*Şekiller Tablosu*

[*Şekil 1.Katmanlı Mimari Yapısı 4*](#_Toc61444743)

[*Şekil 2.Access Tabloları 6*](#_Toc61444744)

[*Şekil 3 Access Kitap Bilgileri 6*](#_Toc61444745)

[*Şekil 4.Access Öğrenci Bilgileri 7*](#_Toc61444746)

[*Şekil 5. Kitap İade ve Emanet Bilgileri 7*](#_Toc61444747)

[*Şekil 6.Bağlantı Sınıfı 8*](#_Toc61444748)

[*Şekil 7. DAL Sınıfı sorgu alanı 9*](#_Toc61444749)

[*Şekil 8.DAL Sınıfı Ekle Sil Güncelle Fonksiyonları 10*](#_Toc61444750)

[*Şekil 9.DAL Sınıfı Veri Çekme Fonksiyonları 10*](#_Toc61444751)

[*Şekil 10.Proje Sınıf Ekleme 11*](#_Toc61444752)

[*Şekil 11.İş Katmanı- İade Alım İşlemleri 12*](#_Toc61444753)

[*Şekil 12.İş Katmanı- İade Alım İşlemleri Kayıt Listeleme 12*](#_Toc61444754)

[*Şekil 13.İş Katmanı- İade Alım İşlemleri Emanet Kısmı 13*](#_Toc61444755)

[*Şekil 14.İş Katmanı Verileri Listeleme 13*](#_Toc61444756)

[*Şekil 15.İşlemler Sınıfı Listeleme Operatörü 14*](#_Toc61444757)

[*Şekil 16.İşlemler Sınıfı Alma İşlemi 14*](#_Toc61444758)

[*Şekil 17.Kitap İşlem Kütüphaneleri 15*](#_Toc61444759)

[*Şekil 18. Kitap Ekleme ve Güncelleme İşlemi 15*](#_Toc61444760)

[*Şekil 19.Kitap Silme İşlemi 16*](#_Toc61444761)

[*Şekil 20.Öğrenci İşlemleri Kütüphaneleri 16*](#_Toc61444762)

[*Şekil 21.Öğrenci Ekleme, Güncelleme, Silme İşlemleri 17*](#_Toc61444763)

[*Şekil 22.Anasayfa 18*](#_Toc61444764)

[*Şekil 23.Anasayfa Kütüphaneleri 19*](#_Toc61444765)

[*Şekil 24.Anasayfa İşlemleri 19*](#_Toc61444766)

[*Şekil 25.Öğrenci İşlemleri Form Görüntüsü 20*](#_Toc61444767)

[*Şekil 26.Öğrenci İşlemleri Kütüphaneleri 21*](#_Toc61444768)

[*Şekil 27. Öğrenci Listeleme İşlemleri 21*](#_Toc61444769)

[*Şekil 28.Öğrenci Arama İşlemi 22*](#_Toc61444770)

[*Şekil 29.Kitap İşlemleri 23*](#_Toc61444771)

[*Şekil 30.Kitap Ekleme İşlemi 24*](#_Toc61444772)

[*Şekil 31.Kitap Güncelleme, Listeleme ve Silme İşlemleri 24*](#_Toc61444773)

[*Şekil 32.Kitap Arama İşlemi 25*](#_Toc61444774)

[*Şekil 33.Kitap Bilgileri 25*](#_Toc61444775)

[*Şekil 34.Öğrenci Emanet İade Listesi 26*](#_Toc61444776)

[*Şekil 35.Öğrenci Emanet İade Listesi Fonksiyonları 27*](#_Toc61444777)

[*Şekil 36.Kitap Emanet İade Listesi 28*](#_Toc61444778)

[*Şekil 37.Kitap Emanet İade Listesi Kütüphaneleri 29*](#_Toc61444779)

[*Şekil 38.Anasayfaya Dönme Fonksiyonu 29*](#_Toc61444780)

[*Şekil 39.Kitap Emanet İade İşlemleri 30*](#_Toc61444781)

[*Şekil 40.Kitap Emanet Ve İade İşlemleri Kütüphaneleri 31*](#_Toc61444782)

[*Şekil 41.Kitap Emanet ve İade İşlemleri Fonksiyonları 32*](#_Toc61444783)

[*Şekil 42.Kitabın Teslim Tarihinin Belirlenmesi 33*](#_Toc61444784)

[*Şekil 43.Kitap İade Fonksiyonları 34*](#_Toc61444785)

[*Şekil 44.Listenin Renklendirilmesi 34*](#_Toc61444786)

[*Şekil 45.Listenin Sıralanması 35*](#_Toc61444787)

[*Şekil 46.Grafik Görünümü 35*](#_Toc61444788)

[*Şekil 47.Grafik Kütüphaneleri 36*](#_Toc61444789)

[*Şekil 48.Grafik Fonksiyonları 36*](#_Toc61444790)

[*Şekil 49.Entity Sınıfları 37*](#_Toc61444791)

[*Şekil 50.Öğrenciler Sınıfı 37*](#_Toc61444792)

[*Şekil 51.Kitaplar Sınıfı 38*](#_Toc61444793)

[*Şekil 52.Alma Verme Sınıfı 38*](#_Toc61444794)

*İçindekiler*

[*Katmanlı Mimari* 4](#_Toc61445522)

[*Peki çok katmanlı mimari kullanmak bize ne gibi avantajlar sağlar?* 4](#_Toc61445523)

[*Çok katmanlı mimarinin dezavantajları ise* 4](#_Toc61445524)

[*Proje oluştururken yapılan adımlar* 5](#_Toc61445525)

[*Veri Katmanı (Data Access Layer)* 5](#_Toc61445526)

[*Veritabanı Oluşturma* 6](#_Toc61445527)

[*Kitap Bilgilerimizin Yer Aldığı Tablo* 6](#_Toc61445528)

[*Öğrenci Bilgilerimizin Yer Aldığı Tablo* 7](#_Toc61445529)

[*Kitap İade ve Emanet Verme Bilgilerimizin Yer Aldığı Tablo* 7](#_Toc61445530)

[*İş Katmanı (Business Layer)* 11](#_Toc61445531)

[*Sunum Katmanı (Presentation Layer)* 17](#_Toc61445532)

[*Entities Katmanı* 37](#_Toc61445533)

[*Projenin Setup Dosyası* 38](#_Toc61445534)

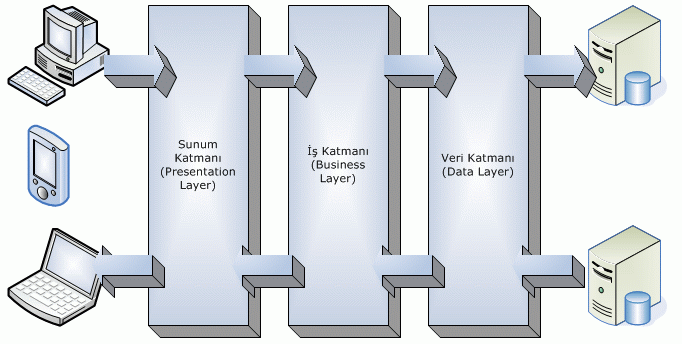
# *Katmanlı Mimari*

İlk başta katmanlı mimari nedir bunu öğrenelim.

*Katmanlı mimari* projelerimizi belirli bir standart ve düzene göre geliştirmemizi sağlayan, kodun okunabilirliğini arttıran, projelerimizin daha derli toplu olmasını sağlayan ve hata yönetimini daha kolay hale getiren bir yapıdır.

Bu yapı 3 ana katmandan oluştuğu gibi günümüzde Çok Katmanlı Mimari yapıda denebilir. Temelde 3 ana katman üzerinde durur. Bu katmanlar;

* Veri Katmanı (Data Access Layer)
* İş Katmanı (Business Layer)
* Sunum Katmanı (Presentation Layer)



Şekil 1.Katmanlı Mimari Yapısı

## *Peki çok katmanlı mimari kullanmak bize ne gibi avantajlar sağlar?*

* Kompleks ve gitgide karmaşıklaşan projelere daha kolay hakimiyet.
* Hata ayıklamanın kolaylaşması.
* Projede ekip olarak çalışmayı kolaylaştırır.
* Değişim sürecini hızlandırır.
* Esnek bir yapı sağlar.

## *Çok katmanlı mimarinin dezavantajları ise*

* Deployment süresinin uzunluğu

## *Proje oluştururken yapılan adımlar*

Projem aşağıda bulunan unsurları içerişinde barındırmaktadır.

Kütüphane otomasyonu temel unsurları:

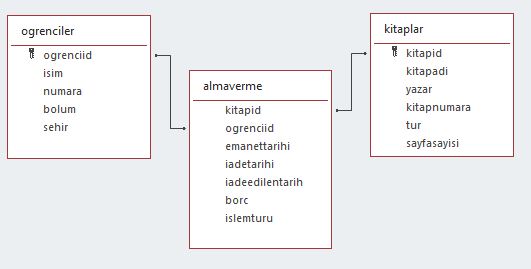
* Katmanlı Mimari tabanlı bir projedir.
* Öğrenci kayıt ekleme, güncelleme ve silme penceresi oluşturdum.
* Kitap ekleme, silme, güncelleme penceresi oluşturdum.
* Öğrencinin bir kitabı alma durumunu gösteren pencere oluşturdum. Bu pencerede mevcut öğrenci için kitap alımı ve iade işlemleri yapılabiliyor.
* Öğrencinin şimdiye kadar almış olduğu teslim etmiş ve teslim etmemiş olduğu kitaplar listesi sıralanıyor.
* Öğrencinin teslim süresini aştığı kitap için öğrenciye ait kitap listesinde kırmızı olarak uyarı veriyor.
* Teslim süresine 2 gün kalan kitaplar için uyarı amaçlı sarı renk ile bilgilendirme yapılıyor.
* Teslim edilmiş kitap listesinde her bir kitap yeşil renkte gözüküyor.
* Kitap alma ve iade işlemleri tarihsel olarak tablo bulunuyor.
* Her bir kitap için kitap işlemleri sayfasında ilgili kitap takibi yapılabiliyor. Yani kitapların listeleme butonuna tıklandığında kitap id’si, adı, yazarı, numarası, türü, sayfa sayısı bilgilerini tabloda görüntülenebiliyor.
* Kitap Emanet-İade İşlemleri sayfasında kitabın id’si ile o kitabı teslim almış ve teslim etmiş olan öğrencilerin id’sine göre liste yer alıyor. Hangi tarihlerde alınıp hangi tarihlerde teslim edildiği bilgileri de tabloda bulunuyor.
* Kütüphanede bulunan tüm kitapların, öğrenciye verilmiş olan ve kütüphanede verilmeye hazır olan kitapların sayısını grafik olarak gösteriliyor. Bu grafiğe yakınlaştırma, uzaklaştırma özellikleri bulunuyor.
* Uygulamanın diğer bilgisayarlara kurulumu için set up dosyası yer alıyor ve kişi bu uygulamayı istediği bilgisayar kurup o şekilde kullanabiliyor.

## *Veri Katmanı (Data Access Layer)*

 Bu katman adından da anlaşılacağı üzere veriler ve veritabanı ile uğraşacağımız bölümdür. Bu katmanda sadece veritabanı işlemleri yapılmaktadır. Bu katmanın görevi veriyi ekleme, silme, güncelleme ve veritabanından çekme işlemidir. Bu katmanda bu işlemlerden başka herhangi bir işlem yapılmamaktadır.

## *Veritabanı Oluşturma*

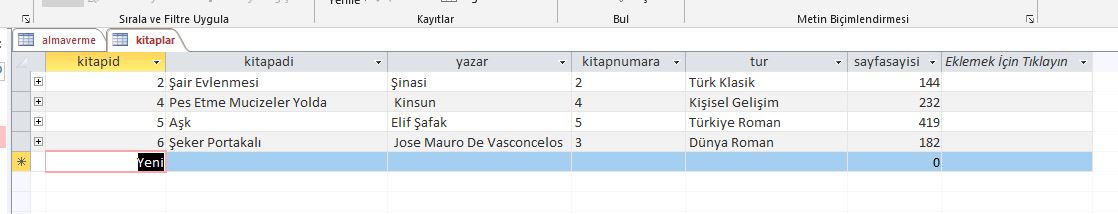
Şimdi projemize başlayalım. İlk başta Access’ten veri tabanını çekebilmemiz için tablolarımızı oluşturalım.



Şekil 2.Access Tabloları

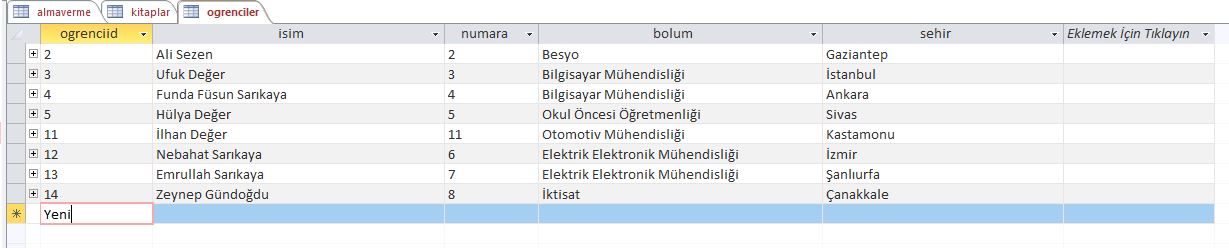
Girilecek değerlerimizi Access’te tablomuza yazabiliriz. Projemizi oluşturduktan sonrada ekleme, listeleme, silme ve güncelleme fonksiyonlarımızla girdiğimiz verilerin tablomuzda değiştiğini görebiliriz.

## *Kitap Bilgilerimizin Yer Aldığı Tablo*



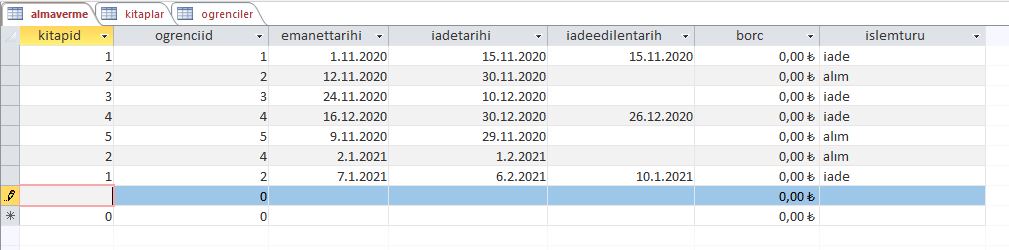
Şekil 3 Access Kitap Bilgileri

## *Öğrenci Bilgilerimizin Yer Aldığı Tablo*



Şekil 4.Access Öğrenci Bilgileri

## *Kitap İade ve Emanet Verme Bilgilerimizin Yer Aldığı Tablo*



Şekil 5. Kitap İade ve Emanet Bilgileri

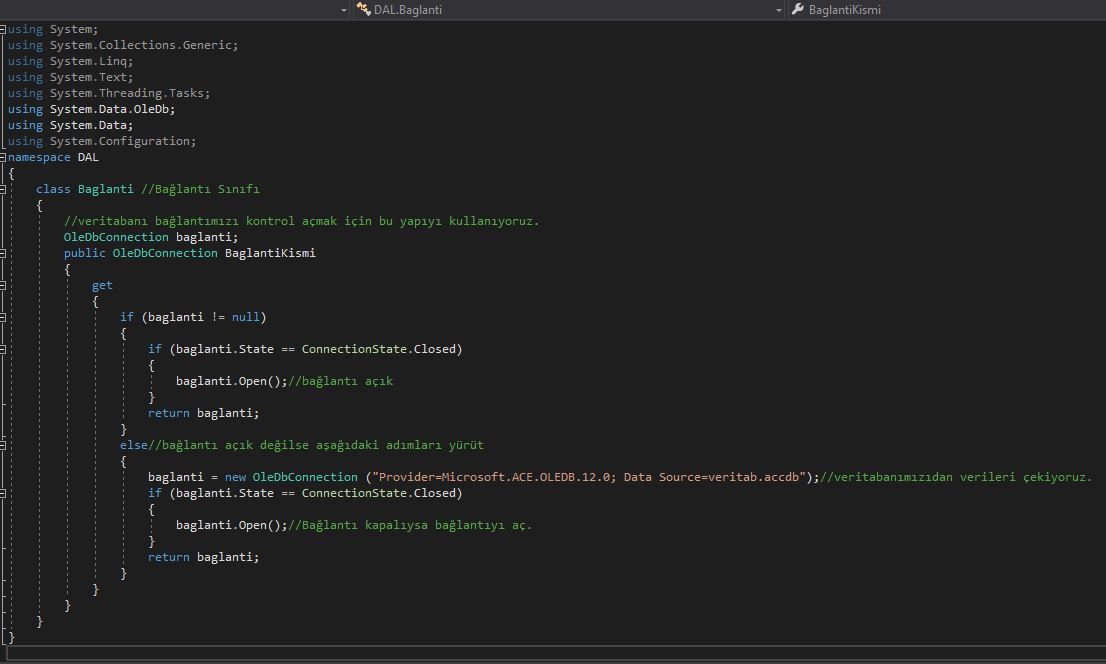
Access’te tablolarımızı oluşturduğumuza göre artık projemizin iskeletini oluşturabiliriz.

Visual Studio uygulamamızı açalım. File> New> Project> Class Library (.NET Framework) diyerek Data Access Layer Katmanımızı oluşturalım.

Sınıfımızın adını bağlantı olarak değiştirelim ve burada Access ile bağlantımızın doğru çalışıp çalışmadığını kontrol edelim.

Bu projemizin içine 1 adet daha class açalım. Bu class’ımızın adını da DAL yapalım ve burada veri tabanımıza ait sorguları gerçekleştirelim.

Bağlantı sınıfıma ait kodlar aşağıdaki görselde yer almaktadır.

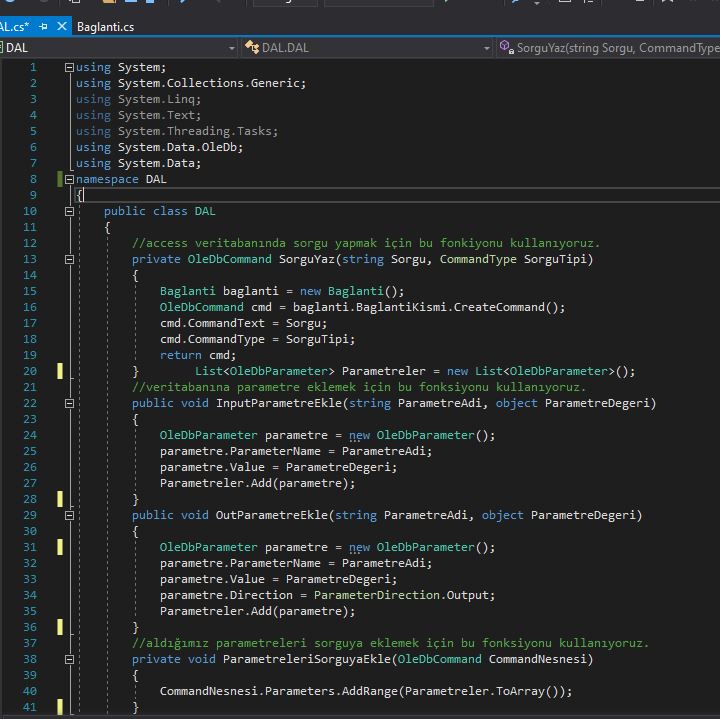


Şekil 6.Bağlantı Sınıfı

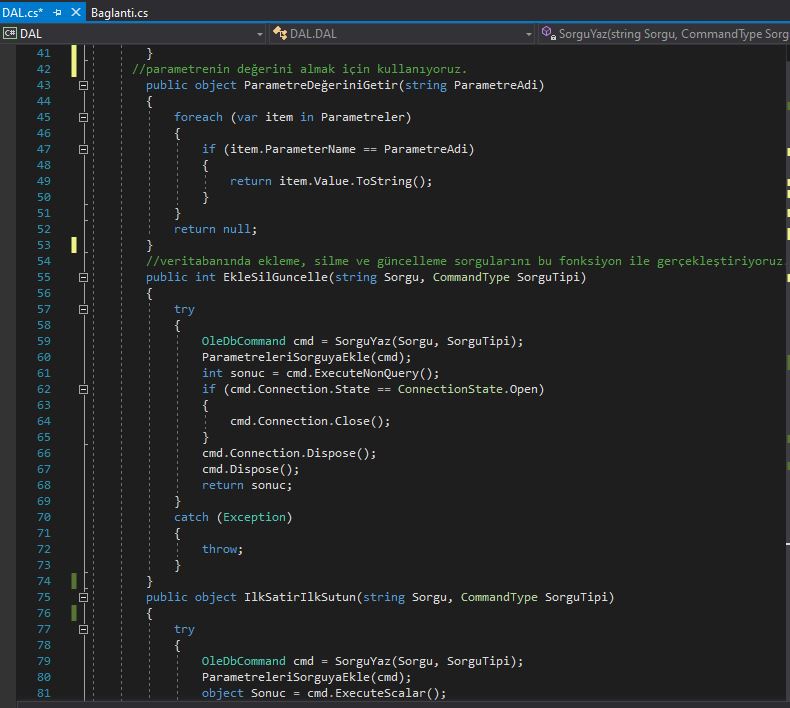
(“Provider=Microsoft.ACE.OLEDB.12.0; Data Source=veritab.accdb”);

Bu kod parçasında Access’te oluşturduğumuz veritabanı tablolarımızın yolunu yazıyoruz.

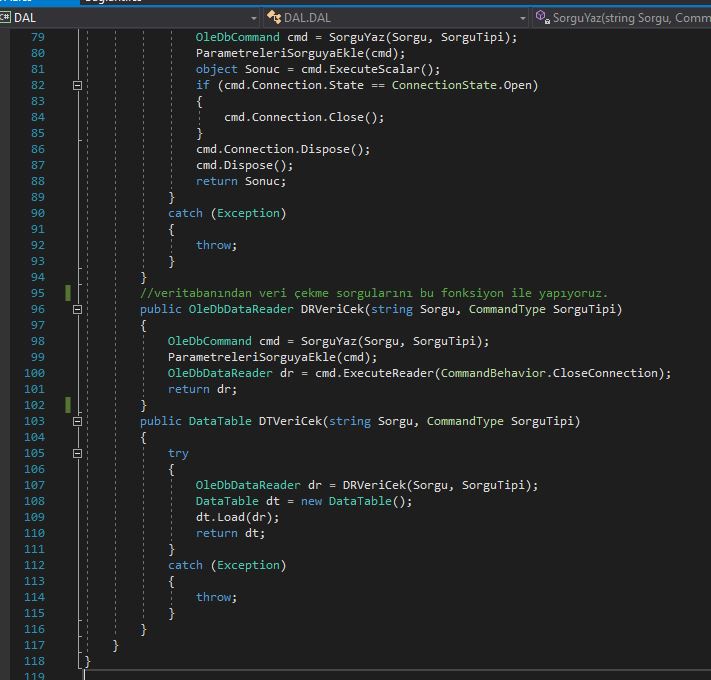
DAL sınıfıma ait kodlar aşağıdaki görsellerde yer almaktadır.



Şekil 7. DAL Sınıfı sorgu alanı



Şekil 8.DAL Sınıfı Ekle Sil Güncelle Fonksiyonları



Şekil 9.DAL Sınıfı Veri Çekme Fonksiyonları

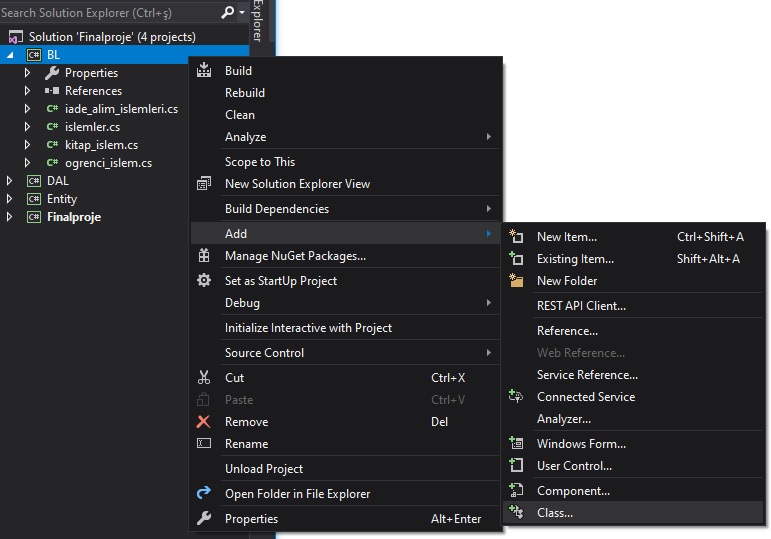
Artık yapacağımız işlemler için iş katmanımızı oluşturmalıyız.

## *İş Katmanı (Business Layer)*

Bu katmanda ise asıl olaylar başlıyor. İş yüklerimizi yazıyoruz. Öncelikle şunu söylemeliyim bu katman Data Access tarafından projeye çekilmiş olan verileri alarak işleyecek olan katmandır. Biz uygulamalarımızda Data Access katmanını direk olarak kullanmayız. Veritabanından verileri hangi sorguyla alacağımızı, verileri nasıl okuyacağımızı, verileri okuduktan sonra ne yapacağımızı bütün bu olayları bu katmanda yapıyoruz. Aslında veritabanıyla projemiz arasında bir köprü oluşturmayı sağlıyor.

Şimdi Solution yazan kısımda sağ tık Add>New Project> Class Library (.NET Framework) diyerek BL (Business Layer) Katmanımızı oluşturalım.

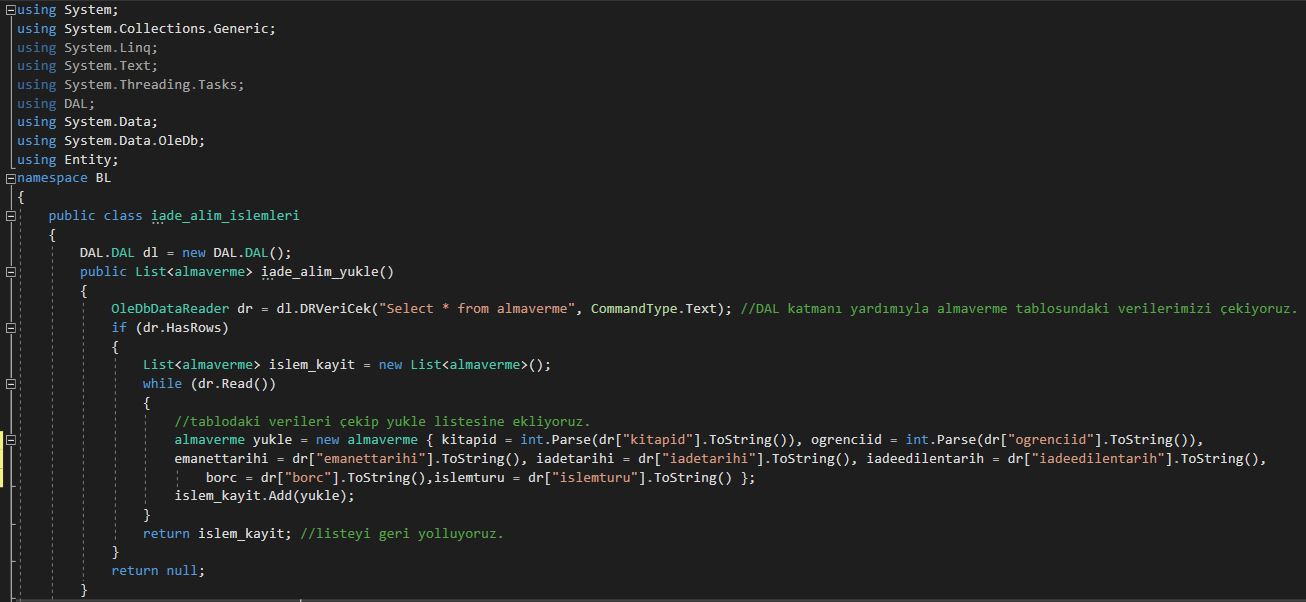
4 adet Sınıf oluşturalım. Sınıf oluştururken sağ tık yapıp sonra aşağıda gösterilen yolu takip edebilirsiniz.



Şekil 10.Proje Sınıf Ekleme

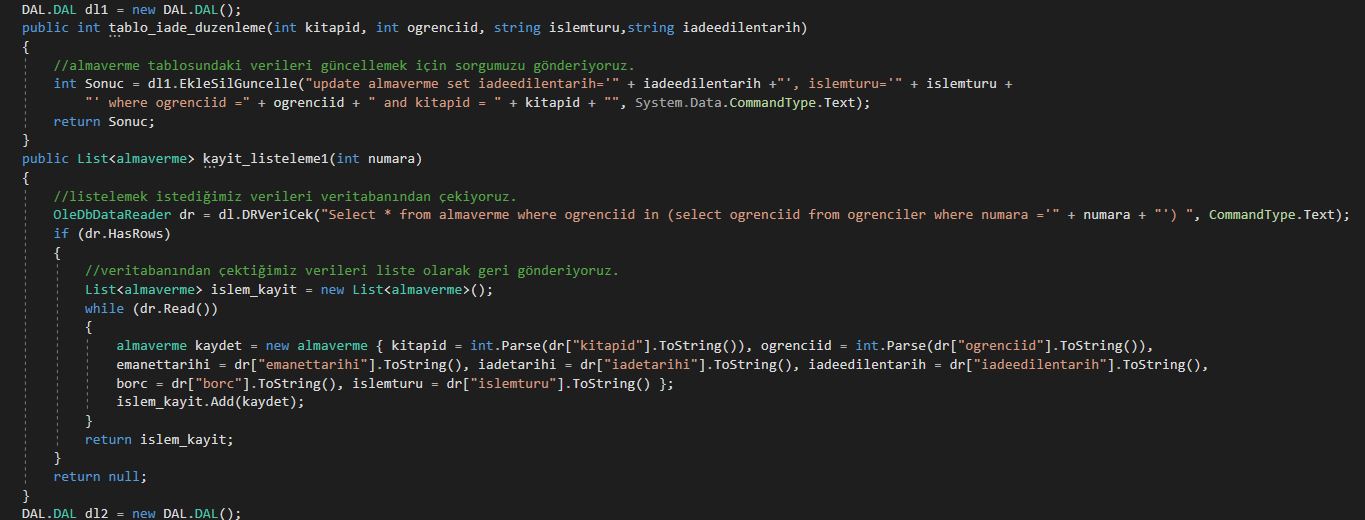
İade Alım İşlemleri classımızdaki fonksiyonlarımıza bakalım.

Kütüphanelerimizi tanımlarken classlardaki değişkenleri kullanacağımızdan Entity katmanımızı da referans olarak aldıktan sonra dahil ediyoruz. Aynı şekilde Database’imizden verileri çekebilmek için referans olarak alıp kütüphanelere dahil ediyoruz.



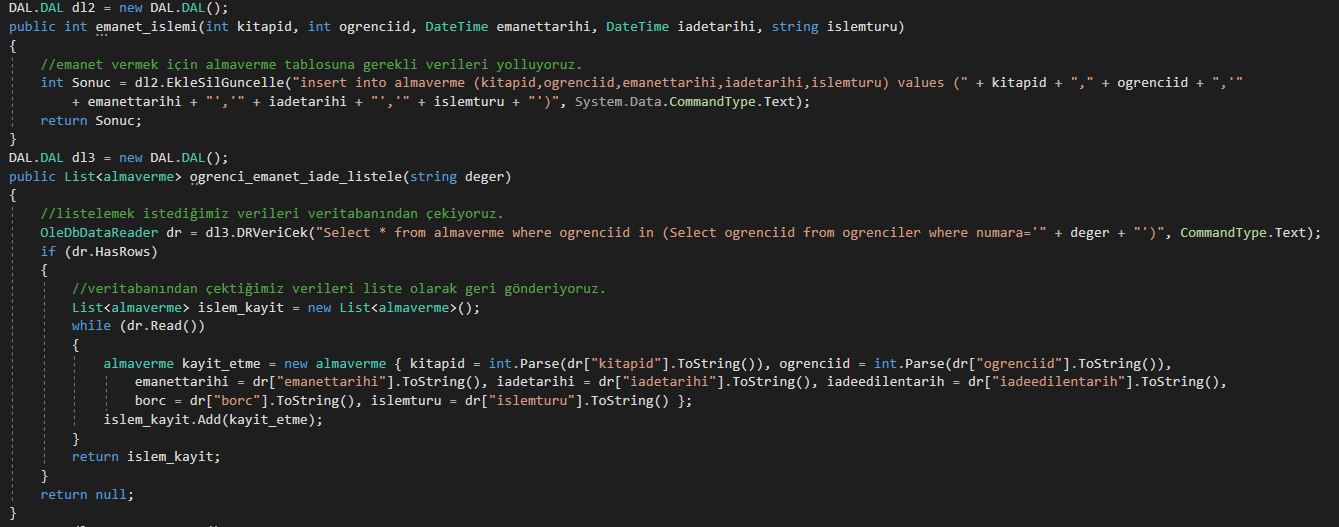
Şekil 11.İş Katmanı- İade Alım İşlemleri

Liste oluşturup databaseden aldığımız verileri listemize yüklüyoruz.



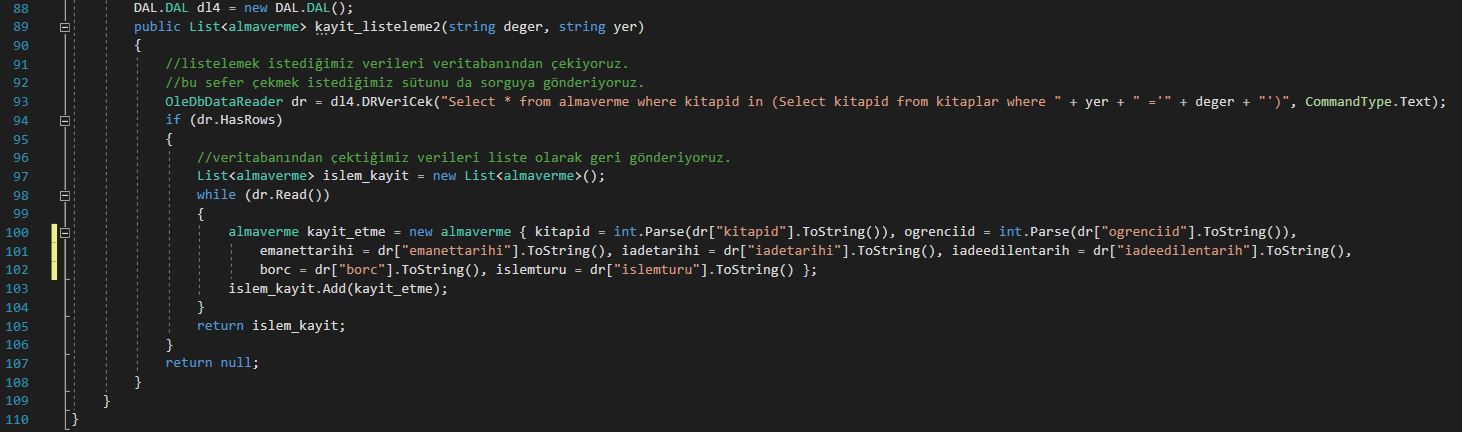
Şekil 12.İş Katmanı- İade Alım İşlemleri Kayıt Listeleme

Kayıtlarımızı güncellemek istediğimizde programa sorgu gönderiyoruz.



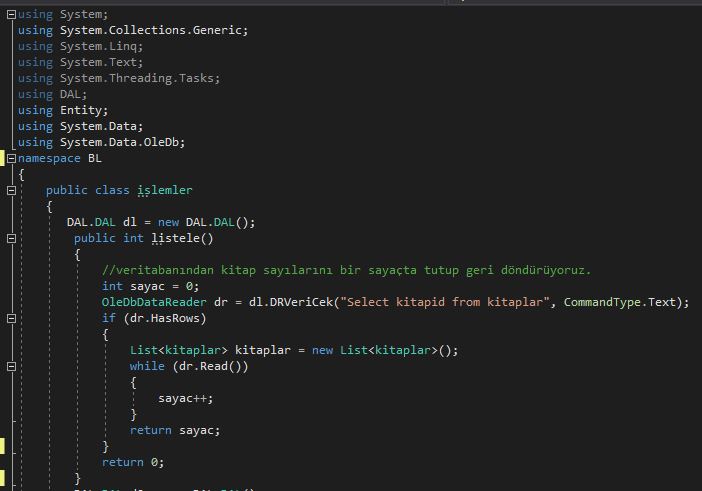
Şekil 13.İş Katmanı- İade Alım İşlemleri Emanet Kısmı

Emanetle alakalı işlemlerimizi tamamladık. Sütun sorgusuna göre listeleme işlemi içinse aşağıdaki kodları kullanıyorum.



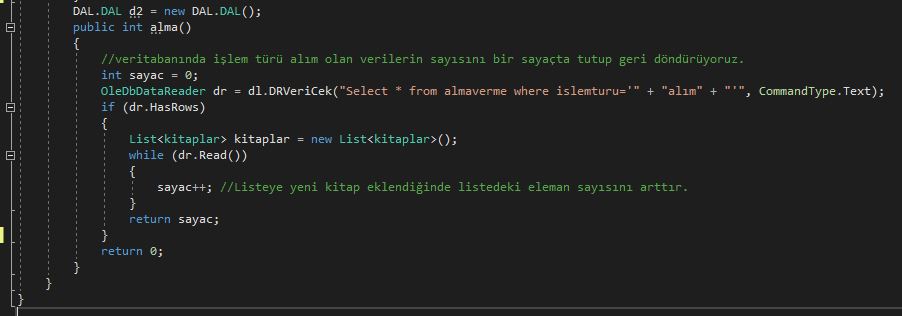
Şekil 14.İş Katmanı Verileri Listeleme

İşlemler class’ı oluşturalım. Bu classımızda kitap id’sine göre listeleme işlemimize bakalım.



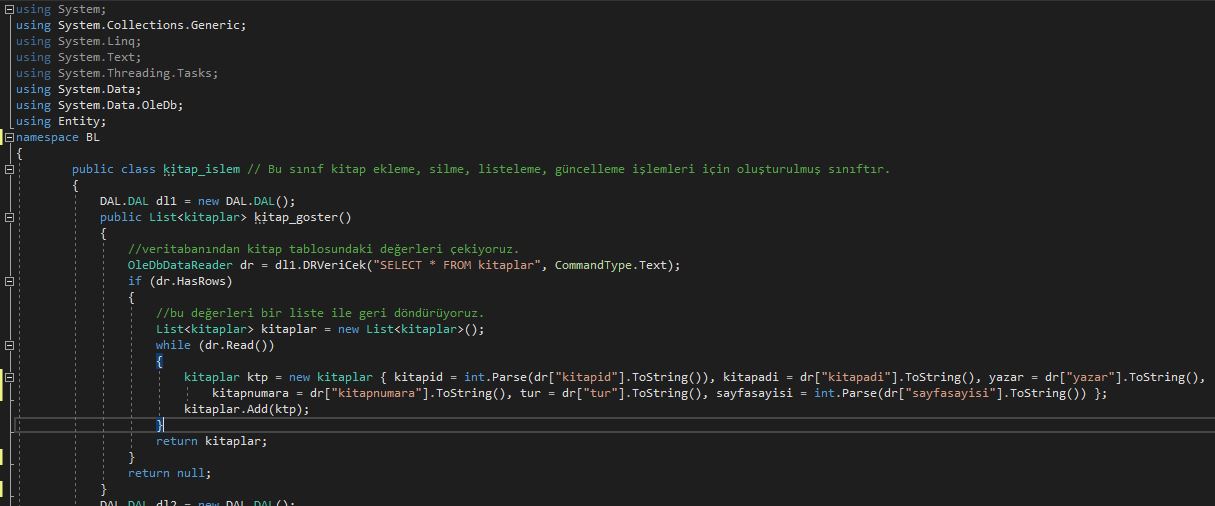
Şekil 15.İşlemler Sınıfı Listeleme Operatörü

Şimdi işlemler classımızın içine kitap alma işlemi gerçekleştiğinde kitaplar listemizin içine eklemesi için gerekli olan kodu yazalım.



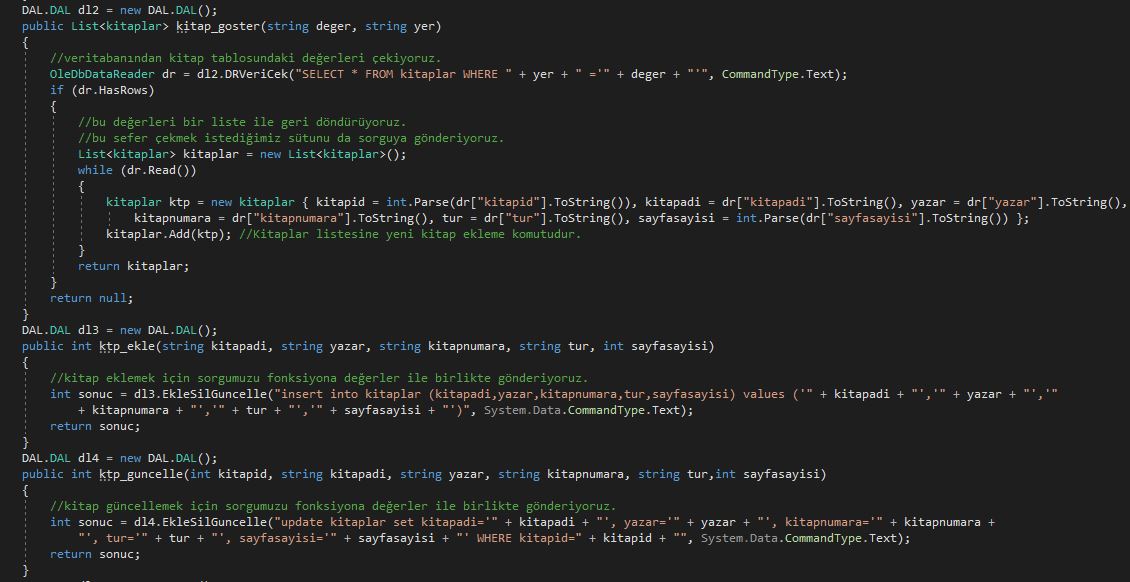
Şekil 16.İşlemler Sınıfı Alma İşlemi

Şimdi kitap işlem sınıfı oluşturalım ve bu sınıfımızda kitap ekleme, listeleme, silme ve güncelleme işlemlerimizi gerçekleştirelim.



Şekil 17.Kitap İşlem Kütüphaneleri

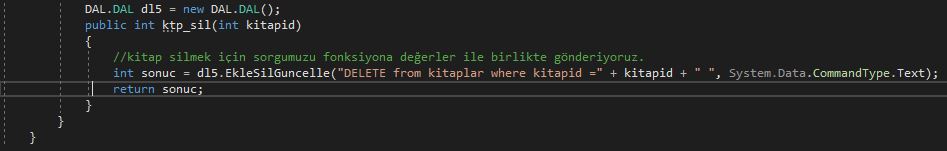
Kitap listesini görüntülemek, yeni kitap eklemek ve güncellemek için:



Şekil 18. Kitap Ekleme ve Güncelleme İşlemi

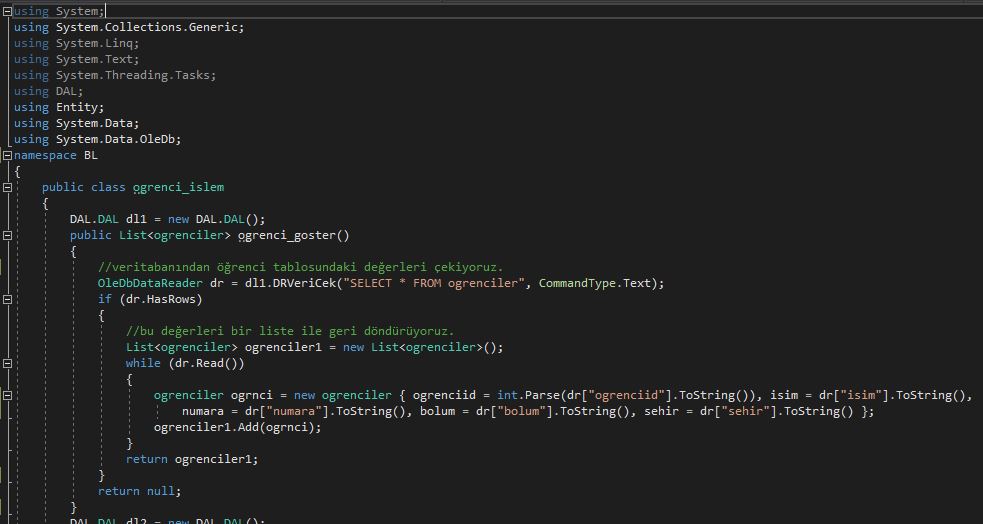
Kitap eklemek ve güncellemek için sorgumuzu fonksiyona değerler ile birlikte gönderiyoruz.

Son olarak kitap silme işlemi için yapılacaklar ise:



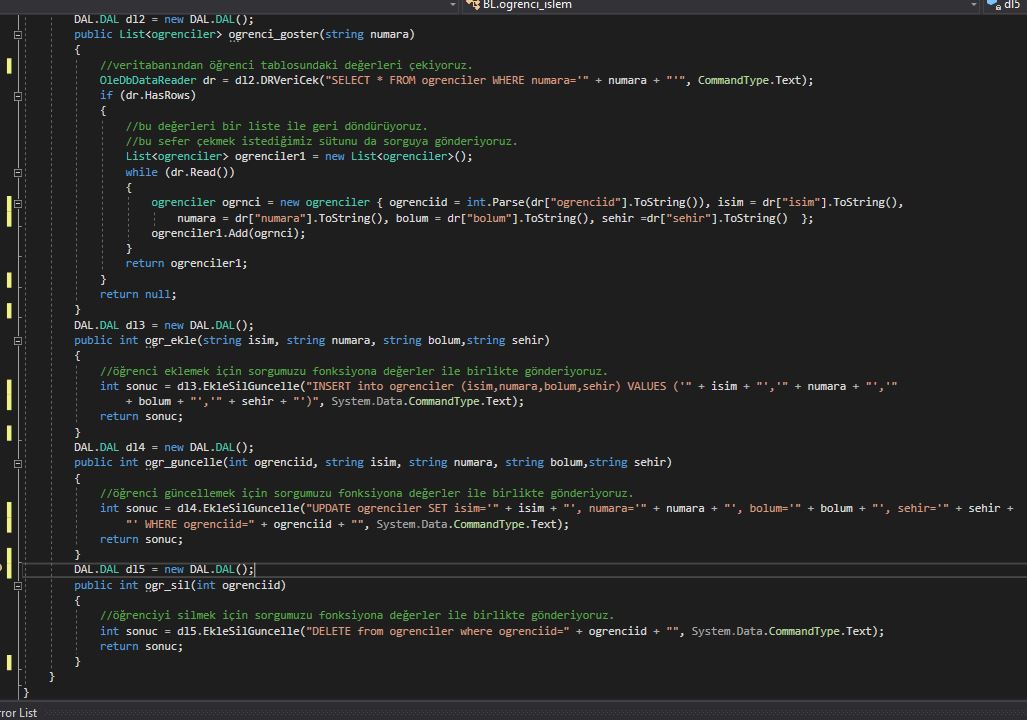
Şekil 19.Kitap Silme İşlemi

Öğrenciyle alakalı işlemlerimiz için öğrenci işlem class’ı tanımlayalım. Öğrenci ekleme, listeleme, güncelleme ve silme işlemlerini bu classımızda gerçekleştirelim.



Şekil 20.Öğrenci İşlemleri Kütüphaneleri

Verilerimizi çekip listemize ekledik. Şimdide ekleme, güncelleme, silme yapalım.



Şekil 21.Öğrenci Ekleme, Güncelleme, Silme İşlemleri

İş katmanımızı tamamladık. Artık form kısmımıza geçebiliriz. Sunum katmanımı final proje olarak adlandırdım. Burada tasarım işlemlerimizi yapacağız.

## *Sunum Katmanı (Presentation Layer)*

Bu katmanda ise veritabanında çektiğimiz verileri kullanıcıya sunma katmanımızdır. Bu katmanda daha çok sayfalarımızın olacağı katmandır. Kullanıcıya bu katmanda ulaşacağız. Burada temel amaç kullanıcıya verileri göstermek ve kullanıcıdan gelen verileri Business Katmanı ile Data Access’e iletmektir. Kullanıcı dilerse buradan veri çekme işlemini başlatacak dilerse burada temel işlemleri yapacaktır.

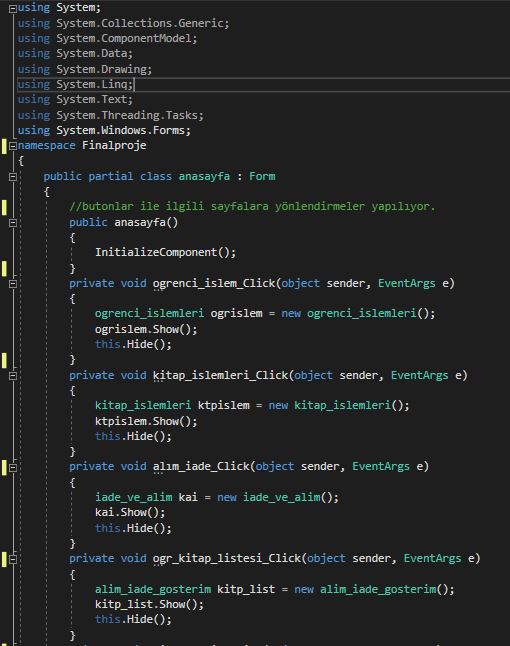
Anasayfamızdan diğer sayfalara geçiş için butonlar oluşturuyoruz.



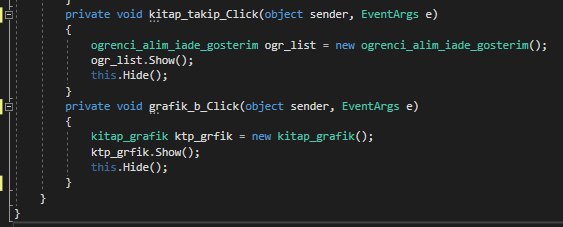
Şekil 22.Anasayfa

Butonlara tıkladığımızda aktif olması için Buton\_Click özelliğini veriyoruz.

Kod kısmı aşağıdaki gibidir.

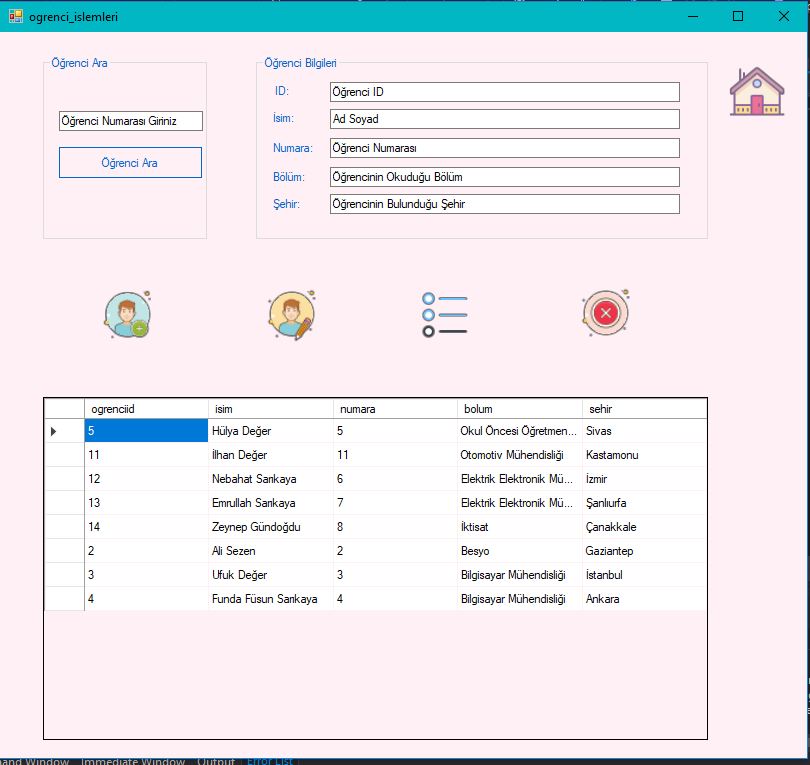


Şekil 23.Anasayfa Kütüphaneleri



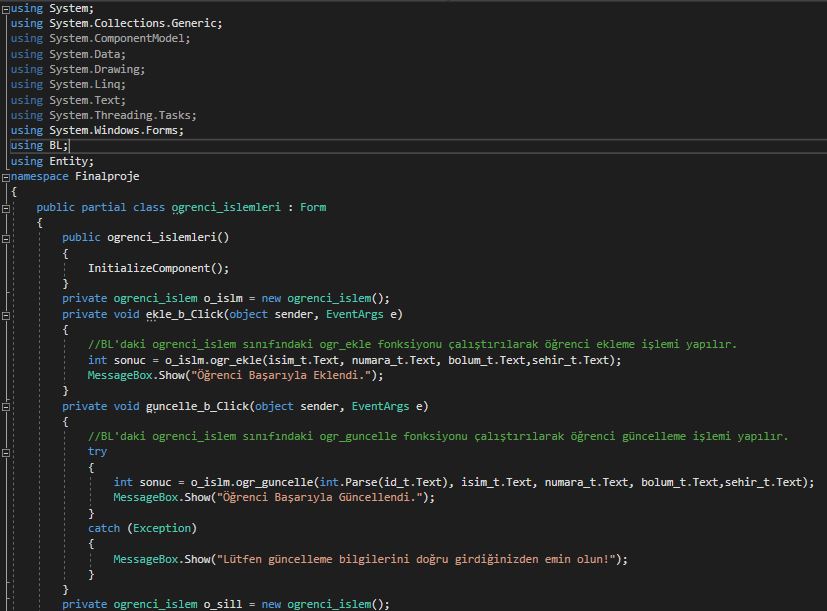
Şekil 24.Anasayfa İşlemleri

Öğrenci işlemleri sayfamızda öğrenci id’si ile öğrenci güncelleme ve silme işlemlerimizi gerçekleştirebiliyoruz. Yeni öğrenci ekleyebiliyoruz. Mevcut öğrenci listemizi görüntüleyebiliyoruz.

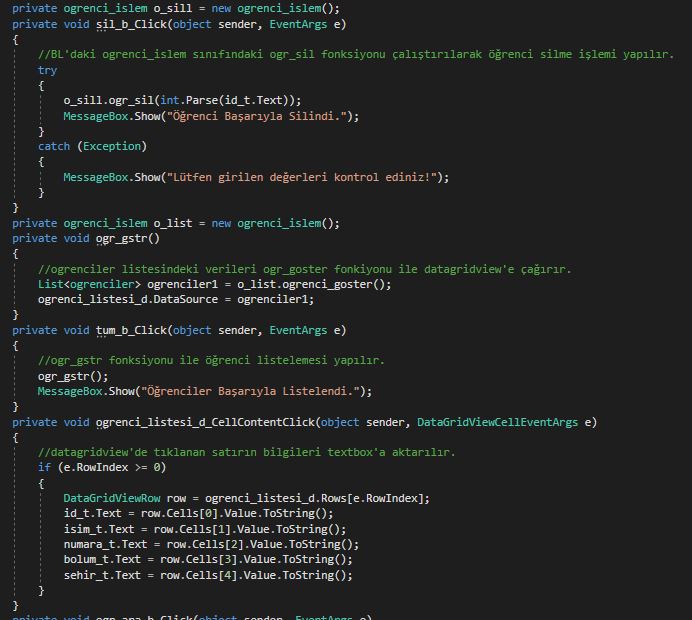


Şekil 25.Öğrenci İşlemleri Form Görüntüsü

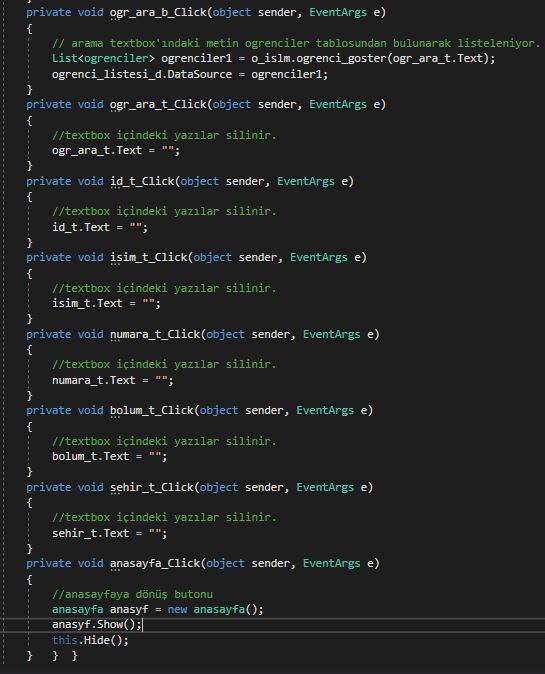
Şimdi bu tasarımın arka planında neler olduğuna bakalım.



Şekil 26.Öğrenci İşlemleri Kütüphaneleri

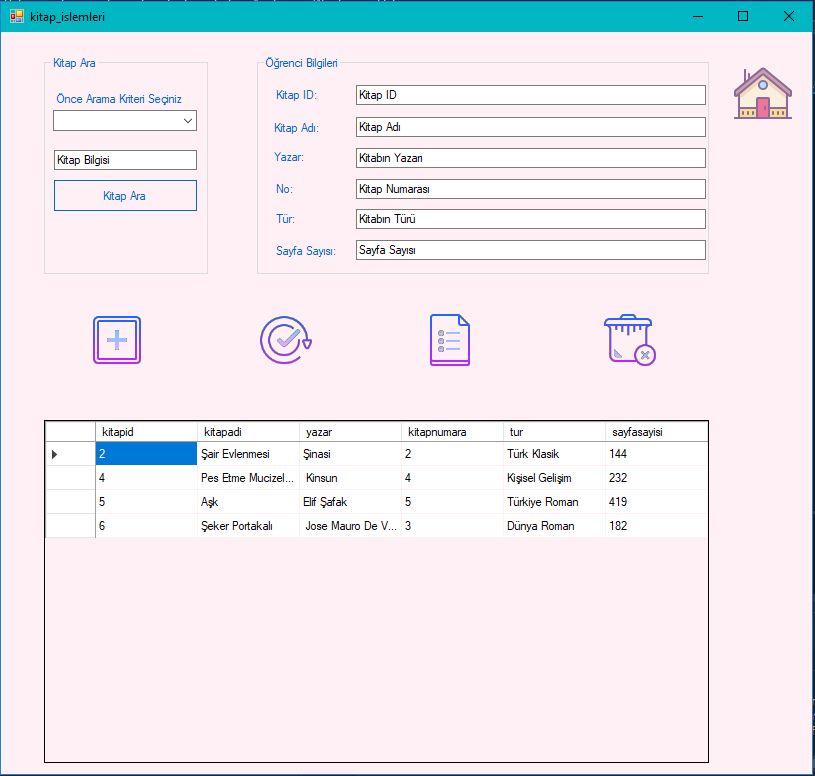


Şekil 27. Öğrenci Listeleme İşlemleri



Şekil 28.Öğrenci Arama İşlemi

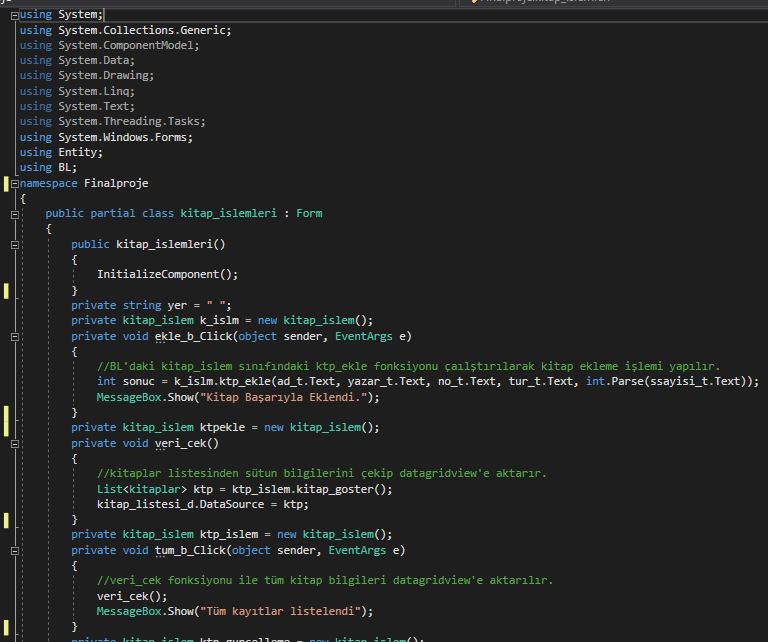
Öğrenci işlemleri sayfamızı tamamladık. Şimdi kitap işlemleri sayfamızı inceleyelim.



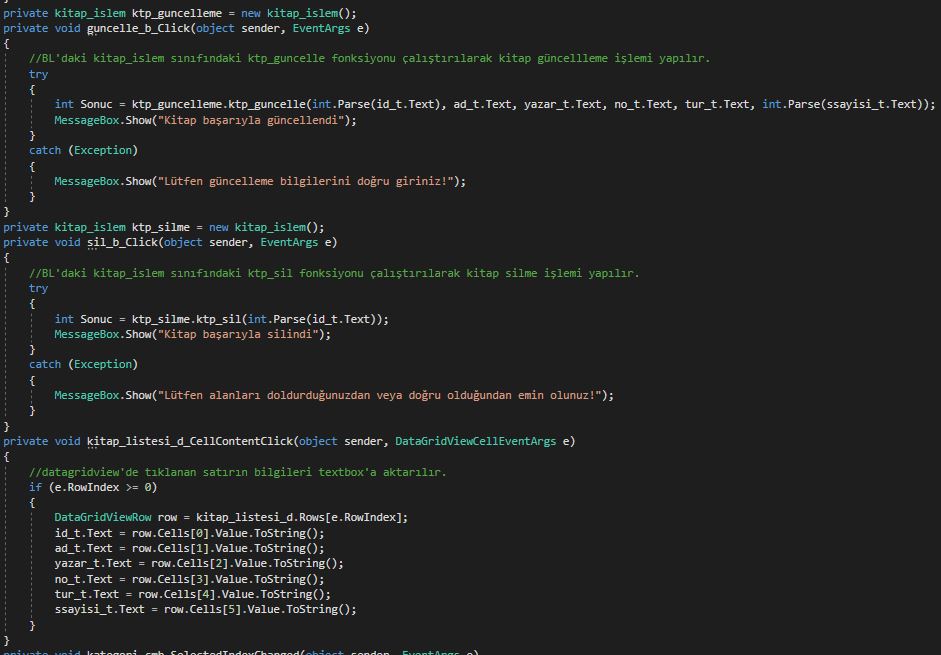
Şekil 29.Kitap İşlemleri

Kitap işlemleri sayfamızda öğrenci id’si ile kitap güncelleme ve silme işlemlerimizi gerçekleştirebiliyoruz. Yeni öğrenci ekleyebiliyoruz. Mevcut kitap listemizi görüntüleyebiliyoruz.

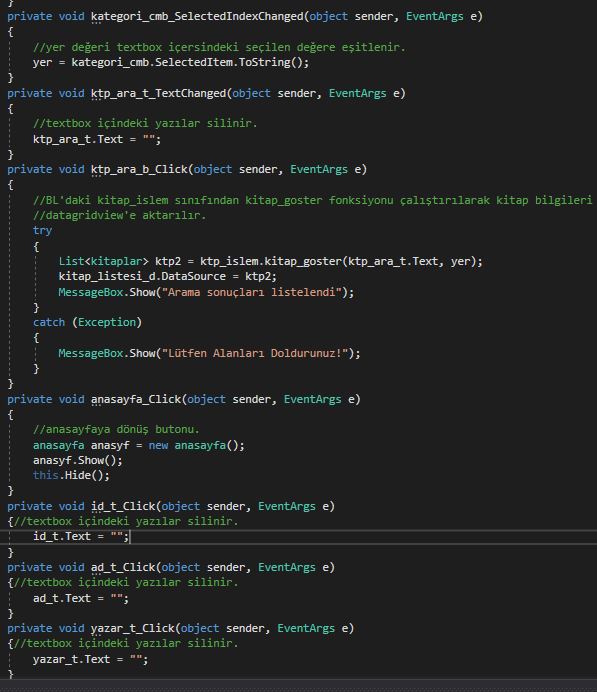
Şimdi bu tasarımın arka planında neler olduğuna bakalım.



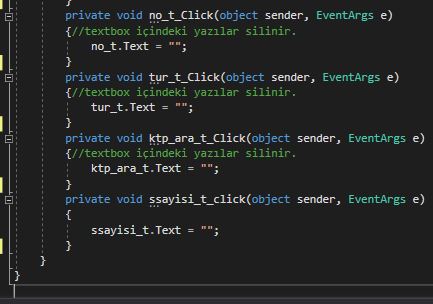
Şekil 30.Kitap Ekleme İşlemi



Şekil 31.Kitap Güncelleme, Listeleme ve Silme İşlemleri

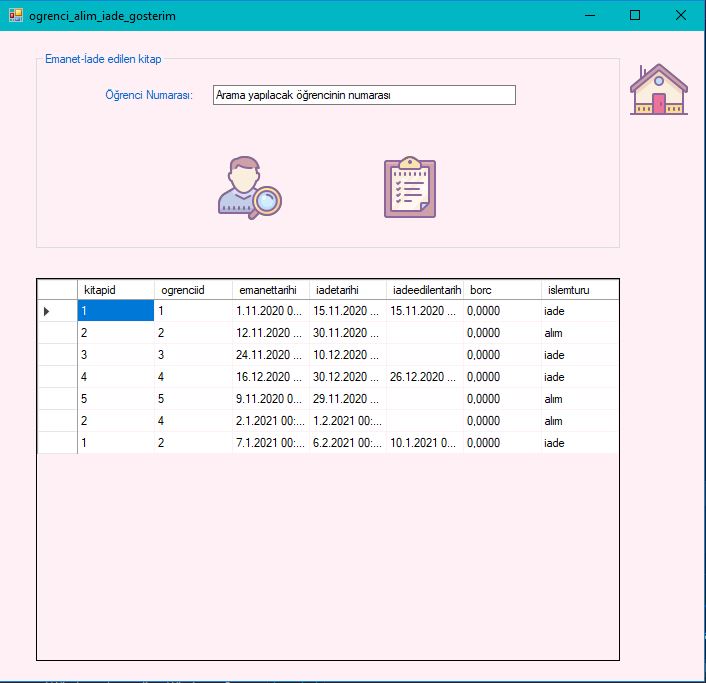


Şekil 32.Kitap Arama İşlemi



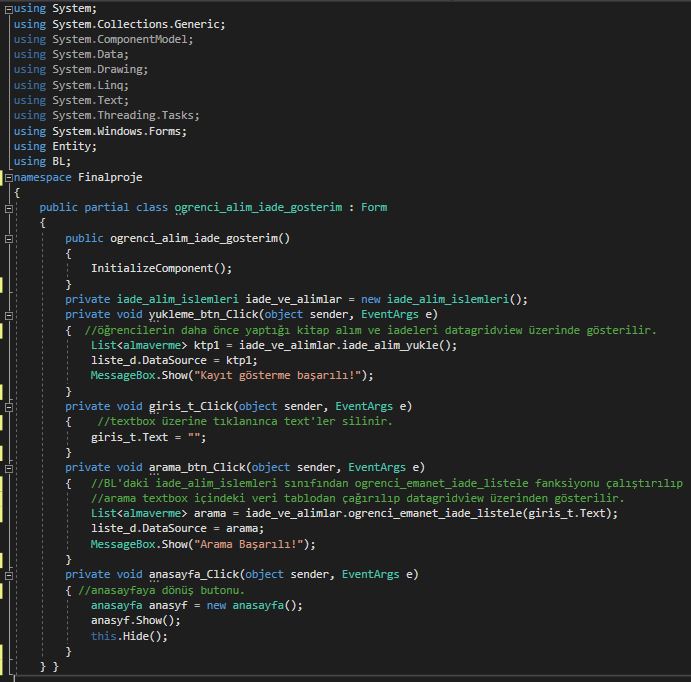
Şekil 33.Kitap Bilgileri

Öğrenci Emanet-İade Listesi butonumuzun içine bakalım.



Şekil 34.Öğrenci Emanet İade Listesi

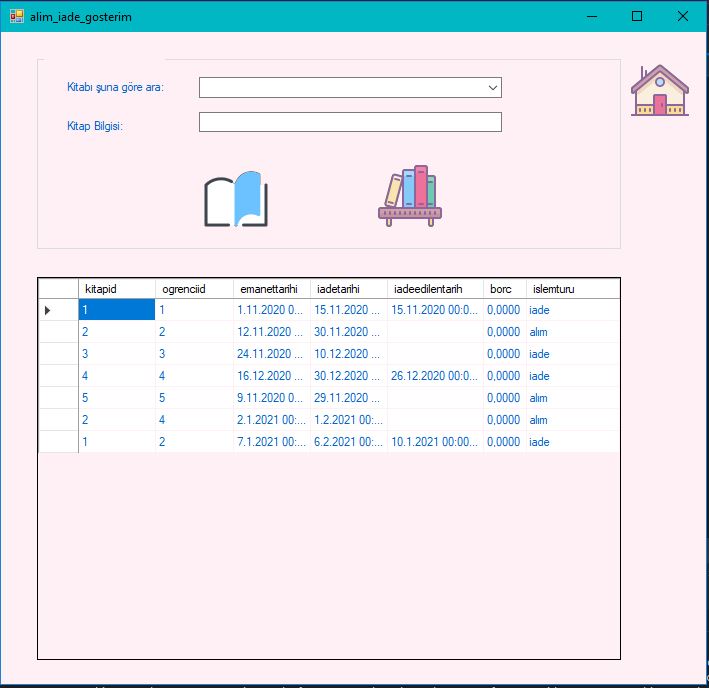
Listeleme işlemi yaptığımızda hangi öğrencinin hangi tarihlerde hangi kitabı aldığını veya iade ettiğini gösterir. Bu arayüzün arkasında ise aşağıdaki kodlar yer alır.



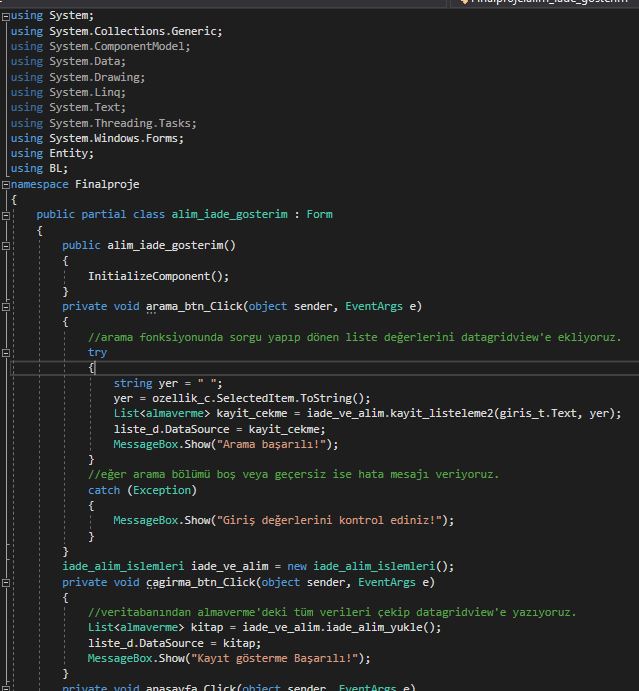
Şekil 35.Öğrenci Emanet İade Listesi Fonksiyonları

Öğrencilerin daha önceden aldığı kitapları tablomuza getirmek için oluşturduğumuz listeyi getiriyoruz.

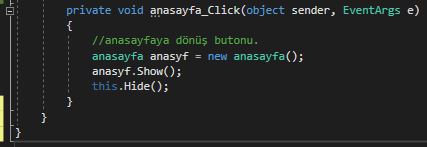
Kitap Emanet -İade listesine bakalım.



Şekil 36.Kitap Emanet İade Listesi

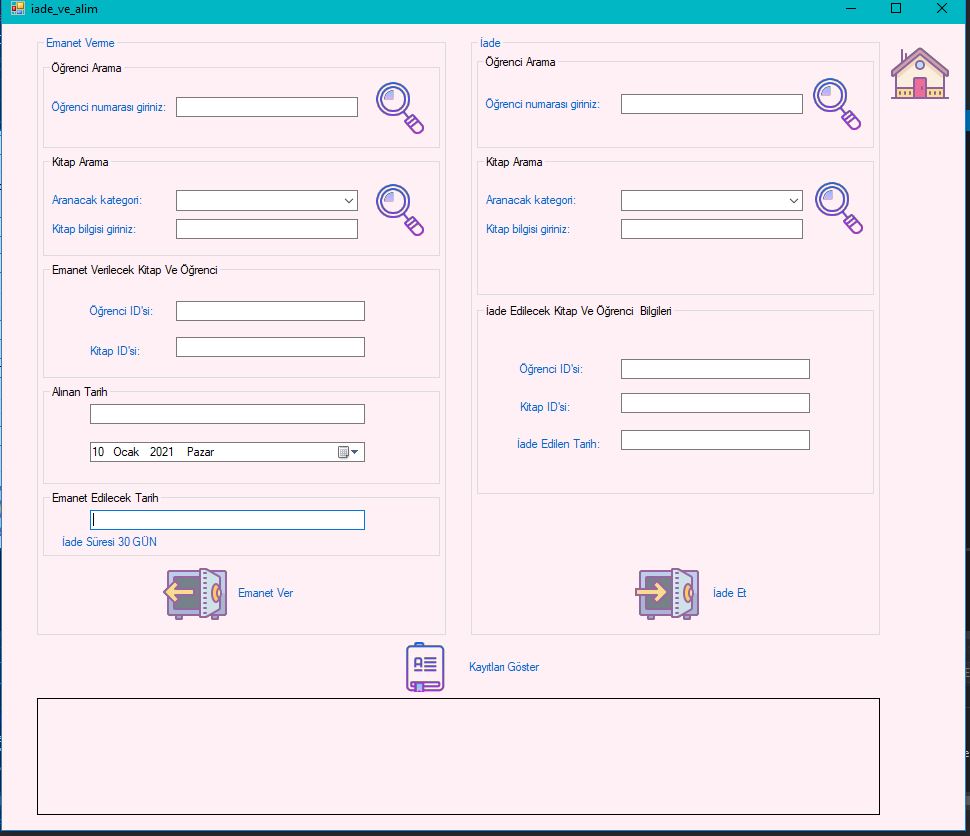
Kitap adı, yazar, tür bilgisine göre arama işlemi gerçekleştirebiliriz. Listeleme kısmımızda kitap id’sine göre hangi öğrencide o kitabın bulunduğunu, alım veya iade durumunu listeleyebiliriz.

Şekil 37.Kitap Emanet İade Listesi Kütüphaneleri



Şekil 38.Anasayfaya Dönme Fonksiyonu

Kitap Emanet ve İade İşlemleri sayfamıza bakalım.

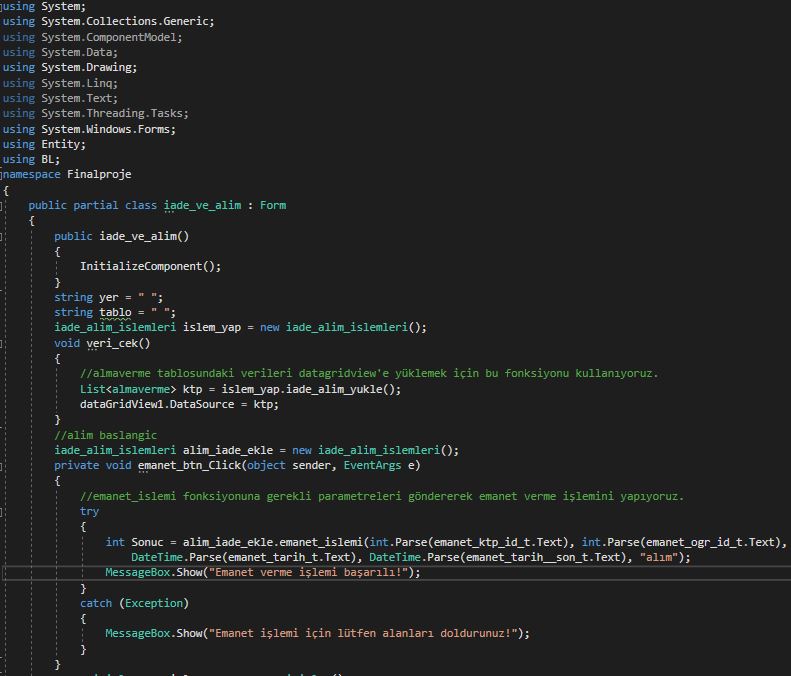


Şekil 39.Kitap Emanet İade İşlemleri

Bu sayfamızda kitap emanet verme ve kitap iade etme işlemlerimizi kitap ve öğrenci id’si bilgileri ile gerçekleştiriyoruz.

Kayıtları göster butonu ile listeleme yapıyoruz.

Listeleme kısmımızda kitabın durumuna göre satırımız renkleniyor. Yani eğer kitap iade edildiyse satır yeşil, kitap başka bir öğrenci fakat iade süresi geçtiyse kırmızı olarak renklendiriyoruz.



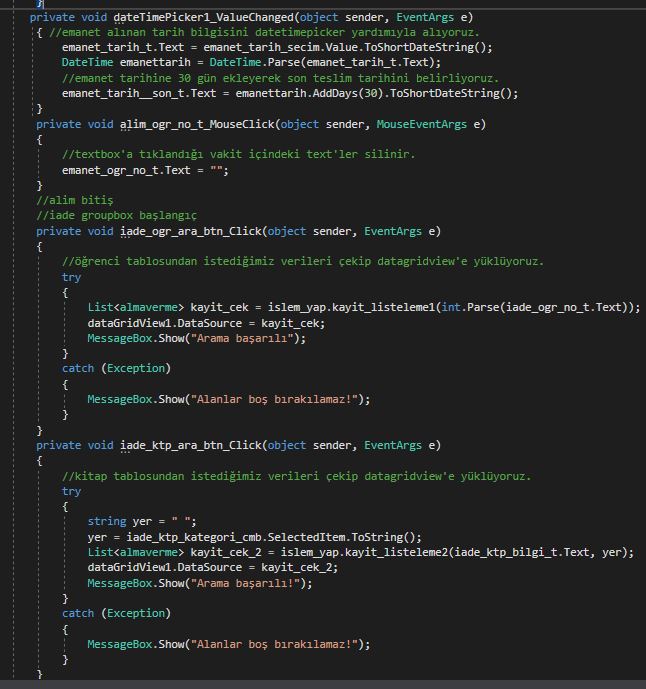
Şekil 40.Kitap Emanet Ve İade İşlemleri Kütüphaneleri

Entity ve Business Layer katmanındaki fonksiyonları kullandığımız için projemizin kütüphanesine ekliyoruz.



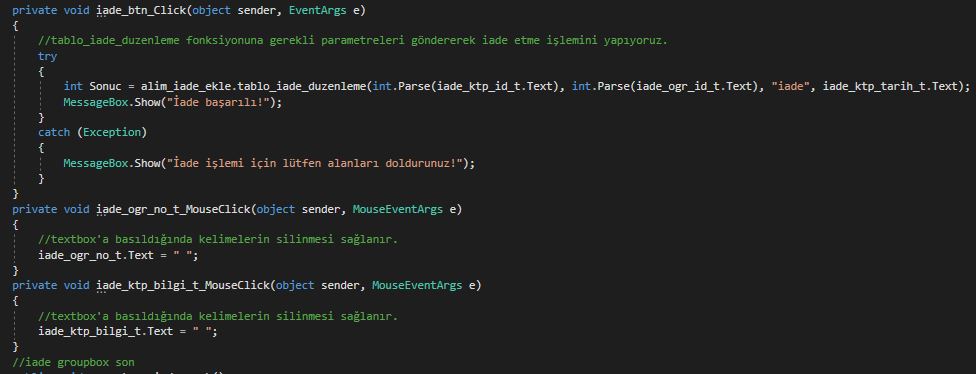
Şekil 41.Kitap Emanet ve İade İşlemleri Fonksiyonları

Kitap tablosundan aradığımız kitabın bilgisini kendi tablomuza yüklüyoruz.

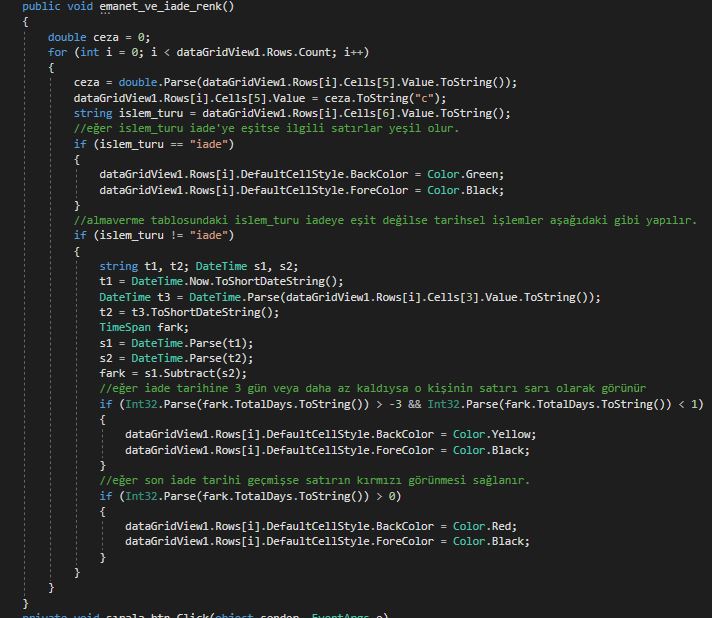


Şekil 42.Kitabın Teslim Tarihinin Belirlenmesi

Kitap emanet olarak alındıktan sonra teslim süresi en geç 30 gün sonrası olarak ayarlıyoruz.

İade işlemleri gerçekleşirken aşağıdaki kod satırları yürütülür. 

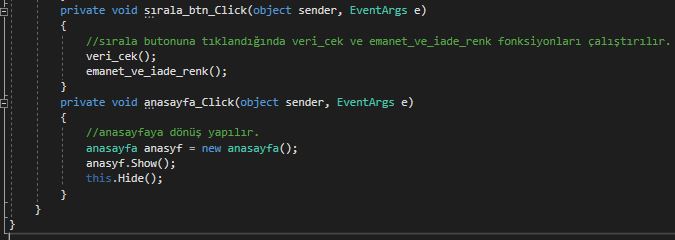
Şekil 43.Kitap İade Fonksiyonları



Şekil 44.Listenin Renklendirilmesi

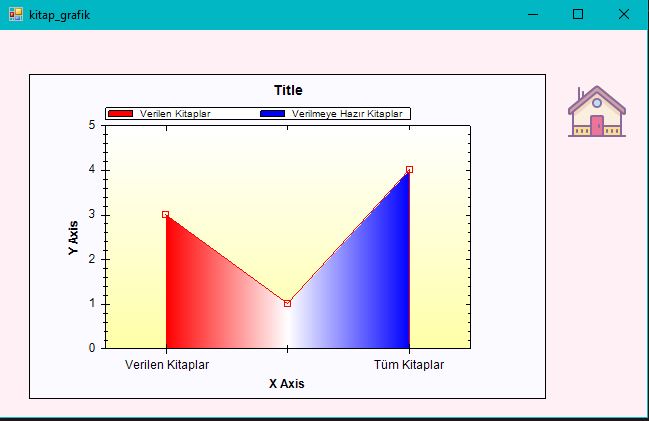
Emanet ve iade işlemlerinde satırın renklendiğinden bahsetmiştim. Rengin ne olması gerektiğini yukardaki sorgularla belirliyoruz.

Yaptığımız sorguya göre listenin sıralanmasını ve anasayfaya dönmek için gerekli olan butonu aşağıdaki gibi aktif hale getiriyoruz.



Şekil 45.Listenin Sıralanması

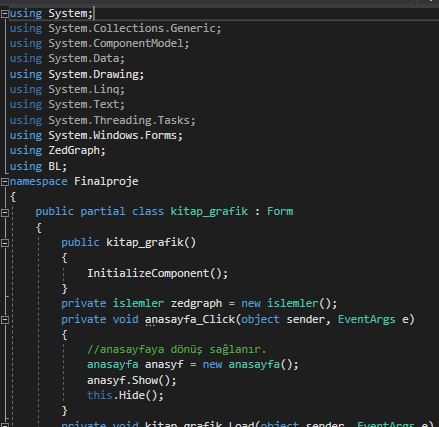
Kitaplarımızın verilmeye hazır veya bir öğrencide bulunduğunu gösteren grafiğimizin tasarımına bakalım.



Şekil 46.Grafik Görünümü

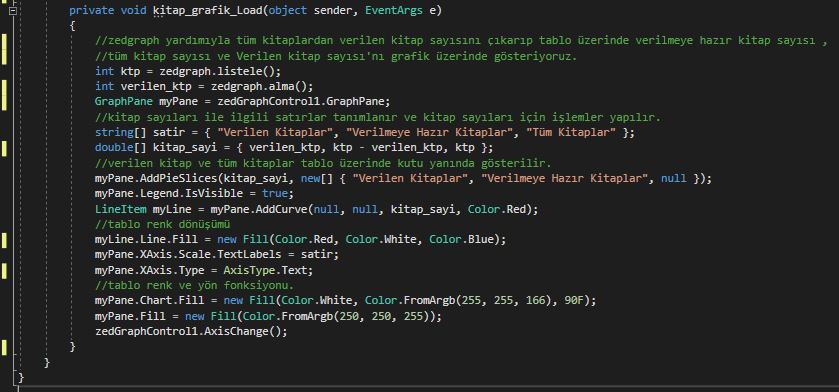
Bu tasarımı yapabilmek için Zedgraff.dll uzantısını dosyamıza eklemeliyiz. Zedgraff Control ile grafiğimizi oluşturduktan sonra kod kısmımıza bakabiliriz.

İlk başta Zedgraff kütüphanemizi dahil edelim ve anasayfaya dönme butonumuzu aktifleştirelim.



Şekil 47.Grafik Kütüphaneleri

Şimdide grafiğimizin kod kısımlarına bakalım.



Şekil 48.Grafik Fonksiyonları

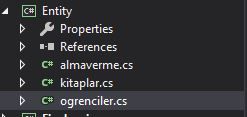
Grafiğimizin renklendirmesini ve adlandırmasını gerçekleştirdik.

Şimdi Solution yazan kısımda sağ tık Add>New Project> Class Library (.NET Framework) diyerek Entity Katmanımızı oluşturalım.

## *Entities Katmanı*

Bu katmanda proje boyunca kullanacağımız ana classlarımızı belirliyoruz yani gerçek nesnelerimizi belirlediğimiz yer burası. Bu katmanı hem Data Access hem Business hem de Presentation katmanı kullanmaktadır.

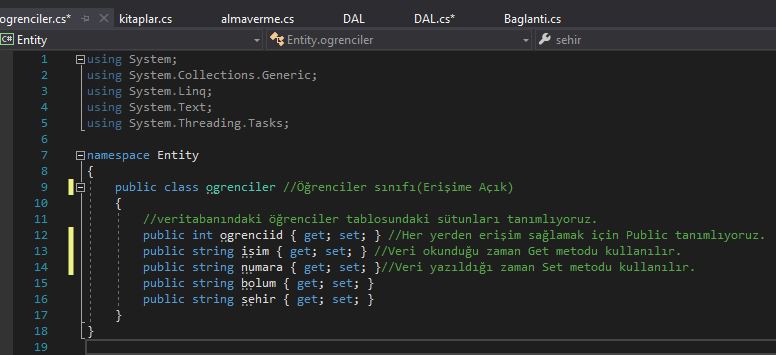
3 adet Sınıf oluşturalım. Sınıflarımızı oluştururken kod okunabilirliği açısından Access tablolarımızla aynı isimleri verelim.



Şekil 49.Entity Sınıfları

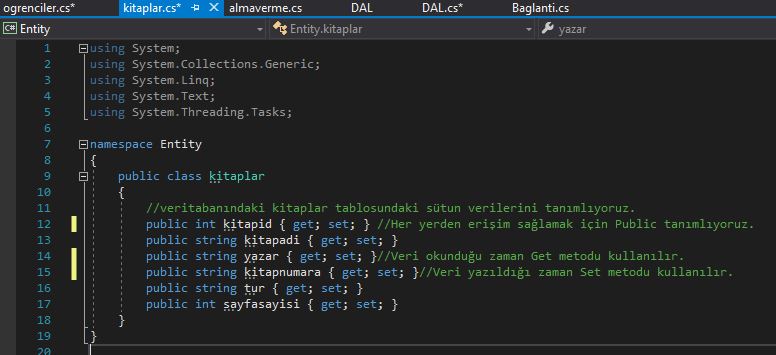
Access de oluşturduğumuz tablolardaki verilerimizi aynı şekilde adlandırıp class içerisinde tanımlamalarımızı gerçekleştirelim.

*Öğrenciler Sınıfımıza Ait Kodlar*



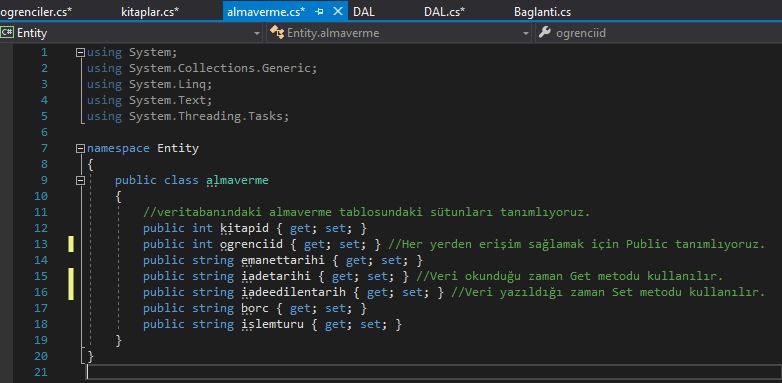
Şekil 50.Öğrenciler Sınıfı

*Kitaplar Sınıfımıza Ait Kodlar*



Şekil 51.Kitaplar Sınıfı

*Alma Verme Sınıfımıza Ait Kodlar*



Şekil 52.Alma Verme Sınıfı

Access bağlantımızı yaptık. Sınıflarımızı tanımladık. Form tasarımlarımızı oluşturduk. Artık projemiz sorunsuz bir şekilde çalışır.

## *Projenin Setup Dosyası*

Projenin Setup Dosyası

C:\Users\FundaFüsun\Desktop\Finalproje\Finalproje\Kutuphane\Debug

Bu yolda yer almaktadır.