

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 1 REPORT

**MUHAMMED OKUMUŞ
151044017**

Course Assistant: FATMA NUR ESİRCİ

Introduction

In this project, we designed a simple hotel automation system. It might lack some implementations at some points but idea is still there.

- **Problem Definition**

Naturally our hotel will welcome guests, this requires the guest to be able to make a reservation in the first place. The system has to handle tasks like booking a room, canceling reservations, and logging in as a unique user. And also the hotel staff, receptionists, should be able to check-in and check-out guests, cancel reservations if necessary. To hold all the information system also needs some sort of logbook so guests and receptionist access to current status of the rooms when needed.

- **System Requirements**

ManagementSystem class methods

`public void initUsers(String filename)` = Takes an input file of users as an address, `List<User> users` fill it with Guest and Receptionist objects.

`public void initRooms(String filename)` = Takes an input file of books as an address, `List<User> rooms` fills it with Room objects.

`private void welcomeMessage(User aUser)` = Takes an User objects, prints message according to type of User(Guest or Receptionist).

`public Pair loginUser(String userName, String password)` = Takes userName and password as String, check if user is registered and has entered to correct combination. On successful login returns `Pair(LoggedUser,Int)`, key value is required to pass type of the user to other methods.

On fail returns `Pair(null,0)`.

`public void guestLogged(Guest aGuest)` =

`public void receptionistLogged(Receptionist aReceptionist)` = ...Logged functions takes a User object type parameter. Shows the methods which user can do, handles input flow.

Exits if the requested method number does not exist.

CSVReader class methods

`public List<String[]> csvFileToRowsOfStrings(String filename)` = Takes an input file formatted as a csv, puts each line of file to a `List<String[]>`.

User class methods

`public String getUserID()` = Returns ID of the calling User object.

`protected String getPassword()` = Returns password of the calling User object.

`public void bookRoom(Map<Room, Guest> rooms, Room aRoom, Guest aGuest)` = Takes Map of rooms(aka logbook for hotel), if the target room (`Room aRoom`) is available marks it a reserved, and marks guest (`Guest aGuest`) for that room number.

Guest class methods

`public void setStayingInRoomNumber(int roomNo)` = Marks guest data as the `int roomNo` parameter. Used only if booking is successful or when a reservation is being cancelled.

`public void cancelReservation(List<Room> rooms)` = Sets room that the canceled reservation made to Available again. Needs the logbook of the rooms as the parameter.

Room class methods

`public int getRoomNumber()` = Returns the calling objects room number.

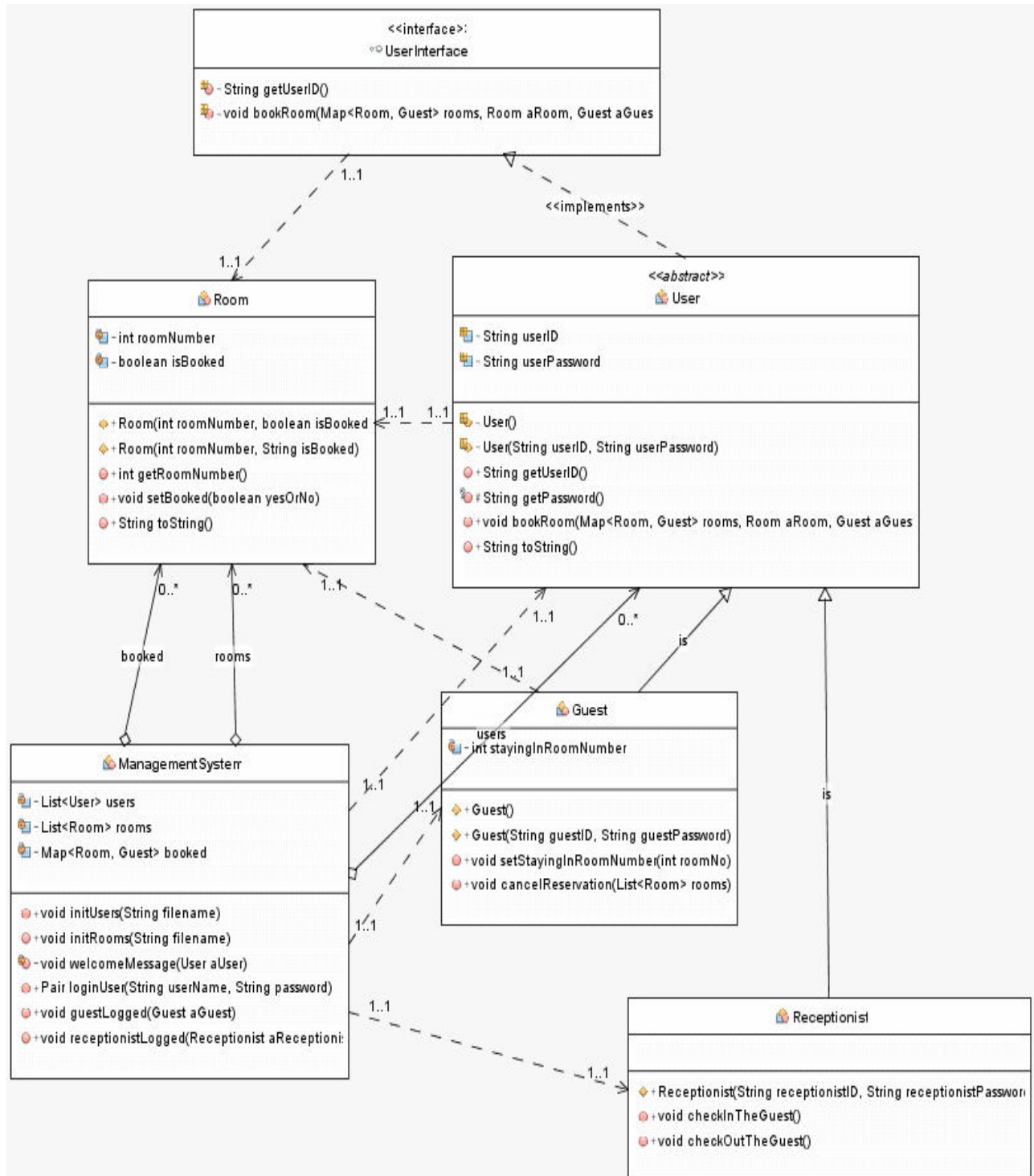
`public void setBooked(boolean yesOrNo)` = Sets the calling objects availability to true or false according to boolean parameter.

Receptionist class methods

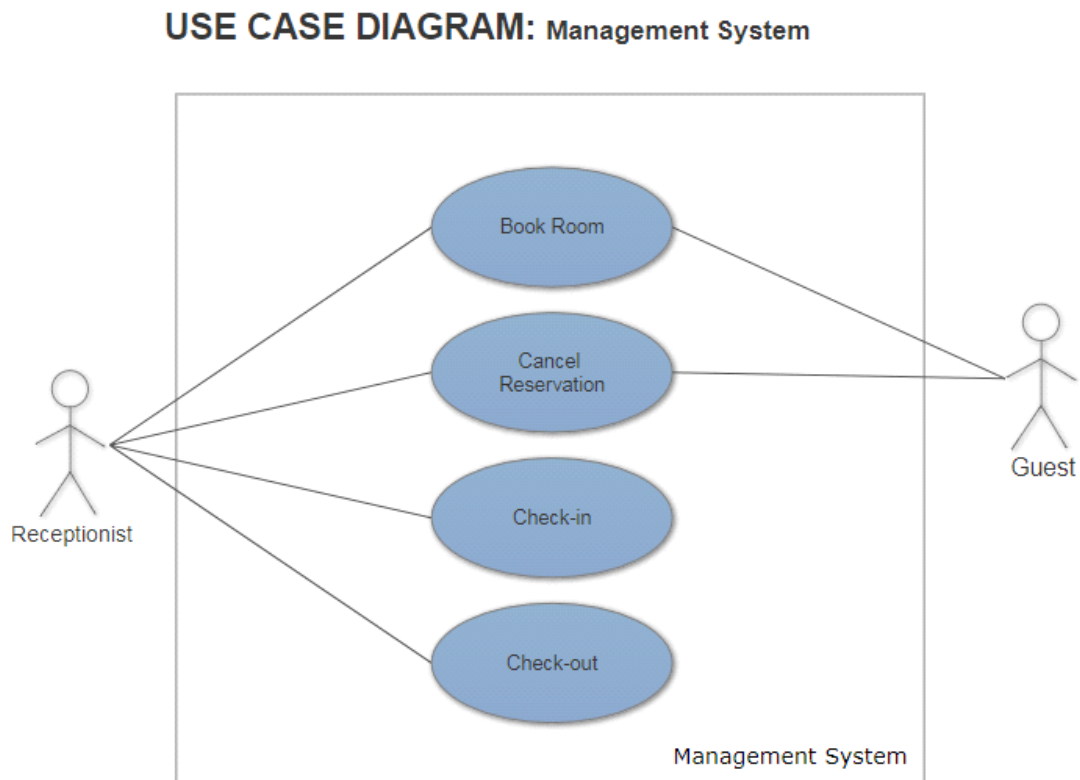
This part is currently incomplete will talk more in Results section if not changed before submission.

- METHOD

- Class Diagrams



- Use Case Diagrams



- Problem Solution Approach

RESULT

Results are not good. I have learned some new things, i had to change some implementations because of flaws in my design, but then it was too late. So i couldn't run meaningful tests. I did run some tests on windows with groovy sdk, i tried to run the same project on VM but then i realised it's a third party library and i don't know if the tester will have that library. Here is those tests:

```
ManagementSystem.java × ManagementSystemTest.groovy × rooms.csv × test.csv × Receptionist.java ×
1 package com.muhammedokumus
2
3 import javafx.util.Pair
4
5 class ManagementSystemTest extends groovy.util.GroovyTestCase {
6
7
8 void testLoginUser() {
9     ManagementSystem hotel = new ManagementSystem();
10    hotel.initUsers( filename: "Files/users.csv");
11    hotel.initRooms( filename: "Files/rooms.csv");
12
13    String testID = "WrongName";
14    String testPass = "Wrong Password";
15
16    Pair<User,Integer> scRes = hotel.loginUser(testID,testPass);
17
18    assertEquals(scRes.getKey(), actual: null)
19
20    String rightID = "user1";
21    String wrongPass = "123";
22    String rightPass = "111";
23
24    scRes = hotel.loginUser(rightID,wrongPass);
25
26    assertEquals(scRes.getKey(), actual: null)
27
28    scRes = hotel.loginUser(rightID,rightPass);
29
30    assertEquals(scRes.getValue(), actual: 1);
31
32 }
33
34
35
ManagementSystemTest > testLoginUser()

1 test passed - 160ms

"C:\Program Files\Java\jdk1.8.0_151\bin\java" ...
Wrong username or password.
Wrong username or password.
Welcome, user1.
1-> Book a room.
2-> Cancel reservation.
3-> Log out.
>
Process finished with exit code 0
```

The screenshot shows an IDE with four tabs: CSVReader.java, CSVReaderTest.groovy, ManagementSystem.java, and ManagementSystem.groovy. The CSVReaderTest.groovy file is active, displaying the following code:

```
1 package com.muhammedokumus
2
3 class CSVReaderTest extends GroovyTestCase {
4     void testCsvFileToRowsOfStrings() {
5         //TEST FOR WRONG FILE NAME,
6         String filename = "Files/teststr.csv"
7
8         CSVReader usersFile = new CSVReader();
9         List<String[]> rawUserData;
10
11         rawUserData = usersFile.csvFileToRowsOfStrings(filename);
12         System.out.println("Expecting rawUserData to be empty on wrong input");
13         assertEquals([], rawUserData)
14
15         //TEST FOR RANDOM FILE
16         filename = "Files/test.csv"
17         rawUserData = usersFile.csvFileToRowsOfStrings(filename);
18         System.out.println(rawUserData);
19         assertEquals( expected: 4, rawUserData.size());
20     }
21 }
```

Below the code editor, the test runner shows "CSVReaderTest" and a green progress bar indicating "1 test passed - 126ms". The console output at the bottom reads:

```
"C:\Program Files\Java\jdk1.8.0_151\bin\java" ...
User data file not found. Please check Files folder under project for users.csv
Expecting rawUserData to be empty on wrong input.
[[Created], [For], [CSVReaderTest], [, , , 222, , 2]]

Process finished with exit code 0
```

I'm not pleased with my code, mistakes were made but i learned some new things and will try to do better next time. I will try to have a better foundation for unit testing in my future projects. Thanks for reading.