

**Gebze Technical University
ComputerEngineering**

CSE 222 -2018 Spring

HOMEWORK 1 REPORT

**Yunus ÇEVİK
141044080**

Course Assistant: Fatmanur ESİRCİ

1 INTRODUCTION

Hotel Management System programı bir otel otomasyon programıdır. Bu program ile otelde bulunan odaları rezerve etme ve rezervasyonu iptal etme , yapılan rezervasyonları check-in etme ve check-in edilmiş odaları check-out etme gibi işlemler yapılabilmektedir. Bu işleri yapan kişiler resepsiyonistler ve misafirlerdir. Hotel Management System e giriş yapan bir misafirin görevi rezervasyon yapma ve rezervasyonu iptal etmektir.

Resepsiyonistin görevi ise rezervasyon yapmak ve iptal etmek, check-in ve check-out yapmakla birlikte tüm kayıtlara erişebilmektir. Bu sistem içinde misafirlerin tutulduğu “Guests.csv”, odaların bilgilerinin tutulduğu “Rooms.csv” ve tüm kayıtların tutulduğu “Register.csv” dosyaları mevcuttur.

Note: Register işlemi istenilen özelliklere ek olarak yapılmıştır. Bu “Register.csv” dosyası oda bilgilerini, misafir bilgilerini, işlem yapan kişiyi, yapılan işlemi ve işlemin yapıldığı tarihi tutmaktadır.

1.1 Problem Definition

Bir otel otomasyon sisteminin ihtiyaç duyduğu gereksinimlerden en önemlisi otele giriş çıkışların sağlıklı bir şekilde kontrol edilmesidir. Bu durumda ortaya çıkan sistem açıklıklarından önemli olanları şunlardır:

1. Rezerve edilmiş bir odaya başka bir kişinin tekrardan rezerve edilmesi.
2. Check-in yapılmış bir odayı başka kişilerin rezerve ya da tekrardan check-in yapması.
3. Boş odaya rezervasyon iptali ve check-out işlemi yapılması.
4. İsim benzerliklerinden kaynaklanan rezervasyon ve check-in hataları.
5. Sisteme giriş yapmış olan misafirlerin bilgisinin tutulmaması ve aynı misafirin sürekli bilgilerini girmek zorunda kalması.

1.2 System Requirements

- 1) Hotel Management System veri girdileri, eğer mainTest üzerinden gerçekleştirilirse “mainTest.txt” dosyasından veriler alınarak sistem akışı sağlanır. Ancak main olarak çalışırıldığında kullanıcı bilgileri klavyeden input olarak girer ve sistem akışı bu şekilde sağlanır.

- 2) Hotel Management System, Receptionist ve Guest olmak üzere iki farklı grup tarafından kullanılır. Bu otel takip sistemi her iki gruba kullanım aşamasında farklı erişim hakları sağlar. Sistem kullanımı sade ve anlaşılırdır. Sistem çeşitli seçenekler sunarak ve bu seçenekleri örnekler ile açıklayarak kullanıcıya yol gösterir.
- 3) Hotel Management System, Guest' e sunulan rezervasyon yapma ve yapılan rezervasyonu iptal etme sisteminin istikrarlı bir şekilde ilerlemesi için öncelikle Guest aşağıda belirtilen bilgileri doldurarak rezervasyon işlemini yapabilir. Bu bilgiler şunlardır;
- Room ID
 - Guest ID
 - Guest Name
 - Guest Surname
- 4) Hotel Management System, Receptionist' e sunulan rezervasyon yapma, yapılan rezervasyonu iptal etme, check-in ve check-out yapma sisteminin istikrarlı bir şekilde ilerlemesi için Receptionist aşağıda belirtilen bilgileri doldurarak programa giriş yapmalıdır.
- Bu bilgiler şunlardır;
- User Name (Test için "Yunus")
 - Password (Test için "1234")
- 5) Bu otel takip sisteminde, oda numaraları ile ilişkili olarak oda kapasitesi belirlenir. Örnek verilmesi gerekirse 101 - 120 arası odalar tek kişilik, 301 - 320 arasındaki odalar üç kişilik kapasiteye sahiptir. Rezervasyon yapma veya check – in işlemlerinde Receptionist veya Guest oda kapasitesini belirlemelidir.
- 6) Sistem belirlenen oda kapasitesine bağlı olarak, boş (E), rezerve (R) ve dolu (F) odaların yer aldığı bir oda listesini kullanıcıya sunacaktır. Eğer kullanıcı rezerve veya dolu bir odaya tekrardan rezervasyon veya check-in işlemi yapmak isterse, sistem hata mesajı ile kullanıcıyı uyaracaktır. Bu şekilde sistemde bireysel hatadan kaynaklı problemlerin çıkması engellenecektir. Ayrıca kullanıcılar otelin doluluk bilgisine de kolayca erişebilme imkanına sahip olacaktır.
- 7) Hotel Management System' e daha önce giriş yapmış olan Guest' in bilgileri saklanır. Bu şekilde Guest bir sonraki işleminde bilgilerini tekrardan kaydetmek zahmetinde bulunmaz. Sadece kimlik bilgisini girmesi yeterlidir.
- 8) Sistemde meydana gelen değişimler bir kayıt defterinde tutulur. Bu kayıt defteri içerisinde;

- Oda bilgilerini (Room ID(oda numarası), Room Status(odanın durumu “E, R, F”), Room Capacity(odanın kaç kişilik olduğu))
- Misafir bilgilerini (Guest ID(misafir kimlik numarası), Guest Name(misafir adı), Guest Surname(misafir soyadı))
- Operant Person (İşlemi yapan kişi(resepsiyonist/misafir))
- Turnover (Yapılan işlemin sonucunda geri dönüşte ne olduğunu belirtir(Booking, Cancel Booking, Check-in, Check-out))
- Date (İşlem yapılan tarihi ve zamanı kaydeder.)

vardır.

public void run(); => Main de seçilen kişiler sonucu hangi sınıfın neleri çalıştıracağını belirleyen metoddur.

public int Operations(); => Resepsiyonist ya da Misafirin hangi işlemleri yaptıracağını belirten metoddur. Seçilen değerlere göre yanlış bir değer seçildiği takdirde kullanıcıdan tekrar input girilmesini bekler. Fonksiyon sonuç olarak integer döndürür.

public int queryRoomCapacity(); => Çeşitli odalara sahip otelde kaç kişilik odaların seçileceğini soran metoddur. Fonksiyon sonuç olarak, seçilen oda kapasitesi sonucu kaç kişilik olduğunu integer değer olarak döndürür. Ayrıca oda kapasitesine göre ekranda sadece 1 kişilik odalar görünür ve o odaların seçimi sağlanır.

public int bookRoom(); => Resepsiyonist ya da Misafirin çağırması sonucu oda bilgilerinin girilerek o odayı rezerve yapma işini gerçekleştiren metoddur. Ayrıca önceden giriş yapmış olan Misafir var ise sadece kimlik numarası ile misafirlerin kayıtlarının tutulduğu veri tabanı taranır. Daha sonra aynı kişi sisteme giriş yapmış ise diğer bilgilerini girmesine gerek olmadan odayı rezerve yapabilir. Fonksiyon sonuç olarak integer döndürür. Bu değerler “-2” rezervasyon işleminden bir önceki menüye geri dönmek için kullanılır. “RoomID” işlem tamamlandıktan sonra başka bir metodun kullanılması için integer değer döndürür.

public int cancelBookRoom(String operantPerson) throws IOException; => Resepsiyonist ya da Misafirin çağırması sonucu oda bilgilerinin girilerek rezerve edilmiş odayı iptal edip oda bilgilerini sıfırlayan metoddur. “operantPerson” (iptal işini yapan kişi) parametre girdisine ihtiyaç vardır. Fonksiyon sonuç olarak integer döndürür. Bu değerler “-2” rezervasyon işleminden bir önceki menüye geri dönmek için kullanılır. “0” metodun başarılı sonuçlandığı integer değerdir. Ayrıca IOException Metod içerisinde çağrılan dosyaya yazma işlemi sonucu, bir sorunla karşılaşılabilir gibisinden fırlatılan exceptiondır.

public boolean checkRoomID(int id, char status); => Parametrelere verilen değerler sonucu girilen oda numarasının var olup olmadığını ve kim tarafından çağrıldığını belirten

status değeri ile rezervasyon yapma ya da iptal etme, check-in ya da check-out yapma işlemlerinde nasıl odaları kontrol edeceğini belirten metoddur. Yani metodlar içinde çağırılması ile farklı yapıda çalışan bir metoddur. "id" (kontrol edilecek oda numarası) ve "status" (Hangi metodun çağırdığını belirten parametredir.) parametreler girdilerine ihtiyaç vardır. Fonksiyon sonuç olarak boolean değer döndürür. Eğer oda numarası mevcut ve olumlu bir işlem gerçekleştirilmiş ise "true", değil ise "false" return eder.

public boolean checkGuestID(int guestID,int roomID,char status); => Çağırılan metodlara göre farklı çalışan bir metoddur. Alınan parametrelerde statuse göre işlem gerçekleştirir. Eğer status değeri 'R' ise rezervasyon yapılacaktır ve sadece misafir kimlik numarasına göre eskiden sisteme giriş yapıp yapmadığı kontrol edilir. Eğer status değeri 'R' haricinde ise diğer metodlar çağırmış demektir ve odalar içerisinde misafir kimlik numarası ile sorgulama yapılır. Kişi kimlik numarası oda içerisinde bulunan kimlik değeri ile eşleşip eşleşmediği kontrol edilir. "guestID" (Misafirin kimlik numarası), "roomID" (Otelde bulunan oda numarası) ve "status" (Bu metodu hangi metodun ne amaçla çağırdığını belirten parametre) parametreler girdilerine ihtiyaç vardır. Fonksiyon sonuç olarak boolean değer döndürür. Eğer oda numarası mevcut ve olumlu bir işlem gerçekleştirilmiş ise "true", değil ise "false" return eder.

public int roomAndGuestIDInput(char status); => Oda numaraları ve misafir kişilerin input gireceği metoddur. Ancak status durumu = 'R' rezervasyon durumunda sadece kullanıcıya oda numarası girdisi sunmaktadır. "status" (Bu metodu hangi metodun ne amaçla çağırdığını belirten parametre) parametre girdisine gerek vardır. Fonksiyon sonuç olarak integer değer döndürür. Bu değerler "-2" rezervasyon işleminden bir önceki menüye geri dönmek için kullanılır. "RoomID" işlem tamamlandıktan sonra başka bir metodun kullanılması için integer değer döndürür.

public void registerWriteFile(String fileName, String operantPerson, String turnOver, Date date) throws IOException; => "Register.csv" dosyasında tüm yapılan işlemlerin tutulduğu bu dosyaya yazma metodudur. Oda bilgileri ile birlikte girilen parametre değerlerini dosyaya yazma işlemini sağlar. "fileName" (İşlem yapılması istenen dosya ismi), "operantPerson" (Yapılması gereken işlemi gerçekleştiren kişi. (Resepsiyonist ya da Guest)) ve "date" (Yapılması gerek işlemin ne zaman yapıldığını belirten tarih) parametreler girdilerine ihtiyaç vardır. Ayrıca IOException Metod içerisinde çağrılan dosyaya yazma işlemi sonucu, bir sorunla karşılaşılabilir gibisinden fırlatılan exceptiondır.

public void roomsWriteFile(String fileName) throws IOException; => "Rooms.csv" dosyasında oda ile ilgili yapılan değişikliklerin kaydedildiği dosya yazma metodudur. "fileName" (İşlem yapılması istenen dosya ismi) parametre girdisine ihtiyaç vardır. Ayrıca

IOException Metod içerisinde çağrılan dosyaya yazma işlemi sonucu, bir sorunla karşılaşılabılır gibisinden fırlatılan exceptiondır.

public void guestsWriteFile(String fileName) throws IOException; => "Guests.csv" dosyasında oda ile ilgili yapılan değişikliklerin kaydedildiği dosya yazma metodudur. "fileName" (İşlem yapılması istenen dosya ismi) parametre girdisine ihtiyaç vardır. Ayrıca IOException Metod içerisinde çağrılan dosyaya yazma işlemi sonucu, bir sorunla karşılaşılabılır gibisinden fırlatılan exceptiondır.

public void roomsReadFile(String fileName); => "Rooms.csv" dosyası içerisinde bulunan değerlerin satır satır okunup, split edilerek roomArr arrayine atılması işlemi yapan metoddur. "fileName" (İşlem yapılması istenen dosya ismi) parametre girdisine ihtiyaç vardır.

public void GuestsReadFile(String fileName); => "Guests.csv" dosyası içerisinde bulunan değerlerin satır satır okunup, split edilerek guestArr arrayine atılması işlemi yapan metoddur. "fileName" (İşlem yapılması istenen dosya ismi) parametre girdisine ihtiyaç vardır.

private void receptionistLogin(); => Resepsiyonistin login olması için istenen Kullanıcı adı ve password bilgilerini isteyen metoddur. Şu anlık tek bir resepsiyonist olduğundan test için şu şekilde kontrol edilmektedir.

FOR TESTING : User Name: = 'Yunus'

FOR TESTING : Password: = '1234'

protected int checkINorOut(String inORout) throws IOException; => Check IN ya da Check OUT yapılması için resepsiyonistin kullanabildiği metoddur. Yapılacak işleme göre metodlar çağrılır ve işlem gerçekleştirilir. "inORout" (CHECK IN ya da CHECK OUT olduğunu belirten parametredir) parametre girdisine gerek vardır. Fonksiyon sonuç olarak integer döndürür. Bu değerler "-2" Check IN ya da Check OUT işlemlerinden bir önceki menüye geri dönmek için kullanılır. "0" metodun başarılı sonuçlandığı integer değerdir. Ayrıca IOException Metod içerisinde çağrılan dosyaya yazma işlemi sonucu, bir sorunla karşılaşılabılır gibisinden fırlatılan exceptiondır.

private boolean checkIn() => Check IN yapılıp yapılamayacağını belirten metoddur. Eğer rezervasyon yaptırdıysa oda numarası ve kimlik numarasıyla, rezerve yaptırmadıysa tüm bilgileri ile check in yaptırabilir. Check IN yapılması için kullanıcıdan confirm() metoduna Yes(Y/y) cevabını vermesi yeterlidir. Fonksiyon sonuç olarak boolean değer döndürür. Bu değerler kullanıcı Check IN yapabilir ve onay verdiğinde return true - yapamaz ya da onay vermez ise return false.

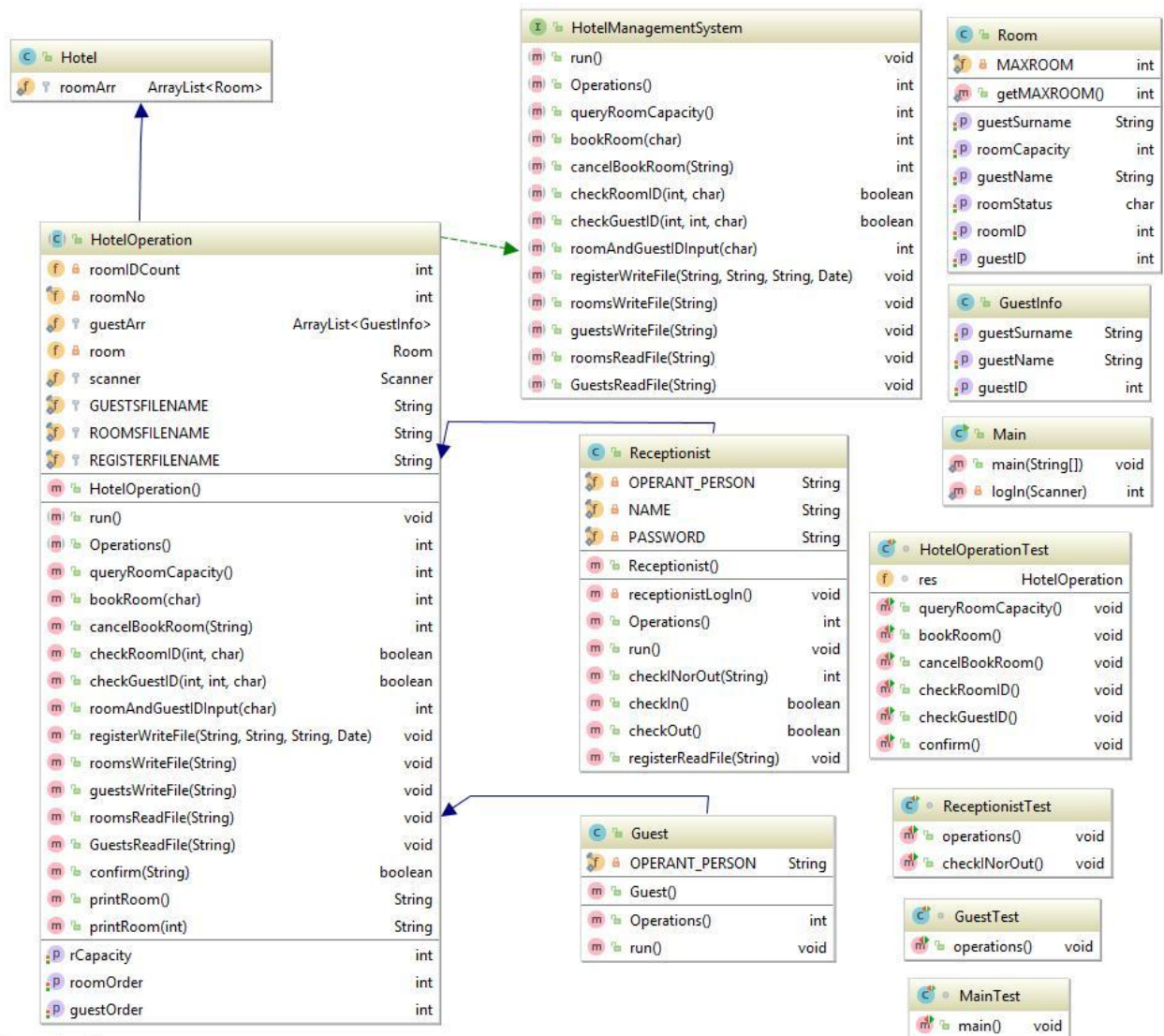
private boolean checkOut() throws IOException; => Check OUT yapılıp

yapılamayacağını belirten metoddur. Check OUT yapılması için kullanıcıdan confirm() metoduna Yes(Y/y) cevabını vermesi yeterlidir. Ayrıca Check OUT yapıldığı takdirde oda bilgilerinin tutulduğu ArrayListten de Check OUT yapılan oda bilgileri sıfırlanır. Fonksiyon sonuç olarak boolean değer döndürür. Bu değerler kullanıcı Check IN yapabilir ve onay verdiğinde return true - yapamaz ya da onay vermez ise return false.

public void registerReadFile(String fileName); => Resepsiyonist yapmış olduğu tüm bilgileri görmek için "Register.csv" dosyası içindeki bilgileri ekranında görebilmesi için oluşturulmuş metoddur. "fileName" işlem yapılması istenen dosya ismi parametre girdisine gerek vardır.

2 METHOD

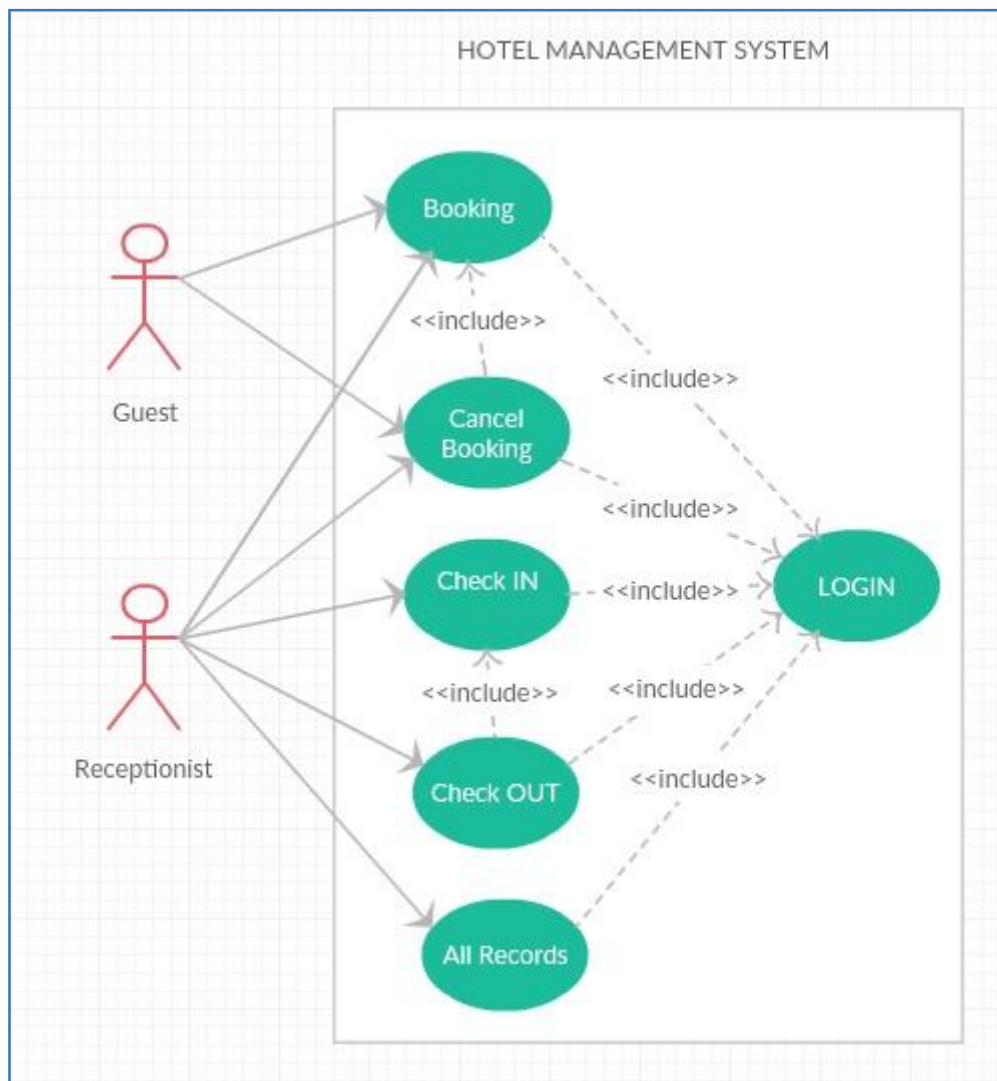
2.1 Class Diagrams



- Interface olan HotelManagementSystem sınıfının objesi oluşturulmadığından ve içerisinde implementation yapılamadığından başka bir sınıf olan HotelOperation'a implements etmektedir.
- Hotel sınıfı sahip olduğu bilgileri miras bırakabilmek için, abstract sınıfı olan HotelOperation'a extends etmektedir.
- Abstract olan HotelOperation sınıfının objesi oluşturulmadığından dolayı, Receptionist ve Guest sınıflarına extends ederek içerisindeki bilgiler bu sınıflarda ortak kullanılmaktadır.

2.2 Use Case Diagrams

a) Diagram



b) Use Case Diagrams Documentation:

b.1.)

Adı: Yeni Rezervasyon Oluřturma (Booking)

Aktörler: Resepsiyonist ve Misafir.

Ön Kořullar: Sisteme login olunmalıdır. Login işleminde kullanıcıya resepsiyonist mi, yoksa misafir mi olduđu sorulur. Kullanıcı eđer resepsiyonist ise 1'e, misafir ise 2'e basarak sisteme giriş yapabilir.

Son Durum: "Guests.csv", "Rooms.csv" ve "Register.csv" dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Genişleme Noktası: Rezervasyon işleminin gerçekleşmesi ve müşteri bilgi kayıdır.

Başarılı Senaryo:

1. Eđer Login işleminde 1 yani Resepsiyonist giriş yapmış ise (A1)
2. Login işleminde 2 yani Misafir giriş yapmış ise.
3. Kullanıcıya oda tipleri sunulur. Otel 101 – 120, 201 – 220, 301 – 320, 401 – 420, 501 – 520 arasında olmak üzere 100 adet odaya sahiptir. Odaların kişi kapasitesi oda numaraları ile ilişkilidir. Örneğin 315 numaralı oda 3 kişilik bir odayı temsil eder. Ayrıca oda numaraları yanında oda durumunu anlatan "E (Empty)" boş odayı, "R (Reservation)" rezerve olan odayı ve "F (Full)" Check-in yapılmış dolu odayı temsil eder.
4. Kullanıcıya sorulan işlem tiplerinden 1 yani rezervasyon yapma (Booking) seçilir.
5. Kullanıcı oda kapasitesine göre tipini seçer.
6. Odaya ait bilgiler kullanıcıdan istenir. Bu bilgiler oda numarası, misafir kimlik numarasıdır (Bu sistemde kimlik numarası 9 haneli integer değerlerdir. Ayrıca sisteme misafir bilgileri girilmiş ise eski bilgi "Guests.csv" dosyasına kaydedildiğinden dolayı kullanıcı tekrardan ad ve soyad bilgilerini girmek zorunda kalmaz.), misafir adı ve misafir soyadıdır.
7. Kullanıcıya işlem onayı yönlendirilir. Eđer Yes (Y/y) seçilirse onay gerçekleşir. Eđer No (N/n) seçilirse işlem iptal edilir (A2)
8. Rezervasyon yapılır.
9. "Guests.csv", "Rooms.csv" ve "Register.csv" dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Alternatif Yollar:

- A1. Resepsiyonist kullanıcı adını ve şifresini girmelidir. Örnek kullanıcı adı “Yunus”, şifre “1234” olarak girildiği takdirde Resepsiyonist giriş yapıp kapsamlı işlemleri yapma yetkisine sahip olur. Daha sonra 3. adıma geri dönlür.
- A2. Use case iptal edilir ve herhangi bir deęişiklik yapılmaz ve 6. adıma geri döner.

b.2.)

Adı: Rezervasyon İptali (Cancel Booking)

Aktörler: Resepsiyonist ve Misafir.

Ön Koşullar: Sisteme login olunmalı ve önceden rezervasyon yaptırmalıdır. Login işleminde kullanıcıya resepsiyonist mi, yoksa misafir mi olduęu sorulur. Kullanıcı eęer resepsiyonist ise 1’e, misafir ise 2’e basarak sisteme giriş yapabilir.

Son Durum: “Guests.csv”, “Rooms.csv” ve “Register.csv” dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Genişleme Noktası: Rezervasyon işleminin iptali ve oda bilgilerinin sıfırlanmasıdır.

Başarılı Senaryo:

1. Eęer Login işleminde 1 yani Resepsiyonist giriş yapmış ise (A3)
2. Login işleminde 2 yani Misafir giriş yapmış ise.
3. Kullanıcıya oda tipleri sunulur. Otel 101 – 120, 201 – 220, 301 – 320, 401 – 420, 501 – 520 arasında olmak üzere 100 adet odaya sahiptir. Odaların kişi kapasitesi oda numaraları ile ilişkilidir. Örneęin 315 numaralı oda 3 kişilik bir odayı temsil eder. Ayrıca oda numaraları yanında oda durumunu anlatan “E (Empty)” boş odayı, “R (Reservation)” rezerve olan odayı ve “F (Full)” Check-in yapılmış dolu odayı temsil eder.
4. Kullanıcıya sorulan işlem tiplerinden 2 yani rezervasyon iptal etme (Cancel Booking) seçilir.
5. Kullanıcı oda kapasitesine göre tipini seçer.
6. Oda numarası ve misafir kimlik numarası girilir.
7. Kullanıcıya işlem onayı yönlendirilir. Eęer Yes (Y/y) seçilirse onay gerçekleşir. Eęer No (N/n) seçilirse işlem iptal edilir (A4)
8. Rezervasyon iptal edilir.
9. “Rooms.csv” ve “Register.csv” dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Alternatif Yollar:

- A3. Resepsiyonist kullanıcı adını ve şifresini girmelidir. Örnek kullanıcı adı “Yunus”, şifre “1234” olarak girildiği takdirde Resepsiyonist giriş yapıp kapsamlı işlemleri yapma yetkisine sahip olur. Daha sonra 3. adıma geri dönlür.
- A4. Use case iptal edilir ve herhangi bir deęişiklik yapılmaz ve 6. adıma geri döner.

b.3.)

Adı: Check IN

Aktörler: Resepsiyonist.

Ön Koşullar: Sisteme login olunmalıdır. Login işleminde kullanıcıya resepsiyonist mi, yoksa misafir mi olduęu sorulur. Kullanıcı eęer resepsiyonist ise 1’e, misafir ise 2’e basarak sisteme giriş yapabilir.

Son Durum: “Guests.csv”, “Rooms.csv” ve “Register.csv” dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Genişleme Noktası: Rezervasyon işleminin gerçekleşmesi ve müşteri bilgi kaydıdır.

Başarılı Senaryo:

1. Eęer Login işleminde 1 yani Resepsiyonist giriş yapmış ise (A5)
2. Login işleminde 2 yani Misafir giriş yapmış ise.(Check-in yapamaz.)
3. Kullanıcıya oda tipleri sunulur. Otel 101 – 120, 201 – 220, 301 – 320, 401 – 420, 501 – 520 arasında olmak üzere 100 adet odaya sahiptir. Odaların kişi kapasitesi oda numaraları ile ilişkilidir. Örneęin 315 numaralı oda 3 kişilik bir odayı temsil eder. Ayrıca oda numaraları yanında oda durumunu anlatan “E (Empty)” boş odayı, “R (Reservation)” rezerve olan odayı ve “F (Full)” Check-in yapılmış dolu odayı temsil eder.
4. Kullanıcıya sorulan işlem tiplerinden 3 yani check-in seçilir.
5. Kullanıcı oda kapasitesine göre tipini seçer.
6. Kullanıcı, misafirin daha önce rezervasyon yapıp yapmadığını öğrenir. Eęer rezervasyon yapmamış ise (A6), eęer rezervasyon yapmış ise (A7)
7. Kullanıcıya işlem onayı yönlendirilir. Eęer Yes (Y/y) seçilirse onay gerçekleşir. Eęer No (N/n) seçilirse işlem iptal edilir (A6)
8. Check-in yapılır.
9. “Rooms.csv” ve “Register.csv” dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Alternatif Yollar:

- A5. Resepsiyonist kullanıcı adını ve şifresini girmelidir. Örnek kullanıcı adı “Yunus”, şifre “1234” olarak girildiği takdirde Resepsiyonist giriş yapıp kapsamlı işlemleri yapma yetkisine sahip olur. Daha sonra 3. adıma geri dönlür.
- A6. Odaya ait bilgiler kullanıcıdan istenir. Bu bilgiler oda numarası, misafir kimlik numarasıdır (Bu sistemde kimlik numarası 9 haneli integer değerlerdir. Ayrıca sisteme misafir bilgileri girilmiş ise eski bilgi “Guests.csv” dosyasına kaydedildiğinden dolayı kullanıcı tekrardan ad ve soyad bilgilerini girmek zorunda kalmaz.), misafir adı ve misafir soyadıdır. Daha sonra 20. adıma geri döner.
- A7. Oda numarası ve misafir kimlik numarası girilir. 20. adıma geri döner.
- A8. Use case iptal edilir ve herhangi bir değişiklik yapılmaz ve 6. adıma geri döner.

b.4.)

Adı: Check OUT.

Aktörler: Resepsiyonist.

Ön Koşullar: Sisteme login olunmalı ve önceden check – in yaptırılmalıdır. Login işleminde kullanıcıya resepsiyonist mi, yoksa misafir mi olduğu sorulur. Kullanıcı eğer resepsiyonist ise 1’e, misafir ise 2’e basarak sisteme giriş yapabilir.

Son Durum: “Rooms.csv” ve “Register.csv” dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Genişleme Noktası: Check-out işleminin yapılması ve oda bilgilerinin sıfırlanmasıdır.

Başarılı Senaryo:

1. Eğer Login işleminde 1 yani Resepsiyonist giriş yapmış ise (A9)
2. Login işleminde 2 yani Misafir giriş yapmış ise. (Check-out yapamaz.)
3. Kullanıcıya oda tipleri sunulur. Otel 101 – 120, 201 – 220, 301 – 320, 401 – 420, 501 – 520 arasında olmak üzere 100 adet odaya sahiptir. Odaların kişi kapasitesi oda numaraları ile ilişkilidir. Örneğin 315 numaralı oda 3 kişilik bir odayı temsil eder. Ayrıca oda numaraları yanında oda durumunu anlatan “E (Empty)” boş odayı, “R (Reservation)” rezerve olan odayı ve “F (Full)” Check-in yapılmış dolu odayı temsil eder.
4. Kullanıcıya sorulan işlem tiplerinden 4 yani Check-out yapma seçilir.
5. Kullanıcı oda kapasitesine göre tipini seçer.
6. Oda numarası ve misafir kimlik numarası girilir.

7. Kullanıcıya işlem onayı yönlendirilir. Eğer Yes (Y/y) seçilirse onay gerçekleşir. Eğer No (N/n) seçilirse işlem iptal edilir (A10)
8. Check-out yapılır.
9. “Rooms.csv” ve “Register.csv” dosyalarına yapılan bu işlem sonucu bilgiler kaydedilir.

Alternatif Yollar:

- A9. Resepsiyonist kullanıcı adını ve şifresini girmelidir. Örnek kullanıcı adı “Yunus”, şifre “1234” olarak girildiği takdirde Resepsiyonist giriş yapıp kapsamlı işlemleri yapma yetkisine sahip olur. Daha sonra 3. adıma geri dönülür.
- A10. Use case iptal edilir ve herhangi bir değişiklik yapılmaz ve 6. adıma geri döner.

b.5.)

Adı: Tüm kayıtlar (All Records)

Aktörler: Resepsiyonist.

Ön Koşullar: Sisteme login olunmalıdır. Login işleminde kullanıcıya resepsiyonist mi, yoksa misafir mi olduğu sorulur. Kullanıcı eğer resepsiyonist ise 1’e, misafir ise 2’e basarak sisteme giriş yapabilir.

Son Durum: “Register.csv” dosyası okunur ve ekranda çıktı görüntülenir.

Başarılı Senaryo:

1. Eğer Login işleminde 1 yani Resepsiyonist giriş yapmış ise (A11)
2. Login işleminde 2 yani Misafir giriş yapmış ise. (All Records yapamaz.)
3. Kullanıcıya oda tipleri sunulur. Otel 101 – 120, 201 – 220, 301 – 320, 401 – 420, 501 – 520 arasında olmak üzere 100 adet odaya sahiptir. Odaların kişi kapasitesi oda numaraları ile ilişkilidir. Örneğin 315 numaralı oda 3 kişilik bir odayı temsil eder. Ayrıca oda numaraları yanında oda durumunu anlatan “E (Empty)” boş odayı, “R (Reservation)” rezerve olan odayı ve “F (Full)” Check-in yapılmış dolu odayı temsil eder.
4. Kullanıcıya sorulan işlem tiplerinden 5 yani All Records yapma seçilir.
5. “Register.csv” dosyası okunur ve ekranda çıktı görüntülenir.

Alternatif Yollar:

- A11. Resepsiyonist kullanıcı adını ve şifresini girmelidir. Örnek kullanıcı adı “Yunus”, şifre “1234” olarak girildiği takdirde Resepsiyonist giriş yapıp kapsamlı işlemleri yapma yetkisine sahip olur. Daha sonra 3. adıma geri dönülür.

2.3 OtherDiagrams (optional)

Diğer diagramlara ihtiyaç duymadım gerekli olarak gördüklerim Use Case ve Class diagramlardır. (No need other diagrams.)

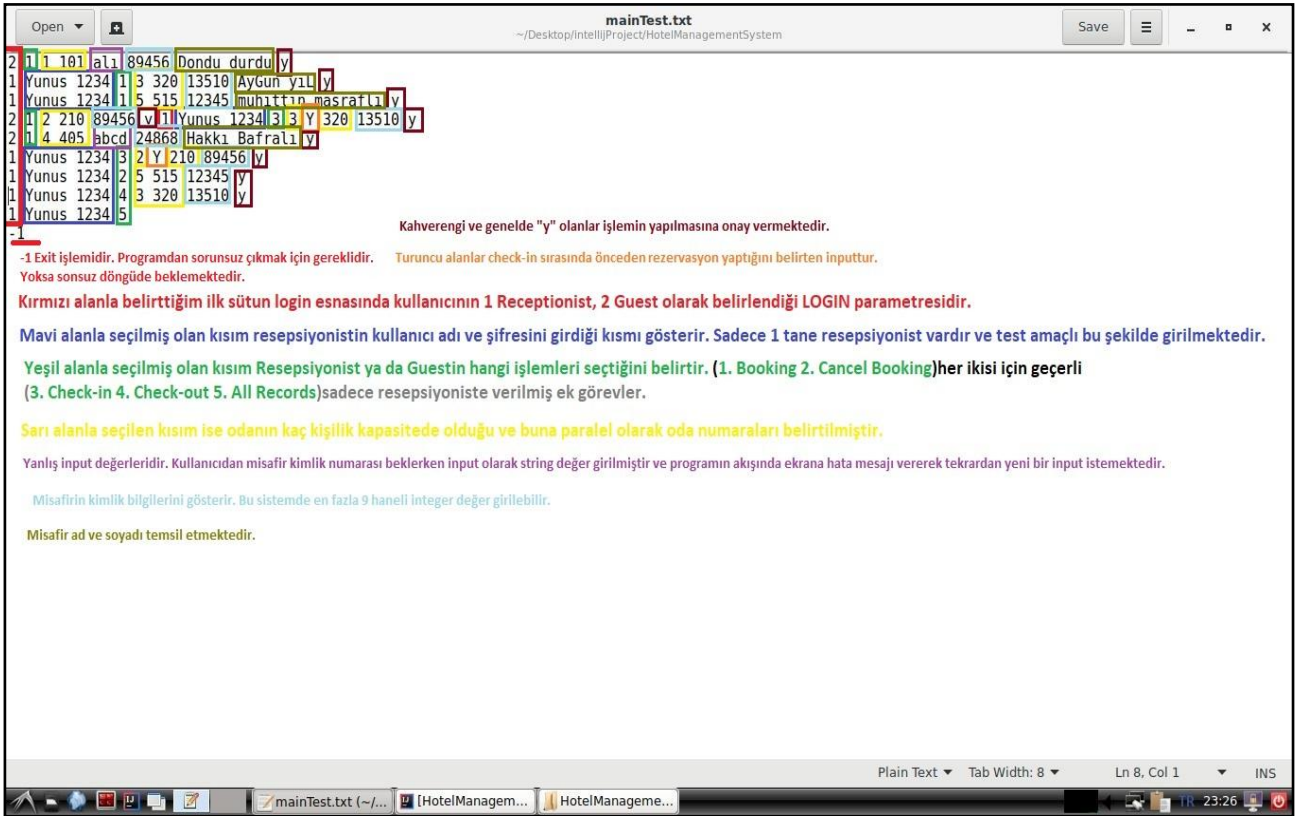
2.4 Problem Solution Approach

Oteller gerek sistemsel gerekse misafirler açısından karmaşanın oluşabileceği alanlardır. Bu karmaşıklık yöneticilerden ve sistemden kaynaklı olabilir. Bu durum misafir memnuniyetini ve yöneticiler için yoğunluk kontrolünü etkiler. Bu kontrol kaybını ve memnuniyetsizliği minimuma indirmek HotelManagementSystem' in temel amacıdır. Bu amaçla hareket eden sisteme iki tip kullanıcı atanmıştır ve sistem kullanıcıları sahip oldukları ID ile ayrılırlar ve kontrol edilirler. Otelde bulunan odalara, oda numaraları ile ilişkili olarak oda kapasitesi atanmıştır. Bu durum ihtiyaca bağlı olarak oda seçimini kolaylaştırır. Ayrıca otelde rezerve edilen odalar “ R ”, dolu odalar “ F ”, boş odalar “ E ” harfleri ile sembolize edilmiştir. Bu şekilde resepsiyonist ya da misafir oda bilgilerinin tutulduğu panele bakarak odaların seçilebilirliği hakkında bilgi sahibi olabilirler. Eğer resepsiyonist ya da misafir rezerve veya dolu bir oda üzerinde tekrardan rezervasyon veya check-in yapmak isterse sistem kullanıcıyı hata mesajı ile uyarır. Bu şekilde kullanıcı yeni bir oda seçmek için yönlendirilir ve aynı oda birden çok kişi tarafından rezerve edilemez. Bu sistem resepsiyonistlerin ve misafirlerin yaptığı işlemleri, işlemlerin gerçekleştiği tarihi ve saati kayıt defterine kayıt ederek oluşabilecek güvenlik veya güvenilirlik kaybının önüne geçer.

3 RESULT

3.1 Test Cases

- a) **MainTest:** Programın genel akışını proje dosyası içinde yer alan “mainTest.txt” dosyasından inputlar ile test ettiğimiz bir test casedir. Bu test case de olayın akışı bozulmayacak şekilde inputlar verilmiştir. Aşağıda “mainTest.txt” dosyası bulunmaktadır screen shot üzerinde nelerin olduğu belirtilmektedir.



```
2 1 1 101 ali 89456 Dondurdu y
1 Yunus 1234 1 3 320 13510 Aygun yil y
1 Yunus 1234 1 5 515 12345 muhittin masraflı y
2 1 2 210 89456 y 11 Yunus 1234 3 3 y 320 13510 y
2 1 4 405 abcd 24868 Hakkı Bafra y
1 Yunus 1234 3 2 y 210 89456 y
1 Yunus 1234 2 5 515 12345 y
1 Yunus 1234 4 3 320 13510 y
1 Yunus 1234 5
```

Kahverengi ve genelde "y" olanlar işlemin yapılmasına onay vermektedir.

-1 Exit işlemidir. Programdan sorunsuz çıkmak için gereklidir. Turuncu alanlar check-in sırasında önceden rezervasyon yaptığını belirten inputtur. Yoksa sonsuz döngüde beklemektedir.

Kırmızı alanla belirttiğim ilk sütun login esnasında kullanıcının 1 Receptionist, 2 Guest olarak belirlendiği LOGIN parametresidir.

Mavi alanla seçilmiş olan kısım resepsiyonistin kullanıcı adı ve şifresini girdiği kısmı gösterir. Sadece 1 tane resepsiyonist vardır ve test amaçlı bu şekilde girilmektedir.

Yeşil alanla seçilmiş olan kısım Resepsiyonist ya da Guestin hangi işlemleri seçtiğini belirtir. (1. Booking 2. Cancel Booking) her ikisi için geçerli (3. Check-in 4. Check-out 5. All Records) sadece resepsiyoniste verilmiş ek görevler.

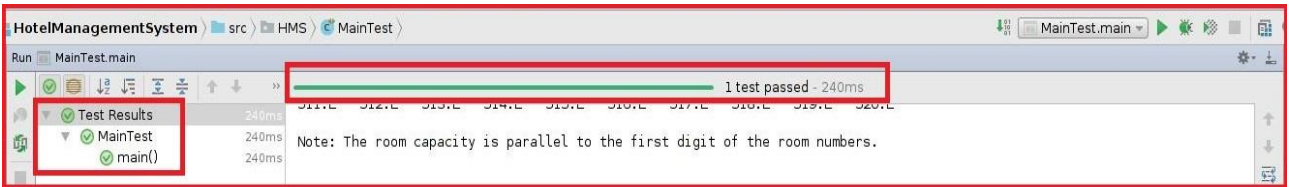
Sarı alanla seçilen kısım ise odanın kaç kişilik kapasitede olduğu ve buna paralel olarak oda numaraları belirtilmiştir.

Yanlış input değerleridir. Kullanıcıdan misafir kimlik numarası beklerken input olarak string değer girilmiştir ve programın akışında ekrana hata mesajı vererek tekrardan yeni bir input istemektedir.

Misafirin kimlik bilgilerini gösterir. Bu sistemde en fazla 9 haneli integer değer girilebilir.

Misafir ad ve soyadı temsil etmektedir.

Main Test işleminde alınan inputlardan çıkan test sonucu ise aşağıda belirtmektedir.



```
Run MainTest.main
Test Results
MainTest
main()
1 test passed - 240ms
Note: The room capacity is parallel to the first digit of the room numbers.
```

Note: Main Test işlemi bittikten sonra tekrar çalıştırmak istendiğinde rezerve yapılmış odaları tekrardan rezerve yapmak isteyeceğinden hata verecektir. Bu yüzden “Guests.csv”, “Rooms.csv” ve “Register.csv” dosyalarının proje klasöründen silinerek, tekrardan çalıştırmada herhangi bir sıkıntı çıkmayacaktır.

b) **Unit Test:** Classların içerisinde bulunan metodların bireysel olarak test edilmesidir.

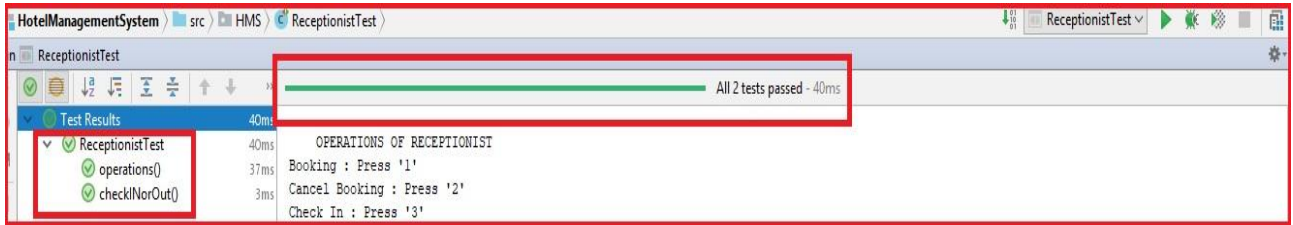
Bazı metodlar void return edeceğinden ve içerisinde birden fazla metodun çağrılacağından test etmedim. Ancak içerisinde bulunan yardımcı metodların bir kısmını test ettim. Ayrıca Main Testte de olay akışı bozulmadan çalıştığından ve klavyeden input girerek test ettiğimden bir sıkıntı çıkmamaktadır.

- **ReceptionistTest:** Receptionist classının içerisinde operations() ve checkInNorOut() metodlarının testinde input parametreleri olarak

operations() metoduna HotelOperation.*scanner* = **new** Scanner("1 2 3 4 5");

checkInNorOut() metoduna check-in ve check-out için

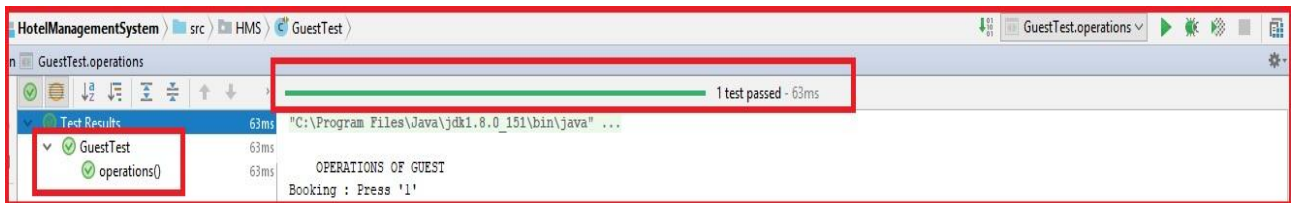
HotelOperation.*scanner* = **new** Scanner("y 1 101 -2 -1"); parametreleri verdim ve sonuç olarak aşağıda testlerden geçtiği görünmektedir.



- **GuestTest:** Guest classının içerisinde bulunan operations() metodu testinde input parametreleri olarak

operations() metoduna HotelOperation.*scanner* = **new** Scanner("1 2 -1");

parametrelerini verdim ve sonuç olarak aşağıda testten geçtiği görünmektedir.



- **HotelOperationTest:** HotelOperation abstract classının içerisinde bulunan queryRoomCapacity(), bookRoom(), cancelBookRoom(), checkRoomID(), checkGuestID(), confirm() metodlarının testinde input parametreleri olarak

queryRoomCapacity() metoduna HotelOperation.*scanner* = **new** Scanner("1 2 3 4 5");

bookRoom() metoduna HotelOperation.*scanner* = **new** Scanner("1 110 5555 Hakkı Bulut y -1");

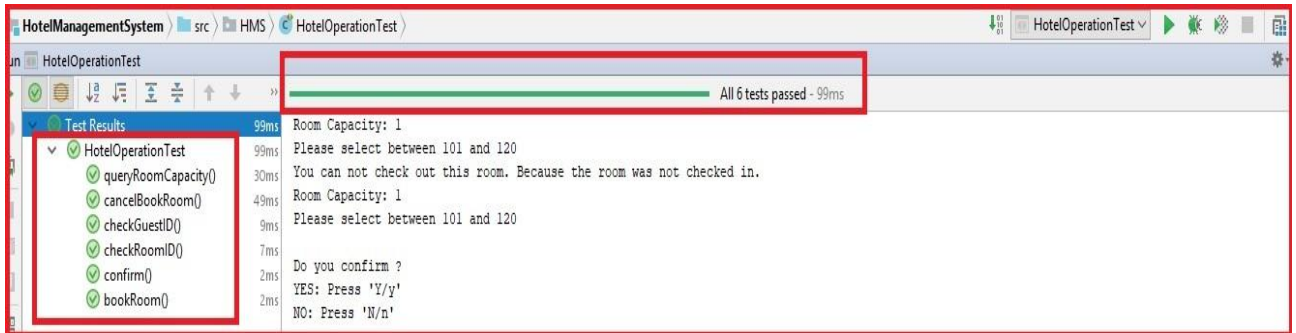
cancelBookRoom() metoduna `HotelOperation.scanner = new Scanner("1105555 y -1")`;

checkRoomID() metoduna `res.setCapacity(1)`;
`assertTrue(res.checkRoomID(110,'R'))`;
`assertFalse(res.checkRoomID(110,'I'))`;
`assertFalse(res.checkRoomID(110,'O'))`;

checkGuestID() metoduna `res.setCapacity(1)`;
`assertTrue(res.checkGuestID(5555,110,'R'))`;
`assertTrue(res.checkGuestID(5555,110,'I'))`;
`assertFalse(res.checkGuestID(5555,100,'O'))`;

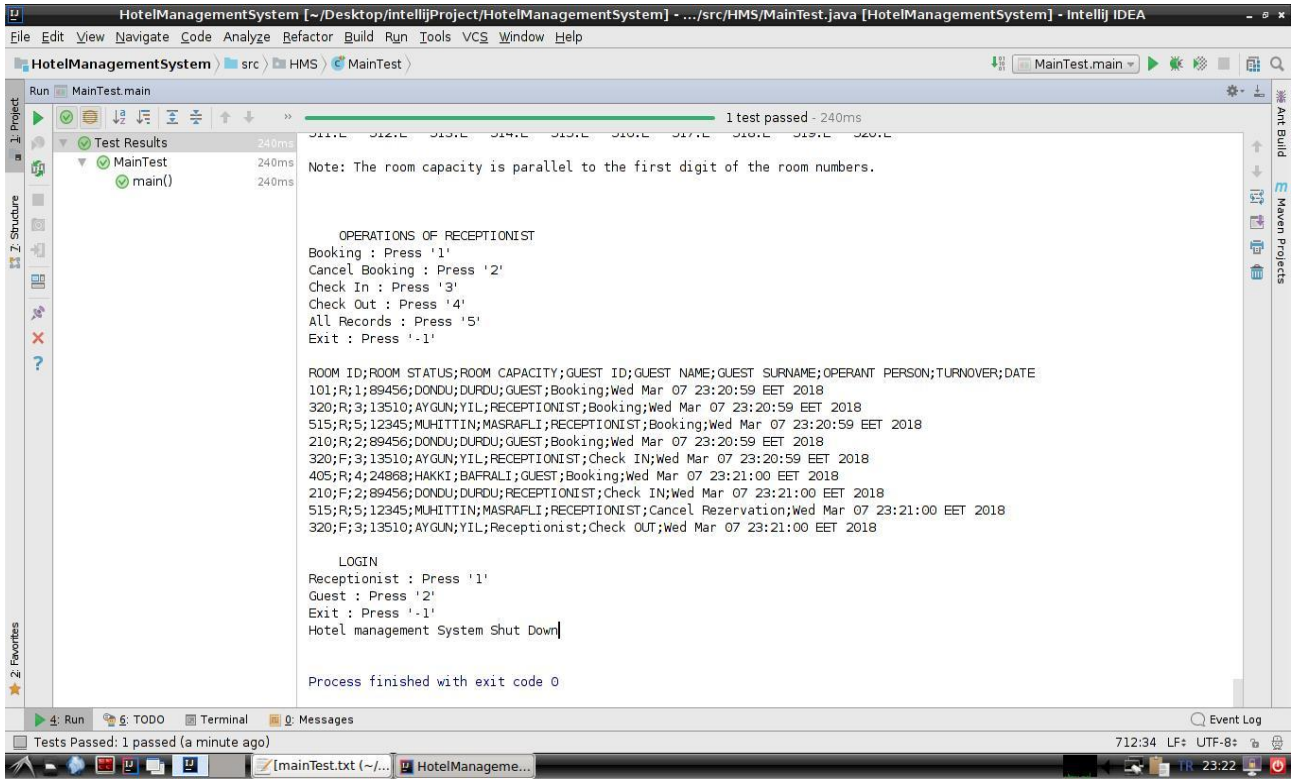
confirm() metoduna `HotelOperation.scanner = new Scanner("Y y N n -1")`;

parametreleri verdim ve sonuç olarak aşağıda testlerden geçtiği görünmektedir.



3.2 Running Results

Main Test işleminden sonra “mainTest.txt” içerisinde bulunan input değerleri kullanıcı klavyeden girmiş gibi okunur ve işleme tabi tutulur. Main Test ten geçtikten sonra resepsiyonistin özelliklerinden biri olan tüm kayıtlar(All Records) çağrılarak bütün kayıtların tutulduğu “Register.csv” dosyasından bilgiler çekilir ve ekrana çıktı olarak sunulur. Main Testin başarılı olduğu hangi işlemlerin yapıldığı iş bitiminde kayıt edildiğinden bu kısımdan nelerin yapıldığı anlaşılabilir.



Ayrıca virtual machine de çalışan Main Test sonucu değişen dosyaların çıktıları ve bu dosyaların Microsoft Exel gibi programda tablo şeklinde gösterimi gösterilmektedir.

Open

Register.csv
~/Desktop/intellijProject/HotelManagementSystem

```
ROOM ID;ROOM STATUS;ROOM CAPACITY;GUEST ID;GUEST NAME;GUEST SURNAME;OPERANT PERSON;TURNOVER;DATE
101;R;1;89456;DONDURDU;GUEST;Booking;Thu Mar 08 11:43:34 EET 2018
320;R;3;13510;AYGUN;YIL;RECEPTIONIST;Booking;Thu Mar 08 11:43:34 EET 2018
515;R;5;12345;MUHITTIN;MASRAFLI;RECEPTIONIST;Booking;Thu Mar 08 11:43:34 EET 2018
210;R;2;89456;DONDURDU;GUEST;Booking;Thu Mar 08 11:43:34 EET 2018
320;F;3;13510;AYGUN;YIL;RECEPTIONIST;Check IN;Thu Mar 08 11:43:34 EET 2018
405;R;4;24868;HAKKI;BAFRALI;GUEST;Booking;Thu Mar 08 11:43:34 EET 2018
210;F;2;89456;DONDURDU;RECEPTIONIST;Check IN;Thu Mar 08 11:43:34 EET 2018
515;R;5;12345;MUHITTIN;MASRAFLI;RECEPTIONIST;Cancel Rezervation;Thu Mar 08 11:43:34 EET 2018
320;F;3;13510;AYGUN;YIL;Receptionist;Check OUT;Thu Mar 08 11:43:34 EET 2018
```

A	B	C	D	E	F	G	H	I
ROOM ID	ROOM STATUS	ROOM CAPACITY	GUEST ID	GUEST NAME	GUEST SURNAME	OPERANT PERSON	TURNOVER	DATE
101	R		1	89456	DONDURDU	GUEST	Booking	Thu Mar 08 11:55:45 EET 2018
320	R		3	13510	AYGUN	YIL	RECEPTIONIST	Booking
515	R		5	12345	MUHITTIN	MASRAFLI	RECEPTIONIST	Booking
210	R		2	89456	DONDURDU	GUEST	Booking	Thu Mar 08 11:55:45 EET 2018
320	F		3	13510	AYGUN	YIL	RECEPTIONIST	Check IN
405	R		4	24868	HAKKI	BAFRALI	GUEST	Booking
210	F		2	89456	DONDURDU	RECEPTIONIST	Check IN	Thu Mar 08 11:55:46 EET 2018
515	R		5	12345	MUHITTIN	MASRAFLI	RECEPTIONIST	Cancel Rezervation
320	F		3	13510	AYGUN	YIL	Receptionist	Check OUT

Ayrıca misafirlerin bilgilerinin tutulduğu “Guests.csv” dosyasında ise unic kimlik numaralarına göre misafir bilgileri tutulmaktadır. Misafir tekrardan bir oda rezerve etmek isterse kimlik bilgisini girmesi yeterli olacaktır diğer bilgiler otomatik olarak çekilecektir. “Guest.csv” dosyasının screen shotı aşağıda belirtildiği gibidir.

	A	B	C
1	GUEST ID	GUEST NAME	GUEST SURNAME
2	89456	DONDU	DURDU
3	13510	AYGUN	YIL
4	12345	MUHITTIN	MASRAFLI
5	24868	HAKKI	BAFRALI
6			