

**Gebze Technical University
Computer Engineering**

CSE 222 - 2018 Spring

HOMEWORK 1 REPORT

**ZEYNEB AKDOĞAN
181041038**

Course Assistant:

1 INTRODUCTION

1.1 Problem Definition

Bu çalışmada basit bir otel yönetim sistemi programı yapıldı. Otelde bulunan yemekhane, konferans salonları gibi diğer bölümler göz ardı edilerek, otelin sadece odalardan oluştuğu düşünüldü. Bu sistemi hem otel ziyaretçileri hem de resepsiyonisler kullanabilmektedir. Sistem kullanıcıları sisteme hangi kullanıcının girdiğine göre değişiklik gösteren log in ile girebilmektedirler.

Konuklar, otelden oda rezervasyon yapabilir ya da daha yaptıkları rezervasyonları iptal edebilir. Resepsiyonisler de konuklara oda rezervasyonu yapabilir. Buna ek olarak resepsiyonistler ayrıca müşterilerin odalara check in ve check out yapmalarını da mümkün kılarlar.

1.2 System Requirements

Bu çalışma aşağıda maddeler halinde sıralanan 9 sınıftan oluşmuştur:

- 1.Guest
- 2.Hotel
- 3.IGuest
- 4.IReceptionist
5. Receptionist
6. Reservation
7. Room
8. User
9. HotelFactory

1. Guest: Konuk bilgileri Guest sınıfında yer almaktadır.

```
public class Guest extends User implements IGuest {
```

User tarafından türeyen bu sınıf “**IGuest**” arayüzünü kullanmaktadır. Konuğun adı sisteme girilerek kaydının yapılması amaçlanır. Aslında daha başka guest özellikleri de verilebilirdi ama ben şimdilik ancak bu kadar yapabildim. Konuğun ilgili rezervasyonu User sınıfında hem konuk hem de resepsiyonist için ortak tanımlanan bookARoom içinde

oluşturulur ve buraya atılır. Sisteme girip rezervasyon yapmayan kullanıcılara da henüz rezervasyon yapmadıklarının hatırlatılması oldukça yararlı olacağı düşünülmüştür. Ayrıca sisteme bir rezervasyon alan konunun aynı anda bir tane daha almaması uygun görülmüştür.

```
private String name;  
private Reservation reservation
```

2. Hotel: Senaryoda oteller sadece odaları tutuyor diye ifade edildiğinden otelin sadece odalardan oluştuğu düşünülmüştür. Ayrıca bir çok yardımcı metot bu sınıfta yer almaktadır. Bu satırda otel nesnesi oluşturulurken çağrılan constructor tanımlanmıştır. Buradaki “rooms” otelde tanımlı odalardır. Hotel nesnesi oluşturulurken şu constructor çağırılır:

```
public Hotel(List<Room> rooms) {  
    this.rooms = rooms;  
}
```

```
public Room findAnEmptyRoom() {
```

Yukarıdaki metotla henüz ayrılmamış rezerve edilmemiş ilk odaya geri döner.

```
for (int i=0; i<rooms.size();i++) {  
    Room room = rooms.get(i);  
    if (room.isEmpty()) {  
        return room;  
    }  
}
```

Son olarak otelde tanımlı odaları tek tek kontrol eder ve ilk boş odaya döner.

3. IGuest: Guest objesinin yapabileceği operasyonları tanımlayan ara yüz.

```
public interface IGuest {
```

Rezervasyonu iptal etme operasyonu. Sonuç olarak rezervasyon iptali başarılı mı başarısız mı sonucunu döner.

```
public boolean cancelReservation();
```

4. IReceptionist: Iresepsiyonist sınıfı resepsiyonist objesinin yapabileceği işleri tanımlayan bir arayüzdür. Resepsiyonist checkin ve checkout operasyonları da yapar. Check in için rezervasyon yapmış ama henüz checkin yapılmamış konuk listesini bulur ve erkanda listelenir. Bu listeden seçim yapılarak checkin operasyonu tanımlanır. Aşağıdaki kodla checkin başarılı mı bilgisi döndürülür.

```
public boolean checkin();
```

Aynı şekilde aşağıdaki kod da checkout operasyonu başarılı mı bilgisi ölçülür:

```
public boolean checkout();
```

5. Receptionist: Resepsiyonist sınıfı da Konuk sınıfı gibi User klasından türer ve IReceptionist ara yüzünü uygular. Resepsiyonist sınıfında yazdığımız bu kodla da üst sınıf userin constructoruna parametre geçilir.

```
public class Receptionist extends User implements IReceptionist {
```

```
public Receptionist(Hotel hotel){  
    super(hotel); // üst sınıf Userin constructor'ına parametre geçiyoruz  
    this.hotel = hotel;  
}
```

Resepsiyonist sınıfı, kendisinin üst sınıfı olan User clasının metodunu alırken kodun başında @Override tanımlanır: Aynı şekilde rezervasyon checkout da bu şekilde olur.

```
@Override  
public Room bookARoom(IGuest guest) {  
    Room room = guest.bookARoom();  
    return room;  
}
```

6. Reservation: Konuk rezervasyon bilgileri bu sınıfta tutulur. Bu kod sayesinde rezervasyonun tutulduğu odayı tutar. Ayrıca burada check in yapılıp yapılmadığı belirtilir:

```
private Room room;
```

```
public Reservation(Room room) {  
    this.room = room;  
    this.checkInMade = false;  
}
```

Rezervasyon da burada iptal edilir.

```
public void cancel() {  
    if(room.isEmpty())  
    {  
        throw new UnsupportedOperationException("Oda reserve edilmemis.");  
    }  
    room.cancelRoomReservation();  
    System.out.println(String.format("Reservation for room '%s' cancelled.", room.getRoomNo()));  
    room = null;  
}
```

Oda rezervasyonu için check in işlemi yapılır.

```
public void checkIn() {  
    if(room.isEmpty())  
    {
```

```

        throw new UnsupportedOperationException("Oda reserve edilmemis.");
    }
    if (checkInMade)
    {
        throw new UnsupportedOperationException("zaten check in yapilmis.");
    }
    this.checkInMade = true;
    System.out.println(String.format("Oda %s icin checkin
yapildi.", room.getRoomNo()));
}

```

Rezervasyonun yapıldığı oda bilgisini döner:

```

public Room getRoom() {
    return room;
}

```

checkin yapmış mı yapılmamış mı bilgisi:

```

public boolean isCheckInMade() {
    return checkInMade;
}

```

7. Room: Oteldeki oda bilgilerinin tutulduğu sınıftır. Odayı rezerve eden konuk bilgisi de yer alır.

```

private IGuest guest;

```

Konstruktor:

```

public Room(String theRoomNo, int theBedCount) {
    roomNo = theRoomNo;
    bedCount = theBedCount;
    guest = null;
}

```

Oda boş mu kontrolü yapan metod:

```

public boolean isEmpty() {
    if (guest == null) {
        return true;
    } else {
        return false;
    }
    // return guest == null;
}

```

Oda rezervasyonu yapılırken çağırılan metod:

```

public void cancelRoomReservation() {
    guest = null;
}

```

Oda numarasını döner:

```

public String getRoomNo() {
    return roomNo;
}

```

Odayla ilişkili konuk bilgisini veren metod:

```
public IGuest getGuest() {  
    return guest;  
}
```

Odaya konuk ilişkilendirmek için çağırılan metod:

```
public void setGuest(IGuest guest) {  
    if(getGuest() != null) {  
        throw new UnsupportedOperationException("Oda zaten rezerve edilmiş");  
    }  
    this.guest = guest;  
}
```

8. USER: Bu programda hem konuklar hem de resepsiyonist, her ikisi de oda rezervasyonu yapabilecekleri için onlar için bir üst sınıf tanımlamayı uygun gördüm."User" sınıfı konuk ve resepsiyonistin üst sınıfıdır. İki alt sınıf bu üst sınıftan özelliklerini miras alır. Ayrıca user için IGuest adında bir de arayüz sınıfı oluşturulmuştur.

```
public abstract class User {
```

Uygulamanın tamamında kullanılan hotel nesnesinin referansı:

```
public User(Hotel hotel) {  
    this.hotel = hotel;  
}
```

Aşağıdaki konstrüktör süperdir. Alt sınıflardan super(hotel) şeklinde çağırılır.

```
public User(Hotel hotel) {  
    this.hotel = hotel;  
}
```

9. HotelFactory: Otel nesnesini oluşturan faktory. On odalı bir otel oluşturulur. (Butik otel) ve bütün uygulamada bu otel nesnesi kullanılır.

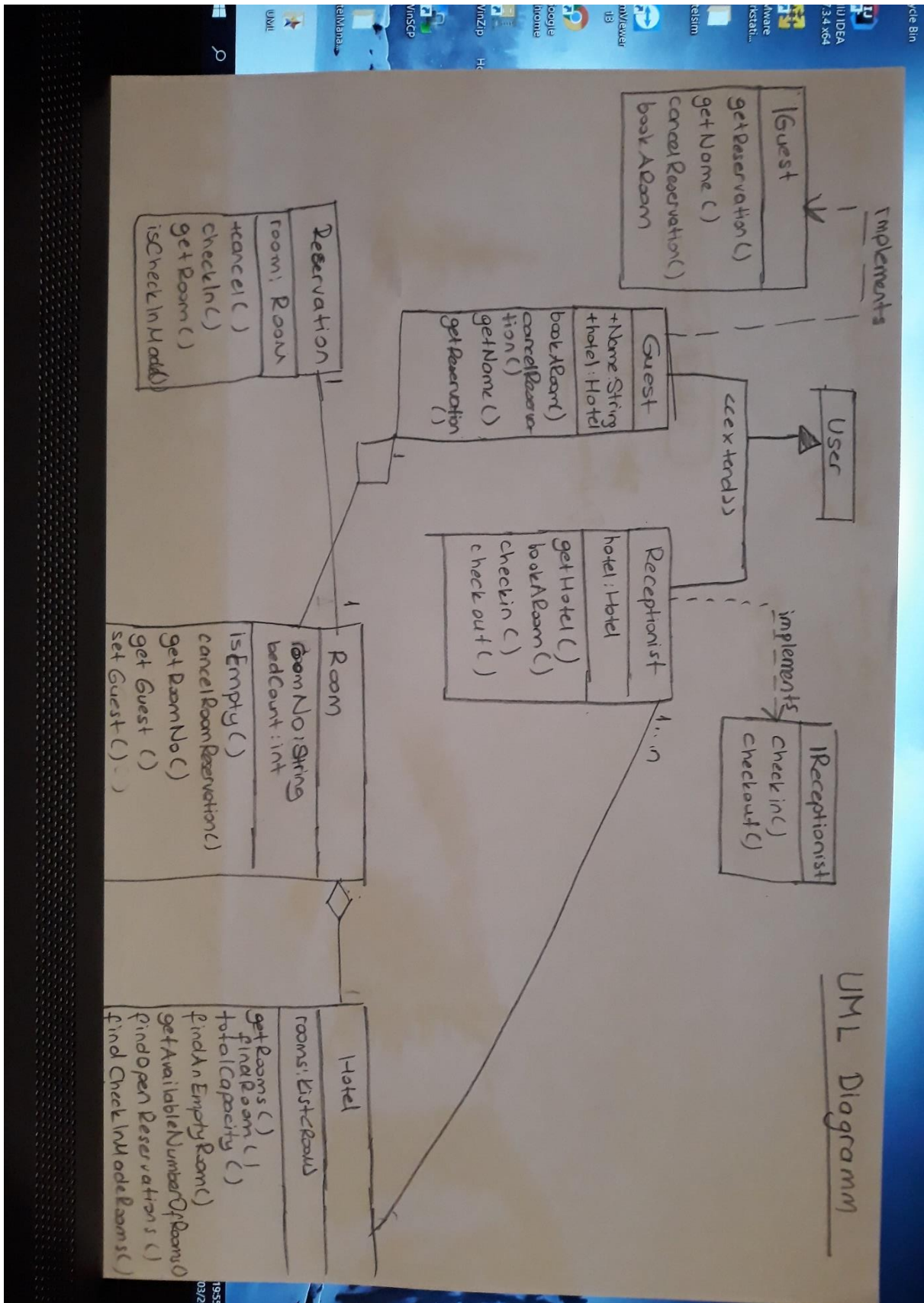
```
public class HotelFactory {  
    public static Hotel createHotelWithTenRoom()  
    {  
        return new Hotel(createRooms());  
    }  
}
```

Burada otel nesnesi için oda listesi oluşturulur:

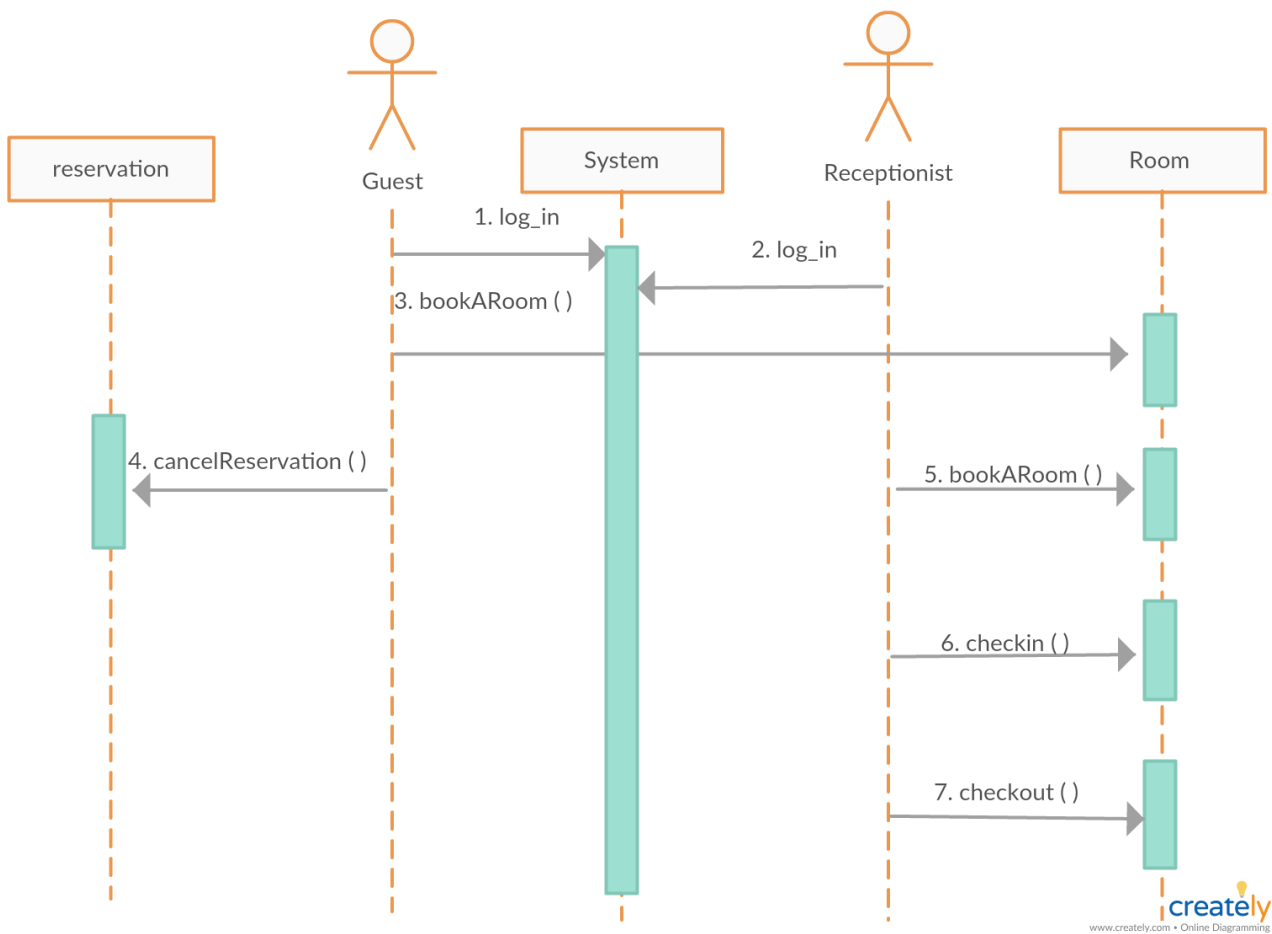
```
private static List<Room> createRooms() {  
    List<Room> roomsList = new ArrayList<>();  
  
    roomsList.add(new Room("101",1));  
    roomsList.add(new Room("102",2));  
    roomsList.add(new Room("103",1));  
    roomsList.add(new Room("104",1));  
    roomsList.add(new Room("105",1));  
    roomsList.add(new Room("106",1));  
    roomsList.add(new Room("107",1));  
    roomsList.add(new Room("108",1));  
    roomsList.add(new Room("109",2));  
    roomsList.add(new Room("110",2));  
  
    return roomsList;  
}
```

2. METHOD

2.1. Class Diagrams



2.2. Sequenz Diagrams



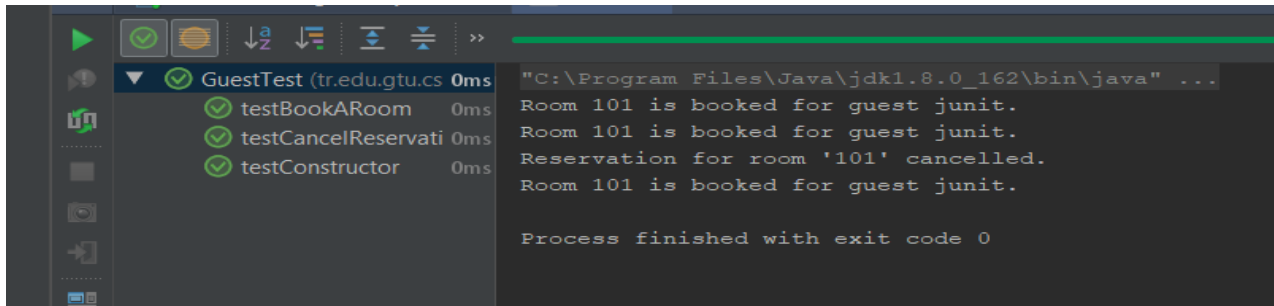
2.3. Problem Solution Approach

Unit testleri yapmak için çok uğraştım fakat main testi yapamadım.

3. RESULT

3.1. Test Cases

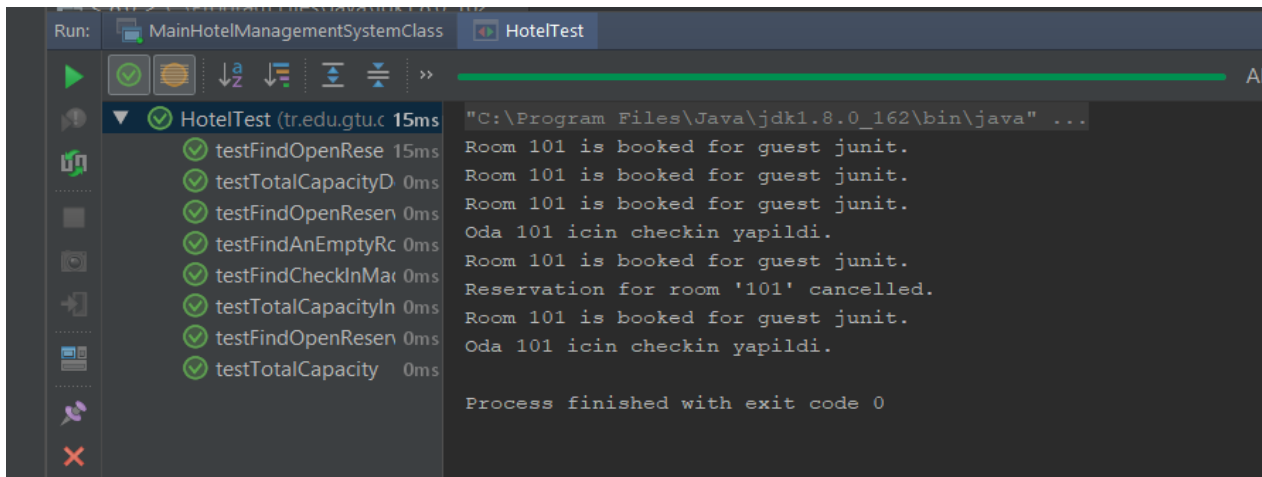
2.1.1. GuestTest:



```
"C:\Program Files\Java\jdk1.8.0_162\bin\java" ...
Room 101 is booked for guest junit.
Room 101 is booked for guest junit.
Reservation for room '101' cancelled.
Room 101 is booked for guest junit.

Process finished with exit code 0
```

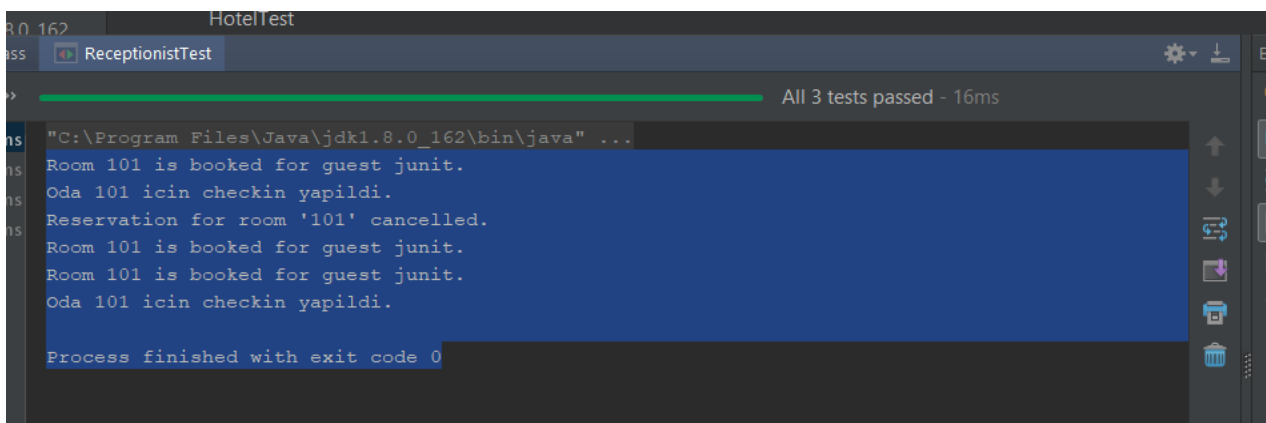
2.1.2. HotelTest:



```
"C:\Program Files\Java\jdk1.8.0_162\bin\java" ...
Room 101 is booked for guest junit.
Room 101 is booked for guest junit.
Room 101 is booked for guest junit.
Oda 101 icin checkin yapildi.
Room 101 is booked for guest junit.
Reservation for room '101' cancelled.
Room 101 is booked for guest junit.
Oda 101 icin checkin yapildi.

Process finished with exit code 0
```

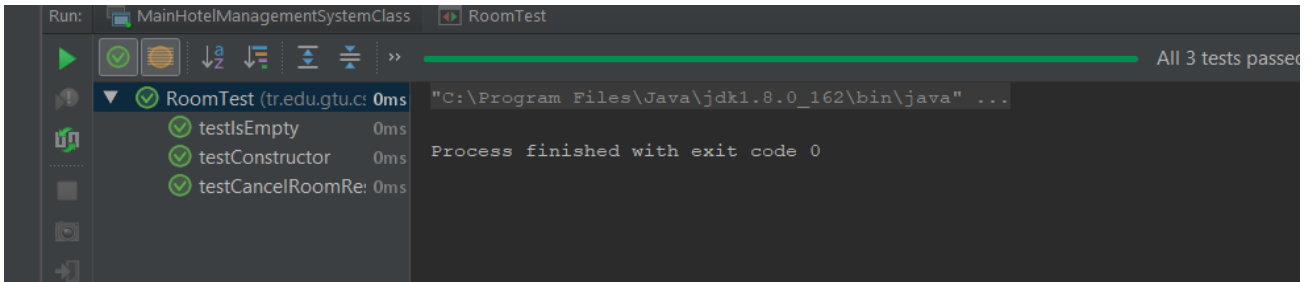
2.1.3. Receptionist Test:



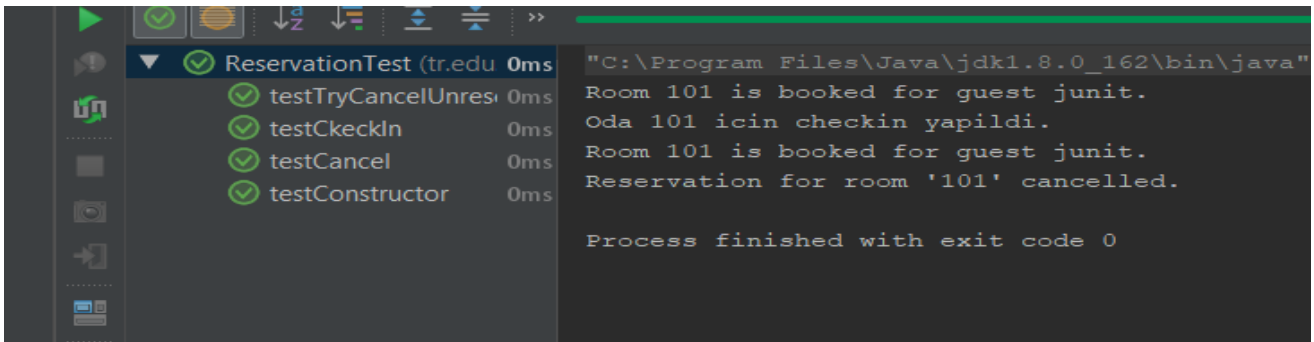
```
"C:\Program Files\Java\jdk1.8.0_162\bin\java" ...
Room 101 is booked for guest junit.
Oda 101 icin checkin yapildi.
Reservation for room '101' cancelled.
Room 101 is booked for guest junit.
Room 101 is booked for guest junit.
Oda 101 icin checkin yapildi.

Process finished with exit code 0
```

2.1.4. RoomTest:

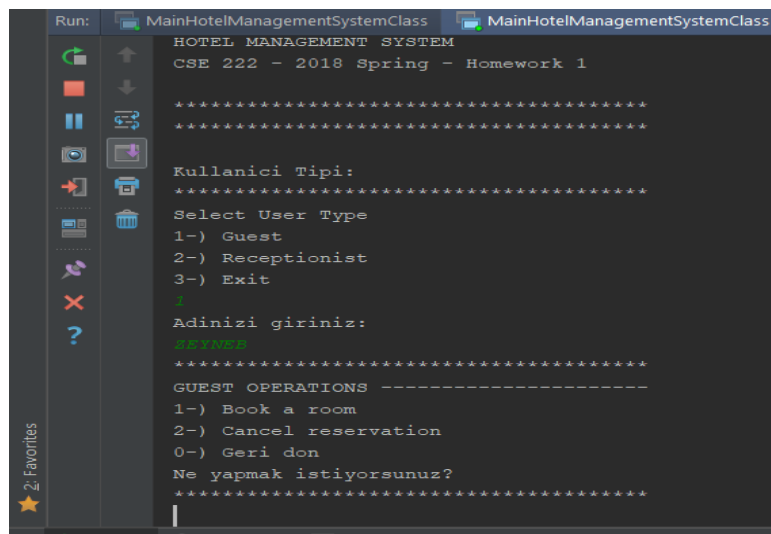


2.1.5. Reservation Test:



2.2. Running Results

1. Kullanıcı adı seçildi.. 1 numara ile guest ile sisteme girildi. Ad zeynep olarak girildi. Sonrasında hangi işlemi yapmak istediğimiz soruldu.



2. Seçeneklerden 1 ile oda rezervasyonu yapılmak istendi. Sonrasında 2 ile oda iptal edildi.

```
Room
MainHotelManagementSystemClass MainHotelManagementSystemClass
Run:
2-) Cancel reservation
0-) Geri don
Ne yapmak istiyorsunuz?
*****
Room 101 is booked for guest ZEYNEB.
*****
GUEST OPERATIONS -----
1-) Book a room
2-) Cancel reservation
0-) Geri don
Ne yapmak istiyorsunuz?
*****
Reservation for room '101' cancelled.
*****
GUEST OPERATIONS -----
1-) Book a room
2-) Cancel reservation
0-) Geri don
Ne yapmak istiyorsunuz?
*****
```

3. Guest kullanıcısı 0 seçeneği ile geri dönmek istedi.

```
RoomTest
MainHotelManagementSystemClass MainHotelManagementSystemClass
GUEST OPERATIONS -----
1-) Book a room
2-) Cancel reservation
0-) Geri don
Ne yapmak istiyorsunuz?
*****
Reservation for room '101' cancelled.
*****
GUEST OPERATIONS -----
1-) Book a room
2-) Cancel reservation
0-) Geri don
Ne yapmak istiyorsunuz?
*****
Kullanici Tipi:
*****
Select User Type
1-) Guest
2-) Receptionist
3-) Exit
```

```
*****
Kullanici Tipi:
*****
Select User Type
1-) Guest
2-) Receptionist
3-) Exit
*****
RECEPTIONIST OPERATIONS -----
1-) Book a room
2-) Checkin
3-) Checkout
4-) Save to csv
5-) Load from csv
0-) Back to main menu
Ne yapmak istiyorsunuz?
*****
Resepionist Reservation
Input guest name:
```

Kullacıya sisteme hangi kullanıcı olarak giriş yapmak istediği soruldu. 2. Seçenekle resepsiyonist olarak girildi. Resepsiyoniste neler yapmak istediği soruldu. Ve ardından kullacının adı soruldu.

```
Run: MainHotelManagementSystemClass MainHotelManagementSystemClass
Input guest name:
ahmet
Room 103 is booked for guest ahmet.
*****
RECEPTIONIST OPERATIONS -----
1-) Book a room
2-) Checkin
3-) Checkout
4-) Save to csv
5-) Load from csv
0-) Back to main menu
Ne yapmak istiyorsunuz?
*****
2
*****
Check-in Reservation
ACIK REZERVASYONLAR
1-) Room: 101 Guest: Zeyneb
2-) Room: 102 Guest: zeyneb
3-) Room: 103 Guest: ahmet
0-) Geri don
Checkin yapmak istediginiz kaydi seciniz
```

Ahmet isimli kullanıcı sisteme girildi ardından resepsiyoniste hangi işleme yapmak istediği tekrar sorulunca bu sefer checkin yapmak istedi 2. Seçenekle. Sistem resepsiyoniste checkin yapması için sistemde daha önce oda rezervasyonu yapan fakat checkin yapmamış kullacıları gösterdi.

```
Run: MainHotelManagementSystemClass MainHotelManagementSystemClass
0-) Geri don
Checkin yapmak istediginiz kaydi seciniz
2
Oda 103 icin checkin yapildi.
Room 103 checkin completed.
*****
RECEPTIONIST OPERATIONS -----
1-) Book a room
2-) Checkin
3-) Checkout
4-) Save to csv
5-) Load from csv
0-) Back to main menu
Ne yapmak istiyorsunuz?
*****
3
*****
Check-out Reservation
CHECKIN YAPILMIS ODALAR
1-) Room: 103 Guest: ahmet
0-) Geri don
Checkout yapmak istediginiz kaydi seciniz
```

Bu sefer resepsiyonist 3 numaralı kullacıyı seçerek onun için check in yaptı. Ardında da check out yapmak için 3. Seçeneği seçen resepsiyoniste hangi kullanıcı için check out yapmak istediği soruldu.

```
Run: MainHotelManagementSystemClass MainHotelManagementSys
Ne yapmak istiyorsunuz?
*****
3
*****
Check-out Reservation
CHECKIN YAPILMIS ODALAR
1-) Room: 103 Guest: ahmet
0-) Geri don
Checkout yapmak istediginiz kaydi seciniz
3
Reservation for room '103' cancelled.
Room 103 checkout completed.
*****
RECEPTIONIST OPERATIONS -----
1-) Book a room
2-) Checkin
3-) Checkout
4-) Save to csv
5-) Load from csv
0-) Back to main menu
Ne yapmak istiyorsunuz?
*****
```

Resepsiyonist istediği kullanıcıyı seçerek check out yaptı. Tekrar resepsiyonist sistemden çıkmak için 0 seçeneğini seçti ve sistemden çıktı.

```
MainHotelManagementSystemClass MainHotelManagementSystemClass
0-) Geri don
Checkout yapmak istediginiz kaydi seciniz
3
Reservation for room '103' cancelled.
Room 103 checkout completed.
*****
RECEPTIONIST OPERATIONS -----
1-) Book a room
2-) Checkin
3-) Checkout
4-) Save to csv
5-) Load from csv
0-) Back to main menu
Ne yapmak istiyorsunuz?
*****
Kullanici Tipi:
*****
Select User Type
1-) Guest
2-) Receptionist
3-) Exit
```

- Main titles -> 16pt , 2 line break
- Subtitles -> 14pt, 1.5 line break
- Paragraph -> 12pt, 1.5 line break