**HOTEL RESERVATION SYSTEM**

**(Assignment # ……………………02)**

**Submitted To:**

Dr. Muhammad Usman Ali

**Submitted By:**

Eman Razi

**(Roll #......)**

22021519-151

**Semester:**

2nd (BS\_CS\_23)(A)

**Course Title:**

Object Oriented Programming

**Course Code:**

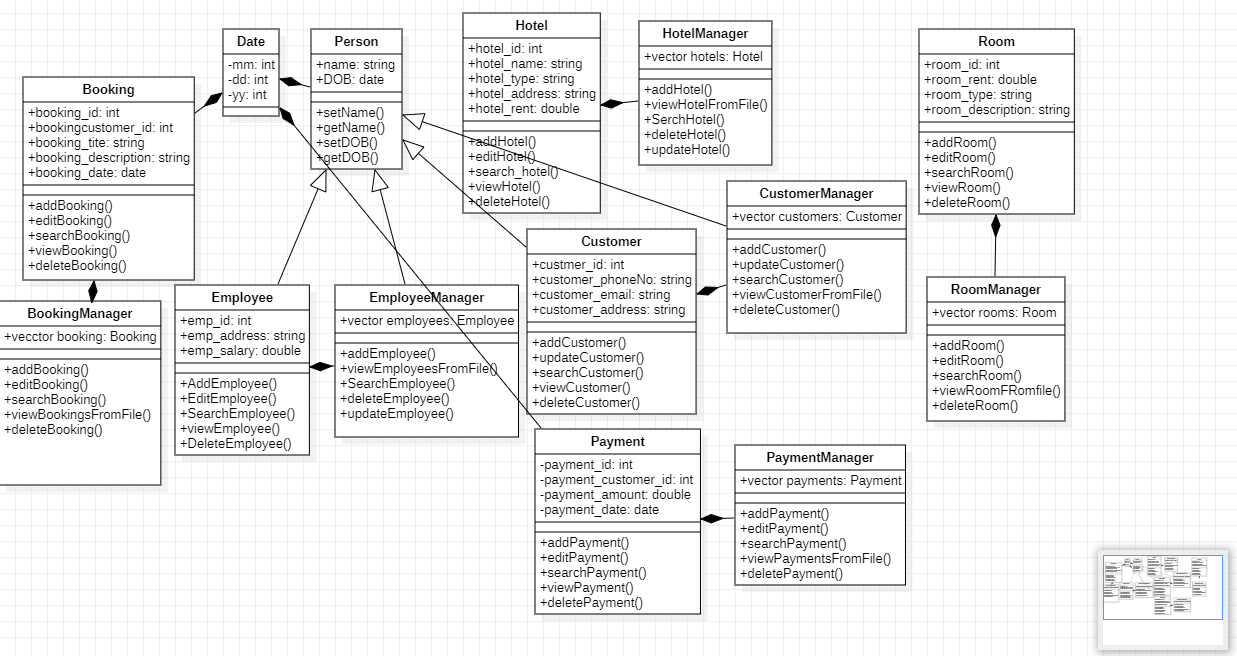
CS-103

**Department of Computer Science**

A picture containing text, logo, graphics, graphic design

Description automatically generated

**CLASS DIAGRAM**

****

**CODE**

**#include<iostream>**

**#include<stdlib.h>**

**#include<conio.h>**

**#include<fstream>**

**#include<vector>**

**#include<string.h>**

**#include<windows.h>**

**#include<graphics.h>**

**using namespace std;**

**void display1()**

**{**

**int gd = DETECT, gm;**

**initgraph(&gd, &gm, (char\*)"C:\\Users\\Razi\\Desktop\\cpp\\Project1oop.cpp");**

**//font**

**for (int i = 1; i < 400; i++)**

**{**

**settextstyle(1, 0, 4);//(font\_style,direction,size)**

**moveto(100 + i, 170);**

**setcolor(RED);**

**outtext((char\*)"HOTEL SYSTEM");**

**moveto(100, 170 + i);**

**setcolor(BLUE);**

**outtext((char\*)"HOTEL SYSTEM");**

**moveto(100 - i, 170);**

**setcolor(GREEN);**

**outtext((char\*)"HOTEL SYSTEM");**

**moveto(100, 170 - i);**

**setcolor(WHITE);**

**outtext((char\*)"HOTEL SYSTEM");**

**moveto(100, 170);**

**setcolor(YELLOW);**

**outtext((char\*)"HOTEL SYSTEM");**

**delay(20);**

**}**

**}**

**void display()**

**{**

**HANDLE color = GetStdHandle(STD\_OUTPUT\_HANDLE);**

**HANDLE size = GetStdHandle(STD\_OUTPUT\_HANDLE);**

**int k;**

**char arr[] = { 'E','N','C','O','D','E','D',' ','B','Y',':',' ','E','M','A','N',' ','R','A','Z','I' };**

**cout << "\n\t";**

**for (int i = 0; i < 21; i++)**

**{**

**k > 9 ? k = 0 : k++;**

**if (k == 0)**

**{**

**SetConsoleTextAttribute(color, 1);**

**}**

**else**

**{**

**SetConsoleTextAttribute(color, k);**

**}**

**cout << arr[i];**

**Sleep(300);**

**}**

**system("cls");**

**}**

**void gotoxy(short x, short y)**

**{**

**COORD pos = { x,y };**

**SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), pos);**

**}**

**void registerUser()**

**{**

**string username, password;**

**cout << "Enter a username: ";**

**cin >> username;**

**cout << endl;**

**cout << "Enter a password: ";**

**cin >> password; cout << endl;**

**// Open the file in append mode to store the new user credentials**

**std::ofstream file("C:\\Users\\Razi\\Desktop\\cpp\\user.txt", ios::app);**

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

**file.close();**

**cout << "Registration successful!" << std::endl;**

**}**

**bool loginUser()**

**{**

**string username, password;**

**cout << "Enter your username: ";**

**cin >> username;**

**cout << endl;**

**cout << "Enter your password: ";**

**cin >> password;**

**cout << endl;**

**ifstream file("C:\\Users\\Razi\\Desktop\\cpp\\user.txt");**

**string storedusername, storedPassword;**

**while (file >> storedusername >> storedPassword)**

**{**

**if (username == storedusername && password == storedPassword)**

**{**

**return true;**

**}**

**}**

**return false;**

**}**

**class date**

**{**

**public:**

**int mm;**

**int dd;**

**int yyyy;**

**public:**

**date() :mm(0), dd(0), yyyy(0)**

**{**

**}**

**};**

**class Person**

**{**

**public:**

**string name;**

**date DOB;**

**public:**

**date d;**

**Person()**

**{**

**}**

**void setName(string Name)**

**{**

**name = Name;**

**}**

**string getName()**

**{**

**return name;**

**}**

**void setDOB(date dob)**

**{**

**DOB = dob;**

**}**

**date getDOB()**

**{**

**return DOB;**

**}**

**};**

**class Hotel**

**{**

**public:**

**int hotel\_id;**

**string hotel\_name;**

**string hotel\_type;**

**string hotel\_address;**

**double hotel\_rent;**

**vector<int>HotelIDs;**

**public:**

**static void HotelMenu();**

**Hotel()**

**{**

**hotel\_id = 0;**

**hotel\_rent = 0;**

**hotel\_name = '\0';**

**hotel\_type = '\0';**

**hotel\_address = '\0';**

**}**

**void addHotel()**

**{**

**int new\_hotel\_id;**

**cout << "Enter hotel Id: ";**

**cin >> new\_hotel\_id;**

**cin.ignore();**

**if (isHotelID(new\_hotel\_id))**

**{**

**cout << "Error: Hotel with ID " << new\_hotel\_id << " already exists and is occupied." << endl;**

**return;**

**}**

**hotel\_id = new\_hotel\_id;**

**HotelIDs.push\_back(new\_hotel\_id);**

**string var;**

**while (true)**

**{**

**cout << "Hotel Rent: ";**

**getline(cin, var);**

**stringstream ss(var);**

**if (!(ss >> hotel\_rent) || !(ss.eof()))**

**{**

**cout << "Invalid! Please enter an integer value." << endl;**

**continue;**

**}**

**else**

**break;**

**}**

**cout << "Hotel Name: ";**

**getline(cin, hotel\_name);**

**cout << "Hotel Type: ";**

**getline(cin, hotel\_type);**

**cout << "Hotel Address: ";**

**getline(cin, hotel\_address);**

**cout << "Hotel Added Successfully" << endl;**

**WriteDatatoFile();**

**}**

**bool isHotelID(int id)**

**{**

**for (int hotel\_id : HotelIDs)**

**{**

**if (hotel\_id == id)**

**{**

**return true;**

**}**

**}**

**return false;**

**}**

**void WriteDatatoFile()**

**{**

**ofstream WriteFile("C:\\Users\\Razi\\Desktop\\cpp\\hotel.txt", ios::app);**

**if (!WriteFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**else**

**{**

**WriteFile << "Hotel ID: " << hotel\_id << endl;**

**WriteFile << "Hotel Name: " << hotel\_name << endl;**

**WriteFile << "Hotel Rent: " << hotel\_rent << endl;**

**WriteFile << "Hotel Type: " << hotel\_type << endl;**

**WriteFile << "Hotel Address: " << hotel\_address << endl;**

**}**

**cout << "Data has wriiten successully" << endl;**

**WriteFile.close();**

**}**

**void viewHotel() {**

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

**cout << "Hotel ID: " << hotel\_id << endl;**

**cout << "Hotel Name: " << hotel\_name << endl;**

**cout << "Hotel Rent: " << hotel\_rent << endl;**

**cout << "Hotel Type: " << hotel\_type << endl;**

**cout << "Hotel Address: " << hotel\_address << endl;**

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

**cout << "Read Data From file Successfully" << endl;**

**}**

**void SearchHotel()**

**{**

**cout << "Search found" << endl;**

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

**cout << "Hotel ID: " << hotel\_id << endl;**

**cout << "Hotel Name: " << hotel\_name << endl;**

**cout << "Hotel Rent: " << hotel\_rent << endl;**

**cout << "Hotel Type: " << hotel\_type << endl;**

**cout << "Hotel Address: " << hotel\_address << endl;**

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

**}**

**void editHotel()**

**{**

**}**

**void deleteHotel()**

**{**

**}**

**};**

**class HotelManager {**

**public:**

**vector<Hotel> hotels;**

**Hotel h; //composition::Hotel has a hotel MAnager**

**public:**

**void addHotel() {**

**h.addHotel();**

**hotels.push\_back(h);**

**}**

**void WriteDatatoFile()**

**{**

**for (auto& hotel : hotels) {**

**hotel.WriteDatatoFile();**

**}**

**}**

**void viewHotelsFromFile() {**

**if (hotels.empty())**

**{**

**cout << "No record found!" << endl;**

**}**

**ifstream ReadFile("C:\\Users\\Razi\\Desktop\\cpp\\hotel.txt");**

**if (!ReadFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**int hotelId;**

**double hotelRent;**

**string hotelName, hotelType, hotelAddress;**

**while (ReadFile >> hotelId >> hotelName >> hotelRent >> hotelType >> hotelAddress) {**

**Hotel h; // Create a new hotel object**

**h.hotel\_id = hotelId;**

**h.hotel\_name = hotelName;**

**h.hotel\_rent = hotelRent;**

**h.hotel\_type = hotelType;**

**h.hotel\_address = hotelAddress;**

**hotels.push\_back(h); // Add the hotel object to the vector**

**}**

**ReadFile.close();**

**}**

**void SearchHotel()**

**{**

**int searchId;**

**cout << "Enter the Id you Want To Search " << endl;**

**cin >> searchId;**

**bool found = false;**

**if (hotels.empty())**

**{**

**cout << "NO RECORD FOUND" << endl;**

**}**

**for (auto& hotel : hotels) {**

**if (searchId == hotel.hotel\_id)**

**{**

**hotel.SearchHotel();**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout << "invalid id!" << endl;**

**}**

**}**

**void editHotel()**

**{**

**}**

**void deleteHotel()**

**{**

**}**

**};**

**void Hotel::HotelMenu()**

**{**

**int hotelchoice;**

**char choice;**

**HotelManager hotelobj;**

**Hotel h;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tHOTEL MENU" << endl;**

**cout << "\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. ADD HOTEL" << endl;**

**cout << "\n\t\t2. EDIT HOTEL" << endl;**

**cout << "\n\t\t3. SEARCH HOTEL" << endl;**

**cout << "\n\t\t4. VIEW HOTEL " << endl;**

**cout << "\n\t\t5. DELETE HOTEL" << endl;**

**cout << "\n\t\t0. EXIT" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> hotelchoice;**

**system("cls");**

**switch (hotelchoice)**

**{**

**case 1:**

**{**

**do**

**{**

**hotelobj.addHotel();**

**cout << "Do you want to add more record (y/n)";**

**cin >> choice;**

**cin.ignore();**

**system("cls");**

**} while (choice == 'y' || choice == 'Y');**

**}**

**case 2:**

**{**

**hotelobj.editHotel();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 3:**

**{**

**hotelobj.SearchHotel();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 4:**

**{**

**hotelobj.viewHotelsFromFile();**

**for (auto& h : hotelobj.hotels) {**

**h.viewHotel();**

**}**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 5:**

**{**

**hotelobj.deleteHotel();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 0:**

**break;**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (hotelchoice != 0);**

**}**

**class Employee :public Person**

**{**

**public:**

**int emp\_id;**

**string emp\_address;**

**string emp\_dept;**

**double emp\_salary;**

**public:**

**Employee()**

**{**

**emp\_id = 0;**

**emp\_salary = 0;**

**emp\_address = '\0';**

**emp\_dept = "";**

**}**

**int getEmployeeID()**

**{**

**return emp\_id;**

**}**

**double getEmployeeSalary()**

**{**

**return emp\_salary;**

**}**

**string getEmpDepartment()**

**{**

**return emp\_dept;**

**}**

**string getEmployeeAddress()**

**{**

**return emp\_address;**

**}**

**void setEmployeeID(int id)**

**{**

**emp\_id = id;**

**}**

**void setEmployeeSalary(double salary) {**

**emp\_salary = salary;**

**}**

**string setEmpDepartment(string dept)**

**{**

**emp\_dept = dept;**

**}**

**void setEmployeeAddress(string address) {**

**emp\_address = address;**

**}**

**static void EmployeeMenu();**

**void addEmployee()**

**{**

**cout << "Enter Employee ID: ";**

**cin >> emp\_id;**

**cin.ignore();**

**cout << "Enter name :";**

**getline(cin, name);**

**cin.ignore();**

**cout << "Enter DOB(dd-mm-yyyy):";**

**cin >> d.dd >> d.mm >> d.yyyy;**

**cin.ignore();**

**cout << endl;**

**cout << "Enter Employee salary: ";**

**cin >> emp\_salary;**

**cin.ignore();**

**cout << "Enter Employee Department: ";**

**getline(cin, emp\_dept);**

**cin.ignore();**

**cout << "Enter Employee address:";**

**getline(cin, emp\_address);**

**WriteDatatoFile();**

**}**

**void WriteDatatoFile()**

**{**

**ofstream WriteFile("C:\\Users\\Razi\\Desktop\\cpp\\Employee.txt", ios::app);**

**if (!WriteFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**else**

**{**

**WriteFile << "Employee ID: " << emp\_id << endl;**

**WriteFile << "Employee Name: " << name << endl;**

**WriteFile << "Employee DOB : " << d.dd << d.mm << d.yyyy << endl;**

**WriteFile << "Employee Salary: " << emp\_salary << endl;**

**WriteFile << "Employee Dept: " << emp\_dept << endl;**

**WriteFile << "Employee Address: " << emp\_address << endl;**

**WriteFile << "\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**cout << "Data has wriiten successully" << endl;**

**WriteFile.close();**

**}**

**void viewEmployee()**

**{**

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

**cout << "Employee Id: " << getEmployeeID() << endl;**

**cout << "Employee Name: " << getName() << endl;**

**cout << "Date Of Birth: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**cout << "Employee Salary: " << getEmployeeSalary() << endl;**

**cout << "Employee Department: " << getEmpDepartment() << endl;**

**cout << "Employee Address : " << getEmployeeAddress() << endl;**

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

**cout << "Raed data from file successfully" << endl;**

**}**

**void SearchEmployee()**

**{**

**cout << "Search found" << endl;**

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

**cout << "Employee ID: " << emp\_id << endl;**

**cout << "Employee Name: " << name << endl;**

**cout << "Date Of Birth: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**cout << "Employee salary: " << emp\_salary << endl;**

**cout << "Employee Department: " << emp\_dept << endl;**

**cout << "Employee Address: " << emp\_address << endl;**

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

**}**

**void editEmployee()**

**{**

**}**

**void deleteEmployee()**

**{**

**}**

**};**

**class EmlpoyeeManager : Person**

**{**

**public:**

**Employee emp;**

**vector<Employee>employees;**

**void addEmployee()**

**{**

**emp.addEmployee();**

**employees.push\_back(emp);**

**}**

**void WriteDatatoFile()**

**{**

**for (auto& employee : employees)**

**{**

**employee.WriteDatatoFile();**

**}**

**}**

**void viewEmployeesFromFile()**

**{**

**if (employees.empty())**

**{**

**cout << "No record found!" << endl;**

**}**

**ifstream ReadFile("C:\\Users\\Razi\\Desktop\\cpp\\Employee.txt");**

**if (!ReadFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**int empId;**

**double empSalary;**

**date d;**

**string empName, empDepartment, empAddress;**

**while (ReadFile >> empId >> empName >> d.dd >> d.mm >> d.yyyy >> empSalary >> empDepartment >> empAddress) {**

**Employee e; // Create a new employee object**

**e.emp\_id = empId;**

**e.name = empName;**

**e.emp\_salary = empSalary;**

**e.emp\_dept = empDepartment;**

**e.emp\_address = empAddress;**

**employees.push\_back(e); // Add the employee object to the vector**

**}**

**ReadFile.close();**

**}**

**void SearchEmployee()**

**{**

**int searchId;**

**cout << "Enter the Id you Want To Search " << endl;**

**cin >> searchId;**

**bool found = false;**

**if (employees.empty())**

**{**

**cout << "NO RECORD FOUND" << endl;**

**}**

**for (auto& employee : employees) {**

**if (searchId == employee.getEmployeeID())**

**{**

**employee.SearchEmployee();**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout << "Invalid id!" << endl;**

**}**

**}**

**void editEmployee()**

**{**

**}**

**void deleteEmployee()**

**{**

**}**

**};**

**void Employee::EmployeeMenu()**

**{**

**int employeechoice;**

**char choice;**

**EmlpoyeeManager empobj;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tEMPLOYEE MENU" << endl;**

**cout << "\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. ADD EMPLOYEE" << endl;**

**cout << "\n\t\t2. EDIT EMPLOYEE" << endl;**

**cout << "\n\t\t3. SEARCH EMPLOYEE" << endl;**

**cout << "\n\t\t4. VIEW EMPLOYEE " << endl;**

**cout << "\n\t\t5. DELETE EMPLOYEE" << endl;**

**cout << "\n\t\t0. BACK TO MENU" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> employeechoice;**

**system("cls");**

**switch (employeechoice)**

**{**

**case 1:**

**{**

**do**

**{**

**empobj.addEmployee();**

**cout << "Do you want to add more record (y/n)";**

**cin >> choice;**

**cin.ignore();**

**system("cls");**

**} while (choice == 'y' || choice == 'Y');**

**break;**

**}**

**case 2:**

**{**

**empobj.editEmployee();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 3:**

**{**

**empobj.SearchEmployee();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 4:**

**{**

**empobj.viewEmployeesFromFile();**

**for (auto& e : empobj.employees) {**

**e.viewEmployee();**

**}**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 5:**

**{**

**empobj.deleteEmployee();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 0:**

**break;**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (employeechoice != 0);**

**}**

**class Room**

**{**

**public:**

**int room\_id;**

**double room\_rent;**

**string room\_type;**

**string room\_description;**

**vector<int> roomIDs;**

**bool is\_availabe;**

**public:**

**Room()**

**{**

**room\_id = 0;**

**room\_rent = 0;**

**room\_type = '\0';**

**room\_description = '\0';**

**}**

**static void RoomMenu();**

**void addRoom()**

**{**

**int new\_room\_id;**

**cout << "Enter Room ID: ";**

**cin >> new\_room\_id;**

**if (isRoomIDTaken(new\_room\_id))**

**{**

**cout << "Error: Room with ID " << new\_room\_id << " already exists and is occupied." << endl;**

**return;**

**}**

**room\_id = new\_room\_id;**

**roomIDs.push\_back(new\_room\_id);**

**cout << "Enter Room rent: ";**

**cin >> room\_rent;**

**cin.ignore();**

**cout << "Enter Room description: ";**

**getline(cin, room\_description);**

**cout << "Enter Room Type : ";**

**getline(cin, room\_type);**

**WriteDatatoFile();**

**}**

**bool isRoomIDTaken(int id)**

**{**

**for (int room\_id : roomIDs)**

**{**

**if (room\_id == id)**

**{**

**return true;**

**}**

**}**

**return false;**

**}**

**void WriteDatatoFile()**

**{**

**ofstream WriteFile("C:\\Users\\Razi\\Desktop\\cpp\\room.txt", ios::app);**

**if (!WriteFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**else**

**{**

**WriteFile << "Room ID : " << room\_id << endl;**

**WriteFile << "Room Rent : " << room\_rent << endl;**

**WriteFile << "Room Type : " << room\_type << endl;**

**WriteFile << "Room Description: " << room\_description << endl;**

**WriteFile << "\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**cout << "Data has wriiten successully" << endl;**

**WriteFile.close();**

**}**

**void viewRoom()**

**{**

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

**cout << "Room Id: " << room\_id << endl;**

**cout << "Room Rent: " << room\_rent << endl;**

**cout << "Room Type: " << room\_type << endl;**

**cout << "Room Description: " << room\_description << endl;**

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

**cout << "Read data from file Successfuly" << endl;**

**}**

**void SearchRoom()**

**{**

**cout << "Search found" << endl;**

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

**cout << "Room Id: " << room\_id << endl;**

**cout << "Room Rent: " << room\_rent << endl;**

**cout << "Room Type: " << room\_type << endl;**

**cout << "Room Description: " << room\_description << endl;**

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

**}**

**void editRoom()**

**{**

**}**

**void deleteRoom()**

**{**

**}**

**};**

**class RoomManager**

**{**

**public:**

**Room room;**

**vector<Room>rooms;**

**void addRoom()**

**{**

**room.addRoom();**

**rooms.push\_back(room);**

**}**

**void viewRoomFromFile()**

**{**

**if (rooms.empty())**

**{**

**cout << "No record found!" << endl;**

**}**

**ifstream ReadFile("C:\\Users\\Razi\\Desktop\\cpp\\room.txt");**

**if (!ReadFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**int roomId;**

**double room\_Rent;**

**string room\_Type, room\_Description;**

**while (ReadFile >> roomId >> room\_Rent >> room\_Type >> room\_Description) {**

**Room r; // Create a new room object**

**r.room\_id = roomId;**

**r.room\_rent = room\_Rent;**

**r.room\_type = room\_Type;**

**r.room\_description = room\_Description;**

**rooms.push\_back(r); // Add the room object to the vector**

**}**

**ReadFile.close();**

**}**

**void SearchRoom()**

**{**

**int searchId;**

**cout << "Enter the Id you Want To Search " << endl;**

**cin >> searchId;**

**bool found = false;**

**if (rooms.empty())**

**{**

**cout << "NO RECORD FOUND" << endl;**

**}**

**for (auto& room : rooms) {**

**if (searchId == room.room\_id)**

**{**

**room.SearchRoom();**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout << "Invalid id!" << endl;**

**}**

**}**

**};**

**void Room::RoomMenu()**

**{**

**int roomchoice;**

**char choice;**

**RoomManager roomobj;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tROOM MENU" << endl;**

**cout << "\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. ADD ROOM" << endl;**

**cout << "\n\t\t2. EDIT ROOM" << endl;**

**cout << "\n\t\t3. SEARCH ROOM" << endl;**

**cout << "\n\t\t4. VIEW ROOM " << endl;**

**cout << "\n\t\t5. DELETE ROOM" << endl;**

**cout << "\n\t\t0. BACK TO MENU" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> roomchoice;**

**system("cls");**

**switch (roomchoice)**

**{**

**case 1:**

**{**

**do**

**{**

**roomobj.addRoom();**

**cout << "Do you want to add more record (y/n)";**

**cin >> choice;**

**cin.ignore();**

**system("cls");**

**} while (choice == 'y' || choice == 'Y');**

**break;**

**}**

**/\*case 2:**

**{**

**roomobj.editRoom();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 3:**

**{**

**roomobj.SearchRoom();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 4:**

**{**

**roomobj.viewRoomFromFile();**

**for (auto& r : roomobj.rooms) {**

**r.viewRoom();**

**}**

**system("pause");**

**system("cls");**

**break;**

**}**

**/\* case 5:**

**{**

**roomobj.deleteRoom();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 0:**

**break;**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (roomchoice != 0);**

**}**

**class Payment**

**{**

**public:**

**int payment\_id;**

**vector<int>PaymentIDs;**

**int payment\_custom\_id;**

**vector<int>PayCustomIDs;**

**double payment\_amount;**

**date payment\_date;**

**date d;**

**public:**

**Payment()**

**{**

**payment\_id = 0;**

**payment\_custom\_id = 0;**

**payment\_amount = 0;**

**}**

**static void PaymentMenu();**

**void addPayment()**

**{**

**int new\_payment\_id;**

**cout << " Payment Id: ";**

**cin >> new\_payment\_id;**

**cin.ignore();**

**if (isPaymentID(new\_payment\_id))**

**{**

**cout << "Error: paymentt with ID " << new\_payment\_id << " already paid." << endl;**

**return;**

**}**

**payment\_id = new\_payment\_id;**

**PaymentIDs.push\_back(new\_payment\_id);**

**string var;**

**while (true)**

**{**

**cout << "Payment Amount: ";**

**getline(cin, var);**

**stringstream ss(var);**

**if (!(ss >> payment\_amount) || !(ss.eof()))**

**{**

**cout << "Invalid! Please enter numeric value." << endl;**

**continue;**

**}**

**else**

**break;**

**}**

**int new\_custom\_id;**

**cout << "payment Customer Id: ";**

**cin >> new\_custom\_id;**

**cin.ignore();**

**if (isPaymentCustomID(new\_custom\_id))**

**{**

**cout << "Error: paymentt with ID " << new\_custom\_id << " already paid." << endl;**

**return;**

**}**

**payment\_custom\_id = new\_custom\_id;**

**PaymentIDs.push\_back(new\_payment\_id);**

**cout << "Payment Date: " << endl;**

**cin >> d.dd >> d.mm >> d.yyyy;**

**cout << "PAyment Added Successfully" << endl;**

**WriteDatatoFile();**

**}**

**bool isPaymentID(int id)**

**{**

**for (int payment\_id : PaymentIDs)**

**{**

**if (payment\_id == id)**

**{**

**return true;**

**}**

**}**

**return false;**

**}**

**bool isPaymentCustomID(int id)**

**{**

**for (int payment\_custom\_id : PayCustomIDs)**

**{**

**if (payment\_custom\_id == id)**

**{**

**return true;**

**}**

**}**

**return false;**

**}**

**void WriteDatatoFile()**

**{**

**ofstream WriteFile("C:\\Users\\Razi\\Desktop\\cpp\\Payment.txt", ios::app);**

**if (!WriteFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**else**

**{**

**WriteFile << "payment ID: " << payment\_id << endl;**

**WriteFile << "Payment Customer Id: " << payment\_custom\_id << endl;**

**WriteFile << "Payment Amount: " << payment\_amount << endl;**

**WriteFile << "Payment date: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

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

**}**

**cout << "Data has written successully" << endl;**

**WriteFile.close();**

**}**

**void viewPayment()**

**{**

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

**cout << "payment ID: " << payment\_id << endl;**

**cout << "Payment Customer Id: " << payment\_custom\_id << endl;**

**cout << "Payment Amount: " << payment\_amount << endl;**

**cout << "Payment date: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

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

**cout << "Read Data From file Successfully" << endl;**

**}**

**void SearchPayment()**

**{**

**cout << "Search found!" << endl;**

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

**cout << "payment ID: " << payment\_id << endl;**

**cout << "Payment Customer Id: " << payment\_custom\_id << endl;**

**cout << "Payment Amount: " << payment\_amount << endl;**

**cout << "Payment date: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

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

**}**

**void editPayment()**

**{**

**}**

**void deletePayment()**

**{**

**}**

**};**

**class PaymentManager**

**{**

**public:**

**vector<Payment>payments;**

**Payment p;**

**void addPayment()**

**{**

**p.addPayment();**

**payments.push\_back(p);**

**}**

**void viewPaymentsFromFile()**

**{**

**if (payments.empty())**

**{**

**cout << "No record found!" << endl;**

**}**

**ifstream ReadFile("C:\\Users\\Razi\\Desktop\\cpp\\Payment.txt");**

**if (!ReadFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**int paymentId;**

**double paymentAmount;**

**int paymentCustomID;**

**date paymentDate;**

**date d;**

**while (ReadFile >> paymentId >> paymentCustomID >> paymentAmount >> d.dd >> d.mm >> d.yyyy) {**

**Payment p; // Create a new payment object**

**p.payment\_id = paymentId;**

**p.payment\_custom\_id = paymentCustomID;**

**p.payment\_amount = paymentAmount;**

**p.payment\_date = paymentDate;**

**payments.push\_back(p); // Add the payment object to the vector**

**}**

**ReadFile.close();**

**}**

**void SearchPayment()**

**{**

**int searchId;**

**cout << "Enter the Id you Want To Search " << endl;**

**cin >> searchId;**

**bool found;**

**if (payments.empty())**

**{**

**cout << "NO RECORD FOUND" << endl;**

**}**

**for (auto& payment : payments) {**

**if (searchId == payment.payment\_id)**

**{**

**payment.SearchPayment();**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout << "Invalid id!" << endl;**

**}**

**}**

**};**

**void Payment::PaymentMenu()**

**{**

**int paymentchoice;**

**char choice;**

**PaymentManager paymentobj;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tPAYMENT MENU" << endl;**

**cout << "\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. ADD PAYMENT" << endl;**

**cout << "\n\t\t2. EDIT PAYMENT" << endl;**

**cout << "\n\t\t3. SEARCH PAYMENT" << endl;**

**cout << "\n\t\t4. VIEW PAYMENT " << endl;**

**cout << "\n\t\t5. DELETE PAYMENT" << endl;**

**cout << "\n\t\t0. BACK TO MENU" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> paymentchoice;**

**system("cls");**

**switch (paymentchoice)**

**{**

**case 1:**

**{**

**do**

**{**

**paymentobj.addPayment();**

**cout << "Do you want to add more record (y/n)";**

**cin >> choice;**

**cin.ignore();**

**system("cls");**

**} while (choice == 'y' || choice == 'Y');**

**break;**

**}**

**/\*case 2:**

**{**

**paymentobj.editPayment();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 3:**

**{**

**paymentobj.SearchPayment();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 4:**

**{**

**paymentobj.viewPaymentsFromFile();**

**for (auto& p : paymentobj.payments) {**

**p.viewPayment();**

**}**

**system("pause");**

**system("cls");**

**break;**

**}**

**/\* case 5:**

**{**

**paymentobj.deletePayment();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 0:**

**break;**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (paymentchoice != 0);**

**}**

**class Customer :public Person**

**{**

**public:**

**int customer\_id;**

**date customer\_dob;**

**string customer\_address;**

**string phone\_no;**

**string customer\_email;**

**public:**

**Customer()**

**{**

**customer\_id = 0;**

**customer\_address = "";**

**phone\_no = "";**

**customer\_email = "";**

**}**

**int getCustomerID()**

**{**

**return customer\_id;**

**}**

**string getCustomerphoneNo()**

**{**

**return phone\_no;**

**}**

**string getCustomerAddress()**

**{**

**return customer\_address;**

**}**

**string getCustomerEmail()**

**{**

**return customer\_email;**

**}**

**void setCustomerPhoneNo(string PhoneNo) {**

**phone\_no = PhoneNo;**

**}**

**void setCustomerEmail(string email)**

**{**

**customer\_email = email;**

**}**

**void setCustomerAddress(string address)**

**{**

**customer\_address = address;**

**}**

**void setCustomerId(int id) {**

**customer\_id = id;**

**}**

**static void CustomerMenu();**

**void addCustomer()**

**{**

**cout << "Enter Customer ID: ";**

**cin >> customer\_id;**

**cin.ignore();**

**cout << "Enter name :";**

**getline(cin, name);**

**cin.ignore();**

**cout << "Enter DOB(dd-mm-yyyy):";**

**cin >> d.dd >> d.mm >> d.yyyy;**

**cin.ignore();**

**cout << endl;**

**cout << "Enter Customer Phone NO: ";**

**cin >> phone\_no;**

**cin.ignore();**

**cout << "Enter Customer Email: ";**

**getline(cin, customer\_email);**

**cin.ignore();**

**cout << "Enter Customer address:";**

**getline(cin, customer\_address);**

**WriteDatatoFile();**

**}**

**void WriteDatatoFile()**

**{**

**ofstream WriteFile("C:\\Users\\Razi\\Desktop\\cpp\\Customer.txt", ios::app);**

**if (!WriteFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**else**

**{**

**WriteFile << "Customer ID: " << customer\_id << endl;**

**WriteFile << "CUstomer Name: " << name << endl;**

**WriteFile << "CUstomer DOB : " << d.dd << d.mm << d.yyyy << endl;**

**WriteFile << "CUstomer PhoneNO: " << phone\_no << endl;**

**WriteFile << "CUstomer Email: " << customer\_email << endl;**

**WriteFile << "Customer Address: " << customer\_address << endl;**

**WriteFile << "\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**cout << "Data has wriiten successully" << endl;**

**WriteFile.close();**

**}**

**void viewCustomer()**

**{**

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

**cout << "Customer ID: " << getCustomerID() << endl;**

**cout << "Customer Name: " << getName() << endl;**

**cout << "Customer DOB: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**cout << "Customer PhoneNO: " << getCustomerphoneNo() << endl;**

**cout << "Customer Email: " << getCustomerEmail() << endl;**

**cout << "Customer Address : " << getCustomerAddress() << endl;**

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

**cout << "Raed data from file successfully" << endl;**

**}**

**void SearchCustomer()**

**{**

**cout << "Search found!" << endl;**

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

**cout << "Customer ID: " << customer\_id << endl;**

**cout << "Customer Name: " << name << endl;**

**cout << "Customer DOB: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**cout << "Customer PhoneNO: " << phone\_no << endl;**

**cout << "Customer Email: " << customer\_email << endl;**

**cout << "Customer Address : " << customer\_address << endl;**

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

**}**

**void editCustomer()**

**{**

**}**

**void deleteCustomer()**

**{**

**}**

**};**

**class CustomerManager :public Person**

**{**

**public:**

**Customer custom;**

**vector<Customer>customers;**

**void addCustomer()**

**{**

**custom.addCustomer();**

**customers.push\_back(custom);**

**}**

**void WriteDatatoFile()**

**{**

**for (auto& customer : customers)**

**{**

**customer.WriteDatatoFile();**

**}**

**}**

**void viewCustomersFromFile()**

**{**

**if (customers.empty())**

**{**

**cout << "No record found!" << endl;**

**}**

**ifstream ReadFile("C:\\Users\\Razi\\Desktop\\cpp\\Customer.txt");**

**if (!ReadFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**int CustomId;**

**date customDOB;**

**date d;**

**string customName, customphoneNo, customAddress, customEmail;**

**while (ReadFile >> CustomId >> customName >> d.dd >> d.mm >> d.yyyy >> customphoneNo >> customEmail >> customAddress) {**

**Customer c; // Create a new customer object**

**c.customer\_id = CustomId;**

**c.name = customName;**

**c.customer\_dob = customDOB;**

**c.phone\_no = customphoneNo;**

**c.customer\_email = customEmail;**

**c.customer\_address = customAddress;**

**customers.push\_back(c); // Add the customer object to the vector**

**}**

**ReadFile.close();**

**}**

**void SearchCustomer()**

**{**

**int searchId;**

**cout << "Enter the Id you Want To Search " << endl;**

**cin >> searchId;**

**bool found = false;**

**if (customers.empty())**

**{**

**cout << "NO RECORD FOUND" << endl;**

**}**

**for (auto& customer : customers) {**

**if (searchId == customer.getCustomerID())**

**{**

**customer.SearchCustomer();**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout << "Invalid id!" << endl;**

**}**

**}**

**};**

**void Customer::CustomerMenu()**

**{**

**int customerchoice;**

**char choice;**

**CustomerManager customerobj;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tCUSTOMER MENU" << endl;**

**cout << "\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. ADD CUSTOMER" << endl;**

**cout << "\n\t\t2. EDIT CUSTOMER" << endl;**

**cout << "\n\t\t3. SEARCH CUSTOMER" << endl;**

**cout << "\n\t\t4. VIEW CUSTOMER " << endl;**

**cout << "\n\t\t5. DELETE CUSTOMER" << endl;**

**cout << "\n\t\t0. EXIT" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> customerchoice;**

**system("cls");**

**switch (customerchoice)**

**{**

**case 1:**

**{**

**do**

**{**

**customerobj.addCustomer();**

**cout << "Do you want to add more record (y/n)";**

**cin >> choice;**

**cin.ignore();**

**system("cls");**

**} while (choice == 'y' || choice == 'Y');**

**break;**

**}**

**/\* case 2:**

**{**

**customerobj.editCustomer();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 3:**

**{**

**customerobj.SearchCustomer();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 4:**

**{**

**customerobj.viewCustomersFromFile();**

**for (auto& c : customerobj.customers) {**

**c.viewCustomer();**

**}**

**system("pause");**

**system("cls");**

**break;**

**}**

**/\* case 5:**

**{**

**customerobj.deleteCustomer();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 0:**

**break;**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (customerchoice != 0);**

**}**

**class Booking**

**{**

**public:**

**int booking\_id;**

**vector<int>BookingIDs;**

**int bookingcustomer\_id;**

**string booking\_title;**

**string booking\_description;**

**date booking\_date;**

**date d;**

**public:**

**Booking()**

**{**

**booking\_id = 0;**

**bookingcustomer\_id = 0;**

**booking\_title = '\0';**

**booking\_description = '\0';**

**}**

**static void BookingMenu();**

**void addBooking()**

**{**

**int new\_booking\_id;**

**cout << "Enter Booking Id: ";**

**cin >> new\_booking\_id;**

**cin.ignore();**

**if (isBookingID(new\_booking\_id))**

**{**

**cout << "Error: Hotel with ID " << new\_booking\_id << " already exists and is occupied." << endl;**

**return;**

**}**

**booking\_id = new\_booking\_id;**

**BookingIDs.push\_back(new\_booking\_id);**

**cout << "Booking Customer Id: ";**

**cin >> bookingcustomer\_id;**

**cin.ignore();**

**cout << "Booking Date : ";**

**cin >> d.dd >> d.mm >> d.yyyy;**

**cout << endl;**

**cin.ignore();**

**cout << "Booking Title: ";**

**getline(cin, booking\_title);**

**cout << "Booking Description: ";**

**getline(cin, booking\_description);**

**cout << "Booking Added Successfully" << endl;**

**WriteDatatoFile();**

**}**

**bool isBookingID(int id)**

**{**

**for (int booking\_id : BookingIDs)**

**{**

**if (booking\_id == id)**

**{**

**return true;**

**}**

**}**

**return false;**

**}**

**void WriteDatatoFile()**

**{**

**ofstream WriteFile("C:\\Users\\Razi\\Desktop\\cpp\\Booking.txt", ios::app);**

**if (!WriteFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**else**

**{**

**WriteFile << "Booking ID: " << booking\_id << endl;**

**WriteFile << "Booking Customer Id: " << bookingcustomer\_id << endl;**

**WriteFile << "Booking Date: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**WriteFile << "Booking Title: " << booking\_title << endl;**

**WriteFile << "Booking description: " << booking\_description << endl;**

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

**}**

**cout << "Data has wriiten successully" << endl;**

**WriteFile.close();**

**}**

**void viewBooking() {**

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

**cout << "Booking ID: " << booking\_id << endl;**

**cout << "Booking Customer Id: " << bookingcustomer\_id << endl;**

**cout << "Booking Date: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**cout << "Booking Title: " << booking\_title << endl;**

**cout << "Booking description: " << booking\_description << endl;**

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

**cout << "Read Data From file Successfully" << endl;**

**}**

**void SearchBooking()**

**{**

**cout << "Search found!" << endl;**

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

**cout << "Booking ID: " << booking\_id << endl;**

**cout << "Booking Customer Id: " << bookingcustomer\_id << endl;**

**cout << "Booking Date: " << d.dd << "-" << d.mm << "-" << d.yyyy << endl;**

**cout << "Booking Title: " << booking\_title << endl;**

**cout << "Booking description: " << booking\_description << endl;**

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

**}**

**void editBooking()**

**{**

**}**

**void deleteBooking()**

**{**

**}**

**};**

**class BookingManager**

**{**

**public:**

**date d;**

**vector<Booking> bookings;**

**Booking b; //composition::Hotel has a hotel MAnager**

**public:**

**void addBooking() {**

**b.addBooking();**

**bookings.push\_back(b);**

**}**

**void WriteDatatoFile()**

**{**

**for (auto& booking : bookings) {**

**booking.WriteDatatoFile();**

**}**

**}**

**void viewBookingsFromFile()**

**{**

**if (bookings.empty())**

**{**

**cout << "No record found!" << endl;**

**}**

**ifstream ReadFile("C:\\Users\\Razi\\Desktop\\cpp\\Booking.txt");**

**if (!ReadFile.is\_open())**

**{**

**cout << "FILE IS NOT OPENED" << endl;**

**return;**

**}**

**int bookingId;**

**int bookingCustomId;**

**date bookingDate;**

**string bookingTitle, bookingDescription;**

**while (ReadFile >> bookingId >> bookingCustomId >> d.dd >> d.mm >> d.yyyy >> bookingTitle >> bookingDescription) {**

**Booking b; // Create a new booking object**

**b.booking\_id = bookingId;**

**b.bookingcustomer\_id = bookingCustomId;**

**b.booking\_date = bookingDate;**

**b.booking\_title = bookingTitle;**

**b.booking\_description = bookingDescription;**

**bookings.push\_back(b); // Add the booking object to the vector**

**}**

**ReadFile.close();**

**}**

**void SearchBooking()**

**{**

**int searchId;**

**cout << "Enter the Id you Want To Search " << endl;**

**cin >> searchId;**

**bool found = false;**

**if (bookings.empty())**

**{**

**cout << "NO RECORD FOUND" << endl;**

**}**

**for (auto& booking : bookings) {**

**if (searchId == booking.booking\_id)**

**{**

**booking.SearchBooking();**

**found = true;**

**break;**

**}**

**}**

**if (!found)**

**{**

**cout << "invalid id!" << endl;**

**}**

**}**

**};**

**void Booking::BookingMenu()**

**{**

**system("cls");**

**int bookingchoice;**

**BookingManager bookingobj;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tBOOKING MENU" << endl;**

**cout << "\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. ADD BOOKING" << endl;**

**cout << "\n\t\t2. EDIT BOOKING" << endl;**

**cout << "\n\t\t3. SEARCH BOOKING" << endl;**

**cout << "\n\t\t4. VIEW BOOKING " << endl;**

**cout << "\n\t\t5. DELETE BOOKING" << endl;**

**cout << "\n\t\t0. BACK TO MAIN" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> bookingchoice;**

**system("cls");**

**switch (bookingchoice)**

**{**

**case 1:**

**{**

**bookingobj.addBooking();**

**system("pause");**

**system("cls");**

**break;**

**}**

**/\* case 2:**

**{**

**bookingobj.editBooking();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 3:**

**{**

**bookingobj.SearchBooking();**

**system("pause");**

**system("cls");**

**break;**

**}**

**case 4:**

**{**

**bookingobj.viewBookingsFromFile();**

**for (auto& b : bookingobj.bookings) {**

**b.viewBooking();**

**}**

**system("pause");**

**system("cls");**

**break;**

**}**

**/\*case 5:**

**{**

**bookingobj.deleteBooking();**

**system("pause");**

**system("cls");**

**break;**

**}\*/**

**case 0:**

**break;**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (bookingchoice != 0);**

**}**

**void DisplayMainMenu()**

**{**

**int choice;**

**do**

**{**

**cout << "\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t\tMAIN MENU" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\n\t\t1. HOTEL" << endl;**

**cout << "\n\t\t2. EMPLOYEE" << endl;**

**cout << "\n\t\t3. ROOM" << endl;**

**cout << "\n\t\t4. PAYMENT" << endl;**

**cout << "\n\t\t5. CUSTOMER" << endl;**

**cout << "\n\t\t6. BOOKING" << endl;**

**cout << "\n\t\t0. EXIT" << endl;**

**cout << "\n\t\t\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cin >> choice;**

**system("cls");**

**switch (choice)**

**{**

**case 1:**

**{**

**system("color 3");**

**Hotel::HotelMenu();**

**break;**

**}**

**case 2:**

**{**

**system("color 1");**

**Employee::EmployeeMenu();**

**break;**

**}**

**case 3:**

**{**

**system("color 5");**

**Room::RoomMenu();**

**break;**

**}**

**case 4:**

**{**

**system("color 6");**

**Payment::PaymentMenu();**

**break;**

**}**

**case 5:**

**{**

**system("color 9");**

**Customer::CustomerMenu();**

**break;**

**}**

**case 6:**

**{**

**system("color 2");**

**Booking::BookingMenu();**

**break;**

**}**

**case 0:**

**exit(0);**

**default:**

**cout << "\nINVALID CHOICE!" << endl;**

**system("pause");**

**system("cls");**

**break;**

**}**

**} while (choice != 0);**

**getch();**

**return;**

**}**

**int main()**

**{**

**display1();**

**system("pause");**

**display();**

**cout << "Welcome!" << std::endl;**

**int choice1;**

**cout << "1. Register\n2. Login\nChoose an option: ";**

**cin >> choice1;**

**if (choice1 == 1)**

**{**

**registerUser();**

**char choice;**

**cout << "Do you want login for login y/ n" << endl;**

**cin >> choice;**

**if (choice == 'Y' || (choice == 'y'))**

**{**

**system("cls");**

**loginUser();**

**cout << "LOgin Successful!" << endl;**

**system("pause");**

**system("cls");**

**DisplayMainMenu();**

**}**

**}**

**else if (choice1 == 2)**

**{**

**if (loginUser())**

**{**

**cout << "Login successful!" << std::endl;**

**system("pause");**

**system("cls");**

**DisplayMainMenu();**

**}**

**else**

**{**

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

**system("pause");**

**system("cls");**

**}**

**}**

**else**

**{**

**cout << "Invalid choice. Exiting..." << endl;**

**}**

**}**