

**Univerzitet u Novom Sadu Fakultet tehničkih nauka Inženjerstvo informacionih sistema**

Uputstvo za razvoj WPF aplikacije

Novi Sad, 2018. god.

Sadržaj

[1. Uvod 3](#_Toc503231057)

[2. Zadatak 3](#_Toc503231058)

[3. Kreiranje UML dijagrama 4](#_Toc503231059)

[3.1 Dijagram slučajeva upotrebe 4](#_Toc503231060)

[3.2 Dijagram klasa 8](#_Toc503231061)

[4. Kreiranje baze podataka 9](#_Toc503231062)

[5. Kreiranje WPF aplikacije (desktop aplikacije) 23](#_Toc503231063)

[6. Testiranje aplikacije 73](#_Toc503231064)

# Uvod

U okviru ovog uputstva opisani su svi potrebni koraci koje studenti treba da prođu kako bi razvili WPF (engl. Windows Presentation Foundation) aplikaciju i upoznali se sa osnovnim principima UML modelovanja, kao i rada u Microsoft SQL Server-u i Microsoft Visual Studiju.

# Zadatak

Cilj zadatka je da se kreira WPF aplikacija koja simulira rad apoteke. Njena komunikacija sa bazom podataka ostvarivaće se putem ADO.NET tehnologije. Aplikacija treba da pruži mogućnost prikazivanja podataka o uradjenim ispitima, instrumentima, zaposlenim profesorima i ispitanicima, ucenicima koji su radili ispit, samom ispitu i na kraju, pitanjima koja su moguca da se pojave na ispitu.

Takođe, treba da pruža mogućnost dodavanja, uređivanja i brisanja podataka o navedenim entitetima.

# Kreiranje UML dijagrama

## Dijagram slučajeva upotrebe

Na samom početku pristupamo inicijalnom modelovanju sistema odnosno kreiranju dijagrama slučajeva upotrebe na osnovu datog opisa za primjer muzicke skole.

**Slučaj upotrebe:** Evidencija djaka

**Kratak opis:** Evidencija djaka u bazu podataka

**Učesnici:** Ucenik(U), Administrator(A)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ucenik(U) je dosao za vreme rezervisano za prijemni ispit, sa svim potrebnim dokumentima

**Opis:** Ucenik(U) daje podatke Administratoru(A), ime, prezime, JMBG, datum rodjenja. Administrator(A) ubacuje podatke u bazu podataka.

**Izuzeci:**

[Upisani su pogresni podaci] - *Ukoliko su u bazu podataka upisani pogresni podaci, Administrator(A) preko opcije "uredi" menja iste.*

[Ucenik(U) je vec u bazi podataka] - *Ukoliko je Ucenik(U) vec izlazio na prijemni ispit, u bazu podataka se upisuje samo datum ponovnog izlazenja (u odgovarajucu kolonu).*

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Podaci moraju biti uneseni u bazu podataka.

**Slučaj upotrebe:** Izdavanje djacke knjizice

**Kratak opis:** Izdavanje djacke knjizice Uceniku

**Učesnici:** Ucenik(U), Administrator(A)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ucenik je izabrao instrument i dobrio profesora.

**Opis:** Administrator(A) ispisuje novu djacku knjizicu. Unosi Ucenikove podatke i daje mu knjizicu. Unosi dodatne podatke o Uceniku u skolsku bazu podataka (razred, polugodiste, datum upisa). Nakon ovoga, Ucenik je upisan u muzicku skolu.

**Izuzeci:**

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Podaci moraju biti uneseni u bazu podataka. Knjizica mora biti ispisana.

**Slučaj upotrebe:** Odabri instrumenata

**Kratak opis:** Odabri instrumenata

**Učesnici:** Ucenik(U), Administrator(A)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ucenik(U) je dobio potvrdu o polozenom ispitu.

**Opis:** Administrator(A) predstavlja instrumente Uceniku(U). Ucenik(U) bira na koji ce instrument da se upise. Administrator(A) na osnovu Ucenikovog izbora upisuje profesora za taj instrument u bazu podataka, u odgovarajucu kolonu.

**Izuzeci:**

[Nema slobodnih profesora] - Ukoliko nema slobodnih profesora za taj instrument, Administrator ponovo pita Ucenika da izabere instrument.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Uceniku mora biti dodeljen instrument i profesor.

**Slučaj upotrebe:** Izdavanje potvrde o ispitu

**Kratak opis:** Izdavanje potvrde o ispitu

**Učesnici:** Ucenik(U), Ispitanik(I)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ucenik(U) zavrsio prethodna dva dela ispita.

**Opis:** Ispitanik(I) ispisuje potvrdu o polozenom ili nepolozenom ispitu. Ukoliko je Ucenik(U) ostvario bar minimalni broj bodova na prvom delu ispita i pokazao dovoljnu sposobnost na drugom, dobija potvrdu o polozenom ispitu. U suprotnom, ako jedan od dva dela nije uspesno zavrsen, Ispitanik(I) izdaje potvrdu o nepolozenom ispitu.

**Izuzeci:**

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Ucenik mora dobiti potvrdu.

**Slučaj upotrebe:** Prijemni ispit

**Kratak opis:** Prijemni ispit

**Učesnici:** Ucenik(U), Ispitanik(I)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ucenik(U) je upisan u bazu podataka od strane Administratora(A), i poslat je u ucionicu gde se obavlja ispit.

**Opis:** Ispitanik(I) daje Uceniku(U) set pitanja koje on mora da resi u odredjenom vremenskom periodu.

**Izuzeci:**

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Ucenik je predao svoje odgovore.

**Slučaj upotrebe:** Provera muzickih sposobnosti

**Kratak opis:** Proveravanje muzickih sposobnosti Ucenika(U)

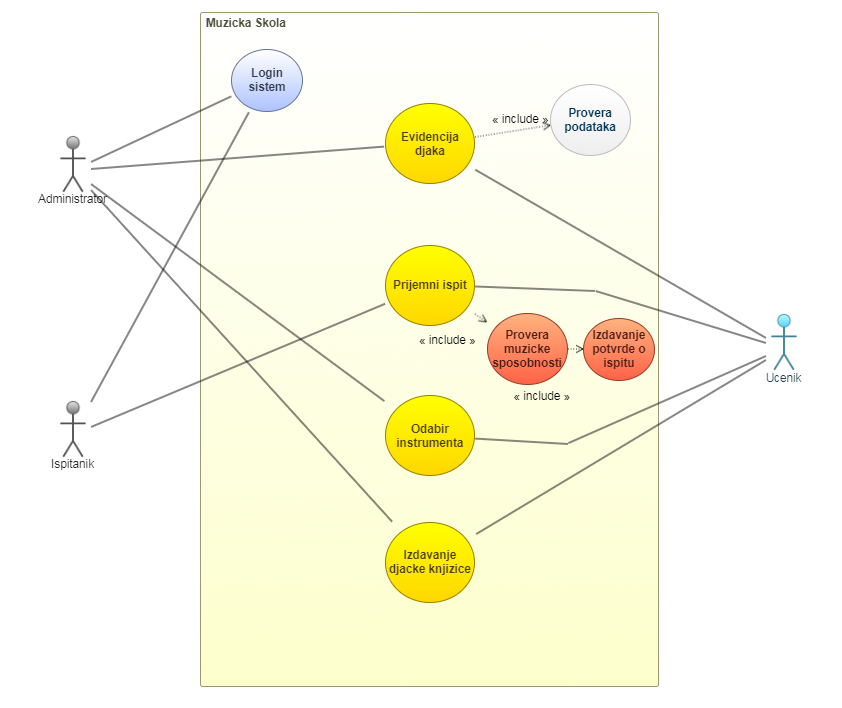
**Učesnici:** Ucenik(U), Ispitanik(I)

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ucenik(U) je zavrsio prvi deo ispita.

**Opis:** Ispitanik(I) zadaje vise zadataka (pevanje, slusanje) Uceniku(U).

**Izuzeci:**

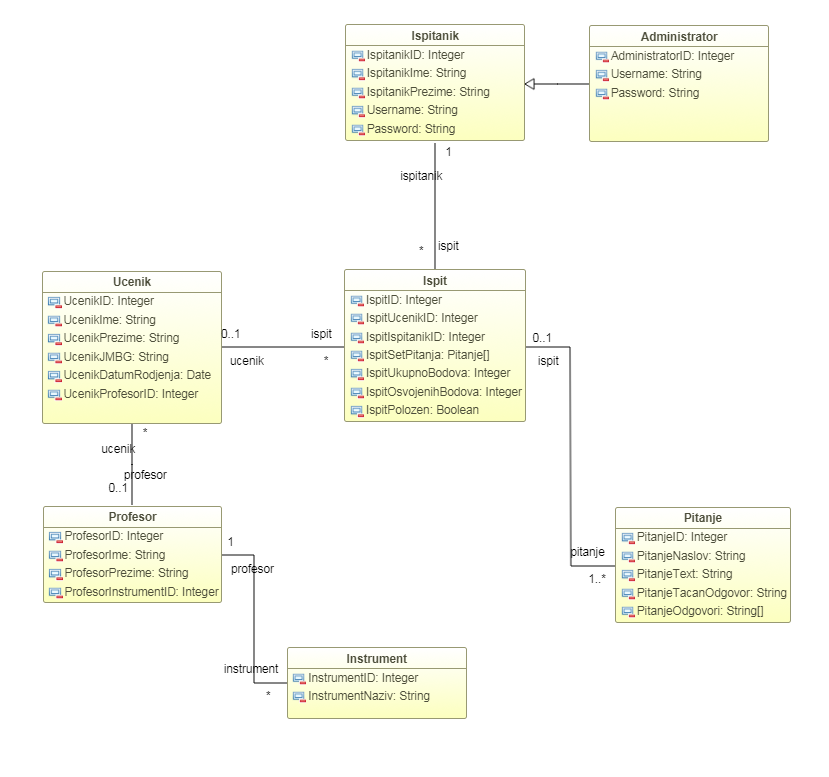
**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Ispitanik je zavrsio sa proverom Ucenikovih muzickih sposobnosti.



Slika 1 - Dijagram slučajeva upotrebe

## Dijagram klasa

Nakon uspesno kreiranog dijagrama slučajeva upotrebe, pristupamo izradi dijagrama klasa koji predstavlja vizuelni prikaz modela staticke strukture sistema. U slucaju primera muzicke skole, kreiran je dijagram na *Slici 2.*



Slika 2 - Dijagram klasa

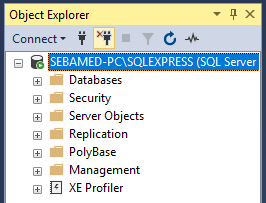
# Kreiranje baze podataka

Sledeci korak u izradi aplikacije jeste kreiranje baze podataka. Na samom pocetku, potrebno je pokrenuti **SQL Server Management Studio Express**. Kada se to uradi, pojavice se prozor sa dijalogom, kao sto je prikazano na *Slici 3*. U prikazanom dijalogu, potrebno je podesiti parameter **Server type** i **Authentication** , kao sto je prikazano na *Slici 3*. Napomena: Server name nije potrebno mijenjati, ono predstavlja naziv mašine na kojoj se radi.



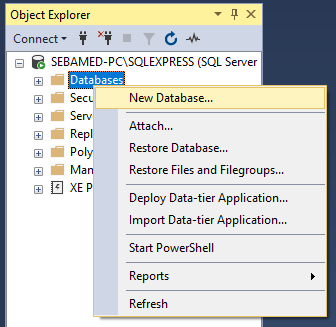
Slika 3 - Ostvarivanje konekcije na server

Nakon podesavanja parametara, klikom na dugme *Connect*, dijalog će se zatvoriti, a otvoriće se novi koji treba da izgleda kao što je prikazano na *Slici 4*.



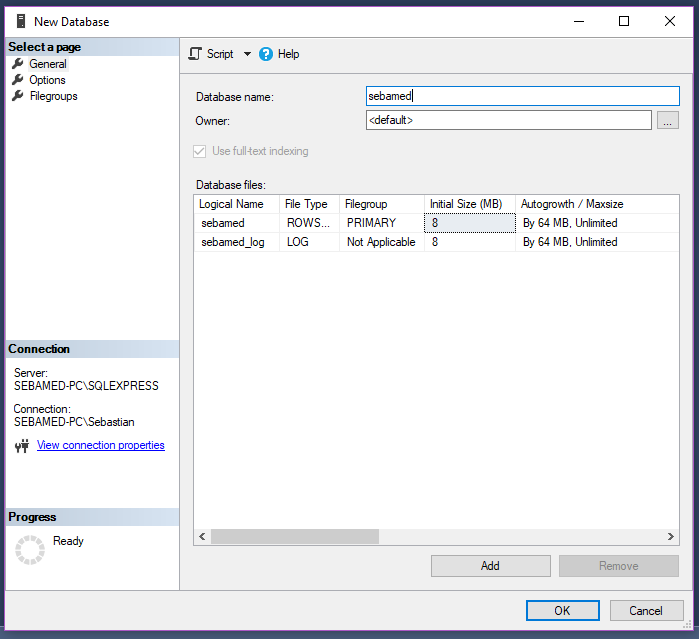
Slika 4 – Izgled Object Explorera nakon ostvarivanja konekcije na bazu

Sada je potrebno kreirati bazu podataka za aplikaciju. Baza se kreira tako što se klikne desnim klikom na **Database** u *treeview*-u koji se nalazi sa lijeve strane prozora u okviru *Object Explorer*-a. Otvoriće se lista u kojoj je potrebno selektovati **New Database** (Slika 5.).



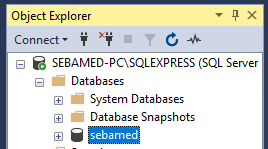
Slika 5 – Kreiranje baze podataka

Otvoriće se prozor u kome će se kreirati baza podataka. Potrebno je dati naziv bazi, u ovo slučaju naziv će biti PoIS\_Apoteka. Ostale parameter nije potrebno mijenjati. Kliknuti **OK**. U ovom slucaju ime baze je “sebamed” umesto “MuzickaSkola” zbog prebacivanja baze i podataka sa Ubuntu masine. (*Slika 6.)*



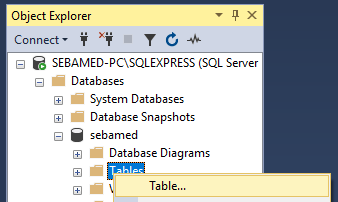
Slika 6 – Podešavanje parametara pri kreiranju baze podataka

Posle uspešnog kreiranja baze podataka u Object Explorer-u, u folderu Database se pojavljuje kreirana baza podataka **sebamed** (*Slika 7.).*



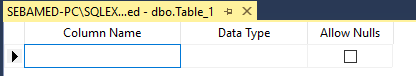
Slika 7 - Prikaz kreirane baze podataka

Proširivanjem baze **sebamed** u stablu *Object Explorer*-a, pojavljuje se vise datoteka. Desnim klikom kliknuti na datoteku *Tables* i odabrati *Table* (*Slika 7*). Nakon klika na ovu opciju, sa desne strane otvara se prozor za definisanje tabele, pri čemu će tabela imati predefinisani naziv *Table\_1. (Slika 8.)*



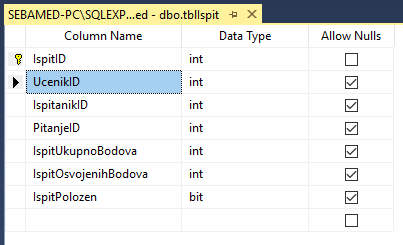
Slika 8 - Kreiranje tabele unutar baze podataka

Sledeće što je potrebno uraditi jeste kreirati kolone tabele tako što će se popuniti polja **Column Name, Data Type, Allow Nulls**. U kolonu *Column Name* se upisuje naziv atributa. U koloni *Data Type* se bira tip podatka za željeni atribut, dok se u koloni *Allow Nulls* selektuje da li određeno polje može imati *null* vrednost. Preporučuje se da *Allow Nulls* bude check-irano za sve kolone osim za primarni ključ kako bi se izbjegli određeni problem prilikom upisa rekorda u tabelu (*Slika 9*).



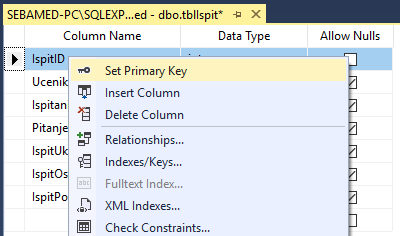
Slika 9 – Dodavanja obelezja unutar tabele

Svaka tabela neke baze mora da posjeduje primarni ključ – ID i bilo koja kolona može da se deklariše kao primarni ključ. U ovom slučaju, kolona *IspitID* treba da se deklariše kao primarni ključ. Kao takva, ova kolona će zbog svoje jedinstvene vrijednosti imati tip podatka int i neće imati *null* vrijednost. (*Slika 10.*)



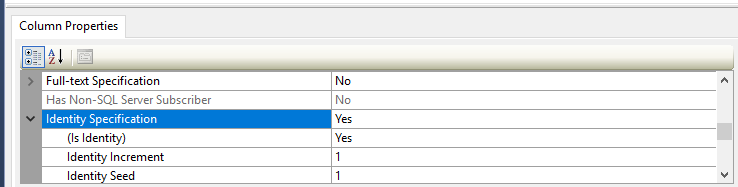
Slika 10 – Prikaz kreiranih obelezja

Da bi se *IspitID* deklarisala kao primarni ključ, potrebno je kliknuti desnim klikom na crnu strelicu koja se nalazi pored naziva ove kolone. Otvoriće se lista u kojoj treba odabrati opciju *Set Primary Key (Slika 11.)*. Ovom komandom se kolona *IspitID* deklariše kao primarni ključ.



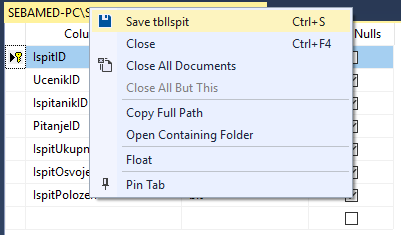
Slika 11 - Definisanje primarnog ključa

Da bi SUBP automatski kreirao primarni ključ, potrebno je u *Column Properties* pronaći opciju *Identity Specification*. Proširivanjem ove opcije *Identity Specification* dobija se opcija *Is Identity*, koju treba promjeniti na *Yes*. Ovdje je, takođe, moguće podesiti parameter *Identity Increment* i *Identity Seed.* *Identity Increment*-om se definiše za koliko raste sledeća vrijednost ključa, *Identity Seed*-om od koje vrijednosti počinju da se memorišu ključevi (Slika 12.).



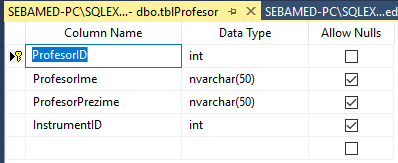
Slika 12 – Podešavanje automatske inkrementacije vrijednosti obelezja primarnog ključa

Čuvanje tabele se vrši pritiskanjem tastera **CTRL** i **S** (CTRL + S). Posto je ova baza podataka preuzeta sa Ubuntu masine, vec ima sacuvane sve tabele pa ih nije potrebno cuvati kao prvi put. Tabelu cuvamo sa prefiksom *tbl*, a citav naziv ce biti *tblIspit* (*Slika 13.*)

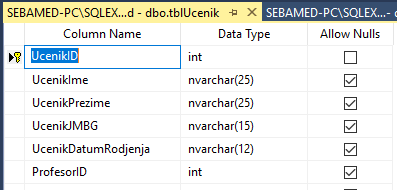


Slika 13 - Čuvanje kreirane tabele

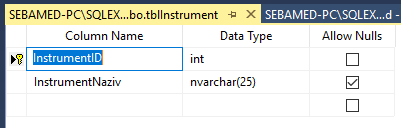
Isti proces bi se trebao ponoviti i za kreiranje tabela *tblProfesor, tblUcenik, tblInstrument, tblIspitanik, tblIspit i tblPitanje(Slike 14-21)*



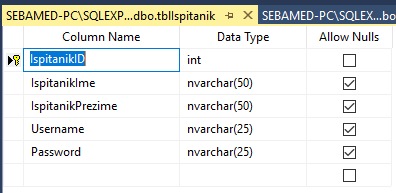
Slika 14 – Kreiranje tabele Profesor



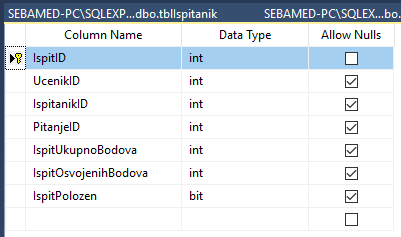
Slika 15 - Kreiranje tabele Ucenik



Slika 16 - Kreiranje tabele Instrument



Slika 17 – Kreiranje tabele Ispitanik

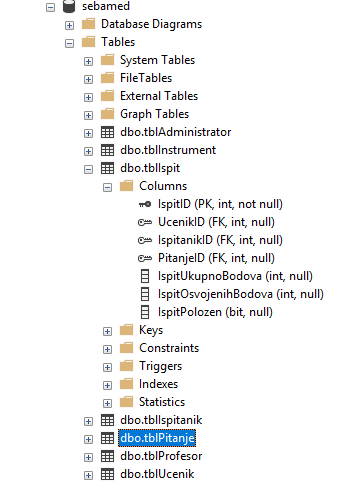


Slika 18 - Kreiranje tabele Ispit



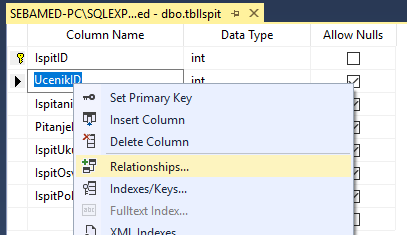
Slika 19 - Kreiranje tabele Pitanje

Nakon toga bi u *treeview*-u koji se nalazi sa leve strane prozora u okviru *Object Explorer*-a trebala da se nalazi baza podataka *sebamed* u kojoj se pored već kreirane *tblIspit* nalaze i novokreirane tabele sa odgovarajućim atributima (*Slika 22*.).



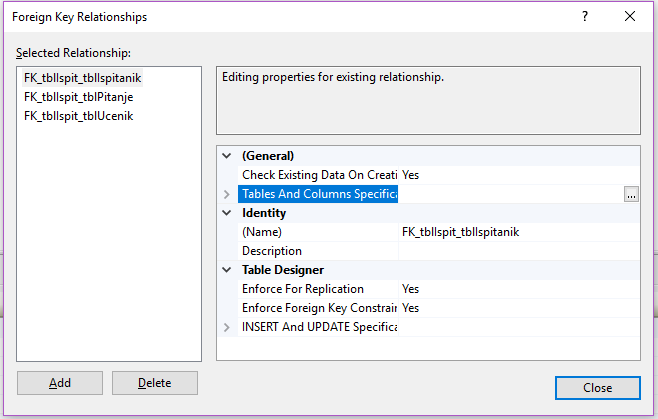
Slika 22 - Object Explorer prikaz kreiranih tabela

Kao što možete videti na slici 22, u *tblIspit* se nalaze tri *Foreign key-a (UcenikID, IspitanikID, PitanjeID)*. Prilikom unosa atributa u tabelu tblIspit potrebno je unijeti i attribute koji imaju nazive *UcenikID, IspitanikID, PitanjeID*, a vrijednost *DataType-a* za njih staviti na **int**. Zatim je potrebno desnim klikom kliknuti na crnu strelicu koja se nalazi pored naziva ove kolone. Otvoriće se lista u kojoj treba odabrati opciju *Relationships (Slika 23.)*



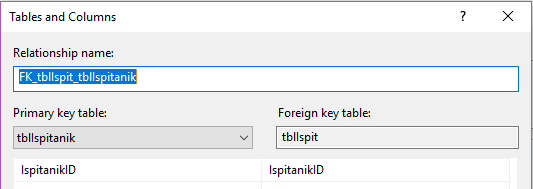
Slika 23 – Dodavanje veza

Otvara se prozor *Foreign key Relationships* (*Slika 24.)* u kojem je potrebno prvo kliknuti na dugme *Add* a zatim proširiti polje *Tables and Columns Specification* i kliknuti na tri tačke koje se nalaze desno.



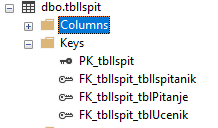
Slika 24 - Dodavanje stranog ključa

U otvorenom prozoru *Tables and Columns* potrebno je u okviru *Foreign key table,* konkretno iz tabele *tblIspit,* iz padajuće liste izabrati atribut koji smo mi definisali a koji želimo da povežemo sa drugom tabelom. Zatim u okviru *Primary key table* biramo tabelu sa kojom vršimo povezivanje i iz padajuće liste biramo ključ te tabele sa kojim ćemo povezati željeni atribut. Nakon što smo sve ovo odradili, prozor treba da nam izgleda kao na sledećoj slici (*Slika 25*).



Slika 25 - Kreiranje veza između tabela

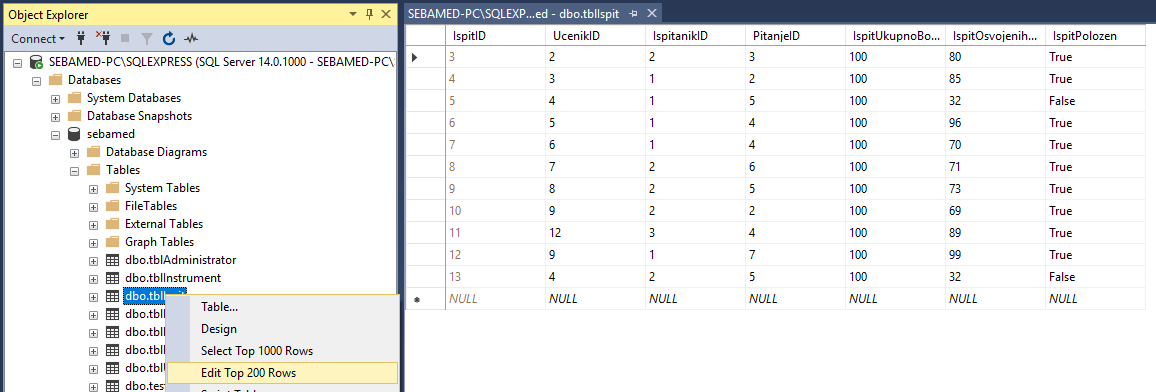
Dodavanjem *relationships*-a atributima *UcenikID, ProfesorID i PitanjeID*  u tabeli *tblIspit* oni postaju strani ključevi. Da li smo uspešno odradili povezivanje tabela znaćemo tako štoćemo proveriti da li se oni nalaze u *treeview*-u u okviru tabele *tblIspit* pod kategorijom *Keys .*(*Slika* *26.*)



Slika 26 - TreeView prikaz kreiranih ključeva

Isti postputak potrebno je ponoviti za sve tabele koje u sebi sadrže strane ključeve koji su primarni ključevi u drugim tabelama tblProfesor (*InstrumentID*) i tblUcenik(*ProfesorID)*

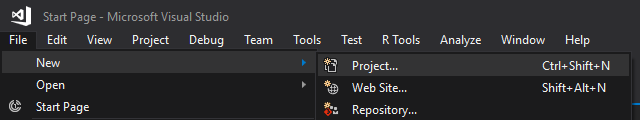
Sada su tabele kreirane i povezane, a dodavanje novih podataka u ovu bazu se može izvršiti desnim klikom na naziv tabele, gde se iz liste opcija treba odabrati **Edit Top 200 Rows** *(Slika 27*.).



Slika 27 - Prikaz kreirane tabele i dodavanje podataka

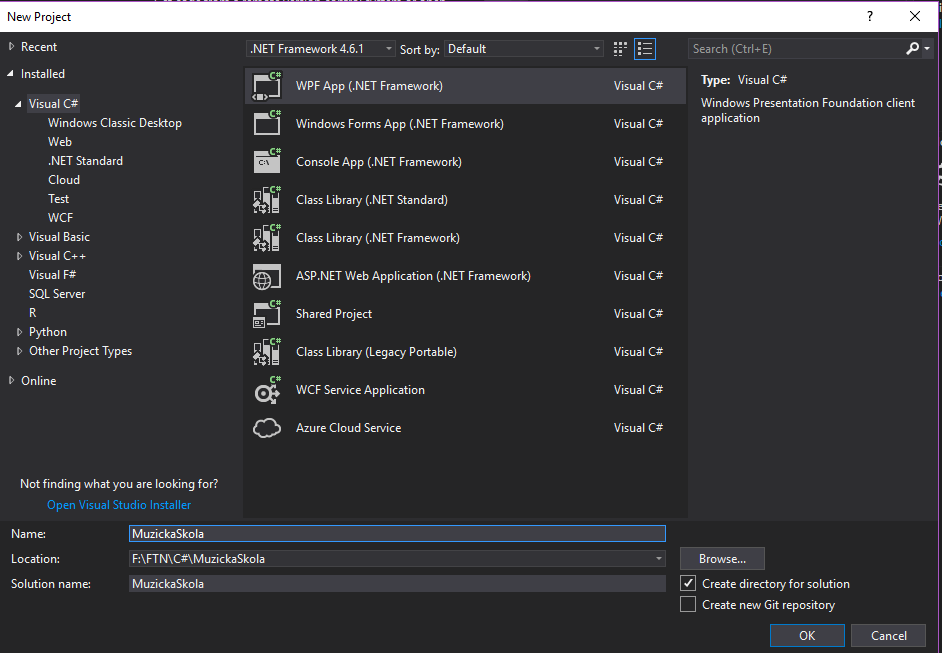
# Kreiranje WPF aplikacije (desktop aplikacije)

U okviru ovog poglavlja objasniće se kako se kreira **WPF** aplikacija. Nakon pokretanja Visual Studija potrebno je kliknuti na *File* menu a zatim odabrati *New Project. (Slika 28.)*



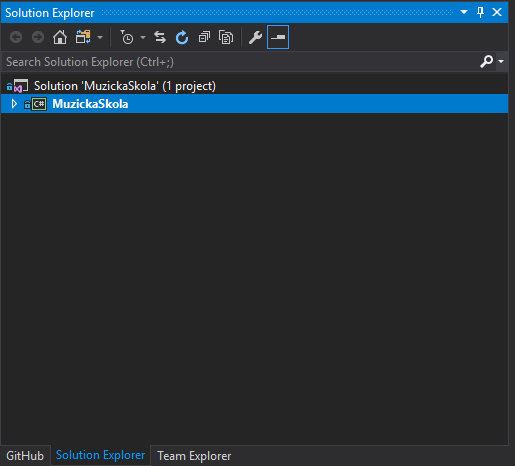
Slika 28 - Kreiranje novog projekta u Visual Studiju

Odabrati *Windows Classic Desktop*, zatim *WPF Application*. Navesti ime aplikacije i kliknuti **OK** (*Slika 29.*).



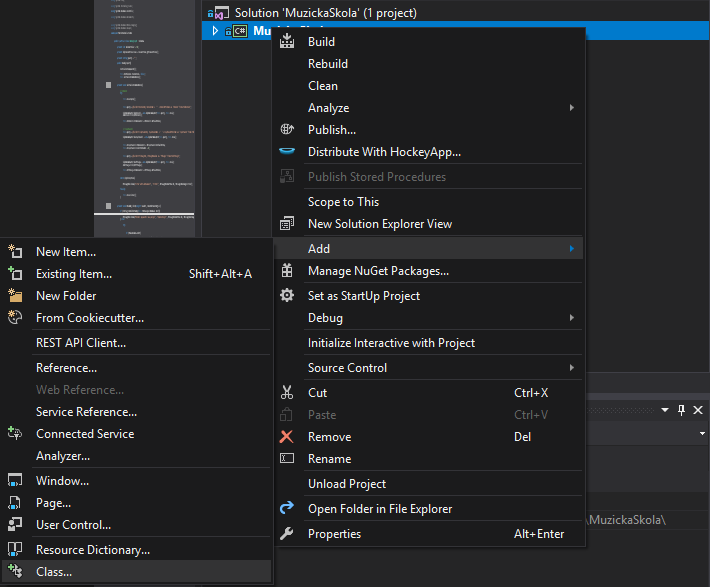
Slika 29 - Kreiranje WPF aplikacije

U *Solution Explorer-u* pojaviće se kreirana aplikacija. *(Slika 29.1.)*



Slika 29.1 - Solution Explorer kreirane WPF aplikacije

Sledeći korak je kreiranje konekcije u aplikaciji ka bazi podataka. Za početak, kreira se nova klasa u kojoj će se nalaziti statička metoda putem koje će se ostvarivati veza na prethodno kreiranu bazu. Klasa se kreira desnim klikom na naziv projekta (*MuzickaSkola*), a zatim *Add/Class*… *(Slika 30.)*



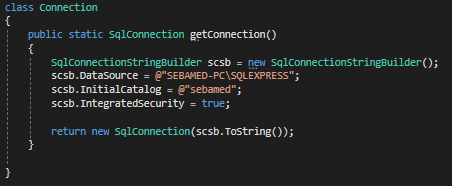
Slika 30 - Kreiranje klase

Novokreiranoj klasi se dodjeljuje naziv “*Connection”* i njen sadržaj prikazan je na Slici 31. Na samom početku uključujemo imenski prostor *System.Data.SqlClient*. Telo klase sadrži samo jednu statičku metodu pod nazivom “*getConnection*”, koja će se u kasnijem razvoju aplikacije pozivati po potrebi odnosno svaki put kada budemo zeleli da ostvarimo vezu sa bazom podataka, s’toga je povratini tip ove metode instanca klase *SqlConnection*. Unutar same metode prikazana su podešavanja parametara konekcije i to:

1) DataSource – Naziv servera na kojem je baza podataka smeštena.

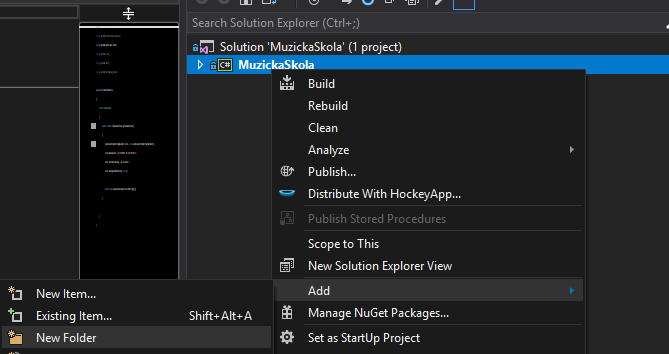
1. InitialCatalog – Naziv baze podataka kojoj želimo da pristupimo.
2. IntegratedSecurity – Ukoliko se baza nalazi na lokalnoj mašini ovaj parametar postavljamo na true.

Nakon podešavanja ovih parametara, ceo string prosleđujemo instanci klase *SqlConnection*.



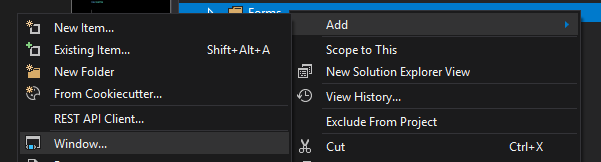
Slika 31 - Klasa konekcija

Sledeći korak u izradi aplikacije biće kreiranje formi za dodavanje objekata u bazu podataka. Kako bi sve forme bile uredno smještene na jednom mestu, potrebno je kreirati folder pod nazivom *Forme* u kom će se nalaziti svih 6 formi za, prethodno kreiranih, 6 tabela u bazi podataka. Kreiranje foldera se vrši na sličan način kao kreiranje klase, desnim klikom na projekat, a zatim *Add/ New Folder*. (*Slika 32*)



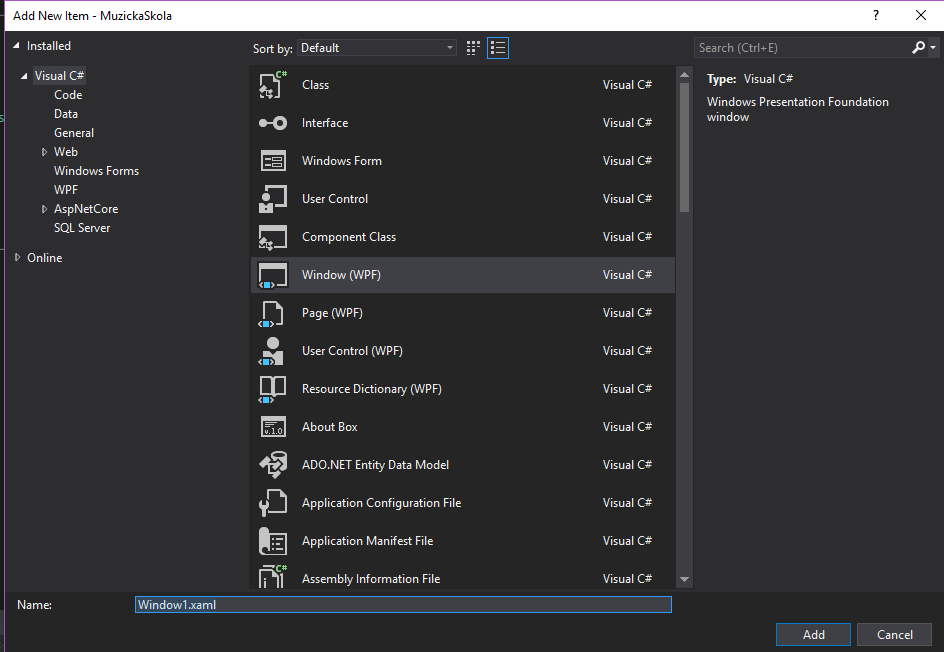
Slika 32 - Kreiranje foldera

Nakon kreiranja novog foldera, potrebno je dodati novi prozor za svaku od formi. WPF prozor, unutar foldera, kreira se desnim klikom na folder, a zatim *Add/Window*. (*Slika 33*.)



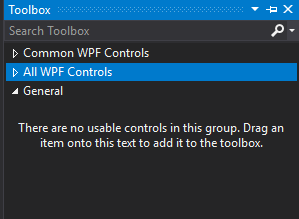
Slika 33 - Dodavanje formi unutar novog foldera

Nakon toga prikazaće se prozor da dodavanje novog *Item*-a unutar kog je potrebno podesiti naziv i zatim kliknuti na *Add (Slika 34.)*.



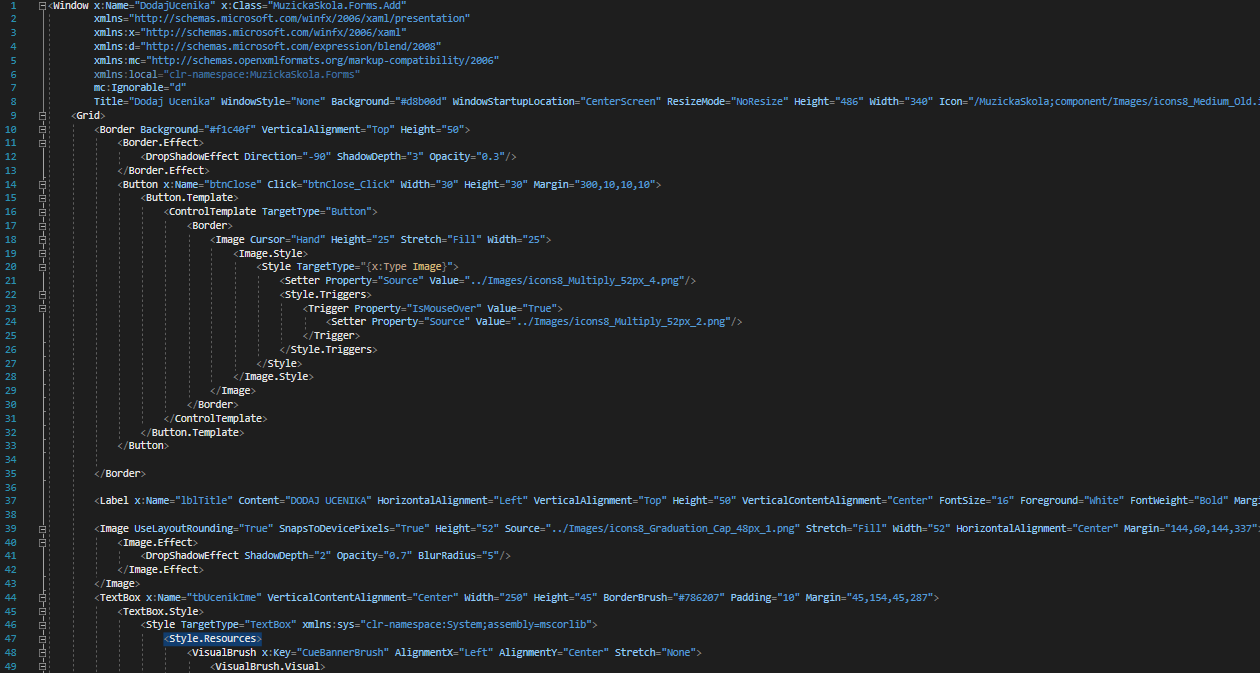
Slika 34 - Dodavanje novog prozora

Kada je novi prozor izgenerisan, potrebno je, unutar njega, dodati određene elemente putem kojih će se ostvarivati potrebne funkcionalnosti. Na levoj strani prozora, klikom na *Toolbox*, prikazaće se lista svih ugrađenih WPF komandi(*Slika 35*.). Jednostavnimprevlačenjemtih komandi na centralni prozor, izgenerisaće se kod u XAML deklarativnom jeziku unutar kojeg će se vršiti detaljna podešavanja svih komandi unutar prozora (*Slika 36.)*



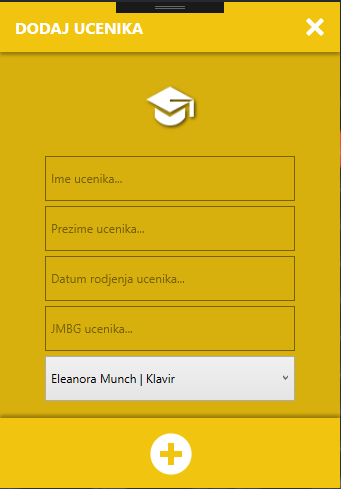
Slika 36 - Dodavanje komandi iz Toolbox-a

Forma za dodavanje novog Ucenika u bazu podataka, treba da sadrži sledeće elemente (*Slika* *36.).* Cetiriosnovna *TextBox*-ova unutar kojih će biti upisane vrednosti obeležja i jedan ComboBox koji ce prikazivati listu profesora. Takođe, uz svaki *TextBox*, mora postojati naziv (*lbl*) koji će opisivati njegovu namkenu, a uz ComboBox (*cb*). Na kraju, svaka formamorati imati ugrađeno dugme koje simulira text *DODAJ*, u ovom slucaju plus ikonica, putem kojih će se crednosti upisivati u bazu, i dugme izlaz iz forme (*X*)- koje korisniku omogućava zatvaranje forme (*Slika 37.*). Ove funkcionalnosti dugmića postižu se klikom miša na odgovarajuće dugme čime se u *Code Behind*-u generišu metode *btnAdd\_Click* i *btnClose\_Click*.



Slika 37 - XAML kod kreiranih komponenti (XAML kod je prevelik da bi stao u jednu sliku, pa ga mozemo pronaci celog na [ovom linku](https://pastebin.com/GKusL8Fj).

Vizuelni prikaz forme za dodavanje novog Ucenika izgledaće kao na *Slici 38.*



Slika 38 - Vizuelni prikaz forme za dodavanje novog Ucenika

U *Code Behind*-u ove forme, trebalo bi da se nalazi sledeća poslovna logika (*Slika 39*.). Najprije je potrebno importovati već poznati imenski prostor *System.Data.SqlClient*. Zatim, potrebno je napraviti novi objekat *SqlConnection* klase, putem kojeg će se ostvarivati veza ka bazi podataka nakon klika na dugme sa ikonicom *+*. Takodje smo dodali i promenljivu “query” tipa string, koja ce se koristiti u kodu kao globalna promenljiva za pisanje upita.



Slika 39 - Sadržaj pocetnog dela Code Behind-a forme Dodaj Ucenika. (Ceo kod na [ovom linku](https://pastebin.com/wxhxgeC7))

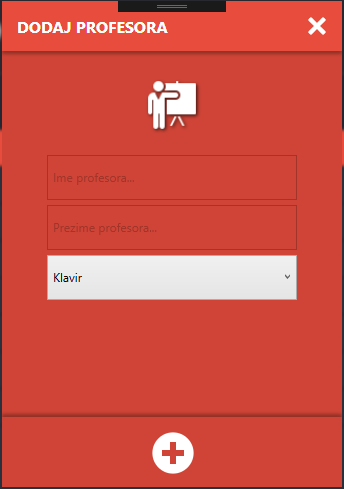
Ovakav prikaz pozadinskog koda gotovo je identičan za svaku od 6 kreiranih formi. Ono u čemu je svaka od ovih formi jedinstvena jeste pozadinska funkcionalnost koja se dešava u trenutku kada korisnik klikne na dugme za cuvanje. Na [ovom linku](https://pastebin.com/wxhxgeC7), u metodi *btnAdd\_click* prikazan je sadržaj izgenerisane metode *btnAdd \_Click* . Na samom početku, potrebno je otvoriti konekciju ka bazi podataka. Zatim, kreira se string koji u sebi sadrži *insert into* naredbu putem koje se dodaje nove objekat u bazu podataka. U *values* ove naredbe prosledjuju se pokupljene unete vrijednosti iz *TextBox*-ova ove forme. Potrebno je voditi računa da se vrednosti stringova ograde apostrofima. Nakon uspesno napravljene *insert* komande, ovaj string se prosleđuje objektu klase *SqlCommand*. Ovaj objekat, kao parametre prima prethodno kreirani string ali i konekciju putem koje će znati u koju bazu podataka treba da upiše prosleđeni zapis. Nakon toga, komanda se izvršava putem metode *ExecuteNonQuery()*, nakon čega se prozor zatvara. Ovaj deo koda potrebno je ograditi *try* blokom iz razloga što postoji mogućnost nastanka greške usljed prosljeđivanja unosa pogrešnog tipa podatka od strane korisnika. Kako prilikom ovakvog unosa, program ne bi “pukao”, potrebno je dodati catch blok unutar kojeg će se “hvatati” izuzetak tipa *SqlException*. Na kraju, u finally bloku, potrebno je postarati se da se prethodno otvorena konekcija sigurno zatvori.

Ukoliko konekcija ostane otvorena, niko drugi neće moći da pristupi bazi podataka na isti način dok se ova instanca konekcije ne zatvori.

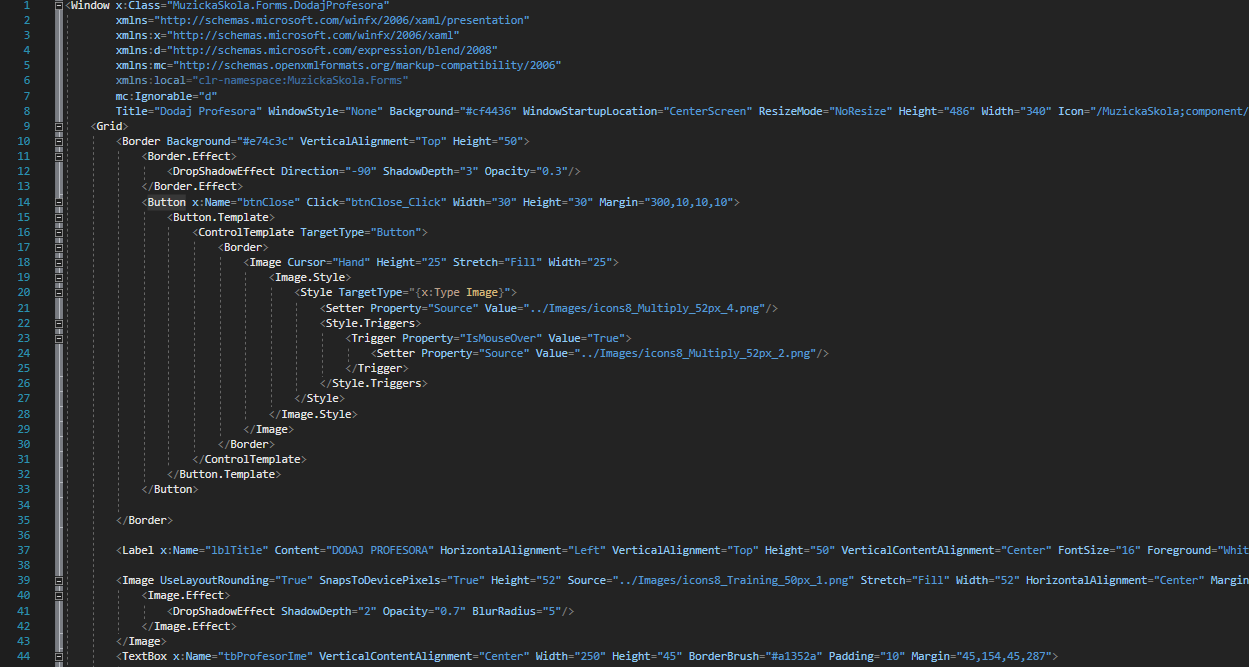
U nastavku će biti prikazan izgled i kod ostalih 5 formi.

Svaki od ovih formi stavlja blur efekat na pozadinski prozor (*MainWindow.xaml*) kako bi se postigao efekat fokusa.

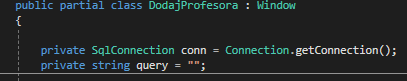
Forma za dodavanje novog Profesora ima veoma sličan vizuelni izgled kao i forma za dodavanje novog Ucenika. (*Slika 41.*)



Slika 41 – Dizajn forme za dodavanje novog profesora



Slika 42 – Deo XAML kod-a za dodavanje novog profesora ([ceo kod](https://pastebin.com/g9VMd7wB))

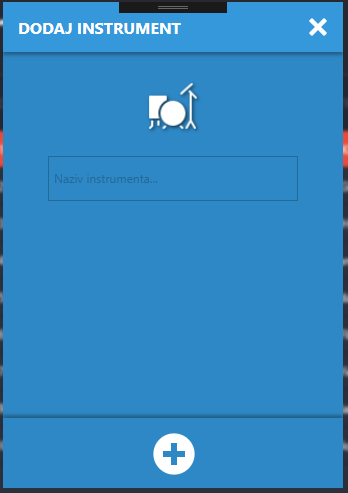


Slika 43 - Deo Code Behind-a za Dodaj novog profesora ([ceo kod](https://pastebin.com/0jV52HG0))

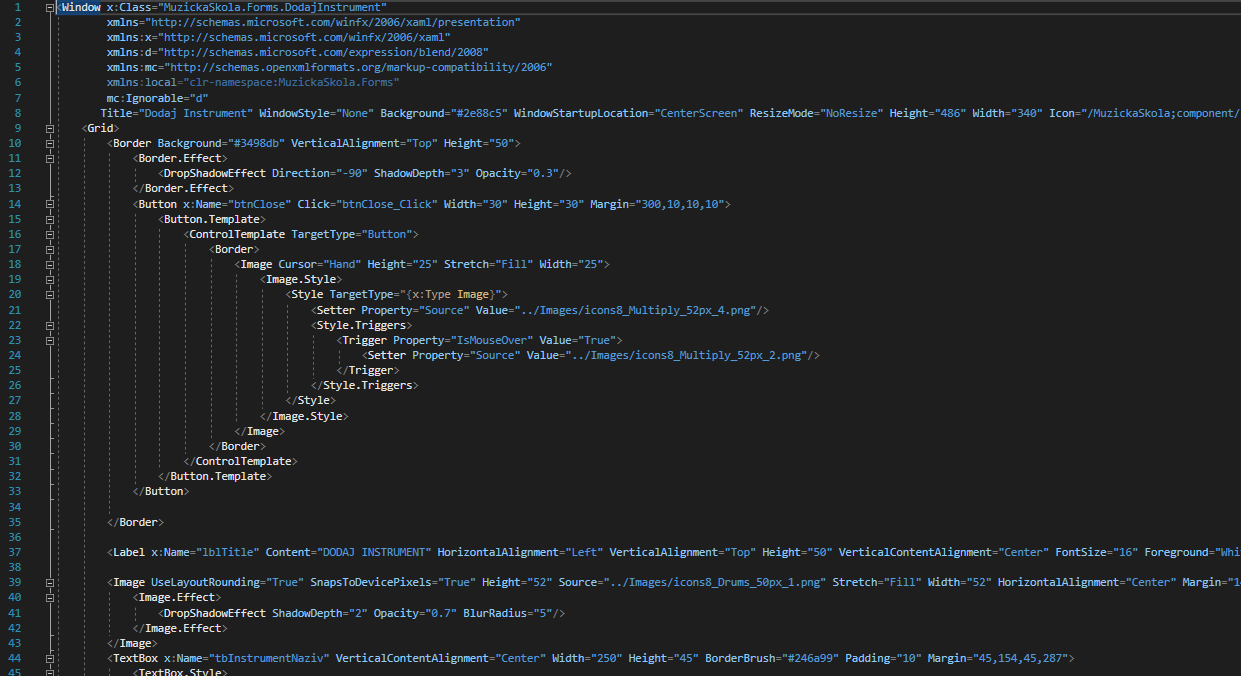


Slika 44 - Sadržaj metode btnAdd\_Click za insert novog Profesora

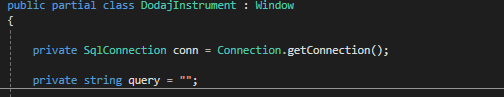
Izgled forme za dodavanje novog Instrumenta takođe ima veoma sličan dizajn (*Slika 45.*).



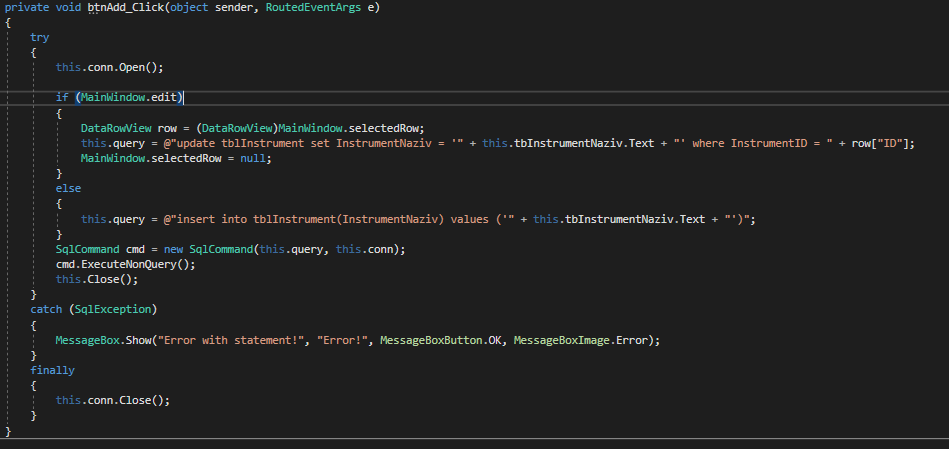
Slika 45 - Dizajn forme za dodavanje novog Instrumenta



Slika 46 - Deo XAML kod-a za formu dodavanja novog Instrumenta ([ceo kod](https://pastebin.com/7AXKha5q))

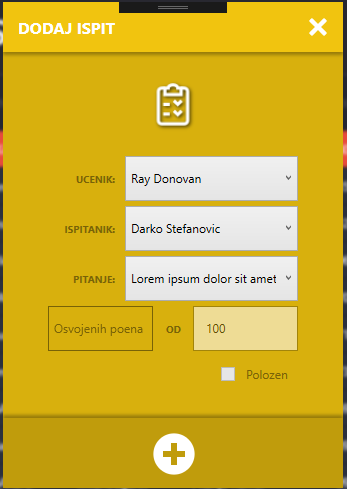


Slika 47 – Izgled Code Behind-a za formu dodavanja novog korisnika ([ceo kod](https://pastebin.com/VS7kAite))



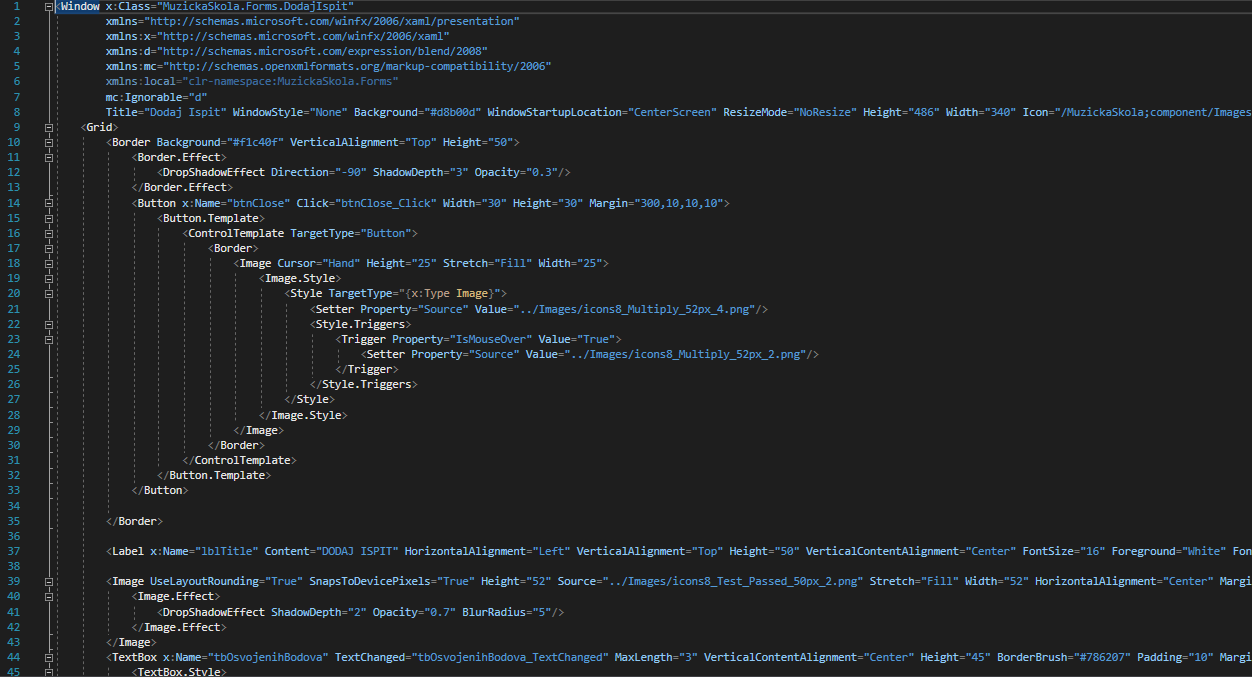
Slika 48 - Sadržaj metode btnAdd\_Click za insert novog instrumenta

Tabele tblIspit, tblUcenik, tblProfesor sadrže u sebi strane ključeve drugih tabela na osnovu kojih se pristupa sadržajima tih tabela. U samoj aplikaciji, potrebno je korisniku obezbijediti vizuelni prikaz željenih obilježja neke od povezanih tabela. U WPF tehnologiji to se najjednostavniji način postiže korišćenjem *ComboBox*-a. U formi za dodavanje novog ispita je implementiran system automatskog odredjivanja da li je ispit polozen ili ne, potrebno je samo uneti broj osvojenih bodova na testu. Takodje, sistem automatski brise sve sto je uneto u polje “Osvojeni poena” ako nije broj, a ako se upise broj veci od 100, postavlja se na 100. Ukoliko broj osvojenih poena predje 50, CheckBox “Polozen” se postavlja u stanje “Checked”, ako se upise 50 ili manje, postavlja se u stanje “Unchecked”. Tu grnicu mozemo promeniti u Code Behind-u, a ona je smestena u posebnu promenljivu pod nazivom “minZaProlaz”, tipa int.



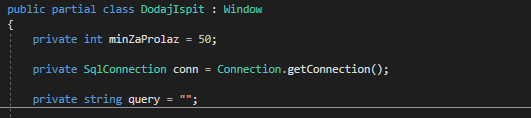
Slika 49‑Dizajn forme za dodavanje novog ispita

XAML kod ove forme prikazan je na *Slici 50.* Prilikom kreiranje novog *ComboBox*-a, potrebno je da definišemo *DisplayMemberPath,* koji predstavlja obelezje povezane tabele koje će se prikazati u padajućoj listi kada korisnik klikne na taj *ComboBox*, kao i *SelectedValuePath* koji predstavlja obilježje primarnog ključa na osnovu kog će se uneta vrijednost iz prikazanog *ComboBox*-a povezati sa svojim identifikacionim obelezimo i kao takva proslijediti u bazupodataka.



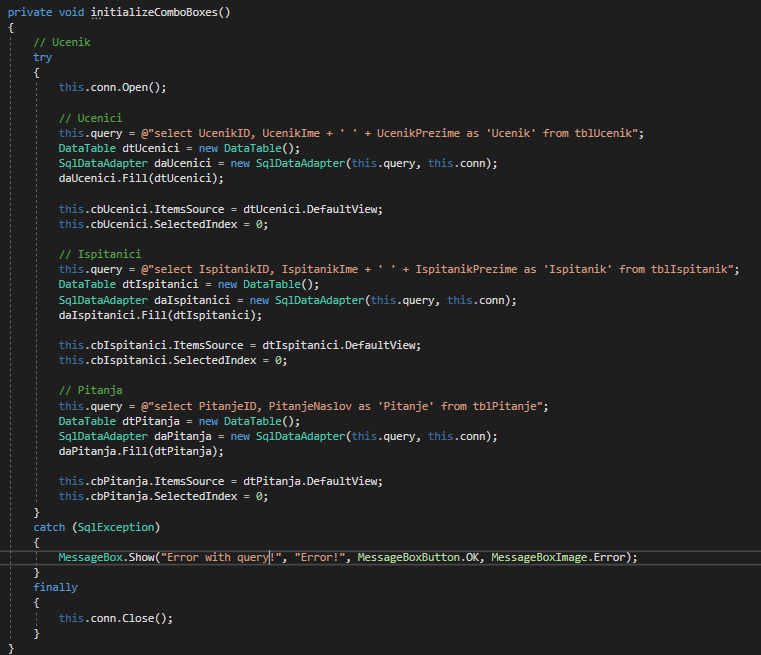
Slika 50 - Deo XAML kod-a forme za dodavanje novog ispita ([ceo kod](https://pastebin.com/pmJ8PdUM))

*CodeBehind* formi koje u sebi sadrže *ComboBox*-ove razlikovaće se od ostalih.

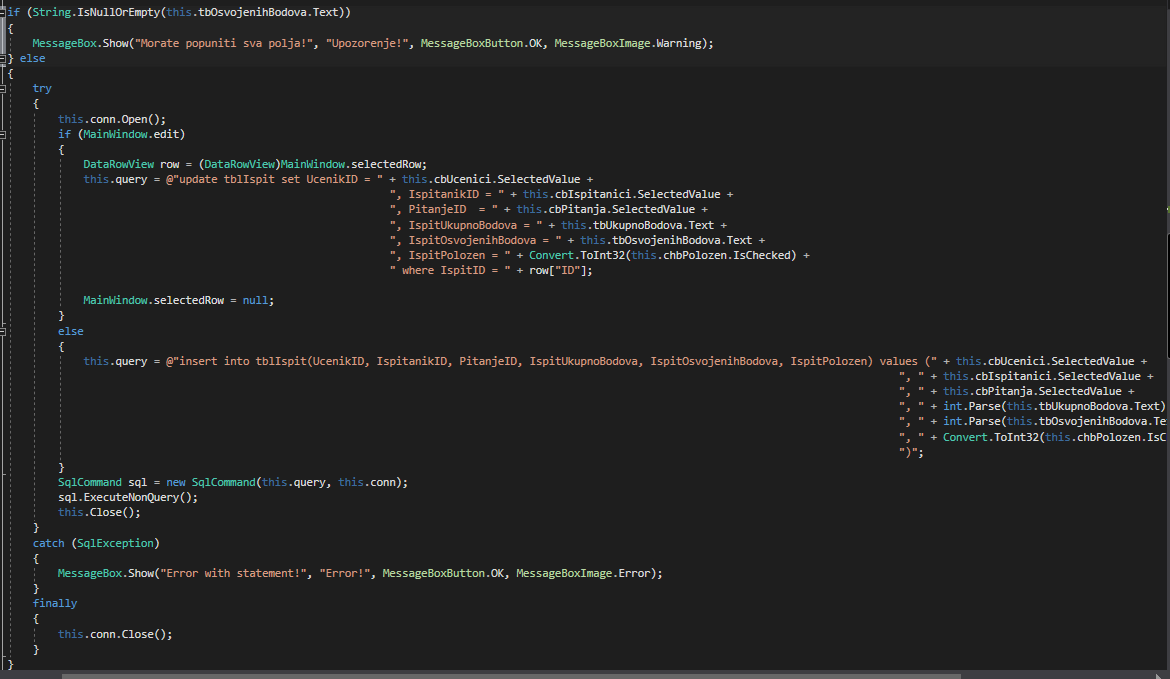


Slika 51 - Code Behind forme za dodavanje novog ispita ([ceo kod](https://pastebin.com/PR6rFS9A))

*Try* blok prikazan na Slici 52. , smesten je unutar metode koja se poziva u konstruktoru ove klase s obzirom da se,pri instanciranju novog prozora ove forme, podaci koji će biti prikazani u *ComboBox*-ovima moraju negdje privremeno uskladištiti. Za postizanje toga, koristi se objekat *DataTable* klase koji predstavlja kopiju tabele iz baze podataka (sa obelezjima koja se proslijede unutar *select* upita), sačuvanu u memoriji aplikacije. Objekat klase *DataTable* se popunjava pomoću objekta klase *SqlDataAdapter* kojem prosleđujemo odgovarajući *select* upit kao i konekciju ka odgovarajućojbazi podataka. Zatim se taj *DataTable* postavlja kao *ItemsSource* za odgovarajući *ComboBox*.Na samom kraju, u *finally* bloku, potrebno je obezbijediti da se konekcija ka bazi bezbijedno zatvori.

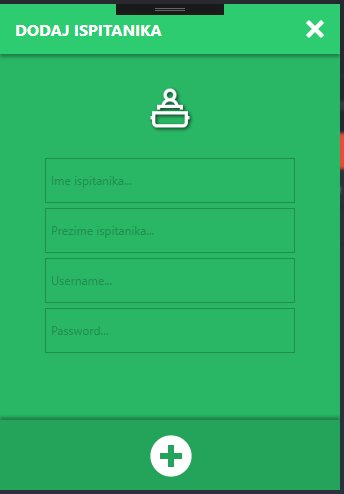


Slika 52 - Izgled metode za dodavanje novog ispita

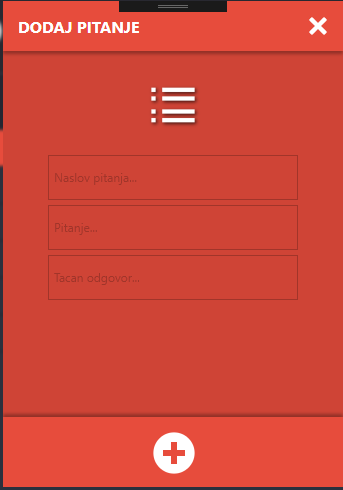


Slika 53 – Izgled btnAdd\_Click za odavanje novog lijeka

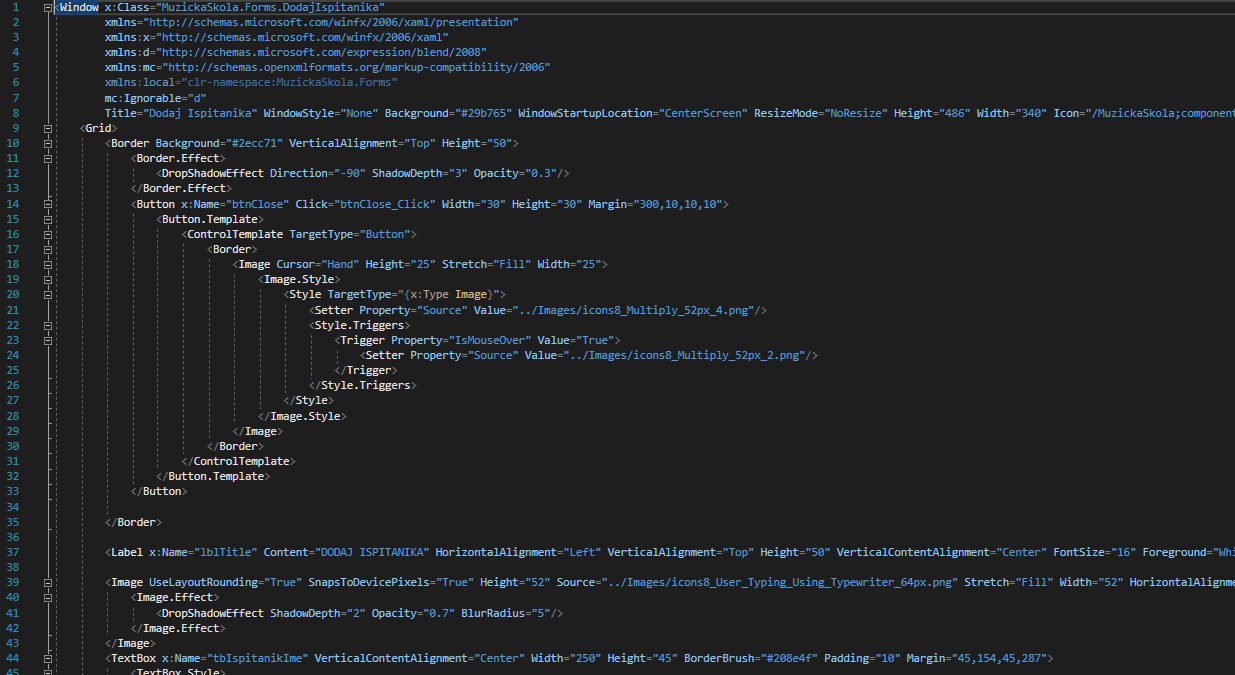
Forme za *pravljenje novog Ispitanika i Pitanja* izgledaju sasvim identično.



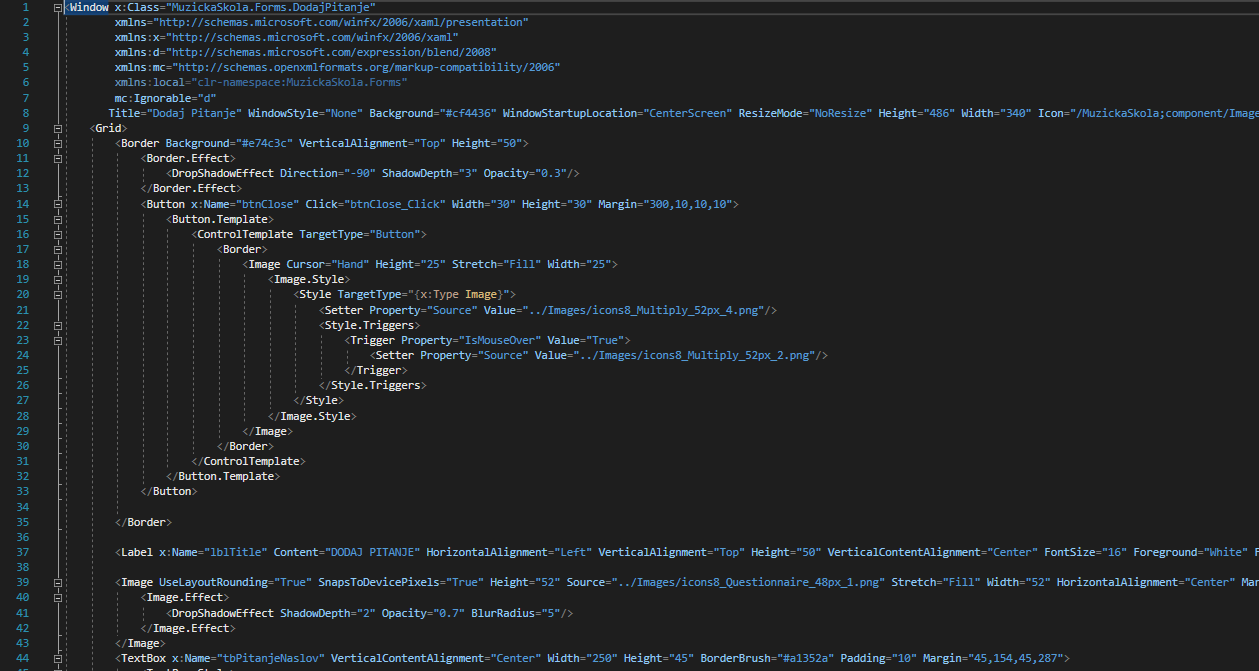
Slika 54 - Dizajn forme za pravljenje novog Ispitanika



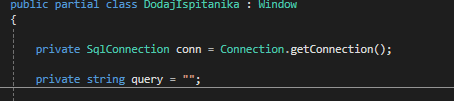
Slika 55 - Dizajn forme za pravljenje novog pitanja



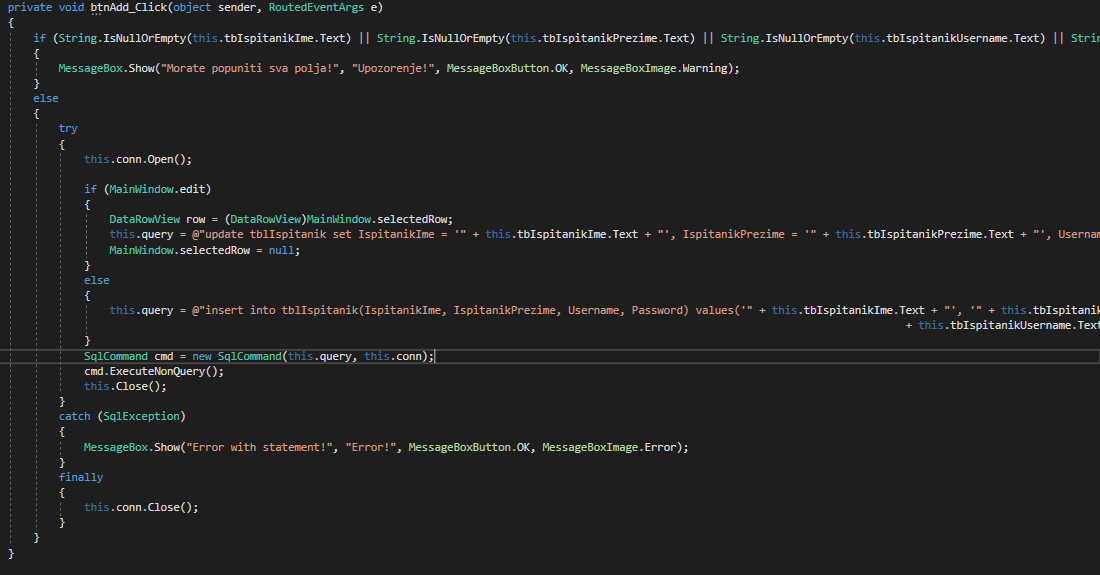
Slika 56 - Deo XAML kod-a za pravljenje nove porudžbine ([ceo kod](https://pastebin.com/bq07W8cj))



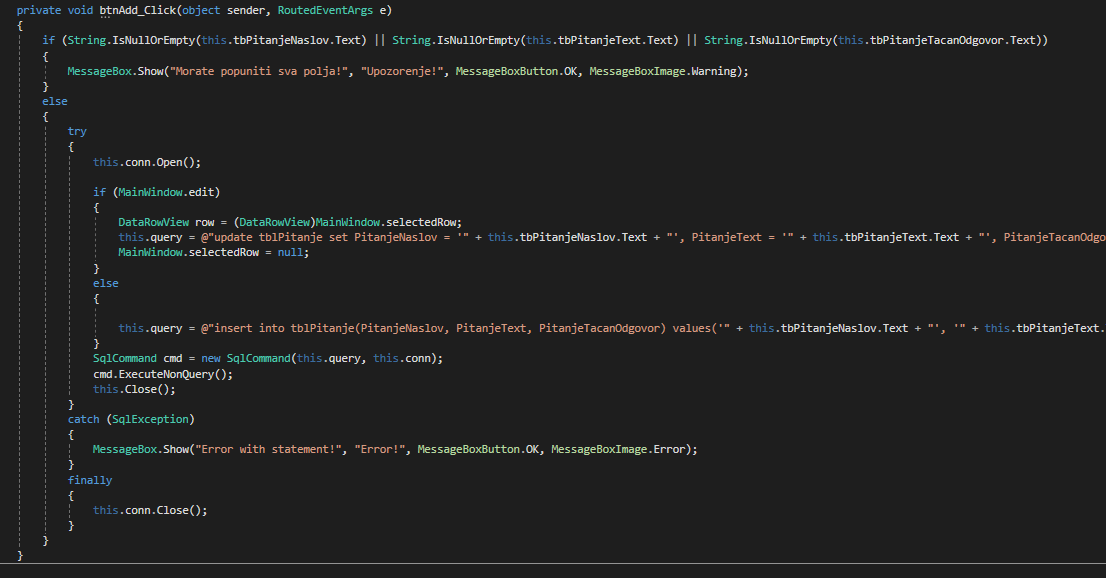
Slika 57 - Deo XAML kod-a za pravljenje nove prodaje ([ceo kod](https://pastebin.com/X2fgxnqi))



Slika 58 - Deo Code Behind-a forme za dodavanje novog ispitanika ([ceo kod](https://pastebin.com/vLdQjbd9))



Slika 59 - Izgled btnAdd\_Click za insert novog ispitanika

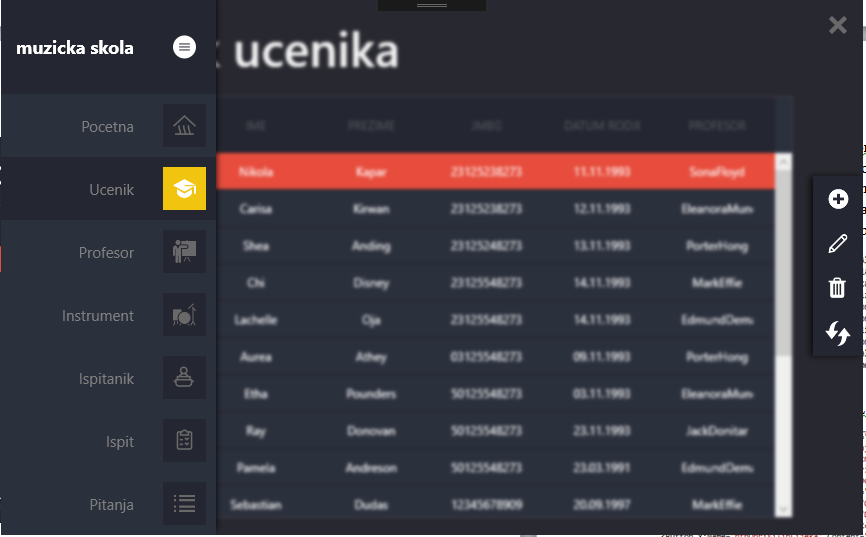


Slika 60 - Izgled btnAdd\_Click za pravljenje novog pitanja

Nakon kreiranja formi za svaku od tabela, potrebno je nešto slično napraviti i za glavni (*Main*) prozor. Main prozor predstavlja prvi, inicijalni, kontakt korisnika sa aplikacijom i ujedno i bazni prozor iz kojeg će se otvarati prethodno kreirane forme i u koji će se korisnik vraćati nakon zatvaranja tih formi. Na njemu bi trebalo da se nalazi dugme za svaku od 6 kreiranih tabela u bazi, čijim će se klikom, na sredini forme iščitavati ti podaci. Takodje, treba obezbediti i meni za CRUD opcije koje ce korisnik moci da koristi za svaku od tabela. U ovom primeru, to je uradjeno na kreativan nacin, te sa desne strane aplikacije mozemo videti CRUD meni sa zanimljivim efektima i bojama. Sa leve strane se nalazi meni za navigaciju kroz aplikaciju i on se klikom na dugme (*skroz gore*) otvara/zatvara. Ovo je prikazano na *Slici 61*.

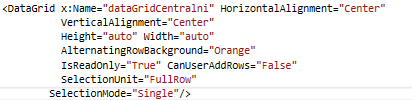


Slika 61 - Vizuelni prikaz glavnog prozora sa zatvorenim glavnim menijem

  
Slika 61 - Vizuelni prikaz glavnog prozora sa otvorenim glavnim menijem

U nastavku, bliže ćemo sagledati XAML kod ove forme.

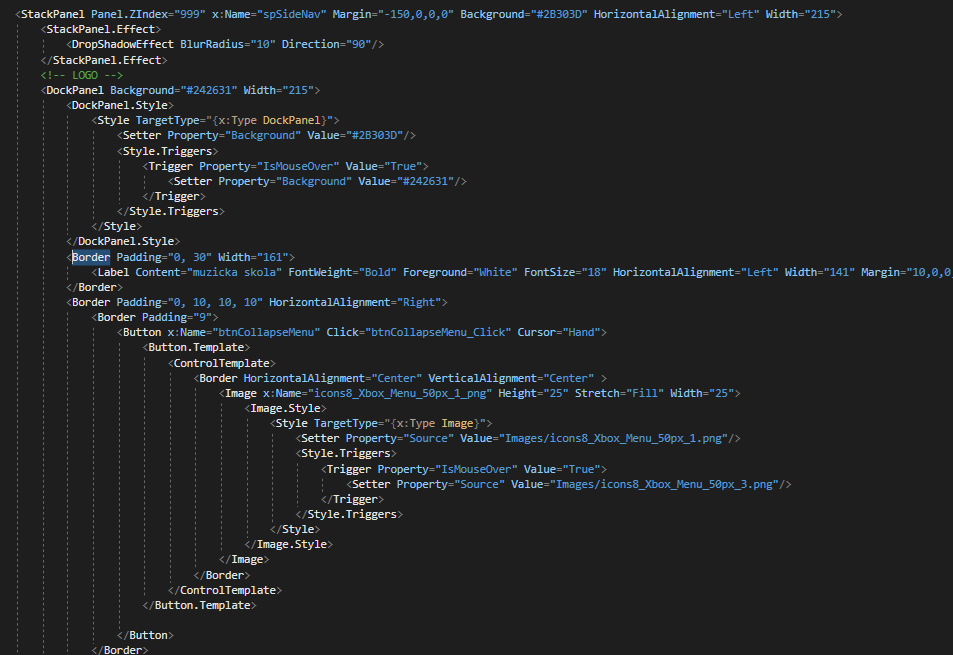
Na samom početku, potrebno je definisati centralni *DataGrid* u okviru kojeg će biti ispisani učitani podaci iz baze. Podešavanja ovog grida takođe su prikazana u okviru *Slike 75*.



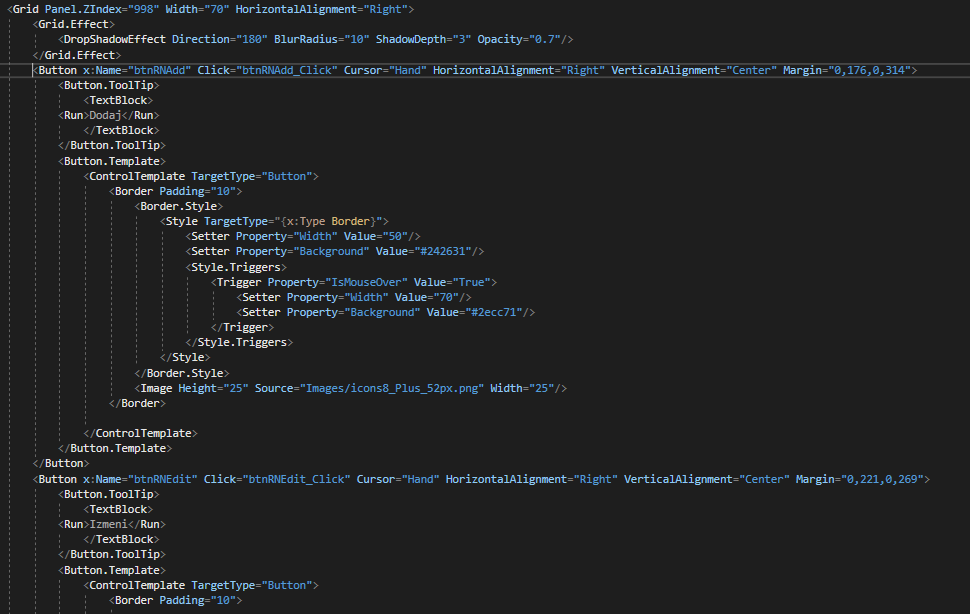
Slika 75 - XAML kod centralnog grid-a

Kao sto mozemo primetiti, otvaranjem menija se takodje dodaje blur efekat na elemente koji ne spadaju i meni, kako bi se dobio efekat fokusiranosti. *Slika 61*

Kako bi elementi unutar glavnog menija izgledali kako treba, koristicemo *StackPanel* kao parent i skup *DockPanela* kao child-ove *StackPanela.* (*Slika 62*)



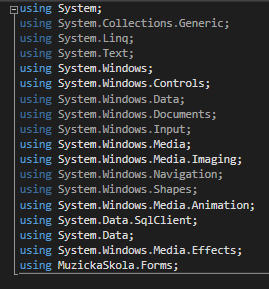
Slika 62 - Deo XAML kod-a Stack Panel-a glavnog prozora ([ceo kod](https://github.com/sebamed/muzicka-skola/blob/master/MuzickaSkola/MainWindow.xaml))



Slika 63 - Deo XAML kod-a za CRUD meni

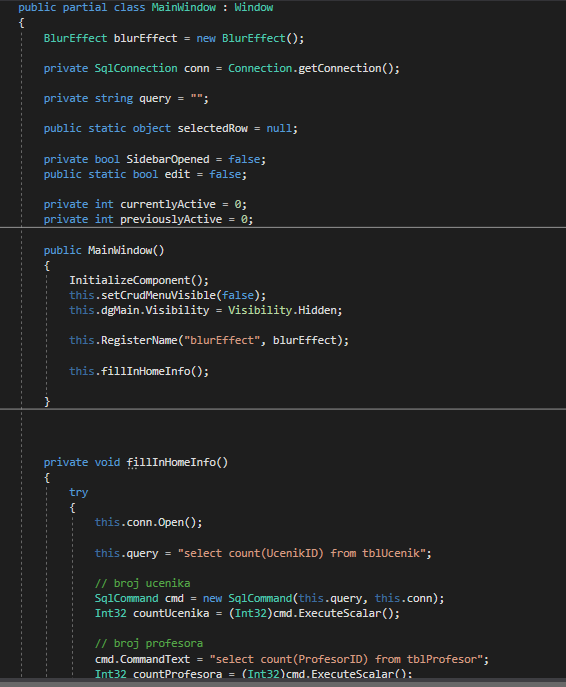
Na *Slici 63* je prikazan deo koda CRUD menija. Ovaj segment je stavljen u *Grid*, a svaka kontrola predstavljena je u obliku dugmeta.

U nastavku biće prikazan Code Behind glavnog prozora gdje je smješten najveći dio poslovne logike. Na samom početku, potrebno je importovati odgovarajuće imenske prostore kao što je prikazano na *Slici 64.*



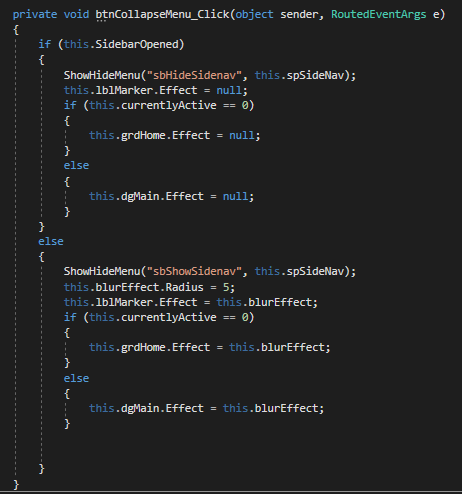
Slika 64 - Imenski proctor za glavni prozor

Sadržina glavnog prozora sadržaće instancu konekcije kao bazi podataka, metodu za popunjavanje početnog grida pri pokretanju prozora, konstruktor za kreiranje novog prozora kao izgenerisane metode iz XAML koda, putem kojih će se, klikom na odgovarajuće dugme, obavljati određene funkcionalnosti. Sadržina je prikazana na *Slici 65.*



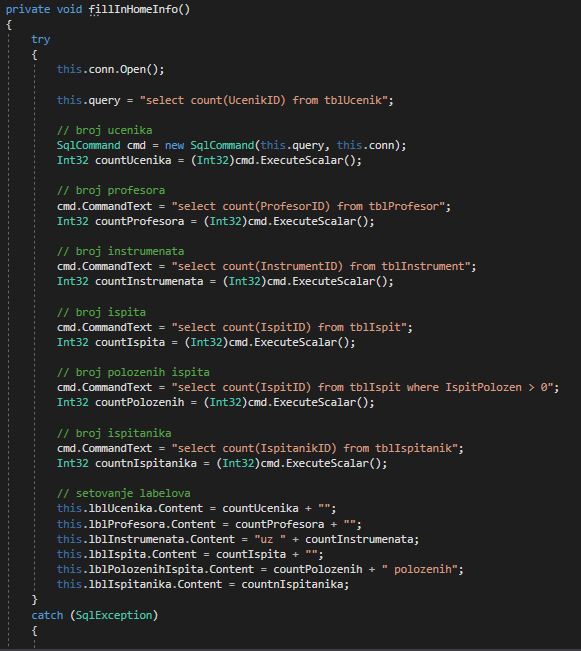
Slika 65 - Sadržaj Code Behind-a glavnog prozora ([ceo kod](https://github.com/sebamed/muzicka-skola/blob/master/MuzickaSkola/MainWindow.xaml.cs))

U aplikaciju je implementiran sistem trenutnog selektovanog. Ovakav sistem void racuna o trenutnom selektovanom delu aplikacije iz glavnog menija,takodje i o proslom selektovanom. Ako pogledamo *Sliku 66,* videcemo kako je ta implementacija uradjena. Ova vrsta pristupa, omogucava lakse kretanje kroz aplikaciju u kodu i otvara nove mogucnosti poput cuvanja memorije, itd. Takodje, ova implementacija je krucijalna za rad glavnog menija. U odnosu na trenutno selektovani segment aplikacije, jedna tablela menja podatke. Takodje, odredjeni delovi aplikacije su sakriveni/prikazani u odnosu na trenutno selektovani segment.



Slika 66 - Sadržaj Code Behind-a glavnog prozora (akcija dodaj)

Nakon instanciranja konekcije, sledeća stvar je kreiranje metode koja će popunjavati *DataGrid* pri otvaranju novog prozora. U ovom primeru, kartica Pocetna ce biti pocetna, odnosno tabele ce se prikazivati tek nakon odabira ostalih kartica iz menija. Na osnovu toga, potrebno je kreirati odgovarajući *Select* upit koji će izlistavati kod svake kartice, korisniku ce na pocetku biti prikazana Pocetna stranica sa kratkim podacima koji se vade iz baze podataka (*Slika 67*).



Slika 67 - Sadrzaj metode za punjenje pocetne kartice

Ove SQL naredbe vracaju broj popunjenih redova od svake od navedenih tabela. Zatim ih smestamo na pocetnu karticu (*Slika 68*)

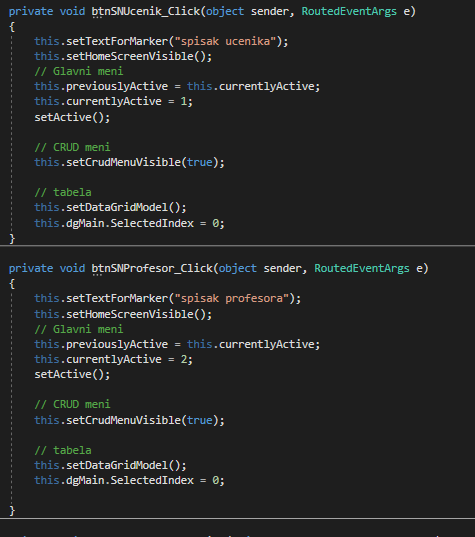


Slika 68 - Visuelni prikaz pocetne kartice

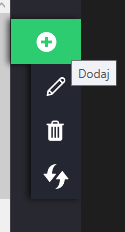
Sledi pregled svake od posebnih metoda vezanih za klik na dugme koje služi za ispis podataka iz tabela. Prva stvar koju ova metoda treba da obezbijedi jeste da se, pored ispisa podataka iz baze, omogući prikaz CRUD menija. Tako nešto postiže se pomoću komandi Visibilty, na način prikazan na slici 69. Posto je, po ovakvom sistemu navigacije, svaka metoda klika ista, prikazacemo samo dve metode.

Slika 84 - Sadržaj metode btnLijekovi\_Click

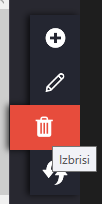
Ispis podataka iz preostalih 8 tabela, vrši se na gotovo identičan način kao za tabelu lijeka sa tom razlikom da će se *Select* upit pisati direktno u metodi. Slijedi primjer za Prodaju (Slike 85-86.). Na ostalim slikama, prikazani su ispisi i za ostalih 7 tabela: Korisnici (Slike 87-88.), Dobavljaci (Slike 89-90.), Kupci (Slike 91-92.), Porudžbina(Slike 93-94.), Tipovi Lijekova (Slike 95-96.), Proizvodjaci (Slike 97-98), Proizvodi (Slike 99-100.).



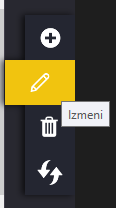
Slika 69



Slika 70 - Vizuelni prikaz CRUD menija (opcija dodaj)



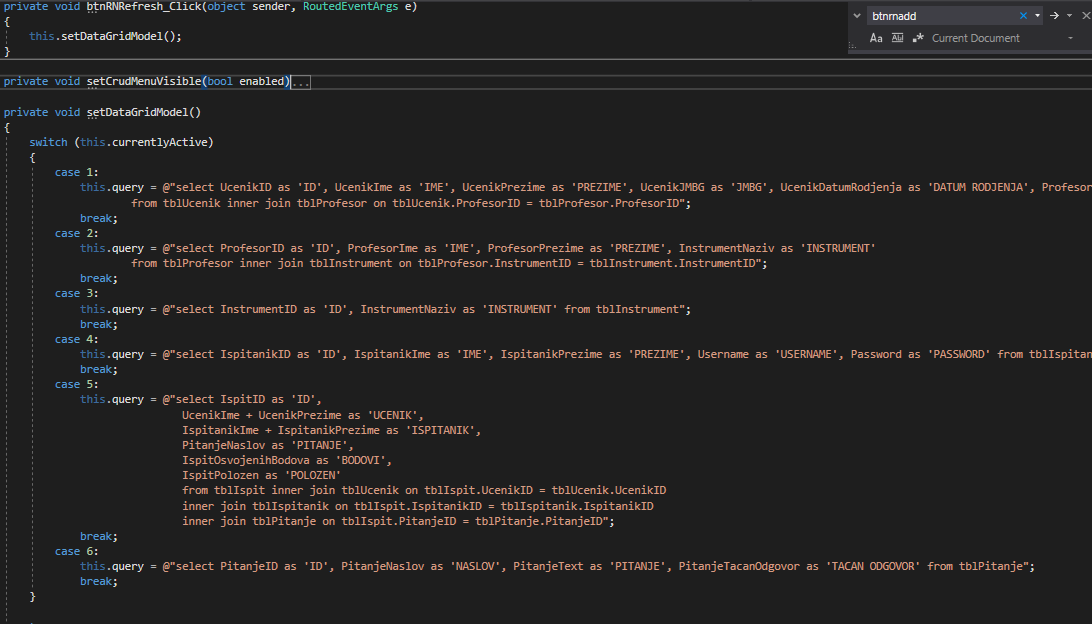
Slika 71 - Vizuelni prikaz CRUD menija (opcija izbrisi)



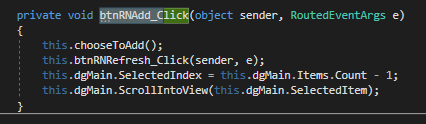
Slika 72 - Vizuelni prikaz CRUD menija (opcija izmeni)



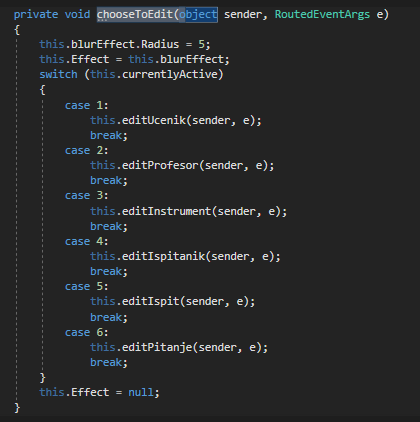
Slika 73 - Vizuelni prikaz CRUD menija (opcija refresuj)



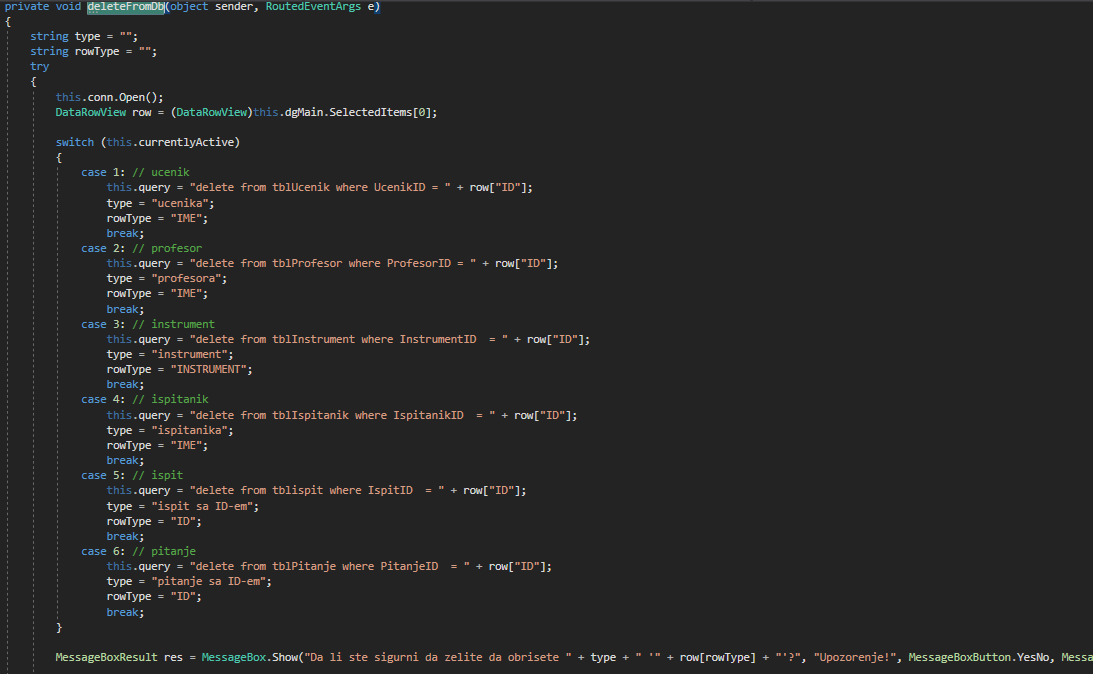
Slika 74 - Sadrzaj metode btnRNRefresh\_Click



Slika 75 – Sadržaj metode btnRNAdd\_Click

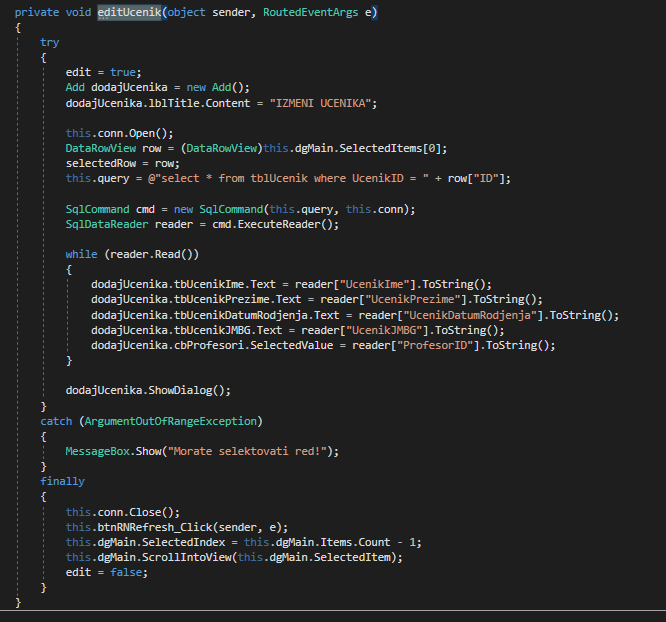


Slika 76 – Sadržaj metode btnRNEdit\_Click

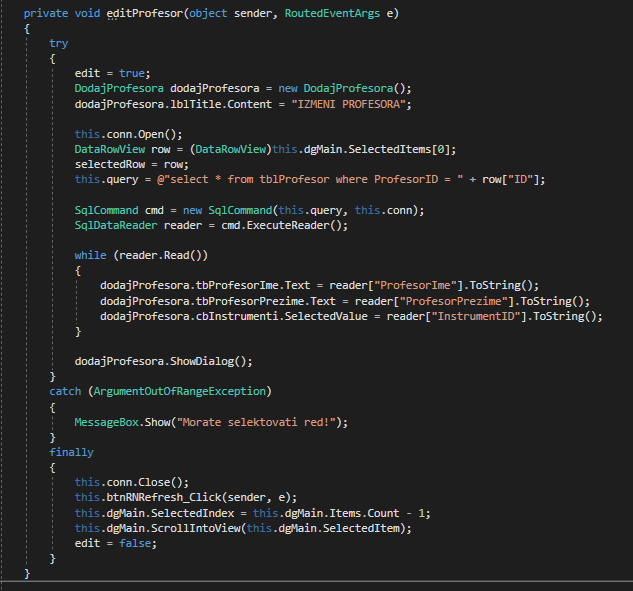


Slika 77 – Sadrzaj metode btnRNRemove\_Click

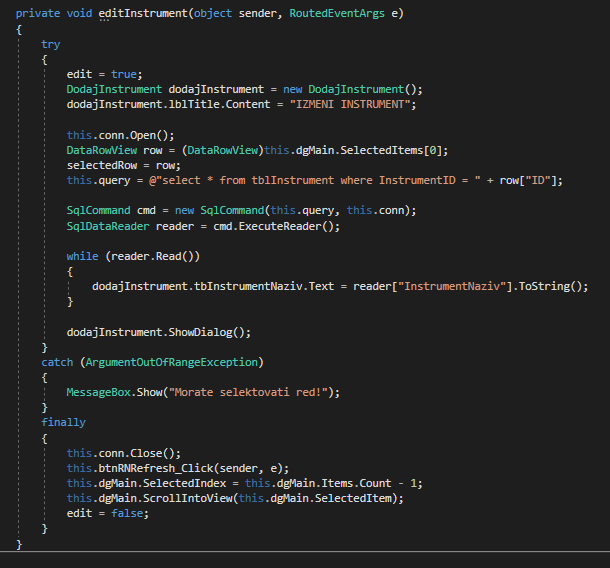
Editovanje svake tabele ove aplikacije ima svoju metodu. Njihov prikaz na *Slikama(78-84)*



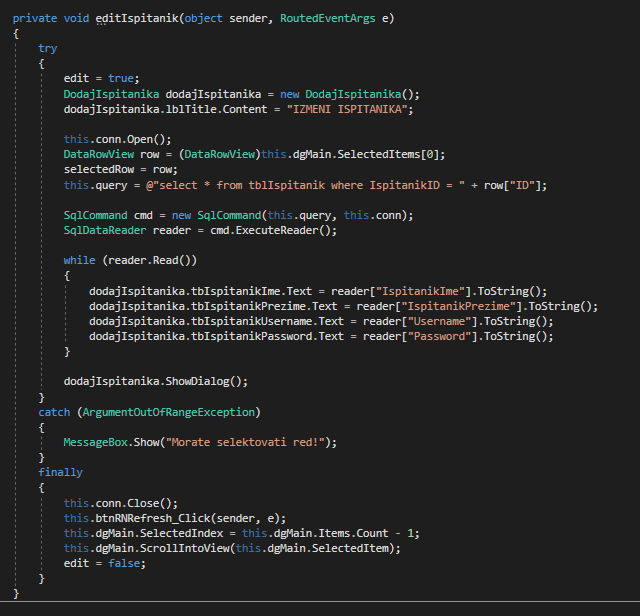
Slika 78 – Sadrzaj metode editUcenik



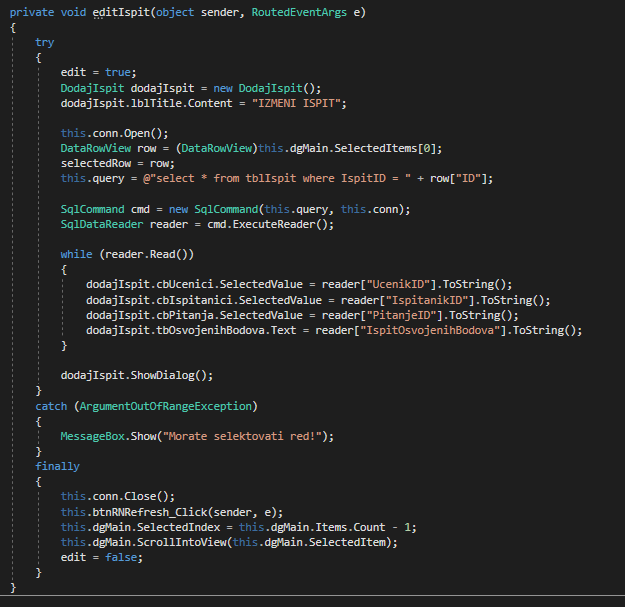
Slika 79 – Sadrzaj metode editProfesor



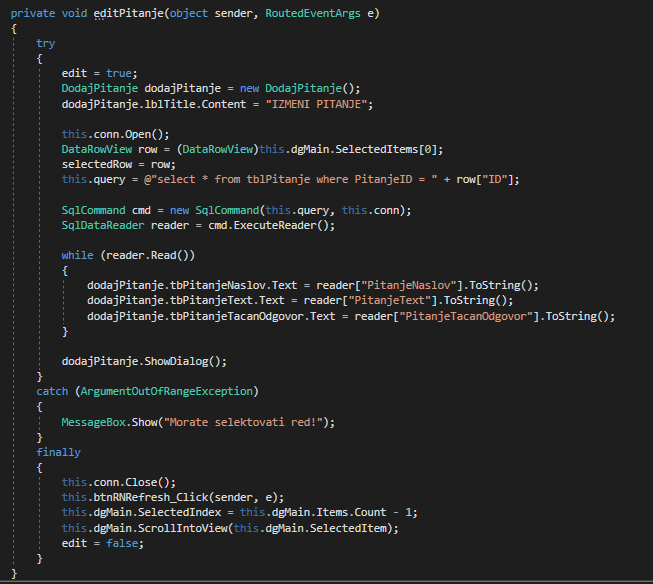
Slika 80 – Sadrzaj metode editInstrument



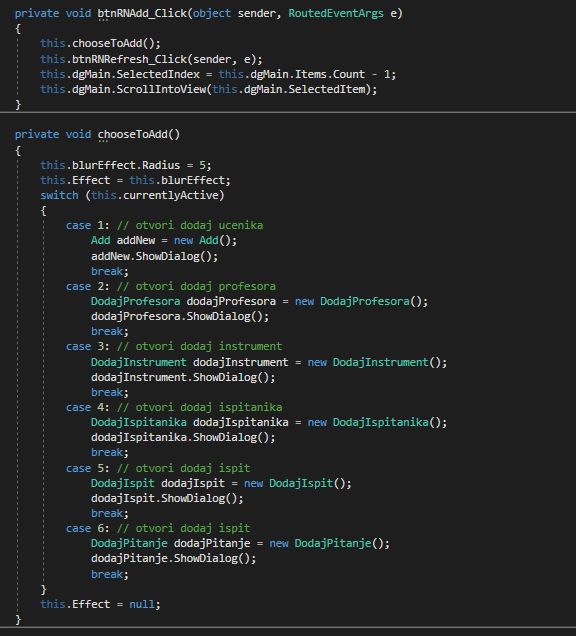
Slika 81 – Sadrzaj metode editIspitanik



Slika 82 – Sadrzaj metode editIspit



Slika 83 – Sadrzaj metode editPitanje



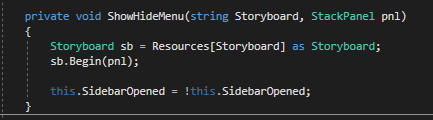
Slika 84 - Sadržaj metode chooseToAdd

Kao na *Slici 84,* metoda choose to add se poziva nakon klika na dugme, i ona regulise koji ce se prozor otvoriti. U metodi btnRNAdd\_Click, nakon pozivanja metode chooseToAdd, imamo refreshovanje tabele i postavljajne selektovanog reda na poslednji, preko metode Items koja vraca kolekciju Item-a iz *DataGrida,* a zatim metode Count koja vraca broj Item-a u kolekciji. Posto *DataGrid* broji po indexima, moramo oduzeti 1 od broja kojeg metoda Count vrati, kako bi smo izbegli *OutOfRange* gresku. (*Slika 85*)



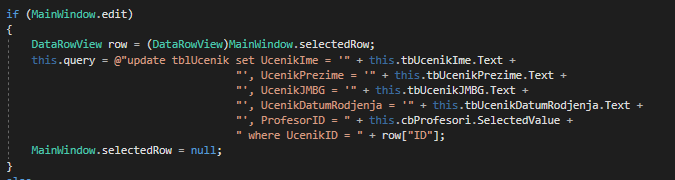
Slika 85 - Refresh, selektovanje reda, i skrol na kraj

Posto smo gore izostavili diskusiju o otvaranju i zatvaranju menija, na *Slici 86* cemo prikazati kako to funkcionise. Klikom na dugme *btnCollapseMenu* se poziva metoda koja otvara ili zatvara meni u zavisnosti od trenutnog stanja, koje je postavljeno kao staticka Boolean promenljiva u MainWindow.xaml.cs fajlu. Nakon poziva ove funkcije, Boolean promenljiva menja svoje stanje a meni se otvara/zatvara. Takodje, tu je i, ranije spomenut, blur efekat koji je uskladjen tako da daje efekat fokusa na meni.

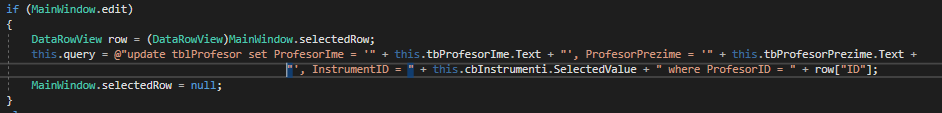


Slika 86- Sadrzaj metode koja otvara ili zatvara meni

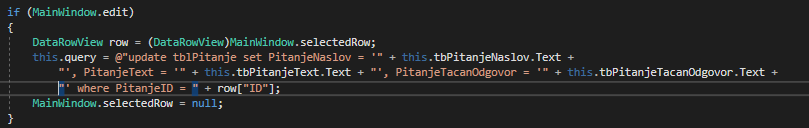
Na narednim slikama prikazano je menjanje zapisa i iz tabela



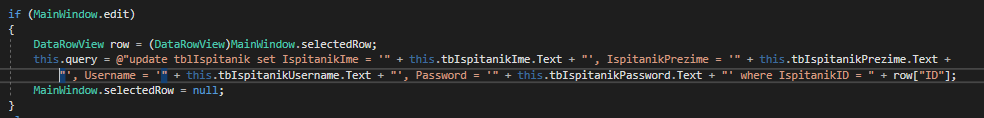
Slika 87 – Editovanje ucenika



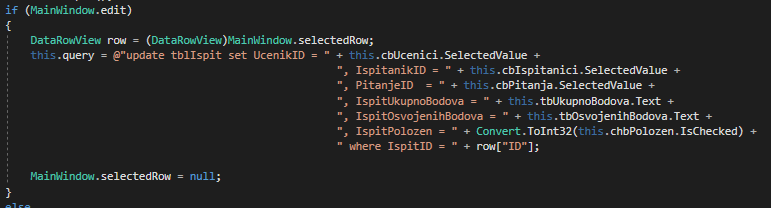
Slika 88 – Editovanje profesora



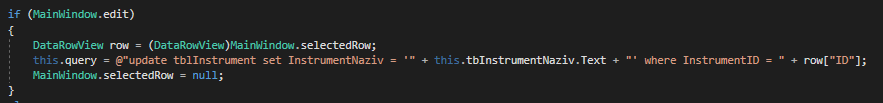
Slika 89 – Editovanje pitanja



Slika 90 – Editovanje ispitanika

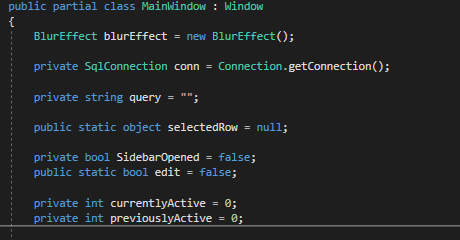


Slika 91 – Editovanje ispita



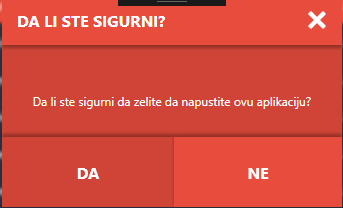
Slika 92 – Editovanje instrumenta

Za modifikaciju podataka, kao i za dodavanje, potrebno je da postoji odgovarajuća forma u koju će korisnik mijenjati odnosno unositi željene vrijednosti. U ovom primeru, za svaku od tabela već je kreirana forma putem koje se novi zapisi upisuju u bazu podataka. Kako ne bismo morali da kreiramo novu formu specijalno za modifikaciju podataka može se iskoristiti postojeća za dodavanje s tim što će jedina razlika biti u tome koju akciju izvršava dugme *Sačuvaj* te forme kada ga korisnik klikne. Na samom početku potrebno je, u okviru glavnog prozora deklarisati dve promenljive (*Slika 93.)*. Prva je tipa *boolean* i koristiće se kao uslov koji će dugmetu *Add* govoriti da li je potrebno podatke upisati u bazu kao novi entitet ili izmeniti. Druga je tipa object i služi za pamćenje vrednosti selektovanog reda kada iz glavnog prozora pređemo u neku drugu formu. Takodje, medju ovim promenjivama se nalaze i varijable instance, indikatori trenutnog i proslog aktivnog polja, niz karaktera query i potrebne stvari za funkcionisanje glavnog menija (*BlurEffect* i *Boolean SidebarOpened)*

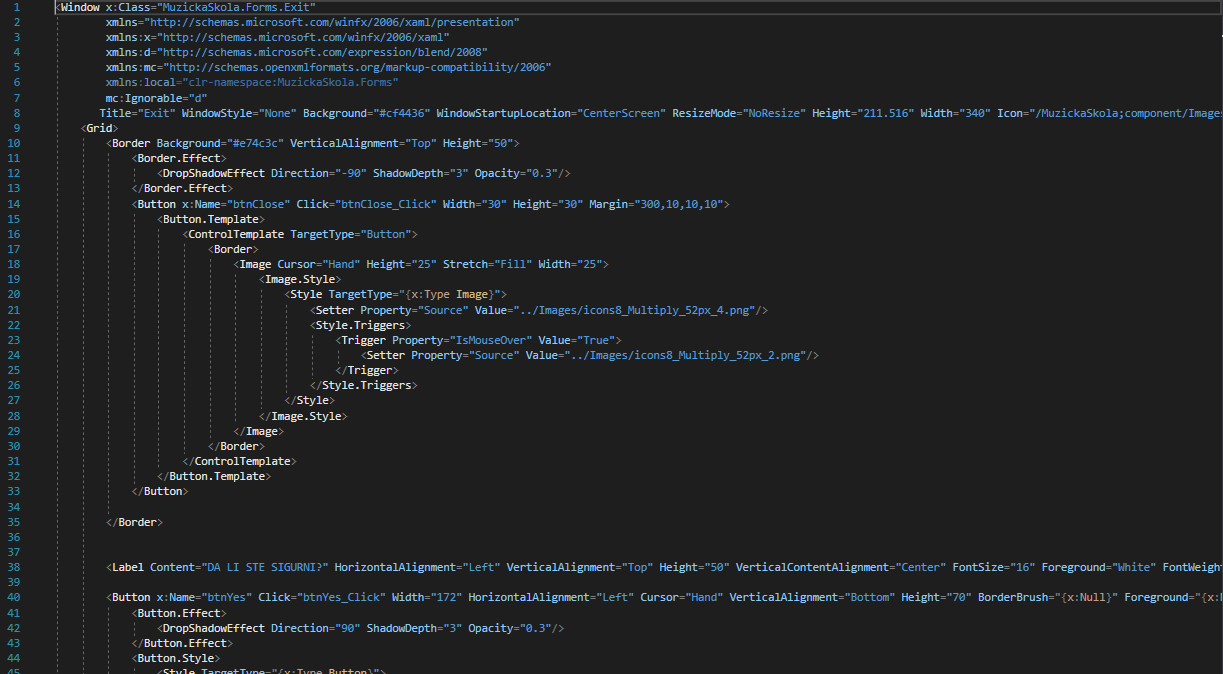


Slika 93 – Deklaracija promjenljivih

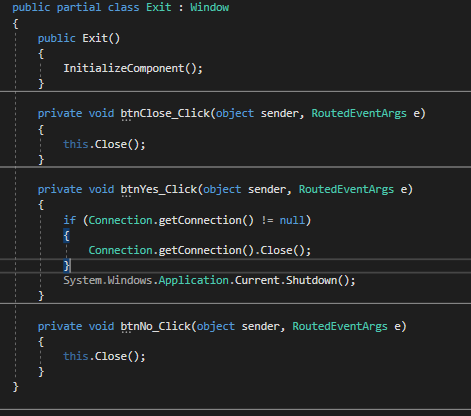
Ako pogledamo *Sliku 61 ili 62*, primeticemo da nema “default-nih” komandi za minimizovanje, povecavanje i gasenje programa. To je zato sto su one izbacene sa namerom. Umesto njih je uveden novi prozor koji izgleda kao na *Slici 94*, i ima funkciju da, ako je odgovor pozitivan zatvori konekciju sa bazom, a ako je negativan, vrati tamo gde je korisnik stao. Takodje, i ovaj prozor pali blur efekat za pozadinu radi fokusa.



Slika 94 – Vizuelni prikaz Exit menija



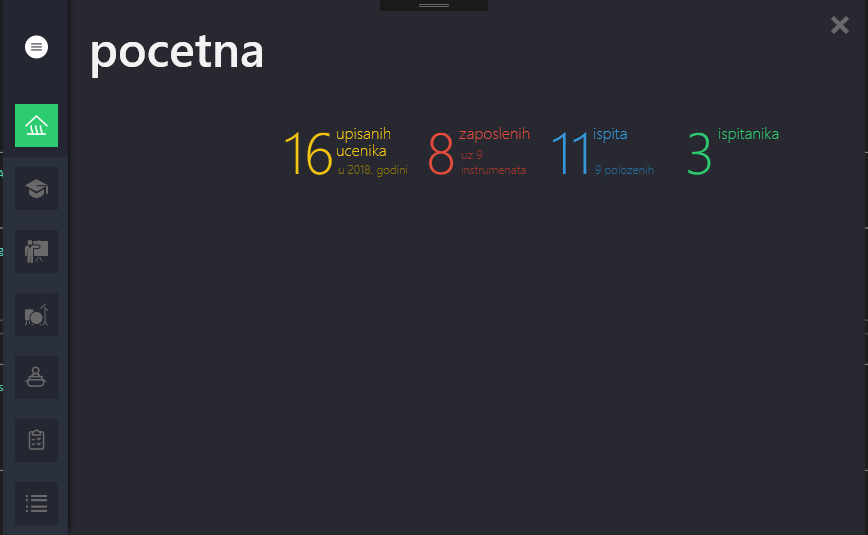
Slika 95 – Deo XAML kod-a Exit menija ([ceo kod](https://github.com/sebamed/muzicka-skola/blob/master/MuzickaSkola/Forms/Exit.xaml))



Slika 96 – Pozadinski kod Exit menija

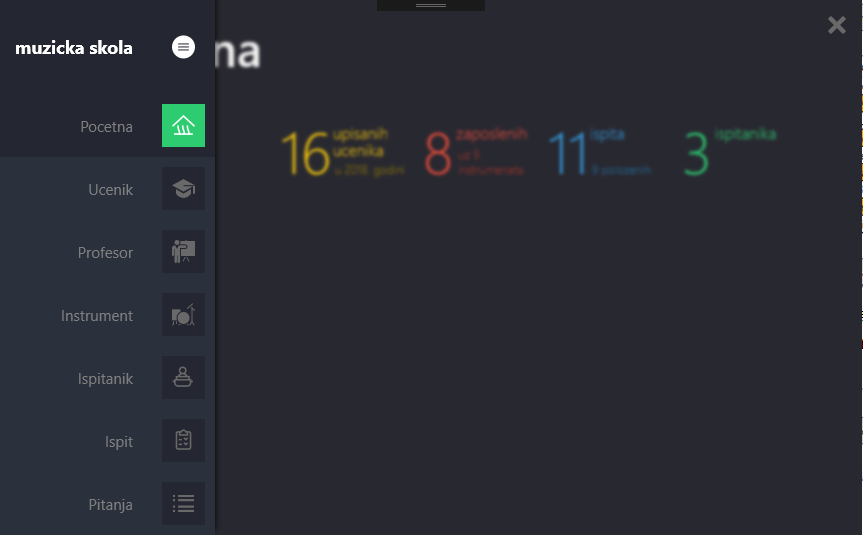
# Testiranje aplikacije

Pri pokretanju aplikacije generiše se prozor kao na *Slici 97.*



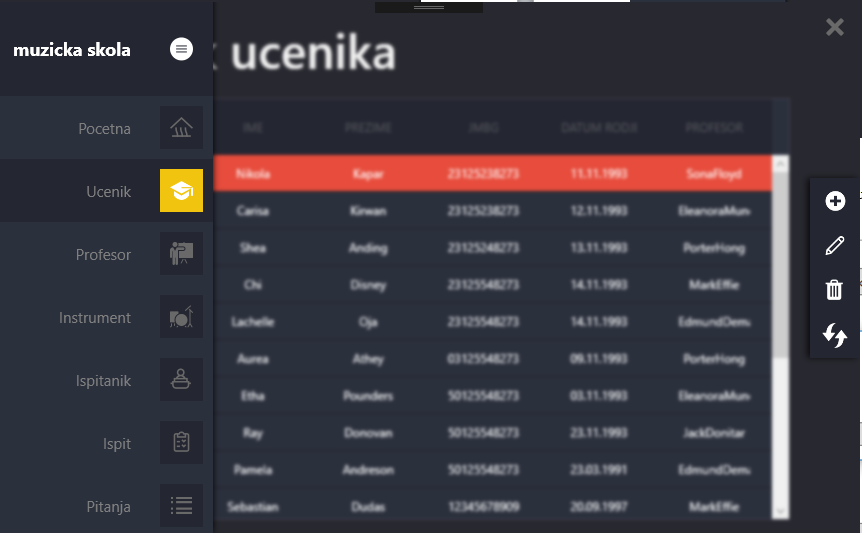
Slika 97 - Početni prozor pri pokretanju aplikacije

Klikom na dugme za otvaranje menija vrsi se prikaz sa *Slike 98*.



Slika 98 - Prikaz menija

Klikom na bilo koje dugme u meniju, pocetna kartica se sakriva, i otkriva se DataGrid sa podacima u zavisnosti od kliknutog dugmeta (*Slika* *99.*)



Slika 99 - Prikaz drugih kartica uz otvoreni meni

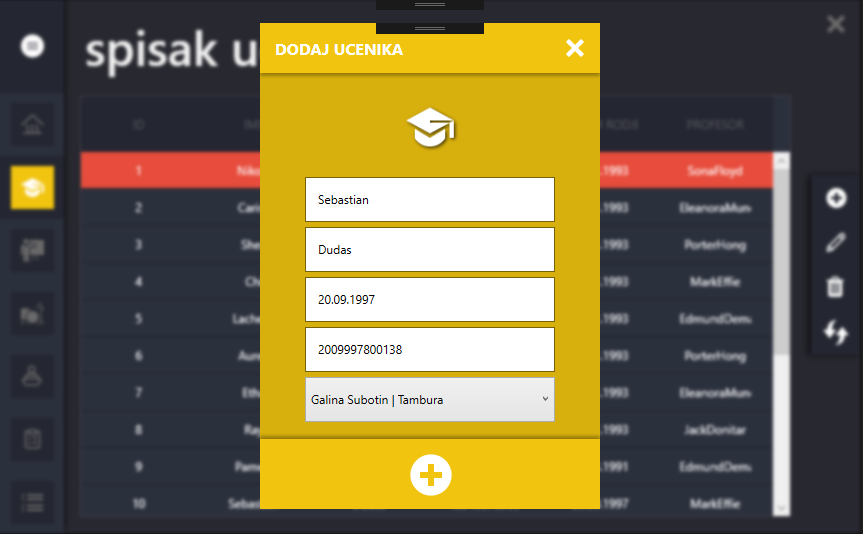
Da bi smo zatvorili meni, i odmaglili tabelu, potrebno je da ponovo pritisnemo dugme za otvaranje/zatvaranje menija.



Slika 100 - Prikaz drugih kartica bez otvorenog menija

Prvi red u bilo kojoj tabeli je automatski selektovan, i omogucene su komande iz CRUD menija. Posto je jedan red uvek selektovan, nema potrebe za prozorom koji upozorava na to.

Dodavanje novog ucenika *Slika 101*.

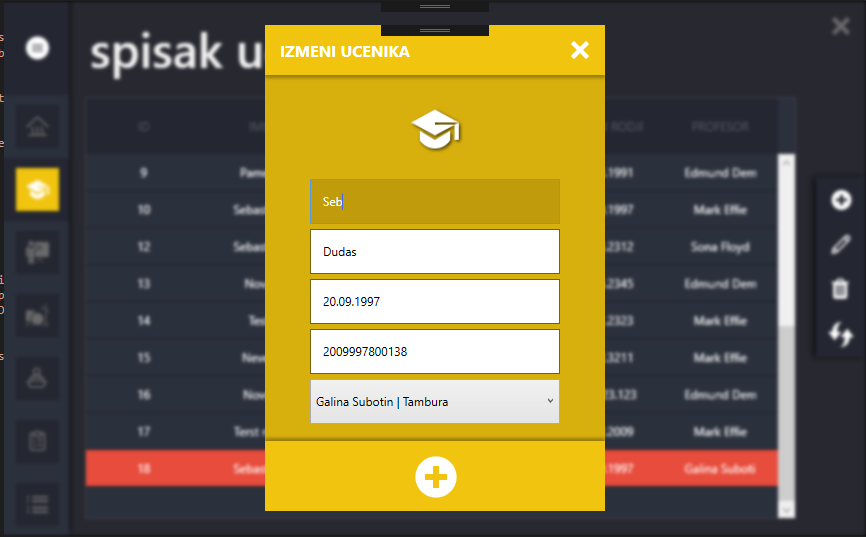


Slika 101 - Dodavanje ucenika



Slika 102 – Dodat ucenik i oznacen, refreshovana tabela

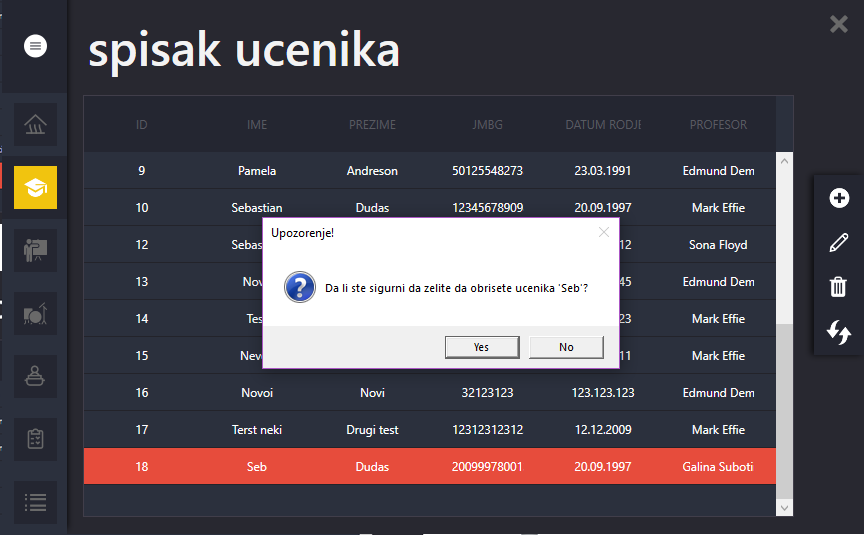
Izmena podataka takodje menja naslov prozora za izmenu iz DODAJ u IZMENI, *Slika 103*



Slika 103 - Izmena podataka



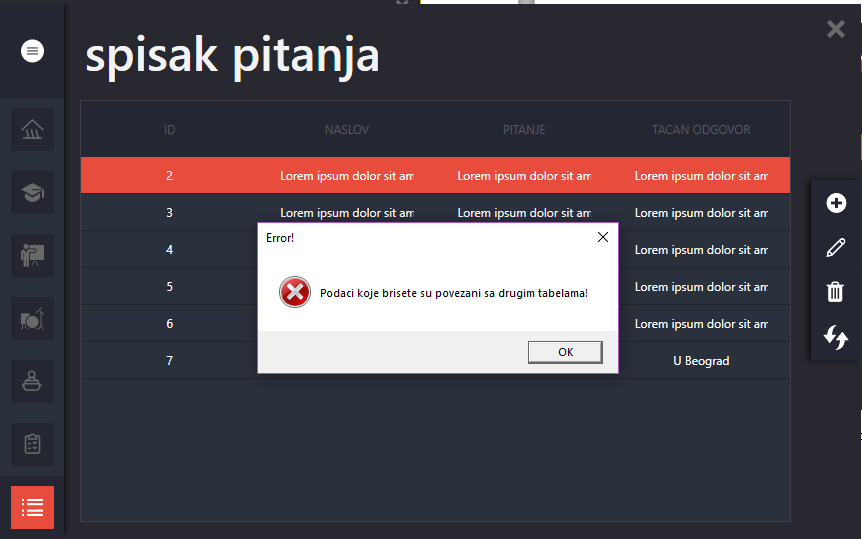
Slika 104 - Izmena podataka uspesno izvrsena



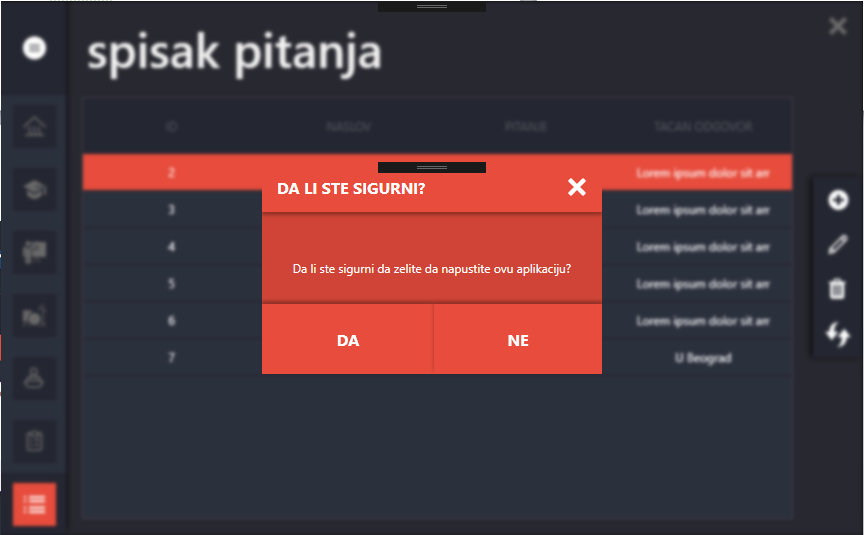
Slika 105 - brisanje podataka sa upitom



Slika 106 - brisanje podataka uspesno izvrseno i selektovan prvi red



Slika 106 - pokusaj brisanja podataka koji su povezani sa ostalim podacima iz drugih tabela



Slika 107 - Izlazak iz aplikacije