

**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**



**BLM3722 YAZILIM MÜHENDİSLİĞİ**

**Grup 2: Yunus Emre Selçuk**

20011012 - Selahattin Emre Kutlu

19011006 - Mevlana Halit Kaya

20011045 - Muhammed Ali Lale

20011044 - Yusuf Enes Kurt

19011503 – Semih Erken

## KABUL VE KISITLAR

### Kabul Edilenler:

Projenin belirlenen zaman çizelgesine uygun bir şekilde tamamlanması gerekmektedir.

Projenin belirlenen bütçe sınırları içinde kalması önemlidir.

Projenin sonuçları ve çıktıları, ilgili paydaşlar tarafından kabul edilmelidir.

Projenin ilgili yasal ve düzenleyici gereksinimlere uygun olması önemlidir.

Proje ekibi, verimli bir şekilde çalışmalı ve işbirliği yapmalıdır.

Proje, kullanıcıların ve müşterilerin beklentilerini karşılamalıdır.

Proje, kalite standartlarına uygun olarak başarıyla tamamlanmalıdır.

Projenin sürdürülebilirlik prensiplerine uygun olması önemlidir.

### Kısıtlar:

Zaman Kısıtı: 16 Mayıs 2023 saat 01.00'e kadar projenin teslim edilmesi gerekmektedir.

Bütçe kısıtı: Kullanılacak programların ve diğer bileşenlerin toplam masrafı 1000₺'yi geçmemelidir.

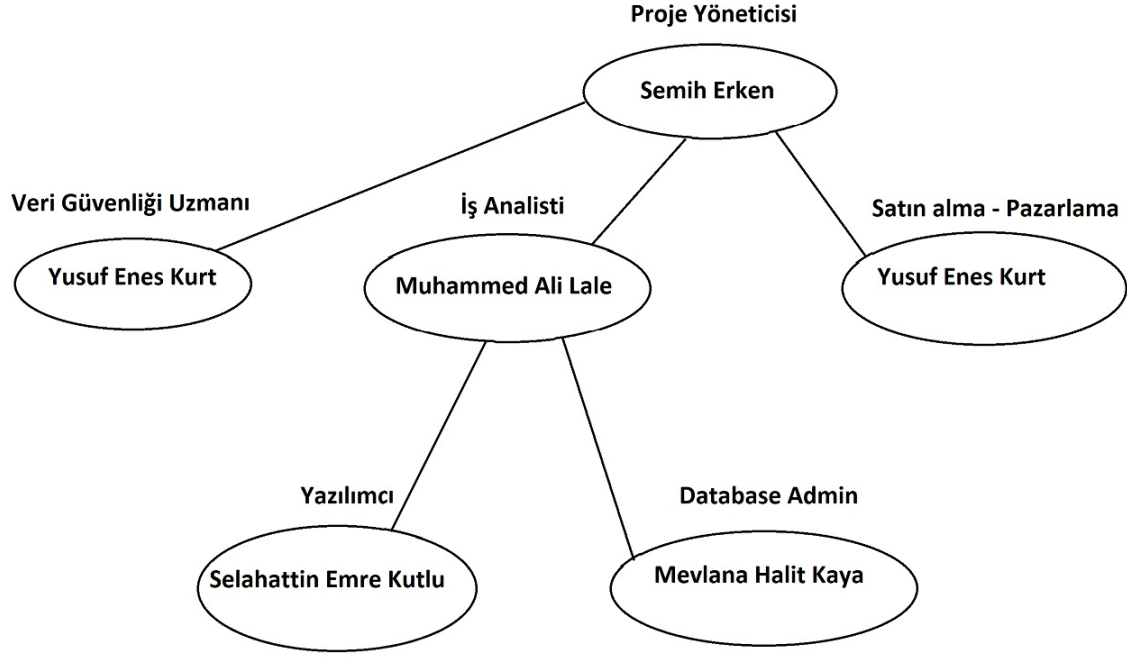
İnsan Kaynağı: 5 adet Jr. Yazılımcıdan teknik anlamda daha bilgili kişiler olmamalıdır.

Teknoloji Kısıtı: Yüksek donanım özelliklerine sahip teknoloji gereksinimlerine ihtiyaç duymamalıdır.

Yasal Kısıt: Program kullanıcılarının sadece yetkili kişilerden oluşması ve kvkk hükümlerine uymaları gerekmektedir.

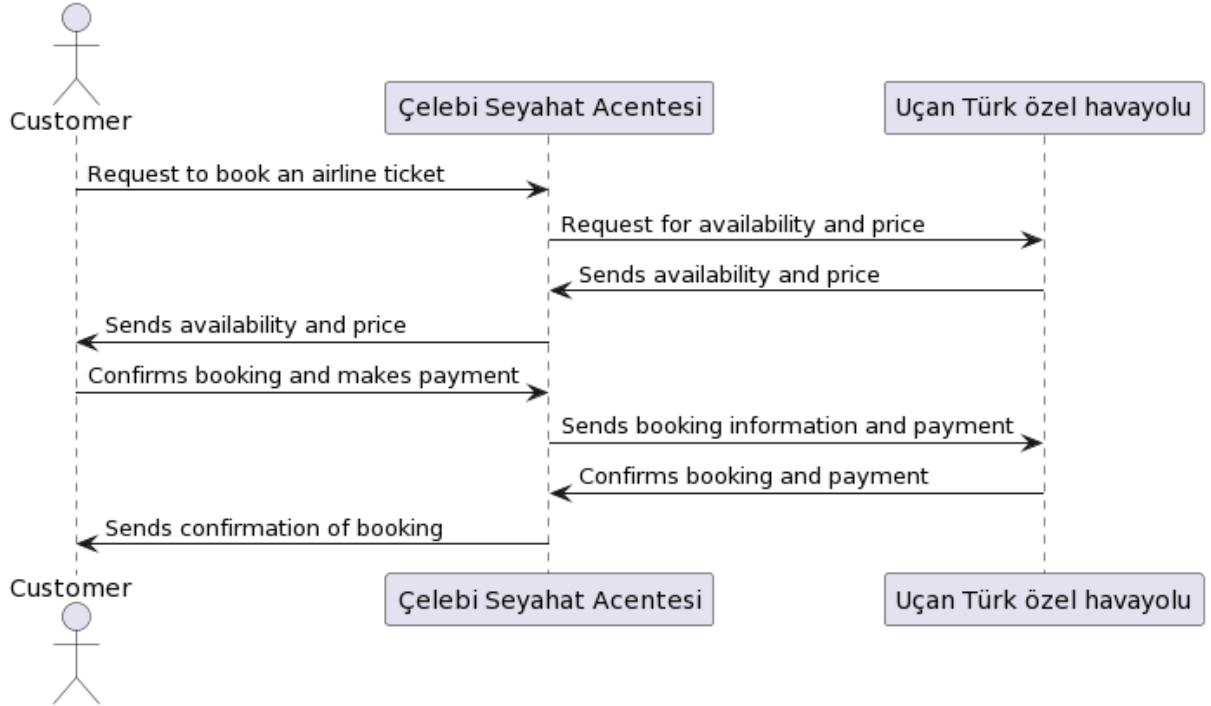
Paydaşlar: Dr. Yunus Emre Selçuk, Akademisyen asistanı, ve 5 adet ytü Öğrencisi yazılımcı.

## PROJE ORGANİZASYON ŞEMASI



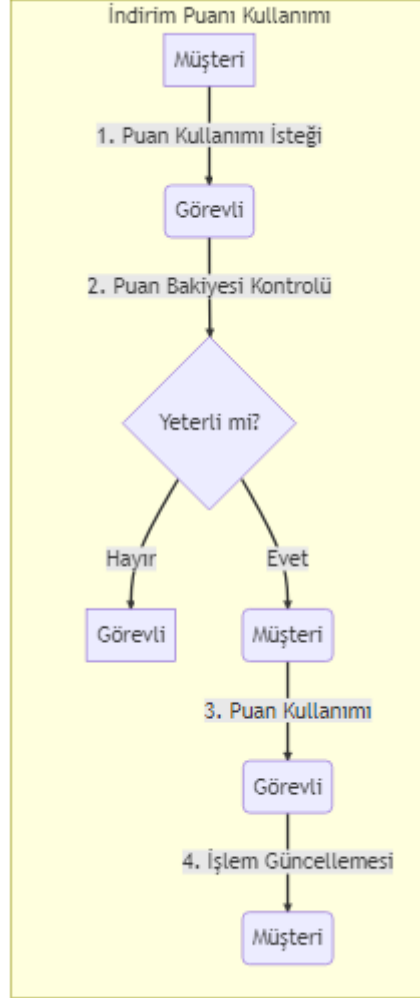
## SIRALAMA DİYAGRAMI

### Booking an Airline Ticket

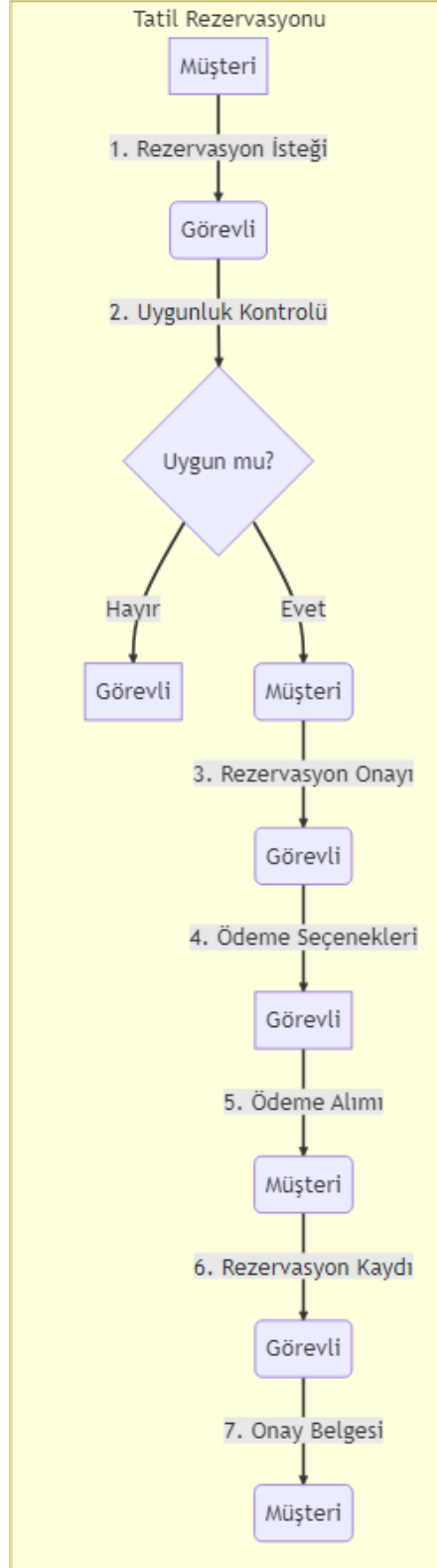


# ETKİNLİK DİYAGRAMI

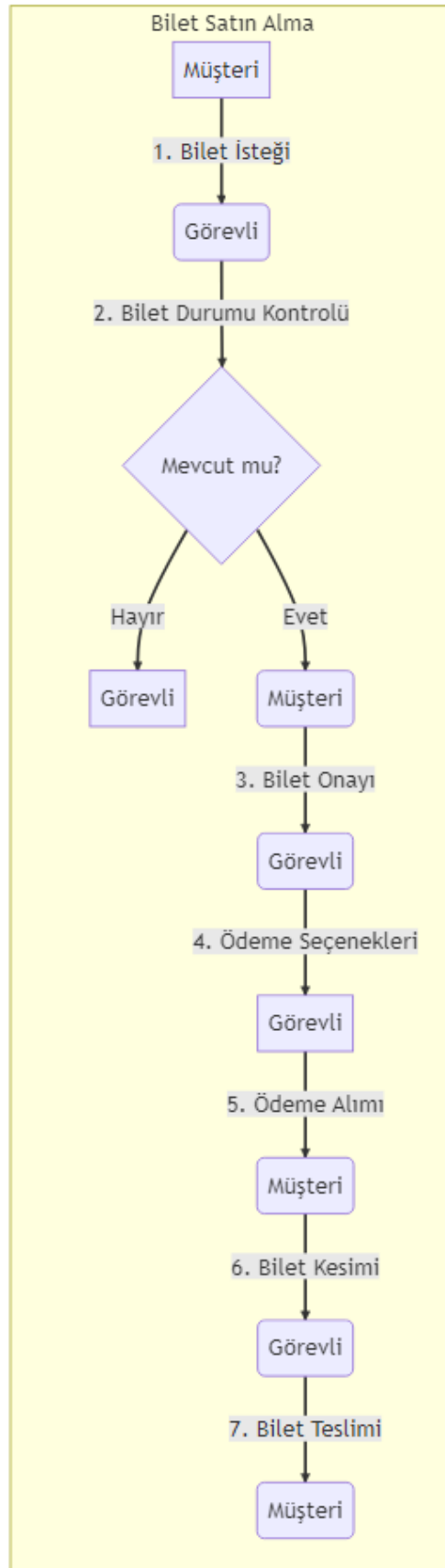
İndirim puanı kullanıldığı senaryoda oluşacak etkinlik diyagramı



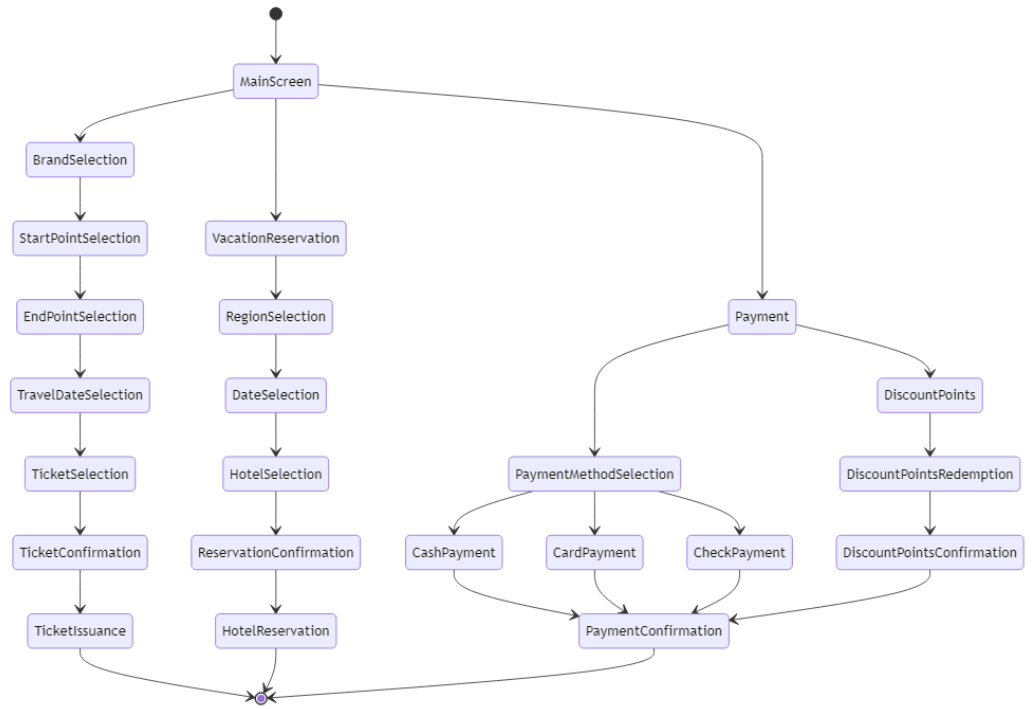
## Tatil rezervasyonu senaryosunun etkinlik diyagramı



## Bilet satın alma senaryosunun etkinlik diyagramı



# DURUM DİYAGRAMI



## RİSK TABLOSU

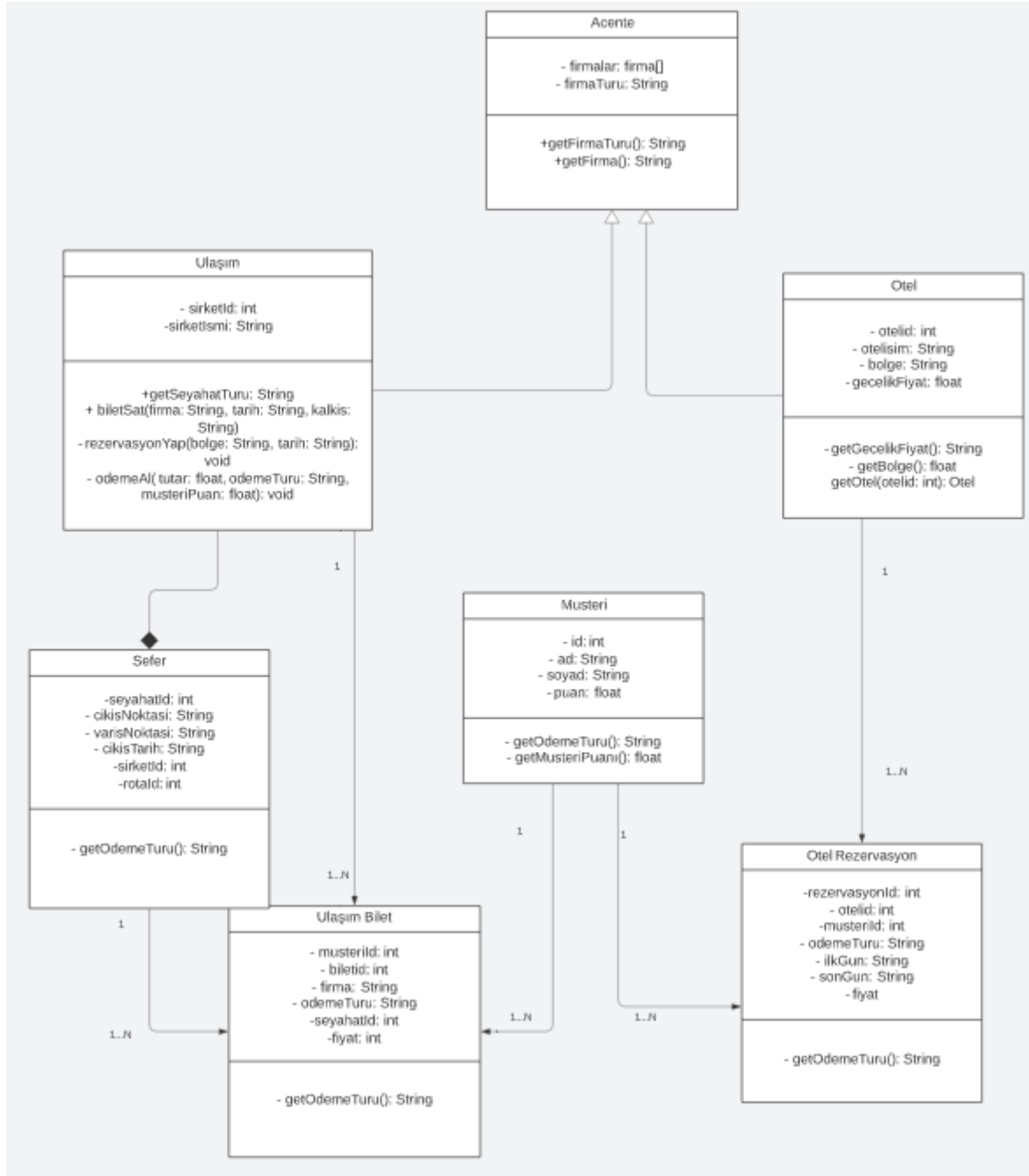
No	Risk Tanımı	Etki	Olasılık	Risk Düzeyi	Risk Önlemi	Sorumlu Kişi	İzleme ve Raporlama
1	Yetersiz Kaynaklar	Projenin gecikmesi	Yüksek	Yüksek	Proje bütçesini ve kaynakları yeniden değerlendirme	Proje Yöneticisi	Proje süresince kaynak kullanımını takip etme
2	Teknolojik Sorunlar	Veri kaybı, sistem çökmesi	Orta	Orta	Düzenli yedeklemeler, güvenlik güncellemeleri	Sistem Uzmanı	Sistem performansını ve güvenliğini izleme
3	İletişim Sorunları	Bilgi kaybı, hatalı anlaşmalar	Orta	Düşük	Düzenli iletişim toplantıları, net iletişim	Proje Yöneticisi	İletişim protokollerini izleme ve değerlendirme
4	Talep Değişiklikleri	Kapsam dışına çıkma, ek süre	Yüksek	Orta	Değişiklik yönetimi süreçleri, müşteri ile iletişim	Proje Yöneticisi	Talep değişikliklerini takip etme
5	Tedarikçi Problemleri	Gecikmeli teslimatlar, kalite sorunları	Orta	Düşük	Alternatif tedarikçi seçenekleri, sözleşme yönetimi	Satınalma Sorumlusu	Tedarikçi performansını izleme
6	Veri Güvenliği İhlali	Veri kaybı, kötü niyetli erişim	Yüksek	Orta	Güçlü şifreleme, erişim kontrolleri	Veri Güvenliği Uzmanı	Güvenlik önlemlerini ve raporları izleme



## GEREKSİNİM İZLENEBİLİRLİK TABLOSU

Gereksinim No	Gereksinim Açıklaması	Gerekçe	Sahibi	Öncelik	Durum
1	Firma ve müşteri verilerinin tutulması	Firma ve müşteri bilgilerinin kaydedilmesi	Veri Güvenliği Uzmanı	Yüksek	Aktif
2	Seferlerin kaydedilmesi ve yönetilmesi	Sefer bilgilerinin takip edilmesi	İş Analisti	Yüksek	Aktif
3	Otellerin bilgilerinin saklanması	Otel bilgilerinin depolanması	Database Admin	Orta	Aktif
4	Ödeme işlemlerinin takibi ve kaydedilmesi	Ödeme işlemlerinin izlenmesi ve kaydedilmesi	Satın Alma-Pazarlama	Yüksek	Aktif
5	Firma ve müşteri puanlama sistemi	Firma ve müşterilerin puanlanması	Proje Yöneticisi	Orta	Aktif

# UML SINIF DİYAGRAMI



# BİRİM SINAMALARI

## Yusuf Enes Kurt

```
@Test // Kullanıcı Puanı güncelleme işleminin başarısının kontrolü
public void testUpdateCustomerPoints() {
    String customerId = "2";
    Float reward = 5.5f;

    assertTrue(op.updateCustomerPoint(customerId, reward));
}
```

```
@Test // Ödeme işleminin başarısının kontrolü
public void testPayment() {
    String choice = "cash";
    Float price = 100f;

    assertTrue(op.updatePayments(choice, price));
}
```

## Mevlana Halit Kaya

```
@Test // Bilet sorgulama işleminde alınan biletin doğruluğunun kontrolü
public void testTicketQuery() {
    String firmName = "THY";
    String place = "Ankara";
    String destination = "Yozgat";
    String date = "2023-5-26";

    Travel travel = op.getTravels(firmName, place, destination, date).get(0);

    assertTrue(travel.getFirmName().matches(firmName));
    assertTrue(travel.getPlace().matches(place));
    assertTrue(travel.getDestination().matches(destination));
}
```

```
@Test // Firmaların puanlarının güncellenmesi işleminin başarısının kontrolü
public void testUpdateFirmPoints() {
    String firmName = "Pegasus";
    Float point = 4.5f;

    assertTrue(op.updateFirmPoints(firmName, point));
}
```

## Muhammed Ali Lale

```
@Test // Otellerin puanlarının güncellenmesi işleminin başarısının kontrolü
public void testUpdateHotelPoints() {
    String firmName = "ALTINKUMSAL";
    Float point = 4.5f;

    assertTrue(op.updateHotelPoints(firmName, point));
}
```

```
@Test // Koltuk Sayısı eksiltme işleminin başarısının kontrolü
public void testDecreaseSeats(){

    Travel travel = new Travel("TCDD", "Ankara", "İzmir", "2023-05-15", "14:30", 200f, 1);

    assertTrue(op.decreaseSeats(travel));
}
```

## Semih Erken

```
@Test // Önceki koltuk sayısının ödeme sonrasındaki koltuk sayısından 1 eksik olduğunun kontrolü
public void testSeats(){

    Travel travel = op.getTravels("TCDD", "Ankara", "İzmir", "2023-05-15").get(0);
    int seatsBefore = travel.getSeats();

    op.decreaseSeats(new Travel("TCDD", "Ankara", "İzmir", "2023-05-15", "14:30", 200f, 1));

    travel = op.getTravels("TCDD", "Ankara", "İzmir", "2023-05-15").get(0);
    int seatsAfter = travel.getSeats();

    assertEquals(seatsBefore-1, seatsAfter);
}
```

```
@Test // Yeni ulaşım firması ekleme işleminin başarısının kontrolü
public void testAddFirm(){
    String firmName = "LÜKS ARTVİN";
    Float point = 1.8f;

    assertTrue(op.addFirm(firmName, point));
}
```

## Selahattin Emre Kutlu

```
@Test // Yeni sefer ekleme işleminin başarısının kontrolü
public void testAddTravel(){

    assertTrue(op.addTravel("LÜKS ARTVİN", "Giresun", "Yozgat", "2023-08-28", "11:30", 420f, 38));

}
```

```
@Test //Firma puanı güncelleme işlemi sonrası firmanın puanının istenen puanda olup olmadığının kontrolü
public void testFirmPoints(){
    String firmName = "Pegasus";
    Float point = 6.4f;

    op.updateFirmPoints(firmName, point);

    assertFalse(!point.equals(op.getPoint(firmName)));
}
```

