

Programsko inženjerstvo

Ak. god. 2020./2021.

dogGO

Dokumentacija, Rev. 2

Grupa: *Exception*

Voditelj: *Eva Jagodić*

Datum predaje: 14.1.2021.

Nastavnik: *Doria Bukić*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
2.1	Motivacija i osnovna ideja	5
2.2	Detaljan opis rješenja	6
2.2.1	Općenito	6
2.2.2	Vrste korisnika	6
2.3	Opis funkcionalnosti aplikacije	8
2.3.1	Pregled pasa / udruga	8
2.3.2	Rezervacija i potvrda termina šetnje	8
2.3.3	Pregled statistika	8
2.4	Implementacijski detalji	9
2.5	Pregled konkurentskih rješenja i stanje tržišta	9
3	Specifikacija programske potpore	12
3.1	Funkcionalni zahtjevi	12
3.1.1	Obrasci uporabe	14
3.1.2	Sekvencijski dijagrami	25
3.2	Ostali zahtjevi	28
4	Arhitektura i dizajn sustava	29
4.1	Baza podataka	31
4.1.1	Opis tablica	32
4.1.2	Dijagram baze podataka	36
4.2	Dijagram razreda	37
4.3	Dijagram stanja	42
4.4	Dijagram aktivnosti	43
4.5	Dijagram komponenti	51
5	Implementacija i korisničko sučelje	53
5.1	Korištene tehnologije i alati	53

5.1.1	Backend tehnologije	53
5.1.2	Frontend tehnologije	53
5.1.3	Puštanje u pogon	54
5.1.4	Dokumentacija	54
5.1.5	Timska komunikacija	54
5.2	Ispitivanje programskog rješenja	55
5.2.1	Ispitivanje komponenti	55
5.2.2	Ispitivanje sustava	59
5.3	Dijagram razmještaja	62
5.4	Upute za puštanje u pogon	63
5.4.1	Pokretanje baze	63
5.4.2	Pokretanje backenda	65
5.4.3	Pokretanje frontenda	67
6	Zaključak i budući rad	68
	Popis literature	70
	Indeks slika i dijagrama	72
	Dodatak: Prikaz aktivnosti grupe	73

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Jagodić	15.10.2020.
0.2	Dodan opis baze podataka.	Miletić	26.10.2020.
0.3	Dodan opis projektnog zadatka.	Smoljan	28.10.2020.
0.4	Dodani funkcionalni i ostali zahtjevi te obrasci uporabe.	Pezo, Sentinella- Jerbić	29.10.2020.
0.4.1	Popravljeni funkcionalni zahtjevi.	Sentinella- Jerbić	30.10.2020.
0.5	Dodan uvod u arhitekturu i dizajn sustava.	Sentinella- Jerbić	6.11.2020.
0.6	Dodani dijagrami obrazaca uporabe	Jagodić	9.11.2020.
0.7	Dodani sekvencijski dijagrami	Golub, Miletić	10.11.2020.
0.8	Dodana prva verzija dijagrama razreda.	Sentinella- Jerbić	11.11.2020.
0.8.1	Popravci dijagrama razreda.	Golub, Sentinella- Jerbić	11.11.2020.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Jagodić	13.11.2020.
1.1	Dodane korištene tehnologije i alati	Sentinella- Jerbić	12.1.2021.
1.2	Dodani dijagrami stanja	Pezo	12.1.2021.
1.3	Dodani dijagrami aktivnosti i razmještaja	Terović	12.1.2021.
1.4	Ispitivanje programskog rješenja	Golub, Miletić	14.1.2021.
1.5	Zaključak i budući rad	Jagodić	14.1.2021.
1.6	Dodan dijagram komponenti	Smoljan	14.1.2021
1.7	Ažurirani dijagrami razreda	Sentinella- Jerbić	14.1.2021
1.8	Dodane upute za puštanje u pogon	Smoljan	14.1.2021

Rev.	Opis promjene/dodatka	Autori	Datum
2.0	Konačni tekst predloška dokumentacije	Jagodić	14.1.2021.

2. Opis projektnog zadatka

2.1 Motivacija i osnovna ideja

Broj napuštenih pasa u skloništima u Hrvatskoj, pa i ostalim europskim zemljama, svaki dan je sve veći i veći. Kao posljedica toga, psi u prosjeku provode sve više vremena u skloništu čekajući svog novog vlasnika. S druge strane, postoje brojni ljudi koji vole pse i žele im pomoći, ali nisu u mogućnosti udomiti psa, bilo zbog manjka prostora ili nedostatka vremena.

Jedan od najboljih načina na koji mogu pomoći, a da ne udomi psa, je šetnjama. Tako se psima olakšava čekanje i smanjuje vrijeme provedeno u skloništu, te ga se ispunjava na koristan način – fizičkom aktivnošću i socijalizacijom. Ljudima se pruža prilika da uživaju u druženju sa psima, a možda da upoznaju i svog novog najboljeg prijatelja.

Kako je većina ljudi vrlo zaposlena ideja je stvoriti web aplikaciju koja bi na jednostavan i brz način spojila udruge za zaštitu životinja i zainteresirane građane-šetače pasa. Olakšavanjem postupka same prijave za šetnju i kreiranjem atraktivne aplikacije koja je lagana za korištenje, smatramo da bi se znatno povećao broj zainteresiranih građana.

Aplikacija je namijenjena dvjema skupinama krajnjih korisnika: građanima - potencijalnim šetačima pasa i udrugama za zaštitu životinja.

Prijavom u sustav udruge unose podatke o psima koje žele uključiti u aktivnost. Građani zatim mogu pregledati sve udruge i pse raspoložive za šetnju, te odabrati psa i rezervirati termin šetnje. Odabrani termin vidljiv je udruzi koja u dogovoreno vrijeme predaje psa šetaču. Aplikacija također vodi statistiku o svim provedenim šetnjama te ju, uz korisnikovu suglasnost, prikazuje javno u obliku rang liste najaktivnijih šetača.

2.2 Detaljan opis rješenja

2.2.1 Općenito

Glavna namjena aplikacije je spajanje ljubitelja životinja s udrugama kojima je njihova pomoć, u vidu šetnji, potrebna. Aplikacija bi bila realizirana kao web aplikacija, uključujući i prilagodbu za prikaz na mobilnim uređajima. Bila bi dostupna na području države, te potencijalno korištena od strane većeg broja korisnika (pregledi zainteresiranih građana).

2.2.2 Vrste korisnika

Korisnik je svaka osoba, ne nužno registrirana, koja u bilo kojem trenutku pregledava ili koristi neki dio aplikacije. Razlikujemo dva tipa korisnika: šetače i udruge.

1) Građani - šetači

Pod šetače spadaju svi korisnici koji aplikaciju upotrebljavaju s ciljem pronalaska psa za šetnju. Razlikujemo neregistrirane korisnike (javni posjetitelji) te registrirane korisnike.

a) Neregistrirani korisnici (javni posjetitelji)

Jedan od glavnih ciljeva aplikacije je zainteresirati što veći broj ljudi za šetanje pasa. Stoga i neregistrirani korisnici imaju mogućnost pregleda profila svih udruga, profila njihovih pasa koji sudjeluju u programu, pregled statistika šetnje svih pasa unutar neke udruge te lokaciju same udruge. Statistika šetnji također nudi informacije o psima koji su manje šetani od ostalih te koji imaju veću potrebu za šetnjom. Korisniku se nudi i opcija prijave za šetanje pasa, a ako korisnik odabere tu opciju, korisnika se obavještava da kako bi napravio registraciju treba biti prijavljen u sustav.

b) Registrirani korisnici

Kako bi se korisnik registrirao za šetača, mora navesti:

- ime
- prezime
- adresu e-pošte
- željeno korisničko ime
- lozinku

Nakon što korisnik obavi registraciju, ima opciju pregleda udruga i pasa raspoloživih za šetnju, isto kao i neregistrirani korisnik. Međutim, kao registriranom korisniku pružaju mu se i mogućnosti odabira psa za šetnju, odabir željenog termina te prijava za samu šetnju. Nakon odabira termina šetnje, bilo javne ili grupne, korisniku se termin šetnje prikazuje u kalendaru, a svoj raspored može preuzeti i u PDF obliku. Također, svi registrirani korisnici imaju opciju pregleda vlastitog profila, uključujući nadolazeće i prethodne šetnje, statistike, osobne podatke i slično. Mogu označiti vlastite statistike šetnji kao javne, u kojem slučaju se one mogu pojaviti na javnoj rang listi. Imaju i opciju uređivanja svojih podataka, te brisanja računa na stranici.

2) Udruge

Kako bi se obavila registracija, udruga mora navesti:

- ime i prezime osobe koja predstavlja udrugu
- adresu e-pošte
- naziv udruge
- OIB udruge
- željeno korisničko ime
- lozinku

Nakon obavljene prijave, udruga može započeti sa stvaranjem profila za pse u svojem skrbništvu.

Za stvaranje profila jednog psa, udruga navodi sljedeće informacije:

- ime
- slika (neobavezno)
- opis psa
- preferirani način šetnje
 - individualna šetnja / šetnja s drugim psima
- vrsta

Ti profili, zajedno sa osnovnim informacijama o udruzi te statistikama šetnji svih njenih pasa, javno su dostupni svim korisnicima web aplikacije. Udruga u bilo kojem trenutku može obrisati neki od profila pasa, u kojem slučaju se otkazuju i svi budući termini šetnji na koje je prijavljen.

Udruga također ima pristup svim rezervacijama za svoje pse, koje imaju opciju otkazati.

Udruga također ima opciju pregleda svojih podataka i uređivanja istih, kao i brisanja svog profila iz aplikacije. U tom slučaju se automatski brišu i profili svih pasa registriranih preko te udruge.

2.3 Opis funkcionalnosti aplikacije

2.3.1 Pregled pasa / udruga

Ovo je jedna od glavnih funkcionalnost aplikacije. I registriranim i neregistriranim korisnicima omogućen je prikaz svih pasa koji sudjeluju u programu, te je omogućeno i filtriranje pasa prema lokaciji, udruzi, preferiranom načinu šetnje itd. Moguć je i pregled profila svih udruga, te pojedinih pasa o kojima neka udruga skrbi (a koji sudjeluju u programu) .

Profil psa sadrži kratke informacije o psu, kao i njegovu sliku te termine u kalendaru i točno vrijeme kada je pas dostupan za šetnju.

2.3.2 Rezervacija i potvrda termina šetnje

Nakon što korisnik pronađe psa kojeg želi šetati, sljedeći korak je rezervacija termina šetnje. Korisnik prvo pregledava termine u kalendaru kada je pas dostupan, te odabire termin koji mu odgovara. Ukoliko je korisnik odabrao psa koji voli grupne šetnje, tada korisnik u istom terminu može odabrati još jednog psa koji voli grupne šetnje, pod uvjetom da su iz iste udruge.

2.3.3 Pregled statistika

Postojio nekoliko vrsta statistike koje aplikacija vodi:

1. Statistike šetača pasa

- uključuju broj šetnji, broj prošetanih pasa te ukupnu duljinu šetnji
- po defaultu privatna statistika, korisnik ju sam može učiniti javnom, u kojem slučaju se prikazuje i na naslovnoj stranici, u sklopu liste najaktivnijih šetača

2. Statistike udruge

- uključuje broj pasa u skrbi udruge te ukupan broj šetnji obavljen sa psima udruge
- statistika javno dostupna svim korisnicima

3. Statistike admina

- ukupan broj aktivnih korisnika stranice
- ukupan broj aktivnih udruga na stranici
- ukupan broj pasa uključenih u program stranice

2.4 Implementacijski detalji

Kako je jedan od glavnih ciljeva aplikacije privući što više korisnika u program, aplikacija mora podržavati istovremeni rad većeg broja korisnika. Za implementaciju backend dijela aplikacije odabran je programski jezik Java, uz pomoć frameworka Spring, dok je za frontend korišten React.

Aplikacija se krajnim korisnicima prikazuje u obliku web-stranice, prilagođene za prikaz na mobilnim uređajima kao i na osobnim računalima.

2.5 Pregled konkurentskih rješenja i stanje tržišta

Najbliže konkurentske aplikacije su aplikacije za plaćeno šetanje pasa, poput BarkleyPets (<https://barklypets.com/>), WagWalking(<https://wagwalking.com/>), DogHero (<https://play.google.com/store/apps/details?id=br.com.doghero.astro>) i slično. Iako ove aplikacije predstavljaju određenu konkurenciju, u smislu da ciljaju istu skupinu korisnika kao i naša aplikacija (ljude koje vole pse i žele provoditi vrijeme s njima), naglasak naše aplikacije je na volontiranju, a ne na plaćenom šetanju, čime efektivno ipak korisnicima nudimo drugačiju vrstu usluge nego navedene aplikacije.

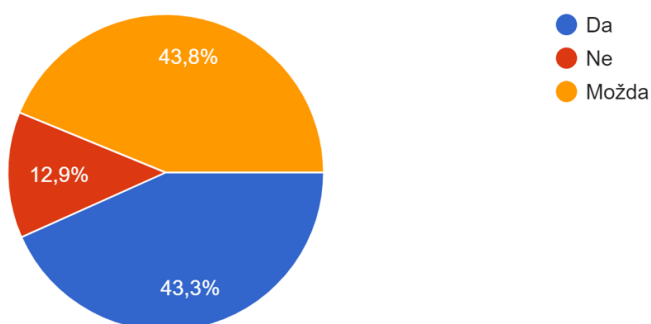
Pod ostalu konkurenciju možemo svrstati forme za prijavu za volontiranje na stranicama raznih skloništa, poput <https://www.bluecross.org.uk/volunteer>, <https://helpingpaws.net/how-to-volunteer/>. Međutim, prijava preko formi na tim

stranicama samo je prvi od mnogo koraka koje potencijalni volonter mora proći prije nego što mu se pruži prilika izvesti psa u šetnju. Naša aplikacija je i osmišljena da olakšava i ubrza postupak prijave za volontiranje, te eliminira potrebu za službenom prijavom poput navedenih.

Također, bitno je naglasiti da navedene konkurentne aplikacije nisu bazirane na hrvatskom tržištu po čemu se razlikuje naša aplikacija. Iz tog razloga, smatramo da naša aplikacija može pokriti dosad relativno neiskorišten dio tržišta, i uz pravi marketing, privući velik broj korisnika.

Dodatan korak koji smo odlučili napraviti da bismo dobili osjećaj o stvarnom interesu građana za ovakvu vrstu aplikacije, bilo je provođenje male verzije istrage tržišta. Koristeći alate koje nudi Google Forms, izradili smo javnu anketu i podijelili je među ljudima takozvanom *chain message* metodom. Naime, poznanicima smo poslali anketu na ispunjavanje s prijateljskim prijedlogom da ju i oni podijele sa svojim poznanicima. Jasno je da takva statistika nije nepristrana, ali ipak daje dozu uvida u potražnju tržišta i dobra je polazna točka za ozbiljnija istraživanja u budućnosti. Na slici 2.1 moguće je vidjeti rezultate odgovora na pitanje o interesu građana.

Jesi li zainteresiran/a za korištenje ovakve aplikacije?
194 odgovora



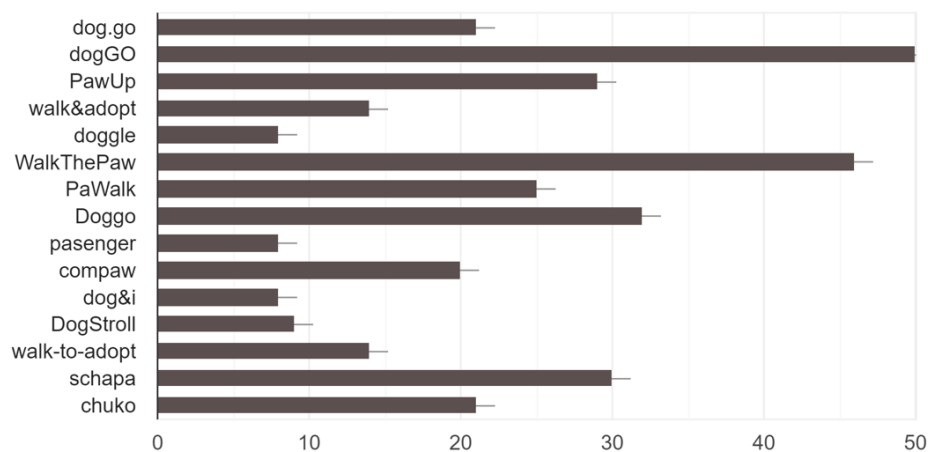
Slika 2.1: Rezultat anketnog pitanja o interesu za aplikaciju

Osim pitanja o interesu, naša je anketa služila i odabiru imena aplikacije. Vjerujemo da iza svake dobre aplikacije stoji pamtljivo ime koje ju dobro predstavlja.

Uzevši prijedloge svih članova tima složili smo anketno pitanje o odabiru najboljeg i uključili ga u javnu anketu. Rezultat tog dijela ankete jasan je po naslovu dokumenta, a ostale prijedloge i detalje o rezultatu moguće je vidjeti na slici 2.2.

Koje ime za aplikaciju ti zvuči najbolje?

194 odgovora



Slika 2.2: Rezultat anketnog pitanja o imenu aplikacije

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Udruge za životinje
2. Građani
 - (a) registrirani korisnici
 - (b) javni posjetitelji
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) pregledati listu svih registriranih udruga
 - (b) pregledati svaku pojedinu registriranu udruhu i dobiti prikaz općih informacija (ime udruge, slika, OIB, e-mail, adresa)
 - (c) pregledati pse pojedine udruge i dobiti prikaz općih informacija (ime, slika, opis, vrsta šetnje, pasmina, lokacija, raspoloživost)
 - (d) pogledati rang listu registriranih šetača s javnim profilom prema različitim kriterijima:
 - i. broj odrađenih šetnji
 - ii. broj različitih šetanih pasa
 - iii. duljina odrađenih šetnji
 - (e) registrirati se u sustav kao šetač - stvoriti novi korisnički račun za koji su potrebni ime, prezime, korisničko ime, lozinka i e-mail adresa
 - (f) registrirati se u sustav kao udruha - stvoriti novi korisnički račun za koji su potrebni ime, prezime, ime udruge, OIB udruge, lozinka, e-mail
 - (g) prijaviti se već izrađenim korisničkim računom šetača ili udruge u sustav

2. Udruga (inicijator) može:

- (a) kreirati i uređivati profile pasa za koji su potrebni ime, slika, opis, vrsta šetnje (individualna ili skupna), lokacija i raspoloživost
- (b) pregledati prošle i nadolazeće rezervacije šetnji za vlastite pse te urediti njihov status
- (c) uređivati i izbrisati vlastiti korisnički račun

3. Šetač (inicijator) može:

- (a) napraviti, pregledati i urediti status rezervacije psa za šetnju
- (b) pogledati i preuzeti raspored vlastitih šetnji
- (c) pogledati vlastitu statistiku obavljenih šetnji
- (d) omogućiti da vlastita statistika bude javna
- (e) uređivati i izbrisati vlastiti korisnički račun

4. Administrator (inicijator) može:

- (a) pregledati sve rezervacije te urediti njihov status
- (b) pregledati, uređivati i obrisati sve profile šetača, udruga i pasa

5. Baza podataka (sudionik) može:

- (a) pohranjuje sve podatke o registriranim šetačima i udrugama
- (b) pohranjuje sve podatke o psima i njihovoj raspoloživosti
- (c) pohranjuje sve podatke o ugovorenim rezervacijama

3.1.1 Obrasci uporabe

UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun šetača ili udruge
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju šetača ili udruge
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je unio neispravan e-mail, podatke u nedozvoljenom formatu ili odabrao već zauzeto korisničko ime i/ili e-mail
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC2 - Prijava

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik registriran u sustav, ali ne i prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik unosi korisničko ime i lozinku
 2. Sustav provjerava ispravnost unesenih podataka
 3. Korisnik dobiva pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Korisnik unosi neispravno korisničko ime ili lozinku
 1. Sustav obavještava korisnika o neuspjeloj prijavi i vraća ga na stranicu za prijavu

UC3 - Pregled osobnih podataka šetača

- **Glavni sudionik:** Šetač
- **Cilj:** Pregledati osobne podatke

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao šetač
- **Opis osnovnog tijeka:**
 1. Šetač odabire odgovarajuću opciju "Moj profil"
 2. Aplikacija prikazuje osobne podatke korisnika

UC4 - Pregled podataka vlastite udruge

- **Glavni sudionik:** Udruga
- **Cilj:** Pregledati podatke o vlastitoj udruzi
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao udruga
- **Opis osnovnog tijeka:**
 1. Udruga odabire opciju "Moja udruga"
 2. Aplikacija prikazuje podatke o udruzi i psima udruge

UC5 - Promjena osobnih podataka

- **Glavni sudionik:** Šetač
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao šetač
- **Opis mogućih odstupanja:**
 1. Šetač pri pregledu osobnih podataka odabire opciju za promjenu osobnih podataka
 2. Šetač unosi nove podatke
 3. Šetač sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 3.a Šetač promijeni podatke, ali novi podatci imaju nedozvoljene vrijednosti
 1. Sustav obavještava šetača da su unesene nedozvoljene vrijednosti

UC6 - Promjena podataka udruge

- **Glavni sudionik:** Udruga
- **Cilj:** Promijeniti podatke o udruzi
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao udruga
- **Opis osnovnog tijeka:**

1. Udruga pri pregledu osobnih podataka odabire opciju za promjenu podataka udruge
 2. Udruga unosi nove podatke
 3. Udruga sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 3.a Udruga promijeni podatke, ali novi podatci imaju nedozvoljene vrijednosti
 1. Sustav obavještava udruhu da su unesene nedozvoljene vrijednosti

UC7 - Brisanje korisničkog računa

- **Glavni sudionik:** Šetač
- **Cilj:** Izbrisati korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Šetač odabire opciju "Moj profil"
 2. Šetač odabire opciju za brisanje računa
 3. Šetač šalje upit je li korisnik siguran
 4. Šetač potvrđuje da je siguran
 5. Otvara se stranica za registraciju
- **Opis mogućih odstupanja:**
 - 4.a Šetač odabire da nije siguran
 1. Sustav vraća šetača na stranicu s osobnim podacima

UC8 - Brisanje računa udruge

- **Glavni sudionik:** Udruga
- **Cilj:** Izbrisati račun udruge
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Udruga odabire opciju "Moja udruga"
 2. Udruga odabire opciju za brisanje računa
 3. Udruga šalje upit je li korisnik siguran
 4. Udruga potvrđuje da je siguran
 5. Baza podataka označava da je šetač obrisao

6. Otvara se stranica za registraciju

- **Opis mogućih odstupanja:**

4.a Udruga odabire da nije sigurna

1. Sustav vraća korisnika na stranicu s podacima udruge

UC9 - Pregled udruga

- **Glavni sudionik:** Neregistrirani korisnik, šetač, udruga
- **Cilj:** Pregledati registrirane udruge i dobiti prikaz općih informacija
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju “Pregledaj udruge”
 2. Aplikacija prikazuje sve udruge upisane u bazu podataka i osnovne informacije

UC10 - Pregled detalja udruge

- **Glavni sudionik:** Neregistrirani korisnik, šetač, udruga
- **Cilj:** Pregled detaljnijih informacija o odabranoj udruzi
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
 1. Korisnik na listi svih udruga odabire udruhu koju želi pregledati
 2. Aplikacija prikazuje detaljne podatke o odabranoj udruzi i sve njene pse

UC11 - Pregled detalja psa

- **Glavni sudionik:** Neregistrirani korisnik, šetač, udruga
- **Cilj:** Pregled detaljnijih informacija o odabranom psu
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
 1. Korisnik se nalazi na profilu udruge
 2. Aplikacija prikazuje detaljne podatke o odabranom psu i nudi opciju rezervacije

UC12 - Rezervacija šetnje

- **Glavni sudionik:** Šetač

- **Cilj:** Rezervirati termin šetnje odabranog psa
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
 1. Šetač odabire opciju rezervacije za odabranog psa
 2. Aplikacija prikazuje raspoloživost psa, nudi opciju rezervacije termina i opciju grupne šetnje
 3. Šetač odabire slobodan termin za individualnu šetnju psa
 4. Šetač potvrđuje svoj odabir
 5. Šetač dobiva potvrdu o rezervaciji
- **Opis mogućih odstupanja:**
 - 2.a Korisnik još nije registriran u sustav
 1. Sustav vodi korisnika na stranicu registracije
 - 3.a Šetač odabire opciju grupne šetnje
 1. Aplikacija prikazuje ostale dostupne pse iz iste udruge u odabranom terminu
 2. Šetač odabire željene pse za dodati u rezervaciju šetnje
 3. Šetač potvrđuje svoj odabir
 4. Šetač dobiva potvrdu o rezervaciji

UC13 - Pregled kalendara

- **Glavni sudionik:** Šetač
- **Cilj:** Pregledati vlastiti raspored ugovorenih šetnji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao šetač
- **Opis osnovnog tijeka:**
 1. Šetač se nalazi na kartici Moj profil
 2. Aplikacija prikazuje sve ugovorene šetnje korisnika i opciju preuzimanja rasporeda u pdf formatu

UC14 - Preuzimanje rasporeda

- **Glavni sudionik:** Šetač
- **Cilj:** Preuzeti vlastiti raspored ugovorenih šetnji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao šetač
- **Opis osnovnog tijeka:**

1. Šetač pri pregledu vlastitog kalendara odabire opciju preuzimanja rasporeda u pdf formatu
2. Aplikacija prikazuje opcije vremenskog razdoblja sadržanog u pdf rasporedu
3. Šetač odabire željenu opciju
4. Na šetačevo računalo preuzima se raspored odabranog vremenskog razdoblja

UC15 - Brisanje rezervacije od strane šetača

- **Glavni sudionik:** Šetač
- **Cilj:** Obrisati prethodno ugovorenu šetnju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik ima ugovorenu barem jednu rezervaciju
- **Opis osnovnog tijeka:**
 1. Šetač pri pregledu ugovorenih šetnji odabire odgovarajuću rezervaciju i odabire opciju brisanja
 2. Sustav šalje upit je li korisnik siguran
 3. Šetač potvrđuje da je siguran
 4. Aplikacija ponovno prikazuje stranicu sa svim ugovorenim šetnjama
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odabire da nije siguran
 1. Sustav vraća korisnika na stranicu "Moj profil"
 - 4.a Brisanje nije uspješno obavljeno
 1. Aplikacija prikazuje povratnu informaciju o grešci prilikom brisanja

UC16 - Pregled svih rezerviranih šetnji udruge

- **Glavni sudionik:** Udruga
- **Cilj:** Pregledati sve rezervacije pasa vlastite udruge
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao udruga
- **Opis osnovnog tijeka:**
 1. Udruga odabire opciju pregleda rezervacija
 2. Aplikacija prikazuje sve buduće i prošle ugovorene šetnje za pse vlastite udruge

UC17 - Brisanje rezervacije od strane udruge

- **Glavni sudionik:** Udruga
- **Cilj:** Obrisati prethodno ugovorenu šetnju
- **Sudionici:** Baza podataka
- **Preduvjet:** Pas udruge ima ugovorenu barem jednu rezervaciju
- **Opis osnovnog tijeka:**
 1. Udruga pri pregledu svih ugovorenih šetnji odabire odgovarajuću rezervaciju i odabire opciju brisanja
 2. Sustav šalje upit je li udruga sigurna
 3. Udruga potvrđuje da je sigurna
 4. Aplikacija ponovno prikazuje stranicu sa svim ugovorenim šetnjama
- **Opis mogućih odstupanja:**
 - 3.a Udruga odabire da nije sigurna
 1. Sustav vraća korisnika na stranicu "Moj profil"
 - 4.a Brisanje nije uspješno obavljeno
 1. Aplikacija prikazuje povratnu informaciju o grešci prilikom brisanja

UC18 - Pregled pasa vlastite udruge

- **Glavni sudionik:** Udruga
- **Cilj:** Pregledati sve informacije o određenom psu iz vlastite udruge
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao udruga
- **Opis osnovnog tijeka:**
 1. Udruga pri pregledu podataka o udruzi odabire opciju pregleda željenog psa
 2. Aplikacija prikazuje sve informacije o psu i odgovarajuće šetnje

UC19 - Dodavanje psa u udugu

- **Glavni sudionik:** Udruga
- **Cilj:** Dodati novog psa u aplikaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao udruga
- **Opis osnovnog tijeka:**
 1. Udruga pri pregledu podataka o udruzi odabire opciju za dodavanje psa
 2. Aplikacija otvara formu za upis informacija o psu
 3. Udruga sprema upisane podatke

4. Novi podaci pohranjuju se u bazu podataka
 5. Udruga dobiva potvrdu o uspješnom izvršenju
 6. Aplikacija prikazuje sve informacije o psu i odgovarajuće šetnje
- **Opis mogućih odstupanja:**
 - 3.a Udruga unese podatke o psu, ali ne spremi upis
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

UC20 - Brisanje psa

- **Glavni sudionik:** Udruga
- **Cilj:** Obrisati psa iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao udruga
- **Opis osnovnog tijeka:**
 1. Udruga pri pregledu svih pasa odabire opciju pregleda željenog psa
 2. Udruga odabire opciju brisanja psa iz aplikacije
 3. Sustav šalje upit korisniku je li siguran
 4. Udruga potvrđuje da je siguran
 5. Sustav šalje korisnika natrag na stranicu sa psima
- **Opis mogućih odstupanja:**
 - 4.a Udruga odabire da nije siguran
 1. Sustav vraća korisnika na stranicu sa psima

UC21 - Pregled rang liste

- **Glavni sudionik:** Neregistrirani korisnik, šetač, udruga
- **Cilj:** Pregledati rang listu svih javnih šetača
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Šetači"
 2. Aplikacija prikazuje rang listu svih šetača prema zadanom kriteriju

UC22 - Pregled svih šetača i udruga

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati sve šetače i udruge u sustavu
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen kao administrator
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za pregled svih korisnika sustava
 2. Aplikacija prikazuje listu svih šetača i svih udruga

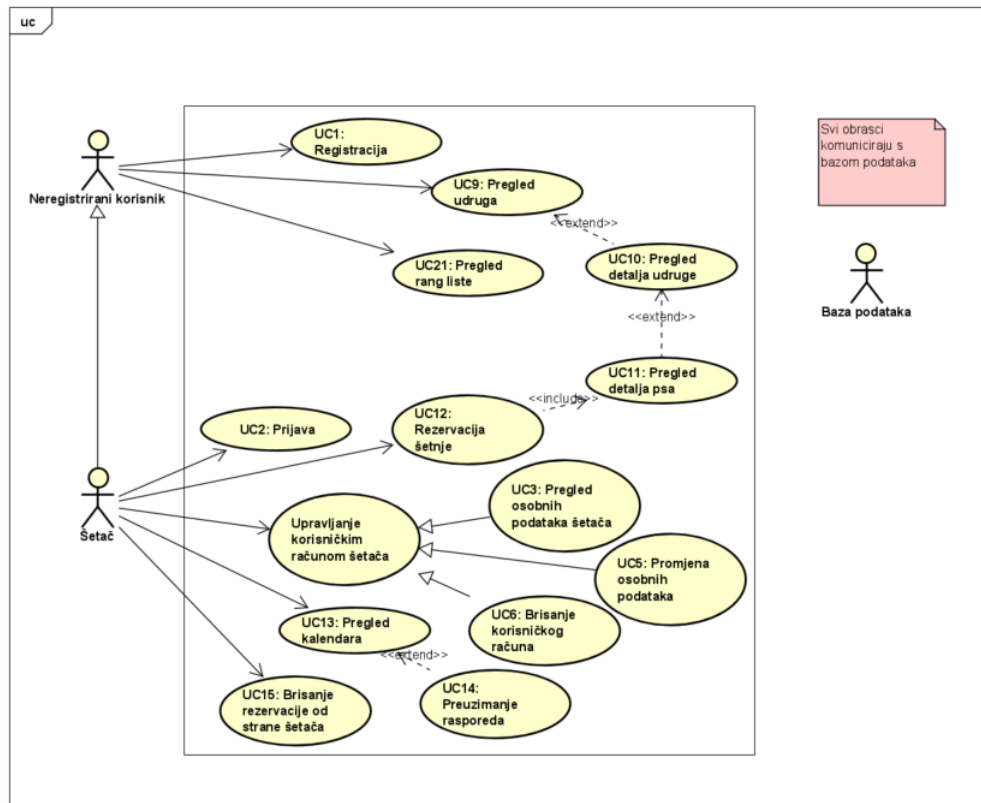
UC23 - Pregled svih rezervacija svih udruga

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati sve rezervacije svih udruga
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao administrator
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za pregled svih rezervacija
 2. Aplikacija prikazuje listu svih rezervacija po udrugama

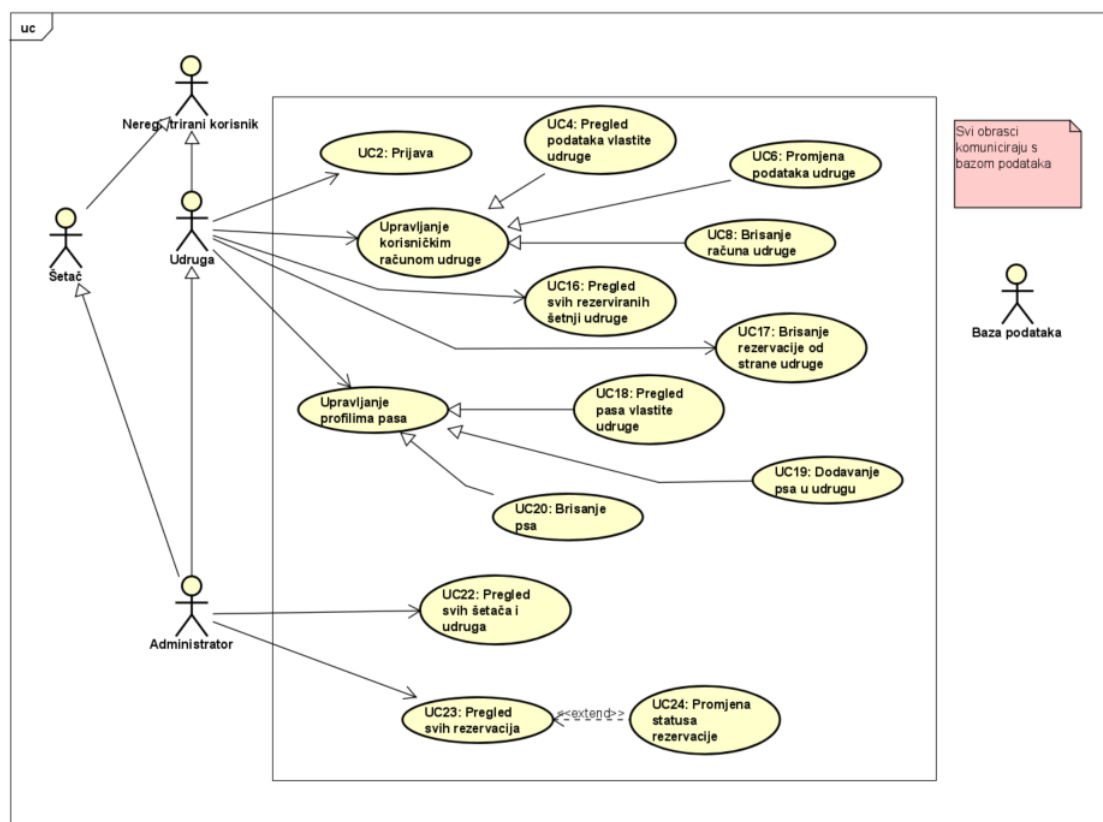
UC24 - Promjena statusa rezervacije

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti status rezervacije (aktivna, otkazana)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao administrator
- **Opis osnovnog tijeka:**
 1. Administrator odabire rezervaciju čiji status želi promijeniti
 2. Administrator mijenja status rezervacije

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost neregistriranog korisnika i šetača

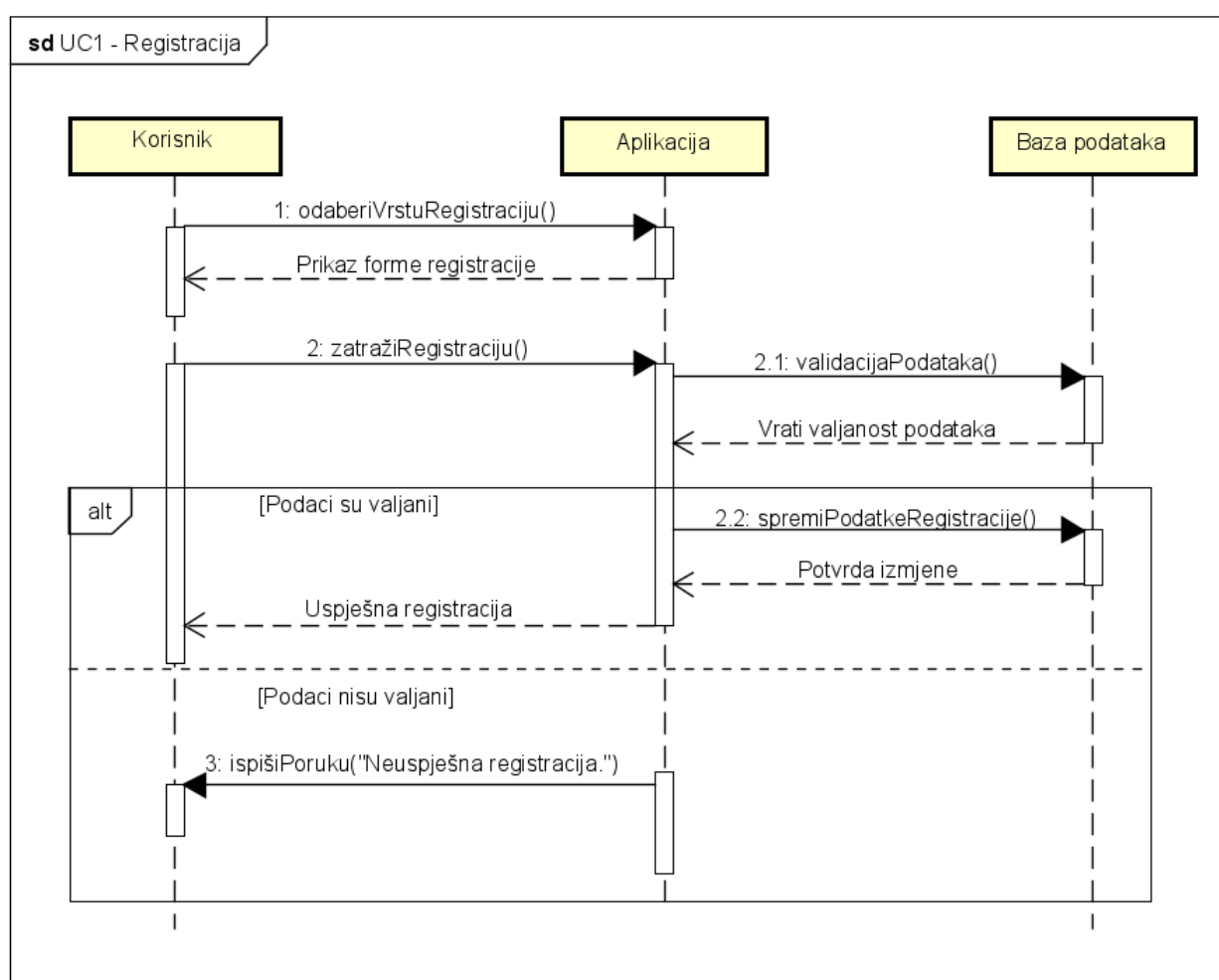


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost udruge i administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC1 - Registracija

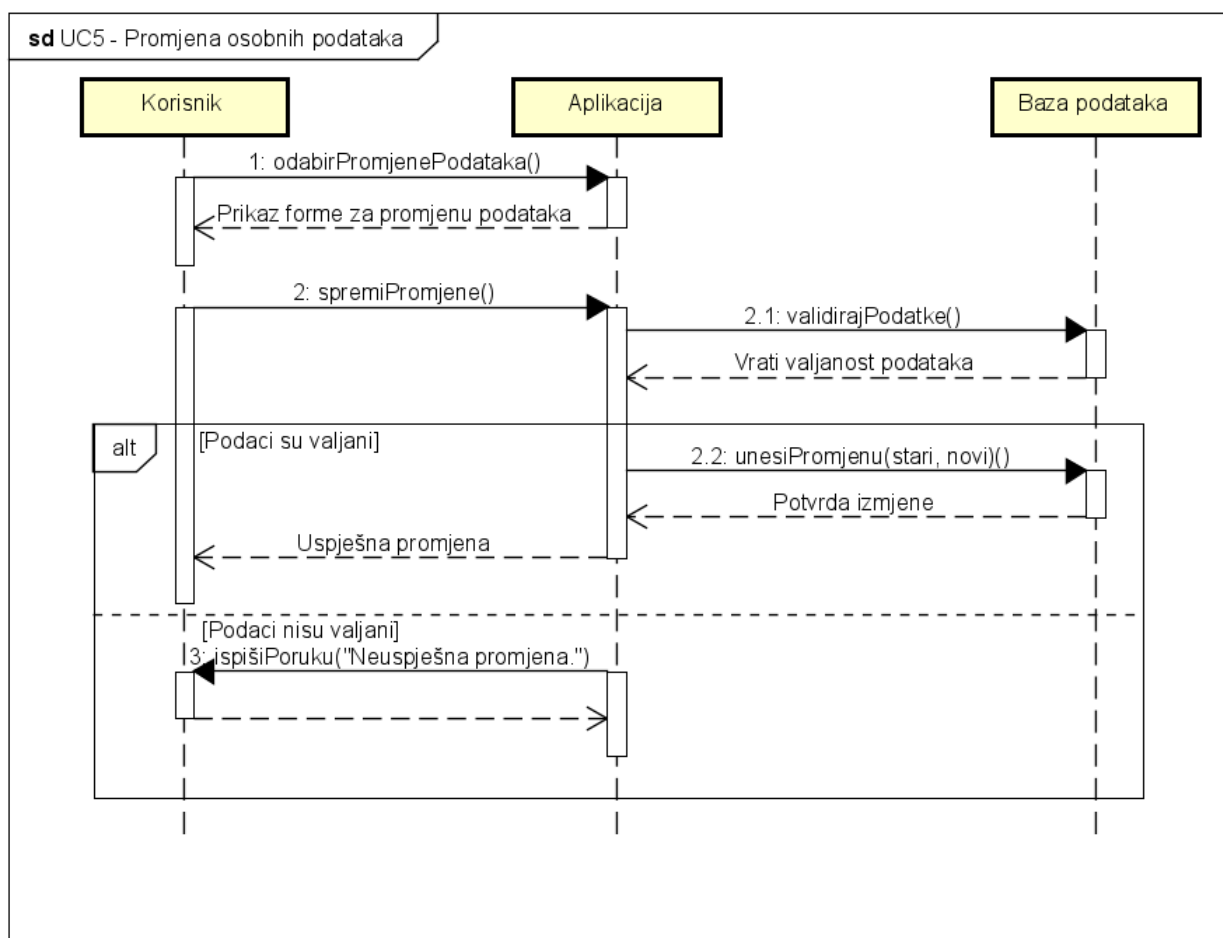
Korisnik bira opciju za registraciju šetača ili udruge te šalje zahtjev za registraciju. Poslužitelj prikazuje formu za odabranu registraciju. Korisnik unosi potrebne korisničke podatke. Poslužitelj provjerava ispravnost unesenih podataka i ako su valjani, u bazu podataka sprema podatke novog korisnika i obavještava ga o uspješnoj registraciji. Ako su elektronička pošta, korisničko ime ili formati nedozvoljeni, korisnika se obavještava o neuspješnoj registraciji i vraća na stranicu za registraciju.



Slika 3.3: Sekvencijski dijagram za registraciju korisnika

Obrazac uporabe UC5 - Promjena osobnih podataka

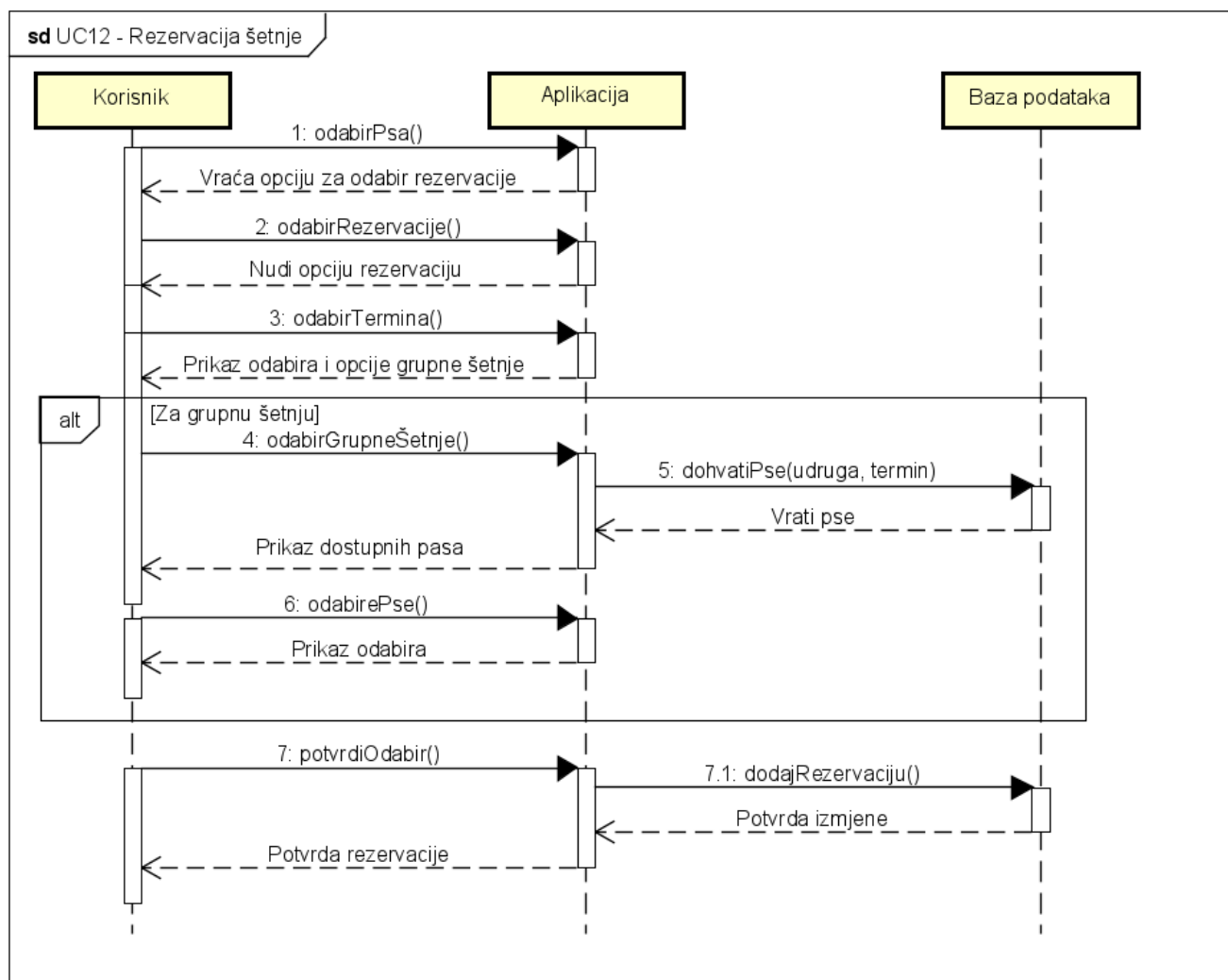
Korisnik bira opciju “Promijeni osobne podatke”. Poslužitelj vraća formu za unos novih podataka. Korisnik potom unosi nove podatke i potvrđuje promjenu. Poslužitelj provjerava valjanost novih podataka preko baze podataka. Ako su podaci valjani, poslužitelj šalje podatke korisnika u bazu podataka kako bi se stare vrijednosti ažurirale. Baza podataka vraća potvrdu poslužitelju koja šalje poruku o uspješnoj promjeni. U suprotnom, korisnika se obavještava o neuspješnoj promjeni podataka.



Slika 3.4: Sekvencijski dijagram za promjenu osobnih podataka korisnika

Obrazac uporabe UC12 - Rezervacija šetnje

Korisnik bira opciju “Odaberi psa.”. Ako korisnik nije registriran, sustav navodi korisnika na stranicu za registraciju. Poslužitelj korisniku vraća opciju za rezervaciju pasa. Korisnik bira određenog psa na što poslužitelj prikazuje njegovu raspoloživost te nudi opciju rezervacije. Korisnik bira termin na što mu poslužitelj vraća prikaz odabira i opciju grupne šetnje. Korisnik potvrđuje svoj odabir ako se odluči za individualnu šetnju. Odabirom opcije grupne šetnje poslužitelj iz baze dohvaća dostupne pse iz iste udruge u odabranom terminu i prikazuje korisniku. Korisnik bira pse za grupnu šetnju i potvrđuje svoj odabir, nakon čega poslužitelj dodaje novu rezervaciju u bazu koja vraća poruku potvrde i obavještava korisnika o uspješnoj rezervaciji.



Slika 3.5: Sekvencijski dijagram za rezerviranje šetnji

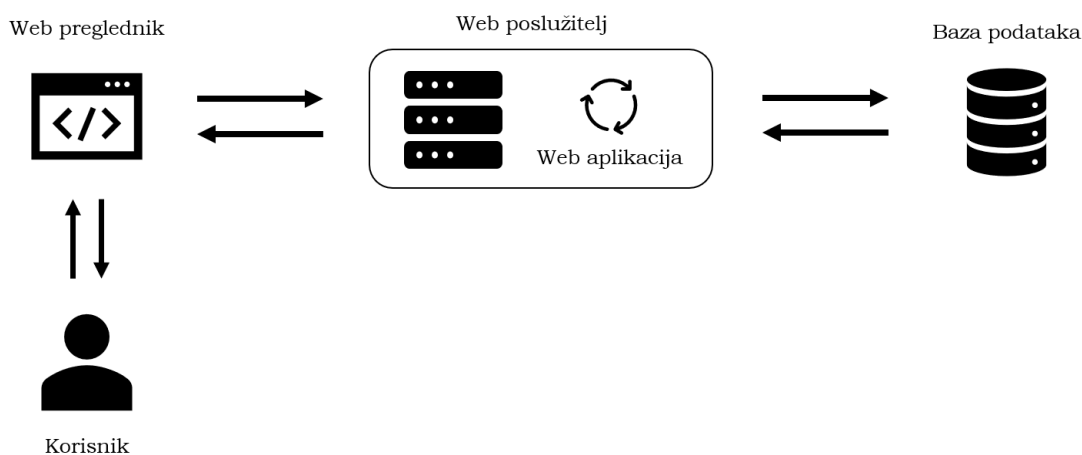
3.2 Ostali zahtjevi

- Sustav treba biti implementiran kao web-aplikacija koristeći objektno orijentiranu paradigmu
- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Sustav treba podržavati funkcionalnosti neovisno o vrsti uređaja, preglednika ili veličini ekrana
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati dulje od nekoliko sekundi
- Korisnički podaci trebaju biti sigurno pohranjeni i odgovarajuće enkriptirani
- Pristup sustavu mora biti omogućen iz javne mreže
- Korisničko sučelje treba biti jednostavno, intuitivno i pregledno, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Sustav treba koristiti odgovarajući europski format datuma (*DD.MM.YYYY.*)

4. Arhitektura i dizajn sustava

Ključne komponente za ostvarenje našeg sustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Komunikacija krajnjeg korisnika s našim sustavom započinje u web pregledniku pristupom iz javne mreže preko odgovarajuće poveznice. Konceptualnu shemu navedenog sustava moguće je pratiti na slici 4.1.

Web preglednik (engl. *web browser*) program je koji korisniku omogućuje pronalaženje i pregled podataka na Internetu u obliku web stranica. Pruža grafičko sučelje i prikaz multimedijalnih sadržaja dohvaćenih s poslužitelja te prati korisničke akcije nad sučeljem kako bi generirao odgovarajuće zahtjeve i delegirao posao prema dijelu sustava koji ih obrađuje. Korišteni tip zahtjeva jest HTTP (engl. *Hyper Text Transfer Protocol*).

Web poslužitelj (engl. *web server*) prima zahtjeve web preglednika. U našoj implementaciji pristupa mu se pomoću REST (engl. *Representational State Transfer*) us-

luga koje koriste JSON format. Pokreće komunikaciju s web aplikacijom i prosljeđuje joj korisničke zahtjeve.

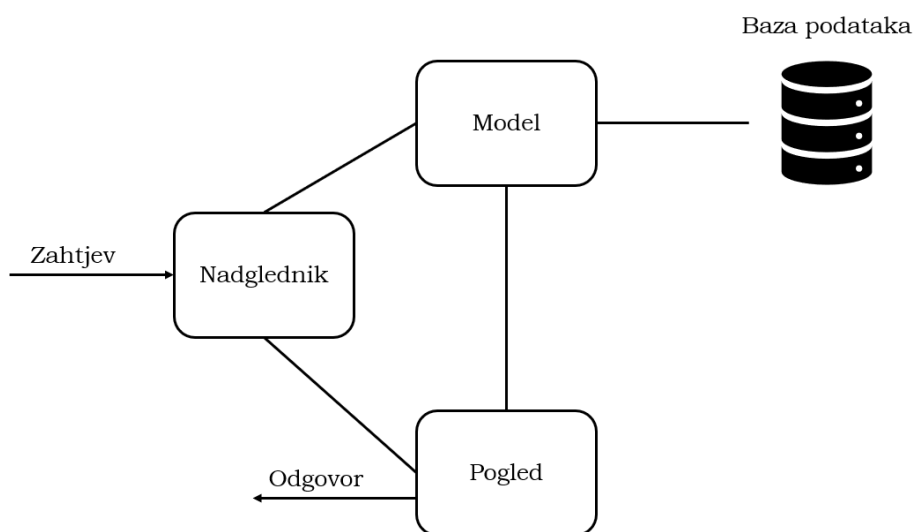
Web aplikacija (engl. *web app*) je ključni dio sustava koji obrađuje korisničke zahtjeve. Obrada uključuje određenu komunikaciju s bazom podataka poput dohvata, uređivanja ili brisanja. Nakon što se odgovarajuće procesira zahtjev, šalje se odgovor poslužitelju koji se ultimativno prikazuje korisniku u web pregledniku.

Naš se produkt konceptualno temelji na troslojnoj arhitekturi kojom se najčešće koriste poslovne aplikacije i sastoji se od sljedećih slojeva:

1. **Prezentacijski sloj** - korisničko sučelje aplikacije koje korisniku predstavlja značajke i podatke aplikacije
2. **Servisni sloj** - sadrži poslovnu logiku koja pokreće osnovne funkcionalnosti aplikacije poput donošenja odluka, izračuna, procjena i obrade podataka koji prolaze između druga dva sloja
3. **Podatkovni sloj** - odgovoran za interakciju s bazama podataka radi spremanja i vraćanja podataka aplikacije

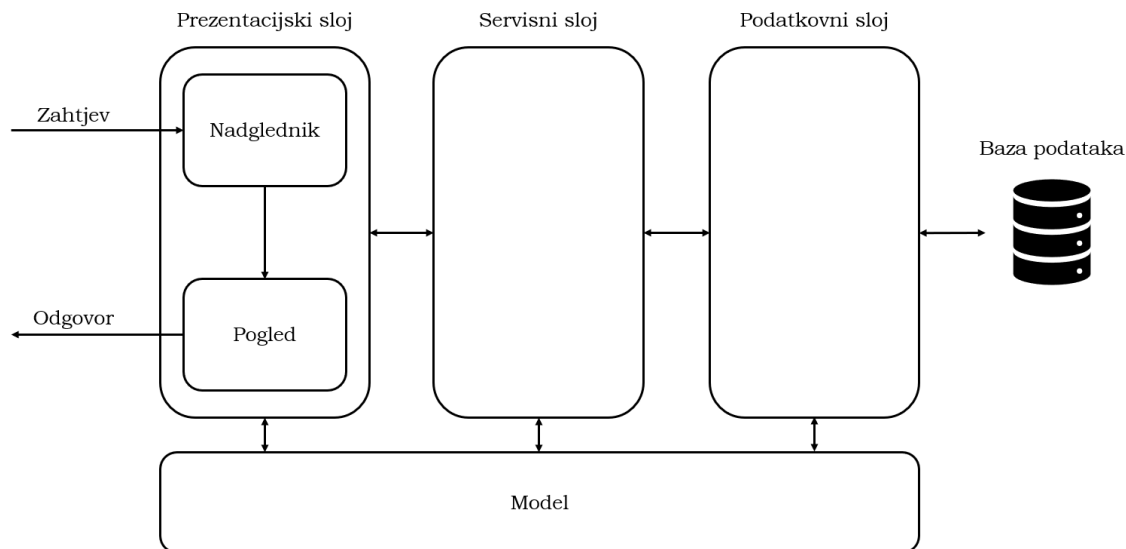
Specifičnije, korišteni Spring razvojni okvir koristi se i MVC arhitekturom (eng. *Model-View-Controller*). To podrazumijeva sljedeću podjelu uloga vidljivoj i na slici 4.2.

- **Model** - dohvat i manipulacija podacima
- **Pogled** - prezentacija dostavljenih podataka
- **Nadglednik** - prima zahtjeve te upravlja s pogledom i modelom



Slika 4.2: MVC arhitektura

U prošlosti se linearnost troslojnog modela smatrala poželjnom, no razvojem web tehnologija "trokutasti" MVC model dobio je na popularnosti pa danas mnogi sustavi koriste hibridnu formu navedenih arhitekturnih oblika pa tako i Spring razvojni okvir.



Slika 4.3: Konceptualno hibridna arhitektura Spring razvojnog okvira

4.1 Baza podataka

Kako bi olakšali modeliranje stvarnog svijeta korištena je relacijska baza podataka. Modelirana je relacijama, odnosno tablicama koje imaju svoje ime i attribute. Tablice su različito povezane preko zajedničkih atributa. Baza podataka služi za brzu i jednostavnu pohranu podataka, izmjenu, ažuriranje te dohvat podataka za daljnju obradu. Ova aplikacija koristi bazu podataka sa sljedećim entitetima:

- RegisteredUser
- RegisteredUserLogin
- Association
- AssociationLogin
- Dog
- WalkJournal
- AssociationLocation
- DogAvailability
- RankingList
- OtherReservations

4.1.1 Opis tablica

Walker Entitet sadrži sve bitne informacije o registriranom korisniku aplikacije. Njegovi atributi su: ID korisnika, ime i prezime, elektronička pošta i broj mobitela korisnika, javnost podataka i obrisan (atribut koji postaje true ako korisnik obriše svoj račun). Entitet RegisteredUser je u *Many-to-One* vezi s entitetom RankingList preko jedinstvenog identifikatora korisnika i u *One-to-Many* vezi s entitetom WalkJournal preko jedinstvenog identifikatora korisnika.

Walker		
UserID	BIGINT	Jedinstveni identifikator korisnika
FirstName	VARCHAR	Ime korisnika
LastName	VARCHAR	Prezime korisnika
Email	VARCHAR	Elektronička pošta korisnika
PhoneNumber	VARCHAR	Broj mobitela korisnika
Public	BOOLEAN	Javnost podataka korisnika
Deleted	BOOLEAN	Račun obrisan

WalkerLogin Entitet sadrži podatke za prijavu korisnika na aplikaciju. Njegovi atributi su: identifikator korisnika, korisničko ime i lozinka. Entitet WalkerLogin je u *One-to-One* vezi s entitetom Walker preko jedinstvenog identifikatora korisnika.

WalkerLogin		
UserID	BIGINT	Jedinstveni identifikator korisnika
UserName	VARCHAR	Korisničko ime za prijavu
Password	VARCHAR	Lozinka za prijavu

Association Entitet sadrži sve bitne informacije o registriranoj udruzi za pse. Njegovi atributi su: ID, OIB i naziv udruge, ime i prezime vlasnika udruge, elektroničku pošta, web-stranica udruge, opis, web-adresa logo-a udruge, telefonski broj za kontaktiranje udruge i obrisan (atribut koji postaje true ako udruga obriše svoj račun). Entitet Association je u *One-to-One* vezi s entitetom AssociationLocation preko jedinstvenog identifikatora udruge i u *One-to-Many* vezi s entitetom Dog preko jedinstvenog identifikatora udruge.

Association		
AssociationID	BIGINT	Jedinstveni identifikator udruge
AssociationOIB	VARCHAR	OIB udruge
Name	VARCHAR	Naziv udruge
FirstName	VARCHAR	Ime vlasnika udruge
LastName	VARCHAR	Prezime vlasnika udruge
Email	VARCHAR	Elektronička pošta udruge
WebAddress	VARCHAR	Web-stranica udruge
Description	VARCHAR	Opis udruge
PictureURL	VARCHAR	Logo udruge
PhoneNumber	VARCHAR	Kontaktни broj udruge
Deleted	BOOLEAN	Račun obrisan

AssociationLogin Entitet sadrži podatke za prijavu udruge na aplikaciju. Njegovi atributi su: identifikator udruge, korisničko ime i lozinka. Entitet AssociationLogin je u *One-to-One* vezi s entitetom Association preko jedinstvenog identifikatora udruge.

AssociationLogin		
AssociationID	BIGINT	Jedinstveni identifikator udruge
UserName	VARCHAR	Korisničko ime za prijavu
Password	VARCHAR	Lozinka za prijavu

Dog Entitet sadrži informacije o psu koji je registriran u udruzi. Atributi tog entiteta su: ID psa, ID udruge u kojoj je registriran, ime psa, pasmina, web-adresa slike psa, preferirani način šetnje, opis psa i obrisan (atribut koji postaje true ako udruge obriše psa). Entitet Dog je u *One-to-One* vezi s entitetom DogAvailability preko jedinstvenog identifikatora psa, u *Many-to-One* vezi s entitetom Association preko jedinstvenog identifikatora udruge te u *Many-to-One* vezi s entitetom WalkJournal preko jedinstvenog identifikatora psa.

Dog		
DogID	BIGINT	Jedinstveni identifikator psa
AssociationID	BIGINT	ID udruge u kojoj se nalazi pas
Name	VARCHAR	Ime psa
Breed	VARCHAR	Pasmina

Dog		
PictureURL	VARCHAR	Web-adresa slike psa
PreferredWalkStyle	VARCHAR	Preferirani način šetnje psa(individualna ili grupna šetnja)
Description	VARCHAR	Opis psa
Deleted	BOOLEAN	Pas obrisao

Reservation Ovaj entitet sadrži podatke o šetnjama. Atributi su: ID rezervacije šetnje, ID korisnika koji je šetao psa/e, datum šetnje, vrijeme početka i kraja šetnje, način šetnje te informaciju o tome je li šetnja otkazana ili ne. Entitet WalkJournal je u *Many-to-One* vezi s entitetom Walker preko jedinstvenog identifikatora korisnika, u *One-to-Many* vezi s entitetom Dog preko jedinstvenog identifikatora psa te u *Many-to-One* vezi s entitetom ReservationDogList preko jedinstvenog identifikatora rezervacije šetnje.

Reservation		
ReservationID	BIGINT	Jedinstveni identifikator rezervacije šetnje
UserID	BIGINT	Jedinstveni identifikator korisnika koji je šetač
date	DATE	Datum zakazane šetnje
StartTime	TIME	Vrijeme početka šetnje
ReturnTime	TIME	Vrijeme povratka iz šetnje
WalkStyle	VARCHAR	Način šetnje(individualna/grupna)
Cancelled	BOOLEAN	Je li šetnja otkazana

AssociationLocation Ovaj entitet sadrži podatke o lokacijama na kojima se nalazi udruga. Atributi ovog entiteta su: ID udruge, grad, ulica i kućni broj adrese udruge. Entitet AssociationLocation je u *One-to-One* vezi s entitetom Association preko jedinstvenog identifikatora udruge.

AssociationLocation		
AssociationID	BIGINT	Jedinstveni identifikator udruge
City	VARCHAR	Grad
Street	VARCHAR	Ulica
HouseNumber	VARCHAR	Kućni broj

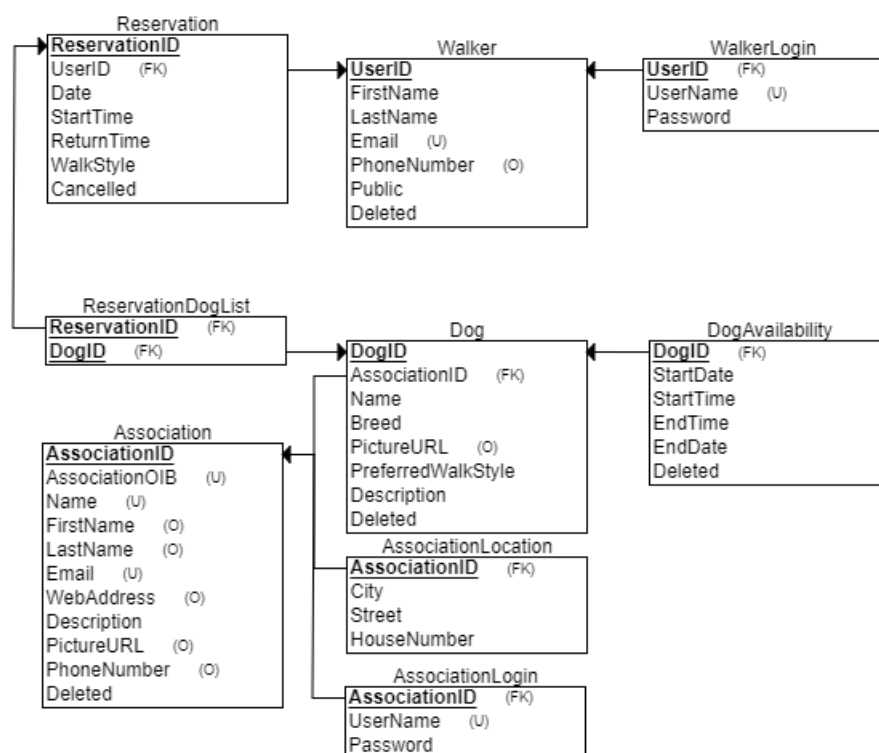
DogAvailability Entitet sadrži podatke o vremenu kada je pas slobodan za šetnju. Atributi tog entiteta su: ID psa, datum, vrijeme od i vrijeme do kada se može rezervirati šetnja. Entitet DogAvailability je u *One-to-One* vezi s entitetom Dog preko jedinstvenog identifikatora psa.

DogAvailability		
DogID	BIGINT	Jedinstveni identifikator psa
StartDate	DATE	Datum od kada se može rezervirati šetnja
EndDate	DATE	Datum do kada se može rezervirati šetnja
StartTime	TIME	Vrijeme od kada se može rezervirati šetnja
EndTime	TIME	Vrijeme do kada se može rezervirati šetnja
Deleted	BOOLEAN	Oznacava je li dostupnost obrisana

ReservationDogList Ovaj slabi entitet služi kao popis pasa koji su bili u šetnji u jednoj rezervaciji. Ključ ovog entiteta je jedinstveni identifikator rezervacije i identifikator psa. Entitet ReservationDogList je u *One-to-Many* vezi s entitetom WalkJournal preko jedinstvenog identifikatora rezervacije.

ReservationDogList		
ReservationID	BIGINT	Jedinstveni identifikator rezervacije
DogID	BIGINT	Jedinstveni identifikator psa

4.1.2 Dijagram baze podataka

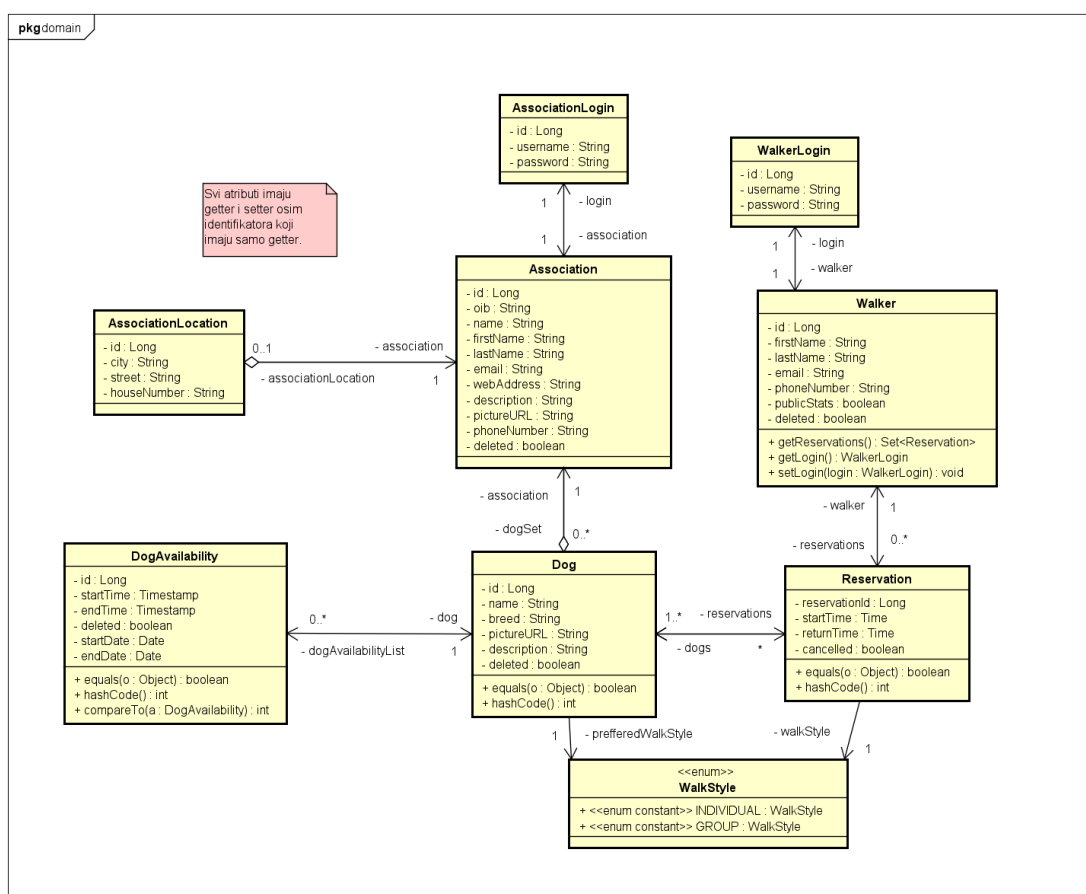


Slika 4.4: Dijagram baze podataka

4.2 Dijagram razreda

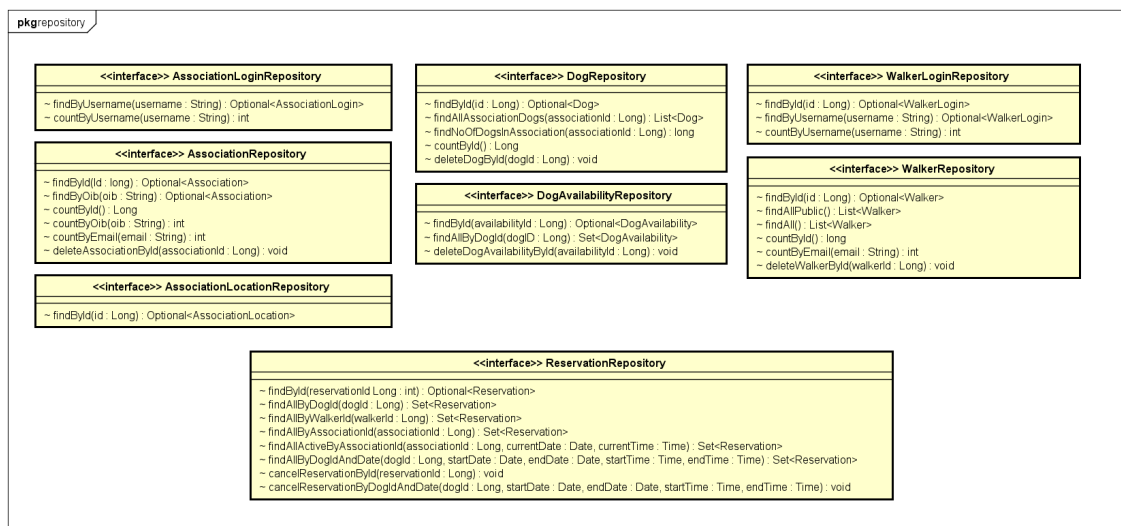
Dijagrami razreda omogućavaju pojednostavljen pogled i lakše razumijevanje komponenti arhitekture objektno orijentiranog sustava. Razredi sustava izrađenih u Springu prate norme razvojnog okvira. U našem sustavu zato imamo četiri glavna paketa i odgovarajuće potpakete. Njihov sadržaj moguće je vidjeti na sljedećim slikama:

- **domain** - Slika 4.5
- **repository** - Slike 4.6 i 4.7
- **service** - Slika 4.8
 - **impl** - Slika 4.9
- **rest** - Slika 4.10
 - **dto** - Slika 4.12

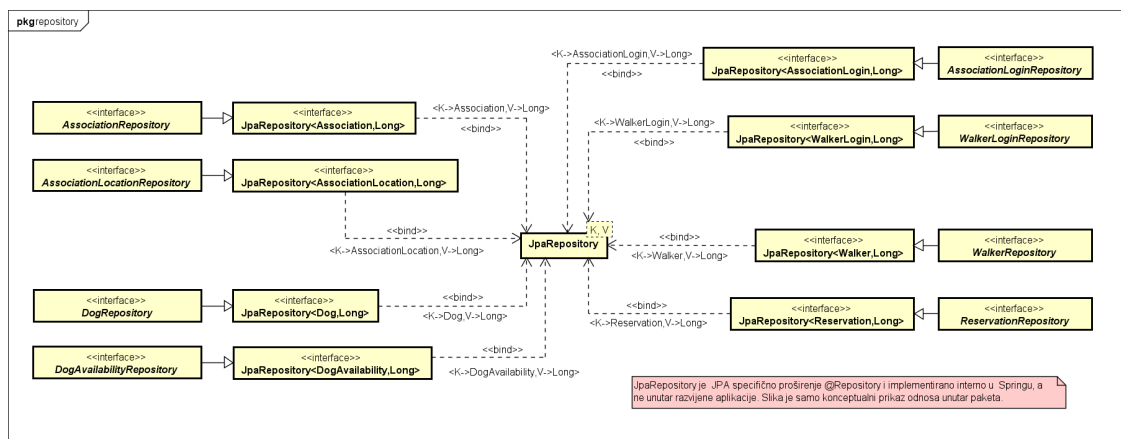


Slika 4.5: Dijagram domenskih razreda

Na dijagramima je moguće vidjeti da kod prati načela troslojne arhitekture navedena na početku poglavlja. Klase paketa *repository* služe za spremanje i dohvaćanje podataka iz baze pri čemu Spring nudi gotovo rješenje većine komunikacije s bazom u obliku JPA repozitorija (engl. *Java Persistence API*). Repozitoriji našeg koda nasljeđuju JPA repozitorije kao što je moguće vidjeti na slici 4.7.

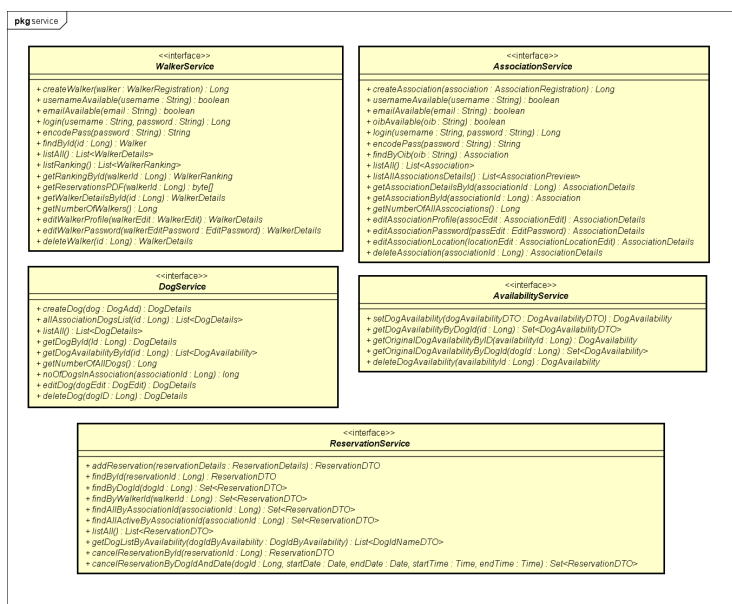


Slika 4.6: Dijagram spremišnih razreda

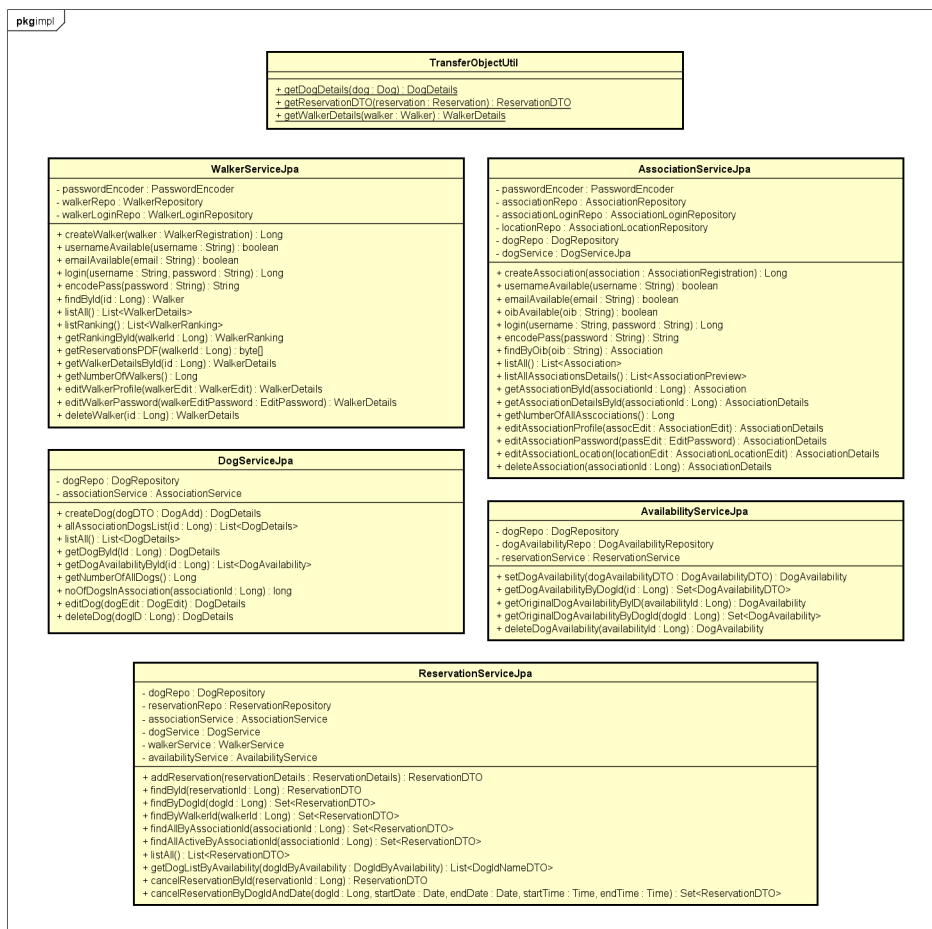


Slika 4.7: Nasljeđivanje JPA repozitorija

Paket *service* sadrži sučelja koja definiraju funkcionalnosti servisnog sloja. Ona se ostvaruju u potpaketu *impl*. Implementacijske klase sadrže reference na potrebne repozitorije kako bi mogle dohvaćati i spremati podatke te manipulirati njima.

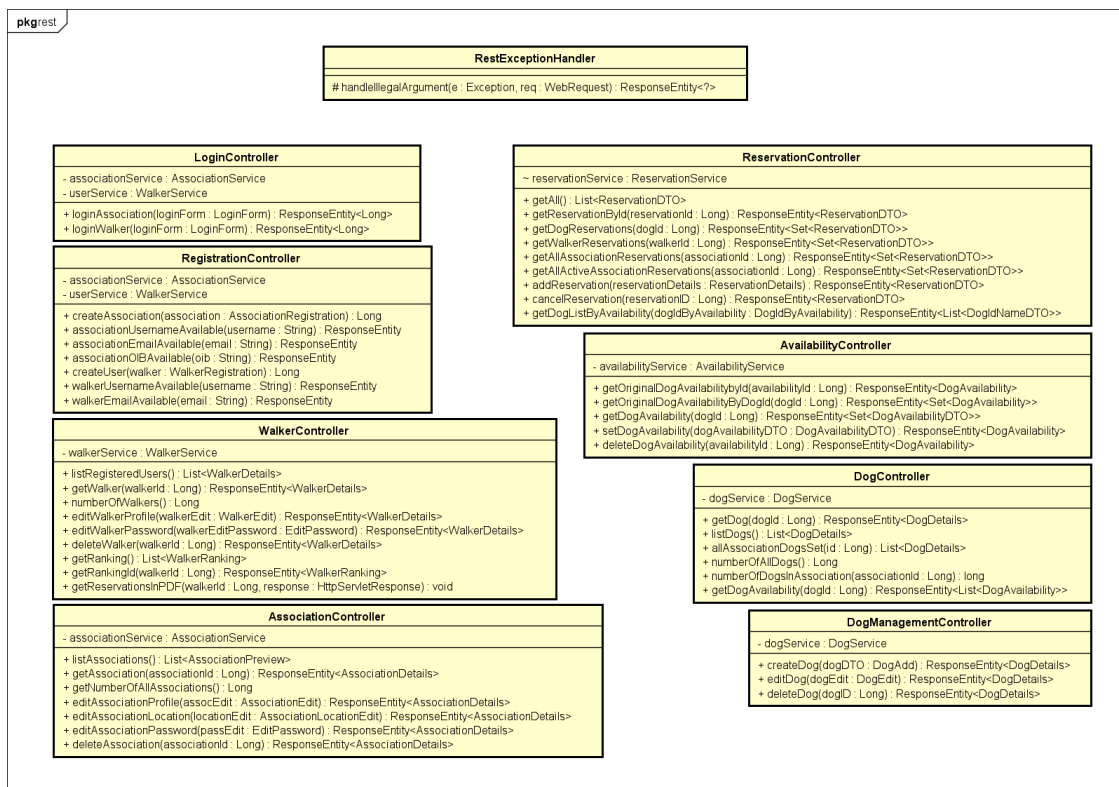


Slika 4.8: Dijagram servisnih sučelja



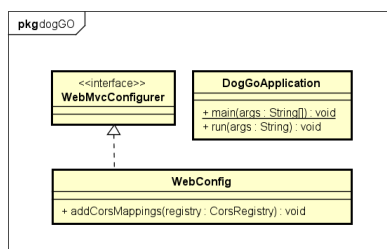
Slika 4.9: Dijagram razreda implementacije servisnih sučelja

Razredi paketa *rest* komuniciraju s *frontend* stranom razvijenog sustava i nazivaju se i kontrolerima. Razmijenjuju informacije u formatu JSON i odgovarajućim HTTP status kodom. Za formatiranje odgovora koriste se razredima paketa DTO (engl. *Data Transfer Object*). Sadrže reference na servisna sučelja kojima dohvaćaju obrađene podatke ili ih šalju na obradu.

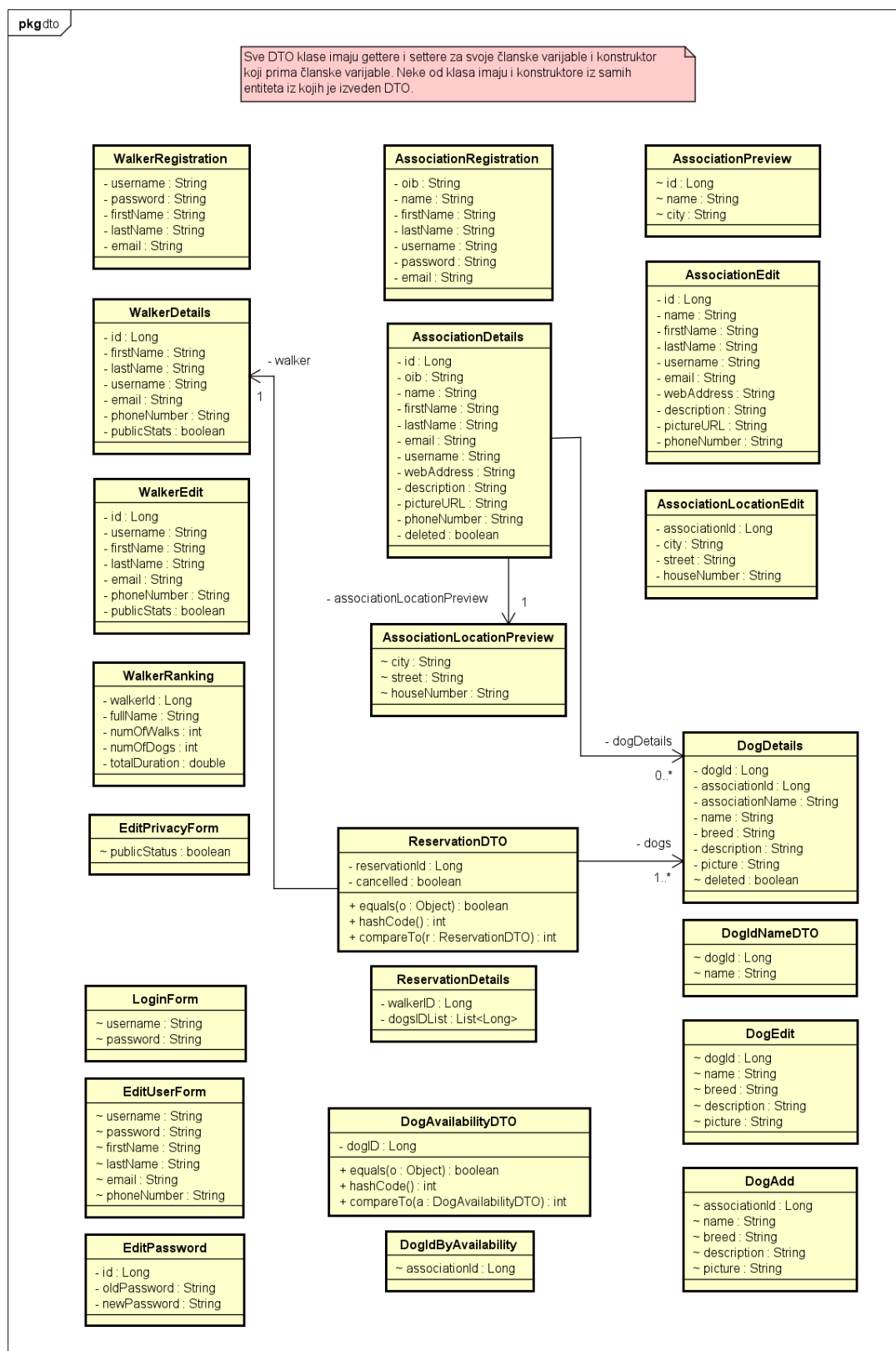


Slika 4.10: Dijagram razreda kontrolera

Na slici 4.11 moguće je vidjeti ishodišni razred cijele aplikacije *DogGoApplication* i razred konfiguracije dozvoljenih HTTP zahtjeva u aplikaciji koji nasljeđuje već gotovi razred Spring razvojnog okvira *WebMvcConfigurer*.



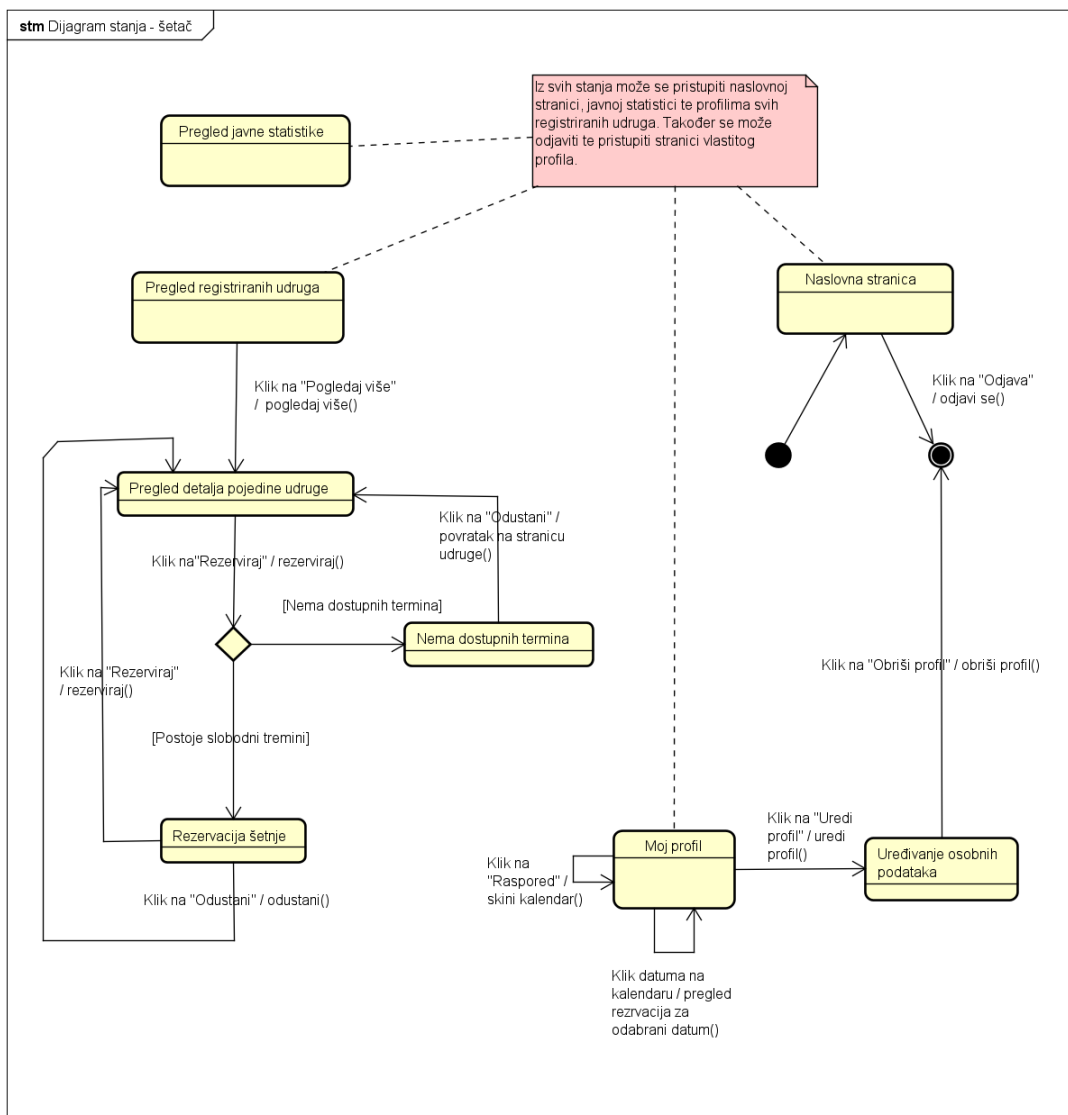
Slika 4.11: Ishodišni paket aplikacije



Slika 4.12: Dijagram razreda objekata prijenosa podataka

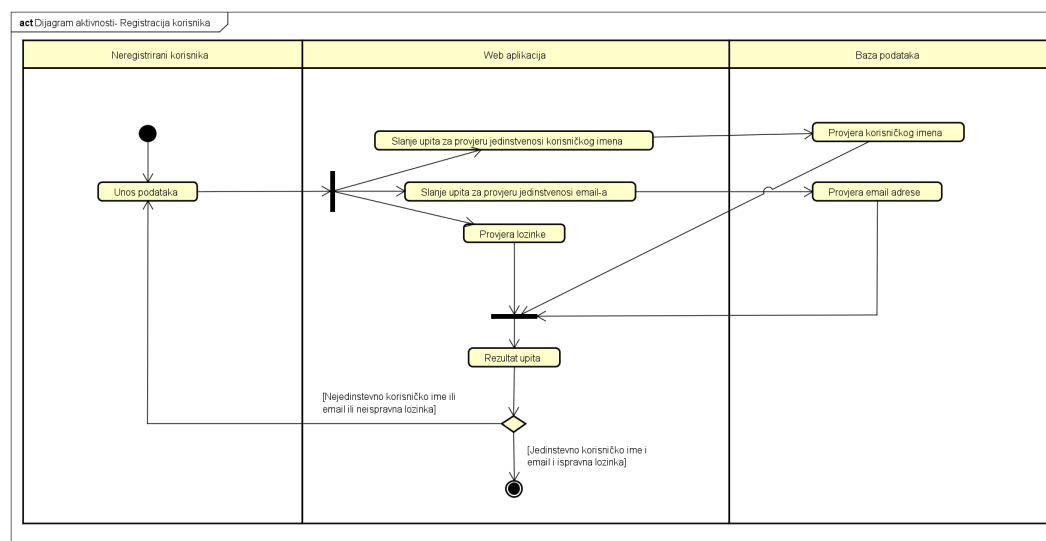
4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekata te prijelaze iz jednog u drugo stanje na temelju događaja. Na slici 4.13 je prikazan dijagram stanja za registriranog šetača. Neregistriranom korisniku se prikazuje naslovna stranica. Od tamo, korisnik može pregledati javnu statistiku te može dobiti uvid u sve registrirane udruge i njihove profile gdje su prikazane informacije o njima i svim njihovim psima. Nakon prijave korisnika kao šetača, korisnik dobiva mogućnost rezervacije šetnje odabranog psa u slobodnom terminu. Na stranici „Moj profil“ korisnik može urediti osobne podatke, pregledati kalendar rezerviranih šetnji te u obliku pdf-a skinuti vlastiti raspored nadolazećih šetnji te obrisati svoj profil.

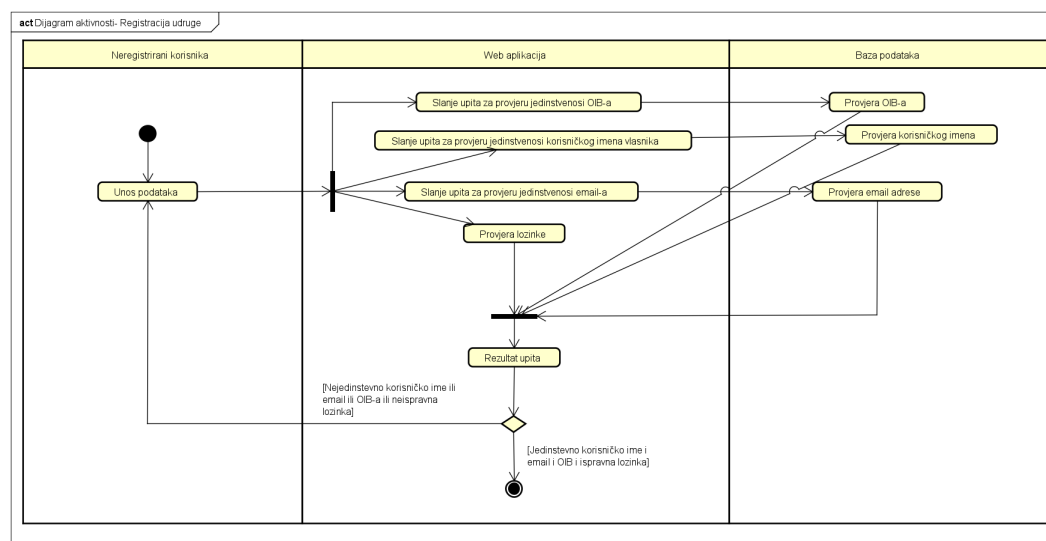


Slika 4.13: Dijagram stanja

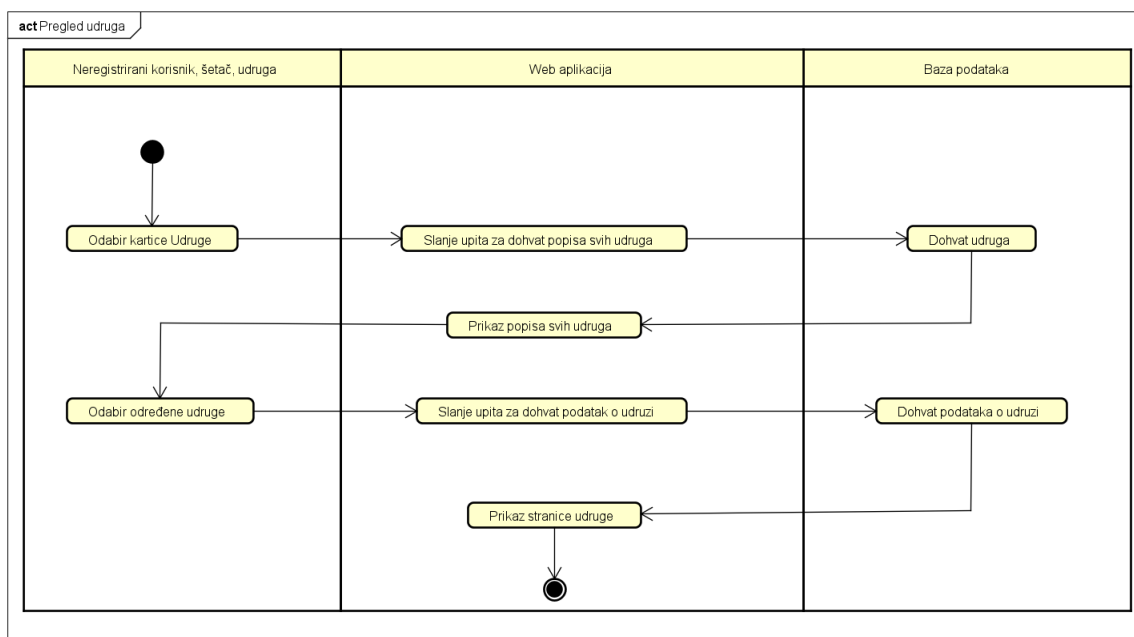
4.4 Dijagram aktivnosti



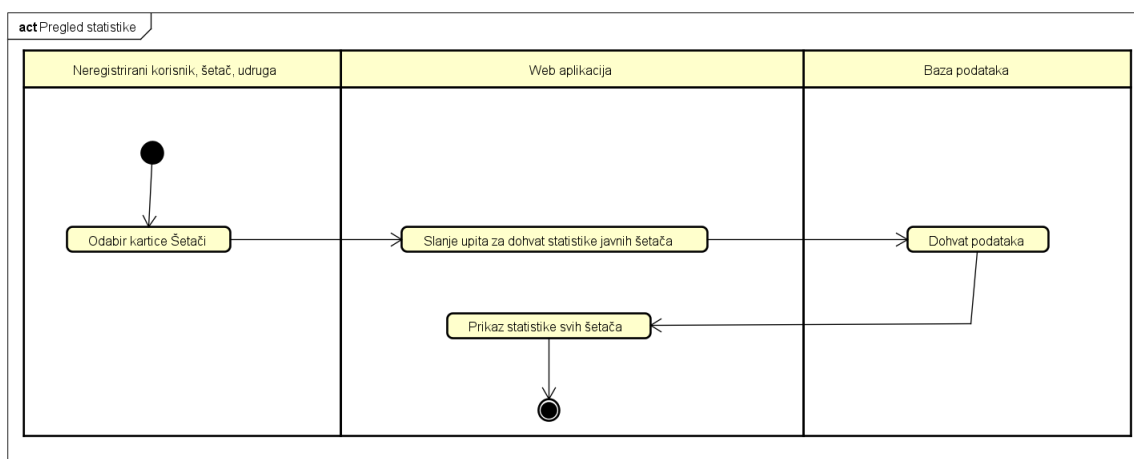
Slika 4.14: Dijagram aktivnosti za registraciju korisnika



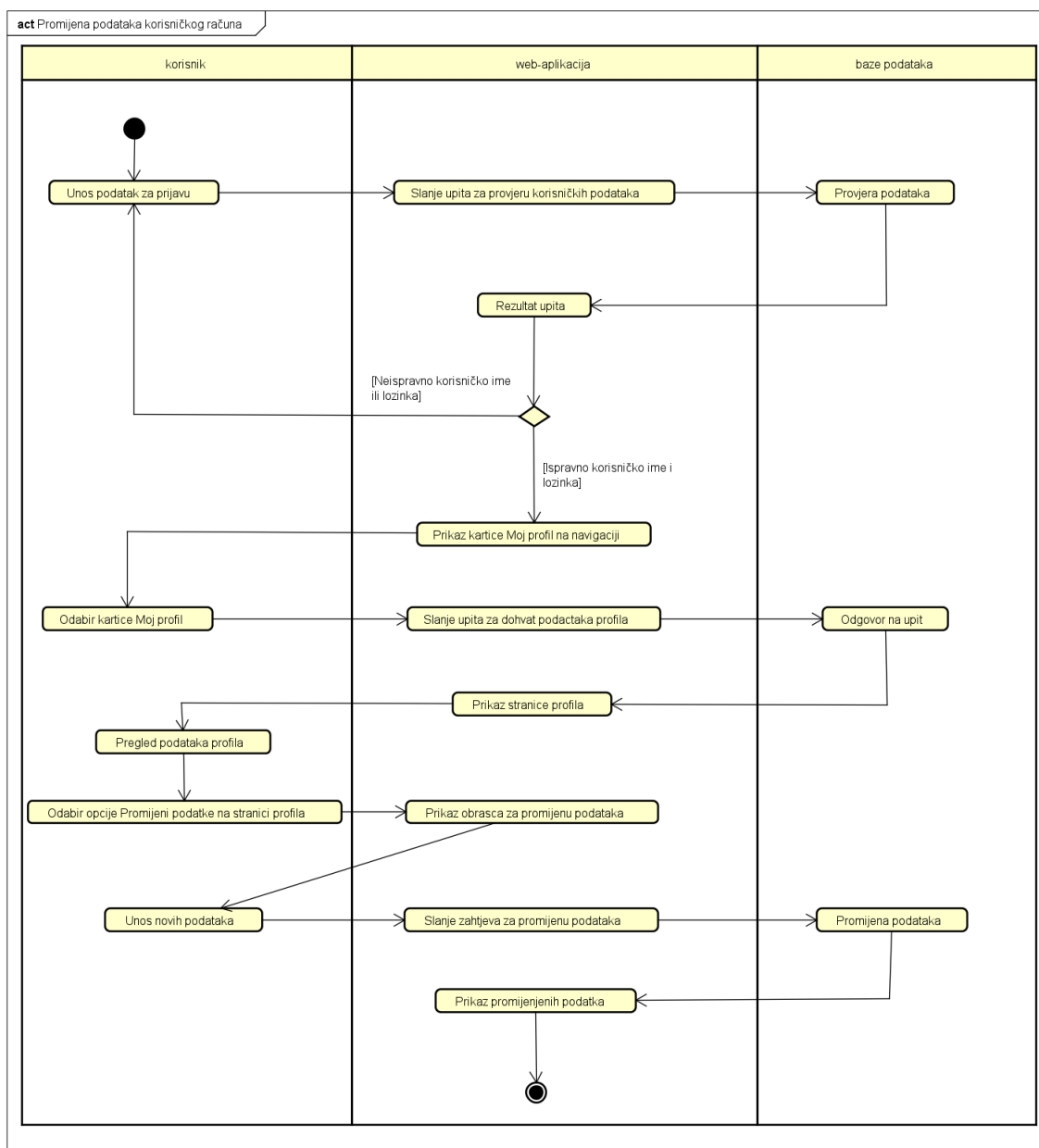
Slika 4.15: Dijagram aktivnosti za registraciju udruga



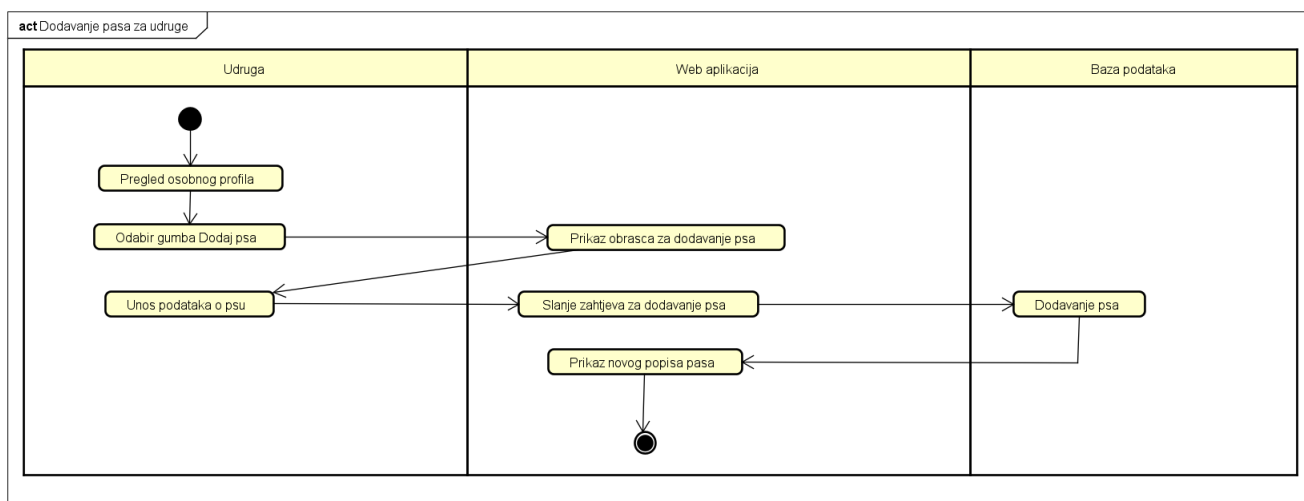
Slika 4.16: Dijagram aktivnosti za pregled svih udruga



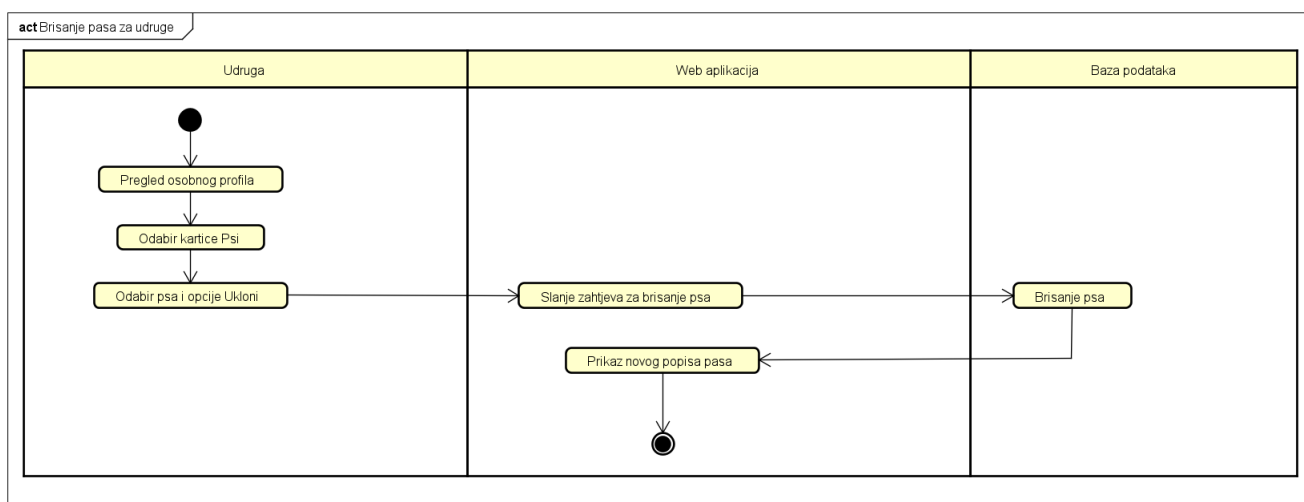
Slika 4.17: Dijagram aktivnosti za pregled statistike šetača



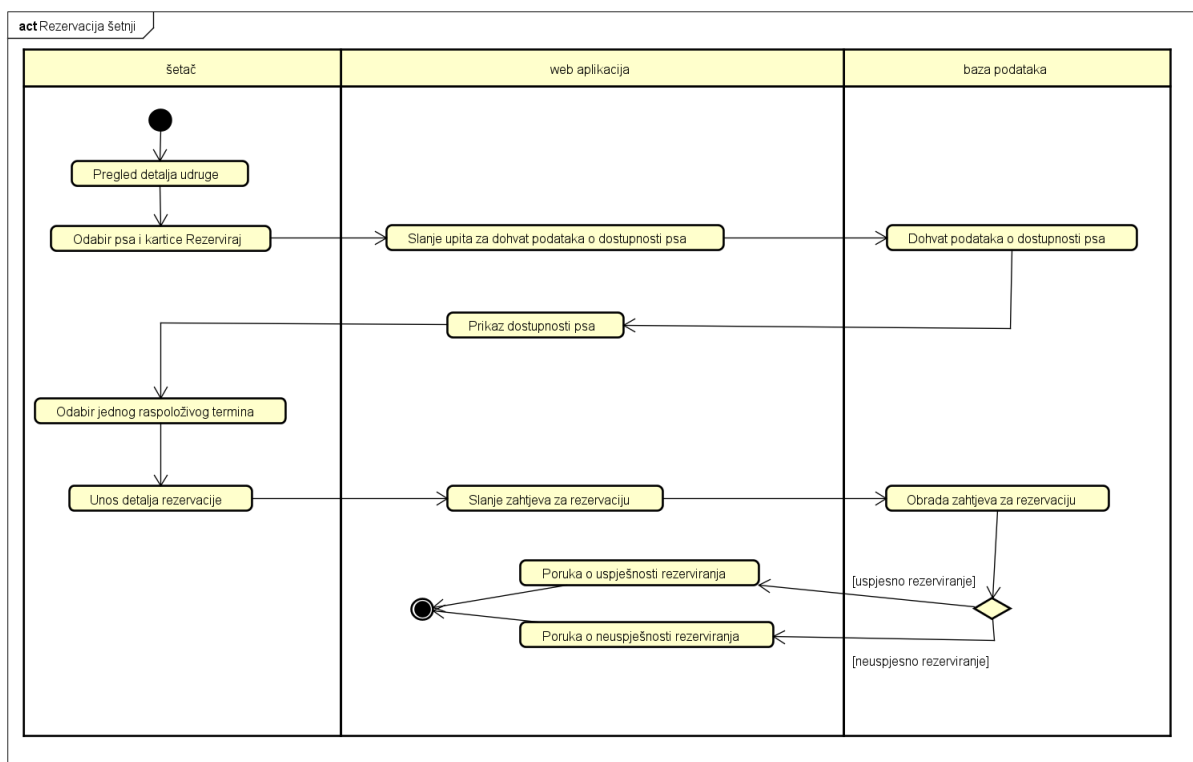
Slika 4.18: Dijagram aktivnosti za promjenu podataka korisničkog računa



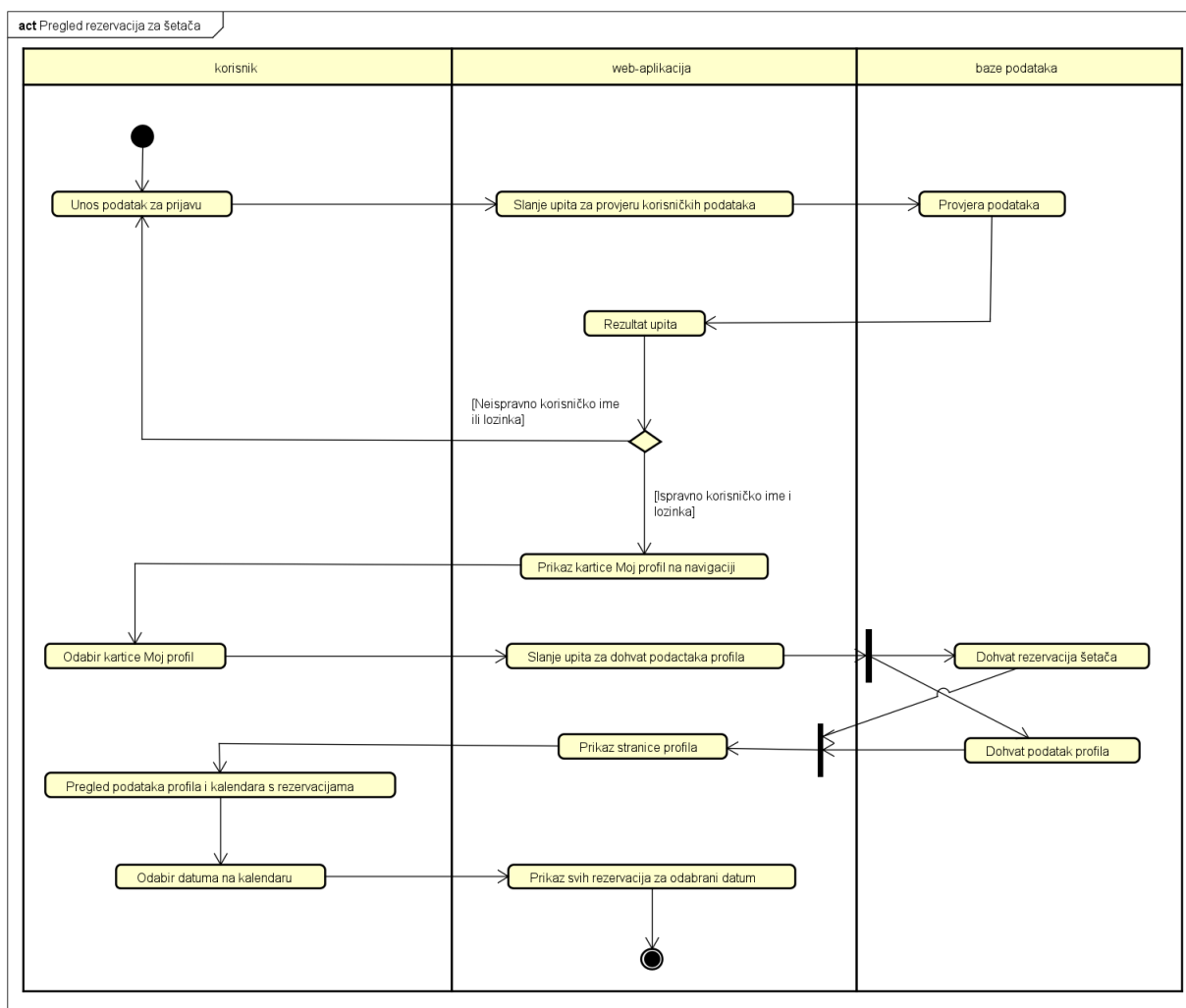
Slika 4.19: Dijagram aktivnosti za dodavanje pasa kod udruge



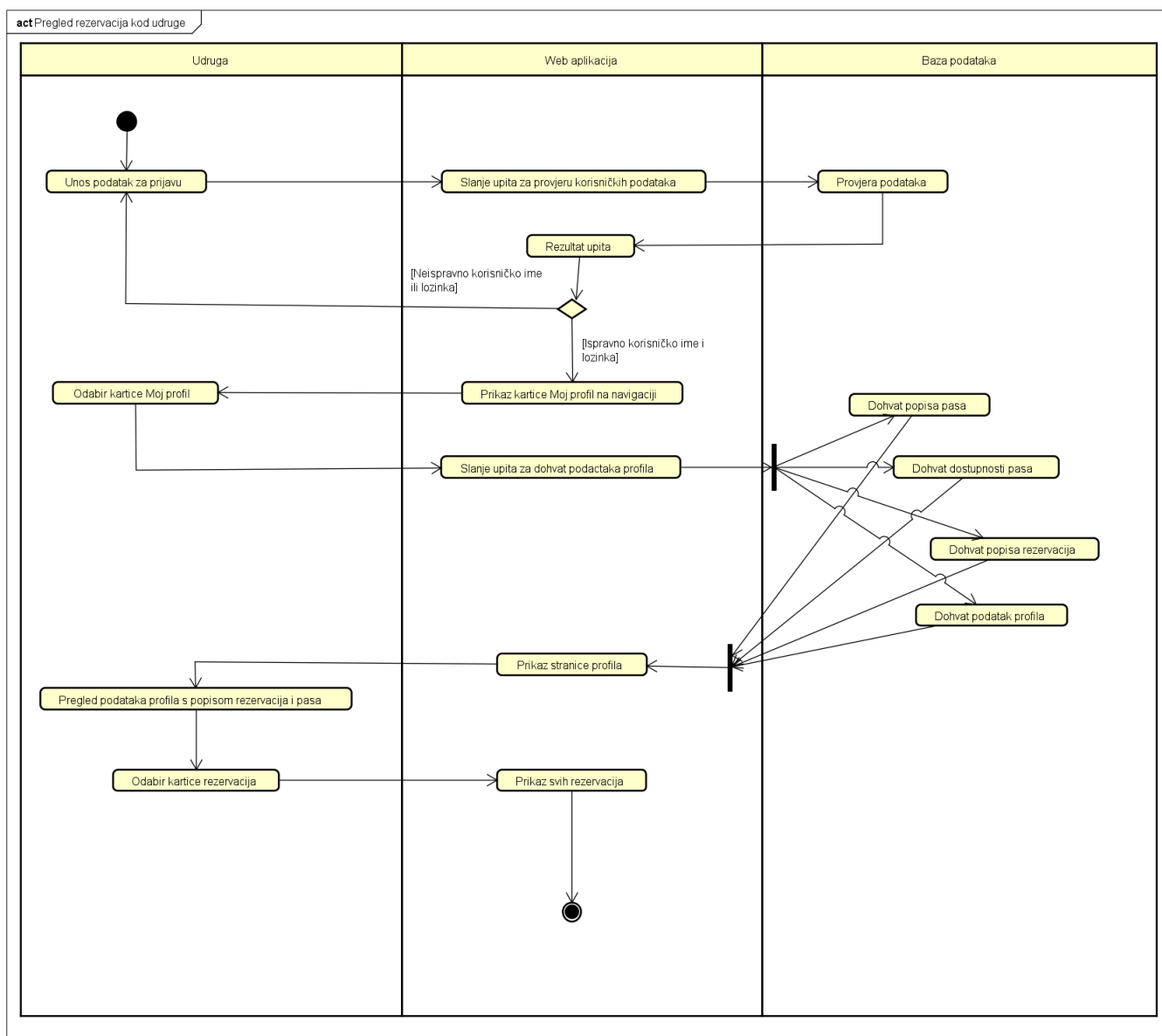
Slika 4.20: Dijagram aktivnosti za brisanje pasa kod udruge



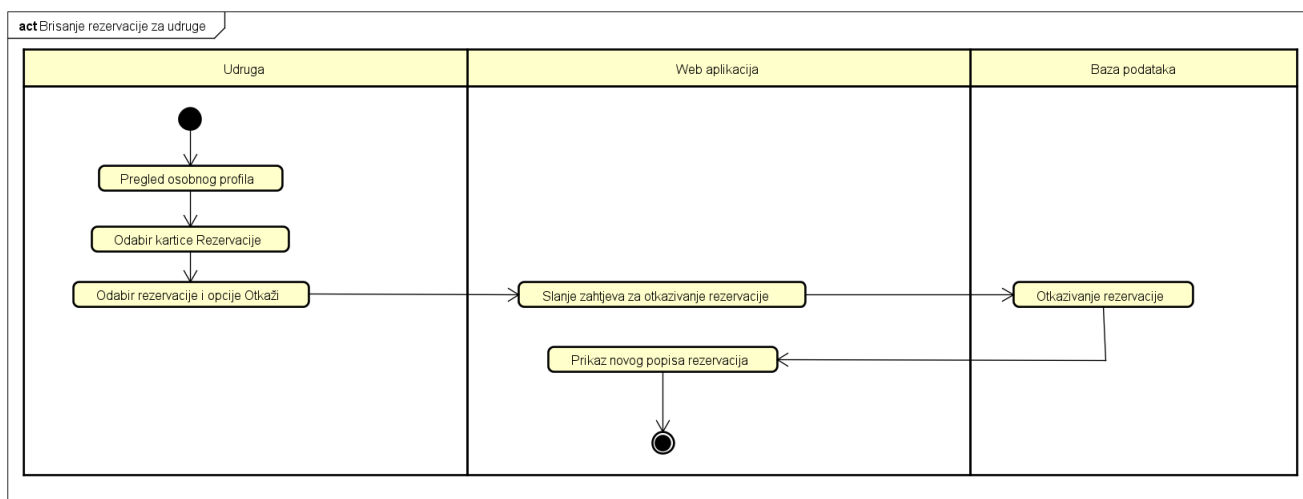
Slika 4.21: Dijagram aktivnosti za rezervaciju šetnji



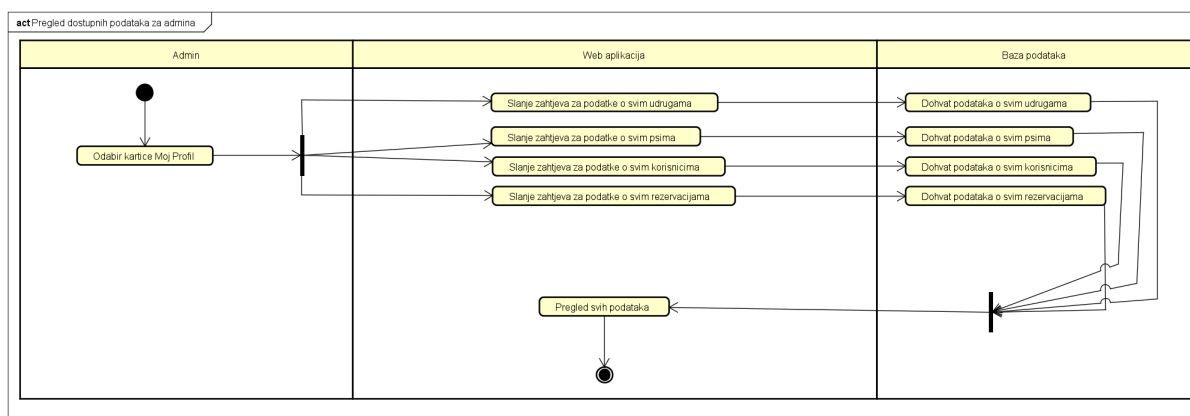
Slika 4.22: Dijagram aktivnosti za pregled rezervacija šetača



Slika 4.23: Dijagram aktivnosti za pregled rezervacija udruge



Slika 4.24: Dijagram aktivnosti za brisanje rezervacije kod udruge



Slika 4.25: Dijagram aktivnosti za pregled dostupnih podataka za admina

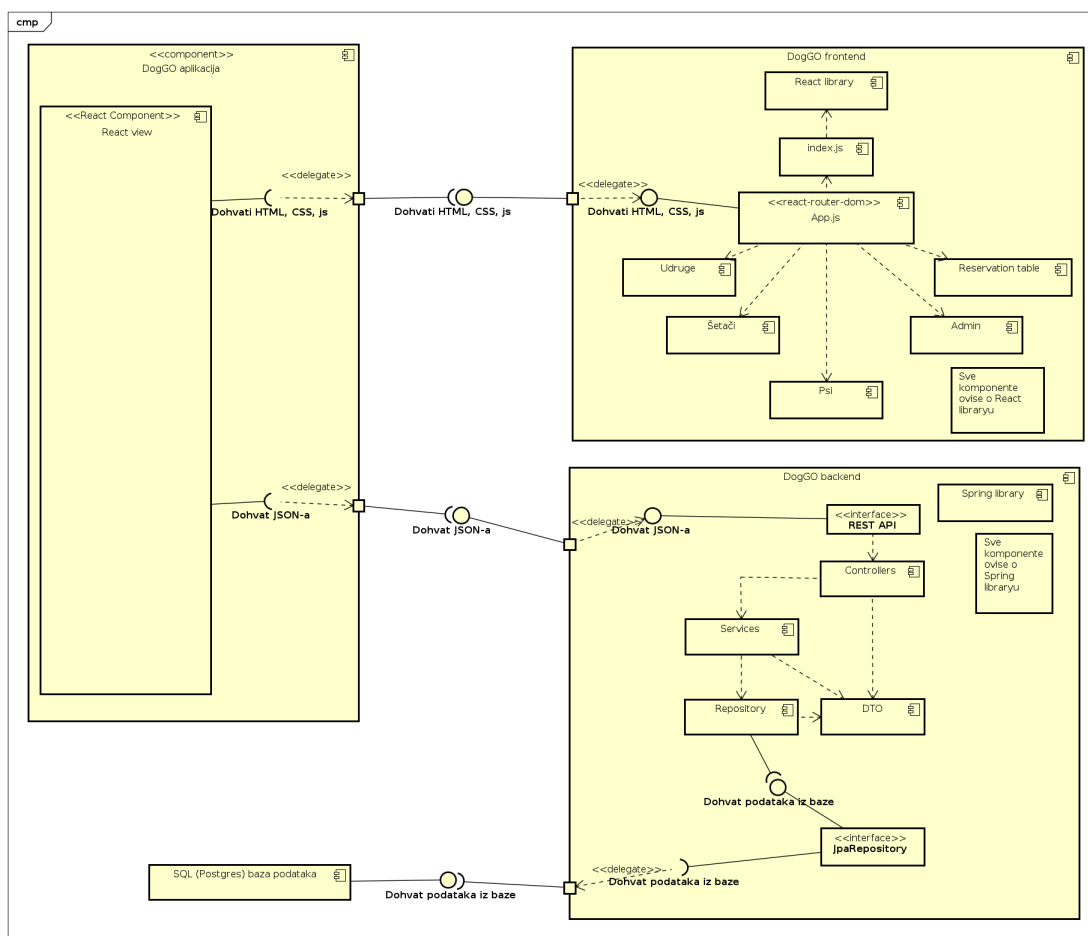
4.5 Dijagram komponenti

Dijagram komponenti je strukturni, statički UML dijagram koji vizualizira organizaciju i međuovisnost komponenata aplikacije, s naglaskom na implementaciju sustava, stoga je koristan za stjecanje okvirne ideje o implementaciji sustava, bez ulaženja u prevelike detalje.

U prikazanom dijagramu našu aplikaciju smo podijelili na 4 glavne komponente: komponentu React view, koji je zadužen za de-facto upravljanje cijelom aplikacijom, komponentu frontend logike, zaduženu za dohvat odgovarajućih HTML, CSS i .js datoteka, komponentu backend logike, zaduženu za dohvat i obradu podataka iz baze, te samu SQL bazu podataka.

Preko sučelja za dohvat HTML-a, CSS-a i .js datoteka pristupa se komponenti frontend logike. Njen centralni dio je router komponenta (u našem slučaju datoteka App.js), zadužena za usklađivanje rada ostalih komponenti i određivanje koja će se datoteka proslijediti za prikaz u klijentovom browseru. Naravno, sve frontend komponente ovise o React biblioteci.

Komunikacija s komponentom backend logike se odvija preko REST sučelja (zahtjevi GET, PUT, POST, DELETE). Zahtjevi se, nakon primitka u kontrolerima, prosljeđuju servisima, odnosno komponentama repozitorija, koji preko sučelja Jpa-Repository komunicira s bazom i dohvaća tražene podatke. Prijenos podataka između komponenti backend logike ostvaren je korištenjem DTO klasa, stoga sve komponente imaju ovisnost prema njima. Sve backend komponente ovise o Spring biblioteci (naravno, osim baze, no ona je zasebna komponenta u odnosu na backend).



Slika 4.26: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

5.1.1 Backend tehnologije

Odabrani programski jezik *backenda* naše web-aplikacije jest Java¹ u sklopu razvojnog okvira Spring Boot² koji pruža razrađeni model programiranja i konfiguracije te je jedan od najčešće korištenih Java EE okvira za izgradnju aplikacija. Korištene su i pripadne Spring Data³ i Spring Security⁴ podrške. Za strukturiranje i automatizaciju razvojnog ciklusa projekta korišten je alat Maven⁵. Odabrani sustav upravljanja relacijskim bazama podataka jest PostgreSQL⁶, dok je u razvojnoj i testnoj fazi projekta korišten *in-memory* sustav H2⁷. Za lakšu navigaciju i testiranje aplikacijskog programskog sučelja korišten je alat Swagger UI⁸. Za generiranje PDF dokumenata unutar aplikacije, korištena je knjižica IText⁹. Korištena razvojna okruženja na *backendu* su Eclipse IDE¹⁰ i IntelliJ IDEA¹¹.

5.1.2 Frontend tehnologije

Korisničko sučelje aplikacije temeljeno je na označnom jeziku HTML5¹² i stilskom jeziku CSS¹³. Modelirano je pomoću razvojnog okvira React¹⁴ u programskom jeziku JavaScript. React se koristi za izradu jednostraničnih aplikacija, tj. stranica koje komuniciraju s korisnikom dinamički mijenjajući trenutnu web stranicu

¹<https://www.java.com/en/>

²<https://spring.io/>

³<https://spring.io/projects/spring-data>

⁴<https://spring.io/projects/spring-security>

⁵<https://maven.apache.org/>

⁶<https://www.postgresql.org/>

⁷<https://www.h2database.com/>

⁸<https://swagger.io/tools/swagger-ui/>

⁹<https://itextpdf.com/en>

¹⁰<https://www.eclipse.org/eclipseide/>

¹¹<https://www.jetbrains.com/idea/>

¹²<https://html.spec.whatwg.org/>

¹³<https://www.w3.org/Style/CSS/Overview.en.html>

¹⁴<https://reactjs.org/>

umjesto učitavanja cijele nove stranice. Složenije jednostranične aplikacije osim React-a koriste i niz drugih knjižica za nadogradnju komponenti. Za upravljanje knjižicama koristi se npm¹⁵. Neke od mnogih korištenih knjižica u izrađenoj aplikaciji su Bootstrap¹⁶, Material UI¹⁷, Semantic UI¹⁸ i Redux¹⁹. Korištena razvojna okolina na *frontendu* jest Visual Studio Code²⁰.

5.1.3 Puštanje u pogon

Cijeli sustav uspostavljen je i pušten u pogon na platformi Heroku²¹, servisu temeljenom na oblaku.

5.1.4 Dokumentacija

Za dokumentaciju programskog rješenja korišten je Astah²², alat za modeliranje UML dijagramima. Za vizualizaciju i modeliranje baze podataka korišten je alat Erdplus²³.

5.1.5 Timska komunikacija

Za svrhe komunikacije unutar tima korištene su sljedeće platforme:

- Trello²⁴ - organizacijska platforma temeljena na kanban metodi
- Discord²⁵ - VoIP platforma za brzu komunikaciju
- WhatsApp²⁶ - aplikacija za razmjenu poruka
- Microsoft Teams²⁷ - platforma za video konferencije

Za distribuirano upravljanje kodom i verzioniranje korišten je sustav Git²⁸ s udaljenim repozitorijem posluženim na GitLab²⁹ platformi.

¹⁵<https://www.npmjs.com/>

¹⁶<https://getbootstrap.com/>

¹⁷<https://material-ui.com/>

¹⁸<https://semantic-ui.com/>

¹⁹<https://redux.js.org/>

²⁰<https://code.visualstudio.com/>

²¹<https://www.heroku.com/>

²²<https://astah.net/>

²³<https://erdplus.com/>

²⁴<https://trello.com/en>

²⁵<https://discord.com/>

²⁶<https://www.whatsapp.com/>

²⁷<https://www.microsoft.com/en-us/microsoft-365/microsoft-teams/group-chat-software>

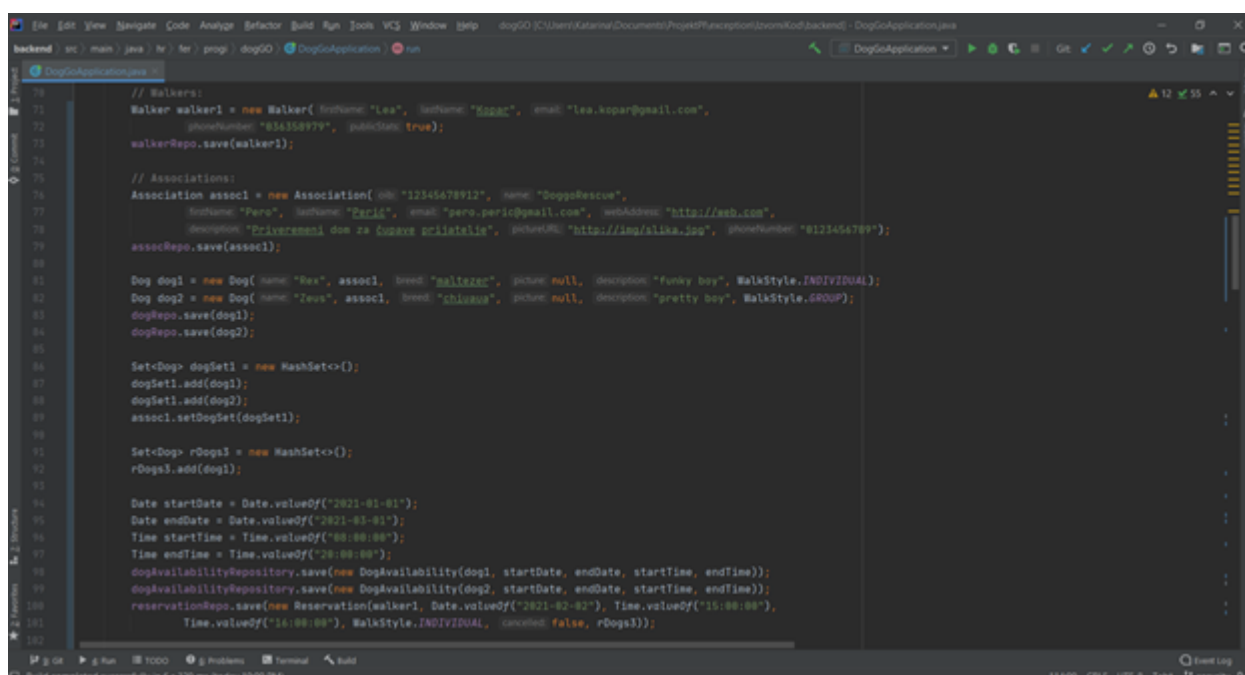
²⁸<https://git-scm.com/>

²⁹<https://about.gitlab.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Uspješna rezervacija je temeljna funkcionalnost ove aplikacije. JUnit testovima ispitani su rubni i redovni slučajevi te oni koji izazivaju pogreške. Priložen je kod kojim su u bazu dodani entiteti potrebni za ispitivanje funkcionalnosti rezervacije te kod Junit testova.



Slika 5.1: Prikaz inicijalizacije testnih podataka

Ispitni slučaj 1: Pokušaj rezervacije gdje je sat početka vremenski nakon sata završetka šetnje

Nije moguće napraviti rezervaciju na isti dan, a da je pritom vrijeme početka šetnje veće od vremena završetka šetnje. Sljedećim testom izazivamo iznimku RequestDeniedException s odgovarajućom porukom


```
@Test
void addInvalidReservation() {
    List<Long> dogList = new ArrayList<>();

    dogList.add(dogDetails1.getDogId());

    ReservationDetails reservationDetails = new
ReservationDetails(walker1.getId(), Date.valueOf("2021-02-10"),
        Time.valueOf("10:00:00"), Time.valueOf("8:00:00"), dogList);

    ex = Assertions.assertThrows(RequestDeniedException.class,
        () -> reservationService.addReservation(reservationDetails));
    Assertions.assertEquals("The interval of the reservation is invalid.",
ex.getMessage());
}
```

Slika 5.2: Prikaz 1. Junit testa

Ispitni slučaj 2: Pokušaj rezervacije gdje je pas za grupnu šetnju jedini u rezervaciji

Psi koji su namijenjeni grupnoj šetnji ne mogu biti jedini u rezervaciji, stoga sljedećim testom želimo napraviti rezervaciju s listom pasa koja ima samo jednog psa, a koji mora biti u grupi s još minimalno jednim psom da bi rezervacija bila uspješna. Pokušaj rezervacije završava iznimkom `RequestDeniedException` s odgovarajućom porukom.

```
@Test
void addGroupDogToIndividual() {
    List<Long> dogList = new ArrayList<>();
    dogList.add(dogDetails2.getDogId());

    ReservationDetails reservationDetails = new
ReservationDetails(walker1.getId(), Date.valueOf("2021-02-10"),
        Time.valueOf("10:00:00"), Time.valueOf("11:00:00"), dogList);

    ex = Assertions.assertThrows(RequestDeniedException.class,
        () -> reservationService.addReservation(reservationDetails));
    Assertions.assertEquals("Cannot add dog with walk style GROUP to an
individual walk.", ex.getMessage());
}
```

Slika 5.3: Prikaz 2. Junit testa

Ispitni slučaj 3: Pokušaj rezervacije gdje je pas za individualnu šetnju dodan u grupnu šetnju

Psi koji su namijenjeni individualnoj šetnji ne mogu biti stavljani u grupnu šetnju, pa pokušaj rezervacije gdje je takav pas stavljen u grupu s drugima izaziva `RequestDeniedException` iznimku s odgovarajućom porukom.

```
@Test
void addIndividualDogToGroup() {
    List<Long> dogList = new ArrayList<>();
    dogList.add(dogDetails1.getDogId());
    dogList.add(dogDetails2.getDogId());

    ReservationDetails reservationDetails = new
ReservationDetails(walker1.getId(), Date.valueOf("2021-02-10"),
        Time.valueOf("10:00:00"), Time.valueOf("11:00:00"), dogList);

    ex = Assertions.assertThrows(RequestDeniedException.class,
        () -> reservationService.addReservation(reservationDetails));
    Assertions.assertEquals("Cannot add dog with walk style INDIVIDUAL to a
group walk.", ex.getMessage());
}
```

Slika 5.4: Prikaz 3. Junit testa

Ispitni slučaj 4: Pokušaj rezervacije kada je pas nedostupan

Ako se pokuša napraviti rezervacija u terminu kada pas nije dostupan, potrebno je izazvati iznimku RequestDeniedException s odgovarajućom porukom.

```
@Test
void unavailableDog() {
    List<Long> dogList = new ArrayList<>();
    dogList.add(dogDetails1.getDogId());

    ReservationDetails reservationDetails = new
ReservationDetails(walker1.getId(), Date.valueOf("2021-02-02"),
        Time.valueOf("15:00:00"), Time.valueOf("16:00:00"), dogList);

    ex = Assertions.assertThrows(RequestDeniedException.class,
        () -> reservationService.addReservation(reservationDetails));
    Assertions.assertEquals("Cannot make a reservation for dog with ID " + 1
        + " as that dog isn't available during specified reservation time",
ex.getMessage());
}
```

Slika 5.5: Prikaz 4. Junit testa

Ispitni slučaj 5: Pokušaj rezervacije kada je šetač nedostupan

Ako šetač pokuša napraviti rezervaciju u terminu u kojem je već prije napravio rezervaciju također se izaziva RequestDeniedException s odgovarajućom porukom.

```
@Test
void walkerAddReservation() {
    List<Long> dogList = new ArrayList<>();
    dogList.add(8L);

    ReservationDetails reservationDetails = new
ReservationDetails(walker1.getId(), Date.valueOf("2021-02-01"),
        Time.valueOf("13:00:00"), Time.valueOf("14:00:00"), dogList);

    ex = Assertions.assertThrows(RequestDeniedException.class,
        () -> reservationService.addReservation(reservationDetails));
    Assertions.assertEquals("This walker has already made a reservation at
this time.", ex.getMessage());
}
```

Slika 5.6: Prikaz 5. Junit testa

Ispitni slučaj 6: Pokušaj rezervacije kada je šetač nedostupan

Najbitnije je ispitati je li rezervacija koja ispunjava sve uvjete uspješno napravljena. Sljedećim testom provjeravamo jednakost hardkodiranih detalja rezervacije i detalja metode kojom vraćaju detalji uspješne rezervacije.

```
@Test
void successfulReservation() {
    List<Long> dogList = new ArrayList<>();
    dogList.add(dogDetails1.getDogId());

    ReservationDetails reservationDetails = new
ReservationDetails(walker1.getId(), Date.valueOf("2021-02-03"),
        Time.valueOf("15:00:00"), Time.valueOf("16:00:00"), dogList);

    ReservationDTO reservationDTO =
reservationService.addReservation(reservationDetails);

    Set<DogDetails> set = new HashSet<>();
    set.add(dogDetails1);

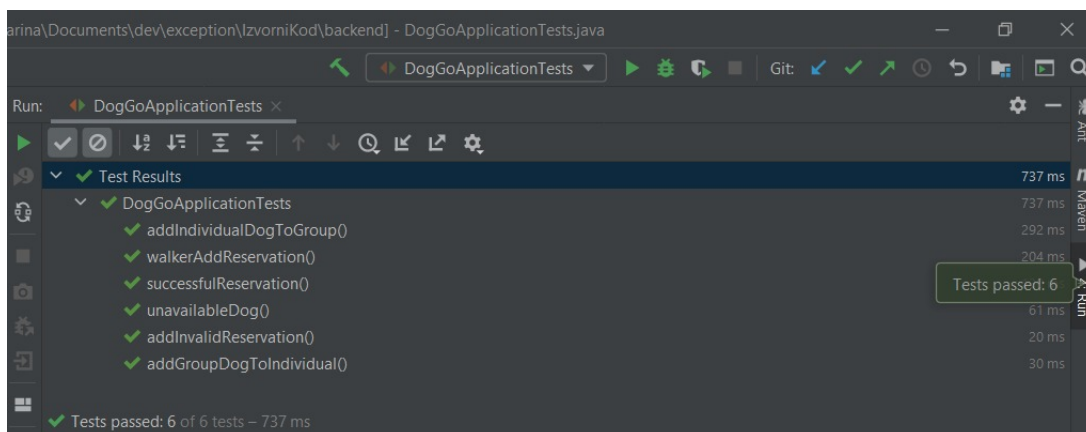
    ReservationDTO reservationDTO1 = new ReservationDTO(12L, walker1,
Date.valueOf("2021-02-03"),
        Time.valueOf("15:00:00"), Time.valueOf("16:00:00"),
WalkStyle.INDIVIDUAL, false, set);

    Assertions.assertTrue(reservationDTO.equals(reservationDTO1));
}
```

Slika 5.7: Prikaz 6. Junit testa

Rezultati testiranja

Svi navedeni Junit testovi su se uspješno proveli.



Slika 5.8: Prikaz rezultata testova

5.2.2 Ispitivanje sustava

Ispitni slučaj 1: Rezervacija šetnje

Ulaz:

1. Otvaranje početne stranice u web pregledniku.
2. Otvaranje stranice popisa udruge.
3. Odabir udruge i pritisak kursomom na gumb REZERVIRAJ.
4. Odabir datuma i vremenskog intervala šetnje.
5. Pritisak kursomom na gumb Rezerviraj.

Očekivani rezultat:

1. Prikaz naslovne stranice.
2. Prikaz popisa udruge.
3. Prikaz profila udruge i popis pasa.
4. Prikaz forme za rezervaciju i prikaz odabranog datuma i vremenskih intervala šetnje.
5. Prikaz poruke uspješne rezervacije i profila udruge.

Rezultat: Očekivani rezultat [4.] nije zadovoljen jer je isteklo implicitno vrijeme čekanja. Zbog toga se očekivani rezultat [5.] nije odradio. **Aplikacija nije prošla test.**

Log	Reference
Running 'Rezervacija šetnje'	
1. open on /home OK	
2. setWindowSize on 1043x824 OK	
3. click on id=2 OK	
4. click on css=.ant-table-row:nth-child(1) span OK	
5. Trying to find xpath=//div[@id='button']/button/span... OK	
6. Trying to find css=.ant-picker-focused input... OK	
Warning Element found with secondary locator xpath=//input[@value='']][7]. To use it by default, update the test step to use it as the primary locator.	
7. Trying to find css=.ant-picker-time-panel-column:nth-child(1) > .ant-picker-time-panel-cell:nth-child(9) > .ant-picker-time-panel-cell-inner... OK	
8. click on css=.ant-btn-sm > span OK	
9. mouseOver on css=.ant-btn-sm > span OK	
10. mouseOut on css=.ant-btn-sm > span OK	
11. click on css=.ant-picker-focused input OK	
12. Trying to find css=div:nth-child(11) .ant-picker-time-panel-column:nth-child(1) > .ant-picker-time-panel-cell:nth-child(10) > .ant-picker-time-panel-cell-inner:nth-child(1)... Failed: Implicit Wait timed out after 30000ms	
'Rezervacija šetnje' ended with 1 error(s)	

Slika 5.9: Prikaz rezultata 1. ispitnog slučaja u Selenium-u

Ispitni slučaj 2: Otkazivanje šetnje od strane udruge

Ulaz:

1. Otvaranje početne stranice u web pregledniku.
2. Otvaranje profila udruge.
3. Odabir popisa rezervacija udruge.
4. Pritisak kursomom na gumb OTKAŽI.

Očekivani rezultat:

1. Prikaz naslovne stranice.
2. Prikaz profila udruge i popis pasa.
3. Prikaz popisa rezervacija na profilu udruge.
4. Prikaz profila udruge i popis pasa.

Rezultat: Aplikacija je prošla test.

Log	Reference
Running 'Otkazivanje rezervacije od strane udruge'	
1. open on /home OK	
2. setWindowSize on 1552x840 OK	
3. click on id=2 OK	
4. click on id=5 OK	
5. click on css=#reserve-tab > .nav-value OK	
6. mouseOver on css=.MuiTableRow-root:nth-child(1) .MuiButton-label OK	
7. click on css=.MuiTableRow-root:nth-child(1) .MuiButton-label OK	
'Otkazivanje rezervacije od strane udruge' completed successfully	

Slika 5.10: Prikaz rezultata 2. ispitnog slučaja u Selenium-u

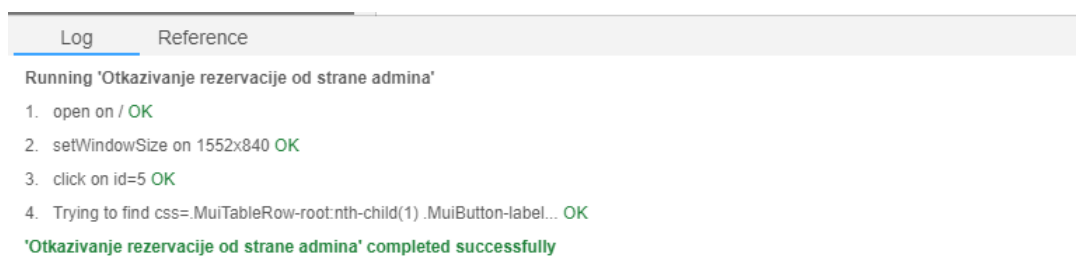
Ispitni slučaj 3: Otkazivanje šetnje od strane admina**Ulaz:**

1. Otvaranje početne stranice u web pregledniku.
2. Otvaranje admin profila.
3. Pomicanje ekrana do prikaza popisa rezervacija.
4. Pritisak kursorom na gumb OTKAŽI.

Očekivani rezultat:

1. Prikaz naslovne stranice.
2. Prikaz admin profila.
3. Prikaz popisa rezervacija na admin profilu.
4. Prikaz ažuriranog popisa rezervacija na admin profilu.

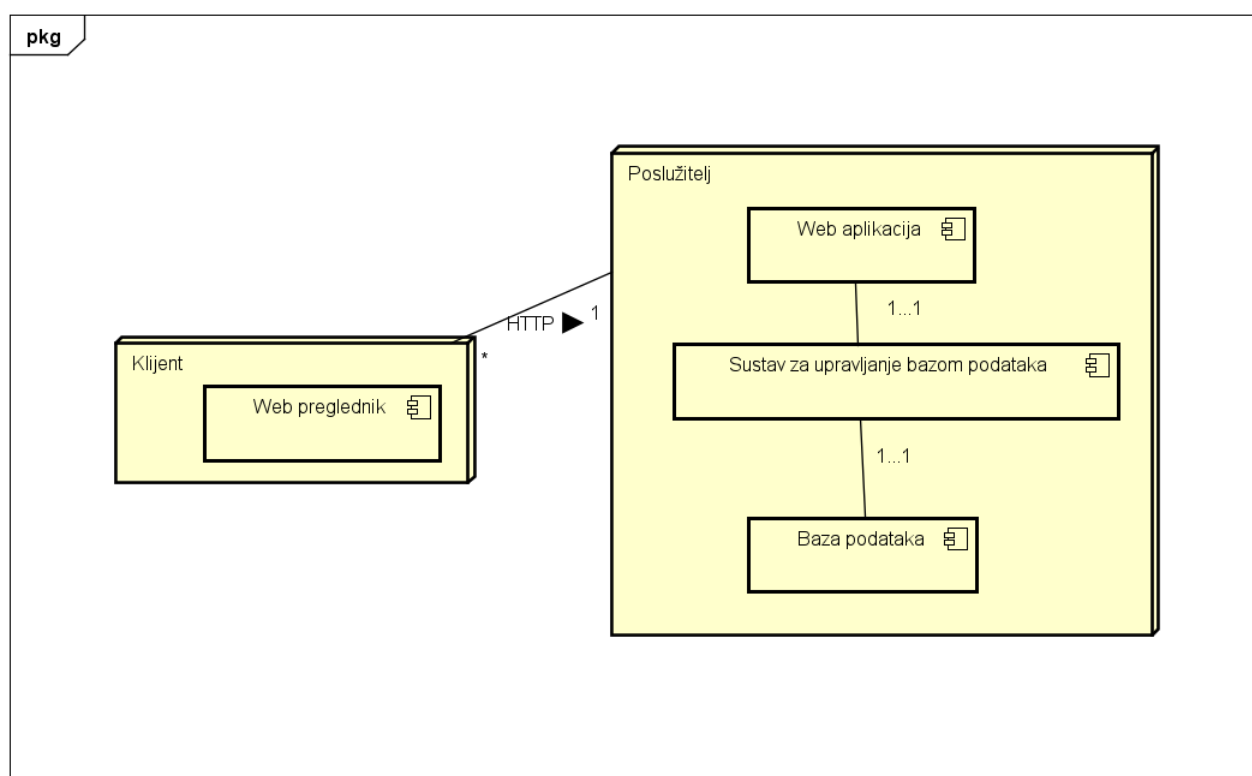
Rezultat: Aplikacija je prošla test.



Slika 5.11: Prikaz rezultata 3. ispitnog slučaja u Selenium-u

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Arhitektura sustava DogGO aplikacije je "klijent – poslužitelj" gdje web poslužitelj i poslužitelj baze podataka se nalaze na poslužiteljskom računalu a klijenti koriste web preglednik kako bi pristupili web aplikaciji. Komunikacija između računala korisnika i poslužitelja odvija se preko HTTP veze.



Slika 5.12: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Puštanje aplikacije u pogon možemo podijeliti na 3 glavna koraka

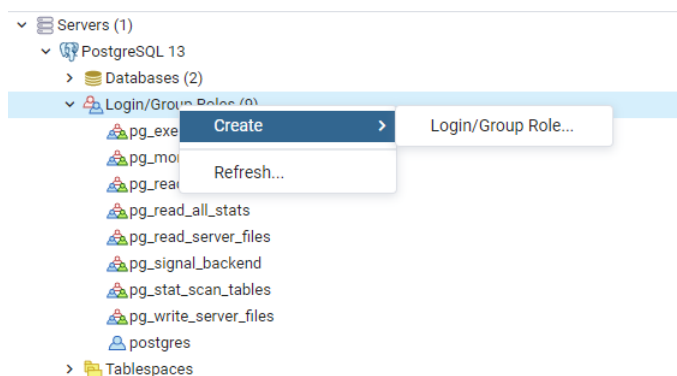
- pokretanje Postgres baze
- pokretanje backenda (Spring Boot)
- pokretanje frontenda (npm, React)

5.4.1 Pokretanje baze

Za pokretanje baze prvo je potrebno skinuti Postgres bazu. Preuzmemo installer sa <https://www.postgresql.org/download/> (verzija 13), te slijedimo upute za instalaciju. Zatim preuzimamo i instaliramo PgAdmin (<https://www.pgadmin.org/download/>), koji nam olakšava upravljanje bazom i pregled podataka.

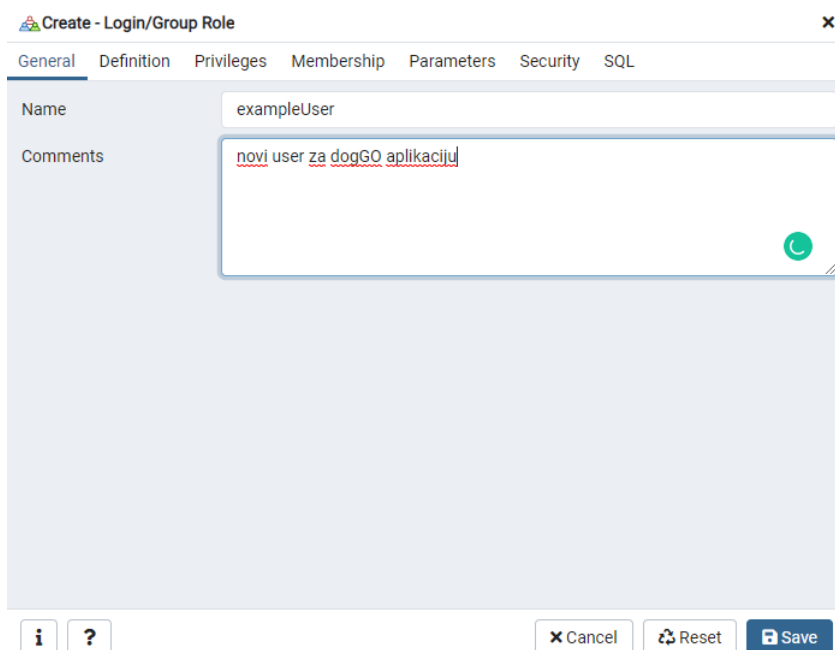
Nakon instalacije navedenih komponenti, moramo još kreirati bazu, odnosno korisničko ime koje ćemo koristiti za spajanje na bazu.

1) Otvorimo pgAdmin. Prilikom prvog pokretanja, aplikacija nas traži upisivanje glavne lozinke, koji smislimo i upišemo. Zatim klikom na Login/Group Roles -> Create otvaramo izbornik za stvaranje novog korisnika.



Slika 5.13: Kreiranje usera

U odgovarajuća polja upišemo željeno korisničko ime i password (u primjeru korišten exampleUser/password123)



Create - Login/Group Role

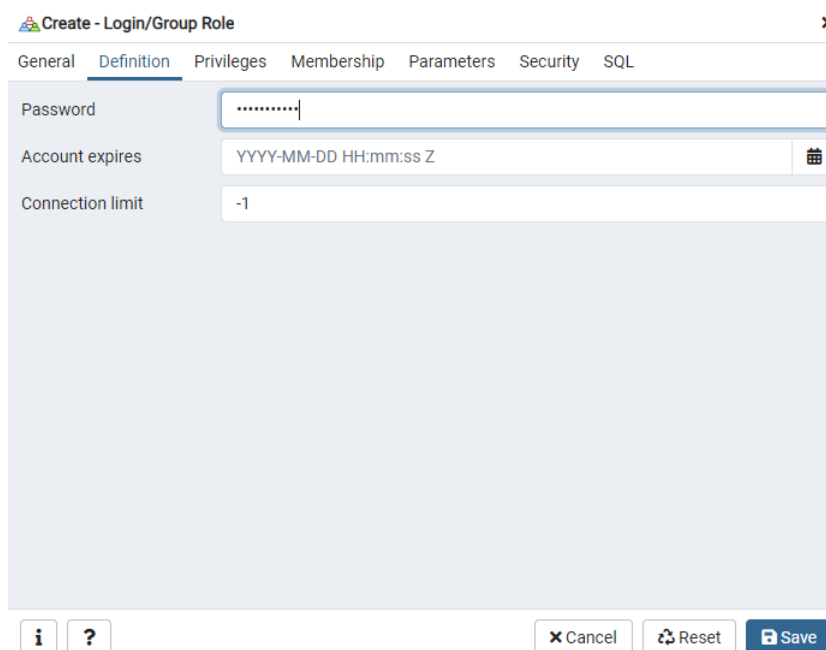
General Definition Privileges Membership Parameters Security SQL

Name: exampleUser

Comments: novi user za dogGO aplikaciju

Buttons: Cancel, Reset, Save

Slika 5.14: Kreiranje usera - upis korisničkog imena



Create - Login/Group Role

General Definition Privileges Membership Parameters Security SQL

Password:

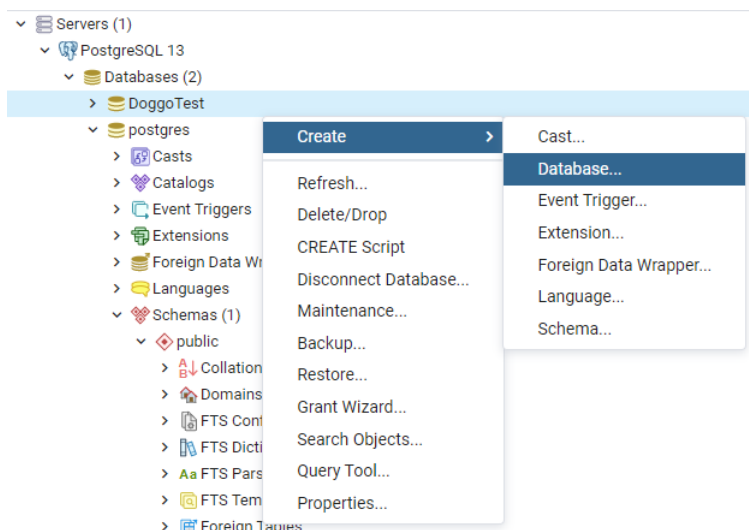
Account expires: YYYY-MM-DD HH:mm:ss Z

Connection limit: -1

Buttons: Cancel, Reset, Save

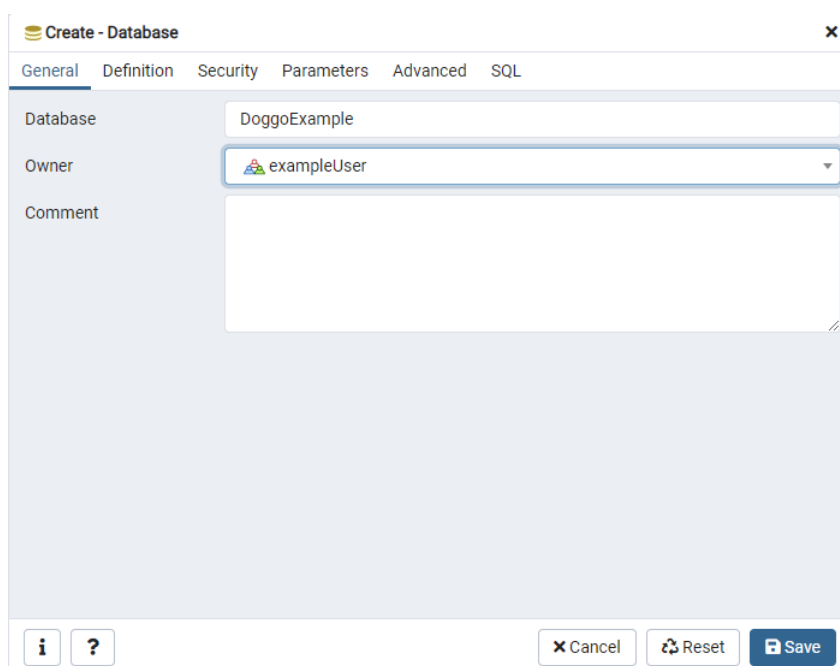
Slika 5.15: Kreiranje usera 3 - upis lozinke

Nakon što smo stvorili korisnika, možemo stvoriti i novu bazu.



Slika 5.16: Izbornik za stvaranje baze

Jedino što je potrebno upisati je ime baze (u primjeru DoggoExample), te vlasnika baze - novostvoreni korisnik exampleUser.



Slika 5.17: Izbornik za stvaranje baze - upis imena i vlasnika

5.4.2 Pokretanje backenda

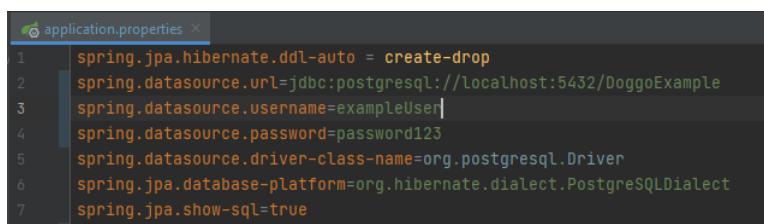
Za ispravno pokretanje backenda, prvo moramo instalirati sljedeće aplikacije

- alat za upravljanje kodom Maven (najnovija trenutačna verzija). Ukoliko se

backend pokreće iz naredbenog retka sa naredbom mvnw, instalacija nije potrebna (tj. mvnw je maven wrapper, program koji sam preuzima i privremeno instalira potrebnu verziju Mavena)

- za instalaciju slijedimo upute na <https://maven.apache.org/install.html>
- Java verzije 15 ili novija
 - za instalaciju slijedimo upute na <https://docs.oracle.com/en/java/javase/15/install/overview-jdk-installation.html>
- (opcionally) neki IDE (IntelliJ/Eclipse)

Nakon preuzimanja svih potrebnih aplikacija, prvo moramo podesiti spajanje na našu novostvorenu bazu. U nekom uređivaču teksta ili IDE-u otvorimo datoteku IzvorniKod/backend/src/main/java/resources/application.properties, te tamo unesemo sljedeće podatke:

A screenshot of a code editor showing the content of the application.properties file. The text is as follows:

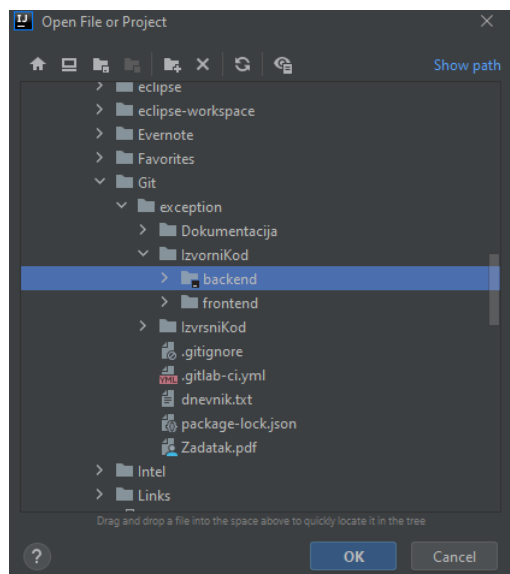
```
1 spring.jpa.hibernate.ddl-auto = create-drop
2 spring.datasource.url=jdbc:postgresql://localhost:5432/DoggoExample
3 spring.datasource.username=exampleUser
4 spring.datasource.password=password123
5 spring.datasource.driver-class-name=org.postgresql.Driver
6 spring.jpa.database-platform=org.hibernate.dialect.PostgreSQLDialect
7 spring.jpa.show-sql=true
```

Slika 5.18: application.properties

U ovom trenutku imamo sve spremno za pokretanje backenda, što možemo učiniti na 2 načina:

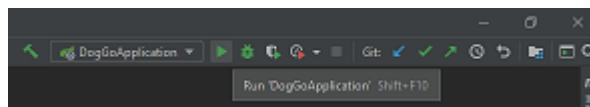
a) Korištenjem naredbenog retka (command line) Za pokretanje iz naredbenog retka korištenjem Windows powershella/Linux terminala ili slično pozicioniramo se u direktorij IzvorniKod/backend/ te tamo pokrenemo naredbu **mvn clean install**. Nakon dovršenja ove naredbe, iz istog direktorija pokrenemo i **mvn spring-boot:run**. Aplikacija se pokrenula na URL-u <http://localhost:8080>.

b) Korištenjem IDE-a (IntelliJ) Za pokretanje aplikacije iz IDE-a, prvo moramo otvoriti naš projekt u IntelliJ-u. Klikom na File/Open otvara nam se izbornik u kojem pronađemo naš projekt te u njemu odaberemo folder backend.



Slika 5.19: Otvaranje projekta u IntelliJ-u

Nakon otvaranja projekta, u gornjem desnom uglu pronađemo zelenu ikonicu u obliku trokuta, te ju kliknemo. Klikom na nju, aplikacija se pokreće na URL-u <http://localhost:8080>.



Slika 5.20: Pokretanje aplikacije iz IntelliJ-a

5.4.3 Pokretanje frontenda

Za pokretanje iz naredbenog retka potrebno se pozicionirati u direktorij IzvorniKod/frontend/app/src i u njemu izvesti sljedeće naredbe:

npm install

Naredba instalira sve node module potrebne za pokretanje aplikacije.

npm start

Pokreće aplikaciju na <http://localhost:3000>. Stranica se ponovno učitava svakom spremljenom promjenom u kodu.

6. Zaključak i budući rad

Primarni cilj ovog projekta bio je razviti aplikaciju koja omogućuje građanima da ugovore šetnje pasa u skloništima za pse i tako pomognu u socijalizaciji pasa i povećaju vjerojatnost udomljavanja. Iz projektnog zadatka bilo je potrebno izlučiti funkcionalne i nefunkcionalne zahtjeve, konceptualno osmisliti i dokumentirati aplikaciju a zatim provesti implementaciju osmišljenog.

Provedba projekta bila je podijeljena u dva ciklusa. U prvom ciklusu oformili smo projektni tim, uspostavili kanale komunikacije i započeli s radom. Bilo nam je važno upoznati se i pokušati stvoriti dobru radnu atmosferu, usprkos tome što je komunikacija uživo bila otežana. Za brzu komunikaciju smo odabrali WhatsApp, a organizaciju projekta radili smo na stranici Trello koja podržava kanban metodu organizacije. Vrlo intenzivnim fokusom na razradu zahtjeva odmah na početku rada, stvorili smo čvrstu bazu za daljnji rad i osigurali da je svim članovima tima jasna slika aplikacije koju izrađujemo.

U drugom ciklusu veći naglasak bio je na samoj implementaciji aplikacije. S obzirom na to da su svim članovima tima tehnologije s kojima smo radili bile dosad nepoznate, u drugom ciklusu se pokazala posvećenost i inicijativa članova. Svi članovi su samostalno istraživali tehnologije i tražili rješenja za probleme koji su se pojavljivali. Također je do izražaja došla važnost komunikacije i surađivanja. Nakon nekoliko nesporazuma i udvostručavanja odrađenog posla, stavili smo veći naglasak na poboljšanje komunikacije i učestalo izvještavanje o problemima koje treba riješiti za napredak. Počeli smo koristiti Discord koji nam je omogućio da više timova radi istovremeno na različitim komponentama i međusobno se brzo posavjetuje.

Radom na projektu svi smo stekli nova znanja i zato bismo da sada krećemo s projektom neke probleme vjerojatno riješili na brži i kvalitetniji način. U budućoj nadogradnji aplikacije vjerojatno bi započeli s refaktoriranjem pojedinih dijelova koda koji su nekonzistentni ili smo iskustvom rada na drugim komponentama uvidjeli da se mogu bolje riješiti. Osim toga, kao moguće proširenje funkcionalnosti aplikacije dodali bismo mogućnost grupnih šetnja s drugim ljudima, što mislimo da bi bilo zanimljivo i za prijatelje i za upoznavanje novih ljudi.

Svim članovima tima je ovaj projekt pružio nova znanja i vještine s kojima se dosad nismo susreli na FER-u. Osim tehničkih znanja iz korištenih tehnologija, stekli smo vrijedno iskustvo timskog rada i općenito samostalnog rada na projektu.

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. L. Shklar, R. Rosen, "Web Application Architecture: Principles, Protocols, And Practices", 1st ed, John Wiley & Sons Ltd, 2003.
8. Službena dokumentacija Spring razvojnog okvira, <https://docs.spring.io/spring-framework/docs/current/reference/html/>
9. Spring Boot backend demo aplikacija, CROZ d.o.o., <https://gitlab.com/hrvojesimic/progi-project-teams-backend>
10. Izrada frontenda u Reactu, CROZ d.o.o., <https://gitlab.com/jtomic/opp-project-teams-frontend/-/blob/master/education.md>
11. Online platforma Baeldung, <https://www.baeldung.com/>
12. Online platforma Spring.io, <https://spring.io/>

Indeks slika i dijagrama

2.1	Rezultat anketnog pitanja o interesu za aplikaciju	10
2.2	Rezultat anketnog pitanja o imenu aplikacije	11
3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranog korisnika i šetača	23
3.2	Dijagram obrasca uporabe, funkcionalnost udruge i administratora	24
3.3	Sekvencijski dijagram za registraciju korisnika	25
3.4	Sekvencijski dijagram za promjenu osobnih podataka korisnika	26
3.5	Sekvencijski dijagram za rezerviranje šetnji	27
4.1	Arhitektura sustava	29
4.2	MVC arhitektura	30
4.3	Konceptualno hibridna arhitektura Spring razvojnog okvira	31
4.4	Dijagram baze podataka	36
4.5	Dijagram domenskih razreda	37
4.6	Dijagram spremišnih razreda	38
4.7	Nasljeđivanje JPA repozitorija	38
4.8	Dijagram servisnih sučelja	39
4.9	Dijagram razreda implementacije servisnih sučelja	39
4.10	Dijagram razreda kontrolera	40
4.11	Ishodišni paket aplikacije	40
4.12	Dijagram razreda objekata prijenosa podataka	41
4.13	Dijagram stanja	42
4.14	Dijagram aktivnosti za registraciju korisnika	43
4.15	Dijagram aktivnosti za registraciju udruga	43
4.16	Dijagram aktivnosti za pregled svih udruga	44
4.17	Dijagram aktivnosti za pregled statistike šetača	44
4.18	Dijagram aktivnosti za promjenu podataka korisničkog računa	45
4.19	Dijagram aktivnosti za dodavanje pasa kod udruga	46
4.20	Dijagram aktivnosti za brisanje pasa kod udruga	46
4.21	Dijagram aktivnosti za rezervaciju šetnji	47

4.22	Dijagram aktivnosti za pregled rezervacija šetača	48
4.23	Dijagram aktivnosti za pregled rezervacija udruge	49
4.24	Dijagram aktivnosti za brisanje rezervacije kod udruge	50
4.25	Dijagram aktivnosti za pregled dostupnih podataka za admina . . .	50
4.26	Dijagram komponenti	52
5.1	Prikaz inicijalizacije testnih podataka	55
5.2	Prikaz 1. Junit testa	56
5.3	Prikaz 2. Junit testa	56
5.4	Prikaz 3. Junit testa	57
5.5	Prikaz 4. Junit testa	57
5.6	Prikaz 5. Junit testa	58
5.7	Prikaz 6. Junit testa	58
5.8	Prikaz rezultata testova	59
5.9	Prikaz rezultata 1. ispitnog slučaja u Selenium-u	60
5.10	Prikaz rezultata 2. ispitnog slučaja u Selenium-u	60
5.11	Prikaz rezultata 3. ispitnog slučaja u Selenium-u	61
5.12	Dijagram razmještaja	62
5.13	Kreiranje usera	63
5.14	Kreiranje usera - upis korisničkog imena	64
5.15	Kreiranje usera 3 - upis lozinke	64
5.16	Izbornik za stvaranje baze	65
5.17	Izbornik za stvaranje baze - upis imena i vlasnika	65
5.18	application.properties	66
5.19	Otvaranje projekta u IntelliJ-u	67
5.20	Pokretanje aplikacije iz IntelliJ-a	67
6.1	Doprinosi dev-u	78
6.2	Doprinosi devdoc-u	78

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.

1. sastanak

- Datum: 8. listopada 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - upoznavanje
 - određivanje kanala komunikacije
 - odabir teme
 - odabir tehnologija

2. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - definiranje funkcionalnosti
 - raspodjela za pisanje dokumentacije
 - tim za uspostavu baze podataka
 - naziv aplikacije

3. sastanak

- Datum: 22. listopada 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - dodatne razrade funkcionalnosti

- podjela u timove za backend i frontend
- raspored stranica

4. sastanak

- Datum: 3. studeni 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - predstavljanje obavljenog na frontendu i backendu i dogovor oko nastavka rada
 - podjela posljednjih zadataka prije prve predaje
 - razmjena informacija o tehnologijama

5. sastanak

- Datum: 10. studeni 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - predstavljanje obavljenog na frontendu i backendu
 - dogovaranje oko posljednjih promjena prije prve predaje

6. sastanak

- Datum: 29. studeni 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - prolazak kroz logiku aplikacije i priprema za kolokviranje prvog ciklusa
 - bilježenje stvari koje treba popraviti u nastavku rada

7. sastanak

- Datum: 7. prosinac 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - bilježenje posla za drugi ciklus
 - dodatna razrada nekih funkcionalnosti
 - okvirna podjela posla

8. sastanak

- Datum: 28. prosinac 2020.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - predstavljanje napravljenog
 - rješavanje nejasnoća
 - dodatna podjela posla za nastavak rada

9. sastanak

- Datum: 5. siječnja 2021.
- Prisustvovali: A.Terović, A.Miletić, D.Smoljan, E.Jagodić, F.Sentinella-Jerbić, I.Pezo, K.Golub
- Teme sastanka:
 - rasprava o napravljenom
 - potres usred sastanka
 - završna podjela posla

Tablica aktivnosti

Kontinuirano osvježavanje

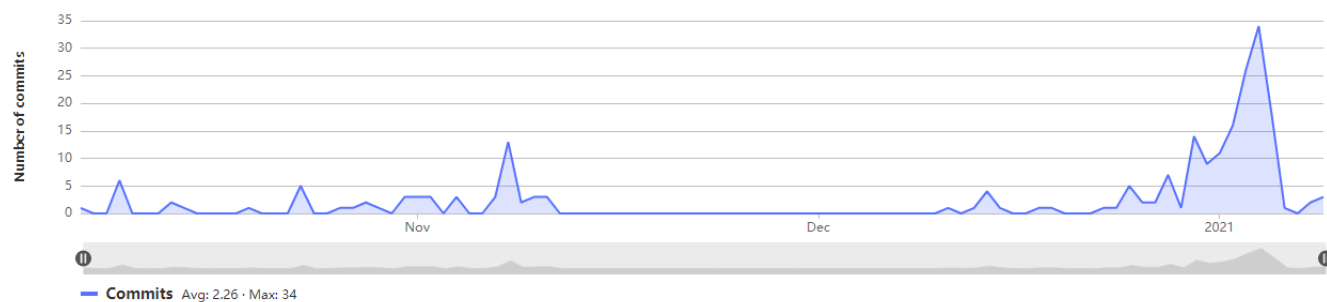
Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	Eva Jagodić	Ana Terović	Antonio Miletić	Dorian Smoljan	Fani Sentinella-Jerbić	Iva Pezo	Katarina Golub
Upravljanje projektom	24						
Opis projektnog zadatka	2			5	2		
Funkcionalni zahtjevi					5	4	
Opis pojedinih obrazaca					3	3	
Dijagram obrazaca	4	2					
Sekvencijski dijagrami			4				2
Opis ostalih zahtjeva					1		
Arhitektura i dizajn sustava					9		
Baza podataka	2	3	20		1		4
Dijagram razreda					24		4
Dijagram stanja						4	
Dijagram aktivnosti		6					
Dijagram komponenti				2			
Korištene tehnologije i alati					3		
Ispitivanje programskog rješenja			20				7
Dijagram razmještaja		0.5					
Upute za puštanje u pogon				3			
Dnevnik sastajanja	1						
Zaključak i budući rad	2						
Popis literature					1		
Revizija dokumentacije	2					2	

	Eva Jagodić	Ana Terović	Antonio Miletić	Dorian Smoljan	Fani Sentinella-Jerbić	Iva Pezo	Katarina Golub
<i>Izrada početne stranice</i>	1					6	
<i>Deploy sustava</i>	2			20			
<i>Istraživanje tehnologija</i>	10	6	15		20	7	3
<i>Back end</i>	80			45	60		50
<i>Front end</i>	1	100	150			100	
<i>Integracija</i>	18	10	15				

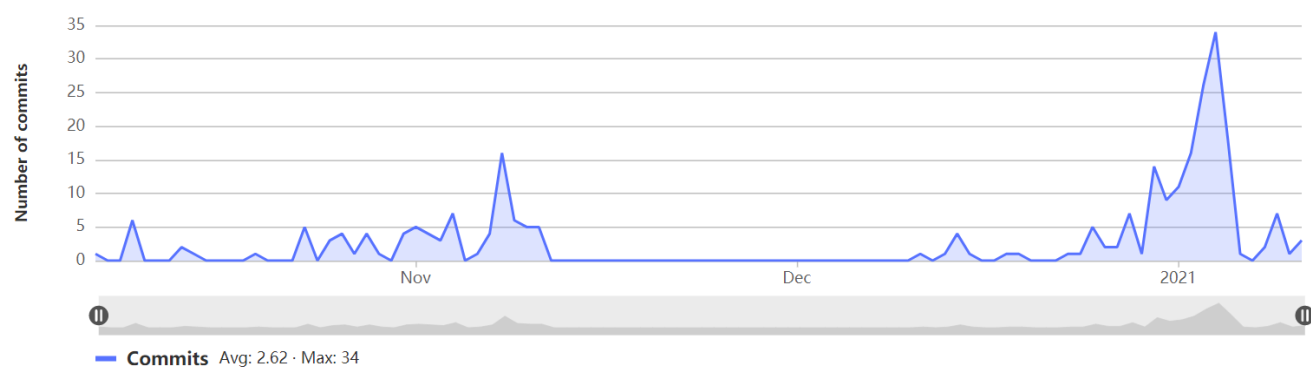
Dijagrami pregleda promjena

Excluding merge commits. Limited to 6,000 commits.



Slika 6.1: Doprinosi dev-u

Excluding merge commits. Limited to 6,000 commits.



Slika 6.2: Doprinosi devdoc-u