Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 01 REPORT

Azmi Utku Sezgin
131044048

Course Assistant: Fatma Nur Esirci

1 INTRODUCTION

1.1 Problem Definition

Designing a hotel management system where users can book rooms, cancel their booked rooms, receptionists can book rooms, cancel them for guests also check-in and check-out guests.

1.2 System Requirements

There must be 2 .csv files, one for Room lists which holds **RoomNo, isEmpty, isBooked, isCheckedIn and ownerID** respectively. **isEmpty, isBooked, isCheckedIn** are boolean variables but in file you must use 1 for true and 0 for false. Other one is for Signed-up users which holds, **loginInfoNo, name, surname, username, password and guestOrRecep** respectively. **guestOrRecep** can be either 'g' for guest or 'r' for receptionist.

AbstractManagementSystem class has only 1 constructor which gets 2 csv file names as argument that mentioned above.

Person class also has only 1 constructor which gets **name, surname, username, password, guestOrRecep** respectively. **guestOrRecep** can be either 'g' for guest or 'r' for receptionist. Also an important note: In order to login to system, there has to be match in user database.

You must be logged-in(All of the operations) and Authorized (only for check-in and check-out) to perform operations you want.

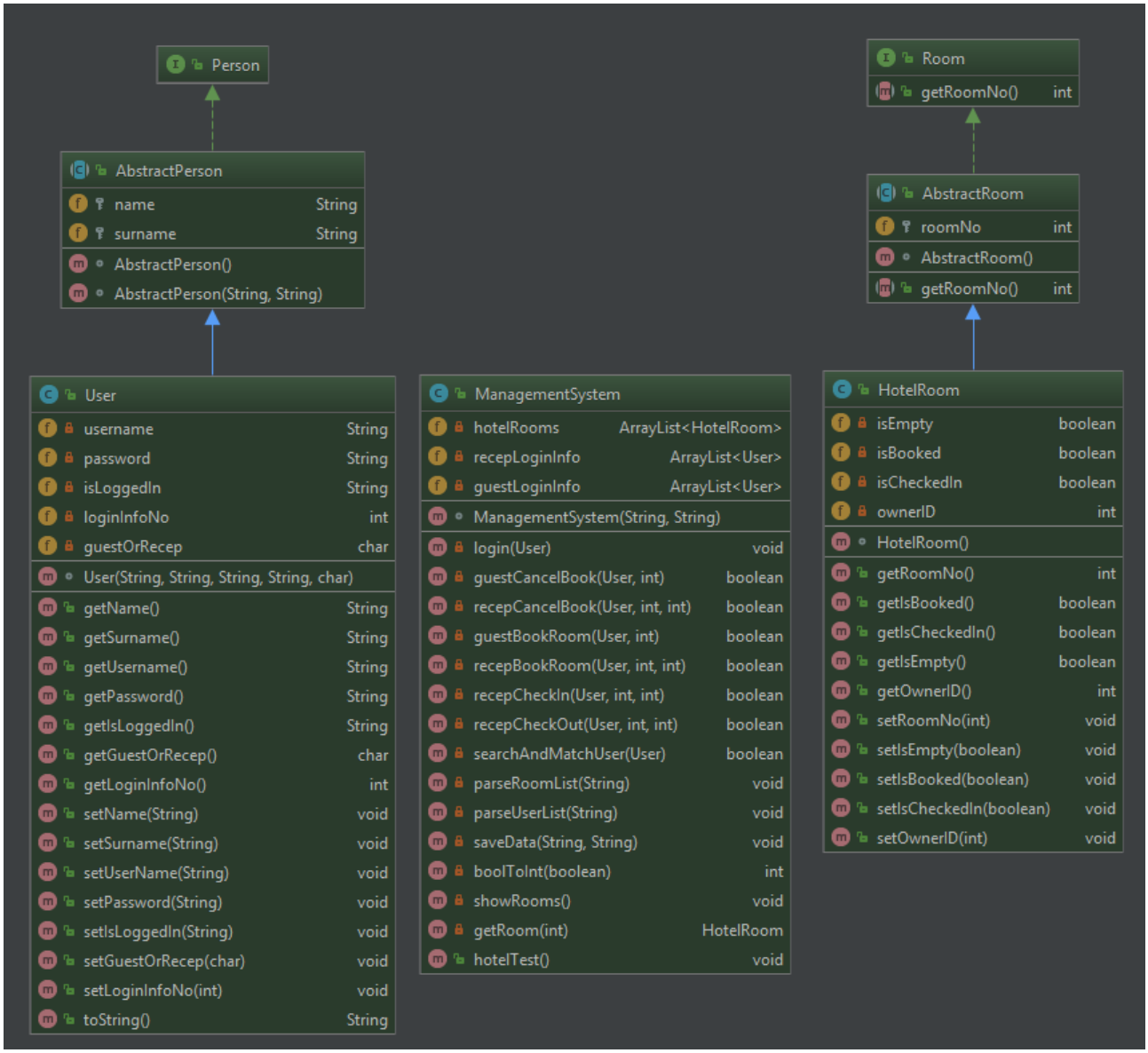
Important

At the start of **hotelTes**t method, you need to create 2 User object using **User(String name, String surname, String username, String password, char guestOrRecep)** constructor. And if they are not in hotel system database they can't log-in therefore they won't be able to use the system operations.

In order to login to server your username and password needs to be correct other informations are not related.

2 METHOD

2.1 Class Diagrams



2.2 Problem Solution Approach

The main idea was having a System with a database that holds Room and User informations and only those users can use the system to Book rooms, Cancel them, Check-in, Check-out. There are 2 csv files. 1 for user database the other one is for room database.

ManagementSystem class has only 1 constructor which gets 2 csv files that are mentioned above. Since these files are crucial to system there's no other way to create an object of this class. Currently, there's only 1 simple test method called **hotelTest()** where 2 User tests the operations of the system. The operations for all the logged-in users are, booking rooms, and canceling them. In addition to that receptionists can check-in and check-out guests. Also they have their own different method for booking rooms. Guest operations start with guest and receptionist methods start with recep. **showRoom()** method works for everyone logged-in or not. **parse** is a prefix for reading and parsing csv files. **getRoom(int)** returns room at given index. **searchAndMatch(User)** method is a helper method for **login(User)** method which searches the User through the database and gives User their **userID** and returns true for **login(User)** method to finish the operation.

User class extends the AbstractPerson abstract class which implements the Person interface. Person interface is used as a tag/marker interface. AbstractPerson has Name and Surname variables and 2 constructors one of which is no parameter constructor that just sets name and surname as "Empty" which is being used at **parseUserList** and 2 parameter constructor that gets Name and Surname. User class has username, password, log-in info, userID and guest or receptionist data fields and all the necessary getters and setters. Since the system is built and works for users only, there has to be a match for the user in system's database.

HotelRoom class extends the **AbstractRoom** abstract class which implements the **Room** interface. Room Interface has only getRoomNo() method. AbstractRoom has roomNo variable, a no-parameter constructor which sets roomNo to 0 as default. Also implements **getRoomNo()** method. HotelRoom class has **isEmpty**, **isBooked**, **isCheckedIn** and **ownerID** variable in addition. Also a no-parameter constructor which basicly creates an empty room. The reason there's only 1 constructor is Room informations are stored in csv files and will be restored from there and no point to create a random room that doesn't exist in the hotel.

3 RESULT

3.1 Test Cases

Case 1-

```
User guest1 = new User( name: "Utk", surname: "sezgin", username: "us",
        password: "us95", guestOrRecep: 'g');
User recep1 = new User( name: "utku", surname: "sezgn", username: "usezgin",
        password: "utk123", guestOrRecep: 'r');
```

Both of the Users are in the system's database.

Case 2-

```
try {
    User guest1 = new User( name: "Utk", surname: "sezgin", username: "unknown",
            password: "us95", guestOrRecep: 'g');
    User recep1 = new User( name: "utku", surname: "sezgn", username: "usezgin",
            password: "utk123", guestOrRecep: 'r');
```

Guest1 doesn't exist in system's database.

3.2 Running Results

Case 1- For the first case, Both guest and receptionist book a room, cancel them and receptionist check-in a guest and check them out. And you can see the changes of related variables before and after screenshow below.

```
Guest signing in.
User successfully logged in.
Receptionist signing in.
User successfully logged in.
RoomNo:1  Available
RoomNo:2  Unavailable
RoomNo:3  Booked
RoomNo:4  Unavailable
RoomNo:5  Available
RoomNo:6  Unavailable
RoomNo:7  Available
RoomNo:8  Available
RoomNo:9  Booked
RoomNo:10  Available

 isBooked variable before the guest booking the room: false
The room successfully booked
isBooked variable after the guest booking the room: true

isBooked variable before the guest cancel the book: true
Book successfully canceled!
isBooked variable after the guest cancel the book: false

isBooked variable before the receptionist booking the room: false
The room successfully booked
isBooked variable after the receptionist booking the room: true

isBooked variable before the receptionist cancel the book: true
Book successfully canceled!
isBooked variable after the receptionist cancel the book: false

The room successfully booked
Before Check-in isEmpty: false isBooked: true isCheckedIn: false
User Checked-in Successfully!
After Check-in isEmpty: false isBooked: false isCheckedIn: true

Before Check-in isEmpty: false isBooked: false isCheckedIn: true
User Checked-out Successfully!
After Check-in isEmpty: true isBooked: false isCheckedIn: false

RoomNo:1  Available
RoomNo:2  Unavailable
RoomNo:3  Booked
RoomNo:4  Unavailable
RoomNo:5  Available
RoomNo:6  Unavailable
RoomNo:7  Available
RoomNo:8  Available
RoomNo:9  Booked
RoomNo:10  Available
Data successfully saved.
```

Case 2-
For the second case, Guest informations doesn't match with anyone in the database so guest cant log-in therefore cant perform any operations whereas Receptionist can still perform their operations.

```
Receptionist signing in.
User successfully logged in.
RoomNo:1  Available
RoomNo:2  Unavailable
RoomNo:3  Booked
RoomNo:4  Unavailable
RoomNo:5  Available
RoomNo:6  Unavailable
RoomNo:7  Available
RoomNo:8  Available
RoomNo:9  Booked
RoomNo:10  Available

 isBooked variable before the guest booking the room: false
You have to log-in to book a room
isBooked variable after the guest booking the room: false

isBooked variable before the guest cancel the book: false
You have to log-in to book a room
isBooked variable after the guest cancel the book: false

isBooked variable before the receptionist booking the room: false
The room successfully booked
isBooked variable after the receptionist booking the room: true

isBooked variable before the receptionist cancel the book: true
Book successfully canceled!
isBooked variable after the receptionist cancel the book: false

You have to log-in to book a room
Before Check-in isEmpty: true isBooked: false isCheckedIn: false
User Checked-in Successfully!
After Check-in isEmpty: true isBooked: false isCheckedIn: true

Before Check-in isEmpty: true isBooked: false isCheckedIn: true
User Checked-out Successfully!
After Check-in isEmpty: true isBooked: false isCheckedIn: false

RoomNo:1  Available
RoomNo:2  Unavailable
RoomNo:3  Booked
RoomNo:4  Unavailable
RoomNo:5  Available
RoomNo:6  Unavailable
RoomNo:7  Available
RoomNo:8  Available
RoomNo:9  Booked
RoomNo:10  Available
Data successfully saved.
```