

İNTERNET BANKACILIĞI OTOMASYONU

Semih Kavutcu
Kırıkkale Üniversitesi Bilgisayar Mühendisliği

Özet— Teknolojinin hızlı bir biçimde değişmesi ekonomik hayatta karşımıza internet bankacılığı, e- ticaret gibi yeni kavramlar çıkarmıştır. Teknolojinin yarattığı bu yeni kavramlar sayesinde bilginin dolaşım hızı artmış, bilgiye ihtiyaç duyanlar dünyanın çeşitli yerlerindeki bilgiye ulaşarak onları harmanlama ve katma değer içeren bilgileri üretme olanağına kavuşmuştur. Bankaların elektronik ticaret stratejilerinin bir parçası olan internet bankacılığı, bankalar tarafından alternatif bir araç olarak kabul görmektedir. Bankalar şubelerindeki müşteri kuyruklarının ve yoğunluğunun getirdiği olumsuzlukları internet bankacılığı alternatifi ile gidermeye çalışmaktadırlar. Kısa vadede göz önünde bulundurulan amaç, personel ve şube giderlerinin azaltılmasından çok şubelerin normal potansiyelleri kapsamında işlem yapmalarına olanak sağlayan bir ferahlamanın sağlanmasıdır. Yapılan bu çalışmada, bilgiye ulaşmanın kolaylaştığı ticaret ortamında internet bankacılığının rolünü belirleyerek müşteriler ve finans sektörü açısından bu sistemin kolaylığı ve ulaşılabilirliği gösterilmiştir.

1.GİRİŞ (INTRODUCTION)

Bankacılık sektörü, finans sektörü içinde önemli bir yere sahip olmakla birlikte tüm finansal ve reel sektördeki işleyişte para akışının sağlanmasında da önemli rollere sahiptir. Bununla beraber bankacılık sektörü sürekli gelişen küresel ve teknolojik değişimlere uyum sağlamakta, bu değişimleri tüm sistemlerine yansıtmaktadır. Günümüzde internet ve mobil bankacılık uygulamaları ile banka müşterileri şubelere veya ATM'lere gitmeden birçok işlemi rahatlıkla bilgisayarlar ve mobil cihazları üzerinden gerçekleştirebilmektedirler. İnternet bankacılığı, mekan ve zaman kısıtı olmadan herhangi bir akıllı telefon, bilgisayar, elektronik tablet üzerinden bankacılık hizmetlerinin çevrimiçi olarak verilebilmesine imkan tanıyan bir sistem olarak ifade edilmektedir. Bireyler internete erişebildikleri her noktadan bankacılık hizmetlerinden yararlanmaktadır. Bankacılık işlemlerinin neredeyse tamamı internet üzerinden yapılabilmektedir. Fakat bu işlemler içerisinde sadece nakit çekim işleminin internet üzerinden yapılabilmesi mümkün değildir.

2. KULLANIM (USAGE)

2.1. KAYIT / GİRİŞ

Programı ilk başlattığımızda karşımıza kullanıcıyı müşteri değilse; üye/kayıt olma formuna (**Kayıt.cs**), müşteri ise giriş yapma formuna (**Giriş.cs**) yönlendiren **KayıtGiriş.cs** formu çıkıyor. Bu formla kullanıcıyı karşılayıp yapmak istediği işleme göre butonlar yardımıyla diğer formlara yönlendiriyoruz.

```
namespace InternetBanking
{
    public partial class FormKayıtGiriş : Form
    {
        public FormKayıtGiriş()
        {
            InitializeComponent();
        }
    }
}
```

```

    }
    private void uyeolbutonu_Click(object sender, EventArgs e)
    {
        Kayıt kayıt = new Kayıt();          kayıt.Show();
    }
    private void girisyapbutonu_Click(object sender, EventArgs e)
    {
        Giriş giriş = new Giriş();          giriş.Show();
    }
}
}
}

```



Kullanılanlar;

Form, Label, PictureBox, TextBox, Panel, Buton

2.2. KAYIT , Üyelik Sözleşmesi

Form1

ÜYE OLMA EKRANI

İSİM:

SOYİSİM:

TC KİMLİK NO:

TELEFON NO:

ŞİFRE:

☐ Üyelik Sözleşmesini Okudum, Anladım, Kabul Ediyorum.

Üyelik Sözleşmesi

Üye Ol!

listBox1

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

namespace InternetBanking
{
    public partial class Kayıt : Form
    {
        //SQL BAĞLANTISI

        public static SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
A3ICUHU; Initial Catalog = AtmApp; Integrated Security = True");
```

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

private void üyeolbutonu_Click(object sender, EventArgs e)
{
    string isim, soyisim, tc, şifre, telefon;
    isim = üyeisim.Text;
    soyisim = üyesoyisim.Text;
    telefon = üyetelno.Text;
    tc = üyetc.Text;
    şifre = üyeşifre.Text;

    if (üyeisim.Text == "" || üyesoyisim.Text == "" || üyeşifre.Text == "" || üyetelno.Text == "" ||
        üyetc.Text == "")
    {
        MessageBox.Show("Eksik Bilgilerinizi Giriniz!");
    }

    else if (checkBox1.Checked == false)
    {
        MessageBox.Show("Üyelik Sözleşmesini Okuyup Onaylayınız!");
    }

    else if (telefon.Length != 11)
    {
        MessageBox.Show("Telefon Numarası 11 haneli olmalıdır.Yeniden Giriniz!");
    }

    else if (tc.Length != 11)
    {
        MessageBox.Show("TC Kimlik Numarası 11 haneli olmalıdır.Yeniden Giriniz!");
    }
    else
```

```

{

//Kayıt olmak isteyen kullanıcı eğer daha önceden kayıtlıysa SQL'deki müşteriler tablosu yardımıyla
bunu belirleyip kullanıcıya uyarı veriyorum.

SqlCommand komut = new SqlCommand();
komut.Connection = baglanti;
baglanti.Open();
komut.CommandText = "SELECT TC FROM müşteriler WHERE TC=@TC";
komut.Parameters.AddWithValue("@TC", üyetc.Text);
object varMi = komut.ExecuteScalar();
komut.Parameters.Clear();
baglanti.Close();
if (varMi != null)

MessageBox.Show("Bu kullanıcı zaten var.!.");
else
    {

//Yeni kayıt olan kullanıcının SQL'deki müşteriler tablosunda Bakiye kısmı null değeri alıyor.
Burada Bakiye kısmına sıfır değerini atayıp ana menüde yeni kayıt olan kullanıcının para yatırma
işlemi yaparak bakiyesini artırmasını sağlıyorum.

baglanti.Open();
int sayi = 0;
sayi.ToString();

//Burada SQL'i kullanarak kullanıcıdan istediğim değerleri müşteriler tabloma ekliyorum.

SqlCommand command = new SqlCommand("insert into müşteriler (İsim,Soyisim,TC,Telefon,Şifre,Bakiye)
values (@00,@01,@02,@03,@04,@05)", baglanti);
command.Parameters.AddWithValue("@00", üyeisim.Text);
command.Parameters.AddWithValue("@01", üyesoyisim.Text);
command.Parameters.AddWithValue("@02", üyetc.Text);
command.Parameters.AddWithValue("@03", üyetelno.Text);
command.Parameters.AddWithValue("@04", üyeşifre.Text);
command.Parameters.AddWithValue("@05", sayi.ToString());
command.ExecuteNonQuery();
baglanti.Close();

MessageBox.Show("Kaydınız Başarıyla Gerçekleştirilmiştir.Lütfen Önceki Ekrandan Giriş Yapınız.");

//ListBox'ı kullanarak kullanıcıya giriş ekranında kullanacağı kayıt olurken girdiği tc ve
şifresini gösteriyorum.


listBox1.Items.Add("GİRİŞ BİLGİLERİNİZ: TC Kimlik No:" + üyetc.Text + "
Şifre:" + üyeşifre.Text);

}

}

}

```

 Sözleşme

ÜYELİK SÖZLEŞMESİ

Üyelik Sözleşmesi :

Mağazamıza üye olmak için Üye İşlemleri sayfamıza giderek ilgili bölümleri doldurmanız yeterlidir. Üye bilgilerinizi doğru ve eksiksiz doldurmanız iletişim ve ulaşım problemleri olasılığı açısından çok önemlidir. Ürün ve hizmetlerin çabuk ve sıhhatli ulaşması için lütfen dikkatle ve eksiksiz doldurunuz.

Üye olmak oldukça basit ve hızlı bir işlemdir. Üye olmanız hiç bir yükümlülük altına girdiğinizi ifade etmez. Ancak alışveriş yapmadan önce Satış sözleşmesini dikkatle okuyunuz. Dilediğiniz zaman üyelğinizi sonlandırma hakkına sahipsiniz.

Üyelik İptali :

Üye dilediği zaman üyelğini sonlandırma hakkına sahiptir. Üyelğinizi sonlandırdıktan sonra sitemiz ile tüm ilişkiniz kesilmiş olacaktır. Üyelğinizi sonlandırmak için üye girişi yaptıktan sonra iletişim bilgileri sayfamızdan talebinizi iletmeniz.

Aşağıdaki Durumlarda Üyelik İptal Edilemez

- 1 - Üye son verdiği sipariş üzerinden 60 gün geçmemişse.
- 2 - Farklı bir e-posta adresi ile Üyelik İptal talebinde bulunulduysa
- 3 - Üyeye Kayıtlı bilgilerinden ulaşılmıyorsa

Aşağıdaki Durumlarda Üyelik Tarafımızdan İptal Edilir

- 1 - Üye Ahlak dışı, mesajlar gönderiyor ya da yorum yazıyorsa
- 2 - Sahtekarlık girişiminde bulunduysa
- 3 - Sisteme veya "Bubanyo" ismine zarar vermeye yönelik girişimde bulunduysa.

Üye Güvenliği :

Üyenin güvenliği için mağazamızda her türlü önlem alınmıştır. Bu alınan önlemlerin yanında sizlerde üye bilgilerinizin güvenliğinden sorumlusunuz. Mağazamıza giriş için kullandığınız bilgilerinizi hiç kimse ile paylaşmayın, güvenli olduğundan emin olmadığınız bilgisayarlardan sisteme giriş yapmayın.

Farklı Adres :

Her üye için Sipariş Sonlandırırken kendi kayıtlı adresi haricinde farklı adres girebileceği bir adres bölümü bulunmaktadır. Adres bölümünü arkadaşlarınıza hediye gönderirken kullanabileceğiniz gibi, farklı adreslerde olduğunuz dönemlerde kullanabilmemiz için düşünülmüştür.

Örneğin: Çalıştığınız şirketin farklı şubelerinde bulunacağınız dönemlerde veya yazıkta kaldığınız dönemlerde. Yada faturanızın size siparişinizin bir arkadaşınıza ulaşmasını istiyorsanız

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

```
{
```

```
//Üyelik Sözleşmesi Butonu kullanıcıyı üyelik sözleşmesini okumak için Sözleşme.cs formuna gönderiyor.
```

```
Sözleşme sözleşme = new Sözleşme();  
sözleşme.Show();
```

```
}
```

```
}
```

}

Kullanılanlar:

Form, Label, TextBox, Panel, Buton, CheckBox, ListBox, PictureBox

SQL Bağlantısı, Kayıt İşlemi 2.2.1 SQL BAĞLANTISI

SQL' de açtığım AtmApp veritabanında oluşturduğum müşteriler tablosundan hem müşteri olan kullanıcıların bilgilerini programa çekiyorum hem de yeni kayıt olan müşterilerin bilgilerini buraya aktarabiliyorum.

2.3. GİRİŞ

```
using System;  
using System.Collections.Generic;
```

```
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using AtmApp._2AnaMenü;

namespace InternetBanking
{
    public partial class Giriş : Form
    {
        public Giriş()
        {
            InitializeComponent();
        }

        //Giriş ekranında girilen tc yardımıyla diğer formlardan veri çekmek amacıyla tc tanımladım.

        static public string tc;

        //KAPSÜLLEME VE KALITIM İŞLEMİ: Kapsülleme ile private olan tc ve şifre değerlerine get ve set
        özelliklerini kullanarak doğrudan erişimi engelledim. Daha sonra bu değerleri Giriş formuna
        kalıtarak bu formun içinde kullanılmasını sağladım.

        class değerler : Giriş
        {
            private string Tckimlikno;
            private string Şifree;

            public string TCKİMLİKNO
            {
                get { return Tckimlikno; }
                set { Tckimlikno = value; }
            }

            public string ŞİFREE
            {
                get { return Şifree; }
                set { Şifree = value; }
            }
        }
    }
}
```



```

private void girişekranbutonu_Click(object sender, EventArgs e)
{
    değerler DEĞERLER = new değerler();
    string Tckimlikno = DEĞERLER.TCKİMLİKNO;
    string Şifree = DEĞERLER.ŞİFREE;

    Tckimlikno = tctext.Text;

    Şifree = sifretext.Text;

    SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-A3ICUHU; Initial
    Catalog=AtmApp; Integrated Security=TRUE");
    baglanti.Open();

    SqlCommand komut = new SqlCommand("SELECT * FROM müşteriler WHERE TC = '" + tctext.Text + "' and
    şifre= '" + sifretext.Text + "'", baglanti);

    //Yine SQL'i kullanarak aldığım değerleri doğruluğunu sorgulamak amacıyla okutuyorum.

    SqlDataReader oku = komut.ExecuteReader();
    if (oku.Read())
    {
        tc = Convert.ToString(oku["TC"]);

        Anamenü anamenü = new Anamenü();

        anamenü.Show();

    }
    else
    {
        MessageBox.Show("Kullanıcı adı veya şifre hatalı!!! ");
    }
}
}

```

Kullanıcının giriş ekranında girdiği bilgiler SQL bağlantısı yardımıyla kontrol edilir.Eğer başarılı bir şekilde giriş yaparsa AnaMenü'ye yönlendirilecektir.

Kullanılanlar:

Form, Label, TextBox, Panel, Buton,

SQL Bağlantısı, Kullanıcı Girişi, Kapsülleme, Kalıtım

AnaMenü Ekranı İletişim MenüStrip'inin içinde **DiğerŞubelerimiz.cs**, Kampanyalar MenüStrip'inde **Kampanyalar.cs** ve daha sonrasında **ParaÇekme.cs**, **ParaYatırma.cs**, **HavaleEft.cs**, **ŞifreDeğiştirme.cs**, **HesabıSilme.cs** olarak 7 farklı forma açılır. Butonlar ve menüstripler yardımıyla kullanıcı yapmak istediği işleme göre diğer formlara yönlendirilir.

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

```
namespace InternetBanking._2AnaMenü
{
    public partial class Anamenü : Form
    {
        public Anamenü()
        {
            InitializeComponent();
        }
    }
}
```

```
//AnaMenü ekranına SQL'den bakiyeyi çektim. Bu bakiyeyi diğer formlarda kullanabilmek için
kullanıcıbakiye tanımladım.
```

```
static public string kullanıcıbakiye;
```

```
private void AnaMenü_Load(object sender, EventArgs e) {
```

```

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-A3ICUHU; Initial
Catalog = AtmApp; Integrated Security = True");

//AnaMenü ekranına SQL'den Giriş Ekranında tanımladığım tc yardımıyla Kullanıcı İsmi
Çekiyorum.

SqlCommand komut = new SqlCommand("SELECT * FROM müşteriler İsim WHERE TC='" + Giriş.tc + "'",
baglanti);
baglanti.Open();

SqlDataReader dr = komut.ExecuteReader();
if (dr.Read())
{
//Yukarıda tanımladığım kullanıcı bakiyeye kayıtlı bakiyeyi okutuyorum.

kullanıcıbakiye = Convert.ToString(dr["Bakiye"]);
//AnaMenü'de istediğim labela kullanıcı ismini çekiyorum.

kullanıcıadilabel.Text = dr["İsim"].ToString();
//AnaMenü'de istediğim labela kullanıcıya bakiyesini göstermek için bakiyeyi çekiyorum.

ilkbakiye.Text = dr["Bakiye"].ToString();
}
}
private void paraçekmebuton_Click(object sender, EventArgs e)
{
ParaÇekme paraçekme = new ParaÇekme();
paraçekme.Show();
}
private void parayatırmabuton_Click(object sender, EventArgs e)
{
ParaYatırma parayatırma = new ParaYatırma();
parayatırma.Show();
}
private void havaleeftbuton_Click(object sender, EventArgs e)
{
HavaleEft havaleeft = new HavaleEft();
havaleeft.Show();
}
private void şifredeğiştirmebuton_Click(object sender, EventArgs e)
{
ŞifreDeğiştirme şifredeğiştirme = new ŞifreDeğiştirme();
şifredeğiştirme.Show();
}
private void hesabısilbuton_Click(object sender, EventArgs e)
{
HesabıSilme hesapsilme = new HesabıSilme();
hesapsilme.Show();
}
private void çıkışbuton_Click(object sender, EventArgs e)
{

//Kullanıcının yaptığı işleme göre kalan bakiyesini hem SQL'de Bakiye kısmına aktarıp hem de
tekrardan giriş yaptığında AnaMenüdeki ilk bakiyesini önceki işlemde kalan bakiyesiyle
güncelliyorum.

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-A3ICUHU; Initial

```

```

Catalog = AtmApp; Integrated Security = True");
baglanti.Open();

SqlCommand komut1 = new SqlCommand("update müşteriler set Bakiye='" + sonbakiye.Text + "' where
TC='" + Giriş.tc + "'", baglanti); komut1.ExecuteNonQuery();
baglanti.Close();

        Application.Exit();
    }
    private void yardımToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("www.ekofinansbanks.com.tr Sitesine Ulaşıp Yardım Sekmesine
        Gidiniz.");
    }
    private void diğerŞubelerimizToolStripMenuItem_Click(object sender, EventArgs e)
    {
        DiğerŞubelerimiz diğerşubeler = new DiğerŞubelerimiz(); diğerşubeler.Show();
    }

    private void müşteriHizmetleriToolStripMenuItem_Click(object sender, EventArgs e)
    {
        MessageBox.Show("0(312) 410 65 45 Dahili Numaramızdan Müşteri Hizmetlerine
        Ulaşabilirsiniz.");
    }

    private void kampanyalarToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Kampanyalar KAMPANYALAR = new Kampanyalar();
        KAMPANYALAR.Show();
    }
}

```

2.5. DİĞER ŞUBELERİMİZ

```

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

namespace InternetBanking._2AnaMenü
{
    public partial class DiğerŞubelerimiz : Form
    {
        public DiğerŞubelerimiz()
    }
}

```

```

{
    InitializeComponent();
}
private void İletişim_Load(object sender, EventArgs e)
{
    comboBox1.Items.Add("Ankara");
    comboBox1.Items.Add("İzmir");
    comboBox1.Items.Add("İstanbul");
    comboBox1.Items.Add("Trabzon");
    comboBox1.Items.Add("Samsun");
    comboBox1.Items.Add("Ordu");
    comboBox1.Items.Add("Manisa");
    comboBox1.Items.Add("Sakarya");
    comboBox1.Items.Add("Aydın");
    comboBox1.Items.Add("Denizli");
}
private void göster_Click(object sender, EventArgs e)
{
    SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-A3ICUHU; Initial
Catalog = AtmApp; Integrated Security = True");
    baglanti.Open();
    string iller= "SELECT * from iletişim where DiğerŞubeİlleri=@DiğerŞubeİlleri";
    SqlCommand komut = new SqlCommand(iller, baglanti);
    komut.Parameters.AddWithValue("@DiğerŞubeİlleri", comboBox1.Text);
    //iller parametremize combobox'dan girilen değeri SQL DataAdapter ile iletişim tablosundan
    DataGridView'e listelemek için aktarıyoruz.
    SqlDataAdapter da = new SqlDataAdapter(komut);
    DataTable dt = new DataTable();
    da.Fill(dt);
    dataGridView1.DataSource = dt;
    baglanti.Close();
}
}
}

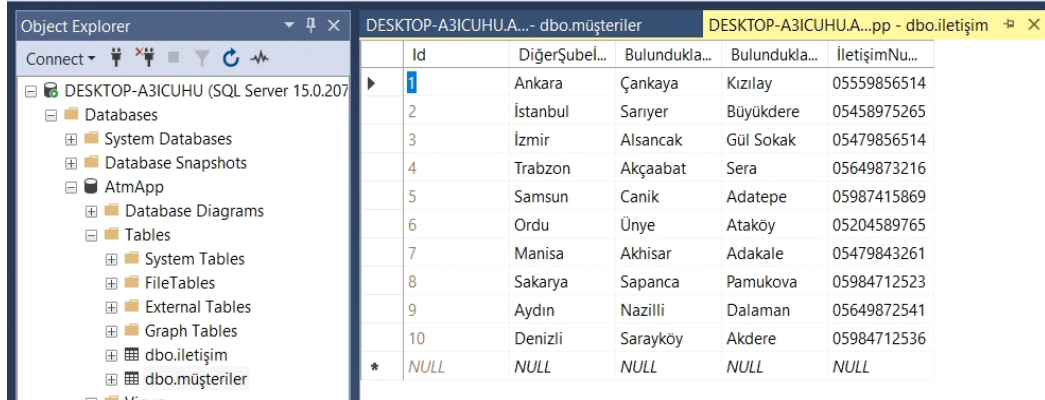
```

2.5.1 SQL BAĞLANTISI

Kullanılanlar:

Form, Label, Buton, Panel, ComboBox,
DataGridView

SQL Bağlantısı , Veri Tabanındaki
Öğeleri Listeleme



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'DESKTOP-A3ICUHU (SQL Server 15.0.207)'. The 'Databases' folder is expanded, showing 'System Databases', 'Database Snapshots', 'AtmApp', 'Database Diagrams', 'Tables', 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', 'dbo.iletisim', and 'dbo.musteriler'. The 'dbo.iletisim' table is selected. On the right, the 'Query Results' pane shows the data for the 'dbo.iletisim' table. The table has 5 columns: 'Id', 'DiğerŞubel...', 'Bulundukla...', 'Bulundukla...', and 'İletişimNu...'. The data is as follows:

Id	DiğerŞubel...	Bulundukla...	Bulundukla...	İletişimNu...
1	Ankara	Çankaya	Kızılay	05559856514
2	İstanbul	Sarıyer	Büyükdere	05458975265
3	İzmir	Alsancak	Gül Sokak	05479856514
4	Trabzon	Akçaabat	Sera	05649873216
5	Samsun	Canik	Adatepe	05987415869
6	Ordu	Ünye	Ataköy	05204589765
7	Manisa	Akhisar	Adakale	05479843261
8	Sakarya	Sapanca	Pamukova	05984712523
9	Aydın	Nazilli	Dalaman	05649872541
10	Denizli	Sarayköy	Akdere	05984712536
*	NULL	NULL	NULL	NULL

2.6. KAMPANYALARIMIZ

```

namespace InternetBanking._2AnaMenü
{
    public partial class Kampanyalar : Form

```

```

{
    public Kampanyalar()
    {
        InitializeComponent();
    }
}
//ABSTRACT KULLANIMI
Benzer özellikler taşıyan sınıflar kullandığım için tekrar tekrar aynı kodları yazmaktansa soyut sınıfları kullanarak kolaylık sağladım.
Kullanıcı kampanyaları göster butonuna basınca sınıflarda tanımladığım metotları kullanarak label'a istediğimi yazacak.
abstract class kredi
{
    public string konutkredisi;          public string araçkredisi;          public
string ihtiyackredisi;

    abstract public void kampanyalar();
}
class kredi1 : kredi
{
    public override void kampanyalar()
    {
        konutkredisi = " 120 Aya Kadar Vade İmkânı İle Hayalinizdeki Evin Sahibi
Olun.";
    }
}
class kredi2 : kredi
{
    public override void kampanyalar()
    {
        araçkredisi = "60 Aya Varan Vade Seçenekleri ve Cazip Faiz Oranlarıyla
Arabanız Kapınızda!";
    }
}
class kredi3 : kredi
{
    public override void kampanyalar()
    {
        ihtiyackredisi = "İhtiyaç Kredisi Kampanyası Kapsamında 36 Aya Varan
Taksitlerle 3 Ay Ertelemeli Kredi Sunuyoruz.";
    }
}

private void button1_Click(object sender, EventArgs e)
{
    kredi1 KREDİ1 = new kredi1();          KREDİ1.kampanyalar();          kredi2
KREDİ2 = new kredi2();          KREDİ2.kampanyalar();          kredi3 KREDİ3 = new kredi3();
    KREDİ3.kampanyalar();
    label1.Text = KREDİ1.konutkredisi;          label2.Text = KREDİ2.arackkredisi;
    label3.Text = KREDİ3.ihtiyackkredisi;
}
}

```

Kullanılanlar:

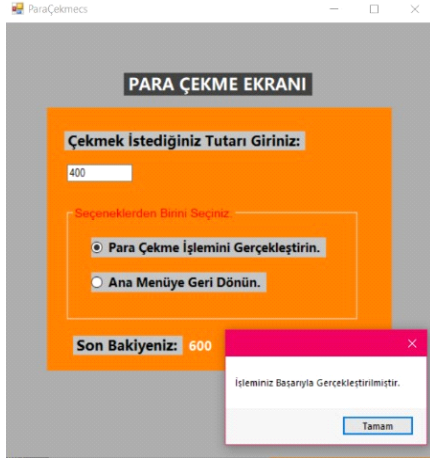
Form, Label, Buton, GroupBox

Abstract

2.7. PARA ÇEKME

Para çekme işleminden önce Ana Menü'ye ilk girdiğimizde;

Para çekme işlemini gerçekleştiriyoruz;



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

namespace AtmApp._2AnaMenü
{
    public partial class ParaÇekme : Form
    {
        public ParaÇekme()
        {
            InitializeComponent();
        }
        private void paraçekmerbutton_CheckedChanged(object sender, EventArgs e)
        {
            if(paraçekmerbutton.Checked == true)
            {
                //Müşteriden çekmek istediği tutarı alıp int veri tipine çevirdim. int
                çekilentutar = 0;
                çekilentutar = Convert.ToInt32(çekilentutartext.Text);
                //Giriş ekranında girilen tc ye göre müşteriye ait olan bakiyeyi çekip int veri tipine
                değiştirdim.
                SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
                A3ICUHU;Initial Catalog = AtmApp;Integrated Security = True"); baglanti.Open();
                string sorgu = "SELECT Bakiye FROM müşteriler WHERE TC='" + Giriş.tc + "'";
                SqlCommand cmd = new SqlCommand(sorgu, baglanti);
                int kullanıcıbakiye = Convert.ToInt32(cmd.ExecuteScalar());
            }
        }
    }
}
```

```

        if(çekilentutar > kullanıciBakiye)
        {
            MessageBox.Show("Bakiyeniz Yetersiz!");
        }
        else
        {
            paraçekmeBakiye.Text= (kullanıciBakiye - çekilentutar).ToString();
            Anamenü anamenüü = new Anamenü();
            //Kalan tutarı ana menüdeki son bakiye labelına aktardım.
            anamenüü.sonbakiye.Text = paraçekmeBakiye.Text;
            MessageBox.Show("İşleminiz Başarıyla Gerçekleştirilmiştir.");
            anamenüü.Show();
        }
    }
}

private void geridönmerbutton_CheckedChanged(object sender, EventArgs e)
{
    Anamenü anamenüü = new Anamenü();
    anamenüü.Show();
}
}
}

```

Kullanılanlar:

Form, Label, Buton, GroupBox, RadioButton, TextBox, Panel

SQL Bağlantısı

2.8. PARA YATIRMA

Para Yatırma İşlemi;

Para Yatırdıktan Sonra;

```

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

namespace InternetBanking._2AnaMenü
{
    public partial class ParaYatırma : Form
    {
        public ParaYatırma()
        {
            InitializeComponent();
        }

        private void parayatırmarbutton_CheckedChanged(object sender, EventArgs e)
        {
            if (parayatırmarbutton.Checked == true)
            {
                //Müşteriden yatırmak istediği tutarı alıp int veri tipine çevirdim.
                int yatırılantutar = 0;
                yatırılantutar = Convert.ToInt32(yatırılantutartext.Text);
                //Giriş ekranında girilen tc ye göre müşteriye ait olan bakiyeyi çekip int veri
                tipine değiştirdim.
                SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-

```



```

A3ICUHU;Initial Catalog = AtmApp;Integrated Security = True");                baglanti.Open();
        string sorgu = "SELECT Bakiye FROM müşteriler WHERE TC='" + Giriş.tc + "'";
SqlCommand cmd = new SqlCommand(sorgu, baglanti);
        int kullanıcbakiye = Convert.ToInt32(cmd.ExecuteScalar());

        label3.Text = (kullanıcbakiye + yatırılantutar).ToString();
Anamenü anamenüü = new Anamenü();                anamenüü.sonbakiye.Text = label3.Text;
        MessageBox.Show("İşleminiz Başarıyla Gerçekleştirilmiştir.");
anamenüü.Show();

        }
        }
        private void geridönmerbutton2_CheckedChanged(object sender, EventArgs e)
        {
            Anamenü anamenüü = new Anamenü();                anamenüü.Show();
        }
    }
}

```

Kullanılanlar:

Form, Label, Buton, GroupBox, RadioButton, TextBox, Panel

SQL Bağlantısı

2.9. HAVALA/EFT İŞLEMLERİ

Havale/EFT İşlemi;

Havaleden Sonra;

```

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

```

```

namespace InternetBanking._2AnaMenü
{
    public partial class HavaleEft : Form
    {
        public HavaleEft()
        {
            InitializeComponent();
        }
        private void havaleeftrbutton_CheckedChanged(object sender, EventArgs e)

```

```

        {
            if (havaleeftrbutton.Checked == true)
            {

//Müşteriden havale etmek istediği tutarı alıp int veri tipine çevirdim.
int havaleeftutarı = 0;

havaleeftutarı = Convert.ToInt32(havaleeftutar.Text);

//Giriş ekranında girilen tc ye göre müşteriye ait olan bakiyeyi çekip int veri tipine değiştirdim.

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-
A3ICUHU;Initial Catalog = AtmApp;Integrated Security = True");                baglanti.Open();

string sorgu = "SELECT Bakiye FROM müşteriler WHERE TC='" + Giriş.tc + "'";
SqlCommand cmd = new SqlCommand(sorgu, baglanti);

int kullanıcıbakiye = Convert.ToInt32(cmd.ExecuteScalar());

if (havaleeftutarı > kullanıcıbakiye)
{
    MessageBox.Show("Bakiyeniz Yetersiz!");
}
else
{

    label3.Text = (kullanıcıbakiye - havaleeftutarı).ToString();                Anamenü anamenüü =
    new Anamenü();
    anamenüü.sonbakiye.Text = label3.Text;

    MessageBox.Show("İşleminiz Başarıyla Gerçekleştirilmiştir.");                anamenüü.Show();

}

}

}

private void geridönmerbutton3_CheckedChanged(object sender, EventArgs e)                {
    Anamenü anamenüü = new Anamenü();                anamenüü.Show();

}

}

}

```

Kullanılanlar:

Form, Label, Buton, GroupBox, RadioButton, TextBox, Panel

SQL Bağlantısı

2.10. ŞİFRE DEĞİŞTİRME

Şifre Değiştirmeden Önce;



Id	İsim	Soyisim	TC	Telefon	Sifre	Bakiye
1	İrem	Çetin	1111111111	05525325901	123İrem	600
4	Berke	Yılmaz	4444444444	05478963231	berke123	8000
6	Okan	Yıldırım	6666666666	05854759869	okan123	10500
7	Yıldız	Sarı	7777777777	05615897441	yıldız123	6000
8	Fatma	Yanar	8888888888	05356987458	fatma123	4000
9	Kemal	Aydoğan	9999999999	05558796323	kemal123	3500
10	Burak	Çoban	12345678910	05056549878	burak123	4000
12	Cansın	Turan	98765432101	05059876541	cansın123	5000
13	Furkan	Şimşek	65432198710	05058579856	furkan123	600
14	Aslı	Sarıca	32165498701	05558962545	aslı123	900
51	Ömer	Oker	98765432170	05059874562	ömer123	10
52	Kaan	Yaman	98765432180	05052586932	kaan123	8274

Şifre Değiştirmeden Sonra;



Id	İsim	Soyisim	TC	Telefon	Sifre	Bakiye
1	İrem	Çetin	1111111111	05525325901	İremİrem	600
4	Berke	Yılmaz	4444444444	05478963231	berke123	9000
6	Okan	Yıldırım	6666666666	05854759869	okan123	5500
7	Yıldız	Sarı	7777777777	05615897441	yıldız123	6000
8	Fatma	Yanar	8888888888	05356987458	fatma123	4000
9	Kemal	Aydoğan	9999999999	05558796323	kemal123	3500
10	Burak	Çoban	12345678910	05056549878	burak123	4000
12	Cansın	Turan	98765432101	05059876541	cansın123	5000
13	Furkan	Simsek	65432198710	05058579856	furkan123	600

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

```
namespace InternetBanking._2AnaMenü
```

```
{
```

```
    public partial class ŞifreDeğiştirme : Form
```

```
    {
```

```
        public ŞifreDeğiştirme()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

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

```
        {
```

```

SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-A3ICUHU; Initial
Catalog = AtmApp; Integrated Security = True");
baglanti.Open();
SqlCommand komut = new SqlCommand("UPDATE müşteriler SET Şifre='" + yenişifrettext.Text
+ "' WHERE TC='" + yenişifretcttext.Text + "'", baglanti);
komut.ExecuteNonQuery();
baglanti.Close();
MessageBox.Show("Şifre Değiştirme İşleminiz Gerçekleşmiştir.");

}
}
}

```

Kullanılanlar:

Form, Label, Buton, Panel

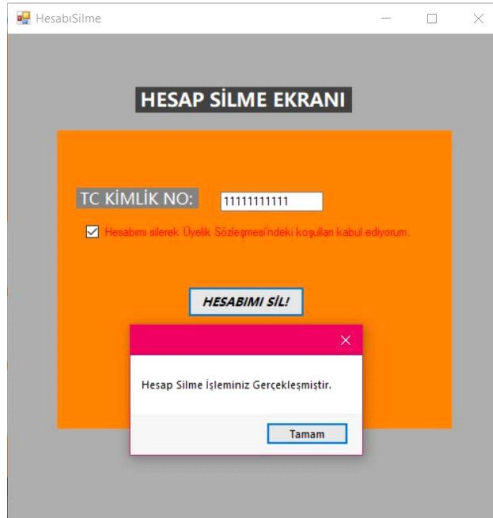
SQL Bağlantısı, Düzeltme İşlemi

2.11. HESABI SİLME

Hesabı Silmeden Önce;

Object Explorer	DESKTOP-A3ICUHU.A... - dbo.müşteriler	DESKTOP-A3ICUHU.A... - dbo.iletişim	SQLQuery2.sql - not co
Connect			
DESKTOP-A3ICUHU (SQL Server 15.0.207)			
Databases			
System Databases			
Database Snapshots			
AtmApp			
Database Diagrams			
Tables			
System Tables			
FileTables			
External Tables			
Graph Tables			
dbo.iletişim			
dbo.müşteriler			
Views			
External Resources			
Synonyms			
Programability			

Hesap Silme İşlemi;



Hesap Silindikten Sonra;

Id	İsim	Soyisim	TC	Telefon	Şifre	Bakiye
4	Berke	Yılmaz	44444444444	05478963231	berke123	9000
6	Okan	Yıldırım	66666666666	05854759869	okan123	5500
7	Yıldız	Sarı	77777777777	05615897441	yıldız123	6000
8	Fatma	Yanar	88888888888	05356987458	fatma123	4000
9	Kemal	Aydoğan	99999999999	05558796323	kemal123	3500
10	Burak	Çoban	12345678910	05056549878	burak123	4000
12	Cansın	Turan	98765432101	05059876541	cansın123	5000
13	Furkan	Şimşek	65432198710	05058579856	furkan123	600
14	Aslı	Sarıca	32165498701	05558962545	aslı123	900
51	Ömer	Oker	98765432170	05059874562	omer123	10
52	Kaan	Yaman	98765432180	05052586932	kaan123	8274
53	Arzu	Öztürk	98765432150	05057418596	arzu123	8299
54	Ömer	Oker	12345678950	05058574152	omer123	33
55	Kutay	Öztürk	33333333333	05059875423	kutay123	8199

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

```
namespace InternetBanking._2AnaMenü
{
    public partial class HesabıSilme : Form
    {
        public HesabıSilme()
        {
            InitializeComponent();
        }
        private void hesabımısılbutonu_Click(object sender, EventArgs e)
        {
            if (checkBox1.Checked == false)
            {
                işlemonaylama ONAY = new işlemonaylama();
                ONAY.onayla();
            }
        }
    }
}
```

```

    }

    else
    {
        SqlConnection baglanti = new SqlConnection("Data Source=DESKTOP-A3ICUHU;
Initial Catalog = AtmApp; Integrated Security = True");
        baglanti.Open();
        SqlCommand komut = new SqlCommand("DELETE from müşteriler WHERE TC='" +
hesapsilmetc.Text + "' ", baglanti);
        komut.ExecuteNonQuery();
        baglanti.Close();
        işlemtamam TAMAM = new işlemtamam();
        TAMAM.onayla();
    }

}

}

public class işlemonaylama
{
    public virtual void onayla()
    {
        MessageBox.Show("Üyelik Sözleşmesini Okuyup Onaylayınız!");
    }
}

public class işlemtamam : işlemonaylama
{
    public override void onayla()
    {
        MessageBox.Show("Hesap Silme İşleminiz Gerçekleşmiştir.");
    }
}
}
}

```

Kullanılanlar:

Form, Label, Buton, TextBox, Panel, CheckBox

SQL Bağlantısı, Silme İşlemi, Polimorfizm

2. SONUÇ

Her sektörde olduğu gibi finans ve bankacılık sektöründe de telekomünikasyon ve BT altyapısının yoğun bir şekilde kullanılmakta olması internetin bu sektörde de kullanımını zorunlu hale getirmektedir. Özellikle gelişen teknoloji ile birlikte artan rekabet şartları bu zorunluluğu tetikleyen bir unsur olarak karşımıza çıkmaktadır. Teknolojik gelişmelere adapte olan bankacılık sektörü web üzerinde internet bankacılığı uygulamalarını müşterileri ile paylaşmaktadırlar.

Ülkemizde internet bankacılığı bankalar tarafından amaç yerine alternatif bir araç olarak kabul görmektedir. Bankalar şubelerindeki müşteri kuyruklarının ve yoğunluğunun getirdiği olumsuzlukları internet bankacılığı alternatifini ile gidermeye çalışmaktadırlar. Bu gereklilikle birlikte, bankalar internet bankacılığı kapasitelerini hızla geliştirmektedirler. Türkiye’de birçok banka internet üzerinden servislerini vermeyi işletme stratejilerinin temel bileşeni olarak benimsemektedirler.

