İçindekiler Tablosu

[**1. PROJE NEDİR**](#ProjeNedir1)**1**

[**2. Hangi Amaçla Yazıldı**](#HangiAmaçlaYazıldı2)**2**

[**3. Ne İşe Yarıyor**](#NeİşeYarıyor3)**2**

[**4. Nereye Kadar Tamamlandı**](#NereyeKadarTamamlandı4)**2**

[**5. Veri Tabanı Diyagramı Ve Tablolar**](#VeriTabanıDiyagramıVeTablolar5)**3**

[5.1 Veri Tabanı Diyagramı 4](#_Toc60946415)

[5.2 Tedarikçi Tablosu 5](#_Toc60946416)

[5.3 Kategori Tablosu 6](#_Toc60946417)

[5.4 Ürün Tablosu 7](#_Toc60946418)

[5.5 Müşteri Tablosu 8](#_Toc60946419)

[5.6 Sipariş Tablosu 9](#_Toc60946420)

[5.7 Sipariş Ürün Detay Tablosu 10](#_Toc60946421)

[**6. Arayüz Görselleri Ve Açıklaması**](#Arayüzgörselleri)**11**

[6.1 Ana Sayfa Ara Yüzü 11](#_Toc60946422)

[6.2 Müşteri Ara Yüzü 12](#_Toc60946423)

[6.3 Sipariş Ara Yüzü 13](#_Toc60946424)

[6.4 Ürünler Ara Yüzü 14](#_Toc60946425)

[6.5 Ürün Tedarikçileri Ara Yüzü 15](#_Toc60946426)

[6.6 Kategori Ara Yüzü 16](#_Toc60946427)

[6.7 Sipariş Rapor Ara Yüzü 17](#_Toc60946428)

[6.7.1 Ürün Rapor Ara Yüzü 18](#_Toc60946429)

[6.7.2 Müşteri Rapor Ara Yüzü 19](#_Toc60946430)

[6.7.3 Tedarikçi Rapor Ara Yüzü 20](#_Toc60946431)

[6.7.4 Kategori Rapor Ara Yüzü………………………………………………………………………………………………………….21](#_Toc60946432)

[**7. Proje Kodları Ve Veri Tabanı Scripti**](#projekodlarıveritabıscripti)**4**

[7.1 Mağaza Otomasyon Programının Ana Sayfa Formunun Proje Kodları 22](#_Toc60946433)

[7.2 Mağaza Otomasyon Programının Müşteri Formunun Proje Kodları 24](#_Toc60946434)

[7.3 Mağaza Otomasyon Programının Sipariş Formunun Proje Kodları 32](#_Toc60946435)

[7.4 Mağaza Otomasyon Programının Ürün Formunun Proje Kodları 40](#_Toc60946436)

[7.5 Mağaza Otomasyon Programının Tedarikçiler Formunun Proje Kodları 47](#_Toc60946437)

[7.6 Mağaza Otomasyon Programının Kategori Formunun Proje Kodları 54](#_Toc60946438)

[7.7 Mağaza Otomasyon Programının Rapor Formunun Proje Kodları 61](#_Toc60946439)

[7.8 Mağaza Otomasyon Programının frm Sınıfın Kodları 70](#_Toc60946440)

[7.9 Mağaza Otomasyon Programının db Sınıfının Kodları 71](#_Toc60946441)

[7.10 Veri Tabanının Scripti 81](#_Toc60946442)

# **MAĞAZA OTOMASYON PROGRAMI**

**Melih KOÇAK**

Kayseri Üniversitesi Develi Hüseyin Şahin M.Y.O. Bilgisayar Programcılığı Bölümü

**Özet**

Bu rapor mağaza otomasyon programına aittir. 2020 - 2021 eğitim öğretim yılı 2. sınıf güz dönemi içerisinde Araştırma Yöntem Ve Teknikleri dersi için ödev olarak hazırlanmıştır. Bu program ile bir mağazanın ürün tedarikçilerden ürün tedarik edip, satışına kadar gerekli olan işlemlerin birçoğunu karşılaması düşünülüp o amaç doğrultusunda geliştirilmiştir. Programın hedef kitlesi genel olarak küçük işletmelerdir.

1. PROJE NEDİR

Küçük bir işletmenin ürün tedarikçilerinin bilgilerini işleme, satın alınan ürünlerin bilgilerini mağazadaki kategorilere göre işleme, mağazadan ürün alan müşterilerin bilgilerini işleme ve siparişlerin bilgilerini işlemek için C# ve MSSQL kullanılarak tasarlanan bir otomasyon programıdır.

2. HANGİ AMAÇLA YAZILDI

Gün geçtikçe dijitalleşen dünyamız da son yıllara bakılırsa bilgisayar ve elektronik eşya kullanımı yüksek bir oranda artmakta ve artmaya devam ediyor. Küçük, büyük birçok işletme artık dijitalleşme yolunda ilerliyor. Çoğu işletme işlerini dijital ortamda gerçekleştiriyor, işletmeler dijitalleşmek için çeşitli programlar kullanmaktadır. İşletme bünyesinde kullanılan programlar işletmelere birçok yönden avantaj sağlamakta ve en basitinden kağıt daha az kullanılmakta, kayıt edilen bilgiler bilgisayar ortamında saklandığı için fiziksel olarak nerdeyse yer kaplamamaktadır. Not kalabalığı, yanma ve işletmenin bilgilerine zarar verecek çeşitli tehlikelerden de yedekleme yöntemi ile bilgilerimiz kolayca zarar gelmeyecek şekilde saklanabilir. Tasarladığım bu mağaza otomasyonu, işletmelerin dijitalleşme yolunda kullanabilmesi, çeşitli kolaylıklar sağlamak ve gündelik işlerini rahatlıkla halledebilmek için yazılmıştır.

3. NE İŞE YARIYOR

Temel bir mağaza otomasyon programı şu işlere:

1. Yetkililerinin ürün tedarikçilerinden ürün tedarik ettiğinde tedarikçi bilgilerini ve ürün bilgilerini kaydetmek veya silmek, güncellemek istendiğinde
2. Ürün satışı gerçekleştirmek için müşteri bilgilerini kaydetme, silme, güncelleme durumunda
3. İlgili ürünlerin satış işlemi gerçekleştiğinde
4. Ürünleri ilgili kategorilere ayırmak gerektiğinde vb. durumlarda

Rahatlıkla cevap verebiliyor olması gerekmektedir. Nitekim benim geliştirdiğim mağaza otomasyon programı bu isteklere ve bu gibi isteklere günümüz standartlarına uygun, kullanımı kolay, hızlı ve sade arayüzü ile rahatlıkla cevap verebilmektedir.

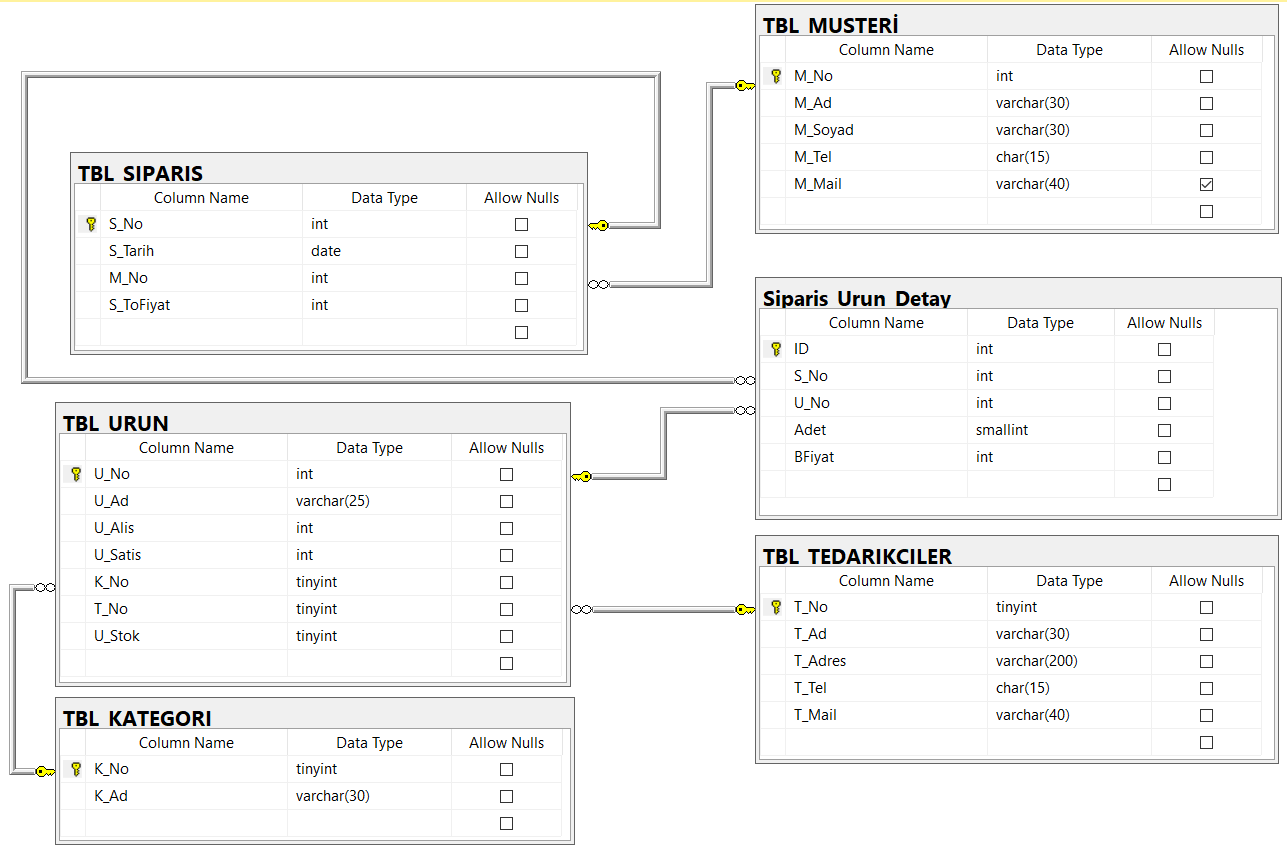
4.NEREYE KADAR TAMAMLANDI

C# programlama dili ve SQL veri işleme dili kullanarak geliştirdiğim mağaza otomasyon programı tamamen hazır hale getirilip, son kullanıcının rahatla ve sorunsuz bir şekilde kullanabilmesi sağlanmıştır.

5. VERİTABANI DİYAGRAMI VE TABLOLAR

Bu bölümde mağaza otomasyonu programı için gerekli olan veri tabanını oluştururken tasarlanan tabloları ve tabloların birbiri ile olan ilişkilerini gösteren veri tabanı diyagramının ne için kullanıldıkları anlatılacaktır.

### 5.1 Veri Tabanı Diyagramı



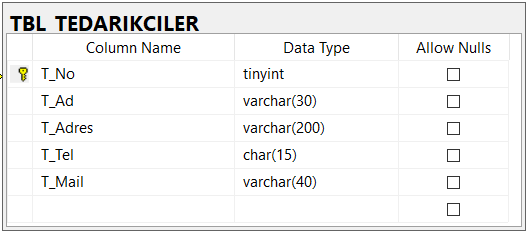
Resim 1: Mağaza Otomasyonu Programının Veri Tabanı Diyagramı

Mağaza otomasyonu programının gerekli olan Veri Tabanı tabloları ve tablo ilişkileri resimde görüldüğü gibidir. (BKNZ RESİM1) Veri Tabanı tasarlanırken bir senaryo düşünülerek hazırlanmıştır. Senaryo şöyledir:

1. Mağazamıza ürün sağlayan tedarikçilerin ismini, adresini, telefon numarasını ve mail adresinin bilgilerini tutmak istiyoruz.
2. Mağazamızda bulunan ürünlerimizin ismini, stok miktarını, alış ve satış fiyatını, hangi tedarikçiden alındığı ve hangi kategoriye ait olduğunun bilgisini tutmak istiyoruz.
3. Mağazamıza ürün satın almak için gelen müşterilerimizin ismini, soy ismini, telefon numarasını ve gerekli durumlarda mail adresinin bilgilerini tutmak bizim için yeterli olacaktır.
4. Müşterinin siparişini gerçekleştirirken hangi ürünü hangi müşteriye sattığımızın kaydını, sipariş tarihini, alınan ürünlerin ismini ve adetini, siparişin toplam tutarını tutmak istiyoruz.

Bu senaryo düşünülerek veri tabanı için gerekli tablolar oluşturulup, kendi aralarında ilişkilendirilmiştir.

### 5.2 Tedarikçi Tablosu



Resim 2: Mağaza Otomasyonu Programının Tedarikçi Tablosu

Düşünülen senaryoya uygun olarak ‘TBL\_TEDARIKCILER’ adında bir tablo tasarlanmıştır. (BKNZ RESİM2)

Düşünülen senaryoda kullanıcı bizden mağazaya ürün sağlayan tedarikçinin ismini, adresini, telefon numarasını ve mail adresini tutmasının gerekli olduğunu söylemektedir.

* Kullanıcı bahsetmese, istemese bile biz tablolar arası ilişkileri korumak için, ekleme, silme, güncelleme işlemlerini yapabilmek için tablolarımıza Primary Key ekleriz. TBL\_TEDARIKCILER tablosuna ‘T\_No’ kolonu eklenip, Primary Key yapılmıştır.
* Tedarikçinin isim bilgisini ‘TBL\_TEDARIKCILER’ tablosunda ‘T\_Ad’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her tedarikçinin bir ismi olması gerektiği” düşüncesiyle boş geçilemeyecek ve 30 karakterin tedarikçi ismi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Tedarikçinin adres bilgisini ‘TBL\_TEDARIKCILER’ tablosunda ‘T\_Adres’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her tedarikçinin bir adresi mutlaka vardır” düşüncesiyle boş geçilemeyecek ve 200 karakterin tedarikçinin adresi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Tedarikçinin telefon numarası bilgisi için ‘TBL\_TEDARIKCILER’ tablosunda ‘T\_Tel’ isminde char veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her tedarikçinin bir telefon numarası olması gerekir” düşüncesiyle boş geçilemeyecek ve 15 karakterin tedarikçinin telefon numarası için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Tedarikçinin mail adres bilgisi için TBL\_TEDARIKCILER’ tablosunda ‘T\_Mail’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her tedarikçinin bir mail adresi olması gerekir” düşüncesiyle boş geçilemeyecek ve 40 karakterin tedarikçinin mail adresi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.

### 5.3 Kategori Tablosu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 3: Mağaza Otomasyonu Programının Kategori Tablosu

Düşünülen senaryoya uygun olarak ‘TBL\_KATEGORI’ adında bir tablo tasarlanmıştır. (BKNZ RESİM3)

Düşünülen senaryoda kullanıcı sadece ürünün hangi kategoriye ait olduğunu bilmek istemektedir.

* Kullanıcı bahsetmese, istemese bile biz tablolar arası ilişkileri korumak için, ekleme, silme, güncelleme işlemlerini yapabilmek için tablolarımıza Primary Key ekleriz. ‘TBL\_KATEGORI’ tablosuna ‘K\_No’ kolonu eklenip, Primary Key yapılmıştır.
* Ürünün isim bilgisini ‘TBL\_KATEGORU’ tablosunda ‘K\_Ad’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun boş geçilemeyecek ve 30 karakterin ürün ismi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.

### 5.4 Ürün Tablosu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 4: Mağaza Otomasyonu Programının Ürün Tablosu

Düşünülen senaryoya uygun olarak ‘TBL\_URUN’ adında bir tablo tasarlanmıştır. (BKNZ RESİM4)

Düşünülen senaryoda kullanıcı bizden ürünün ismini, stok miktarını, alış ve satış fiyatını, hangi tedarikçiden aldığını ve hangi kategoride bulunduğunun bilgisi tutmasının gerekli olduğunu söylemektedir.

* Kullanıcı bahsetmese, istemese bile biz tablolar arası ilişkileri korumak için, ekleme, silme, güncelleme işlemlerini yapabilmek için tablolarımıza Primary Key ekleriz. ‘TBL\_URUN’ tablosuna ‘U\_No’ kolonu eklenip, Primary Key yapılmıştır.
* Ürünün isim bilgisini ‘TBL\_URUN’ tablosunda ‘U\_Ad’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her ürünün bir ismi olması gerektiği” düşüncesiyle boş geçilemeyecek ve 25 karakterin ürün ismi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Ürünün alış fiyat bilgisini ‘TBL\_URUN’ tablosunda ‘T\_Alis’ isminde int veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her ürünün bir alış fiyatı mutlaka vardır” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Ürünün satış fiyat bilgisini ‘TBL\_URUN’ tablosunda ‘T\_Satis’ isminde int veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her ürünün bir satış fiyatı mutlaka vardır” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Ürünün hangi kategoride bulunduğunun bilgisi için ‘TBL\_URUN‘ tablosunda ‘K\_No’ isminde tinyint veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her ürünün bir kategoriye aittir” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Ürünün hangi tedarikçiden tedarikçiden tedarik edildiğinin bilgisi için ‘TBL\_URUN‘ tablosunda ‘T\_No’ isminde tinyint veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her ürünün bir tedarikçisi mutlaka vardır” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Ürünün stok bilgisi için ‘TBL\_URUN‘ tablosunda ‘U\_Stok’ isminde tinyint veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her ürünün bir stok sayısı vardır” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.

### 5.5 Müşteri Tablosu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 5: Mağaza Otomasyonu Programının Müşteri Tablosu

Düşünülen senaryoya uygun olarak ‘TBL\_MUSTERİ’ adında bir tablo tasarlanmıştır. (BKNZ RESİM5)

Düşünülen senaryoda kullanıcı bizden müşterinin ismini, soy ismini, telefon numarasını ve gerekli durumlarda mail adresinin bilgilerini tutmanın yeterli olduğunu söylemektedir.

* Kullanıcı bahsetmese, istemese bile biz tablolar arası ilişkileri korumak için, ekleme, silme, güncelleme işlemlerini yapabilmek için tablolarımıza Primary Key ekleriz. ‘TBL\_MUSTERİ’ tablosuna ‘M\_No’ kolonu eklenip, Primary Key yapılmıştır.
* Müşterinin isim bilgisini ‘TBL\_MUSTERİ’ tablosunda ‘M\_Ad’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her müşterinin bir ismi olması gerektiği” düşüncesiyle boş geçilemeyecek ve 30 karakterin müşteri ismi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Müşterinin soy isim bilgisini ‘TBL\_MUSTERİ’ tablosunda ‘M\_Soyad’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her müşterinin bir soy ismi olması gerektiği” düşüncesiyle boş geçilemeyecek ve 30 karakterin müşteri ismi için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Müşterinin telefon numarası bilgisi için ‘TBL\_MUSTERİ’ tablosunda ‘M\_Tel’ isminde char veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her müşterinin bir telefon numarası verebileceği” düşüncesiyle ve boş geçilemeyecek, 15 karakterin müşteri telefon numarası için kafi olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Müşterinin mail bilgisini ‘TBL\_MUSTERİ’ tablosunda ‘M\_Mail’ isminde varchar veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “mail adresinin gerekli olduğunda girilmesi” düşüncesiyle boş bırakabilir ve 40 karakterin müşteri mail adresi için yeterli olduğunu düşündüm ve ona göre ayarlamalar yaptım.

### 5.6 Sipariş Tablosu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 6: Mağaza Otomasyonu Programının Sipariş Tablosu

Düşünülen senaryoya uygun olarak ‘TBL\_SIPARIS’ adında bir tablo tasarlanmıştır. (BKNZ RESİM6)

Düşünülen senaryoda kullanıcı bizden sipariş işlemi gerçekleşirken siparişin tarihini, hangi ürünün hangi müşteriye satıldığını ve siparişin toplam fiyat bilgilerini tutulmasını istemiştir.

* Kullanıcı bahsetmese, istemese bile biz tablolar arası ilişkileri korumak için, ekleme, silme, güncelleme işlemlerini yapabilmek için tablolarımıza Primary Key ekleriz. ‘TBL\_SIPARIS’ tablosuna ‘S\_No’ kolonu eklenip, Primary Key yapılmıştır.
* Siparişin tarih bilgisini ‘TBL\_SIPARIS’ tablosunda ‘S\_Tarih’ isminde date veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her siparişin bir tarihi olur” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Ürünü hangi müşteriye sattığımızın kaydını tutmak için ‘TBL\_SIPARIS’ tablosunda ‘M\_No’ isminde int veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “her sipariş bir müşterinindir” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.
* Siparişin toplam fiyatının kaydını tutmak için ‘TBL\_SIPARIS’ tablosunda ‘S\_ToFiyat’ isminde int veri tipinde bir kolon oluşturarak tutmaya karar verdim. Bu kolonun “alınan ürünlerin bir fiyatı” düşüncesiyle boş geçilemeyecek olduğunu ön gördüm ve ona göre gerekli ayarlamaları yaptım.

### 5.7 Sipariş Ürün Detay Tablosu

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 7: Mağaza Otomasyonu Programının Sipariş Ürün Detay Tablosu

Düşünülen senaryoda ‘TBL\_URUN’ ve ‘TBL\_SIPARIS’ tablolarının çoğa çok ilişkisinden doğan ‘Siparis\_Urun\_Detay’ tablosudur. (BKNZ RESİM7)

* Kullanıcı bahsetmese, istemese bile biz tablolar arası ilişkileri korumak için, ekleme, silme, güncelleme işlemlerini yapabilmek için tablolarımıza Primary Key ekleriz. ‘Siparis\_Urun\_Detay’ tablosuna ‘ID’ kolonu eklenip, Primary Key yapılmıştır.
* İlişkisel veri tabanında çoğa çok ilişkilerin tabloya dönüştürülürken, çoğa çok ilişkisi olan iki tablonun da birincil anahtarı çoğa çok ilişkiden ortaya çıkan tabloya yabancıl anahtar olarak eklenir. Bu bağlamda ‘Siparis\_Urun\_Detay’ tablosuna ‘S\_No’ ve ‘U\_No’ kolonları yabancıl anahtar olarak eklenmiştir.
* Adet ve Birim Fiyat durumları sipariş sırasında oluştuğu için çoğa çok ilişkiden ortaya çıkan ‘Siparis\_Urun\_Detay’ tablosuna ‘Adet’ ve ‘BFiyat’ kolonu olarak eklenip boş geçilemez olarak ayarlanmıştır.

6.ARAYÜZ GÖRSELLERİ VE AÇIKLAMASI

Bu bölümde Mağaza Otomasyonu programı için geliştirdiğim ara yüzünü fotoğraflarını ekleyerek ara yüz üzerindeki kısımların ne işe yaradığını anlatacağım.

### 6.1 Ana Sayfa Ara Yüzü



Resim 8: Mağaza Otomasyonu Programının Ana Sayfa Arayüzü

* Kayseri Üniversitesi logosunun altında üstten ilk sırada yer alan ‘Müşteri’ yazan kısım butondur. Müşteri ara yüzünü sağ tarafta boş görünen alanda açmaktadır.
* Kayseri Üniversitesi logosunun altında üstten ikinci sırada yer alan ‘Sipariş’ yazan kısım butondur. Ürünler Sipariş ara yüzünü sağ tarafta boş görünen alanda açmaktadır.
* Kayseri Üniversitesi logosunun altında üstten üçüncü sırada yer alan ‘Ürünler’ yazan kısım butondur. Ürünler ara yüzünü sağ tarafta boş görünen alanda açmaktadır.
* Kayseri Üniversitesi logosunun altında üstten dördüncü sırada yer alan ‘Ürün Tedarikçileri’ yazan kısım butondur. Tedarikçi ara yüzünü sağ tarafta boş görünen alanda açmaktadır.
* Kayseri Üniversitesi logosunun altında üstten beşinci sırada yer alan ‘Kategori’ yazan kısım butondur. Kategori ara yüzünü sağ tarafta boş görünen alanda açmaktadır.
* Kayseri Üniversitesi logosunun altında üstten altıncı sırada yer alan ‘Raporlama’ yazan kısım butondur. Raporlar ara yüzünü sağ tarafta boş görünen alanda açmaktadır.
* Kayseri Üniversitesi logosunun altında üstten yedinci sırada yer alan ‘Çıkış’ yazan yerin yanındaki simge butondur. Uygulamadan çıkış yapmaya yarar.

### 6.2 Müşteri Ara Yüzü

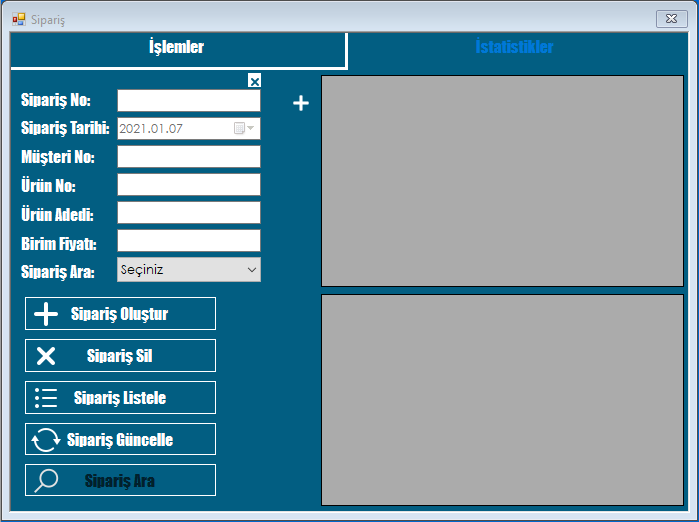
tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 9: Mağaza Otomasyonu Programının Müşteri Ara Yüzü

* En altta bulunan DataGridView aracını veri tabanımızda ki ‘TBL\_MUSTERİ’ tablosundaki verileri listelemek için kullanırız.
* ‘Müşteri Ekle’ butonu sol tarafta ilgili alanlara girilen bilgileri veri tabanımızda ki ilgili tabloya ekleme işlemi yapmaktadır.
* ‘Müşteri Sil’ butonu ‘Müşteri No’ alanına girilen değere göre tablomuzda silme işlemi gerçekleştirir.
* ‘Müşteri Listele’ butonu veri tabanımızda ki ‘TBL\_MUSTERİ’ tablosundaki verileri DataGridView aracında listeler.
* ‘Müşteri Güncelle’ butonu sol tarafta ilgili alanlara girilen bilgileri ‘Müşteri No’ alanını baz alarak tablomuzda güncelleme işlemi gerçekleştirir.
* ‘Müşteri Ara’ butonu sol taraftaki ‘Müşteri Arama’ ComboBox aracında seçtiğimiz değere göre arama yapmaktadır.
* TextBox ların üzerinde bulunan küçük ‘X’ şeklindeki buton TextBoxları sıfırlamaya yaramaktadır.
* En üstte ‘İşlemler’ sekmesinin yanında ki ‘İstatistikler’ sekmesi ise müşterilere ait bazı istatistikleri gösterir.

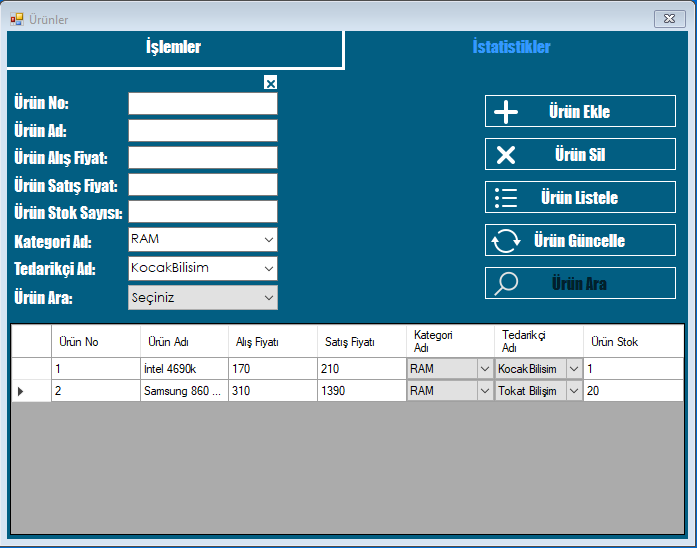
### 6.3 Sipariş Ara Yüzü



Resim 10: Mağaza Otomasyonu Programının Sipariş Ara Yüzü

* Sağ tarafta iki adet bulunan araçlar DataGridView aracıdır. İlgili tablolarımızı listelemek için kullanırız.
* Sipariş No alanının yanındaki ‘ + ’ simgeli buton son kaydın bir fazlasını Sipariş No alanına yazdırmaktadır.
* ‘Sipariş Oluştur’ yukarıdaki ilgili alanlara girilen bilgileri veri tabanımızdaki ilgili tablolara veri ekleme işlemi yapmaktadır.
* ‘Sipariş Sil’ butonu ‘Sipariş No’ alanına girilen değere göre tablomuzda silme işlemi gerçekleştirir.
* ‘Sipariş Listele’ butonu veri tabanımızda ki ‘TBL\_SIPARIS’ ve ‘Siparis\_Urun\_Detay’ tablosundaki verileri ilgili DataGridView aracında listeler.
* ‘Sipariş Güncelle’ butonu sol tarafta ilgili alanlara girilen bilgileri ‘Sipariş No’ alanını baz alarak tablolarımızda güncelleme işlemi gerçekleştirir.
* ‘Sipariş Ara’ butonu sol taraftaki ‘Sipariş Arama’ ComboBox aracında seçtiğimiz değere göre arama yapmaktadır.
* TextBoxların üzerinde bulunan küçük ‘X’ şeklindeki buton TextBoxları sıfırlamaya yaramaktadır.
* En üstte ‘İşlemler’ sekmesinin yanında ki ‘İstatistikler’ sekmesi ise siparişlere ait bazı istatistikleri gösterir.

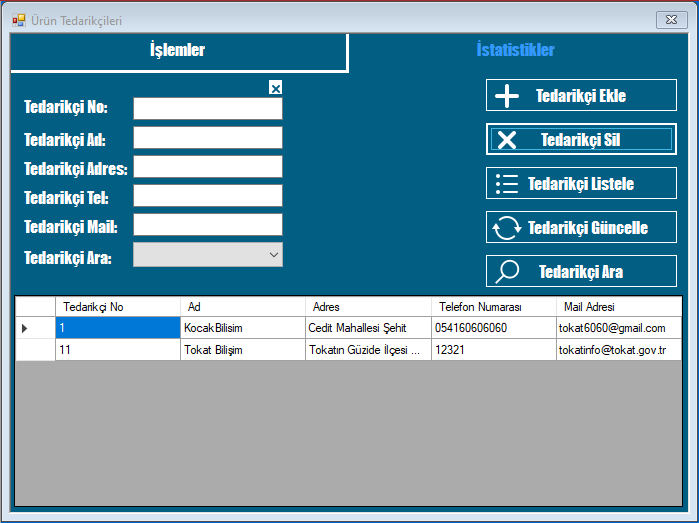
### 6.4 Ürünler Ara Yüzü



Resim 11: Mağaza Otomasyonu Programının Ürünler Ara Yüzü

* En altta bulunan DataGridView aracını veri tabanımızda ki ‘TBL\_URUN’ tablosundaki verileri listelemek için kullanırız.
* ‘Ürün Ekle’ butonu sol tarafta ilgili alanlara girilen bilgileri veri tabanımızda ki ilgili tabloya ekleme işlemi yapmaktadır.
* ‘Ürün Sil’ butonu ‘Ürün No’ alanına girilen değere göre tablomuzda silme işlemi gerçekleştirir.
* ‘Ürün Listele’ butonu veri tabanımızda ki ‘TBL\_URUN’ tablosundaki verileri DataGridView aracında listeler.
* ‘Ürün Güncelle’ butonu sol tarafta ilgili alanlara girilen bilgileri ‘Ürün No’ alanını baz alarak tablomuzda güncelleme işlemi gerçekleştirir.
* ‘Ürün Ara’ butonu sol taraftaki ‘Ürün Arama’ ComboBox aracında seçtiğimiz değere göre arama yapmaktadır.
* TextBoxların üzerinde bulunan küçük ‘X’ şeklindeki buton TextBoxları sıfırlamaya yaramaktadır.
* En üstte ‘İşlemler’ sekmesinin yanında ki ‘İstatistikler’ sekmesi ise müşterilere ait bazı istatistikleri gösterir.

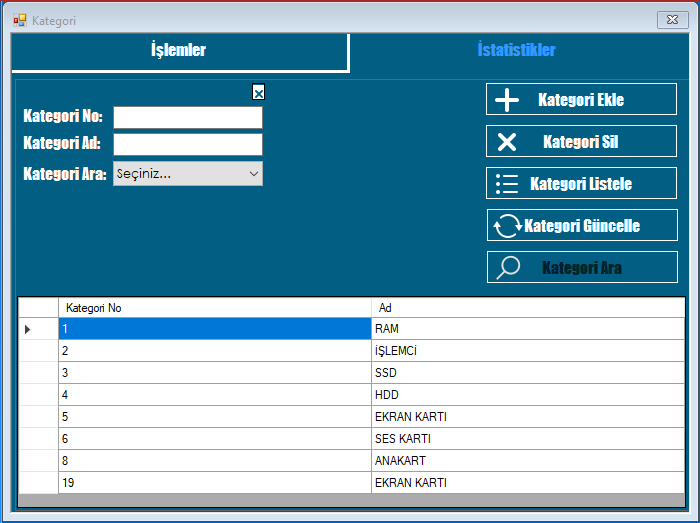
### 6.5 Ürün Tedarikçileri Ara Yüzü



Resim 12: Mağaza Otomasyonu Programının Ürün Tedarikçileri Ara Yüzü

* En altta bulunan DataGridView aracını veri tabanımızda ki ‘TBL\_TEDARIKCILER’ tablosundaki verileri listelemek için kullanırız.
* ‘Tedarikçi Ekle’ butonu sol tarafta ilgili alanlara girilen bilgileri veri tabanımızda ki ilgili tabloya ekleme işlemi yapmaktadır.
* ‘Tedarikçi Sil’ butonu ‘Tedarikçi No’ alanına girilen değere göre tablomuzda silme işlemi gerçekleştirir.
* ‘Tedarikçi Listele’ butonu veri tabanımızda ki ‘TBL\_TEDARIKCILER’ tablosundaki verileri DataGridView aracında listeler.
* ‘Tedarikçi Güncelle’ butonu sol tarafta ilgili alanlara girilen bilgileri ‘Tedarikçi No’ alanını baz alarak tablomuzda güncelleme işlemi gerçekleştirir.
* ‘Tedarikçi Ara’ butonu sol taraftaki ‘Tedarikçi Arama’ ComboBox aracında seçtiğimiz değere göre arama yapmaktadır.
* TextBoxların üzerinde bulunan küçük ‘X’ şeklindeki buton TextBoxları sıfırlamaya yaramaktadır.
* En üstte ‘İşlemler’ sekmesinin yanında ki ‘İstatistikler’ sekmesi ise müşterilere ait bazı istatistikleri gösterir.

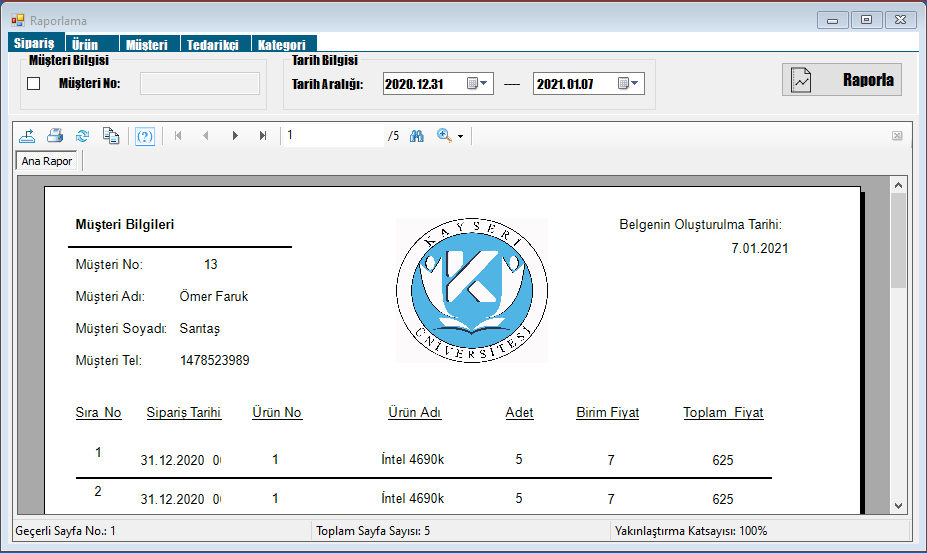
### 6.6 Kategori Ara Yüzü



Resim 13: Mağaza Otomasyonu Programının Kategori Ara Yüzü

* En altta bulunan DataGridView aracını veri tabanımızda ki ‘TBL\_KATEGORI’ tablosundaki verileri listelemek için kullanırız.
* ‘Kategori Ekle’ butonu sol tarafta ilgili alanlara girilen bilgileri veri tabanımızda ki ilgili tabloya ekleme işlemi yapmaktadır.
* ‘Kategori Sil’ butonu ‘Kategori No’ alanına girilen değere göre tablomuzda silme işlemi gerçekleştirir.
* ‘Kategori Listele’ butonu veri tabanımızda ki ‘TBL\_KATEGORI’ tablosundaki verileri DataGridView aracında listeler.
* ‘Kategori Güncelle’ butonu sol tarafta ilgili alanlara girilen bilgileri ‘Kategori No’ alanını baz alarak tablomuzda güncelleme işlemi gerçekleştirir.
* ‘Kategori Ara’ butonu sol taraftaki ‘Kategori Arama’ ComboBox aracında seçtiğimiz değere göre arama yapmaktadır.
* TextBoxların üzerinde bulunan küçük ‘X’ şeklindeki buton TextBoxları sıfırlamaya yaramaktadır.
* En üstte ‘İşlemler’ sekmesinin yanında ki ‘İstatistikler’ sekmesi ise müşterilere ait bazı istatistikleri gösterir.

### 6.7 Sipariş Rapor Ara Yüzü



Resim 14: Mağaza Otomasyonu Programının Sipariş Raporlama Ara Yüzü

* Raporla butonuna basıldığında ‘Müşteri Bilgisi’ GroupBoxındaki CheckBox işaretli değilse ‘Tarih Bilgisi’ GroupBoxında belirlediğimiz tarihler arasındaki siparişlerin hepsini görebiliriz.
* Raporla butonuna basıldığında ‘Müşteri Bilgisi’ GroupBoxındaki CheckBox işaretli ise belirlediğimiz müşterinin ‘Tarih Bilgisi’ GroupBoxında belirlediğimiz tarihler arasındaki siparişlerin hepsini görebiliriz.

### 6.7.1 Ürün Rapor Ara Yüzü

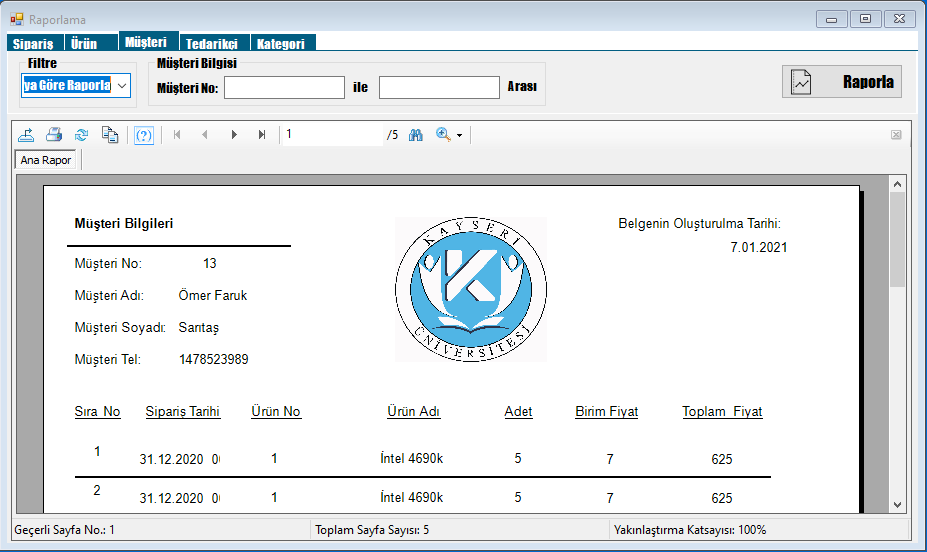
tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 15: Mağaza Otomasyonu Programının Ürün Raporlama Ara Yüzü

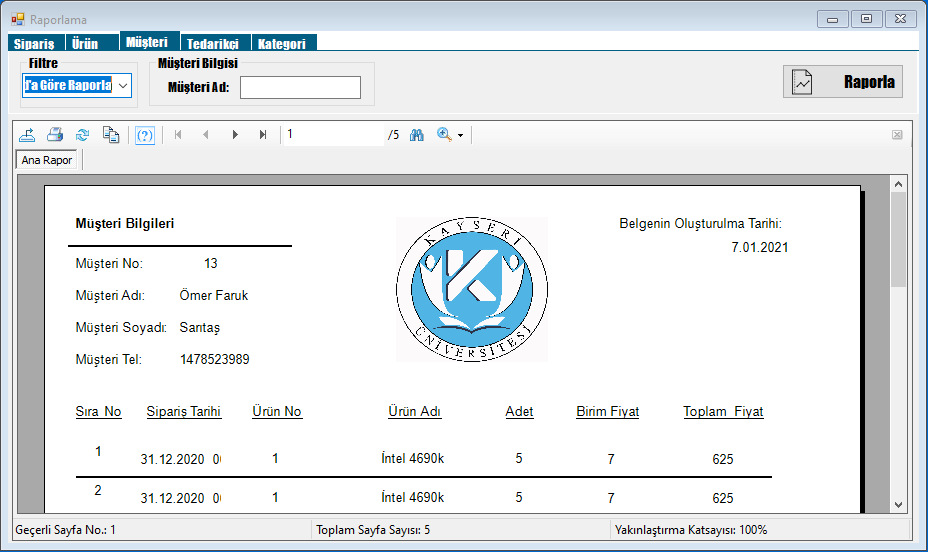
* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının birinci değeri seçili ise ‘Tedarikçi No’ ya göre raporlama yapılacaktır.
* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının birinci değeri seçili ise ‘Tedarikçi No’ ya göre ve ‘Fiyat Aralığı’ GroupBoxındaki CheckBox da işaretliyse fiyat aralığı da girilmesine izin verilecek ona göre raporlama yapılacaktır.
* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının ikinci değeri seçili ise ‘Kategori No’ ya göre raporlama yapılacaktır.
* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının ikinci değeri seçili ise ‘Kategori No’ ya göre ve ‘Fiyat Aralığı’ GroupBoxındaki CheckBox da işaretliyse fiyat aralığı da girilmesine izin verilecek ve ona göre raporlama yapılacaktır.

### 6.7.2 Müşteri Rapor Ara Yüzü



Resim 16: Mağaza Otomasyonu Programının Müşteri Raporlama Ara Yüzü

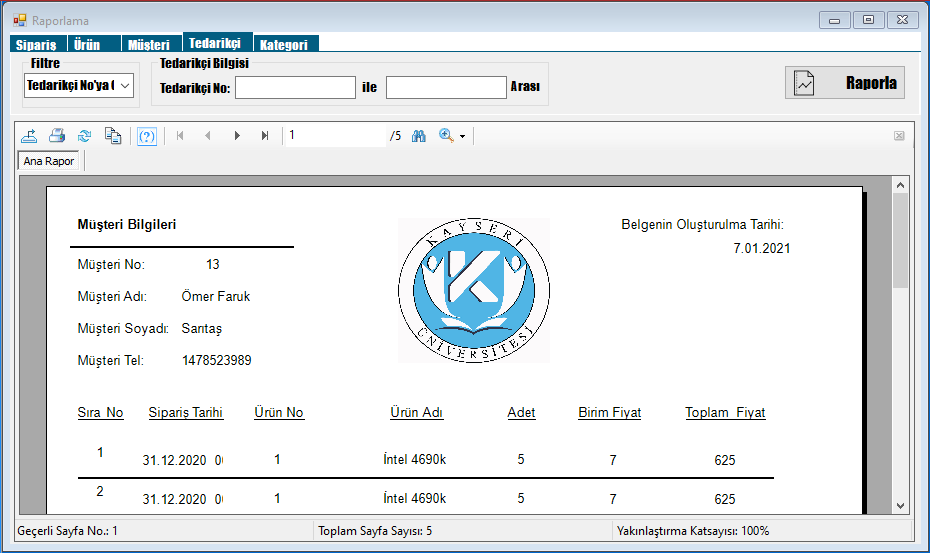
* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının birinci değeri seçili ise ‘Müşteri No’ alanının bizim belirleyeceğimiz aralığa göre raporlama yapılacaktır.



Resim 17: Mağaza Otomasyonu Programının Müşteri Raporlama Ara Yüzü 2

* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının ikinci değeri seçili ise ‘Müşteri Ad’ alanına göre raporlama yapılacaktır.

### 6.7.3 Tedarikçi Rapor Ara Yüzü



Resim 18: Mağaza Otomasyonu Programının Tedarikçi Raporlama Ara Yüzü

* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının birinci değeri seçili ise ‘Tedarikçi No’ alanının bizim belirleyeceğimiz aralığa göre raporlama yapılacaktır.

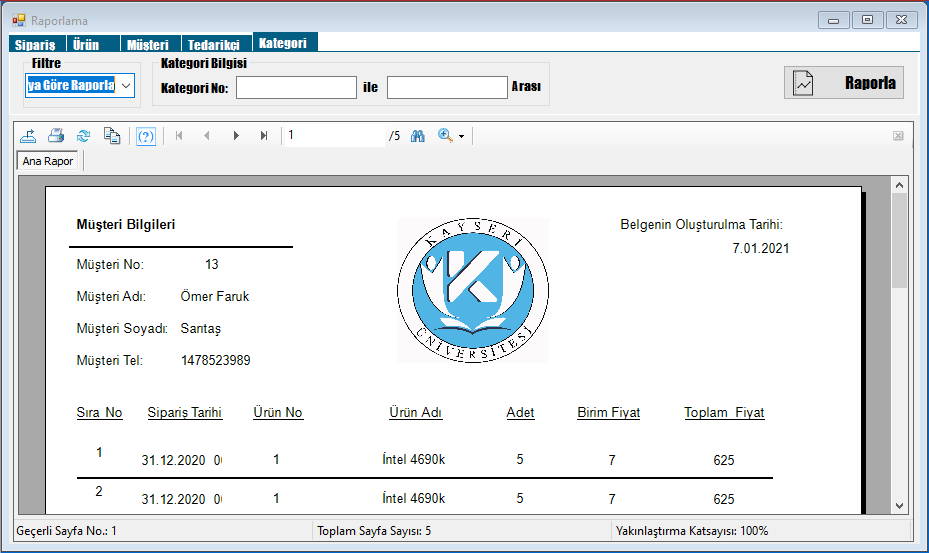
metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Resim 19: Mağaza Otomasyonu Programının Tedarikçi Raporlama Ara Yüzü 2

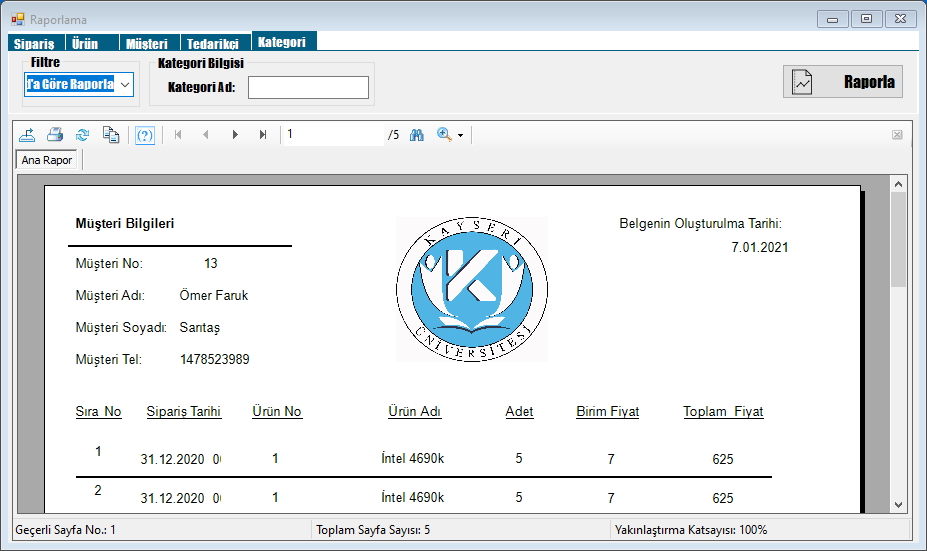
* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının ikinci değeri seçili ise ‘Tedarikçi Ad’ alanına göre raporlama yapılacaktır.

### 6.7.4 Kategori Rapor Ara Yüzü



Resim 20: Mağaza Otomasyonu Programının Kategori Raporlama Ara Yüzü

* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının birinci değeri seçili ise ‘Kategori No’ alanının bizim belirleyeceğimiz aralığa göre raporlama yapılacaktır.



Resim 21: Mağaza Otomasyonu Programının Kategori Raporlama Ara Yüzü 2

* Raporla butonuna basıldığında ‘Filtre’ GroupBoxındaki Filtreleme ComboBox aracının ikinci değeri seçili ise ‘Kategori Ad’ alanına göre raporlama yapılacaktır.

7. PROJE KODLARI VE VERİ TABANI SCRİPTİ

Bu bölümde proje için yazdığım bütün proje kodları ve veri tabanı scriptleri bulunmaktadır.

### 7.1 Mağaza Otomasyon Programının Ana Sayfa Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace melihkocak0385

{

public partial class frm\_ana : Form

{

db k = new db();

frm f = new frm();

public void frm\_ana\_Load(object sender, EventArgs e)

{

this.Location = Screen.AllScreens[0].WorkingArea.Location; //frm\_ana formunun formborderstyle özelliği none olduğu için hareket etmiyordu bu kod diğer bazı kodlar hareket ettirmek için yazıldı.

}

public frm\_ana()

{

InitializeComponent();

}

//

// Müşteri Butonu

//

private void btn\_musteri\_Click(object sender, EventArgs e)

{

frm\_musteri a = new frm\_musteri(); //Müşteri formundan nesne türettim

a.Name = "kontrol";//aktif bir form varken başka bir form açılmasını engellemek için name kısmınına "kontrol" adını verdim.

f.formgetir(a, pnl\_orta,a);//frm classında bulunan bu metot 'pnl\_orta' paneline daha önceden açık bir form varmı yokmu kontrolünü yaparak içine gönderdiğimiz form nesnesini getiriyor. eğer daha önceden açık bir form varsa hata mesajı veriyor.

pnl\_gecis.Top = btn\_musteri.Top;//Butona tıklayınca yandaki panelin geçişini sağlayan yok

}

//

// Ürün Butonu

//

private void btn\_urun\_Click(object sender, EventArgs e)

{

frm\_urun a = new frm\_urun();//Ürün formundan nesne türettim

a.Name = "kontrol";//başka bir form açılmasını engellemek için name kısmınına "kontrol" adını verdim.

f.formgetir(a, pnl\_orta,a);//frm classında bulunan bu metot 'pnl\_orta' paneline daha önceden açık bir form varmı kontrolünü yaparak içine gönderdiğimiz form nesnesini getiriyor. eğer daha önceden açık bir form varsa hata mesajı veriyor.

pnl\_gecis.Top = btn\_urun.Top;//Butona tıklayınca yandaki panelin geçişini sağlayan yok

}

//

// Kategori Butonu

//

private void btn\_kategori\_Click(object sender, EventArgs e)

{

frm\_kategori a = new frm\_kategori();//Kategori formundan nesne türettim

a.Name = "kontrol";//başka bir form açılmasını engellemek için name kısmınına "kontrol" adını verdim.

f.formgetir(a, pnl\_orta,a);//frm classında bulunan bu metot 'pnl\_orta' paneline daha önceden açık bir form varmı kontrolünü yaparak içine gönderdiğimiz form nesnesini getiriyor. eğer daha önceden açık bir form varsa hata mesajı veriyor.

pnl\_gecis.Top = btn\_kategori.Top;//Butona tıklayınca yandaki panelin geçişini sağlayan yok

}

//

// Sipariş Butonu

//

private void btn\_siparis\_Click(object sender, EventArgs e)

{

frm\_siparis a = new frm\_siparis();//Sipariş formundan nesne türettim

a.Name = "kontrol";//başka bir form açılmasını engellemek için name kısmınına "kontrol" adını verdim.

f.formgetir(a, pnl\_orta, a);//frm classında bulunan bu metot 'pnl\_orta' paneline daha önceden açık bir form varmı kontrolünü yaparak içine gönderdiğimiz form nesnesini getiriyor. eğer daha önceden açık bir form varsa hata mesajı veriyor.

pnl\_gecis.Top = btn\_siparis.Top;//Butona tıklayınca yandaki panelin geçişini sağlayan yok

}

//

// Çıkış Butonu

//

private void btn\_exit\_Click(object sender, EventArgs e)

{

Application.Exit();

}

//

// Tedarikçi Butonu

//

private void btn\_tedarikciler\_Click(object sender, EventArgs e)

{

frm\_tedarikciler a = new frm\_tedarikciler();//Tedairkçiler formundan nesne türettim

a.Name = "kontrol";//başka bir form açılmasını engellemek için name kısmınına "kontrol" adını verdim.

f.formgetir(a, pnl\_orta, a);//frm classında bulunan bu metot 'pnl\_orta' paneline daha önceden açık bir form varmı kontrolünü yaparak içine gönderdiğimiz form nesnesini getiriyor. eğer daha önceden açık bir form varsa hata mesajı veriyor.

pnl\_gecis.Top = btn\_tedarikciler.Top;//Butona tıklayınca yandaki panelin geçişini sağlayan yok

}

//

//Rapor Butonu

//

private void btn\_rapor\_Click(object sender, EventArgs e)

{

WindowsFormsApp3.Formlar.frm\_rapor a = new WindowsFormsApp3.Formlar.frm\_rapor();//Rapor formundan nesne türettim

a.Name = "kontrol";//başka bir form açılmasını engellemek için name kısmınına "kontrol" adını verdim.

f.formgetir(a, pnl\_orta, a);//frm classında bulunan bu metot 'pnl\_orta' paneline daha önceden açık bir form varmı kontrolünü yaparak içine gönderdiğimiz form nesnesini getiriyor. eğer daha önceden açık bir form varsa hata mesajı veriyor.

pnl\_gecis.Top = btn\_rapor.Top;//Butona tıklayınca yandaki panelin geçişini sağlayan yok

}

//

//formborderstyle none olunca hareket etmesi için

//

//-----------------------------------------------------------------------------

private void pnl\_ust\_MouseDown(object sender, MouseEventArgs e)

{

f.dizi[0] = 1;

f.dizi[1] = e.X; //formborderstyle none olunca hareket etmesi için

f.dizi[2] = e.Y;

}

private void pnl\_ust\_MouseMove(object sender, MouseEventArgs e)

{

if (f.dizi[0]==1)

{

this.SetDesktopLocation(MousePosition.X - f.dizi[1], MousePosition.Y - f.dizi[2]); //formborderstyle none olunca hareket etmesi için

}

}

private void pnl\_ust\_MouseUp(object sender, MouseEventArgs e)

{

f.dizi[0] = 0; //formborderstyle none olunca hareket etmesi için

}

//------------------------------------------------------------------------------

//

// Butonların Altında Panelin Renk Ve Boyutunu Ayarlayan Kodlar

//

//-------------------------------------------------------------------------------

private void btn\_musteri\_MouseHover(object sender, EventArgs e)//Müşteri butonunun mouse ile üzerine gelince gerçekleşecek event

{

pnl\_musteri.Size = new Size(180, 3);//Boyutunu ayarlıyorum, küçültüyorum.

pnl\_musteri.BackColor = Color.Brown;//Rengini kahverengi yapıyorum.

pnl\_musteri.Location= new Point(14, 179);//Küçük olunca düzgün durması için yerini değiştirdim.

}

private void btn\_musteri\_MouseLeave(object sender, EventArgs e)//Müşteri butonunun mouse ile üzerine gelip ayrıldıktan sonra gerçekleşecek event

{

pnl\_musteri.Size = new Size(200, 3);//Boyutunu ayarlıyorum. Eski hale getiriyorum

pnl\_musteri.Location = new Point(5, 179);//Boyut büyüyünce yerinide ilk haline getiriyorum.

pnl\_musteri.BackColor = Color.White;//Tekrardan rengini beyaz yapıyorum.

}

private void btn\_urun\_MouseHover(object sender, EventArgs e)

{

pnl\_urun.Size = new Size(180, 3);

pnl\_urun.BackColor = Color.Brown;

pnl\_urun.Location = new Point(15, 321);

}

private void btn\_urun\_MouseLeave(object sender, EventArgs e)

{

pnl\_urun.Size = new Size(200, 3);

pnl\_urun.BackColor = Color.White;

pnl\_urun.Location = new Point(6, 321);

}

private void btn\_siparis\_MouseHover(object sender, EventArgs e)

{

pnl\_siparis.Size = new Size(180, 3);

pnl\_siparis.BackColor = Color.Brown;

pnl\_siparis.Location = new Point(14, 252);

}

private void btn\_siparis\_MouseLeave\_1(object sender, EventArgs e)

{

pnl\_siparis.Size = new Size(200, 2);

pnl\_siparis.BackColor = Color.White;

pnl\_siparis.Location = new Point(6, 252);

}

private void btn\_kategori\_MouseHover(object sender, EventArgs e)

{

pnl\_kategori.Size = new Size(180, 3);

pnl\_kategori.BackColor = Color.Brown;

pnl\_kategori.Location = new Point(14, 459);

}

private void btn\_kategori\_MouseLeave(object sender, EventArgs e)

{

pnl\_kategori.Size = new Size(200, 3);

pnl\_kategori.BackColor = Color.White;

pnl\_kategori.Location = new Point(6, 459);

}

private void btn\_tedarikciler\_MouseHover(object sender, EventArgs e)

{

pnl\_tedarikciler.Size = new Size(180, 3);

pnl\_tedarikciler.BackColor = Color.Brown;

pnl\_tedarikciler.Location = new Point(14, 390);

}

private void btn\_tedarikciler\_MouseLeave(object sender, EventArgs e)

{

pnl\_tedarikciler.Size = new Size(200, 3);

pnl\_tedarikciler.BackColor = Color.White;

pnl\_tedarikciler.Location = new Point(6, 390);

}

private void btn\_rapor\_MouseHover(object sender, EventArgs e)

{

pnl\_rapor.Size = new Size(180, 3);

pnl\_rapor.BackColor = Color.Brown;

pnl\_rapor.Location = new Point(14, 528);

}

private void btn\_rapor\_MouseLeave(object sender, EventArgs e)

{

pnl\_rapor.Size = new Size(200, 3);

pnl\_rapor.BackColor = Color.White;

pnl\_rapor.Location = new Point(4, 528);

}

//------------------------------------------------------------------------------------

}

}

### 7.2 Mağaza Otomasyon Programının Müşteri Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

using System.Net.Configuration;

namespace melihkocak0385

{

public partial class frm\_musteri : Form

{

db d = new db();

frm f = new frm();

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-I6L92IT\SQLEXPRESS03;Initial Catalog=IlkOtomasyon;Integrated Security=True"); //database bağlantısı

public frm\_musteri()

{

InitializeComponent();

}

//

//Formun Loadı

//

private void frm\_musteri\_Load(object sender, EventArgs e)

{

this.tBL\_MUSTERİTableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_MUSTERİ); //Datagridviewi form yüklenirken doldurmak için

cmbx\_ara.SelectedIndex = 0; //arama kutusunun ilk aşamada seçili gelmesi için

ToolTip aciklama = new ToolTip(); //Tooltip için nesne türettim

aciklama.SetToolTip(btn\_sıfırla, "Aşağıda Yazılı Olan Bilgileri Siler(Herhangi Bir Veri Kaybı Yaşanmaz)"); //Textboxların üstündeki küçük buton için tooltip

error\_no.BlinkStyle = ErrorBlinkStyle.NeverBlink; //ErrorProviderın blink style ayarı

error\_ad.BlinkStyle = ErrorBlinkStyle.NeverBlink; //ErrorProviderın blink style ayarı

error\_soyad.BlinkStyle = ErrorBlinkStyle.NeverBlink; //ErrorProviderın blink style ayarı

}

//

//Müşteri Listeleme Butonu

//

private void btn\_listele\_Click(object sender, EventArgs e)

{

d.listele("Select \* from TBL\_MUSTERİ", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

}

//

//Müşteri Ekleme Butonu

//

private void btn\_ekle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Kayıt Eklerken Primary Alanının Boş Girilemeyeceği için textboxın boş girilmesini engelledim.

{

MessageBox.Show("Lütfen Gerekli Alanları Doldurunuz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez."); //ErrorProvdier İle uyarıyoruz.

error\_ad.SetError(txt\_soyad, "Bu Alan Boş Geçilemez."); //ErrorProvdier İle uyarıyoruz.

error\_soyad.SetError(txt\_ad, "Bu Alan Boş Geçilemez."); //ErrorProvdier İle uyarıyoruz.

}

else

{

d.prkeykontrol("prkeykontrolmusteri", txt\_no);//Primary key alanına aynı veri iki kere girilmediği için db classındaki prkeykontrol metodu ile alanı kontrol ediyorum

int a = Convert.ToInt32(d.y);//db classındaki primary key kontorlü için değişken oluşturmuştum onu int veri tipine çeviriyorum.

if (a == 1)//kayıt etkilendiyse yani kayıt varsa hata mesajı verecek.

{

MessageBox.Show("Lütfen No Alanını Daha Önce Kullanılmamış Bir Değer Giriniz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

}

else

{

d.eklegunsil("insert into TBL\_MUSTERİ values(@p1,@p2,@p3,@p4,@p5)", txt\_no, txt\_ad, txt\_soyad, txt\_tel, txt\_mail); //Eğer ilgili textbox boş değilse db classındaki eklegünsil metodu textboxlara yazılan değerleri içine gönderdiğim sql kodu ile işleyecek.

if (d.x != 0)

{

d.listele("Select \* from TBL\_MUSTERİ", dataGridView1); //Db classındaki x değişkeni row affectedi tutuyor 0 dan farklı ise işlemleri gerçekleştiriyor

MessageBox.Show("Ekleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

}

}

}

}

//

//Müşteri Güncelle Butonu

//

private void btn\_guncelle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Kayıt Güncellerken Primary Alanının Boş Girilemeyeceği için textboxın boş girilmesini engelledim.

{

MessageBox.Show("Müşteri Numarasına Göre Güncelleme İşlemi Yapılacaktır 'Müşteri No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu alan boş geçilemez"); //ErrorProvdier İle uyarıyoruz.

}

else

{

d.eklegunsil("update TBL\_MUSTERİ set M\_Ad=@p1,M\_Soyad=@p2,M\_Tel=@p3,M\_Mail=@p4 where M\_No=@p5", txt\_ad, txt\_soyad, txt\_tel, txt\_mail, txt\_no); //db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod

if (d.x == 0) //Olmayan bir kaydı güncellemeyi engelliyoruz.

MessageBox.Show("Olmayan Bir Kayıdı Güncellemeye Çalıştınız. Lütfen Dikkat Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

else

{

d.listele("Select \* from TBL\_MUSTERİ", dataGridView1); //Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değer gösterecek

MessageBox.Show("Güncelleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Müşteri Silme Butonu

//

private void btn\_sil\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Müşteri Numarasına Göre Silinme İşlemi Yapılacaktır 'Müşteri No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez."); //Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("delete from TBL\_MUSTERİ where M\_No=@p1", txt\_no); //db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod

if (d.x == 0) //Olmayan bir kaydı silmeyi engelliyoruz.

{

MessageBox.Show("Olmayan Bir Kayıdı Silmeye Çalıştınız. Gerekli Alanları Kontrol Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez."); //Errorprovdier ile uyarıyoruz.

}

else

{

MessageBox.Show("Silme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

d.listele("Select \* from TBL\_MUSTERİ", dataGridView1); //Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değer gösterecektir.

f.sifirla(txt\_no, txt\_ad, txt\_soyad, txt\_mail, txt\_tel);//sildikten sonra textboxları sıfırlıyoruz.

}

}

}

//

//Müşteri Ara ComboBox SelectedIndexChanged

//

private void cmbx\_ara\_SelectedIndexChanged(object sender, EventArgs e) //Kulanıcın kafası karışmasını engellemek amaçlı yapılmıştır.

{

if (cmbx\_ara.SelectedIndex == 0) //Comboboxın SelectedIndex 0 Yani 'Seçiniz' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_soyad.Visible = true;

txt\_tel.Visible = true;

txt\_mail.Visible = true;

txt\_no.Visible = true;

txt\_ad.Visible = true;

btn\_ekle.Enabled = true;

btn\_guncelle.Enabled = true;

btn\_listele.Enabled = true;

btn\_sil.Enabled = true;

btn\_ara.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 1) //Comboboxın SelectedIndex 1 Yani 'No'ya Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_soyad.Visible = false;

txt\_tel.Visible = false;

txt\_mail.Visible = false;

txt\_no.Visible = true;

txt\_ad.Visible = false;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 2) //Comboboxın SelectedIndex 2 Yani 'Ad'a Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_soyad.Visible = false;

txt\_tel.Visible = false;

txt\_mail.Visible = false;

txt\_no.Visible = false;

txt\_ad.Visible = true;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

}

}

//

//Müşteri Arama Butonu

//

private void btn\_ara\_Click(object sender, EventArgs e)

{

if (cmbx\_ara.SelectedIndex==1) // Birinci seçenek seçili ise No ya göre ara yapılacak

d.ara("select \* from TBL\_MUSTERİ where M\_No = @p1", txt\_no, dataGridView1); //db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak No'ya göre arama işlemini gerçekleştiriyor.

else if (cmbx\_ara.SelectedIndex == 2)

d.ara("select \* from TBL\_MUSTERİ where RTRIM(M\_Ad) LIKE '%"+txt\_ad.Text+"%'", txt\_no, dataGridView1); //db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak Ada göre arama işlemini gerçekleştiriyor.

}

//

//TextBoxları Sıfırlama Butonu

//

private void btn\_sıfırla\_Click(object sender, EventArgs e)

{

f.sifirla(txt\_no, txt\_ad, txt\_soyad, txt\_mail,txt\_tel); //db classındaki 'sifirla' metodu parametre olarak verdiğimiz textboxları sıfırlar

}

//

//İşlemler Ve İstatistikler Arası Geçiş Segmenti

//

private void sgmt\_musteri\_IndexChanged(object sender, EventArgs e) //Burada bir trick kullanılmıştır ilk başta aktif olan işlemler panelidir eğer segment değiştirilirse işlemler paneli gizlenir istatistik paneli gösterilir.

{

if (sgmt\_musteri.SelectedIndex == 0) //XanderUI sınıfından segment aracının seçili değeri 0 ise işlemler panelini(ekleme, silme, güncelleme....) Gösteriyoruz.

{

pnl\_istatistik.Hide();

pnl\_islem.Show();

chart1.Series.Remove(chart1.Series["Müşteri Sayısı"]); //Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoruz.

}

else if (sgmt\_musteri.SelectedIndex == 1) ////XanderUI sınıfından segment aracının seçili değeri 1 ise istatistikler panelini Gösteriyoruz.

{

d.istatistiky("Select count(M\_No) from TBL\_MUSTERİ", chart1, "Müşteri Sayısı"); //İstatistikler segmentine geçince db classındaki istatistiky metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

pnl\_islem.Hide();

pnl\_istatistik.Show();

pnl\_istatistik.Location = pnl\_islem.Location;

}

}

//

//DataGridView CellClick

//

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

d.tasima(txt\_no, txt\_ad, txt\_soyad, txt\_tel, txt\_mail, dataGridView1, e); //db classındaki 'tasima' metodu datagridviewin hücrelerine tıklayınca değerleri ilgili textboxlara taşıma işlemini yapar

}

//--------------ToolStrip Öğelerinin Click Eventi------------------------------------

private void tumunusectoolStripMenuItem1\_Click(object sender, EventArgs e) //Textboxlardaki verinin tümünü seçmek için ContextMenuStrip aracının 'tumunusec' öğesinin click eventi

{

f.tumunusec(this); //frm classındaki tumunusec metodu ile textboxlardaki verinin tümünü seçiyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kopyalaToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kopyalamak için ContextMenuStrip aracının 'kopyala' öğesinin click eventi

{

f.kopyala(this); //frm classındaki kopyala metodu ile textboxlardaki veriyi kopyalama işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void yapistirToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlara veriyi yapıştırmak için ContextMenuStrip aracının 'yapistir' öğesinin click eventi

{

f.yapistir(this); //frm classındaki yapistir metodu ile textboxlara panodaki veriyi yapıştırıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kesToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kesmek için ContextMenuStrip aracının 'kes' öğesinin click eventi

{

f.kes(this); //frm classındaki kes metodu ile textboxlardaki veriyi kesme işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void çıkışToolStripMenuItem\_Click(object sender, EventArgs e) //Formdan sağ tık yaparak çıkmak için ContextMenuStrip aracının 'cikis' öğesinin click eventi

{

this.Close(); //İlgili formu kapatıyoruz.

}

//--------------------------------------------------------------------------------------

//-----Error Provdilerin Silinmesi İçin Gerekli Eventler--------

private void txt\_no\_TextChanged(object sender, EventArgs e)

{

if (txt\_no.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_no.Clear();

}

private void txt\_ad\_TextChanged(object sender, EventArgs e)

{

if (txt\_ad.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_ad.Clear();

}

private void txt\_soyad\_TextChanged(object sender, EventArgs e)

{

if (txt\_soyad.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_soyad.Clear();

}

//--------------------------------------------------------------------------------------

//--TextBoxlara Yanlış Veri Girişini Engellemek İçin Gerekli Eventler--

private void txt\_no\_KeyPress(object sender, KeyPressEventArgs e) //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}

private void txt\_tel\_KeyPress(object sender, KeyPressEventArgs e) //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}

private void txt\_ad\_KeyPress(object sender, KeyPressEventArgs e)//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece metinsel ifadelere izin veriyor.

{

e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) && !char.IsSeparator(e.KeyChar);

}

private void txt\_soyad\_KeyPress(object sender, KeyPressEventArgs e) //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece metinsel ifadelere izin veriyor.

{

e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) && !char.IsSeparator(e.KeyChar);

}

//---------------------------------------------------------------------

}

}

### 7.3 Mağaza Otomasyon Programının Sipariş Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace melihkocak0385

{

public partial class frm\_siparis : Form

{

db d = new db();

frm f = new frm();

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-I6L92IT\SQLEXPRESS03;Initial Catalog=IlkOtomasyon;Integrated Security=True"); //database bağlantısı

public frm\_siparis()

{

InitializeComponent();

}

//

//Formun Loadı

//

private void frm\_siparis\_Load(object sender, EventArgs e)

{

dateTimePicker1.Enabled = false;//Satış Tarihinde oynama yapılırsa veri kaybı olur diye DateTimePicker ı enabled false yaptım

error\_sno.BlinkStyle = ErrorBlinkStyle.NeverBlink;//ErrorProviderın blink style ayarı

error\_mno.BlinkStyle = ErrorBlinkStyle.NeverBlink;//ErrorProviderın blink style ayarı

error\_uno.BlinkStyle = ErrorBlinkStyle.NeverBlink;//ErrorProviderın blink style ayarı

ToolTip aciklama = new ToolTip();//Tooltip için nesne türettim

aciklama.SetToolTip(btn\_sifirla, "Aşağıda Yazılı Olan Bilgileri Siler(Herhangi Bir Veri Kaybı Yaşanmaz)"); //textboxların üstündeki küçük butonun açıklaması

cmbx\_ara.SelectedIndex = 0; //arama kutusunun ilk aşamada seçili gelmesi için

}

//

//Sipariş Ekleme Butonu

//

private void btn\_ekle\_Click(object sender, EventArgs e)

{

if (txt\_sno.Text == null || txt\_sno.Text == "" || txt\_mno.Text == "" || txt\_adet.Text == "") //Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Lütfen Gerekli Alanları Doldurunuz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_sno.SetError(txt\_sno, "Bu alan boş geçilemez");//ErrorProvdier İle uyarıyoruz.

error\_mno.SetError(txt\_mno, "Bu alan boş geçilemez");//ErrorProvdier İle uyarıyoruz.

error\_uno.SetError(txt\_uno, "Bu alan boş geçilemez");//ErrorProvdier İle uyarıyoruz.

}

else

{

d.prkeykontrol("prkeykontrolsiparis", txt\_sno);//Primary key alanına aynı veri iki kere girilmediği için db classındaki prkeykontrol metodu ile alanı kontrol ediyorum

int a = Convert.ToInt32(d.y);//db classındaki primary key kontorlü için değişken oluşturmuştum onu int veri tipine çeviriyorum.

if (a == 1)//kayıt etkilendiyse yani kayıt varsa hata mesajı verecek.

{

MessageBox.Show("Lütfen No Alanını Daha Önce Kullanılmamış Bir Değer Giriniz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

}

else

{

d.toplamfiyat(txt\_bfiyat, txt\_adet, txt\_tfiyat);//Sipariş Oluşturulurken 'txt\_bfiyat' ve 'txt\_adet' alanlarına değer girilince bu değerleri çarpıp 'txt\_tfiyat' textbox sına yazdıran metot 'txt\_tfiyat textboxı visible özelliği false dır kullanıcı görmez

d.eklegunsil("insert into TBL\_SIPARIS (S\_Tarih, S\_No, M\_No, S\_ToFiyat) values (@p1, @p2, @p3, @p4)", dateTimePicker1, txt\_sno, txt\_mno, txt\_tfiyat); //db classındaki eklegünsil metodu TextBox lara ve DateTimePicker a yazılan değerleri içine gönderdiğim sql kodu ile işleyecek.

d.eklegunsil("insert into Siparis\_Urun\_Detay(adet, U\_No, S\_No, BFiyat) values(@p1, @p2, @p3, @p4)", txt\_adet, txt\_uno, txt\_sno, txt\_bfiyat);//db classındaki eklegünsil metodu textboxlara yazılan değerleri içine gönderdiğim sql kodu ile işleyecek.

d.listele("select \* from TBL\_SIPARIS", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

d.listele("select \* from Siparis\_Urun\_Detay", dataGridView2);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

MessageBox.Show("Ekleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Sipariş Silme Butonu

//

private void btn\_sil\_Click(object sender, EventArgs e)

{

SqlCommand komut = new SqlCommand();

if (txt\_sno.Text == null || txt\_sno.Text == "")//Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Sipariş Numarasına Göre Silinme İşlemi Yapılacaktır 'Sipariş No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_sno.SetError(txt\_sno, "Bu Alan Boş Geçilemez.");//Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("delete from TBL\_SIPARIS where S\_No=@p1", txt\_sno);//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

if (d.x == 0)//Olmayan bir kaydı silmeyi engelliyoruz.

{

MessageBox.Show("Olmayan Bir Kayıdı Silmeye Çalıştınız. Gerekli Alanları Kontrol Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_sno.SetError(txt\_sno, "Bu Alan Boş Geçilemez.");//Errorprovdier ile uyarıyoruz.

}

else

{

d.listele("select \* from TBL\_SIPARIS", dataGridView1);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

f.sifirla(txt\_uno, txt\_sno, txt\_mno, txt\_adet, txt\_bfiyat);//frm classındaki 'sifirla' metodu içine parametre olarak verilen textboxları sıfırlar, silme işlemi bittikten sonra textboxları sıfırlıyoruz.

MessageBox.Show("Silme İşlemi Başarılı Bir Şekilde Gerçekleştirildi. ", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

d.eklegunsil("delete from Siparis\_Urun\_Detay where S\_No=@p1", txt\_sno);//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

d.listele("select \* from Siparis\_Urun\_Detay", dataGridView2);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

}

}

//

//Sipariş Listeleme Butonu

//

private void btn\_listele\_Click(object sender, EventArgs e)

{

d.listele("select \* from TBL\_SIPARIS", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

d.listele("select \* from Siparis\_Urun\_Detay", dataGridView2);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

}

//

//Textboxları Sıfırlama Butonu

//

private void btn\_sıfırla\_Click(object sender, EventArgs e)

{

f.sifirla(txt\_sno, txt\_mno, txt\_sno, txt\_adet, txt\_uno, txt\_bfiyat);//frm classındaki 'sifirla' metodu içine parametre olarak verilen textboxları sıfırlar, silme işlemi bittikten sonra textboxları sıfırlıyoruz.

}

//

//Sipariş Güncelleme Butonu

//

private void btn\_guncelle\_Click(object sender, EventArgs e)

{

SqlCommand komut = new SqlCommand();

if (txt\_sno.Text == "" || txt\_sno.Text == null) //Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Sipariş Numarasına Göre Güncelleme İşlemi Yapılacaktır 'Sipariş No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_sno.SetError(txt\_sno, "Bu alan boş geçilemez");//Errorprovdier ile uyarıyoruz.

}

else

{

d.toplamfiyat(txt\_bfiyat, txt\_adet, txt\_tfiyat);//Sipariş Oluşturulurken 'txt\_bfiyat' ve 'txt\_adet' alanlarına değer girilince bu değerleri çarpıp 'txt\_tfiyat' textbox sına yazdıran metot 'txt\_tfiyat textboxı visible özelliği false dır kullanıcı görmez

d.eklegunsil("update TBL\_SIPARIS set M\_No=@p2, S\_ToFiyat=@p3 where S\_No=@p1", txt\_sno, txt\_mno, txt\_tfiyat);//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

d.eklegunsil("update Siparis\_Urun\_Detay set U\_No=@p2, adet=@p3, BFiyat=@p4 where S\_No=@p1", txt\_sno, txt\_uno, txt\_adet, txt\_bfiyat);//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

if (d.x == 0)

MessageBox.Show("Olmayan Bir Kayıdı Güncellemeye Çalıştınız. Lütfen Dikkat Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else

{

d.listele("select \* from TBL\_SIPARIS", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

d.listele("select \* from Siparis\_Urun\_Detay", dataGridView2);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

MessageBox.Show("Güncelleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Sipariş Arama Butonu

//

private void btn\_ara\_Click(object sender, EventArgs e)

{

if (txt\_sno.Text == "")//Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("'Sipariş No' Alanını Kontrol Ediniz Boş Veya Yanlış Değer Girmediğinizden Emin Olun", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_sno.SetError(txt\_sno, "Bu alan boş geçilemez");// Errorprovdier ile uyarıyoruz.

}

d.ara("select \* from TBL\_SIPARIS where S\_No = @p1", txt\_sno, dataGridView1);//db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak No'ya göre arama işlemini gerçekleştiriyor.

d.ara("select \* from Siparis\_Urun\_Detay where S\_No = @p1", txt\_sno, dataGridView2);//db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak No'ya göre arama işlemini gerçekleştiriyor.

}

//

//Bir Sonraki Sipariş İçin Sipariş No Oluştur Butonu

//

private void btn\_olustur\_Click(object sender, EventArgs e)

{

SqlCommand komut = new SqlCommand(); //SqlCommand Sınıfından komut isminde bir nesne türettim

d.sonproc(komut);//Sipariş Formunda S\_No Alanını Otomatik Artma Şeklinde Yapmadığım İçin Otomatik Artmasını Sağlayan Kod Satırı

txt\_sno.Text = komut.Parameters["@deger"].Value.ToString(); //db classındaki 'sonproc' metodunda stored procuder kullanıldığı için oradan dönen değer '@deger' isminde bir değişkende tutuluyor '@deger' değişkeninin değerini stringe çevirerek 'txt\_sno' textboxsına yazdırdık.

baglanti.Close();

txt\_sno.Enabled = false;//başka değer girilmesin diye enabled özelliğini false yaptım

}

//-----Error Provdilerin Silinmesi İçin Gerekli Eventler--------

private void txt\_sno\_TextChanged(object sender, EventArgs e)

{

if (txt\_sno.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_sno.Clear();

}

private void txt\_mno\_TextChanged(object sender, EventArgs e)

{

if (txt\_mno.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_mno.Clear();

}

private void txt\_uno\_TextChanged(object sender, EventArgs e)

{

if (txt\_uno.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_uno.Clear();

}

//----------------------------------------------------------------------------------------

//----------------- İlgili ToolStrip Öğelerinin Click Eventi-------------------------------------

private void tumunusecToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki verinin tümünü seçmek için ContextMenuStrip aracının 'tumunusec' öğesinin click eventi

{

f.tumunusec(this); //frm classındaki tumunusec metodu ile textboxlardaki verinin tümünü seçiyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kopyalaToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kopyalamak için ContextMenuStrip aracının 'kopyala' öğesinin click eventi

{

f.kopyala(this); //frm classındaki kopyala metodu ile textboxlardaki veriyi kopyalama işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void yapistirToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlara veriyi yapıştırmak için ContextMenuStrip aracının 'yapistir' öğesinin click eventi

{

f.yapistir(this); //frm classındaki yapistir metodu ile textboxlara panodaki veriyi yapıştırıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kesToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kesmek için ContextMenuStrip aracının 'kes' öğesinin click eventi

{

f.kes(this); //frm classındaki kes metodu ile textboxlardaki veriyi kesme işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void cikisToolStripMenuItem\_Click(object sender, EventArgs e) //Formdan sağ tık yaparak çıkmak için ContextMenuStrip aracının 'cikis' öğesinin click eventi

{

this.Close(); //İlgili formu kapatıyoruz.

}

//----------------------------------------------------------------------------------------

//

//İşlemler Ve İstastistikler Arası Geçiş Segmenti

//

private void sgmt\_siparis\_IndexChanged(object sender, EventArgs e)//Burada bir trick kullanılmıştır ilk başta aktif olan işlemler panelidir eğer segment değiştirilirse işlemler paneli gizlenir istatistik paneli gösterilir.

{

if (sgmt\_siparis.SelectedIndex == 0)//XanderUI sınıfından segment aracının seçili değeri 0 ise işlemler panelini(ekleme, silme, güncelleme....) Gösteriyoruz.

{

pnl\_istatistik.Hide();

pnl\_islem.Show();

chrt\_tumsiparis.Series.Remove(chrt\_tumsiparis.Series["Sipariş Sayısı"]);//Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoru

chrt\_gunsiparis.Series.Remove(chrt\_gunsiparis.Series["Sipariş Sayısı"]);//Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoru

}

else if (sgmt\_siparis.SelectedIndex == 1)

{

d.istatistiky("select count(S\_No) from TBL\_SIPARIS", chrt\_tumsiparis, "Sipariş Sayısı");//İstatistikler segmentine geçince db classındaki istatistiky metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

d.istatistiky("select count(S\_No) from TBL\_SIPARIS where S\_Tarih='"+dateTimePicker1.Text+"'", chrt\_gunsiparis, "Sipariş Sayısı");//İstatistikler segmentine geçince db classındaki istatistiky metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

pnl\_islem.Hide();

pnl\_istatistik.Show();

pnl\_istatistik.Location = pnl\_islem.Location;

}

}

//

//Sipariş Ara Comboboxı SelectedIndexChanged

//

private void cmbx\_ara\_SelectedIndexChanged(object sender, EventArgs e)//Arama İşlemi Yapılırken Kulanıcın kafası karışmasını engellemek amaçlı yapılmıştır.

{

if (cmbx\_ara.SelectedIndex == 0)//Comboboxın SelectedIndex 0 Yani 'Seçiniz' ise aşağıdaki işlemler gerçekleştirilecek

{

dateTimePicker1.Visible = true;

txt\_mno.Visible = true;

txt\_adet.Visible = true;

txt\_sno.Visible = true;

txt\_uno.Visible = true;

btn\_ekle.Enabled = true;

btn\_guncelle.Enabled = true;

btn\_listele.Enabled = true;

btn\_sil.Enabled = true;

btn\_ara.Enabled = false;

txt\_bfiyat.Visible = true;

}

else //Comboboxın SelectedIndex 1 Yani 'No'ya Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

dateTimePicker1.Visible = false;

txt\_mno.Visible = false;

txt\_adet.Visible = false;

txt\_sno.Visible = true;

txt\_uno.Visible = false;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

txt\_bfiyat.Visible = false;

txt\_sno.Enabled = true;

}

}

//

//DataGridView CellClick(Satırlara Tıklayınca Verileri İlgili Textboxlara Taşımak İçin)

//

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

d.tasima(txt\_sno, txt\_mno, dateTimePicker1, dataGridView1, e);//db classındaki 'tasima' metodu datagridviewin hücrelerine tıklayınca değerleri ilgili textboxlara taşıma işlemini yapar.

d.tasi(txt\_sno, txt\_uno, txt\_adet, txt\_mno,txt\_bfiyat);//datagridviewdeki satıra tıklayınca veriyi ilgili textboxa taşımak için yazılmıştır ama bu metotta stored procedure kullanılmıştır. stored procedure kullanma amacım ise 'Siparis\_Urun\_Detay' tablosunda bulunmayan M\_No,S\_ToFiyat ve S\_Tarih kolonları 'TBL\_SIPARIS' Tablosunda herhangi bir satıra tıklayınca gelmiyor. Bu sorunu çözmek için yazdım

}

catch (Exception)

{

}

}

//

//DataGridView CellClick(Satırlara Tıklayınca Verileri İlgili Textboxlara Taşımak İçin)

//

private void dataGridView2\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

d.tasima(txt\_uno, txt\_sno, txt\_adet, txt\_bfiyat, dataGridView2, e);//db classındaki 'tasima' metodu datagridviewin hücrelerine tıklayınca değerleri ilgili textboxlara taşıma işlemini yapar.

d.tasi(txt\_sno, txt\_uno, txt\_adet, txt\_mno, txt\_bfiyat);//datagridviewdeki satıra tıklayınca veriyi ilgili textboxa taşımak için yazılmıştır ama bu metotta stored procedure kullanılmıştır. stored procedure kullanma amacım ise 'Siparis\_Urun\_Detay' tablosunda bulunmayan M\_No,S\_ToFiyat ve S\_Tarih kolonları 'TBL\_SIPARIS' Tablosunda herhangi bir satıra tıklayınca gelmiyor. Bu sorunu çözmek için yazdım

}

catch (Exception)

{

}

}

}

}

### 7.4 Mağaza Otomasyon Programının Ürün Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace melihkocak0385

{

public partial class frm\_urun : Form

{

db d = new db();

frm f = new frm();

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-I6L92IT\SQLEXPRESS03;Initial Catalog=IlkOtomasyon;Integrated Security=True");

public frm\_urun()

{

InitializeComponent();

}

//

//Formun Loadı

//

private void frm\_ürünler\_Load(object sender, EventArgs e)

{

this.tBL\_URUNTableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_URUN);//Datagridviewi form yüklenirken doldurmak için

this.tBL\_TEDARIKCILERTableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_TEDARIKCILER);//Datagridviewi form yüklenirken doldurmak için

this.tBL\_KATEGORITableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_KATEGORI);//Datagridviewi form yüklenirken doldurmak için

this.tBL\_URUNTableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_URUN);//Datagridviewi form yüklenirken doldurmak için

cmbx\_ara.SelectedIndex = 0; //arama kutusunun ilk aşamada seçili gelmesi için

ToolTip aciklama = new ToolTip();//Tooltip için nesne türettim

aciklama.SetToolTip(btn\_sıfırla, "Aşağıda Yazılı Olan Bilgileri Siler(Herhangi Bir Veri Kaybı Yaşanmaz)");//Textboxların üstündeki küçük buton için tooltip

}

//

//Ürün Listeleme Butonu

//

private void btn\_listele\_Click(object sender, EventArgs e)

{

d.listele("select \* from TBL\_URUN",dataGridView1); //db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

}

//

//Ürün Ekleme Butonu

//

private void btn\_ekle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "")//Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Lütfen Gerekli Alanları Doldurunuz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez");//Errorprovdier ile uyarıyoruz.

}

else

{

d.prkeykontrol("prkeykontrolurun", txt\_no); //Primary key alanına aynı veri iki kere girilmediği için db classındaki prkeykontrol metodu ile alanı kontrol ediyorum

int a = Convert.ToInt32(d.y); //db classındaki primary key kontorlü için değişken oluşturmuştum onu int veri tipine çeviriyorum.

if (a == 1) //kayıt etkilendiyse yani kayıt varsa hata mesajı verecek.

{

MessageBox.Show("Lütfen No Alanını Daha Önce Kullanılmamış Bir Değer Giriniz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

}

else

{

d.eklegunsil("insert into TBL\_URUN values(@p1,@p2,@p3,@p4,@p5,@p6,@p7)", txt\_no, txt\_ad, txt\_alis, txt\_satis, cmbx\_kategori, cmbx\_tedarik, txt\_stok);//Eğer ilgili textbox boş değilse db classındaki eklegünsil metodu textboxlara yazılan değerleri içine gönderdiğim sql kodu ile işleyecek.

if (d.x != 0)//Db classındaki x değişkeni row affectedi tutuyor 0 dan farklı ise işlemleri gerçekleştiriyor

{

d.listele("select \* from TBL\_URUN", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

MessageBox.Show("Ekleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

}

}

}

//

//Ürün Silme Butonu

//

private void btn\_sil\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "")//Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Ürün Numarasına Göre Silinme İşlemi Yapılacaktır 'Ürün No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez.");//Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("delete from TBL\_URUN where U\_No=@p1", txt\_no);//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

if (d.x == 0)

{

MessageBox.Show("Olmayan Bir Kayıdı Silmeye Çalıştınız Gerekli Alanları Kontrol Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez");//Errorprovdier ile uyarıyoruz.

}

else

{

f.sifirla(txt\_no, txt\_ad, txt\_satis, txt\_alis,txt\_stok);//sildikten sonra textboxları sıfırlıyoruz.

d.listele("select \* from TBL\_URUN", dataGridView1);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

MessageBox.Show("Silme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Textboxları Sıfırlama Butonu

//

private void btn\_sıfırla\_Click(object sender, EventArgs e)

{

f.sifirla(txt\_ad, txt\_alis, txt\_no, txt\_satis,txt\_stok);//db classındaki 'sifirla' metodu parametre olarak verdiğimiz textboxları sıfırlar

}

//

//Ürün Güncelleme Butonu

//

private void btn\_guncelle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text== "")//Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Ürün Numarasına Göre Güncelleme İşlemi Yapılacaktır 'Ürün No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez");//Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("update TBL\_URUN set U\_Ad=@p2,U\_Alis=@p3,U\_Satis=@p4,K\_No=@p5,T\_No=@p6, U\_Stok = @p7 where U\_No=@p1 ", txt\_no, txt\_ad, txt\_alis, txt\_satis, cmbx\_kategori, cmbx\_tedarik,txt\_stok );//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod

if (d.x == 0)//Olmayan bir kaydı güncellemeyi engelliyoruz.

MessageBox.Show("Olmayan Bir Kayıdı Güncellemeye Çalıştınız. Lütfen Dikkat Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

else

{

d.listele("SELECT \* FROM TBL\_URUN", dataGridView1);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değer gösterecek

MessageBox.Show("Güncelleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

}

d.listele("select \* from TBL\_URUN", dataGridView1);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değer gösterecek

}

}

//

//Ürün Arama Butonu

//

private void btn\_ara\_Click(object sender, EventArgs e)

{

if (cmbx\_ara.SelectedIndex == 1)//Comboboxda Birinci seçenek seçili ise No ya göre ara yapılacak

{

d.ara("select \* from TBL\_URUN where U\_No = @p1", txt\_no, dataGridView1);//db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak No'ya göre arama işlemini gerçekleştiriyor.

}

else if (cmbx\_ara.SelectedIndex == 2)//Comboboxda ikinci seçenek seçili ise Ada ya göre ara yapılacak

{

d.ara("select \* from TBL\_URUN where RTRIM(U\_Ad) LIKE '%" + txt\_ad.Text + "%'", txt\_ad, dataGridView1);//db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak Ada göre arama işlemini gerçekleştiriyor.

}

}

//----------------- İlgili ToolStrip Öğelerinin Click Eventi-------------------------------------

private void kopyalaToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kopyalamak için ContextMenuStrip aracının 'kopyala' öğesinin click eventi

{

f.kopyala(this); //frm classındaki kopyala metodu ile textboxlardaki veriyi kopyalama işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void yapistirToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlara veriyi yapıştırmak için ContextMenuStrip aracının 'yapistir' öğesinin click eventi

{

f.yapistir(this); //frm classındaki yapistir metodu ile textboxlara panodaki veriyi yapıştırıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kesToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kesmek için ContextMenuStrip aracının 'kes' öğesinin click eventi

{

f.kes(this); //frm classındaki kes metodu ile textboxlardaki veriyi kesme işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void çıkışToolStripMenuItem\_Click(object sender, EventArgs e) //Formdan sağ tık yaparak çıkmak için ContextMenuStrip aracının 'cikis' öğesinin click eventi

{

this.Close(); //İlgili formu kapatıyoruz.

}

private void tumunusectoolStripMenuItem1\_Click(object sender, EventArgs e)

{

f.tumunusec(this);

}

//----------------------------------------------------------------------------------------

//-----Error Provdilerin Silinmesi İçin Gerekli Eventler--------

private void txt\_no\_TextChanged(object sender, EventArgs e)

{

if (txt\_no.Text != "") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_no.Clear();

}

//---------------------------------------------------------------

//--TextBoxlara Yanlış Veri Girişini Engellemek İçin Gerekli Eventler--

private void txt\_no\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

} //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_alis\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

} //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_satis\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

} //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_stok\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

} //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

//-----------------------------------------------------------------------

//

//Ürün Ara Comboboxı SelectedIndexChanged

//

private void cmbx\_ara\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmbx\_ara.SelectedIndex == 0)//Comboboxın SelectedIndex 0 Yani 'Seçiniz' ise aşağıdaki işlemler gerçekleştirilecek

{

cmbx\_tedarik.Visible = true;

cmbx\_kategori.Visible = true;

txt\_alis.Visible = true;

txt\_satis.Visible = true;

txt\_no.Visible = true;

txt\_ad.Visible = true;

btn\_ekle.Enabled = true;

btn\_guncelle.Enabled = true;

btn\_listele.Enabled = true;

btn\_sil.Enabled = true;

txt\_stok.Visible = true;

btn\_ara.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 1)//Comboboxın SelectedIndex 1 Yani 'No'ya Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

cmbx\_tedarik.Visible = false;

cmbx\_kategori.Visible = false;

txt\_alis.Visible = false;

txt\_satis.Visible = false;

txt\_no.Visible = true;

txt\_ad.Visible = false;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

txt\_stok.Visible = false;

}

else if (cmbx\_ara.SelectedIndex == 2) //Comboboxın SelectedIndex 2 Yani 'Ad'a Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

cmbx\_tedarik.Visible = false;

cmbx\_kategori.Visible = false;

txt\_alis.Visible = false;

txt\_satis.Visible = false;

txt\_no.Visible = false;

txt\_ad.Visible = true;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

txt\_stok.Visible = false;

}

}

//

//İşlemler Ve İstastistikler Arası Geçiş Segmenti

//

private void sgmt\_urun\_IndexChanged(object sender, EventArgs e)//Burada bir trick kullanılmıştır ilk başta aktif olan işlemler panelidir eğer segment değiştirilirse işlemler paneli gizlenir istatistik paneli gösterilir.

{

if (sgmt\_urun.SelectedIndex == 0)//XanderUI sınıfından segment aracının seçili değeri 0 ise işlemler panelini(ekleme, silme, güncelleme....) Gösteriyoruz

{

pnl\_istatistik.Hide();

pnl\_islem.Show();

chrt\_enpahalı.Series.Remove(chrt\_enpahalı.Series["En Pahalı Ürün"]);//Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoruz.

chrt\_enucuz.Series.Remove(chrt\_enucuz.Series["En Ucuz Ürün"]);//Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoruz.

chrt\_tedarik.Series.Remove(chrt\_tedarik.Series["Ürün Sayısı"]);//Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoruz.

}

else if (sgmt\_urun.SelectedIndex == 1)//XanderUI sınıfından segment aracının seçili değeri 1 ise istatistikler panelini gösteriyoruz.

{

pnl\_islem.Hide();

pnl\_istatistik.Location = pnl\_islem.Location;

pnl\_istatistik.Show();

d.istatistikxy("select U\_Ad,U\_Satis from TBL\_URUN where U\_Satis=(select MAX(U\_Satis) from TBL\_URUN)", chrt\_enpahalı, "En Pahalı Ürün");//İstatistikler segmentine geçince db classındaki istatistikxy metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

d.istatistikxy("select U\_Ad,U\_Satis from TBL\_URUN where U\_Satis=(select Min(U\_Satis) from TBL\_URUN)", chrt\_enucuz, "En Ucuz Ürün");//İstatistikler segmentine geçince db classındaki istatistikxy metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

d.istatistikxy("select t\_Ad, count(\*) from TBL\_URUN inner join TBL\_TEDARIKCILER on TBL\_URUN.T\_No = TBL\_TEDARIKCILER.T\_No group by T\_Ad", chrt\_tedarik, "Ürün Sayısı");//İstatistikler segmentine geçince db classındaki istatistikxy metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

}

}

//

//DataGridView CellClick(Satırlara Tıklayınca Verileri İlgili Textboxlara Taşımak İçin)

//

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

d.tasima(txt\_no, txt\_ad, txt\_alis, txt\_satis, cmbx\_kategori, cmbx\_tedarik, txt\_stok, dataGridView1, e);//db classındaki 'tasima' metodu datagridviewin hücrelerine tıklayınca değerleri ilgili textboxlara taşıma işlemini yapar.

}

}

}

### 7.5 Mağaza Otomasyon Programının Tedarikçiler Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace melihkocak0385

{

public partial class frm\_tedarikciler : Form

{

db d = new db();

frm f = new frm();

public frm\_tedarikciler()

{

InitializeComponent();

}

//

//Formun Loadı

//

private void frm\_tedarikciler\_Load(object sender, EventArgs e)

{

ToolTip aciklama = new ToolTip(); //Tooltip için nesne türettim

aciklama.SetToolTip(btn\_sifirla, "Aşağıda Yazılı Olan Bilgileri Siler(Herhangi Bir Veri Kaybı Yaşanmaz)"); //Textboxların üstündeki küçük buton için tooltip

this.tBL\_TEDARIKCILERTableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_TEDARIKCILER); //Datagridviewi form yüklenirken doldurmak için

cmbx\_ara.SelectedIndex = 0;

}

//

//Tedarikçi Silme Butonu

//

private void btn\_sil\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Tedarikçi Numarasına Göre Silinme İşlemi Yapılacaktır 'Tedarikçi No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez."); //Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("delete FROM TBL\_TEDARIKCILER where T\_No=@p1", txt\_no); //db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

if (d.x == 0) //Olmayan bir kaydı silmeyi engelliyoruz.

{

MessageBox.Show("Olmayan Bir Kayıdı Silmeye Çalıştınız. Gerekli Alanları Kontrol Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez"); //Errorprovdier ile uyarıyoruz.

}

else

{

f.sifirla(txt\_no, txt\_ad, txt\_adres, txt\_mail, txt\_tel); //sildikten sonra textboxları sıfırlıyoruz.

d.listele("SELECT \* FROM TBL\_TEDARIKCILER", dataGridView1); //Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

MessageBox.Show("Silme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Tedarikçi Listeleme Butonu

//

private void btn\_listele\_Click(object sender, EventArgs e)

{

d.listele("SELECT \* FROM TBL\_TEDARIKCILER", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

}

//

//Tedarikçi Ekleme Butonu

//

private void btn\_ekle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Kayıt Eklerken Primary Alanının Boş Girilemeyeceği için textboxın boş girilmesini engelledim.

{

MessageBox.Show("Lütfen Gerekli Alanları Doldurunuz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez");//ErrorProvdier İle uyarıyoruz.

error\_ad.SetError(txt\_ad, "Bu Alan Boş Geçilemez");//ErrorProvdier İle uyarıyoruz.

error\_adres.SetError(txt\_adres, "Bu Alan Boş Geçilemez.");//ErrorProvdier İle uyarıyoruz.

}

else

{

d.prkeykontrol("prkeykontrol",txt\_no);//Primary key alanına aynı veri iki kere girilmediği için db classındaki prkeykontrol metodu ile alanı kontrol ediyorum

int a = Convert.ToInt32(d.y);//db classındaki primary key kontorlü için değişken oluşturmuştum onu int veri tipine çeviriyorum.

if (a == 1)//kayıt etkilendiyse yani kayıt varsa hata mesajı verecek.

{

MessageBox.Show("Lütfen No Alanını Daha Önce Kullanılmamış Bir Değer Giriniz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

}

else

{

d.eklegunsil("INSERT INTO TBL\_TEDARIKCILER values(@p1,@p2,@p3,@p4,@p5)", txt\_no, txt\_ad, txt\_adres, txt\_tel, txt\_mail);//Eğer ilgili textbox boş değilse db classındaki eklegünsil metodu textboxlara yazılan değerleri içine gönderdiğim sql kodu ile işleyecek.

if (d.x != 0)//Db classındaki x değişkeni row affectedi tutuyor 0 dan farklı ise işlemleri gerçekleştiriyor

{

d.listele("SELECT \* FROM TBL\_TEDARIKCILER", dataGridView1);//db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

MessageBox.Show("Ekleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

}

}

}

//

//Tedarikçi Güncelleme Butonu

//

private void btn\_guncelle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "")//Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Tedarikçi Numarasına Göre Güncelleme İşlemi Yapılacaktır 'Tedarikçi No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez");//Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("UPDATE TBL\_TEDARIKCILER set T\_Ad=@p1,T\_Adres=@p2,T\_Tel=@p3,T\_Mail=@p4 where T\_No=@p5", txt\_ad, txt\_adres, txt\_tel, txt\_mail, txt\_no);//db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod

if (d.x == 0)//Olmayan bir kaydı güncellemeyi engelliyoruz.

MessageBox.Show("Olmayan Bir Kayıdı Güncellemeye Çalıştınız. Lütfen Dikkat Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else

{

d.listele("SELECT \* FROM TBL\_TEDARIKCILER", dataGridView1);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değer gösterecek

MessageBox.Show("Güncelleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Tedarikçi Arama Butonu

//

private void btn\_ara\_Click(object sender, EventArgs e)

{

if (cmbx\_ara.SelectedIndex == 1)//Comboboxda Birinci seçenek seçili ise No ya göre ara yapılacak

{

d.ara("SELECT \* FROM TBL\_TEDARIKCILER where T\_No = @p1", txt\_no, dataGridView1); //db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak No'ya göre arama işlemini gerçekleştiriyor.

}

else if (cmbx\_ara.SelectedIndex == 2)//Comboboxda ikinci seçenek seçili ise Ada ya göre ara yapılacak

{

d.ara("SELECT \* FROM TBL\_TEDARIKCILER where RTRIM(T\_Ad) LIKE '%" + txt\_ad.Text + "%'", txt\_no, dataGridView1);//db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak Ada göre arama işlemini gerçekleştiriyor.

}

}

//

//DataGridView CellClick(Satırlara Tıklayınca Verileri İlgili Textboxlara Taşımak İçin)

//

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

d.tasima(txt\_no, txt\_ad, txt\_adres, txt\_tel, txt\_mail, dataGridView1, e); //db classındaki 'tasima' metodu datagridviewin hücrelerine tıklayınca değerleri ilgili textboxlara taşıma işlemini yapar.

}

//

//Textboxları Sıfırlama Butonu

//

private void btn\_sifirla\_Click\_1(object sender, EventArgs e)

{

f.sifirla(txt\_no, txt\_ad, txt\_adres, txt\_tel, txt\_mail); //db classındaki 'sifirla' metodu parametre olarak verdiğimiz textboxları sıfırlar

}

//----------------- ToolStrip Öğelerinin Click Eventi-------------------------------------

private void kopyalaToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kopyalamak için ContextMenuStrip aracının 'kopyala' öğesinin click eventi

{

f.kopyala(this); //frm classındaki kopyala metodu ile textboxlardaki veriyi kopyalama işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void yapistirToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlara veriyi yapıştırmak için ContextMenuStrip aracının 'yapistir' öğesinin click eventi

{

f.yapistir(this); //frm classındaki yapistir metodu ile textboxlara panodaki veriyi yapıştırıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kesToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kesmek için ContextMenuStrip aracının 'kes' öğesinin click eventi

{

f.kes(this); //frm classındaki kes metodu ile textboxlardaki veriyi kesme işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void cikisToolStripMenuItem\_Click(object sender, EventArgs e) //Formdan sağ tık yaparak çıkmak için ContextMenuStrip aracının 'cikis' öğesinin click eventi

{

this.Close(); //İlgili formu kapatıyoruz.

}

private void tumunusectoolStripMenuItem1\_Click(object sender, EventArgs e)//Textboxlardaki verinin tümünü seçmek için ContextMenuStrip aracının 'tumunusec' öğesinin click eventi

{

f.tumunusec(this); //frm classındaki tumunusec metodu ile textboxlardaki verinin tümünü seçiyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

//----------------------------------------------------------------------------------------

//-----Error Provdilerin Silinmesi İçin Gerekli Eventler--------

private void txt\_no\_TextChanged(object sender, EventArgs e)

{

if (txt\_no.Text != "")// ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_no.Clear();

}

private void txt\_ad\_TextChanged(object sender, EventArgs e)

{

if (txt\_ad.Text != "")// ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_ad.Clear();

}

private void txt\_adres\_TextChanged(object sender, EventArgs e)

{

if (txt\_adres.Text != "")// ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_adres.Clear();

}

//----------------------------------------------------------------------------------------

//--TextBoxlara Yanlış Veri Girişini Engellemek İçin Gerekli Eventler--

private void txt\_no\_KeyPress(object sender, KeyPressEventArgs e)//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}

private void txt\_tel\_KeyPress(object sender, KeyPressEventArgs e)//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece metinsel ifadelere izin veriyor.

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}

//---------------------------------------------------------------------

//

//İşlemler Ve İstastistikler Arası Geçiş Segmenti

//

private void sgmt\_tedarikciler\_IndexChanged(object sender, EventArgs e)//Burada bir trick kullanılmıştır ilk başta aktif olan işlemler panelidir eğer segment değiştirilirse işlemler paneli gizlenir istatistik paneli gösterilir.

{

if (sgmt\_tedarikciler.SelectedIndex == 0) //XanderUI sınıfından segment aracının seçili değeri 0 ise işlemler panelini(ekleme, silme, güncelleme....) Gösteriyoruz

{

pnl\_istatistik.Hide();

pnl\_islem.Show();

chart1.Series.Remove(chart1.Series["Kayıtlı Tedarikçiler"]); //Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoruz.

}

else if (sgmt\_tedarikciler.SelectedIndex == 1)

{

d.istatistiky("SELECT COUNT(\*) FROM TBL\_TEDARIKCILER", chart1, "Kayıtlı Tedarikçiler"); //İstatistikler segmentine geçince db classındaki istatistiky metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

pnl\_islem.Hide();

pnl\_istatistik.Show();

pnl\_istatistik.Location = pnl\_islem.Location;

}

}

//

//Tedarikçi Ara Comboboxı SelectedIndexChanged

//

private void cmbx\_ara\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmbx\_ara.SelectedIndex == 0) //Comboboxın SelectedIndex 0 Yani 'Seçiniz' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_adres.Visible = true;

txt\_tel.Visible = true;

txt\_mail.Visible = true;

txt\_no.Visible = true;

txt\_ad.Visible = true;

btn\_ekle.Enabled = true;

btn\_guncelle.Enabled = true;

btn\_listele.Enabled = true;

btn\_sil.Enabled = true;

btn\_ara.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 1)//Comboboxın SelectedIndex 1 Yani 'No'ya Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_adres.Visible = false;

txt\_tel.Visible = false;

txt\_mail.Visible = false;

txt\_no.Visible = true;

txt\_ad.Visible = false;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 2)//Comboboxın SelectedIndex 2 Yani 'Ad'a Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_adres.Visible = false;

txt\_tel.Visible = false;

txt\_mail.Visible = false;

txt\_no.Visible = false;

txt\_ad.Visible = true;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

}

}

}

}

### 7.6 Mağaza Otomasyon Programının Kategori Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Data.SqlClient;

namespace melihkocak0385

{

public partial class frm\_kategori : Form

{

db d = new db();

frm f = new frm();

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-I6L92IT\SQLEXPRESS03;Initial Catalog=IlkOtomasyon;Integrated Security=True"); //database bağlantısı

public frm\_kategori()

{

InitializeComponent();

}

//

//Formun Loadı

//

private void frm\_kategori\_Load(object sender, EventArgs e)

{

this.tBL\_KATEGORITableAdapter.Fill(this.ilkOtomasyonDataSet.TBL\_KATEGORI); //Datagridviewi form yüklenirken doldurmak için

ToolTip aciklama = new ToolTip(); //Tooltip için nesne türettim

aciklama.SetToolTip(btn\_sifirla, "Aşağıda Yazılı Olan Bilgileri Siler(Herhangi Bir Veri Kaybı Yaşanmaz)"); //Textboxların üstündeki küçük buton için tooltip

error\_no.BlinkStyle = ErrorBlinkStyle.NeverBlink; //ErrorProviderın blink style ayarı

cmbx\_ara.SelectedIndex = 0; //Arama comboxın form yüklenin seçili gelmesi için.

}

//

//Kategori Ekleme Butonu

//

private void btn\_ekle\_Click(object sender, EventArgs e)

{

int b;

if (txt\_no.Text == null || txt\_no.Text == "")//Kayıt Eklerken Primary Alanının Boş Girilemeyeceği için textboxın boş girilmesini engelledim.

{

MessageBox.Show("Lütfen Gerekli Alanları Doldurunuz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez."); //ErrorProvdier İle uyarıyoruz.

}

else

{ b = Convert.ToInt32(txt\_no.Text);

if (b>=255)//veri tabanında Kategori No tinyint veri tipinde tutuluyor 255 den fazla girince hat veriyor engellemek için yaptım.

{

MessageBox.Show("Kategori No Alanı İçin 255 den Yukarı Bir Değere Girilemez.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İşlem Başarılı Mesajı

}

else

{

d.prkeykontrol("prkeykontrolkategori", txt\_no);//Primary key alanına aynı veri iki kere girilmediği için db classındaki prkeykontrol metodu ile alanı kontrol ediyorum

int a = Convert.ToInt32(d.y);//db classındaki primary key kontorlü için değişken oluşturmuştum onu int veri tipine çeviriyorum.

if (a == 1)//kayıt etkilendiyse yani kayıt varsa hata mesajı verecek.

{

MessageBox.Show("Lütfen No Alanını Daha Önce Kullanılmamış Bir Değer Giriniz.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

}

else

{

d.eklegunsil("insert into TBL\_KATEGORI values(@p1,@p2)", txt\_no, txt\_ad); //Eğer ilgili textbox boş değilse db classındaki eklegünsil metodu textboxlara yazılan değerleri içine gönderdiğim sql kodu ile işleyecek.

if (d.x != 0) //Db classındaki x değişkeni row affectedi tutuyor 0 dan farklı ise işlemleri gerçekleştiriyor

{

d.listele("select \* from TBL\_KATEGORI", dataGridView1); //db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

MessageBox.Show("Ekleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information);//İşlem Başarılı Mesajı

}

}

}

}

}

//

//Kategori Listeleme Butonu

//

private void btn\_listele\_Click(object sender, EventArgs e)

{

d.listele("select \* from TBL\_KATEGORI", dataGridView1); //db classındaki listele metodu içine gönderdiğim sql kodunu işleyip datagridviewe değer gösterecek

}

//

//Kategori Silme Butonu

//

private void btn\_sil\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Kategori Numarasına Göre Silinme İşlemi Yapılacaktır 'Kategori No' Alanını Boş Bırakmayınız!", "Hata İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez."); //Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("delete from TBL\_KATEGORI where K\_No=@p1", txt\_no); //db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod satırı

if (d.x == 0) //Olmayan bir kaydı silmeyi engelliyoruz.

{

MessageBox.Show("Olmayan Bir Kayıdı Silmeye Çalıştınız. Gerekli Alanları Kontrol Ediniz.", "Hata İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez"); //Errorprovdier ile uyarıyoruz.

}

else

{

d.listele("select \* from TBL\_KATEGORI", dataGridView1); //Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değeri gösterecektir.

f.sifirla(txt\_no, txt\_ad); //sildikten sonra textboxları sıfırlıyoruz.

MessageBox.Show("Silme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Textboxları Sıfırlama Butonu

//

private void btn\_sifirla\_Click(object sender, EventArgs e)

{

f.sifirla(txt\_no, txt\_ad); //frm classındaki 'sifirla' metodu parametre olarak verdiğimiz textboxları sıfırlar

}

//

// Kategori Güncelleme Butonu

//

private void btn\_guncelle\_Click(object sender, EventArgs e)

{

if (txt\_no.Text == null || txt\_no.Text == "") //Textboxın boş olup olmadığını kontrol ediyoruz.

{

MessageBox.Show("Kategori Numarasına Göre Güncelleme İşlemi Yapılacaktır 'Kategori No' Alanını Boş Bırakmayınız!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

error\_no.SetError(txt\_no, "Bu Alan Boş Geçilemez"); //Errorprovdier ile uyarıyoruz.

}

else

{

d.eklegunsil("update TBL\_KATEGORI set K\_Ad=@p2 where K\_No=@p1 ", txt\_no, txt\_ad); //db classındaki eklegünsil metodu ile içine gönderdiğimiz sql cümlesini ilgili textboxdan veriyi alarak çalıştıran kod

if (d.x == 0) //Olmayan bir kaydı güncellemeyi engelliyoruz.

MessageBox.Show("Olmayan Bir Kayıdı Güncellemeye Çalıştınız. Lütfen Dikkat Ediniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error); //İlgili Mesajı Gösteriyoruz.

else

{

d.listele("select \* from TBL\_KATEGORI", dataGridView1);//Hata yok ise db classındaki listele metodu içine gönderdiğim sql kodunu işleyip ilgili datagridviewe değer gösterecek

MessageBox.Show("Güncelleme İşlemi Başarılı Bir Şekilde Gerçekleştirildi.", "İşlem Başarılı", MessageBoxButtons.OK, MessageBoxIcon.Information); //İlgili Mesajı Gösteriyoruz.

}

}

}

//

//Kategori Arama Butonu

//

private void btn\_ara\_Click(object sender, EventArgs e)

{

if (cmbx\_ara.SelectedIndex == 1)//Comboboxda Birinci seçenek seçili ise No ya göre ara yapılacak

{

d.ara("select \* from TBL\_KATEGORI where K\_No = @p1", txt\_no, dataGridView1);//db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak No'ya göre arama işlemini gerçekleştiriyor.

}

else if (cmbx\_ara.SelectedIndex == 2)//Comboboxda ikinci seçenek seçili ise Ada ya göre ara yapılacak

{

d.ara("select \* from TBL\_KATEGORI where RTRIM(K\_Ad) LIKE '%" + txt\_ad.Text + "%'", txt\_ad, dataGridView1); //db classındaki ara metodu içine gönderdiğimiz sql cümlesiyle gerekli textboxdan veriyi alarak Ada göre arama işlemini gerçekleştiriyor.

}

}

//----------------- ToolStrip Öğelerinin Click Eventi-------------------------------------

private void tumunusecToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki verinin tümünü seçmek için ContextMenuStrip aracının 'tumunusec' öğesinin click eventi

{

f.tumunusec(this); //frm classındaki tumunusec metodu ile textboxlardaki verinin tümünü seçiyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kopyalaToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kopyalamak için ContextMenuStrip aracının 'kopyala' öğesinin click eventi

{

f.kopyala(this); //frm classındaki kopyala metodu ile textboxlardaki veriyi kopyalama işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void yapistirToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlara veriyi yapıştırmak için ContextMenuStrip aracının 'yapistir' öğesinin click eventi

{

f.yapistir(this); //frm classındaki yapistir metodu ile textboxlara panodaki veriyi yapıştırıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void kesToolStripMenuItem\_Click(object sender, EventArgs e) //Textboxlardaki veriyi kesmek için ContextMenuStrip aracının 'kes' öğesinin click eventi

{

f.kes(this); //frm classındaki kes metodu ile textboxlardaki veriyi kesme işlemi yapıyoruz. İçine bu anlamındaki 'this' sözcüğünü yani ilgili formu yolluyoruz.

}

private void cikisToolStripMenuItem\_Click(object sender, EventArgs e) //Formdan sağ tık yaparak çıkmak için ContextMenuStrip aracının 'cikis' öğesinin click eventi

{

this.Close(); //İlgili formu kapatıyoruz.

}

//-------------------------------------------------------------------------------------------------

//------------Error Provdier Silinmesi İçin Gerekli Event-----------

private void txt\_no\_TextChanged(object sender, EventArgs e)

{

if (txt\_no.Text!="") // ErrorProvdier aracını ilgili textbox boş değilse temizliyoruz.

error\_no.Clear();

}

//------------------------------------------------------------------

//--TextBoxlara Yanlış Veri Girişini Engellemek İçin Gerekli Eventler--

private void txt\_no\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

} //Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_ad\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) && !char.IsSeparator(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

//---------------------------------------------------------------------

//

//Ürün Ara Comboboxı SelectedIndexChanged

//

private void cmbx\_ara\_SelectedIndexChanged(object sender, EventArgs e) //Kulanıcın kafası karışmasını engellemek amaçlı yapılmıştır.

{

if (cmbx\_ara.SelectedIndex == 0) //Comboboxın SelectedIndex 0 Yani 'Seçiniz' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_no.Visible = true;

txt\_ad.Visible = true;

btn\_ekle.Enabled = true;

btn\_guncelle.Enabled = true;

btn\_listele.Enabled = true;

btn\_sil.Enabled = true;

btn\_ara.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 1) //Comboboxın SelectedIndex 1 Yani 'No'ya Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_no.Visible = true;

txt\_ad.Visible = false;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

}

else if (cmbx\_ara.SelectedIndex == 2) //Comboboxın SelectedIndex 2 Yani 'Ad'a Göre Ara' ise aşağıdaki işlemler gerçekleştirilecek

{

txt\_no.Visible = false;

txt\_ad.Visible = true;

btn\_ara.Enabled = true;

btn\_ekle.Enabled = false;

btn\_guncelle.Enabled = false;

btn\_listele.Enabled = false;

btn\_sil.Enabled = false;

}

}

//

//DataGridView CellClick(Satırlara Tıklayınca Verileri İlgili Textboxlara Taşımak İçin)

//

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

d.tasima(txt\_no, txt\_ad, dataGridView1, e); //db classındaki 'tasima' metodu datagridviewin hücrelerine tıklayınca değerleri ilgili textboxlara taşıma işlemini yapar.

}

//

//İşlemler Ve İstastistikler Arası Geçiş Segmenti

//

private void sgmt\_kategori\_IndexChanged(object sender, EventArgs e) //Burada bir trick kullanılmıştır ilk başta aktif olan işlemler panelidir eğer segment değiştirilirse işlemler paneli gizlenir istatistik paneli gösterilir.

{

if (sgmt\_kategori.SelectedIndex == 0) //XanderUI sınıfından segment aracının seçili değeri 0 ise işlemler panelini(ekleme, silme, güncelleme....) Gösteriyoruz.

{

pnl\_istatistik.Hide();

pnl\_islem.Show();

chart1.Series.Remove(chart1.Series["Kategori Sayısı"]); //Chart nesnesini istatistiklerde sıkıntı olmasın diye her seferinde siliyoruz.

}

else if (sgmt\_kategori.SelectedIndex == 1)//XanderUI sınıfından segment aracının seçili değeri 1 ise istatistikler panelini gösteriyoruz.

{

d.istatistiky("select count(\*) from TBL\_KATEGORI", chart1, "Kategori Sayısı"); //İstatistikler segmentine geçince db classındaki istatistiky metodu ile içine gönderdiğim sql cümlesi çalışıp ilgili chart aracına değeri gösteriyor.

pnl\_islem.Hide();

pnl\_istatistik.Show();

pnl\_istatistik.Location = pnl\_islem.Location;

}

}

}

}

### 7.7 Mağaza Otomasyon Programının Rapor Formunun Proje Kodları

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WindowsFormsApp3.Formlar

{

public partial class frm\_rapor : Form

{

public frm\_rapor()

{

InitializeComponent();

}

melihkocak0385.db d = new melihkocak0385.db();

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-I6L92IT\SQLEXPRESS03;Initial Catalog=IlkOtomasyon;Integrated Security=True"); // Veri Tabanı Bağlantısı

//

//Formun Loadı

//

private void frm\_rapor\_Load(object sender, EventArgs e)

{

dateTimePicker1.Format = DateTimePickerFormat.Custom;//veritabanına rapordan dolayı sorgu göndermek için datetimepickerın formatı değiştirildi

dateTimePicker2.Format = DateTimePickerFormat.Custom;

dateTimePicker1.CustomFormat = "yyyy.MM.dd";//DateTimePicker e gerekli değer verildi

dateTimePicker2.CustomFormat = "yyyy.MM.dd";

txt\_min.Enabled = false;//Ürün TabPage Sekmesinde fiyat aralığı girmeyi zorunlu tutmadığım için, ilk başta enabled özelliğini false yaptım

txt\_max.Enabled = false;

ToolTip aciklama = new ToolTip();//tooltip için nesne türettim

aciklama.SetToolTip(chbox\_uonay, "İşaretlendiği Zaman Fiyat Aralığıda Girebilirsiniz."); //Ürün alanındaki checkbox için açıklama

aciklama.SetToolTip(chbox\_sonay, "İşaretlendiği Zaman Müşteri Numarasıda Girebilirsiniz.");//Sipariş alanındaki checkbox için açıklama

aciklama.SetToolTip(grpbox\_mbilgi, "İki Değer Arasında Olan Müşteri Bilgilerini Getirir.");

aciklama.SetToolTip(txt\_mmin, "Küçük Değer");

aciklama.SetToolTip(txt\_mmax, "Büyük Değer");

aciklama.SetToolTip(txt\_kmin, "Küçük Değer");//Fiyat yada no alanlarını girerken aralık vermek için textbox kullanıldı hangi textboxın küçük büyük değeri ifade ettiği anlaşılsın diye tooltip eklendi

aciklama.SetToolTip(txt\_kmax, "Büyük Değer");

aciklama.SetToolTip(txt\_tmin, "Küçük Değer");

aciklama.SetToolTip(txt\_tmax, "Büyük Değer");

aciklama.SetToolTip(txt\_min, "Küçük Değer");

aciklama.SetToolTip(txt\_max, "Büyük Değer");

cmbx\_uara.SelectedIndex = 0;

cmbx\_mara.SelectedIndex = 0;//ComboBoxlarda ilk değer seçili gelmesi için böyle yapıldı.

cmbx\_tara.SelectedIndex = 0;

cmbx\_kara.SelectedIndex = 0;

}

//

//Sipariş Rapor Butonu

//

private void btn\_raporSiparis\_Click(object sender, EventArgs e)

{

Rapor.siparis s = new Rapor.siparis();//'siparis' Sınıfından(CrystalReportta oluşturduğum sınıf) nesne türettim

if (chbox\_sonay.Checked==true && txt\_mno.Text=="")//Sipariş TabPage Sekmesinde Müşteri No girmek için CheckBox işaretlendiyse fakat girilmeyip 'Raporla' butonuna basılırsa hata verir.

MessageBox.Show("Lütfen Müşteri Numarasını Giriniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (chbox\_sonay.Checked==true && txt\_mno.Text != "")//Sipariş TabPage Sekmesinde Müşteri No girmek için CheckBox işaretlendip değer girildiyse ardından 'Raporla' butonuna basılırsa müşterinin belirlenen tarihlerdeki siparişleri rapora sunulur.

d.rapor("select S\_Tarih, b.M\_No, U\_Ad, c.U\_No, Adet, BFiyat, S\_ToFiyat, M\_Ad, M\_Soyad, M\_Tel from TBL\_SIPARIS a, TBL\_MUSTERİ b, Siparis\_Urun\_Detay c, TBL\_URUN d where a.S\_No=c.S\_No and b.M\_No = " + txt\_mno.Text + "and b.M\_No=a.M\_No and c.U\_No = d.U\_no and a.S\_Tarih BETWEEN'" + dateTimePicker1.Text + "' AND '" + dateTimePicker2.Text + "'", s, crystalReportViewer1);

else if(chbox\_sonay.Checked==false)//Sipariş TabPage Sekmesinde Müşteri No girmek için olan CheckBox işaretlenmediyse bu sefer sadece belirlediğimiz tarihler arasındaki müşteri numarası farketmeksizin bütün siparişleri getirir.

d.rapor("select S\_Tarih, b.M\_No, U\_Ad, c.U\_No, Adet, BFiyat, S\_ToFiyat, M\_Ad, M\_Soyad, M\_Tel from TBL\_SIPARIS a, TBL\_MUSTERİ b, Siparis\_Urun\_Detay c, TBL\_URUN d where a.S\_No=c.S\_No and b.M\_No=a.M\_No and c.U\_No = d.U\_no and a.S\_Tarih BETWEEN'" + dateTimePicker1.Text + " ' AND ' " + dateTimePicker2.Text + "'", s, crystalReportViewer1);

}

//

//Ürün Rapor Butonu

//

private void btn\_raporUrun\_Click(object sender, EventArgs e)

{

Rapor.urun u = new Rapor.urun();//'urun' Sınıfından(CrystalReportta oluşturduğum sınıf) nesne türettim

if (cmbx\_uara.SelectedIndex==0 && chbox\_uonay.Checked==false)//Ürün TabPage Sekmesinde ComboBoxda herhangi bir filtreleme işlemi yapılmadıysa ve Kullanıcın fiyat aralığı girebilecek olan CheckBox aktif hale getirilmediyse hata mesajı verir.

MessageBox.Show("Lütfen Filtreleme Yapınız Veya Fiyat Aralığı Giriniz.", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_uara.SelectedIndex == 0 && chbox\_uonay.Checked == true && txt\_min.Text == "" && txt\_max.Text == "")//Ürün TabPage Sekmesinde ComboBoxda herhangi bir filtreleme işlemi yapılmadıysa ama kullanıcın fiyat aralığı girebileceği CheckBox aktif hale getirilip fiyat girilmediyse hata mesajı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_uara.SelectedIndex == 0 && chbox\_uonay.Checked==true && txt\_min.Text != "" && txt\_max.Text != "")//Ürün TabPage Sekmesinde ComboBoxda herhangi bir filtreleme işlemi yapılmadıysa ama kullanıcın fiyat aralığı girebileceği CheckBox aktif hale getirildiyse ve İlgili textboxlarda fiyat değerleri yazıyorsa o fiyat aralığındaki ürünler rapora işlenir.

d.rapor("Select U\_No, U\_Ad, U\_Alis, U\_Satis, K\_Ad, T\_Ad from TBL\_URUN u, TBL\_KATEGORI k, TBL\_TEDARIKCILER t WHERE u.K\_No = k.K\_No AND U\_Satis BETWEEN " + txt\_min.Text + " AND " + txt\_max.Text + " AND t.T\_No=u.T\_No", u, crystalReportViewer1);

else if (cmbx\_uara.SelectedIndex == 1 && chbox\_uonay.Checked == true && txt\_min.Text == "" && txt\_max.Text == "")

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_uara.SelectedIndex==1 && chbox\_uonay.Checked == true && txt\_min.Text != "" && txt\_max.Text != "" && txt\_uno.Text!="")//Ürün TabPage Sekmesinde ComboBox aracının SelectedIndex değeri 1 olduysa yani 'Tedarikçi No'ya Göre Raporla' kısmı seçili ise ve Ürünlerin fiyatlarını girebileceğimiz yeri aktif eden CheckBox İşaretlendiyse 'txt\_min', 'txt\_max', 'txt\_uno' alanları boş değilse

//Bizim belirlediğimiz tedarikçinin yine bizim belirlediğimiz fiyat aralığındaki ürünleri listeleyecektir. sanırım bu yorum çok uzun oldu:)

d.rapor("Select U\_No, U\_Ad, U\_Alis, U\_Satis, K\_Ad, T\_Ad from TBL\_URUN u, TBL\_KATEGORI k, TBL\_TEDARIKCILER t WHERE u.K\_No = k.K\_No AND U\_Satis BETWEEN " + txt\_min.Text + " AND " + txt\_max.Text + " AND t.T\_No = " + txt\_uno.Text + " AND t.T\_No=u.T\_No", u, crystalReportViewer1);

else if (cmbx\_uara.SelectedIndex == 1 && chbox\_uonay.Checked == false && txt\_uno.Text == "")//Ürün TabPage Sekmesinde filtreleme ComboBoxın birinci değeri seçili ise ve fiyat aralığın belirttiğimiz yeri aktif eden CheckBox işaretlenmediyse ve 'txt\_uno' kısmı boş ise hata mesajı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_uara.SelectedIndex == 1 && chbox\_uonay.Checked == false && txt\_uno.Text !="")//Ürün TabPage Sekmesinde filtreleme ComboBoxın birinci değeri seçili ise ve fiyat aralığın belirttiğimiz yeri aktif eden CheckBox işaretlendiyse ve 'txt\_uno' boş değilse Bizim belirlediğimiz tedarikçinin bütün ürünleri rapora işlenir.

d.rapor("Select U\_No, U\_Ad, U\_Alis, U\_Satis, K\_Ad, T\_Ad from TBL\_URUN u, TBL\_KATEGORI k, TBL\_TEDARIKCILER t WHERE u.T\_No = " + txt\_uno.Text + " AND u.K\_No = k.K\_No and u.T\_No=t.T\_No", u, crystalReportViewer1);

else if (cmbx\_uara.SelectedIndex == 2 && chbox\_uonay.Checked == true && txt\_min.Text == "" && txt\_max.Text == "" && txt\_uno.Text == "")//Ürün TabPage Sekmesinde filtreleme ComboBoxın İkinci değeri seçili ise ve fiyat aralığın belirttiğimiz yeri aktif eden CheckBox işaretlendiyse ama 'txt\_min', 'txt\_max', 'txt\_uno' alanları boş ise hata mesahı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_uara.SelectedIndex == 2 && chbox\_uonay.Checked == true && txt\_min.Text != "" && txt\_max.Text != "" && txt\_uno.Text != "")//Ürün TabPage Sekmesinde ComboBox aracının SelectedIndex değeri 2 olduysa yani 'Kategori No'ya Göre Raporla' kısmı seçili ise ve Ürünlerin fiyatlarını girebileceğimiz yeri aktif eden CheckBox İşaretlendiyse 'txt\_min', 'txt\_max', 'txt\_uno' alanları boş değilse

//Bizim belirlediğimiz kategorinin yine bizim belirlediğimiz fiyat aralığındaki ürünleri listeleyecektir. sanırım bu yorumda çok uzun oldu:)

d.rapor("Select U\_No, U\_Ad, U\_Alis, U\_Satis, K\_Ad, T\_Ad from TBL\_URUN u, TBL\_KATEGORI k, TBL\_TEDARIKCILER t WHERE k.K\_No =" + txt\_uno.Text + " AND U\_Satis BETWEEN " + txt\_min.Text + " AND " + txt\_max.Text + " AND t.t\_No=u.t\_No AND k.K\_No = u.K\_No", u, crystalReportViewer1);

else if (cmbx\_uara.SelectedIndex == 2 && chbox\_uonay.Checked == false && txt\_uno.Text == "")//Ürün TabPage Sekmesinde filtreleme ComboBoxın ikinci değeri seçili ise ve fiyat aralığın belirttiğimiz yeri aktif eden CheckBox işaretlenmediyse ve 'txt\_uno' kısmı boş ise hata mesajı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_uara.SelectedIndex == 2 && chbox\_uonay.Checked == false && txt\_uno.Text != "")//Ürün TabPage Sekmesinde filtreleme ComboBoxın ikinci değeri seçili ise ve fiyat aralığın belirttiğimiz yeri aktif eden CheckBox işaretlendiyse ve 'txt\_uno' boş değilse Bizim belirlediğimiz kategorinin bütün ürünleri rapora işlenir.

d.rapor("Select U\_No, U\_Ad, U\_Alis, U\_Satis, K\_Ad, T\_Ad from TBL\_URUN u, TBL\_KATEGORI k, TBL\_TEDARIKCILER t WHERE t.T\_No = u.T\_No AND k.K\_No =" + txt\_uno.Text + " AND k.K\_No=u.K\_No", u, crystalReportViewer1);

}

//

//Müşteri Rapor Butonu

//

private void btn\_musteriRapor\_Click(object sender, EventArgs e)

{

Rapor.musteri m = new Rapor.musteri();//'musteri' Sınıfından(CrystalReportta oluşturduğum sınıf) nesne türettim

if (cmbx\_mara.SelectedIndex == 0)//Herhangi bir filtremele işlemi seçilmediyse hata mesajı verir.

MessageBox.Show("Lütfen Filtrede Bir Seçenek Seçiniz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_mara.SelectedIndex == 1 && txt\_mmin.Text == "" && txt\_mmax.Text == "")//Müşteri TabPage sekmesinin filtreleme ComboBox aracının birinci değeri seçili ise yani 'Müşteri No'ya Göre Raporla' kısmı seçili ise ama 'Müşteri No' aralığı vereceğimiz alanlar boş ise hata mesajı verir

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_mara.SelectedIndex == 1 && txt\_mmin.Text != "" && txt\_mmax.Text != "")//Müşteri TabPage sekmesinin filtreleme ComboBox aracının birinci değeri seçili ise yani 'Müşteri No'ya Göre Raporla' kısmı seçili ve 'Müşteri No' aralığını vereceğimiz alanlar boş değilse bizim belirlediğimiz alanlardaki bütün müşteriler rapora işlenir.

d.rapor("select M\_No, M\_Ad, M\_Soyad, M\_Tel, M\_Mail from TBL\_MUSTERİ where M\_No BETWEEN " + txt\_mmin.Text + " AND " + txt\_mmax.Text, m, crystalReportViewer1);

else if (cmbx\_mara.SelectedIndex == 2 && txt\_mad.Text == "")//Müşteri TabPage sekmesinin filtreleme ComboBox aracının ikinci değeri seçili ise yani 'Müşteri Ad'a Göre Raporla' kısmı seçili ise ama müşteri adı vereceğimiz TextBox boş ise hata mesajı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_mara.SelectedIndex == 2 && txt\_mad.Text != "")//Müşteri TabPage sekmesinin filtreleme ComboBox aracının ikinci değeri seçili ise yani 'Müşteri Ad'a Göre Raporla' kısmı seçili ise ama müşteri adı vereceğimiz TextBox boş değilse bizim belirlediğimiz ada uyan Müşteriler rapora işlenir.

d.rapor("select M\_No, M\_Ad, M\_Soyad, M\_Tel, M\_Mail from TBL\_MUSTERİ where M\_Ad Like '%" + txt\_mad.Text + "%'", m, crystalReportViewer1);

}

//

//Tedarikçi Rapor Butonu

//

private void btn\_tedarikRapor\_Click(object sender, EventArgs e)

{

Rapor.tedarikci t = new Rapor.tedarikci();//'tedarikci' Sınıfından(CrystalReportta oluşturduğum sınıf) nesne türettim

if (cmbx\_tara.SelectedIndex == 0)//Herhangi bir filtremele işlemi seçilmediyse hata mesajı verir.

MessageBox.Show("Lütfen Filtrede Bir Seçenek Seçiniz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if(cmbx\_tara.SelectedIndex == 1 && txt\_tmin.Text == "" && txt\_tmax.Text == "")//Tedarikçi TabPage sekmesinin filtreleme ComboBox aracının birinci değeri seçili ise yani 'Tedarikçi No'ya Göre Raporla' kısmı seçili ise ama 'Tedarikçi No' aralığı vereceğimiz alanlar boş ise hata mesajı verir

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_tara.SelectedIndex == 1 && txt\_tmin.Text != "" && txt\_tmax.Text != "")//Tedarikçi TabPage sekmesinin filtreleme ComboBox aracının birinci değeri seçili ise yani 'Tedarikçi No'ya Göre Raporla' kısmı seçili ve 'Tedarikçi No' aralığını vereceğimiz alanlar boş değilse bizim belirlediğimiz alanlardaki bütün Tedarikçiler rapora işlenir.

d.rapor("select T\_No, T\_Ad, T\_Adres, T\_Tel, T\_Mail from TBL\_TEDARIKCILER where T\_No BETWEEN " + txt\_tmin.Text + " AND " + txt\_tmax.Text, t, crystalReportViewer1);

else if (cmbx\_tara.SelectedIndex == 2 && txt\_tad.Text == "")//Tedarikçi TabPage sekmesinin filtreleme ComboBox aracının ikinci değeri seçili ise yani 'Tedarikçi Ad'a Göre Raporla' kısmı seçili ise ama Tedarikçi adı vereceğimiz TextBox boş ise hata mesajı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);//İlgili Mesajı Gösteriyoruz.

else if (cmbx\_tara.SelectedIndex == 2 && txt\_tad.Text != "")//Tedarikçi TabPage sekmesinin filtreleme ComboBox aracının ikinci değeri seçili ise yani 'Tedarikçi Ad'a Göre Raporla' kısmı seçili ise ama Tedarikçi adı vereceğimiz TextBox boş değilse bizim belirlediğimiz ada uyan Tedarikçiler rapora işlenir.

d.rapor("select T\_No, T\_Ad, T\_Adres, T\_Tel, T\_Mail from TBL\_TEDARIKCILER where T\_Ad Like '%" + txt\_tad.Text + "%'", t, crystalReportViewer1);

}

//

//Kategori Rapor Butonu

//

private void btn\_kategoriRapor\_Click(object sender, EventArgs e)

{

Rapor.kategori k = new Rapor.kategori();//'kategori' Sınıfından(CrystalReportta oluşturduğum sınıf) nesne türettim

if (cmbx\_kara.SelectedIndex == 0)//Herhangi bir filtremele işlemi seçilmediyse hata mesajı verir.

MessageBox.Show("Lütfen Filtrede Bir Seçenek Seçiniz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);

else if (cmbx\_kara.SelectedIndex == 1 && txt\_kmin.Text == "" && txt\_kmax.Text == "")//Kategori TabPage sekmesinin filtreleme ComboBox aracının birinci değeri seçili ise yani 'Kategori No'ya Göre Raporla' kısmı seçili ise ama 'Kategori No' aralığı vereceğimiz alanlar boş ise hata mesajı verir

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);

else if (cmbx\_kara.SelectedIndex == 1 && txt\_kmin.Text != "" && txt\_kmax.Text != "")//Kategori TabPage sekmesinin filtreleme ComboBox aracının birinci değeri seçili ise yani 'Kategori No'ya Göre Raporla' kısmı seçili ve 'Kategori No' aralığını vereceğimiz alanlar boş değilse bizim belirlediğimiz alanlardaki bütün Kategori rapora işlenir.

d.rapor("select K\_No, K\_Ad from TBL\_KATEGORI where K\_No between " + txt\_kmin.Text + " AND " + txt\_kmax.Text, k, crystalReportViewer1);

else if (cmbx\_kara.SelectedIndex == 2 && txt\_kad.Text == "")//Kategori TabPage sekmesinin filtreleme ComboBox aracının ikinci değeri seçili ise yani 'Kategori Ad'a Göre Raporla' kısmı seçili ise ama Kategori adı vereceğimiz TextBox boş ise hata mesajı verir.

MessageBox.Show("Lütfen Gerekli Değerleri Doldurunuz", "Hatalı İşlem", MessageBoxButtons.OK, MessageBoxIcon.Error);

else if (cmbx\_kara.SelectedIndex == 2 && txt\_kad.Text != "")//Kategori TabPage sekmesinin filtreleme ComboBox aracının ikinci değeri seçili ise yani 'Kategori Ad'a Göre Raporla' kısmı seçili ise ama Kategori adı vereceğimiz TextBox boş değilse bizim belirlediğimiz ada uyan Kategori rapora işlenir.

d.rapor("select K\_No, K\_Ad from TBL\_KATEGORI where K\_Ad like '%" + txt\_kad.Text + "%'", k, crystalReportViewer1);

}

//--------------------Comboboxların SelectedIndexChanged Eventleri--------------------------------------

private void cmbx\_tara\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmbx\_tara.SelectedIndex == 0)//ComboBoxın SelectedIndex değeri 0 ise yani 'Seçiniz' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_tbilgi.Visible = false;

grpbox\_tebilgi.Visible = false;

}

else if (cmbx\_tara.SelectedIndex == 1)//ComboBoxın SelectedIndex değeri 1 ise yani 'Müşteri No'ya Göre Raporla' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_tbilgi.Visible = true;

grpbox\_tebilgi.Visible = false;

}

else if (cmbx\_tara.SelectedIndex == 2)//ComboBoxın SelectedIndex değeri 2 ise yani 'Müşteri Ad'a Göre Raporla' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_tbilgi.Visible = false;

grpbox\_tebilgi.Location = new Point(137, 3);

grpbox\_tebilgi.Visible = true;

}

}//Tedarikçi TabPage Sekmesinde filtreleme için kullandım.

private void cmbx\_mara\_SelectedIndexChanged(object sender, EventArgs e)//Müşteri TabPage Sekmesinde filtreleme için kullandım.

{

if (cmbx\_mara.SelectedIndex == 0)//ComboBoxın SelectedIndex değeri 0 ise yani 'Seçiniz' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_mad.Visible = false;

grpbox\_mbilgi.Visible = false;

}

else if (cmbx\_mara.SelectedIndex == 1)//ComboBoxın SelectedIndex değeri 2 ise yani 'Müşteri No'ya Göre Raporla' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_mbilgi.Visible = true;

grpbox\_mad.Visible = false;

}

else if (cmbx\_mara.SelectedIndex == 2)//ComboBoxın SelectedIndex değeri 2 ise yani 'Müşteri Ad'a Göre Raporla' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_mbilgi.Visible = false;

grpbox\_mad.Location = new Point(137, 3);

grpbox\_mad.Visible = true;

}

}

private void cmbx\_kara\_SelectedIndexChanged(object sender, EventArgs e)//Kategori TabPage Sekmesinde filtreleme için kullandım.

{

if (cmbx\_kara.SelectedIndex == 0)//ComboBoxın SelectedIndex değeri 0 ise yani 'Seçiniz' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_kbilgi.Visible = false;

grpbox\_kabilgi.Visible = false;

}

else if (cmbx\_kara.SelectedIndex == 1)//ComboBoxın SelectedIndex değeri 1 ise yani 'Kategori No'ya Göre Raporla' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_kbilgi.Visible = true;

grpbox\_kabilgi.Visible = false;

}

else if (cmbx\_kara.SelectedIndex == 2)//ComboBoxın SelectedIndex değeri 2 ise yani 'Kategori Ad'a Göre Raporla' Kısmında ise olaylar gerçekleşecektir.

{

grpbox\_kbilgi.Visible = false;

grpbox\_kabilgi.Location = new Point(137, 3);

grpbox\_kabilgi.Visible = true;

}

}

private void cmbx\_uara\_SelectedIndexChanged(object sender, EventArgs e)//Ürün TabPage Sekmesinde filtreleme için kullandım.

{

if (cmbx\_uara.SelectedIndex == 0)//ComboBoxın SelectedIndex değeri 0 ise yani 'Seçiniz' Kısmında ise olaylar gerçekleşecektir.

{

txt\_uno.Text = "";

txt\_uno.Enabled = false;

}

else if (cmbx\_uara.SelectedIndex == 1)//ComboBoxın SelectedIndex değeri 1 ise yani 'Tedarikçi' Kısmında ise olaylar gerçekleşecektir.

txt\_uno.Enabled = true;

else if (cmbx\_uara.SelectedIndex == 2)//ComboBoxın SelectedIndex değeri 2 ise yani 'Kategori' Kısmında ise olaylar gerçekleşecektir.

txt\_uno.Enabled = true;

}

//------------------------------------------------------------------------------------------------------

//------------------------Checkboxların CheckChanged Eventi---------------------------------------------

private void chck\_sonay\_CheckedChanged(object sender, EventArgs e)//Sipariş TabPage sekmesinde müşteri numarasına görede raporlama yapmak istenirse diye kullandım.

{

if (chbox\_sonay.Checked == true)

txt\_mno.Enabled = true;

else if (chbox\_sonay.Checked == false)

{

txt\_mno.Enabled = false;

txt\_mno.Text = "";

}

}

private void chbox\_uonay\_CheckedChanged(object sender, EventArgs e)//Ürün TabPage sekmesinde fiyat aralığını aktif etmek için kullandım.

{

if (chbox\_uonay.Checked == true)

{

txt\_min.Enabled = true;

txt\_max.Enabled = true;

}

else

{

txt\_min.Enabled = false;

txt\_max.Enabled = false;

}

}

//------------------------------------------------------------------------------------------------------

//--------TextBoxlara Yanlış Veri Girişini Engellemek İçin Gerekli Eventler-----------------------------

private void txt\_uno\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_min\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_max\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_mno\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_mmin\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_mmax\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_tmin\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_tmax\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_kmin\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_kmax\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_kad\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) && !char.IsSeparator(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_tad\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) && !char.IsSeparator(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

private void txt\_mad\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar) && !char.IsSeparator(e.KeyChar);

}//Veri Tabanındaki alanlara yanlış veri girişi sonucunda hata çıkmasını engellemek için bu kod yazıldı. Sadece Numerik ifadelere izin veriyor.

//------------------------------------------------------------------------------------------------------

}

}

### 7.8 Mağaza Otomasyon Programının frm Sınıfın Kodları

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.ComponentModel;

namespace melihkocak0385

{

class frm

{

public int[] dizi = new int[3]; // formların formborderstyle özelliği none olunca hareket ettirmek için değişken tanımlamak gerekiyordu bu da o değişkenler dizi halinde tutuyorum

//frm\_ana formunda mdi form kullanıyorum formu panel içinde getirmek için yazıldı bu metot

public void formgetir(Form getirilecekform, Panel gösterilecekpanel, Form kontrol)

{

frm\_ana c = new frm\_ana(); //frm\_ana formunda c adında bir nesne türettim

if (Application.OpenForms["kontrol"] == null || kontrol.IsDisposed) //Herhangi bir formun açık olup olmadığını kontrol ediyorum.

{

gösterilecekpanel.Controls.Clear(); //Her ihtimale karşı panelin içini temizliyorum.

getirilecekform.MdiParent = c; //Gösterilecek olan formun mdiparent özelliğine c nesnesini atıyorum.

gösterilecekpanel.Controls.Add(getirilecekform); //Getirilecek olan formu gösterilecek olan panele ekliyorum.

getirilecekform.Show(); //Getirilecek olan formu gösteriyorum.

}

else //Aksi halde ise hata mesajı vermesini istedtim

{

MessageBox.Show("Aktif Pencereyi Kapatmadan Başka Bir Pencere Veya Aynı Pencere Açılamaz!", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//Textboxlar için contextmenüstrip oluşturmuştüm öğelerinden birisi yapıştır idi onun için metot yazdım

public void yapistir(Form kontrol)

{

try //Yapıştırılacak herhangi bir öğe olmadığında hata veriyordu o yüzden try catch bloklarının içine yazdım.

{

Control ctrl = kontrol.ActiveControl; //Control sınıfından nesne türetip nesnenin içine sınıfın aktifliği kontrol edilip döndürülen değer yazılmıştır

TextBox tx = (TextBox)ctrl; //Textbox sınıfından nesne türettim onun içine textboxa çevirerek ctrl nesnesini attım.

if (ctrl != null)

if (ctrl is TextBox) //ctrl nesnesi is cümleciği ile textbox nesnesi olup olmadığı kontrol ediliyor

tx.Paste(); //Panodaki veriler textboxa yapıştırılıyor.

}

catch (Exception)

{

}

}

//Textboxlar için contextmenüstrip oluşturmuştüm öğelerinden birisi kopyala idi onun için metot yazdım

public void kopyala(Form kontrol)

{

try //Kopyalanacak herhangi bir öğe olmadığında hata veriyordu o yüzden try catch bloklarının içine yazdım.

{

Control ctrl = kontrol.ActiveControl; //Control sınıfından nesne türetip nesnenin içine sınıfın aktifliği kontrol edilip döndürülen değer yazılmıştır

TextBox tx = (TextBox)ctrl; //Textbox sınıfından nesne türettim onun içine textboxa çevirerek ctrl nesnesini attım.

if (ctrl != null)

if (ctrl is TextBox) //ctrl nesnesi is cümleciği ile textbox nesnesi olup olmadığı kontrol ediliyor

tx.Copy(); //Textboxtaki veriler panoya kopyalanıyor.

}

catch (Exception)

{

}

}

//Textboxlar için contextmenüstrip oluşturmuştüm öğelerinden birisi kes idi onun için metot yazdım

public void kes(Form kontrol)

{

try //Kesilecek herhangi bir öğe olmadığında hata veriyordu o yüzden try catch bloklarının içine yazdım.

{

Control ctrl = kontrol.ActiveControl; //Control sınıfından nesne türetip nesnenin içine sınıfın aktifliği kontrol edilip döndürülen değer yazılmıştır

TextBox tx = (TextBox)ctrl; //Textbox sınıfından nesne türettim onun içine textboxa çevirerek ctrl nesnesini attım.

if (ctrl != null)

if (ctrl is TextBox) //ctrl nesnesi is cümleciği ile textbox nesnesi olup olmadığı kontrol ediliyor

tx.Cut(); //Textboxtaki veriler panoya Kesme sureti ile alınıyor.

}

catch (Exception)

{

}

}

//Textboxlar için contextmenüstrip oluşturmuştüm öğelerinden birisi Tümünü Seç idi onun için metot yazdım

public void tumunusec(Form kontrol)

{

try //Tümünü Seçecek herhangi bir öğe olmadığında hata veriyordu o yüzden try catch bloklarının içine yazdım.

{

Control ctrl = kontrol.ActiveControl; //Control sınıfından nesne türetip nesnenin içine sınıfın aktifliği kontrol edilip döndürülen değer yazılmıştır

TextBox tx = (TextBox)ctrl; //Textbox sınıfından nesne türettim onun içine textboxa çevirerek ctrl nesnesini attım.

if (ctrl != null)

if (ctrl is TextBox) //ctrl nesnesi is cümleciği ile textbox nesnesi olup olmadığı kontrol ediliyor

tx.SelectAll(); //Textboxdaki veriler tümünü seç ile seçiliyor

}

catch (Exception)

{

}

}

//Textboxların üstündeki küçük buton sıfırlama butonudur. Metodun içine yollanan textboxları sıfırlıyor.

public void sifirla(TextBox birinci, TextBox ikinci, TextBox ucuncu, TextBox dorduncu)

{

birinci.Text = "";

ikinci.Text = "";

ucuncu.Text = "";

dorduncu.Text = "";

}

//Textboxların üstündeki küçük buton sıfırlama butonudur. Metodun içine yollanan textboxları sıfırlıyor.

public void sifirla(TextBox birinci, TextBox ikinci, TextBox ucuncu, TextBox dorduncu, TextBox besinci, TextBox altinci)

{

birinci.Text = "";

ikinci.Text = "";

ucuncu.Text = "";

dorduncu.Text = "";

besinci.Text = "";

altinci.Text = "";

}

//Textboxların üstündeki küçük buton sıfırlama butonudur. Metodun içine yollanan textboxları sıfırlıyor.

public void sifirla(TextBox birinci, TextBox ikinci, TextBox ucuncu, TextBox dorduncu, TextBox besinci)

{

birinci.Text = "";

ikinci.Text = "";

ucuncu.Text = "";

dorduncu.Text = "";

besinci.Text = "";

birinci.Enabled = true;

}

//Textboxların üstündeki küçük buton sıfırlama butonudur. Metodun içine yollanan textboxları sıfırlıyor.

public void sifirla(TextBox birinci, TextBox ikinci)

{

birinci.Text = "";

ikinci.Text = "";

}

}

}

### 7.9 Mağaza Otomasyon Programının db Sınıfının Kodları

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Data;

using System.Windows.Forms;

using System.Net.NetworkInformation;

using System.Windows.Forms.DataVisualization.Charting;

namespace melihkocak0385

{

public class db

{

public int x; // Kullanıcı olmayan Kayıtla İlgili İşlem Yaparsa Engellemek İçin Yöntem Kullandım Ondan Dolayı Global Alanda Değişken Tanımladım.

public string y; //Primary key kontrolü için global alan değişken tanımlamam gerekti.

SqlConnection baglanti = new SqlConnection(@"Data Source=DESKTOP-I6L92IT\SQLEXPRESS03;Initial Catalog=IlkOtomasyon;Integrated Security=True"); // Veri Tabanı Bağlantısı

//5 Adet kolonlu tabloları datagridview nesnesinden ilgili textboxlara taşımak için yazılan metotdur.

public void tasima(TextBox no, TextBox birinci, TextBox ikinci, TextBox ucuncu, TextBox dorduncu, DataGridView f, DataGridViewCellEventArgs e)

{

try //DataGridView'in Header Alanına Tıklayınca Doğal Olarak Hata Veriyordu Bu Sorun 'Try Catch' Blokları İle Çözülmüştür.

{

no.Text = f.Rows[e.RowIndex].Cells[0].Value.ToString(); //Dışarıdan girilen 'no' textbox nesnesine datagridviewin ilgili satırındaki 0. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

birinci.Text = f.Rows[e.RowIndex].Cells[1].Value.ToString();

ikinci.Text = f.Rows[e.RowIndex].Cells[2].Value.ToString();

ucuncu.Text = f.Rows[e.RowIndex].Cells[3].Value.ToString();

dorduncu.Text = f.Rows[e.RowIndex].Cells[4].Value.ToString();

}

catch (Exception)

{

}

}

//4 Adet kolonlu tabloları datagridview nesnesinden ilgili textboxlara taşımak için yazılan metotdur.

public void tasima(TextBox no, TextBox birinci, TextBox ikinci, TextBox ucuncu, DataGridView f, DataGridViewCellEventArgs e)

{

try //DataGridView'in Header Alanına Tıklayınca Doğal Olarak Hata Veriyordu Bu Sorun 'Try Catch' Blokları İle Çözülmüştür.

{

no.Text = f.Rows[e.RowIndex].Cells[0].Value.ToString(); //Dışarıdan girilen 'no' textbox nesnesine datagridviewin ilgili satırındaki 0. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

birinci.Text = f.Rows[e.RowIndex].Cells[1].Value.ToString();

ikinci.Text = f.Rows[e.RowIndex].Cells[2].Value.ToString();

ucuncu.Text = f.Rows[e.RowIndex].Cells[4].Value.ToString();

}

catch (Exception)

{

}

}

//7 Adet kolonlu tabloları datagridview nesnesinden ilgili textboxlara ve comboboxlara taşımak için yazılan metotdur.

public void tasima(TextBox no, TextBox birinci, TextBox ikinci, TextBox ucuncu, ComboBox dorduncu, ComboBox besinci, TextBox altinci, DataGridView f, DataGridViewCellEventArgs e)

{

try //DataGridView'in Header Alanına Tıklayınca Doğal Olarak Hata Veriyordu Bu Sorun 'Try Catch' Blokları İle Çözülmüştür.

{

no.Text = f.Rows[e.RowIndex].Cells[0].Value.ToString(); //Dışarıdan girilen 'no' textbox nesnesine datagridviewin ilgili satırındaki 0. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

birinci.Text = f.Rows[e.RowIndex].Cells[1].Value.ToString();

ikinci.Text = f.Rows[e.RowIndex].Cells[2].Value.ToString();

ucuncu.Text = f.Rows[e.RowIndex].Cells[3].Value.ToString();

dorduncu.Text = f.Rows[e.RowIndex].Cells[4].Value.ToString(); //Dışarıdan girilen 'birincii' combobox nesnesine datagridviewin ilgili satırındaki 4. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

besinci.Text = f.Rows[e.RowIndex].Cells[5].Value.ToString();

altinci.Text = f.Rows[e.RowIndex].Cells[6].Value.ToString();

}

catch (Exception)

{

}

}

//3 Adet kolonlu tabloları datagridview nesnesinden ilgili textboxlar a ve datetimepicker e taşımak için yazılan metotdur.

public void tasima(TextBox no, TextBox birinci, DateTimePicker date, DataGridView f, DataGridViewCellEventArgs e)

{

try

{

no.Text = f.Rows[e.RowIndex].Cells[0].Value.ToString(); //Dışarıdan girilen 'no' textbox nesnesine datagridviewin ilgili satırındaki 0. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

birinci.Text = f.Rows[e.RowIndex].Cells[2].Value.ToString();

date.Text = f.Rows[e.RowIndex].Cells[1].Value.ToString(); //Dışarıdan girilen 'date' DateTimePicker nesnesine datagridviewin ilgili satırındaki 0. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

}

catch (Exception)

{

}

}

//2 Adet kolonlu tabloları datagridview nesnesinden ilgili textboxlara taşımak için yazılan metotdur.

public void tasima(TextBox no, TextBox birinci, DataGridView f, DataGridViewCellEventArgs e)

{

try

{

no.Text = f.Rows[e.RowIndex].Cells[0].Value.ToString(); //Dışarıdan girilen 'no' textbox nesnesine datagridviewin ilgili satırındaki 0. alanda bulunan hücrenin değerini alır string formatına çevirerek aktarır.

birinci.Text = f.Rows[e.RowIndex].Cells[1].Value.ToString();

}

catch (Exception)

{

}

}

//2 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur.

public void eklegunsil(string sqlcumle,TextBox no,TextBox birinci)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

komut.Parameters.AddWithValue("@p2", birinci.Text);

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//3 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur.

public void eklegunsil(string sqlcumle, TextBox no, TextBox birinci, TextBox ikinci)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

komut.Parameters.AddWithValue("@p2", birinci.Text);

komut.Parameters.AddWithValue("@p3", ikinci.Text);

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//4 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur.

public void eklegunsil(string sqlcumle, TextBox no, TextBox birinci, TextBox ikinci, TextBox ucuncu)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

komut.Parameters.AddWithValue("@p2", birinci.Text);

komut.Parameters.AddWithValue("@p3", ikinci.Text);

komut.Parameters.AddWithValue("@p4", ucuncu.Text);

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//5 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur.

public void eklegunsil(string sqlcumle, TextBox no, TextBox birinci, TextBox ikinci, TextBox ucuncu, TextBox dorduncu)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım).

komut.Parameters.AddWithValue("@p2", birinci.Text);

komut.Parameters.AddWithValue("@p3", ikinci.Text);

komut.Parameters.AddWithValue("@p4", ucuncu.Text);

komut.Parameters.AddWithValue("@p5", dorduncu.Text);

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//6 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur.

public void eklegunsil(string sqlcumle, TextBox no, TextBox birinci, TextBox ikinci, TextBox ucuncu, ComboBox birincii, ComboBox ikincii, TextBox altinci)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

komut.Parameters.AddWithValue("@p2", birinci.Text);

komut.Parameters.AddWithValue("@p3", ikinci.Text);

komut.Parameters.AddWithValue("@p4", ucuncu.Text);

komut.Parameters.AddWithValue("@p5", birincii.SelectedValue);

komut.Parameters.AddWithValue("@p6", ikincii.SelectedValue);

komut.Parameters.AddWithValue("@p7", altinci.Text);

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//1 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur.

public void eklegunsil(string sqlcumle, TextBox no)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//3 Adet Kolonlu Tablolarda Kayıt Güncellemek, Kayıt Eklemek, Kayıt Silmek İçin Yazılan Metotdur(DateTimePicker Barındırır.)

public void eklegunsil(string sqlcumle, DateTimePicker tarih, TextBox birinci, TextBox ikinci, TextBox ucuncu)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", tarih.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

komut.Parameters.AddWithValue("@p2", birinci.Text);

komut.Parameters.AddWithValue("@p3", ikinci.Text);

komut.Parameters.AddWithValue("@p4", ucuncu.Text);

x = komut.ExecuteNonQuery(); //ExecuteNonQuery Komutu Gerekli Kodları Aynı SSMS deki Queryi Execute Etmeye Yarar.

baglanti.Close(); //Veri Tabanı Bağlantısı Kapatıyoruz.

}

//Gerekli Forumlarda Kayıt Arama İşlemi Gerçekleştirmek İçin Kullanılan Metotdur.

public void ara(string sqlcumle, TextBox no, DataGridView a)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

komut.Parameters.AddWithValue("@p1", no.Text); //SQL Cümlesinde Kullandığım Parametreleri İlgili Textbox Nesnelerinden Gelen Değerleri AddWithValue Metodu İle Yazdırdım.

SqlDataAdapter da = new SqlDataAdapter(komut); //SqlDataAdapter Sınıfından da Adında Nesne Türettim Ve İçine Komut Nesnesi Gönderdimki Komut Nesnemizi Uyarlasın.

DataTable dt = new DataTable(); //DataTable Sınıfından dt Adında Nesne Türettim.

da.Fill(dt); //DataAdapter Nesnesinden Türettiğim Nesneyi Fill Metodu İle İçine dt Nesnesini Göndererek Doldurdum.

a.DataSource = dt; //DataGridView Nesnesinin Veri Kaynağını dt Nesnesine Eşitledim.

komut.ExecuteNonQuery(); //Gerekli Komutları ExecuteNonQuery Komutu İle Çalıştırdım.

baglanti.Close(); //Veri Tabanı Bağlantısını Kapattım

}

//Bütün Formlarda Kayıtları Listelemek Amaçlı Yazılmış Metotdur.

public void listele(string sqlcumle,DataGridView datagrid)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açıyoruz //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

SqlDataAdapter da = new SqlDataAdapter(sqlcumle,baglanti); //SqlDataAdapter Sınıfından da Adında Nesne Türettim Ve İçine Komut Nesnesi Gönderdimki Komut Nesnemizi Uyarlasın.

DataTable dt = new DataTable(); //DataTable Sınıfından dt Adında Nesne Türettim.

da.Fill(dt); //DataAdapter Nesnesinden Türettiğim Nesneyi Fill Metodu İle İçine dt Nesnesini Göndererek Doldurdum.

datagrid.DataSource = dt; //DataGridView Nesnesinin Veri Kaynağını dt Nesnesine Eşitledim.

baglanti.Close(); //Veri Tabanı Bağlantısını Kapattım

}

//Formlarda Bulunan Çeşitli İstatistikleri Listelemek Amaçlı Yazılmış Metotdur.

public void istatistiky(string sqlcumle, Chart Chrt, string seriesic)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açtım.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

SqlDataReader dr = komut.ExecuteReader(); //Birden Fazla Komut Döneceği İçin komut.ExecuteReader diyorum ve datareaderdan oluşturduğum nesneye atıyorum.

Chrt.Series.Add(seriesic); //Chart Nesnesinin Series Alanına Bir Series Ekliyorum.

while (dr.Read()) //While Döngüsüne dr.read şartını koyarak döngüyü oluşturuyorum

{

Chrt.Series[seriesic].Points.AddY(dr[0]); //Chart Nesnesinin Seriesin dışarıdan eklenen seriesic kısmının Point Y Kısmına dr[0] değerini ekliyorum.

}

baglanti.Close(); //Veri Tabanının Bağlantısını Kapatıyorum.

}

//Formlarda Bulunan Çeşitli İstatistikleri Listelemek Amaçlı Yazılmış Metotdur.

public void istatistikxy(string sqlcumle, Chart Chrt, string seriesic)

{

baglanti.Open(); //Veri Tabanı Bağlantısını Açtım.

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'SQL CÜMLE'sini Ve Veri Tabanı Bağlantısını Ekledim (Bu Metodu Birden Çok Yerde Kullanıyorum O Yüzden 'SQL CÜMLE'sini Dışarıdan String Türünde Aldım).

Chrt.Series.Add(seriesic); //Chart Nesnesinin Series Alanına Bir Series Ekliyorum.

SqlDataReader dr = komut.ExecuteReader();

while (dr.Read()) //While Döngüsüne dr.read şartını koyarak döngüyü oluşturuyorum

{

Chrt.Series[seriesic].Points.AddXY(dr[0], dr[1]); //Chart Nesnesinin Seriesin dışarıdan eklenen seriesic kısmının Point Y Kısmına dr[0], Point X Kısmına dr[1] değerini ekliyorum.

}

baglanti.Close(); //Veri Tabanının Bağlantısını Kapatıyorum.

}

//Sipariş Formunda Bulunan Siparis tablosundaki herhangi bir kayıta tıklandığı zaman Siparis\_Urun\_Detay Tablosundaki Gerekli Kolonları Getiren Metotdur.

public void tasi(TextBox birinci, TextBox ikinci, TextBox ucuncu, TextBox dorduncu, TextBox besinci)

{

SqlCommand komut = new SqlCommand("tasima",baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'tasima' türünde string ifadeyi yolladım bu sayede 'tasima' StoredProcudure çalışacak Ve Veri Tabanı Bağlantısını Ekledim.

komut.CommandType = CommandType.StoredProcedure; //Command Nesnesinin Varsayılan Değeri Textdir Bende StoredProcedure Kullanmak İstediğim İçin CommandType ile Tipini Belirtiyorum.

komut.Parameters.Add("@deger", SqlDbType.Int); //Veritabanında kayıtlı '@deger' değişkenini int türünde olduğu için Add Metodu ile Parametreyi eklerken İnt türünde olduğunu belirterek ekledim.

komut.Parameters["@deger"].Direction = ParameterDirection.Output; //Varsayılan Olarak Direction Özelliği bu parametre için inputtur. Ben bu değişkeni veritabanında output olarak tanımladığım için buradada output olduğunu belirtiyorum

komut.Parameters.Add("@deger2", SqlDbType.Int);

komut.Parameters["@deger2"].Direction = ParameterDirection.Output;

komut.Parameters.Add("@deger4", SqlDbType.Int);

komut.Parameters["@deger4"].Direction = ParameterDirection.Output;

komut.Parameters.Add("@deger5", SqlDbType.Int);

komut.Parameters["@deger5"].Direction = ParameterDirection.Output;

komut.Parameters.Add("@deger3", SqlDbType.Int); //Veritabanında kayıtlı '@deger' değişkenini int türünde olduğu için Add Metodu ile Parametreyi İnt türünde olduğunu belirterek ekledim.

int a = Convert.ToInt32(birinci.Text); //'birinci' adlı textbox nesnesinin text değerini İnt Türüne Çevirerek 'a' Değişkenine atadım.

komut.Parameters["@deger3"].Value = a; //'a' değişkenine atadığım değeri veritabanında bulunan '@deger3' değişkenine atadım.

baglanti.Open();

komut.ExecuteNonQuery(); //'tasima' storedprocedurunu çalıştırdım.

ikinci.Text = komut.Parameters["@deger"].Value.ToString(); //'tasima' storedprocedur undeki '@deger' değişkeninin değerini stringe çevirerek ilgili textboxa aktardım.

ucuncu.Text = komut.Parameters["@deger2"].Value.ToString(); //'tasima' storedprocedur undeki '@deger2' değişkeninin değerini stringe çevirerek ilgili textboxa aktardım.

dorduncu.Text = komut.Parameters["@deger4"].Value.ToString();

besinci.Text = komut.Parameters["@deger5"].Value.ToString();

baglanti.Close();

}

//Sipariş Tablosunda birim fiyat ve adet değerleri girilince gelen değerleri çarpıp toplamfiyat kolonuna yazdıran metot

public void toplamfiyat(TextBox ikinci, TextBox ucuncu, TextBox dorduncu)

{

SqlCommand komut = new SqlCommand("toplamfiyat", baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine 'toplamfiyat' türünde string ifadeyi yolladım bu sayede 'toplamfiyat' StoredProcudure çalışacak Ve Veri Tabanı Bağlantısını Ekledim.

komut.CommandType = CommandType.StoredProcedure; //Command Nesnesinin Varsayılan Değeri Textdir Bende StoredProcedure Kullanmak İstediğim İçin CommandType ile Tipini Belirtiyorum.

komut.Parameters.Add("@tfiyat", SqlDbType.Int); //Veritabanında kayıtlı '@tfiyat' değişkenini int türünde olduğu için Add Metodu ile Parametreyi eklerken İnt türünde olduğunu belirterek ekledim.

komut.Parameters["@tfiyat"].Direction = ParameterDirection.Output; //Varsayılan Olarak Direction Özelliği bu parametre için inputtur. Ben bu değişkeni veritabanında output olarak tanımladığım için buradada output olduğunu belirtiyorum

komut.Parameters.Add("@bfiyat", SqlDbType.Int);

komut.Parameters.Add("@adet", SqlDbType.Int); //Veritabanında kayıtlı '@adet' değişkenini int türünde olduğu için Add Metodu ile Parametreyi İnt türünde olduğunu belirterek ekledim.

int a = Convert.ToInt32(ikinci.Text);

int b = Convert.ToInt32(ucuncu.Text);

komut.Parameters["@bfiyat"].Value = a;

komut.Parameters["@adet"].Value = b; //'a' değişkenine atadığım değeri veritabanında bulunan '@adet' değişkenine atadım.

baglanti.Open();

komut.ExecuteNonQuery(); //'toplamfiyat' storedprocedurunu çalıştırdım.

dorduncu.Text = komut.Parameters["@tfiyat"].Value.ToString(); //'toplamfiyat' storedprocedur undeki '@tfiyat' değişkeninin değerini stringe çevirerek ilgili textboxa aktardım.

baglanti.Close();

}

//Sipariş Formunda S\_No Alanını Otomatik Artma Şeklinde Yapmadığım İçin Kendim Otomatik Artmasını Sağladım.

public void sonproc(SqlCommand komut)

{

komut.Connection = baglanti; //veritabanı bağlantısını alması gerektiği yeri gösterdim.

komut.CommandText = "son1"; //komut.commandtex e son1 storedprocedure nü atadım

komut.CommandType = CommandType.StoredProcedure; //Command Nesnesinin Varsayılan Değeri Textdir Bende StoredProcedure Kullanmak İstediğim İçin CommandType ile Tipini Belirtiyorum.

komut.Parameters.Add("@deger", SqlDbType.Int); //Veritabanında kayıtlı '@deger' değişkenini int türünde olduğu için Add Metodu ile Parametreyi eklerken İnt türünde olduğunu belirterek ekledim.

komut.Parameters["@deger"].Direction = ParameterDirection.Output; //Varsayılan Olarak Direction Özelliği bu parametre için inputtur. Ben bu değişkeni veritabanında output olarak tanımladığım için buradada output olduğunu belirtiyorum

baglanti.Open();

komut.ExecuteNonQuery(); //'son1' storedprocedurunu çalıştırdım.

baglanti.Close();

}

//Rapor Formunda Ürün Raporlamak için oluşturduğum metot

public void rapor(string sqlcumle,WindowsFormsApp3.Rapor.urun u , CrystalDecisions.Windows.Forms.CrystalReportViewer crystalreport)

{

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //Rapor için kullanacağımız sql cümlesini dışarıdan alıyoruz.

SqlDataAdapter da = new SqlDataAdapter(komut); //komut nesnemizi uyarlaması için DataAdapter nesnesi oluşturyoruz.

DataTable dt = new DataTable(); //Sanal bir tablo oluşturuyoruz

dt.Clear(); //Metot birden çok kez kullanıldığı için tabloda veri olmaması için temizliyoruz.

da.Fill(dt); //Sanal Tablomuzu Uyarladığımız Veriyle Dolduruyoruz.

u.SetDataSource(dt); //Raporlamada kullandığım ürün sınıfının kaynağını dt olduğunu belirtiyorum.

crystalreport.ReportSource = u; //Rapor formunda kullandığım crystalreport aracının rapor kaynağını u nesnesi olduğunu belirtiyorum

}

//Rapor Formunda Sipariş Raporlamak için oluşturduğum metot

public void rapor(string sqlcumle, WindowsFormsApp3.Rapor.siparis s, CrystalDecisions.Windows.Forms.CrystalReportViewer crystalreport)

{

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //Rapor için kullanacağımız sql cümlesini dışarıdan alıyoruz.

SqlDataAdapter da = new SqlDataAdapter(komut); //komut nesnemizi uyarlaması için DataAdapter nesnesi oluşturyoruz.

DataTable dt = new DataTable(); //Sanal bir tablo oluşturuyoruz

dt.Clear(); //Metot birden çok kez kullanıldığı için tabloda veri olmaması için temizliyoruz.

da.Fill(dt); //Sanal Tablomuzu Uyarladığımız Veriyle Dolduruyoruz.

s.SetDataSource(dt); //Raporlamada kullandığım siparis sınıfının kaynağını dt olduğunu belirtiyorum.

crystalreport.ReportSource = s; //Rapor formunda kullandığım crystalreport aracının rapor kaynağını s nesnesi olduğunu belirtiyorum

}

//Rapor Formunda Müşteri Raporlamak için oluşturduğum metot

public void rapor(string sqlcumle, WindowsFormsApp3.Rapor.musteri m, CrystalDecisions.Windows.Forms.CrystalReportViewer crystalreport)

{

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //Rapor için kullanacağımız sql cümlesini dışarıdan alıyoruz.

SqlDataAdapter da = new SqlDataAdapter(komut); //komut nesnemizi uyarlaması için DataAdapter nesnesi oluşturyoruz.

DataTable dt = new DataTable(); //Sanal bir tablo oluşturuyoruz

dt.Clear(); //Metot birden çok kez kullanıldığı için tabloda veri olmaması için temizliyoruz.

da.Fill(dt); //Sanal Tablomuzu Uyarladığımız Veriyle Dolduruyoruz.

m.SetDataSource(dt); //Raporlamada kullandığım musteri sınıfının kaynağını dt olduğunu belirtiyorum.

crystalreport.ReportSource = m; //Rapor formunda kullandığım crystalreport aracının rapor kaynağını m nesnesi olduğunu belirtiyorum

}

//Rapor Formunda Tedarikçi Raporlamak için oluşturduğum metot

public void rapor(string sqlcumle, WindowsFormsApp3.Rapor.tedarikci t, CrystalDecisions.Windows.Forms.CrystalReportViewer crystalreport)

{

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //Rapor için kullanacağımız sql cümlesini dışarıdan alıyoruz.

SqlDataAdapter da = new SqlDataAdapter(komut); //komut nesnemizi uyarlaması için DataAdapter nesnesi oluşturyoruz.

DataTable dt = new DataTable(); //Sanal bir tablo oluşturuyoruz

dt.Clear(); //Metot birden çok kez kullanıldığı için tabloda veri olmaması için temizliyoruz.

da.Fill(dt); //Sanal Tablomuzu Uyarladığımız Veriyle Dolduruyoruz.

t.SetDataSource(dt); //Raporlamada kullandığım tedarikci sınıfının kaynağını dt olduğunu belirtiyorum.

crystalreport.ReportSource = t; //Rapor formunda kullandığım crystalreport aracının rapor kaynağını t nesnesi olduğunu belirtiyorum

}

//Rapor Formunda Kategori Raporlamak için oluşturduğum metot

public void rapor(string sqlcumle, WindowsFormsApp3.Rapor.kategori k, CrystalDecisions.Windows.Forms.CrystalReportViewer crystalreport)

{

SqlCommand komut = new SqlCommand(sqlcumle, baglanti); //Rapor için kullanacağımız sql cümlesini dışarıdan alıyoruz.

SqlDataAdapter da = new SqlDataAdapter(komut); //komut nesnemizi uyarlaması için DataAdapter nesnesi oluşturyoruz.

DataTable dt = new DataTable(); //Sanal bir tablo oluşturuyoruz

dt.Clear(); //Metot birden çok kez kullanıldığı için tabloda veri olmaması için temizliyoruz.

da.Fill(dt); //Sanal Tablomuzu Uyarladığımız Veriyle Dolduruyoruz.

k.SetDataSource(dt); //Raporlamada kullandığım kategori sınıfının kaynağını dt olduğunu belirtiyorum.

crystalreport.ReportSource = k; //Rapor formunda kullandığım crystalreport aracının rapor kaynağını k nesnesi olduğunu belirtiyorum

}

//primary key kontrolü için yazılmış bir metot

public void prkeykontrol(string procedure\_ismi ,TextBox deger)

{

SqlCommand komut = new SqlCommand(procedure\_ismi, baglanti); //SqlCommand Sınıfından Komut Adında Nesne Türettim Ve İçine parametre olarak aldığım string ifadeyi yolladım bu sayede StoredProcudure çalışacak Ve Veri Tabanı Bağlantısını Ekledim.

komut.CommandType = CommandType.StoredProcedure; //Command Nesnesinin Varsayılan Değeri Textdir Bende StoredProcedure Kullanmak İstediğim İçin CommandType ile Tipini Belirtiyorum.

komut.Parameters.Add("@kontrol", SqlDbType.Int); //Veritabanında kayıtlı '@kontrol' değişkenini int türünde olduğu için Add Metodu ile Parametreyi eklerken İnt türünde olduğunu belirterek ekledim.

komut.Parameters["@kontrol"].Direction = ParameterDirection.Output; //Varsayılan Olarak Direction Özelliği bu parametre için inputtur. Ben bu değişkeni veritabanında output olarak tanımladığım için buradada output olduğunu belirtiyorum

komut.Parameters.Add("@deger", SqlDbType.Int);

int a = Convert.ToInt32(deger.Text);

komut.Parameters["@deger"].Value = a;

baglanti.Open();

komut.ExecuteNonQuery(); //storedprocedurunu çalıştırdım.

y = komut.Parameters["@kontrol"].Value.ToString(); //'storedprocedur deki '@kontrol' değişkeninin değerini stringe çevirerek ilgili textboxa aktardım.

baglanti.Close();

}

}

}

### 7.10 Veri Tabanının Scripti

USE [IlkOtomasyon]

GO

/\*\*\*\*\*\* Object: UserDefinedFunction [dbo].[son] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

Create Function [dbo].[son](@sayi1 int)

Returns int

As

Begin

set @sayi1 = (select max(s\_no) from TBL\_SIPARIS)

Set @sayi1 = @sayi1+1

return @sayi1

End

GO

/\*\*\*\*\*\* Object: Table [dbo].[Siparis\_Urun\_Detay] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Siparis\_Urun\_Detay](

[ID] [int] IDENTITY(1,1) NOT NULL,

[S\_No] [int] NOT NULL,

[U\_No] [int] NOT NULL,

[Adet] [smallint] NOT NULL,

[BFiyat] [int] NOT NULL,

CONSTRAINT [PK\_Siparis\_Urun\_Detay] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TBL\_KATEGORI] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TBL\_KATEGORI](

[K\_No] [tinyint] NOT NULL,

[K\_Ad] [varchar](30) NOT NULL,

CONSTRAINT [PK\_TBL\_KATEGORİ] PRIMARY KEY CLUSTERED

(

[K\_No] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TBL\_MUSTERİ] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TBL\_MUSTERİ](

[M\_No] [int] NOT NULL,

[M\_Ad] [varchar](30) NOT NULL,

[M\_Soyad] [varchar](30) NOT NULL,

[M\_Tel] [char](15) NOT NULL,

[M\_Mail] [varchar](40) NULL,

CONSTRAINT [PK\_TBL\_MÜŞTERİ] PRIMARY KEY CLUSTERED

(

[M\_No] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TBL\_SIPARIS] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TBL\_SIPARIS](

[S\_No] [int] NOT NULL,

[S\_Tarih] [date] NOT NULL,

[M\_No] [int] NOT NULL,

[S\_ToFiyat] [int] NOT NULL,

CONSTRAINT [PK\_TBL\_FATURA] PRIMARY KEY CLUSTERED

(

[S\_No] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TBL\_TEDARIKCILER] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TBL\_TEDARIKCILER](

[T\_No] [tinyint] NOT NULL,

[T\_Ad] [varchar](30) NOT NULL,

[T\_Adres] [varchar](200) NOT NULL,

[T\_Tel] [char](15) NOT NULL,

[T\_Mail] [varchar](40) NOT NULL,

CONSTRAINT [PK\_TBL\_PERSONEL] PRIMARY KEY CLUSTERED

(

[T\_No] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[TBL\_URUN] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[TBL\_URUN](

[U\_No] [int] NOT NULL,

[U\_Ad] [varchar](25) NOT NULL,

[U\_Alis] [int] NOT NULL,

[U\_Satis] [int] NOT NULL,

[K\_No] [tinyint] NOT NULL,

[T\_No] [tinyint] NOT NULL,

[U\_Stok] [tinyint] NOT NULL,

CONSTRAINT [PK\_TBL\_ÜRÜNLER] PRIMARY KEY CLUSTERED

(

[U\_No] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Siparis\_Urun\_Detay] WITH CHECK ADD CONSTRAINT [FK\_Siparis\_Urun\_Detay\_TBL\_SIPARIS] FOREIGN KEY([S\_No])

REFERENCES [dbo].[TBL\_SIPARIS] ([S\_No])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Siparis\_Urun\_Detay] CHECK CONSTRAINT [FK\_Siparis\_Urun\_Detay\_TBL\_SIPARIS]

GO

ALTER TABLE [dbo].[Siparis\_Urun\_Detay] WITH CHECK ADD CONSTRAINT [FK\_Siparis\_Urun\_Detay\_TBL\_URUN] FOREIGN KEY([U\_No])

REFERENCES [dbo].[TBL\_URUN] ([U\_No])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Siparis\_Urun\_Detay] CHECK CONSTRAINT [FK\_Siparis\_Urun\_Detay\_TBL\_URUN]

GO

ALTER TABLE [dbo].[TBL\_SIPARIS] WITH CHECK ADD CONSTRAINT [FK\_TBL\_SIPARIS\_TBL\_MUSTERİ1] FOREIGN KEY([M\_No])

REFERENCES [dbo].[TBL\_MUSTERİ] ([M\_No])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[TBL\_SIPARIS] CHECK CONSTRAINT [FK\_TBL\_SIPARIS\_TBL\_MUSTERİ1]

GO

ALTER TABLE [dbo].[TBL\_URUN] WITH CHECK ADD CONSTRAINT [FK\_TBL\_URUN\_TBL\_FIRMA] FOREIGN KEY([T\_No])

REFERENCES [dbo].[TBL\_TEDARIKCILER] ([T\_No])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[TBL\_URUN] CHECK CONSTRAINT [FK\_TBL\_URUN\_TBL\_FIRMA]

GO

ALTER TABLE [dbo].[TBL\_URUN] WITH CHECK ADD CONSTRAINT [FK\_TBL\_URUN\_TBL\_KATEGORI] FOREIGN KEY([K\_No])

REFERENCES [dbo].[TBL\_KATEGORI] ([K\_No])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[TBL\_URUN] CHECK CONSTRAINT [FK\_TBL\_URUN\_TBL\_KATEGORI]

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[prkeykontrol] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[prkeykontrol] (@kontrol int output, @deger int)

AS

select \* from TBL\_TEDARIKCILER where T\_No = @deger

set @kontrol =@@ROWCOUNT

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[prkeykontrolkategori] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create proc [dbo].[prkeykontrolkategori] (@kontrol int output, @deger int)

AS

select \* from TBL\_KATEGORI where K\_No = @deger

set @kontrol =@@ROWCOUNT

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[prkeykontrolmusteri] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create proc [dbo].[prkeykontrolmusteri] (@kontrol int output, @deger int)

AS

select \* from TBL\_MUSTERİ where M\_No = @deger

set @kontrol =@@ROWCOUNT

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[prkeykontrolsiparis] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

create proc [dbo].[prkeykontrolsiparis] (@kontrol int output, @deger int)

AS

select \* from TBL\_SIPARIS where S\_No = @deger

set @kontrol =@@ROWCOUNT

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[prkeykontrolurun] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[prkeykontrolurun] (@kontrol int output, @deger int)

AS

select \* from TBL\_URUN where U\_No = @deger

set @kontrol =@@ROWCOUNT

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[son1] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[son1] (@deger int output)

as

BEGIN

SELECT @deger=MAX(S\_No) FROM TBL\_SIPARIS

if @deger = null

begin

set @deger=1

end

else

BEGIN

set @deger=@deger+1

END

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[tasima] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[tasima] (@deger int output, @deger2 int output, @deger4 int output, @deger5 int output, @deger3 int )

as

BEGIN

select @deger=(select U\_No from Siparis\_Urun\_Detay where S\_No=(select S\_No from TBL\_SIPARIS where S\_No = @deger3))

select @deger2=(select adet from Siparis\_Urun\_Detay where S\_No=(select S\_No from TBL\_SIPARIS where S\_No = @deger3))

select @deger4=(select M\_No from TBL\_SIPARIS where S\_No=(select S\_No from Siparis\_Urun\_Detay where S\_No = @deger3))

select @deger5=(select BFiyat from Siparis\_Urun\_Detay where S\_No=(select S\_No from TBL\_SIPARIS where S\_No = @deger3))

END

GO

/\*\*\*\*\*\* Object: StoredProcedure [dbo].[toplamfiyat] Script Date: 7.01.2021 20:59:24 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE proc [dbo].[toplamfiyat] (@tfiyat int output, @bfiyat int, @adet int)

as

set @tfiyat=@adet\*@bfiyat

GO