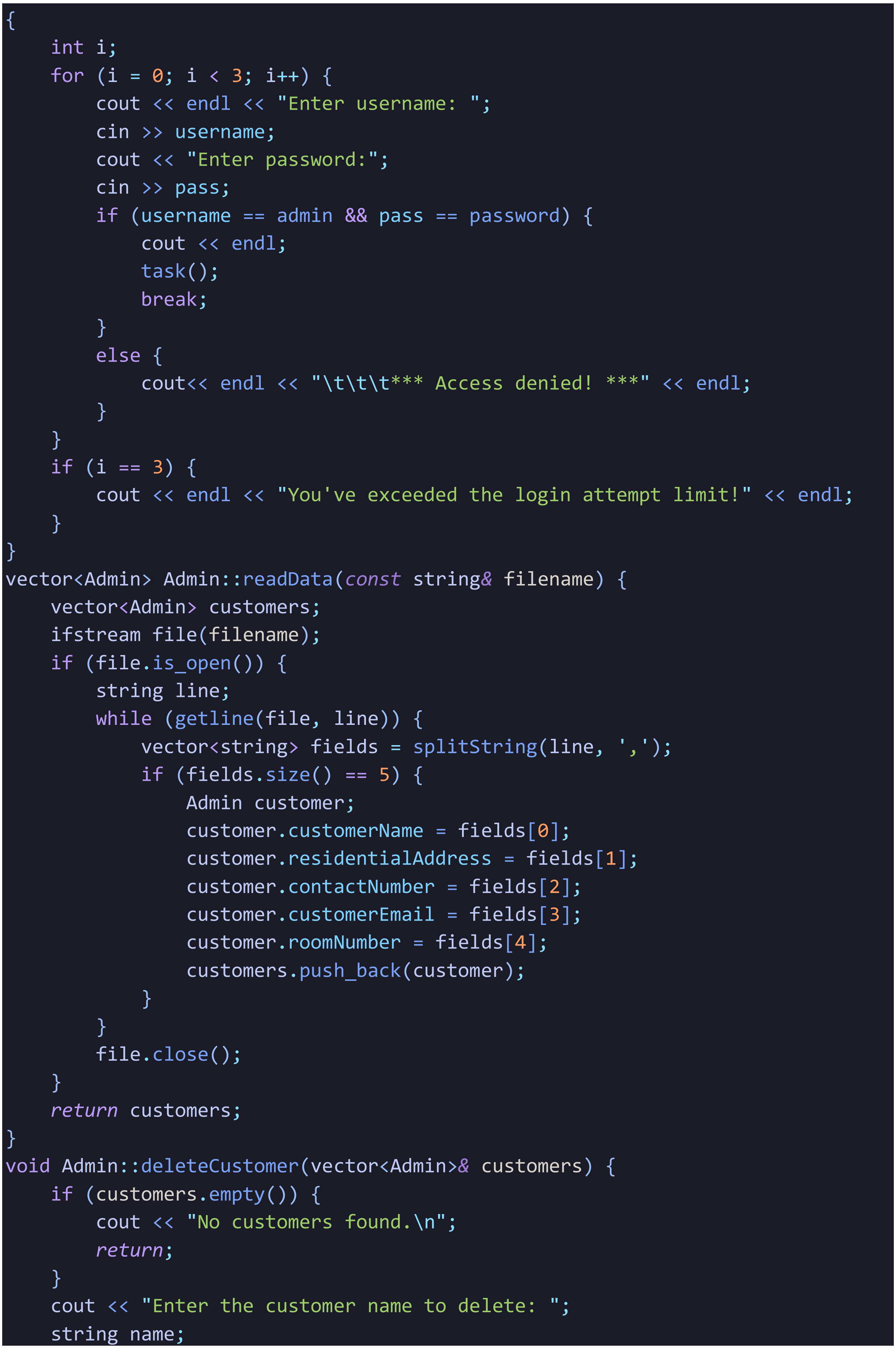
TOPIC: HOTEL MANAGEMENT SYSTEM

CODE:

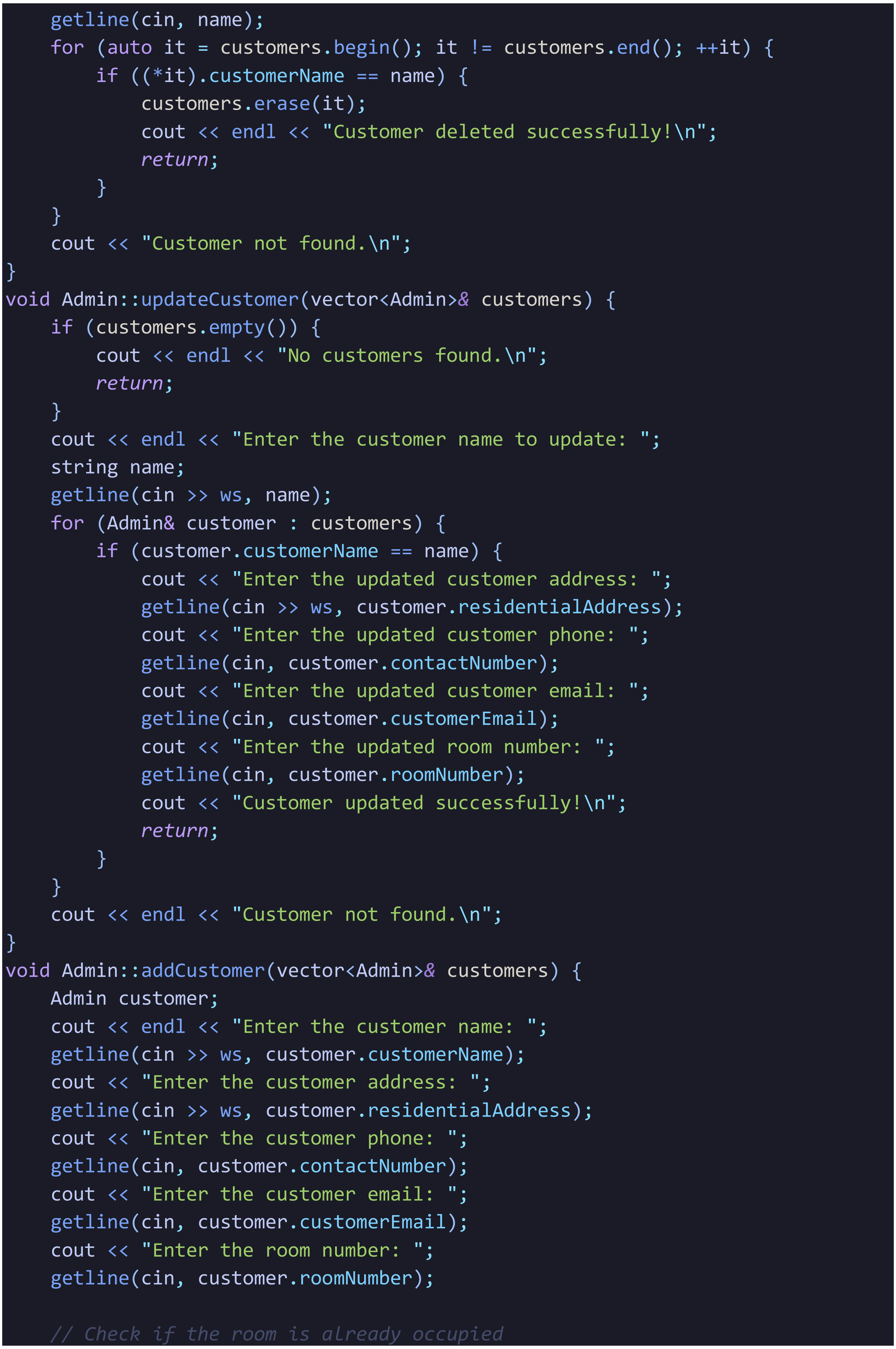
Filename: HMS.cpp

Destination filename (csv file): customers.csv (in the same folder)

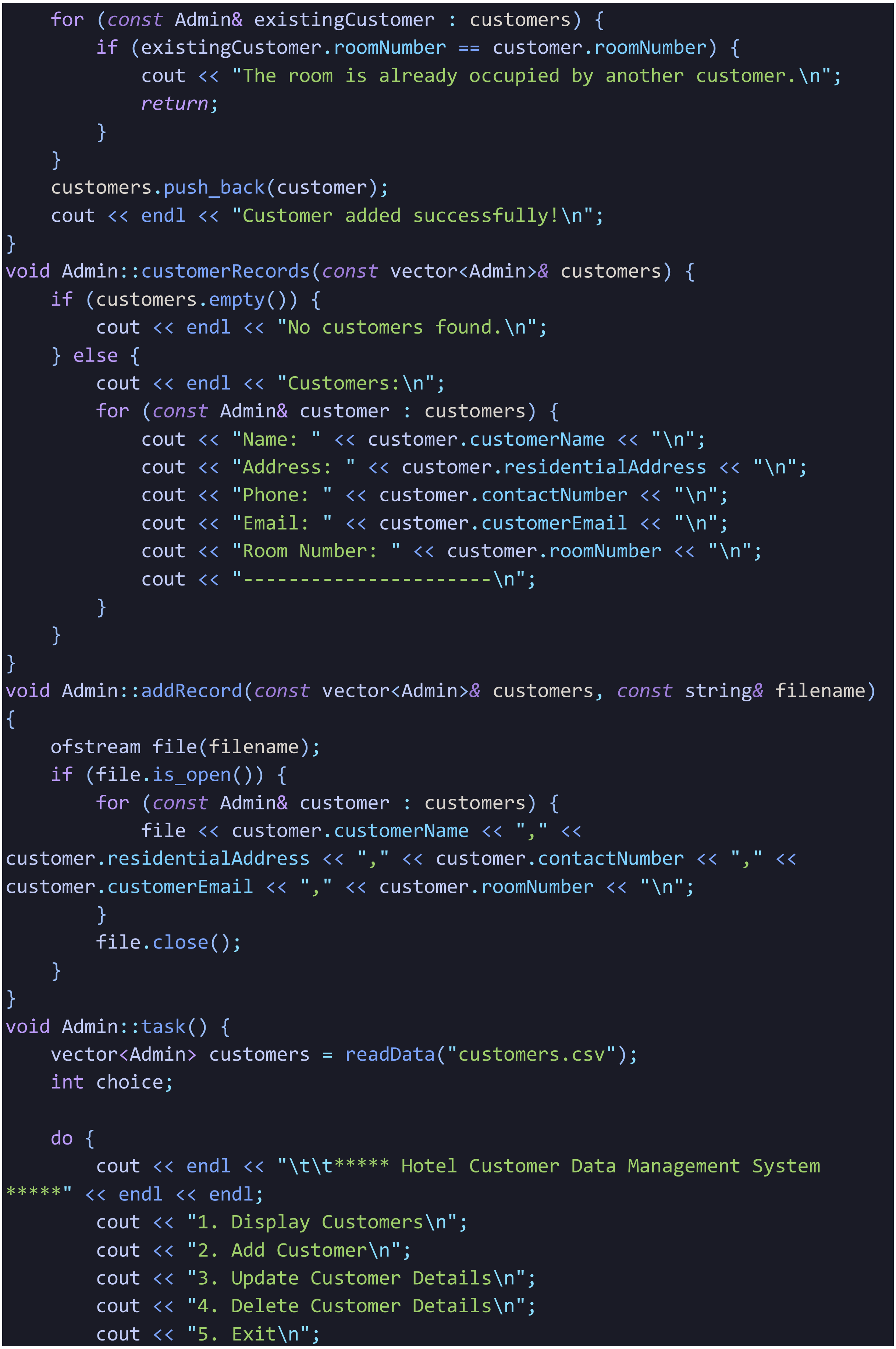
```cpp
#include<iostream>
#include<string>
#include<fstream>       // file stream for file operations
#include<sstream>
#include<vector>
#define admin "admin"
#define password "hotel123"
using namespace std;

vector<string> splitString(const string& str, char delimiter) {
    vector<string> tokens;
    stringstream ss(str);
    string token;
    while (getline(ss, token, delimiter)) {
        tokens.push_back(token);
    }
    return tokens;
}
class Admin {
    protected:
    public:
    string customerName, residentialAddress, contactNumber, customerEmail, roomNumber;
    void task();
    vector<Admin> readData(const string& filename);;
    void deleteCustomer(vector<Admin>& customers);
    void updateCustomer(vector<Admin>& customers);
    void addCustomer(vector<Admin>& customers);
    void customerRecords(const vector<Admin>& customers);
    void addRecord(const vector<Admin>& customers, const string& filename);
};
class Login: public Admin
{
    string username, pass;
    friend class AdminTask;
    public:
    void login();
};
void Login::login()
```



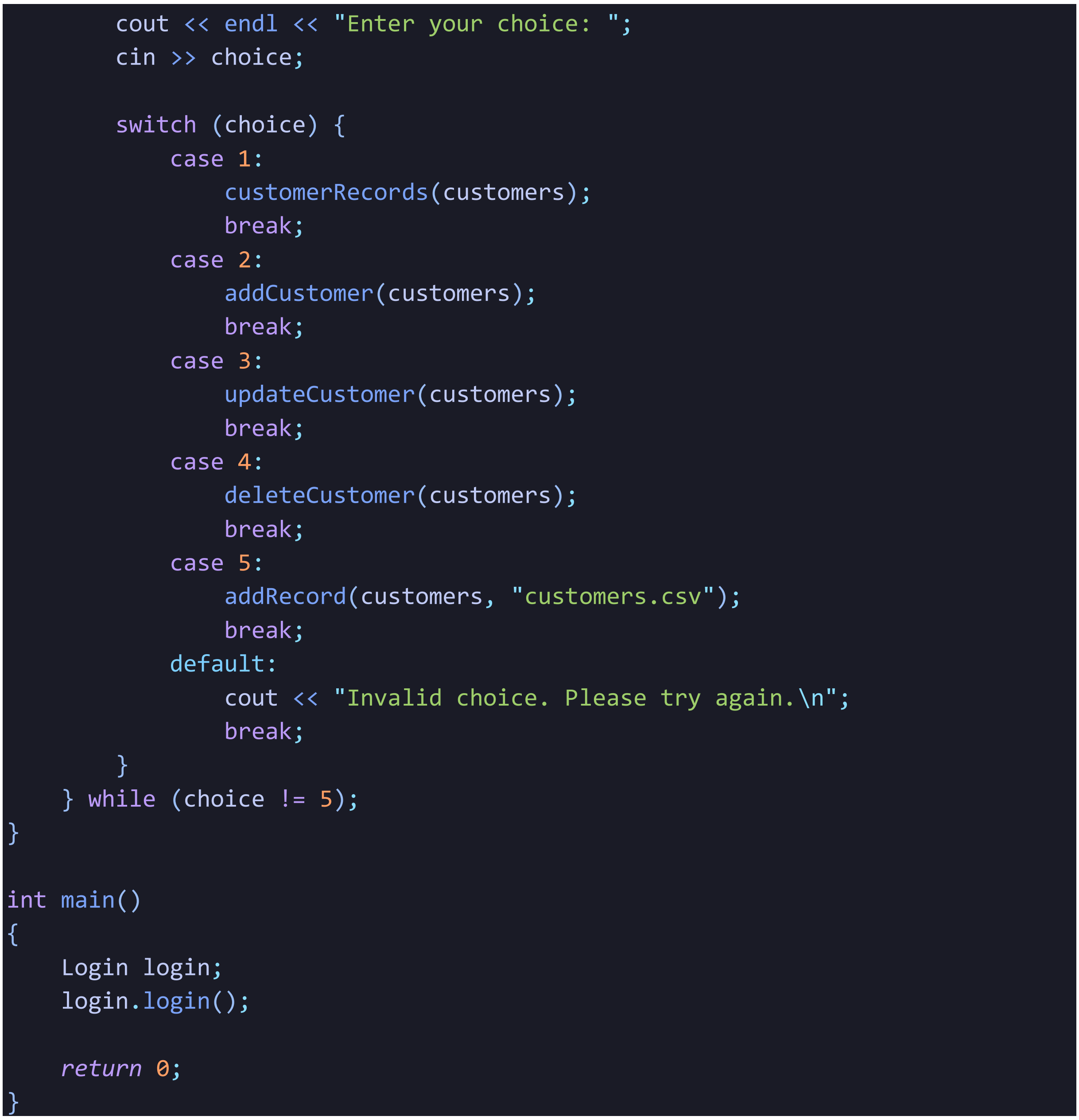
```
{
    int i;
    for (i = 0; i < 3; i++) {
        cout << endl << "Enter username: ";
        cin >> username;
        cout << "Enter password:";
        cin >> pass;
        if (username == admin && pass == password) {
            cout << endl;
            task();
            break;
        }
        else {
            cout<< endl << "\t\t\t*** Access denied! ***" << endl;
        }
    }
    if (i == 3) {
        cout << endl << "You've exceeded the login attempt limit!" << endl;
    }
}
vector<Admin> Admin::readData(const string& filename) {
    vector<Admin> customers;
    ifstream file(filename);
    if (file.is_open()) {
        string line;
        while (getline(file, line)) {
            vector<string> fields = splitString(line, ',');
            if (fields.size() == 5) {
                Admin customer;
                customer.customerName = fields[0];
                customer.residentialAddress = fields[1];
                customer.contactNumber = fields[2];
                customer.customerEmail = fields[3];
                customer.roomNumber = fields[4];
                customers.push_back(customer);
            }
        }
        file.close();
    }
    return customers;
}
void Admin::deleteCustomer(vector<Admin>& customers) {
    if (customers.empty()) {
        cout << "No customers found.\n";
        return;
    }
    cout << "Enter the customer name to delete: ";
    string name;
```



```
    getline(cin, name);
    for (auto it = customers.begin(); it != customers.end(); ++it) {
        if ((*it).customerName == name) {
            customers.erase(it);
            cout << endl << "Customer deleted successfully!\n";
            return;
        }
    }
    cout << "Customer not found.\n";
}
void Admin::updateCustomer(vector<Admin>& customers) {
    if (customers.empty()) {
        cout << endl << "No customers found.\n";
        return;
    }
    cout << endl << "Enter the customer name to update: ";
    string name;
    getline(cin >> ws, name);
    for (Admin& customer : customers) {
        if (customer.customerName == name) {
            cout << "Enter the updated customer address: ";
            getline(cin >> ws, customer.residentialAddress);
            cout << "Enter the updated customer phone: ";
            getline(cin, customer.contactNumber);
            cout << "Enter the updated customer email: ";
            getline(cin, customer.customerEmail);
            cout << "Enter the updated room number: ";
            getline(cin, customer.roomNumber);
            cout << "Customer updated successfully!\n";
            return;
        }
    }
    cout << endl << "Customer not found.\n";
}
void Admin::addCustomer(vector<Admin>& customers) {
    Admin customer;
    cout << endl << "Enter the customer name: ";
    getline(cin >> ws, customer.customerName);
    cout << "Enter the customer address: ";
    getline(cin >> ws, customer.residentialAddress);
    cout << "Enter the customer phone: ";
    getline(cin, customer.contactNumber);
    cout << "Enter the customer email: ";
    getline(cin, customer.customerEmail);
    cout << "Enter the room number: ";
    getline(cin, customer.roomNumber);

    // Check if the room is already occupied
```

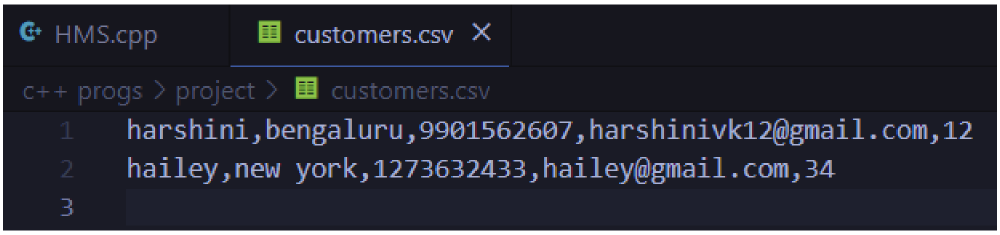


```
    for (const Admin& existingCustomer : customers) {
        if (existingCustomer.roomNumber == customer.roomNumber) {
            cout << "The room is already occupied by another customer.\n";
            return;
        }
    }
    customers.push_back(customer);
    cout << endl << "Customer added successfully!\n";
}
void Admin::customerRecords(const vector<Admin>& customers) {
    if (customers.empty()) {
        cout << endl << "No customers found.\n";
    } else {
        cout << endl << "Customers:\n";
        for (const Admin& customer : customers) {
            cout << "Name: " << customer.customerName << "\n";
            cout << "Address: " << customer.residentialAddress << "\n";
            cout << "Phone: " << customer.contactNumber << "\n";
            cout << "Email: " << customer.customerEmail << "\n";
            cout << "Room Number: " << customer.roomNumber << "\n";
            cout << "----------------------\n";
        }
    }
}
void Admin::addRecord(const vector<Admin>& customers, const string& filename)
{
    ofstream file(filename);
    if (file.is_open()) {
        for (const Admin& customer : customers) {
            file << customer.customerName << "," <<
customer.residentialAddress << "," << customer.contactNumber << "," <<
customer.customerEmail << "," << customer.roomNumber << "\n";
        }
        file.close();
    }
}
void Admin::task() {
    vector<Admin> customers = readData("customers.csv");
    int choice;

    do {
        cout << endl << "\t\t***** Hotel Customer Data Management System
*****" << endl << endl;
        cout << "1. Display Customers\n";
        cout << "2. Add Customer\n";
        cout << "3. Update Customer Details\n";
        cout << "4. Delete Customer Details\n";
        cout << "5. Exit\n";
```

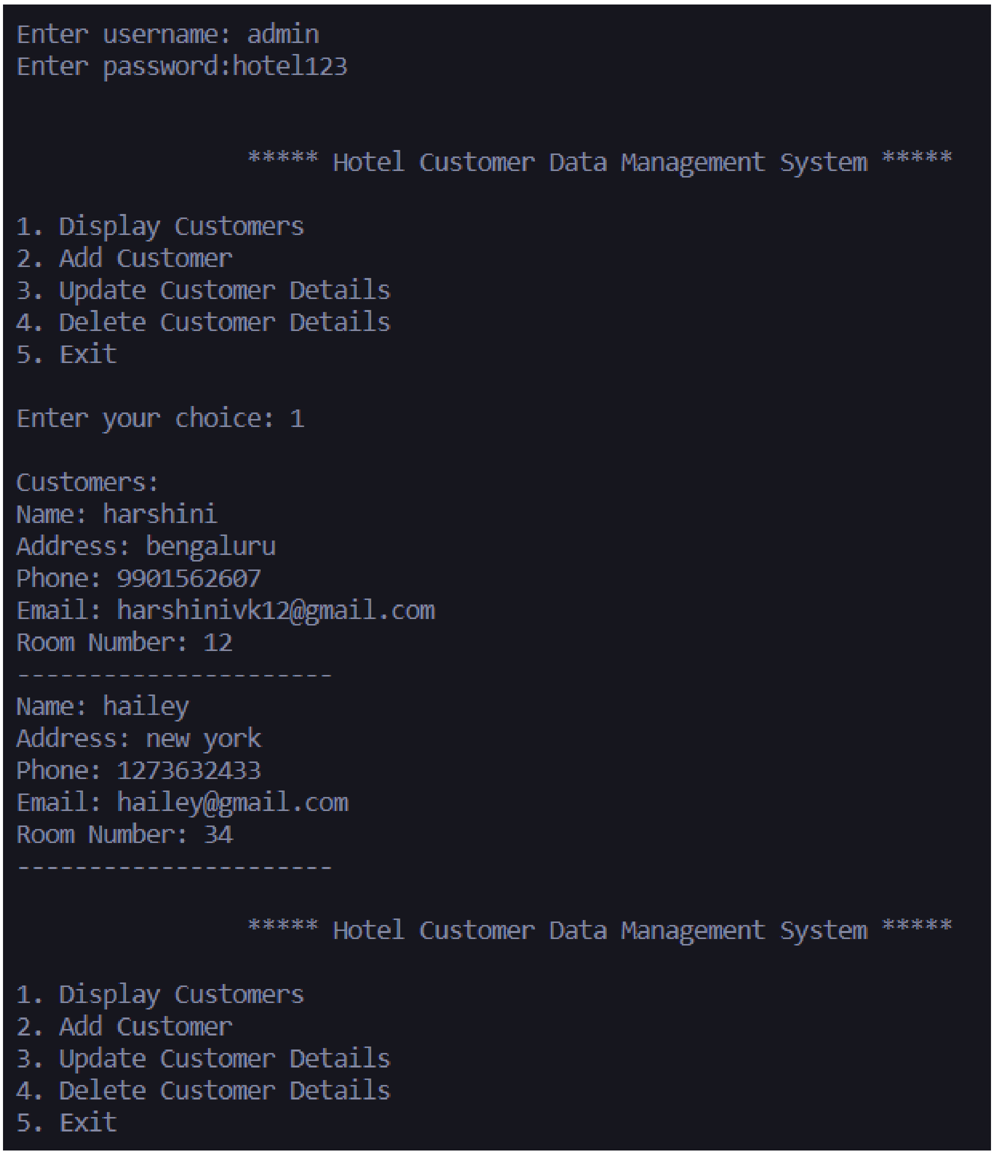


```
        cout << endl << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                customerRecords(customers);
                break;
            case 2:
                addCustomer(customers);
                break;
            case 3:
                updateCustomer(customers);
                break;
            case 4:
                deleteCustomer(customers);
                break;
            case 5:
                addRecord(customers, "customers.csv");
                break;
            default:
                cout << "Invalid choice. Please try again.\n";
                break;
        }
    } while (choice != 5);
}

int main()
{
    Login login;
    login.login();

    return 0;
}
```

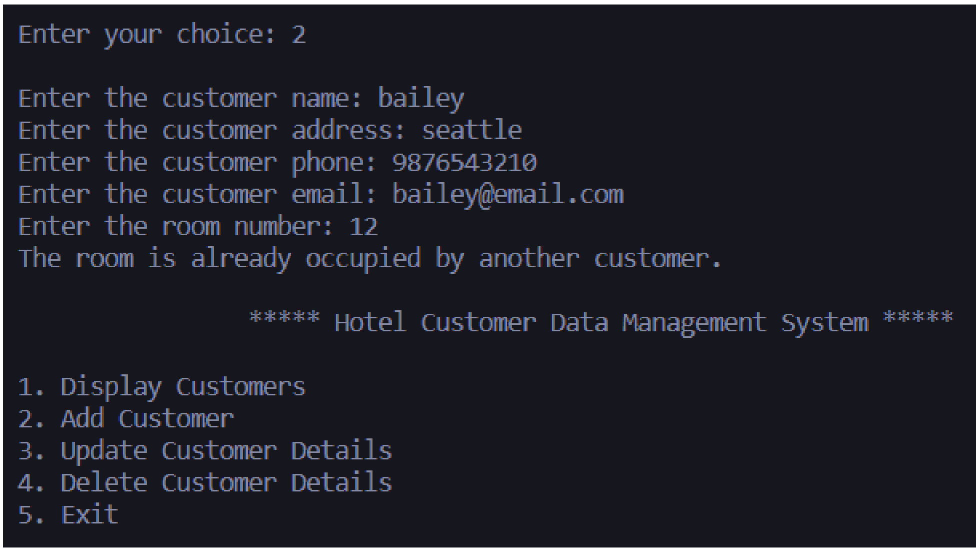
OUTPUT:



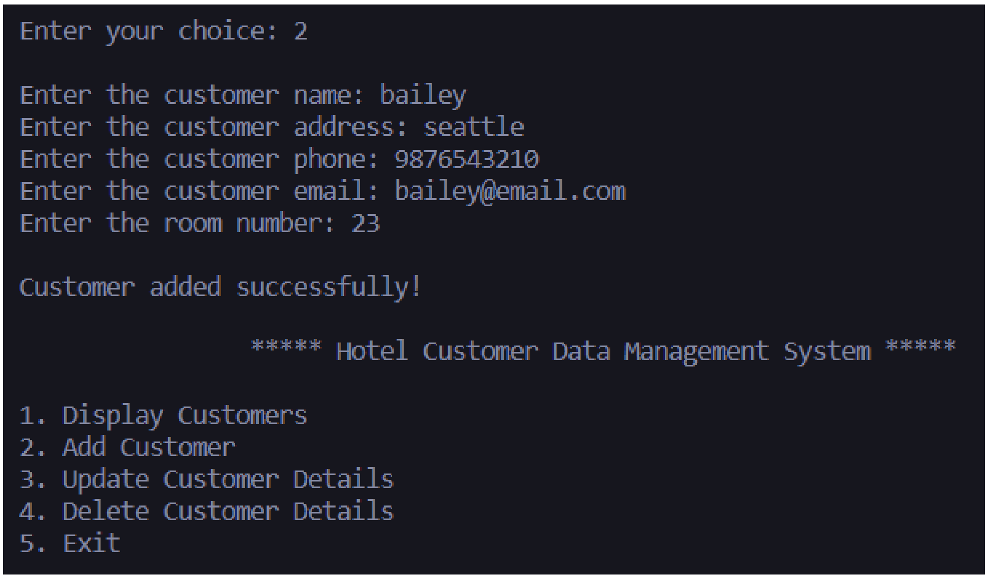
```
harshini,bengaluru,9901562607,harshinivk12@gmail.com,12
hailey,new york,1273632433,hailey@gmail.com,34

```



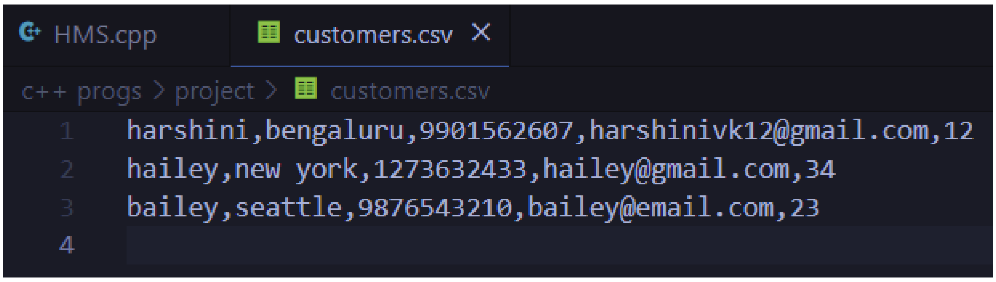
```
Enter username: admin
Enter password:hotel123


                ***** Hotel Customer Data Management System *****

1. Display Customers
2. Add Customer
3. Update Customer Details
4. Delete Customer Details
5. Exit

Enter your choice: 1

Customers:
Name: harshini
Address: bengaluru
Phone: 9901562607
Email: harshinivk12@gmail.com
Room Number: 12
----------------------
Name: hailey
Address: new york
Phone: 1273632433
Email: hailey@gmail.com
Room Number: 34
----------------------

                ***** Hotel Customer Data Management System *****

1. Display Customers
2. Add Customer
3. Update Customer Details
4. Delete Customer Details
5. Exit
```



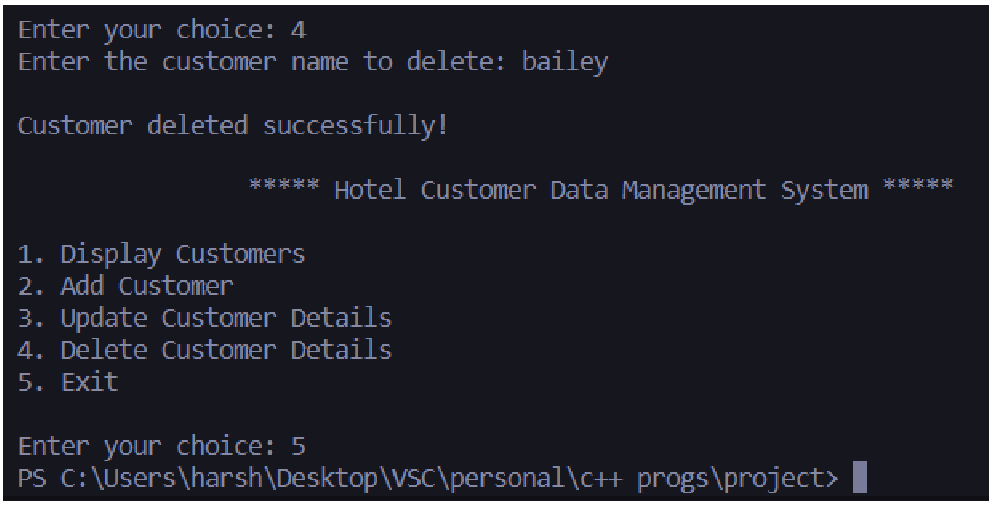
```
Enter your choice: 2

Enter the customer name: bailey
Enter the customer address: seattle
Enter the customer phone: 9876543210
Enter the customer email: bailey@email.com
Enter the room number: 12
The room is already occupied by another customer.

                ***** Hotel Customer Data Management System *****

1. Display Customers
2. Add Customer
3. Update Customer Details
4. Delete Customer Details
5. Exit
```



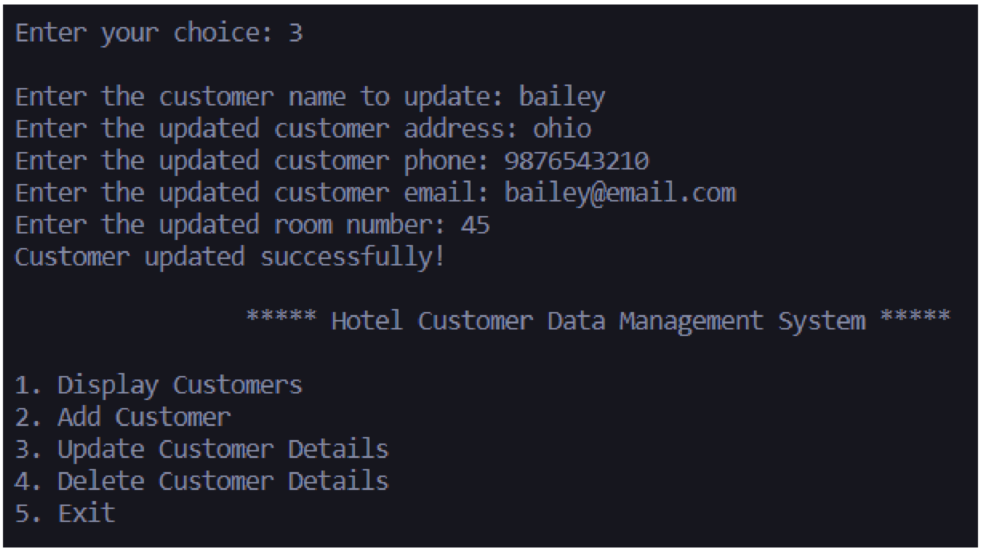
```
Enter your choice: 2

Enter the customer name: bailey
Enter the customer address: seattle
Enter the customer phone: 9876543210
Enter the customer email: bailey@email.com
Enter the room number: 23

Customer added successfully!

                ***** Hotel Customer Data Management System *****

1. Display Customers
2. Add Customer
3. Update Customer Details
4. Delete Customer Details
5. Exit
```



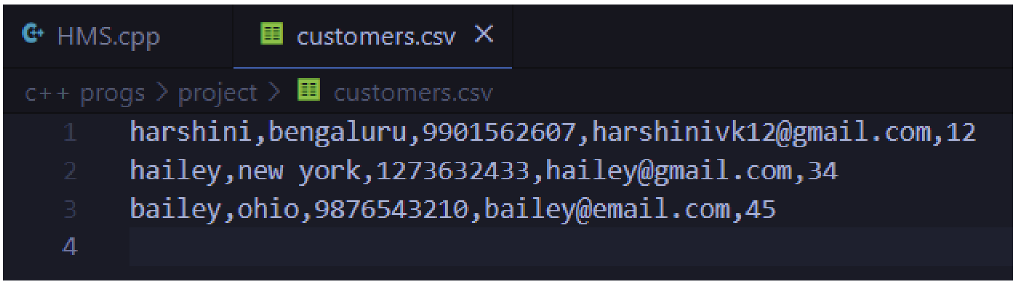
```
harshini,bengaluru,9901562607,harshinivk12@gmail.com,12
hailey,new york,1273632433,hailey@gmail.com,34
bailey,seattle,9876543210,bailey@email.com,23
```



```
Enter your choice: 4
Enter the customer name to delete: bailey

Customer deleted successfully!

                ***** Hotel Customer Data Management System *****

1. Display Customers
2. Add Customer
3. Update Customer Details
4. Delete Customer Details
5. Exit

Enter your choice: 5
PS C:\Users\harsh\Desktop\VSC\personal\c++ progs\project>
```



```
Enter your choice: 3

Enter the customer name to update: bailey
Enter the updated customer address: ohio
Enter the updated customer phone: 9876543210
Enter the updated customer email: bailey@email.com
Enter the updated room number: 45
Customer updated successfully!

                ***** Hotel Customer Data Management System *****

1. Display Customers
2. Add Customer
3. Update Customer Details
4. Delete Customer Details
5. Exit
```



```
harshini,bengaluru,9901562607,harshinivk12@gmail.com,12
hailey,new york,1273632433,hailey@gmail.com,34
bailey,ohio,9876543210,bailey@email.com,45

```