

HASTANE RANDEVU OTOMASYONU

Kullanılan Uygulama Versiyonları

- Visual Studio 2019 sürüm 16.10.0
- Microsoft SQL Server Management Studio 18

KABULLER

Not: Hastane randevu sistemimiz 1 Hastane olarak belirlenmiştir.

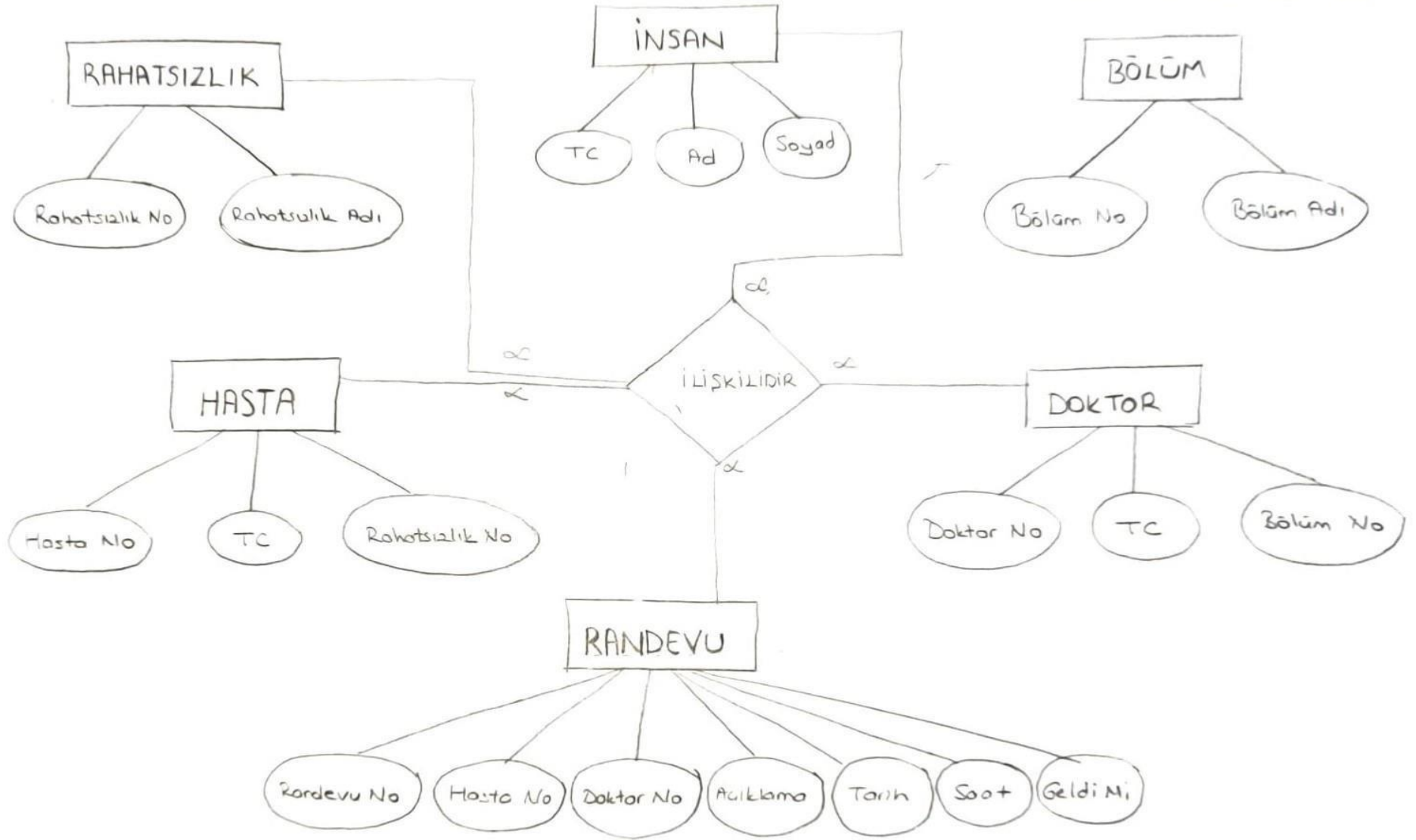
Doktor:

- Bir doktorun gün içinde en fazla 20 hastası olabilir.
- Bir doktor sadece bir bölüme kayıtlı olabilir.
- Doktorda hasta olabilir.

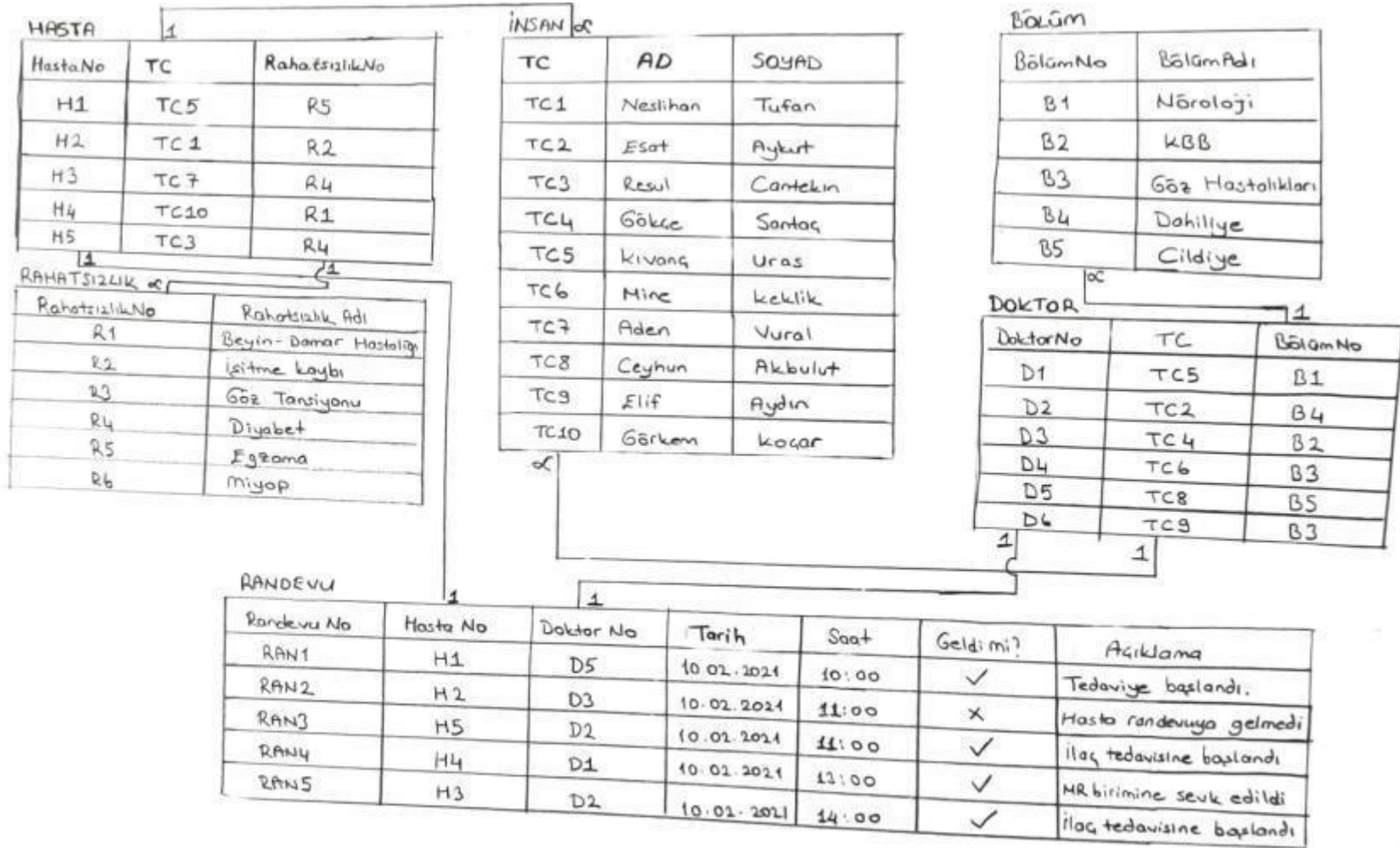
Hasta:

- Bir hasta bir gün içinde en fazla 3 randevu alabilir ancak farklı doktorlardan olmak şartıyla.
- Hasta randevuya 3 defa gelmez ise bir daha randevu alamaz.

ER DİYAGRAMI



TABLÖLAR



EKRAN GÖRÜNTÜLERİ

RANDEVU ALMA SENARYOSU

Hasta giriş penceresinden giriş yapar.



Hasta Giriş ekranı. Sol üstte geri dönüş butonu, sağ üstte kapatma butonu. Ortada 'HASTA GİRİŞ' başlığı. Sol tarafta hasta karakteri, sağ tarafta doktor karakteri. Giriş alanları: Hasta TC: 1234, Şifre: ****, Göster butonu, Giriş butonu.

Hasta Randevu Ekranına giriş yaptıktan sonra Randevu Listesini ve Bilgilerini görebilir. Randevu seçme kısmından Bölüm, Doktor, Tarih ve Saati seçerek Randevusunu oluşturur.



Hasta Randevu Ekranı. Sol üstte geri dönüş butonu, sağ üstte kapatma butonu. Ortada 'Hoşgeldin Melisa' başlığı. Sol tarafta 'BİLGİLERİN' başlığı altında hasta bilgileri: Ad: Melisa, Soyad: Türkmen, TC: 1234, Şifre: 1234, Ceza: 0. Sağ tarafta 'ALDIĞIN RANDEVU LİSTESİ' başlığı altında randevu listesi. Alt kısımda 'RANDEVU ALMA İŞLEMİ' başlığı altında randevu seçme alanları: Bölüm, Doktor, Randevu Tarihi, Randevu Saati, Randevu Al butonu.

Randevu No	Bölüm Adı	Doktor Adı	Doktor Soyadı	Tarih	Saat
81	Cildiye	sg	sdsgsd	29.04.2021	14:20
82	Göz	Alex	Güneş	29.04.2021	09:00
84	Kardiyoloji	Rümeysa	Güneş	27.05.2021	14:00
85	Nöroloji	Hasan	BAŞKURT	26.05.2021	12:00
86	Kardiyoloji	Meryem	ÇİM	26.05.2021	13:00

RANDEVU ALMA İŞLEMİ

Bölüm : Göz
Doktor : Zehra ALKAN
Randevu Tarihi : 26.05.2021
Randevu Saati : 14:20

Randevu Al

Randevu saatini üzerine tıklayıp seçebilirsiniz.

Randevu Al Butonuna tıkladığında randevu işleminin gerçekleştiğini görür.

Hoşgeldin Melisa

BİLGİLERİN

Ad : **Melisa**

Soyad : **Türkmen**

TC : **1234**

Şifre : **1234**

Ceza : **0**

ALDIĞIN RANDEVU LİSTESİ

Randevu No	Bölüm Adı	Doktor Adı	Doktor Soyadı	Tarih	Saat
1	Nöroloji	MERT	Güneş	24.04.2021	12:00
2	Göz	Alex	Güneş	27.04.2021	13:00
3	Nöroloji	MERT	Güneş	25.04.2021	12:00
4	Dahiliye	Selin	YILMAZ	25.04.2021	12:00
5	Cildiye	sa	sdgsd	25.04.2021	12:00
6				25.04.2021	14:20

DEVU ALMA İŞLEMİ

Bölüm :

Doktor :

Randevu Tarihi : 26.05.2021

Randevu Saati :

Randevu Al

RANDEVU İPTAL ETME

Randevu No:

İptal Et

Kayıt Eklendi

Tamam

09:00 09:20 09:40 10:00

10:20 10:40 11:00 11:20

12:00 13:00 13:20 13:40

14:00 14:20 14:40 15:00

Randevu saatini üzerine tıklayıp seçebilirsiniz.

Hasta Randevu aldıktan sonra bu bilgi doktora iletilir.

Doktor TC ve şifresini girerek giriş yapar.

DOKTOR GİRİŞ

Doktor TC:

Şifre: **Göster**

Giriş

Doktor giriş yaptıktan sonra randevu oluşturan hastalarını görebilir. Randevu listesinden hastayı seçerek muayene bilgisini ve hastanın randevuya gelip gelmediğini günceller.

Hoşgeldin Zehra

Hasta Muayene İşlemi **Kaydet**

BİLGİLERİN

Ad : Zehra

Soyad : ALKAN

Bölüm : Göz

TC : 12

Şifre : 12

Adı Soyadı : Melisa Türkmen TC : 1234

Tarih : 26.05.2021 Saat : 14:20 Geldi mi ? : ☒

Açıklama :

RANDEVU LİSTESİ

Randevu No	Hasta Adı	Hasta Soyadı	TC	Tarih	Saat	Geldi mi ?
69	Melisa	Türkmen	1234	27.04.2021	14:00	Gelmedi !
87	Denek	denek	123	26.05.2021	12:00	Gelmedi !
88	Melisa	Türkmen	1234	26.05.2021	14:20	Gelmedi !

Güncellendikten sonra kayıt eklendi bilgisi doktora iletilir.

Hoşgeldin Zehra

Hasta Muayene İşlemi **Kaydet**

BİLGİLERİN

Ad : Zehra

Soyad : ALKAN

Bölüm : Göz

TC : 12

Şifre : 12

Adı Soyadı : TC :

Tarih : Saat : Geldi mi ? : ☐

Kayıt eklendi

Tamam

RANDEVU LİSTESİ

Randevu No	Hasta Adı	Hasta Soyadı	TC	Tarih	Saat	Geldi mi ?
69	Melisa	Türkmen	1234	27.04.2021	14:00	Gelmedi !
87	Denek	denek	123	26.05.2021	12:00	Gelmedi !
88	Melisa	Türkmen	1234	26.05.2021	14:20	Geldi

Bütün bu işlemler oluşturduğumuz veritabanına kaydedilir.

DESKTOP-PC77Q2I....dbo.Randevu_Table								
	Randevu_No	Hasta_Id	Bolum_Id	Doktor_Id	Tarih	Saat	Geldi_mi	Aciklama
	1	1	1	37	24.04.2021	12:00	True	NULL
	2	1	3	40	27.04.2021	13:00	NULL	NULL
	3	1	1	37	25.04.2021	12:00	NULL	NULL
	4	1	4	7	25.04.2021	12:00	NULL	NULL
	5	1	5	43	25.04.2021	12:00	NULL	NULL
	6	1	2	38	25.04.2021	14:20	NULL	NULL
	67	1	3	40	16.07.2021	14:00	NULL	NULL
	68	23	3	40	27.04.2021	14:00	NULL	NULL
	69	1	3	6	27.04.2021	14:00	NULL	NULL
	70	1	1	37	27.04.2021	10:20	NULL	NULL
	71	1	2	38	28.04.2021	11:20	NULL	NULL
	72	1	1	37	29.04.2021	11:20	NULL	NULL
	73	1	1	4	27.04.2021	13:20	True	6541610
	75	1	1	4	28.04.2021	12:00	True	ljbkjb
	76	1	1	4	29.04.2021	14:00	NULL	NULL
	78	1	2	5	30.04.2021	14:00	NULL	NULL
	80	1	4	7	29.04.2021	12:00	NULL	NULL
	81	1	5	43	29.04.2021	14:20	NULL	NULL
	82	1	3	40	29.04.2021	09:00	NULL	NULL
	83	23	5	8	28.04.2021	13:00	False	
	84	1	2	38	27.05.2021	14:00	NULL	NULL
	85	1	1	4	26.05.2021	12:00	NULL	NULL
	86	1	2	5	26.05.2021	13:00	NULL	NULL
	87	23	3	6	26.05.2021	12:00	NULL	NULL
	88	1	3	6	26.05.2021	14:20	True	Miyop başlang
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Randevu alan hastanın id bilgileri.

DESKTOP-PC77Q2I.H...- dbo.Hasta_Table					
	Hasta_Id	Ad	Soyad	TC	Sifre
	1	Melisa	Türkmen	1234	1234
	23	Denek	denek	123	123
*	NULL	NULL	NULL	NULL	NULL

Randevu alınan doktor id bilgileri.

DESKTOP-PC77Q2I.H...dbo.Doktor_Table						
	Doktor_Id	Ad	Soyad	TC	Bolum_Id	Sifre
	4	Hasan	BAŞKURT	1234	1	1234
	5	Meryem	ÇİM	123	2	123
	6	Zehra	ALKAN	12	3	12
	7	Selin	YILMAZ	9632	4	123
	8	Sadullah	DİNÇ	7412	5	123
	37	MERT	Güneş	56789	1	12
	38	Rümeysa	Güneş	8941	2	12
	39	Kazım	Güneş	5678	5	12
	40	Alex	Güneş	4567	3	12
	43	sg	sdgsgd	35235	5	11
*	NULL	NULL	NULL	NULL	NULL	NULL

Randevu alınan bölüm id bilgileri.

DESKTOP-PC77Q2I....- dbo.Bolum_Table		
	Bolum_Id	Bolum_Ad
	1	Nöroloji
	2	Kardiyoloji
	3	Göz
	4	Dahiliye
	5	Cildiye
*	NULL	NULL

SCRIPT


```

USE [Hastane]
GO
/***** Object: Table [dbo].[Bolum_Table]    Script Date: 21.04.2021 11:06:14 *****/ SET
ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Bolum_Table](
    [Bolum_No] [nvarchar](50) NOT NULL,
    [Bolum_Adi] [nvarchar](50) NULL,
    CONSTRAINT [PK_Bolum_Table] PRIMARY KEY CLUSTERED
(
    [Bolum_No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Doktor_Table]    Script Date: 21.04.2021 11:06:14 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Doktor_Table](
    [Doktor_No] [nvarchar](50) NOT NULL,
    [TC] [nvarchar](50) NOT NULL,
    [Bolum_No] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Doktor_Table] PRIMARY KEY CLUSTERED
(
    [Doktor_No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Hasta_Table]    Script Date: 21.04.2021 11:06:14 *****/ SET
ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Hasta_Table](
    [Hasta_No] [nvarchar](50) NOT NULL,
    [TC] [nvarchar](50) NOT NULL,
    [Rahatsizlik_No] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Hasta_Table] PRIMARY KEY CLUSTERED
(
    [Hasta_No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY] )
ON [PRIMARY]
GO
/***** Object: Table [dbo].[Insan_Table]    Script Date: 21.04.2021 11:06:14 *****/ SET
ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Insan_Table](

```

```

        [TC] [nvarchar](50) NOT NULL,
        [Isim] [nvarchar](50) NULL,
        [Soyisim] [nvarchar](50) NULL,
    CONSTRAINT [PK_Insan_Table] PRIMARY KEY CLUSTERED
(
    [TC] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Rahatsizlik_Table]    Script Date: 21.04.2021 11:06:14
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Rahatsizlik_Table](
    [Rahatsizlik_No] [nvarchar](50) NOT NULL,
    [Rahatsizlik_Adi] [nvarchar](50) NULL,
    CONSTRAINT [PK_Rahatsizlik_Table] PRIMARY KEY CLUSTERED
(
    [Rahatsizlik_No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Randevu_Table]    Script Date: 21.04.2021 11:06:14 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Randevu_Table](
    [Randevu_No] [nvarchar](50) NOT NULL,
    [Hasta_No] [nvarchar](50) NULL,
    [Doktor_No] [nvarchar](50) NULL,
    [Tarih] [datetime] NULL,
    [Saat] [time](7) NULL,
    [Geldi/Gelmedi] [bit] NULL,
    [Açıklama] [nvarchar](50) NULL,
    CONSTRAINT [PK_Randevu_Table] PRIMARY KEY CLUSTERED
(
    [Randevu_No] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] GO
ALTER TABLE [dbo].[Doktor_Table] WITH CHECK ADD CONSTRAINT [FK_Doktor_Table_Bolum_Table]
FOREIGN KEY([Bolum_No])
REFERENCES [dbo].[Bolum_Table] ([Bolum_No])
GO
ALTER TABLE [dbo].[Doktor_Table] CHECK CONSTRAINT [FK_Doktor_Table_Bolum_Table] GO
ALTER TABLE [dbo].[Doktor_Table] WITH CHECK ADD CONSTRAINT [FK_Doktor_Table_Insan_Table]
FOREIGN KEY([TC])
REFERENCES [dbo].[Insan_Table] ([TC])
GO

```

```
ALTER TABLE [dbo].[Doktor_Table] CHECK CONSTRAINT [FK_Doktor_Table_Insan_Table] GO
ALTER TABLE [dbo].[Hasta_Table] WITH CHECK ADD CONSTRAINT [FK_Hasta_Table_Insan_Table]
FOREIGN KEY([TC])
REFERENCES [dbo].[Insan_Table] ([TC])
GO
ALTER TABLE [dbo].[Hasta_Table] CHECK CONSTRAINT [FK_Hasta_Table_Insan_Table]
GO
ALTER TABLE [dbo].[Hasta_Table] WITH CHECK ADD CONSTRAINT
[FK_Hasta_Table_Rahatsizlik_Table] FOREIGN KEY([Rahatsizlik_No])
REFERENCES [dbo].[Rahatsizlik_Table] ([Rahatsizlik_No])
GO
ALTER TABLE [dbo].[Hasta_Table] CHECK CONSTRAINT [FK_Hasta_Table_Rahatsizlik_Table] GO
ALTER TABLE [dbo].[Randevu_Table] WITH CHECK ADD CONSTRAINT
[FK_Randevu_Table_Doktor_Table] FOREIGN KEY([Doktor_No])
REFERENCES [dbo].[Doktor_Table] ([Doktor_No])
GO
ALTER TABLE [dbo].[Randevu_Table] CHECK CONSTRAINT [FK_Randevu_Table_Doktor_Table]
GO
ALTER TABLE [dbo].[Randevu_Table] WITH CHECK ADD CONSTRAINT
[FK_Randevu_Table_Hasta_Table] FOREIGN KEY([Hasta_No])
REFERENCES [dbo].[Hasta_Table] ([Hasta_No])
GO
ALTER TABLE [dbo].[Randevu_Table] CHECK CONSTRAINT [FK_Randevu_Table_Hasta_Table]
GO
```

PROGRAM KAYNAK KODLARI

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq; using
System.Threading.Tasks; using
System.Windows.Forms;

namespace HastaneRandevuSistemi
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread] static
        void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new GiriFormu());
        }
    }
}
```

GirisFormu.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HastaneRandevuSistemi
{
    public partial class GiriFormu :
    Form
    {
        public GiriFormu()
        {
            InitializeComponent();
        }
        private void pictureBox3_Click(object sender,
        EventArgs e)
        {
            System.Windows.Forms.Application.Exit();
        }
        private void btnGorevliGiris_Click(object sender, EventArgs e)
        {
            Yoneticigiris grvGiris = new Yoneticigiris(); // Yoneticigiris formundan
            nesne oluřturuldu
            this.Hide(); // Bulunan form
            kapatıldı
        }
    }
}
```

```

        grvGiris.Show(); //Yönetici giris formu
açıldı }
        private void btnHastaLogin_Click(object sender, EventArgs e)
        {
            HastaGiris hastaGiris = new HastaGiris();
this.Hide();
hastaGiris.Show();
        }
        private void btnDoktorIslem_Click(object sender, EventArgs e)
        {
            DoktorGiris drGiris = new DoktorGiris(); // Doktor giris formundan nesne
oluşturuldu
            this.Hide(); // Bulunan form
kapatıldı
            drGiris.Show();
        }
    }
}

```

YoneticiGiris.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class YoneticiGiris :
Form
    {
        public YoneticiGiris()
        {
            InitializeComponent();

            // Veri Tabanı Bağlantısı
            SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");

            private void pictureBox3_Click(object sender, EventArgs e)
            {
                System.Windows.Forms.Application.Exit();
            }
            private void button2_Click(object sender, EventArgs e)
        {
            GirisFormu baslangic = new GirisFormu(); //Geçiş yapılacak formdan nesne
oluşturuldu.
            this.Hide(); //Şuan ki form ekranı
kapatıldı.
            baslangic.Show();
        }
    }
}

```



```

        // Sorgu Fonksiyonu
public int sorgu()
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select *From Yonetici_Table where TC=("
+ txtGorevliTc.Text.ToString() + ") and Sifre=(" + txtGorevliSifre.Text.ToString() +
")", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    int sonuc = 0;          int sayac = 0;
    while (oku.Read())
    {
        sayac++;

    }

    if (sayac > 0)
    {
        sonuc = 1;
    }
    baglanti.Close();
return sonuc;

    }

    // Giris islemi
private void btnYoneticiGiris_Click(object sender, EventArgs e)
{
    if (txtGorevliTc.Text != "" && txtGorevliSifre.Text != "") // Textler boş
olup olmadığı kontrol edildi
    {

        if (sorgu() == 1) //Girilen değerler veri tabanındaki tabloda mevcut
ise işlem gerçekleşti
        {
            MessageBox.Show("Giriş başarılı");
            Yonetici_Islem gorevliGecis = new Yonetici_Islem(); // Gorevli
geçiş formundan nesne üretildi
            this.Hide(); //
            Bulunduğumuz fonksiyon kapatıldı
            gorevliGecis.Show(); //
            Oluşturulan nesnenin formu açıldı
        }

        else
        {
            MessageBox.Show("Hatalı giriş");
        }

    }

    else
    {
        MessageBox.Show("Gerekli alanları doldurunuz !");
    }
}
private void YoneticiGiris_Load(object sender, EventArgs e)
{
}
}

```

```

        // Veri gizleme
        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            if (checkBox1.CheckState == CheckState.Checked) //CheckBox seçili ise
                parolayı goster text e gizle yaz
            {
                txtGorevliSifre.UseSystemPasswordChar = true;
                checkBox1.Text = "Gizle";
            }
            else if (checkBox1.CheckState == CheckState.Unchecked) //CheckBox seçili
                değil ise parolayı gizle ve text e göster yaz
            {
                txtGorevliSifre.UseSystemPasswordChar = false;
                checkBox1.Text = "Göster";
            }
        }

        //Harf Girisi Engelleme
        private void txtGorevliTc_KeyPress(object sender, KeyPressEventArgs e)
        {
            e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
        }
        private void txtGorevliSifre_KeyPress(object sender, KeyPressEventArgs e)
        {
            e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
        }
    }
}

```

Yonetici_Islem.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HastaneRandevuSistemi
{
    public partial class Yonetici_Islem :
        Form
    {
        public Yonetici_Islem()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            YoneticiGiris yonetici = new YoneticiGiris();
            this.Hide();
            yonetici.Show();
        }
        private void pictureBox3_Click(object sender, EventArgs e)
        {
            System.Windows.Forms.Application.Exit();
        }
    }
}

```

```

    }
    private void btnDoktorIslem_Click(object sender, EventArgs e)
    {
        Doktor_Islemleri doktor = new Doktor_Islemleri();
        this.Hide();
        doktor.Show();
    }
    private void btnHastaIslem_Click(object sender, EventArgs e)
    {
        Hasta_Islemleri hasta = new Hasta_Islemleri();
        this.Hide();
        hasta.Show();
    }
}

```

Hasta_Islemleri.cs

```

using
System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class Hasta_Islemleri :
Form
    {
        public Hasta_Islemleri()
        {
            InitializeComponent();

            // Veri Tabanı Bağlantısı
            SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");

            // Bir onceki sayfaya gelme Butonu
            private void button2_Click(object sender, EventArgs e)
            {
                Yonetici_Islem yonetici = new Yonetici_Islem(); this.Hide();
                yonetici.Show();
            }

            // Kapatma Butonu
            private void pictureBox3_Click(object sender, EventArgs e)
            {
                System.Windows.Forms.Application.Exit();
            }

            //Form Baslangic Kismi
            private void Hasta_Islemleri_Load(object sender, EventArgs e)
            {

```

```

        veriliste();
    }
    private void listele_Click(object sender, EventArgs e)
    {

    }

    //Listeleme Fonksiyonu
    private void veriliste()
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Select *From Hasta_Table", baglanti);
        SqlDataReader oku = komut.ExecuteReader();

        while (oku.Read())
        {
            ListViewItem ekle = new ListViewItem();
            ekle.Text = oku["Hasta_Id"].ToString();
            ekle.SubItems.Add(oku["Ad"].ToString());
            ekle.SubItems.Add(oku["Soyad"].ToString());
            ekle.SubItems.Add(oku["TC"].ToString());
            ekle.SubItems.Add(oku["Sifre"].ToString());
            hastaListe.Items.Add(ekle);

        }
        baglanti.Close();
    }

    // Sorgu Fonksiyonu
    public int sorgu()
    {
        int id = int.Parse(txt_IdSil.Text);
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Select *From Hasta_Table where
Hasta_Id=(" + id + ")", baglanti);
        SqlDataReader oku = komut.ExecuteReader();
        int sonuc = 0;          int sayac = 0;
        while (oku.Read())
        {
            sayac++;

        }

        if (sayac > 0)
        {
            sonuc = 1;
        }
        baglanti.Close();
        return sonuc;
    }

    // Ekleme Islemi
    private void btnEkle_Click(object sender, EventArgs e)
    {
        if (txtAd.Text != "" && txtSoyad.Text != "" && txtTc.Text != "" &&
txtSifre.Text != "")
        {

```

```

        baglanti.Open();
        SqlCommand komut = new SqlCommand("insert into Hasta_Table
(Ad,Soyad,TC,Sifre) values ('" + txtAd.Text.ToString() + "','" +
txtSoyad.Text.ToString() + "','" + txtTc.Text.ToString() + "','" +
txtSifre.Text.ToString() + "')", baglanti);
komut.ExecuteNonQuery();
        baglanti.Close();
        hastaListe.Items.Clear(); // ListBox Temizle
verilistele(); // Listeleme Islemi
txtAd.Clear(); txtSoyad.Clear();
txtTc.Clear(); txtSifre.Clear();
        MessageBox.Show("Kayıt Eklendi ");
    }

    else
        MessageBox.Show("Lütfen gerekli alanları doldurunuz !");
}

//Silme Islemi
private void btnSil_Click(object sender, EventArgs e)
{
    if (txt_IdSil.Text != "")
    {
        int id = int.Parse(txt_IdSil.Text);
if (sorgu() == 1)
        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("Delete From Hasta_Table where
Hasta_Id =( " + id + " )", baglanti);
            komut.ExecuteNonQuery();
            baglanti.Close();
            hastaListe.Items.Clear(); // ListBox Temizle
verilistele(); // Listeleme Islemi
            txt_IdSil.Clear();
            MessageBox.Show("Kayıt silindi");
        }
    else
        MessageBox.Show("Kayıt bulunamadı");
    }
    else
        MessageBox.Show("Lütfen id giriniz !");
}

//Guncelleme Islemi
private void btnGuncelle_Click(object sender, EventArgs e)
{
    if(id!=0)
    {
        if (txt_gAd.Text != "" && txt_gSoyad.Text != "" && txt_gTc.Text != ""
&& txt_gSifre.Text != "")
        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("update Hasta_Table set Ad =' " +
txt_gAd.Text.ToString() + "',Soyad =' " + txt_gSoyad.Text.ToString() + "',TC =' " +
txt_gTc.Text.ToString() + "',Sifre =' " + txt_gSifre.Text.ToString() + "' where

```



```

Hasta_Id=" + id + "", baglanti); komut.ExecuteNonQuery();
baglanti.Close();
        hastaListe.Items.Clear(); // ListBox Temizle
verilistele(); // Listeleme Islemi
txt_gAd.Clear(); txt_gSoyad.Clear();
txt_gTc.Clear(); txt_gSifre.Clear();
        MessageBox.Show("Kayıt güncellendi");
id = 0; } else
        MessageBox.Show("Lütfen gerekli alanları doldurunuz !");
    }
else
        MessageBox.Show("Lütfen listeden güncellemek istediğiniz hastayı seçiniz
!");
}

```

```

//Listeden Secme Islemi
int id = 0;
private void hastaListe_MouseClick(object sender, MouseEventArgs e)
{
    id = int.Parse(hastaListe.SelectedItems[0].SubItems[0].Text);
txt_gAd.Text = hastaListe.SelectedItems[0].SubItems[1].Text;
txt_gSoyad.Text = hastaListe.SelectedItems[0].SubItems[2].Text;
txt_gTc.Text = hastaListe.SelectedItems[0].SubItems[3].Text;
txt_gSifre.Text = hastaListe.SelectedItems[0].SubItems[4].Text;
}

// Harf girisi engelleme
private void txtTc_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txtSifre_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txt_IdSil_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txt_gTc_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txt_gSifre_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

// Sayi girisi engelleme
private void txtAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txtSoyad_KeyPress(object sender, KeyPressEventArgs e)
{

```

```

        e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
    }
    private void txt_gAd_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
    }
    private void txt_gSoyad_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
    }
}
}

```

Doktor_Islemleri.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class Doktor_Islemleri :
Form
    {
        // Veri Tabanı Bağlantısı
        SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");

        public Doktor_Islemleri()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Yonetici_Islem yonetici = new Yonetici_Islem();
this.Hide();          yonetici.Show();
        }
        private void pictureBox3_Click(object sender, EventArgs e)
        {
            System.Windows.Forms.Application.Exit();
        }
        private void Doktor_Islemleri_Load(object sender, EventArgs e)
        {
            verilistele();
            lblListe.Text = "DOKTOR BİLGİ LİSTESİ";
            lblBilgi.ForeColor = Color.Yellow;
            lblMuayene.ForeColor = Color.White;
        }
    }
}

```

```

// Muayene Listele
private void muayeneListele()
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select
r.Randevu_No,h.Ad,h.Soyad,h.TC,r.Tarih,r.Saat,r.Geldi_mi,r.Aciklama from
((Doktor_Table d inner join Randevu_Table r on d.Doktor_Id=r.Doktor_Id) inner join
Hasta_Table h on h.Hasta_Id=r.Hasta_Id) where r.Doktor_Id='"+ id + "'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    doktorListe.Clear();

    doktorListe.Columns.Add("Randevu_No",100);
    doktorListe.Columns.Add("Hasta_Ad", 100);
    doktorListe.Columns.Add("Hasta_Soyad", 100);
    doktorListe.Columns.Add("TC", 100);
    doktorListe.Columns.Add("Tarih", 100);
    doktorListe.Columns.Add("Saat", 100);
    doktorListe.Columns.Add("Geldi_mi", 100);
    doktorListe.Columns.Add("Aciklama", 100);
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Randevu_No"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TC"].ToString());
        ekle.SubItems.Add(oku["Tarih"].ToString());
        ekle.SubItems.Add(oku["Saat"].ToString());
        ekle.SubItems.Add(oku["Geldi_mi"].ToString());
        ekle.SubItems.Add(oku["Aciklama"].ToString());
        doktorListe.Items.Add(ekle);
    }
    baglanti.Close();
}
//Listeleme Fonksiyonu
private void veriliste()
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select
d.Doktor_Id,d.Ad,d.Soyad,d.TC,b.Bolum_Ad from (Doktor_Table d inner join Bolum_Table b
on d.Bolum_Id=b.Bolum_Id)", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    doktorListe.Clear();

    doktorListe.Columns.Add("Doktor_Id", 100);
    doktorListe.Columns.Add("Ad", 100);
    doktorListe.Columns.Add("Soyad", 100);
    doktorListe.Columns.Add("TC", 100);
    doktorListe.Columns.Add("Bolum_Ad", 100);
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Doktor_Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TC"].ToString());
        ekle.SubItems.Add(oku["Bolum_Ad"].ToString());
        doktorListe.Items.Add(ekle);
    }
}

```

```

    }
    baglanti.Close();
}

// Sorgu Fonksiyonu
public int sorgu()
{
    int id = int.Parse(txt_IdSil.Text);
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select *From Doktor_Table where
Doktor_Id=(" + id + ")", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    int sonuc = 0;          int sayac = 0;
    while (oku.Read())
    {
        sayac++;
    }
    if (sayac >
0)
    {
        sonuc = 1;
    }
    baglanti.Close();
    return sonuc;
}
int deger;
//Bolum Id private int
bolumId(string combo)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select *From Bolum_Table where
Bolum_Ad='" + combo + "'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();

    while (oku.Read())
    {
        deger = int.Parse(oku["Bolum_Id"].ToString());
    }
    baglanti.Close();
    return deger;
}
//Ekleme Islemi
int veri;
private void btnEkle_Click(object sender, EventArgs e)
{
    if (txtAd.Text != "" && txtSoyad.Text != "" && txtTc.Text != "" &&
comboBoxBolum.Text != "")
    {
        veri = bolumId(comboBoxBolum.Text);
        baglanti.Open();
        SqlCommand komut = new SqlCommand("insert into Doktor_Table
(Ad,Soyad,TC,Bolum_Id) values ('" + txtAd.Text.ToString() + "','" +
txtSoyad.Text.ToString() + "','" + txtTc.Text.ToString() + "','" + veri + "'",
baglanti);

```

```

        komut.ExecuteNonQuery();
baglanti.Close();
        doktorListe.Items.Clear(); // ListBox Temizle
verilistele(); // Listeleme Islemi-
txtAd.Clear(); txtSoyad.Clear();
txtTc.Clear();
        comboBoxBolum.Text = null;
MessageBox.Show("Kayıt Eklendi ");
    }

    else
        MessageBox.Show("Lütfen gerekli alanları doldurunuz !");
}

// sayı engelleme
private void txtAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txtSoyad_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txt_gAd_KeyPress(object sender,
KeyPressEventArgs e) {
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txt_gSayfa_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}

//Harf girisi engelleme
private void txtTc_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txt_gYazar_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txt_IdSil_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

//Silme Islemi
private void btnSil_Click(object sender, EventArgs e)
{
    if (txt_IdSil.Text != "")
    {
        int id = int.Parse(txt_IdSil.Text);
if (sorgu() == 1)

```



```

        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("Delete From Doktor_Table where
Doktor_Id =(" + id + ")", baglanti);
            komut.ExecuteNonQuery();
            baglanti.Close();

            doktorListe.Items.Clear(); // ListBox Temizle
            veriliste(); // Listeleme Islemi
            txt_IdSil.Clear();
            MessageBox.Show("Kayıt silindi");
        }
    else
        MessageBox.Show("Kayıt bulunamadı");
    }
else
    MessageBox.Show("Lütfen id giriniz !");
}

//Listeden Secme Islemi
int id = 0;
private void doktorListe_MouseClick(object sender, MouseEventArgs e)
{
    if(lblListe.Text != "MUAYENE BİLGİ LİSTESİ")
    {
        id = int.Parse(doktorListe.SelectedItems[0].SubItems[0].Text);
        txt_gAd.Text = doktorListe.SelectedItems[0].SubItems[1].Text;
        txt_gSoyad.Text = doktorListe.SelectedItems[0].SubItems[2].Text;
        txt_gTc.Text = doktorListe.SelectedItems[0].SubItems[3].Text;
        comboBox_gBolum.Text = doktorListe.SelectedItems[0].SubItems[4].Text;
    }
}

//Guncelleme Islemi
private void btnGuncelle_Click(object sender, EventArgs e)
{
    veri = bolumId(comboBox_gBolum.Text);
    if (id != 0)
    {
        if (txt_gAd.Text != "" && txt_gSoyad.Text != "" && txt_gTc.Text != ""
&& comboBox_gBolum.Text != "")
        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("update Doktor_Table set Ad ='
+ txt_gAd.Text.ToString() + ',' ,Soyad =' + txt_gSoyad.Text.ToString() + ',' ,TC =' +
txt_gTc.Text.ToString() + ',' ,Bolum_Id =' + veri + "' where Doktor_Id=" + id + "'",
            baglanti);
            komut.ExecuteNonQuery();
            baglanti.Close();

            doktorListe.Items.Clear(); // ListBox Temizle
            veriliste(); // Listeleme Islemi
            txt_gAd.Clear();
            txt_gSoyad.Clear();
            txt_gTc.Clear();
            comboBox_gBolum.Text = null;
            MessageBox.Show("Kayıt güncellendi");
            id = 0;
        }
        else
            MessageBox.Show("Lütfen gerekli alanları doldurunuz !");
    }
else

```

```

        MessageBox.Show("Lütfen listeden güncellemek istediğiniz doktoru seçiniz
!");
    }

    //Muayene bilgisi click
    private void pictureBoxIadeBilgi_MouseClick(object sender, MouseEventArgs e)
    {
    }

    //Doktor bilgi click
    private void pictureBoxKitapBilgi_Click(object sender, EventArgs e)
    {
        veriliste();
        lblListe.Text = "DOKTOR BİLGİ LİSTESİ"; // Labeldaki yazı değişti
        lblBilgi.ForeColor = Color.Yellow; // İade label i pasif rengi aldı
        lblMuayene.ForeColor = Color.White;
        lblBilgilendirme.Text = "Not: Muayene bilgisini öğrenmek istediğiniz
doktorun listeden üzerine tıklayıp iade bilgisi butonuna basınız.";
    }
    private void pictureBoxMuayeneBilgi_Click(object sender, EventArgs e)
    {
        if (id != 0)
        {
            muayeneListele();
            lblListe.Text = "MUAYENE BİLGİ LİSTESİ";
            lblBilgi.ForeColor = Color.White; lblMuayene.ForeColor
            = Color.Yellow;
            lblBilgilendirme.Text = "Not: Doktor bilgilerine geri dönmek için doktor
bilgisi butonuna basınız.";
        }
    }
    else
        MessageBox.Show("Lütfen doktor seçiniz !");

    }
}
}

```

HastaGiris.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class HastaGiris : Form
    {
        public HastaGiris()
    }
}

```

```

    {
        InitializeComponent();
    }

    // Veri Tabanı Bağlantısı
    SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");

    // Sorgu Fonksiyonu
    public int sorgu()
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Select *From Hasta_Table where TC=(" +
txtTc.Text.ToString() + ") and Sifre=(" + txtSifre.Text.ToString() + ")", baglanti);
        SqlDataReader oku = komut.ExecuteReader();
        int sonuc = 0;          int sayac = 0;
        while (oku.Read())
        {
            sayac++;
            lblHastaId.Text= oku["Hasta_Id"].ToString();
        }

        if (sayac > 0)
        {
            sonuc = 1;
        }
        baglanti.Close();
        return sonuc;
    }

    private void button2_Click(object sender, EventArgs e)
    {
        GirisFormu baslangic = new GirisFormu(); //Geçiş yapılacak formdan nesne
        oluşturuldu.
        this.Hide(); //Şuan ki form ekranı
        kapatıldı. baslangic.Show();
    }

    private void pictureBox3_Click(object sender, EventArgs e)
    {
        System.Windows.Forms.Application.Exit();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (txtTc.Text != "" && txtSifre.Text != "") // Textler boş olup olmadığı
        kontrol edildi
        {
            if (sorgu() == 1) //Girilen değerler veri tabanındaki tabloda mevcut
            ise işlem gerçekleşti
            {
                MessageBox.Show("Giriş başarılı");
                Hasta_Randevu_Islem gorevliGecis = new Hasta_Randevu_Islem(); //
                Gorevli geçiş formundan nesne üretildi
                gorevliGecis.lblHastaId.Text = lblHastaId.Text;
                this.Hide(); // Bulunduğumuz fonksiyon
                kapatıldı
                gorevliGecis.Show(); //
                Oluşturulan nesnenin formu açıldı
            }
        }
    }

```

```

        }

        else
        {
            MessageBox.Show("Hatalı giriş");
        }

    }

    else
    {
        MessageBox.Show("Gerekli alanları doldurunuz !");
    }

}

}

private void HastaGiris_Load(object sender, EventArgs e)
{
}

//Veri gizleme
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.CheckState == CheckState.Checked) //CheckBox seçili ise
        parolayı göster text e gizle yaz
    {
        txtSifre.UseSystemPasswordChar = true;
        checkBox1.Text = "Gizle";
    }
    else if (checkBox1.CheckState == CheckState.Unchecked) //CheckBox seçili
        değil ise parolayı gizle ve text e göster yaz
    {
        txtSifre.UseSystemPasswordChar = false;
        checkBox1.Text = "Göster";
    }
}

//Harf girişi engelleme
private void txtTc_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txtSifre_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
}
}

```

Hasta_Randevu_İslem.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class Hasta_Randevu_Islem :
    Form
    {
        public Hasta_Randevu_Islem()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            HastaGiris hasta = new HastaGiris();
this.Hide();          hasta.Show();
        }
        private void pictureBox3_Click(object sender, EventArgs e)
        {
            System.Windows.Forms.Application.Exit();
        }

        // Veri Tabanı Bağlantısı
        SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");

        //Listeleme Fonksiyonu
        private void veriliste()
        {
            int id = int.Parse(lblHastaId.Text.ToString());
            baglanti.Open();
            SqlCommand komut = new SqlCommand("Select *From Hasta_Table where
Hasta_Id=(" + id + ")", baglanti);
            SqlDataReader oku = komut.ExecuteReader();

            while (oku.Read())
            {
                lblAd.Text = oku["Ad"].ToString();
                lblSoyad.Text = oku["Soyad"].ToString();
                lblTc.Text = oku["TC"].ToString();          lblSifre.Text
                = oku["Sifre"].ToString();
                lblHasta.Text="Hoşgeldin " + oku["Ad"].ToString();
            }
            baglanti.Close();
        }

        //Bolum Listeleme Fonksiyonu
        private void bolumliste()
        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("Select *From Bolum_Table", baglanti);
            SqlDataReader oku = komut.ExecuteReader();
            while
(oku.Read())
            {
                comboBoxBolum.Items.Add(oku["Bolum_Ad"].ToString());
            }
        }
    }
}

```

```

        baglanti.Close();
    }

    //Doktor Listeleme Fonksiyonu

    private void doktorlistele()
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Select * From Doktor_Table where
Bolum_Id='" + id+"'", baglanti);
        SqlDataReader oku = komut.ExecuteReader();

        while (oku.Read())
        {
            comboBoxDoktor.Items.Add(oku["Ad"].ToString()+" "+
oku["Soyad"].ToString());
            //doktorId = int.Parse(oku["Doktor_Id"].ToString());
        }
        baglanti.Close();
    }
    private void Hasta_Randevu_Islem_Load(object sender, EventArgs e)
    {
        lblCeza.Text = gelmeKontrolu().ToString();
        randevuListele();          veriliste();
        bolumlistele();

    }
    private void lblHastaId_Click(object sender, EventArgs e)
    {
    }

    //Bolum tiklama
    int id;
    private void comboBoxBolum_MouseClick(object sender, MouseEventArgs e)
    {

    }

    //ComboBox secme
    private void comboBoxBolum_SelectedIndexChanged(object sender, EventArgs e)
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Select *From Bolum_Table where
Bolum_Ad='" + comboBoxBolum.Text + "'", baglanti);
        SqlDataReader oku = komut.ExecuteReader();

        while (oku.Read())
        {
            id = int.Parse(oku["Bolum_Id"].ToString());
        }
        baglanti.Close();
        comboBoxDoktor.Items.Clear();          doktorlistele();
    }

```

```

    }
    private void btn1_Click(object sender,
EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn1.Text;
    }
    private void btn2_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn2.Text;
    }
    private void btn3_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn3.Text;
    }
    private void btn4_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn4.Text;
    }
    private void btn5_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn5.Text;
    }
    private void btn6_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn6.Text;
    }
    private void btn7_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn7.Text;
    }
    private void btn8_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn8.Text;
    }
    private void btn9_Click(object sender,
EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn9.Text;
    }
    private void btn10_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn10.Text;
    }
    private void btn11_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn11.Text;
    }

    private void btn12_Click(object sender, EventArgs e)
    {

```

```

        txtSaat.Text = "";
txtSaat.Text = btn12.Text;
    }
    private void btn13_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn13.Text;
    }
    private void btn14_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn14.Text;
    }
    private void btn15_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn15.Text;
    }
    private void btn16_Click(object sender, EventArgs e)
    {
        txtSaat.Text = "";
txtSaat.Text = btn16.Text;
    }
}

//Button renklendirme fonksiyonu
private void butonRenk()
{
    //renklendirme
    btn1.BackColor = Color.Lime; btn2.BackColor = Color.Lime; btn3.BackColor =
Color.Lime; btn4.BackColor = Color.Lime;
    btn5.BackColor = Color.Lime; btn6.BackColor = Color.Lime; btn7.BackColor =
Color.Lime; btn8.BackColor = Color.Lime;
    btn9.BackColor = Color.Lime; btn10.BackColor = Color.Lime; btn11.BackColor
= Color.Lime; btn12.BackColor = Color.Lime;
    btn13.BackColor = Color.Lime; btn14.BackColor = Color.Lime;
    btn15.BackColor = Color.Lime; btn16.BackColor = Color.Lime;
    // enabled kontrol
    btn1.Enabled = true; btn2.Enabled = true; btn3.Enabled = true;
    btn4.Enabled = true;
    btn5.Enabled = true; btn6.Enabled = true; btn7.Enabled = true;
    btn8.Enabled = true;
    btn9.Enabled = true; btn10.Enabled = true; btn11.Enabled = true;
    btn12.Enabled = true;
    btn13.Enabled = true; btn14.Enabled = true; btn15.Enabled = true;
    btn16.Enabled = true;
}

// Saat doluluğu kontrolu
private void saatKontrolu(DateTime tarih)
{
    string saatsizTarih;
    string[] splittedText = tarih.ToString().Split('
');
    saatsizTarih = splittedText[0];
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select * from Randevu_Table where
Doktor_Id='"+doktorId+"' and Tarih='"+saatsizTarih+"'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();

```



```

        while (oku.Read())
        {
            if (oku["Saat"].ToString() == btn1.Text)
            {
                btn1.BackColor = Color.Gray;
btn1.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn2.Text)
            {
                btn2.BackColor = Color.Gray;
btn2.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn3.Text)
            {
                btn3.BackColor = Color.Gray;
btn3.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn4.Text)
            {
                btn4.BackColor = Color.Gray;
btn4.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn5.Text)
            {
                btn5.BackColor = Color.Gray;
btn5.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn6.Text)
            {
                btn6.BackColor = Color.Gray;
btn6.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn7.Text)
            {
                btn7.BackColor = Color.Gray;
btn7.Enabled = false;
            }
            else if (oku["Saat"].ToString() ==
btn8.Text)
            {
                btn8.BackColor = Color.Gray;
btn8.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn9.Text)
            {
                btn9.BackColor = Color.Gray;
btn9.Enabled = false;
            }

            else if (oku["Saat"].ToString() == btn10.Text)
            {

```

```

        btn10.BackColor = Color.Gray;
btn10.Enabled = false;
    }

    else if (oku["Saat"].ToString() == btn11.Text)
    {
        btn11.BackColor = Color.Gray;
btn11.Enabled = false;
    }

    else if (oku["Saat"].ToString() == btn12.Text)
    {
        btn12.BackColor = Color.Gray;
btn12.Enabled = false;
    }

    else if (oku["Saat"].ToString() == btn13.Text)
    {
        btn13.BackColor = Color.Gray;
btn13.Enabled = false;
    }

    else if (oku["Saat"].ToString() == btn14.Text)
    {
        btn14.BackColor = Color.Gray;
btn14.Enabled = false;
    }

    else if (oku["Saat"].ToString() == btn15.Text)
    {
        btn15.BackColor = Color.Gray;
btn15.Enabled = false;
    }
    else if (oku["Saat"].ToString() == btn16.Text)
    {
        btn16.BackColor = Color.Gray;
btn16.Enabled = false;
    }

    }
    baglanti.Close();
}
// Geldi mi fonksiyonu
private int gelmeKontrolu()
{
    int sonuc=0;
    int id = int.Parse(lblHastaId.Text.ToString());
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select Geldi_mi from Randevu_Table
where Hasta_Id = '" + id + "'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();

    while (oku.Read())
    {

        if (oku["Geldi_mi"].ToString() == "False")

sonuc++;

```

```

    }
    baglanti.Close();
return sonuc;
}

//Randevu Listeleme
private void randevuListele()
{
    int id = int.Parse(lblHastaId.Text.ToString());
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select
r.Randevu_No,b.Bolum_Ad,d.Ad,d.Soyad,r.Tarih,r.Saat,r.Geldi_mi from ((Doktor_Table d
inner join Randevu_Table r on d.Doktor_Id=r.Doktor_Id) inner join Bolum_Table b on
b.Bolum_Id=d.Bolum_Id)where r.Hasta_Id = '" + id+"'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();

    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Randevu_No"].ToString();
        ekle.SubItems.Add(oku["Bolum_Ad"].ToString());
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["Tarih"].ToString());
        ekle.SubItems.Add(oku["Saat"].ToString());
        randevuListe.Items.Add(ekle);
    }
    baglanti.Close();
}
int tarihSayac = 0;
int saatSayac = 0;
private int randevuDoktorKontrol()
{
    int sayac = 0;

    int id = int.Parse(lblHastaId.Text.ToString());
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select * from Randevu_Table where
Hasta_Id = '" + id + "'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while
(oku.Read())
    {
        if (doktorId.ToString()== oku["Doktor_Id"].ToString() &&
dtRandevu.Text == oku["Tarih"].ToString())
            sayac++;

        if (dtRandevu.Text == oku["Tarih"].ToString())
            tarihSayac++;

        if (dtRandevu.Text == oku["Tarih"].ToString() &&
txtSaat.Text==oku["Saat"].ToString())
            saatSayac++;
    }
    baglanti.Close();
return sayac;
}
//Randevu ekleme

```

```

private void randevuEkleme()
{
    DateTime bugun = DateTime.Today;
    //DateTime secimTarih = DateTime.Parse(dtRandevu.Text);
    DateTime secimTarih = trh;
string saatsizTarih;

    string[] splittedText = secimTarih.ToString().Split(' ');
saatsizTarih = splittedText[0];

    if (comboBoxBolum.Text != "" && comboBoxDoktor.Text != "" &&
dtRandevu.Text != "" && txtSaat.Text != "")
    {
        if(secimTarih>=bugun)
        {
            if (randevuDoktorKontrol() == 0)
            {
                if (tarihSayac < 5)
                {
                    if (saatSayac == 0)
                    {
                        int hastaId = int.Parse(lblHastaId.Text.ToString());

                        baglanti.Open();
                        SqlCommand komut = new SqlCommand("insert into
Randevu_Table (Hasta_Id,Bolum_Id,Doktor_Id,Tarih,Saat) values ('" + hastaId + "','" +
id + "','" + doktorId + "','" + saatsizTarih + "','" + txtSaat.Text + "')", baglanti);
                        komut.ExecuteNonQuery();
baglanti.Close();

                        randevuListe.Items.Clear(); // ListBox Temizle
randevuListele(); // Listeleme Islemi
saatKontrolu(dtRandevu.Value);
comboBoxBolum.Text = null;
comboBoxDoktor.Text
= null;
dtRandevu.Text = null;
txtSaat.Text = null;

                        MessageBox.Show("Kayıt Eklendi ");

                    }
                    else
                        MessageBox.Show("Gün içerisinde aynı saatte başka bir
randevunuz var !");
                    saatSayac = 0;
                }
            }
            else
                MessageBox.Show("Gün içerisinde en fazla 3 randevu
alabilirsiniz !");
            tarihSayac = 0;
        }
    }
    else
    {
        MessageBox.Show("Gün içerisinde aynı doktordan sadece 1 defa
randevu alabilirsiniz !");
    }
}

```

```

        else
            MessageBox.Show("Bugünden geçmiş bir tarihe randevu alınamaz !");

    }

    else
        MessageBox.Show("Lütfen gerekli alanları doldurunuz !");

    }
    private void btnAl_Click(object sender, EventArgs e)
    {
        if(int.Parse(lblCeza.Text)<3)
        {
            randevuEkleme();
            butonRenk();
        }

        else
            MessageBox.Show("3 defa randevuya gelmediğiniz için randevu alamazsınız
!");
    }

    // Sorgu Fonksiyonu

    public int sorgu()
    {
        int idSil = int.Parse(txt_IdSil.Text);
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Select *From Randevu_Table where
Randevu_No=(" + idSil + ")", baglanti);
        SqlDataReader oku = komut.ExecuteReader();
        int sonuc = 0;          int sayac = 0;
        while (oku.Read())
        {
            sayac++;

        }
        if (sayac >
0)
        {
            sonuc = 1;
        }
        baglanti.Close();
        return sonuc;

    }
    // Randevu İptal Etme
    private void randevuIptal()
    {

        if (txt_IdSil.Text != "")
        {
            int idSil = int.Parse(txt_IdSil.Text);
            int id = int.Parse(txt_IdSil.Text);          if
(sorgu() == 1)
            {

```

```

        baglanti.Open();
        SqlCommand komut = new SqlCommand("Delete From Randevu_Table where
Randevu_No =(" + idSil + ")", baglanti); komut.ExecuteNonQuery();
        baglanti.Close();
        randevuListe.Items.Clear(); // ListBox Temizle
        randevuListele(); // Listeleme Islemi
        txt_IdSil.Clear();
        MessageBox.Show("Randevu iptal edildi");

    }

else
    MessageBox.Show("Kayıt bulunamadı");

}

else
    MessageBox.Show("Lütfen id giriniz !");
}
//Doktor Id
int doktorId;
private void comboBoxDoktor_SelectedIndexChanged(object sender, EventArgs e)
{
    string ad, soyad;

    string[] splittedText = comboBoxDoktor.Text.ToString().Split(' ');
    ad=splittedText[0]; soyad= splittedText[1];

    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select * From Doktor_Table where Ad='"
+ ad + "'and Soyad='" + soyad + "'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();

    while (oku.Read())
    {
        doktorId = int.Parse(oku["Doktor_Id"].ToString());
    }
    baglanti.Close();
    dtRandevu.Value = DateTime.Today;
    butonRenk();
    saatKontrolu(dtRandevu.Value);
    randevuDoktorKontrol();

}

private void btnTeslim_Click(object sender,
EventArgs e)
{
    randevuIptal();
}

//Harf girisi engelleme
private void txt_IdSil_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

//Tarih
DateTime trh;
private void dtRandevu_ValueChanged(object sender, EventArgs e)

```

```

        {
            butonRenk();
            trh = dtRandevu.Value;
            saatKontrolu(trh);
        }
    }
}

```

DoktorGiris.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class DoktorGiris : Form
    {
        public DoktorGiris()
        {
            InitializeComponent();

            // Veri Tabanı Bağlantısı
            SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");
            // Sorgu Fonksiyonu
            public int sorgu()
            {
                baglanti.Open();
                SqlCommand komut = new SqlCommand("Select *From Doktor_Table where TC=( " +
                txtTc.Text.ToString() + ") and Sifre=( " + txtSifre.Text.ToString() + ")", baglanti);
                SqlDataReader oku = komut.ExecuteReader();
                int sonuc = 0;                int sayac = 0;
                while (oku.Read())
                {
                    sayac++;
                    lblDoktorId.Text = oku["Doktor_Id"].ToString();
                }
                if (sayac >
0)
                {
                    sonuc = 1;
                }
                baglanti.Close();
            return sonuc;
        }
        private void pictureBox3_Click(object sender, EventArgs e)
        {

```

```

        System.Windows.Forms.Application.Exit();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        GirisFormu grs = new GirisFormu();
        this.Hide();
        grs.Show();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        if (txtTc.Text != "" && txtSifre.Text != "") // Textler boş olup olmadığı kontrol edildi
        {
            if (sorgu() == 1) //Girilen değerler veri tabanındaki tabloda mevcut ise işlem gerçekleşti
            {
                MessageBox.Show("Giriş başarılı");
                DoktorRandevuIslem grs = new DoktorRandevuIslem(); // Gorevli geçiş formundan nesne üretildi
                grs.lblDoktorId.Text = lblDoktorId.Text;
                this.Hide();
                grs.Show(); // Oluşturulan nesnenin formu açıldı
            }
            else
            {
                MessageBox.Show("Hatalı giriş");
            }
        }
        else
        {
            MessageBox.Show("Gerekli alanları doldurunuz !");
        }
    }

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
        if (checkBox1.CheckState == CheckState.Checked) //CheckBox seçili ise parolayı göster text e gizle yaz
        {
            txtSifre.UseSystemPasswordChar = true;
            checkBox1.Text = "Gizle";
        }
        else if (checkBox1.CheckState == CheckState.Unchecked) //CheckBox seçili değil ise parolayı gizle ve text e göster yaz
        {
            txtSifre.UseSystemPasswordChar = false;
            checkBox1.Text = "Göster";
        }
    }

    //Harf girişi engelleme
    private void txtTc_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    }
    private void txtSifre_KeyPress(object sender, KeyPressEventArgs e)

```



```

        {
            e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
        }
        private void DoktorGiris_Load(object sender, EventArgs e)
        {
        }
    }
}

```

Doktor_Islemleri.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data; using
System.Drawing; using
System.Linq; using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace HastaneRandevuSistemi
{
    public partial class Doktor_Islemleri : Form
    {
        // Veri Tabanı Bağlantısı
        SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
PC77Q2I;Initial Catalog=HastaneRandevu;Integrated Security=True");

        public Doktor_Islemleri()
        {
            InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            Yonetici_Islem yonetici = new Yonetici_Islem();
            yonetici.Show();
        }
        private void pictureBox3_Click(object sender,
        EventArgs e)
        {
            System.Windows.Forms.Application.Exit();
        }
        private void Doktor_Islemleri_Load(object sender, EventArgs e)
        {
            veriliste();
            lblListe.Text = "DOKTOR BİLGİ LİSTESİ";
            lblBilgi.ForeColor = Color.Yellow;
            lblMuayene.ForeColor = Color.White;
        }

        // Muayene Listele
        private void muayeneListele()
        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("Select
            r.Randevu_No,h.Ad,h.Soyad,h.TC,r.Tarih,r.Saat,r.Geldi_mi,r.Aciklama from

```

```

((Doktor_Table d inner join Randevu_Table r on d.Doktor_Id=r.Doktor_Id) inner join
Hasta_Table h on h.Hasta_Id=r.Hasta_Id) where r.Doktor_Id='"+ id + "' ", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    doktorListe.Clear();

    doktorListe.Columns.Add("Randevu_No",100);
    doktorListe.Columns.Add("Hasta_Ad", 100);
    doktorListe.Columns.Add("Hasta_Soyad", 100);
    doktorListe.Columns.Add("TC", 100);
    doktorListe.Columns.Add("Tarih", 100);
    doktorListe.Columns.Add("Saat", 100);
    doktorListe.Columns.Add("Geldi_mi", 100);
    doktorListe.Columns.Add("Aciklama", 100);           while
    (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Randevu_No"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TC"].ToString());
        ekle.SubItems.Add(oku["Tarih"].ToString());
        ekle.SubItems.Add(oku["Saat"].ToString());
        ekle.SubItems.Add(oku["Geldi_mi"].ToString());
        ekle.SubItems.Add(oku["Aciklama"].ToString());
        doktorListe.Items.Add(ekle);

    }
    baglanti.Close();
}
//Listeleme Fonksiyonu
private void veriliste()
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select
d.Doktor_Id,d.Ad,d.Soyad,d.TC,b.Bolum_Ad from (Doktor_Table d inner join Bolum_Table b
on d.Bolum_Id=b.Bolum_Id)", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    doktorListe.Clear();

    doktorListe.Columns.Add("Doktor_Id", 100);
    doktorListe.Columns.Add("Ad", 100);
    doktorListe.Columns.Add("Soyad", 100);
    doktorListe.Columns.Add("TC", 100);
    doktorListe.Columns.Add("Bolum_Ad", 100);           while
    (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Doktor_Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TC"].ToString());
        ekle.SubItems.Add(oku["Bolum_Ad"].ToString());
        doktorListe.Items.Add(ekle);

    }
    baglanti.Close();
}

```

```

        // Sorgu Fonksiyonu
public int sorgu()
{
    int id = int.Parse(txt_IdSil.Text);
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select *From Doktor_Table where
Doktor_Id=(" + id + ")", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    int sonuc = 0;          int sayac = 0;
    while (oku.Read())
    {
        sayac++;

    }

    if (sayac > 0)
    {
        sonuc = 1;
    }
    baglanti.Close();
return sonuc;

    }
int deger;
//Bolum Id
private int bolumId(string combo)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Select *From Bolum_Table where
Bolum_Ad='" + combo + "'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while
(oku.Read())
    {

        deger = int.Parse(oku["Bolum_Id"].ToString());
    }
    baglanti.Close();
return deger;
    }
//Ekleme Islemi
int veri;
private void btnEkle_Click(object sender, EventArgs e)
{

    if (txtAd.Text != "" && txtSoyad.Text != "" && txtTc.Text != "" &&
comboBoxBolum.Text != "")
    {
        veri = bolumId(comboBoxBolum.Text);
        baglanti.Open();
        SqlCommand komut = new SqlCommand("insert into Doktor_Table
(Ad,Soyad,TC,Bolum_Id) values ('" + txtAd.Text.ToString() + "','" +
txtSoyad.Text.ToString() + "','" + txtTc.Text.ToString() + "','" + veri + "')",
baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        doktorListe.Items.Clear(); // ListBox Temizle
        verilistele();           // Listeleme Islemi-

```

```

txtAd.Clear();                txtSoyad.Clear();
txtTc.Clear();
        comboBoxBolum.Text = null;
MessageBox.Show("Kayıt Eklendi ");
    }

    else
        MessageBox.Show("Lütfen gerekli alanları doldurunuz !");
}

// sayı engelleme
private void txtAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txtSoyad_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txt_gAd_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}
private void txt_gSayfa_KeyPress(object sender,
KeyPressEventArgs e)
{
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) &&
!char.IsSeparator(e.KeyChar);
}

//Harf girisi engelleme
private void txtTc_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}
private void txt_gYazar_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void txt_IdSil_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

//Silme Islemi
private void btnSil_Click(object sender, EventArgs e)
{
    if (txt_IdSil.Text != "")
    {
        int id = int.Parse(txt_IdSil.Text);
if (sorgu() == 1)
        {
            baglanti.Open();

```

```

        SqlCommand komut = new SqlCommand("Delete From Doktor_Table where
Doktor_Id =(" + id + ")", baglanti); komut.ExecuteNonQuery();
baglanti.Close();

        doktorListe.Items.Clear(); // ListBox Temizle
verilistele(); // Listeleme Islemi
        txt_IdSil.Clear();
        MessageBox.Show("Kayıt silindi");

    }

else
        MessageBox.Show("Kayıt bulunamadı");
    }

else
        MessageBox.Show("Lütfen id giriniz !");
    }

//Listeden Secme Islemi
int id = 0;
private void doktorListe_MouseClick(object sender, MouseEventArgs e)
{
    if(lblListe.Text != "MUAYENE BİLGİ LİSTESİ")
    {
        id = int.Parse(doktorListe.SelectedItems[0].SubItems[0].Text);
        txt_gAd.Text = doktorListe.SelectedItems[0].SubItems[1].Text;
        txt_gSoyad.Text = doktorListe.SelectedItems[0].SubItems[2].Text;
        txt_gTc.Text = doktorListe.SelectedItems[0].SubItems[3].Text;
        comboBox_gBolum.Text = doktorListe.SelectedItems[0].SubItems[4].Text;
    }

}

//Guncelleme Islemi
private void btnGuncelle_Click(object sender, EventArgs e)
{
    veri = bolumId(comboBox_gBolum.Text);
    if (id != 0)
    {
        if (txt_gAd.Text != "" && txt_gSoyad.Text != "" && txt_gTc.Text != ""
&& comboBox_gBolum.Text != "")
        {
            baglanti.Open();
            SqlCommand komut = new SqlCommand("update Doktor_Table set Ad ="
+ txt_gAd.Text.ToString() + "',Soyad =' " + txt_gSoyad.Text.ToString() + "',TC =' " +
txt_gTc.Text.ToString() + "',Bolum_Id =' " + veri + "' where Doktor_Id=" + id + "",
baglanti);
            komut.ExecuteNonQuery();
            baglanti.Close();

            doktorListe.Items.Clear(); // ListBox Temizle
verilistele(); // Listeleme Islemi
            txt_gAd.Clear(); txt_gSoyad.Clear();
            txt_gTc.Clear();
            comboBox_gBolum.Text = null;
            MessageBox.Show("Kayıt güncellendi");
            id = 0;
        }
        else
            MessageBox.Show("Lütfen gerekli alanları doldurunuz !");
    }

else

```

```

        MessageBox.Show("Lütfen listeden güncellemek istediğiniz doktoru seçiniz
!");
    }

    //Muayene bilgisi click
    private void pictureBoxIadeBilgi_MouseClick(object sender, MouseEventArgs e)
    {

    }

    //Doktor bilgi click
    private void pictureBoxKitapBilgi_Click(object sender, EventArgs e)
    {
        veriliste();
        lblListe.Text = "DOKTOR BİLGİ LİSTESİ";    // Labeldaki yazı değişti
        lblBilgi.ForeColor = Color.Yellow;        // İade label i pasif rengi aldı
        lblMuayene.ForeColor = Color.White;
        lblBilgilendirme.Text = "Not: Muayene bilgisini öğrenmek istediğiniz
doktorun listeden üzerine tıklayıp iade bilgisi butonuna basınız.";
    }
    private void pictureBoxMuayeneBilgi_Click(object sender, EventArgs e)
    {
        if (id != 0)
        {
            muayeneListele();
            lblListe.Text = "MUAYENE BİLGİ LİSTESİ";
            lblBilgi.ForeColor = Color.White;        lblMuayene.ForeColor
= Color.Yellow;
            lblBilgilendirme.Text = "Not: Doktor bilgilerine geri dönmek için doktor
bilgisi butonuna basınız.";
        }
    else
        MessageBox.Show("Lütfen doktor seçiniz !");

    }
}
}
}

```