

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI**

**Hastane Otomasyon Sistemi**

**G181210030 - Mustafa Hasan Kuzulu**

**Bölüm** : **BİLGİSAYAR MÜHENDİSLİĞİ**  
**Danışman** : **Prof. Dr. Ümit KOCABIÇAK**

**2021-2022 Güz Dönemi**

## ÖNSÖZ

Günümüzde yazılım dünyasında oldukça fazla konuşulan yapıların başında microservisler gelmektedir. Bu bağlamda yazılım dünyasında yer alabilmek için bu mimari yapısını bilmek, microservis yapısı kullanarak projeler geliştirmek oldukça önemli hale gelmiştir. Şu anda büyük ölçekli yazılım geliştiren firmalar microservis yapısını sıklıkla kullanmaktadır.

Bitirme projemde microservis yapısını uygulayabildiğim bir hastane otomasyon uygulaması geliştirmenin mezun olduktan sonraki iş hayatımda bana oldukça yardımcı olacağına inanıyorum. Bu bağlamda microservis mimarisini uygulayabileceğim büyük ölçekli bir uygulama yapmaya karar verdim. Bu hedefimi gerçekleştirmek için hastane otomasyon sistemi geliştirmek üzere çalışmalarına başladım.

## İÇİNDEKİLER

İÇİNDEKİLER .....	3
ÖNSÖZ .....	4
GİRİŞ.....	5
1.1.Backend Microservis mimarisi.....	6-7
1.2.Frontend React.js Uygulaması.....	8-9
Uygulama .....	
Security – Authentication -Authorization	10-12
Çok dillilik	13
Admin Kullanıcısı Yetkileri	13-14
Randevu Alma Sistemi	15-17
Doktor Profil Sayfası	18
Hastane Bölüm Sayfası	19-20
Mikroservis iletişim yöntemleri	21-22
Olasılık Etki Matrisi	23-24
Proje Başarı Kriterleri	25-27
Sonuç	28
Kaynakça	29
Özgeçmiş .....	30
Bilgisayar Mühendisliği Tasarımı Değerlendirme ve Sözlü Sınav Tutanağı	31

## ÖZET

Anahtar kelimeler: microservis, frontend,backend,react.js,spring boot, hystrix, zipkin

Bu tasarım çalışması ile backend tarafında Spring boot ve Spring could , frontend tarafında ise React.js kullanarak tamamlanmış bir hastane otomasyon sistemi geliştirmek hedeflenmektedir. Uygulama geliştirilirken mikroservis mimarisi kullanılmaya çalışılacaktır.

Microservis mimarisi günümüzde oldukça sık kullanılan bir yapı haline gelmiştir. Genellikle büyük ölçekli yazılım projelerine daha uyumlu olan bu mimariyi kullanarak bir tasarım projesi gerçekleştirmek istenmiştir.

Microservis mimarisinin getirdiği modüler mimari avantajlarını kullanarak proje birbirinden bağımsız parçalara bölünerek, ayrı olarak deploy edebilme, servisler arasındaki iletişimin sağlanması, ayrı servislerde farklı teknolojiler kullanarak çok daha güçlü servisler oluşturmak gibi avantajlarda faydalanmak hedeflenmektedir. Aynı zamanda servisler arasındaki iletişimi daha güçlü hale getirebilmek güncel teknolojilerden faydalanacaktır.

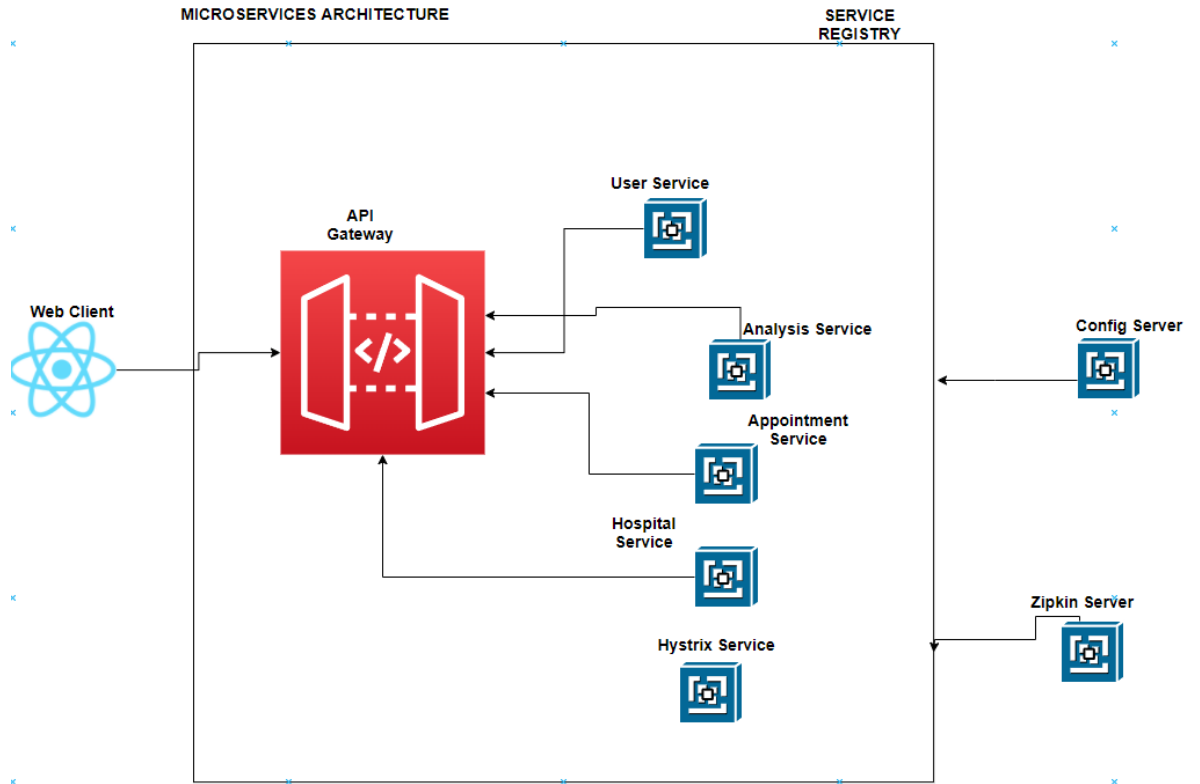
Bu projeyi tamamlayarak günümüzde oldukça popüler olan microservis mimarisini özelliklerini iyi bir şekilde öğrenmiş ve uygulamış olacağım. Aynı zamanda geliştirdiğim hastane otomasyon sistemi ile büyük kurumlarda geliştirilen sistemlerin özellikleri, analizleri hakkında bilgi sahibi olacağım.

## BÖLÜM 1. GİRİŞ

Geliştirilmesi hedeflenen bir hastane otomasyon sistemidir. Bu sistem backend tarafında spring boot , frontend tarafında ise React.js kullanılarak geliştirilecektir. Backend tarafında mikroservis mimarisi kullanılmıştır.

İlk olarak Backend tarafında kullanılan mikroservis mimarisindeki özellikleri ardından ise frontend tarafında kullanılan React.js kütüphanesi ve web sayfaları hakkında bilgiler verilmiştir.

### 1.1. Backend – Microservis Mimarisi



#### 1.1.1 API Gateway

Uygulamada web sayfasında gelen istekler ilk olarak API Gateway e uğrayacak, burada **Authentication – Authorization , Routing , Logging** Gibi işlemler yapılarak diğer servislere yönlendirilecekler.

### 1.1.2 Service Registry

Service Registry bütün microservislerin mevcutta host edildikleri network lokasyonlarını tutan merkezi bir bileşendir. Microservisler tarafından **registry**'e kaydı bulunan ve network communication yapmak istediği **service** 'in adresini **registry** 'den sorgulayarak kullanırlar.

### 1.1.3 User Service

Bu service kullanıcı işlemlerinden sorumlu olacaktır. Kullanıcı kaydetme, güncelleme işlemleri, kullanıcı sorgulama işlemleri bu servis üzerinden yapılacaktır.

### 1.1.4 Analysis Service

Analysis Service hastaların hastanede yaptıkları tahliller ve test sonuçlarını, gördükleri tedavi bilgileri ile alakalı işlerden sorumlu olacaktır. Bu servis yardımıyla hastalara ait her türlü test ve tahlilin sorgulaması yapılabilecektir.

### 1.1.5 Analysis Service

Appointment Service hastaların doktorlardan aldığı randevu işlerinden sorumlu olacaktır. Aynı zamanda hastanelerde genelde doktorların kapısında bulunan randevu izleme sistemini de içerecektir. Bu servis yardımıyla hastalar randevu alabilecekler ve bu randevularını takip edebileceklerdir.

### 1.1.6 Hospital Service

Hospital Service hastane ile alakalı genel verilerden sorumlu olacaktır.

### 1.1.7 Hystrix Service

**Hystrix**, ilgili servislerdeki başarısız istekler için dinleme yapmaktadır. Böyle bir sorun varsa, devreyi “**circuit open**” yaparak, yani devreyi açarak çağırıcıyı “**fallback method**” olarak yazdığımız bir geri dönüş metoduna iletecektir. Bu sayede bir servis düştüğünde tüm servisler bundan etkilenmeyecek gerekli önlemler alınarak sistemin devamlılığı sağlanacaktır.

### 1.1.8 Config Service

Config server uygulamadaki konfigürasyon ayarlarının tutulmasından ve bu ayarların dinamik olarak değiştirilebilmesinden sorumlu olacaktır. Bu sayede bir her ayar değiştirdiğimizde tekrar deploy yapmamız gerekmeyecek onun yerine tüm servisler Config servis aracılığıyla güncellenecektir.

### 1.1.9 Zipkin Service

Zipkin service bize servisler aracığında gerçekleşen işlemlerin takip edilmesinde yardımcı olacaktır. Zipkin sayesinde hangi servisin, hangi servisle iletişim kurduğunun anlayabileceğiz ve doğru loglamalar sayesinde de isteklerin takibini yapabileceğiz.

## **1.2. Frontend – React.Js**

Uygulama için React.js kullanarak bir arayüz geliştireceğim. Bu arayüz üzerinden uygulamamız backend e bağlanarak işlemleri gerçekleştirecektir.

### **1.2.1 React.js Uygulamasında Kullanılacak bazı teknolojiler**

**Material UI**

**React Router**

**React Notification**

**Axios**

### **1.2.2 Uygulamada Bulunacak Web Sayfaları**

#### **1.2.2.1 Anasayfa**

Anasayfa da hastane hakkında bilgiler ve kısa tanıtım yazıları, yol tarifleri bulunacaktır.

#### **1.2.2.2 İletişim Sayfası**

Bu sayfada hastanenin iletişim bilgileri bulunacaktır.

#### **1.2.2.3 Doktorlarımız Sayfası**

Bu sayfada hastanedeki doktorlar görüntülenebilecek ve branşlarına göre arama yapılabilir.

#### **1.2.2.4 Bölümlerimiz Sayfası**

Bu sayfada hastanedeki bölümler görüntülenebilecek ve bu bölümlerde çalışan doktorlar görüntülenebilecek. Aynı zamanda bölüm hakkında bilgi de verilecektir.



#### **1.2.2.5 Login sayfası**

Uygulama üzerinde üye olma sistemi olacaktır. Üyeler online olarak hastaneden randevu alabilecekler, geçmişte aldıkları randevuları kontrol edebilecekler, tahlil sonuçlarını online olarak görebileceklerdir.

#### **1.2.2.6 Doktor login sayfası**

Uygulama üzerinde doktor olanlar farklı yetkilerle farklı sayfalara erişim sağlayabilecekler. Doktorlar kendi arayüzlerinde randevu alan hastaları görebilecek, hastaların bilgilerine erişim sağlayabileceklerdir.

#### **1.2.2.7 Admin login sayfası**

Uygulama üzerinde admin yetkisi olanlar farklı yetkilerle farklı sayfalara erişim sağlayabilecekler. Admin yetkisine sahip kullanıcılar hastane ile alakalı tüm bilgileri değiştirebilecekler, bunu yanında hastaneye doktor ekleyebilecek ve doktorlar ile ilgili tüm güncellemeleri ve değişiklikleri yapabilecekler.

Aynı zamanda admin olan kullanıcılar uygulama tarafındaki veriler ile oluşturulmuş raporları görüntüleyebilecekler, bu raporları kullanarak kullanıcı deneyimini arttırmak üzere işlemlerde bulunabileceklerdir.

#### **1.2.2.8 Kullanıcı profil sayfası**

Bu sayfada kullanıcılar kendi bilgilerini görebilir ve daha önceden yaptıkları randevuları görebilirler. Kullanıcılar dilerse randevularını iptal edebilirler.

Aynı zamanda kullanıcılar hastane tahlil sonuçlarını da yine bu ekran üzerinden görebilirler.

#### **1.2.2.9 Doktor profil sayfası**

Bu sayfa üzerinden doktorların profilleri incelenebilir. Doktor hakkında bilgiler veren bu sayfa üzerinde doktroların hasta randevu ekranları da bulunmaktadır. Bu sayede doktorun hangi saatte hangi hastaya randevusu olduğu görülebilmektedir.

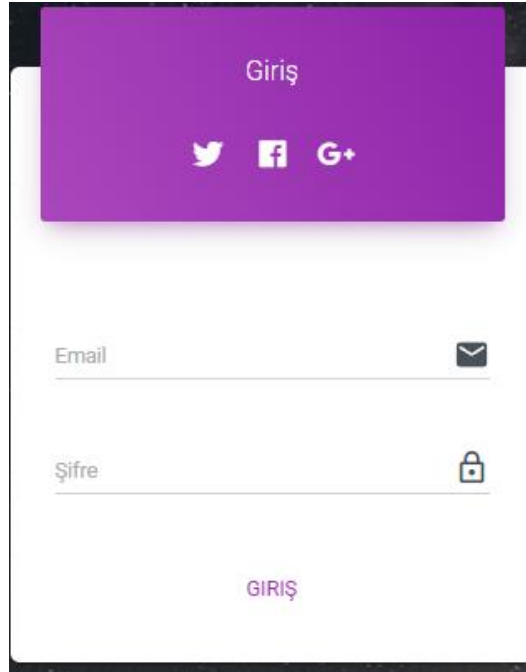
## Uygulama Authentication ve Authorization

Uygulamamız üzerinde birden fazla rol bulunmaktadır. Bu rollerin ayrımı ve yetkilendime işlemleri için Spring boot içerisindeki Spring Security kütüphanesini kullanıldı. Bununla beraber client üzerinde login olana kullanıcı JWT teknolojisi ile isteklerinde bu token kullanarak Authentication ve authorization gerçekleştirebilirler.

### Nasıl Çalışır?

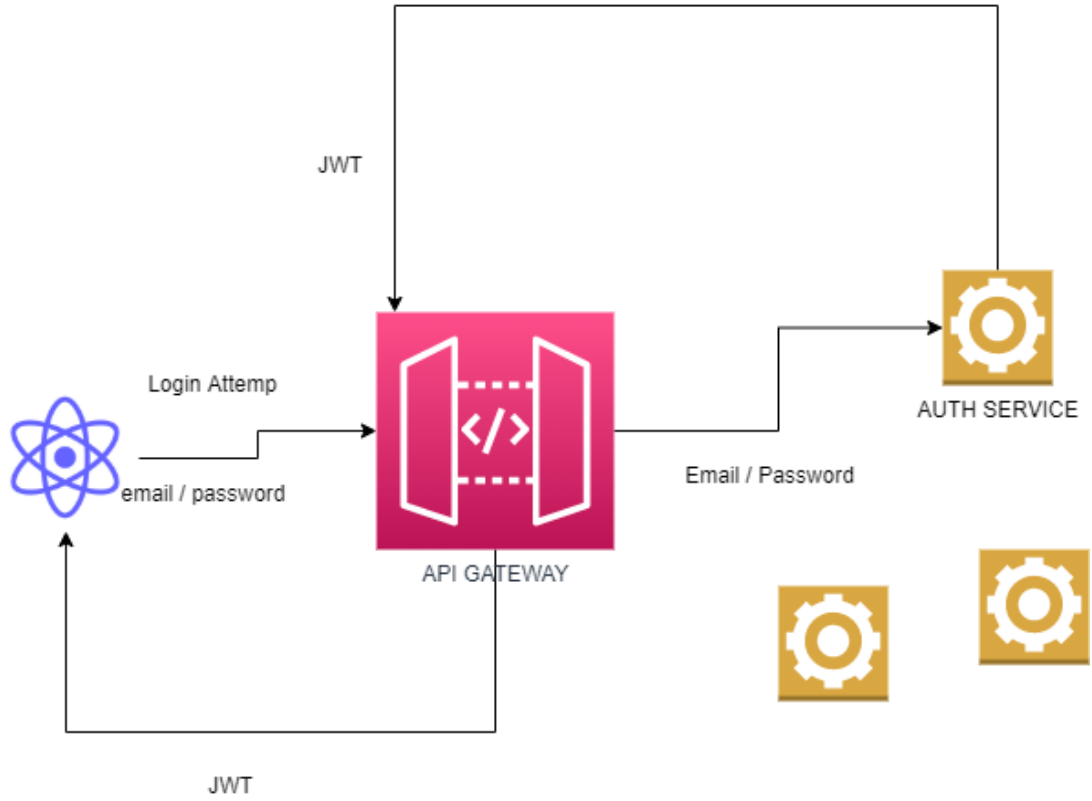
İlk olarak yeni bir servis eklendi Auth Service, bu servisin görevi kullanıcı adı ve şifre ile gelen kişinin yetkilerini bulmak ve onu login etmek. Eğer kullanıcı başarılı bir giriş yaptıysa geriye bir token döndürür ve kullanıcı bu tokeni kullanarak diğer isteklerinde de login olabilmeyi sağlar.

Kullanıcıların tüm istekleri Api Gateway üzerinden geçer. Api gateway üzerinde yapılan tanımlamalar sayesinde hangi url'lerin izinli hangilerinin yetki gerektiği belirtilebilir. Bu sayede kullanıcılar login olmadan diğer kaynaklara erişemezler. Aynı zamanda API gateway kullanıcının yetkilerini de diğer servislere bildirir bu sayede yetki kontrolü de sağlanmış olur.

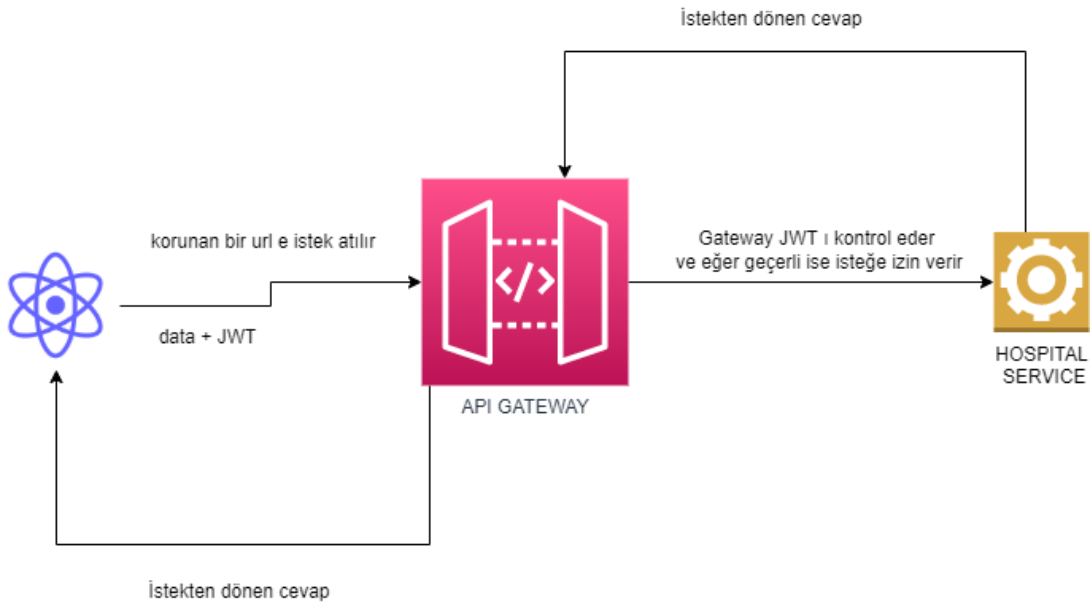


The image shows a mobile application login screen. At the top, there is a purple header with the word "Giriş" (Login) in white. Below the header, there are three social media icons: Twitter, Facebook, and Google+. The main area of the screen is white and contains two input fields. The first field is labeled "Email" and has a small envelope icon to its right. The second field is labeled "Şifre" (Password) and has a small lock icon to its right. At the bottom of the screen, there is a purple button with the word "GİRİŞ" (Login) in white.

## Login İsteği Backend



## Korunan bir kaynağa atılan istek örnek



Gateway üzerinde hangi URL'lerin authentication gerektirdiği tanımlıdır. Bazı URL'lere ise izin verilebilmektedir. Bu sayede kullanıcılar login isteğini token olmadan yapabilirler. Ama token a sahip olmayan bir kullanıcı yani login olmamış bir kullanıcı randevu oluşturamaz.

## Çok dil Özelliği

Uygulama üzerinde çok dil özelliği bulunmaktadır. Arayüz tarafında bu özellik 18n-react kütüphanesi kullanılarak sağlanmıştır.

Dil değiştirme işlemi arayüz üzerindeki menüden yapılabilir.



TR ve EN butonlarına basarak dil değişimi yapılabilir

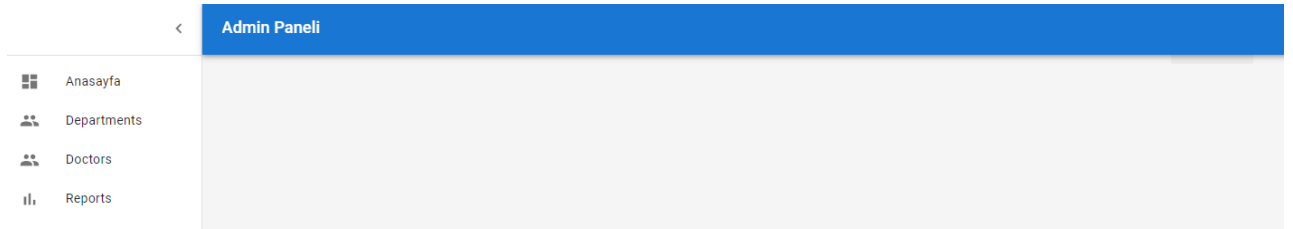
İngilizce Örnek:



## Admin Kullanıcısı Yetkileri

Uygulamada admin kullanıcılar bulunmaktadır. Bu kullanıcılar admin paneline erişebilir ve buradan hastane hakkında işlemler yapabilirler. Bu işlemlere örnek vermek gerekirse, hastane bilgilerini güncelleyebilme, doktor ve hasta bilgilerine erişebilme, hastaneye yeni departmanlar ekleyip bu departmanları güncelleyebilme vb.

Admin kullanıcılar menü üzerindeki DASHBOARD butonuna tıklayarak admin paneline giriş yapabilirler. Bu link normal kullanıcılar için gözükmez ve erişilemezdir.



Admin kullanıcılar dashboard üzerinde sol taraftaki linkler yardımıyla modüller arasında geçiş yapabilirler.

Admin Paneli							
DOKTORLAR							
ID	İsim	Soyad	Bölüm	Uzmanlık	email	Actions	
21	Ayşe	Yaman	Kulak burun boğaz	dr	ayse.yaman@gmail.com		
17	hasan mustafa	KUZULU	Kulak burun boğaz	prof. dr	hasan.kuzulu@gmail.com		
22	Merve	Demir	Göğüs Hastalıkları	prof. dr.	merve.demir@gmail.com		
20	Mehmet	Özel	Göğüs Hastalıkları	doc.	mehmet.ozel@gmail.com		
5 rows   1-4 of 4							

Örnek ekranda Doktorlar sayfasına gelinmiştir. Buradan doktor ekleme / silme / güncelleme / listeleme işlemleri yapılabilmektedir. Admin kullanıcı bu ekrana kullanarak sisteme kayıt ekleyebilirler.

### Doktor Ekleme Ekranı

Bir isim giriniz

Adı

Soyad Giriniz

soyadı

Bölüm Seçiniz



Uzmanlık Alanı Giriniz

uzmanlık

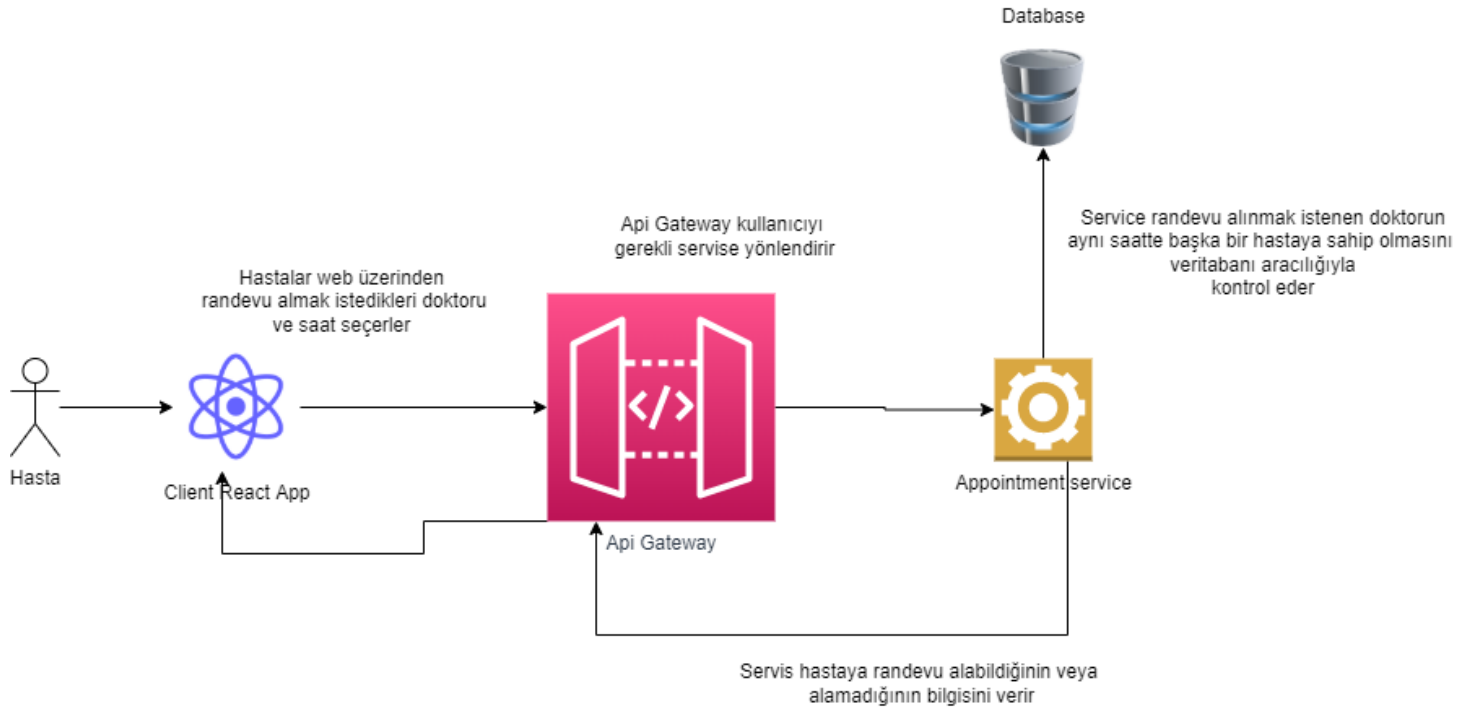
Email Adresi Giriniz

email adresi

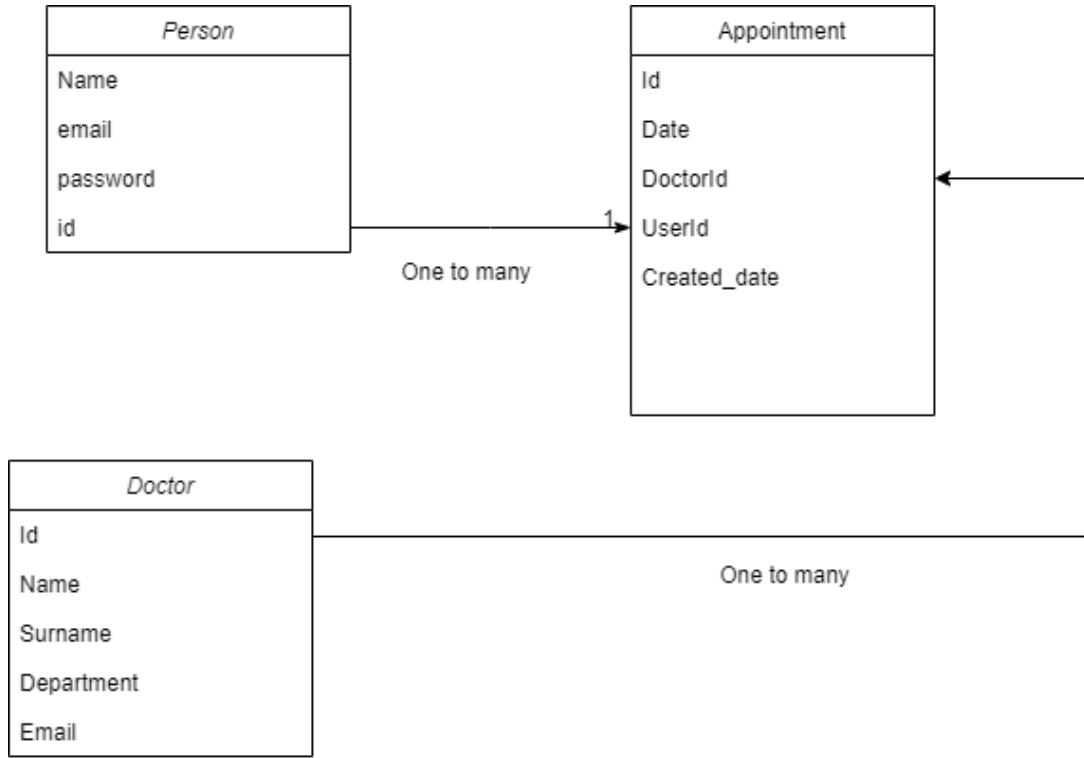
KAYDET

## Randevu alma Sistemi

Geliştirilen sistemde hastanemiz online olarak randevu verebilecektir. Bunu sağlamak için hastalarımızın site üzerinden kayıt oluşturarak giriş yapmaları gerekmektedir. Ardından doktorların profillerinden randevu alabileceklerdir.



Hastalar online sistem üzerinden randevu alabilmektedir. Bir hasta randevu almaya çalıştığında aynı anda randevu alan başka bir hastayla karışmaması için backend tarafında veritabanından kontrol edilir. Her hasta randevusu veritabanına bir kayıt olarak yazılmaktadır.



### Arayüz Üzerinde Randevu Sistemi

Sistem üzerinde randevular her saat için 10'ar dakikalık olmak üzere, (öğle saati hariç) ayarlanmıştır. Giriş yapmış olan kullanıcılar randevu sayfasından istedikleri bölüm ve doktora göre randevu alabilmektedir.

The screenshot shows the 'Randevu sayfası' (Appointment page) interface. It features a title 'Randevu sayfası' and a dropdown menu labeled 'Bölüm Seçiniz' (Select Department) with the text 'Lütfen bir seçim yapınız' (Please make a selection). The dropdown is currently open, showing a list of departments.

Bölüm seçtikten sonra o bölümde olan doktorların listesi gelir.

### Randevu sayfası

The screenshot shows the 'Randevu sayfası' (Appointment page) interface after a department has been selected. It displays a list of doctors available for appointments. The first doctor listed is 'Merve Demir prof. dr.' and the second is 'aydın çelik doc.'. Each doctor's name is accompanied by a circular profile icon and a calendar icon.



Ardında doktoların adının yazıkları yerin yanındaki takvim butonuna tıklayarak doktorların randevu sayfasına gidilebilir.

### Randevu sayfası

DOC. AYDIN ÇELİK

GÖĞÜS HASTALIKLARI

Friday 2022-01-07	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Monday 2022-01-10	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Tuesday 2022-01-11	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Wednesday 2022-01-12	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Thursday 2022-01-13	09:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00

[RANDEVU OLUŞTUR](#)

Randevu sisteminde en fazla 5 gün sonrasına randevu verilebilmektedir. Kişi bölüm ve doktor seçtikten sonra randevu saati seçmelidir. Seçilen randevu saatinde uygun randevular varsa onları seçip “Randevu Oluştur” butonuna tıklayarak randevu oluşturulabilir.

Friday 2022-01-07

09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00

10:00

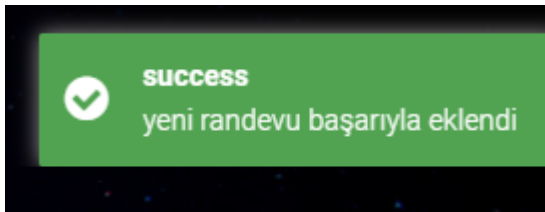
10:10

10:20

10:30

10:40

10:50



Friday 2022-01-07

09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00

10:00

10:10

10:20

10:30

10:40

10:50

Randevu ekledikten sonra aynı randevu başkaları tarafından alınamaz ve buton disable olur.

Kullanıcılar aldıkları randevuyu profillerinde görebilirler ve iptal edebilirler.



**aydın çelik**

Göğüs Hastalıkları

randevu saatiniz : 10:20 , 2022-01-07

İPTAL

### Doktor Profil Sayfası

Bu sayfada doktorların profilleri incelenebilmektedir. Aynı zamanda doktorların randevu takvimi de gözükmemektedir. Bu sayfa aracılığıyla doktorların hangi hastayla ilgilendiği gözlenebilir.



**aydın çelik**

MEHMET.OZEL@GMAIL.COM

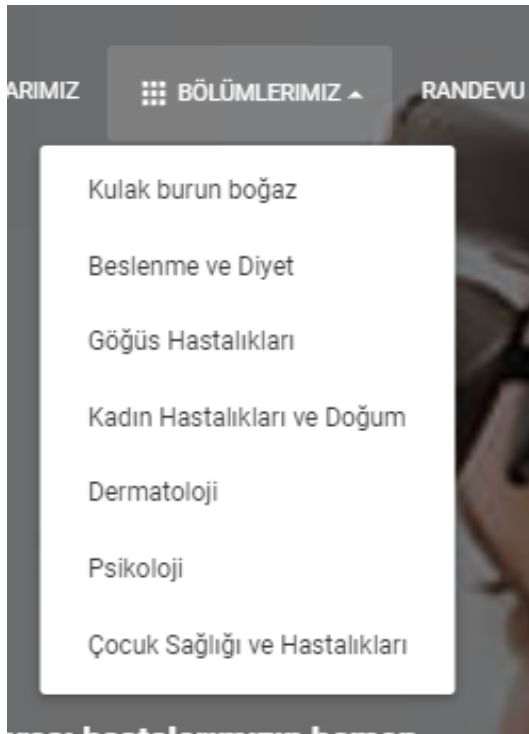


## Günlük Randevu Ekranı

Adı	Soyadı	randevu saati
mehmet	demir	10:20
hasan	keskin	11:30
ayşe	kılıç	14:40
Efe	gök	16:50

## Hastane Bölüm Sayfaları

Hastanedeki her bölüm için ayrı bir sayfa bulunmaktadır. Bölümler admin paneli üzerinden eklenebilmektedir. Bölümlerin sayfasına girebilmek için menü üzerinden “bölümler” sekmesine tıklayıp ilgili bölüm seçilmelidir.



Bu sayfada bölüm hakkında bilgiler bulunur ve alt tarafta bu bölüme ait olan doktorların listesi bulunur.



## Kulak burun boğaz

Poliklinik, laboratuvar, radyolojik tetkikler ve odiyolojik tetkikler sonrası hastalarımızın hemen tamamında hastanelerimizde tanı koyup tedaviye geçilmektedir. KBB muayene odaları klasik muayene gereçlerinin yanı sıra endoskopik muayenelere de olanak veren endovizyon sistemi ile donatılmıştır.

Ameliyathanelerimizde hastanın konforu ve başarılı bir ameliyat sürecine katkı sağlayacak gerekli cerrahi setlerle donatılmıştır. Bu setler, endoskopik sinüs cerrahisi, orta kulak ve mastoid cerrahisi alanlarındaki hekim pratiğine katkı sağlayan donanımlar olarak ön plana çıkmaktadır.

Ameliyathanelerimizde baş ve boyun bölgesinin büyük kanserlerinin ameliyatları da başarıyla gerçekleştirecek tıbbi donanım, hekim kadrosu ve yoğun bakım hizmetleri bulunmaktadır.

▶ WATCH VIDEO

## Doktorlarımız



**Ayşe Yaman**

dr

Kulak burun boğaz

PROFILE GİT



**hasan mustafa KUZULU**

prof. dr

Kulak burun boğaz

PROFILE GİT

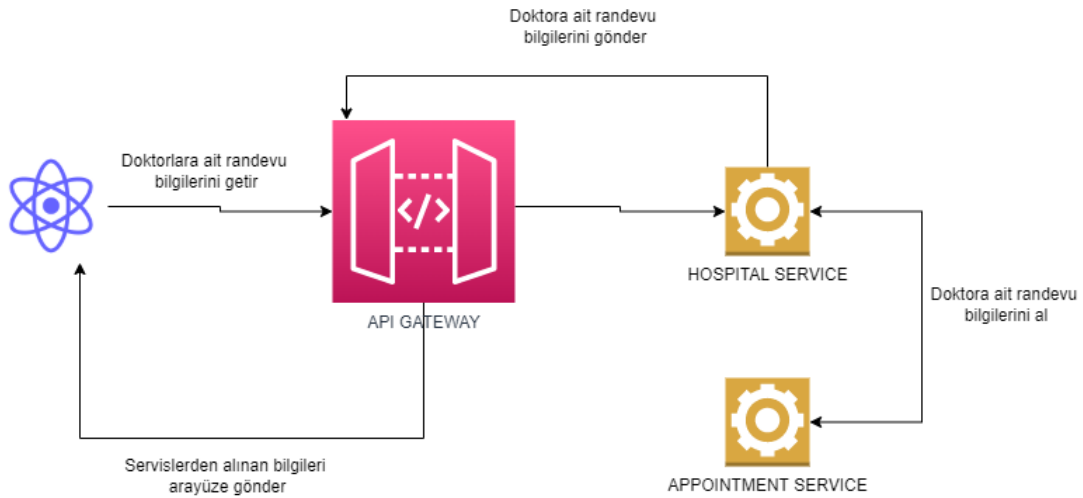


**Bu kısımdaki “profile git” butonu aracılığıyla doğrudan doktorların profillerine gidilebilmektedir.**

## Microservis mimarisi iletişim örnekleri

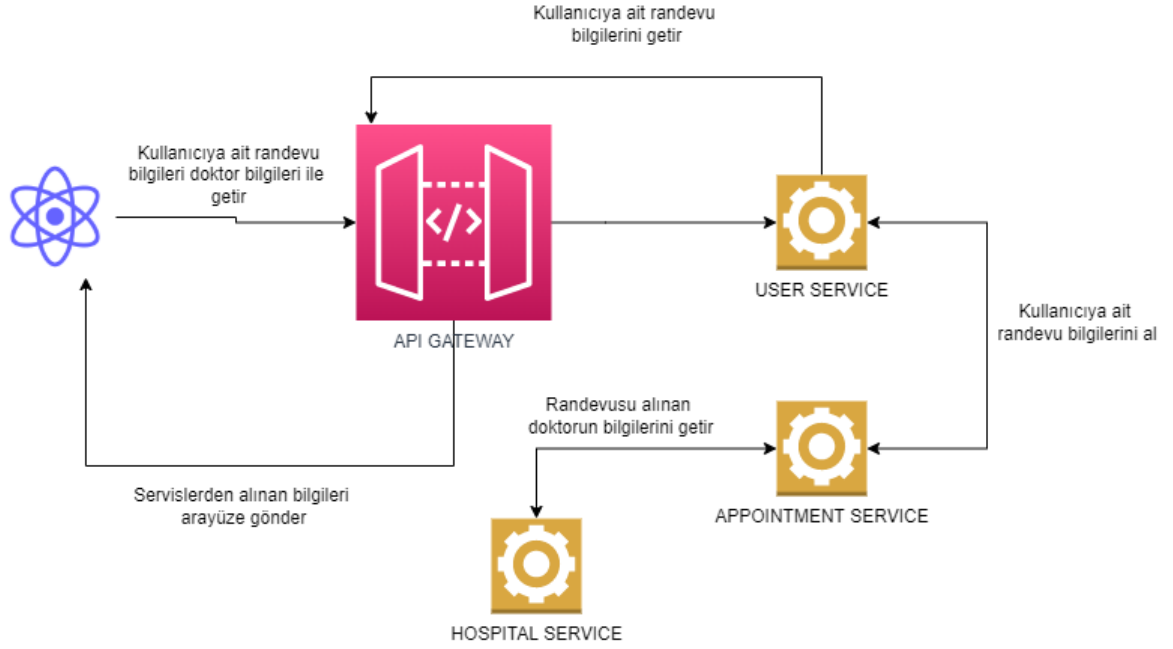
Uygulamamızda backend tarafında microservis mimarisi kullanılmıştır. Servislerin birbiriyle haberleştiği birçok örnek geliştirilmiştir. Bunlardan birkaçı aşağıda örnek olarak verilmiştir.

### Doktora ait randevular nasıl alınabilir ?



Örnekte ifade edildiği üzere arayüz tarafından doktor bilgilerini aynı doktora ait randevu bilgileri ile getirilmek için bir istek atıldığında servis tarafında ilk olarak istek "hospital service" e gelir. Buradan doktor bilgisi alınır ve ilgili doktora ait randevu bilgilerini alabilmek için "appointment service" e istek atılarak randevu bilgileri alınır ve sonuçta elde edilen veri arayüze gönderilir.

## Kullanıcıya ait randevu bilgileri Doktor bilgileri ile nasıl alınır?



Arayüz tarafında kişiye ait randevuları ve bu randevunun ait olduğu doktorun bilgilerini almak için izlenen yol yukarıdaki şemada gösterilmiştir. Servisler kendi aralarında iletişim kurarak her servis ilgili olduğu veriyi sağlar ve elde edilen veri API GATEWAY aracılığıyla client tarafına gönderilir.

## Olasılık Etki Matris

### Etki Derecelendirme Tablosu

ETKİ	DERECELENDİRME	PUAN
ÇOK HAFİF	Sistemde bir aksama yok –basit teknik destek	1
HAFİF	Sistemde aksama var - Yoğun Teknik destek	2
ORTA	Sistem kullanımında sıkıntı var - Müdür destekli bakım	3
CİDDİ	Servislerden biri çökmüş ya da kullanılamaz halde - Uzun süreli Destek	4
ÇOK CİDDİ	Tüm Servisler kullanılamaz halde – felaket durumu	5

### Olasılık Değerlendirme Tablosu

OLASILIK	PUAN
Neredeyse Mümkün Değil (Yılda bir)	1
Az olasılıkla (Yılda Birkaç Key)	2
Olasılık Dahilinde (ayda bir)	3
Yüksek olasılık (Haftada bir)	4
Kaçınılmaz	5

## OLASILIK ETKİ MATRİSİ

ETKİ

PUAN	1	2	3	4	5
2	1	4	6	8	10
3	3	6	9	12	15
4	4	8	12	16	20
5	5	10	15	20	25

OLASILIK

SONUÇ

⇒ 1 ve 2 puan

Kabul edilebilir durumda olan , ciddi önlem gerektirmeyen risk

⇒ 3, 4 ve 6 puan

Dikkat edilmesi gereken, Uzun dönemde önlem alınabilecek risk

⇒ 5, 8, 9 ve 10 puan

Ciddi derece önemi olan, Kısa sürede önlem alınması gereken

⇒ 12, 15 ve 16 puan

Yüksek derecede önemli, Derhal önlem alınması gereken risk

⇒ 20 ve 25 puan

Kabul edilemez, Önlem alınmadan işe başlanmaması gereken risk



## **Başarı Kriterleri**

### **1. Tüm servisleri hatasız bir şekilde çalışıyor olması**

İlk başarı kriterimiz geliştirdiğimiz uygulamadaki servislerin başarılı bir şekilde çalışıyor olması. Bunu sağlamak için geliştirdiğimiz uygulama üzerinde unit testler, integration testler uygulayarak servislerimiz her şart altında başarılı bir şekilde çalıştığından emin oluyoruz.

Uygulamamızın omurgasını oluşturan ve veritabanı ile arayüz arasındaki bağlantıyı sağlayan bu servislerin sorunsuz çalışıyor olması en önemli başarı kriterlerimizden birisidir.

### **2. Uygulamanın güvenli olması**

Son zamanlar oldukça artan siber tehditler yazılım dünyasındaki endişeleri de arttırmış bulunmaktadır. İş yaptığımız kurumun verilerinin güvenliğini sağlayabilmek için geliştirilen yazılımın oldukça güvenli olması gerekmektedir. SQL injection gibi kullanıcı arayüzünden backend tarafına yapılan saldırıların engellenmesi ve kullanıcı deneyiminin korunması gerekmektedir. Bu bağlamda geliştirilen uygulamanın güvenli olması da başarı kriterlerimiz arasında yer almaktadır.

### **3. Uygulamada çalışan arayüzün anlaşılabilir olması**

Uygulamamız için geliştirdiğimiz arayüzün de kritik bir önemi bulunmaktadır. Kullanıcı deneyimi açısından doğru bir şekilde geliştirilmiş, UX/UI araştırmaları yapılmış bir arayüze sahip olmak gerekmektedir. Kullanıcılar geliştirilen arayüz üzerinden uygulama ile iletişime geçecekleri için anlaşılabilir ve hatasız olması kritik bir öneme sahiptir, aksi taktirde uygulama ne kadar iyi olursa olsun kullanıcı gerekli etkileşimi yapamayacak ve başarılı bir deneyim yaşayamayacaktır.

#### **4. Müşteri memnuniyeti**

Müşteri memnuniyeti, müşterilerin yapılan işi benimsemesi ve etkili bir şekilde kullanabilmesi için oldukça önemlidir. Uygulama geliştirilmeye başlamadan önce müşteri ile iletişime geçilmeli ve istekleri ayrıntılı bir şekilde not alınmalıdır. Uygulama geliştirilirken müşteri tarafından gelen isteklerin dışına çıkılmamalı, yazılım geliştirme ekibinin kendi mantığıyla düşündüğü geliştirmeler kullanıcıya danışmadan ve onayı alınmadan gerçekleştirilmemelidir. Müşterinin memnun olmadığı bir iş başarılı bir iş olarak görülmez bu bakımdan müşteri tarafının isteklerini yerine getirdiğimize emin olmamız gerekmektedir.

#### **5. Zamanında Tamamlanmış olması**

Proje başlamadan önce yapılan analizlerde projenin geliştirilme süresi iyice belirlenmeli, bu süre içerisinde bitirmek için gayret gösterilmelidir. Bu sürenin esnememesi adına müşteri tarafından sonradan gelen ek istekler değerlendirilirken projenin tamamlanma süresi düşünülmeli gerekirse ek süre istenmelidir. Projenin tamamlanması süresinin uzaması hem müşteri açısından hemde yazılım geliştiren ekip açısından ciddi problemler ortaya çıkabilir. Bu sebeple zamanında tamamlanmış olması ciddi derecede önem taşır.

#### **6. Maliyeti Aşmamış Olmak**

Uygulama geliştirme süreci kendine ayrılan bütçeyi aşmamalıdır. Bunun için proje başlamadan önce proje üzerindeki analizlerin dikkatlice yapılması gerekmektedir. Maliyetin aşılması müşteri memnuniyetini kötü şekilde etkileyeceği gibi aynı zamanda müşteri ile olan iletişimi de kötü şekilde etkileyerek yapılan işin kalitesine ciddi zarar verebilir. Bunu engellemek adına maliyeti aşmamaya dikkat etmek gerekmektedir.

## **7. Müşteri ile Anlaşılan kalitede tamamlanmış olması**

Yapılan işin müşteri tarafından beğenilmesi için müşteri tarafından istenilen kalitede geliştirilmiş olması gerekmektedir. Bu bağlamda müşteri dikkatlice dinlenmeli ve istekleri doğru bir şekilde anlaşılmalıdır.

## **8. Teslim edilen ürünün müşteri tarafına başarılı bir şekilde aktarılması**

Ürünü geliştirmek bir tarafa aynı zamanda geliştirilen ürünün kullanıcı tarafına doğru bir şekilde aktarılması gerekmektedir. Aksi takdirde geliştirilen özelliklerin birçoğu kullanıcıların doğru bir şekilde anlamaması nedeniyle hiç kullanılmayacak veya yanlış şekilde kullanılacaktır. Yanlış şekilde kullanılan özellikler müşteri açısından ciddi sonuçlar oluşturabilir (örnek olarak kullanıcılar bir bölüme ait bilgileri güncellemek isterken o bölümdeki tüm doktorların bilgisini silebilirler)

Yukarıda bahsedilen maddeler yazılım geliştirme sürecimizin vazgeçilmez özellikleridir. Bunlardan birisinin eksik olması geliştirilen yazılımın kalitesine ciddi şekilde zarar verecektir. Bu nedenle bu maddelerin iyi anlaşılması ve uygulanması kaliteli bir yazılım süreci geçirmek ve ürünü başarılı bir şekilde müşteri tarafına aktarmak için gereklidir.

## Sonuç

Çalışmamızın konusu olan hastane otomasyon sistemi bir hastanenin ihtiyacı olan sistemleri web üzerinden sağlamak amacıyla geliştirilmiş bir fullstack web yazılımıdır. Geliştirilen uygulamada randevu sistemi, doktor ve bölüm sistemleri, Web ekranları, kullanıcı girişleri gibi birçok modül başarıyla tamamlanmıştır. Tamamlanan bu modüller bir React.js uygulaması ile kullanıcılara sunulmuştur.

Uygulama bir web uygulaması olarak backend tarafında spring boot, frontend tarafında react.js kullanılarak geliştirilmiştir. Bunun yanında backend tarafında mikroservis mimarisi kullanılmış tüm servisler birbirinden bağımsız çalışabilecek şekilde geliştirilmiştir.

Geliştirilen uygulama sayesinde mikroservis mimarisinin çalışma mantığı, servisler arasında güvenliğin ve iletişim nasıl sağlanacağı konusunda örnekler geliştirilmiş ve başarıyla uygulanmıştır.

Hedeflenen proje başarıyla geliştirilmiştir. Çalışma içerisinde backend tarafında servislerin haberleşmesi ve güvenliği sağlanmış, frontend tarafında ise geliştirilen uygulama ile kullanıcılara sunulmuştur. React.js ile geliştirilen web sayfaları aracılığı ile kullanıcı deneyimi başarıyla sağlanmıştır.

## Kaynakça

- [1] <https://spring.io/microservices> spring boot mikroservice
- [2] <https://oril.co/blog/spring-cloud-gateway-security-with-jwt/> microservis uygulamalarında jwt kullanımı
- [3] Youtube Daily Code Buffer  
Mikroservis teknolojileri kullanarak spring boot ile uygulama geliştirme
- [4] Youtube Java Brains  
Mikroservis teknolojileri kullanarak spring boot ile uygulama geliştirme

## ÖZGEÇMİŞ

Mustafa Hasan Kuzulu 30.04.1999'da İstanbul/ Tuzla da doğdu. İlk ve orta öğretimini Tuzla da tamamladı. Tuzla Teknik Anadolu Meslek Lisesi'nde Bilişim teknolojileri bölümünden mezun oldu. Lise eğitimini tamamladıktan sonra 2018 yılında Sakarya Üniversitesi Bilgisayar Mühendisliğine başladı. 2019 yılında 2. Sınıfın başında Toyota proje yarışmasını kazanarak Sakarya Üniversitesindeki Toyota Proje ofisinde çalışmaya başladı. Halen Toyota Türkiye için çalışmaya devam etmektedir.

## BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU : HASTANE OTOMASYON SİSTEMİ  
ÖĞRENCİLER (G181210030/ Mustafa Hasan /Kuzulu):

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
<b>Yazılı Çalışma</b>			
<b>Çalışma klavuza uygun olarak hazırlanmış mı?</b>	x	0-5	
<b>Teknik Yönden</b>			
<b>Problemin tanımı yapılmış mı?</b>	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
<b>Yapılan işlerin zorluk derecesi?</b>	x	0-25	
<b>Sözlü Sınav</b>			
<b>Yapılan sunum başarılı mı?</b>	x	0-5	
<b>Soruları yanıtlama yetkinliği?</b>	x	0-20	
<b>Devam Durumu</b>			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
<b>Diğer Maddeler</b>			
<b>Toplam</b>			

DANIŞMAN: Prof. Dr. Ümit KOCABIÇAK  
DANIŞMAN İMZASI: