

|  |
| --- |
| **İSTANBUL SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ**  **MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ**  **YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ** |
| **YAZ201 – NESNE YÖNELİMLİ PROGRAMLAMA** |
| ***NESNE YÖNELİMLİ PROGRAMLAMA İLKELERİNE DAYANARAK BİR HASTANE İŞLEMLERİNİ HIZLANDIRMAK***  **Özgün Övül ASLAN-210609030**  **Ilgın Çutok-210609011** |
| **PROJE ÖDEVİ**  **OCAK 2023** |

**ÖZET**

Nesne yönelimli programlama dilinden biri olan javayı kullanılarak hastanedeki işlemleri mümkün olduğunca otomatikleştirmek ve hızlandırmak için bu projeyi tasarladık.

**Anahtar Kelimeler:** Doktor,hemşire,hasta,sekreter,laborant,nesne yönelimli programlama

**1. GİRİŞ**

**Bu çalışmanın amacı :** Özet kısmında belirttiğimiz gibi hastanede ağır işleyen sistemleri olabildiğince hızlandırmak.

**2. YAPILAN ÇALIŞMALAR**

Öncelikle nasıl bir yol izleyeceğimizi bilmediğimizden internet üzerinde araştırmalar yaptık. Farklı konulardan oluşan projelerle karşılaştık. Geçen sene Bilgisayar Programlama2 dersinde aklıma gelen fakat tek başına yapamadığım konu olan hastane içi iletişimle alakalı konuyu konuştuk. Bu konuda karar kıldık ve hastanede hangi çalışanların olduğu, kayıt yaptırırken nelere ihtiyaç var diye ufak bir araştırma yaptık. Bu projede serbest, istediğimiz gibi çalışabildiğimizden dolayı kod yazarken sıkıntı yaşamamak için ve kodu kısaltabilmek için internette, yazılım bloglarında ‘try-catch’ bloklarını gördük. Onlarla ilgili videolar izledik nasıl yapılacağını gördük ve ilk kodu D-410 no lu sınıfta yazdık. Bunların yanında liste oluşturmak için w3schooldan nasıl yapılacağına baktık ve öğrendik.

Bir çok sınıf kullanarak bunları nesne tabanlı programlamanın 4 temel ilkesini baz alarak UML şemasını oluşturduk. Oluşturduğumuz sınıfları tek tek yazmak gerekirse admin, arayüz, hasta, hemşire, main, sekreter, laborant ve doktor. Bu sınıfları tek tek açıklayalım.

Main sınıfında arayüz ve doktor için metodu oluşturduk.

Arayüze geldiğimizde ise bütün sınıfların metodlarını oluşturduk. İf, else-if ve else kullanarak hangi işlemleri hangi seçimleri kullandıysak ona göre ilerleyeceğimiz bir şekilde tasarladık. Yazdığımız tüm sınıfların arayüzü burada bulunmaktadır.

Hasta sınıfına geldiğimizde ise hastanın kişisel verilerine ait bütün özellikleri w3schooldan öğrendiğimiz gibi liste şeklinde tanımladık.

Hemşire sınıfında ise hastaları görüntülenebilmesi, silinmesi, eklenmesi, kontrolü gibi işlerin yapıldığı kodları yazdık.

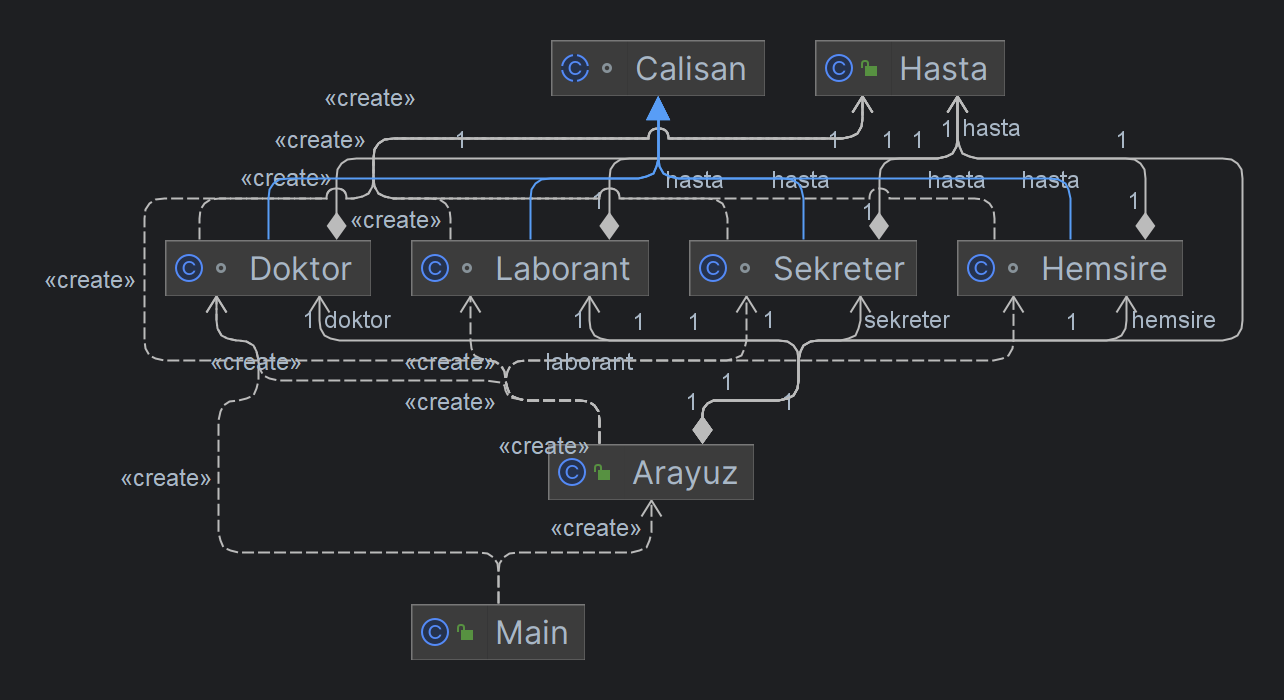
Laborantta ise hastaya ait verileri giriş yapabilmesi için hastaya ait kişisel bilgiler var. Laborant burada kan değerlerini sonuçlarını girebilecek.

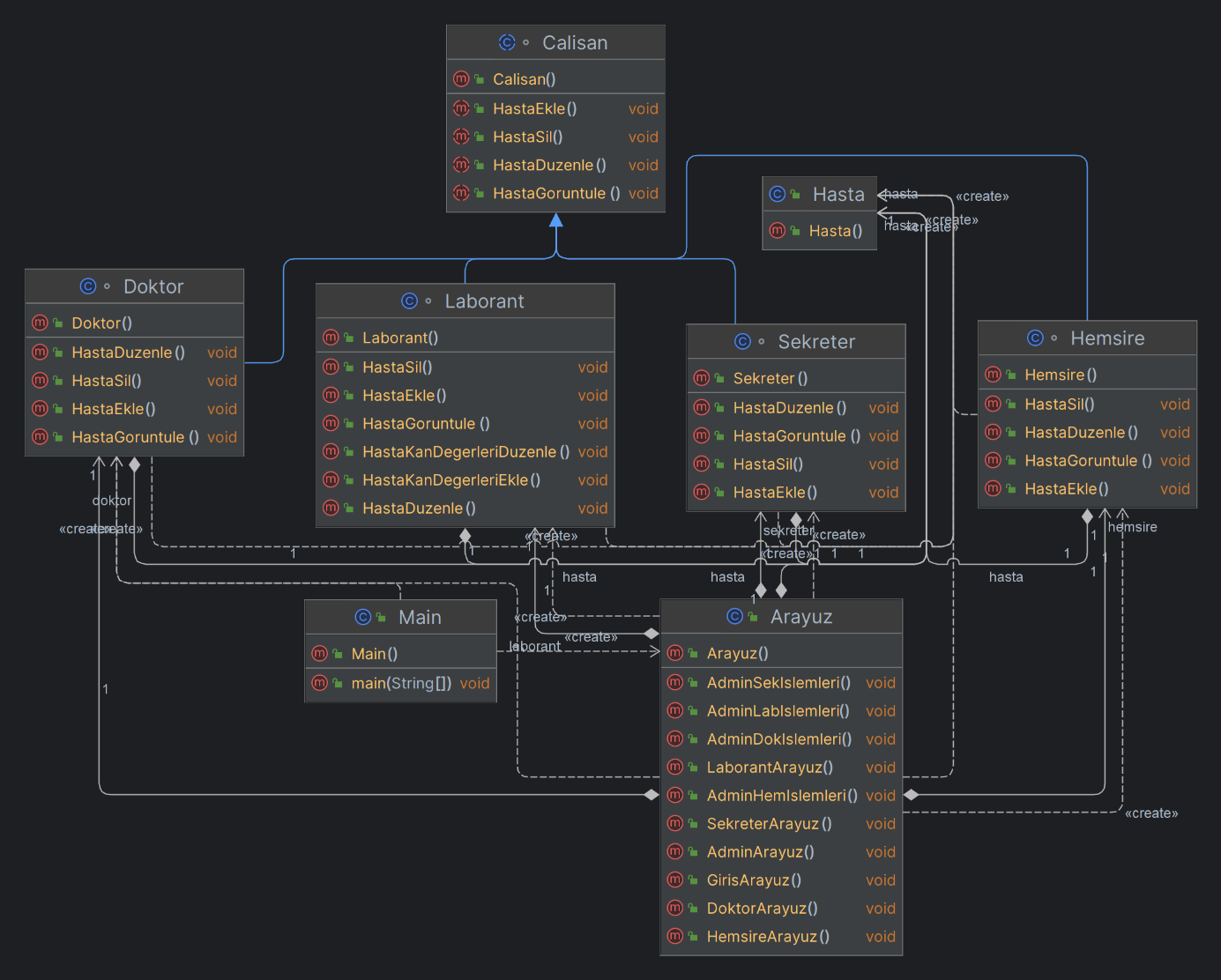
Çalışan sınıfında ise gerekli metodları yazdık.

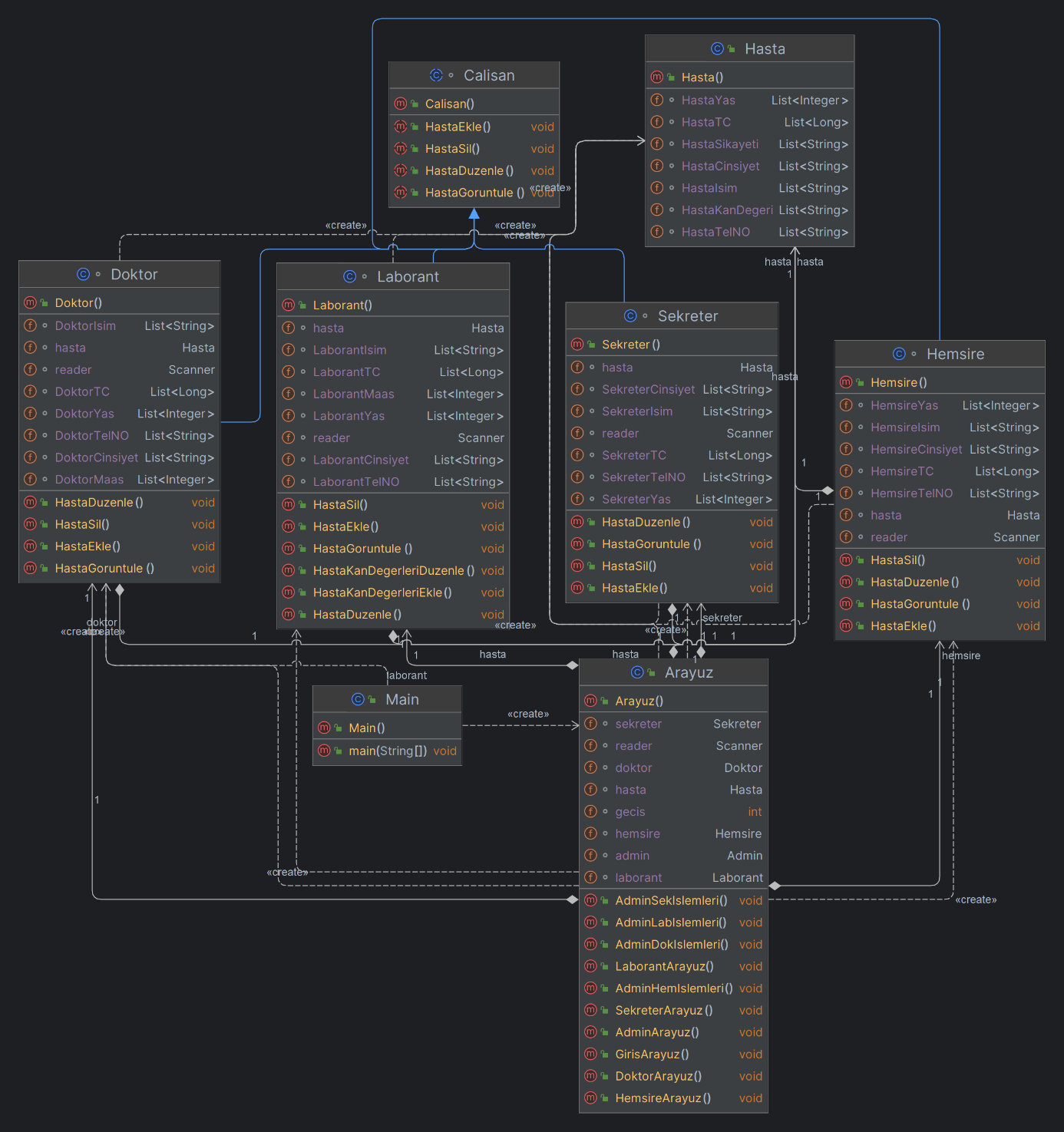
Doktor sınıfında ise doktora ait bilgilerin girileceği, hangi hastaya bakacağı ve hastaya ait bilgiler var.

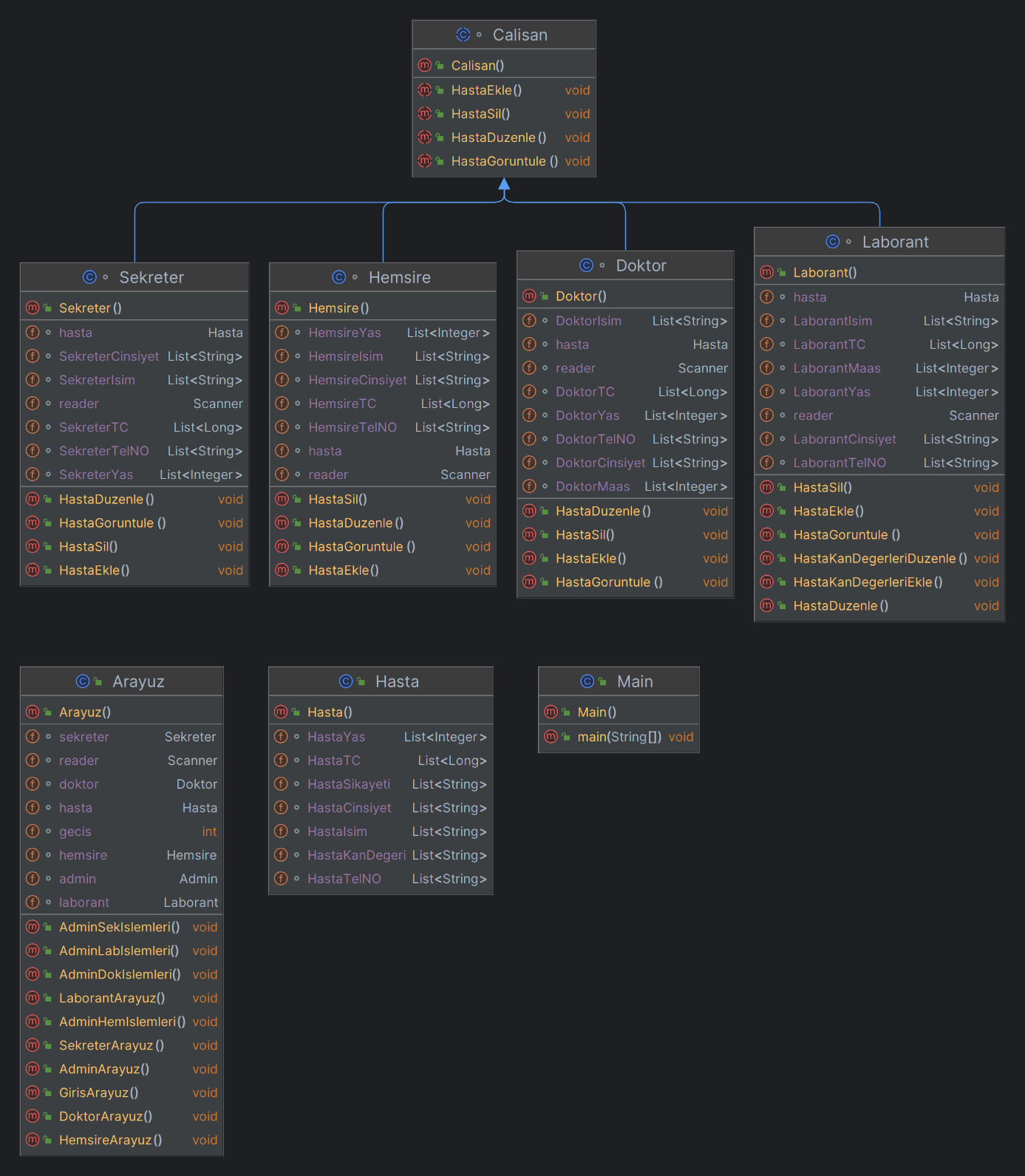
En sonuncusu ve en önemlisi olan admin sınıfı. Oluşturduğumuz diğer bütün sınıfların hepsini burada kullanacağız ve öğrendiğimiz try-catch yapılarını burada if, else-if, else ile burada kullandık.

Projeyi hayat geçirdikten sonra internetten araştırma yaprak UML tasarlayacağımız site bulduk ve UML tasarlamaya başladık. Tasarladığımız UMLler aşağıdadır.



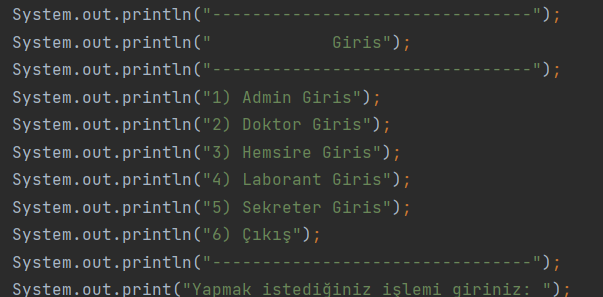






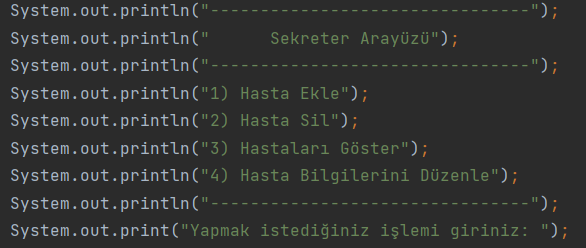
**3. BULGULAR**

*Bu bölümde uygulamaya ait ekran görüntüleri verilecektir.*

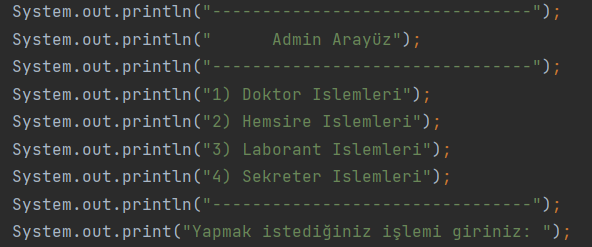


Uygulamayı başlattığımızda ilk olarak karşımıza burada yazdığımız gibi bu arayüz gelecektir. Ardından seçtiğimiz sayılara göre işlemler gelecektir.

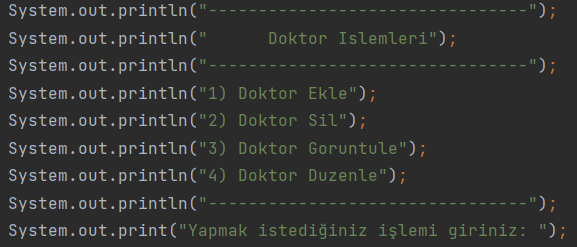
Örneğin 5 nolu işlemi seçtiğimizde çıkacak arayüz budur:



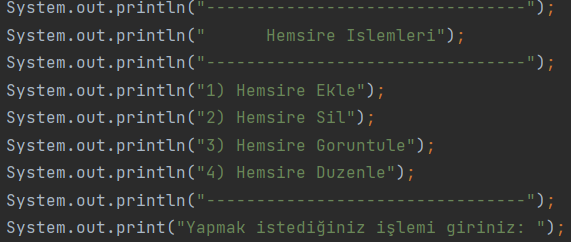
Fakat admin işlemlerinde durum farklı çünkü admin admin işlemleri,doktor işlemleri, sekreter işlemleri, laborant işlemleri yapabilmektir. En başa dönüp admin girişi seçtiğimizi varsayalım. Çıkacak ekran:



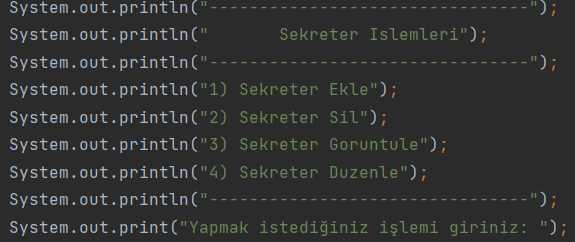
Admin doktor işlemlerinde çıkacak ekran:

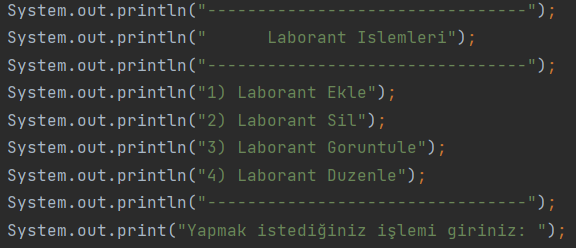


Admin hemşire işlemlerinde çıkacak ekran:



Admin sekreter işlemlerinde çıkacak ekran:



Admin laborant işlemlerinde çıkacak ekran: 

***4.TARTIŞMA VE SONUÇ***

Bu çalışmada günümüzde sağlık sektöründe yaşadığımız yavaşlığı hızlandırmak, işlemleri tek bir yerden kontol edip otomatik hale getirebilmek için bu tarz bir proje tasarlanabilir.

Bizim yaptığımız proje günümüzde kullanılan arayüz ve uygulamalara göre daha basit fakat düşünce olarak tek bir yerde toplama fikri gayet özgün.

5.KAYNAKÇA

1- <https://www.w3schools.com/java/>

2- <https://www.btkakademi.gov.tr/portal/course/java-ile-programlamaya-giris-9617>

3- <https://www.w3schools.com/java/java_try_catch.asp>

4- <https://www.javatpoint.com/try-catch-block>

5- https://www.ihs.com.tr/blog/java-hakkinda-10-ilginc-bilgi/