

PROJE RAPORU

ETKİNLİK-BİLET SİSTEMİ

Biletix, DasDas gibi etkinlikler için bilet işlemlerinin yapılabileceği bir sistem.

İçindekiler

1. Ana Ekran

1.1. Personel Ekran

1.1.1. Personel İşlemler

1.1.1.1. Personel İşlemleri

1.1.1.2. Üye İşlemleri

1.1.1.3. Mekan İşlemleri

1.1.1.4. Etkinlik İşlemleri

1.1.1.5. Muhasebe İşlemleri

1.2. Bilet İşlemleri

1.2.1. Bilet

1. Giriş

Çalışma bakımından güncel olarak kullanılan, kullanıcılar tarafından tercih edilen siteler örnek alındı. Böylelikle arka planda güncel yapılara bağlı olarak kod geliştirildi.

2. Yöntem

Kod kısmı C# dili ile MS Visual Studio kullanılarak, tasarım MS Visual Studio- Windows Form yapısında ve veri tabanı ise MS SQL’de geliştirildi.

Farklı yazım çeşitleri ve algoritmaların denenmesini sağlayan çok yönlü bir proje olduğundan proje içerisinde farklılıklar gözlemlenmektedir. Detaylarına ‘Bulgular’ bölümünden ulaşabilirsiniz.

3. Bulgular

```

public void GetDGV(string whereValue = null, DataGridView DGV = null, Fill showDGV = Fill.Fill, params string
[] Tur)
{
    for (int i = 0; i < dgv.Length; i++)
    {
        dgv[i].Visible = false;

        if (showDGV == Fill.Fill)
        {
            dgv[i].Dock = DockStyle.Fill;
        }
    }

    if (showDGV == Fill.Fill)
    {
        DGV.Visible = true;

        con.Open();

        for (int i = 0; i < Tur.Length; i++)
        {
            if (i == 0)
            {
                whereValue += " EtkinlikTuru.TurAdi= '" + Tur[i] + "' ";
            }
            else
            {
                whereValue += " or EtkinlikTuru.TurAdi= '" + Tur[i] + "' ";
            }
        }

        string sorgu = $"Select Etkinlikler.Id,Ad,Mekan,Salon,Tarih,Saat from Etkinlikler inner join
            EtkinlikTuru on (EtkinlikTuru.Id=Etkinlikler.EtkinlikTurId) where {whereValue} ";

        SqlDataAdapter verigetir = new SqlDataAdapter(sorgu, con);

        DataTable dt = new DataTable();

        verigetir.Fill(dt);

        DGV.DataSource = dt;

        con.Close();
    }
}
21 references
public enum Fill { Fill }

```

Resim1: DataGridView(DGV)lerin İçinin Otomatik Bir Sorgu İle Doldurulması

Resimde görüldüğü üzere sorguda sistemi aynı olup sınırlandırmaları değişen çokça DGV için geliştirilmiş bir yöntem bulunmaktadır. Burada eğer bir sınırlandırıcı eklenmemişse ilk olarak “or” kullanmadan daha sonra eklenenler ise başında “or” kullanılarak eklenmektedir. Her DGV için atanan sınırlandırıcılar params dizisinde tutulur ve oradan çekilerek for döngüsünde döner. Böylece sınırlandırıcıların sırası ve sayısı belli olmaktadır.

Enum Fill metodu ise GroupBox(GB) ve DGV’leri ‘Dock.Fill’ yani sınırları olan ekrana sığdıran bir yöntemdir.

```

1 reference
private void AramaTB_TextChanged(object sender, EventArgs e)
{
    ae.con.Open();
    SqlDataAdapter adapt = new SqlDataAdapter("select * from Etkinlikler where Ad like '" + AramaTB.Text +
        "%'", ae.con);
    DataTable dt = new DataTable();
    adapt.Fill(dt);
    EtkinliklerDataGV.DataSource = dt;
    ae.con.Close();
}

```

Resim2: DGV İçinde Like Arama Yapmak

DGV İçinde veri aramanın birçok yolu vardır. Bunlardan biri Resim2’de görülen yöntemdir. Yöntemin esası aranacak veri ya da verileri TextBox(TB)’a yazarak o değeri SQL sorgusu içinde ‘like’ diyerek sorgulamaktır. Ancak dikkat edilmesi gereken

like DGV'in bütün kolonlarında değil, seçili kolonunda iş görmektedir. Burada 'Ad' isimli kolonda arama yapılmaktadır.

```
private void MIlceCB_SelectedIndexChanged(object sender, EventArgs e)
{
    if (MIlceCB.SelectedIndex != 0 && MIlceCB.SelectedIndex != -1)
    {
        MekanAdGetir();
    }
}

1 reference
private void MAdCB_SelectedIndexChanged(object sender, EventArgs e)
{
    if (MAdCB.SelectedIndex != 0 && MAdCB.SelectedIndex != -1)
    {
        SalonCBGetir();
        MekanGetir();
    }
}

1 reference
private void MSalonCB_SelectedIndexChanged(object sender, EventArgs e)
{
    if (MIlCB.SelectedIndex != -1 && MIlceCB.SelectedIndex != -1 && MAdCB.SelectedIndex != -1 &&
        MSalonCB.SelectedIndex != -1)
    {
        SalonBilgiListGetir();
        SalonGetir();
    }
}
```

Resim3: ComboBox(CB) Etkileşimleri

Unutulmamalıdır ki bu tip sistemlerde çokça il, ilçe, mekan, salon bulunmaktadır. Hepsini CB'lara çağırabilirsiniz ancak sınırlandırmazsanız hangi ilçe hangi ilin ya da hangi mekan hangi ilçeye bağlı sorularıyla karşılaşacaksınız. Bu durumun çözümü için Resim3'teki yöntem uygulanmıştır. İlçe CB'nin index değeri değişince o index değerine bağlı Mekanlar, MekanCB'ye atanacaktır. Sistemde yapılan zincir aşağıdaki gibidir.

İl>İlçe>Mekan>Salon

```
1 reference
private void TarihDTP_ValueChanged(object sender, EventArgs e)
{
    tarih = TarihDTP.Value.Date.ToString("d.M.yyyy");
}
```

Resim4: DateTimePicker(DTP) Kullanımı

Etkinlik tarihi, bilet satış tarihleri vb. tarihlerin verilmesi için araç olarak DTP kullanılmıştır. Tarihlerin birçok yazımı vardır. Burada tercih edilen gün ve ay 1-2 digit olan yapılardır.

Ayrıca tarihler için SQL tarafında anında yapılan için tarihi alınsın diye `values(GETDATE())` yapısı kullanılarak o anın tarihi ve saati çekilmektedir. Sadece tarih istenildiği için GETDATE yapısının atandığı değer 'datetime' olarak değil 'date' olarak tanımlanmıştır.

```

private void PAraTxt_TextChanged(object sender, EventArgs e)
{
    List<string> list = new List<string>();

    foreach (string item in PersonelList.Items)
    {
        list.Add(item);
    }

    PersonelList.Items.Clear();

    foreach (string kisi in list)
    {
        if (kisi.Contains(PAraTxt.Text))
        {
            PersonelList.Items.Add(kisi);
        }
    }
}

```

Resim5: ListBox(LB) İçinde Arama

Like aramanın bir başka yöntemi de 'Ekleme'dir. Boş bir LB içine eldeki verileri foreach ile döndürerek TB'a yazılan veri ya da verileri bulunduranları eklemektir.

```

declare @ToplamBakiye int
select top 1 @ToplamBakiye=ToplamBakiye FROM Muhasebe order by Id desc

```

Resim6: SQL Son Veri Değerini Almak

'order by Id desc' yapısıyla Id değerine göre tabloyu tersten sıralıyor ve 'top 1' değeriyle birlikte en sondaki değer ilk değer olduğu için son değer olarak çekilmiş olur.

```

private Button AddNewButton(string ButtonName)
{
    go:
    if (j <= sirakoltuksayisi * k)
    {
        Buton = new()
        {
            Location = new Point(10 + x, 10 + y),
            BackColor = Color.FromArgb(107, 107, 107),
            ForeColor = Color.WhiteSmoke,
            Size = new Size(50, 50),
            Font = new Font("Segoe UI", 9F, FontStyle.Bold),
            FlatStyle = FlatStyle.Popup,
            BackgroundImage = Properties.Resources.koltuk,
            Text = j.ToString(),
            TextAlign = ContentAlignment.MiddleCenter,
        };
        Buton.Click += Buton_Click;
        j++;
        x += 55;
        KoltukPanel.Controls.Add(Buton);
        ButtonList.Add(Buton);
        return Buton;
    }
    k++;
    x = 0;
    y += 55;
    goto go;
    KoltukPanel.Controls.Add(Buton);
    ButtonList.Add(Buton);
    return Buton;
}

1 reference
private void Buton_Click(object? sender, EventArgs e)
1 reference
public void KoltukOlustur()
{
    SalonBilgisi();
    koltuklar = new int[koltuksirasi, sirakoltuksayisi];

    for (int i = 1; i <= koltuklar.Length; i++)
    {
        string ButtonName = "Koltuk " + i;
        AddNewButton(ButtonName);
    }
}

```

Resim7: Dinamik Butonlarla Koltuk Oluşumu

Satır-Sütun mantığı ile oluşturulan yapıda satırlar; koltuk sırası, sütunlar; sıradaki koltuk sayısıdır. $(\text{Koltuk sırası}) \times (\text{Sıradaki Koltuk Sayısı}) = \text{Salondaki Toplam Koltuk Sayısını}$ vermektedir.

Oluşan her butonlar beraber j değeri bir artmaktadır. Ayrıca oluşan her butonun bir diğeri arasında x mesafesi mevcuttur. Her Sıradaki Koltuk Sayısı dolduğunda koltuk sırasını temsil eden k değeri bir artmaktadır. Bir sonraki koltuk sırasına geçildiğinde x değeri sıfırlanır ve ilk sıra ile ikinci sıra arasında y değeri verilerek mesafe konulur.

Ancak her sıranın en sonunda oluşan koltuk buton değerini(yapısını) almadığı için goto yöntemi kullanılarak en sondaki değerin alınması sağlanmıştır.

Her yapılan oluşum işlemi Buton Listesine ve Panel Control'e eklenmektedir.

```

if @ToplamBakiye Is Null
begin
insert into Muhasebe(Tarih,Turu,Aciklama,Gelir,Gider,HesapBakiyesi,ToplamBakiye)
values(GETDATE(),'Hizmet','Etkinlik Maliyeti',0,@EtkinlikMaliyeti,-@EtkinlikMaliyeti)
end
else if @ToplamBakiye Is Not Null
begin
insert into Muhasebe(Tarih,Turu,Aciklama,Gelir,Gider,HesapBakiyesi,ToplamBakiye)
values(GETDATE(),'Hizmet','Etkinlik Maliyeti',0,@EtkinlikMaliyeti,-@EtkinlikMaliyeti,@ToplamBakiye-@EtkinlikMaliyeti)
set @ToplamBakiye-=@EtkinlikMaliyeti
end
if @ToplamBakiye Is Null
begin
insert into Muhasebe(Tarih,Turu,Aciklama,Gelir,Gider,HesapBakiyesi,ToplamBakiye)
values(GETDATE(),'Kira','Salon Kirası',0,@SalonKirasi,-@SalonKirasi,0-@SalonKirasi)]
end
else if @ToplamBakiye Is Not Null
begin
insert into Muhasebe(Tarih,Turu,Aciklama,Gelir,Gider,HesapBakiyesi,ToplamBakiye)
values(GETDATE(),'Kira','Salon Kirası',0,@SalonKirasi,-@SalonKirasi,@ToplamBakiye-@SalonKirasi)
end

```

Resim8: Is Null ve Is Not Null Yapısı

Eğer tablonuzda hiç veri yoksa veya verdiğiniz değerin null alıp alamayacağını tanımlamak için kullanılır.

Yukarıdaki yapıyı Türkçeleştirecek olursak:

Eğer ‘Toplam Bakiye’ değeri null ise insert into yapısını kullan [ToplamBakiye=0-EtkinlikMaliyeti]

Eğer değer null değilse insert into yapısını kullan [ToplamBakiye=ToplamBakiye-EtkinlikMaliyeti]

Buradaki set değeri ile en son elimize geçen Toplam Bakiye değerini alıyoruz yukarıdaki işlemin aynısını salon kirası için yapıyoruz.

```

foreach (var button in KoltukPanel.Controls.OfType<Button>())
{
    if ((sender as Button).Text == button.Text)
    {
        if ((sender as Button).Name == "0")
        {
            DialogResult koltuksec = new DialogResult();
            koltuksec = MessageBox.Show("Koltuk Eklensin Mi?", "Koltuk Seçme İşlemi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

            if (koltuksec == DialogResult.Yes)
            {
                ksecim.Add((sender as Button).Text);

                for (int j = 0; j < ksecim.Count; j++)
                {
                    KoltuklarLbl.Text += ksecim[j] + " - ";
                    secilenkoltuklar = KoltuklarLbl.Text;
                    (sender as Button).BackgroundImage = Properties.Resources.sari_koltuk;
                }

                AdetNUD.Value += 1;
                toplam += Convert.ToInt32(fiyat);
                ToplamLbl.Text = "Toplam: " + toplam.ToString();
                button.Name = "1";

                MessageBox.Show("Koltuk Seçimi Başarılı");
            }
            else
            {
                MessageBox.Show("Koltuk Seçimi İptal Edildi.");
            }
        }
        else if ((sender as Button).Name == "1")
        {
            // ...
        }
        else if ((sender as Button).Name == "2")
        {
            // ...
        }
    }
}

```

Resim9: Buton Kontrolü İle İşlem Yapma

Bilet İşlemleri kısmında panelin içine dinamik olarak oluşan salona ait koltukları temsil eden butonlar mevcuttur. İlk oluşumlarında Name olarak “0” atanmakta ve BackgroundImage kısmına kırmızı koltuk yüklenmektedir. Name’i “0” olan butona tıkladığımızda seçilme işlemi yapılır ve Name “1” olarak değiştirilir. BackgroundImage’e sarı koltuk atanır. Seçilen koltuğu geri bırakmak için Name’i “1” olan butona tıklanmalıdır. İşlem yapılırken BackgroundImage tekrar kırmızı koltuk olacaktır.

Name’i 1 olduktan sonra bilet alma işlemi yapılır ve Name “2” olarak atanır, BackgroundImage ise yeşil koltuk olur. Satın alınan koltuk iade edilmesi içinse Name’i “2” olan butona tıklanır ve iade işlemi yapılır. BackgroundImage ise tekrar kırmızı koltuğa döner.

Satın alınan koltuk numaraları veri tabanında saklanmakta ve iade edildiğin hem bilet satış tablosundaki verisi revize edilmekte hem de bilet iade tablosuna eklenmektedir.

4. Sonuç ve Tartışma

Sonuç, araştırma yapılmasına ve yöntem geliştirmesine olanak sağlayan bu proje ekstra görsel bir çalışma olmaksızın tamamlanmıştır.

5. Öneriler

Projenin geliştirilmesinde etkinlikler için görsel veriler eklenerek etkinlik like araması ile birleştirip otomatik olarak ana ekrana gelmesi sağlanılabilir.