

FAKULTET TEHNIČKIH NAUKA
UNIVERZITET U NOVOM SADU

WPF PROJEKAT
POZORIŠTE

ŠKORIĆ VANJA / IT28-2022

Asistenti: Andela Todorić, Dušan Krstić

2023/24

Sadržaj

Sadržaj.....	2
1. Opis realnog sistema	3
2. Zadatak.....	3
3. Opis tehnologija	3
3.1 Microsoft Structured Query Language (SQL).....	3
3.2 Microsoft Visual Studio IDE.....	4
3.3 Microsoft ActiveX Data Objects (ADO).NET.....	4
3.4 Windows Presentation Foundation (WPF).....	4
4. UML.....	5
4.1 Dijagram slučajeva upotrebe	5
4.2 Dijagram klasa.....	8
4.3 Dijagram sekvenci	9
5. Baza podataka	10
5.1 Kreiranje baze podataka	10
5.2 Upiti i DML naredbe	20
6. Kreiranje WPF aplikacije.....	21
7. Testiranje aplikacije	51
8. Zaključak.....	55

1. Opis realnog sistema

U ovom projektu će biti detaljno opisan i razrađen sistem osnovnih operacija kroz koje prolazi jedan gledalac koji želi da poseti pozorište. Pre samog odlaska u pozorište gledaoci često imaju želju da se informišu o pozorišnom repertoaru, vide raspored izvedbi, rezervišu ulaznice itd... U ovom projektu su uzete u obzir osnovne opcije informativnog karaktera kao što su recimo raspored izvedbi predstava, informacije o glumcima, informacije o pozorištima, ulaznicama... Ali, i neke dodatne opcije kao što su recimo informacije o zaposlenima i o rediteljima predstava.

2. Zadatak

Cilj zadatka je da se napravi informacioni sistem koji će predstavljati podršku poslovanju pozorišta. U prvom delu zadatka vrši se modelovanje realnog sistema pomoću UML dijagrama koji na adekvatan način reprezentuju posmatrani sistem ili neki njegov deo. Zatim se, na osnovu tako modelovanog sistema, kreira WPF aplikacija koja simulira rad pozorišta. Komunikacija aplikacije sa bazom podataka ostvarivaće se putem ADO.NET tehnologije. Studenti treba da upravljuju celokupnim procesom razvoja WPF aplikacije, od modelovanja sistema (u okviru kojeg treba da kreiraju dijagram slučajeva upotrebe i dijagram klase), do razvoja i testiranja WPF aplikacije koja predstavlja informacioni sistem za podršku poslovanja pozorišta. Informacioni sistem treba da pruži mogućnost prikazivanja podataka o predstavama, njihovim rediteljima, glumcima, izvedbama, gledaocima, ulaznicama i na kraju krajeva o samom pozorištu kao i o njegovim zaposlenima. Takođe, kroz aplikaciju je potrebno omogućiti dodavanje, ažuriranje i brisanje podataka nad klasama od interesa, definisanih dijagramom klase.

3. Opis tehnologija

U ovom poglavlju predstavljene su tehnologije koje će biti korišćene u izradi projektnih zadataka.

3.1 Microsoft Structured Query Language (SQL)

Microsoft SQL Server je sistem za upravljanje bazom podataka. Može se koristiti u različite svrhe kao što su poslovna inteligencija (*eng. Business Intelligence*) ili za skladištenje podataka (*eng. Data Storage*). Ipak, SQL Server se najčešće koristi kao krajnja komponenta za smeštanje podataka, a tradicionalne *client-server* aplikacije i *web-based* aplikacije često koriste SQL Server za smeštanje aplikativnih podataka. Kada aplikacija postavlja upit, modifikuje i dodaje podatke u bazu podataka, ona koristi *Structured Query Language (SQL)*. SQL je standardni jezik koji se koristi za komunikaciju sa bazama podataka. *Reporting Services* je karakteristika SQL servera koja se koristi za automatsko generisanje izveštaja iz SQL server baze podataka. Neke aplikacije zahtevaju da *Reporting Services* bude instaliran da bi se postigla potpuna funkcionalnost. Kada je SQL server instaliran, uobičajeno postoji jedna instanca. Ova instanca nema ime i pristupa joj se koristeći ime servera. Unutar svake instance može postojati više baza podataka. Svaka aplikacija će imati svoju bazu podataka na SQL serveru, ali one mogu da budu u istoj instanci.

3.2 Microsoft Visual Studio IDE

Microsoft Visual Studio je programsko okruženje kreirano za izradu računarskih programa za Microsoft Windows. Takođe se koristi i za izradu veb-sajtova kao i veb i mobilnih aplikacija. Visual Studio koristi platforme kao što su: *Windows API*, *Windows Forms*, *Windows Presentation Foundation*, *Windows Store* i *Microsoft Silverlight*. Visual Studio sadrži deo za pisanje izvornog koda programa sa integrisanim *debugger-om*. Druge alatke su *code profiler*, *forms designer* za izradu aplikacija, *web designer* aplikaciju i druge. Pored alatki i osnovnih funkcionalnosti, moguće je instalirati i dodatke za poboljšanje funkcionalnosti. Visual Studio podržava trideset i šest različitih programskih jezika. Ugrađeni programski jezici su C, C++ i C++/CLI VB.NET, C#, F# i TypeScript. Druge jezike, kao što su Python, Ruby, Node.js i M, je moguće dodati. Podržava i XML/XSLT, HTML/XHTML, JavaScript i CSS.

3.3 Microsoft ActiveX Data Objects (ADO).NET

ADO.NET je skup klasa za rad sa podacima. Snabdevači podataka su klase koje obezbeđuju mogućnost konektovanja na izvor podataka. Osnovni izvori podataka su SQL Server izvor podataka i OLE DB izvor podataka. U *ADO.NET-u* postoje dva osnovna načina rada: konektovani i diskonektovani. Kod konektovanog scenarija resursi se uzimaju sa servera sve dok se konekcija ne zatvori. Korisnik je konstantno povezan na izvor podataka. Prednost konektovanog scenarija je da su podaci uvek ažurni i da se konkurentni pristup resursima servera lakše kontroliše. Nedostatak konektovanog rada je obavezno postojanje konstantne mrežne konekcija, kao i mala skalabilnost aplikacije.

3.4 Windows Presentation Foundation (WPF)

Windows Presentation WPF je grafički podsistem (deo *.NET platforme*) koji služi za prikazivanje korisničkog interfejsa u aplikacijama koje su zasnovane na Windows operativnom sistemu. Ovaj grafički podsistem daje mogućnost izrade desktop aplikacija ili veb aplikacija.

Glavne komponente arhitekture WPF-a su:

- *Presentation Framework*,
- *Presentation Core* i
- *Milcore*.

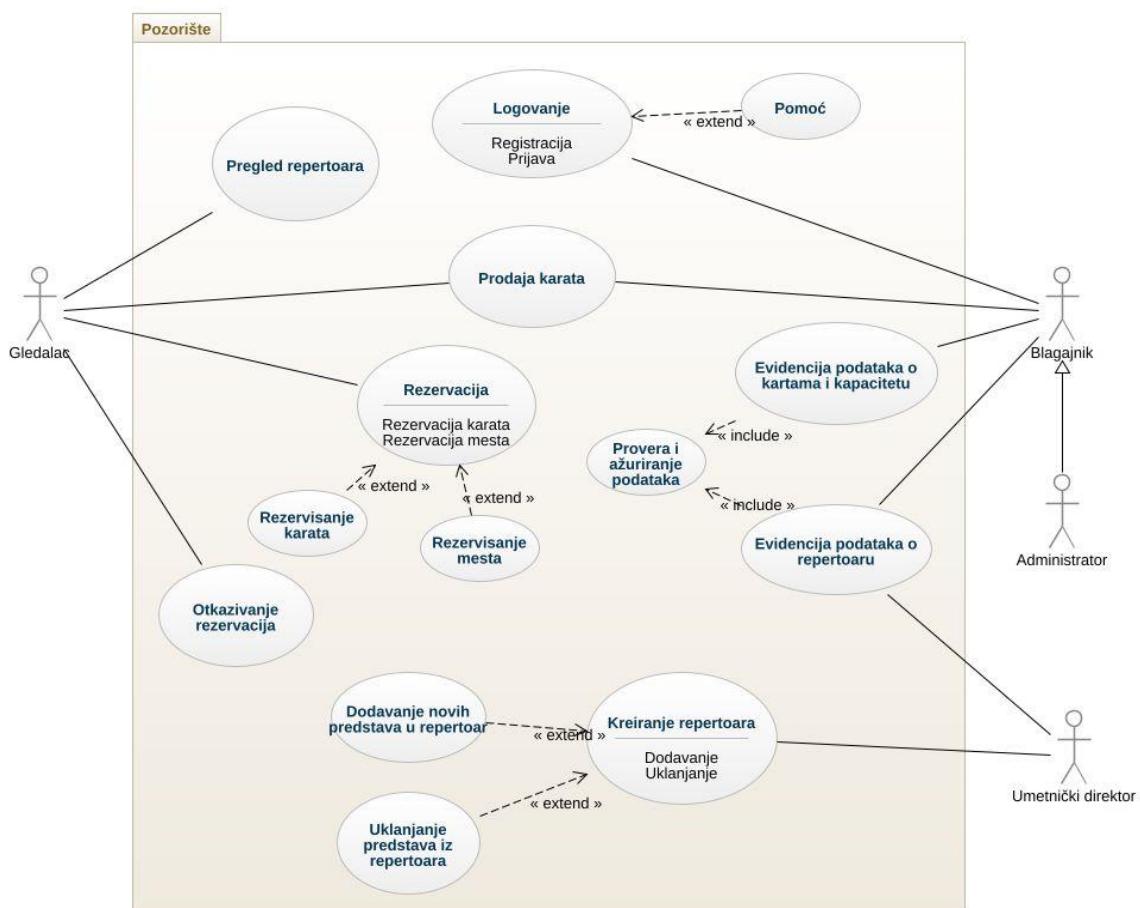
Presentation Framework i *Presentation Core* su napisani u upravljačkom kodu, a *Milcore* je deo koda koji omogućava integraciju sa *DirectX* koji je odgovoran za prikazivanje elemenata. *Common Language Runtime - CLR* omogućava upravljanje memorijom i upravljanje greškama. WPF omogućava konzistentan programski model koji razdvaja korisnički interfejs od XAML, koji je poslovne logike. Koristi *Extensible Application Markup Language* baziran na *Extensible Markup Language - XML*, da bi definisao i povezao elemente interfejsa. Prednost koju donosi XAML je to što je taj jezik potpuno deklarativan i omogućava programeru da opiše ponašanje i integraciju komponenti bez proceduralnog programiranja. Elementi se mogu skalirati bez gubitaka u kvalitetu i pikselizaciji što omogućava veću pristupačnost.

4. UML

UML dijagrami korišćeni za modelovanje predstavljenog realnog sistema (pozorišta) biće opisani u nastavku teksta.

4.1. Dijagram slučajeva upotrebe

Na samom početku pristupa se inicijalnom modelovanju sistema, odnosno kreiranju dijagrama slučajeva upotrebe na osnovu datog opisa za primer poslovanja pozorišta. Ovaj dijagram nam opisuje kako korisnici (gledaoci) interaguju sa sistemom (pozorištem) radi ostvarivanja određenih funkcionalnosti (gleđanja pozorišnog repertoara). Dijagram slučajeva upotreba prikazan je na slici 4.1.



Slika 4.1 - Dijagram slučajeva upotrebe za informacioni sistem pozorišta

1) Slučaj upotrebe: Logovanje

Kratak opis: Logovanje u sistem pozorišta.

Učesnici: Blagajnik

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Blagajnik poseduje nalog u sistemu, i zna potrebne podatke za logovanje u sistem.

Opis: Sistem prikazuje blagajniku formu za prijavu u sistem. Ona se sastoji iz dva dela, prvi deo je za unos lozinke, a drugi deo je za unos korisničkog imena. Kada blagajnik unese svoje podatke vrši se provera da li uneti podaci postoje, kao i da li su dobro uneti podaci (ukoliko nalog postoji). [Izuzetak : Pogrešno uneta lozinka ili korisničko ime]. Nakon što je izvršena provera podataka i utvrđeno da su oni tačni blagajnik može da pristupi funkcionalnostima informacionog sistema.

Izuzeci: [Izuzetak : Pogrešno uneta lozinka ili korisničko ime]: U slučaju da blagajnik pogrešno unese lozinku ili korisnički nalog, sistem obaveštava blagajnik putem ispisa poruke na ekranu.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: Blagajnik je u mogućnosti da koristi sve podatke iz sistema.

2) Slučaj upotrebe: Pregled repertoara

Kratak opis: Pregled trenutno-aktuelnog repertoara.

Učesnici: Gledalac

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Gledalac mora da ima instaliranu aplikaciju pozorišta (ukoliko želi da pristupi repertoaru putem aplikacije), dok u svim drugim slučajevima ne postoji uslov koji mora biti zadovoljen da bi gledalac pristupio pozorišnom repertoaru.

Opis: Prilikom odabira sadržaja koji gledalac želi da gleda u pozorištu, neophodno je da prvo pregleda repertoar pozorišta koji je trenutno aktuelan, kako bi znao koji sadržaj je dostupan u tom momentu.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: Gledalac je upoznat sa aktuelnim repertoarom pozorišta.

3) Slučaj upotrebe: Prodaja karata

Kratak opis: Blagajnik prodaje kartu gledaocima.

Učesnici: Blagajnik, gledalac

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Gledalac daje blagajniku novac u visini cene karte.

Opis: Da bi gledalac pristupio pozorišnoj predstavi, on prvo mora da kupi kartu na blagajni. Prodaja karata podrazumeva da gledalac daje blagajniku novac u visini cene karte u zamenu za kartu. [Izuzetak : Onlajn plaćanje].

Izuzeci: [Izuzetak : Onlajn plaćanje]: Ukoliko je gledalac unapred platio kartu (karticom) putem aplikacije pozorišta , onda mu ne treba novac, već samo vrši preuzimanje karte kod blagajnika.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: U blagajni ima količina novca u visini cene prodatih karata, a gledaoci poseduju karte za određenu predstavu.

4) Slučaj upotrebe: Rezervacija

Kratak opis: Gledalac vrši rezervaciju karata ili mesta za određenu predstavu.

Učesnici: Gledalac

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Gledalac mora da ostavi svoje lične podatke kod blagajnika (ukoliko vrši rezervaciju uživo), ili mora da poseduje aplikaciju i nalog na aplikaciji pozorišta (ukoliko vrši onlajn rezervaciju).

Opis: Da bi sebi osigurao željeno mesto u pozorištu ili određeni broj karata dok se nisu rasprodale, gledalac može da izvrši rezervaciju karata ili rezervaciju mesta unapred, uz rok uplate od 72h od izvršavanja rezervacije.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: Gledalac je dobio željeno mesto na predstavi. Gledalac je obezbedio sebi određen broj karata za predstavu.

5) Slučaj upotrebe: Otkazivanje rezervacija

Kratak opis: Gledalac otkazuje unapred napravljenu rezervaciju.

Učesnici: Gledalac

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Gledalac mora da poseduje već napravljenu rezervaciju na svom imenu.

Opis: Ukoliko iz nekog razloga gledalac promeni svoje mišljenje i više ne želi određena mesta u pozorištu ili određen broj karata – on vrši otkazivanje svoje rezervacije i samim tim oslobađa određen broj mesta/karata. [Izuzetak : Iste rezervacije].

Iuzeci: [Izuzetak : Iste rezervacije]: Ukoliko gledalac nije ispoštovao rok uplate od 72h od napravljenе rezervacije – ona ističe i gledalac više nema rezervisana mesta/karte.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: Gledalac više ne poseduje rezervisane karte/mesta u pozorištu na svom imenu.

6) Slučaj upotrebe: Evidencija podataka o kartama i kapacitetu

Kratak opis: Prikupljanje potrebnih podataka o kartama i kapacitetu mesta u pozorištu.

Učesnici: Blagajnik

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Blagajnik je prijavljen u sistem.

Opis: Blagajnik prikuplja i evidentira podatke o kartama i kapacitetu mesta u pozorištu u sistem za evidenciju. On takođe vrši proveru i ažuriranje podataka. [Izuzetak : Pogrešno uneti podaci].

Izuzeci: [Izuzetak : Pogrešno uneti podaci]: Ukoliko su neki podaci pogrešno uneti u sistem (recimo podatak o broju mesta za određenu predstavu) blagajnik je dužan da ih ispravi.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: U sistemu postoje tačni podaci o kartama i kapacitetu mesta u pozorištu, kojima blagajnik može da pristupi.

7) Slučaj upotrebe: Evidencija podataka o repertoaru

Kratak opis: Prikupljanje potrebnih podataka o repertoaru pozorišta.

Učesnici: Blagajnik, umetnički direktor

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Blagajnik i umetnički direktor su prijavljeni u sistem.

Opis: Umetnički direktor prikuplja podatke o trenutnom i starom repertoaru pozorišta, zatim ih analizira, pravi promene u repertoaru u skladu sa analizama i unosi ih u sistem, a blagajnik vrši ažuriranje i proveru podataka u skladu sa promenama koje je izvršio umetnički direktor. [Izuzetak : Pogrešno uneti podaci].

Izuzeci: [Izuzetak : Pogrešno uneti podaci]: Ukoliko su neki podaci o repertoaru pogrešno uneti u sistem, blagajnik je dužan da o tome obavesti umetničkog direktora i da u dogovoru sa njim ispravi pogrešne podatke.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: U sistemu postoje tačni podaci o repertoaru pozorišta, kojima blagajnik i umetnički direktor mogu da pristupe.

8) Slučaj upotrebe: Kreiranje repertoara

Kratak opis: Kreiranje i sastavljanje repertoara pozorišta.

Učesnici: Umetnički direktor

Uslovi koji moraju biti zadovoljeni pre izvršavanja: Umetnički direktor ima pristup aktuelnom repertoaru pozorišta, kao i prethodnim (starijim) repertoarima.

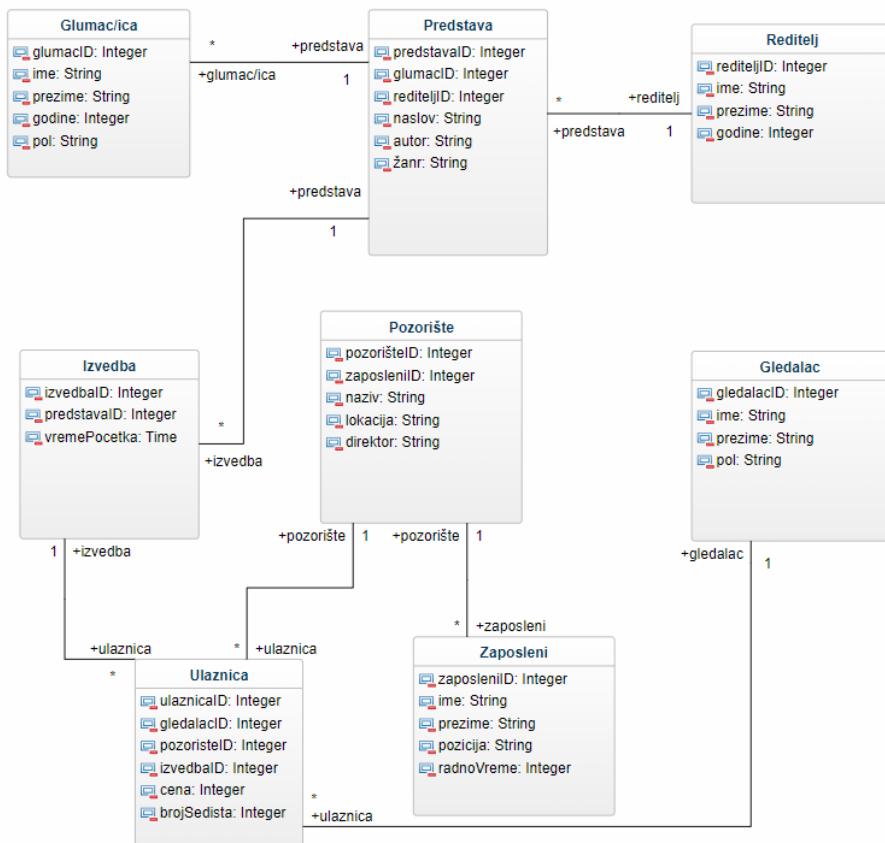
Opis: Umetnički direktor kreira i sastavlja novi repertoar na osnovu istraživanja koje vrši kako bi napravio što bolji repertoar. On može da vrši dodavanje i uklanjanje predstava iz repertoara.

Uslovi koji moraju biti zadovoljeni nakon izvršavanja: Kreiran nov repertoar za pozorište.

4.2.Dijagram klasa

Dijagram klasa je UML dijagram koji sadrži klase, interfejse, pakete i njihove međusobne odnose koji daju logički pregled na ceo softverski sistem ili na neku njegovu funkcionalnost. Svrha dijagrama klasa je da na pojednostavljen način prikaže klase koje se nalaze u jednoj

aplikaciji. Dijagram klasa prikazan je na slici 4.2.

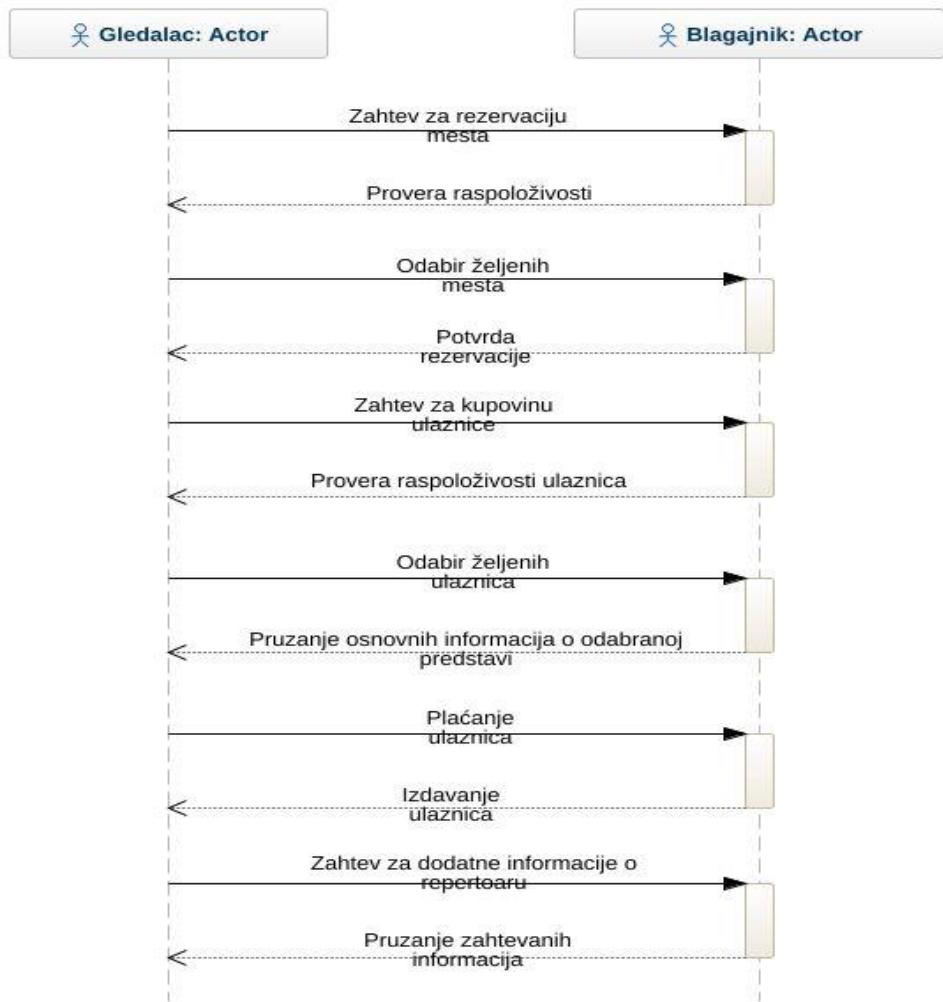


Slika 4.2 - Dijagram klasa za informacioni sistem pozorišta

Identifikovano je osam klasa: pozorište, zaposleni, gledalac, reditelj, predstava, glumac/ica, izvedba i ulaznica. Jedno pozorište može da ima jednog ili više zaposlenih, a zaposleni radi u jednom određenom pozorištu. Isto važi i za gledaoce, ulaznice i predstave (pozorište može da ih ima jednog ili više, dok gledaoci idu u jedno određeno pozorište da gledaju određenu predstavu i kupuju ulaznice za istu). Gledalac može da kupi jednu ili više ulaznica za predstavu, a s druge strane jedna ulaznica je potrebna za jednog gledaoca. Gledalac može da gleda jednu predstavu, dok predstava može da ima jednog ili više gledalaca. Reditelj može da rezira jednu ili više predstava, dok svaka predstava ima svog reditelja. Predstava može da ima jednog ili više glumaca, dok svaki glumac glumi u određenoj predstavi. Predstava poseduje jednu ili više ulaznica za istu, dok je svaka ulaznica određena za jednu konkretnu predstavu. Predstava može da ima jednu ili više izvedbi, dok se svaka izvedba odnosi na konkretno jednu istu predstavu. Za izvedbu neke predstave možemo kupiti jednu ili više ulaznica, a svaka ulaznica je određena za jednu izvedbu. U svim pojedinačnim klasama se nalaze ID obeležja, nazivi ali i dodatne informacije u vidu imena, prezimena, broja godina, cene, datuma itd...A pored toga nalaze se i informacije stranih ključeva.

4.3.Dijagram sekvenci

Dijagram sekvenci je UML dijagram interakcije. On prikazuje hronologiju prenošenja poruka između objekata u sistemu i učesnika. Dijagram sekvenci prikazan je na slici 4.3.

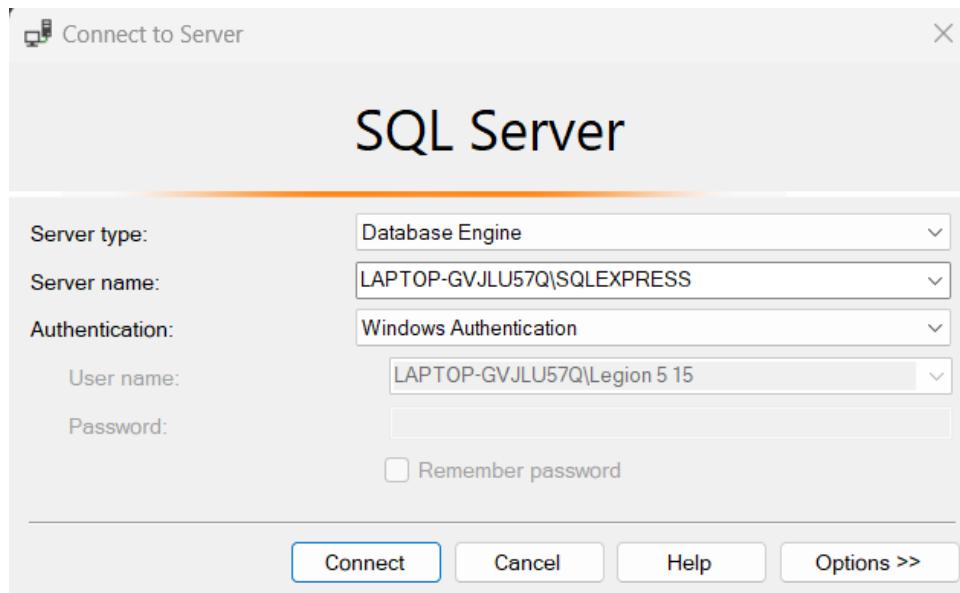


Slika 4.3 - Sekvencijalni dijagram za informacioni sistem pozorište

1. Baza podataka

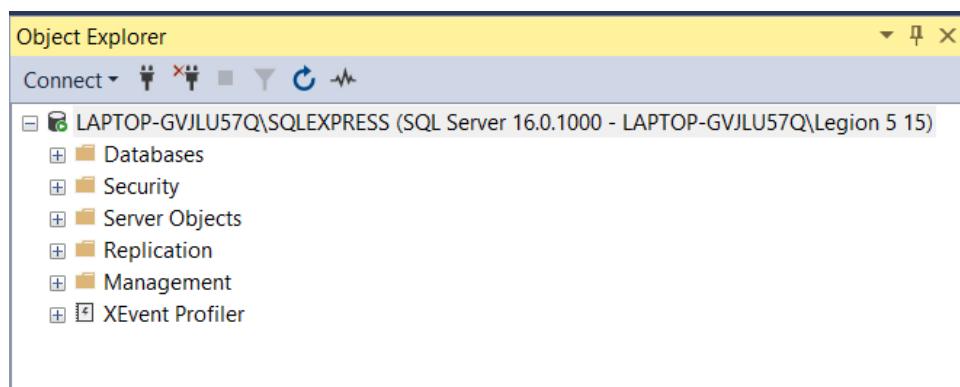
5.1 Kreiranje baze podataka

Sledeći korak u izradi informacionog sistema jeste kreiranje baze podataka. Na samom početku, potrebno je pokrenuti SQL Server Management Studio Express okruženje. Kada se to uradi, pojaviće se prozor sa dijalogom, kao što je prikazano na slici 5.1. U prikazanom dijalogu, potrebno je podesiti parametre *Server type* i *Authentication*, kao što je prikazano na slici 5.1. Napomena: *Server name* nije potrebno menjati, ono predstavlja naziv uređaja na kojem se kreira baza podataka.



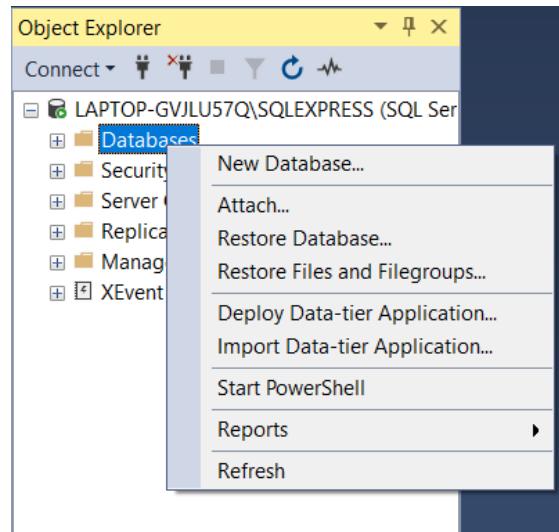
Slika 5.1 - Ostvarivanje konekcije na server

Nakon podešavanja parametara, klikom tastera miša na dugme **Connect** dijalog će se zatvoriti, a otvorice se novi prozor koji treba da izgleda kao što je prikazano na slici 5.2.



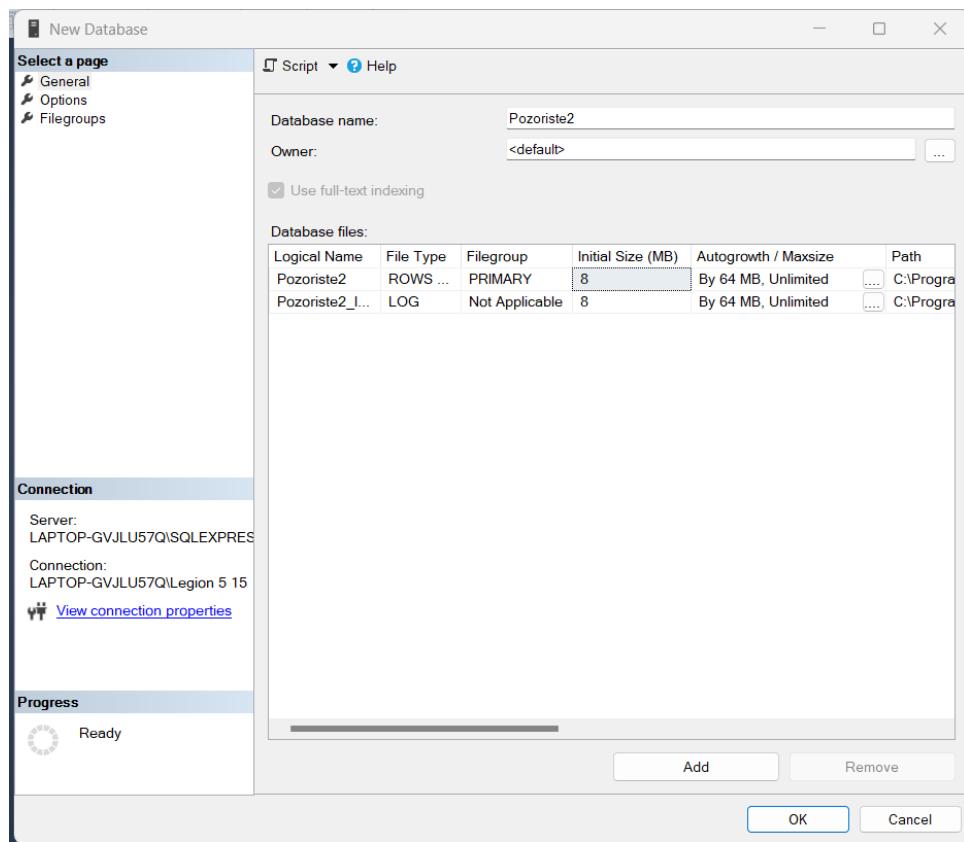
Slika 5.2 – Izgled Object Explorer-a nakon ostvarivanja konekcije sa bazom podataka

Sada je potrebno kreirati bazu podataka za aplikaciju pozorišta. Baza podataka se kreira desnim klikom tastera miša na **Database** u treeviewu koji se nalazi sa leve strane prozora u okviru Object Explorer-a. Otvoriće se lista u kojoj je potrebno označiti **New Database**.



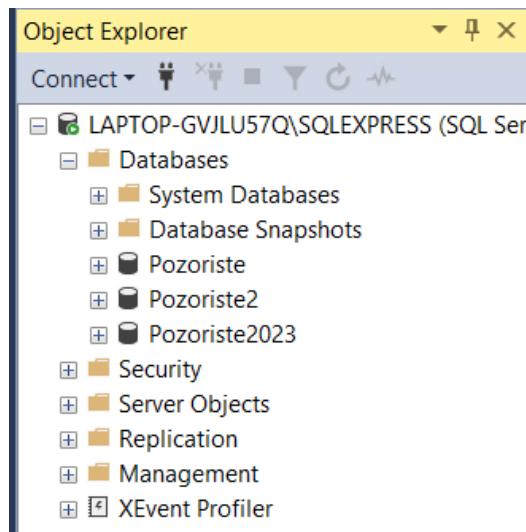
Slika 5.3 - Kreiranje baze podataka

Novi prozor ponudiće opciju za kreiranje baze podataka. Potrebno je dati naziv bazi podataka, u ovom slučaju naziv će biti *Pozoriste*. Ostale parametar nije potrebno menjati.



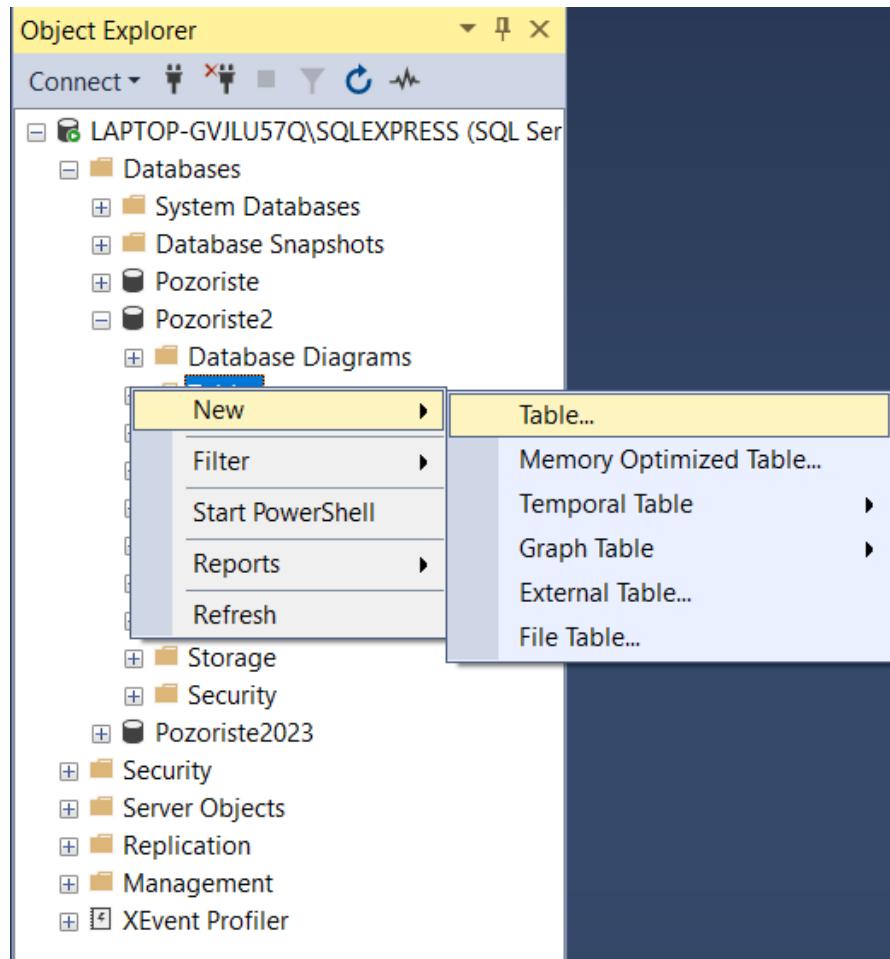
Slika 5.4 – Podešavanje parametara pri kreiranju baze podataka

Posle uspešnog kreiranja baze podataka u *Object Explorer*-u u folderu *Databases* se pojavljuje kreirana baza podataka Pozoriste.



Slika 5.5 – Prikaz kreirane baze podataka

Proširivanjem baze podataka Pozoriste, u stablu Object Explorer-a, pojavljuje se više datoteka. Desnim klikom tastera miša kliknuti na datoteku Tables i odabrati Table (slika 5.6). Nakon klika na ovu opciju, sa desne strane otvara se prozor za definisanje tabele, pri čemu će tabela imati predefinisani naziv Table_1.



Slika 5.6 – 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 označava da li određeno polje može imati null vrednosti. Preporučuje se da Allow Nulls bude označeno za sve kolone, osim za primarni ključ, kako bi se izbegli određeni problemi prilikom upisa rekorda u tabelu (slika 5.7).

Column Name	Data Type	Allow Nulls
		<input type="checkbox"/>

Slika 5.7 – Dodavanje obeležja unutar tabele

Svaka tabela određene baze podataka mora posedovati primarni ključ - ID i bilo koja kolona može da se deklariše kao primarni ključ. U ovom slučaju, kolona pozoristeID treba da se deklariše kao primarni ključ. Kao takva, ova kolona će zbog svoje jedinstvene vrednosti imati tip podatka int i neće imati null vrednosti (slika 5.8).

Column Name	Data Type	Allow Nulls
pozoristeID	int	<input type="checkbox"/>
zaposleniID	int	<input checked="" type="checkbox"/>
naziv	nvarchar(50)	<input checked="" type="checkbox"/>
lokacija	nvarchar(50)	<input checked="" type="checkbox"/>
direktor	nvarchar(50)	<input checked="" type="checkbox"/>

Slika 5.8 – Prikaz kreiranih obeležja

Da bi se kolona pozoristeID deklarisala kao primarni ključ, potrebno je kliknuti desnim klikom tastera miša na crnu strelicu koja se nalazi pored naziva ove kolone. Otvoriće lista u kojoj treba odabratи opciju *Set Primary Key* (slika 5.9). Ovom komandom se kolona pozoristeID deklariše kao primarni ključ.

Column Name	Data Type	Allow Nulls
pozoristeID	int	<input type="checkbox"/>
zaposleniID		<input checked="" type="checkbox"/>
naziv		<input checked="" type="checkbox"/>
lokacija		<input checked="" type="checkbox"/>
direktor		<input checked="" type="checkbox"/>

- Set Primary Key
- Insert Column
- Delete Column
- Relationships...
- Indexes/Keys...
- Fulltext Index...
- XML Indexes...
- Check Constraints...
- Spatial Indexes...
- Generate Change Script...
- Properties

Slika 5.9 – Definisanje kolone pozoristeID za primarni ključ

Da bi sistem za upravljanje bazom podataka - SUBP automatski kreirao primarni ključ, potrebno je u *Column Properties* pronaći opciju *Identity Specification*. Proširivanjem opcije *Identity Specification* dobija se opcija *Is Identity*, koju treba promeniti na *Yes*. Ovde je, takođe, moguće podešiti parametre *Identity Increment* i *Identity Seed*. *Identity Increment-om* se definiše za koliko raste sledeća vrednost ključa, dok *Identity Seed* označava od koje vrednosti počinju da se memorišu ključevi (slika 5.10).

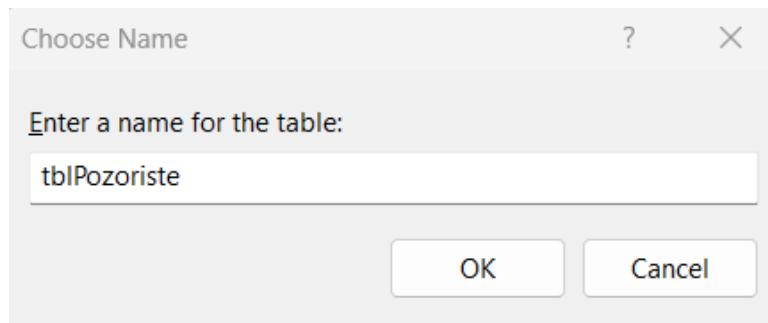
	Column Name	Data Type	Allow Nulls
PK	pozoristeID	int	<input type="checkbox"/>
	zaposleniID	int	<input checked="" type="checkbox"/>
	naziv	nvarchar(50)	<input checked="" type="checkbox"/>
	lokacija	nvarchar(50)	<input checked="" type="checkbox"/>
	direktor	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Slika 5.10 – Prikazana kolona pozoristeID kao primarni ključ

Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes

Slika 5.11 – Podešavanje automatske inkrementacije vrednosti obeležja primarnog ključa

Čuvanje tabele vrši se pritiskanjem tastera *Ctrl* i *S* nakon čega će SQL Server Management Studio zahtevati unos naziva tabele. Naziv tabele se najčešće navodi sa prefiksom „tbl“, u ovom slučaju *tblPozoriste*.



Slika 5.12 – Čuvanje kreirane tabele

Isti proces treba ponoviti za kreiranje tabela *tblGledalac*, *tblGlumac*, *tblIzvedba*, *tblPredstava*, *tblReditelj*, *tblUlaznica* i *tblZaposleni* (slike 5.13-5.19).

LAPTOP-GVJLU57Q\S...- dbo.tblGledalac			
	Column Name	Data Type	Allow Nulls
PK	gledalacID	int	<input type="checkbox"/>
	ime	nvarchar(50)	<input checked="" type="checkbox"/>
	prezime	nvarchar(50)	<input checked="" type="checkbox"/>
	pol	nvarchar(50)	<input checked="" type="checkbox"/>

Slika 5.13 – Kreiranje tabele *tblGledalac*

LAPTOP-GVJLU57Q\S...2 - dbo.tblGlumac			
	Column Name	Data Type	Allow Nulls
PK	glumaclID	int	<input type="checkbox"/>
	ime	nvarchar(50)	<input checked="" type="checkbox"/>
	prezime	nvarchar(50)	<input checked="" type="checkbox"/>
	godine	int	<input checked="" type="checkbox"/>
	pol	nvarchar(50)	<input checked="" type="checkbox"/>

Slika 5.14 – Kreiranje tabele *tblGlumac*

LAPTOP-GVJLU57Q\S...- dbo.tbllzvedba			
	Column Name	Data Type	Allow Nulls
PK	izvedbalID	int	<input type="checkbox"/>
	predstavaID	int	<input checked="" type="checkbox"/>
	vremePocetka	datetime	<input checked="" type="checkbox"/>

Slika 5.15 – Kreiranje tabele *tbllzvedba*

LAPTOP-GVJLU57Q\S...dbo.tblPredstava			
	Column Name	Data Type	Allow Nulls
PK	predstavaID	int	<input type="checkbox"/>
	glumaclID	int	<input checked="" type="checkbox"/>
	rediteljID	int	<input checked="" type="checkbox"/>
	naslov	nvarchar(50)	<input checked="" type="checkbox"/>
	autor	nvarchar(50)	<input checked="" type="checkbox"/>
	zanr	nvarchar(50)	<input checked="" type="checkbox"/>

Slika 5.16 – Kreiranje tabele *tblPredstava*

LAPTOP-GVJLU57Q\S...- dbo.tblReditelj			
	Column Name	Data Type	Allow Nulls
PK	rediteljID	int	<input type="checkbox"/>
	ime	nvarchar(50)	<input checked="" type="checkbox"/>
	prezime	nvarchar(50)	<input checked="" type="checkbox"/>
	godine	int	<input checked="" type="checkbox"/>

Slika 5.17 – Kreiranje tabele *tblReditelj*

Column Name	Data Type	Allow Nulls
ulaznicalD	int	<input type="checkbox"/>
gledalacID	int	<input checked="" type="checkbox"/>
pozoristelD	int	<input checked="" type="checkbox"/>
izvedbalD	int	<input checked="" type="checkbox"/>
cena	int	<input checked="" type="checkbox"/>
brojSedista	int	<input checked="" type="checkbox"/>

Slika 5.18 – Kreiranje tabele *tblUlaznica*

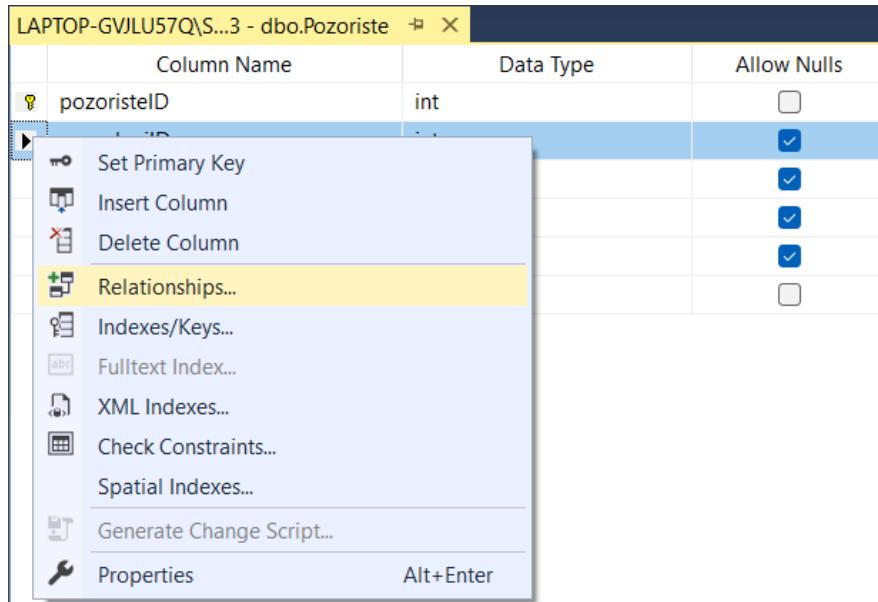
Column Name	Data Type	Allow Nulls
zaposleniD	int	<input type="checkbox"/>
ime	nvarchar(50)	<input checked="" type="checkbox"/>
prezime	nvarchar(50)	<input checked="" type="checkbox"/>
pozicija	nvarchar(50)	<input checked="" type="checkbox"/>
radnoVreme	int	<input checked="" type="checkbox"/>

Slika 5.19 – Kreiranje tabele *tblZaposleni*

Nakon toga bi u *treeview-u*, koji se nalazi sa leve strane prozora u okviru *Object Explorer-a*, trebala da se nalazi baza podataka *Pozoriste*, u kojoj će se, pored već kreirane *tblPozoriste*, naći i novokreirane tabele sa odgovarajućim atributima (slika 5.20).

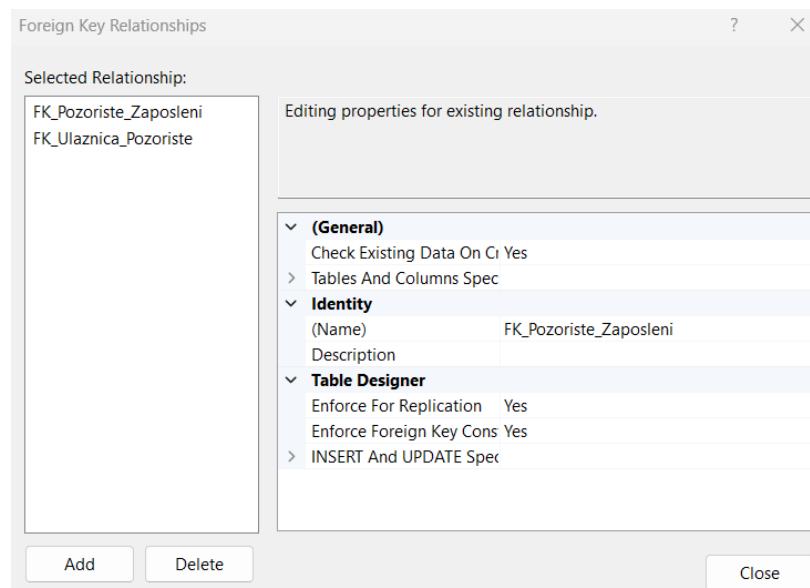
Slika 5.20 – Object Explorer prikaz kreiranih tabela i kolona

Kao što se može videti na slici 5.20, u *tblPozoriste* se nalazi strani ključ (*zaposleniID*). Prilikom unosa atributa u tabelu *tblPozoriste*, potrebno je uneti i atribut koji ima naziv *zaposleniID*, a vrednost *Data Type-a* za njega staviti na int. Zatim je potrebno kliknuti desnim klikom tastera miša na crnu strelicu koja se nalazi pored naziva ove kolone. Otvoriće se lista u kojoj treba odabratи opciju *Relationships* (slika 5.21).



Slika 5.21 – Dodavanje veza

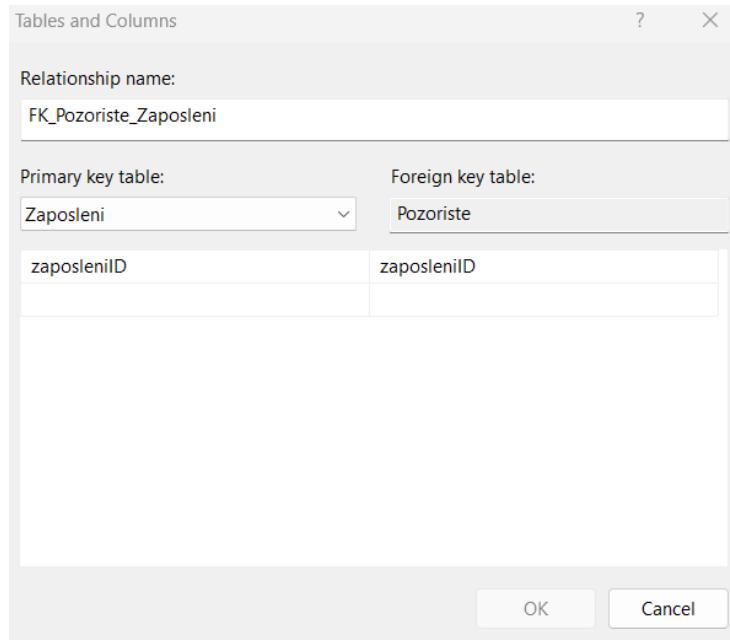
Nakon odabira opcije *Relationships*, otvara se prozor *Foreign key Relationships* (slika 5.22) u kojem je potrebno prvo kliknuti tasterom miša na dugme *Add*, a zatim proširiti polje *Tables and Columns Specifications* i kliknuti na tri tačke koje se nalaze desno.



Slika 5.22 – Dodavanje stranog ključa

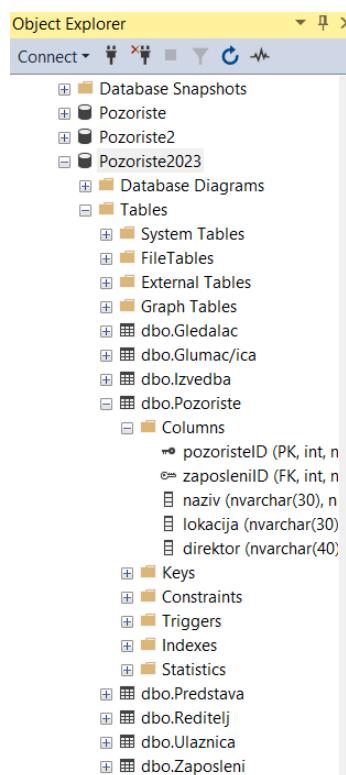
U otvorenom prozoru *Tables and Columns* potrebno je u okviru *Foreign key table*, konkretno iz tabele *tblPozoriste*, iz padajuće liste izabrati atribut definisan prilikom kreiranja tabele, a koji je potrebno povezati sa drugom tabelom. Zatim, u okviru *Primary key table* treba izabrati

tabelu sa kojom se vrši povezivanje, i iz padajuće liste odabrati ključ te tabele sa kojim se povezuje željeni atribut. Nakon što je sve ovo urađeno, prozor treba da izgleda kao na sledećoj slici (slika 5.23).



Slika 5.23 – Kreiranje veza između tabela

Dodavanjem *Relationships-a* atributu *zaposlenID* u tabeli *tblZaposleni*, oni postaju strani ključevi. Da li je povezivanje tabele uspešno urađeno znaće se proverom postojanja u *treeview-u*, u okviru tabele *tblZaposleni*, pod kategorijom *Keys* (slika 5.24).



Slika 5.24 – TreeView prikaz kreiranih ključeva

Isti postupak potrebno je ponoviti za sve tabele koje u sebi sadrže strane ključeve koji su primarni ključevi u drugoj tabeli, *tblPozoriste* (*pozoristeID*). Sada su tabele kreirane i povezane, a dodavanje novih podataka u ovu bazu podataka se može izvršiti desnim klikom tastera miša na naziv tabele, gde iz liste opcija treba odabratи **Edit Top 200 Rows** (slika 5.25).

	pozoristelD	zaposleniID	naziv	lokacija	direktor
▶*	NULL	NULL	NULL	NULL	NULL

Slika 5.26 – Prikaz kreirane tabele i dodavanje podataka

Baza podataka kreirana je na način koji je prikazan kroz celo poglavlje 5. Sledeći korak u izradi projektnog zadatka je kreiranje WPF aplikacije.

5.2 Upiti i DML naredbe

Na slici 5.27 prikazan je kod SQL naredbi sa kojim je izvršen unos inicijalnih pet vrednosti u svaku tabelu u bazu podataka *Pozoriste*.

```

Pozoriste2023Vanja...Q\Legion 5 15 (59)* ↗ X
--TABELA: Glumac/ica
INSERT INTO [Glumac/ica] VALUES ('Nevena', 'Novakovic', 25, 'zenski')
INSERT INTO [Glumac/ica] VALUES ('Mina', 'Markovic', 20, 'zenski')
INSERT INTO [Glumac/ica] VALUES ('Jovan', 'Petrovic', 21, 'muski')
INSERT INTO [Glumac/ica] VALUES ('Milan', 'Milanovic', 23, 'muski')
INSERT INTO [Glumac/ica] VALUES ('Vanja', 'Vojinovic', 20, 'zenski')

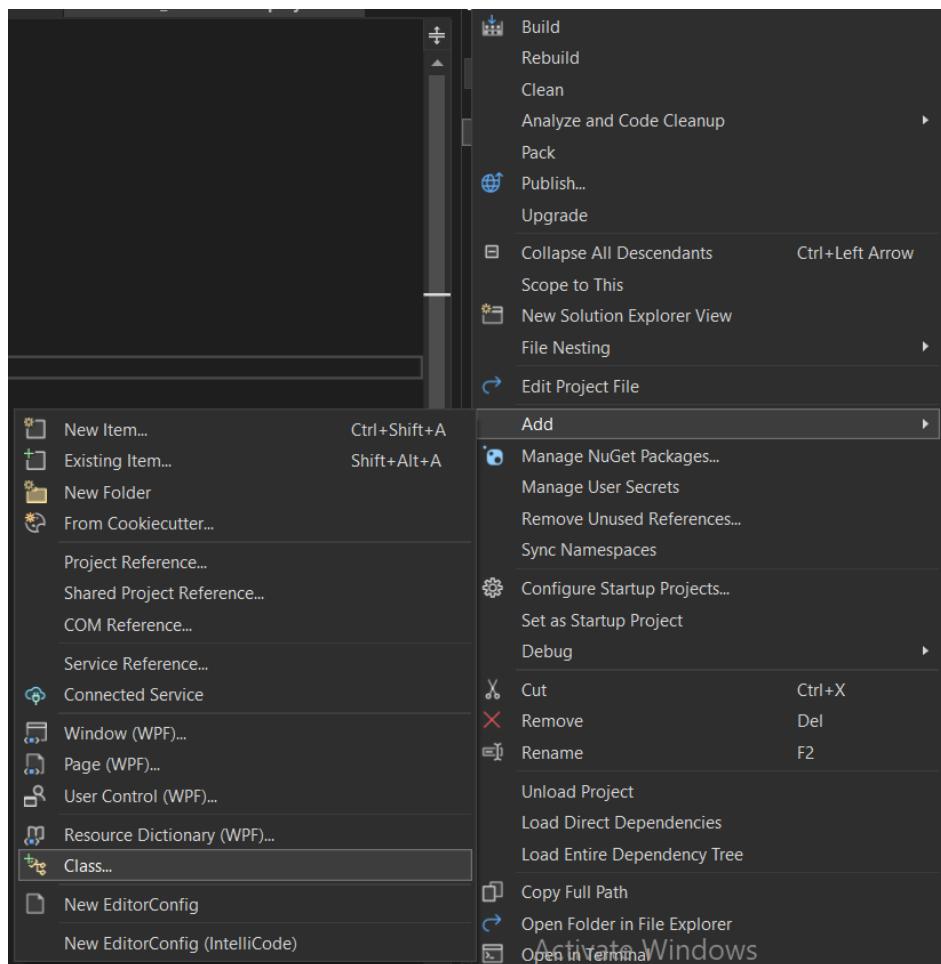
--TABELA: Reditelj
INSERT INTO Reditelj VALUES ('Radenko', 'Radenkovic', 45)
INSERT INTO Reditelj VALUES ('Marica', 'Malinovic', 50)
INSERT INTO Reditelj VALUES ('Jasmina', 'Savic', 39)
INSERT INTO Reditelj VALUES ('Janko', 'Mitrovic', 40)
INSERT INTO Reditelj VALUES ('David', 'Stamenkovic', 35)

```

Slika 5.27 – Unos podataka u bazu podataka

6. Kreiranje WPF aplikacije

U okviru ovog poglavlja biće prikazano na koji način je kreirana WPF aplikacija za informacioni sistem Pozorište. Nakon pokretanja razvojnog okruženja *Visual Studio*, potrebno je tasterom miša kliknuti na *File menu*, a zatim odabratи *New Project*. Odabratи *Windows Classic Desktop*, zatim *WPF Application*. Navesti ime aplikacije i kliknuti OK. U *Solution Explorer-u* pojaviće se napravljena aplikacija. Sledeći korak je kreiranje konekcije u aplikaciji, ka bazi podataka. Za početak, kreira se klasa u kojoj će se nalaziti metoda putem koje će se ostvarivati veza na prethodno kreiranu bazu. Klasa se kreira desnim klikom tastera miša na naziv projekta (IT28G2022_SkoricVanja_Pozoriste), a zatim *Add/Class* (slika 6.1).



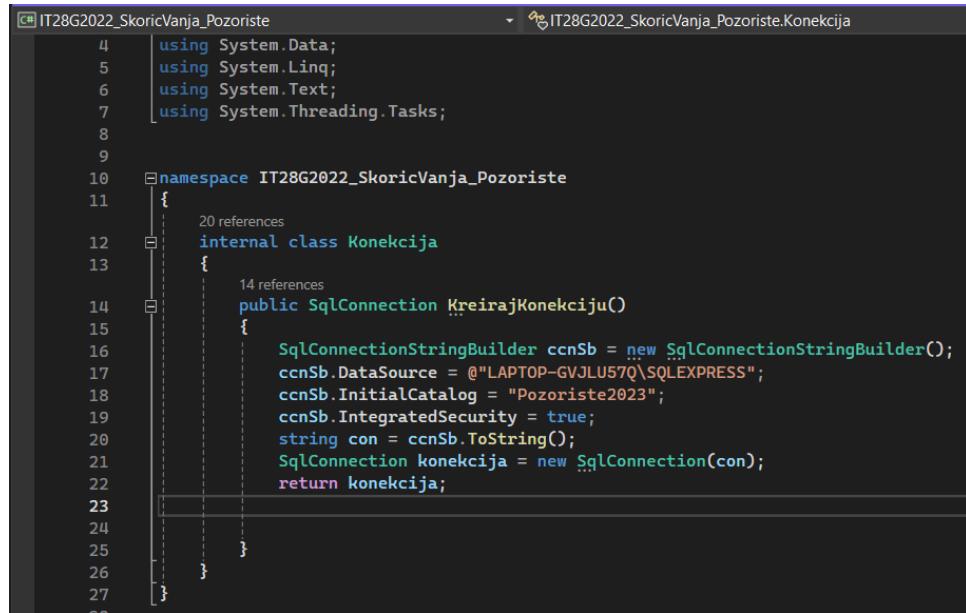
Slika 6.1 – Kreiranje klase

Novokreiranoj klasi se dodeljuje naziv „*Konekcija*” i njen sadržaj prikazan je u listingu 6.1. Na samom početku potrebno je uključiti imenski prostor *System.Data.SqlClient*. Telo klase sadrži samo jednu metodu pod nazivom "*KreirajKonekciju*", koja će se u kasnijem razvoju aplikacije pozivati po potrebi, odnosno svaki put kada je potrebno ostvariti vezu sa bazom podataka, stoga je povratni tip ove metode instanca klase *SqlConnection*. Unutar same metode prikazana su podešavanja parametara konekcije i *Add to Source Control*.

- 1) *DataSource* - naziv servera na kojem je baza podataka smeštena,
- 2) *Initial Catalog* - naziv baze podataka kojoj je potrebno pristupiti i
- 3) *IntegratedSecurity* - ukoliko se baza nalazi na lokalnoj mašini, ovaj parametar postavlja se

na *true*.

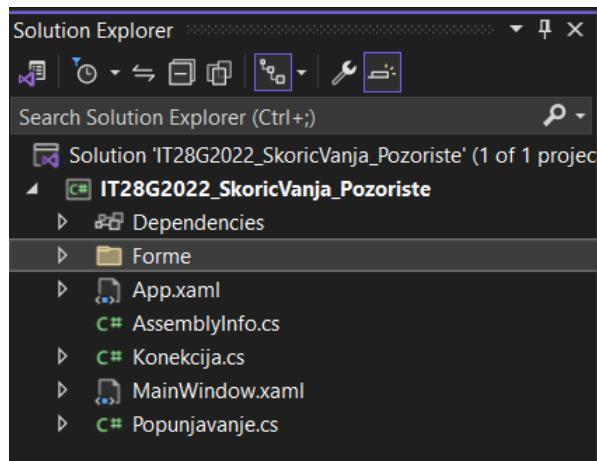
Nakon podešavanja ovih parametara, ceo string se prosleđuje instanci klase *SqlConnection*.



```
IT28G2022_SkoricVanja_Pozoriste      IT28G2022_SkoricVanja_Pozoriste.Konekcija
4   using System.Data;
5   using System.Linq;
6   using System.Text;
7   using System.Threading.Tasks;
8
9
10  namespace IT28G2022_SkoricVanja_Pozoriste
11  {
12      internal class Konekcija
13      {
14          public SqlConnection KreirajKonekciju()
15          {
16              SqlConnectionStringBuilder ccnSb = new SqlConnectionStringBuilder();
17              ccnSb.DataSource = @"LAPTOP-GVJLU57Q\SQLEXPRESS";
18              ccnSb.InitialCatalog = "Pozoriste2023";
19              ccnSb.IntegratedSecurity = true;
20              string con = ccnSb.ToString();
21              SqlConnection konekcija = new SqlConnection(con);
22              return konekcija;
23          }
24      }
25  }
26
27
28
```

Listing 6.1 – Klasa Konekcija

Sledeći korak u izradi WPF aplikacije je kreiranje ekranskih formi (*eng. Forms*) - prozora, za dodavanje objekata u bazu podataka. Kako bi sve forme bile uredno smeštene na jednom mestu, potrebno je kreirati folder pod nazivom Forme u kom će se nalaziti svih osam formi za, prethodno kreiranih, osam tabela u bazi podataka. Kreiranje foldera se vrši na sličan način kao kreiranje klase, desnim klikom tastera miša na projekat, a zatim **Add/New Folder**.



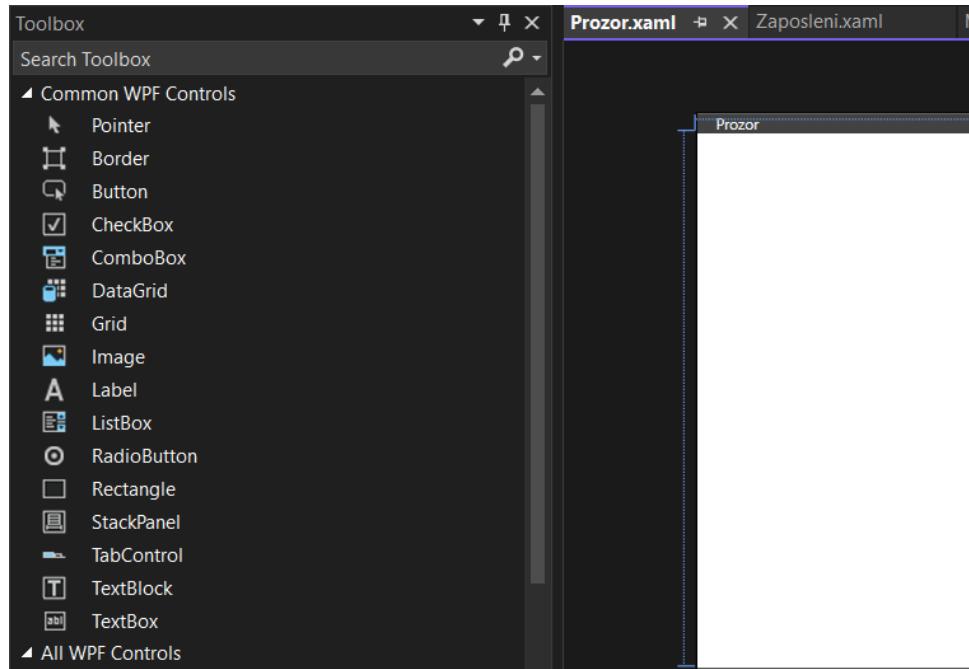
Slika 6.2 – Kreiran folder Forme

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

Nakon toga, prikazaće se prozor za dodavanje novog *Item-a* unutar kog je potrebno podesiti naziv i zatim kliknuti na **Add**.

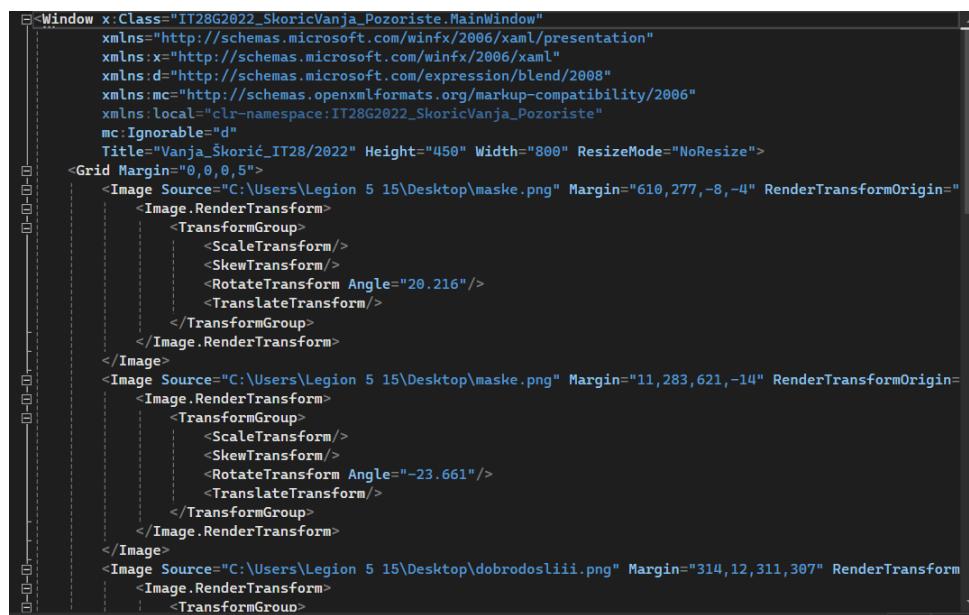
Kada je novi prozor izgenerisan, potrebno je, unutar njega, dodati određene elemente putem kojih će se ostvarivati potrebne funkcionalnosti. Na levoj strani prozora, klikom tastera miša

na *Toolbox*, prikazaće se lista svih ugrađenih WPF komandi (slika 6.3). Jednostavnim prevlačenjem tih komandi na centralni prozor, izgenerisatiće se kod u *XAML* deklarativnom jeziku unutar kojeg će se vršiti detaljna podešavanja svih komandi unutar prozora.



Slika 6.3 - Dodavanje komandi iz ToolBox-a

Forma za dodavanje novog Pozorišta u bazu podataka treba da sadrži elemente prikazane u listingu 6.2. To je osnovni *TextBox* unutar kog će biti upisana vrednost obeležja... Takođe, uz svaki *TextBox*, mora postojati naziv (*Label*) koji će opisivati njegovu namenu. Na kraju, svaka forma mora imati dugme Sačuvaj, na čiji klik tasterom miša će se vrednosti upisivati u bazu podataka, i dugme Poništi, koje korisniku omogućava zatvaranje forme. Ove funkcionalnosti dugmića postižu se dvostrukim klikom tastera miša na odgovarajuće dugme u okviru forme, čime se u *Code Behind*-u generišu metode *BtnSacuvaj_Click* i *BtnZatvori_Click*.



Listing 6.3.1 - XAML kod kreiranih komponenti

```

33 |           <ScaleTransform/>
34 |           <SkewTransform/>
35 |           <RotateTransform Angle="-0.332" />
36 |           <TranslateTransform />
37 |       </TransformGroup>
38 |   </Image.RenderTransform>
39 | </Image>
40 | <DataGrid Name="dataGridCentralni" HorizontalAlignment="Center" VerticalAlignment="Center" Height="auto" AlternatingRowBackground="#IndianRed" CanUserAddRows="False" SelectionUnit="FullRow" SelectedItem="row1" SelectionMode="Single" Width="100%" GridLinesVisibility="None" Margin="10"/>
41 |     <StackPanel HorizontalAlignment="Center" Height="30" VerticalAlignment="Top" Width="auto" Orientation="Horizontal" Margin="0 10 0 0">
42 |       <Button x:Name="btnPozoriste" Content="Pozorište" Height="30" VerticalAlignment="Top" Width="100" Click="btnPozoriste_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
43 |       <Button x:Name="btnPredstava" Content="Predstava" Height="30" VerticalAlignment="Top" Width="100" Click="btnPredstava_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
44 |       <Button x:Name="btnIzvedba" Content="Izvedba" Height="30" VerticalAlignment="Top" Width="100" Click="btnIzvedba_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
45 |       <Button x:Name="btnGlumac" Content="Glumac" Height="30" VerticalAlignment="Top" Width="100" Click="btnGlumac_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
46 |       <Button x:Name="btnReditelj" Content="Reditelj" Height="30" VerticalAlignment="Top" Width="100" Click="btnReditelj_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
47 |       <Button x:Name="btnGledalac" Content="Gledalac" Height="30" VerticalAlignment="Top" Width="100" Click="btnGledalac_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
48 |       <Button x:Name="btnZaposleni" Content="Zaposleni" Height="30" VerticalAlignment="Top" Width="100" Click="btnZaposleni_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
49 |       <Button x:Name="btnUlanzica" Content="Ulanžica" Height="30" VerticalAlignment="Top" Width="100" Click="btnUlanzica_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
50 |   </StackPanel>
51 |   <StackPanel HorizontalAlignment="Center" Height="30" VerticalAlignment="Bottom" Width="360" Orientation="Horizontal" Margin="0 10 0 0">
52 |     <Button x:Name="btnDodaj" Content="Dodaj" Width="121" Click="btnDodaj_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
53 |     <Button x:Name="btnIzmeni" Content="Izmeni" Width="121" Click="btnIzmeni_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
54 |     <Button x:Name="btnObriši" Content="Obriši" Width="120" Click="btnObriši_Click" Background="#FFEDC0" BorderRadius="10" BorderThickness="1" Foreground="Black" Padding="5 0 5 0" Margin="10 0 10 0" Style="text-decoration: none; font-size: 1em; font-weight: bold; color: black; text-align: center; border: 1px solid black; outline: none;"/>
55 |   </StackPanel>
56 |   <Grid.Background>
57 |     <ImageBrush Stretch="None" ImageSource="C:\Users\Legion 5 15\Desktop\pozoriste.jpg" AlignmentY="Top" AlignmentX="Left" Opacity="0.5" />
58 |   </Grid.Background>
59 |
60 |
61 | </Grid>
62 | </Window>
63 |

```

Listing 6.3.2 - XAML kod kreiranih komponenti

Vizuelni prikaz forme za unos novog pozorišta, izgledaće kao na slici 6.4.



Slika 6.4 – Vizuelni prikaz forme za dodavanje/izmenu pozorišta

U *Code Behind*-u ove forme, nalazi se sledeća poslovna logika (listing 6.3). Najpre je potrebno uključiti imenski prostor *System.Data.SqlClient*. Zatim, potrebno je napraviti novi objekat *SqlConnection* klase, putem kojeg će se ostvarivati veza ka prethodno kreiranoj bazi podataka, nakon klika tasterom miša na dugme Sačuvaj. Takođe, u konstruktoru ovog prozora poželjno je podesiti metodu *Focus()*, koja pozicionira kurSOR na prvo polje koje korisnik popunjava pri pokretanju forme. U slučaju forme *frmPozorište*, to je tekst polje za unos naziva pozorišta.

```

namespace IT28G2022_SkoricVanja_Pozoriste.Forme
{
    /// <summary>
    /// Interaction logic for Pozoriste.xaml
    /// </summary>
    6 references
    public partial class Pozoriste : Window
    {
        Konekcija kon = new Konekcija();
        SqlConnection konekcija = new SqlConnection();
        private bool azuriraj;
        private DataRowView red;
        private static string zaposleni = @"select ime + ' ' + prezime as Zaposleni , zaposleniID from Zaposleni";
        1 reference
        public Pozoriste()
        {
            InitializeComponent();
            Popunjavanje.Popuni(cmbZaposleni, zaposleni);
            cmbZaposleni.Focus();
            konekcija = kon.KreirajKonekciju();
        }

        1 reference
        public Pozoriste(bool azuriraj, DataRowView red)
        {
            this.azuriraj = azuriraj;
            this.red = red;
            InitializeComponent();
            Popunjavanje.Popuni(cmbZaposleni, zaposleni);
        }

        1 reference
        private void btnSacuvaj_Click(object sender, RoutedEventArgs e)

```

Listing 6.5 – Sadržaj CodeBehind-a forme DodajPozorište

Ovakav prikaz pozadinskog koda gotovo je identičan za svaku od osam kreiranih formi. Ono u čemu je svaka od ovih formi jedinstvena jeste pozadinska funkcionalnost koja se dešava u trenutku kada korisnik tasterom miša klikne na dugme Sačuvaj. U listingu 6.4 prikazan je sadržaj izgenerisane metode *Btn_Sacuvaj_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 novi objekat u bazu podataka. U odeljak *values* ove naredbe, prosleđuju se vrednosti koje su unete u *TextBox-ove* ove forme. Potrebno je voditi računa da se vrednosti string-ova ograde apostrofima. Nakon uspešno napravljenje insert komande, ovaj string se prosleđuje objektu klase *SqlCommand*. Ovaj objekat, kao parametre prima prethodno kreirani string, ali i objekat klase *Connection* putem kojeg 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 ogradići *try* blokom iz razloga što postoji mogućnost nastanka greške usled prosleđivanja unosa pogrešnog tipa podatka od strane korisnika. Kako prilikom ovakvog unosa, program ne bi "pukao", potrebno je dodati *catch* blok unutar kojeg će se "hvataći" 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.

```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@zaposleniID", SqlDbType.Int).Value = (int)cmbZaposleni.SelectedValue;
        cmd.Parameters.Add("@naziv", SqlDbType.NVarChar).Value = txtNaziv.Text;
        cmd.Parameters.Add("@lokacija", SqlDbType.NVarChar).Value = txtLokacija.Text;
        cmd.Parameters.Add("@direktor", SqlDbType.NVarChar).Value = txtDirektor.Text;

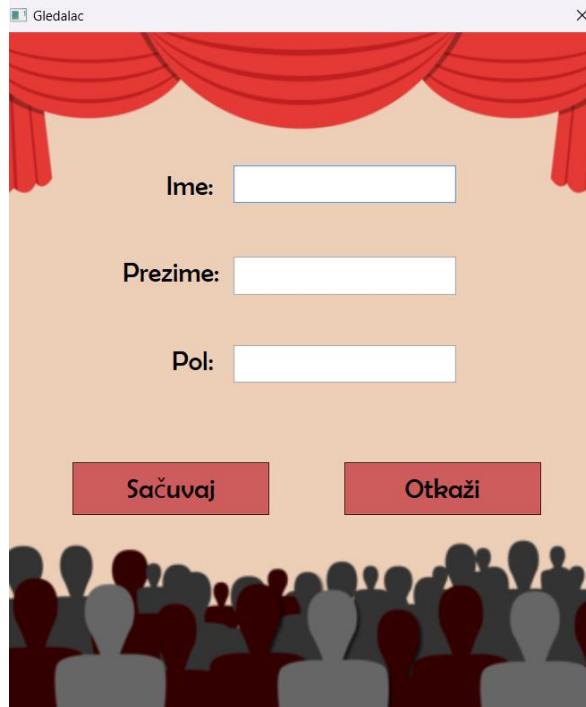
        if (azuriraj)
        {
            cmd.Parameters.Add("@id", SqlDbType.Int).Value = red["ID"];
            cmd.CommandText = @"update Pozoriste
                                set zaposleniID = @zaposleniID, naziv = @naziv, lokacija = @lokacija, direktor = @direktor
                                where pozoristeID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Pozoriste
                                values (@zaposleniID, @naziv, @lokacija, @direktor)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos određenih vrednosti nije validan.");
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Listing 6.6 – Sadržaj metode BtnSacuvaj_Click za unos nove pozorište

U nastavku će biti prikazan izgled i ostalih sedam formi.

Forme za dodavanje gledaoca, glumaca, izvedbi, predstava, reditelja, ulaznica i zaposlenih imaju sličan dizajn i XAML kod koji će biti prikazani u narednim slikama.



Slika 6.7 – Dizajn forme za dodavanje/izmenu gledaoca

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Gledalac"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Gledalac" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\publika.png" Margin="-9,-1,-9,-68" />
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-273,-35,210" />
        <Label x:Name="lblIme" Content="Ime:" HorizontalAlignment="Left" Margin="142,119,0,0" VerticalAlignment="Top"/>
        <Label x:Name="lblPrezime" Content="Prezime:" HorizontalAlignment="Left" Margin="103,197,0,0" VerticalAlignment="Top"/>
        <Label x:Name="lblPol" Content="Pol:" HorizontalAlignment="Left" Margin="147,276,0,0" VerticalAlignment="Top"/>
        <TextBox x:Name="txtIme" HorizontalAlignment="Left" Margin="207,119,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"/>
        <TextBox x:Name="txtPrezime" HorizontalAlignment="Left" Margin="207,201,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top"/>
        <TextBox x:Name="txtPol" HorizontalAlignment="Left" Margin="207,280,0,0" TextWrapping="Wrap" VerticalAlignment="Top"/>
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="63,385,0,0" VerticalAlignment="Bottom"/>
        <Button x:Name="btnOtkazi" Content="Otkazi" HorizontalAlignment="Left" Margin="306,385,0,0" VerticalAlignment="Bottom"/>
    </Grid>
</Window>

```

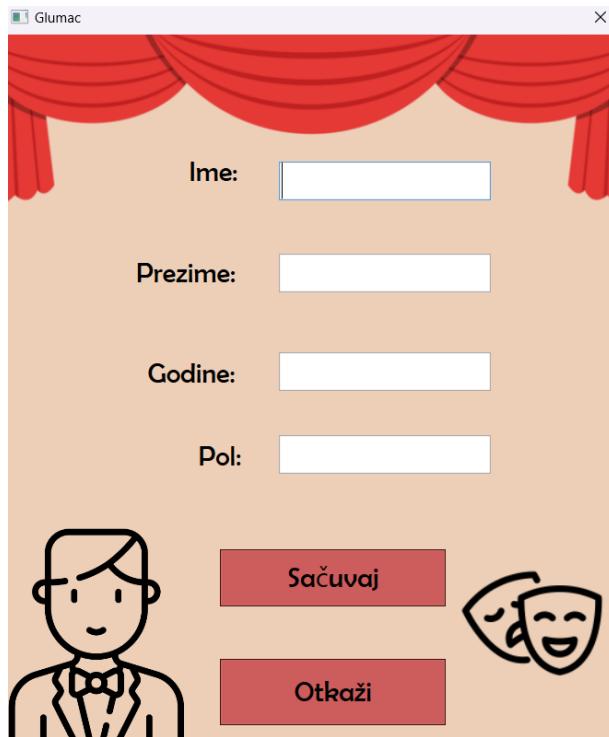
Slika 6.8 - XAML kod forme za dodavanje/izmenu gledaoca

```

1 reference
private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@ime", SqlDbType.NVarChar).Value = txtIme.Text;
        cmd.Parameters.Add("@prezime", SqlDbType.NVarChar).Value = txtPrezime.Text;
        cmd.Parameters.Add("@pol", SqlDbType.NVarChar).Value = txtPol.Text;
        if (azuriraj)
        {
            cmd.Parameters.Add("@id", SqlDbType.Int).Value = red["ID"];
            cmd.CommandText = @"update Gledalac
                                set ime = @ime, prezime = @prezime, pol = @pol
                                where gledalacID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Gledalac
                                values (@ime, @prezime, @pol)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos određenih vrednosti nije validan.");
    }
    finally
    {
        if(konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.9 - Sadržaj metode BtnSacuvaj_Click za unos novog gledaoca



Slika 6.10 – Dizajn forme za dodavanje/izmenu glumca

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Glumac"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Glumac" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-273,-35,210" />
        <Image Source="C:\Users\Legion 5 15\Desktop\maskice.png" Margin="362,306,-19,-102" />
        <Image Source="C:\Users\Legion 5 15\Desktop\glumaac.png" Margin="-12,366,361,-67" />
        <Label x:Name="lblIme" Content="Ime:" HorizontalAlignment="Left" Margin="157,101,0,0" VerticalAlignment="Top" />
        <Label x:Name="lblPrezime" Content="Prezime:" HorizontalAlignment="Left" Margin="111,189,0,0" VerticalAlignment="Top" />
        <Label x:Name="lblGodine" Content="Godine:" HorizontalAlignment="Left" Margin="121,276,0,0" VerticalAlignment="Top" />
        <Label x:Name="lblPol" Content="Pol:" HorizontalAlignment="Left" Margin="165,348,0,0" VerticalAlignment="Top" />
        <TextBox x:Name="txtIme" HorizontalAlignment="Left" Margin="240,110,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" />
        <TextBox x:Name="txtPrezime" HorizontalAlignment="Left" Margin="240,190,0,0" TextWrapping="Wrap" VerticalAlignment="Top" />
        <TextBox x:Name="txtGodine" HorizontalAlignment="Left" Margin="240,276,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" />
        <TextBox x:Name="txtPol" HorizontalAlignment="Left" Margin="240,348,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" />
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="189,447,0,0" VerticalAlignment="Top" />
        <Button x:Name="btnOtkazi" Content="Otkaži" HorizontalAlignment="Left" Margin="189,542,0,0" VerticalAlignment="Top" />
    </Grid>
</Window>

```

Slika 6.11 - XAML kod forme za dodavanje/izmenu glumca

```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@ime", SqlDbType.NVarChar).Value = txtIme.Text;
        cmd.Parameters.Add("@prezime", SqlDbType.NVarChar).Value = txtPrezime.Text;
        cmd.Parameters.Add("@godine", SqlDbType.Int).Value = int.Parse(txtGodine.Text);
        cmd.Parameters.Add("@pol", SqlDbType.NVarChar).Value = txtPol.Text;
        if (azuriraj)
        {
            cmd.CommandText = @"update [Glumac/ica]
                                set ime = @ime, prezime = @prezime, godine = @godine, pol = @pol
                                where glumacID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into [Glumac/ica]
                                values (@ime, @prezime, @godine, @pol)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos određenih vrednosti nije validan.");
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.12 - Sadržaj metode `BtnSacuvaj_Click` za unos novog glumca



Slika 6.13 – Dizajn forme za dodavanje/izmenu izvedbe

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Izvedba"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Izvedba" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-273,-35,210" />
        <Image Source="C:\Users\Legion 5 15\Desktop\scena.png" Margin="38,318,38,-219" />
        <Label x:Name="lblPredstava" Content="Predstava:" HorizontalAlignment="Left" Margin="136,136,0,0" VerticalAlignment="Top" FontSize="24" Di
        <ComboBox x:Name="cmbPredstava" HorizontalAlignment="Left" Margin="259,138,0,0" VerticalAlignment="Top" Width="183" FontSize="24" Di
        <Label x:Name="lblVremePocetka" Content="Vreme početka:" HorizontalAlignment="Left" Margin="59,225,0,0" VerticalAlignment="Top" Font
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="61,339,0,0" VerticalAlignment="Top" FontSize="24" Wi
        <Button x:Name="btnOtkazi" Content="Otkaži" HorizontalAlignment="Left" Margin="319,339,0,0" VerticalAlignment="Top" FontSize="24" Wi
        <DatePicker x:Name="dtVremePocetka" HorizontalAlignment="Left" Margin="259,225,0,0" VerticalAlignment="Top" Height="43" Width="191"
    </Grid>
</Window>

```

Slika 6.14 - XAML kod forme za dodavanje/izmenu izvedbe

```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@predstavaID", SqlDbType.Int).Value = (int)cmbPredstava.SelectedValue;
        cmd.Parameters.Add("@vremePocetka", SqlDbType.DateTime).Value = DateTime.Parse(dtVremePocetka.Text);

        if (azuriraj)
        {
            cmd.Parameters.Add("@id", SqlDbType.Int).Value = red["ID"];
            cmd.CommandText = @"update Izvedba
                                set predstavaID = @predstavaID, vremePocetka = @vremePocetka
                                where izvedbaID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Izvedba
                                values (@predstavaID, @vremePocetka)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos određenih vrednosti nije validan.");
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.15 - Sadržaj metode BtnSacuvaj_Click za unos nove izvedbe



Slika 6.16 – Dizajn forme za dodavanje/izmenu predstave

```
<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Predstava"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Predstava" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-273,-35,210" />
        <Image Source="C:\Users\Legion 5 15\Desktop\lampa.png" Margin="57,442,215,-69" />
        <ComboBox x:Name="cmbGlumac" HorizontalAlignment="Left" Margin="238,122,0,0" VerticalAlignment="Top" Width="170" FontSize="24" Disp...
        <Label x:Name="lblGlumac" Content="Glumac:" HorizontalAlignment="Left" Margin="135,122,0,0" VerticalAlignment="Top" FontSize="24" F...
        <Label x:Name="lblReditelj" Content="Reditelj:" HorizontalAlignment="Left" Margin="138,183,0,0" VerticalAlignment="Top" FontSize="24" ...
        <ComboBox x:Name="cmbReditelj" HorizontalAlignment="Left" Margin="238,183,0,0" VerticalAlignment="Top" Width="170" FontSize="24" Di...
        <Label x:Name="lblAutor" Content="Autor:" HorizontalAlignment="Left" Margin="155,334,0,0" VerticalAlignment="Top" FontSize="24" Fo...
        <Label x:Name="lblZanr" Content="Žanr:" HorizontalAlignment="Left" Margin="166,406,0,0" VerticalAlignment="Top" FontSize="24" Font...
        <Label x:Name="lblNaslov" Content="Naslov:" HorizontalAlignment="Left" Margin="145,269,0,0" VerticalAlignment="Top" FontSize="24" ...
        <TextBox x:Name="txtNaslov" HorizontalAlignment="Left" Margin="238,269,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Wid...
        <TextBox x:Name="txtAutor" HorizontalAlignment="Left" Margin="238,335,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Wid...
        <TextBox x:Name="txtZanr" HorizontalAlignment="Left" Margin="238,408,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Wid...
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="44,516,0,0" VerticalAlignment="Top" FontSize="24" ...
        <Button x:Name="btnOtkazi" Content="Otkaži" HorizontalAlignment="Left" Margin="337,516,0,0" VerticalAlignment="Top" FontSize="24" ...
    </Grid>
</Window>
```

Slika 6.17 - XAML kod forme za dodavanje/izmenu predstave

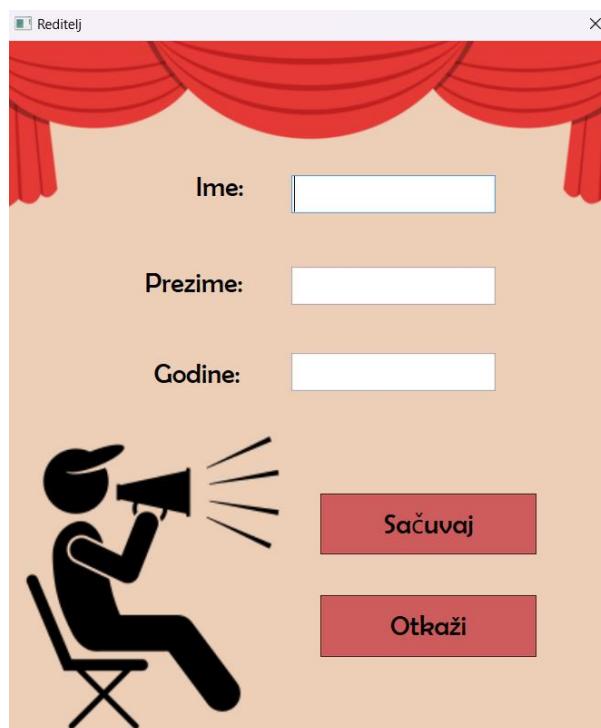
```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.AddWithValue("@glumacID", SqlDbType.Int).Value = (int)cmbGlumac.SelectedValue;
        cmd.Parameters.AddWithValue("@rediteljID", SqlDbType.Int).Value = (int)cmbReditelj.SelectedValue;
        cmd.Parameters.AddWithValue("@naslov", SqlDbType.NVarChar).Value = txtNaslov.Text;
        cmd.Parameters.AddWithValue("@autor", SqlDbType.NVarChar).Value = txtAutor.Text;
        cmd.Parameters.AddWithValue("@zanr", SqlDbType.NVarChar).Value = txtZanr.Text;

        if (azuriraj)
        {
            cmd.CommandText = @"update Predstava
                                set glumacID = @glumacID, rediteljID = @rediteljID, naslov = @naslov, autor = @autor, zanr = @zanr
                                where predstavaID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Predstava
                                values (@glumacID, @rediteljID, @naslov, @autor, @zanr)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos odredjenih vrednosti nije validan.");
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.18 - Sadržaj metode *BtnSacuvaj_Click* za unos nove predstave



Slika 6.19 – Dizajn forme za dodavanje/izmenu reditelja

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Reditelj"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Reditelj" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-273,-35,210" />
        <Image Source="C:\Users\Legion 5 15\Desktop\rrr.png" Margin="0,254,278,-89" />
        <Label x:Name="lblIme" Content="Ime:" HorizontalAlignment="Left" Margin="162,110,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" />
        <Label x:Name="lblPrezime" Content="Prezime:" HorizontalAlignment="Left" Margin="117,195,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" />
        <Label x:Name="lblGodine" Content="Godine:" HorizontalAlignment="Left" Margin="125,275,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" />
        <TextBox x:Name="txtIme" HorizontalAlignment="Left" Margin="251,118,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="150" />
        <TextBox x:Name="txtPrezime" HorizontalAlignment="Left" Margin="251,199,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="150" />
        <TextBox x:Name="txtGodine" HorizontalAlignment="Left" Margin="251,275,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="150" />
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="277,399,0,0" VerticalAlignment="Top" FontSize="24" />
        <Button x:Name="btnOtkazi" Content="Otkaži" HorizontalAlignment="Left" Margin="277,489,0,0" VerticalAlignment="Top" FontSize="24" />
    </Grid>
</Window>

```

Slika 6.20 - XAML kod forme za dodavanje/izmenu reditelja

```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.AddWithValue("@ime", SqlDbType.NVarChar).Value = txtIme.Text;
        cmd.Parameters.AddWithValue("@prezime", SqlDbType.NVarChar).Value = txtPrezime.Text;
        cmd.Parameters.AddWithValue("@godine", SqlDbType.Int).Value = int.Parse(txtGodine.Text);
        if (azuriraj)
        {
            cmd.Parameters.AddWithValue("@id", SqlDbType.Int).Value = red["ID"];
            cmd.CommandText = @"update Reditelj
                                set ime = @ime, prezime = @prezime, godine = @godine
                                where rediteljID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Reditelj
                                values (@ime, @prezime, @godine)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos odredjenih vrednosti nije validan.");
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.21 - Sadržaj metode BtnSacuvaj_Click za unos novog reditelja



Slika 6.22 – Dizajn forme za dodavanje/izmenu ulaznica

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Ulaznica"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Ulaznica" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-273,-35,210" />
        <Image Source="C:\Users\Legion 5 15\Desktop\karte.png" Margin="10,373,398,-99" />
        <Label x:Name="lblGledalac" Content="Gledalac:" HorizontalAlignment="Left" Margin="137,112,0,0" VerticalAlignment="Top" FontSize="24" />
        <Label x:Name="lblPozoriste" Content="Pozorište:" HorizontalAlignment="Left" Margin="135,189,0,0" VerticalAlignment="Top" FontSize="24" />
        <Label x:Name="lblIzvedba" Content="Izvedba:" HorizontalAlignment="Left" Margin="144,275,0,0" VerticalAlignment="Top" FontSize="24" />
        <ComboBox x:Name="cmbGledalac" HorizontalAlignment="Left" Margin="275,111,0,0" VerticalAlignment="Top" Width="138" FontSize="24" />
        <ComboBox x:Name="cmbPozoriste" HorizontalAlignment="Left" Margin="275,189,0,0" VerticalAlignment="Top" Width="138" FontSize="24" />
        <ComboBox x:Name="cmbIzvedba" HorizontalAlignment="Left" Margin="275,274,0,0" VerticalAlignment="Top" Width="138" SelectionChanged="cmbIzvedba_SelectionChanged" />
        <Label x:Name="lblCena" Content="Cena:" HorizontalAlignment="Left" Margin="172,357,0,0" VerticalAlignment="Top" FontSize="24" />
        <Label x:Name="lblBrojSedista" Content="Broj sedišta:" HorizontalAlignment="Left" Margin="111,433,0,0" VerticalAlignment="Top" FontSize="24" />
        <TextBox x:Name="txtCena" HorizontalAlignment="Left" Margin="275,359,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="138" />
        <TextBox x:Name="txtBrojSedista" HorizontalAlignment="Left" Margin="275,434,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="138" />
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="175,549,0,0" VerticalAlignment="Top" FontSize="24" />
        <Button x:Name="btnOtkazi" Content="Otkazi" HorizontalAlignment="Left" Margin="366,549,0,0" VerticalAlignment="Top" FontSize="24" />
    </Grid>
</Window>

```

Slika 6.23 - XAML kod forme za dodavanje/izmenu ulaznica

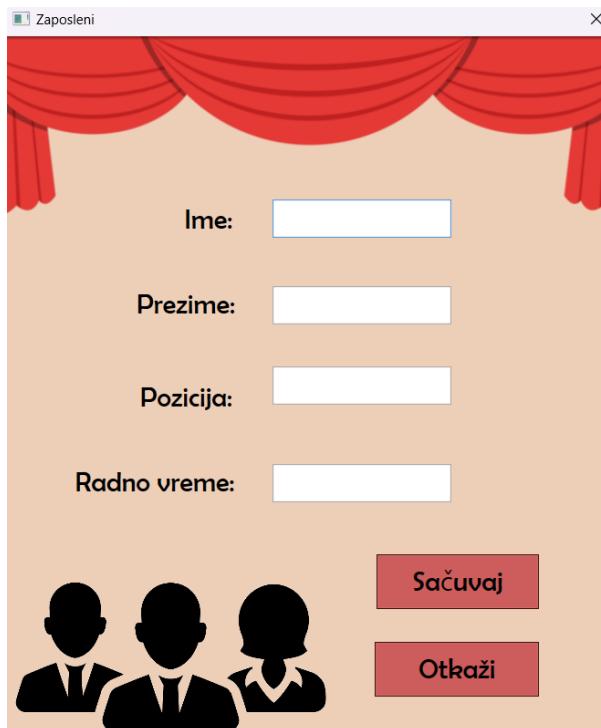
```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@gledalacID", SqlDbType.Int).Value = (int)cmbGledalac.SelectedValue;
        cmd.Parameters.Add("@pozoristeID", SqlDbType.Int).Value = (int)cmbPozoriste.SelectedValue;
        cmd.Parameters.Add("@izvedbaID", SqlDbType.Int).Value = (int)cmbIzvedba.SelectedValue;
        cmd.Parameters.Add("@cena", SqlDbType.Int).Value = int.Parse(txtCena.Text);
        cmd.Parameters.Add("@brojSedista", SqlDbType.Int).Value = int.Parse(txtBrojSedista.Text);

        if (azuriraj)
        {
            cmd.Parameters.Add("@id", SqlDbType.Int).Value = red["ID"];
            cmd.CommandText = @"update Ulaznica
                                set gledalacID = @gledalacID, pozoristeID = @pozoristeID, izvedbaID = @izvedbaID, cena = @cena, brojSedista = @brojSedista
                                where ulaznicaID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Ulaznica
                                values (@gledalacID, @pozoristeID, @izvedbaID, @cena, @brojSedista)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.24 - Sadržaj metode `BtnSacuvaj_Click` za unos novog ulaznika



Slika 6.25 – Dizajn forme za dodavanje/izmenu zaposlenih

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.Forme.Zaposleni"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste.Forme"
    mc:Ignorable="d"
    Title="Zaposleni" Height="650" Width="550" ResizeMode="NoResize">
    <Grid Background="#FFEDCFB8">
        <Image Source="C:\Users\Legion 5 15\Desktop\staff.png" Margin="10,325,254,-151" />
        <Image Source="C:\Users\Legion 5 15\Desktop\zavesa.png" Margin="-35,-264,-35,201" />
        <Label x:Name="lblIme" Content="Ime:" HorizontalAlignment="Left" Margin="153,142,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" Foreground="Black" />
        <Label x:Name="lblPrezime" Content="Prezime:" HorizontalAlignment="Left" Margin="111,217,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" Foreground="Black" />
        <Label x:Name="lblPozicija" Content="Pozicija:" HorizontalAlignment="Left" Margin="114,298,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" Foreground="Black" />
        <TextBox x:Name="txtIme" HorizontalAlignment="Left" Margin="235,143,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="150" FontSize="24" FontFamily="Times New Roman" />
        <TextBox x:Name="txtPrezime" HorizontalAlignment="Left" Margin="235,219,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="150" FontSize="24" FontFamily="Times New Roman" />
        <TextBox x:Name="txtPozicija" HorizontalAlignment="Left" Margin="235,0,0,0" TextWrapping="Wrap" Text="" VerticalAlignment="Center" Width="150" FontSize="24" FontFamily="Times New Roman" />
        <Label x:Name="lblRadnoVreme" Content="Radno vreme:" HorizontalAlignment="Left" Margin="57,373,0,0" VerticalAlignment="Top" FontSize="24" FontFamily="Times New Roman" />
        <TextBox x:Name="txtRadnoVreme" HorizontalAlignment="Left" Margin="235,375,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Text="" Width="150" FontSize="24" FontFamily="Times New Roman" />
        <Button x:Name="btnSacuvaj" Content="Sačuvaj" HorizontalAlignment="Left" Margin="326,454,0,0" VerticalAlignment="Top" FontSize="24" Height="40" Width="150" FontFamily="Times New Roman" />
        <Button x:Name="btnOtkazi" Content="Otkazi" HorizontalAlignment="Left" Margin="325,531,0,0" VerticalAlignment="Top" FontSize="24" Height="40" Width="150" FontFamily="Times New Roman" />
    </Grid>
</Window>

```

Slika 6.26 - XAML kod forme za dodavanje/izmenu zaposleni

```

private void btnSacuvaj_Click(object sender, RoutedEventArgs e)
{
    try
    {
        konekcija.Open();
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@ime", SqlDbType.NVarChar).Value = txtIme.Text;
        cmd.Parameters.Add("@prezime", SqlDbType.NVarChar).Value = txtPrezime.Text;
        cmd.Parameters.Add("@pozicija", SqlDbType.NVarChar).Value = txtPozicija.Text;
        cmd.Parameters.Add("@radnoVreme", SqlDbType.Int).Value = int.Parse(txtRadnoVreme.Text);
        if (azuriraj)
        {
            cmd.Parameters.Add("@id", SqlDbType.Int).Value = red["ID"];
            cmd.CommandText = @"update Zaposleni
                                set ime = @ime, prezime = @prezime, pozicija = @pozicija, radnoVreme = @radnoVreme
                                where zaposleniID = @id";
            red = null;
        }
        else
        {
            cmd.CommandText = @"insert into Zaposleni
                                values (@ime, @prezime, @pozicija, @radnoVreme)";
        }
        cmd.ExecuteNonQuery();
        cmd.Dispose();
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Unos odredjenih vrednosti nije validan.");
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}

```

Slika 6.27 - Sadržaj metode BtnSacuvaj_Click za unos novog zaposlenog

Tabele *tblIzvedba*, *tblPozoriste*, *tblPredstava* i *tblUlaznica* u sebi sadrže strane ključeve drugih tabela na osnovu kojih se pristupa sadržajima tih tabela. U samoj aplikaciji, potrebno je korisniku obezbediti vizuelni prikaz željenih obeležja neke od povezanih tabela. U WPF tehnologiji se to postiže korišćenjem *ComboBox-a*.

```

14 references
class Popunjavanje
{
    14 references
    public static void Popuni(ComboBox cb, string selectUpit)
    {
        Konekcija kon = new Konekcija();
        SqlConnection konekcija = new SqlConnection();
        try
        {
            konekcija = kon.KreirajKonekciju();
            konekcija.Open();

            SqlDataAdapter da = new SqlDataAdapter(selectUpit, konekcija);
            DataTable dt = new DataTable();
            da.Fill(dt);
            if (cb != null)
            {
                cb.ItemsSource = dt.DefaultView;
            }
            da.Dispose();
            dt.Dispose();
        }
        catch (SqlException ex) { MessageBox.Show(ex.Message, "Greska", MessageBoxButtons.OK, MessageBoxIcon.Error); }
        finally
        {
            if (konekcija != null)
                konekcija.Close();
        }
    }
}

```

Listing 6.28 – Kod klase Popunjavanje

Konstruktor ove forme u sebi sadrži metodu **Popuni** koju pozivamo nad prethodno pomenutoj klasi *Popunjavanje* (Listing 6.28) kako bi prikupili podatke za *ComboBox*-ove , a zatim ih i popunili. Na isti način je realizovan i konstruktor sa parametrima za ažuriranje liste.

```

1 reference
public Ulaznica()
{
    InitializeComponent();
    Popunjavanje.Popuni(cmbGledalac, gledalac);
    cmbGledalac.Focus();
    Popunjavanje.Popuni(cmbPozoriste, pozoriste);
    cmbPozoriste.Focus();
    Popunjavanje.Popuni(cmbIzvedba, izvedba);
    konekcija = kon.KreirajKonekciju();
}

1 reference
public Ulaznica(bool azuriraj, DataRowView red)
{
    this.azuriraj = azuriraj;
    this.red = red;
    InitializeComponent();
    Popunjavanje.Popuni(cmbGledalac, gledalac);
    Popunjavanje.Popuni(cmbPozoriste, pozoriste);
    Popunjavanje.Popuni(cmbIzvedba, izvedba);
}

```

Listing 6.29 - Kod konstruktora Ulaznica sa popunjavanjem ComboBox-ova

```

public Predstava()
{
    InitializeComponent();
    Popunjavanje.Popuni(cmbGlumac, glumac);
    cmbGlumac.Focus();
    Popunjavanje.Popuni(cmbReditelj, reditelj);
    cmbReditelj.Focus();
    konekcija = kon.KreirajKonekciju();
}

1 reference
public Predstava(bool azuriraj, DataRowView red)
{
    this.azuriraj = azuriraj;
    this.red = red;
    InitializeComponent();
    Popunjavanje.Popuni(cmbGlumac, glumac);
    Popunjavanje.Popuni(cmbReditelj, reditelj);
}

```

Listing 6.30 - Kod konstruktora Predstava sa popunjavanjem ComboBox-ova

```

public Pozoriste()
{
    InitializeComponent();
    Popunjavanje.Popuni(cmbZaposleni, zaposleni);
    cmbZaposleni.Focus();
    konekcija = kon.KreirajKonekciju();
}

1 reference
public Pozoriste(bool azuriraj, DataRowView red)
{
    this.azuriraj = azuriraj;
    this.red = red;
    InitializeComponent();
    Popunjavanje.Popuni(cmbZaposleni, zaposleni);
}

```

Listing 6.31 - Kod konstruktora Pozoriste sa popunjavanjem ComboBox-ova

```

public Izvedba()
{
    InitializeComponent();
    Popunjavanje.Popuni(cmbPredstava, predstava);
    cmbPredstava.Focus();
    konekcija = kon.KreirajKonekciju();
}

1 reference
public Izvedba(bool azuriraj, DataRowView red)
{
    this.azuriraj = azuriraj;
    this.red = red;
    InitializeComponent();
    Popunjavanje.Popuni(cmbPredstava, predstava);
}

```

Listing 6.32 - Kod konstruktora Izvedba sa popunjavanjem ComboBox-ova

Nakon kreiranja formi za svaku od tabele, potrebno je nešto slično napraviti i za glavni prozor (*eng. Main Window*). Glavni prozor predstavlja prvi, inicijalni, kontakt korisnika sa aplikacijom i ujedno i osnovni 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 osam kreiranih tabela u bazi, čijim će se klikom tastera miša, na sredini forme isčitavati ti podaci. Takođe, klikom tastera miša na dugme određene tabele, potrebno je da se korisniku obezbede dugmići za dodavanje, izmenu i brisanje prikazanog sadržaja. Dodavanjem odgovarajućih funkcionalnosti tim dugmićima u okviru glavne forme, obezbediće se izvršavanje osnovnih CRUD (*eng. Create, Read, Update, Delete*) operacija, što je i bio cilj ovog zadatka.



Slika 6.33 – Vizuelni prikaz glavnog prozora

```

<Window x:Class="IT28G2022_SkoricVanja_Pozoriste.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:IT28G2022_SkoricVanja_Pozoriste"
    mc:Ignorable="d"
    Title="Vanja Škorić_IT28/2022" Height="450" Width="800" ResizeMode="NoResize">
    <Grid Margin="0,0,0,5">
        <Image Source="C:\Users\Legion 5 15\Desktop\maske.png" Margin="610,277,-8,-4" RenderTransformOrigin="0.5,0.5" >
            <Image.RenderTransform>
                <TransformGroup>
                    <ScaleTransform/>
                    <SkewTransform/>
                    <RotateTransform Angle="20.216"/>
                    <TranslateTransform/>
                </TransformGroup>
            </Image.RenderTransform>
        </Image>
        <Image Source="C:\Users\Legion 5 15\Desktop\maske.png" Margin="11,283,621,-14" RenderTransformOrigin="0.5,0.5" >
            <Image.RenderTransform>
                <TransformGroup>
                    <ScaleTransform/>
                    <SkewTransform/>
                    <RotateTransform Angle="-23.661"/>
                    <TranslateTransform/>
                </TransformGroup>
            </Image.RenderTransform>
        </Image>
        <Image Source="C:\Users\Legion 5 15\Desktop\dobrodoslii.png" Margin="314,12,311,307" RenderTransformOrigin="0.5,0.5" >
            <Image.RenderTransform>
                <TransformGroup>
                    <ScaleTransform/>
                    <SkewTransform/>
                    <RotateTransform Angle="-0.332"/>
                    <TranslateTransform/>
                </TransformGroup>
            </Image.RenderTransform>
        </Image>
    </Grid>

```

Listing 6.34.1 – XAML kod centralnog prozora i StackPanela glavnog prozora

```

<Image>
    <DataGrid Name="dataGridCentralni" HorizontalAlignment="Center" VerticalAlignment="Center" Height="auto" Width="auto" IsReadOnly="True"
        AlternatingRowBackground="# IndianRed" CanUserAddRows="False" SelectionUnit="FullRow" SelectionMode="Single" FontSize="18" FontFamily="Bahnschrift">
        <StackPanel HorizontalAlignment="Center" Height="30" VerticalAlignment="Top" Width="auto" Orientation="Horizontal">
            <Button x:Name="btnPozoriste" Content="Pozoriste" Height="30" VerticalAlignment="Top" Width="100" Click="btnPozoriste_Click" FontFamily="Berlin Sans FB" Background="# FFEDCFB8" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnPredstava" Content="Predstava" Height="30" VerticalAlignment="Top" Width="100" Click="btnPredstava_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnIzvedba" Content="Izvedba" Height="30" VerticalAlignment="Top" Width="100" Click="btnIzvedba_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnGlumac" Content="Glumac" Height="30" VerticalAlignment="Top" Width="100" Click="btnGlumac_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnReditelj" Content="Reditelj" Height="30" VerticalAlignment="Top" Width="100" Click="btnReditelj_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnGledalac" Content="Gledalac" Height="30" VerticalAlignment="Top" Width="100" Click="btnGledalac_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnZaposleni" Content="Zaposleni" Height="30" VerticalAlignment="Top" Width="100" Click="btnZaposleni_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnUlaznica" Content="Ulaznica" Height="30" VerticalAlignment="Top" Width="100" Click="btnUlaznica_Click" Background="# FFEDCFB8" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
        </StackPanel>
        <StackPanel HorizontalAlignment="Center" Height="30" VerticalAlignment="Bottom" Width="360" Orientation="Horizontal">
            <Button x:Name="btnDodaj" Content="Dodaj" Width="121" Click="btnDodaj_Click" Background="# FFAF4038" BorderBrush="# FFEDCFB8" Foreground="# FF2D0A0A" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnIzmeni" Content="Izmeni" Width="121" Click="btnIzmeni_Click" Background="# FFAF4038" BorderBrush="# FFEDCFB8" Foreground="# FF2D0A0A" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
            <Button x:Name="btnObrisni" Content="Obrisni" Width="120" Click="btnObrisni_Click" Background="# FFAF4038" BorderBrush="# FFEDCFB8" Foreground="# FF2D0A0A" FontFamily="Berlin Sans FB" Foreground="# FFAF4038" BorderBrush="# FFEDCFB8" BorderThickness="1" Padding="5" Margin="5" />
        </StackPanel>
    <Grid.Background>
        <ImageBrush Stretch="None" ImageSource="C:\Users\Legion 5 15\Desktop\pozoriste.jpg" AlignmentY="Top" AlignmentX="Center" />
    </Grid.Background>

```

Listing 6.34.2 – XAML kod centralnog prozora i StackPanela glavnog prozora

Kako bi elementi unutar ovog prozora bili uredno raspoređeni, koristiće se poseban panel za njihovo smeštanje - *StackPanel*. U okviru gornjeg *StackPanel-a*, biće smešteni dugmići za izlistavanje tabela na centralnom gridu. Potrebno je podesiti poravnanja kao što je prikazano u listingu 6.34,2, a zatim izgenerisati metode koje će sprovoditi odgovarajuće akcije klikom tastera miša na to dugme.

U nastavku će biti prikazana *CodeBehind* sadržina glavnog prozora, gde je smešten najveći deo poslovne logike. U sledećem listingu prikazani su korišteni imenski prostori.

```

using IT28G2022_SkoricVanja_Pozoriste.Forme;
using System.Data;
using System.Data.SqlClient;
using System.Text;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

```

Listing 6.35 – Imenski prostori za glavni prozor

Za početak, sadržina glavnog prozora trebalo bi da izgleda kao u listingu 6.36. Sadržaće instancu konekcije ka bazi podataka, metodu za popunjavanje početnog grida pri pokretanju prozora, konstruktor za kreiranje novog prozora, kao i izgenerisane metode iz XAML koda, putem kojih će se, klikom tastera miša na odgovarajuće dugme, obavljati određene funkcionalnosti.

```

namespace IT28G2022_SkoricVanja_Pozoriste
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    2 references
    public partial class MainWindow : Window
    {
        private string ucitanaTabela;
        Konekcija kpn = new Konekcija();
        SqlConnection koneksiya = new SqlConnection();
        private bool azuriraj;
        private DataRowView red;

        #region Select upiti
        private static string gledalacSelect = @"select gledalacID as ID, ime as Ime, prezime as Prezime, pol as Pol from Gledalac";
        private static string glumacSelect = @"select glumacID as ID, ime as Ime, prezime as Prezime, godine as Godine, pol as Pol from [Glumac/ica]";
        private static string rediteljSelect = @"select rediteljID as ID, ime as Ime, prezime as Prezime, godine as Godine from Reditelj";
        private static string zaposleniSelect = @"select zaposleniID as ID, ime as Ime, prezime as Prezime, pozicija as Pozicija, radnoVreme as 'Radno vreme' from Zaposleni";
        private static string izvedbaSelect = @"select izvedbaID as ID, vremePocetka as 'Vreme pocetka' from Izvedba join Predstava on Izvedba.predstavaID=Predstava.predstavaID";
        private static string pozoristeSelect = @"select pozoristeID as ID, naziv as Naziv, lokacija as Lokacija, direktor as Direktor from Pozoriste join Zaposleni on Pozoriste.zaposleniID=Zaposleni.zaposleniID";
        private static string predstavaSelect = @"select predstavaID as ID, naslov as Naslov, autor as Autor, zanr as Zanr from Predstava join [Glumac/ica] on Predstava.glumacID=[Glumac/ica].glumacID
            join Reditelj on Predstava.rediteljID=Reditelj.rediteljID";
        private static string ulaznicaSelect = @"select ulaznicaID as ID, cena as Cena, brojSedista as 'Broj sedista' from Ulaznica join Gledalac on Ulaznica.gledalacID=Gledalac.gledalacID
            join Pozoriste on Ulaznica.pozoristeID=Pozoriste.pozoristeID
            join Izvedba on Ulaznica.izvedbaID=Izvedba.izvedbaID";
        #endregion
    }
}

```

Listing 6.36.1 – Sadržaj CodeBehind-a glavnog prozora

```

#region Select sa uslovom
private static string selectUslovGledalac = @"select * from Gledalac where gledalacID=";
private static string selectUslovGlumac = @"select * from [Glumac/ica] where glumacID=";
private static string selectUslovReditelj = @"select * from Reditelj where rediteljID=";
private static string selectUslovZaposleni = @"select * from Zaposleni where zaposleniID=";
private static string selectUslovIzvedba = @"select * from Izvedba where izvedbaID=";
private static string selectUslovPozoriste = @"select * from Pozoriste where pozoristeID=";
private static string selectUslovPredstava = @"select * from Predstava where predstavaID=";
private static string selectUslovUlaznica = @"select * from Ulaznica where ulaznicaID=";
#endregion

#region Delete naredbe
private static string gledalacDelete = @"delete from Gledalac where gledalacID=";
private static string glumacDelete = @"delete from Glumac where glumacID=";
private static string rediteljDelete = @"delete from Reditelj where rediteljID=";
private static string zaposleniDelete = @"delete from Zaposleni where zaposleniID=";
private static string izvedbaDelete = @"delete from Izvedba where izvedbaID=";
private static string pozoristeDelete = @"delete from Pozoriste where pozoristeID=";
private static string predstavaDelete = @"delete from Predstava where predstavaID=";
private static string ulaznicaDelete = @"delete from Ulaznica where ulaznicaID=";

#endregion

0 references
public MainWindow()
{
    InitializeComponent();
    konekcija = kon.KreirajKonekciju();
    UcitajPodatke(gledalacSelect);
}

}

```

Listing 6.36.2 – Sadržaj CodeBehind-a glavnog prozora

```

#region Select upiti
private static string gledalacSelect = @"select gledalacID as ID, ime as Ime, prezime as Prezime, pol as Pol from Gledalac";
private static string glumacSelect = @"select glumacID as ID, ime as Ime, prezime as Prezime, godine as Godine, pol as Pol from [Glumac/ica]";
private static string rediteljSelect = @"select rediteljID as ID, ime as Ime, prezime as Prezime, godine as Godine from Reditelj";
private static string zaposleniSelect = @"select zaposleniID as ID, ime as Ime, prezime as Prezime, pozicija as Pozicija, radnoVreme as 'Radno vreme' from Zaposleni";
private static string izvedbaSelect = @"select izvedbaID as ID, vremePocetka as 'Vreme pocetka' from Izvedba join Predstava on Izvedba.predstavaID=Predstava.predstavaID";
private static string pozoristeSelect = @"select pozoristeID as ID, naziv as Naziv, lokacija as Lokacija, direktor as Direktor from Pozoriste join Zaposleni on Pozoriste.zaposleniID=Zaposleni.zaposleniID";
private static string predstavaSelect = @"select predstavaID as ID, naslov as Naslov, autor as Autor, zanr as Zanr from Predstava join [Glumac/ica] on Predstava.glumacID=[Glumac/ica].glumacID
join Reditelj on Predstava.rediteljID=Reditelj.rediteljID";
private static string ulaznicaSelect = @"select ulaznicaID as ID, cena as Cena, brojSedista as 'Broj sedista' from Ulaznica join Gledalac on Ulaznica.gledalacID=Gledalac.gledalacID
join Pozoriste on Ulaznica.pozoristeID=Pozoriste.pozoristeID
join Izvedba on Ulaznica.izvedbaID=Izvedba.izvedbaID";

#endregion

```

Listing 6.37 – SQL kod Select upita

```

#region Select sa uslovom
private static string selectUslovGledalac = @"select * from Gledalac where gledalacID=";
private static string selectUslovGlumac = @"select * from [Glumac/ica] where glumacID=";
private static string selectUslovReditelj = @"select * from Reditelj where rediteljID=";
private static string selectUslovZaposleni = @"select * from Zaposleni where zaposleniID=";
private static string selectUslovIzvedba = @"select * from Izvedba where izvedbaID=";
private static string selectUslovPozoriste = @"select * from Pozoriste where pozoristeID=";
private static string selectUslovPredstava = @"select * from Predstava where predstavaID=";
private static string selectUslovUlaznica = @"select * from Ulaznica where ulaznicaID=";

#endregion

```

Listing 6.38 – SQL kod Select upita sa uslovom

```

#region Delete naredbe
private static string gledalacDelete = @"delete from Gledalac where gledalacID=";
private static string glumacDelete = @"delete from Glumac where glumacID=";
private static string rediteljDelete = @"delete from Reditelj where rediteljID=";
private static string zaposleniDelete = @"delete from Zaposleni where zaposleniID=";
private static string izvedbaDelete = @"delete from Izvedba where izvedbaID=";
private static string pozoristeDelete = @"delete from Pozoriste where pozoristeID=";
private static string predstavaDelete = @"delete from Predstava where predstavaID=";
private static string ulaznicaDelete = @"delete from Ulaznica where ulaznicaID=";

#endregion

```

Listing 6.39 – SQL kod Delete naredbi

U prethodna tri listinga je prikazana komunikacija sa bazom podataka.

Nakon instanciranja konekcije, sledeća stvar je kreiranje metode koja će popunjavati početni DataGrid pri kreiranju novog prozora. U ovom primeru, tabela *tblGledalac* će biti podrazumeumevana tabela, odnosno tabela čiji će se prikaz generisati pri pokretanju ovog prozora. U ovom slučaju, nakon otvaranja konekcije, prikazani upit se kreira na osnovu povezivanja pet tabela iz baze podataka. Zatim, na isti način kao kod popunjavanja ComboBox- ova pri instanciranju prozora neke od formi, ovi podaci se smeštaju u odgovarajući *DataSet* kojeg zatim popunjava *SqlDataAdapter*. U konstruktoru se poziva ova metoda i prosleđuje joj se select upit za tabelu *tblGledalac*.

```
private void UcitajPodatke(string selectUpit)
{
    try
    {
        konekcija.Open();
        SqlDataAdapter dataAdapter = new SqlDataAdapter(selectUpit, konekcija);
        DataTable dataTable = new DataTable();
        dataAdapter.Fill(dataTable);
        if (dataGridCentralni != null)
        {
            dataGridCentralni.ItemsSource = dataTable.DefaultView;
        }
        ucitanaTabela = selectUpit;
        dataAdapter.Dispose();
        dataTable.Dispose();
    }
    catch (SqlException)
    {
        MessageBox.Show("Neuspesno ucitani podaci", "Greska", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (konekcija != null)
        {
            konekcija.Close();
        }
    }
}
```

Listing 6.40 – Sadržaj metode UcitajPodatke

Sledi pregled svake od posebnih metoda, koje se pozivaju nakon klika tasterom miša na odgovarajuće dugme iz gornjeg *StackPanel-a*, koje služe za ispis podataka iz tabele na centralnom gridu. Na osnovu prethodno kreirane metode *UcitajPodatke* i select upita, za sve druge metode, koje predstavljaju odgovor na klik korisnika, poziva se upravo ova metoda i prosleđuje joj se odgovarajući upit (listing 6.41).

```

1 reference
private void btnPozoriste_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(pozoristeSelect);
}

1 reference
private void btnPredstava_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(predstavaSelect);
}

1 reference
private void btnIzvedba_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(izvedbaSelect);
}

1 reference
private void btnGlumac_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(glumacSelect);
}

1 reference
private void btnReditelj_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(rediteljSelect);
}

1 reference
private void btnGledalac_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(gledalacSelect);
}

1 reference
private void btnZaposleni_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(zaposleniSelect);
}

1 reference
private void btnUlaznica_Click(object sender, RoutedEventArgs e)
{
    UcitajPodatke(ulaznicaSelect);
}

```

Listing 6.41 – Prikaz svih metoda koje učitavaju podatke

Sadržaj metoda generisanih na klik tasterom miša na dugme Dodaj je poprilično jednostavan. Klikom tastera miša na dugme Dodaj, instancira se prozor odgovarajuće forme i prikazuje korisniku na ekranu. Nakon zatvaranja tog prozora forme, osvežava se centralni grid sa aktuelnim podacima nakon eventualnog dodavanja novog zapisa u bazu. U listingu 6.42.1 i 6.42.2 prikazan je sadržaj metode za dodavanje nove torke u odgovarajuću tabelu u bazi podataka.

```

private void btnDodaj_Click(object sender, RoutedEventArgs e)
{
    Window prozor;

    if (ucitanaTabela.Equals(zaposleniSelect))
    {
        prozor = new Zaposleni();
        prozor.ShowDialog();
        UcitajPodatke(zaposleniSelect);
    }
    else if (ucitanaTabela.Equals(rediteljSelect))
    {
        prozor = new Reditelj();
        prozor.ShowDialog();
        UcitajPodatke(rediteljSelect);
    }
    else if (ucitanaTabela.Equals(gledalacSelect))
    {
        prozor = new Gledalac();
        prozor.ShowDialog();
        UcitajPodatke(gledalacSelect);
    }
    else if (ucitanaTabela.Equals(glumacSelect))
    {
        prozor = new Glumac();
        prozor.ShowDialog();
        UcitajPodatke(glumacSelect);
    }
    else if (ucitanaTabela.Equals(pozoristeSelect))
    {
        prozor = new Pozoriste();
        prozor.ShowDialog();
        UcitajPodatke(pozoristeSelect);
    }
}

```

Listing 6.42.1 – Sadržaj metode BtnDodaj_Click

```

else if (ucitanaTabela.Equals(izvedbaSelect))
{
    prozor = new Izvedba();
    prozor.ShowDialog();
    UcitajPodatke(izvedbaSelect);
}

else if (ucitanaTabela.Equals(ulaznicaSelect))
{
    prozor = new Ulaznica();
    prozor.ShowDialog();
    UcitajPodatke(ulaznicaSelect);
}

else if (ucitanaTabela.Equals(predstavaSelect))
{
    prozor = new Predstava();
    prozor.ShowDialog();
    UcitajPodatke(predstavaSelect);
}
}

```

Listing 6.42.2 – Sadržaj metode BtnDodaj_Click

Brisanje podataka iz baze podataka obavlja se putem klika tasterom miša na dugme Obriši na način prikazan u listingu 6.43. Nakon otvaranja konekcije, kreira se objekat klase *DataRowView* koji u sebi čuva informaciju o indeksu reda koji je korisnik prethodno označio. Nakon toga kreira se *Delete SQL* komanda putem koje će se u samoj bazi obrisati željena stavka. Pre samog izvršavanja komande, pred korisnikom se pojavljuje dijalog prozor (*MessageBox*) putem kojeg on potvrđuje ili odustaje od brisanja. Dva osnovna izuzetka koja se ovde mogu javiti su: *ArgumentOutOfRangeException* i *SqlException*. Prvi se javlja pri pokušaju brisanja ukoliko nije označen nijedan red, a pritisnuto je dugme za brisanje, a drugi ukoliko se pokuša brisanje entiteta iz tabele koji je primarnim ključem povezan sa nekom drugom tabelom.

```
private void Brisanje(string selectUslov)
{
    try
    {
        konekcija.Open();
        var selectedRow = (DataRowView)dataGridCentralni.SelectedItem;
        object a = selectedRow.Row.ItemArray[0];
        int? id = (int?)a; //upitnik znaci da moze biti i null vrednost
        SqlCommand cmd = new SqlCommand { Connection = konekcija };
        cmd.Parameters.Add("@id", SqlDbType.Int).Value = id;
        cmd.CommandText = selectUslov + " @id";
        cmd.ExecuteNonQuery();
        cmd.Dispose();
    }
    catch (ArgumentOutOfRangeException){
        MessageBox.Show("Niste selektovali red!", "Greska", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    catch (SqlException)
    {
        MessageBox.Show("Ne mozete obrisati element koji se koristi u drugoj tabeli kao strani kljuc!",
            "Greska", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (konekcija != null)
            konekcija.Close();
    }
}
```

Listing 6.43 – Sadržaj metode Brisanje

Kada je tasterom miša kliknuto na dugme Obriši, u zavisnosti od toga koja tabela je trenutno učitana i koju stavku tabele je korisnik izabrao, metodi *Brisanje* se prosleđuju parametri i nakon toga se ponovo učitavaju podaci iz baze podataka, kako bi oni bili ažurni nakon brisanja označenog reda (listing 6.44).

```

private void btnObrisni_Click(object sender, RoutedEventArgs e)
{
    if (dataGridCentralni.SelectedItems.Count == 0)
    {
        MessageBox.Show("Morate selektovati red koji zelite da izmenite!",
                        "Greska", MessageBoxButton.OK, MessageBoxImage.Error);
    }
    else if (dataGridCentralni.SelectedItems.Count > 1)
    {
        MessageBox.Show("Selektujte samo 1 red!", "Greska", MessageBoxButton.OK,
                        MessageBoxImage.Error);
    }
    else
    {

        if (ucitanaTabela.Equals(zaposleniSelect))
        {
            Brisanje(zaposleniDelete);
            UcitajPodatke(zaposleniSelect);

        }
        else if (ucitanaTabela.Equals(pozoristeSelect))
        {
            Brisanje(pozoristeDelete);
            UcitajPodatke(pozoristeSelect);

        }
        else if (ucitanaTabela.Equals(predstavaSelect))
        {
            Brisanje(predstavaDelete);
            UcitajPodatke(predstavaSelect);

        }
    }
}

```

Listing 6.44 – Sadržaj metode BtnObrisni_Click

```

else if (ucitanaTabela.Equals(pozoristeSelect))
{
    Brisanje(pozoristeDelete);
    UcitajPodatke(pozoristeSelect);

}
else if (ucitanaTabela.Equals(predstavaSelect))
{
    Brisanje(predstavaDelete);
    UcitajPodatke(predstavaSelect);

}
else if (ucitanaTabela.Equals(rediteljSelect))
{
    Brisanje(rediteljDelete);
    UcitajPodatke(rediteljSelect);

}
else if (ucitanaTabela.Equals(gledalacSelect))
{
    Brisanje(gledalacDelete);
    UcitajPodatke(gledalacSelect);

}
else if (ucitanaTabela.Equals(glumacSelect))
{
    Brisanje(glumacDelete);
    UcitajPodatke(glumacSelect);

}
else if (ucitanaTabela.Equals(ulaznicaSelect))
{
    Brisanje(ulaznicaDelete);
    UcitajPodatke(ulaznicaSelect);

}
else if (ucitanaTabela.Equals(izvedbaSelect))
{
    Brisanje(izvedbaDelete);
    UcitajPodatke(izvedbaSelect);

}

```

Listing 6.45 – Sadržaj metode BtnObrisni_Click

Za modifikaciju podataka, kao i za dodavanje, potrebno je da postoji odgovarajuća forma u kojoj će korisnik menjati, odnosno unositi željene vrednosti. U ovom primeru, za svaku od tabele već je kreirana forma putem koje se novi zapisi upisuju u bazu podataka. Kako bi se izbeglo kreiranje novih formi specijalno za modifikaciju podataka, može se iskoristiti postojeća forma za dodavanje, s tim što će jedina razlika biti u tome koju akciju izvršava dugme Sačuvaj te forme kada korisnik tasterom miša klikne na njega. Na samom početku potrebno je, u okviru glavnog prozora deklarisati tri promenljive. Prva je tipa podatka *boolean* i koristiće se kao uslov koji će dugmetu Sačuvaj govoriti da li je potrebno podatke upisati u bazu podataka kao novi entitet ili izmeniti. Druga je tipa *DataRowView* i služi za pamćenje vrednosti označenog reda, kada iz glavnog prozora pređemo u neku drugu formu. Treća promenljiva, tipa podatka *string*, koja govorи koja je trenutno tabela učitana.

U naredna dva listinga prikazana je metoda koja se koristi za izmenu. Na samom početku, deklarisana promenljiva se postavlja na *true*. Nakon toga instancira se prozor njene forme i otvara konekciju. Pomoću *DataRowView* objekta, kao i kod brisanja, pamti se koji red je označen. Razlika je u tome što se sada, u drugu promenljivu (označen) dodaje ta vrednost kako bi se iskoristila u drugoj formi. Nakon toga se, pomoću select upita kojem se u *where* uslovu zadaje vrednost primarnog ključa zapisa koji se modifikuje, potrebni podaci smeštaju u odgovarajuća polja forme, pomoću *SqlDataReader*-a. Nakon toga, poziva se metoda prozora *ShowDialog* i on se prikazuje na ekranu sa popunjениm vrednostima traženog zapisa. U okviru ove metode, u *finally* bloku, konekcija se zatvara, a promenljiva ažuriraj, vraća na *false*, u slučaju da, nakon izmene, korisnik želi da doda novi zapis u bazu podataka.

```
private void PopuniFormu(string selectUslov)
{
    try
    {
        konekcija.Open();
        ažuriraj = true;
        red = (DataRowView)dataGridCentralni.SelectedItems[0];
        SqlCommand cmd = new SqlCommand
        {
            Connection = konekcija
        };
        cmd.Parameters.Add("@id", SqlDbType.Int).Value = red["ID"];
        cmd.CommandText = selectUslov + " @id";
        SqlDataReader citac = cmd.ExecuteReader();
        if (citac.Read())
        {
            if (ucitanaTabela.Equals(gledalacSelect))
            {
                Gledalac prozorGledalac = new Gledalac(ażuriraj, red);
                prozorGledalac.txtIme.Text = citac["ime"].ToString();
                prozorGledalac.txtPrezime.Text = citac["prezime"].ToString();
                prozorGledalac.txtPol.Text = citac["pol"].ToString();
                prozorGledalac.ShowDialog();
            }
            else if (ucitanaTabela.Equals(glumacSelect))
            {
                Glumac prozorGlumac = new Glumac(ażuriraj, red);
                prozorGlumac.txtIme.Text = citac["ime"].ToString();
                prozorGlumac.txtPrezime.Text = citac["prezime"].ToString();
                prozorGlumac.txtGodine.Text = citac["godine"].ToString();
                prozorGlumac.txtPol.Text = citac["pol"].ToString();
                prozorGlumac.ShowDialog();
            }
            else if (ucitanaTabela.Equals(izvedbaSelect))
            {
                Izvedba prozorIzvedba = new Izvedba(ażuriraj, red);
                prozorIzvedba.cmbPredstava.SelectedValue = citac["predstavaID"];
                prozorIzvedba.dtVremePocetka.SelectedDate = (DateTime)citac["vremePocetka"];
                prozorIzvedba.ShowDialog();
            }
            else if (ucitanaTabela.Equals(pozeristeSelect))
            {
                Pozeriste prozorPozeriste = new Pozeriste(ażuriraj, red);
                prozorPozeriste.cmbPredstava.SelectedValue = citac["predstavaID"];
                prozorPozeriste.dtVremePocetka.SelectedDate = (DateTime)citac["vremePocetka"];
                prozorPozeriste.ShowDialog();
            }
        }
        finally
        {
            konekcija.Close();
            ažuriraj = false;
        }
    }
}
```

Listing 6.46.1 – Sadržaj metode *PopuniFormu*

```

}
else if (ucitanaTabela.Equals(pozoristeSelect))
{
    Pozoriste prozorPozoriste = new Pozoriste(azuriraj, red);
    prozorPozoriste.cmbZaposleni.SelectedValue = citac["zaposleniID"];
    prozorPozoriste.txtNaziv.Text = citac["naziv"].ToString();
    prozorPozoriste.txtLokacija.Text = citac["lokacija"].ToString();
    prozorPozoriste.txtDirektor.Text = citac["direktor"].ToString();
    prozorPozoriste.ShowDialog();
}
else if (ucitanaTabela.Equals(predstavaSelect))
{
    Predstava prozorPredstava = new Predstava(azuriraj, red);
    prozorPredstava.cmbGlumac.SelectedValue = citac["glumacID"];
    prozorPredstava.cmbReditelj.SelectedValue = citac["rediteljID"];
    prozorPredstava.txtNaslov.Text = citac["naslov"].ToString();
    prozorPredstava.txtAutor.Text = citac["autor"].ToString();
    prozorPredstava.txtZanr.Text = citac["zanr"].ToString();
    prozorPredstava.ShowDialog();
}
else if (ucitanaTabela.Equals(rediteljSelect))
{
    Reditelj prozorReditelj = new Reditelj(azuriraj, red);
    prozorReditelj.txtIme.Text = citac["ime"].ToString();
    prozorReditelj.txtPrezime.Text = citac["prezime"].ToString();
    prozorReditelj.txtGodine.Text = citac["godine"].ToString();
    prozorReditelj.ShowDialog();
}
else if (ucitanaTabela.Equals(ulaznicaSelect))
{
    Ulaznica prozorUlaznica = new Ulaznica(azuriraj, red);
    prozorUlaznica.cmbGledalac.SelectedValue = citac["gledalacID"];
    prozorUlaznica.cmbPozoriste.SelectedValue = citac["pozoristeID"];
    prozorUlaznica.cmbIzvedba.SelectedValue = citac["izvedbaID"];
    prozorUlaznica.txtCena.Text = citac["cena"].ToString();
    prozorUlaznica.txtBrojSedista.Text = citac["brojSedista"].ToString();
    prozorUlaznica.ShowDialog();
}
else if (ucitanaTabela.Equals(zaposleniSelect))
{
    Zaposleni prozorZaposleni = new Zaposleni(azuriraj, red);
    prozorZaposleni.txtIme.Text = citac["ime"].ToString();
    prozorZaposleni.txtPrezime.Text = citac["prezime"].ToString();
}

```

Listing 6.46.2 – Sadržaj metode PopuniFormu

```

    prozorUlaznica.ShowDialog();
}
else if (ucitanaTabela.Equals(zaposleniSelect))
{
    Zaposleni prozorZaposleni = new Zaposleni(azuriraj, red);
    prozorZaposleni.txtIme.Text = citac["ime"].ToString();
    prozorZaposleni.txtPrezime.Text = citac["prezime"].ToString();
    prozorZaposleni.txtPozicija.Text = citac["pozicija"].ToString();
    prozorZaposleni.txtRadnoVreme.Text = citac["radnoVreme"].ToString();
    prozorZaposleni.ShowDialog();
}
}

```

Listing 6.46.3 – Sadržaj metode PopuniFormu

Kada Korisnik izabere stavku koju želi da izmeni, poziva se metoda *BtnIzmeni_Click* koja u zavisnosti od učitane tabele prosleđuje odgovarajući parametar metodi *PopuniFormu* i nakon izvršene izmene ponovo učitava podatke kako bi podaci bili ažurni.

```

private void btnIzmeni_Click(object sender, RoutedEventArgs e)
{
    if (ucitanaTabela.Equals(zaposleniSelect))
    {
        PopuniFormu(selectUslovZaposleni);
        UcitajPodatke(zaposleniSelect);
    }
    else if (ucitanaTabela.Equals(rediteljSelect))
    {
        PopuniFormu(selectUslovReditelj);
        UcitajPodatke(rediteljSelect);
    }
    else if (ucitanaTabela.Equals(gledalacSelect))
    {
        PopuniFormu(selectUslovGledalac);
        UcitajPodatke(gledalacSelect);
    }
    else if (ucitanaTabela.Equals(glumacSelect))
    {
        PopuniFormu(selectUslovGlumac);
        UcitajPodatke(glumacSelect);
    }
    else if (ucitanaTabela.Equals(pozoristeSelect))
    {
        PopuniFormu(selectUslovPozoriste);
        UcitajPodatke(pozoristeSelect);
    }
    else if (ucitanaTabela.Equals(izvedbaSelect))
    {
        PopuniFormu(selectUslovIzvedba);
        UcitajPodatke(izvedbaSelect);
    }
    else if (ucitanaTabela.Equals(ulaznicaSelect))
    {
        PopuniFormu(selectUslovUlaznica);
    }
}

```

Listing 6.47.1 – Sadržaj metode `BtnIzmeni_Click`

```

    }
    else if (ucitanaTabela.Equals(izvedbaSelect))
    {
        PopuniFormu(selectUslovIzvedba);
        UcitajPodatke(izvedbaSelect);
    }
    else if (ucitanaTabela.Equals(ulaznicaSelect))
    {
        PopuniFormu(selectUslovUlaznica);
        UcitajPodatke(ulaznicaSelect);
    }
    else if (ucitanaTabela.Equals(predstavaSelect))
    {
        PopuniFormu(selectUslovPredstava);
        UcitajPodatke(predstavaSelect);
    }
}

```

Listing 6.47.2 – Sadržaj metode `BtnIzmeni_Click`

7. Testiranje aplikacije

Kada je aplikacija kreirana i kada su implementirane sve neophodne funkcionalnosti koje ona treba da podrži, neophodno je testirati aplikaciju kako bi se proverila ispravnost tih funkcionalnosti. U ovom poglavlju dat je prikaz testiranja WPF aplikacije za podršku poslovanja rada pozorišta.

Pri pokretanju aplikacije, generiše se početni prozor kao na slici 7.1.



Slika 7.1 – Početni prozor pri pokretanju aplikacije i prikaz podataka iz tabele Gledalac

Klikom tastera miša na dugmiće iz centralnog menija vrši se prikaz podataka željene tabele na centralnom gridu.



Slika 7.2 – Prikaz podataka iz tabele Predstava

Klikom tastera miša na dugme **DODAJ**, otvara se forma za dodavanje novog zapisa u bazu podataka. U formi se klikom na taster **SAČUVAJ** sačuvavaju odnosno upisuju podaci u bazu i tabela se, nakon zatvaranja forme, automatski ažurira.

Slika 7.3 – Dodavanje zapisa u bazu podataka

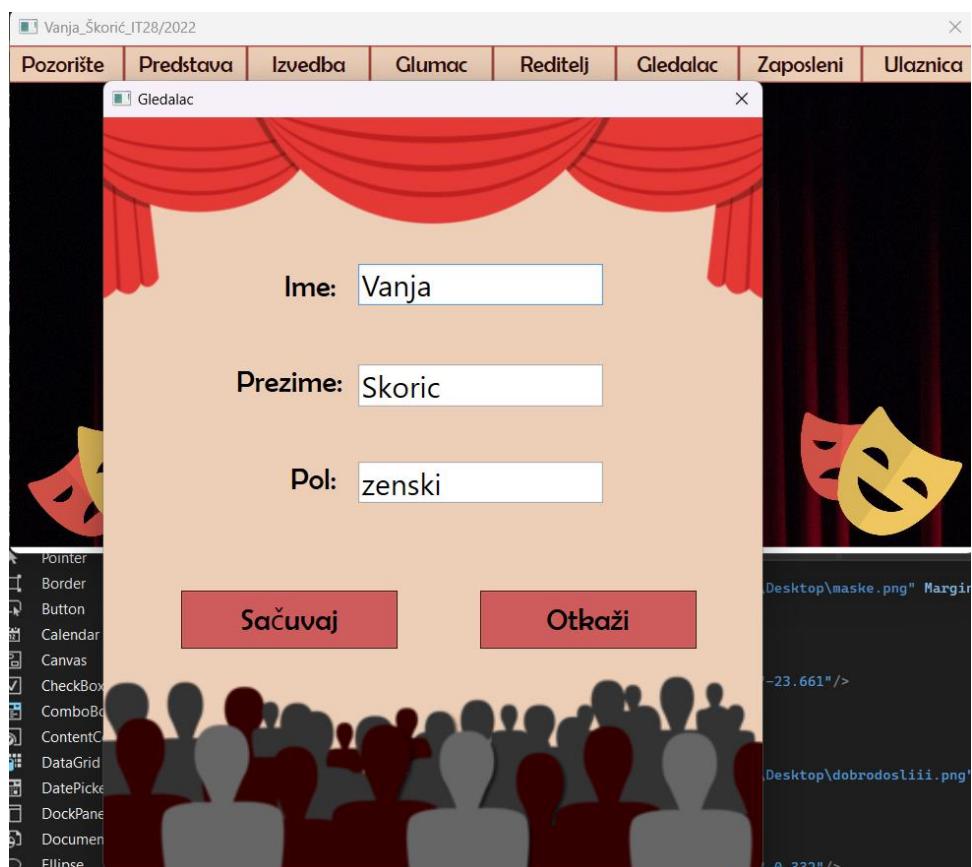
ID	Ime	Prezime	Pol
2	Milica	Misan	zenski
5	Miljana	Petkovic	zenski
8	Srdjan	Petrovic	muski
9	Vanja	Skoric	zenski
10	Stojan	Sladovic	muski
11	Marijana	Baric	zenski
13	Anja	Cojanovic	zenski
14	Marko	Markovic	muski

Slika 7.4 – Dodavanje zapisa u bazu podataka

Označavanjem odgovarajućeg reda otvara se forma sa popunjениm vrednostima prethodnog zapisa koje korisnik može modifikovati. Ukoliko Korisnik klikne na dugme **IZMENI**, a pritom nije izabrao red, onda će mu program ukazati na tu grešku.



Slika 7.5 – Prikaz greške prilikom neselektovanja reda



Slika 7.6 – Prikaz forme sa prethodno učitanim podacima

Brisanje podataka iz tabele se vrši tako što se prvo selektuje red, pa potom klikne na dugme **OBRIŠI**. Nakon klika na dugme **OBRIŠI** selektovano polje nestaje iz tabele (u ovom slučaju to je gledalac Marko Marković).

Vanja_Škorić_IT28/2022 X

Pozorište	Predstava	Izvedba	Glumac	Reditelj	Gledalac	Zaposleni	Ulaznica																																
Welcome!																																							
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID</th> <th>Ime</th> <th>Prezime</th> <th>Pol</th> </tr> </thead> <tbody> <tr><td>2</td><td>Milica</td><td>Misan</td><td>zenski</td></tr> <tr><td>5</td><td>Miljana</td><td>Petkovic</td><td>zenski</td></tr> <tr><td>8</td><td>Srdjan</td><td>Petrovic</td><td>muski</td></tr> <tr><td>9</td><td>Vanja</td><td>Skoric</td><td>zenski</td></tr> <tr><td>10</td><td>Stojan</td><td>Sladovic</td><td>muski</td></tr> <tr><td>11</td><td>Marijana</td><td>Baric</td><td>zenski</td></tr> <tr><td>13</td><td>Anja</td><td>Cojanovic</td><td>zenski</td></tr> </tbody> </table>								ID	Ime	Prezime	Pol	2	Milica	Misan	zenski	5	Miljana	Petkovic	zenski	8	Srdjan	Petrovic	muski	9	Vanja	Skoric	zenski	10	Stojan	Sladovic	muski	11	Marijana	Baric	zenski	13	Anja	Cojanovic	zenski
ID	Ime	Prezime	Pol																																				
2	Milica	Misan	zenski																																				
5	Miljana	Petkovic	zenski																																				
8	Srdjan	Petrovic	muski																																				
9	Vanja	Skoric	zenski																																				
10	Stojan	Sladovic	muski																																				
11	Marijana	Baric	zenski																																				
13	Anja	Cojanovic	zenski																																				
 																																							
Dodaj Izmeni Obriši																																							

Slika 7.7 - Prikaz tabele nakon obrisanog gledaoca Marka Markovića

8. Zaključak

Izradom ovog projekta stekla sam osnovno znanje o funkcionisanju poslovnih sistema, ponovila postojeće i stekla novo znanje što se tiče izrade baza podataka i naučila kako da kreiram WPF aplikacije. Rad na projektu mi je bio izuzetno zanimljiv jer sam povezala “posao” i moj, sada već decenijski, hobi – muziku, zbog koje godinama nastupam u raznim pozorištima, a isto tako ih i rado posećujem kao gledalac.

Veliku pažnju sam posvetila “opcijama” koje jedan korisnik aplikacije može da koristi, i smatram da sam obuhvatila sve neophodne, a možda čak i neke “dodatne opcije” koje bi svakom korisniku aplikacije bile itekako od pomoći pre posete samom pozorištu.