Fakultet tehničkih nauka

Univerzitet u Novom Sadu

Trg Dositeja Obradovica 6

**WPF projekat**

**Fudbalski Tim**

Asistent: Student:

Anđela TodorićMilan Ićitović IT43/2022

Novi Sad 2023/24

Sadržaj:

**1.Opis realnog sistema3**

2.Zadatak3

**3.Opis tehnologija3**

4.UML Dijagrami…………………………………………………………………………………………………………………………………5

4.1Use case dijagram………………………………………………………………………………………………………………5

4.2 Dijagram klasa…………………………………………………………………………………………………………………7

4.3 Dijagram sekvenci11

5.Baza podataka11

5.1 Kreiranje baza podataka………………………………………………………………………………………………………11

5.2 Upiti I DLM naredbe:………………………………………………………………………………………………22

6**.Kreiranje WPF aplikacije**…………………………………………………………………………………………………………………..25

7.**Testiranje** …………………………………………………………………………………………………………………………………………32

8.**Zaključak………………………………………………………………………………………………………………………………….......37**

1. **Opis realnog Sistema**

Fudbalski tim predstavlja kompleksni sportski entitet sačinjen od ključnih elemenata poput igrača, trenera, stadiona, navijača, dresova, sezona i trofeja. Ovaj dinamični sistem združuje individualne veštine igrača, taktičku genijalnost trenera, atmosferu na stadionu koju stvaraju vatreni navijači, jedinstvene dresove koji predstavljaju identitet tima, sezone ispunjene izazovima i, naravno, želju za osvajanjem trofeja.

1. **Zadatak**

Sam cilj projekta je kreiranje WPF aplikacije informacionog Sistema koja će omogućiti evidentiranje podataka o igračima, trenerima, navijačima, stadionima, ligama i fudbalskim timovima i time simulirati sistem podataka nekog fudbalskog tima. Aplikacija ima mogućnosti pregledanja informacija o svim glavnim akterima fudbalskog kluba, kao i njihovu izmenu, brisanje ili dodavanje.

1. **Opis tehnologija**

Što se tiče same izrade aplikacije, korišćeno je nekoliko tehnologija.

Prvi deo izrade jeste bila ideja koju sam sproveo u delo putem kreiranja dijagrama. Dijagrame koje sam koristio su dijagram klase, use-case dijagram i dijagram sekvenci. Ove tri vrste dijagrama izrađene su u okviru alata [Gen my model.](https://app.genmymodel.com/api/login) U kratkim crtama ću objasniti koji su ostali alati koji su bili korišćeni u projektu:

GenMyModelje platforma za modeliranje koja omogućava korisnicima da vizualno projektuju i sarađuju na razvoju softvera putem modela. Pruža alate za stvaranje dijagrama, kao što su dijagrami klasa ili dijagrami toka podataka, olakšavajući timsku saradnju i dizajn softvera.

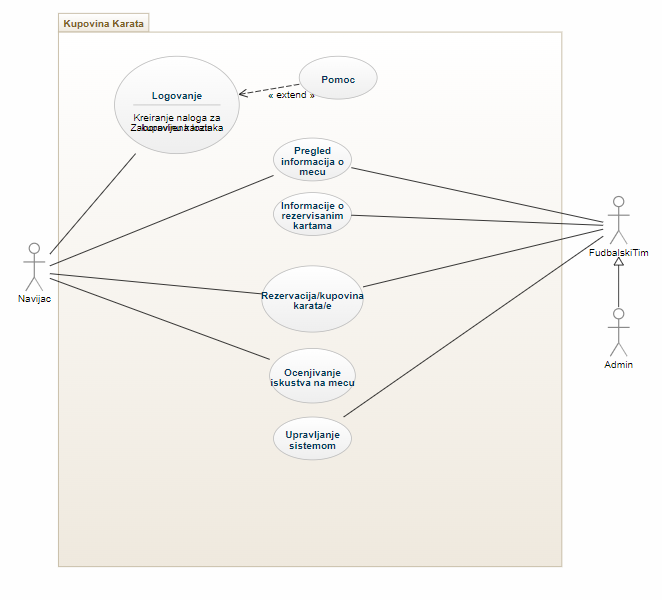
SQL Server Management Studio (SSMS) je alat napravljen od strane Microsoft-a koji omogućava administratorima baza podataka i programerima da upravljaju SQL Server bazama podataka kroz grafički interfejs. SSMS omogućava povezivanje s bazama, izvršavanje upita, dizajniranje struktura baza podataka i obavljanje različitih zadatka vezanih za upravljanje podacima.Prilikom kreiranja aplikacije za salon nameštaja korišćena je Express verzija koja se besplatno može preuzeti.

Visual Studio je integrisano razvojno okruženje (IDE) koje pruža programerima alate za razvoj, testiranje i održavanje softvera na različitim platformama. C# (C-sharp) je moderni programski jezik razvijen od strane Microsofta, često korišćen u okviru Visual Studio-a, koji omogućava razvoj raznolikih aplikacija, uključujući desktop, web i mobilne aplikacije. Izrada aplikacije je rađena u okviru Community Edition verzije Visual Studio-a.

WPF (Windows Presentation Foundation) je tehnologija za kreiranje grafičkih korisničkih interfejsa (GUI) u aplikacijama za operativni sistem Windows. Razvijen od strane Microsofta, WPF omogućava kreiranje modernih Windows desktop aplikacijama kroz upotrebu XAML-a (eXtensible Application Markup Language) i podršku za stilizaciju, animacije i veze sa podacima.

**4) UML dijagrami**

4.1 Dijagram slučaja upotrebe (use-case dijagram)



Slika 1. Dijagram slučaja upotrebe

**Slučaj upotrebe:** Kupovina karata **(Slika 1)**

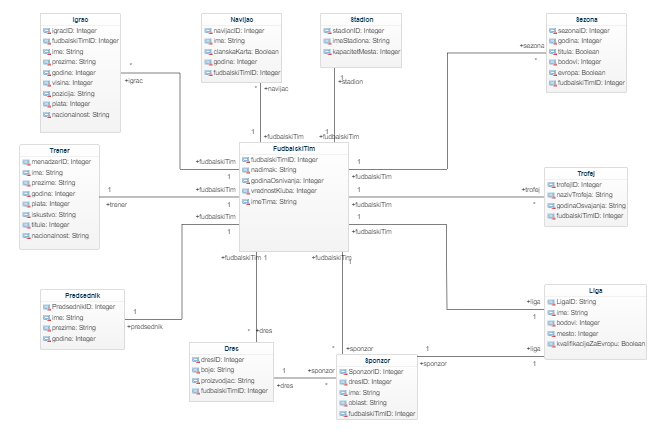
**Kratak opis:** Slučaj upotrebe „Kupovina karata“ opisuje proces kojim navijač može izabrati i obaviti kupovinu tj. Rezervaciju karte za fudbalski meč putem sistema za prodaju karata. Navijač pregleda dostupne opcije i vrste karata, bira željeni broj karata i izvršava transakciju za kupovinu.

**Učesnici:** Navijač, fudbalski tim i admin.

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Navijač mora biti registrovan korisnik sistema. Fudbalski tim mora obezbediti informacije o dostupnim kartama za određeni meč. Admin mora postaviti cene karata i osigurati ispravno funkcionisanje sistema.

**Izuzeci:** Ako nema dostupnih karata za odabrani meč, navijač ne može izvršiti kupovinu. Ako se sistem za prodaju karata sruši ili naiđe na tehnički problem tokom transakcije, kupovina ne može biti izvršena.

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Nakon završetka kupovine, sistem mora ažurirati informacije o preostalim dostupnim kartama za meč. Navijač treba primiti potvrdu o kupovini, uključujući informacije o kartama koje je kupio, datum meča, cenama, redni broj mesta i detaljima o plaćanju. Fudbalski tim treba ažurirati informacije o prodaji karata i očekivanom broju prisutnih navijača na meču.



Slika 4.2. Dijagram klase

U dijagramu klasa je identifikovano 10 klasa. Sada ćemo napraviti kratak opis svake:

**Fudbalski Tim**

Obeležja klase fudbalski tim su sledeća:

fudbalskiTimID (int) – jedinstveni broj određenog tima, tj. Primarni ključ

nadimak (string)

godinaOsnivanja (int)

vrednostKluba (int)

imeTima (string)

ligaID (int)– strani ključ koji nam je potreban kako bi znali u kojoj ligi se tim takmiči

**Igrač**

Obeležja klase igrač su sledeća:

igracID (int) - jedinstveni broj određenog igrača, tj. Primarni ključ

fudbalskiTimID (int) – strani ključ koji nam govori za koji fudbalski klub igrač nastupa

ime (string)

prezime (string)

godine (int)

visina (int)

pozicija (string)

nacionalnost (string)

plata (int)

**Trener**

Obeležja klase trener su sledeća:

trenerID(int) - jedinstveni broj određenog trenera, tj. Primarni ključ

ime (string)

prezime (string)

godine (int)

iskustvo (string)

nacionalnost (string)

titule (int)

plata (int)

**Liga**

Obeležja klase trener su sledeća:

ligaID (int) - jedinstveni broj određene lige, tj. Primarni ključ

ime (string)

bodovi (int)

mesto (int)

kvalifikacijaZaEvropu (boolean)

**Stadion**

Obeležja klase stadion su sledeća:

stadionID (int) - jedinstveni broj određenog stadiona, tj. Primarni ključ

imeStadiona (string)

kapacitetMesta (int)

**Sezona**

Obeležja klase sezona su sledeća:

sezonaID (int) - jedinstveni broj određene sezone, tj. Primarni ključ

fudbalskiTimID(int) – strani ključ, sa njim određujemo tim kojem ćemo gledati učinke u izabranoj sezoni

titula (int)

godina (int)

bodovi (int)

evropa (boolean) – da li je fudbalski klub nastupao na evropskoj sceni

**Trofej**

trofejID(int) -- jedinstveni broj određenog trofeja, tj. Primarni ključ

godinaOsvajanja (int)

nazivTrofeja(string)

fudbalskiTimID(int) – strani ključ, sa njime određujemo klub

**Dres**

dresID (int) - jedinstveni broj određenog dresa, tj. Primarni ključ

boja (string)

proizvodjac (string)

fudbalskiTimID (int) – strani ključ

**Sponzor**

sponzorID (int) - jedinstveni broj određenog sponzora, tj. Primarni ključ

dresID(int) – strani ključ sa kojim određujemo kom dresu ćemo da dodeli sponzora

ime(string)

oblastPoslovanja (string)

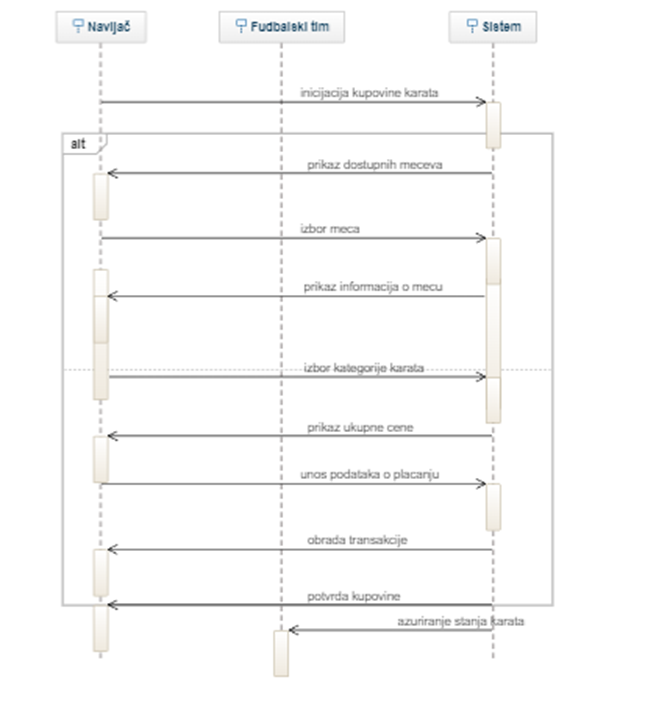
**Navijač**

navijacID (id) - jedinstveni broj određenog navijača, tj. Primarni ključ

ime (string)

clanskaKarta (boolean) – da li se poseduje članska karta ili ne

fudbalskiTimID(int) – strani ključ sa kojim određujemo za koji tim navija navijač



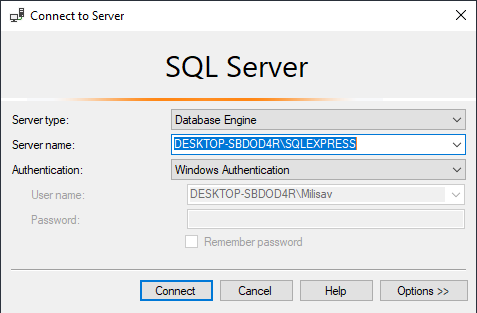
Slika 4.3. Dijagram sekvenci

**5) Baza podataka**

5.1. Kreiranje baze podataka

Na osnovu kreiranog dijagrama klasa u Microsoft SQL Server Management Sudio-u se kreira baza podataka.

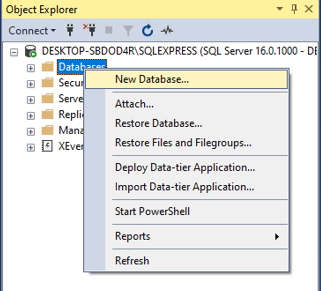
Prvi korak nakon instalacije SSMS je povezivanje na izabrani server prilikom otvaranja SQL Server Management Sudio-a.



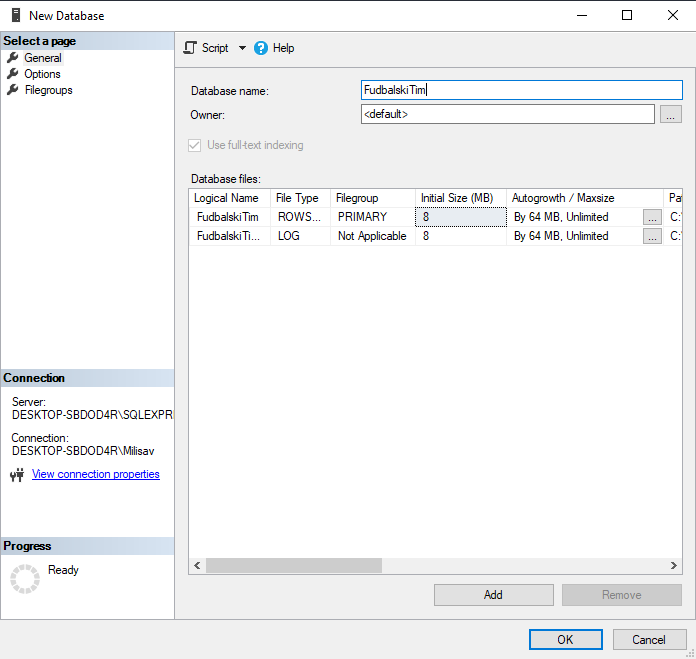
Slika 5.2 Konektovanja na server

Nakon povezivanja kreira se baza podataka na sledeći način:

U okviru Object explorer-a sa desne strane, desni klik na Databases, i u listi koja se otvori treba izabrati opciju New Database.

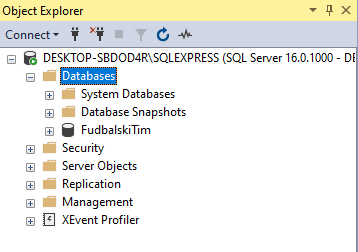
 Slika 5.3 Kreiranje nove baze

Nakon klika se otvara prozor u kome je potrebno zadati ime novoj bazi. Na ovom primeru to će biti FudbalskiTim.



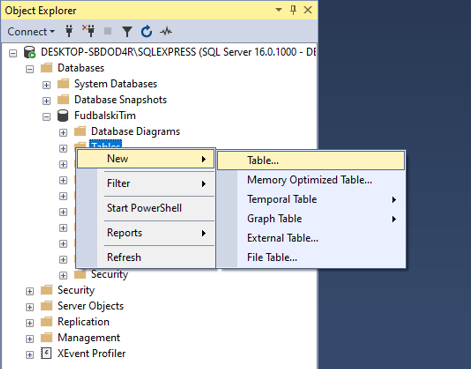
Slika 5.4 Zadavanje imena baze

Nakon klika na OK se u okviru servera, u folderu Databases pojavljuje baza podataka FudbalskiTim.

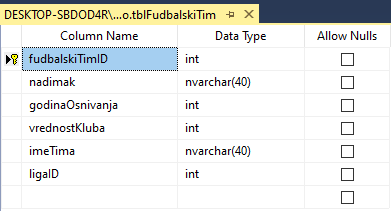


Slika 5.5 Prikaz kreirane baze podataka

Kada se baza podataka kreira potrebno je napraviti nove tabele u okviru nje. Dvoklikom na bazu podataka otvara se lista u kojoj se nalazi folder Tables. Desnim klik na folder Tables -> Mouseover New -> Klik na Table...



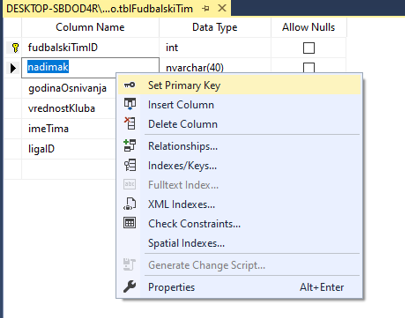
Slika 5.6 Kreiranje naših tabela

U sledećem koraku potrebno je popuniti novu tabelu sa atributima (u okviru column name-a), odrediti tip podatka za svaki atribut (ispod data type) i odrediti da li taj atribu može da prima null vrednosti ili ne. 

Slika 5.7 Dodavanje atributa u naše tabele

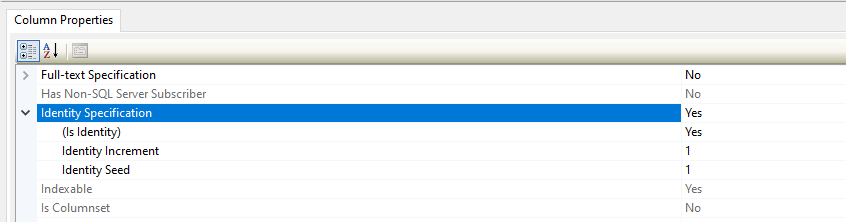
Svaka tabela mora da ima primarni kljuc. Kolona koja se oznaci kao primarni ključ ima naziv nazivAtributaID, mora da bude tipa int i za njega nisu dozvoljene null vrednosti.

Primarni ključ se postavlja desnim klikom na naziv atributa i klik na Set Primary Key (u ovom slučaju je već postavljen za atribut namestajID pa za njega se ne nudi u listi ako se na njega klikne desni klik).



Slika 5.8 Postavljanje primarnog ključa

Kako bi omogućili da se primarni ključ automatki kreira i prilikom svakog novog unosa povećao za jedan idemo na Column Properties, u njemu nađemo Identity Specification koji kada proširimo nalazimo opciju is Identity i nju postavljamo na Yes. Takođe potrebno je postaviti Identity Increment koji određuje za koliko se povećava vrednost sledećeg unesenog ključa i Indentity Seed koja definiše od koje vrednosti počinju da se čuvaju ključevi

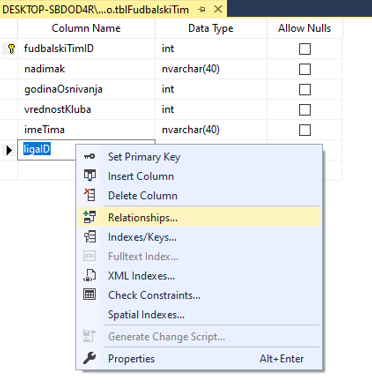


Slika 5.9 Podešavanje Column Properties

Nakon definisanja primarnih ključeva nephodno je definisati, u tabelama u kojima se oni javljaju, stane ključeve i time omogućiti povezanost tabela.

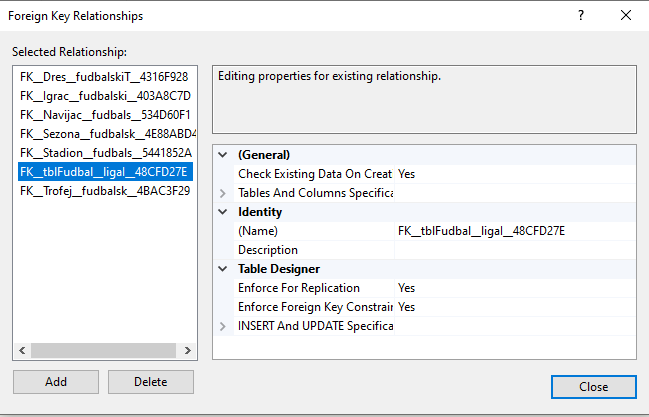
U tabeli tblFudbalskiTim javlja se samo jedan strani ključ. Strani ključ ligaID (koji se u tabeli Liga javlja kao primerni ključ) se može povezati na sledeći način:

Desni klik na strani ključ u okviru tabele tblFudbalskiTim , zatim klik na Relationships.



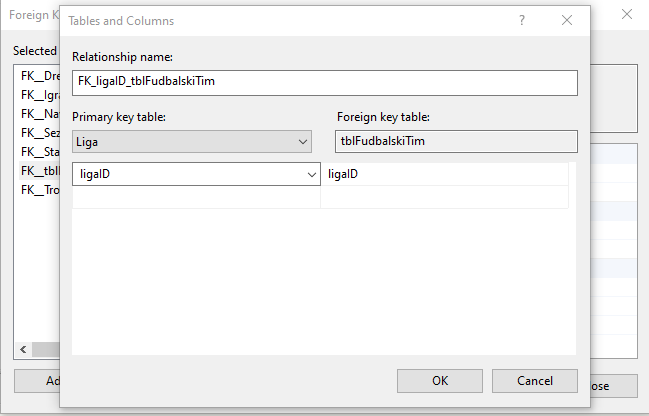
Slika 5.10 Postavljanje veza između ključeva

Posle klika izlazi prozor u okviru kog je potrebno kliknuti dugme add nakon čega se doda taj relationship, zatim ga treba preimenovati u FK\_tbl\_FudbalskiTim\_dbo.Liga.



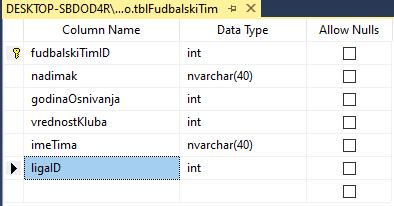
Slika 5.11 Dodavanje veze

Posle ovog koraka potrebno je proširiti Tables and Columns Specification i klikom na ... otvara se novi prozor. U okviru novog prozora treba postaviti vrednosti na sledeći način prikazan na slici 5.12. na kojoj je prikazano postavljanje ključa ligaID koji se u okviru dbo.Liga predstavlja kao primarni ključ a u tbl\_Namestaj ga postavljamo da je u toj tabeli strani ključ.

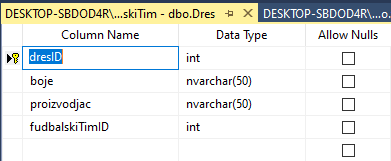


5.13 Dodavanje stanog ključa

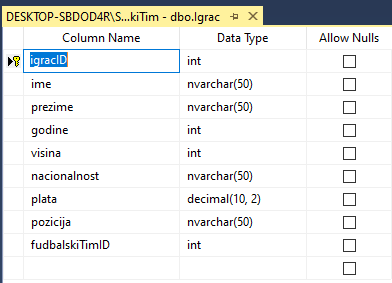
Za bazu FudbalskiTim kreirane su sledeće tabele:



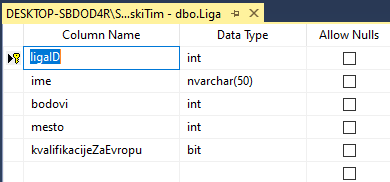
Slika 5.14 tabela FudbalskiTim



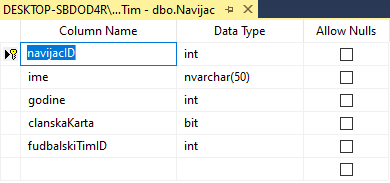
Slika 5.15 tabela Dres



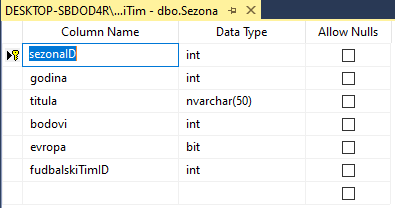
Slika 5.16 tabela Igrac



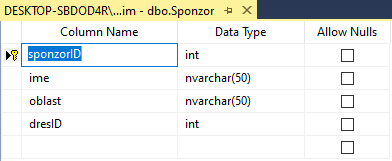
Slika 5.17 tabela Liga



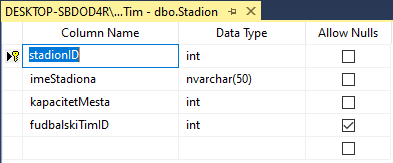
Slika 5.18 tabela Navijac



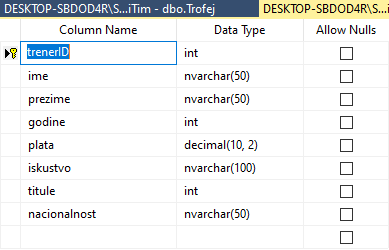
Slika 5.19 Slika tabela Sezona



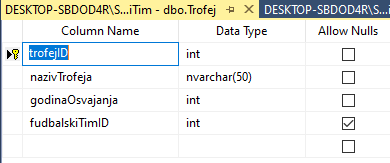
Slika 5.20 tabela Sponzor



Slika 5.21 tabela Stadion



Slika 5.22 tabela Trener



Slika 5.23 tabela Trofej

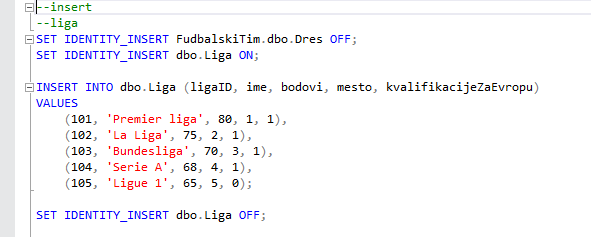
5.2 Upiti I DLM naredbe:

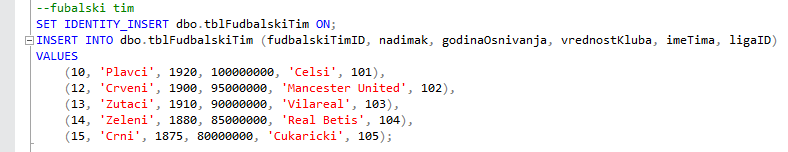
Upiti su zahtevi upućeni bazi podataka. Oni se pišu u Query-u.

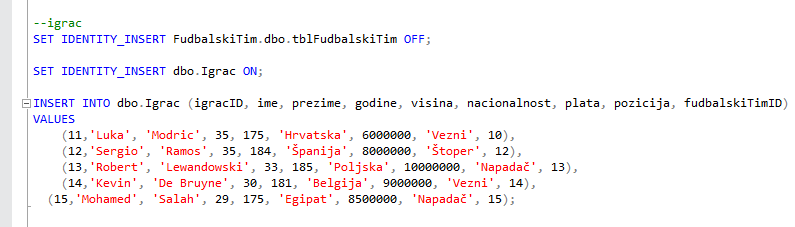
U sql postoje četiri osnovne naredbe:

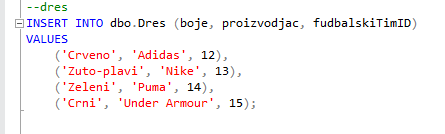
1. *Insert* - služi za ubacivanje podataka u bazu.
2. *Update* - menja unešene podatke.
3. *Select* - služi za prikazivanje podataka iz baze.
4. *Delete* - briše podatke iz baze.

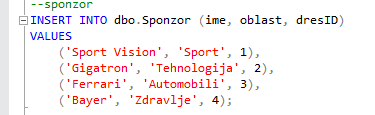
**Insert** naredba za elemente u svakoj tabeli:

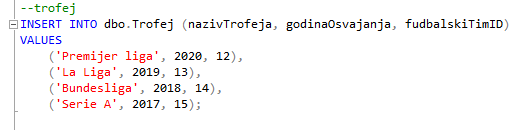


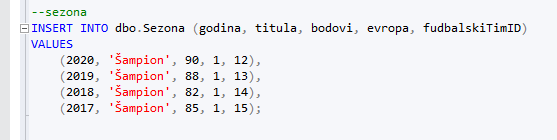


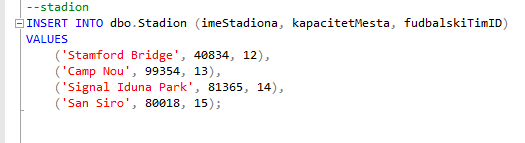


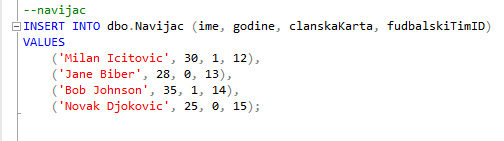




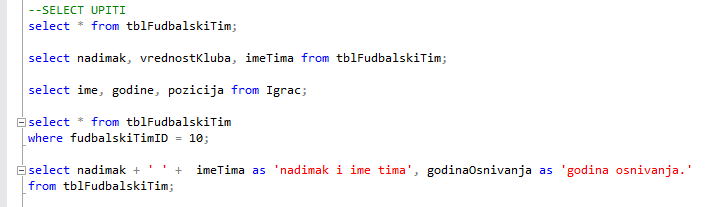




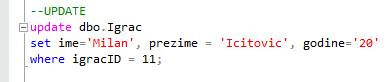




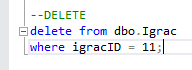
**Select** upiti (omogućava prikaz podataka neke tabele):



**Update** upiti (menja vrednosti postojućih podataka na nekom id-ju):



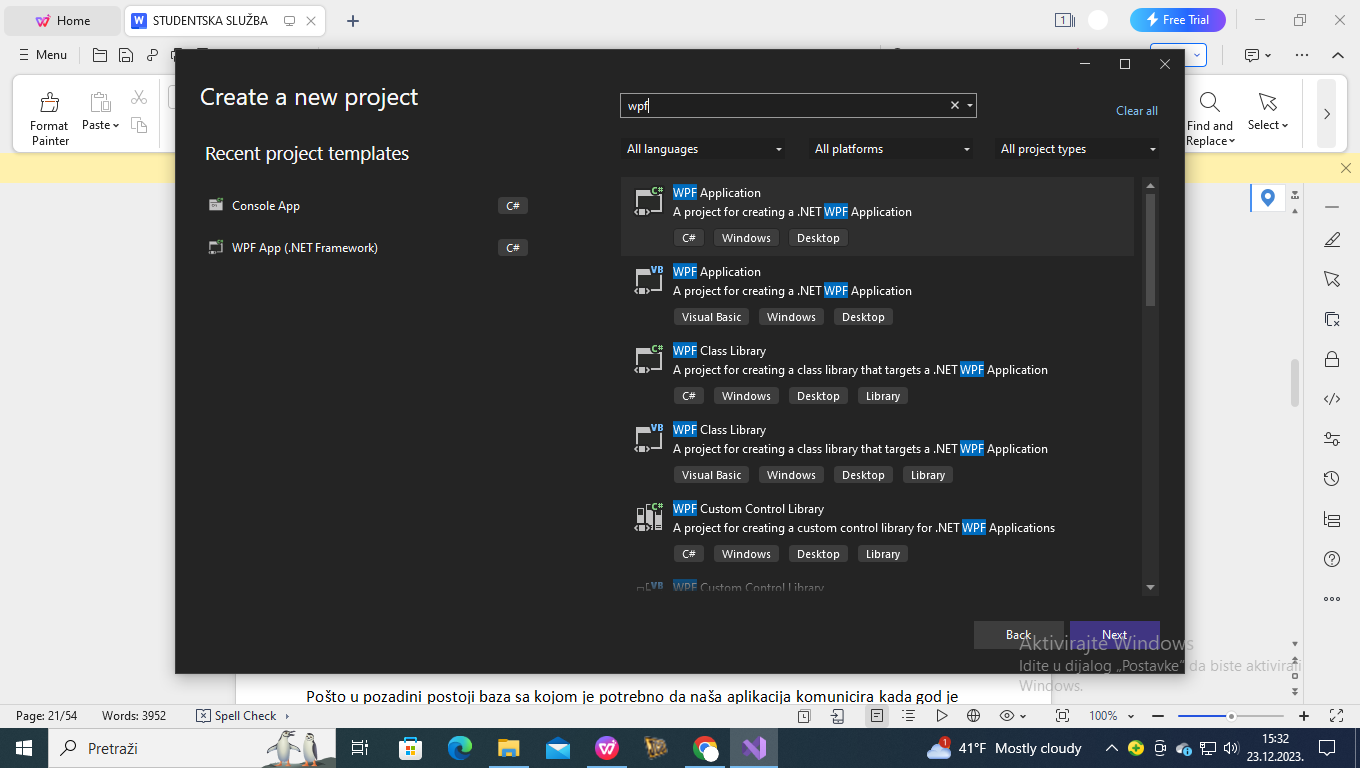
**Delete** upiti (omogućava brisanje podataka iz baze):



Delete naredba neće biti uvek moguća. Do problema dolazi kada postoje podaci koji su povezani između tabela, onsono kada pokušamo da obrišemo direktno neki element koji u drugoj tabeli postoji kao strani ključ.

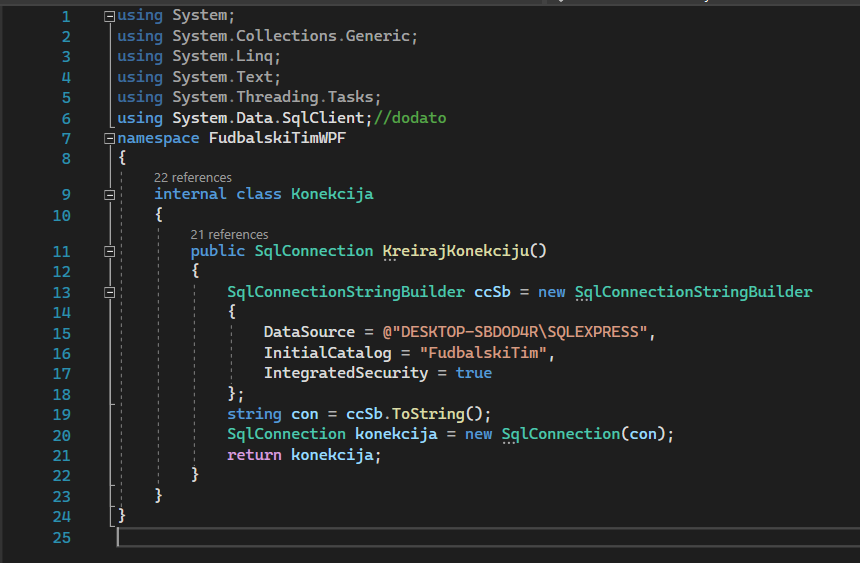
**6) Kreiranje WPF aplikacije**

U okviru ravojnog okruženja Visual Studio kreiramo novi projekat, i odabiramo da taj projekat treba da bude WPF Application i dajemo mu naziv (U ovom projektu to će biti SalonNamestaja).



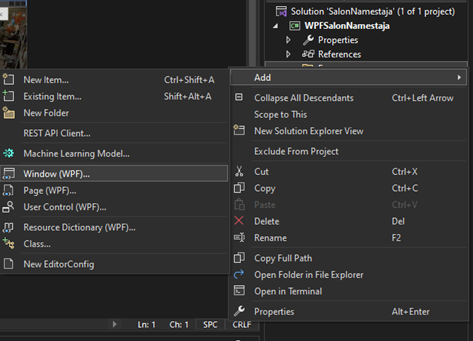
Slika 6.1 Kreiranje WPF Aplikacije

Prvi korak u izradi aplikacije je da se kreira klasa Konekcija u okviru WPFSlonNamestaja. Ta klasa omogućava komunikaciju izmeđy aplikacije i baze podataka



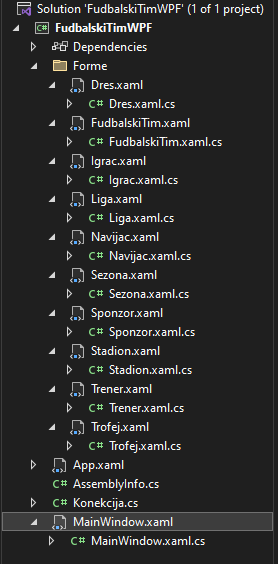
Slika 6.2 Klasa Konekcija

Nakon kreirane konekcije kreiraju se forme za svaku tabelu iz baze i forme se smeštaju u folder pod nazivom forme. Forma se kreira desnim klikom na folder forme, nakon toga se pređe mišem preko Add i odabira opcija Window (WPF).



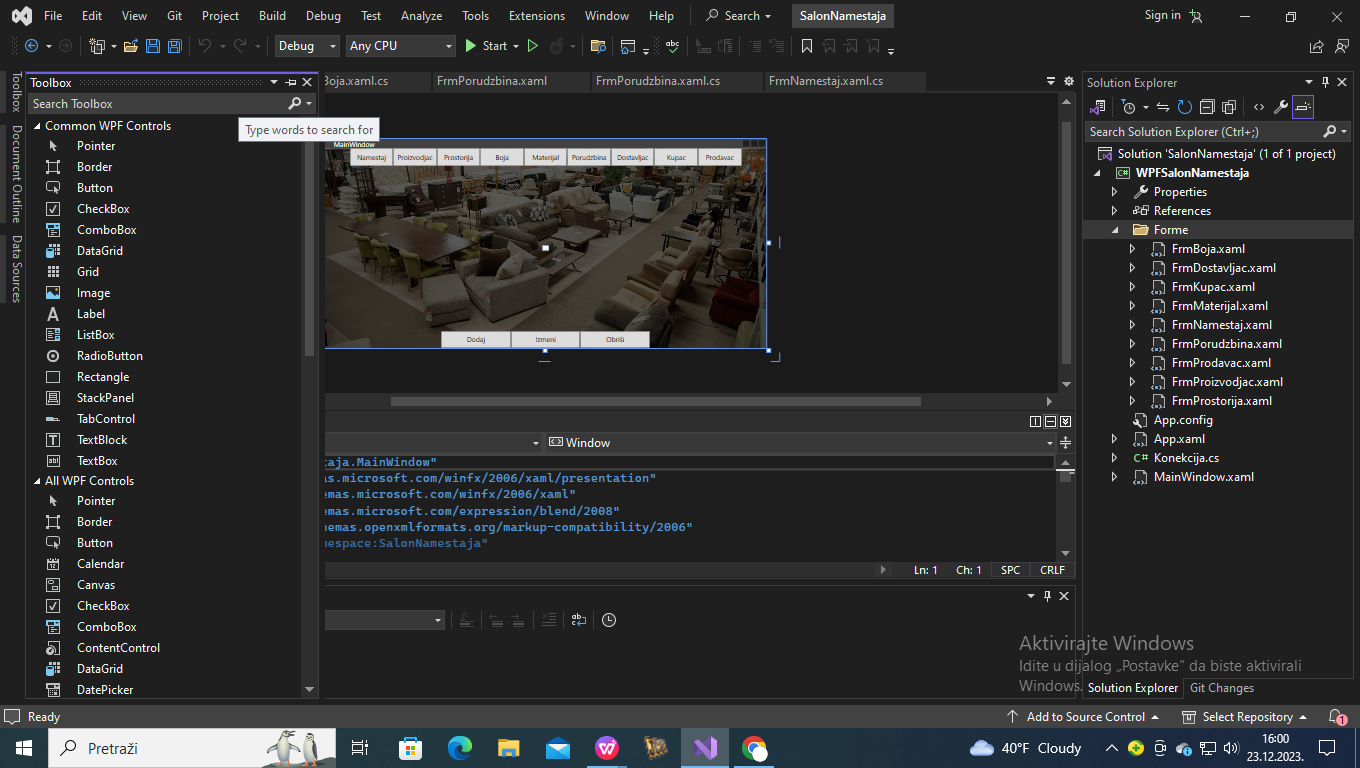
Slika 6.3 Dodavanje forme u projekat

Za aplikaciju FudbalskiTim potrebno je kreirati deset formi za svaku od deset tabela iz baze: Dres, FudbalskiTim, Igrac, Liga, Navijac, Sezona, Sponzor, Stadion, Trener I Trofej.



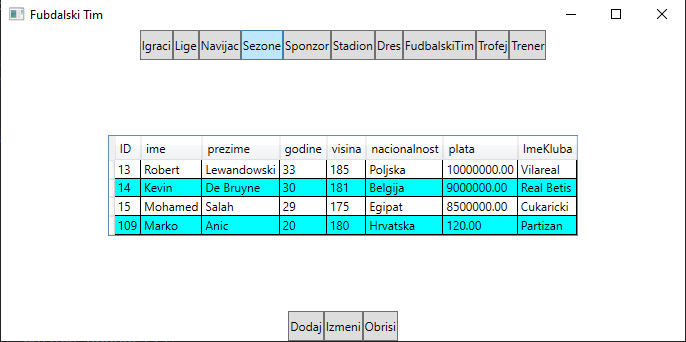
Slika 6.4 Sve potrebne forme za aplikaciju

Komponente koje su korišćene prilikom pravljenja aplikacije za salon nameštaja su: TextBox, CheckBox, Button, ComboBox, Label. Za svaku komponentu moguće je odrediti tačno na kojem mestu u formi će se nalaziti.



Slika 6.5 Komponente u toolbox-u

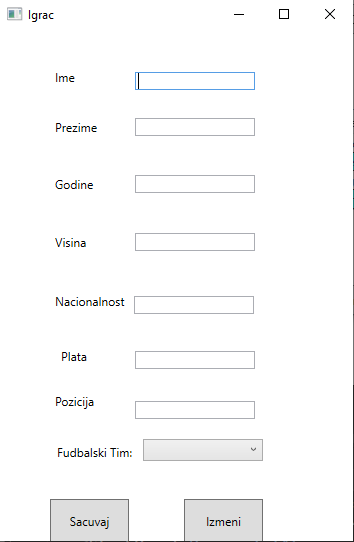
Kada pokrenemo aplikaciju prvi prozor koji će se otvoriti je MainWindow za koji pravimo izgled isto kao i za sve ostale forme



Slika 6.6 Izgled MainWindowa

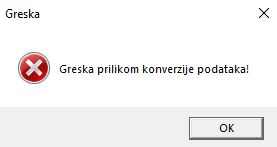
U okviru prvog Stack Panela vidimo dugmadi koji reprezentuju svaku od naših tabela. Prilikom klika na neku od njih, na našem Data Gridu koji se nalazi u centru prozora, će se prikazati informacije iz kliknute tabele (U ovom primeru je izabrana tabela bila Igrac).

Donji stack panel nam služi kako bi izvršavali željenje aktivnosti nad informacijama za, u ovom primeru igrače. Prilikom klika na dugme Dodaj prikazaće se sledeći prozor:

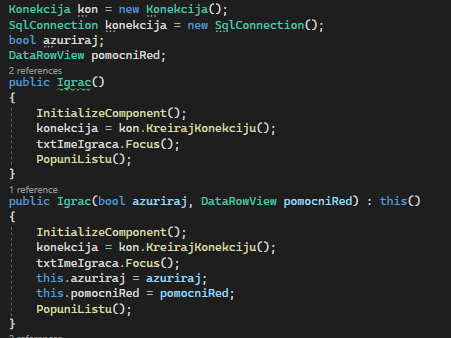


Slika 6.7 Dodaj prozor

U okviru ove forme vidimo sva obeležja koja tabela Igrac sadrži. Na klikom Izmeni se izlazi iz ovog prozora. Prilikom klika na sačuvaj a da se u tom trenutku neka od polja nisu popunila, desiće greška prilikom konverzije podataka, jer vrednosti koje se očekuju da se ubace u tabelu u našoj bazi, nisu ubacila.

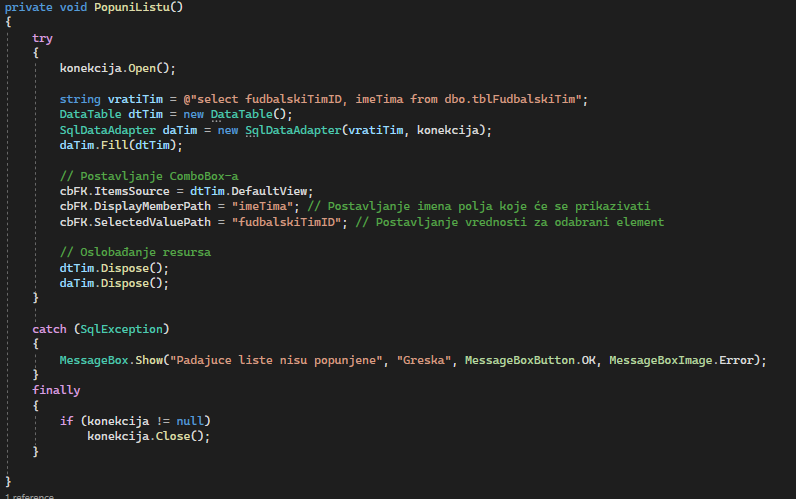


Da bi kod funkcionisao, potrebno je prvo dodati konekciju sa našom bazom.

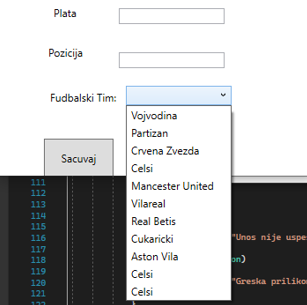


6.8 Konstruktor klase Igrac

Metoda PopuniListu nam služi kako bi se u ComboBox-u prikazale vrednosti Iz željene tabele. Konektujemo se sa našom bazom, izvršimo upit u kom ćemo pokupiti u ovom primeru ID I imeTima, zatim upisati izvučene podatke u comboBox I prikazati ih. Dodajemo Exception koji će da javi da padajuća lista nije popunjena.

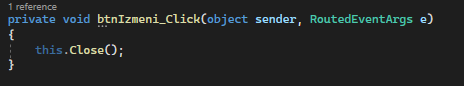


Nakon implementacije metode I poziva iste u oba konstruktora, dobijamo sledeći rezultat:



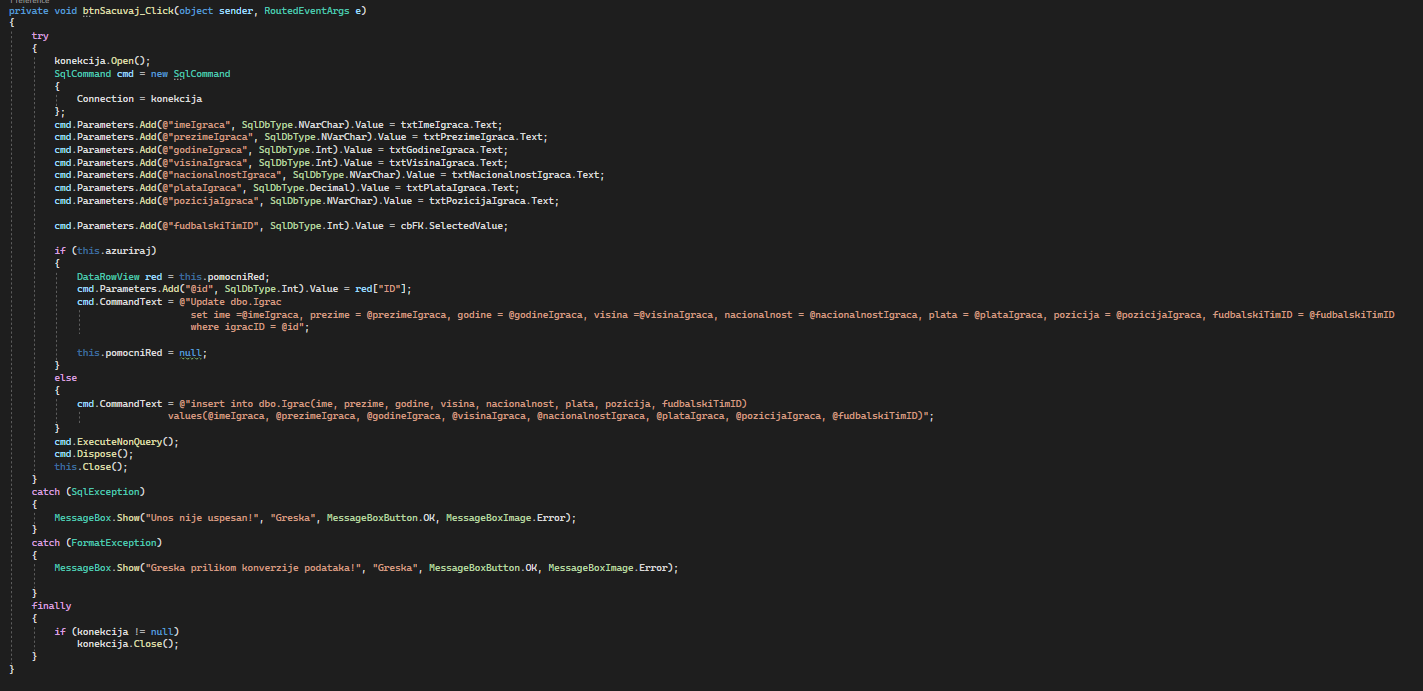
Slika 6.9 Padajuća lista sa popunjenim elementima

Funkcija dugmeta izmeni jeste da se naš prozor ugasi.



Slika 6.10 Izmeni metoda

Što se tiče metoda Sačuvaj, prvo ulazimo u try catch blok, gde se povezujemo sa bazom. Zatim uzimamo vrednosti iz naših textbox-ova I comboBox-ova. Preko njih ćemo ažurirati našu bazu sa upitom Update. Imamo dva Exceptiona, prvi je SqlException koji nam govori da upis nije dobar, a drugi, FormatException, da je došlo do greške prilikom konverzije podataka.

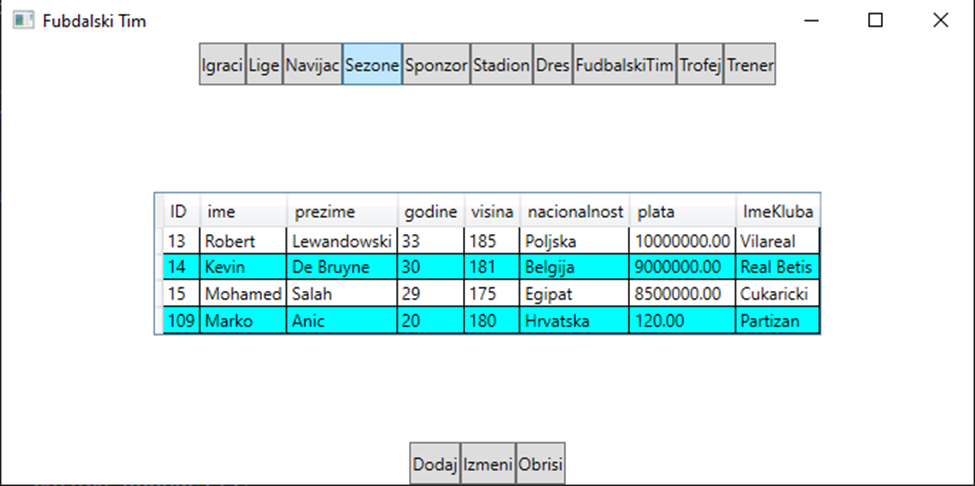


Slika 6.11 Sačuvaj metoda

**7) Testiranje**

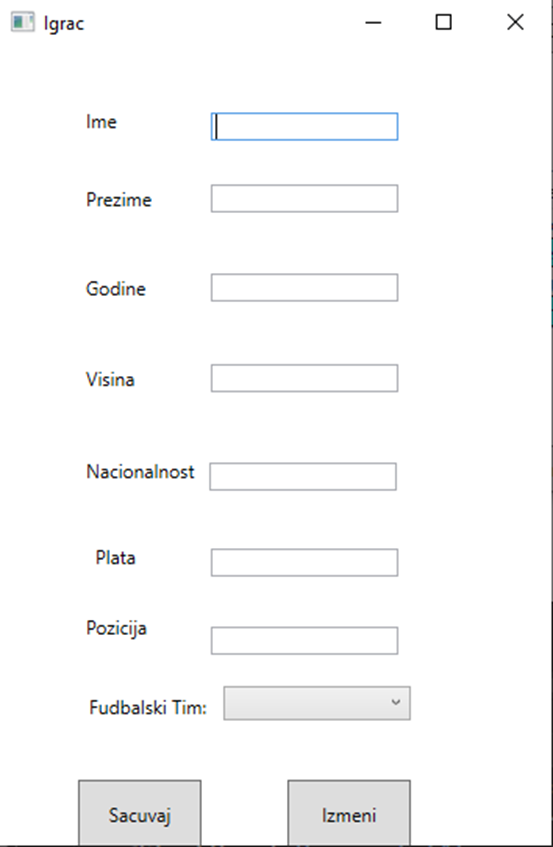
Nakon kreiranja svih potrebnih funkcionalnosti neophodno je izršiti testiranje aplikacije.

Testiranje učitavanja tabela:



Slika 7.1 Učitavanje tabela

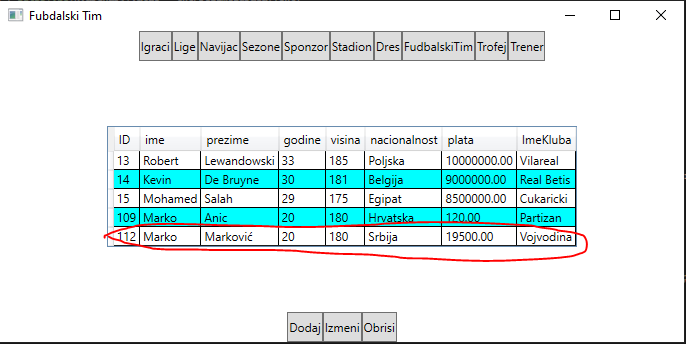
Testiranje dugmeta za dodavanja:



Slika 7.2 Prozor prilikom klika Dodaj

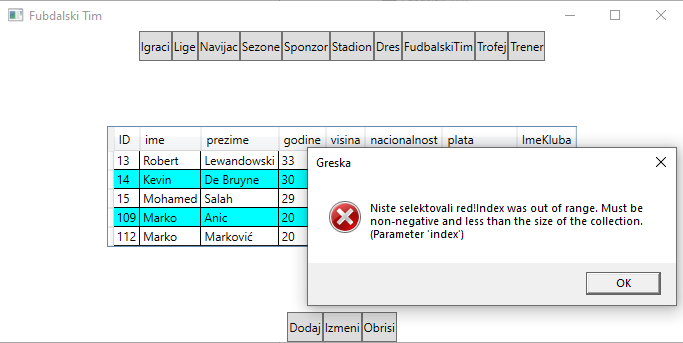
Nakon unetog sadržaja u polja i klika na dugme sačuvaj (čime je i dugme sačuvaj testirano i radi), podaci se uspešno dodaju u bazu podataka, i tabela na glavnom ekranu se ažurira.

Potrebno je isprobati i dugme otkaži koje zatvara otvoreni prozor bez promena.



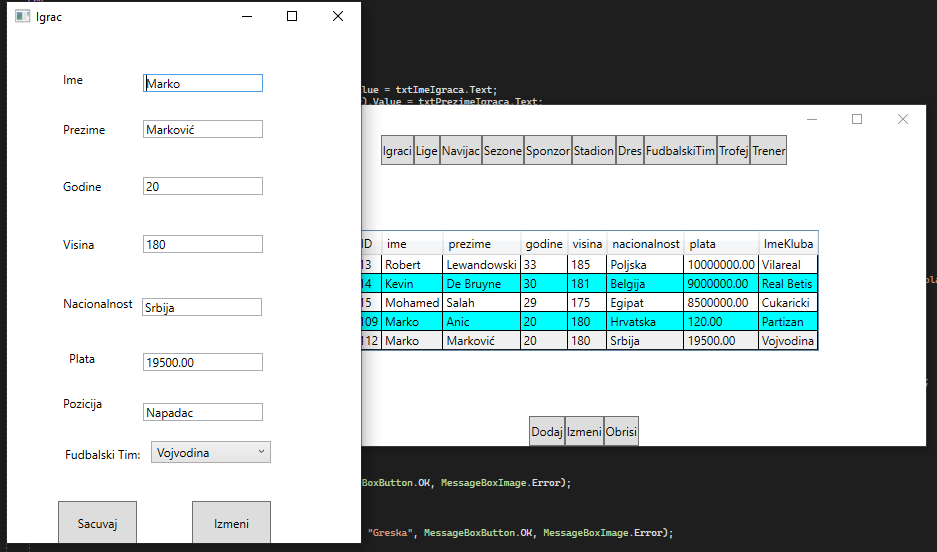
Slika 7.3 Uspešno dodat Igrač

Ukoliko korisnik nije selektovao red za izmenu iskače prozor sa obaveštenjem.



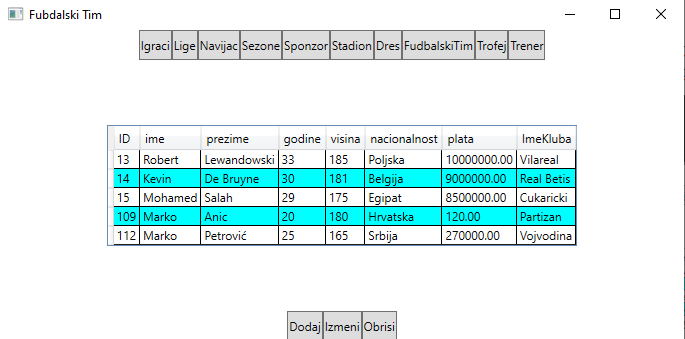
Slika 7.4 Greška ukoliko korisnik nije selektovao red

Nakon selekcije reda i klika na dugme izmeni otvara se prozor u kome su polja popunjena sa podacima iz reda koji je selektovan za izmenu.



Slika 7.5 Prozor nakon selektovanja igrača I klika na dugme Izmeni

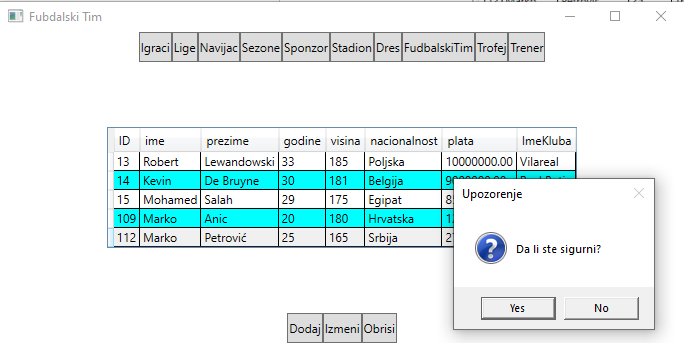
Nakon klika sačuvaj podaci na selektovanom redu se uspešno menjaju i tabela se ažurira.



Slika 7.6 Nakon izmena igrača

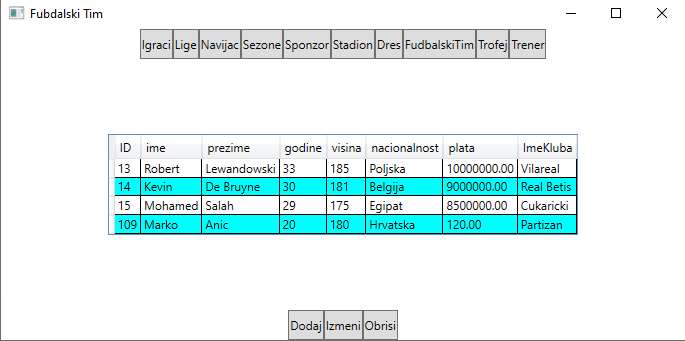
Testiranje dugmeta za brisanje:

Prilikom klika na dugme obrisi otvara se prozor sa pitanjem ukoliko je korisnik siguran da želi da obirše selektovani red.

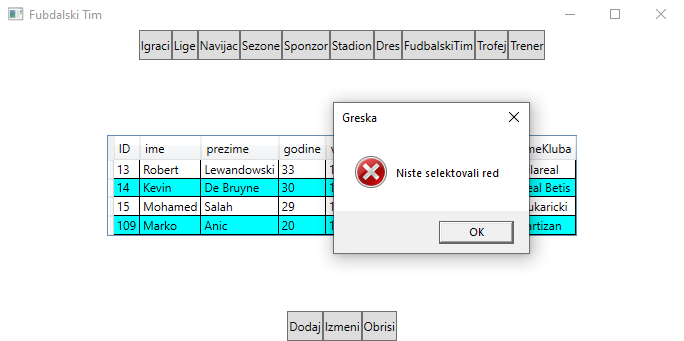


Slika 7.7 Prozor upozorenje tokom brisanja

Nakon klika dugmeta “Da” red iz tabele se brise.



Ukoliko korisnik nije selektovao red izlazi prozor sa obaveštenjem da red nije selektovan.



Slika 7.8 Korisnik nije selektovao red

Ovaj postupak testiranja treba primeniti na svaku od kreiranih tabela kako ne bi došlo do nepredviđenih grešaka u nekoj od njih.

Ukoliko sve tabele i sva dugmad rade, aplikacija je uspešno završena.

**8) Zaključak**

U ovom dokumentu opisan je proces razvoja WPF aplikacije za vođenje dokumentacije informacionog sistema fudbalskog kluba. Podaci su čuvani u SQL server bazi, omogućavajući manipulaciju informacijama o igračima , fudbalskim klubovima, trenerima, navijačima itd.. Ukupno, ova aplikacija omogućava jednostavno praćenje i upravljanje ključnim aspektima fudbalskog tima radi efikasnijeg vođenja poslovanja.