

## 1. Rename a variable with a clearer or more informative name

Ovaj refaktoring smo iskoristili za varijable koje smo prvobitno “loše” imenovali, te smo im sada promijenili imena tako da budu jasnija i pružaju više informacija.

```
private int kID;  
private int id;  
private string vr;  
private List<int> ocj;  
private DateTime procj;  
private string lok;  
  
public class Dojava  
{  
    private int korisnikID;  
    private int id;  
    private string vrsta;  
    private List<int> ocjene;  
    private DateTime procjenaCekanja;  
    private string lokacija;
```

Slika1. Klasa dojava prije i poslije refaktoringa

```
public class Korisnik  
{  
    int id;  
    string ime;  
    string pr;  
    string email;  
    string user;  
    string pass;  
    int br;  
    int brojAkta;  
    ImageSource sl;  
    Boolean premiumK;  
  
public class Korisnik  
{  
    int id;  
    string ime;  
    string prezime;  
    string email;  
    string username;  
    string password;  
    int brojDojava;  
    int brojAktivnihDojava;  
    ImageSource slika;  
    Boolean premiumKorisnik;
```

Slika 2. Klasa korisnike prije i poslije refaktoringa

## 2. Move a complex boolean expression into a well-named boolean function

Ovaj refaktoring smo iskoristili na dva načina.

Prvi način je da smo za komplikovan, dug i slabo čitljiv izraz u if iskazu zamijenili funkcijom rad lakšeg snalaženja u kodu.

```
if (textIme.Text == string.Empty || textPrezime.Text == string.Empty || textEmail.Text == string.Empty || textUser.Text == string.Empty || textPass1.Password == string.Empty)  
{  
    textPostoji.Text = "Molimo unesite sve tražene podatke.";   
    return;  
}
```

Slika 3. Komplikovan izraz u if iskazu

```
if (prazan())  
{  
    textPostoji.Text = "Molimo unesite sve tražene podatke.";   
    return;  
}
```

Slika 4. Zamjena izraza funkcijom-refaktoring

Sljedeće mjesto gdje smo uočili dobar primjer ovog refaktoringa je mjesto u kojem nam se dug izraz u if iskazu ponavlja više puta.

```

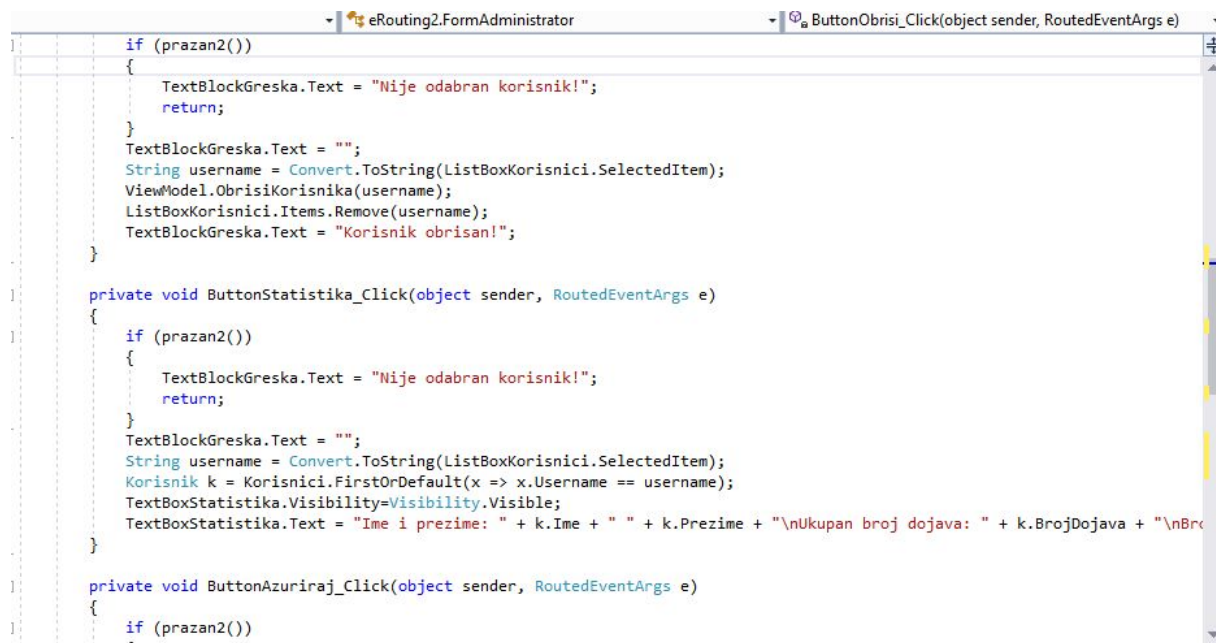
{
    if (ListBoxKorisnici.Items.Count == 0 || ListBoxKorisnici.SelectedItem==null)
    {
        TextBlockGreska.Text = "Nije odabran korisnik!";
        return;
    }
    TextBlockGreska.Text = "";
    String username = Convert.ToString(ListBoxKorisnici.SelectedItem);
    ViewModel.ObrisiKorisnika(username);
    ListBoxKorisnici.Items.Remove(username);
    TextBlockGreska.Text = "Korisnik obrisani!";
}

private void ButtonStatistika_Click(object sender, RoutedEventArgs e)
{
    if (ListBoxKorisnici.Items.Count == 0 || ListBoxKorisnici.SelectedItem == null)
    {
        TextBlockGreska.Text = "Nije odabran korisnik!";
        return;
    }
    TextBlockGreska.Text = "";
    String username = Convert.ToString(ListBoxKorisnici.SelectedItem);
    Korisnik k = Korisnici.FirstOrDefault(x => x.Username == username);
    TextBoxStatistika.Visibility=Visibility.Visible;
    TextBoxStatistika.Text = "Ime i prezime: " + k.Ime + " " + k.Prezime + "\nUkupan broj dojava: " + k.BrojDojava + "\n"
}

private void ButtonAzuriraj_Click(object sender, RoutedEventArgs e)
{
    if (ListBoxKorisnici.Items.Count == 0 || ListBoxKorisnici.SelectedItem == null)
    {
        TextBlockGreska.Text = "Nije odabran korisnik!";
    }
}

```

Slika 5. Kod prije refaktoringa



```

- eRouting2.FormAdministrator
- ButtonObrisi_Click(object sender, RoutedEventArgs e)

if (prazan2())
{
    TextBlockGreska.Text = "Nije odabran korisnik!";
    return;
}
TextBlockGreska.Text = "";
String username = Convert.ToString(ListBoxKorisnici.SelectedItem);
ViewModel.ObrisiKorisnika(username);
ListBoxKorisnici.Items.Remove(username);
TextBlockGreska.Text = "Korisnik obrisani!";
}

private void ButtonStatistika_Click(object sender, RoutedEventArgs e)
{
    if (prazan2())
    {
        TextBlockGreska.Text = "Nije odabran korisnik!";
        return;
    }
    TextBlockGreska.Text = "";
    String username = Convert.ToString(ListBoxKorisnici.SelectedItem);
    Korisnik k = Korisnici.FirstOrDefault(x => x.Username == username);
    TextBoxStatistika.Visibility=Visibility.Visible;
    TextBoxStatistika.Text = "Ime i prezime: " + k.Ime + " " + k.Prezime + "\nUkupan broj dojava: " + k.BrojDojava + "\nBroj"
}

private void ButtonAzuriraj_Click(object sender, RoutedEventArgs e)
{
    if (prazan2())
    {
        TextBlockGreska.Text = "Nije odabran korisnik!";
    }
}

```

Slika 6. Kod nakon refaktoringa

### 3. Facade pattern

Ovaj pattern smo iskoristili jer smo imali više podsistema čije su implementacije usko povezane, što smo objedinili u jednu klasu.

```

namespace eRouting2.ViewModel
{
    class UpravljanjeRacunima
    {
        Administrator admin;
        DBAdministrator bazaAdmin;
        DBKorisnik bazaKorisnik;

        public UpravljanjeRacunima(Administrator admin, DBAdministrator bazaAdmin, DBKorisnik bazaKorisnik)
        {
            Admin = admin;
            BazaAdmin = bazaAdmin;
            BazaKorisnik = bazaKorisnik;
        }

        public Administrator Admin { get => admin; set => admin = value; }
        internal DBAdministrator BazaAdmin { get => bazaAdmin; set => bazaAdmin = value; }
        internal DBKorisnik BazaKorisnik { get => bazaKorisnik; set => bazaKorisnik = value; }
        public int UnesiKorisnika(Korisnik k)...
        public void ObrisiKorisnika(Korisnik k)...
        public int unesiAdministratora(Administrator a)...
    }
}

```

Slika 7. Kod nakon facade patterna