

2024-2025 Güz Yarıyılı Dönemi

Sakarya Üniversitesi

Haşim Gürdamar Bilgisayar ve Bilişim Bilimleri
Mühendisliği Fakültesi Bilgisayar Mühendisliği
Bölümü

Veri Tabanı Yönetim Sistemi Proje Raporu

Ömer Faruk Aydın
G221210370
2. Öğretim A

Dersi Verenler:
Prof.Dr. CELAL ÇEKEN

Kedi Hastanesi

Hasta Yönetim Sistemi

****a) Uygulamanın kısa tanıtımı, iş kuralları, ilişkisel şema (metinsel gösterim)**

Proje Tanıtımı

Bu proje, "Kedi Hastanesi Hasta Yönetim Sistemi" adında bir yazılım uygulamasıdır. Amaç, hasta kayıtlarını, randevuları, tedavi süreçlerini ve ilgili stok yönetimini dijital ortamda takip etmektir. Proje kapsamında PostgreSQL kullanılarak kapsamlı bir veritabanı tasarlanmış ve Node.js tabanlı bir uygulama geliştirilmiştir.

İş Kuralları

















- Her hasta bir veya birden fazla randevuya sahip olabilir.
- Her randevu bir veteriner ve bir hasta ile ilişkilendirilmelidir.
- Randevu sonrası otomatik bir tedavi kaydı oluşturulmalıdır.
- Stok seviyesi kritik eşiklere ulaştığında uyarı verilmelidir.

İlişkisel Şema (Metinsel Gösterim)

Tablo Listesi

- tblHasta** (HastaID: Integer, Isim: Char, Tur: Char, Yas: Integer, Cinsiyet: Char, SaglikDurumu: Text, SahipBilgileri: Text)
- tblPersonel** (PersonelID: Integer, Isim: Char, Telefon: Char, Adres: Text, Pozisyon: Char)
- tblVeteriner** (VeterinerID: Integer, UzmanlikAlani: Char, CalismaSaatleri: Text)
- tblDestekPersoneli** (DestekPersonelID: Integer, GorevTanimi: Text)
- tblRandevuTuru** (RandevuTurID: Integer, TurAdi: Char)
- tblRandevu** (RandevuID: Integer, HastaID: Integer, VeterinerID: Integer, Tarih: Date, Saat: Time, Durum: Char, RandevuTurID: Integer)
- tblTedaviTuru** (TedaviTurID: Integer, TedaviAdi: Char)
- tblTedavi** (TedaviID: Integer, RandevuID: Integer, Aciklama: Text, Tarih: Date, TedaviTurID: Integer)
- tblRecete** (ReceteID: Integer, TedaviID: Integer, IlacBilgisi: Text, KullanimTalimati: Text)
- tblStok** (StokID: Integer, UrunAdi: Char, Kategori: Char, Miktar: Integer, SonKullanmaTarihi: Date)

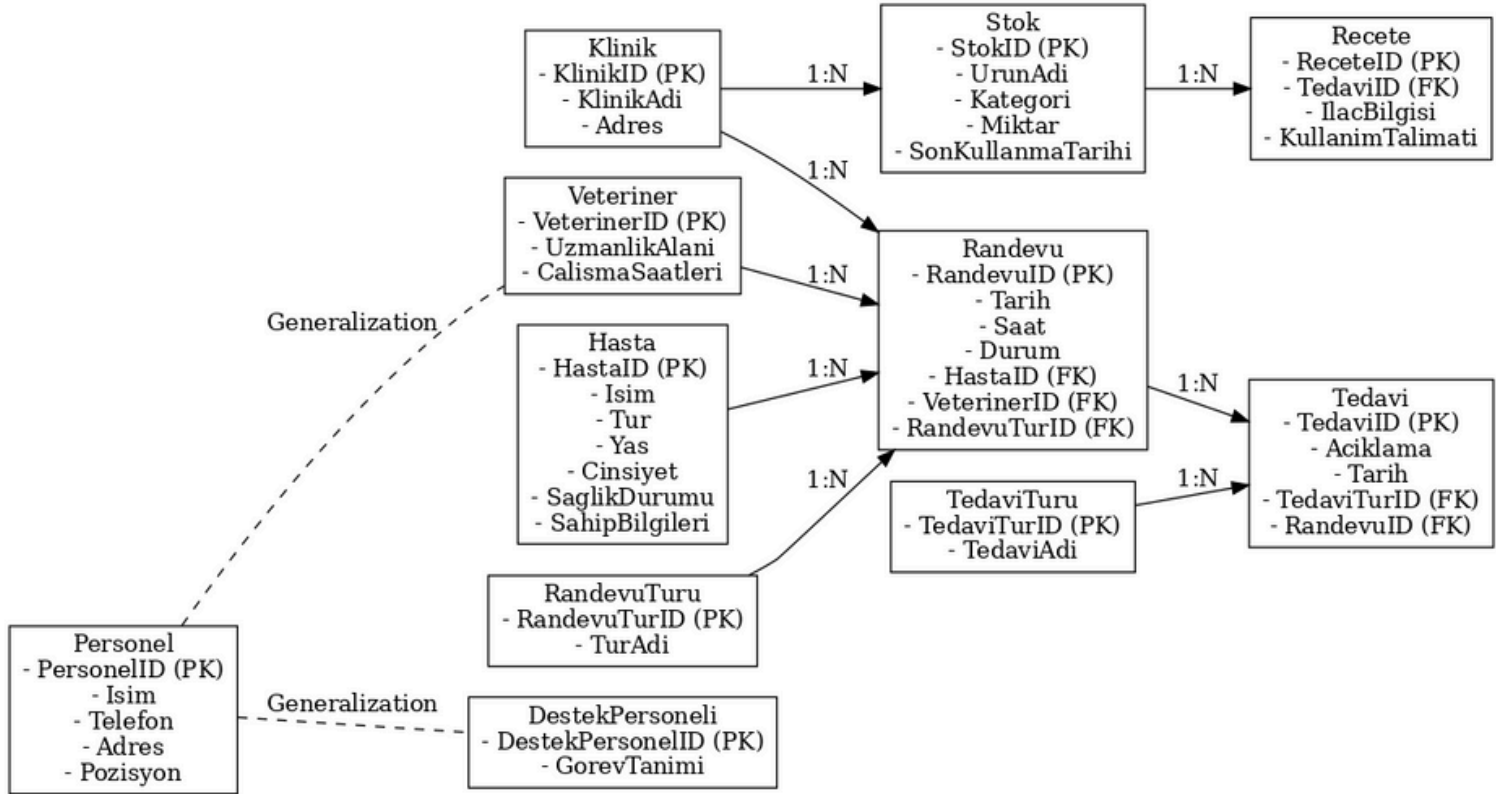
- **tblStokRecete** (StokID: Integer, ReceteID: Integer)
- **tblKlinik** (KlinikID: Integer, KlinikAdi: Char, Adres: Text)
- **tblKlinikRandevu** (KlinikID: Integer, RandevuID: Integer)
- **tblKlinikStok** (KlinikID: Integer, StokID: Integer, Miktar: Integer)
- **tblTedaviKayit** (TedaviKayitID: Integer, TedaviID: Integer, Aciklama: Text, Tarih: Date)

- ▼  Tables (15)
 - >  destekpersoneli
 - >  hasta
 - >  klinik
 - >  klinikrandevu
 - >  klinikstok
 - >  personel
 - >  randevu
 - >  randevuturu
 - >  recete
 - >  stok
 - >  stokrecete
 - >  tedavi
 - >  tedavikayit
 - >  tedaviture
 - >  veteriner

****b) Varlık Bağını modeli (Crow's Foot, Kalıtım)**

Varlık Bağını Modeli

- Varlık bağını diyagramı Crow's Foot notasyonu ile hazırlanmıştır.
- **Genelleme (Kalıtım):** Personel tablosu üst tablodur; Veteriner ve DestekPersoneli alt tablolardır.



****c) Veritabanını, içerisindeki verilerle birlikte oluşturmayı sağlayan SQL ifadeleri**

Veritabanını ve Verileri Oluşturan SQL Scriptleri

```
-- Hasta Tablosu
CREATE TABLE Hasta (
    HastaID SERIAL PRIMARY KEY,
    Isim VARCHAR(100) NOT NULL,
    Tur VARCHAR(50),
    Yas INT,
    Cinsiyet VARCHAR(10),
    SaglikDurumu TEXT,
    SahipBilgileri TEXT
);

-- Personel Tablosu (Üst Tablo)
CREATE TABLE Personel (
    PersonelID SERIAL PRIMARY KEY,
    Isim VARCHAR(100) NOT NULL,
    Telefon VARCHAR(15),
    Adres TEXT,
    Pozisyon VARCHAR(50) NOT NULL
);

-- Veteriner Tablosu (Personel'den türetilmiş)
CREATE TABLE Veteriner (
    VeterinerID INT PRIMARY KEY,
    UzmanlikAlani VARCHAR(100),
    CalismaSaatleri TEXT,
    FOREIGN KEY (VeterinerID) REFERENCES Personel(PersonelID)
);

-- DestekPersoneli Tablosu (Personel'den türetilmiş)
CREATE TABLE DestekPersoneli (
    DestekPersonelID INT PRIMARY KEY,
    GorevTanimi TEXT,
    FOREIGN KEY (DestekPersonelID) REFERENCES Personel(PersonelID)
);

-- Randevu Türü Tablosu
CREATE TABLE RandevuTuru (
    RandevuTurID SERIAL PRIMARY KEY,
    TurAdi VARCHAR(50) NOT NULL
);
```

```
-- Randevu Tablosu
CREATE TABLE Randevu (
    RandevuID SERIAL PRIMARY KEY,
    HastaID INT NOT NULL,
    VeterinerID INT NOT NULL,
    Tarih DATE NOT NULL,
    Saat TIME NOT NULL,
    Durum VARCHAR(50),
    RandevuTurID INT,
    FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),
    FOREIGN KEY (VeterinerID) REFERENCES Veteriner(VeterinerID),
    FOREIGN KEY (RandevuTurID) REFERENCES RandevuTuru(RandevuTurID)
);

-- Tedavi Türü Tablosu
CREATE TABLE TedaviTuru (
    TedaviTurID SERIAL PRIMARY KEY,
    TedaviAdi VARCHAR(100) NOT NULL
);

-- Tedavi Tablosu
CREATE TABLE Tedavi (
    TedaviID SERIAL PRIMARY KEY,
    RandevuID INT NOT NULL,
    Aciklama TEXT,
    Tarih DATE,
    TedaviTurID INT,
    FOREIGN KEY (RandevuID) REFERENCES Randevu(RandevuID),
    FOREIGN KEY (TedaviTurID) REFERENCES TedaviTuru(TedaviTurID)
);

-- Recete Tablosu
CREATE TABLE Recete (
    ReceteID SERIAL PRIMARY KEY,
    TedaviID INT NOT NULL,
    IlacBilgisi TEXT,
    KullanimTalimati TEXT,
    FOREIGN KEY (TedaviID) REFERENCES Tedavi(TedaviID)
);
```

****d) Saklı yordam (4 adet), tetikleyici (4 adet) (Kullanılan saklı yordam/fonksiyon ve tetikleyiciler (.sql koddan bağımsız olarak) maddeler halinde gösterilmeli ve işlevsel olarak ne yaptıkları kısaca açıklanmalıdır.)**

Saklı Yordamlar

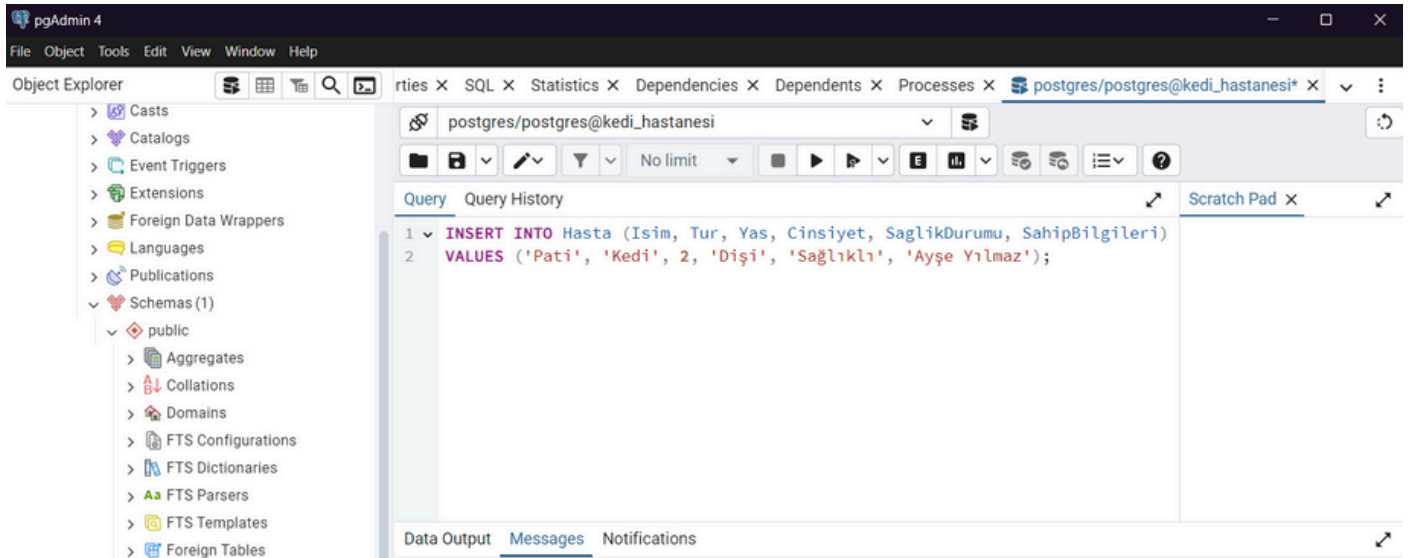
Functions (4)

- hastaekle(p_isim character varying, p_tur character varying, p_yas integer, p_cinsiyet character varying, p_saglikdurumu text, p_sahipbilgileri text)
- klunikstokkontrol(p_klinikid integer, p_urunadi character varying)
- stokazalt(p_stokid integer, p_miktar integer)
- tedaviucrethesapla(p_tedaviturid integer)

1. Yeni Hasta Ekleme

İşlev: Yeni bir hasta kaydı eklemek için kullanılır. Parametre olarak hastanın bilgilerini alır ve Hasta tablosuna yeni bir kayıt ekler.

```
CREATE OR REPLACE FUNCTION HastaEkle(  
  p_isim VARCHAR, p_Tur VARCHAR, p_Yas INT, p_Cinsiyet VARCHAR, p_SaglikDurumu TEXT, p_SahipBilgileri TEXT  
) RETURNS VOID AS $$  
BEGIN  
  INSERT INTO Hasta(Isim, Tur, Yas, Cinsiyet, SaglikDurumu, SahipBilgileri)  
  VALUES (p_isim, p_Tur, p_Yas, p_Cinsiyet, p_SaglikDurumu, p_SahipBilgileri);  
END;  
$$ LANGUAGE plpgsql;
```



Data Output Messages Notifications

NOTICE: Yeni hasta kaydı eklendi: Pati
INSERT 0 1

Query returned successfully in 42 msec.

✓ Query returned successfully in 42 msec. ✕

Total rows: Query complete 00:00:00.042

CRLF Ln 2, Col 63

Kayıtlı Hastalar						
Isim	Tur	Yas	Cinsiyet	Saglik Durumu	Sahip Bilgileri	İşlemler
Pati	Kedi	2	Dişi	Sağlıklı	Ayşe Yılmaz	Sil Görüntüle

2. Tedavi Ücret Hesaplama

İşlev: Tedavi türüne göre ücret hesaplar. Tedavi türü parametresi alır ve buna bağlı olarak sabit bir ücret döndürür.

```
CREATE OR REPLACE FUNCTION TedaviUcretHesapla(p_TedaviTurID INT) RETURNS NUMERIC AS $$
DECLARE
    ucret NUMERIC;
BEGIN
    SELECT CASE
        WHEN p_TedaviTurID = 1 THEN 100.00
        WHEN p_TedaviTurID = 2 THEN 200.00
        ELSE 150.00
    END INTO ucret;
    RETURN ucret;
END;
$$ LANGUAGE plpgsql;
```

3. Stok Azaltma

İşlev: Stoktaki miktarı azaltır. Kritik seviyenin altına düşerse hata mesajı oluşturur.

```
CREATE OR REPLACE FUNCTION StokAzalt(p_StokID INT, p_Miktar INT) RETURNS VOID AS $$
BEGIN
    UPDATE Stok SET Miktar = Miktar - p_Miktar WHERE StokID = p_StokID;
    IF (SELECT Miktar FROM Stok WHERE StokID = p_StokID) < 0 THEN
        RAISE EXCEPTION 'Stok yetersiz';
    END IF;
END;
$$ LANGUAGE plpgsql;
```

4. Klinik Stok Kontrol

İşlev: Belirli bir klinikteki stok miktarını kontrol eder. Klinik ID ve ürün adı parametrelerini alır, mevcut miktarı döndürür.

```
CREATE OR REPLACE FUNCTION KlinikStokKontrol(p_KlinikID INT, p_UrunAdi VARCHAR) RETURNS INT AS $$
DECLARE
    miktar INT;
BEGIN
    SELECT ks.Miktar INTO miktar
    FROM KlinikStok ks
    JOIN Stok s ON ks.StokID = s.StokID
    WHERE ks.KlinikID = p_KlinikID AND s.UrunAdi = p_UrunAdi;
    RETURN miktar;
END;
$$ LANGUAGE plpgsql;
```

Tetikleyiciler

- ▼ Trigger Functions (5)
 - randevusonrasitedavi()
 - stokkritikseviye()
 - tedavisilmekontrol()
 - yenihastabildirim()
 - yenihastabildirimi()

1. Yeni Hasta Bildirimi

İşlev: Yeni bir hasta kaydedildiğinde, hasta adını içeren bir bildirim oluşturur.

```
CREATE OR REPLACE FUNCTION YeniHastaBildirimi() RETURNS TRIGGER AS $$
BEGIN
    RAISE NOTICE 'Yeni hasta kaydı eklendi: %', NEW.Isim;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER YeniHastaBildirimiTrigger
AFTER INSERT ON Hasta
FOR EACH ROW
EXECUTE FUNCTION YeniHastaBildirimi();
```

2. Randevu Sonrası Tedavi Oluşturma

İşlev: Yeni bir randevu kaydedildiğinde otomatik olarak tedavi kaydı oluşturur.

```
CREATE OR REPLACE FUNCTION RandevuSonrasiTedavi() RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO Tedavi (RandevuID, Aciklama, Tarih)
    VALUES (NEW.RandevuID, 'Otomatik Tedavi Kaydı', CURRENT_DATE);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER RandevuSonrasiTedaviTrigger
AFTER INSERT ON Randevu
FOR EACH ROW
EXECUTE FUNCTION RandevuSonrasiTedavi();
```


3. Stok Kritik Seviyesi

İşlev: Stok seviyesi kritik seviyeye ulaştığında uyarı bildirimi oluşturur.

```
CREATE OR REPLACE FUNCTION StokKritikSeviye() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.Miktar < 10 THEN
        RAISE NOTICE 'Stok kritik seviyede: %', NEW.UrunAdi;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER StokKritikSeviyeTrigger
AFTER UPDATE ON Stok
FOR EACH ROW
EXECUTE FUNCTION StokKritikSeviye();
```

4. Tedavi Silme Kontrolü

İşlev: Tedavi silindiğinde ilişkili reçeteleri otomatik olarak siler.

```
CREATE OR REPLACE FUNCTION TedaviSilmeKontrol() RETURNS TRIGGER AS $$
BEGIN
    DELETE FROM Recete WHERE TedaviID = OLD.TedaviID;
    RETURN OLD;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER TedaviSilmeKontrolTrigger
BEFORE DELETE ON Tedavi
FOR EACH ROW
EXECUTE FUNCTION TedaviSilmeKontrol();
```

****e) Arama, Ekleme, Silme, Güncelleme işlemlerine ait ekran görüntüleri**

Uygulama Ekran Görüntüleri

Uygulama Başlangıç

```
Komut İstemi - node index.js x + v
Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\leviv>cd C:\Users\leviv\kedi_hastanesi1

C:\Users\leviv\kedi_hastanesi1>node index.js
Server running at http://localhost:3000
Veritabanına başarıyla bağlanıldı. Sunucu zamanı: 2024-12-20T13:16:11.894Z
|
```

Kedi Hastanesi Hasta Yönetimi

Yeni Hasta Ekle

İsim:

Tür:

Yaş:

Cinsiyet:

Sağlık Durumu:

Sahip Bilgileri:

Kaydet

Hasta Ara

İsim:

Tür:

Ara

Kayıtlı Hastalar

İsim	Tür	Yaş	Cinsiyet	Sağlık Durumu	Sahip Bilgileri	İşlemler
Minik	Kedi	2	Dişi	Sağlıklı	Ayşe Yılmaz	Sil Güncelle

Ekran Görüntüleri: Arama, ekleme, silme ve güncelleme işlemleri için uygulama ekran görüntüleri aşağıdaki gibidir:

1.Hasta Ekleme Kısmı

Kedi Hastanesi Hasta Yönetimi

Yeni Hasta Ekle

İsim:

Tür:

Yaş:

Cinsiyet:

Sağlık Durumu:

Sahip Bilgileri:

Kaydet

Hasta Ekleme (gif):

Kedi Hastanesi Hasta Yönetimi

Yeni Hasta Ekle

İsim:

Tür:

Yaş:

Cinsiyet:

Sağlık Durumu:

Sahip Bilgileri:

Kaydet

Hasta Ara

İsim:

Tür:

Ara

Kayıtlı Hastalar

İsim	Tür	Yaş	Cinsiyet	Sağlık Durumu	Sahip Bilgileri	İşlemler
Musa	Kedi	2	Dişi	Sağlıklı	Ayran Vitrasi	Sil Görüntüle

2.Hasta Arama Kısmı

Hasta Ara

İsim:

Tür:

Ara

Hasta arama (gif):

Kedi Hastanesi

localhost:3000

Kedi Hastanesi Hasta Yönetimi

Yeni Hasta Ekle

İsim:

Tür:

Yaş:

Cinsiyet:

Sağlık Durumu:

Sahip Bilgileri:

Kaydet

Hasta Ara

İsim:

Tür:

Ara

Kayıtlı Hastalar

İsim	Tür	Yaş	Cinsiyet	Sağlık Durumu	Sahip Bilgileri	İşlemler
Minik	Kedi	2	Dişi	Sağlıklı	Ayşe Yılmaz	<button>Sil</button> <button>Düzenle</button>
Mümin	Tekir	5	Dişi	Ayapında incirler var	Emel Hanım Aytemir	<button>Sil</button> <button>Düzenle</button>

3.Hasta Listeleme Kısmı

Kayıtlı Hastalar						
İsim	Tür	Yaş	Cinsiyet	Sağlık Durumu	Sahip Bilgileri	İşlemler
Minik	Kedi	2	Dişi	Sağlıklı	Ayşe Yılmaz	<button>Sil</button> <button>Düzenle</button>

4.Hasta Güncelleme Kismi

Düzenleme (gif):

localhost:3000 web sitesinin mesajı

Yeni İsim:

Tamam İptal

İsim:

Yaş:

Cinsiyet:

Sağlık Durumu:

Sahip Bilgileri:

Kaydet

Hasta Ara

İsim:

Tür:

Ara

Kayıtlı Hastalar

İsim	Tür	Yaş	Cinsiyet	Sağlık Durumu	Sahip Bilgileri	İşlemler
Elina	İnsi	2	Dişi	Çiğdem	Ayşe Yılmaz	Sil Güncelle
Mahir	İnsi	5	Erkek	Ayagında yarım var	Ömer Faruk Aydın	Sil Güncelle

Silme (gif):

Kedi Hastanesi Hasta Yönetimi

Yeni Hasta Ekle

İsim:

Tür:

Yaş:

Cinsiyet:

Sağlık Durumu:

Sahip Bilgileri:

Kaydet

Hasta Ara

İsim:

Tür:

Ara

Kayıtlı Hastalar

İsim	Tür	Yaş	Cinsiyet	Sağlık Durumu	Sahip Bilgileri	İşlemler
Mahir	İnsi	5	Dişi	Ayagında yarım var	Ömer Faruk Aydın	Sil Güncelle
Mahir	İnsi	2	Erkek	İyi	Ayşe Yılmaz	Sil Güncelle

İşlemler

Sil

Düzenle

****f) Uygulamanın kaynak kodları (Kaynak kodlarını Rapor.pdf içerisine ekleyebilirsiniz ya da uygulamanızın kaynak kodlarını içeren depo oluşturarak (github, bitbucket vb.) bağlantı adresini verebilirsiniz)**

6. Uygulamanın Kaynak Kodları

Tüm kaynak kodlar GitHub deposunda bulunmaktadır. [GitHub Linki](https://github.com/leviSEC/DatabaseManagmentSystems-Kedi_Hastanesi)

https://github.com/leviSEC/DatabaseManagmentSystems-Kedi_Hastanesi