2020-2021 BAHAR DÖNEMİ

YAZILIM MÜHENDİSLİĞİ DÖNEM PROJESİ RAPORU

ŞİFA POLİKLİNİĞİ BİLGİ SİSTEMİ

16011016 Ahmet ELGÜN
16011055 Orkun AVCI
17011615 Said EROĞLU
18011071 Elif Yağmur DURAN
20501066 Mohamad SAWAS

İÇİNDEKİLER

1) PROJE TANIMI

a) Proje alan tanımı

Şifa polikliniği sistemi, hasta takip ve doktor randevu gibi işlemlere yönelik ihtiyacı sağlayan bir bilgi sistem programıdır.

Sistem hasta kaydı, muayene rezervasyonu, muayene geçmişinin sorgulanması, ücret hesabı ve ödemeye olanak sağlar.

b) Kabul ve kısıtlar

- Poliklinikte göz, üroloji, ortopedi, psikiyatri klinikleri bulunmakta ve ileride ekleme yapılması düşünülmektedir.
- Muayene süreleri 30 dk olarak belirlenmiştir.
- Poliklinik rezervasyon ile hasta kabul etmektedir.
- Rezervasyon kayıt görevlisi tarafından hastanın isteğine göre gerçekleştirilir. Eğer hastanın istediği saat uygun değilse farklı bir saate yönlendirir.
- Hastanın sisteme kaydı yoksa kayıt görevlisi tarafından sisteme kaydedilir.
- Doktor, hastanın muayene geçmişini sorgulayabilme ve muayene sonrası güncelleyebilme yetkisine sahiptir.
- Ödeme işlemi, veznede yapılır. Veznedar sistemden hastayı sorgular ve ödemeyi hesaplattırır.

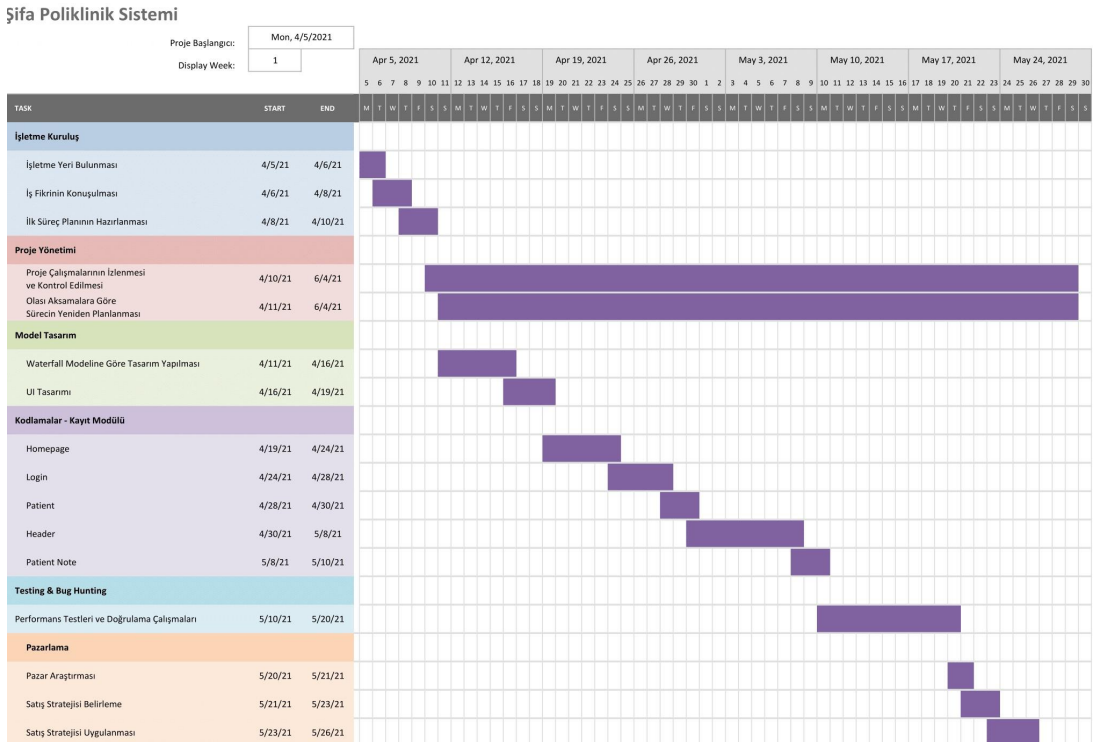
c) Proje iş-zaman çizelgesi

Şifa Poliklinik Sistemi

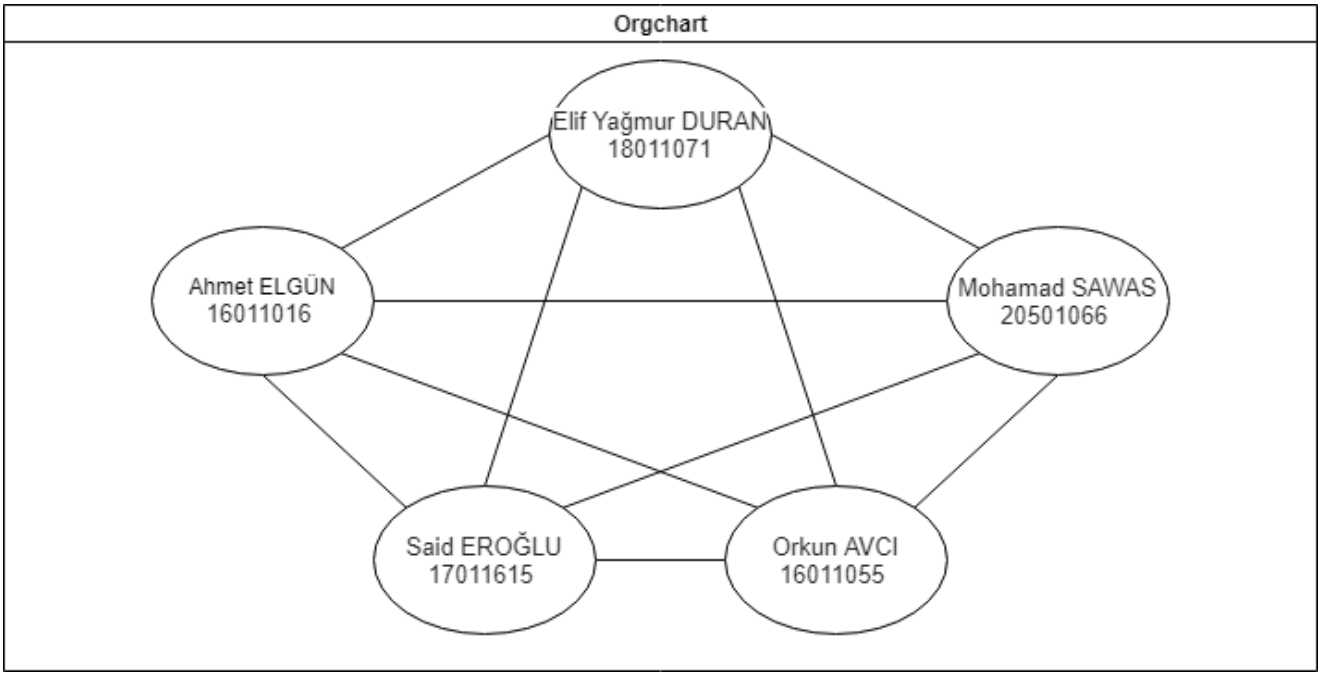
Proje Başlangıcı: Mon, 4/5/2021

Display Week: 1

TASK	START	END
İşletme Kuruluş		
İşletme Yeri Bulunması	4/5/21	4/6/21
İş Fikrinin Konuşulması	4/6/21	4/8/21
İlk Süreç Planının Hazırlanması	4/8/21	4/10/21
Proje Yönetimi		
Proje Çalışmalarının İzlenmesi ve Kontrol Edilmesi	4/10/21	6/4/21
Olası Aksamalara Göre Sürecin Yeniden Planlanması	4/11/21	6/4/21
Model Tasarım		
Waterfall Modeline Göre Tasarım Yapılması	4/11/21	4/16/21
UI Tasarımı	4/16/21	4/19/21
Kodlamalar - Kayıt Modülü		
Homepage	4/19/21	4/24/21
Login	4/24/21	4/28/21
Patient	4/28/21	4/30/21
Header	4/30/21	5/8/21
Patient Note	5/8/21	5/10/21
Testing & Bug Hunting		
Performans Testleri ve Doğrulama Çalışmaları	5/10/21	5/20/21
Pazarlama		
Pazar Araştırması	5/20/21	5/21/21
Satış Stratejisi Belirleme	5/21/21	5/23/21
Satış Stratejisi Uygulanması	5/23/21	5/26/21



d) Ekip organizasyon şeması, görev dağılımları



e) Risk Tablosu

ID	Risk	Türü. Grubu	Etkisi	Olasılık
1	Yazılımın geç teslim edilmesi	Proje	Büyük	Düşük
2	Yazılımsal mantık hataları	Yazılım	Büyük	Düşük
3	Kaynakların tüketilmesi/aşılması	Proje	Orta	Düşük
4	Sözleşme ve kanun ile ilgili riskler	Yasal	Büyük	Orta
5	Görev dağılımın yapılamaması	Yönetim	Orta	Düşük

2) İSTEKLERİN MODELLENMESİ

a) Kullanım Senaryoları

Randevu Alma

Kayıt Görevlisi:

1. Hasta tarafından gelen aramayı cevaplar.

2. Sisteme giriş yapar.

3. Kayıt Görevlisi hastanın verdiği bir tarih ve zamanda boş bir doktor var mı diye kontrol eder.

4. Boş bir doktor ile bir randevu kaydı oluşturur.

4.1 Boş bir doktor yok ise uygun bir zaman Kayıt Görevlisine sistem tarafından tavsiye edilir.

4.2 Hasta tarafından kabul edilen bir tarih ve saat çifti bulunana kadar 4.1.1 tekrar eder.

5. Yeni randevu sisteme kayıt edilir.

5.1 Hastanın sistemde bir kaydı yoksa Kayıt Görevlisi tarafından bu kayıt oluşturulur.

5.2 Hastanın randevusu bu yeni hesapla birlikte sisteme kaydedilir.

6. Hasta ile olan arama sonlanır.

Sistem Admin:

1. Eğer Kayıt Görevlisinin sistem üzerinde bir hesabı yoksa bu hesap Sistem Admin tarafından oluşturulur.

Muayene Zamanı

Doktor:

1. Sisteme giriş yapar.

2. Doktor Hastanın TC kimlik numarası ile sistem üzerinde arama yapar.

3. Sistem Hastanın bilgilerini ve geçmiş muayene kayıtlarını Doktora sunar.

4. Doktor sistemde şu anda aktif olan muayene kaydına notlarını ve reçeteyi ekler.

5. Doktor güncellenmiş notları sisteme kaydeder.

6. Hastanın muayenesi sonlanır.

Veznedar:

1. Hastanın TC numarasını alır.

2. TC numarası ile sigortası sorgulanır.

3. Sistem dönen cevaba göre gerekli indirimleri yapar.

Hasta:

1. Hasta nakit olarak son ücreti öder.

2. Hasta kredi kartı ile ödemek isterse kart bilgilerini verir.

3. Sistem son ücretin ödemesini kredi kartından alır.

4. Hasta binadan ayrılır.

Sistem Admin:

1. Eğer sistem üzerinde Doktorun bir hesabı yok ise Sistem Admin tarafından bu hesap oluşturulur.

Doktor Hesabı Oluşturma:

Sistem Admin:

1. Doktorun bilgileri daha önce çalıştığı sistemden alınır.

2. Bu bilgilerle sistemde yeni bir hesap oluşturulur.

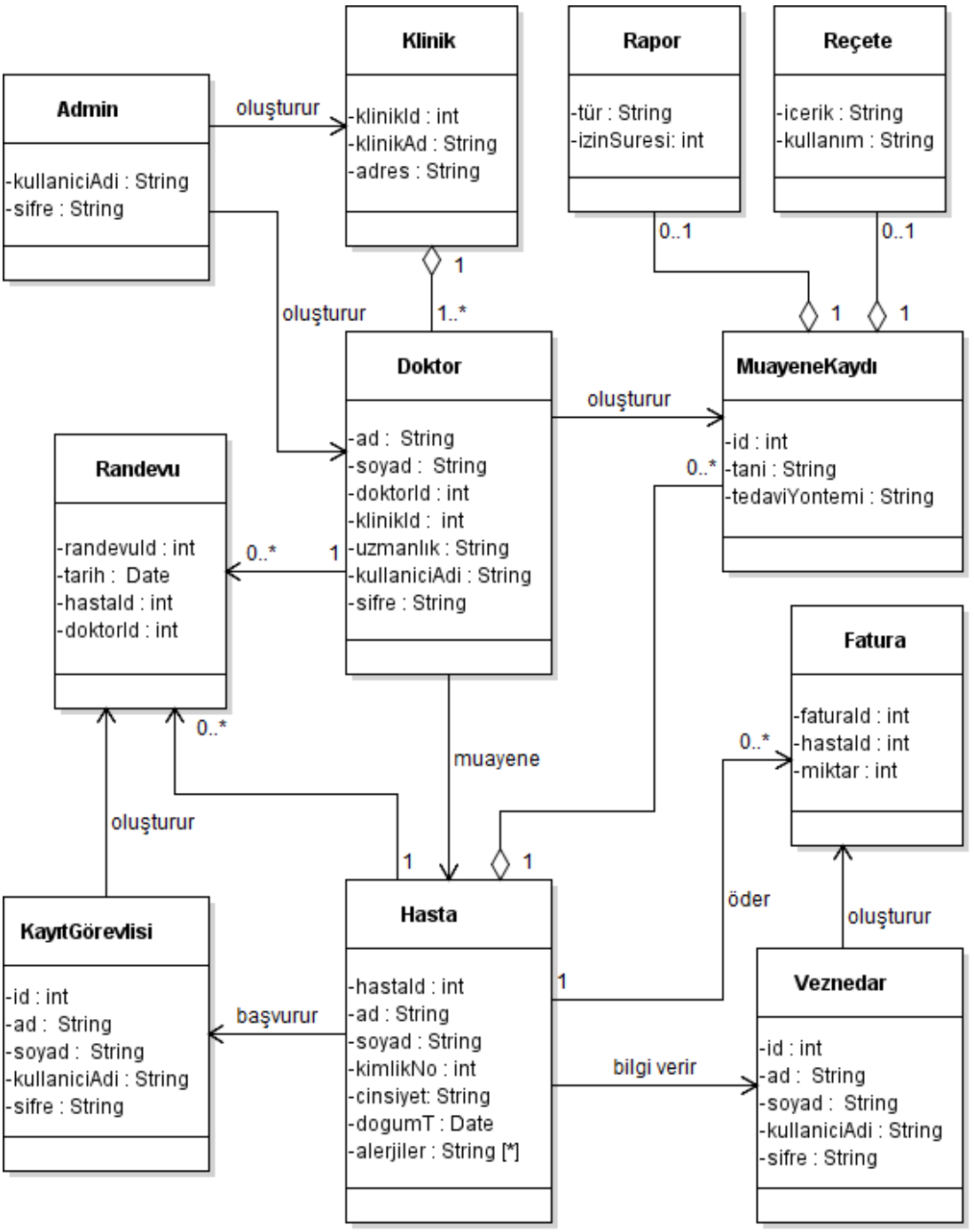
3. Doktor uygun kliniğe eklenir.

3.1 Eğer Doktorun kliniği sistemde yok ise Sistem Admin tarafından oluşturulur.

b) İzlenebilirlik tablosu

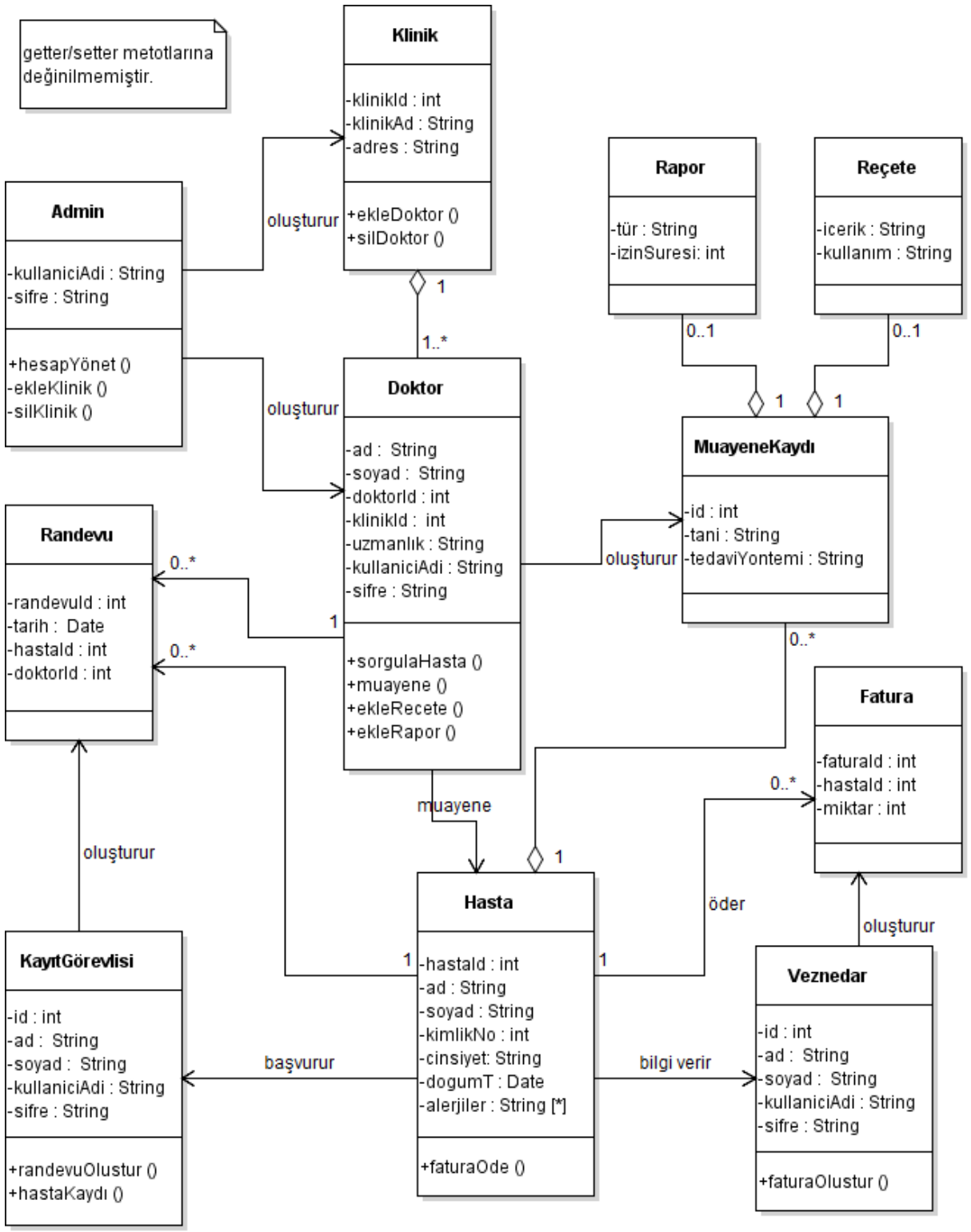
Gereksinim \ Bakış Açıları	Hesap yönetimi	Reçete sorgulama	Randevu sorgulama	Randevu alma	Sigorta sorgulama	Ödeme yapma
Reçete verme		x				
Klinik oluşturma			x			
Hasta kaydı	x			x		
Hasta takibi	x	x	x		x	x
Geçmiş kayıtların takibi		x	x		x	
Kredi kartı ile ödeme	x					x

3) NESNEYE DAYALI MODELLEME

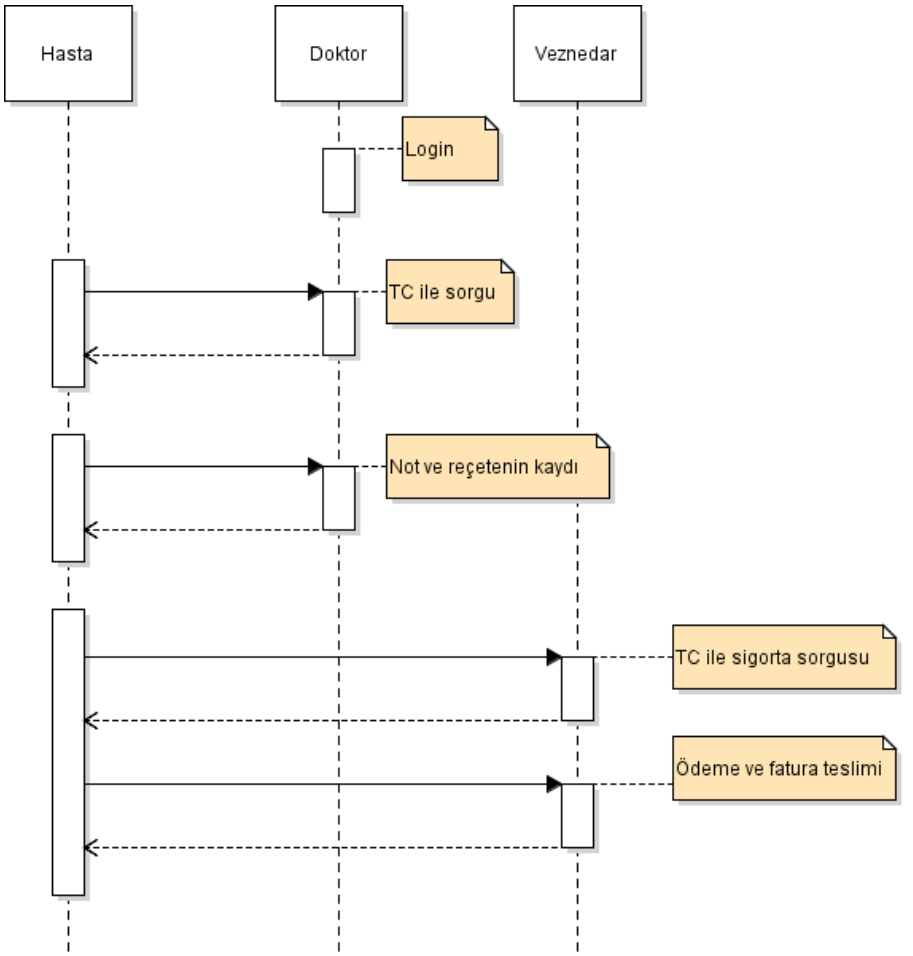


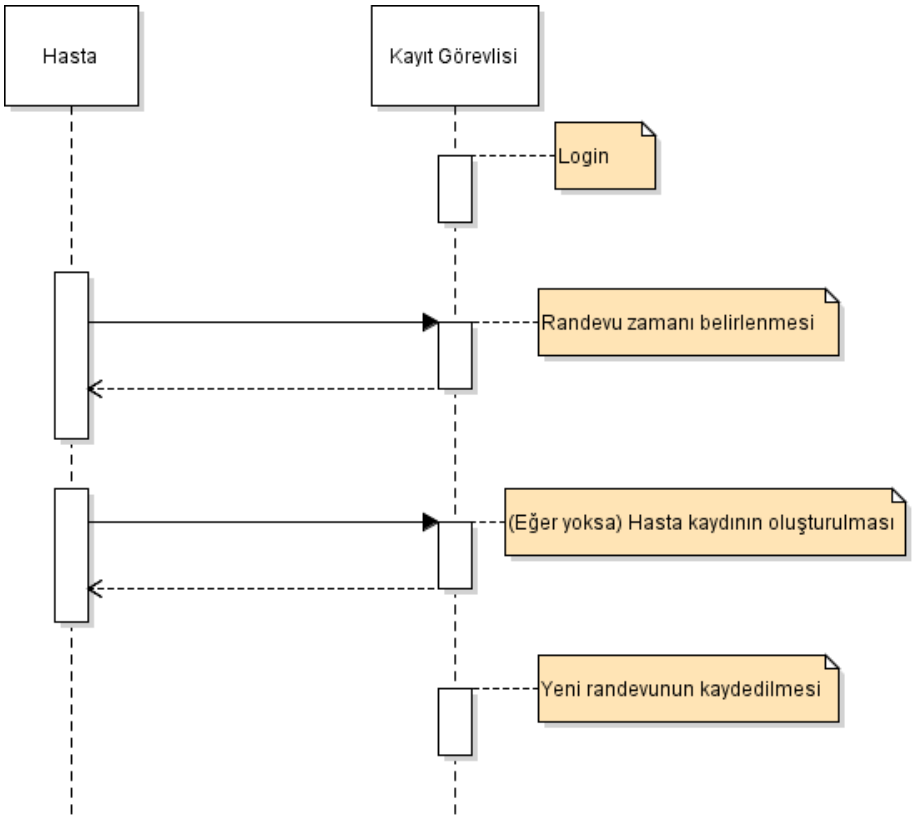
4) NESNEYE DAYALI TASARIM

a) Sınıf Diyagramı (Tasarım Modeli)

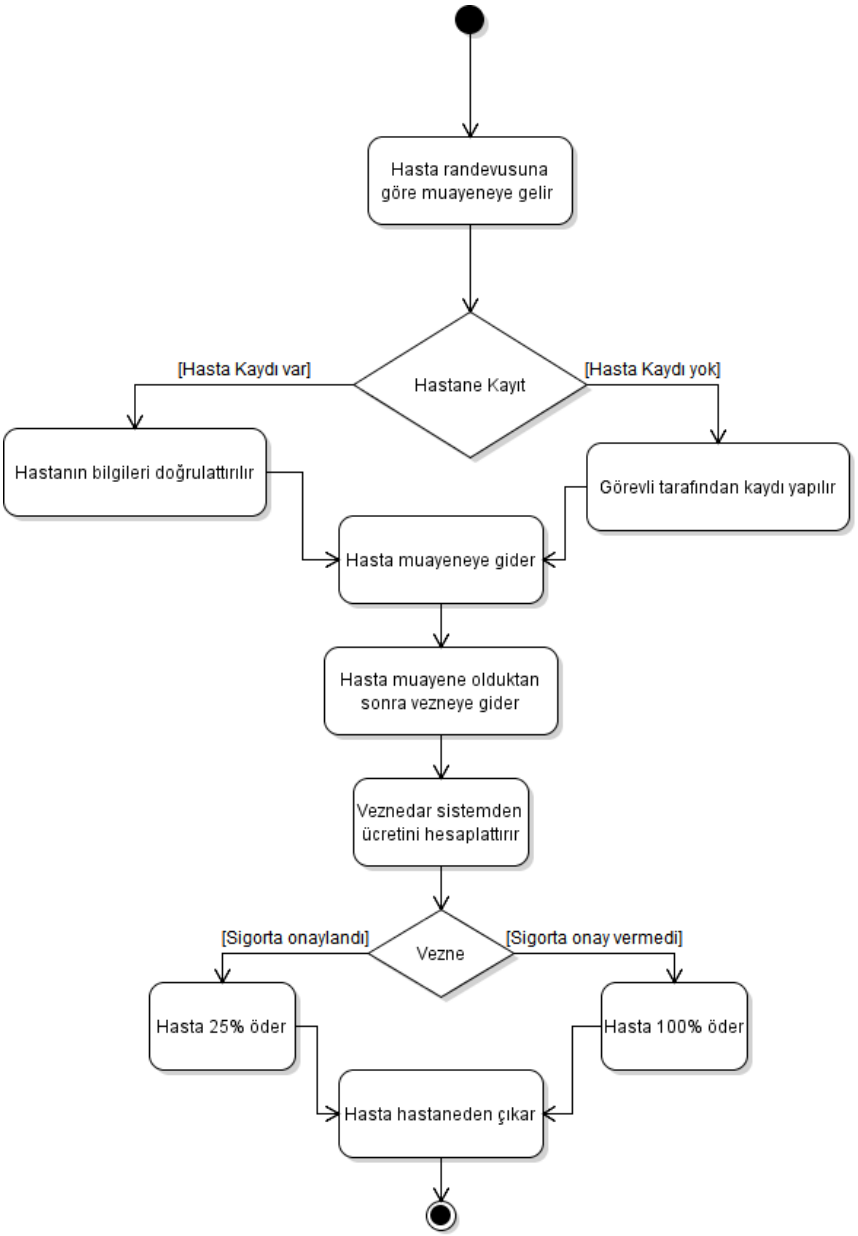


b) Sıralama Diyagramı

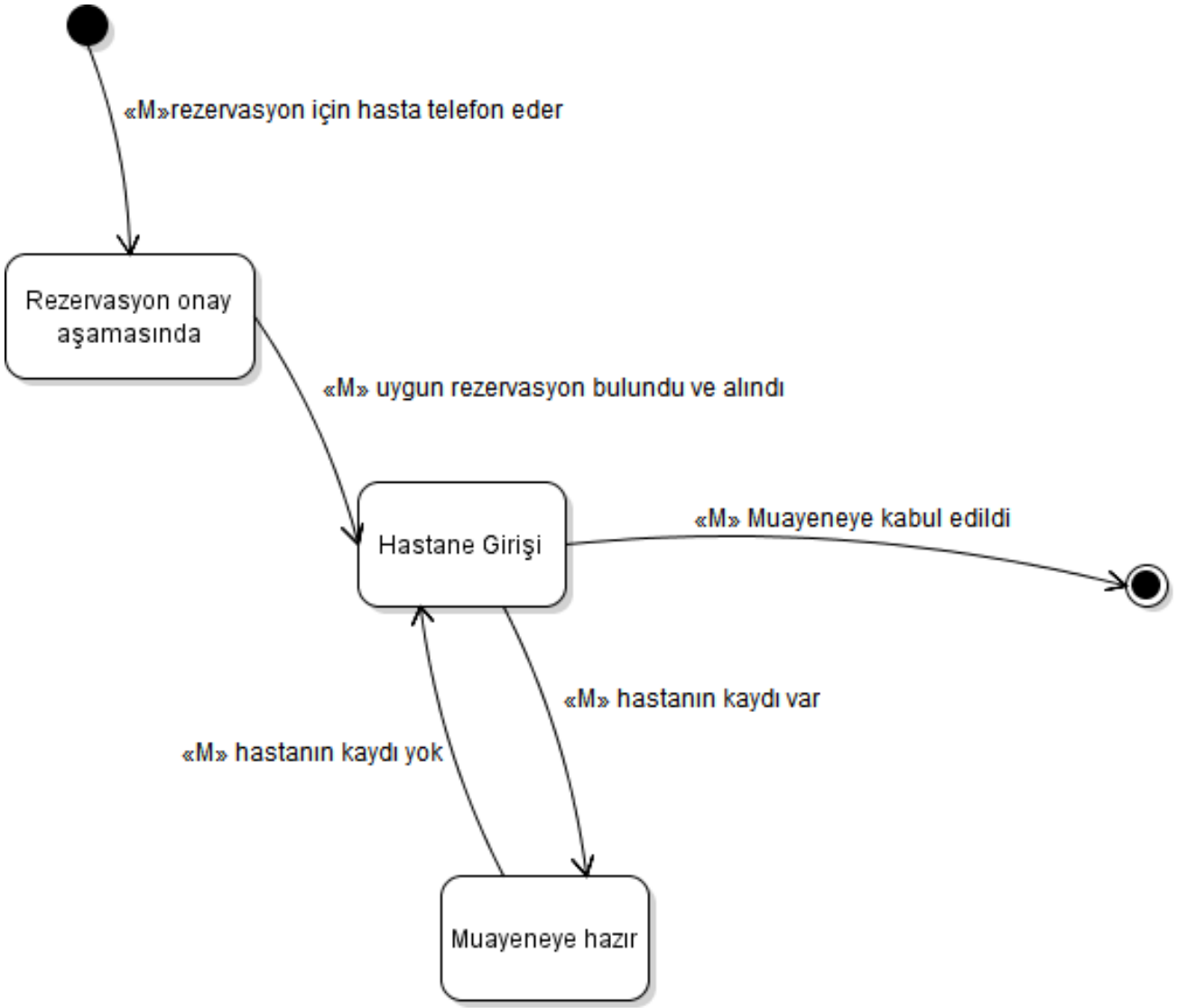




c) Etkinlik Diyagramı



d) Durum Diyagramı



5) BİRİM TEST SINAMALARI

a) Birim Test Kodları

```
from datetime import datetime
import unittest
from controllers import patients
from dotenv import load_dotenv
import os
from models import create_session, Doctor, Date

DATABASE_URL = os.getenv('DATABASE_URL')

class TestGetPatientData(unittest.TestCase):
    def setUp(self):
        self.session = create_session(DATABASE_URL)

    def test_get_patient_data(self):
        patient = patients.get_patient_data(self.session, 123)
        self.assertFalse(patient)

        data = patients.get_patient_data(self.session, 12345678901)
        self.assertEqual(data['patient']['tc'], "12345678901")

    def test_is_patient_exist(self):
        is_exist = patients.is_patient_exist(self.session, 123)
        self.assertFalse(False)

        patient = patients.is_patient_exist(self.session, 12345678901)
        self.assertEqual(patient.tc, "12345678901")

    def test_update_patient_data(self):
        result = patients.update_patient_data(self.session, 123, "Asd")
        self.assertFalse(result)

        result = patients.update_patient_data(self.session, 12345678901,
"Asd")
        self.assertTrue(result)

    def test_create_appintment_data(self):
        doc = self.session.query(Doctor).first()
        start_date = datetime.fromtimestamp(1622388667)
        end_date = datetime.fromtimestamp(1622398967)

        date = Date(start_date=start_date, end_date=end_date)

        appointment = patients.create_appointment_data(self.session,
123, doc, date, "ASD")
        self.assertFalse(appointment)

        appointment = patients.create_appointment_data(self.session,
12345678901, doc, date, "ASD")
        self.assertEqual(appointment.doctor.id, doc.id)
if __name__ == '__main__':
    unittest.main()
```

b) Birim Test Ekran Görüntüsü

```
➜  backend git:(master) ✗ python -m unittest
....
----------------------------------------------------------------------
Ran 4 tests in 0.047s

OK
```