



## YÖNLENDİRİLMİŞ ÇALIŞMA VİZE PROJESİ

### DEPO YÖNETİM SİSTEMİ

( FAST WMS )



ÖZGE ALTINTAŞ

HR210037

2022/2023 BAHAR DÖNEMİ

## İçindekiler

Proje Hakkında .....	3
Veritabanı Diagramı .....	4
Veritabanı Tabloları.....	5
1- Dbo.KullaniciKayıt .....	5
2- Dbo. Grup.....	5
Ekran Detayları .....	6
E1-Kullanıcı Girişi.....	6
E1-2 Bağlantı Kodları .....	6
E1-2 Ekran Geçişleri Kodları .....	7
E1-3 Şifreyi Göster Kodları .....	7
E1-4 Rakam ve Sayı Kontrolü Kodları .....	8
E2-Şifremi Unuttum.....	8
Depo Ekranı .....	9
E4-1 Ürün Adresi Kodları .....	9
E4-2 Stok Takibi Kodları .....	11
E4-3 Sipariş Toplama Kodları .....	12
E4-4 Mal Kabul Kodları .....	14
BackOffice Ekranı .....	15
E5-1 Cari Kayıt Kodları.....	16
E5-2 Stok Kayıt Kodları.....	17
E5-3 Alım Faturası Kodları .....	20
E5-4 Satış Faturası Kodları.....	22
Yönetici Paneli Ekranı .....	23
E6-1 Kullanıcı Kayıt Kodları .....	23
E6-1- 1 Kontrol Ayarları Kodları.....	24

E6-1-2 Veri Tabanı Kayıt Kodları.....	25
E6-1-3 Şifre Güvenliği Kodları.....	26
E6-1-4 Gruplandırma Kodları .....	27
E6-2 Yetkiler Kodları.....	27
E6-3 Raflar Kodları.....	29
E6-4 Yardım Kodları.....	30

## Projenin Hakkında

Depo Yönetim Sistemleri (WMS) giriş, çıkış, takip, kontrol gibi işlemleri tek bir sistem üzerinden sürdürmek amacı ile kullanılır.

Bir ambarda gerçekleşebilecek tüm operasyonları etkin ve verimli şekilde yönetmek, operasyonların kâğıtsız, en kısa sürede, en az insan ve makine kaynağı ile hatasız tamamlanmasını sağlamak için kullanılır.

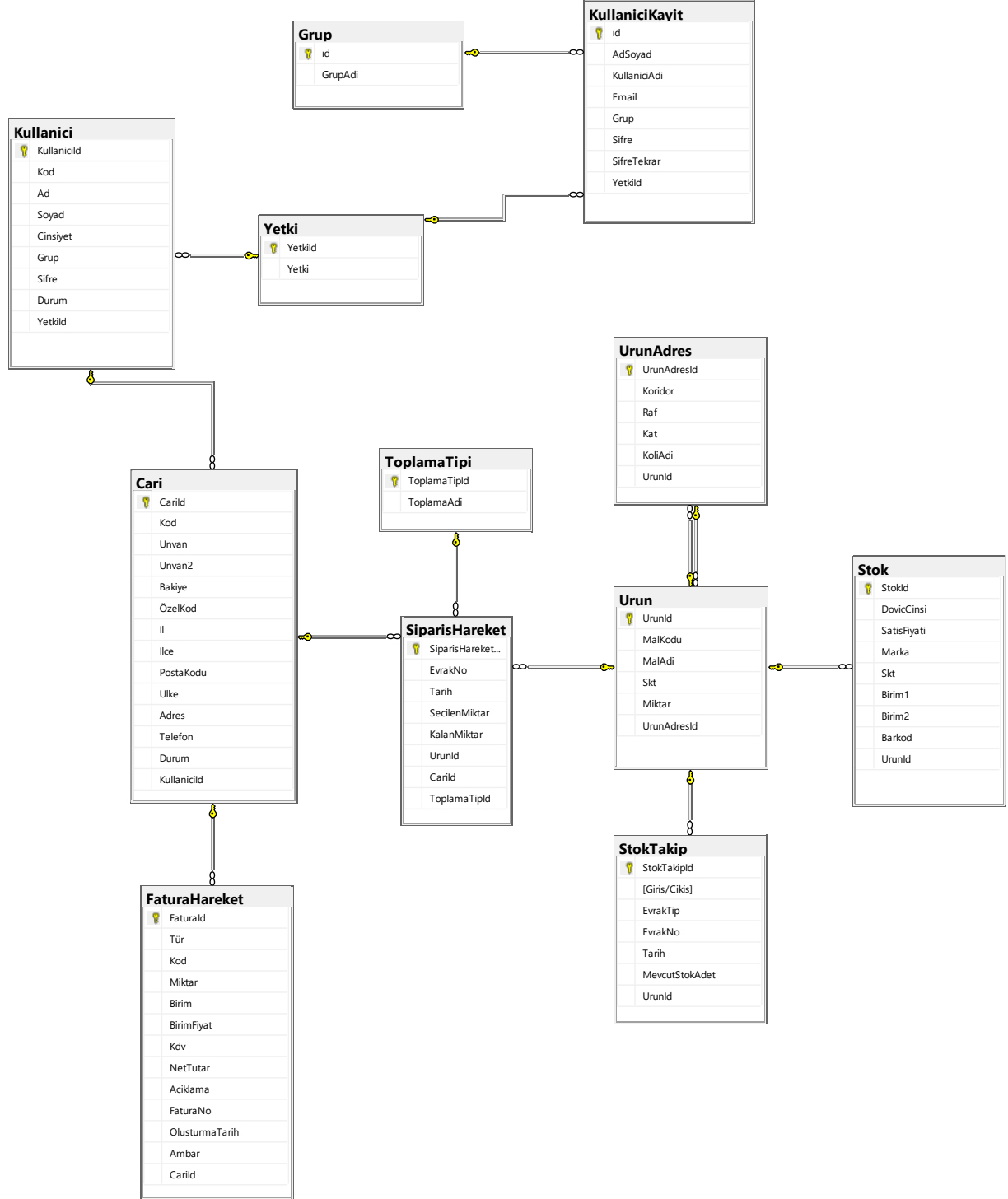
Depo sistemlerinin avantajları söz konusu olunca daha iyi depolama alanı, stok ve iş gücü yönetimi; israfı en aza indirmeye ve maliyetleri düşürmeye yardımcı oluyor.

WMS yazılımı sayesinde daha etkin ve verimli şekilde tüm kaynaklar ve süreçler yönetilebildiği için işletmelerin hızı artıyor.

İnsan gücünü ve hatasını azaltma, Sistemsel çalışmayı arttırma, Sipariş doğruluğunu artırmak ve depoda harcanan zamanı azaltmak için mal kabul süreçlerinde basit ve verimli süreç sunulmakta. Böylece, manuel işlemlerde karşılaşılabilecek olası insan hataları azalıyor.

Proje C# (Microsoft Visual Studio) ve Microsoft SQL Server Management Studio kullanılarak yapılmaktadır.


## Veritabanı Diagramı



## Veritabanı Tabloları

### Dbö.KullaniciKayit Tablosu:

	id	AdSoyad	KullaniciAdi	Email	Grup	Sifre	SifreTekrar
	1	Yasemin Kaya	ysmn	yaseminkaya@...	1	22078	22078
	6	Özge Altıntaş	ozgealtnts	ozgealtintas@h...	1	99457	99457
	7	Merve Çınar	cinarmerve	cinarmerve@h...	2	2564	2564
	8	Salih Yaman	salihyaman	yamansalih@o...	3	3587	3587

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	AdSoyad	nvarchar(50)	<input checked="" type="checkbox"/>
	KullaniciAdi	nvarchar(50)	<input checked="" type="checkbox"/>
	Email	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Grup	int	<input checked="" type="checkbox"/>
	Sifre	int	<input checked="" type="checkbox"/>
	SifreTekrar	int	<input checked="" type="checkbox"/>
	Yetkild	int	<input checked="" type="checkbox"/>


### Açıklama:

Kullanıcı giriş ve kayıt bilgilerinin tutulduğu tablodur, veriler bu tablo üzerinden çekilir.

Grup tablosu ile ilişkilendirilmiştir

### Dbö. Grup Tablosu:

	id	GrupAdi
	1	Depo ...
	2	BackOffice ...
	3	Yönetici ...

	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	GrupAdi	nchar(50)	<input checked="" type="checkbox"/>

### Açıklama:

Kullanıcının grup çeşitlerinin bulunduğu tablodur.

## Giriş Sayfası:

Tablo 1/Kullanıcı Girişi

## -KODLARI-

```
1 reference
private void chkGoster_CheckedChanged(object sender, EventArgs e)
{
    //Şifre gösterme ve gizleme özelliği verir
    txtSifre.UseSystemPasswordChar= chkGoster.CheckState == CheckState.Checked? true : false;
}

2 references
private void btnGiris_Click(object sender, EventArgs e)
{
    //kullanıcı adı veya şifrenin boş geçilmesini engeller
    if (String.IsNullOrEmpty(txtKullaniciAdi.Text) || string.IsNullOrEmpty(txtSifre.Text))
    {
        MessageBox.Show("Boş Geçilemez");
        return;
    }

    //sql bağlantısı kurulur
    con = new SqlConnection("Data Source =DESKTOP-F2SN8FC; Initial Catalog =WMS; Integrated Security =true");
    com = new SqlCommand();
    con.Open();//bağlantı açılır
    com.Connection = con;

    //KullaniciKayit tablosundan girilen veriler birbirini tutuyormu diye sorgu yapıyoruz
    com.CommandText = "Select * From KullaniciKayit where KullaniciAdi = " + txtKullaniciAdi.Text + " And Sifre= " + txtSifre.Text + ";";

    dr = com.ExecuteReader();
}
```

Tablo 2/Bağlantı

## Açıklama:

Veri tabanıyla bağlantı kurulmadan önce girilen değerlerin boş olup olmadığı kontrol edilir. daha sonra SQL ile bağlantı kurulur ve girilen kullanıcı adı ve şifre sorgu ile veri tabanından çekilir. Ve okunur.

```

if (dr.Read())
{
    int kullanıcıGrup = Convert.ToInt32(dr["Grup"]);
    //Eğer kullanıcının Grubu 1 ise Depo ekranını göster.
    if (kullanıcıGrup == 1)
    {
        Depo form = new Depo();
        form.Show();
    }
    //Eğer kullanıcının Grubu 2 ise BackOffice ekranını göster.
    else if (kullanıcıGrup == 2)
    {
        BackOffice form = new BackOffice();
        form.Show();
    }
    //Eğer kullanıcının Grubu 3 ise Yönetici Paneli ekranını göster.
    else if (kullanıcıGrup == 3)
    {
        YoneticiPanel form = new YoneticiPanel();
        form.Show();
    }
    //Şartların hiçbirisi değil ise mesaj ekranını yazdır
    else
    {
        MessageBox.Show("Kullanıcıya grup ataması yapılmamış.");
        return;
    }
    //Giriş sayfasını gizle
    this.Hide();
}
//Girilen bilgiler yanlışsa uyarı mesajı verir
else
{
    MessageBox.Show("Hatalı Kullanıcı Adı veya Şifre!");
}
con.Close();
//Bağlantı Kapatılır

```

Tablo 3/Ekran Geçişleri

Açıklama:

Else if ile girilen bilgilerin kontrolü sağlanır.

Veri tabanında kullanıcıya ait olan grup bulunarak kullanıcıyı kendi ekranına yönlendirir.

Eğer girilen bilgiler yanlış ise hata alınır ve bağlantı kapatılır.

```

1 reference
private void chkGoster_CheckedChanged(object sender, EventArgs e)
{
    //Şifre gösterme ve gizleme özelliği verir

    txtSifre.UseSystemPasswordChar= chkGoster.CheckState == CheckState.Checked? true : false;
}

```

Tablo 4/Şifreyi Göster

Açıklama:

Eğer şifre gizli ise şifreyi gösterme butonu true olur ve şifre görünür.

```

1 reference
private void txtKullaniciAdi_KeyPress(object sender, KeyPressEventArgs e)
{
    //Kullanıcı adına sayı girilmesini engeller
    e.Handled = !char.IsLetter(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);

    //Boşluk engeller
    if ((int)e.KeyChar == 32)
    {
        e.Handled = true;
    }
}

1 reference
private void txtSifre_KeyPress(object sender, KeyPressEventArgs e)
{
    //Şifreye harf girilmesi engeller
    e.Handled = !char.IsNumber(e.KeyChar) && !char.IsControl(e.KeyChar)
        && !char.IsSeparator(e.KeyChar);

    //Boşluk engeller
    if ((int)e.KeyChar == 32)
    {
        e.Handled = true;
    }

    //Enter tuşuna basılınca giriş butony aktif edilir
    else if (e.KeyChar == Convert.ToChar(Keys.Enter))
    {
        btnGiris_Click(null, null);
    }
}

```


Tablo 5/Rakam ve Sayı Kontrolü

Açıklama:

Şifre ve kullanıcı adına rakam , harf ve boşluk engelleme gibi özel durumlar için kodlar kullanılır.

Şifremi Unuttum Sayfası:

Şifremi Unuttum



**Giriş Yaparken Sorun mu Yaşıyorsunuz?**

E-posta adresini ve kullanıcı adını giriniz.

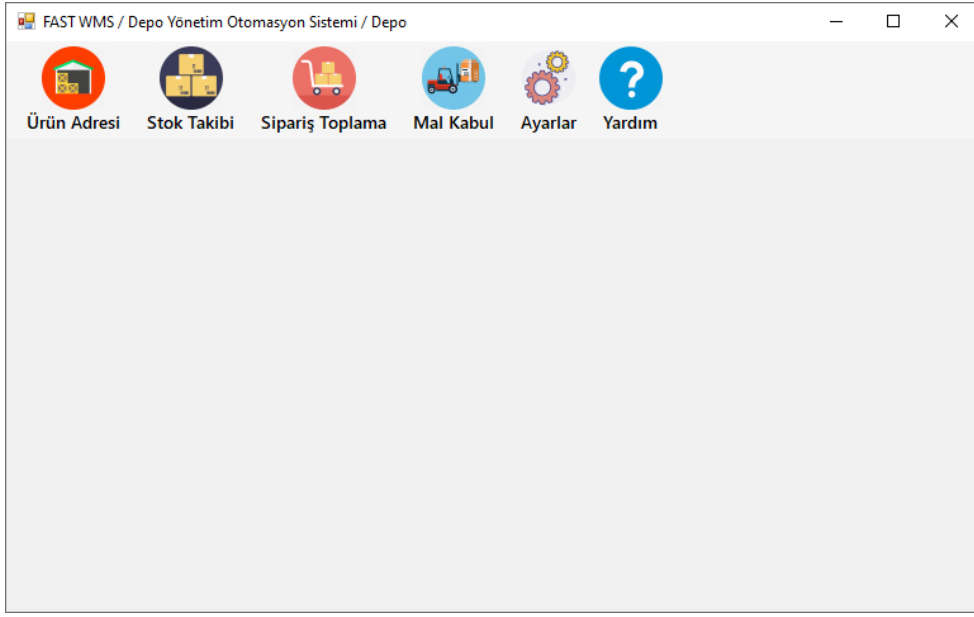
Kullanıcı Adı

E-Posta

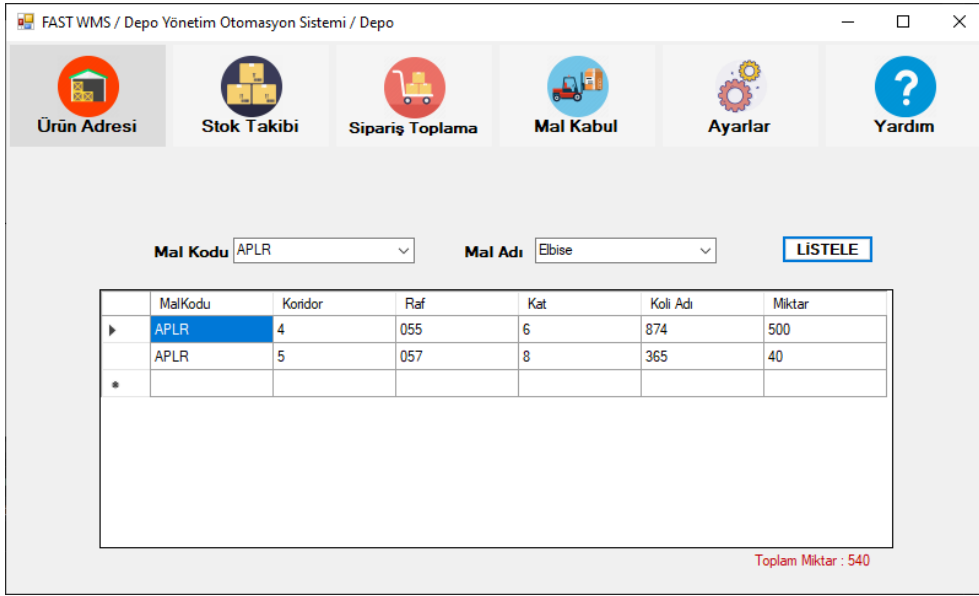
**Mail Gönder**



## Depo Ekranı:



## Ürün Adresi Sayfası:



## -KODLARI-

```
SqlConnection baglanti = new SqlConnection("Data Source =DESKTOP-F2SN8FC; Initial Catalog =WMS; Integrated Security =true");
SqlConnection connection = new SqlConnection("Data Source =DESKTOP-F2SN8FC; Initial Catalog =WMS; Integrated Security =TRUE");
SqlCommand cmd;
SqlDataReader dr;
SqlDataAdapter da;

1 reference
private void ToplamMiktar()
{
    //Seçilen 5.Sütundaki verilerin toplama hesabı
    double toplam = 0;
    for (int i = 0; i < dtListele.Rows.Count; i++)
    {
        toplam += Convert.ToDouble(dtListele.Rows[i].Cells[5].Value);
    }
    lblToplam.Text = "Toplam Miktar : " + toplam;
    //label'a miktar yazdırılır
}

1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    //Listeleme butonuna tıklandığında sorgu çalışır
    SqlCommand komut = cmd = new SqlCommand(string.Format(@"
SELECT
    Urun.MalKodu,UrunAdres.Koridor, UrunAdres.Raf, UrunAdres.Kat, UrunAdres.MaliAdi AS [Mali Adı], Urun.Miktar
FROM Urun
LEFT JOIN UrunAdres ON UrunAdres.UrunAdresId=Urun.UrunAdresId
WHERE MalKodu='{0}''", cmbxmlkodu.Text), connection);
    //Girilen mal koduna göre listeleme sorgusu
    SqlDataAdapter dradapter = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    dradapter.Fill(dt);
    dtListele.DataSource = dt;
    baglanti.Close();
    ToplamMiktar();
}
```

### Açıklama:

Sql bağlantısı kurulur toplam miktar adında metot oluşturulur. 5.sütundaki toplam miktar hesaplanılır ve lbltoplam içine yazılır.

Daha sonra listele butonuna basılır ve seçilen mal koduna göre bize kayıtlı olan verileri listeler.

```
1 reference
private void UrunAdresi_Load(object sender, EventArgs e)
{
    //ürün tablosundan veri çekip comboBox içerisine koyuyoruz.
    cmbxmlkodu.Items.Clear();
    cmd = new SqlCommand(string.Format(@"select * from Urun", cmbxmlkodu.Text ), connection);
    da = new SqlDataAdapter(cmd);
    DataTable tablo = new DataTable();
    da.Fill(tablo);
    cmbxmlkodu.DisplayMember = "MalKodu";
    cmbxmlkodu.ValueMember = "UrunId";
    cmbxmlkodu.DataSource = tablo;

    cmbxmladi.DisplayMember = "MaliAdi";
    cmbxmladi.ValueMember = "UrunId";
    cmbxmladi.DataSource = tablo;
}
```

### Açıklama:

Veritabanından ürünler çekilir ve ComboBox içine veriler yüklenir seçilen mal koduna ait mal adi otomatik olarak yazdırılır.

## Stok Takibi Sayfası:

FAST WMS / Depo Yönetim Otomasyon Sistemi / Depo

Ürün Adresi Stok Takibi Sipariş Toplama Mal Kabul Ayarlar Yardım

Mal Kodu APLR Mal Adı Elbise **LİSTELE**

	MalKodu	EvrakNo	Giris/Cikis	EvrakTip	Tarih	Miktar
▶	APLR	2547	G	GF	20.02.2022	500
	APLR	2547	G	GF	20.02.2022	40
*						

Toplam Miktar : 540

## -KODLARI-

```
public StokTakibi()
{
    InitializeComponent();
}

SqlConnection baglanti = new SqlConnection("Data Source =DESKTOP-F2SN8FC; Initial Catalog =WMS; Integrated Security =true");
SqlConnection connection = new SqlConnection("Data Source =DESKTOP-F2SN8FC; Initial Catalog =WMS; Integrated Security =TRUE");
SqlCommand cmd;
SqlDataReader dr;
SqlDataAdapter da;

1 reference
private void ToplamMiktar()
{
    double toplam = 0;
    for (int i = 0; i < dtListele.Rows.Count; i++)
    {
        toplam += Convert.ToDouble(dtListele.Rows[i].Cells[5].Value);
    }
    lblToplam.Text = "Toplam Miktar : " + toplam;
}

1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    SqlCommand komut = cmd = new SqlCommand(string.Format@"
SELECT
    Urun.MalKodu,StokTakip.EvrakNo,StokTakip.[Giris/Cikis],StokTakip.EvrakTip,StokTakip.Tarih, Urun.Miktar
FROM Urun
LEFT JOIN UrunAdres ON UrunAdres.UrunAdresId=Urun.UrunAdresId
LEFT JOIN StokTakip ON StokTakip.StokTakipId=StokTakip.StokTakipId
WHERE MalKodu='{0}',", cmbxmlkodu.Text), connection);
SqlDataAdapter dradapter = new SqlDataAdapter(komut);
DataTable dt = new DataTable();
dradapter.Fill(dt);
dtListele.DataSource = dt;
baglanti.Close();
ToplamMiktar();
}
```

## Açıklama:

Sql bağlantısı kurulur toplam miktar adında metot oluşturulur. 5.sütundaki toplam stok adedi hesaplanılır ve lbltoplam içine yazılır.

Daha sonra listele butonuna basılır ve ComboBox 'tan seçilen mal koduna göre bize kayıtlı olan verileri listeler.

Bağlantı kapatılır.

```

1 reference
private void StokTakibi_Load(object sender, EventArgs e)
{
    //ürün tablosundan veri çekip comboBox içerisine koyuyoruz.
    cmbboxmlkodu.Items.Clear();
    cmd = new SqlCommand(string.Format(@"select * from Urun", cmbboxmlkodu.Text), connection);
    da = new SqlDataAdapter(cmd);
    DataTable tablo = new DataTable();
    da.Fill(tablo);
    cmbboxmlkodu.DisplayMember = "MalKodu";
    cmbboxmlkodu.ValueMember = "UrunId";
    cmbboxmlkodu.DataSource = tablo;

    cmbboxmaladi.DisplayMember = "MalAdi";
    cmbboxmaladi.ValueMember = "UrunId";
    cmbboxmaladi.DataSource = tablo;
}

```

Açıklama:

Veritabanından ürünler çekilir ve ComboBox içine veriler yüklenir seçilen mal koduna ait mal adi otomatik olarak yazdırılır.

Sipariş Toplama Sayfası:

FAST WMS / Depo Yönetim Otomasyon Sistemi / Depo

**Ürün Adresi**

**Stok Takibi**

**Sipariş Toplama**

**Mal Kabul**

**Ayarlar**

**Yardım**

Mal Kodu

Miktar

EKLE

Evrak Numarası

Seçilen Miktar

LISTELE

Tarih

Kalan Miktar

SİL

	SiparisHareketId	MalKodu	EvrakNo	Tarih	Miktar	SecilenMiktar	KalanMiktar
▶	2	5744	25487	2.02.2022	50	3	2
	3	5748	25578	2.02.2022	30	25	24
	5	2145	25	2.02.2022	25	25	25
*							

## -KODLARI-

```
1 reference
private void btnEkle_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand(@"
Insert into SiparisHareket
(MalKodu,EvrakNo,Tarih,Miktar,SecilenMiktar,KalanMiktar)
values
(@MalKodu,@EvrakNo,@Tarih,@Miktar,@SecilenMiktar,@KalanMiktar)", baglanti);
    komut.Parameters.Clear();
    komut.Parameters.AddWithValue("@MalKodu", txtMalKodu.Text);
    komut.Parameters.AddWithValue("@EvrakNo", txtEvrakNo.Text);
    komut.Parameters.AddWithValue("@Tarih", txtTarih.Text);
    komut.Parameters.AddWithValue("@Miktar", txtMiktar.Text);
    komut.Parameters.AddWithValue("@SecilenMiktar", txtSecilenMiktar.Text);
    komut.Parameters.AddWithValue("@KalanMiktar", txtKalanMiktar.Text);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show(" İşleminiz başarıyla oluşturulmuştur.", "Kayıt Ekranı");
}

1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    string getir = "select SiparisHareketId,MalKodu,EvrakNo,Tarih,Miktar,SecilenMiktar,KalanMiktar from SiparisHareket";
    SqlCommand komut = new SqlCommand(getir, baglanti);
    SqlDataAdapter dradapter = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    dradapter.Fill(dt);
    dtListeleme.DataSource = dt;
    baglanti.Close();
}
```

Listeleme butonuna tıklanınca Sipariş Toplama sorgusu çalışır ve veriler getirilir.

Ekleme butonuna tıklanınca yeni bir Sipariş Toplama kaydı oluşturulur ve işlemin başarıyla oluşturulduğu hakkında bilgi mesaj kutusu açılır.

```
1 reference
private void BtnSil_Click(object sender, EventArgs e)
{
    DialogResult secim = new DialogResult();
    secim = MessageBox.Show("Seçilen Kaydı Silmek İstediyinize Emin Misiniz ? ", "Silme", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (secim == DialogResult.Yes)
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand(" delete from SiparisHareket where MalKodu= '" + dtListeleme.CurrentRow.Cells["MalKodu"].Value.ToString() + "'", baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kaydınız Silindi");
    }

    if (secim == DialogResult.No)
    {
        MessageBox.Show("Kaydınız Silinemedi!");
    }
}
```

Sil butonuna basınca ilk önce kullanıcıya emin olup olmadığı sorulur ve evet butonuna tıklanılır ise veriler silinir. Hayır der ise sekme kapatılır ve hiçbir işlem yapılmaz.

## Mal Kabul Sayfası:

FAST WMS / Depo Yönetim Otomasyon Sistemi / Depo

Ürün Adresi

Stok Takibi

Sipariş Toplama

Mal Kabul

Ayarlar

Yardım

Mal Kodu

2145

Giriş / Çıkış

g

LİSTELE

Mal Adı

etek

Evrak Tipi

gk

EKLE

Evrak Numarası

8574

Tarih

02-02-2022

SİL

Miktar

25

	MalKabulId	MalKodu	MalAdi	EvrakNo	Giris/Cikis	Evrak Tipi	Tarih	Miktar
▶	1	1254	Elbise	2547	G	GF	2.02.2022	20
	6	1114	Çanta	1114	G	GF	5.05.2023	58
	14	2145	etek	8574	g	gk	2.02.2022	25
	15	2145	etek	8574	g	gk	2.02.2022	25
*								

## -KODLARI-

```
1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    //mal kabul sorgusu çekilir.
    string getir = "Select * from MalKabul";
    SqlCommand komut = new SqlCommand(getir, baglanti);
    SqlDataAdapter dradapter = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    dradapter.Fill(dt);
    dtListeleme.DataSource = dt;
    baglanti.Close();
}

1 reference
private void btnEkle_Click(object sender, EventArgs e)
{
    //yeni bir malkabul ekleme sorgusu
    baglanti.Open();
    SqlCommand komut = new SqlCommand(@"
Insert into MalKabul
(MalKodu,MalAdi,EvrakNo,[Giris/Cikis],EvrakTipi,Tarih,Miktar)
values
(@MalKodu,@MalAdi,@EvrakNo,@GirisCikis,@EvrakTip,@Tarih,@Miktar)", baglanti);
    komut.Parameters.Clear();
    //Her bir sütun için parametre açılır.
    komut.Parameters.AddWithValue("@MalKodu", txtMalKodu.Text);
    komut.Parameters.Add("@MalAdi", SqlDbType.NVarChar).Value = txtMalAdi.Text;
    komut.Parameters.AddWithValue("@EvrakNo", txtEvrakNo.Text);
    komut.Parameters.AddWithValue("@GirisCikis", txtGC.Text);
    komut.Parameters.AddWithValue("@EvrakTipi", txtEvrakTipi.Text);
    komut.Parameters.AddWithValue("@Tarih", txtTarih.Text);
    komut.Parameters.AddWithValue("@Miktar", txtMiktar.Text);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("İşleminiz başarıyla oluşturulmuştur.", "Kayıt Ekranı");
}
```

## Açıklama:

Listeleme butonuna tıklanınca Mal kabul sorgusu çalışır ve veriler getirilir.

Ekleme butonuna tıklanınca yeni bir mal kabul kaydı oluşturulur ve işlemin başarıyla oluşturulduğu hakkında bilgi mesaj kutusu açılır.

```

1 reference
private void dtListeleme_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
    txtMalKodu.Text = dtListeleme.CurrentRow.Cells["MalKodu"].Value.ToString();
    txtMalAdi.Text = dtListeleme.CurrentRow.Cells["MalAdi"].Value.ToString();
    txtEvrakNo.Text = dtListeleme.CurrentRow.Cells["EvrakNo"].Value.ToString();
    txtGC.Text = dtListeleme.CurrentRow.Cells["Giris/Cikis"].Value.ToString();
    txtEvrakTip.Text = dtListeleme.CurrentRow.Cells["EvrakTip"].Value.ToString();
    txtTarih.Text = dtListeleme.CurrentRow.Cells["Tarih"].Value.ToString();
    txtMiktar.Text = dtListeleme.CurrentRow.Cells["Miktar"].Value.ToString();
}

1 reference
private void BtnSil_Click(object sender, EventArgs e)
{
    DialogResult secim = new DialogResult();
    secim = MessageBox.Show("Seçilen Kaydı Silmek İstediyinize Emin Misiniz ? ", "Silme", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (secim == DialogResult.Yes)
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand(" delete from MalKabul where MalKodu= '" + dtListeleme.CurrentRow.Cells["MalKodu"].Value.ToString() + "'", baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kaydınız Silindi");
    }

    if (secim == DialogResult.No)
    {
        MessageBox.Show("Kaydınız Silinemedi!");
    }
}

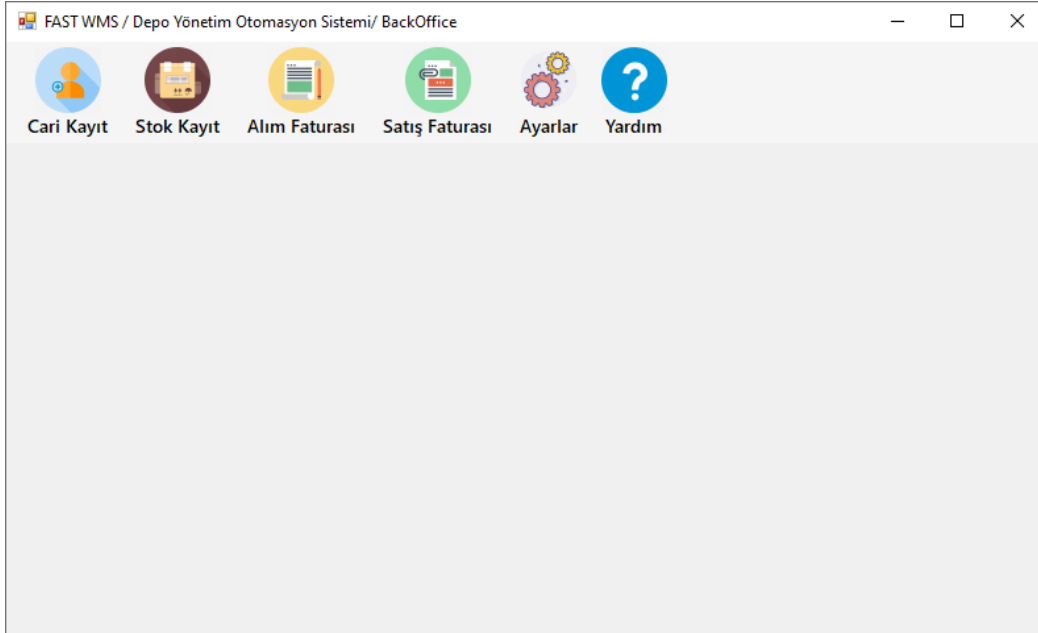
```

Açıklama:

Listeleme ekranında 2 kez tıklanınca veriler yukarıdaki text içine dolar.

Sil butonuna basınca ilk önce kullanıcıya emin olup olmadığı sorulur ve evet butonuna tıklanılır ise veriler silinir. Hayır der ise sekme kapatılır ve hiçbir işlem yapılmaz.

BackOffice Ekranı:



## Cari Kayıt Sayfası:

FAST WMS / Depo Yönetim Otomasyon Sistemi/ BackOffice

**Cari Kayıt** **Stok Kayıt** **Alım Faturası** **Satış Faturası** **Ayarlar** **Yardım**

Cari Kodu  Bakiye  Posta Kodu  Telefon   
Unvan  İl  Ülke   
Unvan2  İlçe  Adres   
[Cari Kod Arama](#)

**EKLE** **GÜNCELLE** **SİL**

**LİSTELE** **TEMİZLE**

Toplam Cari Kayıt 4

	CariId	Kod	Unvan	Unvan2	Bakiye	İl	İlçe	PostaKodu
▶	5	DP0087	ŞENAY MERT		25,00	İstanbul	Ümraniye	34773
	6	DP0017	BEYZA DURMUŞ		500,00	Ankara	Çankaya	06590
	7	DP0036	ENGİN POLAT		1000000,00	Hatay	Antakya	31060
	8	DP0005	BERKE KIRIK		450,00	İzmir	Gazimir	35410
*								

## -KODLARI-

```
1 reference
public void verilerigoster(string veriler)
{
    //verileti datagridde göstermek için metot oluşturulur.
    SqlDataAdapter adapter = new SqlDataAdapter(veriler, baglanti);
    DataSet dataset = new DataSet();
    adapter.Fill(dataset);
    dtListemele.DataSource = dataset.Tables[0];
}

1 reference
private void Kayitsayisiyaz()
{
    //toplam cari sayısı saydırılır. -1 var çünkü halihazırda 1 satır bulunmaktadır onu saymaması için yapılır.
    int kayitsayisi = dtListemele.Rows.Count - 1;
    lblToplam.Text = Convert.ToString(kayitsayisi);
}

1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    //sql den cari kayıtları listelenir ve Kayitsayisiyaz() metodu çağırılır.
    string getir = "Select * from Cari";
    SqlCommand komut = new SqlCommand(getir, baglanti);
    SqlDataAdapter dradapter = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    dradapter.Fill(dt);
    dtListemele.DataSource = dt;
    baglanti.Close();
    Kayitsayisiyaz();
}
```

## Açıklama:

2 tane metot oluşturulur kayıt sayısı yaz ile toplam cari sayısı bulma formülü yazılır ve listeleme butonundan çağırılır.

Listeleme butonuna basılınca Sql den cariler listelenir. Bağlantı kapatılır.





## -KODLARI-

```
1 reference
private void Kayitsayisiyaz()
{
    //mevcut stok adedi hesaplanılır.
    int kayitsayisi = dtListele.Rows.Count - 1;
    lblToplam.Text = Convert.ToString(kayitsayisi);
}

1 reference
private void btnBarkodOlustur_Click(object sender, EventArgs e)
{
    //Barkod oluşturulur.
    Zen.Barcode.Code128BarcodeDraw brc = Zen.Barcode.BarcodeDrawFactory.Code128WithChecksum;
    barkod.Image = brc.Draw(txtBarkod.Text, 75);
}

1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    // Stoklar listelenir.
    string getir = "Select * from Stok";
    SqlCommand komut = new SqlCommand(getir, baglanti);
    SqlDataAdapter dradapter = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    dradapter.Fill(dt);
    dtListele.DataSource = dt;
    baglanti.Close();
    Kayitsayisiyaz();
}

1 reference
private void btnTemizle_Click(object sender, EventArgs e)
{
    //temizle butonuna tıklanınca kutudaki veriler silinir
    txtUrunKodu.Clear();
    txtUrunAdi.Clear();
    txtUrunAdi2.Clear();
    txtDovizCinsi.Clear();
    txtFiyat.Clear();
    txtMarka.Clear();
    txtBirim.Clear();
    txtSkt.Clear();
}
```

Açıklama:

Mevcut stok sayısını öğrenmek için metot oluşturulur.

Barkod oluşturmak için yeni bir metot yazılır.

Listeleme butonuna tıklanınca stok tablosundan veriler çekilir. Ve Kayitsayisiyaz metodu çağırılır.

Temizle butonuna tıklanınca seçilen stoklar textbox içine yazıldığı için clear () ile kutuların içleri temizlenir.

```

private void dtListele_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    txtUrunKodu.Text = dtListele.CurrentRow.Cells["UrunKodu"].Value.ToString();
    txtUrunAdi.Text = dtListele.CurrentRow.Cells["UrunAdi"].Value.ToString();
    txtUrunAdi2.Text = dtListele.CurrentRow.Cells["UrunAdi2"].Value.ToString();
    txtDovizCinsi.Text = dtListele.CurrentRow.Cells["DovizCinsi"].Value.ToString();
    txtFiyat.Text = dtListele.CurrentRow.Cells["SatisFiyati"].Value.ToString();
    txtMarka.Text = dtListele.CurrentRow.Cells["Marka"].Value.ToString();
    txtBirim.Text = dtListele.CurrentRow.Cells["Birim"].Value.ToString();
    txtSkt.Text = dtListele.CurrentRow.Cells["Skt"].Value.ToString();
    txtBarkod.Text = dtListele.CurrentRow.Cells["Barkod"].Value.ToString();
}

1 reference
private void btnGuncelle_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Update Stok set UrunAdi=@UrunAdi,UrunAdi2=@UrunAdi2,DovizCinsi=@DovizCinsi,SatisFiyati=@SatisFiyati,Marka=@Marka", baglanti);
    komut.Parameters.AddWithValue("@UrunKodu", txtUrunKodu.Text);
    komut.Parameters.AddWithValue("@UrunAdi", txtUrunAdi.Text);
    komut.Parameters.AddWithValue("@UrunAdi2", txtUrunAdi2.Text);
    komut.Parameters.AddWithValue("@DovizCinsi", txtDovizCinsi.Text);
    komut.Parameters.AddWithValue("@SatisFiyati", Double.Parse(txtFiyat.Text));
    komut.Parameters.AddWithValue("@Marka", txtMarka.Text);
    komut.Parameters.AddWithValue("@Birim", txtBirim.Text);
    komut.Parameters.AddWithValue("@Skt", txtSkt.Text);
    komut.Parameters.AddWithValue("@Barkod", txtBarkod.Text);
    komut.ExecuteNonQuery();
    verilerigoster("Select * from Stok");
    baglanti.Close();
    MessageBox.Show("Stok Kaydı Güncellendi.", "Güncelleme Ekranı");
    foreach (Control item in this.Controls)
    {
        if (item is TextBox)
        {
            item.Text = "";
        }
    }
}

```

Açıklama:

Datagride çift tıklandığı zaman veriler textbox içene dolar

Güncelle butonuna tıklandığı zaman stok tablosundan update işlemi gerçekleştirilir. Her bir alan için parametre oluşturuyoruz. Verilerigoster metodu ile stok tablosunu çekiyoruz.

Bağlantıyı kapatıyoruz.

Ve kayıt güncelleme işlemi başarılıdır diye mesaj ekranı gösterilir.

Kutuların içindeki veriler temizlenir.

```

1 reference
private void BtnSil_Click(object sender, EventArgs e)
{
    DialogResult secim = new DialogResult();
    //Kullanıcıya seçim yaptırıyoruz
    secim = MessageBox.Show("Seçilen Kaydı Silmek İstediginize Emin Misiniz ? ", "Silme", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    //Evet der ise kayıt silinir.
    if (secim == DialogResult.Yes)
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand(" delete from Stok where UrunKodu= '" + dtListele.CurrentRow.Cells["UrunKodu"].Value.ToString() + "'", baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kaydınız Silindi");
    }
    //Hayır der ise kaydınız silinemedi mesajı verilir.
    if (secim == DialogResult.No)
    {
        MessageBox.Show("Kaydınız Silinemedi!");
    }
}

```

Açıklama:

Sil butonuna tıklanıldığında zaman dialogResult ekranı açılır ve kullanıcıya seçme şansı verilir eğer evet der ise veri silinir kullanıcı hayır der ise veriler silinmez.

```

private void btnEkle_Click(object sender, EventArgs e)
{
    if (String.IsNullOrEmpty(txtUrunKodu.Text) || string.IsNullOrEmpty(txtUrunKodu.Text) || string.IsNullOrEmpty(txtUrunKodu.Text) || string.IsNullOrEmpty(txtFiyat.Text) || string.IsNullOrEmpty(txtMarka.Text) || string.IsNullOrEmpty(txtBirim.Text) || string.IsNullOrEmpty(txtSkt.Text) || string.IsNullOrEmpty(txtBarkod.Text))
    {
        MessageBox.Show("Boş Alanları Doldurunuz", "Eksik İşlem Ekranı");
        return;
    }
    else
    {
        cmd = new SqlCommand();
        connection.Open(); //bağlantı açılır
        cmd.Connection = connection;
        //Kullanıcı adının kayıtlı olup olmadığını kontrol eder
        cmd.CommandText = "Select * from Stok where UrunKodu ='" + txtUrunKodu.Text + "'";
        dr = cmd.ExecuteReader();
        //Eğer girilen kullanıcı adı veritabanında var ise uyarı ekranı alırız
        if (dr.Read())
        {
            MessageBox.Show("Girilen Ürün Kodu Sisteme Kayıtlıdır. Lütfen Yeni Bir Ürün Kodu Giriniz ", "Uyarı Ekranı");
            return;
        }
        connection.Close();
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Insert into Stok (UrunKodu,UrunAdi,UrunAdi2,DovizCinsi,SatisFiyati,Marka,Birim,Skt,Barkod) values (@UrunKodu,@UrunAdi,@UrunAdi2, @DovizCinsi,@SatisFiyati,@Marka,@Birim,@Skt,@Barkod)", connection);
        komut.Parameters.AddWithValue("@UrunKodu", txtUrunKodu.Text);
        komut.Parameters.AddWithValue("@UrunAdi", txtUrunAdi.Text);
        komut.Parameters.AddWithValue("@UrunAdi2", txtUrunAdi2.Text);
        komut.Parameters.AddWithValue("@DovizCinsi", txtDovizCinsi.Text);
        komut.Parameters.AddWithValue("@SatisFiyati", txtFiyat.Text);
        komut.Parameters.AddWithValue("@Marka", txtMarka.Text);
        komut.Parameters.AddWithValue("@Birim", txtBirim.Text);
        komut.Parameters.AddWithValue("@Skt", txtSkt.Text);
        komut.Parameters.AddWithValue("@Barkod", txtBarkod.Text);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Müşteri kaydınız başarıyla oluşturulmuştur.", "Kayıt Ekranı");
    }
}

```

Açıklama:

Ekle butonuna tıklanıldığı zaman ilk önce boşlukların dolu olup olmadığı kontrol edilir. Eğer boş ise uyarı ekranı verir değil ise ilk olarak kayıtlı stok var mı kontrol edilir var ise uyarı ekranı alınır yok ise yeni stok eklenilir.

Alım Faturası Sayfası:

FAST WMS / Depo Yönetim Otomasyon Sistemi/ BackOffice

**Cari Kayıt**

**Stok Kayıt**

**Alım Faturası**

**Satış Faturası**

**Ayarlar**

**Yardım**

Cari Kodu

Fatura Numarası

**KAYDET**

Unvan

Tarih

**LİSTELE**

	FaturaId	Tür	Kod	Miktar	Birim	BirimFiyat	Kdv	NetTutar
▶	1			10		20.00	50	250
*								

Açıklama :

Ara Toplam :

KDV :

Net Toplam :

## -KODLARI-

```
public void hesapla()
{
    int araToplam = 0;
    int kdv = 0;
    int netToplam = 0;

    for (int i = 0; i < dtListele.Rows.Count - 1; i++)
    {
        var miktar = Convert.ToInt32(dtListele[3, i].Value.ToString() == "" ? 0 : dtListele[3, i].Value);
        var birimFiyat = Convert.ToInt32(dtListele[5, i].Value.ToString() == "" ? 0 : dtListele[5, i].Value);

        araToplam += miktar * birimFiyat;
        kdv += Convert.ToInt32(dtListele.Rows[i].Cells[6].Value.ToString() == "" ? 0 : dtListele.Rows[i].Cells[6].Value);
        netToplam += Convert.ToInt32(dtListele.Rows[i].Cells[7].Value.ToString() == "" ? 0 : dtListele.Rows[i].Cells[7].Value);
    }
    txtAraToplam.Text = araToplam.ToString();
    textBox5.Text = kdv.ToString();
    textBox4.Text = netToplam.ToString();
}

1 reference
private void dtListele_CellValueChanged(object sender, DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 3 || e.ColumnIndex == 5 || e.ColumnIndex == 6) //Miktar, BirimFiyat, Kdv
    {
        //eğer miktar birimfiyat boş gelir ise
        if (dtListele[3, e.RowIndex].Value.ToString() == "" || dtListele[5, e.RowIndex].Value.ToString() == "")
        {
            //kdv ile net tutar 0 gelir.
            dtListele[6, e.RowIndex].Value = 0;
            dtListele[7, e.RowIndex].Value = 0;
        }
        else //boş değil ise datagridview üzerinde işlemler yapılır
        {
            var miktar = Convert.ToInt32(dtListele[3, e.RowIndex].Value.ToString() == "" ? 0 : dtListele[3, e.RowIndex].Value);
            var birimFiyat = Convert.ToInt32(dtListele[5, e.RowIndex].Value.ToString() == "" ? 0 : dtListele[5, e.RowIndex].Value);
            var kdv = Convert.ToInt32(dtListele[6, e.RowIndex].Value.ToString() == "" ? 0 : dtListele[6, e.RowIndex].Value);
            //net tutar hesaplanır.
            dtListele[7, e.RowIndex].Value = miktar * birimFiyat + kdv;
        }
        hesapla();
    }
    return;
}
```

Açıklama:

Hesapla metodu oluşturulur ve DataGridView içinde olan ara toplam, kdv ve net toplam textbox içine yazdırılır.

DataGridView içine yazılan miktar, birim fiyat ve kdv değerleri ilk önce dolu mu boş mu olduğuna bakılarak boş ise değerlerini 0 girerek işleme başlar.

Daha sonra miktar ile birim fiyat çarpılır ve üstüne kdv eklenilerek net toplam bulunur.

```
1 reference
private void btnEkle_Click(object sender, EventArgs e)
{
    //datagridview içine yazılan bilgiler sqle kayıt edilir.
    builder = new SqlCommandBuilder(da);
    da.Update(dataSetds, "AlimFaturaHareketi");
    MessageBox.Show("Girmiş olduğunuz bilgiler başarıyla kaydedildi.");
}

1 reference
void Listele()
{
    //Alım fatura verileri listelenir
    baglanti.Open();
    da = new SqlDataAdapter("select * from AlimFaturaHareketi", baglanti);
    da.Fill(dataSetds, "AlimFaturaHareketi");
    dtListele.DataSource = dataSetds.Tables["AlimFaturaHareketi"];
}


1 reference
private void Alim_Load(object sender, EventArgs e)
{
    Listele();
}
```


Açıklama:


Ekle butonuna tıklanınca DataGridView içindeki bilgiler SQL'e kaydedilir. Veriler listelenir


## Satış Faturası Sayfası:


FAST WMS / Depo Yönetim Otomasyon Sistemi/ BackOffice


  
Cari Kayıt

  
Stok Kayıt

  
Alım Faturası

  
Satış Faturası

  
Ayarlar

  
Yardım

Cari Kodu

Fatura Numarası

KAYDET

Ünvan

Tarih

LİSTELE

	Faturalıd	Tür	Kod	Miktar	Birim	BirimFiyat	Kdv	NetTutar
▶				40		5	20	220
				50		4	100	300
*								

Açıklama :

Ara Toplam :

KDV :

Net Toplam :

## -KODLARI-

```
public void hesapla()
{
    int araToplam = 0;
    int kdv = 0;
    int netToplam = 0;

    for (int i = 0; i < dtListele.Rows.Count-1; i++)
    {
        var miktar = Convert.ToInt32(dtListele[3, i].Value.ToString() == "" ? 0 : dtListele[3, i].Value);
        var birimFiyat = Convert.ToInt32(dtListele[5, i].Value.ToString() == "" ? 0 : dtListele[5, i].Value);

        araToplam += miktar * birimFiyat;
        kdv += Convert.ToInt32(dtListele.Rows[i].Cells[6].Value.ToString() == "" ? 0 : dtListele.Rows[i].Cells[6].Value);
        netToplam += Convert.ToInt32(dtListele.Rows[i].Cells[7].Value.ToString() == "" ? 0 : dtListele.Rows[i].Cells[7].Value);
    }

    txtAraToplam.Text = araToplam.ToString();
    textBox5.Text = kdv.ToString();
    textBox4.Text = netToplam.ToString();
}

1 reference
private void dtListele_CellValueChanged(object sender, DataGridViewCellEventArgs e)
{
    if (e.ColumnIndex == 3 || e.ColumnIndex == 5 || e.ColumnIndex == 6)//Miktar, BirimFiyat, Kdv
    {
        if (dtListele[3, e.RowIndex].Value.ToString() == "" || dtListele[5, e.RowIndex].Value.ToString() == "")
        {
            dtListele[6, e.RowIndex].Value = 0;
            dtListele[7, e.RowIndex].Value = 0;
        }
        else
        {
            var miktar = Convert.ToInt32(dtListele[3, e.RowIndex].Value.ToString() == "" ? 0 : dtListele[3, e.RowIndex].Value);
            var birimFiyat = Convert.ToInt32(dtListele[5, e.RowIndex].Value.ToString() == "" ? 0 : dtListele[5, e.RowIndex].Value);
            var kdv = Convert.ToInt32(dtListele[6, e.RowIndex].Value.ToString() == "" ? 0 : dtListele[6, e.RowIndex].Value);

            dtListele[7, e.RowIndex].Value = miktar * birimFiyat + kdv;
        }

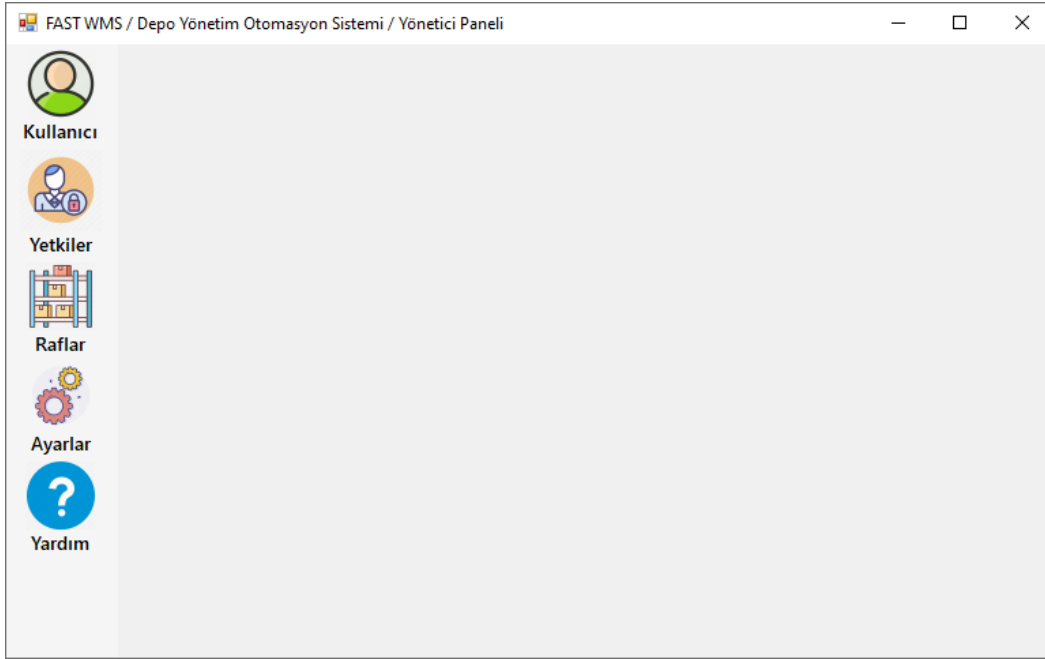
        hesapla();
        return;
    }
}
```

## Açıklama:

Hesapla metodu oluşturulur ve DataGridView içinde olan ara toplam, kdv ve net toplam textbox içine yazdırılır.

DataGridView içine yazılan miktar, birim fiyat ve kdv değerleri ilk önce dolu mu boş mu olduğuna bakılarak boş ise değerlerini 0 girerek işleme başlar. Daha sonra miktar ile birim fiyat çarpılır ve üstüne kdv eklenilerek net toplam bulunur.

## Yönetici Paneli:



## Kullanıcı Kayıt Sayfası:

The screenshot shows the 'Yeni Kayıt' (New Registration) page within the 'Yönetici Paneli' of the FAST WMS / Depo Yönetim Otomasyon Sistemi. The window title is 'FAST WMS / Yönetici Paneli / Yeni Kayıt'. The left sidebar is identical to the previous screenshot. The main area features a large graphic of a network of user icons. Below this graphic, there is a registration form with the following fields and labels: 'Ad Soyad' (Full Name), 'Kullanıcı Adı' (Username), 'E-mail', 'Grup' (Group) with a dropdown menu showing 'Depo', 'Şifre' (Password), and 'Şifre Tekrar' (Repeat Password). At the bottom of the form is a button labeled 'EKLE' (Add).

## -KODLARI-

```
1 reference
private void button1_Click(object sender, EventArgs e)//yeni kayıt
{
    try
    {
        //alanların boş olup olmadığını kontrol eder doğrusu messageBoks çalışır
        if (String.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) || string.IsNullOrEmpty(comboBox1.Text)
            || string.IsNullOrEmpty(textBox3.Text) || string.IsNullOrEmpty(textBox4.Text) || string.IsNullOrEmpty(textBox5.Text))
        {
            MessageBox.Show("Boş Alanları Doldurunuz", "Eksik İşlem Ekranı");
        }
        //Girilen iki şifre birbiriyle aynı değişse messageBox çalışır
        else if (textBox3.Text != textBox5.Text)
        {
            MessageBox.Show("Şifre Tekrarı Hatalı", "Parola Kayıt Ekranı");
        }
        else
        {
            cmd = new SqlCommand();
            connection.Open();//bağlantı açılır
            cmd.Connection = connection;
            //kullanıcı adının kayıtlı olup olmadığını kontrol eder
            cmd.CommandText = "Select * From KullaniciKayit where KullaniciAdi='" + textBox2.Text + "'";

            dr = cmd.ExecuteReader();
            //Eğer girilen kullanıcı adı veritabanında var ise uyarı ekranı alırız
            if (dr.Read())
            {
                MessageBox.Show("Kullanıcı Mevcut", "Uyarı Ekranı");
                return;
            }
        }
        dr.Close();
    }
}
```

*Tablo 6/Kontrol Ayarları*

### Açıklama:

Verilen her bir textbox doldurulmak zorunludur. Eğer boş geçilirse boş geçilemez hatası ile karşılaşırız.

Kaydedilmek istenilen kullanıcıdan eğer var ise aynı kullanıcıyı kaydedemeyeceğimiz için kullanıcı mevcut hatası alırız.

Veya girilen şifre ve şifre tekrarı birbiri ile uyuşmaz ise yine hata alınır ve kullanıcı kaydedilemez.



```

//Kayıt sorgusu oluşturulur ve textlere yazılan veriler sql de kaydolur.
//oluşturulan parametreleri textboxtan çekip Kullanicikayıtlar tablomuzaya arat
cmd = new SqlCommand(@"
Insert into Kullanicikayıtlar
(AdSoyad,KullanicikayıtlarAdi,Email,Grup,Sifre,SifreTekrar)
values
(@AdSoyad,@KullanicikayıtlarAdi,@Email,@Grup,@Sifre,@SifreTekrar)", connection);

//oluşturulan parametreleri textboxtan çekip Kullanicikayıtlar tablomuzaya arat
cmd.Parameters.AddWithValue("@AdSoyad", textBox1.Text);
cmd.Parameters.AddWithValue("@KullanicikayıtlarAdi", textBox2.Text);
cmd.Parameters.AddWithValue("@Email", textBox4.Text);
cmd.Parameters.AddWithValue("@Grup", comboBox1.SelectedValue);
cmd.Parameters.AddWithValue("@Sifre", textBox3.Text);
cmd.Parameters.AddWithValue("@SifreTekrar", textBox5.Text);
cmd.ExecuteNonQuery();
connection.Close(); //bağlantı kapanır ve yeni kişi oluşturulur

MessageBox.Show("Kaydınız başarıyla oluşturulmuştur.", "Kayıt Ekranı");

//kişi oluşturulunca textlerin içini temizler.
textBox1.Clear();
textBox2.Clear();
comboBox1.Text = "";
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();

this.Hide();//sayfa kapansın
Giris f2 = new Giris();
f2.Show();//giriş sayfasına yönlendirilsin.
}
}
catch (Exception) //hata yakalarsa gelecek olan mesaj
{
    MessageBox.Show("Hatalı Giriş Yaptınız", "HataEkranı");
}
finally
{
    connection.Close(); //bağlantı kapatılır
}
}

```

Tablo 7/Veri Tabanı Kayıt

Açıklama:

İlk önce kayıt sorgusu yazıp daha sonra parametrelere değer verilir.

Sql bağlantısı kurularak veri tabanına yeni bir kullanıcı girilir ve girilen kullanıcı kaydedilerek tekrardan giriş ekranına yönlendirilir ve yeni kullanıcı ile giriş gerçekleşir.

Kayıt Sayfası Şifre Güvenliği:

Şifre	<input type="password"/>	<div><div></div></div> Şifre Zayıf
Şifre	<input type="password"/>	<div><div></div></div> Şifre Orta
Şifre	<input type="password"/>	<div><div></div></div> Şifre Güçlü

## -KODLARI-

```
1 reference
private void textBox3_TextChanged(object sender, EventArgs e)
{
    //Şifre güvenliği için progressBar kullanıyoruz girilen karakter sayısına göre
    int karaktersayisi = textBox3.Text.Length;
    //1 ve 2 karakterse şifre zayıf
    if (karaktersayisi > 0 && karaktersayisi < 3)
    {
        progressBar1.Value = 1; //uzunluk
        progressBar1.ForeColor = Color.DarkRed; //arka plan rengi
        label6.Text = "Şifre Zayıf";
    }
    //3,4,5 karakterse şifre orta
    else if (karaktersayisi >= 3 && karaktersayisi < 6)
    {
        progressBar1.Value = 2;
        progressBar1.ForeColor = Color.Yellow;
        label6.Text = "Şifre Orta";
    }
    //6,7,8 karakterse şifre güçlü
    else if (karaktersayisi >= 6 && karaktersayisi <= 8)
    {
        progressBar1.Value = 3;
        progressBar1.ForeColor = Color.DarkGreen;
        label6.Text = "Şifre Güçlü";
    }
    //hiçbiri değil 0 ise hiçbir işlem yapmaz
    else
    {
        progressBar1.Value = 0;
    }
}
```

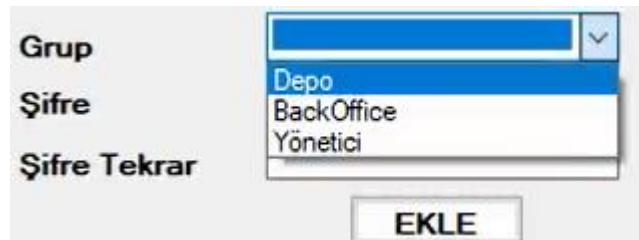
Tablo 8/Şifre Güvenliği

### Açıklama:

Girilen şifrenin güvenliğini ölçmek adına progressbar eklenerek şifrenin boyutuna göre değerlendirilme yapılır.

If Else yapısı kullanılarak şifrenin uzunluğuna bakılır ve ona göre değer verilir.

### Kayıt Sayfası Grup Belirleme:



The screenshot shows a registration form with three labels: "Grup", "Şifre", and "Şifre Tekrar". To the right of the "Grup" label is a dropdown menu with a blue header and a list of options: "Depo", "BackOffice", and "Yönetici". Below the dropdown menu is a button labeled "EKLE".

## -KODLARI-

```
1 reference
private void YöneticiPaneli_Load(object sender, EventArgs e)
{
    //Grup tablosundan veri çekip comboBox içerisine koyuyoruz.
    cmd = new SqlCommand("select * from Grup" , connection);
    da = new SqlDataAdapter(cmd);
    DataTable table = new DataTable();
    da.Fill(table);
    comboBox1.DisplayMember = "GrupAdi";
    comboBox1.ValueMember = "id";
    comboBox1.DataSource = table;
}
```

Tablo 9/Gruplandırma


Açıklama:


ComboBox içinden seçilen veri kullanıcıya ait grup kodu olarak KullaniciKayit tablosuna kaydedilir.


Ve giriş yaparken verilecek olan anasayfa ekranını belirler.


Yetkiler Sayfası:


FAST WMS / Depo Yönetim Otomasyon Sistemi / Yönetici Paneli

  
Kullanıcı

  
Yetkiler

  
Raflar

  
Ayarlar

  
Yardım

Yasemin Kaya

LİSTELE

KAYDET

	MenuId	MenuAdi	Yetkild	Kaydetme	Silme	Guncelleme
▶	1	Kullanıcı	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	2	Yetkiler	2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	3	Raflar	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	4	Ürün Adresi	6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	5	Stok Takibi	7	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	6	Sipariş Toplama	8	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	7	Mal Kabul	9	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	8	Cari Kayıt	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	9	Stok Kayıt	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	10	Satış Faturası	12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	11	Alım Faturası	13	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	12	Ayarlar	14	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## -KODLARI-

```
1 reference
private void Yetkiler_Load(object sender, EventArgs e)
{
    cmd = new SqlCommand("select * from KullaniciKayit", connection);
    da = new SqlDataAdapter(cmd);
    DataTable tablo = new DataTable();
    da.Fill(tablo);
    cmbokullanici.DisplayMember = "AdSoyad";
    cmbokullanici.ValueMember = "Id";
    cmbokullanici.DataSource = tablo;
}

2 references
private void btnListeleme_Click(object sender, EventArgs e)
{
    string KullaniciId = cmbokullanici.SelectedValue.ToString();

    string getir = @"
SELECT Menu.MenuId, Menu.MenuAdi, Yetki.YetkiId,
ISNULL(Kaydetme, 0) as Kaydetme, ISNULL(Silme,0) AS Silme, ISNULL(Guncelleme,0) AS Guncelleme
FROM Menu
LEFT JOIN Yetki ON Menu.MenuId=Yetki.MenuId AND Yetki.KullaniciKayitId=" + KullaniciId;
SqlCommand komut = new SqlCommand(getir, baglanti);
SqlDataAdapter dradapter = new SqlDataAdapter(komut);
DataTable dt = new DataTable();
dradapter.Fill(dt);
dtListele.DataSource = dt;
baglanti.Close();
}
```

### Açıklama:

Kullanıcı kayıt tablosundan veriler çekilir. ComboBox içinde ad soyad seçilir ve ona göre yetkiler verilir.

Listeleme butonuna tıklanınlınca yetki tablosundan da veriler çekilerek her bir kullanıcıya yetkileri verilir.

```
1 reference
private void button1_Click(object sender, EventArgs e)
{
    int counter = 0;
    baglanti.Open();
    for (counter = 0; counter < dtListele.RowCount; counter++)
    {
        var rows = dtListele.Rows[counter];
        if (rows.Cells["YetkiId"].Value != null && rows.Cells["YetkiId"].Value.ToString() != "")
        {
            SqlCommand komut = new SqlCommand("Update Yetki set Kaydetme=@Kaydetme, Silme=@Silme, Guncelleme=@Guncelleme where YetkiId=@YetkiId ", baglanti);
            komut.Parameters.AddWithValue("@YetkiId", rows.Cells["YetkiId"].Value);
            komut.Parameters.AddWithValue("@Kaydetme", rows.Cells["Kaydetme"].Value);
            komut.Parameters.AddWithValue("@Silme", rows.Cells["Silme"].Value);
            komut.Parameters.AddWithValue("@Guncelleme", rows.Cells["Guncelleme"].Value);
            komut.ExecuteNonQuery();
        }
        else
        {
            SqlCommand komut = new SqlCommand(@"
Insert into Yetki (MenuId,KullaniciKayitId,Kaydetme,Silme,Guncelleme)
values
(@MenuId,@KullaniciKayitId,@Kaydetme)", baglanti);
            komut.Parameters.AddWithValue("@MenuId", rows.Cells["MenuId"].Value);
            komut.Parameters.AddWithValue("@KullaniciKayitId", cmbokullanici.SelectedValue.ToString());
            komut.Parameters.AddWithValue("@Kaydetme", rows.Cells["Kaydetme"].Value);
            komut.Parameters.AddWithValue("@Silme", rows.Cells["Silme"].Value);
            komut.Parameters.AddWithValue("@Guncelleme", rows.Cells["Guncelleme"].Value);
            komut.ExecuteNonQuery();
        }
    }
    baglanti.Close();
    MessageBox.Show("Kayıt Güncellendi.", "Güncelleme Ekranı");
    btnListeleme_Click(null, null);
}
```

### Açıklama:

Kaydet butonuna tıklanınldığı zaman eğer yetkiId null ve boş değil ise update işlemi yapılır.

Eğer boş ise insert into işlemi yapılır. Bağlantı kapatılır ve kayıt güncellendi ekranı verilir.

## Raflar Sayfası:

FAST WMS / Depo Yönetim Otomasyon Sistemi / Yönetici Paneli

**Kullanıcı**

**Koridor**  **Kat**

**Raf**  **Koli Adı**

**EKLE** **GÜNCELLE** **SİL**

**LİSTELE** Toplam Ürün Adresi 2

	UrunAdresId	Koridor	Raf	Kat	KoliAdi
▶	4	4	055	6	874
	6	5	057	8	365
*					

< >

## -KODLARI-

```
2 references
private void Kayitsayisiyaz()
{
    //toplam kayıtlı olan raf sayısını yazar
    int kayitsayisi = dtListeleme.Rows.Count -1;
    lblToplam.Text= Convert.ToString(kayitsayisi);
}

1 reference
private void btnListeleme_Click(object sender, EventArgs e)
{
    //Ürünün adresi listelenir
    string getir = "Select * from UrunAdres";
    SqlCommand komut = new SqlCommand(getir, baglanti);
    SqlDataAdapter dradapter = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    dradapter.Fill(dt);
    dtListeleme.DataSource = dt;
    baglanti.Close();
    Kayitsayisiyaz();
}

1 reference
private void dtListeleme_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
    //Çift tıklanıldığında veriler kutulara doldurulur.
    txtKoridor.Text = dtListeleme.CurrentRow.Cells["Koridor"].Value.ToString();
    txtRaf.Text = dtListeleme.CurrentRow.Cells["Raf"].Value.ToString();
    txtKat.Text = dtListeleme.CurrentRow.Cells["Kat"].Value.ToString();
    txtKoliAdi.Text = dtListeleme.CurrentRow.Cells["KoliAdi"].Value.ToString();
    Kayitsayisiyaz();
}
```

Açıklama:

Toplam ürün adresinin hesaplandığı metot yazılır.

Listeleme butonuna basılınca ürünün adresleri listelenir.

DataGride çift tıklanıldığında zaman veriler kutulara dolar.

```

private void btnEkle_Click(object sender, EventArgs e)
{
    //Mutular boş mu değil mi kontrol edilir.Boş ise uyarı ekranı gelir
    if (String.IsNullOrEmpty(txtKoridor.Text) || string.IsNullOrEmpty(txtRaf.Text) || string.IsNullOrEmpty(txtMat.Text) || string.IsNullOrEmpty(txtMoliAdi.Text))
    {
        MessageBox.Show("Boş Alanları Doldurunuz", "Eksik İşlem Ekranı");
        return;
    }
    else //Dolu ise Kayıt oluşturulur.
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("Insert into UrunAdres (Koridor,Raf,Mat,MoliAdi) values (@Koridor,@Raf,@Mat,@MoliAdi)", baglanti);
        komut.Parameters.AddWithValue("@Koridor", txtKoridor.Text);
        komut.Parameters.AddWithValue("@Raf", txtRaf.Text);
        komut.Parameters.AddWithValue("@Mat", txtMat.Text);
        komut.Parameters.AddWithValue("@MoliAdi", txtMoliAdi.Text);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Ürün Adresi kaydınız başarıyla oluşturulmuştur.", "Kayıt Ekranı");
    }
}

1 reference
private void BtnSil_Click(object sender, EventArgs e)
{
    DialogResult secim = new DialogResult();
    secim = MessageBox.Show("Seçilen Kaydı Silmek İstediyinize Emin Misiniz ? ", "Silme", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (secim == DialogResult.Yes)
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand(" delete from UrunAdres where Koridor= '" + dtListeleme.CurrentRow.Cells["Koridor"].Value.ToString() + "'", baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        MessageBox.Show("Kaydınız Silindi");
    }
    if (secim == DialogResult.No)
    {
        MessageBox.Show("Kaydınız Silinemedi!");
    }
}

```

### Açıklama:

Ekle butonuna tıklanınlınca alanların boş mu dolu mu olduğu kontrol edilir. Eğer boş ise uyarı ekranı verir dolu ise insert into ile kayıt işlemi tanımlanıyor.

Sil butonuna tıklandığı zaman kullanıcıya emin olup olmadığı soruluyor eğer evet der ise kayıt siliniyor hayır der ise kayıt silinmez ve hiçbir işlem yapılmaz.

### Yardım Sayfası:

