KONU: KATMANLI MİMARİ İLE KÜTÜPHANE OTOMASYONU

Melisa AK

OCAK 2021

**Şekiller tablosu**

[**Şekil 1-Katmanlı Mimari Yapısı** 5](#_Toc61446022)

[**Şekil 2-Veritabanı tabloları** 10](#_Toc61446023)

[**Şekil 3-Öğrenciler Tablosu** 10](#_Toc61446024)

[**Şekil 4-Kitaplar tablosu** 11](#_Toc61446025)

[**Şekil 5-Alma Verme tablosu** 11](#_Toc61446026)

[**Şekil *6*- Veritabanı İlişkileri** 11](#_Toc61446027)

[**Şekil *7*-DAL Bağlantı Class’ı** 12](#_Toc61446028)

[**Şekil 8-DAL DAL Class’ı** 13](#_Toc61446029)

[**Şekil 9-DAL DAL Class’ı** 14](#_Toc61446030)

[**Şekil 10-DAL DAL Class’ı** 15](#_Toc61446031)

[**Şekil 10-BL Kitap işlem Class’ı** 16](#_Toc61446032)

[**Şekil 11-BL Kitap işlem Class’ı** 16](#_Toc61446033)

[**Şekil 12-BL Kitap işlem Class’ı** 17](#_Toc61446034)

[**Şekil 13-BL Öğrenci işlem Class’ı** 17](#_Toc61446035)

[**Şekil 14-BL Öğrenci işlem Class’ı** 18](#_Toc61446036)

[**Şekil 15-BL Öğrenci işlem Class’ı** 18](#_Toc61446037)

[**Şekil 16-BL İade Emanet işlem Class’ı** 19](#_Toc61446038)

[**Şekil 17-BL İade Emanet işlem Güncelleme Class’ı** 19](#_Toc61446039)

[**Şekil 18-BL İade Emanet işlem Listeleme Class’ı** 19](#_Toc61446040)

[**Şekil 19-BL İade Emanet işlem Emanet verme Listeleme Class’ı** 20](#_Toc61446041)

[**Şekil 20-BL İade Emanet işlem Emanet verme Listeleme Class’ı** 20](#_Toc61446042)

[**Şekil 21-BL Grafik Class’ı** 20](#_Toc61446043)

[**Şekil 22- Anasayfa** 21](#_Toc61446044)

[**Şekil 23- Anasayfa Form** 22](#_Toc61446045)

[**Şekil 24- Öğrenci İşlemleri** 23](#_Toc61446046)

[**Şekil 25- Öğrenci Ekleme Butonu** 23](#_Toc61446047)

[**Şekil 26- Öğrenci Güncelleme Butonu** 24](#_Toc61446048)

[**Şekil 27- Öğrenci Silme Butonu** 24](#_Toc61446049)

[**Şekil 28- Öğrenci Listeleme İşlemleri** 24](#_Toc61446050)

[**Şekil 29- Öğrenci Arama Butonu** 25](#_Toc61446051)

[**Şekil 30- Öğrenci textbox işlemleri** 25](#_Toc61446052)

[**Şekil 31- Kitap İşlemleri** 26](#_Toc61446053)

[**Şekil 32- Kitap Ekleme Butonu** 26](#_Toc61446054)

[**Şekil 33- Kitap Güncelleme Butonu** 27](#_Toc61446055)

[**Şekil 34- Kitap Silme Butonu** 27](#_Toc61446056)

[**Şekil 35- Kitap Arama Butonu** 27](#_Toc61446057)

[**Şekil 36- Kitap Listeleme Butonu** 28](#_Toc61446058)

[**Şekil 37- Kitap textbox işlemleri** 28](#_Toc61446059)

[**Şekil 38- Kitap Emanet ve İade İşlemleri** 29](#_Toc61446060)

[**Şekil 39- Kitap Emanet verme İşlemleri** 29](#_Toc61446061)

[**Şekil 40- Emanet Öğrenci Arama İşlemleri** 30](#_Toc61446062)

[**Şekil 41- Emanet Kitap Arama İşlemleri** 30](#_Toc61446063)

[**Şekil 42- Emanet Kitap Kategori Arama İşlemleri** 30](#_Toc61446064)

[**Şekil 43- Emanet Tarih İşlemleri** 31](#_Toc61446065)

[**Şekil 44- İade Öğrenci Arama İşlemleri** 31](#_Toc61446066)

[**Şekil 45- İade Kitap Arama İşlemleri** 31](#_Toc61446067)

[**Şekil 46- İade Verme İşlemleri** 32](#_Toc61446068)

[**Şekil 47- İade Renklendirme İşlemleri** 32](#_Toc61446069)

[**Şekil 48- İade Renklendirme İşlemleri Görünümü** 33](#_Toc61446070)

[**Şekil 49- Öğrenci Emanet kitap listeleme İşlemleri** 33](#_Toc61446071)

[**Şekil 50- Öğrenci Emanet kitap listeleme İşlemleri Formu** 34](#_Toc61446072)

[**Şekil 51- Kitap Emanet listeleme İşlemleri** 34](#_Toc61446073)

[**Şekil 52- Kitap Emanet listeleme İşlemleri Formu** 35](#_Toc61446074)

[**Şekil 53- Kitap Arama listeleme İşlemleri Formu** 36](#_Toc61446075)

[**Şekil 54- Grafik İşlemleri** 37](#_Toc61446076)

[**Şekil 55- Grafik İşlemleri Formu** 37](#_Toc61446077)

[**Şekil 56- Entity Kitaplar** 38](#_Toc61446078)

[**Şekil 57- Entity Öğrenciler** 38](#_Toc61446079)

[**Şekil 58- Entity Alma Verme** 39](#_Toc61446080)

İçindekiler

[I.Katmanlı Mimari 5](#_Toc61444321)

[Veri Katmanı (Data Layer) 6](#_Toc61444322)

[İş Katmanı (Business Layer) 7](#_Toc61444323)

[Sunum Katmanı (Presentation Layer) 7](#_Toc61444324)

[Entity 7](#_Toc61444325)

[1-Uygulama 8](#_Toc61444326)

[1-Kullanılan Teknolojiler 8](#_Toc61444327)

[**1.1. Microsoft Visual Studio 2019** 8](#_Toc61444328)

[**1.2. C# 4.0 Programlama Dili** 8](#_Toc61444329)

[**1.3.ORM ve Ado.Net Entity Framework** 9](#_Toc61444330)

[**1.4.Microsoft Access Veritabanı** 9](#_Toc61444331)

[2-Uygulama Mimarisi 10](#_Toc61444332)

[**2.1.Veritabanı Katmanı (Data Access Layer)** 10](#_Toc61444333)

[**2.2.İş Katmanı (Business Logic Layer)** 15](#_Toc61444334)

[**2.3.Sunum Katmanı (Presentation Layer)** 21](#_Toc61444335)

[**2.4.Entity Layer** 38](#_Toc61444336)

# 

# I.Katmanlı Mimari

**Katmanlı** **Mimari** projelerin daha derli toplu durmasını sağlayan, kodun okunulabilirliğini arttıran, ekip çalışmasını arttıran, hata yönetiminin kolay olmasını sağlayan bir yapıdır. Katmanlara ayrılmış bir projeyi gerektiğinde birden fazla geliştirici kodlayabileceği gibi oluşan sorunların hangi katmanda ortaya çıktığı da kolayca tespit edilebilmektedir. Gerekli katmanda yapılan iyileştirme uygulamanın genelinde aynı etkiyi yaratacağından uzun vadede geliştirme zamanı, test süreleri ve alınacak riskleri azaltmaktadır. Aslında bu yapı ile proje yazımını bir standart hale getirilmiş olur. Bu yapı 3 ana katmandan oluştuğu gibi günümüzde **Çok Katmanlı Mimari** yapıda denebilir. Ama aslında 3 ana katman üzerinde durur. Bu katmanlar;

* Veri Katmanı (Data Layer)
* İş Katmanı (Business Layer)
* Sunum Katmanı (Presenation Layer)

Sunum Katmanı (Presenation Layer)

İş Katmanı

(Business Layer)

Veri Katmanı

(Data Layer)

Database

**Şekil 1-Katmanlı Mimari Yapısı**

Yazılımın kolay geliştirilebilir olması çok önemlidir. Otomasyon sistemleri, çok fazla değişikliğe uğrayabilecek sistemler olduğundan (yeni algılayıcı kullanımı, yeni özellik eklenmesi, bir özelliğin çıkarılması vb.) yazılım sisteminin değişikliklere açık bir yapıda olması gereklidir. Yazılımın aynı zamanda donanım bağımsız olması gerekmektedir.

Bu sayede yazılım mekanizmalarında yapılan değişiklikler üst seviyede kalır. Aynı zamanda bir kez hazırlanan yazılım, farklı farklı donanımlarla kullanılabilecektir.

Çoklu yöntemli otomasyon sistemi uygulamalarında kullanılan donanımlar da farklılık gösterebileceğinden, donanımdan bağımsız bir yazılım çok önemlidir. Uzaktan çoklu yöntemle denetimde, sistem yazılımının taşıması gereken genel özellikler şu şekilde sıralanabilinir.

* Gerçek Zamanlı Çalışabilme
* Hızlı yazılım geliştirilebilme, kolayca algoritma değiştirebilme.
* Kolayca ve hızlıca yeni özellikler ekleyebilme, çıkarabilme, değiştirebilme.
* Donanımdan bağımsız olma.
* Çok sayıda denetleyici ile çalışabilme.
* Ortak veritabanı kullanımı.
* Ortak güvenlik sağlanması.

Yazılımlarda veriye nasıl erişileceği, üzerinde nasıl işlemler yapılacağı ve bu işlemlerin kullanıcıya nasıl gösterileceği gibi işlemleri katmanlı mimari ile çok iyi bir şekilde yönetilebilir. Katmanlı mimari sayesinde bu yapıyı parçalara ayırarak bu işlemlerin daha iyi yönetilebilmesi sağlanır. Örneğin Windows form uygulaması geliştirildiğin de ve veri tabanındaki bilgiler ekrana yazdırılmak istendiğinde bu ilgili formun altına gerekli kodlara yazılarak yapılabilir. Ancak projede bir değişiklik yapılacağı zaman tüm kodlar yapılan değişikliğe göre değiştirmek durumunda kalabilir. Birden fazla formun olması durumunda ve bu formların hepsinde benzer işlemlerin yapılması gereksiz kod tekrarına neden olur. Bundan dolayı proje parçalara bölünerek, sürdürülebilir duruma getirilir.

## Veri Katmanı (Data Layer)

Bu katmanda sadece veri tabanı işlemleri yapılmaktadır. Bu katmanın görevi veriyi ekleme, silme, güncelleme ve veri tabanından veri çekme işlemidir. Bu katmanda bu işlemlerden başka herhangi bir işlem yapılmamaktadır. Bu katmanın en önemli özelliği, veri tabanına direkt olarak entegre olan veri tabanıyla direkt konuşan katman olduğundan, kendisine gelen bilgileri veri tabanında oluşturmak (create), güncellemek (update), silmek (delete) veya okumak (read) gibi işlemleri yapacağından, bu bilgileri direkt olarak doğru ve uygun kabul eder. Yani bu bilgilerin ilgili veri tablosuna yapısal olarak uygunluğunu incelemek, gerekli validasyonları yapmak veri katmanının görevi değildir. Diğer katmanlarda, veri katmanına bilgi ulaşana kadar gerekli validasyonlar yapılıp bilgi uygunluğunun kontrol edilmiş olması gerekir.

## İş Katmanı (Business Layer)

Bu katmanda iş yüklerini yazılır. Data Access tarafından projeye çekilmiş olan verileri alarak işleyecek olan katmandır. Uygulamalarda Data Access katmanı direk olarak kullanılmaz. Araya Business katmanı konarak yapılacak olanların Business katmanında yapılması sağlanır. Kullanıcıdan gelen veriler öncelikle Business katmanına gider oradan işlenerek Data Access katmanına aktarılır. Bu katmanda ayrıca bu verilere kimlerin erişeceği belirtilebilir.

Sunum katmanından gelen bilgileri gerekli koşullara göre işleyerek veya denetleyerek veri katmanının sağladığı metotları kullanarak veritabanına gönderen aynı şekilde veritabanından da bilgileri alarak gerekli süreçlerden geçirip presentation katmanına gönderen katmandır.

## Sunum Katmanı (Presentation Layer)

Bu katman kullanıcı ile etkileşimin yapıldığı katmandır. Burası Windows form da olabilir, Web’te olabilir veya Bir Consol uygulaması da olabilir. Burada temel amaç kullanıcıya verileri göstermek ve kullanıcıdan gelen verileri Business Katmanı ile Data Access’e iletmektir. Kullanıcıyla direkt olarak etkileşimde bulunulan katmandır. Bu katman veri katmanının direkt olarak veri tabanıyla bağlantıda olması gibi direkt olarak kullanıcıyla bağlantıdadır. Bu katman aracılığıyla kullanıcı tarafından girilen bilgiler alınır.

## Entity

Entity Katmanında ise genelde domain olarak adlandırılan classlar tanımlanır. Bu Entity katmanının ismi domain olarakta değiştirebilir veya Common katmanıda yapabilir. Bu katmanda proje boyunca classlar tanımlanır.

# 1-Uygulama

1-Kullanılan Teknolojiler

Uygulama kapsamında .Net uygulama mimarisi çerçevesinde geliştirilen teknolojiler kullanılarak geliştirilmiştir. Kullanılan programlar ve teknolojiler:

* Microsoft Visual Studio 2019
* C# 4.0 Programlama Dili
* ORM ve Ado.Net Entity Framework
* Microsoft Access Veritabanı

### **1.1.** **Microsoft Visual Studio 2019**

Microsoft Visual Studio Microsoft tarafından geliştirilmiş olunan bir uygulama geliştirme ortamıdır. Bu ortam ile windows mağazası uygulamaları, masaüstü uygulamaları, mobil uygulamalar, ASP.NET web uygulamaları ve XML web hizmetleri uygulamaları geliştirilebilir Visual Basic, Visual C#, Visual C++, Visual F# ve JavaScript gibi programlama diller ile Visual Studio’da uygulama geliştirilebilir. Kaynak ve makine seviyesinde hata ayıklayıcıya sahiptir. Kod editörünün “code refactoring” ve “intellisense” desteği vardır. Web, windows, sınıf, veri tabanı şema tasarımcısı vb. pek çok yerleşik araca sahiptir. Bu araçlar sayesinde çok daha hızlı ve doğru bir şekilde uygulama geliştirmek mümkündür. Bu yerleşik araçlar dışında birlikte kullanılabilinen pek çok eklentiyi desteklemektedir. Bu eklentiler direk olarak uygulama geliştirme ortamı üzerinden kullanılabilinmektedir.

### **1.2. C# 4.0 Programlama Dili**

Microsoft tarafından geliştirilmiş C++ ve Java dillerine benzeyen bir programlama dilidir. C# nesne tabanlı, tip güvenli (type-safe) ve gösterici (pointer) kullanılabilinen bir dildir. Sözdizimi (syntax) olarak Java diline benzemektedir. C ve C++’ın aksine C# dili hem alt seviye hem de üst seviyede program geliştirilmesini mümkün kılar.

### **1.3.ORM ve Ado.Net Entity Framework**

Normal şartlarda veri tabanında bir tablo için nesne yönelimli bir uygulama geliştirirken bu tabloya karşılık gelen bir sınıf elle oluşturulur. Daha sonra tablodaki datanın yönetimi için update, insert, delete, select sorguları hazırlanır. ORM geliştirici yaptığı bütün bu işlemleri arka planda otomatik olarak veri erişimini OOP mantığında basitleştirerek hızlandırır.

ORM (object relational mapping), ilişkisel veri tabanındaki objelerimizi nesneye yönelik programlar ile yönetip kullanmak için kod tarafında bir nesne olarak modellenmesi tekniğidir. OOP mantığına uygun bir şekilde veri tabanında bulunan tablolar için birer nesne örneklerini oluşturulur. Veri tabanı içerisinde bulunan tablolar nesne, tablo kolonları ise özellik olarak kod tarafında tanımlanır.

ADO.NET Entity Framework ile veri tabanımızın nesne modeli ve veri tabanında gerçekleştirilen CRUD "Create, Read, Update, Delete" sorguları arka planda otomatik olarak oluşturulur ve sorgular otomatik olarak SQL sorgularına dönüştürülür.

Kod içinde veri tabanın bir modeli Entity Data Model (EDM) ile oluşturulur. Bu model kod içinde EDMX uzantılı xml tabanlı bir dosyaya yazılır. Bu dosya çalışma zamanında "Conceptual Schema Defination Language (cdsl), Mapping Specification Language (msl), Store Schema Defination Language (ssdl)" üç farklı dosyaya bölünmektedir. Bu dosyaların hepsi otomatik olarak oluşmaktadır ve kod içine gömülüdür.

ADO.Net Entity Framework ilk çıktığında sadece veri tabanından nesne modeli oluşturuyordu. Visual Studio 2010 ile artık boş veri tabanı üzerinden nesne modeli oluşturulmasına izin vermektedir ve bu modelden veri tabanı oluşturulmasını sağlamaktadır.

### **1.4.Microsoft Access Veritabanı**

Access , İlişkisel Veri Tabanı Yönetim Sistemi ile çalışan bir veri tabanı oluşturma programıdır. İlişkisel Veri Tabanı Yönetim Sistemi sisteminde bir veri tabanı dosyasında birden fazla tablo oluşturulabilir ve bu tablolar arasında birbirleriyle ilişki kurulabilir. Kurulan ilişkiler  sayesinde farklı tablolardaki veriler sanki aynı tablodaymış gibi kullanılabilir.

Microsoft Access bir İlişkisel Veri Tabanı Yönetim Sistemi uygulamasıdır. Bir veri tabanını oluşturmak ve kullanmak Access ile diğer veri tabanı uygulamalarına göre çok daha kolaydır. Bunun nedeni Access’in, Windows ortamının Grafiksel Kullanıcı Arabiriminin sağladığı avantajların tümünden yararlanma imkanı vermesidir. Grafiksel Kullanıcı Arabirimi, karmaşık komut dizilerini öğrenmeyi gerektirmeden, ekran üzerindeki nesneler ve simgeler yardımıyla, fare desteğinden de yararlanarak kullanıcının çalışmasına olanak verir. Örneğin, geleneksel veri tabanı uygulamalarında iki tablo arasında bağlantı kurmak için oldukça karmaşık komut dizileri yazmak gerekirken, Access’te bu iş basit bir fare hareketiyle gerçekleştirilebilir.

## 2-Uygulama Mimarisi

### **2.1.Veritabanı Katmanı (Data Access Layer)**



**Şekil 2-Veritabanı tabloları**

Kütüphane Otomasyonu uygulaması için hazırlanan veritabanında üç adet tablo mevcuttur. Bunlar; almaerme, kitaplar ve öğrencilerdir. Tablolar şekil 2’de gösterilmiştir.



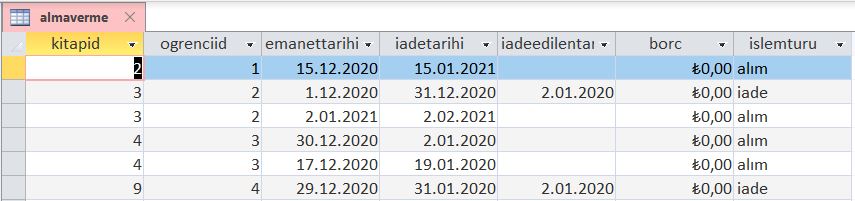
**Şekil 3-Öğrenciler Tablosu**

Şekil 3’te öğrenciler tablosu verilmiştir. Öğrenciler tablosunda ogrenciid, isim, numara, bolum, email, adres alan adlarıdır. Bu alanlarda öğrenci bilgileri tutulmaktadır. Öğrenci tablosu için ogrenciid primary key tanımlanmıştır yani kayıtların tekrarı engellenmiştir.



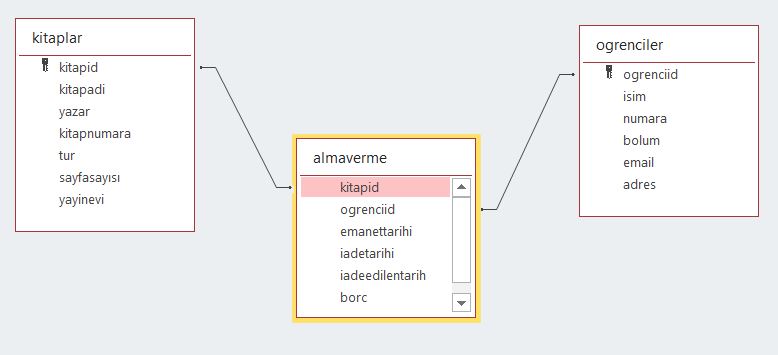
**Şekil 4-Kitaplar tablosu**

Şekil 4’te kitaplar tablosu vardır. Kitaplarla ilgili kitapid, kitapadi, yazar, kitapnumara, tur, sayfasayısı, yayinevi gibi alan adları bulunmaktadır. Kitapid tekrarı önlemek için primary key yani benzersiz tanımlanmıştır.



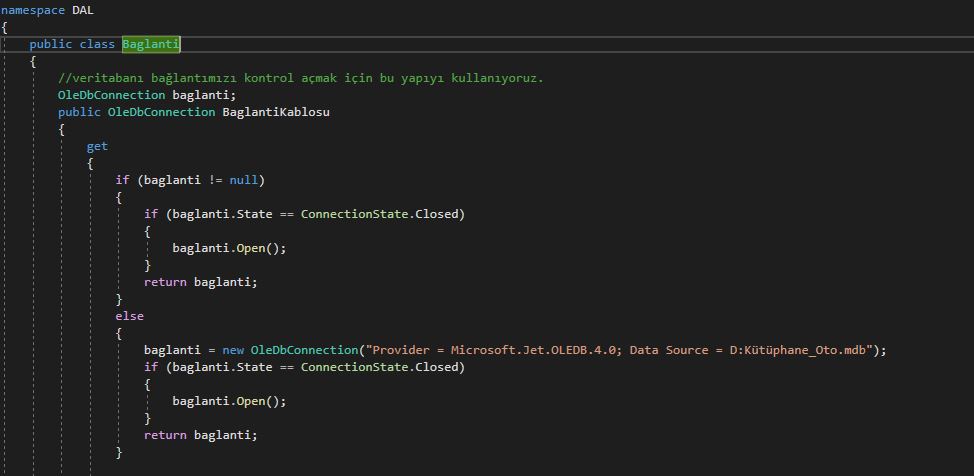
**Şekil 5-Alma Verme tablosu**

Şekil 5’te almaverme tablosu vardır. Öğrenci ve kitapların emanet veya iade işlemlerini göstermektedir.



**Şekil *6*- Veritabanı İlişkileri**

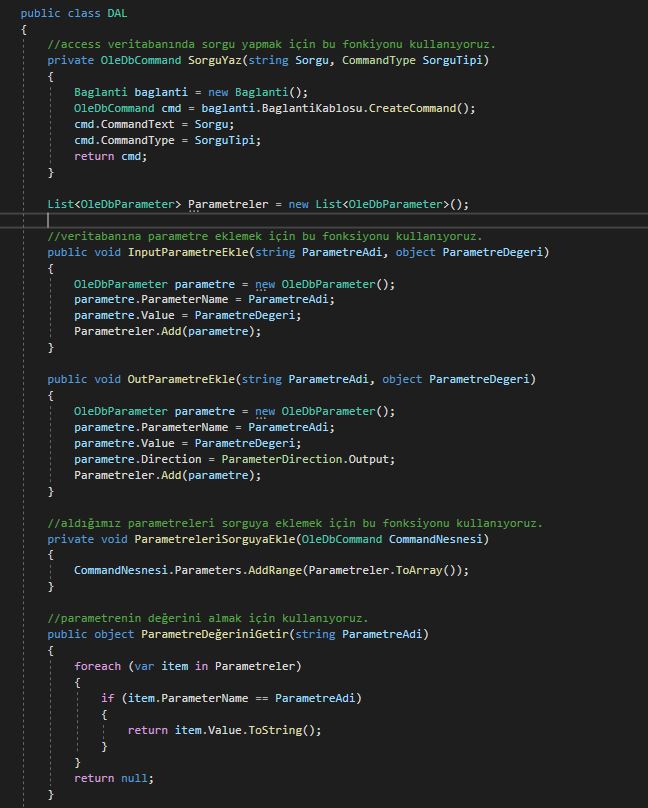
Şekil 6 da Veritabanı ilişkileri verilmiştir. Böylece veritabanında yapılacak değişiklikler için bilgi tutarlılığı sağlanacaktır.



**Şekil *7*-DAL Bağlantı Class’ı**

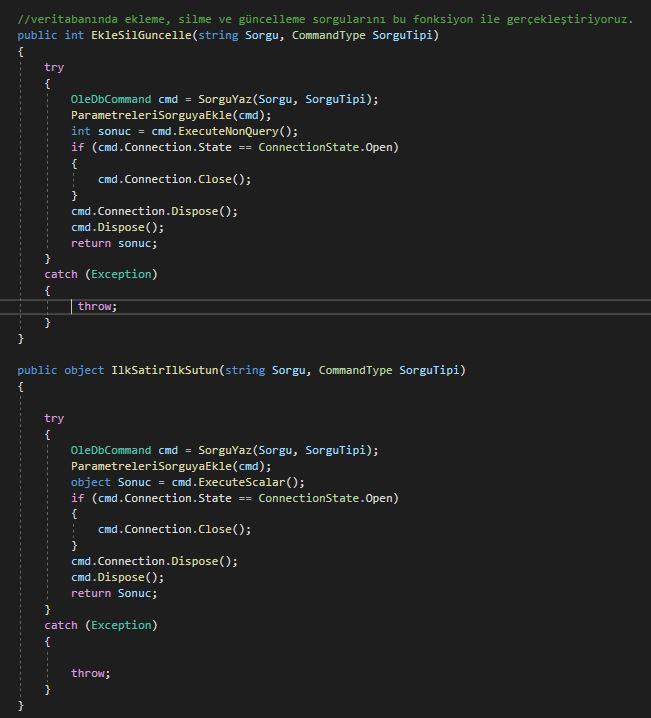
Veritabanı hazırlandıktan sonra Data Access layer için class library üzerinden çalışılır. Data Access layer da her bir tablo için ayrı bir class oluşturulur. Bunun dışın bu katmanda veritabanı ile bağlantıyı sağlatacak bir class tanımlanmalıdır. Bu katman için entity katmanı başvuralara eklenir. Bağlantının sağlanacağı bağlantı Class’ı Şekil 7 de verilmiştir.

.



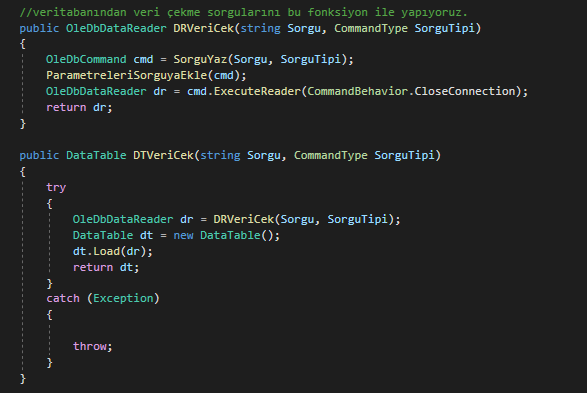
**Şekil 8-DAL DAL Class’ı**

Veritabanında yapılacak işlemler için Dal Class’ı tanımlanmıştır. Şekil 8 de görüldüğü gibi Veritabanında Sorgulama yapmak , parametre eklemek , parametreleri sorguya eklemek, onların değerlerini almak için kullanılacaktır.



**Şekil 9-DAL DAL Class’ı**

Şekil 9 da aynı Dal sınıfı içerisindeki veritabanına ekle sil güncelle fonksiyonları gerçekleştirilmiştir.

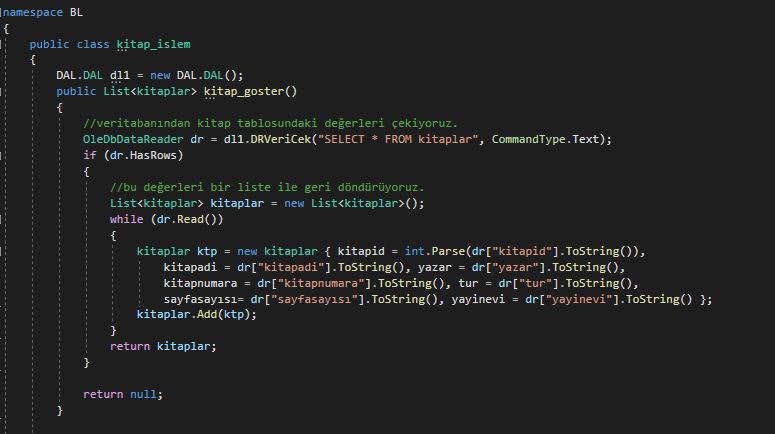


**Şekil 10-DAL DAL Class’ı**

Son olarak Dal Class’ı içerisinde veritabanındaki veri çekme sorguları için bir fonksiyon tanımlıyoruz ve DAL alanındaki kod işlemleri bitiyor.

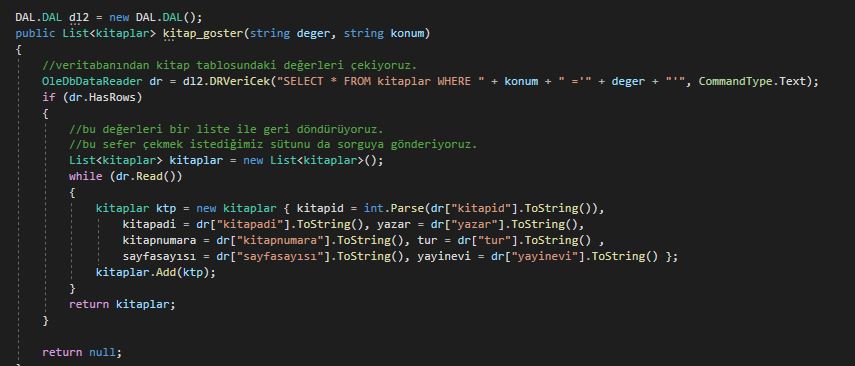
### **2.2.İş Katmanı (Business Logic Layer)**

Business logic layer da veri katmanında olduğu gibi classlibrary şeklinde kullanılır. Burada isimlendirme tekil sorumluluk mekanizmasına göre yapılır. Burada mantıksal işlerin yapılacağı classlar yer alır. Data Access Layer’da olduğu gibi her tablo için bir class oluşturulur ve oluşturulan classlarda mantıksal işlemlerin yapıldığı metodlar yer almaktadır. Bu katman için entity layer ve data Access layer katmanları başvurulara eklenmelidir. Bunu daha iyi anlayabilmek için uygulama kapsamında bulunan aşağıda verilmiş olan Business Logic Layer classlarını inceleyelim.



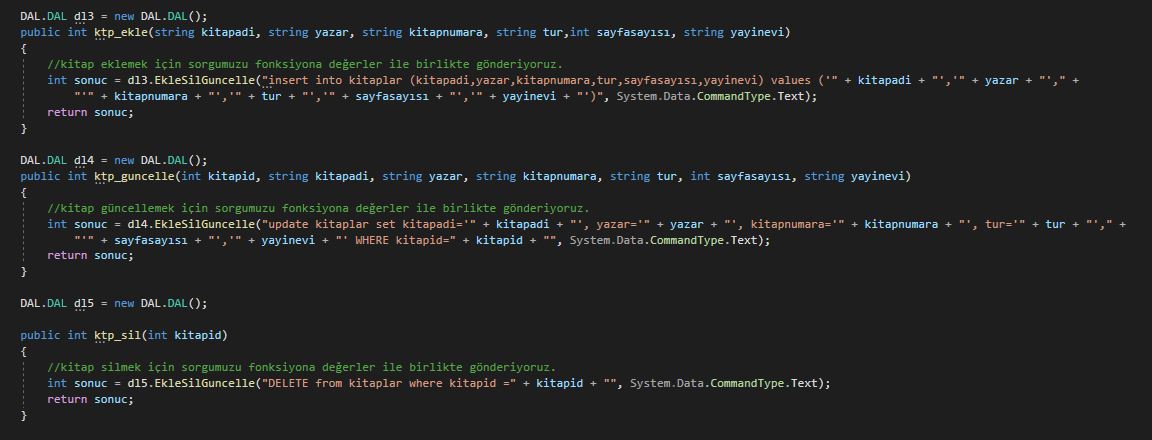
**Şekil 10-BL Kitap işlem Class’ı**

Şekil 10 da görüldüğü gibi kitapları veri tabanından çekiyoruz ve çektiğimiz değerleri liste ile geri döndürüyoruz.



**Şekil 11-BL Kitap işlem Class’ı**

Şekil 11 de şekil 10 da ki işlemler yapılır ve onlara ek olarak çekilmek istenen sütun da sorguya gönderilir.



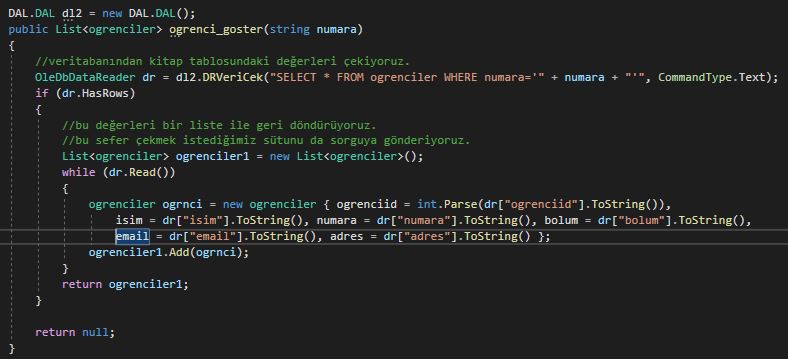
**Şekil 12-BL Kitap işlem Class’ı**

Şekil 12 de kitap ekleme, güncelleme ve silme işlemleri yapılmaktadır. Silme ve güncelleme alanlarında id bilgisi referans alınmaktadır.



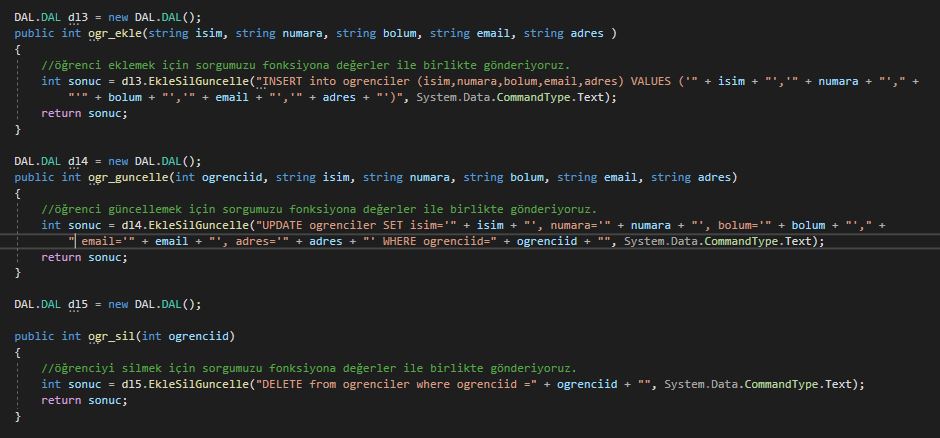
**Şekil 13-BL Öğrenci işlem Class’ı**

Şekil 13 de görüldüğü gibi Öğrencileri veri tabanından çekiyoruz ve çektiğimiz değerleri liste ile geri döndürüyoruz.



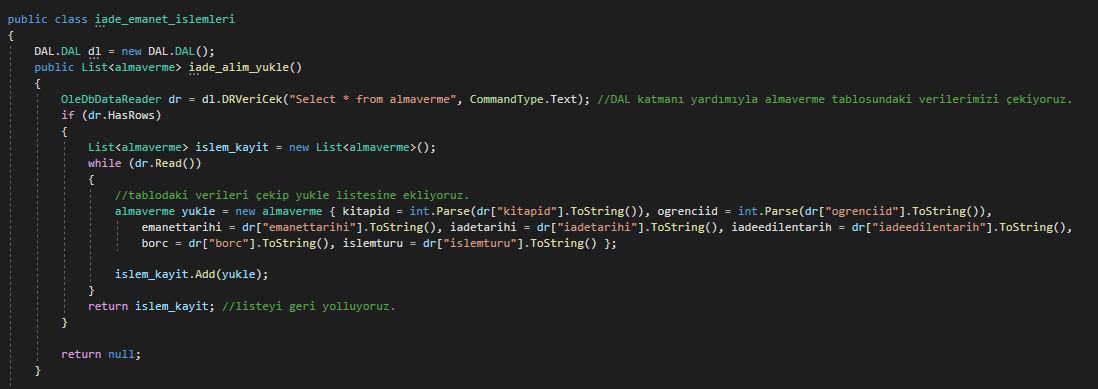
**Şekil 14-BL Öğrenci işlem Class’ı**

Şekil 14’te şekil 13’te ki işlemler yapılır ve onlara ek olarak çekilmek istenen sütun da sorguya gönderilir.



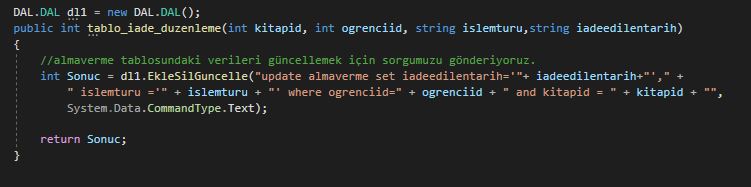
**Şekil 15-BL Öğrenci işlem Class’ı**

Şekil 15’te öğrenci ekleme, güncelleme ve silme işlemleri yapılmaktadır. Silme ve güncelleme alanlarında id bilgisi referans alınmaktadır.

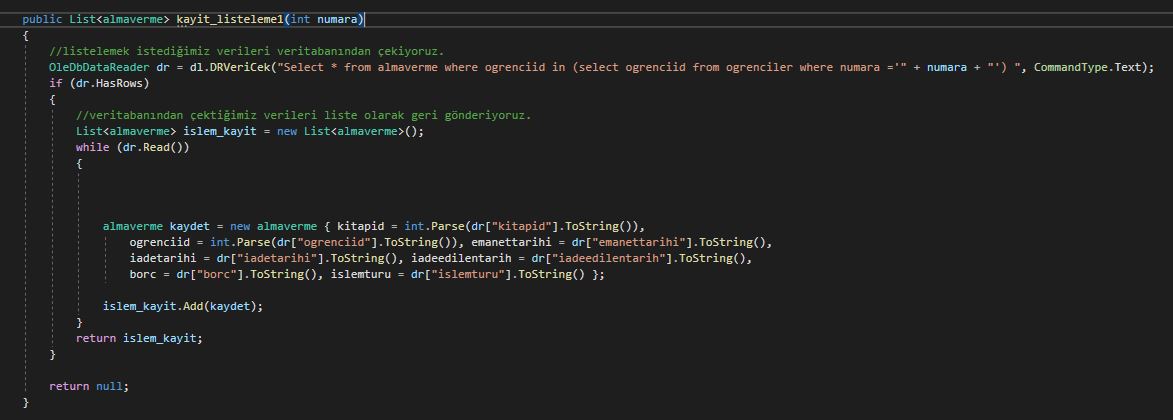


**Şekil 16-BL İade Emanet işlem Class’ı**

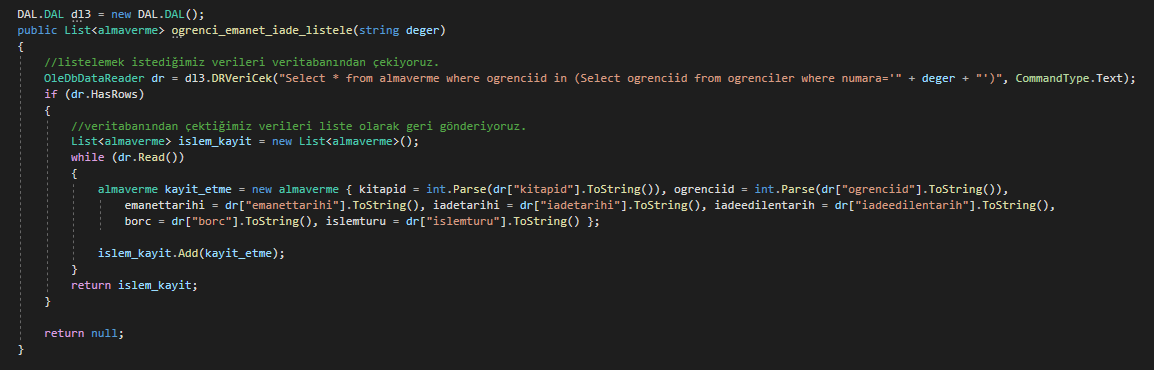
Şekil 16 da almaverme tablosundan veriler çekilir, listeye eklenir sonra da liste geri yollanır.



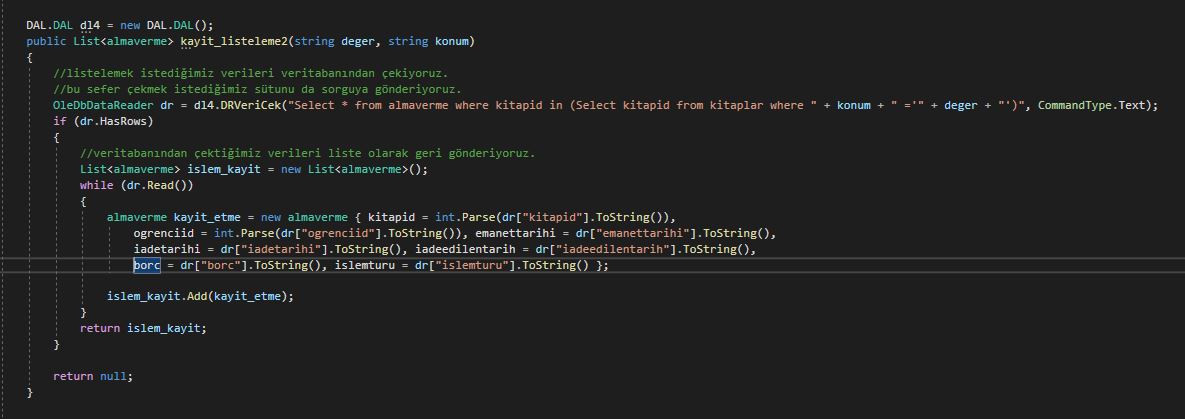
**Şekil 17-BL İade Emanet işlem Güncelleme Class’ı**



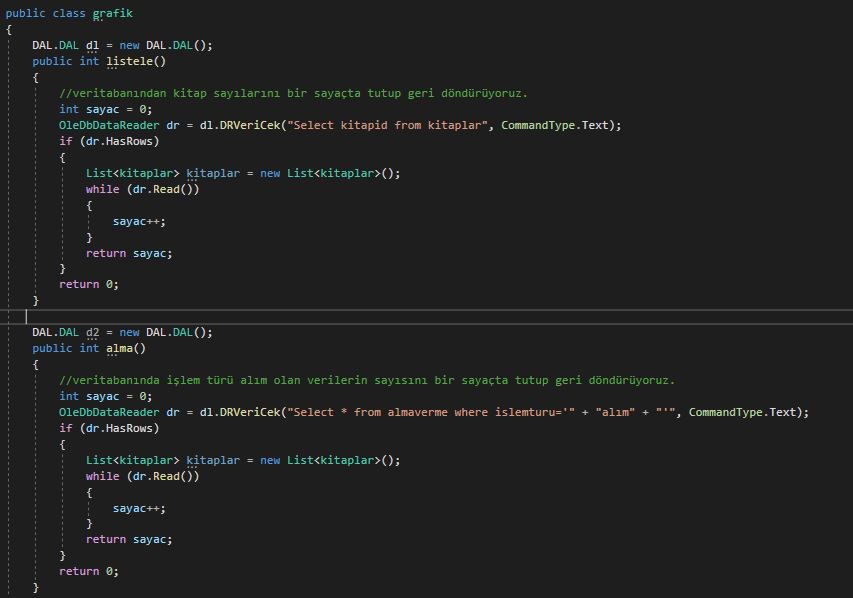
**Şekil 18-BL İade Emanet işlem Listeleme Class’ı**



**Şekil 19-BL İade Emanet işlem Emanet verme Listeleme Class’ı**



**Şekil 20-BL İade Emanet işlem Emanet verme Listeleme Class’ı**



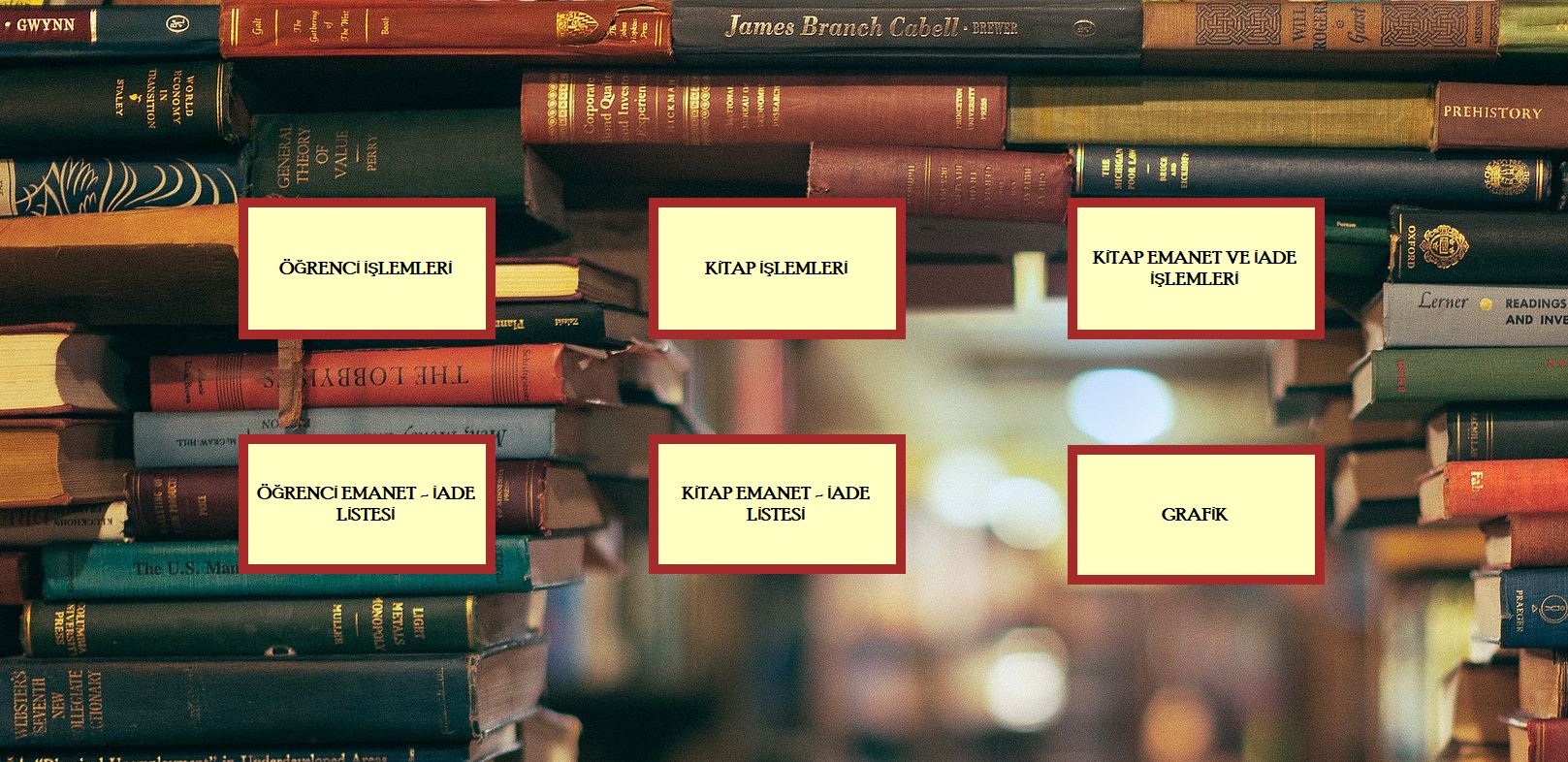
**Şekil 21-BL Grafik Class’ı**

Şekil 21 de yani grafik işlemlerinde veritabanından kitap sayılarını bir sayaçta tutup geri döndürüyoruz. Sonra da veritabanından kitap sayılarını bir sayaçta tutup geri döndürüyoruz.

### **2.3.Sunum Katmanı (Presentation Layer)**

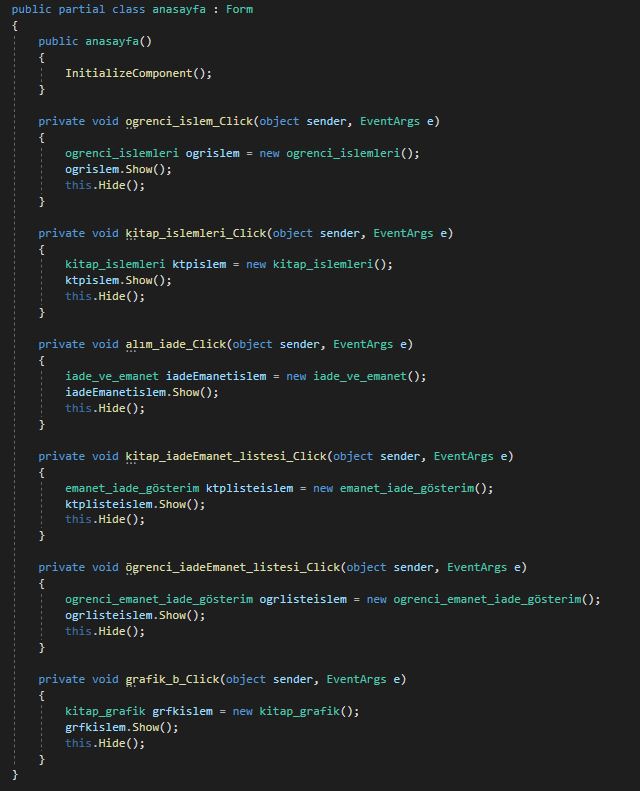
Uygulama kapsamında geliştirilen sunum katmanı projesinde Windows form application tercih edilmiştir. Bu katman için entity layer ve business logic layer katmanı başvurulara eklenmelidir. Uygulama için geliştirilmiş olan Ara yüzleri inceleyelim.

Uygulama için Öğrenci işlemleri, kitap işlemleri, kitap emanet ve iade işlemleri, öğrenci emanet iade listesi, kitap emanet listesi, grafik gibi yönlendirmelerin bulunduğu Anasayfa bulunmaktadır. Aşağıda tasarımı verilmiştir.



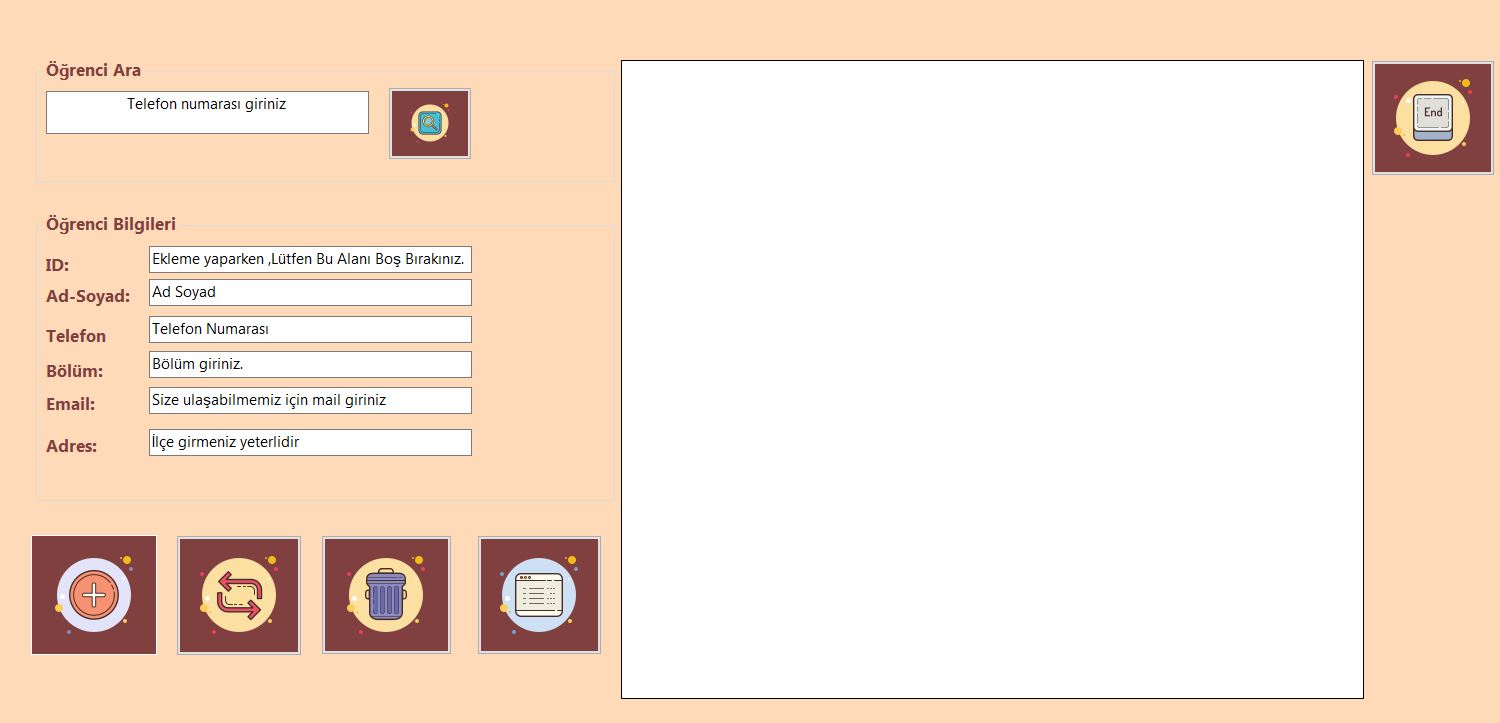
**Şekil 22- Anasayfa**

Anasayfa form tasarımı şekil 22’te gösterildiği gibidir. Anasayfa için gerekli kodlamaların yapıldığı anasayfa.cs sayfası aşağıda bulunan şekil 23’te verilmiştir.



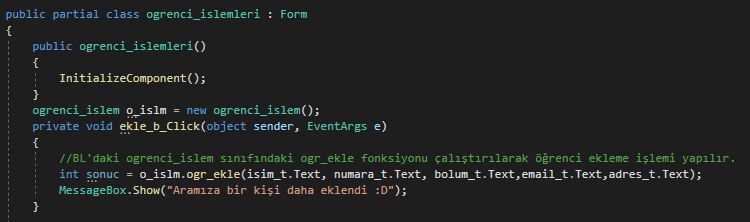
**Şekil 23- Anasayfa Form**

Anasayfa üzerinde Öğrenci İşlemleri butonuna tıklandığında Aşağıdaki ekran açılacaktır.

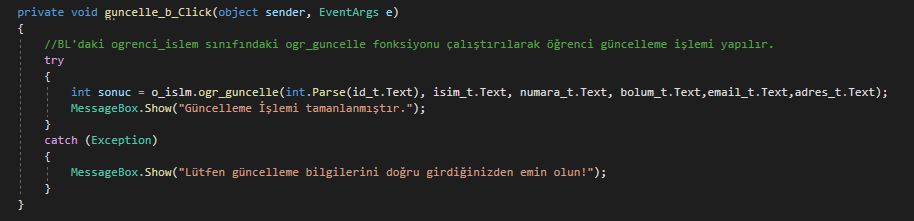


**Şekil 24- Öğrenci İşlemleri**

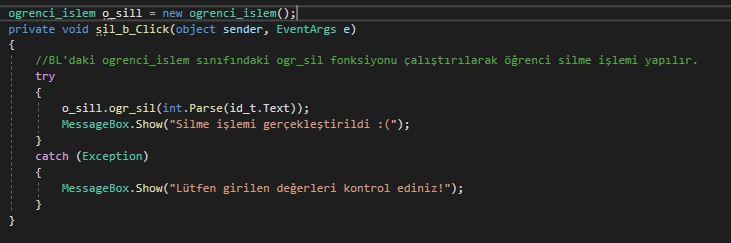
Şekil 24’te verilmiş olanlara gerekli bilgiler girildiğinde ekranda var olan ekle, listele, güncelle ve sil komutları çalışacaktır. Böylece yeni öğrenci kayıtları yapılabilecek, var olan öğrenciler görüntülenebilecek, olası bir hata durumunda ögrenciid ile kayıtlar güncellenebilecek ve aynı şekilde ögrenciid ile istenilmeyen kayıtlar silinebilecektir. Öğrenci işlemlerinin formu için yapılmış olan kodlamaları aşağıda verilmiştir.



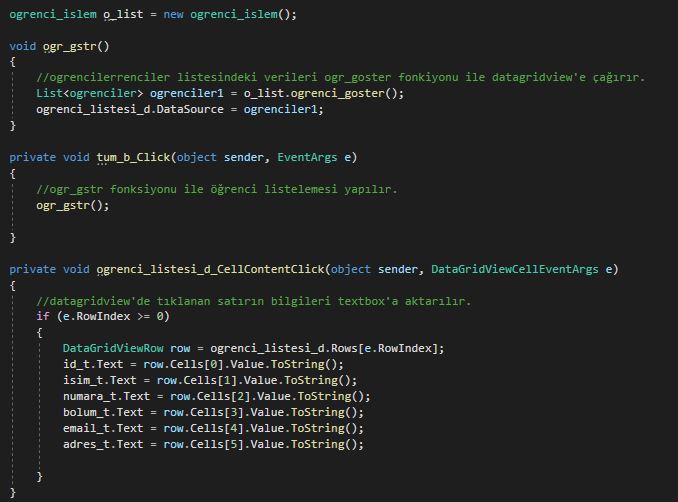
**Şekil 25- Öğrenci Ekleme Butonu**



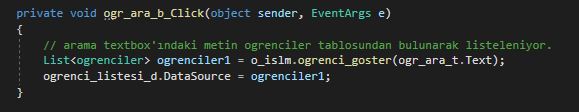
**Şekil 26- Öğrenci Güncelleme Butonu**



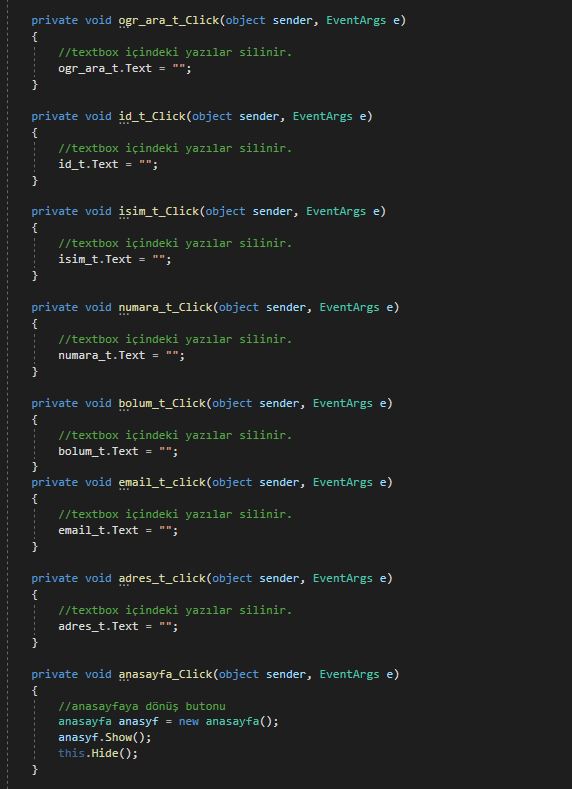
**Şekil 27- Öğrenci Silme Butonu**



**Şekil 28- Öğrenci Listeleme İşlemleri**



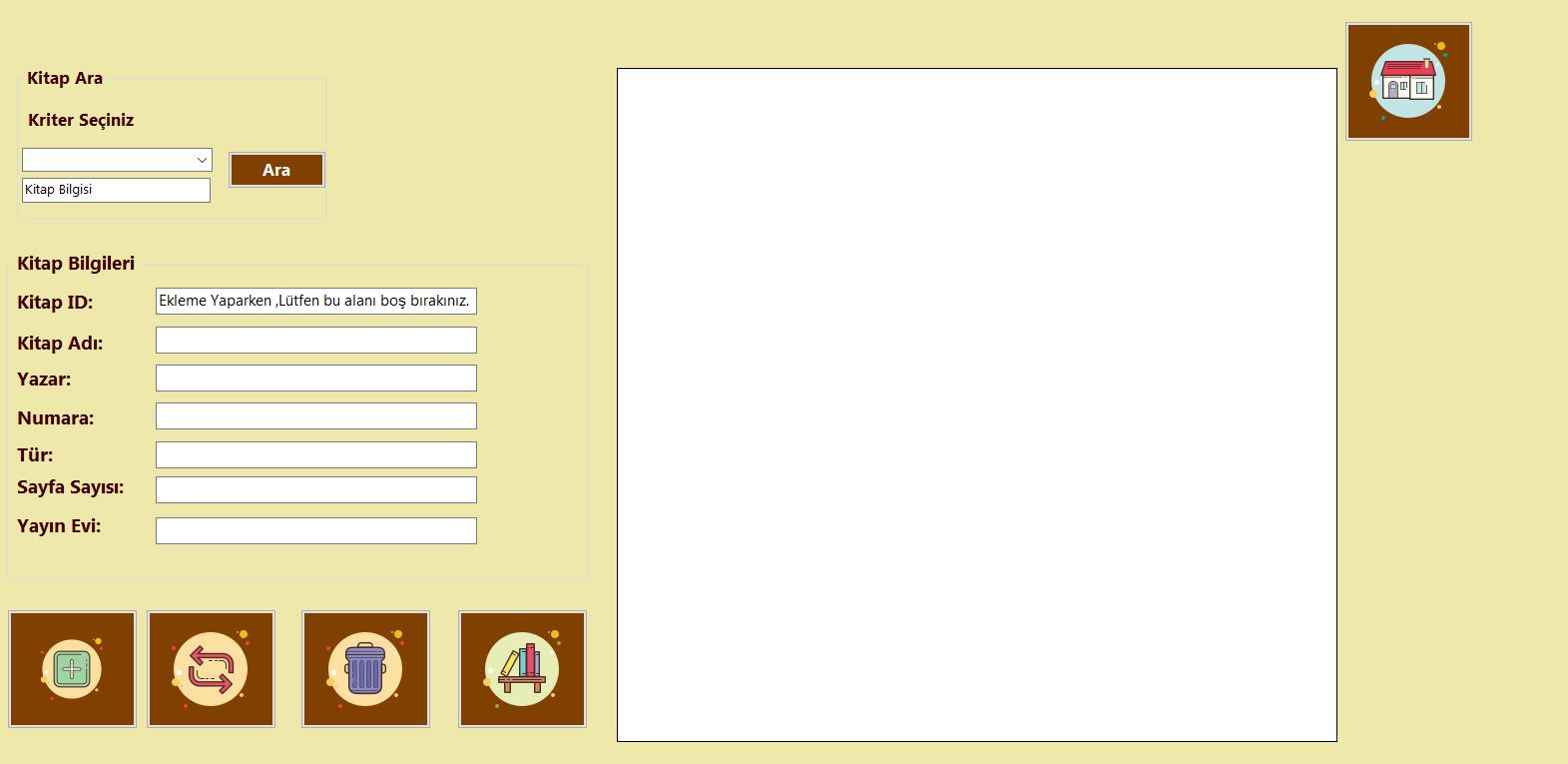
**Şekil 29- Öğrenci Arama Butonu**



**Şekil 30- Öğrenci textbox işlemleri**

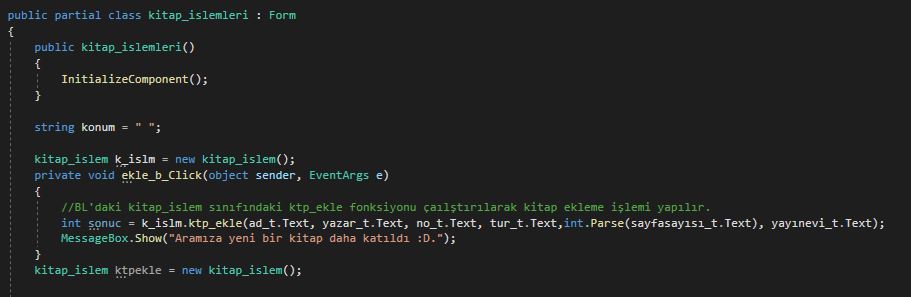
Şekil 30 da textbox a tıklandığı zaman içindeki verilerin silinmesi için gereken kodlar verilmiştir. Ayrıca anasayfaya dönmek içinde buton eklenmiştir.

Anasayfa da kitap işlemleri butonuna tıklandığında ekrana aşağıdaki kitap form tasarımı gelecektir.

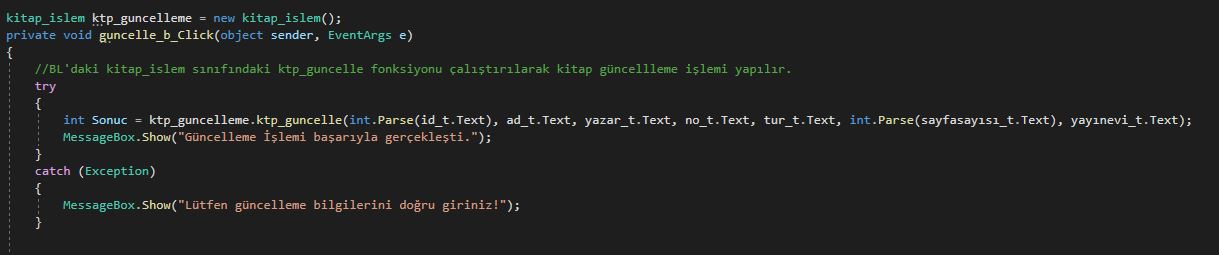


**Şekil 31- Kitap İşlemleri**

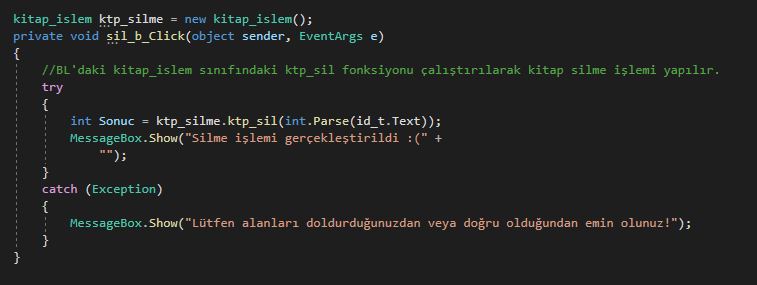
Şekil 31 de verilmiş olanlara gerekli bilgiler girildiğinde ekranda var olan ekle, listele, güncelle ve sil komutları çalışacaktır. Böylece yeni kitap kayıtları yapılabilecek, var olan kitaplar görüntülenebilecek, olası bir hata durumunda kitapid ile kayıtlar güncellenebilecek ve aynı şekilde kitapid ile istenilmeyen kayıtlar silinebilecektir. Kitap işlemlerinin formu için yapılmış olan kodlamaları aşağıda verilmiştir.



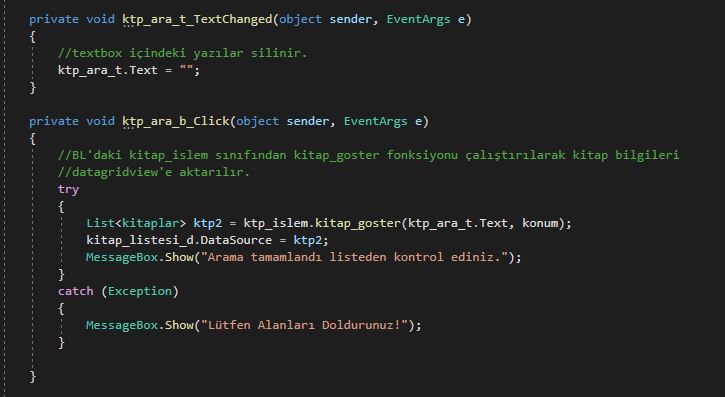
**Şekil 32- Kitap Ekleme Butonu**



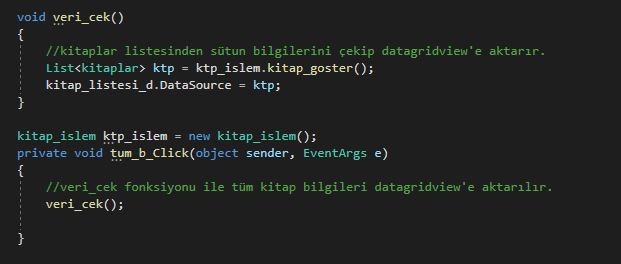
**Şekil 33- Kitap Güncelleme Butonu**



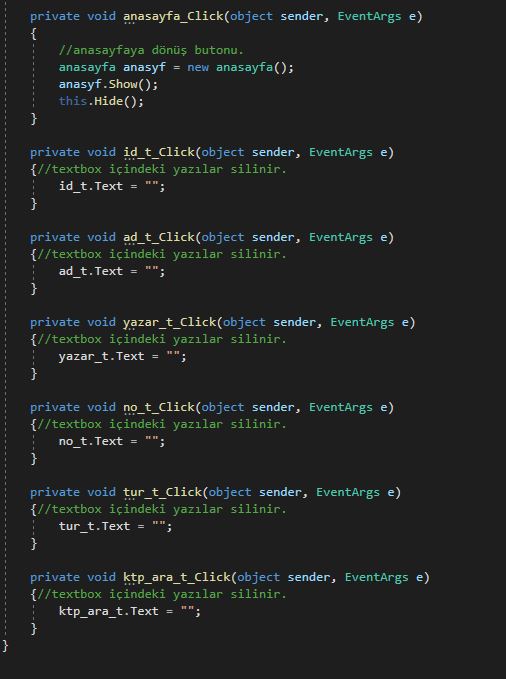
**Şekil 34- Kitap Silme Butonu**



**Şekil 35- Kitap Arama Butonu**



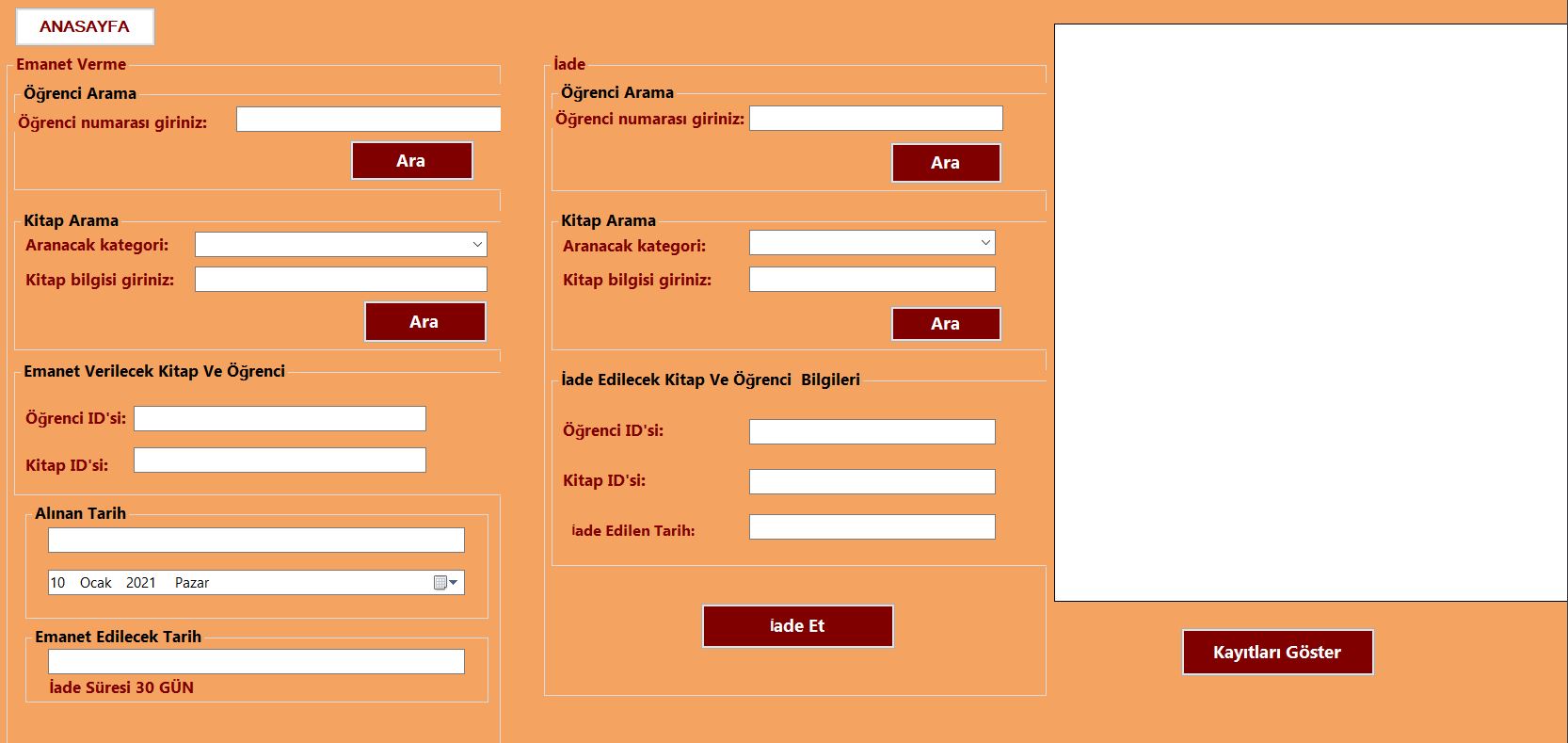
**Şekil 36- Kitap Listeleme Butonu**



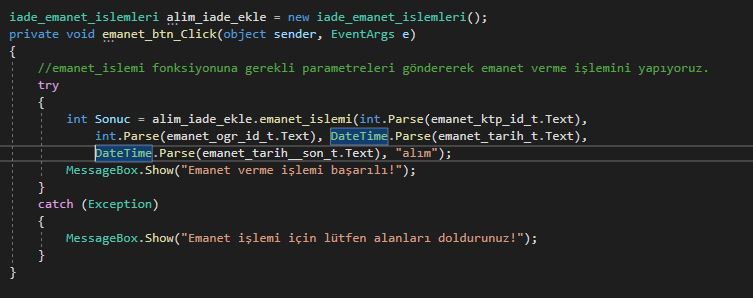
**Şekil 37- Kitap textbox işlemleri**

Şekil 37 de textbox a tıklandığı zaman içindeki verilerin silinmesi için gereken kodlar verilmiştir. Ayrıca anasayfaya dönmek içinde buton eklenmiştir.

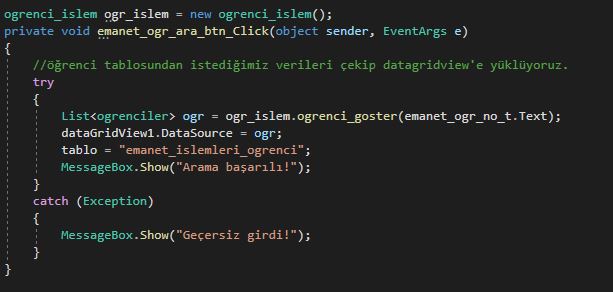
Anasayfa da Kitap emanet ve iade işlemleri butonuna tıklandığında karşıma aşağıdaki form tasarımı çıkacaktır.



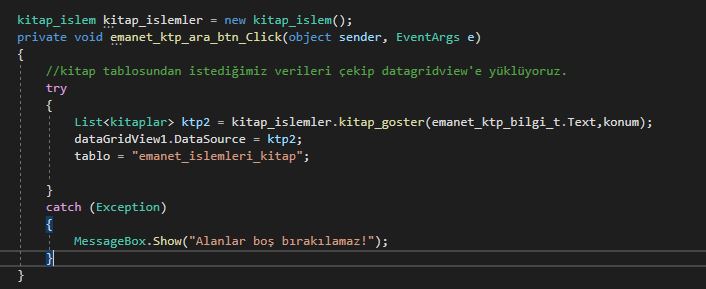
**Şekil 38- Kitap Emanet ve İade İşlemleri**



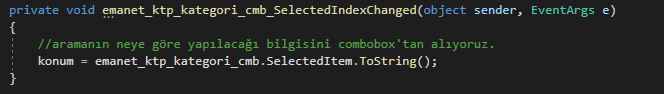
**Şekil 39- Kitap Emanet verme İşlemleri**



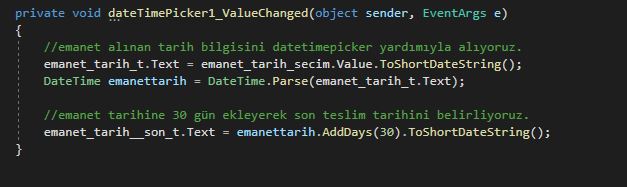
**Şekil 40- Emanet Öğrenci Arama İşlemleri**



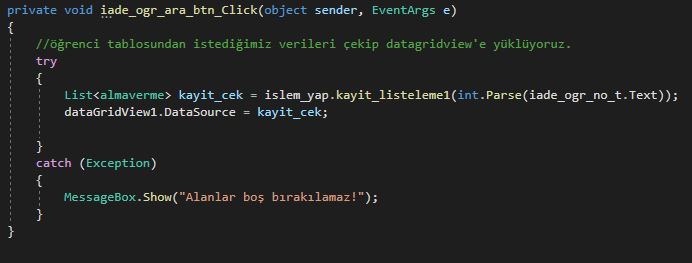
**Şekil 41- Emanet Kitap Arama İşlemleri**



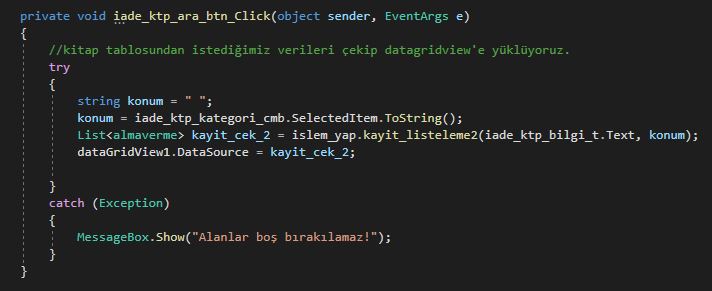
**Şekil 42- Emanet Kitap Kategori Arama İşlemleri**



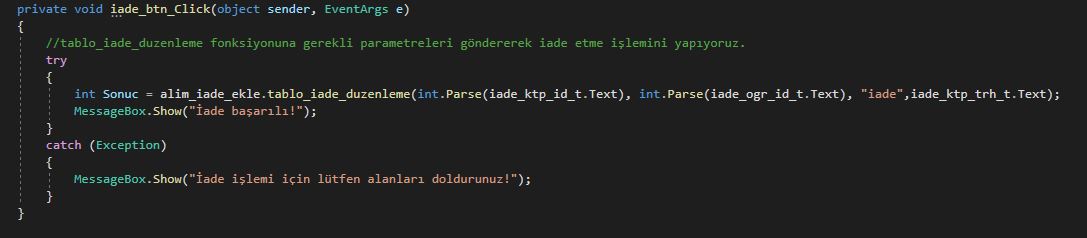
**Şekil 43- Emanet Tarih İşlemleri**



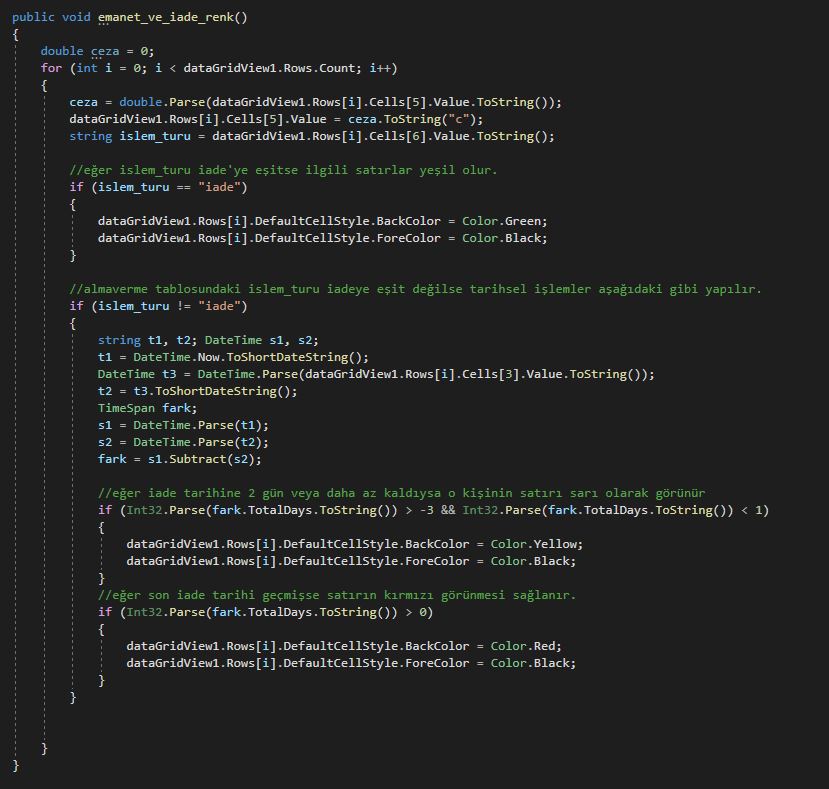
**Şekil 44- İade Öğrenci Arama İşlemleri**



**Şekil 45- İade Kitap Arama İşlemleri**

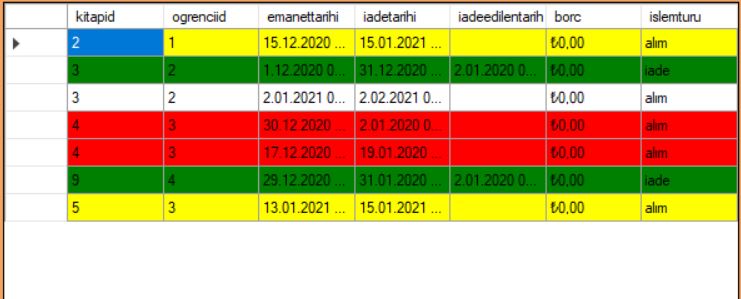


**Şekil 46- İade Verme İşlemleri**



**Şekil 47- İade Renklendirme İşlemleri**

Şekil 47 de görüldüğü gibi eğer öğrenci kitabı getirmesi gereken tarihten geç getirirse satır kırmızı olur. Kitabın emanet süresine 2 gün kaldıysa sarı renk olur. İade işlemi tamamlandığında yeşil olur.



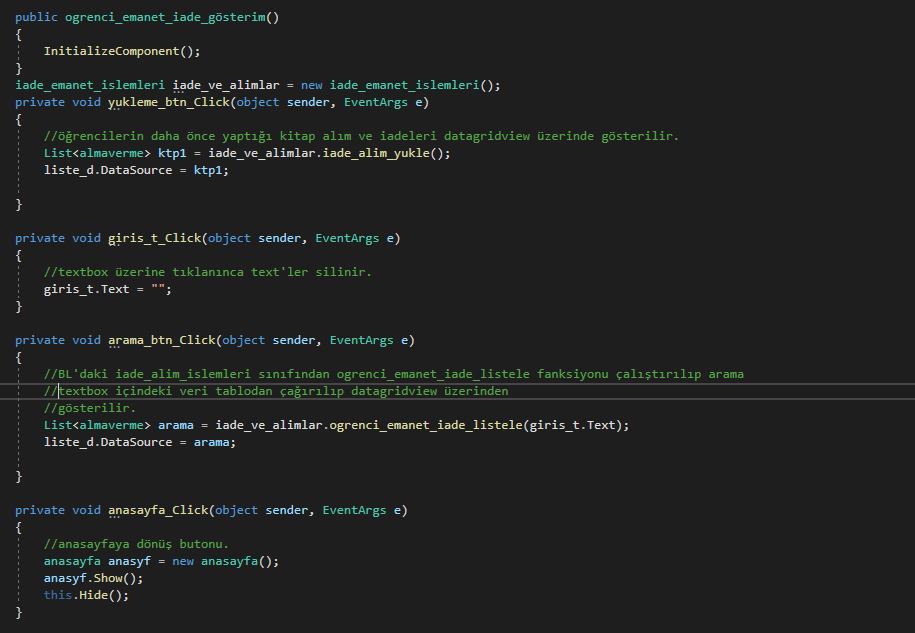
**Şekil 48- İade Renklendirme İşlemleri Görünümü**

Şekil 47 deki kodlara göre tablomuzda şekil 48 gibi bir görünüm oluşacaktır.

Öğrenci Emanet Listesine tıkladığımızda aşağıdaki tasarım sayfası ekrana gelir.

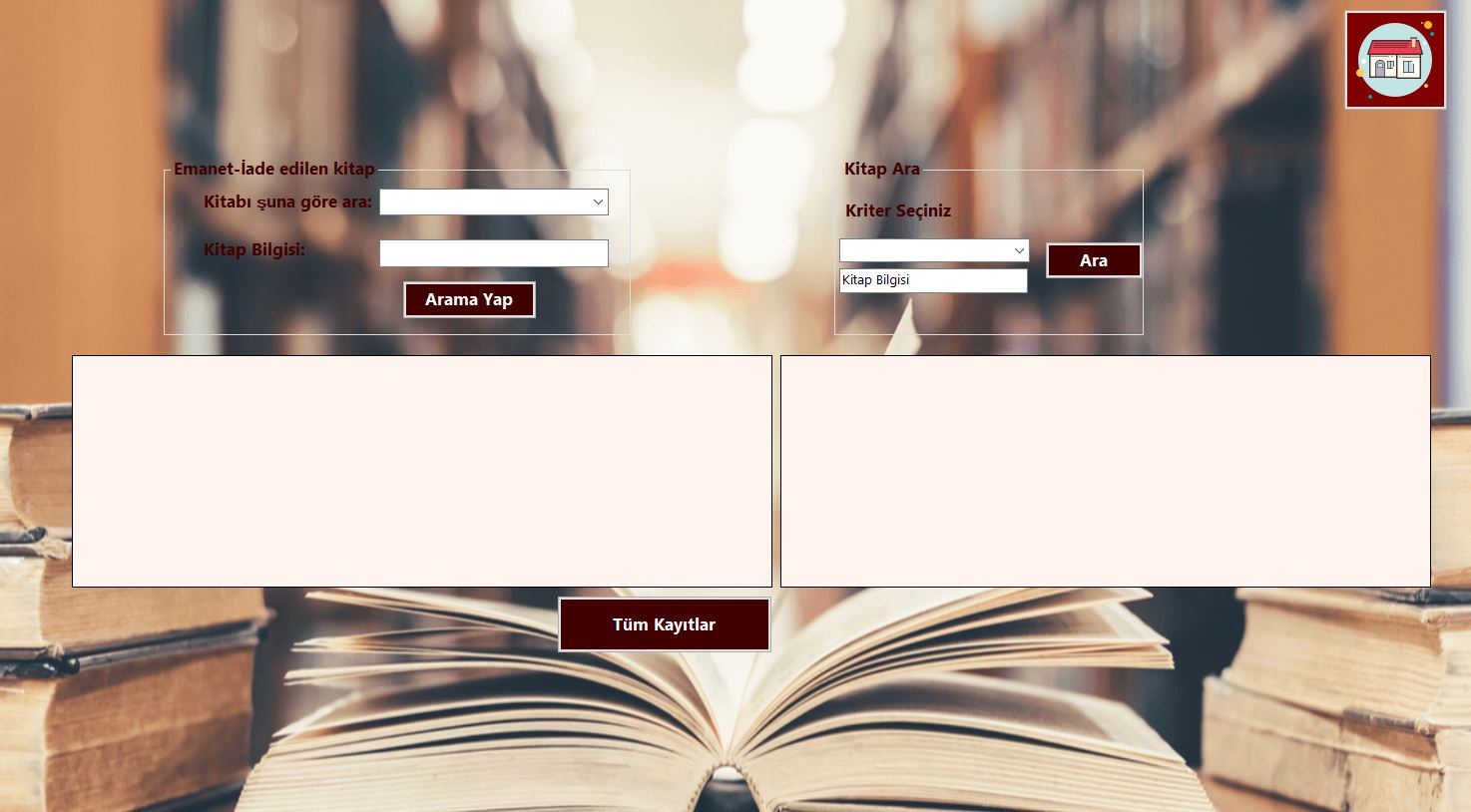


**Şekil 49- Öğrenci Emanet kitap listeleme İşlemleri**



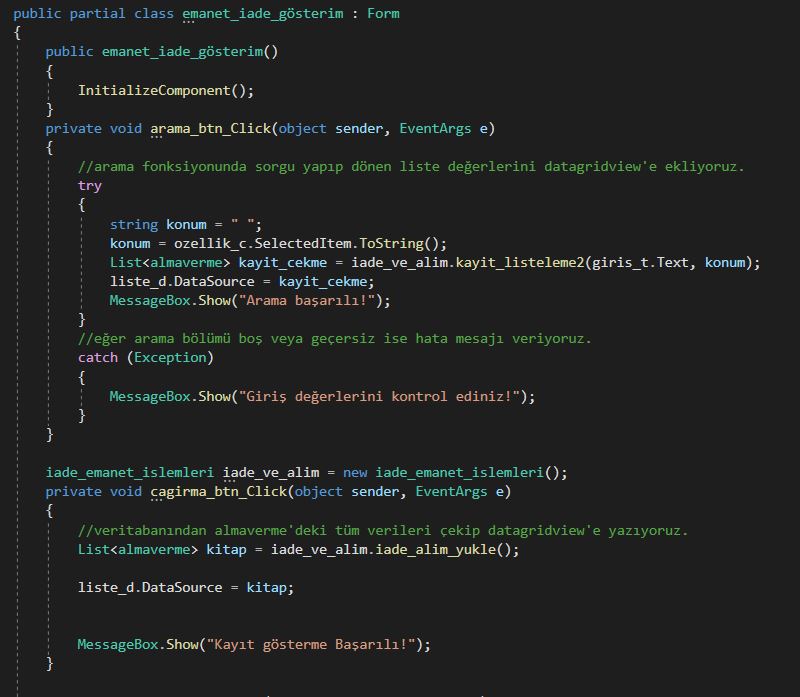
**Şekil 50- Öğrenci Emanet kitap listeleme İşlemleri Formu**

Emanet verilen kitapların listeleneceği form tasarımı aşağıda verilmiştir.

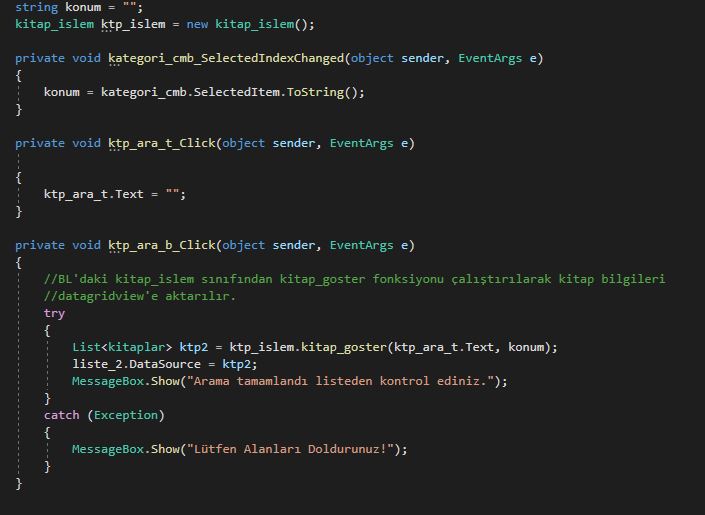


**Şekil 51- Kitap Emanet listeleme İşlemleri**

Şekil 50 de sol tarafta listeleme yapıldığında emanet verilen kitaplar listelenecektir. Sağ tarafta ise girilen kitap kriterine göre detaylı kitap bilgisi verilecektir. Sağ taraftan kitapid’sine bakılarak hangi kitabın emanet verildiği görülebilir.

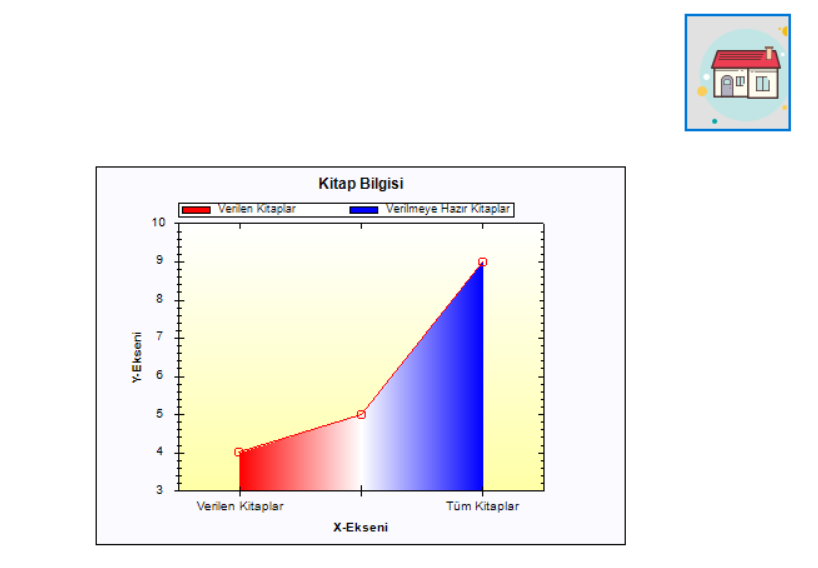


**Şekil 52- Kitap Emanet listeleme İşlemleri Formu**

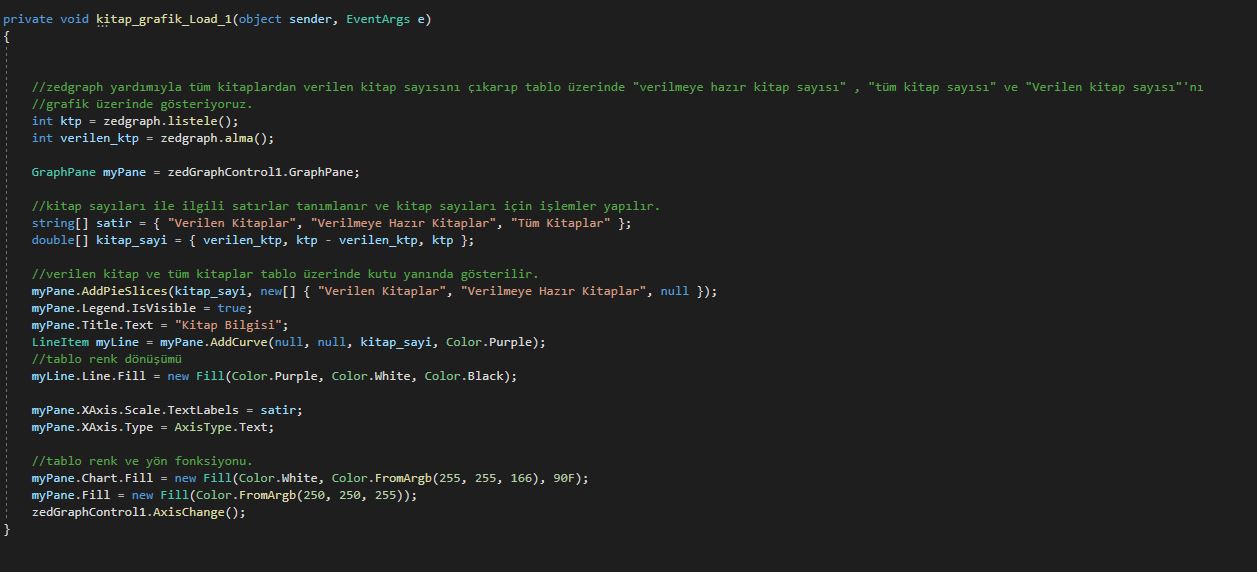


**Şekil 53- Kitap Arama listeleme İşlemleri Formu**

Sunum katmanındaki son form grafik formudur. Grafik form tasarımı aşağıda verilmiştir.



**Şekil 54- Grafik İşlemleri**

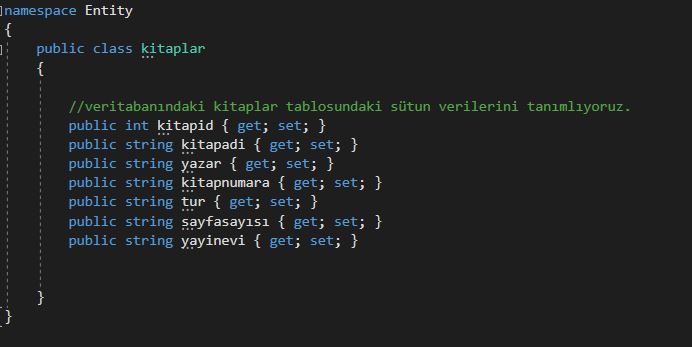


**Şekil 55- Grafik İşlemleri Formu**

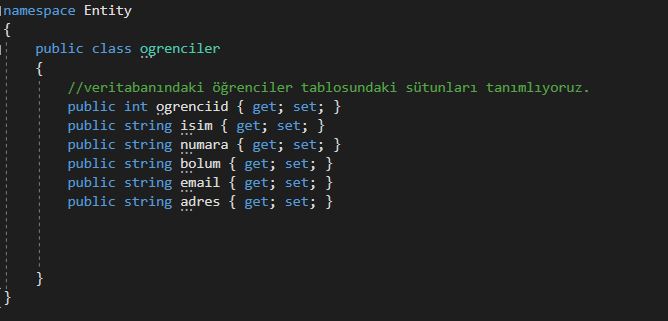
Grafik işlemlerinde zedgraph yardımıyla verilmeye hazır kitap sayıları , tüm kitap sayıları ve verilen kitap sayılarıyla oluşturulan grafik gösteriliyor.

### **2.4.Entity Layer**

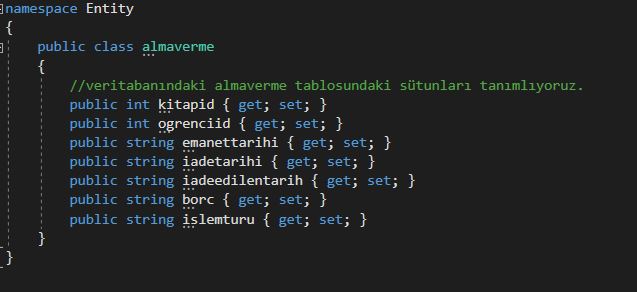
Entity Katmanında tabloda yer alan, alan adları kapsülleme işlemine tabii tutulur ve diğer katmanlar için entity layer başvurulara eklenir. Entity layerda da diğer katmanlarda olduğu gibi her bir tablo için ayrı bir class oluşturulur ve entity layer da class library şeklinde tanımlanır. Aşağıda entity layer katmanı için classlar verilmiştir.



**Şekil 56- Entity Kitaplar**



**Şekil 57- Entity Öğrenciler**



**Şekil 58- Entity Alma Verme**

Projenin kod kısmı bu şekilde tamamlanmış oluyor. Projeyi kendi bilgisayarınıza kurmak için Kurulum>Debug>kurulum alanına tıklayıp gerekli kurulum işlemlerini yapabilirsiniz.