



Staj Proje Raporu

ÖĞRENCİ BİLGİ SİSTEMİ

YUSUF KAYA-190205011009

MÜHENDİSLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

İçindekiler

| | |
|--|----|
| Staj Proje Raporu | 0 |
| Giriş..... | 2 |
| Projenin Amacı..... | 2 |
| Kazanımlar | 2 |
| Projeye Giriş | 3 |
| Varlıklar ve Veri Tabanı Modeli | 3 |
| Proje Tasarımı ve Kodları..... | 4 |
| Veri tabanı bağlantısı Classı | 4 |
| Giriş Ve Şifremi Unuttum Ekranları | 5 |
| Personel Kontrol Paneli | 8 |
| Personel Ekleme Ekranı | 9 |
| Personel Düzenleme Ekranı | 10 |
| Yeni Öğrenci Kayıt Ekranı..... | 12 |
| Öğrenci Düzenleme Silme Ekranı..... | 13 |
| Akademisyen Ekleme Ekranı..... | 15 |
| Akademisyen Düzenleme Ekranı | 16 |
| Bölüm Ekleme Ekranı | 17 |
| Ders Ekleme ve Ders Listesi Ekranları | 18 |
| Not Girişi Ekranı | 19 |
| Devamsızlık Girişi Ekranı | 20 |
| Sorumlu Olduğum Dersler Ekranı | 21 |
| Duyuru Gönderme ve Gönderilen Duyuruları Listeleme Ekranları | 22 |
| Sınav Sonuçları Ekranı..... | 23 |
| Devamsızlık Görüntüleme Ekranı | 24 |
| Duyuru Görüntüleme Ekranı (Öğrenci)..... | 25 |
| EKLER | 26 |
| Proje Github Linki | 26 |

Giriş

Projenin Amacı

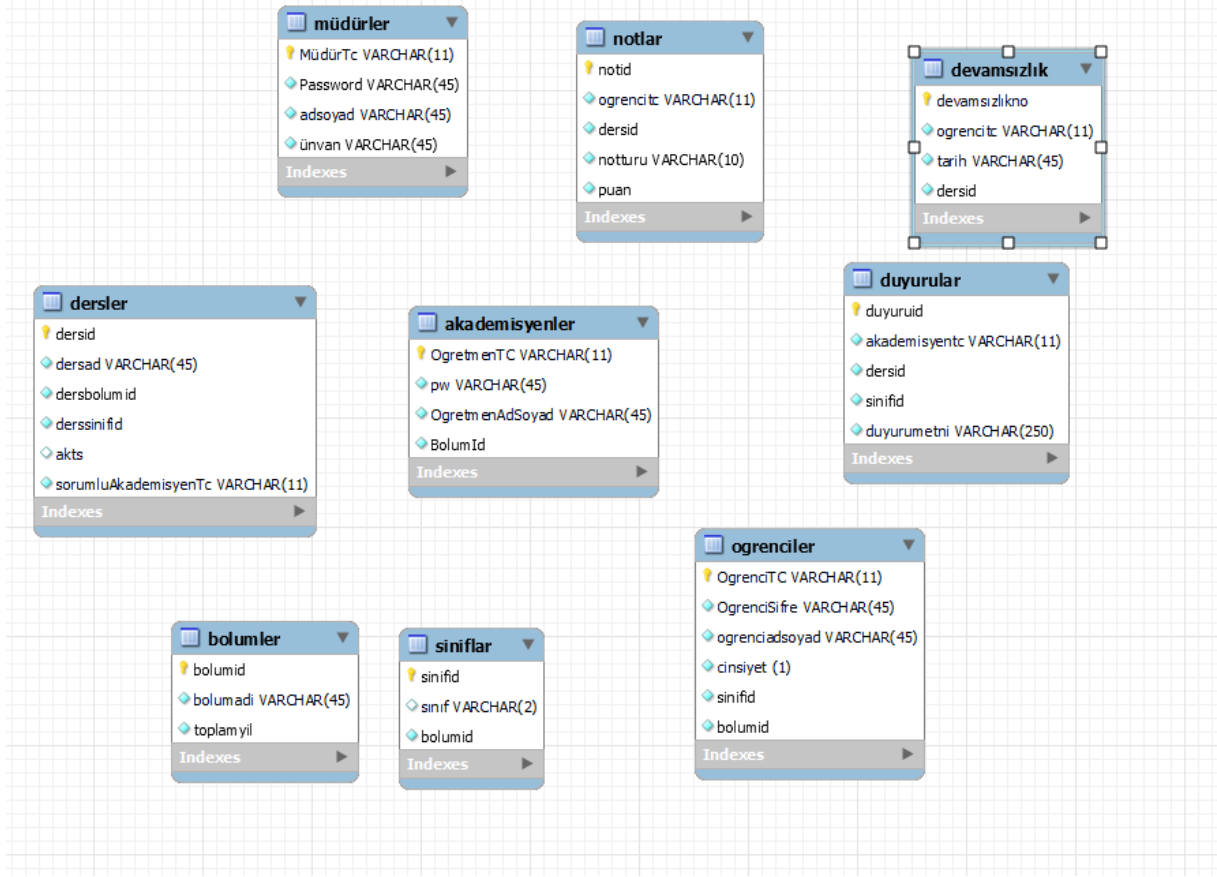
Projenin amacı bir üniversitenin veya okulun içindeki varlıkların kolaylıkla yönetilmesi ve raporlanabilmesini sağlayabilmektir Öğrenci Bilgi Sistemi Personeller Akademisyenler ve Öğrenciler arasında bir köprü görevi görmektedir.

Kazanımlar

Bu Projeyi tasarlarken ve kodlarken meslek hayatımda karşıma çıkacak sorunlarla nasıl başa çıkabileceğimi bir problemi nasıl çözebileceğimi öğrendim.

Projeye Giriş

Varlıklar ve Veri Tabanı Modeli



Projede kullandığım tüm varlıklar veri tabanı modelinde mevcuttur bu modeli baz alarak projemi geliştirdim.

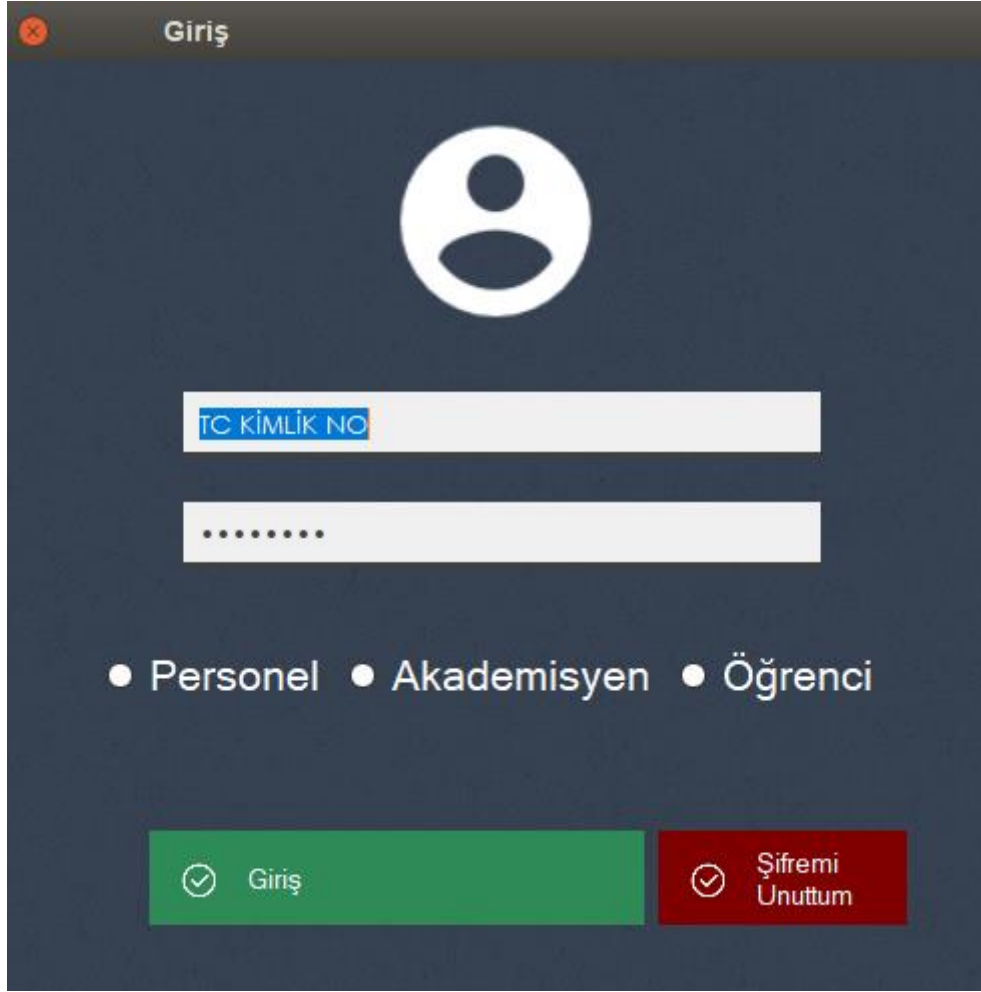
Proje Tasarımı ve Kodları

Veri tabanı bağlantısı Classı

Projedeki Tüm İşlemlerin altyapısını sağlayan classtır .Statik olduğu için nesne örneği üretmemize gerek yoktur.

```
DbController.cs  X
OgrenciBilgiSistemi  OgrenciBilgiSistemi.DbController
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Data;
7  using MySql.Data.MySqlClient;
8  using System.Windows.Forms;
9
10 namespace OgrenciBilgiSistemi
11 {
12     99 references
13     class DbController
14     {
15         public static MySqlConnection conn = new MySqlConnection(@"server=localhost;user id=root;password=admin;database=ogrencibilgisistem");
16
17         40 references
18         public static void Connect()
19         {
20             if (conn.State != ConnectionState.Open)
21             {
22                 try
23                 {
24                     conn.Open();
25                 }
26                 catch (Exception ex)
27                 {
28                     MessageBox.Show(ex.ToString());
29                 }
30             }
31         }
32     }
33
34     18 references
35     public static void Disconnect()
36     {
37         if (conn.State != ConnectionState.Closed)
38         {
39             try
40             {
41                 conn.Close();
42             }
43             catch (Exception ex)
44             {
45                 MessageBox.Show(ex.ToString());
46             }
47         }
48     }
49
50
51 }
```

Giriş Ve Şifremi Unuttum Ekranları



Projede 3 adet giriş yöntemi olduğu için bu giriş yöntemlerine ait 3 fonksiyon vardır.

```
public bool MüdürLogin(string tcno, string password)
{
    if (tcno.Length == 0 && password.Length == 0)
    {
        return false;
    }
    else
    {
        query = "SELECT * FROM müdürler where MüdürTc=@P1 AND password=@P2";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", tcno);
        cmd.Parameters.AddWithValue("@P2", password);
        DbController.Connect();
        dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            DbController.Disconnect();
            dr.Close();
            return true;
        }
        else
        {
            DbController.Disconnect();
            dr.Close();
            return false;
        }
    }
}
```

Personel Classındaki personel giriş kodları

```

public bool OgrenciLogin(string tcno, string password)
{
    if (tcno.Length == 0 && password.Length == 0)
    {
        return false;
    }
    else
    {
        query = "SELECT * FROM ogrenciler where OgrenciTC=@P1 AND OgrenciSifre=@P2";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", tcno);
        cmd.Parameters.AddWithValue("@P2", password);
        DbController.Connect();
        dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            DbController.Disconnect();
            dr.Close();
            return true;
        }
        else
        {
            DbController.Disconnect();
            dr.Close();
            return false;
        }
    }
}

```

Öğrenci classındaki Öğrenci giriş kodları

```

public bool AkademisyenLogin(string tcno, string password)
{
    if (tcno.Length == 0 && password.Length == 0)
    {
        return false;
    }
    else
    {
        query = "SELECT * FROM akademisyenler where OgretmenTC=@P1 AND pw=@P2";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", tcno);
        cmd.Parameters.AddWithValue("@P2", password);
        DbController.Connect();
        dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            DbController.Disconnect();
            dr.Close();
            return true;
        }
        else
        {
            DbController.Disconnect();
            dr.Close();
            return false;
        }
    }
}

```

Akademisyen classında bulunan Akademisyen Giriş Kodları

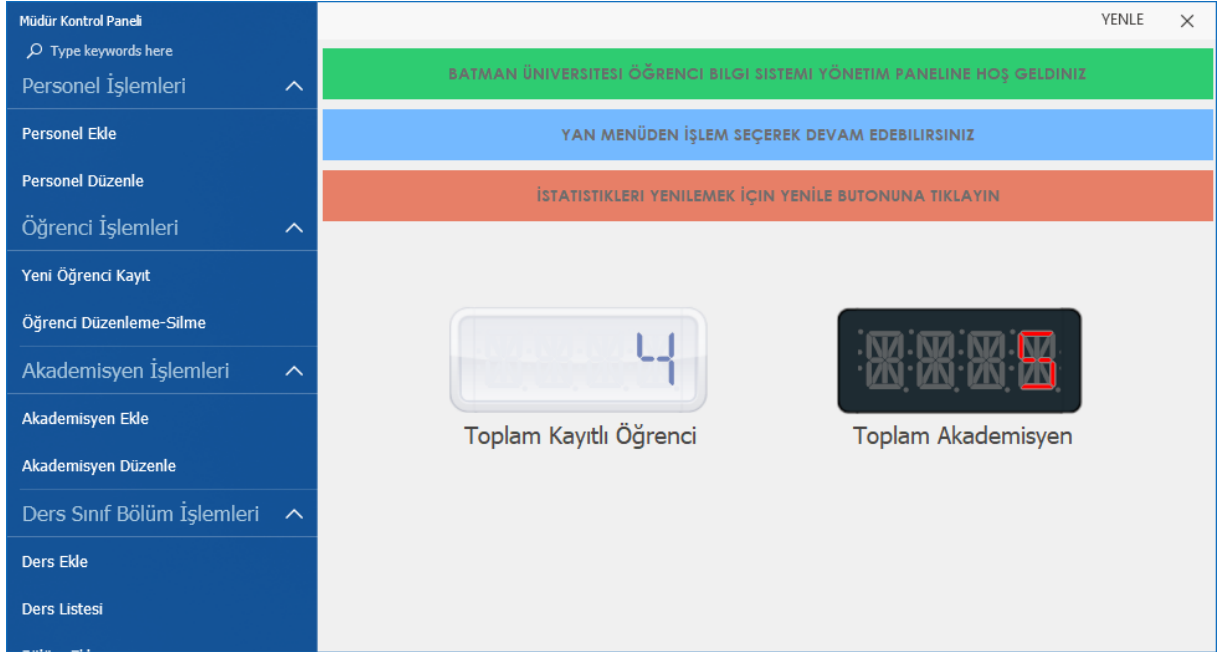
Yine girişte olduğu gibi şifremi unuttum ekranında da üç farklı seçenek olduğu için 3 classta ayrı ayrı metotlar vardır.her birinin mantığı aynı olduğu için sadece öğrenci şifremi unuttum metodunu göstereceğim.

```
public string getForgetPassword(string tc)
{
    try
    {
        query = "SELECT OgrenciSifre FROM ogrenciler where OgrenciTc=@P1";
        DbController.Connect();
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", tc);
        return cmd.ExecuteScalar().ToString();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return null;
    }
}
```

Girilen bilgilere göre maile gönderilen mesajın kodları

```
if (Ogrenci_radio.Checked && tc_textbox.Text.Length == 11 && emailbx.Text.Length > 0)
{
    if (ogrenci.getForgetPassword(tc_textbox.Text) != null)
    {
        var client = new SmtpClient("smtp.gmail.com", 587)
        {
            Credentials = new NetworkCredential("obssifresender@gmail.com", "YusuFk47"),
            EnableSsl = true
        };
        client.Send("obssifresender@gmail.com", emailbx.Text, "Üniversite Bilgi Sistemi Şifre İsteği", "Merhaba," +
            "\n Girilen Tc Numarasına Ait Öğrenci Şifresi:" + ogrenci.getForgetPassword(tc_textbox.Text) + " dir \n İyi Günler Dileriz!");
        MessageBox.Show("Gönderildi!");
    }
    else
        MessageBox.Show("Tc Kimlik No hatalı");
}
else MessageBox.Show("Gerekli Alanları Doldurun!");
```


Personel Kontrol Paneli



Personel ekranında personellerin yapabileceği işlemleri listeledim. Ekrandaki Sayaç verilerini SQL Select Count komutuyla çektim

```
public int countStudent()  
{  
    query = "SELECT COUNT(*) FROM ogrenciler";  
    DbController.Connect();  
    cmd = new MySqlCommand(query, DbController.conn);  
    Int32 count = Int32.Parse(cmd.ExecuteScalar().ToString());  
    DbController.Disconnect();  
    return count;  
}
```

Öğrenci Sayısını Çeken Kod

```
public int countAkademişyen()  
{  
    query = "SELECT COUNT(*) FROM akademisyonlar";  
    DbController.Connect();  
    cmd = new MySqlCommand(query, DbController.conn);  
    Int32 count = Int32.Parse(cmd.ExecuteScalar().ToString());  
    DbController.Disconnect();  
    return count;  
}
```

Akademişyen Sayısını Çeken Kod

Personel Ekleme Ekranı

Müdür-Müdür Yardımcısı Ekle

Tc Kimlik No

Şifre

Ad Soyad

Ünvan

Yeni Yönetici Ekle

Personelin bu ekranda girdiği belgeler veri tabanındaki müdürler tablosuna aşağıdaki kod yardımıyla insert edilmektedir.

```
public bool addnewPersonel(string tc, string pw, string adsoyad, string unvan)
{
    if (tc.Length == 11 && pw.Length > 0 && adsoyad.Length > 0 && unvan.Length > 0)
    {
        try
        {
            DbController.Connect();
            query = "INSERT INTO müdürler values(@P1,@P2,@P3,@P4)";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", tc);
            cmd.Parameters.AddWithValue("@P2", pw);
            cmd.Parameters.AddWithValue("@P3", adsoyad);
            cmd.Parameters.AddWithValue("@P4", unvan);
            cmd.ExecuteNonQuery();

            return true;
        }
        catch (MySqlException ex)
        {
            if (ex.Number == 1062)
            {
                MessageBox.Show("Girilen Tc İle Eşleşen Bir Kayıt Mevcuttur!");
                return false;
            }
        }
    }
    else return false;
}
```

Personel Düzenleme Ekranı

Sistemde kayıtlı personellerin düzenlenmesini veya silinmesini sağlayan ekrandır

YetkiliEdit

| | Düzenle | Sil | TC KİMLİK NO | ŞİFRE | AD SOYAD | ÜNVAN |
|---|---------|-----|--------------|-----------|------------------|----------------------|
| ► | Düzenle | Sil | 1 | 1 | Deneme Personel | 1 |
| | Düzenle | Sil | 10203040501 | 123213212 | AA BB | Öğrendi İşleri Da... |
| | Düzenle | Sil | 12391913037 | 123a | deneme personeli | deneme |

TC

AD SOYAD

ŞİFRE

ÜNVAN

Seçileni Düzenle

Personel Silme Kodları

```
1 referenced
public bool deleteMüdür(string tc)
{
    if (!String.IsNullOrEmpty(tc))
    {
        try
        {
            DbController.Connect();
            query = "DELETE FROM müdürler WHERE MüdürTc=@P1";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", tc);

            cmd.ExecuteNonQuery();

            return true;
        }
        catch (MySqlException ex)
        {
            int errorno = ex.Number;
            switch (errorno)
            {
                default:
                    MessageBox.Show(ex.Message.ToString());
                    break;
            }
            return false;
        }
    }
    else return false;
}
```

Personel Düzenleme kodları

```
public bool updateMüdür(string tc, string pw, string name, string title)
{
    try
    {
        DbController.Connect();
        query = "UPDATE müdürler SET adsoyad=@P1,Password=@P2,ünvan=@P3 WHERE MüdürTc=@P4";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", name);
        cmd.Parameters.AddWithValue("@P2", pw);
        cmd.Parameters.AddWithValue("@P3", title);
        cmd.Parameters.AddWithValue("@P4", tc);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch (MySqlException ex)
    {
        int errorno = ex.Number;
        switch (errorno)
        {
            case 2628:
                MessageBox.Show("Girilen bilgiler çok uzun...");
                break;

            default:
                MessageBox.Show(ex.Message.ToString());
                break;
        }
        return false;
    }
}
```

Yeni Öğrenci Kayıt Ekranı

addStudent

×

Tc Kimlik No

Şifre

Ad Soyad

Cinsiyet: ☐ Kadın ☐ Erkek

Sınıf Seçiniz

Bölüm Seçiniz

Sınıfların yüklenmesi için önce bölüm seçiniz

☒ Öğrenci Kaydet

Yeni öğrenci kayıt etme ekranı sisteme kayıt edilen bölüm ve sınıflardan seçilecek olan bilgiler ile öğrenci bilgilerini veritabanına kaydederek yeni öğrenci oluşturur.

```
public bool addStudent(string tc, string pw, string ad, int bolumid, int sinif, int cinsiyet)
{
    if (tc.Length >= 1 && ad.Length >= 1 && sinif.ToString().Length >= 1)
    {
        int sinifid = getsinifbyids(bolumid, sinif.ToString());
        try
        {
            DbController.Connect();
            query = "INSERT INTO ogrenciler values(@P1,@P2,@P3,@P4,@P5,@P6)";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", tc);
            cmd.Parameters.AddWithValue("@P2", pw);
            cmd.Parameters.AddWithValue("@P3", ad);
            cmd.Parameters.AddWithValue("@P4", cinsiyet);
            cmd.Parameters.AddWithValue("@P5", sinifid);
            cmd.Parameters.AddWithValue("@P6", bolumid);

            cmd.ExecuteNonQuery();
            return true;
        }
        catch (MySqlException ex)
        {
            if (ex.Number == 1062)
            {
                MessageBox.Show("Girilen Tc İle Eşleşen Bir Kayıt Mevcuttur!");
                return false;
            }
        }
    }
    else return false;
}
```

Öğrenci Düzenleme Silme Ekranı

Sistemde kayıtlı öğrencilerin düzenlenmesini veya silinmesini sağlayan ekrandır.

OğrenciListesi

| | Düzenle | Sil | TC KİMLİK NO | AD SOYAD | ŞİFRE | CİNSİYET |
|---|---------|-----|--------------|---------------------|-------------|-------------------------------------|
| ▶ | Düzenle | Sil | 00000000000 | Yusuf Kaya | 12334511111 | <input checked="" type="checkbox"/> |
| | Düzenle | Sil | 1 | Deneme Öğrenci | 1 | <input checked="" type="checkbox"/> |
| | Düzenle | Sil | 10000000000 | Veysel Aydın | 123345789 | <input checked="" type="checkbox"/> |
| | Düzenle | Sil | 20000000000 | Halil İbrahim Demir | 4444444112 | <input checked="" type="checkbox"/> |

TC

AD SOYAD

ŞİFRE

Cinsiyet: ☐ Kadın ☐ Erkek

☒ Seçileni Düzenle

1 adet gridviewdan oluşur gridview kaynağı veritabanındaki öğrenciler tablosudur.

```
public bool updateStudent(string tc, string pw, string name, int cinsiyet)
{
    try
    {
        DbController.Connect();
        query = "UPDATE ogrenciler SET ogrenciadsoyad=@P1,OgrenciSifre=@P2,cinsiyet=@P3 WHERE OgrenciTC=@P4";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", name);
        cmd.Parameters.AddWithValue("@P2", pw);
        cmd.Parameters.AddWithValue("@P3", cinsiyet);
        cmd.Parameters.AddWithValue("@P4", tc);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch (MySqlException ex)
    {
        int errorno = ex.Number;
        switch (errorno)
        {
            case 2628:
                MessageBox.Show("Girilen bilgiler çok uzun...");
                break;

            default:
                MessageBox.Show(ex.Message.ToString());
                break;
        }
        return false;
    }
}
```

Öğrenci Bilgi güncelleme kodları

```
public bool deleteStudent(string tc)
{
    if (!String.IsNullOrEmpty(tc))
    {
        try
        {
            DbController.Connect();
            query = "DELETE FROM ogrenciler WHERE OgrenciTc=@P1";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", tc);
            cmd.ExecuteNonQuery();
            return true;
        }
        catch (MySqlException ex)
        {
            int errorno = ex.Number;
            switch (errorno)
            {
                default:
                    MessageBox.Show(ex.Message.ToString());
                    break;
            }
            return false;
        }
    }
    else return false;
}
```

Öğrencinin Sistemden silinmesini sağlayan kod.

Akademisyen Ekleme Ekranı

addAkademisyen

Tc Kimlik No

Şifre

Ad Soyad

Bölüm Seçiniz

✓

✓ Akademisyen Ekle

Akademisyen Bilgileri ve bölüm girilerek akademisyen sisteme kayıt edilir.

```
public bool addAkademisyen(string tc, string pw, string ad, int bolumid)
{
    if (tc.Length == 11 && ad.Length >= 1 && bolumid != -1)
    {
        try
        {
            DbController.Connect();
            query = "INSERT INTO akademisyenler values(@P1,@P2,@P3,@P4)";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", tc);
            cmd.Parameters.AddWithValue("@P2", pw);
            cmd.Parameters.AddWithValue("@P3", ad);
            cmd.Parameters.AddWithValue("@P4", bolumid);

            cmd.ExecuteNonQuery();
            return true;
        }
        catch (MySqlException ex)
        {
            if(ex.Number==1062)
            {
                MessageBox.Show("Girilen Tc ile Eşleşen Bir Kayıt Mevcuttur!");
                return false;
            }
        }
    }
    else return false;
}
```


Akademisyen Düzenleme Ekranı

Akademisyen düzenleme sayfası öğrenci düzenleme sayfasının bire bir aynısıdır sadece varlık olarak öğrenci değil akademisyen ele alınmıştır.

AkademisyenListesi

| | Düzenle | Sil | TC KİMLİK NO | ŞİFRE | AD SOYAD |
|---|---------|-----|--------------|----------|------------------|
| ▶ | Düzenle | Sil | 1 | 1 | Deneme Akadem... |
| | Düzenle | Sil | 12120030442 | 123a | Aziz Harman |
| | Düzenle | Sil | 12345678900 | 123456 | Abidin Çalışkan |
| | Düzenle | Sil | 13456789000 | 123 | Ramazan Tekin |
| | Düzenle | Sil | 34521678900 | 71234565 | Süleyman Kardeş |

TC

AD SOYAD

ŞİFRE

Seçileni Düzenle

Akademisyen görüntüleme düzenleme ve silme işlemlerinin kodları

```
public bool updateAkademisyen(string tc, string pw, string name)
{
    try
    {
        DbController.Connect();
        query = "UPDATE akademisyenler SET OgretmenAdSoyad=@P1,pw=@P2 WHERE OgretmenTc=@P3";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", name);
        cmd.Parameters.AddWithValue("@P2", pw);
        cmd.Parameters.AddWithValue("@P3", tc);

        cmd.ExecuteNonQuery();

        return true;
    }
    catch (MySqlException ex)
    {
        int errorno = ex.Number;
        switch (errorno)
        {
            case 1028:
                MessageBox.Show("Girilen bilgiler çok uzun...");
                break;
            default:
                MessageBox.Show(ex.Message.ToString());
                break;
        }
        return false;
    }
}

public bool deleteAkademisyen(string tc)
{
    if (!string.IsNullOrEmpty(tc))
    {
        try
        {
            DbController.Connect();
            query = "DELETE FROM akademisyenler WHERE OgretmenTc=@P1";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", tc);
            cmd.ExecuteNonQuery();
            return true;
        }
        catch (MySqlException ex)
        {
            int errorno = ex.Number;
            switch (errorno)
            {
                default:
                    MessageBox.Show(ex.Message.ToString());
                    break;
            }
            return false;
        }
    }
}
```

Bölüm Ekleme Ekranı

AddBolum

Bolum Adi

Bölüm Adı Giriniz

Bolum Toplam Yıl Seçiniz

✓ BÖLÜM EKLE

Bölüm adı ve bölüm yılı seçilerek sisteme hem bölüm hem de gerekli sınıflar tanımlanır.

```
public bool addnewBolum(string bolumadi, int bolumyil)
{
    try
    {
        DbController.Connect();
        query = "INSERT INTO bolumlar(bolumadi,toplamyil) values(@P1,@P2)";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", bolumadi);
        cmd.Parameters.AddWithValue("@P2", bolumyil);
        cmd.ExecuteNonQuery();
        if (addClasses(bolumyil, bolumadi))
            return true;
        else return false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return false;
    }
}

1 reference
public bool addClasses(int sinifsayisi, string bolumad)
{
    int bolumid = getbolumidbyname(bolumad);
    try
    {
        for (int i = 1; i <= sinifsayisi; i++)
        {
            query = "INSERT INTO siniflar(sinif,bolumid) values(@P1,@P2)";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", i);
            cmd.Parameters.AddWithValue("@P2", bolumid);
            cmd.ExecuteNonQuery();
        }
        return true;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return false;
    }
}

1 reference
```

Ders Ekleme ve Ders Listesi Ekranları

Ders Adı Giriniz

Akts Seçiniz

Bölüm Seçiniz

Sınıf Seçiniz

Sınıfların yüklenmesi için önce bölüm seçiniz

Ders Sorumlusu Seçiniz

| | TC KİMLİK NO | ŞİFRE | AD SOYAD |
|---|--------------|----------|-----------------|
| ▶ | 1 | 1 | Deneme Akader |
| | 12120030442 | 123a | Aziz Harman |
| | 12345678900 | 123456 | Abidin Çalışkan |
| | 13456789000 | 123 | Ramazan Tekin |
| | 34521678900 | 71234565 | Süleyman Karde |

Ders Ekle

Ders adı ders sorumlu akademisyeni bölüm ve sınıf seçilerek sisteme ders kaydı yapılabilir.

```
public bool addDers(string dersadi, string akts, int bolumid, string sinif,string sorumlutc)
{
    if (dersadi.Length >= 1 && sinif.Length >= 1 && bolumid.ToString().Length >= 1)
    {
        int sinifid = getsinifidbyids(bolumid, sinif.ToString());
        try
        {
            DbController.Connect();
            query = "INSERT INTO dersler(dersad,dersbolumid,derssinifid,akts,sorumluAkademisyenTc) values(@P1,@P2,@P3,@P4,@P5)";
            cmd = new MySqlCommand(query, DbController.conn);
            cmd.Parameters.AddWithValue("@P1", dersadi);
            cmd.Parameters.AddWithValue("@P2", bolumid);
            cmd.Parameters.AddWithValue("@P3", sinifid);
            cmd.Parameters.AddWithValue("@P4", akts);
            cmd.Parameters.AddWithValue("@P5", sorumlutc);

            cmd.ExecuteNonQuery();
            return true;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.ToString());
            return false;
        }
    }
    else return false;
}
```

Not Giriş Ekrani

Not girişi ekranı akademisyenlerin sorumlu olduğu derslere kayıtlı olan öğrencilere sınav/proje notu girmesini sağlayan ekrandır.

Akademisyenden İlk önce ders seçmesi istenir ders seçildikten sonra tabloda o dersi alan öğrenciler gösterilir akademisyen tablodan öğrenci seçtikten sonra öğrenci sınav notu girebilir.

Not_Giriş

| | TC KİMLİK NO | AD SOYAD | CİNSİYET |
|---|--------------|---------------------|-------------------------------------|
| | 00000000000 | Yusuf Kaya | <input checked="" type="checkbox"/> |
| | 1 | Deneme Öğrenci | <input checked="" type="checkbox"/> |
| | 10000000000 | Veysel Aydın | <input checked="" type="checkbox"/> |
| ▶ | 20000000000 | Halil İbrahim Demir | <input checked="" type="checkbox"/> |

Ders Seçiniz

Veri Yapıları

20000000000

Halil İbrahim Demir

☐ Vize ☒ Final ☐ Bütünleme

100

☒ Notu Kaydet

```
public bool addNot(string tc, string notturu, string dersad,int not)
{
    int dersid = getdersidbydersname(dersad);
    try
    {
        DbController.Connect();
        query = "INSERT INTO notlar (ogrencitc,dersid,notturu,puan) values(@P1,@P2,@P3,@P4)";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", tc);
        cmd.Parameters.AddWithValue("@P2", dersid);
        cmd.Parameters.AddWithValue("@P3", notturu);
        cmd.Parameters.AddWithValue("@P4", not);

        cmd.ExecuteNonQuery();
        return true;
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message.ToString());
        return false;
    }
}
```

Devamsızlık Giriş Ekranı

Öğrencilerin derslerde yaptığı devamsızlık günlerini akademisyen tarafından sisteme girilebilmesini sağlayan ekrandır.

Devamsızlık Giriş

| | TC KİMLİK NO | AD SOYAD | CİNSİYET |
|---|--------------|---------------------|-------------------------------------|
| ▶ | 00000000000 | Yusuf Kaya | <input checked="" type="checkbox"/> |
| | 1 | Deneme Öğrenci | <input checked="" type="checkbox"/> |
| | 10000000000 | Veysel Aydın | <input checked="" type="checkbox"/> |
| | 20000000000 | Halil İbrahim Demir | <input checked="" type="checkbox"/> |

Ders Seçiniz

Veri Yapıları

Seçilen Öğrenci TC KİMLİK NO

Seçilen Öğrenci Ad Soyad

24-08-2021

Devamsızlık Kaydet

```
public bool adddevamsızlık(string tc, string tarih, string dersad)
{
    int dersid = getdersidbydersname(dersad);
    try
    {
        DbController.Connect();
        query = "INSERT INTO devamsızlık (ogrencitc,tarih,dersid) values(@P1,@P2,@P3)";
        cmd = new MySqlCommand(query, DbController.conn);
        cmd.Parameters.AddWithValue("@P1", tc);
        cmd.Parameters.AddWithValue("@P2", tarih);
        cmd.Parameters.AddWithValue("@P3", dersid);
        cmd.ExecuteNonQuery();
        return true;
    }
    catch (MySqlException ex)
    {
        MessageBox.Show(ex.Message.ToString());
        return false;
    }
}
```

Sorumlu Olduğum Dersler Ekranı

Akademisyene tanımlanmış olan derslerin listesini gösteren ekrandır.

| Sorumlu Olduğum Dersler | | | | | |
|-------------------------|--------------|-------------------------|-----------|----------------------------------|-------|
| | TC KİMLİK NO | DERS ADI | AKTS YÜKÜ | BÖLÜM ADI | SINIF |
| ▶ | 1 | Veri Yapıları | 4 | BİLGİSAYAR MÜHENDİSLİĞİ | 3 |
| | 1 | Diferansiyel Denklemler | 6 | Elektrik Elektronik Mühendisliği | 2 |

```
public DataSet getSorumluDersler()
{
    try
    {
        DbController.Connect();
        query = @"SELECT dersler.sorumluAkademisyenTc AS `TC KİMLİK NO`,akademisyenler.OgretmenTC,dersler.dersad AS `DERS ADI`,
dersler.aks AS `AKTS YÜKÜ`,bolumler.bolumadi AS `BÖLÜM ADI`,siniflar.sinif AS SINIF
FROM
dersler INNER JOIN siniflar
ON siniflar.sinifid = dersler.derssinifid
INNER JOIN bolumler
ON bolumler.bolumid = dersler.dersbolumid
INNER JOIN akademisyenler
ON akademisyenler.OgretmenTC = dersler.sorumluAkademisyenTc
WHERE dersler.sorumluAkademisyenTc = @P1";
        da = new MySqlDataAdapter(query, DbController.conn);
        da.SelectCommand.Parameters.AddWithValue("@P1", Giriş.loginusertc);

        da.Fill(ds, "müdürler");
        DbController.Disconnect();
        return ds;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return ds;
    }
}
```

Duyuru Gönderme ve Gönderilen Duyuruları Listeleme Ekranları

Akademisyenin öğrencileriyle arasındaki iletişimi sağlayan ekrandır akademisyenin gönderdiği duyurular eş zamanlı olarak öğrenci panelinde okunabilmektedir.

Duyuru_Gönder

Ders Seçiniz

Veri Yapıları

Duyuru Metni Max 250 Karakter

Duyuru Gönder

| Gönderilen Duyurular | | | |
|----------------------|---|-------|-------------------------|
| | Duyuru Metni | Sınıf | Ders Adı |
| ► | DERSE GELİRKEN 3.SLAYTA ÇALIŞMAYI UNUTMAYINIZ QUIZ YAP... | 3 | Veri Yapıları |
| | bu hafta integral konusuna geçeceğiz hazırlanıp geliniz | 2 | Diferansiyel Denklemler |
| | test duyuru123 | 3 | Veri Yapıları |

Sınav Sonuçları Ekranı

Akademisyenler tarafından öğrencilere verilen notların öğrenciler tarafından görülmesi sağlayan ekrandır.

| Sınav Sonuçları | | | | | | |
|-----------------|--------|---------------|----------|------|-----------|------------------|
| | dersid | Ders Adı | Not Türü | Puan | Akts Yüğü | Ders Sorumlusu |
| ▶ | 7 | Veri Yapıları | Vize | 100 | 4 | Deneme Akadem... |
| | 7 | Veri Yapıları | Final | 65 | 4 | Deneme Akadem... |

```
public DataSet getSınavSonuç(string tc)
{
    try
    {
        DbController.Connect();
        query = @"SELECT
dersler.dersid,
dersler.dersad AS `Ders Adı`,
notlar.notturu AS `Not Türü`,
notlar.puan AS Puan,
dersler.akts AS `Akts Yüğü`,
akademisyenler.OgretmenAdSoyad AS `Ders Sorumlusu`
FROM dersler
INNER JOIN notlar
ON dersler.dersid = notlar.dersid
INNER JOIN akademisyenler
ON dersler.sorumluAkademisyenTc = akademisyenler.OgretmenTc
WHERE ogrencitc=@P1";
        da = new MySqlDataAdapter(query, DbController.conn);
        da.SelectCommand.Parameters.AddWithValue("@P1", tc);
        da.Fill(ds, "ogrenciler");
        DbController.Disconnect();
        return ds;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return ds;
    }
}
```


Devamsızlık Görüntüleme Ekranı

Öğrencinin derslerde yaptığı devamsızlıkları görüntüleyebileceği ekrandır.

| Devamsızlık Kayıtları | | | | |
|-----------------------|-----|---------------|------------------|---------------------|
| | Kod | Ders Adı | Ders Sorumlusu | Tarih |
| ▶ | 1 | Veri Yapıları | Deneme Akadem... | 24.08.2021 00:34:24 |
| | 3 | Veri Yapıları | Deneme Akadem... | 4.08.2021 00:34:24 |

```
public DataSet getDevamsızlık(string tc)
{
    try
    {
        DbController.Connect();
        query = @"SELECT
devamsızlık.devamsızlıkno AS Kod,
dersler.dersad AS `Ders Adı`,
akademisyenler.OgretmenAdSoyad AS `Ders Sorumlusu`,
devamsızlık.tarih AS Tarih
FROM devamsızlık
INNER JOIN dersler
ON devamsızlık.dersid = dersler.dersid
INNER JOIN akademisyenler
ON dersler.sorumluAkademisyenTc = akademisyenler.OgretmenTc
WHERE devamsızlık.ogrencitc = @P1";
        da = new MySqlDataAdapter(query, DbController.conn);
        da.SelectCommand.Parameters.AddWithValue("@P1", tc);
        da.Fill(ds, "Devamsızlık");
        DbController.Disconnect();
        return ds;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return ds;
    }
}
```

Duyuru Görüntüleme Ekranı (Öğrenci)

Akademisyen tarafından öğrencilere gönderilen duyuruların öğrenciler tarafından görüntülenmesini sağlayan ekrandır.

| Duyurular | | | |
|---------------------|--------------------|---|---------------|
| Sorumlu Akademisyen | Ders Adı | Duyuru Metni | Bölüm |
| Deneme Akadem... | Veri Yapıları | DERSE GELİRKEN 3.SLAYTA ÇALIŞMAYI UNUTM... | BİLGİSAYAR MÜ |
| Süleyman Kardaş | Nesneye Dayalı ... | dduyuru mesajı deneme dduyuru mesajı denem... | BİLGİSAYAR MÜ |
| Süleyman Kardaş | Nesneye Dayalı ... | dduyuru mesajı deneme dduyuru mesajı denem... | BİLGİSAYAR MÜ |
| Deneme Akadem... | Veri Yapıları | test duyuru123 | BİLGİSAYAR MÜ |

```
public DataSet getDuyurular(string tc)
{
    int sinifid = getOgrenciClassByTc(tc);
    try
    {
        DbController.Connect();
        query = @"SELECT
akademisyenler.OgretmenAdSoyad AS `Sorumlu Akademisyen`,
dersler.dersad AS `Ders Adı`,
duyurular.duyurumetni AS `Duyuru Metni`,
bolumler.bolumadi AS Bölüm
FROM akademisyenler
INNER JOIN duyurular
ON akademisyenler.OgretmenTC = duyurular.akademisyentc
INNER JOIN dersler
ON duyurular.dersid = dersler.dersid
INNER JOIN bolumler
ON dersler.dersbolumid = bolumler.bolumid where duyurular.sinifid=@P1";
        da = new MySqlDataAdapter(query, DbController.conn);
        da.SelectCommand.Parameters.AddWithValue("@P1", sinifid);
        da.Fill(ds, "Duyurular");
        DbController.Disconnect();
        return ds;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.ToString());
        return ds;
    }
}
```

EKLER

Proje Github Linki

<https://github.com/kayalogg/OgrenciBilgiSistemiStaj>