

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



ÇELEBİ SEYAHAT ACENTESİ BİLET-TATİL SİSTEMİ

20011610 — Fatih ALTINCI
20011611 — İlker BAHAR
21011607 — Ahmet Kemal YILMAZ
21011613 — Şimal EKİZ

BLM3722 - Yazılım Mühendisliği Projesi

Mayıs, 2023

İÇİNDEKİLER

ŞEKİL LİSTESİ	iii
TABLO LİSTESİ	iv
ÖZET	v
1 Proje Planı	1
1.1 Proje Alan Tanımı	1
1.2 Kabul ve Kısıtlar	1
1.3 İş-Zaman Çizelgesi	2
1.4 Ekip	3
1.4.1 Organizasyon Şeması	3
1.4.2 Görev Dağılımları	3
1.5 Risk Tablosu	3
2 İsteklerin Modellenmesi	4
2.1 Kullanım Senaryosu	4
2.2 İzlenebilirlik Tablosu	7
3 Nesneye Dayalı Modelleme	8
4 Nesneye Dayalı Tasarım	9
5 Testler	12
5.1 Cüzdan Ödeme Testi	13
5.2 Kredi Kartı Ödeme Testi	14
5.3 Çek Bakiye Hesaplama Birim Testi	15
5.4 Tatil Puanı Hesaplama Birim Testi	15
5.5 Tatil Seçim Ekranı Buton Birim Testi	16
5.6 Ulaşım Seçim Ekranı Birim Testi	17

ŞEKİL LİSTESİ

Şekil 1.1	Gantt Diyagramı	2
Şekil 1.2	Organizasyon Şeması	3
Şekil 2.1	Kullanım Senaryosu Diyagramı	4
Şekil 3.1	Sınıf Diyagramı	8
Şekil 4.1	Main Sınıfının Sıralama Diyagramı	9
Şekil 4.2	Ödeme Modülünün Etkinlik Diyagramı	10
Şekil 4.3	Bilet Bilgileri Modülünün Durum Diyagramı	11
Şekil 5.1	Cüzdan Ödeme Birim Testi	13
Şekil 5.2	Cüzdan Ödeme Birim Testi Sonucu	13
Şekil 5.3	Kredi Kartı Ödeme Birim Testi	14
Şekil 5.4	Kredi Kartı Ödeme Birim Testi Sonucu	14
Şekil 5.5	Çek Bakiye Hesaplama Birim Testi	15
Şekil 5.6	Tatil Puanı Hesaplama Birim Testi	15
Şekil 5.7	Tatil Seçim Ekranı Buton Birim Testi	16
Şekil 5.8	Ulaşım Seçim Ekranı Birim Testi - 1	17
Şekil 5.9	Ulaşım Seçim Ekranı Birim Testi - 2	18

TABLO LİSTESİ

Tablo 1.1	Risk Tablosu	3
Tablo 2.1	Bilet Bilgileri Kullanım Senaryosu	4
Tablo 2.2	Mevcudiyet ve Fiyat Bilgileri Kullanım Senaryosu	5
Tablo 2.3	Tatil Bilgileri Kullanım Senaryosu	5
Tablo 2.4	Ücret Ödeme Kullanım Senaryosu	5
Tablo 2.5	Bilgilendirme Kullanım Senaryosu	6
Tablo 2.6	İzlenebilirlik Tablosu	7

Çelebi Seyahat Acentesi Bilet-Tatil Sistemi

Fatih ALTINCI

İlker BAHAR

Ahmet Kemal YILMAZ

Şimal EKİZ

Bilgisayar Mühendisliği Bölümü

BLM3722 - Yazılım Mühendisliği Projesi

Çelebi Seyahat Acentesi, tatil ve ulaşım hizmetlerini etkili bir şekilde yönetmek amacıyla bir bilgi sistemi geliştirmeyi hedeflemektedir. Bu sistem, Uçan Türk özel havayolu, Devlet Demir Yolları (TCDD) ve YTUR otobüs firmalarının biletlerinin satışını ve çeşitli otellerle tatil rezervasyonlarını yönetmeyi sağlayacaktır. Müşteriler, talep ettikleri bilet veya rezervasyon bilgilerini acente görevlisine iletecek ve görevli bu bilgileri sisteme girecektir. Sistem, ilgili firmaların sistemleriyle entegre olarak mevcudiyet ve fiyat bilgilerini alacak ve müşterilerin taleplerine uygun seçenekleri sunacaktır. Ödemeler, nakit, kredi kartı ve çek gibi yöntemlerle kabul edilecek ve müşterilere aldıkları biletler ve tatiller üzerinden puanlar verilecektir. Müşteriler, biriken puanlarını ödemelerde kullanabilecek ve puanlarının tutarında indirim yapılacaktır.

Anahtar Kelimeler: Kullanım senaryosu, nesneye dayalı modelleme, nesneye dayalı tasarım

1

Proje Planı

1.1 Proje Alan Tanımı

Çelebi Seyahat Acentesi'nin bilet satışı ve tatil rezervasyonu hizmetlerini yönetmek amacıyla bir bilgi sistemi tasarlamayı hedeflemektedir. Acentenin sunduğu hizmetler arasında, Uçan Türk özel havayolu, Devlet Demir Yolları (TCDD) ve YTUR otobüs firmalarının biletleri satışı ve çeşitli otellerle anlaşmalar yapılarak tatil rezervasyonu bulunmaktadır. Müşterilerin bilet ve rezervasyon işlemleri sırasında ödeme yapmak için nakit, kredi kartı ve çek kabul edilmektedir. Ayrıca, müşterilere aldıkları biletler ve tatiller için belirli puanlar verilerek, biriken puanlar ödeme yapmak için kullanılabilir.

1.2 Kabul ve Kısıtlar

Projede kabul edilen unsurlar:

Çoklu Hizmetler: Çelebi Seyahat Acentesi'nin bilet satışı ve tatil rezervasyonu hizmetlerinin aynı bilgi sistemi üzerinde yönetilmesi, müşteri deneyimini geliştirebilir.

Çeşitli Ödeme Yöntemleri: Nakit, kredi kartı ve çek gibi farklı ödeme seçenekleri sunmak, müşteri memnuniyetini artırabilir ve farklı müşteri ihtiyaçlarını karşılayabilir.

Puan Sistemi: Müşterilere bilet ve tatiller için puan verilmesi ve bu puanların ödemelerde kullanılabilmesi, müşteri sadakatini artırabilir ve müşteriye ekonomik fayda sağlayabilir.

Genişleyebilirlik: Proje, ileride başka havayolu veya otobüs firmalarının eklenmesiyle genişletilebilir.

Anlaşmalar: Acente, çeşitli otellerle anlaşmalar yaparak, müşterilere farklı tatil seçenekleri sunabilir.

Projede kısıtlar ve dikkat edilmesi gereken unsurlar:

Güvenlik: Müşteri bilgilerinin güvenliği önemlidir ve bu bilgilerin yetkisiz kişilerin eline geçmesini önlemek için gerekli önlemler alınmalıdır.

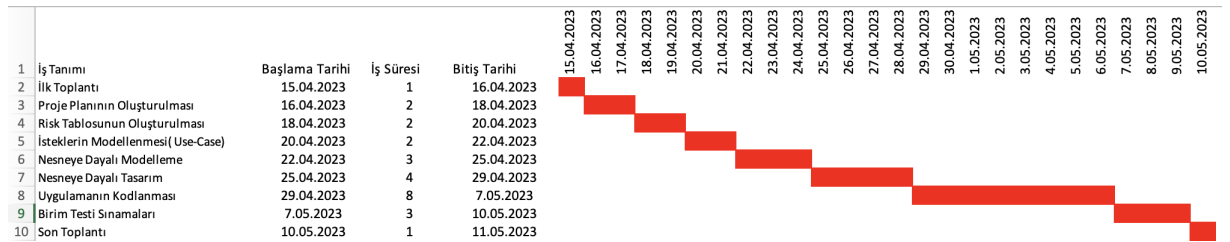
Fiyatlandırma: Farklı firmaların fiyatlandırma politikaları değişebilir ve bu, müşterilere farklı fiyatlar sunulmasına neden olabilir. Bu durumda, acente müşterilere adil bir fiyat sunmak için gereken önlemleri almalıdır.

Bilet ve Otel Stoğu: Müşterilerin talepleri doğrultusunda bilet ve otel stoğunun yönetimi, doğru bir şekilde yapılmalıdır. Aksi takdirde, müşterilerin talepleri karşılanamayabilir.

Uyumluluk: Çeşitli havayolu ve otobüs firmaları ile çalışmak, uyumluluk sorunlarına neden olabilir. Bu nedenle, sistem farklı firmaların sistemleri ile uyumlu olacak şekilde tasarlanmalıdır.

Müşteri Memnuniyeti: Müşterilerin taleplerine hızlı bir şekilde yanıt vermek ve doğru bilgi sağlamak, müşteri memnuniyeti açısından önemlidir. Bu nedenle, müşteri hizmetleri personeli eğitilmeli ve müşteri taleplerine hızlı bir şekilde yanıt verebilecek sistemler oluşturulmalıdır.

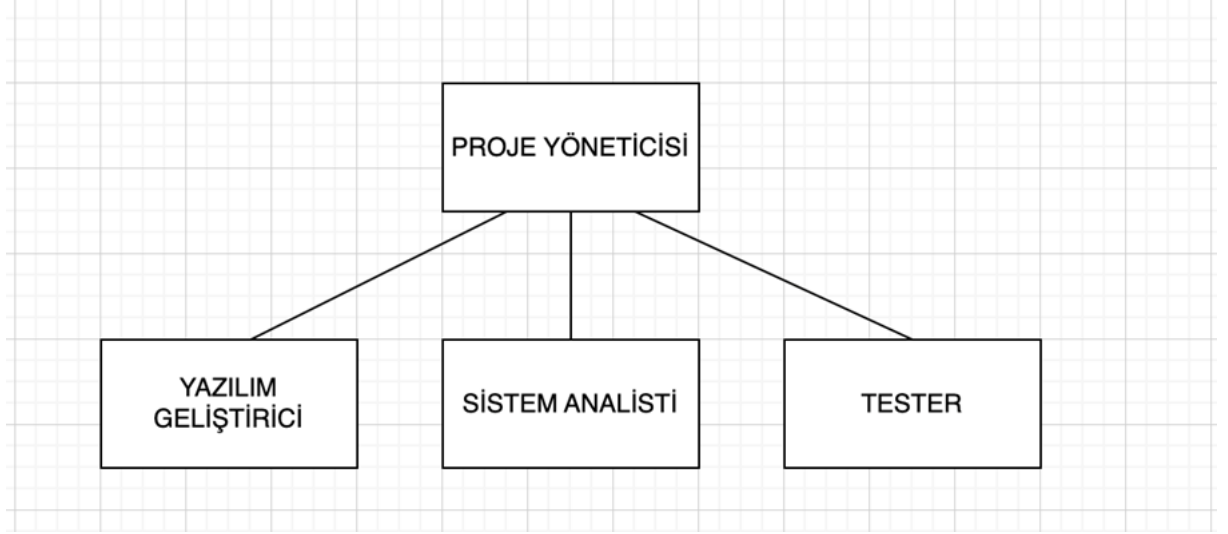
1.3 İş-Zaman Çizelgesi



Şekil 1.1 Gantt Diyagramı

1.4 Ekip

1.4.1 Organizasyon Şeması



Şekil 1.2 Organizasyon Şeması

1.4.2 Görev Dağılımları

Proje Yöneticisi: İlker BAHAR

Yazılım Geliştirici: Fatih ALTINCI

Sistem Analisti ve Tasarımcısı: Şimal EKİZ

Tester: Ahmet Kemal YILMAZ

1.5 Risk Tablosu

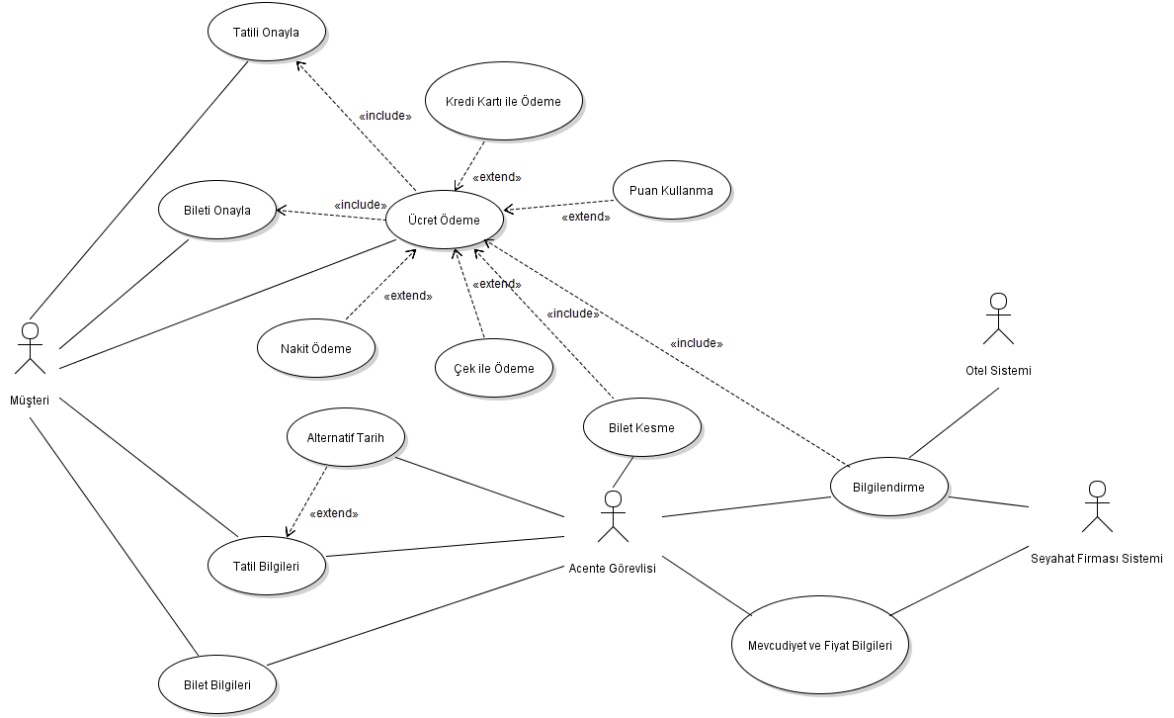
Risk No	Risk Açıklaması	Etki	Olasılık	Önlem
1	Yazılım kütüphanelerinin güncel olmaması	Yazılım geliştirme sürecinde gecikmelerin yaşanmasına sebep olması	Düşük	Projedeki tüm kütüphanelerin güncelliğinin kontrol edilmesi.
2	Kullanıcılar için puan sisteminin çalışmaması	Kullanıcıların yaptıkları ödemeler sonrası puan alamamaları.	Düşük	Uygulamadaki modülün düzgün bir şekilde veri tabanı ile uyumlu çalışacak hale getirilmesi.
3	Test aşamasında problemler yaşanması	Proje gereksinimlerini gerçeklemede sorunlarla karşılaşılması	Orta	Java için uygun kütüphanelerin kullanılması
4	Projedeki istenilen gereksinimlerin gerçekleştirilememesi	Proje için gerekli olan tüm gereksinimlerin karşılanamaması	Düşük	Proje planındaki görev dağılımının uygun yapılması

Tablo 1.1 Risk Tablosu

2 İsteklerin Modellenmesi

Kullanım senaryosu çizilirken Violet UML CASE aracı kullanılmıştır.

2.1 Kullanım Senaryosu



Şekil 2.1 Kullanım Senaryosu Diyagramı

Kullanım Senaryosu	Bilet Bilgileri
Birincil Aktör	Müşteri
İlgililer ve Beklentileri	Müşteri: Tercih ettiği firmanın, belirttiği tarihlerdeki istediği çıkış-varış noktasına dair bilet uygunluğunu ve fiyatını öğrenmek ister. Acente Görevlisi: Müşterinin istediği biletin mevcudiyetini ve fiyatını kontrol etmek ister.
Ön Koşullar	Acente Görevlisi'nin Müşteri'ye geri dönüş sağlaması için seyahat firmasından mevcudiyet ve fiyat bilgilerini kontrol eder.
Son Koşullar	Bilet bilgileri Müşteri'nin onayına hazır hale gelir.
Ana Senaryo	1.Müşteri istediği bilet bilgilerini Acente Görevlisi'ne söyler. 2.Acente Görevlisi biletin uygunluğunu Seyahat Firması Sistemi üzerinden kontrol eder. 3.Müşteri bileti onaylar.
Alternatif Akış	

Tablo 2.1 Bilet Bilgileri Kullanım Senaryosu

Kullanım Senaryosu	Mevcudiyet ve Fiyat Bilgileri
Birincil Aktör	Acente Görevlisi
İlgililer ve Beklentileri	Acente Görevlisi: Müşterinin istediği biletin mevcudiyetini ve fiyatını kontrol etmek ister. Seyahat Firması Sistemi: Acente Görevlisi'nin belirttiği bilet bilgilerinin uygunluğunu bildirmek ister.
Ön Koşullar	Müşteri bilet bilgilerini Acente Görevlisi'ne söylemiştir.
Son Koşullar	Acente Görevlisi biletin uygun olup olmadığını öğrenmiştir.
Ana Senaryo	1. Acente Görevlisi müşterinin istediği biletin uygunluğunu Seyahat Firması Sistemin'e aktarır. 2.Sistem biletin mevcudiyet ve fiyat bilgilerini Acente Görevlisine bildirir. 3.Acente Görevlisi Müşteri'ye biletin mevcudiyet ve fiyat bilgilerini aktarır.
Alternatif Akış	

Tablo 2.2 Mevcudiyet ve Fiyat Bilgileri Kullanım Senaryosu

Kullanım Senaryosu	Tatil Bilgileri
Birincil Aktör	Müşteri
İlgililer ve Beklentileri	Müşteri: Gitmek istediği bölgede ve belirttiği tarihlerde otellerin uygunluğunu öğrenmek ister. Acente Görevlisi: Müşterinin istediği bölgede ve belirttiği tarihlerde uygun olan otelleri müşteriye sunmak ister.
Ön Koşullar	Acente Görevlisi'nin Müşteri'ye geri dönüş sağlaması için acente sistemi üzerinden tarihleri kontrol etmesi gerekir.
Son Koşullar	Rezervasyon bilgileri Müşteri'nin onayına hazır hale gelir.
Ana Senaryo	1.Müşteri istediği rezervasyon bilgilerini Acente Görevlisi'ne söyler. 2.Acente Görevlisi biletin uygunluğunu acente firmasının sistemi üzerinden kontrol eder. 3.Müşteri rezervasyonu onaylar.
Alternatif Akış	2.a.Müşterinin istediği tarihlerde otel uygun değildir. 1.Acente Görevlisi Müşteri'ye alternatif tarihler sunar.

Tablo 2.3 Tatil Bilgileri Kullanım Senaryosu

Kullanım Senaryosu	Ücret Ödeme
Birincil Aktör	Müşteri
İlgililer ve Beklentileri	Müşteri: Ödemesini yapıp, seyahatine/tatiline hazırlanmak ister. Acente Görevlisi: Müşterinin ödemesini alıp, sisteme işlemek ve ardından Otel Sistemi/Seyahat Firması Sistemi'ne bildirmek ister.
Ön Koşullar	1. Müşteri acenteye seyahat için görüşüyse: Müşteri ve Acente Görevlisi bilet tarihini kararlaştırmıştır. Müşteri alacağı bileti onaylamıştır. 2. Müşteri acenteye tatil için görüşüyse: Müşteri ve Acente Görevlisi rezervasyon tarihini kararlaştırmıştır. Müşteri tatili onaylamıştır.
Son Koşullar	Ödemenin başarıyla alınıp fatura hazırlanması gerekir. Müşteri seyahat için görüşüyse Acente Görevlisi bilet kesmiştir. Otel Sistemi/Seyahat Firması Sistemi bilgilendirilmiştir.
Ana Senaryo	1.Müşteri bileti/tatili onaylar. 2.Müşteri ödemeyi yapar. 3.Acente Görevlisi bilet keser. 4.Acente Görevlisi Otel Sistemi'ni/Seyahat Firması Sistemi'ni bilgilendirir.
Alternatif Akış	2.a.Müşteri ödeme işlemini nakit olarak gerçekleştirmek ister. 1.Ücret müşteriden nakit olarak alınır. 2.b.Müşteri ödeme işlemini kredi kartı ile gerçekleştirmek ister. 1. Ücret müşteriden kredi kartı ile alınır. 2.c.Müşteri ödeme işlemini çek ile gerçekleştirmek ister. 1. Ücret müşteriden çek ile alınır. 2.d.Müşteri ödeme işlemini puan kullanarak gerçekleştirmek ister. 1. Müşterinin biriken puanları ödenecek tutar üzerinden düşülür. 2. Tutarın geri kalanı müşterinin istediği diğer ödeme yöntemi ile gerçekleştirilir. 3. Müşterinin biriken puanı, kullandığı puan kadar eksiltilir.

Tablo 2.4 Ücret Ödeme Kullanım Senaryosu

Kullanım Senaryosu	Bilgilendirme
Birincil Aktör	Acente Görevlisi
İlgililer ve Beklentileri	Acente Görevlisi: Müşterinin onaylamış olduğu bilgileri Otel Sistemi'ne/Seyahat Firması Sistemi'ne aktarmak ister. Otel Sistemi: Rezervasyon tarihlerini öğrenmek ister. Seyahat Firması Sistemi: Satıldığı biletin bilgilerini öğrenmek ister.
Ön Koşullar	Müşteri bileti/tatili onaylamıştır. Müşteri ödemeyi yapmıştır. Müşteri seyahat için görüşmeye geldiyse Acente Görevlisi bileti kesmiştir.
Son Koşullar	Alınan bilet Seyahat Firması Sistemi'ne girilmiştir. Rezervasyon tarihleri Otel Sistemi'ne girilmiştir.
Ana Senaryo	1.Müşteri ödemeyi yapar. 2.Acente Görevlisi bileti keser. 3.Acente Görevlisi bilgilendirme işlemini gerçekleştirir.
Alternatif Akış	3.a.Müşteri tatil için görüşmeye gelmiştir. 1.Acente Görevlisi bilgilendirmeyi Otel Sistemi'ne yapar. 3.b.Müşteri seyahat için görüşmeye gelmiştir. 1. Acente Görevlisi bilgilendirmeyi Seyahat Firması Sistemi'ne yapar.

Tablo 2.5 Bilgilendirme Kullanım Senaryosu

2.2 İzlenebilirlik Tablosu

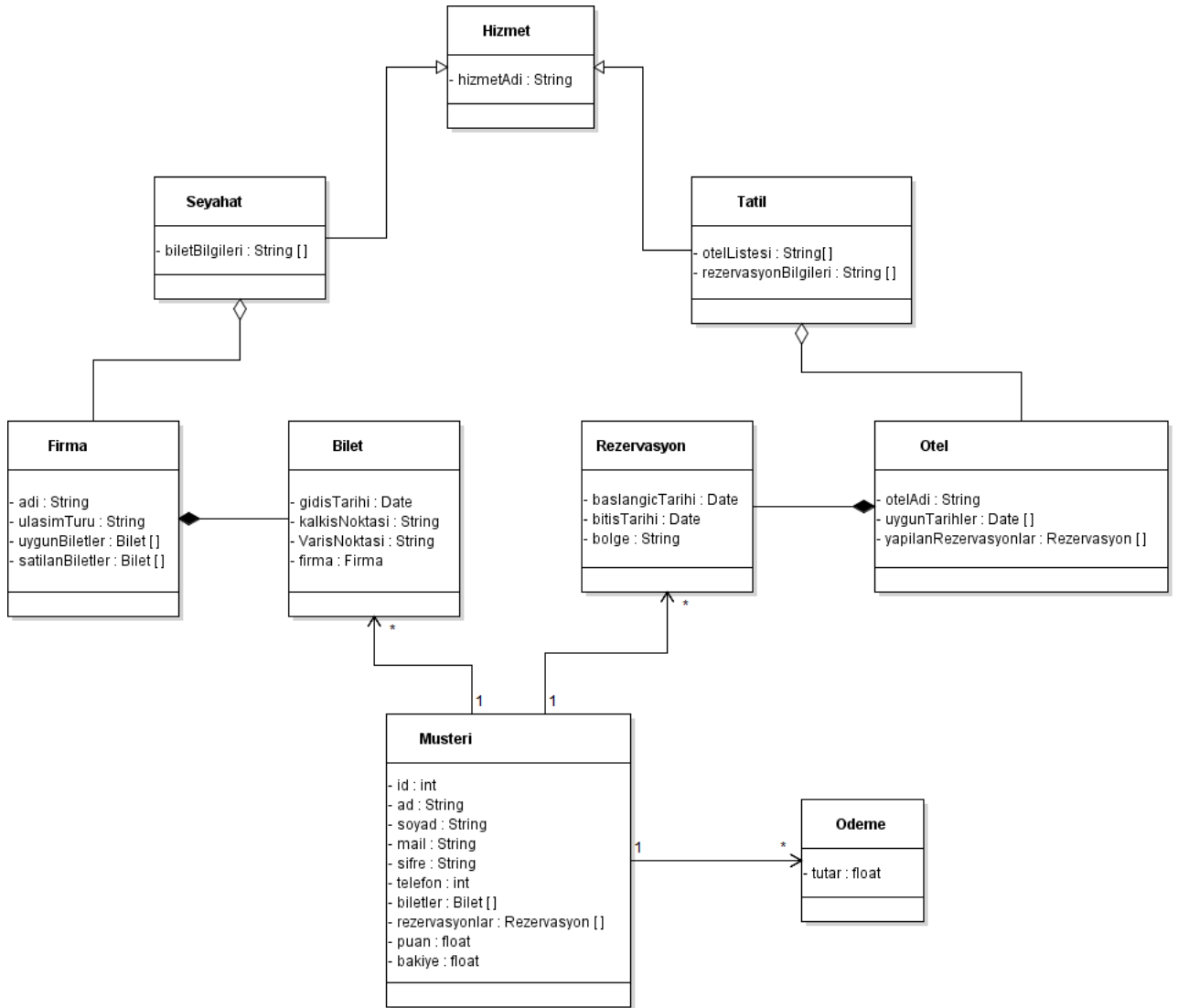
Sınıf Adı	Değişiklik Tarihi	Değişiklik Açıklaması	Değişikliği Yapan Kişi
Main	29.04.2023	Main sınıfında giriş ekranı oluşturuldu.	Fatih
Main	29.04.2023	Giriş ekranındaki diğer sınıfların kodlamaları yapıldı.	Fatih
Ulaşım Seçim Ekranı	30.04.2023	Ekran tasarımı oluşturuldu.	Fatih
Ulaşım Seçim Ekranı	30.04.2023	Müşteriden istenilecek seyahat bilgilerinin girdileri oluşturuldu.	Fatih
Bilet Seçim Ekranı	1.05.2023	Ekran tasarımı oluşturuldu.	Fatih
Bilet Seçim Ekranı	1.05.2023	Girilen bilgilerdeki filtreleme vb. yöntemler ile veri tabanı bağlantısı sağlandı.	Fatih
Bilet Seçim Ekranı	1.05.2023	Veri tabanıyla gerekli kontroller sağlandı.	Fatih
Bilet Ödeme Sayfası	2.05.2023	Ekran tasarımı oluşturuldu.	Fatih
Bilet Ödeme Sayfası	2.05.2023	Müşterinin kullanabileceği ödeme yöntemleri için girdi alınması ve ödeme işlemlerinin gerçekleştirilmesi	Fatih
Bilet	2.05.2023	Bilet sınıfı, nesnenin neler içereceği, yapıcı metodu, getter-setter'lar vb. oluşturuldu.	Fatih
Tatil Seçim Ekranı	3.05.2023	Ekran tasarımı oluşturuldu.	Fatih
Tatil Seçim Ekranı	3.05.2023	Müşteriden istenecek otel bilgilerinin girdileri kodlandı.	Fatih
Tatil Rezervasyon Ekranı	4.05.2023	Ekran tasarımı oluşturuldu.	Fatih
Tatil Rezervasyon Ekranı	4.05.2023	Girilen bilgilerdeki filtreleme ile veri tabanı bağlantısı sağlandı.	Fatih
Tatil Rezervasyon Ekranı	4.05.2023	Veri tabanıyla gerekli kontroller sağlandı.	Fatih
Tatil Ödeme Sayfası	4.05.2023	Ekran tasarımı oluşturuldu.	Fatih
Tatil Ödeme Sayfası	5.05.2023	Müşterinin kullanabileceği ödeme yöntemleri için girdi alınması ve ödeme işlemlerinin gerçekleştirilmesi.	Fatih
Otel	5.05.2023	Otel sınıfı, nesnenin neler içereceği, yapıcı metodu, getter-setter'lar vb. oluşturuldu.	Fatih
Çek	5.05.2023	Çek bozdurma yöntemiyle ödeme yapıldığı ekranın tablo şeklinde gösterilmesi için sınıf ve sınıfın neler içereceği oluşturuldu.	Fatih
Bakiye İşlemleri	5.05.2023	Ekran tasarımı oluşturuldu.	Fatih
Bakiye İşlemleri	6.05.2023	Hangi ödeme yöntemleri ile cüzdana para eklenebileceğine dair girdiler oluşturuldu.	Fatih
Bakiye İşlemleri	6.05.2023	Mevcut cüzdan bakiyesi ve mevcut puan gösterimleri için düzenlemeler yapıldı.	Fatih
Bakiye İşlemleri, Bilet Ödeme Sayfası, Tatil Ödeme Sayfası	6.05.2023	Puan kazanmaya uygun girdilerin puan hesaplaması ve bakiye işlemlerinde gösterilmesi sağlandı.	Fatih
Test Cüzdan Ödeme Birim Testi	7.05.2023	Cüzdanla ödemenin birim testi yapıldı.	Fatih
Test Kredi Kartı Ödeme Birim Testi	7.05.2023	Kredi Kartıyla ödemenin birim testi yapıldı.	Fatih
Çek Bakiye Hesaplama Birim Testi	7.05.2023	Çek Bozdurma işleminden sonraki bakiyenin birim testi yapıldı.	Şimal
Tatil Puanı Hesaplama Birim Testi	7.05.2023	Firmaların belirlediği oranda otellerden puan kazanma işleminin birim testi yapıldı.	Şimal

Tablo 2.6 İzlenebilirlik Tablosu

3

Nesneye Dayalı Modelleme

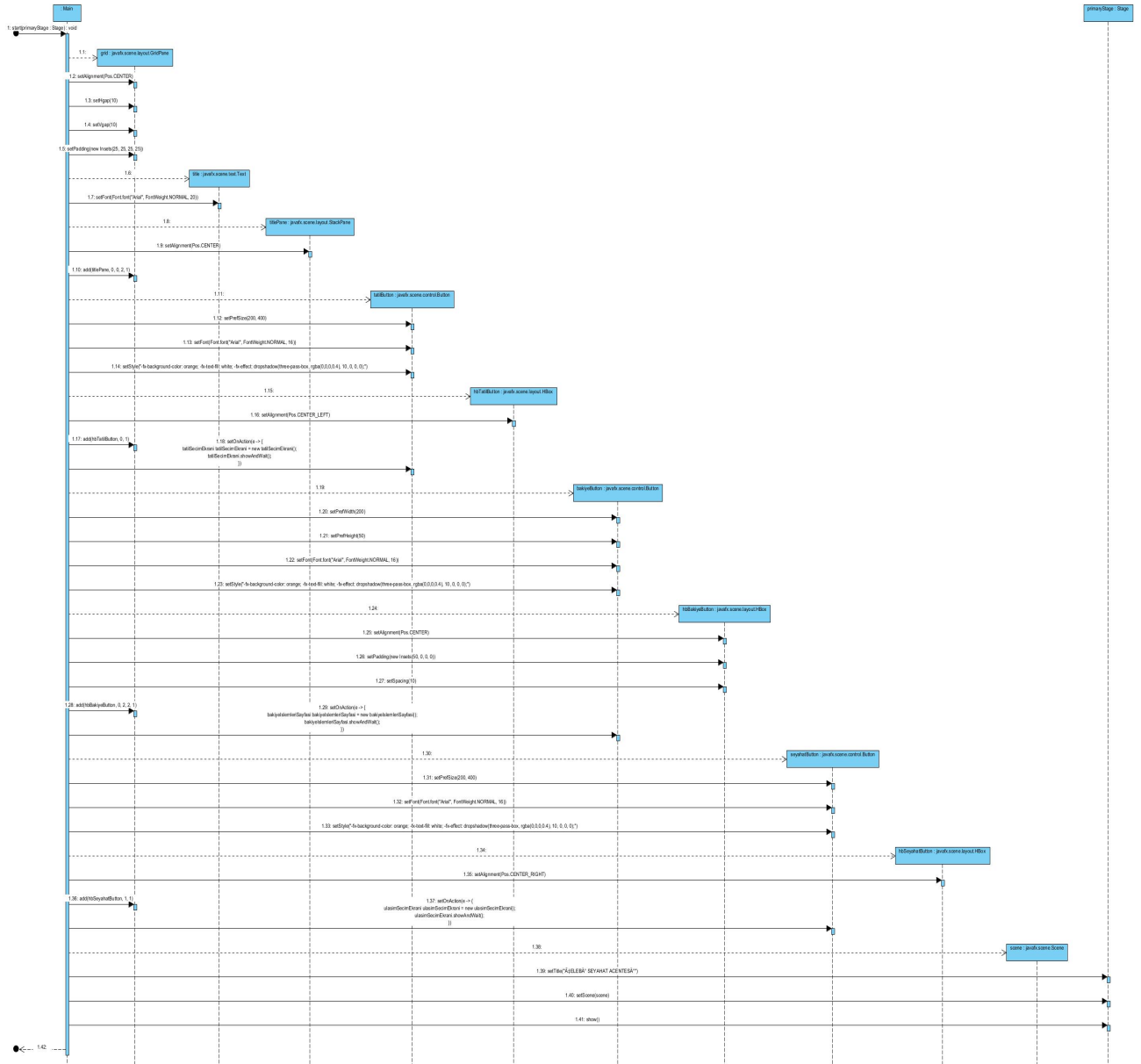
Uygulama alanında sınıf diyagramı çizilirken Violet UML CASE aracı kullanılmıştır.



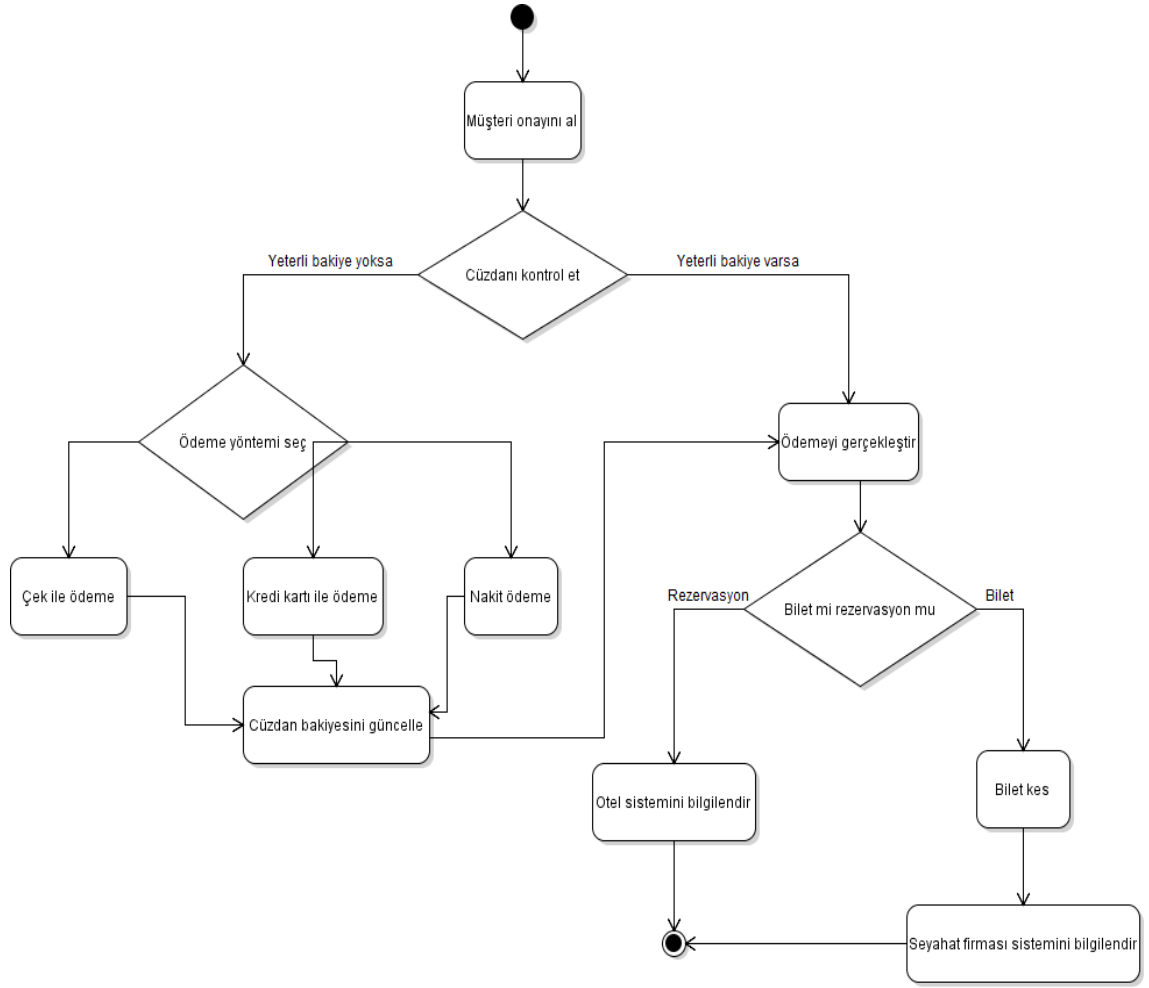
Şekil 3.1 Sınıf Diyagramı

4

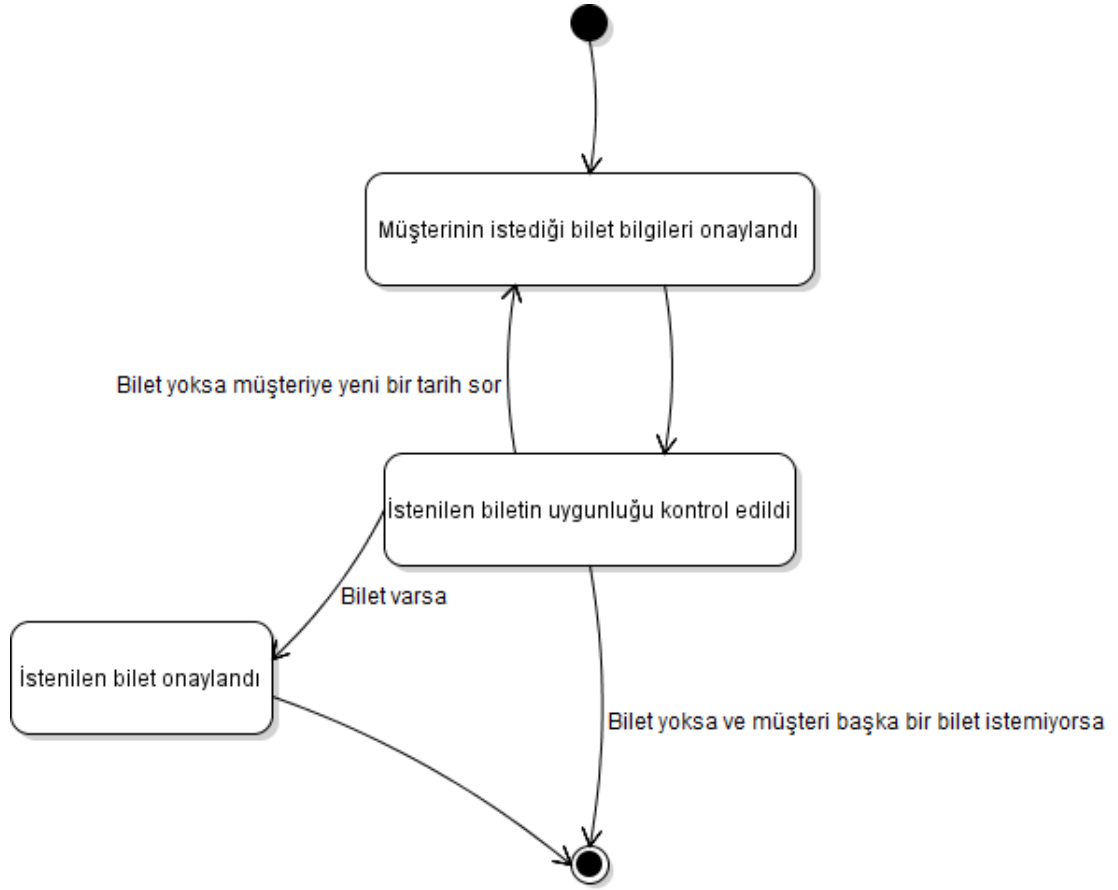
Nesneye Dayalı Tasarım



Şekil 4.1 Main Sınıfının Sıralama Diyagramı



Şekil 4.2 Ödeme Modülünün Etkinlik Diyagramı



Şekil 4.3 Bilet Bilgileri Modülünün Durum Diyagramı

5 Testler

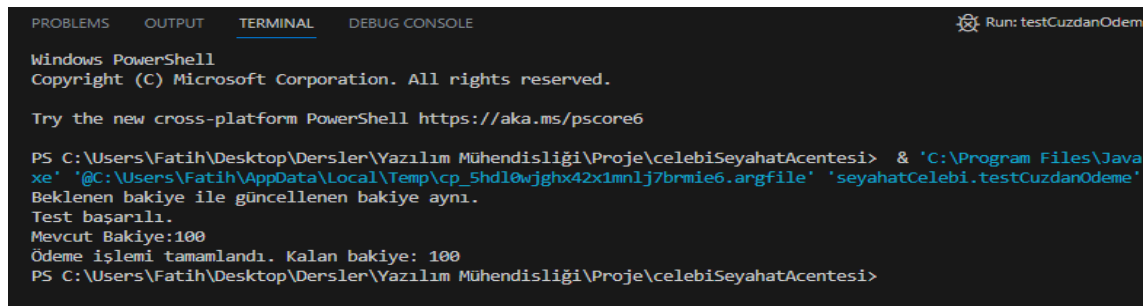
Projeyi sınamak adına Cüzdan Ödeme, Kredi Kartı Ödeme, Çek Bakiye Hesaplama, Tatil Puanı Hesaplama, Tatil Seçim Ekranı ve Ulaşım Seçim Ekranı birim testleri tasarlanmıştır.

Fatih ALTINCI: Cüzdan Ödeme ve Kredi Kartı Ödeme birim testlerini tasarlamıştır. **Şimal EKİZ:** Çek Bakiye Hesaplama ve Tatil Puanı Hesaplama birim testlerini tasarlamıştır. **İlker BAHAR:** Tatil Seçim Ekranı ve Ulaşım Seçim Ekranı birim testlerini tasarlamıştır.

5.1 Cüzdan Ödeme Testi

```
1 package seyahatCelebi;
2 import java.sql.*;
3 public class testCuzdanOdeme {
4     public static void main(String[] args) {
5         testCuzdanlaOdeme();
6     }
7     public static void testCuzdanlaOdeme() {
8         try {
9             // Örnek veritabanı bağlantısı oluşturma
10            Connection conn = DriverManager.getConnection("jdbc:sqlite:celebiSeyahat/src/seyahatCelebi/db/bakiye.db");
11            int otelFiyati = 100; // Test için örnek otel adı ve fiyatı
12            int cuzdanBakiyesi = 200; // Örnek cüzdan bakiyesi
13            int beklenenBakiye = 100; // Beklenen değer
14            // Test öncesinde cüzdan bakiyesini veritabanına ekleyin
15            String insertQuery = "INSERT INTO bakiye (miktar) VALUES (?)";
16            PreparedStatement insertStatement = conn.prepareStatement(insertQuery);
17            insertStatement.setInt(1, cuzdanBakiyesi);
18            insertStatement.executeUpdate();
19            // Ödeme işlemi için test edilecek kod parçası
20            if (otelFiyati <= cuzdanBakiyesi) {
21                cuzdanBakiyesi -= otelFiyati;
22                // Cüzdan bakiyesini güncelle
23                String updateQuery = "UPDATE bakiye SET miktar = ?";
24                PreparedStatement updateStatement = conn.prepareStatement(updateQuery);
25                updateStatement.setInt(1, cuzdanBakiyesi);
26                updateStatement.executeUpdate();
27                if (beklenenBakiye == cuzdanBakiyesi) {
28                    System.out.println("Beklenen bakiye ile güncellenen bakiye aynı.");
29                    // Test başarılı, beklenen sonuçlar elde edildi
30                    System.out.println("Test başarılı.");
31                } else {
32                    System.out.println("Beklenen bakiye ile güncellenen bakiye farklı.");
33                    // Test başarısız, beklenen sonuçlar elde edilemedi
34                    System.out.println("Test başarısız.");
35                }
36                // Ödeme işlemi başarılı olduğunda yapılacak işlemler
37                System.out.println("Mevcut Bakiye: " + cuzdanBakiyesi);
38                System.out.println("Ödeme işlemi tamamlandı. Kalan bakiye: " + cuzdanBakiyesi);
39                // Puan hesaplama ve güncelleme işlemlerini de buraya ekleyebilirsiniz
40            } else {
41                // Yeterli bakiye yok uyarısı ver
42                System.out.println("Yeterli bakiye yok.");
43                // Test başarısız, beklenen sonuçlar elde edilemedi
44                System.out.println("Test başarısız.");
45            }
46            conn.close(); // Veritabanı bağlantısını kapat
47        } catch (SQLException e) {
48            e.printStackTrace();
49            // Test başarısız, bir hata oluştu
50            System.out.println("Test başarısız.");
51        }
52    }
53 }
```

Şekil 5.1 Cüzdan Ödeme Birim Testi



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE Run: testCuzdanOdeme

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

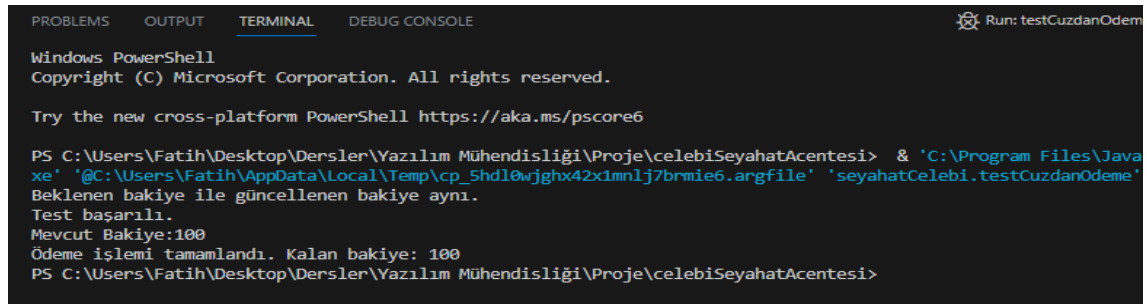
PS C:\Users\Fatih\Desktop\Dersler\Yazılım Mühendisliği\Proje\celebiSeyahatAcentesi> & 'C:\Program Files\Java\
xe' '@C:\Users\Fatih\AppData\Local\Temp\cp_5hd10wjghx42x1mnlj7brmie6.argfile' 'seyahatCelebi.testCuzdanOdeme'
Beklenen bakiye ile güncellenen bakiye aynı.
Test başarılı.
Mevcut Bakiye:100
Ödeme işlemi tamamlandı. Kalan bakiye: 100
PS C:\Users\Fatih\Desktop\Dersler\Yazılım Mühendisliği\Proje\celebiSeyahatAcentesi>
```

Şekil 5.2 Cüzdan Ödeme Birim Testi Sonucu

5.2 Kredi Kartı Ödeme Testi

```
1 package seyahatCelebi;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 public class testKrediKartiOdeme {
8     public static void main(String[] args) {
9         testKrediKartiylaOdeme();
10    }
11    public static void testKrediKartiylaOdeme() {
12        // Test için gerekli başlangıç değerleri oluşturulur
13        String paymentMethod = "Kredi Kartı";
14        String cardNumber = "1234567890123456";
15        String expiryDate = "12/24";
16        String cvv = "123";
17        String otelAdi = "Örnek Otel";
18        double otelFiyati = 300.0;
19        double beklenenOtelFiyati = 300.0;
20        // Kredi kartı bilgileri girilir
21        if (paymentMethod.equals("Kredi Kartı") && cardNumber.matches("\\d{16}")
22            && expiryDate.matches("\\d{2}/\\d{2}") && cvv.matches("\\d{3}")) {
23            // Ödeme tamamlandı, otel fiyatı alındı
24            System.out.println("Ödeme tamamlandı.");
25            try {
26                // Veritabanı bağlantısı oluşturma
27                Connection conn = DriverManager.getConnection("jdbc:sqlite:celebiSeyahat/src/seyahatCelebi/db/otel.db");
28                // Otel adına göre fiyat sorgulama
29                String query = "SELECT fiyat FROM otel WHERE otelAdi = ?";
30                PreparedStatement ps = conn.prepareStatement(query);
31                ps.setString(1, otelAdi);
32                ResultSet rs = ps.executeQuery();
33                // Bilet fiyatını al
34                if (rs.next()) {
35                    otelFiyati = rs.getDouble("fiyat");
36                }
37                // Bağlantıyı kapat
38                rs.close();
39                ps.close();
40                conn.close();
41            } catch (SQLException e) {
42                e.printStackTrace();
43            }
44            // Beklenen otel fiyatıyla gerçek otel fiyatı karşılaştırılır
45            if (otelFiyati == beklenenOtelFiyati) {
46                System.out.println("Test Başarılı"); // Test başarılı
47            } else {
48                System.out.println("Test Başarısız"); // Test başarısız
49            }
50        } else {
51            System.out.println("Test Başarısız"); // Test başarısız
52        }
53    }
54 }
```

Şekil 5.3 Kredi Kartı Ödeme Birim Testi



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE Run: testCuzdanOdeme

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Fatih\Desktop\Dersler\Yazılım Mühendisliği\Proje\celebiSeyahatAcentesi> & 'C:\Program Files\Java
xe' '@C:\Users\Fatih\AppData\Local\Temp\cp_5hd10wjghx42x1mnlj7brmie6.argfile' 'seyahatCelebi.testCuzdanOdeme'
Beklenen bakiye ile güncellenen bakiye aynı.
Test başarılı.
Mevcut Bakiye:100
Ödeme işlemi tamamlandı. Kalan bakiye: 100
PS C:\Users\Fatih\Desktop\Dersler\Yazılım Mühendisliği\Proje\celebiSeyahatAcentesi>
```

Şekil 5.4 Kredi Kartı Ödeme Birim Testi Sonucu

5.3 Çek Bakiye Hesaplama Birim Testi

```
1 public class testCekBakiyeHesaplayici{
2     @Test
3     public void toplama_dogru(){
4         var bakiye = 182.5;
5         var cek = 200;
6         double beklenenToplam = bakiye + cek;
7         var cekBakiye = new cekBakiyeHesaplayici();
8         Assert.AreEqual(cekBakiye.hesapla(bakiye, cek), beklenenToplam);
9     }
10 }
```

Şekil 5.5 Çek Bakiye Hesaplama Birim Testi

5.4 Tatil Puanı Hesaplama Birim Testi

```
1 public class testTatilPuaniHesaplayici{
2     @Test
3     public void carpmaDogru(){
4         var gunlukUcret = 300;
5         var gunSayisi = 15;
6         var oran = 2;
7         double beklenenCarpim = gunlukUcret * gunSayisi * oran;
8         var puanHesabi = new tatilPuaniHesaplayici();
9         Assert.AreEqual(puanHesabi.hesapla(gunlukUcret, gunSayisi, oran), beklenenCarpim);
10     }
11 }
```

Şekil 5.6 Tatil Puanı Hesaplama Birim Testi

5.5 Tatil Seçim Ekranı Buton Birim Testi

```
30 @Test
31 public void testOnayButtonAction() {
32     tatilSecimEkranı tatilSecimEkranı = new tatilSecimEkranı();
33
34     ComboBox<String> konumComboBox = (ComboBox<String>) tatilSecimEkranı.getScene().lookup("#konumComboBox");
35     DatePicker baslangicPicker = (DatePicker) tatilSecimEkranı.getScene().lookup("#baslangicPicker");
36     DatePicker bitisPicker = (DatePicker) tatilSecimEkranı.getScene().lookup("#bitisPicker");
37
38     String konum = "İstanbul";
39     LocalDate baslangicTarihi = LocalDate.now();
40     LocalDate bitisTarihi = baslangicTarihi.plusDays(5);
41     Platform.runLater(() -> {
42         konumComboBox.setValue(konum);
43         baslangicPicker.setValue(baslangicTarihi);
44         bitisPicker.setValue(bitisTarihi);
45     });
46     Button onayButton = (Button) tatilSecimEkranı.getScene().lookup("#onayButton");
47     Platform.runLater(onayButton::fire);
48
49     Label konumLabel = (Label) tatilSecimEkranı.getScene().lookup("#konumLabel");
50     Label tarihLabel = (Label) tatilSecimEkranı.getScene().lookup("#tarihLabel");
51
52     Assertions.assertEquals("Konum: " + konum, konumLabel.getText());
53     Assertions.assertEquals("Tarih Aralığı: " + baslangicTarihi.toString() + " - " + bitisTarihi.toString(), tarihLabel.getText());
54 }
55 }
```

Şekil 5.7 Tatil Seçim Ekranı Buton Birim Testi

5.6 Ulaşım Seçim Ekranı Birim Testi

```
1 import javafx.application.Platform;
2 import javafx.embed.swing.JFXPanel;
3 import javafx.scene.control.*;
4 import javafx.stage.Stage;
5
6 import java.time.LocalDate;
7
8 public class UlasimSecimEkranıTest {
9
10     public static void main(String[] args) {
11         Thread thread = new Thread() -> {
12             new JFXPanel();
13             Platform.runLater() -> {
14                 testUlasimSecimEkranı();
15             };
16         };
17         thread.start();
18     }
19
20     public static void testUlasimSecimEkranı() {
21         ChoiceBox<String> kalkisYeriBox = new ChoiceBox<>();
22         kalkisYeriBox.getItems().addAll("İstanbul", "Ankara", "İzmir", "Antalya", "Bodrum");
23         kalkisYeriBox.setValue("İstanbul");
24
25         ChoiceBox<String> varisYeriBox = new ChoiceBox<>();
26         varisYeriBox.getItems().addAll("İstanbul", "Ankara", "İzmir", "Antalya", "Bodrum");
27         varisYeriBox.setValue("Ankara");
28
29         DatePicker tarihPicker = new DatePicker();
30         LocalDate tarih = LocalDate.now();
31         tarihPicker.setValue(tarih);
32
33         RadioButton ucakButton = new RadioButton("Uçak");
34         RadioButton trenButton = new RadioButton("Tren");
35         RadioButton otobusButton = new RadioButton("Otobüs");
36         ToggleGroup ulasimAraci = new ToggleGroup();
37         ucakButton.setToggleGroup(ulasimAraci);
38         trenButton.setToggleGroup(ulasimAraci);
39         otobusButton.setToggleGroup(ulasimAraci);
40
41         Button araButton = new Button("ARA");
```

Şekil 5.8 Ulaşım Seçim Ekranı Birim Testi - 1

```

42
43     araButton.setOnAction(event -> {
44         String kalkisYeri = kalkisYeriBox.getValue();
45         String varisYeri = varisYeriBox.getValue();
46         LocalDate secilenTarih = tarihPicker.getValue();
47         String ulasim = "";
48
49         if (ucakButton.isSelected()) {
50             ulasim = ucakButton.getText();
51         } else if (trenButton.isSelected()) {
52             ulasim = trenButton.getText();
53         } else if (otobusButton.isSelected()) {
54             ulasim = otobusButton.getText();
55         }
56
57         biletSecimEkranı biletSecim = new biletSecimEkranı(kalkisYeri, varisYeri, secilenTarih.toString(), ulasim);
58
59         if (biletSecim.getKalkisYeri().equals("İstanbul") &&
60             biletSecim.getVarisYeri().equals("Ankara") &&
61             biletSecim.getTarih().equals(secilenTarih.toString()) &&
62             biletSecim.getUlasim().equals(ulasim)) {
63             System.out.println("Test başarılı!");
64         } else {
65             System.out.println("Test başarısız!");
66         }
67
68         Platform.exit();
69     });
70
71     Platform.runLater(() -> {
72         Stage stage = new Stage();
73         ulasimSecimEkranı ulasimSecim = new ulasimSecimEkranı();
74         ulasimSecim.setKalkisYeriBox(kalkisYeriBox);
75         ulasimSecim.setVarisYeriBox(varisYeriBox);
76         ulasimSecim.setTarihPicker(tarihPicker);
77         ulasimSecim.setUcakButton(ucakButton);
78         ulasimSecim.setTrenButton(trenButton);
79         ulasimSecim.setOtobusButton(otobusButton);
80         ulasimSecim.setAraButton(araButton);
81         ulasimSecim.start(stage);
82     });
83 }
84 }

```

Şekil 5.9 Ulaşım Seçim Ekranı Birim Testi - 2