

ÖĞRETİM: **ZAFER GÜNEY**

DERS : **Görsel Programlama (FET309)**  ÖDEV : Ödev-Proje

KURUM : İSTANBUL TOPKAPI ÜNİVERSİTESİ

ÖĞRENCI AD SOYAD: MUSA AL AHMED

ÖĞRENCI NO : 23040301118

BÖLÜM \ SINIF : YAZILIM MÜHENDISLIĞI \ 3

### **Hastane Otomasyon Sistemi Projesi Raporu**

### **İçindekiler**

1. **Giriş (Müdafaa)**
   1. Hastane Otomasyon Sistemi Projesine Genel Bakış.
2. **Proje Tanımı**
   1. Sistem İşlevsellikleri.
   2. Kullanıcı Türleri ve Yetkileri.
   3. Proje Kapsamı ve Teknik Detaylar.
3. **Veri Tabanı Tabloları**
   1. Tabloların Yapısı (Tablo Adları ve Alanları).
   2. Alan Tipleri ve İlişkiler.
   3. Veritabanı Görselleri ve Örnek Veriler.
4. **Kodlar ve Açıklamalar**
   1. Kodun Genel Mimarisi.
   2. Fonksiyonlar ve İşlevleri.
   3. Kullanılan Teknolojiler ve Metotlar.
   4. Kodun Örnek Bölümleri ve Açıklamalar.

### **1. Giriş (Müdafaa)**

Hastane Otomasyon Sistemi Projesi, modern tıp ve sağlık hizmetlerinde verimliliği artırmak amacıyla geliştirilmiştir. Teknolojinin ilerlemesiyle birlikte hastanelerdeki operasyonel yükün azaltılması, hasta memnuniyetinin artırılması ve zamanın etkili kullanılması temel hedeftir.

Bu proje, hastanelerin randevu sistemlerinden personel yetkilendirme mekanizmalarına kadar birçok özelliği bir araya getirerek; hastaların, doktorların ve sekreterlerin iş yükünü azaltmayı amaçlamaktadır. Aynı zamanda, kolay erişim ve veri bütünlüğü gibi önemli konular da projenin kapsamına dahil edilmiştir.

Bu raporda, projenin temel amaçları, teknik detayları ve sunulan çözümler detaylı bir şekilde ele alınmış ve yazılım mimarisi ayrıntıları ile birlikte açıklanılmıştır.

### **2. Proje Tanımı**

#### **Sistem İşlevsellikleri**

Hastane Otomasyon Sistemi, hastanelerin günlük operasyonlarını kolaylaştırmak ve kullanıcılarına daha iyi bir deneyim sunmak için şu işlevselliklere sahiptir:

* Randevu alma, iptal etme ve güncelleme.
* Hasta bilgileri ve şikayet kayıtlarının tutulması.
* Doktor ve sekreter yetkilendirme işlemleri.
* Duyuru oluşturma ve yayınlam.
* Kullanıcı dostu arayüzler ile kolay erişim ve veri yönetimi.

#### **Kullanıcı Türleri ve Yetkileri**

Bu sistemde üç ana kullanıcı türü vardır:

**Hasta:**

* 1. Randevu alma, iptal etme ve şikayet kayıtları.
  2. Geçmiş randevularını görüntüleme.

1. **Doktor:**
   1. Hasta bilgilerine erişim.
   2. Randevu listelerini kontrol etme ve düzenleme.
2. **Sekreter:**
   1. Doktor ve randevu yönetimi.
   2. Duyuru yayınlama ve güncelleme.

#### **Proje Kapsamı ve Teknik Detaylar**

* **Proje Kapsamı:** Hastanelerin randevu sistemi, hasta kayıt ve veri yönetimi işlemlerini dijitalleştirerek daha etkili ve hatasız bir sistem sunmak.
* **Kullanılan Teknolojiler:**
  + Programlama Dili: C#
  + Veritabanı: SQL Server
  + Arayüz Teknolojisi: Windows Forms

### **Projenin Değerlendirme Kriterleri**

1. Matematiksel formüller kullanılarak hesaplamalar olmalı

2. Görsel olarak hareket, dönen cisimler bulunmalı

3. Form doldurma işlemi olacak. Kayıt yaptıran kişinin bilgileri girilecek ve bilgilerin doğru girip girmediği kontrol edilecek (alan özellikleri: tel :10 ve TC 11 hane gibi).

4. Form girişinde fotograf yükleme olacak. Foto nun boyutları düzenlenebilecek.

5. Bilgiler veritabanına kaydedilecek. /ekle sil güncelle listele dahil

6. Bilgileri görebilmek için kişiler yada admin Üye girişi yapacak. Bunun için İlk kayıtta Kullanıcı adı ve şifreleri de kayıt altına alınacak.

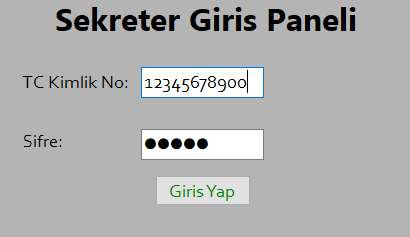
7. Bilgileri listelenirken kişilerin yanında küçük olarak fotografları gösterilecek.

8. Listeleme ekranında detay düğmesine tıklayınca başka sayfa açılacak burada onunla ilgili detay bilgileri listelenecek güncellemeyide burada yapabilirsiniz.

9. Veriler veri tabanına kaydedilirken aynı esnada derste işlediğimiz gibi form üzerinde yer alan datagrid üzerindende işlem(ekle sil güncelle listele) yapılabilecek

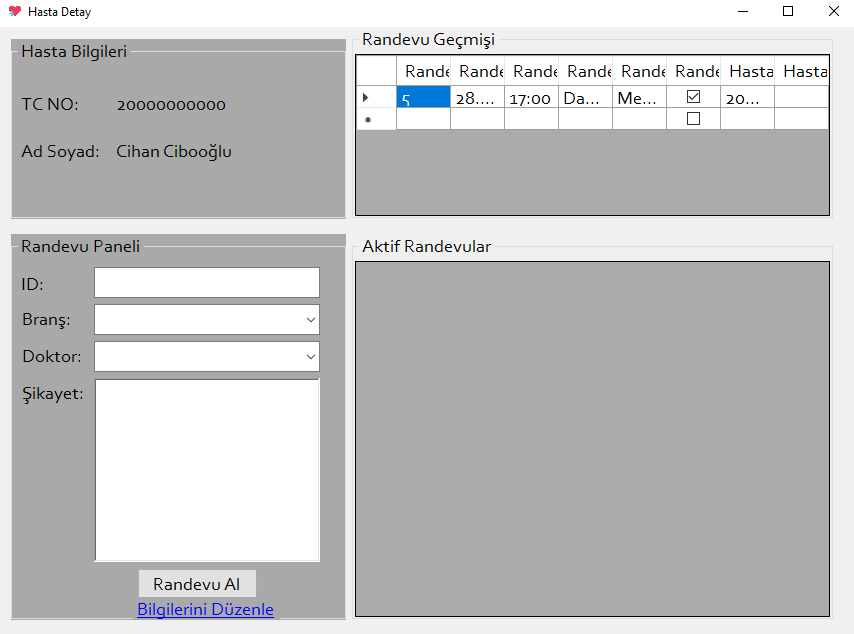
1.PNG (Hastane Otomasyonu)

Görsel olarak hareket, dönen cisimler bulunmalı

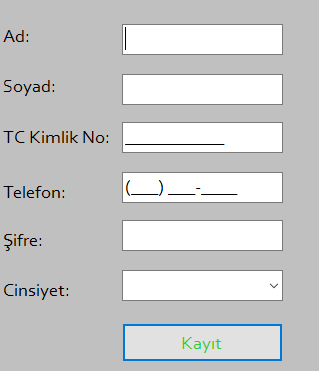


2: **Form doldurma işlemi olacak (Kayıt yaptıran kişinin bilgileri girilecek ve doğruluğu kontrol edilecek)**

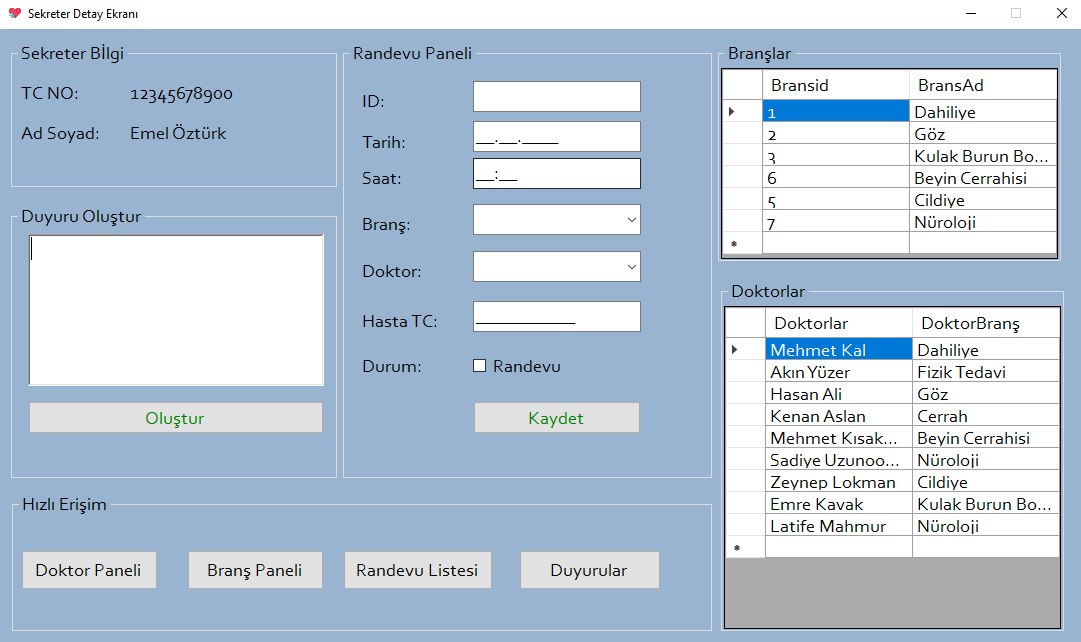
**(Hasta Detay - İlk Görünüm):**



**(Hasta Kayıt):**

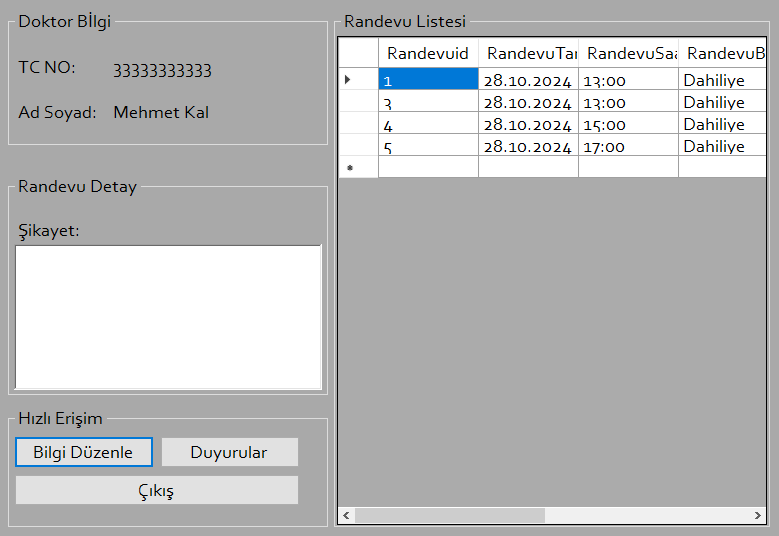


**(Sekreter Detay - İlk Görünüm):**



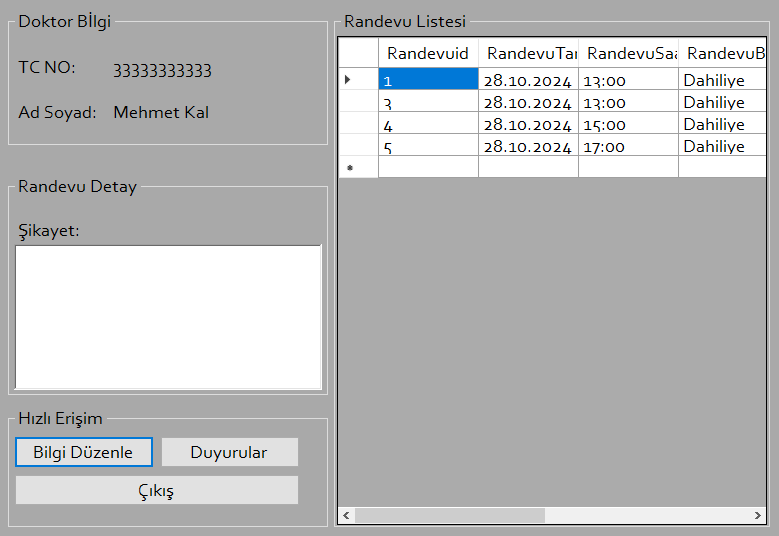
**3 : Form girişinde fotograf yükleme olacak** Form girişinde fotograf yükleme olacak. Foto nun boyutları düzenlenebilecek.

(Hasta Kayıt)

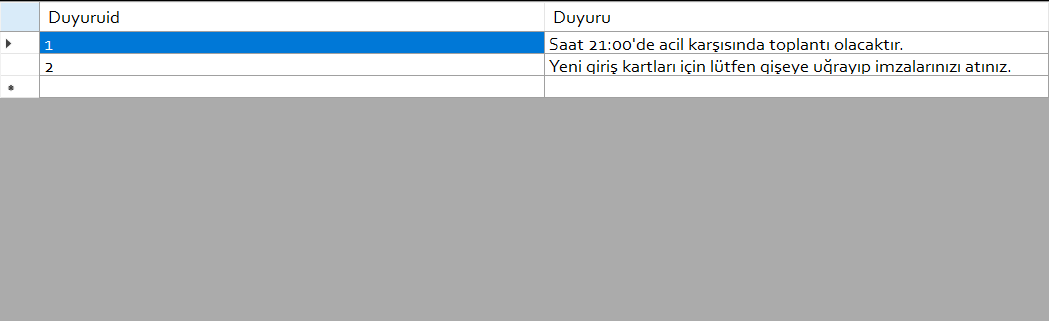


4: **Bilgiler veri tabanına kaydedilecek (ekle, sil, güncelle, listele)**

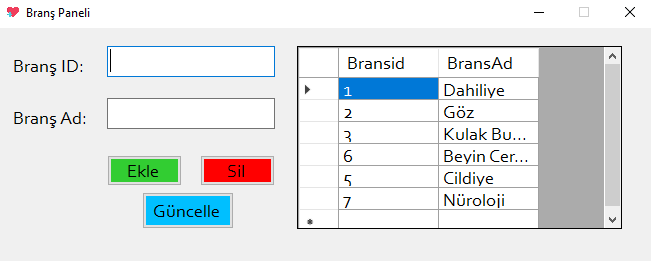
(Doktor Bilgi).



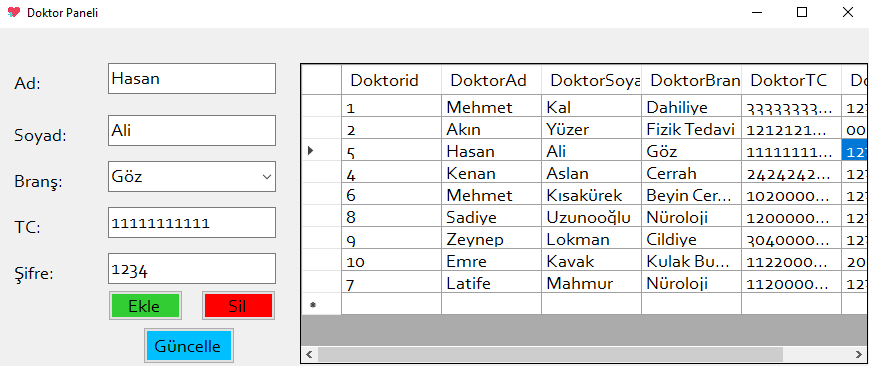
(Duyurular)



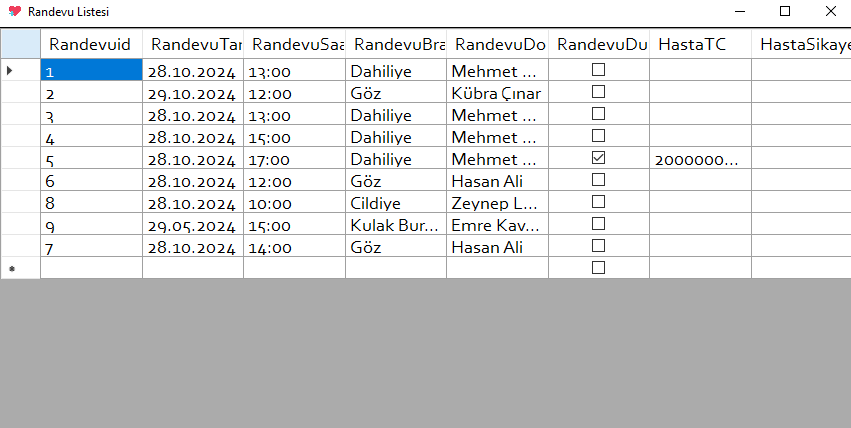
(Branş Paneli)



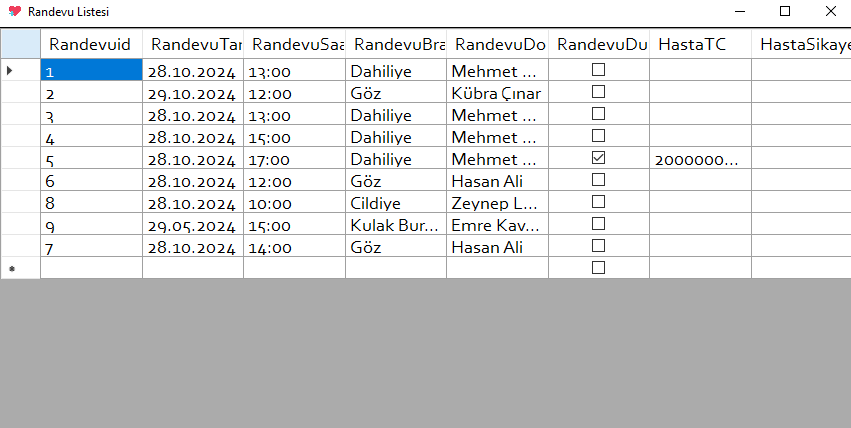
(Doktor Yönetimi)



(Randevu Listesi)

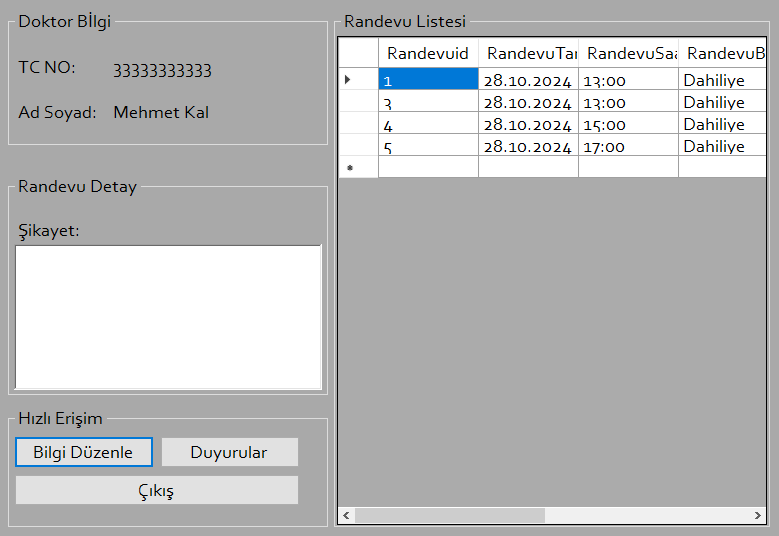


### **5: Bilgileri listelenirken kişilerin yanında küçük olarak fotografları gösterilecek**

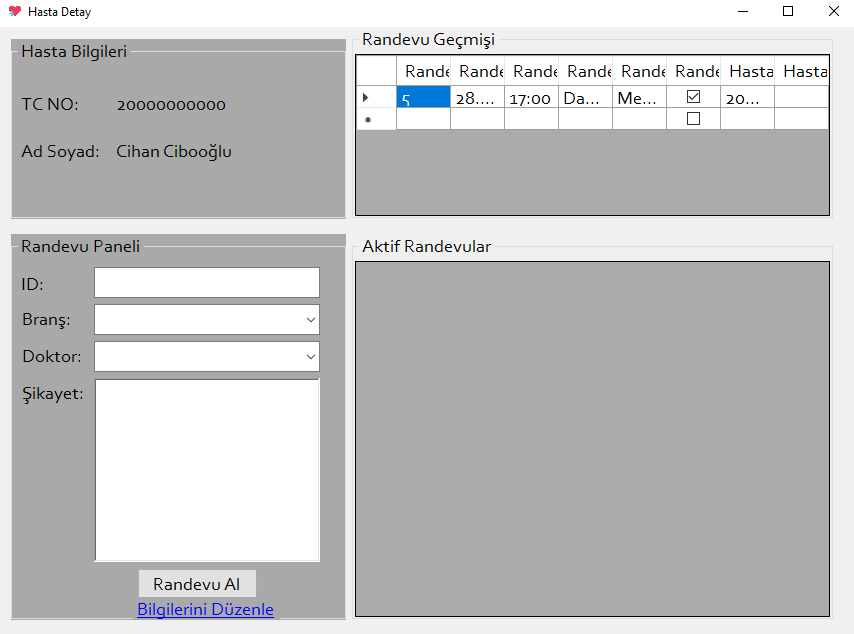


### **6: Listeleme ekranında detay düğmesine tıklayınca başka sayfa açılacak**

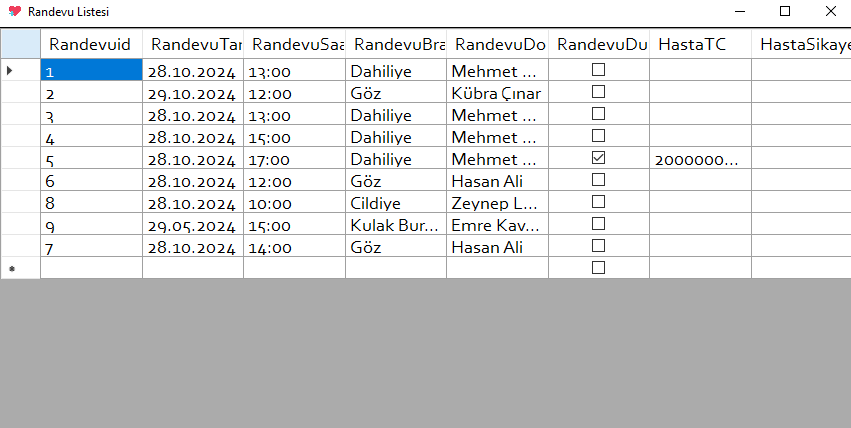
(Doktor Bilgi)



(Hasta Detay - Aktif Randevular)

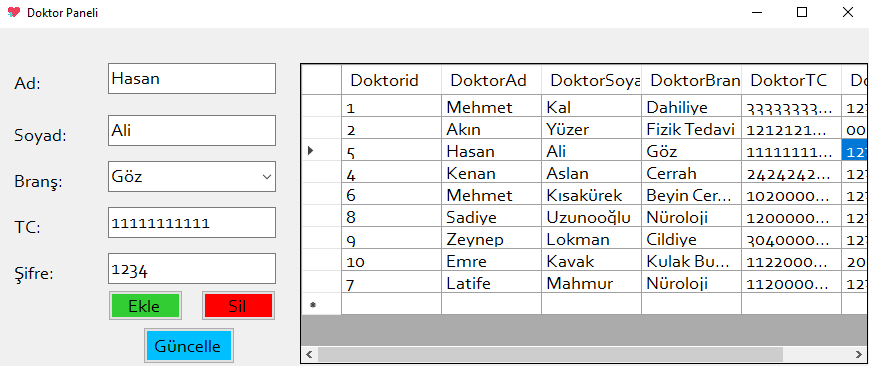


(Randevu Listesi)

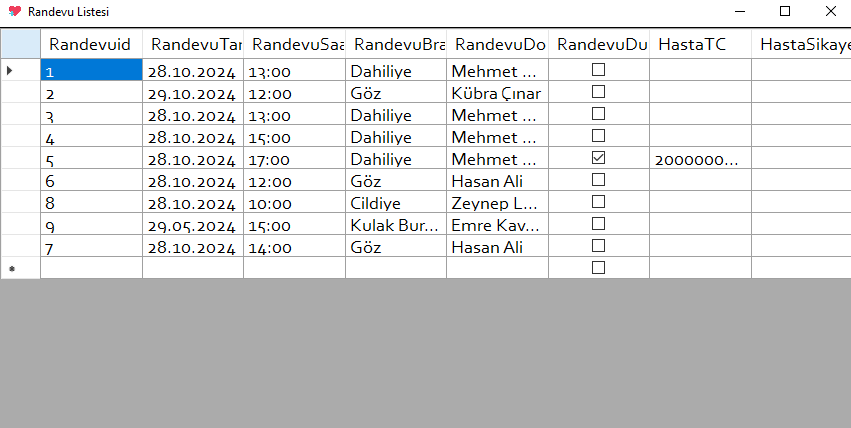


7: **Veriler datagrid üzerinden işlem yapılabilecek (ekle, sil, güncelle, listele)**

(Doktor Yönetimi)

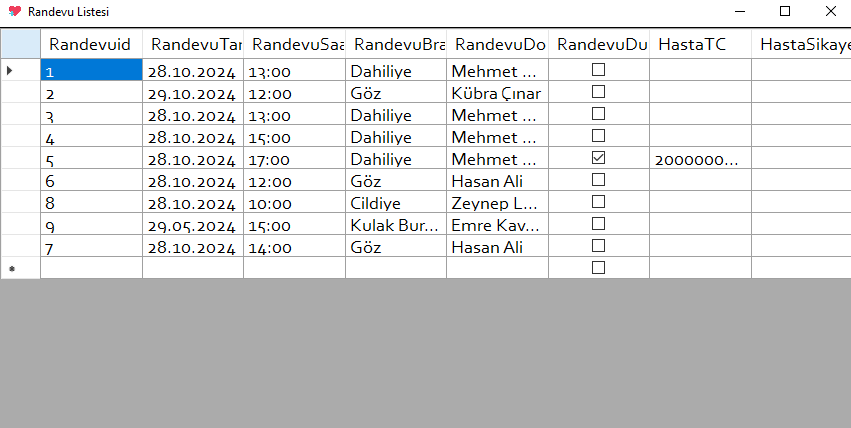


(Randevu Listesi)



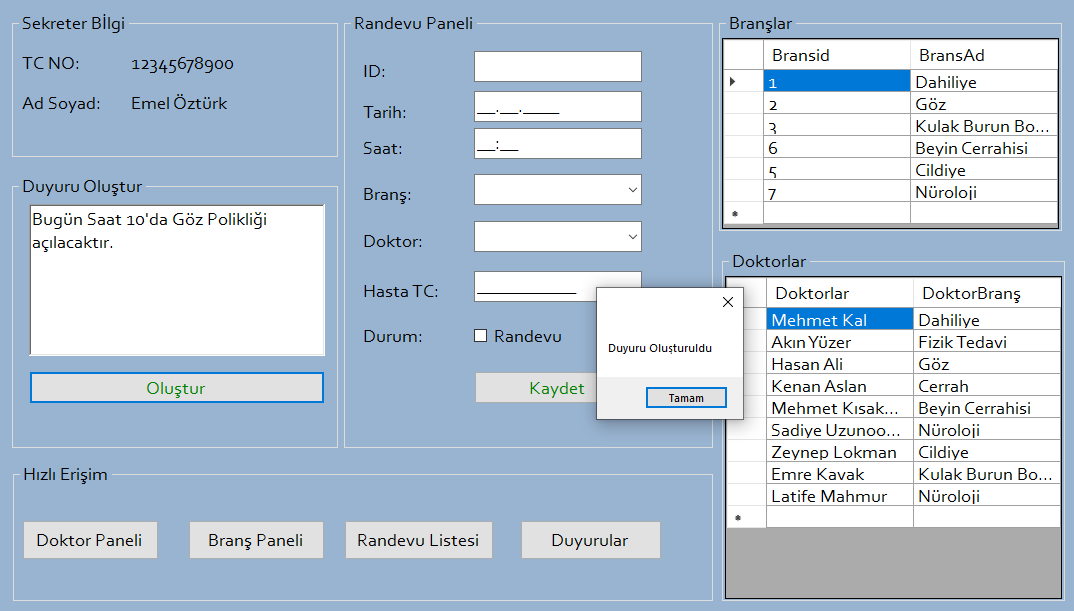
8: Matematiksel formüller kullanılarak hesaplamalar olmalı

(Randevu Listesi)



9: Form girişinde fotograf yükleme olacak

(Hasta Kayıt)



### **3. Veri Tabanı Tabloları**

#### **Tabloların Yapısı (Tablo Adları ve Alanları)**

* **Hasta Tablosu:**
  + Alanlar:
    - HastaID (Primary Key): Hasta’nın benzersiz kimliği.
    - Ad: Hastanın adı.
    - Soyad: Hastanın soyadı.
    - TCKimlikNo: 11 haneli T.C. Kimlik Numarası.
    - Telefon: 10 haneli telefon numarası.
    - Cinsiyet: Hastanın cinsiyeti.
* **Doktor Tablosu:**
  + Alanlar:
    - DoktorID (Primary Key): Doktor’un benzersiz kimliği.
    - Ad: Doktorun adı.
    - Soyad: Doktorun soyadı.
    - Branş: Doktorun branşı.
    - TCKimlikNo: 11 haneli T.C. Kimlik Numarası.
* **Randevu Tablosu:**
  + Alanlar:
    - RandevuID (Primary Key): Randevunun benzersiz kimliği.
    - HastaID (Foreign Key): Hastanın kimliği.
    - DoktorID (Foreign Key): Doktorun kimliği.
    - Tarih: Randevu tarihi.
    - Saat: Randevu saati.
    - Şikayet: Hastanın şikayeti.

#### **Alan Tipleri ve İlişkiler**

* **Hasta ve Randevu Tabloları Arasında İlişki:**
  + HastaID alanı üzerinden bire-bir ilişki.
* **Doktor ve Randevu Tabloları Arasında İlişki:**
  + DoktorID alanı üzerinden bire-bir ilişki.

#### **SQL ile Tabloların Oluşturulması**

-- Hasta Tablosunun Oluşturulması

CREATE TABLE Hasta (

HastaID INT PRIMARY KEY, -- Benzersiz hasta kimliği

Ad NVARCHAR(50), -- Hastanın adı

Soyad NVARCHAR(50), -- Hastanın soyadı

TCKimlikNo CHAR(11), -- 11 haneli T.C. Kimlik numarası

Telefon CHAR(10), -- 10 haneli telefon numarası

Cinsiyet NVARCHAR(10) -- Hastanın cinsiyeti

);

-- Doktor Tablosunun Oluşturulması

CREATE TABLE Doktor (

DoktorID INT PRIMARY KEY, -- Benzersiz doktor kimliği

Ad NVARCHAR(50), -- Doktorun adı

Soyad NVARCHAR(50), -- Doktorun soyadı

Branş NVARCHAR(50), -- Doktorun uzmanlık alanı

TCKimlikNo CHAR(11) -- 11 haneli T.C. Kimlik numarası

);

-- Randevu Tablosunun Oluşturulması

CREATE TABLE Randevu (

RandevuID INT PRIMARY KEY, -- Benzersiz randevu kimliği

HastaID INT, -- Hasta kimliği (Foreign Key)

DoktorID INT, -- Doktor kimliği (Foreign Key)

Tarih DATE, -- Randevu tarihi

Saat TIME, -- Randevu saati

Şikayet NVARCHAR(250), -- Hastanın şikayeti

FOREIGN KEY (HastaID) REFERENCES Hasta(HastaID),

FOREIGN KEY (DoktorID) REFERENCES Doktor(DoktorID)

);

#### **Veritabanı Görselleri ve Örnek Veriler**

* Hasta Tablosu:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **HastaID** | **Ad** | **Soyad** | **TCKimlikNo** | **Telefon** | **Cinsiyet** |
| 1 | Ahmet | Kaya | 12345678901 | 5551234567 | Erkek |
| 2 | Ayşe | Yılmaz | 98765432109 | 5559876543 | Kadın |

* Doktor Tablosu:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DoktorID | Ad | Soyad | Branş | TCKimlikNo |
| 1 | Mehmet | Kal | Dahiliye | 33333333333 |
| 2 | Hasan | Ali | Göz | 11111111111 |

* Randevu Tablosu:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| RandevuID | HastaID | DoktorID | Tarih | Saat | Şikayet |
| 1 | 1 | 1 | 2024-12-29 | 10:00 | Baş ağrısı |
| 2 | 2 | 2 | 2024-12-30 | 15:30 | Göz rahatsızlığı |

### **4. Kodlar ve Açıklamalar**

#### **Kodun Genel Mimarisi**

Proje, hastane otomasyon sistemini desteklemek için katmanlı bir mimari kullanılarak tasarlanmıştır:

* **Arayüz Katmanı (UI):** Kullanıcı ekranlarını kapsar (Windows Forms).
* **İş Mantığı Katmanı (Business Logic):** Doğrulama ve iş kuralları burada yer alır.
* **Veri Katmanı (Data Layer):** SQL Server ile veri saklama ve erişim sağlanır.

#### **Fonksiyonlar ve İşlevleri**

1. **Kullanıcı Girişi:** TC Kimlik No ve şifre ile giriş.
2. **Randevu Sistemi:** Randevu alma, iptal ve güncelleme.
3. **Duyuru Yayını:** Sekreterler için duyuru oluşturma.
4. **Veri Girişi ve Güncelleme:** Hasta, doktor ve randevu bilgilerini düzenleme.

#### **Kullanılan Teknolojiler ve Metotlar**

* **Programlama Dili:** C#
* **Veritabanı:** SQL Server
* **Arayüz:** Windows Forms

#### **Kodun Örnek Bölümleri**

**1. Kullanıcı Girişi:**

SqlCommand komut = new SqlCommand("SELECT \* FROM Hasta WHERE TCKimlikNo=@p1 AND Sifre=@p2", baglanti.baglanti());

komut.Parameters.AddWithValue("@p1", TxtTC.Text);

komut.Parameters.AddWithValue("@p2", TxtSifre.Text);

SqlDataReader dr = komut.ExecuteReader();

if (dr.Read())

{

FrmHastaDetay fr = new FrmHastaDetay();

fr.tc = TxtTC.Text;

fr.Show();

this.Hide();

}

else

{

MessageBox.Show("Hatalı TC veya Şifre");

}

**2. Randevu Oluşturma:**

komut = new SqlCommand("INSERT INTO Randevu (Tarih, Saat, DoktorID, HastaID, Sikayet) VALUES (@p1, @p2, @p3, @p4, @p5)", baglanti.baglanti());

komut.Parameters.AddWithValue("@p1", MskTarih.Text);

komut.Parameters.AddWithValue("@p2", MskSaat.Text);

komut.Parameters.AddWithValue("@p3", CmbDoktor.SelectedValue);

komut.Parameters.AddWithValue("@p4", TxtHastaID.Text);

komut.Parameters.AddWithValue("@p5", RchSikayet.Text);

komut.ExecuteNonQuery();

MessageBox.Show("Randevu Başarıyla Oluşturuldu");