



**İNÖNÜ ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİNDEKİ GELİŞMELER-4 DERSİ**

**PROJE RAPORU**

**PROJE ADI: Dış Hekimliği Randevu Sistemi**

Proje Ekibi Ad Soyad	Okul No	Sınıf/Şube
Hakan Atalay	02185076039	İ.Ö
Ozan Duman	02185076040	İ.Ö
Seyfullah Şalış	02185076051	İ.Ö

**Contents**

1- PROJE TANIMI:.....	1
2- PROJE ARAYÜZÜ: .....	2
2.1- Arayüz Görseli: .....	2
2.2- FXML Olay Fonksiyonları: .....	6
3- SINIF HİYERARŞİSİ ve UML DİYAGRAMLARI:.....	15
3.1- UML Diyagramı:.....	18
3.2- UML Sınıf ve Metotlarının Açıklanması: .....	18

## 1- PROJE TANIMI:

### Proje Konusu

Projemiz, Diş Kliniğinde kullanılması planlanan bu sistem gelen hastalara randevu verebilmek amaçlı oluşturulacak bir sistemdir. Randevu verirken alınacak olan hasta bilgileri ile doktor eşleştirmesi yaparak alınacak randevuda tedavi yöntemi de belirtilebilecektir.

Örneğin:

Hasta Adı-Soyadı : Ahmet Karataş  
Doktor İsmi : Mehmet Şahin  
Tedavi : Kanal Tedavisi  
Randevu Zamanı : 07.04.2020 (13:30)

Yukarıdaki örnek gibi Randevu Listeleme de bu şekilde gösterip Doktor, Tedavi ve Hasta bilgileri diğer sınıflardan çekilerek gösterilecektir.

### Projenin Özellikleri

Projenin sahip olduğu özellikler;

Tedavi Ekleme, Silme, Listeleme, Düzenleme  
Hasta Ekleme, Silme, Listeleme, Düzenleme  
Doktor Ekleme, Silme, Listeleme, Düzenleme  
Randevu Ekleme, Silme, Listeleme, Arama, Düzenleme,

### Sınıf Arayüz Ve Dosya Yapısı

Klasör yapımız, src klasörü altında 4 klasörden oluşmaktadır. Bunlar;

- ➔ assets : Css dosyasını barındırır.
- ➔ assets.image : Kullanılan resimler burada barındırılır.
- ➔ models : Sınıfları barındırır.
- ➔ views : Arayüz dosyaları ve Controller'ları barındırır.

src klasörü dışında ise kütüphane ve dosyalar klasörümüz bulunmaktadır.

- ➔ kutuphane : proje içerisinde kullanılan kütüphaneleri barındırır.
- ➔ dosyalar: proje içerisinde oluşturulan dosyalar burada barındırılır.

## 2- PROJE ARAYÜZÜ:

### Çalışma Mantığı

İlk başta proje Giriş Yap sayfasıyla açılır giriş işlemi doğru bir şekilde yapıldıktan sonra karşımıza Randevu ekranı açılır.

Sol tarafta bulunan Tedavi,Hasta,Doktor (Butonlarıyla) navigasyon sağlanır.

Randevuda olduğu gibi Ekleme,Düzenleme ve Silme Butonları Hasta, Doktor, Tedavi’de de mevcuttur ve bu butonlar ekleme düzenleme ve silme aksiyonlarını gerçekleştirir. Bunlara ek olarak;

Tedavide, Tedavi Görüntüle butonu ve Randevuda ise Arama için kullandığımız Textbox alanı da mevcuttur.

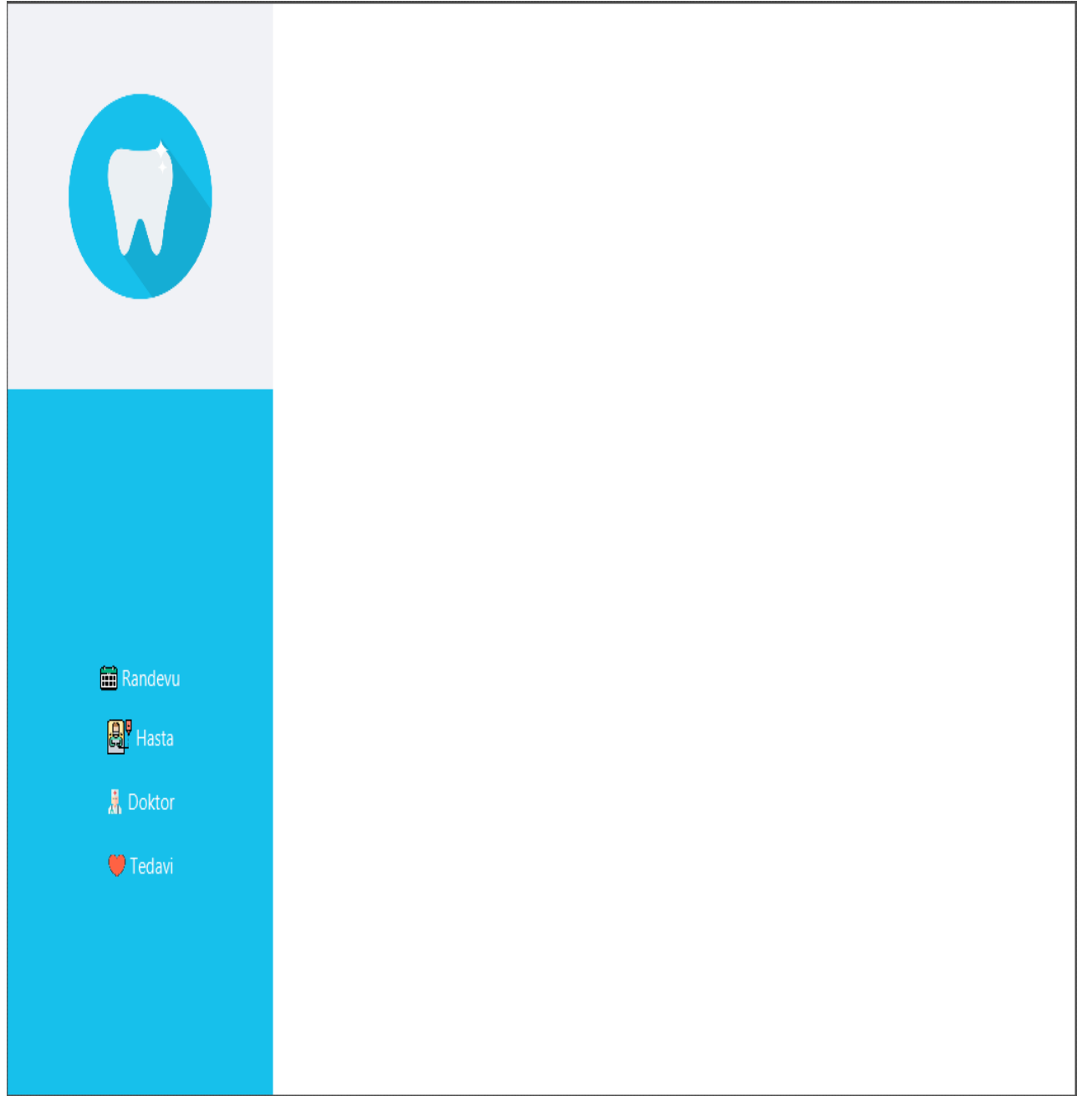
Kullandığımız Tableviewler ise bütün Sınıflar için mevcuttur ve Listeleme işlemini Tableviewlerle gerçekleştiriyoruz.

### 2.1- Arayüz Görseli:

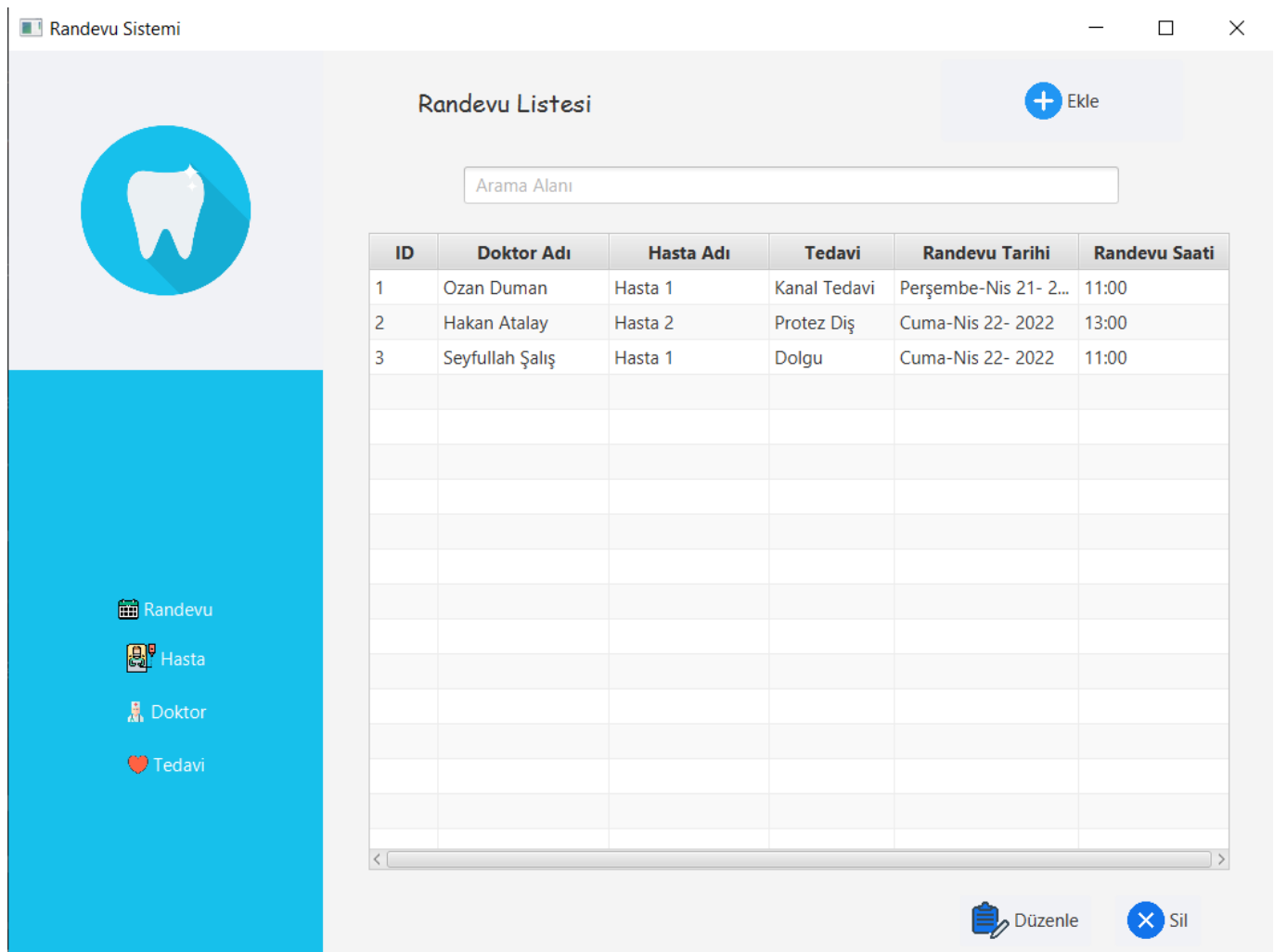
#### Modül 1 – Giriş Yap Ekranı

Bu ekranda kullanıcı giriş yaparak Ana Sayfaya yönlendirilir. Eğer yanlış işlem yapılırsa hata verir.

## Modul 1.2 – Menü (Navigasyon) Modülü



## Modul 2 – Anasayfa (Randevu Listesi)



Bu ekranda Randevu Listelenir.

Randevu Ekleme, Düzenleme ve Silme işlemleri için gerekli butonlar bulunur.

## Modul 2.1 – Randevu Ekleme

Randevu Ekle

ID

Doktor Adı

Lütfen Doktor Seçiniz.

Hasta Adı

Lütfen Eklediğiniz Hastayı Giriniz.

Tedaviler

Lütfen Tedavi Seçiniz.

Tarih

Saat

Lütfen Saat Seçiniz

İptal Kaydet

Bu ekranda gereken alanlar doldurularak Randevu ekleme işlemi gerçekleştirilir.

## Modul 2.2 – Randevu Düzenleme

Randevu Düzenle

Randevu Ekle

ID

1

Doktor Adı

Ozan Duman

Hasta Adı

Hasta 1

Tedaviler

Kanal Tedavi

Tarih

21.04.2022

Saat

11:00

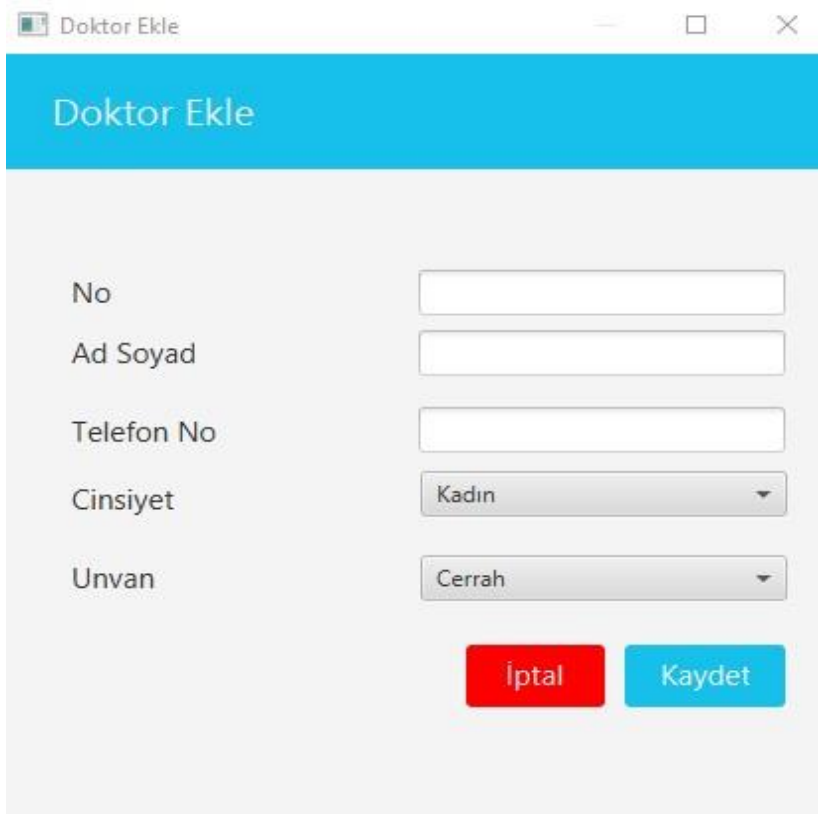
İptal Kaydet

Bu ekranda listeden seçilen randevu için düzenleme işlemi yapılır.

[illegible]

Doktor Ekleme, Düzenleme ve Silme işlemleri için gerekli butonlar bulunur.

### Modul 3.1 – Doktor Ekleme



Doktor Ekle

No

Ad Soyad

Telefon No

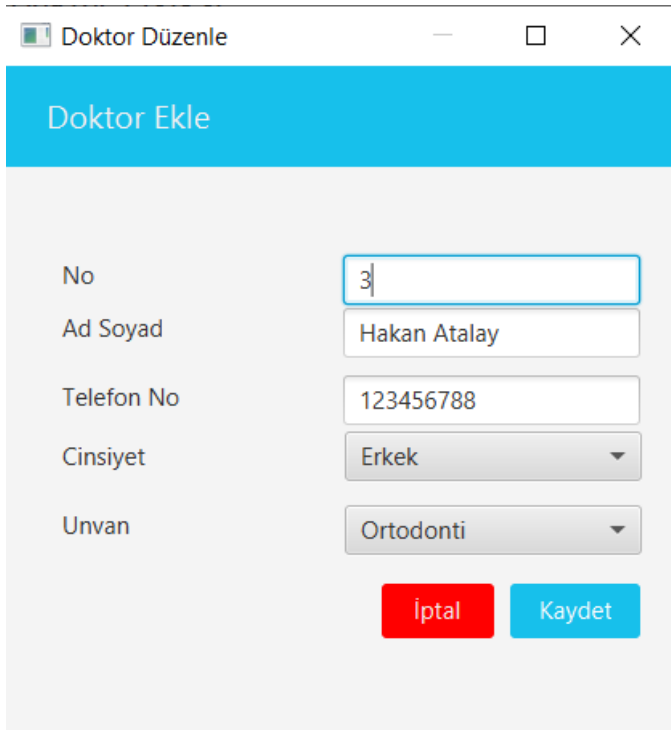
Cinsiyet Kadın

Unvan Cerrah

İptal Kaydet

Bu ekranda gereken alanlar doldurularak Doktor ekleme işlemi gerçekleştirilir.

### Modul 3.2 – Doktor Düzenleme



Doktor Ekle

No 3

Ad Soyad Hakan Atalay

Telefon No 123456788

Cinsiyet Erkek

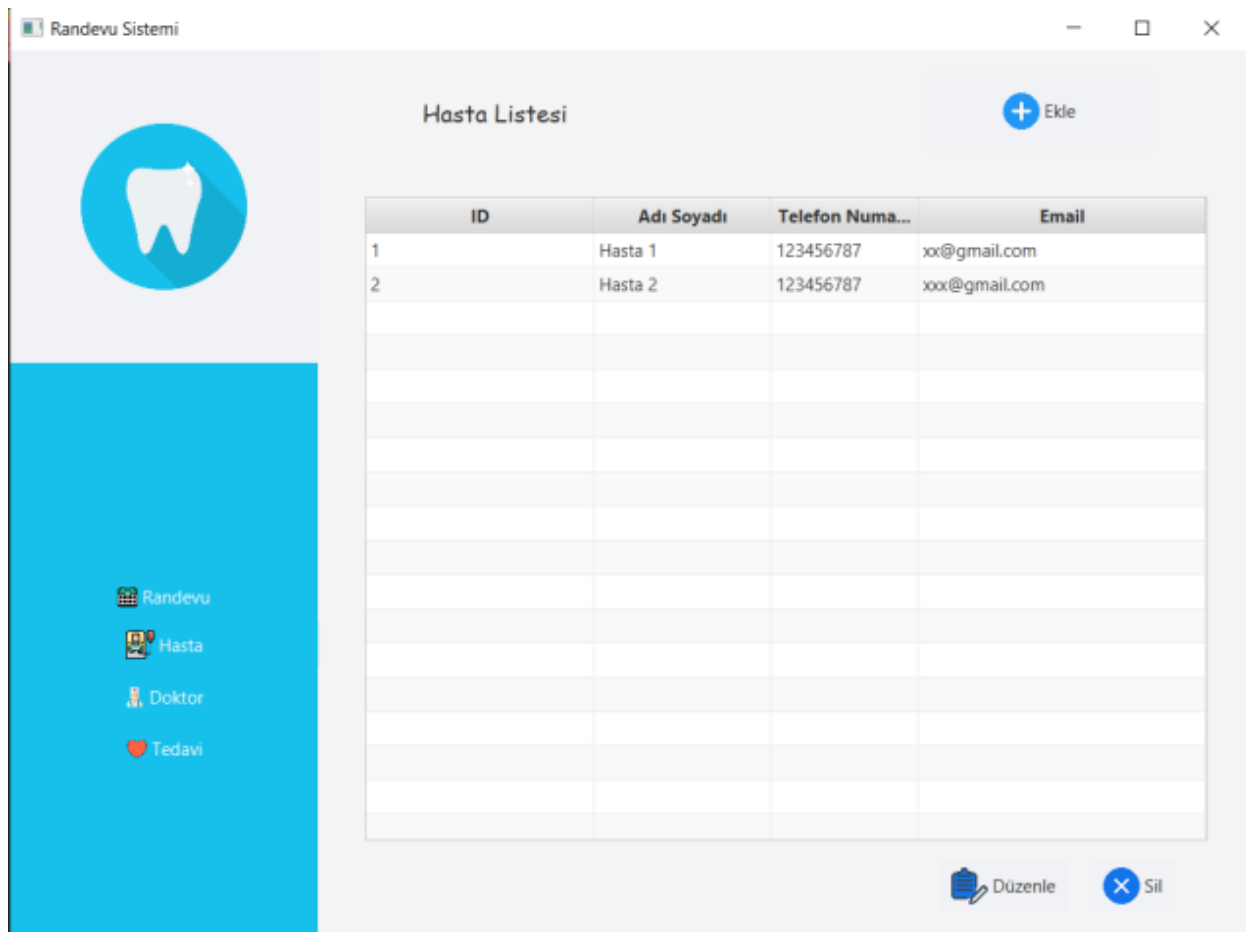
Unvan Ortodonti

İptal Kaydet

Bu ekranda listeden seçilen doktor için düzenleme işlemi yapılır.



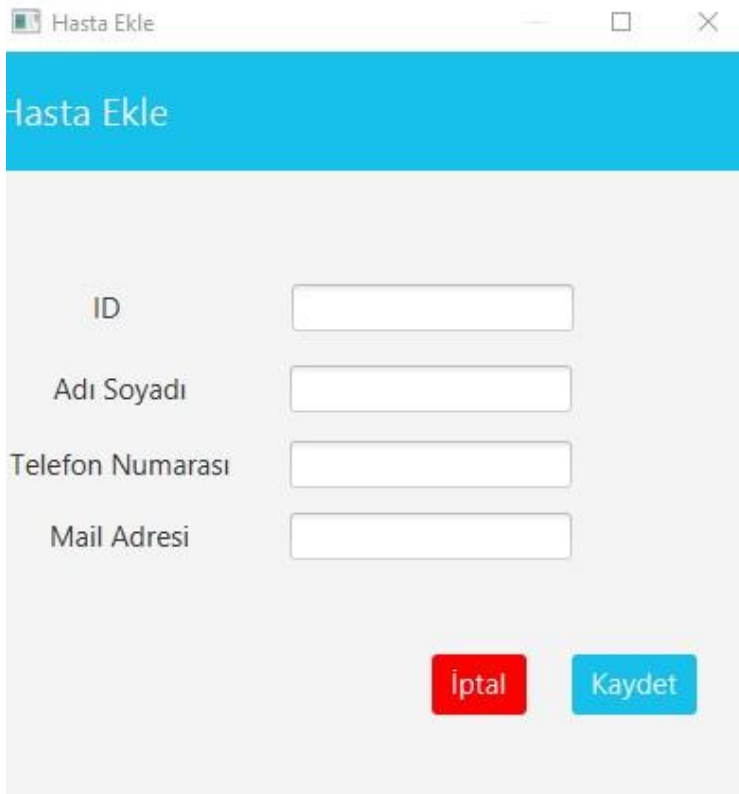
## Modul 4 – Hasta Listesi



Bu ekranda Hastalar Listelenir.

Hasta Ekleme, Düzenleme ve Silme işlemleri için gerekli butonlar bulunur.

## Modul 4.1 – Hasta Ekle



Hasta Ekle

ID

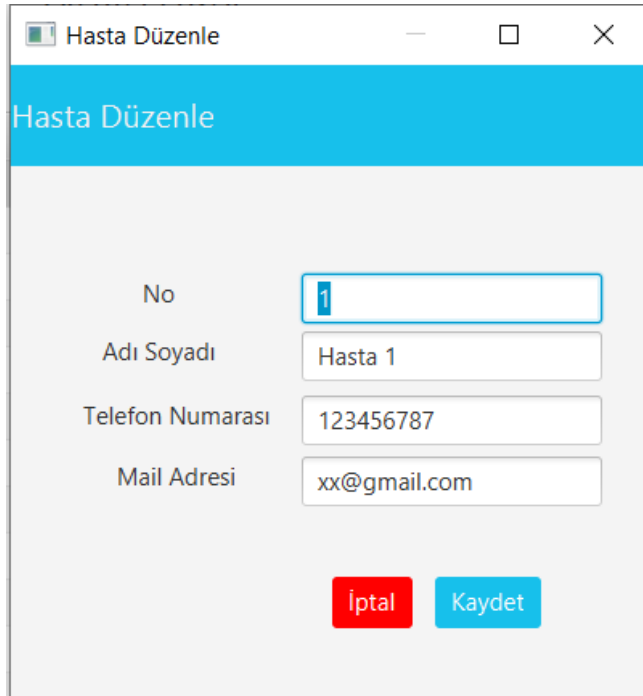
Adı Soyadı

Telefon Numarası

Mail Adresi

Bu ekranda gereken alanlar doldurularak Hasta ekleme işlemi gerçekleştirilir.

## Modul 4.2 – Hasta Düzenle



Hasta Düzenle

No

Adı Soyadı

Telefon Numarası

Mail Adresi

Bu ekranda listeden seçilen hasta için düzenleme işlemi yapılır.

## Modul 5 – Tedavi Listele

Bu ekranda Tedaviler Listelenir.

Tedavi Ekleme, Düzenleme, Silme ve Görüntüleme işlemleri için gerekli butonlar bulunur.

### Modul 5.1 – Tedavi Ekle

Bu ekranda gereken alanlar doldurularak Tedavi ekleme işlemi gerçekleştirilir.

## Modul 5.2 – Tedavi Düzenle



Tedavi Düzenle

ID: 1

Adı: Dolgu

Açıklama: dfgfgdfg

İptal Kaydet

Bu ekranda listeden seçilen tedavi için düzenleme işlemi yapılır.

## Modul 5.3 – Tedavi Görüntüle



Tedavi Görüntüle

Tedavi ID: 1

Adı : Dolgu

Açıklaması: dfgfgdfg

Bu ekranda tedavinin detayları gösterilmektedir.

## **2.2- FXML Olay Fonksiyonları:**

### **Modul 1 – Giriş Yap Olay Aksiyonu**

Giriş Yap Aksiyonu (Butonu) : Kullanıcı adı ve Şifreye bakarak bir kontrol sağlar eğer doğru ise Anasayfa'ya yönlendirir yanlış ise Hata mesajı ekrana verir.

### **Modul 1.2 – Navigasyon Olay Aksiyonları**

Solda bulunan Randevu, Doktor, Hasta, Tedavi Butonları navigasyon için kullanılır o butonlara tıklayarak menüler arasında gezinti sağlanır.

### **Modul 2 – Randevu Listele Olay Aksiyonları**

Ekle Aksiyonu (Butonu) : Randevu ekleme penceresini açar.

Düzenle Aksiyonu (Butonu) : Listedden seçilen randevunun düzenlenmesi için gereken pencereyi açar. Eğer listeden randevu seçilmez ise hata verir.

Sil Aksiyonu (Butonu) : Listedden seçilen randevuyu siler.

### **Modul 2.1 – Randevu Ekleme Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Textbox ,ComboBox ve DatePicker'daki gelen verileri alıp yeni bir Randevu nesnesi oluşturup onu dosyaya yazar.

### **Modul 2.2 – Randevu Düzenleme Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Listedden aldığımız randevuyu düzenleyip tekrardan yeni bir nesneymiş gibi dosyaya yazdırma işlemini yapar.

### **Modul 3 – Doktor Listele Olay Aksiyonları**

Ekle Aksiyonu (Butonu) : Doktor ekleme penceresini açar.

Düzenle Aksiyonu (Butonu) : Listedden seçilen doktorun düzenlenmesi için gereken pencereyi açar. Eğer listeden doktor seçilmez ise hata verir.

Sil Aksiyonu (Butonu) : Listedden seçilen doktoru siler.

### **Modul 3.1 – Doktor Ekleme Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Textbox , Comboboxtaki verileri alıp Unvan'ına göre yeni bir Doktor nesnesi oluşturup dosyaya yazar.

### **Modul 3.2 – Doktor Düzenleme Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Listedden aldığımız doktoru düzenleyip tekrardan yeni bir nesneymiş gibi dosyaya yazdırma işlemini yapar.

### **Modul 4 – Hasta Listele Olay Aksiyonları**

Ekle Aksiyonu (Butonu) : Hasta ekleme penceresini açar.

Düzenle Aksiyonu (Butonu) : Listedden seçilen hastanın düzenlenmesi için gereken pencereyi açar. Eğer listeden hasta seçilmez ise hata verir.

Sil Aksiyonu (Butonu) : Listedden seçilen hastayı siler.

### **Modul 4.1 – Hasta Ekleme Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Textboxdaki verileri alıp yeni bir hasta nesnesi oluşturup dosyaya yazar.

### **Modul 4.2 – Hasta Düzenleme Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Listedden aldığımız hastayı düzenleyip tekrardan yeni bir nesneymiş gibi dosyaya yazdırma işlemini yapar.

### **Modul 5 – Tedavi Listele Olay Aksiyonları**

Ekle Aksiyonu (Butonu) : Tedavi ekleme penceresini açar.

Düzenle Aksiyonu (Butonu) : Listedden seçilen tedaviyi düzenlenmesi için gereken pencereyi açar. Eğer listeden doktor seçilmez ise hata verir.

Sil Aksiyonu (Butonu) : Listedden seçilen tedaviyi siler.

Görüntüle Aksiyonu (Butonu) : Listedden seçilen tedavinin detayını yeni bir pencerede gösterir.

### **Modul 5.1 – Tedavi Ekle Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Textboxdaki verileri alıp yeni bir tedavi nesnesi oluşturup dosyaya yazar.

### **Modul 5.2 – Tedavi Düzenle Olay Aksiyonları**

İptal Aksiyonu (Butonu) : Pencereyi kapatır.

Kaydet Aksiyonu (Butonu) : Listedden aldığımız tedaviyi düzenleyip tekrardan yeni bir nesneymiş gibi dosyaya yazdırma işlemini yapar.

### **Modul 5.3 – Tedavi Görüntüle Olay Aksiyonları**

Burada herhangi bir olay aksiyonu bulunmamaktadır sadece tedavinin detay görüntülemesi işlemi sağlanır.

### 3- SINIF HİYERARŞİSİ ve UML DİYAGRAMLARI:

**Toplamda 29 adet sınıf tasarlanmıştır. Bu Sınıflar;**

**models klasörü içindeki sınıflar;**

Bu sınıfları veri yazmada bir örnek oluşturması yani nesne oluşturması amacıyla tasarlanmıştır.

Bu sınıfların çoğunda metotlar bulunmamaktadır sadece getter/setterlar ve özelliklerini barındırırlar.

**Dosyaİslemleri Sınıfı:** Bu sınıfta dosya işlemleri gönderilen dosya ismine bağlı olarak çalışır. Bir metot bütün sınıflar için çalışır. Dosyalar için ana dizin oluşturulup sistem yolu belirtilir.

**Hasta Sınıfı:** Hastaların bir modelini oluşturmak için kullanılır. İçerisinde getter/setter ve nesnelerin dosyaya anlamlı bir şekilde yazılabilmesi için toString metodu override edilmiştir ekstradan bu sınıf içerisinde dosyadanHastaAdıGetir() adında bir metot bulunur bu metot ise dosyadaki hasta isimlerini alıp bir Observable listeye atar ve bu listeyi Randevu verirken kullanırız.

**Doktor Sınıfı:** Abstract sınıfımızdır. Soyutlama için kullanırız.

Bu sınıftan nesne türetilmez. Sadece Doktorun bir modelini oluşturmak için kullanırız Hasta'da olduğu bundada toString ve randevu verirken kullanmak amaçlı dosyadanDoktorAdıGetir() adlı metot bulunur ekstradan bu sınıfta getUnvan adında bir metot bulunur alt sınıftaki Unvan Özelliğine erişebilmek için bu metot kullanılmıştır. Alt sınıfları şunlardır;

Ortodonti,Endodonti,Pedodonti,Cerrah'tır ve bu sınıflarda toString ve getter/setter harici metot bulunmaz Unvan'a göre ayırma yaptığımız sınıflardır.

**Tedavi Sınıfı:** Tedavilerin bir modelini oluşturmak için kullanılır. İçerisinde getter/setter ve nesnelerin dosyaya anlamlı bir şekilde yazılabilmesi için toString metodu override edilmiştir. Ekstradan bu sınıf içerisinde randevuda kullanılmak için dosyadanTedaviAdıGetir() adlı metot ve alt sınıflardaki Tedavi Açıklama Özelliğine erişmek için getTedaviAçıklama() adında bir metot bulunur. Alt sınıfları şunlardır;

Dolgu,DisCekimi,KanalTedavi,ProtezDis'tir ve bu sınıflarda toString ve getter/setter harici metot bulunmaz İsme göre ayırma yaptığımız sınıflardır.



**Randevu Sınıfı:** Randevunun bir modelini oluşturmak için kullanılır. İçerisinde sınıfın özellikleri, getter/setter ve nesnelerin dosyaya anlamlı bir şekilde yazılabilmesi için toString() metodu barındırır.

### **Views klasörü içindeki sınıflar;**

Bu sınıfları fxml ile bağlantı kurmak için tasarladık. Fxml'de yapılacak işlemlerin metotları(Ekleme,Silme,Düzenleme) bu sınıflarda bulunur.

**LoginController Sınıfı:** login.fxml'in Controller'idir. Fxml'de yapılacak olan login işleminin kontrolü bu sınıfta yazılmıştır.

**HomeController Sınıfı:** Bu sınıf home.fxml'in controlleridir.home.fxml içerisinde bulunan doktor, hasta ,randevu ve tedavi butonlarının gerekli navigasyon(menü) işlemini yapmasını sağlar.

**DoktorController Sınıfı:** doktor.fxml'in Controller'idir. Listeleme, ekleme, düzenleme ve silme metotlarının olduğu sınıftır.

**DoktorEklemeController Sınıfı :** doktorEkle.fxml'in Controller'idir. Doktordan bir nesne oluşturup bunu bir observable list'e ekleyip o listeyi dosyaya yazmayı sağlar.

**DoktorDuzenleController Sınıfı:** doktorDuzenle.fxml'in Controller'idir.

Listeden seçilen Doktoru düzenleyip seçilen doktoru listeden silip düzenlenen doktor için yeni bir nesne oluşturulup o nesneyi observable listeye yazıp sonra o listeyi dosyaya yazdırırız.

**Main Sınıfı:** Uygulamanın çalışması için gereken sınıftır. Uygulamanın login.fxml'den çalışmasını sağlar.

**HastaController Sınıfı:** hasta.fxm'in controllerıdır. Listeleme, ekleme, düzenleme ve silme metotlarının olduğu sınıftır.

**HastaEkleController:** hastaEkle.fxml'in controlleridir. Bir hastadan bir nesne oluşturup bunu observableListe atıp bunu dosyaya yazmayı sağlar.

**HastaDuzenleController:** hastaduzenle.fxml'in controlleridir. Listeden seçilen hastayı düzenleyip seçilen hastayı listeden silip düzenlenen hastayı yeni nesne olarak oluşturarak observableList e ekler ve bu listeyi tekrar dosyaya yazar.

**TedaviController:** tedavi.fxml'in controller'idir. Listeleme, ekleme, düzenleme ve silme metotlarının olduğu sınıftır.

**TedaviEkleController:** tedaviEkle.fxml'in controlleri'dir. Tedaviden bir nesne oluşturup bunu bir observable list'e ekleyip o listeyi dosyaya yazmayı sağlar.

**TedaviDüzenleController:** tedaviDuzenle.fxml'in Controller'idir. Listedden seçilen tedaviyi düzenleyip seçilen tedaviyi listeden silip düzenlenen tedaviyi yeni nesne olarak oluşturarak observableList e ekler ve bu listeyi tekrar dosyaya yazar.

**TedaviGoruntuleController:** tedaviGoruntule.fxml'in controller'idir. Listedden seçilen tedavinin açıklamasını göstermesi için tasarlanmıştır.

**RandevuControler:** randevu.fxml'in controlleridir. Listeleme, ekleme, düzenleme ve silme metotlarının olduğu sınıftır.

**RandevuEkleController:** randevuEkle.fxml'in controlleridir. Bir randevu nesnesi oluşturur ve observableList'e ekler ardından bu observableList'i dosyaya yazar.

**RandevuDüzenleController:** randevuDüzenle.fxml'in controlleridir. Listedden seçilen randevuyu düzenleyip seçilen randevuyu listeden silip yeni nesne oluşturarak observableList'e ekler ve bu listeyi dosyaya yazar.

### 3.1- UML Diyagramı:



### 3.2- UML Sınıf ve Metotlarının Açıklanması:

#### Models klasörü içindeki Sınıflar:

##### Doktor Sınıfı:

- ➔ **getDoktorAdiFromFile()** : Dosyadan Doktor adlarını alıp bir observableList'e ekler ve o listeyi geri dönderir. Randevu Ekle'de kullanılmak için yazılmıştır.
- ➔ getter/setter metotları
- ➔ **toString()**: Nesneyi dosyaya anlamlı bir şekilde yazması için kullanıldı.
- ➔ **getUnvan()**: Altsınıftaki getUnvan metodunu çağırır hangi sınıftan çağırılmışsa onun Unvanını getirir.

##### Cerrah Sınıfı:

- ➔ getter/setter metotları
- ➔ **toString()**: Nesneyi dosyaya anlamlı bir şekilde yazması için kullanıldı.

##### Endodonti Sınıfı:

- ➔ getter/setter metotları
- ➔ **toString()**: Nesneyi dosyaya anlamlı bir şekilde yazması için kullanıldı.

##### Ortodonti Sınıfı:

- ➔ getter/setter metotları
- ➔ **toString()**: Nesneyi dosyaya anlamlı bir şekilde yazması için kullanıldı.

##### Pedodonti Sınıfı:

- ➔ getter/setter metotları
- ➔ **toString()**: Nesneyi dosyaya anlamlı bir şekilde yazması için kullanıldı.

### **DosyaIslemleri Sınıfı:**

- ➔ dosyayaYaz(ObservableList list, String dosya): Gelen listeyi alır belirtilen dosyaya yazma işlemini gerçekleştirir.
- ➔ dosyayiCagir(String dosya): Gelen dosya adına göre istenilen dosyayı BufferedReader ile çağırır.
- ➔ dosyadanOku(String path): Gelen path'e göre dosyadaki verileri okuyup observableListesine ekler. Genel bir okumadır. Sınıf Ayrımı yapmaz.
- ➔ setDosyalarIcinAnaDizin(String dosyaAdi): Dosya adına göre dizin oluşturup o dosyayı oraya oluşturur.. Eğer dosya varsa oluşturma yapmaz ama yoksa dosyayı o yola oluşturur.

### **Randevu Sınıfı:**

- ➔ getter/setter metotları
- ➔ toString(): Nesneyi dosyaya anlamlı bir şekilde yazması için kullanıldı.

### **Tedavi:**

- ➔ getter/setter metotları
- ➔ toString: Bu sınıftan oluşturulan nesnelerin dosyaya okunabilir biçimde yazılması için.
- ➔ getTedaviAciklama: Alt sınıftaki tedavi\_aciklama özelliğini geri döndürür.
- ➔ getTedaviAdiFromFile(): Randevu sınıfında kullanılmak üzere dosyadaki tedavi isimlerini geri döndürür.

### **DisCekimi:**

- ➔ getter/setter metotları
- ➔ toString: Nesnelerin anlamlı bir biçimde yazılması için kullanıldı.

### **Dolgu:**

- ➔ getter/setter metotları
- ➔ toString: Nesnelerin anlamlı bir biçimde yazılması için kullanıldı.

### **KanalTedavi:**

- ➔ getter/setter metotları
- ➔ toString: Nesnelerin anlamlı bir biçimde yazılması için kullanıldı.

### **ProtezDis:**

- ➔ getter/setter metotları
- ➔ toString: Nesnelerin anlamlı bir biçimde yazılması için kullanıldı.

## **Hasta:**

- ➔ getter/setter metotları
- ➔ toString: Nesnelerin anlamlı bir biçimde yazılması için kullanıldı.
- ➔ getHastaAdiFromFile():Randevu sınıfında kullanılmak üzere dosyadaki hasta isimlerini geri döndürür.

## **Views içerisindeki Sınıflar**

### **DoktorController:**

- ➔ getDoktorFromFile(): Doktor dosyasına gidip içindeki verileri bir nesne oluşturup listeye atıp o listeyi geri döndürür.
- ➔ doktorEkleDialog(): doktorEkle.fxml'i yeni bir pencerede çağırır.
- ➔ doktorDuzenleDialog(): Listedden seçilen Doktor'u yeni pencerede açar ve doktorDuzenle.fxml'e set eder. Eğer listeden seçili doktor yoksa hata verir.
- ➔ initialize(): TableView'in doldurulması bu metotta gerçekleştirilir. Silme metoduda burada tetiklenir.

### **DoktorEkleController:**

- ➔ initialize(): Unvan ve Cinsiyet Comboboxlarına ilk ataması yapılır.
- ➔ doktorEkle(): Textboxlara yazılan verileri alıp yeni bir nesne oluşturup bu nesneyi observableList'e yazıp bu listeyide dosyaya yazar işlem bitince pencereyi kapatır.
- ➔ setDoktorList(): DoktorController'daki listeyi burdaki listeye eşitlemek için kullanılır.
- ➔ doktorEkleClose():İptal butonuna tıklanınca pencereyi kapatır.
- ➔ closeStage(): Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.

### **DoktorDuzenleController:**

- ➔ setDoktorList(): DoktorController'daki listeyi burdaki listeye eşitlemek için kullanılır.
- ➔ initData(Doktor d): DoktorController'daki listeden seçilen doktoru DuzenleController'da verileri doldurulmuş bir şekilde getirmeyi sağlar.
- ➔ doktorDuzenle(): Önce seçilmiş olan Doktor'u listeden siler daha sonra yeni bir doktor oluşturup, oluşturulan bu Doktor'u observableList'e ekledikten sonra Dosyaya listeyi yazar.
- ➔ doktorDuzenleClose():İptal butonuna tıklanınca pencereyi kapatır.
- ➔ closeStage():Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.

- ➔ initialize(): Unvan ve Cinsiyet Comboboxlarına ilk ataması yapılır.

### **HomeController:**

- ➔ initialize(): Açılışta randevu.fxml'i ekranda göstermeyi sağlar.
- ➔ randevu(): Tıklanınca randevu.fxml'i ekranda gösterir.
- ➔ hasta(): Tıklanınca hasta.fxml'i ekranda gösterir.
- ➔ doktor(): Tıklanınca doktor.fxml'i ekranda gösterir.
- ➔ tedavi(): Tıklanınca tedavi.fxml'i ekranda gösterir.
- ➔ loadPage(): fxml'in ekranın hangi kısmına ekleneceğini gösterir gelen fxml'i ekrandaki ilgili yere set eder.

### **LoginController:**

- ➔ girisYap(): Kullanıcı adı ve şifreye göre kontrole sağlar eğer girilen bilgiler doğruysa Anasayfaya yönlendirir. Eğer girilen bilgiler yanlış ise hata mesajı ekrana verir.
- ➔ loadWindow(): Eğer girilen bilgiler doğru ise programın açılmasını ve anasayfaya yönlendirme yapar.

### **Main:**

- ➔ start(): Programın başladığı kısımdır login.fxml'i çağırır.

### **TedaviController:**

- ➔ getTedaviFromFile(): Tedavi dosyasına gidip içindeki verileri bir nesne oluşturup o nesneyi listeye atıp o listeyi geri döndürür.
- ➔ initialize(): ObservableList'in içi boşaltılır. Daha sonra TableView'e veriler çekilir. Listedden seçilen tedavi silme metodu buradadır.
- ➔ tedaviEkleDialog(): tedaviEkle.fxml'i yeni bir pencerede çağırır.
- ➔ tedaviDuzenleDialog(): Listedden seçilen Tedaviyi yeni pencerede açar ve tedaviDuzenle.fxml'e set eder. Eğer listeden seçili tedavi yoksa hata verir.
- ➔ tedaviGoruntuleDialog(): Listedden seçilen Tedaviyi yeni pencerede açar ve tedaviGoruntule.fxml'e set eder. Sadece tedavi detayı gösterilmesi için yazılan metottur. Eğer listeden seçili tedavi yoksa hata verir.

### **TedaviEkleController:**

- ➔ initialize(): Otomatik Doldurma işlemi için ControlsFx Kütüphanesi kullanılmıştır. Onun metodu set edilmiştir.
- ➔ tedaviEkle(): Textboxlara yazılan verileri alıp yeni bir nesne oluşturup bu nesneyi observableList'e yazıp bu listeyide dosyaya yazar işlem bitince pencereyi kapatır.

- ➔ tedaviEkleClose():İptal butonuna tıklanınca pencereyi kapatır.
- ➔ closeStage():Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.
- ➔ setTedaviList():TedaviController'daki observableList'eyi burada kullanılan listeye set etmek için kullanılır.

### **TedaviDuzenleController**

- ➔ initData():TedaviController'daki listeden seçilen tedaviyi DuzenleController'da verileri doldurulmuş bir şekilde getirmeyi sağlar. Otomatik Doldurma yapması için ControlsFx Kutuphanesi metodu kullanılmıştır.
- ➔ setTedaviList():TedaviController'daki observableList'i burada kullanılan listeye set etmek için kullanılır.
- ➔ tedaviDuzenle():Önce seçilmiş olan Tedaviyi listeden siler daha sonra yeni bir Tedavi oluşturup, oluşturulan bu Tedavi'i observableList'e ekledikten sonra listeyi Dosyaya yazar.
- ➔ tedaviDuzenleClose():İptal butonuna tıklanınca pencereyi kapatır.
- ➔ closeStage():Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.

### **TedaviGoruntuleController**

- ➔ initData():TedaviController'daki listeden seçilen tedaviyi GoruntuleController'da verileri doldurulmuş bir şekilde getirmeyi sağlar. Burada sadece label'lara tedavinin açıklamasını yazdırmak için bu metot kullanılmıştır.

### **HastaController:**

- ➔ getHastaFromFile(): Hastaların kaydının tutulduğu txt dosyasından hastaları çeker ve observableList'e yazar ardından bu listeyi döndürür.
- ➔ hastaDuzenleDialog(): Ara yüzde hasta menüsünde bulundan düzenle butonuna basıldığında listeden hasta seçilip seçilmediğinin kontrolünü yapar eğer hasta seçili ise hastaDuzenle.fxml i çalıştırır değilse uyarı verir.
- ➔ hastaEkleDialog(): Ara yüzde hasta menüsünde bulundan ekle butonuna basıldığında hastaEkle.fxml'i çalıştırır.
- ➔ initialize(): Hasta sınıfında bulunan özellikleri tablonun ilgili sütununa yazar. Ayrıca ara yüzde bulunan sil butonuna basıldığında seçilen hastayı siler. Eğer listeden hasta seçilmedi ise hiçbir işlem yapmaz.

## **HastaDuzenleController:**

- ➔ setHastaList(): Hasta contorllerda kullanılan listeyi bu sınıftaki liste ile eşitlemek için kullanılır.
- ➔ initData(): HastaController'da seçilen hastayı HastaDuzenleController'a özellikleri doldurulmuş biçimde getirmeyi sağlar.
- ➔ hastaDuzenle(): hastaDuzenle.fxml içerisindeki textboxları doldurur. Ardından HastaController'da seçilen hastayı siler ve textboxlar içerisindeki veriler ile yeni bir hasta nesnesi oluşturarak silinen nesnenin yerinde yazar ve listeyi yeniden dosyaya yazar.
- ➔ hastaDuzenleClose(): Düzenleme penceresinin kapanmasını sağlar.
- ➔ closeStage(): Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.

## **HastaEkleController:**

- ➔ hastaEkle(): hastaEkle.fxml içerisindeki textboxların doldurulmasını sağlar. Ardından buradaki veriler ile yeni bir hasta nesnesi oluşturup listeye ekler ve bu listeyi dosyaya yazar.
- ➔ setHastaList(): Hasta contorllerda kullanılan listeyi bu sınıftaki liste ile eşitlemek için kullanılır.
- ➔ closeStage(): Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.
- ➔ hastaEkleClose():İptal butonuna tıklanınca pencereyi kapatır.

## **RandevuController:**

- ➔ getRandevuFromFile(): Randevu kayıtlarının tutulduğu txt dosyasından çekerek listeye yazar ardından bu listeyi döndürür.
- ➔ Initialize(): Randevu sınıfında bulunan özellikleri tablonun ilgili sütununa yazar ve ara yüzde bulunan sil butonuna basıldığında seçilen hastayı siler. Eğer listeden hasta seçilmedi ise hiçbir işlem yapmaz. Ayrıca randevu penceresinin üst kısmında bulunan textbox'a girilen veri için arama işlemi yapar.
- ➔ randevuEkleDialog(): Ara yüzde hasta menüsünde bulundan ekle butonuna basıldığında randevuEkle.fxml'i çalıştırır.
- ➔ randevuDuzenleDialog(): Ara yüzde hasta menüsünde bulundan düzenle butonuna basıldığında listeden hasta seçilip seçilmediğinin kontrolünü yapar eğer hasta seçili ise randevuDuzenle.fxml i çalıştırır değilse uyarı verir.



### **RandevuEkleController:**

- ➔ setRandevuList(): Randevu contorllerda kullanılan listeyi bu sınıftaki liste ile eşitlemek için kullanılır.
- ➔ Initialize(): combobox'lardan herhangi bi veri seçili değilse seçilmesi gerektiğini belirtir.
- ➔ randevuKaydet(): Ara yüzde bulunan textbox, combobox ve datepicker içerisine veri girilmesini sağlar bu verile ile yeni bir randevu nesnesi oluşturur ve listeye ekler ardından listeyi dosyaya yazar.
- ➔ closeStage(): Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.

### **RandevuDuzenleController:**

- ➔ initData(): RandevuController'da seçilen hastayı RandevuDuzenleController'a özellikleri doldurulmuş biçimde getirmeyi sağlar.
- ➔ randevuDuzenle(): randevuDuzenle.fxml içerisindeki textboxlar, comboboxları ve datepicker'ı doldurur. Ardından RandevuController'da seçilen hastayı siler ve textboxlar, comboboxları ve datepicker içerisindeki veriler ile yeni bir randevu nesnesi oluşturarak silinen nesnenin yerine yazar ardından listeyi yeniden dosyaya yazar.
- ➔ closeStage(): Pencerenin Ekle ve İptal butonuna tıklanınca kapatması bu metotla sağlanır.