

# DEPO TAKİP OTOMASYONU

## Analiz Dokümanı

[Mustafa TAŞAN, İŞ ANALİSTİ]

# İçindekiler

ŞEKİL TABLOSU.....	ii
1.GİRİŞ.....	1
1. DEPO OTOMASYON SİSTEMİ PROJESİ.....	1
1.1. Gereksinim Analizi.....	1
1.2. Sistem Gereksinimleri.....	1
1.3. İşlevsel Gereksinimler.....	2
1.4. İşlevsel Olmayan Gereksinimler.....	2
1.5. Veritabanı Tasarımı.....	2
1.6. Arayüz Tasarımı ve Gerçekleştirim.....	5
1.6.1. Anasayfa.....	5
1.6.2. Personel Girişi.....	7
1.6.3. Şifremi Unuttum Personel Girişi.....	11
1.6.4. Stok Girdi İşlemleri.....	13
1.6.5. Stok Girdi Detay.....	17
1.6.6. Stok Çıktı Detay.....	24
1.6.7. Rapor.....	25
1.6.8. Yönetici Girişi.....	28
1.6.9. Şifremi Unuttum Yönetici Girişi.....	32
1.6.10. Yönetici İşlemleri Detay.....	38
1.6.11. Personel İşlemleri.....	40
2.6.14. Ürün/Birim/Depo İşlemleri.....	43

## ŞEKİL TABLOSU

Şekil 1. Personel Tablosu.....	4
Şekil 2. Yönetici Tablosu.....	5
Şekil 3. Ürün Tablosu.....	5
Şekil 4. Malzeme Tablosu.....	5
Şekil 5. Depo Tablosu.....	6
Şekil 6. Birim Tablosu .....	6
Şekil 7. Çıkış Tablosu.....	6
Şekil 8. Anasayfa.....	7
Şekil 9. Personel Girişi.....	9
Şekil 10. Şifremi Unuttum Personel Girişi.....	12
Şekil 11. Stok Girdi İşlemleri.....	14
Şekil 12. Stok Girdi Detay.....	18
Şekil 13. Stok Çıktı İşlemleri.....	21
Şekil 14. Stok Çıktı Detay.....	25
Şekil 15. Rapor.....	26
Şekil 16.Yönetici Girişi.....	30
Şekil 17. Şifremi Unuttum Yönetici Girişi.....	33
Şekil 18. Yönetici İşlemleri.....	35
Şekil 19. Yönetici İşlemleri Detay.....	39
Şekil 20. Personel İşlemleri.....	41
Şekil 21. Ürün/Birim/Depo İşlemleri.....	45

# 1.GİRİŞ

Depo yönetimi başta lojistik hizmetleri olmak üzere birçok firmanın en önemli parçası haline gelmiştir. Ancak teknoloji ne kadar ilerlemiş olsa da depolar uzun bir süre boyunca manuel yönetilen alanlar olmuşlardır. Manuel yönetim bir süre sonra depoların etkin ve verimli bir şekilde iş süreçlerinin devam ettirilmemesine ve ciddi maliyet kaybına sebebiyet vermiştir. Ancak bu durum yönetim şeklinin verimliliğinden ziyade bu alanda teknolojinin kullanılmaya başlanmamasından kaynaklıdır. Geliştirdiğim depo otomasyon sistemi sayesinde manuel operasyonların ve tekrarlı işlemlerin önüne geçilebilecektir. Depo otomasyon sistemi çok basit olarak, mevcutta bulunan ya da depoya yeni giriş yapan ürünlerin sisteme kaydedilmesi ya da sistemde hali hazırda kayıtlı olan ürünlerin depodan çıkışlarının yapılmasını sağlayan bir sistemdir. Depolardaki iş süreçlerinin verimliliğini ve hızını artırılabilecek, iş süreçlerinde oluşan hatalar ve karışıklıklar en aza indirilecek ve böylece hata kaynaklı oluşan maliyetlerin de düşürülmesi sağlanacaktır. Ayrıca depoda fazla ürün bulundurmaktan kaynaklı depo işgaline ve eksik stok bulundurmaktan kaynaklı ürünlerin çıkış süresinin uzaması sorununa da çözüm olacaktır. Depo otomasyon sistemi projesi Microsoft Visual Studio ortamında C# dili kullanılarak yazılmıştır. Veritabanı olarak ise Microsoft SQL Server kullanılmıştır.

## 1. DEPO OTOMASYON SİSTEMİ PROJESİ

### 1.1. Gereksinim Analizi

Depo Takip Sistemi projesi, gerçek bir projede yer alarak eğitim süresince edindiğim bilgi birikiminin pratiğe dökülmesini sağlamak, yazılım dünyası hakkında bilgilenmem, yazılım geliştirme yeteneklerimin artırılması, stok takibini yapan depo personeli ve onu denetleyen yöneticinin işini kolaylaştırmak, bilgileri daha sistemli bir şekilde saklamak, geçmişe yönelik bilgilerin kaybolmasını önlemek amacıyla geliştirilmesi belirlendi. Yaptığım çalışmalarla mevcut bilgi seviyemin daha yukarılara çıkartılması, özellikle C# dilinde gelişme kaydetmem hedeflenmiştir.

### 1.2. Sistem Gereksinimleri

- Kullanıcı adı/E-mail ve şifre ile giriş
- Personel ekleme, silme, güncelleme ve excel 'e aktarma işlemleri
- Yönetici ekleme, silme, güncelleme ve excel 'e aktarma işlemleri
- Yönetici 'nin ürün/birim/depo ekleme silme işlemleri
- Şifre unutulduğunda mail ile değiştirebilme
- Personel 'in stok girdi işlemleri
- Personel 'in stok çıktı işlemleri
- Personel 'in raporlama işlemleri

### 1.3. İşlevsel Gereksinimler

- Yöneticiler ve personeller
- Depoya gelen ve depodan çıkan ürün bilgileri
- Rapor bilgileri
- Ürün, birim ve depo bilgileri.

### 1.4. İşlevsel Olmayan Gereksinimler

- Programı kaldırabilecek herhangi bir aksaklık ve gecikme yapmayacak bir işletim sistemi kullanılmalıdır.
- Veri kaybını önlemek için veriler mutlaka veritabanına kaydedilmelidir.
- Aynı anda depoya birçok ürün girişi ve çıkışı olacağından zaman kaybı olmaması için sistemin hızlı çalışması.
- Herhangi olası bir durum için depoya giren ve çıkan ürünlerin takibini yapan personel Id 'sinin doğru girilmesi
- Karışıklığı önlemek için ürün miktarının, ürünün depoya giriş/çıkış saat ve tarihinin iyi belirlenmesi.

### 1.5. Veritabanı Tasarımı

Öncelikle erenproje ve erenproje2 adında 2 adet veritabanı oluşturdum. Yönetici işlemleri için tüm verileri erenproje2 de, Personel, depo, ürün, birim, çıkış ve malzeme işlemleri için tüm verileri erenproje veritabanında tuttum. Ardından veritabanlarında projemde kullanmam için gerekli olan tabloları oluşturdum. İlk olarak personel ve yönetici bilgilerinin saklanmış olduğu tabloları oluşturarak oluşturduğum bu tablonun içindeki verileri doldurdum. Oluşturulan 2 adet tablo Şekil 1 ve Şekil 2' de görüldüğü gibidir.

```
CREATE TABLE [dbo].[Personel](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Ad] [nvarchar](30) NULL,
    [Soyad] [nvarchar](20) NULL,
    [TCNo] [nchar](11) NULL,
    [EMail] [varchar](100) NULL,
    [Sifre] [varchar](8) NULL,
    [IL] [nvarchar](50) NULL,
    [ILCE] [nvarchar](50) NULL,
    [Cinsiyet] [varchar](5) NULL,
    [Telefon] [char](16) NULL,
    [presim] [varchar](100) NULL,
    CONSTRAINT [PK_Personel] PRIMARY KEY CLUSTERED
    ([Id] ASC
```

Şekil 1. Personel Tablosu

```
CREATE TABLE [dbo].[Yönetici](
    [Y_Id] [int] IDENTITY(1,1) NOT NULL,
    [Ad] [nvarchar](50) NULL,
    [Soyad] [nvarchar](50) NULL,
    [TCNo] [nchar](11) NULL,
    [EMail] [varchar](100) NULL,
    [Sifre] [varchar](30) NULL,
    [IL] [nvarchar](50) NULL,
    [ILCE] [nvarchar](50) NULL,
    [Cinsiyet] [varchar](5) NULL,
    [Telefon] [char](16) NULL,
    [yresim] [varchar](50) NULL
) ON [PRIMARY]
GO
```

Şekil 2. Yönetici Tablosu

Personel girişi ve Yönetici girişi için gerekli olan tablolarımı gerçekleştirdikten sonra Ürün, Malzeme ve Depo bilgilerinin saklandığı tabloları oluşturdum. Ardından tablonun içindeki ürün malzeme ve depo verilerini doldurdum. Oluşturulan 3 adet tablo Şekil 3, Şekil 4 ve Şekil5’ de görüldüğü gibidir.

```
CREATE TABLE [dbo].[URUN](
    [urun_ID] [int] IDENTITY(1,1) NOT NULL,
    [urunadi] [nvarchar](50) NULL,
    [birimadi] [nchar](50) NULL
) ON [PRIMARY]
GO
```

Şekil 3. Ürün Tablosu

```

CREATE TABLE [dbo].[Malzemee](
    [MZM_ID] [int] IDENTITY(1,1) NOT NULL,
    [Depo] [nvarchar](50) NOT NULL,
    [MZMAdi] [nvarchar](50) NOT NULL,
    [MZMAdet] [int] NOT NULL,
    [Fiyat] [varchar](50) NULL,
    [Tarih] [nchar](30) NULL,
    [Saat] [nchar](30) NULL,
    [Aciklama] [nvarchar](max) NULL,
    [Toplam] [nvarchar](50) NULL,
    [Id] [int] NOT NULL,
    CONSTRAINT [PK_Malzemee] PRIMARY KEY CLUSTERED
(
    [MZM_ID] ASC
)

```

Şekil 4. Malzeme Tablosu

```

CREATE TABLE [dbo].[DEPO](
    [depo_ID] [int] IDENTITY(1,1) NOT NULL,
    [depoadi] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_DEPO] PRIMARY KEY CLUSTERED
(
    [depo_ID] ASC
)

```

Şekil 5. Depo Tablosu

Son olarak da ürünlerin hangi birime, ne zaman, hangi personel tarafından çıktığına vb. bilgilerin tutulduğu Birim ve Çıkış adında 2 adet tablo oluşturdum. Oluşturulan 2 adet tablo Şekil 6 ve Şekil7' de görüldüğü gibidir.

```

CREATE TABLE [dbo].[Birim](
    [Birim_ID] [int] IDENTITY(1,1) NOT NULL,
    [birimadi] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Birim] PRIMARY KEY CLUSTERED
(
    [Birim_ID] ASC
)

```

Şekil 6. Birim Tablosu

```

CREATE TABLE [dbo].[CIKIS](
    [C_ID] [int] IDENTITY(1,1) NOT NULL,
    [depoo] [nvarchar](50) NOT NULL,
    [c_adi] [nvarchar](50) NOT NULL,
    [c_adet] [int] NOT NULL,
    [Fiyat] [int] NULL,
    [c_tarih] [nchar](30) NULL,
    [c_saat] [nchar](30) NULL,
    [c_birim] [nchar](30) NOT NULL,
    [c_aciklama] [nvarchar](max) NULL,
    [c_toplam] [nvarchar](50) NULL,
    [Id] [int] NOT NULL,
    CONSTRAINT [PK_CIKIS] PRIMARY KEY CLUSTERED
(
    [C_ID] ASC

```

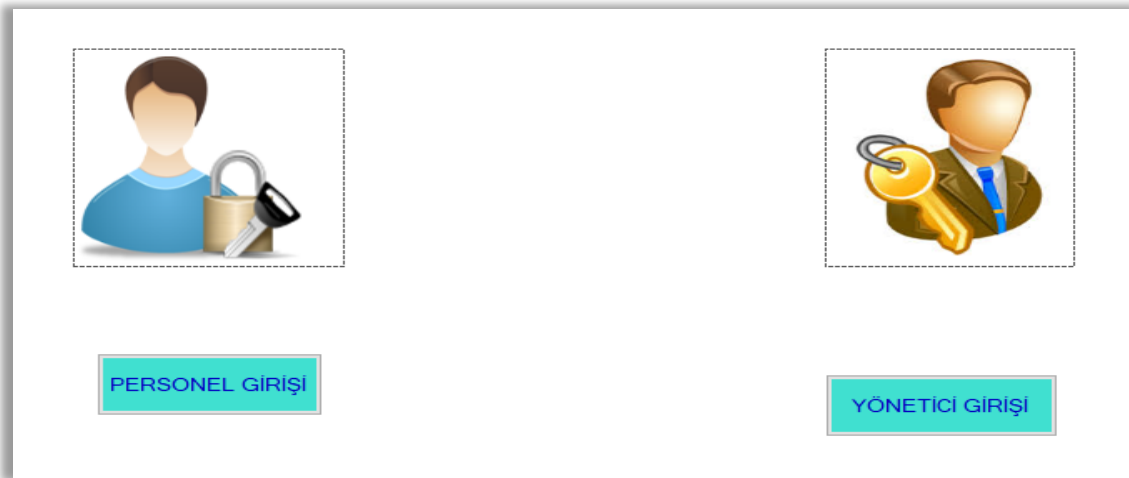
Şekil 7. Çıkış Tablosu

## 1.6. Arayüz Tasarımı ve Gerçekleştirim

Anasayfa, Personel girişi, Şifremi unuttum personel girişi, Stok girdi işlemleri, Stok çıktı işlemleri, Rapor, Detay ekranları, Yönetici girişi, Şifremi unuttum yönetici girişi, Yönetici işlemleri, Personel işlemleri ve Ürün/Birim/Depo işlemleri olmak üzere toplam 13 adet form ekranı oluşturdum.

### 1.6.1. Anasayfa

Uygulamayı kullanan Personel veya Yöneticinin karşılaştığı ilk ekrandır. Personel girişine ve Yönetici girişine bağlantının yapılabilceği 2 adet buton ve 2 adet picturebox yer almaktadır (Şekil 8.) Fare ile butonun üstüne gidince butonun renginin değişmesi, fareyi butondan çekince de eski rengine dönmesi için gerekli kod parçacıkları yer almaktadır. Fare personel girişi butonunun üstünde iken buton mavi, yönetici butonunun üstünde iken de kırmızı olmaktadır.





## Şekil 8. Anasayfa

### Kod Parçasığı;

```
private void button1_MouseMove(object sender, MouseEventArgs e)
{
    button1.BackColor = Color.Turquoise;
}

private void button1_MouseHover(object sender, EventArgs e)
{
    button1.BackColor = Color.Aqua;
}

private void button3_MouseHover(object sender, EventArgs e)
{
    button3.BackColor = Color.Red;
}

private void button3_Click(object sender, EventArgs e)
{
    Form3 frm3 = new Form3();
    frm3.Show();
}

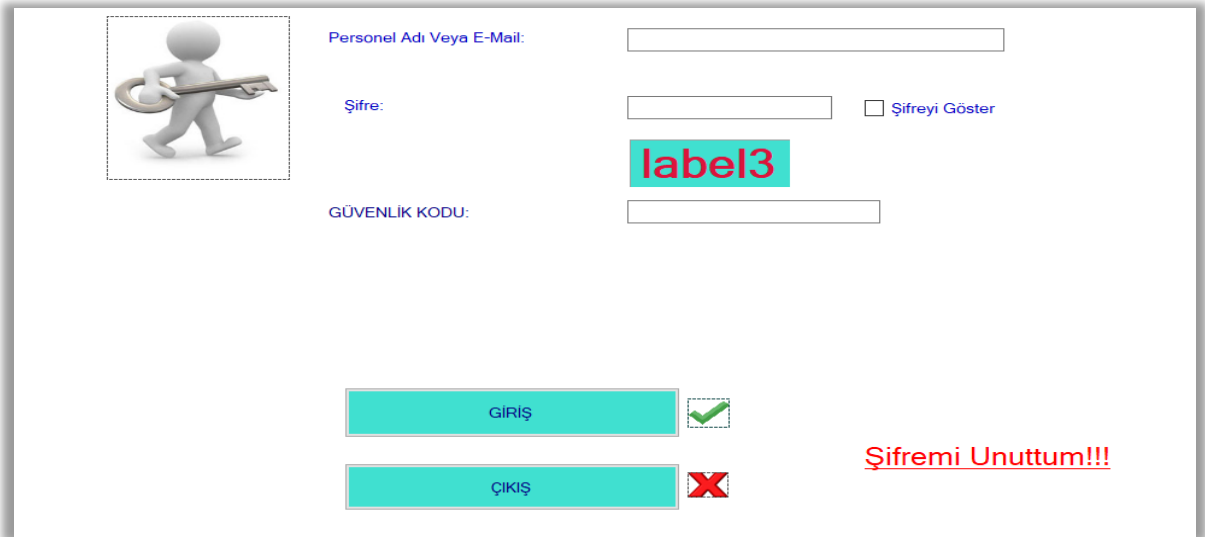
private void button1_Click(object sender, EventArgs e)
{
    Form2 yeni = new Form2();
    yeni.Show();
}

private void button3_MouseMove(object sender, MouseEventArgs e)
{
    button3.BackColor = Color.Turquoise;
}
```

## 1.6.2. Personel Giriş

Anasayfada personel girişi butonu tıklandığında bu form ekranı gelmektedir. Personel Giriş form ekranı için 4 adet label, 3 adet textbox, 2 adet buton, 1 adet checkbox, 1 adet link label ,3 adet picturebox ve 1 adet errorprovider kullanılmıştır.

Yönetici tarafından kaydedilen personel, kullanıcı adı veya e-mail, şifre ve random olarak değişen güvenlik kodunu doğru girdiği takdirde stok girdi işlemleri form ekranına yönlendirilecektir. Yanlış girdiği takdirde ise hata mesajı ile karşılaşacaktır. Şifre 8 karakter belirlendiği içim 8 karakterden fazla veya az olan şifrelerde errorprovider sayesinde hata icon' u yanıp sönecektir. Checkbox sayesinde de istendiğinde şifrenin gizlenmesi, istendiğin de ise şifrenin görülmesini sağlamıştır. Çıkış butonuna basıldığında ise anasayfaya yönlendirme gerçekleştirilmiştir.



The image shows a login form titled "Personel Giriş". On the left, there is a placeholder image of a person holding a key. The form contains three input fields: "Personel Adı Veya E-Mail:", "Şifre:", and "GÜVENLİK KODU:". The password field has a "Şifreyi Göster" checkbox and a red "label3" error message. Below the inputs are two buttons: "GİRİŞ" (with a green checkmark icon) and "ÇIKIŞ" (with a red X icon). To the right of the buttons is a red link labeled "Şifremi Unuttum!!!".

Şekil 9. Personel Giriş

**Kod Parçacığı;**

```

private void Form2_Load(object sender, EventArgs e)
{
    int a, d, s, t;
    Random r = new Random();
    a = r.Next(1, 10);
    d = r.Next(1, 10);
    s = r.Next(1, 10);
    t = r.Next(1, 10);
    label3.Text = a.ToString() + d.ToString() + s.ToString() + t.ToString();
    textBox2.MaxLength = 8;
}

private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        textBox2.PasswordChar = '\0';
    }
    else
    {
        textBox2.PasswordChar = '*';
    }
}

private void button1_MouseHover(object sender, EventArgs e)
{
}

```

```

private void button2_Click(object sender, EventArgs e)
{
    {
        DialogResult cikis = new DialogResult();

        cikis=MessageBox.Show("ÇIKMAK İSTEDİĞİNİZDEN EMİN MİSİNİZ?",
"UYARI",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button2);
        if (cikis == DialogResult.Yes)
        {
            Form1 frm1 = new Form1();

            frm1.Show();

            this.Close();
        }
        if (cikis == DialogResult.No)
        {
            MessageBox.Show("PROGRAM KAPATILMADI.");
        }
    }
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    if (textBox2.Text.Length < 8 || textBox2.Text.Length > 8)
        errorProvider1.SetError(textBox2, "8 KARAKTER OLMALI!");
    else
    {
        errorProvider1.Clear();
    }
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
}

SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        baglanti.Open();
        string sql = "Select*From Personel Where Ad=@adi OR EMail=@maili AND
Sifre=@sifresi";
        SqlParameter prm1 = new SqlParameter("adi",textBox1.Text.Trim());
        SqlParameter prm3 = new SqlParameter("maili", textBox1.Text.Trim());
        SqlParameter prm2 = new SqlParameter("sifresi", textBox2.Text.Trim());

        SqlCommand komut = new SqlCommand(sql, baglanti);

```

```

        komut.Parameters.Add(prm1);
        komut.Parameters.Add(prm3);
        komut.Parameters.Add(prm2);
        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(komut);
        da.Fill(dt);
        if(dt.Rows.Count > 0
    )
    {
        if (label3.Text == textBox3.Text)
        {
            Form7 frm7 = new Form7();
            frm7.Show();
        }
        else
        {
            MessageBox.Show("TEKRAR DENEYİNİZ...");
            int a, d, s, t;
            Random r = new Random();
            a = r.Next(1, 10);
            d = r.Next(1, 10);
            s = r.Next(1, 10);
            t = r.Next(1, 10);
            label3.Text = a.ToString() + d.ToString() + s.ToString() + t.ToString();
        }
    }

}


}
catch (Exception)
{
    MessageBox.Show("PERSONEL ADI/EMAIL VEYA ŞİFRE YANLIŞ!");
}
}

private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    Form11 frm11 = new Form11();
    frm11.Show();
}

```

### 1.6.3. Şifremi Unuttum Personel Girişi

Personel girişindeki Şifremi unuttum linklabel' ı tıklandığı zaman Şifremi Unuttum Personel Girişi Form ekranı gelmektedir. Bu form ekranı için 3 adet label, 2 adet textbox, 2 adet button ve 2 adet picturebox kullanılmıştır. Personel doğru Tc No ve E-mail adresini girdiği takdirde ekrana bilgilendirme mesajı ve şifresi e-mail adresine gelmektedir Personelin Tc No ve E-mail adresi eşleşmediği takdirde ise hata mesajı vermektedir.




ŞİFREMI UNUTTUM

PERSONEL TC NO:

PERSONEL E-MAIL:

E-MAIL GÖNDER

ÇIKIŞ



Şekil 10. Şifremi Unuttum Personel Girişi

**Kod Parçacığı;**

```

public partial class Form11 : Form
{
    public Form11()
    {
        InitializeComponent();
    }
    SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");
    public bool Gonder(string konu, string icerik)
    {
        MailMessage ePosta = new MailMessage();
        ePosta.From = new MailAddress("erencaan09@gmail.com

        ePosta.To.Add(textBox2.Text);
        ;

        ePosta.Subject = konu;

        ePosta.Body = icerik;
        //
        SmtpClient smtp = new SmtpClient();

        smtp.Credentials = new System.Net.NetworkCredential("erencaan09@gmail.com",
"erencaan58.");

        smtp.Port = 587;
        smtp.Host = "smtp.gmail.com";
        smtp.EnableSsl = true;
        object userState = ePosta;
        bool kontrol = true;
        try
        {
            smtp.SendAsync(ePosta, (object)ePosta);
        }
        catch (SmtpException ex)

            {
                kontrol = false;
                System.Windows.Forms.MessageBox.Show(ex.Message, "Mail Gönderme
Hatasi");
            }
        return kontrol;
    }
}

```

## 1.6.4. Stok Girdi İşlemleri

Doğru Personel Adı/E Mail ve Şifre ile giriş yapan kullanıcının yönlendirildiği ekrandır. Bu ekranda 8 adet label, 2 adet combobox, 1 adet numericUpDown, 3 adet listBox, 2 adet textbox, 1 adet richTextBox, 1 adet menüstrip, 2 adet pictureBox, 5 adet button ve, 1 adet dateTimePicker yer almaktadır. Personel' in depo ya gelen ürünü kaydetmesi için bu form ekranını tasarlanıp kodlanmıştır. Personel bu ekrana yönlendiğinde Personel Id otomatik olarak dolu gelmektedir. Depo ve Malzeme combobox alanına veriler depo ve malzeme tablolarından çekilerek getirilmiştir. Depo' ya birden fazla ürün geldiğinde her ürün için personelin aynı bilgileri tekrar ve tekrar girmesini önlemek için malzeme adlarını, adetini ve girdi fiyatını tutan 3 adet listBox ve listBox' a veri ekleyip silen ekle ve sil butonu kullanılmıştır.

Personel ilgili alanları doldurup “KAYDET” butonuna bastığında kayıt işlemi gerçekleşip, kayıt işleminin gerçekleştiğine dair bilgilendirme mesajı çıkmaktadır.

Personel alanlardaki verileri “TEMİZLE” butonuna basarak kolayca temizleyebilmektedir.

Şekil 11. Stok Girdi İşlemleri

**Kod Parçacığı;**



```

public string UserName { get; set; }
public Form7(string userName)
{
    InitializeComponent();
    UserName = userName;
}

private void Form7_Load(object sender, EventArgs e)
{
    SqlConnection baglanti = new SqlConnection();
    baglanti.ConnectionString = @"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True;MultipleActiveResultSets=True";
    SqlCommand komut = new SqlCommand();
    komut.CommandText = "SELECT *FROM URUN";
    komut.Connection = baglanti;
    komut.CommandType = CommandType.Text;
    baglanti.Open();
    SqlCommand command2= new SqlCommand("select * from personel where
EMail=@email",baglanti);
    command2.Parameters.AddWithValue("email",UserName);
    var data= command2.ExecuteReader();
    while (data.Read())
    {
        textBox3.Text = data["Id"].ToString();
    }
    SqlDataReader dr;

    dr = komut.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Items.Add(dr["urunadi"]);
    }
    baglanti.Close();
    depoo();
}
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");
private void verilerigoster()
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Malzemes ", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["MZM_Id"].ToString();
        ekle.SubItems.Add(oku["Depo"].ToString());
        ekle.SubItems.Add(oku["MZMAdi"].ToString());
        ekle.SubItems.Add(oku["MZMAdet"].ToString());
        ekle.SubItems.Add(oku["Fiyat"].ToString());
        ekle.SubItems.Add(oku["Tarih"].ToString());
    }
}

```

```

        ekle.SubItems.Add(oku["Saat"].ToString());
        ekle.SubItems.Add(oku["Aciklama"].ToString());

        ekle.SubItems.Add(oku["Id"].ToString());
    }
    baglanti.Close();
}

private void depoo()
{
    SqlCommand komut = new SqlCommand();
    komut.CommandText = "SELECT *FROM DEPO";
    komut.Connection = baglanti;
    komut.CommandType = CommandType.Text;

    SqlDataReader dr;
    baglanti.Open();
    dr = komut.ExecuteReader();
    while (dr.Read())
    {
        comboBox2.Items.Add(dr["depoadi"]);
    }
    baglanti.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    Form10 lst1 = new Form10();
    lst1.Show();
}

private void listView1_SelectedIndexChanged(object sender, EventArgs e)
{
}
int i = 0;
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < listBox1.Items.Count; i++)
    {
        baglanti.Open();
    }
}

```

```

SqlCommand komut = new SqlCommand("insert into Malzeme(Depo, MZMAAdi,
MZMAAdet, Fiyat, Tarih, Saat, Açıklama, Id)values('" + comboBox2.Text.ToString() + "','"+
listBox1.Items[i].ToString() + "','"+ listBox2.Items[i].ToString() + "','"+
listBox3.Items[i].ToString() + "','"+ dateTimePicker1.Value.ToString() + "','"+
textBox2.Text.ToString() + "','"+ richTextBox1.Text.ToString() + "','"+
textBox3.Text.ToString() + "','")", baglanti);

```

```

komut.ExecuteNonQuery();
baglanti.Close();
MessageBox.Show("STOK GİRDİ BAŞARIYLA GERÇEKLEŞTİRİLMİŞTİR");

```

```

}

```

```

}

```

```

private void sTOKCIKTIİSLEMLERİToolStripMenuItem_Click(object sender,
EventArgs e)

```

```

{
    Form8 frm8 = new Form8(Username);
    frm8.Show();

```

```

private void button3_Click(object sender, EventArgs e)

```

```

{
    listBox1.Items.Add(comboBox1.Text);
    listBox2.Items.Add(numericUpDown1.Text);
    listBox3.Items.Add(textBox1.Text);

```

```

}

```

```

private void button4_Click(object sender, EventArgs e)

```

```

{
    listBox1.Items.Remove(listBox1.SelectedItem);
    listBox2.Items.Remove(listBox2.SelectedItem);
    listBox3.Items.Remove(listBox3.SelectedItem);

```

```

}

```

```

private void rAPORTToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    Form12 frm12 = new Form12();
    frm12.Show();
}

```

```

private void button5_Click_2(object sender, EventArgs e)

```

```

{
    comboBox1.ResetText();
    comboBox2.ResetText();
    numericUpDown1.ResetText();
    listBox1.Items.Clear();
    listBox2.Items.Clear();
    listBox3.Items.Clear();
}

```

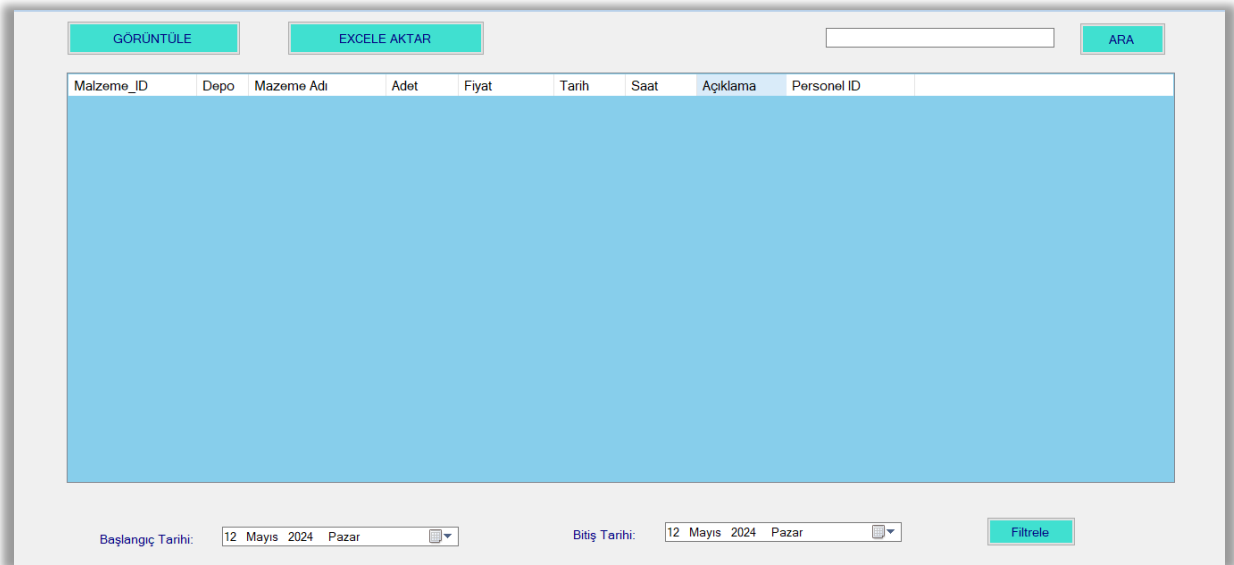
```

richTextBox1.Clear();
textBox3.Clear();
textBox1.Clear();
dateTimePicker1.ResetText();
listBox1.ClearSelected();
listBox2.ClearSelected();
listBox3.ClearSelected();
}

```

### 1.6.5. Stok Girdi Detay

Stok girdi ekranında “DETAY” butonuna basıldığında personelin yönlendirildiği ekrandır. Bu ekranda 4 adet buton, 1 adet listView, 1 adet textbox ve 2 adet dateTimePicker yer almaktadır. Personel bu ekranda “GÖRÜNTÜLE” butonuna bastığında depoya giriş kaydı yapılan tüm malzemeleri görüntüleyebilmekte ve “EXCELE AKTAR” butonuna basarak tüm dataları excel’e aktarabilmektedir. Görüntüleme işlemi yaparken tarih aralığı seçilerek filtreleme yapılabilir ayrıca search alanında tüm sütun başlıklarına göre arama yapılabilir.



Şekil 12. Stok Girdi Detay

#### Kod Parçacığı;

```

SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");

private void verilerigoster()
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Malzeme ", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {

```

```

        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["MZM_Id"].ToString();
        ekle.SubItems.Add(oku["Depo"].ToString());
        ekle.SubItems.Add(oku["MZMAđı"].ToString());
        ekle.SubItems.Add(oku["MZMAđet"].ToString());
        ekle.SubItems.Add(oku["Fiyat"].ToString());
        ekle.SubItems.Add(oku["Tarih"].ToString());
        ekle.SubItems.Add(oku["Saat"].ToString());
        ekle.SubItems.Add(oku["Açıklama"].ToString());
        ekle.SubItems.Add(oku["Id"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    verilerigoster();
}

private void button5_Click(object sender, EventArgs e)
{
    Microsoft.Office.Interop.Excel.Application xla = new
Microsoft.Office.Interop.Excel.Application();
    xla.Visible = true;
    Microsoft.Office.Interop.Excel.Workbook wb =
xla.Workbooks.Add(Microsoft.Office.Interop.Excel.XlSheetType.xlWorksheet);

    Microsoft.Office.Interop.Excel.Worksheet ws =
(Microsoft.Office.Interop.Excel.Worksheet)xla.ActiveSheet;
    ws.Cells[1, 1] = "ID";
    ws.Cells[1, 2] = "Depo";
    ws.Cells[1, 3] = "Mlzm Adı";
    ws.Cells[1, 4] = "Adet";
    ws.Cells[1, 5] = "Fiyat";
    ws.Cells[1, 6] = "Tarih";
    ws.Cells[1, 7] = "Saat";
    ws.Cells[1, 8] = "Açıklama";
    ws.Cells[1, 9] = "PersonelID";
    int i = 2;
    int j = 1;
    foreach (ListViewItem item in listView1.Items)
    {
        ws.Cells[i, j] = item.Text.ToString();
        foreach (ListViewItem ListViewItem subitem in item.SubItems)
        {
            ws.Cells[i, j] = subitem.Text.ToString();
            j++;
        }
        i = 1;
        i++;
    }
}

```

```

}
}
private void button1_Click_1(object sender, EventArgs e)
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Malzemee where MZMAAdi
like '%" + textBox1.Text + "%' or MZM_Id like '%" + textBox1.Text + "%' or Depo like '%" +
textBox1.Text + "%' or MZMAdet like '%" + textBox1.Text + "%' or Fiyat like '%" +
textBox1.Text + "%' or Tarih like '%" + textBox1.Text + "%' or Saat like '%" + textBox1.Text
+ "%' or Aciklama like '%" + textBox1.Text + "%' or Id like '%" + textBox1.Text + "%'",
baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["MZM_Id"].ToString();
        ekle.SubItems.Add(oku["Depo"].ToString());
        ekle.SubItems.Add(oku["MZMAAdi"].ToString());
        ekle.SubItems.Add(oku["MZMAdet"].ToString());
        ekle.SubItems.Add(oku["Fiyat"].ToString());
        ekle.SubItems.Add(oku["Tarih"].ToString());
        ekle.SubItems.Add(oku["Saat"].ToString());
        ekle.SubItems.Add(oku["Aciklama"].ToString());
        ekle.SubItems.Add(oku["Id"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}

private void button3_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();
    baglanti.Open();
    var minDate = dateTimePicker1.Value.ToString("yyyy-dd-MM");

    var maxDate = dateTimePicker2.Value.ToString("yyyy-dd-MM");
    SqlCommand komut = new SqlCommand($"select * from Malzemee where Tarih
between @minDate and @maxDate", baglanti);
    komut.Parameters.AddWithValue("minDate", minDate);
    komut.Parameters.AddWithValue("maxDate", maxDate);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["MZM_Id"].ToString();
        ekle.SubItems.Add(oku["Depo"].ToString());
        ekle.SubItems.Add(oku["MZMAAdi"].ToString());
        ekle.SubItems.Add(oku["MZMAdet"].ToString());
        ekle.SubItems.Add(oku["Fiyat"].ToString());
    }
}

```



```

ekle.SubItems.Add(oku["Tarih"].ToString());
ekle.SubItems.Add(oku["Saat"].ToString());
ekle.SubItems.Add(oku["Açıklama"].ToString());
ekle.SubItems.Add(oku["Id"].ToString());
listView1.Items.Add(ekle);
}
baglanti.Close();
}

```

## Stok Çıktı İşlemleri

Personel'in depodan çıkan ürünü kaydetmesi için kullandığı ekrandır. Bu ekranda 9 adet label, 3 adet combobox, 1 adet numericUpDown, 3 adet listbox, 2 adet textbox, 1 adet richTextBox, 1 adet menüstrip, 2 adet picturebox, 5 adet button ve 1 adet datetimepicker yer almaktadır. Personel bu ekrana yönlendiğinde Personel Id otomatik olarak dolu gelmektedir. Depo, Malzeme ve Çıkış Birimi combobox alanına veriler depoi malzeme ve çıkış tablolarından çekilerek getirilmiştir. Depo' dan birden fazla ürün çıktığı için personelin aynı bilgileri tekrar ve tekrar girmesini önlemek için malzeme adlarını, adetini ve çıktı fiyatını tutan 3 adet listbox ve listbox' a veri ekleyip silen ekle ve sil butonu kullanılmıştır. Personel ilgili alanları doldurup "KAYDET" butonuna bastığında kayıt işlemi gerçekleşip, kayıt işleminin gerçekleştiğine dair bilgilendirme mesajı çıkmaktadır. Personel alanlardaki verileri "TEMİZLE" butonuna basarak kolayca temizleyebilmektedir.

Şekil 13. Stok Çıktı İşlemleri

### Kod Parçacığı;

```

public string UserName { get; set; }
public Form8(string userName)
{
    InitializeComponent();
    UserName = userName;
}
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True;MultipleActiveResultSets=True");
private void verilerigoster()
{

```

```

        baglanti.Open();
        SqlCommand komut = new SqlCommand("select*from CIKIS", baglanti);
        SqlDataReader oku = komut.ExecuteReader();
        while (oku.Read())
        {
            ListViewItem ekle = new ListViewItem();
            ekle.Text = oku["C_ID"].ToString();
            ekle.SubItems.Add(oku["depo"].ToString());
            ekle.SubItems.Add(oku["sadi"].ToString());
            ekle.SubItems.Add(oku["sadet"].ToString());
            ekle.SubItems.Add(oku["Fiyat"].ToString());
            ekle.SubItems.Add(oku["s_tarih"].ToString());
            ekle.SubItems.Add(oku["s_saat"].ToString());
            ekle.SubItems.Add(oku["s_birim"].ToString());
            ekle.SubItems.Add(oku["s_aciklama"].ToString());

            ekle.SubItems.Add(oku["Id"].ToString());

        }
        baglanti.Close();
    }

private void gosterr()
{
    SqlCommand komut = new SqlCommand();
    komut.CommandText = "SELECT *FROM Birim";
    komut.Connection = baglanti;
    komut.CommandType = CommandType.Text;
    SqlDataReader dr;
    baglanti.Open();
    dr = komut.ExecuteReader();
    while (dr.Read())
    {
        comboBox2.Items.Add(dr["birimadi"]);
    }

    baglanti.Close();
}

private void gosterdepo()
{
    SqlCommand komut = new SqlCommand();
    komut.CommandText = "SELECT *FROM DEPO";
    komut.Connection = baglanti;
    komut.CommandType = CommandType.Text;
    SqlDataReader dr;
    baglanti.Open();
    dr = komut.ExecuteReader();

```



```

while (dr.Read())
{
    comboBox3.Items.Add(dr["depoadi"]);

    baglanti.Close();
}

private void Form8_Load(object sender, EventArgs e)
{
    SqlConnection baglanti = new SqlConnection();
    baglanti.ConnectionString = @"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True;MultipleActiveResultSets=True";
    SqlCommand komut = new SqlCommand();
    komut.CommandText = "SELECT*from URUN ";
    komut.Connection = baglanti;
    komut.CommandType = CommandType.Text;
    baglanti.Open();
    SqlCommand command2 = new SqlCommand("select * from personel where
EMail=@email", baglanti);
    command2.Parameters.AddWithValue("email", UserName);
    var data = command2.ExecuteReader();
    while (data.Read())
    {
        textBox3.Text = data["Id"].ToString();
    }

    SqlDataReader dr;

    dr = komut.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Items.Add(dr["urunadi"]);
    }
    baglanti.Close();

    goster();
    gosterdepo();
}

private void button2_Click(object sender, EventArgs e)
{
    Form13 lst2 = new Form13();
    lst2.Show();
}

private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < listBox1.Items.Count; i++)
    {

```

```

        baglanti.Open();
        SqlCommand komut1 = new SqlCommand("SELECT MZMAAdi,sum(MZMAAdet)
AS Toplam from Malzeme where Malzeme.MZMAAdi=@malzemeAdi GROUP BY
MZMAAdi", baglanti);
        komut1.Parameters.AddWithValue("@malzemeAdi", listBox1.Items[i].ToString());
        SqlDataReader oku1 = komut1.ExecuteReader();

        while (oku1.Read())
        {
            int toplam_girdi = 0;
            toplam_girdi = Convert.ToInt32(oku1["Toplam"]);
            string ad = oku1["MZMAAdi"].ToString();
            if (toplam_girdi < Convert.ToInt32(listBox2.Items[i]))
            {
                MessageBox.Show("Stokta olmayan ürünleri çıkış yapmaya çalışıyorsunuz, "
+ ad + " için yeterli miktarda stok mevcut değil, diğer eklenecekler kontrol ediliyor. ", "Hata",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
                break;
            }
            else
            {
                SqlCommand komut = new SqlCommand("insert into CIKIS(Depoo,
s_adi,c_adet, Fiyat, s_tarih,c_saat,c_birim,s_aciklama, Id,c_toplam)values(" +
comboBox3.Text.ToString() + "," + listBox1.Items[i].ToString() + "," + " +
listBox2.Items[i].ToString() + "," + listBox3.Items[i].ToString() + "," + " +
dateTimePicker1.Value.ToString() + "," + textBox2.Text.ToString() + "," + " +
comboBox2.Text.ToString() + "," + richTextBox1.Text.ToString() + "," + " +
textBox3.Text.ToString() + ")", baglanti);
                komut.ExecuteNonQuery();

                MessageBox.Show("STOK ÇIKTI BAŞARIYLA
GERÇEKLEŞTİRİLMİŞTİR");
            }
        }
        baglanti.Close();
    }
}

private void button3_Click(object sender, EventArgs e)
{
    listBox1.Items.Add(comboBox1.Text);
    listBox2.Items.Add(numericUpDown1.Text);
    listBox3.Items.Add(textBox1.Text);
}

private void button4_Click(object sender, EventArgs e)
{
    listBox1.Items.Remove(listBox1.SelectedItem);
    listBox2.Items.Remove(listBox2.SelectedItem);
    listBox3.Items.Remove(listBox3.SelectedItem);
}

private void button5_Click_1(object sender, EventArgs e)

```

```

{
    comboBox1.ResetText();
    comboBox2.ResetText();
    comboBox3.ResetText();
    numericUpDown1.ResetText();
    listBox1.Items.Clear();
    listBox2.Items.Clear();
    listBox3.Items.Clear();
    richTextBox1.Clear();
    textBox2.ResetText();
    textBox3.Clear();
    dateTimePicker1.ResetText();
    listBox1.ClearSelected();
    listBox2.ClearSelected();
    listBox3.ClearSelected();
}
}
}

```

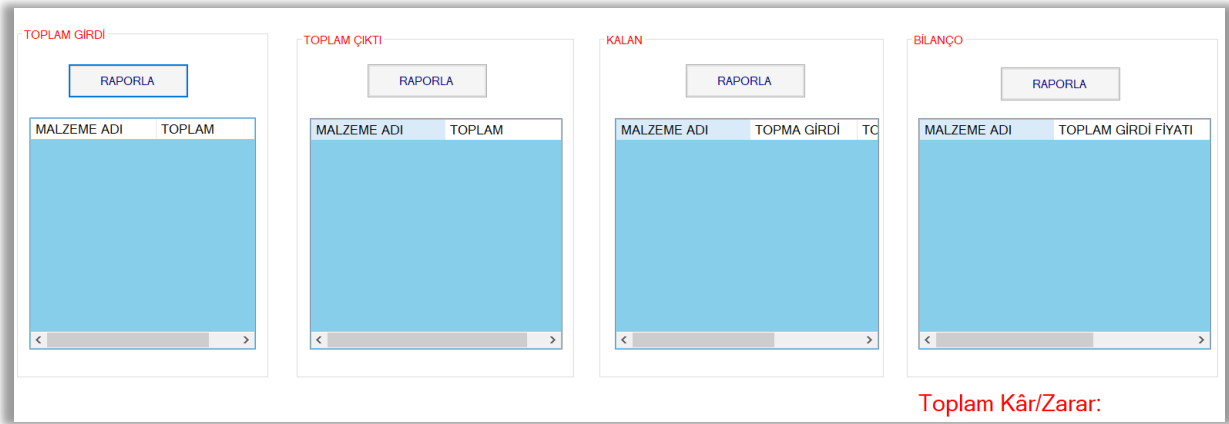
### 1.6.6. Stok Çıktı Detay

Stok çıktı ekranında “DETAY” butonuna basıldığında personelin yönlendirildiği ekrandır. Bu ekranda 4 adet buton, 1 adet listView, 1 adet textbox ve 2 adet dateTimePicker yer almaktadır. Personel bu ekranda “GÖRÜNTÜLE” butonuna bastığında depodan çıkış kaydı yapılan tüm malzemeleri görüntüleyebilmekte ve “EXCELE AKTAR” butonuna basarak tüm dataları excel’e aktarabilmektedir. Görüntüleme işlemi yaparken tarih aralığı seçilerek filtreleme yapılabilir ayrıca search alanında tüm sütun başlıklarına göre arama yapılabilir.

Şekil 14. Stok Çıktı Detay

### 1.6.7. Rapor

Personelin depoya giren ve çıkan toplam malzeme adetini, malzemelerin depodaki mevcut kalan bilgisini ve ilgili malzemelerden elde ettiği kar-zarar bilgisini görüntülediği ekrandır. Bu ekranda 4 adet groupbox, 4 adet listView, 4 adet buton ve 1 adet label kullanılmıştır. Personel “TOPLAM GİRDİ” groupbox’ındaki raporla butonuna bastığında depoya giren tüm malzemeler ve toplam adeti listelenir. Personel “TOPLAM ÇIKTI” groupbox’ındaki raporla butonuna bastığında depodan çıkan tüm malzemeler ve toplam çıkış adeti listelenir. Personel “KALAN” groupbox’ındaki raporla butonuna bastığında depoya girişi ve çıkışı olan tüm malzemelerin adeti ve depodaki kalan adet miktarı listelenir. Personel “BİLANÇO” groupbox’ındaki raporla butonuna bastığında giren malzemelerin toplam fiyatı, çıkan malzemelerin toplam fiyatı ve kar/zarar bilgisi görüntülenir.



Şekil 15. Rapor

#### Kod Parçacığı;

```
public Form12()
{
    InitializeComponent();
}

SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True;MultipleActiveResultSets=True");

private void gostergirdi()
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("SELECT MZMAAdi,sum(MZMAAdet) AS
Toplam from Malzemes GROUP BY MZMAAdi", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["MZMAAdi"].ToString();
        ekle.SubItems.Add(oku["Toplam"].ToString());
    }
}
```

```

listView1.Items.Add(ekle);
    }
    baglanti.Close();
}
private void gosterCikti()
{
    listView2.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("SELECT c_adi,sum(c_adet) AS c_toplam
from CIKIS GROUP BY c_adi", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["c_adi"].ToString();
        ekle.SubItems.Add(oku["c_toplam"].ToString());
        listView2.Items.Add(ekle);
    }
    baglanti.Close();
}
private void gosterrr()
{
    listView3.Items.Clear();
    baglanti.Open();
    SqlCommand komut1 = new SqlCommand("SELECT MZMAdi,sum(MZMAdet) AS
Toplam from Malzemee GROUP BY MZMAdi", baglanti);
    SqlDataReader oku1 = komut1.ExecuteReader();
    while (oku1.Read())
    {
        int toplam_girdi = 0;
        int toplam_cikti = 0;
        int kalan = 0;
        toplam_girdi = Convert.ToInt32(oku1["Toplam"]);
        string urun_adi = oku1["MZMAdi"].ToString();
        SqlCommand komut2 = new SqlCommand("SELECT c_adi,sum(c_adet) AS
c_toplam from CIKIS GROUP BY c_adi", baglanti);
        SqlDataReader oku2 = komut2.ExecuteReader();
        while (oku2.Read())
        {
            string urun_adi2 = oku2["c_adi"].ToString();
            if (urun_adi == urun_adi2 )
            {
                toplam_cikti = Convert.ToInt32(oku2["c_toplam"]);
                kalan = toplam_girdi - toplam_cikti;

                break;
            }
            continue;
        }
        if (toplam_cikti == toplam_girdi)

```

```

{
    kalan = 0;
}
else if (kalan == 0)
{
    kalan = Convert.ToInt32(oku1["Toplam"]);
}
ListViewItem ekle = new ListViewItem();
ekle.Text = oku1["MZMAdi"].ToString();
ekle.SubItems.Add(toplam_girdi.ToString());
ekle.SubItems.Add(toplam_cikti.ToString());
ekle.SubItems.Add(kalan.ToString());
listView3.Items.Add(ekle);
}
baglanti.Close();
}

private void button1_Click(object sender, EventArgs e)
{
    gosterGirdi();
}

private void button2_Click(object sender, EventArgs e)
{
    gosterCikti();
}

private void button3_Click(object sender, EventArgs e)
{
    gosterErr();
}

private void button4_Click(object sender, EventArgs e)
{
    listView4.Items.Clear();
    baglanti.Open();
    SqlCommand komut1 = new SqlCommand("SELECT MZMAdi,sum(MZMAdet) AS
Toplam, sum(Fiyat) AS Fiyat from Malzemee GROUP BY MZMAdi", baglanti);
    SqlDataReader oku1 = komut1.ExecuteReader();
    int toplamkar = 0;

    while (oku1.Read())
    {
        int toplam_girdi = 0;
        int toplam_cikti = 0;
        int kalan = 0;
        ListViewItem ekle = new ListViewItem();
        toplam_girdi = Convert.ToInt32(oku1["Fiyat"]);
        string urun_adi = oku1["MZMAdi"].ToString();
    }
}

```

```

SqlCommand komut2 = new SqlCommand("SELECT c_adi,sum(c_adet) AS c_toplam,
sum(Fiyat) AS Fiyat from CIKIS GROUP BY c_adi", baglanti);
SqlDataReader oku2 = komut2.ExecuteReader();
while (oku2.Read())
{
    string urun_adi2 = oku2["c_adi"].ToString();
    if (urun_adi == urun_adi2)
    {
        toplam_cikti = Convert.ToInt32(oku2["Fiyat"]);
        kalan = toplam_girdi - toplam_cikti;
        break;
    }
    continue;
}
ekle.SubItems.Add(toplam_girdi.ToString());
ekle.SubItems.Add(toplam_cikti.ToString());

if (toplam_girdi > toplam_cikti)
{
    toplamkar -= kalan;
    ekle.SubItems.Add(kalan.ToString().Insert(0, "-"));
}
if (toplam_cikti > toplam_girdi)
{
    toplamkar -= kalan;
    ekle.SubItems.Add(kalan.ToString().Trim('-'));
}
ekle.Text = oku1["MZMAdi"].ToString();
listView4.Items.Add(ekle);
label10.Text = "Toplam Kâr: " + toplamkar;
}
baglanti.Close();
}

```

### 1.6.8. Yönetici Girişi

Anasayfada yönetici girişi butonu tıklandığında bu form ekranı gelmektedir. Yönetici Girişi form ekranı için 4 adet label, 3 adet textbox, 2 adet buton, 1 adet checkbox, 1 adet link label, 3 adet picturebox ve 1 adet errorprovider kullanılmıştır. Yönetici kullanıcı adı veya e-mail, şifre ve random olarak değişen güvenlik kodunu doğru girdiği takdirde yönetici işlemleri form ekranına yönlendirilecektir. Yanlış girdiği takdirde ise hata mesajı ile karşılaşacaktır. Şifre 8 karakter belirlendiği içim 8 karakterden fazla veya az olan şifrelerde errorprovider sayesinde de hata icon' u yanıp sönecektir. Checkbox sayesinde de istendiğinde şifrenin gizlenmesi, istendiğin de ise şifrenin görülmesini sağlanmıştır. Çıkış butonuna basıldığında ise anasayfaya yönlendirme gerçekleştirilmiştir.



Şekil 16. Yönetici Girişi

#### Kod Parçacığı;

```
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    if (checkBox1.Checked)
    {
        textBox2.PasswordChar = '\0';
    }
    else
    {
        textBox2.PasswordChar = '*';
    }
}

private void button2_Click(object sender, EventArgs e)
{
    DialogResult cikis = new DialogResult();

    cikis = MessageBox.Show("ÇIKMAK İSTEDİĞİNİZDEN EMİN MİSİNİZ ?",
"UYARI",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question,
    MessageBoxDefaultButton.Button2);

    if (cikis == DialogResult.Yes)
    {
        Form1 frm1 = new Form1();
    }
}
```



```

frmM.Show();

    this.Close();
}
if (cikis == DialogResult.No)
{
    MessageBox.Show("PROGRAM KAPATILMADI");
}
}
}

private void Form3_Load(object sender, EventArgs e)
{
    int a, d, s, t;
    Random r = new Random();
    a = r.Next(1, 10);
    d = r.Next(1, 10);
    s = r.Next(1, 10);
    t = r.Next(1, 10);
    label4.Text = a.ToString() + d.ToString() + s.ToString() + t.ToString();
    textBox2.MaxLength = 8;
}

private void textBox2_TextChanged(object sender, EventArgs e)
{
    if (textBox2.Text.Length < 8 || textBox2.Text.Length > 8)
        errorProvider1.SetError(textBox2, "8 KARAKTER OLMALI!");
    else
    {
        errorProvider1.Clear();
    }
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    if (textBox1.Text == "")
        errorProvider2.SetError(textBox1, "BU ALAN BOŞ GEÇİLEMEZ!");
    else
    {
        errorProvider2.Clear();
    }
}

private void pictureBox3_Click(object sender, EventArgs e)
{
}

SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=erenproje2;Integrated Security=True");
private void button1_Click(object sender, EventArgs e)

```

```

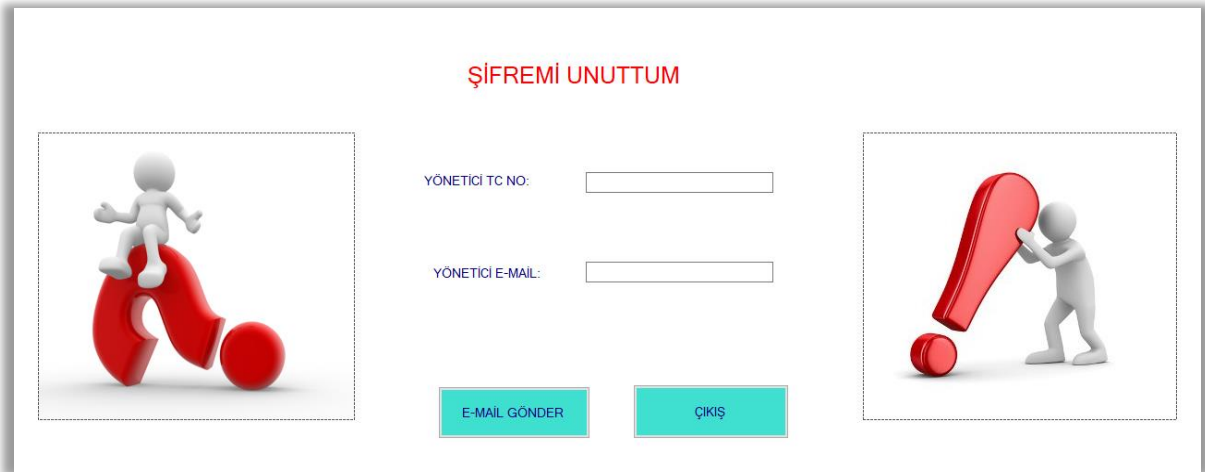
{
    try
    {
        baglanti.Open();
        string sql = "Select*From Yönetici Where Ad=@adi OR EMail=@maili AND Sifre=@sifresi";
        SqlParameter prm1 = new SqlParameter("adi", textBox1.Text.Trim());
        SqlParameter prm3 = new SqlParameter("maili", textBox1.Text.Trim());
        SqlParameter prm2 = new SqlParameter("sifresi", textBox2.Text.Trim());
        SqlCommand komut = new SqlCommand(sql, baglanti);
        komut.Parameters.Add(prm1);
        komut.Parameters.Add(prm3);
        komut.Parameters.Add(prm2);

        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(komut);
        da.Fill(dt);
        if (dt.Rows.Count > 0)
        {
            if (label4.Text == textBox3.Text)
            {
                Form5 frm55 = new Form5();
                frm55.Show();
            }
            else
            {
                MessageBox.Show("TEKRAR DENEYİNİZ...");
                int a, d, s, t;
                Random r = new Random();
                a = r.Next(1, 10);
                d = r.Next(1, 10);
                s = r.Next(1, 10);
                t = r.Next(1, 10);
                label4.Text = a.ToString() + d.ToString() + s.ToString() + t.ToString();
            }
        }
    }
}
catch (Exception)
{
    MessageBox.Show("YÖNETİCİ ADI/EMAIL VEYA ŞİFRE YANLIŞ!");
    baglanti.Close();
}

```

### 1.6.9. Şifremi Unuttum Yönetici Girişi

Yönetici girişindeki Şifremi unuttum linklabel' ı tıklandığı zaman Şifremi Unuttum Yönetici Girişi Form ekranı gelmektedir. Bu form ekranı için 3 adet label, 2 adet textbox, 2 adet button ve 2 adet picturebox kullanılmıştır. Yönetici doğru Tc No ve E-mail adresini girdiği takdirde ekrana bilgilendirme mesajı ve şifresi mail adresine gelmektedir. Yönetinin Tc No ve E-mail adresi eşleşmediği takdirde ise hata mesajı vermektedir.



Şekil 17. Şifremi Unuttum Yönetici Girişi

#### Kod Parçacığı;

```
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");
public bool Gonder(string konu, string icerik)
{
    MailMessage ePosta = new MailMessage();
    ePosta.From = new MailAddress(erencaan09@gmail.com);
    ePosta.To.Add(textBox2.Text);
    ePosta.Subject = konu;
    ePosta.Body = icerik;
    SmtpClient smtp = new SmtpClient();
    smtp.Credentials = new System.Net.NetworkCredential("erencaan09@gmail.com",
"erencaan58.");
    smtp.Port = 587;
    smtp.Host = "smtp.gmail.com";
    smtp.EnableSsl = true;
    object userState = ePosta;
    bool kontrol = true;
    try
    {
        smtp.SendAsync(ePosta, (object)ePosta);
    }
}
```

```

catch (SmtpException ex)
{
    kontrol = false;
    System.Windows.Forms.MessageBox.Show(ex.Message, "Mail Gönderme
    Hatasi");
}
return kontrol;

}
private void button1_Click(object sender, EventArgs e)
{
    baglanti.Open();

    SqlCommand com = new SqlCommand("Select * from Yönetici where TCNo=" +
    textBox1.Text.ToString() +
    "and EMail=" + textBox2.Text.ToString() + "", baglanti);
    SqlDataReader oku = com.ExecuteReader();
    string sifre;
    if (oku.Read())
    {
        sifre = oku["Sifre"].ToString();
        MessageBox.Show("Girmiş Oldunuz Bilgiler Uyuşuyor Şifreniz Mail adresinize
        yollanıyor");
        Gonder("Unutmuş Olduğunuz Şifreniz Ektedir", sifre);
        baglanti.Close();
    }
    else
    {
        MessageBox.Show("Bilgileriniz Uyuşmadı");
    }
    baglanti.Close();
}

private void button2_Click(object sender, EventArgs e)
{
    DialogResult cikis = new DialogResult();

    cikis = MessageBox.Show("ÇIKMAK İSTEDİĞİNİZDEN EMİN MİSİNİZ?",
    "UYARI",
    MessageBoxButtons.YesNo, MessageBoxIcon.Question,
    MessageBoxDefaultButton.Button2);
    if (cikis == DialogResult.Yes)
    {
        Form3 frm3 = new Form3();

        frm3.Show();

        this.Close();
    }
    if (cikis == DialogResult.No)
    {
        MessageBox.Show("ÇIKIŞ YAPILMADI.");
    }
}

```

## Yönetici İşlemleri

Yönetici doğru kullanıcı adı/ e-mail, şifre ve güvenlik kodunu girdiği takdirde bu form ekranına yönlendirilmektedir. Bu arayüz ekranında 12 adet label, 7 adet textbox, 3 adet combobox, 1 adet listview, 1 adet menustrip, 6 adet button, 1 adet combobox ve 2 adet picturebox bulunmaktadır. İl, İlçe ve Cinsiyet Tercihini elle manuel olarak değil de combobox ların içine yazarak seçilmesini ve seçildiği takdirde de labelde gözükmesi sağlanmıştır. Yönetici görüntüle butonuna bastığında veritabanının da kayıtlı olan tüm yöneticilere ait kişisel bilgileri listview sayesinde görebilmekte, yeni bir yöneticiyi ekle butonu ile ekleyebilmektedir. Herhangi bir yöneticinin ayrılması durumunda ise ayrılan yöneticiyi sil butonu sayesinde silebilmekte ve yöneticilerden herhangi birinin telefon numarası, soyadı vb. bilgilerinin değişmesi doğrultusunda güncelle butonu ile güncelleme işlemini gerçekleştirebilmektedir.

Y_Id	Ad	Soyad	TCNo	EMail	Sifre	İL	İLCE	Cinsiyet	Telefon
------	----	-------	------	-------	-------	----	------	----------	---------

Şekil 18. Yönetici İşlemleri

### Kod Parçacığı;

```
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE2;Integrated Security=True");
private void verilerigoster()
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Yönetici", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Y_Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
    }
}
```

```

        ekle.SubItems.Add(oku["TCNo"].ToString());
        ekle.SubItems.Add(oku["EMail"].ToString());
        ekle.SubItems.Add(oku["Sifre"].ToString());
        ekle.SubItems.Add(oku["IL"].ToString());
        ekle.SubItems.Add(oku["ILCE"].ToString());
        ekle.SubItems.Add(oku["Cinsiyet"].ToString());
        ekle.SubItems.Add(oku["Telefon"].ToString());
        ekle.SubItems.Add(oku["Yresim"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    label1.Text = "Seçiminiz : " + comboBox1.SelectedItem;
}

private void comboBox1_SelectedIndexChanged_2(object sender, EventArgs e)
{
    label1.Text = "Seçiminiz : " + comboBox1.SelectedItem;
}
private void comboBox2_SelectedIndexChanged_1(object sender, EventArgs e)
{
    label2.Text = "Seçiminiz : " + comboBox2.SelectedItem;
}

private void comboBox3_SelectedIndexChanged_1(object sender, EventArgs e)
{
    label3.Text = "Seçiminiz : " + comboBox3.SelectedItem;
}

private void button1_Click(object sender, EventArgs e)
{
    Form14 lst14 = new Form14();
    lst14.Show();
}

private void button2_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("insert into
Yönetici(Ad,Soyad,TCNo,EMail,Sifre,IL,ILCE,Cinsiyet,Telefon,yresim)values('" +
textBox1.Text.ToString() + "','" + textBox2.Text.ToString() + "','" + textBox3.Text.ToString()
+ "','" + textBox4.Text.ToString() + "','" + textBox5.Text.ToString() + "','" +
comboBox1.Text.ToString() + "','" + comboBox2.Text.ToString() + "','" +
comboBox3.Text.ToString() + "','" + textBox6.Text.ToString() + "','" +
textBox7.Text.ToString() + "')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
}

```

```

MessageBox.Show("KAYIT EKLENMİŞTİR");
verilerigoster();
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
comboBox1.Text = "";
comboBox2.Text = "";
comboBox3.Text = "";
textBox6.Clear();
textBox7.Clear();
pictureBox2.Text = "";

}

private void button4_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("update Yönetici set Ad=" +
textBox1.Text.ToString() + ", Soyad=" + textBox2.Text.ToString() + ", TCNo=" +
textBox3.Text.ToString() + ", EMail=" + textBox4.Text.ToString() + ", Sifre=" +
textBox5.Text.ToString() + ", IL=" + comboBox1.Text.ToString() + ", ILCE=" +
comboBox2.Text.ToString() + ", Cinsiyet=" + comboBox3.Text.ToString() + ", Telefon="
+ textBox6.Text.ToString() + ", Yresim=" + textBox7.Text.ToString() + " where Y_Id=" +
Y_Id + "", baglanti);

    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT GÜNCELLENMİŞTİR");
    verilerigoster();
}
int Y_Id = 0;

private void button3_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Delete from Yönetici where Y_Id=("
+ Y_Id + ")", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT SİLİNİMİŞTİR");
    verilerigoster();
}

private void listView1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    Y_Id = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[1].Text;
    textBox2.Text = listView1.SelectedItems[0].SubItems[2].Text;
    textBox3.Text = listView1.SelectedItems[0].SubItems[3].Text;
    textBox4.Text = listView1.SelectedItems[0].SubItems[4].Text;
    textBox5.Text = listView1.SelectedItems[0].SubItems[5].Text;

```



```

comboBox1.Text = listView1.SelectedItems[0].SubItems[6].Text;
comboBox2.Text = listView1.SelectedItems[0].SubItems[7].Text;
comboBox3.Text = listView1.SelectedItems[0].SubItems[8].Text;
textBox6.Text = listView1.SelectedItems[0].SubItems[9].Text;
textBox7.Text = listView1.SelectedItems[0].SubItems[10].Text;
textBox1.Clear();
textBox2.Clear();
textBox3.Clear();
textBox4.Clear();
textBox5.Clear();
comboBox1.Text = "";
comboBox2.Text = "";
comboBox3.Text = "";
textBox6.Clear();
textBox7.Clear();
pictureBox2.Text = "";

}

private void personelİslemleriToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form6 frm6 = new Form6();
    frm6.Show();
}

private void button5_Click(object sender, EventArgs e)
{
    OpenFileDialog dosya = new OpenFileDialog();
    dosya.Filter = "Resim Dosyası |*.jpg;*.nef;*.png | Tüm Dosyalar |*. *";
    dosya.ShowDialog();
    string dosyayolu = dosya.FileName;
    textBox7.Text = dosyayolu;
    pictureBox2.ImageLocation = dosyayolu;
}

private void gİRİSToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 frm4 = new Form4();
    frm4.Show();
}

private void button6_Click(object sender, EventArgs e)
{
    verilerigoster();
}
}

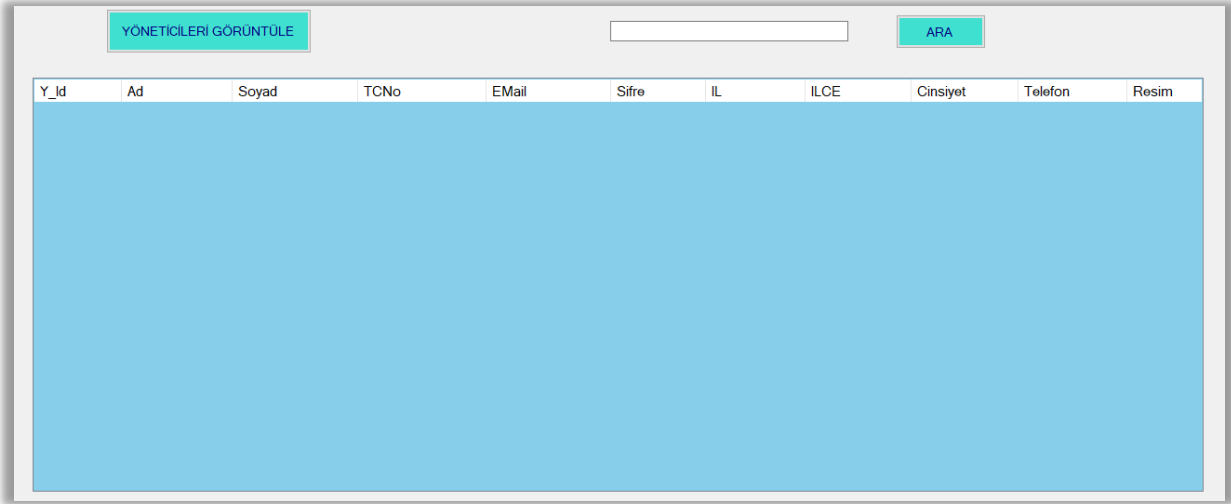
```



### 1.6.10. Yönetici İşlemleri Detay

Yönetici İşlemleri ekranında “DETAY” butonuna basıldığında yöneticinin yönlendirildiği ekrandır. Bu ekranda 2 adet buton, 1 adet listView ve 1 adet textbox yer almaktadır.

Yönetici “GÖRÜNTÜLE” butonuna bastığında veritabanında kayıtlı olan tüm yöneticilere ait kişisel bilgileri görüntüleyebilmekte ve search alanında tüm sütun başlıklarına göre arama yapılabilir.



Şekil 19. Yönetici İşlemleri Detay

#### Kod Parçası;

```
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE2;Integrated Security=True");
private void verilerigoster()
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Yönetici", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Y_Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TCNo"].ToString());
        ekle.SubItems.Add(oku["EMail"].ToString());
        ekle.SubItems.Add(oku["Sifre"].ToString());
        ekle.SubItems.Add(oku["IL"].ToString());
    }
}
```

```

        ekle.SubItems.Add(oku["ILCE"].ToString());
        ekle.SubItems.Add(oku["Cinsiyet"].ToString());
        ekle.SubItems.Add(oku["Telefon"].ToString());
        ekle.SubItems.Add(oku["Yresim"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}
private void button1_Click(object sender, EventArgs e)
{
    verilerigoster();
}
int Y_Id = 0;

private void button2_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Yönetici where TCNo
like'" + textBox1.Text + "%' or Y_Id like'" + textBox1.Text + "%' or Ad like'" +
textBox1.Text + "%' or Soyad like'" + textBox1.Text + "%' or EMail like'" +
textBox1.Text + "%' or Sifre like'" + textBox1.Text + "%' or IL like'" + textBox1.Text +
"%' or ILCE like'" + textBox1.Text + "%' or Cinsiyet like'" + textBox1.Text + "%' or
Telefon like'" + textBox1.Text + "%' ", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Y_Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TCNo"].ToString());
        ekle.SubItems.Add(oku["EMail"].ToString());
        ekle.SubItems.Add(oku["Sifre"].ToString());
        ekle.SubItems.Add(oku["IL"].ToString());
        ekle.SubItems.Add(oku["Cinsiyet"].ToString());
        ekle.SubItems.Add(oku["Telefon"].ToString());
        ekle.SubItems.Add(oku["Yresim"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}

```

## 1.6.11. Personel İşlemleri

Personelin kişisel bilgileri ile işlemleri yönetici yapmaktadır. Yönetici menustripte bulunan personel işlemlerine tıkladığı takdirde bu form ekranına geçiş yapmaktadır. Yönetici işlemleriyle aynı şekilde olup sadece farklı database ve tablodan verileri çekme işlemi gerçekleştirilmiştir. Ekranda 10 adet label, 8 adet textbox, 3 adet combobox, 1 adet listview, 1 adet menustrip, 6 adet button, 1 adet combobox ve 1 adet picturebox bulunmaktadır. İl, İlçe ve Cinsiyet Tercihini elle manuel olarak değil de combobox ların içine yazarak seçilmesini ve seçildiği takdirde de labelde gözükmesi sağlanmıştır. Yönetici görüntüle butonuna bastığında veritabanının da kayıtlı olan tüm personellere ait kişisel bilgileri görebilmekte, yeni bir personeli ekle butonu ile ekleyebilmektedir. Herhangi bir personelin ayrılması durumunda ise ayrılan personeli sil butonu sayesinde silebilmekte ve personellerden herhangi birinin telefon numarası, soyadı vb. bilgilerinin değişmesi doğrultusunda güncelle butonu ile güncelleme işlemini gerçekleştirebilmektedir. Ayrıca search alanında tüm sütun başlıklarına göre arama yapılabilmektedir

Id	Ad	Soyad	TCNo	EMail	Sifre	IL	ILCE	Cinsiyet	Telefon	Resim
----	----	-------	------	-------	-------	----	------	----------	---------	-------

Şekil 20. Personel İşlemleri

### Kod Parçacığı;

```
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");
private void verilerigoster()
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Personel", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
    }
}
```

```

        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TCNo"].ToString());
        ekle.SubItems.Add(oku["EMail"].ToString());
        ekle.SubItems.Add(oku["Sifre"].ToString());
        ekle.SubItems.Add(oku["IL"].ToString());
        ekle.SubItems.Add(oku["ILCE"].ToString());
        ekle.SubItems.Add(oku["Cinsiyet"].ToString());
        ekle.SubItems.Add(oku["Telefon"].ToString());
        ekle.SubItems.Add(oku["presim"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}
private void button1_Click(object sender, EventArgs e)
{
    verilerigoster();
}

private void button2_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("insert into
Personel(Ad,Soyad,TCNo,EMail,Sifre,IL,ILCE,Cinsiyet,Telefon,presim)values('" +
textBox1.Text.ToString() + "','" + textBox2.Text.ToString() + "','" + textBox3.Text.ToString()
+ "','" + textBox4.Text.ToString() + "','" + textBox5.Text.ToString() + "','" +
comboBox3.Text.ToString() + "','" + comboBox2.Text.ToString() + "','" +
comboBox1.Text.ToString() + "','" + textBox9.Text.ToString() + "','" +
textBox6.Text.ToString() + "')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT EKLENMİŞTİR");
    verilerigoster();
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    textBox5.Clear();
    comboBox3.Text = "";
    comboBox2.Text = "";
    comboBox1.Text = "";
    textBox9.Clear();
    textBox6.Clear();
}
int Id = 0;
private void button3_Click(object sender, EventArgs e)
{
    baglanti.Open();

    SqlCommand komut = new SqlCommand("Delete from Personel where Id=(" + Id +
    ")", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT SİLİNİMİŞTİR");
    verilerigoster();
}

```

```

private void listView1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    Id = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[1].Text;
    textBox2.Text = listView1.SelectedItems[0].SubItems[2].Text;
    textBox3.Text = listView1.SelectedItems[0].SubItems[3].Text;
    textBox4.Text = listView1.SelectedItems[0].SubItems[4].Text;
    textBox5.Text = listView1.SelectedItems[0].SubItems[5].Text;
    comboBox3.Text = listView1.SelectedItems[0].SubItems[6].Text;
    comboBox2.Text = listView1.SelectedItems[0].SubItems[7].Text;
    comboBox1.Text = listView1.SelectedItems[0].SubItems[8].Text;
    textBox9.Text = listView1.SelectedItems[0].SubItems[9].Text;
    textBox6.Text = listView1.SelectedItems[0].SubItems[10].Text;
    textBox1.Clear();
    textBox2.Clear();
    textBox3.Clear();
    textBox4.Clear();
    textBox5.Clear();
    comboBox3.Text = "";
    comboBox2.Text = "";
    comboBox1.Text = "";
    textBox9.Clear();
    textBox6.Clear();
}

private void button4_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("update Personel set Ad=" +
    textBox1.Text.ToString() + " ,Soyad=" + textBox2.Text.ToString() + " ,TCNo=" +
    textBox3.Text.ToString() + " ,EMail=" + textBox4.Text.ToString() + " ,Sifre=" +
    textBox5.Text.ToString() + " ,IL=" + comboBox3.Text.ToString() + " ,ILCE=" +
    comboBox2.Text.ToString() + " ,Cinsiyet=" + comboBox1.Text.ToString() + " ,Telefon="
    + textBox9.Text.ToString() + " ,presim=" + textBox6.Text.ToString() + " where Id=" + Id +
    "" , baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT GÜNCELLENMİŞTİR");
    verilerigoster();
}

private void button5_Click(object sender, EventArgs e)
{
    OpenFileDialog dosya = new OpenFileDialog();
    dosya.Filter = "Resim Dosyası |*.jpg;*.nef;*.png | Tüm Dosyalar |*. *";
    dosya.ShowDialog();
    string dosyayolu = dosya.FileName;
    textBox6.Text = dosyayolu;
    pictureBox1.ImageLocation = dosyayolu;
}

```



```

private void button7_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Personel where TCNo like
'" + textBox7.Text + "%' or Id like'" + textBox7.Text + "%' or Ad like'" +
textBox7.Text + "%' or Soyad like'" + textBox7.Text + "%' or EMail like'" +
textBox7.Text + "%' or Sifre like'" + textBox7.Text + "%' or IL like'" + textBox7.Text +
"%' or ILCE like'" + textBox7.Text + "%' or Cinsiyet like'" + textBox7.Text + "%' or
Telefon like'" + textBox7.Text + "%'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["Id"].ToString();
        ekle.SubItems.Add(oku["Ad"].ToString());
        ekle.SubItems.Add(oku["Soyad"].ToString());
        ekle.SubItems.Add(oku["TCNo"].ToString());
        ekle.SubItems.Add(oku["EMail"].ToString());
        ekle.SubItems.Add(oku["Sifre"].ToString());
        ekle.SubItems.Add(oku["IL"].ToString());
        ekle.SubItems.Add(oku["ILCE"].ToString());
        ekle.SubItems.Add(oku["Cinsiyet"].ToString());
        ekle.SubItems.Add(oku["Telefon"].ToString());
        ekle.SubItems.Add(oku["presim"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}

```

## 2.6.14. Ürün/Birim/Depo İşlemleri

Yönetici menüstripte bulunan Ürün/Birim/Depo işlemlerine tıkladığı takdirde bu form ekranına geçiş yapmaktadır. Bu arayüz ekranı için 3 adet groupbox, 3 adet label, 3 adet textbox, 3 adet listview ve 12 adet button yer almaktadır. 3 işlem de aynı form ekranı üzerinde gerçekleştirildiği için groupboxlar yardımı ile işlemler birbirinden ayırmıştır. Bir işlem gerçekleşirken diğerinin gerçekleşmemesi durumu olabileceği için 3 adet listview kullanarak her işlem için veritabanı bilgisi ayrı ayrı gerçekleştirilmiştir. Yönetici görüntüle butonuna bastığında hangi ürünlerin olduğunu, hangi çıkış birimlerini olduğunu ve hangi depoların olduğunu görebilmektedir. Kaydet butonu ile yeni bir ürün, birim ve depo ekleyebilir ve aynı şekil de silebilmektedir. Yönetici ürün adı, birim ve depo girip Ara butonuna basarak ilgili aramayı yapabilmektedir. Yönetici herhangi bir ürün veya depo eklediği takdirde personel stok girdi işlemi yaparken depo seçtiği combobox a eklediği bu yeni ürün veya depo gelmektedir. Yönetici herhangi bir birim eklediği takdirde ise personel stok çıktı işlemi yaparken çıkış birimi seçtiği comobox' a yeni eklenen çıkış birimi gelmektedir. Aynı şekilde de silme işlemi yapıldığında comboboxlar'ın içerisinde silinmektedir.



Şekil 21. Ürün/Birim/Depo İşlemleri

#### Kod Parçacığı;

```
SqlConnection baglanti = new SqlConnection(@"Data Source=Z1R0F;Initial
Catalog=ERENPROJE;Integrated Security=True");
private void verilerigoster()
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from URUN", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["urun_ID"].ToString();
        ekle.SubItems.Add(oku["urunadi"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}
private void goster()
{
    listView2.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from Birim", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
```

```

        ekle.Text = oku["birim_ID"].ToString();
        ekle.SubItems.Add(oku["birimadi"].ToString());
        listView2.Items.Add(ekle);
    }
    baglanti.Close();
}
private void depogoster()
{
    listView3.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from DEPO", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["depo_ID"].ToString();
        ekle.SubItems.Add(oku["depoadi"].ToString());
        listView3.Items.Add(ekle);
    }
    baglanti.Close();
}
private void button2_Click(object sender, EventArgs e)
{
    verilerigoster();
}

private void button1_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("insert into URUN(urunadi)values('" +
textBox1.Text.ToString() + "')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT EKLENMİŞTİR");
    verilerigoster();
    textBox1.Clear();
}

int urun_ID = 0;
private void button3_Click(object sender, EventArgs e)
{
    if (listView1.SelectedItems.Count > 0)
    {
        ListViewItem item = listView1.SelectedItems[0]
    }

    baglanti.Open();

```



```

SqlCommand komut = new SqlCommand("Delete from URUN where urun_ID=(" + urun_ID
+ ")", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT SİLİNMIŞTİR");
    verilerigoster();
    textBox1.Clear();
}

private void listView1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    urun_ID = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[1].Text;
}

private void button5_Click(object sender, EventArgs e)
{
    goster();
}

private void button6_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("insert into Birim(birimadi)values('" +
textBox2.Text.ToString() + "')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT EKLENMIŞTİR");
    goster();
    textBox2.Clear();
}

}

int Birim_ID = 0;
private void button4_Click_1(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Delete from Birim where Birim_ID=(" +
Birim_ID + ")", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT SİLİNMIŞTİR");
    goster();
    textBox2.Clear();
}

private void listView2_MouseDoubleClick(object sender, MouseEventArgs e)
{
    Birim_ID = int.Parse(listView2.SelectedItems[0].SubItems[0].Text);
    textBox2.Text = listView2.SelectedItems[0].SubItems[1].Text;
}

private void button8_Click(object sender, EventArgs e)
{
    depogoster();
}

```

```

private void button9_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("insert into DEPO(depoadi)values('" +
textBox3.Text.ToString() + "')", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT EKLENMİŞTİR");

    depogoster();
    textBox3.Clear();
}
int depo_ID = 0;
private void button7_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand komut = new SqlCommand("Delete from DEPO where depo_ID =(" +
depo_ID + ")", baglanti);
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("KAYIT SİLİNİMİŞTİR");
    depogoster();
    textBox3.Clear();
}

private void listView3_MouseDoubleClick(object sender, MouseEventArgs e)
{
    depo_ID = int.Parse(listView3.SelectedItems[0].SubItems[0].Text);
    textBox3.Text = listView3.SelectedItems[0].SubItems[1].Text;
}

private void button10_Click(object sender, EventArgs e)
{
    listView1.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from URUN where urunadi like '%" +
textBox1.Text + "%' or urun_ID like '%" + textBox1.Text + "%'", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["urun_ID"].ToString();
        ekle.SubItems.Add(oku["urunadi"].ToString());
        listView1.Items.Add(ekle);
    }
    baglanti.Close();
}

private void button11_Click(object sender, EventArgs e)
{
    listView2.Items.Clear();
    baglanti.Open();

```

```

SqlCommand komut = new SqlCommand("select*from Birim where birimadi like '%" +
textBox2.Text + "%' or Birim_ID like '%" + textBox2.Text + "%' ", baglanti);
SqlDataReader oku = komut.ExecuteReader();
while (oku.Read())
{
    ListViewItem ekle = new ListViewItem();
    ekle.Text = oku["Birim_ID"].ToString();
    ekle.SubItems.Add(oku["birimadi"].ToString());

    listView2.Items.Add(ekle);
}
baglanti.Close();
}

private void button12_Click(object sender, EventArgs e)
{
    listView3.Items.Clear();
    baglanti.Open();
    SqlCommand komut = new SqlCommand("select*from DEPO where depoadi like '%" +
+ textBox3.Text + "%' or depo_ID like '%" + textBox3.Text + "%' ", baglanti);
    SqlDataReader oku = komut.ExecuteReader();
    while (oku.Read())
    {
        ListViewItem ekle = new ListViewItem();
        ekle.Text = oku["depo_ID"].ToString();
        ekle.SubItems.Add(oku["depoadi"].ToString());
        listView3.Items.Add(ekle);
    }
    baglanti.Close();
}

private void listView1_MouseClick(object sender, MouseEventArgs e)
{
    urun_ID = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[1].Text;
}

private void listView2_MouseClick(object sender, MouseEventArgs e)
{
    urun_ID = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[1].Text;
}

private void listView3_MouseClick(object sender, MouseEventArgs e)
{
    urun_ID = int.Parse(listView1.SelectedItems[0].SubItems[0].Text);
    textBox1.Text = listView1.SelectedItems[0].SubItems[1].Text;
}
}

```