

Bilgisayar Mühendisliği Programı
Yazılım Mühendisliği Dersi 2021/2022 Dönemi
Dönem Projesi

Dil Kursu Otomasyon Yazılımı

Hazırlayanlar

Utku Magemizoğlu, 19011097
Ömer Talha BAYSAN, 18011103
Muhammed Hakan Kılıç, 19011081
Gıyat Mosa, 18011113
Rayene Bech, 18011115

20 Mayıs 2022

Dr. Öğr. Üyesi Yunus Emre SELÇUK
Yazılım Mühendisliği

İÇİNDEKİLER

1. Proje Planı
 - 1.1 Proje Alan Tanımı
 - 1.2 Kabul ve Kısıtlar
 - 1.3 Proje İş-Zaman Çizelgesi (Gantt Diyagramı)
 - 1.4 Ekip Organizasyon Şeması, Görev Dağılımları
 - 1.5 Risk Tablosu
2. İsteklerin Modellenmesi
 - 2.1 Kullanım Senaryoları
 - 2.2 UML Kullanım Şemaları
3. Nesneye Dayalı Modelleme
 - 3.1 UML Sınıf Şemaları
4. Nesneye Dayalı Tasarım
 - 4.1 UML Sıralama Şeması (Sequence Diagram)
 - 4.2 UML Etkinlik Şeması (Activity Diagram)
 - 4.3 UML Durum Diyagramı (State Diagram)
5. Birim Testi Sınamaları
6. Kullanıcı Arayüzleri

Dil Kursu Otomasyon Yazılımı

“Bir Lisan Bir İnsan” dil kursu şirketi öğretmen kadrosunu genişletmeyi, yeni şubeler açmayı planlamaktadır ve bir otomasyon yazılımına gereksinim duymaktadır. Belirtilen özelliklere uygun bir bilgi sistemi tasarlanmıştır.

1. Proje Planı

1.1 Proje Alan Tanımı

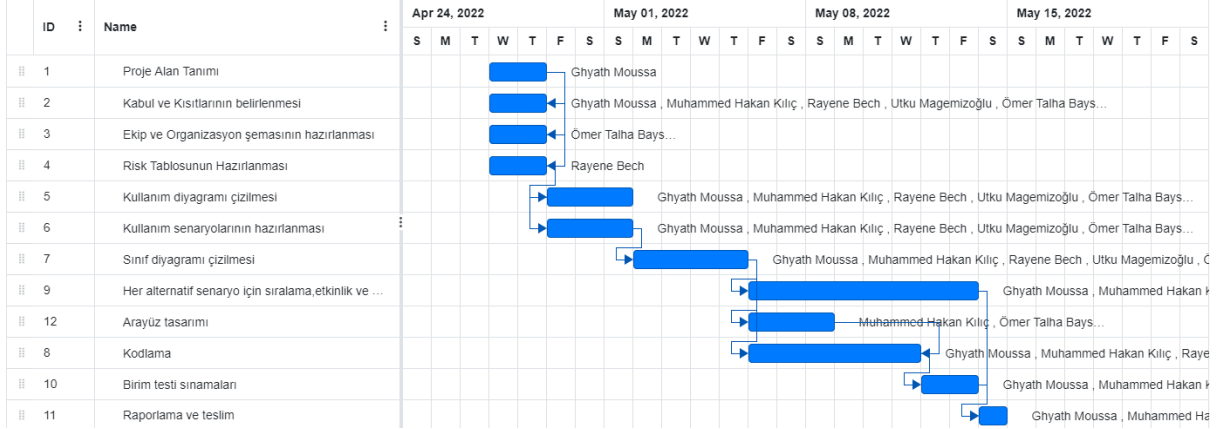
Dil kursu otomasyon sistemi, bir dil kursunun ihtiyaç duyduğu derslerin, öğrencilerini, ödeme kayıtlarının kayıt altına alınması, yeni şubeler açılması, yeni öğretmenler eklenmesi için bu yazılım geliştirilmiştir. Oluşturduğumuz yazılım bu işlemlerin elektronik olarak yönetilebilmesini sağlayarak zaman ve maliyet açısından tasarruf yapılmasına olanak sağlamıştır.

Amaç, Öğrencilerin istedikleri kursu kolayca seçilmesi ve yöneticiler tarafından sınıfların şubelerin ve öğretmenlerin kolay bir şekilde yönetilmesini sağlamaktır.

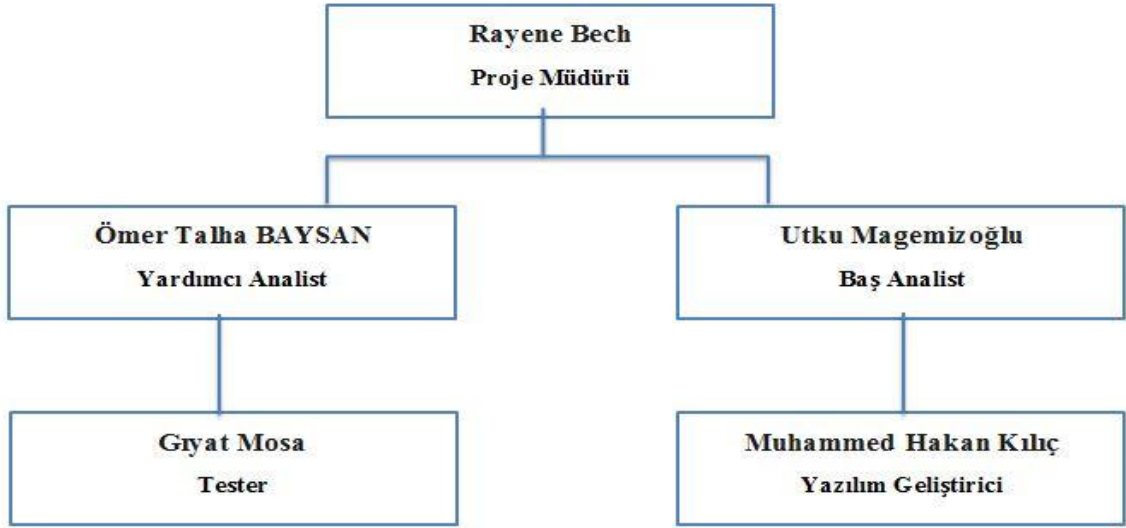
1.2 Kabul ve Kısıtlar

- Projemizde sadece öğrenci ekleme ve öğrenciye ders ekleme işlemlerini kodladık. Bu nedenle sisteme hardcoded olarak şubeler, derslikler, dersler işlenmiştir.
- Ders ödemelerinde peşin veya 6 taksit yapılır.
- Bir öğrenci sadece bir dil kursuna kayıt olabilir.
- Bir öğrenci silme işlemi için öğrenci önce kurstan silinmeli.

1.3 Proje İş-Zaman Çizelgesi (Gantt Diyagramı)



1.4 Ekip Organizasyon Şeması, Görev Dağılımları



1.5 Risk Tablosu

Risk ID	Adı	Türü / Grubu	Etkisi	Olasılık
01	Zaman Baskısı	Proje	orta	yüksek
02	Organizasyon	Proje	büyük	düşük
03	Bütçe Kısıtları	Proje	büyük	orta
04	Personel eksikliği	Proje	orta	düşük
05	Müşteri Gereksinimlerinin Değişimi	Proje	büyük	düşük
06	Kullanıcı arayüzlerinin ihtiyaçları karşılamaması	Proje	büyük	düşük

07	Güvenlik	Teknik	büyük	orta
08	Doğruluk	Teknik	büyük	yüksek
09	Ölçeklenebilirlik	Teknik	orta	orta
10	Dokümantasyon eksikliği	Teknik	orta	büyük
11	Pazarlama	İş	büyük	düşük
12	Sözleşme ve yasal riskler	İş	orta	düşük

2. İsteklerin Modellenmesi

Öğrencilerin istedikleri kursu kolayca seçilmesi ve yöneticiler tarafından sınıfların, şubelerin ve öğretmenlerin kolay bir şekilde yönetilmesini sağlamaktır.

2.1 Kullanım Senaryoları

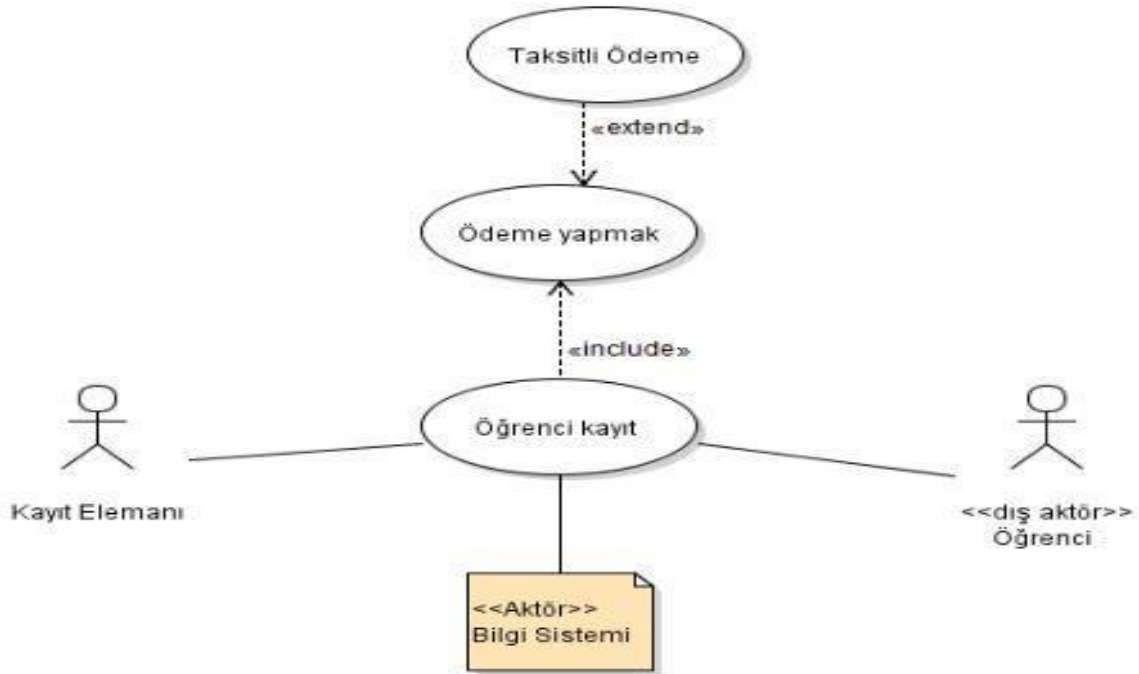
Kullanım Senaryosu	Şube Bilgisi Girişi
Birincil Aktör	Sistem Yöneticisi
İlgililer ve Beklentileri	Sistem yöneticisi: Sisteme erişip, yeni şube bilgilerini girerek şubeyi sisteme eklemek ister.
Ön Koşullar	Sistem yöneticisi kimliği doğrulanır.
Son Koşullar	Şube bilgileri sisteme hatasız ve eksiksiz işlenir.
Ana Senaryo	<ol style="list-style-type: none"> 1. Sistem yöneticisi yeni şube adını belirler. 2. Sistem yöneticisi yeni şubenin adresini, gerekli toplu taşıma veya özel otomobil ile ulaşım talimatlarının girişini yapar. 3. Sistem yöneticisi şubenin sosyal olanaklarının girişini yapar. 4. Sistem yöneticisi şube binasındaki dersliklerin girişini yapar. 5. Sistem bilgileri kaydeder.
Alternatif Senaryo	<ol style="list-style-type: none"> 1. Aynı isimde bir şube vardır. <ol style="list-style-type: none"> a. Sistem, yönetici bir uyarı mesajı verir ve yeni şube ismi ister. 4. Derslik limiti dolmuştur. <ol style="list-style-type: none"> a. Belirli sayıdan fazla derslik eklenmeye çalışıldığında system hata vermektedir.

Kullanım Senaryosu	Şube Bilgisi Girişi
Birincil Aktör	Sistem Yöneticisi
İlgililer ve Beklentileri	Sistem yöneticisi: Sisteme erişip, yeni şube bilgilerini girerek şubeyi sisteme eklemek ister.
Ön Koşullar	Sistem yöneticisi kimliği doğrulanır.
Son Koşullar	Şube bilgileri sisteme hatasız ve eksiksiz işlenir.
Ana Senaryo	<ol style="list-style-type: none"> 1. Sistem yöneticisi yeni şube adını belirler. 2. Sistem yöneticisi yeni şubenin adresini, gerekli toplu taşıma veya özel otomobil ile ulaşım talimatlarının girişini yapar. 3. Sistem yöneticisi şubenin sosyal olanaklarının girişini yapar. 4. Sistem yöneticisi şube binasındaki dersliklerin girişini yapar. 5. Sistem bilgileri kaydeder.
Alternatif Senaryo	<ol style="list-style-type: none"> 1. Aynı isimde bir şube vardır. <ol style="list-style-type: none"> a. Sistem, yönetici bir uyarı mesajı verir ve yeni şube ismi ister. 4. Derslik limiti dolmuştur. <ol style="list-style-type: none"> a. Belirli sayıdan fazla derslik eklenmeye çalışıldığında system hata vermektedir.

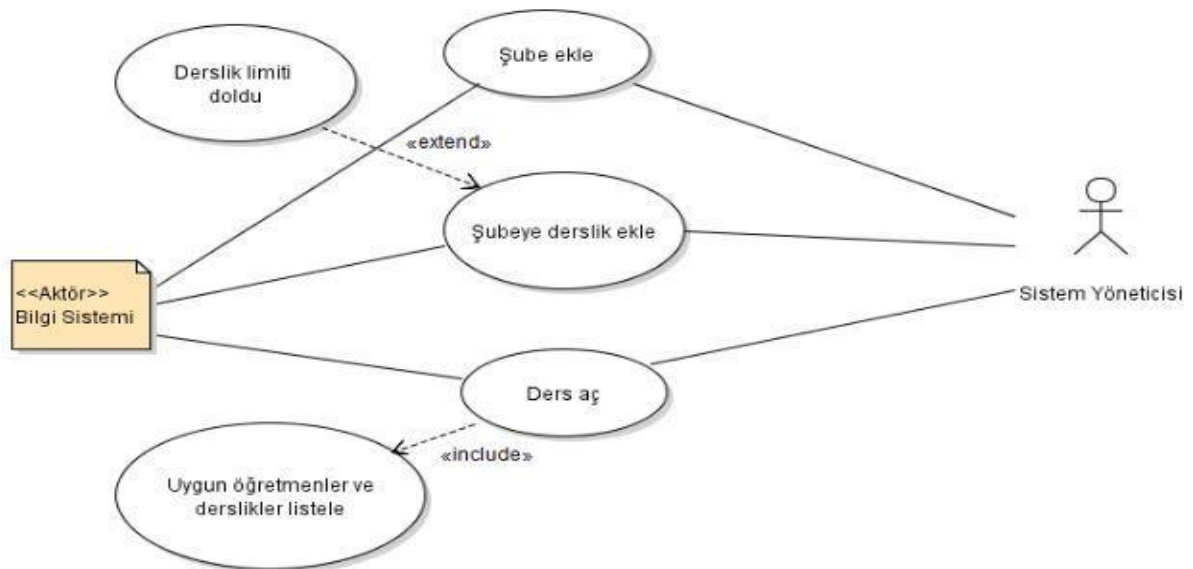
Kullanım Senaryosu	Şube Bilgisi Girişi
Birincil Aktör	Sistem Yöneticisi
İlgililer ve Beklentileri	Sistem yöneticisi: Sisteme erişip, yeni şube bilgilerini girerek şubeyi sisteme eklemek ister.
Ön Koşullar	Sistem yöneticisi kimliği doğrulanır.
Son Koşullar	Şube bilgileri sisteme hatasız ve eksiksiz işlenir.
Ana Senaryo	<ol style="list-style-type: none"> 1. Sistem yöneticisi yeni şube adını belirler. 2. Sistem yöneticisi yeni şubenin adresini, gerekli toplu taşıma veya özel otomobil ile ulaşım talimatlarının girişini yapar. 3. Sistem yöneticisi şubenin sosyal olanaklarının girişini yapar. 4. Sistem yöneticisi şube binasındaki dersliklerin girişini yapar. 5. Sistem bilgileri kaydeder.
Alternatif Senaryo	<ol style="list-style-type: none"> 1. Aynı isimde bir şube vardır. <ol style="list-style-type: none"> a. Sistem, yönetici bir uyarı mesajı verir ve yeni şube ismi ister. 4. Derslik limiti dolmuştur. <ol style="list-style-type: none"> a. Belirli sayıdan fazla derslik eklenmeye çalışıldığında system hata vermektedir.

2.2 UML Kullanım Şemaları

- Öğrenci Kayıt Kullanım Şeması

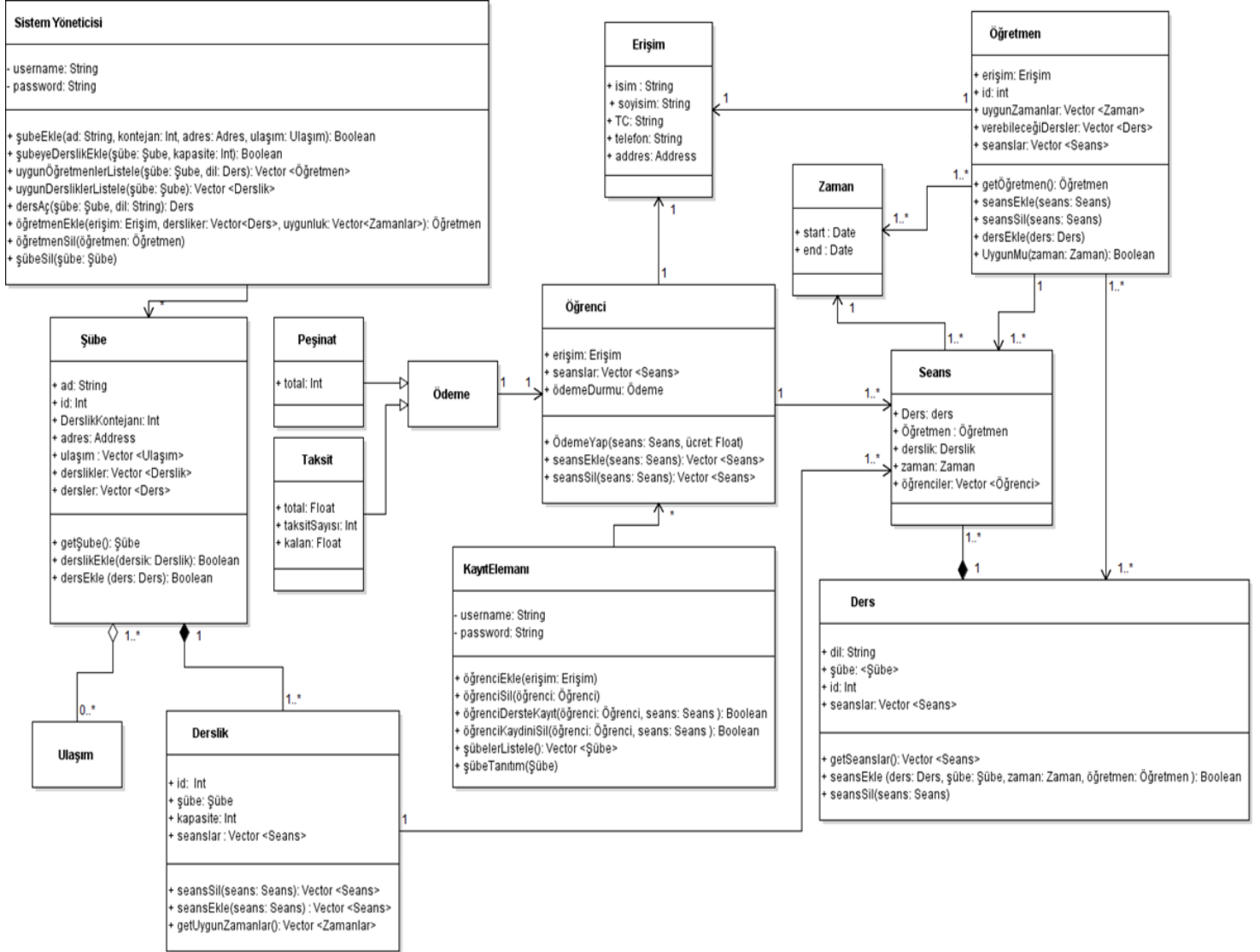


- Admin Kullanım Şeması



3. Nesneye Dayalı Modelleme

3.1 UML Sınıf Semaları

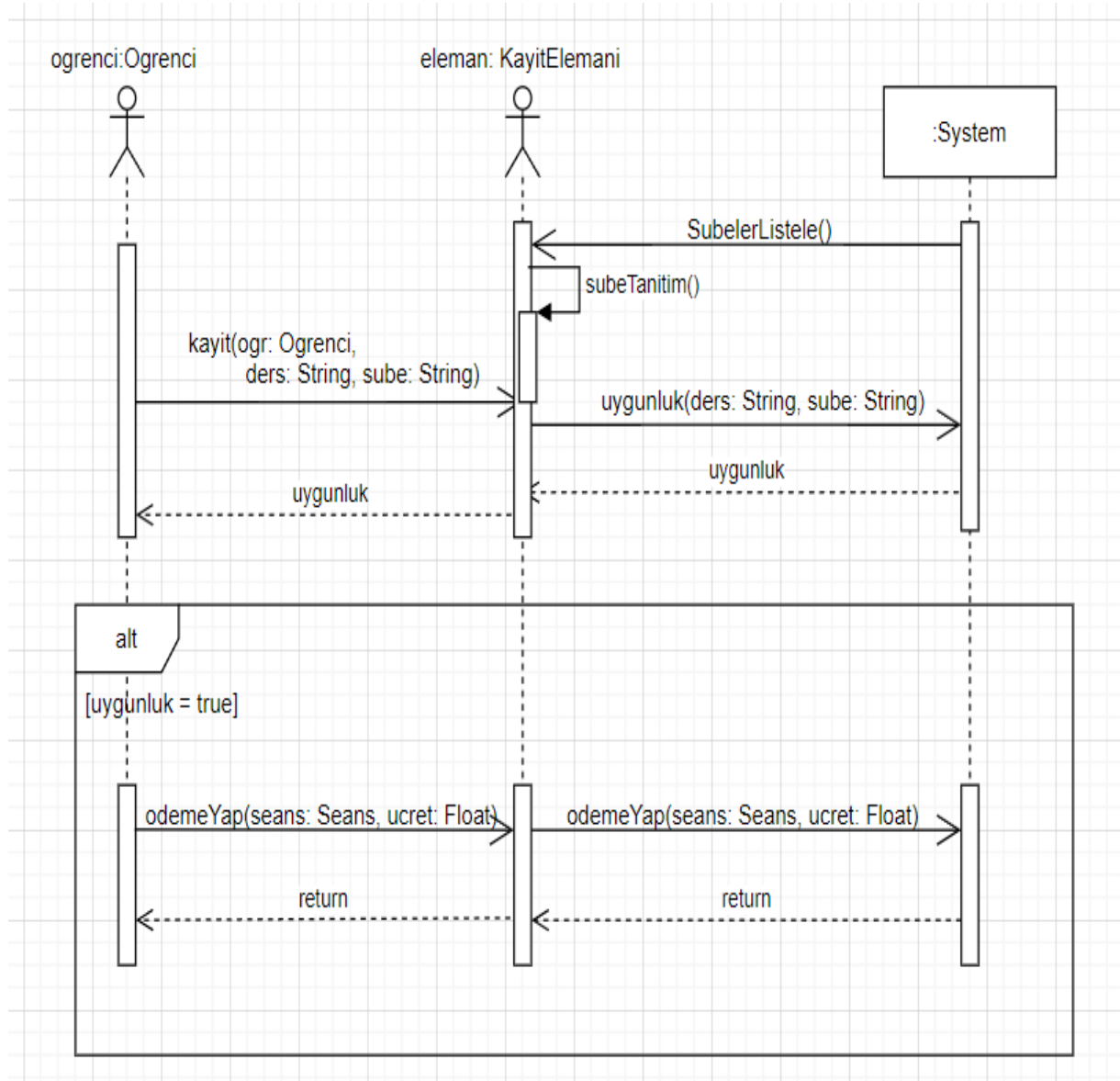


4. Nesneye Dayalı Tasarım

Tasarım; sıralama (sequence), etkinlik (activity) ve durum (state) diyagramları ile gösterilmiştir.

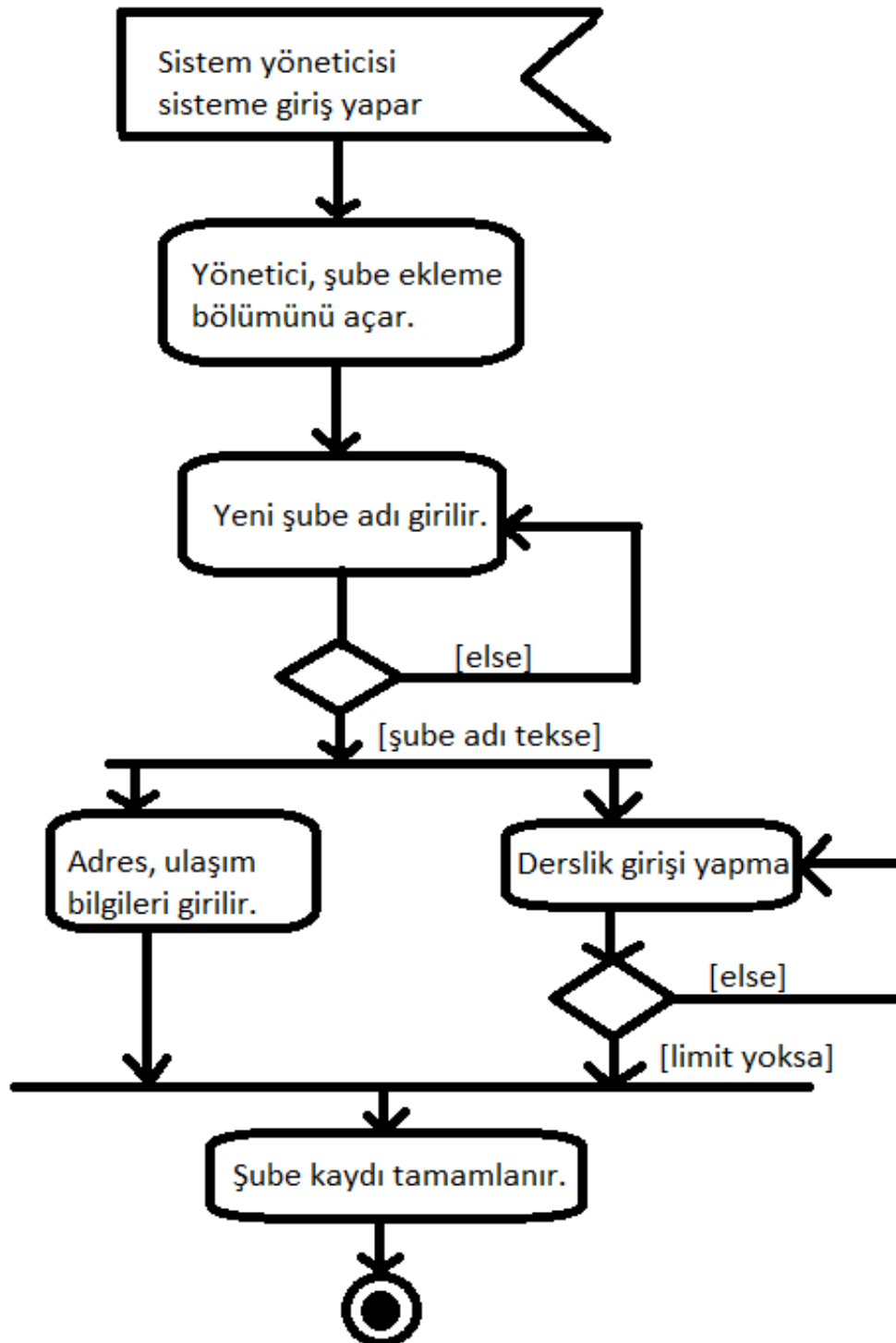
4.1 UML Sıralama Şeması (Sequence Diagram)

- Öğrenci Kayıt Sequence Diagram



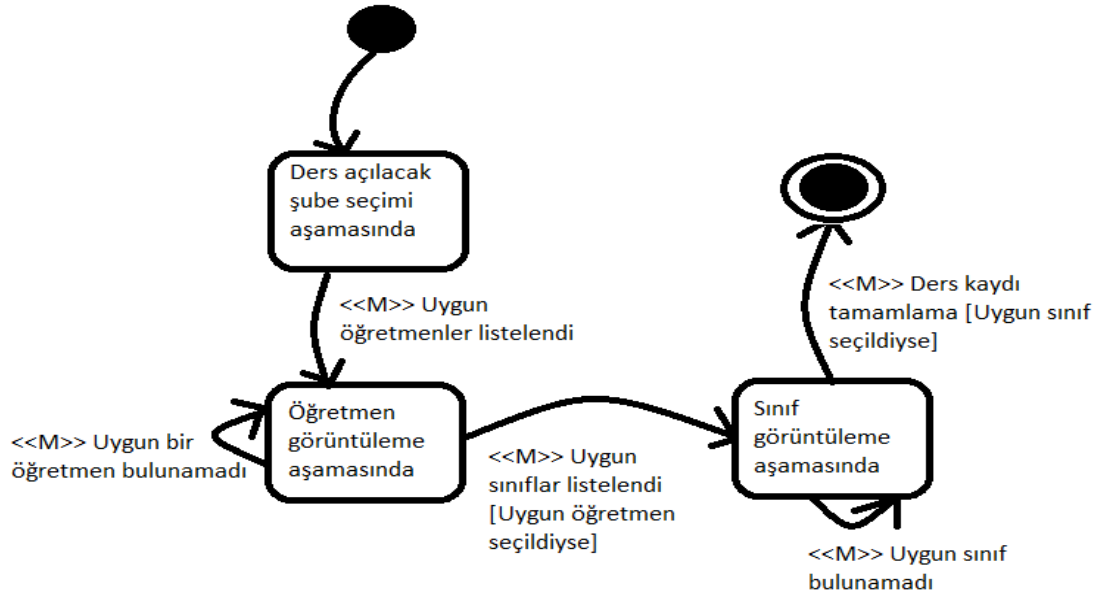
4.2 UML Etkinlik Şeması (Activity Diagram)

- Şube Girişi Activity Diagram



4.3 UML Durum Diyagramı (State Diagram)

- Ders Açma State Diagram



5. Kullanıcı Arayüzleri

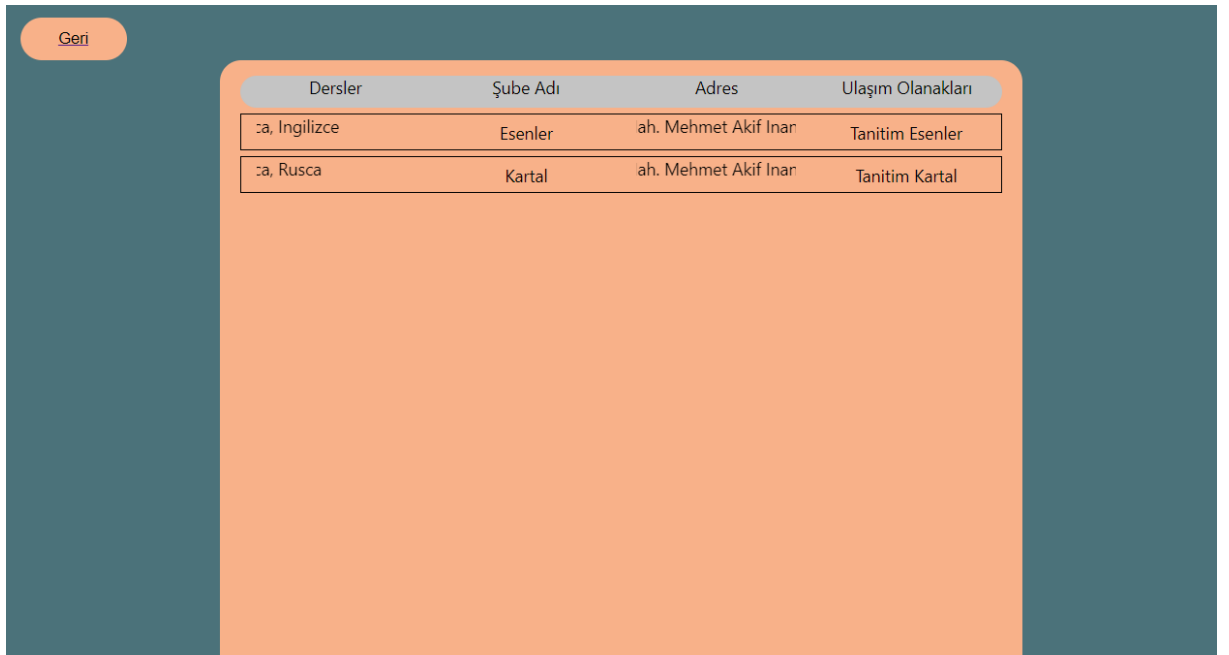
- Giriş Ekranı

The login interface is displayed on a dark teal background. It features two orange input fields for 'Kullanıcı Adı' (Username) and 'Şifre' (Password). Below these fields is a green 'Giriş Yap' (Login) button.

- Ana Ekran



- Şube Listele Ekranı



- Şube Listele Pop-Up Ekranı

Geri

Dersler	Şube Adı	Adres	Ulaşım Olanakları
Fransızca, İngilizce	Esenler	İsmet Akif Inan Cad. No: 10	Tanitim Esenler
Fransızca	Esenler	İsmet Akif Inan Cad. No: 10	Tanitim Kartal

Derslerin Seansları

İngilizce / cuma / From 12:05:00 To 13:05:00 / 6000TL

İngilizce / Persembe / From 16:05:00 To 17:05:00 / 6000TL

Fransızca / Sali / From 14:05:00 To 15:05:00 / 3500TL

- Öğrenci Ekle Ekranı

Geri

Ad

Soyad

T.C. No. 10 digits

Telefon No. 10 digits

Adres

Kaydet

- Öğrenciyi Kursa Ekle Ekranı

[Geri](#)

[Kaydet](#)

- Öğrenci Listele Ekranı

[Geri](#)

İsim	Soyisim	T.C. No.	Ders id	Ödeme Bilgisi
Utku	Magemizoglu	43512373898	1	Odendi
Rayene	Bech	1234568956	5	5 taksit var
Hakan	Kılıç	7645983412	5	6 taksit var

- Öğrenci Sil Ekranı

Geri

T.C. No.

Sil

- Taksit Al Ekranı

Geri

T.C. No.

Taksit Al

6. Birim Testi Sınamaları

```

/***** test for show all students *****/

/***** Ghyath: *****/

describe('Unit Tests for students operations',() => {
  /***** get all students *****/
  it('Get /show-students', async () => {
    const res = await request.get('/show-students')
    expect(res.status).toBe(200)
  })

/***** Ghyath: *****/

  /***** test for register student *****/
  it('başarılı öğrenci kaydı', async () => {
    const res = await request.post('/register-student')
      .send({
        name: 'Ghyath',
        surname: 'Moussa',
        tc: 12345678982,
        phone: 23428154968,
        address: 'Istanbul'
      })
    expect(res.status).toBe(200)
  })

/***** Utku: *****/

  it('başarısız öğrenci kaydı', async () => {
    const res = await request.post('/register-student')
      .send({
        name: 'Ghyath',
        surname: 'Moussa',
        tcr: 12345678982,
        phone: 23428154968,
        adresse: 'Istanbul'
      })
    expect(res.status).toBe(500)
  })
})
```



```

/***** Utku: *****/

/***** test for registerstudent *****/
it(' başarılı öğrenci silme', async () => {
  const res = await request.post('/delete-student')
    .send({
      tc: 12345678982,
    })
  expect(res.status).toBe(200)
})

/***** Hakan: *****/

it(' başarısız öğrenci silme', async () => {
  const res = await request.post('/delete-student')
    .send({
      tc: 23334,
    })
  expect(res.status).toBe(500)
})

/***** Hakan: *****/

describe('login', function() {
  it('responds with elaman_id', async function() {
    const reponse = await request.post('/login')
      .send({
        username: "kayitelemanı1",
        password: "kayitelemanısifre1",
      })
    expect(reponse.status).toBe(200)
  });
});

/***** Reyene: *****/

describe('hatalı login', function() {
  it('responds with elaman_id', async function() {
    const reponse = await request.post('/login')
      .send({
        username: "hatalı",
        password: "hatalı",
      })
    expect(reponse.status).toBe(404)
  });
});

```

```

/***** Reyene: *****/

describe('show courses ', function() {
  it('Get /show-courses', async () => {
    const res = await request.get('/show-courses')
    expect(res.status).toBe(200)
  })
});

/***** Talha: *****/

describe('Şübeler unit testi ',() => {
  it('bütün şübeler listele',async () =>{
    const res = await request.get('/show-branches')
    expect(res.status).toBe(200)
  })
})

/***** Talha: *****/

describe('Şübedeki dersleri göster',() => {
  it('Şübedeki dersleri göster', async () =>{
    const res = await request.post('/show-languages-branch')
    .send({
      branch_id:1
    })
    expect(res.status).toBe(200)
  })
})

```

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	64.1	45.23	53.57	64.1	
api	100	100	100	100	
server.js	100	100	100	100	
api/app/config	85.71	50	100	85.71	
db_config.js	85.71	50	100	85.71	15
api/app/controllers	53.38	42.1	51.85	53.38	
...chController.js	46.87	30	50	46.87	...27,38-39,51-58
courseContoller.js	81.81	50	100	81.81	10-11
loginController.js	76.92	75	66.66	76.92	7-8,23
...ntController.js	46.77	40	42.85	46.77	...0,80-81,96-103
api/app/routes	100	100	100	100	
route.js	100	100	100	100	

Test Suites: 5 passed, 5 total
 Tests: 11 passed, 11 total
 Snapshots: 0 total
 Time: 8.214 s, estimated 11 s
 Ran all test suites.