

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

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI

AİLE HEKİMİ HASTA TAKİP SİSTEMİ

G191210077 – Hüseyin Burhan BAŞARAN

Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ
Tez Danışmanı : Prof.(Doç.) Dr. Nejat YUMUŞAK

2022 – 2023 Güz Dönemi

ÖNSÖZ

Günümüzde teknolojinin gelişmesi daha önce insanların yaptığı işleri kolaylaştırmış ya da yerine geçmiştir. Bunun yanında yeni iş alanlar oluşturduğu gibi üretimde çeşitliliği arttırmıştır. Hastaneler de bu gelişmeden etkilenmiş ve teknolojiyi lehine kullanmışlardır.

Bu projeyi kendisiyle yapmama fırsat veren değerli danışman hocam ve dekanım Prof. Dr. Nejat YUMUŞAK hocama teşekkürlerimi iletilirim.

İÇİNDEKİLER

ÖNSÖZ.....	ii
İÇİNDEKİLER.....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	v
ŞEKİLLER LİSTESİ.....	vi
ÖZET.....	vii

BÖLÜM 1.

GİRİŞ	1
1.1. Projenin Amacı	1

BÖLÜM 2.

PROJE SÜRECİ VE KULLANILAN TEKNOLOJİLER	2
2.1. Projenin Süreci	2
2.2. Projede Kullanılan Teknolojiler	2
2.2.1. MSSQL	2
2.2.2. .NET Core	3
2.2.2.1. Modeller	3
2.2.2.2. Katmanlı Mimari	3
2.2.2.3. CRUD İşlemleri	6
2.2.2.4. Authorization ve Authentication İşlemleri	7
2.2.2.5. API işlemleri	8
2.2.3 IIS	15
2.2.4. React.....	15
2.2.5. React Native	16

BÖLÜM 3.

SONUÇLAR VE ÖNERİLER.....	12
---------------------------	----

KAYNAKLAR.....	13
----------------	----

ÖZGEÇMİŞ	14
----------------	----

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI	17
---	----

SİMGELER VE KISALTMALAR LİSTESİ

SQL	: Structured Query Language
MSSQL	: Microsoft SQL Server
CRUD	: CREATE, READ, UPDATE and DELETE
IIS	: Internet Information Services
API	: Application Programming Interface
JSON	: JavaScript Object Notation
UI	: User Interface

ŞEKİLLER LİSTESİ

Şekil 2.1.	İlişkisel Veritabanı Modeli	2
Şekil 2.2.	Proje Katmanları	3
Şekil 2.3.	Aile Hekimi Modeli	4
Şekil 2.4.	Hasta Modeli	4
Şekil 2.5.	İlaç Modeli	4
Şekil 2.6.	Hasta İlaç Modeli	5
Şekil 2.7.	İlaç Saati Modeli	5
Şekil 2.8.	Kullanıcı Bilgi Modeli	5
Şekil 2.9.	Kullanıcı Şifre Modeli	6
Şekil 2.10.	Kullanıcı İşlem Talep Modeli	6
Şekil 2.11.	İşlem Talep Modeli	6
Şekil 2.12.	Ekle, Sil ve Getir Metotları	7
Şekil 2.13.	Liste getir ve Güncelle Metotları	7
Şekil 2.14.	Giriş yapmamış ise hata döndür	8
Şekil 2.15.	Yetkisi Doktor olanlar girebilir	8
Şekil 2.16.	Kullanıcı API	9
Şekil 2.17.	Giriş JSON	9
Şekil 2.18.	Kayıt Ol JSON	9
Şekil 2.19.	Döndürülen cevap (token)	9
Şekil 2.20	Postman’de başlığa eklenmiş token	9
Şekil 2.21.	Aile Hekimi API	10
Şekil 2.22.	Kullanıcı Bilgileri JSON	10
Şekil 2.23.	Aile Hekimi JSON	10
Şekil 2.24.	Hasta API	11
Şekil 2.25.	Hasta JSON	11
Şekil 2.26.	Hasta İlaç API	12
Şekil 2.27.	Hasta İlaç JSON	12
Şekil 2.28.	Hasta İlaç cevap JSON	13
Şekil 2.29.	İlaç API	14
Şekil 2.30.	İlaç JSON	14

Şekil 2.31.	İlaç cevap JSON	14
Şekil 2.32.	İlaç Saati API	15
Şekil 2.33.	İlaç Saati JSON	15
Şekil 2.34.	getById metodu cevap JSON	16
Şekil 2.35.	getByIlacId metodu cevap JSON	16
Şekil 3.1.	Aile Hekimi Kayıt Formu	17
Şekil 3.2.	Aile Hekimi Giriş Formu	18
Şekil 3.3.	Aile Hekimi Profil Sayfası	18
Şekil 3.4.	Aile Hekimi'nin hastalarım sayfası	19
Şekil 3.5.	Hastalarım Sayfasındaki hastanın ilaçları	19
Şekil 3.6.	Hasta Ekle Formu	20
Şekil 3.7.	Hasta başarıyla eklendi cevabı	20
Şekil 3.8.	Hastayı kaldır formu	20
Şekil 3.9.	Hasta başarıyla silindi cevabı	21
Şekil 3.10.	Hastaya ilaç ekleme formu	21
Şekil 3.11.	Hastaya ilaç ekleme formundaki ilaç listesi	21
Şekil 3.12.	Hastaya ilaç başarıyla eklendi cevabı	22
Şekil 3.13.	Hastanın seçili ilacını düzenleme formu	22
Şekil 3.14.	Hastanın seçili ilacı başarıyla güncellendi cevabı	22
Şekil 3.15.	Hastanın seçili ilacını sil formu	23
Şekil 3.16.	Hastanın seçili ilacı başarıyla silindi cevabı	23
Şekil 4.1.	İlaç listesi	24
Şekil 4.2.	İlaç detayları	24
Şekil 4.3.	İlaç ekle formu	25
Şekil 4.4.	İlaç düzenle formu	25
Şekil 4.5.	İlaç sil formu	26
Şekil 4.6.	İlaç saati ekle formu	26
Şekil 4.7.	İlaç saati düzenle formu	27
Şekil 4.8.	İlaç saati sil formu	27

Şekil 5.1.	Mobil uygulama giriş sayfası	28
Şekil 5.2.	Mobil uygulama kayıt sayfası	28
Şekil 5.3.	Mobil uygulama anasayfası	29
Şekil 5.4.	Mobil uygulama menü	29
Şekil 5.5.	Mobil uygulama geçmiş tarih	30
Şekil 5.6.	Mobil uygulama gelecek tarih	30
Şekil 5.7.	Mobil uygulama bildirim	31

ÖZET

Anahtar kelimeler: Hasta, Aile Hekimi, .Net Core, JSON, React, React Native, API

Her geçen gün ilaç sayısı, çeşitliliği ve ilaç kullanan insan sayısı artmaktadır. Bunun kontrol edilmesi ve listelenmesi de gün geçtikçe zorlaşmaktadır. İnsanların kullanması gereken ilaçların adet ve saat takibi de zorlaşmaktadır.

Aile hekiminin kendi hastası için vereceği ilaçları seçmesi, yeni ilaçlar ekleyebilmesi sayesinde hastanın kullanması gereken ilaçlar hastaya tanımlanmış olur.

Hasta'nın mobil cihazında günlük alması gereken ilaçlar listelenir. Bu ilaçların alınması gereken adetini, saatini ve kalan süreyi görebilir. Hasta'nın ilaç alması gerektiğini hatırlatmaktadır.

BÖLÜM 1. GİRİŞ

Sağlık sektörü dünya üzerindeki en büyük sektörlerden biridir. Bu sektör insanların hayatı ve yaşamı için çok önemli ve gereklidir. İnsan sağlığı için bu sektörün gelişmesi çok büyük önem arz eder.

1.1. Projenin Amacı

Günümüzde insanların ilaç ve sağlık için kullanılan takviyeler hızla artmaktadır. Hastaların ihtiyaçları doğrultusunda aile hekiminin hasta için tanımladığı ilaçları bu uygulama sayesinde kullanım süresini, ne zaman, ne kadar ve nasıl kullanması gerektiğini rahatlıkla görüntüleyebilir. Bu sayede hastanın ilaç alması gerektiğini unutmaması ve hastanın hayatını kolaylaştırması amaçlanmaktadır.

BÖLÜM 2. PROJE SÜRECİ VE KULLANILAN TEKNOLOJİLER

Sağlık sektörü dünya üzerindeki en büyük sektörlerden biridir. Bu sektör insanların hayatı ve yaşamı için çok önemli ve gereklidir. İnsan sağlığı için bu sektörün gelişmesi çok büyük önem arz eder.

2.1. Projenin Süreci

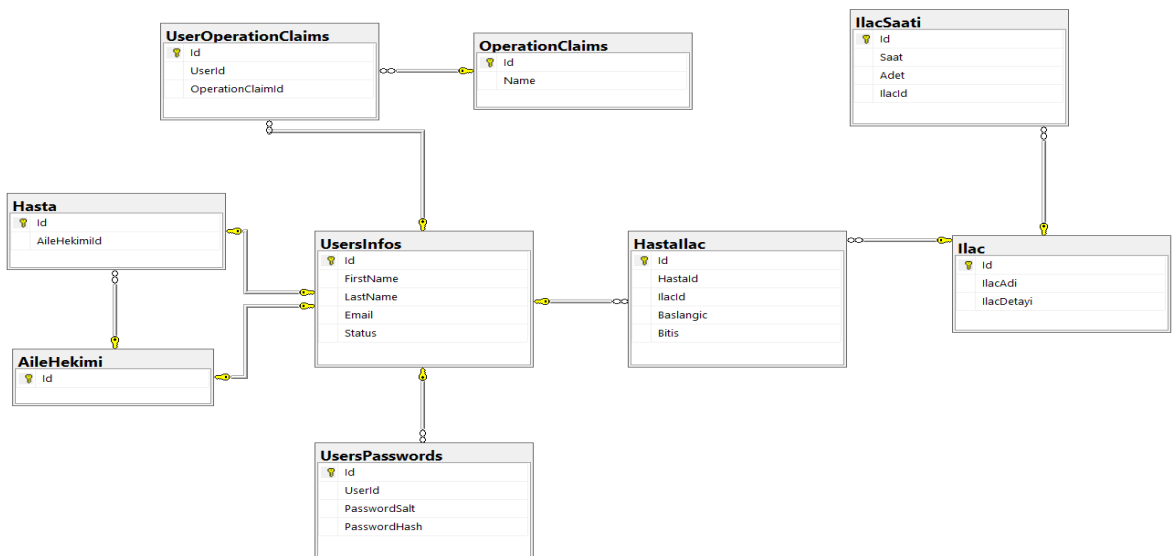
Proje veritabanına bağlı bir server ve bu serverden gelen verileri görsel olarak bize sunan bir web sayfası ve bir mobil uygulamaya sahiptir.

2.2. Projede Kullanılan Projeler

Projede veritabanı için MSSQL, server için .Net Core, web için React, mobil için React Native kullanıldı.

2.2.1. MSSQL

MSSQL, Microsoft'un veritabanı sağlayıcısıdır. Burada veritabanı tabloları ve tablolar arasındaki ilişkiler bulunur.



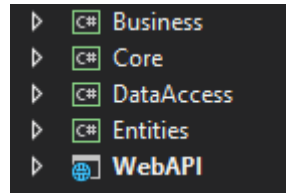
Şekil 2.1. İlişkisel Veritabanı Modeli

2.2.2. .NET Core

.Net Core projenin server yani backend kısmıdır. Bu bölümde veritabanı sorgu, ekleme, düzenleme ve silme işlemleri belirli is kuralları çerçevesinde yapılır.

2.2.2.1. Katmanlı Mimari

Bu proje dört katmandan oluşur.



Şekil 2.2. Proje Katmanları

İlk katmanda varlıklar(nesneler) bulunur bu katmana Entities katmanı denir.

İkinci katmanda ise bütün projelerdeki ortak modeller ve metotlar bulunur bu katmana da Core katmanı denir.

Üçüncü katmanda ise Bu modeller kullanılarak veritabanından veri getirme, ekleme, silme ve değiştirme işlemleri yapılır. Bu katmana Data Access katmanı denir.

Dördüncü katman ise iş kurallarının yazıldığı bu işlemler ve doğrulamalar sonucunda data access katmanından ekleme işlemlerinin yapıldığı katmandır. Son olarak API proje bulunur. Bu proje Business katmanını çağırarak kullanıcıya sunar.

2.2.2.2. Modeller

Modeller veritabanında bulunan tabloların server kısmında birer kalıcı nesne olarak adlandırılmasına denir. Her bir nesnenin veritabanında birer karşılığı vardır. Bu nesnelerin bulunduğu katmana Entities(varlıklar) katmanı denir.

Aile Hekimi varlığı hasta listesine sahiptir.

```
public class AileHekimi : IEntity
{
    public int Id { get; set; }
    public List<Hasta> Hastalar { get; set; }
}
```

Şekil 2.3. Aile Hekimi Modeli

Hasta Aile hekimine sahiptir. Bu da Aile Hekimi ve Hasta arasında bire çok bir ilişki sağlar. Hasta ayrıca Hasta İlaçları listesine sahiptir.

```
public class Hasta : IEntity
{
    public int Id { get; set; }
    public int AileHekimiId { get; set; }
    public List<HastaIlac> HastaIlacLari { get; set; }
}
```

Şekil 2.4. Hasta Modeli

İlac varlığı İlac Saatleri listesine sahiptir. İlaç adı ve Detayı olası bir fazla boşluk girmeyi engellemek için Trim metodudunu kullanır.

```
public class Ilac:IEntity
{
    public int Id { get; set; }
    public string IlacAdi { set { ad = value.Trim(); } get { return ad; } }
    public string IlacDetayi { set { detay = value.Trim(); } get { return detay; } }

    public List<IlacSaati> IlacSaatleri { get; set; }

    private string ad;
    private string detay;
}
```

Şekil 2.5. İlaç Modeli

Hasta İlac varlığı Hasta id ve İlac id'ne sahiptir. Bu varlık Hasta ve İlaç arasında çokla çok ilişki sağlayan ara bir varlıktır. Aynı zamanda Hasta'nın Kullanacağı ilacın başlangıç ve bitiş tarihlerini de tutar.

```

public class HastaIlac : IEntity
{
    public int Id { get; set; }

    public int HastaId { get; set; }

    public int IlacId { get; set; }
    public Ilac Ilac { get; set; }

    public DateTime Baslangic { get; set; }
    public DateTime Bitis { get; set; }
}

```

Şekil 2.6. Hasta İlaç Modeli

İlaç Saatleri varlığı ilacın saat ve adet ve id'sini tutar bu da İlaç ile arasında bire çok ilişki sağlar.

```

public class IlacSaati : IEntity
{
    public int Id { get; set; }
    public string Saat { get; set; }
    public int Adet { get; set; }
    public int IlacId { get; set; }
}

```

Şekil 2.7. İlaç Saati Modeli

User Info varlığı kullanıcının adını, soyadını, e-mail ve durum bilgilerini tutar.

```

public class UserInfo : IEntity
{
    public int Id { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }
    public bool Status { get; set; }
}

```

Şekil 2.8. Kullanıcı Bilgi Modeli

Kullanıcının şifre bilgileri veritabanında şifrelenmiş şekilde tutulur. Bu varlıkta şifrele şifreli ve şifresiz bir şekilde tutulur.

```
public class UserPassword : IEntity
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public byte[] PasswordSalt { get; set; }
    public byte[] PasswordHash { get; set; }
}
```

Şekil 2.9. Kullanıcı Şifre Modeli

Bu varlıkta her kullanıcının kullanıcı türü tutulur. Örnek olarak: admin, hasta, aile hekimi. Her kullanıcının birden fazla türü olabilir.

```
public class UserOperationClaim:IEntity
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public int OperationClaimId { get; set; }
}
```

Şekil 2.10. Kullanıcı İşlem Talep Modeli

Bu varlıkta ise o türler tutulur.

```
public class OperationClaim
{
    public int Id { get; set; }
    public string Name { get; set; }
}
```

Şekil 2.11. İşlem Talep Modeli

2.2.2.3. CRUD İşlemleri

CRUD (CREATE, READ, UPDATE and DELETE) veritabanı işlemleri Data Access katmanında yapılır.

```

public void Add(TEntity entity)
{
    using (var context=new TContext())
    {
        var addedEntity = context.Entry(entity);
        addedEntity.State = EntityState.Added;
        context.SaveChanges();
    }
}

public void Delete(TEntity entity)
{
    using (var context = new TContext())
    {
        var deletedEntity = context.Entry(entity);
        deletedEntity.State = EntityState.Deleted;
        context.SaveChanges();
    }
}

public TEntity Get(Expression<Func<TEntity, bool>> filter)
{
    using (var context = new TContext())
    {
        return context.Set<TEntity>().SingleOrDefault(filter);
    }
}

```

Şekil 2.12. Ekle, Sil ve Getir Metotları

```

public IList<TEntity> GetList(Expression<Func<TEntity, bool>> filter = null)
{
    using (var context = new TContext())
    {
        return filter == null
            ? context.Set<TEntity>().ToList()
            : context.Set<TEntity>().Where(filter).ToList();
    }
}

public void Update(TEntity entity)
{
    using (var context = new TContext())
    {
        var updatedEntity = context.Entry(entity);
        updatedEntity.State = EntityState.Modified;
        context.SaveChanges();
    }
}

```

Şekil 2.13. Liste getir ve Güncelle Metotları

Diğer sınıflar Generic şekilde oluşturulan bu sınıftan kalıtım alarak gönderilen valık üzerinde işlemler yapar. Get ve GetList metotlarında parameter olarak sorgu ve filtre gönderilebilir bu sayede veriler istenilen şekilde veritabanından çekilir.

2.2.2.4. Authorization ve Authentication İşlemleri

Authentication Türkçesi ile kimlik doğrulama, kullanıcıların giriş yapıp yapmadıklarını kontrol eden sistemdir. Giriş yapmış kullanıcının ID'sini getirir.

```
var id = _curentUserService.UserId();
if (id == null)
    return BadRequest("Not Authenticated");
```

Şekil 2.14. Giriş yapmamış ise hata döndür

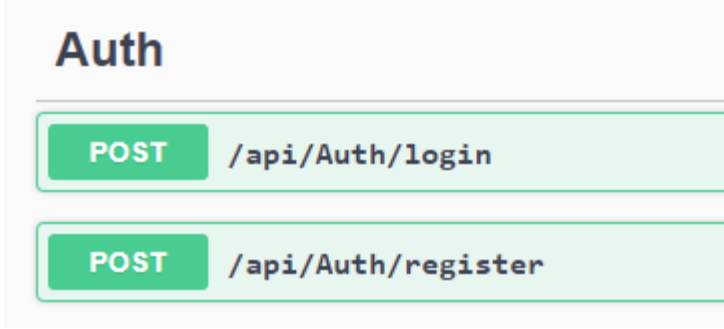
Authorization ise yetkilendirme kontorlu yapan sistemdir. Bu sistemde giriş yapmış kullanıcının istenen veriyi okuma veya yazma izni olup olmadığını control eder ve eğer kullanıcı yetkili ise işlemi gerçekleştirilir. Eğer yetkili değil ise kullanıcıya hata mesajı gönderilir. Giriş yapmış kullancının rolü doktor ise kod çalışır. Giriş yapmış kullanıcının rolü ve kim olduğu Sunucuya gönderilen isteklerin başlık kısmında bulunan token den kontrol edilir.

```
[Authorize(Roles = "doktor")]
```

Şekil 2.15. Yetkisi Doktor olanlar girebilir.

2.2.2.5. API İşlemleri

API iki uygulamanın birbiri ile haberleşmesini sağlayan uygulama programlama arayüzüdür. Bu projede sunucu kısmına mobil veya web üzerinden JSON şeklinde gönderilen istekler cevaplanır.



Şekil 2.16. Kullanıcı API

Giriş ve kayıt işlemi için sunucuya form bilgileri POST edilir. Eğer iş kuralları ve kontroller sağlanmaz ise hata cevapları döndürülür.

```
{
  "email": "string",
  "password": "string"
}
```

Şekil 2.17. Giriş JSON

```
{
  "email": "string",
  "password": "string",
  "firstName": "string",
  "lastName": "string"
}
```

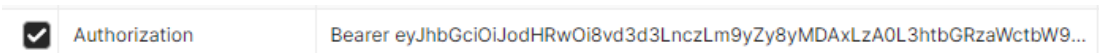
Şekil 2.18. Kayıt Ol JSON

Kullanıcı girişi sağlanır ise kullanıcı için özel oluşturulan token üretilir ve bununla birlikte tokenin kullanım dışı olacağı tarih cevap olarak gönderilir.

```
{
  "token": "eyJhbGciOiJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbW9yZSNoYWFiZXNoYTI1NiIjLnRpMlliclI6IjIiCjBWFpbCI6ImFAY55hIiwiaHR0cDovL3NjaGVtYXMuZW50c2VucmVud3MvbmJhNS8wNS9pL4LzA2L2lkZW50aXR5L2NsYWltcy9yb2x1IjoideXNlcisIm5iziI6MTY3MTU1ODkyMCwiZXhwIjoxNjc5NDIyOTIwLCJPI",
  "expiration": "2022-12-30T20:55:20.3899291+03:00"
}
```

Şekil 2.19. Döndürülen cevap (token)

Token başlık kısmına eklenir. Bu sayede kullanıcı giriş yapmış olur.



Şekil 2.20. Postman’de başlığa eklenmiş token

Aile hekimi ve Hastanın ID değerleri User Infosun birincil anahtarının, yabancı anahtarıdır. Yani Aile Hekimi ve Hasta Kullanıcı tablosunda olmak zorundadır.

AileHekimi	
GET	/api/AileHekimi/getloggedInAileHekimi
POST	/api/AileHekimi/add
POST	/api/AileHekimi/update
POST	/api/AileHekimi/delete

Şekil 2.21. Aile Hekimi API

getLoggedInAileHekimi giriş yapmış aile hekiminin bilgilerini cevap olarak gönderir.

```
{
  "id": 5,
  "firstName": "Burhan",
  "lastName": "Basaran",
  "email": "asd@asd.asd",
  "status": true
}
```

Şekil 2.22. Kullanıcı bilgileri JSON

Ekle, güncelle ve sil API'larına aile hekiminin ID si POST edilir ve istek gerçekleşir.

```
{
  "id": 0
}
```

Şekil 2.23. Aile Hekimi JSON

getLoggedInUser giriş yapmış hastanın bilgilerini cevap olarak gönderir.

Gethastainfolist giriş yapmış aile hekiminin hastalarının bilgilerini liste olarak gönderir.

Hasta	
GET	/api/Hasta/getLoggedInUser
GET	/api/Hasta/gethastainfolist
GET	/api/Hasta/GetHastaWithoutAileHekimi
POST	/api/Hasta/add
POST	/api/Hasta/update
POST	/api/Hasta/delete

Şekil 2.24. Hasta API

Hasta ekle, güncelle ve sil metotları için hasta id'si ve aile hekimi id'si POST edilir.

```
{
  "id": 0,
  "aileHekimiId": 0
}
```

Şekil 2.25. Hasta JSON

Hastallac	
GET	/api/HastaIlac/getallbyhastaid
GET	/api/HastaIlac/getall
POST	/api/HastaIlac/add
POST	/api/HastaIlac/update
POST	/api/HastaIlac/delete

Şekil 2.26. Hasta İlaç API

Hastaİlaç ekle, güncelle ve sil metotları için ilacı kullanacak hastanın id'si hastanın kullanacağı ilacın id'si, başlangıç ve bitiş tarihleri POST edilir.

```
{
  "id": 0,
  "hastaId": 0,
  "ilacId": 0,
  "baslangic": "2022-12-21T08:48:16.720Z",
  "bitis": "2022-12-21T08:48:16.720Z"
}
```

Şekil 2.27. Hasta İlaç JSON

getAll metodu giriş yapmış kullanıcının getAllById ise Id'si parametre olarak gönderilen kullanıcının, ilaçları ve ilaç saatleri cevap olarak gönderilir.

```
[
  {
    "id": 5,
    "hastaId": 2,
    "ilacId": 2,
    "ilac": {
      "id": 2,
      "ilacAdi": "Asist Plus",
      "ilacDetayi": "Balgam Sökücü",
      "ilacSaatleri": [
        {
          "id": 4,
          "saat": "11:00:00",
          "adet": 5,
          "ilacId": 2
        },
        {
          "id": 6,
          "saat": "10:00:00",
          "adet": 1,
          "ilacId": 2
        }
      ]
    }
  },
  "baslangic": "2022-11-13T00:00:00",
  "bitis": "2022-12-18T00:00:00"
},
{
  "id": 10,
  "hastaId": 2,
  "ilacId": 1,
  "ilac": {
    "id": 1,
    "ilacAdi": "Parol",
    "ilacDetayi": "Agri Kesici",
    "ilacSaatleri": [
      {
        "id": 2,
        "saat": "14:00:00",
        "adet": 1,
        "ilacId": 1
      },
      {
        "id": 9,
        "saat": "10:00:00",
        "adet": 2,
        "ilacId": 1
      }
    ]
  }
},
"baslangic": "2022-12-08T00:00:00",
"bitis": "2022-12-17T00:00:00"
}
]
```

Şekil 2.28. Hasta İlaç cevap JSON

İlac	
GET	/api/Ilac/getbyid
GET	/api/Ilac/gettall
POST	/api/Ilac/add
POST	/api/Ilac/update
POST	/api/Ilac/delete

Şekil 2.29. İlaç API

İlaç ekle, güncelle ve sil metotları için ilacın adı ve detayı POST edilir.

```
{
  "id": 0,
  "ilacAdi": "string",
  "ilacDetayi": "string"
}
```

Şekil 2.30. İlaç JSON

getbyId metodu parametre olarak gönderilen ilacın id'si ile ilacın detaylarının cevap olarak gönderir. GetAll metodu bütün ilaçları liste olarak gönderir.

```
{
  "id": 1,
  "ilacAdi": "Parol",
  "ilacDetayi": "Agri Kesici",
  "ilacSaatleri": [
    {
      "id": 2,
      "saat": "14:00:00",
      "adet": 1,
      "ilacId": 1
    },
    {
      "id": 9,
      "saat": "10:00:00",
      "adet": 2,
      "ilacId": 1
    }
  ]
}
```

Şekil 2.31. İlaç cevap JSON

İlacSaati	
GET	/api/İlacSaati/getbyid
GET	/api/İlacSaati/getbyilacid
POST	/api/İlacSaati/add
POST	/api/İlacSaati/delete
POST	/api/İlacSaati/update
POST	/api/İlacSaati/deletebyilacid

Şekil 2.32. İlaç Saati API

İlaç Saati ekle, güncelle ve sil metotları için ilacın kullanılacağı saat ve adet ile birlikte ilacın id'si POST edilir.

```
{  
  "id": 0,  
  "saat": "string",  
  "adet": 0,  
  "ilacId": 0  
}
```

Şekil 2.33. İlaç Saati JSON

deleteByIlacId metodu gönderilen ilacın bütün ilaç saatlerini temizler.

GetById ilaç saati id inden saati geri döndürür.


```
{
  "id": 2,
  "saat": "14:00:00",
  "adet": 1,
  "ilacId": 1
}
```

Şekil 2.34. getById metodu cevap JSON

getByIlacId metodu girilen ilacın id'sinden o ilacın saatlerini geri döndürür.

```
[
  {
    "id": 4,
    "saat": "11:00:00",
    "adet": 5,
    "ilacId": 2
  },
  {
    "id": 6,
    "saat": "10:00:00",
    "adet": 1,
    "ilacId": 2
  }
]
```

Şekil 2.35. getByIlacId metodu cevap JSON

2.2.3. IIS

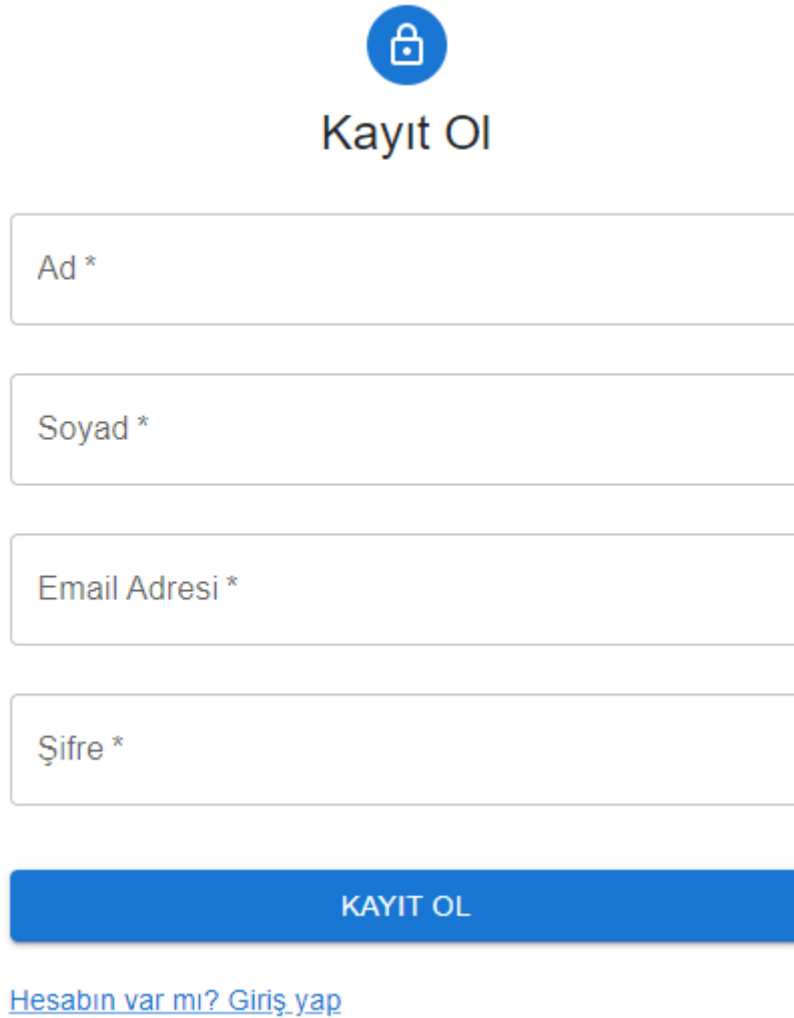
IIS (Internet Information Services) Microsoft tarafından Windows sistemlerde çalışmak üzere geliştirilmiş bir web sunucusudur. Projenin server kısmı IIS ile paylaşılır bu sayede cihazın IP adresi ile diğer cihazlar bu servere erişim sağlar.

2.2.4. React


React bir JavaScript kütüphanesidir. Facebook önderliğinde geliştirilen, açık kaynak kodlu, değişimleri anında görebildiğimiz UI (kullanıcı arayüzü) oluşturmaya yarayan bir teknoloji olduğu için günümüzde çok tercih edilmektedir.

Bu projenin web kısmı yani Aile Hekimi Yönetim Sistemi React ile gerçekleştirildi.

Proje Giriş ve Kayıt Ol sayfası ile açılmaktadır.



Kayıt Ol formu, kullanıcıların hesap oluşturmasını sağlayan bir arayüzdür. Formun başında, kullanıcıyı yönlendiren bir ikon ve başlık bulunmaktadır. İkon, bir kilit simgesiyle temsil edilir. Başlık, 'Kayıt Ol' olarak belirtilmiştir. Formun içeriği, kullanıcı bilgilerini girmek için dört adet metin alanı ve bir kayıt butonu içerir. Alanlar, 'Ad *', 'Soyad *', 'Email Adresi *' ve 'Şifre *' olarak etiketlenmiştir. Buton, 'KAYIT OL' yazısıyla belirtilmiştir. Formun altına, 'Hesabın var mı? Giriş yap' bağlantısı yerleştirilmiştir.


Kayıt Ol

Ad *

Soyad *

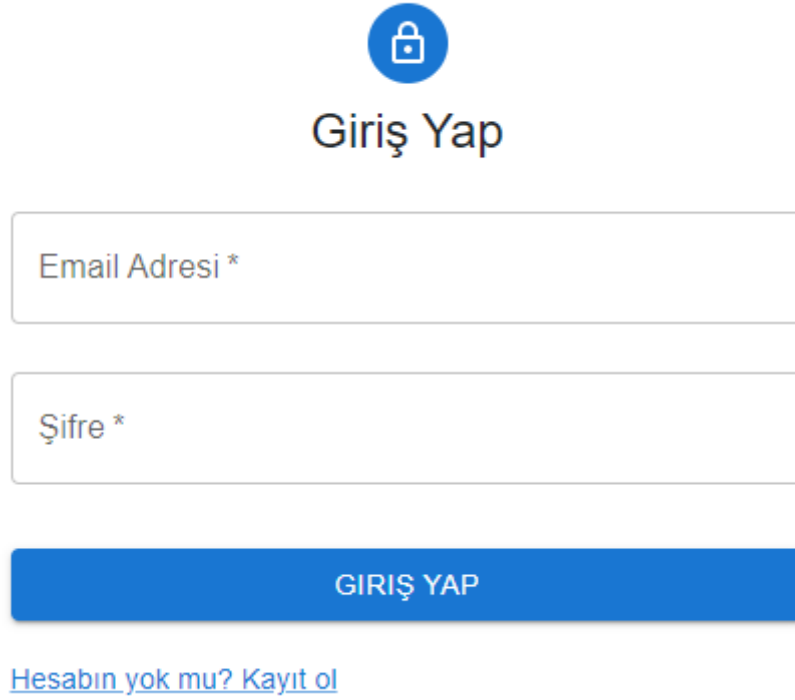
Email Adresi *


Şifre *

KAYIT OL

[Hesabın var mı? Giriş yap](#)

Şekil 3.1. Aile Hekimi Kayıt Formu



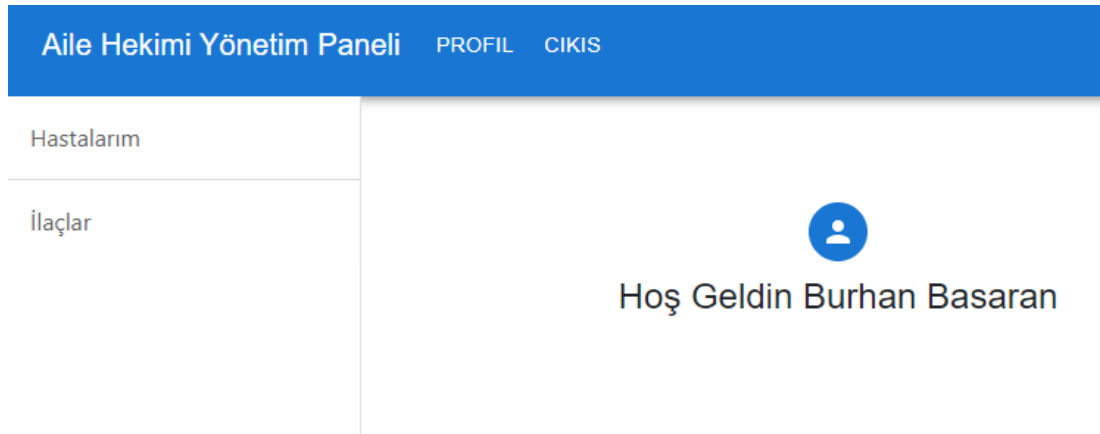


Giriş Yap

[Hesabın yok mu? Kayıt ol](#)

Şekil 3.2. Aile Hekimi Giriş Formu


Aile Hekimleri e-mail ve şifreleri ile burada giriş yaparlar. Authorization ve Authentication işlemleri sonucunda yönetim paneline yönlendirilirler.



Aile Hekimi Yönetim Paneli
PROFIL
ÇIKIŞ

Hastalarım

İlaçlar



Hoş Geldin Burhan Basaran

Şekil 3.3. Aile Hekimi Profil Sayfası

Yönetim Panelinde giriş yapan aile hekiminin hastaları ve ilaçlar sayfası solda yer alır.

HASTA EKLE				
Hasta İlaçları	Ad Soyad	Email	Durum	İşlem
▼ 1	Burhan2 Basaran	assd	true	KALDIR
▼ 2	Hüseyin Başaran	a@a.a	true	KALDIR
▼ 4	asds asdd	as@as.asd	true	KALDIR

Şekil 3.4. Aile Hekimi'nin hastalarım sayfası

Hastalarım sayfasında Aile hekiminin hastalarının bilgileri, yeni hasta ekle ve kaldır butonları bulunur. Hastaların detayları için soldaki ok butonu ile detayları listelenir.

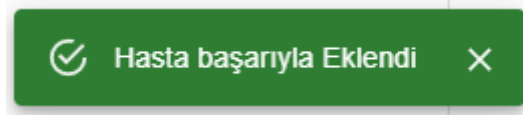
^ 2	Hüseyin Başaran	a@a.a	true	KALDIR																									
<div>Hasta İlaçları İLAÇ EKLE</div> <table> <tr> <th>İlaç Adı</th><th>Başlangıç</th><th>Bitiş</th><th colspan="2">İşlem</th></tr> <tr> <td>Arveles</td><td>06/11/2022</td><td>09/12/2022</td><td>DÜZENLE</td><td>SIL</td></tr> <tr> <td>Asist Plus</td><td>13/11/2022</td><td>18/12/2022</td><td>DÜZENLE</td><td>SIL</td></tr> <tr> <td>Parol</td><td>08/12/2022</td><td>17/12/2022</td><td>DÜZENLE</td><td>SIL</td></tr> <tr> <td>Majezik</td><td>10/12/2022</td><td>10/12/2022</td><td>DÜZENLE</td><td>SIL</td></tr> </table>					İlaç Adı	Başlangıç	Bitiş	İşlem		Arveles	06/11/2022	09/12/2022	DÜZENLE	SIL	Asist Plus	13/11/2022	18/12/2022	DÜZENLE	SIL	Parol	08/12/2022	17/12/2022	DÜZENLE	SIL	Majezik	10/12/2022	10/12/2022	DÜZENLE	SIL
İlaç Adı	Başlangıç	Bitiş	İşlem																										
Arveles	06/11/2022	09/12/2022	DÜZENLE	SIL																									
Asist Plus	13/11/2022	18/12/2022	DÜZENLE	SIL																									
Parol	08/12/2022	17/12/2022	DÜZENLE	SIL																									
Majezik	10/12/2022	10/12/2022	DÜZENLE	SIL																									
▼ 4	asds asdd	as@as.asd	true	KALDIR																									

Şekil 3.5. Hastalarım Sayfasındaki hastanın ilaçları

Burada hastanın kullandığı ilaçlar ve kullanması gereken tarih aralığı listelenir. Bu hastaya yeni ilaç eklenebileceği gibi var olan ilaç düzenlenebilir ya da silinebilir.

Şekil 3.6. Hasta Ekle Formu

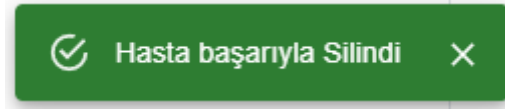
Hasta ekle butonu ile aile hekimi olmayan hastalar listesinden bir hasta seçip bu hastanın yeni aile hekimi olması sağlanır. Eğer bu işlem başarılı ise ekrana bildirim gelir.



Şekil 3.7. Hasta başarıyla eklendi cevabı

Hasta kaldır butonu ile seçili hastanın aile hekimliğinden çıkılır ve başarılı ise bildirim gönderilir.

Şekil 3.8. Hastayı kaldır formu



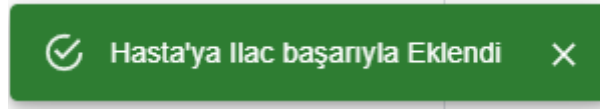
Şekil 3.9. Hasta başarıyla silindi cevabı

Bir hasta seçilip bu hasta için ilaç ekle butonuna basılınca bizeden elenecek ilacı seçmemizi ve tarih aralığını girmemiz istenir.

Şekil 3.10. Hastaya ilaç ekleme formu

Şekil 3.11. Hastaya ilaç ekleme formundaki ilaç listesi

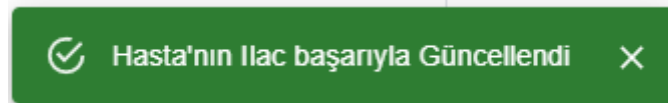
İşlem başarılı gerçekleşirse kullanıcıya bildirilir.



Şekil 3.12. Hastaya ilaç başarıyla eklendi cevabı

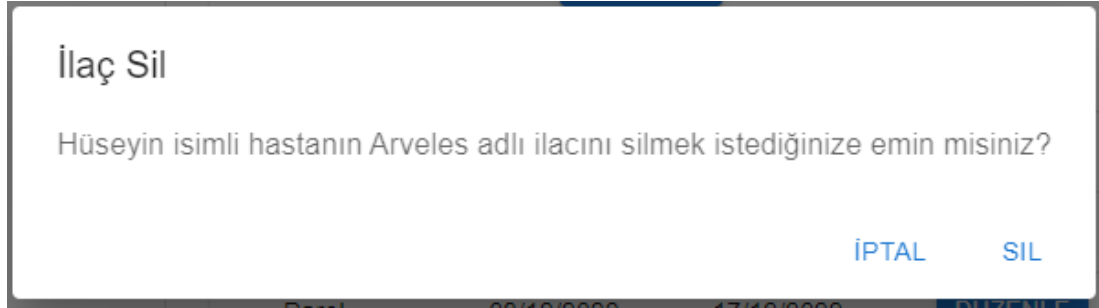
İlaç düzenleme butonu ile seçili hastanın ilaç bilgileri değiştirilip kaydedilir işlem başarılı ise kullanıcıya bildirilir.

Şekil 3.13. Hastanın seçili ilacını düzenleme formu

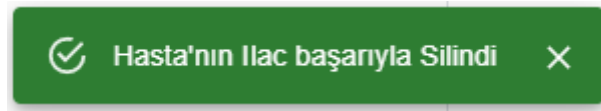


Şekil 3.14. Hastanın seçili ilacı başarıyla güncellendi cevabı

Hastanın var olan ilacı silinmek istenir ise sil butonu ile bu işlem gerçekleştirilebilir ve işlem sonucu kullanıcıya bildirilir.



Şekil 3.15. Hastanın seçili ilacını sil formu



Şekil 3.16. Hastanın seçili ilacı başarıyla silindi cevabı

İlaç sayfasında bütün ilaçlar adları ve detayları ile listelenir. Yeni ilaç ekle düzenle ve sil butonları bulunur.

İLAÇ EKLE			
İlac Saatleri	İlac Adı	İlac Detayı	İşlem
▼ 1	Parol	Ağrı Kesici	DÜZENLE SIL
▼ 2	Asist Plus	Balgam Sökücü	DÜZENLE SIL
▼ 3	Arveles	Ağrı Kesici	DÜZENLE SIL
▼ 30	Majezik	Ağrı Kesici	DÜZENLE SIL

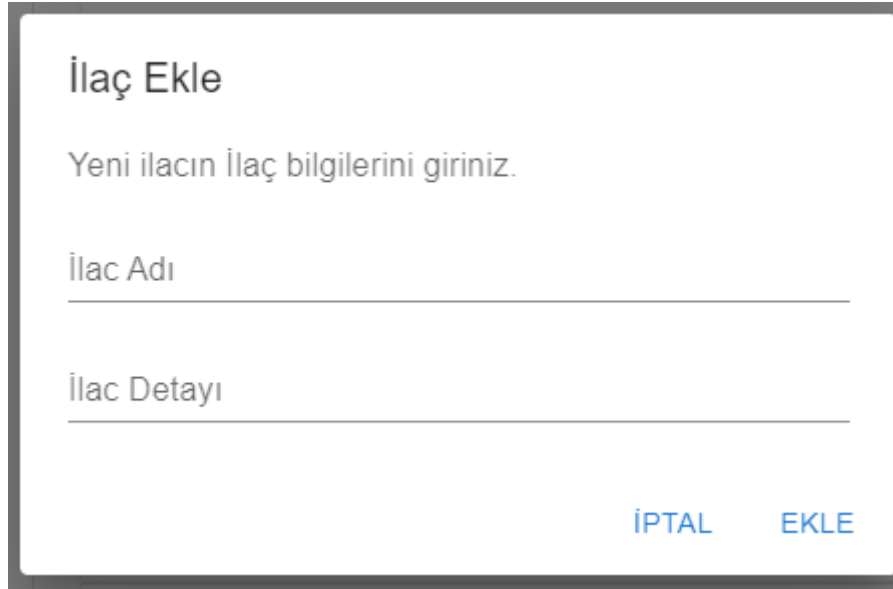
Şekil 4.1. İlaç listesi

İlaçların detayları için soldaki ok butonu ile detayları listelenir. Burada ilaçların kullanılması gereken saat ve adet bilgileri yer alır. Bu ilaç için yeni saat ekle, düzenle ve sil butonları bulunur.

^ 1	Parol	Ağrı Kesici	DÜZENLE	SIL
<div>İlac Saatleri</div> <div>İLAC SAATI EKLE</div>				
Saat	Adet	İşlem		
14:00:00	1	DÜZENLE	SIL	
10:00:00	2	DÜZENLE	SIL	

Şekil 4.2. İlaç detayları

İlaç ekle butonu ile eklenecek ilacın adı detay bilgileri istenen form doldurulur.



İlaç Ekle

Yeni ilacın İlaç bilgilerini giriniz.

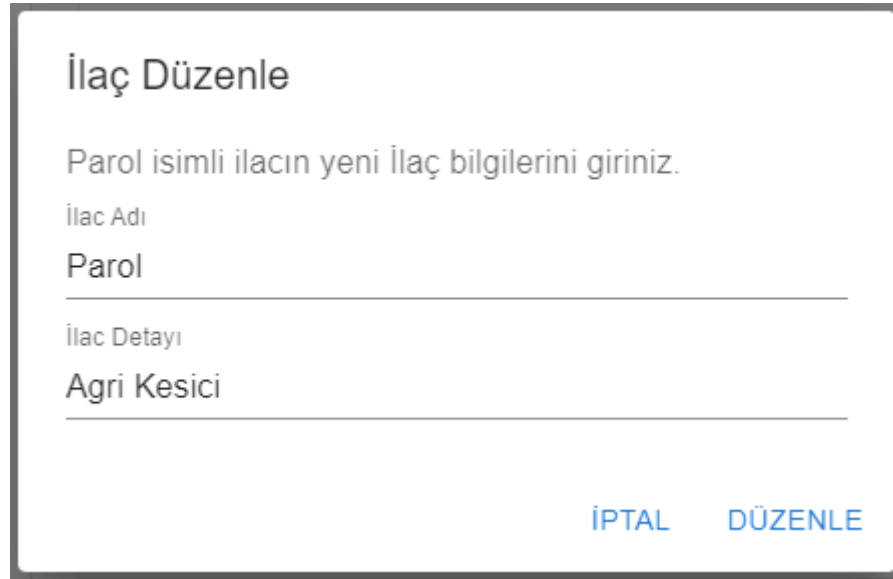
İlac Adı

İlac Detayı

İPTAL EKLE

Şekil 4.3. İlaç ekle formu

Düzenlenecek ilacın eski bilgiler yeni bilgiler ile değiştirilir.



İlaç Düzenle

Parol isimli ilacın yeni İlaç bilgilerini giriniz.

İlac Adı

Parol

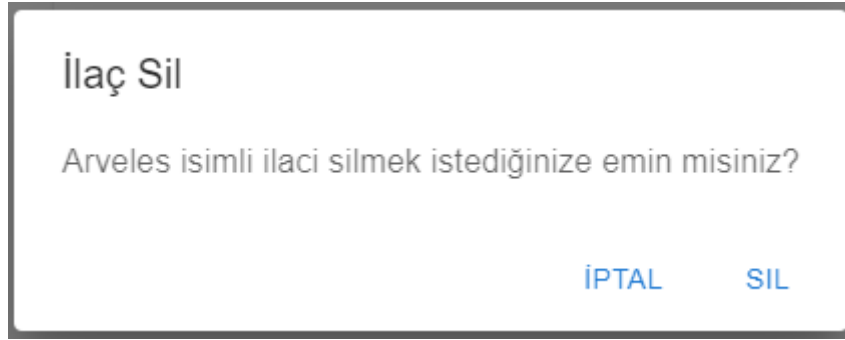
İlac Detayı

Agri Kesici

İPTAL DÜZENLE

Şekil 4.4. İlaç düzenle formu

Silinecek ilaç sil butonu ile silinir.



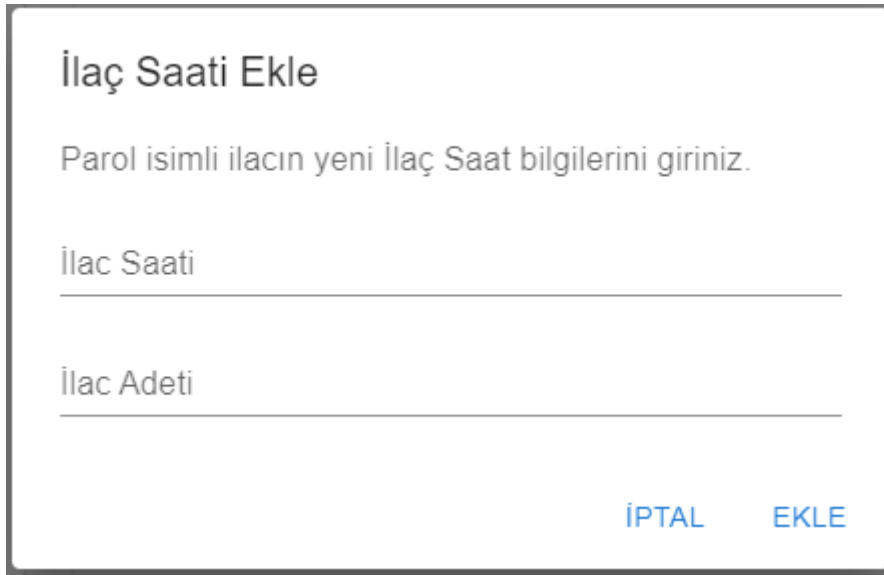
İlaç Sil

Arveles isimli ilaci silmek istediğimize emin misiniz?

İPTAL SIL

Şekil 4.5. İlaç sil formu

Seçili ilaca yeni ilaç saati eklemek için ilaç saati ve adeti istenen form doldurulur.



İlaç Saati Ekle

Parol isimli ilacın yeni İlaç Saat bilgilerini giriniz.

İlac Saati

İlac Adeti

İPTAL EKLE

Şekil 4.6. İlaç saati ekle formu

Düzenlenecek ilacın eski saat bilgileri yeni bilgiler ile değiştirilir.

İlaç Saati Düzenle

Parol isimli ilacın 14:00:00 saatinin yeni İlaç Saat bilgilerini giriniz.

İlaç Saati

14:00:00

İlaç Adeti

1

İPTAL DÜZENLE

Şekil 4.7. İlaç saati düzenle formu

Silinecek ilaç saati sil butonu ile silinir.

İlaç Saati Sil

Parol isimli ilacın 14:00:00 saatini silmek istediğinize emin misiniz?

İPTAL SIL

Şekil 4.8. İlaç saati sil formu

2.2.5. React Native

React Native cross-platform mobil uygulama geliştirme frameworküdür.

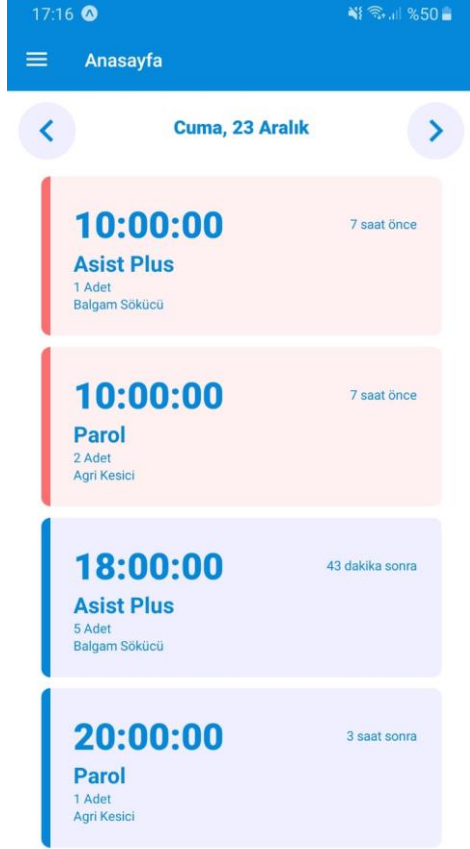


Şekil 5.1. Mobil uygulama giriş sayfası

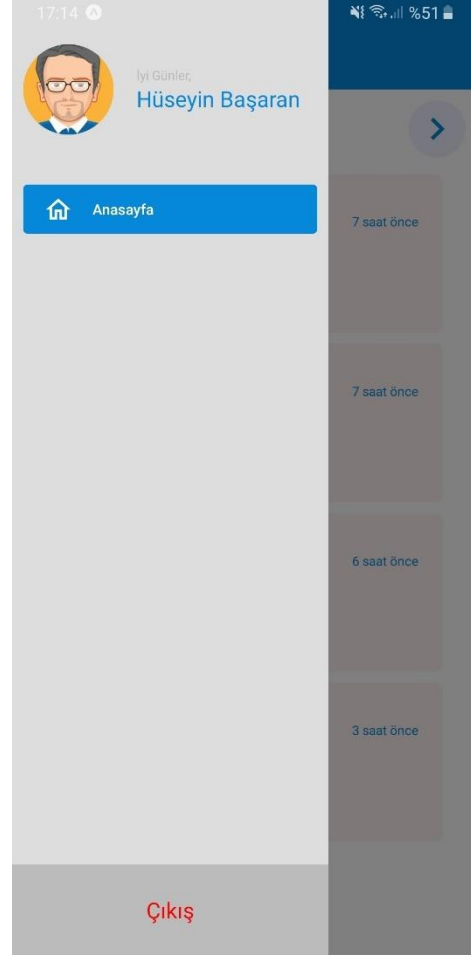


Şekil 5.2. Mobil uygulama kayıt sayfası

Mobil uygulama giriş sayfası ile başlar. Eğer hasta bir hesaba sahip ise giriş yapar değil ise kayıt olur.



Şekil 5.3. Mobil uygulama anasayfası

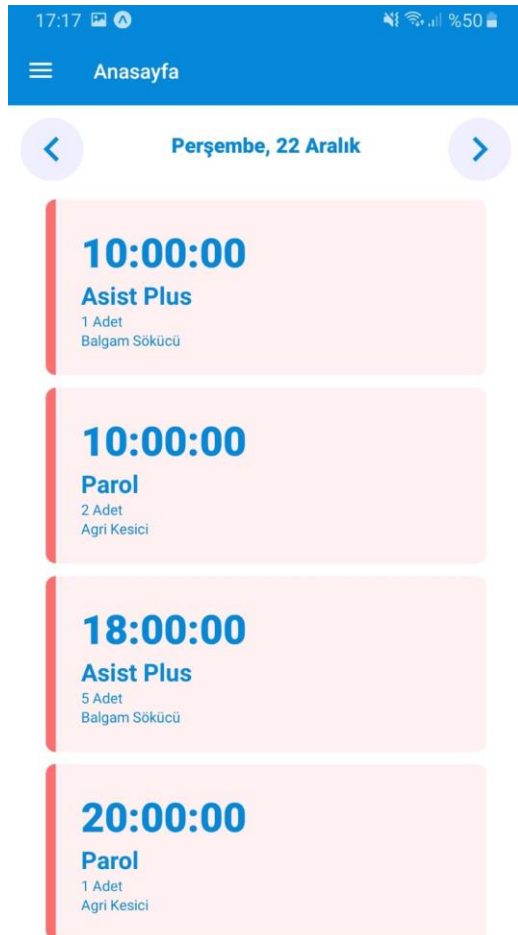


Şekil 5.4. Mobil uygulama menü

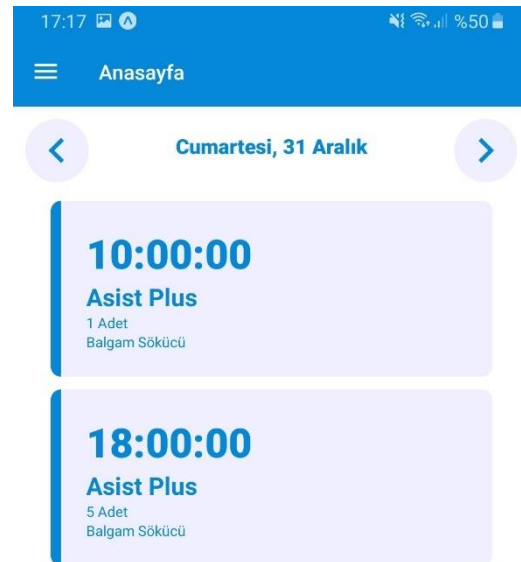
Uygulamaya giriş yapan kullanıcı anasayfaya yönlendirilir. Anasayfada hastanın bugün alması gereken ilaçlar, saat ve adetleri ile listelenir. Sol üstteki menu tuşu ile menü açılır.

İlaç saatlerine kalan ve geçen süre ilaçların sağ tarafında gösterilir. İlaçlar, eğer ilaç saati geçmiş ise kırmızı geçmemiş ise mavi renkte gösterilir.

Sayfanın üst kısmındaki tarihin sağ ve solundaki butonlar ile ileriki veya geçmiş tarihler listelenebilir. Bu listelerde de tarih geçmiş ise kırmızı geçmemiş ise mavi gösterim kullanılır.

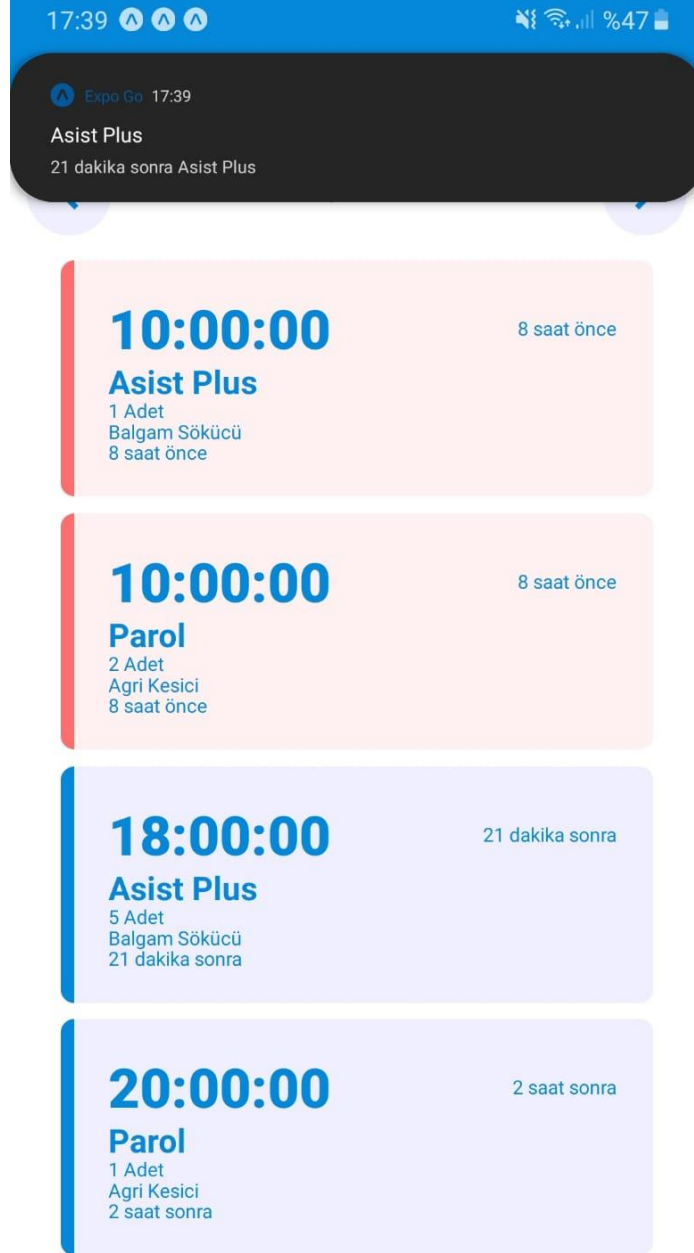


Şekil 5.5. Mobil uygulama geçmiş tarih



Şekil 5.6. Mobil uygulama gelecek tarih

Eğer ilacın kullanılması gereken süre 30 dakikadan az ise telefona bu bilgi bildirim olarak gönderilir. Hastaya bu sayede yaklaşan ilaç saatini hatırlatmış olur.



Şekil 5.7. Mobil uygulama bildirim

BÖLÜM 3. SONUÇLAR VE ÖNERİLER

Projenin sunucu kısmının veritabanına verileri iş kurallarına uygun bir şekilde eklememizi ve sorgulamamızı sağladığını, aile hekiminin verileri görüntüleyebildiği ve düzenleyebildiğini, hastaların ilaç saalarının listelendiğini ve hatırlatıcı sayesinde önceden haberdar oluklarını görmüş olmaktadır.

Hastaların aile hekimlerinden randevu alabilmeleri de projeye dahil edilebilir.

KAYNAKLAR

- [1] <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0>
- [2] <https://reactjs.org/tutorial/tutorial.html>
- [3] <https://reactnative.dev/docs/getting-started>
- [4] <https://www.titck.gov.tr/dinamikmodul/43>
- [5] <https://hastaneler.erciyes.edu.tr/c/42-1/hastane-ilac-listesi>

ÖZGEÇMİŞ

Hüseyin Burhan Başaran, 01.07.2000 tarihinde Ankara’da doğdu. İlk, orta ve lise eğitimini Ankara’da tamamladı. 2018 yılında Pursaklar Anadolu Lisesi’nden mezun oldu. 2019 yılında Sakarya Üniversitesi Bilgisayar Mühendisliği Bölümü’nü kazandı ve aktif eğitim hayatına Sakarya Üniversitesi bünyesinde devam etmektedir. 2022 yılında Sakarya Belediyesinde donanım stajını yapmıştır.

BSM 401 BİLGİSAYAR MÜHENDİSLİĞİ TASARIM ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU: Aile Hekimi Hasta Takip Sistemi

ÖĞRENCİLER (Öğrenci No/AD/SOYAD): G191210077 / Hüseyin Burhan Başaran

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

DANIŞMAN: PROF. DR. NEJAT YUMUŞAK

DANIŞMAN İMZASI: