

Sadržaj

Uvod	1
1. Web-aplikacija za glazbene albume	2
1.1. Funkcionalni zahtjevi	2
1.1.1. Aktori i njihovi funkcionalni zahtjevi:	2
1.2. Ostali zahtjevi	3
2. Baza podataka.....	4
2.1. ER model	4
2.1.1. Opis ER modela.....	4
2.1.2. Entiteti	5
2.1.3. Veze	6
2.2. Relacijski model	7
2.2.1. Popis relacija	8
2.2.2. Primjeri create naredbi.....	18
2.2.3. Primjeri insert naredbi	21
2.2.4. Okidači	22
2.2.5. Uloge	24
3. Implementacija web-aplikacije.....	25
3.1. Arhitektura i dizajn sustava	25
3.2. Višeslojna arhitektura radnog okvira Spring.....	26
3.3. JavaScript biblioteka React	29
4. Korisničke upute.....	30
4.1. Neregistrirani/neprijavljeni korisnik	30
4.1.1. Početna stranica	30
4.1.2. Pregled vinyl ploča	30

4.1.3.	Pregled izvođača.....	33
4.1.4.	Pregled izdavačkih kuća	34
4.1.5.	Registracija i prijava.....	35
4.2.	Prijavljeni korisnik	36
4.2.1.	Profil	36
4.2.2.	Dodavanje ploča na popise	37
4.2.3.	Postavljanje recenzije	37
4.3.	Administrator.....	38
4.3.1.	Dodavanje sadržaja.....	38
4.3.2.	Brisanje sadržaja.....	39
	Zaključak	40
	Literatura	41
	Sažetak.....	42
	Summary.....	43

Uvod

Posljednjih nekoliko desetljeća svijet se u potpunosti digitalizirao, sve više stvari nam postaje dostupno u neopipljivom obliku. Kako se svijet postepeno prebacuje u jedinice i nule tako se pojavljuju ljubitelji onog „starog“, opipljivog. Vinyl ploče i gramofoni pali su u zaborav 90tih godina prošlog stoljeća kada ih zamjenjuju manji i kompaktniji uređaji za pohranu, no posljednjih nekoliko godina doživljavaju velik povratak.

Odlučila sam napraviti sustav koji bi omogućio ljubiteljima vinyla da na jednostavan i brz način mogu pregledavati i pretraživati svoje omiljene albume, izvođače i provjeriti izdanja različitih izdavačkih kuća. Sustav bi registriranim korisnicima omogućavao pohraniti i ažurirati podatke svojih kolekcija i lista želja te ostavljanje recenzija i ocjena za albume. Detaljni opis zahtjeva sustava može se pronaći u prvom poglavlju.

Izgradnja ovakvog sustava može se podijeliti na dva dijela. Prvo je potrebno izgraditi model baze podataka i popisati sve potrebne entitete i attribute za pohranu podataka, o čemu ću detaljnije govoriti u drugom poglavlju. Zatim je potrebno izgraditi samo korisničko sučelje koje treba biti intuitivno i jednostavno za korištenje te korisniku treba omogućiti brz i efikasan pregled relevantnih podataka iz baze. Treće poglavlje rada sadrži detaljniji opis same implementacije web-aplikacije, dok se u četvrtom poglavlju nalaze upute za korištenje.

Pri realizaciji sustava za izradu baze podataka koristit će se sustav PostgreSQL, a za izradu web-aplikacije programski jezik Java te Spring radni okvir.

1. Web-aplikacija za glazbene albume

1.1. Funkcionalni zahtjevi

1.1.1. Aktori i njihovi funkcionalni zahtjevi:

- Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - a. Pretraživati albume po imenu, žanru, vremenskom razdoblju i vrsti ploče
 - b. Pretraživati izvođače i izdavačke kuće po imenu
 - c. Vidjeti popis najtraženijih, najpopularnijih i najbolje ocijenjenih albuma
 - d. Vidjeti popis najvećih kolekcionara
 - e. Registrirati se
- Prijavljeni korisnik (inicijator) može:
 - a. Pretraživati albume po imenu, žanru, vremenskom razdoblju i vrsti ploče
 - b. Pretraživati izvođače i izdavačke kuće po imenu
 - c. Vidjeti popis najtraženijih, najpopularnijih i najbolje ocijenjenih albuma
 - d. Vidjeti popis najvećih kolekcionara
 - e. Pregledavati i mijenjati osobne podatke
 - f. Obrisati svoj profil
 - g. Dodati/ukloniti album iz svoje kolekcije/liste želja
 - h. Ocijeniti i recenzirati album
- Administrator (inicijator) može:
 - a. Pretraživati albume po imenu, žanru, vremenskom razdoblju i vrsti ploče
 - b. Pretraživati izvođače i izdavačke kuće po imenu
 - c. Vidjeti popis najtraženijih, najpopularnijih i najbolje ocijenjenih albuma
 - d. Vidjeti popis najvećih kolekcionara
 - e. Dodati nove albume, izvođače, izdavačke kuće u sustav

- f. Ukloniti albume, izvođače, izdavačke kuće iz sustava
- Baza podatka (sudionik) može:
 - a. Pohraniti/urediti/obrisati sve podatke o albumima, izvođačima i izdavačkim kućama
 - b. Pohraniti/urediti/obrisati sve podatke o registriranim korisnicima i njihovim popisima

1.2. Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje treba biti intuitivno i jednostavno za korištenje
- Neispravno korištenje sustava ne smije narušiti njegove funkcionalnosti
- Sustav treba omogućiti sigurnu pohranu korisničkih zaporki
- Običan korisnik ne bi smio biti u mogućnosti obaviti funkcije koje administrator može čak i kada se sustav pokušava koristiti na neispravan način, i obrnuto
- Nadogradnja sustava ne smije narušiti njegove postojeće funkcionalnosti

2. Baza podataka

2.1. ER model

Za ostvarivanje funkcionalnosti sustava prvo i najbitnije za lakši daljnji razvoj je definirati dobar model baze podataka. Najčešće se koristi ER model (engl. *Entity-Relationship model*). To je postrelacijski model koji se sastoji od entiteta opisanih atributima, a entiteti su međusobno povezani vezama [1]. Veze mogu biti u odnosu:

- 1:1 (jedan-prema-jedan)
- 1:N (jedan-prema-više)
- N:1 (više-prema-jedan)
- N:N (više-prema-više).

Ovakav model omogućuje eksplicitni prikaz veza koje u sebi sadrže važne semantičke informacije te jednostavno generiranje relacijskog modela podataka.

2.1.1. Opis ER modela

U ovom sustavu pohranjuju se informacije o vinyl pločama, izvođačima, izdavačkim kućama i korisnicima. Slika 2.1 prikazuje cijeli ER model.

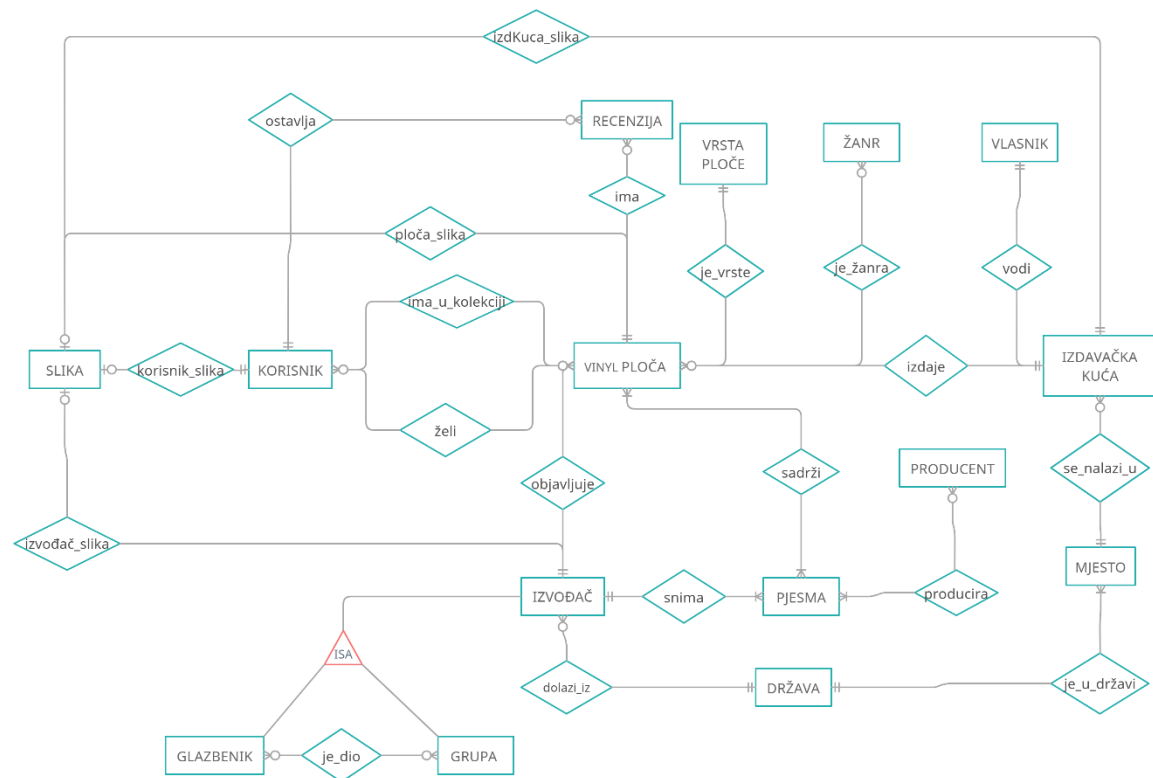
Korisnik je povezan vinyl pločom preko veza 'ima_u_kolekciji' i 'želi', svaki korisnik može imati više ploča u kolekciji ili na listi želja, svaka ploča može biti na popisima različitih korisnika. Korisnik može ostaviti recenziju i ocjenu za bilo koju ploču i preko veze 'korisnik_slika' može postaviti svoju profilnu fotografiju.

Vinyl ploča povezana je i s entitetom žanr i vrsta ploče koji nam govori o kakvoj ploči je riječ. Ploču izdaje izdavačka kuća, a objavljuje ju izvođač, također sadrži i popis pjesma i može imati postavljenu naslovnu fotografiju. Izdavačka kuća također može imati postavljenu fotografiju, ima svog vlasnika te se nalazi u nekom mjestu.

Jedan od važnijih entiteta u ovoj bazi je naravno i izvođač koji objavljuje vinyl ploče. Izvođač može biti glazbenik ili grupa, glazbenik isto tako može biti jedan od sudionika neke

grupe. Izvođač može snimiti jednu ili više pjesma, ima svoju državu porijekla i može imati postavljenu fotografiju.

Kao što smo rekli pjesma je povezana s vinyl pločom preko veze 'sadrži' gdje se još definira i redni broj pjesme na albumu. Pjesmu producira jedan ili više producenata.



Slika 2.1 ER model baze podataka

2.1.2. Entiteti

- **KORISNIK**: idKorisnik, korisnickoIme, zaporka, email
- **VINYL PLOČA**: idVinylPloca, nazivVinylPloca, godinaIzdanja, infoVinylPloca, verzija, ukOcjena
- **RECENZIJA**: idRecenzija, tekst, datumPostavljanja, ocjena
- **ŽANR**: idZanr, nazivZanr

- IZDAVČKA KUĆA: idIzdavackaKuca, nazivIzdavackaKuca , infoIzdavackaKuca
- MJESTO: idMjesto, pbr, nazivMjesto
- PRODUCENT: idProducent, nazivProducent, pocDjelovanjaProd, krajDjelovanjaProd
- VLASNIK: idVlasnik, nazivVlasnik
- VRSTA PLOCE: idVrsta, nazivVrstaPloce
- IZVOĐAČ: idIzvodac, nazivIzvodac, infoIzvodac, jeGrupa
- GLAZBENIK: idGlazbenik, datumRod, datumSmrti
- GRUPA: idGrupa, pocDjelovanjaGrupa, krajDjelovanjaGrupa
- PJESMA: idPjesma, nazivPjesma, trajanjePjesma
- DRŽAVA: idDrzava, nazivDrzava
- SLIKA: idSlika, slika

2.1.3. Veze

- ostavlja: idRecenzija, idKorisnik
- ima: idRecenzija, idVinylPloca
- ima_u_kolekciji: idKorisnik, idVinylPloca, datDodavanja
- želi: idKorisnik, idVinylPloca
- je_vrste: idVinylPloca, idVrsta
- je_žanra: idVinylPloca, idZanr
- izdaje: idVinylPloca, idIzdavackaKuca
- sadrži: idVinylPloca, idPjesma, redniBr
- snima: idPjesma, idIzvodac

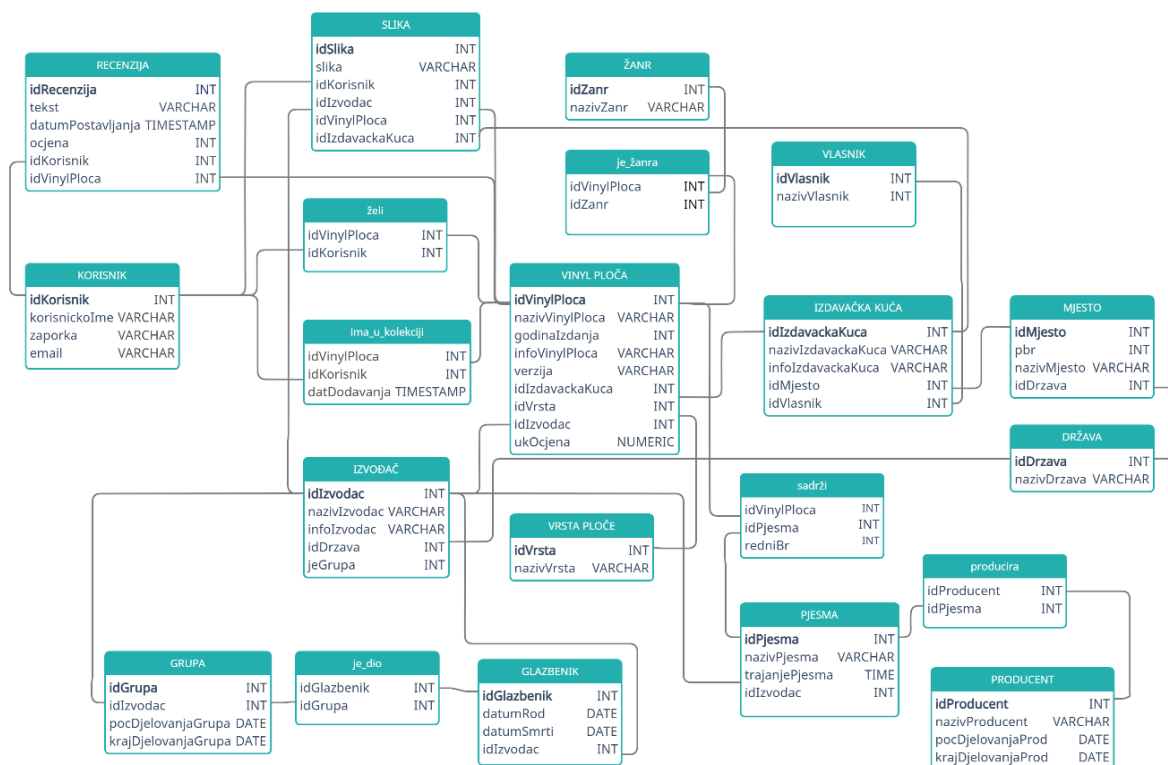
- objavljuje: idVinylPloca, idIzvodac
- dolazi_iz: idIzvodac, idDrzava
- je_dio: idGrupa, idGlazbenik
- izvođač_slika: idSlika, idIzvodac
- korisnik_slika: IdSlika, idKorisnik
- ploča_slika: idSlika, idVinylPloca
- izdKuća_slika: idSlika, idIzdavackaKuca
- je_u_državi: idMjesto, idDrzava
- se_nalazi_u: idIzdavackaKuca, idMjesto
- producira: IdPjesma, idProducent
- vodi: idIzdavackaKuca, idVlasnik

2.2. Relacijski model

Iz prethodno opisanog ER modela generira se relacijski model, tako dobivamo sve tablice (relacije) koje će biti potrebne za razvoj aplikacije, Slika 2.2 prikazuje generirani model.

Kod veza 1:N i N:1 višestruki entitet proširuje se sa stranim ključem koji je primarni ključ drugog povezanog entiteta, veza N:N pretvara se u zasebnu relaciju čiji je primarni ključ kombinacija primarnih ključeva dvaju entiteta koje povezuje, dok se veza 1:1 može se transformirati na više načina ovisno o situaciji.

Ovaj model implementirati ću u sustavu PostgreSQL [3], specifično u alatu pgAdmin.



Slika 2.2 Relacijski model baze podataka

2.2.1. Popis relacija

- Drzava

Relaciju Drzava čine idDrzava i nazivDrzava. Svi atributi su obvezni. Tablica 2.1 prikazuje cijelu relaciju.

Tablica 2.1 Relacija Drzava

Atribut	Tip podataka	Ograničenje	Null
idDrzava	integer	primarni ključ	ne
nazivDrzava	varchar(30)		ne

- Glazbenik

Relacija Glazbenik je generalizacija relacije Izvodac, osim jedinstvenog id-a sadrži i strani ključ relacije Izvodac. Postavljeno je ograničenje da datum smrti ne smije biti prije datuma rođenja. Tablica 2.2 prikazuje cijelu relaciju.

Tablica 2.2 Relacija Glazbenik

Atribut	Tip podataka	Ograničenje	Null
idGlazbenik	integer	primarni ključ	ne
idIzvodac	varchar(30)	strani ključ	ne
datumRod	date	datumRod<=datumSmrti	da
datumSmrti	date	datumRod<=datumSmrti	da

- Grupa

Relacija Grupa je generalizacija relacije Izvodac, osim jedinstvenog id-a sadrži i strani ključ relacije Izvodac. Postavljeno je ograničenje da kraj djelovanja grupe ne smije biti prije početka djelovanja. Tablica 2.3 prikazuje cijelu relaciju.

Tablica 2.3 Relacija Grupa

Atribut	Tip podataka	Ograničenje	Null
idGrupa	integer	primarni ključ	ne
idIzvodac	varchar(30)	strani ključ	ne
pocDjelovanjaGrupa	integer	pocDjelovanjaGrupa <=krajDjelovanjaGrupa	da
krajDjelovanjaGrupa	integer	pocDjelovanjaGrupa <=krajDjelovanjaGrupa	da

- imaUKolekciji

Relacija imaUKolekciji nastala je iz N:N veze između Relacija Korisnik i VinylPloca. Osim primarnog ključa sadrži i datum dodavanja albuma u kolekciju. Tablica 2.4 prikazuje cijelu relaciju.

Tablica 2.4 Relacija imaUKolekciji

Atribut	Tip podataka	Ograničenje	Null
idKorisnik	integer	primarni ključ	ne
idVinylPloca	integer	primarni ključ	ne
datDodavanja	timestamp		ne

- IzdavackaKuca

Relacija IzdavackaKuca sadrži sve potrebne informacije o izdavačkoj kući. Referencira relacije Mjesto i Vlasnik. Tablica 2.5 prikazuje cijelu relaciju.

Tablica 2.5 Relacija IzdavackaKuca

Atribut	Tip podataka	Ograničenje	Null
idIzdavackaKuca	integer	primarni ključ	ne
idMjesto	integer	Strani ključ	ne
idVlasnik	Integer	Strani ključ	ne
nazivIzdavackaKuca	varchar(30)		ne
infoIzdavackaKuca	varchar(1000)		da

- Izvodac

Relacija Izvodac sadrži osnovne informacije o izvođaču, nasljeđuju ju relacije Glazbenik i Grupa. Referencira relaciju Država. Atribut jeGrupa označava je li izvođač grupa ili glazbenik. Tablica 2.6 prikazuje cijelu relaciju.

Tablica 2.6 Relacija Izvodac

Atribut	Tip podataka	Ograničenje	Null
idIzvodac	integer	primarni ključ	ne
idDrzava	integer	Strani ključ	da
nazivIzvodac	varchar(30)		ne
infoIzvodac	varchar(1000)		da
jeGrupa	integer		ne

- jeDio

Relacija jeDio nastala je iz veze Glazbenika i Grupe. Glazbenik može biti član više grupa, a grupa se može sastojati od više glazbenika. Tablica 2.7 prikazuje cijelu relaciju.

Tablica 2.7 Relacija jeDio

Atribut	Tip podataka	Ograničenje	Null
idGlazbenik	integer	primarni ključ	ne
idGrupa	integer	primarni ključ	ne

- jeZanra

Relacija jeZanra nastala je iz N:N veze relacija VinylPloca i Zanr. Tablica 2.8 prikazuje cijelu relaciju.

Tablica 2.8 Relacija jeZanra

Atribut	Tip podataka	Ograničenje	Null
idZanr	integer	primarni ključ	ne
idVinylPloca	integer	primarni ključ	ne

- Korisnik

Relacija Korisnik sadrži osnovne informacije o registriranom korisniku: njegovo korisničko ime i email koji trebaju biti jedinstveni te zaporku koja se ne sprema u originalnom obliku nego kao kriptografski sažetak (engl. *hash*). Tablica 2.9 prikazuje cijelu relaciju. Korisničko ime mora biti dulje od 4 znaka i kraće od 21 znak.

Tablica 2.9 Relacija Korisnik

Atribut	Tip podataka	Ograničenje	Null
idKorisnik	integer	primarni ključ	ne
email	varchar(255)	jedinstvena vrijednost	ne
korisnickoIme	varchar(20)	jedinstvena vrijednost 5 ≤ length(korisnickoIme) ≤ 20	ne
Zaporka	varchar(999)		ne

- Mjesto

Relacija Mjesto sadrži naziv i poštanski broj mjesta, te referencira relaciju Drzava. Poštanski broj mjesta je jedinstvena vrijednost. Tablica 2.10 prikazuje cijelu relaciju.

Tablica 2.10 Relacija Mjesto

Atribut	Tip podataka	Ograničenje	Null
idMjesto	integer	primarni ključ	ne
pbr	integer	jedinstvena vrijednost	ne
nazivMjesto	varchar(20)		ne
idDrzava	integer	strani ključ	ne

- Pjesma

Relacija Pjesma sadrži njen naziv i trajanje te preko 1:N veze s izvođačem referencira istoimenu tablicu. Tablica 2.11 prikazuje cijelu relaciju.

Tablica 2.11 Relacija Pjesma

Atribut	Tip podataka	Ograničenje	Null
idPjesma	integer	primarni ključ	ne
nazivPjesma	varchar(50)		ne
trajanjePjesma	time		da
idIzvodac	integer	strani ključ	ne

- Producent

Relacija producent sadrži osnovne informacije o producentu pjesme. Postavljeno je ograničenje da kraj djelovanja producenta ne smije biti prije početka njegovog djelovanja. Tablica 2.12 prikazuje cijelu relaciju.

Tablica 2.12 Relacija Producent

Atribut	Tip podataka	Ograničenje	Null
idProducent	integer	primarni ključ	ne
nazivProducent	varchar(20)		ne
pocDjelovanjaProd	date	pocDjelovanjaProd <= krajDjelovanjaProd	da
krajDjelovanjaProd	date	pocDjelovanjaProd <= krajDjelovanjaProd	da

- producira

Relacija producira nastaje iz N:N veze između relacija Pjesma i Producent. Tablica 2.13 prikazuje cijelu relaciju.

Tablica 2.13 Relacija producira

Atribut	Tip podataka	Ograničenje	Null
idPjesma	integer	primarni ključ	ne
idProducent	integer	primarni ključ	ne

- Recenzija

Relacija Recenzija sadrži ocjenu i tekst recenzije te datum i vrijeme postavljanja recenzije. Referencira relacije Korisnik i VinylPloca. Postavljeno je ograničenje na atribut ocjena da mora biti između 1 i 5. Tekst nije obavezan. Tablica 2.14 prikazuje cijelu relaciju.

Tablica 2.14 Relacija Recenzija

Atribut	Tip podataka	Ograničenje	Null
---------	--------------	-------------	------

idRecenzija	integer	primarni ključ	ne
tekst	varchar(255)		da
datumPostavljanja	timestamp		ne
ocjena	integer	$0 < \text{ocjena} < 6$	ne
idKorisnik	integer	strani ključ	ne
idVinylPloca	integer	strani ključ	ne

- sadrzi

Relacija sadrzi nastaje iz N:N veze relacija VinylPloca i Pjesma. Osim primarnih ključeva spomenutih relacija sadrži i redni broj pjesme na albumu. Tablica 2.15 prikazuje cijelu relaciju.

Tablica 2.15 Relacija sadrzi

Atribut	Tip podataka	Ograničenje	Null
idPjesma	integer	primarni ključ	ne
idVinylPloca	integer	primarni ključ	ne
redniBr	Integer		da

- Slika

Relacija Slika sadrži naslovnu sliku korisnika, vinyl ploče, izdavačke kuće ili izvođača. Slika se u pretvori u base64 i tako pohrani. Tablica 2.16 prikazuje cijelu relaciju.

Tablica 2.16 Relacija Slika

Atribut	Tip podataka	Ograničenje	Null
idSlika	integer	primarni ključ	ne

slika	varchar(1000)		ne
idKorisnik	integer	strani ključ	da
idIzvodac	integer	strani ključ	da
idVinylPloca	integer	strani ključ	da
idIzdavackaKuca	integer	strani ključ	da

- VinylPloca

Relacija VinylPloca sadrži naziv, godinu izdanja, verziju te dodatne informacije o albumu. Također sadrži atribut ukOcjena koji se računa kao prosjek svih ocjena iz relacije Recenzija za danu ploču. Referencira relacije Izvođač, VrstaPloce i IzdavačkaKuća. Tablica 2.17 prikazuje cijelu relaciju.

Tablica 2.17 Relacija VinylPloca

Atribut	Tip podataka	Ograničenje	Null
idVinylPloca	integer	primarni ključ	ne
nazivVinylPloca	varchar(30)		ne
godinaIzdanja	integer		da
infoVinylPloca	varchar(1000)		da
Verzija	varchar(15)		da
idIzdavackaKuca	integer	strani ključ	da
idIzvodac	integer	strani ključ	ne
idVrsta	integer	strani ključ	ne

- Vlasnik

Relacija Vlasnik sadrži informacije o vlasniku izdavačke kuće. Tablica 2.18 prikazuje cijelu relaciju.

Tablica 2.18 Relacija Vlasnik

Atribut	Tip podataka	Ograničenje	Null
idVlasnik	integer	primarni ključ	ne
nazivVlasnik	varchar(50)		ne

- VrstaPloce

Relacija VrstaPloce sadrži informacije o vrsti vinyl ploče koja može biti: Long Play (LP), Extended Play i Single. Tablica 2.19 prikazuje cijelu relaciju.

Tablica 2.19 Relacija VrstaPloce

Atribut	Tip podataka	Ograničenje	Null
idVrsta	integer	primarni ključ	ne
nazivVrsta	varchar(15)		ne

- Zavr

Relacija Zavr sadrži osnovne podatke o žanru vinyl ploče. Tablica 2.20 prikazuje cijelu relaciju.

Tablica 2.20 Relacija Zavr

Atribut	Tip podataka	Ograničenje	Null
idZavr	integer	primarni ključ	ne
nazivZavr	varchar(20)		ne

- zeli

Relacija zeli nastala je iz N:N veze između Relacija Korisnik i VinylPloca. Tablica 2.21 prikazuje cijelu relaciju.

Tablica 2.21 Relacija zeli

Atribut	Tip podataka	Ograničenje	Null
idKorisnik	integer	primarni ključ	ne
idVinylPloca	integer	primarni ključ	ne

* Za svaku od relacija zapisuje se id koji se automatski generira. On služi za jednostavniju pohranu i jedinstvenu identifikaciju podataka

2.2.2. Primjeri create naredbi

Tablica 2.22 sadrži naredbe kreiranja za neke od prethodno opisanih relacija.

Tablica 2.22 Create naredbe

Korisnik	<pre>CREATE TABLE "Korisnik" ("idKorisnik" SERIAL NOT NULL , "email" VARCHAR(255) NOT NULL, "korisnickoIme" VARCHAR(20) NOT NULL, "zaporka" VARCHAR(999) NOT NULL, UNIQUE("email"), UNIQUE("korisnickoIme"), PRIMARY KEY("idKorisnik"), CONSTRAINT korisnickoime_len_check CHECK (length("korisnickoIme"::text) <= 20 AND length("korisnickoIme"::text) >= 5));</pre>
Izvodac	<pre>CREATE TABLE "Izvodac" ("idIzvodac" SERIAL NOT NULL, "nazivIzvodac" VARCHAR(30) NOT NULL,</pre>

	<pre>"infoIzvodac" VARCHAR(1000),"idDrzava" INTEGER , PRIMARY KEY("idIzvodac"), FOREIGN KEY ("idDrzava") REFERENCES "Drzava" ("idDrzava") ON DELETE SET NULL);</pre>
Grupa	<pre>CREATE TABLE "Grupa" ("idGrupa" SERIAL NOT NULL, "idIzvodac" INTEGER NOT NULL, "pocDjelovanjaGrupa" INTEGER, "krajDjelovanjaGrupa" INTEGER, PRIMARY KEY("idGrupa"), FOREIGN KEY("idIzvodac") REFERENCES "Izvodac"("idIzvodac") ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT djelovanje_grupa_check CHECK ("pocDjelovanjaGrupa" <= "krajDjelovanjaGrupa"));</pre>
Glazbenik	<pre>CREATE TABLE "Glazbenik" ("idGlazbenik" SERIAL NOT NULL, "idIzvodac" INTEGER NOT NULL, "datumRod" DATE, "datumSmrti" DATE, PRIMARY KEY("idGlazbenik"), FOREIGN KEY("idIzvodac") REFERENCES "Izvodac"("idIzvodac") ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT rodenje_smrt_check CHECK ("datumRod" <= "datumSmrti"));</pre>
VinylPloca	<pre>CREATE TABLE "VinylPloca" ("idVinylPloca" SERIAL NOT NULL, "nazivVinylPloca" VARCHAR(30) NOT NULL, "godinaIzdanja" INTEGER, "infoVinylPloca" VARCHAR(1000), "verzija" VARCHAR(15), "idIzdavackaKuca" INTEGER, "idVrsta" INTEGER NOT NULL, "idIzvodac" INTEGER NOT NULL, "ukOcjena" NUMERIC(3,1),</pre>

	PRIMARY KEY("idVinylPloca"), FOREIGN KEY("idIzdavackaKuca") REFERENCES "IzdavackaKuca"("idIzdavackaKuca") ON DELETE SET NULL, FOREIGN KEY("idVrsta") REFERENCES "VrstaPloce"("idVrsta") ON DELETE CASCADE, FOREIGN KEY ("idIzvodac") REFERENCES "Izvodac"("idIzvodac") ON DELETE CASCADE);
imaUKolekciji	CREATE TABLE "imaUKolekciji" ("idVinylPloca" INTEGER NOT NULL, "idKorisnik" INTEGER NOT NULL, "datDodavanja" TIMESTAMP NOT NULL, PRIMARY KEY("idVinylPloca", "idKorisnik"), FOREIGN KEY("idVinylPloca") REFERENCES "VinylPloca"("idVinylPloca") ON DELETE CASCADE, FOREIGN KEY("idKorisnik") REFERENCES "Korisnik"("idKorisnik") ON DELETE CASCADE);
Recenzija	CREATE TABLE "Recenzija" ("idRecenzija" SERIAL NOT NULL, "tekst" VARCHAR(255) , "datumPostavljanja" TIMESTAMP NOT NULL, "ocjena" INTEGER NOT NULL, "idKorisnik" INTEGER NOT NULL, "idVinylPloca" INTEGER NOT NULL, PRIMARY KEY("idRecenzija"), FOREIGN KEY("idVinylPloca") REFERENCES "VinylPloca"("idVinylPloca") ON DELETE CASCADE, FOREIGN KEY("idKorisnik") REFERENCES "Korisnik"("idKorisnik") ON DELETE CASCADE, CONSTRAINT ocjena_check CHECK ("ocjena" > 0 AND "ocjena" < 6));

2.2.3. Primjeri insert naredbi

Isječak koda 2.1 prikazuje nekolicinu izdvojenih naredbi za unošenje podataka u prethodno opisane tablice.

```
insert into "VinylPloca"("nazivVinylPloca", "godinaIzdanja",
"infoVinylPloca", "verzija", "idIzdavackaKuca", "idVrsta", "idIzvodac",
"ukOcjena") VALUES ('Bad Girls',1979,'Bad Girls je sedmi studijski album
američke pjevačice i kantautorice Donne Summer, objavljen 25. travnja
1979. u izdanju Casablanca Records. Izvorno izdana kao dvostruki album,
Bad Girls postao je najprodavaniji i najcjenjeniji album Donnine
karijere. To je ujedno bio i njezin posljednji studijski album za
Casablanca Records.','NBLP-2-7150',3,1,6,null);

insert into "Pjesma"("nazivPjesma", "trajanjePjesma", "idIzvodac")
VALUES ('Hot Stuff','00:05:13',6);

insert into "Pjesma"("nazivPjesma", "trajanjePjesma", "idIzvodac")
VALUES ('Bad Girls','00:04:57',6);

insert into "Pjesma"("nazivPjesma", "trajanjePjesma", "idIzvodac")
VALUES ('Love Will Always Find You','00:04:01',6);

insert into "sadrzi"("idPjesma", "idVinylPloca", "redniBr") VALUES
(1,1,1);

insert into "sadrzi"("idPjesma", "idVinylPloca", "redniBr") VALUES
(2,1,2);

insert into "jeDio"("idGlazbenik", "idGrupa") VALUES (4,4);

insert into "Zanr" ( "nazivZanr") VALUES ('Rock');
insert into "Zanr" ( "nazivZanr") VALUES ('Electronic');
insert into "Zanr" ( "nazivZanr") VALUES ('Zabavna glazba');
```

Isječak koda 2.1 Create naredbe

2.2.4. Okidači

Sustav za upravljanje bazom podataka koji za sada imamo je pasivan – operacije nad tablicama izvode se samo kada korisnik to eksplicitno zada [1]. U nekim slučajevima voljeli bi da se određene operacije automatski izvedu nakon nekog od naših zahtjeva. Za to su nam potrebni okidači.

Okidači rade na principu:

Kada se nešto dogodi

Ako je uvjet zadovoljen onda izvrši funkciju

U slučaju ovog sustava bilo bi zgodno da se nakon svakog dodavanja u tablicu Recenzija, izračunava prosječna ocjena za zadanu vinyl ploču na temelju atributa ocjena. To ostvarujemo okidačem 'insRecenzija' (Isječak koda 2.2) koji poziva funkciju 'insert_recenzija()' (Isječak koda 2.3).

```
create trigger "insRecenzija"  
  after insert  
  on "Recenzija"  
  for each row  
execute procedure insert_recenzija();
```

Isječak koda 2.2 Trigger insRecenzija

```
create function insert_recenzija() returns trigger  
  language plpgsql  
as  
$$  
BEGIN  
  UPDATE vinylploca SET "ukOcjena" = (SELECT AVG("ocjena") FROM  
"Recenzija" WHERE "Recenzija"."idVinylPloca" =  
"VinylPloca"."idVinylPloca")  
  WHERE "idVinylPloca" = NEW. "idVinylPloca";  
  RETURN NEW;  
END;  
$;
```

Isječak koda 2.3 Funkcija insert_recenzija()

Također je potrebno dodati okidač (Isječak koda 2.4) koji nakon što korisnik doda ploču u kolekciju poziva funkciju (Isječak koda 2.5) koja provjerava nalazi li se ista ploča na korisnikovoj listi želja. Ako je uvjet istinit ploča se uklanja s liste želja.

```
create trigger "insImaUKolekciji"  
  after insert  
  on "imaUKolekciji"  
  for each row  
execute procedure plocadodanaukolekciju();
```

Isječak koda 2.4 Trigger insImaUKolekciji

```
create function plocadodanaukolekciju() returns trigger  
  language plpgsql  
as  
$$  
BEGIN  
  DELETE FROM "zeli" WHERE ("zeli"."idVinylPloca" =  
NEW."idVinylPloca" AND "zeli"."idKorisnik" = NEW."idKorisnik");  
  RETURN NULL;  
END;  
$;
```

Isječak koda 2.5 Funkcija plocadodanaukolekciju()

2.2.5. Uloge

Jedna od najbitnijih značajki svakog sustava je sigurnost. Ne bi bilo poželjno da bilo tko tko dobije pristup bazi može pregledavati i modificirati podatke. Zbog toga je potrebno postaviti ovlasti za korisnike. No, u slučajevima kada velik broj korisnika može pristupiti bazi, nezgodno je svakom pojedinačnom korisniku postavljati ovlasti. Tada je korisno definirati uloge mogućih korisnika sustava.

U ovom sustavu odlučila sam definirati tri vrste uloga. Prva uloga 'admin' omogućit će korisniku upravljanje (stvaranje, brisanje, modificiranje) svim tablicama u sustavu. Sljedeća uloga je 'korisnik', ona se odnosi na klasičnog korisnika ovog sustava koji će imati mogućnost pregledati sve tablice, ali samo neke će moći modificirati. Zadnja uloga 'readonly' biti će namijenjena svim ostalim korisnicima, korisnici kojima je pridijeljena ova uloga imati će mogućnost pregledati samo neke tablice.

Kod za stvaranje navedenih uloga prikazan je u nastavku.

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE "Plocnik" FROM PUBLIC;
CREATE ROLE readonly, korisnik, admin;

GRANT CONNECT ON DATABASE "Plocnik" TO readonly;
GRANT USAGE ON SCHEMA public TO readonly;
GRANT SELECT ON TABLE
"Drzava","Glazbenik","Grupa","IzdavackaKuca","Izvodac","jeDio","jeZan
ra","Mjesto","Pjesma","Producent","Producira","Recenzija","Sadrzi","S
lika","VinylPloca","Vlasnik","Vrsta","Zanr" TO readonly;

GRANT CONNECT ON DATABASE "Plocnik" TO admin;
GRANT USAGE, CREATE ON SCHEMA public TO admin;
GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA public
TO admin;
CREATE USER admin1 WITH PASSWORD 'adminadmin';
GRANT admin TO admin1;

GRANT CONNECT ON DATABASE "Plocnik" TO korisnik;
GRANT USAGE ON SCHEMA public TO korisnik;
GRANT SELECT ON ALL TABLES IN SCHEMA public to korisnik;
GRANT INSERT,UPDATE,DELETE ON TABLE
"Zeli","imaUKolekciji","Recenzija" to korisnik;
```

3. Implementacija web-aplikacije

Nakon uspješno kreirane baze podataka, potrebno je implementirati samu web-aplikaciju. Prvo uz pomoć radnog okvira Spring [7] uspostavljam vezu s bazom te implementiram sve funkcionalnosti sustava, zatim putem JavaScript biblioteke React [5] implementiram korisničko sučelje.

3.1. Arhitektura i dizajn sustava

Arhitektura sustava može se podijeliti na tri podsustava:

- Web poslužitelj
- Web-aplikacija
- Baza podataka

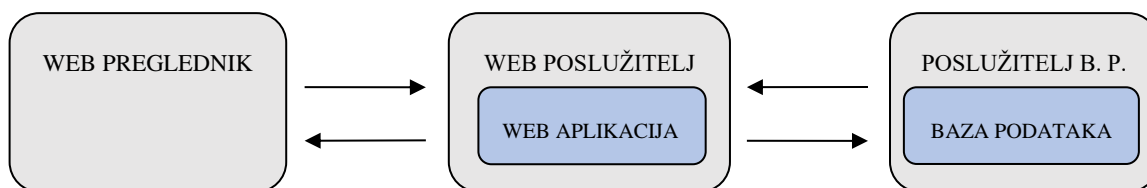
Slika 3.1 ilustrira komunikaciju između podsustava.

Web preglednik omogućuje klijentu prikaz web-stranice. Klijent preko korisničkog sučelja komunicira s poslužiteljem preko HTTP zahtjeva.

Web poslužitelj jezgra je sustava, on je posrednik između korisničkog sučelja i aplikacije. Poslužitelj prima HTTP zahtjev klijenta, pokreće aplikaciju i prosljeđuje joj zahtjev, nakon što aplikacija obradi zahtjev klijentu se natrag šalje HTTP odgovor.

Web-aplikacija zahtjev prima u JSON (*JavaScript Object Notation*) formatu, i uz pomoć zadanog URL-a zna koju metodu treba izvršiti. Pri izvršavanju metode aplikacija može komunicirati s bazom. Kada se metoda izvrši aplikacija prosljeđuje odgovor web poslužitelju.

Baza podataka već opisana u prethodnom poglavlju služi za pohranu podataka.



Slika 3.1 Arhitektura sustava

3.2. Višeslojna arhitektura radnog okvira Spring

Backend ovog sustava implementiran je u Spring radnom okviru, specifično Spring Bootu [4]. Ovaj radni okvir temelji se na višeslojnoj arhitekturi [2], najbitniji slojevi navedeni su u nastavku:

- Nadglednik/Controller
- Usluga/Service
- Repozitorij/Repository
- Domena/Domain

Nadglednici služe za komunikaciju s klijentskom stranom. Oni primaju zahtjeve od korisnika te na temelju URL-a pokreću određenu metodu. Također se u njima može provjeravati autentičnost korisnika uz pomoć Spring Security-a koji prilikom prijave u sustav korisniku dodjeljuje ulogu.

U implementaciji ove aplikacije koriste se uloge 'ROLE_ADMIN' i 'ROLE_KORISNIK'. Isječak koda 3.1 prikazuje metodu u nadgledniku kojoj se pristupa putem DELETE HTTP zahtjeva koju može izvršiti samo administrator prikazan je u nastavku, dok Slika 3.2 prikazuje dijagram razreda nadglednika čitavog sustava.

```

@DeleteMapping("/{id}/obrisiIzdavackuKucu")
@Secured({"ROLE_ADMIN"})
@ResponseBody
public ResponseEntity<Object> deleteIzdavackaKuca( @PathVariable("id") String
izdavackaKucaId, @AuthenticationPrincipal User user) {

    try {

        IzdavackaKuca izdavackaKuca =
izdavackaKucaService.fetch(Long.valueOf(izdavackaKucaId));

        if(izdavackaKuca!=null) {

            IzdavackaKuca deletedIzdavackaKuca =
izdavackaKucaService.deleteIzdavackaKuca(izdavackaKuca.getIdIzdavackaKuca());

            return
ResponseEntity.status(HttpStatus.OK).body(deletedIzdavackaKuca);

        } catch (RequestDeniedException ex) {

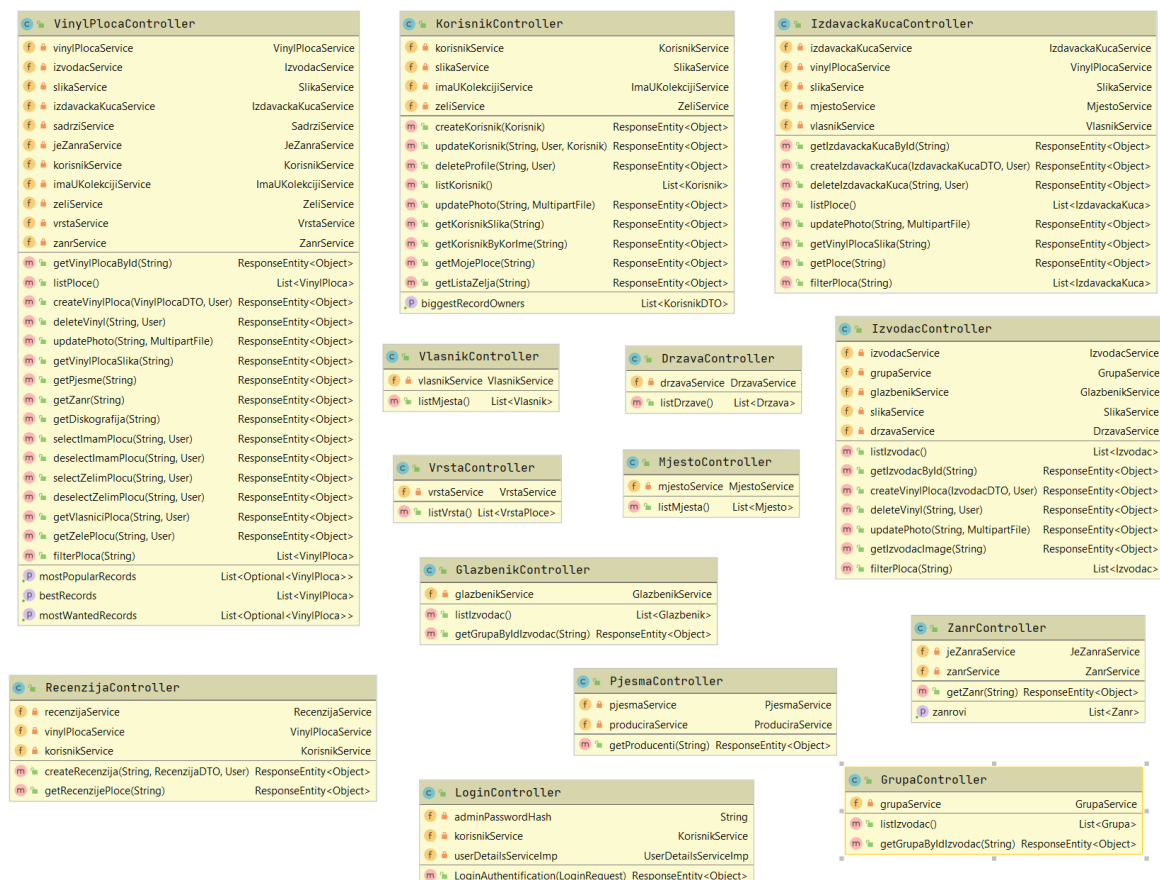
            return ResponseEntity.badRequest().body(ex);

        }

        return ResponseEntity.badRequest().body(new RequestDeniedException("Samo
admin može brisati!"));
    }
}

```

Isječak koda 3.1 Metoda deleteIzdavackaKuca()



Slika 3.2 Dijagram razreda - nadglednici

Nakon što nadglednik primi zahtjev poziva odgovarajuće razrede **usluga** ili servisa koji taj zahtjev obrađuju uglavnom pozivajući metode razreda repozitorija. Sloj usluga zaslužan je za temeljnu funkcionalnost web-aplikacije.

Isječak koda 3.2 prikazuje metodu razreda servis koja putem repozitorija dohvaća najbolje ocijenjene vinyl ploče.

Sloj **repozitorij** pruža mogućnost jednostavne komunikacije s bazom podataka gdje u većini slučajeva nije potrebno pisati SQL naredbe, nego su te funkcionalnosti već ugrađene, što je ostvareno uz pomoć JDBC (*Java Database Connectivity*) i JPA (*Jakarta Persistence*) usluga Spring Data u sklopu radnog okvira Spring .

Domena je je preslika modela baze podataka u programskom jeziku Java.

```
@Override
public List<VinylPloca> getBestRecords() {
    List<VinylPloca> ploce = vinylPlocaRepo.findAllOrderByUkOcjenaDesc();
    List<VinylPloca> remove = new ArrayList<>();
    for(VinylPloca ploca : ploce){
        if(ploca.getUkOcjena()==null){
            remove.add(ploca);
        }
    }
    ploce.removeAll(remove);
    List<VinylPloca> firstNElementsList =
    ploce.stream().limit(5).collect(Collectors.toList());
    return firstNElementsList;
}
```

Isječak koda 3.2 Metoda getBestRecords()

3.3. JavaScript biblioteka React

Putem biblioteke React i React DOM implementirano je korisničko sučelje. React omogućuje jednostavan prijenos podataka i uspostavu veze s backendom. Isječak koda 3.4 i Isječak koda 3.5 prikazuju dohvat liste izvođača s backenda te očuvanje i pohranu stanja komponente.

Čitav kod korisničkog sučelja ostvaren je putem HTML-a, CSS-a i JavaScripta-a. Izgled aplikacije implementiran je uz pomoć CSS radnog okvira Materialize[6].

```
izvodacService.getIzvodaci().then(async response => {
    const data = await response.json();
    if (!response.ok) {
        const error = (data && data.message) ||
response.statusText;
        return Promise.reject(error);
    }
    this.setState({
        izvodaci:data
    })
})
.catch(error => {
    this.setState({ errorMessage: error.toString() });
    console.error('There was an error!', error);
});
```

Isječak koda 3.4 Poziv metode getIzvodaci()

```
getIzvodaci(){
    const requestOptions ={
        method: 'GET',
        headers:{'Content-Type': 'application/json'}
    };
    return fetch('/izvodac/list', requestOptions);
}
```

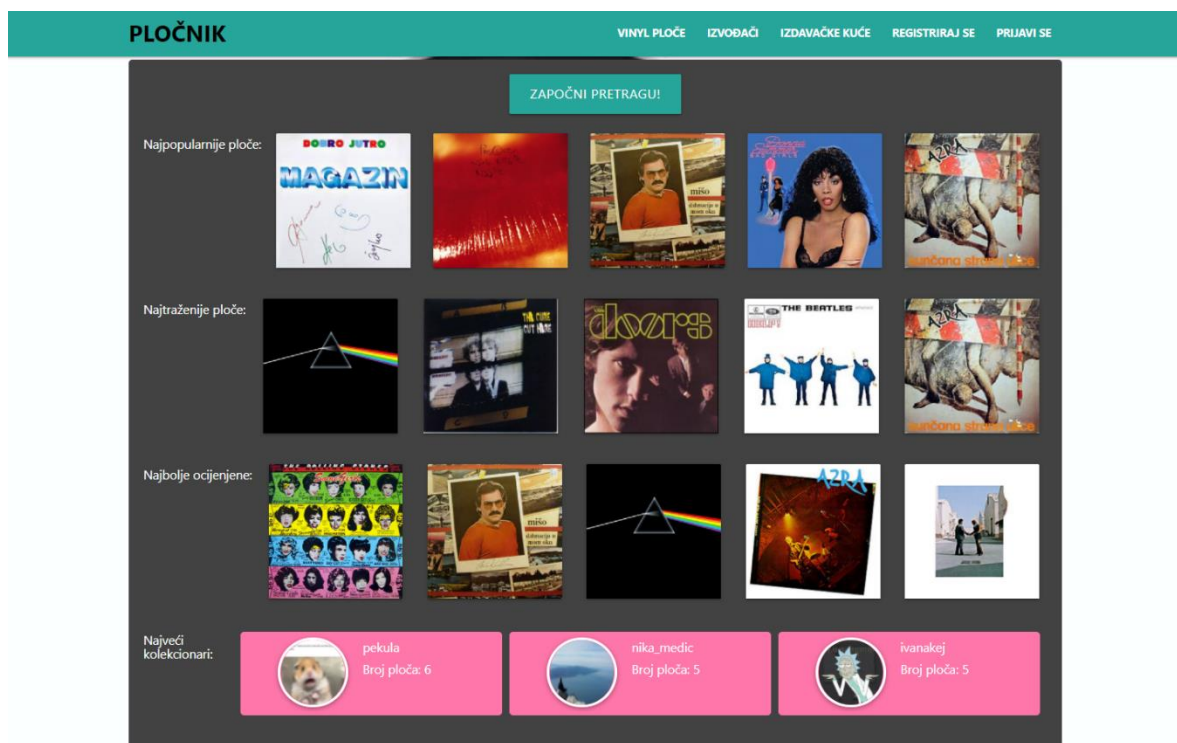
Isječak koda 3.5 Slanje HTTP GET zahtjeva backendu

4. Korisničke upute

4.1. Neregistrirani/neprijavljeni korisnik

4.1.1. Početna stranica

Nakon uspješno pokrenute aplikacije u web pregledniku otvori se početna stranica (Slika 4.1) na kojoj se nalazi navigacijska traka te nekoliko osnovnih informacija kao što su: najpopularnije ploče, najtraženije ploče, najbolje ocijenjene ploče i najveći kolekcionari.

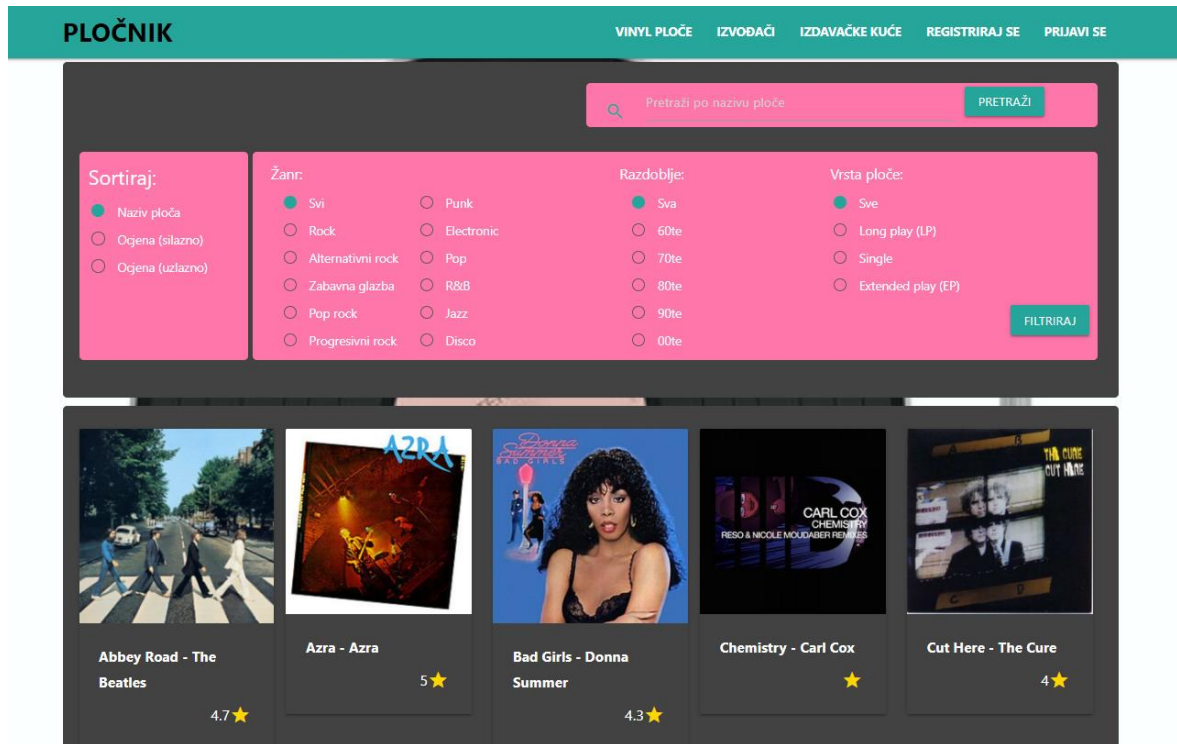


Slika 4.1 Početna stranica

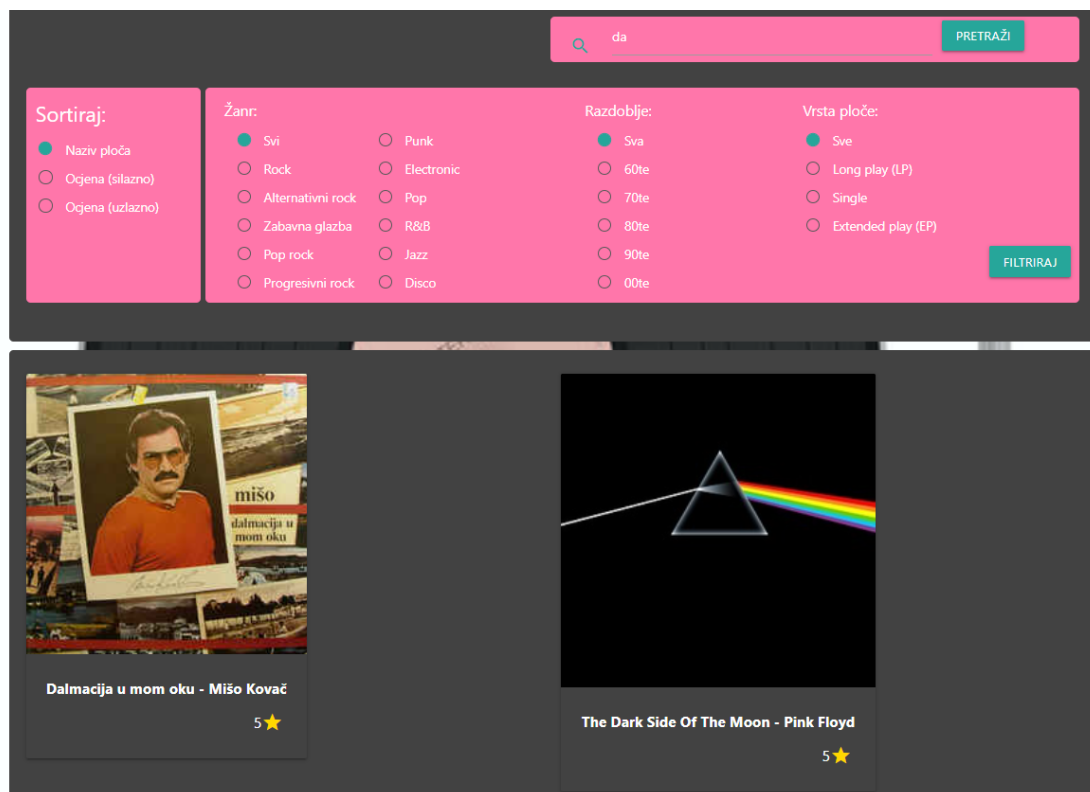
4.1.2. Pregled vinyl ploča

Klikom na 'VINYL PLOČE' u navigacijskoj traci ili na gumb 'ZAPOČNI PRETRAGU' korisnik dolazi do stranice na kojoj se nalazi popis svih ploča (Slika 4.2). Ploče se mogu pretražiti prema nazivu (Slika 4.3) , sortirati prema nazivu i ocjeni (ulazno ili silazno) i filtrirati. Filtriranje je moguće po žanru, razdoblju i vrsti ploče, također je ostvareno i

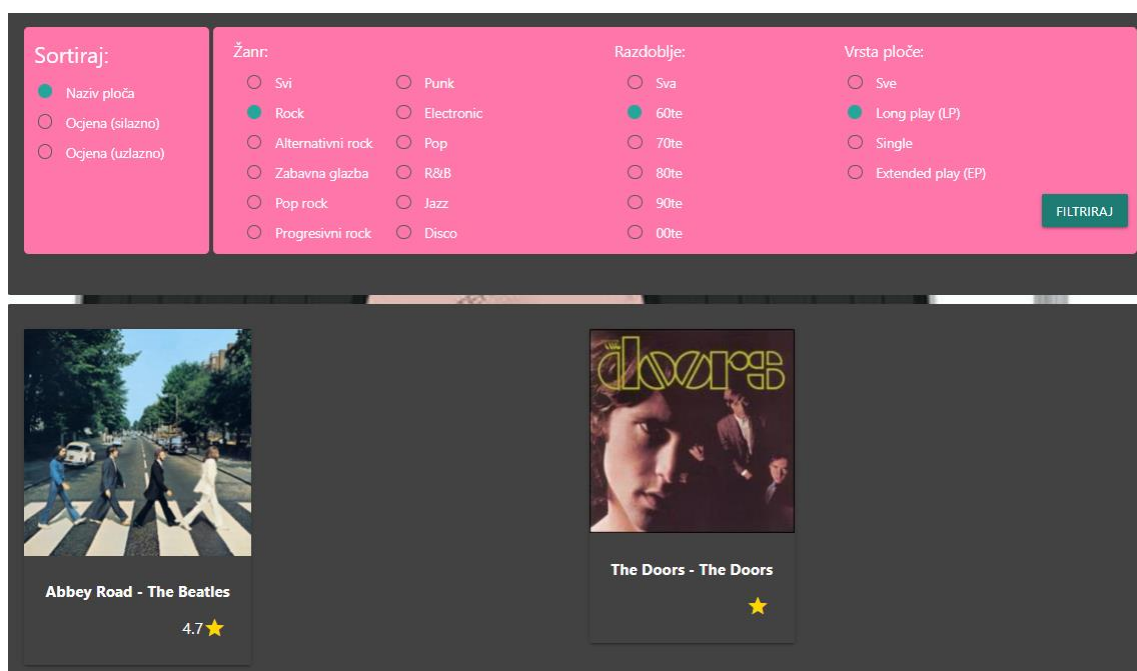
kombiniranje filtera. Slika 4.4 prikazuje rezultat filtriranja gdje se pretražuju rock LP albumi iz 60tih godina prošlog stoljeća.



Slika 4.2 Pregled vinyl ploča



Slika 4.3 Pretraga prema nazivu 'da'



Slika 4.4 Filtriranje

Klikom na jednu od ploča dolazi se do stranice sa svim informacijama o albumu (Slika 4.5), kao što su: žanr, godina izdanja, izdavačka kuća, kratki opis, popis pjesma,...

The Cure - Disintegration 4.2 ★

Žanr: Rock Electronic
 Godina izdanja: 1989
 Izdavačka kuća: [Fiction Records](#)
 Vrsta ploče: Long play
 Verzija: FIXH 14

Osmi studijski album The Cure. Dobio je dvostruko platinasti status u SAD-u, što ga čini najprodavanijim albumom benda do sada. Zbog produženog vremena trajanja albuma, "Last Dance" i "Homesick" nisu bili uključeni u originalno izdanje albuma s vinilima . Ponovno izdano 2010. godine s bonus CD-om rjetkih pjesama i "Entreat plus"

Sadržaj:

	Naziv pjesme	Trajanje	Producent
1.	Plainsong	05:16	David M. Allen
2.	Pictures Of You	07:28	David M. Allen

Slika 4.5 Informacije o vinylu

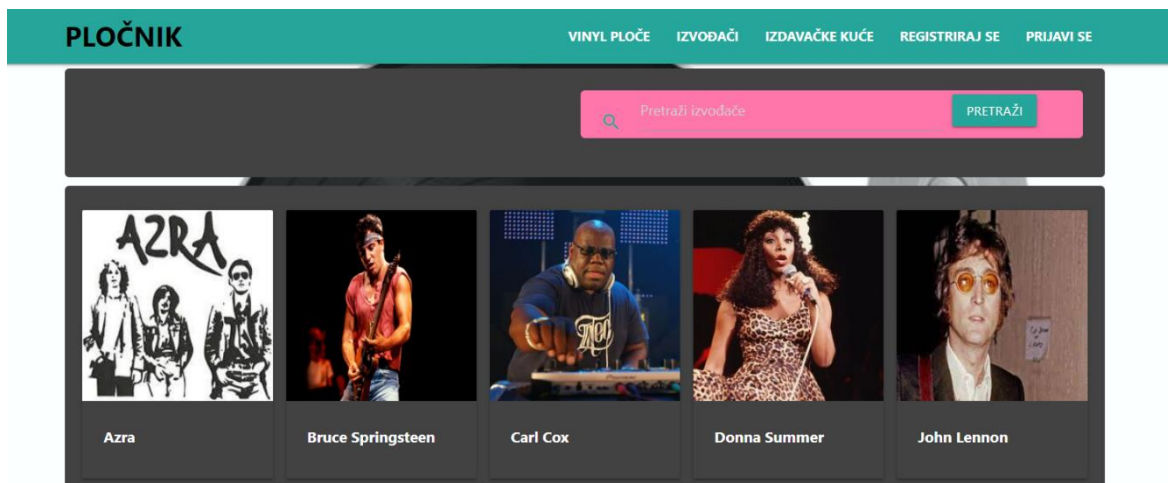
Na dnu stranice moguće je pročitati recenzije (Slika 4.6). Kao što se vidi, moguće je ostaviti samo ocjenu, tekst nije obavezan.



Slika 4.6 Recenzije

4.1.3. Pregled izvođača

Klikom na 'IZVOĐAČI' u navigacijskoj traci dolazi se do stranice s popisom svih izvođača (Slika 4.7). Izvođače je kao i ploče moguće pretražiti po nazivu. Klikom na specifičnog izvođača dolazi se do stranice s više informacija (Slika 4.8), u slučaju grupe prikazuje se početak i kraj djelovanja, a za glazbenika se prikazuju datum rođenja i smrti. Nakon kratkog opisa prikazana je diskografija izvođača koja prikazuje albume zabilježene u sustavu, klikom na naziv jednog od njih moguće je doći na stranicu tog albuma. Sa stranice albuma isto je moguće doći do stranice izvođača.



Slika 4.7 Pregled izvođača



Azra

Početak djelovanja : 1977

Kraj djelovanja : 1988

Azra je bio sastav novog vala koji je bio vrlo popularan za vrijeme 1980-ih godina na području bivše Jugoslavije. Frontmen Azre bio je Branimir "Johnny" Štulić, basist je bio Mišo Hrnjak, a bubnjar Boris Leiner. U djelu YU 100: najbolji albumi jugoslavenske rock i pop glazbe uvršteno je čak 5 Azriinih albuma, od kojih su 3 među prvih deset.

Diskografija:

Godina izdanja	Naziv ploče
1981.	Sunčana strana ulice
1980.	Azra


Slika 4.8 Informacije o izvođaču

4.1.4. Pregled izdavačkih kuća


Klikom na 'IZDAVAČKE KUĆE' u navigacijskoj traci dolazi se do popisa izdavačkih kuća (Slika 4.9), koje je također moguće pretražiti prema nazivu. Klikom na specifičnu izdavačku kuću dolazi se do stranice s više informacija (Slika 4.10) gdje su prikazani naziv, lokacija, vlasnik te kratki opis. Ispod kratkog opisa nalazi se popis albuma koje je ta kuća objavila, klikom na naziv jedne od njih moguće je doći na stranicu tog albuma. Sa stranice albuma isto je moguće doći do stranice izdavačke kuće.

PLOČNIK


VINYL PLOČE
IZVOĐAČI
IZDAVAČKE KUĆE
REGISTRIRAJ SE
PRIJAVI SE




Apple Records




Casablanca Records



Elektra

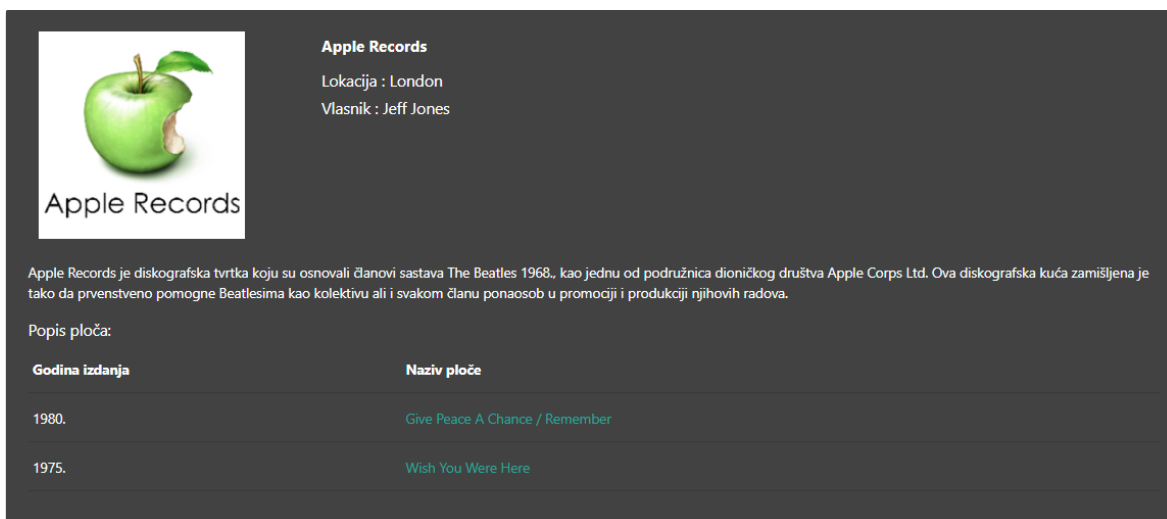


EMI Records



Fiction Records

Slika 4.9 Pregled izdavačkih kuća



Slika 4.10 Informacije o izdavačkoj kući

4.1.5. Registracija i prijava

Slika 4.11 Prozorčić za registraciju

Neprijavljeni korisnik ima mogućnost registracije ili prijave u sustav ukoliko već ima otvoren račun. Klikom na 'REGISTRIRAJ SE' korisniku se otvara prozorčić za registraciju (Slika 4.11), gdje se traži unos emaila, korisničkog imena, zaporka te ponovljene zaporka. U slučaju da se unese email ili korisničko ime koje već postoji u bazi podataka na ekranu se prikaže odgovarajuća poruka, isto vrijedi u slučaju kada svi podatci nisu uneseni ili kada se lozinke ne podudaraju.

Pri uspješnoj prijavi korisnika se preusmjeri na stranicu za prijavu do koje se može doći i klikom na 'PRIJAVI SE' u navigacijskoj traci. Tada se prikaže prozorčić (Slika 4.12) koji zahtjeva unos korisničkog imena i zaporka, ako je jedan od tih podataka neispravan

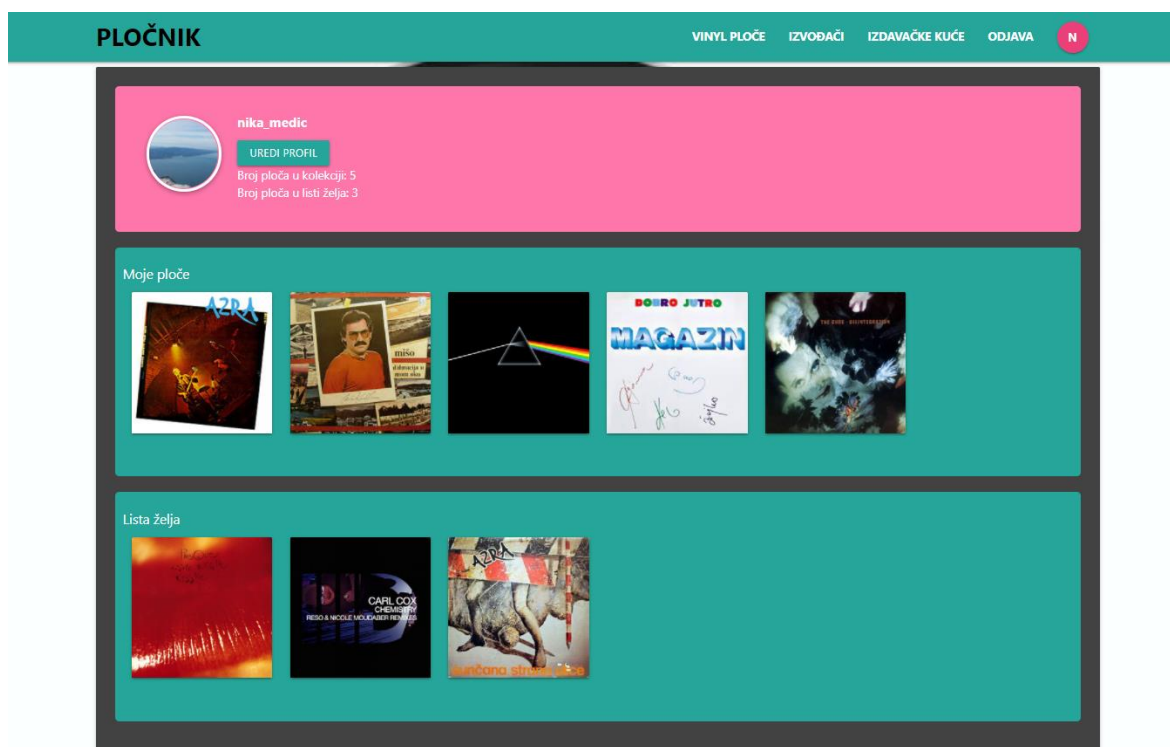
korisniku se prikazuje poruka 'Neispravno korisničko ime ili lozinka.'. Pri uspješnoj prijavi u sustav korisnik je preusmjeren na početnu stranicu.

Slika 4.12 Prozorčić za prijavu

4.2. Prijavljeni korisnik

4.2.1. Profil

Kada se uspješno prijavi korisnik ima mogućnost odlaska na svoj profil gdje može vidjeti svoj popis ploča i listu želja (Slika 4.13). Klikom na gumb 'UREDİ PROFIL' otvara se prozorčić gdje korisnik dobiva mogućnost promjene korisničkog imena ili zaporke te mogućnost brisanja vlastitog profila (Slika 4.14).




Slika 4.13 Profil prijavljenog korisnika


Uredi profil


OBRIŠI PROFIL

Email: nikamedic14@gmail.com

 Korisničko ime

nika_medic

 Nova zaporka


 Ponovi novu zaporku

POHRANI PROMJENE

Slika 4.14 Uređivanje profila

4.2.2. Dodavanje ploča na popise

Kada prijavljeni korisnik dođe na stranicu vinyl ploče osim osnovnih informacija prikazuju se i dva gumba 'DODAJ U KOLEKCIJU' i 'DODAJ NA LISTU ŽELJA'. Klikom na njih korisnik dodaje taj album na svoj popis. U slučaju da je korisnik ima neki album na jednom od popisa natpis na tom gumbu se mijenja. Ako korisnik u kolekciju doda album koji već ima na listi želja onda se taj album uklanja s liste želja. Slika 4.15 prikazuje ploču koju korisnik ima u kolekciji.



The Cure - Disintegration

Žanr: Rock Electronic

Godina izdanja: 1989

Izdavačka kuća: Fiction Records

Vrsta ploče: Long play

Verzija: FIXH 14

4.2 ★

UKLONI IZ KOLEKCIJE

DODAJ NA LISTU ŽELJA

Osmi studijski album The Cure. Dobio je dvostruko platinasti status u SAD-u, što ga čini najprodavanijim albumom benda do sada. Zbog produženog vremena trajanja albuma, "Last Dance" i "Homesick" nisu bili uključeni u originalno izdanje albuma s vinilima. Ponovno izdano 2010. godine s bonus CD-om rijetkih pjesama i "Entreat plus"

Sadržaj:

	Naziv pjesme	Trajanje	Producent
1.	Plainsong	05:16	David M. Allen
2.	Pictures Of You	07:28	David M. Allen

Slika 4.15 Pregled ploče prijavljenog korisnika

4.2.3. Postavljanje recenzije

Na dnu stranice vinyl ploče korisniku se pojavljuje mogućnost postavljanja recenzije (Slika 4.16). Odabire ocjenu od jedan do pet te po želji ostavlja tekst. U slučaju postavljanja recenzije bez označne ocjene na stranici se ispisiuje poruka 'Ocjena je obavezna!'. Pri uspješnom postavljanju recenzije pokreće se okidač 'insRecenzija' (Isječak koda 2.2) te se ponovno izračunava prosječna ocjena ploče.

Slika 4.16 Forma za postavljanje recenzije

4.3. Administrator

Administrator ima sve mogućnosti kao neregistrirani korisnik, uz to može brisati i dodavati vinyl ploče, izvođače i izdavačke kuće. Iako se prijavljuje u sustav, administrator nema mogućnost pogledati svoj profil, dodati neku ploču na popis ili postaviti recenziju.

4.3.1. Dodavanje sadržaja

Na navigacijskoj traci administratoru se pojavljuje gumb 'DODAJ SADRŽAJ' gdje odabire mogućnost dodavanja nove ploče, izvođača ili izdavačke kuće. Klikom na jednom od njih pojavljuje se odgovarajuća forma za unos. U slučaju dodavanja stranih ključeva ostvareni su padajući izbornici. U slučaju dodavanja novog izvođača korisnik odabire ili grupu ili glazbenika. Slika 4.17 prikazuje formu za dodavanje nove ploče, forme za dodavanje izvođača i izdavačke kuće su slične.

Slika 4.17 Forma za dodavanje vinyl ploče

4.3.2. Brisanje sadržaja

Na stranici ploče, izvođača ili izdavačke kuće administrator ima mogućnost obrisati odabrani sadržaj (Slika 4.18). Klikom na gumb 'OBRIŠI' iskače pop-up prozorčić postavljajući pitanje o sigurnosti odluke.



Slika 4.18 Obriši izvođača

Administrator također može dodati naslovnu sliku za odabrani sadržaj.

Zaključak

U ovom završnom radu opisane su osnovne stavke bitne za izradu ovakvog sustava: prvo su popisani svi funkcionalni i nefunkcionalni zahtjevi, zatim su rezimirane osnovne informacije bitne za stvaranje sustava baze podataka za prosječnu web-aplikaciju, nakon čega su opisani način rada sustava, arhitektura i dizajn. Na samom kraju dodane su upute za korištenje.

Glavni cilj ovog zadatka bio je ostvariti dobro organiziranu bazu podataka i sustav koji je intuitivan i jednostavan za korištenje, a u isto vrijeme zorno prikazuje sve relevantne podatke i pruža mogućnost različitih funkcionalnosti. Sustav u trenutnoj verziji ispunjava sve zadane funkcionalnosti: korisnici imaju mogućnost pretražiti vinyl ploče, izvođače i izdavačke kuće te svoje omiljene albume mogu pohraniti, ocijeniti i recenzirati.

Kroz daljnji rad na ovom sustavu mogle bi se ostvariti funkcionalnosti praćenja drugih korisnika, stvaranja vlastitih popisa za reprodukciju te po mogućnosti ostvariti dodavanje većih količina podataka kroz korisničko sučelje.

Literatura

- [1] *Nastavni materijali iz kolegija Baze podataka*, Fakultet elektrotehnike i računarstva, 2020.
- [2] *Nastavni materijali iz kolegija Programsko inženjerstvo*, Fakultet elektrotehnike I računarstva, 2020.
- [3] *PostgreSQL 13.3 Documentation*,
Poveznica: <https://www.postgresql.org/docs/13/index.html>, pristupljeno: 27.03.2021.
- [4] *Spring Boot Reference Documentation*
Poveznica: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/#legal>, pristupljeno: 10.04.2021.
- [5] *React*
Poveznica: <https://reactjs.org/>, pristupljeno: 23.04.2021.
- [6] *Materialize*
Poveznica: <https://materializecss.com/>, pristupljeno: 10.05.2021.
- [7] *Spring Framework Documentation*
Poveznica: <https://docs.spring.io/spring-framework/docs/current/reference/html/>, pristupljeno: 10.04.2021.

Sažetak

BAZA PODATAKA I WEB-APLIKACIJA ZA GLAZBENE ALBUME

U ovom radu obrađen je postupak izrade baze podataka i web-aplikacije za glazbene albume. Model baze podataka obuhvaća sve relevantne podatke vezane za glazbenu industriju, prvo je opisan ER model, koji je zatim pretvoren u relacijski i onda implementiran u sustavu PostgreSQL. Korisničko sučelje intuitivno je, sigurno i jednostavno za korištenje, omogućuje dohvat i pohranu potrebnih podataka u bazu. Implementacija korisničkog sučelja ostvarena je u radnom okviru Spring, specifično modulu Spring Boot i u JavaScript biblioteci React.

Ključne riječi: vinyl ploča, izvođač, izdavačka kuća, relacijski model, PostgreSQL, Spring Boot, React

Summary

DATABASE AND WEB APPLICATION FOR MUSIC ALBUMS

This paper deals with the process of creating a database and web application for music albums. The database model includes all relevant data related to the music industry, the ER model is described first, which is then converted to a relational one and then implemented in the PostgreSQL system. The user interface is intuitive, secure and easy to use, allowing you to retrieve and store the necessary data in the database. The implementation of the user interface was accomplished in the Spring framework, specifically the Spring Boot module, and in the React JavaScript library.

Keywords: vinyl record, artist, publishing house, relational model, PostgreSQL, Spring Boot, React