

KOCAELİ ÜNİVERSİTESİ
KOCAELİ MESLEK YÜKSEKOKULU

BİLGİSAYAR TEKNOJİLERİ BÖLÜMÜ
BİLGİSAYAR PROGRAMCILIĞI PROGRAMI

ÖN LİSANS TEZİ

MARKET OTOMASYONU

ENGİN YENİCE

KOCAELİ 2017

ÖNSÖZ VE TEŞEKKÜR

Yardımlarını eksik etmeyen değerli hocalarıma teşekkür ederim. Ayrıca hayatım boyunca beni destekleyen aileme ve hayallerimi gerçekleştirmemde katkısı olan herkese sonsuz minnet duygularımı sunarım.

Aralık - 2017

Engin YENİCE

ÖZET

Süpermarket, marketlerde ve bakkallarda iş kolaylığı sağlamak amacıyla geliştirilmiştir. Yapılan projede C# dili ve ACCESS veri tabanı kullanılmıştır. Market otomasyonunda birçok alanda geliştirilmeler yapılmıştır ve bu geliştirmeler:

1. Güvenli giriş
2. Müşteri kontrolü
3. Satış kontrolü
4. Ürün (ekleme, düzenleme ve silme) işlemleri

Ürün işlemleri yaparak ürünlerin daha düzenli ve kontrol edilebilir hale getirilmesi planlanmıştır. Satış kontrolü ile yapılan kazançların kaydı tutulması planlanmıştır. Müşteri kontrolü eklenerek firmaya gelen müşterilerin bilgilerinin saklandığı bir veri tablosu geliştirilerek firmanın müşteriler ile iletişim sağlanması hedeflenmiştir. Güvenli giriş sistemleri ile sisteme güvenilir ve yetkiye göre işlemlerin yapılabileceği bir basit arayüz tasarlanarak yapılacak işlerin daha hızlı gerçekleştirilmesi için çalışmalar yapılmıştır.

Anahtar Kelimeler: ACCESS, C# , Kolay ara yüz, Market Otomasyonu, RecX

ABSTRACT

It was developed to provide business convenience to supermarkets, markets and grocery stores. C# language and ACCESS database were used in the project. Many field improvements have been made in the automation of the market and these developments are:

1. Secure login
2. Customer control
3. Sales check
4. Product (add, edit and delete) operations

It is planned to make the products more regular and controllable by carrying out product operations. It is planned to record the earnings made by sales control. By adding a customer control, a database storing the information of the customers coming to the company was developed and it was aimed to communicate with the customer of the company. Safe login systems have been designed to design a simple interface that can be trusted and trusted by the system to perform tasks faster.

Key words: ACCESS, C #, Easy interface, Market automation, RecX

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
ÖZET.....	ii
ABSTRACT.....	iii
GİRİŞ	1
GENEL BİLGİLER	3
C# Nedir?	3
Microsoft Access Nedir?	3
PROGRAMIN TÜM KODLARI.....	4
1. Giriş Ekranı (Login)	4
2. Ana Yüz Ekranı (Home).....	6
3. Ara Yüz (Menu)	8
4. Ürün Giriş (Perakendeurungiris)	11
5. Firma / Müşteri Girişi (Cari_Firmagir)	20
6. Kategori İşlemleri (Kategoriislem).....	22
7. Satış Bilgisi(Satisbilgi).....	25
8. Müşteri Listesi (Musterilistesi)	31
9. Şirket Bilgileri Girişi (Sbilgi).....	38
10. Hesap Oluştur(Hesapolustur).....	41
11. Hesap Yetkilendirme (Hesapyetkilendir)	43
12. Hesap Şifre Değiştir (Yonetimhesap)	45
13. Hesap Silme(Hesapsil).....	47
14. Duyuru İşlemleri (Duyuruislem)	49
15. Satış İşlemi (Perakendesatisgirisi)	51
16. Perakende Satış Ürün Seçimi(Perakendesatissecimi).....	62
17. Satış İşlemi Müşteri Seçimi (Perakendesatismusteri).....	65
KAYNAKLAR	68
EKLER.....	69
ÖZGEÇMİŞ	77

GİRİŞ

Proje marketlerdeki iş kolaylığının ve kontrolünün sağlanmasıdır. Projenin konusu “market otomasyonu programının geliştirilmesi”dir. Projede kullanılan programlama dili olarak C# kullanılmıştır. Veri tabanı olarak ACCESS tercih edilmiştir.

İş kolaylığının sağlanması için yapılan çalışmalar:

- 1) Stok kontrolü
- 2) Müşteri kontrolü
- 3) Kasa kontrolü

İş kontrolünün sağlanması için yapılan çalışmalar:

- 1) Stok raporlarının oluşturulması
- 2) Müşteri raporlarının oluşturulması
- 3) Kasa raporlarının oluşturulması
- 4) Fiş yazdırılarak anlık satış raporunun oluşturulması

Stok kontrolü alanında yapılan geliştirmelerde ürünlerin kategorilere ayrılması sonucunda daha kontrollü bir yapı oluşturulmuştur. Ürünleri sisteme ekleme, düzenleme ve silme işlemleri gerçekleştirilebilir. Anlık olarak ürün listesinin raporları alınabilir.

Müşteri kontrolü alanında yapılan geliştirmelerde, müşterilerin sistemde kayıtları tutularak müşteriler hakkında bilgiler alınabilir. Anlık olarak tüm müşterilerin listelendiği bir raporlama sistemi oluşturulabilir.

Kasa kontrolü alanında yapılan geliştirmelerde satış işlemi gerçekleştirildikten sonra veriler kasaya kayıt edilmektedir. Bu kayıt işlemlerinin sonucunda kasadan

günlük ve aylık satış verilerine ulaşılabilir. Tüm satış verilerinin bulunduğu bir rapor alınabilir.

Bu geliştirmeler sonrasında proje tamamlandı ve adı “RecX Market Otomasyonu” olarak belirlendi.

GENEL BİLGİLER

C# Nedir?

1950’li yıllardan başlayıp günümüze kadar uzanan yazılım sektöründe her geçen gün yeni şeyler duymak mümkün. Çünkü insanlık tarih boyunca işlerini kolaylaştıracak ürünler üretmeye meyilli olmuştur. Programlama ise 20. ve 21. yüzyılda insanların hayatını kolaylaştıran ürünler üretmek için kullanılan bir araç olmuştur.

Günümüze kadar pek çok programlama dilleri geliştirilmiştir. Bunlar kullanılacak platformlara göre ya da dil yapısına göre farklı alanlarda kullanılır. Tüm dillerden arasında özellikle nesnel programlama alanında iki programlama dili insanlık için oldukça önemlidir. Bu dillerin ilki ortak platform olarak çalıştırılabilen Java ikincisi ise.net kütüphanesi ile bütünleşmiş edilerek tüm dillerle ortak platformda programlanabilir ve kolay kodlama yapısı ile C# (CSharp) programlama dilidir.

Microsoft Access Nedir?

Access, İlişkisel Veri Tabanı Yönetim Sistemi ile çalışan bir veri tabanı oluşturma programıdır. İlişkisel Veri Tabanı Yönetim Sistemi sisteminde bir veri tabanı dosyasında birden fazla tablo oluşturulabilir ve bu tablolar arasında birbirleriyle ilişki kurulabilir. Kurulan ilişkiler sayesinde farklı tablolardaki veriler sanki aynı tablodaymış gibi kullanılabilir.

Microsoft Access bir İlişkisel Veri Tabanı Yönetim Sistemi uygulamasıdır. Bir veri tabanını oluşturmak ve kullanmak Access ile diğer veri tabanı uygulamalarına göre çok daha kolaydır. Bunun nedeni Access’in, Windows ortamının Grafiksel Kullanıcı Arabiriminin sağladığı avantajların tümünden yararlanma imkânı vermesidir. Grafiksel Kullanıcı Arabirimi, karmaşık komut dizilerini öğrenmeyi gerektirmeden, ekran üzerindeki nesneler ve simgeler yardımıyla, fare desteğinden de yararlanarak kullanıcının çalışmasına olanak verir.

PROGRAMIN TÜM KODLARI

1. Giriş Ekranı (Login)

Programın giriş ekranında kullanıcı adı şifre sistemi bulunmaktadır bu sistem ile veri tabanından kullanıcı olup olmadığı bilgisi teyit edilir ve varsa eğer kullanıcının kullanıcı adı, adı soyadı, yetkisi alınarak ara yüz ekranına gönderilir.

Kodlar :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class Login : Form
    {
        public Login()
        {
            InitializeComponent();
        }
        OleDbConnection con;
        OleDbCommand cmd;
        OleDbDataReader dr;
        private void BtnGiris_Click(object sender, EventArgs e)
        {
            string ad = TxtID.Text;
            string sifre = TxtPass.Text;
            string ym_ad = "", ym_soyad = "";
            string ym_yetki = "";
            try
            {
                con = new OleDbConnection(@"Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
                con.Open();
                cmd = new OleDbCommand();
                cmd.Connection = con;
```

```

        cmd.CommandText = "SELECT * FROM yonetim where ym_kadi='" +
TxtID.Text + "' AND ym_ksif='" + TxtPass.Text + "'";
    }
    catch (Exception gata)
    {

        MessageBox.Show("Bağlanamadı"+ gata);

    }
    dr = cmd.ExecuteReader();
    if (dr.Read())
    {
        ym_ad = dr["ym_ad"].ToString();
        ym_soyad = dr["ym_soyad"].ToString();
        ym_yetki = dr["ym_yetki"].ToString();
        Home home = new Home();
        home.label1.Text = TxtID.Text;
        home.label3.Text = ym_ad + " " + ym_soyad;
        home.label5.Text = ym_yetki;

        home.Show();
        this.Hide();
    }
    else
    {
        label3.Text = "Kullanıcı adı ya da şifre hatalı !";
    }
    con.Close();
}

private void Login_Load(object sender, EventArgs e)
{
    this.Size = new Size(310, 360);
    this.MaximumSize = this.Size;
    this.MinimumSize = this.Size;
}

private void BtnGiris_MouseHover(object sender, EventArgs e)
{
    BtnGiris.ForeColor = Color.White;
    BtnGiris.BackColor = Color.Green;
}

private void BtnGiris_MouseLeave(object sender, EventArgs e)
{
    BtnGiris.ForeColor = Color.Black;
    BtnGiris.BackColor = Color.Transparent;
} } }

```

2. Ana Yüz Ekranı (Home)

Ana yüz ekranı ile sistem saatini, yetki durumuna göre açılacak menünün boyutu ve görüntülenecek grup kutularını yansıtıyorum. Kullanıcılara gösterilmesi için duyuru sistemi bu alanda bulunmaktadır. MDiClient kullanılarak açılacak menü, ana yüzden dışarı çıkamamaktadır.

Kodlar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class Home : Form
    {
        public Menu menu;
        public Home()
        {
            menu = new Menu();
            InitializeComponent();
        }

        public OleDbConnection bag = new
        OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
        Source=Data/MarketData.accdb");
        public DataTable yonetim = new DataTable();
        public DataTable musteri = new DataTable();
        public DataTable sirket = new DataTable();
        public OleDbCommand kmt = new OleDbCommand();//acces komut kullanımı
        değişkeni tanımlanıyor.
        public bool durum;
        private void duyurular()
        {
            OleDbConnection baglanti = new OleDbConnection();
            baglanti.ConnectionString = "Provider=Microsoft.ACE.Oledb.12.0;Data
            Source=Data/MarketData.accdb";
            OleDbCommand komut = new OleDbCommand();
```

```

        komut.CommandText = "SELECT *FROM duyurular ORDER BY duyuru_id
DESC";
        //ORDER BY kolon_adı(ları) ASC|DESC
        komut.Connection = baglanti;
        komut.CommandType = CommandType.Text;

        OleDbDataReader dr;
        baglanti.Open();
        dr = komut.ExecuteReader();
        while (dr.Read())
        {

            listBox1.Items.Add(dr["duyuru_text"]);
        }
        baglanti.Close();
    }
    private void Home_Load(object sender, EventArgs e)
    {

        // this.Size = new Size(900, 300);
        this.MinimumSize = new Size(1000, 600);
        duyurular();
        timer1.Start();
        Menu menu = new Menu();
        musterilistesi mlistesi = new musterilistesi();
        if (label5.Text == "False")
        {
            menu.groupBox1.Visible = false;
            menu.groupBox5.Visible = false;
            menu.Size = new Size(450, 272);
            label7.Text = "Kasiyer";
        }
        else
        {
            label7.Text = "Yönetici";

            menu.groupBox1.Visible = true;
            menu.groupBox5.Visible = true;

            this.Size = new Size(655, 342);
            //655; 342

        }
        /* Arka Plan Rengi İçin Kodlar Başlangıç */
        MdiClient ctlMDI;
        foreach (Control ctl in this.Controls)
        {
            try
            {

```

```

        ctlMDI = (MdiClient)ctl;
        ctlMDI.BackColor = this.BackColor;
    }
    catch (InvalidCastException)
    {
    }
}
/* Arka Plan Rengi İçin Kodlar Son */
menu.MdiParent = this;
menu.Show();

}
private void button1_Click(object sender, EventArgs e)
{
    Login login = new Login();
    Application.Exit();
}
private void timer1_Tick(object sender, EventArgs e)
{
    label8.Text = "Tarih : "+DateTime.Now.ToLongDateString() + "\nSaat : " +
DateTime.Now.ToLongTimeString() + "\nSistem saatiniz gösterilir!";
}

private void button2_Click(object sender, EventArgs e)
{
    if (this.FormBorderStyle == FormBorderStyle.None)
    {
        this.FormBorderStyle = FormBorderStyle.Fixed3D;
    }
    else
    {
        this.FormBorderStyle = FormBorderStyle.None;
        this.WindowState = FormWindowState.Maximized;
    }
}

private void Home_FormClosed(object sender, FormClosedEventArgs e)
{
    Environment.Exit(0);

} } }

```

3. Ara Yüz (Menu)

Formların açılma işlemini yapmaktadır. Formlar tanıtlıdır ve açılma komutları verilmiştir.

Kodlar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace RecX_Market_Otomasyonu
{
    public partial class Menu : Form
    {
        public Menu()
        {
            InitializeComponent();
        }

        private void Menu_Load(object sender, EventArgs e)
        {
            this.MaximumSize = this.Size;
            this.MinimumSize = this.Size;
        }

        private void button8_Click(object sender, EventArgs e)
        {
            Sbilgi sbilgi = new Sbilgi();
            Home home = new Home();
            sbilgi.ShowDialog();
        }

        private void urunlist_Click(object sender, EventArgs e)
        {
            Perakendeurungiris perakendeurungiris = new Perakendeurungiris();
            perakendeurungiris.ShowDialog();
        }

        private void button9_Click(object sender, EventArgs e)
        {
            Cari_firmagir firmagir = new Cari_firmagir();
            firmagir.ShowDialog();
        }

        private void button10_Click(object sender, EventArgs e)
```

```

{
    Perakendeurungiris perakendeurungiris = new Perakendeurungiris();
    perakendeurungiris.ShowDialog();
}

private void button11_Click(object sender, EventArgs e)
{
    musterilistesi musterilistesi = new musterilistesi();
    musterilistesi.ShowDialog();
}

private void button12_Click(object sender, EventArgs e)
{
    Perakendesatisgirisi satisgirisi = new Perakendesatisgirisi();
    satisgirisi.ShowDialog();
}

private void button1_Click(object sender, EventArgs e)
{
    yonetimhesap yonetimhesap = new yonetimhesap();
    yonetimhesap.ShowDialog();
}

private void button2_Click(object sender, EventArgs e)
{
    hesapolustur hesapolustur = new hesapolustur();
    hesapolustur.ShowDialog();
}

private void button3_Click(object sender, EventArgs e)
{
    hesapyetkilendir hesapyetkilendir = new hesapyetkilendir();
    hesapyetkilendir.ShowDialog();
}

private void button4_Click_2(object sender, EventArgs e)
{
    hesapsil hesapsil = new hesapsil();
    hesapsil.ShowDialog();
}

private void button5_Click(object sender, EventArgs e)
{
    Kategoriislem kategoriislem = new Kategoriislem();
    kategoriislem.ShowDialog();
}

private void button6_Click(object sender, EventArgs e)
{
    satisbilgi satisbilgi = new satisbilgi();
    satisbilgi.ShowDialog();
}

```

```
private void button7_Click(object sender, EventArgs e)
{
    duyuruislemler duyuruislemler = new duyuruislemler();
    duyuruislemler.ShowDialog();
} } }
```

4. Ürün Giriş (Perakendeciler Girişi)

Veri tabanına ürün giriş yapılan ekrandır burada ürün girişi yapılabilir, ürün silenebilir, ürün düzenlenebilir ve ürün arama işlemleri yapılabilir.

Kodlar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Collections;

namespace RecX_Market_Otomasyonu
{
    public partial class PerakendecilerGiris : Form
    {
        public Home home;
        OleDbConnection con;
        OleDbDataAdapter da;
        OleDbCommand cmd;
        DataSet ds;

        public PerakendecilerGiris()
        {
            InitializeComponent();
        }
        StringFormat strFormat;
        ArrayList arrColumnLefts = new ArrayList();
        ArrayList arrColumnWidths = new ArrayList();
        int iCellHeight = 0;
        int iTotalWidth = 0;
        int iRow = 0;
        bool bFirstPage = false;
        bool bNewPage = false;
        int iHeaderHeight = 0;
```



```

void griddoldur()
{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect *from urunler", con);
    ds = new DataSet();
    con.Open();
    da.Fill(ds, "urunler");
    dataGridView1.DataSource = ds.Tables["urunler"];
    con.Close();
}
void kategoridoldur()
{
    comboBox1.Items.Clear();
    comboBox2.Items.Clear();
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    cmd = new OleDbCommand();
    OleDbDataReader dr;
    con.Open();
    cmd.Connection = con;
    cmd.CommandText = "SELECT * FROM kategori";
    dr = cmd.ExecuteReader();

    while (dr.Read())
    {
        comboBox1.Items.Add(dr["kategori_adi"]);
        comboBox2.Items.Add(dr["kategori_adi"]);
    }
    con.Close();
}
private void kategoriid()
{
    OleDbConnection con = new
OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    con.Open(); //accdb dosyaları için
    OleDbCommand cmd = new OleDbCommand("Select kategori_id from
kategori where kategori_adi= '"+comboBox1.Text+"'", con);
    OleDbDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        textBox1.Text = dr["kategori_id"].ToString();
    }
    con.Close();
}

void kontrol()

```

```

    {
        Home home = new Home();
    }

private void Perakendeurungiris_Load(object sender, EventArgs e)
{
    this.Size = new Size(1140, 583);
    this.MaximumSize = this.Size;
    this.MinimumSize = this.Size;
    griddoldur();
    kategoridoldur();
    kontrol();
    dataGridView1.RowHeadersWidth = 60;
    dataGridView1.AllowUserToAddRows = false;
    dataGridView1.Columns[0].HeaderText = "ID";
    dataGridView1.Columns[1].HeaderText = "Ürün Adı";
    dataGridView1.Columns[2].HeaderText = "Kategori";
    dataGridView1.Columns[3].HeaderText = "Adet";
    dataGridView1.Columns[4].HeaderText = "Giriş Fiyatı(TL)";
    dataGridView1.Columns[5].HeaderText = "Satış Fiyatı(TL)";
    dataGridView1.Columns[6].HeaderText = "Barkod Numarası";
    dataGridView1.Columns[7].HeaderText = "Kategori ID";
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        kategoriid();
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;

        cmd.CommandText = "insert into urunler
(urun_adi,urun_kategori,urun_adi,urun_fiyati,urun_sfiyati,urun_barkod,kategori_id) values ('" + urun_adi.Text + "','" + comboBox1.Text + "','" + urun_adi.Text + "','"
+ urun_fiyati.Text + "','" + urun_sfiyati.Text + "','" + urun_barkod.Text + "','" +
textBox1.Text + "')";
        cmd.ExecuteNonQuery();
        con.Close();
        griddoldur();
    }
    catch (Exception)
    {
        MessageBox.Show("Hatalı işlem yaptınız lütfen kontrol ediniz");
    }
}

private void button3_Click(object sender, EventArgs e)

```

```

    {
        try
        {
            kategoriid();
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "update urunler set urun_adi='" + urun_adi.Text +
            "',urun_kategori='" + comboBox1.Text + "',urun_adedi='" +
            Int32.Parse(urun_adedi.Text) + "',urun_fiyati='" + urun_fiyati.Text +
            "',urun_sfiyati='" + urun_sfiyati.Text + "',urun_barkod='" +
            Int32.Parse(urun_barkod.Text) + "',kategori_id='" + textBox1.Text + "' where
            urun_id=" + Int32.Parse(urun_id.Text) + """;
            cmd.ExecuteNonQuery();
            con.Close();
            griddoldur();
        }
        catch (Exception)
        {
            MessageBox.Show("Hatalı işlem yaptınız kontrol ediniz");
            con.Close();
        }
    }

    private void button2_Click(object sender, EventArgs e)
    {
        DialogResult secenek = MessageBox.Show(urun_adi.Text + " " + "Silmek
        istiyor musunuz?", "Bilgilendirme Penceresi", MessageBoxButtons.YesNo,
        MessageBoxIcon.Information);
        if (secenek == DialogResult.Yes)
        {
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "delete from urunler where urun_id=" + urun_id.Text
            + "";
            cmd.ExecuteNonQuery();
            con.Close();
            griddoldur();
        }
        else if (secenek == DialogResult.No)
        {
            MessageBox.Show("Silme işlemi gerçekleştirilmedi", "Silme İşlemi",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void urun_ara_TextChanged(object sender, EventArgs e)
    {

```

```

        con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
        da = new OleDbDataAdapter("SElect *from urunler where urun_adi like '" +
urun_ara.Text + "%'", con);
        ds = new DataSet();
        con.Open();
        da.Fill(ds, "urunler");
        dataGridView1.DataSource = ds.Tables["urunler"];
        con.Close();
    }
    private void button4_Click(object sender, EventArgs e)
    {
        griddoldur();
    }
    private void dataGridView1_CellEnter(object sender,
DataGridViewCellEventArgs e)
    {
        urun_id.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
        urun_adi.Text = dataGridView1.CurrentRow.Cells[1].Value.ToString();
        comboBox1.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
        urun_adi.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
        urun_fiyati.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
        urun_sfiyati.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
        urun_barkod.Text = dataGridView1.CurrentRow.Cells[6].Value.ToString();
        textBox1.Text = dataGridView1.CurrentRow.Cells[7].Value.ToString();
    }
    private void comboBox2_TextChanged(object sender, EventArgs e)
    {
        con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
        da = new OleDbDataAdapter("SElect *from urunler where urun_kategori like
'" + comboBox2.Text + "%'", con);
        ds = new DataSet();
        con.Open();
        da.Fill(ds, "urunler");
        dataGridView1.DataSource = ds.Tables["urunler"];
        con.Close();
    }
    private void urun_sfiyati_KeyPress(object sender, KeyPressEventArgs e)
    {
        if ((int)e.KeyChar >= 48 && (int)e.KeyChar <= 57)
        {
            e.Handled = false; //eğer rakamsa yazdır.
        }
        else if ((int)e.KeyChar == 44)
        {
            e.Handled = false; //eğer basılan tuş virgöl ise yazdır.
        }
        else if ((int)e.KeyChar == 08)
    }

```

```

    {
        e.Handled = false;//eğer basılan tuş backspace ise yazdır.
    }
    else
    {
        e.Handled = true;//bunların dışındaysa hiçbirisini yazdırma
    }
}

private void urun_fiyati_KeyPress(object sender, KeyPressEventArgs e)
{
    if ((int)e.KeyChar >= 48 && (int)e.KeyChar <= 57)
    {
        e.Handled = false;//eğer rakamsa yazdır.
    }
    else if ((int)e.KeyChar == 44)
    {
        e.Handled = false;//eğer basılan tuş virgül ise yazdır.
    }
    else if ((int)e.KeyChar == 08)
    {
        e.Handled = false;//eğer basılan tuş backspace ise yazdır.
    }
    else
    {
        e.Handled = true;//bunların dışındaysa hiçbirisini yazdırma
    }
}

private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void urun_adi_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void urun_barkod_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    try
    {
        int iLeftMargin = e.MarginBounds.Left;
        int iTopMargin = e.MarginBounds.Top;
        bool bMorePagesToPrint = false;
        int iTmpWidth = 0;
    }
}

```

```

bFirstPage = true;

if (bFirstPage)
{
    foreach (DataGridViewColumn GridCol in dataGridView1.Columns)
    {
        iTmpWidth = (int)(Math.Floor(((double)((double)GridCol.Width /
            (double)iTotalWidth * (double)iTotalWidth *
            ((double)e.MarginBounds.Width / (double)iTotalWidth))));

        iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
            GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;
        arrColumnLefts.Add(iLeftMargin);
        arrColumnWidths.Add(iTmpWidth);
        iLeftMargin += iTmpWidth;
    }
}
while (iRow <= dataGridView1.Rows.Count - 1)
{
    DataGridViewRow GridRow = dataGridView1.Rows[iRow];

    iCellHeight = GridRow.Height + 5;
    int iCount = 0;

    if (iTopMargin + iCellHeight >= e.MarginBounds.Height +
e.MarginBounds.Top)
    {
        bNewPage = true;
        bFirstPage = false;
        bMorePagesToPrint = true;
        break;
    }
    else
    {
        if (bNewPage)
        {
            e.Graphics.DrawString("Ürün          Listesi",          new
Font(dataGridView1.Font, FontStyle.Bold),
            Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
e.Graphics.MeasureString("Ürün          Listesi",          new
Font(dataGridView1.Font,
            FontStyle.Bold), e.MarginBounds.Width).Height - 13);

            String strDate = DateTime.Now.ToShortDateString() + " " +
DateTime.Now.ToShortTimeString();

            e.Graphics.DrawString(strDate,          new Font(dataGridView1.Font,
FontStyle.Bold),

```

```

        Brushes.Black, e.MarginBounds.Left +
(e.MarginBounds.Width -
    e.Graphics.MeasureString(strDate, new
Font(dataGridView1.Font,
    FontStyle.Bold), e.MarginBounds.Width).Width),
e.MarginBounds.Top -
    e.Graphics.MeasureString("Ürün Listesi", new Font(new
Font(dataGridView1.Font,
    FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
    iTopMargin = e.MarginBounds.Top;
    foreach (DataGridViewColumn GridCol in
dataGridView1.Columns)
    {
        e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
            new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
                (int)arrColumnWidths[iCount], iHeaderHeight));
        e.Graphics.DrawRectangle(Pens.Black,
            new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
                (int)arrColumnWidths[iCount], iHeaderHeight));
        e.Graphics.DrawString(GridCol.HeaderText,
GridCol.InheritedStyle.Font,
            new SolidBrush(GridCol.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
                (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
        iCount++;
    }
    bNewPage = false;
    iTopMargin += iHeaderHeight;
}
iCount = 0;
foreach (DataGridViewCell Cel in GridRow.Cells)
{
    if (Cel.Value != null)
    {
        e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font, new
SolidBrush(Cel.InheritedStyle.ForeColor), new
            RectangleF((int)arrColumnLefts[iCount],
(float)iTopMargin, (int)arrColumnWidths[iCount], (float)iCellHeight), strFormat);
    }
    e.Graphics.DrawRectangle(Pens.Black, new
Rectangle((int)arrColumnLefts[iCount],
        iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));

    iCount++;
}
}
iRow++;
iTopMargin += iCellHeight;
}

```

```

if (bMorePagesToPrint)
    e.HasMorePages = true;
else
    e.HasMorePages = false;
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

private void printDocument1_BeginPrint(object sender,
System.Drawing.Printing.PrintEventArgs e)
{
    try
    {
        printDocument1.OriginAtMargins = true;
        printDocument1.DefaultPageSettings.Margins.Left = 10;
        printDocument1.DefaultPageSettings.Margins.Right = 40;
        printDocument1.DefaultPageSettings.Margins.Top = 40;
        printDocument1.DefaultPageSettings.Margins.Bottom = 0;
        strFormat = new StringFormat();
        strFormat.Alignment = StringAlignment.Near;
        strFormat.LineAlignment = StringAlignment.Center;
        strFormat.Trimming = StringTrimming.EllipsisCharacter;
        arrColumnLefts.Clear();
        arrColumnWidths.Clear();
        iCellHeight = 0;
        iRow = 0;
        bFirstPage = true;
        bNewPage = true;
        iTotWidth = 0;
        foreach (DataGridViewColumn dgvGridCol in dataGridView1.Columns)
        {
            iTotWidth += dgvGridCol.Width;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}

private void button5_Click(object sender, EventArgs e)
{
    PrintPreviewDialog onizleme = new PrintPreviewDialog();
    onizleme.Document = printDocument1;
}

```



```

        onizleme.ShowDialog();
    }
}
}

```

5. Firma / Müşteri Girişi (Cari_Firmagir)

Müşteri veya firma girişlerini buradan yapıyorum tabloyu çok geniş hazırlamıştım fakat sonrasında fazla bilgiye gerek duymadığım için veri tabanından ve formdan kaldırdım fakat kod bölümünde tekrar eklenmesi istenebilir diye bilgi satırına çevirerek kodlamasını bıraktım. Resim seçimi yapılırken resim veri tabanında veya klasörde aynı isim ile olabileceği ihtimali ile resimlere benzersiz isim vererek kayıt işlemi gerçekleştirdim bu sayede ileride karşılaşacağım sorunlardan bir nebze kurtulmuş oldum.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;

namespace RecX_Market_Otomasyonu
{
    public partial class Cari_firmagir : Form
    {
        public Home home;
        public Cari_firmagir()
        {
            InitializeComponent();
        }
        string dosyaYolu;
        string dosyaAdi;
        private void button3_Click(object sender, EventArgs e)
        {
            try
            {
                dosyaAdi = Path.GetFileName(dosyaYolu); //Dosya adını alma
            }
            catch { }
        }
    }
}

```

```

string kaynak = dosyaYolu;
string hedef = Application.StartupPath + @"Data\img\Musteri/";
try
{
    string yeniad = Guid.NewGuid() + ".jpg"; //Benzersiz isim verme
    File.Copy(kaynak, hedef + yeniad);
    mus_fotoyol.Text = hedef + yeniad;
}
catch
{
    //Boş Resim Seçilen Müşterilerde Belirli bir foto gelmesi için
    string boshedef = Application.StartupPath + @"Data\img\AppIMG/";
    mus_fotoyol.Text = boshedef + "musteri.png";
}
string vtyolu = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=Data/MarketData.accdb;Persist Security Info=True";
OleDbConnection baglanti = new OleDbConnection(vtyolu);
baglanti.Open();
string ekle = "insert into musteriler ( mus_fir,mus_ad, mus_unvan, mus_tel1,
mus_cep, mus_fotoyol, mus_email, mus_tcno, mus_dogtar, mus_mahkoy)" +
"
values
(@mus_fir,@mus_ad,@mus_unvan,@mus_tel1,@mus_cep,@mus_fotoyol,@mus_e
mail,@mus_tcno,@mus_dogtar,@mus_mahkoy)";
OleDbCommand komut = new OleDbCommand(ekle, baglanti);
komut.Parameters.AddWithValue("@mus_fir", mus_fir.Text);
komut.Parameters.AddWithValue("@mus_ad", mus_ad.Text);
komut.Parameters.AddWithValue("@mus_unvan", mus_unvan.Text);
komut.Parameters.AddWithValue("@mus_tel1", mus_tel1.Text);
komut.Parameters.AddWithValue("@mus_cep", mus_cep.Text);
komut.Parameters.AddWithValue("@mus_fotoyol", mus_fotoyol.Text);
komut.Parameters.AddWithValue("@mus_email", mus_email.Text);
komut.Parameters.AddWithValue("@mus_tcno", mus_tcno.Text);
komut.Parameters.AddWithValue("@mus_dogtar",
datePicker1.Value);
komut.Parameters.AddWithValue("@mus_mahkoy", mus_mahkoy.Text);
komut.ExecuteNonQuery();
MessageBox.Show(mus_ad.Text + " " + mus_unvan.Text + " Müşterisi Kayıt Edildi.");
this.Close();
}
catch (Exception)
{
    MessageBox.Show("İşlem Gerçekleştirilemedi !!!");
}
}
private void button4_Click(object sender, EventArgs e)
{
    OpenFileDialog dosya = new OpenFileDialog();
    dosya.Filter = "Resim Dosyası |*.jpg;*.nef;*.png | Tüm Dosyalar |*.*";
    dosya.ShowDialog();
}

```

```

        dosyaYolu = dosya.FileName;
        pictureBox1.ImageLocation = dosyaYolu;
    }
    private void mus_fir_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    }
    private void mus_tel1_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    }
    private void mus_cep_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    }
    private void mus_tcno_KeyPress(object sender, KeyPressEventArgs e)
    {
        e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
    }
}
}

```

6. Kategori İşlemleri (Kategoriislem)

Buradan kategori oluşturabilir, oluşturulan kategoriye silebilir ve ismini düzenleyebilirsiniz. 3 Adet buton aracılığı ile grupları gizleme ve görünür kılma özelliği ile küçük bir pencerede bu işlemleri basit ve kullanışlı bir şekilde gerçekleştirmeyi sağladım.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class Kategoriislem : Form
    {
        OleDbConnection con;
    }
}

```

```

OleDbDataAdapter da;
OleDbCommand cmd;
DataSet ds;

public Kategoriislem()
{
    InitializeComponent();
}
void griddoldur()
{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect *from kategori", con);
    ds = new DataSet();
    con.Open();
    con.Close();
    kategorigetir();
}
void kategorigetir()
{
    comboBox1.Items.Clear();
    comboBox2.Items.Clear();
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    cmd = new OleDbCommand();
    OleDbDataReader dr;
    con.Open();
    cmd.Connection = con;
    cmd.CommandText = "SELECT * FROM kategori";
    dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Items.Add(dr["kategori_adi"]);
        comboBox2.Items.Add(dr["kategori_adi"]);
    }
    con.Close();
}

private void Kategoriislem_Load(object sender, EventArgs e)
{
    griddoldur();
    this.Size = new Size(459, 254);
    this.MaximumSize = this.Size;
    this.MinimumSize = this.Size;
}
private void button1_Click(object sender, EventArgs e)
{
    try
    {

```

```

        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "insert into kategori (kategori_adi) values (" +
textBox1.Text + ")";
        cmd.ExecuteNonQuery();
        con.Close();
        label2.Text = textBox1.Text + " Adın bir kategori oluşturuldu";
        griddoldur();
    }
    catch (Exception hata)
    {
        label2.Text = textBox1.Text + "Oluşturulurken bir hata meydana geldi\n"
+hata;
    }
}
private void button2_Click(object sender, EventArgs e)
{
    try
    {
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "update kategori set kategori_adi=" + textBox2.Text + " where
kategori_adi=" + comboBox1.Text + """;
        cmd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show(comboBox1.Text+" İsimli kategori " + textBox2.Text+" Şeklinde
değiştirilmiştir");
        griddoldur();
    }
    catch (Exception gunhat)
    {
        MessageBox.Show("Değiştirme işleminde hata oluştu" +gunhat);
    }
}
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "delete from kategori where kategori_adi=" +
comboBox2.Text + """;
        cmd.ExecuteNonQuery();
        con.Close();
        MessageBox.Show(comboBox2.Text + " İsimli kategori silindi");
        griddoldur();
    }
    catch (Exception gunhat)
    {
        MessageBox.Show("Silme işleminde hata oluştu" +gunhat);
    }
}

```

```

    }
    catch (Exception silhat)
    {
        MessageBox.Show("Silme İşleminde Bir Hata Meydana Geldi\n"+silhat);
    }
}
private void button4_Click(object sender, EventArgs e)
{
    groupBox1.Visible = true;
    groupBox1.Location = new Point(12, 67);
    groupBox2.Visible = false;
    groupBox3.Visible = false;
    label2.Text = "";
}
private void button5_Click(object sender, EventArgs e)
{
    groupBox2.Visible = true;
    groupBox2.Location = new Point(12, 67);
    groupBox1.Visible = false;
    groupBox3.Visible = false;
    label2.Text = "";
}
private void button6_Click(object sender, EventArgs e)
{
    groupBox3.Visible = true;
    groupBox3.Location = new Point(12, 67);
    groupBox1.Visible = false;
    groupBox2.Visible = false;
    label2.Text = "";
}
}
}

```

7. Satış Bilgisi(Satisbilgi)

Bu formu günlük ve aylık satışlarınızı görebileceğiniz şekilde oluşturdum bu sayede kasada gerçekleşen satış durumlarınızı görmeyi sağladım.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
using System.Collections;
using Microsoft.VisualBasic;

namespace RecX_Market_Otomasyonu
{
    public partial class satisbilgi : Form
    {
        public satisbilgi()
        {
            InitializeComponent();
        }
        StringFormat strFormat;
        ArrayList arrColumnLefts = new ArrayList();
        ArrayList arrColumnWidths = new ArrayList();
        int iCellHeight = 0;
        int iTotalWidth = 0;
        int iRow = 0;
        bool bFirstPage = false;
        bool bNewPage = false;
        int iHeaderHeight = 0;
        OleDbConnection con;
        OleDbDataAdapter da;
        // OleDbCommand cmd;
        DataSet ds;
        private void vericek()
        {
            con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
            da = new OleDbDataAdapter("Select *from satislar", con);
            ds = new DataSet();
            con.Open();
            da.Fill(ds, "satislar");
            dataGridView1.DataSource = ds.Tables["satislar"];
        }
        private string Daily_Total()
        {
            DateTime date = new DateTime(monthCalendar1.SelectionRange.Start.Year,
monthCalendar1.SelectionRange.Start.Month,
monthCalendar1.SelectionRange.Start.Day);
            string Query = "select sum(urun_sfiyati) as daily_total from satislar where
sale_date=@sale_date";
            OleDbCommand cmd = new OleDbCommand(Query, con);
            cmd.Parameters.AddWithValue("@sale_date", date);
            OleDbDataReader result = cmd.ExecuteReader();
            while (result.Read())

```

```

        {
            string value = result["daily_total"].ToString();

            if (!String.IsNullOrEmpty(value))
            {
                return value;
            }
        }
        return "0,00";
    }
}

private string Monthly_Total()
{
    DateTime start_date = new
    DateTime(monthCalendar1.SelectionRange.Start.Year,
    monthCalendar1.SelectionRange.Start.Month, 1);
    DateTime end_date = new
    DateTime(monthCalendar1.SelectionRange.Start.Year,
    monthCalendar1.SelectionRange.Start.Month, 1).AddMonths(1);
    string Query = "select sum(urun_sfıyati) as monthly_total from satislar where
    sale_date>=@sale_date_start and sale_date<@sale_date_end";
    OleDbCommand cmd = new OleDbCommand(Query, con);
    cmd.Parameters.AddWithValue("@sale_date_start", start_date);
    cmd.Parameters.AddWithValue("@sale_date_end", end_date);
    OleDbDataReader result = cmd.ExecuteReader();
    while (result.Read())
    {
        string value = result["monthly_total"].ToString();
        if (!String.IsNullOrEmpty(value))
        {
            return value;
        }
    }
    return "0,00";
}

private void satisbilgi_Load(object sender, EventArgs e)
{
    this.Size = new Size(790, 371);
    this.MaximumSize = this.Size;
    this.MinimumSize = this.Size;
    vericek();
    dataGridView1.AllowUserToAddRows = false;
    dataGridView1.Columns[0].HeaderText = "Satış ID";
    dataGridView1.Columns[0].Width = 100;
    dataGridView1.Columns[1].HeaderText = "Tarih";
    dataGridView1.Columns[1].Width = 100;
    dataGridView1.Columns[2].HeaderText = "Alınan Ücret (TL)";
    dataGridView1.Columns[2].Width = 140;
    dataGridView1.Columns[3].HeaderText = "Müşteri Adı";
    dataGridView1.Columns[3].Width = 120;
    label1.Text = Daily_Total() + " TL";
}

```



```

        label2.Text = Monthly_Total() + " TL";
    }
    private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
    {
        try
        {
            int iLeftMargin = e.MarginBounds.Left;
            int iTopMargin = e.MarginBounds.Top;
            bool bMorePagesToPrint = false;
            int iTmpWidth = 0;
            bFirstPage = true;
            if (bFirstPage)
            {
                foreach (DataGridViewColumn GridCol in dataGridView1.Columns)
                {
                    iTmpWidth = (int)(Math.Floor((double)((double)GridCol.Width /
                        (double)iTotalWidth * (double)iTotalWidth *
                        ((double)e.MarginBounds.Width / (double)iTotalWidth))));

                    iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
                        GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;

                    arrColumnLefts.Add(iLeftMargin);
                    arrColumnWidths.Add(iTmpWidth);
                    iLeftMargin += iTmpWidth;
                }
            }
            while (iRow <= dataGridView1.Rows.Count - 1)
            {
                DataGridViewRow GridRow = dataGridView1.Rows[iRow];
                iCellHeight = GridRow.Height + 5;
                int iCount = 0;
                if (iTopMargin + iCellHeight >= e.MarginBounds.Height +
e.MarginBounds.Top)
                {
                    bNewPage = true;
                    bFirstPage = false;
                    bMorePagesToPrint = true;
                    break;
                }
                else
                {
                    if (bNewPage)
                    {
e.Graphics.DrawString("Kasa Raporu", new Font(dataGridView1.Font,
FontStyle.Bold),
Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -

```

```

        e.Graphics.MeasureString("Kasa Raporu", new
Font(dataGridView1.Font,
        FontStyle.Bold), e.MarginBounds.Width).Height - 13);

        String strDate = DateTime.Now.ToShortDateString() + " " +
DateTime.Now.ToShortTimeString();

        e.Graphics.DrawString(strDate, new Font(dataGridView1.Font,
FontStyle.Bold),
        Brushes.Black, e.MarginBounds.Left +
(e.MarginBounds.Width -
        e.Graphics.MeasureString(strDate, new
Font(dataGridView1.Font,
        FontStyle.Bold), e.MarginBounds.Width).Width),
e.MarginBounds.Top -
        e.Graphics.MeasureString("Kasa Raporu", new Font(new
Font(dataGridView1.Font,
        FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);

        iTopMargin = e.MarginBounds.Top;
        foreach (DataGridViewColumn GridCol in
dataGridView1.Columns)
        {
            e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
            new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
            (int)arrColumnWidths[iCount], iHeaderHeight));
            e.Graphics.DrawRectangle(Pens.Black,
            new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
            (int)arrColumnWidths[iCount], iHeaderHeight));
            e.Graphics.DrawString(GridCol.HeaderText,
GridCol.InheritedStyle.Font,
            new SolidBrush(GridCol.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
            (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
            iCount++;
        }
        bNewPage = false;
        iTopMargin += iHeaderHeight;
    }
    iCount = 0;

    foreach (DataGridViewCell Cel in GridRow.Cells)
    {
        if (Cel.Value != null)
        {
            e.Graphics.DrawString(Cel.Value.ToString(),
Cel.InheritedStyle.Font,
            new SolidBrush(Cel.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], (float)iTopMargin,

```

```

                                (int)arrColumnWidths[iCount],        (float)iCellHeight),
strFormat);
                                }
                                e.Graphics.DrawRectangle(Pens.Black,                                new
Rectangle((int)arrColumnLefts[iCount],
                                iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));

                                iCount++;
                                }
                                }
                                iRow++;
                                iTopMargin += iCellHeight;
                                }
if (bMorePagesToPrint)
    e.HasMorePages = true;
else
    e.HasMorePages = false;
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message,    "Error",    MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}
private    void    printDocument1_BeginPrint(object    sender,
System.Drawing.Printing.PrintEventArgs e)
{
    try
    {
        printDocument1.OriginAtMargins = true;
        printDocument1.DefaultPageSettings.Margins.Left = 10;
        printDocument1.DefaultPageSettings.Margins.Right = 40;
        printDocument1.DefaultPageSettings.Margins.Top = 40;
        printDocument1.DefaultPageSettings.Margins.Bottom = 0;
        strFormat = new StringFormat();
        strFormat.Alignment = StringAlignment.Near;
        strFormat.LineAlignment = StringAlignment.Center;
        strFormat.Trimming = StringTrimming.EllipsisCharacter;
        arrColumnLefts.Clear();
        arrColumnWidths.Clear();
        iCellHeight = 0;
        iRow = 0;
        bFirstPage = true;
        bNewPage = true;

        iTotalWidth = 0;
        foreach (DataGridViewColumn dgvGridCol in dataGridView1.Columns)
        {
            iTotalWidth += dgvGridCol.Width;

```

```

        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
private void button1_Click(object sender, EventArgs e)
{
    PrintPreviewDialog onizleme = new PrintPreviewDialog();
    onizleme.Document = printDocument1;
    onizleme.ShowDialog();
} } }

```

8. Müşteri Listesi (Musterilistesi)

Müşterilerinizin tamamını görebileceğiniz bir ekran oluşturdum bu ekranda müşterileri güncelleyebilir ve silebilirsiniz ekle seçeneği ile müşteri ekleme ekranına giderek ekleme işlemi gerçekleştirebilirsiniz. Yenile seçeneği ile yeni müşteriler eklenmesi durumunda ekranı güncelleyebilirsiniz. Müşteri adına ve TC numarasına göre müşteri arayabilirsiniz.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Collections;

namespace RecX_Market_Otomasyonu
{
    public partial class musterilistesi : Form
    {
        public Home home;
        OleDbConnection con;
        OleDbDataAdapter da;
        OleDbCommand cmd;
    }
}

```

```

DataSet ds;
StringFormat strFormat;
ArrayList arrColumnLefts = new ArrayList();
ArrayList arrColumnWidths = new ArrayList();
int iCellHeight = 0;
int iTotalWidth = 0;
int iRow = 0;
bool bFirstPage = false;
bool bNewPage = false;
int iHeaderHeight = 0;
public musterilistesi()

{
    InitializeComponent();
}

void griddoldur()
{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect *from musteri", con);
    ds = new DataSet();
    con.Open();
    da.Fill(ds, "musteri");
    dataGridView1.DataSource = ds.Tables["musteri"];
    con.Close();
}

void kontrol()
{
    Home home = new Home();
    dataGridView1.AllowUserToAddRows = false;
    dataGridView1.Columns[0].HeaderText = "ID";
    dataGridView1.Columns[0].Width = 50;
    dataGridView1.Columns[1].HeaderText = "Firma Kodu";
    dataGridView1.Columns[1].Width = 113;
    dataGridView1.Columns[2].HeaderText = "Adı";
    dataGridView1.Columns[2].Width = 90;
    dataGridView1.Columns[3].HeaderText = "Soyadı / Unvanı";
    dataGridView1.Columns[3].Width = 130;
    dataGridView1.Columns[4].HeaderText = "Telefon";
    dataGridView1.Columns[4].Width = 90;
    dataGridView1.Columns[5].HeaderText = "Cep Telefon";
    dataGridView1.Columns[5].Width = 115;
    dataGridView1.Columns[6].HeaderText = "Fotoğraf Yolu";
    dataGridView1.Columns[6].Width = 123;
    dataGridView1.Columns[7].HeaderText = "Mail Adresi";
    dataGridView1.Columns[7].Width = 120;
    dataGridView1.Columns[8].HeaderText = "TC Kimlik No";
    dataGridView1.Columns[8].Width = 123;
}

```

```

        dataGridView1.Columns[9].HeaderText = "Doğum Tarihi";
        dataGridView1.Columns[9].Width = 123;
        dataGridView1.Columns[10].HeaderText = "Adresi";
        dataGridView1.Columns[10].Width = 151;
    }
    private void dataGridView1_CellEnter(object sender,
DataGridViewCellEventArgs e)
    {
        mus_id.Text = dataGridView1.CurrentRow.Cells[0].Value.ToString();
        mus_ad.Text = dataGridView1.CurrentRow.Cells[2].Value.ToString();
        mus_unvan.Text = dataGridView1.CurrentRow.Cells[3].Value.ToString();
        mus_tel1.Text = dataGridView1.CurrentRow.Cells[4].Value.ToString();
        mus_cep.Text = dataGridView1.CurrentRow.Cells[5].Value.ToString();
        mus_adres.Text = dataGridView1.CurrentRow.Cells[10].Value.ToString();
    }
    private void musterilistesi_Load(object sender, EventArgs e)
    {
        this.Size = new Size(1287, 465);
        this.MaximumSize = this.Size;
        this.MinimumSize = this.Size;

        griddoldur();
        kontrol();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Cari_firmagir carifirmagir = new Cari_firmagir();
        carifirmagir.ShowDialog();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "update musteri set mus_ad=" + mus_ad.Text +
        "",mus_unvan=" + mus_unvan.Text + "",mus_tel1=" + mus_tel1.Text +
        "",mus_cep="+ mus_cep.Text+ "",mus_mahkoy="+mus_adres.Text+ "" where
        mus_id=" + mus_id.Text + "";
        cmd.ExecuteNonQuery();
        con.Close();
        griddoldur();
    }
    private void button2_Click(object sender, EventArgs e)
    {
        DialogResult secenek = MessageBox.Show(mus_ad.Text + " " +
        mus_unvan.Text+ " Numaralı müşteriye silmek istiyor musunuz?" , "Bilgilendirme
        Penceresi",MessageBoxButtons.YesNo,MessageBoxIcon.Information);
    }

```

```

        if (secenek == DialogResult.Yes)
        {
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "delete from musteri where mus_id=" + mus_id.Text
+ "";
            cmd.ExecuteNonQuery();
            con.Close();
            griddoldur();
        }
        else if (secenek == DialogResult.No)
        {
            MessageBox.Show("Silme işlemi gerçekleştirilmedi" , "Silme İşlemi"
,MessageBoxButtons.OK,MessageBoxIcon.Error);
        }
    }
}

```

```

private void textBox1_TextChanged(object sender, EventArgs e)

```

```

{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect *from musteri where mus_ad like "" +
textBox1.Text + "%"" , con);
    ds = new DataSet();
    con.Open();
    da.Fill(ds, "musteri");
    dataGridView1.DataSource = ds.Tables["musteri"];
    con.Close();
}

```

```

private void button4_Click(object sender, EventArgs e)

```

```

{
    griddoldur();
}

```

```

private void textBox2_TextChanged(object sender, EventArgs e)

```

```

{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect *from musteri where mus_teno like "" +
textBox2.Text + "%"" , con);
    ds = new DataSet();
    con.Open();
    da.Fill(ds, "musteri");
    dataGridView1.DataSource = ds.Tables["musteri"];
    con.Close();
}

```

```

    }
    private void printDocument1_PrintPage(object sender,
System.Drawing.Printing.PrintPageEventArgs e)
    {
        try
        {
            int iLeftMargin = e.MarginBounds.Left;
            int iTopMargin = e.MarginBounds.Top;
            bool bMorePagesToPrint = false;
            int iTmpWidth = 0;
            bFirstPage = true;
            if (bFirstPage)
            {
                foreach (DataGridViewColumn GridCol in dataGridView1.Columns)
                {
                    iTmpWidth = (int)(Math.Floor(((double)((double)GridCol.Width /
                        (double)iTotalWidth * (double)iTotalWidth *
                        ((double)e.MarginBounds.Width / (double)iTotalWidth))));
                    iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
                        GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;
                    arrColumnLefts.Add(iLeftMargin);
                    arrColumnWidths.Add(iTmpWidth);
                    iLeftMargin += iTmpWidth;
                }
            }

            while (iRow <= dataGridView1.Rows.Count - 1)
            {
                DataGridViewRow GridRow = dataGridView1.Rows[iRow];
                iCellHeight = GridRow.Height + 5;
                int iCount = 0;

                if (iTopMargin + iCellHeight >= e.MarginBounds.Height +
e.MarginBounds.Top)
                {
                    bNewPage = true;
                    bFirstPage = false;
                    bMorePagesToPrint = true;
                    break;
                }
                else
                {
                    if (bNewPage)
                    {
                        e.Graphics.DrawString("Müşteri Listesi", new
Font(dataGridView1.Font, FontStyle.Bold),
Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -

```



```

        e.Graphics.MeasureString("Müşteri Listesi", new
Font(dataGridView1.Font,
        FontStyle.Bold), e.MarginBounds.Width).Height - 13);
        String strDate = DateTime.Now.ToShortDateString() + " " +
DateTime.Now.ToShortTimeString();
        e.Graphics.DrawString(strDate, new Font(dataGridView1.Font,
FontStyle.Bold),
        Brushes.Black, e.MarginBounds.Left +
(e.MarginBounds.Width -
        e.Graphics.MeasureString(strDate, new
Font(dataGridView1.Font,
        FontStyle.Bold), e.MarginBounds.Width).Width),
e.MarginBounds.Top -
        e.Graphics.MeasureString("Müşteri Listesi", new Font(new
Font(dataGridView1.Font,
        FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
        iTopMargin = e.MarginBounds.Top;
        foreach (DataGridViewColumn GridCol in
dataGridView1.Columns)
        {
            e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
(int)arrColumnWidths[iCount], iHeaderHeight));
            e.Graphics.DrawRectangle(Pens.Black,
new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
(int)arrColumnWidths[iCount], iHeaderHeight));
            e.Graphics.DrawString(GridCol.HeaderText,
GridCol.InheritedStyle.Font,
new SolidBrush(GridCol.InheritedStyle.ForeColor),
new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
(int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
            iCount++;
        }
        bNewPage = false;
        iTopMargin += iHeaderHeight;
    }
    iCount = 0;
    foreach (DataGridViewCell Cel in GridRow.Cells)
    {
        if (Cel.Value != null)
        {
            e.Graphics.DrawString(Cel.Value.ToString(),
Cel.InheritedStyle.Font,
new SolidBrush(Cel.InheritedStyle.ForeColor),
new RectangleF((int)arrColumnLefts[iCount],
(float)iTopMargin,
(int)arrColumnWidths[iCount], (float)iCellHeight),
strFormat);
        }
    }

```

```

        e.Graphics.DrawRectangle(Pens.Black,
Rectangle((int)arrColumnLefts[iCount],
        iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));
        iCount++;
    }
}
iRow++;
iTopMargin += iCellHeight;
}

if (bMorePagesToPrint)
    e.HasMorePages = true;
else
    e.HasMorePages = false;
}
catch (Exception exc)
{
    MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
}
}

```

```

private void printDocument1_BeginPrint(object sender,
System.Drawing.Printing.PrintEventArgs e)

```

```

{
    try
    {
        printDocument1.OriginAtMargins = true;
        printDocument1.DefaultPageSettings.Margins.Left = 0;
        printDocument1.DefaultPageSettings.Margins.Right = 0;
        printDocument1.DefaultPageSettings.Margins.Top = 40;
        printDocument1.DefaultPageSettings.Margins.Bottom = 0;
        strFormat = new StringFormat();
        strFormat.Alignment = StringAlignment.Near;
        strFormat.LineAlignment = StringAlignment.Center;
        strFormat.Trimming = StringTrimming.EllipsisCharacter;
        arrColumnLefts.Clear();
        arrColumnWidths.Clear();
        iCellHeight = 0;
        iRow = 0;
        bFirstPage = true;
        bNewPage = true;
        iTotWidth = 0;
        foreach (DataGridViewColumn dgvGridCol in dataGridView1.Columns)
        {
            iTotWidth += dgvGridCol.Width;
        }
    }
}

```

```

    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    }
}
private void button5_Click(object sender, EventArgs e)
{
    PrintPreviewDialog onizleme = new PrintPreviewDialog();
    onizleme.Document = printDocument1;
    onizleme.ShowDialog();
} } }

```

9. Şirket Bilgileri Girişi (Sbilgi)

Bu ekrandan veritabanında 1 adet olan şirket bilgilerini düzenleyebilirsiniz. Silme işlemi ya da yeni bir kayıt oluşturamazsınız.

Not: Sadece yönetici yetkisine sahip kişiler görebilir.

Kodlar

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class Sbilgi : Form
    {
        OleDbConnection con;
        OleDbDataAdapter da;
        OleDbCommand cmd;
        DataSet ds;
        public Sbilgi()
        {
            InitializeComponent();
        }
    }
}

```

```

void griddoldur()
{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect *from sirket", con);
    ds = new DataSet();
    con.Open();
    da.Fill(ds, "sirket");
    con.Close();
}
private void Sbilgi_Load(object sender, EventArgs e)
{
    griddoldur();
}
private void BtnKaydet_Click(object sender, EventArgs e)
{
    try
    {
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "update sirket set s_sk=" + s_sk.Text + ",s_ad=" +
s_ad.Text
        + ",s_tel=" + s_tel.Text + ",s_faks=" + s_faks.Text + ",s_mail="
        + s_mail.Text + ",s_web=" + s_web.Text + ",s_adres=" + s_adres.Text
        + ",s_sehir=" + s_sehir.Text + " where s_id=" + "1" + """;
        cmd.ExecuteNonQuery();
        con.Close();
        griddoldur();
        MessageBox.Show("Kayıt İşlemi Gerçekleşti");
    }
    catch (Exception)
    {
        MessageBox.Show("Bir Hata Oluştı");
    }
}

private void button4_Click(object sender, EventArgs e)
{
    con.Open();
    OleDbCommand cmd = new OleDbCommand("Select s_sk, s_ad,
s_tel,s_faks,s_mail,s_web,s_adres,s_sehir from sirket where s_id=1", con);
    OleDbDataReader dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        s_sk.Text = dr["s_sk"].ToString();
        s_ad.Text = dr["s_ad"].ToString();
        s_tel.Text = dr["s_tel"].ToString();
        s_faks.Text = dr["s_faks"].ToString();
    }
}

```

```

        s_mail.Text = dr["s_mail"].ToString();
        s_web.Text = dr["s_web"].ToString();
        s_adres.Text = dr["s_adres"].ToString();
        s_sehir.Text = dr["s_sehir"].ToString();
    }
    con.Close();
    griddoldur();

```

```

    }
    private void button5_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void BtnTemizle_MouseHover(object sender, EventArgs e)
    {
        BtnTemizle.ForeColor = Color.Red;
    }
    private void BtnTemizle_MouseLeave(object sender, EventArgs e)
    {
        BtnTemizle.ForeColor = Color.Black;
    }
    private void BtnKaydet_MouseHover(object sender, EventArgs e)
    {
        BtnKaydet.ForeColor = Color.Red;
    }
    private void BtnKaydet_MouseLeave(object sender, EventArgs e)
    {
        BtnKaydet.ForeColor = Color.Black;
    }
    private void button4_MouseHover(object sender, EventArgs e)
    {
        button4.ForeColor = Color.Red;
    }
    private void button4_MouseLeave(object sender, EventArgs e)
    {
        button4.ForeColor = Color.Black;
    }
    private void button5_MouseHover(object sender, EventArgs e)
    {
        button5.ForeColor = Color.Red;
    }
    private void button5_MouseLeave(object sender, EventArgs e)
    {

```

```

        button5.ForeColor = Color.Black;
    }
    private void BtnTemizle_Click(object sender, EventArgs e)
    {
        foreach (Control item in this.Controls)
        {
            if (item is TextBox)
            {
                TextBox tbox = (TextBox)item;

                tbox.Clear();
            }
        }
    }
}

```

10. Hesap Oluştur(Hesapolustur)

Bu ekran ile hesap oluşturabilirsiniz ve hesabınızın yetkisini belirleyebilirsiniz.

Not: Sadece yönetici yetkisine sahip kişiler görebilir.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class hesapolustur : Form
    {
        OleDbConnection con;

        OleDbCommand cmd;
        public hesapolustur()
        {
            InitializeComponent();
        }
    }
}

```

```

}
private void hesapOlustur_Load(object sender, EventArgs e)
{
    this.Size = new Size(540, 300);
    this.MaximumSize = this.Size;
    this.MinimumSize = this.Size;
}
private void button1_Click(object sender, EventArgs e)
{
    string veri;
    if (comboBox1.Text == "Yönetici")
    {
        veri = "Yes";
    }
    else
    {
        veri = "No";
    }
    try
    {
        if (textBox1.Text == "" && textBox2.Text == "" && textBox3.Text == ""
        && textBox4.Text == "" && textBox5.Text == "" && textBox6.Text == "" &&
        comboBox1.Text == "")
        {
            MessageBox.Show("Alanlar boş bırakılamaz !!");
        }
        else
        {
            con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "INSERT INTO
yonetim(ym_kadi,ym_ksif,ym_ad,ym_soyad,ym_tel,ym_adres,ym_yetki) VALUES
('" + textBox1.Text + "','" + textBox2.Text + "','" + textBox3.Text + "','" +
textBox4.Text + "','" + textBox5.Text + "','" + textBox6.Text + "','" + veri + "')";
            cmd.ExecuteNonQuery();
            label8.BackColor = Color.DarkGreen;
            label8.Text = textBox3.Text + " " + textBox4.Text + " Hesabınız
Oluşturuldu.\n" + "Hesabınızın Yetki Durumu : " + comboBox1.Text;
            textBox1.Enabled = false;
            textBox2.Enabled = false;
            textBox3.Enabled = false;
            textBox4.Enabled = false;
            textBox5.Enabled = false;
            textBox6.Enabled = false;
            comboBox1.Enabled = false;
            button2.Visible = true;

```

```

        button1.Enabled = false;
        con.Close();
    }
    catch (Exception)
    {
        label8.BackColor = Color.DarkRed;
        label8.Text = "İşlem yapılırken bir hata meydana geldi.\nAlanları Kontrol ediniz
!";
    }
}

private void button2_Click(object sender, EventArgs e)
{
    label8.Text = "";
    button1.Enabled = true;
    button2.Visible = false;
    textBox1.Enabled = true;
    textBox2.Enabled = true;
    textBox3.Enabled = true;
    textBox4.Enabled = true;
    textBox5.Enabled = true;
    textBox6.Enabled = true;
    comboBox1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    textBox5.Text = "";
    textBox6.Text = "";
    comboBox1.Text = "";

}

private void textBox5_KeyPress(object sender, KeyPressEventArgs e)
{
    e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);
}

private void textBox5_TextChanged(object sender, EventArgs e)
{
}
}
}
}

```

11. Hesap Yetkilendirme (Hesapyetkilendir)

Oluşturulmuş olan bir kullanıcının yetkisini burdan değiştirebilirsiniz.

Not: Sadece yönetici yetkisine sahip kişiler görebilir.

Kodlar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
namespace RecX_Market_Otomasyonu
{
    public partial class hesapyetkilendir : Form
    {
        OleDbConnection con;
        OleDbCommand cmd;
        void kullaniciadi()
        {
            comboBox1.Items.Clear();
            con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
            cmd = new OleDbCommand();
            OleDbDataReader dr;
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "SELECT * FROM yonetim";
            dr = cmd.ExecuteReader();
            while (dr.Read())
            {
                comboBox1.Items.Add(dr["ym_kadi"]);
            }
            con.Close();
        }
        public hesapyetkilendir()
        {
            InitializeComponent();
        }
        private void hesapyetkilendir_Load(object sender, EventArgs e)
        {
            kullaniciadi();
            this.Size = new Size(361, 224);
            this.MaximumSize = this.Size;
            this.MinimumSize = this.Size;
        }
    }
}
```

```

private void button1_Click(object sender, EventArgs e)
{
    string veri;
    if (comboBox2.Text == "Yönetici")
    {
        veri = "Yes";
    }
    else
    {
        veri = "No";
    }
    try
    {
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "update yonetim set ym_yetki = " + veri + " where
ym_kadi = '" + comboBox1.Text + "'";
        cmd.ExecuteNonQuery();
        label3.Text = "Yetkilendirme işlemi başarıyla gerçekleştirildi\nKullanıcı :
"+comboBox1.Text+" Yetki : "+comboBox2.Text;
        comboBox1.Enabled = false;
        comboBox2.Enabled = false;
        button2.Visible = true;
        button1.Visible = false;
        con.Close();
    }
    catch (Exception)
    {
        label3.Text = "Bir hata meydana geldi";
    }
}
private void button2_Click(object sender, EventArgs e)
{
    comboBox1.Text = "";
    comboBox2.Text = "";
    comboBox1.Enabled = true;
    comboBox2.Enabled = true;
    button1.Visible = true;
    button2.Visible = false;
    label3.Text = "";
}
}
}

```

12. Hesap Şifre Değiştir (Yonetimhesap)

Oluşturduğunuz hesapların şifrelerini bu alandan değiştirebilirsiniz.

Not: Sadece yönetici yetkisine sahip kişiler görebilir.

Kodlar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class yonetimhesap : Form
    {
        OleDbConnection con;
        OleDbCommand cmd;
        public yonetimhesap()
        {
            InitializeComponent();
        }
        void kullaniciadi()
        {
            comboBox1.Items.Clear();
            con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
            cmd = new OleDbCommand();
            OleDbDataReader dr;
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "SELECT * FROM yonetim";
            dr = cmd.ExecuteReader();

            while (dr.Read())
            {
                comboBox1.Items.Add(dr["ym_kadi"]);
            }
            con.Close();
        }
        private void yonetimhesap_Load(object sender, EventArgs e)
        {
            kullaniciadi();
        }
        private void button1_Click(object sender, EventArgs e)
```

```

    {
        try
        {
            con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "UPDATE yonetim SET ym_ksif=" + textBox2.Text
+ " WHERE ym_kadi=" + comboBox1.Text + """;
            cmd.ExecuteNonQuery();
            label3.Text = "Şifreniz başarıyla güncellendi";
            con.Close();
        }
        catch (Exception)
        {
            con.Close();
            label3.Text = "Şifreniz güncelleştirilirken bir hata meydana geldi !";
        }
    }
}

```

13. Hesap Silme(Hesapsil)

Hesap silme işlemini buradan gerçekleştirebilirsiniz.

Not: Sadece yönetici yetkisine sahip kişiler görebilir.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class hesapsil : Form
    {
        OleDbConnection con;
        OleDbDataAdapter da;
    }
}

```

```

OleDbCommand cmd;

public hesapsil()
{
    InitializeComponent();
}
void kullaniciadi()
{
    comboBox1.Items.Clear();
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    cmd = new OleDbCommand();
    OleDbDataReader dr;
    con.Open();
    cmd.Connection = con;
    cmd.CommandText = "SELECT * FROM yonetim";
    dr = cmd.ExecuteReader();
    while (dr.Read())
    {
        comboBox1.Items.Add(dr["ym_kadi"]);
    }
    con.Close();
}
private void hesapsil_Load(object sender, EventArgs e)
{
    kullaniciadi();
    this.Size = new Size(323, 137);
    this.MaximumSize = this.Size;
    this.MinimumSize = this.Size;
}

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
        da = new OleDbDataAdapter("SElect *from yonetim", con);
        cmd = new OleDbCommand();
        con.Open();
        cmd.Connection = con;
        cmd.CommandText = "delete from yonetim where ym_kadi = '" +
comboBox1.Text + "'";
        cmd.ExecuteNonQuery();
        MessageBox.Show("Silme İşlemi Gerçekleşti");
        this.Close();
    }
}

```

```

catch (Exception)
{
    MessageBox.Show("Bir Hata Meydana Geldi");
}
}
}

```

14. Duyuru İşlemleri (Duyuruislemler)

Duyuruları buradan oluşturabilirsiniz. Varolan duyuruları silebilirsiniz.

Not: Sadece yönetici yetkisine sahip kişiler görebilir.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class duyuruislemler : Form
    {
        OleDbConnection con;
        OleDbDataAdapter da;
        OleDbCommand cmd;
        DataSet ds;
        public duyuruislemler()
        {
            InitializeComponent();
        }
        void duyurugetir()
        {
            comboBox1.Items.Clear();

            con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
            cmd = new OleDbCommand();
            OleDbDataReader dr;
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "SELECT * FROM duyurular";

```

```

dr = cmd.ExecuteReader();

while (dr.Read())
{
    comboBox1.Items.Add(dr["duyuru_text"]);
}
con.Close();
}
void griddoldur()
{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SElect * from duyurular", con);
    ds = new DataSet();
    con.Open();
    duyurugetir();
    con.Close();
}
private void button1_Click(object sender, EventArgs e)
{
    {
        try
        {
            griddoldur();
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "insert into duyurular(duyuru_text) values
("+textBox1.Text+")";
            cmd.ExecuteNonQuery();
            con.Close();
            label1.Text = textBox1.Text + " Duyurusu Oluşturuldu.";
            griddoldur();
        }
        catch (Exception hata)
        {
            label1.Text = textBox1.Text + "Oluşturulurken bir hata meydana geldi\n"
+ hata;
        }
    }
}

private void button2_Click(object sender, EventArgs e)
{
    {
        try
        {
            cmd = new OleDbCommand();
            con.Open();
            cmd.Connection = con;
            cmd.CommandText = "delete from duyurular where duyuru_text=" +
comboBox1.Text + """;
            cmd.ExecuteNonQuery();
            con.Close();
        }
        catch (Exception hata)
        {
            label1.Text = textBox1.Text + "Silinirken bir hata meydana geldi\n"
+ hata;
        }
    }
}

```

```

        label1.Text = comboBox1.Text + " [Duyurusunu siliyorsunuz]";

        griddoldur();
    }
    catch (Exception silhat)
    {

        label1.Text = "Silme İşleminde Bir Hata Meydana Geldi\n" + silhat;
    } }
    private void duyuruisleml_Load(object sender, EventArgs e)
    {
        griddoldur();
        duyurugetir();
    } } }

```

15. Satış İşlemi (Perakendesatisgirisi)

Satış işlemleri buradan yapılmaktadır. Barkod seçimi sırasında barkod işaretine tıklayarak açılan formdan da barkod seçebilir ya da yanında bulunan metin kutucuğuna barkod numarasını girerek de ürüne ulaşabilirsiniz. Barkod seçiminden sonra adet seçimi yapıldıktan sonra aktif olan butonlar ile müşteri ismini seçebilir ya da isimsiz müşteri olarak satış yapabilirsiniz.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.IO;
using System.Collections;
using Microsoft.VisualBasic;

namespace RecX_Market_Otomasyonu
{
    public partial class Perakendesatisgirisi : Form
    {
        public static OleDbConnection Con;
        private Perakendesatissecimi Form2 = new Perakendesatissecimi();
    }
}

```



```

private ArrayList ticket_products = new ArrayList();
private ArrayList ticket_counts = new ArrayList();
private ArrayList ticket_prices = new ArrayList();
private ArrayList products = new ArrayList();
private ArrayList types = new ArrayList();
OleDbConnection con;
OleDbDataAdapter da;
OleDbCommand cmd;
DataSet ds;
void griddoldur()
{
    con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    da = new OleDbDataAdapter("SELECT TOP 1 * FROM satislar ORDER BY
id DESC;", con);
    ds = new DataSet();
    con.Open();
    da.Fill(ds, "satislar");
    dataGridView1.DataSource = ds.Tables["satislar"];
    con.Close();
}
StringFormat strFormat;
ArrayList arrColumnLefts = new ArrayList();
ArrayList arrColumnWidths = new ArrayList();
int iCellHeight = 0;
int iTotalWidth = 0;
int iRow = 0;
bool bFirstPage = false;
bool bNewPage = false;
int iHeaderHeight = 0;
private OleDbDataReader product;
void yenile()
{
    Con = new OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    Con.Open();
    Con.Close();

}
private string BackupLoc;
public Perakendesatisgirisi()
{
    InitializeComponent();
}
private void test_Load(object sender, EventArgs e)
{
    this.Size = new Size(473, 529);
    griddoldur();
    this.MaximumSize = this.Size;
}

```

```

this.MinimumSize = this.Size;
dataGridView1.AllowUserToAddRows = false;
dataGridView1.Columns[0].HeaderText = "Satış ID";
dataGridView1.Columns[1].HeaderText = "Tarih";
dataGridView1.Columns[2].HeaderText = "Alınan Ücret (TL)";
dataGridView1.Columns[3].HeaderText = "Müşteri Adı";
try
{
    Con = new OleDbConnection(@"Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    Con.Open();
}
catch (Exception err)
{
    MessageBox.Show(err.Message, "Market Otomasyonu - Veritabanı
Bağlantı Hatası", MessageBoxButtons.OK, MessageBoxIcon.Error);
    Application.Exit();
}
Directory.CreateDirectory(Directory.GetCurrentDirectory()
@"\Data\Backups");
Directory.CreateDirectory(Directory.GetCurrentDirectory()
@"\Data\Images");
Backup();
GetTypes();
GetProducts();
}
private void Backup()
{
    bool BackupEnabled = false;
    string Query = "select * from settings";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    OleDbDataReader result = cmd.ExecuteReader();
    while (result.Read())
    {
        switch (result["id"].ToString())
        {
            case "auto_daily_backup":
                if (result["val"].ToString() == "1")
                {
                    BackupEnabled = true;
                }
                else
                {
                    BackupEnabled = false;
                }
                break;
            case "backup_loc":
                BackupLoc = result["val"].ToString();

                if (!Directory.Exists(BackupLoc))

```

```

        {
            BackupLoc = Directory.GetCurrentDirectory() +
@"\Data\Backups";
        }

        break;
    }
    else
    {
        string BackupFileLoc = BackupLoc + @"\" +
DateTime.Now.ToString("yyyy-MM-dd") + ".accdb";
        if (!File.Exists(BackupFileLoc))
        {
            File.Copy(@"Data\MarketData.accdb", BackupFileLoc);
        }
    }
}

private void GetTypes()
{
    types = new ArrayList();
    string Query = "select * from urunler order by urun_id desc";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    OleDbDataReader result = cmd.ExecuteReader();
    while (result.Read())
    {
        types.Add(Convert.ToInt32(result["urun_id"]));
    }
}

private void GetProducts()
{
    products = new ArrayList();
    string Query = "select * from urunler order by urun_id desc";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    OleDbDataReader result = cmd.ExecuteReader();
}

private void DeleteTypeImages(int type_id)
{
    string img_loc;
    string Query = "select id from urunler where kategori_id=@kategori_id";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    cmd.Parameters.AddWithValue("@kategori_id", type_id);
    OleDbDataReader result = cmd.ExecuteReader();
    while (result.Read())
    {
        img_loc = @"Data\Images\" + result["urun_id"].ToString();
        if (File.Exists(img_loc))
        {
            File.Delete(img_loc);
        }
    }
}

private string GetTypeNames(int type_id)
{
    string Query = "select kategori_adi from kategori where
kategori_id=@kategori_id";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    cmd.Parameters.AddWithValue("@kategori_id", type_id);
    OleDbDataReader result = cmd.ExecuteReader();
    if (result.Read())
    {

```

```

        return result["kategori_adi"].ToString();    }
    return "";
}
private string GetProductName(int product_id)
{
    string Query = "select urun_adi from urunler where urun_id=@urun_id";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    cmd.Parameters.AddWithValue("@urun_id", product_id);
    OleDbDataReader result = cmd.ExecuteReader();
    result.Read();
    return result["urun_adi"].ToString();
}
private int GetProductDailySale(int product_id, int ticket_id)    {
    string Query = "select count(*) as total from satislar where urun_id=@urun_id
and ticket_id=@ticket_id";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    cmd.Parameters.AddWithValue("@urun_id", product_id);
    cmd.Parameters.AddWithValue("@ticket_id", ticket_id);
    OleDbDataReader result = cmd.ExecuteReader();
    result.Read();
    return Convert.ToInt32(result["total"]);
}
private string Daily_Total()    {
    DateTime date = new DateTime(monthCalendar1.SelectionRange.Start.Year,
monthCalendar1.SelectionRange.Start.Month,
monthCalendar1.SelectionRange.Start.Day);
    string Query = "select sum(price) as daily_total from satislar where
sale_date=@sale_date";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    cmd.Parameters.AddWithValue("@sale_date", date);
    OleDbDataReader result = cmd.ExecuteReader();
    while (result.Read())    {
        string value = result["daily_total"].ToString();
        if (!String.IsNullOrEmpty(value))    {
            return value;
        }
    }
    return "0,00";
}
private string Monthly_Total()
{
    DateTime start_date = new
DateTime(monthCalendar1.SelectionRange.Start.Year,
monthCalendar1.SelectionRange.Start.Month, 1);
    DateTime end_date = new
DateTime(monthCalendar1.SelectionRange.Start.Year,
monthCalendar1.SelectionRange.Start.Month, 1).AddMonths(1);
    string Query = "select sum(price) as monthly_total from satislar where
sale_date>=@sale_date_start and sale_date<@sale_date_end";
    OleDbCommand cmd = new OleDbCommand(Query, Con);

```

```

cmd.Parameters.AddWithValue("@sale_date_start", start_date);
cmd.Parameters.AddWithValue("@sale_date_end", end_date);
OleDbDataReader result = cmd.ExecuteReader();

while (result.Read())
{
    string value = result["monthly_total"].ToString();

    if (!String.IsNullOrEmpty(value))
    {
        return value;
    }
}
return "0,00";
}

private int GetNextBarcode()
{
    string Query = "select max(urun_barkod) as biggest from urunler";
    OleDbCommand cmd = new OleDbCommand(Query, Con);
    OleDbDataReader result = cmd.ExecuteReader();
    result.Read();
    if (!String.IsNullOrEmpty(result["biggest"].ToString()))
    {
        return Convert.ToInt32(result["biggest"]) + 1;
    }
    return 0;
}

private int StockLoss(int product_id)
{
    int loss = 0;
    for (int i = 0; i < listBox1.Items.Count; i++)
    {
        if (Convert.ToInt32(ticket_products[i]) == product_id)
        {
            loss += Convert.ToInt32(ticket_counts[i]);
        }
    }
    return loss;
}

private void textBox1_TextChanged(object sender, EventArgs e)
{
    int barcode;
    if (int.TryParse(textBox1.Text, out barcode))
    {
        string Query = "select * from urunler where urun_barkod=@urun_barkod";
        OleDbCommand cmd = new OleDbCommand(Query, Con);
        cmd.Parameters.AddWithValue("@urun_barkod", barcode);
        product = cmd.ExecuteReader();
        if (product.Read())
        {
            label7.Text = product["urun_barkod"].ToString();
            label6.Text = product["urun_adi"].ToString();
            label5.Text = product["urun_sfiyati"].ToString() + " TL";
        }
    }
}

```

```

        int type_id;
        if (int.TryParse(product["kategori_id"].ToString(), out type_id))
        {
            label8.Text = GetTypeName(Convert.ToInt32(type_id));
        }
        numericUpDown1.Maximum = Convert.ToInt32(product["urun_adedi"]);
- StockLoss(Convert.ToInt32(product["urun_id"]));
        if (numericUpDown1.Maximum > 0)
        {
            numericUpDown1.Value = 1;
        }
        return;
    }
    label7.Text = null;
    label8.Text = null;
    label6.Text = null;
    label5.Text = null;
    numericUpDown1.Maximum = 0;
    numericUpDown1.Value = 0;
}
private void numericUpDown1_ValueChanged(object sender, EventArgs e)
{
    if (numericUpDown1.Value == 0)
    {
        button8.Enabled = false;
    }
    else
    {
        button8.Enabled = true;
    }
}

private void button9_Click(object sender, EventArgs e)
{
    Form2.ShowDialog();

    if (!String.IsNullOrEmpty(Form2.sel_barcode))
    {
        textBox1.Text = Form2.sel_barcode;
    }
}

private void button8_Click(object sender, EventArgs e)
{
    ticket_products.Add(Convert.ToInt32(product["urun_id"]));
    ticket_prices.Add(Convert.ToDecimal(product["urun_sfiyati"]));
    ticket_counts.Add(Convert.ToInt32(numericUpDown1.Value));
    listBox1.Items.Add(numericUpDown1.Value + " x " + product["urun_adi"]);
    textBox1.Text = "";
    listBox1.SelectedIndex = listBox1.Items.Count - 1;
}
private void button7_Click(object sender, EventArgs e)
{
    ticket_products.Remove(ticket_products[listBox1.SelectedIndex]);
    ticket_prices.Remove(ticket_prices[listBox1.SelectedIndex]);
    ticket_counts.Remove(ticket_counts[listBox1.SelectedIndex]);
    listBox1.Items.Remove(listBox1.SelectedItem);
}

```

```

private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    decimal amount = 0;
    if (listBox1.SelectedIndex == -1)
    {
        button2.Enabled = false;
        button5.Enabled = false;
        button6.Enabled = false;
        button7.Enabled = false;
    }
    else
    {
        button2.Enabled = true;
        button5.Enabled = true;
        button6.Enabled = true;
        button7.Enabled = true;
    }
    for (int i = 0; i < ticket_prices.Count; i++)
    {
        amount += Convert.ToDecimal(ticket_prices[i]) *
Convert.ToInt32(ticket_counts[i]);
    }
    label24.Text = amount.ToString();
}

private void button5_Click(object sender, EventArgs e) {
    try {
        string Query;
        OleDbCommand cmd;
        int product_id, count;
        for (int i = 0; i < listBox1.Items.Count; i++)
        {
            product_id = Convert.ToInt32(ticket_products[i]);
            count = Convert.ToInt32(ticket_counts[i]);
            Query = "update urunler set urun_adedi=(urun_adedi - @count) where
urun_id=@urun_id";
            cmd = new OleDbCommand(Query, Con);
            cmd.Parameters.AddWithValue("@count", count);
            cmd.Parameters.AddWithValue("@urun_id", product_id);
            cmd.ExecuteNonQuery();
        }
        string mus_adi = mus_ad.Text;
        if(mus_ad.Text == "") {
            mus_adi = "Tanımsız Müşteri";
        }
        Query = "insert into satislar (urun_sfiyati,mus_ad)" +
        "Values(@urun_sfiyati,'" + mus_adi + "')";
        cmd = new OleDbCommand(Query, Con);
        cmd.Parameters.AddWithValue("@urun_sfiyati", label24.Text);
    }
}

```

```

        cmd.ExecuteNonQuery();
        button6.PerformClick();
        MessageBox.Show("Satış Başarıyla Tamamlandı. Fiş Kes diyerek fişinizi
alabilirsiniz.");
        mus_ad.Clear();
        button4.Enabled = true;
    }
    catch (Exception)
    {

        MessageBox.Show("Satış Tamamlanamadı", "Dikkat",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
private void button6_Click(object sender, EventArgs e)
{
    ticket_products = new ArrayList();
    ticket_prices = new ArrayList();
    ticket_counts = new ArrayList();
    listBox1.SelectedIndex = -1;
    listBox1.Items.Clear();
}
private void button2_Click(object sender, EventArgs e)
{
    perakendesatismusteri permus = new perakendesatismusteri();
    permus.ShowDialog();
    if (!String.IsNullOrEmpty(permus.musteri))
    {
        mus_ad.Text = permus.musteri;
    }
}
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        double sonuc;
        sonuc = Double.Parse(textBox2.Text) - Double.Parse(label24.Text);
        if (sonuc < 0)
        {
            label11.ForeColor = Color.Red;
        }
        else
        {
            label11.ForeColor = Color.Black;
        }
        label11.Text = sonuc.ToString() + " TL Para üstü veriniz...";
    }
    catch (Exception)
    {
        label11.Text = "Hatalı İşlem !!";
    }
}
private void textBox2_KeyPress(object sender, KeyPressEventArgs e)
{
    if ((int)e.KeyChar >= 48 && (int)e.KeyChar <= 57)
    {
        e.Handled = false; //eğer rakamsa yazdır.
    }

    else if ((int)e.KeyChar == 44)
    {
        e.Handled = false; //eğer basılan tuş virgül ise yazdır.
    }
    else if ((int)e.KeyChar == 08)
    {

```



```

        e.Handled = false;//eğer basılan tuş backspace ise yazdır.    }
    else    {
        e.Handled = true;//bunların dışındaysa hiçbirisini yazdırma    }    }
private    void    printDocument1_PrintPage(object    sender,
System.Drawing.Printing.PrintPageEventArgs e)
{
    try    {
        int iLeftMargin = e.MarginBounds.Left;
        int iTopMargin = e.MarginBounds.Top;
        bool bMorePagesToPrint = false;
        int iTmpWidth = 0;
        bFirstPage = true;
        if (bFirstPage)    {
            foreach (DataGridViewColumn GridCol in dataGridView1.Columns)
            {
                iTmpWidth = (int)(Math.Floor(((double)((double)GridCol.Width /
                    (double)iTotalWidth * (double)iTotalWidth *
                    ((double)e.MarginBounds.Width / (double)iTotalWidth))));
                iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
                    GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;
                arrColumnLefts.Add(iLeftMargin);
                arrColumnWidths.Add(iTmpWidth);
                iLeftMargin += iTmpWidth;
            }
        }
        while (iRow <= dataGridView1.Rows.Count - 1)    {
            DataGridViewRow GridRow = dataGridView1.Rows[iRow];
            iCellHeight = GridRow.Height + 5;
            int iCount = 0;
            if (iTopMargin + iCellHeight >= e.MarginBounds.Height +
e.MarginBounds.Top)    {
                bNewPage = true;
                bFirstPage = false;
                bMorePagesToPrint = true;
                break;
            }
            else
            {
                if (bNewPage)    {
                    e.Graphics.DrawString("Satış    Çıktısı",    new
Font(dataGridView1.Font, FontStyle.Bold),
Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
e.Graphics.MeasureString("Satış    Çıktısı",    new
Font(dataGridView1.Font,
FontStyle.Bold), e.MarginBounds.Width).Height - 13);
                    String strDate = DateTime.Now.ToLongDateString() + " " +
DateTime.Now.ToShortTimeString();
                    e.Graphics.DrawString(strDate,    new    Font(dataGridView1.Font,
FontStyle.Bold),

```

```

        Brushes.Black,          e.MarginBounds.Left          +
(e.MarginBounds.Width -
    e.Graphics.MeasureString(strDate,          new
Font(dataGridView1.Font,
        FontStyle.Bold),          e.MarginBounds.Width).Width),
e.MarginBounds.Top -
    e.Graphics.MeasureString("Satış Çıktısı", new Font(new
Font(dataGridView1.Font,
    FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
    iTopMargin = e.MarginBounds.Top;
    foreach (DataGridViewColumn GridCol in
dataGridView1.Columns)
    {
        e.Graphics.FillRectangle(new SolidBrush(Color.LightGray),
            new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
                (int)arrColumnWidths[iCount], iHeaderHeight));
        e.Graphics.DrawRectangle(Pens.Black,
            new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
                (int)arrColumnWidths[iCount], iHeaderHeight));
        e.Graphics.DrawString(GridCol.HeaderText,
GridCol.InheritedStyle.Font,
            new SolidBrush(GridCol.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], iTopMargin,
                (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
        iCount++;
    }
    bNewPage = false;
    iTopMargin += iHeaderHeight;
}
iCount = 0;
foreach (DataGridViewCell Cel in GridRow.Cells)
{
    if (Cel.Value != null)
    {
        e.Graphics.DrawString(Cel.Value.ToString(),
Cel.InheritedStyle.Font,
            new SolidBrush(Cel.InheritedStyle.ForeColor),
            new RectangleF((int)arrColumnLefts[iCount], (float)iTopMargin,
                (int)arrColumnWidths[iCount], (float)iCellHeight),
strFormat);
    }
    e.Graphics.DrawRectangle(Pens.Black,          new
Rectangle((int)arrColumnLefts[iCount],
        iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));

    iCount++;
}
iRow++;

```

```

        iTopMargin += iCellHeight;
    }
    if (bMorePagesToPrint)
        e.HasMorePages = true;
    else
        e.HasMorePages = false;
    }
    catch (Exception exc)
    {
        MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
    } }

    private void printDocument1_BeginPrint(object sender,
    System.Drawing.Printing.PrintEventArgs e) {
        try {
            strFormat = new StringFormat();
            strFormat.Alignment = StringAlignment.Near;
            strFormat.LineAlignment = StringAlignment.Center;
            strFormat.Trimming = StringTrimming.EllipsisCharacter;
            arrColumnLefts.Clear();
            arrColumnWidths.Clear();
            iCellHeight = 0;
            iRow = 0;
            bFirstPage = true;
            bNewPage = true;
            iTotalWidth = 0;
            foreach (DataGridViewColumn dgvGridCol in dataGridView1.Columns)
            {
                iTotalWidth += dgvGridCol.Width;
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
        } }

    private void button4_Click(object sender, EventArgs e)
    {
        griddoldur();
        PrintPreviewDialog onizleme = new PrintPreviewDialog();
        onizleme.Document = printDocument1;
        onizleme.ShowDialog();
    } } }

```

16. Perakende Satış Ürün Seçimi(Perakendesatissecimi)

Barkod alanına barkod numarasını yazmadan bu alandan ürün seçimi yapabilirsiniz.

Kodlar:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.OleDb;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Collections;

namespace RecX_Market_Otomasyonu
{
    public partial class Perakendesatissecimi : Form
    {
        private ArrayList type_ids = new ArrayList();
        private ArrayList product_ids = new ArrayList();

        public string sel_barcode { get; set; }
        public Perakendesatissecimi()
        {
            InitializeComponent();
        }
        private void Form2_Load(object sender, EventArgs e)
        {
            listBox2.ClearSelected();
            listBox2.Items.Clear();
            listBox1.Items.Clear();
            this.Icon = Icon.ExtractAssociatedIcon(Application.ExecutablePath);
            ListTypes();
            listBox1.SelectedIndex = 0;
        }
        private void ListTypes()
        {
            type_ids.Add(0);
            listBox1.Items.Add("« Tümü »");
            string Query = "select * from kategori";
            OleDbCommand cmd = new OleDbCommand(Query,
Perakendesatisgirisi.Con);
            OleDbDataReader result = cmd.ExecuteReader();
            while (result.Read())
            {
                type_ids.Add(Convert.ToInt32(result["kategori_id"]));
                listBox1.Items.Add(result["kategori_adi"].ToString());
            }
        }
    }
}
```

```

private void GetBarcode()
{
    if (!String.IsNullOrEmpty(textBox1.Text))
    {
        this.sel_barcode = textBox1.Text;
        this.Close();
    }
}
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    string Query;
    int type_id = Convert.ToInt32(type_ids[listBox1.SelectedIndex]);
    product_ids = new ArrayList();
    listBox2.ClearSelected();
    listBox2.Items.Clear();
    if (type_id == 0)
    {
        Query = "select * from urunler";
    }
    else
    {
        Query = "select * from urunler where kategori_id=@kategori_id";
    }
    OleDbCommand cmd = new OleDbCommand(Query,
Perakendesatisgirisi.Con);
    cmd.Parameters.AddWithValue("@kategori_id", type_id);
    OleDbDataReader result = cmd.ExecuteReader();
    while (result.Read())
    {
        product_ids.Add(Convert.ToInt32(result["urun_id"]));
        if (Convert.ToInt32(result["urun_adedi"]) == 0)
        {
            listBox2.Items.Add("(Stokta Yok) " + result["urun_adi"].ToString());
        }
        else
        {
            listBox2.Items.Add(result["urun_adi"].ToString());
        }
    }
}

private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox2.SelectedItems.Count == 0)
    {
        textBox1.Text = null;
        button1.Enabled = false;
    }
    else
    {
        textBox1.Text = null;
    }
}

```

```

        button1.Enabled = true;
        int product_id = Convert.ToInt32(product_ids[listBox2.SelectedIndex]);
        string Query = "select urun_barkod from urunler where
urun_id=@urun_id";
        OleDbCommand cmd = new OleDbCommand(Query,
Perakendesatisgirisi.Con);
        cmd.Parameters.AddWithValue("@urun_id", product_id);
        OleDbDataReader result = cmd.ExecuteReader();
        result.Read();
        textBox1.Text = result["urun_barkod"].ToString();
    } }
    private void button1_Click(object sender, EventArgs e)
    {
        GetBarcode();
    }
    private void listBox2_DoubleClick(object sender, EventArgs e)
    {
        GetBarcode();
    } } }

```

17. Satış İşlemi Müşteri Seçimi (Perakendesatismusteri)

Bu form ile sisteme kayıtlı bir müşteriniz varsa eğer müşteri ismi seçerek müşteri adına satış işlemi gerçekleştirebilirsiniz.

Kodlar:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;

namespace RecX_Market_Otomasyonu
{
    public partial class perakendesatismusteri : Form
    {
        public perakendesatismusteri()
        {
            InitializeComponent();
        }
        public string musteri { get; set; }
    }
}

```

```

private void Musteri()
{
    if (!String.IsNullOrEmpty(mus_id.Text))
    {
        this.musteri = mus_id.Text;
        this.Close();
    }
}

private void perakendesatismusteri_Load(object sender, EventArgs e)
{
    OleDbConnection baglan = new
OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    baglan.Open();
    OleDbCommand command = new OleDbCommand();
    command.Connection = baglan;
    command.CommandText = "select* from musteriler";
    OleDbDataReader reader = command.ExecuteReader();
    while (reader.Read()) //while'i tüm öğeleri seçmek için kullandık.
    {
        listBox1.Items.Add(reader["mus_id"]); //verileri listboxa ekliyoruz .
    }
    try
    {
        listBox1.SelectedIndex = 0;
    }
    catch (Exception)
    {
        MessageBox.Show("Kayıtlı müşteri bulunamadı veya seçim işleminde bir hata
meydana geldi");
    }
    baglan.Close(); // en son bağlantıyı kapatıyoruz.
}

private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    OleDbConnection baglan = new
OleDbConnection("Provider=Microsoft.ACE.Oledb.12.0;Data
Source=Data/MarketData.accdb");
    baglan.Open();
    OleDbCommand command = new OleDbCommand();
    command.Connection = baglan;
    command.CommandText = "select * from musteriler where
mus_id="+listBox1.Text;
    OleDbDataReader reader = command.ExecuteReader();
    if (reader.Read())
    {
        mus_id.Text = reader["mus_ad"].ToString();
    }
    else

```

```
{
    mus_id.Text = "veri cekilemedi";
}
baglan.Close(); // en son bağlantıyı kapatıyoruz.
}
private void button1_Click(object sender, EventArgs e)
{
    Musteri();
} } }
```


KAYNAKLAR

ACCESS&SQL: <http://www.access-sql.com/>

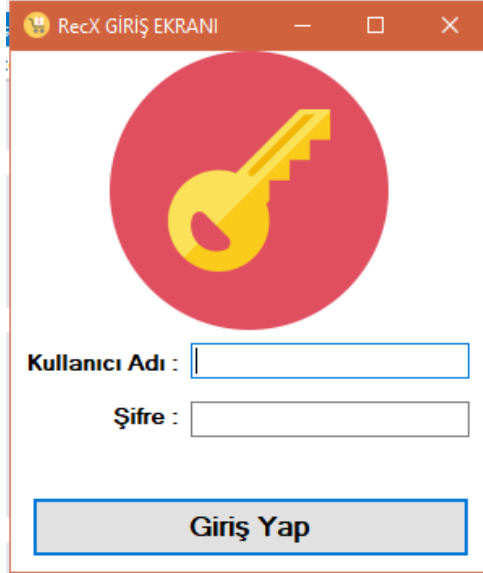
Ahmet Can Sever: <http://www.ahmetcansever.com/programlama/c-access-veri-tabani-baglantisiselect-insert-update-delete-ornek-uygulama/>

Hikmet Okutmuş: <http://www.hikmetokumus.com/makale/11/datagridview%27-dabulunan-satirlari-yazdirmak>


Microsoft Social : <https://social.msdn.microsoft.com>

EKLER

Giriş Ekranı



RecX GİRİŞ EKRANI

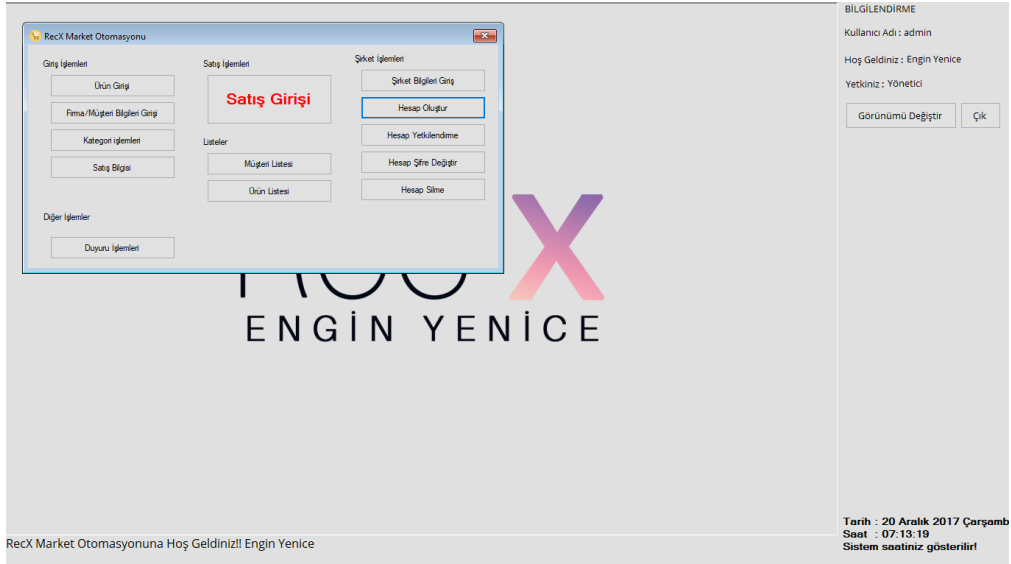


Kullanıcı Adı :

Şifre :

Giriş Yap

Yönetici Hesap Ana Yüz



RecX Market Otomasyonu

Giriş İşlemleri

- Ürün Giriş
- Firma/Müşteri Bilgileri Giriş
- Kategori İşlemleri
- Satış Bilgi

Satış İşlemleri

- Satış Girişi**
- Listeler
 - Müşteri Listesi
 - Ürün Listesi

Şirket İşlemleri

- Şirket Bilgileri Giriş
- Hesap Oluştur
- Hesap Yetkilendirme
- Hesap Şifre Değiştir
- Hesap Silme

Diğer İşlemler

- Duyuru İşlemleri

BİLGİLENDİRME

Kullanıcı Adı : admin

Hoş Geldiniz : Engin Yenice

Yetkiniz : Yönetici

Görünümü Değiştir Çık

ENGİN YENİCE

RecX Market Otomasyonuna Hoş Geldiniz!! Engin Yenice

Tarih : 28 Aralık 2017 Çarşamba
Saat : 07:13:19
Sistem saatiniz gösteriliyor

Kasiyer Hesap Ana Yüz

RecX Market Otomasyonu

Giriş İşlemleri

Ürün Girişi

Firma/Müşteri Bilgileri Girişi

Kategori İşlemleri

Satış Bilgisi

Satış İşlemleri

Satış Girişi

Listeler

Müşteri Listesi

Ürün Listesi

BİLGİLENDİRME

Kullanıcı Adı : demo

Hoş Geldiniz : Kasiyer Hesap

Yetkiniz : Kasiyer

Görünümü Değiştir

Çık

RecX

ENGİN YENİCE

RecX Market Otomasyonuna Hoş Geldiniz!! Engin Yenice

Tarih : 20 Aralık 2017 Çarşamba
Saat : 07:13:51
Sistem saatiniz gösteriliyor

Ürün Girişi – Ürün Listesi

Perakende Ürün Girişi | RecX Market Otomasyonu

ID	Ürün Adı	Kategori	Adet	Giriş Fiyatı(TL)	Satış Fiyatı(TL)	Barkod Numarası	Kategori ID
170	Test	RecX Kategori	100	11,5	14,5	101	63

Ürün İşlemleri

Ürün ID :

170

Ürün Barkod :

101

Ürün Adı :

Test

Ürün Adedi :

100

Ürün Giriş Fiyatı :

11,5

Ürün Satış Fiyatı :

14,5

Kategori ID :

63

Ürün Kategorisi :

Arama İşlemleri

Ürün Ara :

Kategori Listele :

İşlem Yap

Ekle

Sil

Güncelle

Yenile

Rapor Al

Ürün Listesi – Rapor

Baskı önizleme

Kapat

Sayfa 1

Ürün Listesi

23.12.2017 16:53

ID	Ürün Adı	Kategori	Adet	Giriş Fiyatı(TL)	Satış Fiyatı(TL)	Barkod Numarası	Kategori ID
170	Test	RecX Kategori	100	11,5	14,5	101	63

70

Müşteri Kayıt

Müşteri Kayıt Ekranı

Müşteri Kayıt Ekranı

Firma Kodu :

Müşteri Adı :

Müşteri Unvan :

Telefon 1 :

Cep Telefonu :

E-Posta :

TC Kimlik Numarası :

Doğum Tarihi : 20.12.2017

Adres :

Fotoğraf Seç

Kaydet

Kategori İşlemleri

Kategori İşlemleri | RecX Market Otomasyonu

Kategori Oluştur Kategori Düzenle Kategori Sil

Kategori Düzenle

Düzenlenecek Kategori :

Yeni Kategori Adı :

Değiştir

Duyuru İşlemleri

Duyuru İşlemleri

Duyuru Oluştur

Duyuru Sil

Sil

Duyuru Oluştur

Satış Bilgisi

Satış Bilgisi | RecX Market Otomasyonu

Satış ID	Tarih	Alınan Ücret (TL)	Müşteri Adı
160	23.12.2017 16:...	43,5	Engin
161	23.12.2017 17:...	14,5	Tanımsız Müşteri

Günlük Toplam Ciro : **0,00 TL**

Aylık Toplam Ciro : **58 TL**

Aralık 2017

Pzt	Sal	Çar	Per	Cum	Cmt	Paz
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Bugün: 23.12.2017

Kasa Raporu Al

Kasa Raporu

Baskı önizleme

Kapat

Sayfa 1/5

Kasa Raporu

23.12.2017 17:05

Satış ID	Tarih	Alınan Ücret (TL)	Müşteri Adı
160	23.12.2017 16:58:20	43,5	Engin
161	23.12.2017 17:03:51	14,5	Tanımsız Müşteri

Müşteri Listesi

Müşteri Listesi | RecX Market Otomasyonu

ID	Firma Kodu	Adı	Soyadı / Ünvanı	Telefon	Cep Telefon	Fotoğraf Yolu	Mail Adresi	TC Kimlik No	Doğum Tarihi	Adresi
306		Engin	Yenice	5430000000	5430000000	C:\Program Files ...			23.12.2000 16:37	Eskişehir

Müşteri ID: 306

Müşteri Adı: Engin

Müşteri Ünvanı: Yenice

Müşteri Telefon: 5430000000

Müşteri Cep: 5430000000

Müşteri Adres: Eskişehir

Müşteriyi adı ile bul:

Müşteriyi TC Kimlik numarası ile bul:

Ekle

Güncelle

Yenile

Sil

Rapor Al

Müşteri Listesi – Rapor

Baskı önizleme

Müşteri Listesi

23.12.2017 16:56

ID	Firma Kodu	Adı	Soyadı / Unvanı	Telefon	Cep Telefon	Fotoğraf Yolu	Mail Adresi	TC Kimlik No	Doğum Tarihi	Adresi
306		Engin	Yenice	5430000000	5430000000	C:\Program Files (x86)\...			23.12.2000 16:37:43	Eskişehir

Şirket Bilgileri

Firma Bilgileri | RecX Market Otomasyonu

Şirket Kodu : 26

Şirket Adı : Engin RecX Market Otomasyonu

Telefon : 5436391365

Faks : --

Mail : enginyenice2626@gmail.com

Web Site : --

Adres : Gökmeşdan Mahallesi Fıdangül Sokak No 9 Daire 18

Şehir : Eskişehir

Temizle Kaydet Gözlem Çıkış

Hesap Oluştur

Kullanıcı Hesabı Oluştur

Adınız :

Soyadınız :

Kullanıcı Adı :

Şifreniz :

Yetki Durumu :

Adres :

Telefon No :

Kullanıcı Ekle

Hesap Yetkilendir

Hesap Yetkilendirme | RecX Mar...

Kullanıcı Seçimi :

Kullanıcı Yetki Seçimi :

Yetkilendir

Hesap Şifre Yenileme

Şifre Değişimi

Kullanıcı adını giriniz :

Yeni şifrenizi giriniz :

Şifreyi Güncelle

Hesap Silme

Hesap Silme İşlemi

Hesabı Seçiniz :

Hesabı Kalıcı Olarak Sil

Satış Girişi

Perakende Satış | RecX Market Otomasyonu

Tutar: Sil

0 +

Para Üstü Hesapla

Verilen Para Hesapla

Satış Yap Satış İptal Et

Müşteri Seç

Ürün Bilgileri

Barkod: İsim: Yenile

Cins: Fiyat: Fiş Kes

Ürün Seçimi

Ürün Seçimi | RecX Market Otomasyonu

Cinsler:

- « Tümü »
- Yiyecek
- İçecek
- Temizlik
- Elektronik
- Çiklet
- Çikolatalar
- Kitaplar
- Yağlar

Ürünler:

- Beypiliç Tam Tavuk
- Bulgur
- Ayran
- Bisküvit
- Coca Cola
- Kulaklık
- Sabun
- Bisküvit
- Sütaş Peynir

Barkodu Al

Müşteri Seçimi

Müşteri Seç

58

59

76

78

79

283

55

80

81

82

John

Müşteriyi Seç

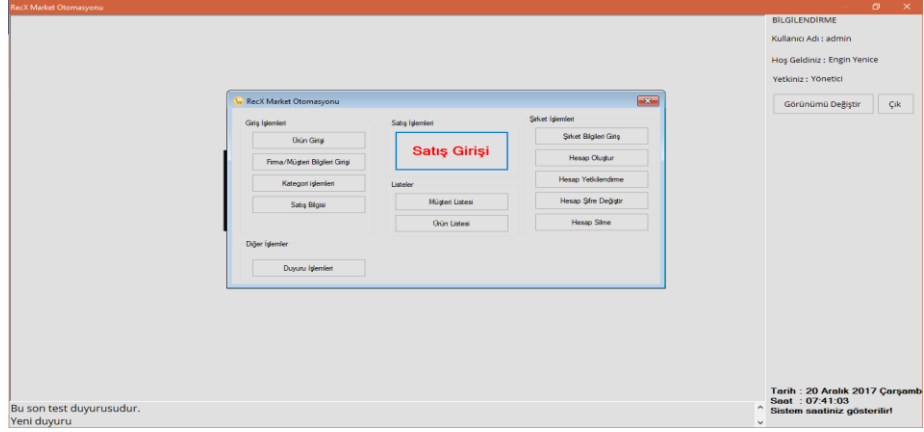
Satış Raporu

Satış Çıktısı

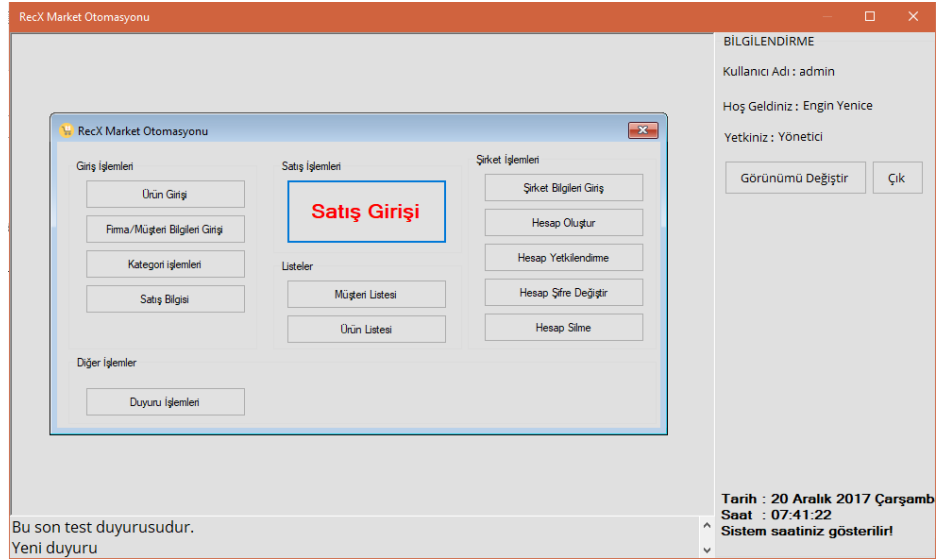
23 Aralık 2017 Cumartesi 16:58

Satış ID	Tarih	Alınan Ücret (TL)	Müşteri Adı
160	23.12.2017 16:58:20	43,5	Engin

Görünüm Değiştir Özelliği



Sabit Küçük Boy Özelliği



ÖZGEÇMİŞ

1998 yılında Eskişehir’de doğdu. İlkokulu ve ortaokulu Eskişehir’in Mahmudiye ilçesinde Necatibey İlköğretim Okulun’da tamamladı. Liseyi Eskişehir’de Toki Şehir İsmail Tetik Anadolu Lisesinde tamamladı. 2016 yılında Kocaeli Üniversitesi Bilgisayar Programcılığı bölümüne başladı ve okumaya devam etmektedir.