**KÜTÜPHANE OTOMASYONU PROJESİ**

1. **Projenin Amacı ve İşlevselliği**

Kütüphane otomasyon sisteminin amacı, kütüphanedeki tüm kitap, üye ve ödünç alma işlemlerini dijital ortamda yönetmektir. Bu, kütüphanecilere, üyelerine ve yöneticilere zaman kazandıracak ve veri kayıplarını en aza indirecektir. Sistemin temel işlevleri şunlar olacaktır.

* Kitapların eklenmesi, güncellenmesi ve silinmesi
* Üyelerin kaydı, güncellenmesi ve silinmesi
* Kitap ödünç alma ve iade işlemleri
* Kitapların stok takibi
* Kitap ve üye verilerinin raporlanması

Bu sistemin kullanıcılara sunduğu temel faydalar şunlar olacaktır:

* Kütüphanede bulunan kitapların düzenli takibi
* Üyelerin ödünç alma işlemlerinin izlenmesi
* Hızlı veri erişimi ve işlemler
* Kütüphane yönetiminden daha verimli bir iş akışı

1. **Projede Kullanılacak Teknolojiler**:

MySQL, C#

1. **Projenin Modülleri (Form Sayfaları)**

Projenin temel modülleri veya form sayfaları aşağıdaki gibi olabilir:

1. **Ana Menü Formu**

* Sistemin başlangıç noktasıdır.
* Kullanıcı, diğer modüllere buradan erişir.
* "Kitaplar", "Üyeler", "Ödünç İşlemleri" gibi modüllere yönlendiren butonlar içerir.

1. **Kitaplar Formu**

* Kitap ekleme, düzenleme ve silme işlemleri yapılabilir.
* Kitap adı, yazar, yayınevi, yayın yılı, kategori gibi bilgiler girilebilir.
* Kitaplar, bir liste veya tablo şeklinde görüntülenebilir.
* Arama fonksiyonu ile kitaplara hızlı erişim sağlanabilir.

1. **Üyeler Formu**

* Üye kaydı yapılabilir, mevcut üyeler listelenebilir.
* Üye adı, soyadı, adres, telefon numarası, e-posta adresi gibi bilgilerin tutulması sağlanabilir.
* Üye silme, güncelleme işlemleri yapılabilir.

1. **Ödünç İşlemleri Formu**

* Kitap ödünç alma ve iade işlemleri gerçekleştirilir.
* Ödünç alınan kitap ve üye bilgileri kaydedilir.
* İade tarihleri kontrol edilerek geç ödeme durumunda uyarılar verilir.
* Üyelerin ödünç alma geçmişi izlenebilir.

1. **Raporlar Formu**

* Kitaplar ve üyeler hakkında raporlar oluşturulabilir.
* Ödünç alınan kitapların takibi, iade tarihlerinin geçip geçmediği raporlanabilir.
* Kütüphane yöneticisi için borç takibi, en çok ödünç alınan kitaplar gibi raporlar sunulabilir.

1. **Arama ve Filtreleme**

* Kitaplar ve üyeler üzerinde arama yapılabilir.
* Kriterlere göre filtreleme (örneğin, kategori bazında kitap arama veya üyelerin durumuna göre filtreleme).

1. **Veritabanı Yapısı**

* **Kitaplar Tablosu**: Kitap ID, başlık, yazar, yayınevi, basım yılı, kategori gibi bilgiler içerir.
* **Öğrenciler Tablosu**: Öğrenci ID, adı, soyadı, okul numarası, adres, telefon, e-posta gibi bilgiler içerir.
* **Ödünç Kitaplar Tablosu**: Ödünç alınan kitaplar ile ilgili öğrenci ID, kitap ID, ödünç alınan tarih, iade tarihi gibi veriler yer alır.

1. **Raporlama ve İstatistikler**

* Detaylı raporlama özellikleri eklenebilir. Örneğin, kitap kategorileri bazında popülerlik analizi, üye bazında ödünç alma sıklığı gibi istatistikler oluşturulabilir.

1. **Sonuç ve Değerlendirme**

Kütüphane otomasyonu projesi, kütüphanedeki tüm işlemleri dijitalleştirerek verimliliği artırır ve kütüphane yönetimini kolaylaştırır. Kitapların ve öğrenci bilgilerin doğru bir şekilde yönetilmesini ve ödünç işlemlerinin etkin bir şekilde izlenmesini sağlar.

Bu sistem ile hem kütüphanelerin günlük işleyişini iyileştirmek hem de kullanıcıların daha verimli bir şekilde kütüphaneyi kullanmalarını sağlamak için ideal bir çözüm olacaktır. C# kullanarak geliştirilecek olan bu proje, kullanıcı dostu bir arayüz ile kullanıcıya büyük kolaylık sağlayacaktır.

Bu proje, kütüphanelerin yönetim süreçlerini iyileştirecek ve kullanıcıların deneyimini geliştirecektir. Sistemin gelecekteki gelişmeleri, kullanıcı ihtiyaçlarına göre şekillendirilebilir ve daha fazla özellik eklenebilir.

**KÜTÜPHANE OTOMASYONU PROJESİ ZAMAN PLANLAMASI**

10.03.2025 – 17.03.2025 🡪 Veri tabanı tasarımının oluşturulması

17.03.2025 – 24.03.2025 🡪 Kütüphane Otomasyonu Arayüzünün genel taslağının hazırlanması (Süreç içerisinde eklemeler yapılacak)

24.03.2025 – 31.03.2025 🡪 Ana Menü Formunun oluşturulması. Kitaplar Formunun oluşturulması (Kitap kaydı, listeleme, silme, güncelleme, arama)

Veri tabanı ile c# bağlantısının kurulması

31.03.2025 – 07.04.2025 🡪 Üyeler Formunun oluşturulması. (Üye kaydı, listeleme, silme ve güncelleme)

07.04.2025 – 14.04.2025 🡪 Ödünç İşlemleri Formunun oluşturulması

14.04.2025 – 21.04.2025 🡪 Raporlar Bölümünün oluşturulması

21.04.2025 – 28.04.2025 🡪 Arama Filtreleme Bölümünün oluşturulması

**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

**1. HAFTA**

1. **GİRİŞ:** Bu rapor, kütüphane otomasyonu projesi kapsamında oluşturulacak veri tabanının tasarımı ve uygulanması ile ilgili temel bilgileri içermektedir. Proje, öğrencilerin kitap alıp iade etme süreçlerini takip etmeyi ve kütüphane yönetimini kolaylaştırmayı amaçlamaktadır. Veri tabanı, MySQL kullanılarak tasarlanmıştır.
2. **VERİ TABANI TASARIMI:** Kütüphane otomasyonu veri tabanı, aşağıdaki üç temel tabloyu ve ilgili alan adlarını içermektedir:

* **Kitaplar Tablosu**
  + Kitap ID (Primary Key)
  + Kitap Adı
  + Yazar
  + Yayınevi
  + Sayfa Sayısı
  + Kategori (Kitap Türü)
* **Öğrenciler Tablosu**
  + Öğrenci No (Primary Key)
  + Adı
  + Soyadı
  + Sınıfı
  + Cinsiyet
  + Telefon
* **Ödünç Kitaplar Tablosu**
  + Ödünç ID (Primary Key)
  + Öğrenci No (Foreign Key)
  + Kitap ID (Foreign Key)
  + Ödünç Alınan Tarih
  + İade Tarihi
  + Açıklama

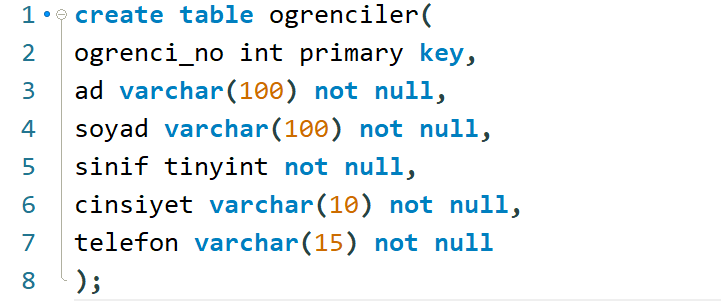
Bu tasarım, kitap ve öğrenci bilgilerinin düzenli saklanmasını ve ödünç alma-iade süreçlerinin takip edilmesini sağlar.

1. **TABLO TANIMLARI VE SQL KODLARI** Aşağıda her bir tablonun MySQL kullanılarak oluşturulmasına dair SQL kodları aşağıdaki gibidir:

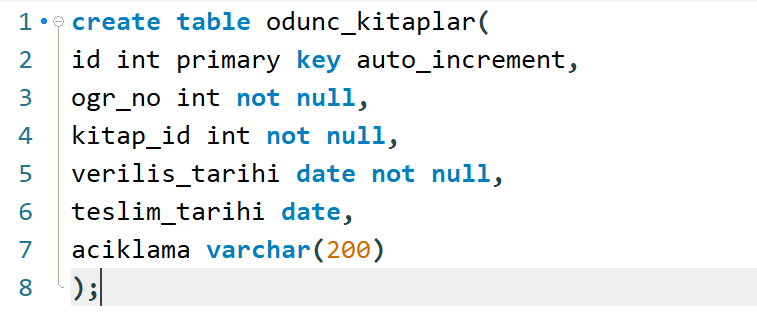
**Kitaplar Tablosu**



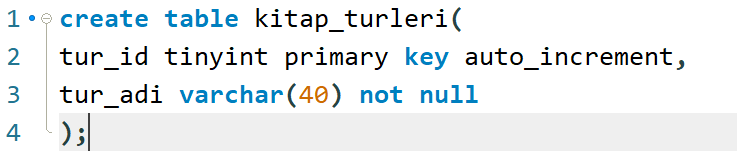
**Öğrenciler Tablosu**



**Ödünç Kitaplar Tablosu**



**Kitap Türleri Tablosu**



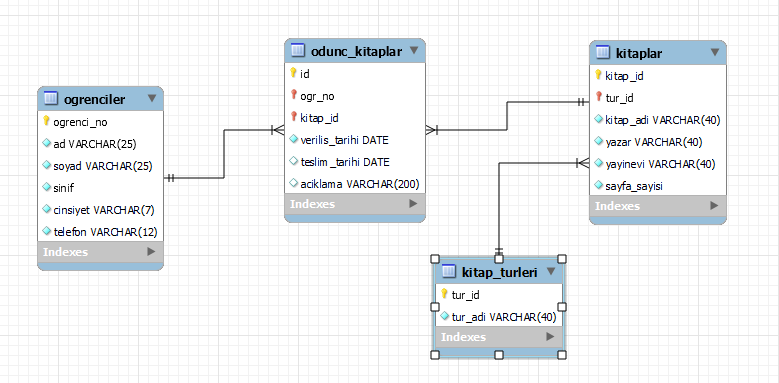
1. **İLİŞİKİSEL VERİ TABANI TASARIMI:**

İlişkisel veri tabanı tasarımına uygun olarak bir MySql EER (Enhanced Entity-Relationship) diyagramı aşağıdaki gibidir.

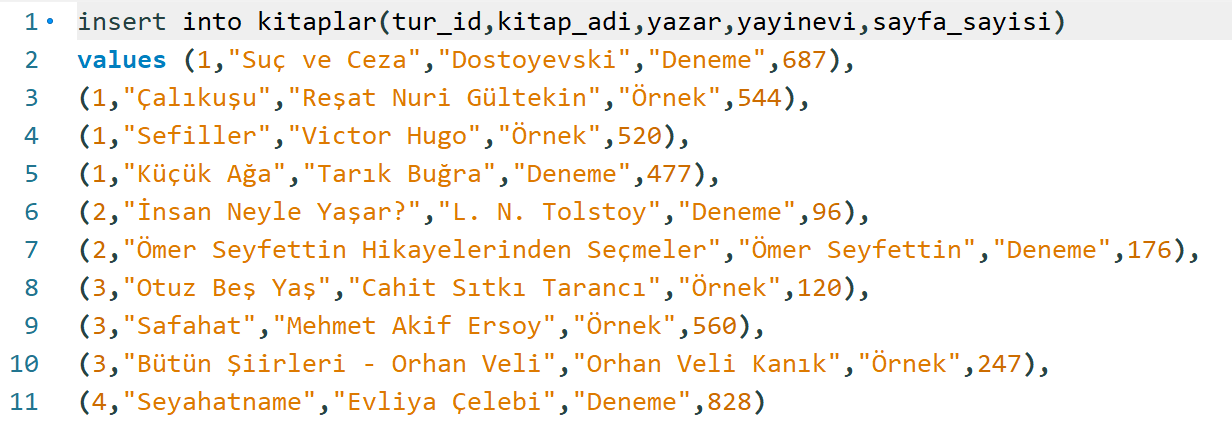
**Kitaplar** ve **Ödünç Kitaplar** tablosu arasında arasında **1-N** ilişkisi vardır. (Bir kitap birden fazla kez ödünç alınabilir).

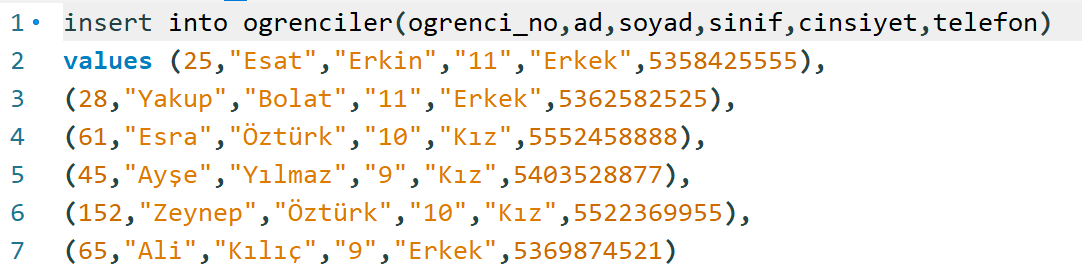
**Öğrenciler** ve **Ödünç Kitaplar** tablosu arasında **1-N** ilişkisi vardır. (Bir öğrenci birden fazla kitap ödünç alabilir).

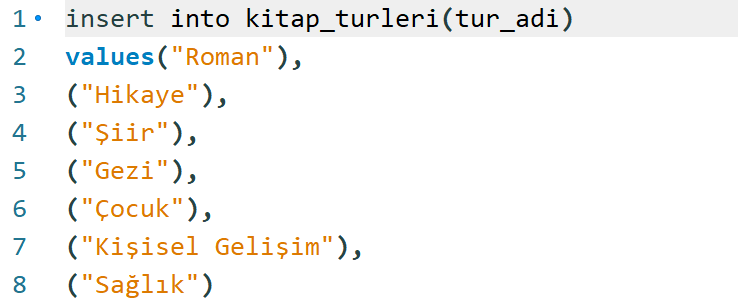
Kitaplar tablosunda kullanılacak tür değerlerini içeren kitap\_turleri tablosu oluşturulmuştur. Kitaplar tablosundaki tur\_id alanına yalnızca bu tablodaki türlerden biri girilebilir. Böylelikle veri doğruluğu sağlanmış olur.

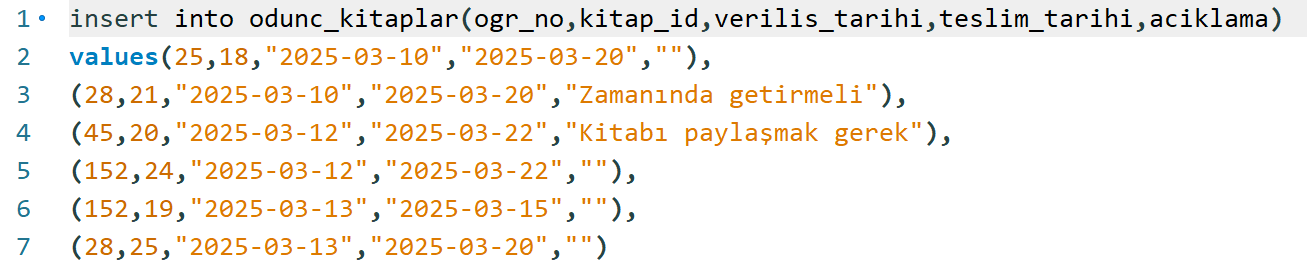


1. **ÖRNEK VERİ GİRİŞİ:** Test amacıyla “***kitaplar, ogrenciler, kitap\_turleri ve odunc\_kitaplar***” tablolarına ilişkisel veri tabanına aşağıdaki örnek veriler eklenmiştir:









Tablolardan biri olan “***odunc\_kitaplar***” tablosuna girilen değerlerin doğruluğu aşağıdaki SQL komutuyla test edildi. Ayrıca diğer tabloların da doğruluğu test edildi.



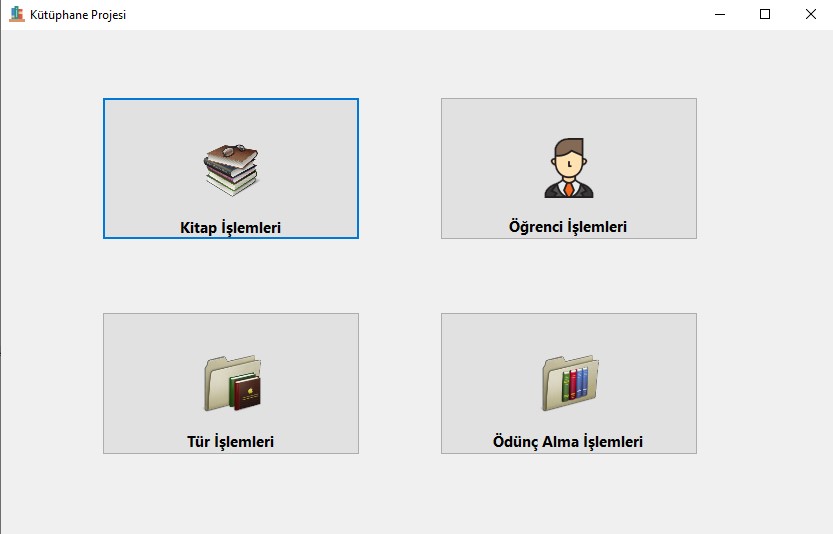
1. **SONUÇ VE GELECEK GELİŞTİRMELER:** Bu veri tabanı tasarımı, kütüphane işleyişini kolaylaştırmak ve kitap takibini düzenlemek için temel bir alt yapı sunmaktadır. Gelecekte, sisteme eklemeler yapılabilir: (Yayınevi, Kitap stok takibi)

**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

**2. HAFTA**

1. **GİRİŞ:** Bu rapor, Kütüphane Otomasyonu Arayüzünün genel taslağını içermektedir. ***Ana Form, Kitap İşlemleri, Öğrenci İşlemleri, Ödünç İşlemleri ve Tür İşlemleri bölümlerinin*** bulunduğu sayfalarla ilgili yapılan görsel çalışmaları içermektedir. Arayüz tasarımında kullanılmak üzere Geliştirme ortamı olarak(IDE) ***Visual Studio 2022***, programlama dili olarak ***C#*** kullanılmıştır.
2. **KÜTÜPHANE OTOMASYONU ARAYÜZ TASARIMI:**
3. **Proje Ana Sayfasının Tasarımı**



Projede yönlendirme işlemlerinin yapılması için ilk olarak bir Ana Sayfa hazırlanır. Bu sayfa içinde diğer sayfaların açılmasını sağlayacak butonlar bulunur.

Form nesnesinde ve içindeki form elemanlarında bazı özelliklerin aşağıdaki gibidir.

**Form Nesnesi:**

“**Text**” özelliği : **Kütüphane Projesi**

“**Name**” özelliği : **formAnaSayfa**

“**Size**” özelliği :**850;615**

“**FormBorderStyle**” özelliği : **FixedSingle**

“**StartPosition**” özelliği : **CenterScreen**

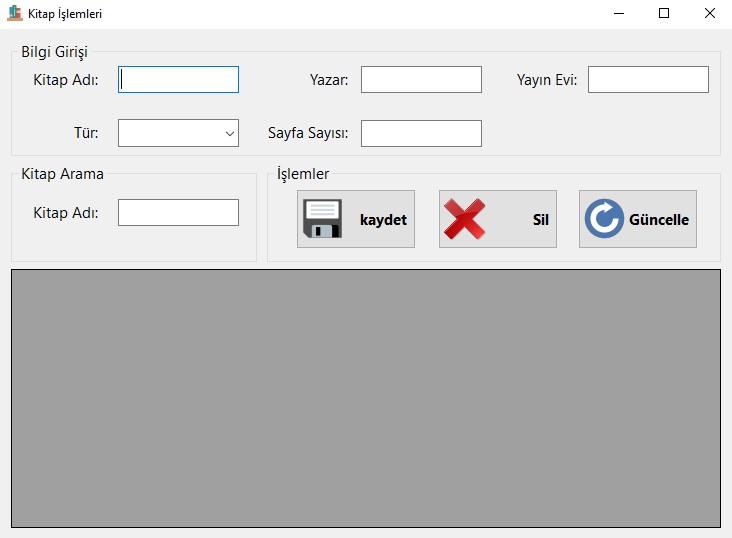
**Button Nesneleri**

“**Text**” özellikleri : **Kitap İşlemleri**, **Öğrenci İşlemleri**, **Tür İşlemleri** ve **Ödünç Alma İşlemleri**

“**Name**” özellikleri : **btnKitap**, **btnOgrenci**, **btnTur**, **btnOdunc**

“**Size**” özellikleri : **260;140**

1. **Kitap İşlemleri Sayfasının Tasarımı**



Kitap İşlemleri sayfası; kütüphanedeki kitaplar ile ilgili ekleme, silme, güncelleme, listeleme ve arama gibi işlemlerin yapılacağı sayfadır. Formun alt bölümündeki “DataGridView” nesnesi, verilerin listelenmesi için kullanılacaktır.

Form nesnesinde ve içindeki form elemanlarında bazı özelliklerin aşağıdaki gibidir.

**Form Nesnesi**

“**Text**” özelliği : **Kitap İşlemleri**

“**Name**” özelliği : **formKitap**

“**Size**” özelliği : **750;550**

“**FormBorderStyle**” özelliği : **FixedSingle**

“**StartPosition**” özelliği : **CenterScreen**

**TextBox Nesneleri**

“**Name**” özellikleri : **txtKitapAdi, txtYazar, txtYayinEvi, txtSayfaSayisi ve txtKitapAra**

**ComboBox Nesnesi**

“**Name**” özelliği : **comboKitapTur**

**Button Nesneleri**

“**Name**” özellikleri : **btnKaydet, btnSil, btnGuncelle**

“**Text**” özellikleri : **Kaydet, Sil, Güncelle**

“**Size**” özellikleri : **120,60**

**DataGridView Nesnesi**

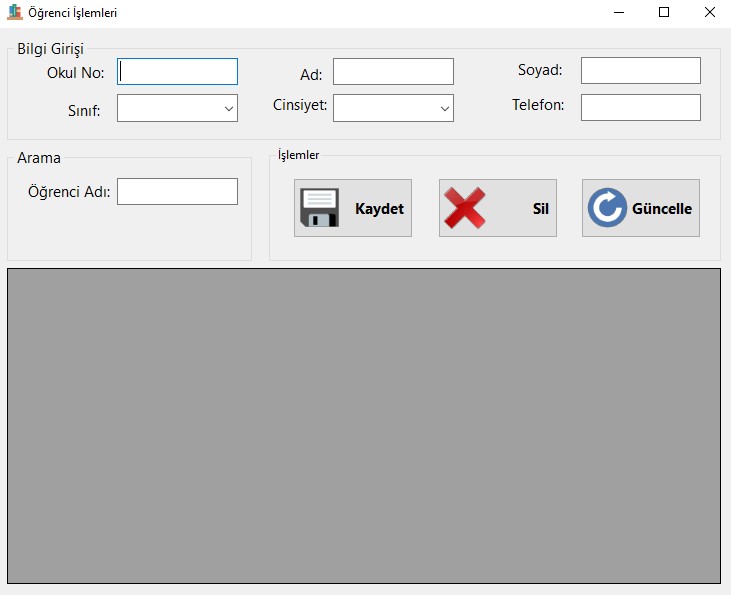
“**Name**” özelliği : **gridKitap**

“**EditMode**” özelliği : **EditProgrammatically**

“**SelectionMode**” özelliği : **FullRowSelect**

“**AutoSizeColumnsMode**” : **Fill**

1. **Öğrenci İşlemleri Sayfasının Tasarımı**



Öğrenci İşlemleri sayfası; öğrenciler ile ilgili ekleme, silme, güncelleme, listeleme ve arama gibi işlemlerin yapılacağı sayfadır.

Form nesnesinde ve içindeki form elemanlarında bazı özelliklerin aşağıdaki gibidir.

**Form Nesnesi**

“**Text**” özelliği : **Öğrenci İşlemleri**

“**Name**” özelliği : **formOgrenci**

“**Size**” özelliği : **750;600**

“**FormBorderStyle**” özelliği : **FixedSingle**

“**StartPosition**” özelliği : **CenterScreen**

**TextBox Nesneleri**

“**Name**” özellikleri : **txtNo, txtAd, txtSoyad, txtTelefon ve txtOgrenciAra**

**ComboBox Nesnesi**

“**Name**” özelliği : **comboSinif, comboCinsiyet**

**Button Nesneleri**

“**Name**” özellikleri : **btnKaydet, btnSil, btnGuncelle**

“**Text**” özellikleri : **Kaydet, Sil, Güncelle**

“**Size**” özellikleri : **120,60**

**DataGridView Nesnesi**

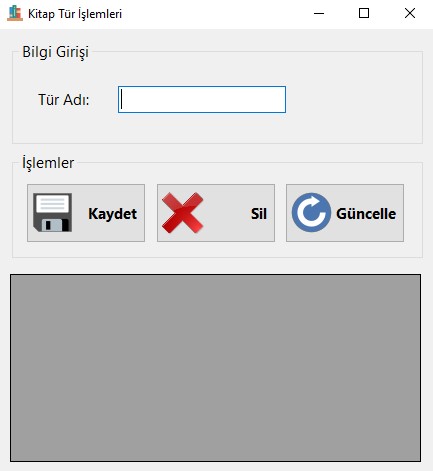
“**Name**” özelliği : **gridOgrenci**

“**EditMode**” özelliği : **EditProgrammatically**

“**SelectionMode**” özelliği : **FullRowSelect**

“**AutoSizeColumnsMode**” : **Fill**

1. **Kitap Tür İşlemleri Sayfasının Tasarımı**



Kitap Tür İşlemleri sayfası; kitap türleri ile ilgili ekleme, silme, güncelleme, listeleme gibi işlemlerin yapılacağı sayfadır.

Form nesnesinde ve içindeki form elemanlarında bazı özelliklerin aşağıdaki gibidir.

**Form Nesnesi**

“**Text**” özelliği : **Kitap Tür İşlemleri**

“**Name**” özelliği : **formKitapTur**

“**Size**” özelliği : **450;485**

“**FormBorderStyle**” özelliği : **FixedSingle**

“**StartPosition**” özelliği : **CenterScreen**

**TextBox Nesneleri**

“**Name**” özellikleri : **txtTurAdi**

**Button Nesneleri**

“**Name**” özellikleri : **btnKaydet, btnSil, btnGuncelle**

“**Text**” özellikleri : **Kaydet, Sil, Güncelle**

“**Size**” özellikleri : **120,60**

**DataGridView Nesnesi**

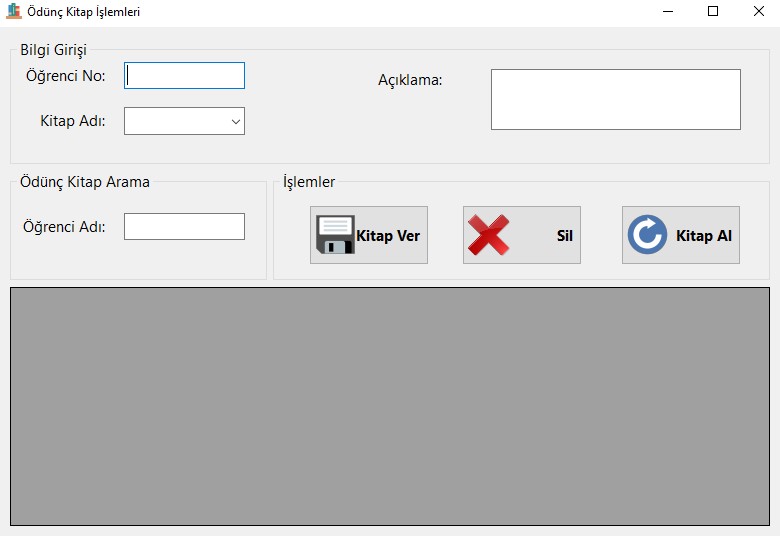
“**Name**” özelliği : **gridKitapTur**

“**EditMode**” özelliği : **EditProgrammatically**

“**SelectionMode**” özelliği : **FullRowSelect**

“**AutoSizeColumnsMode**” : **Fill**

1. **Ödünç Kitap İşlemleri Sayfasının Tasarımı**



Ödünç Kitap İşlemleri sayfası, kütüphanedeki kitapların öğrencilere ödünç olarak verilmesi ve geri alınması işlemlerinin yapılacağı sayfadır. Ödünç kitap verme ve geri alma işlemlerinde tarih olarak, sistem tarafından o günün tarihi otomatik olarak verilecektir.

Form nesnesinde ve içindeki form elemanlarında bazı özelliklerin aşağıdaki gibidir.

**Form Nesnesi**

“**Text**” özelliği : **Ödünç Kitap İşlemleri**

“**Name**” özelliği : **formOduncKitap**

“**Size**” özelliği : **800;550**

“**FormBorderStyle**” özelliği : **FixedSingle**

“**StartPosition**” özelliği : **CenterScreen**

**TextBox Nesneleri**

“**Name**” özellikleri : **txtNo, txtAciklama ve txtAramaOgrenci**

**ComboBox Nesnesi**

“**Name**” özelliği : **comboKitap**

**Button Nesneleri**

“**Name**” özellikleri : **btnKitapVer, btnSil, btnKitapAl**

“**Text**” özellikleri : **Kitap Ver, Sil, Kitap Al**

“**Size**” özellikleri : **120,60**

**DataGridView Nesnesi**

“**Name**” özelliği : **gridOduncKitaplar**

“**EditMode**” özelliği : **EditProgrammatically**

“**SelectionMode**” özelliği : **FullRowSelect**

“**AutoSizeColumnsMode**” : **Fill**

**“Image” Özellikleri**

Form ve Butonlara eklenecek iconlar ücretsiz resimlerin olduğu <https://icon-icons.com/> web sayfasından indirilmiştir. İndirilen resimler araçlar menüsünden eklenen **ImageList** bileşeniyle birlikte Form nesnelerine ve Butonlara eklenmiştir.

1. **FORMLAR ARASI GEÇİŞ KODLARI**

Oluşturulan Ana Form bölümünde butonlara tıklandığında ilgili sayfalara yönlendirme işleminin yapılması için butonların “click” olayına bazı kodların yazılması gerekir. Kodlar aşağıdaki gibidir.

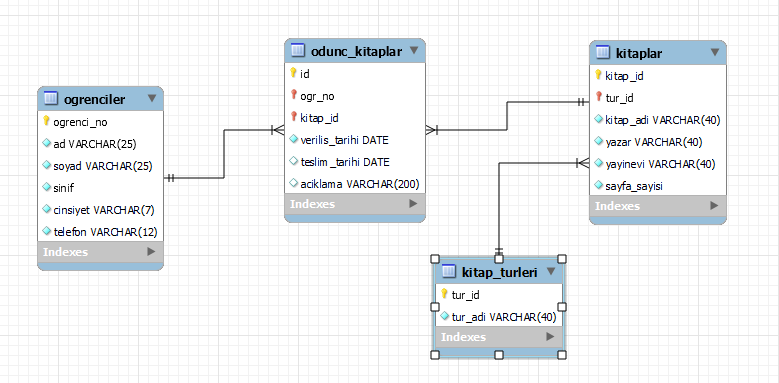
|  |
| --- |
| private void btnOgrenci\_Click(object sender, EventArgs e)  {  formOgrenci ogrenci = new formOgrenci();  ogrenci.ShowDialog();  }  private void btnTur\_Click(object sender, EventArgs e)  {  formKitapTur kitapTur = new formKitapTur();  kitapTur.ShowDialog();  }  private void btnKitap\_Click(object sender, EventArgs e)  {  formKitap kitap = new formKitap();  kitap.ShowDialog();  }  private void btnOdunc\_Click(object sender, EventArgs e)  {  formOduncKitap oduncKitap = new formOduncKitap();  oduncKitap.ShowDialog();  } |

**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

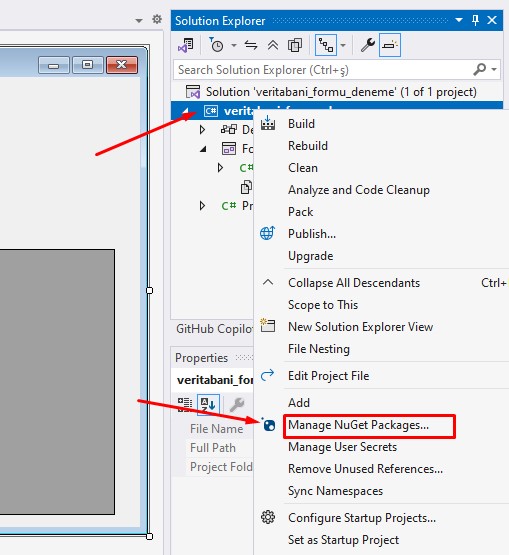
**3. HAFTA**

1. **GİRİŞ:** Bu rapor, ***Ana Form*** üzerinde bulunan ***Kitap Tür İşlemleri formunun*** veri tabanıyla bağlantı kurularak, *Listeleme*, *Kitap Kaydı*, *Silme*, *Güncelleme*, *Arama* işlemlerinin yapıldığı çalışmaları içermektedir.
2. **KİTAP TÜR İŞLEMLERİ FORM SAYFASI KODLAMALARI VE VERİ TABANI TASARIMI**

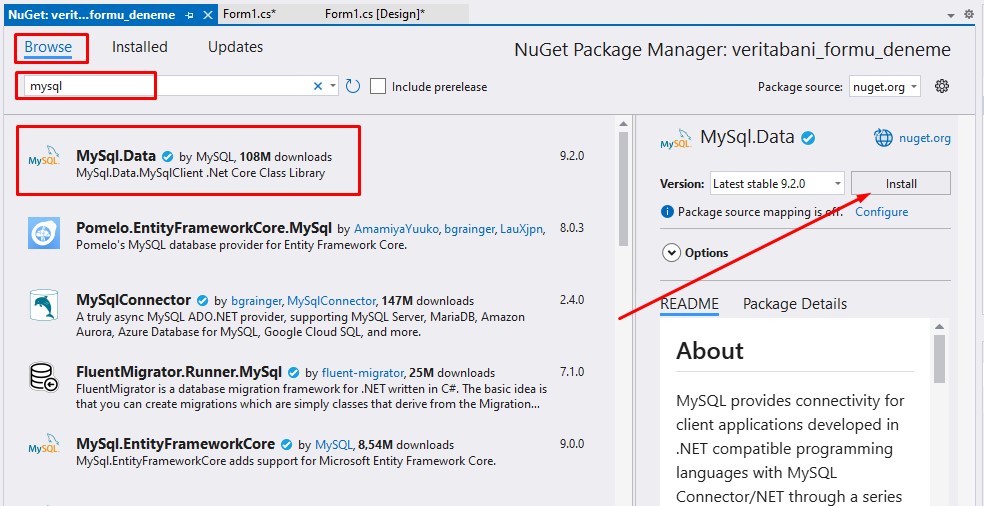
İlişkisel veri tabanında ***kitaplar*** tablosundaki foreing key alanı ***kitap\_turleri*** tablosundaki *primary key* alanına bağladığımız için öncelikle **Kitap Tür İşlemleri Form Sayfasını** tasarlamamız gerekmektedir.

1. **MySql.Data paketinin yüklenmesi**

C# ile MySQL veri tabanı bağlantısı kurmak için **MySql.Data** Referansını projemize eklememiz gerekmektedir. **NuGet** Paket Yöneticisini kullanarak **MySql.Data** yazılımını projemize ekliyoruz.



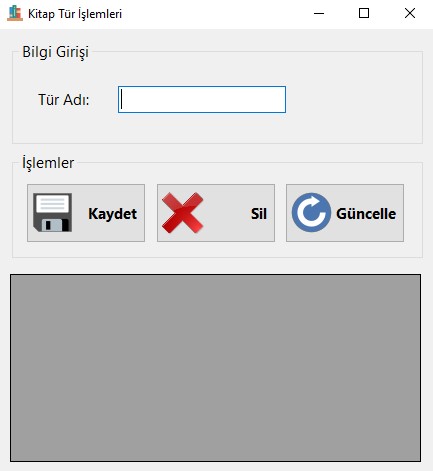
Paket yöneticisi için Solution Explorer penceresinde proje üzerinde sağ tıklayarak **Manage NuGet Packages** tıklanır.



Açılan pencerede **Browse** sekmesinde *MySql* araması yaparak MySql.Data paketinin kurulumunu yapılır.

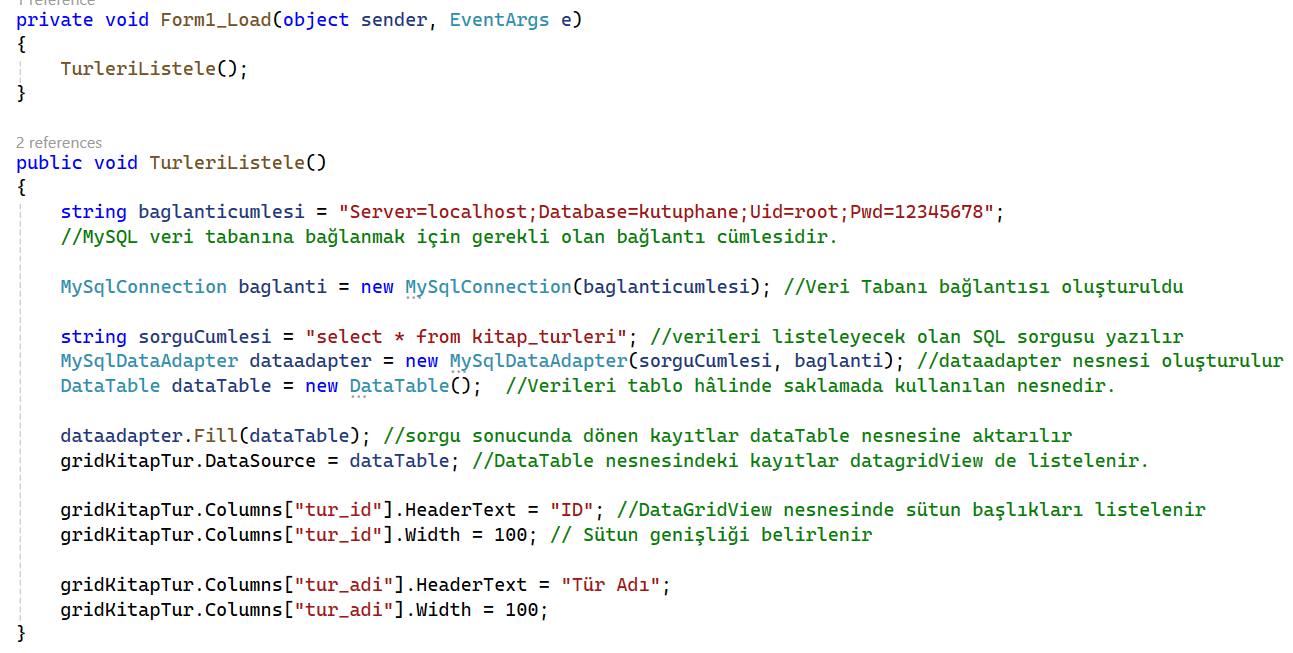
Bu işlem sonucunda MySql.Data referansı projemize eklenmiş olacaktır.

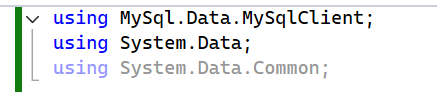
1. **Kitap Tür Formu ve Veri Tabanı Bağlantısı (Listeleme İşlemi)**

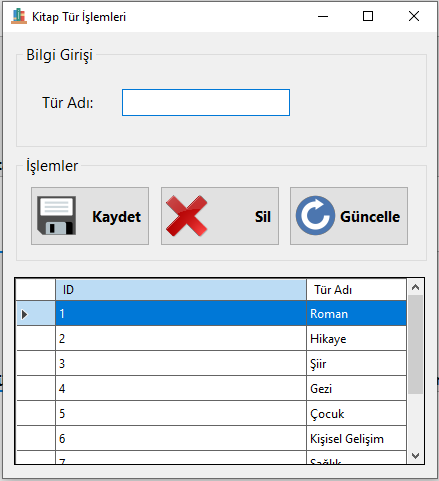
\* *Önceki hafta oluşturulan raporda belirtildiği gibi yandaki gibi bir form oluşturulur.*

Kitap türlerini listeleme kodları **TurleriListele()** metodu içine yazılır ve bu metot, formun yüklenmesi olayında çağrılır. Bu işlemden sonra Veri tabanından çekilecek verilerin (kitap türlerinin) **DataGridView** nesnesinde listelendiği görülür

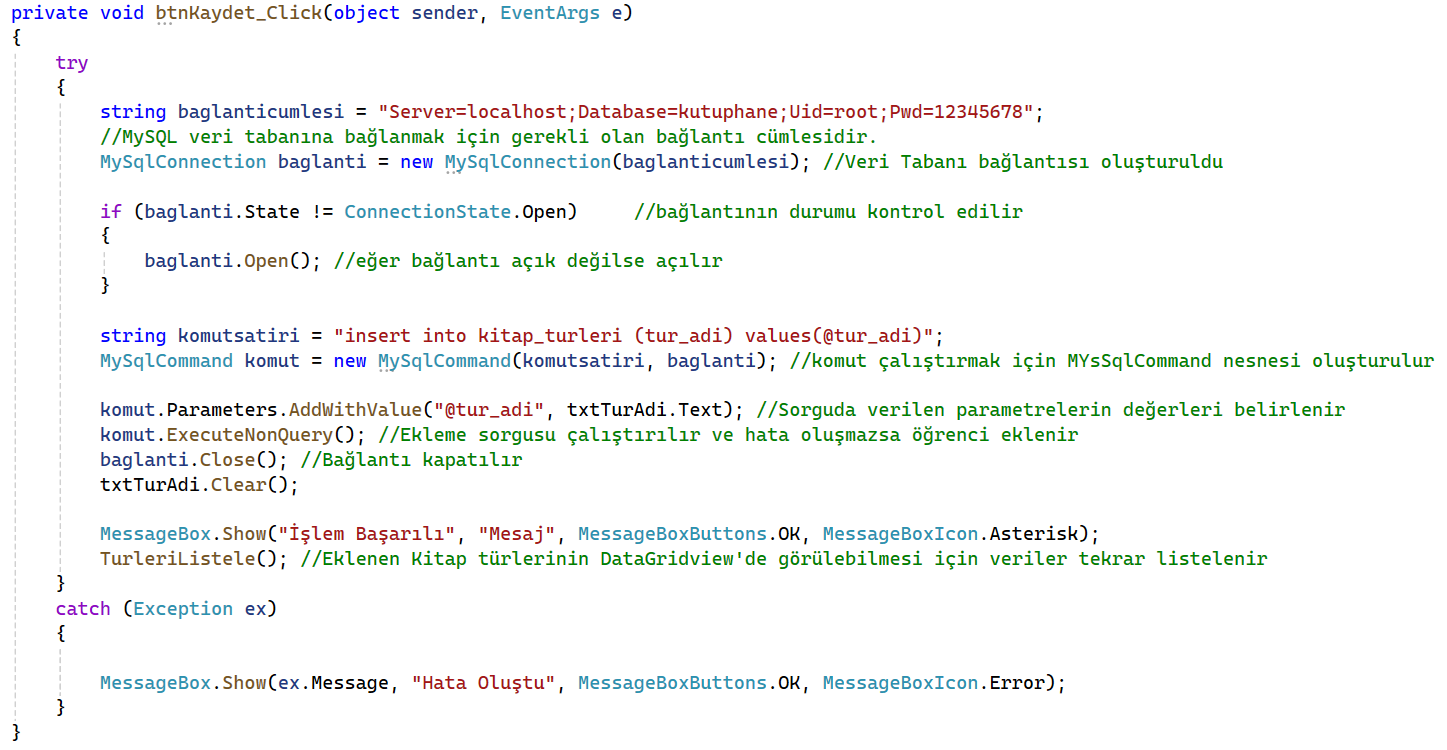
**TurleriListele()** fonksiyonunun içindeki kod bölümleriyle ilgili açıklamalar açıklama satırı olarak eklenmiştir.



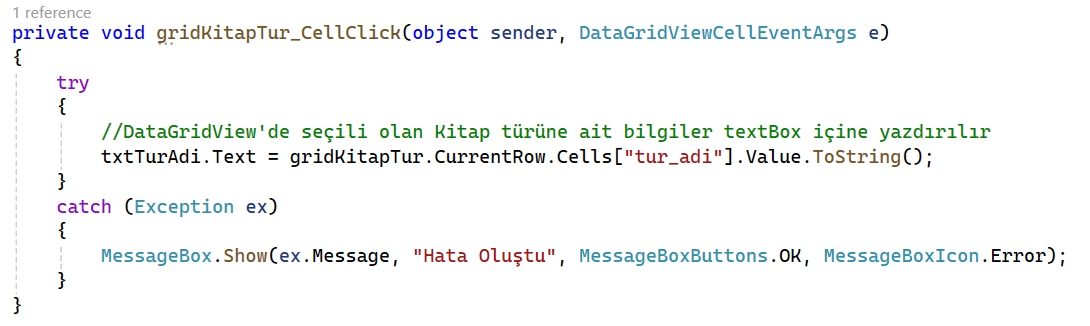
Yukarıdaki bazı kodların çalışması için yandaki kütüphanelerin eklenmesi otomatik olarak yapılır. Eklenme yapılmadıysa elle ekleme yapılır.

Veri tabanıyla bağlantı kurulmuş olup yandaki gibi bir görsel elde edilir.

1. **Kitap Tür Formu Kayıt İşlemi**

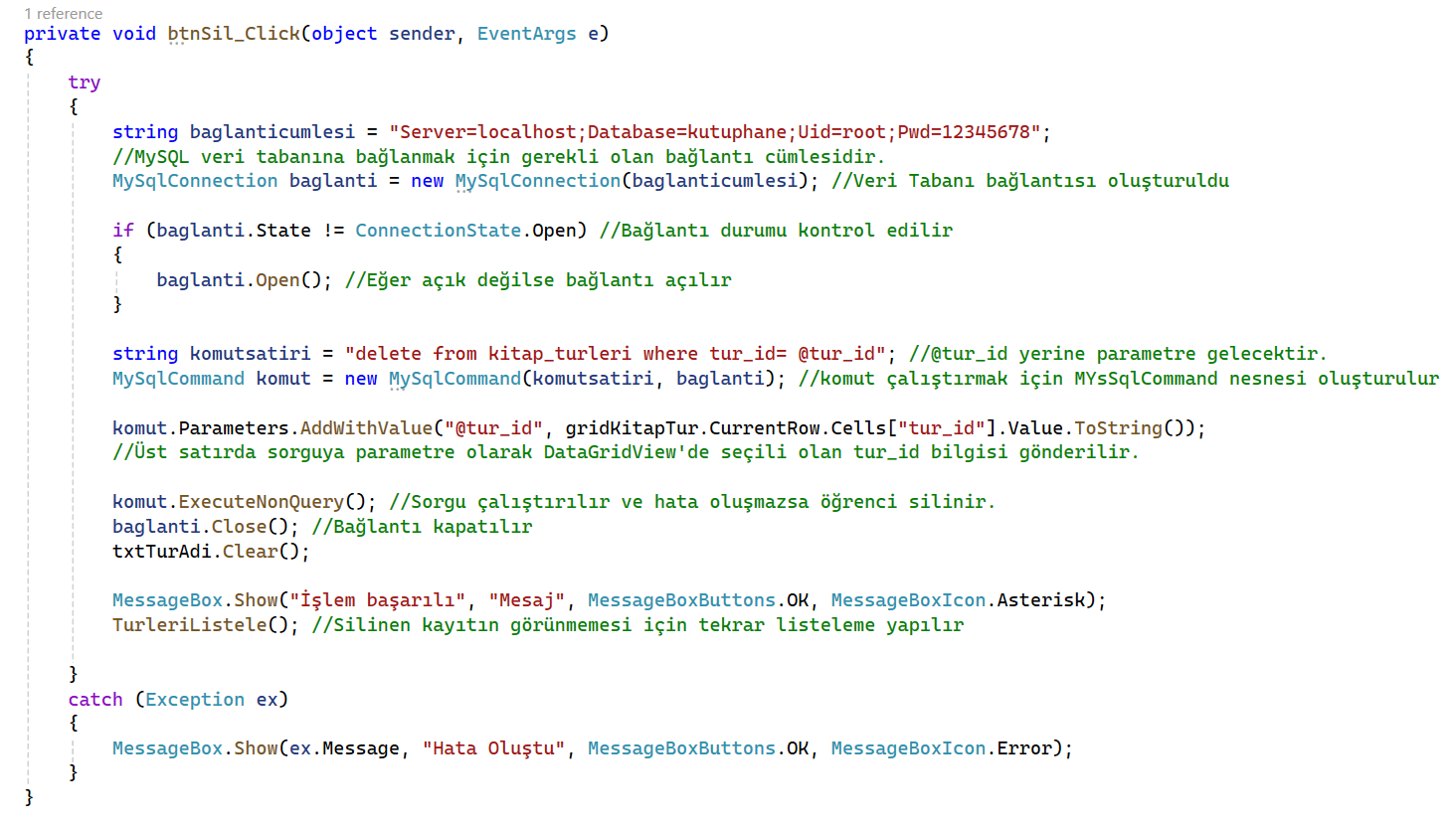
Kitap Tür İşlemleri form ekranına tür adı bilgisi girişi yapıldıktan sonra “**Kaydet**” butonuna tıklandığında aşağıdaki kodlar çalıştırılır ve kitap türü kayıt işlemi gerçekleştirilir. Ekleme kodları “Kaydet” butonunun tıklanma olayına yazılır. Try-Catch bloğu içine yazılan kodlarda hata olması durumunda uyarı mesajı devreye girecektir.

1. **Kitap Tür Formu Silme İşlemi**

Kitap türünü ***silme ve güncelleme*** işlemleri yapılmadan önce kitap türünün ***DataGridView*** üzerinde seçilmesi ve o kayıt ile ilgili bilginin ***TextBox*** nesnesinde gösterilmesi gerekir. Bunun için nesnenin “CellClick” olayına aşağıdaki kodlar yazılır.

***DataGridView*** üzerinde silinmesi istenen kitap türü seçildikten sonra o türün adı ***TextBox*** nesnesinde gösterilir. “***Sil***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve kayıt, veri tabanından silinir. Kodlar “***Sil***” butonunun tıklanma olayına yazılır.

**Ek bilgi:** “***Kitaplar***” ve “***kitap\_turleri***” tabloları arasında ilişki oluşturulduğu için kayıtlı bir kitaba ait tür sistemden silinemez.



1. **Kitap Tür Formu Güncelleme İşlemi**

DataGridView üzerinde güncellenmesi istenen kitap türü seçildikten sonra o türün adı TextBox nesne sinde gösterilir. Yeni değer girildikten sonra “***Güncelle***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve kitap türünü güncelleme işlemi gerçekleştirilir. Kodlar “***Güncelle***” butonunun tıklanma olayına yazılır.

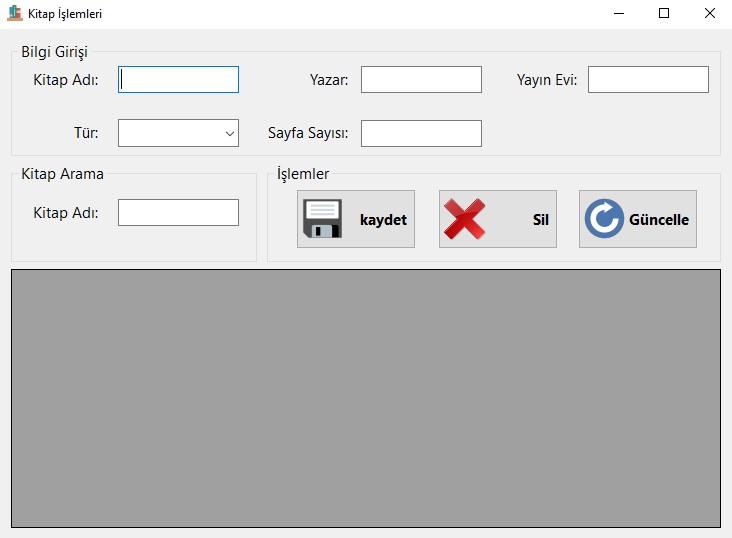


**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

**4. HAFTA**

1. **GİRİŞ:** Bu rapor, ***Ana Form*** üzerinde bulunan ***Kitap İşlemleri formunun*** veri tabanıyla bağlantı kurularak, *Listeleme*, *Kitap Kaydı*, *Silme*, *Güncelleme*, *Arama* işlemlerinin yapıldığı çalışmaları içermektedir.
2. **KİTAP İŞLEMLERİ FORM SAYFASI KODLAMALARI VE VERİ TABANI TASARIMI**
3. **Kitap İşlemleri Formu ve Veri Tabanı Bağlantısı (Listeleme İşlemi)**



\* *Yukarıdaki gibi bir form oluşturulur.*

**Kitap Türlerini (Combobox) Listeleme İşlemi**

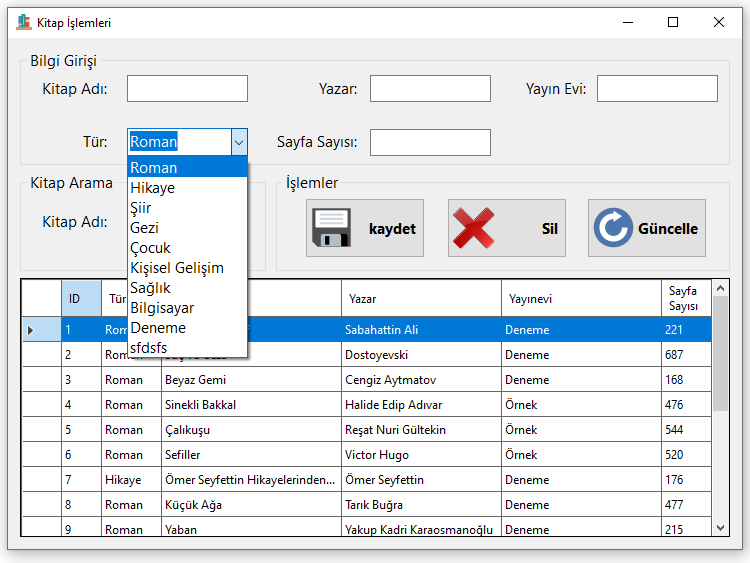
ComboBox nesnesinde kitap türlerinin listelenmesi için gerekli olan kodlar ***KitapTurYukle()*** metodunun içine yazılır. Daha sonra bu metot, formun yüklenmesi olayında çağrılır.



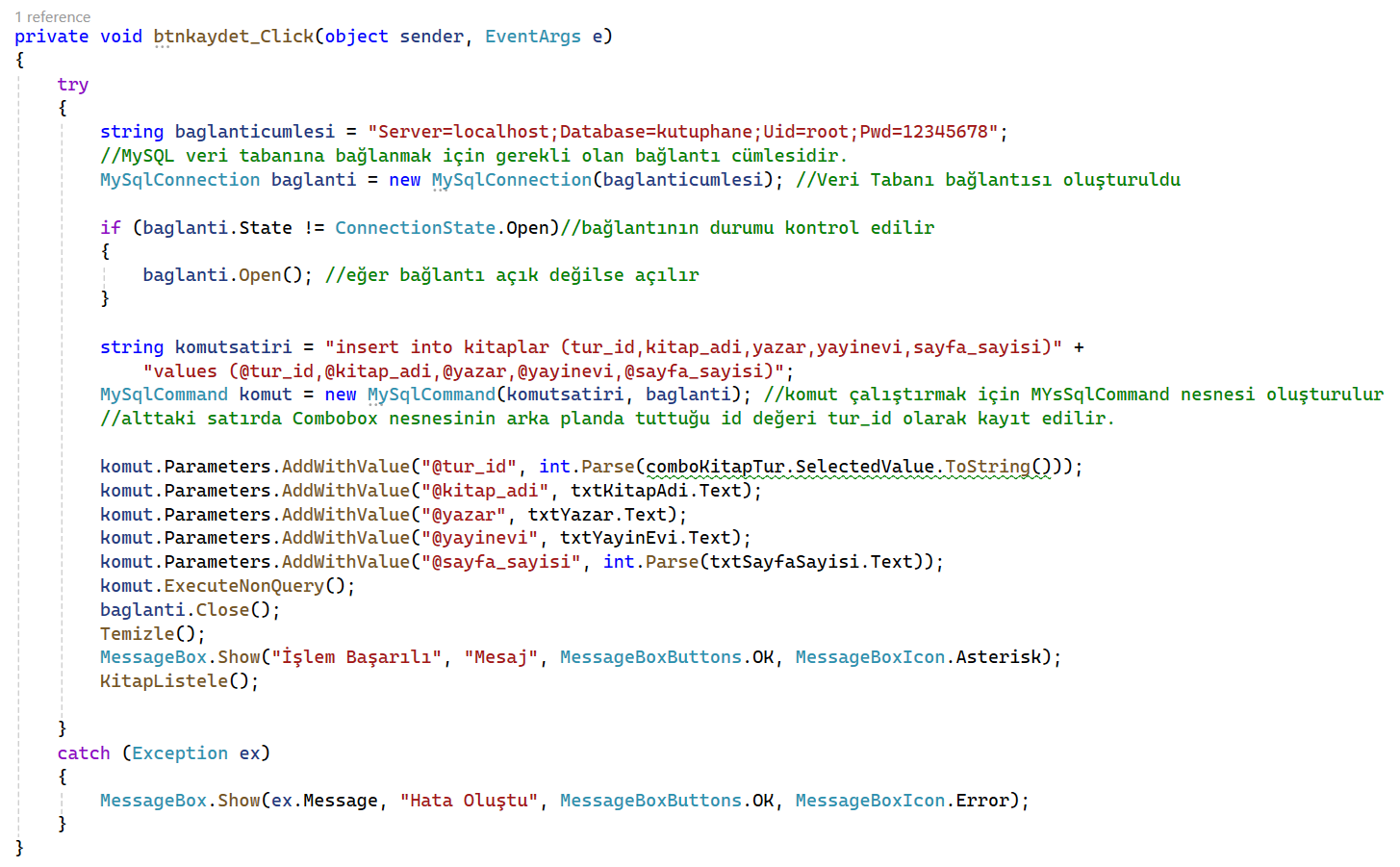
**Kitapların Listeleme İşlemi**

DataGridView nesnesinde kitapların listelenmesi için gerekli olan kodlar ***KitapListele()*** metodunun içine yazılır. Daha sonra bu metot, formun yüklenmesi olayında çağrılır.

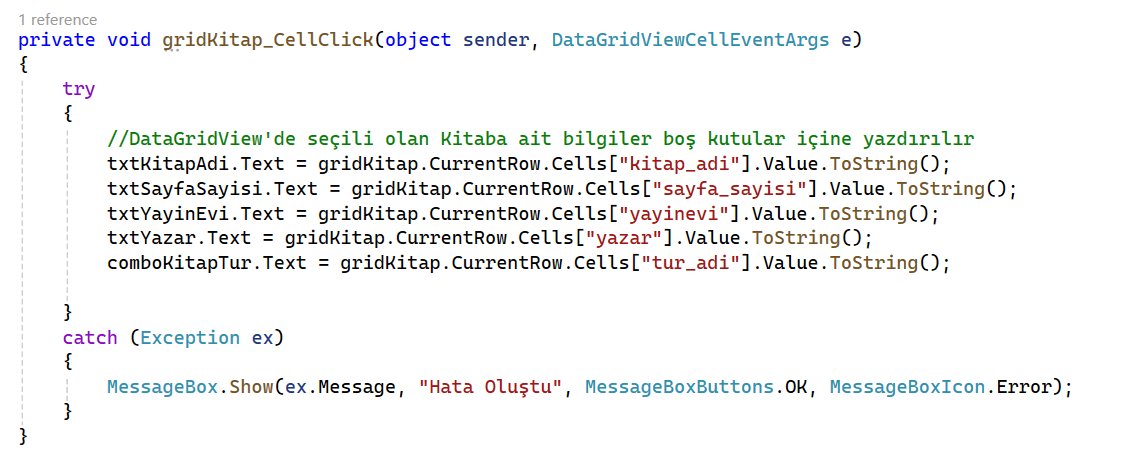


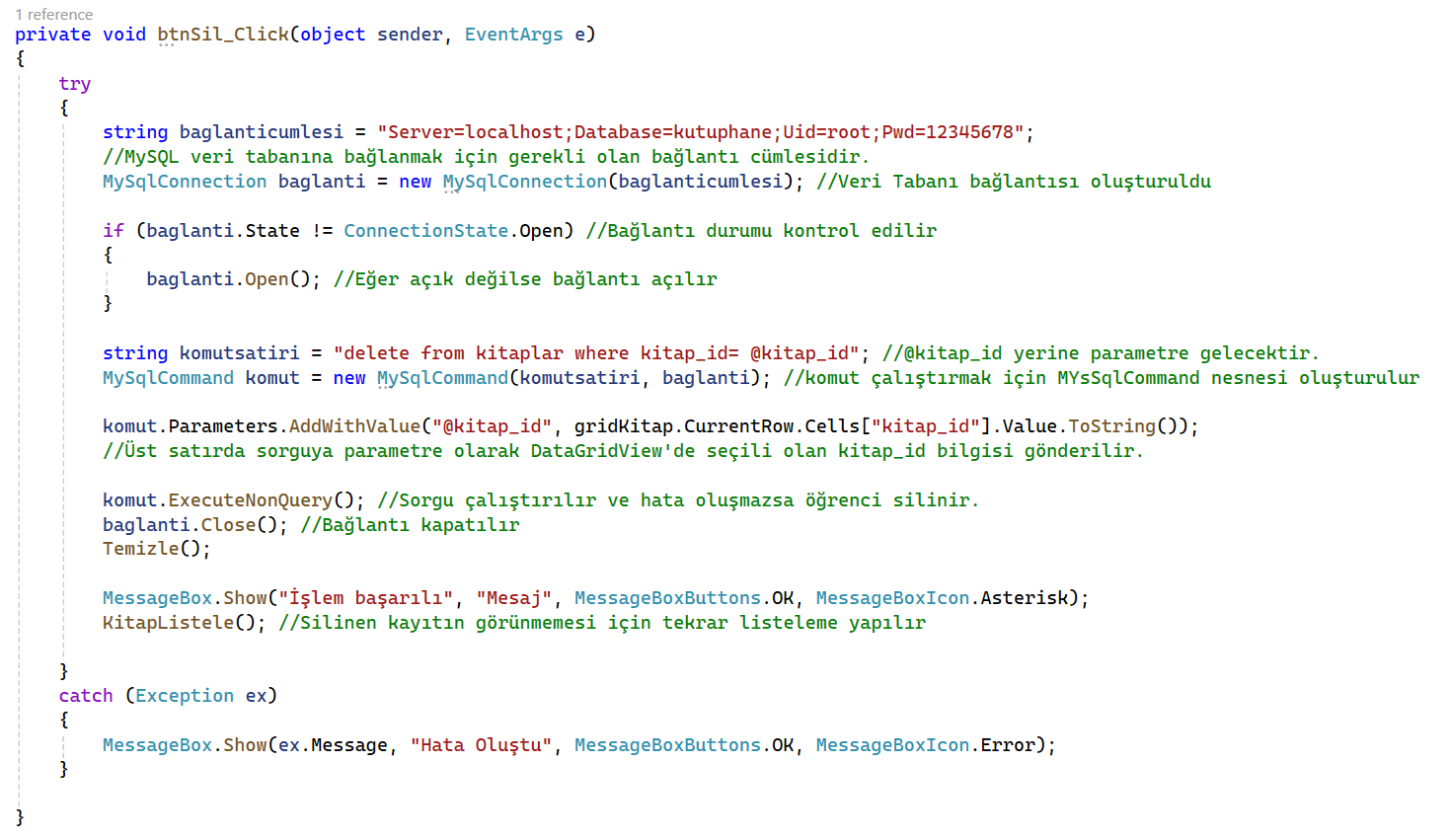
Bu işlemlerden sonra kitap bilgilerinin ***DataGridView*** nesnesinde, kitap türlerinin de ***ComboBox*** nesnesinde listelendiği görülür.

1. **Kitap İşlemleri Formu Kayıt İşlemi**

Kitap İşlemleri form ekranına veri girişi yapıldıktan sonra “**Kaydet**” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve kitap kayıt işlemi gerçekleştirilir. Kodlar “**Kaydet**” butonunun tıklanma olayına yazılır.

1. **KİTAP İŞLEMLERİ FORM SAYFASI KODLAMALARI VE VERİ TABANI TASARIMI**
2. **Kitap İşlemleri Formu Silme İşlemi**

Kitap silme ve güncelleme işlemleri yapılmadan önce işlem yapılacak kitabın ***DataGridView*** üzerinde seçilmesi ve o kayıt ile ilgili bilgilerin form elemanlarında gösterilmesi gerekir. Bunun için nesnenin “CellClick” olayına aşağıdaki kodlar eklenmiştir.

***DataGridView*** üzerinde silinmesi istenen kitap seçildikten sonra o kitaba ait bilgiler form elemanların da gösterilir. “***Sil***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve kitap bilgileri veri tabanından silinir. Kodlar “***Sil***” butonunun tıklanma olayına yazılır. Ödünç verilen bir kitap tablolar arası bağlantıdan dolayı silinemez.

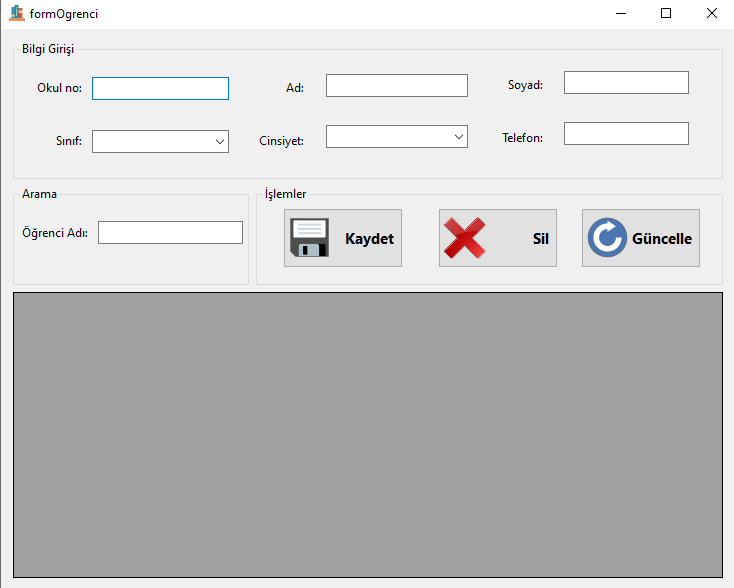
1. **Kitap İşlemleri Formu Güncelleme İşlemi**



**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

**5. HAFTA**

1. **GİRİŞ:** Bu rapor, ***Ana Form*** üzerinde bulunan ***Öğrenci İşlemleri formunun*** veri tabanıyla bağlantı kurularak, *Listeleme*, *Kitap Kaydı*, *Silme*, *Güncelleme*, *Arama* işlemlerinin yapıldığı çalışmaları içermektedir.
2. **ÖĞRENCİ İŞLEMLERİ FORM SAYFASI KODLAMALARI VE VERİ TABANI TASARIMI**
3. **Öğrenci Formu ve Veri Tabanı Bağlantısı (Listeleme İşlemi)**

\* *Yukarıdaki gibi bir form oluşturulur.*

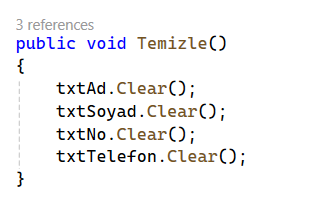
Öğrenci listeleme kodları bir metot içine yazılır ve bu metot, formun yüklenmesi olayında çağrılır. Bu işlemden sonra öğrenci bilgilerinin DataGridView nesnesinde listelendiği görülür.





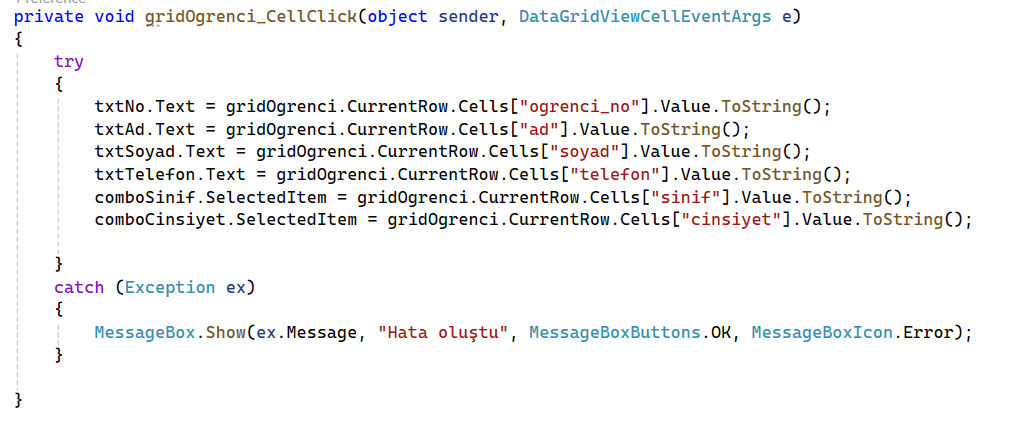
1. **Öğrenci İşlemleri Formu Kayıt İşlemi**

Öğrenci İşlemleri form ekranına veri girişi yapıldıktan sonra “***Kaydet***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve öğrenci kayıt işlemi gerçekleştirilir. Kodlar “***Kaydet***” butonunun tıklanma olayına yazılır. Öğrenci eklenirken aynı numaraya sahip bir öğrenci daha kaydedilmek istenirse program hata verecektir. Bunun nedeni, ***öğrenci\_no*** alanının ***PK*** olarak belirlenmesidir. Kayıt işleminin başarılı olması için bilgilerin doğru bir şekilde girilmesi gerekir.



1. **Öğrenci İşlemleri Formu Silme İşlemi**

Öğrenci silme ve güncelleme işlemleri yapılmadan önce öğrencinin DataGridView üzerinde seçilmesi ve o kayıt ile ilgili bilgilerin form elemanlarında gösterilmesi gerekir. Bunun için nesnenin “***CellClick*** (Hücrenin tıklanması)” olayına aşağıdaki kodların yazılması gerekir.



DataGridView üzerinde silinmesi istenen öğrenci seçildikten sonra o öğrenciye ait bilgiler form elemanlarında gösterilir. “Sil” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve öğrenci bilgileri veri tabanından silinir. Kodlar “Sil” butonunun tıklanma olayına yazılır. “***Ogrenciler***” ve “***odunc\_kitaplar***” tabloları arasında ilişki oluşturulduğu için ödünç kitap alan öğrenci sistemden silinemez.

1. **Öğrenci İşlemleri Formu Güncelleme İşlemi**

DataGridView üzerinde güncellenmesi istenen öğrenci seçildikten sonra o öğrenciye ait bilgiler form elemanlarında gösterilir. Yeni değerler girildikten sonra “***Güncelle***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve öğrenci bilgileri güncellenir. Kodlar “***Güncelle***” butonunun tıklanma olayına yazılır.



1. **Öğrenci İşlemleri Formu Arama İşlemi**

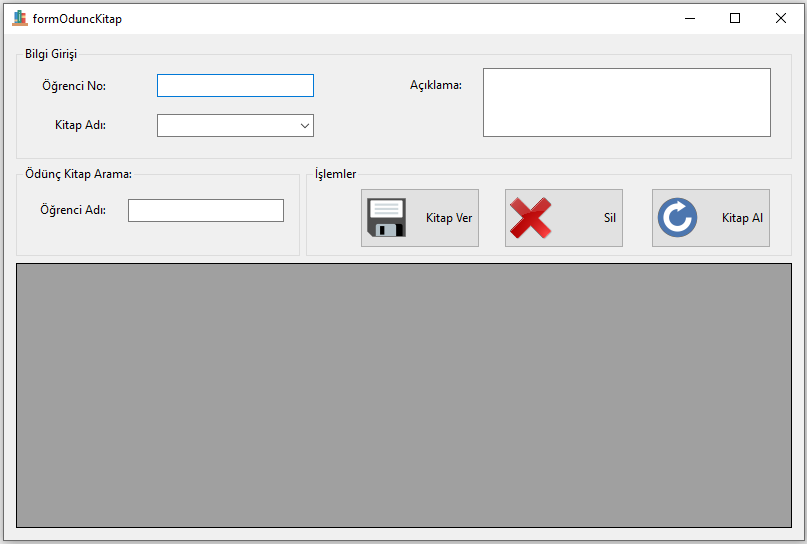
Form üzerindeki arama bölümünde öğrenci adına göre arama yapılabilmesini sağlayan kodlar aşağıdaki gibidir. Kodlar “***TextChanged***” olayına yazıldığı için arama kutusundaki her değişiklikte arama işlemi tekrarlanır ve sonuçlar DataGridView nesnesinde gösterilir.



**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

**6. HAFTA**

1. **GİRİŞ:** Bu rapor, ***Ana Form*** üzerinde bulunan ***Ödünç Kitap İşlemleri formunun*** veri tabanıyla bağlantı kurularak, *Listeleme*, *Ödünç Kitap Kaydı*, *Silme*, *Kitap Geri Alma*, *Arama* işlemlerinin yapıldığı çalışmaları içermektedir.
2. **ÖDÜNÇ KİTAP İŞLEMLERİ FORM SAYFASI KODLAMALARI VE VERİ TABANI TASARIMI**
3. **Ödünç Kitap İşlemleri Formu ve Veri Tabanı Bağlantısı (Listeleme İşlemi)**

\* *Yukarıdaki gibi bir form oluşturulur.*

Ödünç Kitap İşlemleri listeleme kodları ***Listele()*** metodu içine yazılır ve bu metot, formun yüklenmesi olayında çağrılır. Ayrıca ***KitapYukle()*** metoduyla veri tabanındaki kitap bilgileri combobox a yüklenir. Bu işlemden sonra Ödünç Kitap bilgilerinin DataGridView nesnesinde listelendiği görülür.





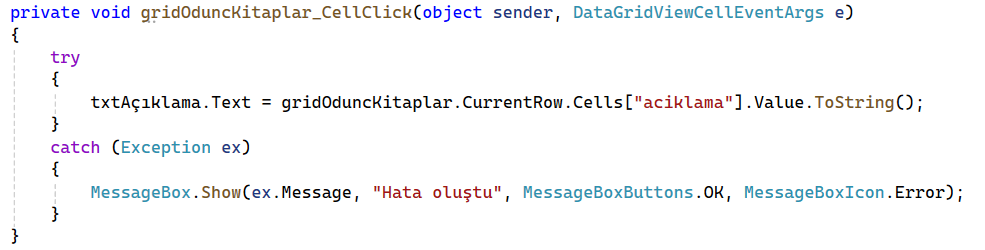
1. **Ödünç Kitap Verme İşlemleri**

Ödünç Kitap İşlemleri form ekranında öğrenci ve emanet verilecek kitap seçildikten sonra ***“Kitap Ver***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır. Kodlar “***Kitap Ver***” butonunun tıklanma olayına yazılır.



1. **Ödünç Kitap İşlemleri Formu Silme İşlemi**

Ödünç Kitap İşlemleri silme ve güncelleme işlemleri yapılmadan önce verilmiş olan kitap kaydının DataGridView üzerinde seçilmesi ve o kayıt ile ilgili bilgilerin form elemanlarında gösterilmesi gerekir. Bunun için nesnenin “***CellClick*** (Hücrenin tıklanması)” olayına aşağıdaki kodların yazılması gerekir.



DataGridView üzerinde silinmesi istenen kayıt seçildikten sonra o kayıta ait bilgiler form elemanlarında gösterilir. “Sil” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır ve öğrenci bilgileri veri tabanından silinir. Kodlar “Sil” butonunun tıklanma olayına yazılır.



1. **Ödünç Kitap İşlemleri Formu Geri Alma İşlemi**

DataGridView üzerinde teslim alınacak ödünç kitap kaydı seçildikten sonra açıklama bölümü doldurularak ***“Kitap Al”*** butonu tıklandığında aşağıdaki kodlar çalıştırılır ve kitabı geri alma işlemi gerçekleştirilir.



1. **Ödünç Kitap İşlemleri Formu Arama İşlemi**

Form üzerindeki arama bölümünde öğrenci adına göre arama yapılabilmesini sağlayan kodlar aşağıdaki gibidir. Kodlar ***“TextChanged”*** olayına yazıldığı için arama kutusundaki her değişiklikte arama işlemi tekrarlanır ve sonuçlar DataGridView nesnesinde gösterilir.



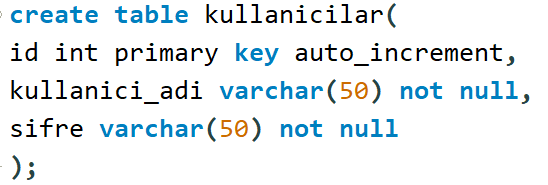
**KÜTÜPHANE OTOMASYONU PROJESİ**

**HAFTALIK SUNUM VE RAPORLAMA**

**7. HAFTA**

1. **GİRİŞ:** Bu rapor, ayrı bir Giriş Formu ile Kullanıcı adı ve Şifre girişiyle veri tabanıyla bağlantı kurularak, diğer form sayfalarına erişim izni verilen işlemlerinin yapıldığı çalışmaları içermektedir.
2. **KÜTÜPHANE OTOMASYONU GİRİŞ PANELİ VERİ TABANI TASARIMI**

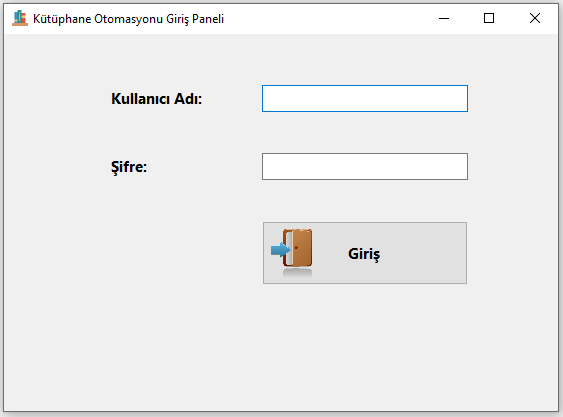
**Kullanıcılar Tablosu Oluşturulması**



**Kullanıcılar Tablosu Veri girişi**



1. **KÜTÜPHANE OTOMASYONU GİRİŞ PANELİ FORM SAYFASI KODLAMALARI**



Yukardaki gibi bir görsel panel hazırlanır. Kullanıcı adı ve şifre alanları doldurulduktan sonra ***“Giriş***” butonuna tıklandığı zaman aşağıdaki kodlar çalıştırılır.

Kullanıcı adı ve şifre alanlarının boş bırakılması ***“IsNullOrEmpty”*** metoduyla kontrol edilmiştir. Veri tabanıyla bağlantı kurularak kayıtlar kontrol edilerek eşleşen kayıt varsa AnaForm sayfamıza erişim sağlanacaktır.

