

Programsko inženjerstvo

Ak. god. 2020./2021.

Humanitarni šetači pasa

Dokumentacija, Rev. 2

Grupa: *OneClick*

Voditelj: *Benjamin Horvat*

Datum predaje: *14. 1. 2021.*

Nastavnik: *Doria Bukić*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
2.1 Opis problema i cilj projekta . . . . .	5
2.2 Funkcionalnosti . . . . .	5
<b>3 Specifikacija programske potpore</b>	<b>7</b>
3.1 Funkcionalni zahtjevi . . . . .	7
3.1.1 Obrasci uporabe . . . . .	9
3.1.2 Sekvencijski dijagrami . . . . .	16
3.2 Ostali zahtjevi . . . . .	19
<b>4 Arhitektura i dizajn sustava</b>	<b>20</b>
4.1 Baza podataka . . . . .	21
4.1.1 Opis tablica . . . . .	21
4.1.2 Dijagram baze podataka . . . . .	25
4.2 Dijagram razreda . . . . .	26
4.3 Dijagram stanja . . . . .	31
4.4 Dijagram aktivnosti . . . . .	32
4.5 Dijagram komponenti . . . . .	34
<b>5 Implementacija i korisničko sučelje</b>	<b>35</b>
5.1 Korištene tehnologije i alati . . . . .	35
5.2 Ispitivanje programskog rješenja . . . . .	38
5.2.1 Ispitivanje komponenti . . . . .	38
5.2.2 Ispitivanje sustava . . . . .	48
5.3 Dijagram razmještaja . . . . .	55
<b>6 Zaključak i budući rad</b>	<b>56</b>
<b>Popis literature</b>	<b>58</b>

<b>Indeks slika i dijagrama</b>	<b>60</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>61</b>

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Promijenjena naslovna stranica.	Horvat	15.10.2020.
0.2	Upisani dionici.	Mužević	22.10.2020.
0.3	Upisani <i>Use Case</i> vi: 1-4, 6, 6.1, 7 i 7.1. Upisani aktori i njihovi funkcionalni zahtjevi.	Čovran Doljanin Mužević	28.10.2020.
0.4	Upisani <i>Use Case</i> vi: 5, 8, 9, 10.	Doljanin	02.11.2020.
0.5	UML i sekvencijski dijagrami. Baza podataka i ER dijagram. UC10 promijenjen u UC10.1 te dodan novi UC10.	Horvat	06.11.2020.
0.6	Opis projektnog zadatka	Zekić	10.11.2020.
0.7	Dijagrami razreda. Izbrisane smjernice. Predgovor za arhitekturu sustava. Popravak UC1.	Horvat, Mirković, Doljanin	11.11.2020.
0.8	Upis broja sati. Popravak dokumentacije. Upisan dnevnik sastajanja.	Zekić, Mirković, Doljanin, Čovran, Rodek	12.11.2020.
1.0	Ažurirani dijagrami razreda. Zadnji popravci.	Horvat	13.11.2020.
1.1	Dijagram stanja. Dijagram aktivnosti. Dijagram komponenti. Dijagram razmještaja. Dijagrami razreda.	Mirković	10.1.2021.
1.2	Testiranje.	Mirković, Horvat	13.1.2021.

Rev.	Opis promjene/dodatka	Autori	Datum
1.3	Popravci dokumentacije. Zaključak i budući rad. Puštanje u pogon. Sastanci. Tablica aktivnosti i dijagrami pregleda promjena.	Mirković, Doljanin, Zekić, Čovran, Rodek, Mužević	14.1.2021.
2.0	Završna verzija.	Mirković	14.1.2021.

## 2. Opis projektnog zadatka

### 2.1 Opis problema i cilj projekta

U današnjem svijetu nezbrinute i napuštene životinje nisu stran pojam. Procijenjeno je kako u Hrvatskoj gotovo 10.000 životinja nema svoj dom. Nasreću, postoje udruge i ustanove koje napuštenim životinjama pružaju osnovnu njegu i toplinu. Te udruge i skloništa spašavaju ranjene i nezbrinute životinje, no često su im kapaciteti popunjeni, a radnici imaju previše posla. S motivacijom da se s jedne strane pomogne radnicima pri brizi za životinje, a s druge strane potakne građane na angažiranost i udomljavanje životinja, nastala je ideja za aplikaciju „Humanitarni šetači pasa“. Glavni je zadatak aplikacije povezati udruge za životinje s građanima koji imaju želju i vrijeme za šetanje pasa te time povećati izgled udomljavanja pasa i psihološkog efekta dobiti socijalizacije za psa i za čovjeka.

### 2.2 Funkcionalnosti

Pokretanjem aplikacije otvara se naslovna stranica na kojoj je prikazano zaglavlje te popis svih registriranih udruga.

Neregistrirani korisnik ima mogućnost pregleda popisa te pretraživanja udruga po nazivu ili lokaciji. Odabirom pojedine udruge otvaraju se detalji njezina profila. U sklopu profila udruge korisnik može dobiti informacije o psima (slike i opisi, raspoloživost za šetnju u određenom vremenskom periodu (datum i vrijeme) te jesu li psi predodređeni za skupne ili za individualne šetnje). Dodatno, prilikom pregleda profila korisniku je dostupna i rang lista registriranih šetača poredanih s obzirom na broj šetnji, broj pasa te duljinu šetnje koju su odradili u posljednjih mjesec dana. Također, dostupne su mu informacije na kojoj se lokaciji nalazi udruge, statistika o šetnjama svih pasa te mogućnost da se prijavi za šetnju pasa. Ako se korisnik odluči na potonju opciju, ima mogućnost prijave u sustav ili registracije ukoliko još nema račun.

Prilikom registracije korisnik bira želi li se prijaviti kao građanin ili kao udruge.

Za registraciju građanina potrebno je unijeti:

- korisničko ime
- e-mail
- ime i prezime
- lozinka
- broj mobitela

Za registraciju udruge potrebno je unijeti:

- naziv
- OIB udruge
- korisničko ime
- adresu
- mjesto
- e-mail
- lozinku

Registracijom u sustav kao građanin, korisniku se otvara mogućnost prijave za šetnju, pregledavanja osobnih podataka, vlastite statistike šetnji te pregledavanja i preuzimanja vlastitog rasporeda u PDF formatu. On svoju statistiku šetnji može označiti javnom kako bi ona dospjela na rang listu na javnoj stranici.

Udruga registracijom, osim mogućnosti prijave u sustav te pregleda vlastitih osobnih podataka, dobiva i mogućnost mijenjanja profila životinja.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

Dionici:

1. Administrator
2. Građanin
3. Javni posjetitelj
4. Razvojni tim
5. Udruga za životinje
6. Životinja

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
  - (a) Pristupiti naslovnoj stranici na kojoj može pregledati sve udruge
  - (b) Pristupiti detaljima profila udruge za životinje
    - i. Dobiti uvid u profile pasa
    - ii. Dobiti uvid u statistike
  - (c) Pristupiti rang listi
  - (d) Imati mogućnost registracije kao građanin ili udruga za životinje
2. Građani (inicijator) mogu:
  - (a) Imati sve mogućnosti neprijavljenog/neregistiranog korisnika
  - (b) Prijaviti se u vlastiti profil
  - (c) Pregledati svoj raspored šetnji i svoje statistike šetnji
  - (d) Učiniti svoje statistike šetnji javnima
  - (e) Odabrati psa i željeni termin šetnje
  - (f) Preuzeti svoj raspored šetnji kao PDF
  - (g) Mijenjati podatke o svom profilu
3. Udruge za životinje (inicijator) mogu:



- (a) Prijaviti se u vlastiti profil
- (b) Mijenjati podatke o svom profilu
- (c) Pristupiti naslovnoj stranici na kojoj može pregledati sve udruge
- (d) Pristupiti profilima životinja i udruga za životinje

4. Administrator (inicijator) može:

- (a) Upravljači profilima građana, udruga i životinja
- (b) Upravljači šetnjama

5. Baza podataka (sudionik)

- (a) Dohvaćati podatke
- (b) Pohranjivati podatke

### 3.1.1 Obrasci uporabe

#### UC1 - Registracija u sustav

- **Glavni sudionik:** Javni posjetitelj
- **Cilj:** Registracija u sustav kao građanin ili udruga za životinje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisničko ime i e-mail ne smiju biti već iskorišteni
- **Opis osnovnog tijeka registracije udruge:**
  1. Odabir registracije udruge
  2. Unos korisničkog imena
  3. Unos e-mail adrese
  4. Unos lozinke
  5. Unos naziva udruge
  6. Unos broja telefona udruge
  7. Unos OIB-a udruge
  8. Unos adrese udruge
  9. Kliknuti gumb "Registracija"
- **Opis osnovnog tijeka registracije građanina:**
  1. Odabir registracije građanina
  2. Unos korisničkog imena
  3. Unos lozinke
  4. Unos imena
  5. Unos prezimena
  6. Unos e-mail adrese
  7. Unos broja telefona
  8. Kliknuti gumb "Registracija"
- **Opis mogućih odstupanja:**
  - 1.a Korisničko ime već postoji u sustavu
    1. Sustav obavještava korisnika da unese drugo korisničko ime
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije
  - 2.a E-mail već postoji u sustavu
    1. Sustav obavještava korisnika da unese drugi e-mail
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

**UC2 - Prijava u sustav**

- **Glavni sudionik:** Građanin ili udruga
- **Cilj:** Prijava u sustav kao građanin ili udruga
- **Sudionici:** Baza podataka
- **Preduvjet:** Građanin ili udruga moraju biti registrirani u sustav
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena
  2. Unos lozinke
  3. Kliknuti gumb "Prijava"
- **Opis mogućih odstupanja:**
  - 1.a Korisničko ime ne postoji u sustavu
    1. Sustav obavještava korisnika da je uneseno nepostojeće korisničko ime
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od prijave
  - 2.a Korisnik je unio krivu lozinku za navedeno korisničko ime
    1. Sustav obavještava korisnika da unese ispravnu lozinku
    2. Korisnik unosi ispravnu lozinku ili odustaje od prijave

**UC3 - Pregled naslovne stranice**

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled liste udruga
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji
- **Opis osnovnog tijeka:**
  1. Otvaranje aplikacije ili klik na gumb "Početna stranica"
  2. Pregled dostupnih podataka

**UC3.1 - Pregled detalja udruga**

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz detalja udruga
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Odabir željene udruge na naslovnoj stranici

## 2. Prikaz podataka udruge

### UC4 - Pretraživanje udruga

- **Glavni sudionik:** Korisnik
- **Cilj:** Pronaći udrugu preko upisa imena ili lokacije
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Odabrati gumb "Pretraživanje" i upisati u polje za pretraživanje
  2. Odabrati željenu udrugu

### UC5 - Prijava šetnje

- **Glavni sudionik:** Građanin
- **Cilj:** Prijaviti šetnju sa željenom životinjom i željenim terminom
- **Sudionici:** Životinje, Baza podataka
- **Preduvjet:** Prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Odabire se željeni termin i duljina šetnju
  2. Odaberu se svi željeni psi za šetnju koji su slobodni u odabranom terminu te koji odgovaraju vrsti šetnje
  3. Odabire se opcija "Prijavi šetnju"
  4. Pokaže se stranica s pregledom podataka o šetnji
  5. Potvrda šetnje
- **Opis mogućih odstupanja:**
  - 1.a Građanin se u odabranom terminu već prijavio za drugu šetnju
    1. Sustav obavještava korisnika da je u željenom terminu već zauzet te zatraži od njega promjenu termina
    2. Građanin mijenja termin ili odustaje od šetnje
  - 5.a Nema slobodnih pasa u odabranom terminu
    1. Nakon potvrde šetnje, sustav obavještava korisnika da su odabrani psi već zauzeti u željenom terminu

### UC6 - Pregled vlastitog rasporeda

- **Glavni sudionik:** Građanin
- **Cilj:** Pregledati vlastiti raspored šetnji
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Odabire se opcija “Raspored”
  2. Prikaz vlastitog rasporeda
- **Opis mogućih odstupanja:** -

#### UC6.1 - Preuzimanje vlastitog rasporeda

- **Glavni sudionik:** Građanin
- **Cilj:** Preuzeti svoj raspored šetnji
- **Sudionici:** Životinja, Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Na vlastitom profilu otvori se raspored
  2. Odabire se početni i završni datum za raspored koji se želi preuzeti
  3. Klikne se “Preuzmi raspored”
- **Opis mogućih odstupanja:** -

#### UC7 - Pregled osobnih podataka

- **Glavni sudionik:** Građanin, Udruga
- **Cilj:** Pregled osobnih podataka koji su zapisani u sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Građanin ili Udruga odabiru opciju “Profil”
  2. Prikazuju se osobni podaci
- **Opis mogućih odstupanja:** -

#### UC7.1 - Promjena osobnih podataka

- **Glavni sudionik:** Udruga, Građanin
- **Cilj:** Promjena osobnih podataka na nove ili ispravne vrijednosti
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Građanin ili Udruga pregledavaju svoje podatke
  2. Odabire se opcija “Uredi podatke”
  3. Korisnik uredi podatke

4. Odabirom opcije “Spremi podatke” se spremaju podaci

- **Opis mogućih odstupanja:**

4.a Podaci nisu ispravni

1. Sustav prepoznaje neispravne podatke te zatraži njihov ispravak

### UC8 - Pristup statistici

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregled rang liste šetača

- **Sudionici:** Baza podataka

- **Preduvjet:** -

- **Opis osnovnog tijeka:**

1. Odabire se opcija “Statistika”

2. Prikaže se statistika svih šetnji te rang liste

- **Opis mogućih odstupanja:** -

### UC9 - Pregled vlastitih statistika šetnji

- **Glavni sudionik:** Građanin

- **Cilj:** Pregled vlastitih statistika šetnji i mogućnost odabira šetnje kao javne

- **Sudionici:** Baza podataka

- **Preduvjet:** Prijavljen u sustav

- **Opis osnovnog tijeka:**

1. Na vlastitom profilu otvori se statistika šetnji

2. Odabirom opcije “Uredi podatke” za statistiku se može postaviti je li javna ili privatna

- **Opis mogućih odstupanja:** -

### UC10 - Pregled profila životinja

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregled podataka za pojedinu životinju

- **Sudionici:** Životinja, Baza podataka

- **Preduvjet:** -

- **Opis osnovnog tijeka:**

1. Odabire se udruga čije se životinje žele pregledati

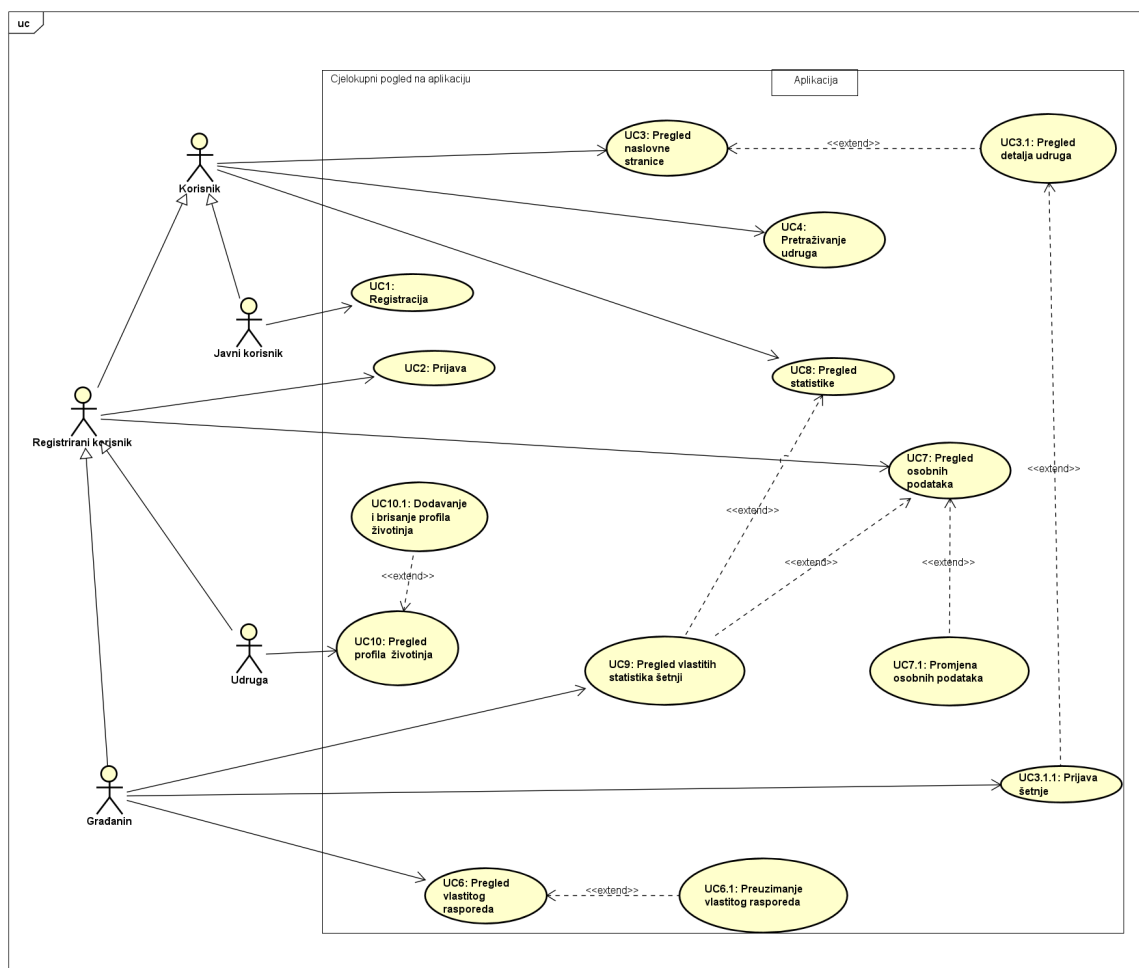
2. Prikazuju se podaci životinja

- **Opis mogućih odstupanja:** -

### UC10.1 - Dodavanje i brisanje životinja

- **Glavni sudionik:** Udruga
- **Cilj:** Dodavanje i brisanje životinja u sustavu
- **Sudionici:** Životinja, Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav kao udruga
- **Opis osnovnog tijeka:**
  1. Odabire se opcija "Uredi pse" na vlastitom profilu udruge
  2. Odabirom opcije "Izbriši životinju" pored odabrane životinje ona se briše iz sustava
  3. Odabirom opcije "Dodaj psa" otvara se forma za popunjavanje podataka o novom psu te, klikom na gumb "Dodaj", on se dodaje u sustav
- **Opis mogućih odstupanja:** Podaci se nisu uspješno spremili

## Dijagrami obrazaca uporabe



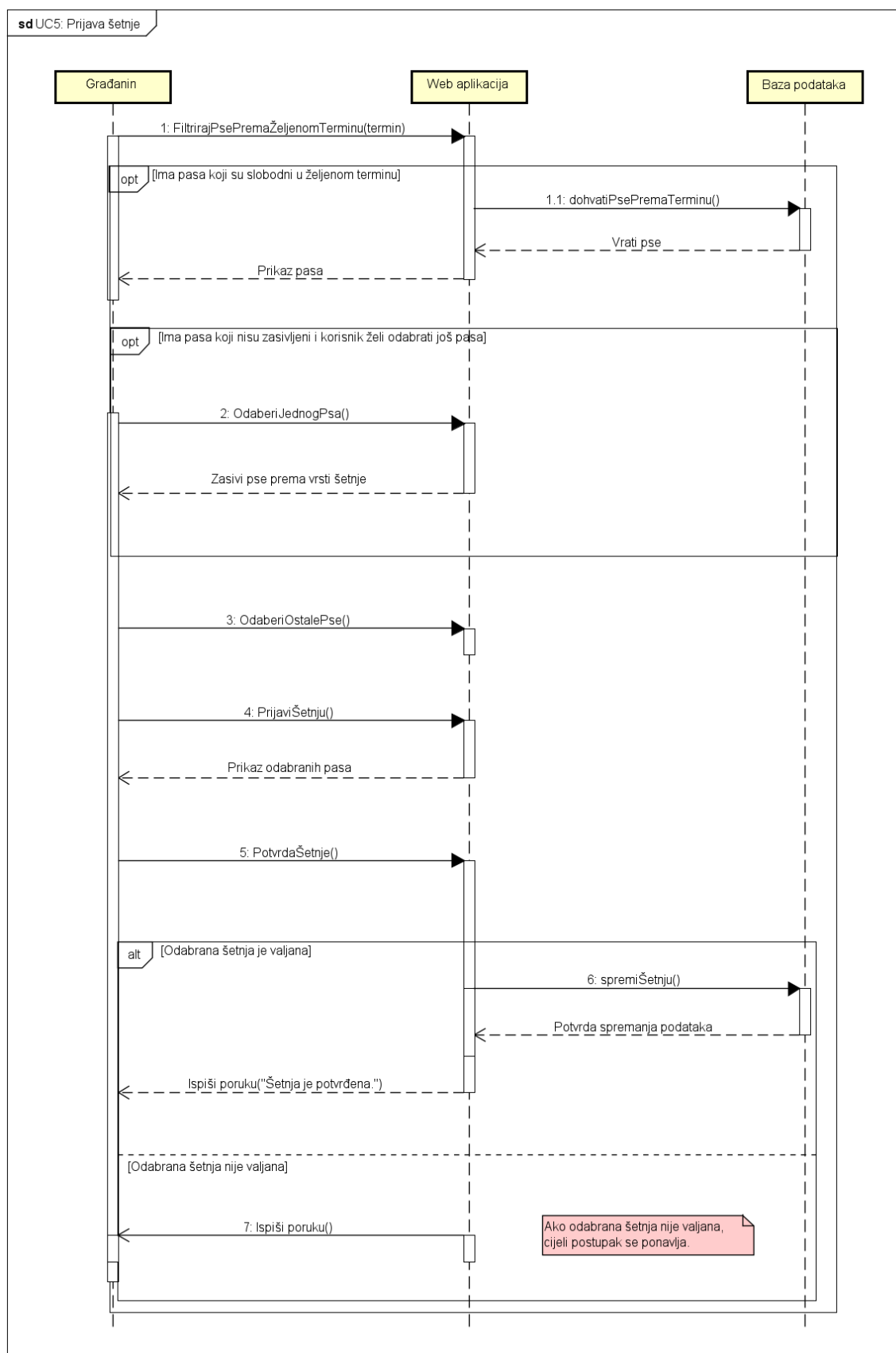
Slika 3.1: Dijagram svih obrasca uporabe



### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC5 - Prijava šetnje

Građanin pregledom Udruge te odabirom željenog termina, šalje zahtjev za prikaz dostupnih životinja u terminu. Poslužitelj dohvaća sve slobodne životinje u odabranom terminu iz baze podataka, te ih prikazuje. Ako je Građanin zadovoljan odabirom slobodnih životinja, odabire željeni oblik šetnje. Zatim se životinje filtriraju po odabranoj vrsti šetnje i prikazuju. Građanin odabire željene životinje za termin. Kada je Građanin zadovoljan s odabirom, može poslati zahtjev za potvrdu šetnje. Potvrdom šetnje se zauzima termin životinja te se prikazuje poruka "Šetnja prijavljena".



Slika 3.2: Sekvencijski dijagram za UC5

**Obrazac uporabe UC6 i UC6.1 - Pregled i preuzimanje vlastitog rasporeda**

Građanin šalje zahtjev za pregled vlastitog rasporeda. Poslužitelj dohvaća raspored te mu ga prikazuje. Građanin je u mogućnosti odabirom opcije "Preuzmi vlastiti raspored" zatražiti preuzimanje vlastitog raspored u PDF formatu.



Slika 3.3: Sekvencijski dijagram za UC6 i UC6.1

## 3.2 Ostali zahtjevi

- Sustav treba biti intuitivan i jednostavan za korištenje korisnicima koji prvi put koriste aplikaciju.
- Sustav mora omogućiti pristup i rad više korisnika istovremeno.
- Sustav treba biti implementiran kao web aplikacija i mora mu biti omogućen pristup iz javne mreže pomoću HTTPS protokola.
- Neregularne aktivnosti unutar sustava ne smiju narušiti njegov rad i daljnje normalno funkcioniranje.
- Sustav mora podržavati hrvatska slova (dijakritičke znakove).
- Baza podataka mora biti zaštićena, brza i otporna na moguće pogrešne zahtjeve.
- Dohvaćanje podataka iz baze ne smije trajati duže od nekoliko sekundi.
- Sustav treba biti moguće nadograđivati, unaprjeđivati i razvijati bez narušavanja postojećih funkcionalnosti.
- Prilikom prijave šetnje njezino trajanje mora biti zaokruženo na puni sat.
- Statistika prikazuje podatke stare do 30 dana.
- Datum mora biti u obliku dd.mm.yyyy.

## 4. Arhitektura i dizajn sustava

Projekt kao cjelina sastoji se od sljedećih podsustava:

1. Web poslužitelja
2. Web aplikacije
3. Baze podataka

**Web preglednik** je program koji omogućuje pregled web-stranica i njihovih multimedijalnih sadržaja. Web preglednik prevodi jezik kojim je Web stranica pisana u jezik razumljiv korisniku te šalje korisničke zahtjeve web poslužitelju.

**Web poslužitelj** je računalo koje pohranjuje web stranice i sadržaje potrebne za njezin prikaz (primjerice HTML dokumente, CSS stilove, JavaScript datoteke i fotografije). Njegova uloga je povezivanje web preglednika s web aplikacijom kako bi ju učinio dostupnom korisniku.

**Web aplikacija** je program koji radi na poslužitelju, a izvodi se u web pregledniku kojem pruža grafičko korisničko sučelje. Kada korisnik zatraži izvođenje nekog dijela aplikacije, web preglednik taj zahtjev šalje web poslužitelju. Aplikacija izvodi zadatak na poslužitelju i generira rezultat koji poslužitelj šalje web pregledniku, gdje je rezultat vidljiv korisniku kao odgovor na njegov upit. Web aplikacija je povezana s **bazom podataka** iz koje dohvaća podatke, a ulogu posrednika u komunikaciji između poslužiteljske i klijentske strane ima REST API.

Programski jezik koji smo odabrali za izradu naše web aplikacije je Java, zajedno s JavaSpring radnim okvirom i React front-end knjižnicom za izgradnju korisničkog sučelja te programskim jezikom TypeScript. Razvojna okruženja u kojima ćemo raditi su Eclipse/IntelliJ te Visual Studio Code.

Arhitektura našeg sustava temeljit će se na MVC (Model-View-Controller) obrascu. JavaSpring radni okvir omogućuje korištenje MVC softverske arhitekture te gotove komponente za razvoj web aplikacija. Arhitektura zasnovana na MVC obrascu pruža mogućnost odvajanja pojedinih dijelova aplikacije u zasebne komponente, što omogućuje njihov nezavisan razvoj te na taj način olakšava procese optimiranja

i održavanja aplikacije.

**MVC** obrazac softverske arhitekture sastoji se od:

1. **Model** -Sadrži podatke, logike i funkcije ugrađene u program, upravlja njima te kao takva čini središnji dio sustava.
2. **View** -Predstavlja prikaz podataka (primjerice obrazac, tablica ili dijagram). Iste podatke je moguće prikazati na više različitih načina.
3. **Controller** -Upravlja korisničkim zahtjevima tako što pruža vanjsko sučelje web aplikacije u obliku RESTful web usluge (REST API-ja), prihvata HTTP zahtjeve, poziva odgovarajuće usluge na sloju usluge te vraća odgovore klijentu u obliku JSON datoteke.

## 4.1 Baza podataka

Za pohranjivanje podataka koristimo relacijsku bazu podataka, konkretno PostgreSQL. Baza podataka sadrži sljedeće entitete:

- Korisnik
- Građanin
- Udruga
- Mjesto
- Životinja
- Vrsta
- Šetnja
- Šetnja Životinja

### 4.1.1 Opis tablica

**Korisnik** Ovaj entitet sadržava sve informacije za prijavu Građanina i Udruga u sustav. Sadrži attribute: ID, KorisničkoIme, Email, Slika, Lozinka, BrojMobitela, Javan. Ovaj entitet je u *One-To-One* vezi s Građaninom i Udrugom preko atributa ID.

Korisnik		
ID	INT	jedinstveni identifikator korisnika
Korisničko ime	VARCHAR	korisničko ime korisnika
Email	VARCHAR	e-mail adresa korisnika
Slika	LONGBLOB	slika profila korisnika
Lozinka	VARCHAR	hash lozinke
Broj mobitela	VARCHAR	broj mobitela korisnika
Javan	BOOLEAN	oznaka je li korisnički račun javan

**Gradanin** Ovaj entitet sadržava sve informacije vezane za Gradanina. Sadrži attribute: ID, ID korisnika, Ime, Prezime. Ovaj entitet je u *One-To-One* vezi s Korisnikom preko atributa ID korisnika, te u *One-To-Many* vezi sa Šetnjom preko atributa ID.

Gradanin		
ID	INT	jedinstveni identifikator građanina
ID korisnika	INT	jedinstveni identifikator korisnika (korisnik.ID)
Ime	VARCHAR	ime građanina
Prezime	VARCHAR	prezime građanina

**Udruga** Ovaj entitet sadržava sve informacije vezane za Udrugu. Sadrži attribute: ID, ID korisnika, Oib, Ime, Adresa, ID mjesta. Ovaj entitet je u *One-To-One* vezi s Korisnikom preko atributa ID korisnika. Entitet je također u *One-To-Many* vezi sa Šetnjom preko atributa ID, te u *Many-To-One* vezi s Mjestom preko atributa ID mjesta.

Udruga		
ID	INT	jedinstveni identifikator udruge
ID korisnika	INT	jedinstveni identifikator korisnika (korisnik.ID)
OIB	CHAR	OIB udruge
Ime	VARCHAR	ime udruge
Adresa	VARCHAR	adresa udruge
ID mjesta	INT	jedinstveni identifikator mjesta (mjesto.ID)

**Mjesto** Ovaj entitet sadržava sve informacije vezane za Mjesto. Sadrži attribute: ID, Ime. Ovaj entitet je u *One-To-Many* vezi s Udrugom preko atributa ID.

Mjesto		
ID	INT	jedinstveni identifikator mjesta
Ime	VARCHAR	ime mjesta

**Životinja** Ovaj entitet sadržava sve informacije vezane za Životinju. Sadrži attribute: ID, Slika, Opis, Ime, ID vrste, Godina rođenja, Preferirana vrsta šetnje, Spol, ID udruge. Ovaj entitet je u *Many-To-One* vezi s Udrugom preko atributa ID udruge, te s Vrstom preko atributa ID vrste. Entitet je u *Many-To-Many* vezi sa Šetnja Životinja preko atributa ID.

Životinja		
ID	INT	jedinstveni identifikator životinje
Slika	LOB	slika životinje
Opis	VARCHAR	opis životinje
Ime	VARCHAR	ime životinje
ID vrste	INT	jedinstveni identifikator vrste (vrsta.ID)
Godina rođenja	INT	godina rođenja životinje
Preferirana vrsta šetnje	VARCHAR	preferirana vrsta šetnje životinje
Spol	VARCHAR	spol životinje
ID udruge	INT	jedinstveni identifikator udruge (udrug.ID)

**Vrsta** Ovaj entitet sadržava sve informacije vezane za Vrstu. Sadrži attribute: ID, Ime, Visina, Težina, Životni vijek, Grupa. Ovaj entitet je u *One-To-Many* vezi sa Životinjom preko atributa ID.

Vrsta		
ID	INT	jedinstveni identifikator vrste
Ime	VARCHAR	ime vrste
Visina	VARCHAR	prosječna visina vrste
Težina	VARCHAR	prosječna težina vrste
Životni vijek	VARCHAR	prosječni životni vijek vrste



Vrsta		
Grupa	VARCHAR	grupa vrste

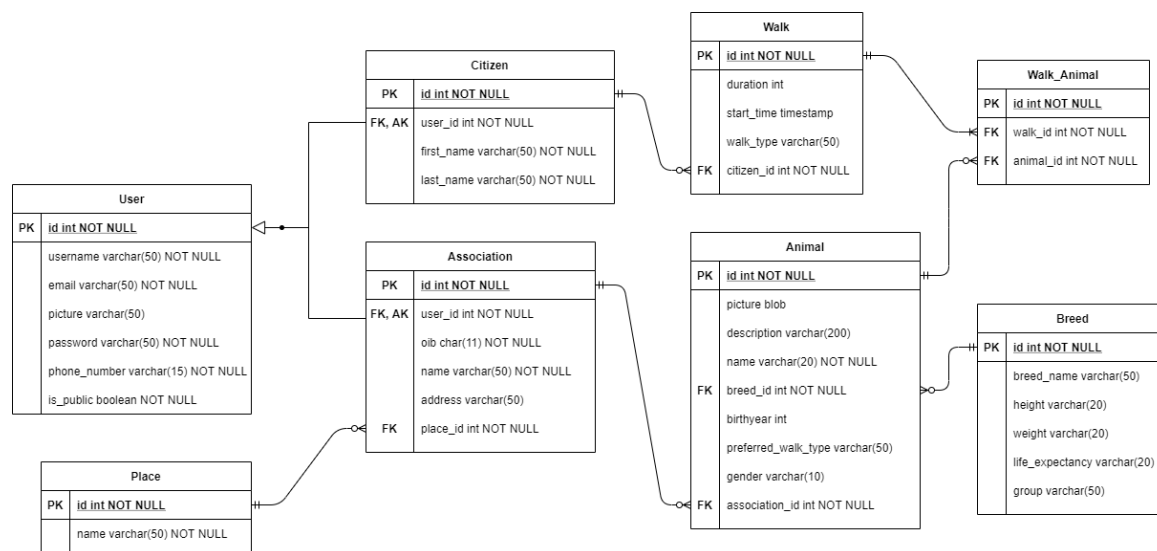
**Šetnja** Ovaj entitet sadržava sve informacije vezane za Šetnju. Sadrži attribute: ID, Trajanje, Početak, Vrsta, ID korisnika. Ovaj entitet je u *Many-To-One* vezi s Građaninom preko atributa ID građanina, te u *One-To-Many* vezi sa Šetnja Životinja preko atributa ID.

Šetnja		
ID	INT	jedinstveni identifikator šetnje
Trajanje	INT	vremensko trajanje šetnje
Početak	DATETIME	vrijeme početka šetnje
Vrsta šetnje	VARCHAR	vrsta šetnje
ID građanina	INT	jedinstveni identifikator građanina (građanin.ID)

**Šetnja Životinja** Ovaj entitet sadržava sve informacije o odnosu Šetnje i Životinje. Sadrži attribute: ID, ID šetnje, ID životinje. Ovaj entitet je u *Many-To-One* vezi sa Šetnjom preko atributa ID šetnje, te u *Many-To-Many* vezi sa Životinjom preko atributa ID životinje.

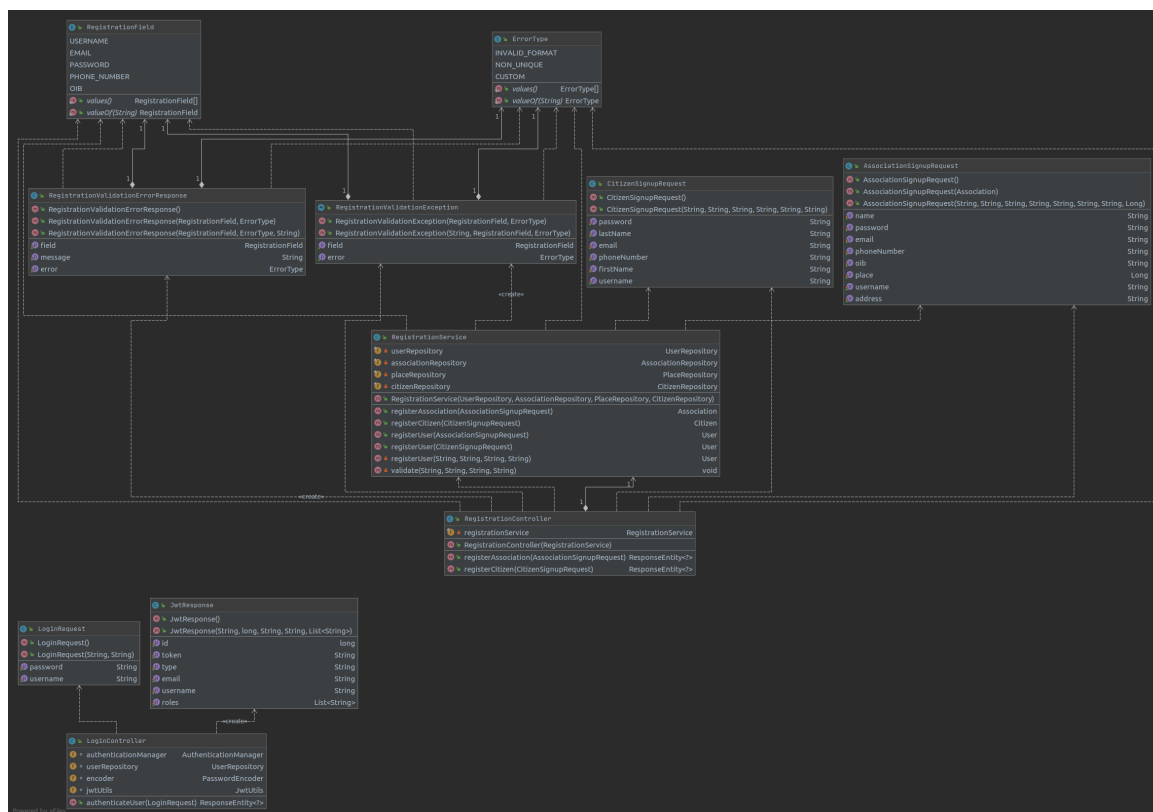
Šetnja Životinja		
ID	INT	jedinstveni identifikator šetnje životinje
ID šetnje	INT	jedinstveni identifikator šetnje (šetnja.ID)
ID životinje	INT	jedinstveni identifikator životinje (životinja.ID)

## 4.1.2 Dijagram baze podataka

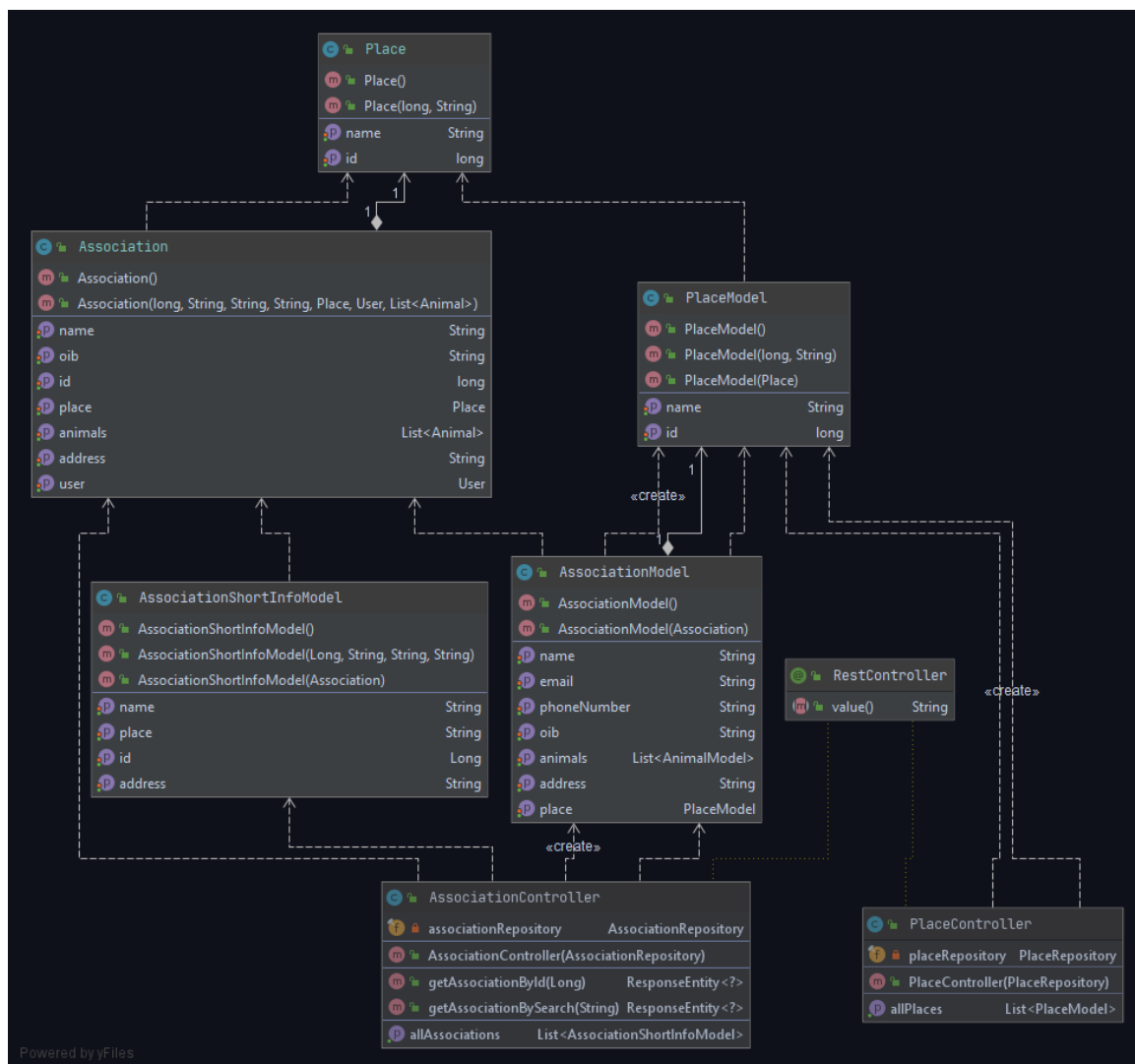


Slika 4.1: ER dijagram baze podataka

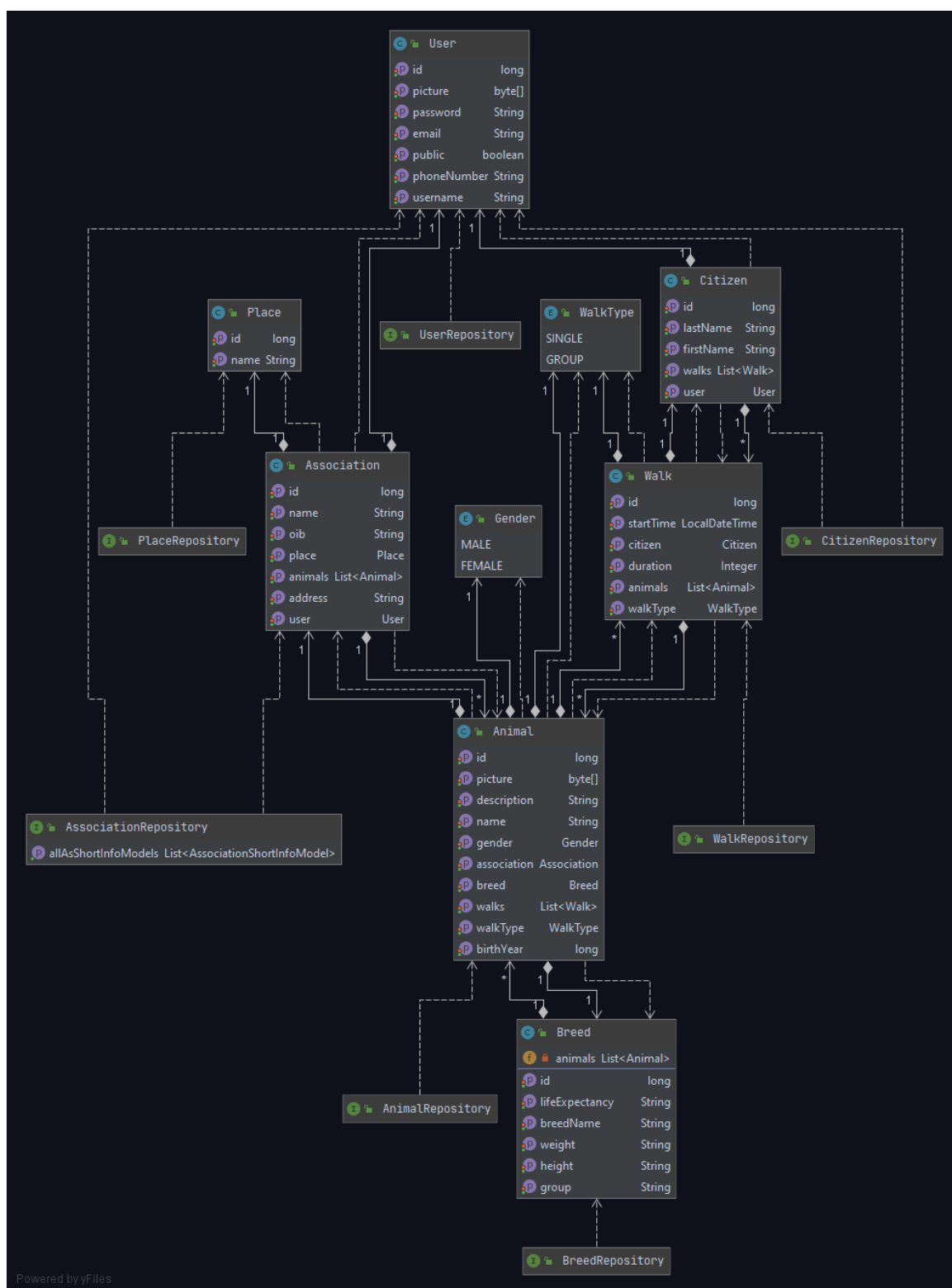
## 4.2 Dijagram razreda



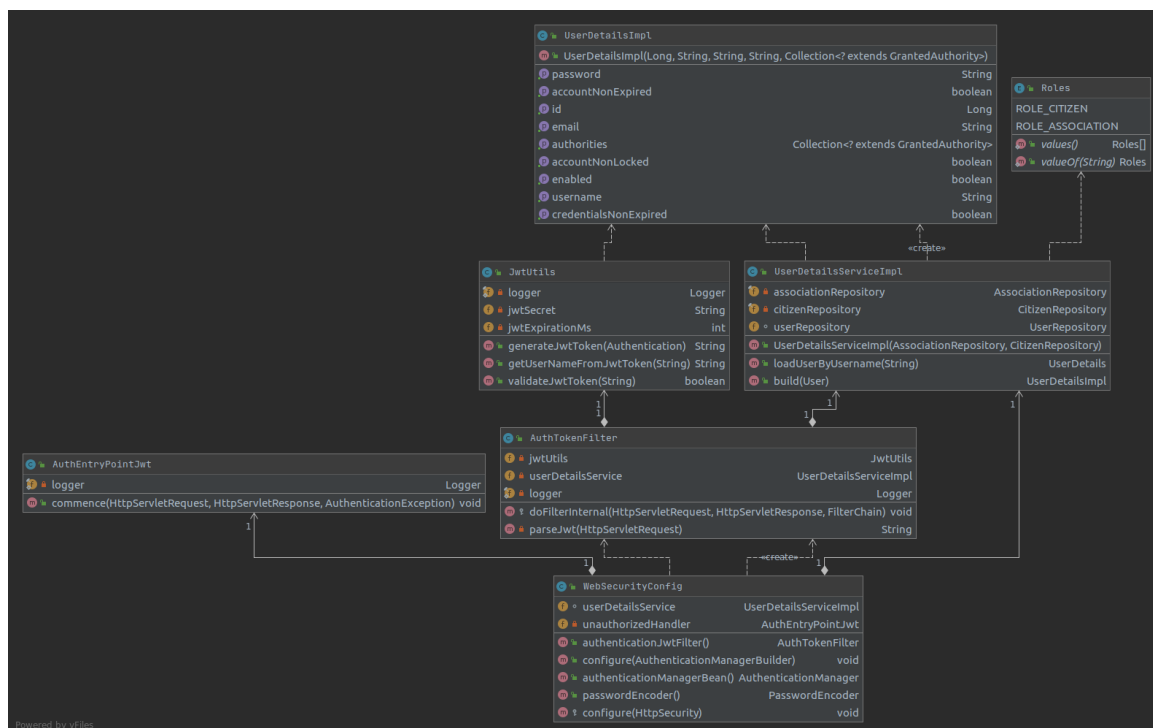
Slika 4.2: Kontroleri i servisi za login i registraciju



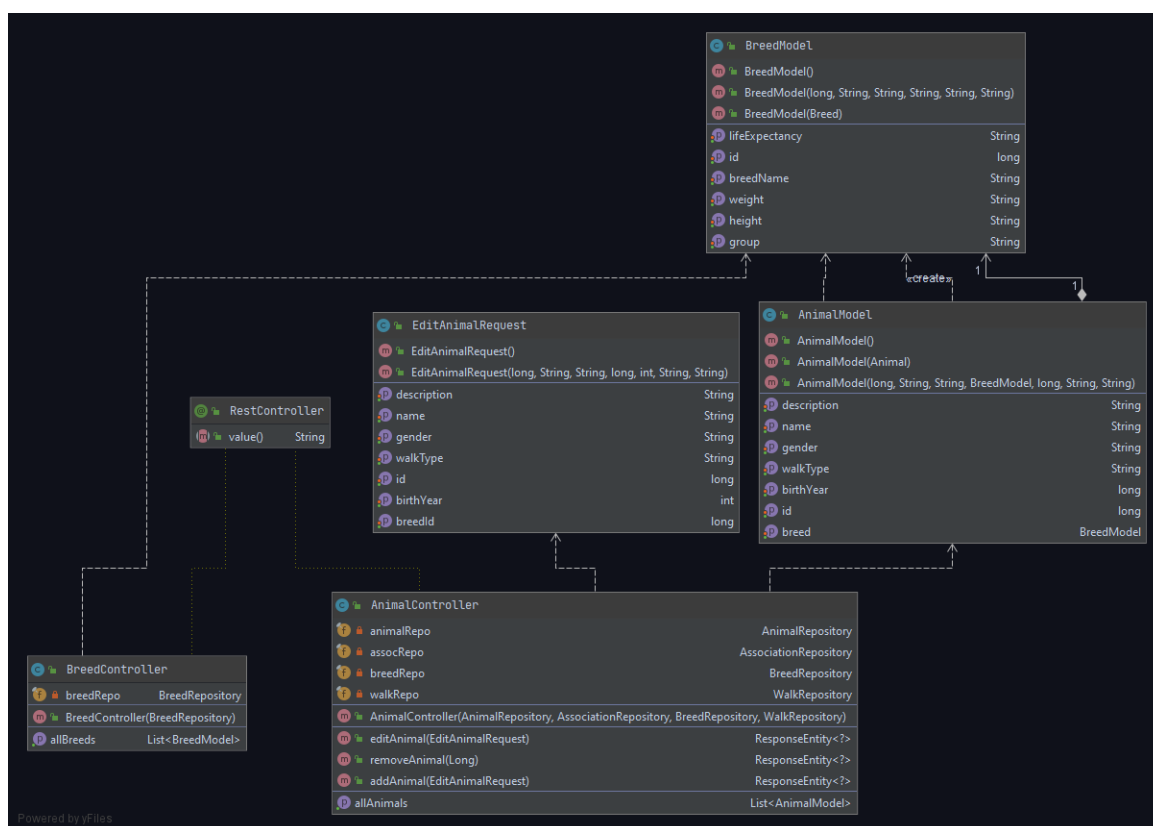
Slika 4.3: Kontroleri za udruge i mjesta



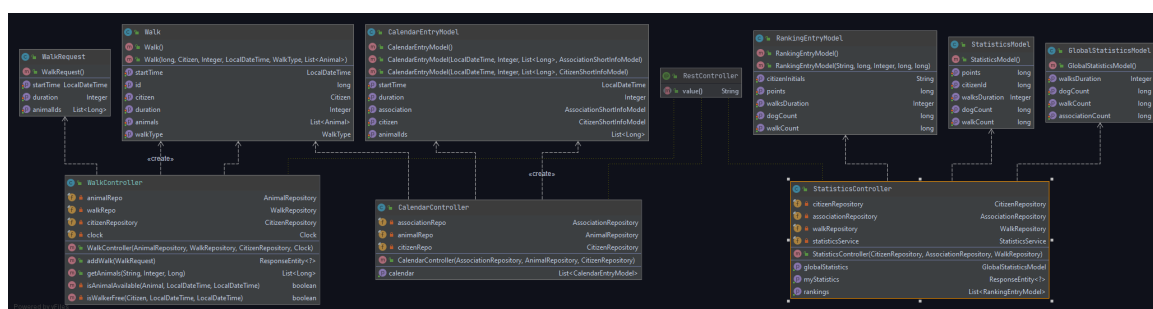
Slika 4.4: Entiteti i repozitoriji



Slika 4.5: Konfiguracija sigurnosti



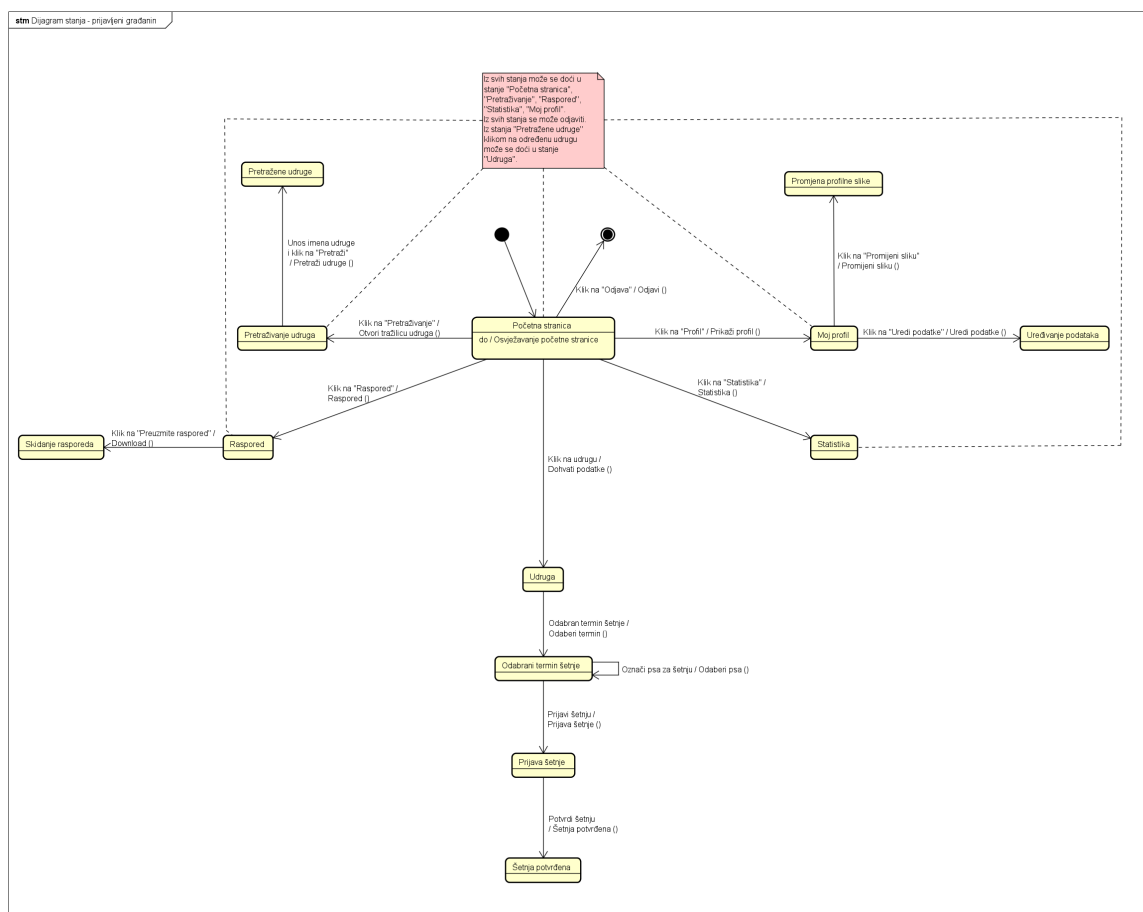
Slika 4.6: Kontroleri za životinje i pasmine



Slika 4.7: Kontroleri za šetnje, statistike i rasporede

### 4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekta. Prijelazi iz jednog stanja u drugo temelje se na događajima. Na slici 4.8 prikazan je dijagram stanja za prijavljenog građanina. Na početnoj stranici građaninu je prikazano zaglavlje stranice sa mogućnošću prikaza vlastitog profila, statistike, rasporeda i pretraživanja. Na vlastitom profilu je moguće promijeniti profilnu sliku te urediti podatke. Raspored ima mogućnost preuzimanja. Osim zaglavlja, na početnoj stranici građaninu su vidljive udruge. Klikom na određenu udruhu moguć je pregled podataka o udruzi te prijava šetnje nakon odabira termina i životinja koje se želi šetati. Ako su odabrani termin i životinje ispravni, šetnja se može potvrditi.

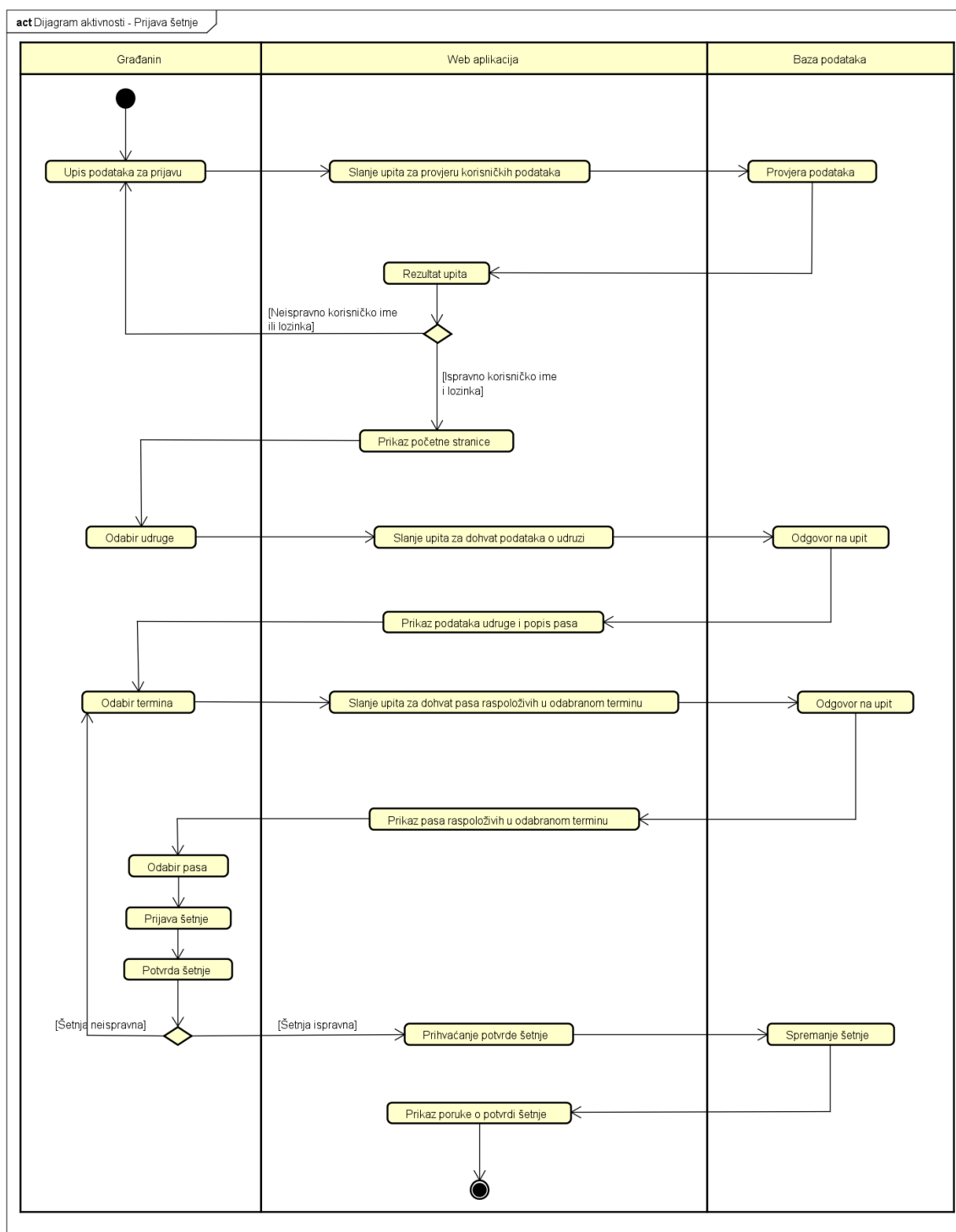


Slika 4.8: Dijagram stanja



## 4.4 Dijagram aktivnosti

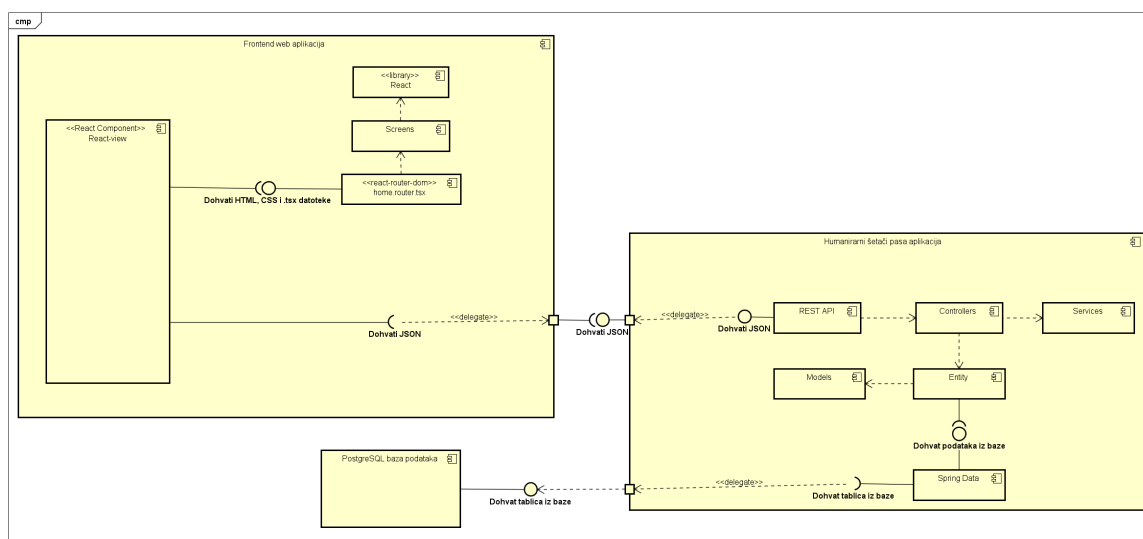
Dijagram aktivnosti koristi se za prikaz toka upravljanja ili toka podataka. Svaki korak se poduzima nakon što je prethodni završio. Na slici 4.9 prikazan je dijagram aktivnosti za prijavu šetnje. Građanin unese podatke za prijavu u sustav. Ako su uneseni podaci ispravni, prikaže mu se početna stranica. Ako uneseni podaci nisu ispravni, građanin mora ponovno unijeti podatke za prijavu. Nakon prijave u sustav, građaninu se prikaže početna stranica na kojoj odabire jednu od udruga. Nakon odabira udruge, građanin odabire termin i životinje te prijavljuje šetnju. Ako su odabrani termin i životinje ispravni, šetnja se potvrđuje te se prikazuje poruka o potvrdi šetnje.



Slika 4.9: Dijagram aktivnosti

## 4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici 4.10 prikazuje organizaciju sustava, međuovisnost komponenti, interne strukture i odnose prema okolini. Omogućen je pristup preko dva sučelja. Preko sučelja za dohvat HTML, CSS i .tsx datoteka poslužuje se frontend dio aplikacije. Home.router.tsx je komponenta koja na upit s url-om određuje koja datoteka će se poslužiti na sučelje. Frontend dio se sastoji od niza TypeScript datoteka. Sve TypeScript datoteke ovise o React knjižnici za koju je karakteristično dohvaćanje gotovih komponenti koje se mogu više puta koristiti. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje backend dio aplikacije. Spring Data je zadužen za dohvaćanje tablica iz baze podataka pomoću SQL upita. Podaci iz baze se šalju MVC arhitekturi u obliku entiteta.



Slika 4.10: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Za komunikaciju u timu korištene su aplikacije WhatsApp<sup>1</sup> i Discord<sup>2</sup>. Na web platformi GitLab<sup>3</sup> nalazi se udaljeni repozitorij projekta, a kao sustav za upravljanje izvornim kodom korišten je Git<sup>4</sup>. Za razvoj su korištena dva razvojna okruženja, Microsoft Visual Studio Code<sup>5</sup> i Intellij IDEA<sup>6</sup>. Microsoft Visual Studio Code je integrirano razvojno okruženje tvrtke Microsoft, a koristi se za razvoj računalnih programa, web stranica, web aplikacija, web usluga i mobilnih aplikacija. Intellij IDEA integrirano je razvojno okruženje napisano u Javi za razvoj računalnog softvera, razvio ga je JetBrains, a dostupan je kao izdanje licenci za zajednicu Apache 2 i u vlastitom komercijalnom izdanju.

Aplikacija je napisana koristeći aplikacijski okvir Spring Framework<sup>7</sup> i jezik Java<sup>8</sup> za izradu backenda. Spring je skup biblioteka i alata koji olakšava razvoj aplikacija te čini programiranje u Javi bržim, lakšim i sigurnijim. Usredotočenost Springa na brzinu, jednostavnost i produktivnost učinila ga je najpopularnijim okvirom za Javu. Za lakše uključivanje i integriranje modula, postavljanje securityja i mapiranje objektnog modela na relacijsku bazu podataka korišten je Spring Boot. Spring Boot<sup>9</sup> radi automatsko podešavanje i povezivanje različitih modula tako da analizira što smo uključili u classpath te sam zaključuje kako te module povezati u smisleni cjelinu. Za razvoj Web servisa korišten je REST. REST, ili REpresentational State Transfer, arhitekturni je stil za razvoj Web servisa koji pruža standard između računalnih sustava na webu, olakšavajući njihovu međusobnu komunikaciju.

---

<sup>1</sup><https://www.whatsapp.com/>

<sup>2</sup><https://discord.com/>

<sup>3</sup><https://gitlab.com>

<sup>4</sup><https://git-scm.com/>

<sup>5</sup><https://visualstudio.microsoft.com/>

<sup>6</sup><https://www.jetbrains.com/idea/>

<sup>7</sup><https://spring.io/>

<sup>8</sup><https://www.java.com/en/>

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

Za praćenje promjena u bazi podataka korišten je Liquibase<sup>10</sup>. Liquibase je knjižnica sa neovisnom bazom podataka otvorenog koda za praćenje, upravljanje i primjenu promjena sheme baze podataka. Pokrenut je 2006. godine kako bi omogućio lakše praćenje promjena u bazama podataka. Sve promjene u bazi podataka pohranjuju se u tekstualne datoteke (XML, JSON) i identificiraju se kombinacijom oznake "id" i "author", kao i imenom same datoteke. Za upravljanje bazom podataka korišten je besplatan sustav PostgreSQL<sup>11</sup> koji poštuje ACID principe pri izvođenju transakcija. Za testiranje backenda korišten je jednostavan okvir za pisanje ponovljivih testova JUnit<sup>12</sup>. JUnit je instanca xUnit arhitekture za okvire za jedinstveno testiranje.

Za izradu frontenda korišten je JavaScript library React<sup>13</sup> i jezik TypeScript<sup>14</sup>. React se koristi za izgradnju korisničkog sučelja ili UI komponenti. Održava ga Facebook. React se može koristiti kao osnova u razvoju aplikacija. Za realizaciju pojedinih prikaza kao što su okviri za prijavu i registraciju, tablice za statistiku, gumbi i slično, korišten je Material-UI<sup>15</sup>. To je open-source projekt koji sadrži React komponente koje implementiraju Googleov Material Design. Za preuzimanje rasporeda šetnji korišten je React-pdf koji služi za jednostavno generiranje PDF datoteka. HTTP zahtjevi i odgovori ostvareni su korištenjem Axios-a<sup>16</sup>. Axios je vrlo popularan JavaScript library za izvršavanje HTTP zahtjeva. Podržava starije i sve moderne preglednike, radi u Node.js-u te provodi automatsku transformaciju JSON podataka. Temelji se na obećanjima, što omogućuje pisanje async/await koda za vrlo lako izvršavanje XHR zahtjeva. Dizajn web aplikacije napravljen je pomoću stilskog jezika CSS<sup>17</sup>. Testiranje frontenda ostvareno je koristeći radni okvir Selenium<sup>18</sup>.

Dokumentacija je napisana u jeziku LaTeX<sup>19</sup> u integriranom okruženju za pisanje TeXStudio<sup>20</sup>. Za izradu UML dijagrama korišten je alat Astah UML<sup>21</sup>. ER dijagram baze podataka izrađen je u besplatnom online softveru draw.io<sup>22</sup>, koji se

---

<sup>10</sup><https://www.liquibase.org/>

<sup>11</sup><https://www.postgresql.org/>

<sup>12</sup><https://junit.org/>

<sup>13</sup><https://reactjs.org/>

<sup>14</sup><https://www.typescriptlang.org/>

<sup>15</sup><https://material-ui.com/>

<sup>16</sup><https://www.npmjs.com/package/axios>

<sup>17</sup><https://www.w3.org/Style/CSS/Overview.en.html>

<sup>18</sup><https://www.selenium.dev/>

<sup>19</sup><https://www.latex-project.org/>

<sup>20</sup><https://www.texstudio.org/>

<sup>21</sup><https://astah.net/downloads/>

<sup>22</sup><https://app.diagrams.net/>

između ostalog koristi i za izradu dijagrama toka i procesa, organizacijskih dijagrama, UML, ER i mrežnih dijagrama.

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Za testiranje smo odabrali WalkController i AssociationController koji implementiraju glavne funkcionalnosti sustava. Testirali smo controllere kako bi provjerili rad sustava u cjelini. Prije testova postavljeni su testni podaci. Na slici je prikazano postavljanje Spring Boot Security podataka.

```
Authentication authentication = Mockito.mock(Authentication.class);
when(authentication.getPrincipal()).thenReturn(new UserDetailsImpl( id: 1L,
    username: "bhorvat",
    email: "bbhorvatt@gmail.com",
    password: "keks",
    List.of(new SimpleGrantedAuthority( role: "ROLE_CITIZEN"))));
SecurityContext securityContext = Mockito.mock(SecurityContext.class);
Mockito.when(securityContext.getAuthentication()).thenReturn(authentication);
SecurityContextHolder.setContext(securityContext);
```

Slika 5.1: Postavljanje sigurnosti

Zbog testiranja prijave šetnje u prošlom vremenu, postavili smo fiksni datum kako je prikazano na slici dolje. Zbog vremenskih zona zahtjevi koji se šalju su u vremenskoj zoni GMT+0, a u sustavu se pohranjuju u vremenskoj zoni GMT+1.

```
fixedClock = Clock.fixed(LocalDateTime.of( year: 2020,
    month: 1,
    dayOfMonth: 8,
    hour: 0,
    minute: 0).toInstant(ZoneOffset.ofHours(0)), ZoneId.systemDefault());
doReturn(fixedClock.getZone()).when(clock).getZone();
doReturn(fixedClock.instant()).when(clock).instant();
```

Slika 5.2: Postavljanje fiksnog vremena

Zbog testiranja prijave šetnje kad je pas već zauzet, potrebna su nam dva građana.

```
Citizen citizen = new Citizen( id: 1, firstName: "Benjamin", lastName: "Horvat", user: null);
Citizen alternateCitizen = new Citizen( id: 2, firstName: "Vid", lastName: "Muzevic", user: null);
Association association = new Association();
Breed breed = new Breed( id: 1, breedName: "Jazavcar", height: "12-15cm", weight: "15-25kg", lifeExpectancy: "12-16", group: "Toy");
association.setBreed(breed);
```

Slika 5.3: Postavljanje 2 građana, 1 udruge i 1 pasmine

Postavili smo testne podatke za dva psa koji su predodređeni za pojedinačnu šetnju i dva psa koji su predodređeni za grupnu šetnju. Zatim su postavljeni podaci za pet šetnji sa prije definiranim životinjama.

```
Animal animal1 = new Animal( id: 1, picture: null, description: "Descr", name: "Kyra", breed, birthYear: 2013, WalkType.SINGLE, Gender.FEMALE, association);
Animal animal2 = new Animal( id: 2, picture: null, description: "Descr", name: "Snoopy", breed, birthYear: 2013, WalkType.SINGLE, Gender.MALE, association);
Animal animal3 = new Animal( id: 3, picture: null, description: "Descr", name: "Felix", breed, birthYear: 2013, WalkType.GROUP, Gender.MALE, association);
Animal animal4 = new Animal( id: 4, picture: null, description: "Descr", name: "Gina", breed, birthYear: 2013, WalkType.GROUP, Gender.FEMALE, association);

List<Animal> animals = List.of(animal1, animal2, animal3, animal4);

Walk walk1 = new Walk( id: 1L, citizen, duration: 1, LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 10, hour: 13, minute: 00), WalkType.GROUP, List.of(animal3, animal4));
Walk walk2 = new Walk( id: 2L, citizen, duration: 1, LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 10, hour: 12, minute: 00), WalkType.SINGLE, List.of(animal1));
Walk walk3 = new Walk( id: 3L, citizen, duration: 2, LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 10, hour: 12, minute: 00), WalkType.SINGLE, List.of(animal2));
Walk walk4 = new Walk( id: 4L, citizen, duration: 1, LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 10, hour: 16, minute: 00), WalkType.SINGLE, List.of(animal1));
Walk walk5 = new Walk( id: 5L, alternateCitizen, duration: 1, LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 10, hour: 8, minute: 00), WalkType.SINGLE, List.of(animal1));

animal1.setWalks(List.of(walk2, walk4, walk5));
animal2.setWalks(List.of(walk3));
animal3.setWalks(List.of(walk1));
animal4.setWalks(List.of(walk1));

citizen.setWalks(List.of(walk1, walk2, walk3, walk4));
alternateCitizen.setWalks(List.of(walk5));
```

Slika 5.4: Postavljanje životinja i šetnji



Postavljanje ponašanja dvojnika za navedene pozive metoda AnimalRepositoryja.

```
given(animalRepository.getByAssociationId(1L))
|     .willReturn(animals);
given(animalRepository.existsById(1L))
|     .willReturn(true);
given(animalRepository.existsById(2L))
|     .willReturn(true);
given(animalRepository.existsById(3L))
|     .willReturn(true);
given(animalRepository.existsById(4L))
|     .willReturn(true);
given(animalRepository.findById(1L))
|     .willReturn(Optional.of(animal1));
given(animalRepository.findById(2L))
|     .willReturn(Optional.of(animal2));
given(animalRepository.findById(3L))
|     .willReturn(Optional.of(animal3));
given(animalRepository.findById(4L))
|     .willReturn(Optional.of(animal4));

given(citizenRepository.getByUserId(1L))
|     .willReturn(citizen);
```

Slika 5.5: Postavljanje dvojnika

Na API poziv `"/api/walk"` šaljemo ispravne podatke šetnje. Očekivani rezultat je statusni kod 200.

```
@Test
public void test_addWalk() throws Exception {
    mvc.perform(post( uriTemplate: "/api/walk")
        .accept(MediaType.APPLICATION_JSON)
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"startTime\":\"2020-01-10T10:00:00\", \"duration\":1, \"animalIds\":[1]}"))
        .andExpect(status().isOk());
}
```

Slika 5.6: Testiranje ispravne šetnje

Na API poziv `"/api/walk"` šaljemo početak šetnje koji je prošao. Očekivani rezultat je statusni kod 400.

```
@Test
public void test_addWalk_timePassed() throws Exception { //Vrijeme je postavljeno na 8.1.2020. 00:00
    mvc.perform(post( uriTemplate: "/api/walk")
        .accept(MediaType.APPLICATION_JSON)
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"startTime\":\"2020-01-07T09:00:00\", \"duration\":1, \"animalIds\":[1]}"))
        .andExpect(status().isBadRequest());
}
```

Slika 5.7: Testiranje šetnje u prošlom vremenu

Na API poziv `"/api/walk"` šaljemo podatke o šetnji u terminu u kojem je građanin već zauzet. Očekivani rezultat je statusni kod 400.

```
@Test
public void test_addWalk_citizenNotAvailable() throws Exception {
    mvc.perform(post( uriTemplate: "/api/walk")
        .accept(MediaType.APPLICATION_JSON)
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"startTime\":\"2020-01-10T11:00:00\", \"duration\":1, \"animalIds\":[1]}"))
        .andExpect(status().isBadRequest());
}
```

Slika 5.8: Testiranje šetnje za koju građanin nije dostupan

Na API poziv `"/api/walk"` šaljemo podatke o šetnji bez navedene životinje. Očekivani rezultat je statusni kod 400.

```
@Test
public void test_addWalk_noAnimals() throws Exception {
    mvc.perform(post( uriTemplate: "/api/walk")
        .accept(MediaType.APPLICATION_JSON)
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"startTime\":\"2020-01-10T09:00:00\", \"duration\":1, \"animalIds\":[]}"))
        .andExpect(status().isBadRequest());
}
```

Slika 5.9: Testiranje šetnje bez odabrane životinje

Na API poziv `"/api/walk"` šaljemo podatke o šetnji u terminu u kojem je životinja već zauzeta. Očekivani rezultat je statusni kod 400.

```
@Test
public void test_addWalk_animalNotAvailable() throws Exception {
    mvc.perform(post( uriTemplate: "/api/walk")
        .accept(MediaType.APPLICATION_JSON)
        .contentType(MediaType.APPLICATION_JSON)
        .content("{\"startTime\":\"2020-01-10T07:00:00\", \"duration\":1, \"animalIds\":[1]}"))
        .andExpect(status().isBadRequest());
}
```

Slika 5.10: Testiranje šetnje za koju životinja nije dostupna

Na API poziv `"/api/walk/animals"` šaljemo podatke o željenom početku šetnje, trajanju i odabranu udrugu. Cilj nam je provjeriti dostupnost životinje u navedenom terminu. Očekivani rezultat je statusni kod 200.

```
@Test
public void test_getAnimals() throws Exception {
    mvc.perform(get( uriTemplate: "/api/walk/animals")
        .accept(MediaType.APPLICATION_JSON)
        .param( name: "startTime", ...values: "2020-01-10T12:00:00") //pretvori se u 13:00 radi vremenskih zona...
        .param( name: "duration", ...values: "1")
        .param( name: "associationId", ...values: "1"))
        .andExpect(status().isOk())
        .andExpect(content().json( jsonContent: "[1]"));

    mvc.perform(get( uriTemplate: "/api/walk/animals")
        .accept(MediaType.APPLICATION_JSON)
        .param( name: "startTime", ...values: "2020-01-10T13:00:00")
        .param( name: "duration", ...values: "1")
        .param( name: "associationId", ...values: "1"))
        .andExpect(status().isOk())
        .andExpect(content().json( jsonContent: "[2,3,4,1]"));

    mvc.perform(get( uriTemplate: "/api/walk/animals")
        .accept(MediaType.APPLICATION_JSON)
        .param( name: "startTime", ...values: "2020-01-10T11:00:00")
        .param( name: "duration", ...values: "1")
        .param( name: "associationId", ...values: "1"))
        .andExpect(status().isOk())
        .andExpect(content().json( jsonContent: "[3,4]"));
}
```

Slika 5.11: Testiranje dostupnosti životinja

Slijede rezultati navedenih testova.

## WalkControllerTest

all > [hr.fer.progi.oneclick.humanitarnisetacipasa.controllers](#) > WalkControllerTest

<b>6</b> tests	<b>0</b> failures	<b>0</b> ignored	<b>0.270s</b> duration	<b>100%</b> successful
-------------------	----------------------	---------------------	---------------------------	---------------------------

Tests

Standard output

Test	Duration	Result
test_addWalk()	0.016s	passed
test_addWalk_animalNotAvailable()	0.015s	passed
test_addWalk_citizenNotAvailable()	0.162s	passed
test_addWalk_noAnimals()	0.015s	passed
test_addWalk_timePassed()	0.012s	passed
test_getAnimals()	0.050s	passed

Wrap lines ☐

Generated by [Gradle 6.7](#) at Jan 11, 2021, 2:14:09 PM

Slika 5.12: Rezultati testiranja WalkControllera

Postavljeni su podaci za jednu udrugu.

```
Association association = new Association( id: 1L,  
    oib: "01234567891",  
    name: "Udruga Spas Varazdin",  
    address: "Nemetin 3",  
    new Place( id: 1L, name: "Varazdin"),  
    new User( id: 1L, username: "bbhorvat", email: "bbhorvatt@gmail.com", picture: null, password: null, phoneNumber: "049123912", isPublic: true),  
    new ArrayList<>());  
  
when(associationRepository.getAllAsShortInfoModels())  
    .thenReturn(List.of(new AssociationShortInfoModel(association)));  
when(associationRepository.findById(1L))  
    .thenReturn(Optional.of(association));  
when(associationRepository.findById(2L))  
    .thenReturn(Optional.empty());
```

Slika 5.13: Postavljanje testnih podataka za testiranje AssociationControllera

Na API poziv `"/api/association"` šaljemo zahtjev za dohvat svih udruga. Očekivani rezultat je statusni kod 200.

```
@Test  
public void test_getAll() throws Exception {  
    mvc.perform(get( uriTemplate: "/api/associations")  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(status().isOk())  
        .andExpect(content().json( jsonContent: "[{\"id\":\"1\",\"name\":\"Udruga Spas Varazdin\",\"address\":\"Nemetin 3\", \"place\":\"Varazdin\"}]"));  
}
```

Slika 5.14: Testiranje dohvata svih udruga

Na API poziv `"/api/association/1"` šaljemo zahtjev za dohvat udruge sa šifrom 1. Očekivani rezultat je statusni kod 200.

```
@Test  
public void test_getAssociation() throws Exception {  
    mvc.perform(get( uriTemplate: "/api/associations/1")  
        .accept(MediaType.APPLICATION_JSON))  
        .andExpect(status().isOk());  
}
```

Slika 5.15: Testiranje dohvata odabrane udruge

Na API poziv `"/api/association/2"` šaljemo zahtjev za dohvat udruge sa šifrom 2. Očekivani rezultat je statusni kod 400.

```
@Test
public void test_getAssociation_notExists() throws Exception {
    mvc.perform(get( uriTemplate: "/api/associations/2")
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isBadRequest());
}
```

Slika 5.16: Testiranje dohvata udruge koja ne postoji

Slijede rezultati navedenih testova.

## AssociationControllerTest

all > [hr.fer.progi.oneclick.humanitarnisetacipasa.controllers](#) > AssociationControllerTest

<b>3</b> tests	<b>0</b> failures	<b>0</b> ignored	<b>0.308s</b> duration	<b>100%</b> successful
-------------------	----------------------	---------------------	---------------------------	---------------------------

Tests

Standard output

Test	Duration	Result
test_getAll()	0.248s	passed
test_getAssociation()	0.031s	passed
test_getAssociation_notExists()	0.029s	passed

Wrap lines ☐

Generated by [Gradle 6.7](#) at Jan 11, 2021, 2:14:09 PM

Slika 5.17: Rezultati testiranja AssociationControllera



## 5.2.2 Ispitivanje sustava

Ulaz za prvi test je nevaljano korisničko ime i lozinka. Očekivani rezultat je prikazivanje poruke "Pogrešni podaci za prijavu!".

	Command	Target	Value
1	open	/	
2	set window size	1200x700	
3	mouse over	css= .MuiButtonBase-root:nth-child(5) > .MuiButton-label	
4	click	css= .MuiButtonBase-root:nth-child(5) > .MuiButton-label	
5	mouse out	css= .MuiButtonBase-root:nth-child(5) > .MuiButton-label	
6	click	id=username	
7	type	id=username	fail
8	mouse over	css= .MuiButton-contained > .MuiButton-label	
9	type	id=password	totalFail
10	click	css= .MuiButton-contained > .MuiButton-label	
11	mouse out	css= .MuiButton-contained > .MuiButton-label	
12	click	css= .centered	
13	assert element present	css= .centered	

Slika 5.18: Koraci prvog testa

Rezultat prvog testa je prikaz poruke "Pogrešni podaci za prijavu!".

**Prijavite se:**

Pogrešni podaci za prijavu!

**Korisničko ime**

Unesite korisničko ime

fail

**Lozinka**

Unesite lozinku

.....

**PRIJAVA**

Nemate korisnički račun? [Registrirajte se](#)

Slika 5.19: Rezultat prvog testa


Ulaz za drugi test su ispravni podaci. Očekivani rezultat je prijava u sustav.

	Command	Target	Value
1	open	/	
2	set window size	1200x800	
3	click	css=.MuiButtonBase-root:nth-child(5) > .MuiButton-label	
4	click	id=username	
5	click	id=username	
6	type	id=username	gradanin1
7	type	id=password	korisnik
8	send keys	id=password	\$(KEY_ENTER)
9	assert element present	css=.MuiButtonBase-root:nth-child(5) > .MuiButton-label	
10	click	css=.MuiButtonBase-root:nth-child(5) > .MuiButton-label	
11	click	css=.MuiButtonBase-root:nth-child(5) > .MuiButton-label	
12	assert text	css=.MuiTypography-root:nth-child(3)	Muzevic
13	assert element present	css=.MuiTypography-root:nth-child(3)	

Slika 5.20: Koraci drugog testa

Rezultat drugog testa je prijava u sustav s danim korisničkim imenom i lozinkom.

POČETNA STRANICA
PRETRAŽIVANJE
STATISTIKA
RASPORED
PROFIL



**Moj profil**

Notvid  
Muzevic  
gradanin1@gradaninmail.com  
gradanin1  
0958267411

UREDÍ PODATKE
PROMIJENI SLIKU

Moja statistika:

4

Od 6h

7

555

šetnji

provedeno u šetnjama

pasa šetano

hodova

Slika 5.21: Rezultat drugog testa

Ulaz za treći test su podaci za registraciju. Očekivani rezultat je registracija u sustav.

	Command	Target	Value
1	open	/	
2	set window size	1696x1036	
3	click	css=.MuiButtonBase-root:nth-child(4) > .MuiButton-label	
4	click	css=.registration-button > .MuiButton-label	
5	type	id=username	vmuzevic
6	type	id=email	vmuzevic@gmail.com
7	type	id=password	Zxcvbnm1
8	type	id=firstName	Vid
9	type	id=lastName	Muzevic
10	type	id=phoneNumber	0991728472
11	click	css=.MuiButton-contained > .MuiButton-label	
12	click	css=.MuiButtonBase-root:nth-child(5) > .MuiButton-label	
13	send keys	id=username	vmuzevic
14	type	id=password	Zxcvbnm1
15	click	css=.MuiButton-contained > .MuiButton-label	
16	click	css=.MuiButtonBase-root:nth-child(3) > .MuiButton-label	
17	assert element present	css=.makeStyles-titles-7:nth-child(7)	

Slika 5.22: Koraci trećeg testa

Rezultat trećeg testa je registracija u sustav s danim podacima.

šetači pasa

POČETNA STRANICA   PRETRAŽIVANJE   STATISTIKA   RASPORED   PROFIL

Moj profil

Vid  
Muzevic  
vmuzevic@gmail.com  
vmuzevic  
0991728472

UREDÍ PODATKE   PROMIJENÍ SLIKU

Moja statistika:

0  
Šetnji

0d 0h  
provedeno u šetnjama

0  
pasa šetano

0  
bodova

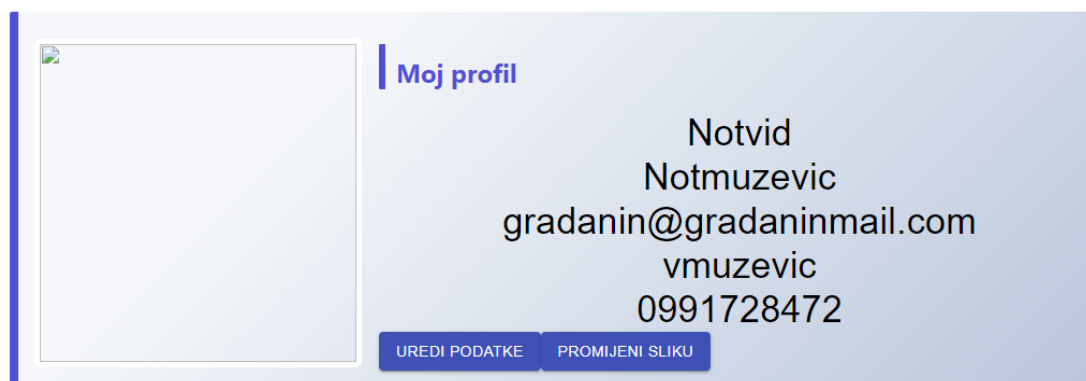
Slika 5.23: Rezultat trećeg testa

Ulaz za četvrti test su podaci koje želimo urediti, tj. promijeniti njihovu vrijednost. Očekivani rezultat su promijenjeni podaci na stranici profila.

	Command	Target	Value
1	open	/	
2	set window size	1696x1036	
3	click	css= MuiButtonBase-root:nth-child(5) > MuiButton-label	
4	click	css= MuiButton-contained:nth-child(7) > .MuiButton-label	
5	type	id=password	korisnik
6	type	id=email	gradanin@gradaninmail.com
7	type	id=firstName	Notvid
8	type	id=lastName	Notmuzevic
9	click	css= MuiButton-contained	
10	run script	window.scrollTo(0,0)	
11	click	css= MuiButtonBase-root:nth-child(5) > MuiButton-label	
12	assert text	css= MuiTypography-h4:nth-child(2)	Notvid
13	assert text	css= MuiTypography-root:nth-child(3)	Notmuzevic
14	assert text	css= MuiTypography-root:nth-child(4)	gradanin@gradaninmail.com

Slika 5.24: Koraci četvrtog testa

Rezultat četvrtog testa su promijenjeni podaci na stranici profila.



Slika 5.25: Rezultat četvrtog testa

Ulaz petog testa su podaci za prijavu šetnje, tj. datum, vrijeme početka i trajanje šetnje. U testu se odabir datuma i vremena šetnje provodi klikanjem na elemente za odabir datuma i vremena. Očekivani rezultat je prijava šetnje.

	Command	Target	Value
1.	open	/	
2.	set window size	1696x1036	
3.	click	css= MuIconButton-root:nth-child(5) > .MuiButton-label	
4.	type	id=username	vmuzevic
5.	type	id=password	korisnik
6.	click	css= MuButton-contained	
7.	click	css= .makeStyles-root-1:nth-child(3) .MuiCardMedia-root	
8.	click	id=startTime	
9.	click	css= MuTypography-subtitle1	
10.	click	css= MuTypography-root:nth-child(123)	
11.	click	css= MuPickersToolbarText-toolbarBtnSelected	
12.	click	css= MuPickersCalendar-week:nth-child(1) > div:nth-child(7) .MuiIconButton-label	
13.	click	css= MuPickersClock-squareMask	
14.	click	css= MuButton-textPrimary:nth-child(2) > .MuiButton-label	
15.	type	id=duration	2
16.	click	css= MuPaper-root:nth-child(6) .MuiCardMedia-root	
17.	click	css= MuPaper-root:nth-child(3) .MuiTypography-root:nth-child(2)	
18.	click	css= animal-selector-confirm-button .MuiButton-label	
19.	click	css= MuButton-outlined > .MuiButton-label	
20.	assert text	css= .horizontal-center	Šetnja je potvrđena, sada ju možete vidjeti u svom rasporedu.
21.	click	css= MuIconButton-root:nth-child(4) > .MuiButton-label	
22.	assert text	css= MuListSubheader-root	01.01.2022.
23.	assert text	css= MuListItemText-primary	18-20 sati
24.	assert text	css= MuListItemText-secondary	Ne - Savska 4, Ivanić-Grad

Slika 5.26: Koraci petog testa

Rezultat petog testa je prijavljena šetnja koja se može vidjeti u rasporedu.

[Preuzmite raspored!](#)

01.01.2022.

18-20 sati

Ne - Savska 4, Ivanić-Grad







Slika 5.27: Rezultat petog testa

Ulaz u šesti test su podaci o novoj životinji koju se želi dodati na stranicu udruge. Pretpostavka je da smo prijavljeni u sustav kao udruga. Očekivani rezultat je dodana životinja na stranicu udruge.

	Command	Target	Value
1	open	/	
2	set window size	1696x1036	
3	click	css= MuiButtonBase-root:nth-child(5) > .MuiButton-label	
4	click	css= MuiButton-contained:nth-child(3) > .MuiButton-label	
5	click	css= adddogbtn > .MuiButton-label	
6	type	id=name	Test
7	click	id=breedid	
8	click	css= MuiListItem-root:nth-child(1)	
9	type	id=birthYear	2013
10	click	id=gender	
11	click	css= MuiListItem-root:nth-child(1)	
12	click	id=walkType	
13	click	css= MuiListItem-root:nth-child(1)	
14	type	id=description	Pes
15	click	css= MuiButton-contained > .MuiButton-label	
16	assert text	css= MuiTableRow-root:nth-child(7) > .MuiTableCell-root:nth-child(1)	Test
17	assert text	css= MuiTableRow-root:nth-child(7) > .MuiTableCell-root:nth-child(3)	Jazavčar
18	assert text	css= MuiTableRow-root:nth-child(7) > .MuiTableCell-root:nth-child(4)	2013

Slika 5.28: Koraci šestog testa

Rezultat šestog testa je dodana životinja na stranicu udruge.

Humanitarni šetači pasa							POČETNA STRANICA	PRETRAŽIVANJE	STATISTIKA	RASPORED	PROFIL	ODJAVI
Ime	Slika	Pasmina	Godina rođenja	Spol	Vrsta šetnje	Opis						
Felix	 PROMIJENI SLIKU	Sibirski haski	2010	Muško	Grupna	Drag, mekan i povučen.	IZBRIŠI ŽIVOTINJU					
Kyra	 PROMIJENI SLIKU	Američki stafordski terijer	2013	Žensko	Pojedinačna	Jako energična, ali poslušna.	IZBRIŠI ŽIVOTINJU					
Gina	 PROMIJENI SLIKU	Sibirski haski	2013	Žensko	Grupna	Dadada	IZBRIŠI ŽIVOTINJU					
Test	 PROMIJENI SLIKU	Jazavčar	2013	Muško	Pojedinačna	Pes	IZBRIŠI ŽIVOTINJU					




Slika 5.29: Rezultat šestog testa

Ulaz sedmog testa je klik na gumb "Izbriši životinju". Očekivani rezultat je brisanje životinje dodane u prethodnom testu.

	Command	Target	Value
1.	<code>open</code>	<code>/</code>	
2.	<code>set window size</code>	<code>1696x1036</code>	
3.	<code>click</code>	<code>css= .MuiButtonBase-root:nth-child(5) &gt; .MuiButton-label</code>	
4.	<code>click</code>	<code>css= .MuiButton-contained:nth-child(3) &gt; .MuiButton-label</code>	
5.	<code>click</code>	<code>css= .MuiTableRow-root:nth-child(7) &gt; .MuiTableCell-root:nth-child(1)</code>	
6.	<code>assert element present</code>	<code>css= .MuiTableRow-root:nth-child(7) &gt; .MuiTableCell-root:nth-child(1)</code>	
7.	<code>click</code>	<code>css= .MuiTableRow-root:nth-child(7) &gt; .MuiTableCell-root:nth-child(8) .MuiButton-label</code>	
8.	<code>assert element not present</code>	<code>css= .MuiTableRow-root:nth-child(7) &gt; .MuiTableCell-root:nth-child(1)</code>	

Slika 5.30: Koraci sedmog testa

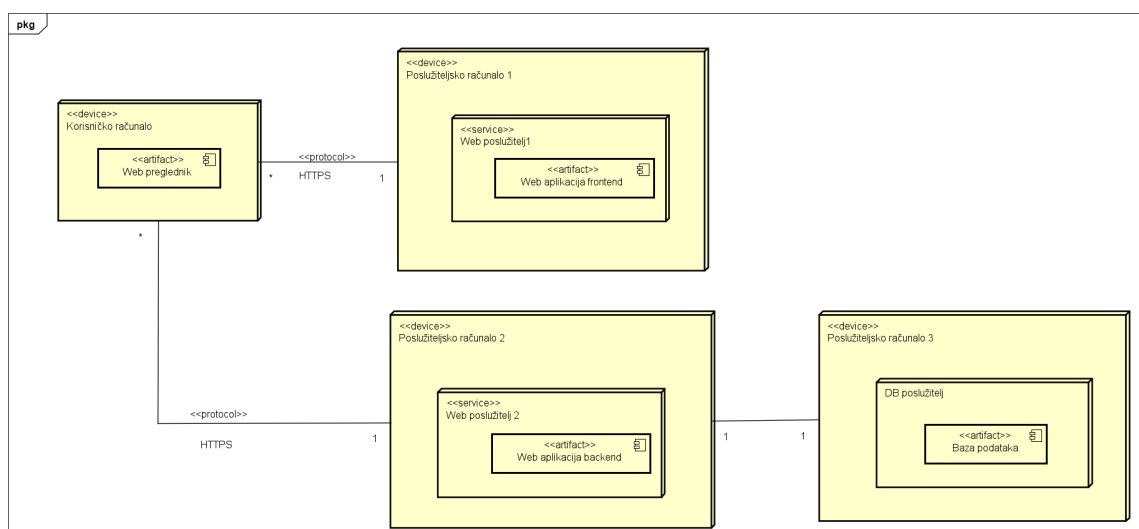
Rezultat sedmog testa je brisanje životinje dodane u prethodnom testu.

Humanitarni šetači pasa							
<div>POČETNA STRANICA   PRETRAŽIVANJE   STATISTIKA   RASPORED   PROFIL   ODJAVA   DODAJ PSA</div>							
Ime	Slika	Pasmina	Godina rođenja	Spol	Vrsta šetnje	Opis	
Felix	 PROMIJENI SLIKU	Sibirski haski	2010	Muško	Grupna	Drag, mekan i povučen.	IZBRIŠI ŽIVOTINJU
Kyra	 PROMIJENI SLIKU	Američki staffordski terijer	2013	Žensko	Pojedinačna	Jako energična, ali poslušna.	IZBRIŠI ŽIVOTINJU
Gina	 PROMIJENI SLIKU	Sibirski haski	2013	Žensko	Grupna	Darada	IZBRIŠI ŽIVOTINJU

Slika 5.31: Rezultat sedmog testa

## 5.3 Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom okruženju. Na poslužiteljskom računalu 1 nalazi se web poslužitelj za frontend, na poslužiteljskom računalu 2 nalazi se web poslužitelj za backend, a na poslužiteljskom računalu 3 se nalazi poslužitelj baze podataka. Klijenti koriste web preglednik kako bi pristupili web aplikaciji. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala korisnika i poslužitelja odvija se preko HTTPS veze.



Slika 5.32: Dijagram razmještaja



## 6. Zaključak i budući rad

Naš zadatak bio je razviti web aplikaciju (prilagođenu za mobilne uređaje) za humanitarno šetanje pasa koja omogućuje udrugama za životinje da ponude nezbrinute pse za šetnju, a građanima da dobrovoljno šetaju pse uz mogućnost pristupa vlastitom rasporedu šetnji i statistici šetača. Nakon 16 tjedana timskog rada, ostvarili smo zadani cilj, a taj proces odvijao se u dvije faze.

Prva faza projekta započela je okupljanjem razvojnog tima, dodjelom projektnog zadatka, upoznavanjem sa zahtjevima te njihovim definiranjem i dokumentiranjem. Detaljno definiranje zahtjeva na početku rada na projektu pokazalo se kao velika prednost pri implementaciji te omogućilo lakše planiranje rada i podjelu posla. Definiranje dijagrama obrazaca uporabe, sekvencijskih dijagrama, modela baze podataka i dijagrama razreda pružili su nam jasnu i jednoznačnu ideju ostvarenja funkcionalnosti naše aplikacije.

Druga faza projekta velikim dijelom sastojala se od kodiranja i implementacije traženih funkcionalnosti. Članovi su si izabrali zadatke prema osobnim predznanjima i interesima, no svi su se barem s nekom od korištenih tehnologija susreli po prvi puta, što ih je prisililo i potaknulo na samostalno učenje i napredak te međusobno dijeljenje znanja. Dobro definirani zahtjevi na početku spriječili su moguće nesporazume koji u kasnijom fazi implementacije mogu biti vrlo problematični i vremenski skupi. Također, bilo je potrebno dokumentirati dijagrame stanja, aktivnosti, komponenti i razmještaja te napisati upute za korištenje razvijene aplikacije.

Za vrijeme rada na projektu, članovi su komunicirali putem Whatsappa, na kojem su dijelili najvažnije informacije i dogovarali termine sastanaka, te putem Discorda, gdje su održavali sastanke, pratili dotadašnji napredak i razrađivali daljnje korake rada. Komunikacija članova tima bila je vrlo kvalitetna te je vladala pozitivna radna atmosfera.

Sudjelovanje u ovom projektu bilo je vrlo ugodno i poučno iskustvo te nas je kao buduće inženjere obogatilo ne samo tehničkim znanjima, nego i boljim komunikacijskim vještinama i iskustvom rada u grupi. Iako bi prethodno iskustvo rada na sličnim projektima uvelike olakšalo i ubrzalo razvoj naše aplikacije te konačni

proizvod zasigurno učinilo kvalitetnijim, zadovoljni smo postignutim rezultatima i vjerujemo da ćemo mnoga stečena znanja ponijeti sa sobom u buduće projekte koji nas čekaju kroz našu karijeru.

# 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>

# Indeks slika i dijagrama

3.1	Dijagram svih obrasca uporabe . . . . .	15
3.2	Sekvencijski dijagram za UC5 . . . . .	17
3.3	Sekvencijski dijagram za UC6 i UC6.1 . . . . .	18
4.1	ER dijagram baze podataka . . . . .	25
4.2	Kontroleri i servisi za login i registraciju . . . . .	26
4.3	Kontroleri za udruge i mjesta . . . . .	27
4.4	Entiteti i repozitoriji . . . . .	28
4.5	Konfiguracija sigurnosti . . . . .	29
4.6	Kontroleri za životinje i pasmine . . . . .	30
4.7	Kontroleri za šetnje, statistike i rasporede . . . . .	30
4.8	Dijagram stanja . . . . .	31
4.9	Dijagram aktivnosti . . . . .	33
4.10	Dijagram komponenti . . . . .	34
5.1	Postavljanje sigurnosti . . . . .	38
5.2	Postavljanje fiksnog vremena . . . . .	38
5.3	Postavljanje 2 građana, 1 udruge i 1 pasmine . . . . .	39
5.4	Postavljanje životinja i šetnji . . . . .	39
5.5	Postavljanje dvojnika . . . . .	40
5.6	Testiranje ispravne šetnje . . . . .	41
5.7	Testiranje šetnje u prošlom vremenu . . . . .	41
5.8	Testiranje šetnje za koju građanin nije dostupan . . . . .	41
5.9	Testiranje šetnje bez odabrane životinje . . . . .	42
5.10	Testiranje šetnje za koju životinja nije dostupna . . . . .	42
5.11	Testiranje dostupnosti životinja . . . . .	43
5.12	Rezultati testiranja WalkControllera . . . . .	44
5.13	Postavljanje testnih podataka za testiranje AssociationControllera . . . . .	45
5.14	Testiranje dohvata svih udruga . . . . .	45
5.15	Testiranje dohvata odabrane udruge . . . . .	45
5.16	Testiranje dohvata udruge koja ne postoji . . . . .	46

5.17	Rezultati testiranja AssociationControllera . . . . .	47
5.18	Koraci prvog testa . . . . .	48
5.19	Rezultat prvog testa . . . . .	48
5.20	Koraci drugog testa . . . . .	49
5.21	Rezultat drugog testa . . . . .	49
5.22	Koraci trećeg testa . . . . .	50
5.23	Rezultat trećeg testa . . . . .	50
5.24	Koraci četvrtog testa . . . . .	51
5.25	Rezultat četvrtog testa . . . . .	51
5.26	Koraci petog testa . . . . .	52
5.27	Rezultat petog testa . . . . .	52
5.28	Koraci šestog testa . . . . .	53
5.29	Rezultat šestog testa . . . . .	53
5.30	Koraci sedmog testa . . . . .	54
5.31	Rezultat sedmog testa . . . . .	54
5.32	Dijagram razmještaja . . . . .	55
6.1	Ukupni commitovi . . . . .	68
6.2	Commitovi: Benjamin Horvat i Mihael Rodek . . . . .	68
6.3	Commitovi: Dora Doljanin i Karolina Mirković . . . . .	68
6.4	Commitovi: Vid Mužević i Petar Čovran . . . . .	69
6.5	Commitovi: Iva Zekić . . . . .	69

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 7. listopada 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - upoznavanje članova
  - dogovor o komuniciranju putem Discord servera
  - početna riječ o načelima vođenja projekta koji će se primijeniti
  - komentiranje zadanog zadatka

### 2. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - instalacija Git-a, TeXLive i TeXStudio kod svih članova grupe
  - upoznavanje s Git-om
  - upoznavanje s GitLab-om
  - dogovaranje načina korištenja branchova, issuea i merge requestova
  - početak pisanja funkcionalnih zahtjeva
  - početak razrade strukture baze podataka
- Zadaci do idućeg sastanka:
  - provjera funkcionalnih zahtjeva i sastavljanje pitanja vezano za njih

### 3. sastanak

- Datum: 19. listopada 2020.
- Prisustvovali: Benjamin Horvat, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:

- dovršavanje funkcionalnih zahtjeva
- raspisivanje svih use caseova
- Zadaci do idućeg sastanka:
  - pregled raspisanih use caseova

#### 4. sastanak

- Datum: 22. listopada 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - stvaranje UML dijagrama
  - podjela issuea za upisivanje dokumentacije u LaTeX
- Zadaci do idućeg sastanka:
  - riješiti issuee na GitLabu

#### 5. sastanak

- Datum: 27. listopada 2020.
- Prisustvovali: Benjamin Horvat, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - stvaranje frontend React projekta sa TypeScriptom
  - stvaranje backend Java Spring Boot projekta
  - stvaranje sekvencijskih dijagrama
  - stvaranje ER dijagrama baze podataka
- Zadaci do idućeg sastanka:
  - raspisati issuee
  - dovršiti ER i sekvencijske dijagrame

#### 6. sastanak

- Datum: 28. listopada 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - dovršavanje sekvencijskih dijagrama
  - dovršavanje ER dijagrama
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 7. sastanak

- Datum: 2. studenoga 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - objašnjavanje Reacta i Java Spring Boota
  - objašnjavanje i podjela issueea
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 8. sastanak

- Datum: 12. studenoga 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - dovršavanje dokumentacije za prvu predaju
  - probna prezentacija

## 9. sastanak

- Datum: 2. prosinca 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - ispravljanje grešaka nakon prve predaje
  - objašnjavanje i podjela issueea
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 10. sastanak

- Datum: 12. prosinca 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - rad na frontendu
  - objašnjavanje i podjela issueea
- Zadaci do idućeg sastanka:
  - raspisati issuee



## 11. sastanak

- Datum: 22. prosinca 2020.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - rad na backendu
  - objašnjavanje i podjela issueea
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 12. sastanak

- Datum: 2. siječnja 2021.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - rad na dokumentaciji
  - objašnjavanje i podjela issueea
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 13. sastanak

- Datum: 7. siječnja 2021.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - objašnjavanje i podjela issueea
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 14. sastanak

- Datum: 10. siječnja 2021.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - implementacija pretraživanja udruga, implementacija dodavanja, brisanja i dohvata životinja
  - objašnjavanje i podjela issuea
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 15. sastanak

- Datum: 11. siječnja 2021.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - rad na testiranju
  - dodana stranica za dodavanje i uređivanje životinja
  - objašnjavanje i podjela issuea
  - dodana karta na stranicu udruge
- Zadaci do idućeg sastanka:
  - raspisati issuee

## 16. sastanak

- Datum: 14. siječnja 2021.
- Prisustvovali: Benjamin Horvat, Iva Zekić, Petar Čovran, Mihael Rodek, Dora Doljanin, Vid Mužević, Karolina Mirković
- Teme sastanka:
  - dovršavanje dokumentacije
  - ispravljanje bugova

## Tablica aktivnosti

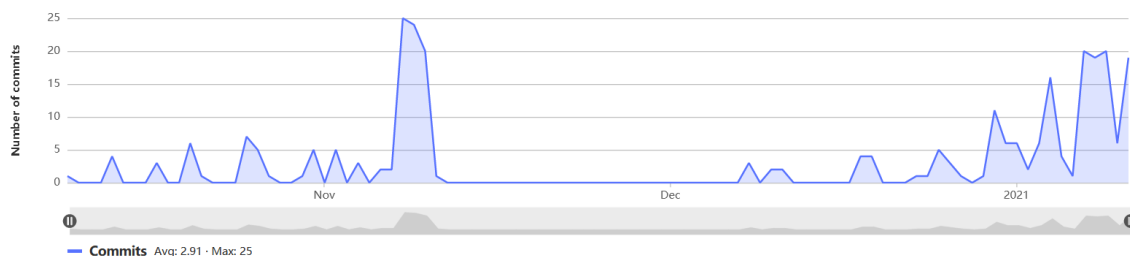
	Benjamin Horvat	Iva Zekić	Petar Čovran	Mihael Rodek	Dora Doljanin	Vid Mužević	Karolina Mirković
Upravljanje projektom	60	7	10	10	10	10	10
Opis projektnog zadatka		3					
Funkcionalni zahtjevi	4	4	5	4	6	4	4
Opis pojedinih obrazaca	4	4	4	4	5	4	4
Dijagram obrazaca		4	4		4		4
Sekvencijski dijagrami	1			4			5
Opis ostalih zahtjeva			1		1		1
Arhitektura i dizajn sustava					3		
Baza podataka	5		3		3		
Dijagram razreda	2		1		1		1
Dijagram stanja							1
Dijagram aktivnosti							1
Dijagram komponenti							3
Korištene tehnologije i alati			3				
Ispitivanje programskog rješenja	5					1	4
Dijagram razmještaja							4
Upute za puštanje u pogon		1		3			
Dnevnik sastajanja			1	1		1	4
Zaključak i budući rad					2		
Popis literature							
<i>izrada početne stranice</i>	2		3		7		
<i>izrada baze podataka</i>	3			1		3	
<i>spajanje s bazom podataka</i>	3						
<i>back end - login</i>	3						

	<b>Benjamin Horvat</b>	<b>Iva Zekić</b>	<b>Petar Čovran</b>	<b>Mihael Rodek</b>	<b>Dora Doljanin</b>	<b>Vid Mužević</b>	<b>Karolina Mirković</b>
<i>back end - registracija</i>	4					2	
<i>back end - statistika</i>		2					7
<i>back end - prijava šetnje</i>	2						
<i>back end - raspored</i>	2						
<i>back end - uređivanje životinja</i>						2	
<i>back end - entiteti</i>		4				1	
<i>front end - login</i>	4		3	2			
<i>front end - registracija</i>	5			2	3		3
<i>front end - statistika</i>							9
<i>front end - stranica udruge</i>	2		2	4	1		
<i>front end - prijava šetnje</i>	2			2			
<i>front end - raspored</i>	3		1				
<i>front end - profil</i>		5			2		
<i>front end - uređivanje profila</i>	4				1		
<i>izrada headera</i>	1				8		2
<i>deployment</i>	2			15			
<i>stvaranje projekta</i>	4						

## Dijagrami pregleda promjena

### Commits to master

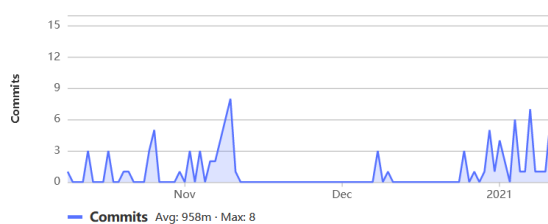
Excluding merge commits. Limited to 6,000 commits.



Slika 6.1: Ukupni commitovi

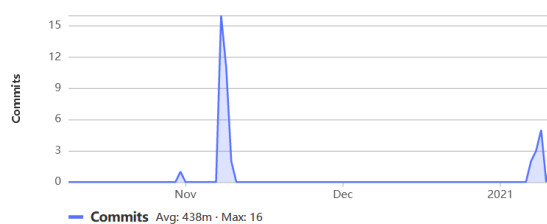
### Benjamin Horvat

92 commits (bh51358@fer.hr)



### Mihael Rodek

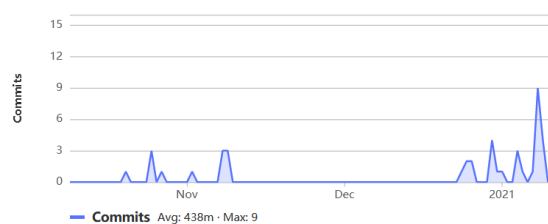
42 commits (mihael.rodek@fer.hr)



Slika 6.2: Commitovi: Benjamin Horvat i Mihael Rodek

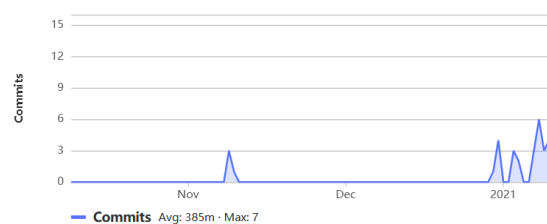
### Dora Doljanin

42 commits (48406565+doradoljanin@users.noreply.github.com)



### Karolina Mirković

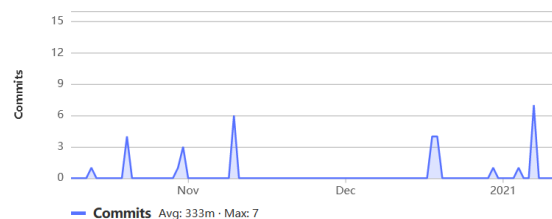
37 commits (km51437@fer.hr)



Slika 6.3: Commitovi: Dora Doljanin i Karolina Mirković

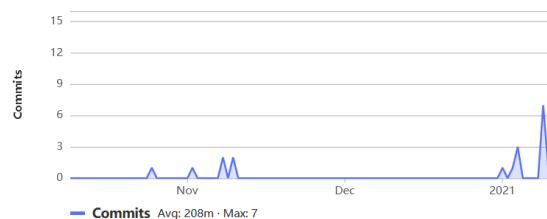
## Vid Muzevic

32 commits (vm51691@fer.hr)



## Petar Čovran

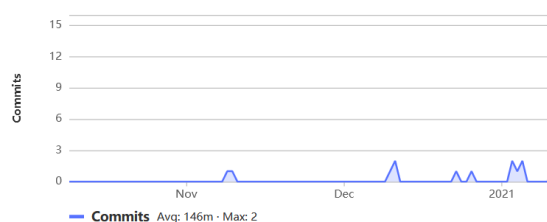
20 commits (pc51971@fer.hr)



Slika 6.4: Commitovi: Vid Mužević i Petar Čovran

## Iva Zekić

14 commits (iva.zekic@fer.hr)



Slika 6.5: Commitovi: Iva Zekić