```
// Team Members : Roba Al-Shahrani - Asiya Saquib - Alin Al-Qahtani - Nazilah Al-Azmi - Fatimah Al-Marri- Alhanoof Aldossary

#include <iostream>
#include <string>
#include <fstream>
#include <cstdlib>
#include <ctime>
using namespace std;
const int SIZE = 10; //global constant
int customer_Count = 0; //actul size of array
string filename="preserve.txt"; //creat file name
//to count the records in statistic report
int addCount = 0;
int updateCount = 0;
int deleteCount = 0;
int sortCount = 0;
int searchCount = 0;
int displayCount = 0;

time_t lastTime;
//time_t is a data type that represents time in numbers
//lastTime is a declared variable

//Creat structure
struct Hotel
{
        string First_name;
    int Duration_of_stay;
    char Room_type;
    int Room_number;
    string Customer_mobile_number;

};

// Function prototype
//if we dont declare it here it wont work in the other functions
void savedFile(Hotel cus[]);

// Read the data from the file
void readData(Hotel cus[]){
        ifstream read ;
        read.open("preserve.txt");

        if(read.is_open()==true){
                int i=0;
                while(!read.eof()){

                read >> cus[i].First_name ;
                read>>cus[i].Duration_of_stay ;
                read>>cus[i].Room_type ;
        read>> cus[i].Room_number ;
            read>> cus[i].Customer_mobile_number ;
            i++;

            }
                    customer_Count=i-1;;
                read.close();
        }
        else
                cout<<"Unable to open the file "<<endl ;

        // Print the array read data
        if (customer_Count > 0) {
                cout<<"Here's the file content: "<<endl;
        for (int i = 0; i < customer_Count; i++) {
            cout << "First Name: " << cus[i].First_name << endl;
            cout << "Duration of stay: " << cus[i].Duration_of_stay << " days" << endl;
            cout << "Room type: " << cus[i].Room_type << endl;
            cout << "Room number: " << cus[i].Room_number << endl;
            cout << "Mobile number: +966 " << cus[i].Customer_mobile_number << endl << endl;

        }
    }
}
//Function to show the menu
int menu()
{
    int choice;
    cout << "\n\t           ### Hotel System ###      \n\n";
    cout << "\t-----------------------------------------------------"
         << "\n\tEnter   1:\tTo Add New Customer Details"
         << "\n\t-----------------------------------------------------"
         << "\n\tEnter   2:\tTo Search a customer"
         << "\n\t-----------------------------------------------------"
         << "\n\tEnter   3:\tTo Update Existing Customer Details"
         << "\n\t-----------------------------------------------------"
```

```
         << "\n\tEnter   4:\tTo Delete a Customer Details "
         << "\n\t-----------------------------------------------------"
         << "\n\tEnter   5:\tTo Sort Customers names alphabetically "
         << "\n\t-----------------------------------------------------"
         << "\n\tEnter   6:\tTo display customers list "
         << "\n\t-----------------------------------------------------"
         << "\n\tEnter   7:\tTo Exit"
         << "\n\t-----------------------------------------------------";
    cout << "\n\t\tPlease Enter Your Choice: ";
    cin >> choice;
    return choice;
}
// Overloading functions to check the name and the room type Duration of stay
void check(Hotel cus[], string name,int customerCount  ){
        bool hasDigit;
   do{
            hasDigit=false;
          int len =name.length();

        for (int i = 0; i < len; i++){
         if (isdigit(name[i])) {
         hasDigit=true;
          }
          }

          if (hasDigit){
                  cout << "Invalid !. Please enter the first name again without any digits: " << endl;
            name=" ";
        cout << "Customer First Name: ";
        cin >>name;
          }  else{
                  cus[customerCount].First_name=name;
                  return;
          }

}while(hasDigit==true );
}

void check(Hotel cus[],char roomtype,int customerCount ){
        while(true){
        if(roomtype=='S' || roomtype=='D') {
        cus[customerCount].Room_type=roomtype;
        return;
        }
        else {
        cout<<"Invild entery ! Please enter either S for single bed room or D for double bed room: "<<endl;
           cin>>roomtype;}

}
}
void check( Hotel cus[] ,int Duration_of_stay, int customerCount){

    bool stay=false;
        do {
    if (Duration_of_stay >= 1 && Duration_of_stay <= 30) {
        stay = true;
        break;
    } else {
        cout << "Invalid input, please enter again a number between 1-30: ";
        cin >> Duration_of_stay;
    }
        } while (!stay);
        cus[customerCount].Duration_of_stay = Duration_of_stay;}

////Overloading functions named check1 to check the mobile number for the function add
//Function 1:
void check1( Hotel cus[] , string mobile_number, int customerCount){

    bool correct_number = false;
    do {
        int len = mobile_number.length();

        if (len == 9 && mobile_number[0] == '5') {
            correct_number = true;
            for (int i = 0; i < len; i++) {
                if (!isdigit(mobile_number[i])) {
                    correct_number = false;
                    break;
                }
            }
        }
        if (!correct_number) {
            cout << "Something wrong! Please enter the mobile number again ,Customer mobile number: +966 " ;
             cin >> mobile_number;
        } else {
            cus[customerCount].Customer_mobile_number = mobile_number;
        }
```

```
    } while (!correct_number);
        }
        //Function 2:
                string check1( string mobile_number){

    bool correct_number = false;
    do {
        int len = mobile_number.length();

        if (len == 9 && mobile_number[0] == '5') {
            correct_number = true;
            for (int i = 0; i < len; i++) {
                if (!isdigit(mobile_number[i])) {
                    correct_number = false;
                    break;
                }
            }
        }
        if (!correct_number) {
            cout << "Something wrong! Please enter the mobile number again ,Customer mobile number: +966 " ;
             cin >> mobile_number;
        } else {
            return mobile_number ;
        }
    } while (!correct_number);
        }

//random generator for room number
int rand_generator(){
        srand(time(0));//for the random generator
        return (100+ rand() % (900 - 100 + 1));
}
//Function to add customer details
void add_customer(Hotel cus[],int &customerCount)
{
 string mobile_num;
   bool exists;

   do {
    cout << "Enter Customer mobile number: +966 ";
    cin >> mobile_num;
    mobile_num=check1(mobile_num); //return the the mobile number by writing it correctly
    exists = false;

    // Check if the mobile number already exists
    for (int i = 0; i < customerCount; ++i)
    {
        if (cus[i].Customer_mobile_number == mobile_num)
        {
            cout << "This mobile number is already registered. "<<endl;
            cout<<"Would you like to enter another number? (Yes/No): ";
            do {

                        string choice2;
            cin >> choice2;
            if (choice2 == "Yes" || choice2 == "yes")
            {
                exists = true;
                break; // Break the loop
            }
            else if (choice2 == "No" || choice2 == "no")
            {
                return;
            }
            else
            {
                cout << "Invalid choice. Please enter either 'Yes' or 'No'.";
            }}while (true);
        }
    }
} while (exists);

           //Enter the first name
        string name;
        cout << "Customer First Name: ";
     cin >>name;
     check(cus,name, customerCount);

     // Enter Duration of stay of the customer
    int Duration_of_stay;
    cout << "Duration of stay of the customer (number of days 1-30): ";
    cin >> Duration_of_stay;
    check(cus,Duration_of_stay,customerCount);

        // Enter Room type
    char Roomtype;
    cout << "Customer room type, enter either S for single bed room or D for double bed room: ";
```

```
    cin >> Roomtype;
    check(cus,Roomtype,customerCount);

    // Generate a random Room number
    int random_number = rand_generator();
    cus[customerCount].Room_number = random_number;
    cout << "Customer room number: " << cus[customerCount].Room_number << endl;

    // Enter mobile number
    string mobile_number;
    cout << "Confirm the customer's mobile number: +966 ";
    cin >> mobile_number;
    check1(cus,mobile_number,customerCount);

   cout << "Customer information added successfully\n";
    addCount++;
    customerCount++;
    savedFile(cus);

}

// Function to search By mobile_number
int search_By_mobile_number(Hotel cus[],int customerCount, string mobile_number) {


        // Check the validity of the mobile number
    mobile_number=check1(mobile_number);


    for (int i = 0; i < customerCount; ++i) {
        if (cus[i].Customer_mobile_number == mobile_number) {
                searchCount++;

            return i;
        }
    }
    return -1;



}

//Function to Update existing customer details
void Update_customer(Hotel cus[], int index, int &customerCount) {
    // Use the index to update the customer details
    cout << "Customer First new Name: ";
    cin >> cus[index].First_name;
    check(cus, cus[index].First_name, index);
    cout << "Duration of new stay of the customer (number of days 1-30): ";
    cin >> cus[index].Duration_of_stay;
    check(cus, cus[index].Duration_of_stay, index);
    cout << "Customer room new type, enter either S for single bed room or D for double bed room: ";
    cin >> cus[index].Room_type;
    check(cus, cus[index].Room_type, index);
    int random_number = rand_generator();
    cus[index].Room_number = random_number;
    cout << "Customer new room number: " << cus[index].Room_number << endl;

    cout << "Customer information updated\n";
    updateCount++;
    savedFile(cus);
}

//Function to Delete a Customer Details
void delete_customer(Hotel cus[],int &customerCount, string mobile_number)
{
    cout << "Enter the mobile number of the customer you want to delete: +966 ";
    cin >> mobile_number;
    mobile_number=check1(mobile_number);

    bool found = false;
    for (int i = 0; i < customerCount; i++)
    {
        if (cus[i].Customer_mobile_number == mobile_number)
        {
            for (int j = i; j < customerCount - 1; j++)
            {
                cus[j] = cus[j + 1];
            }
            customerCount--;
            found = true;
            cout << "Customer with mobile number " << mobile_number << " has been deleted successfully.\n";
            deleteCount++;
            savedFile(cus);
            break;
        }
    }
```

```
    if (!found){
    cout<<"Customer not found."<<endl;

   do {
        string choice1;
        cout << "Do you want to try again? (Yes/No): ";
        cin >> choice1;
        if (choice1 == "Yes" || choice1 == "yes") {
            delete_customer(cus, customerCount, mobile_number);
            return; // Exit the function after retrying
        } else if (choice1 == "No" || choice1 == "no") {
            return; // Exit the function if user chooses not to retry
        } else {
            cout << "Invalid entry! Please enter either 'Yes' or 'No'.\n";
        }
    } while (true);}
}

//Function to Sort customers alphabetically
void sort_customers_alphabetically(Hotel cus[],int customerCount, string mobile_number)
{

    for (int i = 0; i < customerCount - 1; i++)
    {
        for (int j = i + 1; j < customerCount; j++)
        {
            if (cus[j].First_name < cus[i].First_name)
            {
                Hotel temp = cus[i];
                cus[i] = cus[j];
                cus[j] = temp;
            }
        }
    }
    cout << "Customers sorted alphabetically.\n";
    sortCount++;
    savedFile(cus);
}
//Function to Display Hotel booking list
void displayData(Hotel cus[],int customerCount) {

        if(customerCount>0){
    for (int i = 0; i < customerCount; ++i) {
        cout << "Customer Details number :" << i + 1 << ":\n";
        cout << "First Name: " << cus[i].First_name << endl;
        cout << "Duratin of stay :" << cus[i].Duration_of_stay<<" days"<<endl;
        cout << "Room Type: " << cus[i].Room_type << endl;
         cout << "Room Number: " << cus[i].Room_number << endl;
        cout << "Mobile Number: +966 " << cus[i].Customer_mobile_number << endl << endl;
        displayCount++;
    }
} else
cout<<"No records has been added yet !";
}

//Function of the statistic report
void statistic_report()
{
    lastTime = time(NULL);//converts last time to currrent time
    ofstream outFile("statistical_report.txt"); // Open a file for writing the report
    if (outFile.is_open())
    {
        outFile << "--- Statistical Report ---" << endl;
        outFile << "Customer Added: " << addCount << " times" << endl;
        outFile << "Customer Updated: " << updateCount << " times" << endl;
        outFile << "Customer Deleted: " << deleteCount << " times" << endl;
        outFile << "Customer Sorted: " << sortCount << " times" << endl;
        outFile << "Customer Searched: " << searchCount << " times" << endl;
        outFile << "Customer Displayed: " << displayCount << " times" << endl;
        outFile << "Last Update Time: " << ctime(&lastTime) << endl;
        outFile << "Thank you." << endl;
        outFile.close(); // Close the file
    }
    else
    {
        cout << "Unable to open file for writing." << endl;
    }
}
    //ctime(&lastTime): ctime will convert the time from time_t to a human readable form in the string lastTime
    //which will present as day mon year, time
    //& is used here because ctime first gets provided with the address of last time which allows it
    //to get access of the value of last time and convert it.

int main()
{
    Hotel cus[SIZE]; // creat array of struct
    cout<<"Welcome in Hotel System , A system that allows you to modify data easily :> \n" ;
```

```
        readData(cus); // to read the data from the file
        string mobile_number;
    int choice;
    do
    {
        choice = menu();
        switch (choice)
        {
        case 1:
            if (customer_Count < SIZE)
            {
                add_customer(cus,customer_Count);
            }
            else
            {
                cout << "\nThe array is full you need to delete first.\n";
            }
            break;

        case 2: {
           if (customer_Count < 1) {
               cout << "Nothing to Search.";
                  break;
               }
                cout << "Enter the mobile number of the customer you want to search: +966 ";
                cin >> mobile_number;
           int index = search_By_mobile_number(cus, customer_Count, mobile_number);
           if (index != -1) {
               cout << "Customer Details:" << endl;
               cout << "First Name: " << cus[index].First_name << endl;
               cout << "Duration of stay: " << cus[index].Duration_of_stay << " days" << endl;
               cout << "Room type: " << cus[index].Room_type << endl;
               cout << "Room number: " << cus[index].Room_number << endl;
               cout << "Mobile number: +966" << cus[index].Customer_mobile_number << endl;

         } else {
               cout << "Customer not found.\n";
                           }
                break;
         }

case 3: {
    if (customer_Count < 1) {
        cout << "Nothing to Update." << endl;
        break;
    }

    string mobile_number;
    cout << "Enter the mobile number of the customer whose information you want to update: +966  ";
    cin >> mobile_number;
     mobile_number=check1(mobile_number);

    bool found = false;
    int index = -1;

    for (int i = 0; i < customer_Count; i++) {
        if (cus[i].Customer_mobile_number == mobile_number) {
            index = i;
            found = true;
            break;
        }
    }

    if (found) {
        // Call Update_customer with the found index
        Update_customer(cus, index, customer_Count);
    } else {
        cout << "Customer not found." << endl;

        do {
            string choice;
            cout << "Do you want to try again? (Yes/No): ";
            cin >> choice;
            if (choice == "Yes" || choice == "yes") {
                // Prompt for mobile number again
                cout << "Enter the mobile number of the customer you want to update: +966 ";
                cin >> mobile_number;
                 mobile_number=check1(mobile_number);
                // Search again with the new mobile number
                found = false;
                for (int i = 0; i < customer_Count; i++) {
                    if (cus[i].Customer_mobile_number == mobile_number) {
                        index = i;
                        found = true;
                        break;
                    }
                }
```

```
                if (found) {
                    // Call Update_customer with the found index
                    Update_customer(cus, index, customer_Count);
                    break;
                }
            } else if (choice == "No" || choice == "no") {
                break;
            } else {
                cout << "Invalid entry! Please enter either 'Yes' or 'No'." << endl;
            }
        } while (true);
    }

    break;
}




        case 4:{
                if(customer_Count<1){
           cout<<"Nothing to Delete.";
            break;
            }
            delete_customer(cus, customer_Count,  mobile_number);
            break;}
        case 5:{
                if(customer_Count<2){
           cout<<"Nothing to Sort.";
            break;
            }
            sort_customers_alphabetically(cus,customer_Count,  mobile_number);
                        break;
                }
            case 6:{
                                if(customer_Count<1){
           cout<<"Nothing to Display.";
            break;}
                    displayData(cus,customer_Count);
                                break;}

            case 7: {
                      cout<<"You have successfully logged out of the Hotel System :) !"<<endl;
                                break;
                           }
                default:
                cout<<"Invalid entry !! , try again";

            }
    } while (choice !=7 );


       int num;
            if (choice==7){
                    statistic_report();
            cout << "Your operations are saved in the statistical report! Thank you. ";}

    return 0;
}

//Function to save data automatically
void savedFile(Hotel cus[])
{
    ofstream outFilePreserve("preserve.txt");
    if (outFilePreserve.is_open())
    {
        for (int i = 0; i <customer_Count; i++)
        {
            outFilePreserve << cus[i].First_name <<endl;
                        outFilePreserve <<cus[i].Duration_of_stay <<endl;
                         outFilePreserve <<cus[i].Room_type <<endl;
                         outFilePreserve<< cus[i].Room_number <<endl;
                    outFilePreserve<< cus[i].Customer_mobile_number << endl;
        }
        outFilePreserve.close();
        cout << "Customer data saved successfully!\n";
    }
    else
    {
        cout << "Unable to save data\n";
    }
}

//END of the CODE.
```