2, "Receptionist", "Night", "Ahmer Ali");
        employees[2] = Employee(3, "Security", "Night", "Abubakar Abbasi");
        addedRoom = AddRoom("Default Room", 0);
    }

    void viewEmployeeDetails()
    {
        for (int i = 0; i < 3; i++)
        {
            employees[i].displayEmployeeDetails();
        }
    }

    void viewRooms()
    {
        cout << "Room Options:" << endl;
        for (int i = 1; i <= 9; i++)
        {
```

```
            cout << i << ". " << Room::getRoomName(i) << " - Price: " <<
Room::getRoomPrice(i) << endl;
        }
    }

    void addRoom()
    {
        string roomName;
        double roomPrice;

        cout << "Enter room name: ";
        cin.ignore();
        getline(cin, roomName);

        cout << "Enter room price: ";
        cin >> roomPrice;

        addedRoom = AddRoom(roomName, roomPrice);

        cout << "Room '" << roomName << "' added successfully." << endl;

        cout << "Added Room Name: " << addedRoom.getRoomName() << endl;
        cout << "Added Room Price: " << addedRoom.getRoomPrice() << endl;
    }

    void editRoom()
    {
        int roomType;
        cout << "Enter room type (1-9): ";
        cin >> roomType;
        if (roomType >= 1 && roomType <= 9)
        {
            EditRoom editRoom;
            editRoom.editRoom(roomType);
        }
        else
        {
            cout << "Invalid room type. Please try again." << endl;
        }
    }

    void removeRoom()
    {
        int roomType;
        cout << "Enter room type (1-9): ";
        cin >> roomType;
        if (roomType >= 1 && roomType <= 9)
        {
            RemoveRoom removeRoom;
            removeRoom.removeRoom(roomType);
        }
        else
        {
```

```
            cout << "Invalid room type. Please try again." << endl;
        }
    }

    void viewHotelDetails()
    {
        Hotel hotel;
        hotel.displayHotelInfo();
    }
};


class Customer
{
private:
    string name;
    int number;
    string address;
    int age;
    string gender;
    Room room;

public:
    Customer(string n, int num, string addr, int a, string g, Room r)
        : name(n), number(num), address(addr), age(a), gender(g), room(r) {}

    void setCustomerInfo()
    {
        cout << "Enter your name: ";
        cin.ignore();
        getline(cin, name);
        cout << "Enter your number: ";
        cin >> number;
        cout << "Enter your address: ";
        cin.ignore();
        getline(cin, address);
        cout << "Enter your age: ";
        cin >> age;
        cin.ignore();
        cout << "Enter your gender (Male/Female): ";
        getline(cin, gender);
    }

    void displayRoomOptions()
    {
        cout << "Room Options:" << endl;
        for (int i = 1; i <= 9; i++)
        {
            cout << i << ". " << Room::getRoomName(i) << " - Price: " <<
Room::getRoomPrice(i) << endl;
        }
    }
```

```
    void setRoomType()
    {
        cout << "Enter room type (1-9): ";
        int roomType;
        cin >> roomType;
        if (roomType >= 1 && roomType <= 9)
        {
            room = Room(roomType, Room::getRoomPrice(roomType));
        }
        else
        {
            cout << "Invalid room type. Please try again." << endl;
        }
    }

    int getRoomType() const
    {
        return room.getRoomType();
    }

    string getName() const
    {
        return name;
    }

    int getNumber() const
    {
        return number;
    }

    string getAddress() const
    {
        return address;
    }

    int getAge() const
    {
        return age;
    }

    string getGender() const
    {
        return gender;
    }
};

class RoomService
{
public:

    virtual void displayServices()
    {
        int choice;
```

```
        cout << "What would you like to do?" << endl;
        cout << "1. Order From Available Menu" << endl;
        cout << "2. Get Housekeeping" << endl;
        cout << "3. Room Amenities" << endl;
        cin >> choice;

        switch (choice)
        {
        case 1:
            displayMenu();
            break;
        case 2:
            cout << "Housekeeping service has been requested." << endl;
            break;
        case 3:
            cout << "Here are the room amenities: TV, Wi-Fi, Mini-bar, Safety
Locker, Hairdryer." << endl;
            break;
        default:
            cout << "Invalid choice. Please try again." << endl;
            break;
        }
    }

    void displayMenu()
    {
        int menuChoice;
        cout << "What would you like to order?" << endl;
        cout << "1. Food" << endl;
        cout << "2. Drink" << endl;
        cin >> menuChoice;

        switch (menuChoice)
        {
        case 1:
            orderFood();
            break;

        case 2:
            orderDrink();
            break;

        default:
            cout << "Invalid choice. Please try again." << endl;
            break;
        }
    }

    void orderFood()
    {
        int foodChoice;
        cout << "Select food item:" << endl;
        cout << "1. Chicken Steak - 1100" << endl;
```

```
        cout << "2. Zinger Burger - 750" << endl;
        cout << "3. Crown Crust Pizza - 1000" << endl;
        cin >> foodChoice;

        switch (foodChoice)
        {
        case 1:
            cout << "You've ordered Chicken Steak." << endl;
            break;
        case 2:
            cout << "You've ordered Zinger Burger." << endl;
            break;
        case 3:
            cout << "You've ordered Crown Crust Pizza." << endl;
            break;
        default:
            cout << "Invalid choice. Please try again." << endl;
            return;
        }

        cout << "Your order has been received. Thank you!" << endl;
    }

    void orderDrink()
    {
        int drinkChoice;
        cout << "Select drink item:" << endl;
        cout << "1. Strawberry Vanilla Shake - 400" << endl;
        cout << "2. Oreo Milkshake - 500" << endl;
        cout << "3. Mint Margarita - 450" << endl;
        cin >> drinkChoice;

        switch (drinkChoice)
        {
        case 1:
            cout << "You've ordered Strawberry Vanilla Shake." << endl;
            break;
        case 2:
            cout << "You've ordered Oreo Milkshake." << endl;
            break;
        case 3:
            cout << "You've ordered Mint Margarita." << endl;
            break;
        default:
            cout << "Invalid choice. Please try again." << endl;
            return;
        }

        cout << "Your order has been received. Thank you!" << endl;
    }
};
```

```
class ValetService
{
private:

    string carName;
    string carPlateNumber;
    string carColor;

public:

    ValetService()
        : carName(""), carPlateNumber(""), carColor("") {}


    ValetService(const string& carN, const string& carP, const string&
carCol)
        : carName(carN), carPlateNumber(carP), carColor(carCol) {}

    void setCarName()
    {
        cout << "Enter your car's name: ";
        cin.ignore();
        getline(cin, carName);
    }

    void setCarPlateNumber()
    {
        cout << "Enter your car's plate number: ";
        getline(cin, carPlateNumber);
    }

    void setCarColor()
    {
        cout << "Enter your car's color: ";
        getline(cin, carColor);
    }

    string getCarName() const
    {
        return carName;
    }

    string getCarPlateNumber() const
    {
        return carPlateNumber;
    }

    string getCarColor() const
    {
        return carColor;
    }
};
```

```
class PetService
{
private:

    string petName;
    string petType;
    string petBreed;

public:
    PetService()
        : petName(""), petType(""), petBreed("") {}
    PetService(const string& petN, const string& petT, const string& petB)
        : petName(petN), petType(petT), petBreed(petB) {}

    void setPetName()
    {
        cout << "Enter the name of your pet: " << endl;
        cin.ignore();
        getline(cin, petName);
    }

    void setPetType()
    {
        cout << "Enter the type of your pet (Dog/Cat): " << endl;
        cin.ignore();
        getline(cin, petType);
    }

    void setPetBreed()
    {
        cout << "Enter the breed of your pet: " << endl;
        cin.ignore();
        getline(cin, petBreed);
    }

    virtual void displayServices()
    {
        int pChoice;
        cout << "The following services are available for pets: " << endl;
        cout << "1. Pet Sitting " << endl;
        cout << "2. Pet Grooming " << endl;
        cout << "3. Pet Spa Services" << endl;

        cout << "Please select your option: " << endl;
        cin >> pChoice;
        if (pChoice == 1)
        {
            int sittingTime;
            cout << "How many hours of pet sitting would you like to request?
" << endl;
            cin >> sittingTime;
            cout << "Thank you for your input. Please deliver your pet to the
receptionist when you leave your room. " << endl;
```

```
        }

        else if (pChoice == 2)
        {
            cout << "Please deliver your pet to the receptionist for Grooming
Service " << endl;
        }

        else if (pChoice == 3)
        {
            cout << "Please deliver your pet to the receptionist for Spa
Service" << endl;
        }
    }
};

class LaundryService
{
private:

    int clothesAmount;
    string laundryType;

public:
    LaundryService() : clothesAmount(0), laundryType("") {}

    LaundryService(const int& cAmount, const string& lType) :
clothesAmount(cAmount), laundryType(lType) {}

    virtual void displayServices()
    {
        cout << "Select your type of service:  " << endl;
        cout << "1. Dry Cleaning " << endl;
        cout << "2. Pressing " << endl;
        cout << "3. Washing " << endl;
        int choice;
        cin >> choice;

        if (choice == 1)
        {
            laundryType = "Dry Cleaning";
        }
        else if (choice == 2)
        {
            laundryType = "Pressing";
        }
        else if (choice == 3)
        {
            laundryType = "Washing";
        }

        cout << "Enter the amount of clothes you want to provide: " << endl;
        cin >> clothesAmount;
```

```
        cout << "Your clothes will be picked up shortly for " << laundryType
<< endl;
    }
};

class GymService
{
private:
    string serviceName[5];
    double servicePrice[5];

public:
    GymService()
    {

        serviceName[0] = "Personal Training";
        serviceName[1] = "Group Fitness";
        serviceName[2] = "Yoga";
        serviceName[3] = "Massage Therapy";
        serviceName[4] = "Nutrition Counseling";

        servicePrice[0] = 50.0;
        servicePrice[1] = 30.0;
        servicePrice[2] = 40.0;
        servicePrice[3] = 60.0;
        servicePrice[4] = 70.0;
    }

    virtual void displayServices()
    {
        cout << "Gym Services:" << endl;
        for (int i = 0; i < 5; i++)
        {
            cout << "Service Name: " << serviceName[i] << endl;
            cout << "Service Price: " << servicePrice[i] << endl;
            cout << endl;
        }
        cout << "Enter the name of the service that you would like " << endl;
        string gymServiceChoice;
        cin.ignore();
        getline(cin, gymServiceChoice);

        cout << "You have successfully subscribed. Thank You!" << endl;


    }
};

void displayHotelSign()
{
    cout << "\033[1;35m";
```

```
    cout << "      ___           ___           ___           ___
___       " << endl;
    cout << "     /\\  \\         /\\  \\         /\\  \\         /\\  \\
/\\__\\     " << endl;
    cout << "    /::\\  \\       /::\\  \\       /::\\  \\       /::\\  \\
/:/  /     " << endl;
    cout << "   /:/\\:\\
\  \\     /:/\\:\\  \\     /:/\\:\\  \\     /:/\\:\\  \\     /:/  /      " <<
endl;
    cout << "  /:/  \\:\\  \\   /:/  \\:\\  \\   /:/  \\:\\  \\   /::\\~\\:\\
\\   /:/  /  ___  " << endl;
    cout << " /:/__/ \\:\\__\\ /:/__/ \\:\\__\\ /:/__/_\\:\\__\\ /:/\\:\\
\\:\\__\\ /:/__/  /\\__\\" << endl;
    cout << " \\:\\  \\  \\/__/ \\:\\  \\  \\/__/ \\:\\  / \\:\\  /\\ \\/__/
\\/__\\:\\/:/  / \\:\\  \\  \\/__/" << endl;
    cout << "  \\:\\  \\        \\:\\  /:/  /   \\:\\/ \\:\\__\\        \\::/
/   \\:\\  /:/  / " << endl;
    cout << "   \\:\\  \\        \\:\\/:/  /     \\:\\ \\/__/        /:/  /
\\:\\/:/  /  " << endl;
    cout << "    \\:\\__\\        \\::/  /       \\:\\__\\         /:/  /
\\::/  /   " << endl;
    cout << "     \\/__/         \\/__/         \\/__/         \\/__/
\\/__/    " << endl;
    cout << "\033[0m";
    cout << "\033[1;33m";
    cout << "\n----------------------------------------------------------------
------------------\n";
    cout << "  Welcome to Royal Paradise Hotel
" << endl;
    cout << "----------------------------------------------------------------
----------------\n";
    cout << "\033[0m";
}


void displayMainMenu() {
    cout << "\033[1;32m";
    cout << "======================" << endl;
    cout << "||       Main Menu   ||" << endl;
    cout << "======================" << endl;
    cout << "Are you:" << endl;
    cout << "1. \033[1;36mManager\033[1;32m" << endl;
    cout << "2. \033[1;36mEmployee\033[1;32m" << endl;
    cout << "3. \033[1;36mCustomer\033[1;32m" << endl;
    cout << "4. \033[1;36mExit\033[1;32m" << endl;
    cout << "\033[0m";
}

int main() {
    displayHotelSign();
    int choice = 0;

    while (choice != 4) {
```

```
        displayMainMenu();

        cin >> choice;

        if (choice == 1) {
            int password;
            cout << "\033[1;36m-------------------------------------
\033[1;32m" << endl;
            cout << "\033[1;32mEnter your password: \033[0m" << endl;
            cin >> password;
            cout << "\033[1;36m-------------------------------------
\033[1;32m" << endl;

            if (password == 230966) {
                Manager manager;
                int option;

                do {
                    cout << "======================" << endl;
                    cout << "||  \033[1;36mManager Options\033[1;32m  ||" <<
endl;
                    cout << "======================" << endl;
                    cout << "1. \033[1;36mView Employee Details\033[1;32m" <<
endl;
                    cout << "2. \033[1;36mView Rooms\033[1;32m" << endl;
                    cout << "3. \033[1;36mAdd Room\033[1;32m" << endl;
                    cout << "4. \033[1;36mEdit Room\033[1;32m" << endl;
                    cout << "5. \033[1;36mRemove Room\033[1;32m" << endl;
                    cout << "6. \033[1;36mView Hotel Details\033[1;32m" <<
endl;
                    cout << "7. \033[1;36mExit\033[1;32m" << endl;

                    cin >> option;

                    switch (option) {
                    case 1:
                        manager.viewEmployeeDetails();
                        cout << "\033[1;36m---------------------------------
----\033[1;32m" << endl;
                        break;
                    case 2:
                        manager.viewRooms();
                        cout << "\033[1;36m---------------------------------
----\033[1;32m" << endl;
                        break;
                    case 3:
                        manager.viewRooms();
                        manager.addRoom();
                        cout << "\033[1;36m---------------------------------
----\033[1;32m" << endl;
                        break;
                    case 4:
                        manager.viewRooms();
```

```
                        manager.editRoom();
                        cout << "\033[1;36m----------------------------------
----\033[1;32m" << endl;
                        break;
                    case 5:
                        manager.viewRooms();
                        manager.removeRoom();
                        break;
                    case 6:
                        manager.viewHotelDetails();
                        cout << "\033[1;36m----------------------------------
----\033[1;32m" << endl;
                        break;
                    case 7:
                        break;
                    default:
                        cout << "\033[1;31mInvalid choice. Please try
again.\033[0m" << endl;
                        cout << "\033[1;36m----------------------------------
----\033[1;32m" << endl;
                    }

                } while (option != 7);
            }
            else {
                cout << "\033[1;31mWrong Password\033[0m" << endl;
                cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
            }
        }
        else if (choice == 2) {
            Employee e;
            int employeeOption = 0;

            while (employeeOption != 3) {
                cout << "======================" << endl;
                cout << "|| \033[1;36mEmployee Options\033[1;32m ||" << endl;
                cout << "======================" << endl;
                cout << "1. \033[1;36mClock In\033[1;32m" << endl;
                cout << "2. \033[1;36mView Tasks\033[1;32m" << endl;
                cout << "3. \033[1;36mExit to Main Menu\033[1;32m" << endl;

                cin >> employeeOption;

                switch (employeeOption) {
                case 1:
                    e.setName();
                    e.setID();
                    e.setPosition();
                    e.setShift();

                    {
                        ofstream outFile("Hotel Info.txt", ios::app);
```

```
                        outFile << "Employee: " << endl;
                        outFile << "Name: " << e.getName() << endl;
                        outFile << "ID: " << e.getID() << endl;
                        outFile << "Position: " << e.getPosition() << endl;
                        outFile << "Shift: " << e.getShift() << endl;
                        outFile << endl;
                    }

                    e.displayEmployeeDetails();

                    cout << "You have clocked in. You may now start your
shift." << endl;
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    break;
                case 2:
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    e.displayTasks();
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    break;
                case 3:
                    cout << "\033[1;33mExiting to main menu.\033[0m" << endl;
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    break;
                default:
                    cout << "\033[1;31mInvalid choice. Please try
again.\033[0m" << endl;
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                }
            }
        }
        else if (choice == 3) {
            int customerOption = 0;

            while (customerOption != 3) {
                cout << "======================" << endl;
                cout << "|| \033[1;36mCustomer Options\033[1;32m ||" << endl;
                cout << "======================" << endl;
                cout << "1. \033[1;36mMake a reservation\033[1;32m" << endl;
                cout << "2. \033[1;36mCustomer Feedback\033[1;32m" << endl;
                cout << "3. \033[1;36mExit to Main Menu\033[1;32m" << endl;

                cin >> customerOption;

                switch (customerOption) {
                case 1: {
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    Room defaultRoom(1, Room::getRoomPrice(1));
```

```
                    Customer c("John Doe", 12345, "123 Main St", 30, "Male",
defaultRoom);

                    c.setCustomerInfo();
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    c.displayRoomOptions();
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    c.setRoomType();
                    {
                        ofstream outFile("Hotel Info.txt", ios::app);
                        outFile << "Customer: " << endl;
                        outFile << "Name: " << c.getName() << endl;
                        outFile << "Room Type: " <<
Room::getRoomName(c.getRoomType()) << endl;
                        outFile << "Room Price: " <<
Room::getRoomPrice(c.getRoomType()) << endl;
                        outFile << endl;
                    }

                    cout << "Your room type is: " <<
Room::getRoomName(c.getRoomType()) << endl;
                    cout << "Your name is: " << c.getName() << endl;
                    cout << "Your number is: " << c.getNumber() << endl;
                    cout << "Your address is: " << c.getAddress() << endl;
                    cout << "Your age is: " << c.getAge() << endl;
                    cout << "Your gender is: " << c.getGender() << endl;

                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    cout << "How many nights are you staying?" << endl;
                    int nights;
                    cin >> nights;
                    {
                        ofstream outFile("Hotel Info.txt", ios::app);
                        outFile << "Nights Staying: " << nights << endl;
                    }

                    double total = nights *
Room::getRoomPrice(c.getRoomType());

                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    cout << "Total: " << total << endl;
                    {
                        ofstream outFile("Hotel Info.txt", ios::app);
                        outFile << "Total Bill: " << total << endl;
                        cout << "\033[1;36m----------------------------------
----\033[1;32m" << endl;
                    }
                    cout << "Please make payment: ";
                    double payment;
```

```
                    cin >> payment;

                    while (payment < total) {
                        cout << "\033[1;36m--------------------------------
----\033[1;32m" << endl;
                        cout << "Insufficient payment. Please enter the
correct amount:" << endl;
                        cin >> payment;
                    }

                    if (payment > total) {
                        double change = payment - total;
                        cout << "Your change is: " << change << endl;
                        cout << "\033[1;36m--------------------------------
----\033[1;32m" << endl;
                    }

                    cout << "Payment processed. You may now proceed to your
room." << endl;
                    int additionalServiceChoice = 1;
                    while (additionalServiceChoice == 1) {
                        cout << "Would you like to select any additional
services?" << endl;
                        cout << "1. Yes" << endl;
                        cout << "2. No" << endl;
                        cin >> additionalServiceChoice;

                        if (additionalServiceChoice == 1) {
                            cout << "\033[1;36m------------------------------
--------\033[1;32m" << endl;
                            cout << "The following services are available:"
<< endl;
                            cout << "1. Room Service" << endl;
                            cout << "2. Valet Service" << endl;
                            cout << "3. Pet Service" << endl;
                            cout << "4. Laundry Service" << endl;
                            cout << "5. Gym Service" << endl;
                            cout << "\033[1;36m------------------------------
--------\033[1;32m" << endl;

                            cout << "Enter your choice:" << endl;
                            int serviceType;
                            cin >> serviceType;
                            cout << "\033[1;36m------------------------------
--------\033[1;32m" << endl;

                            switch (serviceType) {
                            case 1: {
                                RoomService roomService;
                                roomService.displayServices();
                                cout << "\033[1;36m--------------------------
------------\033[1;32m" << endl;
                                {
```

```
                                    ofstream outFile("Hotel Info.txt",
ios::app);
                                    outFile << "Service Type: Room Service"
<< endl;
                                }
                                break;
                            }

                            case 2: {
                                ValetService valetService;
                                valetService.setCarName();
                                valetService.setCarPlateNumber();
                                valetService.setCarColor();

                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                cout << "Car Name: " <<
valetService.getCarName() << endl;
                                cout << "Car Plate: " <<
valetService.getCarPlateNumber() << endl;
                                cout << "Car Color: " <<
valetService.getCarColor() << endl;

                                cout << "Your car will be brought to the
entrance. Please collect your keys from the hotel reception." << endl;
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                {
                                    ofstream outFile("Hotel Info.txt",
ios::app);
                                    outFile << "Service Type: Valet Service"
<< endl;
                                }

                                break;
                            }

                            case 3: {
                                PetService petService;
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                petService.setPetName();
                                petService.setPetType();
                                petService.setPetBreed();
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                petService.displayServices();
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                {
                                    ofstream outFile("Hotel Info.txt",
ios::app);
```

```
                                    outFile << "Service Type: Pet Service" <<
endl;
                                }
                                break;
                            }

                            case 4: {
                                LaundryService laundryService;
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                laundryService.displayServices();
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                {
                                    ofstream outFile("Hotel Info.txt",
ios::app);
                                    outFile << "Service Type: Laundry
Service" << endl;
                                }
                                break;
                            }

                            case 5: {
                                GymService gymService;
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                gymService.displayServices();
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;

                                {
                                    ofstream outFile("Hotel Info.txt",
ios::app);
                                    outFile << "Service Type: Gym Service" <<
endl;
                                }

                                break;
                            }

                            default:
                                cout << "\033[1;36m-------------------------
------------\033[1;32m" << endl;
                                cout << "\033[1;31mInvalid choice. Please try
again.\033[0m" << endl;
                            }
                        }
                    }
                }
                    break;

                case 2: {
                    string feedback;
```

```
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    cout << "Please enter your feedback:" << endl;
                    cin.ignore();
                    getline(cin, feedback);

                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    cout << "Thank you for your feedback! It is highly
appreciated." << endl;
                    {
                        ofstream outFile("Hotel Info.txt", ios::app);
                        outFile << "Feedback: " << feedback << endl;
                    }
                    break;
                }

                case 3:
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    cout << "\033[1;33mExiting to the main menu.\033[0m" <<
endl;
                    break;

                default:
                    cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
                    cout << "\033[1;31mInvalid choice. Please try
again.\033[0m" << endl;
                }
            }

        }
        else if (choice == 4) {
            cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
            cout << "\033[1;36mThank you for using our service.
Goodbye!\033[0m" << endl;
            cout << "\033[1;36m--------------------------------------
\033[1;32m" << endl;
            break;

        }
        else {
            cout << "\033[1;31mInvalid choice. Please try again.\033[0m" <<
endl;
        }
    }

    return 0;
}
```