



piLogin.dll komponenta - Upute

SADRŽAJ

Podaci	1
Korisnik	2
Akcija	3
Uloga	3
piLogin.dll	5
Demo aplikacija	9
Pomoć	21

UVOD

piLogin komponenta je napravljena kako bi studentima pružila izvor podataka za njihove projekte na kolegiju Programsko inženjerstvo. Uz sam izvor podataka, komponenta pruža i osnovnu funkcionalnost provjere autentičnosti korisnika, odnosno njegovo identificiranje sa korisničkim imenom i lozinkom.

Komponentni razvoj aplikacija je trend već nekoliko desetljeća, tako da ova mala komponenta ne bi trebala predstavljati veliki problem i izazov. Kako bi korištenje komponente proteklo glatko uz nju dolazi i pripadajući *intellisense*, msdn dokumentacija, izvorni kôd demo projekta te ovaj dokument.

gl hf.



PODACI

Podaci u aplikaciji su organizirani kao liste/popis korisnika, pripadajućih uloga i na kraju pripadajuće akcije za svaku ulogu.

Korisnik

Podaci su strukturirani tako da svaki korisnik sustava može imati jednu ili više uloga. Svaki korisnik je jedinstveno identificiran svojim korisničkim imenom. Uz korisničko ime je pohranjena i njegova lozinka te ime, prezime broj telefona, adresa, zvanje, zanimanje i email.

U nastavku slijedi popis korisnika, gdje je prva vrijednost redni broj (nije bitan) zatim slijedi korisničko ime i lozinka odvojeni zarezom (,)

POPIS KORISNIKA (USERNAME, LOZINKA):

1. btomas, btopass
2. zstapic, zstpass
3. rpicek, rpipass
4. nastavnik, naspass
5. tajnik, tajpass
6. dekan, dekpas
7. prodekan, propass
8. racunovoda, racpass
9. uljez, uljpass

Napomena: korisnik sa oznakom **uljez** je uljez u sustavu te je njegova uloga postavljena u None (vidi uloge). Korisnik je definiran sučeljem *iUser*.



Akcije

Sustav za prijavu definira za svaku ulogu set akcija koje odgovarajući korisnik može izvršiti u aplikaciji. U nastavku slijedi popis svih akcija koje se mogu pojaviti, redni broj nije dio enumeracije:

POPIS AKCIJA

1. **Create** – korisnik može kreirati novi nalog
2. **Read** – korisnik može čitati naloge
3. **Write** – korisnik može mijenjati sadržaj postojećeg naloga
4. **Delete** – korisnik može obrisati nalog
5. **Approve** – korisnik može odobriti nalog
6. **Liquidate** – korisnik može likvidirati nalog
7. **Sign** – korisnik može potpisati nalog.
8. **Grant** – korisnik može svoje ovlasti dodijeliti drugom korisniku.
9. **None** – korisnik nema ovlasti.

Akcije su definirane enumeracijom *eAction*.

Uloge

Uloga je osnovni kriterij po kojem se pojedinom korisniku dodjeljuju prava korištenja dijelova aplikacije. Svaka uloga je jedinstveno identificirana sa svojim imenom (rolename)

U nastavku slijedi popis uloga, gdje je prva vrijednost redni broj (nije bitan), zatim slijedi ime uloge i maska aktivnost (odvojeno zarezom)

POPIS ULOGA(NAZIV ULOGE (AKCIJA1, AKCIJA2,...))

1. Nastavnik (Read, Write)
2. Tajnik (Create, Read, Delete)
3. Dekan (Read, Approve, Grant)
4. Prodekan (Read, Sign, Grant)
5. Racunovoda (Read, Write, Delete, Liquidate)
6. None ()

Uloge su definirane sučeljem *iRole*. Primjetimo da uloga *None* nema nikakvih prava i kao takvu ju se treba tretirati.



U nastavku slijedi popis u kojem su definirani odnosi korisnika i uloga u sustavu. Prikazani su korisničko ime i naziv uloge kojoj korisnik pripada, odvojeni zarezom (,).

KORISNICI I ULOGE (KORISNIČKO IME, NAZIVI ULOGA)

1. btomas, Nastavnik
2. zstapic, Nastavnik
3. rpicek, Nastavnik
4. nastavnik, Nastavnik
5. tajnik, Tajnik
6. dekan, Nastavnik, Dekan
7. prodekan, Nastavnik, Prodekan
8. racunovoda, Racunovoda
9. uljez, None



PILOGIN.DLL

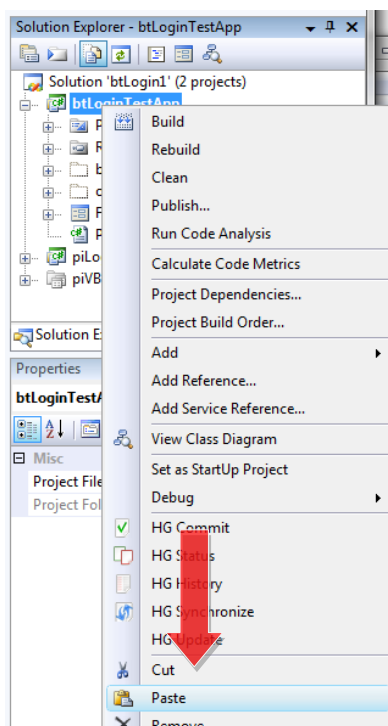
Sama piLogin komponenta se sastoji od sljedećih datoteka:

1. piLogin.dll
2. piLogin.xml
3. piDB3.s3db

Prva datoteka je osnovna datoteka u kojoj se nalazi sav koristan kod. Druga datoteka sadrži *Intellisense* za komponentu koji je iscrpan i sadrži sve potrebne opise. Preporuča se koristiti *intellisense* kod rada s komponentom jer se time uvelike olakšava i ubrzava razvoj aplikacije. Treća datoteka je jednostavna baza podataka koju koristi komponenta. Sve tri datoteke trebaju biti zajedno na zajedničkoj lokaciji.

Komponenta se u projekt uključuje na sljedeći način:

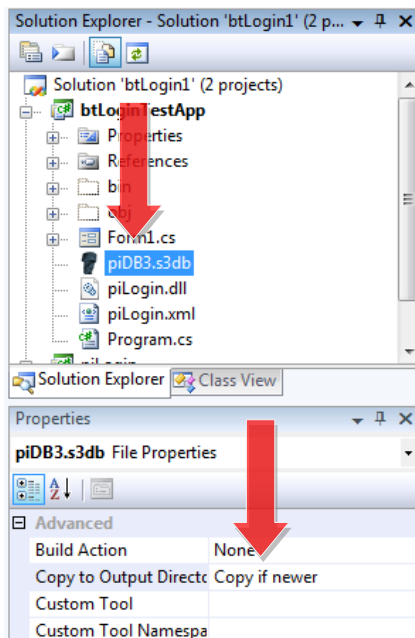
1. Otvorite novi Windows Application VB.Net projekt
2. Kopirajte u clipboard sve tri datoteke.
3. Na projekt, **ne solution**, stisnite desnim klikom miša i odaberite „Paste“:



Slika 1 Paste lokacija

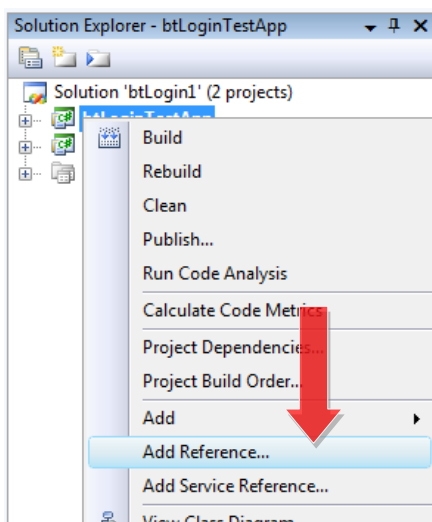


4. Nakon toga u projektu se nalaze tri datoteke. Kliknite na piDB3.s3db datoteku u projektu te u postavkama datoteke odaberite „Copy if newer“:



Slika 2 Copy if newer

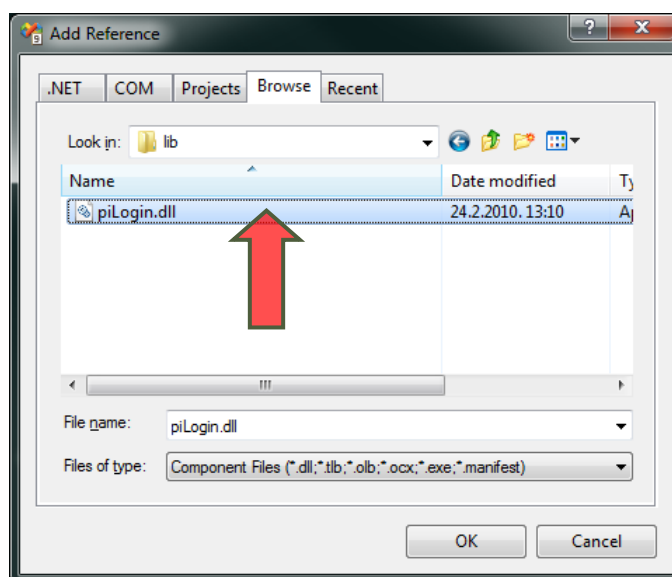
4. Na projekt, **ne solution**, stisnite desnim klikom miša i odaberite „Add Reference“:



Slika 3 Add reference - lokacija

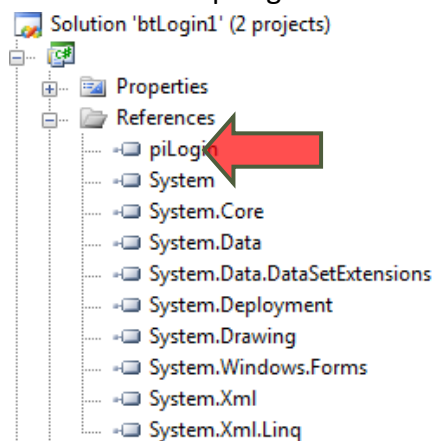


5. Nakon toga se prikazuje sljedeći prikaz:



Slika 4 Add reference dijalog

6. Odaberite Browse tab u kojem potražite **piLogin.dll** datoteku (vidi sliku iznad).
Napomena: Jednom postavljenu, fizičku lokaciju .dll datoteke nemojte mijenjati, zbog dobre prakse stavite .dll datoteku u direktorij gdje vam je projekt (korak 3), zatim prođite ovaj korak.
7. Pritisnite OK.
8. Nakon toga u popisu referenci bi trebao biti piLogin referenca (vidi sliku ispod).



Slika 5 popis referenci

9. Komponenta je dodana u projekt.
Kako bi već sada vidjeti što se nalazi u toj komponenti potrebno ju je označiti i pokrenuti „Object Explorera“ pritiskom na tipku F2.



Za početak je preporučljivo **napraviti jednu instancu** te **uključivanje *foi.pi* namespacea** u projekt, to sve izgleda , u pravilu, ovako:

```
Imports foi.pi 'uključivanje namespacea
Public Class piMainForm
    Public piLoginInstance As Login = New Login() 'javno instanciranje objekta
    Login klase iz piLogin komponente

    Private Sub piMainForm_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        End Sub

End Class
```

piLogin komponenta posjeduje sljedeće javne klase, pobrojenja (enumeracije) i sučelja:

1. eActions – pobrojenje svih akcija
2. eGrant – pobrojenje za dodjelu uloga
3. iRole – sučelje koje predstavlja ulogu
4. iUser – sučelje koje predstavlja korisnika
5. Login – glavna klasa komponente



DEMO APLIKACIJA

Za potrebe kolegija izrađena je demo aplikacija koja koristi piLogin.dll komponentu. U aplikaciji su obrađeni svi scenariji korištenja s kojima se studenti mogu susresti. U nastavku će biti opisan svaki segment aplikacije, izostavljeni su elementi koji su specijalizirani za dizajn aplikacije, dakle, naglasak je na prikaz onog kôda kojim se direktno koristi *piLogin* komponenta.

Slika 6 Sučelje demo aplikacije



Sučelje aplikacije se sastoji od ovih elemenata (kako prikazuje Slika 6):

1. Osnovna prijava
2. Ispis korisnika
3. Ispis uloga
4. Ispis akcija
5. Pretraga korisnika
6. Ispis uloga za korisnika
7. Ispis korisnika u ulozi
8. Ispis akcija za ulogu
9. Pretraga uloga

U nastavku slijedi detaljni opis i implementacija svakog.

OSNOVNA PRIJAVA

Basic Login

Username:

Password:

Result: OK

Login user's details, roles and actions:

name = Nastavnik
surname = Dobri
phone = +3859996656
address = Pavlinska 2
title = dr.sc.

Slika 7 Osnovna prijava

Gornja slika prikazuje osnovnu prijavu korisnika. Korisnik unosi svoje korisničko ime i lozinku, pritiskom na tipku *Login* se prikazuje odgovarajuća poruka u *Result* polju: OK ili Fail. Desno se nalaze informacije o prijavljenom korisniku kao i uloga u kojoj se nalazi „prijavljen“ korisnik te akcije (u zagradama) za svaku ulogu kojoj korisnik pripada.



Pojednostavljeni kôd koji se izvršava nakon pritiska na tipku *Login*:

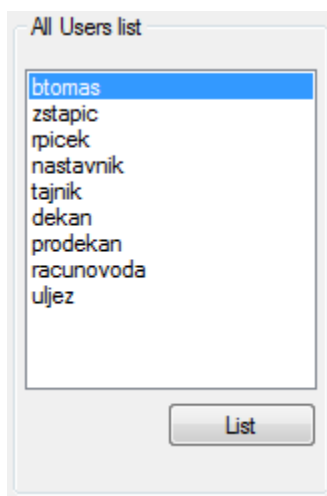
```
piLBBasicRolesList.Items.Clear()  
Dim tmpUser As IUser =  
piLoginInstance.Authenticate(piTxtUserName.Text, piTxtPassword.Text)  
If tmpUser Is Nothing Then  
    piLblResult.Text = "Fail"  
Else  
    piLblResult.Text = "OK"  
    piLBBasicRolesList.Items.Add("name = " & tmpUser.Name)  
    piLBBasicRolesList.Items.Add("surname = " & tmpUser.Surname)  
    piLBBasicRolesList.Items.Add("phone = " & tmpUser.Phone)  
    piLBBasicRolesList.Items.Add("address = " & tmpUser.Address)  
    piLBBasicRolesList.Items.Add("title = " & tmpUser.Title)  
    piLBBasicRolesList.Items.Add("occupation = " &  
tmpUser.Occupation)  
    piLBBasicRolesList.Items.Add("email = " & tmpUser.Email)  
    piLBBasicRolesList.Items.Add("Roles (Actions):")  
    For Each tmprole In tmpUser.AssignedRoles  
        piLBBasicRolesList.Items.Add(tmprole.RoleName & " (" &  
tmprole.Actions.ToString() & ")")  
    Next  
End If
```

Gornji kôd radi sljedeće: U varijablu tipa *IUser* pohranjuje se rezultat poziva metode *Authenticate(string, string)* Ova metoda „vrati“ odgovarajući objekt tipa *IUser* odnosno *Nothing* ukoliko korisničko ime i lozinka ne odgovaraju. Nakon uspješne prijave za prijavljenog korisnika se ispisuju njegove uloge i akcije za svaku ulogu. *IUser* sučelje sadrži *AssignedRoles* polje u kojem se nalazi lista svih uloga u kojima se korisnik nalazi. Prikazano je korištenje For Each sintakse za iteraciju kroz sve elemente *AssignedRoles* liste.

Napomena: *piLoginInstance* je instancirani objekt tipa *piLogin* klase (vidi *piLogin.dll* sekciju).



ISPIS KORISNIKA



Slika 8 Ispis korisnika

Gornja slika prikazuje dio demo aplikacije koji ispisuje sve korisnike u sustavu pritiskom na tipku *List*.

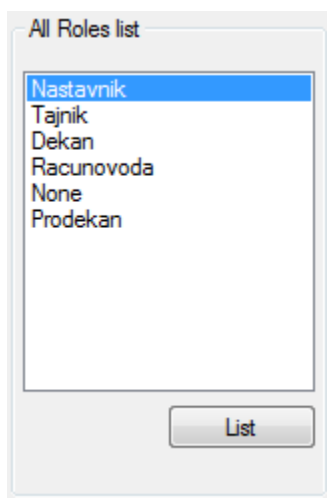
Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
For Each tmpusr In piLoginInstance.GetUser()  
    piLBUsersList.Items.Add(tmpusr.UserName)  
Next
```

Gornji kôd radi sljedeće: Koristeći *For Each* naredbu prelistavaju se svi korisnici sustava te se zapisuju u obični *ListBox* objekt (*piLBUsersList*). Korisnici se dohvaćaju uz pomoć *GetUser()* metode koja vraća listu čiji elementi su tipa *iUser*.



ISPIS ULOGA



Slika 9 Ispis uloga

Gornja slika prikazuje dio demo aplikacije koji ispisuje sve moguće uloge koje može imati korisnik u sustavu.

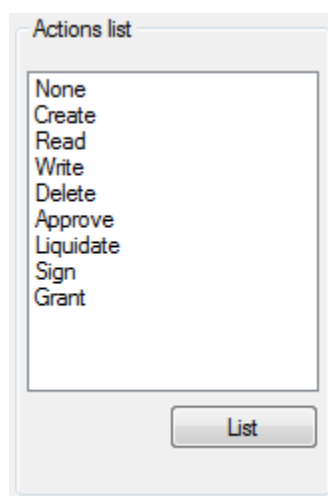
Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
For Each tmprole In piLoginInstance.GetRoles()  
    piLBRolesList.Items.Add(tmprole.RoleName)  
Next
```

Gornji kôd radi sljedeće: Koristeći *For Each* naredbu prelistavaju se sve uloge u sustavu te se zapisuju u obični *ListBox* objekt. Uloge se dohvaćaju uz pomoć *GetRoles()* metode koja vraća listu čiji elementi su tipa *iRole*.



ISPIS AKCIJA



Slika 10 Ispis akcija

Gornja slika prikazuje dio demo aplikacije koji ispisuje sve moguće akcije koje može imati uloga u sustavu.

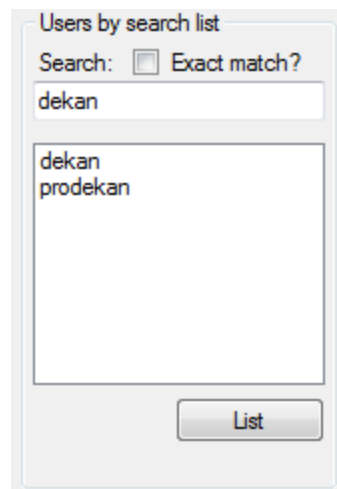
Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
For Each tmpAction In [Enum].GetNames(GetType(foi.pi.eActions))  
    piLBActionsList.Items.Add(tmpAction)  
Next
```

Gornji kôd radi sljedeće: Koristeći *For Each* naredbu prelistavaju se sve akcije u sustavu te se zapisuju u obični *ListBox* objekt. Akcije su definirane u *enum* pobrojenju. Prikazani kôd je napredni i vrlo je mala vjerojatnost da će se studenti susresti s istim. Za naprednije, klasa *Enum* sadrži *GetNames()* metodu koja kao parametar prima *Enum* tip podataka koji je u primjeru dohvaćen sa *GetType()* metodom. *GetNames()* vraća listu odnosno polje string varijabli u kojima se nalazi definirano ime svakog segmenta *eActions* pobrojenja. Dohvaćanje samih vrijednosti *eActions* pobrojenja se postiže *GetValues()* metodom.



PRETRAGA KORISNIKA



Slika 11 Pretraga korisnika

Gornja slika prikazuje dio demo aplikacije koji prikazuje rezultate pretrage za korisnike prema korisničkom imenu.

Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
For Each tmpusr In  
    piLoginInstance.GetUser(piTxtSearchUsername.Text,  
        piCBUserExactMatch.Checked)  
        piLBUsersListBySearch.Items.Add(tmpusr.UserName)  
Next
```

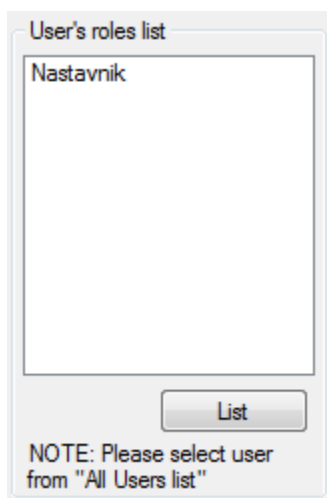
Gornji kôd radi sljedeće: Koristeći For Each naredbu prelistava se lista korisnika odnosno *iUser* objekata. Sadržaj liste je uvjetovan pozivom *GetUser(string, bool)* metode koja kao parametre prihvaća, redom: tekst koji se pretražuje, te bool vrijednost kojom se određuje stil pretraživanja odnosno striktnost. Gornja slika prikazuje slučaj u kojem se kao drugi parametar prosljeđuje vrijednost *false* pa je rezultat dekan i prodekan. Kada bi taj parametar bio *true* (označen checkBox) tada bi rezultat bila lista samo sa jednim elementom (dekan).

Kako provjeriti da li lista sadrži rezultat (nije u demo aplikaciji):

```
If piLoginInstance.GetUser(piTxtSearchUsername.Text,  
    piCBUserExactMatch.Checked).Count > 0 'postoji barem jedan rezultat  
End If
```



ISPIS ULOGA ZA KORISNIKA



Slika 12 Ispis uloga za korisnika

Gornja slika prikazuje dio demo aplikacije koji prikazuje uloge za odabranog korisnika u ListBox elementu u „All Users list“ dijelu aplikacije (Ispis korisnika).

Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

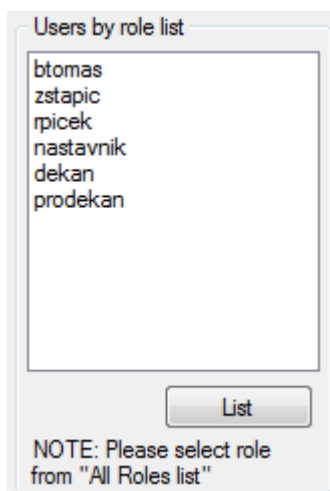
```
For Each tmprole In  
piLoginInstance.GetUser(piLBUsersList.SelectedItem.ToString(),  
True) (0).AssignedRoles  
                piLBUsersRolesList.Items.Add(tmprole.RoleName)  
Next
```

Gornji kôd radi sljedeće: Koristeći For Each naredbu prelistava se lista korisnika odnosno *iUser* objekata. Sadržaj liste je uvjetovan pozivom *GetUser(string, bool)*. Tehnički se ovdje poziva preopterećena *GetUser()* metoda koja je opisana u djelu u kojem se pretražuju korisnici. Kao parametar ova metoda prima vrijednost odabranog reda (korisnika) iz ListBoxa u kojem se nalazi popis svih korisnika. Prije poziva ove metode potrebno je provjeriti da li je odabran samo jedan korisnik iz *List*Box elementa:

```
If piLBUsersList.SelectedItems.Count = 1 Then  
    'odabran je samo jedan korisnik, dakle može se nastaviti.  
End If
```




ISPIS KORISNIKA U ULOZI



Slika 13 Ispis korisnika u ulozi

Gornja slika prikazuje dio demo aplikacije koji prikazuje korisnike za odabranu ulogu u ListBox elementu u „All Roles list“ dijelu aplikacije (Ispis uloga).

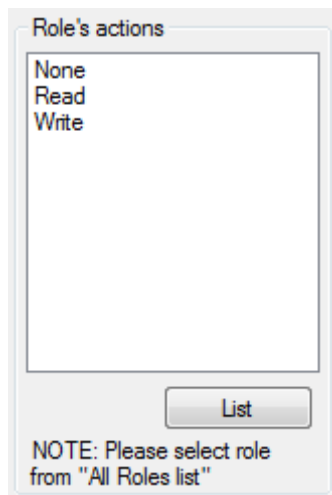
Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
For Each tmpusr In  
    piLoginInstance.GetUser(piLoginInstance.GetRoles(piLBRolesList.SelectedItem.  
        ToString()), True)(0))  
        piLBUsersListByRole.Items.Add(tmpusr.UserName)  
Next
```

Gornji kôd radi sljedeće: Koristeći For Each naredbu prelistava se lista korisnika odnosno *iUser* objekata. Sadržaj liste je uvjetovan pozivom *GetUser(iRole)*. Tehnički se ovdje poziva preopterećena *GetUser()* metoda koja je opisana u djelu u kojem se pretražuju korisnici. Kao parametar ova metoda prima vrijednost odabranog reda (uloge) iz ListBoxa u kojem se nalazi popis svih uloga, no prije toga, poziva se metoda *GetRoles()*. Nakon poziva prikazane *GetUser(iRole)* metode vraća se lista korisnika koji se nalaze u proslijeđenoj ulozi (*iRole*).



ISPIS AKCIJA ZA ULOGU



Slika 14 Ispis akcija za ulogu

Gornja slika prikazuje dio demo aplikacije koji prikazuje akcije za odabranu ulogu u ListBox elementu u „All Roles list“ dijelu aplikacije (Ispis uloga).

Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
Dim tmpAction As eActions =  
piLoginInstance.GetRoles(piLBRolesList.SelectedItem.ToString(),  
True)(0).Actions  
For Each tmpAllAct In [Enum].GetValues(GetType(foi.pi.eActions))  
    If (tmpAction Or tmpAllAct) = tmpAction Then  
        piLBRolesActionList.Items.Add(tmpAllAct)  
    End If  
Next
```

Gornji kôd on radi sljedeće: Koristeći For Each naredbu prelistava se lista naziva akcija koristeći već opisanu tehniku sa *[Enum]* klasom. U ovom slučaju se poziva *GetValues()* metoda kako bi se dohvatilo polje vrijednosti *eActions* pobrojenja. Svaki element tog polja se uspoređuje sa akcijama odabrane uloge u „All Roles list“ djelu. Studenti se sa funkcionalnošću ovog kôda ne bi trebali susresti.



PRETRAGA ULOGA

Roles by search list

Search: ☐ Exact match?

nik

Nastavnik
Tajnik

List

Slika 15 Pretraga uloga

Gornja slika prikazuje dio demo aplikacije koji prikazuje rezultate pretrage za uloge prema nazivu uloge.

Pojednostavljeni kôd koji se izvršava pritiskom na tipku *List*:

```
For Each tmprole In piLoginInstance.GetRoles(piTxtSearchRolename.Text,  
piCBRoleExactMatch.Checked)  
    piLBRolesListBySearch.Items.Add(tmprole.RoleName)  
Next
```

Gornji kôd radi sljedeće: Koristeći For Each naredbu prelistava se lista uloga odnosno *iRole* objekata. Sadržaj liste je uvjetovan pozivom *GetRole(string, bool)* metode koja kao parametre prihvaća: tekst koji se pretražuje, te bool vrijednost kojom se određuje stil pretraživanja odnosno striktnost. Gornja slika prikazuje slučaj u kojem se kao drugi parametar prosljeđuje vrijednost *false* pa je rezultat Nastavnik i Tajnik jer oboje sadrže riječ „nik“. Kada bi taj parametar bio *true* (označen checkBox) tada bi rezultat bila prazna lista.



DODJELA ULOGA

Ovaj dio se ne nalazi u demo aplikaciji. Logika je da korisnik koji ima ovlasti može dodijeliti svoju ulogu drugom korisniku.

Logiku tko kome i kako dodjeljuje ovlasti određuje programer, konkretno korisnik drugom korisniku može dodijeliti samo vlastite uloge i to na određeni vremenski rok.

Komponenta ovim mehanizmom omogućuje dodjelu svih ovlasti svim korisnicima.

Pojednostavljeni kôd za dodjelu ovlasti „Dekan“ korisniku „btomas“:

```
piLoginInstance.Grant(piLoginInstance.GetUser("btomas", True)(0),  
piLoginInstance.GetRoles("Dekan", True)(0), foi.pi.eGrant.Add)
```

Pojednostavljeni kôd za brisanje ovlasti „Dekan“ korisniku „btomas“:

```
piLoginInstance.Grant(piLoginInstance.GetUser("btomas", True)(0),  
piLoginInstance.GetRoles("Dekan", True)(0), foi.pi.eGrant.Remove)
```

Gornja dva kôda rade sljedeće: Koristeći već poznate metode za pretraživanje korisnika odnosno uloga, pridružuju korisniku novu ulogu odnosno briše korisnika iz uloge. Prije toga je potrebno provjeriti da li „prijavljeni“ korisnik ima ulogu sa mogućnošću *Grant* opcije, te na kraju provjeriti da li je uloga koja se dodjeljuje ista onoj u kojoj se korisnik nalazi.

Napomena: Programer je dužan voditi brigu o tome kome i na koji rok je dodijeljena odgovarajuća uloga, nakon isteka roka dodijele dodijeljenu ulogu je potrebno ukloniti.

Savjet: Obzirom da se sve izmjene u korisničkim ulogama moraju dogoditi i na vašim lokalnim bazama podataka (koje su medij za komunikaciju sa drugim aplikacijama), napredniji studenti bi mogli zaključiti kako je ovaj dio komponente u neku ruku nepotreban što je i donekle istinito.



POMOĆ

Sve eventualne pogreške i pitanja dojavite na email asistenata. Probleme i nejasnoće u korištenju prijavite na moodle forumu kako bi i ostali studenti mogli sudjelovati.