

# Yazılım Mühendisliği Proje Ödevi

## Çelebi Seyahat Acentesi Bilet-Tatil Sistemi



Yunus Emre SELÇUK

19011042 Yusuf Mert ÇELİKARSLAN

20011027 Yusuf Talha ATAŞ

19011077 Ahmet Enis ŞİMŞİR

19011093 Fatih ECEVİT

19011076 Muhammet Ali BUĞDAY

# PROJE PLANI

## Proje Alan Tanımı

- Çelebi Seyahat Acentesi, tatil ve ulaşım hizmetlerinin satışını gerçekleştiren bir firma olarak bilgi sistemi ihtiyacı duymaktadır.
- Bilgi sistemi, acente tarafından satılan biletlerin ve tatil rezervasyonlarının yönetimi için tasarlanacaktır.
- Acente, Uçan Türk özel havayolu, Devlet Demir Yolları (TCDD) ve YTUR otobüs firmalarının biletlerini satmaktadır. İleride başka firmaların biletlerinin satışının da yapılması düşünülmektedir.
- Müşteri bilet ya da tatil rezervasyonu yapmak istediğinde, hangi firmanın biletini istediğini ve tarih ile çıkış-varış noktalarını acente görevlisine bildirecektir.
- Görevli, müşterinin verdiği bilgileri sisteme girecek, ilgili firmanın bilet satış sistemi ile iletişime geçerek, mevcudiyet ve fiyat bilgilerini alacaktır.
- Müşteri, gösterilen fiyat ve seçenekler arasından tercihini yapacak, ödemesini yapacaktır. Görevli, biletin kesilmesi ve ilgili firmanın sistemine haberdar edilmesi işlemlerini gerçekleştirecektir.
- Acente, tatil rezervasyonları için anlaşmalı olduğu otellerin bulundukları bölgeler ve rezervasyon kabul ettikleri tarihleri sisteminde tanımlayacaktır.
- Müşteri, gitmek istediği bölgeyi ve tarihi görevliye söylediğinde, görevli bu bilgileri sisteme girecek, uygun alternatifleri müşteriye sunacaktır.
- Müşteri, seçtiği zaman ve otel için rezervasyon işlemini gerçekleştirecek, ödemesini yapacaktır. Görevli, ilgili otelin sistemine rezervasyon bilgilerini kaydedecek ve ödeme yapıldığında ilgili otel sistemini haberdar edecektir.
- Acente, nakit, kredi kartı ve çek ile ödemeyi kabul etmektedir. Başka ödeme türleri için planlanmamaktadır.
- Müşterilere, aldıkları biletler ve tatillerine göre belirli puanlar verilecektir. Puanlar, yönetim tarafından belirlenecek ve istendiğinde değiştirilebilecektir.

Müşteri, biriken puanlarını bilet ya da tatil rezervasyonu ödemesinde kullanabilir. Bu durumda, tutarda müşterinin puanı kadar indirim yapılacak ve indirilen tutar kadar puan silinecektir.

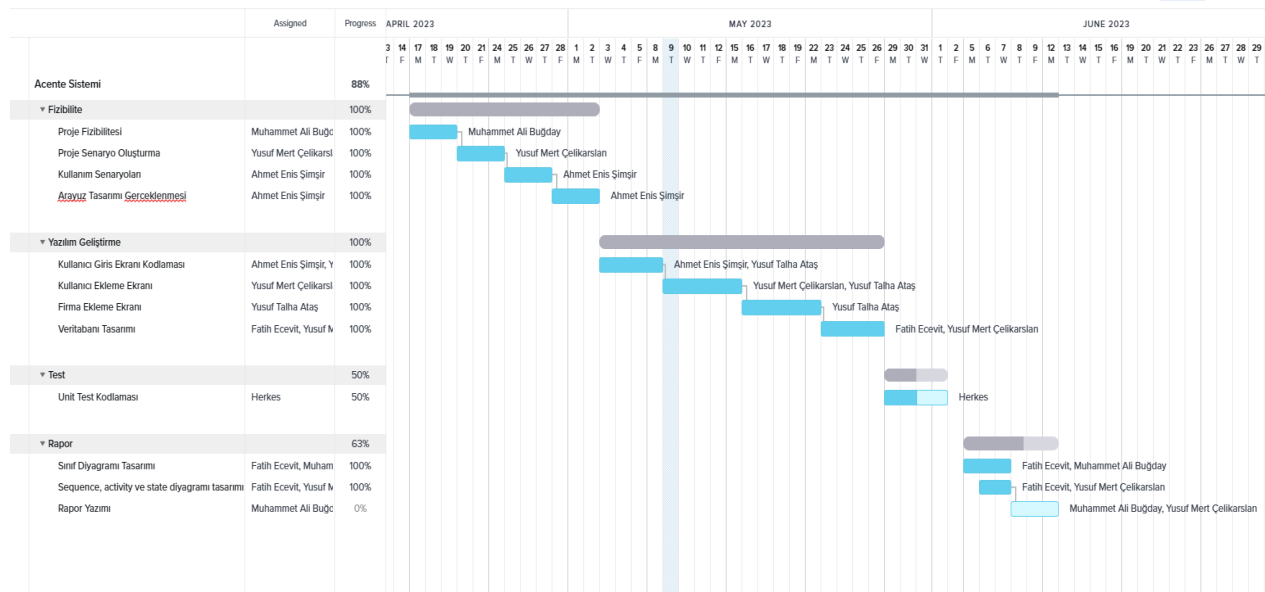
## Kabul Edilenler:

- Çelebi Seyahat Acentesi, Uçan Türk özel havayolu, Devlet Demir Yolları (TCDD) ve YTUR otobüs firmalarının biletlerini satabilir.
- Acente, çeşitli otellerle anlaşma yaparak tatil rezervasyonu da yapabilir.
- Müşteri bilet almak veya tatil rezervasyonu yapmak için, tarih ile çıkış-varış noktalarını veya gitmek istediği bölgeyi ve tarihi belirtmelidir.
- Görevli, müşterinin verdiği bilgileri sisteme girecek, mevcudiyet ve fiyat bilgilerini ilgili firmanın sisteminden alacak ve müşteriye sunacak.
- Müşteri ödemeyi yaparsa, bileti keser ve ilgili firmanın sistemini haberdar eder.
- Acente, nakit, kredi kartı ve çek ile ödeme kabul eder.
- Müşterilere aldıkları biletler ve tatillerine göre belirli puanlar verilir.
- Müşteri, biriken puanlarını ödeme yaparken kullanabilir.

## Kısıtlar:

- Sistemde sadece Uçan Türk, TCDD ve YTUR firmalarının biletleri satılabilir. Başka firmaların biletleri için ek özellik eklenemez.
- Müşterinin bilet veya tatil rezervasyonu yaparken, sisteme kaydedilecek olan bilgiler tarih, çıkış-varış noktaları ve bölge ile sınırlıdır. Müşterilerin kişisel bilgileri kaydedilemez.
- Sistemde ödeme yapmak için sadece nakit, kredi kartı ve çek kabul edilir. Başka ödeme türleri eklenemez.
- Puan verme işlemi sadece yönetim tarafından belirlenen kriterlere göre yapılır. Müşteri talebi üzerine puan verilemez veya puan miktarı değiştirilemez.
- Müşteri puanlarını sadece ödeme yaparken kullanabilir. Puanlar nakit olarak geri ödenemez veya başka amaçlar için kullanılamaz.

### Proje İş-Zaman Çizelgesi (Gantt):



Ekip organizasyon şeması, görev dağılımları:



## Risk Tablosu:

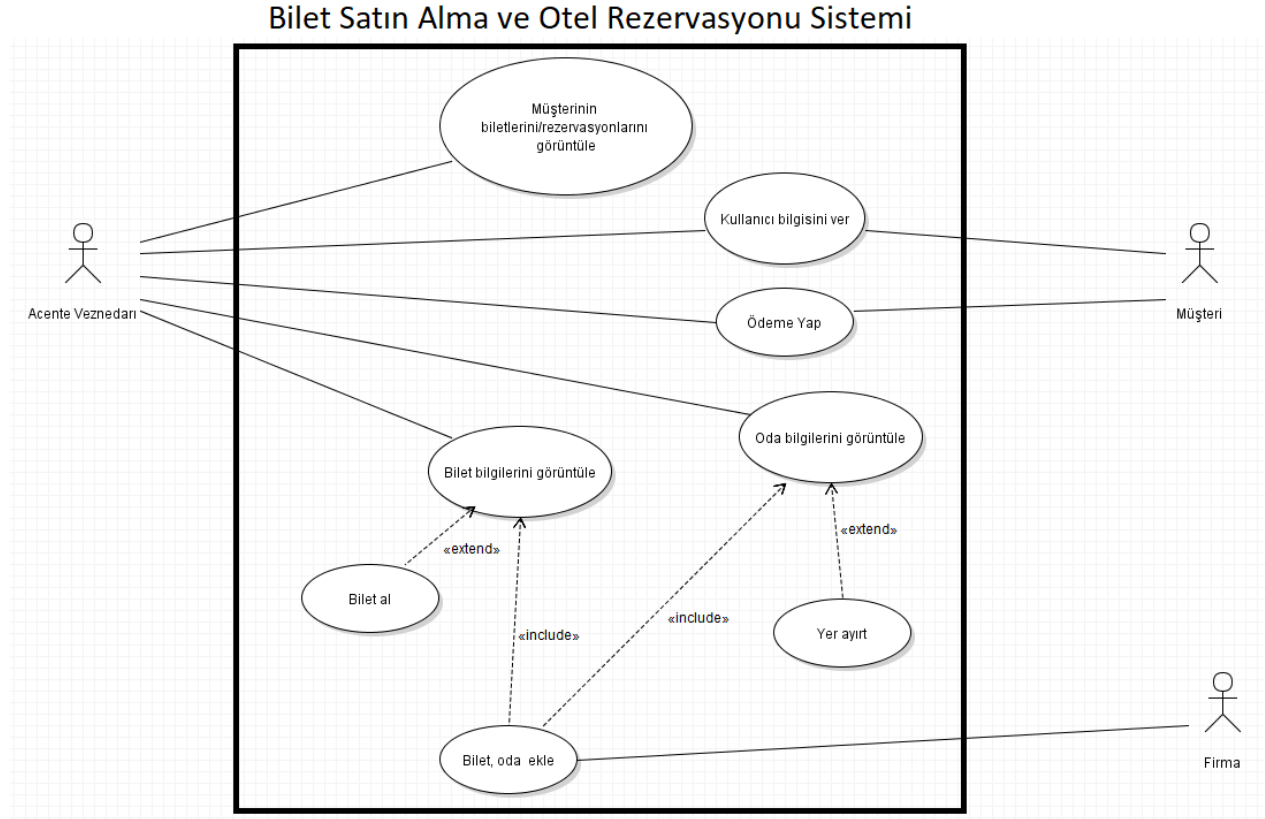
| Tehlike                                     | Olası Sonuç  | Risk Seviyesi | Önlem  |
|---|--|---------------|--|
| Güvenlik açıkları nedeniyle veri hırsızlığı | Kişisel ve ödeme bilgilerinin çalınması, şirketin itibarının zarar görmesi, müşteri kaybı                  | Yüksek        | Firestore gibi güvenli bir veri tabanı kullanarak, düzenli olarak veri yedeklemesi yaparak ve yazılım güncellemelerini takip ederek risk seviyesi azaltılabilir.   |
| İşlem yaparken sistem kesintisi             | Müşterilerin işlemlerinin tamamlanmaması, müşteri kaybı  | Orta          | Sistemin yüksek kapasitesine dayanabilecek bir altyapı kullanmak ve yedek sistemlerin oluşturulması gibi önlemler alınarak risk seviyesi azaltılabilir.  |
| Yanlış bilgi veya işlem hatası              | Yanlış fiyat veya yanlış uçuş saatleri gibi yanlış bilgilendirme, müşteri memnuniyetsizliği, mali kayıplar | Orta          | Sistemin test edilmesi ve doğru bir şekilde yapılandırılması, müşteri hizmetlerinde eğitimli çalışanların bulunması gibi önlemler alınarak risk seviyesi azaltılabilir.                                      |
| Sistem yöneticisi hataları                  | Yanlış veri silme veya yanlışlıkla veri değiştirme, mali kayıplar, şirket itibarının zarar görmesi         | Orta          | Verilerin düzenli olarak yedeklenmesi, sistem yöneticilerinin eğitilmesi, yetkilendirme ve erişim haklarının kontrol edilmesi gibi önlemler alınarak risk seviyesi azaltılabilir.                            |
| Yazılım hataları                            | Uygulamanın çökmesi, müşteri memnuniyetsizliği, mali kayıplar  | Orta-Yüksek   | Yazılımın doğru şekilde test edilmesi, düzenli yazılım güncellemeleri, açık kaynak kod kullanımı ve güvenlik kontrollerinin sıkı bir şekilde uygulanması gibi önlemler alınarak risk seviyesi azaltılabilir. |

# İSTEKLERİN MODELLENMESİ

## Kullanıcı Sözleşmeleri:

|                              |   |
|------------------------------|---|
| Birincil Aktör               | Kasiyer   |
| İlgililer ve İlgili Alanları | Müşteri: İstediği tarihlere uygun bilet almak ister<br>Kasiyer: Müşterinin bilet işlemlerini halledip<br>ödemeyi almak ister  |
| Ön Koşullar                  | Müşteri kullanıcı adını ve almak istediği bilet tipini<br>söyler  |
| Son Koşullar                 | Müşteri biletini alıp kasiyere ödemeyi yapar  |
| Ana Senaryo                  | Müşteri kullanıcı adını söyler<br>Kasiyer müşterinin adına giriş yapar, kayıtlı değilse<br>önce kayıt olur<br>Kasiyer eğer müşteri otel bileti almak istiyorsa<br>otel bileti ekranına, otobüs bileti almak istiyorsa<br>otobüs bileti ekranına girer<br>Eğer otobüs bileti ekranına girdiyse müşterinin<br>istediği firmayı seçer<br>Sonrasında tarih bilgisini girer ve o tarihteki<br>seferler listelenir<br>Müşteri seçim yapar ve istediği koltuk numarasını<br>girer<br>Kasiyer onaylar ve ödeme ekranı gelir<br>Müşteri kredi kartı, nakit veya çek ile ödemeyi<br>seçebilir<br>Eğer müşterinin indirim puanı varsa o puan kadar<br>indirim yapılır<br>Müşteri kasiyere parayı öder ve ücretin yüzde<br>10'u kadar indirim puanı kazanır |
| Alternatif Senaryo           | <b>Müşteri otobüs değil, otel bileti almak ister</b><br>Otel seçim ekranında istenen tarih ve otel şirketi<br>seçilir<br>Uygun odaların bilgisi gelir<br>Müşteri seçim yapar, kasiyer o seçimin olduğu<br>butona tıklar<br>Ödeme ekranı gelir ve kasiyer müşterinin isteği<br>doğrultusunda kredi kartı, nakit veya çek yoluyla<br>ödemeyi seçer. Müşterinin indirim puanı kadar<br>ödemeden kesinti yapılır.<br>Müşteri kasiyere parayı öder. Ücretin yüzde onu<br>kadar indirim puanı kazanır   |

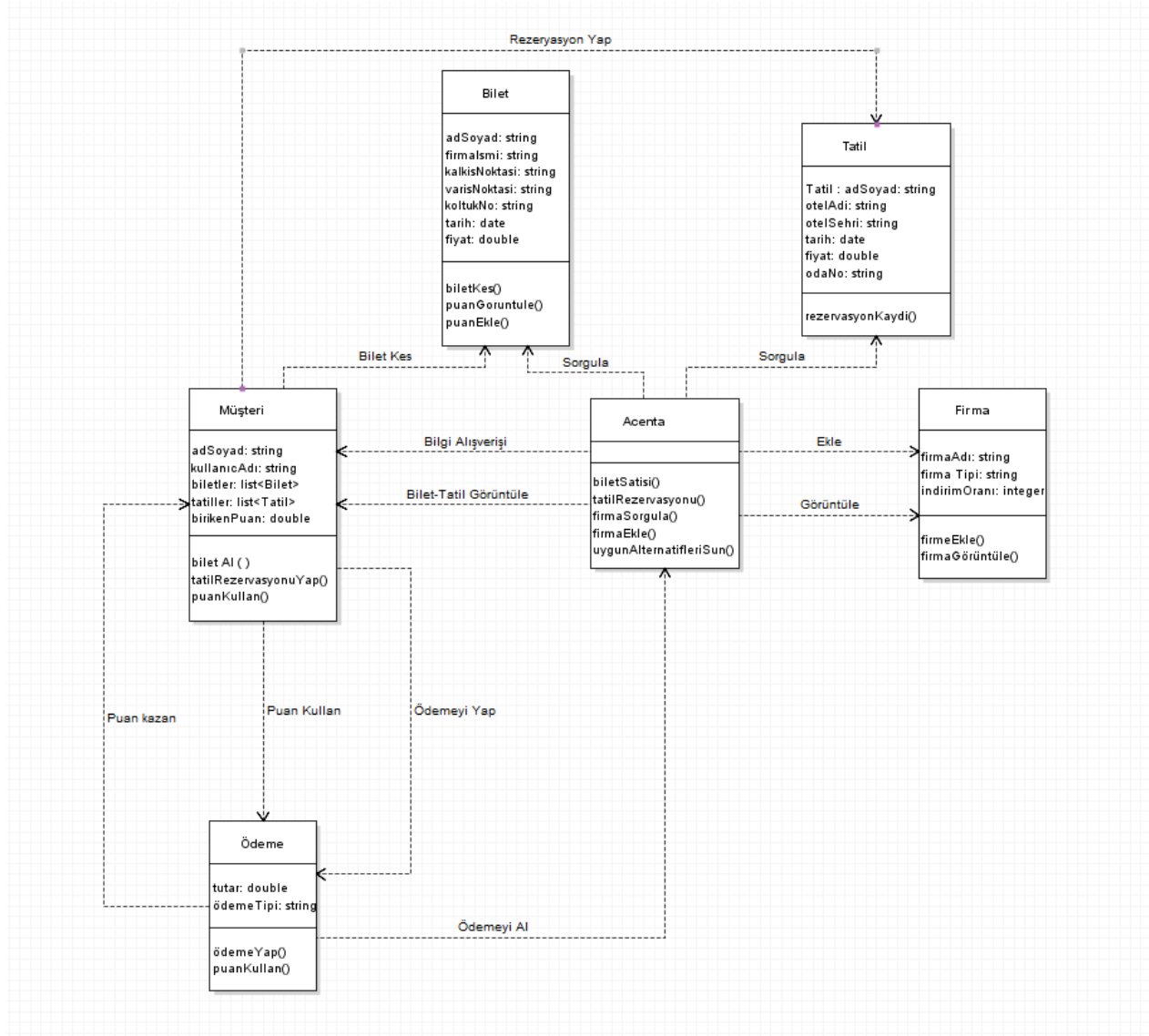
## Projenin use-case modeli:



## İzlenebilirlik Tablosu:

|                          | Acente   | Müşteri  | Firma    |
|--------------------------|----------|----------|----------|
| Ödeme yapma              |          | <u>X</u> |          |
| İndirim puanı kazanma    |          | <u>X</u> |          |
| Yeni bilet ekleme        |          |          | <u>X</u> |
| Yeni otel rez ekleme     |          |          | <u>X</u> |
| Biletleri listeleme      | <u>X</u> |          |          |
| Otel odalarını listeleme | <u>X</u> |          |          |
| Bilet kesme              | <u>X</u> |          |          |
| Rezervasyon yapma        | <u>X</u> |          |          |
| Yeni kullanıcı oluşturma | <u>X</u> |          |          |
| Ödeme alma               | <u>X</u> |          | <u>X</u> |

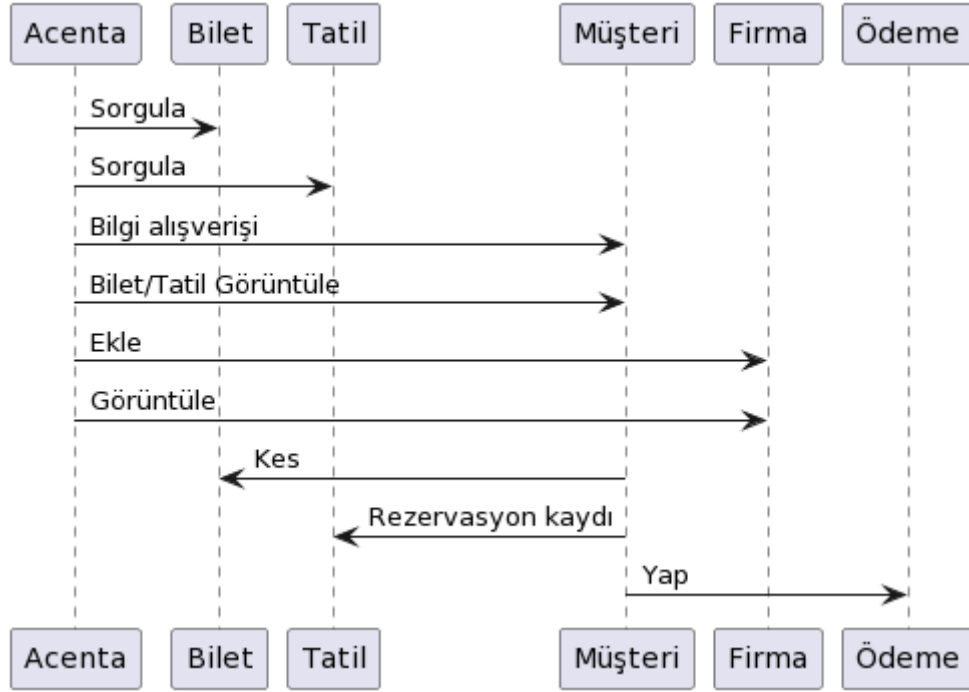
# Nesneye Dayalı Modelleme



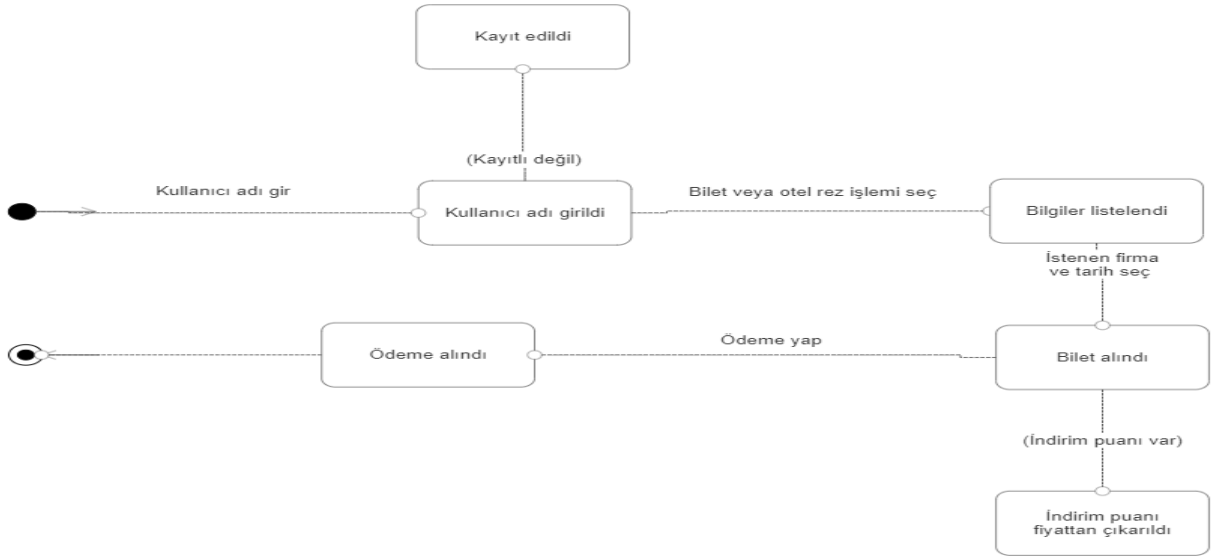


# Nesneye Dayalı Tasarım

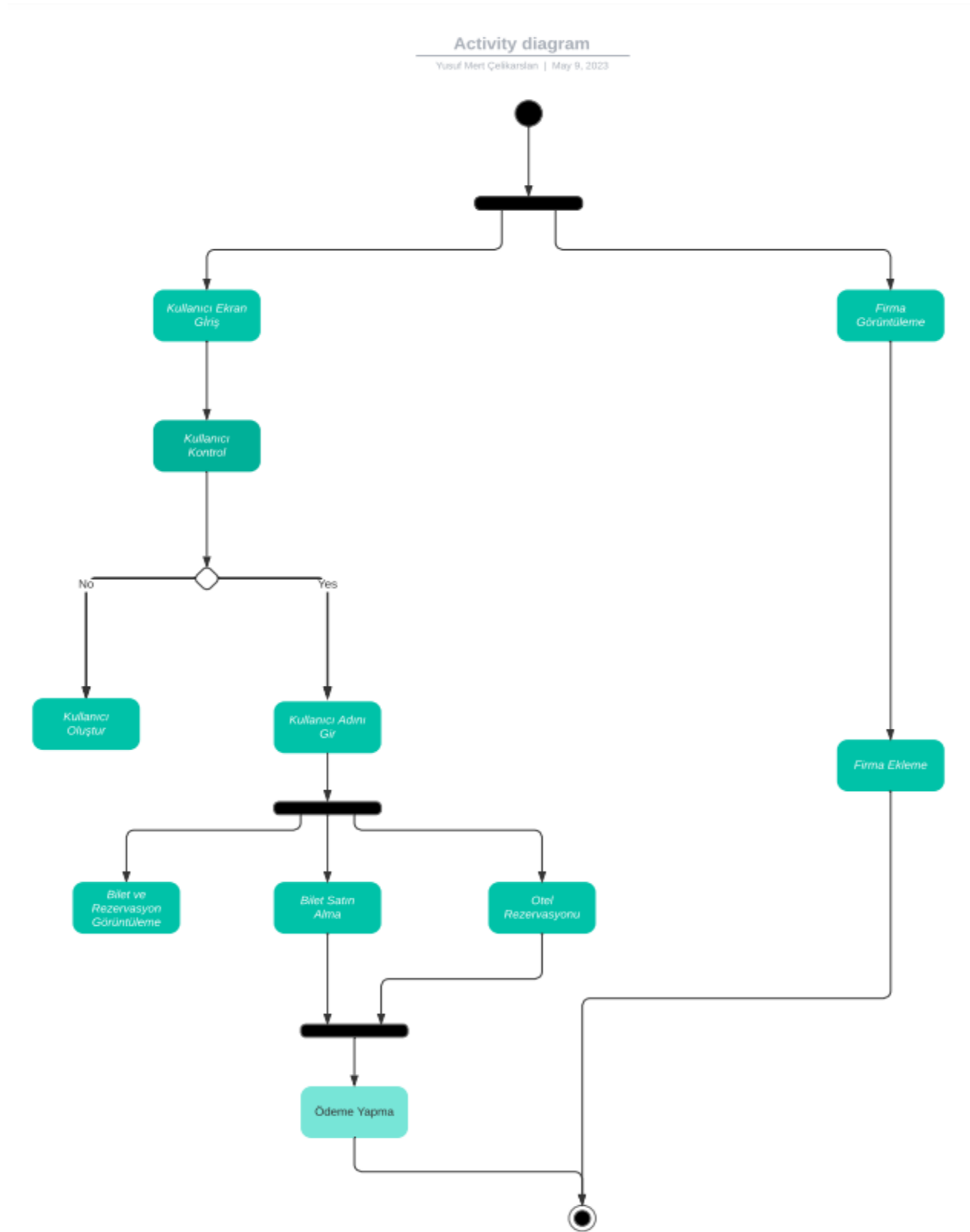
Sıralama (Sequence) Diagram:



## Durum (State) Diagram:



## Etkinlik (Activity) Diagram:



# BİRİM TEST SINAMALARI

```
> CelebiSeyehat@0.0.1 test
> jest
```

```
watchman warning: opendir(/Users/yusuftalhaatas/Library/Containers/com.apple.corerecents.recentsd/
Data/Library/Recents) -> Operation not permitted. Marking this portion of the tree deleted
To clear this warning, run:
`watchman watch-del '/Users/yusuftalhaatas' ; watchman watch-project '/Users/yusuftalhaatas'`
```

```
PASS  __tests__/components/customButton.test.tsx
PASS  __tests__/screens/SelectScreen.test.tsx
PASS  __tests__/screens/AgentyScreen.test.tsx
PASS  __tests__/components/firmTypeButton.test.tsx
PASS  __tests__/components/ticketCard.test.tsx
PASS  __tests__/components/firmCard.test.tsx
PASS  __tests__/components/customTextInput.test.tsx
PASS  __tests__/components/hotelCard.test.tsx
```

```
Test Suites: 8 passed, 8 total
Tests:       12 passed, 12 total
Snapshots:   7 passed, 7 total
Time:        1.119 s
Ran all test suites.
```

# Yusuf Talha Ataş

```
import React from 'react';
import {render, fireEvent} from '@testing-library/react-native';
import CustomButton from '../src/components/customButton';

describe('components/CustomButton', () => {
  test('renders correctly and responds to press events', () => {
    const onPressMock = jest.fn();
    const {getByText, toJSON} = render(
      <CustomButton title="Test Button" onPress={onPressMock} />,
    );

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // onPress event test
    fireEvent.press(getByText('Test Button'));
    expect(onPressMock).toHaveBeenCalled();
  });

  test('renders correctly when disabled', () => {
    const onPressMock = jest.fn();
    const {getByText, toJSON} = render(
      <CustomButton title="Test Button" onPress={onPressMock} disabled />,
    );

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // onPress event should not trigger
    fireEvent.press(getByText('Test Button'));
    expect(onPressMock).not.toHaveBeenCalled();
  });
});
```

# Ahmet Enis ŞİMŞİR

```
import React from 'react';
import {render, fireEvent} from '@testing-library/react-native';
import CustomInput from '../../src/components/customTextInput';

describe('components/CustomInput', () => {
  test('renders correctly and responds to text changes', () => {
    const onChangeTextMock = jest.fn();
    const {getByPlaceholderText, toJSON} = render(
      <CustomInput
        placeholder="Test Input"
        onChangeText={onChangeTextMock}
        value=""
      />,
    );

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // onChangeText event test
    fireEvent.changeText(getByPlaceholderText('Test Input'), 'test');
    expect(onChangeTextMock).toHaveBeenCalledWith('test');
  });
});
```

# Fatih Ecevit

```
import {render} from '@testing-library/react-native';
import FirmCard from '../../src/components/firmCard';
import {firm} from '../../src/types/firm';

describe('components/FirmCard', () => {
  test('renders correctly', () => {
    const firm: firm = {
      firmName: 'Test Firm',
      firmType: 'transport',
      discountRate: 20,
    };

    const {getByText, toJSON} = render(<FirmCard {...firm} />);

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // props test
    expect(getByText('Test Firm')).toBeDefined();
    expect(getByText('Discount:%20')).toBeDefined();
  });
});
```

```
import {render, fireEvent} from '@testing-library/react-native';
import FirmTypeButton from '../../src/components/firmTypeButton';

describe('components/FirmTypeButton', () => {
  test('renders correctly and responds to onPress', () => {
    const onPressMock = jest.fn();
    const {getByText, toJSON} = render(
      <FirmTypeButton name="Test Button" onPress={onPressMock} />,
    );

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // onPress event test
    fireEvent.press(getByText('Test Button'));
    expect(onPressMock).toHaveBeenCalled();
  });
});
```

# Yusuf Mert Çelikarslan

```
import React from 'react';
import {render} from '@testing-library/react-native';
import HotelCard from '../src/components/hotelCard';

describe('components/HotelCard', () => {
  test('renders correctly', () => {
    const hotelData = {
      firmName: 'Test Hotel',
      city: 'Test City',
      firmType: 'Hotel',
      roomNo: '101',
      date: '2023-05-15',
      name: 'Test Name',
    };

    const {getByText, toJSON} = render(<HotelCard {...hotelData} />);

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // Check if all fields are displayed correctly
    expect(getByText(`Name: ${hotelData.name}`)).toBeDefined();
    expect(getByText(`Hotel Name: ${hotelData.firmName}`)).toBeDefined();
    expect(getByText(`Date: ${hotelData.date}`)).toBeDefined();
    expect(getByText(`Room No: ${hotelData.roomNo}`)).toBeDefined();
    expect(getByText(`City: ${hotelData.city}`)).toBeDefined();
  });
});
```

```
import React from 'react';
import {render} from '@testing-library/react-native';

import TicketCard from '../src/components/ticketCard';

describe('components/TicketCard', () => {
  test('renders correctly', () => {
    const ticketData = {
      date: '2023-05-15',
      start: 'Test Start',
      finish: 'Test Finish',
      firmName: 'Test Firm',
      no: 'A101',
      name: 'Test Name',
    };

    const {getByText, toJSON} = render(<TicketCard {...ticketData} />);

    // snapshot test
    expect(toJSON()).toMatchSnapshot();

    // Check if all fields are displayed correctly
    expect(getByText(`Name: ${ticketData.name}`)).toBeDefined();
    expect(getByText(`Firm: ${ticketData.firmName}`)).toBeDefined();
    expect(getByText(`${ticketData.start}-${ticketData.finish}`)).toBeDefined();
    expect(getByText(`Date: ${ticketData.date}`)).toBeDefined();
    expect(getByText(`Seat No: ${ticketData.no}`)).toBeDefined();
  });
});
```



# Muhammet Ali Buğday

```
import React from 'react';
import {render, fireEvent} from '@testing-library/react-native';
import AgentyScreen, {AgentyScreenProps} from '../../src/screens/AgentyScreen';

describe('AgentyScreen', () => {
  it('should navigate to UserInputScreen when "USER PROCESSES" button is pressed', () => {
    const mockNavigation = {navigate: jest.fn()} as any;
    const props: any = {navigation: mockNavigation};

    const {getByText} = render(<AgentyScreen {...props} />);
    const userProcessesButton = getByText('USER PROCESSES');

    fireEvent.press(userProcessesButton);

    expect(mockNavigation.navigate).toHaveBeenCalledWith('UserInputScreen');
  });

  it('should navigate to FirmsScreen when "FIRM PROCESSES" button is pressed', () => {
    const mockNavigation = {navigate: jest.fn()} as any;
    const props: any = {navigation: mockNavigation};

    const {getByText} = render(<AgentyScreen {...props} />);
    const firmProcessesButton = getByText('FIRM PROCESSES');

    fireEvent.press(firmProcessesButton);

    expect(mockNavigation.navigate).toHaveBeenCalledWith('FirmsScreen');
  });
});
```

```
import React from 'react';
import {render, fireEvent} from '@testing-library/react-native';
import SelectScreen from '../../src/screens/SelectScreen';

describe('SelectScreen', () => {
  it('should navigate to TicketBuyScreen when "BUY TRANSPORT TICKET" button is pressed', () => {
    const mockNavigation = {navigate: jest.fn()} as any;
    const props: any = {navigation: mockNavigation};

    const {getByText} = render(<SelectScreen {...props} />);
    const buyTicketButton = getByText('BUY TRANSPORT TICKET');

    fireEvent.press(buyTicketButton);

    expect(mockNavigation.navigate).toHaveBeenCalledWith('TicketBuyScreen');
  });

  it('should navigate to HotelReserveScreen when "HOTEL RESERVE" button is pressed', () => {
    const mockNavigation = {navigate: jest.fn()} as any;
    const props: any = {navigation: mockNavigation};

    const {getByText} = render(<SelectScreen {...props} />);
    const hotelReserveButton = getByText('HOTEL RESERVE');

    fireEvent.press(hotelReserveButton);

    expect(mockNavigation.navigate).toHaveBeenCalledWith('HotelReserveScreen');
  });
});
```

# UYGULAMA GÖRÜNTÜLERİ

2:08

AgentyScreen

ÇELEBİ SEYAHAT ACENTASI

USER PROCESSES

FIRM PROCESSES

2:08

UserInputScreen

USER ENTER

UserName

Invalid username

CONTINUE

User Add

2:09

UserInputScreen

USER ENTER

yusuftalha

Correct usurname

CONTINUE

User Add

2:09

UserAddScreen

ENTER USER INFO

Enter Unique UserName

yusuftalha

Enter Your Name

Yusuf Talha Atay

ADD

2:12

SelectScreen

SELECT

BUY TRANSPORT TICKET

HOTEL RESERVE

USERS TICKETS/HOTELS

2:10

TicketBuyScreen

ENTER TICKET INFO

Puan:18

Select Date

22 Ocak 2023

23 Şubat 2021

24 Mart 2022

25 Nisan 2023

26 Mayıs 2024

27 Haziran 2025

28 Temmuz 2026

İnegöl Seyahat

Uçan Türk Özel Hava yolu

YTUR

Metro Turizim

TCDD

Seat No

OKAY

2:10

TicketBuyScreen

ENTER TICKET INFO

Puan:18

Select Date

22 Ocak 2023

23 Şubat 2021

24 Mart 2022

25 Nisan 2023

26 Mayıs 2024

27 Haziran 2025

28 Temmuz 2026

Start: istanbul price: 1000TL

Dest: gaziantep

Start: bursa price: 300TL

Dest: istanbul

Start: bursa price: 400TL

Dest: izmir

12

OKAY

2:11

PayingScreen

Name: Yusuf Talha Atay

Firm: TCDD

bursa-izmir

Date:5/25/2023

Seat No:12

Credit Card

Cash

Check

Ticket Price + 400TL

Discount Puan- 18TL

Total Price: 382TL

You will earn 4TL from this purchase

BUY

2:11

SuccessScreen

Successful purchase

OKAY

2:13

HotelReserveScreen

ENTER HOTEL INFO

Select Date

|    |         |      |
|----|---------|------|
| 20 | Şubat   | 2020 |
| 21 | Mart    | 2021 |
| 22 | Nisan   | 2022 |
| 23 | Mayıs   | 2023 |
| 24 | Haziran | 2024 |
| 25 | Temmuz  | 2025 |
| 26 | Ağustos | 2026 |

Hilton

OKAY

2:13

HotelReserveScreen

ENTER HOTEL INFO

Select Date

|    |         |      |
|----|---------|------|
| 20 | Şubat   | 2020 |
| 21 | Mart    | 2021 |
| 22 | Nisan   | 2022 |
| 23 | Mayıs   | 2023 |
| 24 | Haziran | 2024 |
| 25 | Temmuz  | 2025 |
| 26 | Ağustos | 2026 |

Date: 5/23/2023 price: 900TL  
City: Antalya

OKAY

2:14

HotelPaymentScreen

Name: Yusuf Talha Atas

Hotel Name: Hilton

Date: 5/23/2023

Room No: A111

City: Antalya

Credit Card

Cash

Check

Ticket Price + 900TL  
Discount Puan- 4TL  
Total Price: 896TL

You will earn 18TL from this purchase

BUY

2:23

UserTicketsHotelsScreen

Hotels

Name: Yusuf Talha Atas

Hotel Name: Hilton

Date: 04/20/2023

Room No: A288

City: Antalya

Name: Yusuf Talha Atas

Hotel Name: Hilton

Tickets

Name: Yusuf Talha Atas

Firm: TCDD

bursa-istanbul

Date: 6/20/2023

Seat No: 12

Name: Yusuf Talha Atas

Firm: Türk Hava Yolları

2:24

AgencyScreen FirmsScreen

Hilton

Type: hotel

Discount: %2

İnegöl Seyahat

Type: transport

Discount: %1

Uçan Türk Özel Havayolu

Type: transport

Discount: %1

YTUR

Type: transport

Discount: %1

Metro Turizim

Type: transport

Discount: %1

TCDD

Type: transport

Discount: %1

ADD FIRM

2:24

FirmsScreen FirmAddScreen

Enter Firm Name

Firm Name

Enter Discount Rate

Discount Rate

Hotel

Transport

ADD FIRM