



**T.C.
KASTAMONU ÜNİVERSİTESİ
MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BİTİRME PROJESİ I

KONU

Hastane Otomasyonu

DANIŞMAN

Doç. Dr. Kemal AKYOL

Nisan- 2022
KASTAMONU

**T.C.
KASTAMONU ÜNİVERSİTESİ
MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

BİTİRME PROJESİ I

KONU

Hastane Otomasyonu

HAZIRLAYAN

194410006 – Eren YARDIM

DANIŞMAN

Doç. Dr. Kemal AKYOL

Nisan- 2022
KASTAMONU

23/04/ 2022

ETİK BEYAN

Kastamonu Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Mühendislik Tamamlama Programı, Tez Hazırlama Kılavuzu’nda yer alan kurallara uygun olarak hazırladığım bu çalışmada; proje içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirmeye ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, proje çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğim, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu projede sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Öğrenci Numarası : 194410006

İmza

Adı Soyadı : Eren YARDIM

ÖZET

Bitime Projesi

HASTANE OTOMASYONU

Eren YARDIM

**Kastamonu Üniversitesi
Bilgisayar Mühendisliği Bölümü**

Bitirme Projesi Danışmanı:

Doç. Dr. Kemal AKYOL

Nisan 2022, 27 sayfa

Bu bitirme projesinde ilk olarak programlama dilleri hakkında bilgiler verilmiştir. Daha sonra nesneye yönelik programlama dillerinden olan C# dili üzerinde araştırmalar yapılmıştır. C# programlama dili derleyicisi olarak kullanılan Visual Studio 2019 programı incelenmiş ve Visual Studio 2019 ilk defa kullanacaklara yönelik programın arayüzüne aşina olabilmeleri için arayüz bileşenleri verilmiştir. Bunlarla birlikte bitirme projemin konusunu oluşturan otomasyon sistemi için kullanılmak üzere veri tabanları incelenmiştir. Projemizde kullanılan MICROSOFT SQL Server ve MY SQL hakkında bilgiler işlenmiştir. Projem herhangi bir hastane servisinin gereksinim duyacağı, hastanenin ihtiyaçlarına cevap verebilecek bir otomasyon sistemi ve bu sistemle Web Service üzerinden haberleşen Eczane Otomasyonu içermektedir. Yazılım geliştirme süreç aşamaları işlenmiş ve proje görselleriyle birlikte tanıtılmıştır.

Anahtar Sözcükler : C#, MICROSOFT SQL Server, Visual Studio 2019, MVC, Microsoft SQL Server Management Studio, MY SQL, WEB Service

ABSTRACT

Senior Project

HOSPITAL AUTOMATION

Eren YARDIM

Kastamonu University

**Faculty of Engineering and Architecture
Department of Computer Engineering**

Project Advisor:

Assoc. Dr. Kemal AKYOL

April 2022, 27 pages

In this graduation project, firstly, information about programming languages is given. Later, researches were made on C# language, which is one of the object-oriented programming languages. The Visual Studio 2019 program, which is used as a C# programming language compiler, has been examined and the interface components have been given for those who will use Visual Studio 2019 for the first time so that they can be familiar with the interface of the program. Along with these, databases were examined to be used for the automation system, which is the subject of my graduation project. Information about MICROSOFT SQL Server and MY SQL used in our project has been processed. My project includes an automation system that can meet the needs of any hospital service and Pharmacy Automation that communicates with this system via Web Service. The software development process stages were processed and introduced with the project visuals.

Key Words : C#, MS SQL, Visual Studio 2019, MVC, MICROSOFT SQL Server Management Studio, MY SQL, WEB Service

İÇİNDEKİLER	SAYFA
BİTİRME PROJESİ I	i
DANIŞMAN.....	i
BİTİRME PROJESİ I	ii
HAZIRLAYAN	ii
DANIŞMAN.....	ii
ÖZET	iv
ABSTRACT.....	v
Eren YARDIM	v
Kastamonu University	v
Project Advisor:	v
1. GİRİŞ ve TANITIM	8
2. PROJE KONUSU HAKKINDA GENEL BİLGİLER.....	8
3. PROJE ÇALIŞMASINDA KULLANILAN MATERYALLER	9
3.1. C# Programlama Dili.....	9
3.2. MVC.....	10
3.2.1. Model Nedir?	11
3.2.2. View Nedir?	12
3.2.3. Controller Nedir?	12
3.2.4. MVC'nin Yaşam Döngüsü?	12
3.2. Visual Studio 2019.....	13
3.3 MICROSOFT SQL Server.....	14
3.4 MY SQL	15
3.4.1 MySQL Nerede Kullanılır.....	16
3.4.2MySQL WorkBench	16
3.5. Web Service	17
4. Hastane Otomasyonu Proje Görünüm ve Açıklamaları	18
4.1. Hastane Veri Tabanı.....	18
4.2. Anasayfa Ekranı.....	21

4.3. Login Ekranı.....	21
4.4. Randevu Sistemi.....	24
4.5. Hasta Muayene İşlemi	27
4.6. Hasta Muayene Geçmişİ	30
4.7. Doktor İzin Sistemi	31
4.8. Doktorlar Arası Mesajlaşma.....	32
4.9. Personel Ekleme ve Listeleme.....	34
4.10. Doktor İzin Onay	36
4.11. Doktorlar ve Personeller	38
4.12. Personel Ekle	39
4.13. Hastane CRUD işlemleri	41
4.14. Klinik CRUD İşlemleri	45
4.15. İlaç CRUD İşlemleri.....	48
5. Eczane Otomasyonu Görünüm ve Açıklamaları.....	50
5.1. Eczane Veri Tabanı.....	50
5.2. Reçeteli İlaç Ver	53
5.3. Faturalar	56
5.4. Personel CRUD İşlemleri	57
5.6. İlaç CRUD İşlemleri.....	61
6. Projeyi Somee Sunucusuna Yükleme	64
7. SONUÇLAR VE ÖNERİLER.....	68
KAYNAKLAR	69
ÖZGEÇMİŞ.....	70
ADRES BİLGİLERİ.....	70

1. GİRİŞ ve TANITIM

Hastane otomasyon programları, sağlık merkezinize başvuran hastanın detaylı kişisel iletişim ve medikal geçmiş bilgilerini kayıt altına alıp takip etmekle kalmaz aynı zamanda cari hesap oluşturmanızı, ön muhasebe, maliyet yönetimi ve personel yönetimi yapmanızı, kolayca randevu oluşturmanızı, hasta kabul, yatis ve çıkışlarını düzenlemenizi, anlaşmalı kurum takibi yapmanızı, doktor hak edişleri hesaplamanızı ve sağlık merkezinizin sağlıklı şekilde çalışmasını sağlayan daha pek çok işlemi gerçekleştirmenizi sağlar.

Bütünleşik bir yapıda çalışan hastane otomasyon yazılımları, bir birime yapılan veri girişini aynı anda diğer birimlere de yansıtır. Böylece hem bilgilerin tekrar tekrar girilmesine gerek kalmaz hem de hata oranı sıfırlanır. Pratik ve kolay kullanım sunan bu programlar, dileğiniz ayrıntıda raporlar sunarak kapsamlı maliyet ve gelir analizi yapmanıza olanak tanır. Böylece merkezinizin geleceğini ilgilendiren önemli kararları alırken elinizde daha sağlam veriler olur.

Eczane otomasyonu, muayene edilen hastaya verilen reçete kodu ile reçetede bulunan ilaç barkodlarıyla örtüsen ilaçların belirlenmesi ve satılması işlemlerini yürütür. İlaç stok takibi ve Personel CRUD işlemleri gerçekleştirmektedir.

2. PROJE KONUSU HAKKINDA GENEL BİLGİLER

Hastane otomasyonu, genel olarak sistem üzerinden hasta randevu alma işlemi ve doktor muayene işlemi gibi faaliyetler gerçekleştirmeyi hedefler. Sistem üzerinde hastane yönetimi tarafından personel ekleme, silme ve izin kontrolü faaliyetleri gerçekleştirilecek. Doktor muayenesi sonucunda hastaneden hastaya verilmiş olan ilaçların temin edilmesi için Eczane otomasyonu da yapılacak. Hastane ve eczane otomasyonu arasında veri alışverişi Web Service üzerinden gerçekleştirilecek.

3. PROJE ÇALIŞMASINDA KULLANILAN MATERİYALLER

3.1. C# Programlama Dili

C# Programlama Dili Microsoft'un geliştirmiş olduğu yeni nesil dilidir. Yine Microsoft tarafından geliştirilmiş .NET Teknolojisi için geliştirilmiş dillerden biridir.

C programlama dilinde bir tamsayı değişkeni 1 attırmak için ++ soneki kullanılır. C++ dili adını, C diliyle Nesneye Yönelimli Programlama yapabilmek için eklentiler (C With Classes) almıştır. Benzer şekilde C++ diline yeni eklentiler yapılarak ((C++)++) bir adım daha ileriye götürülmüş ve tamamen nesneye yönelik tasarlanmış C# dilinin isimlendirilmesinde, + karakterlerinin birbirlerine yakınlaşmış hali ve bir melodi anahtarı olan C# Major kullanılmıştır.

Bu dilin tasarlanmasına Pascal, Delphi derleyicileri ve J++ programlama dilinin tasarımlarıyla bilinen Anders Hejlsberg liderlik etmiştir. Birçok alanda Java'yı kendisine örnek alır ve C# da java gibi C ve C++ kod sözdizimine benzer bir kod yapısındadır. .NET kütüphanelerini kullanmak amacıyla yazılan programların çalıştığı bilgisayarlarda uyumlu bir kütüphanenin ve yorumlayıcının bulunması gereklidir. Bu, Microsoft'un .Net Framework'u olabileceği gibi ECMA standartlarına uygun herhangi bir kütüphane ve yorumlayıcı da olabilir.

Özellikle nesne yönelimli programlama kavramının gelişmesine katkıda bulunan en aktif programlama dillerinden biridir .NET platformunun anadili olduğu bazı kesimler tarafından kabul görse de bazıları bunun doğru olmadığını savunur. C#, .NET orta seviyeli programlama dillerindendir. Yani hem makine diline hem de insan algısına eşit seviyededir. Buradaki orta ifadesi dilin gücünü değil makine dili ile günlük konuşma diline olan mesafesini göstermektedir. Örneğin; Visual Basic .NET (VB.NET) yüksek seviyeli bir dildir. Dersek bu dilin insanların günlük yaşantlarında konuşma biçimine yakın şekilde yazıldığını ifade etmektedir.

3.1.1. Razor Yapısı

Razor View Engine; View'lar da HTML, CSS, JavaScript ile beraber C#'ın da kullanılabilmesini sağlamaktadır. View'lara yazılan HTML, CSS ve JavaScript kodları istemci tarafında çalışmaktadır. C# kodları ise sunucu tarafında çalışır ve sonuç istemciye gönderilir.

View'de C# kodları yazabilmek için “@{ }” ile Razor kod bloğu oluşturmamız gerekmektedir. Örneğimiz de Razor kod bloğu içerisinde 3 adet değişken tanımladık. Razor kod bloğu içerisinde tanımladığımız değişkenlere, kod bloğu dışında erişebilmek için başlarına “@” işaretini koymamız gerekmektedir.

```
@{  
    string Ad = "Eren";  
    string Soyadı = "Yardım";  
    string Mail = "erenym3@gmail.com";  
}  
<b>Adı:</b>@Ad <br/>  
<b>Soyadı:</b>@Soyadı <br/>  
<b>Mail Adresi:</b>@Mail <br/>s
```

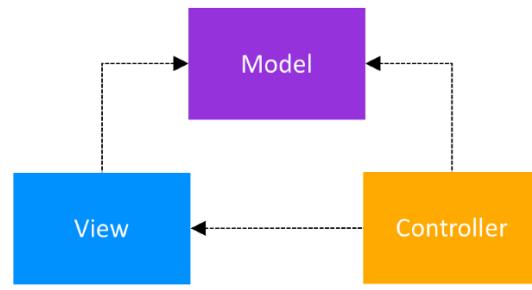
3.2. MVC

MVC, Yazılım Mühendisliği’nde önemli bir yere sahip architectural patterns (yazılım mimari desenleri)’ın bir parçasıdır. Model, View ve Controller kelimelerinin baş harflerinden oluşan MVC (Model-View-Controller), 1979 yılında Tygve Reeskaug tarafından oluşturulmuş ve yazılım gelişmede birçok projede kullanılmıştır. Son

dönemlerde Microsoft'un MVC desenini Asp.Net teknolojisi ile birleştirmesi ile popüleritesi daha da artmıştır.

MVC ile ilgili en yanlış bilgi, MVC'nin Microsoft tarafından çıkartıldığı düşüncesidir. Yukarıdaki paragrafta da bahsedildiği üzere, MVC'nin Asp.Net'e entegre edilmesinden önce bu deseni birçok (.Net) yazılım geliştiricisi bilmemektedir, bilse de kullanmuyorlardı. Asp.Net MVC'nin gelişiminin ardından MVC'ye ilgi oldukça artmıştır.

MVC deseni, 3 katmandan oluşmaktadır ve katmanları birbirinden bağımsız (birbirini etkilemeden) olarak çalışmaktadır. Bu sebeple çoğunlukla büyük çaplı projelerde projelerin yönetiminin ve kontrolünün daha rahat sağlanabilmesi için tercih edilmektedir. MVC ile geliştirilen projelerde projenin detaylarına göre birçok kişi eş zamanlı olarak kolaylıkla çalışabilmektedir.



Şekil 3.1. MVC

3.2.1. Model Nedir?

Model, MVC'de projenin iş mantığının (business logic) oluşturulduğu bölümdür. İş mantığıyla beraber doğrulama (validation) ve veri erişim (data access) işlemleri de bu bölümde gerçekleştirilmektedir.

Model tek katmandan oluşabileceği gibi kendi içinde birden fazla katmandan da oluşabilir. İç yapılandırma projenin büyüklüğü ile yazılım geliştiricinin planlamasına

kalmış bir durumdur. Eğer proje büyük çaplı ise modeli birden çok katmana ayırmak projenin yönetimi açısından faydalı olacaktır.

3.2.2. View Nedir?

View, MVC'de projenin arayüzlerinin oluşturulduğu bölümdür. Bu bölümde projenin kullanıcılarına sunulacak olan HTML dosyaları yer almaktadır. Projenin geliştirildiği yazılım dillerine göre dosya uzantıları da değişebilmektedir. Projelerin büyüklüğüne göre dikkat edilmesi gereken bir nokta ise, klasörlemedir.

Eğer bir web projesi geliştirmiyorsanız, projenin View'larının yer aldığı klasörlerinin hiyerarşisi, ilerleyen dönemlerde karmaşıklığa sebep olmaması için dikkatli yapılmalıdır. Kimi yazılım geliştiriciler web projelerinde HTML dosyaları ile Javascript, CSS ve resim dosyalarını aynı klasör içinde barındırmaktadır. Ufak bir ayrıntı gibi görünse de projenin ilerleyen dönemlerinde ciddi problemler oluşturmaktadır. View'ın bir görevi de kullanıcılarından alınan istekleri controller'a iletmektir.

3.2.3. Controller Nedir?

Controller, MVC'de projenin iç süreçlerini kontrol eden bölümdür. Bu bölümde View ile Model arasındaki bağlantı kurulur. Kullanıcılardan gelen istekler (request) Controller'larda değerlendirilir, isteğin detayına göre hangi işlemlerin yapılabacağı ve kullanıcıya hangi View'ın geleceği (response) belirtilir.

3.2.4. MVC'nin Yaşam Döngüsü?

MVC'nin parçaları olan Model, View ve Controller'in ne olduğu yukarıdaki bölümde anlatıldı. Şimdi bu bilgileri toparlayıp MVC'nin yaşam döngüsünü (çalışma prensibini) detaylıca inceleyelim.



Şekil 3.2. MVC Yaşam Döngüsü

1. HTTP Request: Sizin her ASP.NET MVC uygulamasını görüntülemek istemeniz bir request(istek) tir. Bu istediginizi HTTP üzerinden IIS tarafından alınır. Her

yaptığınız istek Server tarafından bir yanıtla
son bulması gereklidir.

2. Routing: ASP.NET MVC uygulamasını her istek yaptığınızda, yaptığınız yanıt UrlRoutingModule HTTP Module tarafından durdurulur. UrlRoutingModule bir isteği durdurduğu zaman, gelen istek RouteTable'dan hangi Controller tarafından üstleneceğine karar verilir.

3. Controller: RouteTable'dan gelen route bilgisine göre Controller hangi Action'ı çalıştıracağsa o View çalıştırılır. View, Controller tarafından render edilmez. Controller tarafından geriye ViewResult döndürülür.

4. ViewResult: ViewResult, View'i render etmek için aktif View Engine'i çağırır.

5. ViewEngine : Bir CSHTML dosyayı oluşturduğunuzda içerisindeki script ve markuplar, Razor View Engine tarafından bazı ASP.NET API'lerini sayfalarınızı HTML'e çevirmek için kullanır.

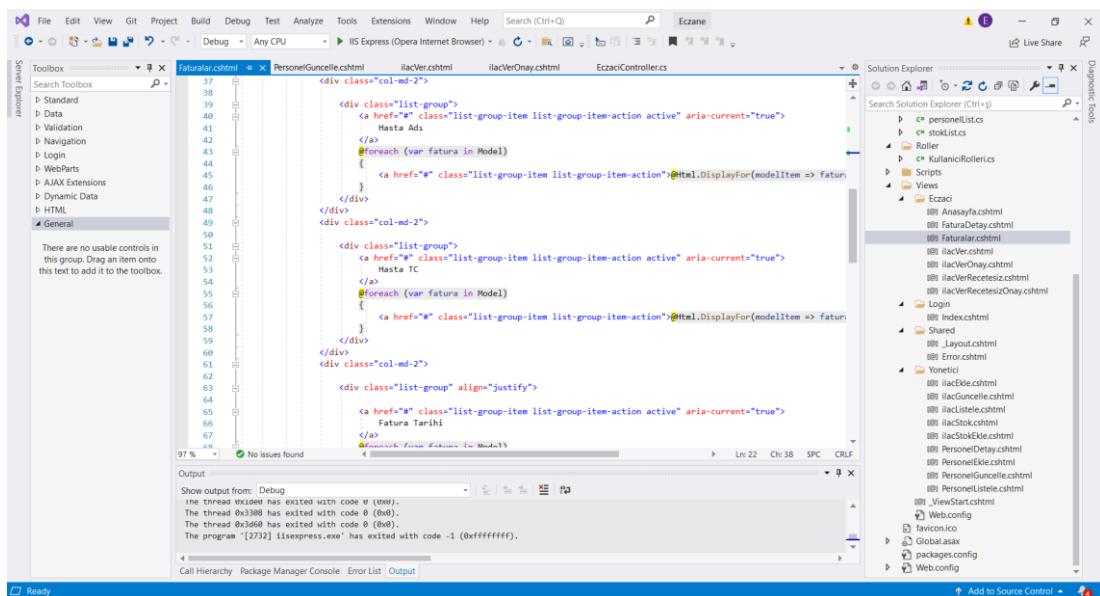
6. View: View Engine tarafından HTML'e çevrilen kodlar kullanıcıya sunulur.

7. Response: HTTP üzerinden View kullanıcıya gösterilir.

3.2. Visual Studio 2019

Microsoft Visual Studio, Windows için IDE yani entegre bir geliştirme ortamıdır. Visual Studio, Microsoft Windows, Windows Mobil, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve web siteleri, web uygulamaları ve web hizmetleriyle ilgili kodlar yazmak için kullanışlı bir programdır.

Bu programı bu kadar özel kılan ve kullanıcılar için tercih sebebi olma nedeniyse kullanıcılarına kod düzenleyici, hata ayıklayıcı, GUI tasarım aracı ve veri tabanı şeması tasarımları seçeneği, revizyon kontrol sistemi gibi çeşitli özellikler sunmasından kaynaklıdır.



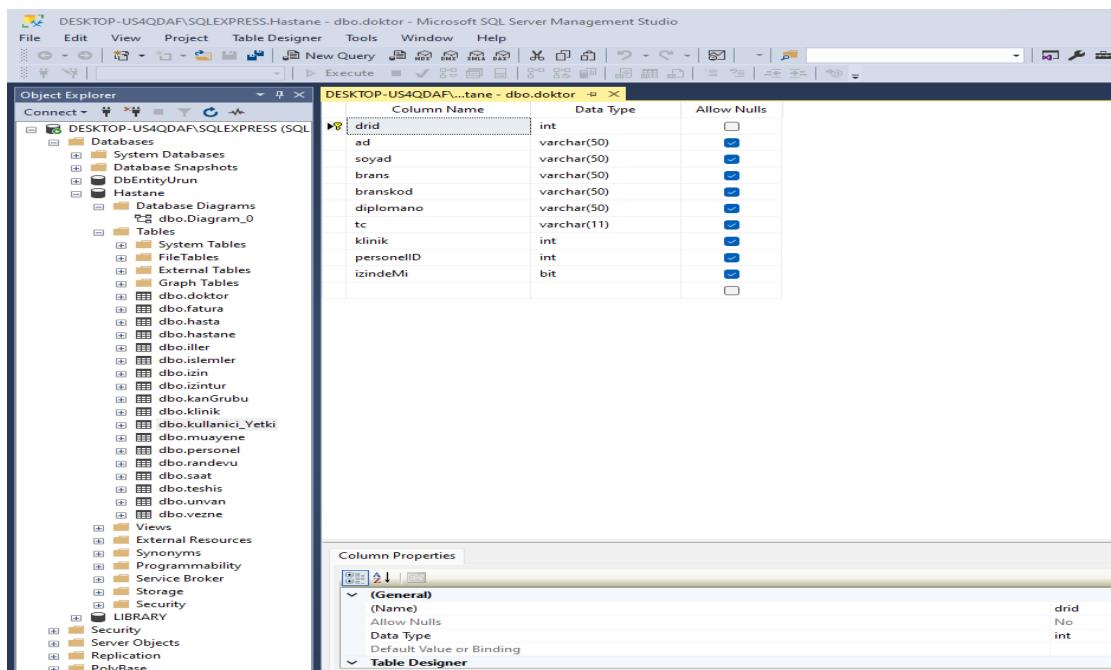
Şekil 3.3. Visual Studio 2019

3.3 MICROSOFT SQL Server

Microsoft SQL Server, verilerin güvenle ve bütünlük içerisinde depolanmasını ve aynı anda birden fazla kullanıcı tarafından erişilmesini sağlayan kurumsal çaplı bir ilişkisel veri tabanı yönetim sistemidir (RDBMS). Birbirile ilişkili verilerin sistematik bir şekilde kaydedilmesini ve bu verilerden beslenen uygulamalar tarafından ihtiyaç anında kullanılmasına olanak sağlayarak işletmelerin BT altyapılarındaki veri aktarımını sorunsuz bir şekilde yönetmelerini sağlar.

SQL Server, işletmeler için kritik önem taşıyan verileri şifreleme, verilere erişim sağlayan kişileri gözleme ve erişim kısıtlamaları tanımlama gibi güvenlik özellikleri sayesinde kullanıcılara kapsamlı bir denetim kapasitesi sunar. Yüksek Kullanılabilirlik (High Availability) ve olağanüstü durum kurtarma özelliklerini tek bir çözümde birleştiren SQL Server, hızlı yük devretme ve yük dengeleme desteği ile aksaklı süresini en aza indirir ve güvenli veri akışının optimize edilmesini sağlar.

Bilişim, sağlık ve finans gibi çeşitli sektörlerde önde gelen şirketler tarafından tercih edilen Microsoft SQL Server, her tip kullanıcının ihtiyaçlarına uyum sağlayabilen ölçülebilin bir yapıya sahiptir.



Şekil 3.4. MS SQL Server Management Studio

3.4 MY SQL

MySQL, modern büyük veri ekosistemindeki en tanınmış teknolojilerden biridir. MySQL ile, ilişkisel sistemlerde yeni olanlar bile hızlı, güçlü ve güvenli veri depolama sistemleri oluşturabilir. MySQL 'in programatik sözdizimi ve ara yüzleri, diğer popüler sorgu dillerinin ve yapılandırılmış veri depolarının geniş dünyasına açılan mükemmel bir ağ geçidiidir.

MySQL, Oracle tarafından geliştirilmiş, yapılandırılmış sorgu diline (SQL) dayalı bir ilişkisel veri tabanı yönetim sistemidir (RDBMS).

Açık kaynak yapısı, istikrarı ve Oracle'nin sürekli geliştirme desteğiyle zengin bir özellik setine sahiptir. Facebook, Flickr, Twitter, Wikipedia ve YouTube gibi internet açısından kritik kuruluşların hepsi MySQL kullanır.

Verileri bir veri tabanında depolamak için kullanabileceğiniz farklı yaklaşımlar vardır. MySQL, ilişkisel veri tabanı adı verilen bir yaklaşımı tercih eder.

İlişkisel bir veri tabanıyla, verileriniz, her şeyi tek bir büyük depolama biriminde bir araya getirmek yerine, tablo adı verilen çok sayıda ayrı depolama alanına böler.

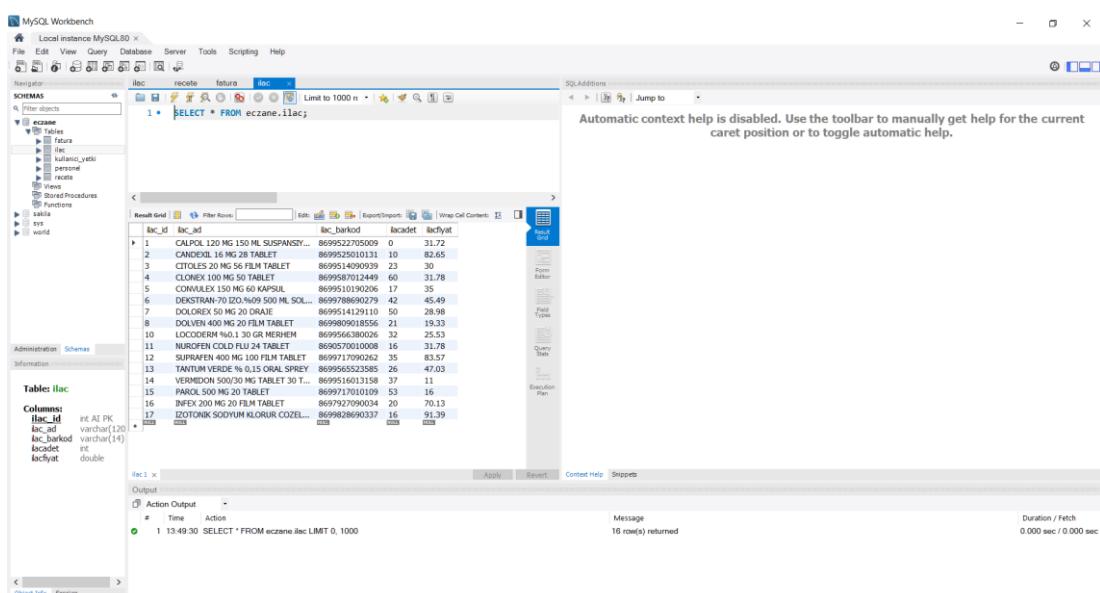
3.4.1 MySQL Nerede Kullanılır

MySQL için en yaygın kullanım, bir web veri tabanı içindir. Tek bir bilgi kaydından çevrimiçi mağaza için mevcut ürünlerin tüm envanterine kadar her şeyi depolamak için kullanılabilir.

PHP veya Perl gibi bir betik diliyle ilişkili olarak, kategorize edilmiş ve aranabilir bilgileri bir web sitesi kullanıcısına hızlı bir şekilde göstermek için bir MySQL veri tabanı ile gerçek zamanlı olarak etkileşime girecek web siteleri oluşturmak mümkündür.

3.4.2 MySQL WorkBench

MySQL Workbench, veri tabanı mimarları, geliştiricileri ve DBA'lar için birleşik bir görsel araçtır. Veri tabanlarını tasarlama, geliştirmek ve yönetmek için çeşitli işlevler sağlar. 3 sürümü mevcuttur: Community, Standard ve Enterprise. Community, temel bir dizi özellik sunan ücretsiz bir açık kaynak sürümüdür, geri kalan ikisi genişletilmiş işlevselliğe sahip ticari sürümlerdir.



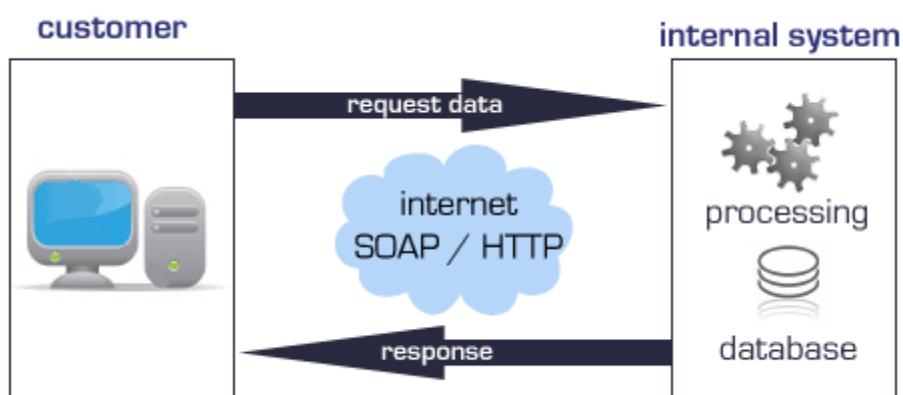
Şekil 3.5. MySQL WorkBench

3.5. Web Service

Genel olarak web servisler SOAP (Simple Object Access Protocol/Basit Nesne Erişim Protokolü) XML yapısını kullanarak çalışmaktadır. Bu yapının bizlere sağlamış olduğu avantaj platformdan bağımsız olarak istenilen yerden erişilmesini sağlamaktır.

Örneğin; .Net ile yazılan bir web servisi Javascript veya JSON ile ve birçok yazılım diliyle kullanabilmekteyiz. Bunun sebebi web servisten istediğimiz sonucun bizlere XML olarak sonuçları getirmesidir.

Web servisler birçok alanda kullanılmaktadır. Örneğin; TCMB döviz kurlarını almak istersek kurumun vermiş olduğu web servisten istediğimiz veriyi alıp işlemlerimizi yapabiliriz. Web servis kullanırken dikkat etmemiz gereken bir husus ise web servis yapısına uygun değerleri göndermemiz olmalıdır.

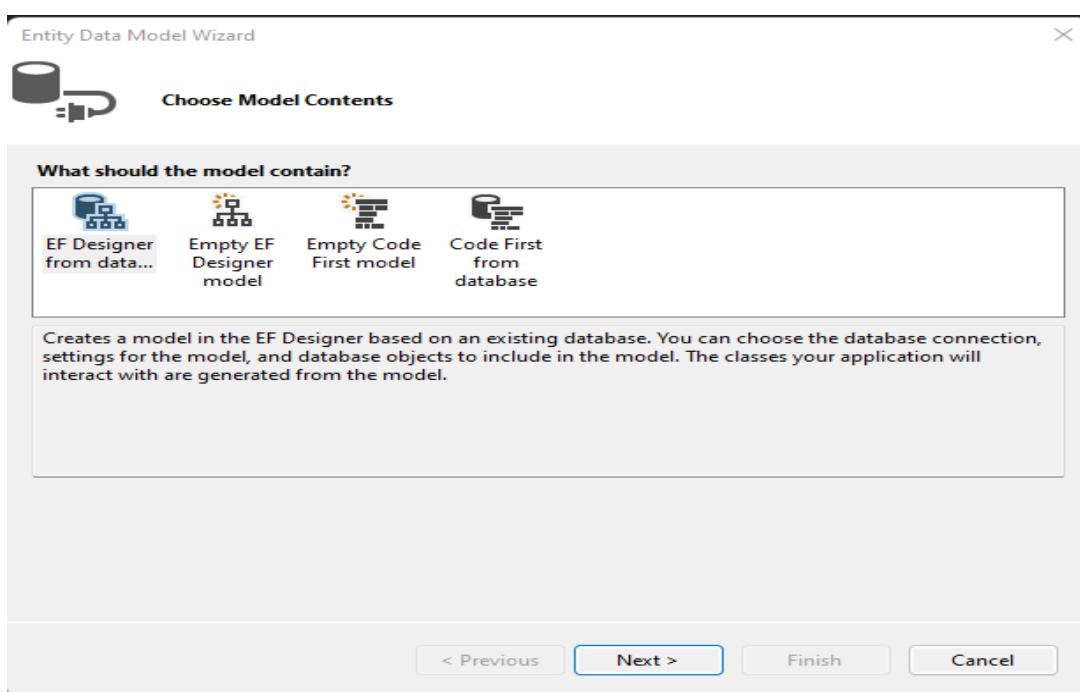


Şekil 3.6. Web Service

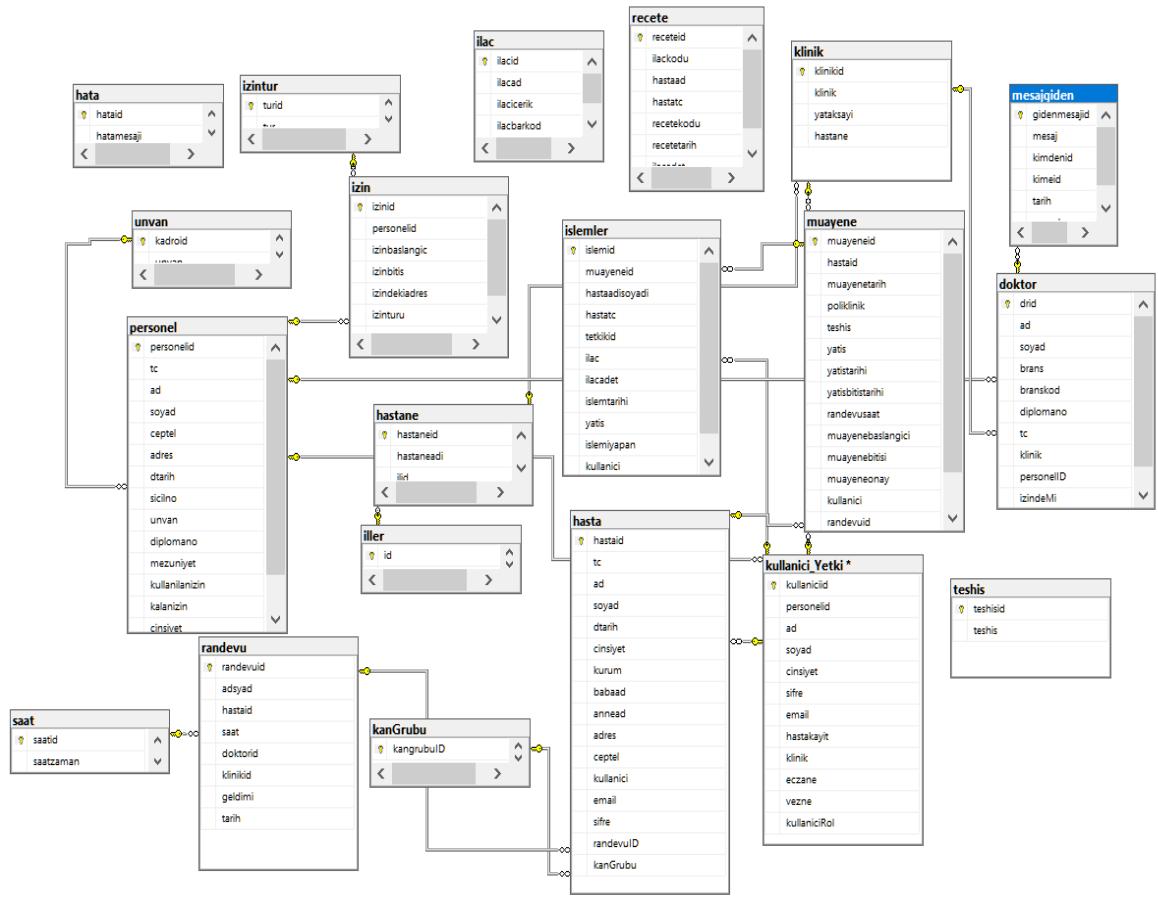
4. Hastane Otomasyonu Proje Görünüm ve Açıklamaları

4.1. Hastane Veri Tabanı

MS SQL'de oluşturduğum veri tabanını projeye ADO.NET Entity Data Model üzerinden EF Designer from database yaklaşımı ile dahil ettim. Modele dahil edilecek veritabanı bağlantısını, model için ayarları ve veritabanı nesnelerini seçtikten sonra uygulamamızın etkileşimde bulunacağı sınıflar modelden oluşturulur.



Şekil 4.1. MS SQL Bağlantısı.



Şekil 4.2. MS SQL Tablolar Arası İlişki

Tablolar arası ilişkiler:

<u>Bağlantılı Tablolar</u>	<u>Primary Key</u>	<u>Foreing Key</u>
1. randevu-saat	saat(saatid)	randevu(saat)
2. unvan-personel	unvan(kadroid)	personel(kadroid)
3. izintur-izin	izintur(turid)	izin(izinturu)
4. izin-personel	personel(personelid)	izin(personelid)
5. iller-hastane	iller(id)	hastane(ilid)
6. hastane-klinik	hastane(hastaneid)	klinik(hastane)
7. klinik-doktor	klinik(klinikid)	doktor(klinik)
8. klinik-muayene	klinik(klinikid)	muayene(poliklinik)
9. doktor-mesajgiden	doktor(drid)	mesajgiden(kimeid)
10. randevu-hasta	randevu(randevuid)	hasta(randevuID)
11. kanGrubu-hasta	kanGrubu(kangrubuID)	hasta(kanGrubu)
12. personel-doktor	personel(personelid)	doktor(personelID)
13. personel-kullanici_Yetki	personel(personelid)	kullanici_Yetki(personelid)
14. kullanici_Yetki-hasta	kullanici_Yetki(kullaniciid)	hasta(kullanici)
15. kullanici_Yetki-islemeler	kullanici_Yetki(kullaniciid)	islemeler(kullanici)
16. kullanici_Yetki-muayene	kullanici_Yetki(kullaniciid)	muayene(kullanici)
17. hasta-muayene	hasta(hastaid)	muayene(hastaid)
18. muayene-islemeler	muayene(muayeneid)	islemeler(muayeneid)

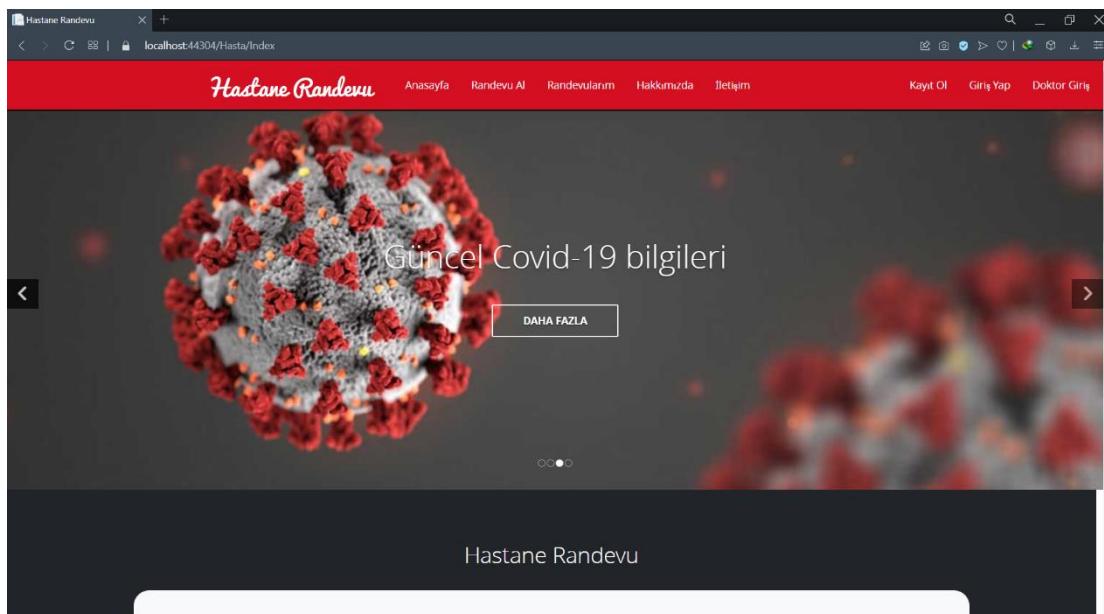
Stored Procedure’ün normal Query kodundan farklı query kodları çalıştırıldığında hata kontrolü yapılır ancak Stored Procedurer’de buna gerek yoktur. Bu nedenle Stored Procedure’ün çalıştırılması daha hızlıdır.

```
USE [Hastane]
GO
***** Object: StoredProcedure [dbo].[sp_SelectDoktor]   Script Date: 9.06.2022 10:41:41 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER proc [dbo].[sp_SelectDoktor]
As
Begin
Select drid , ad,soyad,tc,brans from doktor
End
```

Şekil 4.3. Stored Procedure

4.2. Anasayfa Ekranı

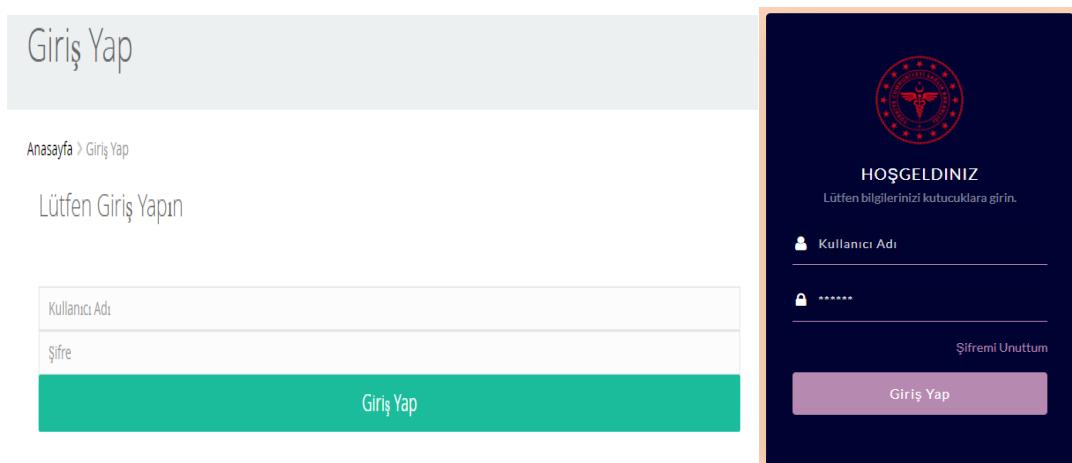
Anasayfa ekranında Bootstrap elemanlarından Carousel kullandım.



Şekil 4.5. Anasayfa Sayfası

4.3. Login Ekranı

Projemde hasta ve doktor olmak üzere iki farklı login ekranı bulunmaktadır. Site üzerinde giriş yapılmadığı taktirde kullanıcının erişmesini istemediğimiz sekmeleri Authorize ve RoleProvider Class yapısı sayesinde engelliyyorum.



Şekil 4.1. Hasta Giriş Ekranı

Şekil 4.2. Doktor Giriş Ekranı

```
330 [HttpGet]
331 public ActionResult HastaGiris()
332 {
333     return View();
334 }
335 [HttpPost]
336 public ActionResult HastaGiris(hasta P)
337 {
338
339     var hastagiris = hs.hasta.Where(x => x.email == P.email && x.sifre == P.sifre).FirstOrDefault();
340     if (hastagiris != null)
341     {
342         FormsAuthentication.SetAuthCookie(hastagiris.email, false);
343         Session["HastaEmail"] = hastagiris.email;
344         Session["HastaID"] = hastagiris.hastaid;
345         Session["HastaGirdi"] = "evet";
346         Session["HastaAdi"] = hastagiris.ad.ToString() + " " + hastagiris.soyad.ToString();
347         return RedirectToAction("Index");
348     }
349     else
350     {
351         ViewBag.hata = "Hatalı kullanıcı girişi";
352         return View();
353     }
354 }
355 }
```

Şekil 4.6. Hasta Giriş Kod

```
22 public ActionResult Index(kullanici_Yetki p)
23 {
24     kullanici_Yetki kullaniciGiris = hs.kullanici_Yetki.Where(x => x.email == p.email && x.sifre == p.sifre).FirstOrDefault();
25
26     if (kullaniciGiris != null)
27     {
28         doktorGiris = hs.doktor.Where(x => x.personelID == kullaniciGiris.personelid).FirstOrDefault();
29         ViewModels vm = new ViewModels();
30         FormsAuthentication.SetAuthCookie(kullaniciGiris.email, false);
31         Session["PersonelEmail"] = kullaniciGiris.email.ToString();
32         Session["PersonelID"] = kullaniciGiris.personelid.ToString();
33         Session["KlinikID"] = kullaniciGiris.klinik;
34         Session["PersonelCinsiyet"] = kullaniciGiris.cinsiyet.ToString().Trim();
35         string tut = Session["PersonelCinsiyet"].ToString();
36         if (doktorGiris != null)
37         {
38             Session["DoktorID"] = Convert.ToInt32(doktorGiris.drid);
39         }
40
41         Session["PersonelAd"] = kullaniciGiris.ad.ToString() + " " + kullaniciGiris.soyad.ToString();
42
43
44
45         return RedirectToAction("Index", "Doktor");
46     }
47     else
48     {
49         ViewBag.yanlisgiris = "Yanlış kullanıcı girişi!";
50         return View();
51     }
52 }
```

Şekil 4.7. Doktor Giriş Kod

```
40 public override string[] GetRolesForUser(string username)
41 {
42     HastaneEntities2 hs = new HastaneEntities2();
43
44     var kullanici = hs.kullanici_Yetki.FirstOrDefault(x => x.email == username);
45     return new string[] { kullanici.kullaniciRol };
46 }
47 }
```

Şekil 4.8. Role Provider Class Yapısı

```

@if (User.IsInRole("D"))
{
    <li class="nav-item">
        <p class="nav-link">
            <i class="nav-icon fas fa-user-injured"></i>
            @Html.ActionLink("Randevu Alan Hastalar", "HastaRandevuListele", "Doktor")
        </p>
    </li>
    <li class="nav-item">
        <p class="nav-link">
            <i class="nav-icon fas fa-users"></i>
            @Html.ActionLink("Muayene Geçmiş", "Hastalarim", "Doktor")
        </p>
    </li>
}

```

Şekil 4.9. Role Provider Rol Kontrol

```

31   <div class="col-md-6 col-lg-4">
32     <div class="login-wrap py-5">
33       <div class="img d-flex align-items-center justify-content-center" style="background-image: url(/AdminLTE-3.0.4/dist/img/login.jpg)">
34         <h3 class="text-center mb-0">Hosgeldiniz</h3>
35         <p class="text-center">Lütfen bilgilerinizi kutucuklara girin.</p>
36         @using (Html.BeginForm("Index", "Login", FormMethod.Post))
37         {
38             <form action="#" class="login-form">
39                 <div class="form-group">
40                   <div class="icon d-flex align-items-center justify-content-center"><span class="fa fa-user"></span>
41                   <input name="email" type="text" class="form-control" placeholder="Kullanici Adı" required>
42               </div>
43               <div class="form-group">
44                 <div class="icon d-flex align-items-center justify-content-center"><span class="fa fa-lock"></span>
45                 <input name="sifre" type="password" class="form-control" placeholder="*****" required>
46             </div>
47             <div class="form-group d-md-flex">
48                 <div class="w-50 text-md-left">
49                     <a href="#" style="color:red">@ViewBag.yanlisgiris</a>
50                 </div>
51                 <div class="w-100 text-md-right">
52                     <a href="#">Sifremi Unuttum</a>
53                 </div>
54             </div>
55             <div class="form-group">
56                 <button type="submit" class="btn form-control btn-primary rounded submit px-3">Giriş Yap</button>
57             </div>
58         </form>
59     }
59   <div class="form-group">
60     <small>Copyright © 2020. Tüm Hakkı Saklıdır. All Rights Reserved</small>
61   </div>

```

Şekil 4.10. Login View Tasarım Kodu

4.4. Randevu Sistemi

Randevu sisteminde hasta seçtiği ildeki randevu almak istediği hastanenin ilgili polikliniğinde bulunan doktorlardan belirlediği tarihte ve saatte randevuamasına imkân tanıyor. Hasta aynı doktordan aynı gün randevu alma durumu engellenmiştir. Hasta randevularım sekmesinde geçmiş tarihli randevularını görebilir, gelecek tarihteki randevularından iptal etmek istediği randevuyu iptal edebilir.

Anasayfa > Randevu Al > /Randevu Onay

Şehir İsmi
SAMSUN

Hastane İsmi
Samsun Anadolu Hastanesi

Klinik İsmi
Genel Cerrahi

Doktor İsmi
Hakan Durmuş

Tarih
2022-06-02

Randevu Saati
13:00

Randevu Saatini Onayla

Şekil 4.11. Randevu Al

```
10<script>
11    $(function () {
12        $('#Dpil').change(function () {
13            var id = $('#Dpil').val();
14            $.ajax({
15                url: '/Hasta/hastanegetir',
16                data: { p: id },
17                type: "POST",
18                dataType: "Json",
19                success: function (data) {
20                    console.log(data);
21                    $('#Phastane').empty();
22                    for (var i = 0; i < data.length; i++) {
23                        $('#Phastane').append("<option value='" + data[i].Value + "'>" + data[i].Text + "</Option>");
24                    }
25                }
26            });
27        });
28        $('#Phastane').change(function () {
29            var id = $('#Phastane').val();
30            $.ajax({
31                url: '/Hasta/klinikgetir',
32                data: { p: id },
33                type: "POST",
34                dataType: "Json",
35                success: function (data) {
36                    console.log(data);
37                    $('#klinikid').empty();
38                    for (var i = 0; i < data.length; i++) {
39                        $('#klinikid').append("<option value='" + data[i].Value + "'>" + data[i].Text + "</Option>");
40                    }
41                }
42            });
43        $('#klinikid').change(function () {
44            var id = $('#klinikid').val();
```

Şekil 4.12. Randevu Al Script Komutları

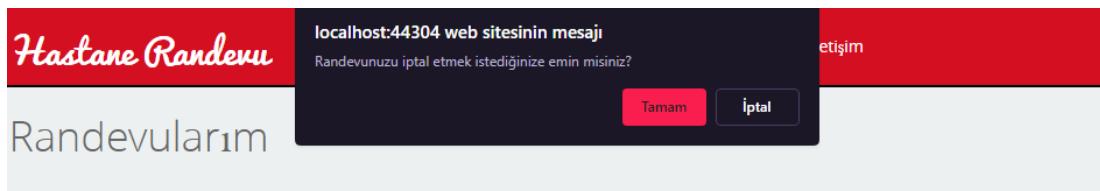
```
[HttpGet]
0 references
public ActionResult RandevuOnay(int? DPil, int? DPhastane, int? klinikid, DateTime tarih, int doktorid)
{
    var randevu = hs.randevu.Where(x => x.tarih == tarih && x.doktorid == doktorid).ToList();
    var saatListe = hs.saat.ToList();
    TempData["iID"] = DPil;
    TempData["hastaneID"] = DPhastane;
    var klinik = hs.klinik.Find(klinikid);
    TempData["klinikID"] = klinikid;
    TempData["tarih"] = tarih;
    TempData["drid"] = doktorid;
    List<SelectListItem> saatListesi = new List<SelectListItem>();

    int[] saatDizi = new int[saatListe.Count()];
    int[] randevuDizi = new int[randevu.Count()];
    int sayac = 0;
    foreach (var rnd in randevu)
    {
        int saatid = Convert.ToInt32(rnd.saat);
        var saat = hs.saat.Find(saatid);
        saatListe.Remove(saat);
    }
    foreach (var saat in saatListe)
    {
        saatListesi.Add(new SelectListItem { Text = saat.saatzaman, Value = saat.saatid.ToString() });
    }

    ViewBag.saatlistesi = saatListesi;
    return View();
}
[HttpPost]
^ ^

[HttpPost]
0 references
public ActionResult RandevuOnay(int? saat,int? klinikid,DateTime tarihDate,int? doktorid)
{
    int hastaid = Convert.ToInt32(Session["HastaID"]);
    var hasta = hs.hasta.Find(hastaid);
    randevu r = new randevu()
    {
        hastaid = hastaid,
        adsyad = Session["HastaAdi"].ToString(),
        saat = saat,
        doktorid = doktorid,
        tarih = tarihDate,
        klinikid = klinikid,
        geldimi = false
    };
    if (hs.randevu.Where(x => x.doktorid == doktorid && x.tarih == tarihDate && x.saat == saat).FirstOrDefault() == null)
    {
        hs.randevu.Add(r);
        hs.SaveChanges();
        var rnd = hs.randevu.Where(x => x.hastaid == hastaid && x.tarih == tarihDate).FirstOrDefault();
        hasta.randevuID = rnd.randevuid;
        hs.SaveChanges();
        TempData["kontrol"] = "true";
    }
    else
    {
        TempData["kontrol"] = "false";
    }
    return RedirectToAction("Index", "Hasta");
}
^ ^
```

Şekil 4.13. Randevu Al Kodları



Anasayfa > /Randevularım

Ad	Tarih	Saat	Randevu Detay	Randevu İptal
Efe Taş	9.04.2022 00:00:00	10:00	Detay	Geçmiş randevu
Efe Taş	9.04.2022 00:00:00	11:00	Detay	Geçmiş randevu
Efe Taş	26.04.2022 00:00:00	13:00	Detay	İptal et
Efe Taş	30.04.2022 00:00:00	11:00	Detay	İptal et

Şekil 4.14. Randevularım ve Randevu İptal

```

146 [HttpGet]
147 public ActionResult Randevularim()
148 {
149     int hastaid = int.Parse(Session["HastaID"].ToString());
150     randevuList rl = new randevuList();
151     rl.randevuListesi = hs.randevu.Where(x => x.hastaid == hastaid).ToList();
152     if (rl == null)
153     {
154         ViewBag.randevuYok = "Henüz randevunuz bulunmamaktadır.";
155     }
156     return View(rl);
157 }
158 [HttpGet]
159 public ActionResult RandevularimIptal(int? randevuID)
160 {
161     var rnd = hs.randevu.Find(randevuID);
162     if (rnd != null)
163     {
164         hs.randevu.Remove(rnd);
165         hs.SaveChanges();
166     }
167     return RedirectToAction("Randevularim", "Hasta");
168 }
169 [HttpGet]
170 public ActionResult RandevularimDetay(int? randevuID)
171 {
172     randevuList rl = new randevuList();
173     var randevum = hs.randevu.Find(randevuID);
174     int id = Convert.ToInt32(randevum.doktorid);
175     rl.randevuListesi = hs.randevu.Where(x => x.randevuid == randevuID).ToList();
176     rl.doktorListesi = hs.doktor.Where(x => x.drid == id).ToList();
177     return View(rl);
178 }

```

Şekil 3.15. Randevularım ve Randevu İptal Kodları

4.5. Hasta Muayene İşlemi

Öncelik olarak randevu alan hastalar

```
rl.randevuListesi = hs.randevu.Where(x => x.doktorid == drID &&
x.tarih >= DateTime.Today && x.tarih <= DateTime.Today).OrderBy(x => x.saat1.saatzman).ToList();
```

Kodu ile sadece o gün ki hasta randevuları listeleniyor.

Efe Taş	5.06.2022	09:00	Detay	Hasta Muayene
---------	-----------	-------	-------	---------------

Hasta muayene butonuna tıklandığında muayene işlemi başlamış oluyor.

Öncelik olarak hastaya kaç adet ilaç verileceği ve yatış işlemi olup olmayacağı belirleniyor. Ardından verilecek ilaçlar dropdownlist de seçiliğin karşısındaki textboxa ilaç adeti giriliyor. Yatış verilecekse yatış başlangıç ve bitiş tarihleri seçiliyor. İşin Controller tarafında:

```
recetekodu = hasta.ad.Substring(0, 2) + "" + hasta.tc.Substring(0, 3) + "" + rnd.randevuid + "" + dr.drId,
```

Kodu ile hastaya özel reçete kodu oluşturuluyor. Bu kod ve hastanın gerekli bilgileri Web Service üzerindeki “GetRecete” fonksiyonu ile MySQL deki eczane veri tabanına aktarılıyor.

Şekil 4.16. Hasta Muayene Ekranı

```

150     [HttpGet]
151     public ActionResult Hastamuayene(int? hastaID, int? randevuID, int? kacAdet,int? yatisonay)
152     {
153         randevuList rl = new randevuList();
154         rl.randevuListesi = hs.randevu.ToList();
155         rl.ilac = new SelectList(hs.ilac, "ilacid", "ilacad");
156         var hasta = hs.hasta.Find(hastaID);
157         var randevu = hs.randevu.Find(randevuID);
158         ViewBag.hastid = hastaID;
159         ViewBag.randevuid = randevuID;
160         ViewBag.kacadet = kacAdet;
161         ViewBag.yatisonay = yatisonay;
162         TempData["Hastaid"] = hastaID;
163         TempData["Randevuid"] = randevuID;
164
165         List<SelectListItem> unvan = (from x in hs.unvan.ToList()
166                                         select new SelectListItem
167                                         {
168                                             Text = x.unvan1,
169                                             Value = x.kadroid.ToString()
170                                         }).ToList();
171         ViewBag.unvan = unvan;
172
173         List<SelectListItem> klinik = (from x in hs.klinik.ToList()
174                                         select new SelectListItem
175                                         {
176                                             Text = x.klinik1,
177                                             Value = x.klinikid.ToString()
178                                         }).ToList();
179         ViewBag.klinik = klinik;
180
181
182         rl.hastaListesi = hs.hasta.Where(x => x.hastaid == hastaID).ToList();
183         return View(rl);

```

```

185     [HttpPost]
186     public ActionResult Hastamuayene(string teshis, string yatisonay, FormCollection frm, string ilac1, string ilac2, string ilac3, string ilac4, string ilac5, int? ilacadet1, int? ilacadet2, int? ilacadet3)
187     {
188         randevu rnd = hs.randevu.Find(Convert.ToInt32(TempData["Randevuid"]));
189         rnd.geldimi = true;
190         hs.SaveChanges();
191         int doktorid = Convert.ToInt32(Session["DoktorID"]);
192         int personelid = Convert.ToInt32(Session["PersonelID"]);
193         var kullanici = hs.kullanici_Vetkl.Where(x => x.personelid == personelid).FirstOrDefault();
194         var dr = hs.doktor.Find(doktorid);
195         var klinik = hs.klinik.Where(x => x.klinikid == dr.klinik).FirstOrDefault();
196         string ilacism = "";
197         string ilackodu = "";
198         string ilacadet = "";
199         if (ilaci != null)
200         {
201             var ilacad = hs.ilac.Find(Convert.ToInt32(ilaci));
202             ilacism += " " + ilacad.ilacad + " " + ilacadet1 + "-Adet";
203             ilackodu += ilacad.ilacberkod;
204             ilacadet += ilacadet1;
205         }
206         if (ilac2 != null)
207         {
208             var ilacad = hs.ilac.Find(Convert.ToInt32(ilac2));
209             ilacism += " " + ilacad.ilacad + " " + ilacadet2 + "-Adet";
210             ilackodu += "/" + ilacad.ilacberkod;
211             ilacadet += "/" + ilacadet2;
212         }
213         if (ilac3 != null)
214         {
215             var ilacad = hs.ilac.Find(Convert.ToInt32(ilac3));
216             ilacism += " " + ilacad.ilacad + " " + ilacadet3 + "-Adet";
217             ilackodu += "/" + ilacad.ilacberkod;
218             ilacadet += "/" + ilacadet3;

```

```

220     if (ilac4 != null)
221     {
222         var ilacad = hs.ilac.Find(Convert.ToInt32(ilac4));
223         ilacisim += " " + ilacad.ilacad + " " + ilacadet4 + "-Adet";
224         ilackodu += "/" + ilacad.ilacbarkod;
225         ilacadet += "/" + ilacadet4;
226     }
227     if (ilac5 != null)
228     {
229         var ilacad = hs.ilac.Find(Convert.ToInt32(ilac5));
230         ilacisim += " " + ilacad.ilacad + " " + ilacadet5 + "-Adet";
231         ilackodu += "/" + ilacad.ilacbarkod;
232         ilacadet += "/" + ilacadet5;
233     }
234     else if (ilac1 == null & ilac2 == null & ilac3 == null & ilac4 == null & ilac5 == null)
235     {
236         ilacisim = "İlaç verilmedi";
237     }
238     ilacisim = ilacisim.Trim();
239     int hastaID = Convert.ToInt32(TempData["Hastaid"]);
240     var hasta = hs.hasta.Find(hastaID);
241     bool yatiskarar;
242     bool onay=true;
243     SqlDbType yatisonaytarih = SqlDbType.Null;
244     SqlDbType yatisbitis = SqlDbType.Null;
245     if (yatisonay == "1")
246     {
247         yatiskarar = true;
248         if (yatisson > DateTime.Now && yatisbaslangic >= DateTime.Today)
249         {
250             onay = false;
251             yatisonaytarih = DateTime.Now;
252             yatisbitis = yatisson;
253         }
254     }
255     else
256     {
257         Session["hatalitarih"] = "true";
258         return RedirectToAction("HastaRandevuListele", "Doktor");
259     }
260     klinik.yataksayi -= 1;
261 }
262 else
263 {
264     yatiskarar = false;
265 }
266 DateTime tarih = (DateTime)rnd.tarih;
267 muayene m = new muayene()
268 {
269     muayenebaslangici = rnd.tarih,
270     muayenebitis = DateTime.Now,
271     muayenetarihi = tarih,
272     teshis = teshis,
273     randevusaat = rnd.saati.saatzaman,
274     hastaid = hastaid|,
275     poliklinik = dr.klinik,
276     yatis = yatiskarar,
277     muayeneonay = true,
278     kullanici = dr.personelID,
279     yatistaribi = (DateTime)yatisonaytarih,
280     yatisbitistaribi = (DateTime)yatisbitis,
281     randevuid = rnd.randevuid,
282     cikisonay=onay
283 };
284 hs.muayene.Add(m);
285 hs.SaveChanges();
286 var muayene = hs.muayene.Where(x => x.hastaid == hastaID && x.randevuid == rnd.randevuid).FirstOrDefault();

```

```

286     islemler islem = new islemler()
287     {
288         muayeneid = m.muayeneid,
289         islemtarihi = DateTime.Now,
290         islemiyapan = dr.drid,
291         kullanici = kullanici.kullaniciid,
292         yatis = yatiskarar,
293         ilac = ilacismis,
294         hastaadisoyadi = hasta.ad + " " + hasta.soyad,
295         hastatc = hasta.tc
296     };
297     rnd.geldimi = false;
298     hs.islemler.Add(islem);
299     hs.SaveChanges();
300     if (ilackodu != "")
301     {
302         recete rct = new recete()
303         {
304             hastaad = hasta.ad + " " + hasta.soyad,
305             hastatc = hasta.tc,
306             ilackodu = ilackodu,
307             recetekodu = hasta.ad.Substring(0, 2) + "" + hasta.tc.Substring(0, 3) + "" + rnd.randevuid + "" + dr.drid,
308             recetetarih = DateTime.Now,
309             ilacadet = ilacadet
310         };
311         hs.recete.Add(rct);
312         hs.SaveChanges();
313     }
314
315     return RedirectToAction("HastaRandevuListele", "Doktor");
316 }

```

Şekil 4.17. Hasta Muayene Kod

4.6. Hasta Muayene Geçmişi

```

[HttpGet]
0 references
public ActionResult Hastalarim(string hastaad, string hastatc)
{
    int personelid = Convert.ToInt32(Session["PersonelID"]);

    var degerler = from d in hs.muayene where d.kullanici == personelid select d;
    if (!string.IsNullOrEmpty(hastaad))
    {
        degerler = degerler.Where(x => x.hasta.ad.Contains(hastaad));
    }
    if (!string.IsNullOrEmpty(hastatc))
    {
        degerler = degerler.Where(x => x.hasta.tc.Contains(hastatc));
    }
    var dgr = degerler.OrderBy(x => x.muayenetarih).ToList();
    return View(dgr.ToList());
}

```

Sisteme giriş yapmış olan doktorun id si Session["PersonelID"] de tutuluyor. Bu id ye göre geçmiş hasta muayeneleri listeleniyor. Sayfada hastanın adı ve TC kimlik numarasına göre arama işlemi gerçekleştirilebiliyor.

```
var dgr = degerler.OrderBy(x => x.muayenetarih).ToList();
```

Yukarıda belirtilen kod sayesinde muayene listesi muayene tarihine göre sıralanıyor.

The screenshot shows a search interface for medical history. At the top, there are two input fields: 'Aranacak Hasta Adı:' and 'TC ile ara:', each with an 'Ara' button. Below these are two rows of search criteria: 'Hasta Adı', 'Hasta TC', 'Muayene Tarihi', 'Teşhis', and 'Muayene Detay'. The main content is a table with 10 rows of examination data:

Hasta Adı	Hasta TC	Muayene Tarihi	Teşhis	Muayene Detay
Efe Taş	46842133322	3.05.2022	Baş ağrısı	Detay
Efe Taş	46842133322	5.05.2022	Baş ağrısı	Detay
Efe Taş	46842133322	12.05.2022	Mide Bulantısı	Detay
Efe Taş	46842133322	19.05.2022	Mide Bulantısı	Detay
Bekir Gündoğdu	46972487654	19.05.2022	Baş Ağrısı	Detay
Bekir Gündoğdu	46972487654	23.05.2022	Mide Bulantısı	Detay
Efe Taş	46842133322	25.05.2022	Mide Bulantısı	Detay
Bülent Yaşar	46974658798	27.05.2022	Mide Bulantısı	Detay
Bekir Gündoğdu	46972487654	29.05.2022	Baş Ağrısı	Detay
Efe Taş	46842133322	29.05.2022	Mide Bulantısı	Detay

Şekil 4.18. Muayene Geçmişİ Sayfası

4.7. Doktor İzin Sistemi

Doktor sistem üzerinden belirli tarih aralıklarında izin talebi oluşturabilir. Oluşturulan izin talebi yönetici tarafından onaylanması gereklidir.

The screenshot shows a user profile 'Hakan Durmuş' on the left and a 'Anasayfa/Izin Al' (Home Page/Apply for Leave) page on the right. The right page has a sidebar with links: Anasayfa, Randevu Alan Hastalar, Muayene Geçmişİ, İzin Al, İletişim ve mesajlar, and Çıkış. The main form fields are: 'Izindeki Adresiniz' (Address), 'Izin Nedeni' (Reason for Leave), 'Izin Başlangıç' (Start Date), and 'Izin Bitiş' (End Date). A large blue 'İzin Talebi Oluştur' (Create Leave Application) button is at the bottom.

Şekil 4.19. İzin Talebi Sayfası

```
117 [HttpGet]
118 public ActionResult Doktorizin()
119 {
120     List<SelectListItem> izintur = (from x in hs.izintur.ToList()
121                                     select new SelectListItem
122                                     {
123                                         Text = x.tur,
124                                         Value = x.turid.ToString()
125                                     }).ToList();
126     ViewBag.izin = izintur;
127     return View();
128 }
129 [HttpPost]
130 public ActionResult Doktorizin(string izindekiadres, int izinturu, DateTime izinbaslangic, DateTime izinbitis)
131 {
132     HastaneEntities2 hs = new HastaneEntities2();
133     int persid = Convert.ToInt32(Session["PersonelID"]);
134     List<int> sayilar = new List<int> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
135     ViewBag.sayi = sayilar;
136     izin iz = new izin
137     {
138         izinturu = izinturu,
139         izinbaslangic = izinbaslangic,
140         izinbitis = izinbitis,
141         personefid = Convert.ToInt32(Session["PersonelID"]),
142         izindekiadres = izindekiadres,
143         onaylandimi = false
144     };
145     hs.izin.Add(iz);
146     hs.SaveChanges();
147     return RedirectToAction("Index", "Doktor");
148 }
149 }
```

Şekil 4.20. İzin Talebi Kodları

4.8. Doktorlar Arası Mesajlaşma

Doktor, kayıtlı olduğu klinikte bulunan diğer doktorlar ile mesajlaşabiliyor.



Şekil 4.21. Mesajlaşma Ekranı

```
[HttpGet]
0 references
public ActionResult mesaj(int? kimeid)
{
    doktorList dl = new doktorList();

    int klinikid = Convert.ToInt32(Session["klinikID"]);
    int drid = Convert.ToInt32(Session["DoktorID"]);
    dl.doktorListesi = hs.doktor.Where(x => x.klinik == klinikid && x.drid != drid).ToList();
    if (kimeid == null)
    {
        var doktor = hs.doktor.Where(x => x.klinik == klinikid && x.drid != drid).FirstOrDefault();
        kimeid = doktor.drid;
    }

    dl.mesajgidenListesi = hs.mesajgiden.Where(x => x.kimdenid == drid && x.kimeid == kimeid).ToList();
    return View(dl);
}

[HttpPost]
0 references
public ActionResult MesajGonder(string mesaj, int? kimeid)
{
    int drid = Convert.ToInt32(Session["DoktorID"]);
    int sayac = 1;
    var ilkMesajMi = hs.mesajgiden.Where(x => x.kimdenid == drid && x.kimeid == kimeid).ToList();
    if (ilkMesajMi.Count == 0)
    {
        mesajgiden gidenmesaj = new mesajgiden()
        {
            kimdenid = kimeid,
            kimeid = drid,
            mesaj = ".",
            cevaplananmesaj = 0
        };
        sayac -= 1;
        hs.mesajgiden.Add(gidenmesaj);
        hs.SaveChanges();
    }
    var mesajlar = hs.mesajgiden.Where(x => x.kimeid == drid && x.kimdenid == kimeid).ToList();
    int cevaplananmesajid = 0;
    foreach (var msj in mesajlar)
    {
        cevaplananmesajid = msj.gidenmesajid;
    }
    if (sayac == 0)
    {
        var ilkmesaj = hs.mesajgiden.Where(x => x.mesaj == "." && x.kimdenid == kimeid && x.kimeid == drid).FirstOrDefault();
        cevaplananmesajid = ilkmesaj.gidenmesajid;
    }
}
```

```

if (!mesaj.Equals(""))
{
    mesajgiden gidenmesaj = new mesajgiden
    {
        kimdenid = drid,
        kimeid = kimeid,
        tarih = DateTime.Now,
        mesaj = mesaj,
        cevaplananmesaj = cevaplananmesajid
    };
    hs.mesajgiden.Add(gidenmesaj);
    hs.SaveChanges();
}

return RedirectToAction("mesaj", kimeid);
}

```

4.9. Personel Ekleme ve Listeleme

Yönetici tarafından kontrol edilen alanlar.

İsim	TC	Branş	Detay	Bilgilerini Güncelle
Hakan Durmuş	46987456112	Kulak Burun Boğaz	Detay	Güncelle
Nihal Gündoğdu	46871235664	Beğen Cerrahi	Detay	Güncelle
Hatice San	41454646653	Kulak Burun Boğaz	Detay	Güncelle
Doğan Mersin	46972564641	Kardiyoloji	Detay	Güncelle
Melih Yaman	41006575334	Kulak Burun Boğaz	Detay	Güncelle
Nurşah Yılmaz	41878654313	Psikiyatrist	Detay	Güncelle
Ahmet Selim	41054682006	Kulak Burun Boğaz	Detay	Güncelle

Şekil 4.22. Doktor Listele Sayfası

Şekil 4.23. Personel Ekle Sayfası

```

29 [Authorize]
30 [HttpGet]
31 public ActionResult DoktorEkle()
32 {
33     ViewModels vm = new ViewModels();
34
35     List<SelectListItem> unvan = (from x in hs.unvan.ToList()
36                                   where x.kadroid != 5
37                                   select new SelectListItem
38                                   {
39                                       Text = x.unvan1,
40                                       Value = x.kadroid.ToString()
41                                   }).ToList();
42     ViewBag.unvan = unvan;
43
44     List<SelectListItem> klinik = (from x in hs.klinik.ToList()
45                                   select new SelectListItem
46                                   {
47                                       Text = x.klinik1,
48                                       Value = x.klinikid.ToString()
49                                   }).ToList();
50     ViewBag.klinik = klinik;
51     vm.iller = new SelectList(hs.iller, "id", "sehiradi");
52     vm.hastaneler = new SelectList(hs.hastane, "hastaneid", "hastaneadi");
53
54     return View(vm);
55 }

```

```

56 [Authorize]
57 [HttpPost]
58 0 references
59 public ActionResult DoktorEkle(personel p, kullanici_Yetki k, doktor d, string personelgorev)
60 {
61     if (p != null)
62     {
63         p.kalanizin = 8;
64         p.kullanilanizin = 0;
65         hs.personel.Add(p);
66         hs.SaveChanges();
67         var pers = hs.personel.Where(x => x.tc == p.tc).FirstOrDefault();
68         if (personelgorev == "D")
69         {
70             d.personelID = pers.personelid;
71             d.izindeMi = false;
72             hs.doktor.Add(d);
73             hs.SaveChanges();
74             k.kullaniciRol = "D";
75             k.klinik = true;
76             k.vezne = false;
77             k.personelid = pers.personelid;
78             hs.kullanici_Yetki.Add(k);
79             hs.SaveChanges();
80         }
81         return RedirectToAction("Index", "Yonetici");
82     }
83     return View();
84 }
85 
```

Şekil 4.24. Personel Ekle ve Listele Kod

4.10. Doktor İzin Onay

Doktorların oluşturduğu izin talepleri liste biçiminde ekrana geliyor ve “izin ver” butonuna tıklayarak izin verme işlemi gerçekleştiriliyor.

Anasayfa/Doktor İzin					
Personel Adı	İzin Türü	İzindeki Adres	İzin Başlangıç Tarihi	İzin Bitiş Tarihi	İzin
Hatice Sarı	Hastalık	Samsun/İlkadım	31.05.2022	2.06.2022	İzin Ver

Şekil 4.25. Doktor İzin Onay Ekranı

```

```
85 [HttpGet]
86 public ActionResult Doktorizininay()
87 {
88 doktorList dr = new doktorList();
89 dr.doktorListesi = hs.doktor.ToList();
90 dr.personelListesi = hs.personel.ToList();
91 dr.izinListesi = hs.izin.Where(x => x.onaylandimi == false).ToList();
92 return View(dr);
93 }
94 [HttpGet]
95 public ActionResult Doktorizinver(int? personelID, DateTime izinbitis, DateTime izinbas)
96 {
97 doktor doc = new doktor();
98 izin iz = new izin();
99 doc = hs.doktor.Where(x => x.personelID == personelID).FirstOrDefault();
100 iz = hs.izin.Where(x => x.personelID == personelID && x.onaylandimi == false).FirstOrDefault();
101 var per = hs.personel.Find(personelID);
102 TimeSpan kullanilanizin = izinbitis - izinbas;
103 double kullanilanizingun = kullanilanizin.TotalDays;
104 int day = Convert.ToInt32(kullanilanizingun);
105 if (doc != null)
106 {
107 iz.onaylandimi = true;
108 doc.izindeMi = true;
109 per.izinbitis = izinbitis;
110 per.kalanizin -= per.kalanizin - day;
111 hs.SaveChanges();
112 }
113 }
114 }
```

```

Şekil 4.26. Doktor İzin Onay Kod

```

HastaneEntities2 hs = new HastaneEntities2();
var doktor = hs.doktor.Where(x => x.izindeMi == true).ToList();
foreach (var dr in doktor)
{
    var pers = hs.personel.Find(dr.personelID);
    if (DateTime.Today > pers.izinbitis)
    {
        dr.izindeMi = false;
        hs.SaveChanges();
    }
}
```

```

Globalde tanımlanmış yukarıdaki kod ile izin tarihi bitmiş olan doktorların tespiti gerçekleştiriliyor.

## 4.11. Doktorlar ve Personeller

Doktorlar ve personeller sekmelerinin ara yüzü ve kodları oldukça benzer.

Aranan doktorun TC si ile ekrana o doktora ait bilgileri getirebiliyoruz. Doktora ait bilgilerin detayını görebilir ve doktora ait bilgileri güncelleyebiliriz.

| İsim           | TC          | Branş             | Detay | Bilgilerini Güncelle |
|----------------|-------------|-------------------|-------|----------------------|
| Hakan Durmuş   | 46987456112 | Kulak Burun Boğaz | Detay | Güncelle             |
| Nihal Gündoğdu | 46871235664 | Beğen Cerrahi     | Detay | Güncelle             |
| Hatice Sarı    | 41454646653 | Kulak Burun Boğaz | Detay | Güncelle             |
| Doğan Mersin   | 46972564641 | Kardiyoloji       | Detay | Güncelle             |
| Melih Yaman    | 41006575334 | Kulak Burun Boğaz | Detay | Güncelle             |
| Nurşah Yılmaz  | 41878654313 | Psikiyatrist      | Detay | Güncelle             |

Şekil 4.27. Doktorlar Ekranı

```
HastaneEntities2 hs = new HastaneEntities2();
// GET: Yonetici
[HttpGet]
[Authorize(Roles = "Y")]
public ActionResult Index(string doktortc)
{
 doktorList dr = new doktorList();
 dr.personellistesi = hs.personel.ToList();
 var degerler = from d in hs.doktor select d;
 if (!string.IsNullOrEmpty(doktortc))
 {
 degerler = degerler.Where(x => x.tc.Contains(doktortc));
 }
 dr.doktorListesi = degerler.ToList();
 return View(dr);
}
```

Şekil 4.28. Doktorlar Kod

```
[Authorize]
0 references
public ActionResult DoktorDetay(int? personelID)
{
 doktorList dl = new doktorList();
 dl.doktorListesi = hs.doktor.Where(x => x.personelID == personelID).ToList();
 return View(dl);
}
[Authorize]
[HttpGet]
0 references
public ActionResult DoktorGuncelle(int? personelID)
{
 if (personelID != null)
 {
 doktorList dl = new doktorList();
 dl.doktorListesi = hs.doktor.Where(x => x.personelID == personelID).ToList();
 return View(dl);
 }
 else
 {
 return RedirectToAction("Index", "Yonetici");
 }
}
```

Şekil 4.29. Doktorlar Detay ve Güncelle Kod

#### 4.12. Personel Ekle

Seçilen şehirde belirtilen hastanenin kliniğine radio button ile seçilen alanda personel ekleme işlemi gerçekleştirilir.

|                      |                                              |        |                       |         |                       |        |
|----------------------|----------------------------------------------|--------|-----------------------|---------|-----------------------|--------|
| Şehir                | SAMSUN                                       |        |                       |         |                       |        |
| Hastane              | Samsun Anadolu Hastanesi                     |        |                       |         |                       |        |
| Klinik               | Genel Cerrahi                                |        |                       |         |                       |        |
| Ad                   | Ahmet                                        |        |                       |         |                       |        |
| Soyad                | Selim                                        |        |                       |         |                       |        |
| TC                   | 41054682006                                  |        |                       |         |                       |        |
| Ünvan                | Prof. Dr.                                    |        |                       |         |                       |        |
| Branş                | 654320654320                                 |        |                       |         |                       |        |
| Diploma No           | 546506545324                                 |        |                       |         |                       |        |
| Mezun Olduğu Okul    | Istanbul Üniversitesi İstanbul Tıp Fakültesi |        |                       |         |                       |        |
| Sicil No             | 33346892                                     |        |                       |         |                       |        |
| Doğum Tarihi         | 13.10.1990                                   |        |                       |         |                       |        |
| Cinsiyet             | Erkek                                        |        |                       |         |                       |        |
| Email                | ahmet@gmail.com                              |        |                       |         |                       |        |
| Şifre                | *****                                        |        |                       |         |                       |        |
| Cep Numarası         | 05410354105                                  |        |                       |         |                       |        |
| Adresi               | Samsun/Atakum                                |        |                       |         |                       |        |
| Personel Alanı       | <input checked="" type="radio"/>             | Doktor | <input type="radio"/> | Hemşire | <input type="radio"/> | Hademe |
| <b>Personel Ekle</b> |                                              |        |                       |         |                       |        |

Şekil 4.30. Personel Ekleme Ekranı

```

29 [Authorize]
30 [HttpGet]
31 public ActionResult DoktorEkle()
32 {
33 ViewModels vm = new ViewModels();
34
35 List<SelectListItem> unvan = (from x in hs.unvan.ToList()
36 where x.kadroid != 5
37 select new SelectListItem
38 {
39 Text = x.unvan1,
40 Value = x.kadroid.ToString()
41 }).ToList();
42 ViewBag.unvan = unvan;
43
44 List<SelectListItem> klinik = (from x in hs.klinik.ToList()
45 select new SelectListItem
46 {
47 Text = x.klinik1,
48 Value = x.klinikid.ToString()
49 }).ToList();
50 ViewBag.klinik = klinik;
51 vm.iller = new SelectList(hs.iller, "id", "sehiradi");
52 vm.hastaneler = new SelectList(hs.hastane, "hastaneid", "hastaneadi");
53
54 return View(vm);
55 }

```

```

56 [Authorize]
57 [HttpPost]
58 public ActionResult DoktorEkle(personel p, kullanici_Yetki k, doktor d, string personelgorev)
59 {
60 if (p != null)
61 {
62 p.kalanizin = 8;
63 p.kullanilanizin = 0;
64 hs.personel.Add(p);
65 hs.SaveChanges();
66 var pers = hs.personel.Where(x => x.tc == p.tc).FirstOrDefault();
67 if (personelgorev == "D")
68 {
69 d.personelID = pers.personelid;
70 d.izindeMi = false;
71 hs.doktor.Add(d);
72 hs.SaveChanges();
73 k.kullaniciRol = "D";
74 k.klinik = true;
75 k.vezne = false;
76 k.personelid = pers.personelid;
77 hs.kullanici_Yetki.Add(k);
78 hs.SaveChanges();
79 }
80 return RedirectToAction("Index", "Yonetici");
81 }
82 return View();
83 }
84
85 }
```

Şekil 4.31. Personel Ekleme Kod

#### 4.13. Hastane CRUD İşlemleri

Dropdownlistde hastanenin ekleneceği ili seçtikten sonra hastane ismini yazarak ekliyoruz.

|                     |                     |
|---------------------|---------------------|
| Şehir               | ---Şehir Seçiniz--- |
| Hastane İsmi        | İsim Giriniz        |
| <b>Hastane Ekle</b> |                     |

Şekil 4.32. Hastane Ekle Sayfası

```

[HttpPost]
0 references
public ActionResult HastaneEkle(int? DPil, string hastaneadi)
{
 ViewModels vm = new ViewModels();
 vm.iller = new SelectList(hs.iller, "id", "sehiradi");
 var hst = hs.hastane.Where(x => x.hastaneadi == hastaneadi && x.ilid == DPil).FirstOrDefault();
 if (DPil != null && hastaneadi != null && hst == null)
 {
 hastane hastane = new hastane()
 {
 hastaneadi = hastaneadi,
 ilid = DPil
 };
 hs.hastane.Add(hastane);
 hs.SaveChanges();
 return RedirectToAction("Hastaneler", "Yonetici");
 }
 else
 {
 if (DPil == null)
 {
 ViewBag.hataliekleme = "Lütfen hastane seçiniz!";
 }
 else
 {
 ViewBag.hataliekleme = "Kaydetmek istediğiniz klinik zaten sisteme kayıtlı!";
 }
 return View(vm);
 }
}

```

Şekil 4.33. Hastane Ekle Kod

Hastanenin adı ve bulunduğu şehir ile hastane araması gerçekleştirilebiliyor. Listelenme şehir ismine göre A’dan Z’ye olacak şekilde gerçekleştiriyor.

The screenshot shows a web page with two search input fields at the top: 'Aranacak Hastane Adı:' and 'Şehir adı ile ara:'. Below these are four columns of data:

| Hastane İsmi                           | Hangi Şehirde | Hastane Güncelle | Hastane Sil |
|----------------------------------------|---------------|------------------|-------------|
| Acıbadem Adana Hastanesi               | ADANA         | Güncelle         | Sil         |
| Özel Adana Hastanesi                   | ADANA         | Güncelle         | Sil         |
| Acıbadem Taksim Hastanesi              | İSTANBUL      | Güncelle         | Sil         |
| İstanbul Eğitim ve Araştırma Hastanesi | İSTANBUL      | Güncelle         | Sil         |
| Samsun Eğitim Araştırma Hastanesi      | SAMSUN        | Güncelle         | Sil         |
| Samsun Anadolu Hastanesi               | SAMSUN        | Güncelle         | Sil         |
| Büyük Anadolu Hastanesi                | SAMSUN        | Güncelle         | Sil         |
| Gazi Devlet Hastanesi                  | SAMSUN        | Güncelle         | Sil         |

Şekil 4.34. Hastane Listele Ekranı

```

0 references
public ActionResult Hastaneler(string hastaneAd, string sehirAd)
{
 var degerler = from d in hs.hastane select d;
 if (!string.IsNullOrEmpty(hastaneAd))
 {
 degerler = degerler.Where(x => x.hastaneadi.Contains(hastaneAd));
 }
 if (!string.IsNullOrEmpty(sehirAd))
 {
 degerler = degerler.Where(x => x.iller.sehiradi.Contains(sehirAd));
 }
 var dgr = degerler.OrderBy(x => x.iller.sehiradi);
 return View(dgr.ToList());
}

```

Şekil 4.35. Hastaneler Kod

Hastane güncelleştirmek istendiğinde “Hastaneler” sekmesindeki “Güncelle” butonuna tıklayarak güncelleştirilmek istenen hastanenin id sini Html.Actionlink ile “HastaneGuncelle” controllerına gönderir.

```

<div class="col-md-2">
 <div class="list-group">

 Hastane Güncelle

 @foreach (var hastane in Model)
 {
 @Html.ActionLink(" Güncelle", "HastaneGuncelle", "Yonetici", new { hastaneID = hastane.hastaneid }, new { @class = "list-group-item list-group-item-action list-group-item-success active" })
 }
 </div>
</div>

```

|               |                          |
|---------------|--------------------------|
| Hastane Adı   | Acibadem Adana Hastanesi |
| Hangi Şehirde | ADANA                    |

Güncelle

Şekil 4.36. Hastane Güncelleme Ekranı

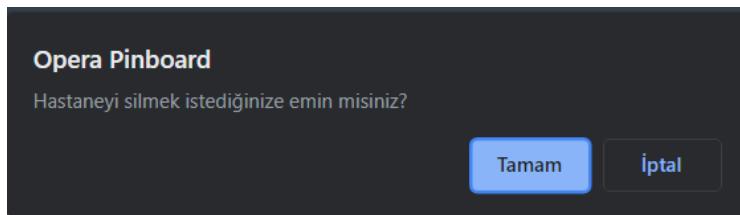
```

[Authorize]
[HttpGet]
0 references
public ActionResult HastaneGuncelle(int? hastaneID)
{
 doktorList dl = new doktorList();
 TempData["HastaneID"] = hastaneID;
 dl.hastaneListesi = hs.hastane.Where(x => x.hastaneid == hastaneID).ToList();
 return View(dl);
}
[Authorize]
[HttpPost]
0 references
public ActionResult HastaneGuncelle(string hastaneadi, string sehiradi)
{
 int hastaneid = Convert.ToInt32(TempData["HastaneID"]);
 var hastane = hs.hastane.Find(hastaneid);
 hastane.hastaneadi = hastaneadi;
 hastane.iller.sehiradi = sehiradi;
 hs.SaveChanges();
 return RedirectToAction("Hastaneler", "Yonetici");
}

```

Şekil 4.37. Hastane Güncelleme Kod

Hastane silinmek istendiğinde “Hastaneler” sekmesindeki “Sil” butonuna tıklandığında gelen onay ekranında tamama tıklandığında hastanenin id sini Html.Actionlink ile “HastaneSil” controllerına gönderir.



```

[Authorize]
[HttpGet]
0 references
public ActionResult HastaneSil(int? hastaneID)
{
 hastane hastane = hs.hastane.Find(hastaneID);
 hs.hastane.Remove(hastane);
 hs.SaveChanges();
 return RedirectToAction("Hastaneler", "Yonetici");
}

```

## 4.14. Klinik CRUD İşlemleri

Dropdownlistde kliniğin ekleneceği ilde ki hastaneyi seçtikten sonra klinik ismini ve yatak sayısını yazarak klinik ekleme işlemini gerçekleştiriyoruz.

|              |                        |
|--------------|------------------------|
| Şehir        | ---Şehir Seçiniz---    |
| Hastane      | ---Hastane Seçiniz---  |
| Klinik İsmi  | Klinik İsmi Giriniz    |
| Yatak Sayısı | Yatak sayısını Giriniz |

**Klinik Ekle**

Şekil 4.38. Klinik Ekle Sayfası

```
[HttpGet]
public ActionResult KlinikEkle()
{
 ViewModels vm = new ViewModels();
 vm.iller = new SelectList(hs.iller, "id", "sehiradi");
 vm.hastaneler = new SelectList(hs.hastane, "hastaneid", "hastaneadi");
 return View(vm);
}
[Authorize]
[HttpPost]
public ActionResult KlinikEkle(int? DPil, int? DPhastane, string klinik, string yataksayisi)
{
 ViewModels vm = new ViewModels();
 vm.iller = new SelectList(hs.iller, "id", "sehiradi");
 vm.hastaneler = new SelectList(hs.hastane, "hastaneid", "hastaneadi");
 var kln = hs.klinik.Where(x => x.klinik1 == klinik && x.hastane == DPhastane).FirstOrDefault();
 if (DPil != null && klinik != null && DPhastane != null && kln == null)
 {
 klinik yeniklinik = new klinik()
 {
 klinik1 = klinik,
 hastane = DPhastane,
 yataksayi = Convert.ToInt32(yataksayisi)
 };
 hs.klinik.Add(yeniklinik);
 hs.SaveChanges();
 return RedirectToAction("Klinikler", "Yonetici");
 }
 else
 {
 if (DPil == null || DPhastane == null)
 {
 ViewBag.hataliekleme = "Lütfen şehir ve hastane bilgilerini eksiksiz giriniz!";
 }
 else
 {
 ViewBag.hataliekleme = "Kaydetmek istediğiniz klinik zaten sisteme kayıtlı!";
 }
 }
 return View(vm);
}
```

Şekil 4.39. Klinik Ekle Kod

Kliniğin adı ve bulunduğu hastane ile hastane araması gerçekleştirilebiliyor. Listelenme hastane ismine göre A'dan Z'ye olacak şekilde gerçekleştiriyor.

| Klinik İsmi                 | Hangi Hastanede                        | Klinik Güncelle | Klinik Sil |
|-----------------------------|----------------------------------------|-----------------|------------|
| Kalp ve Damar               | Acıbadem Adana Hastanesi               | Güncelle        | Sil        |
| Genel Cerrahi               | Acıbadem Taksim Hastanesi              | Güncelle        | Sil        |
| Beyin ve Sinir Hastalıkları | Acıbadem Taksim Hastanesi              | Güncelle        | Sil        |
| Kalp ve Damar               | Acıbadem Taksim Hastanesi              | Güncelle        | Sil        |
| Kalp ve Damar               | Gazi Devlet Hastanesi                  | Güncelle        | Sil        |
| Kalp ve Damar               | İstanbul Eğitim ve Araştırma Hastanesi | Güncelle        | Sil        |
| Beyin ve Sinir Hastalıkları | İstanbul Eğitim ve Araştırma Hastanesi | Güncelle        | Sil        |
| Genel Cerrahi               | İstanbul Eğitim ve Araştırma Hastanesi | Güncelle        | Sil        |
| Genel Cerrahi               | Özel Adana Hastanesi                   | Güncelle        | Sil        |
| Genel Cerrahi               | Samsun Anadolu Hastanesi               | Güncelle        | Sil        |

Şekil 4.40. Klinik Listele Ekranı

```
[Authorize]
0 references
public ActionResult Klinikler(string hastaneAd, string klinik)
{
 var degerler = from d in hs.klinik select d;
 if (!string.IsNullOrEmpty(klinik))
 {
 degerler = degerler.Where(x => x.klinik1.Contains(klinik));
 }
 if (!string.IsNullOrEmpty(hastaneAd))
 {
 degerler = degerler.Where(x => x.hastane1.hastaneadi.Contains(hastaneAd));
 }
 var dgr = degerler.OrderBy(x => x.hastane1.hastaneadi);
 return View(dgr.ToList());
}
...
```

Şekil 4.41. Klinik Listele Kodu

Klinik güncelleştirmek istendiğinde “Klinikler” sekmesindeki “Güncelle” butonuna tıklayarak güncelleştirmek istenen kliniğin id sini Html.Actionlink ile “KlinikGuncelle” controllerına gönderir.

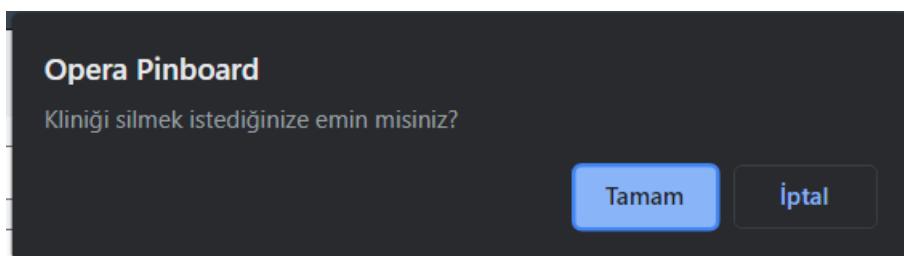


Şekil 4.42. Klinik Güncelle Sayfası

```
[Authorize]
[HttpGet]
0 references
public ActionResult KlinikGuncelle(int? klinikID)
{
 doktorList dl = new doktorList();
 TempData["klinikID"] = klinikID;
 dl.klinikListesi = hs.klinik.Where(x => x.klinikid == klinikID).ToList();
 return View(dl);
}
[Authorize]
[HttpPost]
0 references
public ActionResult KlinikGuncelle(string klinikadi)
{
 int klinikid = Convert.ToInt32(TempData["klinikID"]);
 var yeniklinik = hs.klinik.Find(klinikid);
 yeniklinik.klinik1 = klinikadi;
 hs.SaveChanges();
 return RedirectToAction("Klinikler", "Yonetici");
}
```

Şekil 4.43. Klinik Güncelleme Kod

Klinik silinmek istendiğinde “Klinikler” sekmesindeki “Sil” butonuna tıklandığında gelen onay ekranında tamama tıklandığında kliniğin id sini Html.Actionlink ile “KlinikSil” controllerine gönderir.



```
[HttpGet]
0 references
public ActionResult KlinikSil(int? klinikID)
{
 klinik klinik = hs.klinik.Find(klinikID);
 hs.klinik.Remove(klinik);
 hs.SaveChanges();
 return RedirectToAction("Klinikler", "Yonetici");
}
```

## 4.15. İlaç CRUD İşlemleri

Eklenmek istenen ilaçın ismini ve barkod numarasını girerek ekleme işlemi yapılır.



|             |                     |
|-------------|---------------------|
| İlaç İsmi   | İlaç İsmi Giriniz   |
| İlaç Barkod | İlaç Barkod Giriniz |

**İlaç Ekle**

Şekil 4.44. İlaç Ekle Sayfası

```
[Authorize]
[HttpGet]
0 references
public ActionResult ilacEkle()
{
 return View();
}
[Authorize]
[HttpPost]
0 references
public ActionResult ilacEkle(string ilacad, string ilacbarkod)
{
 var ilc = hs.ilac.Where(x => x.ilacad == ilacad && x.ilacbarkod == ilacbarkod).FirstOrDefault();
 if (ilacad != "" && ilacbarkod != "")
 {
 ilac yeniilac = new ilac()
 {
 ilacad = ilacad,
 ilacbarkod = ilacbarkod
 };
 hs.ilac.Add(yeniilac);
 hs.SaveChanges();
 return RedirectToAction("Ilaclar", "Yonetici");
 }
 else
 {
 ViewBag.hataliekleme = "Kaydetmek istediğiniz ilaç zaten sisteme kayıtlı!";
 return View();
 }
}
```

Şekil 4.45. İlaç Ekle Kod

İlacın ismi ve barkod numarası ile arama işlemi gerçekleştirilebiliyor. Listelenme ilaç ismine göre A’dan Z’ye olacak şekilde gerçekleştiriyor.

| İlaç İsmi                                    | İlaç Barkod   | İlaç Güncelle | İlaç Sil |
|----------------------------------------------|---------------|---------------|----------|
| ACNEMIX 46,6 GR JEL                          | 8699569340089 | Güncelle      | Sil      |
| CALPOL 120 MG 150 ML SUSPANSİYON             | 8699522705009 | Güncelle      | Sil      |
| CANDEXIL 16 MG 28 TABLET                     | 8699525010131 | Güncelle      | Sil      |
| CITOLES 10 MG 56 FILM TABLET                 | 8699514090939 | Güncelle      | Sil      |
| CLONEX 100 MG 50 TABLET                      | 8699587012449 | Güncelle      | Sil      |
| CONVULEX 150 MG 60 KAPSUL                    | 8699510190206 | Güncelle      | Sil      |
| DEKSTRAN-70 IZO.%09 500 ML SOLUSYON SETSİZ   | 8699788690279 | Güncelle      | Sil      |
| DOLOREX 50 MG 20 DRAJE                       | 8699514129110 | Güncelle      | Sil      |
| DOLVEN 400 MG 20 FILM TABLET                 | 8699809018556 | Güncelle      | Sil      |
| FERTAMIR 100 MG EFERVESAN GRANUL İCEREN SASE | 8699580210132 | Güncelle      | Sil      |

Şekil 4.46. İlaçlar Sayfası

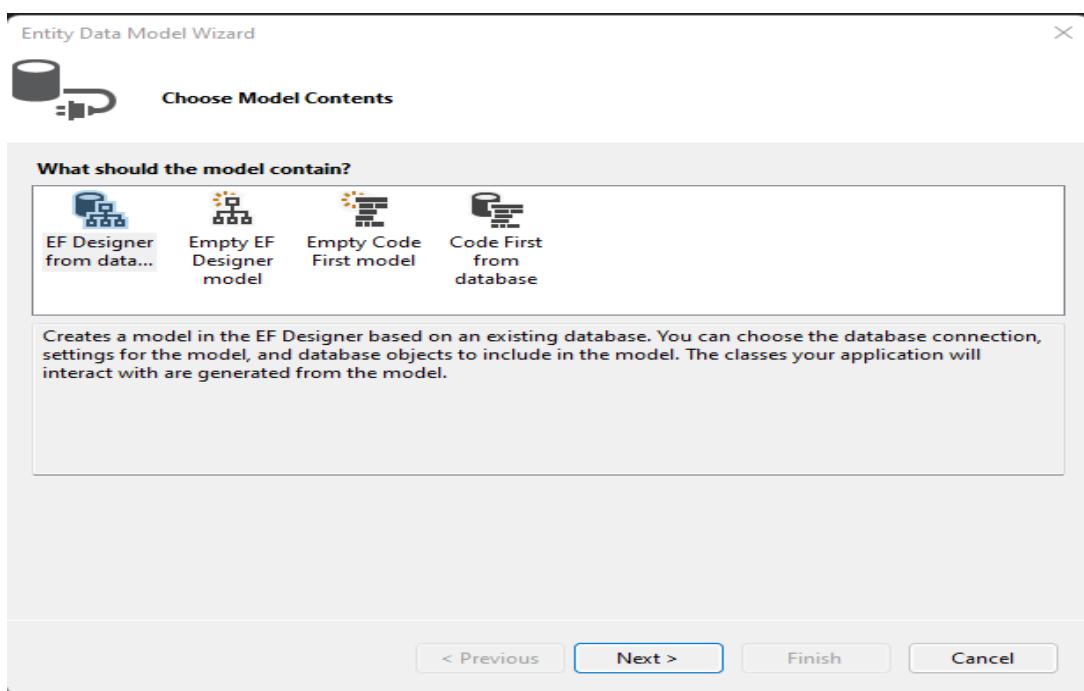
```
[Authorize]
0 references
public ActionResult Ilaclar(string ilacad, string ilacbarkod)
{
 var degerler = from d in hs.ilac select d;
 if (!string.IsNullOrEmpty(ilacad))
 {
 degerler = degerler.Where(x => x.ilacad.Contains(ilacad));
 }
 if (!string.IsNullOrEmpty(ilacbarkod))
 {
 degerler = degerler.Where(x => x.ilacbarkod.Contains(ilacbarkod));
 }
 var dgr = degerler.OrderBy(x => x.ilacad);
 return View(dgr.ToList());
}
```

Şekil 4.47. İlaç Listele Kodu

## 5. Eczane Otomasyonu Görünüm ve Açıklamaları

### 5.1. Eczane Veri Tabanı

MySQL'de oluşturduğum veri tabanını projeye ADO.NET Entity Data Model üzerinden EF Designer from database yaklaşımı ile dahil ettim. Modele dahil edilecek veritabanı bağlantısını, model için ayarları ve veritabanı nesnelerini seçtikten sonra uygulamamızın etkileşimde bulunacağı sınıflar modelden oluşturulur.



Şekil 5.1. MySQL Bağlantısı.

Eczane otomasyonu için MySQL'de "eczane" adlı veri tabanı oluşturdum. Eczane veri tabanına ait tabloları oluşturduğum Query kodları:

- `use eczane;`
- `create table` IF NOT EXISTS `kullanici(`  
  `kullanici_id int auto_increment,`  
  `kullanici_ad varchar(40) not null,`  
  `personel_id int,`  
  `kullanici_soyad varchar(40) not null,`  
  `kullanici_email varchar(70) not null,`  
  `kullanici_sifre varchar(20) not null,`  
  `PRIMARY KEY(kullanici_id, personel_id),`  
  `FOREIGN KEY(personel_id)`  
    `REFERENCES personel(personel_id)`  
  `ON UPDATE RESTRICT`  
  `ON DELETE CASCADE`  
`)ENGINE INNODB;`
- `use eczane;`
- `create table` IF NOT EXISTS `ilac(`  
  `ilac_id int auto_increment primary key,`  
  `ilac_ad varchar(100) not null,`  
  `ilac_barkod varchar(12) not null`  
`)ENGINE INNODB;`
- `use eczane;`
- `create table` IF NOT EXISTS `fatura(`  
  `fatura_id int auto_increment primary key,`  
  `hasta_ad varchar(120) not null,`  
  `hasta_tc varchar(11) not null,`  
  `hasta_odeme_turu varchar(10) not null,`  
  `odenen_tutar varchar(10) not null,`  
  `fatura_traihi Date not null`  
`)ENGINE INNODB;`

```

CREATE TABLE `personel` (
 `personelid` int NOT NULL AUTO_INCREMENT,
 `tc` varchar(11) NOT NULL,
 `ad` varchar(40) NOT NULL,
 `soyad` varchar(40) NOT NULL,
 `dtarih` date NOT NULL,
 `adres` varchar(45) DEFAULT NULL,
 `ceptel` varchar(11) DEFAULT NULL,
 `diplomano` varchar(11) NOT NULL,
 `mezuniyet` varchar(80) NOT NULL,
 `cinsiyet` varchar(5) NOT NULL,
 `kullanilanizin` int NOT NULL,
 `kalanizin` int NOT NULL,
 `izinbitis` datetime DEFAULT NULL,
 PRIMARY KEY (`personelid`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4

```

```

CREATE TABLE `recete` (
 `receteid` int NOT NULL AUTO_INCREMENT,
 `hastaad` varchar(80) NOT NULL,
 `hastatc` varchar(11) NOT NULL,
 `ilacbarkod` varchar(80) NOT NULL,
 `recetekodu` varchar(30) NOT NULL,
 `recetetarih` datetime NOT NULL,
 `ilacadet` varchar(15) DEFAULT NULL,
 `ilacVerildiMi` tinyint DEFAULT NULL,
 PRIMARY KEY (`receteid`)
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4

```

## 5.2. Reçeteli İlaç Ver

İlaç alacak olan hastanın bilgileri ve reçete kodu projenin global sayfasında tanımlanmış olan aşağıdaki kod ile hastane otomasyonunda yapılmış muayeneye ait sonucu Web Service üzerinden eczane veri tabanındaki recete adlı tabloya aktarılıyor.

```
eczaneEntities ec = new eczaneEntities();
WebService1SoapClient servis = new WebService1SoapClient();
List<receteTut> receteListesi = new List<receteTut>();
var receteList = servis.GetRecete();
foreach (var rct in receteList)
{
 recete dgr = new recete()
 {
 hastaad = rct.hastaad,
 hastatc = rct.hastatc,
 ilacbarkod = rct.ilackodu,
 recetekodu = rct.recetekodu,
 recetetarih = (DateTime)rct.recetetarih,
 ilacadet = rct.ilacadet,
 ilacVerildiMi = 0
 };
 var kontrol = ec.recete.Where(x => x.recetekodu == dgr.recetekodu).FirstOrDefault();
 if (kontrol == null)
 {
 ec.recete.Add(dgr);
 ec.SaveChanges();
 }
}
```

---

### GetRecete

#### Sınaması

HTTP POST protokolunu kullanarak işlemi sınamak için, 'Çağır' düğmesini tıklayın.



#### SOAP 1.1

Aşağıdaki örnekte bir SOAP 1.1 isteği ve yanıtı yer almıyor. Gösterilen yer tutucular gerçek değerlerle değiştirilmelidir.

```
POST /WebService/WebService1.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/GetRecete"
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
 <GetRecete xmlns="http://tempuri.org/" />
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
 <GetReceteResponse xmlns="http://tempuri.org/">
 <GetReceteResult>
 <ReceteItem>
 <ilackodu>string</ilackodu>
 <hastaad>string</hastaad>
 <hastatc>string</hastatc>
 <recetekodu>string</recetekodu>
 <ilacadet>string</ilacadet>
 <recetetarih>dateTime</recetetarih>
 </ReceteItem>
 <ReceteItem>
 <ilackodu>string</ilackodu>
 <hastaad>string</hastaad>
 <hastatc>string</hastatc>
 <recetekodu>string</recetekodu>
 <ilacadet>string</ilacadet>
 <recetetarih>dateTime</recetetarih>
 </ReceteItem>
 </GetReceteResult>
 </GetReceteResponse>
</soap:Body>
</soap:Envelope>
```

Şekil 5.2. Web Service GetRecete Fonksiyonu

İlaç almak isteyen kişinin hastanede ki muayenesi sonunda aldığı reçete kodu ile veya reçetesini kaybettiği durumu göz önünde bulundurarak hastanın TC'si ile reçete aratma işlemi gerçekleştiriliyor.

Şekil 5.3. İlaç Ver Sayfası

```
[Authorize]
[HttpGet]
0 references
public ActionResult ilacVer(string recetekodu, string hastatc)
{
 string tutRecete = "";
 string tutTC = "";
 if (recetekodu != "" || hastatc != "")
 {
 tutRecete = recetekodu;
 tutTC = hastatc;
 }
 var degerler = from d in ec.recete where d.recetekodu == tutRecete || d.hastatc == tutTC select d;
 if (!string.IsNullOrEmpty(recetekodu))
 {
 degerler = degerler.Where(x => x.recetekodu.Contains(recetekodu));
 }
 if (!string.IsNullOrEmpty(hastatc))
 {
 degerler = degerler.Where(x => x.hastatc.Contains(hastatc));
 }

 return View(degerler.ToList());
}
```

Şekil 5.4. İlaç Ver Kod

“İlaç Ver” Butonuna tıklandığında “İlaç Ver Onay” sayfasına yönlendiriliyoruz. Bu sayfada hastaya hangi ilaçtan kaç adet verileceği ve hastanın ödeyeceği toplam tutar ekrana geliyor. “İlaç Ver” butonuna tıklayarak ilaç verme işlemini onaylamış oluyoruz. İlaç verme işlemi gerçekleştirildikten sonra verilen ilaçlar stoktan düşürülmüyor.

Anasayfa/İlaç Ver/İlaç Ver Onay

|                    |                                         |                                  |                       |
|--------------------|-----------------------------------------|----------------------------------|-----------------------|
| Hasta Adı          | Bülent Yaşar                            |                                  |                       |
| Hasta TC           | 46974658798                             |                                  |                       |
| İlaç İsmi ve Adeti | CONVULEX 150 MG 60 KAPSUL - 2           | İlaç Fiyatı: 35 TL               | Stokta 17 adet var.   |
| İlaç İsmi ve Adeti | LOCODERM %0.1 30 GR MERHEM - 1          | İlaç Fiyatı: 25,53 TL            | Stokta 32 adet var.   |
| İlaç İsmi ve Adeti | VERMIDON 500/30 MG TABLET 30 TABLET - 1 | İlaç Fiyatı: 11 TL               | Stokta 37 adet var.   |
| Hastanın Ödeyeceği | 95,877 TL                               |                                  |                       |
| Ödeme Türü         | Nakit                                   | <input checked="" type="radio"/> | <input type="radio"/> |
| <b>İlaç ver</b>    |                                         |                                  |                       |

Şekil 5.5. İlaç Ver Onay Sayfası

```

public ActionResult ilacVerOnay(double? ilactutar, int? receteID, string odemeTuru,int? foricin)
{
 int personelid = Convert.ToInt32(Session["PersonelID"]);
 if (receteID != null && ilactutar != null)
 {
 var recete = ec.recete.Find(receteID);
 recete.ilacVerildiMi = 1;
 ec.SaveChanges();
 string ilacismi = "";

 for (int i = 0; i < foricin; i++)
 {
 int ilacID = Convert.ToInt32(TempData["id" + i]);
 int stokyok = Convert.ToInt32(TempData["stokyok" + i]);
 var ilacname = ec.ilac.Find(ilacID);
 ilacismi += " " + ilacname.ilac_ad + " - " + TempData["adet" + i].ToString();
 if (ilacID != stokyok)
 {
 int ilacadet = Convert.ToInt32(TempData["adet" + i].ToString());
 var ilac = ec.ilac.Find(ilacID);
 ilac.ilacadet -= ilacadet;
 ec.SaveChanges();
 }
 }
 fatura ftr = new fatura()
 {
 hasta_ad = recete.hastaad,
 hasta_tc = recete.hastatc,
 fatura_traihi = DateTime.Now,
 odenen_tutar = Convert.ToDouble(ilactutar),
 hasta_odeme_turu = odemeTuru,
 faturayikesenid = personelid,
 satilanilac=ilacismi
 };
 ec.fatura.Add(ftr);
 ec.SaveChanges();
 return RedirectToAction("ilacVer", "Eczaci");
 }
}

```

Şekil 5.6. İlaç Ver Kod

### 5.3. Faturalar

Giriş yapmış olan personelin yapmış olduğu fatura kesimleri listeleniyor. Listeleme işlemi eski fatura tarihinden yeniye doğru olacak şekilde yapılıyor. Hastanın TC sine göre fatura aranan fatura bulunabiliyor.

| Anasayfa/Faturalar |             |               |              |              |
|--------------------|-------------|---------------|--------------|--------------|
| Hasta Adı          | Hasta TC    | Fatura Tarihi | Ödenen Tutar | Fatura Detay |
| Efe Taş            | 46842133322 | 27.05.2022    | 122,23 TL    | Detay        |
| Bekir Gündoğdu     | 46972487654 | 27.05.2022    | 16 TL        | Detay        |

Şekil 5.7. Faturalar Sayfası

| Anasayfa/Faturalar/Fatura Detay |                                                                                                      |
|---------------------------------|------------------------------------------------------------------------------------------------------|
| <b>Hasta Adı</b>                | Bülent Yaşar                                                                                         |
| <b>TC</b>                       | 46974658798                                                                                          |
| <b>Ödeme Tarihi</b>             | 6.06.2022                                                                                            |
| <b>Ödenen Tutar</b>             | 95,877 TL                                                                                            |
| <b>Ödeme Türü</b>               | Nakit                                                                                                |
| <b>Satılan İlaç</b>             | CONVULEX 150 MG 60 KAPSUL - 2 LOCODERM %0.1 30 GR MERHEM - 1 VERMIDON 500/30 MG TABLET 30 TABLET - 1 |
| <b>Faturayı Kesen</b>           | Neslihan Kaplan                                                                                      |

Şekil 5.8. Faturalar Detay Sayfası

```

[Authorize]
[HttpGet]
0 references
public ActionResult Faturalar(string hastatc)
{
 int personelid = Convert.ToInt32(Session["PersonelID"]);
 var degerler = from d in ec.fatura where d.faturayikesenid == personelid select d;
 if (!string.IsNullOrEmpty(hastatc))
 {
 degerler = degerler.Where(x => x.hasta_tc.Contains(hastatc));
 }
 var dgr = degerler.OrderBy(x => x.fatura_traihi);
 if (degerler.Count() == 0)
 {
 ViewBag.faturayok="Henüz fatura kesilmemiş";
 }
 return View(dgr.ToList());
}
[Authorize]
[HttpGet]
0 references
public ActionResult FaturaDetay(int? faturaID)
{
 stokList sl = new stokList();
 sl.faturaListesi = ec.fatura.Where(x => x.fatura_id == faturaID).ToList();
 return View(sl);
}

```

Şekil 5.9. Faturalar Kod

#### 5.4. Personel CRUD İşlemleri

Personeller ekle sayfasında personele ait bilgileri girerek

|                   |                    |
|-------------------|--------------------|
| Ad                | İsim Giriniz       |
| Soyad             | Soyad Giriniz      |
| TC                | TC Giriniz         |
| Doğum Tarihi      | gg.aa.yyyy         |
| Diploma No        | Diploma no Giriniz |
| Mezun Olduğu Okul | Okul İsmi giriniz  |
| Cinsiyet          | Cinsiyet Seçiniz   |
| Email             | Emaili Giriniz     |
| Sıfır             | Şifre Giriniz      |
| Sıfır             | Şifre Giriniz      |
| Cep Numarası      | Numara Giriniz     |
| Adresi            | Adres Giriniz      |

**Personel Ekle**

Şekil 5.10. Personel Ekle Sayfası

```

public ActionResult PersonelEkle(string ad, string soyad, string tc, DateTime dtarih, string diplomano, string mezuniyet,
{
 if (sifre == sifretekrari)
 {
 string tarih = "1900-01-01";

 personel prs = new personel()
 {
 ad = ad,
 soyad = soyad,
 tc = tc,
 dtarih = dtarih,
 adres = adres,
 cinsiyet = cinsiyet,
 ceptel = ceptel,
 diplomano = diplomano,
 mezuniyet = mezuniyet,
 izinbitis = Convert.ToDateTime(tarih),
 kullanilaninizin = 0,
 kalaninizin = 8
 };
 ec.personel.Add(prs);
 ec.SaveChanges();
 var personel = ec.personel.Where(x => x.tc == tc).FirstOrDefault();
 kullanici_yetki ky = new kullanici_yetki()
 {
 personel_id = personel.personelid,
 kullanici_ad = ad,
 kullanici_soyad = soyad,
 kullanicirol = "E",
 kullanici_sifre = sifre,
 kullanici_email = email
 };
 ec.kullanici_yetki.Add(ky);
 ec.SaveChanges();
 return RedirectToAction("PersonelListele", "Yonetici");
 }
}

```

Şekil 5.11. Personel Ekle Kod

Bulmak istenen personel adı ve TC ile aratılabilir.

The screenshot shows a web application interface for managing employees. On the left is a dark sidebar menu with the following items:

- Eczane
- Serkan Işık
- Anasayfa
- Personeller
- Personel Ekle
- İlaç Stok Görüntüle
- İlaç Ekle
- İlaçlar
- Çıkış

The main content area has a header "Anasayfa/Personeller". It contains two search input fields: "Personel Adı ara:" and "TC ile ara:", each with an "Ara" button. Below these are four buttons: "İsim", "TC", "Detay", and "Bilgilerini Güncelle". A table lists three employees:

| İsim            | TC          | Detay | Bilgilerini Güncelle |
|-----------------|-------------|-------|----------------------|
| Levent Sağlam   | 41878650313 | Detay | Güncelle             |
| Neslihan Kaplan | 46986483556 | Detay | Güncelle             |
| Serkan Işık     | 43865190604 | Detay | Güncelle             |

On the far right, there is a column labeled "İşten Çıkar" with three red buttons, each labeled "Çıkar".

Şekil 5.12. Personeller Sayfası

```

[HttpGet]
[Authorize]
0 references
public ActionResult PersonelListele(string personelad,string personeltc)
{
 var degerler = from d in ec.personel select d;
 if (!string.IsNullOrEmpty(personelad))
 {
 degerler = degerler.Where(x => x.ad.Contains(personelad));
 }
 if (!string.IsNullOrEmpty(personeltc))
 {
 degerler = degerler.Where(x => x.tc.Contains(personeltc));
 }
 var dgr = degerler.OrderBy(x => x.ad);
 return View(dgr.ToList());
}

```

Şekil 5.13. Personeller Kod

|              |                                              |
|--------------|----------------------------------------------|
| Ad           | Levent                                       |
| Soyad        | Sağlam                                       |
| TC           | 41878650313                                  |
| Email        | levent@gmail.com                             |
| Şifre        | lvt55                                        |
| Cep Telefonu | 05416819850                                  |
| Adres        | Samsun/Canik Çay Mah. 2054.Sok               |
| Doğum Tarihi | 12.03.1986                                   |
| Cinsiyet     | Erkek                                        |
| Mezuniyet    | İstanbul Üniversitesi İstanbul Tıp Fakültesi |
| Diploma No   | 65781123466                                  |

**Güncelle**

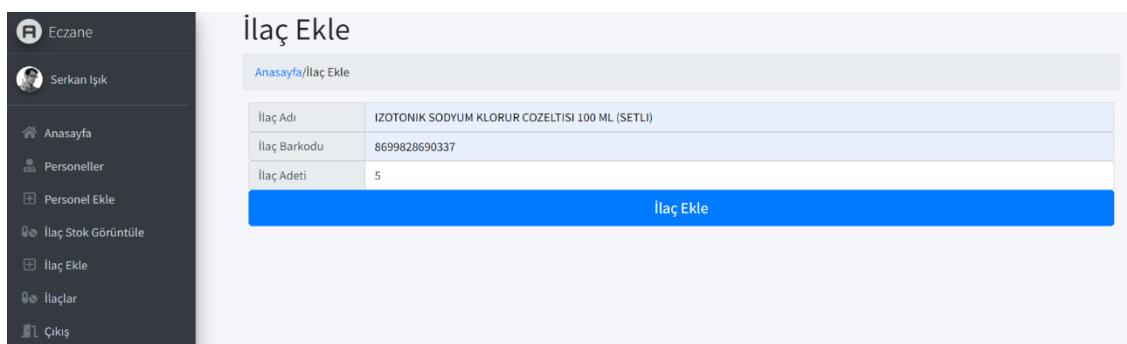
Şekil 5.14. Personel Güncelle Sayfası

```
public ActionResult PersonelGuncelle(int? personelID)
{
 if (personelID != null)
 {
 personelList pl = new personelList();
 pl.personellistesi = ec.personel.Where(x => x.personelid == personelID).ToList();
 return View(pl);
 }
 else
 {
 return RedirectToAction("Personellistele", "Yonetici");
 }
}
[HttpPost]
0 references
public ActionResult PersonelGuncelle(personel p,int? personelID,string email,string sifre)
{
 if (p != null && email !=null && sifre !=null)
 {
 var pers = ec.personel.Find(personelID);
 pers.ad = p.ad;
 pers.soyad = p.soyad;
 pers.tc = p.tc;
 pers.mezuniyet = p.mezuniyet;
 pers.ceptel = p.ceptel;
 pers.adres = p.adres;
 pers.cinsiyet = p.cinsiyet;
 pers.diplomano = p.diplomano;
 ec.SaveChanges();
 var kullanici = ec.kullanici_yetki.Where(x => x.personel_id == personelID).FirstOrDefault();
 kullanici.kullanici_email = email;
 kullanici.kullanici_sifre = sifre;
 return RedirectToAction("Personellistele", "Yonetici");
 }
 else
 {
 ViewBag.Hata = "Hatalı bilgi girişi!";
 return View();
 }
}
```

Şekil 5.15. Personel Güncelle Kod

## 5.6. İlaç CRUD İşlemleri

Ekleme istenen ilaçın adını, barkodunu ve kaç adet ekleneceğini girerek ilaç ekleme işlemini tamamlamış oluyoruz.



Şekil 5.16. İlaç Ekle Sayfası

```
public ActionResult ilacEkle()
{
 return View();
}
[HttpPost]
public ActionResult ilacEkle(ilac p)
{
 if (p != null)
 {
 ec.ilac.Add(p);
 ec.SaveChanges();
 return RedirectToAction("ilacStok");
 }
 else
 {
 return View();
 }
}
```

Şekil 5.17. İlaç Ekle Kod

İlaçlar sekmesinde sisteme kayıtlı ilaçları görüyoruz. Aranan ilaçı barkodu ve ismi aracılığıyla aratabiliyoruz. Güncellemek istediğimiz ilaçı “Güncelle” butonuna tıklayarak güncelleme ekranına yönlendiriliyoruz ve yapacağımız değişiklikleri onaylıyoruz. Silmek istediğimiz ilaçı “Sil” butonuna tıkladığımızda gelen uyarı mesajını onayladıkten sonra silme işlemi gerçekleşiyor.

| İlaç İsmi                                  | Barkod        | İlaç Güncelle | İlaç Sil |
|--------------------------------------------|---------------|---------------|----------|
| CALPOL 120 MG 150 ML SUSPANSİYON           | 8699522705009 | Güncelle      | Sil      |
| CANDEXIL 16 MG 28 TABLET                   | 8699525010131 | Güncelle      | Sil      |
| CITOLES 20 MG 56 FILM TABLET               | 8699514090939 | Güncelle      | Sil      |
| CLONEX 100 MG 50 TABLET                    | 8699587012449 | Güncelle      | Sil      |
| CONVULEX 150 MG 60 KAPSUL                  | 8699510190206 | Güncelle      | Sil      |
| DEKSTRAN-70 IZO.%09 500 ML SOLUSYON SETSİZ | 8699788690279 | Güncelle      | Sil      |
| DOLOREX 50 MG 20 DRAJE                     | 8699514129110 | Güncelle      | Sil      |
| DOLVEN 400 MG 20 FİLM TABLET               | 8699809018556 | Güncelle      | Sil      |
| INFEX 200 MG 20 FILM TABLET                | 8697927090034 | Güncelle      | Sil      |

Şekil 5.18. İlaçlar Sayfası

```
[HttpGet]
[Authorize]
0 references
public ActionResult ilacListele(string ilac_barkod, string ilac_ad)
{
 var degerler = from d in ec.ilac select d;
 if (!string.IsNullOrEmpty(ilac_barkod))
 {
 degerler = degerler.Where(x => x.ilac_barkod.Contains(ilac_barkod));
 }
 if (!string.IsNullOrEmpty(ilac_ad))
 {
 degerler = degerler.Where(x => x.ilac_ad.Contains(ilac_ad));
 }
 var dgr = degerler.OrderBy(x => x.ilac_ad);
 return View(dgr.ToList());
}
```

Şekil 5.19. İlaçlar Kod

```

public ActionResult ilacGuncelle(int? ilacID)
{
 stokList sl = new stokList();
 sl.ilacListesi = ec.ilac.Where(x => x.ilac_id == ilacID).ToList();
 return View(sl);
}
[HttpPost]
0 references
public ActionResult ilacGuncelle(string ilac_ad, string ilac_barkod, double? ilacfiyat, int? ilacID)
{
 if(ilac_ad != null && ilac_barkod != null && ilacfiyat != null)
 {
 var ilac = ec.ilac.Find(ilacID);
 ilac.ilac_ad = ilac_ad;
 ilac.ilac_barkod = ilac_barkod;
 ilac.ilacfiyat = ilacfiyat;
 ec.SaveChanges();
 return RedirectToAction("ilacListele", "Yonetici");
 }
 else
 {
 ViewBag.Hata = "Hatalı bilgi girişi";
 stokList sl = new stokList();
 sl.ilacListesi = ec.ilac.Where(x => x.ilac_id == ilacID).ToList();
 return View(sl);
 }
}
[HttpGet]
[Authorize]
0 references
public ActionResult ilacSil(int? ilacID)
{
 var ilac = ec.ilac.Find(ilacID);
 ec.ilac.Remove(ilac);
 ec.SaveChanges();
 return RedirectToAction("ilacListele", "Yonetici");
}

```

Şekil 5.20. İlaçlar Güncelle ve Sil Kodları

Önceden kayıt edilmiş ilaçların stokunu güncelleyebiliyoruz.

| İlaç barkodu ile ara:            | <input type="text"/> | Ara        | İlaç adı ile ara: | <input type="text"/> | Ara |
|----------------------------------|----------------------|------------|-------------------|----------------------|-----|
| İlaç İsmi                        | Barkod               | İlaç Adeti | Stok Ekle         |                      |     |
| CALPOL 120 MG 150 ML SUSPANSİYON | 8699522705009        | 0          | İlaç stok ekle    |                      |     |
| CANDEXIL 16 MG 28 TABLET         | 8699525010131        | 10         | İlaç stok ekle    |                      |     |
| CITOLES 20 MG 56 FILM TABLET     | 8699514090939        | 23         | İlaç stok ekle    |                      |     |
| CLONEX 100 MG 50 TABLET          | 8699587012449        | 60         | İlaç stok ekle    |                      |     |

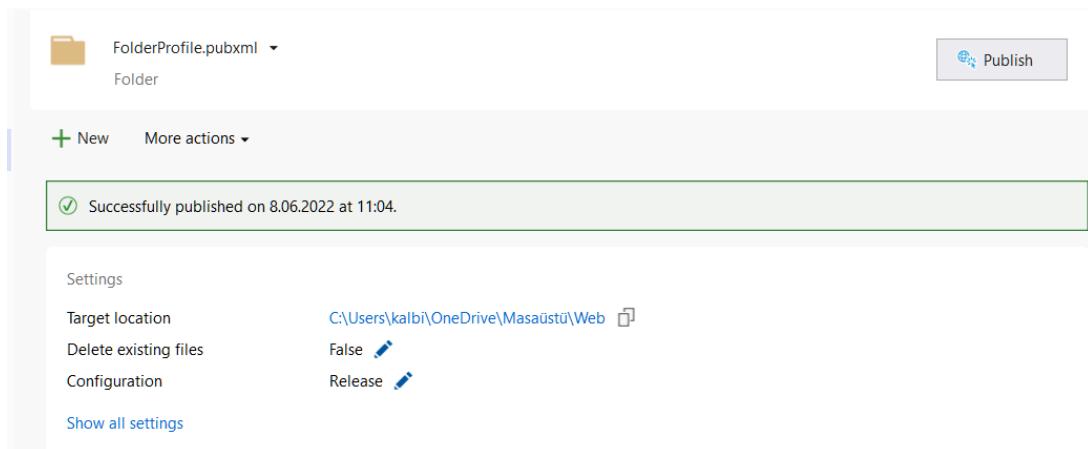
  

|                  |                         |
|------------------|-------------------------|
| İlaç Adı         | CLODEX 100 MG 50 TABLET |
| İlaç Barkodu     | 8699587012449           |
| İlaç Adeti       | 60                      |
| Eklenecek Adet   | İlaç Adeti Giriniz      |
| <b>Adet Ekle</b> |                         |

Şekil 5.21. İlaç Stok Ekle Sayfası

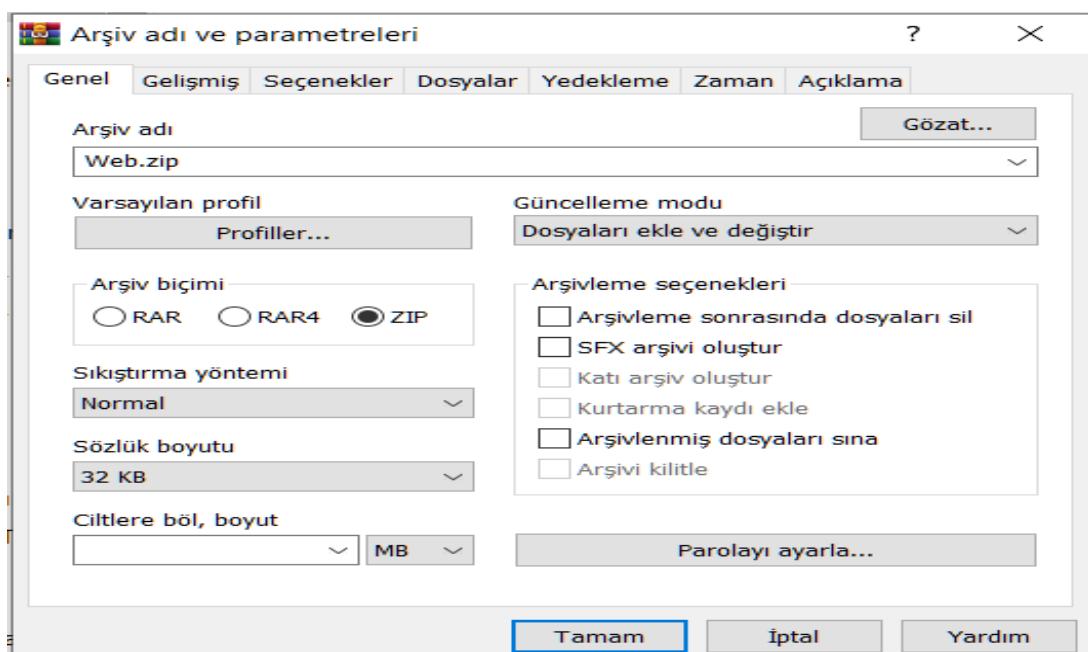
## 6. Projeyi Somee Sunucusuna Yükleme

Projeyi web sunucusuna yüklemek için önce proje dosyalarını “publish” etmemiz lazım.



Şekil 6.1. Projeyi Publish Etme

Publish ettikten sonra oluşan dosyaları zip formatında arşive ekliyoruz.



Şekil 6.2. Proje Dosyalarını Zipleme

Dosyalar ziplendiğinde Somee'de “FileManager” kısmında ziplediğimiz dosyayı upload ediyoruz.

|   | File name            | Size      | Last modified      |
|---|----------------------|-----------|--------------------|
| □ | AdminLTE-3.0.4       |           | 8.06.2022 03:13:28 |
| □ | bin                  |           | 8.06.2022 03:12:40 |
| □ | bootstrap-5.1.2-dist |           | 8.06.2022 03:12:41 |
| □ | Content              |           | 8.06.2022 03:12:41 |
| □ | Models               |           | 8.06.2022 03:12:41 |
| □ | Scripts              |           | 8.06.2022 03:12:41 |
| □ | Tasarim              |           | 8.06.2022 03:12:41 |
| □ | Views                |           | 8.06.2022 03:12:42 |
| □ | WebService           |           | 8.06.2022 03:12:42 |
| □ | AdminLTE-3.0.4.zip   | 23,708 KB | 8.06.2022 03:13:28 |
| □ | favicon.ico          | 31 KB     | 8.06.2022 03:12:42 |
| □ | Global.asax          | 1 KB      | 8.06.2022 03:12:42 |
| □ | libman.json          | 1 KB      | 8.06.2022 03:12:42 |
| □ | Web.config           | 5 KB      | 8.06.2022 03:12:42 |

Web sayfamızı bu haliyle açmaya kalkarsak “customErrors” ve “ConnectionString” hatası alıyoruz. Bu hataları düzeltmek için “Web.config” dosyamızı “Edit text” ile açıp içerisinde eksik kodları ekliyoruz.

### Eksik Kodlar:

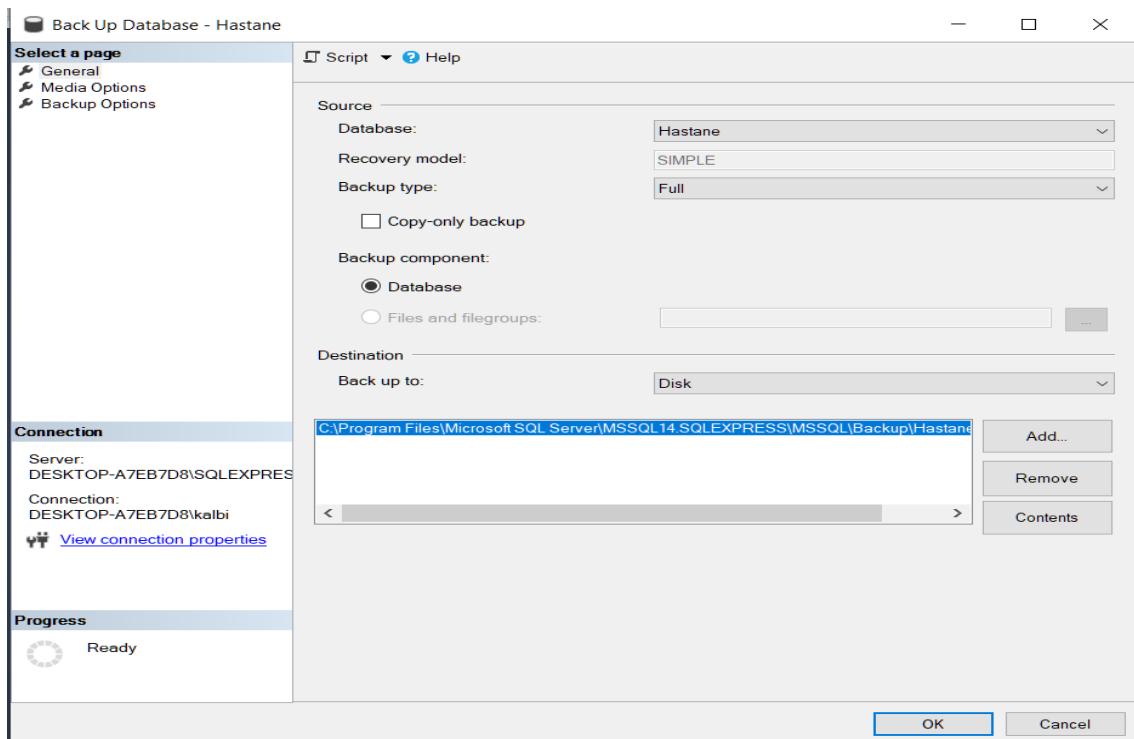
System.web'in içerisinde <customErrors mode="Off"/> yazıyoruz.

connectionString'in içerisinde;

```
<add name="HastaneEntities2" connectionString="metadata=res://*/;provider=System.Data.SqlClient;provider connection string='workstation id=DBHastane.mssql.somee.com;packet size=4096;user id=eren55_SQLLogin_1;pwd=o9fwbs3iaf;data source=DBHastane.mssql.somee.com;persist security info=False;initial catalog=DBHastane;" providerName="System.Data.EntityClient" />
```

Yukarıdaki kodu yazıyoruz.

Veri tabanımızı sunucuya eklemek için öncelikle veritabanımızın backupını oluşturmamız gerekiyor. Backupı oluşturmak için veri tabanımıza sağ tıklayıp “task” sekmesinden “Back Up” seçeneğini seçiyoruz.



Şekil 6.3. Veri Tabanı Backup Oluşturma

“Ok” butonuna tıkladığımızda “.bak” uzantılı bir dosya elde ediyoruz. Somee’de “Database” sekmesinde boş bir veri tabanı oluşturuyoruz.

MS SQL plan: MS SQL Plan "Novice"

Database name (7-30 symbols): DBHastane

Zone name: somee.com

MS SQL login: eren55\_SQLLogin\_1

MS SQL Server version: MS SQL 2019 Express

The process may take up to several minutes. Please be patient.

**CREATE EMPTY DATABASE**

Şekil 6.4. Veri Tabanı Oluşturma

Oluşan bu veri tabanını doldurmak için veri tabanını restore et seçeneği ile önceden oluşturduğumuz “.bak” dosyasını upload ediyoruz.

MS SQL database DBHastane

General    Backup file manager    Run T-SQL    **Restore**    Backup    Attach    Get database files    Delete

Restore to database: **DBHastane**

Change default schema to 'dbo'

① Upload your MS SQL database backup file. The backup file usually has a .bak extension.  
You can get a backup file by using SSMS.  
Open SSMS on your local computer, right click on the Database, and click on 'Back Up...' in the 'Tasks' sub menu.

BROWSE AND UPLOAD BACKUP FILE    Or drop your backup file here

Böylelikle web sunucumuz hazır hale geliyor.

Anasayfa/Hastaneler

Aranacak Hastane Adı:  Ara

Şehir adı ile ara:  Ara

Hastane İsmi	Hangi Şehirde	Hastane Güncelle	Hastane Sil
Acıbadem Adana Hastanesi	ADANA	Güncelle	Sil
Özel Adana Hastanesi	ADANA	Güncelle	Sil
Acıbadem Taksim Hastanesi	İSTANBUL	Güncelle	Sil
İstanbul Eğitim ve Araştırma Hastanesi	İSTANBUL	Güncelle	Sil
Samsun Eğitim Araştırma Hastanesi	SAMSUN	Güncelle	Sil
Samsun Anadolu Hastanesi	SAMSUN	Güncelle	Sil
Büyük Anadolu Hastanesi	SAMSUN	Güncelle	Sil
Gazi Devlet Hastanesi	SAMSUN	Güncelle	Sil

hastanerandevu.somee.com/Hasta/Randevu

**Hastane Randevu**    Anasayfa    Randevu Al    Randevularım    Hakkımızda    İletişim    Bilgilerim    Çıkış Yap

Randevu Al

Anasayfa > Randevu Al

...Şehir Seçiniz...  
...Hastane Seçiniz...  
...Klinik Seçiniz...  
...Doktor Seçiniz...  
Randevu Tarihi Seçiniz  
gg.aa.yyyy  
Seçilen Tarihi Onayla

**Hastane Randevu**

Olupardıoğlu profesyonel kadrosu ve güvenli altyapısıyla Hastanelerimiz siz hastalarımıza en yi hizmeti vermemey hedefliyor.

**İLETİŞİM**  
sancaktepe/fakkekoy 2054.Sok. No:10  
+90(543)6.010458  
crenym3@gmail.com

**SON TWEETLER**  
Sosyal Bağlantı, Koronavirüs risk tanımına cat kat değil, planlayarak uyarınca kader. Sıri 184'ten araya bilgilendiriyor, önden sonra gelişen, SADEM 184'ten size TELEFON GELMEZ. İsteklerinizi yerine getirmek isteyen söyleşenleri içeri almanın KÖTÜ NİFTİ İLE FETE KAŞT DA TEDBİRİ!

**Haber Bülteni**  
Vakılannıza zararlı bir şantaj hevesi yerinde da bilinmesinler, koronavirüs peşini bırakmadır. Virüsü yelen takip altına alan uzmanlar, hem COVID-19'un normal enfeksiyon率a neden olabileceğini, hem de sonbaharda kadar yeni bir eylemle yüzünü yollarına ata.

Şekil 6.5. Sunucu Üzerinden Açılmış Sayfalar

## **7. SONUÇLAR VE ÖNERİLER**

Projenin bu haliyle bir hasta kolay bir şekilde istediği hastaneden, istediği doktordan randevu alabilir. Doktor kendinden randevu alan hastaları görebilir ve vakti geldiğinde hastasını muayene edebilir.

Sistem üzerinden hastane, klinik ve personeller üzerinde CRUD işlemleri yapılabilir. Muayene sonucu hastaya verilen ilaç reçetesi eczane sistemine “Web Service” ile iletildikinden hasta ilaçlarını eczanededen kolay bir şekilde alması sağlanmıştır.

Eczane sistemi üzerinde ilaç ve personel CRUD işlemleri yapılabilir. Hastaya verilecek ilaçlar ve hastanın ödeyeceği tutar otomatik şekilde ekran'a getirilir.

## KAYNAKLAR

1. <https://medium.com/@kdrcandogan/mvc-nedir-mvc-yaşam-döngüsü-life-cycle-8e124f24650c>
2. <https://www.webtekno.com/microsoft-visual-studio-nedir-h92228.html>
3. <https://www.penta.com.tr/markalar/microsoft/urunler/microsoft-sql-server/microsoft-sql-server-nedir/>
4. <https://www.srdrylmz.com/razor-view-engine/>
5. <https://birimler.dpu.edu.tr/app/views/panel/ckfinder/userfiles/95/files/TezYazi.pdf>
6. <https://www.niobehosting.com/blog/mysql/>
7. <https://www.mshowto.org/web-servis-nedir-nereelerde-kullanilir.html>

## **ÖZGEÇMİŞ**

Eren YARDIM 2001'de Samsun'da doğdu; ilk ve orta öğrenimini aynı şehirde tamamladı; Canik İMKB Anadolu Lisesi'nden mezun olduktan sonra 2019 yılında Kastamonu Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne girdi.

### **ADRES BİLGİLERİ**

Adres: Çay Mahallesi

2054.Sokak, No. 10, D: 10

Tekkeköy / SAMSUN

Tel: (543) 6010438

E-posta: erenym3@gmail.com