

Programsko inženjerstvo

Ak. god. 2022./2023.

Čuvari pasa

Dokumentacija, Rev. 1

Grupa: Primavara

Voditelj: Antonio Lukić

Datum predaje: <dan>. <mjesec>. <godina>.

Nastavnik: Antea Šetka

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	23
3.2 Ostali zahtjevi	27
4 Arhitektura i dizajn sustava	28
4.1 Baza podataka	30
4.1.1 Opis tablica	30
4.1.2 Dijagram baze podataka	37
4.2 Dijagram razreda	38
4.3 Dijagram stanja	41
4.4 Dijagram aktivnosti	42
4.5 Dijagram komponenti	43
5 Implementacija i korisničko sučelje	44
5.1 Korištene tehnologije i alati	44
5.2 Ispitivanje programskog rješenja	45
5.2.1 Ispitivanje komponenti	45
5.2.2 Ispitivanje sustava	45
5.3 Dijagram razmještaja	46
5.4 Upute za puštanje u pogon	47
6 Zaključak i budući rad	48
Popis literature	49
Indeks slika i dijagrama	50

Dodatak: Prikaz aktivnosti grupe

51

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Mario Petek	30.10.2022.
0.2	Napisan opis projektnog zadatka	Ivan Kuzmić, Mario Petek	31.10.2022.
0.3	Napisani funkcijski zahtjevi	Ivan Kuzmić, Mario Petek	01.11.2022.
0.4.1	Napisan dio obrazaca uporabe	Ivan Kuzmić, Mario Petek	01.11.2022.
0.4.2	Ažuriranje obrazaca uporabe	Antonio Lukić, Mario Petek	02.11.2022.
0.4.3	Napisan ostatak obrazaca uporabe i popravljivanje grešaka	Ivan Kuzmić	05.11.2022.
0.4.4	Dodavanje potrebnih informacija obrascima uporabe	Ivan Kuzmić	11.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.5	Dodavanje dijagrama obrazaca uporabe i sekvencijskih dijagrama	Ivan Kuzmić, Mario Petek	14.11.2022.
0.6	Dodavanje ostalih zahtjeva, arhitekture i opisa baze podataka	Ivan Kuzmić	14.11.2022.
0.7	Dodavanje dijagrama razreda	Antonio Lukić	16.11.2022.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Ivan Kuzmić	18.11.2022.
1.1			
2.0	Konačni tekst predložka dokumentacije		

2. Opis projektnog zadatka

U današnje vrijeme veliki broj ljudi za svog ljubimca odabire psa, koji većinu vremena provodi uz svog vlasnika. No, što ako vlasnik ima neodgodivu obavezu na koju nikako ne može povesti svog psa i treba biti odsutan neko vrijeme? Kome se obratiti za pomoć, ako mu je nitko koga zna nije u mogućnosti ponuditi? Negdje u blizini se sigurno krije osoba koja bi rado preuzela privremenu brigu, no kako do nje?

Kako bi vlasnici mogli obavljati svoje obaveze bezbrižno znajući da je njihov pas zbrinut, a s druge strane ljudi koji bi rado prošetali ili nahranili psa i privremeno se igrali s njim mogli ispuniti vrijeme na koristan i zabavan način, potrebna je aplikacija preko koje bi se oni mogli povezati i pronaći rješenje za svoje brige.

Glavni cilj ovog projekta je stvaranje aplikacije "*Čuvari pasa*" koja će vlasniku psa/pasa pomoći u pronalasku osobe koja najbolje odgovara za privremenu brigu (od nekoliko sati do nekoliko tjedana ili mjeseci) o njegovom ljubimcu.

Ukratko, ideja je da vlasnik psa u aplikaciji predaje zahtjev kojim traži osobu za privremenu brigu o njegovom psu. S druge strane, osobe koje su voljne čuvati nekog psa na određeno vrijeme u aplikaciji predaju oglas u kojem navode uvjete čuvanja/brige. Na temelju zahtjeva vlasnika i uvjeta čuvara, aplikacija pronalazi najbolji odabir, koji su obje strane u mogućnosti prihvatiti ili odbiti, pa pretražujući ostale zahtjeve/oglase pronaći drugi odabir.

Prilikom pokretanja aplikacije prikazuje se početna stranica na kojoj su svi trenutno aktivni zahtjevi i oglasi s pripadajućim podacima. Pregledavanje tih oglasa omogućeno je i neregistriranim korisnicima, ali za korištenje ostalih funkcionalnosti, kao što su objava vlastitih zahtjeva/oglasa te stupanje u kontakt s objavljivačima, potrebna je registracija ili u slučaju već postojećeg računa, prijava. Registrirati se može bilo tko na način da unese osnovne podatke koji su potrebni za korištenje aplikacije, a to su: ime, prezime, e-mail adresa, korisničko ime, lozinka, broj telefona i uloga (*vlasnik/čuvvar/oboje*).

S druge strane, korisnik koji je već registriran može se prijaviti u sustav unoseći korisničko ime i lozinku. Registracijom u sustav korisniku se dodjeljuje odabrana uloga (*vlasnik/čuvvar/oboje*). On u tom slučaju može pregledavati svoje osobne po-

datke, a u nastavku su opisane funkcionalnosti i mogućnosti za pojedinu ulogu korisnika.

Korisnik (vlasnik psa):

U slučaju da je korisnik odabrao opciju vlasnika psa, on ima mogućnost kreiranja zahtjeva za čuvanje psa u aplikaciji. Pri kreiranju zahtjeva potrebno je unijeti osnovne podatke vezane za psa i čuvanje, a to su: pasmina, dob psa, potrebni period čuvanja, potrebne aktivnosti (npr. šetnja, istrčavanje, hranjenje – koliko često, potrebna količina ako se radi o hrani itd.), lokacija i broj pasa za koje je potrebno čuvanje. Osim osnovnih podataka, vlasnik može objaviti i fotografiju svog psa, kao i poželjne karakteristike čuvara (ima li iskustva u svom poslu, ima li vlastitog psa). Sve zahtjeve koje je korisnik kreirao može pregledati, urediti i brisati na pregledu *Moji zahtjevi*. Također, korisnik na pregledu *Oglas*i ima mogućnost pregleda svih oglasa koje su objavili potencijalni čuvari.

Korisnik (čuvar psa):

U slučaju da je korisnik odabrao opciju čuvara pasa, on ima mogućnost kreiranja oglasa u aplikaciji kojim nudi uslugu čuvanja. Pri kreiranju oglasa potrebno je unijeti osnovne podatke relevantne za oglas, a to su: pasmina koju želi čuvati (uz oznaku mora li nužno biti samo ta pasmina), preferirana dob psa, mogući period čuvanja (fiksno ili fleksibilno), lokacija i broj pasa koje može čuvati (fiksno ili fleksibilno). Uz osnovne podatke, korisnik može navesti i dodatne karakteristike kao što su: imam vlastitog psa i imam iskustva u ovom poslu. Sve oglase koje je korisnik kreirao može pregledati, urediti i brisati na pregledu *Moji oglasi*. Također, korisnik na pregledu *Zahtjevi* ima mogućnost pregleda svih zahtjeva koje su objavili vlasnici pasa. Čuvar psa ima dodatnu mogućnost naspram vlasnika psa, a to je da u slučaju dogovaranja točnog termina, on može urediti svoje podatke unoseći termin kada je dogovoreno čuvanje i kojeg psa u svoj kalendar. Potom te podatke sprema tako da na detaljima svog računa može vidjeti kalendar s terminima i voditi organizaciju vremena.

Uz ulogu korisnika, postoji i uloga administratora.

Administrator sustava, uz funkcionalnosti ranije opisanih korisnika, ima i neke dodatne mogućnosti kao što su: upravljanje zahtjevima/oglasima koje su kreirali vlasnik/čuvar psa, blokiranje korisnika koji narušavaju pravila sustava te dodjeljivanje administratorskih ovlasti drugom korisniku. Osobni podaci registriranog korisnika bit će vidljivi samo administratoru i koristit će se u svrhu ostvarenja kontakta između uparenih vlasnika i čuvara psa. Prije javne objave bilo kojeg zah-

tjeva/oglasa, on se šalje administratoru na uvid u posebnoj kartici koja je vidljiva samo korisnicima s administratorskim ovlastima u aplikaciji, a koji potom odlučuje hoće li taj zahtjev/oglas biti javno dostupan.

Nakon što administrator odobri stvoreni zahtjev/oglas, on postaje javno dostupan te potom korisnik ima dvije mogućnosti za pronalazak najboljeg odgovarajućeg čuvara psa/psa za čuvanje:

- Odabir *"Pronađi najbolji odabir"* kojim aplikacija na temelju pojedinih karakteristika (pasmina, dob psa, period čuvanja, udaljenost lokacija vlasnika i čuvara mora biti manja od 5 kilometara) pronalazi najpogodnijeg čuvara/psa. Ukoliko se pronađe takva osoba, korisniku s ulogom vlasnika psa dolazi obavijest u aplikaciji s najpogodnijim oglasom, a ukoliko korisnik ima ulogu čuvara psa, dolazi mu obavijest u aplikaciji s najpogodnijim zahtjevom. Korisnici imaju mogućnost prihvatanja pronađenog odabira i ako oba korisnika prihvate odabir tada im se na pregledu zahtjeva/oglasa prikazuje i kontakt drugog korisnika te se mogu dogovoriti za detalje. Ako barem jedan korisnik odbije odabir, tada se smatra da nije poželjna daljnja suradnja.
- Korisnik ručno pregledava sve zahtjeve/oglasu u aplikaciji, pronalazi onaj koji mu se najviše sviđa i odabirom istog šalje obavijest čuvaru/vlasniku psa da bi ga htio odabrati. Čuvar/vlasnik psa taj odabir prihvata te se potom prikazuje kontakt za daljnji dogovor oko detalja, ili odbija što bi značilo da jedna od strana ne želi ostvariti suradnju.

Nakon završene suradnje vlasnik psa može ocijeniti koliko je zadovoljan uslugom čuvanja. S druge strane, čuvar isto tako može ocijeniti kakav je bio pas za čuvanje. Prosječna ocjena kojom je ocjenjen čuvar pasa prikazuje se na svakom oglasu tog čuvara kako bi vlasnik psa mogao na temelju tuđih mišljenja procijeniti je li baš taj čuvar pogodan za njegovog ljubimca. Isto tako, prosječna ocjena psa biti će vidljiva na svakom zahtjevu za čuvanje kako bi potencijalni čuvar mogao procijeniti je li taj pas pogodan za čuvanje na temelju tuđih iskustava.

Aplikacija treba biti izvedena kao web aplikacija prilagođena mobilnom uređaju i tabletu kojoj će registrirani korisnici pristupati uz pomoć korisničkog imena i lozinke. Također, sustav treba podržavati rad više korisnika u stvarnom vremenu.

Aplikacija ima puno prostora za nadogradnju u budućnosti. Neke od početnih ideja nadogradnje bile bi dodavanje mogućnosti pregledavanja korisničkih računa korisnika (uz to bi se i nadogradile mogućnosti uređivanja samog korisničkog računa

- npr. moguće objavljivanje fotografija svojih ljubimaca) i slanja zahtjeva za prijateljstvo, kako bi osobe mogle ostati povezane za potencijalnu buduću suradnju. Također, moguće proširenje aplikacije bila bi implementacija sustava kojim se oglašavaju psi za udomljavanje i zbrinjavanje.

Neke funkcionalnosti ove aplikacije mogle bi se usporediti s funkcionalnostima aplikacije za pružanje usluga prijevoza pod nazivom "*Uber*". Aplikacija "*Uber*" ima funkcionalnost pronalaska najboljeg odabira, koja radi na način da uparuje vozače i korisnike s najpogodnijom lokacijskom udaljenosti, tako da korisniku u što kraćem roku bude pružena željena usluga. Po završetku vožnje korisnik također ima opciju ocijeniti koliko je bio zadovoljan vožnjom. Navedene funkcionalnosti su vrlo slične funkcionalnostima aplikacije kojom se bavi ovaj projekt.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Neregistrirani korisnik
2. Korisnik
 - (a) Vlasnik pasa
 - (b) Čuvar pasa
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) Na početnoj stranici pregledavati zahtjeve i oglase
 - (b) Odabrati željeni zahtjev/oglas i vidjeti pojedinosti tog zahtjeva/oglasa (pasmına, dob, lokacija itd.)
 - (c) Registrirati se u sustav, stvoriti novi korisnički račun za koji mu trebaju ime, prezime, korisničko ime, e-mail adresa, lozinka, broj telefona i uloga.
2. Korisnik (inicijator) može:
 - (a) Pregledavati i mijenjati osobne podatke
 - (b) Odjaviti se s korisničkog računa i prijaviti se nekim drugim računom
 - (c) Imati ulogu vlasnika pasa:
 - i. Stvarati novi zahtjev za čuvanje pasa
 - ii. Stupiti u kontakt s potencijalnim čuvarom pasa
 - iii. Spremiti podatke o svom psu
 - iv. Ocjenjivati uslugu čuvara
 - v. Pregledavati vlastite zahtjeve

(d) Imati ulogu čuvara pasa:

- i. Stvarati novi oglas za čuvanje pasa
- ii. Stupiti u kontakt s vlasnikom pasa
- iii. Termine čuvanja pasa spremati u svoj kalendar
- iv. Ocjenjivati ponašanje psa
- v. Pregledavati vlastite oglase

3. Administrator (inicijator) može:

- (a) Obavljati sve funkcionalnosti korisnika
- (b) Dodijeliti administratorsku ulogu drugim korisnicima
- (c) Odobriti ili odbaciti zahtjev/oglas
- (d) Blokirati korisnika koji narušava pravila sustava

4. Baza podataka (sudionik) može:

- (a) Pohraniti sve podatke o korisnicima i njihovim ulogama
- (b) Pohraniti sve podatke o psima
- (c) Pohraniti sve podatke o provedenim oglasima i zahtjevima
- (d) Pohraniti podatke o mogućima aktivnostima sa psima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC01 - Pregledavanje zahtjeva/oglasa

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Pregledati ponuđene oglase i zahtjeve
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Prilikom učitavanja aplikacije prikazuje se lista zahtjeva i lista oglasa
 2. Klikom na zahtjev/oglas prikazuju se dodatne informacije o tom zahtjevu/oglasu

UC02 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za korištenje svih funkcionalnosti aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju za registraciju
 2. Korisnik unosi korisničke podatke potrebne za registraciju i potvrđuje upis
 3. Korisnik dobiva povratnu informaciju o uspješnoj registraciji i preusmjerava se na početnu stranicu
 4. Korisnik se upisuje u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkih podataka u neispravnom formatu, nepodudaranje lozinki
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC03 - Prijava

- **Glavni sudionik:** Korisnik

- **Cilj:** Dobivanje pristupa svim funkcionalnostima aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je pohranjen u bazu podataka
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju za prijavu
 2. Korisnik unosi korisničko ime i lozinku te potvrđuje upis
 3. Korisnik dobiva povratnu informaciju o uspješnoj prijavi i preusmjerava se na početnu stranicu
- **Opis mogućih odstupanja:**
 - 2.a Neispravan unos korisničkog imena i/ili lozinke
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu
 2. Korisnik unosi ispravne podatke za prijavu ili odustaje od prijave

UC04 - Odjava iz sustava

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju "Odjava"
 2. Aplikacija korisnika odjavljuje s trenutnog korisničkog računa
 3. Korisnik se preusmjerava na početnu stranicu gdje mu se pruža lista zahtjeva i oglasa

UC05 - Pregled osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju "Moj račun"
 2. Aplikacija prikazuje osobne podatke korisnika

UC06 – Dodavanje pasa

- **Glavni sudionik:** Korisnik (vlasnik pasa)

- **Cilj:** Dodavanje pasa na web stranicu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu vlasnika pasa
- **Opis osnovnog tijeka:**
 1. Korisnik na padajućem izborniku odabire opciju "Moji psi"
 2. Korisnik na stranici s prikazom vlastitih pasa odabire gumb "Dodaj psa"
 3. Korisnik ispunjava osobne podatke o psu
 4. Korisnik odabire gumb "Spremi"
 5. Podaci o psu se spremaju u sustav
- **Opis mogućih odstupanja:**
 - 4.a Korisnik zaboravlja pritisnuti gumb "Spremi"
 1. Podaci o psu se ne spremaju u sustav

UC07 – Pregledavanje tuđih zahtjeva

- **Glavni sudionik:** Korisnik
- **Cilj:** Pronalaženje najprikladnijeg zahtjeva
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu čuvara
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici prolazi kroz listu tuđih zahtjeva
 2. Korisnik odabire odgovarajući zahtjev
 3. Korisnik pregledava dodatne podatke o tom zahtjevu

UC08 – Pregledavanje tuđih oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Pronalaženje najprikladnijeg oglasa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu vlasnika
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici prolazi kroz listu tuđih oglasa
 2. Korisnik odabire odgovarajući oglas
 3. Korisnik pregledava dodatne podatke o tom oglasu

UC09 – Pregledavanje vlastitih zahtjeva

- **Glavni sudionik:** Korisnik (vlasnik pasa)
- **Cilj:** Pregledavanje svojih kreiranih zahtjeva

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu vlasnika
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju "Moji zahtjevi"
 2. Korisniku se prikazuje lista zahtjeva
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nema kreiranih zahtjeva
 1. Korisniku se prikazuje prikladna poruka

UC10 – Pregledavanje vlastitih oglasa

- **Glavni sudionik:** Korisnik (čuvar pasa)
- **Cilj:** Pregledavanje svojih kreiranih oglasa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu čuvara
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju "Moji oglasi"
 2. Korisniku se prikazuje lista oglasa
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nema kreiranih oglasa
 1. Korisniku se prikazuje prikladna poruka

UC11 – Kreiranje novog zahtjeva

- **Glavni sudionik:** Korisnik (vlasnik pasa)
- **Cilj:** Objavljivanje novog zahtjeva za čuvanje pasa
- **Sudionici:** Baza podataka, administrator
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu vlasnika
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici odabire opciju za kreiranje novog zahtjeva
 2. Korisniku se otvara stranica za kreiranje zahtjeva
 3. Korisnik unosi sve tražene podatke za kreiranje zahtjeva (pasmına, dob, period čuvanja itd.)
 4. Korisnik odabire opciju za objavljivanje zahtjeva
 5. Zahtjev se šalje administratoru na uvid
- **Opis mogućih odstupanja:**
 - 4.a Korisnik nije ispravno unio sve podatke

1. Korisnika se ponovno vraća na stranicu za kreiranje zahtjeva s odgovarajućom porukom o greški

UC12 – Kreiranje novog oglasa

- **Glavni sudionik:** Korisnik (čuvar pasa)
- **Cilj:** Objavljivanje novog oglasa za čuvanje pasa
- **Sudionici:** Baza podataka, administrator
- **Preduvjet:** Korisnik je prijavljen, korisnik ima ulogu čuvara
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici odabire opciju za kreiranje novog oglasa
 2. Korisniku se otvara stranica za kreiranje oglasa
 3. Korisnik unosi sve tražene podatke za kreiranje oglasa (preferirana pasmina, preferirana dob, period čuvanja itd.)
 4. Korisnik odabire opciju za objavljivanje oglasa
 5. Oglas se šalje administratoru na uvid
- **Opis mogućih odstupanja:**
 - 4.a Korisnik nije ispravno unio sve podatke
 1. Korisnika se ponovno vraća na stranicu za kreiranje oglasa s odgovarajućom porukom o greški

UC13 – Pronalaženje najboljeg odabira čuvara pasa

- **Glavni sudionik:** Korisnik (vlasnik pasa)
- **Cilj:** Pronalaženje najprikladnijeg čuvara pasa
- **Sudionici:** Baza podataka, korisnik (čuvar pasa)
- **Preduvjet:** Korisnikov zahtjev je potvrđen od administratora
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju "Moji zahtjevi"
 2. Korisnik na listi zahtjeva pokraj željenog zahtjeva pritišće gumb "Pronalaženje najboljeg odabira"
 3. Korisnik (čuvar) dobiva obavijest sa najpogodnijim zahtjevom
 4. Korisnici prihvataju odabir
 5. Korisnici razmjenjuju detalje
- **Opis mogućih odstupanja:**
 - 4.a Korisnik (vlasnik) odbija zahtjev
 1. Daljnja suradnja između korisnika nije moguća
 - 4.b Korisnik (čuvar) odbija zahtjev

1. Daljnja suradnja između korisnika nije moguća

UC14 – Pronalaženje najboljeg odabira vlasnika pasa

- **Glavni sudionik:** Korisnik (čuvar pasa)
- **Cilj:** Pronalaženje najprikladnijeg psa za čuvanje
- **Sudionici:** Baza podataka, korisnik (vlasnik pasa)
- **Preduvjet:** Korisnikov oglas je potvrđen od administratora
- **Opis osnovnog tijeka:**
 1. Korisnik pristupa padajućem izborniku i odabire opciju "Moji oglasi"
 2. Korisnik na listi oglasa pokraj željenog oglasa pritišće gumb "Pronalaženje najboljeg odabira"
 3. Korisnik (vlasnik) dobiva obavijest sa najboljim odabirom
 4. Korisnici prihvataju odabir
 5. Korisnici razmjenjuju detalje
- **Opis mogućih odstupanja:**
 - 4.a Korisnik (vlasnik) odbija zahtjev
 1. Daljnja suradnja između korisnika nije moguća
 - 4.b Korisnik (čuvar) odbija zahtjev
 1. Daljnja suradnja između korisnika nije moguća

UC15 – Javljanje na oglas

- **Glavni sudionik:** Korisnik (vlasnik pasa)
- **Cilj:** Pronalaženje najprikladnijeg čuvara pasa
- **Sudionici:** Baza podataka, korisnik (čuvar pasa)
- **Preduvjet:** Korisnikov zahtjev je potvrđen od administratora
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici prolazi kroz listu tuđih oglasa
 2. Korisnik odabire odgovarajući oglas
 3. Korisnik pregledava dodatne podatke o tom oglasu
 4. Korisnik odabire oglas
 5. Korisniku (čuvaru) se šalje obavijest o odabiru
 6. Korisnik (čuvar) prihvaća odabir
 7. Korisnici razmjenjuju detalje
- **Opis mogućih odstupanja:**
 - 4.a Korisniku se ne sviđa oglas (ne odabire oglas)
 1. Korisnik se vraća na početnu listu

2. Korisnik nastavlja s pretraživanjem
- 6.a Korisnik (čuvar) odbija zahtjev
 1. Daljnja suradnja između korisnika nije moguća

UC16 – Javljanje na zahtjev

- **Glavni sudionik:** Korisnik (čuvar pasa)
- **Cilj:** Pronalaženje najprikladnijeg psa za čuvanje
- **Sudionici:** Baza podataka, korisnik (vlasnik pasa)
- **Preduvjet:** Korisnikov oglas je potvrđen od administratora
- **Opis osnovnog tijeka:**
 1. Korisnik na početnoj stranici prolazi kroz listu tuđih zahtjeva
 2. Korisnik odabire odgovarajući zahtjev
 3. Korisnik pregledava dodatne podatke o tom zahtjevu
 4. Korisnik odabire zahtjev
 5. Korisniku (vlasniku) se šalje obavijest o odabiru
 6. Korisnik (vlasnik) prihvaća odabir
 7. Korisnici razmjenjuju detalje
- **Opis mogućih odstupanja:**
 - 4.a Korisniku se ne sviđa zahtjev (ne odabire zahtjev)
 1. Korisnik se vraća na početnu listu
 2. Korisnik nastavlja s pretraživanjem
 - 6.a Korisnik (vlasnik) odbija zahtjev
 1. Daljnja suradnja između korisnika nije moguća

UC17 - Unos dogovorenih termina u kalendar

- **Glavni sudionik:** Korisnik (čuvar pasa)
- **Cilj:** Popunjavanje kalendara radi bolje organizacije vremena
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik(vlasnik pasa) i korisnik(čuvar pasa) su dogovorili termin
- **Opis osnovnog tijeka:**
 1. Korisnik u padajućem izborniku odabire opciju "Moj račun"
 2. Korisnik pohranjuje dogovoreni termin u kalendar
 3. Korisnik sprema kalendar
- **Opis mogućih odstupanja:**
 - 3.a Korisniku ne sprema kalendar
 1. Promjene se ne spremaju

UC18 - Ocjenjivanje usluge čuvanja

- **Glavni sudionik:** Korisnik (vlasnik pasa)
- **Cilj:** Pružiti informaciju vlasnicima pasa kakav je čuvar
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik(vlasnik pasa) i korisnik(čuvar pasa) imali su poslovni odnos
- **Opis osnovnog tijeka:**
 1. Korisnik u padajućem izborniku odabire opciju "Prošla čuvanja"
 2. Korisnik odabire željeno čuvanje
 3. Korisnik ocjenjuje iskustvo s čuvarom pasa

UC19 - Ocjenjivanje ponašanja psa

- **Glavni sudionik:** Korisnik (čuvar pasa)
- **Cilj:** Pružiti informaciju čuvarima pasa kakav je pas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik(vlasnik pasa) i korisnik(čuvar pasa) imali su poslovni odnos
- **Opis osnovnog tijeka:**
 1. Korisnik u padajućem izborniku odabire opciju "Prošla čuvanja"
 2. Korisnik odabire željeno čuvanje
 3. Korisnik ocjenjuje iskustvo s vlasnikom pasa

UC20 - Dodjeljivanje administratorske uloge korisniku

- **Glavni sudionik:** Administrator
- **Cilj:** Pružiti korisniku administratorsku ulogu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator preko padajućeg izbornika pristupa stranici za administratorsko upravljanje
 2. Administrator u listi korisnika pronalazi željenog korisnika
 3. Administrator odabranom korisniku dodjeljuje administratorsku ulogu

UC21 - Potvrđivanje stvorenog zahtjeva

- **Glavni sudionik:** Administrator

- **Cilj:** Potvrditi stvoreni zahtjev da se može objaviti
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik(vlasnik pasa) je stvorio zahtjev
- **Opis osnovnog tijeka:**
 1. Administrator preko padajućeg izbornika pristupa stranici za administratorsko upravljanje
 2. Administrator u listi "Zahtjevi u tijeku" pronalazi željeni zahtjev
 3. Administrator pritiskom na gumb "Potvrdi" potvrđuje zahtjev
 4. Zahtjev se objavljuje
- **Opis mogućih odstupanja:**
 - 3.a Administrator smatra zahtjev nevaljanim
 1. Administrator pritiskom na gumb "Odbij" odbija zahtjev
 2. Zahtjev se briše

UC22 - Potvrđivanje stvorenog oglasa

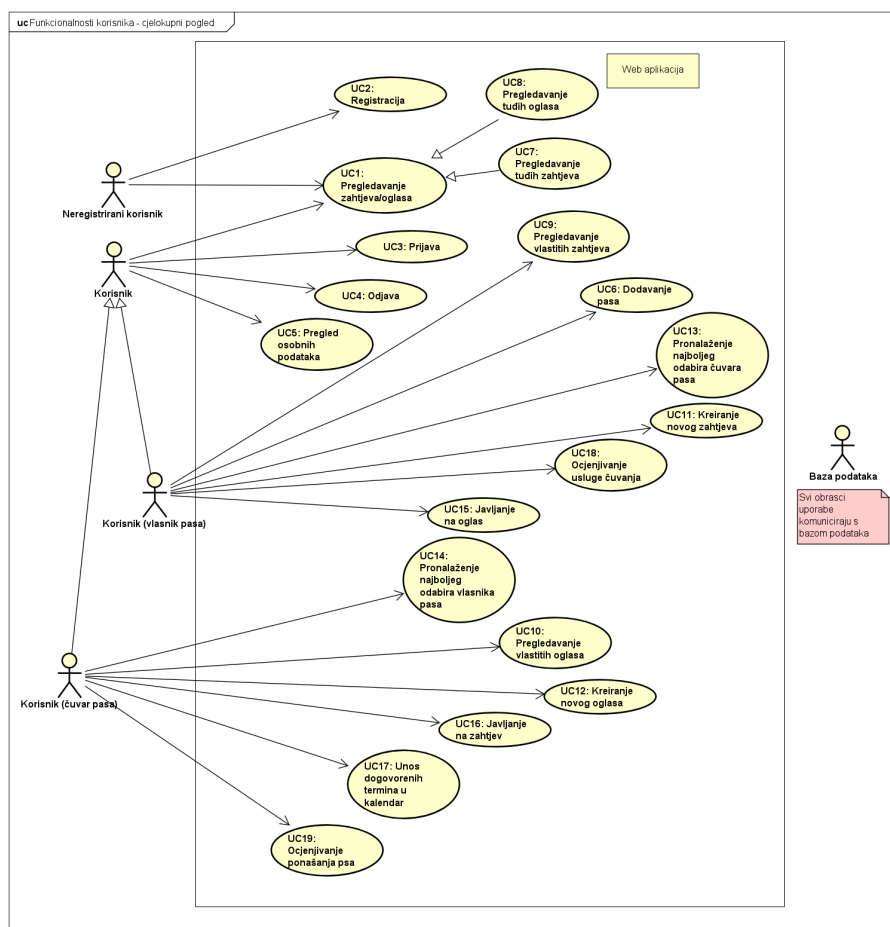
- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi stvoreni oglas da se može objaviti
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik(čuvar pasa) je stvorio oglas
- **Opis osnovnog tijeka:**
 1. Administrator preko padajućeg izbornika pristupa stranici za administratorsko upravljanje
 2. Administrator u listi "Oglasi u tijeku" pronalazi željeni oglas
 3. Administrator pritiskom na gumb "Potvrdi" potvrđuje oglas
 4. Oglas se objavljuje
- **Opis mogućih odstupanja:**
 - 3.a Administrator smatra oglas nevaljanim
 1. Administrator pritiskom na gumb "Odbij" odbija oglas
 2. Oglas se briše

UC23 - Blokiranje korisnika

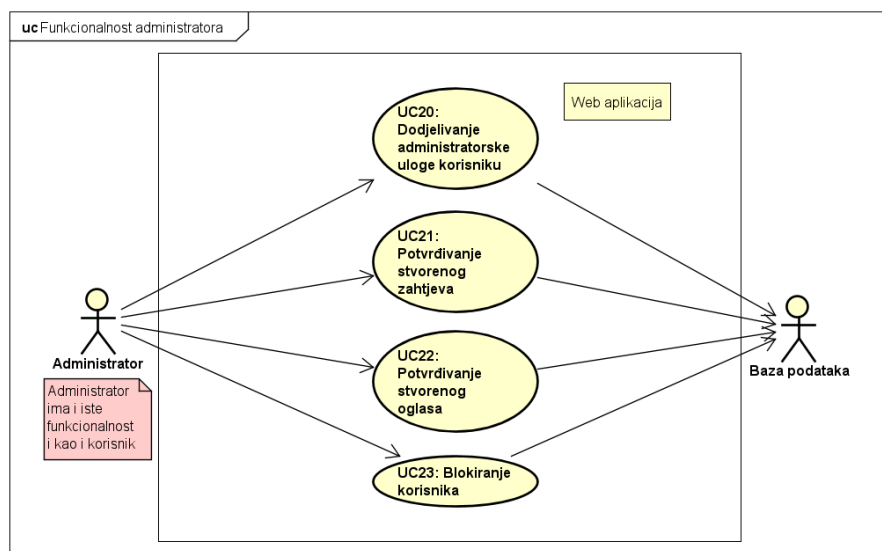
- **Glavni sudionik:** Administrator
- **Cilj:** Blokirati korisnika radi narušavanja pravila sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je narušio pravila sustava
- **Opis osnovnog tijeka:**

1. Administrator preko padajućeg izbornika pristupa stranici za administratorsko upravljanje
2. Administrator u listi korisnika pronalazi željenog korisnika
3. Administrator blokira odabranog korisnika

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost neregistiranog korisnika i korisnika (vlasnika pasa i čuvara pasa)

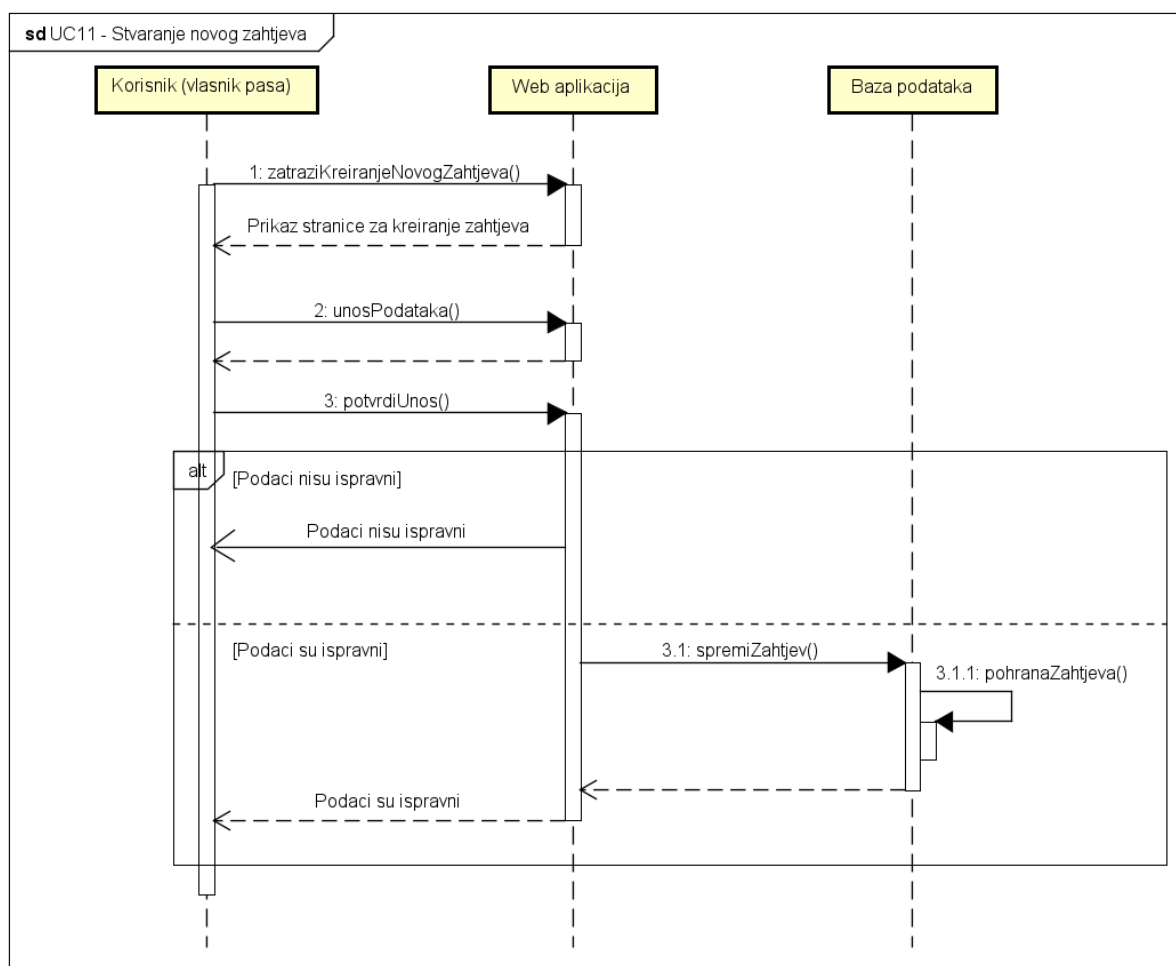


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC11 - Kreiranje novog zahtjeva

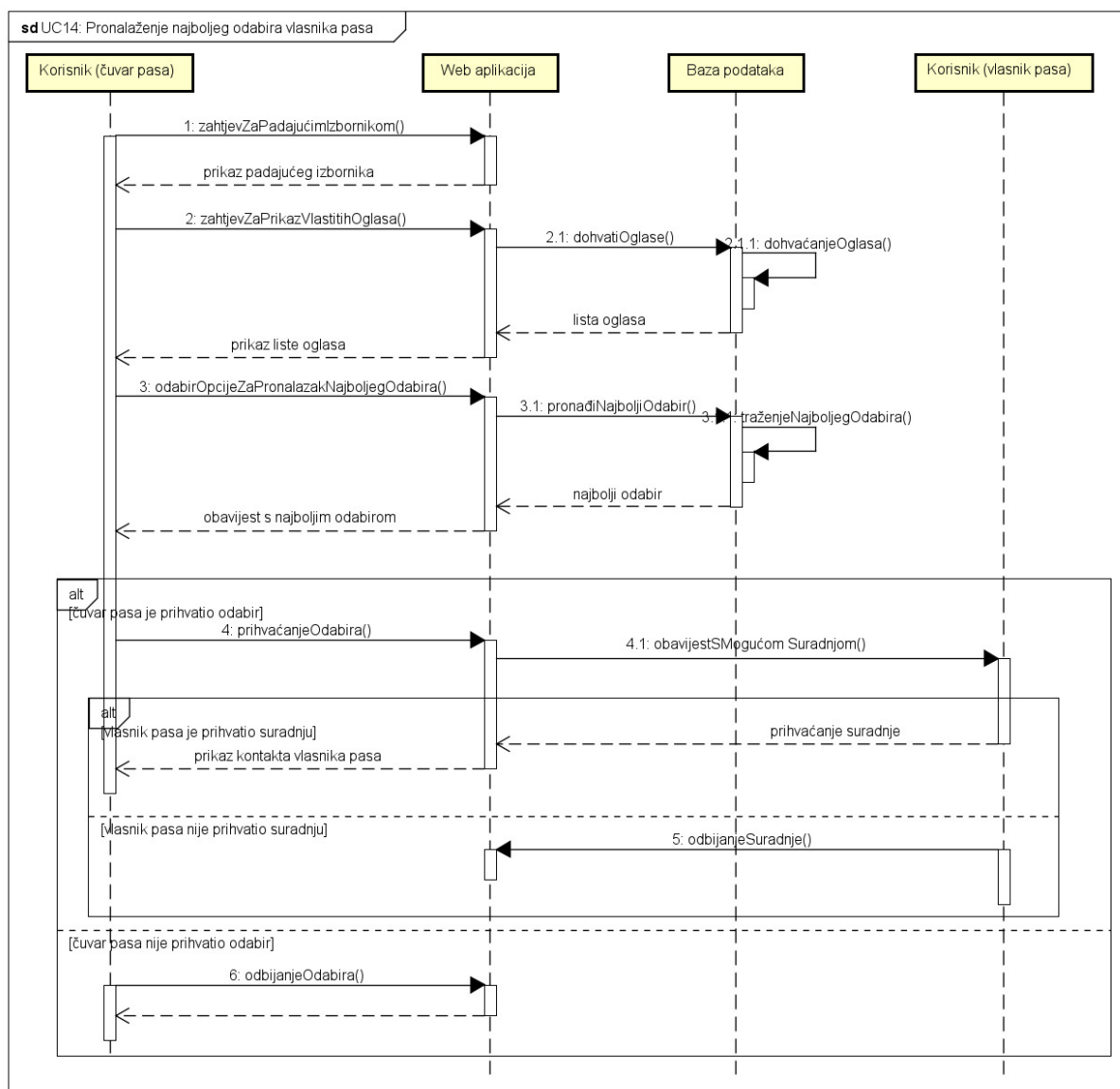
Korisnik (vlasnik pasa) na početnoj stranici odabire opciju za kreiranje novog zahtjeva. Web aplikacija mu otvara stranicu za kreiranje zahtjeva. Korisnik tamo unosi tražene podatke (pasmina, dob, period čuvanja, itd.) za kreiranje zahtjeva. Potom korisnik potvrđuje kreiranje zahtjeva. Ako korisnik nije ispravno unio podatke, prilikom potvrde web aplikacija ga ponovno usmjerava na istu stranicu s prikladnom porukom o grešci. Web aplikacija mu daje odgovor o uspješnom kreiranju zahtjeva. Zahtjev se pohranjuje u bazu podataka.



Slika 3.3: Sekvencijski dijagram za UC11

Obrazac uporabe UC14 - Pronalaženje najboljeg odabira vlasnika pasa

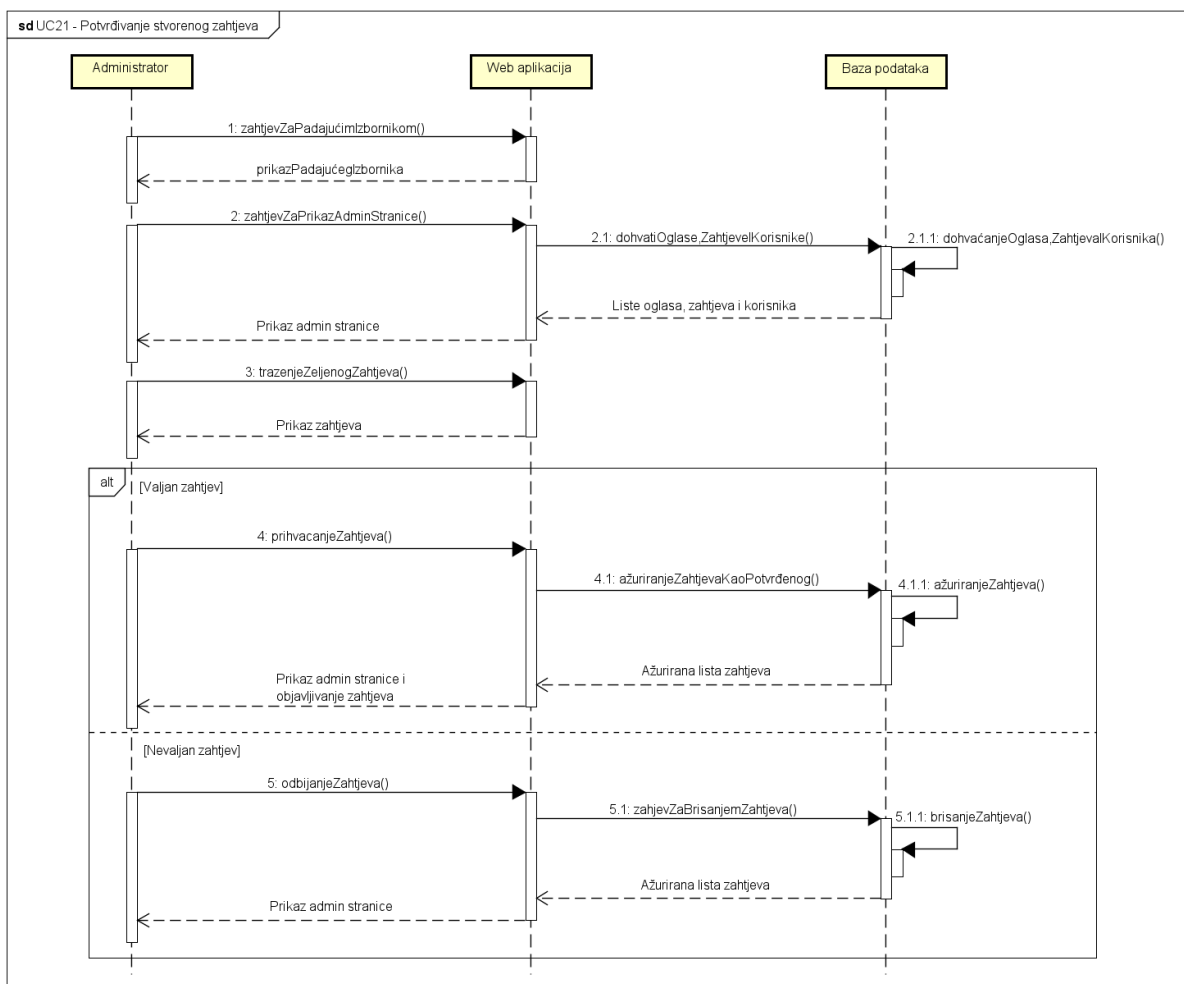
Korisnik (čuvar pasa) na početnoj stranici odabire opciju "Moji oglasi". Web aplikacija mu otvara stranicu s njegovim oglasima. Korisnik tamo pokraj željenog oglasa pritišće gumb "Pronalaženje najboljeg odabira". Web aplikacija pronalazi najboljeg vlasnika pasa i šalje ga korisniku. Korisnik ima mogućnost ga prihvatiti, ali i odbiti. Ako korisnik odbije, nema daljnje suradnje s vlasnikom pasa. Ako ga korisnik prihvati, obavijest se šalje vlasniku pasa da je došlo do najboljeg odabira. Tamo vlasnik pasa ima također opciju prihvatiti suradnju kao i odbiti ju.



Slika 3.4: Sekvencijski dijagram za UC14

Obrazac uporabe UC21 - Potvrđivanje stvorenog zahtjeva

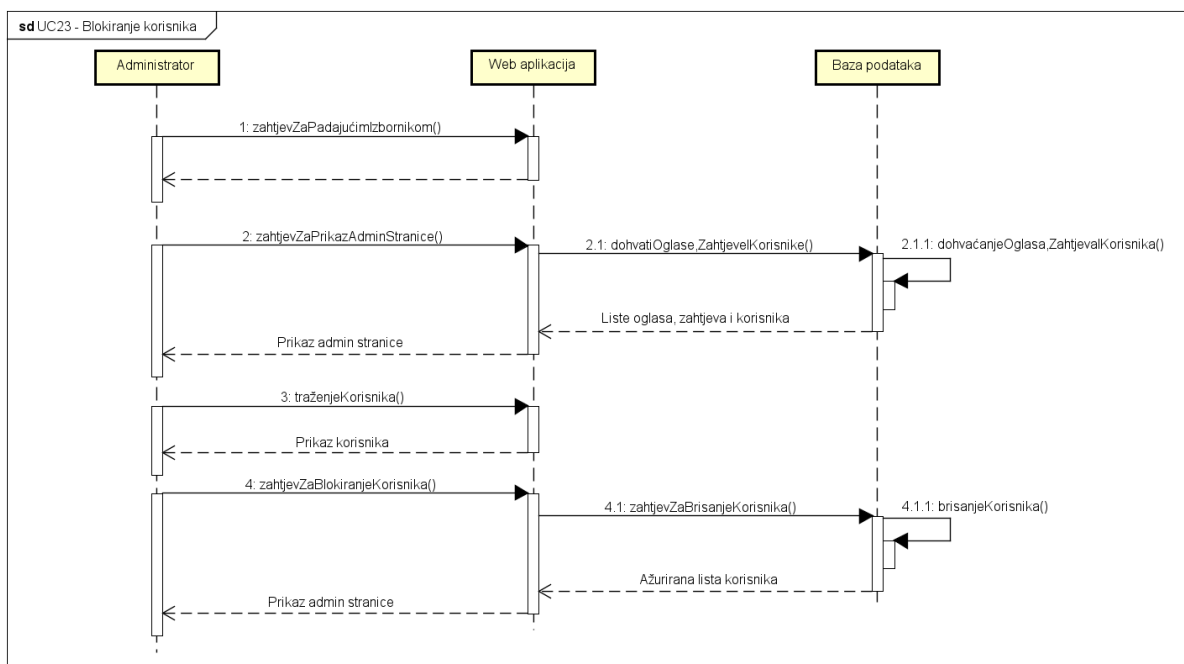
Administrator sustava na početnoj stranici odabire opciju za pristupanje stranici za administratorsko upravljanje. Web aplikacija šalje upit za potrebnim elementima (svi nepotvrđeni zahtjevi i oglasi te svi korisnici sustava) bazi podataka. Primitkom istih, web aplikacija otvara stranicu administratoru. Administrator u listi zahtjeva pronalazi željeni zahtjev te kraj njega pritišće gumb za njegove potvrđivanje. Web aplikacija šalje upit za ažuriranjem zahtjeva bazi podataka. Baza podataka ga ažurira i šalje ažuriranu listu zahtjeva web aplikaciji. Web aplikacija otvara ponovno stranicu administratoru, ali sada s ažuriranom listom zahtjeva.



Slika 3.5: Sekvencijski dijagram za UC21

Obrazac uporabe UC23 - Blokiranje korisnika

Administrator sustava na početnoj stranici odabire opciju za pristupanje stranici za administratorsko upravljanje. Web aplikacija šalje upit za potrebnim elementima (svi nepotvrđeni zahtjevi i oglasi te svi korisnici sustava) bazi podataka. Primitkom istih, web aplikacija otvara stranicu administratoru. Administrator u listi korisnika pronalazi željenog korisnika te kraj njega pritišće gumb na njegovo blokiranje. Web aplikacija šalje upit za brisanjem korisnika u bazi podataka. Baza podataka briše korisnika i šalje ažurirana listu korisnika web aplikaciji. Web aplikacija otvara ponovno stranicu administratoru, ali sada s ažuriranom listom korisnika.



Slika 3.6: Sekvencijski dijagram za UC23

3.2 Ostali zahtjevi

- Aplikacija treba biti izvedena kao web aplikacija prilagođena mobilnom uređaju i tabletu
- Aplikaciji pristupaju registrirani korisnici uz pomoć korisničkog imena i lozinke
- Sustav treba podržavati rad više korisnika u stvarnom vremenu
- Aplikaciju treba implementirati u arhitekturi klijent-poslužitelj
- Na poslužiteljskoj strani se treba koristiti programski jezik Java i radni okvir Spring Boot
- Podaci se trebaju spremati u relacijsku bazu podataka koristeći JPA
- Funkcionalnost web aplikacije se treba izložiti kroz REST Web servis
- Na klijentkoj strani treba implementirati korisničko sučelje u Web pregledniku koristeći React, koje se spaja na navedene servise

4. Arhitektura i dizajn sustava

Web aplikacija "Čuvari pasa" izvedena je u arhitekturi klijent-poslužitelj. Klijenti (korisnici) šalju zahtjeve za nekom uslugom poslužitelju (web aplikaciji). Poslužitelj odgovara sa traženom uslugom ili statusom pogreške (HTTP status 404). Glavni cilj ove arhitekture je centralizirani sustav koji dijeli informacije i usluge. Nadalje se arhitektura može podijeliti na web poslužitelj, web preglednik i bazu podataka.

Web poslužitelj je zapravo funkcionalnost web aplikacije. On komunicira s klijentom preko web preglednika. Primarna uloga mu je "nabavljanje" HTTP zahtjeva i slanje HTTP odgovora. Ovisno o zahtjevu, web poslužitelj komunicira i s bazom podataka u kojoj se nalaze potrebni podaci.

Web preglednik je softver koji služi kao sučelje između poslužitelja i klijenta i web dokumente klijentu. Njegova primarna uloga je slanje HTTP zahtjeva i primanje HTTP odgovora. Web preglednici prevode kod koji dobivaju u HTTP odgovoru te ga zatim prikazuju.

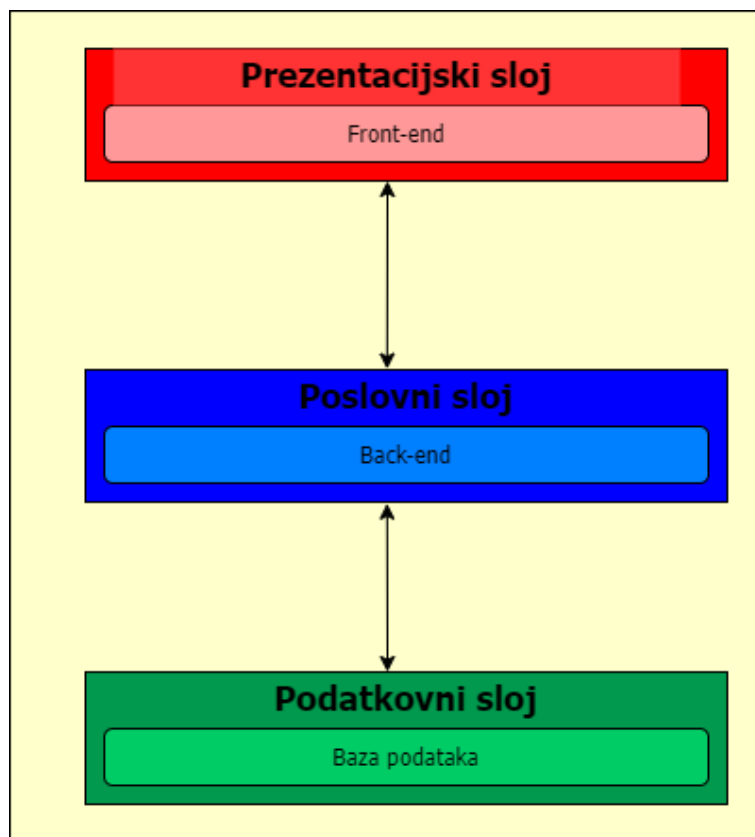
Baza podataka (podatkovni sloj) koristi se za sigurno spremanje podataka. Povezana je s web aplikacijom koja joj šalje upite za potrebnim podacima. Baza podataka je dalje opisana u poglavlju 4.1 Baza podataka.

Web aplikacija koristi arhitekturu MVC (Model-View-Controller). **Model** predstavlja dinamičke strukture podataka. U njemu se izravno upravlja podacima. **View** predstavlja prikaz podataka, a **Controller** prima zahtjeve i prosljeđuje ih gdje su dalje potrebni. MVC je implementiran u našu web aplikaciju tako da su Model i Controller komponente sadržane u back-end dijelu, a view u front-end dijelu. **Front-end** (prezentacijski sloj) je dio aplikacije zaslužan za grafičko korisničko sučelje dok se **back-end** (poslovni sloj) može poistovjetiti sa web poslužiteljem. U njemu se opisuju funkcionalnosti web aplikacije odnosno on upravlja web aplikacijom.

Backend je osim na model (repository) i controller komponente podijeljen i na service komponentu. U njoj se nalazi implementacija poslovne logike.

Front-end smo odlučili raditi u programskom jeziku JavaScript koristeći radni ok-

vir React, a back-end u programskom jeziku Java s radnoim okvirom Spring Boot. Razvojna okruženja u kojima radimo su Visual Studio Code i IntelliJ.



Slika 4.1: Dijagram arhitekture

4.1 Baza podataka

Sustav koristi relacijsku bazu podataka koja će biti implementirana u PostgreSQLu. Relacijsku bazu podataka koristimo radi lakšeg oblikovanja sustava kao stvarnog svijeta, a PostgreSQL jer smo najbolje upoznati s njim. U njemu se entiteti modeliraju kao tablice koje imaju vlastito ime i skup atributa.

Baza podataka nam je potrebna zbog njezine sigurnosti podataka, ali i brzog dohvata, pohrane i izmijene podataka koje sustav koristi za daljnje akcije. Baza podataka ovog sustava korisit sjedeće entitete:

- Role
- App User
- Breed
- Owner
- Guardian
- Dog
- Request Dog
- Request Guardian
- Request Guardians Dog
- Activity
- Request Activity
- Agreed request

4.1.1 Opis tablica

Role je entitet koji sadržava sve važne informacije o rolama aplikacije. Povezan je vezom One-To-Many s tablicom App User preko atributa userId tablice App User. Sadrži attribute: roleId i name.

Role		
roleId	INT	Jedinstven identifikator uloge korisnika
name	VARCHAR	Ime role u sustavu

App User je entitet koji sadrži sve važne informacije o korisniku aplikacije. Povezan je vezom Many-To-One s tablicom Role preko vlastitog atributa userId. Tablica Owner je specifikacija ove tablice pa je povezana s njom vezom Many-To-

One preko userIda. Isto vrijedi i za tablicu Guardian. S tablicom Dog ima vezu Many-To-One koja je isto povezana preko atributa userId. Isto vrijedi za tablice Request Dog, Request Guardian i Agreed Request. Vlastiti atributi su: userId, roleId, username, firstName, lastName, password, ratingSum, ratingCount i email.

App User		
userId	INT	Jedinstveni identifikator korisnika
roleId	INT	Jedinstven identifikator uloge korisnika (Role.roleId)
username	VARCHAR	Korisničko ime u sustavu
firstName	VARCHAR	Ime korisnika
lastName	VARCHAR	Prezime korisnika
password	VARCHAR	Hash lozinke korisnika
ratingSum	INT	Zbroj svih ocjena
ratingCount	VARCHAR	Zbroj svih ocjenjivanja
email	VARCHAR	Email pošta korisnika

Breed je entitet koji razlikuje sve pasmine u sustavu. Povezan je vezom Many-To-One s tablicom Dog preko atributa breedId. Isto vrijedi i za tablicu Request Dog. Sadrži attribute: breedId i name.

Breed		
breedId	INT	Jedinstveni identifikator pasmine
name	VARCHAR	Naziv pasmine

Owner je entitet koji sadrži sve važne informacije o vlasnicima pasa. On je specifikacija entiteta appUser pa sadrži njegove attribute. userId mu je primarni ključ kao i strani ključ koji referencira na atribut userId u entitetu App User. Povezan je vezom Many-To-One s tablicom Dog preko userIda, a isto vrijedi i s tablicom Request Guardian.

Owner		
userId	INT	Jedinstveni identifikator vlasnika pasa, (App User.userId)

Guardian je entitet koji sadrži sve važne informacije o čuvarima pasa. On je specifikacija entiteta appUser pa sadrži njegove atribute. userId mu je primarni ključ kao i strani ključ koji referencira na atribut userId u entitetu App User. Povezan je vezom Many-To-One s tablicom Request Guardian preko atributa userId. Još ima vlastite atribute: hasExperience i hasDog.

Guardian		
userId	INT	Jedinstveni identifikator čuvara pasa, (App User.userId)
hasExperience	BOOLEAN	Ima li čuvar iskustvo
hasDog	BOOLEAN	Ima li čuvar vlastitog pasa

Dog je entitet koji sadrži sve važne informacije o psima vlasnika. Povezan je vezom One-to-Many s tablicom Owner preko userIda i s tablicom Breed preko breedIda. Vezom Many-To-One povezan je s tablicom Request Guardians Dog preko atributa dogId. Još ima atribute: name, dateOfBirth, photo, ratingSum i ratingCount.

Dog		
dogId	INT	Jedinstveni identifikator pasa
name	VARCHAR	Ima li čuvar iskustvo
dateOfBirth	DATE	Datum rođenja psa
photo	VARCHAR	Slika pasa
ratingSum	INT	Suma svih ocjena
ratingCount	VARCHAR	Suma svih ocjenjivanja
userId	INT	jedinstveni identifikator vlasnika pasa, (App User.userId)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Dog		
breedId	INT	jedinstveni identifikator pasmine, (Breed.breedId)

Request Dog je entitet koji sadrži sve važne informacije o zahtjevima koje vlasnici pasa šalju. Povezan je vezom One-To-Many s tablicom Breed preko breedIda i s tablicom Guardian preko atributa userId. Vezom Many-To-One je povezan s tablicom Agreed Request preko atributa requestDogId. Još ima attribute: dogAge, dogTimeBegin, dogTimeEnd, isFlexible, location, numberOfDos, isPublished i isReviewed.

Request Dog		
requestDogId	INT	Jedinstveni identifikator zahtjeva slanog od vlasnika
dogAge	INT	Dob pasa
dogTimeBegin	TIMESTAMP	Početak čuvanja pasa
dogTimeEnd	TIMESTAMP	Završetak čuvanja pasa
isFlexible	BOOLEAN	Je li vrijeme čuvanja fleksibilno
location	VARCHAR	Lokacija stanovanja vlasnika
numberOfDos	INT	Broj pasa
isPublished	BOOLEAN	Je li zahtjev oglašen
isReviewed	BOOLEAN	Je li pas i njegov vlasnik ocjenjivan
userId	INT	jedinstveni identifikator čuvara pasa, (Guardian.userId)
breedId	INT	jedinstveni identifikator pasmine, (Breed.breedId)

Request Guardian je entitet koji sadrži sve važne informacije o oglasima koje čuvari pasa šalju. Povezan je vezom One-To-Many s tablicom Breed preko atributa breedId i s tablicom Owner preko atributa userId. Vezom Many-To-One je povezan s tablicom Agreed Request preko atributa requestGuardianId. Još ima attribute: location, numberOfDos, guardTimeBegin, guardTimeEnd, isPublished,

isReviewed, hasDog i hasExperience.

Request Guardian		
requestGuardianId	INT	Jedinstveni identifikator oglasa slanog od čuvara
location	VARCHAR	Lokacija stanovanja vlasnika
numberOfDogs	INT	Broj pasa
guardianTimeBegin	TIMESTAMP	Početak čuvanja pasa
guardianTimeEnd	TIMESTAMP	Završetak čuvanja pasa
isPublished	BOOLEAN	Je li zahtjev oglašen
isReviewed	BOOLEAN	Je li pas i njegov vlasnik ocjenjivan
hasDog	BOOLEAN	Je li čuvar ima vlastitog pasa
hasExperience	BOOLEAN	Je li čuvar ima iskustvo
userId	INT	jedinstveni identifikator vlasnika pasa, (Owner.userId)

Request Guardians Dog je entitet koji sadrži sve važne informacije o pasu čuvaru koji je objavio oglas. Te podatke će vlasnik pasa željeti vidjeti prilikom pregleda oglasa. Povezan je vezom One-To-Many s tablicom Dog preko atributa dogId. Još ima attribute requestGuardiansDogId i requestGuardianId.

Request Guardians Dog		
requestGuardiansDogId	INT	Jedinstveni identifikator pasa čuvara s oglasa
requestGuardianId	INT	jedinstveni identifikator čuvara pasa, (Request Guardian.requestGuardianId)
dogId	INT	jedinstveni identifikator pasa čuvara, (Dog.dogId)

Activity je entitet koji sadrži sve važne informacije o aktivnosti koje bi pas i čuvar radili. Povezan je vezom Many-To-One s tablicom Request Activity preko atributa activityId. Još ima atribut activityName.

Activity		
activityId	INT	jedinstveni identifikator aktivnosti
activityName	INT	jedinstveni identifikator čuvara pasa, (Request Guardian.requestGuardianId)
dogId	VARCHAR	jedinstveni identifikator pasa čuvara (Dog.dogId)

Request Activity je entitet koji sadrži sve važne informacije o zahtjevu aktivnosti pasa s njevim čuvarom. Povezan je vezom One-To-Many s tablicom Activity preko atributa activityId i s tablicom Request Guardian preko atributa requestGuardianId. Još sadrži attribute requestActivityId i feedingQuantity.

Request Activity		
requestActivityId	INT	jedinstveni identifikator zahtjeva za aktivnosti
activityId	INT	jedinstveni identifikator aktivnosti, (Activity.activityId)
requestGuardianId	VARCHAR	Naziv aktivnosti (Request Guardian.requestGuardianId)

AgreedRequest je entitet koji sadrži sve važne informacije o dogovorenom zahtjevima vlasnika i čuvara. Povezan je vezom One-To-Many s tablicom Request Guardian preko atributa requestGuardianId, s tablicom Request Dog preko atributa requestDogId i s tablicom App User preko userId. Još sadrži attribute: agreedRequestId, isAgreed, agreedTimeBegin, agreeTimeEnd i initiatorUserId.

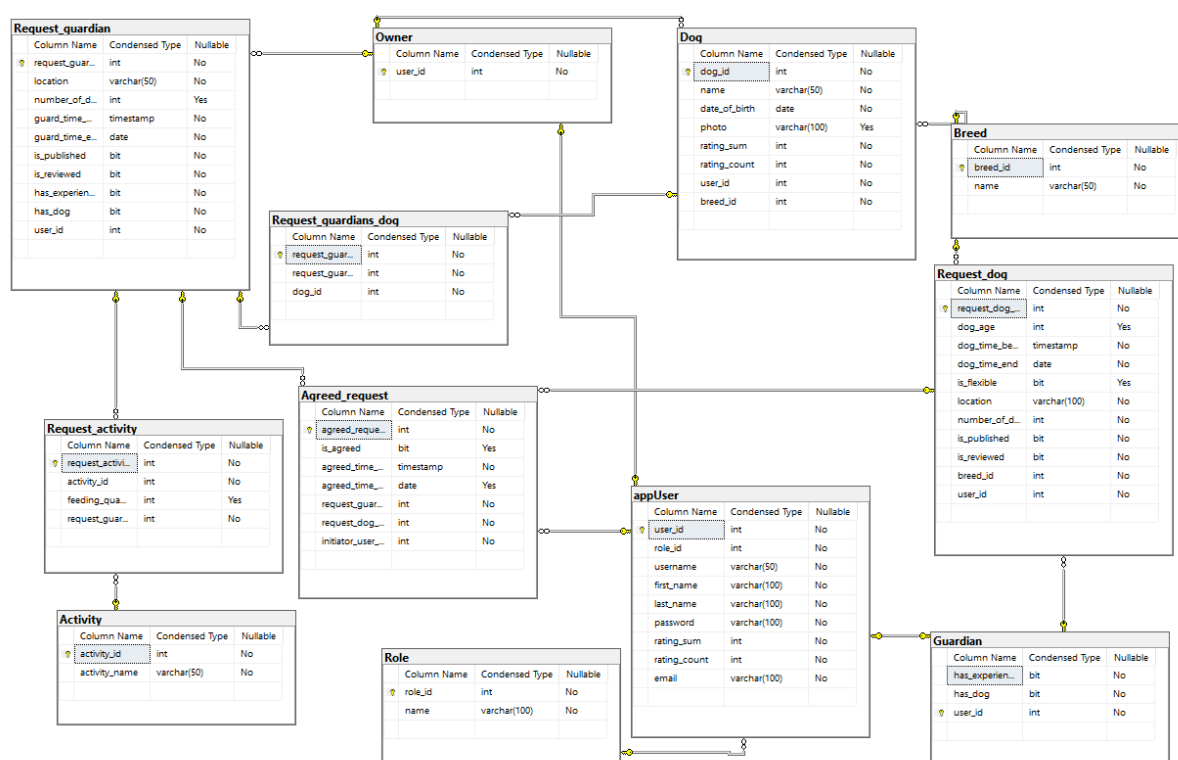
AgreedRequest		
agreedRequestId	INT	Jedinstveni identifikator dogovorenih zahtjeva
isAgreed	BOOLEAN	Je li postignut dogovor
agreedTimeBegin	TIMESTAMP	Dogovoren početak čuvanja
agreeTimeEnd	TIMESTAMP	Dogovoren kraj čuvanja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

AgreedRequest		
requestGuardianId	INT	Jedinstveni identifikator oglasa čuvara (Request Guardian.requestGuardianId)
requestDogId	INT	Jedinstveni identifikator zahtjeva vlasnika (Request Dog.requestDogId)
initiatorUserId	INT	Jedinstveni identifikator korisnika koji je započeo interakciju (App User.userId)

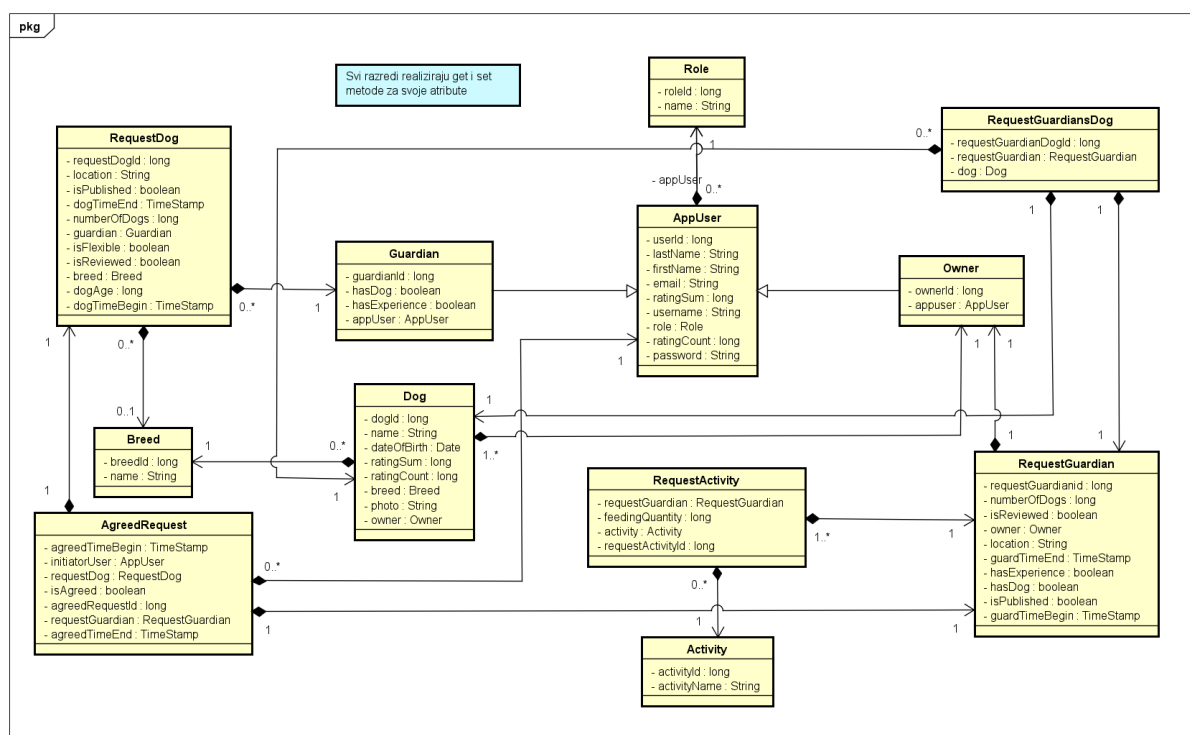
4.1.2 Dijagram baze podataka



Slika 4.2: E-R dijagram baze podataka

4.2 Dijagram razreda

Na slici 4.2 prikazani su razredi koji pripadaju backend dijelu MVC arhitekture. Zbog lakše organizacije, razredi su podijeljeni logički po pravu pristupa metodama određenih aktera, kako bi se smanjila prenapučenost 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.



Slika 4.3: Dijagram razreda - dio Models

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke. Razred AppUser predstavlja korisnika web aplikacije koji se može registrirati unoseći potrebne informacije. Odabравši pritom ulogu (razred Role), on se priključuje web aplikaciji kao vlasnik pasa (razred Owner) i/ili kao čuvar pasa (razred Guardian). Administrator je korisnik aplikacije koji ima sve mogućnosti App Usera. Vlasnik pasa može dodati vlastitog psa u sustav (razred Dog). Vlasnik i čuvar mogu tražiti čuvara za svog pasa (razred RequestGuardian) odnosno psa za čuvanje (razred RequestDog). Vlasnik može još pregledati ima li čuvar pasa (razred Request-

GuardiansDog) i tražiti koju aktivnost će čuvar raditi sa psom (razredi Activity i RequestActivity). Dogovor vlasnika i čuvara predstavlja razred AgreedRequest.

dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

5.2 Ispitivanje programskog rješenja

dio 2. revizije

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

*Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.*

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

dio 2. revizije

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranog korisnika i korisnika (vlasnika pasa i čuvara pasa)	21
3.2	Dijagram obrasca uporabe, funkcionalnost administratora	22
3.3	Sekvencijski dijagram za UC11	23
3.4	Sekvencijski dijagram za UC14	24
3.5	Sekvencijski dijagram za UC21	25
3.6	Sekvencijski dijagram za UC23	26
4.1	Dijagram arhitekture	29
4.2	E-R dijagram baze podataka	37
4.3	Dijagram razreda - dio Models	38

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 20. listopada 2022.
- Prisustvovali: Mario Petek, Ivan Kuzmić, Lovro Malojčić, Antonio Lukić, Ivan Kapusta, Eugen Preglej, Sven Leko
- Teme sastanka:
 - sastanak s asistentom i demonstratorom
 - diskusija o radu na projektu
 - odabir tehnologije

2. sastanak

- Datum: 23. listopada 2022.
- Prisustvovali: Mario Petek, Ivan Kuzmić, Lovro Malojčić, Antonio Lukić, Ivan Kapusta, Eugen Preglej, Sven Leko
- Teme sastanka:
 - sastavljanje svih ekrana
 - općeniti izgled

3. sastanak

- Datum: 27. listopada 2022.
- Prisustvovali: Mario Petek, Antonio Lukić, Lovro Malojčić
- Teme sastanka:
 - sastanak s asistenticom
 - pojašnjenje zadatka
 - poboljšanje baze

4. sastanak

- Datum: 12. studenoga 2022.
- Prisustvovali: Mario Petek, Ivan Kuzmić, Antonio Lukić, Ivan Kapusta, Eugen Preglej, Sven Leko
- Teme sastanka:

- pregled dosadašnjega rada
- prijenos međusobnog znanja ostalim članovima
- definiranje završnog rada za prvu verziju

5. sastanak

- Datum: 17. studenoga 2022.
- Prisustvovali: Mario Petek, Ivan Kuzmić, Antonio Lukić, Ivan Kapusta, Eugen Preglej, Sven Leko
- Teme sastanka:
 - prezentiranje početne stranice
 - login
 - logout
 - deploy

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Antonio Lukić	Ivan Kapusta	Ivan Kuzmić	Mario Petek	Lovro Malojčić	Sven Leko	Eugen Preglej
Upravljanje projektom	15	2	2	2	2	2	2
Opis projektnog zadatka	0	0	1	1	0	0	0
Funkcionalni zahtjevi	0	0	3	3	0	0	0
Opis pojedinih obrazaca	0	0	1	1	0	0	0
Dijagram obrazaca	0	0	2	2	0	0	0
Sekvencijski dijagrami	0.5	0	2	2	0	0	0
Opis ostalih zahtjeva	0	0	0.5	0.5	0	0	0
Arhitektura i dizajn sustava	0	0	3	0	0	0	0
Baza podataka	0	0	4	0	0	0	0
Izrada baze podataka	4	0	0	0	0	0	0
Dijagram razreda	3	0	1	0	0	0	0
Dijagram stanja	0	0	0	0	0	0	0
Dijagram aktivnosti	0	0	0	0	0	0	0
Dijagram komponenti	0	0	0	0	0	0	0
Korištene tehnologije i alati	0	0	0	0	0	0	0
Ispitivanje programskog rješenja	0	0	0	0	0	0	0
Dijagram razmještaja	0	0	0	0	0	0	0

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Antonio Lukić	Ivan Kapusta	Ivan Kuzmić	Mario Petek	Lovro Malojčić	Sven Leko	Eugen Preglej
Upute za puštanje u pogon	0	0	0	0	0	0	0
Dnevnik sastajanja	2	0	0.5	0.5	0	0	0
Zaključak i budući rad	0	0	0	0	0	0	0
Popis literature	0	0	0.5	0.5	0	0	0
Izrada početne stranice	0	0	0	0	0	0	15
Spajanje s bazom podataka	0.5	3	0	0	0	3	0
Setup back-enda	5	2	0	2	5	2	0
Setp front-enda	0	1	0	0	8	1	8
Deploy	2	20	0	0	0	20	0

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.