

Programsko inženjerstvo

Ak. god. 2021./2022.

My Vinyl Collection

Dokumentacija, Rev. 2.0

Grupa: *sCROZ prosječni*

Voditelj: *Josip Srzić*

Datum predaje: *14. 1. 2022.*

Nastavnik: *Vlado Sruk*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
2.1 Primjeri sličnih rješenja	6
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	23
3.2 Ostali zahtjevi	28
4 Arhitektura i dizajn sustava	29
4.1 Baza podataka	31
4.1.1 Opis tablica	31
4.1.2 Dijagram baze podataka	35
4.2 Dijagram razreda	36
4.3 Dijagram stanja	38
4.4 Dijagram aktivnosti	39
4.5 Dijagram komponenti	40
5 Implementacija i korisničko sučelje	41
5.1 Korištene tehnologije i alati	41
5.2 Ispitivanje programskog rješenja	43
5.2.1 Ispitivanje komponenti	43
5.2.2 Ispitivanje sustava	49
5.3 Dijagram razmještaja	56
5.4 Upute za puštanje u pogon	57
6 Zaključak i budući rad	60
Popis literature	62

Indeks slika i dijagrama 64

Dodatak: Prikaz aktivnosti grupe 65

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	T.Drezga	29.10.2021.
0.2	Dopisan dodatak i dnevnik promjene dokumentacije.	T.Drezga	31.10.2021.
0.3	Napisan dio Opisa projektnog zadatka.	T.Drezga	9.11.2021.
0.4	Napisan cijeli Opis projektnog zadatka i ažurirani sastanci.	T.Drezga	12.11.2021.
0.5	Napisani UC-ovi.	Toni Drezga	13.11.2021.
0.6	Dodani grafovi UC-ova i napisani tekstovi sekvencijskih dijagrama	T.Drezga	15.11.2021.
0.7	Opis arhitekture i dizajna sustava	T.Drezga	16.11.2021.
0.8	Završeni UC-ovi i sekvencijski dijagrami. Gotovo 3. poglavlje.	T.Drezga	17.11.2021.
0.9	Izmjene u dokumentaciji: dodani neki UC-ovi, neki UC-ovi prepravljani, sekvencijski dijagrami prepravljani	J.Srzić, T.Krišto, M.Dulibić, T.Drezga	18.11.2021.
0.10	Opisane tablice u bazi	P.Pavlić, T.Krišto, M.Dulibić, T.Drezga	18.11.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.11	Izrađeni dijagrami razreda	L.Aleksić, M.Dulibić, T.Drezga	19.11.2021.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	T.Drezga	19.11.2021.
1.1	Uređivanje dokumentacije: popravak pogreški i izmjene u UC i sekvencijskim dijagramima	T.Drezga	29.12.2021.
1.2	Napravljen dijagrami aktivnosti	T.Drezga	3.1.2022.
1.3	Napravljen dijagrami stanja	T.Drezga	4.1.2022.
1.4	Napravljen dijagrami razmještaja	T.Drezga	5.1.2022.
1.5	Napravljen dijagrami komponenti	T.Drezga	6.1.2022.
1.6	Ispitivanje sustava	T.Drezga	10.1.2022.
1.7	Ispitivanje komponenti	T.Krišto	11.1.2022.
1.8	Napisane upute za puštanje u pogon	M.Dulibić	12.1.2022.
1.9	Napisan zaključak i dodan dijagram pregleda promjena	T.Drezga	13.1.2022.
1.10	Zadnji pregled i prepravci dokumentacije	J.Srzić, L.Aleksić, T.Krišto, M.Dulibić, T.Drezga	13.1.2022.
2.0	Završna verzija dokumentacije	T.Drezga	14.1.2022.

2. Opis projektnog zadatka

Cilj ovog projekta je stvaranje web aplikacije "My Vinyl Collection" koja će služiti kao mjesto na kojem će korisnici, koji su zainteresirani, ljubitelji ili kolekcionari vinila, pronalaziti nove vinile, prodavati, kupovati ili razmjenjivati vinile s drugima. U planu nam je olakšati korisnicima traženje tako da korištenjem njihove lokacije prikazujemo oglase u njihovoj blizini.

Neregistrirani korisnici mogu samo pregledavati i pretraživati tuđe kolekcije i korisnike. Neregistriranom korisniku je omogućeno prijavljivanje postojećim računom (upisivanjem korisničkog imena i lozinke) ili kreiranjem novog računa. Za kreiranje novog računa potrebni su:

- ime
- prezime
- korisničko ime
- e-mail adresa
- lozinka
- potvrda lozinke
- odabir preferiranih žanrova
- lokacija uređaja

Registracijom u sustav korisniku se dodjeljuju određene mogućnosti i značajke.

Registrirani korisnici imaju mogućnosti pregledavanja, mijenjanja osobnih podataka i brisanja profila. Također mogu izrađivati kolekcije i podkolekcije vinila koje posjeduju, objavljivati oglase za prodaju ili razmjenu vinila te imaju pregled kupljenih, aktivnih i prodanih oglasa, tj. vinila. Prilikom objavljivanja oglasa, ako korisnik nema odabranu lokaciju na profilu, od njega se traži odabir lokacije vinila. S obzirom na kojoj su korisnici lokaciji, koju će moći promijeniti ručno u postavkama, na početnoj stranici će im se prikazivati svi vinili žanrova koje je korisnik odabrao pri registraciji. Na početnoj stranici će također korisnici imati mogućnost pretraživanja drugih korisnika, pretraživanja i filtriranja oglasa i kolekcija, mogućnost praćenja drugih korisnika te označavanja njihovih vinila sa

”Sviđa mi se”. Prilikom dodavanja vinila u kolekcije i podkolekcije korisnici će biti obvezni napisati naziv izvođača, naziv albuma, godinu izdanja, žanr vinila te osnovne parametre o samom fizičkom vinilu kao što su: promjer, RPM (*revolutions per minute*), kapacitet, kvaliteta reprodukcije i broj audio kanala. Vezano uz glavne karakteristike vinila korisnici mogu opcionalno napisati podžanr, ocjenu stanja, sliku, te kao dodatak raritet, opis i vrijednost vinila u kunama. Svaka vrijednost vinila će automatski biti preračunata i prikazana, ispod cijene u kunama, u eurima. Kad je registrirani korisnik zainteresiran za oglas može napisati poruku vlasniku oglasa kojem se poruka šalje na mail s kojim se registrirao. E-mail adresa za primanje poruka i sličnih obavijesti se može promijeniti u postavkama te može se razlikovati od registracijskog e-maila.

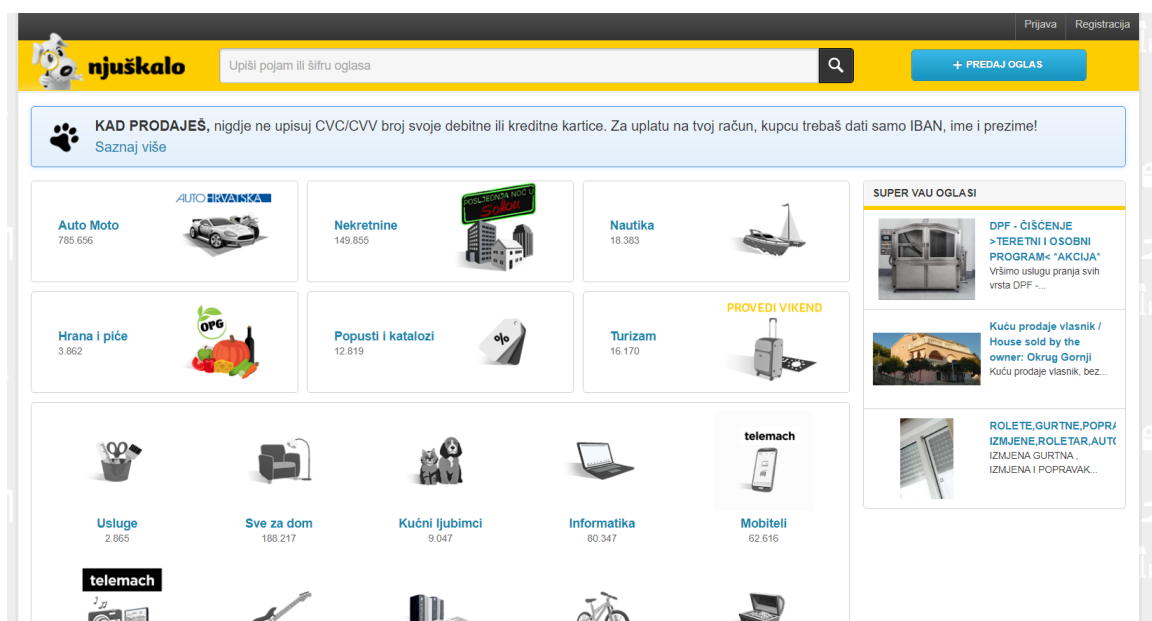
Tijekom razmjene ili kupovine, tj. prodaje kad vinil promijeni vlasnika, ne nalazi se više u kolekciji prodavatelja. Na profilu korisnika postoji lista svih vinila koje je korisnik prodao ili zamijenio.

Uloga *administratora* je potrebna za vođenje računa o objavljenim oglasima i o tome kakav je objavljeni sadržaj. Administratori imaju ovlasti brisanja oglasa, blokiranja i aktiviranja korisničkog računa. Također su u mogućnosti kreiranja događaja iz vanjskih usluga, kao što su *Facebook*, *Instagram*, *Twitter* i slično, o raznim događanjima vezanim za ovu temu (npr. buvljaci, slušaone i slično). Tijekom kreiranja svakog događaja registriranim korisnicima će biti ponuđene osnovne informacije kao što su naslov, slike, datum kratki opis te link na događaj.

2.1 Primjeri sličnih rješenja

1. Njuškalo

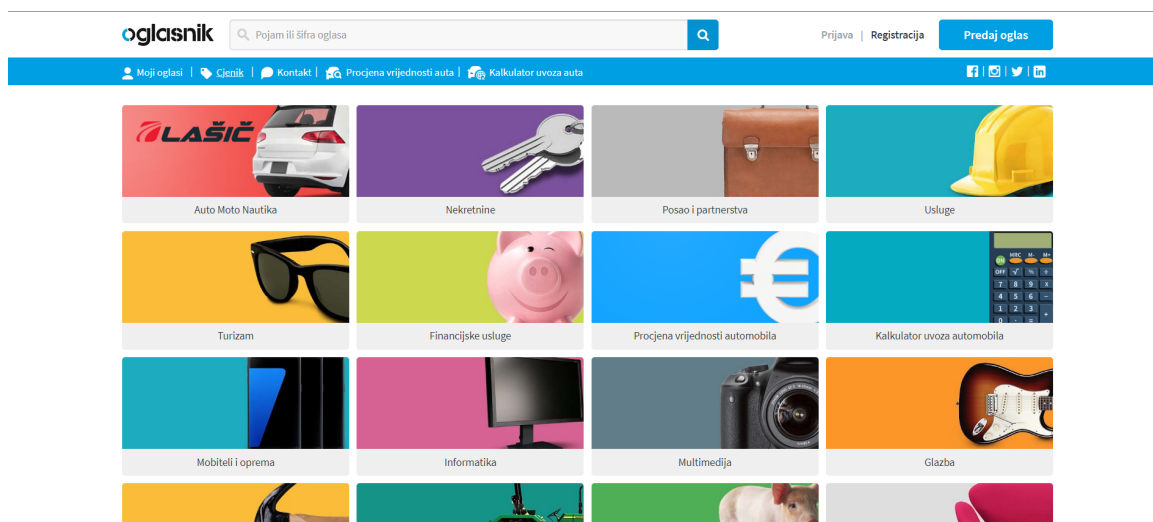
Njuškalo je web stranica na kojoj se objavljuju raznovrsni oglasi od automobila i drugih motornih vozila, nekretnina, pa sve do hrane i pića i popusta u raznim trgovinama. Služi kao mjesto za trgovinu ne samo za korisnike nego i za tvrtke, obrtnike, udruge te agencije, dok je naš projekt web stranica za ljubitelje, kolekcionare i prodavače vinila.



Slika 2.1: Početna stranica Njuškala

2. Oglasnik

Oglasnik je slična web stranica kao i Njuškalo koja ima malo drugačijih značajki za razliku od naših. Od mogućnosti procjene vrijednosti auta, kalkulatora uvoza vozila, online plaćanja pa do korisničke podrške, pravila o postupanju kolačića, uvjeta korištenja i uvjeta kupnje. Također postoji opcija prijave oglasa u slučaju da krši pravila oglašavanja, mogućnost pronalaska ostalih oglasa u blizini (koji se na karti prikazu ovisno o tome koliko se umanju/uveća karta).



Slika 2.2: Početna stranica Oglasnika

Sljedeći korak bi mogao biti proširenje na oglašavanje, tj. prodaju, kupovinu ili razmjenu gramofona te suradnja s tvrtkama, obrtnicima i ostalima za promoviranje njihovih proizvoda i usluga.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnici
 - (a) Neregistrirani
 - (b) Registrirani
2. Administrator
3. Baza podataka

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni (inicijator) može:
 - (a) pregledavati i pretraživati tuđe kolekcije
 - (b) pregledavati i pretraživati korisnike
 - (c) registrirati se u sustav, stvoriti novi korisnički račun za koji su mu potrebni ime, prezime, korisničko ime, e-mail adresa, lozinka, odabir preferiranih žanrova i lokacija uređaja
2. Registrirani/prijavljeni (inicijator) može:
 - (a) pregledavati i mijenjati osobne podatke
 - i. promijeniti e-mail adresu za primanje poruka
 - (b) izbrisati profil
 - (c) stvarati kolekcije i podkolekcije vinila
 - (d) objavlјivati oglase za prodaju ili razmjenu
 - (e) pregledati kupljene, aktivne i prodane oglase (vinile)
 - (f) dodavati vinile u kolekcije i podkolekcije uz koje će biti obvezni napisati naziv izvođača, albuma, godinu izdanja te žanr vinila
 - i. napisati osnovne parametre fizičkog vinila: promjer, RPM, kapacitet, kvaliteta reprodukcije, broj audio kanala

- ii. opcionalno napisati glavne karakteristike vinila: podžanr, ocjena stanja, slika, raritet, opis, cijena
- (g) označiti oglas (vinil) kao da je prodan da se više ne prikazuje na stranici
- (h) pratiti druge korisnike
- (i) označiti vinile sa "Sviđa mi se"
- (j) poslati poruku korisniku za zainteresiranost u oglas

3. Administrator (inicijator) može:

- (a) vidjeti popis svih registriranih korisnika i njihovih podataka
- (b) blokirati i aktivirati korisnički račun
- (c) kreirati događaje iz vanjskih usluga (npr. Facebook, Instagram, Twitter i slično) o raznim događanjima (npr. buvljaci, slušaone i slično)

4. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o vinilima, njihovim karakteristikama i vlasnicima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pregledaj kolekcije

- **Glavni sudionik:** Neregistrirani, registrirani korisnik
- **Cilj:** Pregledati tuđe vinile
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik se nalazi na početnoj stranici
- **Opis osnovnog tijeka:**
 1. Na početnoj stranici prikazani su oglasi koji su u blizini korisnika
 2. Korisnik odabire tražilicu
 3. Korisnik upisuje ime kolekcije u tražilicu
 4. Aplikacija izbací rezultat pretrage
 5. Korisnik pregledava prikazane kolekcije

UC2 - Pregledaj korisnika

- **Glavni sudionik:** Neregistrirani, registrirani korisnik, administrator
- **Cilj:** Pregledati ostale korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik se nalazi na početnoj stranici
- **Opis osnovnog tijeka:**
 1. Korisnik odabire tražilicu
 2. Korisnik upisuje korisničko ime u tražilicu
 3. Aplikacija izbací rezultat pretrage
 4. Korisnik pregledava prikazanog korisnika
- **Opis mogućih odstupanja:**
 - 3.a Korisničko ime je krivo napisano ili ne postoji
 - * Sustav obavještava korisnika o neuspjeloj pretrazi

UC3 - Registriraj se

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik se nalazi na početnoj stranici
- **Opis osnovnog tijeka:**

1. Korisnik odabire gumb za registraciju
 2. Aplikacija otvori stranicu za registraciju
 3. Korisnik unosi potrebne korisničke podatke
 4. Aplikacija verificira korisničke podatke
 5. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 4.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkih podataka u nedozvoljenom formatu ili pružanje neispravnih podataka
 - * Sustav obavještava korisnika o neuspjeloj registraciji i vraća ga na stranicu registracije
 - * Korisnik mijenja podatke i završava registraciju ili odustaje od registracije

UC4 - Prijavi se

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i nalazi se na početnoj stranici
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb za prijavu
 2. Aplikacija otvara stranicu za prijavu
 3. Korisnik unosi korisničko ime i lozinku
 4. Aplikacija verificira korisničko ime i lozinku
 5. Nakon uspješne prijave aplikacija dovodi korisnika na početnu stranicu
- **Opis mogućih odstupanja:**
 - 4.a Neispravno korisničko ime/lozinka
 - * Sustav obavještava korisnika o neuspjeloj prijavi

UC5 - Promjeni osobne podatke

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promijeniti/ažurirati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u aplikaciji
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled profila
 2. Otvara se stranica s osobnim podacima korisnika

3. Korisnik mijenja podatke
 4. Korisnik sprema promijene
 5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 4.a Korisnik nije spremio izmjene
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC6 - Obriši korisnički račun

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Izbrisati svoj korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled profila
 2. Otvara se stranica s osobnim podacima korisnika
 3. Korisnik bira opciju brisanja profila
 4. Korisnik sprema promijene
 5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 4.a Korisnik nije spremio podatke
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC7 - Pregledaj sve oglase

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pogledati sve oglase
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled oglasa
 2. Otvara se stranica sa svim oglasima
 3. Korisnik filtrira koje oglase hoće pregledati
 4. Aplikacija prikazuje filtrirane oglase
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nema objavljenih oglasa

UC8 - Dodaj oglas

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Dodati novi oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb za pregled oglasa
 2. Otvara se stranica sa svim oglasima
 3. Korisnik odabire gumb za dodavanje oglasa
 4. Korisnik odabire vinil iz liste vinila
 5. Korisnik sprema promijene
 6. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 5.a Korisnik nije spremio podatke
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC9 - Uredi oglas

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Urediti/ažurirati oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima barem jedan objavljeni oglas
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled oglasa
 2. Otvara se stranica sa svim oglasima
 3. Korisnik odabire opciju uređivanja nad određenim oglasom
 4. Korisnik dodaje/izmjenjuje informacije u oglasu
 5. Korisnik sprema promijene
 6. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 5.a Korisnik nije spremio promjene
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC10 - Ukloni oglas

- **Glavni sudionik:** Registrirani korisnik, administrator
- **Cilj:** Ukloniti oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima objavljen oglas (samo za registriranog korisnika)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled oglasa
 2. Otvara se stranica sa svim oglasima
 3. Korisnik odabire opciju uklanjanja nad određenim oglasom
 4. Korisnik sprema promijene
 5. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 5.a Korisnik nije spremio promjene
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC11 - Označi vinil sa "Sviđa mi se"

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Označiti vinil koji se korisniku sviđa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i vinil nije označen sa "Sviđa mi se"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled svih vinila
 2. Otvara se stranica sa svim vinilima
 3. Korisnik odabire opciju "Sviđa mi se"
 4. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Korisnik je već označio oglas sa "Sviđa mi se"
 - * Sustav miče oznaku "Sviđa mi se" sa vinila

UC12 - Stvori podkolekciju

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Grupirati vinile u novu podkolekciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**

1. Korisnik odabire izvođača iz liste
 2. Otvara se stranica sa svim oglasima
 3. Korisnik odabire opciju stvaranja podkolekcije
 4. Korisnik dodaje vinile u podkolekciju
 5. Korisnik sprema promijene
 6. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 5.a Korisnik nije spremio promjene
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC13 - Uredi kolekciju/podkolekciju

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Uređivati/ažurirati vinile u kolekciji/podkolekciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima barem jednu kolekciju/podkolekciju
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled oglasa
 2. Otvara se stranica sa svim oglasima i kolekcijama/podkolekcijama
 3. Korisnik odabire opciju uređivanja određene kolekcije/podkolekcije
 4. Korisnik dodaje/uklanja vinile u/iz kolekciju/podkolekciju
 5. Korisnik sprema promijene
 6. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 5.a Korisnik nije spremio promjene
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC14 - Ukloni podkolekciju

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Izbrisati kolekciju/podkolekciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima barem jednu kolekciju/podkolekciju
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled oglasa
 2. Otvara se stranica sa svim oglasima i kolekcijama/podkolekcijama

3. Korisnik odabire opciju brisanja kolekcije/podkolekcije
 4. Korisnik sprema promijene
 5. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Korisnik nije spremio promjene
 - * Sustav obavještava korisnika da nije spremio podatke te nudi korisniku dvije opcije: "Odustani" i "Vrati se i spremi"

UC15 - Javi se na oglas

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Javiti drugom korisniku za interesiranje u oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire željeni oglas
 2. Korisnik odabire korisnički profil iz željenog oglasa
 3. Aplikacija prikazuje stranicu odabranog korisnika
 4. Korisnik čita e-mail adresu korisnika
 5. Korisnik šalje e-mail željenom korisniku preko druge aplikacije za slanje e-mail poruka

UC16 - Zaprati korisnika

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Zaprati ostale registrirane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire korisnički profil iz oglasa ili pretražuje određeni profil
 2. Korisnik na odabranom/pretraženom profilu stiže opciju praćenja korisnika
 3. Promjene se upisuju u bazu podataka
- 4. **Opis mogućih odstupanja:**
 - 2.a Korisnik već prati odabranog korisnika
 - * Sustav uklanja stanje praćenja i korisnik više ne prati odabranog korisnika

UC17 - Blokiraj korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Blokirati korisnika koji je prekršio bilo koja pravila
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator uočava prekršaj pravila
 2. Administrator odabire gumb za pregledavanje svih korisnika
 3. Administrator odabire željenog korisnika
 4. Administrator blokira željenog korisnika
 5. Administrator sprema promjene
 6. Promjene se spremaju u bazu podataka
- **Opis mogućih odstupanja:**
 - 5.a Administrator nije spremio promjene
 - * Sustav obavještava administratora da nije spremio podatke te nudi administratoru dvije opcije: "Odustani" i "Vrati se i spremi"

UC18 - Aktiviraj korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Aktivirati blokirane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregledavanja svih korisnika
 2. Administrator filtrira sve korisnike samo na blokirane
 3. Administrator pronalazi željenog blokiranog korisnika
 4. Administrator aktivira željenog korisnika
 5. Administrator sprema promjene
 6. Promjene se spremaju u bazu podataka

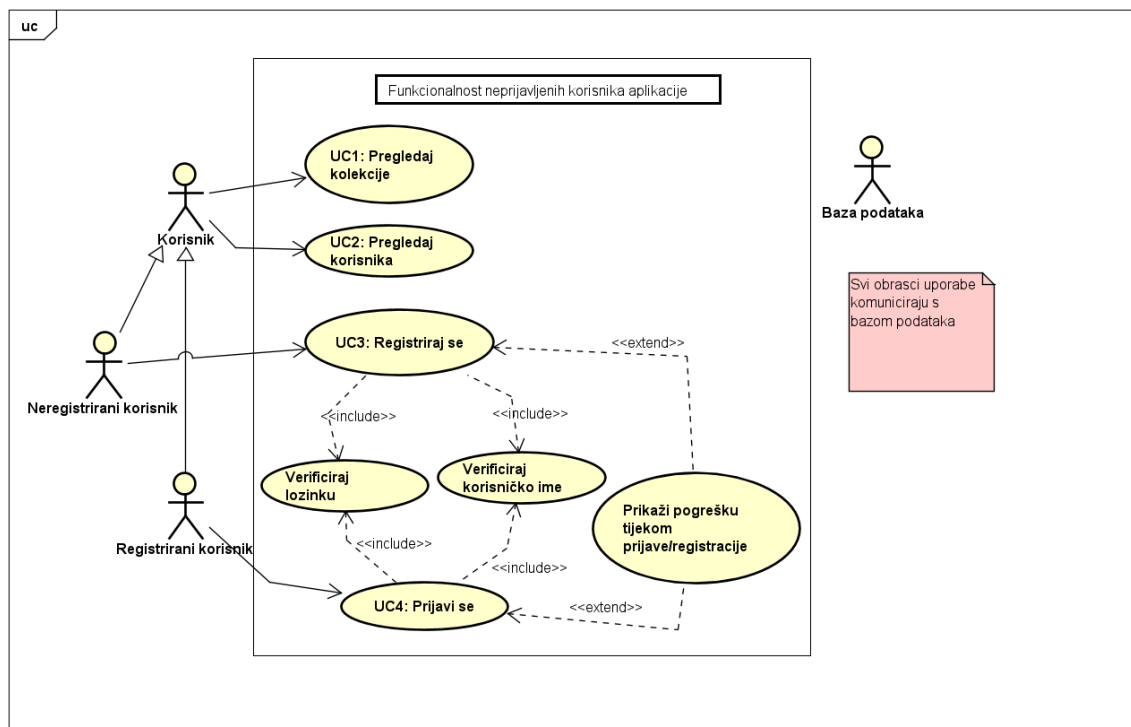
UC19 - Kreiraj događaj

- **Glavni sudionik:** Administrator
- **Cilj:** Kreirati događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen

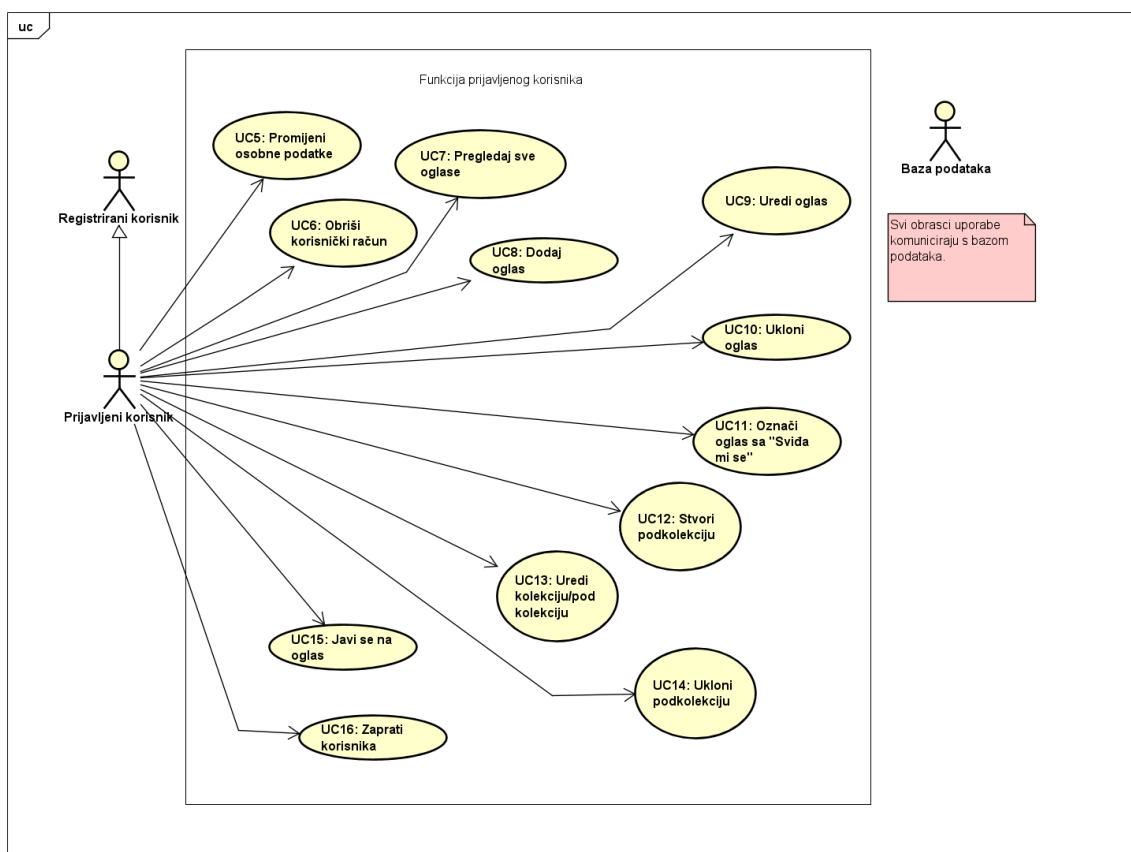
- **Opis osnovnog tijeka:**

1. Administrator odabire opciju stvaranja događaja
2. Administrator upisuje datum, naziv događaja, stavlja slike, kratki opis i link na događaj
3. Administrator kreira događaj
4. Aplikacija stavlja događaj na početnu stranicu
5. Promjene se upisuju u bazu podataka

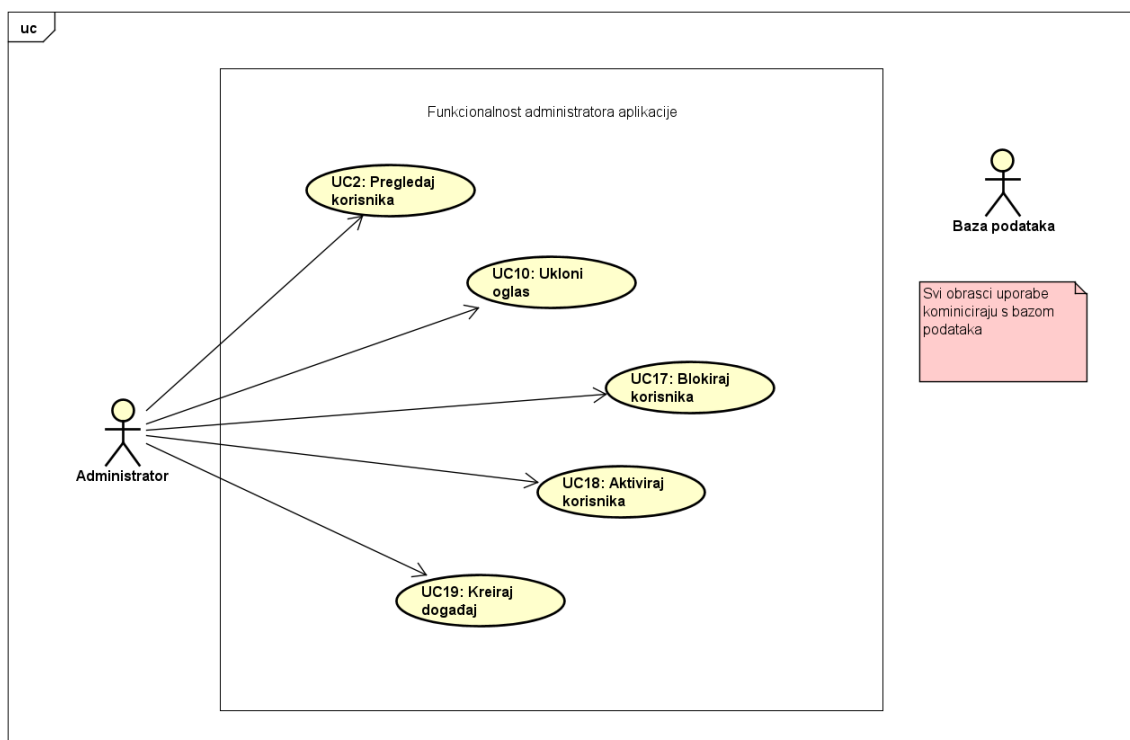
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost neprijavljenog korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost prijavljenog korisnika

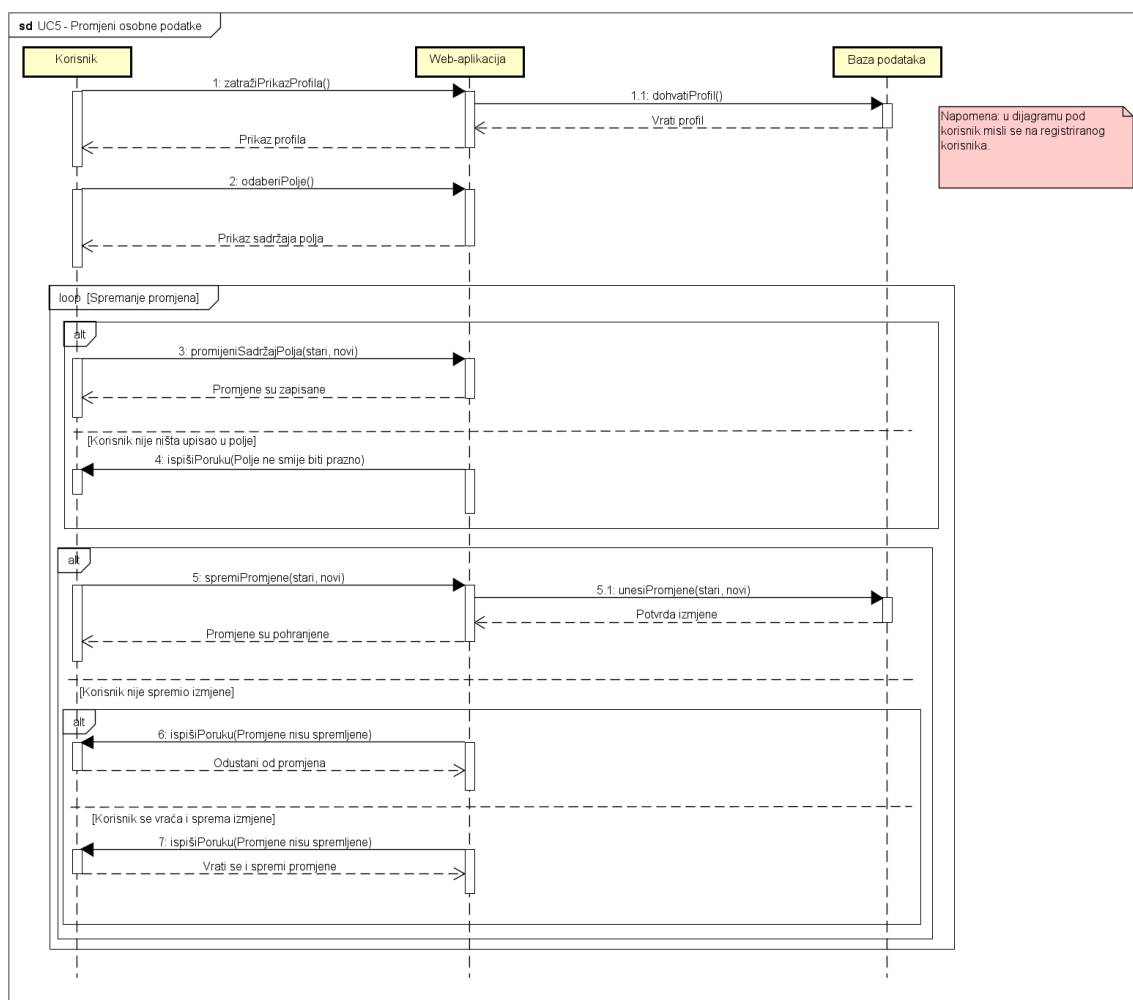


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC5 - Promjena osobnih podataka

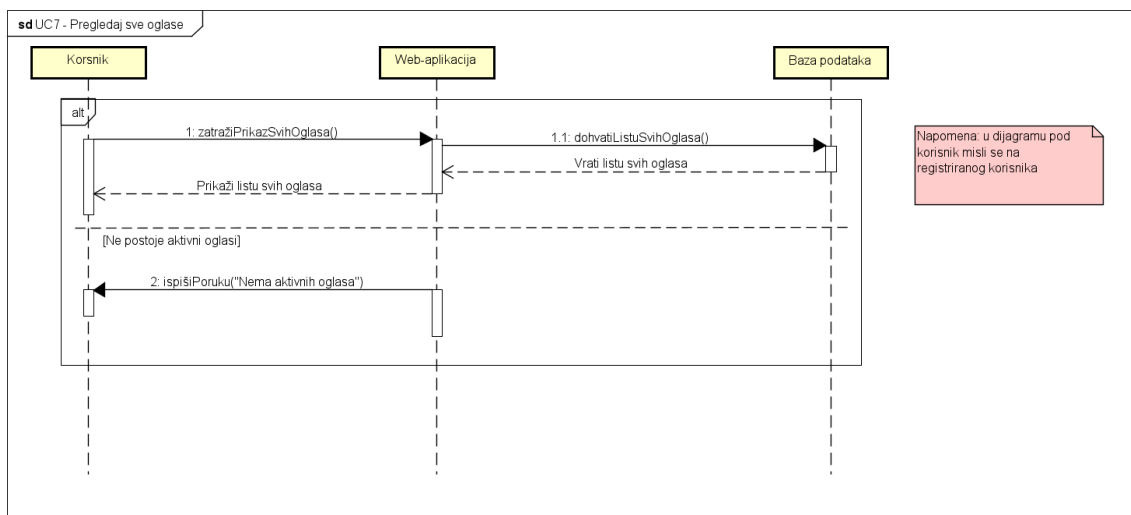
Korisnik šalje zahtjev za odabir svojeg profila. Poslužitelj dohvaća podatke iz baze podataka i prikazuje ih korisniku. Korisnik odabire polje i šalje zahtjev za promjenu osobnih podataka. Poslužitelj mu daje pristup i korisnik unosi nove osobne podatke. Kad korisnik završi sa izmjenom podataka i spremi promjenu, poslužitelj izmjeni podatke u bazi podataka i vraća potvrdu. Ukoliko korisnik nije spremio promjenu ispisuje se poruka prilikom koje može nastaviti dalje i ne spremiti promjene ili vratiti se nazad i spremiti promjene.



Slika 3.4: Sekvencijski dijagram za UC5

Obrazac uporabe UC11 - Pregledaj sve oglase

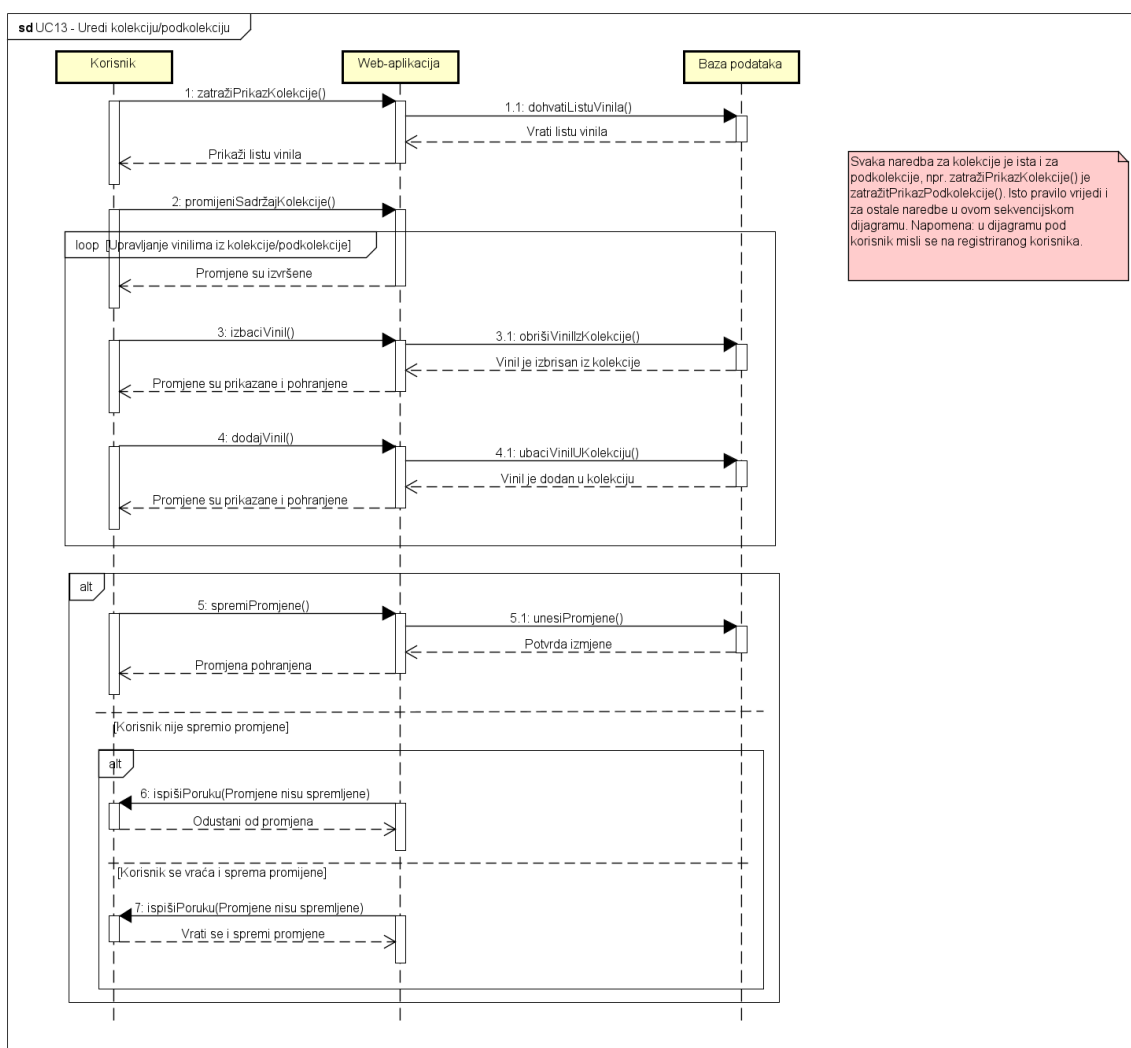
Korisnik šalje zahtjev za prikaz stranice sa svim oglasima. Poslužitelj dohvaća stranicu i prikazuje sve rezultate. Odabirom na određeni oglas, poslužitelj dohvaća osnovne podatke iz baze podataka o oglasu i prikazuje ih klijentu.



Slika 3.5: Sekvencijski dijagram za UC7

Obrazac uporabe UC13 - Uredi kolekciju/podkolekciju

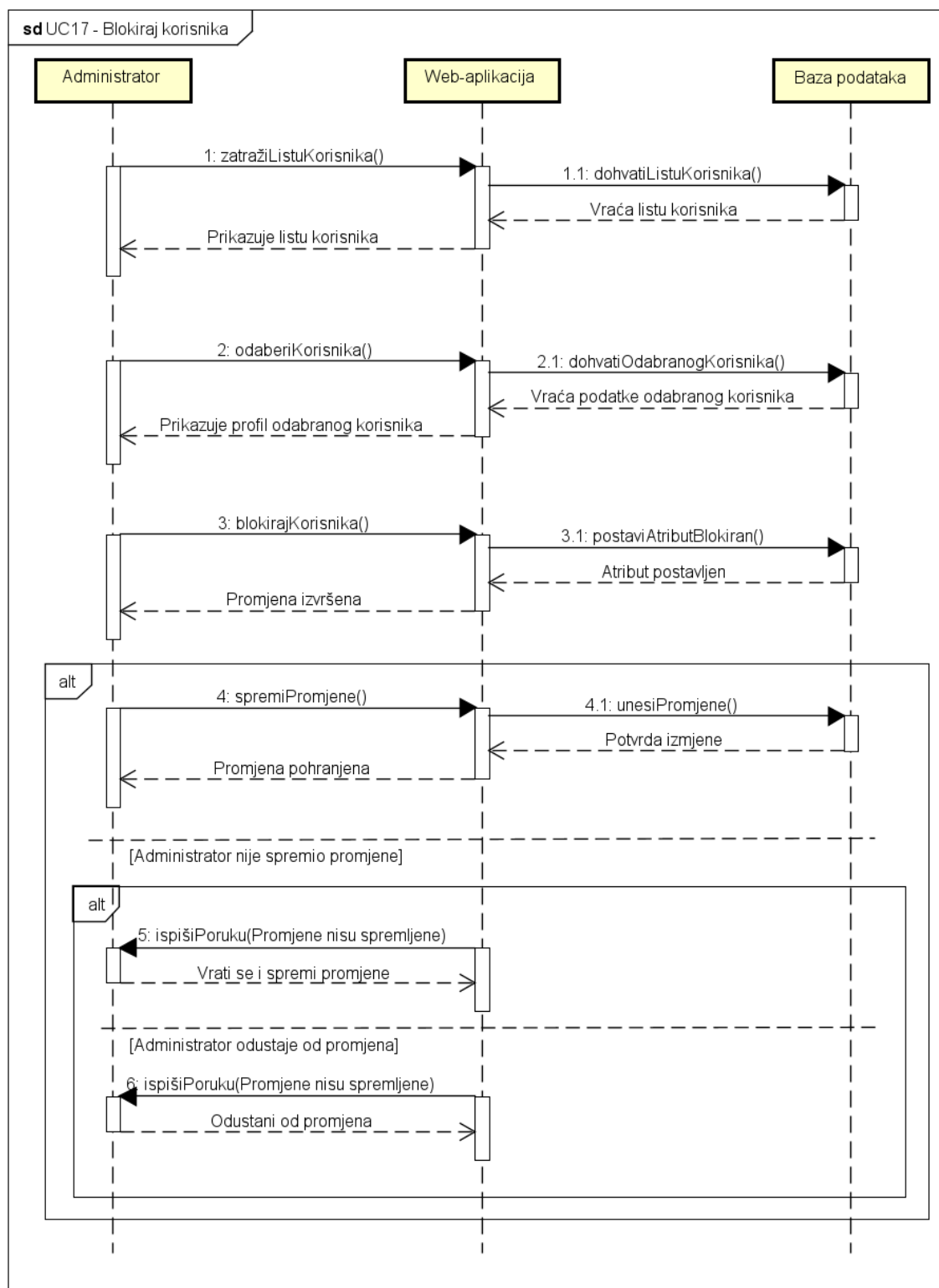
Korisnik šalje zahtjev za odabir određene kolekcije/podkolekcije. Poslužitelj dohvaća podatke iz baze podataka te prikazuje ih korisniku. Korisnik šalje zahtjev za promjenu proizvoda. Poslužitelj mu daje pristup i korisnik dodaje/izbacuje vinile iz kolekcije/podkolekcije te sprema promjenu. Poslužitelj izmjeni proizvod u bazi podataka koja vraća potvrdu. Ukoliko korisnik nije spremio promjenu ispisuje se poruka prilikom koje može nastaviti dalje i ne spremiti promjene ili vratiti se nazad i spremiti promjene.



Slika 3.6: Sekvencijski dijagram za UC13

Obrazac uporabe UC17 - Blokiraj korisnika

Administrator šalje zahtjev za listu korisnika. Polužitelj dohvaća popis iz baze podataka te ih prikazuje administratoru. Administrator odabire željenog korisnika. Polužitelj dohvaća podatke korisnika iz baze podataka i prikazuje ih administratoru. Administrator šalje zahtjev za blokiranje korisnika. Poslužitelj mu daje pristup i administrator blokira korisnika te sprema promjene. Poslužitelj izmjeni podatke korisnika u bazi podataka i vraća potvrdu. Ukoliko administrator nije spremio promjenu ispisuje se poruka prilikom koje može nastaviti dalje i ne spremiti promjene ili vratiti se nazad i spremiti promjene.



Slika 3.7: Sekvencijski dijagram za UC17

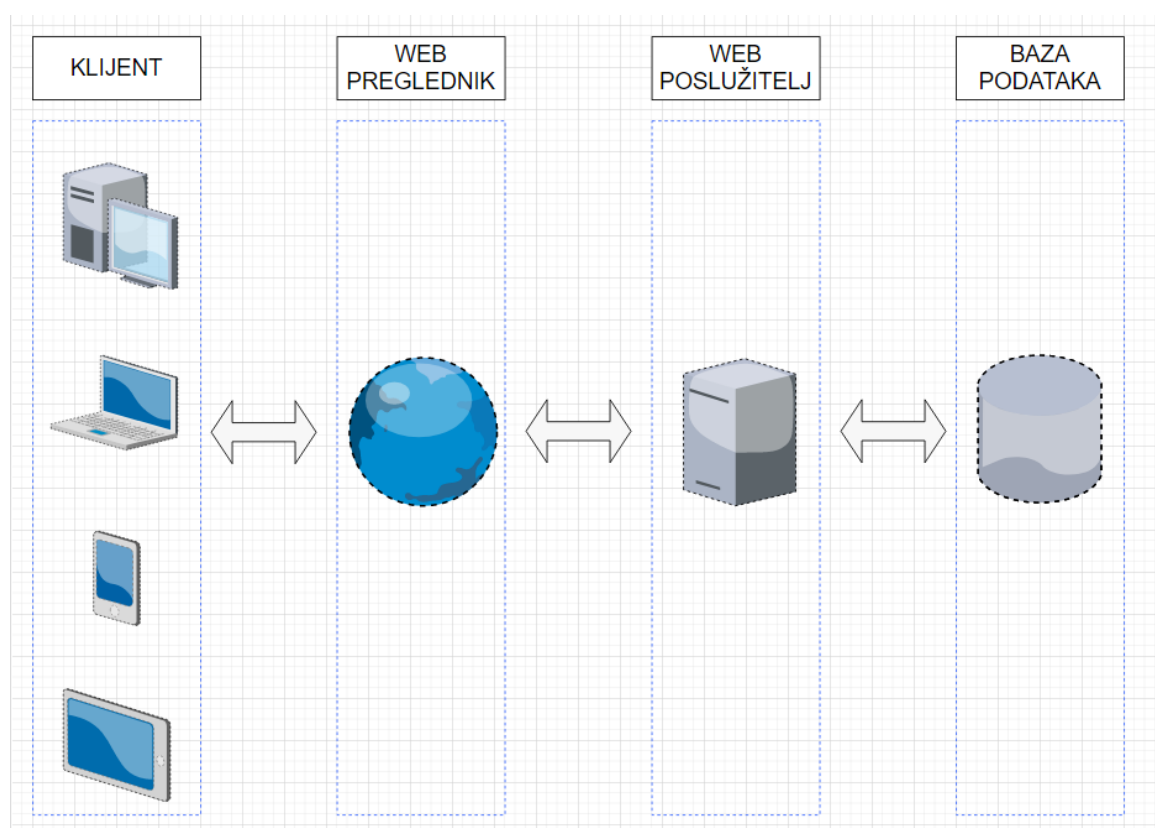
3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnosti i rad sustava
- Sustav treba biti implementiran kao web aplikacija koristeći skriptne jezike
- Veza s bazom podataka mora biti zaštićena, brza i otporna na greške.
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati posotjeće funkcionalnosti sustava
- Pristup bazi podataka ne smije dugo trajati
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS.

4. Arhitektura i dizajn sustava

Arhitektura sustava koju smo koristili se dijeli na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Web preglednik je aplikacija koja omogućuje korisniku pristup *World Wide Web*-u. Kada korisnik zatraži web stranicu, preglednik dohvaća potreban sadržaj, web stranicu i potrebne multimedijske sadržaje, s web poslužitelja i prikazuje na korisnikovom uređaju. Web stranica je hipertekstualni dokument koji se prikazuje korisniku na ekranu. Web stranice se sastoje od jedne ili više tekstualnih datoteka

pisanih u jezicima: HTML (*HyperText Markup Language*), CSS (*Cascade Style Sheets*) te JS (*JavaScript*), koje web preglednik prevodi i prikazuje ih korisniku koji putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj je računalni hardverski softver koji služi za komunikaciju klijenta s aplikacijom. Poslužitelj prima zahtjeve preko HTTP (*Hyper Text Transfer Protocol*) mrežnog protokola stvorenog za distribuciju web sadržaja. Korisnik preko web preglednika pokreće komunikaciju postavljanjem željenih zahtjeva koristeći HTTP, a web poslužitelj, ovisno o zahtjevu koji obrađuje, odgovara sadržajem tog resursa (HTML dokumentom) ili porukom o pogrešci.

Prilikom izrade projekta korišteni su sustav PostgreSQL (za upravljanje relacijskom bazom podataka), programski jezik Java i razvojni okvir Spring Boot (backend) te programski jezik JavaScript i razvojni okvir React (frontend). Odabrano razvojno okruženje je IntelliJ te cloud platforma Heroku (za deploy aplikacije). Arhitektura sustava temelji se na MVC (*Model-View-Controller*) obrascu koji funkcionira na načelu dizajna SoC (*Separation of Concerns*). SoC omogućava odvajanje programa u različite odjeljke tako da se svaki odjeljak bavi svojim dijelom, što olakšava izradu i daljnji razvoj aplikacije.

MVC sastoji se od:

- **Model** - središnja komponenta sustava koja izravno upravlja podacima, logikom i pravilima aplikacije. Neovisna je o korisničkom sučelju te prima ulazne podatke od Controllera.
- **View** - definira kako se podaci aplikacije trebaju prikazivati
- **Controller** - zadužen je za primanje i uređivanje zahtjeva koje prosljeđuje na Model ili View. Upravlja korisničkim zahtjevima i na temelju njih izvodi daljnje korake potrebne s ostalim elementima.

4.1 Baza podataka

Za potrebe sustava našeg projekta korištena je relacijska baza podataka. Osnovna građa baze je relacija, odnosno tablica koja je definirana imenom i skupom atributa. Glavni zadatak baze podataka je brza i jednostavna pohrana, izmjena i dohvat podataka za obradu. Baza podataka ove aplikacije sastoji se od tablica:

- User
- Genre
- Subgenre
- Vinyl
- Artist
- Ad
- Sale ad
- Exchange ad
- Event

4.1.1 Opis tablica

User Ovaj entitet sadrži sve važne informacije o korisniku aplikacije. Sadrži attribute: `user_id`, `name`, `surname`, `username`, `password`, `email`, `contact_email` i `location`. Ovaj entitet je u vezi *Many-to-Many* sa samim sobom preko atributa `user_id` korisnika, *Many-to-Many* sa entitetom `Genre` preko `genre_id`, *One-to-Many* sa entitetom `Ad` preko `user_id`, *Many-to-Many* sa `Vinyl` preko `user_id` te *One-to-Many* sa `Vinyl` preko `vinyl_id` i *Many-to-Many* sa `Artist` preko `user_id`.

User		
<code>user_id</code>	LONG	Jedinstveni identifikator korisnika
<code>name</code>	STRING	Ime korisnika
<code>surname</code>	STRING	Prezime korisnika
<code>username</code>	STRING	Korisničko ime
<code>password</code>	STRING	Lozinka
<code>email</code>	STRING	E-mail korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

User		
contact_email	STRING	E-mail za stupanje u kontakt s korisnikom
location	GEOGRAPHY	Geografska lokacija uređaja korisnika

Genre Ovaj entitet sadrži sve važne informacije o žanrovima. Sadrži attribute: `genre_id` i `name`. Ovaj entitet je u vezi *Many-to-Many* sa User preko atributa `user_id`, *One-to-Many* sa Subgenre preko `genre_id` i *One-to-Many* sa Vinyl preko `genre_id`.

Genre		
genre_id	LONG	Jedinstveni identifikator žanra
name	STRING	Ime žanra

Subgenre Ovaj entitet sadrži sve važne informacije o podžanrovima. Sadrži attribute: `subgenre_id`, `name` i `parent_genre_id`. Ovaj entitet je u vezi *Many-to-One* sa Genre preko atributa `genre_id`.

Subgenre		
subgenre_id	LONG	Jedinstveni identifikator podžanra
name	STRING	Ime podžanra
parent_genre_id	LONG	Identifikator roditeljskog žanra

Vinyl Ovaj entitet sadrži sve važne informacije o vinilima. Sadrži attribute: `vinyl_id`, `album_name`, `release_year`, `genre_id`, `subgenre_id`, `image`, `condition_evaluation`, `isRare`, `description`, `value_kn`, `RPM`, `diameter`, `capacity`, `reproduction_quality`, `nmb_of_audio_channels`, `time_of_reproduction`. Ovaj entitet je u vezi *One-to-Many* sa Ad preko atributa `vinyl_id`, *Many-to-Many* sa User preko `user_id`, *Many-to-One* sa User preko `vinyl_id`, *Many-to-One* sa Genre preko `parent_genre_id` i *Many-to-One* sa Artist preko `artist_id`.

Vinyl		
vinyl_id	LONG	Jedinstveni identifikator vinila
album_name	STRING	Ime albuma
release_year	INT	Godina izdanja
genre_id	LONG	Jedinstveni identifikator žanra
subgenre_id	LONG	Jedinstveni identifikator podžanra
image	BYTEA	Slika vinila
condition_evaluation	INT	Ocjena stanja vinila
isRare	BOOLEAN	Raritet vinila
description	STRING	Kratki opis vinila
value_kn	NUMERIC	Vrijednost vinila
RPM	STRING	Brzina okretaja u minuti
diameter	FLOAT	Promjer vinila
capacity	STRING	Kapacitet vinila
reproduction_quality	STRING	Kvaliteta reprodukcije vinila
nmb_of_audio_channels	INT	Broj audio kanala
time_of_reproduction	TIME	Vrijeme reprodukcije

Artist Ovaj entitet sadrži sve važne informacije o izvođaču. Sadrži attribute: artist_id i name. Ovaj entitet je u vezi *Many-to-Many* sa User preko atributa user_id i *One-to-Many* sa Vinyl preko artist_id.

Artist		
artist_id	LONG	Jedinstveni identifikator izvođača
name	STRING	Ime izvođača

Ad Ovaj entitet sadrži sve važne informacije o oglasu. Sadrži attribute: `ad_id`, `vinyl_id`, `sale_ad_id`, `exchange_ad_id` i `isActive`. Ovaj entitet je u vezi *Many-to-One* sa User preko `user_id`, *Many-to-One* sa Vinyl preko `vinyl_id`, *One-to-One* sa SaleAd preko `ad_id` i *One-to-One* sa ExchangeAd preko `ad_id`.

Ad		
<code>ad_id</code>	LONG	Jedinstveni identifikator oglasa
<code>vinyl_id</code>	STRING	Jedinstveni identifikator vinila
<code>sale_ad_id</code>	LONG	Jedinstveni identifikator oglasa za prodaju
<code>exchange_ad_id</code>	LONG	Jedinstveni identifikator oglasa za razmjenu
<code>isActive</code>	BOOLEAN	Aktivnost oglasa

SaleAd Ovaj entitet sadrži sve važne informacije za oglas koji je označen za prodaju vinila. Sadrži attribute: `ad_id` i `price`. Ovaj entitet je u vezi *One-to-One* sa Ad preko atributa `ad_id`.

SaleAd		
<code>ad_id</code>	LONG	Jedinstveni identifikator oglasa
<code>price</code>	NUMERIC	Cijena vinila u oglasu

ExchangeAd Ovaj entitet sadrži sve važne informacije za oglas koji je označen za razmjenu vinila. Sadrži attribute: `ad_id`, `new_Owner_id` i `exchanged_vinyl_id`. Ovaj entitet je u vezi *One-to-One* sa Ad preko atributa `ad_id`.

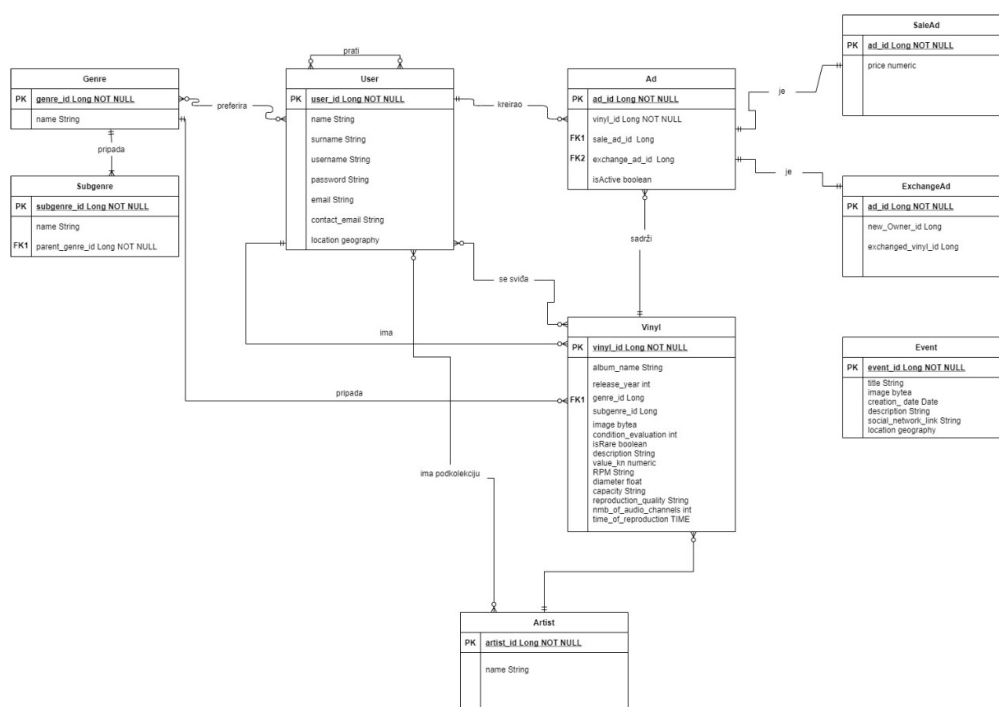
ExchangeAd		
<code>ad_id</code>	LONG	Jedinstveni identifikator oglasa
<code>new_Owner_id</code>	LONG	Jedinstveni identifikator novog vlasnika
<code>exchanged_vinyl_id</code>	LONG	Jedinstveni identifikator razmjenjenog vinila

Event Ovaj entitet sadrži sve važne informacije za događaj. Sadrži attribute:

event_id, title, image, creation_date, description, social_network_link i location. Ovaj entitet nije u vezi niti sa jednim drugim entitetom.

Event		
event_id	LONG	Jedinstveni identifikator događaja
title	STRING	Naslov događaja
image	BYTEA	Slika događaja
creation_date	DATE	Datum događaja
description	STRING	Kratki opis događaja
social_network_link	STRING	Link na događaj
location	GEOGRAPHY	Lokacija događaja

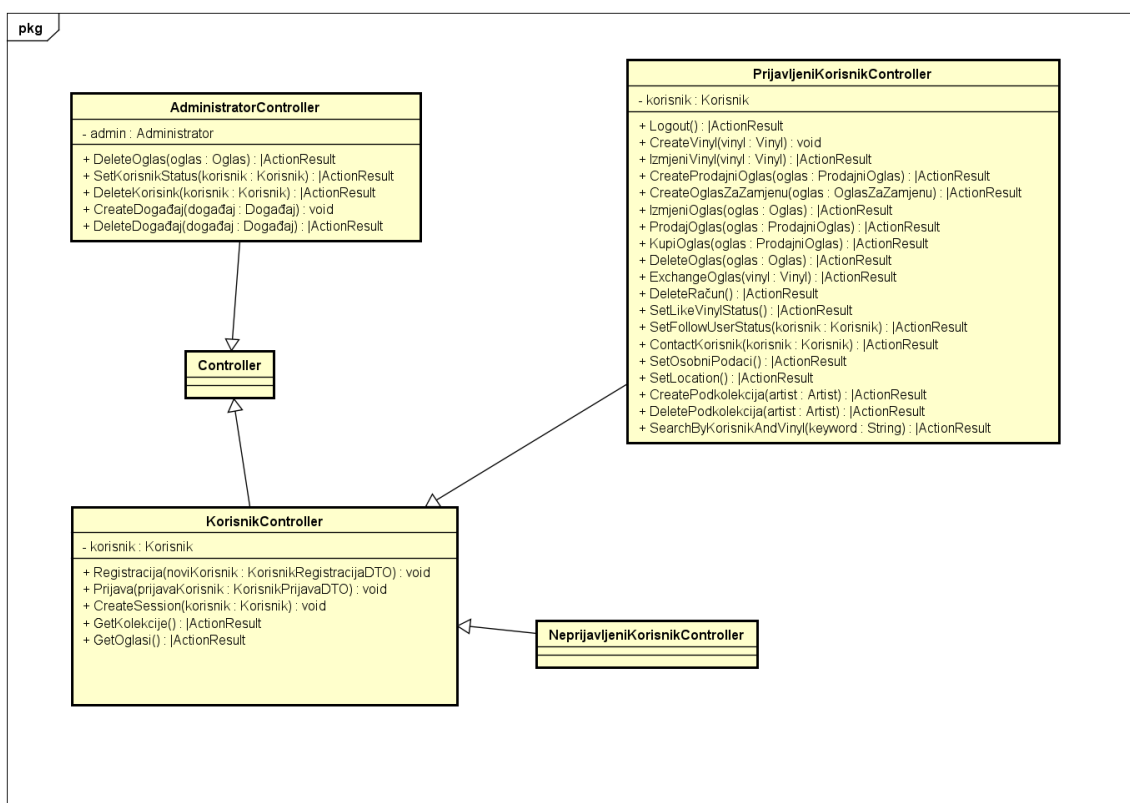
4.1.2 Dijagram baze podataka



Slika 4.2: Dijagram baze podataka

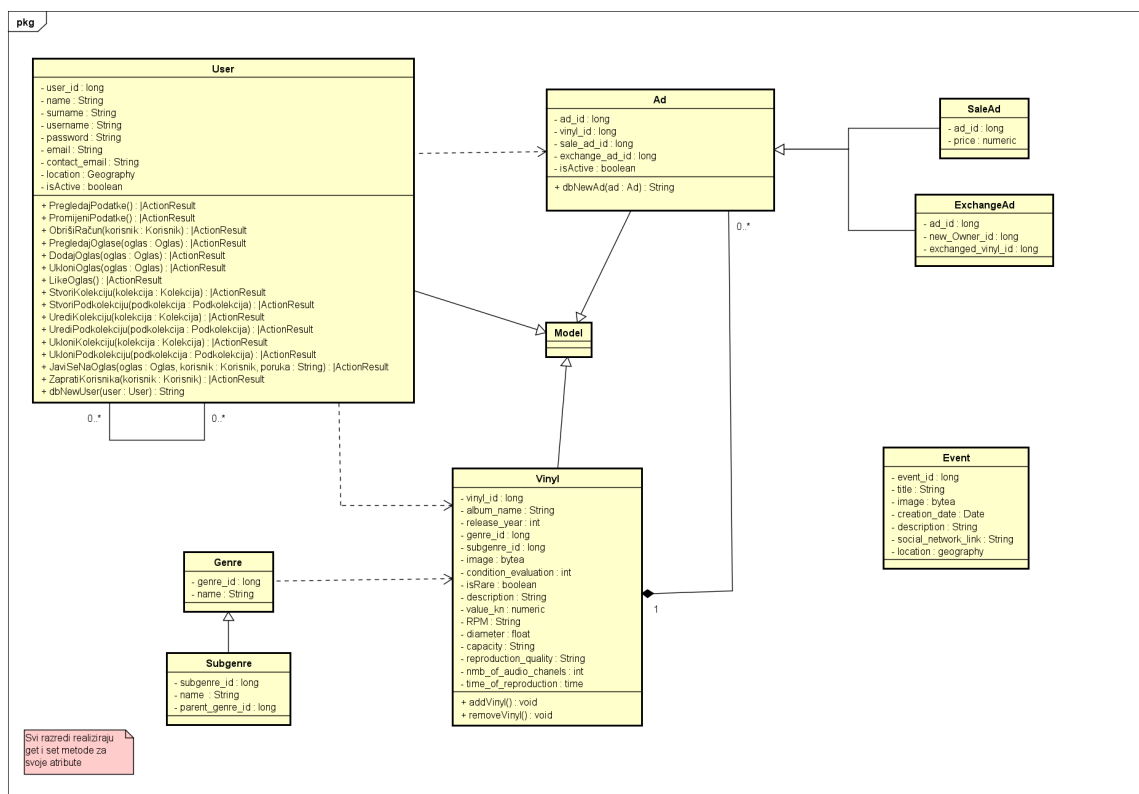
4.2 Dijagram razreda

Slike 4.3 i 4.4 prikazuju dijagrame razreda koji pripadaju *backend* dijelu MVC arhitekture. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Neki metode implementirane u tim razredima manipuliraju s DTO (*Data transfer object*) koji su dohvaćeni pomoću metoda implementiranih u Model razredima. Metode implementirane u Controller razredima vraćaju JSON datoteke s HTML status kodom. Razredi su podijeljeni logički po pravu pristupa metodama određenih aktora.



Slika 4.3: Dijagram razreda controller

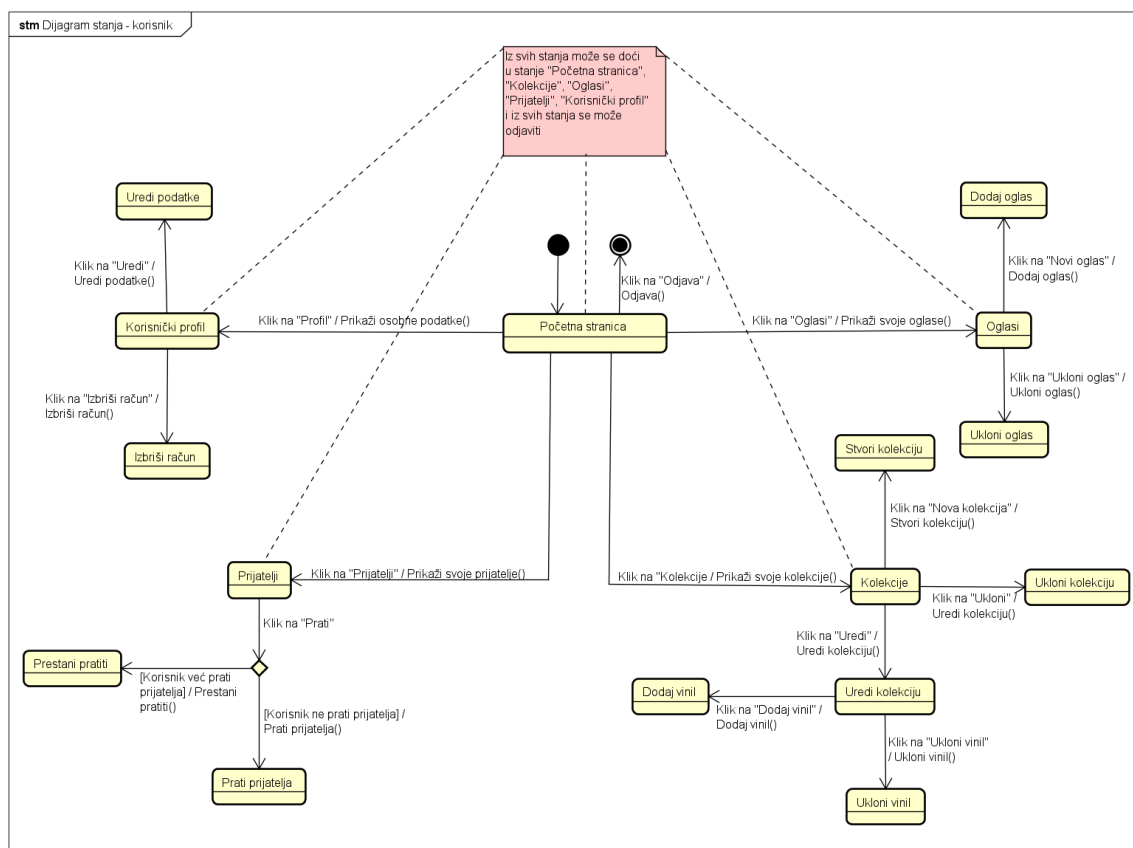
Na slici 4.4 prikazan je dijagram razreda modela koji služi za preslikavanje baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke. Razred User predstavlja prijavljenog korisnika aplikacije te ima sve navedene atribute koji prijavljeni korisnik mora imati kako bi mogao komunicirati s bazom podataka. Razred Ad predstavlja oglase koje korisnik može objaviti a razred Vinyl predstavlja vinile koje korisnik posjeduje i vinil koji se nalazi u oglasu.



Slika 4.4: Dijagram razreda model

4.3 Dijagram stanja

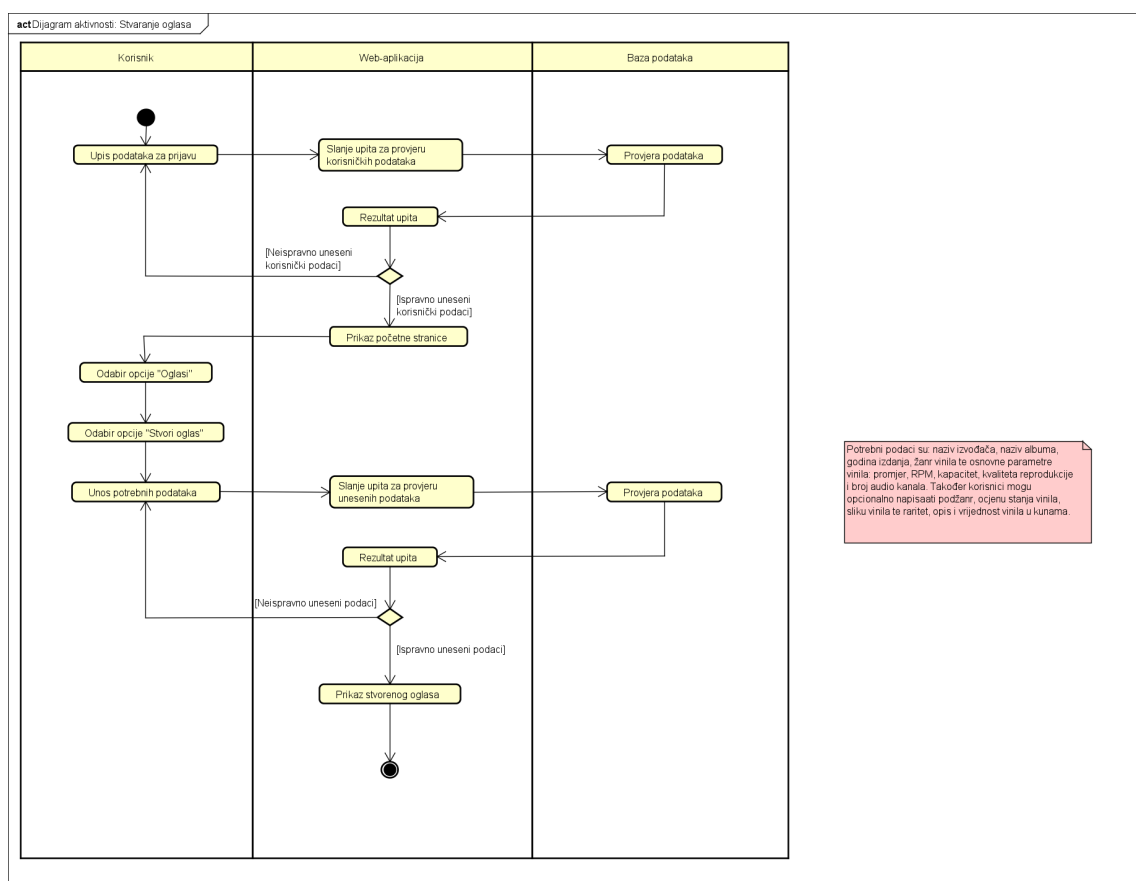
Dijagram stanja opisuje dinamičko ponašanje dijela sustava te prikazuje stanja objekta i prijelaze iz jednog stanja u drugo temeljene na događajima. Na 4.5 prikazan je dijagram stanja za registriranog korisnika. Nakon prijave, korisniku se prikazuje početna stranica na kojoj može pristupiti svom profilu, oglasima, kolekcijama te svojim prijateljima. Na svom korisničkom profilu može mijenjati podatke ili izbrisati račun u slučaju prestanka korištenja aplikacije. Korisnici imaju mogućnost stvaranja kolekcija vinila svojih preferenci te oglašavanja vinila za razmjenu ili prodaju. Prilikom razmjene ili prodaje vinila kada korisnici stupaju u kontakt, imaju mogućnost praćenja, tj. dodavanja kao prijatelja, korisnika.



Slika 4.5: Dijagram stanja

4.4 Dijagram aktivnosti

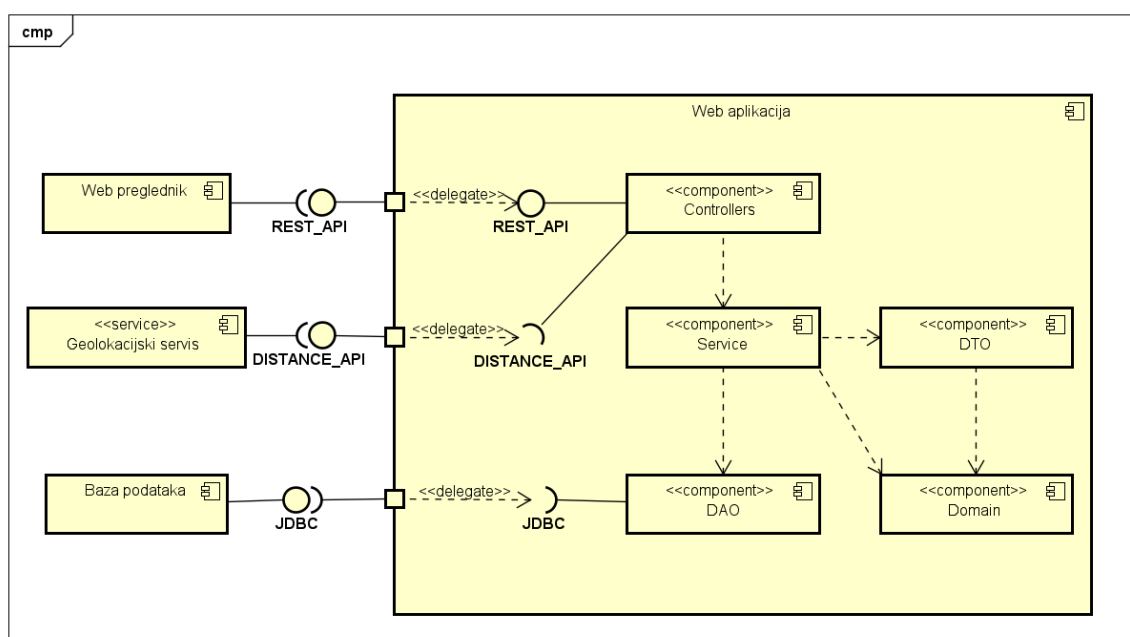
Dijagram aktivnosti primjenjuju se za modeliranje poslovnih procesa te upravljačkog i podaktovnog toka. Na 4.6 prikazuje se tijek stvaranja oglasa od početne do završne točke pokazujući različite smjerove aktivnosti koje se izvršavaju. Radnja završava kada su sve radnje, potrebne za stvaranje oglasa, izpravno izvedene.



Slika 4.6: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti na 4.7 prikazuje povezanost i međuovisnost komponenti, interne strukture i odnos prema vanjskoj okolini. Web aplikacija temeljena je na oblikovnom obrascu MVC (*Model-View-Controller*). REST API radi s podacima koji pripadaju *backend* dijelu aplikacije. SQL API omogućuje interakciju sa tablicama i podacima iz baze podataka pomoću SQL upita. Podaci koji se dohvaćaju iz baze podataka proslijeđuju se MVC strukturi u obliku DTO (*Data Transfer Object*).



Slika 4.7: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Tehnologije koje se bile korištene u svrhu komunikacije tima su bile WhatsApp¹ i Discord². Za izradu obrazaca uporabe i UML dijagrama korišten je alat Astah UML³, a kao sustav za upravljanje kodom projekta Git⁴. Kompletan kod i dokumentacija projekta dostupna je na udaljenom repozitoriju platforme Gitlab⁵.

Kao razvojno okruženje korišteno je IntelliJ IDEA⁶, integrirano razvojno okruženje razvijeno od tvrtke JetBrains. Napisano je u Javi u svrhu razvoja računalnog softvera. IDE pruža integraciju s alatima za izradu/pakiranje kao što su bower, gradle, i sl. Podržava sustave kao Git i SVN te baze podataka tipa Microsoft SQL Server, Oracle, PostgreSQL, i sl.

Frontend dio web aplikacije napisan je korištenjem Javascript biblioteke React⁷ i programskog jezika JavaScript⁸, dok je *backend* dio napisan je korištenjem radnog okvira Spring Boot⁹ i programskog jezika Java¹⁰. React (također poznat kao React.js ili ReactJS) open-source biblioteka za izgradnju korisničkih sučelja temeljenih na UI komponentama. React se bavi samo upravljanjem stanjem i prikazivanjem tog stanja DOM-u, s obzirom na to stvaranje aplikacija obično zahtijeva korištenje dodatnih biblioteka za interakciju s API-jem. Radni okvir Spring Boot olakšava stvaranje samostalnih aplikacija koje se temelje na Springu koje se mogu samo "pokrenuti".

Za upravljanje relacijskom bazom podataka korišten je sustav PostgreSQL¹¹, a za pokretanje i rad aplikacije korištena je platforma Heroku¹². Dokumentacija je

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://astah.net/>

⁴<https://git-scm.com/>

⁵<https://gitlab.com/>

⁶<https://www.jetbrains.com/idea/>

⁷<https://reactjs.org/>

⁸<https://www.javascript.com/>

⁹<https://spring.io/projects/spring-boot>

¹⁰<https://www.java.com/en/>

¹¹<https://www.postgresql.org/>

¹²<https://id.heroku.com/login>

napisana u Latex uredniku Texmaker¹³, UML dijagrami napravljeni su u programu AstahUML¹⁴ i ispitivanje programskog rješenja je odrađeno korištenjem metode JUnit u Javi za ispitivanje komponenti i programa Selenium IDE¹⁵ za ispitivanje sustava.

¹³<https://www.xmlmath.net/texmaker/>

¹⁴<https://astah.net/>

¹⁵<https://www.selenium.dev/selenium-ide/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Za prva tri ispitna slučaja korištena je konfiguracija:

```
@BeforeEach
public void setUp(){

    registerUserDTO = new RegisterUserDTO();
    registerUserDTO.setName("Marta");
    registerUserDTO.setSurname("Dubilić");
    registerUserDTO.setUsername("mdulibic");
    registerUserDTO.setPassword("123456789");
    registerUserDTO.setEmail("mdulibic@gmail.com");

    Genre genre1 = new Genre();
    genre1.setId(1L);
    genre1.setName("Rock");
    Genre genre2 = new Genre();
    genre2.setId(2L);
    genre2.setName("Pop");
```

Slika 5.1: 1. dio konfiguracije

```
userGenrePreference = new ArrayList<Genre>();
given(genreRepository.findById(1L)).willReturn(genre1);
given(genreRepository.findById(2L)).willReturn(genre2);
userGenrePreference.add(genreRepository.findById(1L));
userGenrePreference.add(genreRepository.findById(2L));

user1 = new User(registerUserDTO, userGenrePreference: null, location: null);

given(userRepository.findByUsername("mdulibic")).willReturn(Optional.of(user1));
given(userRepository.findById(1L)).willReturn(Optional.of(user1));
```

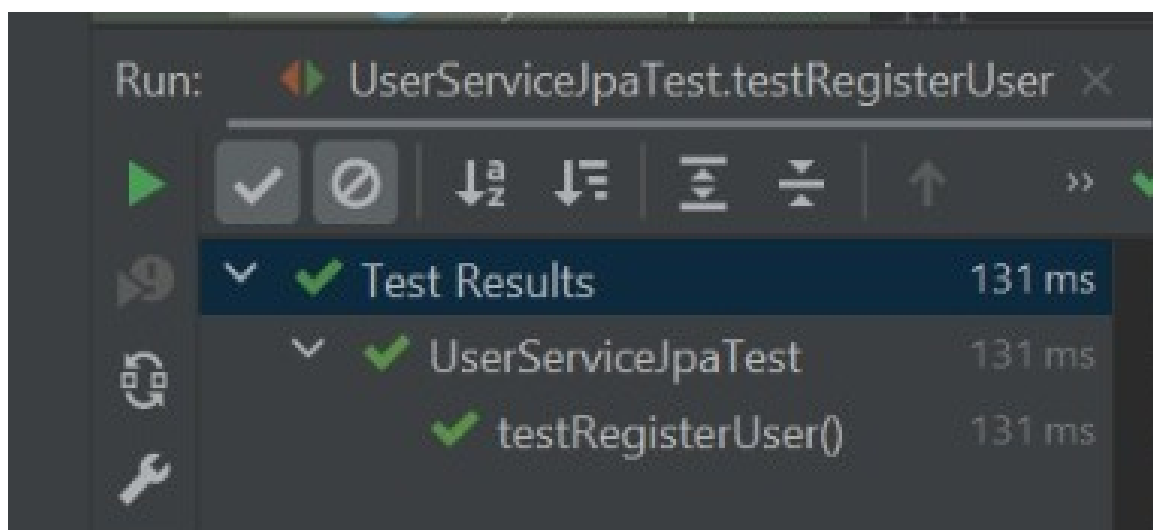
Slika 5.2: 2. dio konfiguracije

Ispitni slučaj 1: Provjera registracije korisnika

Ulaz: Ispravni korisnički podaci.

Očekivani rezultat: Uspješna registracija korisnika.

Rezultat: Očekivani rezultat uspješan.



Slika 5.3: Prikaz rezultata unit testa 1

```
@Test
public void testRegisterUser(){
    User newUser = userServiceJpa.registerUser(registerUserDTO, userGenrePreference: null, location: null);
    ArgumentCaptor<User> userArgumentCaptor = ArgumentCaptor.forClass(User.class);
    verify(userRepository).save(userArgumentCaptor.capture());
    User capturedUser = userArgumentCaptor.getValue();
    assertThat(actual: "mdulibic").isEqualTo(capturedUser.getUsername());
}
```

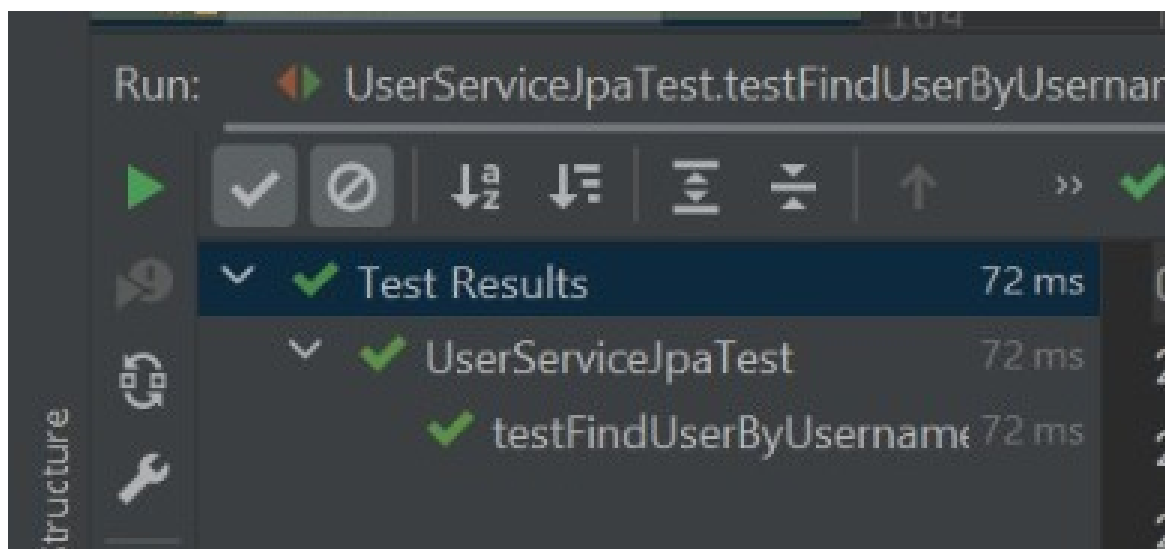
Slika 5.4: Prikaz koda unit testa 1

Ispitni slučaj 2: Pretraga korisnika prema username-u

Ulaz: Korisničko ime.

Očekivani rezultat: Prikazan korisnički profil pretraženog korisničkog imena.

Rezultat: Očekivani rezultat uspješan.



Slika 5.5: Prikaz rezultata unit testa 2



Slika 5.6: Prikaz koda unit testa 2

Ispitni slučaj 3: Prikaz svih korisnika

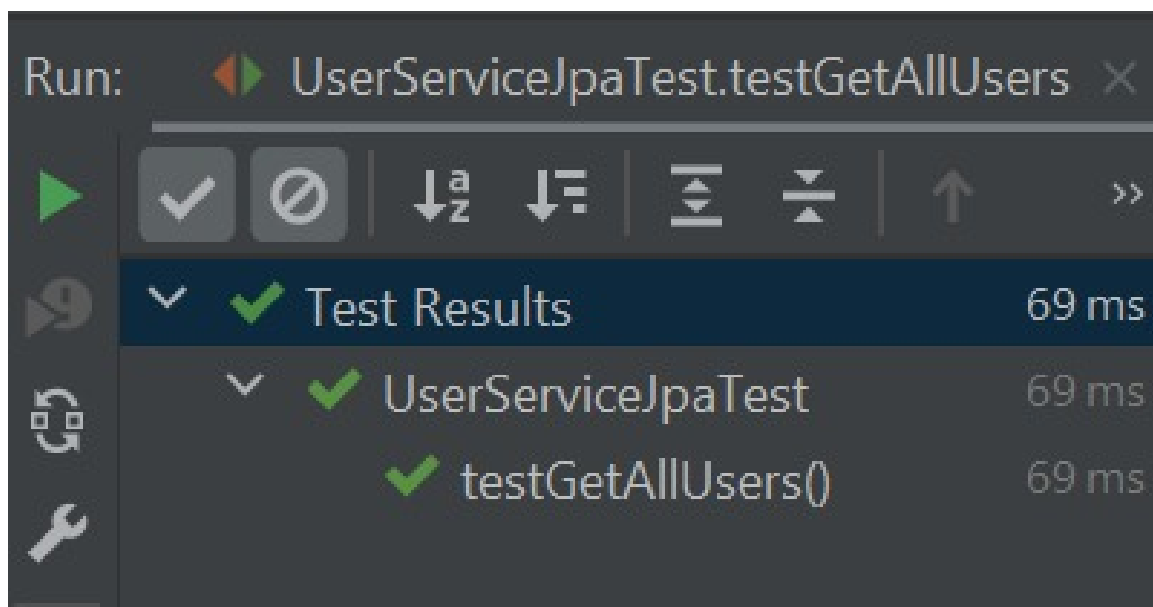
Ulaz: Nema ulaza.

Očekivani rezultat: Lista svih korisnika.

Rezultat: Očekivani rezultat uspješan.

```
@Test
public void testGetAllUsers() {
    userServiceJpa.listAll();
    verify(userRepository).findAll();
}
```

Slika 5.7: Prikaz rezultata unit testa 3



Slika 5.8: Prikaz koda unit testa 3

Za ostala tri ispitna slučaja korištena je konfiguracija:

```
@Before
public void setUp(){
    mockMvc = MockMvcBuilders.standaloneSetup(vinylController).build();

    vinyls = new ArrayList<>();

    Vinyl vinyl1 = new Vinyl();
    vinyl1.setId(1L);
    vinyl1.setAlbum("Album");
    vinyl1.setDescription("best album ever");
    vinyl1.setReproductionQuality("good");
    vinyl1.setNmbOfAudioChannels(3);

    Vinyl vinyl2 = new Vinyl();
    vinyl1.setId(2L);
    vinyl1.setAlbum("AlbumAlbum");
    vinyl1.setReproductionQuality("good");
    vinyl1.setNmbOfAudioChannels(2);
}
```

Slika 5.9: 1. dio druge konfiguracije

```
user = new User();
user.setId(1L);
user.setName("Ime");
user.setSurname("Prezime");
user.setUsername("username");
user.setPassword(passwordEncoder.encode(charSequence: "12345678"));

vinyls.add(vinyl1);
vinyls.add(vinyl2);
```

Slika 5.10: 2. dio druge konfiguracije

Ispitni slučaj 4: Dodavanje novog vinila

Ulaz: Potrebni podaci za stvaranje vinila.

Očekivani rezultat: Uspješno dodan vinil.

Rezultat: Očekivani rezultat uspješan.


```
@Test
public void testAddNewVinyl() throws Exception {
    when(vinylServiceJpa.addVinyl(any(Vinyl.class), user)).thenReturn(vinyls.get(0));
    //when(vinylServiceJpa.addVinyl(vinyls.get(0), user)).thenReturn(vinyls.get(0));

    mockMvc.perform(post( urlTemplate: "/vinyls")
        .contentType(MediaType.APPLICATION_JSON)
        .content(objectMapper.writeValueAsString(vinyls.get(0))))
        .andExpect(status().isCreated());
}
```

Slika 5.11: Prikaz koda unit testa 4

Ispitni slučaj 5: Brisanje oglasa

Ulaz: Ime vinila.

Očekivani rezultat: Vinil maknut iz kolekcije.

Rezultat: Očekivani rezultat uspješan.

```
@Test
public void testDeleteVinyl() throws Exception {
    doThrow(new EntityNotFoundException()).when(vinylServiceJpa).deleteVinyl( vinylId: 1L);

    mockMvc.perform(delete( urlTemplate: "/vinyls/{id}", ...uriVars: 1L))
        .andExpect(status().isNotFound());
}
```

Slika 5.12: Prikaz koda unit testa 5

Ispitni slučaj 6: Dohvati vinil

Ulaz: Zahtjev.

Očekivani rezultat: Odgovor. (200)

Rezultat: Očekivani rezultat uspješan.

```
@Test
public void testGetVinyls() throws Exception {
    mockMvc.perform(get( urlTemplate: "/vinyls")
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk());
}
```

Slika 5.13: Prikaz koda unit testa 6

5.2.2 Ispitivanje sustava

Za ispitivanje sustava koristio se Selenium IDE.

Ispitni slučaj 1: Prijava u sustav - uspješna prijava

Ulaz:

1. Otvaranje naslovne stranice web aplikacije
2. Pomak kursora iznad polja za upis korisničkog imena
3. Pritisak na polje za upis korisničkog imena
4. Upis korisničkog imena
5. Pomak kursora iznad polja za upis lozinke
6. Pritisak na polje za upis lozinke
7. Upis lozinke
8. Pomak kursora iznad gumba za prijavu
9. Pritisak gumba "Prijava"

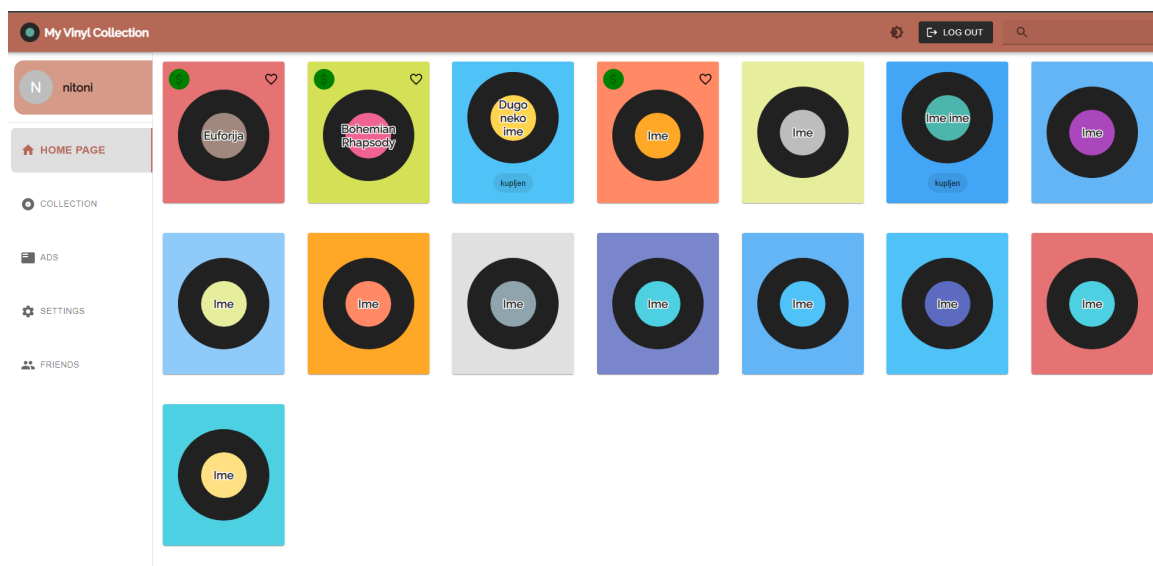
Očekivani rezultat:

1. Prikazuje se početna stranica web aplikacije

Rezultat: Očekivani rezultat je ostvaren.

	Command	Target	Value
1	open	/	
2	set window size	1936x1056	
3	mouse over	css=.MuiButton-containedPrimary	
4	click	css=.MuiButton-containedPrimary	
5	click	name=username	
6	type	name=username	nitoni
7	click	name=password	
8	mouse over	css=.MuiButton-root	
9	type	name=password	tonitoni1
10	click	css=.MuiButton-root	

Slika 5.14: Potrebni koraci za uspješnu prijavu



Slika 5.15: Početna stranica nakon uspješne prijave

Ispitni slučaj 2: Prijava u sustav - neuspješna prijava

Ulaz:

1. Otvaranje naslovne stranice web aplikacije
2. Pomak kursora iznad polja za upis korisničkog imena
3. Pritisak na polje za upis korisničkog imena
4. Upis pogrešnog korisničkog imena
5. Pomak kursora iznad polja za upis lozinke
6. Pritisak na polje za upis lozinke
7. Upis lozinke
8. Pomak kursora iznad gumba za prijavu
9. Pritisak gumba "Prijava"

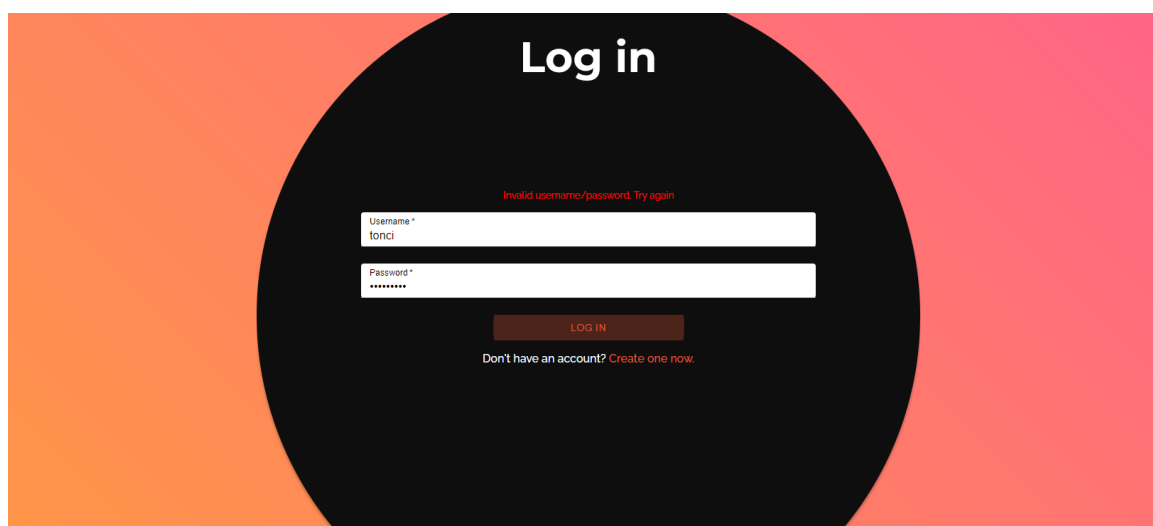
Očekivani rezultat:

1. Prikazuje se naslovna stranica web aplikacije
2. Sustav izbacuje poruku o pogrešno upisanim korisničkim imenom ili lozinkom

Rezultat: Očekivani rezultat je ostvaren.

	Command	Target	Value
1	open	/	
2	set window size	1936x1056	
3	mouse over	css=.MuiButton-containedPrimary	
4	click	css=.MuiButton-containedPrimary	
5	click	name=username	
6	type	name=username	tonci
7	click	name=password	
8	mouse over	css=.MuiButton-root	
9	type	name=password	tonitoni1
10	click	css=.MuiButton-root	

Slika 5.16: Potrebni koraci za neuspješnu prijavu



Slika 5.17: Naslovna stranica nakon neuspješne prijave

Ispitni slučaj 3: Dodavanje oglasa

Ulaz:

1. Pritisak gumba "ADS"
2. Pritisak gumba "Add new ad"
3. Odabir između opcije namjene oglasa: "SALE" i "EXCHANGE"
4. Pritisak na padajući izbornik
5. Pritisak na vinil koji će biti u oglasu

6. Pritisak na praznu kućicu u koju se unosi cijena
7. Upis cijene vinila (u dolarima)
8. Pritisak gumba "OK"

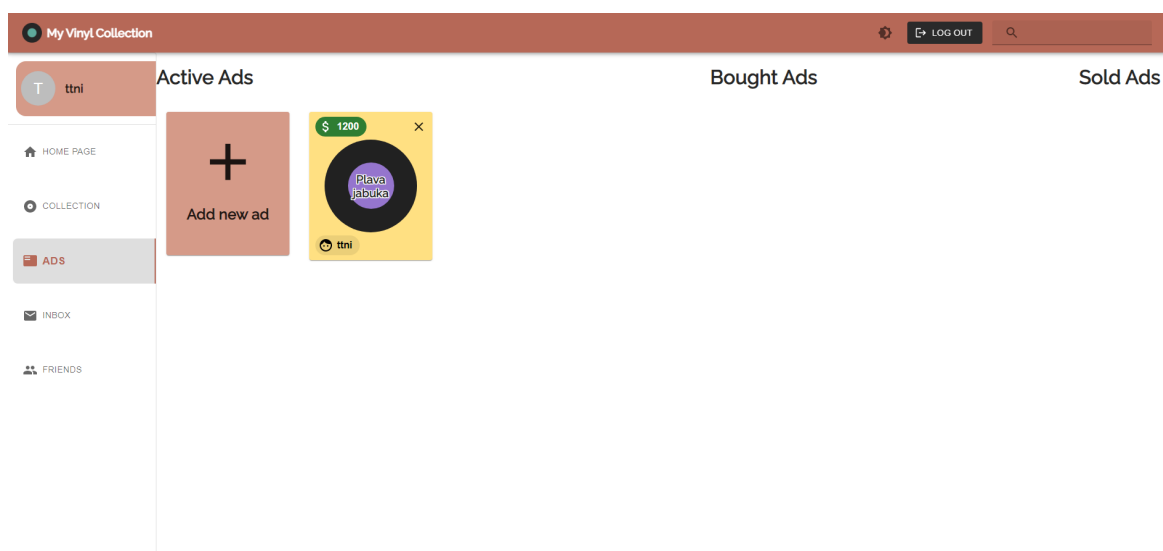
Očekivani rezultat:

1. Prikazuje se stranica "ADS" sa dodanim oglasom

Rezultat: Očekivani rezultat je ostvaren.

	Command	Target	Value
1	open	https://my-vinyl-collection-frontend.herokuapp.com/dashboard/ads	
2	set window size	1898x1008	
3	click	css=.MuiPaper-root > .MuiSvgIcon-root	
4	click	css=.MuiToggleButton-root:nth-child(1)	
5	mouse down	css=.MuiSelect-select	
6	mouse up	css=.MuiBackdrop-invisible	
7	click	css=body	
8	click	css=div > p	
9	click	css=.css-1t8l2tu-MuiInputBase-input-MuiOutlinedInput-input	
10	mouse over	css=.MuiButton-root:nth-child(4)	
11	type	css=.css-1t8l2tu-MuiInputBase-input-MuiOutlinedInput-input	1200
12	click	css=.MuiButton-root:nth-child(4)	

Slika 5.18: Potrebni koraci za dodavanje oglasa



Slika 5.19: Uspješno dodan oglas

Ispitni slučaj 4: Dodavanje vinila u kolekciju**Ulaz:**

1. Pritisak gumba "COLLECTION"
2. Pritisak gumba "Add new vinyl"
3. Pritisak na praznu kućicu za upis imena albuma
4. Upis imena albuma
5. Pritisak na padajući izbornik za odabir izvođača
6. Odabir izvođača
7. Pritisak na praznu kućicu za upis godine izdanja
8. Upis godine izdanja
9. Pritisak na padajući izbornik za odabir žanra
10. Odabir žanra
11. Pritisak na praznu kućicu za upis evaulacije stanja
12. Upis evaulacije stanja
13. Pritisak na praznu kućicu za upis RPM-a
14. Upis RPM-a
15. Pritisak na praznu kućicu za upis kapaciteta
16. Upis kapaciteta
17. Pritisak na praznu kućicu za upis kvalitete reprodukcije
18. Upis kvalitete reprodukcije
19. Pritisak na praznu kućicu za upis broja audio kanala
20. Upis broja audio kanala
21. Pritisak na praznu kućicu za upis vremena reprodukcije
22. Upis vremena reprodukcije
23. Odabir između opcije: "Yes" i "No" za raritet vinila
24. Pritisak gumba "ADD"

Očekivani rezultat:

1. Prikazuje se stranica "COLLECTION" sad dodanim vinilima

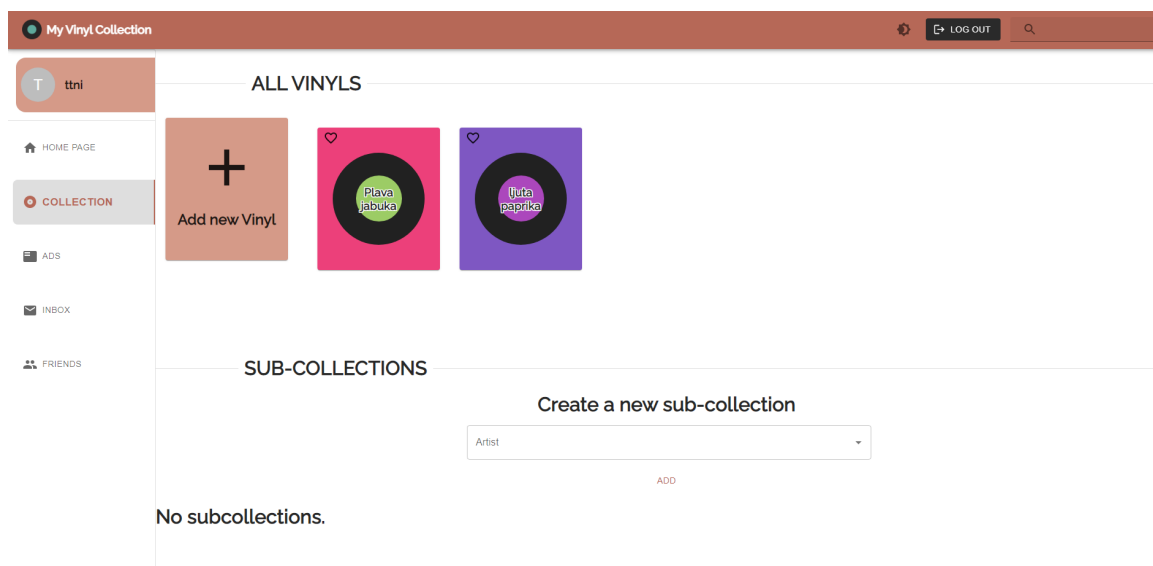
Rezultat: Očekivani rezultat je ostvaren.

	Command	Target	Value
1	<i>open</i>	https://my-vinyl-collection-frontend.herokuapp.com/dashboard/collecti on	
2	<i>set window size</i>	1936x1056	
3	<i>click</i>	css=.MuiPaper-root > .MuiSvgIcon-root	
4	<i>click</i>	name=album	
5	<i>type</i>	name=album	Ijuta paprika
6	<i>click</i>	id=combo-box-demo	
7	<i>click</i>	id=combo-box-demo-option-1	
8	<i>click</i>	name=releaseYear	
9	<i>type</i>	name=releaseYear	2001
10	<i>click</i>	css=.MuiAutocomplete-root:nth-child(5) .MuiSvgIcon-root	
11	<i>click</i>	id=combo-box-demo-option-3	
12	<i>click</i>	name=conditionEvaluation	
13	<i>type</i>	name=conditionEvaluation	5
14	<i>click</i>	name=rpm	
15	<i>run script</i>	window.scrollTo(0,0)	

Slika 5.20: Potrebni koraci dodavanje vinila u kolekciju - 1.dio

16	<i>type</i>	name=rpm	3600
17	<i>click</i>	name=capacity	
18	<i>type</i>	name=capacity	32000
19	<i>click</i>	name=reproductionQuality	
20	<i>type</i>	name=reproductionQuality	5
21	<i>click</i>	name=nmbOfAudioChannels	
22	<i>type</i>	name=nmbOfAudioChannels	4
23	<i>click</i>	name=timeOfReproduction	
24	<i>type</i>	name=timeOfReproduction	00:24:15
25	<i>mouse over</i>	css=.MuiButton-text	
26	<i>click</i>	css=.MuiButton-text	

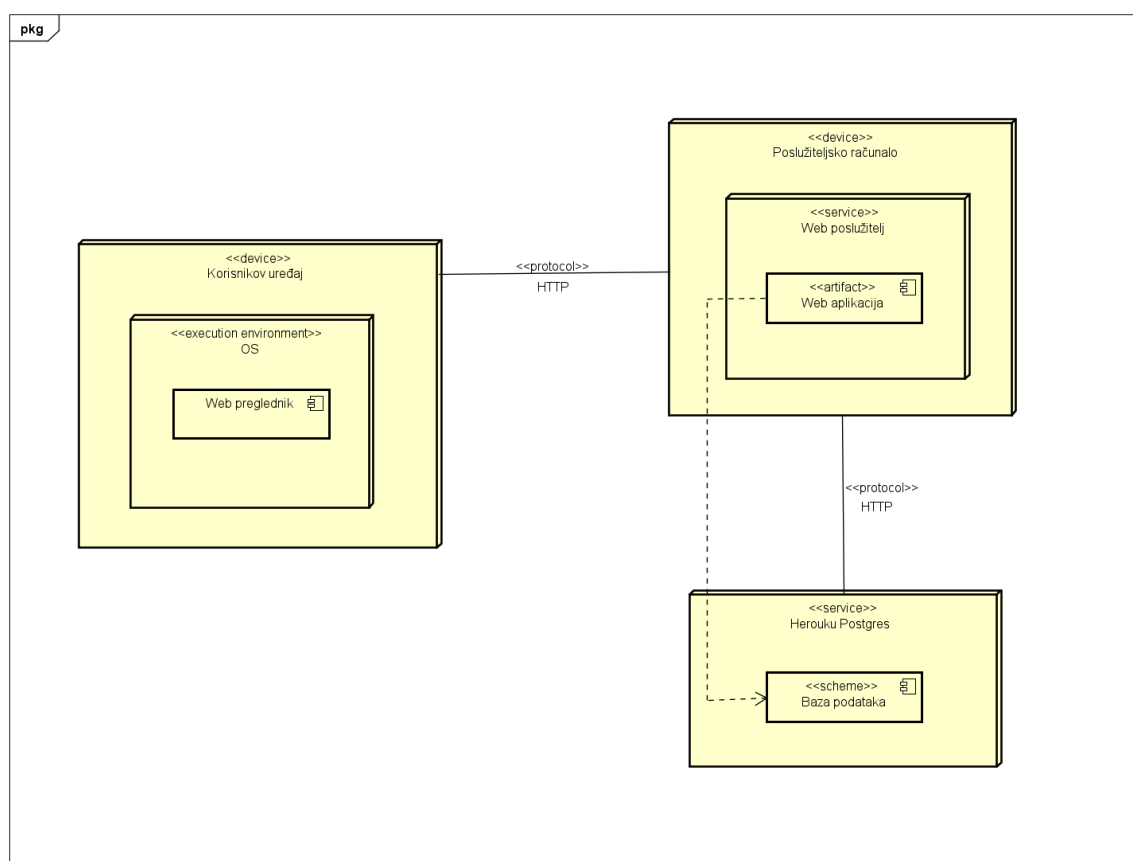
Slika 5.21: Potrebni koraci dodavanje vinila u kolekciju - 2.dio



Slika 5.22: Uspješno dodani vinili u kolekciju

5.3 Dijagram razmještaja

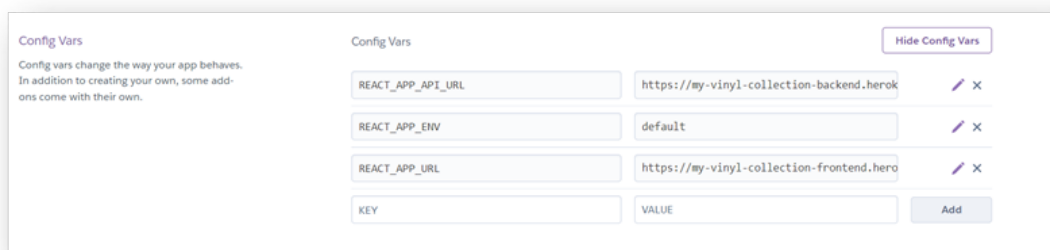
Dijagram razmještaja je strukturni dijagram koji opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Sustav je baziran na tro-razinskoj arhitekturi pri kojoj klijent - poslužitelj i poslužitelj - baza podataka komuniciraju preko HTTP veze. Klijenti upotrebljavaju web preglednik kako bi pristupili web aplikaciji, dok web aplikacija pristupa bazi podataka kako bi pristupala podacima u svrhu provjere ili prikaza korisniku.



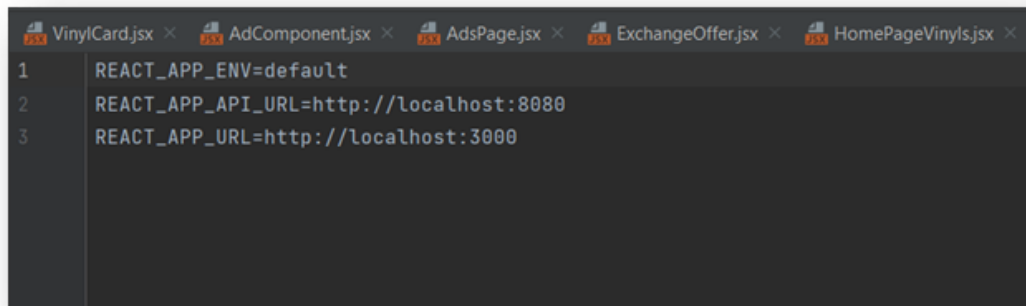
Slika 5.23: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Za deployanje aplikacije korištena je cloud platforma Heroku te se pomoću .yml datoteke na svaki push na granu master automatski pokrene pipeline koji postavlja novi kod na heroku server. Kako se naš repozitorij sastoji od backend i frontend projekta, na Heroku platformi su stvorene dvije zasebne aplikacije. Frontend i backend su povezani preko varijabli okruženja koje moramo definirati i na heroku serveru i u našem projektu kako bi se aplikacija mogla izvoditi i lokalno i na serveru. Primjer varijabli okruženja za frontend:

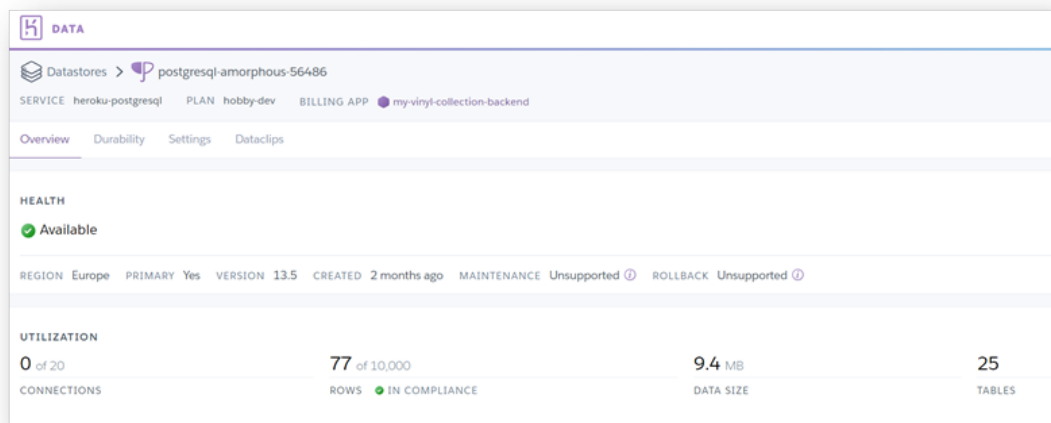


Slika 5.24: Varijable okruženja u postavkama heroku frontend aplikacije

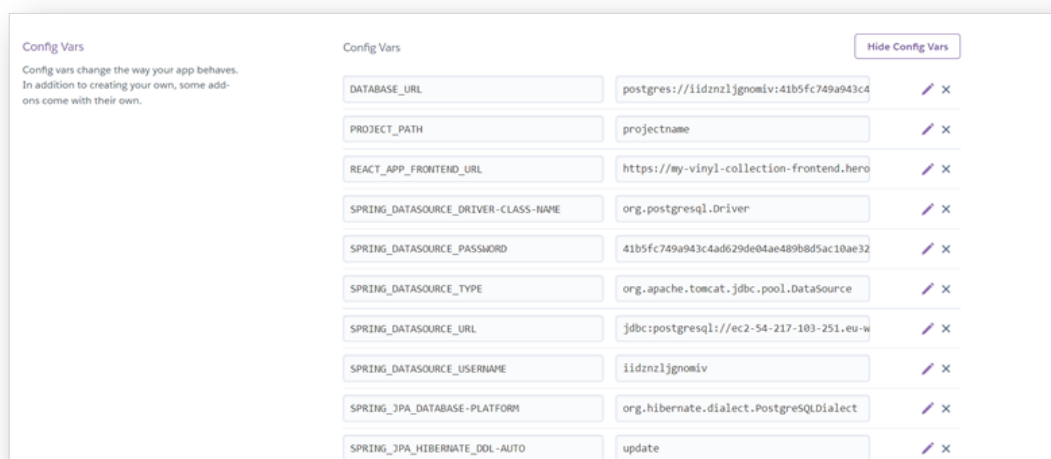


Slika 5.25: Datoteka varijabli okruženja u frontend projektu

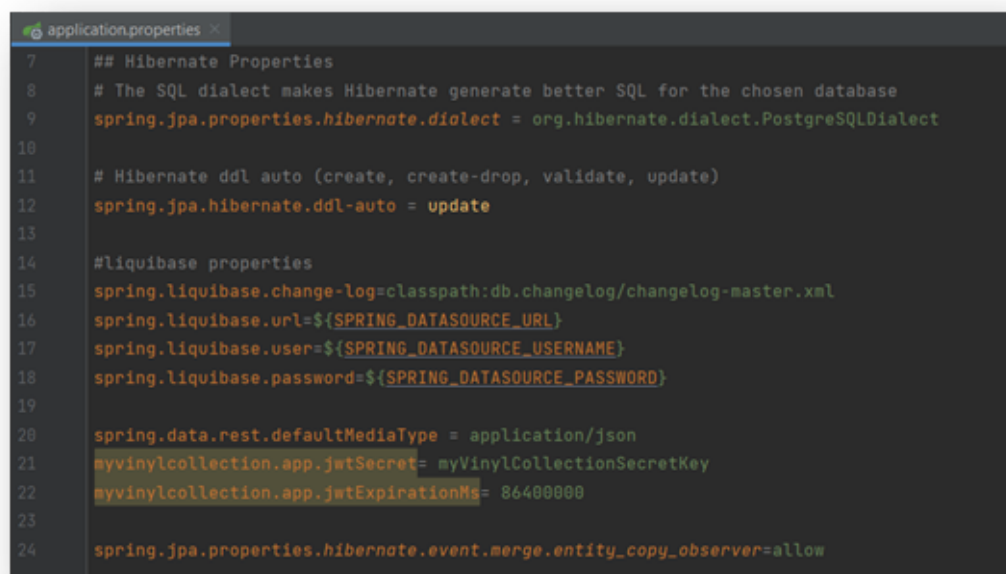
U backend heroku aplikaciji također je pod Resources dodana Heroku Postgres baza te kako bismo se povezali na nju dodane su varijable okruženja u konfiguracijske varijable heroku backend aplikacije i pod *application.properties* u našem backend projektu u repozitoriju:



Slika 5.26: Prikaz Heroku Postgres baze



Slika 5.27: Varijable okruženja u postavkama heroku backend aplikacije

A screenshot of a code editor showing the content of a file named 'application.properties'. The file contains configuration properties for a Spring Boot application, including Hibernate dialect, ddl-auto, Liquibase properties, default media type, JWT secret and expiration, and Hibernate event merge settings. The properties are listed line by line, with line numbers 7 through 24 visible on the left margin.

```
7  ## Hibernate Properties
8  # The SQL dialect makes Hibernate generate better SQL for the chosen database
9  spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect
10
11 # Hibernate ddl auto (create, create-drop, validate, update)
12 spring.jpa.hibernate.ddl-auto = update
13
14 #liquibase properties
15 spring.liquibase.change-log=classpath:db.changelog/changelog-master.xml
16 spring.liquibase.url=${SPRING_DATASOURCE_URL}
17 spring.liquibase.user=${SPRING_DATASOURCE_USERNAME}
18 spring.liquibase.password=${SPRING_DATASOURCE_PASSWORD}
19
20 spring.data.rest.defaultMediaType = application/json
21 myvinylcollection.app.jwtSecret= myVinylCollectionSecretKey
22 myvinylcollection.app.jwtExpirationMs= 86400000
23
24 spring.jpa.properties.hibernate.event.merge.entity_copy_observer=allow
```

Slika 5.28: Datoteka *application.properties* sa varijablama okruženja u backend projektu

Upute za pokretanje na vlastitom računalu

Korisnik prvotno mora klonirati projekt u odabranom razvojnom okruženju sa git repozitorija. Zatim se potrebno pozicionirati putem konzole (cmd-windows/bash-linux) unutar mape sa backend projektom te izvršiti naredbu **mvn spring-boot:run** (u našem projektu koristimo Maven kao alat za automatizaciju gradnje) ili otvoriti projekt u razvojnom okruženju na putanji gdje se nalazi taj projekt te stisnuti gumb Run kako bi se pokrenuo pozadinski servis. Nakon što smo pokrenuli pozadinski servis, pozicioniramo se unutar mape sa frontend projektom opet ili putem konzole ili IDE-a. Ako određeni paketi nisu instalirani, potrebno ih je instalirati izvršavanjem naredbe **npm install „naziv-paketa“** (ili samo sa izvršiti naredbu **npm install** koja onda instalira sve pakete koji nisu u tom trenutku instalirani). Nakon toga, pokrećemo React aplikaciju naredbom **npm start** nakon čega će se naša aplikacija pokrenuti na adresi <http://localhost:3000>.

6. Zaključak i budući rad

Projektni zadatak naše grupe bio je izrada i razvoj web aplikacije za stvaranje kolekcije vinila uz mogućnost razmjene, kupovine i prodaje vinila. Zadatak je ostvaren nakon 16 tjedana grupnog rada tima. Provedba praćenja napretka razvoja aplikacije tima bila je kroz dvije faze.

Prva faza projekta krenula je sporije zbog upoznavanja tima i njihovih iskustva i znanja, organizacije i raspodjele poslova te dogovoru za sastanke koji će pratiti napredak projekta. Uspješnom raspodjelom tima na podtimove obilježio je početak rada na projektu. Cilj prve faze projekta bila je izrada *alfa* inačice projekta, tj. web aplikacija s osnovim radnim funkcionalnostima kao što su *login* i *registracija*. Velika pomoć u izradi i razvoju *frontend-a* i *backend-a* bili su obrasci i dijagrami (obraci uporabe, sekvencijski dijagrami, dijagram razreda i model baze podataka). Vizualni dizajn web aplikacije bio je brzo riješen kako bi tim mogao nastaviti dalje implementirati funkcije.

Druga faza projekta, iako kraća od prve, ali zahtijevnija, krenula je intenzivnije po pitanju posla koji je trebao svaki član tima odraditi. Prilikom izrade implementacijskih rješenja članovi tima morali su samostalno naučiti odabrane alate i programske jezike kako bi zadovoljili dogovoreni cilj. Također je bilo potrebno napraviti korektne UML dijagrame i ispravnu dokumentaciju kako bi budući korisnici, investitori mogli lakše razumijeti funkcionalnost sustava. Kvalitetno izrađen prototip aplikacije i složnost tima, pomogao nam je u drugoj fazi projekta kako bi se mogle izbjeći vremensko složne pogreške i nepotrebne funkcionalnosti aplikacije. Slijedeći korak projektnog zadatka bio bi izrada mobline aplikacije koja bi mogla otvoriti nove funkcionalnosti aplikacije (npr. prepoznavanje ploče s kamerom koja bi popunila osnovne informacije ploče, ...).

Najvažniji dio projekta bila je organizacija i komunikacija tima. Prateći napredak i ostatak posla svakog podtima bila je nužna kako bi se projektni zadatak mogao napraviti u roku. Svaki član tima sudjelovanjem u ovom projektu shvatio je veliku važnost u dobroj vremenskoj organizaciji i komunikaciji i koordinaciji s ostatkom tima. Zadovoljni smo s količinom posla koje smo odradili u zadanom vremenskom periodu i postoji još mjesta za poboljšanje, ali to je sve posljedica ne-

iskustva, koje svi u početku imamo, na takvim i sličnim budućim projektima.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Wikipedija, https://en.wikipedia.org/wiki/Main_Page
8. C# Corner, <https://www.c-sharpcorner.com/>
9. MDN Web Docs, <https://developer.mozilla.org/en-US/>
10. Javatpoint, <https://www.javatpoint.com/>

Indeks slika i dijagrama

2.1	Početna stranica Njuškala	7
2.2	Početna stranica Oglasnika	8
3.1	Dijagram obrasca uporabe, funkcionalnost neprijavljenog korisnika	20
3.2	Dijagram obrasca uporabe, funkcionalnost prijavljenog korisnika . .	21
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	22
3.4	Sekvencijski dijagram za UC5	23
3.5	Sekvencijski dijagram za UC7	24
3.6	Sekvencijski dijagram za UC13	25
3.7	Sekvencijski dijagram za UC17	27
4.1	Arhitektura sustava	29
4.2	Dijagram baze podataka	35
4.3	Dijagram razreda controller	36
4.4	Dijagram razreda model	37
4.5	Dijagram stanja	38
4.6	Dijagram aktivnosti	39
4.7	Dijagram komponenti	40
5.1	1. dio konfiguracije	43
5.2	2. dio konfiguracije	43
5.3	Prikaz rezultata unit testa 1	44
5.4	Prikaz koda unit testa 1	44
5.5	Prikaz rezultata unit testa 2	45
5.6	Prikaz koda unit testa 2	45
5.7	Prikaz rezultata unit testa 3	46
5.8	Prikaz koda unit testa 3	46
5.9	1. dio druge konfiguracije	47
5.10	2. dio druge konfiguracije	47
5.11	Prikaz koda unit testa 4	48
5.12	Prikaz koda unit testa 5	48

5.13	Prikaz koda unit testa 6	48
5.14	Potrebni koraci za uspješnu prijavu	49
5.15	Početna stranica nakon uspješne prijave	50
5.16	Potrebni koraci za neuspješnu prijavu	51
5.17	Naslovna stranica nakon neuspješne prijave	51
5.18	Potrebni koraci za dodavanje oglasa	52
5.19	Uspješno dodan oglas	52
5.20	Potrebni koraci dodavanje vinila u kolekciju - 1.dio	54
5.21	Potrebni koraci dodavanje vinila u kolekciju - 2.dio	54
5.22	Uspješno dodani vinili u kolekciju	55
5.23	Dijagram razmještaja	56
5.24	Varijable okruženja u postavkama heroku frontend aplikacije	57
5.25	Datoteka varijabli okruženja u frontend projektu	57
5.26	Prikaz Heroku Postgres baze	58
5.27	Varijable okruženja u postavkama heroku backend aplikacije	58
5.28	Datoteka <i>application.properties</i> sa varijablama okruženja u backend projektu	59
6.1	Dijagram promjena na grani master	70
6.2	Dijagram promjena na grani devdoc	71

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 7. listopada 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - upoznavanje s timom
 - podijela uloga u timu

2. sastanak

- Datum: 14. listopada 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - rasprava o temi projekta
 - nejasnoće oko teksta zadatka projekta

3. sastanak

- Datum: 21. listopada 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - razjašnjavanje nejasnoća s asistentima
 - predlaganje nekih ideja i alternativnih rješenja za pojedine dijelove zadataka

4. sastanak

- Datum: 26. listopada 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:

- upoznavanje sa Gitom
- dogovor oko razvojnog okruženja

5. sastanak

- Datum: 4. studenoga 2021.
- Prisustvovali: T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - razrada napravljene baze podataka
 - određivanje funkcionalnosti prve inačice projekta

6. sastanak

- Datum: 7. studenoga 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - tjedni nalazak tko je što napravio te kako koji dio projekta napreduje
 - što treba biti gotovo do slijedećeg nalaska

7. sastanak

- Datum: 13. studenoga 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov
- Teme sastanka:
 - backend update
 - frontend update

8. sastanak

- Datum: 15. studenoga 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov
- Teme sastanka:
 - testiranje povezanosti backend-a i frontend-a

9. sastanak

- Datum: 16. studenoga 2021.
- Prisustvovali: J.Srzić, P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - testiranje generičkih funkcionalnosti

10. sastanak

- Datum: 16. studenoga 2021.
- Prisustvovali: P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga

- Teme sastanka:
 - demonstracija generičkih funkcionalnosti asistentima
 - ispravak nekih dijelova dokumentacije

11. sastanak

- Datum: 19. studenoga 2021.
- Prisustvovali: L.Aleksić, M.Dulibić, T.Drezga
- Teme sastanka:
 - završno crtanje i ispravljanje dijagrama razreda

12. sastanak

- Datum: 8. prosinca 2021.
- Prisustvovali: P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - testiranje aplikacije

13. sastanak

- Datum: 22. prosinca 2021.
- Prisustvovali: P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - demonstracija alfa inačice
 - dogovoriranje roka i raspodijela zadataka u timu

14. sastanak

- Datum: 10. siječnja 2022.
- Prisustvovali: P.Pavlić, T.Krišto, L.Aleksić, M.Dulibić, P.Cukrov, T.Drezga
- Teme sastanka:
 - preostali poslovi za dovršit
 - dogovoriranje roka deployanja i završnog testiranja aplikacije

Tablica aktivnosti

	Josip Srzić	Pino Pavlić	Tea Krišto	Lucija Aleksić	Marta Dulibić	Petra Cukrov	Toni Drezga
Upravljanje projektom	15						
Opis projektnog zadatka							4
Funkcionalni zahtjevi							3
Opis pojedinih obrazaca							4
Dijagram obrazaca							5
Sekvencijski dijagrami							6
Opis ostalih zahtjeva							1
Arhitektura i dizajn sustava			1		1		2
Baza podataka	2	5	7		8		4
Dijagram razreda				2	3		4
Dijagram stanja							2
Dijagram aktivnosti							2
Dijagram komponenti							5
Korištene tehnologije i alati							2
Ispitivanje programskog rješenja			5				4
Dijagram razmještaja							3
Upute za puštanje u pogon					3		
Dnevnik sastajanja							1
Zaključak i budući rad							2
Popis literature							1

Nastavljeno na idućoj stranici

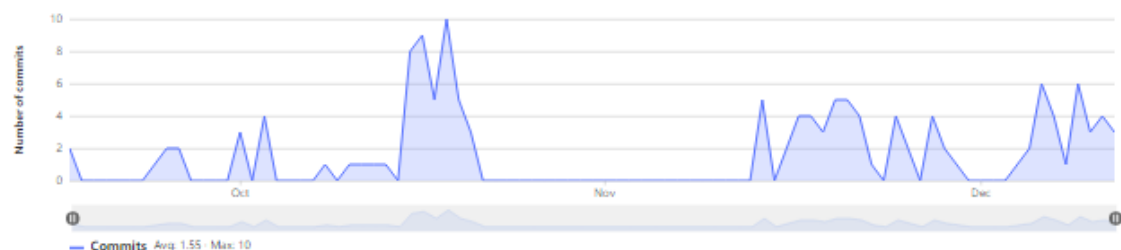
Nastavljeno od prethodne stranice

	Josip Srzić	Pino Pavlić	Tea Krišto	Lucija Aleksić	Marta Dulibić	Petra Cukrov	Toni Drezga
Izrada početne stranice	5						
Izrada baze podataka		3			5		
Spajanje s bazom podataka		1	5		2		
Izgled web aplikacije				15		3	
Front end	75			80		65	
Back end		65	70		75		

Dijagrami pregleda promjena

Commits to master

Excluding merge commits. Limited to 6,000 commits.



Marta

71 commits (marta.dulibic@fer.hr)



Josip Srzić

25 commits (josip.srzic@fer.hr)



Lucija Aleksić

21 commits (lucija.aleksic@fer.hr)



Tea Krišto

11 commits (tea.kristo@fer.hr)



Pino Pavlic

3 commits (pp52327@fer.hr)



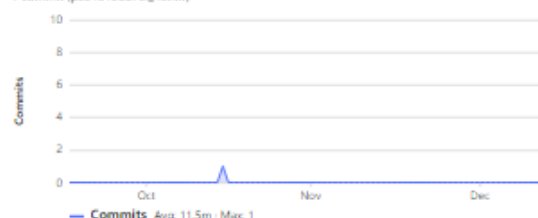
Marta Dulibić

3 commits (md52549@fer.hr)



Petra Cukrov

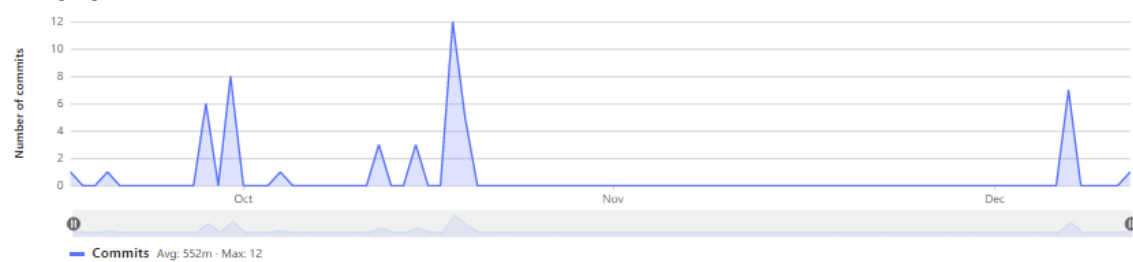
1 commit (pc013108876@fer.hr)



Slika 6.1: Dijagram promjena na grani master

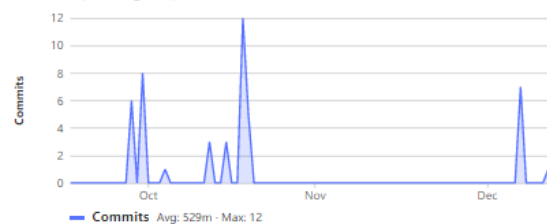
Commits to devdoc

Excluding merge commits. Limited to 6,000 commits.



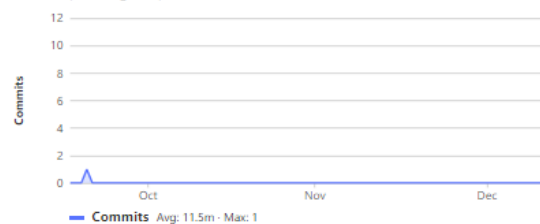
Toni Drezga

46 commits (td52821@fer.hr)



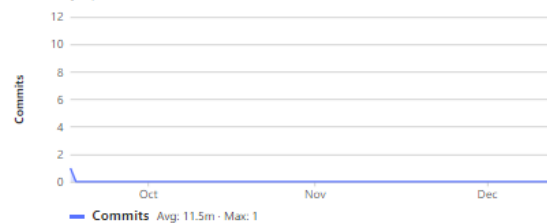
Tea Krišto

1 commit (tk52421@fer.hr)



Josip Srzić

1 commit (josip.srzic@fer.hr)



Slika 6.2: Dijagram promjena na grani devdoc