

Programsko inženjerstvo ak.god 2024./2025.

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Ovaj projekt je rezultat timskog rada u sklopu projektnog zadatka kolegija **Programsko inženjerstvo** na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu.

VinylSwap

Tim: <G16.4>

Ime tima: GramoFer

Članovi tima: Amar Omeragić, Kristijan Kozomara, Josip Rončević, Lovro Kasunić, Sami Perković, Jurica Galić, Tomislav Peraković

Nastavnik: Vlado Sruk

1. Opis projektnog zadatka

Cilj projekta

Cilj projekta VinylSwap je stvoriti funkcionalnu, intuitivnu i sigurnu web stranicu koja će olakšati razmjenu gramofonskih ploča među ljubiteljima glazbe. Na jednom mjestu korisnik će moći objaviti ploče koje nudi za zamjenu, pregledavati i pronalaziti željene ploče i realizirati zamjene.

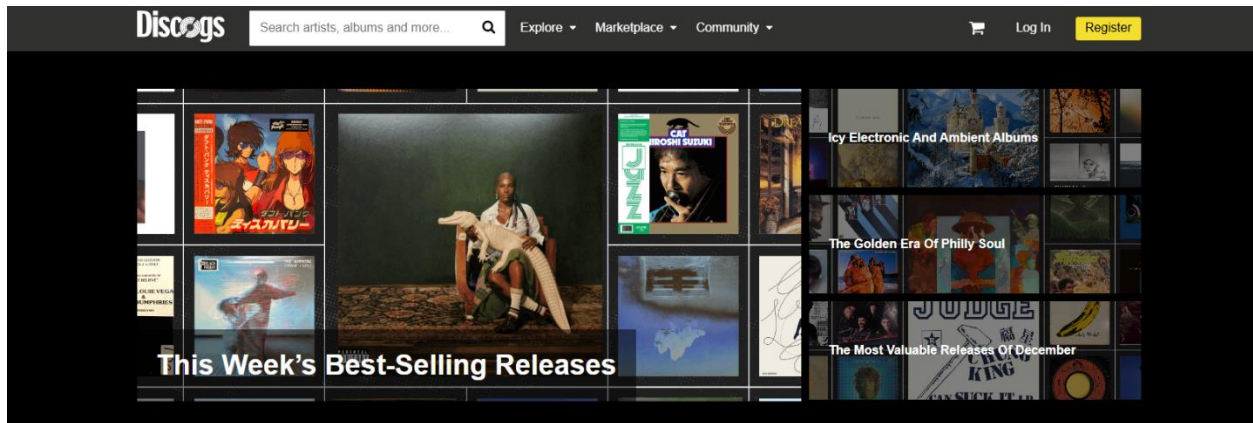
Motivacija

S obzirom na sve veću popularnost kolekcionarstva gramofonskih ploča i nedostatak postojećih platformi specifično posvećenih razmjeni, ljubitelji gramofonskih ploča suočavaju se s izazovima. Izazovi mogu uključivati ograničen broj izdanja, prestanak izdavanja, visoku potražnju i općenito manjak pouzdane mreže za povezivanje kolekcionara, što s vremenom stvara sve veću potrebu za pronalaskom rješenja.

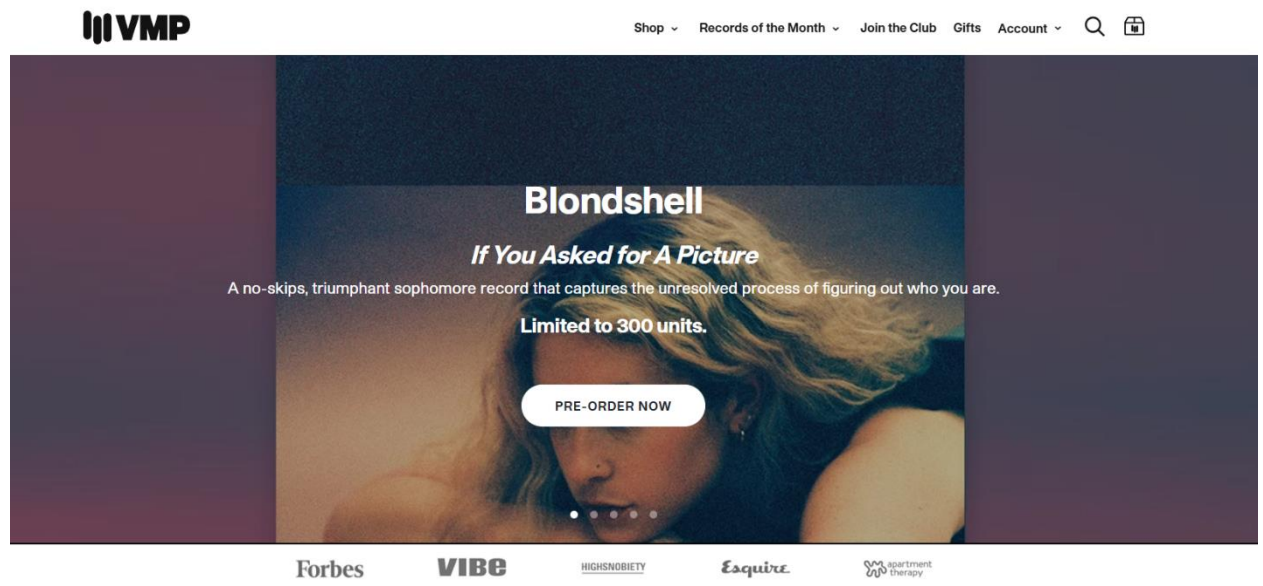
Slična rješenja

U svrhu prevladavanja navedenih izazova, neke web-stranice omogućile su online kupnju gramofonskih ploča, ali uglavnom fokusirajući se na prodaju, a ne na izravnu razmjenu između korisnika. Primjeri uključuju:

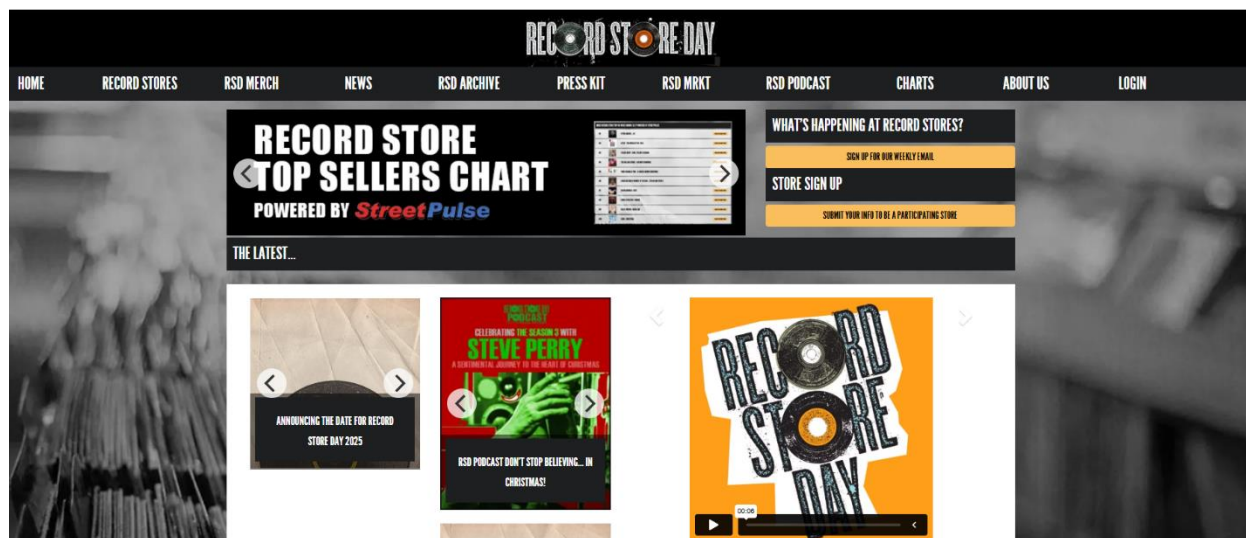
- **Discogs**, koji povezuje kolekcionare za kupnju i prodaju



- **Vinyl Me, Please**, koji nudi pretplatu za mjesečne ploče



- **Record Store Day**, koji organizira godišnja događanja s mogućnošću kupnje ograničenih izdanja



Iako ove platforme nude značajnu uslugu za ljubitelje ploča, njihov glavni fokus ostaje na trgovini i komercijalnoj kupnji. Nasuprot tome, ovim projektom omogućila bi se izravna razmjena ploča među korisnicima, čime se olakšava suradnja bez posrednika i dodatnih troškova.

Potencijalna korist ovog projekta

Korist ovog projekta je u stvaranju platforme koja povezuje ljubitelje gramofonskih ploča omogućujući im jednostavnu razmjenu i priliku za otkrivanje novih izdanja čime se širi glazbena kultura.

Skup potencijalno zainteresiranih korisnika

Skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje obuhvaća glazbene entuzijaste i kolekcionare gramofonskih ploča svih razina iskustva, od onih koji traže priliku za otkrivanje novih izdanja i povezivanje s glazbenom zajednicom, do strastvenih sakupljača koji žele proširiti svoju kolekciju.

Opseg projektnog zadatka

Korisnički zahtjevi

- **Ograničen pristup neregistriranih korisnika:** Neregistrirani korisnici mogu pregledavati popis svih objavljenih ploča uz ograničene funkcionalnosti.
- **Registracija i prijava:** Korisnici se moraju registrirati važećom adresom e-pošte, a autorizacija unutar aplikacije vrši se putem OAuth 2.0 protokola koristeći tokene. Uspješnom registracijom korisnici dobivaju pristup ostalim u nastavku opisanim funkcionalnostima aplikacije.
- **Objava ploča:** Registrirani korisnici mogu objavljivati ploče navodeći oznaku izdanja, naziv albuma, izvođača, godinu izdavanja, glazbeni žanr, stanje ploče i

omota prema Goldmine standardu, fotografiju ploče i omota, lokaciju ploče putem vanjske usluge za geolociranje te po potrebi dodatan opis.

- **Pregled i uređivanje ploča:** Korisnici mogu vidjeti svoje objavljene ploče na pregledu “Moje ploče” i dodatno ih uređivati.
- **Pretraga i filtriranje ploča:** Korisnici mogu koristiti filtriranje po kategorijama i traku za pretraživanje kako bi lakše pronašli ploču koju traže.
- **Ponuda za zamjenu:** Pronalaskom željene ploče korisnik može odabrati opciju “Ponudi zamjenu” i odabrati jednu ili više svojih ploča koje je spreman zamijeniti.
- **Obavijesti o ponudi:** Korisnicima će se nova ponuda za zamjenu uz obavijest na stranici prikazati u pregledu “Ponude” te će zaprimiti e-mail s poveznicom na ponuđenu zamjenu.
- **Prihvatanje i uređivanje zamjene:** Korisnici mogu prihvatiti zamjenu ili urediti zamjenu s novim uvjetima odabirom jedne ili više ploča koje posjeduje ponuditelj.
- **Popis želja:** Korisnici mogu stvoriti popis želja ploča koje traže za koje će ih sustav obavijestiti kada postanu dostupne prikazom u sustavu i putem e-maila.
- **Arhiviranje zamjena:** Izvršene zamjene se arhiviraju i korisnicima se prikazuju u pregledu “Moje zamjene”.
- **Upravljanje korisnicima i oglasima:** Sistemski administratori mogu upravljati korisnicima i uređivati oglase, uključujući brisanje oglasa i deaktiviranje korisničkih računa u slučaju kršenja pravila.
- **Prilagodljivost mobilnom uređaju** - Aplikacija je izvedena kao web aplikacija prilagođena svim uređajima bez gubitka funkcionalnosti ili dizajna.

Mogućnost prilagodbe rješenja

- **Personalizacija** - Korisnici bi mogli dodatno prilagoditi funkcionalnosti stranice svojim potrebama, poput biranja koje obavijesti žele primati, filtriranja ploča po tome koje su najnovije, najbliže ili najpopularnije, biranja jezika sustava, biranja načina komunikacije nakon dogovorene zamjene i slično.
- **Recenziranje drugih korisnika** - Korisnici bi mogli ostavljati recenzije i ocjene drugim korisnicima nakon zamjena, povećavajući povjerenje i transparentnost.
- **Korisnička podrška** - Aplikacija bi nudila mogućnost korisnicima za kontaktiranje podrške putem različitih kanala, poput e-maila, chata ili FAQ sekcije, za rješavanje tehničkih problema, pitanja o funkcionalnostima ili specifičnim potrebama.

Moguće nadogradnje projektnog zadatka

- **Integracija s dostavnim službama** - Povezivanje s dostavnim službama omogućilo bi korisnicima jednostavniju i pouzdaniju razmjenu ploča. Automatiziranjem procesa slanja smanjila bi se potreba za samostalnim organiziranjem poštanskih usluga, dok bi korisnicima i dalje bila pružena fleksibilnost u izboru opcija dostave. Aplikacija bi olakšala koordinaciju slanja i primanja ploča u skladu s korisničkim željama, uz zadržavanje mogućnosti da sami snose troškove dostave.
- **Integracija s društvenim mrežama** - Povezivanjem društvenim mrežama korisnicima bi bila omogućena jednostavnija promocija njihovih ponuda i popisa

želja. Ova funkcionalnost omogućila bi dijeljenje objava o dostupnim pločama za razmjenu, čime bi se povećala vidljivost i dosegnula šira publika.

2. Analiza zahtjeva

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvatanja
F-001	Pregled objavljenih gramofonskih ploča	Visok	Sustav	Korisnik može uspješno pregledati sve objavljene gramofonske ploče.
F-002	Registracija u sustav	Visok	Zahtjev dionika	Korisnik se uspješno može prijaviti u sustav s odgovarajućom e-mail adresom.
F-003	Objava gramofonskih ploča s navedenom oznakom izdanja, nazivom albuma, izvođačem, godinom izdanja, glazbenim žanrom, stanjem ploče, stanjem omota prema Goldmine standardu, fotografijom ploče, lokacijom i dodatnim opisom	Visok	Dokument zahtjeva	Korisnik može objaviti oglas o gramofonskoj ploči i ona će se pojaviti na listi objavljenih gramofonskih ploča.
F-004	Pregled „Moje ploče“	Visok	Sustav	Korisnik će nakon svakog novog oglasa moći pronaći iste oglase u pregledu „Moje ploče“
F-005	Filtriranje po kategorijama	Visok	Sustav	Korisnik unosom filtera vidi samo ploče koje odgovaraju filter.
F-006	Pretraživanje preko trake za pretraživanje (po izvođaču ili albumu)	Visok	Sustav	Korisnik unosom izvođača ili albuma vidi samo ploče koje odgovaraju navedenom.
F-007	Zamjena ploča te dolazak tog zahtjeva u pregledu	Visok	Zahtjev korisnika	Korisnik nakon objave svoje ploče može tražiti

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
	"Ponude"			zamjenu za svoju ploču. Također, netko drugi može ponuditi zamjenu za korisnikovu ploču. Sve mora biti vidljivo u pregledu „Ponude“.
F-008	Stvaranje popisa želja	Visok	Povratna informacija korisnika	Korisnik će dobiti e-mail kada se gramofonska ploča na njegovom popisu želja pojavi u sustavu.
F-009	Upravljanje korisnicima	Visok	Sustav (administrator)	Nakon zabrane od strane administratora, korisnik više neće moći obavljati radnje u sustavu.
F-010	Uređivanje oglasa	Visok	Sustav (administrator)	Nakon uklanjanja oglasa od strane administratora, oglas će nestati iz sustava.

Ostali zahtjevi

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-001	Web stranica mora biti responzivna, što znači da se korisnici mogu registrirati i koristiti ju na računalu i mobilnom uređaju.	Visok
NF-002	Sustav mora biti jednostavan i intuitivan za korištenje tako da se korisnici mogu koristiti sučeljem bez proučavanja ikakvih opširnijih uputa.	Visok
NF-003	Korisnikovo neispravno korištenje aplikacije ne smije poremetiti njen rad ili funkcionalnost.	Visok
NF-004	Sustav treba biti izrađen korištenjem objektno-orijentirane paradigme u obliku web-aplikacije.	Visok
NF-005	Sustavu se pristupa iz javne mreže pomoću protokola HTTPS.	Visok
NF-006	Korisničko sučelje i sustav treba podržavati hrvatske diakritičke znakove za unos i prikaz tekstualnog sadržaja.	Visok
NF-007	Rad na sustavu ne smije narušavati funkcionalnost sustava.	Visok

ID	zahtjeva	Opis	Prioritet
NF-008		Sustav mora podržavati rad više korisnika u stvarnom vremenu.	Visok
NF-009		Svi privatni podaci u sustavu moraju biti zaštićeni.	Visok

Dionici

1. Korisnik
 - (a) Neregistrirani korisnik
 - (b) Registrirani korisnik
 - (c) Administrator
2. Vlasnik
3. Razvojni tim
4. Naručitelji

Aktori i njihovi funkcionalni zahtjevi:

A-1 Aktor 1 (neregistrirani korisnik) može: F-001, F-002

Autorizacija preko OAuth 2.0 protokola koristeći tokene

A-2 Aktor 2 (registrirani korisnik) može: F-001, F-002, F-003, F-004, F-005, F-006, F-007, F-008

Dolazak ponude na e-mail adresu s poveznicom na tu ponudu, Arhiviranje svih zamjena u pregledu "Moje zamjene", Sustavski odziv na popis želja kada se ona pojavi u sustavu

A-3 Aktor 3 (administrator) može: F-001, F-002, F-003, F-004, F-005, F-006, F-007, F-008, F-009, F-010

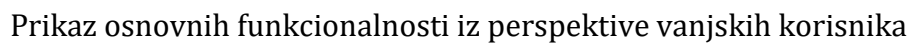
Brisanje objavljenih oglasa, Deaktivacija korisničkog računa, Dolazak ponude na e-mail adresu s poveznicom na tu ponudu, Arhiviranje svih zamjena u pregledu "Moje zamjene", Sustavski odziv na popis želja kada se ona pojavi u sustavu

A-4 Aktor 4 (baza podataka) može: Pohranjivanje podataka o korisnicima i njihovim ovlastima, pohranjivanje podataka o pločama

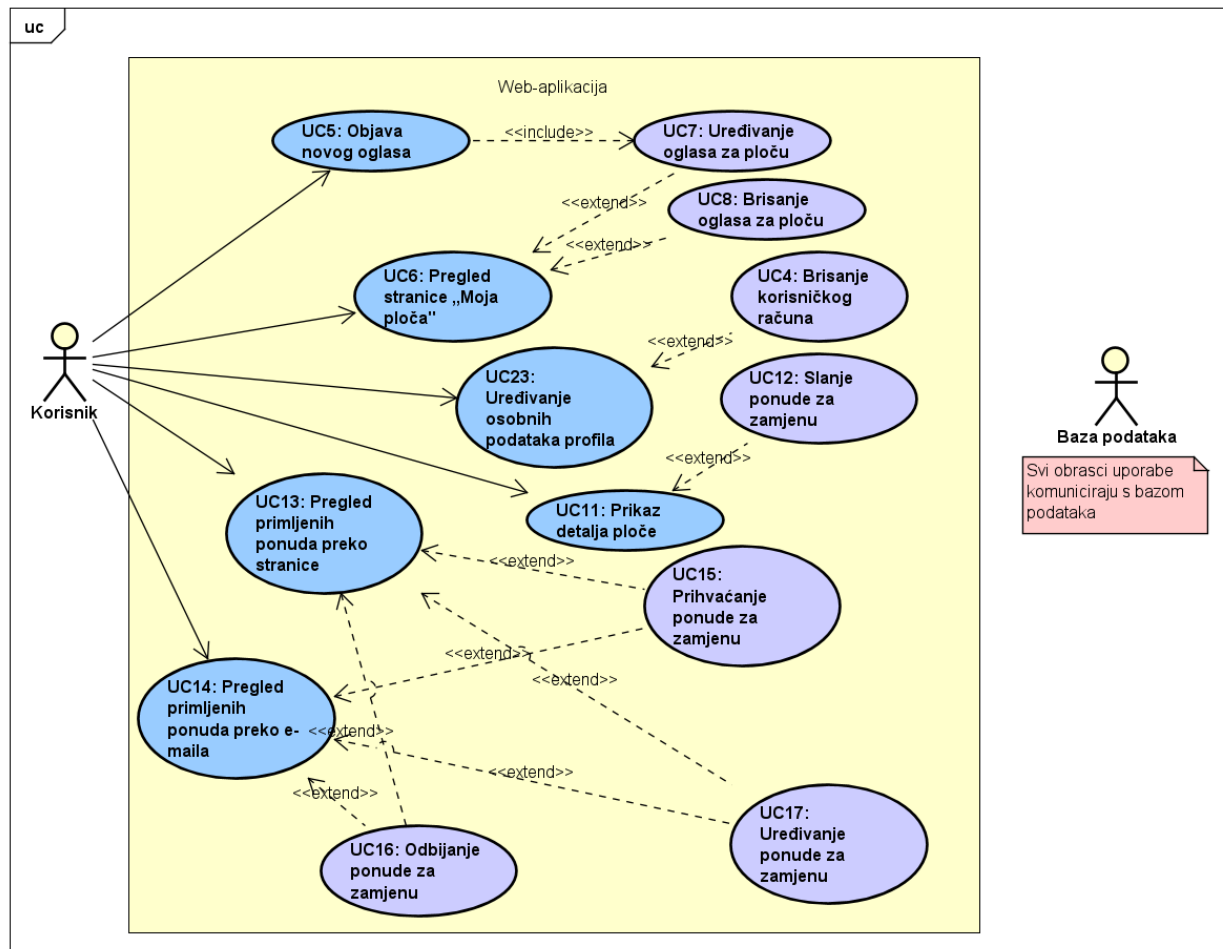
Detalji o svakoj ploči, Poslane ponude za ploče, Dostupnost ploča

3. Specifikacija zahtjeva sustava

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

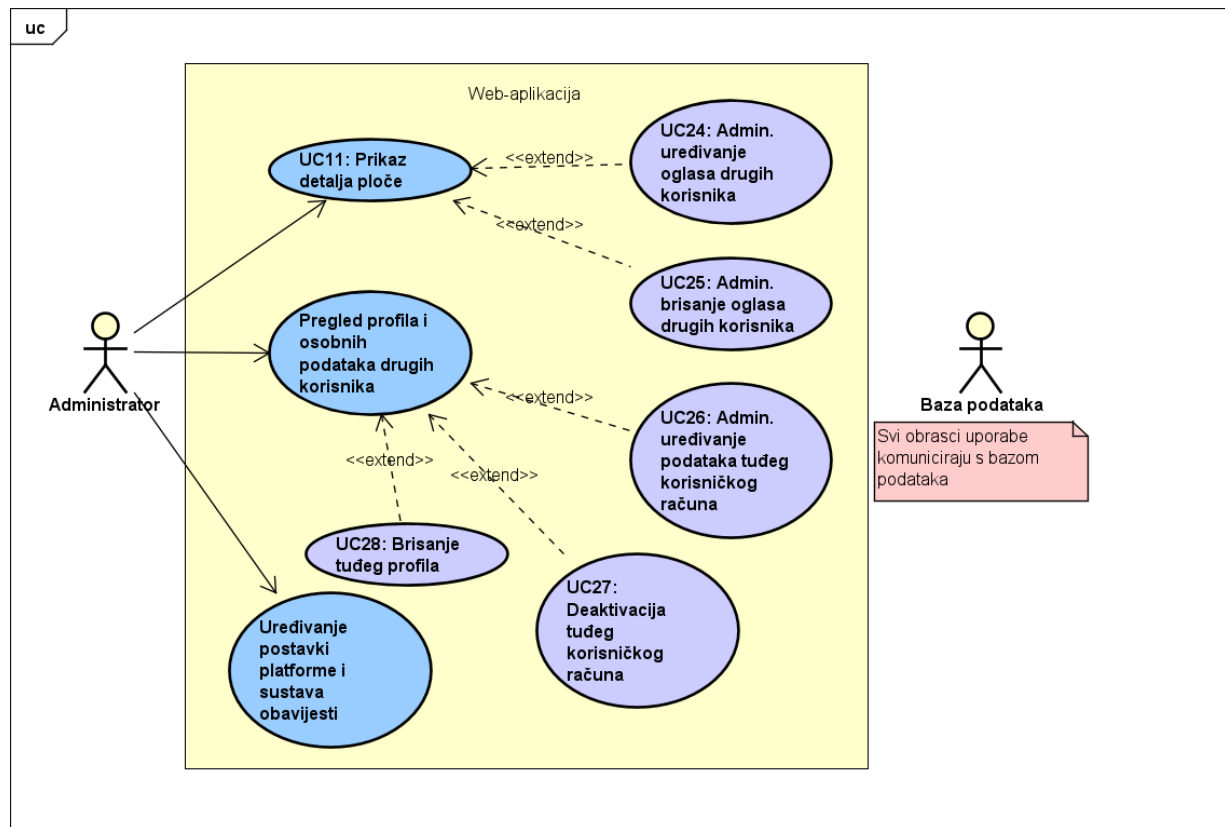


2. Dijagram obrazaca uporabe za ključne značajke



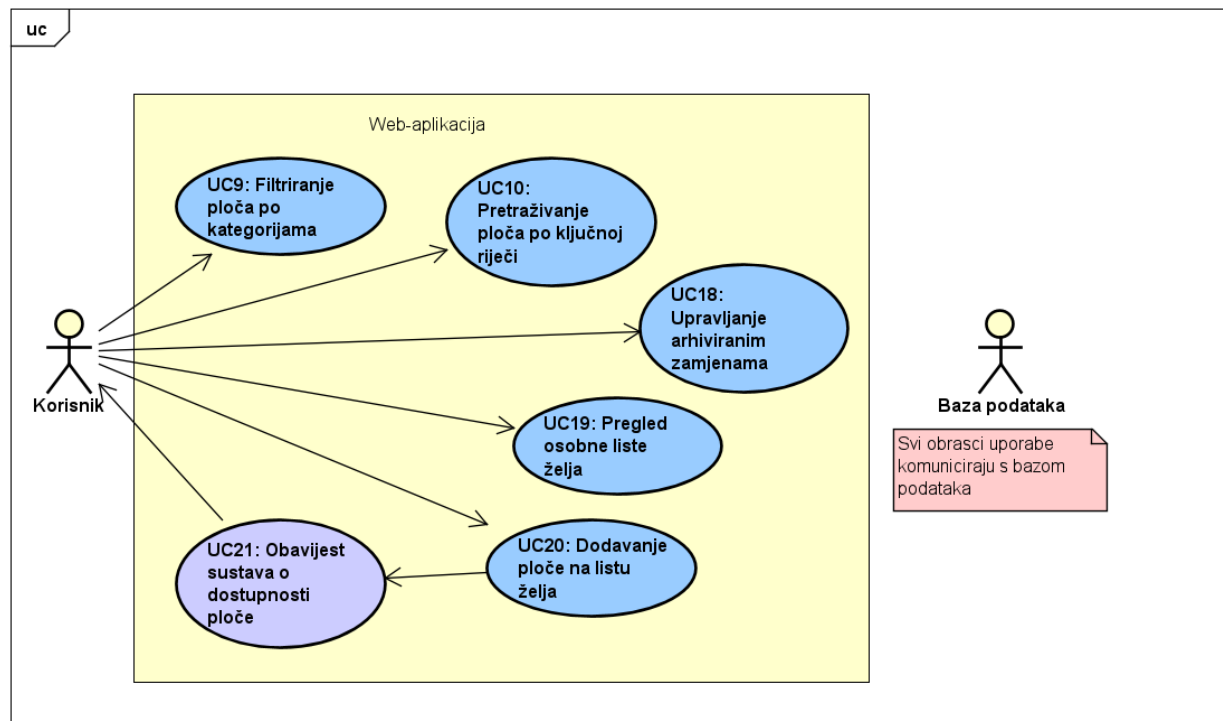
Specifičnije značajke osnovnih korisničkih usluga platforme

3. Dijagram obrazaca uporabe za korisničke uloge



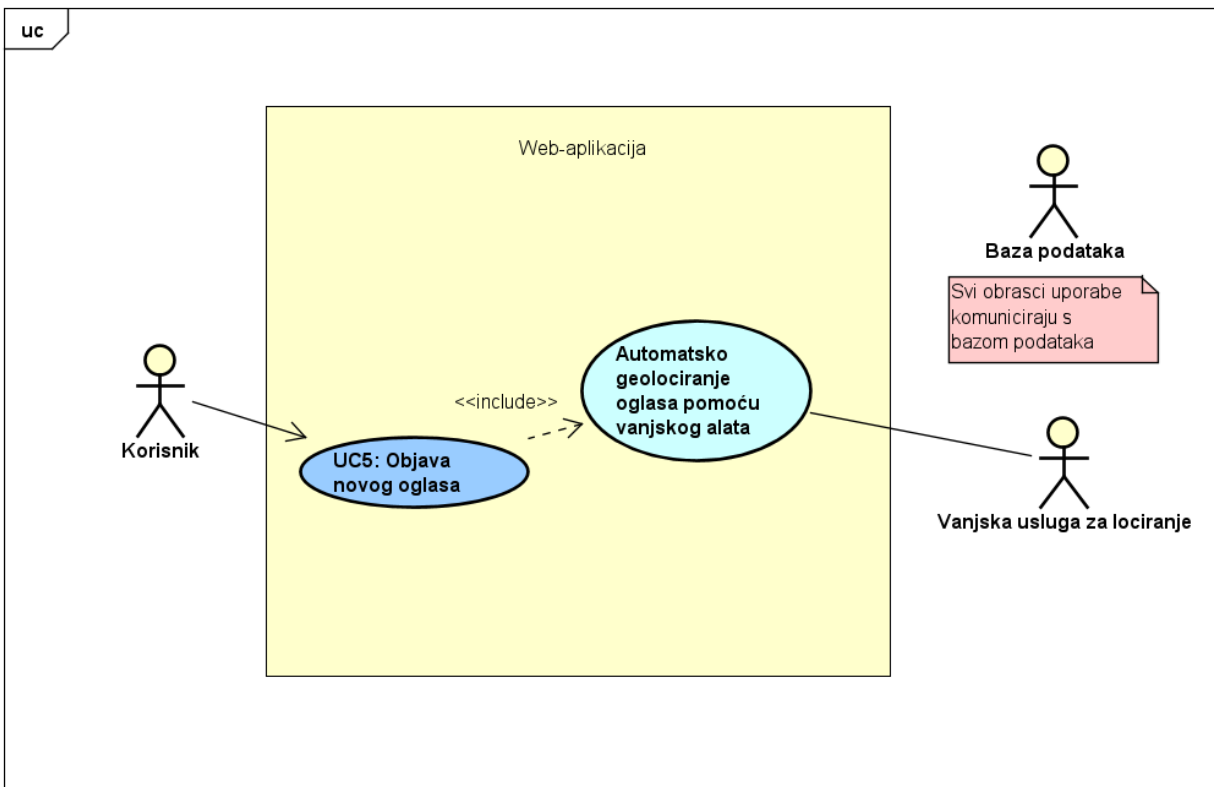
Administratorske funkcionalnosti i interakcije sa sustavom

4. Dijagram obrazaca uporabe za osnovne poslovne procese



Sporedniji obrasci uporabe koji podržavaju poslovne procese organizacije

5. Dijagram obrazaca uporabe za kritične sustave i integracije



Korištenje (integracija) vanjske usluge za lociranje pri objavi oglasa

Opis obrazaca uporabe

UC1 - Registracija

- Glavni sudionik: Korisnik
- Cilj: Stvaranje korisničkog računa
- Sudionici: Baza podataka
- Preuvjeti: -
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju „Registracija” (funkcionalni zahtjev *F-002*)
 2. Korisnik unosi potrebne podatke (adresu e-pošte, korisničko ime, lozinku)
 3. Korisnik odabire opciju „Podnesi”
 4. Korisnik autorizira registraciju koristeći tokene
 5. Korisnik biva preusmjeren na početku stranicu
- Opis mogućih odstupanja:
 - 3.a Korisnik odabire e-mail s kojim je već povezan korisnički račun u sustavu, ili unese tražene podatke u krivom formatu
 1. Korisnik dobiva obavijest o pogrešci

2. Korisnik mijenja podatke i ponovno odabire opciju „Podnesi”, ili odustaje od registracije

UC2 - Prijava u sustav

- Glavni sudionik: Korisnik
- Glavni cilj: dobiti pristup korisničkom dijelu sustava
- Sudionici: Baza podataka
- Preduvjeti: unaprijed registriran korisnički račun (*F-002*)
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Prijava
 2. Korisnik unosi e-mail adresu i lozinku
 3. Korisnik odabire opciju „Podnesi”
 4. Korisnik biva prijavljen i preusmjeren na početnu stranicu
- Opis mogućih odstupanja:
 - 3.a Korisnik upisuje e-mail adresu bez povezanog računa, e-mail u neispravnom formatu ili neispravnu lozinku

Korisnik dobiva poruku pogreške Korisnik ponavlja pokušaj prijave s novim podacima te uspijeva ili odustaje nakon ponovnog neuspjeha

UC3 - Odjava iz sustava

- Glavni sudionik: Korisnik
- Glavni cilj: Odjaviti se iz sustava
- Sudionici: -
- Preduvjet: U trenutku odjave korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju Odjava
 2. Korisnik gubi pristup korisničkom dijelu sustava
 3. Korisnik biva preusmjeren na stranicu „Prijava”

UC4 - Brisanje/deaktivacija korisničkog računa

- Glavni sudionik: Korisnik
- Glavni cilj: Deaktivacija/brisanje osobnog profila
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Korisnik ulazi na svoj profil (pregled osobnih podataka, UC22)
 2. Nakon otvaranja detalja oglasa odabire opciju „Deaktiviraj profil” ili „Izbriši profil”
 3. Korisnik potvrđuje odluku odgovarajućim klikom
 4. Promjene su spremljene te je korisnički račun deaktiviran/izbrisan
 5. Korisnik biva preusmjeren na stranicu prijave

UC5 - Objava novog oglasa za ploču

- Glavni sudionik: Korisnik
- Glavni cilj: Javna objava ponude svoje ploče na platformi
- Sudionici: Baza podataka, vanjska usluga za geolociranje
- Preduvjet: Korisnik je prijavljen u sustav te je dostupna vanjska usluga za geolociranje
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju „Nova ploča”
 2. Korisnik upisuje obavezne podatke o ploči te opcionalan opis ploče (lokacija se automatski određuje vanjskom uslugom)
 3. Korisnik odabire opciju „Objavi ploču”
 4. Objava ploče je uspješna i ona je sad vidljiva ostalim korisnicima na platformi (F-003)
- Opis mogućih odstupanja:
 3. a Korisnik nije ispunio sva polja obaveznih podataka o ploči
 1. Korisniku je dojavljena poruka o grešci.
 2. Korisnik ispunjava preostala prazna polja obaveznih podataka i pokušava ponovno dok ne uspije ili ne odustane.
 3. b Vanjska usluga za geolociranje nije dostupna
 1. Korisniku je dojavljena obavijest o grešci
 2. Korisnik čeka dok usluga nije ponovno dostupna ili odustaje

UC6 - Pregled ploča na stranici „Moje ploče”

- Glavni sudionik: Korisnik
- Glavni cilj: Preglednost popisa svojih oglasa za ploče i lakše uređivanje
- Sudionici: Baza podataka
- Preduvjet: Korisnik je prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik na svom sučelju odabire opciju „Moje ploče”
 2. Otvara se nova stranica na platformi koja na jednom mjestu prikazuje sve objavljene ploče korisnika (F-004)
- Opis mogućih odstupanja:
 - 2.a Korisnik nema objavljenih ploča
 1. Otvara mu se prazna stranica bez oglasa

UC7 - Uređivanje objavljenog oglasa za ploču

- Glavni sudionik: Korisnik
- Glavni cilj: Uređivanje pojedinosti o ploči
- Sudionici: Baza podataka
- Preduvjet: Korisnik je prijavljen u sustav te ima barem 1 objavljen oglas za ploču (F-003)
- Opis osnovnog tijeka:

1. Korisnik ulazi na stranicu „Moje ploče” (*F-004*)
2. Korisnik odabire opciju „Uredi”
3. Korisnik mijenja podatke o ploči koje želi mijenjati
4. Korisnik odabire opciju „Spremi promjene”
5. Promjene su spremljene te korisnik biva preusmjeren na stranicu „Moje ploče”
 - Opis mogućih odstupanja:
 - 5.a Korisnik je ostavio neko od obaveznih polja o ploči prazno
1. Sustav korisniku javlja poruku o grešci te se promjene ne spremaju
2. Korisnik može ispuniti potrebna polja te probati ponovno ili odustati od promjena

UC8 - Brisanje oglasa

- Glavni sudionici: Korisnik
- Glavni cilj: Izbrisati svoj oglas o ploči
- Sudionici: Baza podataka
- Preduvjet: Korisnik je prijavljen te ima barem 1 objavljenu ploču (*F-003*)
- Opis glavnog tijeka:
 1. Korisnik oglas na stranicu „Moje ploče”
 2. Korisnik pronalazi oglas koji želi izbrisati
 3. Na prozoru tog oglasa odabire opciju „Obriši ploču”
 4. Oglas o ploči je izbrisan te više nije vidljiv na stranici „Moje ploče”

UC9 - Filtriranje ploča po kategorijama

- Glavni sudionici: Korisnik
- Glavni cilj: Pretraga ploča drugih korisnika
- Sudionici: Baza podataka
- Preduvjet: Korisnik prijavljen u sustav
- Opis glavnog tijeka:
 1. Korisnik na stranici s ponudama ploča drugih korisnika (*F-001*) odabire opciju „Odaberi žanr”
 2. Otvara se drop-down menu te korisnik klikom odabire ploče kojeg žanra želi vidjeti
 3. Ponuda ploča se filtrira te sada korisnik vidi samo ploče odabranog žanra (*F-005*)

UC10 - Pretraživanje po ključnoj riječi

- Glavni sudionici: Korisnik
- Glavni cilj: Korištenje trake za pretraživanje u svrhu lakšeg nalaženja izvođača ili ploče
- Sudionici: Baza podataka
- Preduvjet: Korisnik prijavljen u sustav
- Opis glavnog tijeka:
 1. Korisnik na stranici s ponudama ploča (*F-001*) otvara traku za pretraživanje
 2. Korisnik upisuje ključne riječi (ime izvođača ili ploče) te odabire opciju „Pretraži”

3. Korisniku se prikazuju ploče koje odgovaraju ključnoj riječi ili riječima (*F-006*)
 - Opis mogućih odstupanja:
 - 3.a Korisnik je u traku za pretraživanje unio ključnu riječ za koju nema podudaranja u listi ponuda
1. Korisniku se prikazuje stranica s porukom o nedostatku podudarajuće ploče

UC11 - Prikaz detalja ploče

- Glavni sudionici: Korisnik
- Glavni cilj: pregled podataka o ploči koju je objavio drugi korisnik
- Sudionici: Baza podataka
- Opis glavnog tijeka:
 1. Korisnik na stranici s ponudama (ili nakon filtriranja/pretraživanja iste) pronalazi ploču koja ga interesira (*F-001*)
 2. Korisnik lijevim klikom miša na prozorčić koji prikazuje ploču otvara prikaz informacija o ploči

UC12 - Slanje ponude za zamjenu

- Glavni sudionici: Korisnik
- Glavni cilj: Korisnik nudi svoje ploče u zamjenu za ploču koja ga zanima
- Sudionici: Baza podataka
- Preduvjet: Korisnik prijavljen u sustav te ima objavljenu barem jednu svoju ploču (*F-003*)
- Opis glavnog tijeka:
 1. Korisnik otvara prikaz detalja ploče koja ga zanima (*F-001*)
 2. Korisnik odabire opciju „Ponudi zamjenu”
 3. Otvara se prozor s prikazom korisnikovih vlastitih objavljenih ploča
 4. Korisnik odabire ploču (ili ploče) koje želi ponuditi u razmjeni
 5. Nakon odabira svojih ploča korisnik odabire opciju „Pošalji”
 6. Ponuda je poslana te se prozor zatvara (*F-007*)

UC13 - Pregled primljenih ponuda za zamjenu preko stranice

- Glavni sudionici: Korisnik
- Glavni cilj: Prikaz i pregled ponuda koje je korisnik dobio za svoje ploče
- Sudionici: Baza podataka
- Preduvjet: Korisnik prijavljen u sustav
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju (pregled) „Ponude”
 2. Korisniku se prikazuje popis svih primljenih ponuda za ploče (*F-007*)
- Opis mogućih odstupanja:
 - 2.a Korisnik nije zaprimio nijednu ponudu

1. Stranica s ponudama je prazna

UC14 - Pregled primljenih ponuda za zamjenu preko e-maila

- Glavni sudionici: Korisnik
- Glavni cilj: Pojedinački pregled ponude koju je korisnik dobio za svoju ploču
- Sudionici: Baza podataka, e-mail usluga
- Preduvjet: Zaprimljena ponuda za ploču (*F-007*)
- Opis osnovnog tijeka:
 1. Korisnik otvara svoju e-mail uslugu
 2. U inboxu pronalazi i otvara poruku poslanu od strane GramoFer-a
 3. Klikom na link u poruci korisniku se otvara pregled zaprimljene ponude za ploču

UC15 - Prihvaćanje ponude za zamjenu

- Glavni sudionik: Korisnik
- Glavni cilj: Prihvaćanje ponude i provedba zamjene ploča s drugim korisnikom
- Sudionici: Baza podataka
- Preduvjet: Zaprimljena ponuda za ploču
- Opis osnovnog tijeka:
 1. Korisnik pristupa ponudi (*F-007*) putem stranice (UC13) ili putem maila (UC14)
 2. Korisnik odabire opciju „Prihvati zamjenu”
 3. Zamjena je uspješno provedena te arhivirana
 4. Korisnik biva preusmjeren na početnu stranicu

UC16 - Odbijanje ponude za zamjenu

- Glavni sudionik: Korisnik
- Glavni cilj: Odbijanje ponude za zamjenu ploča
- Sudionici: Baza podataka
- Preduvjet: Zaprimljena ponuda za ploču
- Opis osnovnog tijeka:
 1. Korisnik pristupa ponudi (*F-007*) putem stranice (UC13) ili putem maila (UC14)
 2. Korisnik odabire opciju „Odbij zamjenu”
 3. Odbijanje je provedeno
 4. Ponuda više nije dostupna pod „Moje ponude” (izbrisana je)

UC17 - Uređivanje ponude za zamjenu

- Glavni sudionik: Korisnik
- Glavni cilj: Uređivanje ponude za zamjenu ploča drugog korisnika
- Sudionici: Baza podataka
- Preduvjet: Zaprimljena ponuda za ploču
- Opis osnovnog tijeka:
 1. Korisnik pristupa ponudi (*F-007*) putem stranice (UC13) ili putem maila (UC14)

2. Korisnik odabire opciju „Uredi zamjenu”
3. Korisnik uređuje detalje zamjene po svojim željama (npr. traži drugu ploču ponuditelja)
4. Korisnik odabire opciju „Pošalji”
5. Ponuda je uređena te poslana originalnom ponuditelju
6. Korisniku ponuda više nije vidljiva na stranici „Moje ponude”

UC18 - Upravljanje arhiviranim zamjenama

- Glavni sudionik: Korisnik
- Glavni cilj: Uvid u pregledan popis svih provedenih zamjena
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju „Moje zamjene”
 2. Korisniku se otvara pregled svih provedenih zamjena (ili poruka „Niste još proveli nijednu zamjenu”)

UC19 - Pregled osobne liste želja

- Glavni sudionik: Korisnik
- Glavni cilj: Pregledan popis željenih ploča
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju „Lista želja”
 2. Otvara se pregled kojim korisnik vidi sve ploče koje je dodao u listu želja
- Opis mogućih odstupanja:
 - 2.a Lista želja je prazna
 1. Ispisuje se poruka „Nimate nijednu ploča na listi želja”

UC20 - Dodavanje ploča na listu želja

- Glavni sudionik: Korisnik
- Glavni cilj: Kreiranje liste željenih ploča
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Korisnik traži željenu ploču (UC9, UC10)
 2. Korisnik klikom na nju otvara njene detalje (UC11)
 3. Korisnik odabire opciju „Dodaj na listu želja”
 4. Ploča je dodana na listu želja (*F-008*) i sad je vidljiva pregledom stranice „Lista želja” (UC19)

UC21 - Obavijest o dostupnosti ploča

- Glavni sudionik: Baza podataka
- Glavni cilj: Obavješćavanje korisnika da je ploča s njegove liste želja dostupna

- Sudionici: E-mail usluga, korisnik
- Opis osnovnog tijeka:
 1. U bazu podataka dodana je ponuda ploče koja je dio liste želja korisnika
 2. Korisniku se prikaže obavijest na stranici
 3. Korisniku je poslana obavijest e-mailom

UC22 - Pregled profila i osobnih podataka korisnika

- Glavni sudionik: Korisnik
- Glavni cilj: Pregled svojih osobnih podataka
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju „Moj profil”
 2. Otvara mu se prikaz profila s osobnim podacima

UC23 - Uređivanje osobnih podataka

- Glavni sudionik: Korisnik
- Glavni cilj: Uređivanje svojih osobnih podataka profila
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Korisnik otvara svoj profil (UC22)
 2. Korisnik odabire opciju „Uredi profil”
 3. Korisnik mijenja/dodaje podatke koje želi
 4. Korisnik odabire opciju „Spremi promjene”
 5. Promjene su spremljene te korisnik biva preusmjeren na prikaz svog profila

UC24 - Administratorsko uređivanje oglasa

- Glavni sudionik: Administrator
- Glavni cilj: Uređivanje oglasa koji krše neke propise platforme ili nije adekvatan
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Administrator pronalazi neadekvatni oglas drugog korisnika (*F-001*)
 2. Nakon otvaranja detalja oglasa odabire administratorsku opciju „Uredi oglas”
 3. Administrator uređuje oglas i odabire administratorsku opciju „Spremi promjene”
 4. Promjene su spremljene (*F-010*) te korisnik obaviješten o izmjeni oglasa

UC25 - Administratorsko brisanje oglasa

- Glavni sudionik: Administrator
- Glavni cilj: Brisanje oglasa koji krše neke propise platforme ili nije adekvatan
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Administrator pronalazi neadekvatni oglas drugog korisnika (*F-001*)

2. Nakon otvaranja detalja oglasa odabire administratorsku opciju „Obriši oglas”
3. Administrator potvrđuje brisanje
4. Oglas je obrisao (*F-010*), a korisnik obaviješten o brisanju oglasa

UC26 - Administratorsko uređivanje korisničkog računa

- Glavni sudionik: Administrator
- Glavni cilj: Uređivanje računa koji krše neke propise platforme ili nije adekvatan
- Sudionici: Baza podataka
- Opis osnovnog tijeka:
 1. Administrator pronalazi neadekvatni račun običnog korisnika
 2. Nakon otvaranja detalja profila odabire administratorsku opciju „Uredi profil”
 3. Administrator uređuje oglas i odabire administratorsku opciju „Spremi promjene”
 4. Promjene su spremljene (*F-009*) te korisnik obaviješten o izmjeni profila

UC27 - Administratorsko deaktiviranje korisničkog računa

- Glavni sudionik: Administrator
- Glavni cilj: Deaktivacija korisničkog računa koji krši neke propise platforme ili nije adekvatan
- Sudionici: Baza podataka, e-mail usluga
- Opis osnovnog tijeka:
 1. Administrator pronalazi neadekvatni oglas drugog korisnika
 2. Nakon otvaranja detalja oglasa odabire administratorsku opciju „Deaktiviraj profil”
 3. Administrator potvrđuje odluku odgovarajućim klikom
 4. Promjene su spremljene (*F-009*) te je korisnički račun deaktiviran, a korisnik obaviješten putem e-maila

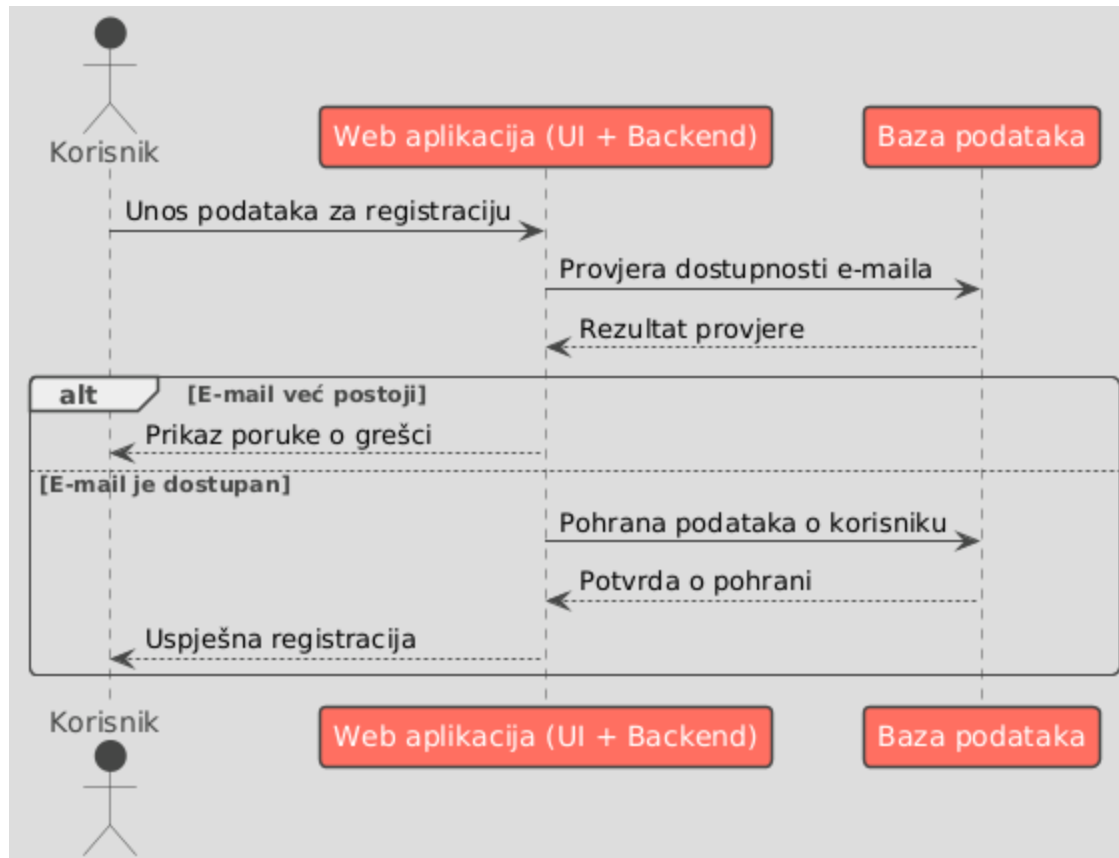
UC28 - Administratorsko brisanje korisničkog računa

- Glavni sudionik: Administrator
- Glavni cilj: Brisanje korisničkog računa koji krši neke propise platforme ili nije adekvatan
- Sudionici: Baza podataka, e-mail usluga
- Opis osnovnog tijeka:
 1. Administrator pronalazi neadekvatni oglas drugog korisnika
 2. Nakon otvaranja detalja oglasa odabire administratorsku opciju „Izbriši profil”
 3. Administrator potvrđuje odluku odgovarajućim klikom
 4. Promjene su spremljene (*F-009*) te je korisnički račun izbrisan, a korisnik obaviješten putem e-maila

Sekvencijski dijagrami

Registracija - obrazac uporabe UC1

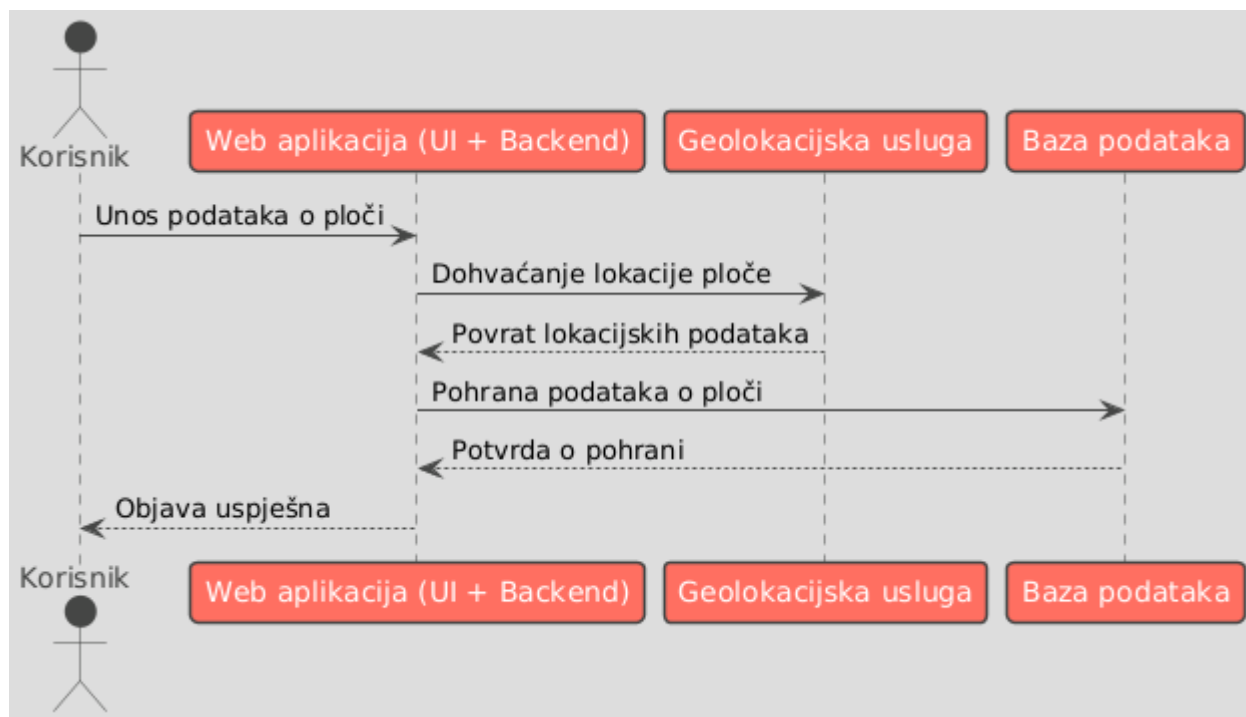
Neregistrirani korisnik upisuje korisničko ime, e-mail i lozinku. Klikom na gumb „Sign up” ti podaci šalju se na validaciju. Ako su uneseni podaci (e-mail) pronađeni u bazi podataka, aplikacija generira token registracije. U suprotnom, šalje se poruka o neispravnoj e-mail adresi.



Sekvencijski dijagram registracije korisnika

Objava oglasa za ploču - obrazac uporabe UC5

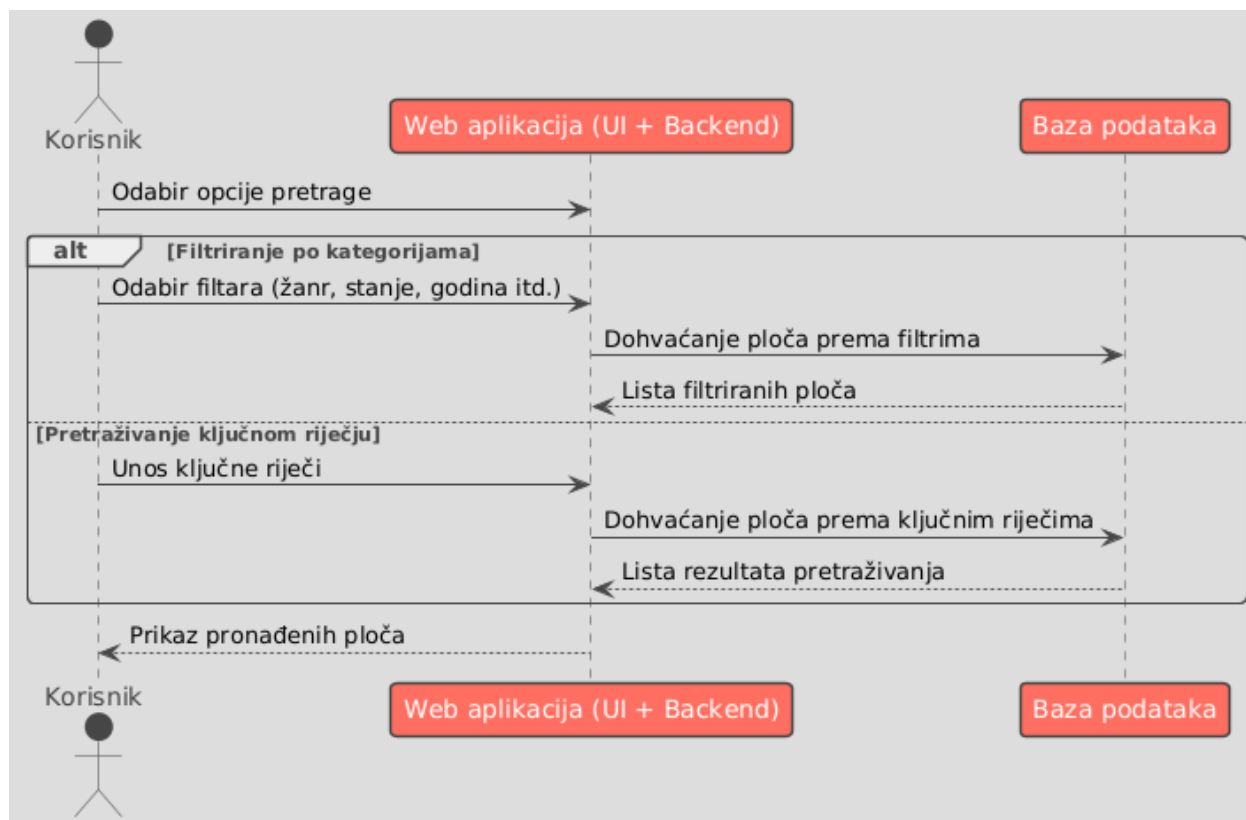
Prijavljeni korisnik odabire opciju „Nova ploča” i nakon upisivanja detalja odabire opciju „Objavi ploču”. Web aplikacija komunicira s backendom koji dohvaća lokaciju oglasa od vanjske usluge te pohranjuje oglas u bazu podataka.



Sekvencijski dijagram objave oglasa za ploču

Pretraživanje ploča - obrasci uporabe UC9 i UC10

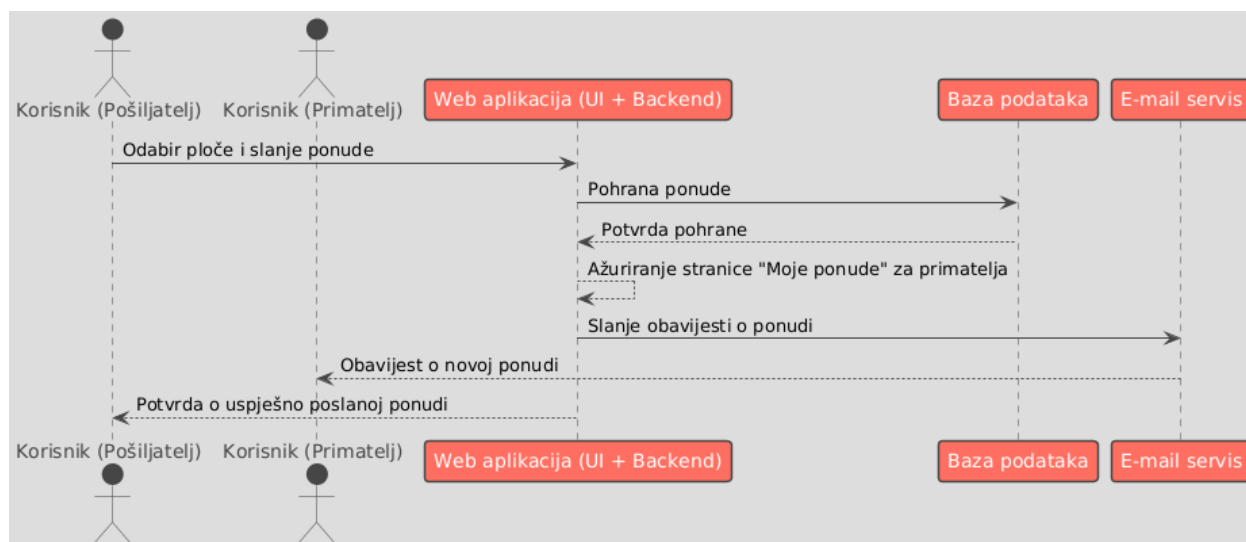
Korisnik odabire između dvije opcije pretraživanja: filtriranjem (UC9) ili po ključnoj riječi (UC10). Nakon toga rezultati se dohvaćaju preko backenda iz baze podataka.



Sekvencijski dijagram pretraživanja ploča

Slanje ponude za zamjenu - UC12

Korisnik odabire ploču te šalje svoju ponudu. Detalji ponude se preko backenda spremaju u bazu podataka, a onda se preko backenda ažurira primateljeva stranica „Moje ponude” te aktivira e-mail usluga koja šalje tu istu ponudu primatelju. Uspješno slanje ponude se potvrđuje ponuditelju.



Sekvencijski dijagram slanja ponude za zamjenu

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

4. Arhitektura i dizajn sustava

Arhitektura sustava

Opis arhitekture

- Stil arhitekture: Klijent-poslužitelj arhitektura
- Podsustavi: Podsustav za korisnika osigurava da korisnik može pristupiti svim funkcionalnostima ako ima dopuštenje od sustava. Baza podataka se koristi za spremanje objavljenih ploča i podataka o razmjenama, željama i korisnicima. Web poslužitelj odgovara na sve zahtjeve koje korisnik u frontendu zatraži.
- Preslikavanje na radnu platformu: Aplikacija će biti pokrenuta putem Azure virtual machine. Cloudinary će se koristiti za fotografije ploča, a sql server za pohranu podataka.
- Spremište podataka: Podaci sustava pohranjuju se u relacijsku bazu podataka koristeći PostgreSQL, dok se medijske datoteke pohranjuju na cloud platformu Cloudinary.
- Mrežni protokoli: HTTPS, OAuth 2.0
- Globalni upravljački tok: Nakon što korisnik pristupi stranici, ukoliko želi sve funkcionalnosti, mora se prijaviti putom Google OAuth2.0 protokola. Kako korisnik šalje zahtjeve, oni se obrađuju u backendu i po potrebi se pohranjuju u bazu podataka.
- Sklopovskoprogramski zahtjevi: Server na kojem se nalazi baza podataka mora podržavati višekorisnički način rada. React aplikaciju (frontend) mora biti moguće otvoriti na računalima i mobilnim uređajima. Cijela aplikacija mora biti pokrenuta konstantno na virtualnom stroju.

Obrazloženje odabira arhitekture

- Stil arhitekture: Ovaj pristup omogućuje integraciju različitih tehnologija i rješavanje specifičnih zahtjeva svakog podsustava, što doprinosi pouzdanosti i učinkovitosti.
- Spremište podataka: Relacijska baza podataka je odabrana jer je najprikladnija za rad s usko povezanim podacima, a PostgreSQL je izabran zbog svoje stabilnosti, brzine obrade i prilagodljivosti. Za pohranu slika odabrana je platforma Cloudinary jer omogućuje jednostavnu pohranu i rukovanje velikim brojem slika.
- Mrežni protokoli: HTTPS protokol je standard za osiguravanje šifrirane i sigurne razmjene podataka

- Sklopovskoprogramski zahtjevi: Server na kojem se nalazi baza podataka podržava višekorisnički način rada.

Organizacija sustava na visokoj razini

- Klijent-poslužitelj: Klijent je frontend(React), a poslužitelj backend(Spring Boot). Putem frontenda klijenti komuniciraju s cijelim sustavom, dok backend manipulira podacima
- Baza podataka: Koristi se PostgreSQL relacijska baza podataka čija je uloga pohranjivanje, upravljanje i omogućavanje pristupa podacima sustava.
- Datotečni sustav: Koristi se cloud platforma Cloudinary čija je uloga pohranjivanje i upravljanje medijskim datotekama.
- Grafičko sučelje: Frontend je razvijen u React aplikaciji i putem fetch zahtjeva uzima podatke iz backenda.

Organizacija aplikacije

- Frontend i Backend slojevi: Frontend je realiziran putem React aplikacije. Putem API komunicira s backendom. Backend je realiziran putem Spring Boot aplikacije, koja obrađuje podatke u sustavu i implementira MVC arhitekturu. Kontroleri primaju zahtjeve iz frontenda.
- MVC arhitektura:
 - Model: upravlja podacima o korisnicima i gramofonskim pločama
 - View: prikaz React frontenda
 - Controller: zaprimanje fetch zahtjeva s frontenda i dostavljanje odgovarajućih podataka nazad

Baza podataka

Za pohranu podataka u sustavu odabrana je PostgreSQL relacijska baza podataka koja organizira podatke u tablice. Tablice predstavljaju entitete sa skupom atributa, a povezanost između entiteta ostvaruje se putem relacija, čime se omogućuje efikasno pohranjivanje i upravljanje podacima. Za potrebe našeg projekta, baza podataka obuhvaća sedam entiteta:

- user_account
- edition
- genre
- vinyl
- wish
- exchange
- reports

Opis tablica

U tabličnim prikazima entiteta u nastavku primarni ključevi označeni su plavom bojom, jedinstveni ključevi žutom bojom te strani ključevi zelenom bojom.

user_account – Ovaj entitet sadrži podatke o korisničkim računima te je u vezama: *One-to-Many* s entitetom *vinyl* preko atributa *user_id*, *One-to-Many* s entitetom *wish* preko atributa *user_id*, *One-to-Many* s entitetom *reports* preko atributa *user_id* te dvije veze *One-to-Many* s entitetom *exchange*, jedna preko atributa *user_id* i druga preko atributa *is_offering_user_id*.

user_account		
Atribut	Tip podataka	Opis varijable
user_id	INT	Identifikacijski ključ korisnika
username	VARCHAR	Jedinstveno korisničko ime
email	VARCHAR	Jedinstvena email adresa
password	VARCHAR	Korisnička lozinka
first_name	VARCHAR	Ime korisnika
last_name	VARCHAR	Prezime korisnika
is_admin	INT	Oznaka je li korisnik administrator
registration_date	DATE	Datum registracije računa
google_id	VARCHAR	Jedinstveni identifikator Google računa
strike_count	INT	Broj prijava za taj korisnički račun

edition – Ovaj entitet sadrži osnovne podatke za izdanja ploče te je u vezama: *One-to-Many* s entitetom *wish* preko entiteta *edition_label*, *Many-to-Many* s entitetom *genre* preko atributa *edition_label* i *genre_id* te *One-to-One* s entitetom *vinyl* preko atributa *edition_label*.

edition		
Atribut	Tip podataka	Opis varijable
edition_label	VARCHAR	Jedinstvena oznaka izdanja ploče
artist_name	VARCHAR	Ime izvođača ploče
release_date	INT	Godina izdavanja ploče
album_name	VARCHAR	Naziv albuma na ploči
country_of_origin	VARCHAR	Lokacija podrijetla ploče

genre – Ovaj entitet sadrži nazive žanrova te je u vezi Many-to-Many s entitetom *edition* preko atributa *genre_id* i *edition_label*.

genre		
Atribut	Tip podataka	Opis varijable
genre_id	INT	Identifikacijski ključ žanra
genre_name	VARCHAR	Naziv žanra

vinyl – Ovaj entitet sadrži dodatne podatke o objavljenim pločama te je u vezama: Many-to-One s entitetom *user_account* preko atributa *user_id*, One-to-One s entitetom *edition* preko atributa *edition_label*, One-to-Many s entitetom *exchange* preko atributa *vinyl_id* te Many-to-Many s entitetom *exchange* preko atributa *exchange_id* i *vinyl_id*.

vinyl		
Atribut	Tip podataka	Opis varijable
vinyl_id	INT	Identifikator ploče
vinyl_condition	VARCHAR	Očuvanost ploče prema Goldmine standardu
cover_condition	VARCHAR	Očuvanost omota prema Goldmine standardu
description	VARCHAR	Korisnikov opis ploče
vinyl_image_path1	VARCHAR	Putanja do prve slike ploče
vinyl_image_path2	VARCHAR	Putanja do druge slike ploče
cover_image_path1	VARCHAR	Putanja do prve slike omota
cover_image_path2	VARCHAR	Putanja do druge slike omota
available	INT	Oznaka je li ploča slobodna za razmjenu
on_location	VARCHAR	Lokacija dostupnosti ploče
user_id	INT	Povezuje ploču s korisničkim računom
edition_label	VARCHAR	Povezuje objavu ploče s osnovnim podacima o samoj ploči

wish – Ovaj entitet sadrži podatke o željama korisnika za određenim pločama koje žele nabaviti i za koju čekaju objavu u sustavu te je u vezama: Many-to-One s entitetom *user_account* preko atributa *user_id* i Many-to-One s entitetom *edition* preko atributa *edition_label*.

wish		
Atribut	Tip podataka	Opis varijable
wish_id	INT	Identifikacijski ključ želje
album_name	VARCHAR	Naziv albuma kojeg korisnik želi
artist_name	VARCHAR	Ime izvođača čiji album korisnik želi
user_id	INT	Povezuje želju s korisničkim računom
edition_label	VARCHAR	Povezuje želju s izdanjem ploče

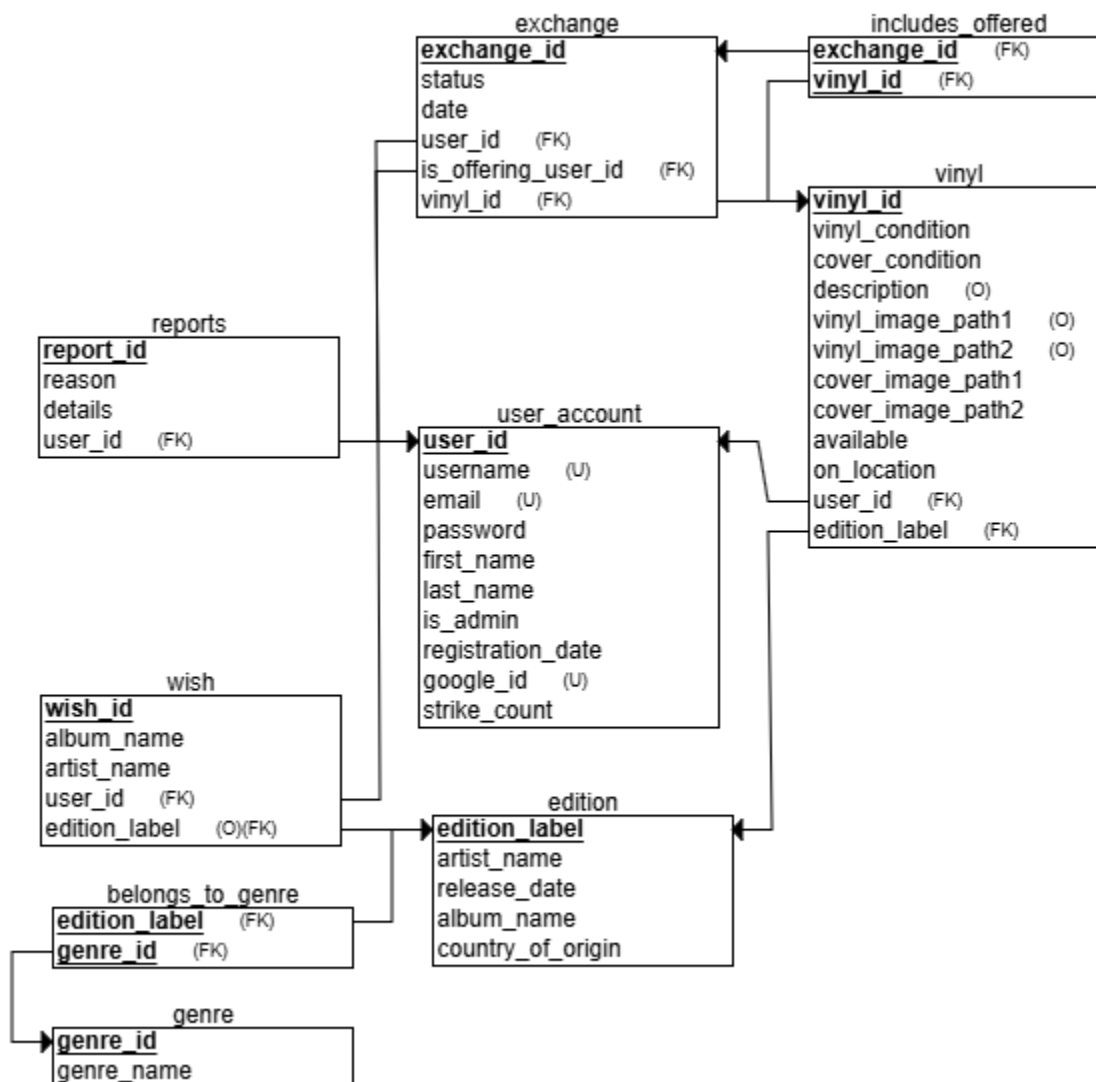
exchange – Ovaj entitet sadrži podatke o aktivnim zamjenama ploča među korisnicima te je u vezama: Many-to-One s entitetom *vinyl* preko atributa *vinyl_id* (po zamjeni jedna zatražena ploča), Many-to-Many s entitetom *vinyl* preko atributa *exchange_id* i *vinyl_id* (po zamjeni jedna ili više ponuđenih ploča) te dvije Many-to-One s entitetom *user_account* preko atributa *user_id* od kojih je jedna za korisnika čija se ploča traži, a druga za korisnika koji nudi ploče za zamjenu.

exchange		
Atribut	Tip podataka	Opis varijable
Exchange_id	INT	Identifikacijski ključ zamjene
status	VARCHAR	Status zamjene: “U tijeku” / “Prihvaćeno” / “Odbijeno”
date	DATE	Datum koji se osvježava ovisno o statusu
User_id	INT	Povezuje zamjenu s korisničkim računom čija se ploča traži
Is_offering_user_id	INT	Povezuje zamjenu s korisničkom računom koji nudi ploče za zamjenu
Vinyl_id	INT	Povezuje zamjenu sa zatraženom pločom

reports – Ovaj entitet sadrži podatke o prijavama korisničkih računa te je u vezi Many-to-One s entitetom *user_account* preko atributa *user_id*.

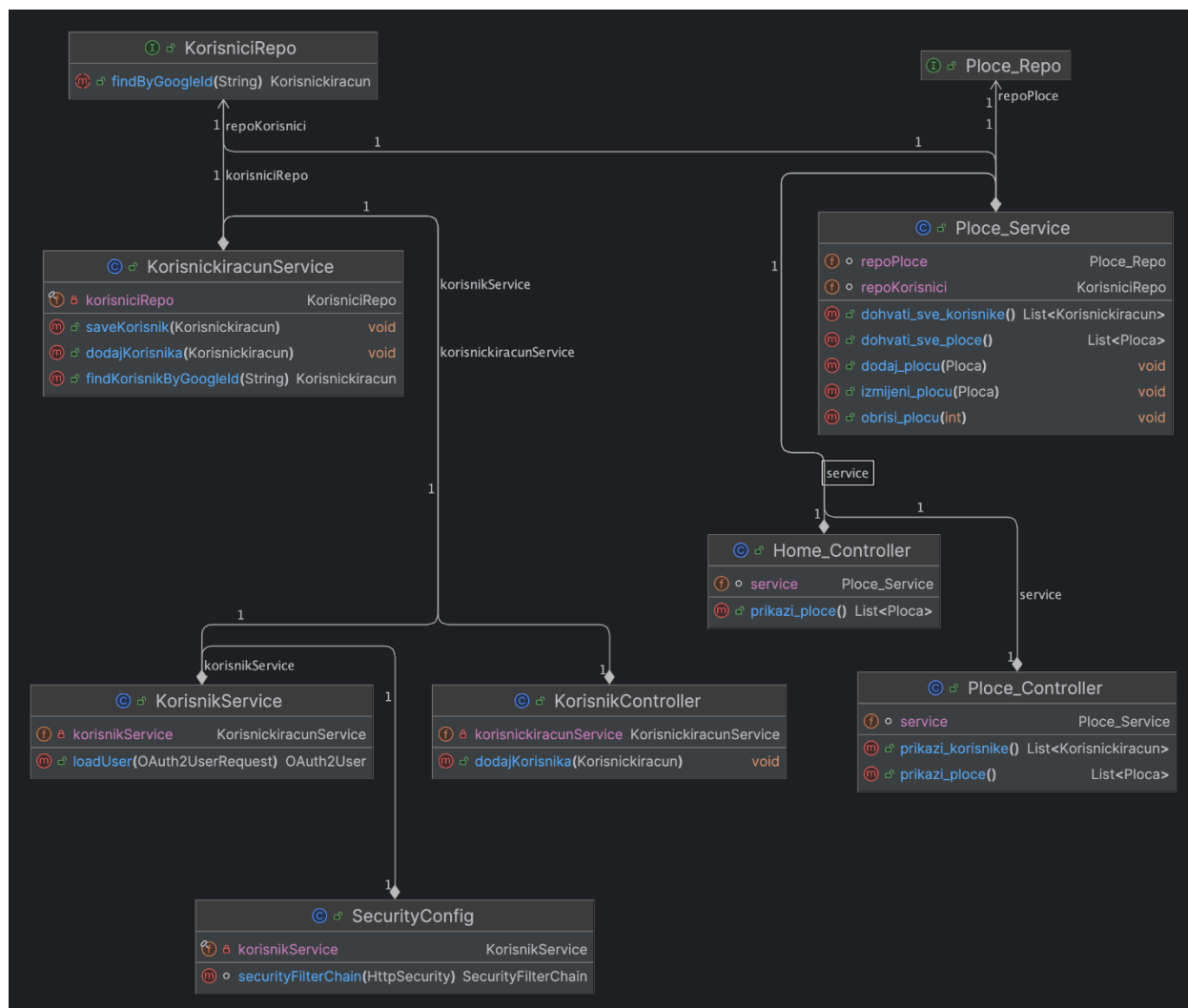
reports		
Atribut	Tip podataka	Opis varijable
report_id	INT	Identifikacijski ključ prijave
reason	VARCHAR	Razlog prijave
details	VARCHAR	Detaljniji opis prijave
user_id	INT	Povezuje prijavu s korisničkim računom

Dijagram baze podataka

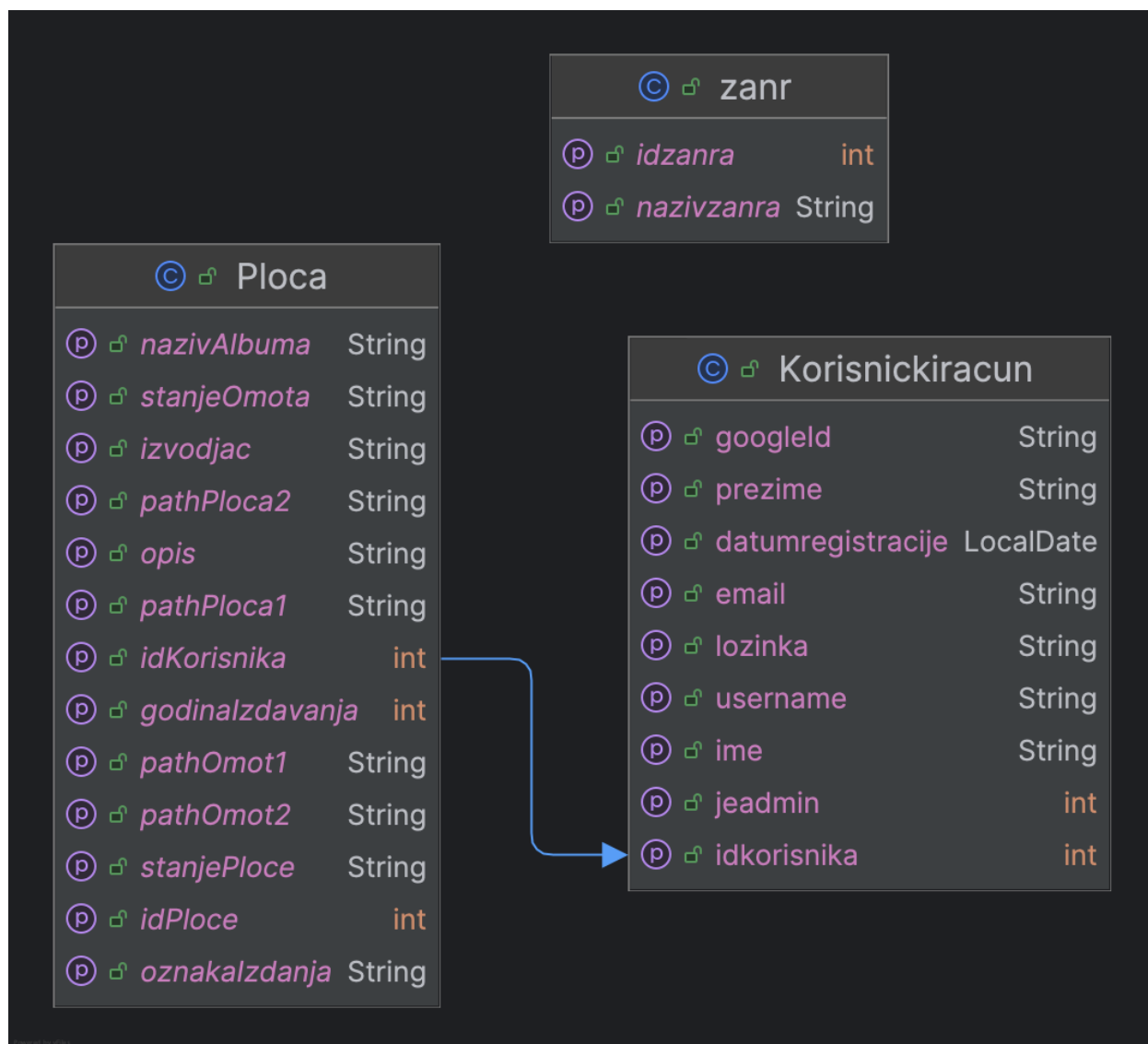


Dijagram razreda

Na slikama su prikazani razredi koji čine backend dio MVC arhitekture.



Razredi prikazani na slici 1 služe dohvaćanju podataka iz baze te obradu i konačno slanje na frontend. Metode implementirane u ove razrede upravljaju podacima iz modela i vraćaju tražene podatke.



Razredi prikazani na slici 2 su modeli i služe preslikavanju podataka iz baze u aplikaciju. Modeli predstavljaju entitete iz baze podataka. Članske varijable svake klase odgovaraju pripadajućeg entiteta u bazi podataka.

Dinamičko ponašanje aplikacije

UML dijagrami stanja

Status korisničkog računa

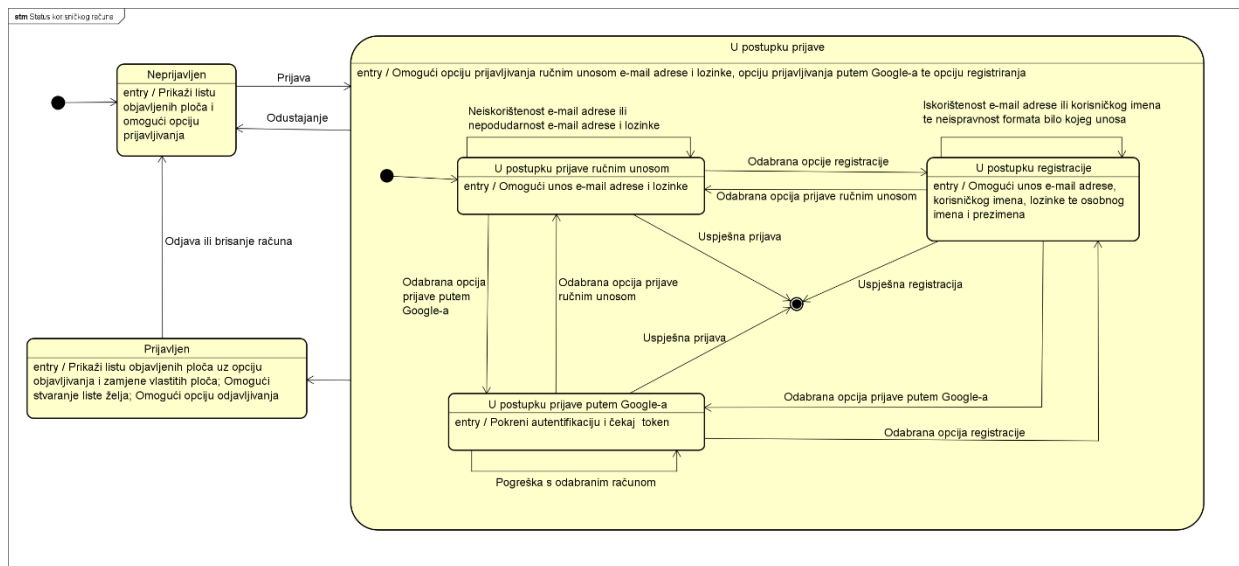
Dijagram stanja za korisnički račun može uključivati:

- Stanja: A) *Neprijavljen*, B) *U postupku prijave*, C) *Prijavljen*
- Prijelaze:
 1. Iz stanja (A) u stanje (B) uz uvjet odabrane opcije prijavljivanja

2. Iz stanja (B) u stanje (A) uz uvjet odustajanja
3. Iz stanja (B) u stanje (C) uz uvjet uspješne prijave ili registracije
4. Iz stanja (C) stanje (A) uz uvjet odjave ili brisanja računa

Dijagram podstanja za stanje *U postupku prijave* može uključivati:

- Podstanja: A) *U postupku prijave ručnim unosom*, B) *U postupku prijave putem Google-a*, C) *U postupku registracije*
- Prijelaze:
 1. Iz stanja (A) u stanje (A) uz uvjet da unesena e-mail adresa nije prethodno iskorištena ili ako se kombinacija e-mail adrese i lozinke ne podudaraju
 2. Iz stanja (B) u stanje (B) uz uvjet pogreške s autorizacijom
 3. Iz stanja (C) u stanje (C) uz uvjet da je unesena e-mail adresa prethodno iskorištena ili ako je neispravan format bilo kojeg unosa
 4. Iz stanja (B) i (C) u stanje (A) uz uvjet odabira opcije prijave ručnim unosom
 5. Iz stanja (A) i (C) u stanje (B) uz uvjet odabira opcije prijave putem Google-a
 6. Iz stanja (A) i (B) u stanje (C) uz uvjet odabira opcije registracije
 7. Iz stanja (A) i (B) u konačno stanje uz uvjet uspješne prijave
 8. Iz stanja (C) u konačno stanje uz uvjet uspješne registracije



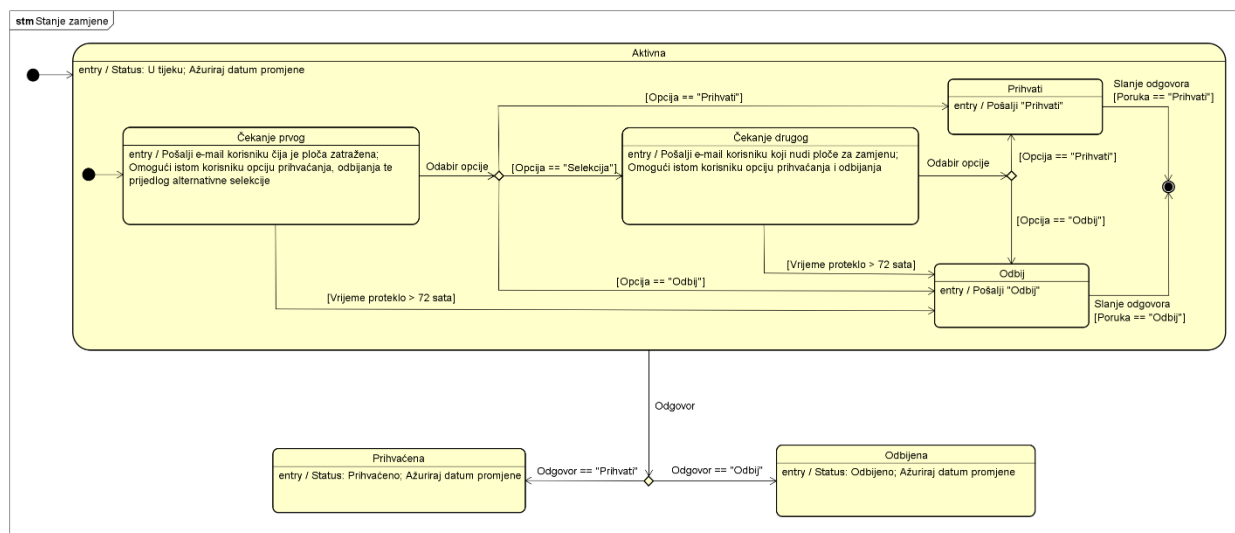
Stanje zamjene

Dijagram stanja za zamjenu ploča može uključivati:

- Stanja: (A) *Aktivna*, (B) *Prihvaćena*, (C) *Odbijena*
- Prijelaze:
 1. Iz stanja (A) u stanje (B) uz uvjet primljenog odgovora “Prihvati”
 2. Iz stanja (A) u stanje (C) uz uvjet primljenog odgovora “Odbij”

Dijagram podstanja za stanje *Aktivna* može uključivati:

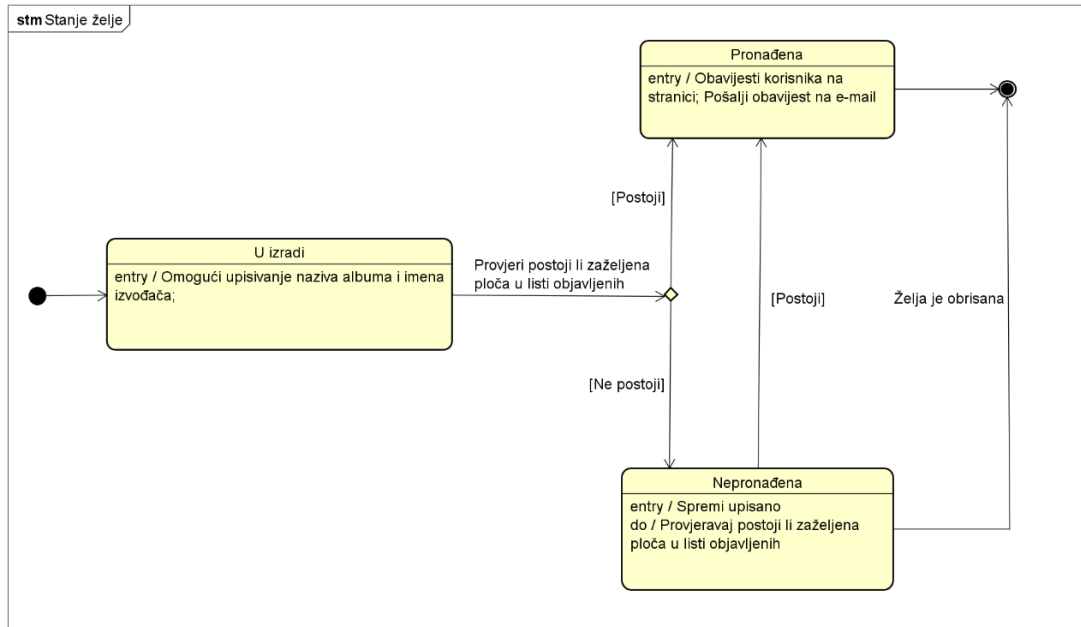
- Podstanja: (A) *Čekanje prvog*, (B) *Čekanje drugog*, (C) *Prihvati*, (D) *Odbij*
- Prijelaze:
 1. Iz stanja (A) u stanje (B) uz uvjet predložene alternativne selekcije
 2. Iz stanja (A) i (B) u stanje (C) uz uvjet prihvatanja ponude
 3. Iz stanja (A) i (B) u stanje (D) uz uvjet odbijanja ponude ili ako odgovor nije poslan unutar 72 sata
 4. Iz stanja (C) ili (D) u konačno stanje



Stanje želje

Dijagram stanja za status zaželjene ploče može uključivati:

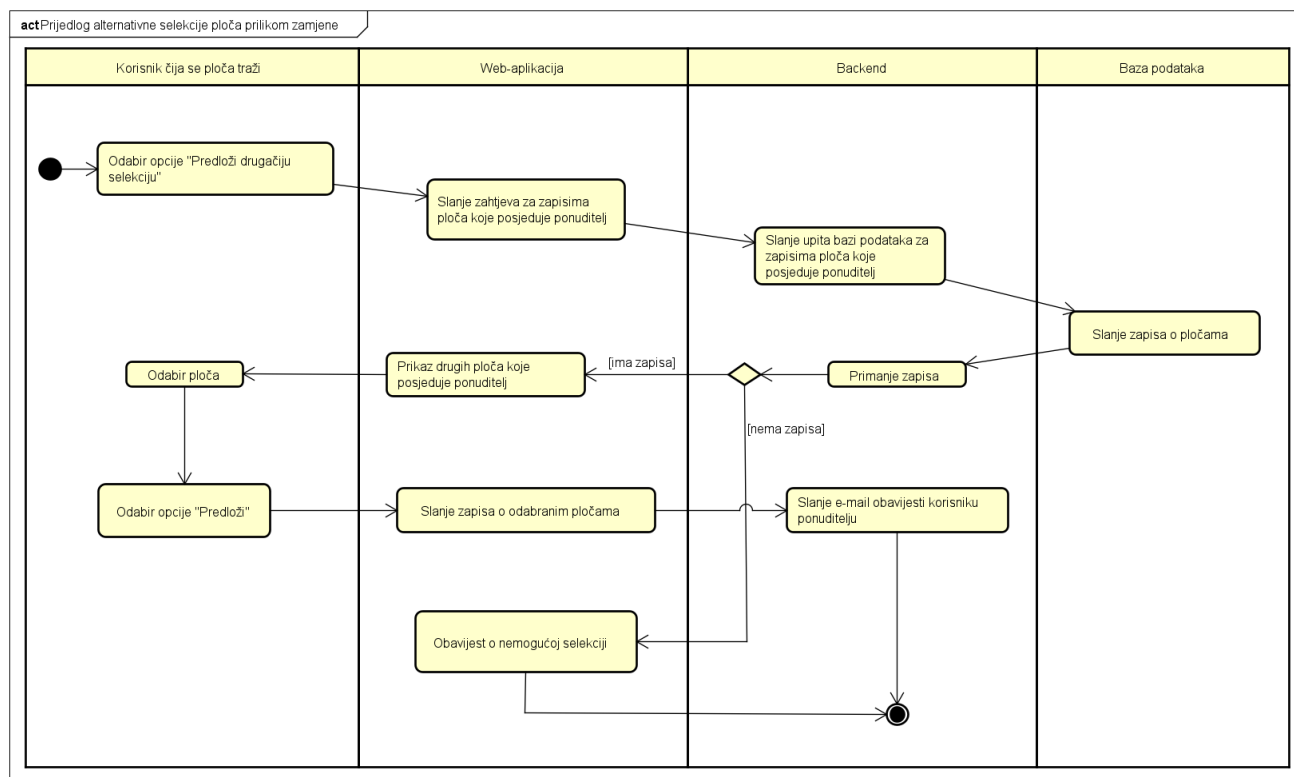
- Stanja: (A) *U izradi*, (B) *Pronađena*, (C) *Nepronadena*
- Prijelaze:
 1. Iz stanja (A) u stanje (B) uz uvjet da je zaželjena ploča već objavljena
 2. Iz stanja (A) u stanje (C) uz uvjet da zaželjena ploča nije objavljena
 3. Iz stanja (C) u stanje (B) uz uvjet da je zaželjena ploča objavljena
 4. Iz stanja (B) u konačno stanje
 5. Iz stanja (C) u konačno stanje uz uvjet da je zaželjena ploča obrisana



UML dijagrami aktivnosti

Objavljivanje ploče

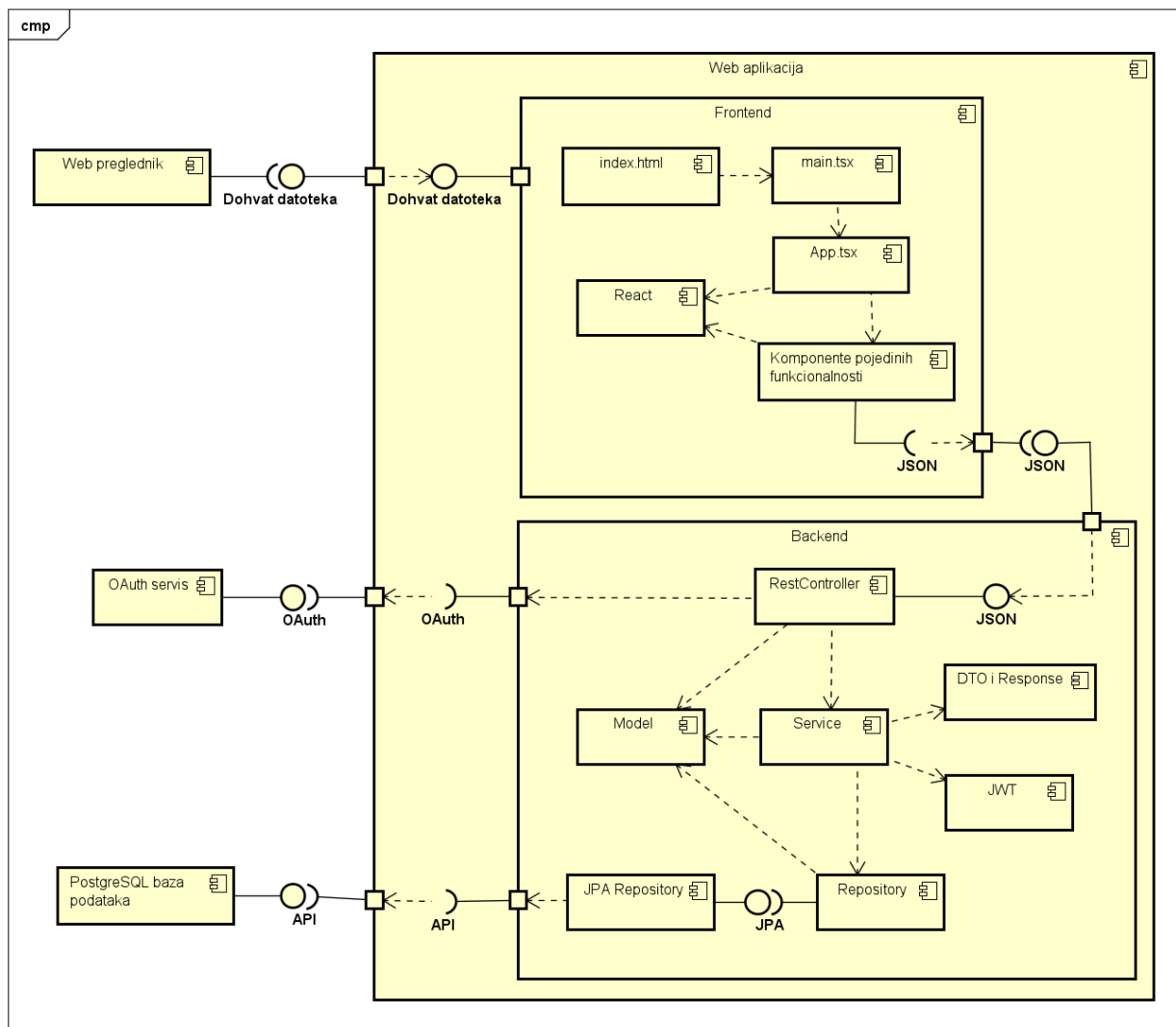
Prijedlog alternativne selekcije ploča



5. Arhitektura komponenata i razmještaja

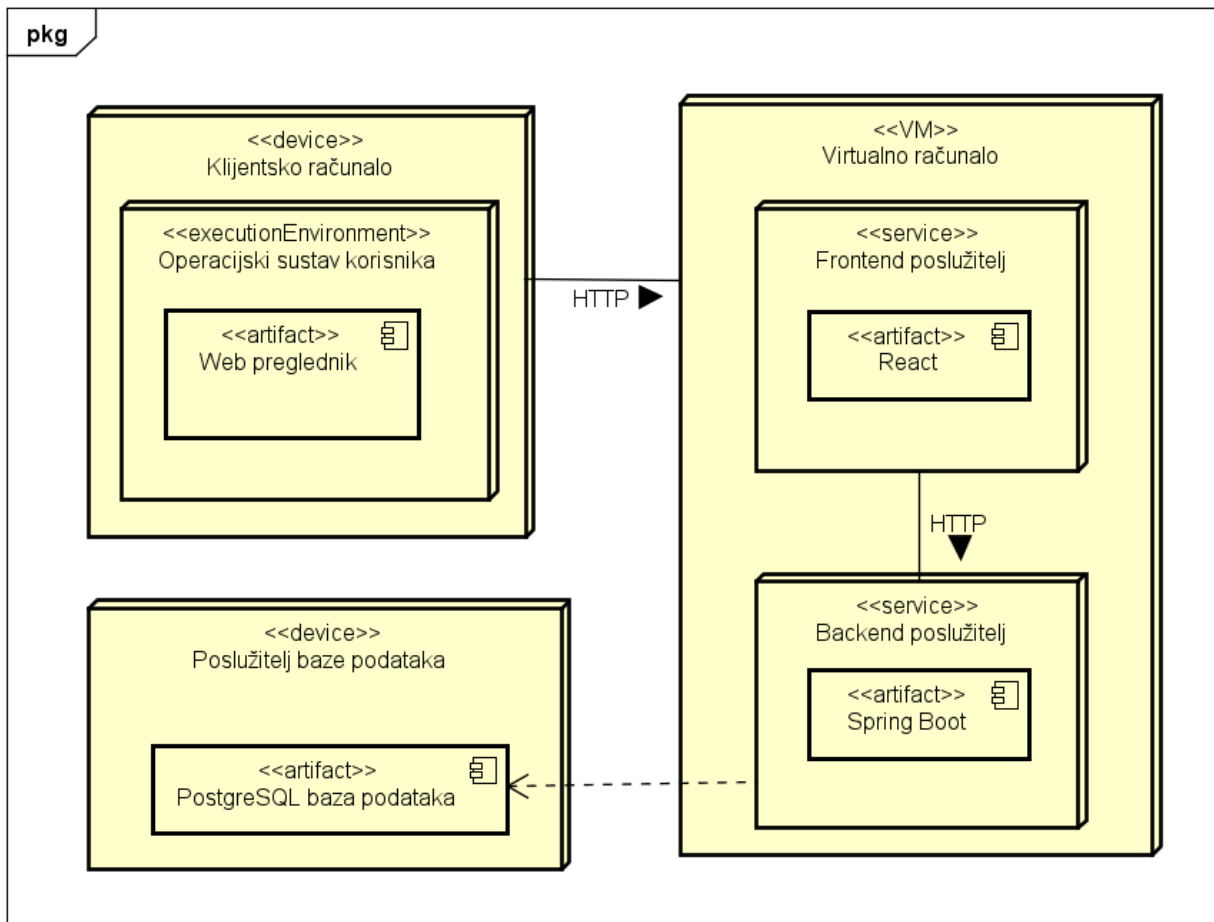
Dijagram komponenata

Dijagram komponenti prikazan u nastavku predstavlja arhitekturu aplikacije podijeljene u komponente. Svaka komponenta ima jasno definirane odgovornosti i uloge unutar sustava, omogućujući jednostavnu komunikaciju i efikasno funkcioniranje aplikacije. Prva komponenta, Web preglednik, služi kao sredstvo komuniciranja korisnika s Web aplikacijom. Druga komponenta, Web aplikacija, sačinjena je od dvije podkomponente: frontend i backend. Frontend je izgrađen pomoću React-a te uključuje komponente koje pružaju navigaciju i interaktivno iskustvo. Komunikacija između frontenda i backenda implementirana je kroz REST API, koji omogućuje razmjenu podataka u obliku JSON-a. Backend upravlja ključnim procesima: obradom funkcionalnosti unutar aplikacije, obradom zahtjeva iz frontenda te interakcijom s PostgreSQL bazom podataka putem JPA Repository sučelja. Također, sustav podržava autentifikaciju putem OAuth servisa kako bi korisnicima osigurao siguran i jednostavan pristup aplikaciji.



Dijagram razmještaja

UML dijagram razmještaja prikazuje infrastrukturni raspored sustava i komunikaciju među komponentama. Klijentsko računalo s web preglednikom služi za pristup aplikaciji. Frontend i backend pokrenuti su na Azure virtualnom računalu, pri čemu frontend (React) obrađuje korisničke interakcije i prosljeđuje zahtjeve backendu (Spring Boot) putem HTTP protokola. Backend komunicira s PostgreSQL bazom podataka na zasebnom poslužitelju, osiguravajući upravljanje i pohranu podataka unutar sustava.



6. Ispitivanje programskog rješenja

Ispitivanje komponenti

1. ispitni slučaj

Ispitana funkcionalnost: Dohvat svih ploča iz repozitorija - redovni slučaj

Ulazni podaci: Nema ulaznih podataka jer se testira samo povrat svih ploča

Očekivani rezultati: Funkcija treba vratiti popis ploča, a veličina popisa treba biti 2. Također, vinylId svakog vinila treba biti točan

Dobiveni rezultati: Test je prošao ako metoda vraća dva vinila i vinylId je točan

Postupak provođenja ispitivanja:

1. Kreirajte dva objekta klase Vinyl i postavite im vinylId vrijednosti

2. Koristite `when(repoVinyl.findAll()).thenReturn(Arrays.asList(vinyl1, vinyl2))` za simulaciju vraćanja tih ploča
3. Provjerite veličinu rezultirajuće liste i osigurajte da su `vinylId` vrijednosti ispravne

2. ispitni slučaj

Ispitana funkcionalnost: Dohvat svih ploča ako je repozitorij prazan - rubni slučaj

Ulazni podaci: Nema ulaznih podataka jer se testira samo povrat svih ploča

Očekivani rezultati: Funkcija treba vratiti prazan popis

Dobiveni rezultati: Test je prošao ako funkcija vrati prazan popis

Postupak provođenja ispitivanja:

1. Simulirajte prazan popis ploča s `when(repoVinyl.findAll()).thenReturn(Arrays.asList())`
2. Provjerite da je rezultat prazan popis

3. ispitni slučaj

Ispitana funkcionalnost: Testira reakciju `fetchVinyls` funkcionalnosti na iznimku koja se baca u slučaju pogreške u bazi podataka - bacanje iznimke

Ulazni podaci: Nema ulaznih podataka, ali simuliramo iznimku koja se baca prilikom dohvaćanja vinila

Očekivani rezultati: Očekuje se da se baci `RuntimeException` s porukom "Database error"

Dobiveni rezultati: Test je prošao ako funkcija baci iznimku `RuntimeException` s odgovarajućom porukom

Postupak provođenja ispitivanja:

1. Simulirajte iznimku s `when(repoVinyl.findAll()).thenThrow(new RuntimeException("Database error"))`
2. Provjerite da se ispravno baca iznimka

4. ispitni slučaj

Ispitana funkcionalnost: Dohvata vinila prema korisniku (`getVinylByUser`) i mapiranje `Vinyl` objekata u `VinylResponseDTO` -redovni slučaj

Ulazni podaci: Objekt `UserAccount` i popis vinila povezani s tim korisnikom

Očekivani rezultati: Funkcija treba vratiti popis `VinylResponseDTO` objekata s točnim podacima poput `vinylId`, `vinylCondition`, `coverCondition`, itd

Dobiveni rezultati: Test je prošao ako je veličina liste ispravna, a podaci unutar objekata odgovaraju onima u testiranim vinilima

Postupak provođenja ispitivanja:

1. Simulirajte repozitorij da vrati popis vinila povezane s korisnikom
2. Provjerite da su svi podaci pravilno mapirani u VinylResponseDTO objekt

5. ispitni slučaj

Ispitana funkcionalnost: Ponašanje funkcije deleteUserById kada korisnik s zadanim ID-em ne postoji u bazi - nepostojeća funkcionalnost

Ulazni podaci: ID korisnika koji ne postoji u bazi podataka

Očekivani rezultati: Očekuje se status 404 i poruka "User not found"

Dobiveni rezultati: Test je prošao ako je status 404 i tijelo odgovora je "User not found". Također, metoda deleteById() ne bi smjela biti pozvana

Postupak provođenja ispitivanja:

1. Simulirajte repozitorij tako da metoda existsById vraća false za korisnika s određenim ID-em
2. Provjerite odgovor i osigurajte da deleteById() nije pozvan

6. ispitni slučaj

Ispitana funkcionalnost: Ponašanje funkcije deleteUserById kada korisnik s zadanim ID-em postoji u bazi - redovni slučaj

Ulazni podaci: ID korisnika koji postoji u bazi podataka

Očekivani rezultati: Očekuje se status 200 i poruka "User deleted successfully"

Dobiveni rezultati: Test je prošao ako je status 200 i tijelo odgovora je "User deleted successfully". Također, metoda deleteById() treba biti pozvana

Postupak provođenja ispitivanja:

1. Simulirajte repozitorij tako da metoda existsById vraća true za korisnika s određenim ID-em
2. Provjerite odgovor i osigurajte da je metoda deleteById() pozvana

Ispitivanje sustava

1. ispitni slučaj

Ispitano ponašanje: Google login - redovni slučaj

Ulazi:

- Akcije: Pritisak gumba „Register / Login“, odabir opcije prijavljivanja putem Google forme, dabit pregleda „My Vinyls“, odabir opcije za odjavu

Koraci ispitivanja:

1. Otvoriti aplikaciju u pregledniku
2. Pritisak gumba „Register / Login“
3. Pritisak gumba „Google Sign in“
4. Verificirati rezultat prijave uspješnim odlaskom u dio „My Vinyls“
5. Pritisak gumba „My Vinyls“
6. Pritisak gumba „Logout“

Očekivani izlaz: Korisnik se uspješno prijavio putem Google login-a te dobio uvid u stranicu My Vinyls, a zatim se uspješno odjavio sa stranice.

Dobiveni izlaz:

Running 'GoogleLogin'

1. open on <https://gramofer.work.gd> OK
2. mouseOver on xpath=//button[contains(.,'Register / Login')] OK
3. click on xpath=//button[contains(.,'Register / Login')] OK
4. click on xpath=//button[contains(.,'Google Sign in')] OK
5. click on xpath=//h3[contains(.,'My Vinyls')] OK
6. click on xpath=//button[contains(.,'Logout')] OK
7. assertAlert on You have been logged out. OK

'GoogleLogin' completed successfully

2. ispitni slučaj

Ispitano ponašanje: Dodavanje ploče - redovni slučaj

Ulazi:

- Podaci: oznaka izdanja, naziv izvođača, stanje ploče, žanr, godina izdanja, naziv albuma, stanje omota, lokacija
- Akcije: odabir opcije „My Vinyls“, unos podataka, pritisak gumba „Add Vinyl“, odabir opcije „My Vinyls“

Koraci ispitivanja:

1. Otvoriti aplikaciju u pregledniku
2. Pritisak gumba „My Vinyls“
3. Unijeti sve podatke
4. Pritisak gumba „Add Vinyl“
5. Verificirati postoji li objavljena ploča odlaskom u dio „My Vinyls“
6. Pritisak gumba „My Vinyls“

Očekivani izlaz: Korisnik je uspješno dodao ploču te ju sada može vidjeti na stranici My Vinyls.

Dobiveni izlaz:

```
Running 'AddVinyl'
1. open on https://gramofer.work.gd OK
2. click on xpath=//h3[contains(.,'My Vinyls')] OK
3. mouseOver on linkText=My Vinyls OK
4. click on name=edition.editionLabel OK
5. type on name=edition.editionLabel with value TestVinyl OK
6. click on name=edition.artistName OK
7. type on name=edition.artistName with value Performer OK
8. click on name=vinylCondition OK
9. type on name=vinylCondition with value Mint OK
10. click on name=edition.genres OK
11. type on name=edition.genres with value Genre OK
12. click on name=edition.releaseDate OK
13. type on name=edition.releaseDate with value 2000 OK
14. click on name=edition.albumName OK
15. type on name=edition.albumName with value Album OK

16. click on name=coverCondition OK
17. type on name=coverCondition with value Mint OK
18. click on name=onLocation OK
19. type on name=onLocation with value UK OK
20. click on xpath=//button[contains(.,'Add Vinyl')] OK
21. assertAlert on Vinyl added successfully! OK
22. click on xpath=//h3[contains(.,'My Vinyls')] OK
23. mouseOver on linkText=My Vinyls OK
'AddVinyl' completed successfully
```

3. ispitni slučaj

Ispitano ponašanje: Dodavanje ploče koja već postoji - rubni slučaj

Ulazi:

- Podaci: oznaka izdanja, naziv izvođača, stanje ploče, žanr, godina izdanja, naziv albuma, stanje omota, lokacija
- Akcije: odabir opcije „My Vinyls“, unos podataka, pritisak gumba „Add Vinyl“

Koraci ispitivanja:

1. Otvoriti aplikaciju u pregledniku
2. Pritisak gumba „My Vinyls“
3. Unijeti sve podatke
4. Pritisak gumba „Add Vinyl“
5. Verificirati očekivane rezultate, dobivena je poruka o neuspjehu

Očekivani izlaz: Ploča nije uspješno dodana te se korisnika obavještava kako dodavanje nije uspjelo.

Dobiveni izlaz:

```
Running 'AddVinylUncorrect'
1. open on https://gramofer.work.gd OK
2. click on css=_link_contain_1rhr_77:nth-child(2) h3 OK
3. click on name=edition.editionLabel OK
4. type on name=edition.editionLabel with value TestVinyl OK
5. click on name=edition.artistName OK
6. type on name=edition.artistName with value Performer OK
7. click on name=vinylCondition OK
8. type on name=vinylCondition with value Mint OK
9. click on name=edition.genres OK
10. type on name=edition.genres with value Genre OK
11. click on name=edition.releaseDate OK
12. type on name=edition.releaseDate with value 2035 OK
13. click on name=edition.albumName OK
14. type on name=edition.albumName with value Album OK
15. click on name=coverCondition OK

16. type on name=coverCondition with value Mint OK
17. click on name=onLocation OK
18. type on name=onLocation with value US OK
19. click on css=_add_button_ijezy_124 OK
20. type on name=edition.releaseDate with value 2023 OK
21. click on css=_add_button_ijezy_124 OK
22. assertAlert on Failed to add vinyl. Please try again. OK
'AddVinylUncorrect' completed successfully
```

4. ispitni slučaj

Ispitano ponašanje: Uređivanje svojih ploča - poziv nepostojećih funkcionalnosti

Ulazi:

- Akcije: odabir opcije „My Vinyls“, pritisak gumba za uređivanje

Koraci ispitivanja:

1. Otvoriti aplikaciju u pregledniku
2. Pritisak gumba „My Vinyls“
3. Pritisak gumba za uređivanje

Očekivani izlaz: Pritiskom na gumb za uređivanje ploče nebi se ništa trebalo dogoditi.

Dobiveni izlaz:

```
Running 'EditVinyI'
1. open on / OK
2. click on xpath=/jh3[contains(.,'My VinyI')] OK
3. click on css=_exchange_buttons_u2kta_1:nth-child(12) > div:nth-child(1) > img OK
4. click on css=_exchange_buttons_u2kta_1:nth-child(12) > div:nth-child(1) > img OK
5. close OK
'EditVinyI' completed successfully
```

7. Tehnologije za implementaciju aplikacije

Korištene tehnologije i alati

Programski jezici: Korištena je Java 21 koja nudi stabilnost i dugoročnu podršku, idealna za razvoj velikih sustava, te HTML5 koji omogućava razvoj modernih, interaktivnih i dinamičnih web aplikacija s podrškom za multimediju, grafiku i napredne funkcionalnosti.

Radni okviri i biblioteke: Korišten je React 18.3.1 omogućava bolje upravljanje stanjem aplikacije i optimizaciju prikaza, SpringBoot 3.3.5. koja omogućava brzi razvoj mikroservisa i aplikacija s minimalnim konfiguracijama, te JavaScript kao osnovni jezik za razvoj dinamičkih i interaktivnih web aplikacija.

Baza podataka: Za upravljanje podacima aplikacije koristi se PostgreSQL 16.6, moderna relacijska baza podataka čija je uloga pohranjivanje, upravljanje i omogućavanje pristupa podacima sustava. PostgreSQL je odabran zbog svoje stabilnosti, brzine obrade i prilagodljivosti, što ga čini najprikladnijim za rad s usko povezanim podacima.

Razvojni alati: IntelliJ IDEA koji pruža odličnu podršku za razvoj u Javi, VS Code kao lagani i brzi editor koji nudi visoku fleksibilnost uz podršku za mnoge jezike i tehnologije te Git 2.34. koji omogućava lakše upravljanje timskim projektima.

Alati za ispitivanje: Selenium 4.0 kao popularni alat za automatizaciju testiranja web aplikacija.

Cloud platforma: Amazon AWS

8. Upute za puštanje u pogon

1. Instalacija

Preduvjeti: Vite 5.4.10., JDK 21, Apache Maven, NPM

Preuzimanje:

```
git clone https://github.com/ThomasCodex2/GramoFer.git
```

Za postavljanje backenda potrebno je otići u direktorij “.\IzvorniKod\Backend”

```
cd .\IzvorniKod\Backend
```

i pokrenuti naredbu:

```
.\mvnw.cmd clean install
```

Za postavljanje frontenda potrebno je otići u direktorij
“.\IzvorniKod\Frontend\GramoFER”

```
cd .\IzvorniKod\Frontend\GramoFER
```

i pokrenuti naredbe

```
npm install
```

```
npx vite build
```

2. Postavke

Potrebno je postaviti env varijable u sustavu.

U Windowsu pretražimo environment variables, gdje je potrebno postaviti environment variables koje se koriste u

.\IzvorniKod\Backend\src\main\resources\application.properties.

Npr:

```
spring.datasource.password=${AIVEN_PASSWORD}
```

U environmental variables sustava pod variable staviti AIVEN_PASSWORD, a pod value lozinku. Na taj način skrivamo podatke koji ne bi trebali biti javni. Potrebno je napraviti isto za svaku env varijablu koja se pojavljuje u application.properties.

Postavke baze podataka:

Potrebno je otići na <https://console.aiven.io/>, napraviti račun, stvoriti server, pritisnuti quick connect, odabrati opciju pgAdmin, kopirati danu JSON datoteku i zalijepiti ju u pgAdminu unutar izbornika import/export servers. Pritiskom na novostvoreni server potrebno je upisati lozinku koja je vidljiva u console.aiven.io nakon stvaranja servera.

3. Pokretanje aplikacije

Upute za pokretanje aplikacije u različitim okruženjima:

Razvojno okruženje:

```
npm run dev
```

Pokretanje poslužitelja:

```
npm start
```

Provjera rada: Potrebno je u preglednik unesti “http://localhost:3000/”.

4. Upute za administratore

Pristup administratorskom sučelju:

Nakon pokretanja aplikacije, i ako je korisnik koji je ulogiran u sustav admin (moguće je prilagoditi mail za korisnika koji bi trebao biti admin u `.\IzvorniKod\Backend\Service\AuthenticationService.java` unutar metode `newOAuthUser`), na početnoj stranici će se pojaviti gumb koji vodi na admin table.

Redovito održavanje:

Arhiviranje baze podataka.

Pregled logova.

Ažuriranje aplikacije.

```
git pull origin main
```

```
npm install
```

```
npm run build
```

```
npm start
```

```
.\mvnw.cmd clean install
```

Pregled unosa u bazu podataka i kontroliranje ponašanja korisnika

5. Amazon AWS platforma

Amazon AWS je popularna cloud platforma za jednostavno smještanje aplikacija.

Deploy.yaml:

```
name: Auto Deploy on Azure
```

```
on:
```

```
  push:
```

```
    branches:
```

```
      - main
```

```
jobs:
```

```
  build:
```

```
    name: Build
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Setup Maven Action
```

```
        uses: s4u/setup-maven-action@v1.16.0
```

```
        with:
```

```
          java-version: 21
```

```
      - name: Build Backend
```

```
        working-directory: ./IzvorniKod/Backend
```

```

    run: mvn clean install -Dmaven.test.skip=true
  - name: Copy Backend to remote Server
    uses: appleboy/scp-action@v0.1.7
    with:
      host: ${ secrets.VM_IP }
      username: ${ secrets.VM_USERNAME }
      key: ${ secrets.SSH_PRIVATE_KEY }
      port: 22
      source: "Izvornikod/Backend/target/gramofer*.jar"
      target: ~/GramoFer_Deploy_Backend/
  - name: Setup Node.js environment
    uses: actions/setup-node@v4.1.0
    with:
      node-version: '18.20.5'
  - name: Build Frontend
    working-directory: ./Izvornikod/Frontend/GramoFER
    run: |
      npm install
      npx vite build
  - name: Copy Frontend to remote Server
    uses: appleboy/scp-action@v0.1.7
    with:
      host: ${ secrets.VM_IP }
      username: ${ secrets.VM_USERNAME }
      key: ${ secrets.SSH_PRIVATE_KEY }
      port: 22
      source: "Izvornikod/Frontend/GramoFER/dist/"
      target: ~/GramoFer_Deploy_Frontend/
deploy:
  needs: build
  name: Deploy
  permissions:
    deployments: write
  runs-on: ubuntu-latest
  steps:
    - name: Checkout Repository
      uses: 'actions/checkout@v4'
      with:
        ref: main
        token: ${ github.token }
    - name: Create GitHub deployment
      uses: chrnorm/deployment-action@v2
      id: deployment
      with:
        token: '${ github.token }'
        environment-url: ${ vars.MY_APP }
        environment: production
    - name: SSH to VPS and Deploy
      uses: appleboy/ssh-action@master
      env:

```

```

    GITHUB_TOKEN: ${GITHUB_TOKEN}
  with:
    host: ${VM_IP}
    username: ${VM_USERNAME}
    key: ${SSH_PRIVATE_KEY}
    port: 22
    script: |
      export GOOGLE_CLIENT_ID=${GOOGLE_CLIENT_ID}
      export GOOGLE_CLIENT_SECRET=${GOOGLE_CLIENT_SECRET}
      export AIVEN_PASSWORD=${AIVEN_PASSWORD}
      export GOOGLE_EMAIL=${GOOGLE_EMAIL}
      export GOOGLE_EMAIL_PASSWORD=${GOOGLE_EMAIL_PASSWORD}
      cd GramoFer_Deploy_Backend/
      kill $(cat ./backend.pid) || true
      rm current.jar || true
      cp Izvornikod/Backend/target/gramofer*.jar current.jar
      nohup java -jar current.jar > /dev/null 2>&1 & echo $! >
./backend.pid
      cd ..
      sudo rm -rf /var/www/html/* ; sudo cp -r
./GramoFer_Deploy_Frontend/Izvornikod/Frontend/GramoFER/* /var/www/html/
- name: Update deployment Status (success)
  if: success()
  uses: chrnorm/deployment-status@v2
  with:
    token: '${GITHUB_TOKEN}'
    environment-url: ${MY_APP}
    state: 'success'
    deployment-id: ${steps.deployment.outputs.deployment_id}
- name: Update deployment status (failure)
  if: failure()
  uses: chrnorm/deployment-status@v2
  with:
    token: '${GITHUB_TOKEN}'
    environment-url: ${MY_APP}
    state: 'failure'
    deployment-id: ${steps.deployment.outputs.deployment_id}

```

Deploy.yaml se nalazi u .github\workflows\deploy.yaml i potrebno je postaviti environmental varijable u GitHub secrets.

```

script: |
  export GOOGLE_CLIENT_ID=${GOOGLE_CLIENT_ID}
  export GOOGLE_CLIENT_SECRET=${GOOGLE_CLIENT_SECRET}
  export AIVEN_PASSWORD=${AIVEN_PASSWORD}
  export GOOGLE_EMAIL=${GOOGLE_EMAIL}
  export GOOGLE_EMAIL_PASSWORD=${GOOGLE_EMAIL_PASSWORD}

```

Ovdje je prikazana konfiguracija za exportanje env varijabli iz secreta GitHuba.

Deployment i build su podijeljeni u dva dijela i sve se izvodi na VM-u koji ima svoj ip, username i generirani SSH password

Postavljanje na Amazon AWS.

Prijavite se na Amazon.

Kreirajte novi Web Service i povežite ga s vašim GitHub repozitorijem.

Konfigurirajte postavke (npr. build i deploy naredbne).

Pokretanje aplikacije: Amazon AWS će automatski preuzeti repozitorij, instalirati ovisnosti i pokrenuti aplikaciju. Nakon deploja, aplikaciji možete pristupiti putem generiranog URL-a (npr. <https://gramofer.work.gd/>).

Opis pristupa aplikaciji na javnom poslužitelju Potrebno je ulogirati se u aplikaciju, moguća je registracija i login putem forme ili prijava s OAuth protokolom. Na stranici Offers se nalaze sve ploče koje su dostupne u sustavu. Na stranici MyVinyls se nalaze sve ploče korisnika i moguće je dodati nove ploče. U MyExchanges se nalaze svi podaci vezani uz zamjene, a u MyWishes je moguće dodati novu želju i pregledati sve želje koje smo stvorili. Na početnoj stranici je nakon klika na ploču moguće ponuditi zamjenu i nakon toga odabrati naše ploče koje smo spremni zamjeniti.

9. Zaključak i budući rad

Svaki član tima GramoFer tijekom zadnja tri mjeseca imao je priliku raditi s modernim web tehnologijama, uključujući razvoj korisničkog i poslužiteljskog dijela sustava, te aktivno sudjelovati u stvaranju i dokumentiranju svih ključnih faza projekta. Osim tehničkih izazova, važno je bilo razviti sposobnost jasne i precizne komunikacije kako bi se osigurala učinkovita suradnja unutar tima, dok su redovni sastanci uživo i komunikacija putem WhatsAppa i Discorda omogućili brzu razmjenu informacija.

Stečena znanja i izazovi

Kroz izradu projekta upoznali smo se s konceptima i procesom korištenja biblioteke React i softverskog okvira Spring Boot, načinom podjele zadataka i vođenja tima, postupkom iterativnog poboljšavanja našeg proizvoda te dokumentiranjem svakog koraka i napretka. Najvažniji element timskog rada, kao i najizazovniji, bila je komunikacija. Ona je bila ključna u ublažavanju prvobitnog problema: odabira, učenja i privikavanja na nove alate i način izrade proizvoda u grupi, ali i na rad s nama dotad nepoznatim ljudima. S obzirom na različite vještine, znanja te općenitu količinu rada koji je grupa trebala obaviti, trebalo je podijeliti zadatke u grupi kako bi razvoj tekao glatko. Tu je ključnu ulogu odigrao voditelj tima koji je dodijelio prvotne „blueprint“ uloge u kojima su se članovi tima kasnije po potrebi nadopunjavali. Podjela je većinom bila na rad na frontendu, backendu, bazi podataka i dokumentaciji, ali članovi tima su kroz dijeljenje savjeta i materijala, aktivnu diskusiju i predlaganje ideja na kraju u svakoj komponenti sudjelovali u nekoj mjeri.

Prilikom izrade dokumentacije smo također shvatili koliko je važno pokriti sve važne elemente kako bi ostali developeri mogli ne samo pratiti naš tok misli i motivaciju za projekt, već i razumjeti donesene odluke, prepoznate izazove s kojima smo se suočili te učinkovitije doprinijeti daljnjem razvoju. Neiskustvo, izazovi u komunikaciji, kao i izazovi u postavljanju jasnih prioriteta tijekom implementacije funkcionalnosti, dovodili su do usmjeravanja naših asistenata koji su pažljivo pratili naš napredak i osiguravali da ostanemo na pravom putu kroz cijeli tijek projekta.

Budući rad

Kao što je već istaknuto, komunikacija se pokazala najvažnijom značajkom izrade projekta u umjereno velikoj grupi. U svrhu bolje koordinacije i efikasnijeg napretka, u budućnosti bi bilo korisno unaprijed savladati alate koji će biti potrebni za izradu projekta. Boljim međusobnim upoznavanjem članova tima svakako bi se povećao ritam rada, stekla veća otvorenost za suradnju, te ubrzao proces donošenja odluka. Nadalje, preuzimanje zadataka u slučaju neiskustva ili nemogućnosti ispunjavanja vremenskog roka treba biti dogovoreno na vrijeme te s punim povjerenjem u ostale članove tima. U sklopu izrade ovog projektnog zadatka, smatramo da bi tim mogao nastaviti raditi na njegovom unapređenju i proširenju. Neke od budućih stavki ove aplikacije mogle bi uključivati korisničku podršku, funkciju izravne komunikacije korisnika i mogućnost personalizacije funkcionalnosti korisničkim potrebama.

A. Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <https://www.fer.unizg.hr/predmet/proinz>
2. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>

B. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Postavljanje početne stranice, opisa zadatka i specifikacija projekta	Tomislav Peraković	23.10.2024.
0.1.1	Prepravci u specifikacijama projekta	Tomislav Peraković	25.10.2024.
0.2	Dodani obrasci uporabe, dijagram obrazaca uporabe	Josip Rončević	27.10.2024.
0.2.1	Ažurirana struktura predloška, dodani podaci na početnoj stranici	Sami Perković	12.11.2024.
0.2.2	Dodani obrasci uporabe	Josip Rončević, Jurica Galić	13.11.2024.
0.3	Dodani obrasci uporabe, prepravci	Josip Rončević, Kristijan Kozomara	14.11.2024.

Rev.	Opis promjene/dodatka	Autori	Datum
0.4	Dodane specifikacije, tablice i dijagram baze podataka	Sami Perković	14.11.2024.
0.4.1	Ažuriran dnevnik promjena dokumentacije	Sami Perković	14.11.2024.
0.4.2	Preuređivanje funkcionalnih zahtjeva	Josip Rončević	14.11.2024.
0.4.3	Pisanje dnevnika sastanaka	Tomislav Peraković	14.11.2024.
0.4.4	Dodani obrasci uporabe	Josip Rončević	14.11.2024.
0.4.5	Dodani dijagrami razreda, prepravak u poglavlju Arhitektura i dizajn sustava	Josip Rončević	14.11.2024.
0.5	Prepravak analize zahtjeva, dodano u poglavlju Arhitektura i dizajn sustava	Tomislav Peraković	15.11.2024.
0.5.1	Premještena uvodna rečenica	Sami Perković	15.11.2024.
0.5.2	Dodavanje opisa projektnog zadatka	Sami Perković	15.11.2024.
0.5.3	Dodani dijagrami obrazaca	Josip Rončević	15.11.2024.
0.5.4	Dodan obrazac uporabe, prepravak u analizi zahtjeva	Josip Rončević	15.11.2024.
0.5.5	Prepravci u poglavlju Arhitektura i dizajn sustava	Amar Omeragić, Lovro Kasunić	15.11.2024.
0.6	Dodani zahtjevi za održavanje	Josip Rončević	15.11.2024.
0.7	Dodani funkcionalni zahtjevi, uređivanje analize zahtjeva	Tomislav Peraković	15.11.2024.
0.7.1	Dodan sekvencijski dijagram	Josip Rončević	15.11.2024.
0.7.2	Ažuriran dnevnik promjena dokumentacije	Sami Perković	15.11.2024.
0.7.3	Prepravci u poglavlju Specifikacija zahtjeva sustava	Josip Rončević	15.11.2024.
0.7.4	Dodavanje tablice aktivnosti	Tomislav Peraković	15.11.2024.
0.7.5	Ažuriran dnevnik promjena dokumentacije	Sami Perković	15.11.2024.
1.0.1	Ažuriran opis projektnog zadatka i baza podataka, ažuriran raspored dokumentacije	Sami Perković	12.1.2024.
1.1	Ažuriran opis projektnog zadatka, dodan dijagram stanja	Sami Perković	13.1.2024.
1.2	Dodan dijagram stanja i dijagram aktivnosti	Sami Perković	14.1.2024.
1.3	Dodan dijagram aktivnosti, ažurirana baza podataka	Sami Perković	15.1.2024.
1.4	Dodan dijagram komponenti	Sami Perković	23.1.2024.
1.5	Dodan dijagram razmještaja	Sami Perković	24.1.2024.
1.6	Napisan zaključak	Josip Rončević	24.1.2024.

C. Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

Datum: 16.10.2024.

Trajanje: 1.5h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, T.Peraković

Teme sastanaka:

Upoznavanje s kolegama

Gruba podjela posla i preference u zadacima projekta

2. sastanak

Datum: 18.10.2024.

Trajanje: 2h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, J.Rončević, J.Galić, K.Kozomara, T.Peraković

Teme sastanaka:

Biranje voditelja

Mogućnost o predlaganju proizvoljne teme

Definitivna podjela zadataka u projektu

Uspostava GitHuba

3. sastanak

Datum: 25.10.2024.

Trajanje: 1h

Prisustvovali:

J.Galić, K.Kozomara, S.Perković, T.Peraković

Teme sastanaka:

Prezentiranje arhitekture projekta

4. sastanak

Datum: 9.11.2024.

Trajanje: 2h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, J.Galić, T.Peraković

Teme sastanaka:

Uspostava baze podataka

5. sastanak

Datum: 11.11.2024.

Trajanje: 1h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković

Teme sastanaka:

Testiranje dostupnosti baze podataka

6. sastanak

Datum: 12.11.2024.

Trajanje: 2h

Prisustvovali:

J.Rončević, L.Kasunić, S.Perković, T.Peraković

Teme sastanaka:

Rad na backendu, povezivanje s bazom podataka

Nadopunjavanje dokumentacije

7. sastanak

Datum: 13.11.2024.

Trajanje: 2h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, T.Peraković, J.Rončević, K.Kozomara, J.Galić

Teme sastanaka:

Frontend development

Backend development

Deployment na Azure VM

8. sastanak

Datum: 3.1.2025.

Trajanje: 1.5h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, T.Peraković, J.Rončević, K.Kozomara, J.Galić

Teme sastanaka:

Planiranje daljnjeg razvoja projekta

9. sastanak

Datum: 10.1.2025.

Trajanje: 2h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković

Teme sastanaka:

Revizija baze podataka, ponovna podjela poslova

10. sastanak

Datum: 11.1.2025.

Trajanje: 2h

Prisustvovali:

T.Peraković, K.Kozomara, J.Galić

Teme sastanaka:

Zajednički rad na frontendu

11. sastanak

Datum: 17.1.2025.

Trajanje: 2h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, T.Peraković, K.Kozomara, J.Galić

Teme sastanaka:

Zajednički rad na frontendu i bekendu

11. sastanak

Datum: 18.1.2025.

Trajanje: 2h

Prisustvovali:

S.Perković, J.Rončević

Teme sastanaka:

Revizija podjele rada na dokumentaciji

12. sastanak

Datum: 19.1.2025.

Trajanje: 2.5h

Prisustvovali:

A.Omeragić, L.Kasunić, T.Peraković, K.Kozomara, J.Galić

Teme sastanaka:

Zajednički rad na frontendu i bekendu

13. sastanak

Datum: 23.1.2025.

Trajanje: 2h

Prisustvovali:

A.Omeragić, L.Kasunić, S.Perković, T.Peraković, K.Kozomara, J.Galić

Teme sastanaka:

Zajednički rad na frontendu i bekendu, rad na dokumentaciji i prezentaciji

Tablica aktivnosti

Kategorija	Amar Omeragić	Kristijan Kozomara	Josip Rončević	Lovro Kasunić	Sami Perković	Jurica Galić	Tomislav Peraković
Upravljanje projektom	-	-	-	-	-	-	4
Opis projektnog zadatka	-	-	-	-	5	-	2
Funkcionalni zahtjevi	-	-	0,5	-	-	-	-
Opis pojedinih obrazaca	-	1	6	-	-	1	-
Dijagram obrazaca	-	-	4	-	-	-	-
Sekvencijski dijagrami	-	-	-	-	-	-	-
Opis ostalih zahtjeva	-	-	1	-	-	-	-
Arhitektura i dizajn sustava	1	-	-	1,5	1,5	-	2
Baza podataka	-	-	-	-	6	-	-
Dijagram razreda	1	-	-	-	-	-	-
Dijagram stanja	-	-	-	-	3,5	-	-
Dijagram aktivnosti	-	-	-	-	2,5	-	-
Dijagram komponenti	-	-	-	-	2	-	-
Korištene tehnologije i alati	-	-	-	-	0,5	-	0,5
Ispitivanje programskog rješenja	6	-	-	-	-	-	-

Kategorija	Amar Omeragić	Kristijan Kozomara	Josip Rončević	Lovro Kasunić	Sami Perković	Jurica Galić	Tomislav Peraković
Dijagram razmještaja	-	-	-	-	1	-	-
Upute za puštanje u pogon	-	-	-	-	-	-	4
Dnevnik sastajanja	-	-	-	-	-	-	2
Zaključak i budući rad	-	-	2	-	-	-	-
Popis literature	-	-	-	-	-	-	0,5
Istraživanje informacija/tehnologija	11,5	10	7,5	12	8,5	9	9
Baza podataka	-	-	-	-	8	-	-
Backend	14,5	-	-	16,5	4	-	10
Frontend	-	9,5	-	-	-	18	10
Izgled aplikacije	-	10	-	-	-	-	-
Puštanje u pogon	-	-	-	-	-	-	6

Dijagram pregleda promjena

Commits over time

Weekly from Oct 20, 2024 to Jan 19, 2025

