DÜZCE ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ GÖRSEL PROGRAMLAMA
PROJE RAPORU

172113036 MELİS İREM KAYA

BMT303 – GÖRSEL PROGRAMLAMA

OCAK 2021

İÇİNDEKİLER

ŞEKİLLER TABLOSU

ÖZET

Görsel Programlama dersi proje konusu Kütüphane Otomasyonudur. Proje C# programlama dili ile yazılmıştır. Veritabanı olarak Microsoft Access 2010 kullanılmıştır. Bu raporda ise katmanlı mimari kavramının ne anlama geldiği ve projede katmanlı mimarinin kullanılması detaylı olarak açıklanmıştır.

Anahtar Kelimeler: Katmanlı mimari, kütüphane, otomasyon, öğrenciler, üyeler, kitaplar, alınan kitaplar

1. GİRİŞ

1.1. KULLANILAN DİLLER

1.1.1. C#

C#; Microsoft tarafından .NET Teknolojisi için geliştirilen modern bir programlama dilidir. Sözdizimi C-like (C benzeri) bir deneyim sunar. Microsoft tarafından geliştirilmiş olsa da ECMA ve ISO standartları altına alınmıştır. C programlama dilinde bir tam sayı değişkeni 1 artırmak için değişkenden sonra "++" eki kullanılır. C++ dilinin adı, C diliyle Nesne Yönelimli Programlama yapabilme olanağı (C with Classes) için eklentiler sağladığı için "C++" şeklindedir. Benzer şekilde C++ diline yeni eklentiler yapılarak ((C++)++) bir adım daha da ileriye götürülmüş ve tamamen nesneye yönelik tasarlanmış C# dilinin isimlendirilmesinde, + karakterlerinin birbirlerine yakınlaşmış hali ve bir melodi anahtarı olan C# Major kullanılmıştır. Nesne yönelimli programlama kavramının gelişmesine katkıda bulunan aktif programlama dillerinden biridir. C#, .NET orta seviyeli programlama dillerindendir. Yani hem makine diline hem de insan algısına eşit seviyededir.

1.1.2. Microsoft Access

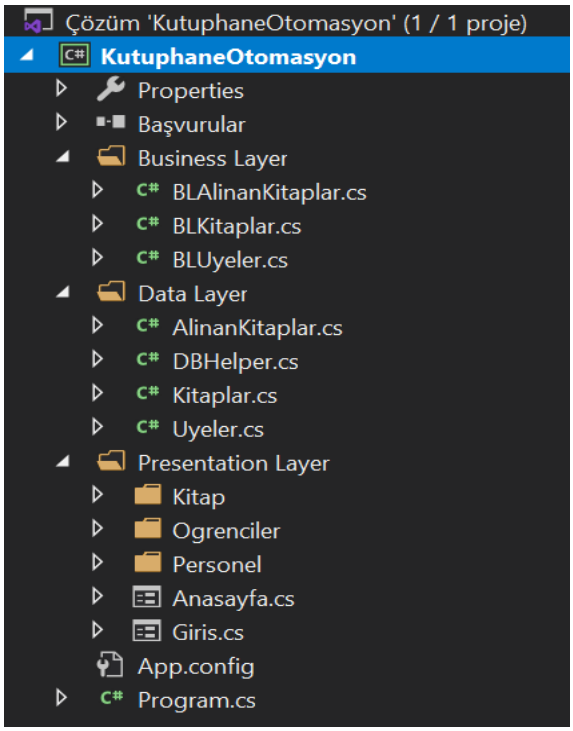
Microsoft Access ya da Microsoft Office Access Microsoft'un ilişkisel veri tabanı yönetim sistemidir. Veri tabanı yönetim sistemleri arasına Access çok sonradan girmiş olmasına rağmen bu alanda önemli ölçüde başarı sağlayarak küçük ölçekli veri tabanları için çok kullanılan bir paket haline gelmiştir. Bunda, Access'in yazılım araçlarının yüksek kullanıcı kolaylığına sahip olmasının etkisi büyüktür. Çoğu zaman hiç tasarım ortamına girmeden, sadece sihirbazlar kullanılarak veri tabanı dosyaları hazırlanabilir. Access'in iki yüzü vardır. Bunlardan birinde hiç program kodu kullanmadan veri tabanı hazırlamak mümkündür.

2. GELİŞME

2.1. C# KATMANLI MİMARİ KULLANIMI

Katmanlı Mimari projelerimizin daha derli toplu durmasını sağlayan, kodun okunulabilirliğini arttıran, ekip çalışmasını arttıran, hata yönetiminin kolay olmasını sağlayan bir yapıdır. Aslında bu yapı ile proje yazımını bir standart hale getirmiş oluyoruz. Bu yapı 3 ana katmandan oluştuğu gibi günümüzde Çok Katmanlı Mimari yapıda denebilir. Ama aslında 3 ana katman üzerinde durur. Bu katmanlar;

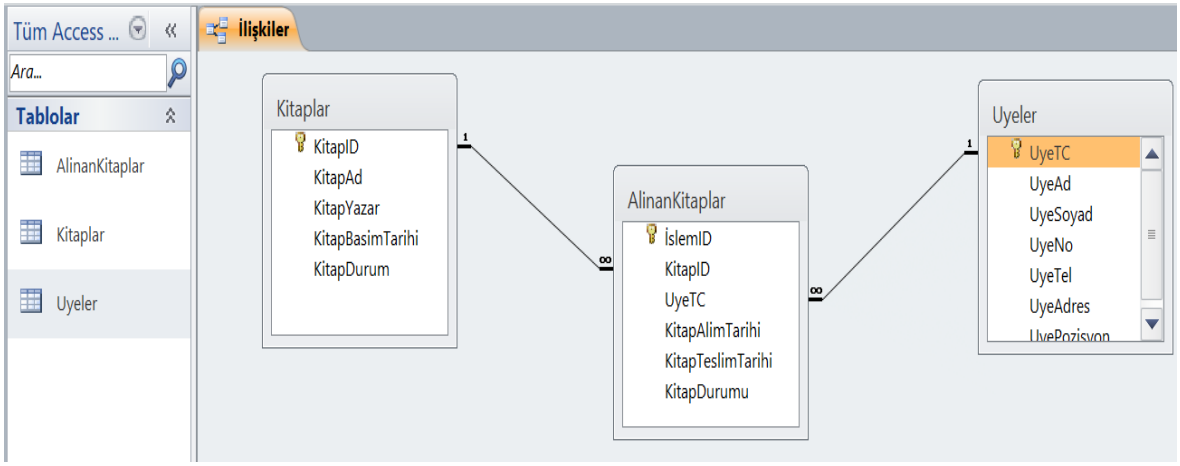
1. Veri Katmanı (Data Layer)
2. İş Katmanı (Business Layer)
3. Sunum Katmanı (Presentation Layer) dır.



Şekil 1 Visual Studio Çözüm Gezgini

2.1.1. VERİ KATMANI (DATA LAYER)

Bu katman adından da anlaşılacağı üzere veriler ve veritabanı ile uğraşacağımız bölümdür. Bu katmanda veri tabanı bağlantısı yapılıp, veri tabanında bulunan tablolar için sınıflar oluşturulur. Bunun için veritabanının bulunması gerekiyor. Kütüphane Otomasyonu projesinin veritabanı için Microsoft Access 2010 da Database1 adlı bir veritabanı oluşturulur.



Şekil 2 Veritabanı İlişkiler

Visual Studio C# Form Application da DataLayer isimli klasör oluşturulduktan sonra bu klasörün içine DBHelper.cs isminde class oluşturulur. DBHelper classında ilk önce veritabanı bağlantısı yapılması gerekmektedir. Kütüphaneye ise `using System.Data.OleDb;` eklenmelidir. Bağlantı adresi yazıldıktan sonra bağlantı açılır. Ardından verileri çekmek için bir sorgu oluşturulur. Farklı tablolardan veri çekip alabilmek için bir fonksiyon oluşturulur.

2.1.1.1. DBHelper.cs

```
    class DBHelper
    {
        OleDbCommand sorgu;
        DataTable dt;
        OleDbDataAdapter adp;
        public OleDbConnection BaglantiAc()
        {
            OleDbConnection baglanti = new
OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=|DataDirectory|\Database1.accdb;Persist Security Info=False;"); //Veritabanı
bağlantısı
            baglanti.Open();//Bağlantı açma
            return baglanti;
        }
        public void BaglantiKapa()//Bağlantı kapama
        {
            OleDbConnection baglanti = BaglantiAc();
            baglanti.Close();
        }
        //CRUD
        //CREATE READ UPDATE DELETE
        public int SorguCalistir(string komut) // Create,update,delete işlemlerinin
yapıldığı sorgu
        {
            sorgu = new OleDbCommand(komut, BaglantiAc());//Sorgu oluşturulup
bağlantısı belirtilir ve açılır
            int kontrol = sorgu.ExecuteNonQuery();//İşlem yapılıp yapılmadığı
kontrol edilir
            BaglantiKapa();//Bağlantı kapatılır
            return kontrol;
        }
```

```
        public DataTable ListeleSorgusu(string komut)//Read işleminin yapıldığı
fonksiyondur
        {
            dt = new DataTable(); //Datatable oluşturulur
            sorgu = new OleDbCommand(komut, BaglantiAc());
            adp = new OleDbDataAdapter(sorgu);//Veri kaynağı sorgu olarak
belirtilir
            adp.Fill(dt);//Datatable içine adapterdaki veriler atanır
            BaglantiKapa();//Bağlantı kapatılır
            return dt;
        }
```

Veritabanı bağlama ve sorgu işlemleri yapıldıktan sonra hangi tablolarla işlem yapılacaksa o tabloları eklemeniz gerekmektedir. Kütüphane Otomasyonu için Kitaplar, Ogrenciler ve AlınanKitaplar tablosu ile ilgilenileceği için DataLayer klasörüne Kitaplar.cs, Uyeler.cs ve AlinanKitaplar.cs olmak üzere üç class daha oluşturulur. Veritabanında hangi sütun varsa buraya da aynı tip olacak şekilde eklenir. “prop” yazarak kolay bir şekilde oluşturulabilir. Sonrasında ise bir Constructor oluşturularak veriler eklenir.

2.1.1.2. Kitaplar.cs

```
class Kitaplar
    {
        public string KitapID { get; set; }
        public string KitapAd { get; set; }
        public string KitapYazar { get; set; }
        public string KitapBasimTarihi { get; set; }
        public string KitapDurum { get; set; }

        public Kitaplar(
            string KitapID,
            string KitapAd,
            string KitapYazar,
            string KitapBasimTarihi,
```

```
            string KitapDurum)
        {
            this.KitapID = KitapID;
            this.KitapAd = KitapAd;
            this.KitapYazar = KitapYazar;
            this.KitapBasimTarihi = KitapBasimTarihi;
            this.KitapDurum = KitapDurum;
        }
    }
```

2.1.1.3. Uyeler.cs

```
class Uyeler
    {
        public string UyeTC { get; set; }
        public string UyeAd { get; set; }
        public string UyeSoyad { get; set; }
        public string UyeNo { get; set; }
        public string UyeTel { get; set; }
        public string UyeAdres { get; set; }
        public string UyePozisyon { get; set; }


        public Uyeler(
            string UyeTC,
            string UyeAd,
            string UyeSoyad,
            string UyeNo,
            string UyeTel,
            string UyeAdres,
            string UyePozisyon
            )

        {
            this.UyeTC = UyeTC;
```

```
            this.UyeAd = UyeAd;
            this.UyeSoyad = UyeSoyad;
            this.UyeNo = UyeNo;
            this.UyeTel = UyeTel;
            this.UyeAdres = UyeAdres;
            this.UyePozisyon = UyePozisyon;

        }
       }
```

2.1.1.4. AlinanKitaplar.cs

```
class AlinanKitaplar
{
    public string KitapID { get; set; }
    public string UyeTC { get; set; }
    public string KitapAlimTarihi { get; set; }
    public string KitapTeslimTarihi { get; set; }
    public string KitapDurumu { get; set; }

    public AlinanKitaplar(
        string KitapID,
        string UyeTC,
        string KitapAlimTarihi,
        string KitapTeslimTarihi,
        string KitapDurumu)
    {
        this.KitapID = KitapID;
        this.UyeTC = UyeTC;
        this.KitapAlimTarihi = KitapAlimTarihi;
        this.KitapTeslimTarihi = KitapTeslimTarihi;
        this.KitapDurumu = KitapDurumu;
    }
  }
```

2.1.2. İŞ KATMANI (BUSINESS LAYER)

Bu katmanda veritabanından verileri hangi sorguyla alacağımızı, verileri nasıl okuyacağımızı, verileri okuduktan sonra ne yapacağımızı ve bütün bu olayları İş Katmanında gerçekleşir. Aslında veritabanıyla projemiz arasında bir köprü oluşturmayı sağlar.

Kütüphane Otomasyonu için Business Layer'da BLKitaplar.cs ve BLOgrenciler.cs classları oluşturularak yapmak istenilen işlemler bu classlarda yapılacaktır. Bu katmanda temel CRUD (Create, Read, Update, Delete) işlemleri yapılacaktır.

2.1.2.1. BLKitaplar.cs

BLKitaplar.cs de öncelikle veri katmanında oluşturulan bağlantı açma ve sorgu yazma fonksiyonlarını alabilmek için DBHelper çağırılması gerekir. DBHelper DataLayer klasöründe olduğundan DataLayer kütüphaneye eklenmelidir. Aynı zamanda veritabanı işlemleri de yapılacağından kütüphaneye

```
using KutuphaneOtomasyon.Data_Layer;
using System.Data.OleDb;
using System.Data;
```

eklenmelidir. Ardından BLKitaplar classında DBHelper ve OleDbCommand çağırılır.

Birbirini tekrar eden kodların olmaması için kitaplar için yapılacak işlemleri tek bir fonksiyonda yazıyoruz. İşlemlerin seçilebilmesi için switch case kullanmak işlerimizi kolaylaştıracaktır.

```
    class BLKitaplar
    {
        DBHelper veritabanı = new DBHelper(); //Veritabanı işlemi yapılacağından
dbhelper çağırılır
        string komut;
```

```
        public int KitapIslem(Kitaplar kitaplar, int islem)
        {
            //CRUD
            //0123
            string id = kitaplar.KitapID;
            string kad = kitaplar.KitapAd;
            string kyazar = kitaplar.KitapYazar;
            string kbasim = kitaplar.KitapBasimTarihi;
            string kdurum = kitaplar.KitapDurum;

            switch (islem)
            {
                case 0://c ekle
                    komut = "INSERT INTO Kitaplar(KitapID, KitapAd, KitapYazar,
KitapBasimTarihi, KitapDurum) VALUES('" + id + "','" + kad + "','" + kyazar + "','"
+ kbasim + "','" + kdurum + "')";
                    break;
                case 1://r listeleme
                    KitapListele("","","");
                    break;
                case 2://u güncelleme
                    komut = "UPDATE Kitaplar SET  KitapAd = '" + kad + "',
KitapYazar = '" + kyazar + "', KitapBasimTarihi = '" + kbasim + "', KitapDurum = '"
+ kdurum + "' WHERE KitapID = '" + id + "'";
                    break;
                case 3://d silme
                    komut = "DELETE FROM Kitaplar WHERE KitapID = '" + id + "'";
                    break;
            }
            return veritabanı.SorguCalistir(komut);
        }
        public int KitapDurumGuncelle(string kitapid) //kitap durumu güncelleme
        {
```

```
            return veritabanı.SorguCalistir("UPDATE Kitaplar SET KitapDurum = '1'
where KitapID = '" + kitapid + "'");
        }
        public DataTable KitapIsmeGoreListele(string kitapadi) //arama butonu için
listeleme
        {
            return veritabanı.ListeleSorgusu("SELECT * FROM Kitaplar WHERE KitapAd
LIKE '%" + kitapadi + "%'");
        }
        public DataTable KitapListele(string uno, string utc, string kitapid)
//Öğrenci numaralarına ve kitapid ye göre kitap listeleme fonksiyonu
        {
            if (string.IsNullOrEmpty(kitapid))//öğrenci no ve tc ye göre listeleme
            {
                komut = "SELECT
K.KitapID,K.KitapAd,A.KitapAlimTarihi,A.KitapTeslimTarihi,A.KitapDurumu,A.İslemID
FROM Uyeler U,AlinanKitaplar A,Kitaplar K  WHERE U.UyeTC =A.UyeTC AND A.KitapID =
K.KitapID AND U.UyeTC = '" + utc + "' AND U.UyeNo='" + uno + "'";
            }
             if(string.IsNullOrEmpty(uno) && string.IsNullOrEmpty(utc)) //kitapid
ye göre listeleme
            {
                komut = "SELECT * FROM Kitaplar where KitapID = '" + kitapid + "'";
            }
             if(string.IsNullOrEmpty(uno) && string.IsNullOrEmpty(utc)&&
string.IsNullOrEmpty(kitapid)) //genel kitap listesi
            {
                komut= "SELECT * FROM Kitaplar";
            }
            return veritabanı.ListeleSorgusu(komut);
        }
        public DataTable KitapDurumListele()//Boşta olan kitapların listesi
        {
```

```
            return veritabanı.ListeleSorgusu("SELECT * FROM Kitaplar where
KitapDurum ='0'");
        }

        public DataTable KitapSayisi()//Tüm kitap sayısı
        {
            komut = ("SELECT COUNT(*) FROM Kitaplar");
            return veritabanı.ListeleSorgusu(komut);
        }

        public DataTable KiralananlarSayisi() //Dolu kitap sayısı
        {
            komut = ("SELECT COUNT(*) FROM Kitaplar WHERE KitapDurum='1'");
            return veritabanı.ListeleSorgusu(komut);
        }

        public DataTable BosKitapSayisi() //Boş kitap sayısı
        {
            komut = ("SELECT COUNT(*) FROM Kitaplar WHERE KitapDurum='0'");
            return veritabanı.ListeleSorgusu(komut);
        }
        public int KitapDurumGuncelleIade(string kid) //İade işlemi yapıldıktan
sonra durum güncelleme
        {
            komut = ("UPDATE Kitaplar SET KitapDurum ='0' WHERE KitapID =
'"+kid+"'");
            return veritabanı.SorguCalistir(komut);
        }
    }
```

2.1.2.2. BLUyeler.cs

BLUyeler classında BLKitaplar da yapılan işlemlerin aynılarının sadece verileri değiştirilerek yazılır.

```
class BLUyeler
{
    DBHelper veritabanı = new DBHelper();
    string komut;
    //CRUD
    //0123
    public int UyeIslem(Uyeler Uyeler, int islem, int pzsyn)
    {
        string tc = Uyeler.UyeTC;
        string ad = Uyeler.UyeAd;
        string soyad = Uyeler.UyeSoyad;
        string no = Uyeler.UyeNo;
        string tel = Uyeler.UyeTel;
        string adres = Uyeler.UyeAdres;
        string pozisyon = Convert.ToString(pzsyn);

        switch (islem)
        {
            case 0://c ekle
                komut ="INSERT INTO Uyeler(UyeTC, UyeAd, UyeSoyad, UyeNo, UyeTel,
UyeAdres,UyePozisyon) VALUES('" + tc + "','" + ad + "','" + soyad + "','" + no + "','"
+ tel + "','" + adres + "','" + pozisyon + "')";
                break;
            case 1://r listele
                UyeListele(pzsyn,Uyeler.UyeTC,Uyeler.UyeNo);
                break;
            case 2://u güncelle
```

```
                    komut = "UPDATE Uyeler SET  UyeAd = '" + ad + "', UyeSoyad = '" +
soyad + "', UyeNo = '" + no + "', UyeTel = '" + tel + "', UyeAdres = '" + adres + "',
UyePozisyon = '" + pozisyon + "' WHERE UyeTC = '" + tc + "'";
                    break;
                case 3://d sil
                    komut = "DELETE FROM Uyeler WHERE UyeTC = '" + tc + "'";
                    break;
            }
            return veritabanı.SorguCalistir(komut);
        }

        public DataTable UyeListele(int pozisyon,string tc,string no)
        {
            //0 = ÖĞRENCİ
            //1 = PERSONEL
            if (pozisyon == -1) //Öğrenci Listeleme
            {
                komut = "SELECT * FROM Uyeler WHERE UyeNo = '"+no+"' AND UyeTC
='"+tc+"'";
            }
            else
            {
                if (string.IsNullOrEmpty(tc)&&string.IsNullOrEmpty(no))
                {
                    komut = "SELECT * FROM Uyeler WHERE UyePozisyon = " + pozisyon;
                }
                else
                {
                komut = "SELECT * FROM Uyeler WHERE UyeNo = '" + no + "' AND UyeTC ='"
+ tc + "' AND UyePozisyon = " + pozisyon;

                }
            }
```

```
            return veritabanı.ListeleSorgusu(komut);//komut dbhelperda bulunan
listelesorgusuna gönderilir
        }
```

2.1.2.3. BLAlinanKitaplar.cs

```
    class BLAlinanKitaplar
    {
        DBHelper veritabanı = new DBHelper();
        public DataTable KitapAlanlar(string kitapid) //genel listeleme
        {
            return veritabanı.ListeleSorgusu("SELECT * FROM AlinanKitaplar WHERE
KitapID = '" + kitapid + "'");
        }
        public int KitapIade(int islemid) //İade işleminde kitap durumu güncelleme
        {
            return veritabanı.SorguCalistir("UPDATE AlinanKitaplar SET KitapDurumu =
'0'  where İslemID = " + islemid);
        }
        public int KitapAl(string kid, string tc, string atarih, string ttarih, string
durum) //Kitap alınınca alınankitaplara eklenme
        {
            return veritabanı.SorguCalistir("INSERT INTO
AlinanKitaplar(KitapID,UyeTC,KitapAlimTarihi,KitapTeslimTarihi,KitapDurumu) VALUES('"
+ kid + "','" + tc + "','" + atarih + "','" + ttarih + "', '" + durum + "')"); ;
        }
    }
```

2.1.3. SUNUM KATMANI (PRESENTATION LAYER)

Bu katmanda çekilen verileri kullanıcıya sunma katmanıdır. Kütüphane Otomasyonu uygulaması Windows Form Application ile yapılacaktır.

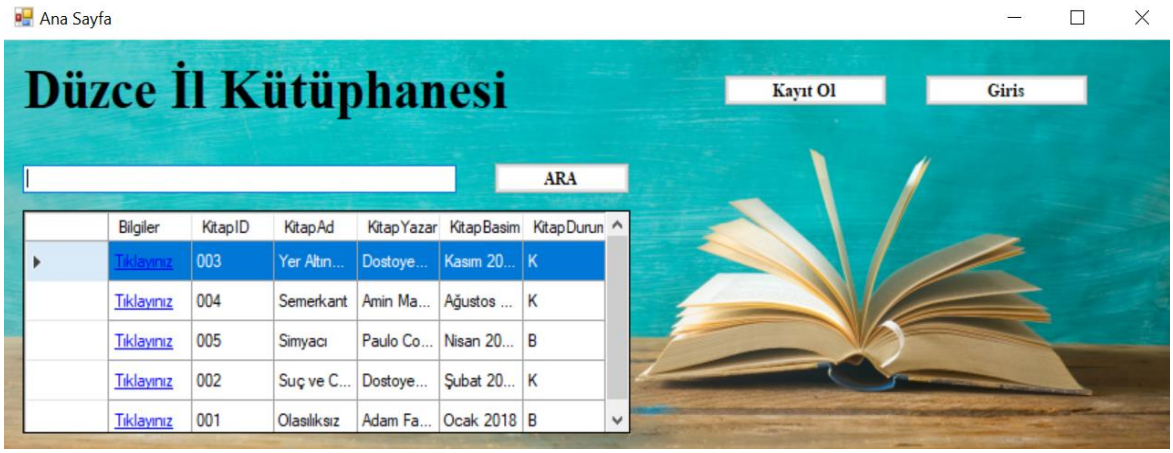
Kullanıcıya bu katmanda ulaşılır. Kullanıcı dilerse veri çekecek, ekleyecek, silecek veya güncelleyecektir. Kullanıcı için yetkilendirme düzeyi bulunmaktadır. Öğrenci için kitap alma ve iade etme işlemleri gerçekleştirilebilecektir. Öğrencinin şimdiye kadar almış olduğu, teslim etmiş ve etmemiş olan kitapların listesi görüntülenecektir. Teslimi geçmiş kitaplar için kırmızı, teslime iki gün kalmış kitaplar için sarı, teslim edilmiş kitaplar içinse yeşil uyarı oluşturulacaktır. Kitaplar için bilgiler, alan kişilerin listesi bulunacaktır. Teslimi geçmiş kitaplar için günlük 1 TL lik ceza işlemi oluşturulacaktır. Kütüphanede bulunan kitapların, öğrenciye verilmiş olan ve verilmemiş olan kitapların sayısı grafik olarak gösterimi yapılacaktır. Son olarak uygulamanın setup dosyayı olacaktır.

Bu katmanda birçok form olduğundan dosya içi dosya oluşturularak hangi formun nerede olduğuna rahatça ulaşılabilmesi hedeflenmiştir. Katmanda bulunan dosyalar: Personel, Kitaplar ve Uyelerdir.

2.1.3.1. Anasayfa

Ana sayfa işlemleri için gerekli tasarımlar yapılması için öncelikle Araç kutusu açılmalıdır. Araç kutusu programın sol tarafında bulunmaktadır. Eğer araç kutusu programın solunda yoksa görünüm panelinden araç kutusuna tıklayarak açabilirsiniz. Araç Kutusundan bir Panel çekilir. Panel tüm sayfayı kaplayacak şekilde ayarlanır ve ardından Özelliklerin BackgroundImage kısmından fotoğraf eklenir. Düzce İl Kütüphane yazısı için araç kutusundan label çekilir. Labelin üzerine sağ tık yaptıktan sonra özelliklere (programın sağ tarafında bulunur) tıkladığınızda labelin fontunu, boyutunu, rengini veya textini değiştirebilirsiniz. Kayıt Ol, Giriş ve Ara butonları için araç kutusundan buttonlar çekilir. Özelliklerden ise buttonların adları değiştirilir.

Bu projede istediğimiz şey ana sayfada kitap arama yapabilmek ve kütüphanede bulunan tüm kitapları görebilmektir. İstediğiniz kitabı aratabilmeniz için gereken şey ise TextBox tır. Kitapların listeleneceği yer ise araç kutusunda bulunan DataGridView dir. DataGridView de bulunan Bilgiler kolonunu eklemek için DataGridView in sağ üstte bulunan oka tıklayarak kolon ekleden, kolon ismine Bilgiler adı verilir. Tüm bu tasarımımız için gerekli araçlar formumuza yerleştirdikten sonra sıra kodlara gelmektedir. İlk olarak Ana sayfada ki kitap arama ve listeleme kısmının kodları yazılacaktır.

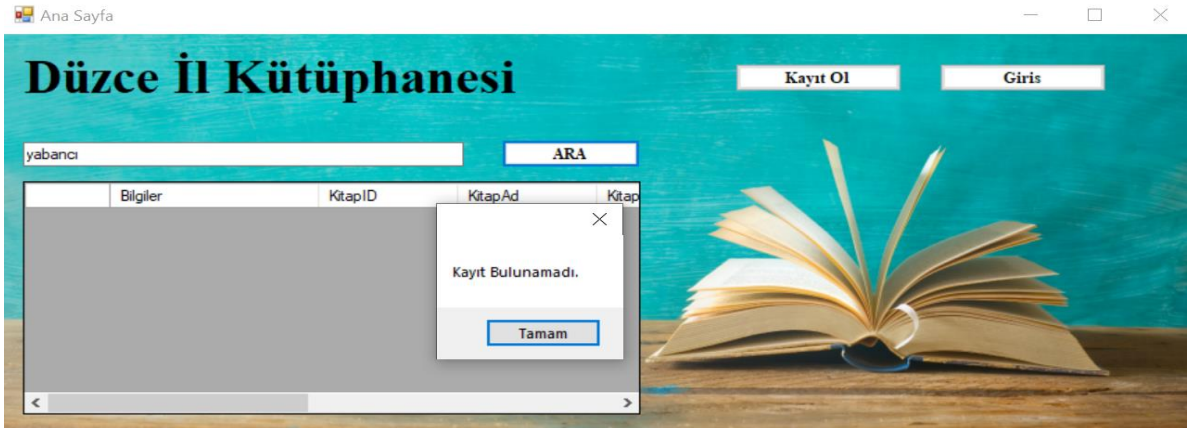


Şekil 3 Anasayfa

Kitaplardan listeleme işlemleri yapılacağından Business Layer dan, `BLKitaplar blkitap = new BLKitaplar();` tanımlanır. Form yüklendiğinde kütüphanede bulunan kitapların listelenmesi için BLKitaplardan KitapListele() fonksiyonu çağırılır. DataGridView1 de bulunan Bilgiler kolonunda her bir satır için Tıklayınız yazdırılır.

```
        private void Form1_Load(object sender, EventArgs e)//Form Yüklendiğinde
        {
            dataGridView1.DataSource = blkitap.KitapListele("", "",
"");//dataGridView1 de BLKitaplarda bulunan kitap listele gelir
            for (int i = 0; i < dataGridView1.Rows.Count; i++)
            {
```

```
                dataGridView1.Rows[i].SetValues("Tıklayınız");//satırda tıklayınız
linki gelir
            }
            }
```

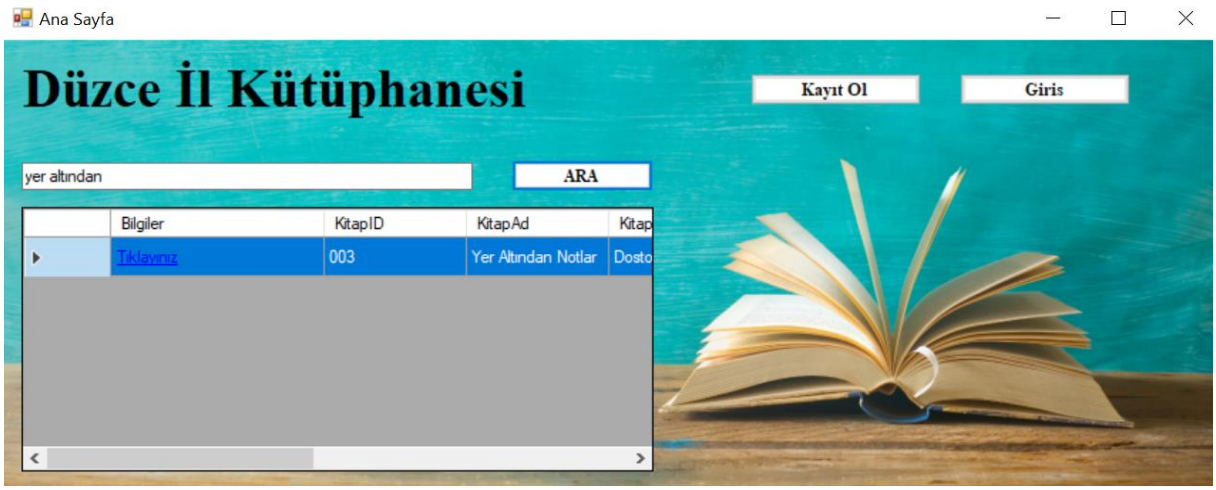


Şekil 4 Kitap Arama

TextBox a yazılan kitap adının arama butonuna tıklandığında kitap yoksa "Kayıt Bulunamadı" uyarısını vermesi eğer kitap var ise o kaydı getirmesi için string kitapadi, textBox1 e eşitlenir. BLKitaplar da bulunan KitapIsmeGoreListele fonksiyonu çağırılır ve bu fonksiyon parametre olarak kitapadi nı alır. Eğer dataGridView1 in satır sayısı, sıfırdan küçük veya eşitse "Kayıt Bulunamadı" uyarısını, değilse o kaydı getirir.

```
        private void button1_Click(object sender, EventArgs e)//Ara butonu
        {
            string kitapadi = textBox1.Text; //textbox a girilen değer kitapadi na
atanır
            dataGridView1.DataSource =
blkitap.KitapIsmeGoreListele(kitapadi);//kitap işlemlerinde bulunan
kitapismegörelistele fonk çağırılır ve kitapadi parametre olarak gönderilir
            if (dataGridView1.RowCount <= 0)
            {
                MessageBox.Show("Kayıt Bulunamadı.");
```

```
            }
            else
            {
                for (int i = 0; i < dataGridView1.Rows.Count; i++)
                {
                    dataGridView1.Rows[i].SetValues("Tıklayınız");
                }
            }
        }
```



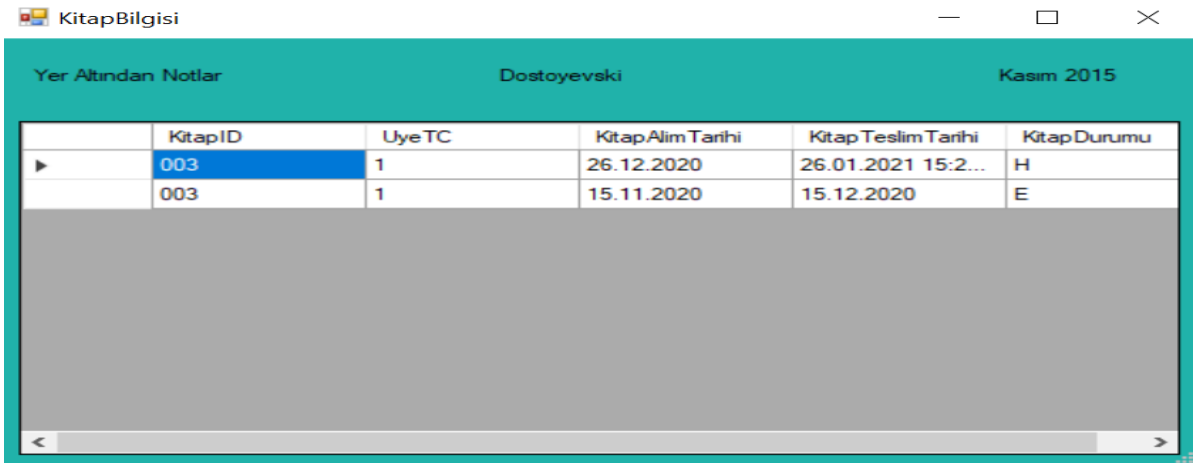
Şekil 5 Kitap Arama

Bilgiler kolonunda bulunan "Tıklayınız" a basıldığında yeni bir form açılacaktır ve o formda kitap bilgileri verilecektir. Bunun için DataGridView1 in özelliklerinde bulunan Events inde Cell Content Click yani Hücre İçeriği Tıklaması seçilir. Seçildikten sonra bu otomatik olarak forma yazılır.

```
        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            Presentation_Layer.Kitaplar.KitapBilgisi kbilgisi = new
Presentation_Layer.Kitaplar.KitapBilgisi();
```

```
            kbilgisi.kid=
dataGridView1.SelectedRows[0].Cells[1].Value.ToString();//dataGridView1 de bulunan
kitapid 1. indistedir ve bu kitapbilgisinin kid ine atanır
            kbilgisi.ShowDialog();//kitap bilgisi formu açma
        }
```

Kitap Bilgisi formunda kitap hakkında bilgiler verildikten sonra kitabı alan öğrencilerin listesi verilecektir. Bunun için KitapBilgisi formunda yazılacak kodlar aşağıda verilmiştir.



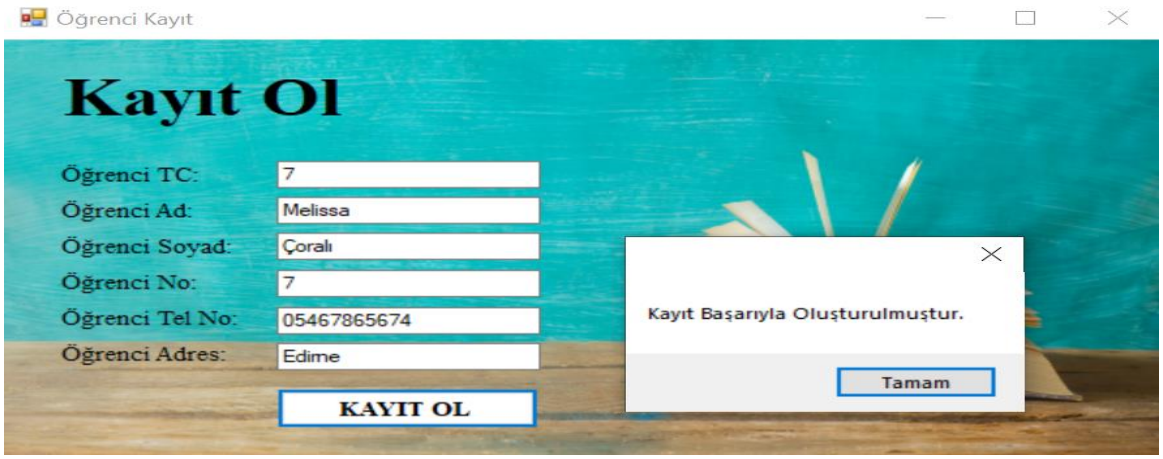
Şekil 6 Kitap Bilgisi

```
        public string kid;
        BLKitaplar blkitaplar = new BLKitaplar(); //listele komutu burada
olduğundan blkitaplar çağırılır
        BLAlinanKitaplar blalinankitaplar = new BLAlinanKitaplar();//kitabı alan
öğrenciler listeleneceğinden blalinankitaplar çağırılır
        private void KitapBilgisi_Load(object sender, EventArgs e) //form
yüklendiğinde
        {
            DataTable dt = blkitaplar.KitapListele("","",kid); //kitapid ye göre
listeleme
            label1.Text = dt.Rows[0][0].ToString(); //indislerin yerine göre
labellere atama
```

```
            label2.Text = dt.Rows[0][1].ToString();
            label3.Text = dt.Rows[0][2].ToString();
            dataGridView1.DataSource =
blalinankitaplar.KitapAlanlar(kid);//dataGridView1 de kitapid ye göre alan
kişilerin listelenmesi
        }
```

Kayıt Ol buttonuna tıklandığında OgrenciKayit formu açılacaktır.

```
        private void button2_Click(object sender, EventArgs e)//Kayıt Ol
        {
            OgrenciKayit kyt = new OgrenciKayit();
            kyt.ShowDialog();
        }
```



Şekil 7 Kayıt Ol

OgrenciKayit Formunda Data Layer da bulunan Uyeler, Business Layerda bulunan BLUyeler tanımlanmalıdır. Aynı zamanda kütüphaneye de

```
using KutuphaneOtomasyon.Business_Layer;
using KutuphaneOtomasyon.Data_Layer;
```

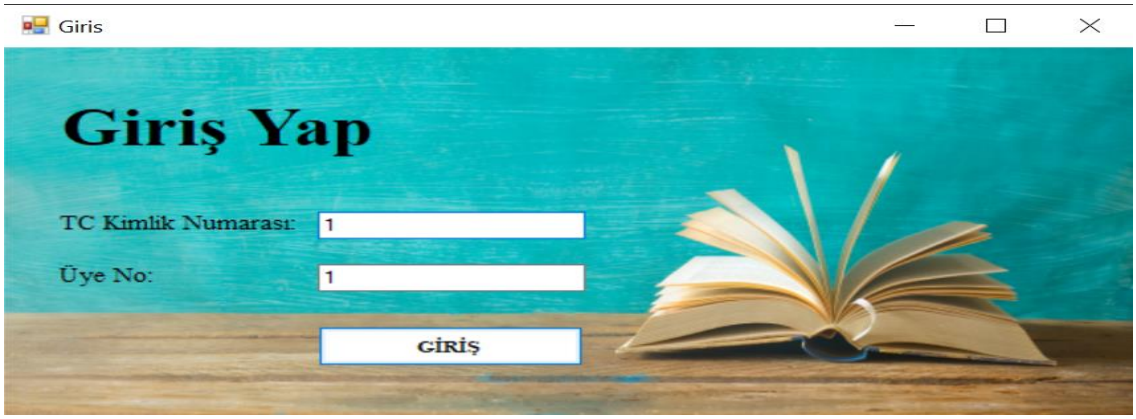
eklenmelidir.

```
        BLUyeler bluye= new BLUyeler(); //üye ekleme işlemi yapılacağından
bluyeler classı çağırılır
        Uyeler uyeler;
        private void button1_Click(object sender, EventArgs e)
        {
            uyeler = new Uyeler();
            //textboxa girilecek değerler veritabanına kayıt edilecek yere
eşitlenir
            uyeler.UyeTC= textBox1.Text;
            uyeler.UyeAd = textBox2.Text;
            uyeler.UyeSoyad = textBox3.Text;
            uyeler.UyeNo = textBox4.Text;
            uyeler.UyeTel = textBox5.Text;
            uyeler.UyeAdres = textBox6.Text;
            uyeler.UyePozisyon = "0";
            int kontrol = bluye.UyeIslem(uyeler,0,0);//üyeişlemde bulunan ekle
fonksiyonuna gönderilir

            if (kontrol > 0)//kayıt kontrol edilir
            {
                MessageBox.Show("Kayıt Başarıyla Oluşturulmuştur.");
            }
            else
            {
                MessageBox.Show("Kayıt Oluşturulamadı. Tekrar Deneyiniz.");
            }
        }
```

Giriş buttonuna tıklandığında Giriş formu açılacaktır.

```
        private void button4_Click(object sender, EventArgs e) //Giriş
        {
```

```
            Giris grs = new Giris();
            grs.ShowDialog();
        }
```



Şekil 8 Giriş

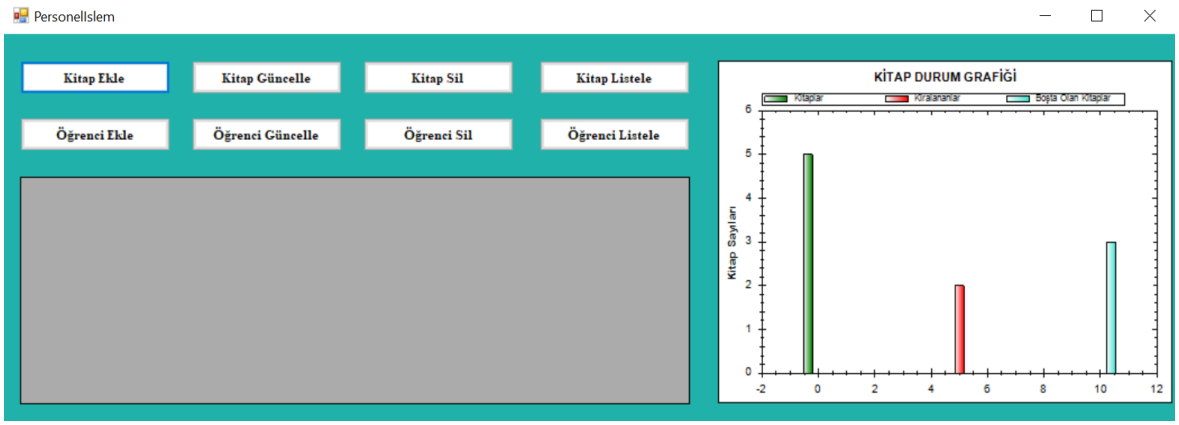
Giriş yap formunda hem personel üyeler (1) hem de öğrenci üyeler (2,3,4..) giriş yapabilmeleri için TC Kimlik numaraları ve Üye Numaralarını girmelidir.

```
        BLUyeler bluyeler = new BLUyeler();
        private void button1_Click(object sender, EventArgs e)
        {
            OgrenciIslem oislem = new OgrenciIslem();//öğrenci girişi yapılırsa
            PersonelIslem pislem = new PersonelIslem();//personel girişi
yapılırsa
            DataTable uyelistesi = bluyeler.UyeListele(-
1,textBox2.Text,textBox1.Text);

                if (uyelistesi.Rows[0][3].ToString() == textBox1.Text &&
uyelistesi.Rows[0][0].ToString() == textBox2.Text)//textboxa girilen değerlerin
listede kontrolü
                {
                    if (Convert.ToInt32(uyelistesi.Rows[0][6].ToString()) ==
0)//öğrenci
```

```
                    {
                        oislem.ogrno = textBox1.Text;
                        oislem.ogrtc = textBox2.Text;
                        oislem.ShowDialog();
                    }
                    else if (Convert.ToInt32(uyelistesi.Rows[0][6].ToString())
== 1)//personel
                    {
                        pislem.ShowDialog();
                    }
                }
            }
```

2.1.3.2. Personel

Personel formunda Araç Kutusundan çekilen buttonlar ve dataGridView ile gerekli bir tasarım yapıldıktan sonra kitap ve üyelere dair ekleme, güncelleme, silme ve listeleme gibi tüm işlemler yaptırılabilmektedir. Güncelleme ve Silme işlemlerinin yapılabilmesi için önce Listeleme yapılması gerekmektedir. Aynı zamanda ZedGraph kullanılarak kitaplar, kiralananlar ve boşta olan kitaplar grafiksel olarak gösterilmiştir.

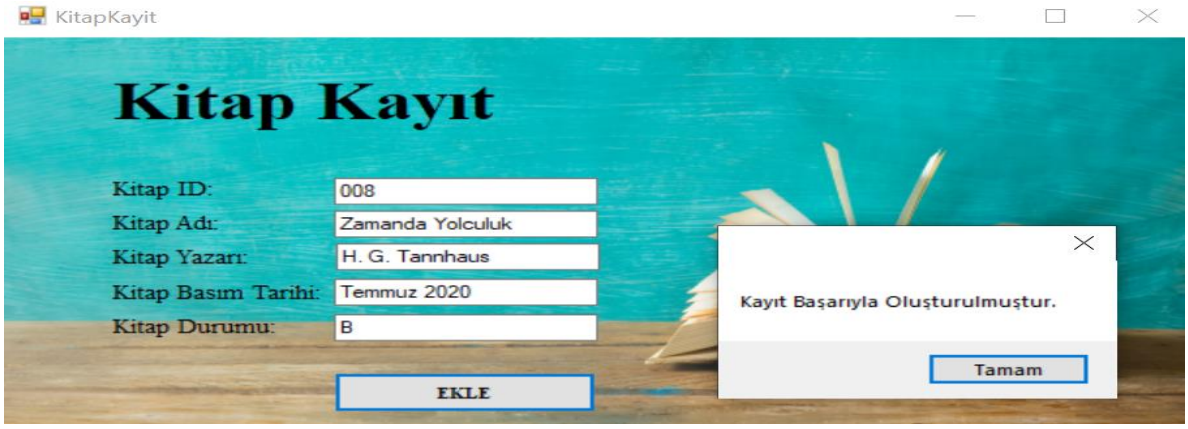


Şekil 9 Personel İşlem

2.1.3.2.1. Kitap İşlemleri

Kitap Ekle butonuna tıklandığında personel Kitap Kayıt formuna yönlendirir.

```
private void button1_Click(object sender, EventArgs e)
{
    //kitap kayıt
    KitapKayit kkayit = new KitapKayit();
    kkayit.ShowDialog();
}
```

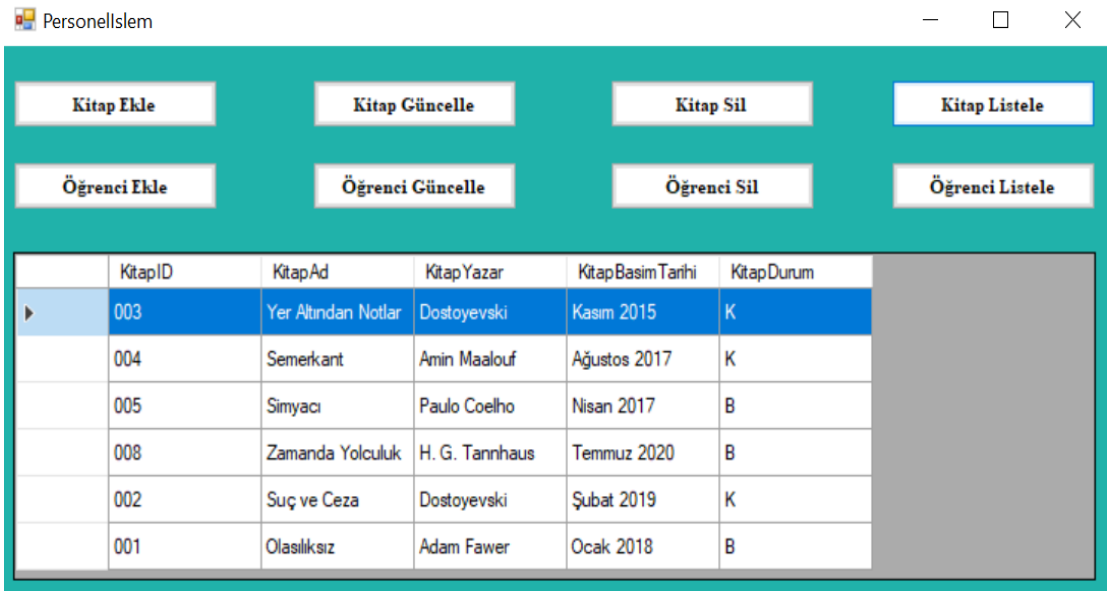


Şekil 10 Personel Kitap Kayıt

```
        BLKitaplar blkitap = new BLKitaplar();
        Data_Layer.Kitaplar kitaplar;

        private void button1_Click(object sender, EventArgs e)
        {
            kitaplar = new Data_Layer.Kitaplar(textBox1.Text, textBox2.Text,
textBox3.Text, textBox4.Text, textBox5.Text);//textboxa girilen değerler
kitaplara yönlendirilir
            int kontrol = blkitap.KitapIslem(kitaplar, 0);//kitapişlemde
kitapları parametre alır ve ekle fonksiyonuna yönlendirir

            if (kontrol > 0)//kontrol
            {
                MessageBox.Show("Kayıt Başarıyla Oluşturulmuştur.");
            }
            else
            {
                MessageBox.Show("Kayıt Oluşturulamadı. Tekrar Deneyiniz.");
            }
        }
```

Kitap Listele butonuna tıklandığında dataGridView1 de kütüphanede bulunan tüm kitaplar listelenir.

```
private void button8_Click(object sender, EventArgs e)
        {
            //kitap listeleme
            dataGridView1.DataSource = blkitap.KitapListele("", "", "");
        }
```

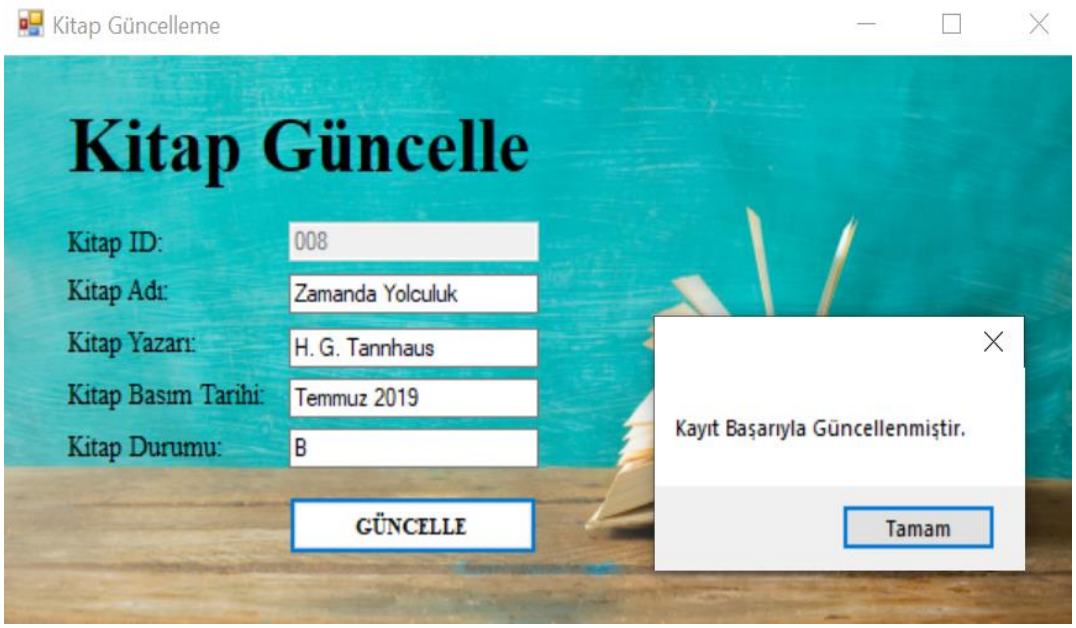


Şekil 11 Personel Kitap Listele

Kitap güncellemek için önce kitap listelenir ve kitap güncelle buttonuna tıklandığında; Kitap Güncelleme formu açılmaktadır.

```
        private void button2_Click(object sender, EventArgs e)
        {
            //kitap guncelleme
            //kitap güncelle formuna dataGridView1 den seçilen satırın bilgileri
aktarılır
            KitapGuncelleme kguncelleme = new KitapGuncelleme();
```

```
            kguncelleme.kid =
dataGridView1.SelectedRows[0].Cells["KitapID"].Value.ToString();
            kguncelleme.kad =
dataGridView1.SelectedRows[0].Cells["KitapAd"].Value.ToString();
            kguncelleme.kyazar =
dataGridView1.SelectedRows[0].Cells["KitapYazar"].Value.ToString();
            kguncelleme.kbasim =
dataGridView1.SelectedRows[0].Cells["KitapBasimTarihi"].Value.ToString();
            kguncelleme.kdurum =
dataGridView1.SelectedRows[0].Cells["KitapDurum"].Value.ToString();
            kguncelleme.ShowDialog();//form açılır
        }
```



Şekil 12 Personel Kitap Güncelle

```
        public string kid, kad, kyazar, kbasim, kdurum;

        private void button1_Click(object sender, EventArgs e)//Güncelle
butonuna tıklanıldığında
```

```
        {
            Data_Layer.Kitaplar kitaplar = new
Data_Layer.Kitaplar(textBox1.Text, textBox2.Text, textBox3.Text, textBox4.Text,
textBox5.Text);//textboxlara girilen değerler datalayerdaki kitaplara
yönlendirilir
            int kontrol = blkitap.KitapIslem(kitaplar,2);//kitap işlemlerine
parametre olarak kitaplar ve güncelle işlemi gönderilir
            if (kontrol > 0)//işlem kontrolü
            {
                MessageBox.Show("Kayıt Başarıyla Güncellenmiştir.");
            }
            else
            {
                MessageBox.Show("Kayıt Güncellenemedi. Tekrar Deneyiniz.");
            }
        }

        private void KitapGuncelleme_Load(object sender, EventArgs e)
//yüklendiğinde bilgiler otomatik yüklenir
        {
            textBox1.Text = kid;
            textBox2.Text = kad;
            textBox3.Text = kyazar;
            textBox4.Text = kbasim;
            textBox5.Text = kdurum;
        }
    }
```

Kitap silme işlemi yapılabilmesi için öncelikle kitaplar listelenir ve listeden silinmek istenen satır seçilir. Satır seçildikten sonra Kitap Sil buttonuna tıklanır.

```
        private void button3_Click(object sender, EventArgs e)
        {
            // kitap silme
```

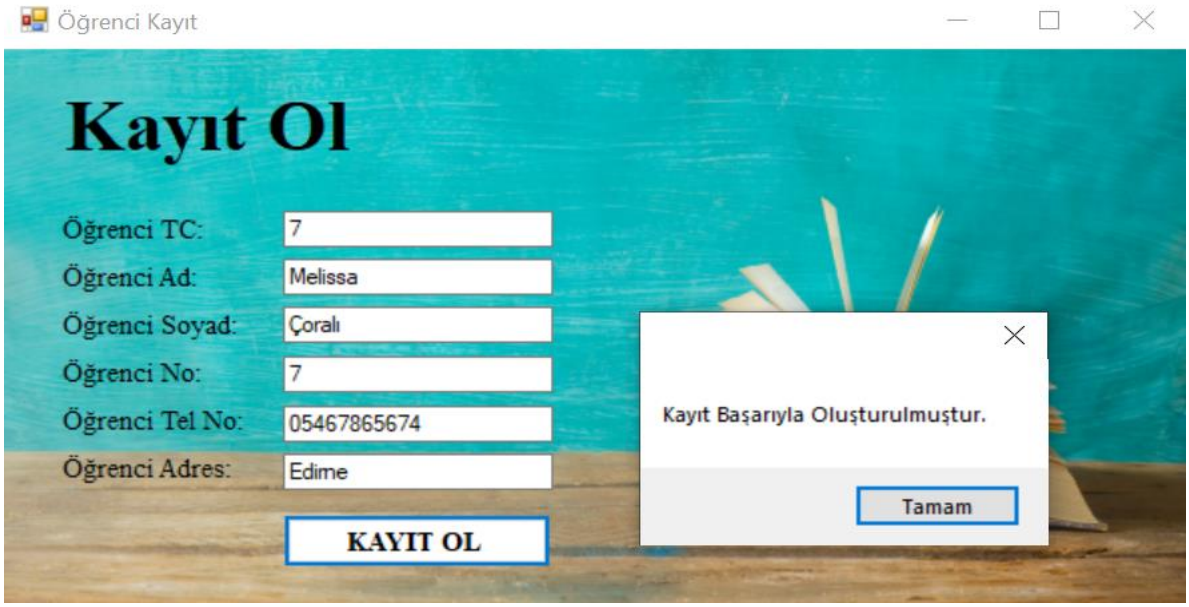
```
            Data_Layer.Kitaplar kitaplar = new
Data_Layer.Kitaplar(dataGridView1.SelectedRows[0].Cells["KitapID"].Value.ToStrin
g(), "", "", "", "");//seçilen satırdaki kitapid gönderilir
            int kontrol = blkitap.KitapIslem(kitaplar, 3);//kitapişlemlerinden
silme işlemine yönlendirilir
            if (kontrol > 0)//işlem kontrolü
            {
                MessageBox.Show("Kayıt Başarıyla Silinmiştir.");
            }
            else
            {
                MessageBox.Show("Kayıt Silinememiştir. Tekrar Deneyiniz.");
            }
            dataGridView1.DataSource = blkitap.KitapListele("", "", "");//silme
işlemi yapıldıktan sonra güncel listeleme
        }
```

Şekil 13 Personel Kitap Sil

2.1.3.2.2. Öğrenci İşlemleri

Öğrenci ekle butonuna tıklanıldığında anasayfada ki Öğrenci Kayıt formuna yönlendirilir. Daha önce bu kısımdan bahsedildiği için kodlar yazılmayacaktır.

```
private void button6_Click(object sender, EventArgs e)
{
    //öğrenci kayıt
    OgrenciKayit ogrkayit = new OgrenciKayit();
    ogrkayit.ShowDialog();
}
```



Şekil 14 Personel Öğrenci Ekleme

Öğrenci Listele buttonuna tıklandığında dataGridView1 de öğrenciler listelenir.

```
private void button7_Click(object sender, EventArgs e)
{
    // öğrenci listeleme
    dataGridView1.DataSource = bluyeler.UyeListele(0, "", "");
}
```

Şekil 15 Personel Öğrenci Listeleme

Öğrenci Silme işlemi yapılabilmesi için öncelikle listeden silinecek olan öğrencin bulunduğu satır seçilir ve Öğrenci Sil buttonuna tıklanır.

```
        private void button4_Click(object sender, EventArgs e)
        {
            //öğrenci silme
            uyeler.UyeTC =
dataGridView1.SelectedRows[0].Cells["UyeTC"].Value.ToString();//seçilen satırdaki
üyetc üyelere atanır
            int kontrol = bluyeler.UyeIslem(uyeler, 3, 0);//üyeişlem de üyeler ve
silme parametresini alır
            if (kontrol > 0)//işlem kontrolü
            {
                MessageBox.Show("Kayıt Başarıyla Silinmiştir.");
            }
            else
            {
                MessageBox.Show("Kayıt Silinememiştir. Tekrar Deneyiniz.");
```

```
            }
            dataGridView1.DataSource = bluyeler.UyeListele(0, "", "");//silme
işlemi yapıldıktan sonra güncel listeleme
        }
```

Şekil 16 Personel Öğrenci Silme

Öğrenci güncelleme işleminin yapılabilmesi için listeden güncellenecek olan öğrenci seçilir. Seçildikten sonra Öğrenci Güncelle butonuna tıklandığında Öğrenci Güncelleme formuna yönlendirilir.

```
        private void button5_Click(object sender, EventArgs e)
        {
            //öğrenci güncelleme
            OgrenciGuncelleme ogrguncelleme = new OgrenciGuncelleme();
            ogrguncelleme.tc =
dataGridView1.SelectedRows[0].Cells["UyeTC"].Value.ToString();
            ogrguncelleme.ad =
dataGridView1.SelectedRows[0].Cells["UyeAd"].Value.ToString();
```

```
            ogrguncelleme.soyad =
dataGridView1.SelectedRows[0].Cells["UyeSoyad"].Value.ToString();
            ogrguncelleme.no =
dataGridView1.SelectedRows[0].Cells["UyeNo"].Value.ToString();
            ogrguncelleme.tel =
dataGridView1.SelectedRows[0].Cells["UyeTel"].Value.ToString();
            ogrguncelleme.adres =
dataGridView1.SelectedRows[0].Cells["UyeAdres"].Value.ToString();
            ogrguncelleme.ShowDialog();//öğrenci güncelleme formu açılır
        }
```

Şekil 17 Personel Öğrenci Güncelleme

```
        Uyeler Uyeler = new Uyeler();
        BLUyeler bluyeler = new BLUyeler();
        private void button1_Click(object sender, EventArgs e)//güncelle butonu
        {
            //textboxa girilen değerler üyelere atanır
            Uyeler.UyeTC = textBox1.Text;
```

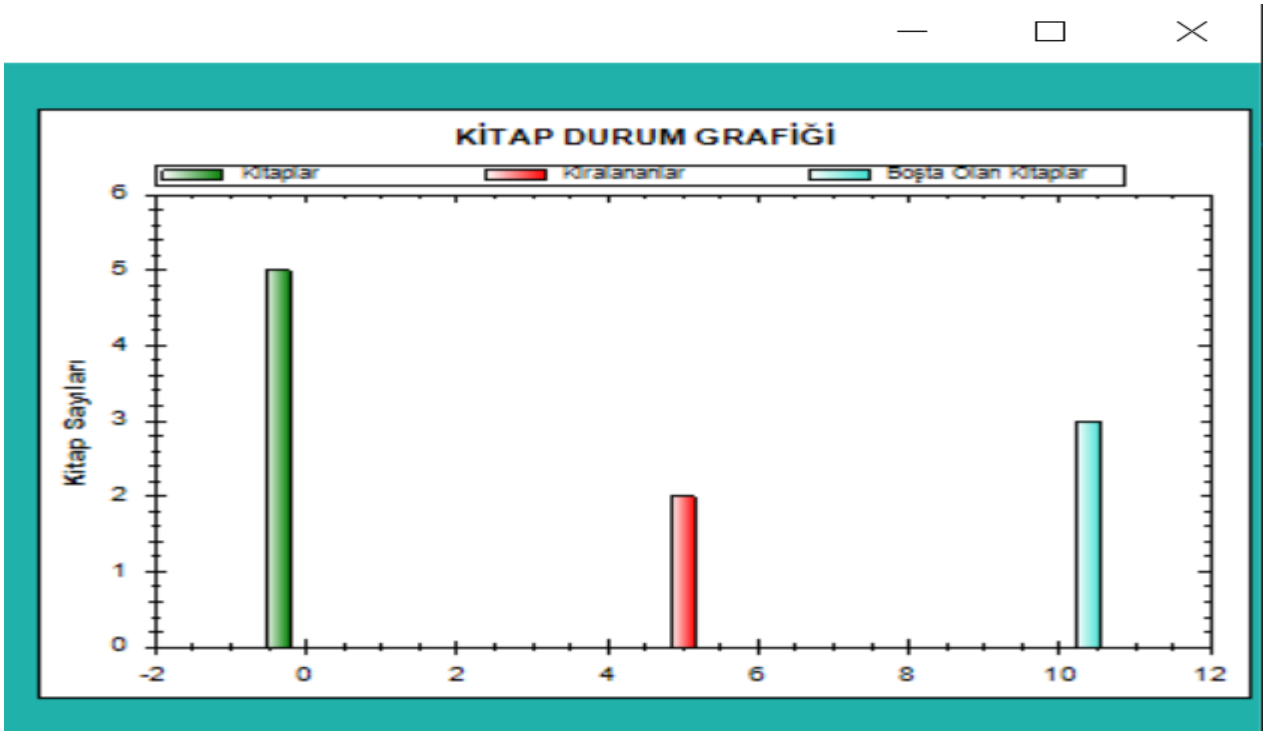
```
            Uyeler.UyeAd = textBox2.Text;
            Uyeler.UyeSoyad = textBox3.Text;
            Uyeler.UyeNo = textBox4.Text;
            Uyeler.UyeTel = textBox5.Text;
            Uyeler.UyeAdres = textBox6.Text;
            int kontrol = bluyeler.UyeIslem(Uyeler,2,0);//üyeişlem üyeleri ve
güncelleyi parametre alır
            if (kontrol > 0)//işlem kontrolü
            {
                MessageBox.Show("Kayıt Başarıyla Güncellenmiştir.");
            }
            else
            {
                MessageBox.Show("Kayıt Güncellenemedi. Tekrar Deneyiniz.");
            }
        }
        public string tc, ad, soyad, no, tel, adres;
        private void OgrenciGuncelleme_Load(object sender, EventArgs
e)//güncelle formu yüklendiğinde
        {
            //dataGridView1 den seçilen değerler textboxa atanır
            textBox1.Text = tc;
            textBox2.Text = ad;
            textBox3.Text = soyad;
            textBox4.Text = no;
            textBox5.Text = tel;
            textBox6.Text = adres;
        }
```

2.1.3.2.3. ZedGraph

Öncelikle http://sourceforge.net/projects/zedgraph/files/ buradan zedgraph dll ini bilgisayarımızın herhangi bir yerine projemize dahil etmemiz için kayıt etmemiz gerekiyor.

1. Add Referance diyerek projemize dahil ediyoruz
2. Bilgisayarda nereye kayıt ettiysek yolunu gösteriyoruz.
3. Araç kutusuna gerekli olan itemi ekliyoruz.
4. Son olarak Formumuza sürükleyip bıraktıktan sonra boş olarak grafik arayüzümüz karşımıza geliyor.

Personel İşlem formu yüklendiğinde Kitap Durum Grafiği de otomatik olarak yüklenir.



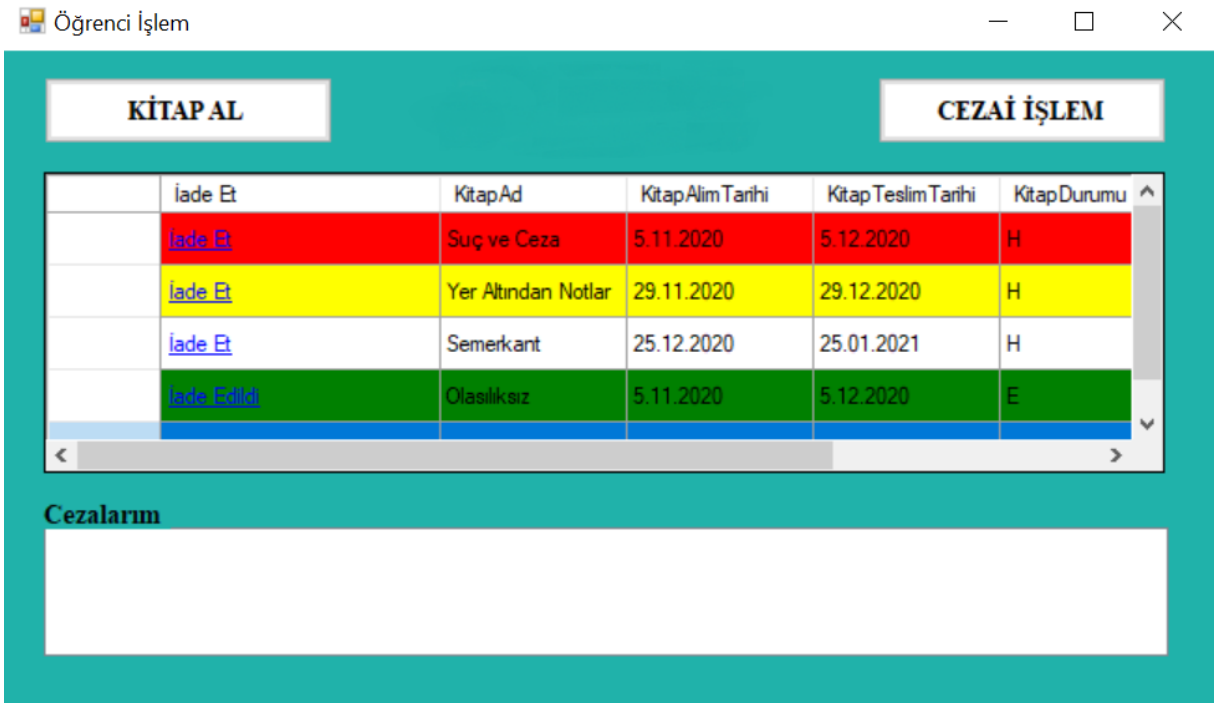
Şekil 18 ZedGraph

```
        GraphPane pane;
        BarItem bi1, bi2, bi3;
        private void PersonelIslem_Load(object sender, EventArgs e)
        {
            pane = zedGraphControl1.GraphPane;//beyaz grafik alanını temsil eder
            pane.Title.Text = "KİTAP DURUM GRAFİĞİ";//Grafik başlığı
            pane.XAxis.Title.Text = ""; //xaxis için başlık verilmiyor
            pane.YAxis.Title.Text = "Kitap Sayıları";//yaxis başlığı

            DataTable dt1 = blkitap.KitapSayisi();//kitap işlemlerinden kitap
sayısı fonksiyonu dt1 e atanır
            double[] kitaplarx = { 0 };//dt1 in x kısmındaki konumu
            double[] kitaplary = { Convert.ToDouble(dt1.Rows[0][0].ToString())
};
            DataTable dt2 = blkitap.KiralananlarSayisi();//kitap işlemlerinden
kiralanan kitap sayısı fonksiyonu dt2 e atanır
            double[] kiralanalarx = { 5 };//dt2 nin x kısmındaki konumu
            double[] kiralanalary = {
Convert.ToDouble(dt2.Rows[0][0].ToString()) };
            DataTable dt3 = blkitap.BosKitapSayisi();//kitap işlemlerinden boş
kitap sayısı fonksiyonu dt3 e atanır
            double[] verilmeyehazirx = { 10 };//dt3 nin x kısmındaki konumu
            double[] verilmeyehaziry = {
Convert.ToDouble(dt3.Rows[0][0].ToString()) };
            //line chart çizmek için nokta çiftleri oluşturulur
            PointPairList pl1 = new PointPairList(kitaplarx, kitaplary);
            PointPairList pl2 = new PointPairList(kiralanalarx, kiralanalary);
            PointPairList pl3 = new PointPairList(verilmeyehazirx,
verilmeyehaziry);
            //baritemlara isimleri, verileri ve renkleri atanır
             bi1 = pane.AddBar("Kitaplar", pl1, Color.Green);
             bi2 = pane.AddBar("Kiralananlar", pl2, Color.Red);
             bi3 = pane.AddBar("Boşta Olan Kitaplar", pl3, Color.Turquoise);
```

```
            TextObj barLabel = new TextObj(bi1.Points[0].Y.ToString(),
bi1.Points[0].X, bi1.Points[0].Y + 5);
            barLabel.FontSpec.Border.IsVisible = false;
            //bir degişiklik yaptımızda, bu değişikliği zedgraph a haber vermek
için çağırmamız gereken fonksiyonlar
            zedGraphControl1.AxisChange();
            zedGraphControl1.Invalidate();
            //Grafiği baştan çizdirmek için
            zedGraphControl1.Refresh();
        }
```

2.1.3.3. Ogrenciler

Öğrenci girişi yapıldığında ekrana gelecek olan form aşağıdaki resimdeki gibidir.



Şekil 19 Öğrenci İşlem

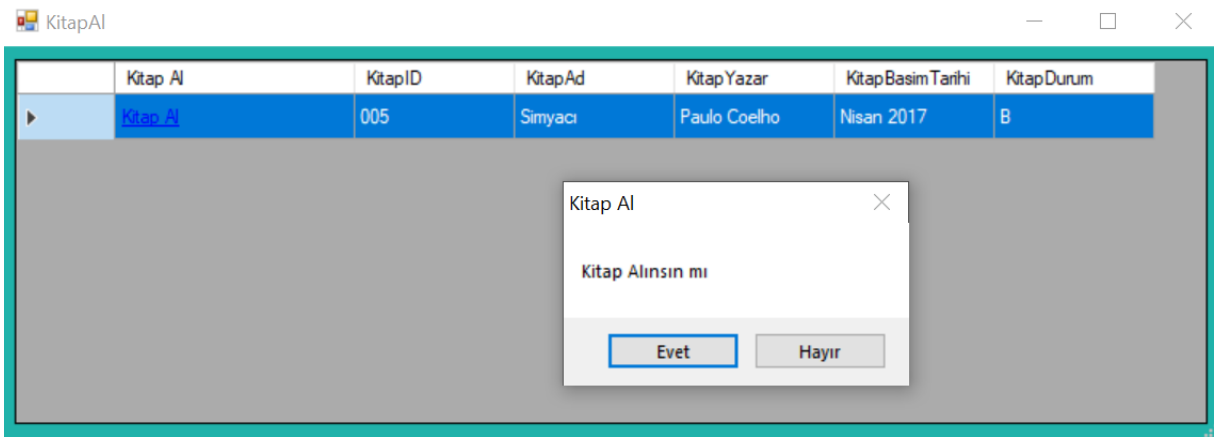
Form yüklendiğinde yazılacak kodlar;

```
        public string ogrtc, ogrno;
        BLKitaplar bLKitaplar = new BLKitaplar();
        BLAlinanKitaplar blalinankitaplar = new BLAlinanKitaplar();

        private void OgrenciIslem_Load(object sender, EventArgs e)
        {
            doldur();//fonksiyon çağırılır
        }
        void doldur()
        {
            dataGridView1.DataSource = bLKitaplar.KitapListele(ogrno,
ogrtc,"");//öğrenci no ve tc ye göre listeleme
            for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
            {
                Application.DoEvents();
                DataGridViewCellStyle renk = new DataGridViewCellStyle();
                string kteslimdurumu =
dataGridView1.Rows[i].Cells["KitapDurumu"].Value.ToString();//kitap durumu dgv1 den
çekilir
                string kteslimtarihi =
dataGridView1.Rows[i].Cells["KitapTeslimTarihi"].Value.ToString();//kitap teslim
tarihi dgv1 den çekilir
                string bugun = DateTime.Now.ToShortDateString();//bugünün tarihi
                double fark = (Convert.ToDateTime(bugun) -
Convert.ToDateTime(kteslimtarihi)).TotalDays;//bugünden teslim edilmesi gereken
tarih çıkartılır bu da farka eşitlenir
                if (fark > 0 && kteslimdurumu == "1") //fark 0 dan büyükse ve kitap
teslim edilmemişse
                {
                    renk.BackColor = Color.Red;//kırmızı uyarı
                }
```

```
                else if (fark == -2 && kteslimdurumu != "0") //teslime 2 gün varsa
ve teslim edilmemişse
                {
                    renk.BackColor = Color.Yellow;//sarı uyarı
                }
                else if (kteslimdurumu == "0") //teslim edilmişse
                {
                    renk.BackColor = Color.Green;//yeşil uyarı
                }
                dataGridView1.Rows[i].DefaultCellStyle = renk;
                string durum = dataGridView1.Rows[i].Cells[5].Value.ToString();//5.
indisi duruma esitler
                if (durum == "1")//eğer durum 1 ise
                {
                    dataGridView1.Rows[i].SetValues("İade Et");//iade et
                }
                else
                {
                    dataGridView1.Rows[i].SetValues("İade Edildi");//iade edildi
uyarısı
                }
            }
        }
```

Kitap Al buttonuna tıklandığında KitapAl formu açılır.

```
        private void button2_Click(object sender, EventArgs e)//kitap al
        {
            Kitap.KitapAl ktpal = new Kitap.KitapAl();
           // this.Close();
            ktpal.ogrno = ogrno;
            ktpal.ogrtc = ogrtc;
            ktpal.ShowDialog();
        }
```



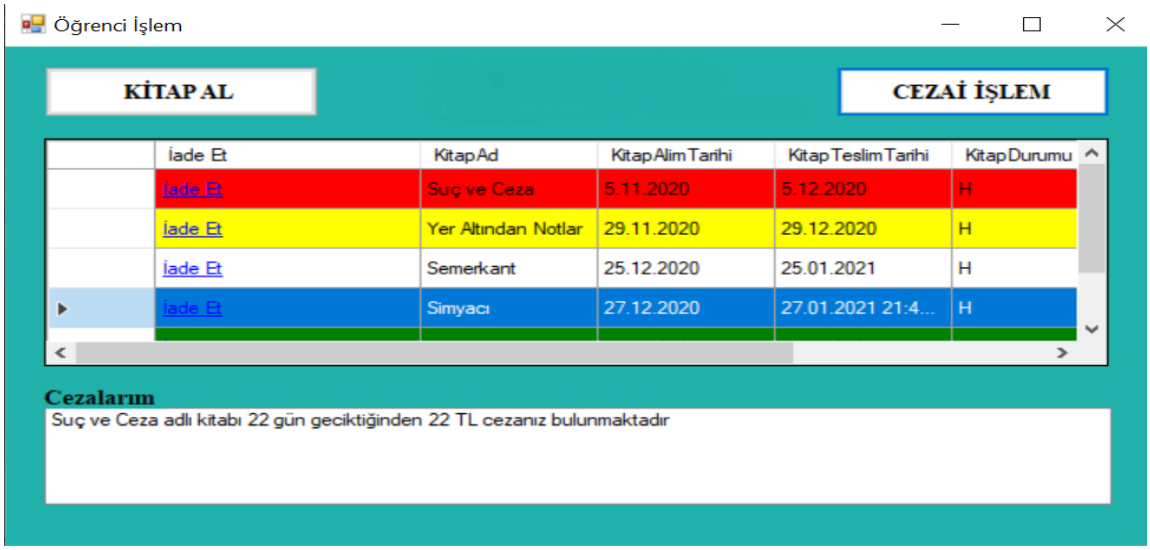
Şekil 20 Öğrenci Kitap Alma

```
        public string ogrtc, ogrno;
        BLKitaplar blkitap = new BLKitaplar();
        BLAlinanKitaplar blalinankitaplar = new BLAlinanKitaplar();


        private void KitapAl_Load(object sender, EventArgs e)
        {
            dataGridView1.DataSource = blkitap.KitapDurumListele();//kitap durumu 0
olan kitaplar listelenir
            for (int i = 0; i < dataGridView1.Rows.Count; i++)
            {
                dataGridView1.Rows[i].SetValues("Kitap Al");
            }
        }

        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
        {
            string kitapid =
dataGridView1.SelectedRows[0].Cells[1].Value.ToString();//listenin 1. indisi kitapid
ye eşitlenir
            DialogResult dialog = new DialogResult();
```

```
            dialog = MessageBox.Show("Kitap Alınsın mı", "Kitap Al",
MessageBoxButtons.YesNo);//işlem yapılacağı zaman sorma
            if (dialog == DialogResult.Yes)//evetse
            {
                blkitap.KitapDurumGuncelle(kitapid);//kitapid parametre olarak
kitapdurumgüncelleye gönderilir
            }
            else
            {
                this.Close();
            }
            string alinan = DateTime.Now.ToShortDateString();//bugünün tarihini
alinana eşitleriz
            string teslim = DateTime.Now.AddMonths(1).ToString();//teslim tarihi 1 ay
olarak ayarlanır
            int kontrol = blalinankitaplar.KitapAl(kitapid, ogrtc, alinan, teslim,
"1"); //değişkenler parametre olarak kitapal fonk gönderilir
            if (kontrol>0)//işlem kontrolü
            {
                MessageBox.Show("Kitap Alındı");
            }
            else
            {
                MessageBox.Show("Tekrar deneyiniz");
            }
            dataGridView1.DataSource = blkitap.KitapDurumListele();//liste duruma göre
yeniden listelenir
            for (int i = 0; i < dataGridView1.Rows.Count; i++)
            {
                dataGridView1.Rows[i].SetValues("Kitap Al");
            }
        }
```

Cezai işlem buttonuna tıklandığında listbox1 de ceza işlemleri görüntülenir.

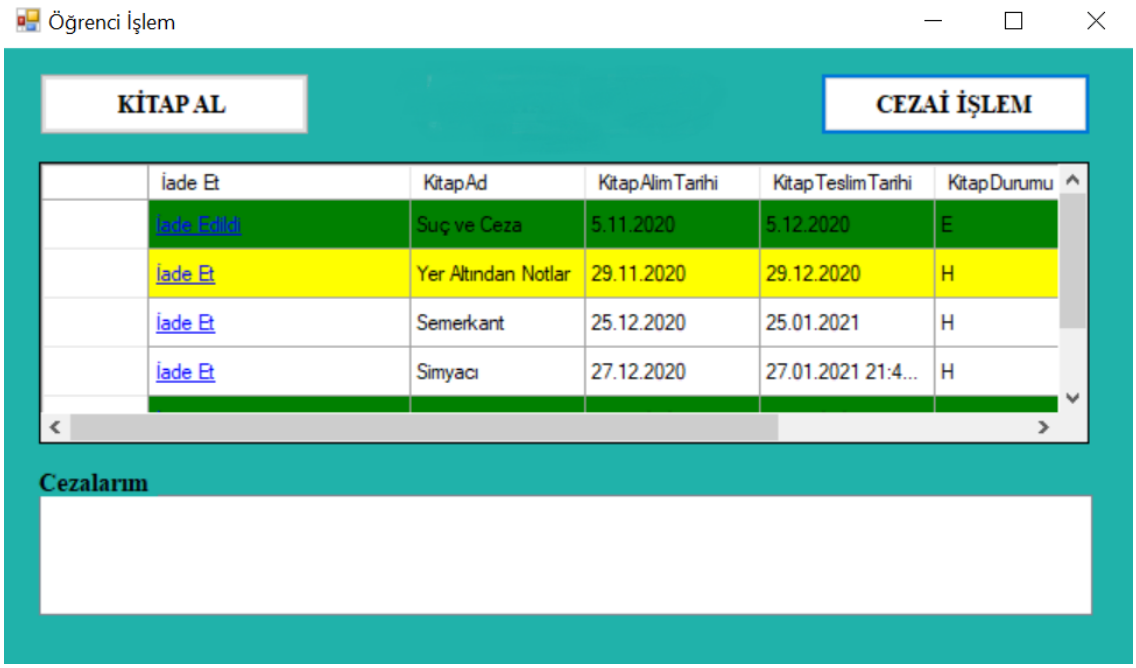
```
        private void button1_Click(object sender, EventArgs e)//ceza işlem
        {
            listBox1.Items.Clear();//listbox temizleme
            DataTable dt = bLKitaplar.KitapListele(ogrno, ogrtc,"");//öğrenci no ve tc
ye göre kitap listeletilir
            for (int i = 0; i < dt.Rows.Count; i++)
            {
                double tutar = 0;
                string kteslimtarihi =
dt.Rows[i]["KitapTeslimTarihi"].ToString();//listeden kitapteslimtarihi
kteslimtarihine eşitlenir
                string kdurum = dt.Rows[i]["KitapDurumu"].ToString();//listeden
kitapdurumu kduruma eşitlenir
                string bugun = DateTime.Now.ToShortDateString();//bugünün tarihi
alınır ve bugune eşitlenir
                double fark = (Convert.ToDateTime(bugun) -
Convert.ToDateTime(kteslimtarihi)).TotalDays;//bugünden teslim edilmesi gereken tarih
çıkartılır bu da farka eşitlenir
                if (fark > 0 && kdurum == "1") //fark 0 dan büyük ve teslim
edilmemişse
                {
                    tutar = fark * 1; //fark*1 tutarı vericektir
                    listBox1.Items.Add(dt.Rows[i]["KitapAd"].ToString() + " adlı
kitabı " + fark.ToString() + " gün geciktiğinden " + tutar.ToString() + " TL cezanız
bulunmaktadır");
                }
            }
        }
```



Şekil 21 Personel Cezai İşlem

İade işlemi dataGridView1 içinde gerçekleşir. dataGridView1 e İade Et adında bir kolon eklenir. Bu kolunun içinde bulunan linkle iade işlemi gerçekleştirilecektir.

```
        private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)//kitap iade
        {
            string islemid =
dataGridView1.SelectedRows[0].Cells[6].Value.ToString();//listenin 6. indisi
islemid ye eşitlenir
            string kid =
dataGridView1.SelectedRows[0].Cells[1].Value.ToString();//listenin 1. indisi kid e
eşitlenir
            int kontrol =
blalinankitaplar.KitapIade(Convert.ToInt32(islemid));//kitapiade fonk çağırılır ve
islemid parametre olarak gönderilir
            bLKitaplar.KitapDurumGuncelleIade(kid);//kitabın kitapdurumu
güncellenir
            doldur();
        }
```



Şekil 22 Öğrenci Kitap İade

3. SONUÇ

3.1. SET UP

Uygulamanın diğer bilgisayarlara kurulumu için set up dosyası olacaktır ve kişi bu uygulamayı istediği bilgisayara kurup o şekilde kullanabilecektir. Install Shield programını internetten indirip Visual studio programı üzerinden kullanıp yapılan programını install dosyasını oluşturulacaktır.

1. Öncelikle sağ tarafta yer alan panelde (solution exproler) projenin üzerine sağ tıklayıp açılan menüde build seçeneğine tıklanır.
2. Proje build edildikten sonra ise alt kısımda yer alan panelde (output) sonuçlar kontrol edilir. Eğer visual studio projeyi build ederken hata ile karşılaşırsa, öncelikle o hataları düzeltmemiz gerekir. İlgili hataları düzelttikten sonra proje tekrar build edilmeli ve aşağıdaki ekran önümüze gelmelidir.
3. Yukarıdaki iki adımı uyguladıktan sonra, tekrar sağ tarafta yer alan panelde (solution exproler) sağ tıklayıp açılan menüde add seçeneğine tıklanır. Sonrasında yan tarafta açılan menü içerisinde new project seçeneğine tıklanır.
4. Gelen ekranda sol taraftaki panelde sırası ile Other Project Types + Setu and Deployment adımları takip edilir. Ardından ok butonuna basılarak devam edilir.
5. Sonrasında gelen ekranda sırası ile altı adım takip edilir. Bu adımlar; uygulama bilgisi, kurulum gereksinimleri, uygulama dosyaları, uygulama kısayolu, kayıt defteri ayarları ve son aşamada ise genel kontroller yapılır.
6. Application Information sekmesinde ekran karşımıza gelir.
7. Installation Requirements sekmesinde ise yazdığımız uygulamanın hangi işletim sistemlerinde çalışabileceğini ve uygulamamızın listelenen yazılımlardan birisine/birkaçına ihtiyaç duyup duymadığını soruyor.

8. Application Files sekmesinde ise add files ya da add folders butonlarından birisine basılır, proje klasörü içerisinde sırasıyla bin + debug adımları takip edilir ve debug içerisindeki tüm dosyalar seçilir.
9. Application Shortcuts sekmesinde ise uygulama kısayolunun başlat menüsünde yer alması, yazılım kurulduktan sonra kısayolunun masaüstüne oluşturulması ve uygulama kısayolunun simgesi belirlenebilir.
10. Application Registery sekmesinde eğer Windows kayıt defterine kayıt atacaksak (lisanslama vb.) burada gerekli işlemler yaptırılabilir.
11. Installation Interview sekmesinde son düzenlemeler yapılır.

KAYNAKÇA

https://sezeromer.com/c-katmanli-mimari/

https://denizozkan92.wordpress.com/2016/07/12/c-dilinde-zedgraph-kullanimi/

https://www.furkanpezek.com.tr/2019/09/c-setup-olusturma/