

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK – ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM3510 – YAZILIM MÜHENDİSLİĞİ

PROJE

Derman Poliklinik Bilgi Sistemi

Öğrenci No: 20011004, 21011004, 21011005, 21011006, 21011055

Ad-Soyad: Furkan KOÇAL, Yavuz ÇETİN, Ahmed Asım SEVİMLİ, Abdullah Emirhan NERGİZ,
Emirhan ÖZSARAY

E-Mail: furkan.kocal@std.yildiz.edu.tr, yavuz.cetin1@std.yildiz.edu.tr,
asim.sevimli@std.yildiz.edu.tr, emirhan.nergiz@std.yildiz.edu.tr,
emirhan.ozsaray@std.yildiz.edu.tr

Öğretim Görevlisi: Mehmet Sıddık Aktaş

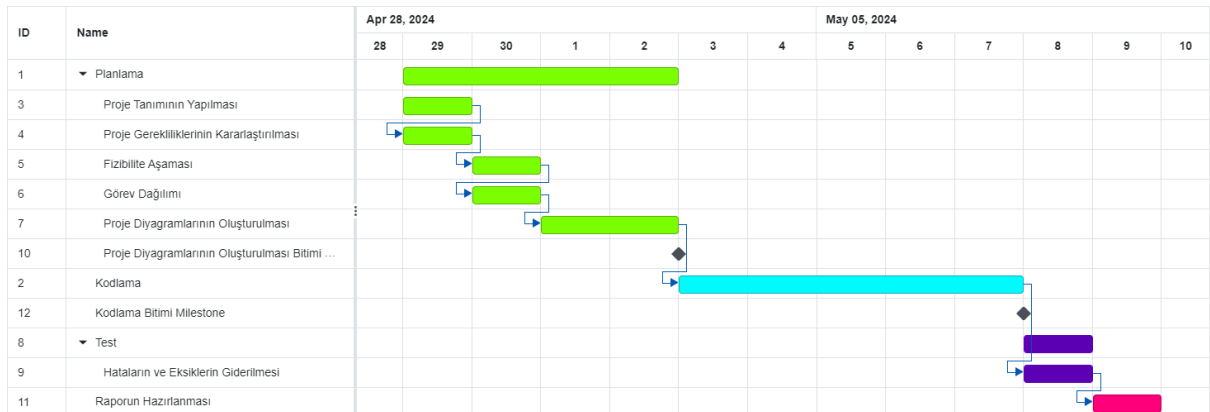
Proje Alan Tanımı

Derman Polikliniği için hasta takip ve doktor randevu işlemleri için bir bilgi sistemi geliştirilecektir. Bu sistem, poliklinikte bulunan kliniklerin ve doktorların yönetimini sağlayacaktır. Her klinikte en az bir doktor bulunmakta olup, muayene süreleri 30 dakikadır. Sistem; hasta kayıtlarının tutulmasını, randevu alma işlemlerini ve muayene sürecinin yönetimini içerecektir. Hasta, polikliniği telefonla arayarak randevu talebinde bulunacak ve görevli, uygun bir tarih ve doktor belirlemek için sistemi kullanacaktır. Arayan hasta sistemin kayıtlarında bulunmuyorsa hasta kayıt görevlisi, yeni bir hasta kaydı açar. Doktor, muayene öncesinde hastanın bilgilerini sistemden sorgulayacak ve muayene sonrasında tedavi ve gerekirse rapor/reçete düzenleme işlemlerini gerçekleştirecektir. Ödeme işlemleri, muayene sonrasında vezneye geçilerek nakit veya kredi kartı ile yapılabilir ve sistem, sigorta durumunu sorgulayarak indirim uygulayabilecektir.

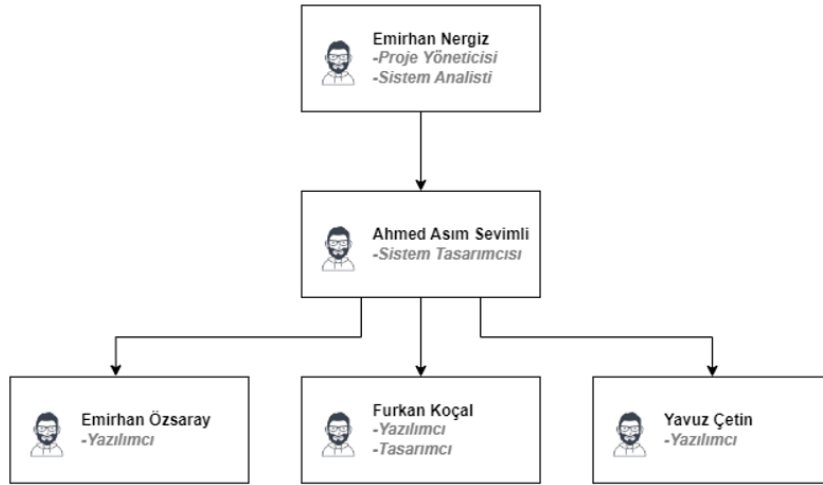
Kabuller ve Kısıtlar

- Sistemde çevik metodoloji kullanılacaktır.
- Sistem Java dili kullanılarak yazılacaktır.
- Arayüz Java Swing ile yapılacaktır.
- Sistemde bir veri tabanı yönetim sistemi kullanılacaktır.
- Muayene fiyatını doktorun belirlemesi.
- Doktorun muayene başına sadece birer adet sevk kağıdı, reçete ve rapor verebilmesi
- Kredi kartı ve sigorta bilgileri veri tabanında tablolarda tutulacaktır.
- Muayene başına her hasta için 30 dakika verilecektir.
- Klinik ekleyebilmek için ayrıca bu kliniğe atanacak bir doktorun da eklenmesi gerekmektedir.
- Nakit ödemenin sistem dışında olması ve görevlinin ödeme yapıldığında nakit ödeme butonuna basması ödemenin yapıldığı anlamına gelmektedir.

Gantt Diyagramı



Ekip Organizasyon Şeması



Görev Dağılımları

Proje Yöneticisi

- Proje planlaması
- Toplantı düzenlemesi
- Görev atama

Sistem Analisti

- Proje alan tanımı
- Proje risk tablosu oluşturma
- Kullanım senaryosu oluşturma
- Sözleşmeler

Sistem Tasarımcısı

- Tasarım sınıf diyagramı
- Zenginleştirilmiş sınıf diyagramı
- Etkinlik, sekans ve durum diyagramları tasarlama
- Kullanım senaryosu diyagramı

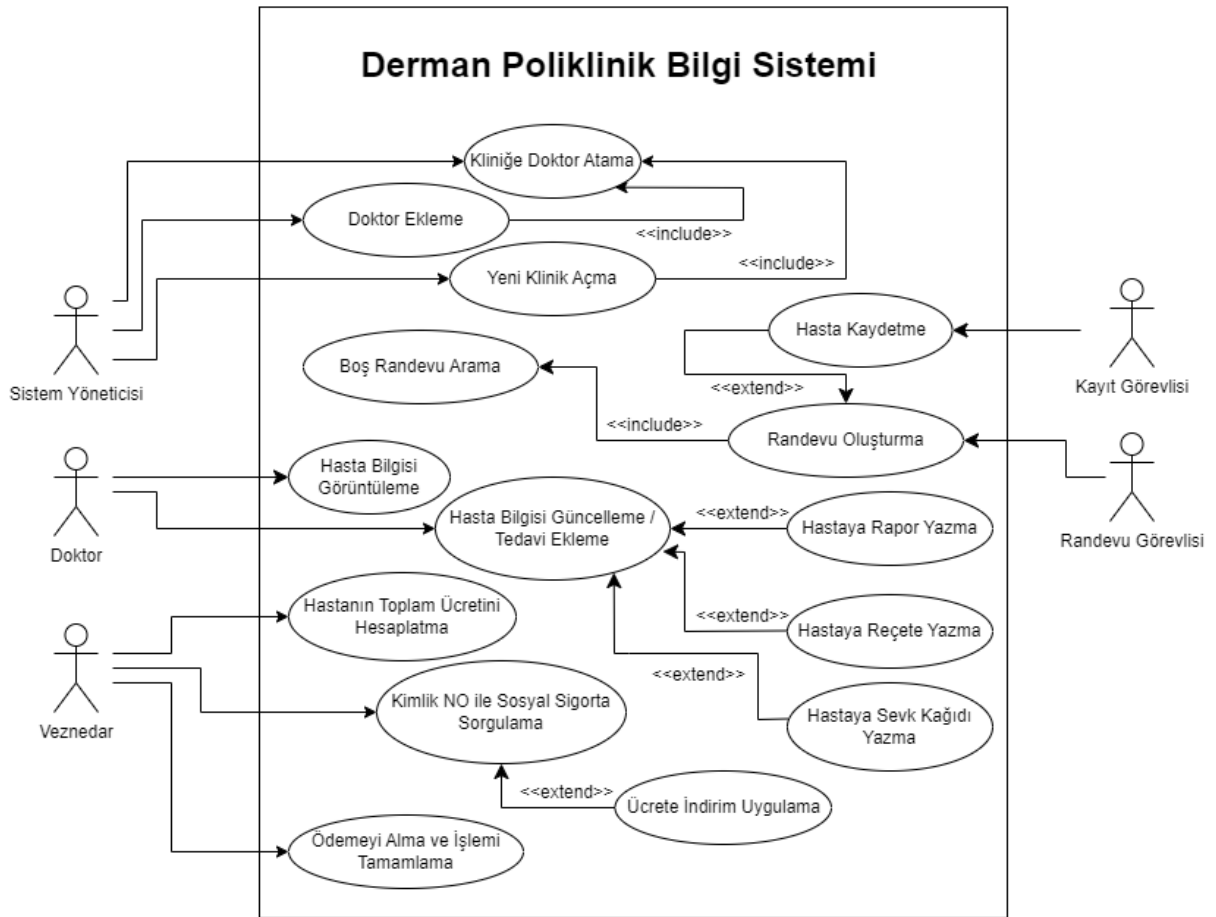
Yazılımcılar

- PostgreSQL ile veri tabanı oluşturma
- Sistem tasarımları kullanılarak Java kodu yazma
- Java Swing UI dizayn ile kullanıcı arayüzü tasarlama

Risk Tablosu

Risk ID	Adı	Türü,Grubu	Etkisi	Olasılık
01	Organizasyon	Proje	Orta	Orta
02	Kullanışlılık	Teknik	Büyük	Düşük
03	Doğruluk	Teknik	Orta	Düşük
04	Güvenilirlik	Teknik	Orta	Düşük
05	Araçlar	Teknik	Orta	Orta
06	İletişim Eksikliği	Proje	Büyük	Orta
07	Zaman Kısıtlılığı	Proje	Büyük	Yüksek

Kullanım Senaryosu



KULLANIM ÖYKÜSÜ: Doktor Ekleme

Birincil Aktör: Yönetici veya Yetkili Personel

İlgililer ve İlgili Alanları:

Yönetici veya Yetkili Personel: Doğru ve eksiksiz doktor bilgilerinin sisteme eklenmesini ister. Yeni doktorların kolayca tanımlanmasını ve erişilebilir olmasını bekler.

Sistem: Yeni doktor kayıtlarını doğru ve güvenilir şekilde oluşturmayı ve saklamayı amaçlar.

Ön Koşullar:

Yönetici veya yetkili personelin kimliği doğrulanır.

Son Koşullar:

Yeni doktor kaydı sisteme eklenir ve doktorun profilinin oluşturulması tamamlanır.

Kullanım Öyküsü: Doktor Ekleme

Giriş Bölümü:

- Yeni bir doktor eklemek isteyen yönetici veya yetkili personel tarafından başlatılır.
- Doktorun kimlik bilgileri ve uzmanlık alanı gibi bilgiler sağlanmalıdır.

Ana Senaryo

- 1.Yetkili personel, sisteme yeni bir doktor eklemek istediğini belirtir.
- 2.Sistem, doktorun kimlik bilgilerini (ad, soyadı, TC kimlik numarası vb.) ve uzmanlık alanını alır.
- 3.Yönetici veya yetkili personel, sağlanan bilgilerle birlikte yeni doktor kaydını oluşturur.
- 4.Yeni doktor kaydı sisteme eklenir ve doktorun profili oluşturulur.

Alternatif Senaryolar:

2.a Doktorun kimlik bilgileri eksik veya geçersizdir.

1. Sistem bir hata mesajı gösterir ve gerekli bilgilerin tamamlanması istenir.

2.b Sistemde aynı TC kimlik numarasına sahip bir doktor vardır

1. Sistem bir uyarı mesajı gösterir ve farklı bir kimlik numarası kullanılmasını istenir.

KULLANIM ÖYKÜSÜ: Randevu Oluşturma

Birincil Aktör: Hasta Kayıt Görevlisi (HKG) veya Yetkili Personel

İlgililer ve İlgili Alanları:

HKG veya Yetkili Personel: Hızlı ve etkili bir şekilde randevuların oluşturulmasını ve hasta memnuniyetinin sağlanmasını bekler.

Sistem: Randevu oluşturma sürecinde doğruluk ve güvenilirlik sağlamayı amaçlar.

Ön Koşullar:

Kullanıcı kimliğinin doğrulanmış olması

Son Koşullar:

Randevunun başarıyla kaydedilip sisteme işlenmesi

Kullanım Öyküsü: Randevu Oluşturma

Giriş Bölümü:

Yeni bir randevu oluşturmak isteyen HKG veya yetkili personel tarafından başlatılır.

Randevunun yapılacağı poliklinik ve tarih belirlenmelidir.

Ana Senaryo:

- 1.HKG veya yetkili personel, sisteme yeni bir randevu oluşturmak istediğini belirtir.
- 2.Sistem, boş randevu saatlerini ve mevcut doktorları listeler.
- 3.HKG veya yetkili personel, uygun poliklinik, doktor ve tarih için randevu oluşturur.
- 4.Sistem, randevuyu kaydeder ve randevu numarasını HKG'ye veya hastaya bildirir.

Alternatif Senaryolar:

- 1.a Hastanın kaydı mevcut değildir.
 1. HKG hastayı sisteme kaydeder ve ardından randevu oluşturulur.
- 2.a İstenen tarihte uygun randevu bulunamaz
 1. Görevli takvimde başka tarihlere tıklayarak uygun bir tarih bulur.

KULLANIM ÖYKÜSÜ: Veznedar

Birincil Aktör: Veznedar

İlgililer ve İlgili Alanları:

Veznedar: Doğru ve hızlı bir şekilde hasta ödemelerini almayı ve kaydetmeyi bekler.

Hastalar: Hızlı ve güvenilir bir şekilde ödemelerini yapmayı ve gerekli belgeleri almayı bekler.

Sistem: Ödemelerin doğru bir şekilde işlenmesini, kaydedilmesini ve gerektiğinde raporlanmasını sağlar.

Ön Koşullar:

Veznedarın ödemeyi almadan önce hastanın muayene veya hizmet aldığından emin olması

Son Koşullar:

Ödeme işleminin başarıyla tamamlanması

Giriş Bölümü:

Bir hasta, muayene sonrası ödeme yapmak için vezneye başvurur.

Ana Senaryo:

- 1.Veznedar, hastanın ödeme yapmak istediğini öğrenir.
- 2.Hasta, ödemesini nakit, kredi kartı veya banka kartı ile yapar.
- 3.Veznedar, ödemeyi sisteme girer ve ödeme tutarını kaydeder.
- 4.Sistem, ödemenin başarıyla kaydedildiğine dair bir onay mesajı gösterir.
- 5.Veznedar, hastaya ödeme makbuzunu veya faturayı verir.

Alternatif Senaryolar:

4.a Hastanın ödemesi eksik veya yanlıştır

1. Veznedar hata mesajı alır ve doğru bilgilerin girilmesini sağlar.

Sözleşmeler

Sözleşme No: 1 – Doktor İşe Alımı

İşlem: addDoctor(name:String, surname:String, clinicID:Integer, username:String, password: String)

Çapraz Başvurular:

- Klinik ilişkilendirme: Yeni doktorun bir kliniğe atanması gerektiğinden “Klinik Oluşturma” veya “Klinik Değişikliği” gibi işlemlerle ilişkilendirilebilir.

Ön Koşullar:

- Doktorun zaten sistemde bulunmaması

Son Koşullar:

- addDoctor işlemi başarıyla gerçekleştirilmiştir.
- Yeni doctor, isim ve soyisim bilgileriyle sisteme eklenmiştir.
- Doctor, belirtilen klinik ID ile ilişkilendirilmiştir.
- Doktorun sisteme giriş yapabilmesi için kullanıcı adı ve şifre oluşturulmuştur.

Sözleşme No: 2 – Hasta Kaydı Oluşturma

İşlem: addPatient(tcNO: String, name: String, surname: String)

Çapraz Başvurular:

- Hasta Randevusu: Hasta kaydı oluşturulduktan sonra hastaya randevu verilebilir.
- Tedavi İşlemi: Hastanın tedavi edilmesi için hastanın bilgilerine ihtiyaç duyulacaktır.

Ön Koşullar:

- Hasta, sisteme daha önce kaydedilmemiş olmalıdır.
- TC kimlik numarası benzersiz olmalıdır.

Son Koşullar:

- addPatient işlemi başarıyla tamamlanmıştır.
- Yeni hasta, sağlanan bilgilerle sisteme eklenmiştir.
- Hasta bilgileri doğrulanmış ve sisteme kaydedilmiştir.
- Hasta randevu alma ve tedavi için hazır hale gelmiştir.

Sözleşme No: 3 – Randevu Oluşturma

İşlem: randevuOluştur(hastaTc:String, tarih: String, Saat: String, doktorID: Integer, klinikID: Integer)

Çapraz Başvurular:

- Klinik Seçimi: Randevu oluşturulurken, belirli bir kliniğe atanması gerektiğinden, klinik seçimi işlemle ilişkilendirilebilir. Bu işlem, hastanın randevu talebinde bulunduğu kliniği belirler.
- Doktor Seçimi: Randevu oluşturulurken, belirli bir doktora atanması gerektiğinden, doktor seçimi işlemle ilişkilendirilebilir. Bu işlem, hastanın randevu talebinde bulunduğu doktoru belirler.

Ön Koşullar:

- Hasta bilgilerinin sisteme kaydedilmiş olması.
- Belirtilen tarihte ve saatte belirtilen doktorun ve kliniğin uygun olması.

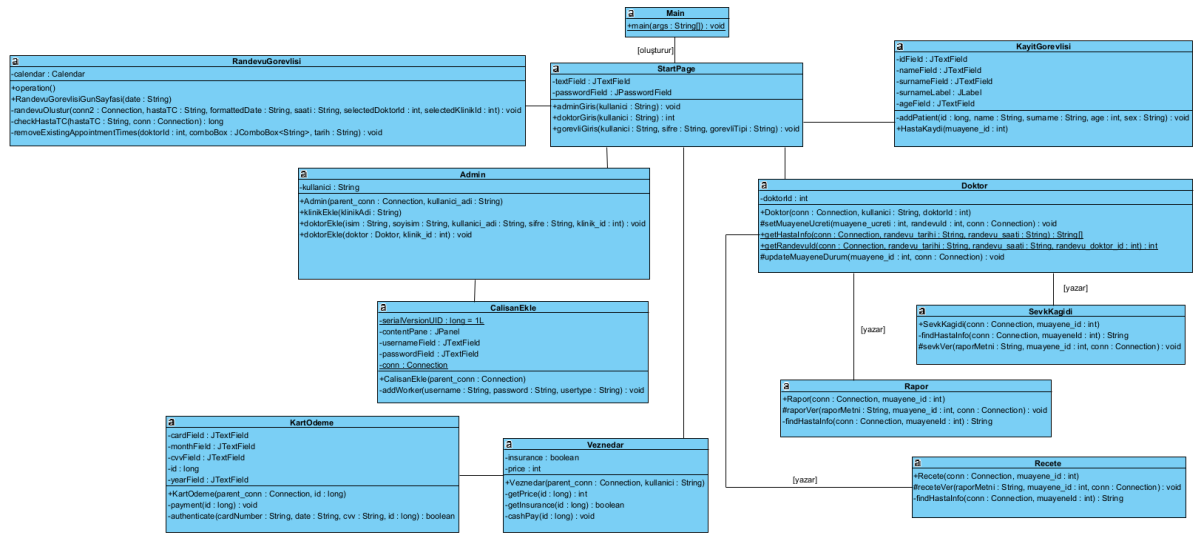
Son Koşullar:

- Randevu başarıyla oluşturulmuştur.
- Randevu, belirtilen hasta, doktor ve klinik ile ilişkilendirilmiştir.
- Randevu tarihi ve saati sisteme kaydedilmiştir.

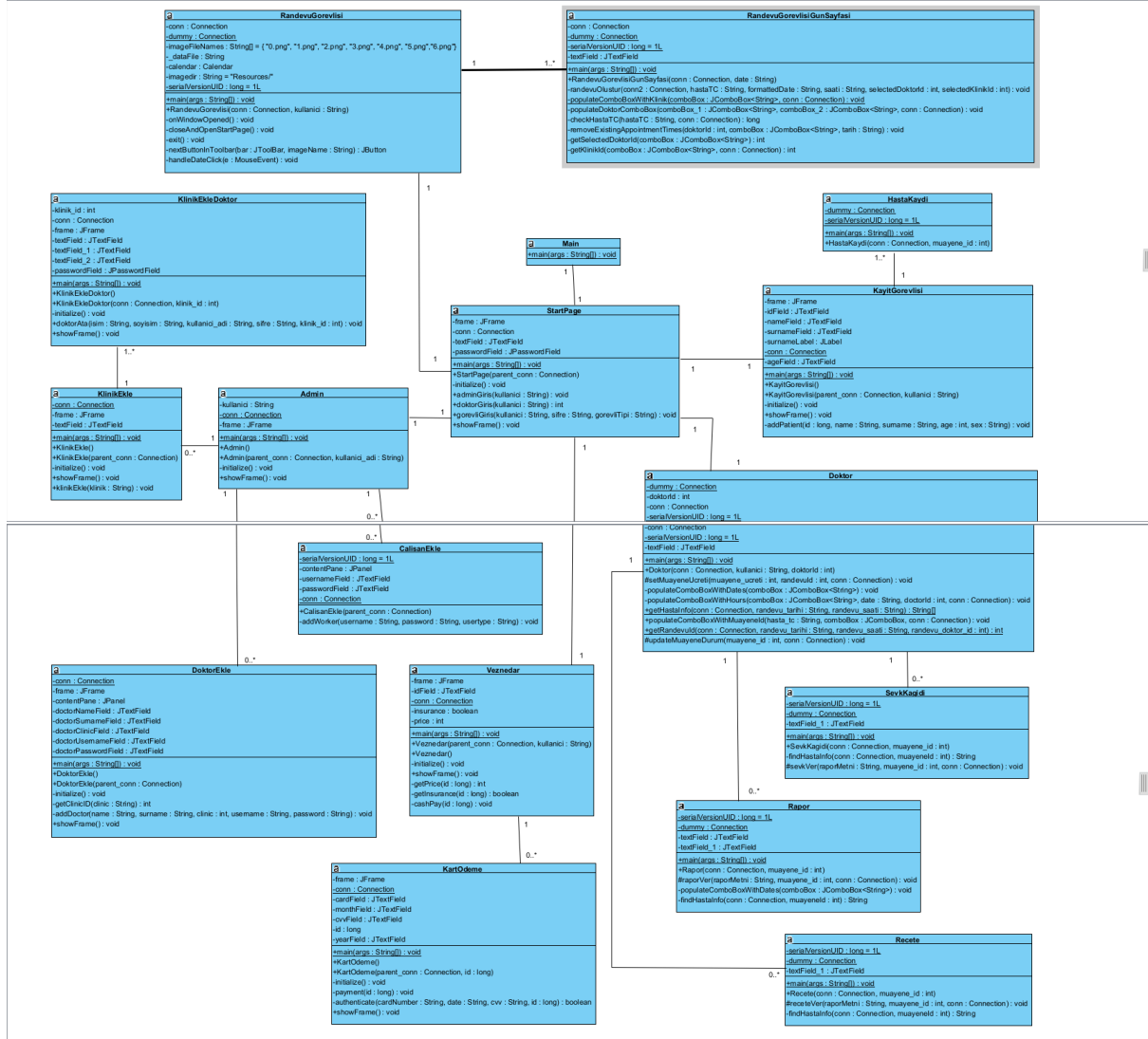
İzlenebilirlik Tablosu

Amaç	Girdiler	Beklenen Çıktılar	Test Sonuçları
Yeni klinik ekleme	Klinik bilgileri	Yeni klinikte en az bir doktor bulunmalıdır.	Başarılı
Doktor ekleme	Doktor bilgileri	Veri tabanına doktorun eklenmesi	Başarılı
Çalışan ekleme	Çalışan bilgileri	Veri tabanına çalışanın eklenmesi	Başarılı
Hasta kaydı açma	Hasta bilgileri	Veri tabanında hasta kaydının oluşturulması	Başarılı
Veri tabanı bağlantısı	PostgreSQL kullanıcı ismi ve şifresi	Veri tabanı bağlantısının kurulması	Başarılı
Hasta bilgisini veri tabanından getirme	Bağlantı bilgisi ve T.C. Kimlik Numarası	Veri tabanının boş olmaması	Başarılı
Sigorta kontrol etme	Bağlantı bilgisi ve T.C. Kimlik Numarası	Sigorta durumu	Başarılı
Kredi kartı ödeme	Kredi kartı bilgileri	Ödeme miktarının karttan çekilmesi	Başarılı
Reçete verme	Hastaya yazılan ilaçlar	Hastanın kaydında reçetenin görülebilmesi	Başarılı

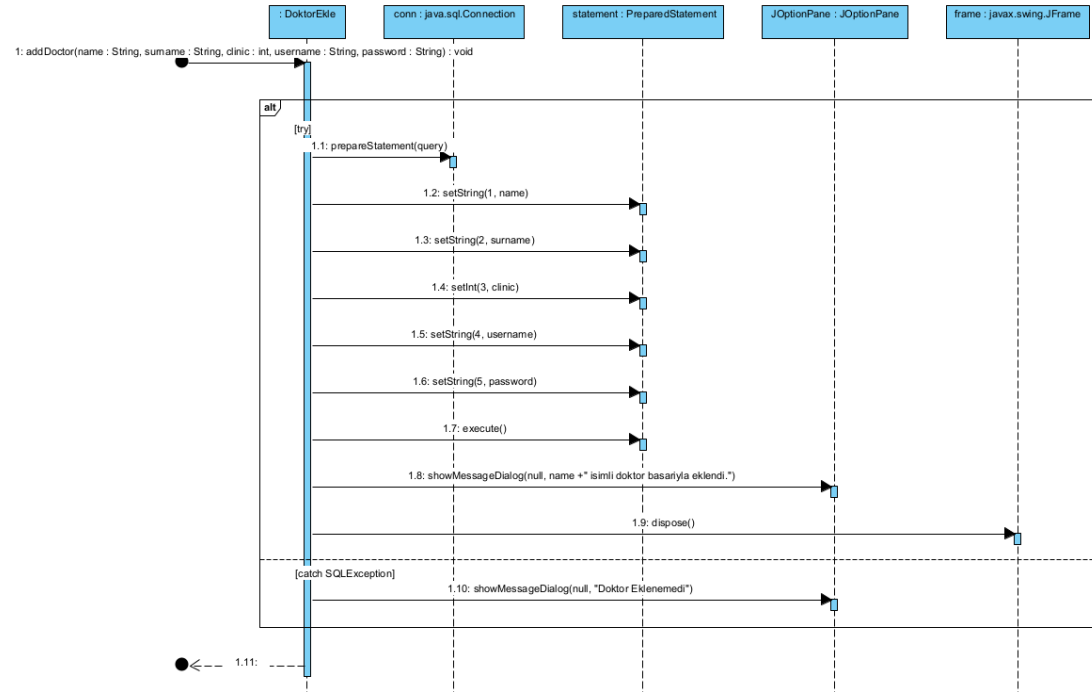
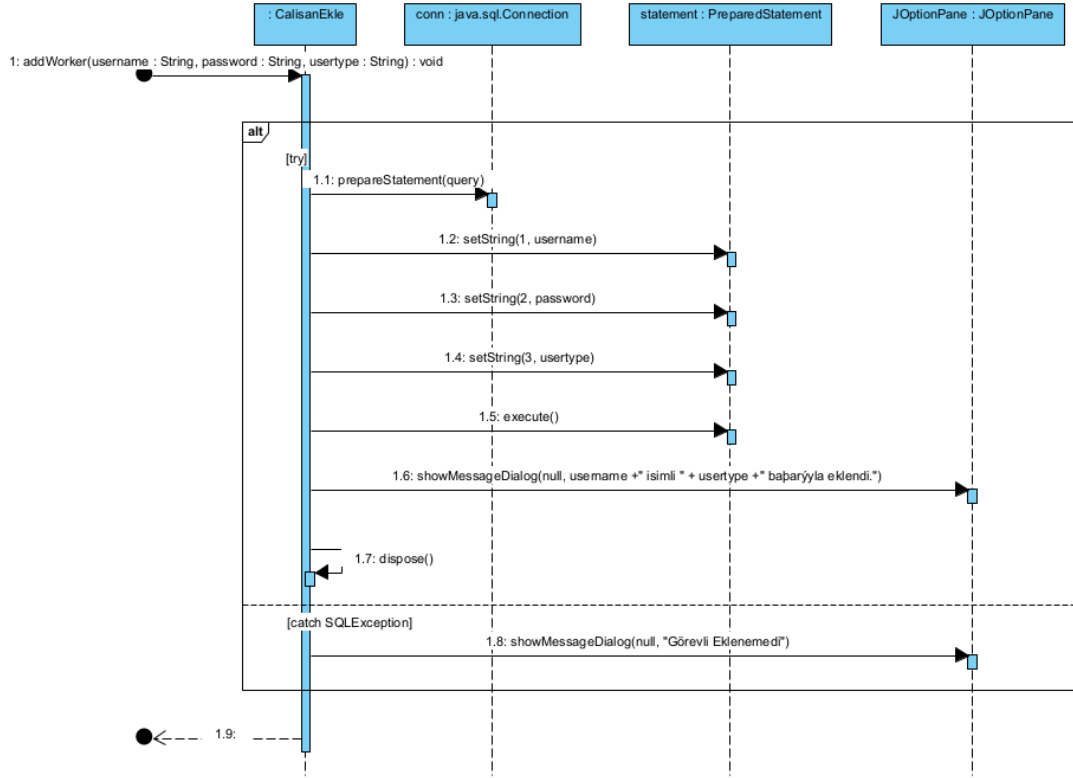
Sınıf Diyagramı

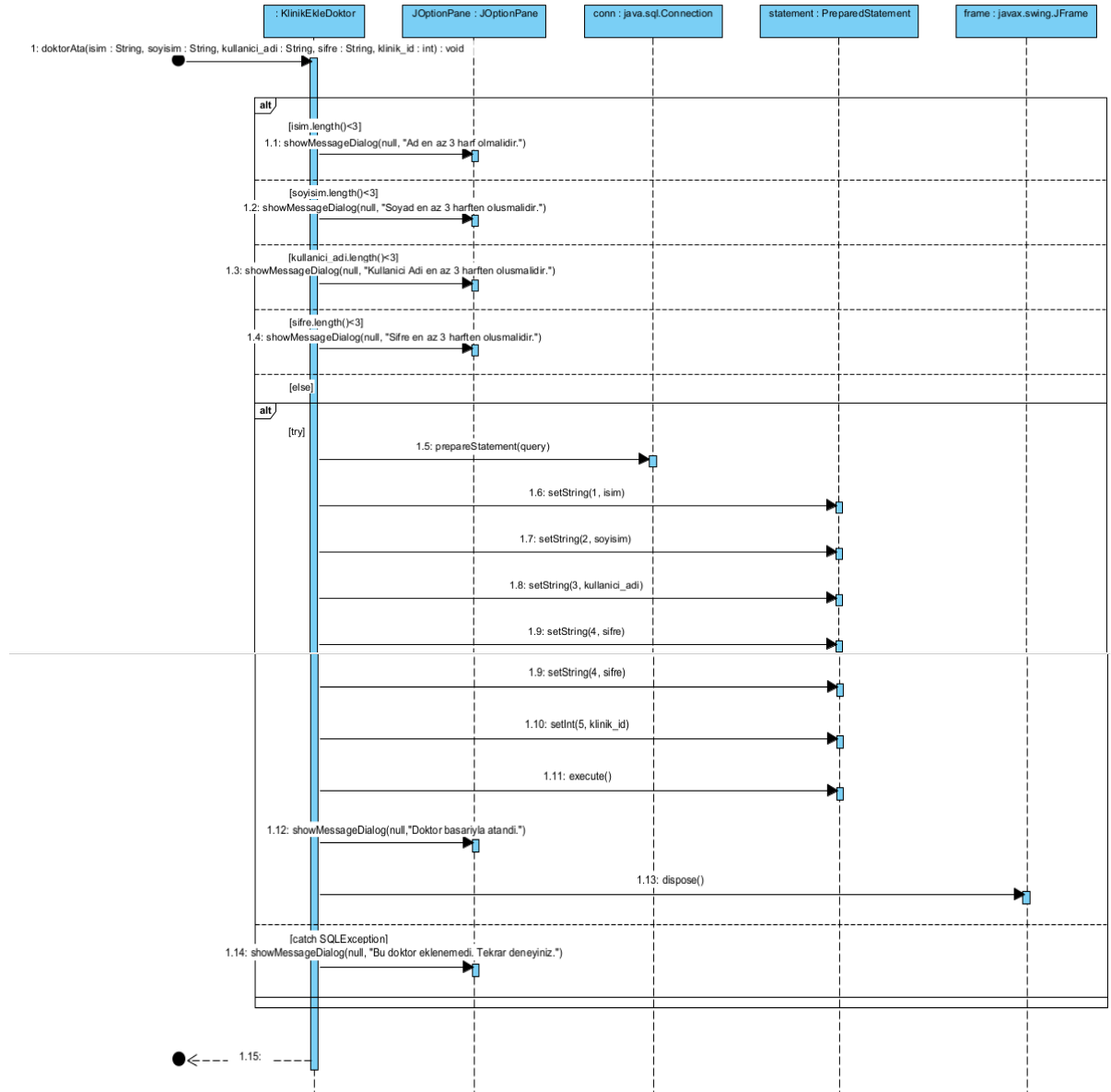


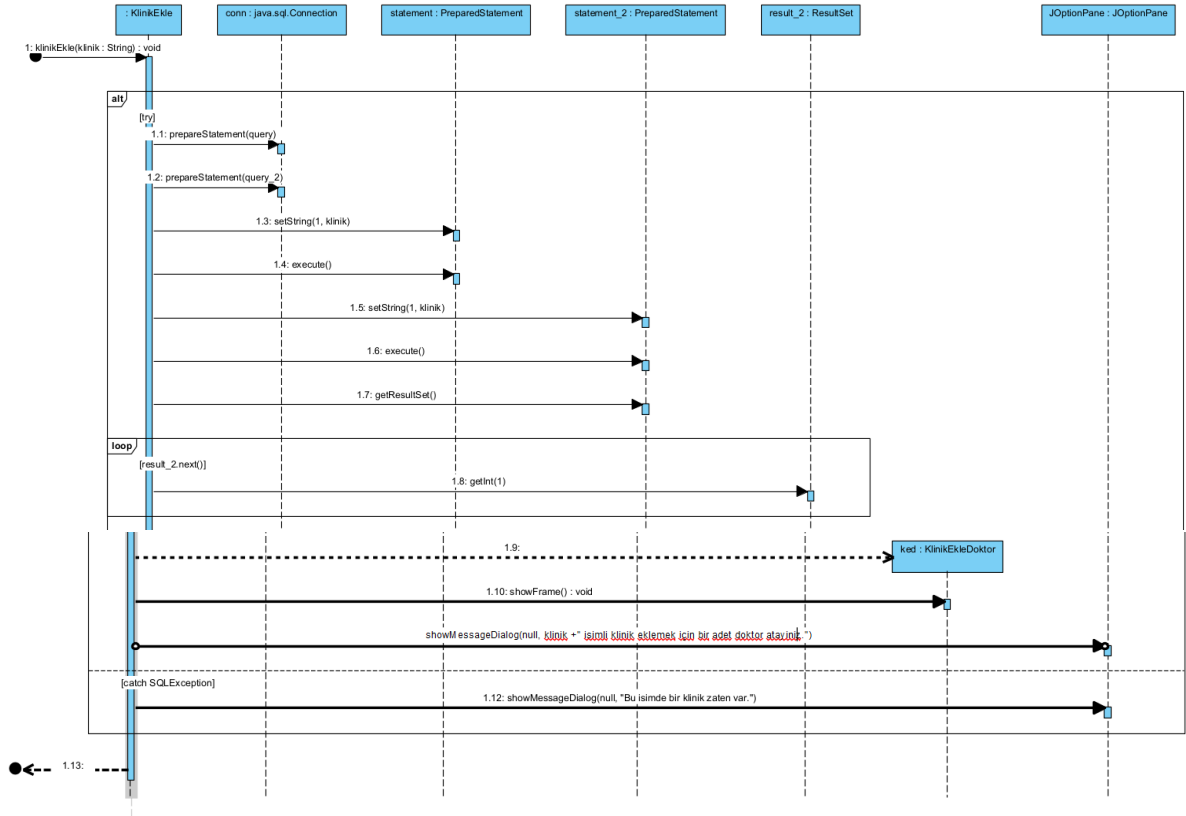
Zengin Sınıf Diyagramı



Sıralama Diyagramı

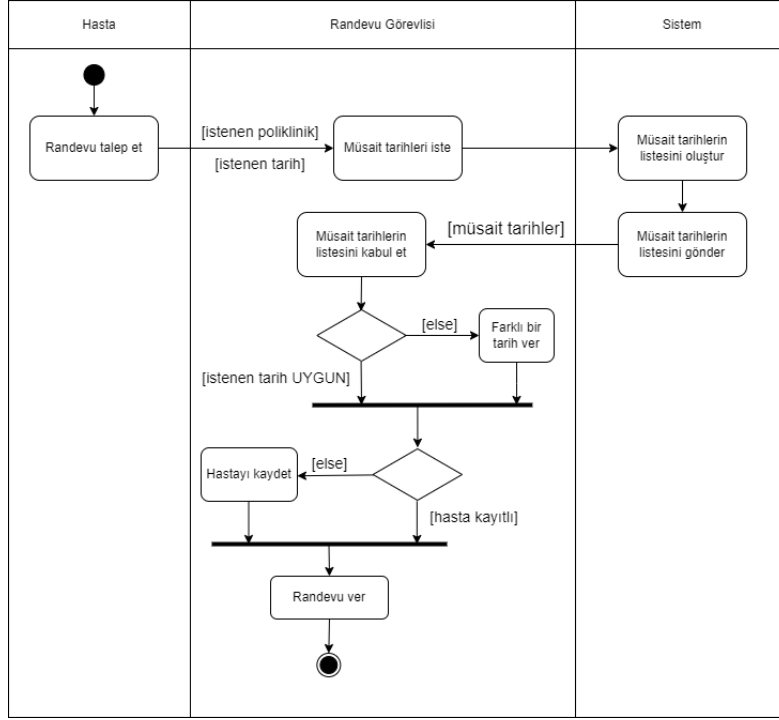




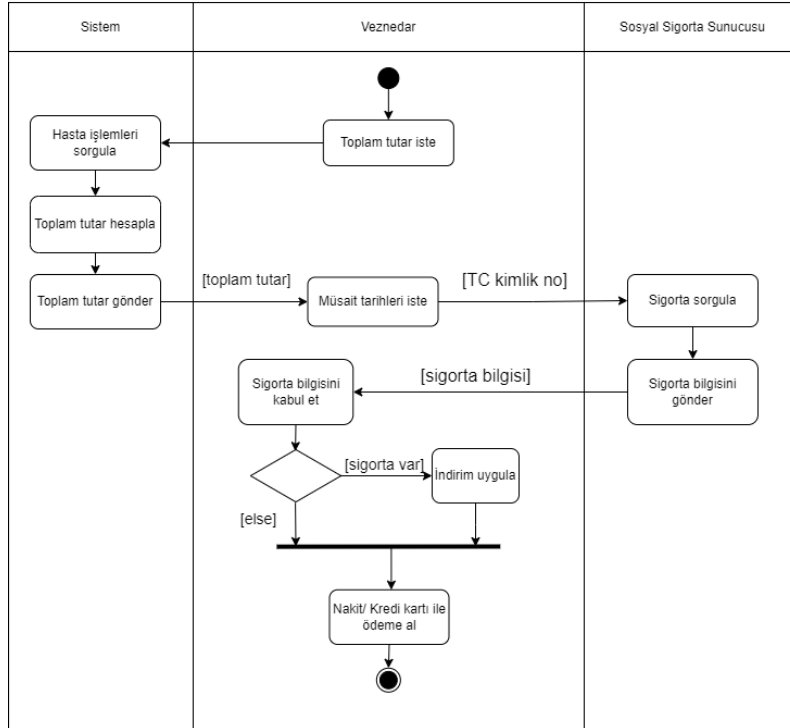


Etkinlik Diyagramı

Randevu Sistemi

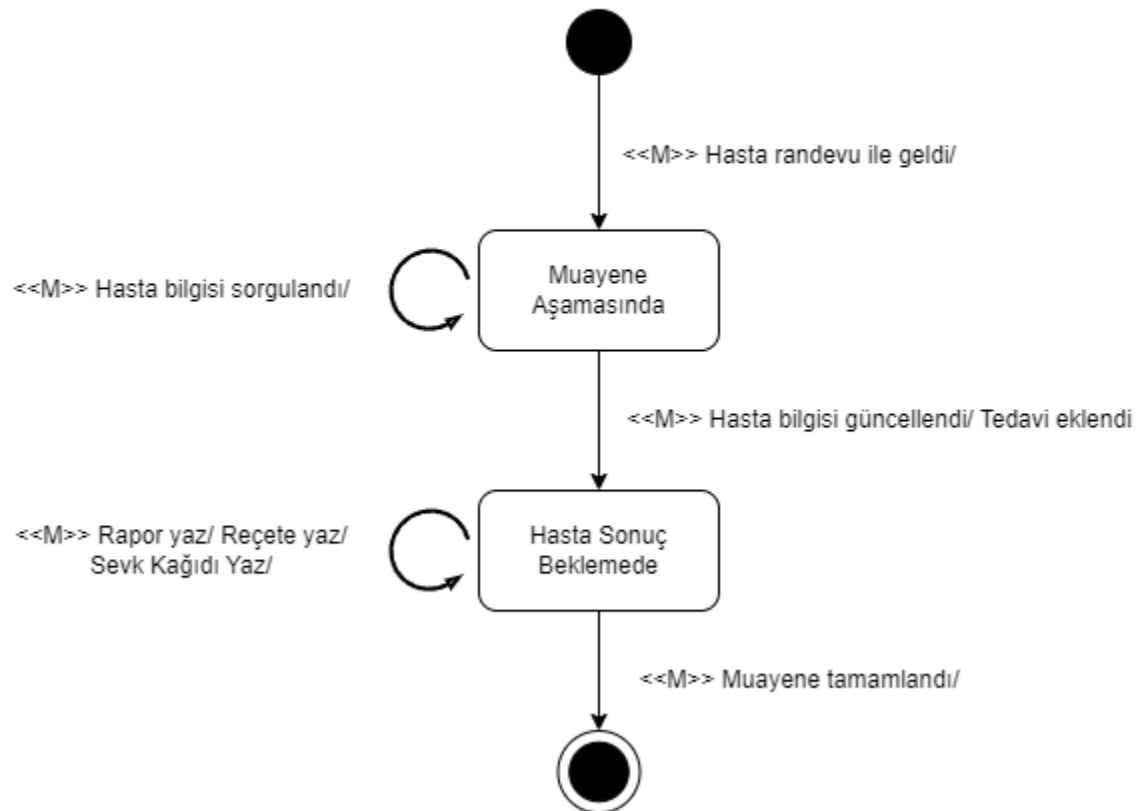


Ödeme Sistemi



Durum Diyagramı

Muayene Sistemi



Birim Testi Sınamaları

Furkan:

```
@Test
public void testGetClinicIDFromMethod() throws SQLException {
    int clinicID = objUnderTest.getClinicID("göz");
    assertEquals(1, clinicID);
}

@Test
public void testStartPage1() throws SQLException{
    int doktorID = doktorIDTest.doktorGiris("furkan");
    Assert.assertNotNull(doktorID);
}
```

Yavuz:

```
@Test
public void testStartPage2() throws SQLException{
    Assert.assertNotNull(doktorIDTest.gorevliGiris("yavuz", "123", "Veznedar"));
}

@Test
public void testStartPage3() throws SQLException{
    Assert.assertNotNull(doktorIDTest.adminGiris("yavuz"));
}
```


Ahmet Asım:

```
@Test
public void testVeznedar() throws SQLException{
    int price = veznedarTest.getPrice(1);
    Assert.assertNotNull(price);
}

@Test
public void testVeznedar2() throws SQLException{
    boolean insurance = veznedarTest.getInsurance(12345678910L);
    Assert.assertNotNull(insurance);
}
```

Abdullah Emirhan:

```
@Test
public void testVeznedar3() throws SQLException{
    Assert.assertNotNull(veznedarTest.getPrice(12345678910L));
}

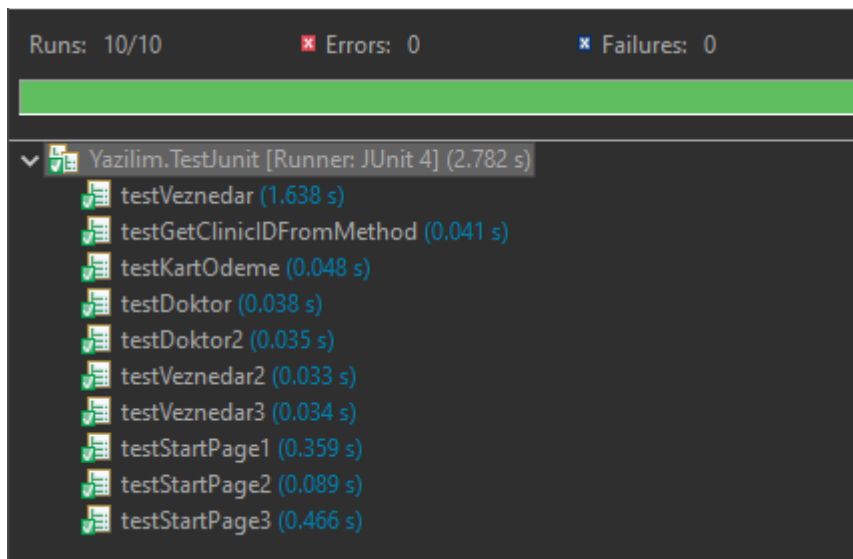
@Test
public void testDoktor2() throws SQLException {
    Assert.assertNotNull(Doktor.getRandevuId(conn, "2024-05-10", "09:30", 1));
}
```

Emirhan:

```
@Test
public void testKartOdeme() throws SQLException {
    Assert.assertNotNull(kartOdemeTest.authenticate("1111222233334444","01/24","111", 12345678910L));
}

@Test
public void testDoktor() throws SQLException {
    Assert.assertNotNull(Doktor.getHastaInfo(conn, "2024-05-10", "09:30"));
}
}
```

Sonuçlar:



The screenshot shows the JUnit test results in an IDE. At the top, it displays 'Runs: 10/10', 'Errors: 0', and 'Failures: 0'. Below this, a green progress bar indicates that all tests passed. The test suite is 'Yazilim.TestJUnit [Runner: JUnit 4] (2.782 s)'. The list of tests and their execution times are:

- testVeznedar (1.638 s)
- testGetClinicIDFromMethod (0.041 s)
- testKartOdeme (0.048 s)
- testDoktor (0.038 s)
- testDoktor2 (0.035 s)
- testVeznedar2 (0.033 s)
- testVeznedar3 (0.034 s)
- testStartPage1 (0.359 s)
- testStartPage2 (0.089 s)
- testStartPage3 (0.466 s)