

Programsko inženjerstvo

Ak. god. 2023./2024.

WildTrack

Dokumentacija, Rev. 2.

Grupa: *Pokret nesvrstanih*

Voditelj: *Emil Prpić*

Datum predaje: 19. 1. 2024.

Nastavnik: *Hrvoje Nuić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	6
3 Specifikacija programske potpore	11
3.1 Funkcionalni zahtjevi	11
3.1.1 Obrasci uporabe	13
3.1.2 Sekvencijski dijagrami	27
3.2 Ostali zahtjevi	31
4 Arhitektura i dizajn sustava	32
4.1 Baza podataka	34
4.1.1 Opis tablica	35
4.1.2 Dijagram baze podataka	42
4.2 Dijagram razreda	43
4.3 Dijagram stanja	45
4.4 Dijagram aktivnosti	46
4.5 Dijagram komponenti	47
5 Implementacija i korisničko sučelje	48
5.1 Korištene tehnologije i alati	48
5.2 Ispitivanje programskog rješenja	49
5.2.1 Ispitivanje komponenti	49
5.2.2 Ispitivanje sustava	53
5.3 Dijagram razmještaja	56
5.4 Upute za puštanje u pogon	57
6 Zaključak i budući rad	62
Popis literature	64
Indeks slika i dijagonama	66

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Sebastian Medja- ković	26.10.2023.
0.2	Opis projekta i funkcionalni zahtjevi.	Sara Gašpar	30.10.2023.
0.3	Nabrojeni obrasci uporabe	Sebastian Medja- ković	1.11.2023.
0.4	Opis obrazaca uporabe	svi	1.11.2023.
0.5	Dodan model baze i nefunkcionalni zahtjevi	Sebastian Medja- ković	2.11.2023.
0.6	Arhitektura i dizajn sustava	Sara Gašpar	2.11.2023.
0.7	Revizija dokumentacije	Sara Gašpar, Sebastian Medja- ković	2.11.2023.
0.8	Dijagrami obrazaca uporabe	Jure Spajić, Filip Vitković	06.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
0.9	Sekvencijski dijagrami	Ivan Cvjetić, Filip Vitković	06.11.2023.
0.10	Opis baze podataka	Sebastian Medjaković	07.11.2023.
0.11	Revizija dokumentacije	Jure Spajić	13.11.2023.
0.12	Dijagrami razreda	Ivan Cvjetić, Emil Prpić, Jure Spajić	15.11.2023.
1.0	Dokumentacija prije prve predaje	Ivan Lisića, Sara Gašpar, Jure Spajić	17.11.2023.
1.1	Ispravak dokumentacije prvog ciklusa	Jure Spajić	2.1.2024.
1.2	Dijagram stanja i aktivnosti	Jure Spajić	8.1.2024.
1.3	Dijagram komponenti i razmještaja	Jure Spajić	11.1.2024.
1.4	Zaključak, korištena tehnologija i literatura	Jure Spajić	12.1.2024.
1.5	Upute za puštanje u pogon	Jure Spajić, Emil Prpić	13.1.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

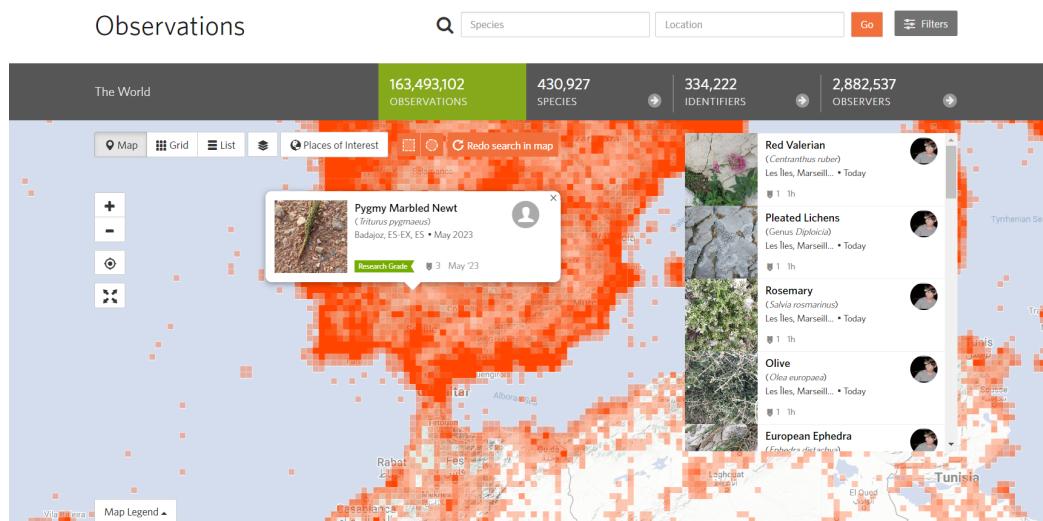
Rev.	Opis promjene/dodataka	Autori	Datum
1.6	Ispitivanje programskog rješenja	Jure Spajić, Sebastian Medja- ković	14.1.2024.
1.7	Dijagrami pregleda promjena	Jure Spajić	16.1.2024.
2.0	Dokumentacija prije druge predaje	Jure Spajić	18.1.2024.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "Wildtrack" koja će korisnicima olakšati koordinaciju prilikom prolazeњa i praćenja divljih životinja. Ovom aplikacijom želimo unaprijediti istraživačke projekte i potaknuti svijest o važnosti očuvanja divljih životinja. Planiramo stvoriti jednostavno korisničko sučelje koje će omogućiti korisnicima da zajedno doprinesu istraživanju i zaštiti divljih životinja. Ova web aplikacija će biti izuzetno korisna za praćenje kretanja životinja, analiziranje njihovih navika i migracije te pružanje važnih podataka za očuvanje njihovih prirodnih staništa.

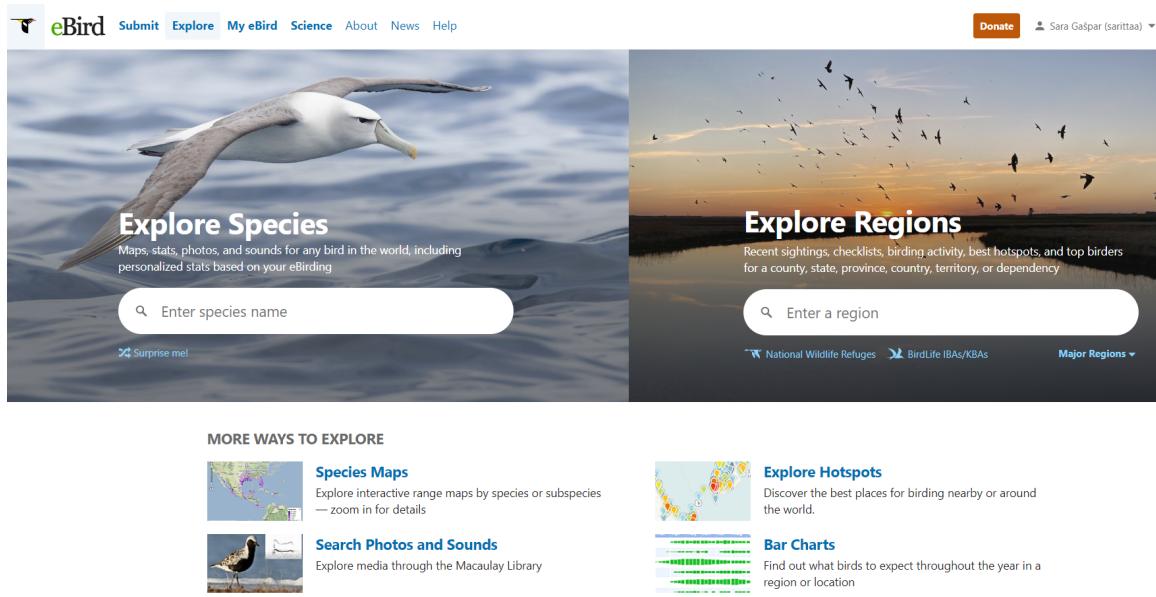
Na tržištu već postoje neke slične aplikacije koje omogućavaju praćenje divljih životinja, ali Wildtrack se ističe svojom inovativnom platformom za suradnju između različitih korisničkih uloga. Neke od postojećih aplikacija su:

- iNaturalist: iNaturalist je globalna mreža istraživača, znanstvenika i ljubitelja prirode koji dijele svoja opažanja divljih životinja. Korisnici mogu fotografirati i dijeliti slike biljaka i životinja, a zajednica pomaže u identifikaciji vrsta.



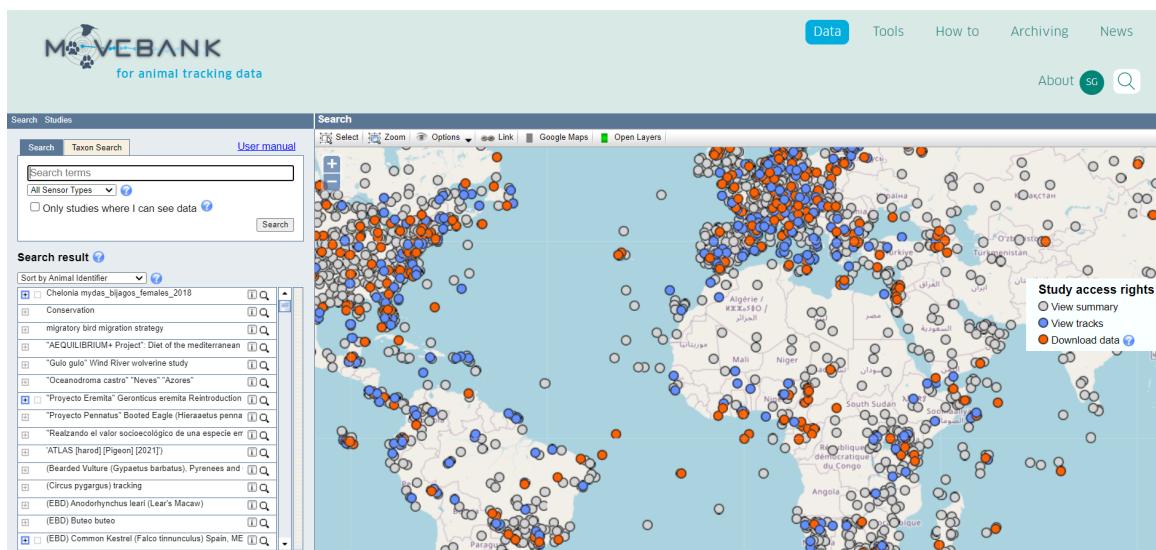
Slika 2.1: iNaturalist

- eBird: eBird je aplikacija razvijena od strane Cornell Lab of Ornithology, fokusirana na ptice. Korisnici mogu bilježiti svoja opažanja o pticama te pridonositi globalnoj bazi podataka o pticama.



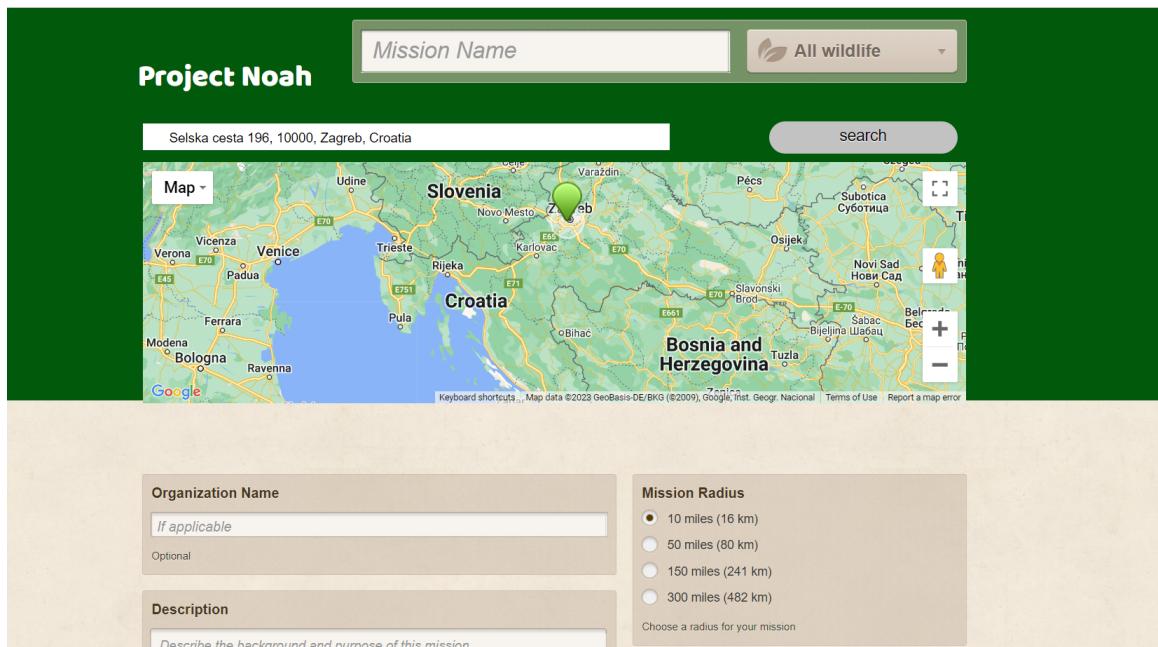
Slika 2.2: eBird

- Movebank: Ova platforma omogućuje istraživačima da prate ponašanje divljih životinja putem GPS uređaja, odašiljača i drugih senzora, te analiziraju ove podatke u stvarnom vremenu.



Slika 2.3: Movebank

- Project Noah: Ova aplikacija omogućava korisnicima da dijele fotografije divljih životinja i biljaka te surađuju s globalnom zajednicom kako bi identificirali vrste.



Slika 2.4: Project Noah

Svaka od ovih aplikacija ima svoje specifične značajke i usmjerena je na različite vrste divljih životinja ili na različite vrste istraživanja. Wildtrack bi mogao kombinirati neke od ovih značajki, kreiranje akcija poput Project Noah-a te praćenje životinja poput Movebank-a pružajući jedinstvenu platformu koja omogućava suradnju između različitih korisničkih uloga i pruža specifične alate za istraživače, voditelje postaja i terenske tragače. U nastavku ćemo detaljnije istražiti ključne značajke i prednosti ove inovativne web aplikacije.

Prilikom pokretanja aplikacije, neregistriranom korisniku otvara se ekran na kojem ima dvije mogućnosti: prijaviti se u sustav već postojećim računom koristeći korisničko ime i lozinku ili stvoriti novi račun registracijom. Prilikom stvaranju novog računa, potrebno je unijeti sljedeće podatke: korisničko ime, fotografiju, lozinku, ime, prezime te email adresu. Registrirati se može kao jedan od tri različite vrste korisnika: voditelj postaje, istraživač te tragač. Registracija se završava potvrdom

preko email adrese, a istraživača i voditelja postaje dodatno treba potvrditi administrator. Tek kada je korisnik registriran i prijavljen u sustav, može iskusiti sve funkcionalnosti naše web aplikacije.

Praćene životinje na sebi imaju GPS uređaj koji aplikaciji odašilje svoju poziciju. O praćenim životinjama se zapisuju povijesni podaci gdje se nalazila, naziv vrste, slika i opis. Korisnicima se te informacije prikazuju na karti.

U aplikaciji postoje četiri različite vrste korisnika: tragač, istražitelj, voditelj postaje, te administrator. Svaka od ovih uloga ima specifične zadatke i odgovornosti. Tragači koriste GPS uređaje za praćenje životinja i obavljanje zadataka koje im dodijele istraživači. Istraživači su kreatori akcija praćenja, određuju vrste i lokacije praćenja te dodjeljuju zadatke tragačima. Voditelji postaje su odgovorni za određeno geografsko područje i organizaciju tragača unutar svoje postaje. U nastavku ćemo opisati mogućnosti i ovlasti svake od vrsta korisnika.

Voditelj postaje ima mogućnost na karti odabratи jednu od ponuđenih postaja koja obuhvaća određeni prostor i ima svoje ime, npr. postaja Lonjsko polje ili Biokovo. Kada je odabrao svoju postaju, ima mogućnost odabratи tragače svoje postaje. Osim toga, voditelj definira na koji način su njegovi tragači sposobljeni izvoditi pretraživanje. Mogu biti sposobljeni za izvođenje zadataka pješke, dronom, automobilom, cross motorom, brodom ili helikopterom. Svaka metoda pretraživanja pruža različitu vidljivost i područje pokrivanja te se na prikidan način prikazuje na karti. Voditelj također može vidjeti prikaz karte sa svojim tragačima.

Istraživač je zadužen za kreiranje novih akcija pretraživanja i praćenja s detaljima o određenim vrstama, jedinkama ili staništima za proučavanje. Svaki istraživač zadužen je samo za jednu akciju. Prilikom stvaranja akcije, istraživač voditelju stanice šalje zahtjev za tragačima. U zahtjevu treba napisati broj ljudi koji mu je potreban za akciju, te zadatke koje je potrebno obaviti. Svaka akcija ima jednog istraživača i više tragača, a odnosi se na postaju jednog ili više voditelja. Ako mu voditelj ili voditelji odobre zahtjev (imaju dovoljan broj kvalificiranih ljudi) šalju mu popis dodijeljenih tragača i tada im istraživač mora pojedinačno podijeliti konkretnе zadatke. Zadatak može biti prolazak određenom rutom, dolazak do lokacije, postavljanje kamere, te postavljanje GPS uređaja za praćenje na

životinje. U slučaju da voditelj u trenutku slanja zahtjeva nema dovoljan broj odgovarajućih tragača, istraživaču će zahtjev biti odbijen, te ima mogućnost poslati novi. Istraživaču se na njegovom početnom ekranu prikazuje interaktivna karta s informacijama o pozicijama životinja, tragača i postaja. Istraživač može odabrati da se za izradu karata koriste neka od idućih informacija: povjesne pozicije svih praćenih životinja, filtrirano po vrsti ili pojedinačno po jedinki te trenutne pozicije praćenih životinja; povjesne pozicije svih tragača na nekoj akciji, filtrirano po tipu prijevoza ili pojedinačno po tragaču te trenutne pozicije tragača aktivnih na akciji.

Tragač ima ulogu praćenja životinja i obavljanja zadataka koje mu je dodijelio istraživač. Njegov početni ekran nakon prijave u sustav sadrži kartu na kojoj su mu prikazani zadaci koje treba obaviti, trenutna pozicija ostalih tragača aktivnih na istoj akciji, te trenutna pozicija praćenih životinja. Tragač na jednoj akciji ne mijenja tip prijevoza. Tragač se može maknuti s akcije završetkom svih potrebnih zadataka, a staze kojima prolazi zapisuju se u bazu.

Administrator je uloga s najvećim ovlastima i ima ju samo jedna osoba. Administrator mora potvrditi registraciju ukoliko se radi o istraživačima i voditeljima postaja. Osim toga, administrator može vidjeti popis svih registriranih korisnika i njihovih osobnih podataka te im mijenjati dodijeljena prava i osobne podatke.

Wildtrack također pruža mogućnost suradnje među korisnicima putem komentara i povratnih informacija. Tragači mogu ostavljati komentare o praćenju životinja, dijeliti svoja iskustva i surađivati u stvarnom vremenu. Istraživači mogu pratiti napredak svojih akcija, analizirati podatke koje prikupljaju tragači te prilagoditi strategije praćenja. Svaka akcija praćenja životinja u Wildtrack aplikaciji pridonosi bazi podataka o divljim životnjama. Ovi podaci služe istraživačima i znanstvenicima za dublje razumijevanje migracija, ponašanja i ekologije različitih vrsta. Wildtrack je više od aplikacije - to je platforma koja potiče zajednički rad i edukaciju o važnosti divljih životinja u našem ekosustavu. Ova inicijativa ima potencijal potaknuti globalnu suradnju u zaštiti prirode i očuvanju biološke raznolikosti našeg planeta.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Voditelj postaje
2. Istraživač
3. Tragač
4. Administrator
5. Neregistrirani korisnik
6. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Voditelj postaje može:

- (a) vidjeti kartu s ponuđenim postajama
- (b) odabrati postaju čiji voditelj želi postati
- (c) odabrati tragače svoje postaje
- (d) definirati način prijevoza svojih tragača - pješke, dronom, automobilom, cross motorom, brodom ili helikopterom
- (e) prihvati ili odbiti zahtjev za tragačima od istraživača

2. Istraživač može:

- (a) kreirati nove akcije pretraživanja i praćenja životinja
- (b) poslati zahtjev za tragačima voditelju postaje
- (c) podijeliti zadatke tragačima koje je dobio od voditelja postaje
- (d) vidjeti interaktivnu kartu s podacima o pozicijama životinja, tragača i postaja
- (e) odabrati kriterije filtriranja karte
- (f) ostaviti komentar tragačima na akciji

3. Tragač može:

- (a) na karti vidjeti svoje zadatke, životinje te ostale tragače na istoj akciji
- (b) Obavljati zadatke koje mu je dodijelio istraživač
- (c) Prilikom obavljanja akcije ostaviti komentar za ostale tragače

4. Administrator može:

- (a) Pregledati popis svih registriranih korisnika i njihovih podataka
- (b) Mijenjati osobne podatke i dodijeljena prava svim korisnicima
- (c) Potvrditi ili odbiti registraciju istraživača i voditelja postaje

5. Neregistrirani korisnik može:

- (a) Poslati zahtjev za registraciju

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik popunjava korisničke podatke
 3. Korisnik prima e-mail za verifikaciju korisničkog računa
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena ili e-mail adrese, unos podataka u pogrešnom formatu, nepotpun unos podataka
 1. Sustav obavještava korisnika o neuspjeloj registraciji i upućuje na pogrešna polja
 2. Korisnik mijenja ili nadopunjava potrebne podatke te završava unos ili odustaje od registracije

UC2 - Prijava u sustav

- **Glavni sudionik:** Korisnik, administrator
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Potvrđena registracija
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke
 2. Potvrda o ispravnosti unesenih podataka
 3. Pristup korisničkom sučelju
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime ili lozinka
 1. Sustav obavještava korisnika o neispravnom unosu podataka i traži korisnika da ponovi unos ili da se registrira
 2. Korisnik unosi ispravne podatke te završava unos ili odustaje od prijave

UC3 - Pregled osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prikaz osobnih podataka"
 2. Sustav prikazuje osobne podatke korisnika

UC4 - Promjena osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prikaz osobnih podataka"
 2. Korisnik odabire opciju "Ažuriraj osobne podatke"
 3. Korisnik mijenja svoje osobne podatke
 4. Korisnik spremi promjene
 5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 3.a Unos podataka u pogrešnom formatu, brisanje podataka iz obaveznih polja
 1. Sustav obavještava korisnika o neispravnom unosu podataka i traži korisnika da ponovi unos
 2. Korisnik unosi ispravne podatke ili odustaje od izmjena

UC5 - Brisanje korisničkog računa

- **Glavni sudionik:** Korisnik
- **Cilj:** Izbrisati svoj korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prikaz osobnih podataka"
 2. Korisnik odabire opciju "Izbriši korisnički račun"

3. Korisnički račun se izbriše iz baze podataka
4. Otvara se početna stranica

UC6 - Potvrda korisnika mailom

- **Glavni sudionik:** Korisnik
- **Cilj:** Potvrditi korisnički račun e-mailom
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je upisao podatke potrebne za registraciju
- **Opis osnovnog tijeka:**
 1. Korisnik prima e-mail za verifikaciju korisničkog računa
 2. Korisnik otvara primljeni e-mail i klikne na link u poruci
 3. Link ga vodi na ekran s porukom o uspješnosti registracije
 4. Podaci o korisniku se spremaju u bazu podataka registriranih korisnika
 5. Korisnik se može prijaviti u sustav sa izabranim korisničkim imenom i lozinkom
- **Opis mogućih odstupanja:**
 - 1.a Korisnik nije dobio e-mail zbog serverske pogreške
 1. Korisnik klikne na opciju ponovno pošalji e-mail
 - 1.b Korisnik nije dobio e-mail zbog unosa krive e-mail adrese
 1. Korisnik klikne na opciju za ponovno unošenje e-mail adrese
 2. Korisnik unosi novu e-mail adresu

UC7 - Potvrda uloge od strane administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi registraciju voditelja postaje ili istraživača
- **Sudionici:** Istraživač, voditelj postaje, baza podataka
- **Preduvjet:** Istraživač ili voditelj su unijeli svoje podatke za registraciju
- **Opis osnovnog tijeka:**
 1. Administrator se prijavljuje u sustav
 2. Administrator vidi popis neriješenih registracija
 3. Administrator otvara jednu od registracija i pregleda
 4. Administrator prihvati ili odbije registraciju
 5. Istraživač ili voditelj dobiju obavijest o prihvaćenoj/odbijenoj registraciji

UC8 - Prikazivanje pozicije životinje

- **Glavni sudionici:** Administrator, istraživač i tragač
- **Cilj:** Prikaz pozicije životinje na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedna životinja s GPS uređajem
- **Opis osnovnog tijeka:**
 1. Svakih 5 sekundi se šalje lokacija životinje i sprema u bazu
 2. Prikaz karte sa lokacijama životinja u bazi
 3. Lokacija se prikazuje u obliku geom. dužine, širine i timestampa
 4. Mogućnost filtriranja karte po vrsti životinje

UC9 - Prikazivanje pozicije tragača

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz pozicije tragača na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedan tragač s GPS uređajem
- **Opis osnovnog tijeka:**
 1. Korisnik otvara opciju "Karta"
 2. Korisnik filtrira kartu tako da pokazuje tragače

UC10 - Prikazivanje pozicije postaje

- **Glavni sudionik:** Korisnik
- **Cilj:** Na karti pokazati geografsku poziciju postaje
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem jedna postaja
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Postaje"
 2. Aplikacija pokazuje kartu sa svim postajama
 3. Korisnik u polje predodređeno za tekst upisuje ime postaje čiju geografsku poziciju želi detaljnije pogledati

UC11 - Prikaz informacija o životinji

- **Glavni sudionik:** Istraživač, tragač
- **Cilj:** Prikaz informacija o pojedinoj životinji (povijest kretanja i trenutna lokacija, opis jedinke, slika i naziv vrste)

- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Istraživač/tragaču su na karti prikazane GPS lokacije životinja
 2. Istraživač/tragač klikne na određenu životinju/GPS lokaciju
 3. Otvaraju se podaci o životinji

UC12 - Odabir postaje

- **Glavni sudionik:** Voditelj
- **Cilj:** Odabir slobodne postaje
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je registriran i potvrđen
- **Opis osnovnog tijeka:**
 1. Nakon registracije voditelju se prikazuje karta na kojoj su označene postaje bez voditelja
 2. Voditelj proizvoljno bira postaju
 3. Promjene se spremaju u bazu
- **Opis mogućih odstupanja:**
 - 2.a Sve postaje već imaju voditelja
 1. Voditelj mora pričekati da neka postaja ostane bez voditelja ili da se nova postaja otvorí
 - 2.b Odabrana postaja nije slobodna
 1. Voditelj bira neku drugu postaju

UC13 -Definiranje kompetencija tragača

- **Glavni sudionik:** Voditelj
- **Cilj:** Odabir tragača unutar jedne postaje te definiranje kompetencija pojedinog tragača
- **Sudionici:** Baza podataka, tragač
- **Preduvjet:** Voditelj je odabrao postaju
- **Opis osnovnog tijeka:**
 1. Nakon odabira postaje voditelju se prikazuje popis svih tragača koji nisu raspoređeni ni u jednoj postaji
 2. Voditelj klikom odabire tragače i njihove kompetencije
 3. Tragaču i u bazi se ažuriraju promjene

- **Opis mogućih odstupanja:**
 - 1.a Ne postoji slobodan tragač
 1. Voditelj čeka da se registrira novi tragač ili da tragač bude otpušten iz postaje kojoj je dodijeljen

UC14 - Uređivanje kompetencija tragača

- **Glavni sudionik:** Voditelj
- **Cilj:** Izmjena kompetencija tragača
- **Sudionici:** Baza podataka, tragač
- **Preduvjet:** Postoji barem jedan tragač u postaji voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj odabire jednog tragača
 2. Voditelj dodaje nove ili briše postojeće kompetencije odabranog tragača
 3. Unešene promjene se spremaju u bazu podataka
 4. Voditelju i tragaču se prikazuju ažurirane kompetencije

UC15 - Brisanje tragača

- **Glavni sudionik:** Voditelj
- **Cilj:** Trajno brisanje tragača iz postaje
- **Sudionici:** Baza podataka, voditelj
- **Preduvjet:** Tragač je dio voditeljeve postaje
- **Opis osnovnog tijeka:**
 1. Voditelj odabire određenog tragača
 2. Voditelj ga briše iz svoje postaje
 3. Unešena promjena se sprema u bazu podataka
 4. Tragač dobiva informaciju o voditeljevoj odluci
 5. Tragač postaje dostupan ostalim voditeljima i može biti izabran za drugu postaju

UC16 - Pregled popisa tragača

- **Glavni sudionik:** Voditelj
- **Cilj:** Pregled svih tragača u njegovoј postaji
- **Sudionici:** Baza podataka, voditelj
- **Preduvjet:** Tragač je dio voditeljeve postaje
- **Opis osnovnog tijeka:**
 1. Voditelj odabire prikaz tragača svoje postaje
 2. Voditelju se prikaže popis njegovih tragača
 3. Voditelj ima mogućnost brisati i uređivati podatke svojih tragača

UC17 - Pregled zahtjeva

- **Glavni sudionik:** Voditelj
- **Cilj:** Započeti komunikaciju između voditelja i istraživača
- **Sudionici:** Voditelj, istraživač
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Istraživač voditelju šalje zahtjev za tragačima
 2. U zahtjevu je zapisan broj potrebnih ljudi i zadaci u akciji
 3. Voditelj dobiva zahtjev
 4. Voditelj može pregledati sve svoje zahtjeve
 5. Voditelj može odabrati zahtjev koji želi obraditi

UC18 - Obrada zahtjeva

- **Glavni sudionik:** Voditelj
- **Cilj:** Prihvatići/odbaciti zahtjev
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i poslan je barem jedan zahtjev od strane istraživača
- **Opis osnovnog tijeka:**
 1. Voditelj odabire opciju "Obrada zahtjeva"
 2. Voditelju se prikazuje karta na kojoj se nalaze samo tragači koji pripadaju njegovoј postaji
 3. Voditelju se prikazuje broj tragača potrebnih za akciju te popis zadataka akcije
 4. Voditelj s obzirom na zadatke odabire kvalificirane tragače

5. Ako ima dovoljan broj kvalificiranih tragača, voditelj šalje popis istraživaču od kojeg je dobio zahtjev, a u suprotnom odbija zahtjev

UC19 - Kreiranje akcije

- **Glavni sudionik:** Istraživač
- **Cilj:** Kreiranje nove akcije
- **Sudionici:** Baza podataka, tragači
- **Preduvjet:** Istraživač je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Istraživač odabire opciju "Nova akcija"
 2. Istraživač odabire broj tragača, potrebne kvalifikacije tragača te upisuje zadatke

UC20 - Slanje zahtjeva voditelju

- **Glavni sudionik:** Istraživač, voditelj
- **Cilj:** Poslati zahtjev voditelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je kreirao novu akciju.
- **Opis osnovnog tijeka:**
 1. Istraživač šalje voditelju/voditeljima zahtjev za tragačima s napisanim zadacima

UC21 - Pregled stanja zahtjeva

- **Glavni sudionik:** Istraživač
- **Cilj:** Uvid u stanje zahtjeva
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je prijavljen u sustav te je poslan barem jedan zahtjev
- **Opis osnovnog tijeka:**
 1. Istraživač odabire opciju "Određeni zahtjev" (npr. "Postavljanje GPS-uređaja na zeca")
 2. Istraživaču se prikazuje stanje "Određenog zahtjeva", je li zahtjev odočaren, odbijen ili je u stanju čekanja

UC22 - Kreiranje zadataka

- **Glavni sudionik:** Istraživač
- **Cilj:** Kreirati zadatke za tragače
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je u postupku kreiranja akcije
- **Opis osnovnog tijeka:**
 1. Ispod podnaslova zadaci istraživač klikne plus
 2. Istraživač definira zadatak

UC23 - Dodjela zadataka

- **Glavni sudionik:** Istraživač
- **Cilj:** Dodijeliti zadatke pojedinim tragačima
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je prijavljen u sustav i voditelj je prihvatio zahtjev za akcijom
- **Opis osnovnog tijeka:**
 1. Istraživaču se prikaže lista zadataka u akciji
 2. Istraživaču se prikaže lista tragača kompatibilnih za određene zadatke
 3. Istraživač odabire koji tragač će obaviti pojedine zadatke

UC24 - Definiranje prikaza na karti

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikazati željene podatke na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je potvrđen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire predmet filtracije podataka na karti

UC25 - Postavljanje komentara na karti

- **Glavni sudionik:** Tragač, istraživač, baza podataka
- **Cilj:** Informiranje i komunikacija istraživača i/ili tragača na akciji
- **Sudionici:** Istraživač, tragač
- **Preduvjet:** Akcija je u tijeku
- **Opis osnovnog tijeka:**
 1. Istraživač/Tragač označava poziciju na karti

2. Istraživač/Tragač zapisuje komentar

UC26 - Pregled komentara

- **Glavni sudionik:** Istraživač, tragač
- **Cilj:** Pregledati komentare
- **Sudionici:** Baza podataka
- **Preduvjet:** Mora postojati barem jedan komentar
- **Opis osnovnog tijeka:**
 1. Istraživač/Tragač filtrira kartu kako bi pokazivala komentare
 2. Odabire komentar koji želi pročitati

UC27 - Brisanje komentara

- **Glavni sudionik:** Istraživač, tragač
- **Cilj:** Obrisati objavljeni komentar
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje komentara
- **Opis osnovnog tijeka:**
 1. Glavni sudionik odabire komentar
 2. Glavni sudionik odabire opciju "Obriši komentar"
 3. Komentar se briše iz baze podataka
 4. Komentar se više ne prikazuje istraživačima i tragačima

UC28 - Prikaz karte s tragačima na akciji

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled lokacija tragača na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen korisnik
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz karte s tragačima
 2. Aplikacija prikazuje kartu s lokacijama tragača na akciji

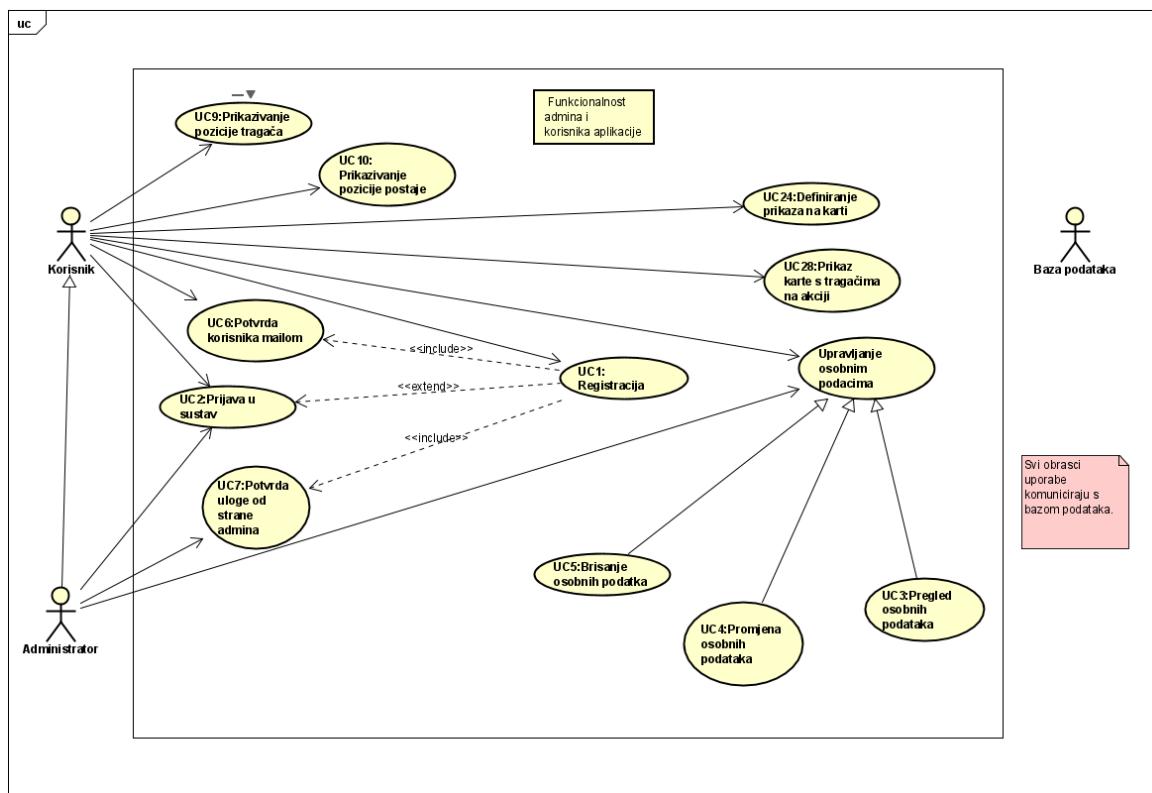
UC29 - Pregled zadatka

- **Glavni sudionik:** Istraživač, tragač
- **Cilj:** Pregled detalja pojedinog zadatka
- **Sudionici:** Baza podataka
- **Preduvjet:** Istraživač je kreirao zadatak i dodijelio ga tragaču

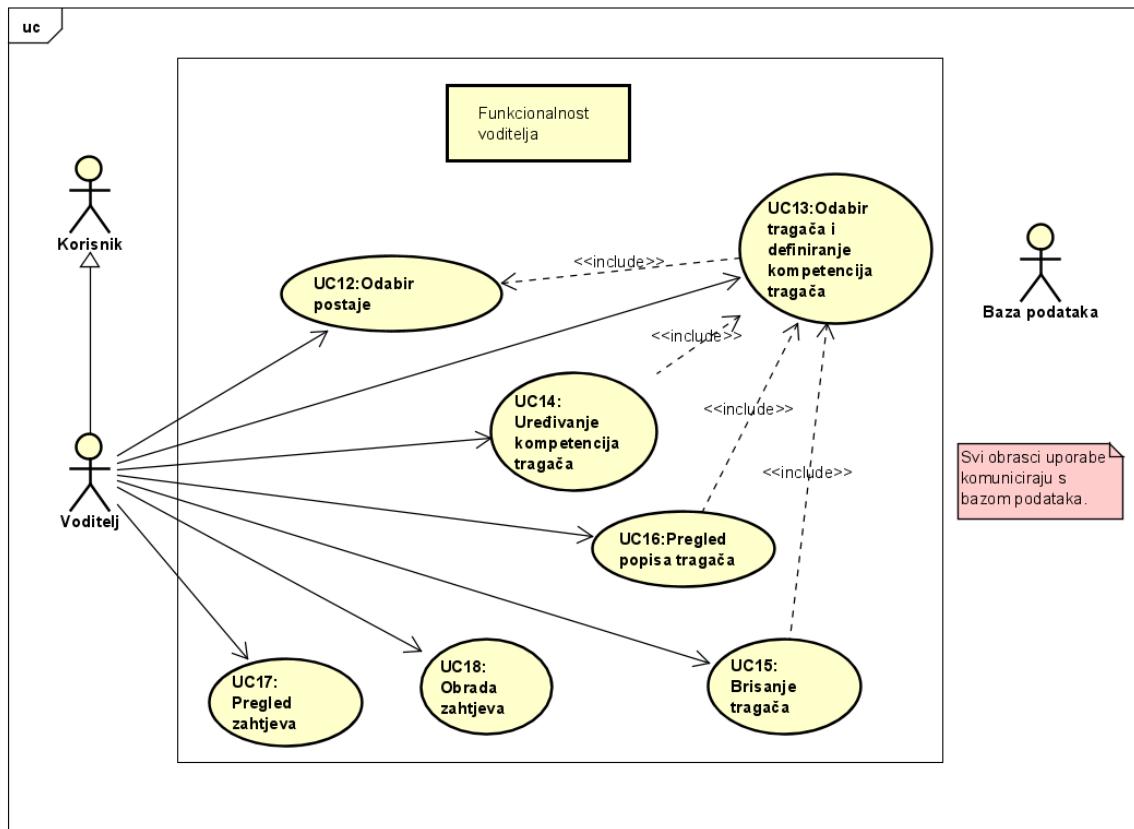
- **Opis osnovnog tijeka:**

1. Glavni sudionik odabire zadatak
2. Aplikacija prikazuje detalje zadatka

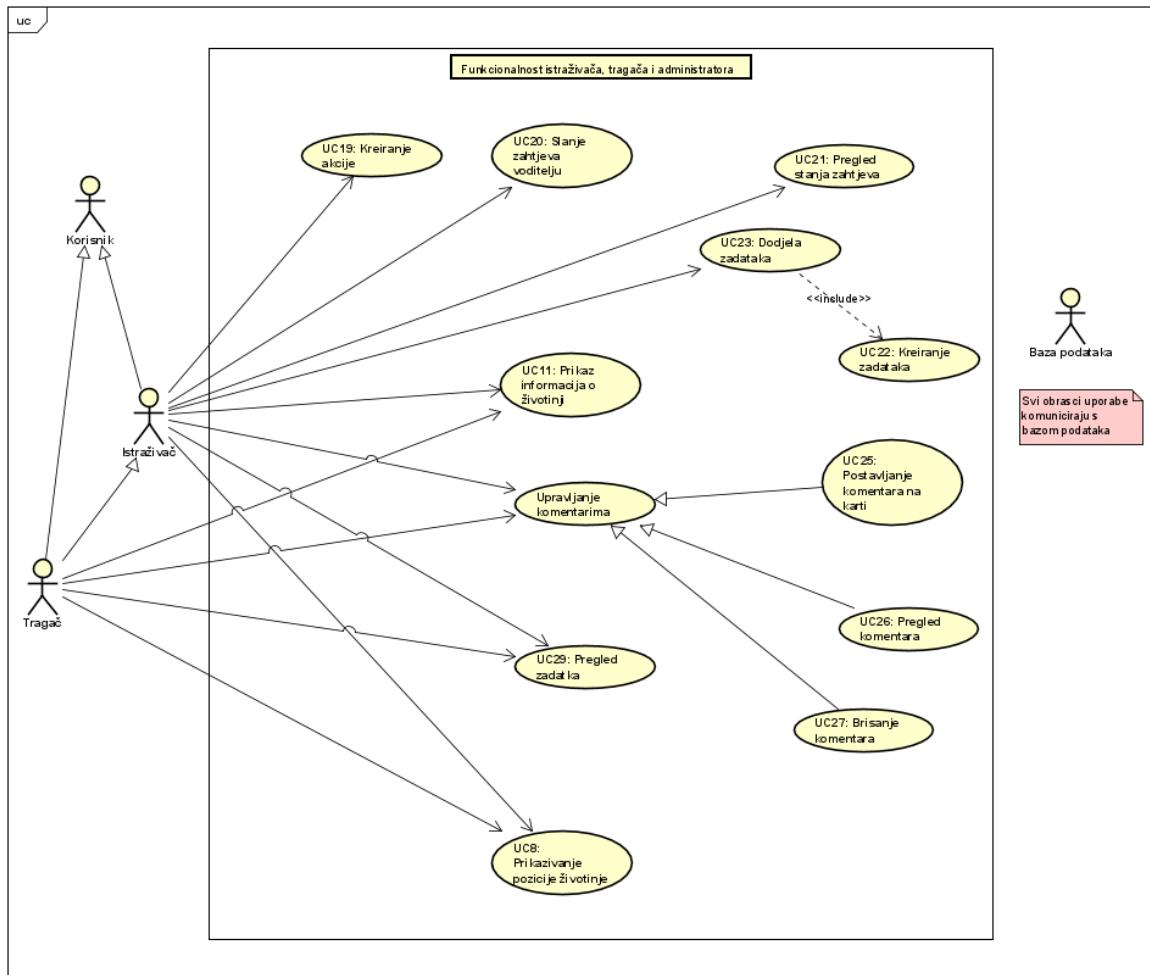
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost admina i korisnika aplikacije



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost voditelja

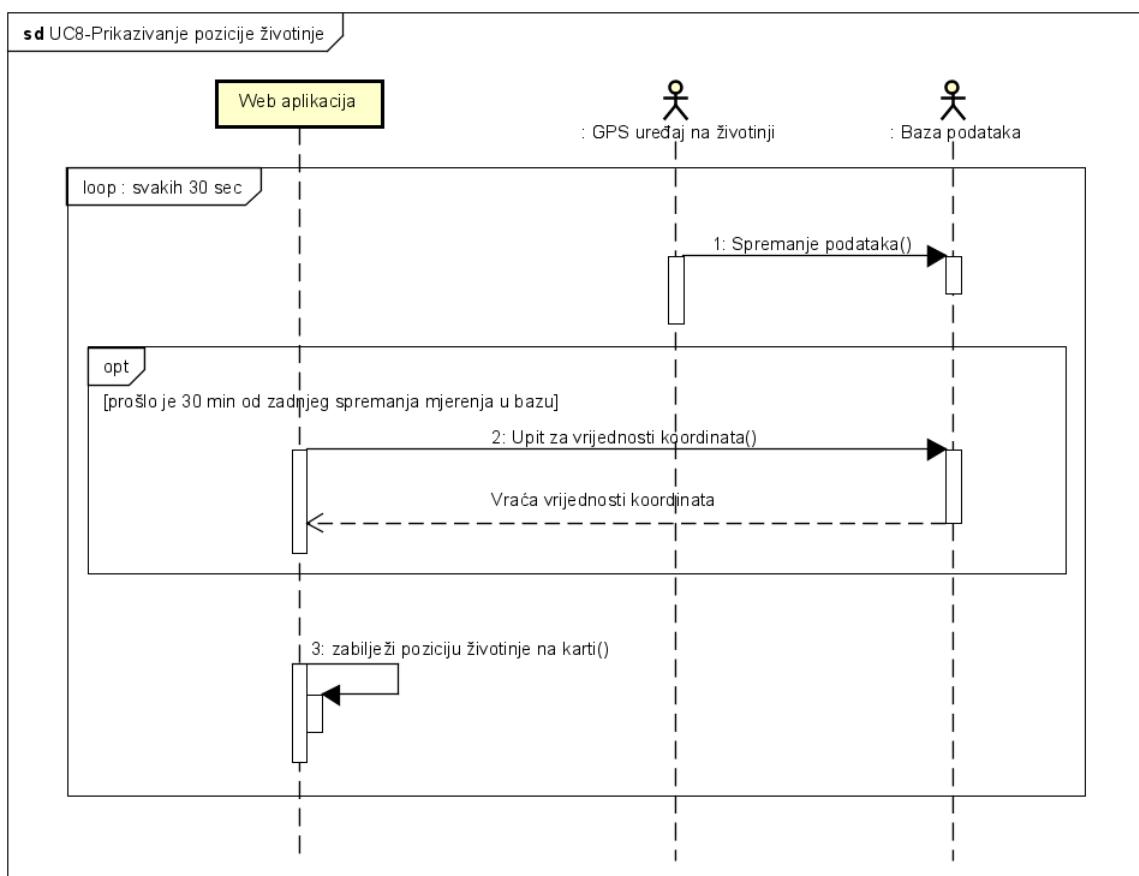


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost istraživača i tragača

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC8 - Prikazivanje pozicije životinje

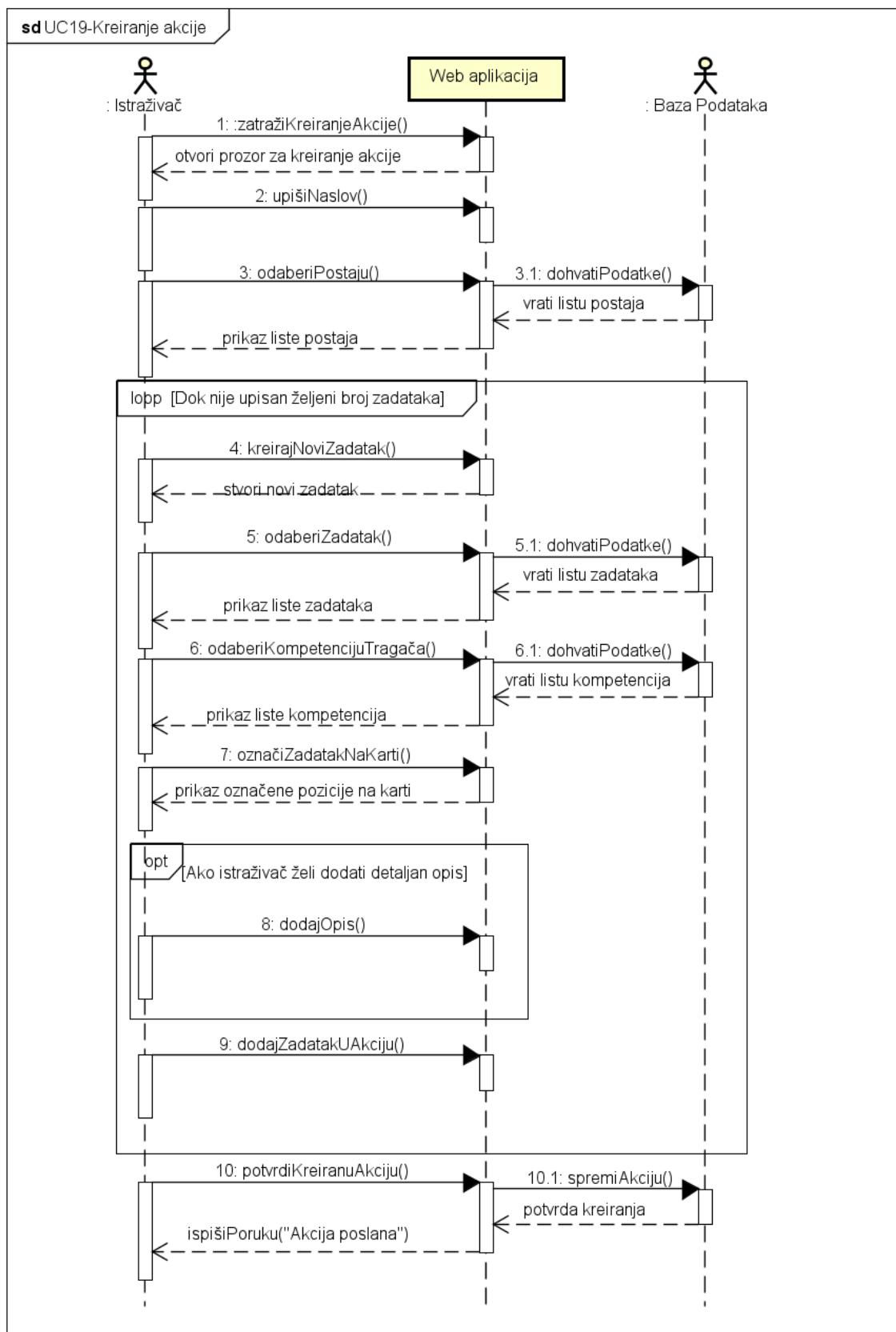
Svakih 30 sekundi spremaju se podaci o GPS uređaju životinje u bazu podataka. Nakon određenog vremena (npr. 30 min), web aplikacija šalje upit bazi podataka za vrijednosti koordinata. Baza podataka vraća vrijednosti koordinata web aplikaciji. Web aplikacija uz pomoć dobivenih koordinata zabilježuje poziciju životinje na karti.



Slika 3.4: Sekvencijski dijagram za UC8

Obrazac uporabe UC19-Kreiranje akcije

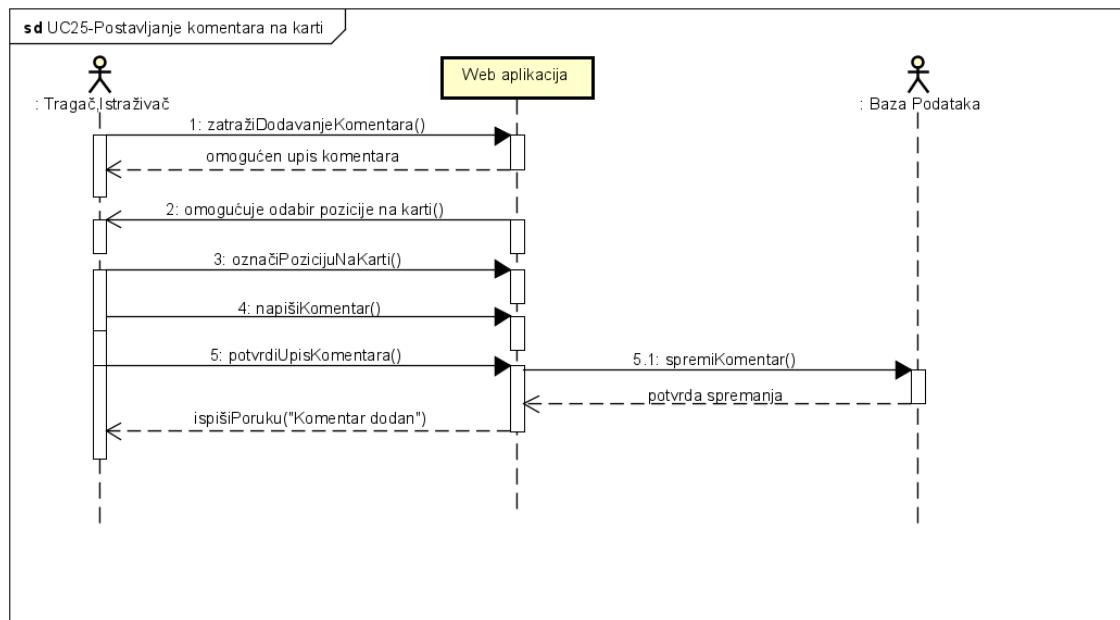
Istraživač zahtjeva kreiranje akcije te se otvara ekran za kreiranje akcije. Istraživač upisuje naslov i odabire postaju kojoj će poslati akciju iz liste postaja koju mu vraća poslužitelj. Istraživač zatražuje dodavanje zadatka te odabire koji zadatak treba obaviti iz liste koju mu vraća poslužitelj. Također, odabire kompetencije koje tragač mora imati za pripadajući zadatak iz liste koju mu vraća poslužitelj. Označuje poziciju na karti gdje treba izvršiti zadatak. Ako istraživač želi dodati detaljan opis upisuje opis u odgovarajući kvadrat. Nakon unesenih svih obveznih podataka, istraživač dodaje zadatak u akciju. Ponavlja proces dodavanja zadataka dok nije upisan željeni broj zadataka. Nakon toga poslužitelju šalje zahtjev za spremanjem akcije. Ako je zahtjev uspješno proveden poslužitelj ispisuje poruku: Akcija poslana.



Slika 3.5: Sekvencijski dijagram za UC19

Obrazac uporabe UC25-Postavljanje komentara na karti

Tragač ili istraživač zatražuje dodavanje komentara na kartu te mu se omogućuje odabir pozicije na karti. Nakon što označi poziciju, upisuje komentar. Potvrđuje upis komentara čime se poslužitelju javlja da spremi komentar na zadanoj poziciji. Ako je komentar uspješno spremljen, šalje poruku korisniku (tragaču ili istraživaču): Komentar dodan.



Slika 3.6: Sekvencijski dijagram za UC25

3.2 Ostali zahtjevi

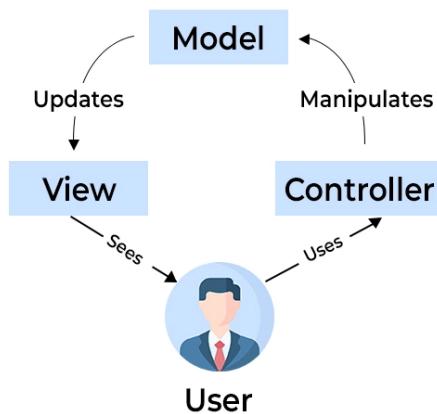
- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orientirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Komunikacija mora biti ostvarena sigurnim protokolom
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS

4. Arhitektura i dizajn sustava

Arhitektura sustava može se podijeliti na tri glavna podsustava:

- *Web preglednik* je program koji korisniku omogućuje pregled web stranica i multimedijalnih sadržaja vezanih uz njih. Kada korisnik posjeti neku web stranicu, preglednik preuzima sadržaj sa servera, prilagodi ga, i pokaže ga na zaslonu računala ili nekog drugog uređaja. Korisnik putem web preglednika šalje zahtjev web poslužitelju i dobiva resurse od njega.
- *Web poslužitelj* je odgovoran za komunikaciju klijenta i web aplikacije. Komunikacija se odvija protokolom za prijenos informacija na webu - HTTP (HyperText Transfer Protocol). Web poslužitelj prima zahtjeve od korisnika putem web preglednika i prosljeđuje ih web aplikaciji na daljnju obradu. Web aplikacija predstavlja središte sustava i obrađuje zahtjeve koje prima od korisnika putem web poslužitelja. Ovisno o zahtjevu, pristupa bazi podataka kako bi dohvatile ili ažurirale podatke.
- *Baza podataka* je podsustav za trajnu pohranu podataka koji se koriste u aplikaciji. Web aplikacija komunicira s bazom podataka kako bi dohvatile, ažurirale, brisale ili pohranile podatke prema korisničkim zahtjevima. Podaci se strukturirano pohranjuju i dohvaćaju putem upita.

Arhitektura sustava temelji se na MVC (Model-View-Controller) konceptu. Ovaj pristup omogućuje jasnu podjelu odgovornosti unutar sustava, olakšava održavanje, testiranje i proširivost aplikacije.



- *Model* predstavlja podatke i poslovnu logiku aplikacije. Model je odgovoran za dohvaćanje, pohranu i upravljanje podacima. Komunicira s Controllerom i obavještava ga o ažuriranim informacijama.
- *View* predstavlja korisničko sučelje aplikacije. Sve što korisnik vidi je dio Viewa. On prikazuje podatke koje dobiva od Modela.
- *Controller* je posrednik između Modela i Viewa. Prima korisničke zahtjeve iz Viewa, interpretira ih i odabire odgovarajuće akcije. Ovisno o korisničkom zahtjevu, Controller može ažurirati Model (kao što je spremanje podataka u bazu) i/ili ažurirati View (kao što je prikazivanje novih podataka na web stranici). Controller šalje informacije od Viewa prema Modelu, a zatim ažurirane podatke iz Modela šalje natrag Viewu za prikaz korisnicima.

Aplikacija ima slojeve frontend i backend, te bazu podataka. Baza podataka zadužena za pohranu podataka naše web aplikacije je PostgreSQL s kojom smo se upoznali na predmetu Baze podataka. To je moćna relacijska baza podataka koja podržava kompleksne SQL upite i omogućuje efikasno pohranjivanje podataka u tablicama. Osim toga, osigurava konzistentnost i pouzdanost podataka. Iz nje backend sloj dohvaća podatke koristeći radni okvir Spring Boot. Ona olakšava brzi razvoj backenda jer Spring Data JPA omogućuje jednostavno rukovanje podacima putem Java objekata, čime se smanjuje količina potrebnog SQL koda. Također omogućuje lak prijenos podataka između frontend i backend slojeva. Za frontend sloj koji korisnicima prikazuje podatke koje dobiva iz backenda putem API-a koristimo Angular. U kombinaciji sa Spring Boot-om i PostgreSQL-om, Angular je moćni radni okvir koja olakšava izradu modernih, skalabilnih, i korisnički prijateljskih web aplikacija. Temelji se na komponentnoj arhitekturi, što olakšava razdvajanje funkcionalnosti i ponovnu upotrebu koda. Komponente su samostalne i

mogu se lako integrirati. Najveća prednost Angulara je što koristi HTML, CSS, te Typescript, jezike koji su poznati svim članovima tima.

4.1 Baza podataka

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

- Status
- Client
- StationLead
- Station
- Request
- Researcher
- ActionComment
- Action
- ExplorerAction
- Explorer
- ExplorerLocation
- AnimalComment
- Animal
- AnimalLocation
- Vehicle
- EducatedFor
- Task
- Token

4.1.1 Opis tablica

Status Ovaj entitet sadržava opis statusa korisnika. Sadrži atribute: statusId i description. Ovaj entitet u vezi je *One-to-Many* s entitetom Researcher preko atributa statusId, u vezi *One-to-Many* s entitetom StationLead preko atributa statusId.

Status		
StatusId	INT	Jedinstveni identifikator statusa korisnika
Description	VARCHAR	Opis značenja određenog identifikatora

Client Ovaj entitet sadržava korisne informacije o korisniku. Sadrži atribute: clientname, password, photoURL, firstName, lastName, email. Ovaj entitet u vezi je *One-to-One* s entitetom Researcher preko atributa clientname, u vezi *One-to-Many* s entitetom ActionComment preko atributa clientname, u vezi *One-to-One* s entitetom StationLead preko atributa clientname, u vezi *One-to-One* s entitetom Explorer preko atributa clientname.

Client		
Client	VARCHAR	Jedinstveno korisničko ime korisnika
Password	VARCHAR	Šifra profila korisnika
PhotoURL	VARCHAR	URL link slike korisnika
FirstName	VARCHAR	Ime korisnika
LastName	VARCHAR	Prezime korisnika
Email	VARCHAR	Jedinstveni Email korisnika

StationLead Ovaj entitet je razdvojeni entitet entiteta Client koji sadrži atribute poseban za voditelja koji govori koju stanicu vodi (stationId), također sadrži i atribute clientname i statusId. Ovaj entitet u vezi je *One-to-One* s entitetom Client preko atributa clientname, u vezi *One-to-One* s entitetom Station preko atributa stationId, u vezi *Many-to-One* s entitetom Status preko atributa statusId, u vezi *One-to-Many* s entitetom Request preko atributa clientname.

StationLead		
Clientname	VARCHAR	korisničko ime voditelja, (client.clientname)
StationId	INT	Identifikator postaje koju voditelj vodi, (station.stationId)
StatusId	INT	Identifikator statusa profila korisnika (verified, declined, pending), (status.statusId)

Station Ovaj entitet sadržava sve važne informacije o postaji. Sadrži atribute: stationId, radius, stationName, stationStatus, Location. U vezi je *One-to-One* s entitetom StationLead preko atributa stationId, u vezi *One-to-Many* s entitetom Explorer preko atributa stationId.

Station		
StationId	INT	Jedinstveni identifikator stanice
Radius	INT	Polumjer kruga koji označava stanicu izražen u metrima
StationName	VARCHAR	Naziv stanice
StationStatus	VARCHAR	Status stanice, može biti zauzeta (ima voditelja) ili slobodna (nema voditelja)
Center	VARCHAR	Lokacija središta stanice

Request Ovaj entitet sadržava informacije o zahtjevu koji istraživač šalje voditelju postaje. Sadrži atribute: requestId, status zahtjeva, numOfPeople, researcher i stationLead. U vezi je *Many-to-One* s entitetom Researcher preko atributa researcher, u vezi *Many-to-One* s entitetom StationLead preko atributa stationLead.

Request		
RequestId	INT	Jedinstveni identifikator zahtjeva
Status	VARCHAR	Status zahtjeva, može biti odobren, odbijen ili u obradi.
NumOfPeople	INT	Broj ljudi koji istraživač zahtjeva od voditelja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Request		
Researcher	VARCHAR	Identifikator istraživača koji šalje zahtjev, (researcher.clientname)
StationLead	VARCHAR	Identifikator voditelja koji prima zahtjev, (stationLead.clientname)

Researcher Ovaj entitet je razdvojeni entitet entiteta Client koji sadrži atribute: clientname istraživača i statusId. U vezi je *One-to-One* s entitetom Client preko atributa clientname, u vezi *Many-to-One* s entitetom Status preko atributa statusId, u vezi *One-to-Many* s entitetom Request preko atributa clientname, u vezi *One-to-Many* s entitetom Action preko atributa clientname.

Researcher		
Clientname	VARCHAR	korisničko ime istraživača, (client.clientname)
StatusId	INT	Identifikator statusa istraživača (verified, declined, pending), (status.statusId)

ActionComment Ovaj entitet sadrži informacije o komentarima ostavljenima na određenoj akciji. Sadrži atribute: CommentId, description, CommentTS, username i actionId akcije kojoj pripada. U vezi je *Many-to-One* s entitetom Client preko atributa clientname, u vezi *Many-to-One* s entitetom Action preko atributa actionId.

ActionComment		
CommentId	INT	Jedinstveni identifikator komentara
Description	VARCHAR	Sadržaj komentara
CommentTS	TIMESTAMP	Vrijeme postavljanja komentara
Location	VARCHAR	Lokacija postavljanja komentara
Clientname	VARCHAR	Identifikator korisnika koji je postavio komentar, (client.clientname)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

ActionComment		
ActionId	INT	Identifikator akcije na koju je postavljen komentar, (action.actionId)

Action Ovaj entitet opisuje koji istraživač je pokrenuo određenu akciju. Ima atribut: actionId, researcher, title i description. U vezi je *Many-to-One* s entitetom Researcher preko atributa researcher. U vezi *One-to-Many* s entitetom ActionComment preko atributa actionId, u vezi *One-to-Many* s entitetom ExplorerAction preko atributa actionId, u vezi *One-to-Many* s entitetom Task preko atributa actionId.

Action		
ActionId	INT	Jedinstveni identifikator akcije
Researcher	VARCHAR	Identifikator istraživača koji je započeo akciju, (researcher.clientname)
Title	VARCHAR	Naslov akcije
Description	VARCHAR	Opis započete akcije

ExplorerAction Ovaj entitet opisuje tragača i akciju na kojoj se nalazio. Ima atribut: explorer i actionId. U vezi je *Many-to-One* s entitetom Explorer preko atributa clientname, u vezi *Many-to-One* s entitetom Action preko atributa actionId.

ExplorerAction		
Explorer	VARCHAR	Korisničko ime tragača, (explorer.clientname)
ActionId	INT	Identifikator akcije tragača, (action.actionId)

Explorer Ovaj entitet je razdvojeni entitet entiteta Client koji sadrži atribut: clientname i stationId. U vezi je *One-to-One* s entitetom Client preko atributa clientname, u vezi *One-to-Many* s entitetom ExplorerAction preko atributa clientname, u vezi *One-to-Many* s entitetom EducatedFor preko atributa clientname, u vezi *One-to-Many* s entitetom Task preko atributa clientname, u vezi *One-to-Many* s entitetom AnimalComment preko atributa clientname, u vezi *One-to-Many* s entitetom

tetom ExplorerLocation preko atributa clientname, u vezi *Many-to-One* s entitetom Station preko atributa stationId.

Explorer		
Clientname	VARCHAR	Korisničko ime tragača, (client.clientname)
StationId	INT	Identifikator postaje kojoj pripada, (station.stationId)

ExplorerLocation Ovaj entitet sadrži lokaciju tragača u određenom vremenu. Ima atribute: LocationTS, explorer i Location. U vezi je *Many-to-One* s entitetom Explorer preko atributa client.name.

ExplorerLocation		
Location	VARCHAR	Lokacija tragača
LocationTS	TIMESTAMP	Vrijeme očitanja lokacije tragača
Explorer	VARCHAR	Korisničko ime tragača, (explorer.clientname)

AnimalComment Ovaj entitet sadrži sve komentare o očitanim životinjama. Ima atribute: commentId, commentTS, comment, explorer, animalId. U vezi je *Many-to-One* s entitetom Explorer preko atributa explorer, u vezi *Many-to-One* s entitetom Animal preko atributa animalId.

AnimalComment		
CommentId	INT	Jedinstveni identifikator komentara
CommentTS	TIMESTAMP	Vrijeme objave komentara
Comment	VARCHAR	Sadržaj komentara
Explorer	VARCHAR	Korisničko ime tragača koji je komentirao, (explorer.clientname)
AnimalId	INT	Identifikator komentirane životinje, (animal.animalId)

Animal Ovaj entitet sadrži sve bitne informacije o očitanoj životinji. Ima atribute: animalId, species, description, photoURL. U vezi je *One-to-Many* s entitetom

AnimalComment preko atributa animalId, u vezi *One-to-Many* s entitetom AnimalLocation preko atributa animalId.

Animal		
AnimalId	INT	Jedinstveni identifikator očitane životinje
Species	VARCHAR	Vrsta životinje
Description	VARCHAR	Opis životinje
PhotoURL	VARCHAR	URL slike životinje

AnimalLocation Ovaj entitet sadrži lokaciju životinje u određenom vremenu. Ima atrbute: LocationTS, Location, AnimalId. U vezi je *Many-to-One* s entitetom Animal preko atributa animalId.

AnimalLocation		
Location	VARCHAR	Lokacija životinje
LocationTS	TIMESTAMP	Vrijeme zapisa lokacije životinje
AnimalId	INT	Identifikator očitane životinje, (animal.animalId)

EducatedFor Ovaj entitet opisuje koji tragači znaju upravljati kojim vozilima. Ima atrbute: vehicleId i explorer. U vezi je *Many-to-One* s entitetom Explorer preko atributa clientname, u vezi *Many-to-One* s entitetom Vehicle preko atributa vehicleId.

EducatedFor		
VehicleId	INT	Identifikator vozila kojeg tragač zna voziti, (vehicle.vehicleId)
Explorer	VARCHAR	Korisničko ime tragača, (explorer.clientname)

Vehicle Ovaj entitet sadrži nazine vozila. Ima atrbute: vehicleId, vehicleType. U vezi je *One-to-Many* s entitetom Task preko atributa vehicleId, u vezi *One-to-Many* s entitetom EducatedFor preko atributa vehicleId.

Vehicle		
VehicleId	INT	Jedinstveni identifikator vozila
VehicleType	VARCHAR	Naziv vozila

Task Ovaj entitet sadrži sve bitne informacije o zadatku. Ima atributе: taskId, description, startLocation, endLocation, taskTS, actionId, explorer, vehicleId. U vezi je *Many-to-One* s entitetom Action preko atributa actionId, u vezi *Many-to-One* s entitetom Explorer preko atributa clientname, u vezi *Many-to-One* s entitetom Vehicle preko atributa vehicleId.

Task		
TaskId	INT	Jedinstveni identifikator zadatka
Description	VARCHAR	Opis zadatka
StartLocation	VARCHAR	Lokacija početka zadatka
EndLocation	VARCHAR	Lokacija završetka zadatka
TaskTS	TIMESTAMP	Vrijeme početka zadatka
ActionId	INT	Identifikator akcije kojoj pripada zadatak, (action.actionId)
Explorer	VARCHAR	Korisničko ime tragača koji izvršava zadatak
VehicleId	INT	Identifikator vozila s kojim se obavlja zadatak, (vehicle.vehicleId)

Token Ovaj entitet sve podatke o tokenu koji se generira pri registraciji ili prijava korisnika u sustav. Ima atributе: tokenId, tokenType, revoked, expired, clientname. U vezi je *Many-to-One* s entitetom Client preko atributa clientname.

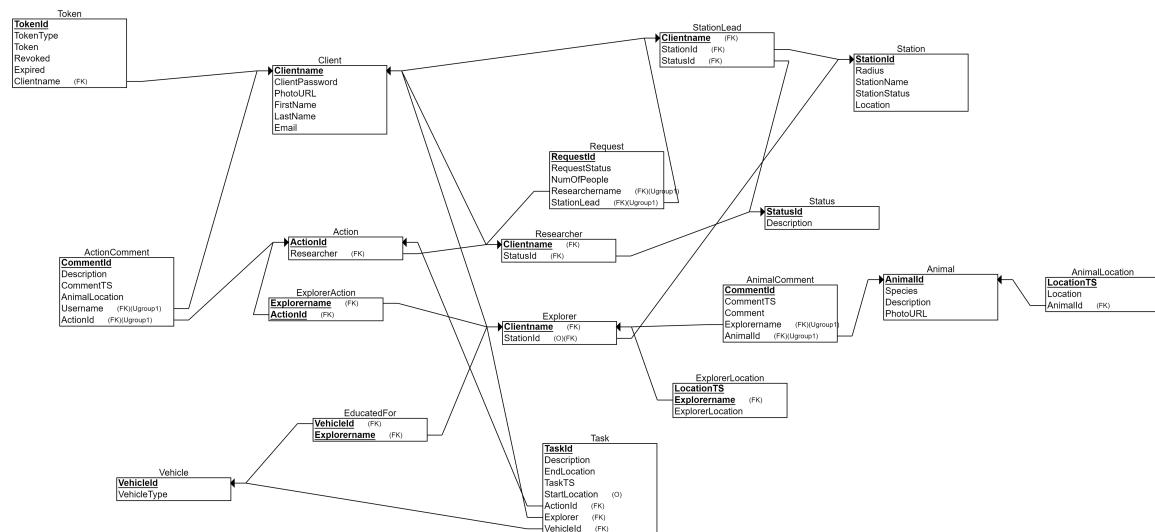
Token		
TokenId	INT	Jedinstveni identifikator tokena
TokenType	VARCHAR	Tip generiranog tokena
Token	VARCHAR	Generirani token

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Token		
Revoked	BOOLEAN	Zastavica koja daje informaciju o tome je li token opozvan
Expired	BOOLEAN	Zastavica koja daje informaciju o tome je li token istekao
Client	VARCHAR	Korisničko ime klijenta koji koristi token, (client.clientname)

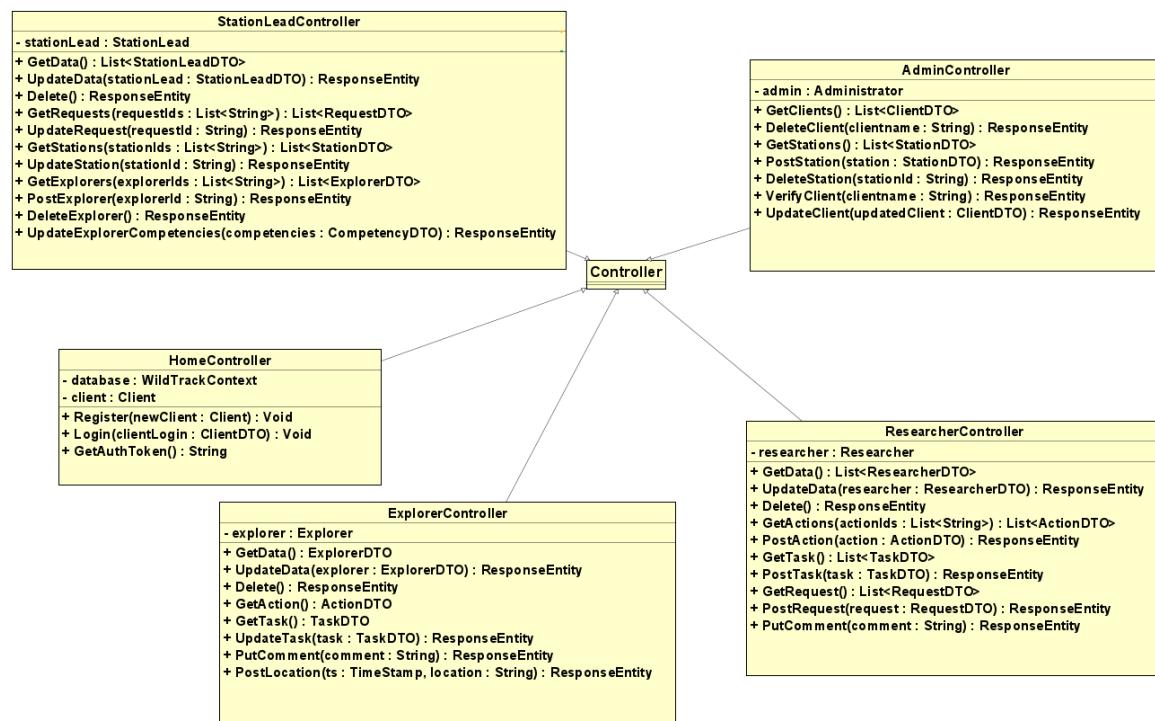
4.1.2 Dijagram baze podataka



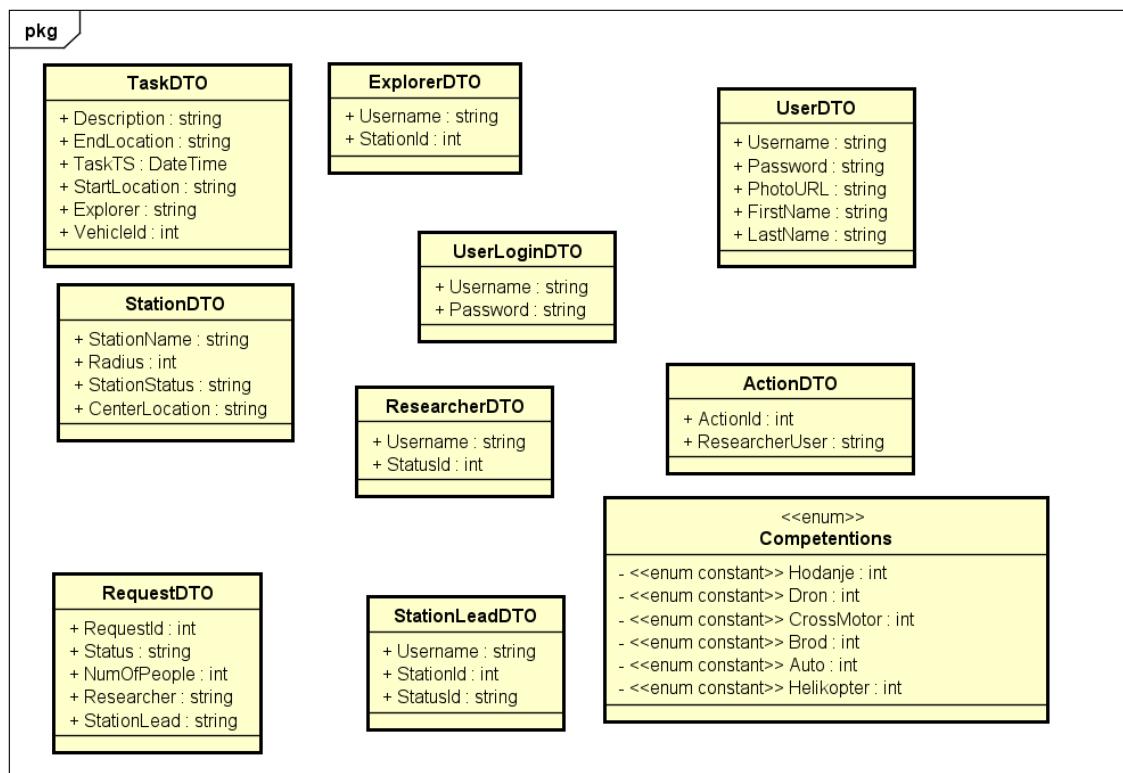
Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda

Na slikama 4.2, 4.3 predstavljeni su razredi koji se odnose na backend komponentu MVC obrasca. Razredi na slici 4.2 nasljeđuju Controller razred. U tim razredima, metode rukuju s Data Transfer Object (DTO), a podaci se dohvaćaju kroz metode definirane u Model razredima. Funkcije implementirane u Controller razredima generiraju JSON datoteke s HTML status kodovima. Metode implementirane u tim razredima manipuliraju modelima i vraćaju zatražene podatke koji su reprezentirani modelima.



Slika 4.2: Dijagram razreda - dio Controllers

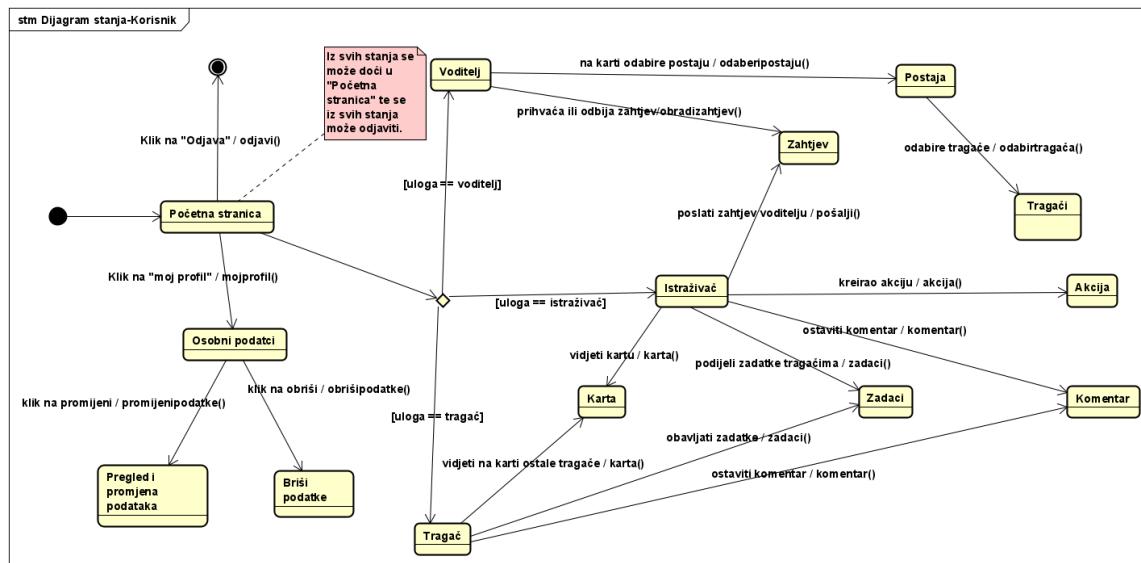


Slika 4.3: Dijagram razreda - dio Data transfer objects

Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke. Razred User predstavlja neregistriranog korisnika koji se može registrirati u sustav unoseći osnovne informacije. Razred UserLogin predstavlja već registriranog korisnika koji se može prijaviti u sustav unoseći korisničko ime i lozinku. Razred Reasearcher predstavlja istraživača koji ima ovlasti kreiranja novih akcija i slanja zahtjeva za tragačima voditelju postaje. Razred StationLead predstavlja voditelja postaje koji ima ovlasti odabrat postaju i tragače svoje vlastite postaje. Razred Explorer predstavlja tragača koji obavlja zadatke koje mu je dodijelio istraživač. Razred Administrator predstavlja administratora sustava koji ima najveće ovlasti.

4.3 Dijagram stanja

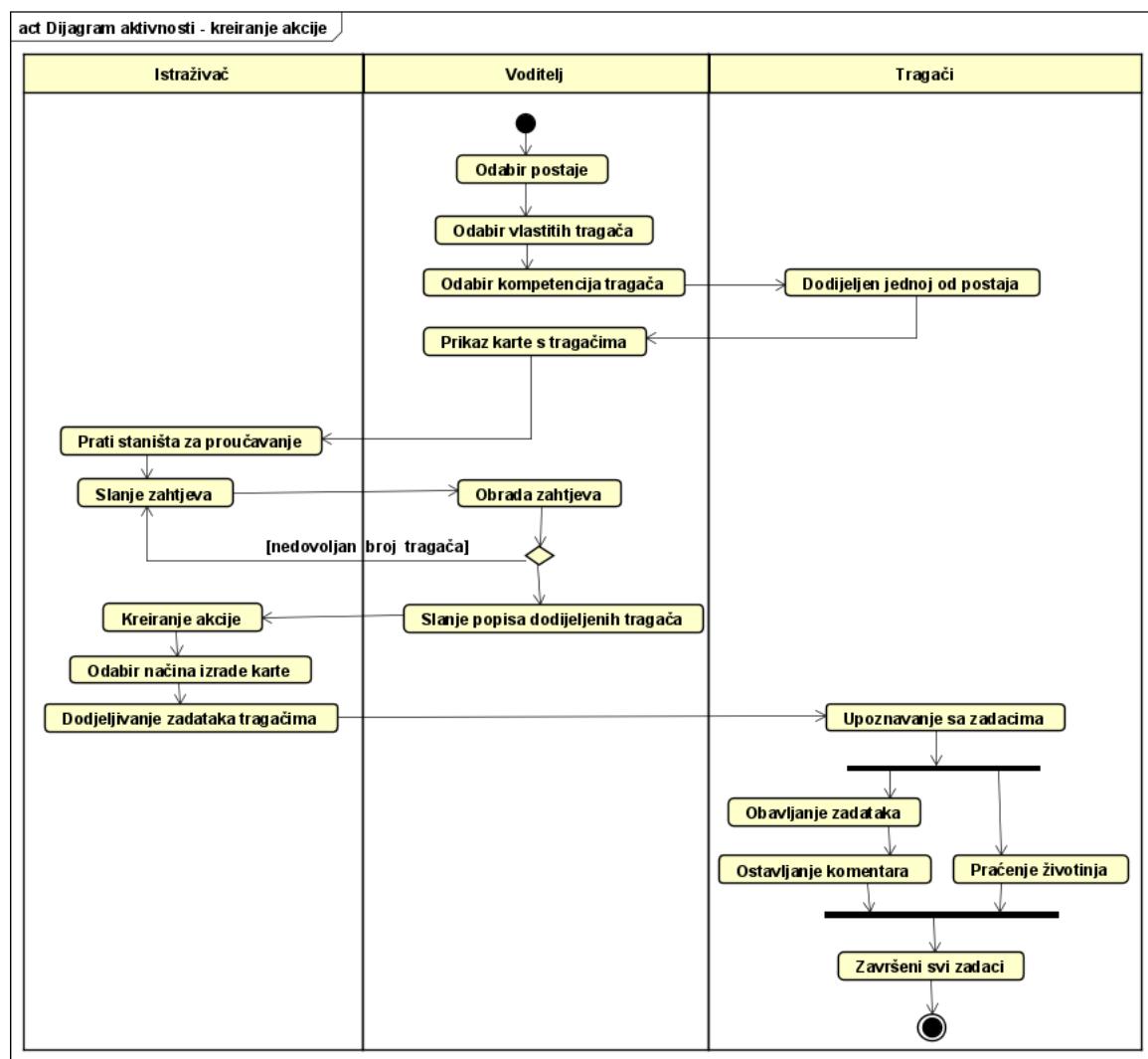
Dijagram stanja jasno prikazuje kako sustav ili njegovi dijelovi prelaze iz jednog stanja u drugo kao odgovor na vanjske ili unutarnje događaje. Na slici 4.4 prikazan je dijagram stanja za registriranog korisnika. Klikom na "Moj Profil" prikazuju mu se njegovi podaci, koje može urediti ili brisati. Ovisno o ulozi koju je odabrao pri registraciji, korisnik može biti voditelj, tragač ili istraživač. Kao voditelj, može odabrati jednu od ponuđenih slobodnih postaja. Nakon toga, ima mogućnost odabratragače svoje postaje, te istovremeno voditelj definira na koji način su njegovi tragači sposobni za izvođenje istraživanja. Voditelj također ima opciju prihvati ili odbiti zahtjev koji mu je posao istraživač. Istraživač kreira akciju i zatim šalje zahtjev voditelju. Nakon što mu se odobri zahtjev, on podijeli zadatke tragačima zaduženima za tu akciju. Tragač na karti može vidjeti svoje zadatke (usput ih obavljati), životinje te ostale tragače na istoj akciji. Prilikom obavljanja zadatka može ostaviti komentare ostalim tragačima.



Slika 4.4: Dijagram stanja

4.4 Dijagram aktivnosti

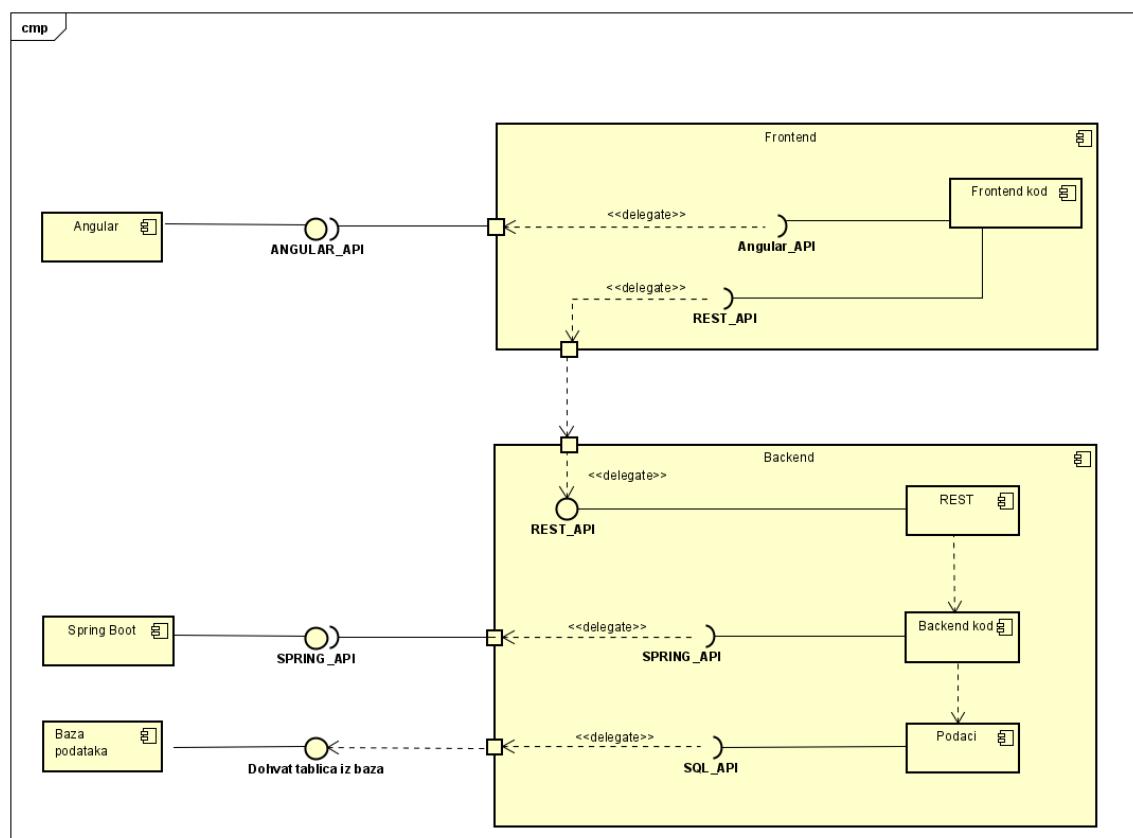
Dijagram aktivnosti upotrebljava se za prikaz tijeka aktivnosti, akcija i odluka procesa u sustavu. Posebno su korisni za vizualizaciju složenih ponašanja kao npr. slijedne i paralelne aktivnosti, kao i točke odlučivanja i uvjetno grananje. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na slici 4.5 prikazan je proces kreiranja akcije. Voditelj odabire postaju i tragače te im odabire kompetencije, a zatim istraživač može poslati zahtjev voditelju. U slučaju prihvatanja zahtjeva istraživač dalje kreira akciju i dodjeljuje zadatke tragačima. Nakon što svi tragači koji sudjeluju u toj akciji obave sve svoje zadatke akcija se smatra završenom.



Slika 4.5: Dijagram aktivnosti

4.5 Dijagram komponenti

Na slici 4.6 prikazan je dijagram komponenti. Korisnik putem web preglednika pristupa aplikaciji putem REST API-ja. Samu aplikaciju možemo podijeliti u dvije komponente. Prva komponenta odgovara frontendu. Ona je izgrađena korištenjem Angulara. Druga komponenta odgovara backendu te je izgrađena pomoću radnog okvira Spring Boot. Komunikacija između frontenda i backenda se odvija pomoću REST API-ja. Koristimo relacijsku bazu podataka kojoj backend pristupa slanjem SQL upita.



Slika 4.6: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu je ostvarena putem korištenja aplikacije WhatsApp¹. Komunikacija sa asistentom ostvarena je korištenjem aplikacije Microsoft Teams². Za izradu dijagrama korišten je Astah Professional³. Za upravljanje izvornim kodom korišten je alat Git⁴. Udaljeni rezervorij projekta nalazi se na platformi GitHub⁵.

Kao razvojno okruženje na backendu korišten je IntelliJ⁶, a na frontendu Visual Studio Code⁷. Visual Studio Code je razvojno okruženje tvrtke Microsoft za Windows, Linux, macOS. Za razvoj softvera koristi Microsoftove platforme poput Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight.

Tehnologije korištene za razvoj backenda su Spring Boot⁸ i programski jezik Java⁹. Tehnologije korištene za razvoj frontenda su Angular¹⁰ i programski jezik TypeScript¹¹. Angular je besplatni okvir(framework) za izgradnju web aplikacija otvorenog izvornog koda temeljen na TypeScriptu. Angular se temelji na principu komponenata gdje nekoliko nezavisnih komponenti surađuje kako bi oblikovali korisničko sučelje. Takva struktura čini aplikacije lako prilagodljivim i održivim.

Za automatizirano testiranje koristen je alat Selenium IDE¹². Za puštanje aplikacije u pogon korištena je platforma Render¹³.

Baza podataka zadužena za pohranu podataka je PostgreSQL¹⁴.

¹<https://www.whatsapp.com/>

²<https://www.microsoft.com/hr-hr/microsoft-teams/group-chat-software/>

³<https://astah.net/products/astah-professional/>

⁴<https://git-scm.com/>

⁵<https://github.com/>

⁶<https://www.jetbrains.com/idea/>

⁷<https://code.visualstudio.com/>

⁸<https://spring.io/projects/spring-boot>

⁹<https://www.java.com/en/>

¹⁰<https://angular.io/>

¹¹<https://www.typescriptlang.org/>

¹²<https://www.selenium.dev/documentation/ide/>

¹³<https://render.com/>

¹⁴<https://www.postgresql.org/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitivanje komponenti ostvareno je pomoću Spring Boot i JUnit 5 alata za ispitivanje.

Na slikama 5.1 i 5.2 prikazani su testovi vezani uz voditelja. Oba testa pokušavaju dodati voditelja na postaju. Na slici 5.2 prikazan je kod iz testa koji uspješno dodaje voditelja na postaju. Slika 5.3 prikazuje kod iz testa gdje se provjerava može li se dodati voditelj na nepostojeću postaju. Vraća da je uspješan test ako se ne može dodati što se i očekuje s ozbirom da je u pitanju nepostojeća postaja.

```
@Test
@DisplayName("Test addStationLeadToStation")
public void addStationLeadToStation() {
    MultipartFile multipartFile = new MockMultipartFile("clientPhoto", "test.jpg",
        "image/jpeg", "test image content".getBytes());
    RegisterDto stationLead = new RegisterDto(clientName: "stationLead", firstName: "test", lastName: "test",
        email: "stationLead@test.test", password: "test", role: "voditeljPostaje", educatedFor: null, multipartFile);
    authenticationService.register(stationLead);
    Station station = new Station(stationId: 1L, stationName: "Postaja test", radius: 2, stationStatus: "Inactive",
        stationLocation: "45.568000, 17.626000", stationLead: null, new ArrayList<>());
    stationRepository.save(station);
    assertTrue(stationLeadService.assignStationToStationLeadByName(stationLeadName: "stationLead", stationName: "Postaja test"));
    clientService.deleteClient(clientName: "stationLead");
}
```

Slika 5.1: Testiranje za uspješno dodavanje voditelja postaji

```
@Test
@DisplayName("Test addStationLeadToStationFail")
public void addStationLeadToStationFail() {
    MultipartFile multipartFile = new MockMultipartFile("clientPhoto", "test.jpg",
        "image/jpeg", "test image content".getBytes());
    RegisterDto stationLead = new RegisterDto(clientName: "stationLead", firstName: "test", lastName: "test",
        email: "stationLead@test.test", password: "test", role: "voditeljPostaje", educatedFor: null, multipartFile);
    authenticationService.register(stationLead);
    assertFalse(stationLeadService.assignStationToStationLeadByName(stationLeadName: "stationLead", stationName: "Nepostojeca postaja"));
    clientService.deleteClient(clientName: "stationLead");
}
```

Slika 5.2: Testiranje za dodavanje voditelja nepostojećoj postaji

Sljedeća tri testa vezana su uz tragača. Prvi od njih nalazi se na slici 5.3 te provjerava jesu li se vozila dobro dodala na tragače pri registraciji. Sljedeći test čiji je kod prikazan na slici 5.4 dodaje tragača na postaju. Zatim provjerava je li postaja prazna? Test je uspješan ako vrati da nije prazna jer smo prethodno dodali tragača. Posljednji test vezan uz tragača miće tog tragača kojeg smo prethodno dodali iz

postaje. Zatim provjerava je li postaja prazna? Ovaj put je test uspješan ako vrati da je postaja prazna. Kod je prikazan na slici 5.5.

```
@Test
@DisplayName("Test addVehiclesToExplorer")
public void addVehiclesToExplorer() {
    List<String> educatedFor = List.of("dron", "helikopter", "hodanje");
    MultipartFile multipartFile = new MockMultipartFile("clientPhoto", "test.jpg",
        "image/jpeg", "test image content".getBytes());
    RegisterDto explorer = new RegisterDto(clientName: "explorer", firstName: "test", lastName: "test",
        email: "explorer@test.test", password: "test", role: "tragac", educatedFor, multipartFile);
    authenticationService.register(explorer);
    Explorer explorer2 = explorerRepository.findByExplorerName(clientName: "explorer");
    List<String> explorerVehicles = new ArrayList<>();
    for (Vehicle vehicle : explorer2.getVehicles()) {
        explorerVehicles.add(vehicle.getVehicleType());
    }
    Collections.sort(explorerVehicles);
    assertEquals(explorerVehicles, educatedFor);
    explorer2.setVehicles(null);
    explorerRepository.save(explorer2);
    clientService.deleteClient(clientName: "explorer");
}
```

Slika 5.3: Testiranje jesu li se vozila dodala na tragače

```
@Test
@DisplayName("Test addExplorerToStation")
public void addExplorerToStation() {
    List<String> educatedFor = List.of("dron");
    MultipartFile multipartFile = new MockMultipartFile("clientPhoto", "test.jpg",
        "image/jpeg", "test image content".getBytes());
    RegisterDto explorer = new RegisterDto(clientName: "explorer", firstName: "test", lastName: "test",
        email: "explorer@test.test", password: "test", role: "tragac", educatedFor, multipartFile);
    authenticationService.register(explorer);
    Station station = new Station(stationId: 1L, stationName: "Postaja test", radius: 2, stationStatus: "Inactive",
        stationLocation: "45.568000, 17.626000", stationLead: null, new ArrayList<>());
    stationRepository.save(station);
    stationLeadService.assignExplorerToStation(stationName: "Postaja test", explorerName: "explorer");
    Station station2 = stationRepository.findByStationName("Postaja test");
    assertFalse(station2.getExplorers().isEmpty());
}
```

Slika 5.4: Testiranje je li postaja prazna

```
@Test
@DisplayName("Test removeExplorerFromStation")
public void removeExplorerFromStation() {
    // Mora se izvoditi nakon addExplorerToStation
    stationLeadService.removeExplorerFromStation( stationName: "Postaja test", explorerName: "explorer");
    Station station2 = stationRepository.findByStationName("Postaja test");
    assertTrue(station2.getExplorers().isEmpty());
    Explorer explorer2 = explorerRepository.findByExplorerName( clientName: "explorer");
    explorer2.setStation(null);
    explorer2.setVehicles(null);
    station2.setExplorers(null);
    stationRepository.save(station2);
    explorerRepository.save(explorer2);
    clientService.deleteClient( clientName: "explorer");
    stationRepository.delete(station2);
}
```

Slika 5.5: Testiranje je li postaja prazna

Na slikama 5.6 i 5.7 prikazani su testovi vezani uz istraživača. Nakon što se korisnik registrira i zatraži ulogu istraživač potrebna je još dodatna potvrda od strane admina. Na slici 5.6 prikazan je test kod kojeg se prihvata istraživač nakon registracije. U slučaju testa na slici 5.7 zahtjev je odbijen.

```
@Test
@DisplayName("Test acceptClient")
public void acceptClient() {
    MultipartFile multipartFile = new MockMultipartFile( name: "clientPhoto", originalFilename: "test.jpg",
        contentType: "image/jpeg", "test image content".getBytes());
    RegisterDto researcher = new RegisterDto( clientName: "researcher", firstName: "test", lastName: "test",
        email: "researcher@test.test", password: "test", role: "istrazivac", educatedFor: null, multipartFile);
    authenticationService.register(researcher);
    clientService.updateClientStatusByClientName( clientName: "researcher", status: 2);
    Client client = clientRepository.findByClientName("researcher").orElseThrow();
    assertEquals( expected: 2, client.getResearcher().getStatus().getStatusId());
    clientService.deleteClient( clientName: "researcher");
}
```

Slika 5.6: Testiranje prihvatanja istraživača

```
@Test
@DisplayName("Test rejectClient")
public void rejectClient() {
    MultipartFile multipartFile = new MockMultipartFile("clientPhoto", "originalFilename: \"test.jpg\"",
        "contentType: \"image/jpeg\"", "test image content".getBytes());
    RegisterDto researcher = new RegisterDto("clientName: \"researcher\"", "firstName: \"test\"", "lastName: \"test\"",
        "email: \"researcher@test.test\"", "password: \"test\"", "role: \"istraziyac\"", "educatedFor: null", multipartFile);
    authenticationService.register(researcher);
    clientService.updateClientStatusByClientName("clientName: \"researcher\"", "status: 3");
    Client client = clientRepository.findByClientName("researcher").orElseThrow();
    assertEquals(expected: 3, client.getResearcher().getStatus().getStatusId());
    clientService.deleteClient("clientName: \"researcher\"");
}
```

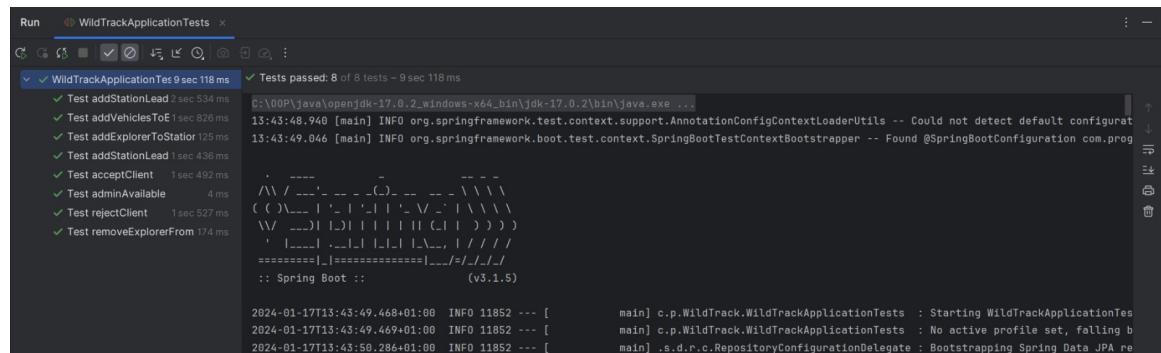
Slika 5.7: Testiranje odbijanja istraživača

Ovaj test provjerava postoji li uopće admin u bazi. Test koda prikazan je na slici 5.8.

```
@Test  
@DisplayName("Test adminAvailable")  
public void adminAvailable() { assertTrue(clientRepository.existsByClientName("admin")); }
```

Slika 5.8: Testiranje postoji li admin u bazi

Na slici 5.9 prikazana je snimka zaslona terminala u IntelliJ IDE-u, kao rezultat izvođenja prethodno navedenih osam ispitnih slučajeva.



Slika 5.9: Rezultati ispitivanja pomoću Spring-a i JUnit-a

5.2.2 Ispitivanje sustava

Ispitivanje sustava provedeno je pomoću dodatka za preglednik Selenium IDE. U nastavku su opisani provedeni ispitni slučajevi. Na slici 5.10 vidimo kako se admin pokušava ulogirati sa korisničkim imenom "admin2". To je pogrešno korisničko ime te se admin neće moći prijaviti. Slika 5.12 prikazuje isto to samo admin se prijavljuje pomoću korisničkog imena "admin" koje je njegovo pravo i uspješno se prijavljuje u aplikaciju.

The screenshot shows the Selenium IDE interface. At the top, there are buttons for running tests, a search bar, and a URL input set to "http://localhost:4200/register". Below the toolbar is a table of test steps:

	Command	Target	Value
1	open	/login	
2	set window size	760x824	
3	click	id=user	
4	type	id=user	admin2
5	click	id=pass	
6	type	id=pass	admin123
7	click	css=.button	

Below the table are four input fields for defining a new step:

- Command: open
- Target: /login
- Value: (empty)
- Description: (empty)

Slika 5.10: Parametri sa pogrešnim korisničkim imenom

Running 'LoginBadCreds'

1. open on /login **OK**
2. setWindowSize on 760x824 **OK**
3. click on id=user **OK**
4. type on id=user with value admin2 **OK**
5. click on id=pass **OK**
6. type on id=pass with value admin123 **OK**
7. click on css=.button **OK**

'LoginBadCreds' completed successfully

Slika 5.11: Rezultat prvog ispitnog slučaja

Command	Target	Value
1 open	/login	
2 set window size	760x824	
3 click	id=user	
4 type	id=user	admin
5 click	id=pass	
6 type	id=pass	admin123
7 click	css=.button	

Command	<input type="text" value="open"/>	<input type="button" value="▼"/>	<input type="button" value="//"/>	<input type="button" value="✖"/>	<input type="button" value="✚"/>
Target	<input type="text" value="/login"/>	<input type="button" value="✖"/>	<input type="button" value="🔍"/>		
Value	<input type="text"/>				
Description	<input type="text"/>				

Slika 5.12: Parametri sa valjanim korisničkim imenom

Running 'LoginGoodCreds'

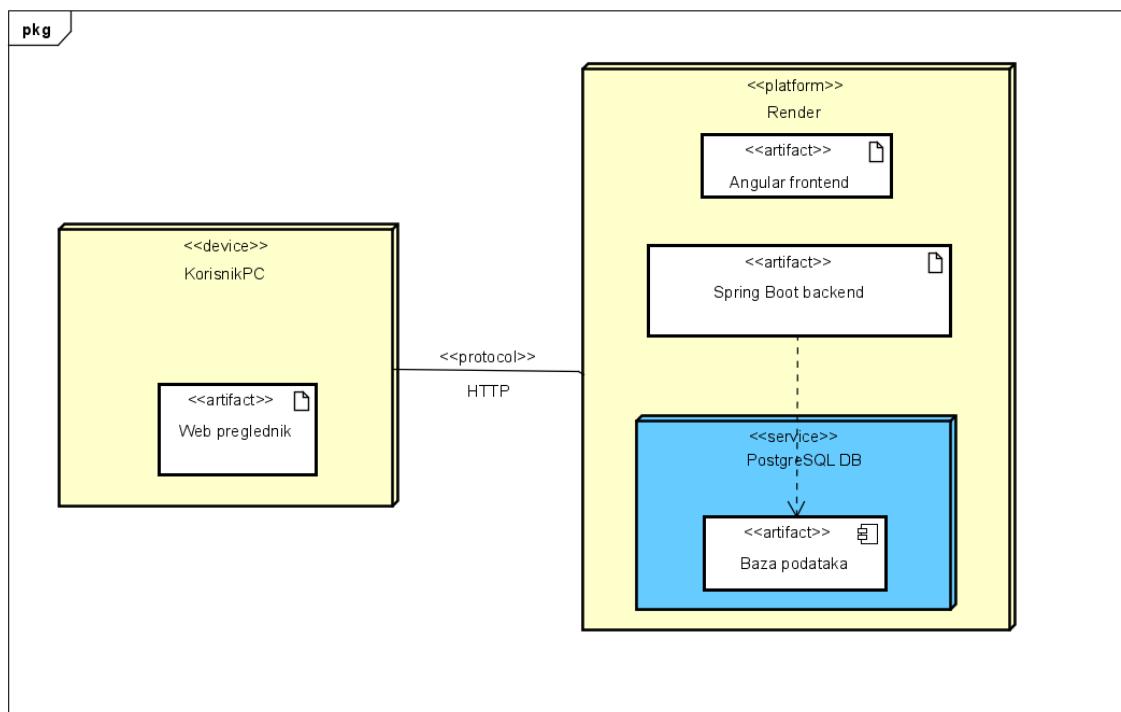
1. open on /login **OK**
2. setWindowSize on 760x824 **OK**
3. click on id=user **OK**
4. type on id=user with value admin **OK**
5. click on id=pass **OK**
6. type on id=pass with value admin123 **OK**
7. click on css=.button **OK**

'LoginGoodCreds' completed successfully

Slika 5.13: Rezultat drugog ispitnog slučaja

5.3 Dijagram razmještaja

Na slici 5.14 nalazi se dijagram razmještaja. Dijagram razmještaja prikazuje fizičku strukturu programskog sustava. Glavna svrha mu je pobliže prikazati arhitekturu razmještaja sustava. Korisnici pristupaju aplikaciji korištenjem web preglednika. Komunikacija se odvija putem protokola HTTP. Na platformi Render se nalaze poslužitelji za frontend, backend i bazu podataka.



Slika 5.14: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Pri puštanju u pogon prvo je potrebno registrirati se na stranici "Render". Preporučamo koristeći vlastiti GitHub profil. Preduvjet za puštanje web aplikacije u pogon je zasebno pustiti u pogon backend, frontend i PostgreSQL bazu podataka. Za deploy baze podataka odabiremo opciju "New" → "PostgreSQL". Nakon toga potrebno je upisati ime baze i odabrati instancu. U našem slučaju ta instanca je "free". Zatim kreiramo bazu. Postupak kako to učiniti prikazan je na slikama 5.15 i 5.16.

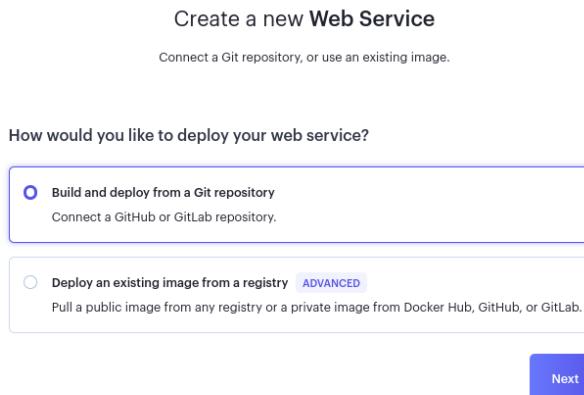
The screenshot shows the Render dashboard with two services listed: "WildTrack-backend" and "wildTrack-frontend-2". Both services are in an "Active" state. A modal window titled "New PostgreSQL" is open on the right, showing options for creating a new database instance. The "Free" plan is selected, which includes 256 MB RAM, 0.1 CPU, and 1 GB storage. The modal also includes sections for "Datadog API Key" and "Instance Type" with options for hobby and professional use.

Slika 5.15: Puštanje u pogon baze

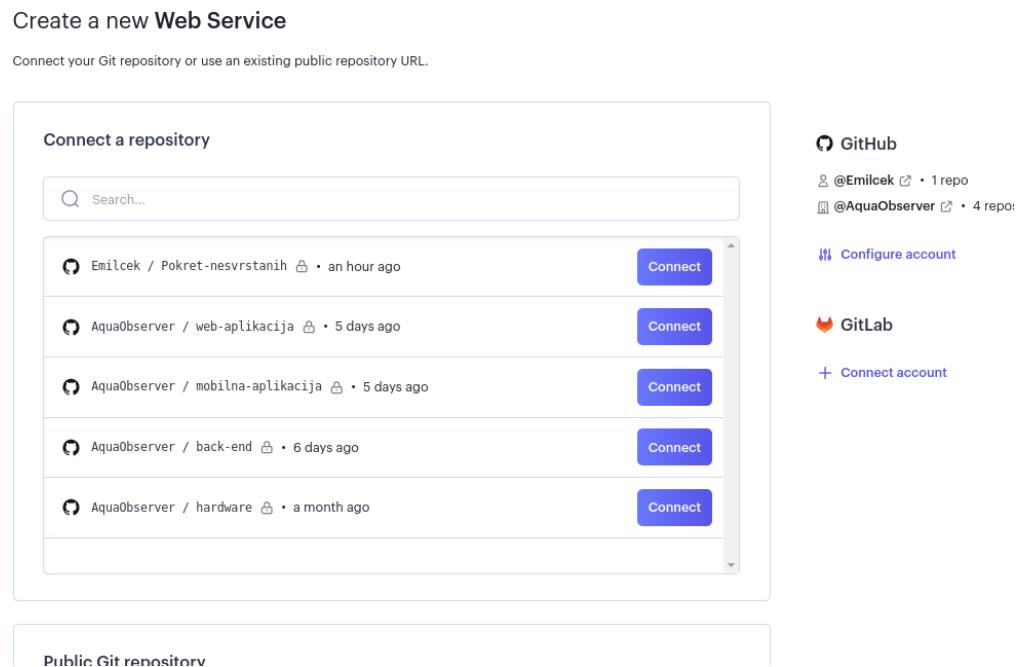
The screenshot shows the "New PostgreSQL" creation form. It includes fields for "Name" (WildTrack-DB), "Database" (optional), "User" (optional), "Region" (Frankfurt (EU Central)), "PostgreSQL Version" (15), and "Instance Type" (Free plan). The "Free" plan is highlighted, showing 256 MB RAM, 0.1 CPU, and 1 GB storage. A note indicates that free instances do not support backups and expire 90 days after creation.

Slika 5.16: Puštanje u pogon baze

Sljedeće ćemo pustiti u pogon backend. Odabiremo opciju "New" → "Web Service" → "Build and deploy from Git repo" → povežete GitHub profil i odaberete projekt koji želite pustiti u pogon. Postavite ime, regiju i root direktorij. Mi koristimo takozvani "monorepo" pa u root upisujemo "IzvorniKod/BackEnd". Postupak kako pustiti u pogon backend vidljiv je na sljedećim slikama.



Slika 5.17: Puštanje u pogon backenda



Slika 5.18: Povezivanje s GitHub profilom

You are deploying a web service for [Emilcek/Pokret-nesvrstanih](#).

The screenshot shows the configuration for a web service named "WildTrack -backend". It includes fields for Name (wildTrack-backend), Region (Frankfurt (EU Central)), Branch (main), Root Directory (IzvorniKod/BackEnd), Runtime (Docker), and Instance Type (Free tier). The Free tier provides 512 MB (RAM) and 0.1 CPU. A note indicates that free instances spin down after periods of inactivity and do not support SSH access, scaling, one-off jobs, or persistent disks.

Slika 5.19: Postavljanje imena, regije i root direktorija

Puštanje u pogon frontenda je vrlo slično. Odaberemo opciju "New" → "Static site" → odabir projekta → postavke konfiguracije možete vidjeti na slici 5.20.

You are deploying a static site for [Emilcek/Pokret-nesvrstanih](#).

The screenshot shows the configuration for a static site named "wildTrack-frontend". It includes fields for Name (wildTrack-frontend), Branch (main), Root Directory (IzvorniKod/FrontEnd), Build Command (\$ npm install; npm run build:prod), Publish directory (dist/my-app), and Environment Variables (NAME_OF_VARIABLE: value). A note specifies that environment variables are used for API keys.

Slika 5.20: Postavke konfiguracije za frontend

Sljedeći vrlo važan korak je postavljanje environment varijabli na frontend i backend. Kako to izgleda prikazano je na slikama 5.21 i 5.22.

The screenshot shows the 'Environment' section of the WildTrack dashboard. On the left, a sidebar lists various monitoring categories: Events, Logs, Disks, Environment (which is selected and highlighted in blue), Shell, Previews, Jobs, Metrics, Scaling, and Settings. The main content area is titled 'Environment Variables' with the sub-instruction: 'Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)' Below this, there is a table with two columns: 'Key' and 'Value'. The keys listed are DB_INIT, DB_PASS, DB_URL, DB_USERNAME, FRONTEND_API_URL, JWT_SECRET_KEY, MAIL_PASS, and MAIL_USER. Each key has a corresponding value field containing several asterisks ('*****') and a trash icon in the top right corner of its row. At the bottom of the table are three buttons: a blue '+' button labeled '+ Add Environment Variable', a white 'Save Changes' button with a blue outline, and a purple 'Save Changes' button.

Slika 5.21: Postavljanje environment varijabli za backend

The screenshot shows the 'Environment' section of the WildTrack dashboard. The sidebar on the left is identical to the previous screenshot, with 'Environment' selected. The main content area is titled 'Environment Variables' with the same sub-instruction: 'Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)' Below this, there is a table with two columns: 'Key' and 'Value'. The key listed is BASE_API_URL, which has a value field containing several asterisks ('*****') and a trash icon in the top right corner. At the bottom of the table are three buttons: a blue '+' button labeled '+ Add Environment Variable', a white 'Save Changes' button with a blue outline, and a purple 'Save Changes' button.

Slika 5.22: Postavljanje environment varijabli za frontend

Kako bi ste došli do varijabli koje su vezane uz bazu na dashboardu odaberete bazu i unutar connections se nalaze sve slike, a API URL za frontend odnosno backend se nađe tako da se odabere frontend ili backend slika.

Connections

Hostname: dpg-cla83hparfss73dbfb70-a

Port: 5432

Database: wildtrack_db_aamc

Username: wildtrack_db_aamc_user

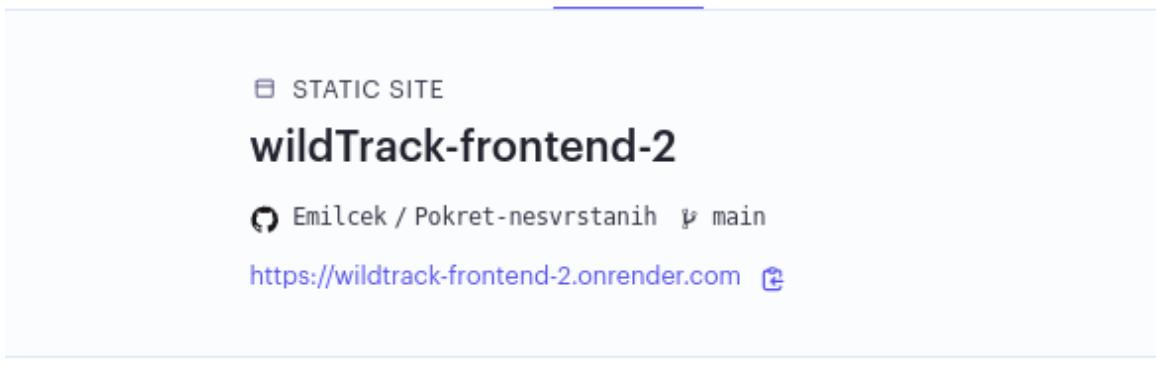
Password:

Internal Database URL:

External Database URL:

PSQL Command:

Slika 5.23: Prikaz connections-a



Slika 5.24: API URL za frontend/backend

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije pod nazivom "Wildtrack". Sama ideja aplikacije je olakšati korisnicima koordinaciju prilikom prolazeњa i praćenja divljih životinja. Nakon 17 tjedana timskog rada, ostvaren je cilj te je projekt uspješno završen. Projekt je bio proveden kroz tri faze.

Prva faza uključivala je okupljanje tima radi diskusije o općim konceptima za aplikaciju, izražavanje individualnih interesa i želja za sudjelovanjem u provedbi projekta te konačno raspodjelom zadataka za tu fazu. Kvalitetna provedba prve faze uvelike je olakšala daljni rad. Krenulo se sa intenzivnim radom na dokumentaciji. U sklopu toga izrađeni su obrasci i dijagrami (obrasci uporabe, sekvenički i dijagrami razreda) koji su bili od pomoći članovima tima koji su zaduženi za razvoj frontenda odnosno backenda aplikacije. Prva faza trajala je prvih 7 tjedana.

U drugoj fazi fokus je bio na implementaciji same aplikacije gdje su svi članovi tima surađivali zajedno na programskom ostvarivanju projekta. Nedostatak iskustva većine članova tima prisilio je tim na neovisno učenje odabranih alata i programskih jezika kako bi ostvarili postavljene ciljeve. Osim što se radilo na implementaciji rješenja, u drugoj fazi projekta bilo je nužno napraviti dodatne UML dijagrame (stanja, aktivnosti, komponenti i razmještaja) te pripremiti pripadajuću dokumentaciju. Time smo značajno uštedjeli vrijeme tijekom razvoja aplikacije.

Treća faza uključivala je testiranje sustava, pronalazak i ispravak grešaka, dorada izgleda aplikacije te implementaciji preostalih funkcionalnosti. Tijekom ove faze pojavilo se niz manjih zadataka koje su članovi tima preuzimali i obavljali samostalno.

Članovi unutar time komunicirali su putem Whatsappa i Discorda, što je omogućilo brzu i efikasnu informiranost o napretku projekta među svim članovima grupe. Jednom tjedno svaki član tima bi prezentirao svoj napredak od prošlog puta i eventualno zatražio pomoć od ostatka tima ako je naišao na neki problem.

Aplikaciju je moguće proširiti na način da tragači ostavljaju komentare voditelju kako im je bilo izvoditi akciju kod pojedinih istraživača, ali isto tako da istraživač šalje povratnu informaciju voditelju kako je zadovoljan s pojedinim tragačem.

Sudjelovanje u ovakovom projektu bilo je izuzetno vrijedno isukstvo za sve članove

timu. Ovo je bio naš prvi ozbiljniji grupni projekt i uspješno smo se snašli u izazovima koje je projekt donio. Osim toga stekli smo uvid u važnost dobre vremenske organizacije i koordinacije među članovima tima. Konflikata unutar tima gotovo da nije ni bilo. Sudjelovanjem u više ovakvih projekata i boljim poznavanjem programskih alata projekt bi zasigurno bio kvalitetnije i brže ostvaren. Ovaj projekt predstavljao nam je prvi ozbiljniji susret s tehnologijama poput Gita i LateX-a. Tijekom razvoja web aplikacije, stekli smo iskustvo u korištenju modernih radnih okvira. Izrazito smo zadovoljni postignutim rezultatima i zajedničkim naporom tima koji je doveo do tih rezultata.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. The Unified Modeling Language, <https://www.uml-diagrams.org/>
5. Astah Community, <http://astah.net/editions/uml-new>
6. BezKoder, <https://www.bezkoder.com/spring-boot-jwt-authentication/>

Indeks slika i dijagrama

2.1	iNaturalist	6
2.2	eBird	7
2.3	Movebank	7
2.4	Project Noah	8
3.1	Dijagram obrasca uporabe, funkcionalnost admina i korisnika aplikacije	24
3.2	Dijagram obrasca uporabe, funkcionalnost voditelja	25
3.3	Dijagram obrasca uporabe, funkcionalnost istraživača i tragača	26
3.4	Sekvencijski dijagram za UC8	27
3.5	Sekvencijski dijagram za UC19	29
3.6	Sekvencijski dijagram za UC25	30
4.1	Dijagram baze podataka	42
4.2	Dijagram razreda - dio Controllers	43
4.3	Dijagram razreda - dio Data transfer objects	44
4.4	Dijagram stanja	45
4.5	Dijagram aktivnosti	46
4.6	Dijagram komponenti	47
5.1	Testiranje za uspješno dodavanje voditelja postaji	49
5.2	Testiranje za dodavanje voditelja nepostojećoj postaji	49
5.3	Testiranje jesu li se vozila dodala na tragače	50
5.4	Testiranje je li postaja prazna	50
5.5	Testiranje je li postaja prazna	51
5.6	Testiranje prihvaćanja istraživača	51
5.7	Testiranje odbijanja istraživača	52
5.8	Testiranje postoji li admin u bazi	52
5.9	Rezultati ispitivanja pomoću Spring-a i JUnit-a	52
5.10	Parametri sa pogrešnim korisničkim imenom	53
5.11	Rezultat prvog ispitnog slučaja	54

5.12 Parametri sa valjanim korisničkim imenom	54
5.13 Rezultat drugog ispitnog slučaja	55
5.14 Dijagram razmještaja	56
5.15 Puštanje u pogon baze	57
5.16 Puštanje u pogon baze	57
5.17 Puštanje u pogon backenda	58
5.18 Povezivanje s GitHub profilom	58
5.19 Postavljanje imena, regije i root direktorija	59
5.20 Postavke konfiguracije za frontend	59
5.21 Postavljanje environment varijabli za backend	60
5.22 Postavljanje environment varijabli za frontend	60
5.23 Prikaz connections-a	61
5.24 API URL za frontend/backend	61
6.1 prikaz aktivnosti na repozitoriju	74

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 20. listopada 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:
 - sastanak s asistentom
 - analiza zadatka
 - upoznavanje s timom
 - odabir tehnologija i alata
 - prvobitna raspodjela posla po članovima (backend/frontend)

2. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:
 - brainstorming ideja
 - opis projektnog zadatka
 - definiranje funkcionalnih zahtjeva
 - detaljnija raspodjela posla po članovima

3. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:
 - definiranje oblikovnih obrazaca
 - transfer znanja
 - raspodjela zadataka (oblikovni obrasci, sekvencijski dijagrami, ostali zahtjevi, arhitektura i dizajn sustava)

4. sastanak

- Datum: 1. studenog 2023.
- Prisustvovali: E.Prpić, I.Lisica, S.Medjaković
- Teme sastanka:
 - definiranje entiteta i veza baze
 - izrada ER i relacijskog modela baze

5. sastanak

- Datum: 2. studenog 2023.
- Prisustvovali: I.Cvijetić, F.Vitković, S.Medjaković
- Teme sastanka:
 - opis oblikovnih obrazaca
 - proučavanje git tehnologije

6. sastanak

- Datum: 4. studenog 2023.
- Prisustvovali: I.Cvijetić, F.Vitković
- Teme sastanka:
 - rad na sekveničkim dijagramima
 - proučavanje astah programa

7. sastanak

- Datum: 5. studenog 2023.
- Prisustvovali: J.Spajić, F.Vitković
- Teme sastanka:
 - dijagrami obrazaca uporabe

8. sastanak

- Datum: 11. studenog 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:
 - revizija dosad napravljenog sadržaja
 - dogovor oko dalnjeg tijeka projekta

9. sastanka

- Datum: 13. studenog 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, I.Lisica, S.Medjaković
- Teme sastanka:

- rasprava o zadnjim implementacijama vezanih za frontend i backend

10. sastanka

- Datum: 15. studenog 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Medjaković, F.Vitković, S. Gašpar
- Teme sastanka:
 - prikaz frontend implementacije
 - dogovor oko posljednjih implementacija prije prve predaje

11. sastanka

- Datum: 15. studenog 2023.
- Prisustvovali: I.Cvijetić, I.Lisica, S.Medjaković
- Teme sastanka:
 - implementacija verifikacije putem maila

12. sastanka

- Datum: 15. studenog 2023.
- Prisustvovali: E.Prpić, S.Medjaković
- Teme sastanka:
 - puštanje aplikacije u pogon

13. sastanka

- Datum: 1. prosinca 2023.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:
 - raspodjela zadataka za drugi ciklus

14. sastanka

- Datum: 20. prosinca 2023.
- Prisustvovali: I.Lisica, S.Medjaković
- Teme sastanka:
 - dogovor oko povezivanja što se do tada napravilo(životinje, akcije)

15. sastanka

- Datum: 3. siječnja 2024.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:

- Izvještaj dosad napravljenog

16. sastanka

- Datum: 4. siječnja 2024.
- Prisustvovali: I.Cvijetić, S.Medjaković,
- Teme sastanka:
 - povezivanje frontenda i backenda za akcije

17. sastanka

- Datum: 6. siječnja 2024.
- Prisustvovali: E.Prpić, F.Vitković
- Teme sastanka:
 - lokacija životinja

18. sastanka

- Datum: 9. siječnja 2024.
- Prisustvovali: I.Cvijetić, F.Vitković
- Teme sastanka:
 - ispravljanje bugova

19. sastanka

- Datum: 9. siječnja 2024.
- Prisustvovali: I.Lisica, F.Vitković
- Teme sastanka:
 - implementacija GPS-a

20. sastanka

- Datum: 10. siječnja 2024.
- Prisustvovali: I.Lisica, F.Vitković, E.Prpić
- Teme sastanka:
 - implementacija GPS-a

21. sastanka

- Datum: 12. siječnja 2024.
- Prisustvovali: E.Prpić, I.Cvijetić, S.Gašpar, I.Lisica, S.Medjaković, J.Spajić, F.Vitković
- Teme sastanka:
 - završni dogovor

22. sastanka

- Datum: 14. siječnja 2024.

- Prisustvovali: E.Prpić, F.Vitković
- Teme sastanka:
 - prezentacija projekta

23. sastanka

- Datum: 18. siječnja 2024.
- Prisustvovali: E.Prpić, J. Spajić
- Teme sastanka:
 - pregled dokumentacije

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Emil Prpić	Ivan Cvijetić	Sara Gašpar	Ivan Lisica	Sebastian Medjaković	Jure Spajić	Filip Vitković
Upravljanje projektom	20				1		
Opis projektnog zadatka	2	2	3	2	3	2	2
Funkcionalni zahtjevi	1	1	1	1	1	1	1
Opis pojedinih obrazaca	4	4	4	4	4	4	4
Dijagram obrazaca					1	3	2
Sekvencijski dijagrami		4				1	3
Opis ostalih zahtjeva					1		
Arhitektura i dizajn sustava	2		2	3		1	
Baza podataka	1				5		
Dijagram razreda	3	3				2	
Dijagram stanja						6	
Dijagram aktivnosti						6	
Dijagram komponenti						5	
Korištene tehnologije i alati						5	
Ispitivanje programskog rješenja	3	1	3		6	2	2
Dijagram razmještaja						4	

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Emil Prpić	Ivan Cvjetić	Sara Gašpar	Ivan Lisica	Sebastian Medjaković	Jure Spajić	Filip Vitković
Upute za puštanje u pogon	1					4	
Dnevnik sastajanja					1	1	
Zaključak i budući rad						5	
Popis literature						1	
Učenje tehnologija	4	40	22	15	19	15	45
Izrada baze podataka	1			3	5		
Spajanje s bazom podataka	2	2	1	5	3		
Akcija životinje backend				4			
GPS api				30			
Backend	10			5	36		
Dizajn aplikacije	2	1	10	1	1	1	1
Pregled dokumentacije	3	1	2	1	2	6	
Frontend		40					50
Frontend spajanje	5	10	40				15
Puštanje u pogon	12	2					1

Dijagrami pregleda promjena



Slika 6.1: prikaz aktivnosti na repozitoriju