

Programsko inženjerstvo

Ak. god. 2023./2024.

# Wild Track

Dokumentacija, Rev. 1

Grupa: Lovci Na

Voditelj: Mia Sara Matušin

Datum predaje: 17. studeni 2023.

Nastavnik: Hrvoje Nuić

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>6</b>
<b>3 Specifikacija programske potpore</b>	<b>12</b>
3.1 Funkcionalni zahtjevi . . . . .	12
3.1.1 Obrasci uporabe . . . . .	14
3.1.2 Sekvencijski dijagrami . . . . .	23
3.2 Ostali zahtjevi . . . . .	28
<b>4 Arhitektura i dizajn sustava</b>	<b>29</b>
4.1 Baza podataka . . . . .	30
4.1.1 Opis tablica . . . . .	31
4.1.2 Dijagram baze podataka . . . . .	43
4.2 Dijagram razreda . . . . .	44
4.3 Dijagram stanja . . . . .	49
4.4 Dijagram aktivnosti . . . . .	50
4.5 Dijagram komponenti . . . . .	51
<b>5 Implementacija i korisničko sučelje</b>	<b>53</b>
5.1 Korištene tehnologije i alati . . . . .	53
5.2 Ispitivanje programskog rješenja . . . . .	54
5.2.1 Ispitivanje komponenti . . . . .	54
5.2.2 Ispitivanje sustava . . . . .	57
5.3 Dijagram razmještaja . . . . .	61
5.4 Upute za puštanje u pogon . . . . .	62
<b>6 Zaključak i budući rad</b>	<b>63</b>
<b>Popis literature</b>	<b>64</b>
<b>Indeks slika i dijagonama</b>	<b>65</b>

**Dodatak: Prikaz aktivnosti grupe**

**66**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Mia Sara Matušin	22.10.2023.
0.2	Dodani funkcionalni zahtjevi	Nikola Jamić	23.10.2023.
0.3	Dodan prvi dio obrazaca uporabe	Gabriela Oroz	23.10.2023.
0.4	Dodan prvi dio opisa projektnog zadatka	Mia Sara Matušin	23.10.2023.
0.5	Dodani obrasci uporabe	Lana Barišić	25.10.2023.
0.5.1	Dovršen opis projektnog zadatka	Mia Sara Matušin	25.10.2023.
0.5.2	Ažuriran dnevnik aktivnosti i tablica aktivnosti	Mia Sara Matušin	25.10.2023.
0.5.4	Uređeni obrasci uporabe	Gabriela Oroz	25.10.2023.
0.6	Sekvencijski dijagrami i ostali zahtjevi	Martin Bugarin	26.10.2023.
0.6.1	Uređeni funkcionalni zahtjevi	Nikola Jamić	30.10.2023.
0.7	Ažurirani obrasci uporabe	Lana Barišić	1.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

<b>Rev.</b>	<b>Opis promjene/dodataka</b>	<b>Autori</b>	<b>Datum</b>
0.8	Detaljan opis baze podataka i ER dijagrami	Dino Baralić, Dorjan Štrbac	2.11.2023.
0.8.1	Ažurirani funkcionalni dijagrami i ostali zah-tjevi	Martin Bugarin	2.11.2023.
0.8.2	Dovršeni obrasci uporabe	Lana Barišić	3.11.2023.
0.9	Dodan opis arhitekture	Lana Barišić	8.11.2023.
0.9.1	Dodani dijagrami razreda	Dorjan Štrbac	15.11.2023.
<b>1.0</b>	<b>Verzija za prvi ciklus</b>	Mia Sara Matušin	16.11.2023.
1.1	Dodan dijagram komponenti	Dino Baralić	15.1.2024.
1.2	Dodan dijagram stanja	Martin Bugarin	15.1.2024.
1.3	Dodan dijagram aktivnosti	Martin Bugarin	15.1.2024.
1.4	Ispravljanje obrazaca uporabe te dijagrama obrazaca	Lana Barišić	17.1.2024.
1.5	Skaliranje slika	Lana Barišić	17.1.2024.
1.6	Ispravljanje opisa baze podataka	Nikola Jamić	19.1.2024.
1.7	Dodan dijagram razmještaja i korištene tehnologije i alati	Martin Bugarin	19.1.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.8	Dodan zaključak i budući rad	Mia Sara Matušin	19.1.2024.
1.9	Dodane upute za deploy aplikacije	Dino Baralić	19.1.2024.
2.0	Završna verzija	Mia Sara Matušin	19.1.2024.

## 2. Opis projektnog zadatka

Cilj je ovog zadatka napraviti aplikacijsko programsko sučelje koje će olakšati komunikaciju između istraživača, voditelja postaje te tragača u akcijama praćenja divljih životinja.

Prvi korak za korištenje aplikacije je registracija. Prilikom registracije korisnik odabire koju ulogu će imati u aplikaciji. On može biti istraživač, voditelj postaje ili tragač. Podaci koje je potrebno unijeti pri registraciji su sljedeći:

- *korisničko ime*
- *fotografija*
- *lozinka*
- *ime*
- *prezime*
- *email adresa.*

Registracija završava potvrdom preko emaila, te dodatno voditelja i istraživača treba potvrditi administrator. Sličan primjer obrasca za registraciju prikazan je u nastavku na slici 2.1.

The registration form consists of several input fields and sections. At the top are three fields: 'Email adresa' (Email address), 'Lozinka' (Password), and 'Potvrda lozinke' (Confirm password). Below these is a section titled 'OSOBNI PODACI' (Personal Data) containing two fields: 'Ime' (Name) and 'Prezime' (Last name). At the bottom are two checkboxes: one for accepting a newsletter and another for terms and conditions. A red 'POTVRDI' (Confirm) button is at the very bottom.

Email adresa	(Email address input field)
Lozinka	(Password input field)
Potvrda lozinke	(Confirm password input field)
<b>OSOBNI PODACI</b>	
Ime	(Name input field)
Prezime	(Last name input field)
<input type="checkbox"/> Želim primati newsletter	
<input type="checkbox"/> Registracijom prihvacam <b>opće uvjete korištenja.</b>	
<b>POTVRDI</b>	

Slika 2.1: Primjer obrasca za registraciju

**Voditelj** vodi svoju postaju koja ima naziv po području na kojem se nalazi (npr. Kopački rit, Velebit...). Također posjeduje popis **tragača** koji su dio njegove postaje te na koji način mogu obavljati istraživanje. Mogućnosti kretanja tragača su sljedeće:

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

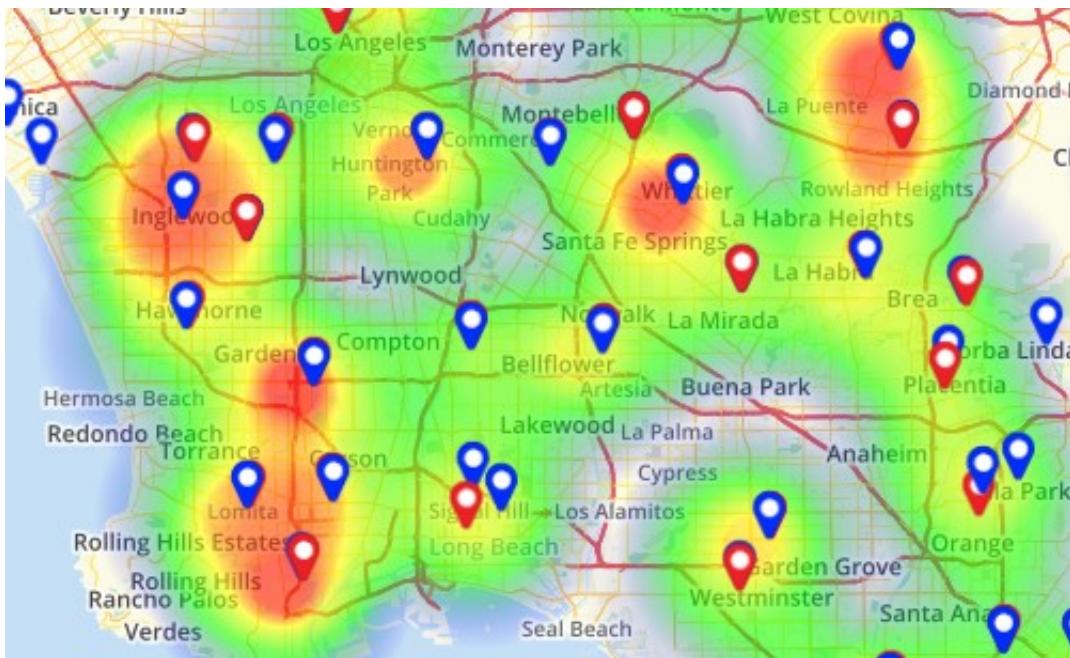
Akcija kreće tako što **istraživač** izda zahtjev za novom akcijom te od voditelja postaje traži da pošalje kvalificirane tragače. Istraživač preko karte koja mu se prikazuje može zadati zadatke svim tragačima na akciji. Tragač se može maknuti s akcije nakon što izvrši sve zadatke koji su mu zadani.

Zadaci su da tragač prođe određenom rutom te dođe do neke lokacije i postavi kameru ili gps uređaj. Na nekoj akciji istraživač i tragač mogu ostaviti komentar na karti.

Cilj svih akcija je pratiti životinje te odrediti povijest njihovog kretanja zbog čega sve praćene životinje imaju gps uređaj. Praćene životinje imaju sljedeće podatke:

- *naziv vrste*
- *slika*
- *opis.*

Sve karte koje se prikazuju su toplinske karte (eng. *heatmaps*). Primjer jedne takve toplinske karte je u nastavku na slici 2.2.



Slika 2.2: Primjer toplinske karte

Istraživač može izabратi koja će mu se karta prikazati. Može pregledavati karte s podacima o praćenim životnjama:

- *povijesne pozicije svih praćenih životinja*
- *trenutne pozicije svih praćenih životinja.*

Dodatna stavka koju treba odabrati je želi li podatke o određenoj jedinki ili o svim životnjama neke vrste.

Osim toga, ima pristup kartama s podacima o tragačima. Karte koje može pre-gledavati sadržavaju sljedeće podatke:

- *povijesne pozicije svih tragača na jednoj akciji*
- *trenutne pozicije svih aktivnih tragača na jednoj akciji.*

Tragače koje želi imati prikazane na karti treba odabrati po tipu prijevoza ili može pregledavati pozicije pojedinačnih tragača.

Također, pristup karti trenutne akcije u kojoj je sudionik ima i tragač. Na njegovoj su karti vidljivi sljedeći podaci:

- *zadaci koje on treba obaviti*
- *trenutna pozicija ostalih aktivnih tragača na toj akciji*
- *trenutna pozicija praćenih životinja.*

Pronašli smo nekoliko aplikacija koje imaju sličnu ulogu kao naša te su navedene u nastavku.

Aplikacija *eWildLife* je aplikacija s geooznakama za praćenje divljih životinja, slučajeva ubijanja istih te sukoba čovjeka i životinja te životinja međusobno. U nastavku na slici 2.3. prikazana je stranica za unošenje novih sukoba u ovoj aplikaciji.

The screenshot shows a mobile application interface titled "SELECT CONFLICT TYPE :". At the top, there is a dropdown menu set to "Authorized Hunting of BLUE BULLS". Below it are two dropdown menus: "DISTRICT" (set to "- Select -") and "Block" (set to "- Select -"). A date input field shows "20/06/2017" with a "SELECT" button next to it. There are two text input fields: "No. of Animals" and "Name of Shooter Hunter", both currently empty. At the bottom, there are two large green buttons: "Submit" on the left and "Take Photo" on the right.

Slika 2.3: Aplikacija eWildLife

Aplikacija *Kwibi* bavi se sličnom problematikom kao prethodno navedena aplikacija, rješava probleme sukoba između ljudi i divljih životinja. Ova aplikacija koristi kartu, što je slično kao i u našoj aplikaciji, a njihov je primjer u nastavku na slici 2.4.



Slika 2.4: Aplikacija Kwibi

Sljedeća je aplikacija *IMammalia* koja je pokrenuta u sklopu projekta Mammal-Net diljem Europe, a između ostalog i na Agronomskom fakultetu u Zagrebu. Na stranicu se može registrirati kao tragač ili osmatrač. Tragači u aplikaciju postavljaju fotografije divljih životinja sa svojih kamera koje osmatrači mogu pregledavati i kategorizirati. U nastavku je primjer jedne takve fotografije na slici 2.5.



Slika 2.5: Aplikacija IMammalia

Naša aplikacija razlikuje se od ostalih po tome što jedina nudi organizaciju cje-lovitog istraživanja i jednostavnu komunikaciju između korisnika (voditelj postaje, istraživač, tragači), dok su ostale aplikacije fokusirane na sukobe između ljudi i životinja te kategorizaciju životinja.

Neke od ideja koje je naš tim smislio, a moglo bi u budućnosti unaprijediti aplikaciju kao što je naša su sljedeće:

- *prikazane male ikonice životinja na karti*
- *klikom na životinju otvara se detaljan opis i fotografija vrste*
- *svijetli i tamni način rada*
- *algoritam za automatsko slanje najkompatibilnijih tragača na akciju, bez posredovanja voditelja postaje.*

# 3. Specifikacija programske potpore

## 3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik (naručitelj)
2. Registrirani korisnici:
  - (a) Voditelji
  - (b) Istraživači
  - (c) Tragači
3. Razvojni tim
4. Administrator

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (sudionik) može:
  - (a) poslati zahtjev za registraciju sa željenom ulogom za koju se prijavljuje (istraživač, voditelj postaje ili tragač na terenu)
2. Istraživač (inicijator) može:
  - (a) stvoriti nove akcije pretraživanja i praćenja
  - (b) poslati zahtjev voditelju za tragača s potrebnim kvalifikacijama
  - (c) pojedinačno zadati zadatke tragačima
    - i. dati komentar za zadani zadatak
  - (d) pristupiti interaktivnoj karti s informacijama o poziciji tragača, životinja i postaja
    - i. ostaviti komentar za ostale sudionike
  - (e) odabrati prikaz na karti (povjesne pozicije životinja, filtrirano po vrsti, filtrirano po jedinku, trenutne pozicije životinja; povjesne pozicije tragača, filtrirano po tipu prijevoza, filtrirano pojedinačno po tragaču, trenutne pozicije tragača)

3. Voditelj postaje (inicijator) može:

- (a) definirati tragače njegove postaje
  - i. definirati način na koji su osposobljeni izvoditi pretraživanje
- (b) odabrati konkretne tragače za pojedinu akciju

4. Tragač na terenu (inicijator) može:

- (a) obavljati zadatke na različite načine (pješke, dronom, automobilom, cross motorom, brodom, helikopterom)
- (b) ostaviti komentar o životinji tijekom akcije
- (c) maknuti se s akcije završetkom potrebnih zadataka
- (d) pristupiti interaktivnoj karti s informacijama o zadacima, poziciji tragača te trenutnim pozicijama životinja
  - i. ostaviti komentar za ostale sudionike

5. Administrator (inicijator) može:

- (a) vidjeti popis registriranih korisnika i njihove podatke
  - i. mijenjati prava i podatke registriranih korisnika
- (b) potvrditi istraživača i voditelja postaje

6. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o postajama i njihovim lokacijama
- (c) pohranjuje sve podatke o životnjama i njihovim lokacijama

### 3.1.1 Obrasci uporabe

#### UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Izrada korisničkog računa za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Odabir opcije registracija
  2. Odabir uloge
  3. Unos osobnih podataka (korisničko ime, fotografija, lozinka, ime, prezime i email adresa)
  4. Slanje zahtjeva za registraciju
  5. Potvrđivanje registracije preko email adrese
- **Opis mogućih odstupanja:**
  - 3.a Zauzeto korisničko ime
    1. Sustav obavještava korisnika da je potrebno upotrijebiti drugo korisničko ime
  - 3.b Krivi format lozinke
    1. Sustav obavještava korisnika o uvjetu koji nije zadovoljen da bi lozinka bila ispravna
  - 3.c Nepotpuna prijava
    1. Sustav obavještava korisnika o podacima koji mu nedostaju za uspješnu registraciju
  - 5.a Nije potvrđena registracija preko email adrese
    1. Sustav obavještava korisnika da mora potvrditi registraciju preko email adrese

#### UC2 - Potvrda registracije od strane administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrda registracije istraživača i voditelja
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač ili voditelj je poslao zahtjev za registraciju i potvrdio registraciju putem mail-a.
- **Opis osnovnog tijeka:**

1. Odabir opcije potvrđivanja registracije istraživača i voditelja

### **UC3 - Prijava u sustav**

- **Glavni sudionik:** Korisnik
- **Cilj:** Korištenje aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Potvrđena registracija
- **Opis osnovnog tijeka:**
  1. Odabir opcije prijava
  2. Unos podataka (korisničko ime i lozinka)
  3. Otvaranje aplikacije
- **Opis mogućih odstupanja:**
  - 2.a Unos pogrešnog korisničkog imena ili lozinke
    1. Sustav obavještava korisnika da je unio pogrešno korisničko ime ili lozinku te mu dozvoljava novi pokušaj

### **UC4 - Uređivanje osobnih podataka**

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijenjeni osobni podaci
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Odabir opcije "Osobni podaci"
  2. Odabir opcije "Uredi"
  3. Promjena željenih podataka
  4. Odabir opcije "Spremi"
  5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 3.a Unos zauzetog korisničkog imena
    1. Sustav obavještava korisnika da je potrebno upotrijebiti drugu korisničko ime
  - 3.b Korisnik unosi krivi format lozinke
    1. Sustav obavještava korisnika o uvjetu koji nije zadovoljen da bi lozinka bila ispravna
  - 4.a Korisnik ne odabire opciju "Spremi"
    1. Sustav obavještava korisnika da nije spremio podatke

### UC5 - Pregled registriranih korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled registriranih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoje registrirani korisnici
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda registriranih korisnika
  2. Pregled podataka

### UC6 - Upravljanje registriranim korisnicima

- **Glavni sudionik:** Administrator
- **Cilj:** Upravljanje registriranim korisnicima
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoje registrirani korisnici
- **Opis osnovnog tijeka:**
  1. Odabir opcije mijenjanja prava i osobnih podataka pojedinog korisnika
  2. Odabir opcije spremanja promjene
  3. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 2.a Administrator ne odabire opciju spremanja promjena
    1. Sustav obaveštava korisnika da nije spremio podatke

### UC7 - Odabir postaje i tragača

- **Glavni sudionik:** Voditelj
- **Cilj:** Voditelj ima dodijeljenu postaju i tragače kojima upravlja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen kao voditelj
- **Opis osnovnog tijeka:**
  1. Voditelj unosi naziv postaje koje će pokrivati
  2. Voditelj odabire tragače koji su dio njegove postaje
- **Opis mogućih odstupanja:**
  - 2.a Voditelj odabire već zauzetu postaju
    1. Sustav obaveštava voditelja da ta postaja već postoji

### UC8 - Definiranje metode pretraživanja tragača

- **Glavni sudionik:** Voditelj

- **Cilj:** Tragač ima definirane metode pretraživanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj ima dodijeljenu postaju i tragače kojima upravlja
- **Opis osnovnog tijeka:**
  1. Voditelj odabire tragača
  2. Voditelj definira načine na koje je odabrani tragač sposoban pretraživati (pješke, dronom, automobilom, cross motorm, brodom ili helikopterom)

### UC9 - Praćenje životinja

- **Glavni sudionik:** Gps uređaj
- **Cilj:** Praćenje životinja
- **Sudionici:** Baza podataka
- **Preduvjet:** Postavljen gps uređaj na životinji
- **Opis osnovnog tijeka:**
  1. Gps uređaj aplikaciji odašilje svoju poziciju
  2. U bazu podataka se zapisuju podaci o životinji
- **Opis mogućih odstupanja:**
  - 1.a Gps uređaj ne radi
    1. Sustav obavještava korisnike da je gps uređaj prestao raditi

### UC10 - Stvaranje akcije pretraživanja i praćenja

- **Glavni sudionik:** Istraživač
- **Cilj:** Stvoriti akciju pretraživanja i praćenja
- **Sudionici:** Baza podataka
- **Opis osnovnog tijeka:**
  1. Istraživač odabire opciju stvaranja nove akcije
  2. Istraživač odabire postaju, broj tragača te potrebne kvalifikacije za akciju
  3. Istraživač šalje zahtjev za tragačima voditelju postaje

### UC11 - Odabir tragača za akciju

- **Glavni sudionik:** Voditelj
- **Cilj:** Odabrati tragače za akciju pretraživanja i praćenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj posjeduje tragače koji nisu niti na jednoj akciji
- **Opis osnovnog tijeka:**

1. Voditelj postaje prima zahtjev od istraživača za određenim brojem tragača s određenim kvalifikacijama
  2. Voditelj postaje odabire tragače za akciju
  3. Voditelj postaje šalje popis odabranih tragača istraživaču
- **Opis mogućih odstupanja:**
    - 2.a Voditelj postaje nema dovoljan broj tragača s potrebnim kvalifikacijama
      1. Voditelj obavještava istraživača kako nema dovoljan broj tragača te mu dodjeljuje one koje ima

### UC12 - Stvaranje zadataka

- **Glavni sudionik:** Istraživač
- **Cilj:** Stvaranje zadataka na pojedinoj akciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir određenog tragača
  2. Odabir opcije dodavanja novog zadatka odabranom tragaču
  3. Odabir zadatka ( prolazak određenom rutom i dolazak do određene lokacije, postavljanje kamere ili uređaja za praćenje)
  4. Istraživač potvrđuje unesene podatke i time dodijeljuje zadatak tragaču
  5. Spremanje podataka u bazu podataka
- **Opis mogućih odstupanja:**
  - 4.a Istraživač ne potvrđuje unesene podatke
    1. Sustav obavještava korisnika da nije potvrdio podatke prije izlaska iz prozora

### UC13 - Ostavljanje komentara za tragače

- **Glavni sudionik:** Istraživač
- **Cilj:** Ostaviti komentare za akciju tragačima
- **Sudionici:** Baza podataka
- **Preduvjet:** Zadaci za tragače su stvoreni
- **Opis osnovnog tijeka:**
  1. Istraživaču se nudi opcija dodavanja komentara na zadatak
  2. Istraživač unosi i spremi komentar
- **Opis mogućih odstupanja:**
  - 4.a Korisnik ne odabire opciju "Spremi"

1. Sustav obaveštava korisnika da nije spremio komentar

#### **UC14 - Prikaz životinja na karti istraživača**

- **Glavni sudionik:** Istraživač
- **Cilj:** Prikaz životinja na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je prijavljen i izradio je akciju
- **Opis osnovnog tijeka:**
  1. Odabir prikaza karte
  2. Odabir prikaza životinja
  3. Odabir informacija koje će se prikazivati na karti (povijesne pozicije svih praćenih životinja filtrirano po vrsti ili pojedinačno po jedinku, trenutne pozicije praćenih životinja)
  4. Odabir opcije generiranja karte
  5. Sustav generira kartu na temelju odabranih podataka (za informacije o povijesnim pozicijama generira se toplinska karta)
- **Opis mogućih odstupanja:**
  - 5.a Korisnik nije odabrao sve potrebne informacije
    1. Sustav obaveštava korisnika da nije odabrao sve potrebne informacije

#### **UC15 - Prikaz tragača na karti istraživača**

- **Glavni sudionik:** Istraživač
- **Cilj:** Prikaz tragača na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je prijavljen i izradio je akciju
- **Opis osnovnog tijeka:**
  1. Odabir prikaza karte
  2. Odabir prikaza tragača
  3. Odabir informacija koje će se prikazivati na karti (povijesne pozicije svih tragača na nekoj akciji, filtrirano po tipu prijevoza ili pojedinačno po tragaču te trenutne pozicije tragača aktivnih na akciji)
  4. Odabir opcije generiranja karte
  5. Sustav generira kartu na temelju odabranih podataka (za informacije o povijesnim pozicijama generira se toplinska karta)
- **Opis mogućih odstupanja:**

5.a Karta se ne generira

1. Sustav obavještava korisnika da nije odabrao sve potrebne informacije

#### **UC16 - Prikaz karte za tragača**

- **Glavni sudionik:** Tragač
- **Cilj:** Prikaz karte za tragača s pripadajućim zadacima te označavanje odrađenih zadataka na akciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Tragač je prijavljen i sudjeluje u određenoj akciji
- **Opis osnovnog tijeka:**
  1. Odabir određene akcije
  2. Odabir prikaza karte za odabranu akciju
  3. Odabir prikaza popisa zadataka, trenutne pozicije ostalih tragača ili trenutne pozicije praćenih životinja za određenu akciju
  4. Tragač može pojedini zadatak označiti odraćenim
  5. Tragač potvrđuje unesene podatke
  6. Spremanje podataka u bazu podataka
- **Opis mogućih odstupanja:**
  - 5.a Tragač ne potvrđuje unesene podatke
    1. Sustav obavještava korisnika da nije potvrdio podatke prije izlaska iz prozora

#### **UC17 - Ostavljanje komentara o životinji**

- **Glavni sudionik:** Tragač/Istraživač
- **Cilj:** Komentar za životinju
- **Sudionici:** Baza podataka
- **Preduvjet:** Tragač je sudjelovao u akciji
- **Opis osnovnog tijeka:**
  1. Pri prikazu karte nudi se opcija "Komentiraj"
  2. Pisanje komentara za praćenu životinju
  3. Spremanje komentara
- **Opis mogućih odstupanja:**
  - 4.a Korisnik ne odabire opciju "Spremi"
    1. Sustav obavještava korisnika da nije spremio komentar

#### **UC18 - Napuštanje akciju**

---

- **Glavni sudionik:** Tragač
- **Cilj:** Tragač više ne sudjeluje u akciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Tragač je ispunio sve dodijeljene zadatke
- **Opis osnovnog tijeka:**
  1. Tragač odabire određenu akciju
  2. Tragač odabire opciju za napuštanje akcije
- **Opis mogućih odstupanja:**
  - 1.a Tragač pokušava napustiti akciju, a nije izvršio sve zadatke
    1. Sustav obavještava tragača da nije ispunio sve zadatke te mu ne dopušta napustiti akciju

#### UC1 - Ocjena i komentiranje završene akcije

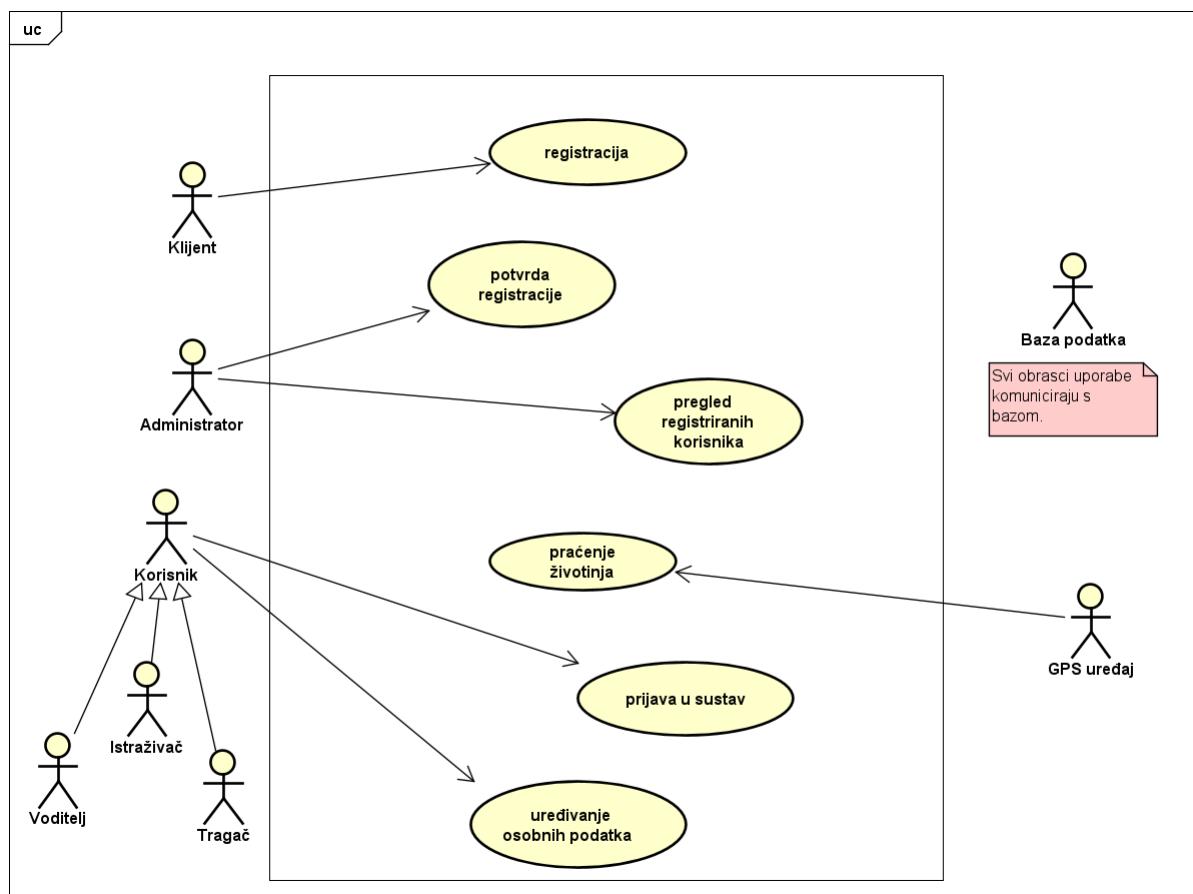
- **Glavni sudionik:** Tragač
- **Cilj:** Tragač ostavlja ocjenu i komentar
- **Sudionici:** Baza podataka
- **Preduvjet:** Tragač je završio akciju
- **Opis osnovnog tijeka:**
  1. Nakon završetka akcije, tragaču se nudi opcija "Ocijeni"
  2. Tragač odabire ocjenu za odrađenu akciju
  3. Tragaču se nudi opcija "Komentiraj"
  4. Tragač piše komentar na akciju
  5. Tragač sprema ocjenu i komentar
- **Opis mogućih odstupanja:**
  - 4.a Korisnik ne odabire opciju "Spremi"
    1. Sustav obavještava korisnika da nije spremio komentar

#### UC20 - Blježenje staza i kretanja tragača

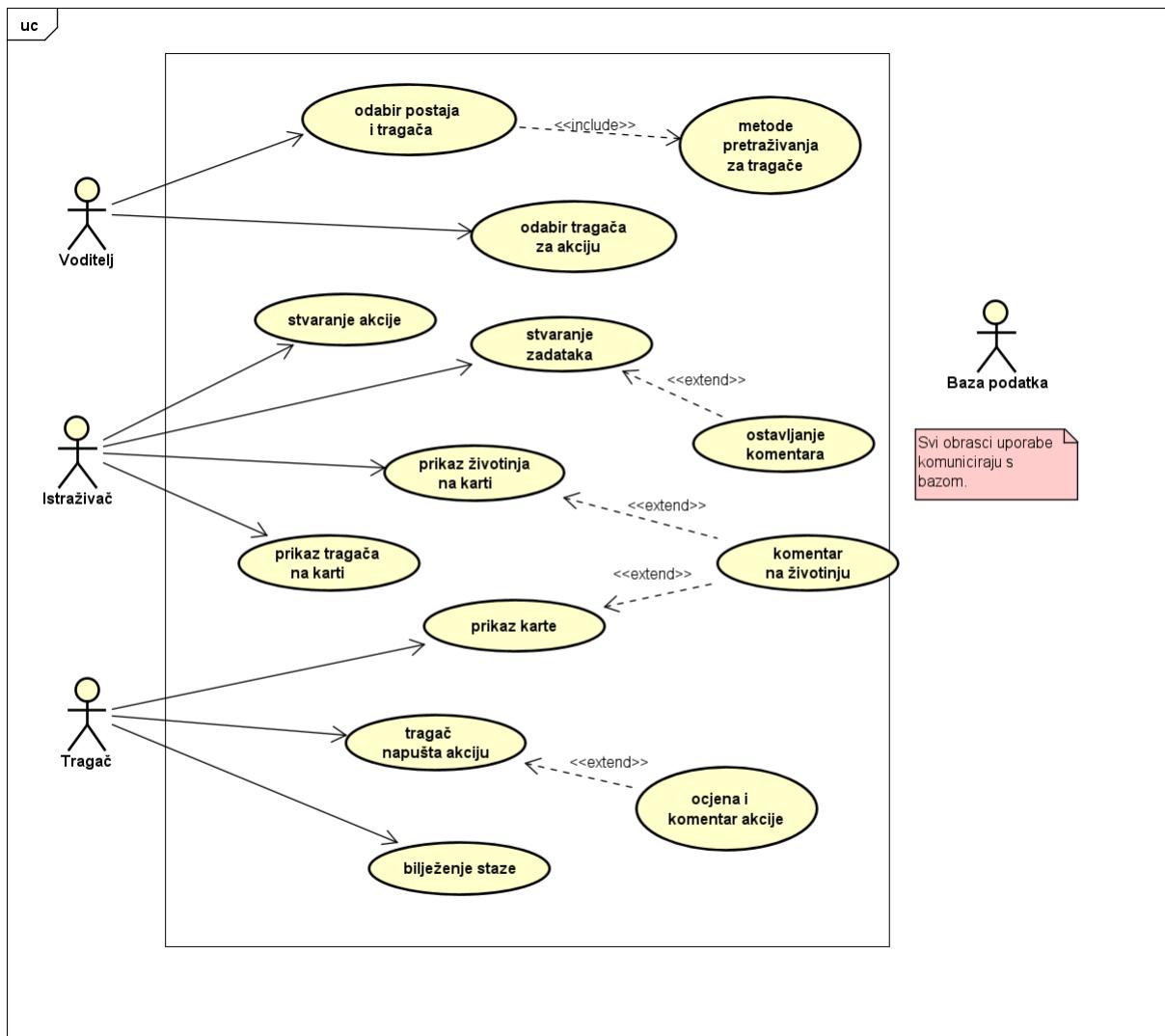
- **Glavni sudionik:** Tragač
- **Cilj:** Bilježenje staza i kretanja tragača
- **Sudionici:** Baza podataka
- **Preduvjet:** Tragač je prijavljen i koristi gps uređaj
- **Opis osnovnog tijeka:**
  1. Tragač ima uključen gps uređaj prilikom obavljanja zadatka
  2. Gps uređaj odašilje svoju poziciju
  3. Tragač ima opciju unijeti način kojim se kretao

4. Tragač potvrđuje unešeni način
- **Opis mogućih odstupanja:**
    - a Tragač nije uključio gps uređaj
      - Sustav obaveštava tragača da nije uključen gps uređaj
    - a Gps uređaj ne radi
      - Sustav obaveštava tragača da gps uređaj ne radi
    - a Tragač ne potvrđuje unesene podatke
      - Sustav obaveštava korisnika da nije potvrdio podatke prije izlaska iz prozora

### Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost administratora, klijenta, korisnika i GPS uređaja

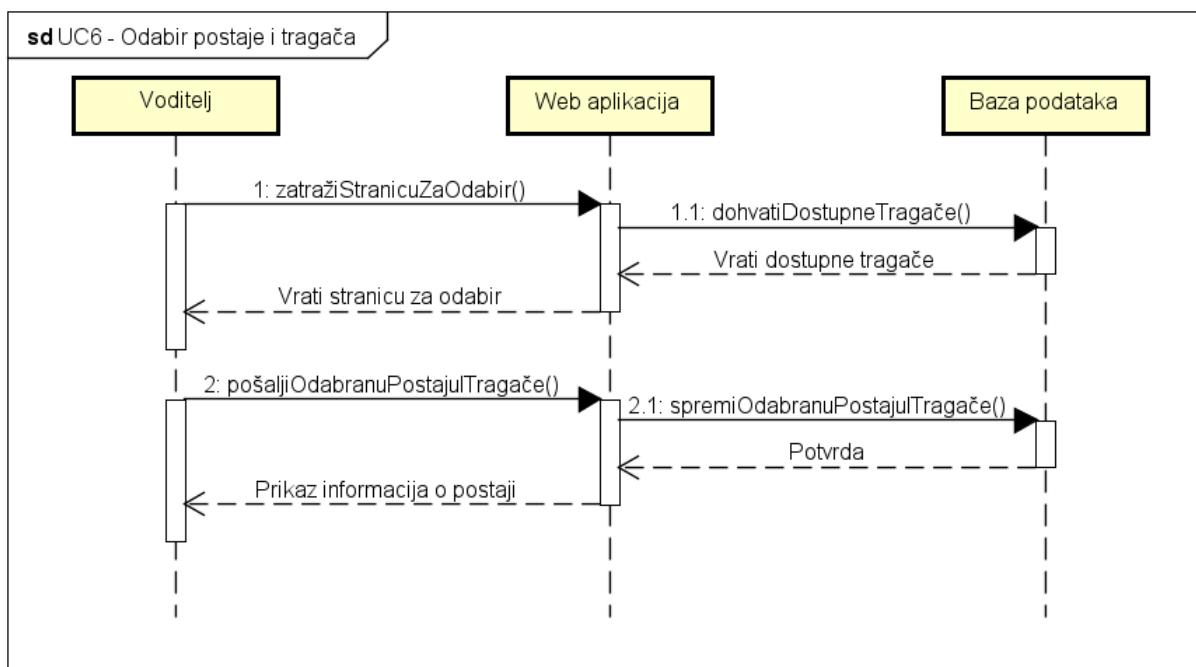


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost voditelja, istraživača i tragača

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC6 - Voditelj odabire svoje postaje i tragače

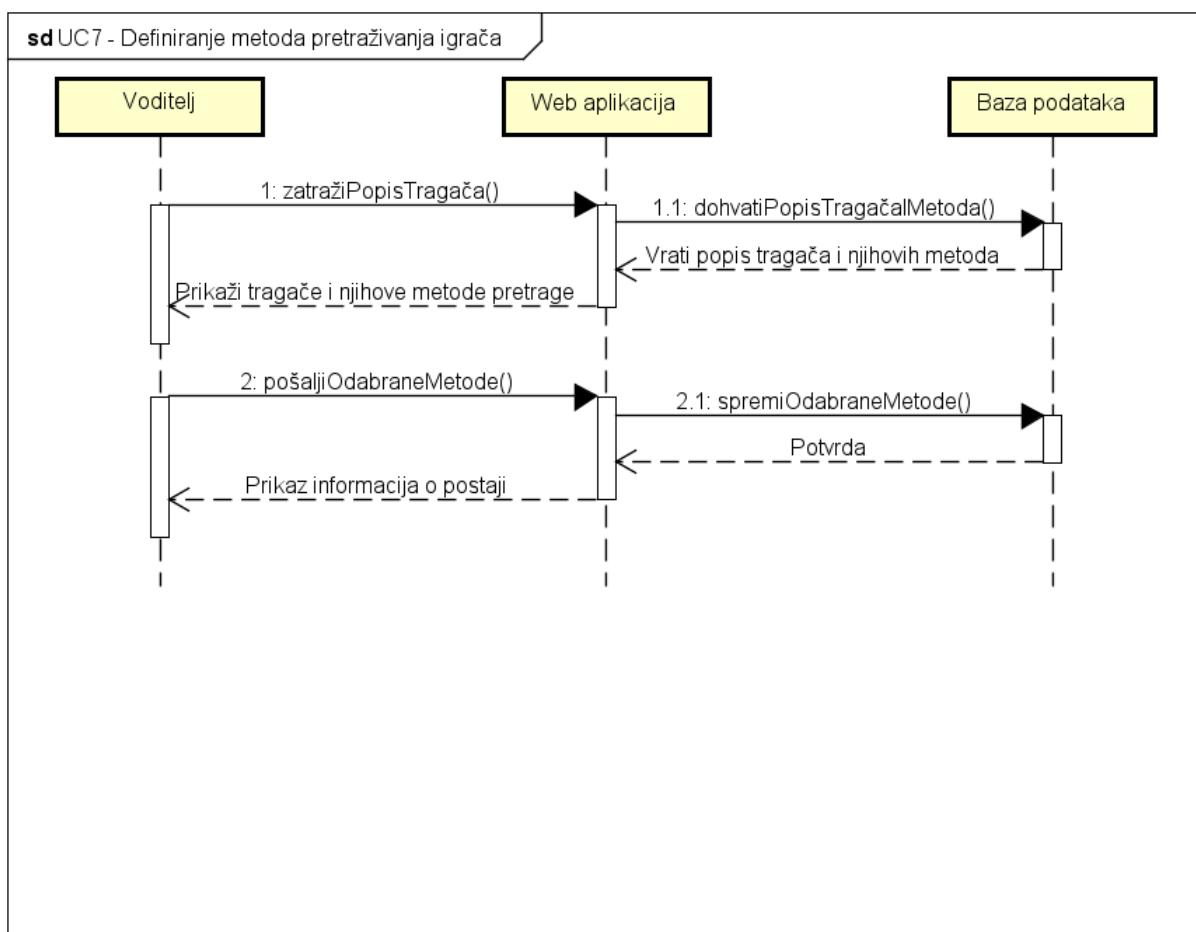
Voditelj šalje zahtjev za prikaz stranice za odabir postaje i tragača. Poslužitelj prima zahtjev te iz baze podataka dohvaća dostupne tragače i prikazuje traženu stranicu voditelju. Voditelj odabire željenu postaju i tragače koji će biti vezani za tu postaju te potvrđuje svoj odabir. Poslužitelj prima potvrdu i sprema podatke o voditelju, postaji i tragačima u bazu podataka. Baza vraća potvrdu, a poslužitelj voditelju prikazuje stranicu s informacijama o njegovoj postaji.



Slika 3.3: Sekvencijski dijagram - UC6

### Obrazac uporabe UC7 - Voditelj definira metode pretraživanja igrača

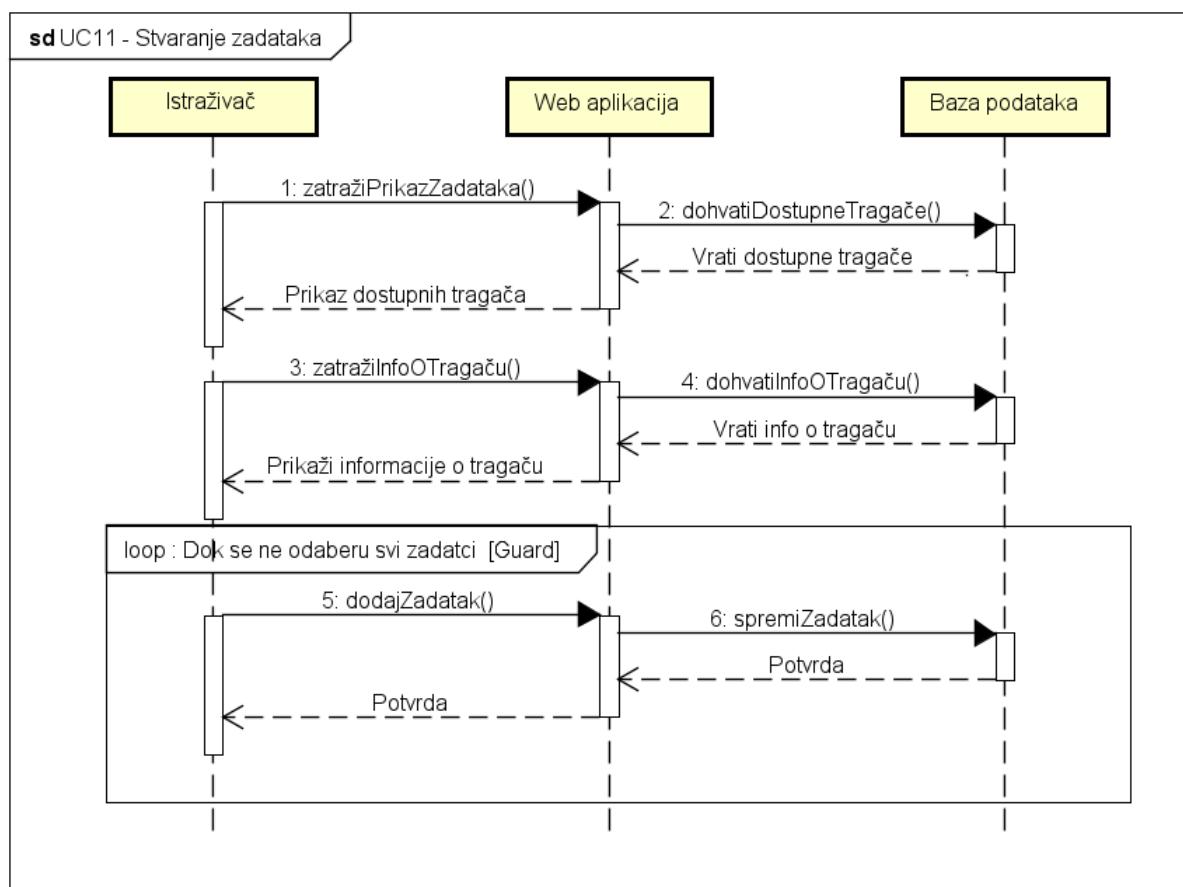
Voditelj šalje zahtjev za prikaz stranice s popisom tragača i njihovih trenutnih metoda pretraživanja. Poslužitelj prima zahtjev te iz baze podataka dohvaća odabrane tragače i njihove metode pretraživanja nakon čega prikazuje voditelju stranicu za promjenu metoda pretraživanja. Ako tragač nema odabranu metodu, nedostupan je za akcije i voditelju je to naznačeno posebnom porukom te je potrebno odabrati jednu od metoda. Voditelj odabire metode pretraživanja svim željenim tragačima te šalje nove podatke poslužitelju. Postlužitelj te podatke šalje u bazu podataka koja vraća poruku potvrde, a poslužitelj voditelju prikazuje stranicu s informacijama o njegovoj postaji.



Slika 3.4: Sekvencijski dijagram - UC7

### Obrazac uporabe UC11 - Stvaranje zadatka

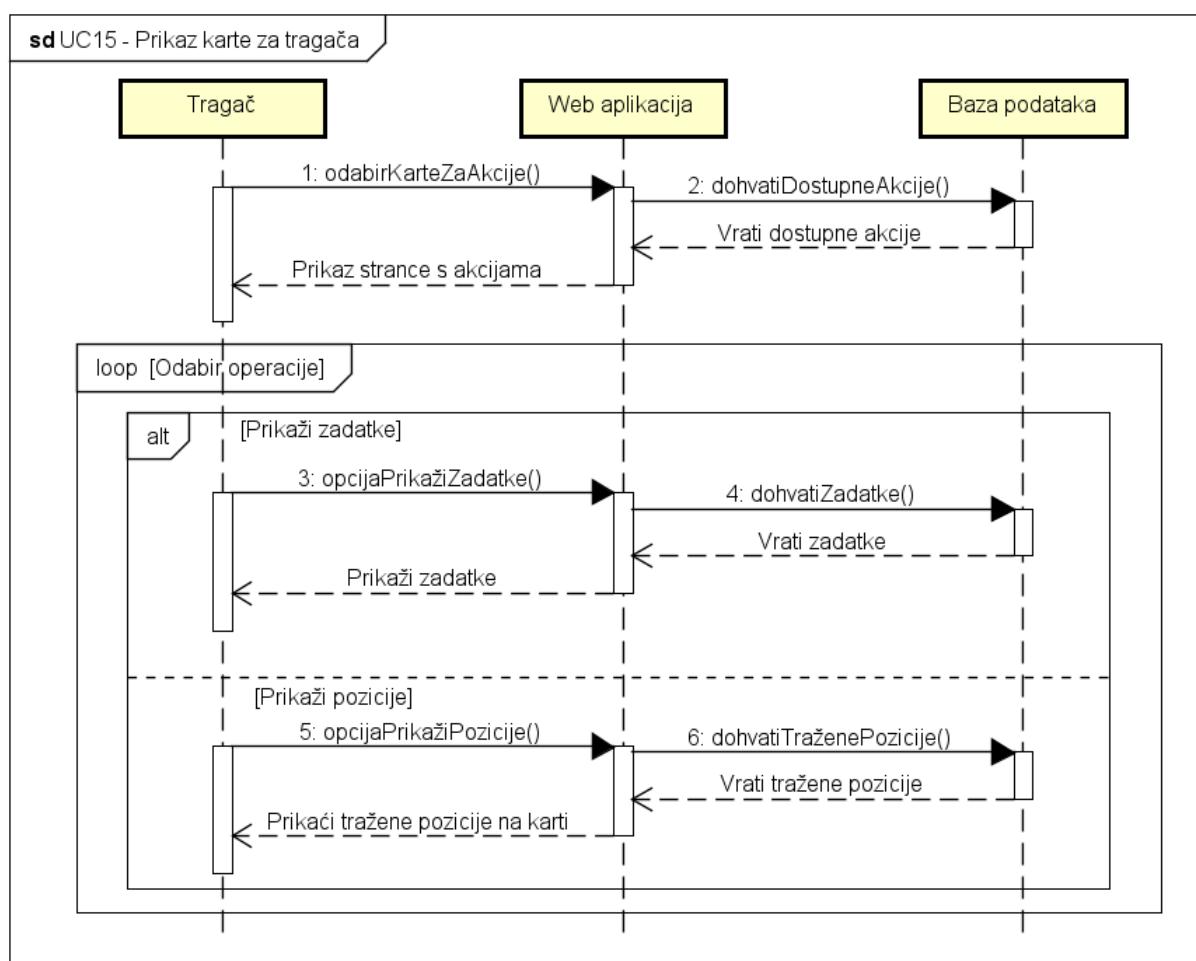
Istraživač šalje zahtjev za prikazom stranice za dodjelu zadataka. Poslužitelj dohvaća dostupne tragače iz baze podataka te ih prikazuje istraživaču. Istraživač dodjeljuje zadatke za svakog tragača posebno (prolazak određenom rutom, dolazak do određene lokacije, postavljanje kamere ili uređaja za praćenje) upisom u formu pojedinačnog tragača. Nakon dodjele zadatka istraživač šalje zahtjev za spremanjem podataka. Poslužitelj prima zahtjev i šalje podatke o tragačima i zadacima u bazu podataka. Poslužitelj odgovara porukom da je sve uspješno izvedeno.



Slika 3.5: Sekvencijski dijagram - UC11

### Obrazac uporabe UC15 - Prikaz karte za tragača

Tragač šalje zahtjev za prikazom stranice za odabir karte. Poslužitelj prima zahtjev, iz baze podataka dohvaća podatke o dostupnim akcijama te tragaču prikazuje stranicu na kojoj može odabrati kartu za određenu akciju te način prikaza informacija na karti (popis zadataka, trenutne pozicije ostalih tragača, trenutne pozicije životinja). Ako tragač odabere popis zadataka, prikazuju mu se njegovi zadaci te ih može pregledati i označiti odraćenim. Ako tragač odabere prikaz karte i trenutnih pozicija, poslužitelj prima zahtjev, dohvaća podatke o pozicijama iz baze podataka i tragaču šalje stranicu na kojoj se prikazuje karta sa svim traženim pozicijama.



Slika 3.6: Sekvencijski dijagram - UC15

### 3.2 Ostali zahtjevi

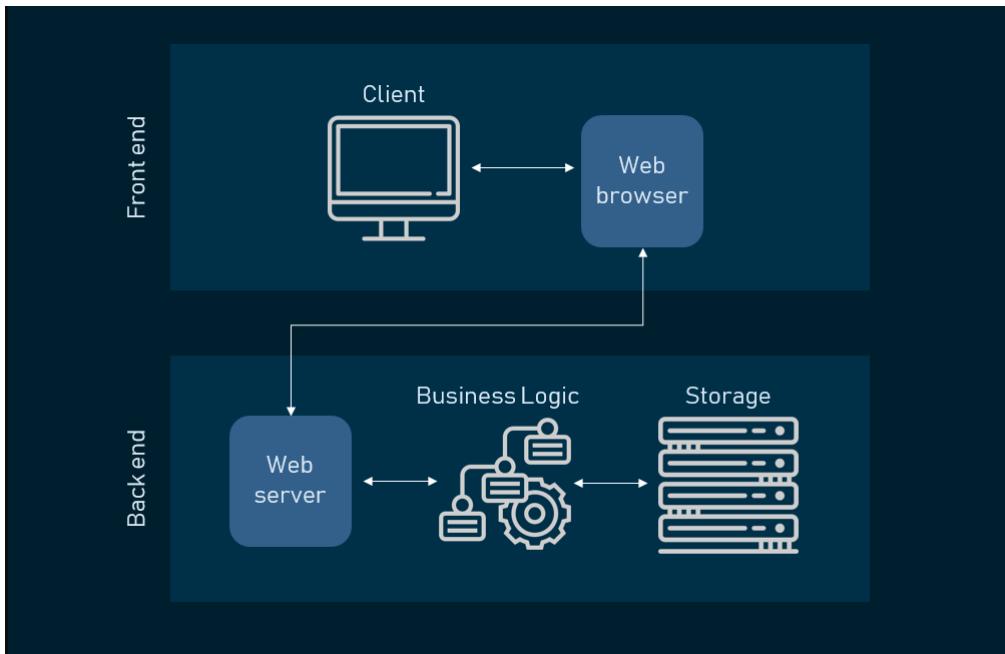
- Vrijeme odziva poslužitelja mora biti maksimalno 10 sekundi.
- Ne smiju postojati neautorizirane naredbe od strane registriranih korisnika.
- Neregistrirani korisnici ne smiju imati pristup informacijama akcija i karata.
- Aplikacija treba podržavati hrvatsku abecedu.
- Neplanirano korištenje korisničkog sučelja ne smije narušiti sigurnost podataka.
- Aplikacija treba biti responzivna.
- Pregledna i lagana navigacija po aplikaciji.
- Prikaz se mora jednostavno i intuitivno prilagoditi veličini zaslona.
- Aplikacija ne smije preopteretiti radnu memoriju.
- Lozinka registriranog korisnika u bazi podataka treba biti kriptirana.
- Komunikacija s bazom podataka treba biti kvalitetno zaštićena.
- Aplikaciju treba moći koristiti više korisnika istovremeno bez poteškoća.

## 4. Arhitektura i dizajn sustava

Arhitektura prati model klijent – server koja omogućava fleksibilnost u razvoju i održavanju budući da backend i frontend mogu evoluirati neovisno jedno o drugome. Komunikacija se vrši putem HTTP protokola. React je korišten za izgradnju frontend-a (klijentske strane), Spring Boot kao backend server, a PostgreSQL kao sustav za upravljanje bazom podataka. Pri izradi korisničkog sučelja komponente su građene tako da su reaktivne i ažuriraju se automatski kada se stanje aplikacije promijeni. Za upravljanje navigacijom unutar aplikacije korišten je React Router. Spring Boot je korišten za izradu backend dijela aplikacije koji obuhvaća:

- poslovnu logiku - servisi komuniciraju s repozitorijima i drugim servisima kako bi izvršili operacije nad podacima
- RESTful API-je - Rest Controlleri su definirani kako bi se obradili API zah-tjevi, mapirani kao RESTful endpointovi
- pristup PostgreSQL bazi podataka – za komunikaciju se definiraju JPA repo-zitoriji koji omogućavaju pristup i upravljanje podacima.

Korisnik putem WEB preglednika šalje zahtjev WEB poslužitelju. WEB poslužitelj omogućava komunikaciju između klijenta i aplikacije putem HTTP protokola. Ukratko, WEB poslužitelj pokreće aplikaciju te joj proslijedi korisnikove zahtjeve. Fron-tend šalje HTTP zahtjeve prema SpringBootu putem fetcha, a backend obrađuje zahtjeve, pristupa bazi podataka, izvršava logiku i vraća rezultat frontendu u obliku JSON objekta. Aplikacija u konačnici vraća odgovor u obliku HTML dokumenta.



Slika 4.1: Komunikacija između frontenda i backenda

## 4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Temeljna jedinica baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Glavna svrha baze podataka je brza i jednostavna pohrana, izmjena i dohvaćanje podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- USER
- RESEARCHER
- MANAGER
- TRACKER
- ANIMAL
- STATION
- ACTION
- ACTION\_ANIMAL

- TASK
- MEDIUM
- QUALIFICATION
- TRACKER\_HISTORY
- ANIMAL\_HISTORY
- ROUTE
- ROUTE\_POINT
- ANIMAL\_COMMENT
- ACTION\_COMMENT
- ACTION\_HABITAT
- CONFIRMATION\_TOKEN
- HABITAT
- REQUEST
- REQUIREMENTS
- SPECIES
- TASK\_ANIMAL
- TASK\_COMMENT
- TRACKER\_ACTION\_MEDIUM

#### 4.1.1 Opis tablica

**USER** Ovaj entitet sadržava sve važne informacije vezane za korisnika aplikacije. Sadrži atribute ID korisnika, korisničko ime korisnika u aplikaciji, fotografiju korisnika, šifru računa korisnika, pravo ime i prezime korisnika, e-mail adresu, tip korisnika (istraživač, tragač ili voditelj stanice) i atribut koji označava je li korisnik registriran tipa BOOLEAN. Ova klasa sadrži zajedničke atribute sva tri tipa

korisnika aplikacije (voditelj, istraživač, tragač) te služi za lakše spremanje podataka koje svaki korisnik mora imati. Ovaj entitet je u vezi *One-to-One* sa svakim tipom korisnika aplikacije. S klasom MAP\_COMMENT je u vezi *One-to-Many* jer više korisnika (mora biti istraživač ili tragač) može ostaviti komentar na mapi.

USER		
id	INT	jedinstveni identifikator (TRACKER.id ili MANAGER.id ili RESEARCHER.id)
username	VARCHAR	korisničko ime
photo	BYTEA	fotografija korisnika
password	VARCHAR	šifra korisničkog računa
name	VARCHAR	ime korisnika
surname	VARCHAR	prezime korisnika
email	VARCHAR	e-mail korisnika
role	VARCHAR	tip korisnika
registered	BOOLEAN	oznaka koja definira je li korisnik registriran

**RESEARCHER** Ovaj entitet sadrži dodatni atribut odobrenja koji je potreban korisniku koji se odlučio za ulogu istraživača u aplikaciji. Povezan je s klasom USER vezom *One-to-One* preko ID ključa. Dodatni atribut nam govori je li korisnik odobren kao istraživač od strane administratora jer je to preduvjet za obavljanje te uloge. U vezi je *One-to-Many* s klasom ACTION (svaki istraživač može organizirati više radnih akcija). Također je s klasom ANIMAL\_COMMENT indirektno preko USER.id u vezi *One-to-Many* jer više istraživača može ostaviti komentar na nekoj životinji.

RESEARCHER		
id	INT	identifikator istraživača (USER.id)
approved	BOOLEAN	oznaka je li istraživač odobren od strane administratora

**MANAGER** Ovaj entitet sadrži dodatne atribute koji opisuju korisnika koji ima ulogu voditelja neke stanice. Moguće je da voditelj vodi samo jednu stanicu i

također da jedna stanica ima samo jednog voditelja i zato je veza klase STATION i MANAGER *One-to-One*. Dodatni atribut approved označava je li korisniku odobren zahtjev za ulogu voditelja od strane administratora (isto kao kod istraživača) te atribut idStation označava jedinstveni identifikator stanice kojoj je korisnik voditelj.

MANAGER		
id	INT	jedinstveni identifikator (USER.id)
approved	BOOLEAN	oznaka odobrenja
idStation	INT	identifikator stanice (STATION.id)

**TRACKER** Ovaj entitet označava korisnika koji obavlja ulogu tragača. Povezan je vezama *One-to-One* s klasom USER i klasom TRACKER\_ACTION\_MEDIUM koja označava tragača koji trenutno obavlja zadatke neke akcije i vozilo kojim se koristi, te vezom *Many-to-One* s klasom STATION i vezom *One-to-Many* s klasom TASK jer jedan tragač može imati više zadataka. Tragač može obavljati zadatke koji su zadani od strane istraživača samo na jednoj stanici, dok stanica može imati više tragača na različitim zadatcima. S klasom ANIMAL\_COMMENT je indirektno preko USER.id u vezi *One-to-Many* jer više tragača može ostaviti komentar na nekoj životinji. Sadrži attribute longitude i latitude koji služe za čuvanje zadnje poznate lokacije tragača. Također je u vezi *Many-to-Many* s klasom MEDIUM što se razrješava tablicom QUALIFICATION.

TRACKER		
id	INT	jedinstveni identifikator tragača (USER.id)
longitude	DOUBLE	geografska dužina
latitude	DOUBLE	geografska širina
idStation	INT	jedinstveni identifikator stanice (STATION.id)

**ANIMAL** Ovaj entitet predstavlja klasu životinja i sadrži sve attribute koji opisuju neku životinju. Sadrži attribute ID životinje, ID vrste životinje, fotografiju životinje, ime jedinke, podatke o lokaciji i tekst koji opisuje životinju. U vezi je *One-to-Many* s klasom TASK jer na nekom zadatku se može pratiti jedna životinja dok više zadataka može pratiti istu životinju. S klasom ANIMAL\_COMMENT

je u vezi *One-to-Many* jer više komentara se može ostaviti za istu životinju. U *One-to-Many* vezi je s klasom ANIMAL\_HISTORY koja sprema podatke o kretnju životinje. Također je u *One-to-Many* vezi s klasom ACTION\_ANIMAL gdje se može vidjeti u kojim akcijama je životinja praćena. Još je u *Many-to-Many* vezi s klasom ACTION što se dodatno razrješava u tablici ACTION\_ANIMAL.

ANIMAL		
id	INT	jedinstveni identifikator
speciesId	INT	vrsta životinje
photo	BYTEA	fotografija životinje
description	TEXT	opis životinje
longitude	DOUBLE	geografska dužina
latitude	DOUBLE	geografska širina
name	VARCHAR	ime jedinke

**STATION** Ovaj entitet predstavlja klasu stanice koja sadrži atribute koje opisuju stanicu; ID stanice, ime stanice, podatke o lokaciji te kratki opis stanice. U vezi je *One-to-One* s klasom MANAGER jer je za svaku stanicu zadužen je točno jedan voditelj te je u vezi *One-to-Many* s klasom TRACKER jer više tragača može raditi za istu stanicu.

STATION		
id	INT	jedinstveni identifikator stanice
longitude	DOUBLE	geografska dužina
latitude	DOUBLE	geografska širina
name	VARCHAR	ime stanice
description	TEXT	opis stanice

**ACTION** Ovaj entitet sadržava sve važne informacije vezane za akciju koju provodi određeni istraživač. Sadrži atribute ID, naslov, ID istraživača koji provodi akciju, ID voditelja postaje gdje se odvija akcija, vrijeme početka, vrijeme kraja te status za trenutno stanje akcije (čeka da postane aktivna, na čekanju, riješena). Ovaj entitet u vezi je *Many-to-One* s entitetom RESEARCHER preko

ID-a istraživača te je u vezi *One-to-Many* s entitetima TASK koji predstavlja zadatak, TRACKER\_ACTION\_MEDIUM koji predstavlja trenutno aktivne tragače u akcijama te njihov način prijevoza, ANIMAL\_COMMENT koji predstavlja komentar tragača vezan za određenu životinju u nekoj akciji te ACTION\_ANIMAL gdje se može vidjeti koje životinje se prate u akciji. Još je u *Many-to-Many* vezi s entitetom HABITAT te ANIMAL što se dodatno razrješava u tablici ACTION\_ANIMAL.

ACTION		
id	INT	jedinstveni identifikator
title	VARCHAR	naslov akcije
idResearcher	INT	ID istraživača (RESEARCHER.id)
idManager	INT	ID voditelja postaje (MANAGER.id)
startOfAction	TIMESTAMP	vrijeme početka
endOfAction	TIMESTAMP	vrijeme kraja
status	INT	kodni broj za trenutno stanje akcije

**ACTION\_ANIMAL** Ovaj entitet sadržava sve važne informacije vezane za praćene životinje u nekoj akciji te razrješava *Many-to-Many* vezu između entiteta ANIMAL i ACTION. Sadrži atribute ID životinje te ID akcije u kojoj je ta životinja praćena. Ovaj entitet u vezi je *Many-to-One* s entitetima ANIMAL preko ID-a životinje i ACTION preko ID-a akcije.

ACTION_ANIMAL		
idAnimal	INT	ID životinje (ANIMAL.id), ujedno i prvi dio kompozitnog primarnog ključa
idAction	INT	ID akcije (ACTION.id), ujedno i drugi dio kompozitnog primarnog ključa

**TASK** Ovaj entitet sadržava sve važne informacije vezane za zadatak koji je dio neke akcije te ga održuje određeni tragač. Sadrži atribute ID, naslov, opis, ID tragača koji održuje zadatku, ID akcije kojoj pripada, trenutnu geografsku širinu i dužinu, geografsku lokaciju početka i kraja zadatka, ID rute za slučaj da je potreban prolazak nekom rutom, vrijeme početka, vrijeme početka i kraja te status koji za trenutno stanje zadatka (čeka da postane aktivan, riješen, prekinut). Ovaj entitet

u vezi je *Many-to-One* s entitetima RESEARCHER preko ID-a istraživača, ACTION preko ID-a akcije, ANIMAL preko ID-a životinje, ROUTE preko ID-a rute.

TASK		
id	INT	jedinstveni identifikator
title	VARCHAR	naslov zadatka
idTracker	INT	ID tragača (TRACKER.id)
idAction	INT	ID akcije (ACTION.id)
latitude	DOUBLE	geografska širina
longitude	DOUBLE	geografska dužina
latStart	DOUBLE	geografska širina početka zadatka
lonStart	DOUBLE	geografska dužina početka zadatka
latFinish	DOUBLE	geografska širina kraja zadatka
lonFinish	DOUBLE	geografska dužina kraja zadatka
idRoute	INT	ID rute (ROUTE.id)
content	TEXT	opis (sadržaj) zadatka
start	TIMESTAMP	vrijeme početka
end	TIMESTAMP	vrijeme kraja
status	TEXT	kodni broj za trenutno stanje zadatka

**MEDIUM** Ovaj entitet sadržava sve važne informacije vezane za sredstva prijevoza koje tragači mogu koristiti u akcijama. Sadrži atribute tip (npr. automobil, zrakoplov...), zračna linija što je oznaka računa li se ruta do neke lokacije za taj tip prijevoza kao pravocrtna (zračna linija), radijus pretraživanja moguć s tim sredstvom, vrijednost na skali koliko dobro se uočavaju detalji s tim tipom sredstva te vrijednost na skali kolika je brzina putovanja tim tipom sredstva. Ovaj entitet u vezi je *Many-to-Many* s entitetom TRACKER što se dodatno razrješava u tablici QUALIFICATION te je u vezi *One-to-Many* s entitetima TRACKER ACTION\_MEDIUM te QUALIFICATION.

MEDIUM		
type	VARCHAR	tip sredstva prijevoza, ujedno i primarni ključ
airline	BOOLEAN	oznaka računa li se pravocrtna ruta
radius	DOUBLE	mogući radijus pretraživanja
detail	DOUBLE	vrijednost na skali koliko dobro se uočavaju detalji
speed	DOUBLE	vrijednost na skali kolika je brzina putovanja

**QUALIFICATION** Ovaj entitet sadržava sve važne informacije vezane za kvalifikacije tragača za tip sredstva prijevoza te razrješava *Many-to-Many* vezu između entiteta TRACKER i MEDIUM. Sadrži atribut ID tragača te tip sredstva prijevoza za koje je taj tragač kvalificiran. Ovaj entitet u vezi je *Many-to-One* s entitetima TRACKER preko ID-a tragača i MEDIUM preko tipa sredstva prijevoza.

QUALIFICATION		
idTracker	INT	ID tragača (TRACKER.id), ujedno i prvi dio kompozitnog primarnog ključa
typeMedium	VARCHAR	tip (ujedno i ID) sredstva prijevoza (MEDIUM.type), ujedno i drugi dio kompozitnog primarnog ključa

**TRACKER\_ACTION\_MEDIUM** Ovaj entitet sadržava sve važne informacije vezane za odnos trenutno aktivnih akcija i tragača koji rade na njima. Sadrži atribut ID, ID tragača, ID akcije te sredstvo prijevoza. Ovaj entitet u vezi je *One-to-One* s entitetom TRACKER preko ID-a tragača te je u vezi *Many-to-One* s entitetima ACTION preko ID-a akcije, MEDIUM preko tipa sredstva prijevoza.

TRACKER_ACTION_MEDIUM		
id	INT	jedinstveni identifikator
idTracker	INT	ID tragača (TRACKER.id), ujedno i primarni ključ
idAction	INT	ID akcije (ACTION.id)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

TRACKER_ACTION_MEDIUM		
typeMedium	VARCHAR	tip (ujedno i ID) sredstva prijevoza (MEDIUM.type)

**TRACKER\_HISTORY** Ova tablica bilježi podatke na mapi, odnosno točke na mapi kojima je tragač prolazio tijekom obavljanja zadataka. U vezi je *Many-to-One* s tablicom TRACKER što znači da za jednog tragača može biti više zabilježenih točaka na mapi. Atributi su ID, ID tragača, vrijeme bilježenja lokacije te zemljopisna dužina i širina.

TRACKER_HISTORY		
id	INT	jedinstveni identifikator
idTracker	INT	identifikator tragača (TRACKER.id)
time	TIMESTAMP	vrijeme bilježenja lokacije
latitude	DOUBLE	zemljopisna širina
longitude	DOUBLE	zemljopisna dužina

**ANIMAL\_HISTORY** Ova tablica zapisane lokacije na kojima je određena životinja prolazila u nekom trenutku. Sadrži atribute ID životinje, vrijeme bilježenja lokacije, zemljopisnu širinu i dužinu. Atributi su ID, ID životinje, vrijeme bilježenja lokacije te zemljopisna dužina i širina. Entitet je u vezi *Many-to-One* s tablicom ANIMAL zato jer se za jednu životinju može zabilježiti više točaka na karti.

ANIMAL_HISTORY		
id	INT	jedinstveni identifikator
idAnimal	INT	identifikator životinje (ANIMAL.id)
time	TIMESTAMP	vrijeme bilježenja lokacije
latitude	DOUBLE	zemljopisna širina
longitude	DOUBLE	zemljopisna dužina

**ANIMAL\_COMMENT** Ovaj entitet sadržava sve važne informacije vezane za komentar nekog tragača o nekoj životinji u nekoj akciji. Sadrži atribute ID, nas-

lov, ID životinje na koju se komentar odnosi, ID tragača koji je napisao komentar, ID akcije u kojoj je komentar napisan, vrijeme izrade te sadržaj komentara. Ovaj entitet u vezi je *Many-to-One* s entitetima ANIMAL preko ID-a životinje, USER preko ID-a korisnika (može biti TRACKER ili RESEARCHER) te ACTION preko ID-a akcije.

ANIMAL_COMMENT		
id	INT	jedinstveni identifikator
title	VARCHAR	naslov komentara
idAnimal	INT	ID životinje (ANIMAL.id)
idUser	INT	ID korisnika (USER.id, indirektno povezano s TRACKER.id ili RESEARCHER.id)
idAction	INT	ID akcije (ACTION.id)
time	TIMESTAMP	vrijeme izrade
content	TEXT	sadržaj komentara

**ACTION\_COMMENT** Ovaj entitet sadržava sve važne informacije vezane za komentar nekog istraživača ili tragača o nekoj nekoj akciji. Sadrži atribute ID, naslov, ID akcije na koju se komentar odnosi, ID korisnika koji je napisao komentar, vrijeme izrade te sadržaj komentara. Ovaj entitet u vezi je *Many-to-One* s entitetima ACTION preko ID-a akcije te USER preko ID-a korisnika (može biti TRACKER ili RESEARCHER).

ACTION_COMMENT		
id	INT	jedinstveni identifikator
title	VARCHAR	naslov komentara
idUser	INT	ID korisnika (USER.id, indirektno povezano s TRACKER.id ili RESEARCHER.id)
idAction	INT	ID akcije (ACTION.id)
time	TIMESTAMP	vrijeme izrade
content	TEXT	sadržaj komentara

**TASK\_COMMENT** Ovaj entitet sadržava sve važne informacije vezane za ko-

mentar nekog tragača ili istraživača o nekom zadatku u nekoj akciji. Sadrži attribute ID, naslov, ID zadatka na koji se komentar odnosi, ID korisnika koji je napisao komentar, vrijeme izrade te sadržaj komentara. Ovaj entitet u vezi je *Many-to-One* s entitetima TASK preko ID-a zadatka i preko njega indirektno i s ACTION te USER preko ID-a korisnika (može biti TRACKER ili RESEARCHER).

TASK_COMMENT		
id	INT	jedinstveni identifikator
title	VARCHAR	naslov komentara
idTask	INT	ID zadatka (TASK.id)
idUser	INT	ID korisnika (USER.id, indirektno povezano s TRACKER.id ili RESEARCHER.id)
time	TIMESTAMP	vrijeme izrade
content	TEXT	sadržaj komentara

**ACTION\_HABITAT** Ovaj entitet sadržava sve važne informacije vezane za praćene životinje u nekom staništu te razrješava *Many-to-Many* vezu između entiteta ANIMAL i HABITAT. Sadrži attribute ID staništa te ID akcije u kojoj je ta životinja praćena. Ovaj entitet u vezi je *Many-to-One* s entitetima HABITAT preko ID-a staništa i ACTION preko ID-a akcije.

ACTION_HABITAT		
idHabitat	INT	ID staništa (HABITAT.id), ujedno i prvi dio kompozitnog primarnog ključa
idAction	INT	ID akcije (ACTION.id), ujedno i drugi dio kompozitnog primarnog ključa

**CONFIRMATION\_TOKEN** Ovaj entitet sadržava sve važne informacije vezane za kreiranje tokena radi sigurnije prijave korisnika pomoću potvrde email adrese. Sadrži attribute ID, tekst koji je sifra tokena, vremena kreiranja, potvrđivanja i isteka tokena te strani ključ idUser. Ovaj entitet u vezi je *Many-to-One* s entitetom USER preko ID-a korisnika.

CONFIRMATION_TOKEN		
id	INT	tip sredstva prijevoza, ujedno i primarni ključ
idUser	INT	ID korisnika (USER.id)
createdAt	TIMESTAMP	vrijeme izrade
confirmedAt	TIMESTAMP	vrijeme potvrde
expiresAt	TIMESTAMP	vrijeme isteka
token	TEXT	šifra tokena

**SPECIES** Ovaj entitet predstavlja klasu vrsta životinja i sadrži sve atribute koji opisuju neku vrstu. Sadrži atribute ID, fotografiju vrste, ime vrste i tekst koji opisuje životinju. U vezi je *One-to-Many* s klasom ANIMAL jer jedna vrsta može imati više jediniki životinja, a jedna jedinka može biti samo jedne vrste.

SPECIES		
id	INT	jedinstveni identifikator
photo	BYTEA	fotografija vrste
description	TEXT	opis vrste
name	VARCHAR	ime vrste

**HABITAT** Ovaj entitet predstavlja klasu različitih staništa i sadrži sve atribute koji opisuju neko stanište. Sadrži atribute ID, fotografiju staništa, ime staništa, tekst koji opisuje stanište, radius koji obuhvaća stanište te geografsku lokaciju staništa. U vezi je *One-to-Many* s klasom ANIMAL jedno stanište može biti stanište više životinja, a jedna životinja ima isključivo to stanište te je u vezi *One-to-Many* s klasom ACTION jer jedna akcija može imati više staništa, a stanište može biti dio više akcija.

HABITAT		
id	INT	jedinstveni identifikator
photo	BYTEA	fotografija staništa
description	TEXT	opis staništa

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

HABITAT		
name	VARCHAR	ime staništa
radius	DOUBLE	radijus staništa
latitude	DOUBLE	geografska širina
longitude	DOUBLE	geografska dužina

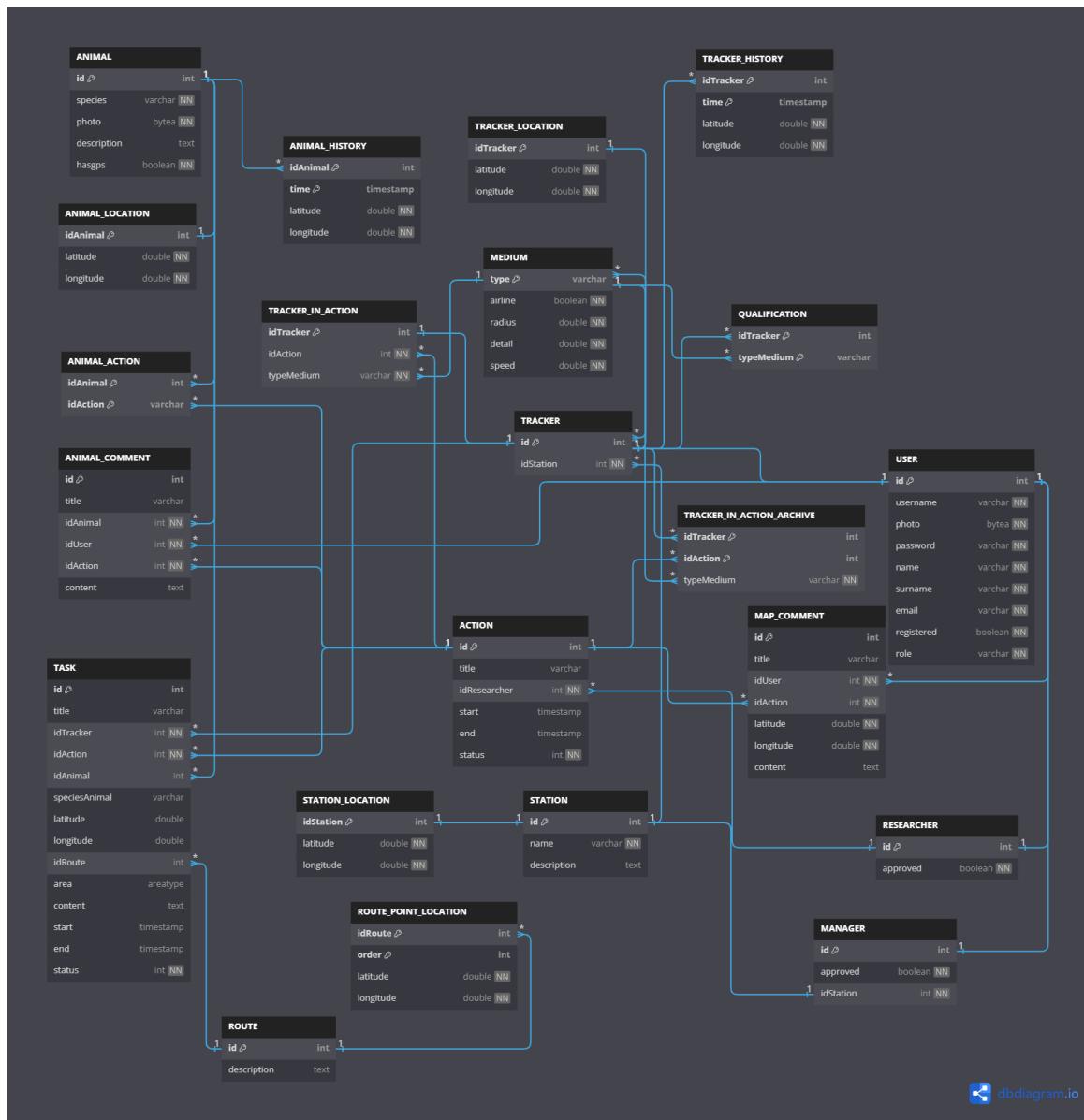
**REQUEST** Ovaj entitet sadržava sve važne informacije vezane za zahtjev unutar određene akcije. Sadrži attribute ID, ID korisnika koji je kreirao zahtjev, ID akcije, tip zahtjeva, status zahtjeva te vrijeme izrade. Ovaj entitet u vezi je *One-to-One* s entitetom USER preko ID-a korisnika te je u vezi *Many-to-one* s entitetom ACTION koji predstavlja akciju.

REQUEST		
id	INT	jedinstveni identifikator
idResearcher	INT	ID korisnika (USER.id)
idAction	INT	ID akcije (ACTION.id)
creationTime	TIMESTAMP	vrijeme izrade
status	VARCHAR	trenutno stanje akcije
type	VARCHAR	tip zahtjeva

**REQUIREMENTS** Ovaj entitet sadržava sve važne informacije vezane za specifične uvjete koje voditelj traži za pojedinu akciju. Sadrži attribute ID akcije, vrstu vozila tragača i količinu tih pojedinih tragača potrebnih za akciju. Ovaj entitet u vezi je *One-to-One* s entitetom ACTION preko ID-a akcije.

REQUIREMENTS		
idAction	INT	ID akcije (ACTION.id)
amount	INT	broj tragača s pojedinim vozilom
type	VARCHAR	tragači s traženim tipom vozila

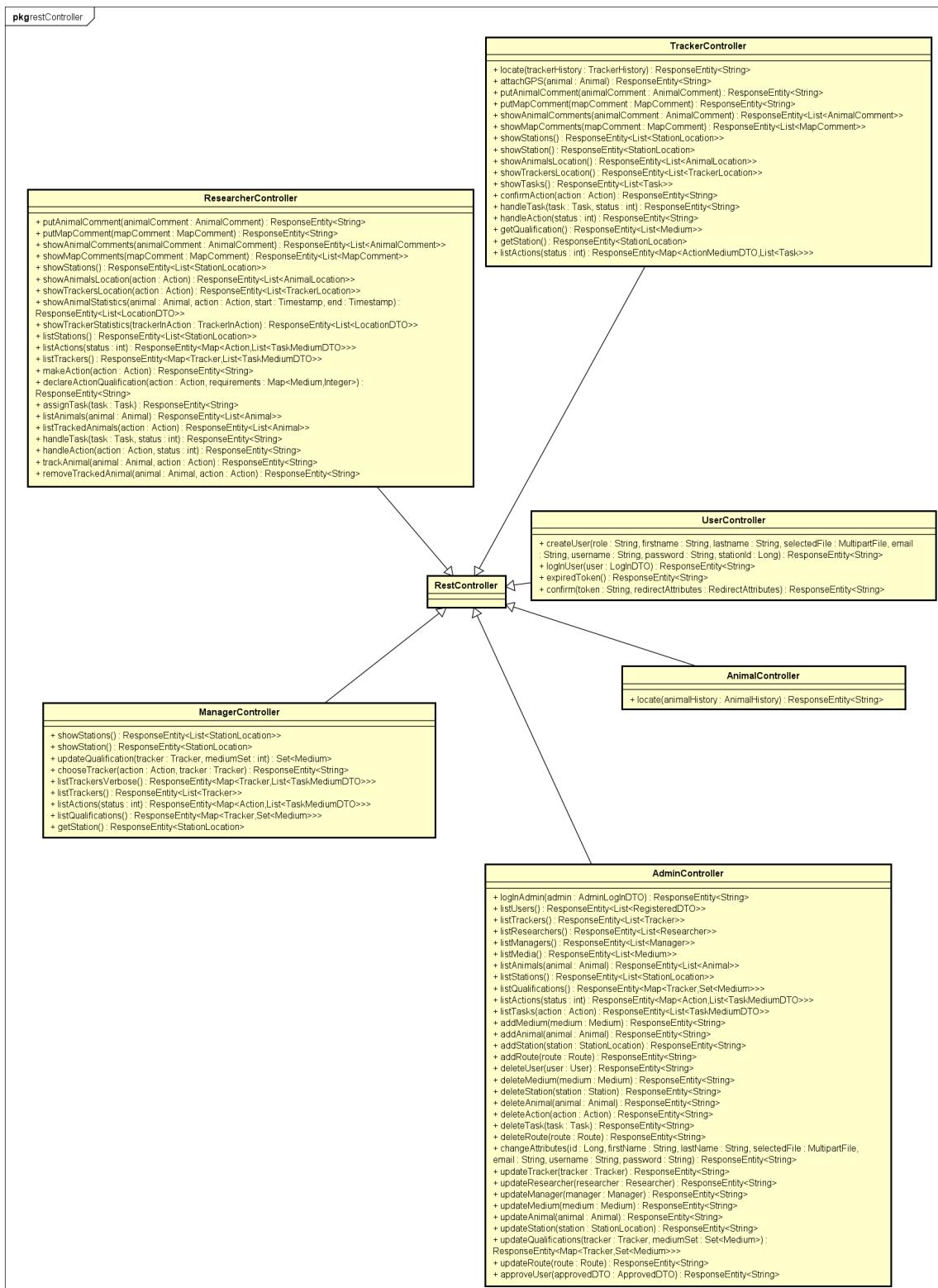
#### 4.1.2 Dijagram baze podataka



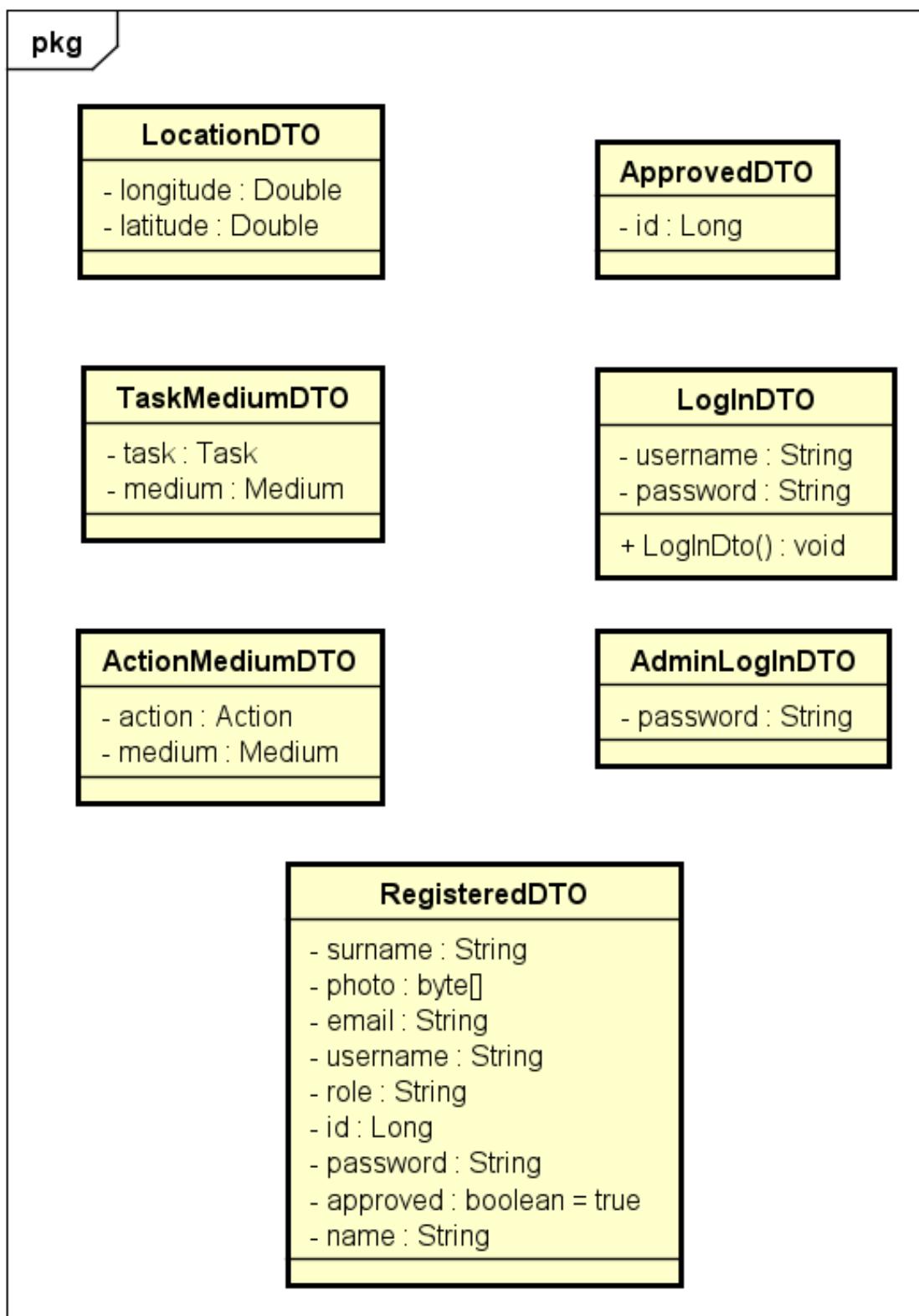
Slika 4.2: E-R dijagram baze podataka

## 4.2 Dijagram razreda

Na slikama 4.4, 4.5 i 4.6 prikazani su razredi koji pripadaju backend dijelu MVC arhitekture. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Metode implementirane u tim razredima obrađuju zahtjeve aktora koji dolaze s frontend servera obavljajući potrebne manipulacije podacima i šaljući odgovarajuće odgovore. Metode u Controller razredima obično vraćaju podatke u JSON formatu, a HTTP status kodovi koriste se za signalizaciju statusa zahtjeva (npr., uspješan odgovor, greška, itd.). Radi lakše organizacije, razredi su logički podijeljeni prema pravu pristupa metodama određenih aktera kako bi se smanjila prepunućenost unutar dijagrama. Prikazane su samo ovisnosti između razreda koji pripadaju istom dijelu dijagrama. Iz naziva i tipova atributa u razredima može se zaključiti vrsta ovisnosti među različitim razredima. Također se ponekad koriste dodatne DTO (Data transfer object) klase za specifične potrebe prijenosa podataka. Kod korištenja objekata koji predstavljaju modele ili DTO objekata u implementaciji selektivno se postavljaju atributi koji su potrebni za određene operacije, a ostali atributi su automatski postavljeni na null. To omogućuje modularno korištenje funkcija i zbog toga modeli i DTO objekti mogu imati reference na druge modele.

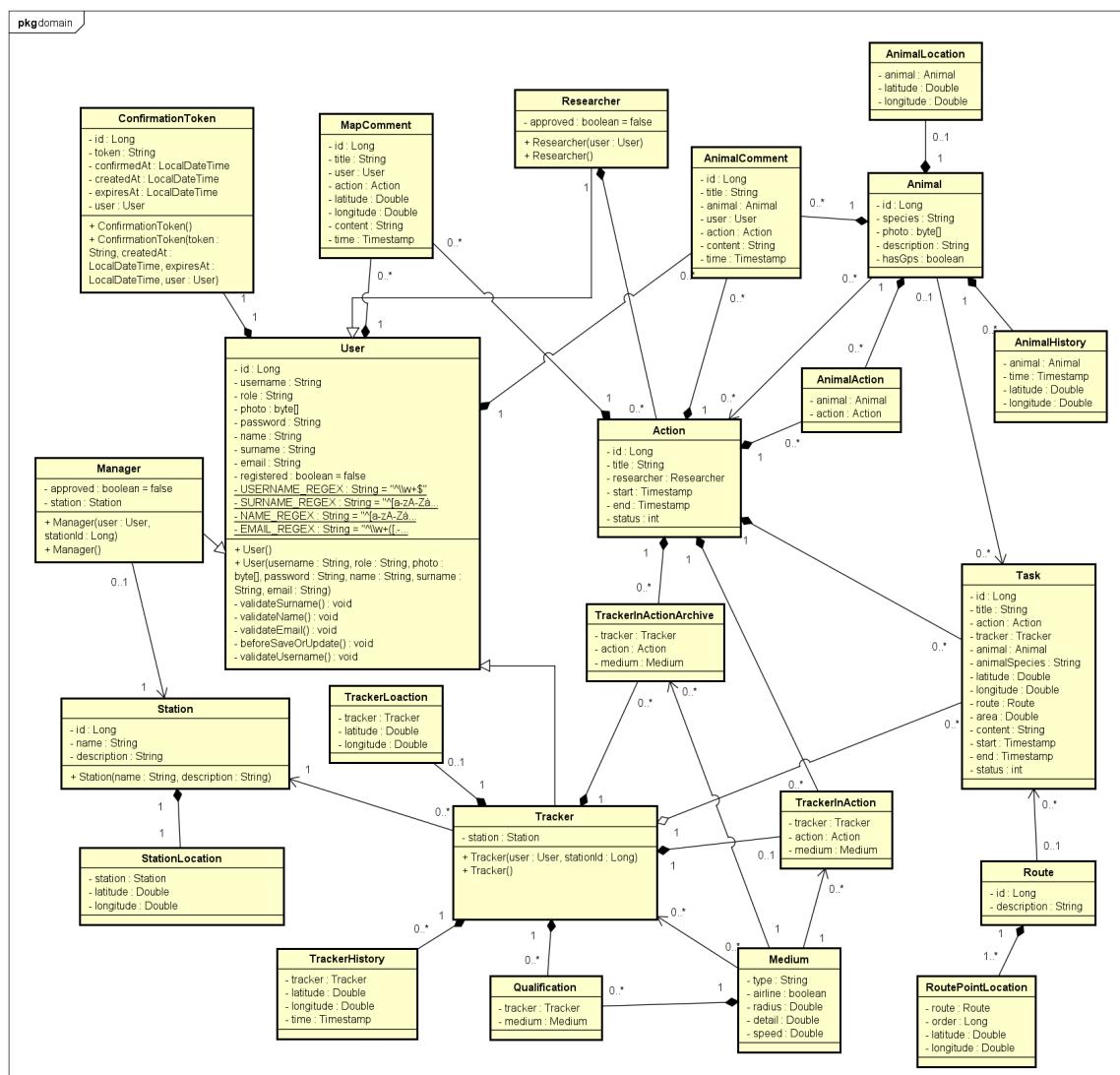


Slika 4.3: Dijagram razreda - dio Controllers



Slika 4.4: Dijagram razreda - dio Data transfer objects

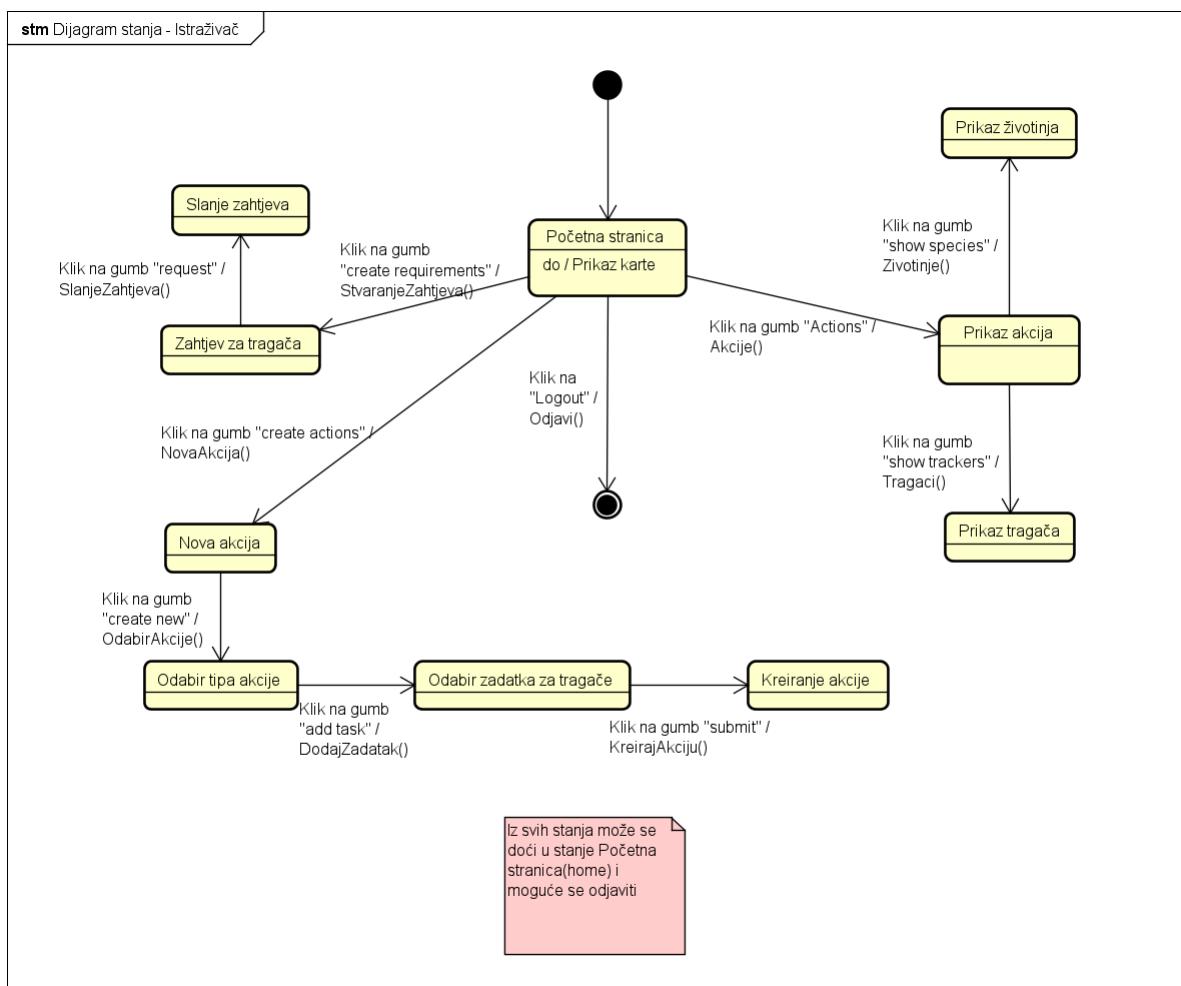
Model razredi preslikavaju strukturu baze podataka u aplikaciji. Svi entiteti realiziraju javne get i set metode za svoje privatne attribute. Modeli Researcher (istraživač), Tracker (tragač) i Manager (voditelj stanice) nasljeđuju model User (korisnik) te imaju svoje specifične attribute. Oni predstavljaju tri glavna tipa korisnika aplikacije od kojih svaki može upućivati specifične zahtjeve koji odražavaju mogućnosti tog tipa korisnika. Svaki korisnik se mora registrirati da bi koristio aplikaciju te korisnike tipa Researcher i Manager administrator mora dodatno potvrditi. Istraživači organiziraju akcije s ciljem praćenja i prikupljanja podataka o životinjama, a na tim akcijama zadatke koje zadaje istraživač odrađuju tragači koji rade na toj akciji i pripadaju određenoj stanici koja ima svog voditelja koji odabire koji će tragači sudjelovati u nekoj akciji na temelju zahtjeva istraživača i kvalifikacija tragača.



Slika 4.5: Dijagram razreda - dio Models

### 4.3 Dijagram stanja

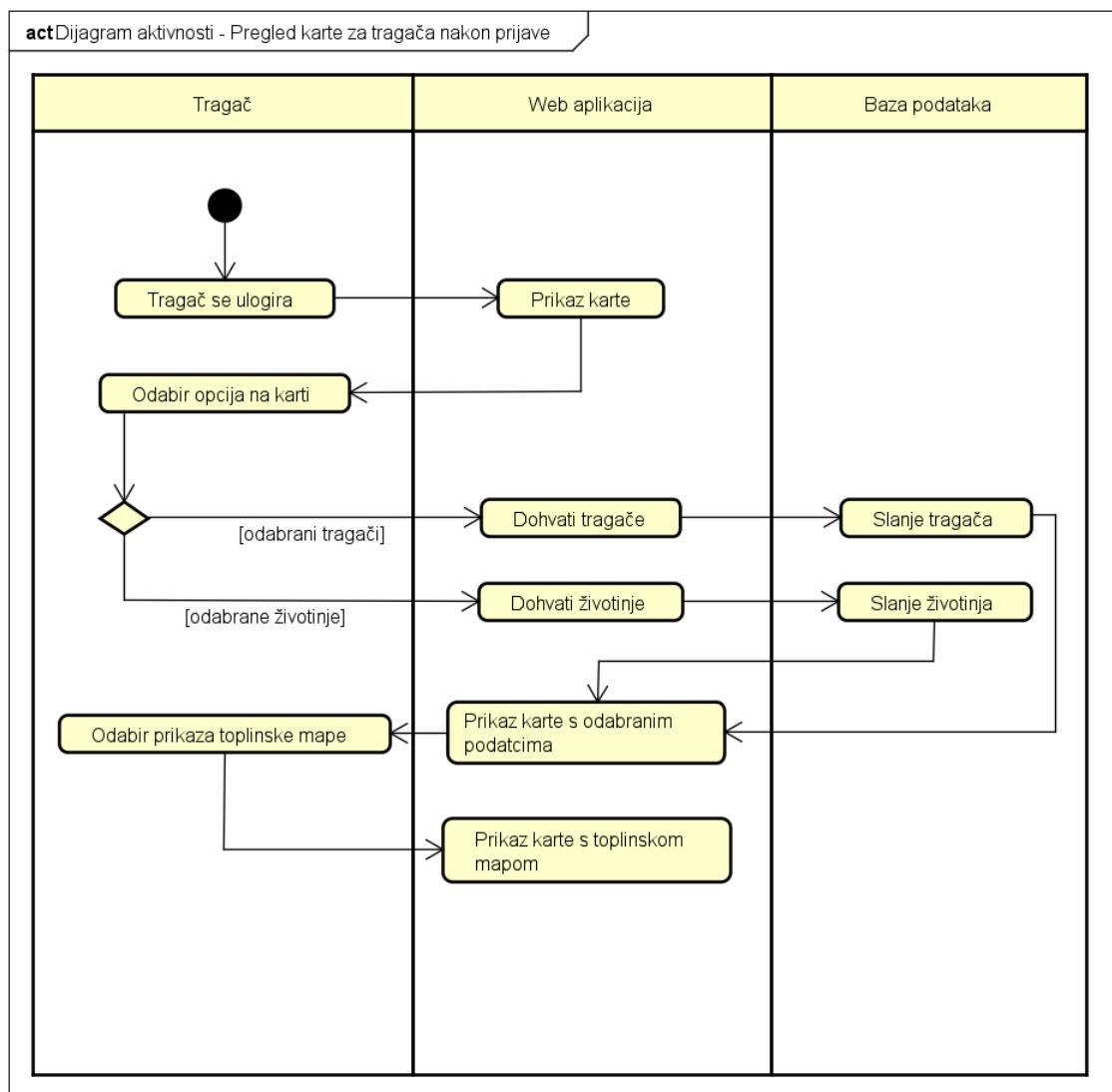
Dijagram stanja opisuje dinamičko ponašanje dijela sustava i prikazuje stanja objekata. Na slici ispod(slika 4.7) prikazuje se dijagram stanja za korisnika registriranog i ulogiranog kao Istraživač(Researcher). Istraživaču se pokazuje početna stranica gdje ima mogućnost odabira 3 opcije: prikaza akcije, slanja zahtjeva za tragača i kreiranja nove akcije. Pri prikazu akcije istraživač može pogledati životinje koje se traže u akciji i može pogledati koji tragači su aktivni u akciji. Istraživač može popuniti zahtjev za tragača koji mu je potreban u akcijama i poslati ga na uvid voditelju. Novu akciju istraživač dodaje tako da klikne na dugme „create actions“, odabere tip akcije, odabere zadatke za svakog tragača i na kraju kreira akciju. Istraživač se može odjaviti u svakom trenutku.



Slika 4.6: Dijagram stanja

## 4.4 Dijagram aktivnosti

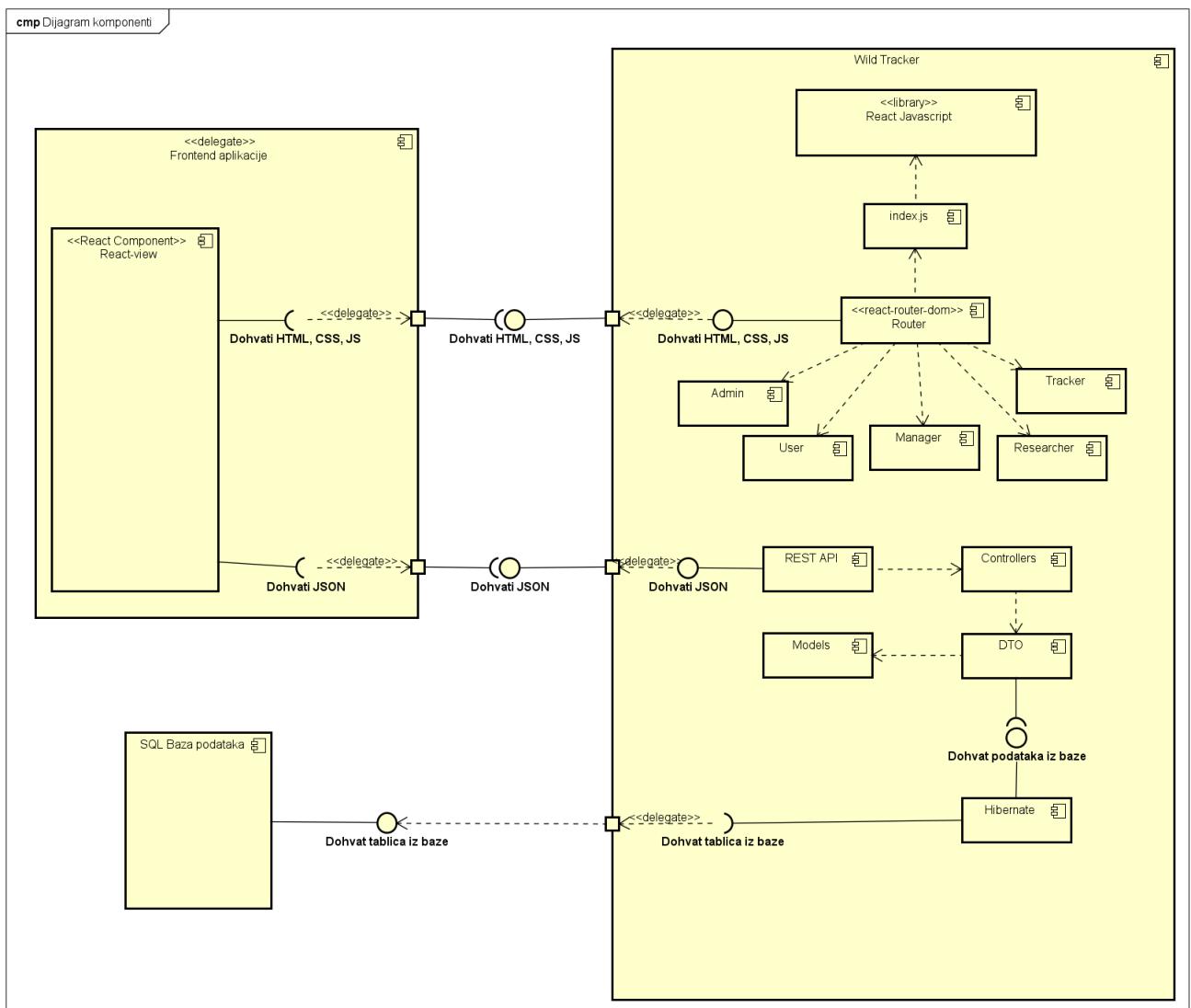
Dijagram aktivnosti koristi se za opis modela toka upravljanja. Svaki korak u dijagramu odvija se nakon prethodno završenog te je dijagram vrlo čitljiv i lako razumljiv. Na slici ispod(slika 4.8) prikazan je dijagram aktivnosti za proces odabira podataka za prikazivanje na karti. Nakon što se korisnik prijavlji u sustav, prikazuje mu se karta. Na njoj može odabrati opciju prikaza tragača ili opciju prikaza životinja. Aplikacija prikazuje željene podatke te korisnik nakon toga može odabrati prikaz toplinske karte i aplikacija mu to prikazuje.



Slika 4.7: Dijagram aktivnosti

## 4.5 Dijagram komponenti

Dijagram komponenti opisuje organizaciju i međuvisnost komponenti, interne strukture i odnose prema okolini. Na slici ispod prikazan je dijagram sustava aplikacije WildTrack. Sustavu se pristupa preko dva sučelja. Pomoću sučelja za dohvata HTML, CSS i JS datoteka poslužuju se datoteke potrebne za frontend dio aplikacije. Pomoću router komponente, koja je dio Reacta, određuje se koje će se datoteke poslati na sučelje aplikacije. Na frontendu se nalaze JavaScript datoteke koje zajedno čine razne komponente nazvane po aktorima kojima se pristupa. Preko sučelja za dohvata podataka u JSON obliku pristupa se REST API komponenti koja poslužuje podatke s backend dijela aplikacije. Hibernate komponenta dohvaća podatke iz SQL baze podataka te ih prosljeđuje DTO-u. Controller komponenta zadužena je primanje upita i odlučuje što se uzima od DTO komponente. React-view komponenta komunicira s aplikacijom i prikazuje potrebne podatke ovisno o zahtjevima.



Slika 4.8: Dijagram komponenti

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

Komunikacija u timu odrađena je pomoću aplikacija Whatsapp<sup>1</sup> i Discord<sup>2</sup>. Za potrebe izrade dijagrama korišten je alat Astah UML<sup>3</sup>. Kao sustav za praćenje verzija i upravljanja kodom korišten je Git<sup>4</sup>, a kao zajednički repozitorij korišten je GitHub<sup>5</sup>. Kao razvojno okruženje za potrebe frontend developmenta korišten je Visual studio code<sup>6</sup> tvrtke Microsoft, a za potrebe izrade backend dijela korišten je IntelliJ<sup>7</sup> tvrtke JetBrains.

Za izradu frontend dijela korišten je React<sup>8</sup>, kojeg je napravila i usavršila tvrtka Meta, Chakra UI<sup>9</sup> kao alternativu css-u, a za izradu backend dijela aplikacije korišten je okvir SpringBoot<sup>10</sup>. React je jedan od najpopularnijih okvira za izradu frontend dijela web aplikacije te se lako mogu pronaći potrebne informacije o njegovom korištenju. React se vrlo dobro uklapa sa SpringBootom koji također ima vrlo široku primjenu i veliku zajednicu developera. Baza podataka je napravljena koristeći PostgreSQL<sup>11</sup> i nalazi se na poslužitelju. Za testiranje aplikacije korišteni su Postman<sup>12</sup> i Selenium<sup>13</sup>.

1. <https://www.whatsapp.com/>
2. <https://discord.com/>
3. <https://astah.net/>
4. <https://git-scm.com/>
5. <https://github.com/>
6. <https://code.visualstudio.com/>
7. <https://www.jetbrains.com/idea/>
8. <https://react.dev/>
9. <https://chakra-ui.com/>
10. <https://spring.io/projects/spring-boot/>
11. <https://www.postgresql.org/>
12. <https://www.postman.com/>
13. <https://www.selenium.dev/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Ispitivanje komponenti koristi se za verificiranje rada programskih dijelova koje je moguće zasebno ispitati u izolaciji. Najčešće se ispituju pojedinačne funkcije ili metode unutar raznih objekata.

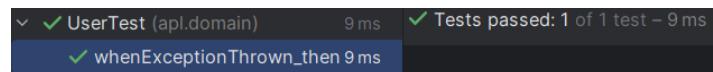
#### 1. ispitni slučaj: Ispitivanje metode za validaciju emaila slanjem neispravnog oblika maila

U ovom ispitivanju testiramo rad metode za provjeru je li email dobrog oblika. Testiramo na način da pošaljemo email koji nije validan. Test hvata iznimku te ako je test prošao, to znači da se dogodila iznimka i email nije validan. Šaljemo mail "email00". U svrhu testa metoda je pretvorena u public metodu, a u kodu je ostavljena kao private.

Kôd testa:

```
@Test(expected = IllegalStateException.class)
public void whenExceptionThrown_thenMailBad() {
    user = new User("Username", "Role",
HexFormat.of().parseHex("e04fd020ea3a6910a2d808002b30309d"), "password",
"name", "surname", "email00");
    user.validateEmail();
    //assertEquals(1, 1);
}
```

Rezultat:



Zadan je neispravan oblik emaila, test je prošao što je u ovom slučaju i očekivano.

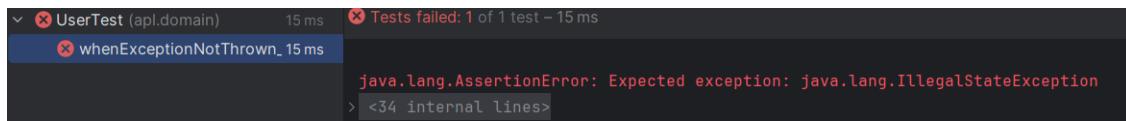
#### 2. ispitni slučaj: Ispitivanje metode za validaciju emaila slanjem ispravnog oblika maila

U ovom ispitivanju testiramo rad metode za provjeru je li email dobrog oblika. Testiramo na način da pošaljemo email koji je validan. Test hvata iznimku te ako je test prošao, to znači da se dogodila iznimka i email nije validan. Ako se nije dogodila iznimka, test nije prošao i email je validan. Šaljemo mail "email@mail.com". U svrhu testa metoda je pretvorena u public metodu, a u kodu je ostavljena kao private.

Kôd testa:

```
@Test(expected = IllegalStateException.class)
public void whenExceptionNotThrown_thenMailGood() {
    user = new User("Username", "Role",
HexFormat.of().parseHex("e04fd020ea3a6910a2d808002b30309d"), "password",
"name", "surname", "email@mail.com");
    user.validateEmail();
    //assertEquals(1, 1);
}
```

Rezultat:



Zadan je ispravan oblik emaila i test nije prošao što je očekivan rezultat.

### 3. ispitni slučaj: ispitivanje metode za validaciju korisničkog imena(username) unoseći ispravan username

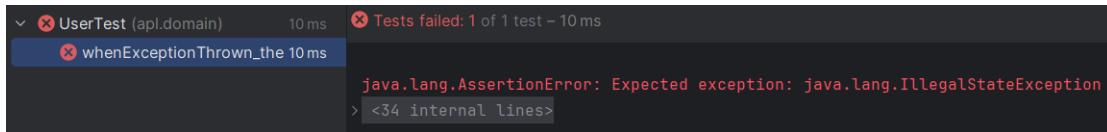
U ovom ispitivanju testiramo rad metode za provjeru je li username dobrog oblika. Testiramo na način da pošaljemo username koji je validan. Test hvata iznimku te ako je test prošao, to znači da se dogodila iznimka i username nije validan. Ako se nije dogodila iznimka, test nije prošao i username je validan. Testiramo username "12345". U svrhu testa metoda je pretvorena u public metodu, a u kodu je ostavljena kao private.

**Kôd testa:**

```

@Test(expected = IllegalStateException.class)
public void whenExceptionThrown_thenUsernameBad() {
    user = new User("12345", "Role",
HexFormat.of().parseHex("e04fd020ea3a6910a2d808002b30309d"), "password",
"name", "surname", "email");
    user.validateUsername();
    //assertEquals(1, 1);
}
  
```

**Rezultat:**



Zadan je ispravan oblik usernamea i test nije prošao što je očekivan rezultat.

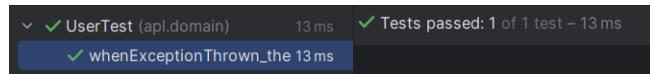
### 4. ispitni slučaj: ispitivanje metode za validaciju korisničkog imena(username) unoseći naizgled ispravan username

U ovom ispitivanju testiramo rad metode za provjeru je li username dobrog oblika. Testiramo na način da pošaljemo username koji je naizgled validan. Test hvata iznimku te ako je test prošao, to znači da se dogodila iznimka i username nije validan. Ako se nije dogodila iznimka, test nije prošao i username je validan. Testiramo username "User1##". U svrhu testa metoda je pretvorena u public metodu, a u kodu je ostavljena kao private.

**Kôd testa:**

```

@Test(expected = IllegalStateException.class)
public void whenExceptionThrown_thenUsernameBad() {
    user = new User("User1##", "Role",
HexFormat.of().parseHex("e04fd020ea3a6910a2d808002b30309d"),
"password", "name", "surname", "email");
    user.validateUsername();
    //assertEquals(1, 1);
}
  
```

**Rezultat:**

Zadan je neispravan oblik usernamea i test je prošao što je očekivan rezultat.

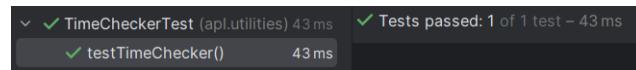
**5. ispitni slučaj: ispitivanje metode za provjeru vremena unutar intervala**

U ovom ispitivanju testiramo rad metode za provjeru je li određeni datum unutar dva zadana intervala. Test će proći ako je zadani datum unutar intervala. Testirat ćemo rubni slučaj koji ispituje nalazi li se datum 13.1.2024. unutar intervala 13.1.2024. – 24.1.2024.

**Kôd testa:**

```
@Test
void testTimeChecker() throws Exception{
    LocalDate ld1 = LocalDate.of(2024, Month.JANUARY, 13);
    LocalTime lt1 = LocalTime.of(14, 30);
    LocalDateTime ldt1 = LocalDateTime.of(ld1, lt1);
    LocalDate ld2 = LocalDate.of(2024, Month.JANUARY, 24);
    LocalTime lt2 = LocalTime.of(14, 30);
    LocalDateTime ldt2 = LocalDateTime.of(ld2, lt2);
    LocalDateTime ldt3 = LocalDateTime.now();

    assertEquals(true, tc.isWithinTimeRange(ldt1, ldt1, ldt2));
}
```

**Rezultat:**

Zadan je datum koji je na granici intervala te je test prošao što je očekivan rezultat.

**6. ispitni slučaj: ispitivanje metode za provjeru vremena unutar intervala**

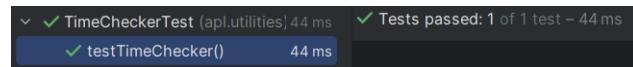
U ovom ispitivanju testiramo rad metode za provjeru je li određeni datum unutar dva zadana intervala. Test će proći ako je zadani datum unutar intervala. Testirat ćemo rubni slučaj koji ispituje nalazi li se datum 13.1.2024. unutar intervala null – null.

**Kôd testa:**

```
@Test
void testTimeChecker() throws Exception{
    LocalDate ld1 = LocalDate.of(2024, Month.JANUARY, 13);
    LocalTime lt1 = LocalTime.of(14, 30);
    LocalDateTime ldt1 = LocalDateTime.of(ld1, lt1);
    LocalDate ld2 = LocalDate.of(2024, Month.JANUARY, 24);
    LocalTime lt2 = LocalTime.of(14, 30);
    LocalDateTime ldt2 = LocalDateTime.of(ld2, lt2);
    LocalDateTime ldt3 = LocalDateTime.now();

    assertEquals(true, tc.isWithinTimeRange(ldt1, null, null));
}
```

**Rezultat:**



Zadani su intervali null i null. Test je prošao što je očekivan rezultat jer se svaki datum nalazi unutar nepostojećeg intervala.

## 5.2.2 Ispitivanje sustava

Proces ispitivanja završene i potpuno integrirane inačice namijenjene distribuciji korisniku. Ispitivanje sustava povećava razinu povjerenja prije nego što proizvod krene na ispitivanje prihvatljištva, a osnovni cilj je provjera podudarnosti sustava s funkcijskim zahtjevima.

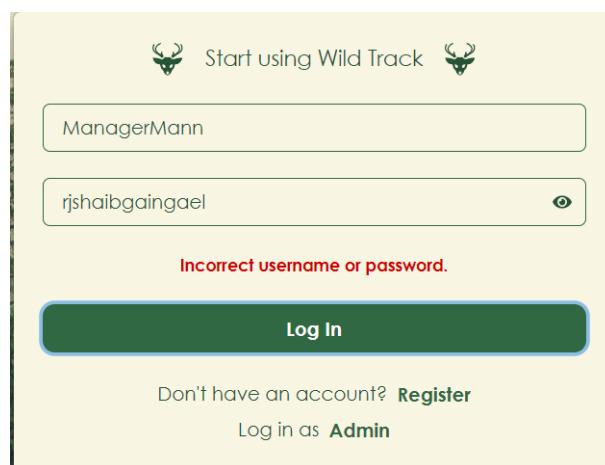
### 1. ispitni slučaj: Ispitivanje unosa pogrešne lozinke prilikom prijave

U ovom ispitivanju testiramo otpornost sustava na unos pogrešne lozinke prilikom korisnikove prijave. Očekuje se da sustav neće dopustiti prijavu te da će ispisati odgovarajuću poruku.

**Test:**

The screenshot shows a test recording interface with a list of 11 recorded steps on the left and their corresponding actions and target elements on the right. The steps include opening the browser, setting window size, clicking on input fields, typing 'ManagerMann' and 'rjshaibgaingael', and performing mouse over/out actions. The right side shows the corresponding CSS and ID mappings for each step. The interface includes command, target, value, and description fields at the bottom.

**Rezultat:**



Aplikacija ne dopušta prijavu i ispisuje se odgovarajuća poruka korisniku.

## 2. ispitni slučaj: Ispitivanje pregleda zahtjeva kod voditelja

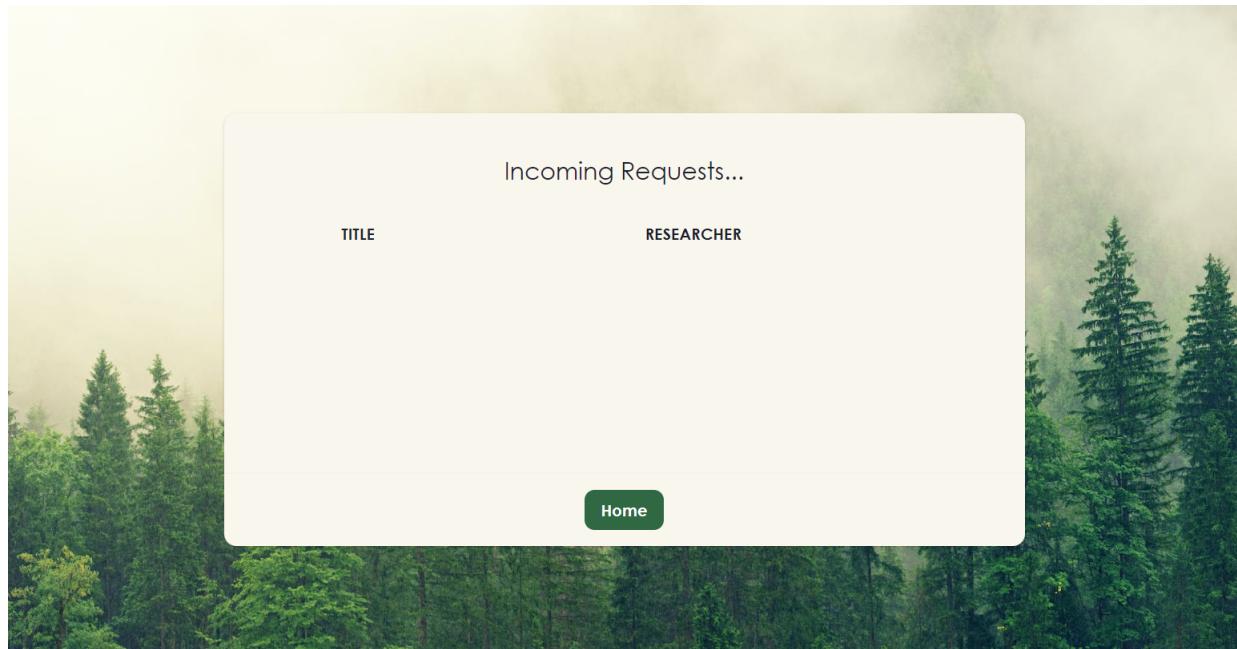
U ovom ispitivanju testiramo kako sustav odgovara ako pokušamo pregledati zahtjeve tragača. Voditelj nije primio niti jedan zahtjev od strane tragača pa očekujemo da sustav ne daje krive informacije.

Test:

	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1250x776	
3	✓ click	css= .css-mx8yt4	
4	✓ click	id=username	
5	✓ type	id=username	ManagerMann
6	✓ type	id=password	Manager#123
7	✓ click	css=.chakra-button	
8	✓ mouse over	css=.chakra-button	
9	✓ run script	window.scrollTo(0,200)	
10	✓ click	css=.css-1wkhh9b	
11	✓ run script	window.scrollTo(0,0)	

Runs: 1 Failures: 0

Rezultat:



Aplikacija ne ispisuje nikakav višak informacija i prikazuje se prazna stranica jer nema zahtjeva.

## 3. ispitni slučaj: Ispitivanje potvrde od strane admina

U ovom ispitivanju testiramo potvrđivanje korisnika od strane administratora. Očekujemo da ad-

ministrator, nakon prijave korisnika, može potvrditi korisnika.

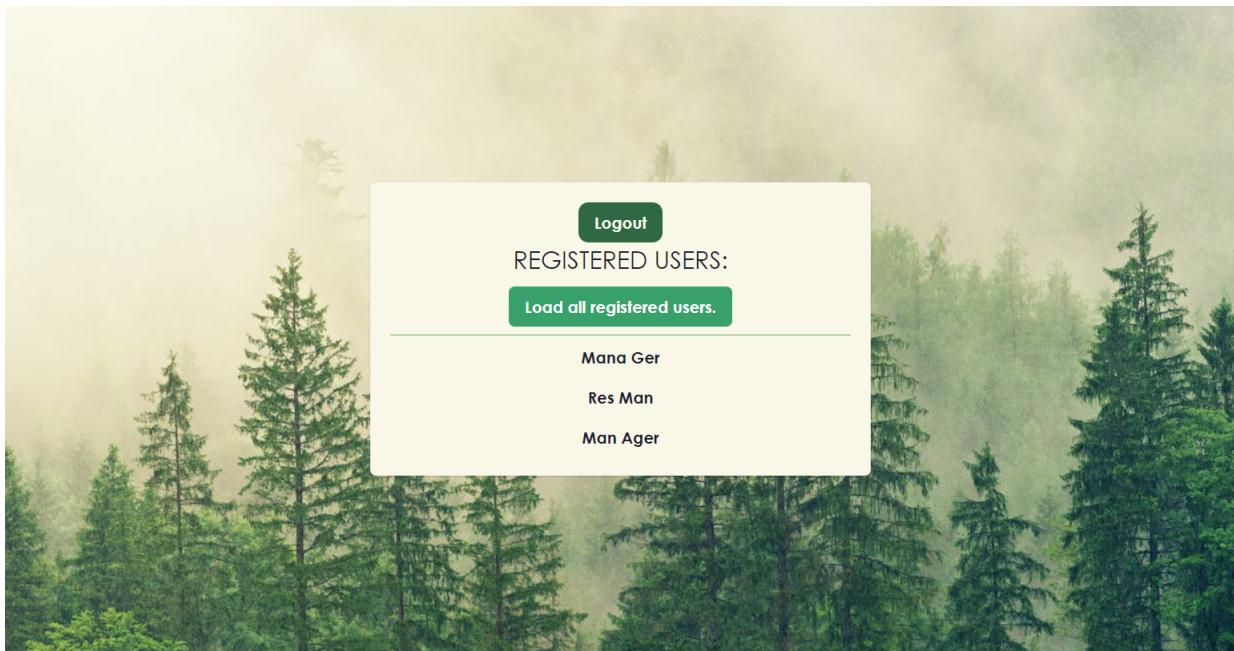
**Test:**

Step	Action	Target	Value	Notes
1	✓ open	/		
2	✓ set window size	1250x776		
3	✓ click	css= css-mx8yf4		
4	✓ click	css= chakra-text:nth-child(2) b		
5	✓ click	id=password		
6	✓ type	id=password	admin	
7	✓ click	css= chakra-input__right-element path		
8	✓ click	css= chakra-button		
9	✓ click	css= css-9857w1		
10	✓ mouse over	css= css-9857w1		
11	✓ mouse out	css= css-9857w1		
12	✓ click	css= css-m4hh83		

Command Target Value Description

Runs: 1 Failures: 0

**Rezultat:**



Administrator može potvrditi korisnike i na kraju dobiva popis svih korisnika aplikacije.

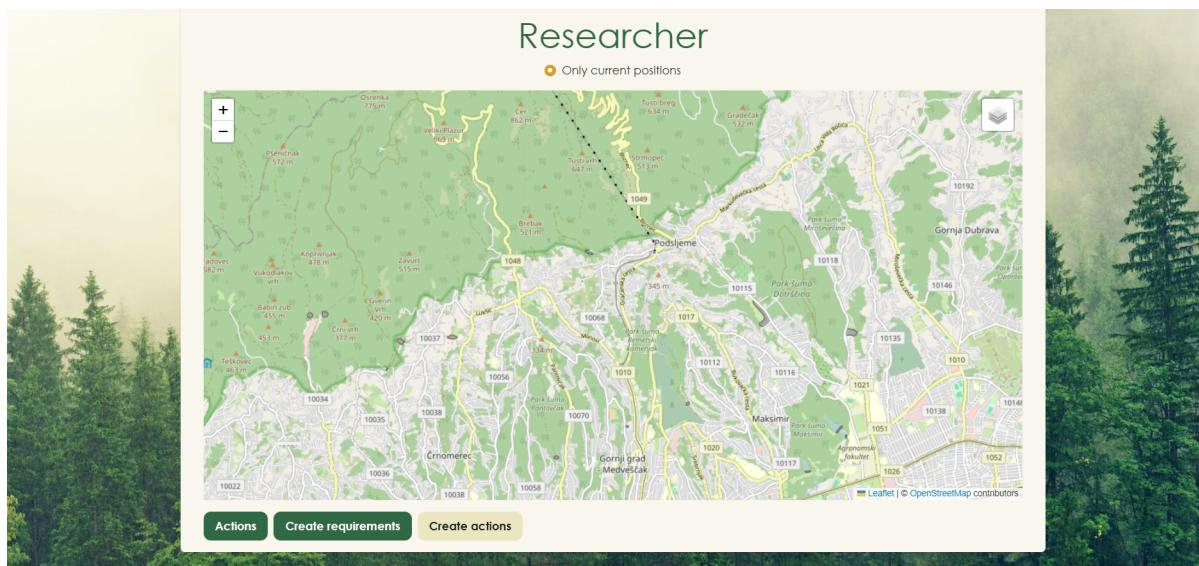
**4. ispitni slučaj: Ispitivanje unosa broja tragača od strane istraživača**

U ovom ispitivanju testiramo zahtjev istraživača za brojem tragača. Istraživač može zatražiti broj tragača za određenu vrstu pretrage. Želimo vidjeti što se dogodi ako se unese samo jedan potreban tragač, a na ostala mesta se upiše 0. Očekujemo da sustav pošalje zahtjev voditelju za jednim tragačem, a ostale vrijednosti ignorira.

**Test:**

13	✓ click	id#title	
14	✓ type	id#title	Akcia1
15	✓ click	id#id	
16	✓ select	id#id	label=Mana Get, Stanica
17	✓ click	id#amount	
18	✓ type	id#amount	1
19	✓ click	css*=css-102741#nth-child(4) > .css-0:nth-child(2) #amount	
20	✓ type	css*=css-102741#nth-child(4) > .css-0:nth-child(2) #amount	0
21	✓ click	css*=css-102741#nth-child(4) > .css-0:nth-child(3) #amount	
22	✓ type	css*=css-102741#nth-child(4) > .css-0:nth-child(3) #amount	0
23	✓ click	css*=css-102741#nth-child(5) > .css-0:nth-child(1) #amount	
24	✓ type	css*=css-102741#nth-child(5) > .css-0:nth-child(1) #amount	0
25	✓ click	css*=css-102741#nth-child(5) > .css-0:nth-child(2) #amount	
26	✓ type	css*=css-102741#nth-child(5) > .css-0:nth-child(2) #amount	0
27	✓ click	css*=css-102741#nth-child(6) > .css-0:nth-child(1) #amount	
28	✓ type	css*=css-102741#nth-child(6) > .css-0:nth-child(1) #amount	0
29	✓ click	css*=css-102741#nth-child(6) > .css-0:nth-child(2) #amount	
30	✓ type	css*=css-102741#nth-child(6) > .css-0:nth-child(2) #amount	0
31	✓ click	css*=css-102741#nth-child(6) > .css-0:nth-child(3) #amount	
32	✓ type	css*=css-102741#nth-child(6) > .css-0:nth-child(3) #amount	0
33	✓ click	css*=css-12env08	

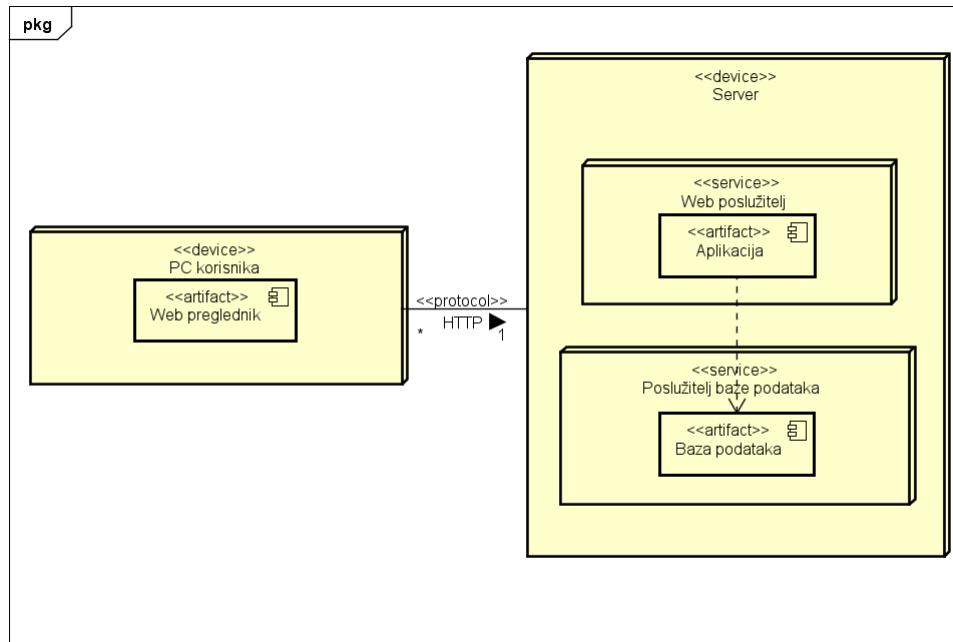
### **Rezultat:**



Istraživač uspješno šalje zahtjev za jednim tragačem i vraća se na početnu stranicu

## 5.3 Dijagram razmještaja

Dijagram razmještaja je strukturalni statički UML dijagram koji opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Na korisničkom računalu nalazi se web preglednik. Na poslužiteljskom računalu nalazi se poslužitelj web aplikacije i poslužitelj baze podataka. Klijenti preko web preglednika pomoću HTTP veze komuniciraju s poslužiteljem web aplikacije.



## 5.4 Upute za puštanje u pogon

### *dio 2. revizije*

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

*Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.*

## 6. Zaključak i budući rad

Naš je zadatak bio napraviti aplikaciju koja služi za praćenje divljih životinja te olakšava organizaciju posla istraživačima, voditeljima postaje te tragačima koji sudjeluju u takvim akcijama.

Aplikaciju smo radili jedan semestar te smo stekli iskustvo rada u timu koje je vrlo korisno za našu buduću karijeru. Koliko god smo mislili da znamo funkcionirati i raditi u timu, ovaj projekt nam je pokazao da je zapravo vrlo izazovno raditi u timu od sedam osoba te u određenom roku napraviti proizvod koji ima sve željene funkcionalnosti. Osim toga, većini tima ova je aplikacija prvi veliki projekt pa smo bogatiji i za jedno iskustvo izrade takve aplikacije.

Tijekom razvoja aplikacije, proces koji je trajao najduže bila je izrada i implementacija baze podataka. Bilo je izazovno osmisiliti bazu koja će dobro funkcionirati u našem zadatku s obzirom da se prvi put susrećemo s tako kompleksnom bazom te smatramo da je to vještina koja nam je možda nedostajala, a znatno bi ubrzala naš rad.

Osim toga, šest od sedam članova našeg tima nikada nije radilo s tehnologijama i alatima koje smo koristili pri izradi ovog projekta te smo tako svi morali vrlo ubrzano učiti i međusobno si pomagati. Tako smo zbog manjka iskustva u više situacija morali izbrisati i ponovno pisati kod za neku funkcionalnost kada bismo shvatili da je ono što smo radili netočno.

Bili smo podijeljeni u dva podtima, jedan za frontend te jedan za backend. Svaki podtim imao je voditelja koji je kontrolirao što ostali članovi rade te smatramo da je to vrlo dobro funkcioniralo.

Implementirali smo sve funkcionalnosti koje su zatražene u zadatku te smo, iako naravno ima mesta za napredak na implementaciji te dodatnim funkcionalnostima aplikacije, vrlo zadovoljni izrađenom aplikacijom s obzirom na sve nabrojane čimbenike.

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. IMammalia, <https://www.agrokub.com/poljoprivredne-vijesti/ukljucite-se-u-projekt-mammalnet-pr-55093/>
3. eWildLife, <https://play.google.com/store/apps/details?id=com.nic.app.ewildlife&hl=hr&gl=US>
4. Kwibi, <https://play.google.com/store/apps/details?id=com.bitri.kwibi&hl=hr&gl=US>
5. Astah, <https://astah.net/>
6. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
7. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
8. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
9. The Unified Modeling Language, <https://www.uml-diagrams.org/>
10. Astah Community, <http://astah.net/editions/uml-new>

# Indeks slika i dijagrama

2.1	Primjer obrasca za registraciju . . . . .	6
2.2	Primjer toplinske karte . . . . .	8
2.3	Aplikacija eWildLife . . . . .	9
2.4	Aplikacija Kwibi . . . . .	10
2.5	Aplikacija IMammalia . . . . .	10
3.1	Dijagram obrasca uporabe, funkcionalnost administratora, klijenta, korisnika i GPS uređaja . . . . .	22
3.2	Dijagram obrasca uporabe, funkcionalnost voditelja, istraživača i tragača . . . . .	23
3.3	Sekvencijski dijagram - UC6 . . . . .	24
3.4	Sekvencijski dijagram - UC7 . . . . .	25
3.5	Sekvencijski dijagram - UC11 . . . . .	26
3.6	Sekvencijski dijagram - UC15 . . . . .	27
4.1	Komunikacija između frontenda i backenda . . . . .	30
4.2	E-R dijagram baze podataka . . . . .	43
4.3	Dijagram razreda - dio Controllers . . . . .	45
4.4	Dijagram razreda - dio Data transfer objects . . . . .	46
4.5	Dijagram razreda - dio Models . . . . .	48
4.6	Dijagram stanja . . . . .	49
4.7	Dijagram aktivnosti . . . . .	50
4.8	Dijagram komponenti . . . . .	52
6.1	Contributors from GitHub . . . . .	69
6.2	Git statistics . . . . .	70

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 16. listopad 2023.
- Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, M. Bugarin, G. Oroz, N. Jamić, D. Štrbac
- Teme sastanka:
  - upoznavanje
  - određivanje voditelja tima

### 2. sastanak

- Datum: 20. listopad 2023.
- Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, M. Bugarin, G. Oroz, N. Jamić, D. Štrbac
- Teme sastanka:
  - sastanak s asistentom
  - analiza zadatka

### 3. sastanak

- Datum: 23. listopad 2023.
- Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, M. Bugarin, G. Oroz, N. Jamić, D. Štrbac
- Teme sastanka:
  - raspodjela posla
  - konačan odabir alata i tehnologija
  - rad na dokumentaciji

### 4. sastanak

- Datum: 27. listopad 2023.
- Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, M. Bugarin, G. Oroz, N. Jamić, D. Štrbac
- Teme sastanka:
  - sastanak s asistentom
  - razrješavanje dilema u vezi dokumentacije
  - daljnja rapodjela zadataka

### 5. sastanak

- Datum: 3. studeni 2023.
- Prisustvovali: M. S. Matušin, N. Jamić

- Teme sastanka:
    - rad na backendu aplikacije
6. sastanak
- Datum: 7. studeni 2023.
  - Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, M. Bugarin, G. Oroz, N. Jamić, D. Šrbac
  - Teme sastanka:
    - zajednički rad na aplikaciji
    - dogovaranje za dizajn stranice
    - daljnja raspodjela zadataka

7. sastanak
- Datum: 12. studeni 2023.
  - Prisustvovali: M. S. Matušin, N. Jamić
  - Teme sastanka:
    - rad na backendu aplikacije

8. sastanak
- Datum: 18. prosinca 2023.
  - Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, M. Bugarin, G. Oroz, D. Šrbac
  - Teme sastanka:
    - plan za drugi ciklus

9. sastanak
- Datum: 2. siječnja 2024.
  - Prisustvovali: D. Šrbac, N. Jamić
  - Teme sastanka:
    - izrada potpune baze podataka u kodu

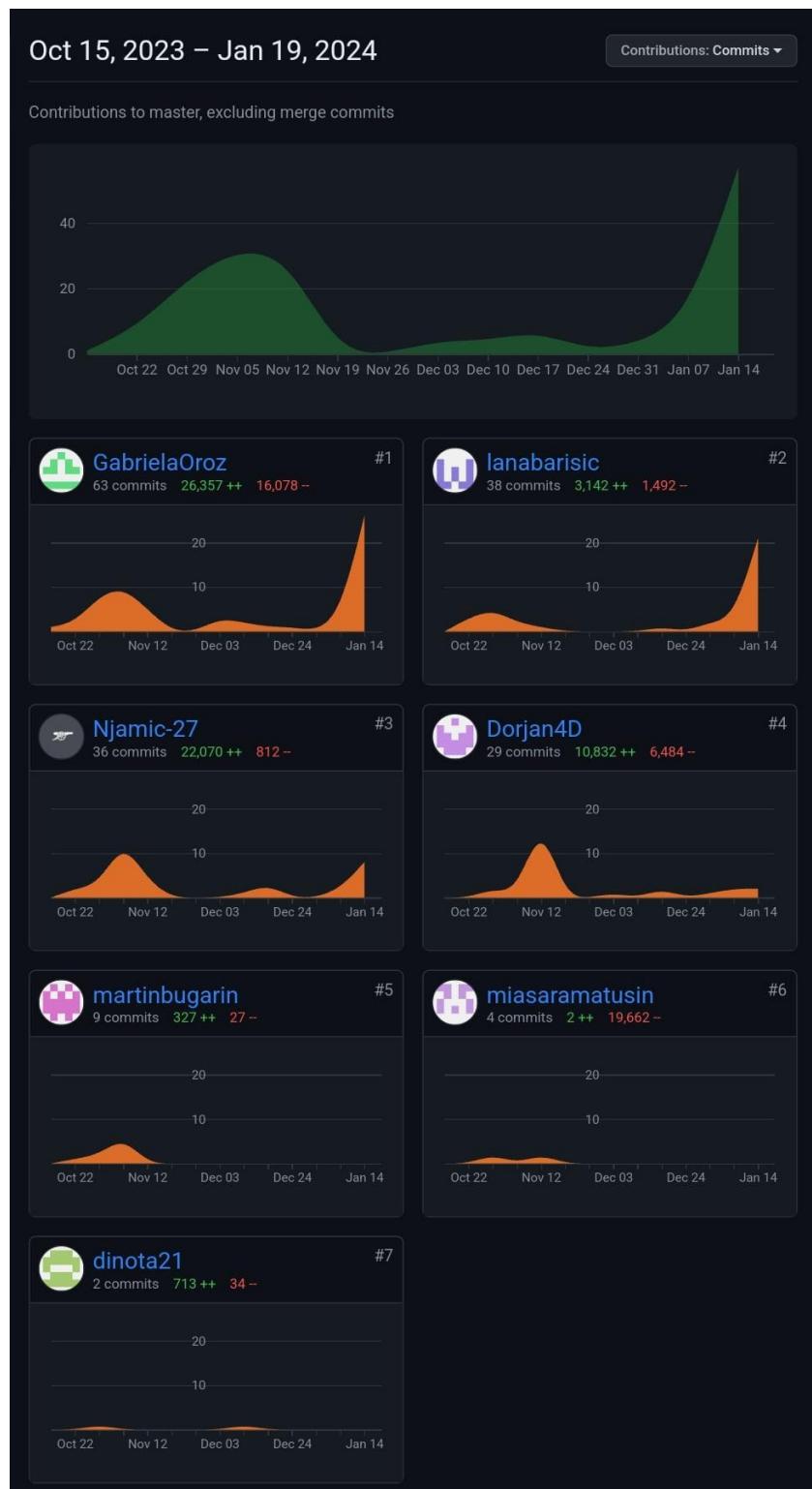
10. sastanak
- Datum: 15. siječnja 2024.
  - Prisustvovali: N. Jamić, L. Barišić, M. S. Matušin
  - Teme sastanka:
    - usklađivanje podataka na frontu i backu

11. sastanak
- Datum: 18. siječnja 2024.
  - Prisustvovali: M. S. Matušin, D. Baralić, L. Barišić, G. Oroz, N. Jamić, D. Šrbac
  - Teme sastanka:
    - dogovor za prezentaciju

## Tablica aktivnosti

	Mia Sara Matušin	Dino Baralić	Lana Barišić	Martin Bugarin	Nikola Jamić	Gabriela Oroz	Dorjan Štrbac
Upravljanje projektom	10						
Opis projektnog zadatka	5		1				
Funkcionalni zahtjevi			1		3		
Opis pojedinih obrazaca			8			5	
Dijagram obrazaca			2				
Sekvencijski dijagrami				4			
Opis ostalih zahtjeva				1			
Arhitektura i dizajn sustava			4				
Baza podataka		10					10
Dijagram razreda		12					12
Dijagram stanja				7			
Dijagram aktivnosti				7			
Dijagram komponenti	10						
Korištene tehnologije i alati				2			
Ispitivanje programskog rješenja	5				5		5
Dijagram razmještaja				2			
Upute za puštanje u pogon		5					
Dnevnik sastajanja	2						
Zaključak i budući rad	2						
Popis literature	1						
Front end			70	21		70	
Back end	80				80		50

## Dijagrami pregleda promjena



Slika 6.1: Contributors from GitHub

```
PS C:\Users\TUF\Desktop\5. semestar\PROGI\lovciNa> git ls-files |ForEach-Object { git blame --line-porcelain $_ } | Select-String '^author ' | Group-Object | Sort-Object Count -Descending | Select-Object Count, Name

Count Name
----- -----
15403 author Mia Sara
14098 author Gabriela Oroz
7504 author Dorjan Járbac
3443 author Martin Bugarin
2083 author Lana Barisic
792 author Nikola
630 author Dino Baralić

PS C:\Users\TUF\Desktop\5. semestar\PROGI\lovciNa> git log --format='%aN' | Sort-Object | Group-Object | Select-Object Count, Name | Sort-Object Count -Descending

Count Name
----- -----
72 Gabriela Oroz
70 Mia Sara
51 Lana Barisic
40 Nikola
32 Dorjan Járbac
11 Martin Bugarin
2 Dino Baralić
1 Nikola Jamić
1 marshmallow47
```

Slika 6.2: Git statistics