Univerzitet u Novom Sadu

Tehnički fakultet „Mihajlo Pupin“

Zrenjanin

Predmet: Projektovanje softvera

Seminarski rad

*WPF aplikacija – Muzička gimnazija*

Profesor: Student:

[Doc. dr Eleonora Brtka](http://www.tfzr.uns.ac.rs/Kadar/NastavnoOsoblje/20) Lara Milutinović SI 29/17

Smer: Softversko inženjerstvo

Asistent:

Msc Siniša Mihajlović

Zrenjanin, 2019.

SADRŽAJ

[Predmetni cilj 3](#_Toc9933147)

[Cilj seminarskog rada 4](#_Toc9933148)

[1. Uvod 5](#_Toc9933149)

[2. Korišćene tehnologije 7](#_Toc9933150)

[2.1. Microsoft Visual Studio 7](#_Toc9933151)

[2.2. C# programski jezik 7](#_Toc9933152)

[2.3. WPF 8](#_Toc9933153)

[2.4. Microsoft SQL Server 8](#_Toc9933154)

[3. Postupak kreiranja aplikacije 10](#_Toc9933155)

[3.1. Kreiranje baze podataka 10](#_Toc9933156)

[3.2. Kreiranje konekcionog stringa 12](#_Toc9933157)

[3.3. Prozor MainWindow 13](#_Toc9933158)

[3.4. Prozor Gimnazijaa 14](#_Toc9933159)

[3.5. Prozor Ucenik 14](#_Toc9933160)

[3.6. Prozor Nastavnik 15](#_Toc9933161)

[3.7. Prozor Predmet 16](#_Toc9933162)

[3.8. Prozor Dnevnik 16](#_Toc9933163)

[4. Ključni delovi koda 18](#_Toc9933164)

[5. Zaključak 20](#_Toc9933165)

[LITERATURA 21](#_Toc9933166)

# Predmetni cilj

Osnovni cilj predmeta je sinteza stečenih znanja za kreiranje efikasnog softvera koji uključuje arhitekturu softvera, implementaciju, testiranje i kreiranje dokumentacije za softverske sisteme. Poseban cilj predmeta je osposobljavanje studenata za samostalnu konstrukciju softvera koja je bazirana na standardnim procesima implementacije, testiranja i kreiranje projektne dokumentacije.

# Cilj seminarskog rada

Cilj seminarskog rada je omogućiti unos gimnazije, nastavnika, učenika, predmeta, i dnevnika.

Na ovaj način se može videti kojoj gimnaziji pripada određeni učenik. Pored toga se može videti kojoj gimnaziji pripada određeni nastavnik, koji predmet ima određeni učenik kao i njegove ocene u dnevniku.

# 1. Uvod

Aplikacija se sastoji od skupa tabela koje su međusobno povezane u jednu bazu podataka. Svaka tabela ima svoj naziv, i podatke koji su određenog tipa koji čine datu tabelu koja se nalazi u bazi podataka. Tabele koje se nalaze u bazi su: *T\_Gimnazija*, *T\_Ucenik*, *T\_Nastavnik,*  *T\_Predmet* i *T\_Dnevnik.*

Podaci tabele *T\_Gimnazija* su: *IdGimnazije*, *IdUcenika*, *Naziv*, *Grad*, *Adresa*, *email*. Podatak *IdGimnazije* je primarni ključ tabele *T\_Gimnazija* i ima vrednost *IDENTITY*, što znači da se vrednost *IdGimnazije* povećava za jedan za svakou novu unetu gimnaziju. Podaci *Naziv*, *Grad*, *Adresa* i *email* se unose sami, dok je podatak *IdUcenika* strani ključ u tabeli *T\_Gimnazija*. Tabela *T\_Gimnazija* je povezana sa tabelom *T\_Ucenik* vezom jedan prema više.U tabelu T\_Gimnazijaa se moraju biti uneti podaci. Podatak *Naziv* predstavlja naziv gimnazije, podatak *Grad* predstavnja grad odakle je učenik, podatak *Adresa* predstavlja adresu prebivalista učenika i podatak *email* e-mail učenika.

Podaci tabele *T\_Ucenik* su: *IdUcenika*, *ImeUcenika*, *PrezimeUcenika* i *Instrument*. Podatak *IdUcenika* je primarni ključ tabele *T\_Ucenik* i ima vrednost *IDENTITY*, što znači da se vrednost *IdUcenika* povećava za jedan za svakog novog unetog učenika. Podaci *ImeUcenika, PrezimeUcenika i Instrument* se unose sami. Tabela *T\_Ucenik* je povezana sa tabelom *T\_Dnevnik* i tabelom *T\_Gimnazija* vezom jedan prema više. Podatak *ImeUcenika* predstavlja ime učenika, podatak *PrezimeUcenika* predstavlja prezime učenika, podatak *Instrument* predstavlja instrument koji učenik svira.

Podaci tabele *T\_Nastavnik* su: *IdNastavnika*, *ImeNastavnika*, *PrezimeNastavnika* i *email*. Podatak *IdNastavnika* je primarni ključ tabele *T\_Nastavnik* i ima vrednost *IDENTITY*, što znači da se vrednost *IdNastavnika* povećava za jedan za svakog novog unetog nastavnika. Podaci *ImeNastavnika, PrezimeNastavnika i email* se unose sami. Tabela *T\_Nastavnik* je povezana sa *T\_Dnevnik* vezom jedan prema više. Podatak *ImeNastavnika* predstavlja ime nastavnika, podatak *PrezimeNastavnika* predstavlja prezime nastavnika i podatak *ema*il predstavlja e-mail nastavnika.

Podaci tabele *T\_Predmet* su: *IdPredmeta*, *ImeNastavnika*, *PrezimeNastavnika* i *NazivPredmeta*. Podatak *IdPredmeta* je primarni ključ tabele *T\_Predmet* i ima vrednost *IDENTITY*, što znači da se vrednost *IdPredmeta* povećava za jedan za svaki novi uneti predmet. Podaci *ImeNastavnika, PrezimeNastavnika i NazivPredmeta* se unose sami. Tabela *T\_Predmet* je povezana sa tabelom *T\_Dnevnik* vezom jedan prema više. Podatak *ImeNastavnika* predstavlja ime nastavnika, podatak *PrezimeNastavnika* predstavlja prezime nastavnika, podatak *NazivPredmeta* predstavlja naziv predmeta.

Podaci tabele *T\_Dnevnik* su: *IdDnevnika*, *IdUcenika*, *IdPredmeta*, *IdGimnazije*, *IdNastavnika*, *Datum*, *Ocena*, *Razred, ImeUcenika, PrezimeUcenika,* i *NazivPredmeta*. Podatak *U IdDnevnika* je primarni ključ tabele *T\_Dnevnik* i ima vrednost *IDENTITY*, što znači da se vrednost *IdDnevnika* povećava za jedan za svaki novi uneti dnevnik. Ostali podaci se unose sami. Tabela *T\_Dnevnik*  je povezana sa tabelama *T\_Gimnazija, T\_Ucenik, T\_Nastavnik, T\_Predmet*  vezom jedan prema više. Podatak *Datum* predstavlja dan dobijene ocene, podatak *Ocena* predstavlja ocenu učenika koju je dobio iz predmeta predstavljenog podatkom *NazivPredmeta*. *Razred* predstavlja razred koji učenik pohađa. Podaci *ImeUcenika* i *PrezimeUcenika* predstavljanju ime učenika i prezime učenika.

# 2. Korišćene tehnologije

Korišćene tehnologije predstavljaju koje su sve tehnologije upotrebljene prilikom izrade *Windows Presentation Foundation* aplikacije. Postoji veliki broj tehnologija koji je potreban za izradu aplikacije.

## 2.1. Microsoft Visual Studio

*Microsoft Visual Studio* je integrisano razvojno okruženje iz *Microsofta*. Koristi se za razvoj kompjuterskih programa za *Microsoft* *Windows*, kao i veb sajtove, veb aplikacije i veb servise. *Visual* *Studio* koristi *Microsoft* platforme za razvoj softvera kao što su *Windows* *API*, *Windows* *Forms*, *Windows* *Presentation* *Foundation*, *Windows* *Store* i *Microsoft* *Silverlight*. Može da se piše i upravlja kodom.

*Visual* *Studio* uključuje editor koda koji podržava *IntelliSense*. Dolazi sa ugrađenim *C#* i *C++* , *VB* *NET* (*Visual* *Basic* *.NET*) i *F#* programskim jezicima. Takođe se može programirati i u drugim programskim jezicima kao što su *JavaScript*, *CSS*, *XML*, *HTML*, itd. instaliranjem jezičkih servisa za ovo razvojno okruženje. *Microsoft* *Visual* *Studio* je jedan od najpopularnijih razvojnih okruženja koji pruža velike mogućnosti u razvoju programa. Njegove mane su donekle ispravljene pomoću raznih ekstenzija razvijenih kako od „*Microsofta*“ tako i od strane drugih programera. Konkretno kod programiranja *Windows* *form* aplikacija, on sam generiše skelet aplikacije i sve podesi automatski kako bi korisnik mogao da se bavi problemom koji treba da reši, a ne pravljenjem prozora aplikacije. Pored desktop aplikacija takođe nudi mogućnost kreiranja i veb aplikacija. Microsoft je jako dobro razvio pronalaženje i uklanjanje grešaka pri programiranju zbog toga predstavlja jako popularnu i korisnu alatku programerima.

## 2.2. C# programski jezik

*C#* (*C Sharp*) programski jezik je jedan od mlađih programskih jezika nastao *2002.* godine kao sastavni deo *MS* *.NET Framework-a 1.0*. *C#* je objektno orijentisani programski jezik namenjen razvoju aplikacija na *.NET Framework* platformi. Pošto je objekno orijentisan programski jezik, prvenstveno je sagrađen od klasa koje čine skupove objekata koji imaju zajedničku strukturu i ponašanje. Takođe kao i mnogi objektno orijentisani programski jezici, sadrži sposobnost ekapsulacije, pisanje metoda, nasleđivanje između klasa i visoke performanse tokom kompajliranja. Iz razloga enkapsulacije postoji obrazac kojim atributima različitih klasa možemo spolјa pristupiti indirektno i to metodama *get* i *set*. *C#* je *case* *sensitive* programski jezik, što znači da je od velike važnosti da li kucamo malim ili velikim slovima.

## 2.3. WPF

WPF, tj. *Windows* *Presentation* *Foundation*, je najnoviji *Microsoftov* pristup *GUI* *framework*-u, koji se koristi sa *.NET framework*-om. *GUI* *framework* je skraćenica za *Graphical* *User* *Interface* (*grafički korisnički interfejs*). Windows ima *GUI* za rad sa računarom, a veb pretraživač ima *GUI* koji omogućava surfovanje internetom.

*GUI* *framework* omogućava da se kreira aplikacija sa širokim spektrom *GUI* elemenata, kao što su *labele*, *tekstboks-*ovi i drugi dobro poznati elementi. Bez *GUI* *framework*-a treba ručno da se nacrtaju ovi elementi i da obradi sve scenarije i interakcije korisnika kao što su tekst unos teksta, i pomeranje miša. Sve ovo predstavlja mnogo posla, pa zbog toga većina programera koristi *GUI* *framework* koji će obaviti osnovne radnje i omogućiti programerima da se fokusiraju na razvijanju aplikacija.

Postoji dosta *GUI* *framework*-a, ali za *.NET* programere, najzanimljivji su *Windows* *Forme* i *WPF*. *WPF* je noviji, ali *Microsoft* i dalje održava i podržava *Windows* *Forme*. Postoji nekoliko razlika između ova dva *framework*-a, ali njihova svrha je ista, da se kreira na najlakši našin sa odličnim *GUI*.

## 2.4. Microsoft SQL Server

*Microsoft* *SQL* *Server* je relacijska baza podataka kojoj je primarni jezik za upite *Transact* *SQL* (*T-SQL*), što znači da osim osnovnih i klasičnih (*SELECT* *tipa*) *SQL* upita dozvoljava i složenije stvari poput menjanja programskog toka (*IF* *naredba*) i slično. *Transact* *SQL* nastao je kao plod saradnje između *Microsofta* i *Sybasea*. *SQL* server je baza podataka koja se smestila na prag između manjih i srednjih baza. Prvi *SQL* *Server* za *Windows* *NT* izašao je isto kada i sami *Windows*-i.

*SQL* *Server* je prva baza podataka na svetu koja je posedovala korisnički interfejs. Takođe, *SQL* *Server* je prva komercijalna baza podataka koja je podržala *Intelovu* *64*-bitnu arhitekturu procesora. Od *SQL* *Servera* *2000* do *SQL* *Servera* *2005* napravljeni su golemi pomaci u poboljšanju same baze podataka. Najbitniji od njih su: Klijentski deo *IDE* alata, *SQL* *Server* *Integration* *Services*, *Reporting* *Server*, *OLAP* server, podrška za *data* *mining*, *Notification* *Services* itd.

*Microsoft* *SQL* podržava „*Open* *Database* *Connectivity*“, odnosno skraćeno *ODBC* tehnologiju. *SQL* *Server* *2005* poseduje i podršku za *Web* *servise*, tj. za *Simple* *Object* *Access* *Protocol*, *Service* *Oriented* *Architecture* *Protocol* odnosno skraćeno *SOAP* *W3C* standard. To je izvanredna stvar zato što omogućuje ne *Windows* klijentima da komuniciraju preko te platforme sa *SQL* *Serverom*. U verziji 2005 takođe je izdan i sertifikovan *JDBC* *API* za komunikaciju sa *Java* aplikacijama. *SQL* *Server* također poseduje i mogućnosti poput *mirroringa*, *clusteringa* i *snapshots*-ova. *Microsoft* se stvarno potrudio, te je omogućio brojne aplikacije koje spremaju podatke na *SQL* *Server*. *SQL* *Server* *2005* uključuje i *Common* *Language* *Runtime* odnosno *CLR*.

# 3. Postupak kreiranja aplikacije

U postupku kreiranja aplikacije prvo se kreira prozor *MainWindow*. Potrebno je da se kreira još 5 prozora koji su sledeći: *Gimnazija*, *Ucenik*, *Nastavnik*, *Predmet*  i *Dnevnik*. Pored toga se kreira i baza podataka.

## 3.1. Kreiranje baze podataka

Potrebno je da se kreira baza podataka sa nazivom *MUZICKASKOLA.* Nakon toga kreiraju se tabele sa njihovim atributima. U svakoj tabeli treba da se nalazi jedan od atributa koji će biti primarni ključ te tabele.

|  |
| --- |
| Create database MUZICKASKOLA;  go  use MUZICKASKOLA;  go  CREATE TABLE[dbo].[T\_Gimnazija] (  [IdGimnazije] [int] NOT NUll IDENTITY(1,1),  [IdUcenika] [int]NUll ,  [Naziv] [nvarchar] (30) NULL,  [Grad] [nvarchar] (30) NULL,  [Adresa] [nvarchar] (30) NULL,  [email] [nvarchar] (30) NULL  )  ALTER TABLE [dbo].[T\_Gimnazija] add  CONSTRAINT[PK\_T\_Gimnazija] PRIMARY KEY CLUSTERED  (  [IdGimnazije]  )  CREATE TABLE[dbo].[T\_Nastavnik] (  [IdNastavnika] [int] NOT NUll IDENTITY (1,1),  [ImeNastavnika] [nvarchar] (30) NULL,  [PrezimeNastavnika] [nvarchar] (30) NULL,  [email] [nvarchar] (30) NULL  )  ALTER TABLE [dbo].[T\_Nastavnik] add  CONSTRAINT[PK\_T\_Nastavnik] PRIMARY KEY CLUSTERED  (  [IdNastavnika]  )  CREATE TABLE[dbo].[T\_Ucenik] (  [IdUcenika] [int] NOT NUll IDENTITY (1,1),  [ImeUcenika] [nvarchar] (30) NULL,  [PrezimeUcenika] [nvarchar] (30) NULL,  [Instrument] [nvarchar] (30) NULL    )  ALTER TABLE [dbo].[T\_Ucenik] add  CONSTRAINT[PK\_T\_Ucenik] PRIMARY KEY CLUSTERED  (  [IdUcenika]  )  CREATE TABLE[dbo].[T\_Predmet] (  [IdPredmeta] [int] NOT NUll IDENTITY (1,1),  [NazivPredmeta] [nvarchar] (30) NULL,  [ImeNastavnika] [nvarchar] (30) NULL,  [PrezimeNastavnika] [nvarchar] (30) NULL    )  ALTER TABLE [dbo].[T\_Predmet] add  CONSTRAINT[PK\_T\_Predmet] PRIMARY KEY CLUSTERED  (  [IdPredmeta]  )  CREATE TABLE[dbo].[T\_Dnevnik] (  [IdDnevnika] [int] NOT NUll IDENTITY (1,1),  [IdUcenika] [int] NULL,  [IdPredmeta] [int] NULL,  [IdGimnazije] [int] NULL,  [IdNastavnika] [int] NULL,  [NazivPredmeta] [nvarchar] (30) NULL,  [ImeUcenika] [nvarchar] (30) NULL,  [PrezimeUcenika] [nvarchar] (30) NULL,  [Razred] [int] NULL,  [Ocena] [int] NULL,  [Datum] [date] NULL  )  ALTER TABLE [dbo].[T\_Dnevnik] add  CONSTRAINT[PK\_T\_Dnevnik] PRIMARY KEY CLUSTERED  (  [IdDnevnika]  )  ALTER TABLE[dbo]. [T\_Gimnazija]  ADD CONSTRAINT [FK\_GimnazijaUcenik]  FOREIGN KEY (IdUcenika) REFERENCES [dbo].[T\_Ucenik](IdUcenika)  ON UPDATE NO ACTION  ON DELETE NO ACTION  ALTER TABLE[dbo]. [T\_Dnevnik]  ADD CONSTRAINT [FK\_DnevnikUcenik]  FOREIGN KEY (  [IdUcenika]  )REFERENCES[dbo].[T\_Ucenik](  [IdUcenika]  )  ON UPDATE NO ACTION  ON DELETE NO ACTION  ALTER TABLE[dbo]. [T\_Dnevnik]  ADD CONSTRAINT [FK\_DnevnikPredmet]  FOREIGN KEY (  [IdPredmeta]  )REFERENCES[dbo].[T\_Predmet](  [IdPredmeta]  )  ON UPDATE NO ACTION  ON DELETE NO ACTION  ALTER TABLE[dbo]. [T\_Dnevnik]  ADD CONSTRAINT [FK\_DnevnikGimnazija]  FOREIGN KEY (  [IdGimnazije]    )REFERENCES[dbo].[T\_Gimnazija](  [IdGimnazije]  )  ON UPDATE NO ACTION  ON DELETE NO ACTION  ALTER TABLE[dbo]. [T\_Dnevnik]  ADD CONSTRAINT [FK\_DnevnikNastavnik]  FOREIGN KEY (  [IdNastavika]  )REFERENCES[dbo].[T\_Nastavnik](  [IdNastavnika]  )  ON UPDATE NO ACTION  ON DELETE NO ACTION |

*Listing 1. Baza podataka MUZICKASKOLA*

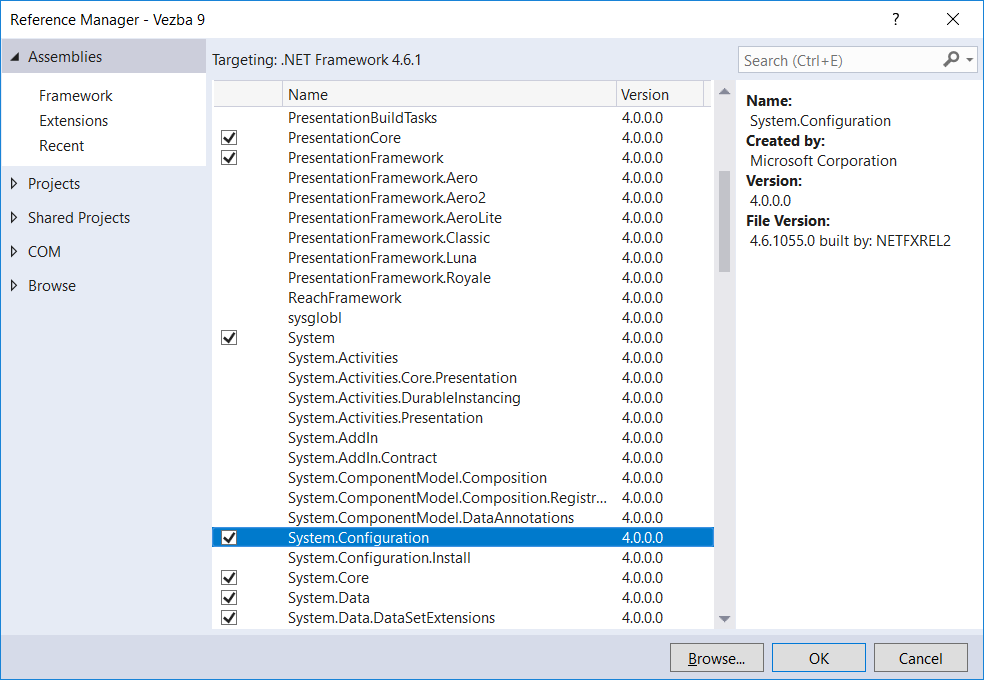
## 3.2. Kreiranje konekcionog stringa

Nakon kreiranja baze podataka, potrebno je da se kreira konekcioni string, koji omogućava povezivanje na bazu podataka. Potrebno je da se u *App.config* doda konekcioni string, a u njemu *Data Source* (*Listing 2.*)*.*

|  |
| --- |
| <connectionStrings>  <add name="connMUZICKASKOLA" connectionString="Data Source=WIN7-PC\SQLEXPRESS;Initial Catalog=MUZICKASKOLA;Integrated Security=True"/>  </connectionStrings> |

*Listing 2. Konekcioni string za povezivanje na bazu podataka*

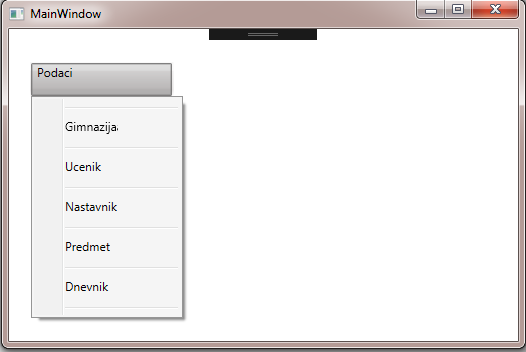
Kada se kreira konekcioni string potrebno je da se doda referenca da prozori mogu da rade sa konfiguracionom bazom u kojem je putanja do baze podataka. Referenca se dodaje klikom na desni klik na *References* – *Add Reference* i otvara se *Reference Manager* u kojem je potrebno da se čekira *System Configuration* (*Slika 1.*).



*Slika 1. Dodavanje reference System Configuration*

## 3.3. Prozor MainWindow

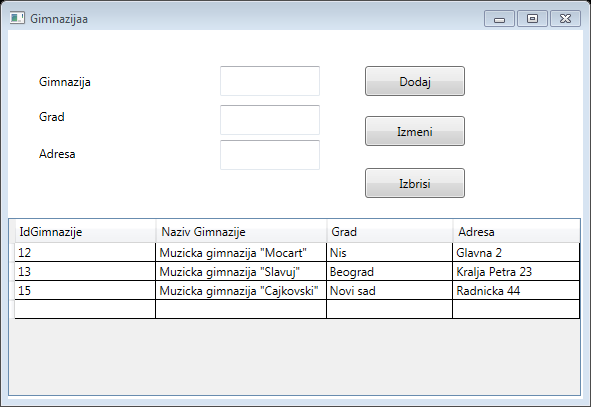
U glavnom prozoru *MainWindow* je ubačena *Menu* alatka sa poljem „*Podaci*“. Polje „*Podaci*“ sadrži 5 podpolja preko kojih se pristupa prozorima za unos podataka o *gimnazijma*, *učenicima*, *nastavnicima*, *predmetima* i *dnevnicima*.



*Slika 2. Izgled prozora MainWindow*

## 3.4. Prozor Gimnazijaa

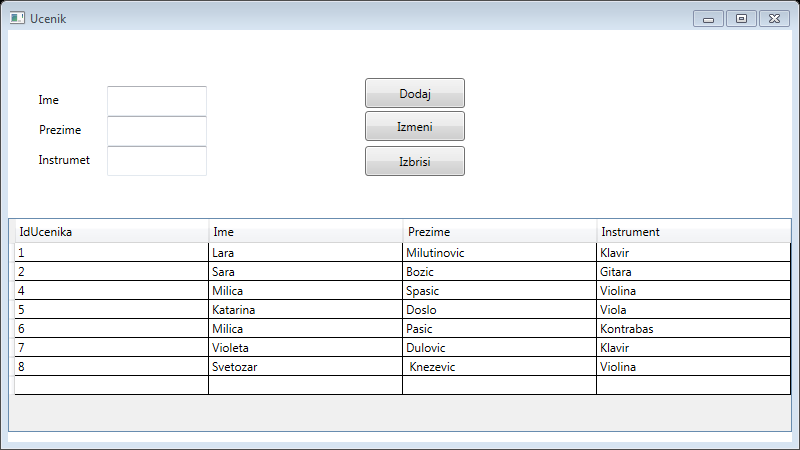
Karakteristike gimnazije su:  *Naziv,* *Grad*, *Adresa*. Sva polja su tipa *textBox* i u njih se vrši upis. U ovoj formi se nalazi i *DataGrid* kontrola *DataGridGimnazijaa*, koja omogućuje prikaz podataka za sve dodate gimnazije u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.



*Slika 3. Izgled prozora Gimnazijaa*

## 3.5. Prozor Ucenik

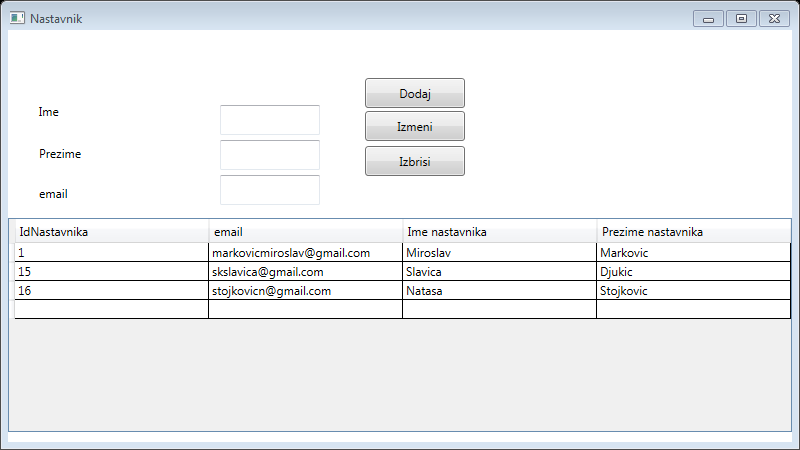
Karakteristike ucenika su njegovo: *Ime*, *Prezime* i *Instrument.* Sve tri karakteristike su polja tipa *textBox* i u njih se vrši upis. U ovoj formi se nalazi i *DataGrid* kontrola *DataGridUcenik*, koja omogućuje prikaz podataka za sve dodate učenike u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.



*Slika 4. Izgled prozora Ucenik*

## 3.6. Prozor Nastavnik

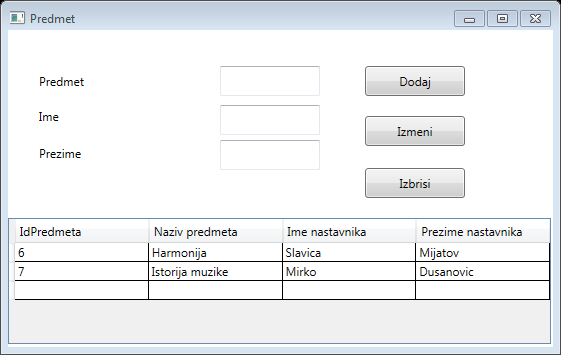
Karakteristike nastavnika su njegovo: *Ime*, *Prezime* i *email*. Sve karakteristike polja su tipa *textBox* i u njih se vrši upis. U ovoj formi se nalazi i *DataGrid* kontrola *DataGridNastavnik*, koja omogućuje prikaz podataka za sve dodate nastavnike u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.

**

*Slika 5. Izgled prozora Nastavnik*

## 3.7. Prozor Predmet

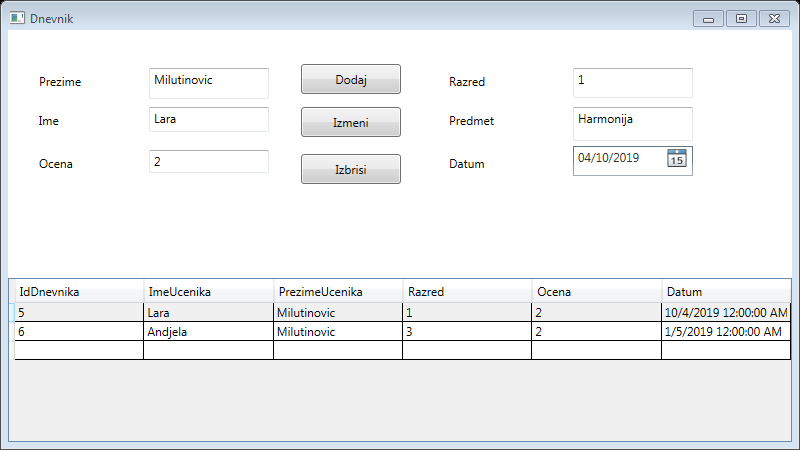
Karakteristike predmeta su njegov: *Naziv*, *Ime* i *Prezime* nastavnika. Sve tri karakteristike su tipa polja *textBox* i u njega se vrši upis. U ovom prozoru se nalazi i *DataGrid* kontrola *DataGridPredmet*, koja omogućuje prikaz podataka za sve dodate predmete u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.



*Slika 6. Izgled prozora Predmet*

## 3.8. Prozor Dnevnik

Karakteristike dnevnika su: *Ime*, *Prezime, Predmet*, *Datum*, *Razred i Ocena*. *Ime i prezime* kao i ostale karakteristike su polja tipa *textBox* i u njih se vrši upis. *U* ovojformi se nalazi i DataGrid kontrola *DataGridDnevik*, koja omogućuje prikaz podataka za sve dodate dnevnike u bazu podataka, dugmad za dodavanje, promenu i brisanje podataka iz baze podataka.



*Slika 7. Izgled prozora Dnevnik*

# 4. Ključni delovi koda

U glavnom prozoru *MainWindow* se nalazi meni preko kojih se pristupa prozorima. Potrebno je da se doda kod iza koji omogućava otvaranje prozora (*Listing 2.*)

|  |
| --- |
| public partial class MainWindow : Window  {  public MainWindow()  { /\*InitializeComponent();\*/}  private void MenuItemGimnazijaa\_Click(object sender, RoutedEventArgs e)  {  Gimnazijaa objGimnazijaa = new Gimnazijaa();  objGimnazijaa.Show();  }  private void MenuItemUcenik\_Click(object sender, RoutedEventArgs e)  {  Ucenik objUcenik = new Ucenik();  objUcenik.Show();  }    private void MenuItemPredmet\_Click(object sender, RoutedEventArgs e)  {  Predmet objPredmet = new Predmet();  objPredmet.Show();  }  private void MenuItemDnevnik\_Click(object sender, RoutedEventArgs e)  {  Dnevnik objDnevnik = new Dnevnik();  objDnevnik.Show();  }  private void MenuItemNastavnik\_Click(object sender, RoutedEventArgs e)  {  Nastavnik objNastavnik = new Nastavnik();  objNastavnik.Show();  }  } |

*Listing 2. Događaji u MainWindow-u za otvaranje svakog prozora*

Prilikom unosa, izmene ili brisanje podataka potrebno je da se poništi podaci koji su uneseni u odgovarajućim poljima. U kodu je prikazan primer iz prozora *Ucenik* (*Listing 3.*).

|  |
| --- |
| private void ponistiUnosTxt()  {  txtidDnevnika.Text = "";  txtImeUcenika.Text = "";  txtPrezimeUcenika.Text = "";  txtOcena.Text = "";  txtRazred.Text = "";  dtDatum.Text = "";  txtNazivPredmeta.Text = "";  } |

*Listing 3. Kod za poništavanje podataka u odgovarajućim poljima u prozoru Ucenik*

Da bi se uradila izmena podataka potrebno je da se zna koji su podaci selektovani. U kodu je prikazan primer iz prozora *Nastavnik* (*Listing 4.*).

|  |
| --- |
| private void DataGridPredmet\_SelectionChanged(object sender, SelectionChangedEventArgs e)  {  DataGrid dg = sender as DataGrid;  DataRowView dr = dg.SelectedItem as DataRowView;  if (dr != null)  {  txtidPredmeta.Text = dr["IdPredmeta"].ToString();  txtImeNastavnika.Text = dr["ImeNastavnika"].ToString();  txtPrezimeNastavnika.Text = dr["PrezimeNastavnika"].ToString();  txtNazivPredmeta.Text = dr["NazivPredmeta"].ToString();  }  } |

*Listing 4. Kod za selektovanje podataka prilikom izmene podataka u prozoru Nastavnik*

# 5. Zaključak

Mogućnosti za dalji razvoj aplikacije su na primer, da se na prozoru *Predmet* doda broj časova u toj godini. Na prozoru *Ucenik* bi se moglo omoguciti da se bira pol učenika pomoću korišćenja checkBox-a. Pored toga može da se doda prozor *Takmičenje i Diplome*. To su neke opcije koje postoje za unapređivanje ovog programa.

# LITERATURA

1. <https://sh.wikipedia.org/wiki/Microsoft_SQL_Server>
2. <http://en.wikipedia.org/wiki/Microsoft_Visual_Studio>
3. <https://en.wikipedia.org/wiki/C_Sharp_%28programming_language%29>
4. <https://wpf-tutorial.com/about-wpf/what-is-wpf/>