

WINDOWS PROGRAMLAMA

Kuaför Müşteri Takip Programı

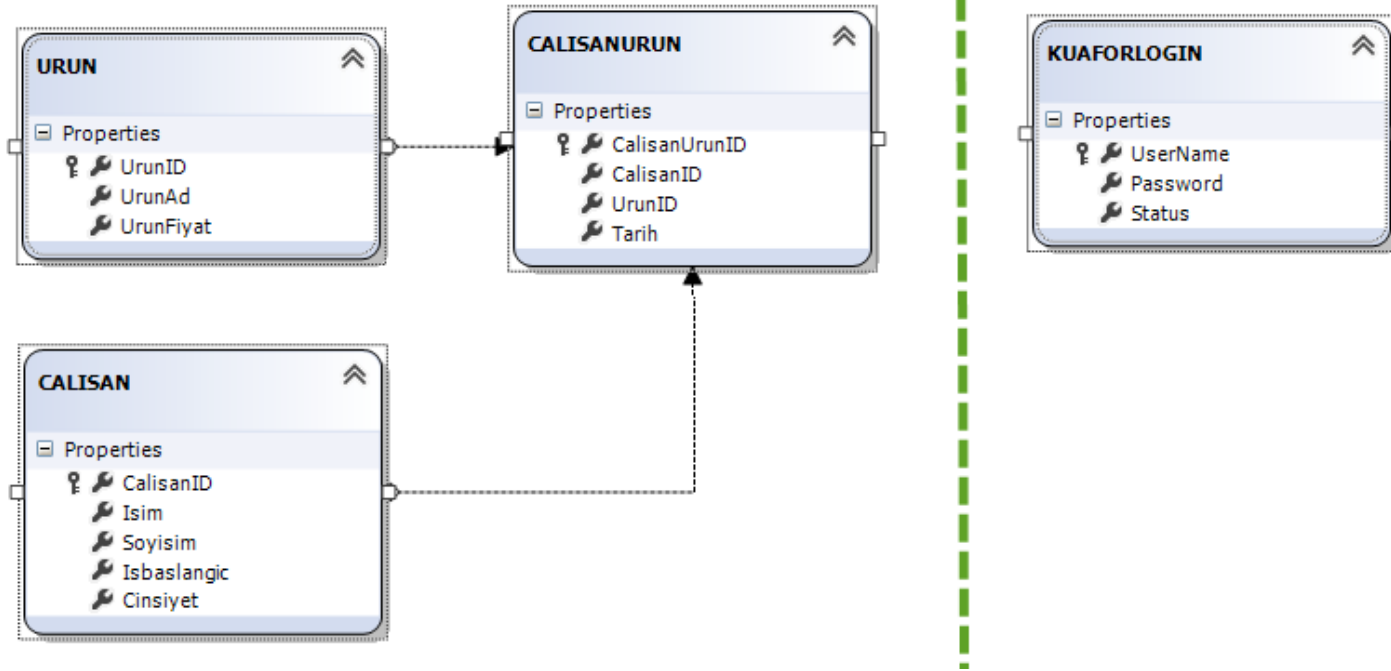
PROJE

Teslim Tarihi : 27.01.2014

Okan KAYA (05-11-94)

Hakan HAKYEMEZ (05-11-84)

PROJEDE KULLANILAN VERİ TABANLARI ve TABLOLARI



URUN: Bu tabloda berberde yapılan işlem çeşitleri(fön-kesim-tıraş) ve bunların fiyatları tutulmaktadır.

CALISANURUN: Bu tabloda hangi çalışanın hangi tip tıraşa ne zaman başladığı tutulmakta.

CALISAN: Berberde çalışan kişileri tutmaktadır. İsim soyisim ise başlangıç ve cinsiyetlerini tutuyor.

KUAFORLOGIN: Form uygulamasına girmek için login ekranında doğru kişinin giriş yapması gerekiyor. Örneğin berberin sahibi bu programa girebilecek.

Program linq to sql bağlantı şekli kullanılarak WFP form nesneleriyle çözülmüştür.

KAYNAK KOD

```
public static class Admin
{
    #region
    public static void AddtabloCalisan(CALISAN tablo)
    {
        using (KuaforDataDataContext conn = new KuaforDataDataContext())
        {
            conn.CALISANs.InsertOnSubmit(tablo);
            conn.SubmitChanges();
        }
    }
    // Calisan ekleme
    public static void UpdatetabloCalisan(CALISAN tablo)
    {
        using (KuaforDataDataContext conn = new KuaforDataDataContext())
        {
            CALISAN tab = (from s in conn.CALISANs
                           where s.CalisanID == tablo.CalisanID
                           select s).FirstOrDefault();
            tab.CalisanID = tablo.CalisanID;
            tab.Isim = tablo.Isim;
            tab.Soyisim = tablo.Soyisim;
            tab.Isbaslangic = tablo.Isbaslangic;
            tab.Cinsiyet = tablo.Cinsiyet;

            conn.SubmitChanges();
        }
    }
    //Calisan guncelleme
    public static void DeleteTabloCalisan(CALISAN tablo)
    {
        using (KuaforDataDataContext conn = new KuaforDataDataContext())
        {
            CALISAN tab = (from s in conn.CALISANs
                           where s.CalisanID == tablo.CalisanID
                           select s).FirstOrDefault();
            conn.CALISANs.DeleteOnSubmit(tab);
            conn.SubmitChanges();
        }
    }
    //Calisan Silme
    public static void AddtabloUrun(URUN tablo)
    {
        using (KuaforDataDataContext conn = new KuaforDataDataContext())
        {
            conn.URUNs.InsertOnSubmit(tablo);
            conn.SubmitChanges();
        }
    }
    //Urun ekleme
    public static void UpdatetabloUrun(URUN tablo)
    {
        using (KuaforDataDataContext conn = new KuaforDataDataContext())
        {
            URUN tab = (from s in conn.URUNs
                        where s.UrunID == tablo.UrunID
                        select s).FirstOrDefault();
            tab.UrunID = tablo.UrunID;
            tab.UrunAd = tablo.UrunAd;
            tab.UrunFiyat = tablo.UrunFiyat;
```

```

        conn.SubmitChanges();
    }
}
//Urun guncelleme
public static void DeleteTabloUrun(URUN tablo)
{
    using (KuaforDataDataContext conn = new KuaforDataDataContext())
    {
        URUN tab = (from s in conn.URUNs
                    where s.UrunID == tablo.UrunID
                    select s).FirstOrDefault();
        conn.URUNs.DeleteOnSubmit(tab);
        conn.SubmitChanges();
    }
}
//Urun guncelleme
public static void AddtabloIs(CALISANURUN tablo)
{
    using (KuaforDataDataContext conn = new KuaforDataDataContext())
    {
        conn.CALISANURUNs.InsertOnSubmit(tablo);
        conn.SubmitChanges();
    }
}
//İş ekleme

public static void DeleteTabloIs(CALISANURUN tablo)
{
    using (KuaforDataDataContext conn = new KuaforDataDataContext())
    {
        CALISANURUN tab = (from s in conn.CALISANURUNs
                            where s.CalisanUrunID == tablo.CalisanUrunID
                            select s).FirstOrDefault();
        conn.CALISANURUNs.DeleteOnSubmit(tab);
        conn.SubmitChanges();
    }
}
//İş silme
public static void DeleteTabloUser(KUAFORLOGIN tablo)
{
    using (UserDataDataContext conn = new UserDataDataContext())
    {
        KUAFORLOGIN tab = (from s in conn.KUAFORLOGINS
                            where s.UserName == tablo.UserName
                            select s).FirstOrDefault();
        conn.KUAFORLOGINS.DeleteOnSubmit(tab);
        conn.SubmitChanges();
    }
}
//Kullanıcı silme
public static void AddtabloUser(KUAFORLOGIN tablo)
{
    using (UserDataDataContext conn = new UserDataDataContext())
    {
        conn.KUAFORLOGINS.InsertOnSubmit(tablo);
        conn.SubmitChanges();
    }
}
//Kullanıcı ekleme
#endregion

```

****Bu kod blogunda projenin tüm tablolarının ekleme silme ve güncelleme işlemleri yapılmaktadır.**

```
try
{
    // Create an ImageBrush
    ImageBrush brush = new ImageBrush();

    // Set image source
    brush.ImageSource = new BitmapImage(new
Uri("C:\\Users\\okan\\documents\\visual studio
2013\\Projects\\KOAFORWPF\\KOAFORWPF\\resim\\" + resim));

    // Set ImageBrush's Stretch property
    brush.Stretch = Stretch.UniformToFill;

    // Set image brush to Chart Background
    Grid.Background = brush;
}
catch (Exception e)
{
}
}
```

****Burada arka plan temasının değişimi yapılmaktadır.**

```
KuaforDataDataContext conn = new KuaforDataDataContext();
//Toplam calisan sayısını hesaplamaktadır

int tabloCalisan = (from s in conn.CALISANs
                    select s
                    ).Count();
//Toplam urun sayısını hesaplamaktadır.
int tabloUrun = (from s in conn.URUNs
                 select s
                 ).Count();
//Toplam Is sayı sını hesaplamaktadır
int tabloIs = (from s in conn.CALISANURUNs
               select s
               ).Count();
//Encok calisan kişilerin ismini sırayla yazdırmaktadır.
var tabloCalisanaa =
    from a in conn.CALISANURUNs
    group a by new { a.CALISAN.CalisanID, a.CALISAN.Isim,
a.CALISAN.Soyisim } into gp
    orderby gp.Count() descending
    select new
    {
        isim= gp.Key.Isim,
        SoyIsim=gp.Key.Soyisim,
        sayi = gp.Count()
    };
//En cok satılan urunlerin listesini sıra ile listelemektedir.
var tabloUrunData =
    from a in conn.CALISANURUNs
    group a by new { a.URUN.UrunID, a.URUN.UrunAd,
a.URUN.UrunFiyat } into gp
    orderby gp.Count() descending
    select new
    {
```

```

            isim = gp.Key.UrunAd,
            UrunFiyat = gp.Key.UrunFiyat,
            sayi = gp.Count()
        };
//Encok para kazandıran ürünlerin listesini ekrana yazdırmaktadır
var tabloUrunFiyat =
    from a in conn.CALISANURUNs
    group a by new { a.URUN.UrunAd, a.URUN.UrunFiyat } into gp
    orderby gp.Sum(x => gp.Key.UrunFiyat) descending
    select new
    {
        isim = gp.Key.UrunAd,
        fiyat=gp.Sum(x =>gp.Key.UrunFiyat)
    };
//Toplam kazancı hesaplamaktadır.
var tabloUrunToplamFiyat =
    from a in conn.CALISANURUNs
    group a by new { a.URUN.UrunFiyat } into gp
    select new
    {
        fiyat = gp.Sum(x => gp.Key.UrunFiyat)
    };

foreach (var item in tabloUrunToplamFiyat)
{
    a+=double.Parse(item.fiyat.ToString());
}

```

//Çalışan kişiyi silme işlemini yapılmaktadır.

```

private void SilButton_Click(object sender, RoutedEventArgs e)
{
    CALISAN selected = TabloGrid.SelectedItem as CALISAN;
    if (selected == null)
    {
        MessageBox.Show("lütfen geçerli alanı seciniz");
    }
    else
    {
        if (MessageBoxResult.Yes == MessageBox.Show("eminmisiniz", "tablo
silme",
        MessageBoxButton.YesNo, MessageBoxImage.Warning))
        {
            Admin.DeleteTabloCalisan(selected);
            Window_Load(null, null);
        }
    }
}

```

//Çalışan ekleme işlemini yapmaktadır.

```

private void ekleButton_Click(object sender, RoutedEventArgs e)
{
    Boolean kontrol = true;
    if (isimBox.Text.Trim() == "")
    {
        kontrol = false;
        MessageBox.Show("Boş İsim Girilemez");
    }
    if (soyisimBox.Text.Trim() == "")
    {

```

```

        kontrol = false;
        MessageBox.Show("Boş Soyİsim Girilemez");
    }
    if (tarihPicker.SelectedDate.Value == null)
    {
        kontrol = false;
        MessageBox.Show("Lütfen Tarih Giriniz");
    }
    CALISAN tablo = new CALISAN();
    if (erkekRadio.IsChecked == true)
    {
        tablo.Cinsiyet = 'E';
    }
    else
        tablo.Cinsiyet = 'K';
    if (kontrol)
    {
        tablo.Isbaslangic = tarihPicker.SelectedDate.Value;
        tablo.Isim = isimBox.Text.Trim();
        tablo.Soyisim = soyisimBox.Text.Trim();
        Admin.AddtabloCalisan(tablo);
        MessageBox.Show("Kişi Eklendi");
        DialogResult = true; //birazdaha bak
    }
}

```

//Calısan listele Formu acıldığında Çalışanları listeler

```

private void Window_Load(object sende, RoutedEventArgs e)
{
    KuaforDataDataContext conn = new KuaforDataDataContext();
    List<CALISAN> tablo = (from s in conn.CALISANs
                          select s).ToList();
    TabloGrid.ItemsSource = tablo;
}

//İş Ekleme işlemini yapar.
private void ekleButton_Click(object sender, RoutedEventArgs e)
{
    CALISAN selectedCalisan = TabloGrid.SelectedItem as CALISAN;
    URUN selectedUrun = TabloGridUrun.SelectedItem as URUN;

    if (selectedCalisan != null)
    {
        if (selectedUrun == null)
        {
            MessageBox.Show("lütfen geçerli Urun alanını seciniz");
        }
        else
        {
            if (tarihPicker.SelectedDate.Value == null)
            {
                MessageBox.Show("Lütfen Tarih Giriniz");
            }
            else
            {
                CALISANURUN calisanurun = new CALISANURUN();
            }
        }
    }
}

```

```

        calisanurun.CalisanID = selectedCalisan.CalisanID;
        calisanurun.UrunID = selectedUrun.UrunID;
        calisanurun.Tarih = tarihPicker.SelectedDate.Value;
        Admin.AddtabloIs(calisanurun);
        MessageBox.Show("İş Eklendi");
        DialogResult = true; //birazdaha bak
    }
}
else
    MessageBox.Show("lütfen geçerli Çalışan alanını seciniz");

}

//Secilen işleri silmesini gerçekleştirir

private void silButton_Click(object sender, RoutedEventArgs e)
{
    CalisanUruns selected = TabloGridUrun.SelectedItem as CalisanUruns;
    CALISANURUN urun = new CALISANURUN();
    urun.CalisanUrunID = selected.CalisanUrunID;
    urun.CalisanID = 1;
    urun.UrunID=1;

    if (selected == null)
    {
        MessageBox.Show("lütfen geçerli alanı seciniz");
    }
    else
    {
        if (MessageBoxResult.Yes == MessageBox.Show("eminmisiniz", "tablo
silme",
        MessageBoxButton.YesNo, MessageBoxImage.Warning))
        {
            Admin.DeleteTabloIs(urun);
            Window_Load(null, null);
        }
    }
}

//İş listeleme formu açıldığında tüm işleri listeler

private void Window_Load(object sende, RoutedEventArgs e)
{
    KuaforDataDataContext conn = new KuaforDataDataContext();

    var tablo3 = (
        from a in conn.URUNS
        join s in conn.CALISANURUNS on a.UrunID equals
        join d in conn.CALISANs on s.CalisanID equals
        select new
        {
            d.Isim,

```



```

        d.Soyisim,
        d.Cinsiyet,
        a.UrunAd,
        a.UrunFiyat,

    }

    );
    TabloGridUrun.ItemsSource = tablo3;

}

//Kullanıcıların şifrelerini ilgili verilerini güncellemesini yapar

private void GirişButton_Click(object sender, RoutedEventArgs e)
{
    KUAFORLOGIN tablo = new KUAFORLOGIN();
    if (KullaniciAdiTextBox.Text.Trim() == "")
    {
        MessageBox.Show("Boş İsim Girilemez");
    }
    else
    {
        if (SifreTextBox.Text.Trim() == "")
        {
            MessageBox.Show("Boş Şifre Girilemez");
        }
        else
        {
            tablo.UserName = KullaniciAdiTextBox.Text.Trim();
            tablo.Password = SifreTextBox.Text.Trim();
            Admin.AddtabloUser(tablo);
            MessageBox.Show("Kullanici Bilgileri Güncellendi");
            DialogResult = true; //birazdaha bak
        }
    }
}

//Ürün ekleme kodu

private void ekleButton_Click(object sender, RoutedEventArgs e)
{
    double number;

    if (isimBox.Text.Trim() == "")
    {
        MessageBox.Show("Boş İsim Girilemez");
    }
    else
    {
        if (!double.TryParse(fiyatBox.Text.Trim(), out number))
        {
            MessageBox.Show("Sayı giriniz ");
        }
        else
        {
            URUN tablo = new URUN();
            tablo.UrunAd = isimBox.Text.Trim();
            tablo.UrunFiyat = number;
            Admin.AddtabloUrun(tablo);
            MessageBox.Show("Urun Eklendi");
            DialogResult = true; //birazdaha bak
        }
    }
}

```

//Secilen ürünü siler

```
private void SilButton_Click(object sender, RoutedEventArgs e)
{
    URUN selected = TabloGrid.SelectedItem as URUN;
    if (selected == null)
    {
        MessageBox.Show("lütfen geçerli alanı seciniz");
    }
    else
    {
        if (MessageBoxResult.Yes == MessageBox.Show("eminmisiniz", "tablo
silme",
        MessageBoxButton.YesNo, MessageBoxImage.Warning))
        {
            Admin.DeleteTabloUrun(selected);
            Window_Load(null, null);
        }
    }
}
```

//Tüm ürünleri form sayfası açıldığında listeleme yapar

```
private void Window_Load(object sende, RoutedEventArgs e)
{
    KuaforDataDataContext conn = new KuaforDataDataContext();
    List<URUN> tablo = (from s in conn.URUNS
                        select s).ToList();
    TabloGrid.ItemsSource = tablo;
}
```

//Anasayfadaki animasyonun XML kodu

```
<DockPanel Margin="0,177,0,0">
    <Viewbox Margin="0,10,0,0">
        <Canvas Width="321" Height="212">

            <!-- The Viewport3D provides a rendering surface for 3-D visual
content. -->
            <Viewport3D Name="MyAnimatedObject"
ClipToBounds="True" Width="321" Height="191" Canvas.Top="10">

                <!-- Defines the camera used to view the 3D object. -->
                <Viewport3D.Camera>
                    <PerspectiveCamera x:Name="myPerspectiveCamera"
Position="0,0,2" LookDirection="0,0,-1"
FieldOfView="60" />
                </Viewport3D.Camera>

                <!-- The ModelVisual3D children contain the 3D models -->
                <Viewport3D.Children>

                    <!-- Two ModelVisual3D define the lights cast in the
scene. Without light, the
                    3D object cannot be seen. Also, the direction of the lights affect
shadowing. -->
                    <ModelVisual3D>
                        <ModelVisual3D.Content>
```

```

0.612372,-0.5,-0.612372" />
        </ModelVisual3D.Content>
    </ModelVisual3D>
    <ModelVisual3D>
        <ModelVisual3D.Content>
            <DirectionalLight Color="#FFFFFF" Direction="-
Direction="0.612372,-0.5,-0.612372" />
            </ModelVisual3D.Content>
        </ModelVisual3D>
    </ModelVisual3D>
    <ModelVisual3D.Content>
        <GeometryModel3D>

            <!-- The geometry specifies the shape of the 3D
plane. In this case, a flat sheet is created. -->
            <GeometryModel3D.Geometry>
                <MeshGeometry3D
TriangleIndices="0,1,2 3,4,5 "
Normals="0,0,1 0,0,1 0,0,1 0,0,1 0,0,1 0,0,1 "
TextureCoordinates="0,0 1,0 1,1 1,1 0,1 0,0 "
Positions="-0.5,-0.5,0.5 0.5,-0.5,0.5 0.5,0.5,0.5 0.5,0.5,0.5 -
0.5,0.5,0.5 -0.5,-0.5,0.5 " />
            </GeometryModel3D.Geometry>

            <!-- The material specifies the material
applied to the plane. In this case it is a linear gradient.-->
            <GeometryModel3D.Material>
                <MaterialGroup>
                    <DiffuseMaterial>
                        <DiffuseMaterial.Brush>
                            <LinearGradientBrush
StartPoint="0,0.5" EndPoint="1,0.5">
<LinearGradientBrush.GradientStops>
                                                                    <GradientStop
Color="Yellow" Offset="0" />
                                                                    <GradientStop
Color="Red" Offset="0.25" />
                                                                    <GradientStop
Color="Blue" Offset="0.75" />
                                                                    <GradientStop
Color="LimeGreen" Offset="1" />
</LinearGradientBrush.GradientStops>
                                                                    </LinearGradientBrush>
                                                                </DiffuseMaterial.Brush>
                    </DiffuseMaterial>
                </MaterialGroup>
            </GeometryModel3D.Material>

            <!-- The Transform specifies how to transform
the 3D object. The properties of the
Rotation object are animated causing the 3D object to rotate
and "wobble" (see Storyboard below).-->
            <GeometryModel3D.Transform>
                <RotateTransform3D>
                    <RotateTransform3D.Rotation>
                        <AxisAngleRotation3D
x:Name="myAngleRotation" Axis="0,3,0" Angle="40" />
                        </RotateTransform3D.Rotation>
                    </RotateTransform3D>
                </GeometryModel3D.Transform>
            </GeometryModel3D>
        </Content>
    </Page>
</UserControl>

```

```

        </GeometryModel3D.Transform>
    </GeometryModel3D>
    </ModelVisual3D.Content>
</ModelVisual3D>
</Viewport3D.Children>

<!-- Trigger the rotation animation when the 3D object loads.
-->

<Viewport3D.Triggers>
    <EventTrigger RoutedEvent="Viewport3D.Loaded">
        <BeginStoryboard>
            <Storyboard>

                <!-- This animation animates the Angle
property of the AxisAngleRotation3D
making the 3D object rotate from -60 degrees to 60 degrees. -->
                <DoubleAnimation
Storyboard.TargetName="myAngleRotation"
Storyboard.TargetProperty="Angle"
From="-60" To="60" Duration="0:0:4" AutoReverse="True"
RepeatBehavior="Forever"/>

                <!-- This animation animates the Axis property
of the AxisAngleRotation3D
making the 3D wobble as it rotates. -->
                <Vector3DAnimation
Storyboard.TargetName="myAngleRotation"
Storyboard.TargetProperty="Axis"
From="0,3,0" To="1,0,1" Duration="0:0:4" AutoReverse="True"
RepeatBehavior="Forever"/>

            </Storyboard>
        </BeginStoryboard>
    </EventTrigger>
</Viewport3D.Triggers>
</Viewport3D>
</Canvas>

</Viewbox>
</DockPanel>

```

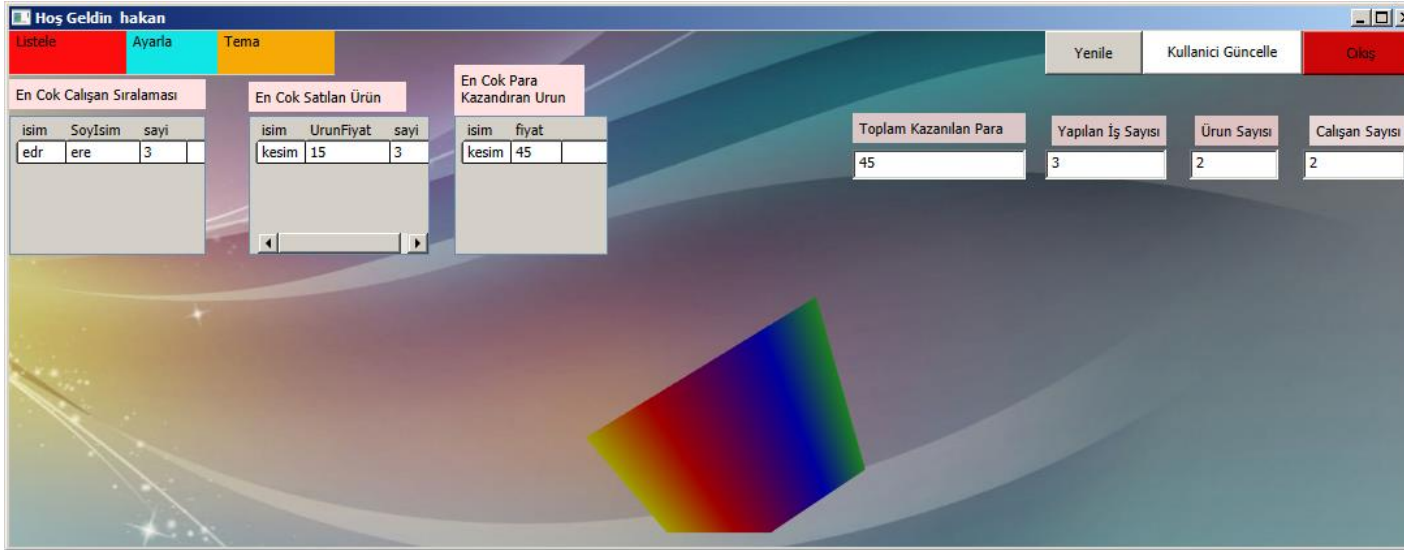
KULLANICI KATALOGU:



1

Databasede yetkisi girilen kişiler, Kullanıcı adı ve şifre girerek programa giriş yapabilmektedir.

Programın Açılış Ekranı:



isim	Soyİsim	sayı
edr	ere	3

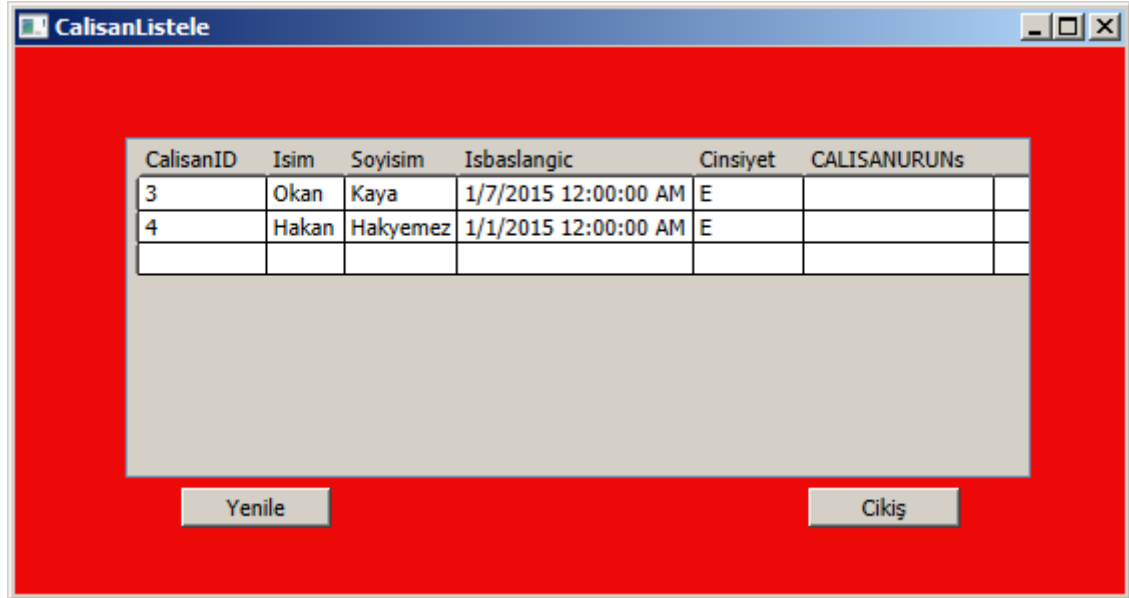
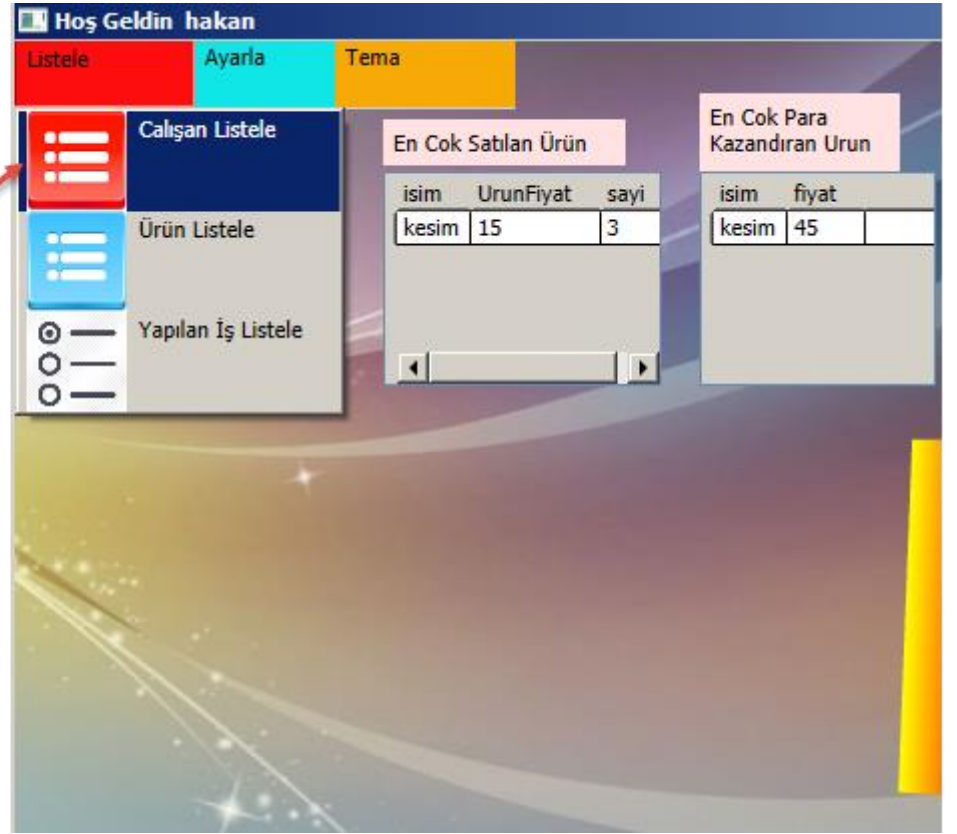
isim	UrunFiyat	sayı
kesim	15	3

isim	fiyat
kesim	45

Toplam Kazanılan Para	Yapılan İş Sayısı	Ürün Sayısı	Çalışan Sayısı
45	3	2	2

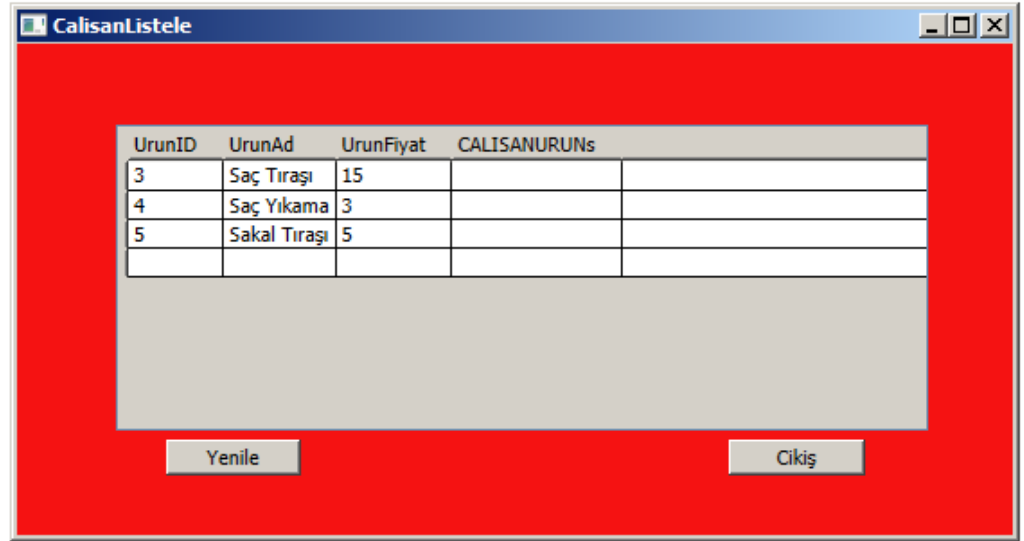
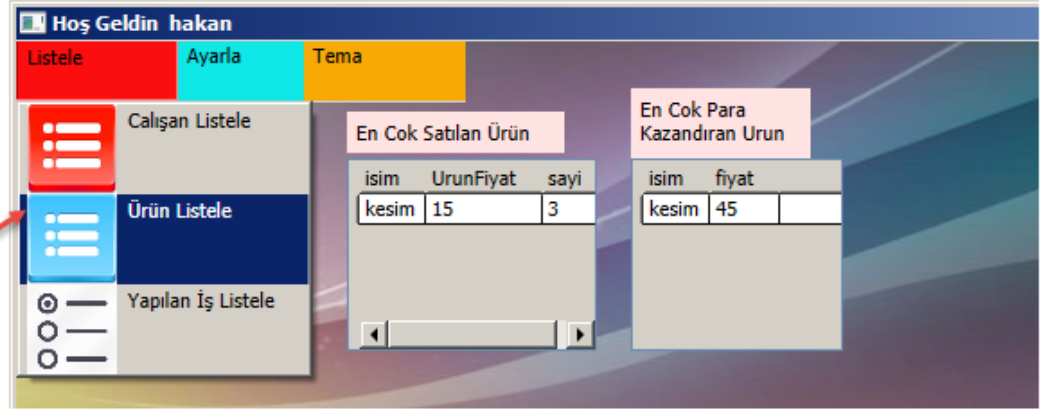
Bu ekranda En çok çalışan kişiler, en çok yapılan tıraş çeşiti, en çok hangi işlemden para kazanıldığı sağ tarafta ise Berber Sahibinin toplam kazandığı para yapılan iş sayısı yapılan ürün çeşiti sayısı ve berberinde çalışan kişi sayısı listelenmektedir.

Bu kısma
tıklandığında



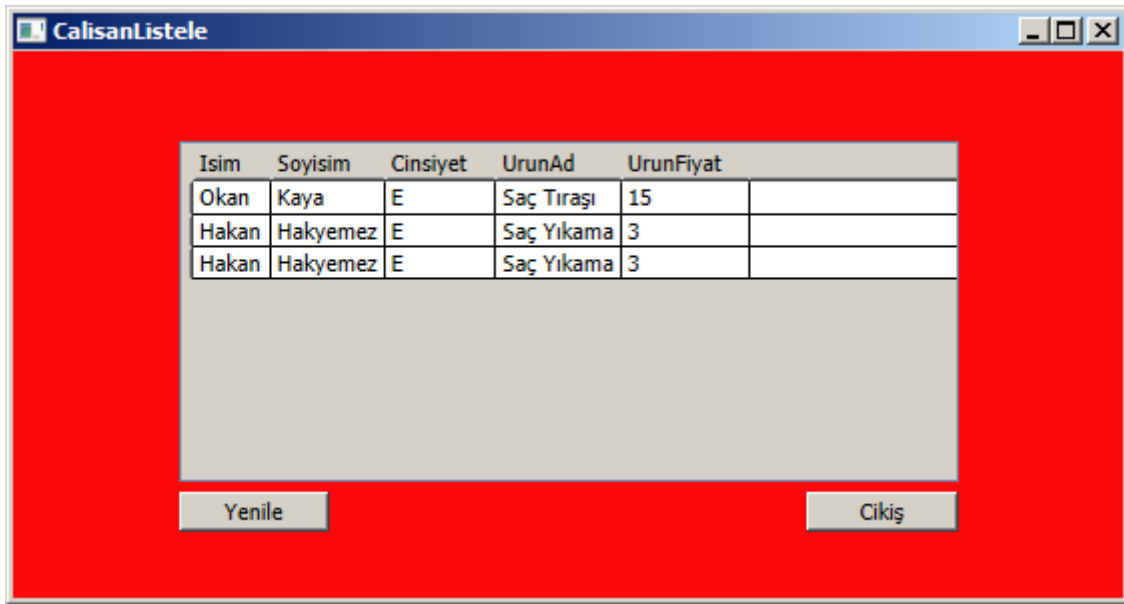
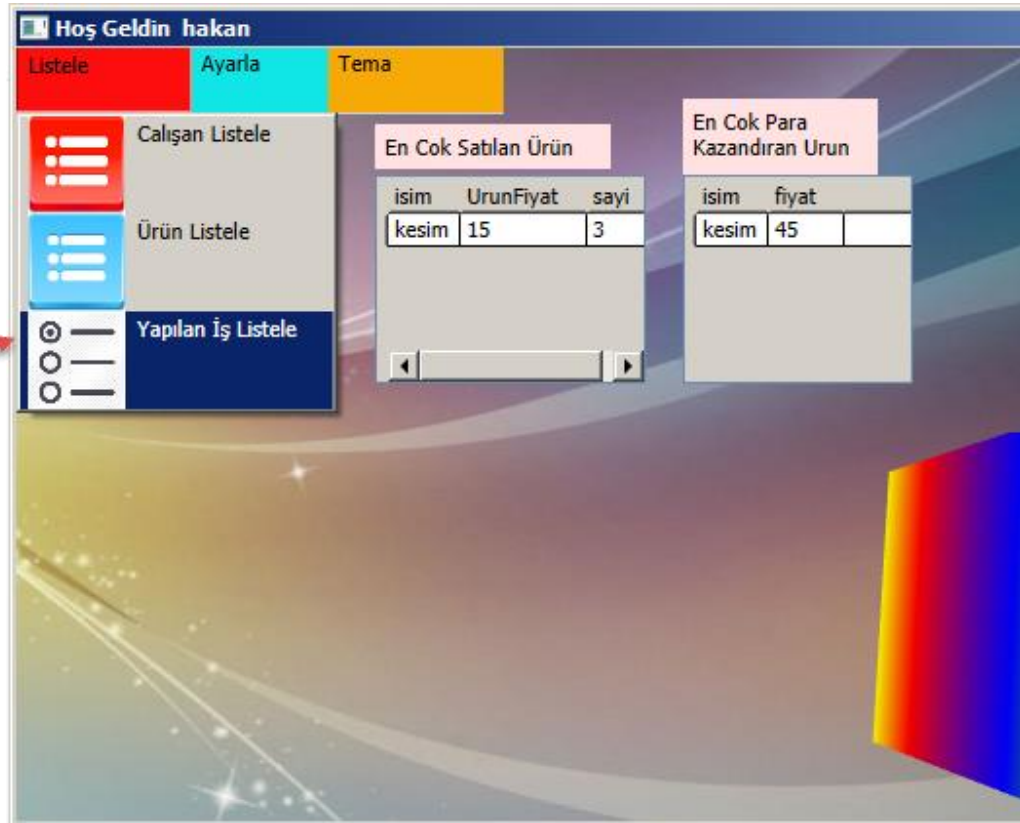
Ekrana berberde çalışan kişiler listelenmektedir. Db den çekilen veriler
GrdiView içinde listelenmektedir.

Ürün Listele
Kısına
Tıkladığımızda



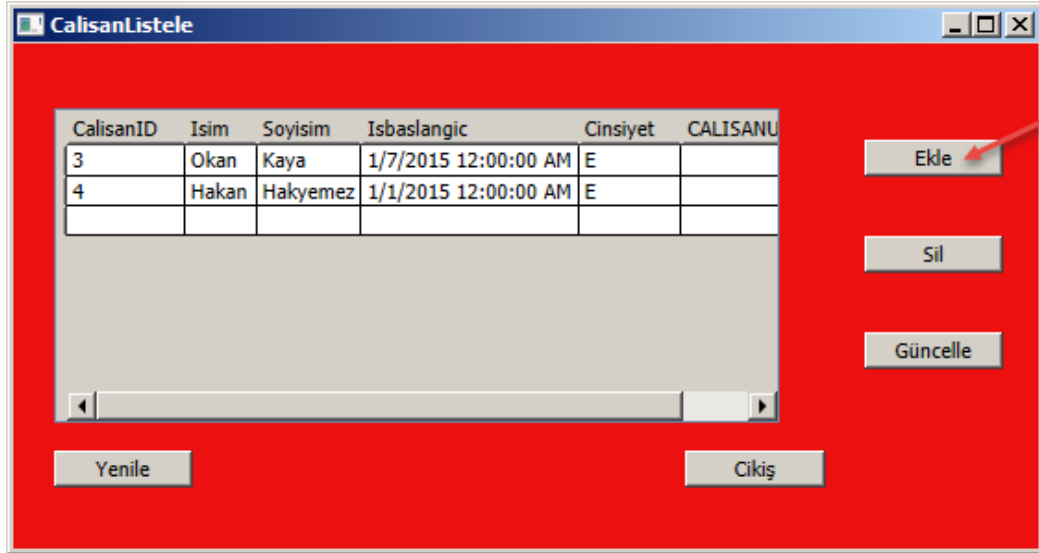
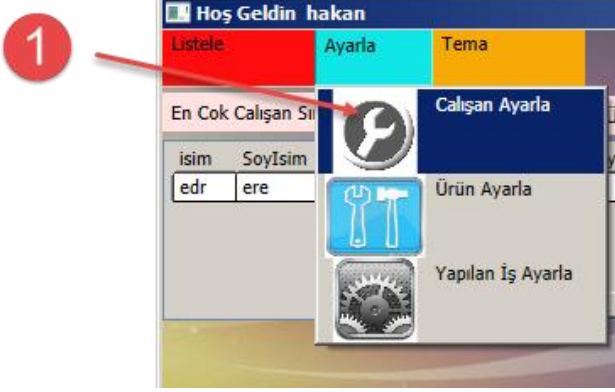
Ekrana berberde yapılan işlemler adları ve fiyatları listelenmektedir.

Yapılan İşleri
Listele
Dediğimizde
ise,



Hangi işçinin hangi tıraş işlemini yaptığı listelenmektedir.

Yeni Bir Çalışan Eklemek İçin,



CalisanEkle

İsim

Soyİsim

Select a date 15

☐ Erkek ☐ Kız

Ekle Cıkış

Kısımları sırayla doldurulmalı ve Ekle butonuna basılmalıdır.

Yeni Bir Ürün Ayarlamak İçin

Hoş Geldin hakan

Listele Ayarla Tema

En Çok Çalışan Sı

isim Soyİsim

edr ere

Calışan Ayarla

Ürün Ayarla

Yapılan İş Ayarla

UrunAyarla

UrunID	UrunAd	UrunFiyat	CALISANURUNs
3	Saç Tıraşı	15	
4	Saç Yıkama	3	
5	Sakal Tıraşı	5	

Ekle

Sil

Güncelle

Yenile Cıkış

UrunEkle

Ürün İsmi

Ürün Fiyatı

Ekle Çıkış

Kısımları sırasıyla ilerlemeli ve doldurulmalıdır.

İşçilerden Herhangi Birinin Yaptığı İş Ayarlayabilmek İçin,

Hoş Geldin hakan

Listele Ayarla Tema

En Çok Çalışan Sı

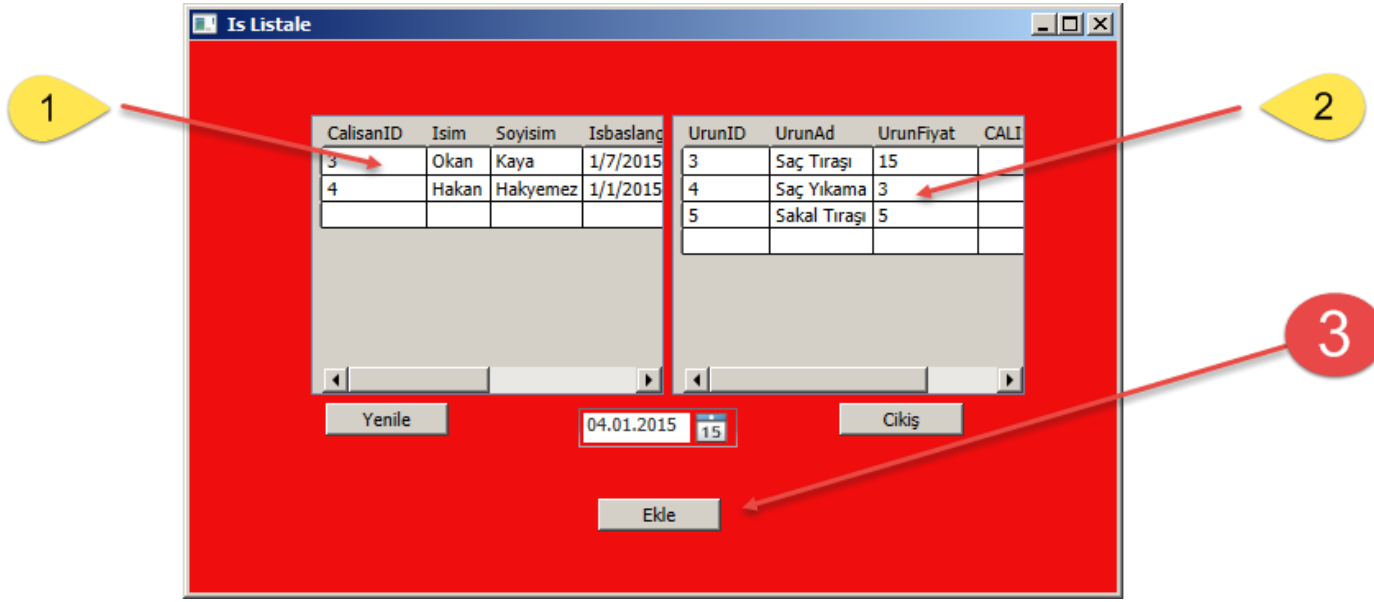
Çalışan Ayarla

Yapılan İş Ayarla

İş Ayarla

CalisanUrunID	Isim	SoyIsim	Cinsiyet	UrunAd	UrunFiyat
4	Okan	Kaya	E	Saç Tıraşı	15
5	Hakan	Hakyemez	E	Saç Yıkama	3
6	Hakan	Hakyemez	E	Saç Yıkama	3

Ekle Sil Yenile Çıkış



Önce çalışan daha sonra yaptığı iş ve tarih seçilip Ekle butonuna basılır.

Ana Ekranda Listelenen Unsurlar

En Çok Çalışan Sıralaması

isim	SoyIsim	sayı
Hakan	Hakyemez	2
Okan	Kaya	1

En Çok Satılan Ürün

isim	UrunFiyat
Saç Yıkama	3
Saç Tıraşı	15

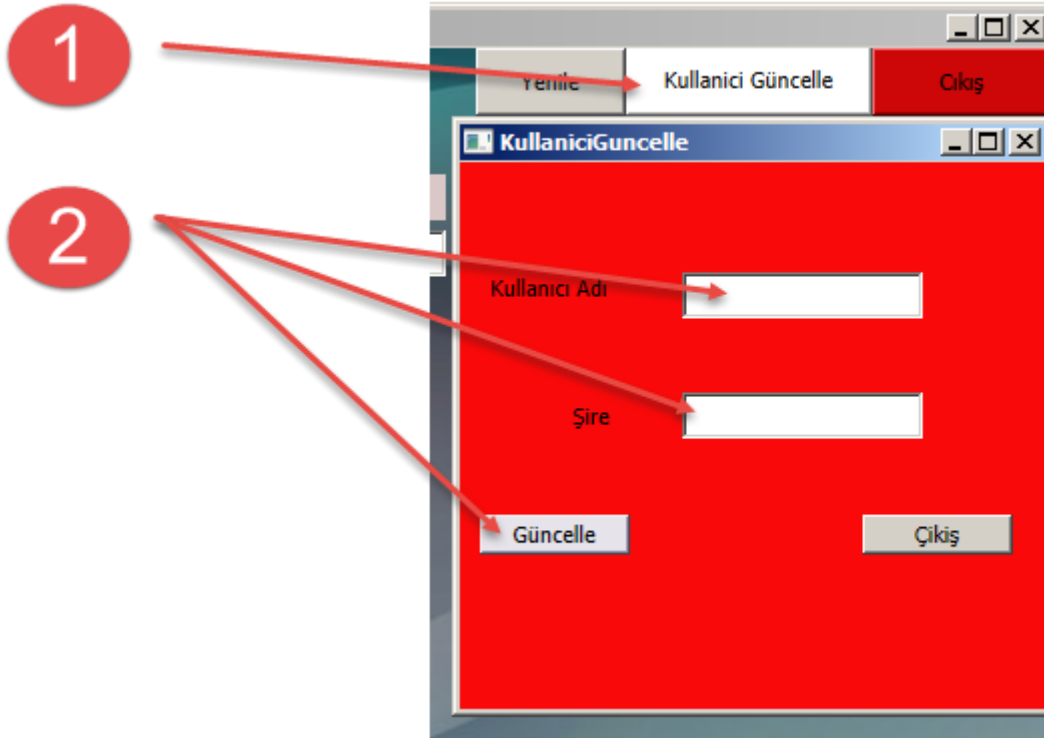
En Çok Para Kazandıran Ürün

isim	fiyat
Saç Tıraşı	15
Saç Yıkama	6

Sol Tarafta,

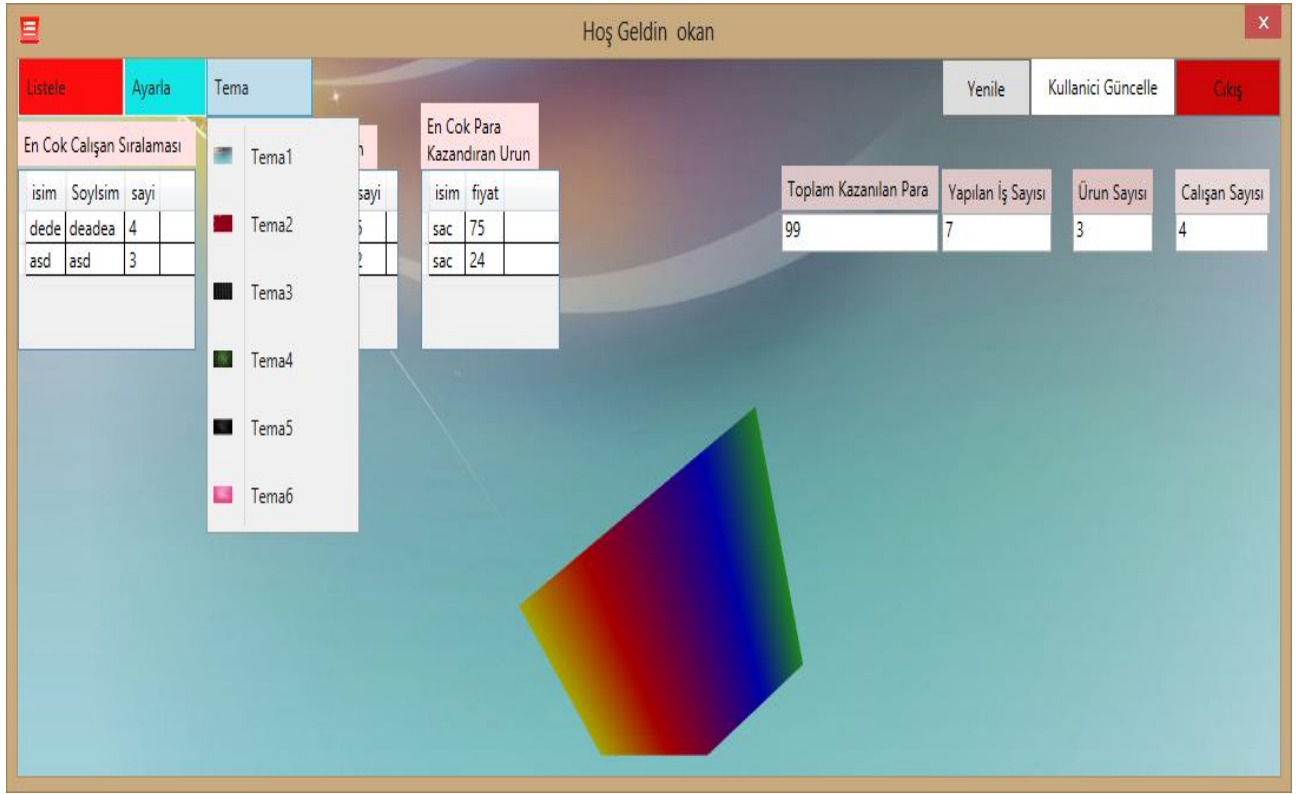
Toplam Kazanılan Para	Yapılan İş Sayısı	Ürün Sayısı	Çalışan Sayısı
21	3	3	2

Yeni Bir Kullanıcı Ekleyebilmek İin,



Yenile butonu, yeni veri girildiğinde istatistiksel verilerin güncellenmesini sağlıyor.

Programı tamamen sonlandırmak için Çıkış butonunu kullanmalısınız.



Kullanıcı Tema secenegine tıkladığında var olan temalar karşısına çıkmaktadır ve kullanıcı ayarlamak istediği arka plana tıkladığında arka plan resmi değişmektedir.