**WPF**

**BIOSKOP**

Student: Nina Stojić, IT19/2022

Predmet: Poslovni informacioni sistemi

Fakultet tehničkih nauka Novi Sad

Opis realnog sistema:

Realni sistem za koji je kreiran ovaj projekat je bioskop. To je mesto u koje ljude dolaze da gledaju film. Vrši se evidencija radnika, kupaca, filmova, hrane i pića, transakcija, karata, sala i žanrova. Radnik može da dodaje, menja i briše podatke o svemu prethodno navedenom.

Zadatak:

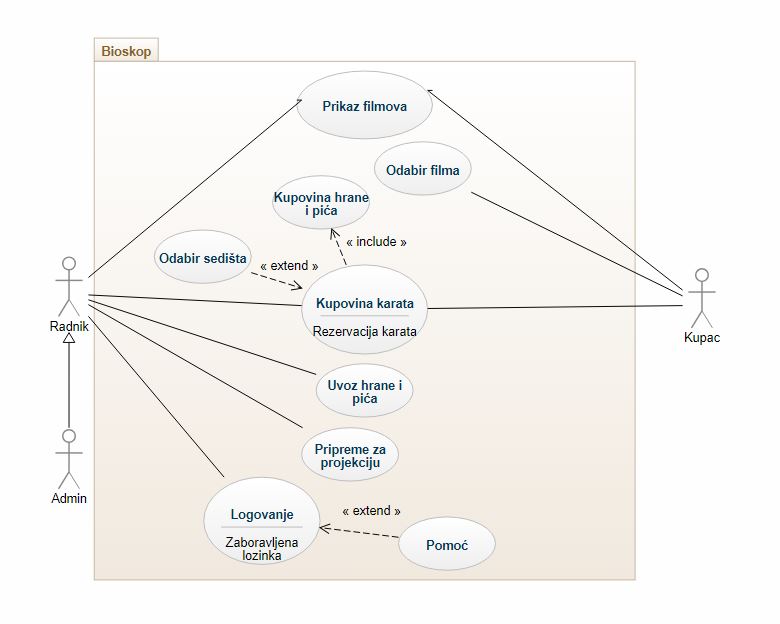
Zadatak je da se kreira aplikacija koja sadrži evidenciju filmova, kupaca, radnika, karata, transakcija, sali, žanrova, hrane i pića. Samo će zaposleni moći da koriste aplikaciju i ona će im pomoći da lakše obavljaju svoj posao.

Opis tehnologija:

Za izradu UML dijagrama korišćen je sajt GenMyModel (genmymodel.com). To je besplatan sajt koji omogućava kreiranje UML dijagrama. Za izradu same aplikacije korišćen je Microsoft VisualStudio IDE, a unutar njega kreirana je WPF aplikacija, dok je kod pisan u C# programskom jeziku. Za čuvanje podataka u bazi korišćen je SQL Server i Microsoft SQL Server Management Studio kao radno okruženje. WPF se nalazi unutar .NET Framework-a. WPF predstavlja grafički podsistem koji se koristi za kreiranje aplikacija sa korisničkim interfejsom baziranom na XAML-u. .NET je okruženje za razvoj softvera razvijano od strane Microsoft-a u okviru kog je omogućeno korišćenje velikog broja programskih jezika od kojih su najpopularniji C#, C++ i VisualBasic.

UML

Dijagram slučajeva upotrebe



Tekstualni opis slučaja upotrebe:

**Slučaj upotrebe:** Priprema za projekciju

**Kratak opis:** Priprema sale

**Učesnici:** Radnik

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Ljudi su kupili/rezervisali karte za projekciju

**Opis:** Čišćenje sale i sedišta. Priprema projektora za puštanje filma

**Izuzeci:** /

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** /

**Slučaj upotrebe**: Uvoz hrane i pića

**Kratak opis:** Nabavka hrane i pića po potrebi

**Učesnici:** Radnik

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Provera zaliha

**Opis:** Radnik kupuje određenu količinu hrane i pića

**Izuzeci: /**

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Namirnice su nabavljene

**Slučaj upotrebe:** Prikaz filmova

**Kratak opis:** Kupac gleda određeni film

**Učesnici:** Radnik, kupac

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Radnik je pripremio salu

**Opis:** Radnik je pustio film u sali kako bi kupci mogli da ga gledaju

**Izuzeci:** /

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja: /**

**Slučaj upotrebe:** Odabir filma

**Kratak opis:** Kupac bira film za gledanje

**Učesnici**: Kupac

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Postoje filmovi za gledanje

**Opis**: Kupac online ili uživo bira film koji će gledati

**Izuzeci: /**

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja: /**

**Slučaj upotrebe:** Kupovina karata

**Kratak opis:** Kupac kupuje kartu

**Učesnici:** Radnik, kupac

**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** Dostupne su karte za izabrani film

**Opis:** Radnik prodaje kupcu kartu za izabrani film

**Izuzeci:** Kupac je rezervisao kartu i želi da odabere sedište

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** Kupac je dobio željeni proizvod

**Slučaj upotrebe:** Logovanje

**Kratak opis:** Kupac rezerviše kartu

**Učesnici:** Radnik

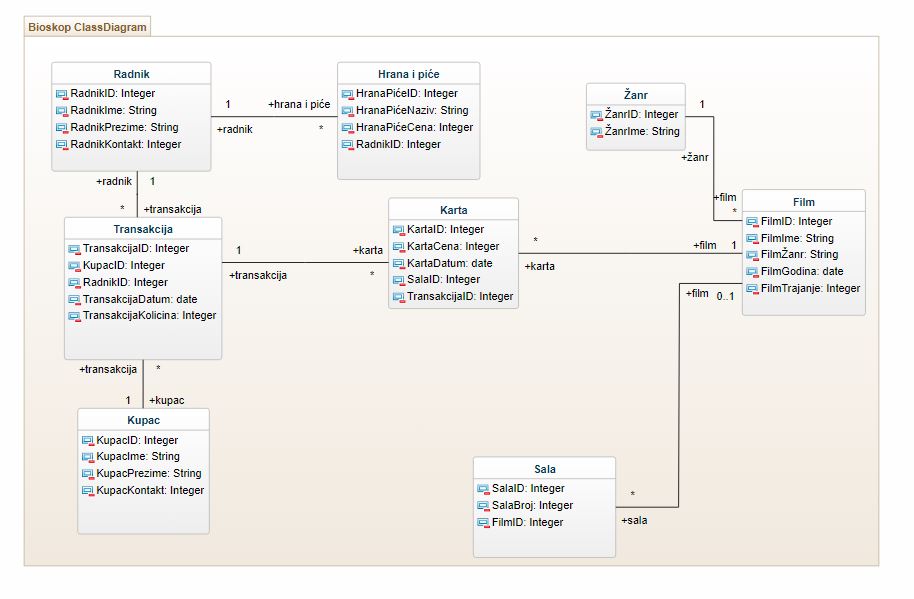
**Uslovi koji moraju biti zadovoljeni pre izvršavanja:** /

**Opis:** Zaposleni se loguje na svoj profil

**Izuzeci:** Problemi pri logovanju, npr. Zabravljena lozinka - ide se na pomoc

**Uslovi koji moraju biti zadovoljeni nakon izvršavanja:** /

Dijagram klasa



**Klase:**

1. Film

Obeležja klase film su:

* FilmID (integer) – jedinstveni broj koji predstavlja određeni film i to bi bio primarni ključ
* FilmIme (string)
* FilmŽanr (string)
* FilmGodina (date) – godina izlaska filma
* FilmTrajanje (integer)

1. Karta

Obeležja klase karta su:

* KartaID (integer) - jedinstveni broj koji predstavlja određenu kartu i to bi bio primarni ključ
* KartaCena (integer)
* KartaDatum(date) – datum projekcije
* SalaID (integer) - strani ključ
* TransakcijaID (integer) – strani ključ

1. Sala

Obeležja klase sala su:

* SalaID (integer) - jedinstveni broj koji predstavlja određenu salu i to bi bio primarni ključ
* SalaBroj (integer) – broj sale u kojoj se održava projekcija
* FilmID (integer) - strani ključ

1. Žanr

Obeležja klase žanr su:

* ŽanrID (integer) - jedinstveni broj koji predstavlja određeni žanr filma i to bi bio primarni ključ
* ŽanrIme (string)

1. Hrana i piće

Obeležja klase hrana i piće su:

* HranaPiceID (integer) - jedinstveni broj koji predstavlja određenu hranu i piće i to bi bio primarni ključ
* HranaPiceNaziv (string)
* HranaPiceCena (integer)
* RadnikID (integer) - strani ključ

1. Radnik

Obeležja klase radnik su:

* RadnikID (integer) - jedinstveni broj koji predstavlja određenog radnika i to bi bio primarni ključ
* RadnikIme (string)
* RadnikPrezime (string)
* RadnikKontakt (integer) – broj telefona radnika

1. Kupac

Obeležja klase kupac su:

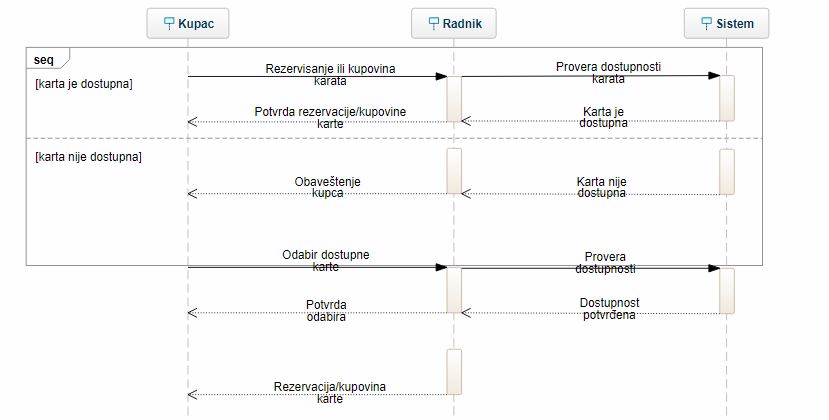
* KupacID (integer) - jedinstveni broj koji predstavlja određenog kupca i to bi bio primarni ključ
* KupacIme (string)
* KupacPrezime (string)
* KupacKontakt (integer) – broj telefona kupca

1. Transakcija

Obeležja klase kupac su:

* TransakcijaID ( integer) - jedinstveni broj koji predstavlja određenu transakciju i to bi bio primarni ključ
* KupacID (integer) - strani ključ
* RadnikID (integer) - strani ključ
* TransakcijaDatum (date) – datum izvršenja transakcije
* TransakcijaKoličina (integer) – količina kaupljenih karata

Dijagram sekvence:

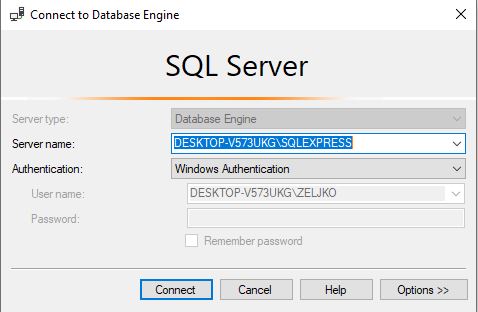


Ovo je dijagram sekvence za kupovinu karata. Ovaj dijagram je urađen na osnovu dijagrama slučaja upotrebe. U ovom procesu učestvuju kupac, radnik i aplikacija. Proces počinje tako što kupac želi da rezerviše ili kupi kartu, zatim zaposleni proverava dostupnost karte. Ako je karta dostupna potvrđuje se rezervacija/kupovina karte. Ako karta nije dostupna, radnik obavesti kupca koji zatim bira dostupnu kartu nakon čega dolazi do rezervacije/kupovine karte.

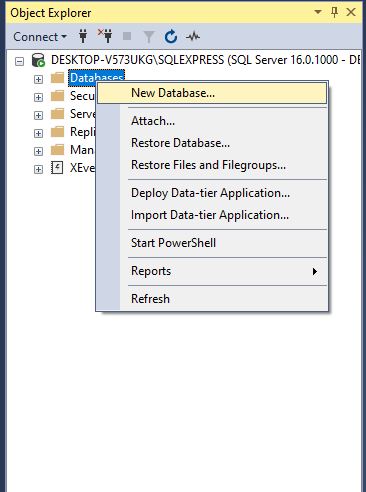
**Baza podataka**

Kreiranje baze podataka:

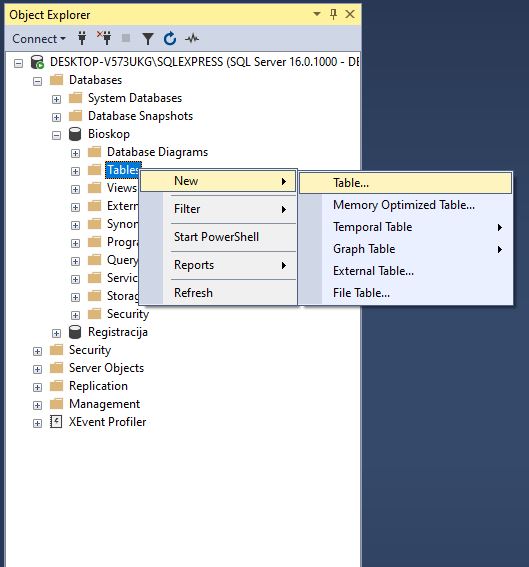
Nakon kreiranja UML dijagrama potrebno je da se kreira baza podataka. Baza se kreira preko Microsoft SQL Server Management Studio. Prvo se uspostavi veza sa serverom tako sto se klikne connect za server name kompjutera.

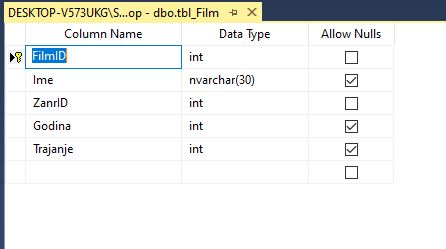


Sledeci korak je kreiranje baze, desnim klikom na Databases/New Database

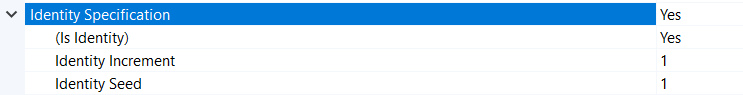


Zatim se prave tabele po uzoru na class diagram, proširivši Databases i novokreiranu bazu desnim klikom na Tables/New/Table



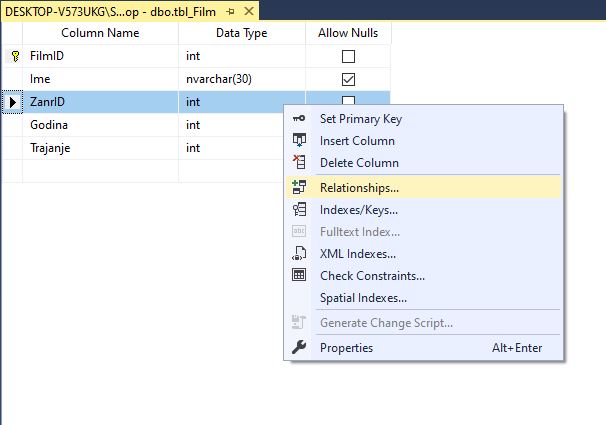


U tabeli je potrebno uneti sve potrebne podatke i polja sa tim da se primaran ključ date tabele piše na početku i dodaje mu se Identity Specification.

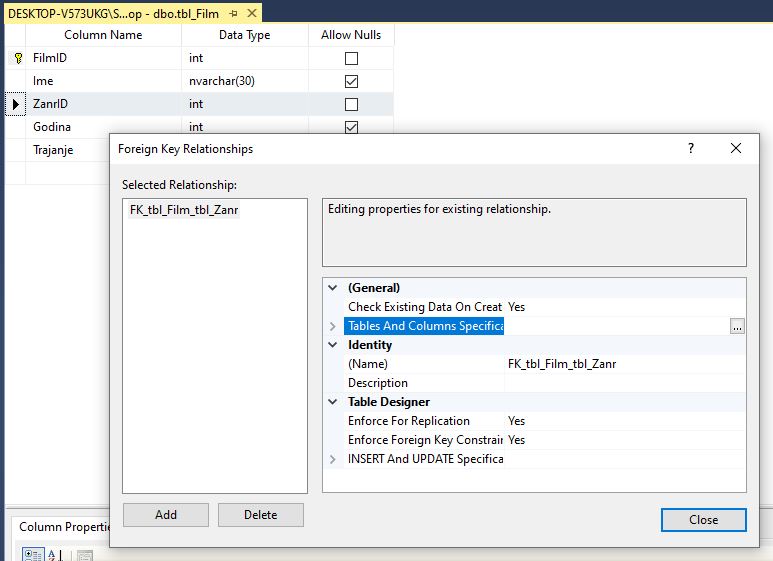


Identity increment i identity seed se postavljaju na 1

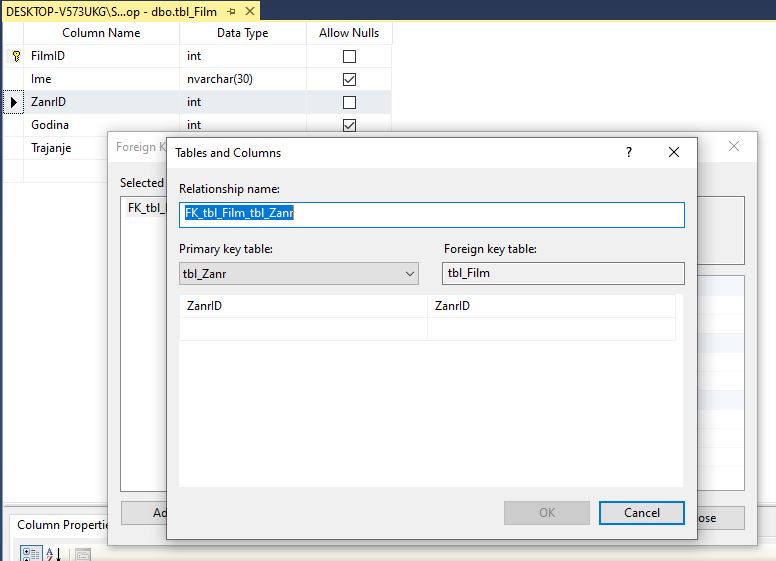
Nakon toga se postavljaju veze između tabela ukoliko je to potrebno preko stranih ključeva. To se radi tako što se desnim klikom stisne na red i izabere se Relationships.



Kada se stisne na Relationships otvara se prozor gde se stisne Add, nakon toga treba stisnuti tri tačke kod Tables and Columns specificaction

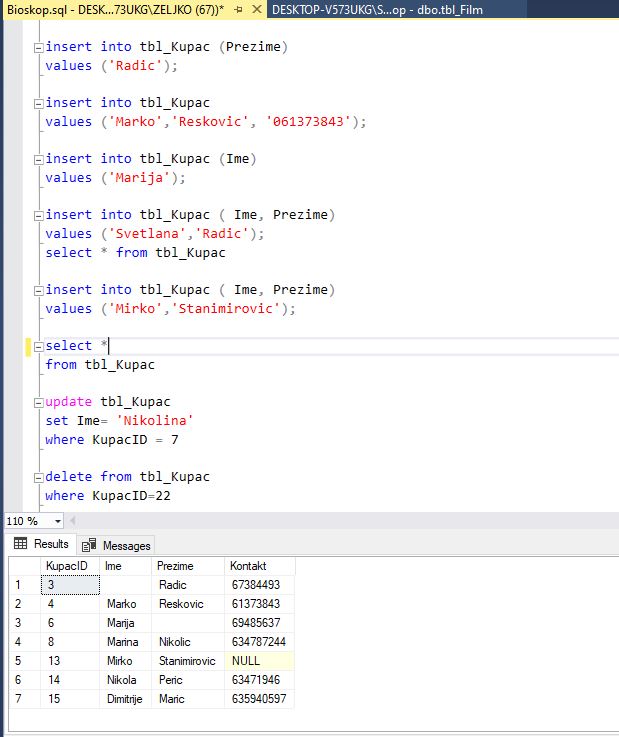


Nakon toga se otvara prozor gde odaberemo tabelu i polje primarnog ključa kao i tabelu i polja stranog ključa i na taj način ih povezati.



Upiti i DML naredbe

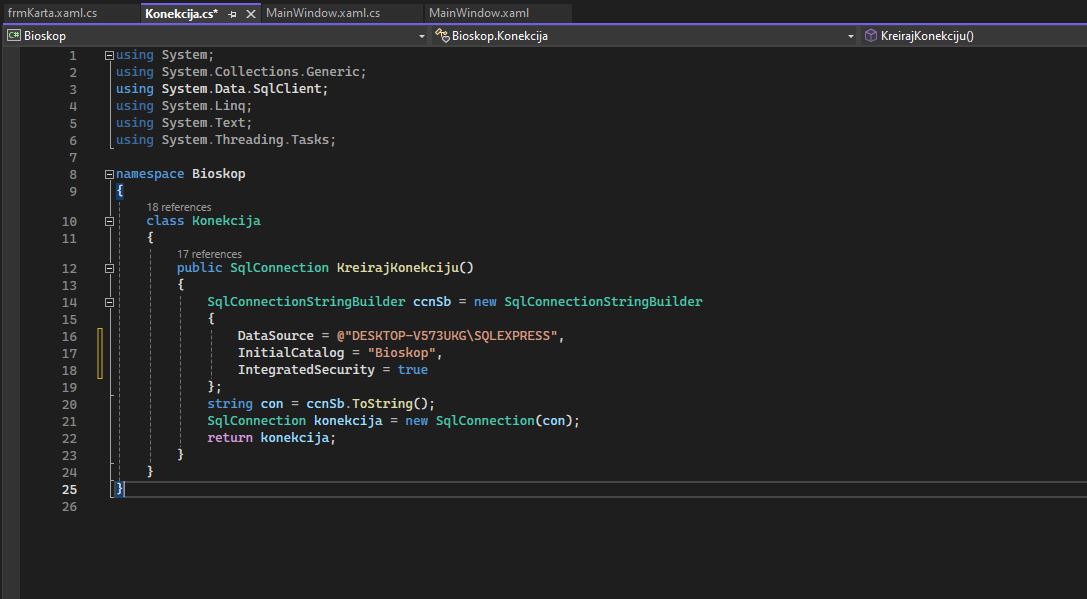
Nakon kreiranja baze radimo par Sql query upita radi dodavanja podataka u tabele i provere ispravnosti tabela.



Ovde vidimo primer komande dodavanja, brisanja, izmene i prikaza podataka u tabeli kupac, gde je trenutno izvršena komanda select.

Kreiranje WPF aplikacije

Konekcija



Metoda KreirajKonekciju() se nalazi u klasi Konekcija i sluzi da kreira konekciju. Prvo se pravi objekat tipa SqlConnectionStringBuilder kome se kao DataSource šalje adresa servera, a kao InitialCatalog naziv baze. Kreira se string koji će imati vrednost StringBuildera. Nakon toga se pravi objekat konekcija tipa Sqlconnection kome se u konstruktoru prosleđuje kreirani string. Na kraju se vraca kreirana konekcija.

MainWindow



Odmah nakon pokretanja aplikacije otvara se MainWindow forma, pritiskom gornjih dugmića mogu da se vide već postojeće tabele. Klikom na dodaj ili izmeni otvoriće nam se odgovarajuće forme.

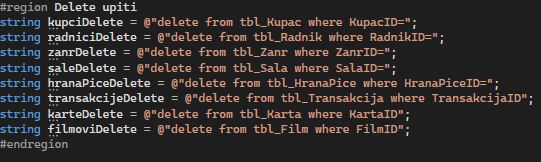
Obriši

Kada izaberemo red u tabeli, možemo da pritisnemo dugme obriši, nakon čega će se izabrani red ukloniti.



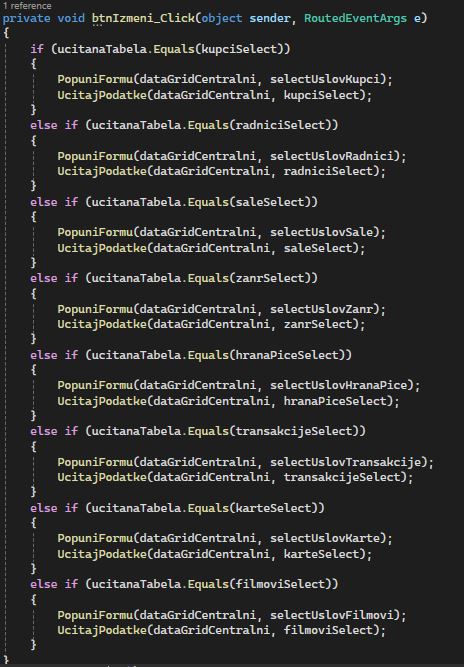
Logika brisanja

Primer delete upita



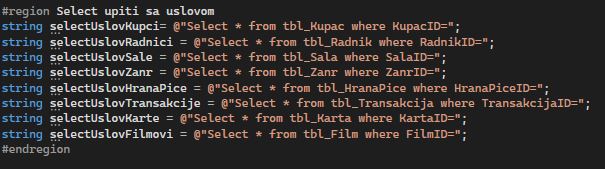
Izmeni

Ona vrši izmenu postojećih podataka u bazi tako što ih unosi u bazu i prikazuje nam te izmenjene parametre. Sama logika se zasniva na proveravanju koja nam je tabela prikazana u tom trentutku kako bi aplikacija znala koji frame treba da nam otvori pošto selektujemo parametar za izmenu.





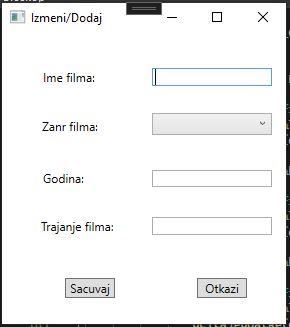
Primer select upita za izmeni



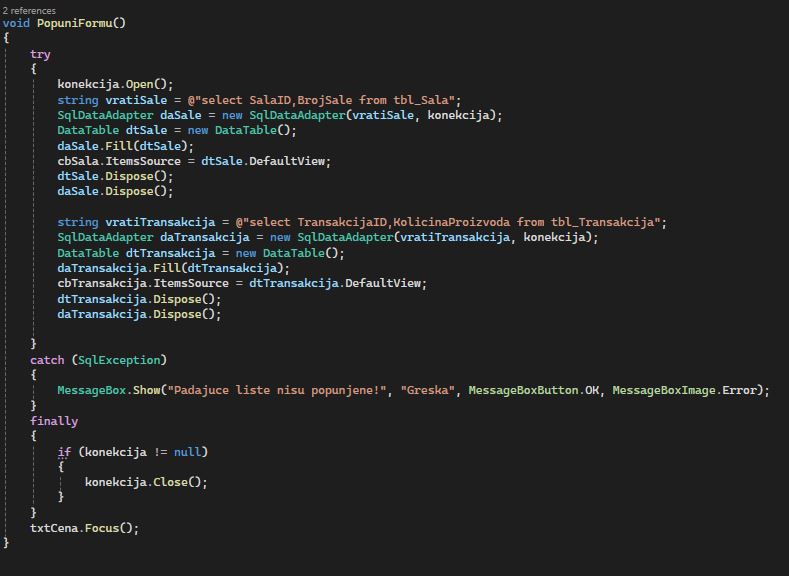
Dodaj



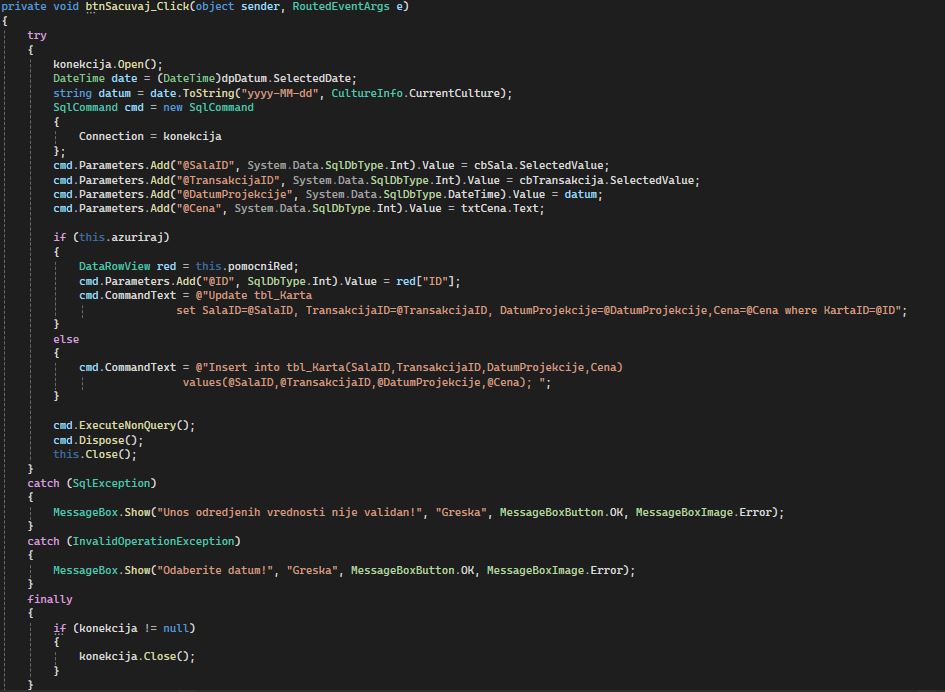
Dodavanje je bitno da bi mogli da iz aplikacije unosimo nove podatke u našu bazu. U zavisnosti od toga na koje smo dugme kliknuli a potom otišli na dodaj u okviru MainWindow-a otvoriće nam se određena forma.



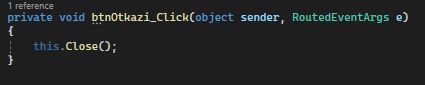
U okviru ovih frame-ova se takođe ubacuje funkcionalnost dugmića Sačuvaj I Otkaži, u okviru xaml.cs frame-ova se ubacuje i to kako će oni da dodaju podatke u bazu i tu je potrebno odraditi logiku za strane ključeve koji su u stvari ComboBox-ovi.



Logika dugmeta Sačuvaj



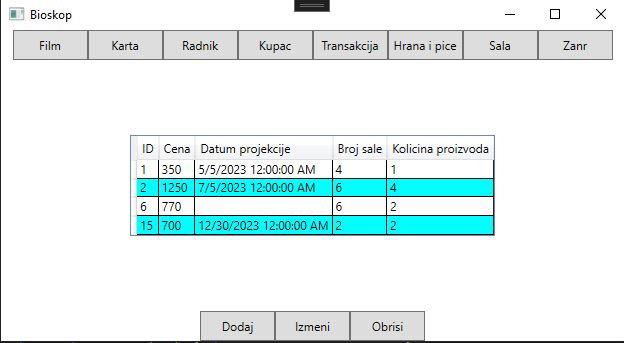
Logika dugmeta Otkaži



Izgled aplikacije

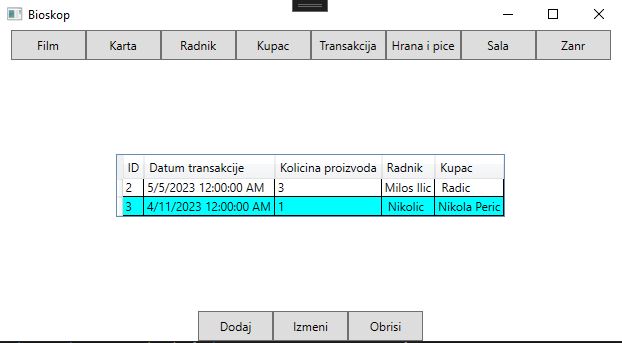
Dalje su prikazani izgledi svih tabela sa podacima, kada se klikne na određeno dugme.



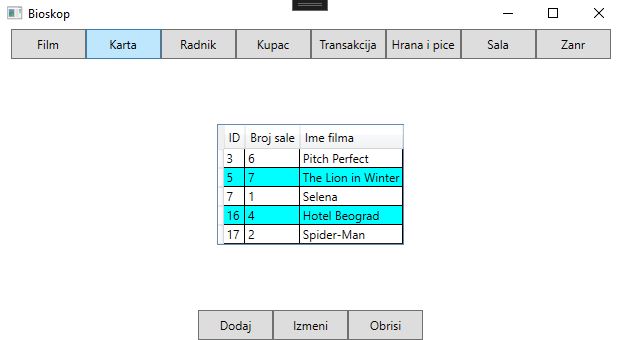






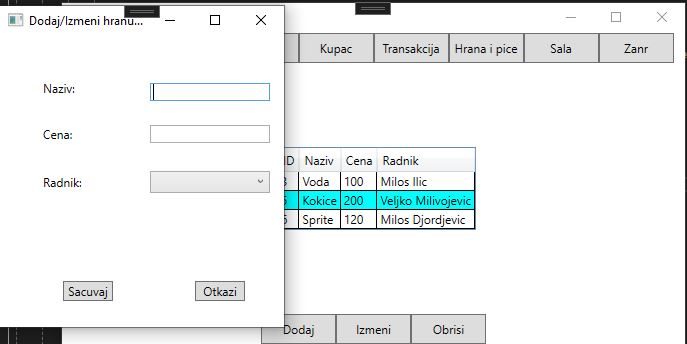




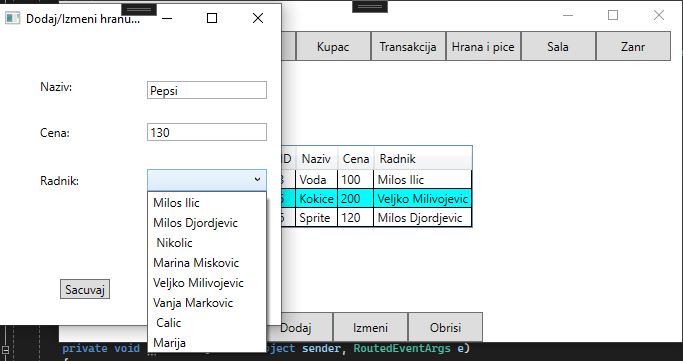




Testiranje aplikacije



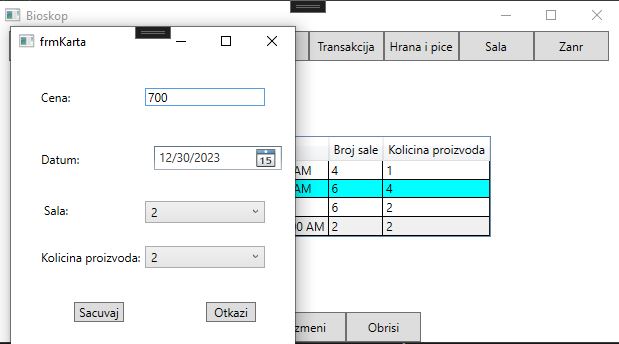
Klik na dugme dodaj



Nakon klika na dugme sačuvaj



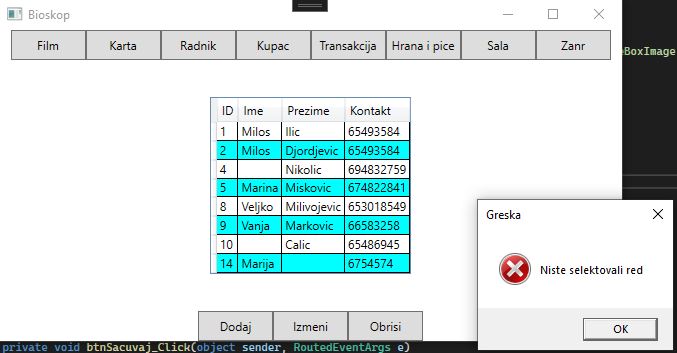
Kada izaberemo red koji želimo da promenimo i kliknemo na dugme izmeni



Kada izaberemo red i pritisnemo dugme Obriši



Ovo se desi ukoliko ne selektujemo red kada želimo da menjamo ili brišemo nešto.



Zaključak

Tokom kreiranja aplikacije stečena su znanja iz Uml dijagrama, Sql-a i kreiranja Wpf aplikacije. Najtezi deo je definitivno bilo kreiranje same aplikacije, dok se Sql velikom većinom zasnivao na Uml dijagramima. Pravljenje same aplikacije je bilo dosta zanimljivo, kao i korisno, te svakako mi deluje kao nešto na čemu ću još raditi kako bih unapredila kako sebe tako i samu aplikaciju. Na kraju, aplikacija je uspešno kreirana i testirana.