

Programsko inženjerstvo

Ak. god. 2023./2024.

# Wild Track

Dokumentacija, Rev. 2.

Grupa: *Aristos*

Voditelj: *Josipa Udovičić*

Datum predaje: 19. siječnja 2024.

Nastavnik: *Hrvoje Nuić*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>7</b>
<b>3 Specifikacija programske potpore</b>	<b>11</b>
3.1 Funkcionalni zahtjevi . . . . .	11
3.1.1 Obrasci uporabe . . . . .	14
3.1.2 Sekvencijski dijagrami . . . . .	27
3.2 Ostali zahtjevi . . . . .	29
<b>4 Arhitektura i dizajn sustava</b>	<b>30</b>
4.1 Baza podataka . . . . .	31
4.1.1 Opis tablica . . . . .	33
4.1.2 Dijagram baze podataka . . . . .	41
4.2 Dijagram razreda . . . . .	42
4.3 Dijagram stanja . . . . .	46
4.4 Dijagram aktivnosti . . . . .	47
4.5 Dijagram komponenti . . . . .	49
<b>5 Implementacija i korisničko sučelje</b>	<b>51</b>
5.1 Korištene tehnologije i alati . . . . .	51
5.2 Ispitivanje programskog rješenja . . . . .	53
5.2.1 Ispitivanje komponenti . . . . .	53
5.2.2 Ispitivanje sustava . . . . .	54
5.3 Dijagram razmještaja . . . . .	63
5.4 Upute za puštanje u pogon . . . . .	64
<b>6 Zaključak i budući rad</b>	<b>73</b>
<b>Popis literature</b>	<b>74</b>
<b>Indeks slika i dijagonama</b>	<b>76</b>

**Dodatak: Prikaz aktivnosti grupe**

77

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Franjo Vuković	04.11.2023.
0.2	Dodani opis projektnog zadatka i funkcionalni zahtjevi.	Josipa Udovičić	04.11.2023.
0.2.1	Promijenjen opis i funkcionalni zahtjevi.	Josipa Udovičić	05.11.2023.
0.3	Dodan dio opisa obrazaca uporabe.	Stela Troškot	05.11.2023.
0.3.1	Dodani svi opisi obrazaca uporabe.	Stela Troskot, Domagoj Jurič	05.11.2023.
0.4	Dodani dijagrami obrazaca uporabe.	Franjo Vuković, Marko Kukolj, Marko Pongrac	05.11.2023.
0.5	Dodani sekvencijski dijagrami.	Franjo Vuković, Marko Kukolj, Marko Pongrac	08.11.2023.

Nastavljeno na idućoj stranici

Nastavljenod prethodne stranice

<b>Rev.</b>	<b>Opis promjene/dodataka</b>	<b>Autori</b>	<b>Datum</b>
0.5.1	Promijenjeni dijagrami obrazaca uporabe.	Stela Troskot, Domagoj Jurič	09.11.2023.
0.5.2	Dodan opis sekvencijskih dijagrama.	Marko Kukolj	09.11.2023.
0.5.3	Dodani ostali zahtjevi.	Josipa Udovičić	09.11.2023.
0.6	Dodan opis arhitekture.	Josipa Udovičić	09.11.2023.
0.7	Napravljena baza podataka.	Franjo Vuković, Marko Pongrac	09.11.2023.
0.7.1	Napravljen dijagram baze podataka.	Franjo Vuković, Marko Pongrac	09.11.2023.
0.7.2	Napisan opis baze podataka.	Gregor Miha- ljević	13.11.2023.
0.8	Dodan dio dijagrama razreda.	Franjo Vuković, Marko Pongrac, Marko Kukolj	15.11.2023.

Nastavljenona idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatak	Autori	Datum
0.8.1	Dodani svi dijagrami razreda.	Franjo Vuković, Marko Pongrac, Marko Kukolj	16.11.2023.
0.9	Opisi razreda i dodatak	Josipa Udovičić	16.11.2023.
1.0	Korigiranje teksta i provjera dokumentacije	Josipa Udovičić	17.11.2023.
1.1	Implementacija.	Josipa Udovičić, Franjo Vuković, Marko Pongrac, Marko Kukolj, Domagoj Jurić	16.01.2024.
1.2	Ispravljeni dijelovi prve predaje.	Stela Troskot, Josipa Udovičić, Franjo Vuković, Marko Kukolj	17.01.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.3	Dodan dijagram stanja i dijagram aktivnosti.	Stela Tro-skot	07.01.2024.
1.4	Dodan dijagram komponenti.	Stela Tro-skot	09.01.2024.
1.5	Dodan dijagram razmještaja.	Stela Tro-skot	11.01.2024.
1.6	Dodan opis tehnologija i alata.	Stela Tro-skot	13.01.2024.
1.7	Ispitivanje programskih rješenja.	Marko Pongrac	19.01.2024.
1.8	Dodane upute za puštanje u pogon.	Stela Tro-skot	19.01.2024.
1.9	Dodan zaključak.	Stela Tro-skot	19.01.2024.
2.0	Konačna verzija dokumentacije.	Stela Tro-skot	19.01.2024.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za web aplikaciju "Wild Track". Ta aplikacija olakšava korisniku pronađak i praćenje divljih životinja. Prilikom otvaranja aplikacije prikazuje se stranica za prijavu te opcija za registraciju.

Neregistriranom korisniku omogućeno mu je prijavljivanje u sustav s postojećim računom (potrebno je upisati korisničko ime i lozinku) ili kreiranje novog računa. Za kreiranje novog računa potrebni su sljedeći podaci:

- uloga za koju se prijavljuje - može biti istraživač, voditelj postaje ili tragač na terenu
- korisničko ime
- fotografija
- lozinka
- ime
- prezime
- email adresa

Registracijom u sustav korisniku se dodjeljuju prava istraživača, voditelja postaje, tragača na terenu ili administratora. Registrirani korisnik može, uz opcije koje ima njegova željena uloga, još i pregledati te mijenjati osobne podatke.

Registracija je završena kad korisnik preko svoje email adrese potvrdi, osim ako je korisnik izabrao biti istraživač ili voditelj postaje. U tom slučaju, administrator mora potvrditi njihovu ulogu. Administrator sustava ima najveće ovlasti. On ima ovlasti da vidi popis svih registriranih korisnika i njihovih osobnih podataka, odnosno pristup bazi s popisom registriranih korisnika, te može mijenjati njihova dodijeljena prava i osobne podatke.

Voditelj postaje može izabrati koji će tragači biti dio njegove postaje i bira na koji način oni mogu izvoditi pretraživanje životinja. Postaje su određena mjesta na karti koja admin bira kada potvrđuje voditelja, to na primjer može biti postaja Biokovo ili postaja Lonjsko polje. Voditelj može pregledati aktivne i neaktivne akcije. Na aktivne akcije može dodati tragače koji su dio njegove postaje. Tragači se na akciju mogu dodavati i nakon pregleda zahtjeva od istraživača. Načine transporta za koji su njegovi tragači sposobljeni voditelj može promijeniti u svakom trenutku.

Ako se korisnik prijavi kao tragač, njemu se na karti mogu prikazati zadaci koje mora obaviti, trenutna pozicija ostalih tragača koji su aktivni na istoj akciji, te trenutna pozicija životinja koje prati. Praćene životinje imaju na sebi gps uređaj koji aplikaciji odašilje njihovu poziciju i tako korisnik može cijelo vrijeme znati njihovu točnu lokaciju. Tragač tijekom svoje akcije može ostavljati komentare o životinji koju je pratilo te također može ostaviti komentar ostalim tragačima i istraživačima, to jest drugim sudionicima u akciji. Također, cijeli put koji tragači prođu se bilježi, odnosno označavaju se staze kojima su putovali i način kojim su se kretali. To će biti potrebno istraživačima koji vizualiziraju njihovo kretanje u obliku toplinskih karata te potom to koriste za analiziranje kretanja životinja i njihovih omiljenih staništa. Tragač može i pregledati podatke o pojedinačnim životnjama(latinski naziv, sliku, opis i komentare). Načini kojima se tragači kreću mogu biti različiti, kao na primjer:

- pješke
- dronom
- automobilom
- cross motorom
- brodom
- helikopterom

Ovisno o tome kojom je tragač metodom osposobljen za obavljanje zadataka, svaka metoda je drugačije prikazana na karti. Ako je tragač odabran da ide pješke njegova će karta biti detaljnija i prikazana na manjem prostoru nego karta tragača koji putuje helikopterom. Svaki tragač koristi samo jednu vozilo i tijekom te akcije se njegov tip prijevoza ne mijenja.

Ako je korisnik odlučio biti istraživač, on može stvoriti nove akcije pretraživanja i praćenja životinja s detaljima o određenim vrstama, jedinkama ili staništima za proučavanje. Svaki istraživač je zadužen za jednu akciju. Ako je istraživaču potreban tragač za pomoć pri istraživanju, istraživač može poslati voditelju postaje zahtjev za tragačima s opisom o potrebnim kvalifikacijama. Voditelj će na taj zahtjev odabrati tragače koji odgovaraju opisu i postaviti ih da sudjeluju u toj akciji. Tragač će biti gotov s akcijom kad završi sve potrebne zadatke. Istraživač, kad dobije određene tragače, zadaje preko karte zadatke pojedinačno svakom tragaču. Zadaci mogu biti različiti, kao na primjer prolazak određenom rutom i dolazak do određene lokacije te postavljanje kamere ili uređaja za praćenje. Prilikom postavljanja zadatka, istraživač može i ostaviti neke dodatne komentare tragačima.

Informacije o poziciji životinje, tragaču i postaji se istraživaču prikazuju preko interaktivne karte. Istraživač može birati da se tijekom izrade karte koriste i neke određene informacije. Na primjer:

- povijesne pozicije praćenih životinja
  - filtriranje životinja po vrsti ili pojedinačno po jedinki
  - trenutne pozicije praćenih životinja
  - povijesne pozicije svih tragača na nekoj akciji
  - filtriranje po tipu prijevoza ili pojedinačno po tragaču
  - trenutne pozicije tragača koji su aktivni na akciji

Ovaj projekt može biti korisan svima koji bi htjeli više naučiti o divljim životinjama, ali isto tako i ljudima koji se bave proučavanjem životinja može olakšati posao. Svaka osoba koja želi može se ulogirati i postati tragač, u svrhu zabave, dodatnog znanja... A time, dok ljudi to rade iz zabave, odnosno svojevoljno, osobe koji su istraživači mogu koristiti njihove informacije i time im se smanjuje dio posla.

Slične aplikacije poput aplikacije "Wild Track" na području Hrvatske ne postoje. Neke aplikacije koje se bave divljim životinjama su: "eWildLife" i "Divlje životinje".

"eWildLife" je aplikacija razvijena u realnom vremenu, ljudi prate i izvještavaju o ubijanju divljih životinja, sukobe čovjek-životinja i viđenja divljih životinja, također mogu i spasiti delfine.

"Divlje životinje" (Slika 2.1) je Android aplikacija koja omogućuje istraživanje svijeta životinja. Aplikacija nudi opsežan katalog divljih životinja, uključujući šumske životinje, životinje iz zooloških vrtova, cirkuske životinje, afričke životinje, šumske životinje itd. Aplikacija pruža detaljan opis svake životinje, uključujući njezinu vrstu, veličinu, kao i podatke o ponašanju i navikama. Također možete pristupiti raznim slikama koje će vam pomoći da bolje razumijete životinje.



Slika 2.1: Aplikacija Divlje Životinje

No, slična aplikacija "Wild Track" aplikaciji izvan Hrvatske postoji. Zove se "Animal Tracker" (Slika 2.2). S pomoću "Animal Tracker" aplikacije moguće je

pratiti kretanje divljih životinja diljem svijeta koje se prate u gotovo stvarnom vremenu. Kretanja se prikupljaju GPS oznakama koje životinje nose i pohranjuju se u Movebank (internetska infrastruktura koju koriste istraživači za upravljanje, dijeljenje, analizu i arhiviranje podataka o kretanju životinja).



Slika 2.2: Aplikacija Animal Tracker

Poboljšanja na aplikacijama se uvijek mogu napraviti, pa tako može i na ovoj. Na primjer, moguće je pratiti otkucaje srca životinja, disanje... Time je moguće određivati zdravlje životinje, kretanje životinje, da li životinja spava ili je budna... Tu smo dobili nove komponente aplikacije koje se mogu nazvati trenutne aktivnosti životinje i zdravlje životinje te tu tragači i istraživači imaju još više informacija o životnjama, a ne samo one općenite. Isto tako možemo dodati tragačima opciju da slikaju i snimaju životinje te im dati mogućnost da slike i videe objavljaju na aplikaciju. Time je opet doživljaj određene životinje bolji, a ne općenit. Aplikaciju bi svakako mogli poboljšati i tako da napravimo vizualno privlačniji i moderniji UI.

# 3. Specifikacija programske potpore

## 3.1 Funkcionalni zahtjevi

Dionici:

1. Tragač na terenu
2. Istraživač
3. Voditelj postaje
4. Administrator
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:

- (a) se registrirati u sustav
  - i. dati svoje podatke: korisničko ime, fotografija, lozinka, ime, prezime i email adresa
  - ii. odabrati svoju ulogu(tragač, istraživač ili voditelj postaje)
  - iii. potvrditi registraciju na svojoj email adresi

2. Korisnik (inicijator) može:

- (a) se prijaviti u sustav
- (b) upravljati svojim podatcima
  - i. pregled osobnih podataka
  - ii. promjena osobnih podataka
  - iii. brisanje korisničkog računa

3. Tragač (inicijator) na terenu može:

- (a) vidjeti zadatke tijekom akcije
  - i. označiti da je zadatak riješen
- (b) završiti akciju
- (c) vidjeti gdje se nalaze ostali tragači na akciji

- (d) vidjeti koje su dostupne akcije
  - i. priхватiti akciju
  - ii. odbiti akciju
- (e) vidjeti pozicije praćenih životinja
- (f) odabrati vozilo kojim će ići u istraživanje
- (g) vidjeti informacije o životinjama
  - i. dodati komentar
  - ii. izbrisati komentar

4. Istraživač (inicijator) može:

- (a) vidjeti informacije o životinjama
  - i. dodati komentar
  - ii. izbrisati komentar
- (b) vidjeti kartu
  - i. izraditi kartu: na temelju životinja ili tragača
- (c) stvoriti novu akciju
  - i. dodati nove zadatke
- (d) poslati voditelju postaje zahtjev za tragačima
  - i. dodati tražene kvalifikacije tragača
- (e) zadati zadatke tragačima
  - i. upravljati zadacima
    - A. dodavati zadatke
    - B. uređivati zadatke
    - C. brisati zadatke
  - ii. dodati komentare na zadatke
- (f) pregledati informacije

5. Voditelj postaje (inicijator) može:

- (a) pregledati sve tragače
- (b) dodati tragača na akciju
- (c) dodijeliti postaju tragaču

6. Administrator (inicijator) može:

- (a) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka

- (b) obrisati korisnika
- (c) mijenjati dodijeljena prava i osobne podatke registriranim korisnicima
- (d) potvrditi istraživača i voditelja postaje

7. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima
- (b) pohranjuje sve podatke o životinjama
- (c) pohranjuje staze kojima tragači putuju(i način kojim su se kretali)

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Registrirajte se"
  2. Korisnik bira ulogu i unosi potrebne korisničke podatke i odabire opciju "Potvrди"
  3. Korisnik prima obavijest na stranici da registraciju mora potvrditi na mailu
  4. Korisniku se šalje mail sa poveznicom za potvrdu registracije
  5. Korisnik klikne na poveznicu koja ga preusmjerava na početno sučelje aplikacije
  6. Ažurira se baza podataka
- **Opis mogućih odstupanja:**
  - 2.a Ako je odabrana uloga voditelja postaje ili istraživača automatski se pri potvrdi preko mail-a šalje zahtjev za potvrdu uloge administratoru
    1. Sustav obavještava korisnika da treba pričekati potvrdu uloge od administratora
  - 2.b Odabir već zauzetog korisničkog imena
    1. Sustav obavještava korisnika o neuspjelom pokušaju i vraća ga na stranicu za registraciju

##### UC2 - Prijava u sustav

- **Glavni sudionik:** Korisnik (voditelj postaje, tragač, istraživač)
- **Cilj:** Pristup registriranog korisnika korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena i lozinke
  2. Potvrda o ispravnosti unesenih podataka

3. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
    - 2.a Neispravno korisničko ime/lozinka
      1. Sustav obavještava korisnika o unosu krivih podataka pokušaju i vraća ga na stranicu za prijavu

### UC3 - Pregled svojih osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled svojih osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik bira opciju "Moj profil"
  2. Prikazuju se korisnikovi osobni podatci

### UC4 - Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati registrirane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Admin je prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Pregled korisnika"
  2. Prikaz liste svih ispravno registriranih korisnika i njihovih osobnih podataka

### UC5 - Brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji korisnički račun
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Pregled korisnika"
  2. Administrator odabire željenog korisnika iz prikazane liste registriranih korisnika
  3. Administrator odabire opciju "Izbriši korisnika"
  4. Baza podataka se ažurira

### **UC6 - Pregled osobnih podataka**

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan korisnički račun
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Pregled korisnika"
  2. Administrator odabire željenog korisnika iz prikazane liste registriranih korisnika
  3. Prikazuju se osobni podatci odabranog korisnika

### **UC7 - Promjena podataka i prava korisnika**

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti prava korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan korisnički račun
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Pregled korisnika"
  2. Administrator odabire željenog korisnika iz prikazane liste registriranih korisnika
  3. Administrator odabire opciju za "Promijeni podatke"
  4. Administrator mijenja željene podatke i/ili prava korisnika
  5. Administrator spremi promjene
  6. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 5.a Administrator nakon promjene podataka ne odabere opciju "Spremi"
    1. Promjene se ne spremaju

### **UC8 - Potvrda istraživača i voditelja postaje**

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrđuje ili odbija zahtjeve za određene uloge
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoje zahtjevi za odabrane uloge
- **Opis osnovnog tijeka:**
  1. Administratoru bira opciju "Pregled zahtjeva"

2. Bira opcije „Potvrdi“ ili „Odbij“
3. Baza podataka se ažurira
- 4.

### UC9 - Pregled svojih tragača

- **Glavni sudionik:** Voditelj postaje
- **Cilj:** Pregled vlastite postaje
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan tragač u postaji
- **Opis osnovnog tijeka:**
  1. Voditelj postaje odabire opciju "Pregled svojih tragača"
  2. Pojavljuje se lista njih

### UC10 - Uređivanje sposobnosti tragača

- **Glavni sudionik:** Voditelj postaje
- **Cilj:** Uređivanje sposobnosti tragača
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan tragač u postaji
- **Opis osnovnog tijeka:**
  1. Voditelj postaje odabire opciju "Pregled svojih tragača"
  2. Pojavljuje se lista njih
  3. Voditelj odabire tragača
  4. Voditelj odabire opciju "Uredi"
  5. Voditelj označuje sposobnosti za koje je tragač sposoban
  6. Baza podataka se ažurira

### UC11 - Pregled svih tragača

- **Glavni sudionik:** Voditelj postaje
- **Cilj:** Pregledati nesvrstane tragače
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoje nesvrstani tragači
- **Opis osnovnog tijeka:**
  1. Voditelj postaje odabire opciju "Pregled svih tragača"
  2. Pojavljuje se lista nesvrstanih tragača

### UC12 - Odabir tragača za postaju

- **Glavni sudionik:** Voditelj postaje
- **Cilj:** Iz nesvrstanih tragača voditelj postaje bira one za svoju postaju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik ima ulogu voditelja i postoje nesvrstani tragači
- **Opis osnovnog tijeka:**
  1. Voditelj bira opciju "Pregled svih tragača"
  2. Prikazuje se lista nesvrstanih tragača
  3. Voditelj odabire jednog ili više tragača za svoju postaju
  4. Voditelj dodjeljuje spodobnosti odabranima
  5. Voditelj odabire opciju "Potvrdi"
  6. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 5.a Voditelj nakon odabira tragača ne odabere opciju "Potvrdi"
    1. Promjene se ne spremaju

#### UC13 - Pregled akcija

- **Glavni sudionik:** Voditelj postaje
- **Cilj:** Pregled akcija u tijeku
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedna akcija
- **Opis osnovnog tijeka:**
  1. Voditelj postaje bira opciju "pregled akcija"
  2. Voditelj postaje odabire opciju "Aktivne akcije" ili "Neaktivne akcije"
  3. Pojavljuje se lista odabralih akcija

#### UC14 - Pregled zahtjeva

- **Glavni sudionik:** Voditelj postaje
- **Cilj:** Pregled zahtjeva
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji neispunjeli zahtjev
- **Opis osnovnog tijeka:**
  1. Voditelj postaje bira opciju "Pregled zahtjeva"
  2. Pojavljuje se lista neispunjениh zahtjeva

#### UC15 - Dodavanje tragača na akciju

- **Glavni sudionik:** Voditelj postaje

- **Cilj:** Dodati tragača na akciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji zahtjev za tragačima
- **Opis osnovnog tijeka:**
  1. Voditelj postaje odabire opciju "pregled zahtjeva" ili opciju "pregled akcija"
  2. Voditelj postaje bira zahtjev ili aktivnu akciju
  3. Prikazuje se lista dostupnih tragača njegove postaje (s traženim kvalifikacijama u slučaju zahtjeva)
  4. Voditelj postaje izabire tragača i odabire opciju "Dodaj"
  5. Baza podataka se ažurira (obrađeni zahtjev se briše)

#### UC16 - Pregled informacija o životinjama

- **Glavni sudionik:** Istraživač, tragač
- **Cilj:** Pregledati informacije o životinji
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedna životinja
- **Opis osnovnog tijeka:**
  1. Glavni sudionik odabire opciju "O životinjama"
  2. Glavni sudionik odabire željenu vrstu i jedinku životinje
  3. Prikazuju se informacije o životinji

#### UC17 - Dodavanje komentara o životinjama

- **Glavni sudionik:** Tragač ili istraživač
- **Cilj:** Dodavanje komentara o praćenoj životinji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je u ulozi tragača i pratio je životinju o kojoj piše komentar ili je istraživač koji je započeo akciju
- **Opis osnovnog tijeka:**
  1. Glavni sudionik odabire opciju "O životinjama"
  2. Glavni sudionik odabire željenu vrstu i jedinku životinje
  3. Prikazuju se informacije o životinji i tekstno polje za unos komentara
  4. Glavni sudionik upisuje komentar u predviđeno polje
  5. Glavni sudionik odabire opciju "Spremi komentar"
  6. Ažurira se baza podataka
- **Opis mogućih odstupanja:**

5.a Glavni sudionik nakon unosa komentara ne odabere opciju "Spremi komentar"

1. Promjene se ne spremaju

#### UC18 - Brisanje komentara

- **Glavni sudionik:** Istraživač
- **Cilj:** Brisanje komentara o životinji
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan komentar o životinji
- **Opis osnovnog tijeka:**
  1. Istraživač odabire opciju "O životinjama"
  2. Istraživač odabire željenu vrstu i jedinku životinje
  3. Prikazuju se informacije o životinji i komentari
  4. Istraživač odabire komentar koji želi izbrisati
  5. Istraživač odabire opciju brisanja komentara (oznaka "x")
  6. Ažurira se baza podataka i osvježava prikazana lista komentara

#### UC19 - Pregled zadataka tijekom akcije

- **Glavni sudionik:** Tragač
- **Cilj:** Pregled zadataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan zadatak u akciji
- **Opis osnovnog tijeka:**
  1. Tragač odabire opciju "Moja akcija"
  2. Tragač se uz ostale informacije o akciji prikazuje i popis zadataka
  3. Tragač odabire željeni zadatak
  4. Prikazuje mu se ruta zadatka na karti, pozicija praćene životinje i komentari istraživača i tragača o životinji zadatka

#### UC20 - Završavanje akcije

- **Glavni sudionik:** Istraživač
- **Cilj:** Označavanje akcije gotovom
- **Sudionici:** Baza podataka
- **Preduvjet:** Svi zadaci su označeni kao gotovi
- **Opis osnovnog tijeka:**
  1. Istraživač sa popisa akcija odabire željenu aktivnu akciju

2. Istraživač odabire opciju "Info" te potom opciju "Završi akciju"
  3. Istraživaču se pojavljuje *pop-up* pitanje "Želite li sigurno završiti akciju?"
  4. Istraživač potvrđuje završavanje akcije i sustav ga obavještava o uspjehu
  5. Ažurira se baza podataka
- **Opis mogućih odstupanja:**
    - 2.a postoji zadatak koji nije označen kao gotov
      1. ne dopušta se odabiranje opcije završetka zadatka

#### UC21 - Označavanje rješenih zadataka

- **Glavni sudionik:** Tragač
- **Cilj:** Označavanje zadatka gotovim
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Tragač odabire opciju "Moja akcija"
  2. Tragač se uz ostale informacije o akciji prikazuje i popis zadataka
  3. Tragač odabire opciju kvačice uz željeni zadatak i označava zadatak go-tovim
  4. Tekst odabranog zadatka na popisu mijenja boju u zelenu
  5. Ažurira se baza podataka

#### UC22 - Pregled ostalih tragača na akciji

- **Glavni sudionik:** Tragač
- **Cilj:** Pregled pozicija ostalih tragača na akciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem dva tragača na akciji
- **Opis osnovnog tijeka:**
  1. Tragač odabire opciju "Moja akcija"
  2. Tragaču se uz ostale informacije o akciji prikazuje i opcija "Ostali tragači na akciji"
  3. Odabirom te opcije prikazuju mu se lokacije ostalih tragača na karti

#### UC23 - Pregled zadataka

- **Glavni sudionik:** Istraživač
- **Cilj:** Pregled zadataka

- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan zadatak
- **Opis osnovnog tijeka:**
  1. Istraživač odabire akciju sa popisa akcija
  2. Pojavljuje se lista zadataka

#### UC24 – Slanje zahtjeva za tragačima

- **Glavni sudionik:** Istraživač
- **Cilj:** Slanje zahtjeva za tragačima
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji aktivna akcija
- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Istraživač odabire opciju "Info" zatim opciju "Zahtjev za tragačima"
  3. Istraživač unosi potrebne podatke
  4. Istraživač šalje zahtjev voditelju postaje
  5. Ažurira se baza podataka

#### UC25 – Izrada zadataka

- **Glavni sudionik:** Istraživač
- **Cilj:** Dodavanje zadatka na već postojeću akciju
- **Sudionici:** Baza podataka
- **Preduvjet:**
- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Istraživač izabire opciju "dodavanje zadataka"
  3. Istraživač bira akciju iz liste aktivnih akcija
  4. Istraživač unosi potrebne podatke, odabire tragača i početak i kraj rute zadatka
  5. Ažurira se baza podataka

#### UC26 – Brisanje zadataka

- **Glavni sudionik:** Istraživač
- **Cilj:** Brisanje zadataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan zadatak

- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Prikazuje se lista zadataka
  3. Istraživač odabire opciju "Obriši" uz željeni zadatak
  4. Ažurira se baza podataka

#### UC27 – Dodavanje komentara na zadatke

- **Glavni sudionik:** Istraživač
- **Cilj:** Dodavanje komentara na zadatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan zadatak
- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Prikazuje se lista zadataka
  3. Istraživač odabire opciju "komentari" uz željeni zadatak
  4. Istraživač unosi komentar u predviđeno tekstno polje
  5. Istraživač odabire opciju "Dodaj komentar"
  6. Ažurira se baza podataka
- **Opis mogućih odstupanja:**
  - 4.a 9. Istarživač nakon unosa ne odabere opciju „Dodaj komentar“
    1. Promjene se ne spremaju

#### UC28 – Stvaranje nove akcije pretraživanja

- **Glavni sudionik:** Istraživač
- **Cilj:** Stvaranje nove akcije pretraživanja i praćenja s detaljima o određenim vrstama, jedinkama ili staništima za proučavanje
- **Sudionici:** Baza podataka
- **Preduvjet:**
- **Opis osnovnog tijeka:**
  1. Istraživač odabire opciju „Dodaj novu akciju“ (oznaka "+")
  2. Ispunjava sve potrebne detalje o akciji
  3. Ažurira se baza podataka
  4. Istraživača se vraća na stranicu pregleda akcija

#### UC29 – Pregled karte na temelju tragača

- **Glavni sudionik:** Istraživač

- **Cilj:** Izrada karte na temelju tragača
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan tragač na akciji
- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Istraživač izabire opciju prikaži na karti: "Tragače"
  3. Prikazuje se lista tragača sa akcije
  4. Istraživač izabire željenog tragača
  5. Prikazuju mu se pozicije tragača na karti

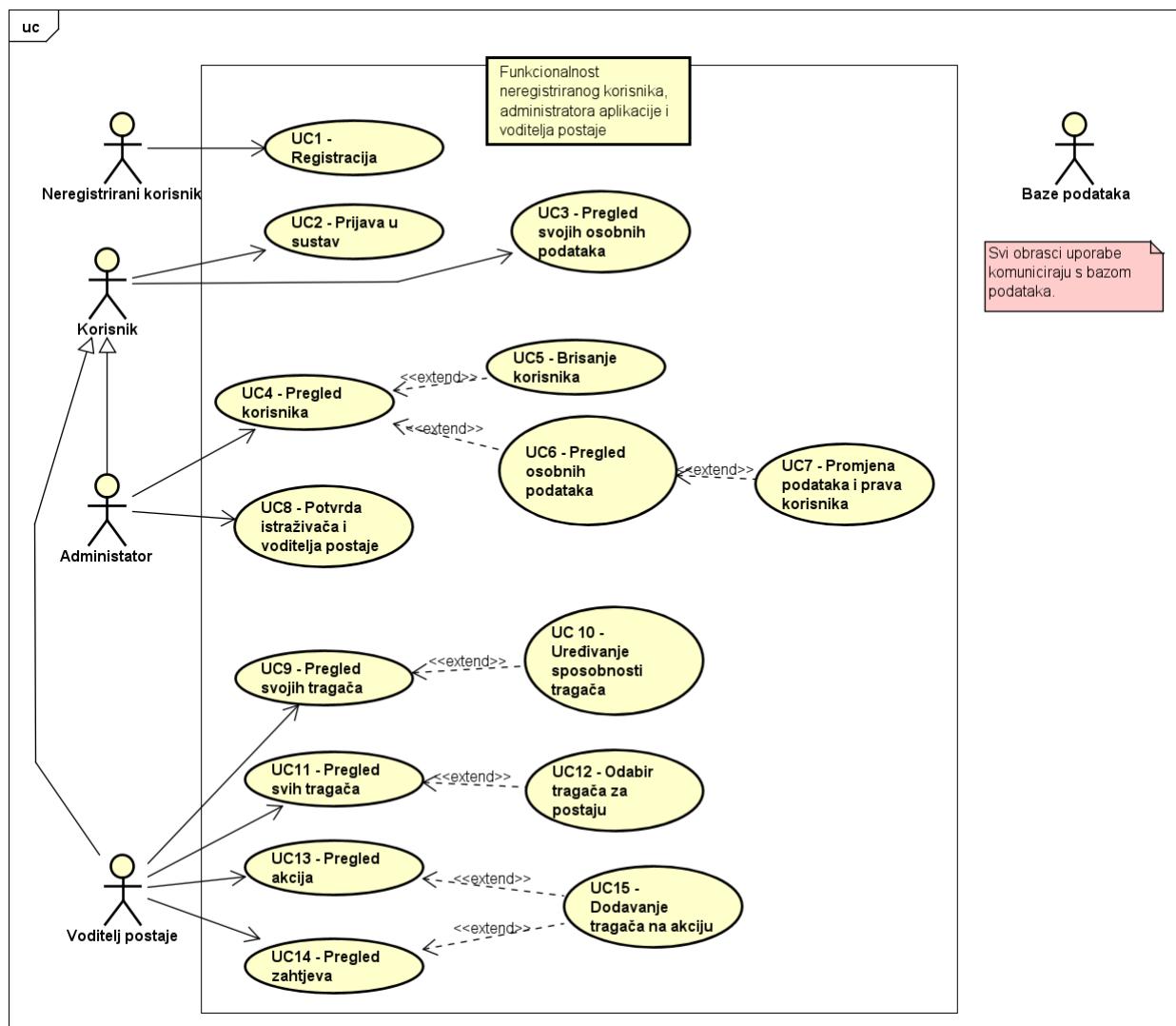
#### UC30 – Pregled karte na temelju životinja

- **Glavni sudionik:** Istraživač
- **Cilj:** Izrada karte na temelju životinja
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedna životinja
- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Istraživač izabire opciju prikaži na karti: "Životinje"
  3. Prikazuju mu se pozicije tragača na karti

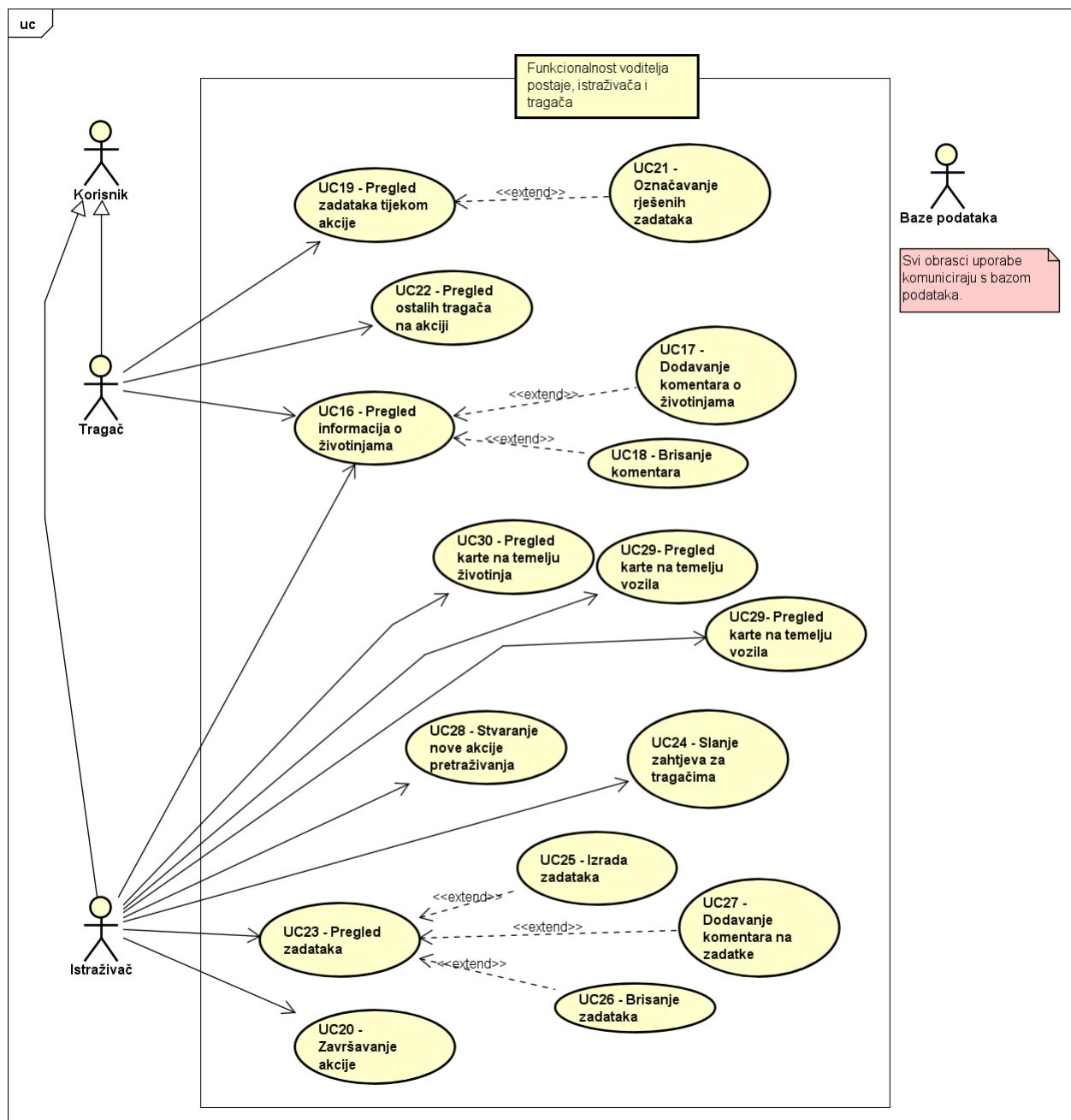
#### UC31 – Pregled karte na temelju vozila

- **Glavni sudionik:** Istraživač
- **Cilj:** Izrada karte na temelju vozila
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan tragač na akciji koji je obavio barem jedan zadatak
- **Opis osnovnog tijeka:**
  1. Istraživač izabire akciju sa popisa akcija
  2. Istraživač izabire opciju prikaži na karti: "Tragače"
  3. Prikazuje se lista tragača sa akcije i vozila na akciji
  4. Istraživač izabire željeno vozilo
  5. Prikazuju mu se povjesne pozicije vozila

## Dijagrami obrazaca uporabe



Slika 3.1: Funkcionalnost neregistriranog korisnika, generaliziranog korisnika, administratora aplikacije i voditelja postaje

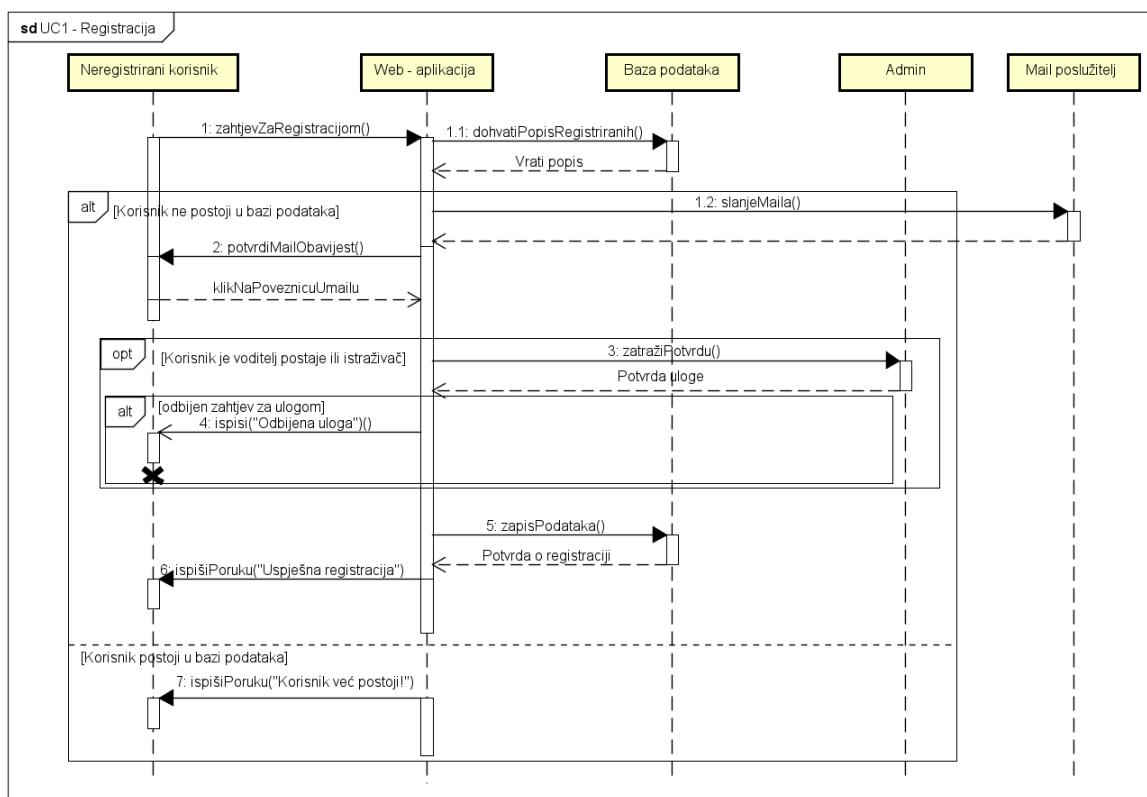


Slika 3.2: Funkcionalnost tragača i istraživača

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1 - Registracija

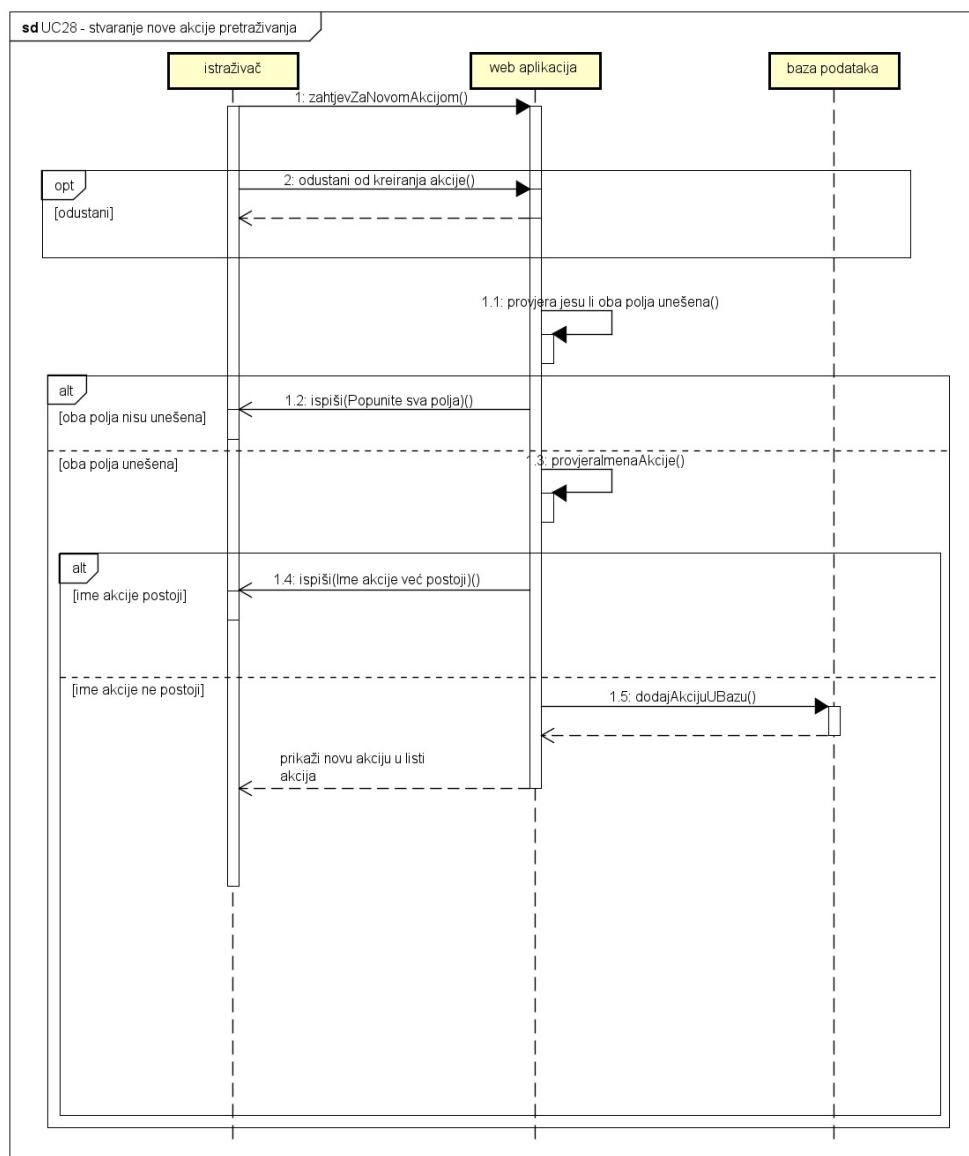
Neregistrirani korisnik šalje zahtjev za registracijom. Aplikacija dohvaća popis registriranih korisnika iz baze podataka. Ako korisnik nije na popisu aplikacija mu šalje potvrdu na mail preko mail-poslužitelja koju korisnik mora potvrditi. U slučaju da se korisnik registrira da bi bio voditelj postaje ili istraživač aplikacija šalje administratoru zahtjev za potvrdom te administrator mora potvrditi registraciju. U slučaju odbijanja zahtjeva za ulogom neregistriranom korisniku se ispisuje poruka o odbijanju. Zatim aplikacija zapisuje podatke u bazu podataka te dobiva potvrdu o uspjehu registracije od baze podataka te aplikacija korisniku prikazuje poruku o uspješnoj registraciji. Ako korisnik postoji u bazi podataka aplikacija, korisnik se obavještava da korisnik već postoji u bazi podataka.



Slika 3.3: Sekvencijski dijagram za UC1

### Obrazac uporabe UC28 - Stvaranje nove akcije pretraživanja

Istraživač odabire opciju stvaranja nove akcije. Prikazuje mu se formular za unos podataka o akciji. Nakon što Istraživač potvrdi stvaranje akcije provjerava se ispravnost podataka (postoji li već željeno ime akcije u bazi) i provjerava se jesu li sva potrebna polja ispunjena te se Istraživaču šalje obavijest ukoliko nešto nije zadovoljeno. Kada su uneseni svi podaci i zadovoljavaju uvjete, stvara se nova akcija i ažurira baza. Istraživač u svakom trenu može odustati od stvaranja nove akcije.



Slika 3.4: Sekvencijski dijagram za UC28

### 3.2 Ostali zahtjevi

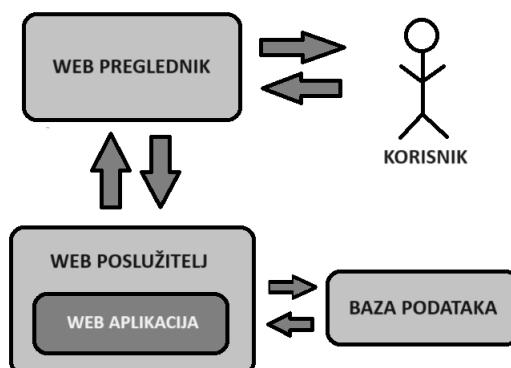
- Sustav treba podržavati istovremeni rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu
- Pristup bazi podataka trebao bi biti učinkovit, s vremenom izvršavanja unutar nekoliko sekundi
- Za izradu sustava kao web aplikacije, koriste se objektno-orientirani jezici
- Pogrešna uporaba korisničkog sučelja ne bi smjela imati negativan utjecaj na funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje
- Pri nadogradnji sustava, ne smiju se narušavati postojeće funkcionalnosti
- Veza s bazom podataka mora biti zaštićena, brza i otporna na vanjske greške
- Pristup sustavu treba biti moguć iz javne mreže, uz korištenje HTTPS-a radi sigurne komunikacije

## 4. Arhitektura i dizajn sustava

Arhitektura sustava može se podijeliti na tri ključna podsustava:

1. Web poslužitelj:
  - (a) Ključan dio web aplikacije.
  - (b) Odgovoran za interakciju između klijenta i aplikacije.
  - (c) Koristi HTTP/HTTPS protokol za prijenos informacija na webu.
  - (d) Inicira pokretanje web aplikacije i proslijeđuje zahtjeve.
2. Web aplikacija:
  - (a) Procesira korisničke zahtjeve i obrađuje ih.
  - (b) Pristupa bazi podataka prema potrebi.
  - (c) Generira odgovore u obliku HTML dokumenata za prikaz u web pregledniku.
3. Baza podataka:
  - (a) Sprema podatke koji se koriste ili modificiraju unutar web aplikacije.

Korisnik, putem web preglednika, šalje zahtjeve web poslužitelju. Web poslužitelj zatim inicira rad web aplikacije, koja procesira zahtjeve, pristupa bazi podataka po potrebi i vraća odgovore u obliku HTML dokumenata. Ova interakcija omogućuje korisnicima pregled i manipulaciju sadržajem putem web sučelja.



Slika 4.1: Arhitektura sustava

Za izradu ovog projekta koristili smo se Spring Boot frameworkom u Javi kroz razvojno okruženje IntelliJ Community Edition, Javascriptom uz React u Visual Studio Code-u te drugim programima za dizajn slika i grafova ( AstahUML itd.).

Spring Boot podržava koncept MVC, odnosno Model-Pogled-Nadglednik (engl. *Model View Controller*), arhitekture, tj. stilističke varijacije arhitekture zasnovane na događajima. Takve arhitekture odlikuje to što se komponente međusobno ne pozivaju eksplicitno, već neke od njih generiraju signale (događaje) ne znajući koja druga "osluškuje" tj. očekuje takav signal i na njega reagira. To se postiže kroz Spring Web MVC modul.

- **Model:** Spring Boot omogućava korištenje Java objekata kao modela. Ovi objekti predstavljaju podatke koji se koriste u aplikaciji. Spring Data može se integrirati za jednostavno upravljanje podacima i komunikaciju s bazom podataka.
- **View:** Spring Boot pruža fleksibilnost u odabiru tehnologije za prikazivanje korisničkog sučelja. Prikazi se često implementiraju kroz HTML datoteke, a moguće je koristiti različite template engines (Thymeleaf, FreeMarker, JSP). Pomoću konfiguracija view resolvera jednostavno se integriraju odrabane tehnologije za prikazivanje podataka korisnicima.
- **Controller:** Anotacije poput @Controller i @RestController omogućuju jednostavno označavanje klasa koje djeluju kao kontroleri. @RequestMapping i slične anotacije omogućuju mapiranje HTTP zahtjeva na određene metode kontrolera. Spring Boot automatski prepoznaće i konfigurira komponente kontrolera.

Kod MVC-a pogodno je što smanjuje međuvisnost korisničkog sučelja i osatka sustava, a omogućuje i nezavisni razvoj, nadogradnje i dodavanje različitih dijelova aplikacije. Sadrži različite gotove predloške za klase koji nam olakšavaju proces izrade.

## 4.1 Baza podataka

Baze podataka neizostavan su dio razvoja programske potpore jer danas gotova svaka domena primjene obiluje mnoštvom podataka koje treba pohraniti na organiziran način kako bi se efikasno dohvaćali, mijenjali i nadopunjivali. Za upravljanje bazom podataka mogu se koristiti različiti sustavi koji obavljaju optimiranje upita i omogućuju rukovanje podatcima. Mi smo odlučili koristiti PostgreSQL koji

nam je bio preporučen na kolegiju Baze podataka.

Relacijski nam model baze podataka omogućuje vjeran prikaz stvarnosti pomoću relacija u koje pohranjujemo vrijednosti odabralih atributa vezanih uz entitete bitne za domenu primjene. Atributi su imenovani stupci te tablice. ER (Entity-Relationship) model podataka zadržava dobra svojstva relacijskog modela, a uz to omogućuje eksplizitni prikaz semantičkih informacija vezanih uz veze (odnose) između entiteta. Kako bismo prikazali kako su entiteti našeg sustava povezani koristiti ćemo ER model baze podataka. za podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Pozicija tragača
- Uloga
- Zadatak
- Pripada postaji
- Ospozobljen za
- Akcija
- Postaja
- Prijevozno sredstvo
- Komentar korisnika
- Životinja
- Pozicija životinje

#### 4.1.1 Opis tablica

**Zadatak** Ovaj entitet sadrži sve važne informacije o zadatcima koje obavljaju tragači tijekom akcije, a stvaraju istraživači. Sadrži atribute: šifra zadatka, korisničko ime, tekst, završen, početna lokacija, krajnja lokacija šifra akcije, šifra životinje i šifra vozila. Ova tablica je u vezi s tablicom Prijevozno sredstvo preko tablice Osposobljen za te izravno preko identifikatora vozila(šifra vozila), s tablicom Akcija izravno preko identifikatora akcije (šifra akcije), s tablicom Životinja izravno preko identifikatora životinje (šifra životinje).

Zadatak		
Šifra Zadatka	INT	Jedinstveni brojčani identifikator zadatka
Tekst	VARCHAR	Opis zadatka
Završen	BOOLEAN	Status je li zadatak završen
Početna lokacija	DOUBLE	Početak rute kojom tragač mora proći
Krajnja lokacija	DOUBLE	Kraj rute kojom tragač mora proći
Korisničko ime	VARCHAR	Korisničko ime korisnika
Šifra akcije	INT	Jedinstveni brojčani identifikator akcije
Šifra životinje	INT	Jedinstveni brojčani identifikator životinje
Šifra vozila	INT	Jedinstveni brojčani identifikator vozila

**Akcija** Ovaj entitet sadrži sve važne informacije o akcijama pretraživanja i praćenja. Sadrži atribute: šifra akcije, naziv akcije, aktivna, započeta, početak akcije, kraj akcije, poslan zahtjev i korisničko ime. Ova tablica je u vezi s tablicom Korisnik preko identifikatora korisnika(korisničko ime), s tablicom Životinja preko tablice Komentar korisnika, s tablicom Zadatak i tablicom Pozicija tragača.

Akcija		
Šifra akcije	INT	Jedinstveni ključ za identifikaciju zadatka

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Akcija		
Naziv akcije	VARCHAR	Puni naziv akcije
Aktivna	BOOLEAN	Status je li akcija aktvna
Započeta	BOOLEAN	Status je li akcija započela
Početak akcije	TIMESTAMP	Kad je akcija započela
Kraj akcije	TIMESTAMP	Kad je akcija završila
Poslan zahtjev	BOOLEAN	Ograničava istraživačima slanje zahtjeva za tragačima
Korisničko ime	VARCHAR	Korisničko ime korisnika

**Komentar korisnika** Ovaj entitet sadrži informacije o komentarima korisnika o praćenoj životinji koje mogu ostaviti tijekom akcije. Sadrži atribute: šifra komentara, šifra životinje, korisničko ime, šifra akcije i komentar.

Komentar korisnika		
Šifra komentara	INT	Jedinstveni brojčani identifikator komentara
Komentar	VARCHAR	Sadržaj komentara
Šifra životinje	INT	Jedinstveni brojčani identifikator zadatka
Korisničko ime	VARCHAR	Korisničko ime korisnika
Šifra akcije	INT	Jedinstveni brojčani identifikator akcije

**Životinja** Ovaj entitet sadrži sve važne informacije o životnjama. Sadrži atribute: šifra životinje, naziv , latinski naziv i opis. Ova tablica povezana je s tablicom Pozicija životinje s tablicom Zadatak preko identifikatora životinje(šifra životinje) te s tablicom Akcija

Životinja		
Šifra životinje	INT	Jedinstveni brojčani identifikator životinje
Naziv	VARCHAR	Puni naziv životinje
Latinski naziv	VARCHAR	Latinski naziv životinje
Opis	VARCHAR	Sadržaj komentara

**Pozicija životinje** Ovaj entitet sadrži informacije o poziciji životinje. Pomoću njega bilježimo gibanje životinje na kartama. Sadrži atribute: šifra životinje, vremenska oznaka, geografska širina i geografska dužina.

Pozicija životinje		
Šifra životinje	INT	Jedinstveni brojčani identifikator životinje
Vremenska oznaka	TIMESTAMP	Zadnje vrijeme u kojem je viđena životinja
Geografska širina	DOUBLE	Iznos geografske širine
Geografska dužina	DOUBLE	Iznos geografske dužine

**Pozicija tragača** Ovaj entitet sadrži informacije o poziciji tragača. Pomoću njega bilježimo kretanje tragača na kartama. Sadrži atribute: korisničko ime, vremenska oznaka, šifra akcije, geografska širina i geografska dužina.

Pozicija tragača		
Korisničko ime	VARCHAR	Korisničko ime korisnika
Vremenska oznaka	TIMESTAMP	Zadnje vrijeme kada je tragač zabilježio svoju poziciju
Šifra akcije	INT	Jedinstveni brojčani identifikator akcije

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Pozicija tragača		
Geografska širina	DOUBLE	Iznos geografske širine
Geografska dužina	DOUBLE	Iznos geografske dužine

**Korisnik** Ovaj entitet sadrži sve važne informacije o korisnicima. Sadrži atribute: korisničko ime, email, lozinka, ime, prezime, fotografija, provjeren mail, potvrdio admin i šifra uloge. Ova tablica je u vezi s tablicama Akcija, Pozicija tragača, Uloga i Komentar korisnika.

Korisnik		
Korisničko ime	VARCHAR	Ime korisnika
Email	VARCHAR	Sadržaj komentara
Lozinka	VARCHAR	Lozinka korisnika
Ime	VARCHAR	Ime korisnika
Prezime	VARCHAR	Prezime korisnika
Fotografija	VARCHAR	Fotografija korisnika
Provjeren email	BOOLEAN	Provjera je li osoba potvrdila mail
Potvrđio admin	BOOLEAN	Voditelja i istraživača mora potvrditi admin
Šifra uloge	INT	Jedinstveni brojčani identifikator uloge

**Pripada postaji** Ova veza sadrži informacije po kojima saznajemo koji korisnik pripada kojoj postaji. Povezuje entitete Korisnik i Postaja te sadrži njihove ključne atribute: korisničko ime i šifra postaje.

Prijava postaji		
Korisničko ime	VARCHAR	Korisničko ime korisnika
Šifra postaje	VARCHAR	Jedinstveni brojčani identifikator postaje

**Postaja** Ovaj entitet sadrži informacije o postajama. Sastoji se od atributa: šifra postaje i naziv postaje.

Postaja		
Šifra postaje	VARCHAR	Jedinstveni brojčani identifikator korisnika
Naziv postaje	VARCHAR	Pun naziv postaje

**O sposobljen za** Ova veza sadrži informacije po kojima saznajemo koji korisnik ima sposobnosti za voziti koje prijevozno sredstvo. Povezuje entitete Korisnik i Prijevozno sredstvo te sadrži njihove ključne attribute: korisničko ime i šifra vozila.

O sposobljen za		
Korisničko ime	VARCHAR	Ime korisnika
Šifra vozila	INT	Jedinstveni brojčani identifikator vozila

**Prijevozno sredstvo** Ovaj entitet sadrži informacije o vozilima za koje tragači mogu biti osposobljeni. Sastoji se od atributa: šifra vozila i naziv vozila.

Prijevozno sredstvo		
Šifra vozila	INT	Jedinstveni brojčani identifikator vozila
Naziv vozila	VARCHAR	Puni naziv vozila

**Uloga** Ovaj entitet sadrži informacije o ulogama korisnika. Sastoji se od atributa: šifra uloge i naziv uloge.

Uloga		
Šifra Uloge	INT	Jedinstveni brojčani identifikator uloge
Naziv uloge	VARCHAR	Puni naziv uloge

**Konfirmacijski token** Ovaj entitet sadrži informacije o konfirmacijskim tokenima. Sastoji se od atributa: šifra tokena, vrijeme potvrde, vrijeme stvaranja, vrijeme isteka, token i korisničko ime. Povezana je s tablicom Korisnik preko identifikatora korisnika(korisničko ime).

Konfirmacijski token		
Šifra tokena	INT	Jedinstveni brojčani identifikator tokena
Vrijeme potvrde	TIMESTAMP	Vrijeme potvrde emaila
Vrijeme stvaranja	TIMESTAMP	Vrijeme stvaranja emaila
Vrijeme isteka	TIMESTAMP	Vrijeme isteka emaila
Token	VARCHAR	Token za potvrdu maila
Korisničko ime	VARCHAR	Korisničko ime korisnika

**Komentari zadatka** Ovaj entitet sadrži informacije o komentarima zadataka. Sastoji se od atributa: šifra komentara, komentar, šifra zadatka i korisničko ime. Povezana je s tablicom Zadatak preko identifikatora zadatka(šifra zadatka) i s tablicom Korisnik preko identifikatora korisnika(korisničko ime).

Komentari zadatka		
Šifra komentara	INT	Jedinstveni brojčani identifikator komentara
Komentar	VARCHAR	Tekst komentara
Šifra zadatka	INT	Jedinstveni brojčani identifikator zadatka

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Komentari zadatka		
Korisničko ime	VARCHAR	Korisničko ime istraživača

**Komentari tragača** Ovaj entitet sadrži informacije o komentarima tragača. Sastoji se od atributa: šifra komentara, komentar, šifra akcije i korisničko ime. Povezana je s tablicom Akcija preko identifikatora akcije(šifra akcije) i s tablicom Korisnik preko identifikatora korisnika(korisničko ime).

Komentari tragača		
Šifra komentara	INT	Jedinstveni brojčani identifikator komentara
Komentar	VARCHAR	Tekst komentara
Šifra akcije	INT	Jedinstveni brojčani identifikator akcije
Korisničko ime	VARCHAR	Korisničko ime tragača

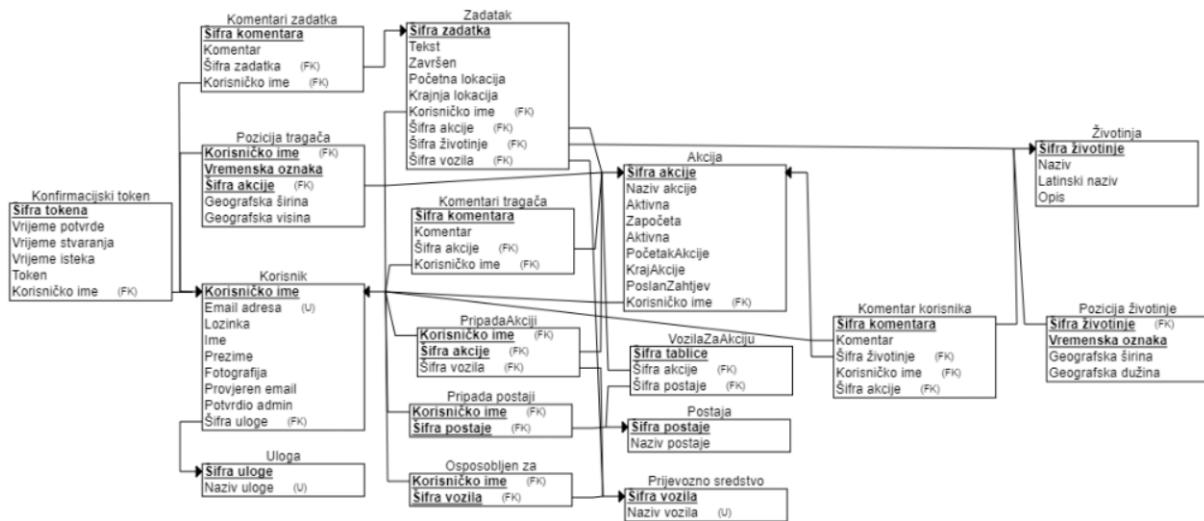
**Pripada akciji** Ova veza sadrži informacije po kojima saznajemo koji tragač pripada kojoj akciji. Povezuje entitete Korisnik i Akcija te sadrži njihove ključne attribute: korisničko ime i šifra akcije. Povezana je s tablicom Prijevozno sredstvo preko identifikatora vozila(šifra vozila).

Pripada akciji		
Šifra akcije	INT	Jedinstveni brojčani identifikator akcije
Korisničko ime	VARCHAR	Korisničko ime tragača
Šifra vozila	INT	Jedinstveni brojčani identifikator vozila

**Vozila za akciju** Ova veza sadrži informacije po kojima saznajemo koja vozila pripadaju kojoj akciji. Povezuje entitete Akcija i Prijevozno sredstvo te sadrži njihove ključne attribute: šifra akcije i šifra postaje. Još sadrži i atribut šifra tablice koji je jedinstveni brojčani identifikator tablice.

<b>Vozila za akciju</b>		
Šifra tablice	INT	Jedinstveni brojčani identifikator tablice
Šifra akcije	INT	Jedinstveni brojčani identifikator akcije
Šifra postaje	INT	Jedinstveni brojčani identifikator postaje

#### 4.1.2 Dijagram baze podataka

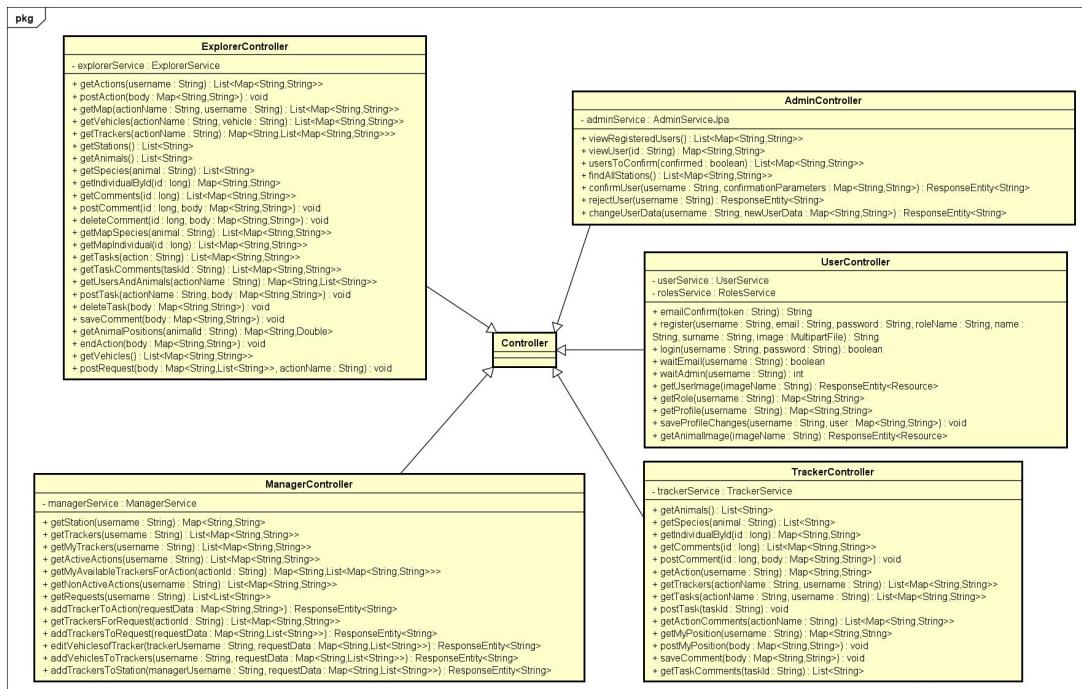


Slika 4.2: Dijagram baze podataka

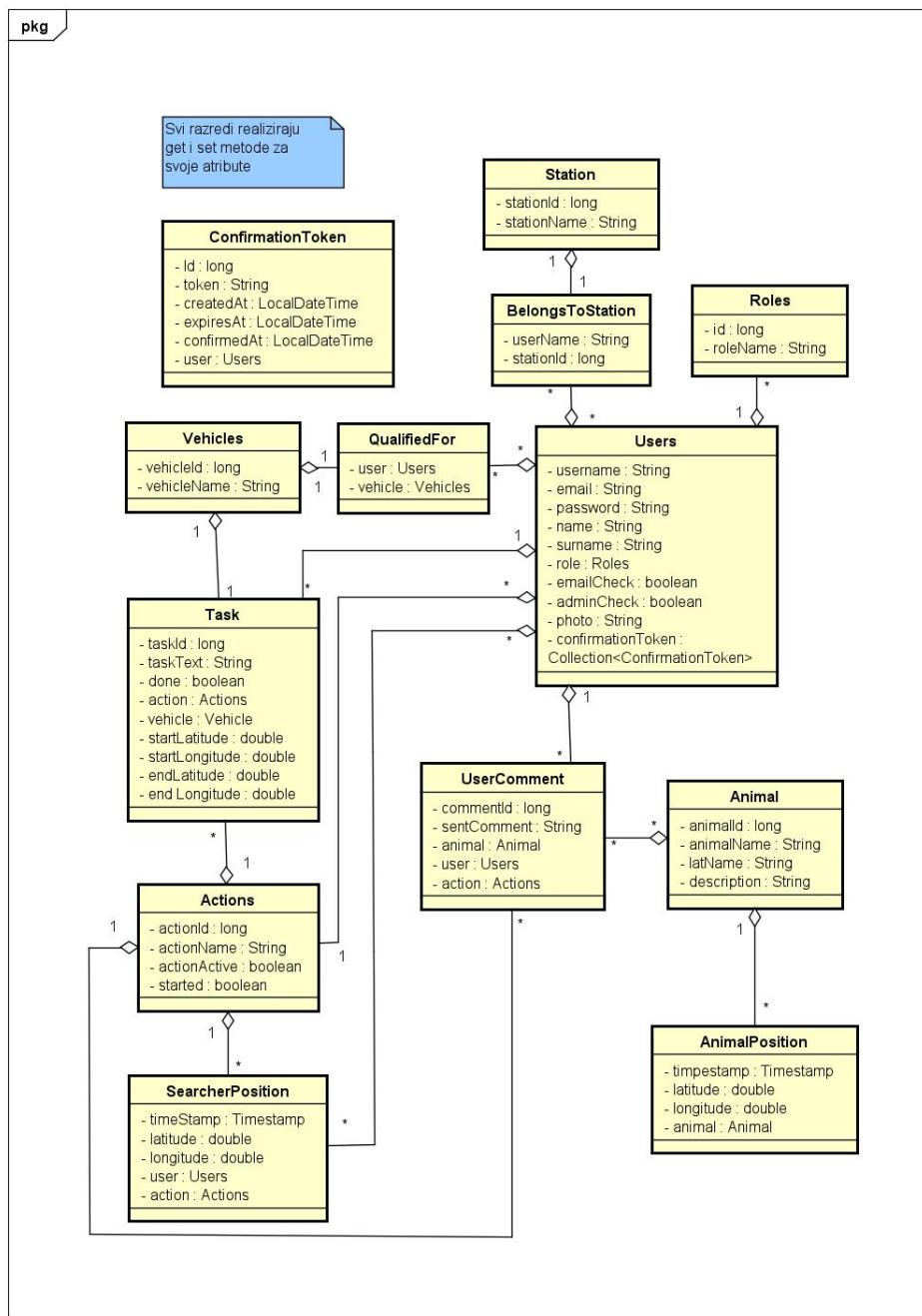
## 4.2 Dijagram razreda

Na slikama 4.3, 4.4 i 4.5 su prikazani klasni dijagrami koji pripadaju backend dijelu MVC arhitekture. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Metode implementirane u tim razredima manipuliraju s Data transfer object, a oni su dohvaćeni s pomoću metoda u Model razredima.

Metode unutar Controller razreda vraćaju JSON datoteke. Razredi su logički podijeljeni prema pravu pristupa metodama specifičnih aktora kako bi se smanjila prenapučenost unutar dijagrama. Pri prikazu su naglašene samo ovisnosti između razreda koji pripadaju istom dijelu dijagrama radi jednostavnije organizacije. Vrsta ovisnosti između različitim razredima može se zaključiti iz naziva i tipova atributa unutar tih razreda. Controller razredi sadrži privatne varijable koje im omogućuju komunikaciju sa Service razredima.

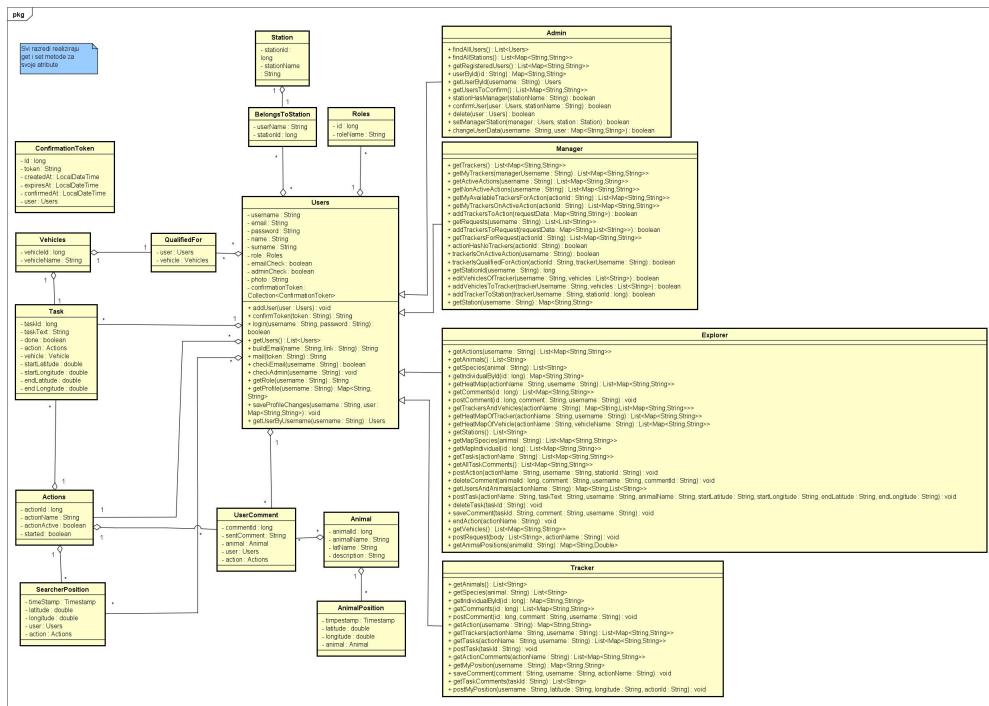


Slika 4.3: Dijagram razreda - dio Controllers



Slika 4.4: Dijagram razreda - dio Data transfer objects

Model razredi prikazuju strukturu baze podataka u aplikaciji. Implementirane metode komuniciraju s bazom podataka te vraćaju tražene podatke. Razred Users predstavlja neregistriranog korisnika koji se može registrirati u sustav unošenjem informacija.

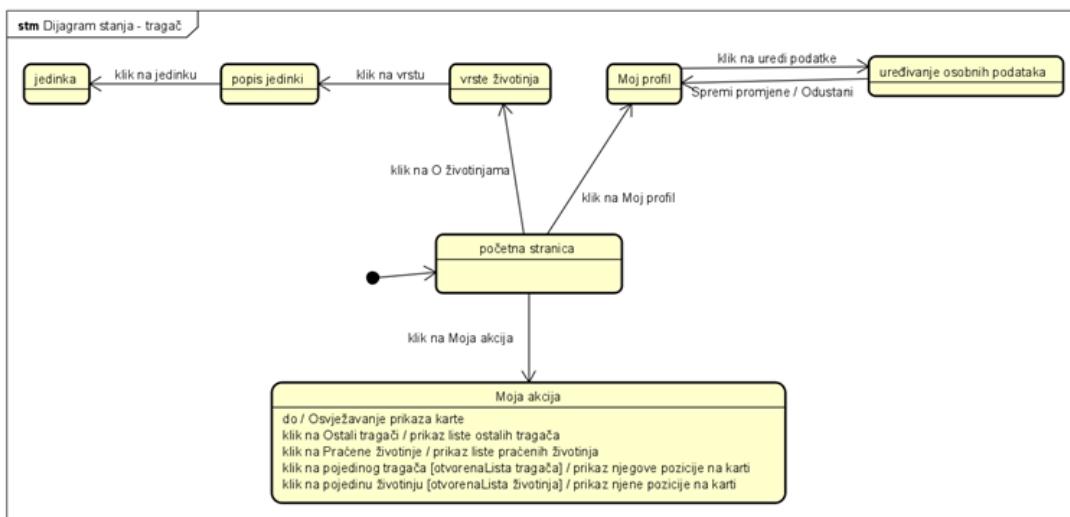


Slika 4.5: Dijagram razreda - dio Models

## 4.3 Dijagram stanja

Dijagram stanja (slika 4.6) opisuje dinamičko ponašanje sustava uslijed raznih mogućih događaja. Uspješnom prijavom prikazana je početna stranica korisnika u ulozi tragača. Na početnoj stranici tragač bira jednu od tri opcije:

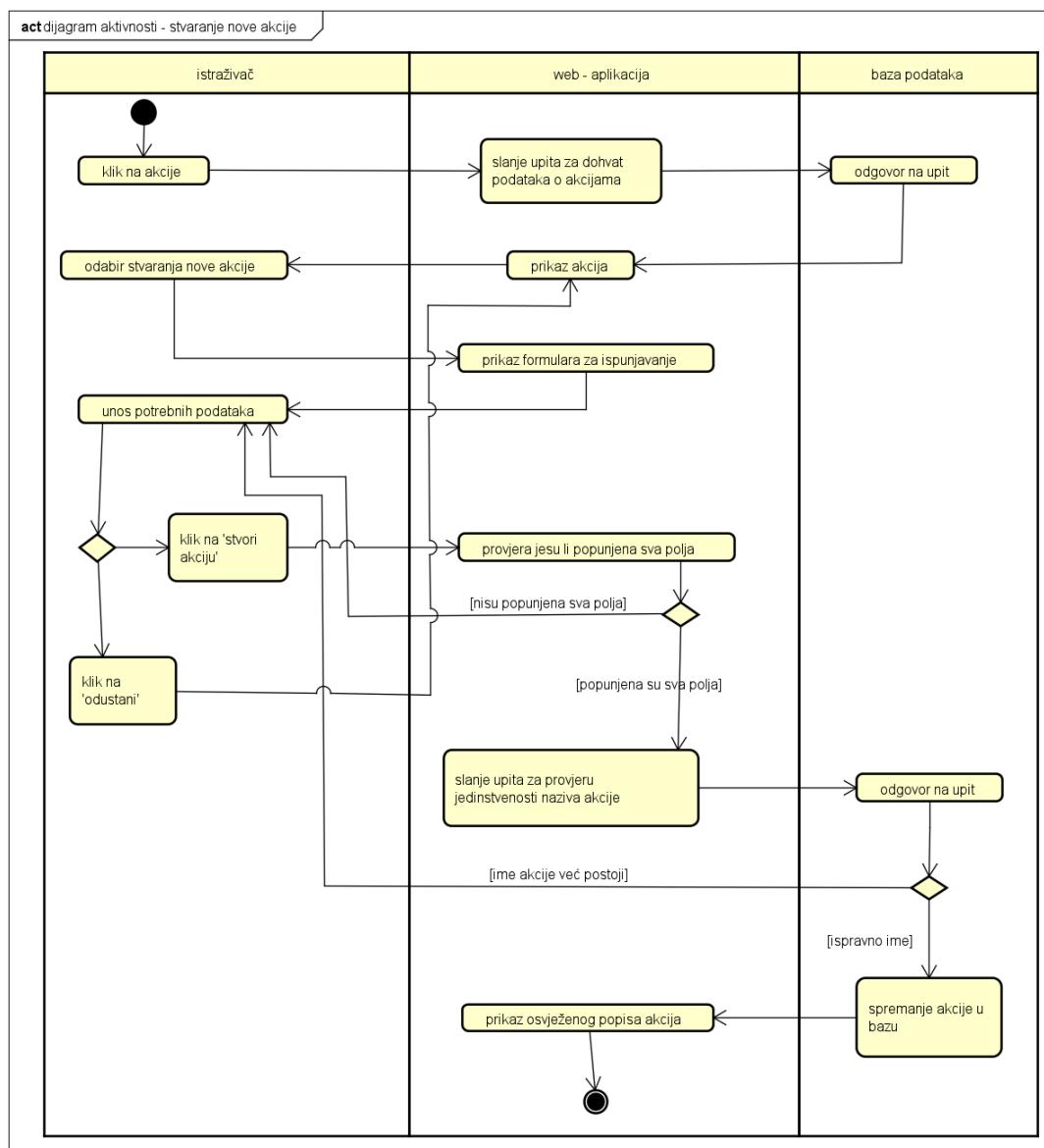
- opciju *Moj profil* dostupnu svakom korisniku, koja ga vodi na stranicu sa osobnim podacima, gdje ih može uređivati
- opciju *O životinjama* koja ga vodi na stranicu na kojoj su prikazane sve vrste životinja u aplikaciji, a klikom na pojedinu vrstu dobiva popis jedinki te vrste i dalje klikom na pojedinu jedinku dobiva opširnije informacije specifično o toj jedinki
- opciju *Moja akcija* koja ga vodi na stranicu s prikazom karte i dodatnim opcijama za pregled zadataka, ostalih tragača i praćenih životinja; nakon klika na praćene životinje ili ostale tragače, tragaču se prikazuje lista istih, a klikom na pojedinca te liste(tragača ili životinju) na karti mu se prikazuje trenutna pozicija odabranog



Slika 4.6: Dijagram stanja - tragač

## 4.4 Dijagram aktivnosti

Dijagrom aktivnosti na slici 4.7 modelirano je stvaranje nove akcije korisnika u ulozi istraživača. Do postupka stvaranja nove akcije dolazi se sa stranice prikaza popisa akcija klikom na *Stvori novu akciju*. Na formularu za novu akciju se prilikom klika na *Stvori novu akciju*, prije spremanja podataka nove akcije u bazu, provjera je su li uneseni svi potrebni podatci i jesu li uneseni podatci ispravni. Nakon uspješnog stvaranja nove akcije, korisniku se prikazuje osvježen popis akcija, a ukoliko korisnik odustane od stvaranja nove akcije klikom na *Odustani* prikaz se vraća na popis akcija.

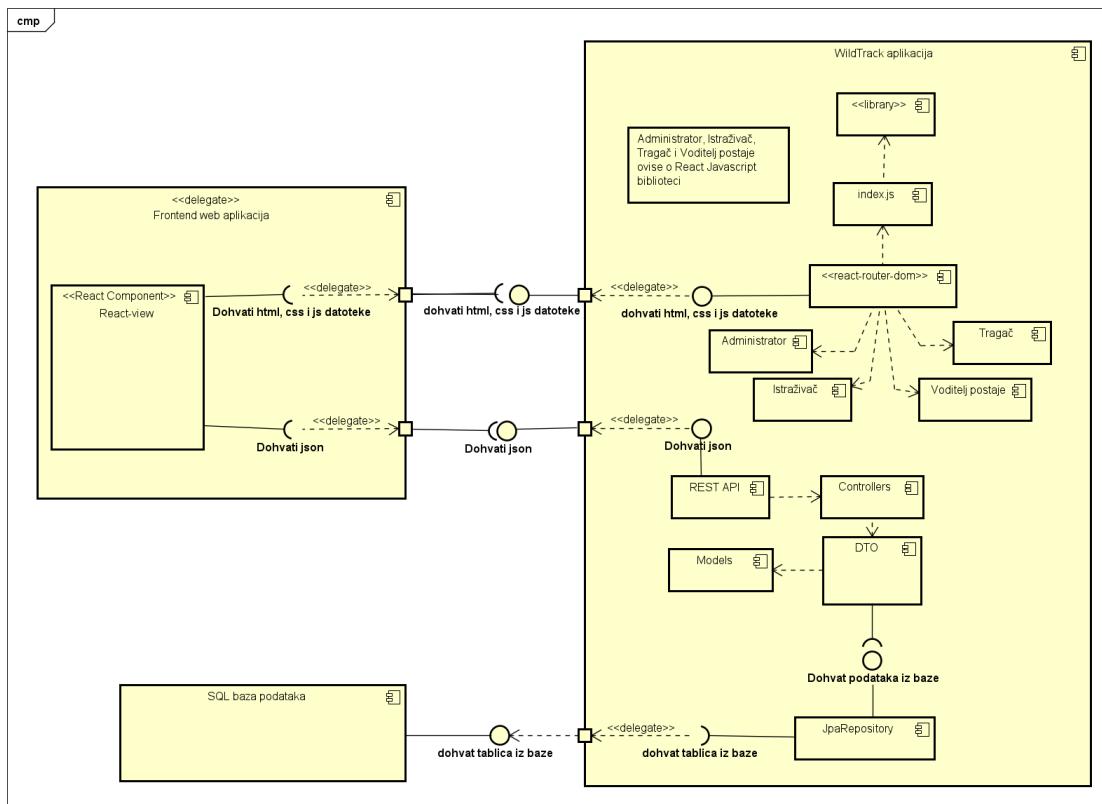


Slika 4.7: Dijagram aktivnosti - stvaranje nove akcije

## 4.5 Dijagram komponenti

UML-dijagram komponenti je vrsta struktturnog UML-dijagrama koji prikazuje organizaciju i odnose komponenti koje čine programsku potporu. Pruža vizualni prikaz arhitekture sustava, naglašavajući modularnu strukturu i interakcije između komponenti, interne strukture i okoline.

Sustavu se pristupa preko dva razlicita sučelja. Preko sučelja za dohvatzanje HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju *frontend* dijelu aplikacije. Router je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje. Frontend dio se sastoji od niza JavaScript datoteka koje su raspoređene u logičke cjeline nazvane po tipovima aktora koji im pristupaju. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente kao što su gumbi, forme i slično. Preko sučelja za dohvatzanje JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju *backend* dijelu aplikacije. JpaRepository je sučelje Spring Data JPA projekta korišteno za jednostavan dohvatzanje i manipulaciju s podacima u bazi podataka, pružajući apstrakciju nad JPA (Java Persistence API) omogućavajući izvođenje operacija nad entitetima bez potrebe za izravnim pisanjem SQL upita. Podaci koji su pristigli iz baze se šalju dalje MVC arhitekturi u obliku DTO (Data transfer object). React-view komponenta preko dostupnih sučelja komunicira sa WildTrack aplikacijom te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke.



Slika 4.8: Dijagram komponenti

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

U izradi projekta većina digitalne komunikacije ostvarena je preko platforme **WhatsApp** (<https://web.whatsapp.com/> ).

Korištene su sljedeće tehnologije i alati:

- **Springboot**
  - open source radni okvir za kreaciju mikro-servisa, idealan za potrebe projekta
  - u njemu je izrađen backend dio projekta
  - <https://spring.io/projects/spring-boot>
- **React**
  - JavaScript biblioteka za izgradnju korisničkih sučelja
  - korišten za izradu čitavog frontenda dijela aplikacije s kojim korisnik dolazi u interakciju
  - provjerava ispravnost podataka unesenih u obrasce
  - <https://www.reactjs.org/>
- **PostgreSQL**
  - jezik u kojem je napravljena baza podataka
  - besplatan i open source sustav za upravljanje relacijskim bazama podataka s naglaskom na mogućnosti proširivanja
  - <https://www.postgresql.org/>
- **pgAdmin**
  - za upravljanje bazom podataka neovisno o backendu
  - <https://www.pgadmin.org/>
- **IntelliJ**
  - IDE za javu u kojem je kodiran backend dio projekta
  - <https://www.jetbrains.com/idea/>

- **Visual Studio Code**

- uređivač izvornog koda za razne programske jezike
- korišten za kodiranje frontend dijela projekta i projektne dokumentacije
- <https://code.visualstudio.com/>

- **Astah UML**

- program za uređivanje svih vrsta UML dijagrama
- svi dijagrami u ovom dokumentu izrađeni su u Astahu
- <https://astah.net/products/astah-uml/>

- **GitHub**

- stranica za funkcionalno zajedničko stvaranje i uređivanje grupnog projekta
- <https://github.com/>

- **Latex**

- programski jezik za pisanje strukturiranih tekstova, često korišten za stvaranje profesionalnih dokumenata
- koristi markup jezik koji prevodi u formatirane dokumente
- korišten za dokumentiranje projekta
- <https://www.latex-project.org/>

- **Render**

- cloud platforma za izradu i besplatno hostanje web-aplikacija i stranica
- korišten za puštanje *backenda* i *frontenda* u pogon i hostanje baze podataka
- <https://www.render.com/>

- **Render**

- cloud platforma za izradu i besplatno hostanje web-aplikacija i stranica
- korišten za puštanje *backenda* i *frontenda* u pogon i hostanje baze podataka
- <https://www.render.com/>

- **Selenium**

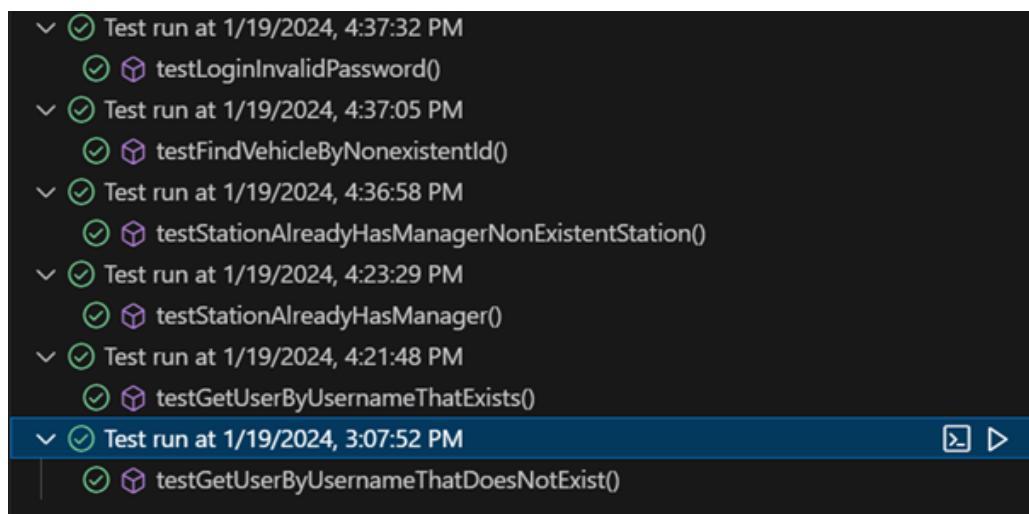
- program za automatizaciju web aplikacija u svrhe testiranja
- korišten za ispitivanje
- <https://www.selenium.dev>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Napravljen je java file za provođenje testova BackendApplicationTests.java sa JUnit testovima koji provjeravaju:

- vraća li grešku funkcija za dohvaćanje korisnika s nepostojećim korisničkim imenom - test GetUserByUsernameThatDoesNotExist
- vraća li funkcija pravog korisnika za dohvaćanje korisnika s postojećim korisničkim imenom - test GetUserByUsernameThatExists
- hoće li funkcija baciti grešku ako je uneseno ime nepostojeće postaje – test StationAlreadyHasManagerNonExistentStation
- hoće li funkcija za nepostojeći ID vozila baciti gresku – test FindVehicleByNonexistentId
- hoće li funkcija za dobro ime, ali lošu lozinku vratiti False – test LoginInvalidPassword
- hoće li funkcija vratiti True ako stanica već ima dodijeljenog voditelja - test StationAlreadyHasManager



Slika 5.1: Zadovoljeni testovi

```

60  @Test
61  @Transactional
62  //Testiranje vraća li grešku funkcija za dohvaćanje korisnika s nepostojećim korisničkim imenom
63  void testGetUserByUsernameThatDoesNotExist() {
64      String username = "testUser";
65      assertThrows(IllegalStateException.class, () -> userService.getUserByUsername(username));
66  }
67
68  @Test
69  @Transactional
70  //Testiranje vraća li funkcija pravog korisnika za dohvaćanje korisnika s postojećim korisničkim imenom
71  void testGetUserByUsernameThatExists() {
72      String username = "testUser";
73      Users expectedUser = new Users(username, "email", "password", "name", "surname"); // Adjust based on your Users entity
74
75      when(userRepository.findById(username)).thenReturn(Optional.of(expectedUser));
76
77      Users actualUser = userService.getUserByUsername(username);
78      assertEquals(expectedUser, actualUser);
79  }
80
81  @Test
82  @Transactional
83  //Test hrće li funkcija vratiti True ako stanica već ima dodijeljenog voditelja
84  void testStationAlreadyHasManager() {
85      String stationName = "TestStation";
86      Long stationId = 1L;
87      String username = "testManager";
88
89      List<Station> stations = Arrays.asList(new Station(stationId, stationName));
90      when(stationRepository.findAll()).thenReturn(stations);
91
92      List<BelongsToStation> belongsToStationList = Arrays.asList(new BelongsToStation(username, stationId));
93      when(belongsToStationRepository.findAll()).thenReturn(belongsToStationList);
94
95      Users testUser = mock(Users.class);
96      when(userRepository.getReferenceById(username)).thenReturn(testUser);
97
98      Roles testRole = mock(Roles.class);
99      when(testUser.getRole()).thenReturn(testRole);
100     when(testRole.getRoleName()).thenReturn("Voditelj postaje");
101
102     assertTrue(adminService.stationHasManager(stationName));
103 }
104
105 @Test
106 @Transactional
107 //Test hrće li funkcija baciti grešku ako je uneseno ime nepostojeće postaje
108 void testStationAlreadyHasManagerNonExistentStation() {
109     String stationName = "NonExistentTestStation";
110     assertThrows(IllegalStateException.class, () -> adminService.stationHasManager(stationName));
111 }
112
113 @Test
114 @Transactional
115 //Test hrće li funkcija za nepostojeći ID vozila baciti grešku
116 void testFindVehicleByNonexistentId() {
117     Long nonExistentVehicleId = 10L;
118     assertThrows(IllegalStateException.class, () -> vehicleService.findVehicleById(nonExistentVehicleId));
119 }
120
121 @Test
122 @Transactional
123 //Test hrće li funkcija za dobro ime, ali lošu lozinku vratiti False
124 public void testLoginInvalidPassword() {
125     String testUsername = "testUser";
126     String testCorrectPassword = "testCorrectPassword";
127     String testWrongPassword = "testWrongPassword";
128     Users testUser = new Users(testUsername, "testUser@gmail.com", testCorrectPassword, "Test", "User");
129
130     when(userRepository.existsById(testUsername)).thenReturn(true);
131     when(userRepository.getReferenceById(testUsername)).thenReturn(testUser);
132
133     assertFalse(userService.login(testUsername, testWrongPassword));
134 }
135 }

```

Slika 5.2: Kod BackendApplicationTests.java

## 5.2.2 Ispitivanje sustava

**Testovi sustava** Provedeni su testovi administratorove mogućnosti uređivanja podataka korisnika, mogućnost voditelja postaje da na svoju postaju doda tragače koji nisu dodijeljeni nijednoj drugoj postaji, mogućnost istraživača da prouči informacije pojedinim jedinkama životinja te mogućnost da mu se na karti prikazuju mesta gdje se pojedine vrste životinja kreću.

### Mijenjanje podataka o korisnicima

Za test mijenjanja podataka o korisnicima koriste se ulazni podaci „admin“ za korisničko ime te „password“ za lozinku.

Pritiskom na gumb „Prijava“ prikazuje nam se početna stranica za administra-

tora. Odaberemo opciju „Pregled korisnika“ te na popisu korisnika odaberemo korisnika s korisničkim imenom „tragac10“. Prikazuje nam se stranica sa svim podacima o navedenom korisniku te opcije „Izbriši korisnika“ i „Promjeni podatke“. Odaberemo opciju „Promjeni podatke“ te nakon što nam se otvori forma za mijenjanje podataka, odaberemo polje ispod teksta „Email:“. Kao novu adresu unešemo tragac15@gmail.hr te pritisnemo na tipku „Spremi promjene“. Nakon toga ponovno nam se otvara stranica s opcijama „Pregled korisnika“ i „Pregled zahjeva“. Ponovno odaberemo opciju „Pregled korisnika“ te pronađemo korisnika s korisničkim imenom „tragač10“. Nakon što pritisnemo na tog korisnika, otvara nam se stranica s njegovim podacima gdje možemo vidjeti njegovu izmijenjenu email adresu.

Prijava se:

Korisničko ime:  
admin

Lozinka:  
\*\*\*\*\*

Prijava

Niste se registrirali? [Registrirajte se](#)

Slika 5.3: Mijenjanje podataka o korisnicima korak 1

### Uredi podatke - korisnik tragac10

**Ime:**  
Marko

**Prezime:**  
Marulić

**Email:**  
tragac10@gmail.com

**Lozinka:**  
••••••••

**Uloga:**  
Tragač ▾

**Spremi promjene** **Odustani**

Slika 5.4: Mijenjanje podataka o korisnicima korak 2

### Uredi podatke - korisnik tragac10

**Ime:**  
Marko

**Prezime:**  
Marulić

**Email:**  
tragac15@gmail.hr

**Lozinka:**  
••••••••

**Uloga:**  
Tragač ▾

**Spremi promjene** **Odustani**

Slika 5.5: Mijenjanje podataka o korisnicima korak 3



Slika 5.6: Mijenjanje podataka o korisnicima korak 4

#### *Dodavanje tragača na postaju*

Za test dodavanja tragača na postaju u ulozi voditelja postaje koriste se ulazni podaci „waiting“ za korisničko ime te „password“ za lozinku. Pritisom na gumb „Prijava“ prikazuje nam se početna stranica za korisnike u ulozi voditelja postaje. Pojavljuje nam se tekst s postajom kojom upravlja prijavljeni voditelj te pet opcija za odabir. Nudi se opcije „Pregled svih tragača“, „Pregled svojih tragača“, „Pregled akcija“, „Pregled zahtjeva“ te „Moj profil“ (u gornjem desnom kutu). Odaberemo opciju „Pregled svih tragača“, nakon čega nam se otvara prozor s tekstrom „Pregled svih tragača“ te ispod njega popis svih tragača koji trenutno nemaju dodijeljenu postaju. Pritisom na kućicu ispod pripadajućeg tragača, nudi nam se šest različitih vozila za dodjeljivanje kao sposobnosti odabranom tragaču. Odaberemo tragača s imenom „test“ te prezimenom „tragac“. Nakon toga mu kao prijevozna sredstva dodijelimo mogućnosti „pješke“ te „dronom“. Pritisom na gumb „Potvrdi“ potvrđujemo svoj odabir. Nakon što smo potvrdili svoj odabir, ponovno se otvara početna stranica korisnika u ulozi voditelja postaje. Ovaj put odaberemo opciju „Pregled svojih tragača“, nakon čega nam se otvara prozor sa tekstrom „Lista svojih tragača“ te ispod njega popis svih tragača dodijeljenih postaji, s njihovim osnovnim korisničkim podacima. Možemo vidjeti da se sada i na popisu nalazi tragač s imenom „test“ i prezimenom „tragac“.



Slika 5.7: Dodavanje tragača na postaju korak 1

Pregled svih tragača

**Ime:** test, **Prezime:** tragac  
✓

Izaberi prijevozno sredstvo:

pješke  
✓

dronom  
☐

automobilom  
☐

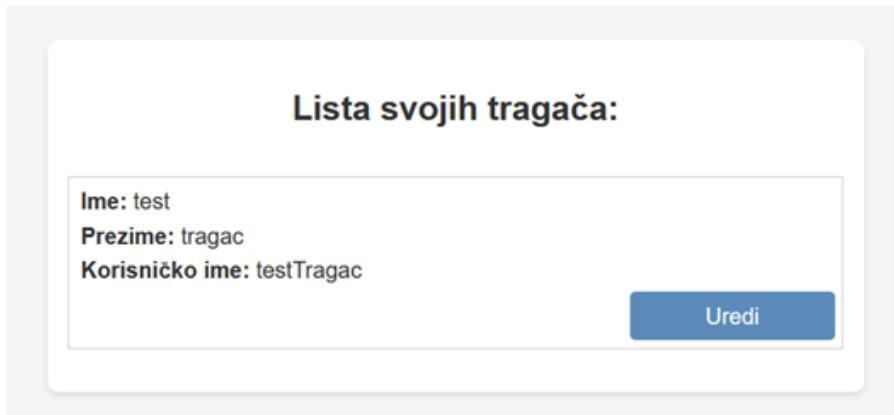
cross motorom  
☐

brodom  
☐

helikopterom  
☐

Potvrdi

Slika 5.8: Dodavanje tragača na postaju korak 2



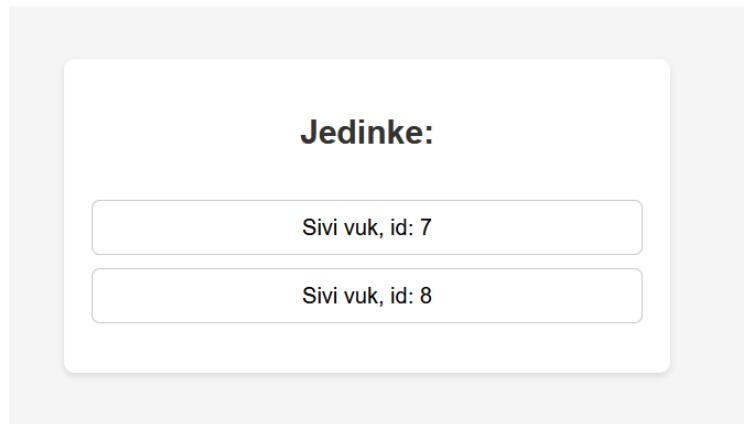
Slika 5.9: Dodavanje tragača na postaju korak 3

### Pregled informacija o životinjama

Za test pregleda informacija o životinjama u ulozi istraživača koriste se ulazni podaci „istrazivacXmouse“ za korisničko ime te „lozinka1“ za lozinku. Pritisom na gumb „Prijava“ prikazuje nam se početna stranica za korisnike u ulozi istraživača. Za odabir nam se nude opcije „Popis akcija“, „Karta – životinje“, „O životinjama“ te „Moj profil“ (u gornjem desnom kutu). Odaberemo opciju „O životinjama“. Otvara nam se izbornik s različitim vrstama životinja. Odaberemo opciju „Sivi vuk“ te nakon što nam se prikaže prozor s popisom jedinki te vrste, odaberemo opciju „Sivi vuk, id: 7“. Otvara nam se prozor u kojem možemo vidjeti sliku pripadajuće vrste, njen latinski naziv, detaljan opis vrste te komentare ostalih korisnika vezanih uz navedenu jedinku. Dodatno, nudi nam se mogućnost brisanja te pisanja vlastitih komentara.



Slika 5.10: Pregled informacija o životinjama korak 1



Slika 5.11: Pregled informacija o životinjama korak 2

A detailed view of a wolf standing in a forest, looking towards the left. The wolf has a thick coat of grey, brown, and black fur.

**Sivi vuk id: 7**

Latinski naziv: Canis lupus

Opis: Zadnji put viđen blizu potoka

**Komentari:**

tragac7: uhvatio pticu	x
ju54164: ima mladunce 14.1.	x
faranjo: novi komentar	x

Napišite komentar ovdje...

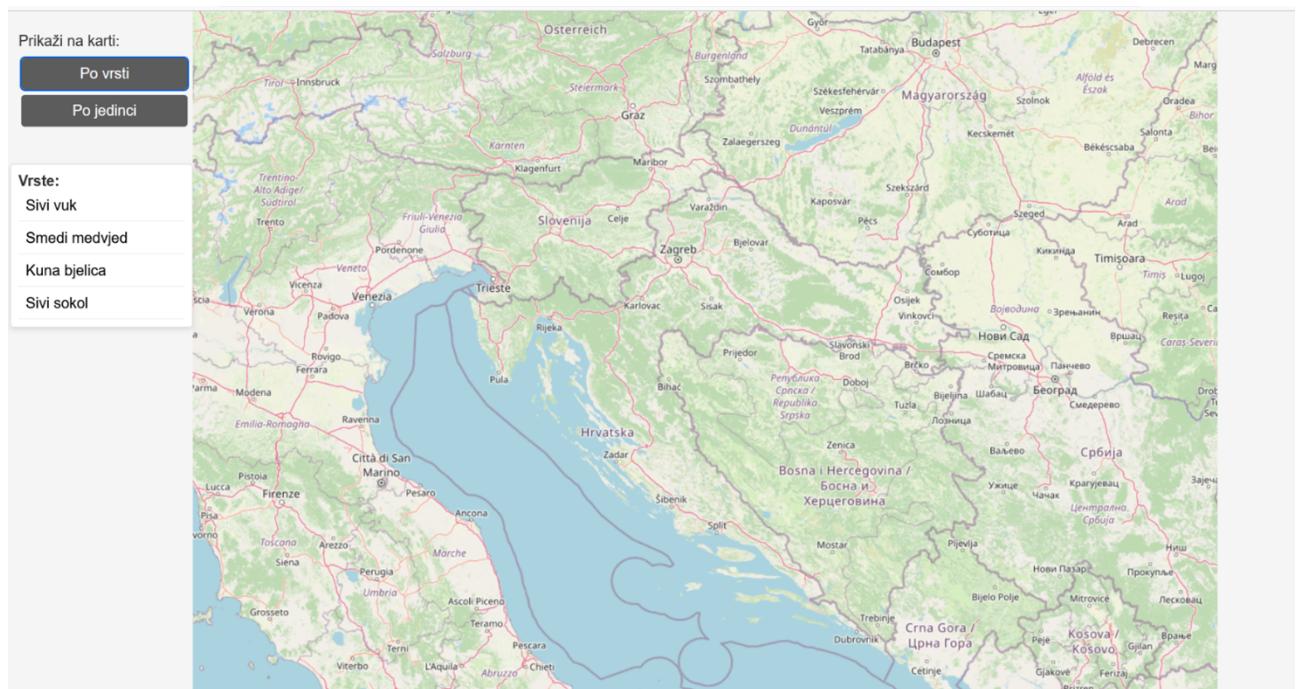
Slika 5.12: Pregled informacija o životinjama korak 3

### **Pregled staništa vrste na karti**

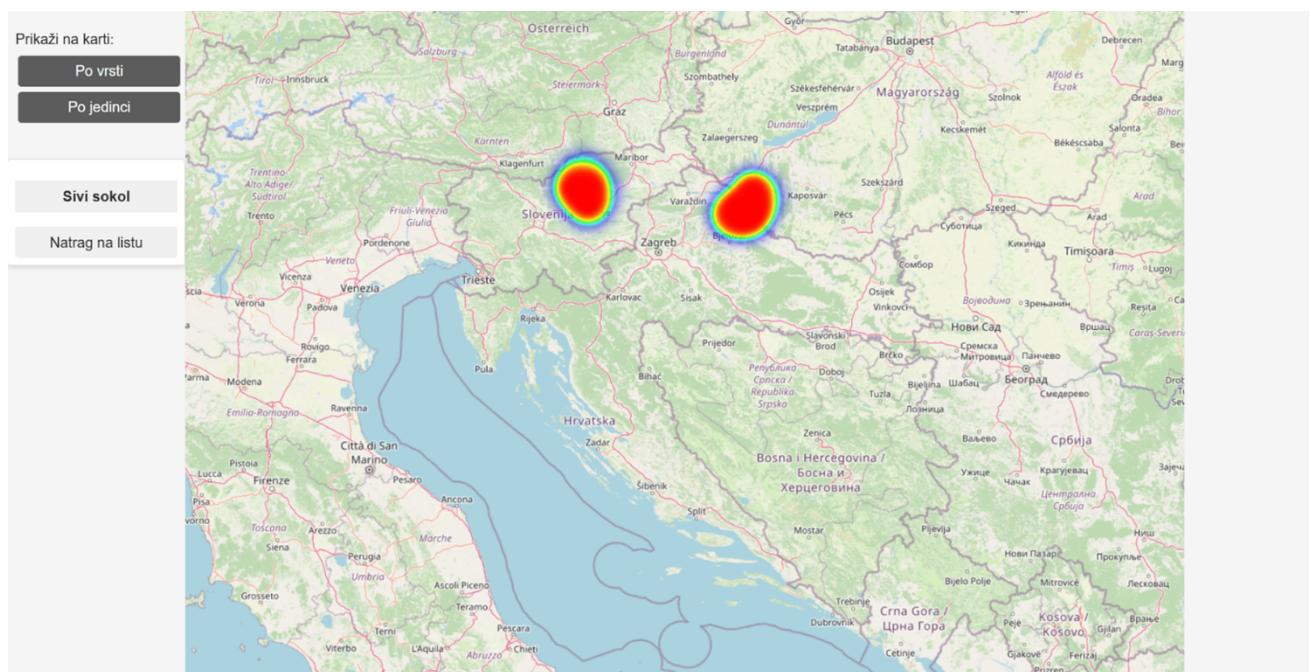
Za test pregleda staništa pojedine vrste životinja na karti u ulozi istraživača koriste se ulazni podaci „ju54164“ za korisničko ime te „lozinka1“ za lozinku. Pritisom na gumb „Prijava“ prikazuje nam se početna stranica za korisnike u ulozi istraživača. Za odabir nam se nude opcije „Popis akcija“, „Karta – životinje“, „O životinjama“ te „Moj profil“ (u gornjem desnom kutu). Odaberemo opciju „Karta – životinje“. Otvara nam se prozor u čijem se središtu nalazi karta, a u gornjem lijevom kutu dvije opcije za prikaz na karti, „Po vrsti“ te „Po jedinci“. Odaberemo opciju „Po vrsti“, nakon čega nam se ispod pojavljuje popis svih mogućih vrsta. Pritisom na gumb „Sivi sokol“, na karti nam se pojavljuje heatmap svih povijesnih lokacija jedinki koje pripadaju toj vrsti.



Slika 5.13: Pregled staništa vrste na karti korak 1



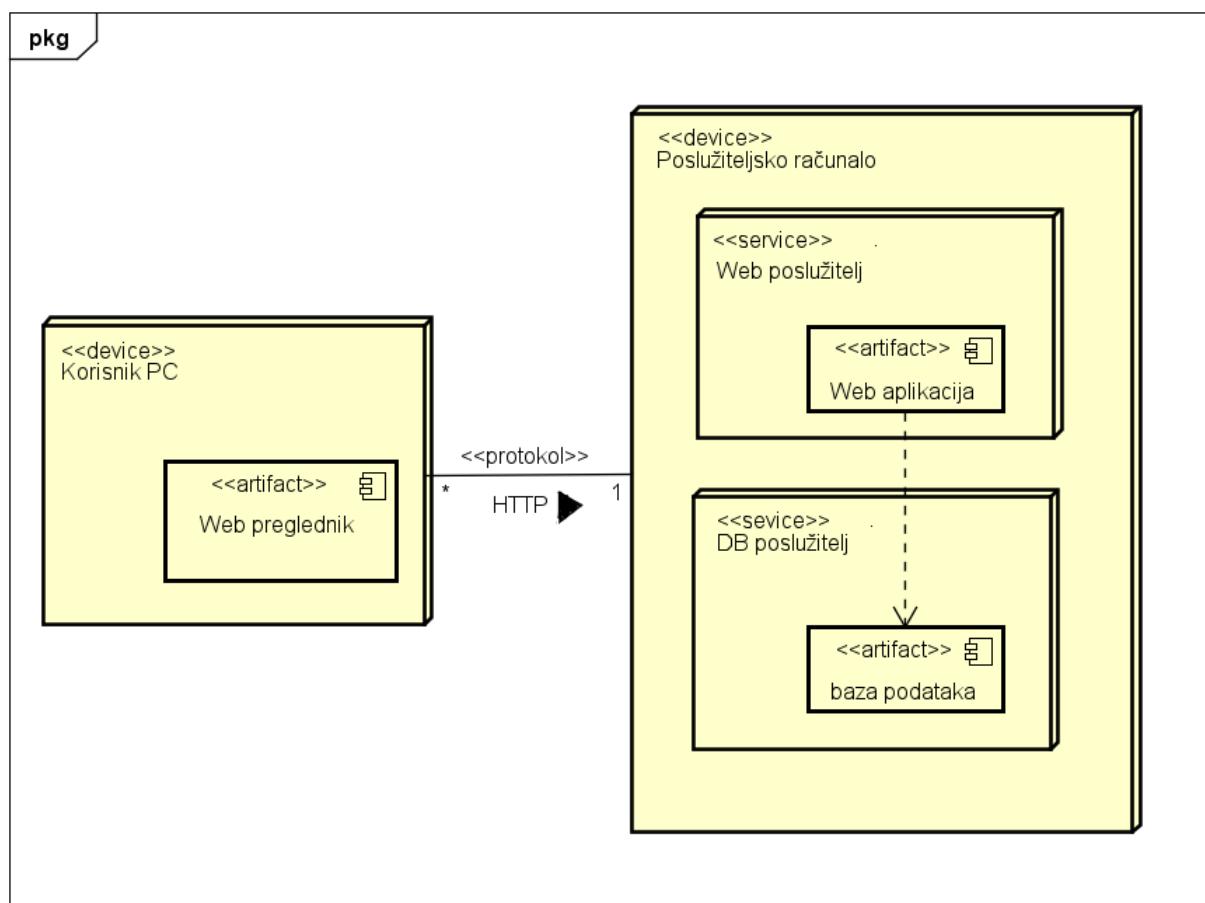
Slika 5.14: Pregled staništa vrste na karti korak 2



Slika 5.15: Pregled staništa vrste na karti korak 3

### 5.3 Dijagram razmještaja

UML-dijagram razmještaja je vrsta struktturnog UML-dijagrama koji prikazuje fizičku arhitekturu i konfiguraciju razmještaja programskog sustava. Dijagram razmještaja koristan je za vizualizaciju kako su komponente sustava raspoređene na fizičkim računalima i kako međusobno komuniciraju. Na poslužiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka. Korisnici preko svog računala koriste web preglednik kako bi pristupili web aplikaciji. Sustav je baziran na arhitekturi "klijent - poslužitelj", a komunikacija između računala korisnika, koji može biti tragač, istraživač, voditelj postaje ili administrator, i poslužitelja se odvija preko HTTP veze, što je tipičan način komunikacije u web okolinama.



Slika 5.16: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

Puštanje web aplikacije u pogon sastoji se od tri segmenta:

- Stvaranje baze podataka
- Puštanje *backend-a* u pogon
- Puštanje *frontend-a* u pogon

### Stvaranje baze podataka

Baza podataka besplatno je spremljena na web-oblaku render.com. Ona se postavlja i osposobljava sljedećim koracima:

#### 1. Stvaranje nove baze

- Potrebno je (nakon registracije na render.com) odabratи opciju New te na padajućem izborniku PostgreSQL
- Pojavljuje se izbornik koji ispunjavamo kao na slici 5.17
- Biramo *Create Database*

## New PostgreSQL

[Read the docs](#)

Name  
A unique name for your PostgreSQL instance.

Database Optional  
The PostgreSQL `dbname`

User Optional

Region  
The [region](#) where your PostgreSQL instance runs. Services must be in the same region to communicate privately and you currently have services running in Frankfurt.

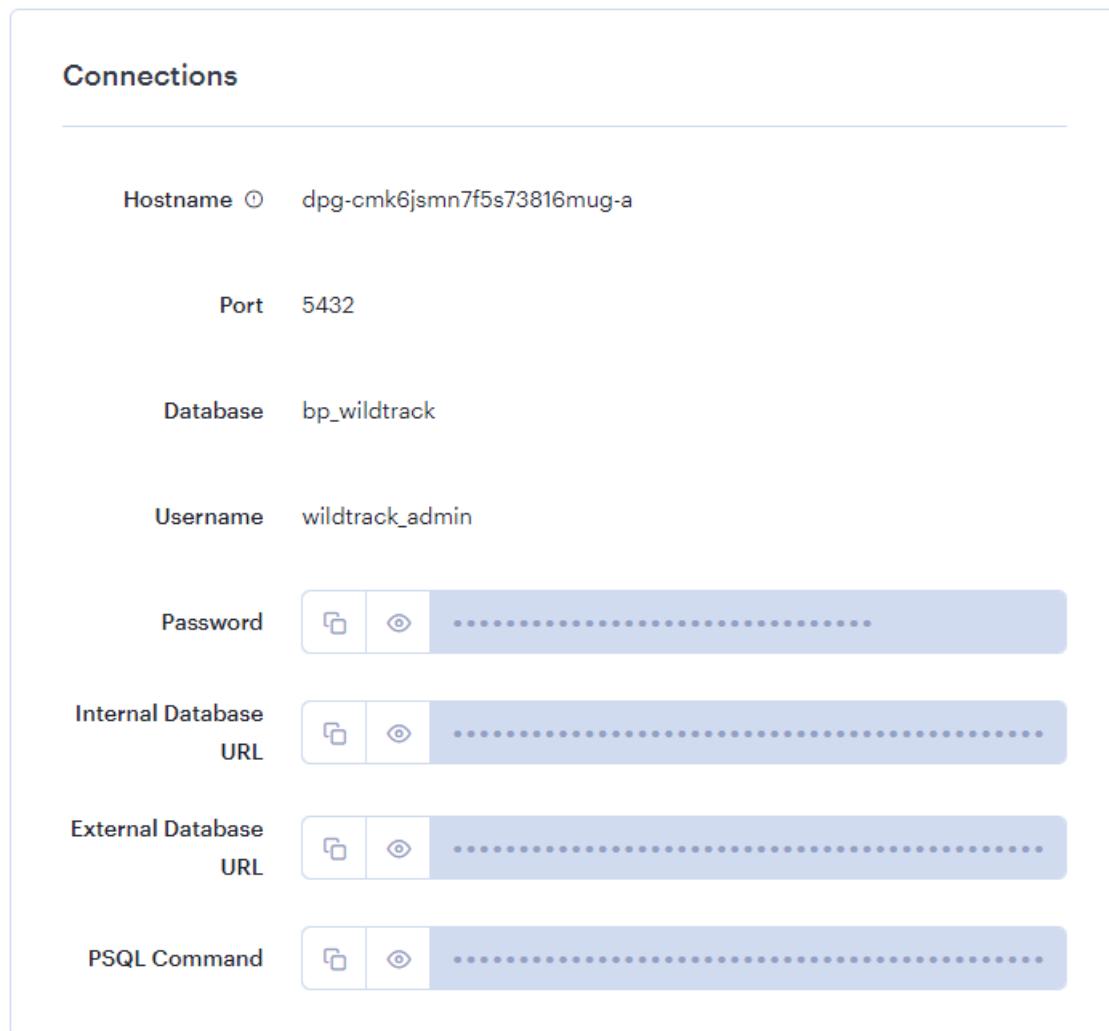
PostgreSQL Version

Datadog API Key Optional  
The API key to use for sending metrics to Datadog. Setting this will enable Datadog monitoring.

Slika 5.17: Stvaranje nove PostgreSQL baze

### 2. Dohvat podataka za spajanje

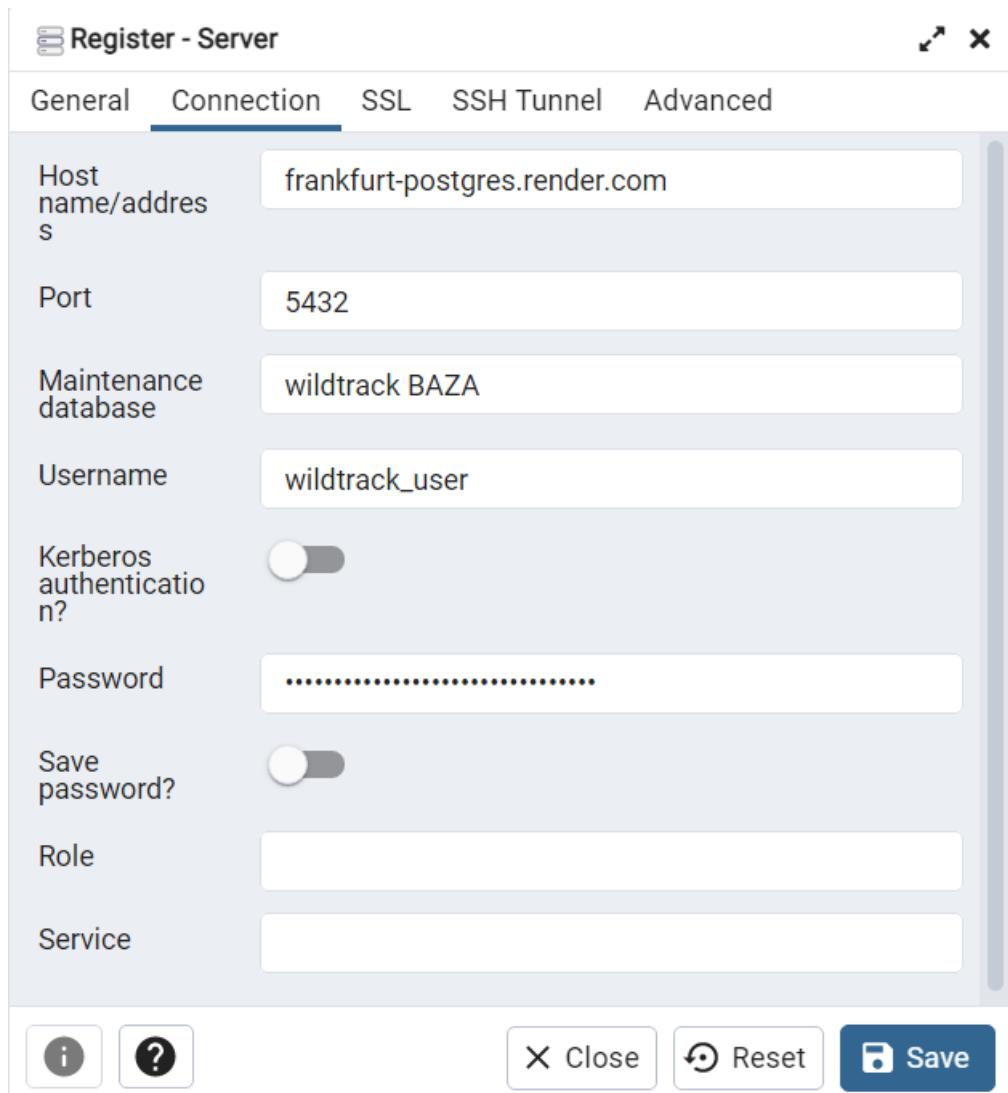
- Odabiremo *Dashboard - wildtrack BAZA - Info*
- Spuštamo stranicu do izbora *Connections* sa informacijama potrebnim za spajanje na bazu (slika 5.18).



Slika 5.18: Dohvat podataka za spajanje baze

### 3. Spajanje putem pgAdmina

- Odabiremo *Object - Create - Server*
- Na kartici *General* unosimo ime servera kojeg otvaramo u pgAdminu
- Na kartici *Connection* unosimo podatke kao na slici 5.19 i spremamo ih sa *Save*



Slika 5.19: Unos podataka za spajanje baze

#### 4. Spajanje iz *backenda*

- U *src/main/resources/application.properties* upisujemo naredbe sa slike 5.20

```

2
3  spring.datasource.url=jdbc:postgresql://dpg-clae49pm6hds73ehlf0-a.frankfurt-postgres.render.com:5432/wildtrack
4  spring.datasource.username=wildtrack_user
5  spring.datasource.password=u4RfMxGnEd04sgksWpMkK4QgMWajyCEh
6  spring.datasource.driver-class-name=org.postgresql.Driver

```

Slika 5.20: Spajanje baze iz *backenda*

## Puštanje backenda u pogon

Backend dio projekta također koristi render.com za puštanje u pogon. Preduvjeti za to su:

- Dodavanje Dockerfile-a prikazanog na slici 5.21
- povezivanje s GitHub-om na renderu

The screenshot shows a code editor window with a dark theme. The file is named 'Dockerfile'. The code is written in Dockerfile syntax and defines a build process for a Java application. It starts by setting the base image to 'openjdk:17-alpine' and creating a builder stage. It then copies the project's source code and build scripts ('mvn', 'mvnw', 'pom.xml', 'src') into the container. It runs 'chmod +x mvnw' to make the build script executable. The next section uses 'RUN ./mvnw clean package' to build the application. It then sets up a production stage using 'openjdk:17-alpine' and copies the built JAR file ('target/\*.jar') into the container. The port '8080' is exposed. Finally, it specifies the entry point as 'java -jar /app.jar'. The Dockerfile is numbered from 1 to 27.

```
1 # Container za izgradnju (build) aplikacije
2 FROM openjdk:17-alpine AS builder
3
4 # Kopiranje izvornog koda u container
5 COPY ../../mvn .mvn
6 COPY ../../mvnw .
7 COPY ../../pom.xml .
8 COPY ../../src src
9 RUN chmod +x mvnw
10
11 # Pokretanje builda
12 RUN ./mvnw clean package
13
14 # Stvaranje containera u kojem ce se vrtiti aplikacija
15 FROM openjdk:17-alpine
16
17 ## Ovdje je moguce instalirati alate potrebne za rad aplikacije. Vjerojatno vam nece trebati, no dobro je znati.
18 ## Linux distro koji se koristi je Alpine, stoga se kao package manager koristi apk
19 #RUN apk install <nesto>
20
21 # Kopiranje izvrsnog JAR-a iz build containera u izvrsni container
22 COPY --from=builder target/*.jar /app.jar
23
24 # Izlaganje porta
25 EXPOSE 8080
26
27 # Naredba kojom se pokreće aplikacija
28 ENTRYPOINT ["java", "-jar", "/app.jar"]
```

Slika 5.21: Potrebni Dockerfile

Stvaranje backenda postižemo na sljedeći način:

- Odabiremo na renderu *New - Web Service*
- Odabiremo željeni GitHub repozitorij
- Unosimo tražene podatke i potvrđujemo s *Create Web Service*

## Puštanje frontenda u pogon

Frontend dio projekta također koristi render.com za puštanje u pogon. Preduvjeti za to su:

- U IzvorniKod/package.json dodajemo dependency-e (primarno http-proxy-middleware, dotenv, express) kao na slici 5.22

```
6  "dependencies": {  
7    "@mapbox/polyline": "^1.2.1",  
8    "@testing-library/jest-dom": "^5.16.5",  
9    "@testing-library/react": "^13.4.0",  
10   "@testing-library/user-event": "^13.5.0",  
11   "@types/jest": "^27.5.2",  
12   "@types/node": "^16.11.65",  
13   "@types/react": "^18.0.21",  
14   "@types/react-dom": "^18.0.6",  
15   "axios": "^1.6.5",  
16   "dotenv": "^16.0.3",  
17   "express": "^4.18.2",  
18   "http-proxy-middleware": "^2.0.6",  
19   "leaflet": "^1.9.4",  
20   "leaflet.heat": "^0.2.0",  
21   "mapbox-gl": "^3.0.1",  
22   "react": "^18.2.0",  
23   "react-dom": "^18.2.0",  
24   "react-helmet": "^6.1.0",  
25   "react-leaflet": "^4.2.1",  
26   "react-map-gl": "^7.1.7",  
27   "react-router-dom": "^6.19.0",  
28   "react-scripts": "5.0.1",  
29   "typescript": "^4.8.4",  
30   "web-vitals": "^2.1.4"  
31 },  
  
```

Slika 5.22: Dio koda package.json *dependency*

- U IzvorniKod/package.json dodajemo kod kao na slici 5.23 zbog funkcionalnosti Node-a

```
32     "scripts": {  
33       "start": "react-scripts start",  
34       "build": "yarn install && react-scripts build",  
35       "start-prod": "node app.js",  
36       "test": "react-scripts test",  
37       "eject": "react-scripts eject"  
38     },  
39     "engines": {  
40       "node": ">=18.18.0 <19.0.0"  
41     },
```

Slika 5.23: Dio koda package.json za funkcionalnost *Node-a*

- Dodajemo /src/setupProxy.js koji služi kao proxy server za lokalni development (preusmjerava api pozive na localhost:8080)
- Dodajemo app.js, u kojem se nalazi express server za produkcijski proxy i posluživanje *frontenda* (slika 5.24)

```
1 const express = require("express");
2 const { createProxyMiddleware } = require("http-proxy-middleware");
3 require("dotenv").config();
4 const path = require("path")
5
6 const app = express();
7
8 // Configuration
9 const { PORT } = process.env;
10 const { HOST } = process.env;
11 const { API_BASE_URL } = process.env;
12
13 // Proxy
14 app.use(
15   "/",
16   createProxyMiddleware({
17     target: API_BASE_URL,
18     changeOrigin: true,
19   })
20 );
21
22 app.use(express.static(path.join(__dirname, 'build')))
23
24 app.listen(PORT, HOST, () => {
25   console.log(`Starting Proxy at ${HOST}:${PORT}`);
26 });
27
28 app.get("*", async (req, res) => {
29   res.sendFile(path.join(__dirname, 'build', 'index.html'))
30 }
31 );
```

Slika 5.24: Kod datoteke app.js

Stvaranje frontenda postižemo na sljedeći način:

- Odabiremo na renderu *New - Web Service*
- Odabiremo željeni GitHub repozitorij
- Unosimo tražene podatke (slika 5.25)
- Potvrđujemo s *Create Web Service*

Name  
A unique name for your web service.

WildTrack

Region  
The [region](#) where your web service runs. Services must be in the same region to communicate privately and you currently have services running in Frankfurt.

Frankfurt (EU Central)

Branch  
The repository branch used for your web service.

main

Root Directory Optional  
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.

<https://github.com/progi-devops/progi-monorepo/tree/main/progi-fe>

Runtime  
The runtime for your web service.

Docker

Slika 5.25: Spajanje frontenda

## 6. Zaključak i budući rad

Zadatak naše grupe Aristos bio je razvoj web aplikacije koja olakšava koordinaciju prilikom praćenja divljih životinja. Rad na projektu sadržavao je mnoge očekivane i neočekivane izazove i zapreke koje smo rješavali kroz marljivu suradnju.

Nakon dodjele projektnog zadatka, оформili smo podjedinice za razvoj *frontend-a*, razvoj *backend-a* i pisanje dokumentacije te počeli s raspodjelom rada na inicijalnim koracima projekta. Organizacija vremena i rada te podjela poslova bili su od ključne važnosti u napretku projekta, baš kao i međusobna suradnja. Kao zajednički oslonac koristili smo rukom crtanu skicu koja prikazuje idejni izgled početne stranice i njezino grananje na razne funkcionalnosti za različite uloge korisnika. Članovi su se, zbog nedostatka iskustva u razvoju implementacijskih rješenja, morali angažirati u samostalnom učenju odabralih alata i programskih jezika kako bi postigli postavljene ciljeve i rokove.

Naša *WildTrack* web aplikacija ima još prostora za potencijalna buduća unapređenja koja smo mogli ostvariti uz malo više vremena i vještine, poglavito stvaranje mobilne aplikacije da se približi suvremenim korisnicima.

Unatoč prostoru za usavršavanje na određenim aspektima aplikacije, koji je posljedica neiskustva, kao tim smo zadovoljni postignutim konačnim rezultatom, vremenom uloženim u projekt te znanjem koje smo stekli radeći ga. Zajedničko stvaranje web-aplikacije koristilo je kao vrijedno novo iskustvo svakom od članova i približilo nas

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. W3Schools, <https://www.w3schools.com/>
3. Latex Forum, <https://latex.org/forum/index.php>
4. Spring Boot, <https://spring.io/projects/spring-boot>
5. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
6. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
7. I. Marsic, Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
8. The Unified Modeling Language, <https://www.uml-diagrams.org/>
9. Astah Community, <http://astah.net/editions/uml-new>
10. <https://play.google.com/store/apps/details?id=com.mpio.movebank&hl=en&gl=US>
11. <https://play.google.com/store/apps/details?id=com.tips.dikihzhivotnyh&hl=hr>

# Indeks slika i dijagrama

2.1 Aplikacija Divlje Životinje . . . . .	9
2.2 Aplikacija Animal Tracker . . . . .	10
3.1 Funkcionalnost neregistriranog korisnika, generaliziranog korisnika, administratora aplikacije i voditelja postaje . . . . .	25
3.2 Funkcionalnost tragača i istraživača . . . . .	26
3.3 Sekvencijski dijagram za UC1 . . . . .	27
3.4 Sekvencijski dijagram za UC28 . . . . .	28
4.1 Arhitektura sustava . . . . .	30
4.2 Dijagram baze podataka . . . . .	41
4.3 Dijagram razreda - dio Controllers . . . . .	43
4.4 Dijagram razreda - dio Data transfer objects . . . . .	44
4.5 Dijagram razreda - dio Models . . . . .	45
4.6 Dijagram stanja - tragač . . . . .	46
4.7 Dijagram aktivnosti - stvaranje nove akcije . . . . .	48
4.8 Dijagram komponenti . . . . .	50
5.1 Zadovoljeni testovi . . . . .	53
5.2 Kod BackendApplicationTests.java . . . . .	54
5.3 Mijenjanje podataka o korisnicima korak 1 . . . . .	55
5.4 Mijenjanje podataka o korisnicima korak 2 . . . . .	56
5.5 Mijenjanje podataka o korisnicima korak 3 . . . . .	56
5.6 Mijenjanje podataka o korisnicima korak 4 . . . . .	57
5.7 Dodavanje tragača na postaju korak 1 . . . . .	58
5.8 Dodavanje tragača na postaju korak 2 . . . . .	58
5.9 Dodavanje tragača na postaju korak 3 . . . . .	59
5.10 Pregled informacija o životinjama korak 1 . . . . .	59
5.11 Pregled informacija o životinjama korak 2 . . . . .	60
5.12 Pregled informacija o životinjama korak 3 . . . . .	60
5.13 Pregled staništa vrste na karti korak 1 . . . . .	61

5.14 Pregled staništa vrste na karti korak 2 . . . . .	62
5.15 Pregled staništa vrste na karti korak 3 . . . . .	62
5.16 Dijagram razmještaja . . . . .	63
5.17 Stvaranje nove PostgreSQL baze . . . . .	65
5.18 Dohvat podataka za spajanje baze . . . . .	66
5.19 Unos podataka za spajanje baze . . . . .	67
5.20 Spajanje baze iz <i>backenda</i> . . . . .	67
5.21 Potrebni Dockerfile . . . . .	68
5.22 Dio koda package.json <i>dependency</i> . . . . .	69
5.23 Dio koda package.json za funkcionalnost <i>Node-a</i> . . . . .	70
5.24 Kod datoteke app.js . . . . .	71
5.25 Spajanje <i>frontenda</i> . . . . .	72
6.1 Dijagrami pregleda promjena . . . . .	82

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 20.listopada 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić
- Teme sastanka:
  - sastanak sa asistentom
  - analiza zadatka

### 2. sastanak

- Datum: 25.listopada 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić
- Teme sastanka:
  - određivanje izgleda aplikacije
  - definiranje funkcionalnih zahtjeva
  - raspodjela poslova

### 3. sastanak

- Datum: 4.studenoga 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić, G. Mihaljević
- Teme sastanka:
  - pregled i popravak dijagrama
  - dogovor u kojem jeziku i programima programiramo
  - raspodjela poslova

### 4. sastanak

- Datum: 10.studenoga 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić

- Teme sastanka:

- sastanak sa demonstratorom - pregled dosadašnjeg rada
  - osmišljavanje stranice i pojedinih pregleda
  - raspodjela poslova

#### 5. sastanak

- Datum: 16.studenoga 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić, G. Mihaljević
- Teme sastanka:
  - provjera svega napravljenoga
  - dogovor tko će koji dio prepraviti
  - dogovor što trebamo dodati

#### 6. sastanak

- Datum: 14.prosinca 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić, G. Mihaljević
- Teme sastanka:
  - dogovor oko izgleda stranice
  - dogovor oko izgleda baze podataka
  - raspodjela poslova

#### 7. sastanak

- Datum: 15.prosinca 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić
- Teme sastanka:
  - dogovor sa asistentom i demonstratorom
  - raspodjela poslova
  - detaljan dogovor oko izgleda stranice voditelja i istraživača

#### 8. sastanak

- Datum: 09.siječnja 2024.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurić
- Teme sastanka:
  - provjera napravljenog
  - dogovor izgleda stranice za tragača

– raspodjela poslova

9. sastanak

- Datum: 15.siječnja 2023.
- Prisustvovali: F. Vuković, M. Pongrac, M. Kukolj, S. Troskot, J. Udovičić, D. Jurič
- Teme sastanka:
  - provjera napravljenog
  - dogovor oko dodavanja posljednjih funkcionalnosti

## Tablica aktivnosti

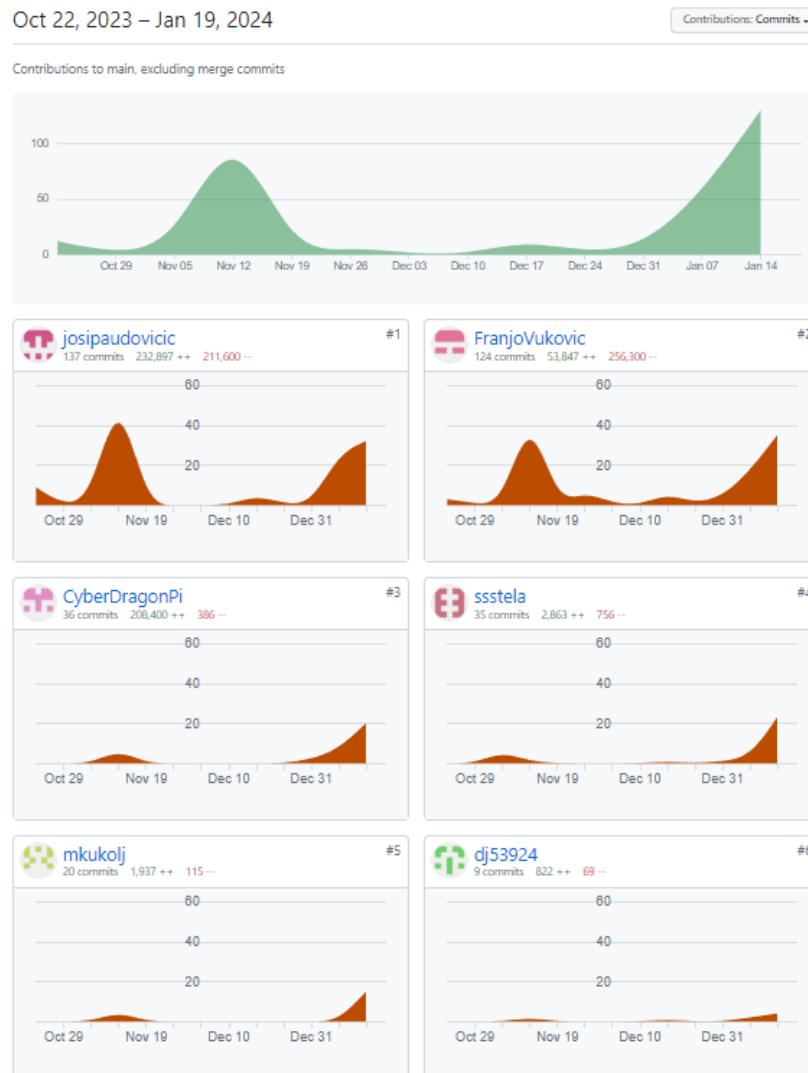
	Josipa Udovičić	Franjo Vuković	Stela Troskot	Marko Pongrac	Marko Kukolj	Domagoj Juric	Gregor Mihaljević
Upravljanje projektom	5	3	1				
Opis projektnog zadatka	4						
Funkcionalni zahtjevi	1						
Opis pojedinih obrazaca			2			2	
Dijagram obrazaca		3	6	3	3		
Sekvencijski dijagrami		2	2	2	2		
Opis sekvencijskih dijagrama					2		
Opis ostalih zahtjeva			13			11	
Arhitektura i dizajn sustava	3						
Baza podataka(opis)	1		1				3
Dijagram razreda		6		6	9		
Dijagram stanja			4				
Dijagram aktivnosti			4				
Dijagram komponenti			5				
Korištene tehnologije i alati			1				
Ispitivanje programskog rješenja				9		3	
Dijagram razmještaja			4				
Upute za puštanje u pogon	2	8	3		3		
Dnevnik sastajanja	1						

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Josipa Udovičić	Franjo Vuković	Stela Troskot	Marko Pongrac	Marko Kukolj	Domagoj Jurič	Gregor Mihaljević
Zaključak i budući rad			2				
Popis literature	1						
izrada početne stranice(registracija)	7						
izrada baze podataka		4		4			
spajanje s bazom podataka		2		2	5		
back end		350		260	180		
front end	300		20			200	

## Dijagrami pregleda promjena



Slika 6.1: Dijagrami pregleda promjena