

**T.C**

**DÜZCE ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**YURT KAYIT SİSTEMİ**

**METEHAN CEYHAN**

**LİSANS BİTİRME TEZİ**

**DANIŞMAN**

**DR.ÖĞR. ÜYESİ SERDAR BİROĞUL**

**DÜZCE, 2021**

T.C.

DÜZCE ÜNİVERSİTESİ

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

YURT KAYIT SİSTEMİ

METEHAN CEYHAN tarafından hazırlanan bitirme tezi çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü’nde **LİSANS BİTİRME TEZİ** olarak kabul edilmiştir.

**Tez Danışmanı**

Dr.Ögr. Üyesi SERDAR BİROĞUL

Düzce Üniversitesi

**Jüri Üyeleri**

Dr.Ögr. Üyesi SERDAR BİROĞUL

Düzce Üniversitesi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. ……….

Düzce Üniversitesi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Dr. ……….

Düzce Üniversitesi \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tez Savunma Tarihi: …/…/2021

**BEYAN**

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

15 Ocak 2021



Metehan Ceyhan

**TEŞEKKÜR**

Lisans öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Dr.Ögr. Üyesi Serdar Biroğul’a en içten dileklerimle teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve çalışma arkadaşlarıma sonsuz teşekkürlerimi sunarım.

**15 Ocak 2021 Metehan Ceyhan**

**İÇİNDEKİLER**

**ŞEKİL LİSTESİ vi**

**ÖZET viii**

**ABSTRACT ix**

**1. GİRİŞ 1**

**1.1 Projenin Tanıtımı 1**

**1.2 Projenin Amacı 2**

**1.3 Projenin Geliştirildiği Araçlar 2**

***1.3.1* Visual Studio 2**

***1.3.2* Sql Server Management Studio 3**

**2. GENEL BİLGİLER 4**

**2.1 Yazılım 4**

**2.1.1 Bir Yazılımda Olması Gereken Temel Özellikler 4**

**2.1.2 Yazılım Çeşitleri 5**

**2.2 Programlama Araçları 5**

**2.2.1 Programlama Dili 5**

**3. PROJE 6**

**4. SONUÇ VE DEĞERLENDİRME 40**

**KAYNAKLAR 41**

**ÖZGEÇMİŞ 42**

ŞEKİL LİSTESİ

**Sayfa No**

[Şekil 2.1. Admin Giriş Formu Görüntüsü](#_Toc458430475)  6

[Şekil 2.2. Sql Bağlantısı Classı.](#_Toc458430476)  6

[Şekil 2.3. Giriş Yap Button’u Kod Blokları. 7](#_Toc458430477)

[Şekil 3.1. Ana Sayfa Formu 8](#_Toc458430478)

[Şekil 3.2. Dataset Oluşturma. 11](#_Toc458430479)

[Şekil 3.3. Dataset Oluşturma 2. Adım. 12](#_Toc458430480)

[Şekil 3.4. Dataset Oluşturma 3. Adım. 12](#_Toc458430480)

[Şekil 3.5. Dataset Oluşturma 4. Adım.](#_Toc458430480) 13

[Şekil 3.6. Dataset Oluşturma 5. Adım. 13](#_Toc458430480)

[Şekil 4.1. Öğrenci Kayıt Formu. 14](#_Toc458430480)

[Şekil 4.2. Öğrenci Kayıt Formu Yüklendiği Anda Çalışan Kod Satırları. 15](#_Toc458430480)

[Şekil 4.3. Kaydet Button’u Kod Satırları. 16](#_Toc458430480)

[Şekil 5.1. Öğrenci Listele Formu. 17](#_Toc458430480)

[Şekil 5.2. DataGridView’a Tıklandığı Anda Çalışan Kod Satırları. 17](#_Toc458430480)

[Şekil 6.1. Öğrenci Düzenle Formu. 18](#_Toc458430480)

[Şekil 6.2. Güncelle Button’u Kod Satırları. 19](#_Toc458430480)

[Şekil 6.3. Sil Button’u Ve FormLoad Kod Satırları 19](#_Toc458430480)

[Şekil 7.1. Bölümler Formu. 20](#_Toc458430480)

[Şekil 7.2. Bölüm Ekle Ve Bölüm Sil Kod Satırları. 21](#_Toc458430480)

[Şekil 7.3. Bölüm Güncelle Ve DataGridView\_CellClick Kod Satırları 22](#_Toc458430480)

[Şekil 8.1. Ödemeler Formu. 23](#_Toc458430480)

[Şekil 8.2. DataGridView\_CellClick Durumu. 23](#_Toc458430480)

[Şekil 8.3. Ödeme Al Button’u. 24](#_Toc458430480)

[Şekil 9.1. Giderler Formu. 25](#_Toc458430480)

[Şekil 9.2. Kaydet Button’u Kod Satırları 25](#_Toc458430480)

[Şekil 10.1. Giderler Listesi Formu. 26](#_Toc458430480)

[Şekil 10.2. DataGridView’a Tıklandığı Anda Çalışan Kod Satırları. 26](#_Toc458430480)

[Şekil 11.1. Gider Güncelle Formu. 27](#_Toc458430480)

[Şekil 11.2. Form Yüklendiğinde Çalışan Kod Satırları. 27](#_Toc458430480)

[Şekil 11.3. Güncelle Button’una Tıklandığı Anda Çalışan Kod Satırları 28](#_Toc458430480)

[Şekil 12.1. Gelir İstatistik Formu. 28](#_Toc458430480)

[Şekil 12.2. FrmGelirİstatistik\_Load. 29](#_Toc458430480)

[Şekil 12.3. CmbAy\_Selected Index Changed. 30](#_Toc458430480)

[Şekil 13.1. Yönetici İşlemleri Formu. 30](#_Toc458430480)

[Şekil 13.2. Kaydet Button’u Kod Satırları 31](#_Toc458430480)

[Şekil 13.3. Güncelle,Sil Button’ları Ve DataGridView\_CellClick Kod Satırları 31](#_Toc458430480)

[Şekil 14.1. Personel İşlemleri Formu. 32](#_Toc458430480)

[Şekil 14.2. Kaydet Ve Sil Button’ları Kod Satırları 33](#_Toc458430480)

[Şekil 14.3. Güncelle Button’u Ve DataGridView\_CellClick Kod Satırları 34](#_Toc458430480)

[Şekil 15.1. Not Ekle Formu. 34](#_Toc458430480)

[Şekil 15.2. Kaydet Button’u Kod Satırları 35](#_Toc458430480)

[Şekil 16.1. Öğrenci Rapor Formu. 36](#_Toc458430480)

[Şekil 16.2. Report Wizard Ekleme. 36](#_Toc458430480)

[Şekil 16.3. Report Wizard 2.Aşama. 37](#_Toc458430480)

[Şekil 16.4. Report Wizard 3.Aşama. 38](#_Toc458430480)

[Şekil 16.5. Report Wizard 4.Aşama. 38](#_Toc458430480)

[Şekil 16.6. Report Wizard 5.Aşama. 39](#_Toc458430480)

[Şekil 16.7. Report Wizard Arayüzü. 39](#_Toc458430480)

ÖZET

**YURT KAYIT SİSTEMİ**

Metehan Ceyhan

Düzce Üniversitesi

Teknoloji Fakültesi

Bilgisayar Mühendisliği Bölümü

Lisans Bitirme Tezi

Danışman: Dr.Ögr Üyesi Serdar Biroğul

Ocak 2021, 58 sayfa

Günümüzde, otomasyon projeleri gittikçe önem kazanmaya başlamıştır. Bu zamanlar da, küçük işletmeler iş akışı içinde verilerini tutabilecekleri, yeni veri girebilecekleri ve hızlı bir şekilde verilerini güncelleyip, silebilecekleri, hızlı, anlaşılabilir ve işlevsel programlara ihtiyaç duymaktadırlar. Ben de bu sektör de özel öğrenci yurtları için bir proje geliştirmeyi amaçladım. Bu çalışmayla, bir özel yurdun öğrenci kaydı yapabileceği, öğrencinin bilgilerini saklayabileceği, yurdun gelir ve gider durumunun bilgilerinin saklanıp işlenebileceği, yeni personel kaydı yapılabileceği ve bilgilerinin saklanabileceği, tutulan her veri üzerinde güncelleme, silme işlemlerinin yapılabileceği bir sistem oluşturulmuştur.

**Anahtar sözcükler:** Veri,Otomasyon,İşlem

ABSTRACT

**Dormitory Registration System**

Metehan Ceyhan

Düzce University

Technology Faculty, Department of Computer Engineering

Undergraduate Graduation Thesis

Supervisor:Dr.Faculty Member Serdar Biroğul

January 2021, 58 pages

Today, automation projects are becoming increasingly important. At this time, small businesses need fast, understandable and functional programs within the workflow where they can keep their data, enter new data, and quickly update, delete, and delete their data. I aimed to develop a project for private student dormitories in this sector. With this study, a system has been established where a private dormitory can register students, store the student's information, store and process the information of the income and expense status of the dormitory, register new personnel and store their information, update and delete every data kept.

**Keywords:** Data,Automation,Function

# GİRİŞ

Günümüz dünyasında masaüstü uygulamaları yazılım sektöründe önemli bir değere sahiptir. Bunun sebebi devamlı yeni işletmelerin açılmasıdır. Ve bu işletmelerin dijital dünyada yerlerini alabilmesi için en azından temel seviyede verilerini kayıt edip, işleyebilecek bir sisteme sahip olmaları gerekmektedir. Bu sebeple yazılım şirketlerine sıkça masaüstü uygulama talepleri gelmektedir. Bu nedenle bazı yazılım şirketleri direkt otomasyon ve danışmanlık şirketi olarak sektörde hizmet vermektedir. Çoğu yazılım şirketi ise bünyesinde otomasyon uzmanları bulundurup hizmet verdikleri seçenekler arasında masaüstü uygulamaları bölümünü bulundururlar.

1.1 PROJENİN TANITIMI

Yurt Kayıt Sistemi’nin bir bilgisayarda çalıştırılabilmesi için bilgisayar da MSSQL veritabanının kurulması gerekmektedir. Kurulduktan sonra YurtKayit veritabanının .ldf ve .mdf uzantılı dosyalarının Yerel Disk(C:)>Program Files>Microsoft Sql Server>MSSQL 14.SQLEEXPRESS(sürüme göre klasör adı degişebilir.)>MSSQL>DATA dizinine yapıştırılması gerekmektedir. Ardından MSSQL Server Management Studio’ya girilip, Databases kısmına sağ tuş tıklanıp ATTACH seçeneğini seçmek gerekiyor. Ardından önümüze açılan pencerede YurtKayit veritabanımızı seçip tamam seçeneğine tıklandıktan sonra veritabanı bilgisayara kurulmuş oluyor. Bu işlem ardından proje kullanılabilir duruma gelmiştir. Uygulamayı kullanabilmek için veritabanında Admin tablosuna manuel olarak kullanıcı adı ve şifre kaydetmemiz gerekmektedir. Ardından uygulama çalıştığında bu kullanıcı adı ve şifre ile giriş yapılır. Ve giriş yapıldığında ekrana, kayıtlı öğrencilerin listelendiği ve diğer sayfalara geçiş yapabilmek için bir arayüz olan programın ana sayfası gelir. Ana sayfa da bulunan ContextMenuStrip yapısından diğer sayfalara erişim sağlanır. ContextMenuStrip’de 7 farklı bölüm vardır. Öğrenci işlemleri yapılabilmesi için Öğrenciler bölümü, yurda öğrenci kaydı yapılırken okudukları bölümleri otomatik combobax’tan seçilebilmesi için bölüm ekleme ve bölüm düzenle bölümü bulunan Bölümlerler bölümü, Öğrenciden ödeme alabilmek için ve alınan ödemenin listesini tutabilmek için Ödemeler bölümü, yurdun gider bilgilerini kayıt altına almak ve listelemek için Giderler bölümü, Yurdun gelir düzeyini istatistik olarak görebilmek için İstatistikler bölümü, Sistemin kullanıcıları için gerekli işlemlerin yapılacağı Yönetici bölümü ve kullanıcıya kolaylık sağlamak için çeşitli bölümlerin olduğu Erişim Kolaylığı bölümü vardır.

* 1. PROJENİN AMACI

Klasik yazılım geliştirme tekniği kullanarak temiz yazılım ilkesi doğrultusunda, günümüz teknolojisine uygun bir uygulama ortaya koyabilmek.

Projenin bir diğer amacı ise, sektör de özel öğrenci yurtlarının yazılım şirketlerinden talep ettikleri sade, anlaşılır ve işlevselliği yüksek bir program bu ihtiyacını karşılayabilmek ve örnek bir prototip ortaya koyabilmek.

* 1. PROJENİN GELİŞTİRİLDİĞİ ARAÇLAR
     1. VİSUAL STUDİO

Microsoft tarafından geliştirilen bir tümleşik geliştirme ortamıdır (IDE). [Microsoft Windows](https://tr.wikipedia.org/wiki/Microsoft_Windows), [Windows Mobile](https://tr.wikipedia.org/wiki/Windows_Mobile), [Windows CE](https://tr.wikipedia.org/wiki/Windows_CE), [.NET Framework](https://tr.wikipedia.org/wiki/.NET_Framework), [.NET Compact Framework](https://tr.wikipedia.org/wiki/.NET_Compact_Framework) ve [Microsoft Silverlight](https://tr.wikipedia.org/wiki/Microsoft_Silverlight) tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve Windows Forms uygulamaları, web siteleri, web uygulamaları ve web servisleri ile birlikte konsol ve grafiksel kullanıcı arayüzü uygulamaları geliştirmek için kullanılır. Visual Studio [IntelliSense](https://tr.wikipedia.org/wiki/IntelliSense)'in yanı sıra "code refactoring" destekleyen bir kod editörü içerir. Entegre hata ayıklayıcı, hem kaynak-seviyesinde hem de makine-seviyesinde çalışır. Diğer yerleşik araçlar, [GUI](https://tr.wikipedia.org/wiki/GUI) uygulamaları, web tasarımcısı, sınıf tasarımcısı ve [veritabanı](https://tr.wikipedia.org/wiki/Veritaban%C4%B1) şema tasarımcısı yaratabilmek için bir form tasarımcısı içerir. Hemen hemen her işlevsellik düzeyinde dahil olmak üzere, kaynak kontrol sistemleri için destek (Subversion ve Visual SourceSafe gibi) sunan eklentileri kabul eder.

* + 1. SQL SERVER MANAGEMENT STUDİO

Microsoft SQL Server, verilerin güvenle ve bütünlük içerisinde depolanmasını ve aynı anda birden fazla kullanıcı tarafından erişilmesini sağlayan kurumsal çaplı bir ilişkisel veri tabanı yönetim sistemidir. Birbiriyle ilişkili verilerin sistematik bir şekilde kaydedilmesini ve bu verilerden beslenen uygulamalar tarafından ihtiyaç anında kullanılmasına olanak sağlayarak işletmelerin BT altyapılarındaki veri aktarımını sorunsuz bir şekilde yönetmelerini sağlar.

SQL Server, işletmeler için kritik önem taşıyan verileri şifreleme, verilere erişim sağlayan kişileri gözlemleme ve erişim kısıtlamaları tanımlama gibi güvenlik özellikleri sayesinde kullanıcılara kapsamlı bir denetim kapasitesi sunar. Yüksek Kullanılabilirlik (High Availability) ve olağanüstü durum kurtarma özelliklerini tek bir çözümde birleştiren SQL Server, hızlı yük devretme ve yük dengeleme desteği ile aksaklık süresini en aza indirir ve güvenli veri akışının optimize edilmesini sağlar.

Bilişim, sağlık ve finans gibi çeşitli sektörlerde önde gelen şirketler tarafından tercih edilen Microsoft SQL Server, her tip kullanıcının ihtiyaçlarına uyum sağlayabilen ölçeklenebilen bir yapıya sahiptir.

1. GENEL BİLGİLER

2.1 Yazılım

Yazılım, elektronik aygıtların belirli bir işi yapmasını sağlayan programların tümüne verilen isimdir. Bir başka deyişle var olan bir problemi çözmek amacıyla bilgisayar dili kullanılarak oluşturulmuş anlamlı anlatımlar bütünüdür. Kelime işlemci programları, bilgisayarınız açıldığında CD sürücünüzü, sabit sürücüleri, RAM'i tanıyan BIOS; işletim sistemi, web tarayıcınız, virüslerin kendileri, antivirüs programları hep birer yazılımdır.

2.1.1. Bir Yazılımda Olması Gereken Temel Özellikler

Doğruluk: Yazılımın belirtilmiş ihtiyaçlarını karşılamasıdır. Güvenilirlik: Gerekli işlevi ne hassaslıkla yerine getireceği beklentisidir.

Verimlilik: İşlevin gerçekleştirilmesi için kullanılması gereken bilgisayar kaynakları ve kod miktarıdır.

Güvenlik(Bütünlük): Yazılım ve bilgilerine, istenmeyen insanlarca ulaşımın ne derece engellenebildiğidir.

Kullanılabilirlik: Programın öğrenilmesi, çalıştırılması, girdi hazırlama ve çıktı yorumlama işlemlerinin kolaylık derecesidir. Hata bulma kolaylığı: Hatanın yerini bulma ve düzeltme kolaylığıdır.

Esneklik: Yazılımda değişiklik yapma kolaylığıdır.

Taşınabilirlilik: Programın farklı donanımlarda ve yazılım sistemi ortamlarında kullanılmasıdır.

Tekrar kullanılabilirlik: Yazılım tamamının ya da bir bölümünün farklı bir uygulamada kullanılabilmesidir.

Birlikte çalışabilirlik: Bir yazılım sisteminin diğerleri ile bağlantı sağlaması kolaylığıdır.

2.1.2. Yazılım Çeşitleri

Bilgisayar yazılımları genel olarak 2 ana grupta incelenebilir.

Sistem Yazılımları (System Software) : Bilgisayarı yöneten, denetleyen, kontrol eden yazılımlardır.

Örneğin: Linux, Pardus, Windows vb.

Uygulama Yazılımları(Application Software): Belli bir alana ve uygulamaya ilişkin olarak kullanıcılar için geliştirilmiş yazılımlardır.

Örneğin: Kütüphane programı, web hazırlama araçları,programlama araçları vb.

Bütün sistem programları içinde en temel yazılım işletim sistemidir ki, bilgisayarın bütün donanım ve yazılım kaynaklarını kontrol ettiği gibi kullanıcılara ait uygulama yazılımlarının da çalıştırılmalarını ve denetlenmelerini sağlar.

2.2. Programlama Araçları

2.2.1. Programlama Dili

İster genel ister özel amaçlı olsun tüm uygulama ve sistem yazılımları programlama dilleriyle yazılır. Bir programlama dili, insanların bilgisayara çeşitli işlemler yaptırmasına imkân veren her türlü sembol, karakter ve kurallar grubudur. Programlama dilleri insanlarla bilgisayarlar arasında tercümanlık görevi yapar. Programlama dilleri, bilgisayara neyi, ne zaman, nasıl yapacağını belirten deyim ve komutlar içerir.

Bir programlama dili şunlardan oluşur.

Genel komutlar: Programlama dilinin anlayacağı komutlardır.

Gelişmiş komutlar: Genel komutları kullanarak oluşturulmuş komutlardır.

API komutları: İşletim sisteminin sunduğu özellikleri kullanan komutlardır.

Derleyici komutları: Komut içinde çalışmayıp derleme esnasında alınan bilgilere göre derleme yapılmasını sağlar.

Aktif nesneler: Button, Menü, Gösterge çubuğu ve Tabpanel gibi bileşenlerin genel adıdır.

1. PROJE

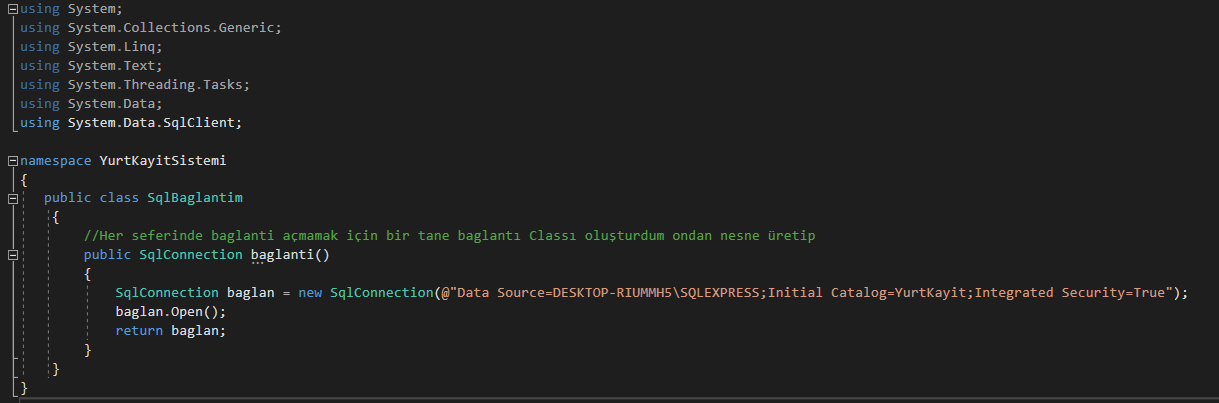
Proje de çalıştırıldığın da ekrana ilk başta Admin Giriş formu gelmektedir.



Şekil 2.1 Admin Giriş Formu Görüntüsü

Projedeki tüm formların MinimizeBox ve MaximizeBox özelliği false olarak seçilmiştir.

Projedeki tüm formların AutoSizeMode özelliği GrowAndShrink olarak seçilmiştir.



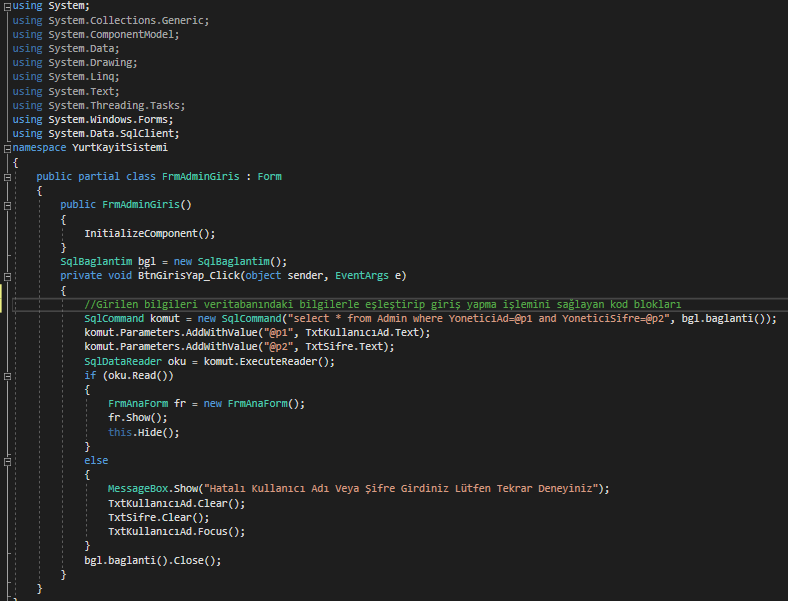
Şekil 2.2 SqlBaglantim Classı

Proje ilk iş olarak her form da ayrı ayrı sql bağlantısı açmamak için bir tane baglanti classı oluşturup formda classtan nesne üretip bağlantıyı oluşturmak istedim.

Şekil 2.1 deki formdaki kodlar ve çalışma mantığı:

using System.Data.SqlClient;

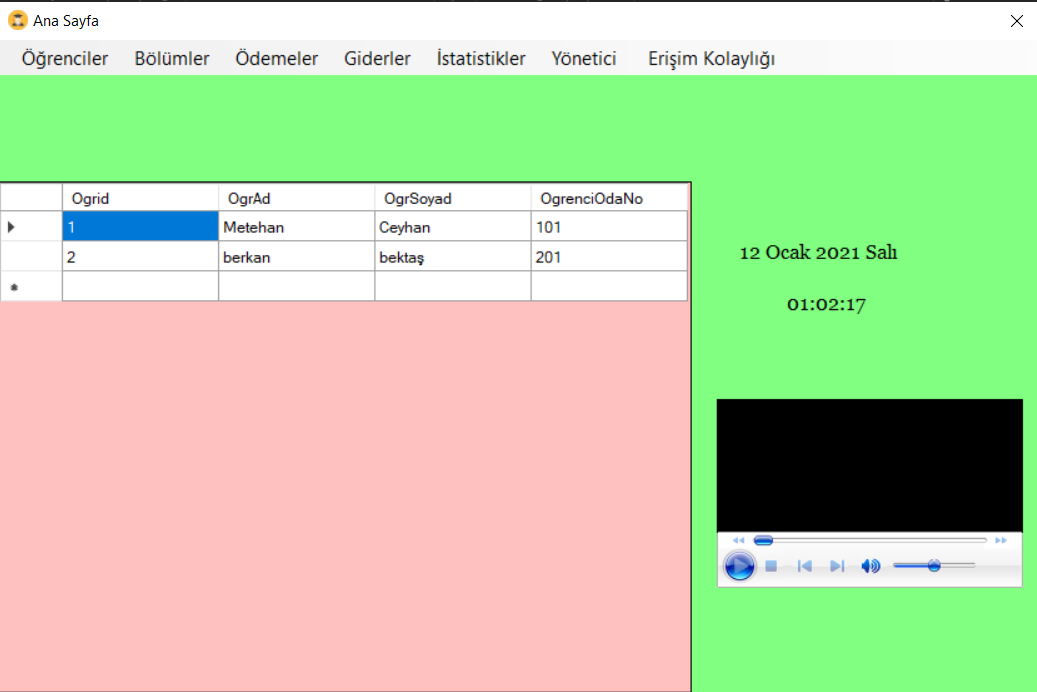
Formda database bağlantısını sağlayabilmek için yukarıdaki kütüphanemizi ekliyoruz.



Şekil 2.3 Giriş Yap Buttonu Kod Blokları

İlk olarak SqlCommand nesnesinden komut adlı nesnemizi üretip parametre olarak girilen değerlere göre Admin tablosundan veri çeken select sorgusu ve bağlantı fonksiyonumuzu veriyoruz. Daha sonra @p parametrelerini veritabanındaki değerlerimize eşitliyoruz. Oku nesnesini oluşturuyoruz ve okuyacağı satırları içine atıyoruz. Eğer okunan satır var ise kullanıcıyı Ana Form’a yönlendiriyoruz yok ise Hata mesajı verip textboxları temizliyoruz. En son bağlantıyı kapatıyoruz.

Admin Giriş sayfasında Giriş Yap button’una tıkladıktan sonra gelen ana sayfa şekilde gösterilmiştir.



Şekil 3.1 Ana Sayfa Formu

ContextMenuStrip item’larından her hangi birine tıklandığı yeni bir formun açılmasını sağlayan kod satırları

private void bölümGüncellemeToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmBolumler fr = new FrmBolumler();

fr.Show();

}

//Ekran da anlık olarak saati göstermemizi sağlayan kod blokları

private void timer1\_Tick(object sender, EventArgs e)

{

label1.Text = DateTime.Now.ToLongDateString();

label2.Text = DateTime.Now.ToLongTimeString();

}

private void hesapMakinesiToolStripMenuItem\_Click(object sender, EventArgs e)

{

System.Diagnostics.Process.Start("Calc.exe"); //Tıklandığı anda hesap makinesini getiren kod satırı

}

private void paintToolStripMenuItem\_Click(object sender, EventArgs e)

{

System.Diagnostics.Process.Start("MsPaint.exe");//Tıklandığı anda paint’i getiren kod satırı

}

//Tıklandığı anda üç tane farklı radyonun anlık olarak yayınını çeken kod satırı

private void radyo1ToolStripMenuItem\_Click(object sender, EventArgs e)

{

axWindowsMediaPlayer1.URL = "http://cast1.taksim.fm:8000";

}

private void radyo2ToolStripMenuItem\_Click(object sender, EventArgs e)

{

axWindowsMediaPlayer1.URL = "http://37.247.98.8/stream/166/";

}

private void radyo3ToolStripMenuItem\_Click(object sender, EventArgs e)

{

axWindowsMediaPlayer1.URL = "http://160.75.86.29:8088/ituklasikmp3 ";

}

private void öğrencToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmOgrKayit fr = new FrmOgrKayit();

fr.Show();

}

private void öğrenciListesiToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmOgrListe fr = new FrmOgrListe();

fr.Show();

}

private void öğrenciDüzenleToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmOgrListe fr = new FrmOgrListe();

fr.Show();

}

private void bölümEkleToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmBolumler fr = new FrmBolumler();

fr.Show();

}

private void ödemeAlToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmOdemeler fr = new FrmOdemeler();

fr.Show();

}

private void giderEkleToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmGider fr = new FrmGider();

fr.Show();

}

private void giderListesiToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmGiderListesi fr = new FrmGiderListesi();

fr.Show();

}

private void gelirİstatistikleriToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmGelirİstatistik fr = new FrmGelirİstatistik();

fr.Show();

}

private void şifreİşlemleriToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmYoneticiDuzenle fr = new FrmYoneticiDuzenle();

fr.Show();

}

private void personelDüzenleToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmPersonel fr = new FrmPersonel();

fr.Show();

}

private void notEkleToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmNotEkle fr = new FrmNotEkle();

fr.Show();

}

private void öğrenciToolStripMenuItem\_Click(object sender, EventArgs e)

{

FrmOgrenciRapor fr = new FrmOgrenciRapor();

fr.Show();

}

private void hakkımızdaToolStripMenuItem\_Click(object sender, EventArgs e)

{

MessageBox.Show("Bu Program Metehan Ceyhan tarafından tamamlanmıştır", "Yurt Kayit Sistemi", MessageBoxButtons.OK, MessageBoxIcon.Information); // hakkımızdaToolStripMenuItem’a tıklandığı an da ekrana mesaj gelmesini sağlayan kod bloğu

}

private void çıkışToolStripMenuItem\_Click(object sender, EventArgs e)

{

Application.Exit(); // uygulamadan çıkış yapmayı sağlar

}

private void FrmAnaForm\_Load(object sender, EventArgs e)

{

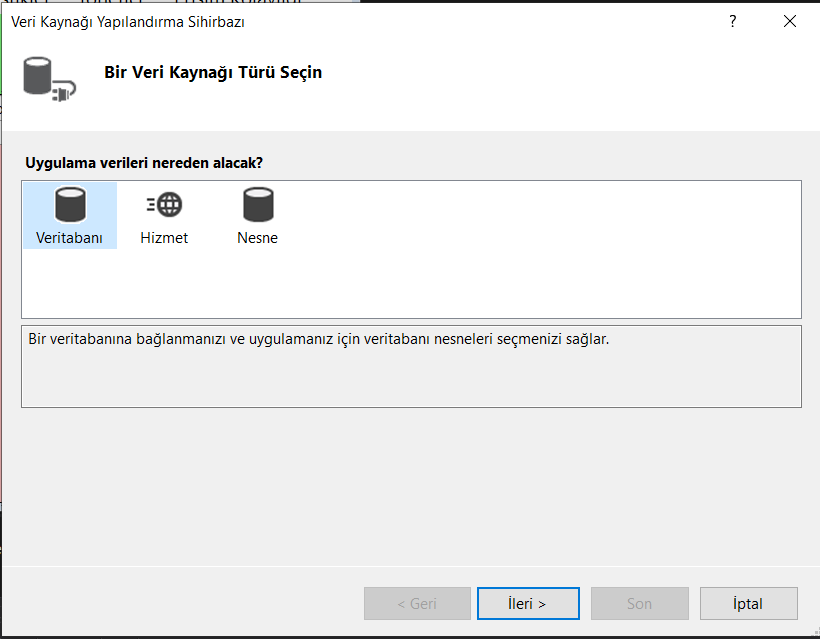
// TODO: Bu kod satırı 'yurtKayitDataSet1.Ogrenci' tablosuna veri yükler. Bunu gerektiği şekilde taşıyabilir, veya kaldırabilirsiniz.

this.ogrenciTableAdapter.Fill(this.yurtKayitDataSet1.Ogrenci);

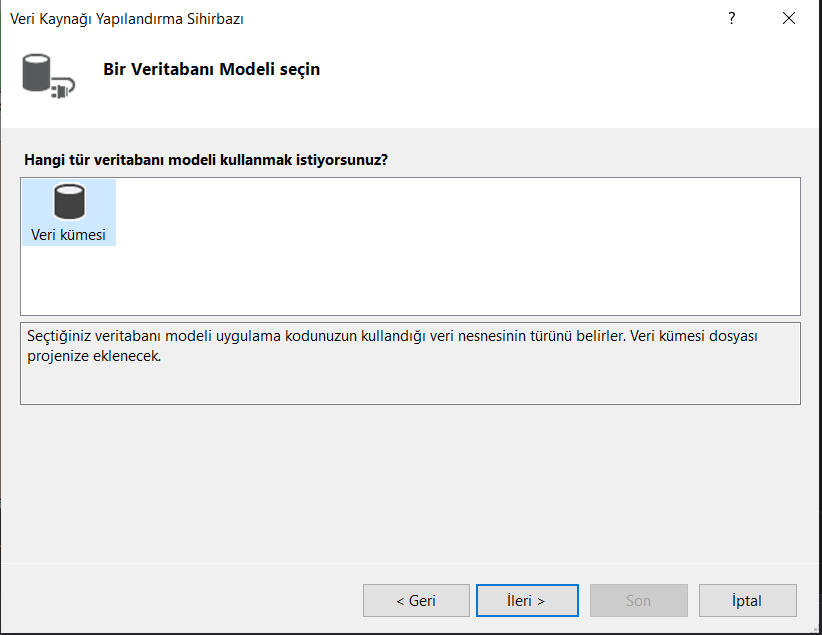
timer1.Start();

}

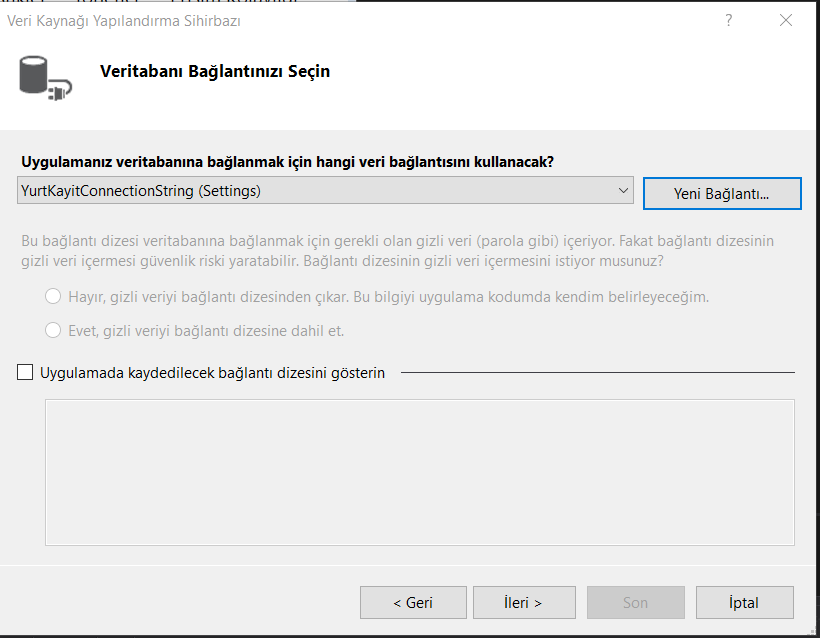
Form yüklendiği anda DataGridView’a veritabanından verileri listememizi sağlayan aşamalar.



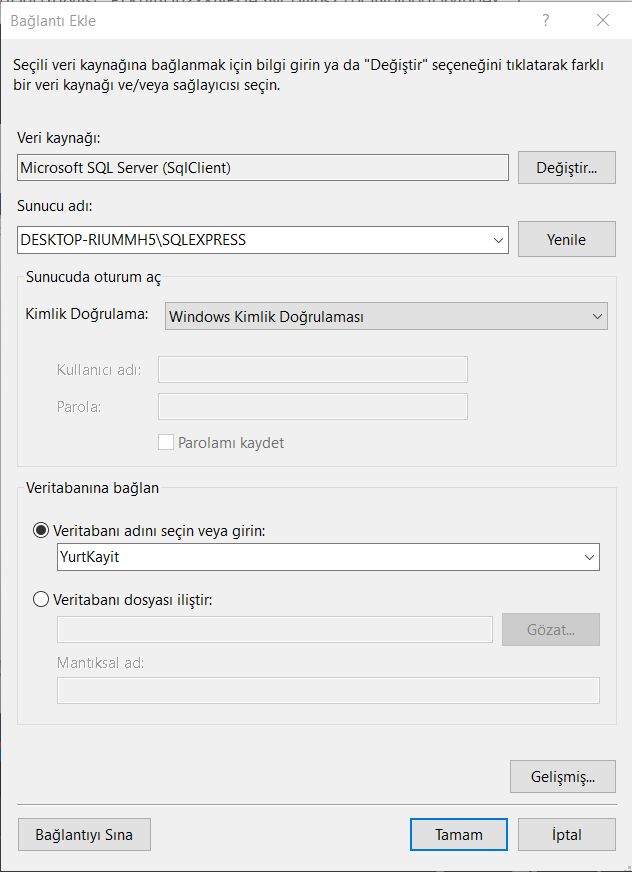
Şekil 3.2 Dataset Oluşturma



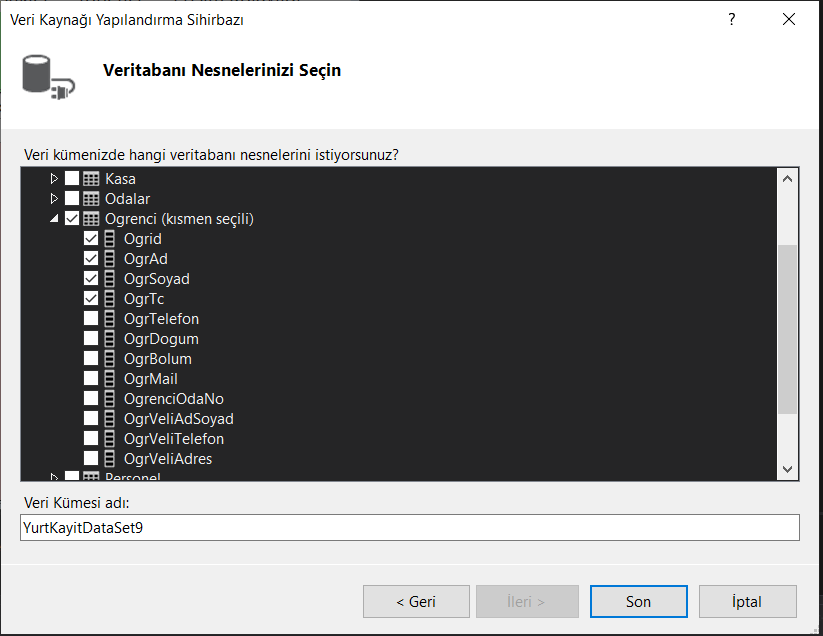
Şekil 3.3 Dataset Oluşturma 2. Adım



Şekil 3.4 Dataset Oluşturma 3. Adım



Şekil 3.5 Dataset Oluşturma 4. Adım

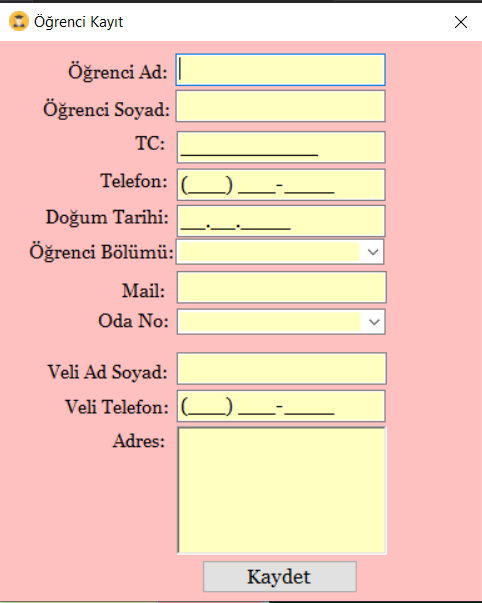


Şekil 3.6 Dataset Oluşturma 5. Adım

Tablolarımızdan ihtiyacımız olan değerleri işaretliyoruz ve son button’una tıkladıktan sonra dataset’imizi oluşturmuş oluyoruz.

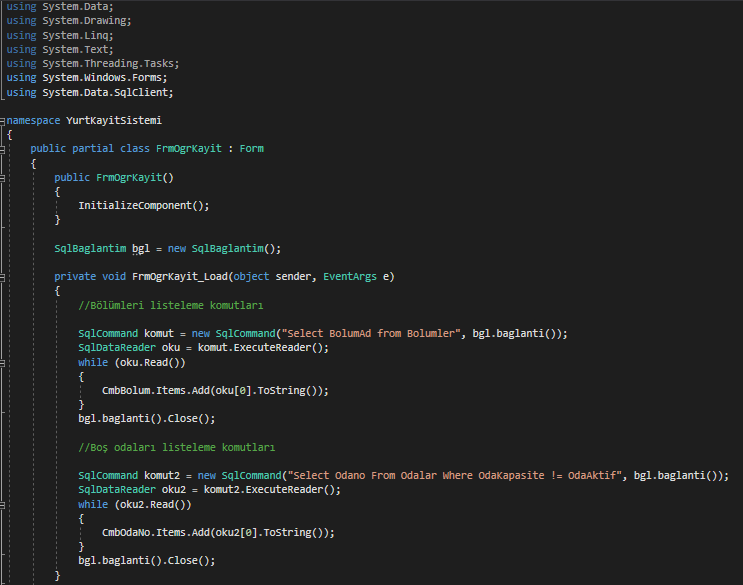
Bu projede 8 tane dataset oluşturulmuştur hepsi yukarıdaki şekillerde anlatılan şekilde oluşturulmuştur.

Ana sayfa da Öğrenciler bölümünde Öğrenci Ekle kısmına tıklandığı zaman karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 4.1 Öğrenci Kayıt Formu

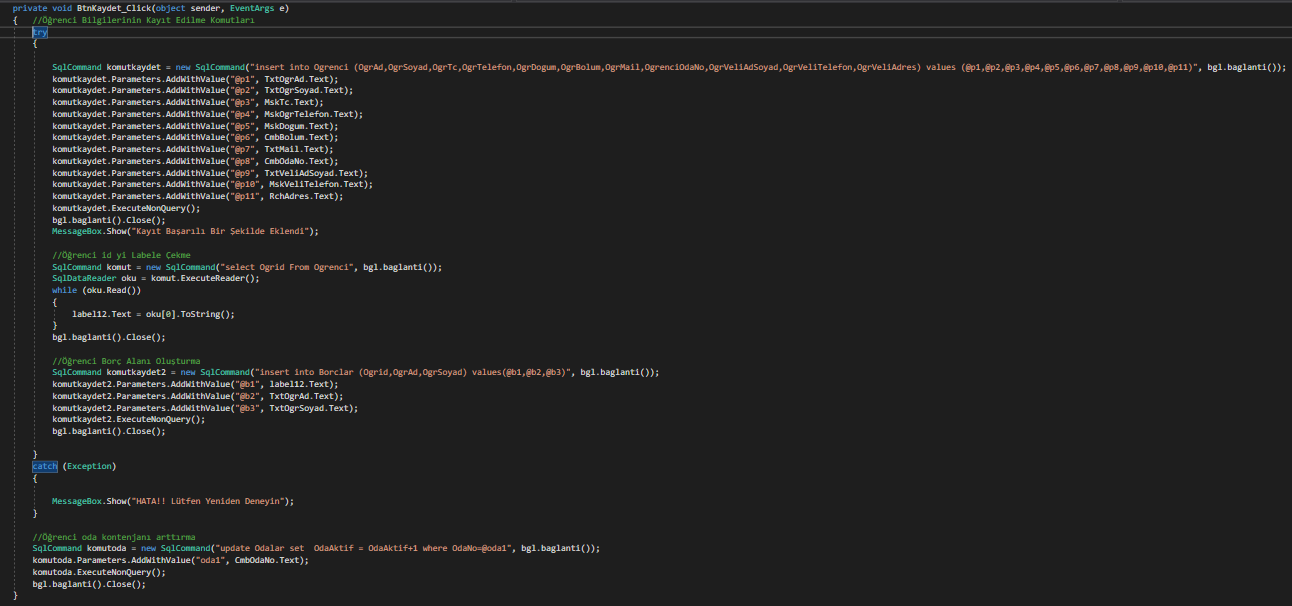
Form yüklendiği anda çalışan kod satırları aşağıdaki görsel de gösterilmiştir.



Şekil 4.2 Öğrenci Kayıt Formu Yüklendiği Anda Çalışan Kod Satırları

Form yüklendiği anda komut nesnemizi oluşturup BolumAd tablosundaki verileri listeleyen select sorgunu ve Sqlbaglantim sınıfından oluşturduğumuz bgl nesnesinin baglanti fonksiyonunu parametre olarak veriyoruz. Daha sonra oluşturduğumuz oku nesnesi ile tablodaki değerleri okutuyoruz ve okunduğu sürece combobox’a verileri ekliyoruz. Ve bölüm combobox’ında veritabanına eklenmiş bölümleri listeliyoruz. Daha sonra veritabanındaki boş odaları listelemek için aynı işlemleri OdaNo combobox’ı için uyguluyoruz. İşlemler arasındaki fark, bu sefer verileri Odalar tablosundan OdaKapasitesi boş olanları çekiyoruz. OdaKapasite değeri OdaAktif’e eşit ise oda dolu anlamına gelir bu sebeple combobox’ta listelenmez.

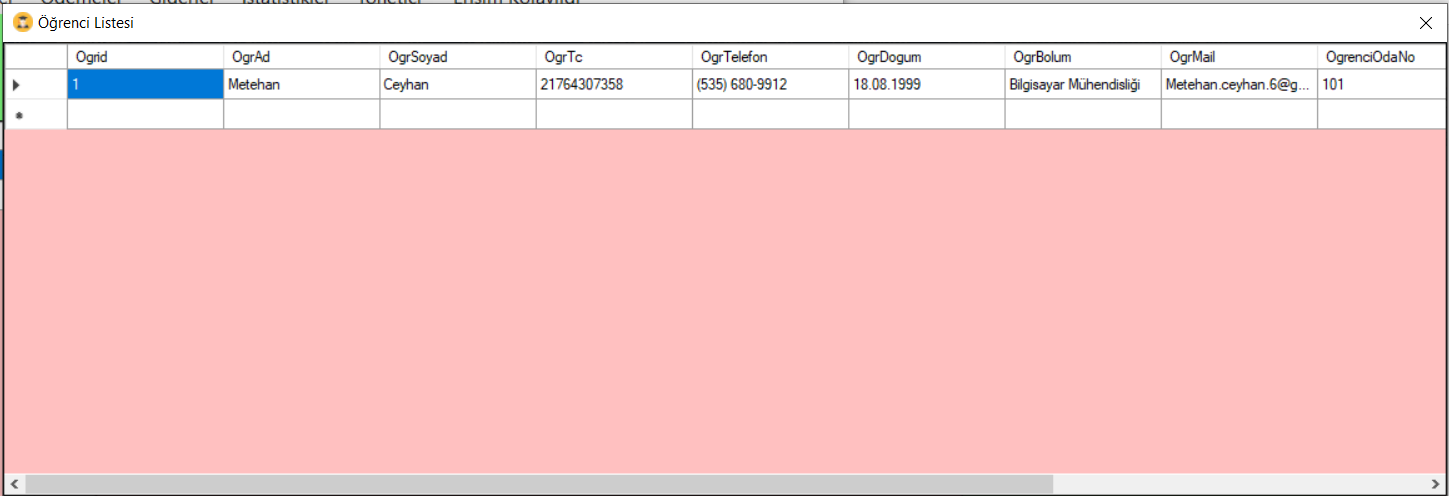
Veriler girildikten sonra kaydet button’una tıklandığı anda çalışan kod satırları aşağıdaki şekilde gösterilmiştir.



Şekil 4.3 Kaydet Button’u Kod Satırları

İlk önce komutkaydet nesnemizi oluşturup parametre olarak Control’lere girilen verileri veritabanına kayıt işlemi yapan insert sorgusunu ve Sqlbaglantim sınıfının baglanti fonksiyonunu veriyoruz. Daha sonra @p parametrelerini veritabanındaki sütunlara eşitliyoruz. Ardından veritabanında sorgumuzu çalıştırıyoruz ve veritabanı bağlantımızı kapatıyoruz. Ekrana başarılı giriş mesajı veriyoruz. Kaydet button’una tıklandığı anda öğrenci eklendiği için otomatik olarak Borclar tablosunda da öğrencinin kaydının oluşturulması gerekmektedir. Ben öğrencinin yıllık ödeyeceği miktarı veritabanında manuel olarak 3000 tl olarak belirledim, bu isteğe bağlı değiştirilebilir. Borçlar tablosunda kayıt oluşturmak için yine yukarıda yaptığımız işlemleri tekrarlıyoruz. Bu sefer farklı olarak formumuzda tuttuğumuz label12’ ye öğrencinin id sini atıyoruz. Ardından yukarıdaki işlemleri yapıyoruz. Borçlar tablosuna Ogrid,OgrAd,OgrAdsoyad ı ekliyoruz. OgrKalanBorc sütunu ise default olarak 3000 olarak gelmektedir. Daha sonra bir öğrenci eklediğimiz için eklediğimiz odanın kontenjanını arttırmamız gerekiyor. Bunun içinde OdaAktif değerini bir arttıran yani güncelleyen update sorgumuzu yapıp çalıştırıyoruz.

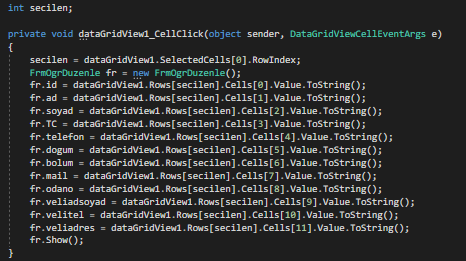
Ana sayfa da Öğrenciler bölümünde Öğrencileri Listele kısmına tıklandığı zaman karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 5.1 Öğrenci Listele Formu

Öğrencileri listelemek için Dataset üretiyoruz Ogrenci tablomuzdaki tüm sütunları seçiyoruz.

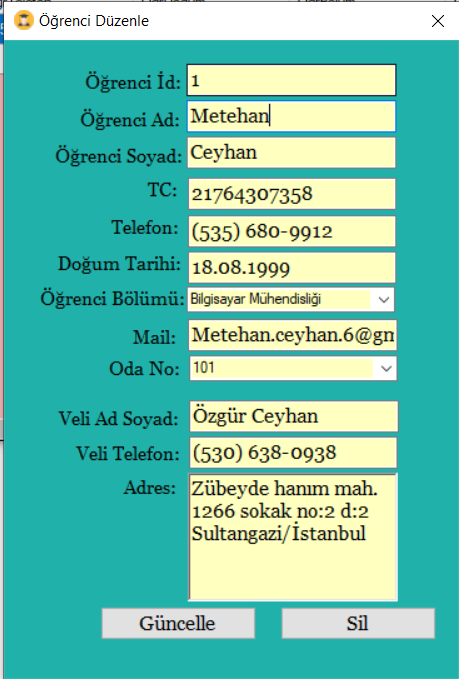
DataGridView’e tıklandığında ise öğrenci bilgilerini güncelleyebilmek veya silebilmek için yeni bir form açılmaktadır.



Şekil 5.2 DataGridView’a Tıklandığı Anda Çalışan Kod Satırları

Tıklandığı anda FrmOgrDuzenle formundan nesne oluşturuyoruz. DataGridView’ın satırlarındaki değerleri FrmOgrDuzenle de oluşturduğumuz değişkenlere atıyoruz. Sonrada FrmOgrDuzenle formunu ekrana getiriyoruz.

DataGridView satırındaki verilere tıklandığı anda gelen sayfa aşağıda gösterilmiştir.



Şekil 6.1 Öğrenci Düzenle Formu

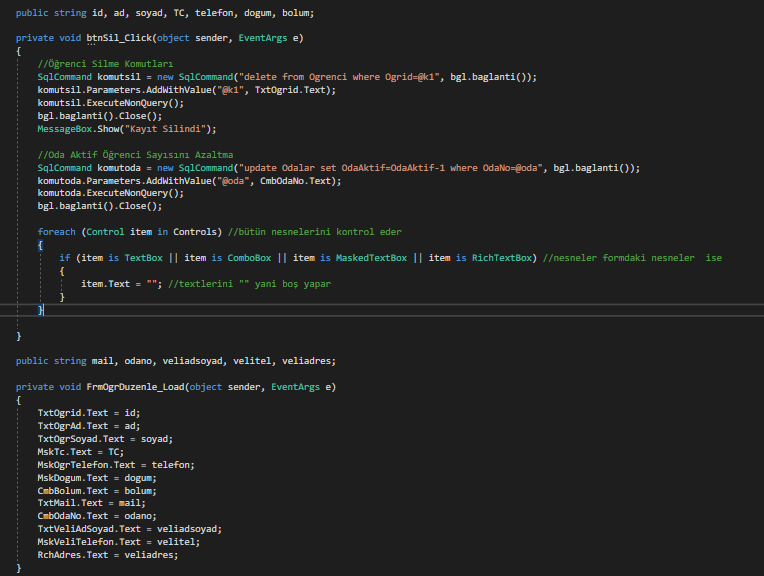
Güncelle button’una tıklandığında çalışan kod satırları aşağıda gösterilmiştir.



Şekil 6.2 Güncelle Button’u Kod Satırları

Komut nesnemizi oluşturup Ögrenci tablosundaki verileri Ogrid’ye göre güncelleyen update sorgsunu ve veritabanı bağlantımızı atıyoruz. Daha sonra tek tek @p parametrelerini veritabanındaki sütunlara eşitliyoruz. Ve veritabanı sorgumuzu çalıştıyoruz. Ekrana kayıt güncellendi mesajını veriyoruz. Eğer kayıt güncellenemezse ekrana hata mesajı verdiriyoruz.

Sil button’una tıklandığında ve form yüklendiğinde çalışan kod satırları aşağıda gösterilmiştir.



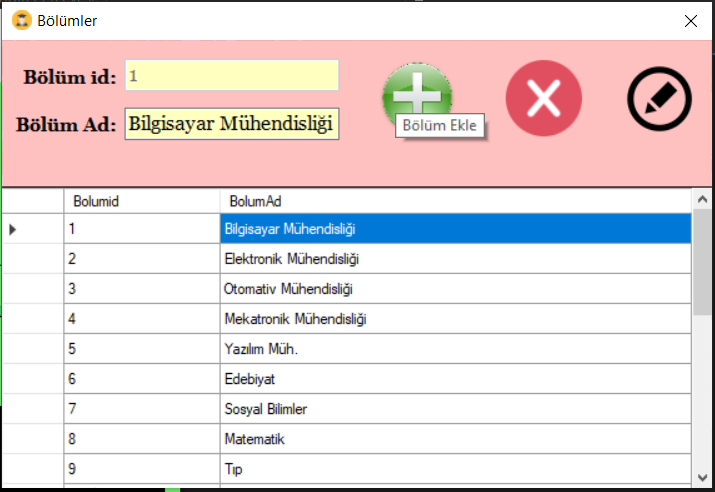
Şekil 6.3 Sil Button’u Ve FormLoad Kod Satırları

KomutSil nesnesi oluşturulup içine Ögrenci tablosundaki Ogrid’ye göre veriyi silen delete sorgusu ve sql bağlantısı atılıyor. Daha sonra @k parametresi veritabanındaki değere eşitleniyor. Daha sonra veritabanı sorgusu çalıştırılıyor ve veritabanı bağlantısı kapatılıyor. Öğrenci silineceği için bulunduğu odadaki sayısının da azaltılması lazım bunun için update sorgusu ile Odalar tablosundaki OdaAktif durumunu bir azaltıyoruz.

Bu işlemlerden sonra kontrollerdeki verileri temizliyoruz.

FrmOgrDüzenle formunda veritabanında Ögrenciler tablosu için kullanacağımız değişkenlerimizi oluşturuyoruz. Daha sonra form yüklendiği anda kontrollerin text’lerine oluşturduğumuz değişkenleri atıyoruz.

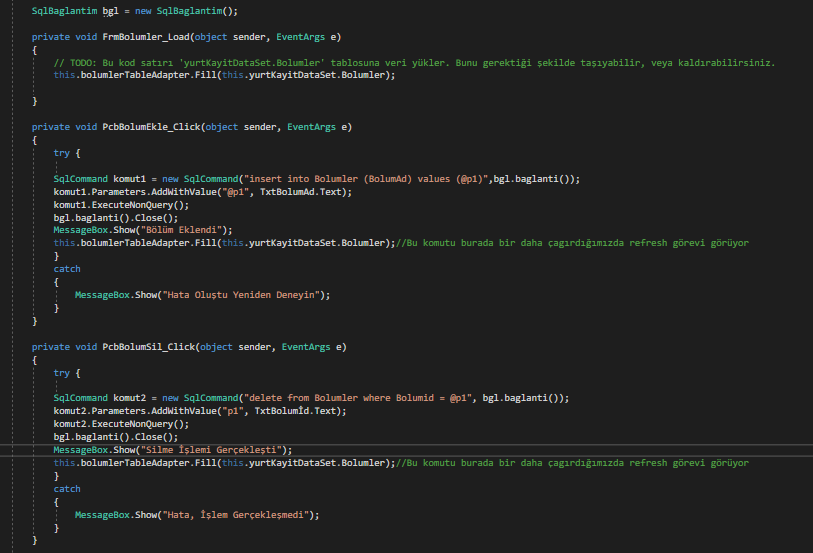
Ana sayfa da bölümler kısmında Bölüm Ekle ve Bölüm Düzenle kısmına tıklandığında da aynı form açılmaktadır. İki işlemi tek bir form da hallediyoruz. Bölümler formu aşağıda gösterilmiştir.



Şekil 7.1 Bölümler Formu

Formda tooltip kullanılmıştır ekle, sil ve güncelle simgelerin üzerine mause ile gelindiğinde yapılacak işlemin adı simgelerin üzerine yazdırılır.

Verileri DataGridView’da listeleyebilmek için Dataset oluşturulur. Bolumler tablosundaki sütunlar seçilir.



Şekil 7.2 Bölüm Ekle Ve Bölüm Sil Kod Satırları

Bölüm ekle Picture Box’ına tıklandığı zaman komut1 nesnesi oluşturulup içine Bolumler tablosuna BolumAd’ı ekleyen insert sorgusu ve sql bağlantısı atılır. @p parametresi veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana bölüm eklendi mesajı gönderilir. DataGridView refresh edilir. Aksi durumda ekrana hata mesajı verilir.

Bölüm sil Picture Box’ına tıklandığı zaman komut2 nesnesi oluşturulup içine Bolumler tablosundan Bolumid’ye göre verileri silen delete sorgusu ve sql bağlantısı atılır. @p parametresi veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana silme işlemi gerçekleşti mesajı gönderilir. DataGridView refresh edilir. Aksi durumda ekrana hata mesajı verilir.

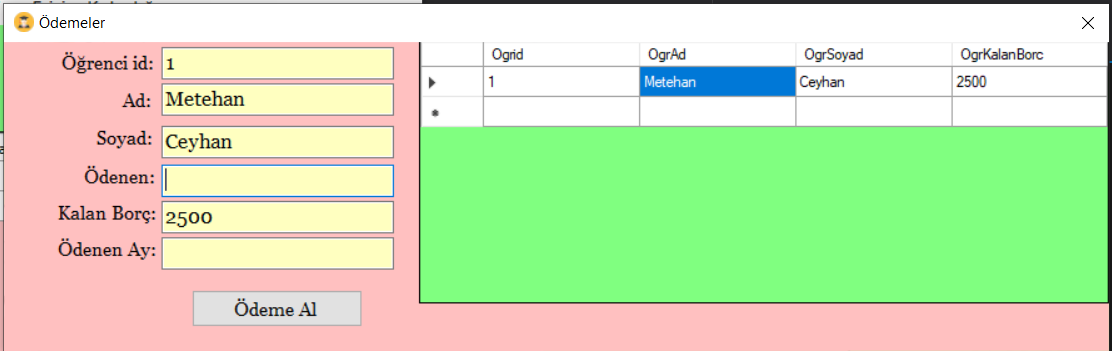


Şekil 7.3 Bölüm Güncelle Ve DataGridView\_CellClick Kod Satırları

Verilerde düzenleme yapılabilmesi DataGrid’deki verilere tıklandığı anda textboxlara gelmeleri gerekmektedir. Bu sebeple DataGride tıklandığı anda secilen,id,bölümad değişkenleri oluşturulur ve DataGridView satırlarındaki değerler değişkenlere atılır. Daha sonra değişkenler textbox’ların text’ine yazdırılır.

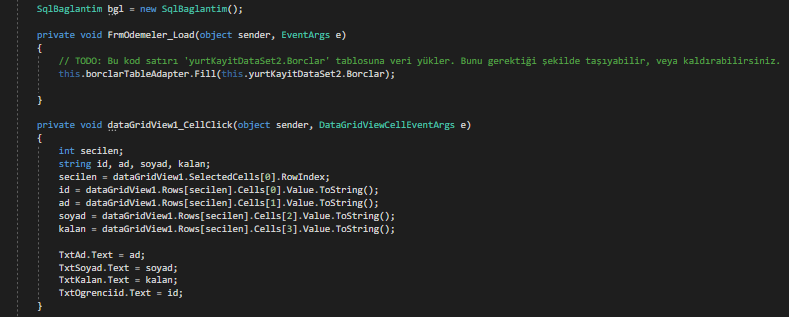
Bölüm Güncelle Picture Box’ına tıklandığı zaman komut3 nesnesi oluşturulup içine Bolumler tablosundan BolumAd sütununu Bolumid’ye göre güncelleyen update sorgusu ve sql bağlantısı atılır. @p parametreleri veritabanındaki değerlere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana Ekrana güncelle gerçekleşti mesajı gönderilir. DataGridView refresh edilir. Aksi durumda ekrana hata mesajı verilir.

Ana sayfa da Ödeme Al bölümüne tıkladığımız zaman açılan sayfa aşağıda gösterilmiştir.



Şekil 8.1 Ödemeler Formu

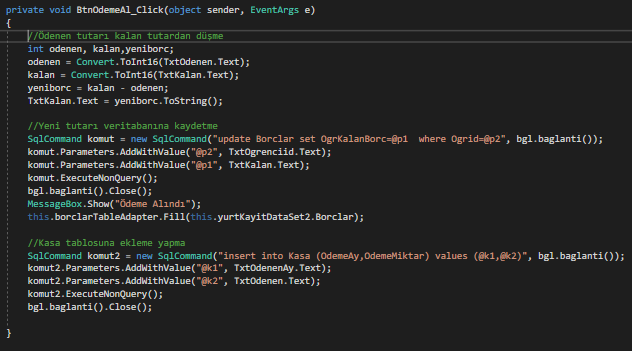
DataGridView’a kayıtları çekmek için Dataset oluşturuyoruz. Borclar tablosundaki tüm sütunları işaretliyoruz.



Şekil 8.2 DataGridView\_CellClick Durumu

Kayıtlardaki değerlere denk gelen değişkenlerimizi oluşturuyoruz. DataGridView’ın satırındaki değeleri sırasıyla değişkenlerimize atıyoruz. Değişkenlerimizi de textbox’ların text’lerine atıyoruz.

Ödeme Al button’una tıklandığı anda çalışan kod satırları aşağıda gösterilmiştir.



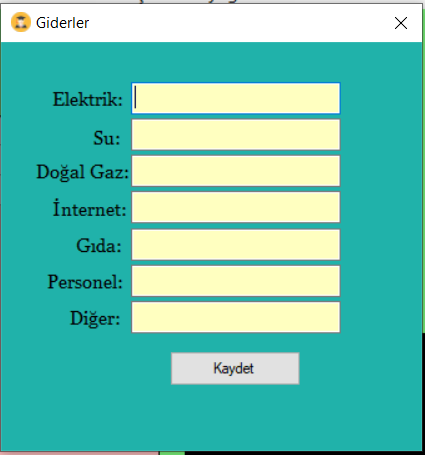
Şekil 8.3 Ödeme Al Button’u

Türü int olan odenen,kalan,yeniborc adında değişkenlerimizi oluşturuyoruz. Odenen ve kalan degişkenlerimize textbox’lardan isimlerine göre değerlerini atıyoruz. Yeniborc degişkenimize de kalan-odenen işleminde çıkan sonucu atıyoruz. Yeniborc değişkenimizi de TxtKalan textbox’ının text’ine atıyoruz.

Komut nesnesi oluşturulup içine Borclar tablosundan OgrKalanBorc ve Ogrid değerlerine göre güncelleme yapan update sorgusu ve sql bağlantısı atılmaktadır. @p parametresi veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana ödeme alındı mesajı gönderilir. DataGridView refresh edilir.

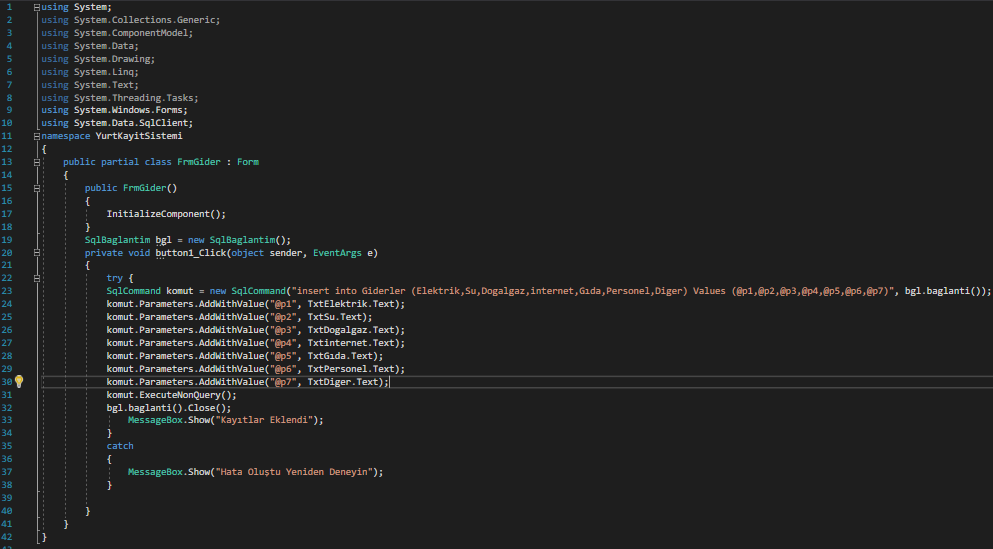
Komut2 nesnesi oluşturulup içine Kasa tablosuna ÖdemeAy ve OdemeMiktarı ekleyen insert sorgusu ve sql bağlantısı atılır. @k parametresi veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır.

Ana sayfa da Giderler bölümünde Gider Ekle kısmına tıklandığı zaman karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 9.1 Giderler Formu

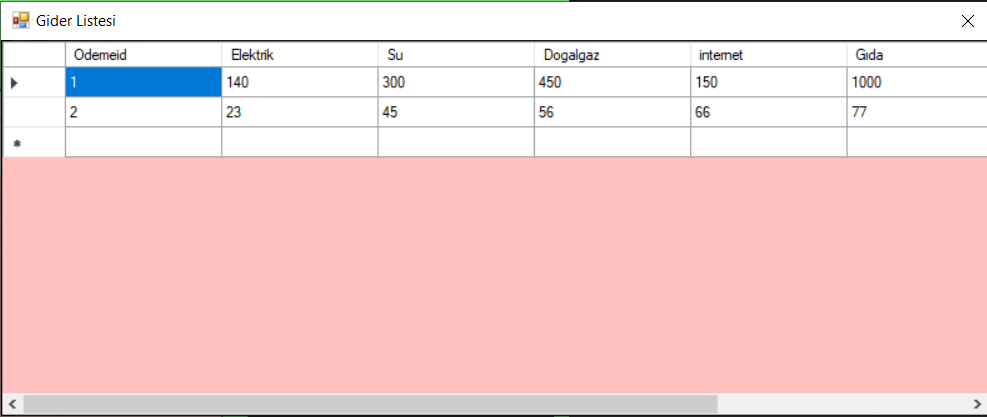
Kaydet button’una tıklandığında çalışan kod satırları aşağıda gösterilmiştir.



Şekil 9.2 Kaydet Button’u Kod Satırları

İlk önce komut nesnemizi oluşturup parametre olarak Control’lere girilen verileri veritabanına kayıt işlemi yapan insert sorgusunu ve Sqlbaglantim sınıfının baglanti fonksiyonunu veriyoruz. Daha sonra @p parametrelerini veritabanındaki sütunlara eşitliyoruz. Ardından veritabanında sorgumuzu çalıştırıyoruz ve veritabanı bağlantımızı kapatıyoruz. Ekrana başarılı giriş mesajı veriyoruz. Aksi durumda Ekrana hata mesajı veriyoruz.

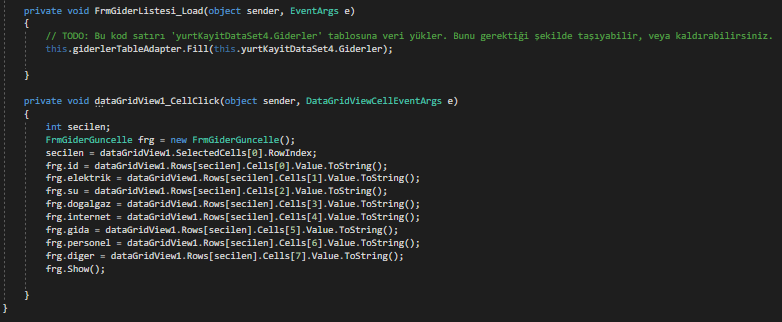
Ana sayfa da Giderler bölümünde Gider Listesi kısmına tıklandığı zaman karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 10.1 Gider Listesi Formu

Giderleri listelemek için Dataset üretiyoruz Giderler tablomuzdaki tüm sütunları seçiyoruz.

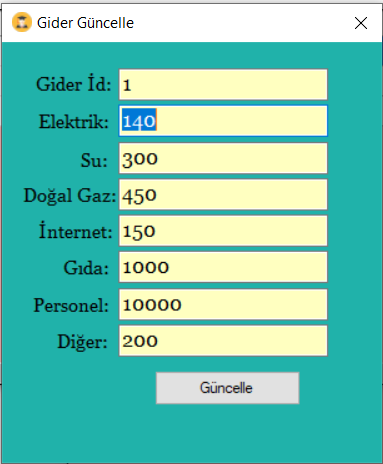
DataGridView’e tıklandığında ise gider bilgilerini güncelleyebilmek için yeni bir form açılmaktadır.



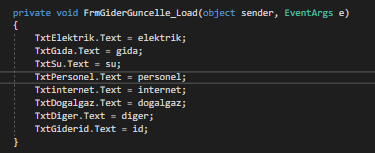
Şekil 10.2 DataGridView’a Tıklandığı Anda Çalışan Kod Satırları

Tıklandığı anda FrmGiderGuncelle formundan nesne oluşturuyoruz. DataGridView’ın satırlarındaki değerleri FrmGiderGuncelle de oluşturduğumuz değişkenlere atıyoruz. Sonrada FrmGiderGuncelle formunu ekrana getiriyoruz.

DataGridView satırındaki verilere tıklandığı anda gelen sayfa aşağıda gösterilmiştir.



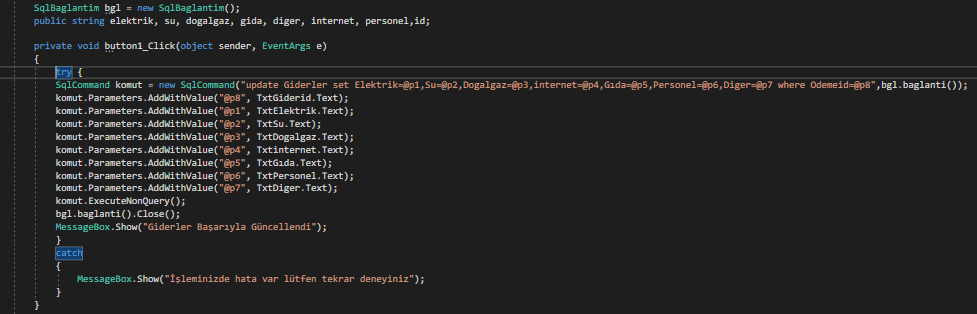
Şekil 11.1 Gider Güncelle Formu



Şekil 11.2 Form Yüklendiğinde Çalışan Kod Satırları

FrmGiderGuncelle formunda veritabanında Giderler tablosu için kullanacağımız değişkenlerimizi oluşturuyoruz. Daha sonra form yüklendiği anda kontrollerin text’lerine oluşturduğumuz değişkenleri atıyoruz.

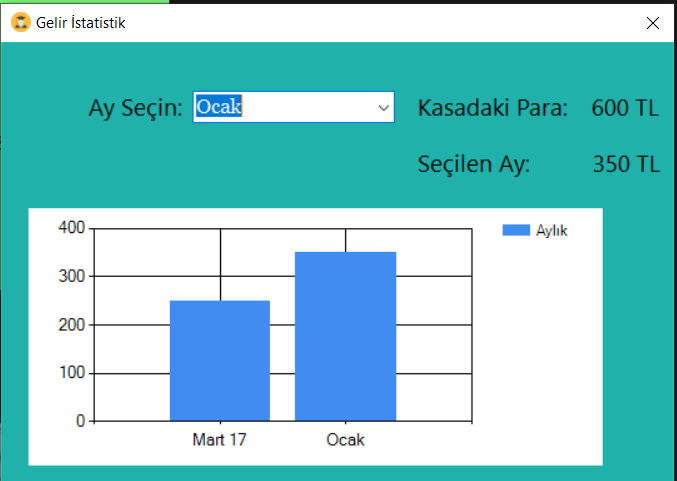
Güncelle button’una tıklandığında çalışan kod satırları aşağıda gösterilmiştir.



Şekil 11.3 Güncelle Button’una Tıklandığında Çalışan Kod Satırları

Komut nesnemizi oluşturup Ögrenci tablosundaki verileri Ogrid’ye göre güncelleyen update sorgsunu ve veritabanı bağlantımızı atıyoruz. Daha sonra tek tek @p parametrelerini veritabanındaki sütunlara eşitliyoruz. Ve veritabanı sorgumuzu çalıştıyoruz. Ekrana kayıt güncellendi mesajını veriyoruz. Eğer kayıt güncellenemezse ekrana hata mesajı verdiriyoruz.

Ana sayfa da İstatistikler bölümünde Gelir İstatistik kısmına tıklandığında karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 12.1 Gelir İstatistik Formu

Form yüklendiği anda çalışan komutlar aşağıda gösterilmiştir.



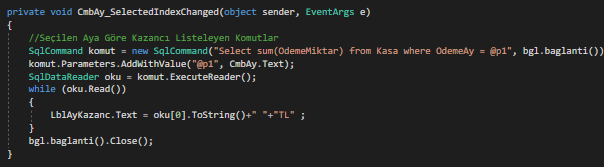
Şekil 12.2 FrmGelirİstatistik\_Load

Kasadaki toplam tutarı hesaplayabilmek için, komut adında nesne oluşturulup içine Kasa tablosundaki OdemeMiktarını sum komutuyla toplayıp listeleyen select sorgusu ve veritabanı bağlantısı atılmaktadır. Daha sonra veritabanındaki değerler okunduğu sürece label1’in text’ine okunan değer yazdırılır ve bağlantı kapatılır.

Tekrarsız olarak ayları combobox’ta listeleyebilmek için, komut2 nesnesi oluşturulup içine Kasa tablosundaki OdemeAy değerlerini distinct komutuyla tekrarsız sıralayıp listeleyen select sorgusu ve veritabanı bağlantısı atılmaktadır. Daha sonra değerler okunduğu sürece Combobox’ın item’larına değerler eklenmektedir. Veritabanı bağlantısı kapatılır.

Chart1 kontrolünde tekrarsız olarak ayları ve toplam tutarı Kasa tablosundaki OdemeAy sütununa göre listeleyebilmek için, komut3 nesnesi oluşturulup içine distinct komutuyla OdemeAy değerlerini sırasız olarak listeleyen ve OdemeMiktar değerlerini sum komutuyla toplayan bunları OdemeAy’a göre yapan select sorgusu ve veritabanı bağlantısı atılmaktadır. Daha sonra değerler okunduğu sürece Chart1 kontrolünün X ve Y koordinatlarına eklenmektedir. Veritabanı bağlantısı kapatılır.

Combobox’taki değerler değişince çalışan komut satırları aşağıda gösterilmiştir.



Şekil 12.3 CmbAy\_SelectedIndexChanged

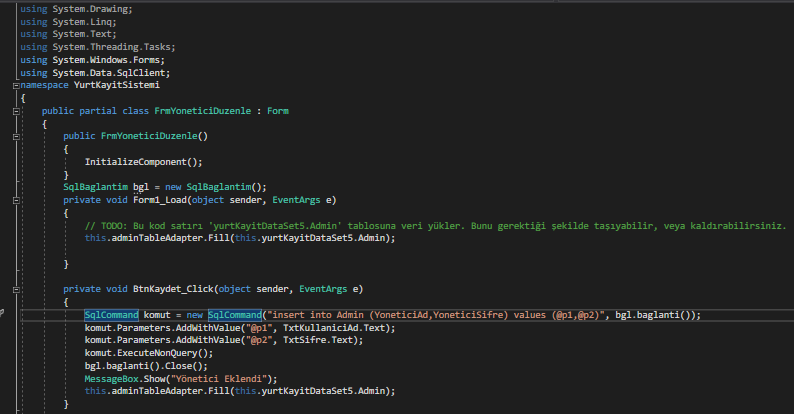
Combobax’taki değer değiştiği anda komut nesnesi oluşturulup içine Kasa tablosundaki OdemeMiktarını sum komutuyla toplayıp OdemeAy değerine göre listeleyen select sorgusu ve veritabanı bağlantısı atılmaktadır. Daha sonra @p1 parametresini veritabanındaki sütuna eşitlenir. Veritabanından değerler okunduğu sürece LblAyKazanc’ın textine okunan değer yazdırılır. Bağlantı kapatılır.

Ana sayfa da Yönetici bölümünde Şifre İşlemleri kısmına tıklandığında karşımıza gelen sayfa aşağıda gösterilmiştir.



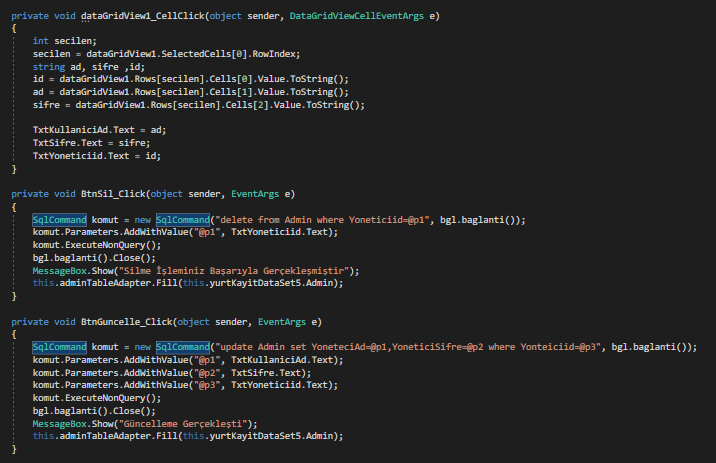
Şekil 13.1 Yönetici İşlemleri Formu

Yönetici şifre bilgilerini listelemek için Dataset üretiyoruz Admin tablomuzdaki tüm sütunları seçiyoruz.



Şekil 13.2 Kaydet Button’u Kod Satırları

Kaydet button’una tıklandığı zaman komut nesnesi oluşturulup içine Admin tablosuna YoneticiAd’ı ve YoneticiSifre’yi ekleyen insert sorgusu ve sql bağlantısı atılır. @p parametreleri veritabanındaki değerlere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana yönetici eklendi mesajı gönderilir. DataGridView refresh edilir.

 Şekil 13.3 Güncelle,Sil Button’u Ve DataGridView\_CellClick Kod Satırları

Verilerde düzenleme yapılabilmesi DataGridView’deki verilere tıklandığı anda textboxlara gelmeleri gerekmektedir. Bu sebeple DataGridView’e tıklandığı anda secilen,id,ad,sifre değişkenleri oluşturulur ve DataGridView satırlarındaki değerler değişkenlere atılır. Daha sonra değişkenler textbox’ların text’ine yazdırılır.

Sil button’una tıklandığı zaman komut nesnesi oluşturulup içine Admin tablosundan Yoneticiid’ye göre verileri silen delete sorgusu ve sql bağlantısı atılır. @p1 parametresi veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana silme işleminiz gerçekleşti mesajı gönderilir. DataGridView refresh edilir.

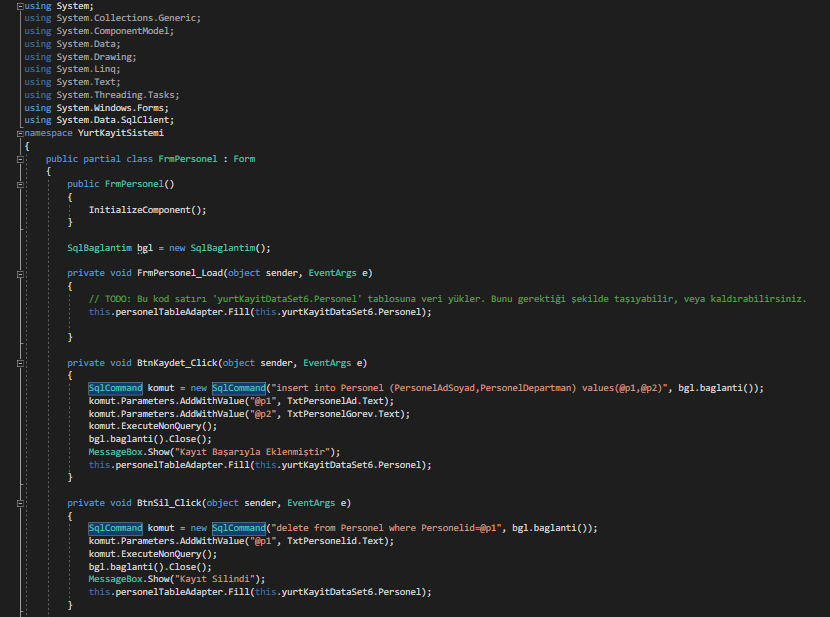
Güncelle button’una tıklandığı zaman komut nesnesi oluşturulup içine Admin tablosundan YoneticiAd ve YoneticiSifre sütununu Yoneticiid’ye göre verileri güncelleyen update sorgusu ve sql bağlantısı atılır. @p parametreleri veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana güncelle gerçekleşti mesajı gönderilir. DataGridView refresh edilir.

Ana sayfa da Yönetici bölümünde Personal Düzenle kısmına tıklandığında karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 14.1 Personel İşlemleri Formu

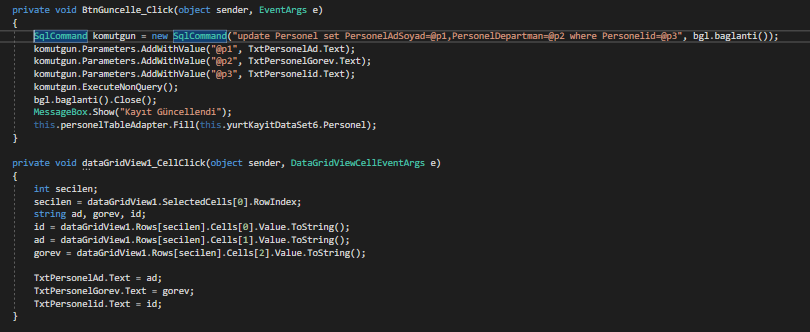
Personel bilgilerini listelemek için Dataset üretiyoruz Personel tablomuzdaki tüm sütunları seçiyoruz.



Şekil 14.2 Kaydet Ve Sil Button’ları Kod Satırları

Kaydet button’una tıklandığı zaman komut nesnesi oluşturulup içine Personel tablosuna PersonelAdSoyad’ı ve PersonelDepartman’ı ekleyen insert sorgusu ve sql bağlantısı atılır. @p parametreleri veritabanındaki değerlere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana kayıt eklendi mesajı gönderilir. DataGridView refresh edilir.

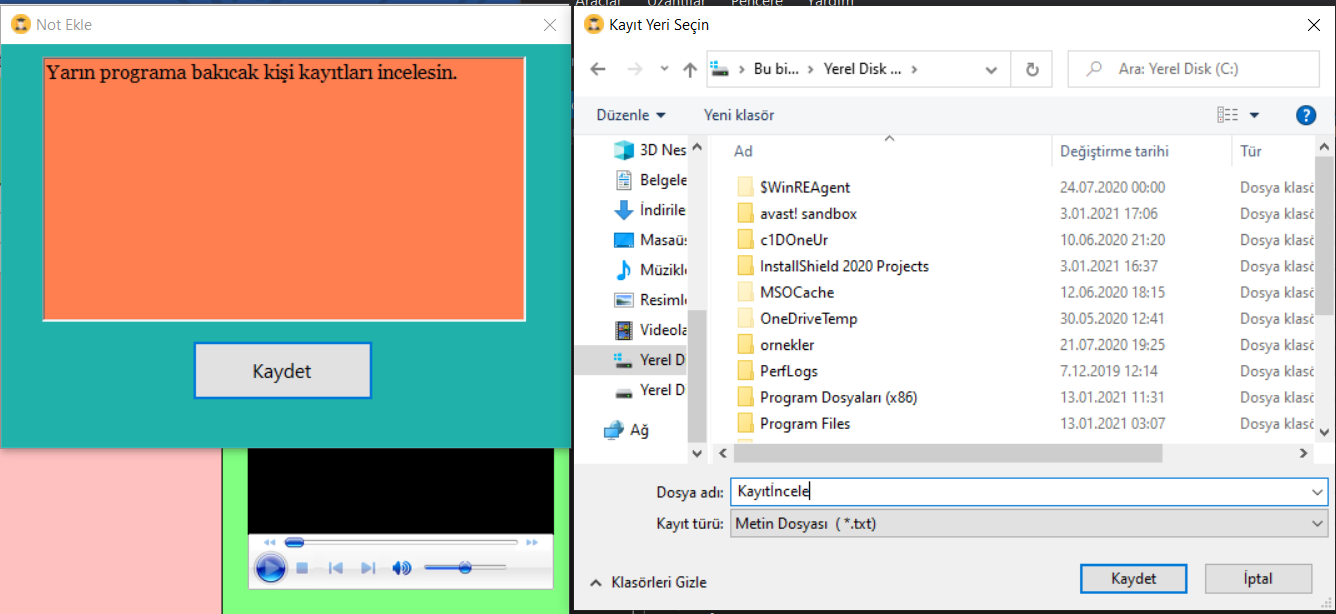
Sil button’una tıklandığı zaman komut nesnesi oluşturulup içine Personel tablosundan Personelid’ye göre verileri silen delete sorgusu ve sql bağlantısı atılır. @p1 parametresi veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana kayıt silindi mesajı gönderilir. DataGridView refresh edilir.

 Şekil 14.3 Güncelle Button’u Ve DataGridView1\_CellClick Kod Satırları

Güncelle button’una tıklandığı zaman komut nesnesi oluşturulup içine Personel tablosundan PersonelAdSoyad ve PersonelDepartman sütununu Personelid’ye göre verileri güncelleyen update sorgusu ve sql bağlantısı atılır. @p parametreleri veritabanındaki değere eşitlenir. Veritabanı sorgusu çalıştırılır ve bağlantı kapatılır. Ekrana güncelle gerçekleşti mesajı gönderilir. DataGridView refresh edilir.

Verilerde düzenleme yapılabilmesi DataGridView’deki verilere tıklandığı anda textboxlara gelmeleri gerekmektedir. Bu sebeple DataGridView’e tıklandığı anda secilen,id,ad,gorev değişkenleri oluşturulur ve DataGridView satırlarındaki değerler değişkenlere atılır. Daha sonra değişkenler textbox’ların text’ine yazdırılır.

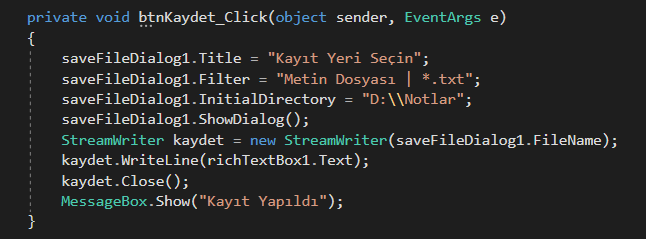
Ana sayfa da Yönetici bölümünde Not Ekle kısmına tıklandığında karşımıza gelen sayfa aşağıda gösterilmiştir.



Şekil 15.1 Not Ekle Formu

Öncelikle formumuza otomatik gelen kütüphanelerin yanında ek olarak

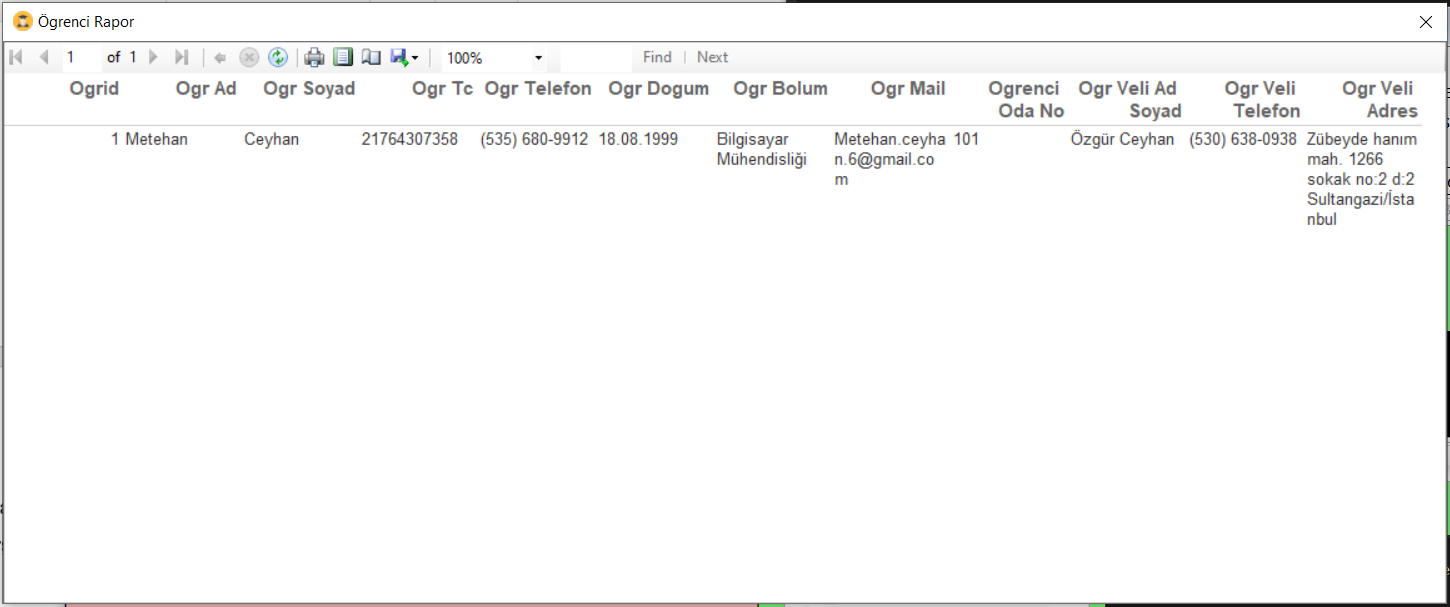
using System.IO; kütüphanesini de ekliyoruz.



Şekil 15.2 Kaydet Button’u Kod Satırları

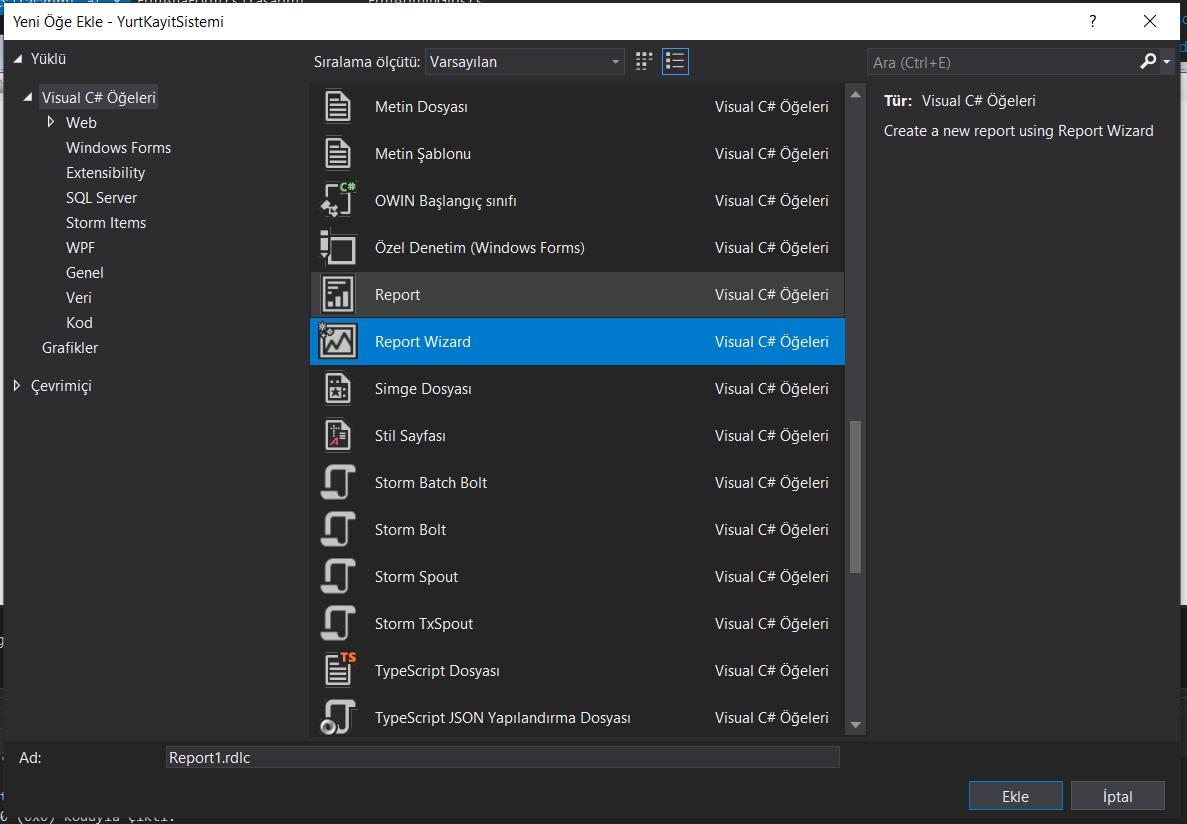
İlk önce programı kullanan kişinin notunu bilgisayarında bir yere kaydedebilmesi için programa saveFileDialog ekliyoruz. Kaydet button’una tıklandığı anda saveFileDialog’un başlığını, dosya türünü ve kaydedilecek yerini belirliyoruz. Daha sonra ekrana getiriyoruz. RichTextBox’tan verileri okuyabilmek için StreamWriter sınıfından kaydet nesnesini üretiyoruz ve saveFileDialog’un dosya adını parametre olarak veriyoruz. Kaydet.Writeline(richTextbox1.Text); komutu ile verileri okuyup belirlenmiş olan txt dosyasına kaydediyoruz. Ardından ekrana Kayıt Yapıldı şeklinde mesaj veriyoruz.

Ana sayfa da Yönetici bölümünde Rapor Al -> Öğrenci Bilgileri kısmına tıklandığında karşımıza gelen sayfa aşağıda gösterilmiştir.

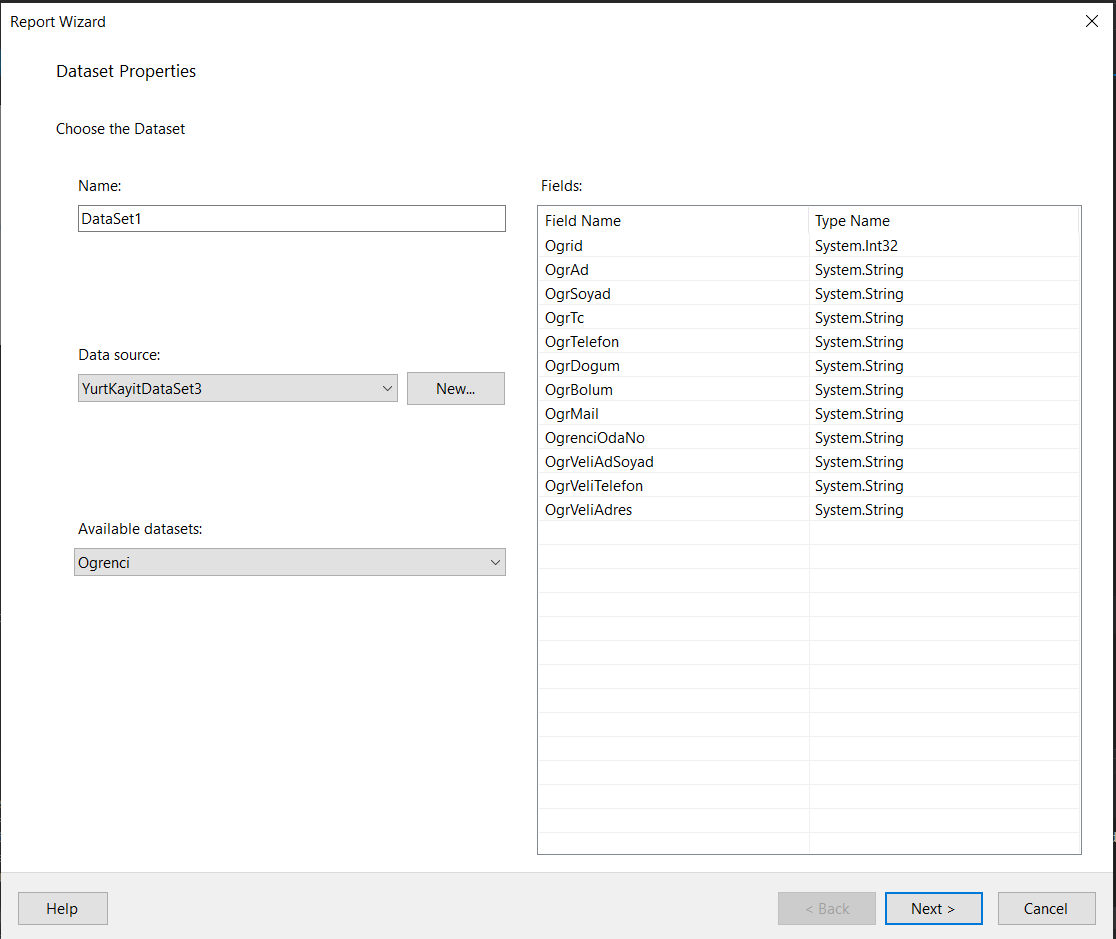


Şekil 16.1 Öğrenci Rapor Formu

Raporları listeleyebilmek ve çıktı alabilmemiz için formumuza Report Viewer ekliyoruz. Daha sonra Proje->Bileşen Ekle yolundan Report Wizard’ı seçiyoruz.

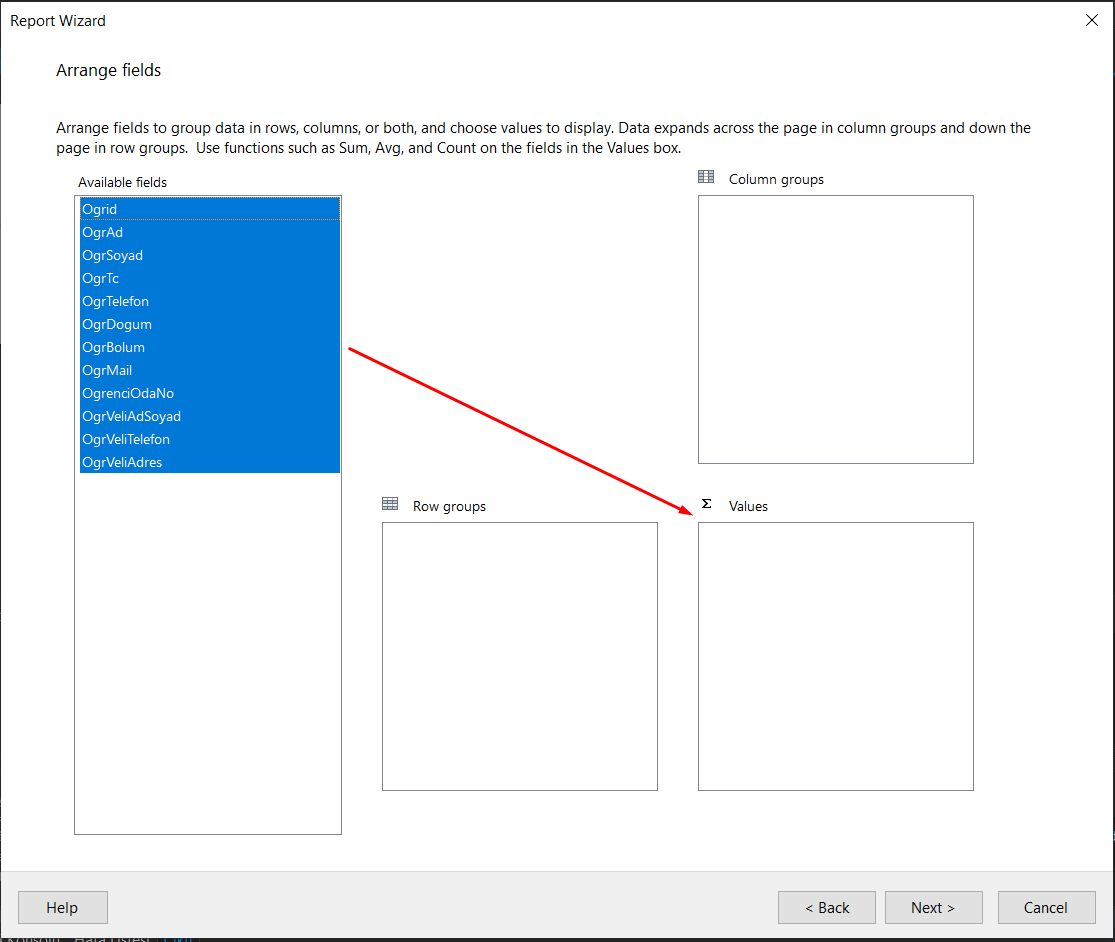


Şekil 16.2 Report Wizard Ekleme



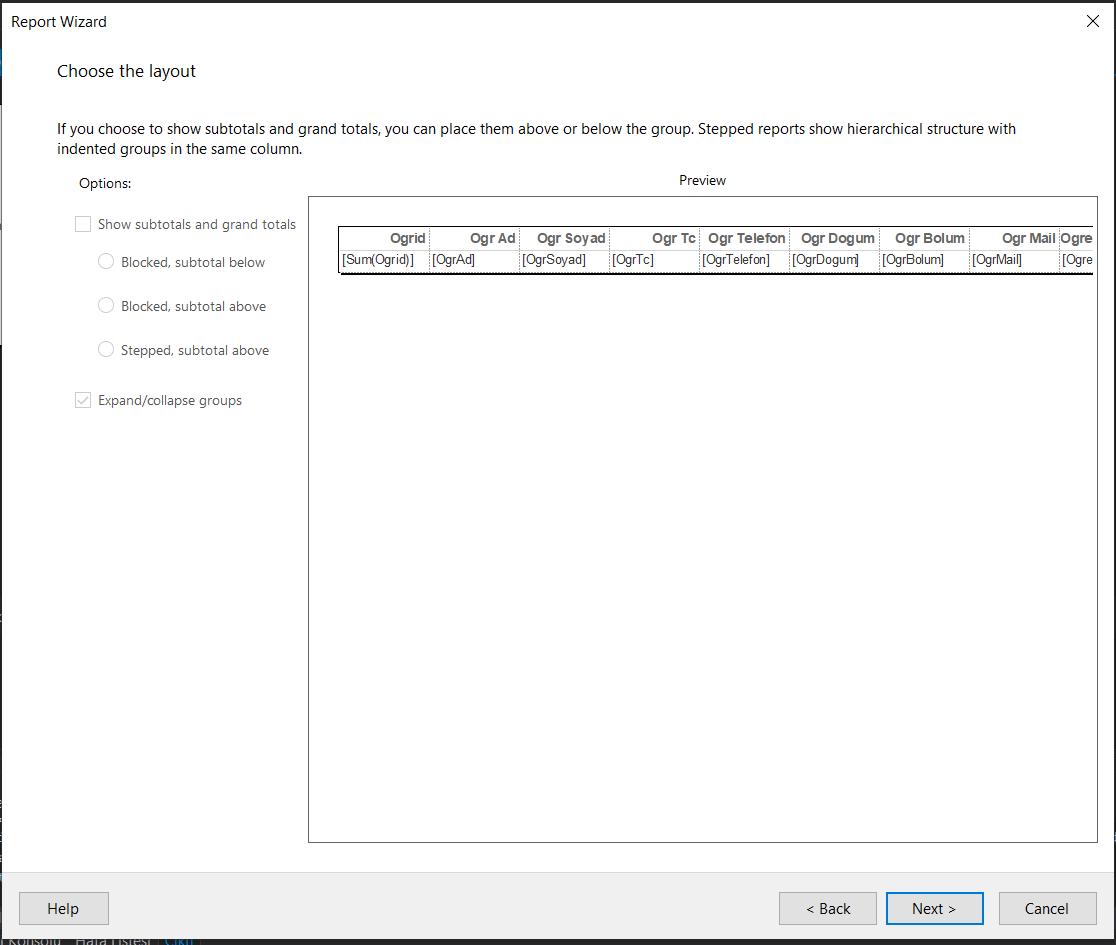
Şekil 16.3 Report Wizard 2. Aşama

Öğrenci bilgilerini listeleyebilmek için daha önce oluşturduğumuz Dataseti seçiyoruz.



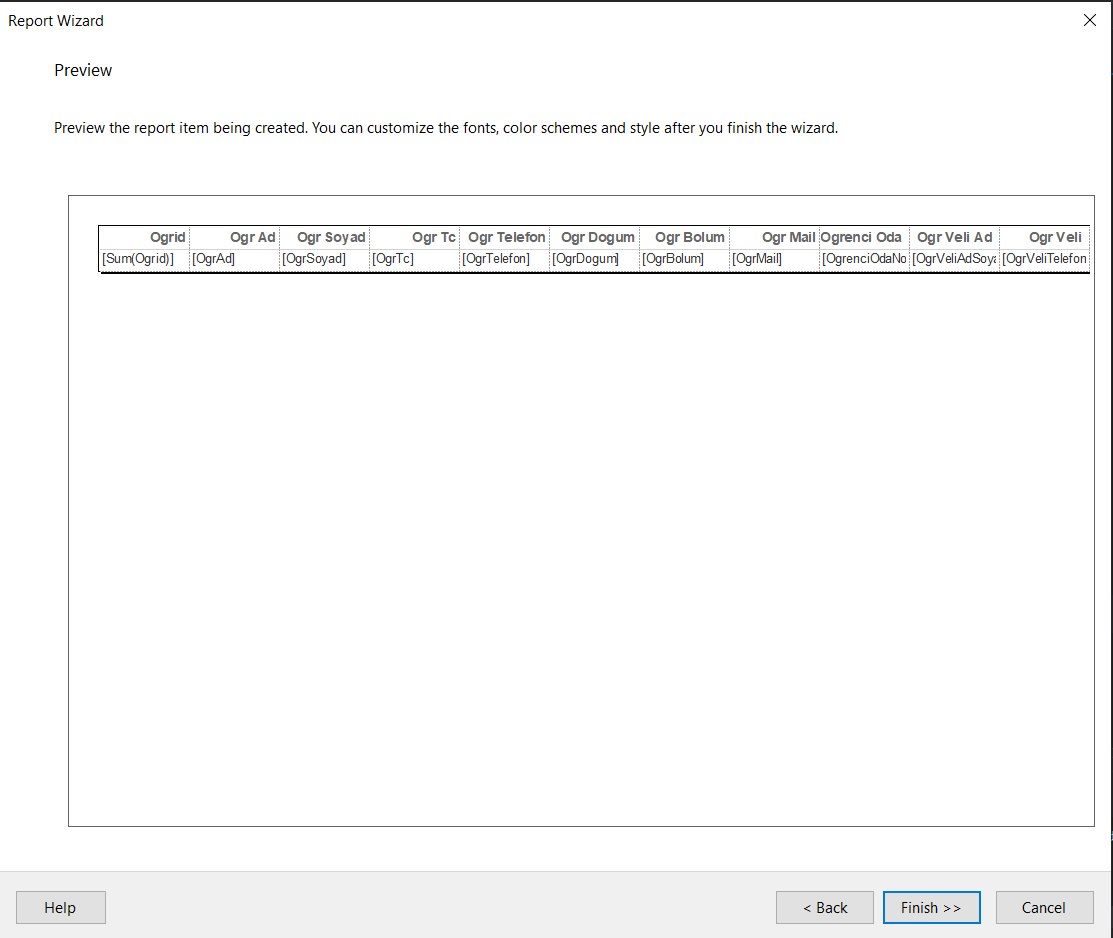
Şekil 16.4 Report Wizard 3. Aşama

Veritabanı sütunlarımızı kopyalayıp Values bölgesine yapıştırıyoruz.



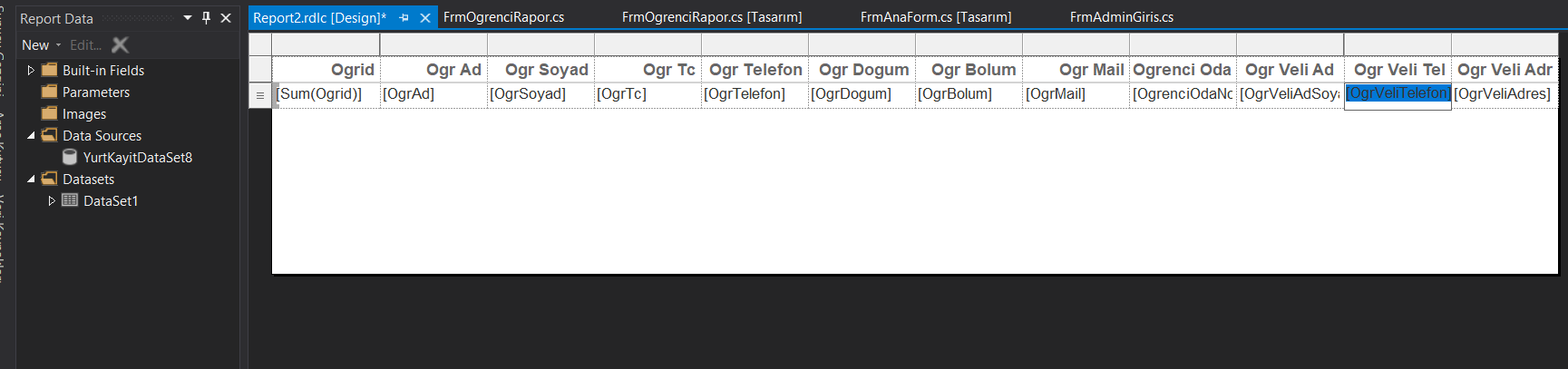
Şekil 16.5 Report Wizard 4. Aşama

Yukarı da tabloyu renklendirmek için olan sayfa gösterilmiştir.



Şekil 16.6 Report Wizard 5. Aşama

Finish button’una tıklandıktan sonra Report Viewer’a verilerimizi göndermiş oluyoruz.



Şekil 16.7 Report Wizard Arayüzü

1. SONUÇ VE DEĞERLENDİRME

Günümüzde en yaygın kullanılan masaüstü uygulama geliştirme ortamlarından biri Visual Studio Form Application’dır. Bu ortam da bir çok sektör için çeşitli tarzlarda uygumalar geliştirilebilmektedir. Benim amacım bu ortam da bir yurdun kullanabileceği sistem için gerekli olan tool’ları kullanarak bir uygulama geliştirebilmekti. Bu amacıma yüksek ölçüde ulaştım. Program da yurdun bilgi işlemini yöneten kişinin yapması gereken işlemlerin hemen hemen hepsi vardır. Programı kullanan kişi öğrenci ekleyebilir ve bilgilerini düzenleyebilir, öğrencileri eklerlen kolaylık sağlanması açısından bölüm ekleyebilir ve düzenleyebilir, öğrencilerin ödediği ücretin miktarını tutup, öğrencinin borcundan otomatik olarak düşebilir, yurdun giderleri sisteme girilebilir, sistem de listenebilir ve istatistik olarak gösterilebilir. Programı kullanan kişi yani yönetici, yeni bir yönetici ekleyebilir, yönetici kullanıcı adı ve şifreyi düzenleyebilir. Yönetici yurttaki personellerin bilgilerini tutabilir ve listeleyebilir gerektiğinde bilgileri düzenleyebilir. Gerektiğinde programdan kayıtlı olan öğrencilerin bilgileri çıktı olarak alınabilir. Programı kullanan kişiye kolaylık yaratılması açısından not alınabilir, paint programı açılabilir, hesap makinesi açılabilir ve radyo dinlenebilir.

Programın tasarım aşamasında program da olmasını düşündüğüm hemen hemen tüm özellikleri oluşturdum. Sonuç olarak program istenildiği takdir de herhangi bir kuruma kurulabilir ve çalışır vaziyettedir.

KAYNAKLAR

[1] G. Pipeleers, B. Demeulenaere, J. Swevers, and L. Vandenberghe, “Extended LMI characterizations for stability and performance of linear systems,” *Systems & Control Letters*, vol. 58, no. 7, pp. 510–518, 2009.

[2] L. Bei, W. Fen, and K. SungWan, “Switching LPV control of an F-16 aircraft via controller state reset,” *IEEE Transactions on Control Systems Technology*, vol. 14, no. 2, pp. 267–277, 2006.

[3] V. Q. Leu, H. H. Choi, and J. W. Jung, “Fuzzy sliding mode speed controller for PM synchronous motors with a load torque observer,” *IEEE Transactions on Power Electronics*, vol. 27, no. 3, pp. 1530–1539, 2012.

[4] M. Alma, J. J. Martinez, I. D. Landau, and G. Buche, “Design and Tuning of Reduced Order H-Infinity Feedforward Compensators for Active Vibration Control,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 2, pp. 554–561, 2012.

[5] K. Graichen, M. Treuer, and M. Zeitz, “Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation,” *Automatica*, vol. 43, no. 1, pp. 63–71, 2007.

[6] X. Litrico, V. Fromion, and G. Scorletti, “Robust feedforward boundary control of hyperbolic conservation laws,” *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5311–5316, 2006.

[7] I. Masubuchi and I. Kurata, “Gain-scheduled control via filtered scheduling parameters,” *Automatica*, vol. 47, no. 8, pp. 1821–1826, 2011.

ÖZGEÇMİŞ

**KİŞİSEL BİLGİLER**

|  |  |
| --- | --- |
| Adı Soyadı | : Metehan Ceyhan |
| Doğum Tarihi ve Yeri | : 18.08.1999/İstanbul |
| Yabancı Dili | : İngilizce |
| E-posta | : Metehan.ceyhan.6@gmail.com |

**ÖĞRENİM DURUMU**

|  |  |  |  |
| --- | --- | --- | --- |
| **Derece** | **Alan** | **Okul/Üniversite** | **Mezuniyet Yılı** |
| Lisans | Bilgisayar Müh. | Düzce Üniversitesi | 2021 |
| Lise |  | Oğuz Canpolat Anadolu Lisesi | 2017 |