

Programsko inženjerstvo ak.god 2025./2026

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Projekt – PawPal

Tim: <TG13.2>

PawPal

Nastavnik: Vlado Sruk

Članovi tima:

Ema Zidar – voditeljica tima, frontend & dizajn

Jakov Ramljak – frontend & dizajn

Nina Zamberlin – baza podataka & dizajn

Lana Šapić – baza podataka & backend

Luka Zadro – backend developer

Leonard Zadro – backend & autorizacija (OAuth 2.0)

Viktor Wilder – frontend developer

Poveznica za GitHub projekta:

<https://github.com/EmaZidar/PsiciUOphodnji>

Upute za korištenje:

Aplikaciji PawPal može se pristupiti preko web preglednika. Za sve funkcionalnosti potrebno je prijaviti se. Nakon prijave možete dodavati i pregledavati objave o ljubimcima.

Izvorni kod aplikacije:

Kod aplikacije bit će podijeljen na backend folder i na folder frontend, prema trenutnom planu.

Opis projektnog zadatka

Sve više vlasnika pasa suočava se s nedostatkom vremena zbog ubrzanog tempa života i poslovnih obaveza. Kako bi se olakšalo pronalaženje pouzdanih i provjerenih šetača pasa, potrebno je razviti web platformu koja vlasnicima omogućava da jednostavno pronađu šetače u svojoj blizini, rezerviraju termin i obave plaćanje usluge putem interneta. Aplikacija će služiti kao posrednik između vlasnika pasa i šetača, osiguravajući sigurno, jednostavno i transparentno korištenje.

Platforma omogućuje registraciju korisnika u dvije osnovne uloge – **vlasnik psa i šetač pasa**. Vlasnici će koristiti aplikaciju besplatno, dok šetači plaćaju mjesecnu ili godišnju članarinu za korištenje platforme, koju će moći podmiriti putem **PayPal-a** ili **kreditnih kartica** korištenjem vanjskog platnog servisa. Registracija i prijava korisnika bit će ostvarene korištenjem **vanjskih servisa za autentifikaciju** poput **OAuth 2.0**, radi sigurnosti i jednostavnosti.

Vrste korisnika

Vlasnici pasa

Vlasnici imaju mogućnost kreiranja profila za svog ljubimca, gdje unose osnovne informacije – ime, pasminu, starost, zdravstvene napomene (npr. lijekovi, alergije), razinu energije i socijalizaciju s drugim psima. Svaki vlasnik može registrirati više pasa te odabrati tip šetnje (individualna ili grupna). Vlasnici mogu pretraživati šetače prema lokaciji, cijeni i ocjenama drugih korisnika, rezervirati termine te odabrati način plaćanja (gotovina, PayPal, kreditna kartica).

Šetači pasa

Šetači imaju javne profile s osnovnim informacijama: ime i prezime, lokacija djelovanja, kontakt telefon, fotografije i popis dostupnih termina šetnji. Za svaki termin šetnje definira se tip (individualna ili grupna), cijena i trajanje.

Na profilu su vidljive i ocjene te recenzije prethodnih klijenata. Nakon što vlasnik rezervira termin, šetač dobiva obavijest i može prihvati ili odbiti zahtjev.

Administratori sustava

Administratori upravljaju korisnicima i postavljaju cijenu članarine za šetače. Također imaju pristup sustavu za nadzor i moderiranje recenzija i profila korisnika.

Potencijalna korist projekta

Ova aplikacija ima potencijal unaprijediti način na koji vlasnici pasa pronalaze i komuniciraju sa šetačima, donoseći niz praktičnih i društvenih koristi:

- **Povećava sigurnost i povjerenje:** svi šetači prolaze proces verifikacije, a sustav ocjena i recenzija omogućuje vlasnicima da donešu odluku na temelju iskustava drugih korisnika.

- **Olakšava organizaciju svakodnevnih obaveza:** vlasnici pasa mogu brzo pronaći slobodnog šetača u svojoj blizini i rezervirati termin prema vlastitom rasporedu.
 - **Podržava profesionalne šetače:** šetačima omogućuje jednostavnu promociju njihovih usluga i transparentan sustav plaćanja.
 - **Pomiči razvoj zajednice:** aplikacija povezuje vlasnike i šetače pasa unutar lokalne zajednice, čime doprinosi boljoj socijalnoj povezanosti i brizi za kućne ljubimce.
-

Postojeća slična rješenja

Rover

Jedna od najpoznatijih aplikacija slične namjene je **Rover**, koja povezuje vlasnike i pružatelje usluga šetnje, čuvanja i njege pasa. Rover nudi sustav rezervacija, ocjena, recenzija i sigurnih plaćanja. Naša aplikacija razlikuje se po tome što se fokusira isključivo na **usluge šetnje pasa**, s lokaliziranim pristupom i jednostavnijim korisničkim iskustvom, te integracijom vanjskih servisa poput **Google Kalendara** i **FreeChat** komunikacije.

Wag!

Aplikacija **Wag!** također omogućava pronalaženje šetača, ali primarno cilja tržište SAD-a. Naša platforma nudi **lokalizirano rješenje** za europsko tržište (npr. zemlje regije), uz podršku više jezika, fleksibilne metode plaćanja i transparentniji sustav članarina za šetače.

Skup korisnika zainteresiran za ostvareno rješenje

- **Vlasnici pasa** koji traže pouzdane šetače u svojoj okolini i žele siguran način rezervacije i plaćanja usluge.
 - **Šetači pasa**, profesionalci i ljubitelji životinja koji žele dodatno zaraditi i oglašavati svoje usluge.
 - **Administratori platforme** koji održavaju sustav, moderiraju sadržaj i prate aktivnosti korisnika.
-

Opseg projektnog zadatka

Projekt uključuje razvoj svih potrebnih tehničkih, dizajnerskih i sigurnosnih komponenti aplikacije, uključujući:

- Registraciju i prijavu korisnika putem vanjskih servisa (**OAuth 2.0**).
- Izradu profila za vlasnike, pse i šetače s relevantnim informacijama.
- Sustav rezervacija termina, uz integraciju s vanjskim servisima kalendara (**Google Calendar**).
- Mogućnost komunikacije između vlasnika i šetača putem besplatnog web chat servisa (**FreeChat**).

- Implementaciju sustava plaćanja putem vanjskih servisa (**PayPal**, kreditne kartice).
 - Sustav ocjena i recenzija za procjenu kvalitete usluga.
 - Administratorsko sučelje za upravljanje korisnicima i članarinama.
 - Testiranje funkcionalnosti aplikacije i provjeru sigurnosti korisničkih podataka.
-

Moguće nadogradnje projekta

U budućnosti, aplikacija bi se mogla proširiti funkcionalnostima poput:

- Uvođenja **sustava nagrađivanja** za najaktivnije korisnike i najviše ocijenjene šetače.
 - Implementacije **lokacijskog praćenja u stvarnom vremenu**, kako bi vlasnici mogli vidjeti rutu šetnje svog psa.
 - Dodavanja **osiguranja za kućne ljubimce** i mogućnosti hitnog kontakta u slučaju incidenta.
 - Integracije **napredne analitike**, koja bi pomogla šetačima u praćenju zarade, broja klijenata i omiljenih ruta.
-

Cilj izrade projekta

Cilj projekta nije samo razvoj funkcionalne i korisne aplikacije, već i stjecanje praktičnih znanja u razvoju softverskih rješenja. Tijekom izrade projekta tim će naučiti koristiti suvremene tehnologije za autentifikaciju, integraciju vanjskih servisa, rad s bazama podataka i sigurnosne prakse.

Osim tehničkog dijela, projekt potiče **timski rad, organizaciju i komunikaciju** među članovima tima. Kroz suradnju na stvarnom problemu, članovi tima razvijaju svoje sposobnosti rješavanja problema, planiranja i međusobnog pomaganja. Na taj način projekt nije samo tehnički izazov, već i prilika za profesionalni i osobni razvoj kroz zajednički trud i stvaranje konkretne vrijednosti za zajednicu.

Analiza zahtjeva

U ovom poglavlju detaljno razraditi postavljene zahtjeve Funkcionalni zahtjevi: Opis funkcionalnosti aplikacije (npr., registracija korisnika, prijava slučajeva, pregled statusa pomoći). Nefunkcionalni zahtjevi: npr. Zahtjevi vezani uz performanse, sigurnost, skalabilnost i korisničko sučelje. Zahtjeve prikazati standardnim formatom dobro strukturirane tablice zahtjeva koja u SRS dokumentu uobičajeno uključuje stupce: ID zahtjeva, Opis, Prioritet, Izvor i Kriteriji prihvaćanja (inačica). Primjer:

Funkcionalni zahtjevi

Tablica 2.01: funkcionalni zahtjevi

ID	Naziv zahtjeva	Prioritet	Izvor	Opis / Kriteriji prihvaćanja
F-000	Registracija korisnika	Visok	zahtjev dionika	Korisnik se može registrirati unosom osobnih podataka (ime, prezime, broj mobitela, lozinka) ili putem vanjskog autentifikacijskog servisa.
F-001	Prijava korisnika	Visok	zahtjev dionika	Korisnik se prijavljuje pomoću registriranih vjerodajnica. Sustav provjerava autentičnost i usmjerava korisnika na početnu stranicu prema njegovoj ulozi.
F-002	Odabir uloge korisnika	Visok	zahtjev dionika	Tijekom prve registracije korisnik odabire ulogu: šetač ili vlasnik psa. Ova postavka određuje dostupne funkcionalnosti u aplikaciji.
F-003	Main page	Nizak	zahtjev dionika	Neprijavljeni i neregistirani korisnik kad otvorí stranicu vidi main page s osnovnim informacijama o aplikaciji.
F-004	Pregled vlastitog profila	Srednji	zahtjev dionika	Korisnik može pregledati svoj profil, uključujući osobne podatke, ocjene i aktivnosti.
F-005	Uređivanje vlastitih podataka	Srednji	zahtjev dionika	Korisnik može mijenjati svoje podatke (ime, kontakt, slike). Promjene se odmah pohranjuju.
F-006	Pregled tuđih profila	Srednji	zahtjev dionika	Korisnici mogu pregledavati javne profile drugih korisnika (npr. vlasnik gleda profil šetača). Profil prikazuje ocjenu, recenzije i dostupne termine.
F-007	Pregled svih korisnika (administrator)	Nizak	postojeći sustav	Administrator može vidjeti popis svih registriranih korisnika, uključujući aktivne, neaktivne i blokirane račune.
F-008	Brisanje korisnika (administrator)	Nizak	postojeći sustav	Administrator može ukloniti korisnika iz sustava. Svi povezani podaci i rezervacije se brišu.

ID	Naziv zahtjeva	Prioritet Izvor	Opis / Kriteriji prihvaćanja
F-009	Uključivanje obavijesti Srednji	zahtjev dionika	Korisnik se može preplatiti na obavijesti (npr. o novim šetačima, prihvaćenim rezervacijama, promjenama termina).
F-010	Plaćanje članarine (šetač)	Visok	zahtjev dionika Šetači plaćaju članarinu putem vanjskog servisa (PayPal, kartica). Sustav pohranjuje status članarine i datum isteka.
F-011	Upravljanje cijenom članarine (administrator)	Srednji	zahtjev dionika Administrator može definirati ili izmijeniti cijenu članarine i učestalost naplate (mjesečna/godišnja).
F-012	Plaćanje šetnje	Visok	zahtjev dionika Vlasnik psa plaća šetnju gotovinom, karticom ili putem PayPala. Sustav bilježi status uplate.
F-013	Rezervacija termina šetnje	Visok	zahtjev dionika Vlasnik odabire termin u kalendaru šetača, unosi specifikacije šetnje (datum, vrijeme, trajanje, tip, adresa, napomene) i potvrđuje rezervaciju.
F-015	Slanje obavijesti šetaču	Srednji	zahtjev dionika Nakon što vlasnik rezervira termin, šetač prima automatsku obavijest putem aplikacije ili e-maila.
F-016	Prihvaćanje ili odbijanje šetnje	Srednji	zahtjev dionika Šetač može prihvatiti ili odbiti rezervaciju. Sustav o tome obaveštava vlasnika.
F-017	Dopisivanje u chatu	Srednji	zahtjev dionika Nakon potvrde šetnje, vlasnik i šetač mogu komunicirati putem integriranog chata. Poruke se pohranjuju na serveru.
F-018	Filtriranje i pretraživanje šetača	Srednji	zahtjev dionika Vlasnik može filtrirati šetače prema lokaciji, cijeni i prosječnoj ocjeni te sortirati rezultate po željenom kriteriju.
F-019	Dodavanje profila psa	Srednji	zahtjev dionika Vlasnik može dodati profil psa s podacima: ime, pasmina, starost, energija, zdravstvene napomene, socijalizacija, poslastice.
F-021	Pregled šetača	Srednji	zahtjev dionika Vlasnik može pregledati sve dostupne šetače u svojoj regiji s prikazom ocjene, cijene i dostupnosti.
F-022	Odabir specifikacija šetnje	Srednji	zahtjev dionika Tijekom rezervacije vlasnik odabire tip šetnje (individualna/grupna), trajanje, početnu adresu i napomene.

ID	Naziv zahtjeva	Prioritet	Izvor	Opis / Kriteriji prihvaćanja
F-023	Otkazivanje šetnje	Nizak	zahtjev dionika	Vlasnik može otkazati rezervaciju najkasnije 24 sata prije početka. Šetač prima obavijest o otkazivanju.
F-024	Ostavljanje recenzije	Srednji	zahtjev dionika	Nakon završene šetnje vlasnik može ostaviti ocjenu (1–5), komentar i fotografije. Prosječna ocjena prikazuje se na profilu šetača.
F-025	Objavljivanje termina šetnje	Srednji	zahtjev dionika	Šetač može u svoj kalendar dodati novi termin kada je dostupan
F-026	Pregled "Mojih šetnji"	Nizak	zahtjev dionika	Vlasnik na svom profilu može vidjeti "Moje šetnje" gdje su prikazane sve prošle šetnje koje može ocijeniti i sve dogovorene buduće šetnje koje može otkazati

Ostali zahtjevi

Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju kako se sustav treba ponašati i koja ograničenja treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...) ... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.

Zahtjevi za performanse

Tablica 2.02: Nefunkcionalni zahtjevi za performanse

ID zahtjeva	Opis	Prioritet
NF-1.1	Aplikacija mora učinkovito upravljati do 200 korisnika.	Srednji
NF-1.2	Sustav treba omogućiti brzo pretraživanje objava i tagova, max 5 sekundi.	Srednji

Zahtjevi za iskustvo korisnika

Tablica 2.03: Nefunkcionalni zahtjevi za iskustvo korisnika

ID zahtjeva	Opis	Prioritet
NF-2.1	Korisničko sučelje treba biti intuitivno i jednostavno za korištenje za sve tipove korisnika.	Visok
NF-2.2	Korisnici trebaju moći navigirati kroz aplikaciju s minimalnim brojem klikova.	Nizak

Zahtjevi za održavanje

Tablica 2.04: Nefunkcionalni zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-3.2	Sustav treba imati preciznu dokumentaciju.	Visok
NF-3.3	Sustav treba biti opisan putem dokumenta oblikovanja /SRS/.	Visok
NF-3.4	Sustav treba biti popraćen "Priručnikom za rad" koji opisuje pravilnu upotrebu sustava.	Nizak
NF-3.5	Sustav treba imati "Plan implementacije" za pravilno postavljanje sustava.	Visok

Zahtjevi za sigurnost i skalabilnost

Tablica 2.05: Nefunkcionalni zahtjevi za sigurnost i skalabilnost

ID zahtjeva	Opis	Prioritet
NF-4.1	Aplikacija treba osigurati sigurnost privatnih podataka korisnika.	Visok
NF-4.2	Aplikacija treba biti skalabilna i podržavati proširenje za nove lokacije, korisničke grupe i objave.	Visok
NF-4.3	Sustav treba omogućiti korištenje na različitim uređajima.	Srednji
NF-4.4	Sustav treba omogućiti jednostavno dodavanje novih funkcionalnosti i integraciju s vanjskim sustavima.	Srednji

Dionici

Navesti dionike koji imaju interes u ovom sustavu ili su nositelji odgovornosti. To su prije svega korisnici, ali i administratori sustava, naručitelji, razvojni tim. Navesti aktore koji izravno koriste ili komuniciraju sa sustavom. Oni mogu imati inicijatorsku ulogu, tj. započinju određene procese u sustavu ili samo sudioničku ulogu, tj. obavljaju određeni posao. Za svakog aktora navesti funkcionalne zahtjeve koji se na njega odnose.

Aktor	Uloga	Funkcionalnosti	ID zahtjeva
Neprijavljeni korisnik	Inicijator	Pregled main page	F-003
Vlasnik psa	Inicijator	Registracija i prijava	F-000, F-001, F-002
Vlasnik psa	Inicijator	Pregled i uređivanje vlastitog profila	F-004, F-005

Aktor	Uloga	Funkcionalnosti	ID zahtjeva
Vlasnik psa	Inicijator	Pregled tuđih profila i šetača	F-006, F-018, F-021
Vlasnik psa	Inicijator	Upravljanje profilom psa	F-019, F-020
Vlasnik psa	Inicijator	Rezervacija i plaćanje šetnje	F-012, F-013, F-022, F-023
Vlasnik psa	Inicijator	Komunikacija i recenzije	F-015, F-017, F-024
Vlasnik psa	Inicijator	Upravljanje obavijestima	F-009
Šetač pasa	Inicijator / Sudionik	Registracija i prijava	F-000, F-001, F-002
Šetač pasa	Sudionik	Pregled i uređivanje vlastitog profila	F-004, F-005
Šetač pasa	Sudionik	Plaćanje članarine	F-010
Šetač pasa	Sudionik	Upravljanje rezervacijama	F-015, F-016
Šetač pasa	Sudionik	Komunikacija s vlasnikom psa	F-017
Administrator	Sudionik	Upravljanje korisnicima	F-007, F-008
Administrator	Sudionik	Upravljanje članarinom	F-011
Administrator	Sudionik	Nadzor i pohrana podataka	F-014

Specifikacija zahtjeva sustava

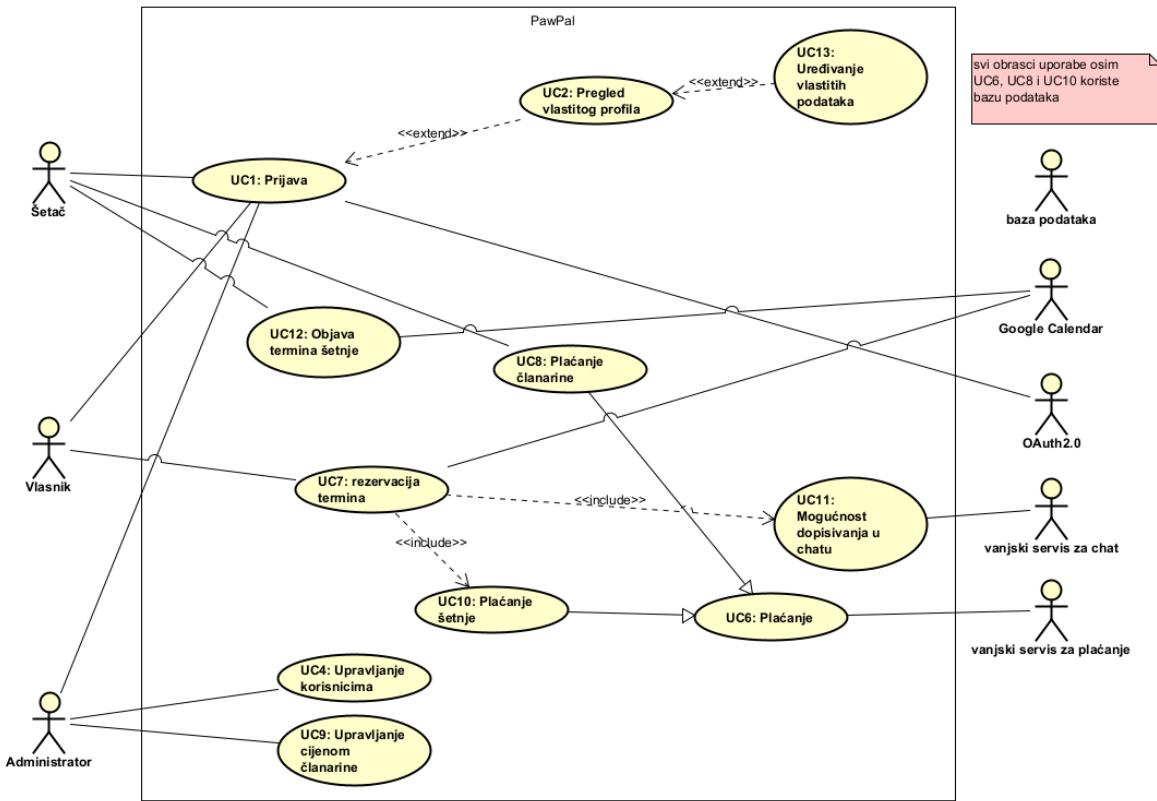
Obrasci uporabe

napomena: aktori s desne strane su uvijek baza podataka ili vanjski servis za neku uslugu te su oni uvijek pasivni aktori. S lijeve strane sustava se prikazuju korisnici koji su aktivni aktori.

Dijagrami obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

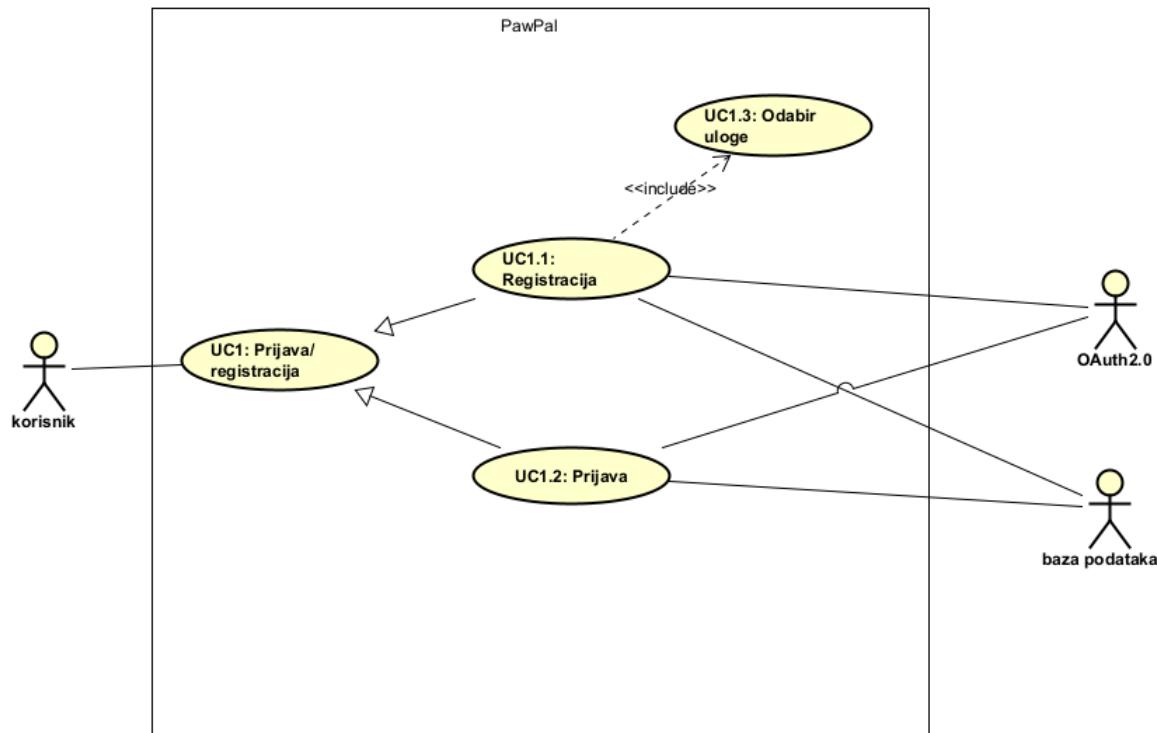
prikazuje ključne funkcionalnosti koje ugrubo prikazuju što radi naša aplikacija



2. dijagram obrazaca uporabe za ključne značajke

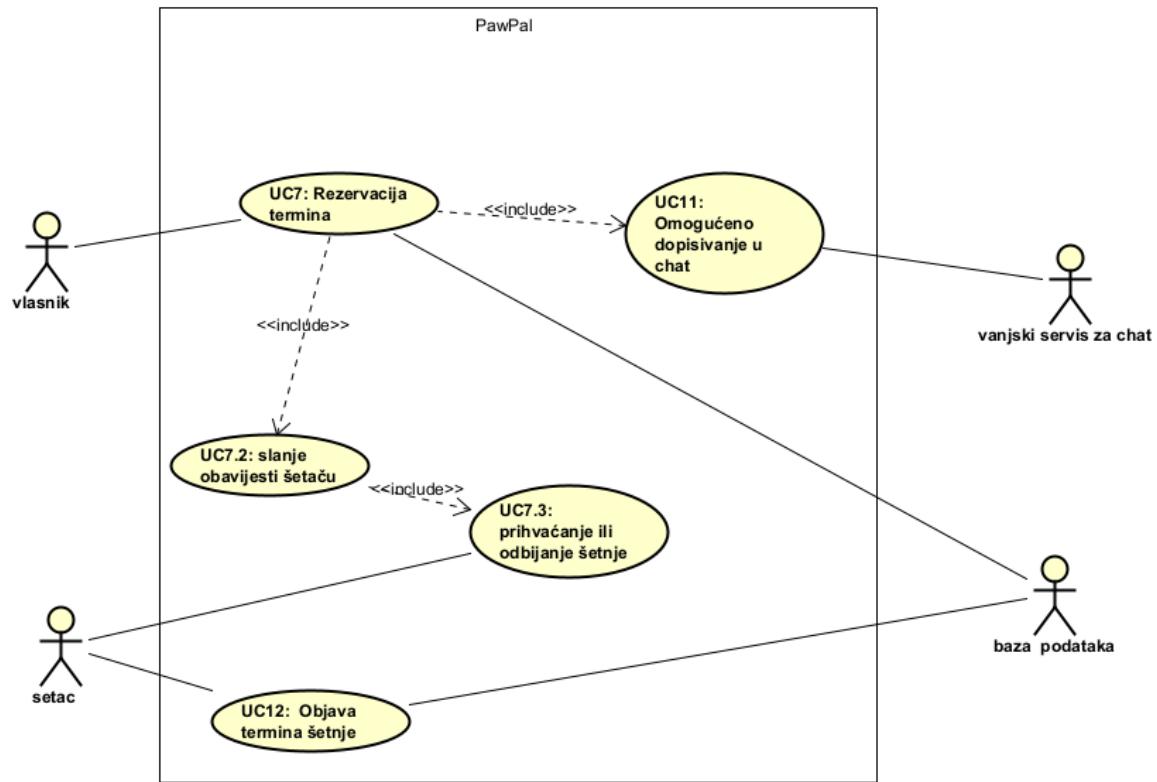
ključne značajke su rastavljene na poddijagrame u kojima se detaljnije vidi kako su osmišljene

2.1.Prijava u sustav



- neregistrirani/neprijavljeni korisnik vidi samo Main Page, kako bi mogao koristiti našu aplikaciju potrebno je obaviti prijavu/ registraciju

2.2. Rezervacija termina

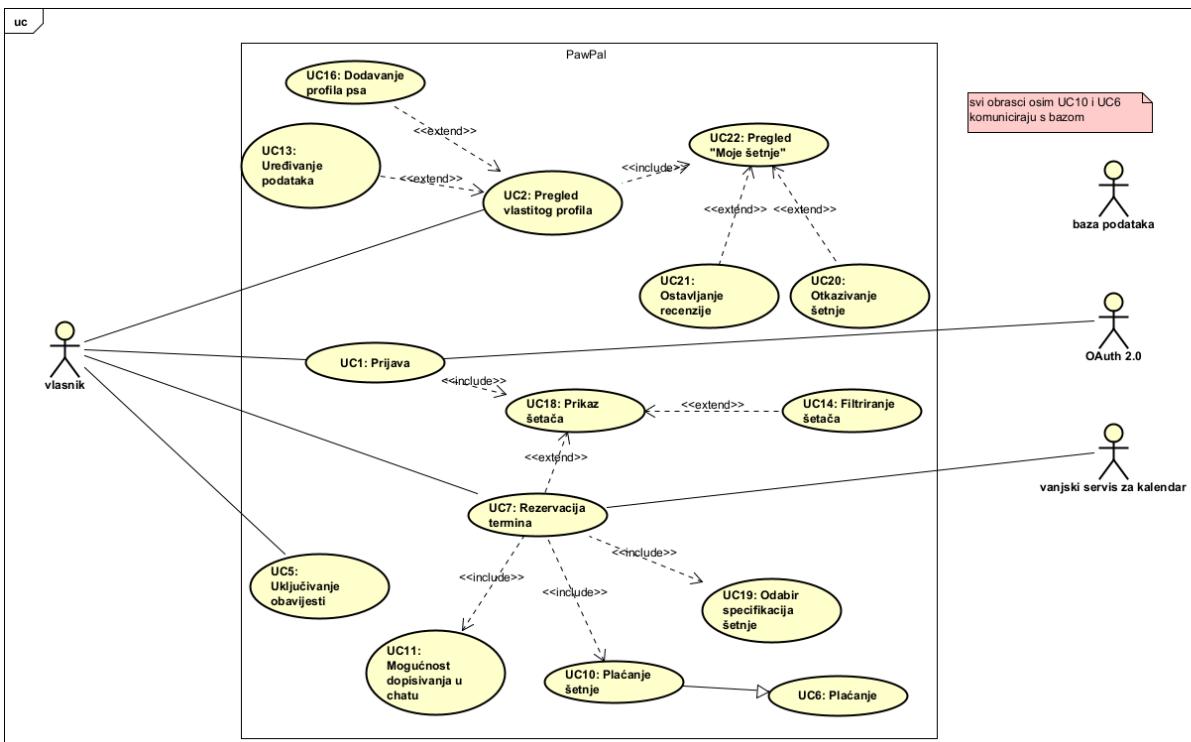


- jako bitna funkcionalnost na kojoj se temelji naša aplikacija: vlasnik bira termin te ga rezervira za šetnju

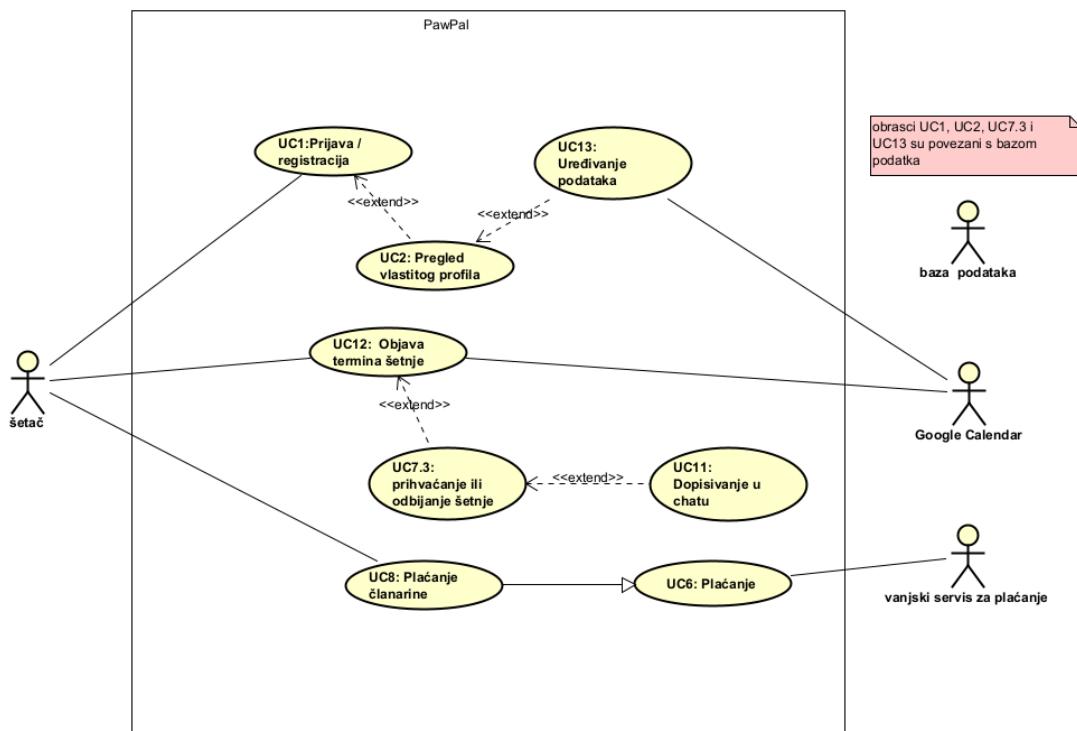
3. dijagram obrazaca uporabe za korisničke uloge

Identificira glavne korisničke aktore i njihove interakcije sa sustavom te definira vrste korisnika (kojih imamo tri, četiri ako brojimo korisnika koji se još nije prijavio/ registrirao u sustav, no on nije prikazan jer nema puno bitnih funkcionalnosti) i funkcionalnosti koje su im potrebne

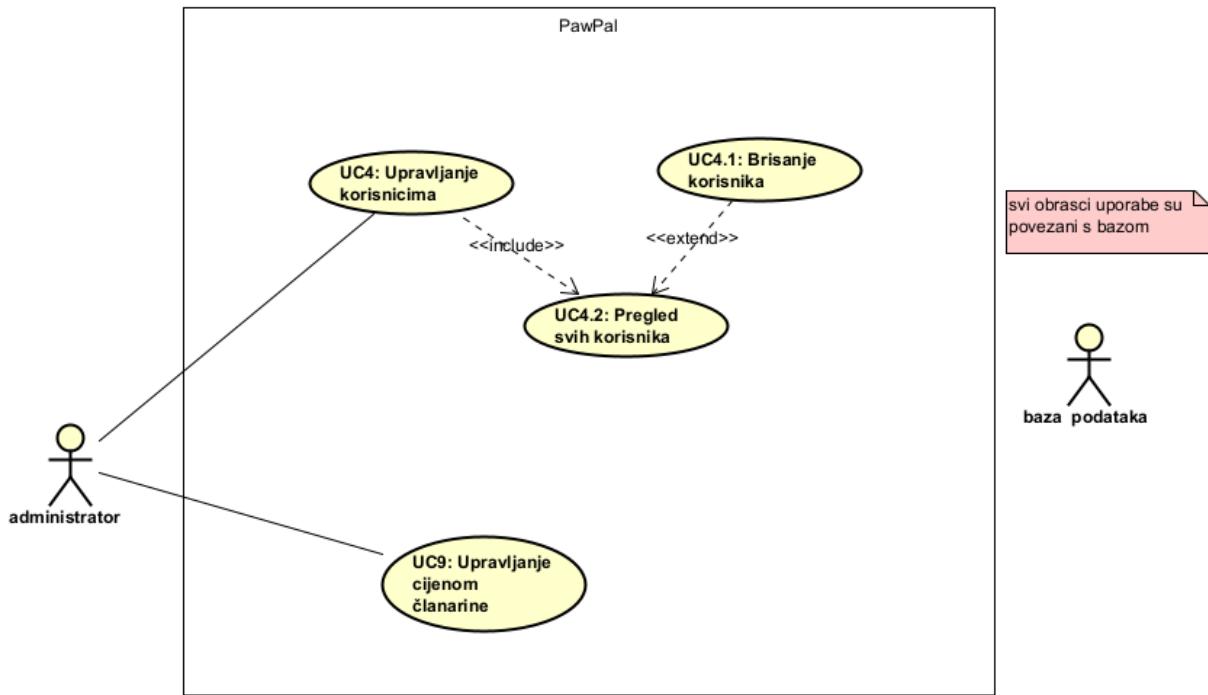
3.1. Vlasnik



3.2. Šetač

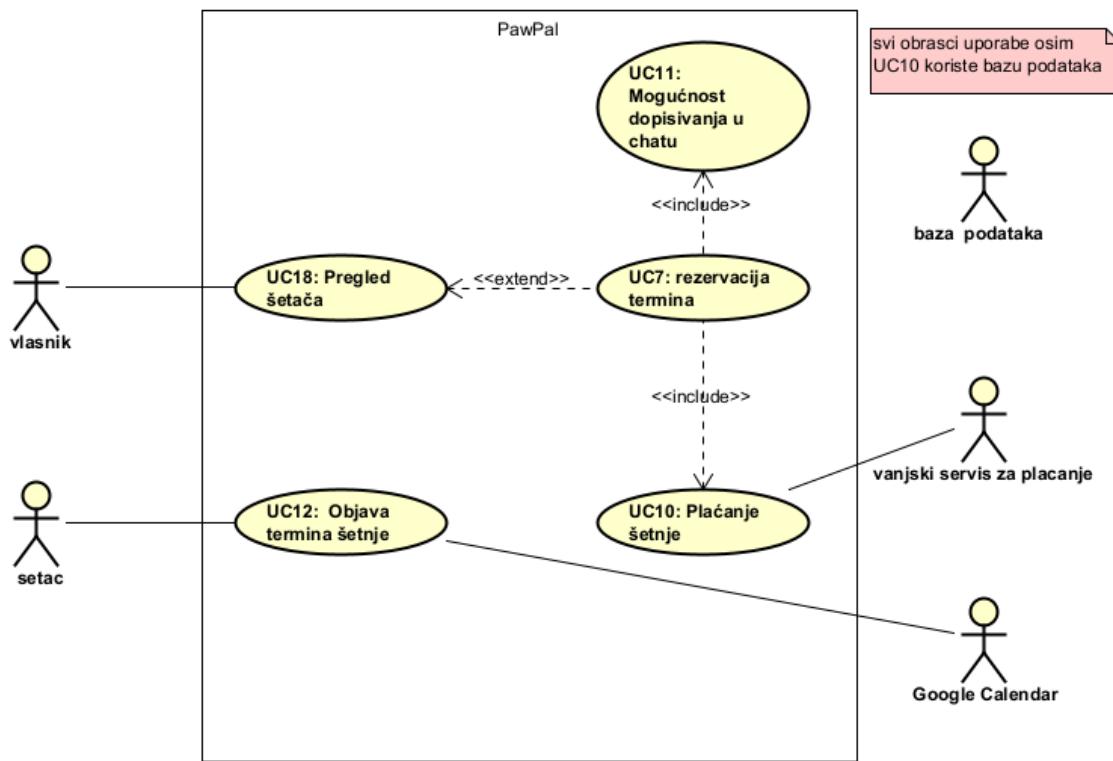


3.3. Administrator



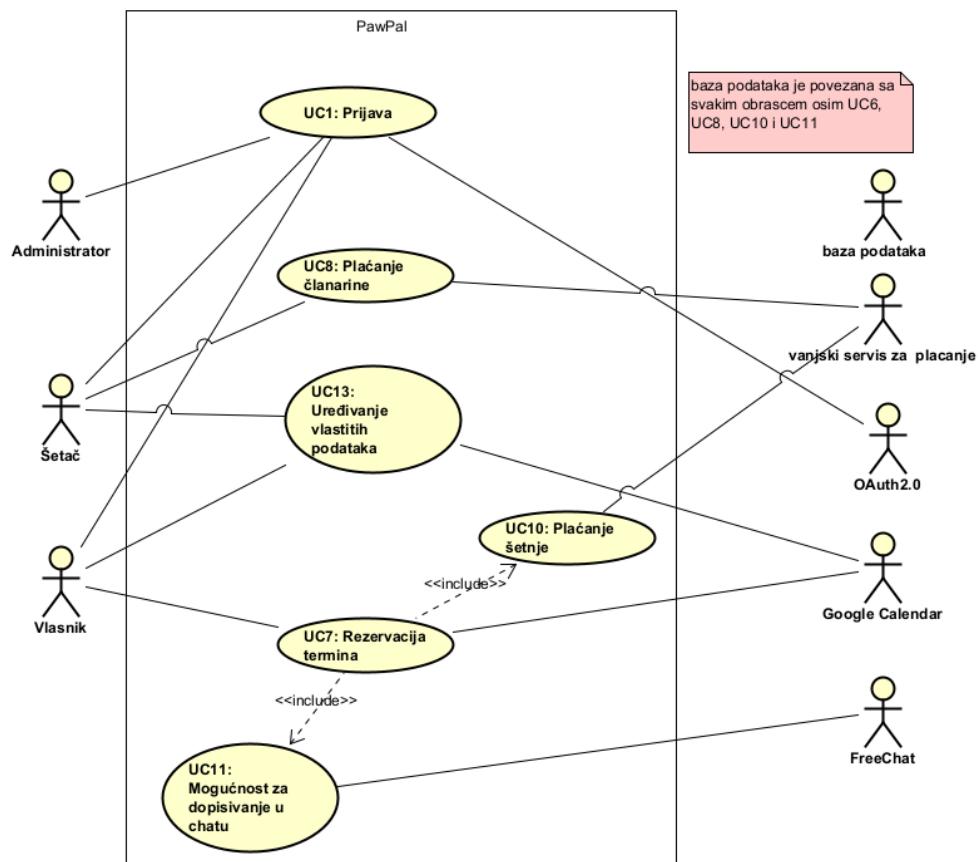
4. dijagram obrazaca uporabe za osnovne poslovne procese

dijagram obrazaca uporabe za osnovne poslovne procese opisuje tijek korištenja naše aplikacije: pregled šetača te rezervacija prehodno objavljenog termina uz dodatne specifikacije i plaćanje. Ovo ukratko predstavlja cilj naše aplikacije: omogućiti vlasnicima rezervaciju termina šetnji.



5. dijagram obrazaca uporabe za kritične sustave i integracije

dijagram prikazuje interakcije sustava s vanjskim sustavima koji su korišteni. U našoj aplikaciji su to Google Calendar za kalendar, FreeChat za chat u aplikaciji, OAuth2.0 za autorizaciju te vanjski servis za plaćanje



Opis obrazaca uporabe

Napomene:

- "korisnik" se odnosi na sve korisnike osim administratora, dakle korisnike prijavljene kao vlasnici pasa ili šetači pasa
- u "Sudionici" nije nabrojan glavni sudionik jer se on podrazumijeva kao sudionik
- za svaki obrazac uporabe kojemu su glavni sudionici vlasnik/šetac se kao preduvjet podrazumijeva prijava/registracija u sustav

UC1 – Prijava / registracija

- Glavni sudionik: korisnik
- Cilj: Omogućiti korisniku da pristupi sustavu putem registracije ili prijave.
- Sudionici: baza podataka
- Preduvjet: korisnik nije prijavljen u sustav
- Opis osnovnog tijeka:

1. korisnik klikne na gumb prijava
2. prelazi na odgovarajuću stranicu za unos podataka (email, lozinka)

UC1.1 – Registracija

- Glavni sudionik: korisnik
- Cilj: Pristup aplikaciji za nove korisnike.
- Sudionici: baza podataka
- Preduvjet: ne postoji korisnički račun s tim emailom
- Opis osnovnog tijeka:
 1. korisnik upisuje email
 2. korisnik upisuje lozinku
 3. korisnik bira ulogu vlasnik ili šetač
 4. zapis novog korisnika u bazu podataka
- Opis mogućih odstupanja:
 1. lozinka/email nije upisan - ispisuje se poruka upozorenja

UC1.2 – Prijava

- Glavni sudionik: korisnik
- Cilj: Pristup aplikaciji za prijavljene korisnike.
- Sudionici: baza podataka, OAuth2.0
- Preduvjet: korisnik je došao na našu web stranicu
- Opis osnovnog tijeka:
 1. korisnik upisuje email
 2. korisnik upisuje lozinku
 3. korisnik klikne na gumb "Prijava"
- Opis mogućih odstupanja:
 1. opis korak jedan
- Opis mogućih odstupanja:
 1. lozinka/email nije točan (ne postoji taj korisnik u bazi)- ispisuje se poruka upozorenja

UC1.3 – Odabir uloge

- Glavni sudionik: korisnik

- Cilj: Postoje dvije vrste korisnika: vlasnik psa ili šetač koje sami odabiru pri registraciji.
- Sudionici: baza podataka
- Preduvjet: korisnik još nije registriran
- Opis osnovnog tijeka:
 1. korisnik klikom bira ulogu vlasnik ili šetač

UC2 – Pregled vlastitog profila

- Glavni sudionik: korisnik
- Cilj: Omogućiti pregled i provjeru vlastitih podataka na profilu.
- Sudionici: baza podataka
- Preduvjet: korisnik klikne na svoj profil
- Opis osnovnog tijeka:
 1. klikom na ikonicu profila korisnik pristupa stranici gdje su dostupni njegovi osobni podatci

UC4 – Upravljanje korisnicima

- Glavni sudionik: administrator
- Cilj: Omogućiti administratoru pregled, uređivanje i brisanje korisnika.
- Sudionici: baza podataka
- Preduvjet: potvrda statusa administratora s OAuth2.0
- Opis osnovnog tijeka:
 1. administrator klikne na gumb "Upravljanje korisnicima"
 2. otvara se pregled korisnika

UC4.1 – Brisanje korisnika

- Glavni sudionik: administrator
- Cilj: Omogućiti administratoru brisanje korisnika iz sustava.
- Sudionici: baza podataka
- Preduvjet: potvrda statusa administratora s OAuth2.0, postojanje barem jednog korisnika
- Opis osnovnog tijeka:
 1. administrator klikom na gumb briše korisnika iz baze

UC4.2 – Pregled svih korisnika

- Glavni sudionik: administrator
- Cilj: Administrator može pregledati popis svih korisnika sustava.
- Sudionici: baza podataka
- Preduvjet: potvrda statusa administratora s OAuth2.0
- Opis osnovnog tijeka:
 1. administrator dobiva popis svih registriranih korisnika

UC5 – Uključivanje obavijesti

- Glavni sudionik: vlasnik
- Cilj: Vlasnik može dobivati obavijesti o novim šetačima.
- Sudionici: baza podataka
- Preduvjet: vlasnik se nalazi na svom profilu
- Opis osnovnog tijeka:
 1. vlasnik klikne na gumb "uključi obavijesti"
 2. vlasnik dobiva obavijesti o novim šetačima

UC6 – Plaćanje

- Glavni sudionik: korisnik
- Cilj: Omogućiti plaćanje svih usluga putem sustava.
- Sudionici: vanjski servis za plaćanje
- Preduvjet: dovoljni resursi na kartici/PayPal profilu
- Opis osnovnog tijeka:
 1. korisnik klikne na gumb "Plati"
 2. korisnik odabire opciju plaćanja
 3. preko vanjskog servisa za plaćanje se obavlja plaćanje
- Opis mogućih odstupanja:
 1. nema dovoljno sredstva na računu - ispisuje se obavijest i prekida plaćanje
 2. ako vlasnik odabere da će platiti uživo šetaču, na aplikaciji se ne izvršava plaćanje

UC7 – Rezervacija termina

- Glavni sudionik: vlasnik
- Cilj: Omogućiti rezervaciju termina za šetnju psa.

- Sudionici: baza podataka
- Preduvjet: vlasnik je na profilu šetača
- Opis osnovnog tijeka:
 1. vlasnik je našao termin koji je šetač objavio koji mu odgovara
 2. vlasnik klikće na gumb "Rezerviraj"
 3. otvara se novi prozor u kojemu vlasnik bira specifikacije šetnje te način plaćanja
 4. vlasnik klikće na gumb "Gotovo"
 5. šetaču se šalje obavijest

UC7.2 – Slanje obavijesti šetaču

- Glavni sudionik: sustav
- Cilj: Obavijestiti šetača o novoj rezervaciji termina.
- Sudionici:
- Preduvjet: napravljena je rezervacija termina za tog šetača
- Opis osnovnog tijeka:
 1. šetač dobiva obavijest s dvije mogućnosti: prihvatanje ili odbijanje

UC7.3 – Prihvatanje ili odbijanje šetnje

- Glavni sudionik: šetač
- Cilj: Šetači mogu prihvatiti zahtjev za šetnju, ili ga odbiti ako mu ne odgovaraju specifikacije šetnje
- Sudionici: baza podataka
- Preduvjet: napravljena je rezervacija termina za tog šetača, poslana je obavijest šetaču
- Opis osnovnog tijeka:
 1. šetač na obavijesti bira "Prihvati" ili "Odbij"
 2. ako je odabrao "Prihvati", rezervacija se sprema u bazu podataka i omogućuje se chat između tog vlasnika i tog šetača
 3. ako je odbio, ne događa se ništa
 4. zatvara se prozor s obavijesti

UC8 – Plaćanje članarine

- Glavni sudionik: šetač
- Cilj: Šetači mogu svoju članarinu platiti putem sustava.

- Sudionici: vanjski servis za plaćanje
- Preduvjet: članarina nije plaćena za taj mjesec
- Opis osnovnog tijeka:
 1. korisnik klikne na gumb "Plati"
 2. korisnik odabire opciju plaćanja
 3. preko vanjskog servisa za plaćanje se obavlja plaćanje
- Opis mogućih odstupanja:
 1. nema dovoljno sredstva na računu - ispisuje se obavijest i prekida plaćanje

UC9 – Upravljanje cijenom članarine

- Glavni sudionik: administrator
- Cilj: Administrator može definirati i izmjeniti cijene članarine.
- Sudionici: baza podataka
- Preduvjet: potvrda statusa administratora s OAuth2.0
- Opis osnovnog tijeka:
 1. administrator klikne na gumb "Upravljanje cijenom članarine"
 2. administrator mijenja uređuje cijenu
- Opis mogućih odstupanja:
 1. administrator stavi cijenu manju ili jednaku 0 - ispisuje se upozorenje i izmjena nije prihvaćena

UC10 – Plaćanje šetnje

- Glavni sudionik: vlasnik
- Cilj: Omogućiti plaćanje obavljene šetnje putem aplikacije.
- Sudionici: vanjski servis za plaćanje
- Preduvjet: rezerviran je termin za šetnju i šetač ju je prihvatio
- Opis osnovnog tijeka:

1 korisnik klikne na gumb "Plati" 2. korisnik odabire opciju plaćanja 3. preko vanjskog servisa za plaćanje se obavlja plaćanje

- Opis mogućih odstupanja:
 1. nema dovoljno sredstva na računu - ispisuje se obavijest i prekida plaćanje

2. ako vlasnik odabere da će platiti uživo šetaču, na aplikaciji se ne izvršava plaćanje

UC11 – mogućnost dopisivanje u chatu

- Glavni sudionik: korisnik
- Cilj: Jednostavna komunikacija između korisnika porukama.
- Sudionici: drugi korisnik
- Preduvjet: vlasnik je rezervirao termin s tim šetačem
- Opis osnovnog tijeka:
 1. vlasnik/šetač klikne na ikonicu razgovora
 2. otvara se chat
 3. vlasnik/šetač može slati poruke

UC12 – Objava novog termina šetnje

- Glavni sudionik: šetač
- Cilj: Šetači mogu na svom javnom profilu postaviti novi termin kojeg vlasnici mogu rezervirati
- Sudionici: baza podataka
- Preduvjet: šetač je na svom profilu
- Opis osnovnog tijeka:
 1. šetač klikne na gumb "Dodaj termin"
 2. otvara se novi prozor i šetač ispunjava polja: datum, vrijeme i trajanje termina
 3. šetač klikne na "Objavi"
 4. zatvara se prozor te se šetaču prikazuje njegov profil *Opis odstupanja:
 1. ako šetač ne ispuni jedno od polja a klikne na gumb "Objavi", termin se neće spremiti u bazu već će šetaču izbaciti upozorenje "Ispunite sva polja"

UC13 – Uređivanje vlastitih podataka

- Glavni sudionik: korisnik
- Cilj: Dodavanje, brisanje i izmjena osobnih podataka na profilu.
- Sudionici: baza podataka
- Preduvjet: korisnik pregledava svoj profil
- Opis osnovnog tijeka:
 1. korisnik klikne na opciju "Uredi"

2. korisniku se prikazuje njegov profil s podatcima koje može uređivati
3. korisnik klikće na "Spremi promjene"
4. korisniku se opet prikazuje pregled njegovog profila

UC14 – Filtriranje šetača

- Glavni sudionik: vlasnik
- Cilj: Mogućnost filtriranja šetača prema kriterijima: lokacija, cijena, ocjena korisnika.
- Sudionici: baza podataka
- Preduvjet: vlasnik se nalazi na HomePage-u
- Opis osnovnog tijeka:
 1. vlasnik odabire opciju filtriraj
 2. odabire jedan od kriterija po kojem se filtrira
 3. iz baze se dohvaćaju samo šetači s tim svojstvom
 4. ti šetači se prikazuju vlasniku

UC16 – Dodavanje profila psa

- Glavni sudionik: vlasnik
- Cilj: Omogućiti vlasniku dodavanje profila svog psa u sustav.
- Sudionici: baza podataka
- Preduvjet: vlasnik ima maksimalno 9 profila pasa te pregledava svoj profil
- Opis osnovnog tijeka:
 1. vlasnik klikne na gumb "Dodaj psa"
 2. otvara se prozor gdje vlasnik treba u odgovarajuća polja upisati ime, pasmina, starost, zdravstvene napomene (npr. lijekovi, alergije), razina energije, socijalizacija s drugim psima, dopuštene poslastice
 3. vlasnik klikne na gumb "Spremi"
 4. novi pas je spreljen u bazu podataka
 5. zatvara se prozor i vlasniku se prikazuje pregled njegovog profila
- Opis mogućih odstupanja:
 1. nisu ispunjena sva polja - ispisuje se upozorenje

UC18 – Pregled šetača

- Glavni sudionik: vlasnik
- Cilj: Omogućiti vlasniku pregled svih dostupnih šetača.
- Sudionici: baza podataka
- Preduvjet: prijava/registracija u sustav
- Opis osnovnog tijeka:
 1. vlasniku se nakon prijave/ registracije otvara HomePage na kojem su izlistani dostupni šetači

UC19 – Odabir specifikacija šetnje

- Glavni sudionik: vlasnik
- Cilj: Vlasnik može definirati detalje šetnje (datum, vrijeme, trajanje šetnje, tip šetnje (individualna/grupna), polazišnu adresu te dodaje eventualne napomene).
- Sudionici: baza podataka
- Preduvjet: vlasnik je na profilu šetača našao termin i kliknuo gumb "Rezerviraj"
- Opis osnovnog tijeka:
 1. otvara se prozor za rezervaciju
 2. vlasnik ispunjava polja: datum, vrijeme, trajanje šetnje, tip šetnje (individualna/grupna), polazišnu adresu te dodaje eventualne napomene
 3. vlasnik nastavlja po obrascu uporabe UC7 – Rezervacija termina
- Opis mogućih odstupanja:
 1. vlasnik nije ispunio sva obavezna polja specifikacija šetnje

UC20 – Otkazivanje šetnje

- Glavni sudionik: vlasnik
- Cilj: Mogućnost otkazivanja već rezervirane šetnje.
- Sudionici: baza podataka
- Preduvjet: u sustavu postoji rezervacija tog termina, vlasnik je na pregledu svog profila
- Opis osnovnog tijeka:
 1. vlasnik pod "Moje šetnje" nađe dogovoren termin za šetnju
 2. vlasnik klikne na gumb "Otkaži"
 3. šetnja se briše iz baze podataka
 4. vlasniku se prikazuje njegov profil

- Opis mogućih odstupanja:
 1. termin šetnje je za manje od 24 sata - iskače prozor s upozorenjem "Više nije moguće otkazati šetnju"

UC21 – Ostavljanje recenzije

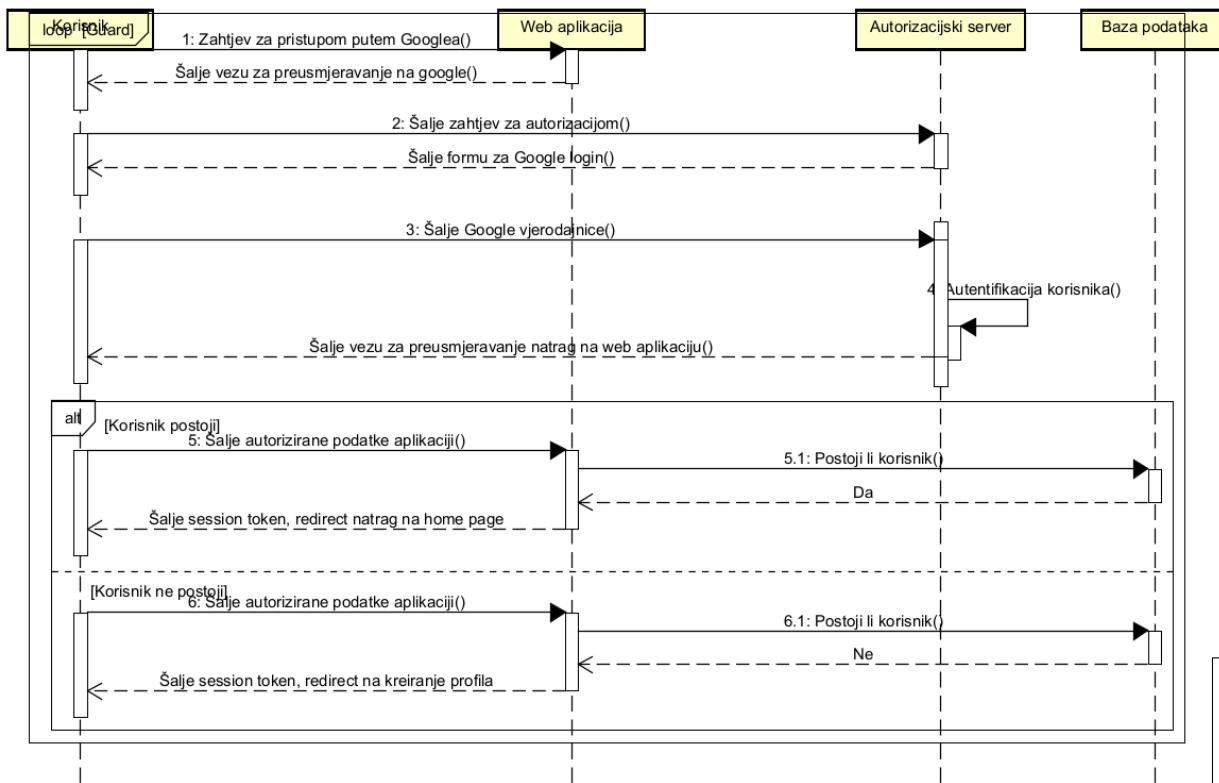
- Glavni sudionik: vlasnik
- Cilj: Ostavljanje ocjene i komentara nakon šetnje. Ocjena ulazi u ukupnu ocjenu šetača koja je vidljiva.
- Sudionici: baza podataka
- Preduvjet: šetač je obavio šetnju (termin šetnje je prošao)
- Opis osnovnog tijeka:
 1. vlasnik pod "Moje šetnje" nađe dogovoren termin za šetnju
 2. vlasnik klikne na gumb "Ocijeni"
 3. otvara se novi prozor u kojem vlasnik ispunjava polja: ocjena (1-5), fotografija, komentar
4 vlasnik klikne na gumb "Gotovo" 5 . vlasniku se prikazuje pregled njegovog profila
- Opis mogućih odstupanja:
 1. vlasnik nije dao ocjenu a stisnuo je "Gotovo" - ispisuje mu se obavijest "Morate dati ocjenu"

UC22 – Pregled "Moje šetnje"

- Glavni sudionik: vlasnik
- Cilj: Vlasnik može pregledati sve svoje prošle i buduće šetnje te ih tamo ocijeniti ili otkazati
- Sudionici: baza podataka
- Preduvjet: vlasnik je na pregledu svog profila
- Opis osnovnog tijeka:
 1. vlasnik pod "Moje šetnje" vidi prvo termine budućih te ispod njih i obavljenih šetnji s pripadajućom opcijom ocijenjivanja ili otkazivanja

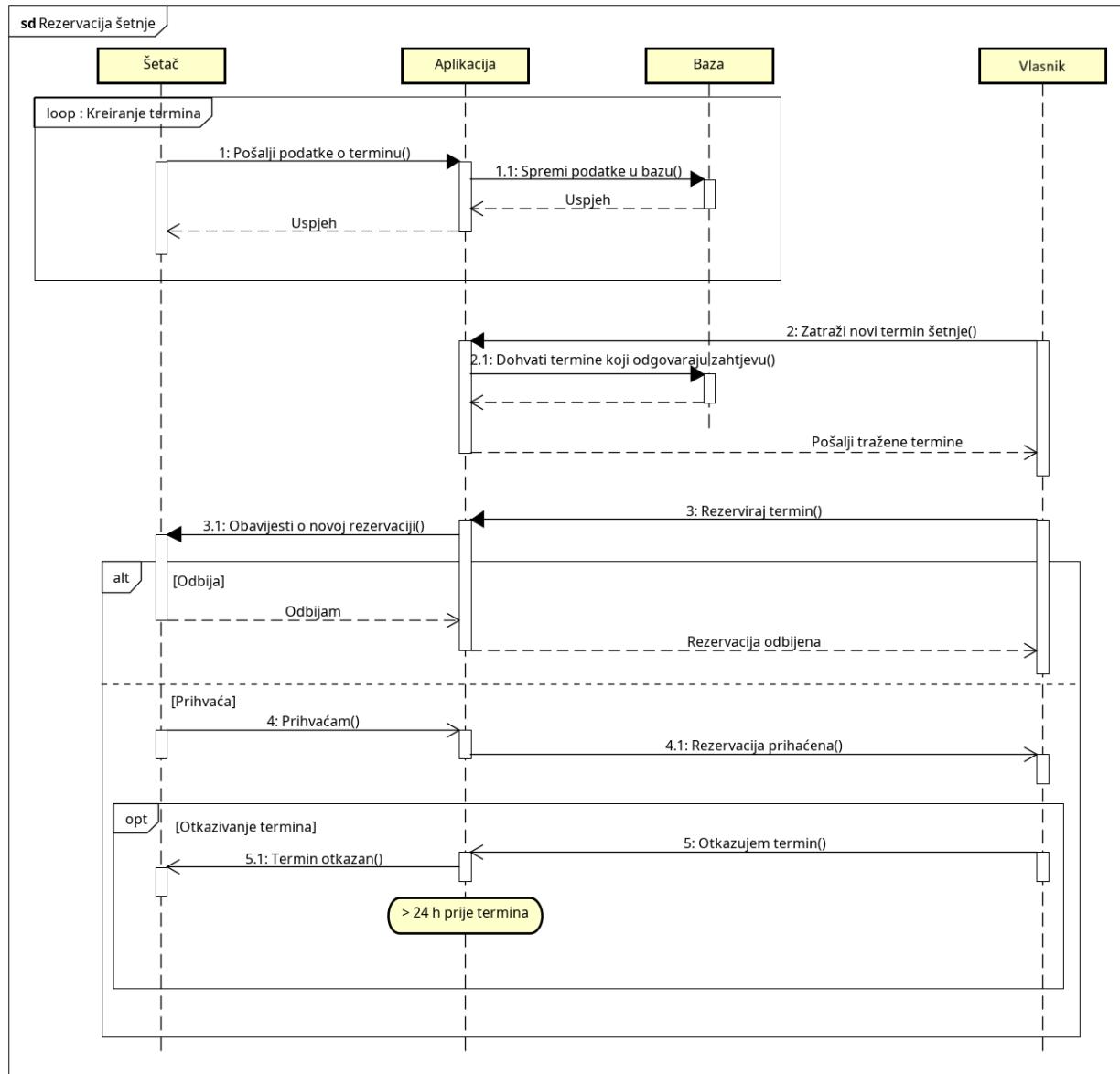
Sekvenički dijagrami

| UC1.2 - Prijava |



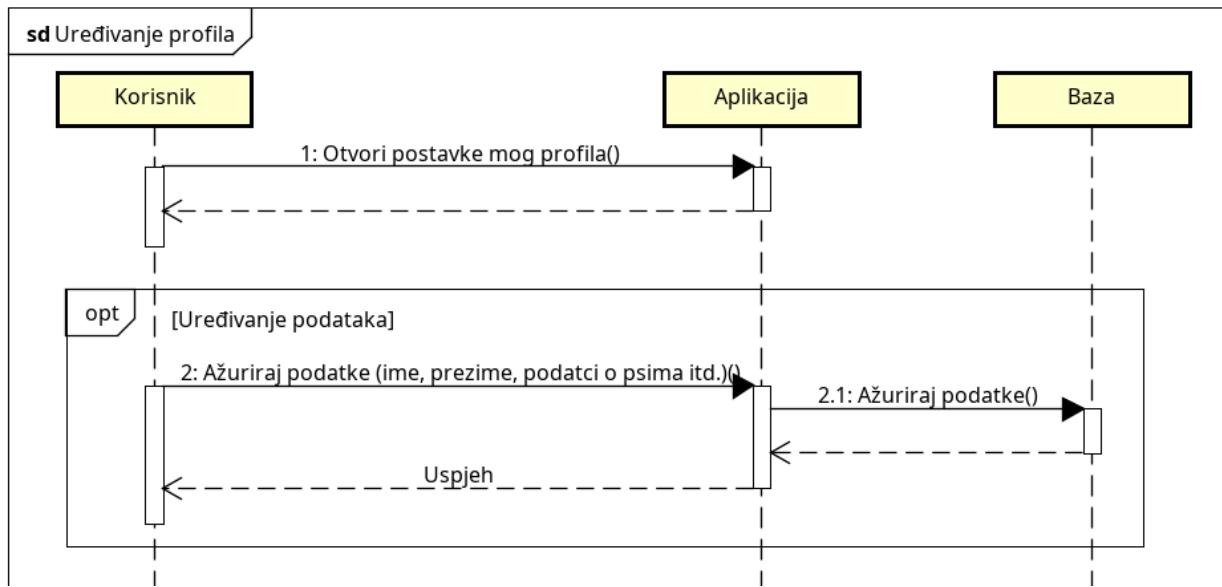
Dijagram prikazuje tijek prijave korisnika putem Google OAuth-a. Korisnik prvo odabire prijavu preko Googlea, nakon čega web aplikacija preusmjerava korisnika na Googleovu stranicu za autentifikaciju. Nakon što korisnik unese Google vjerodajnicu, autorizacijski server provjerava identitet i vraća autorizirane podatke web aplikaciji. Aplikacija zatim provjerava postoji li korisnik u bazi. Ako korisnik već postoji, kreira se session i korisnik se preusmjerava na početnu stranicu. Ako korisnik ne postoji, session se kreira, ali korisnik se preusmjerava na stranicu za dovršetak profila.

Šetnja



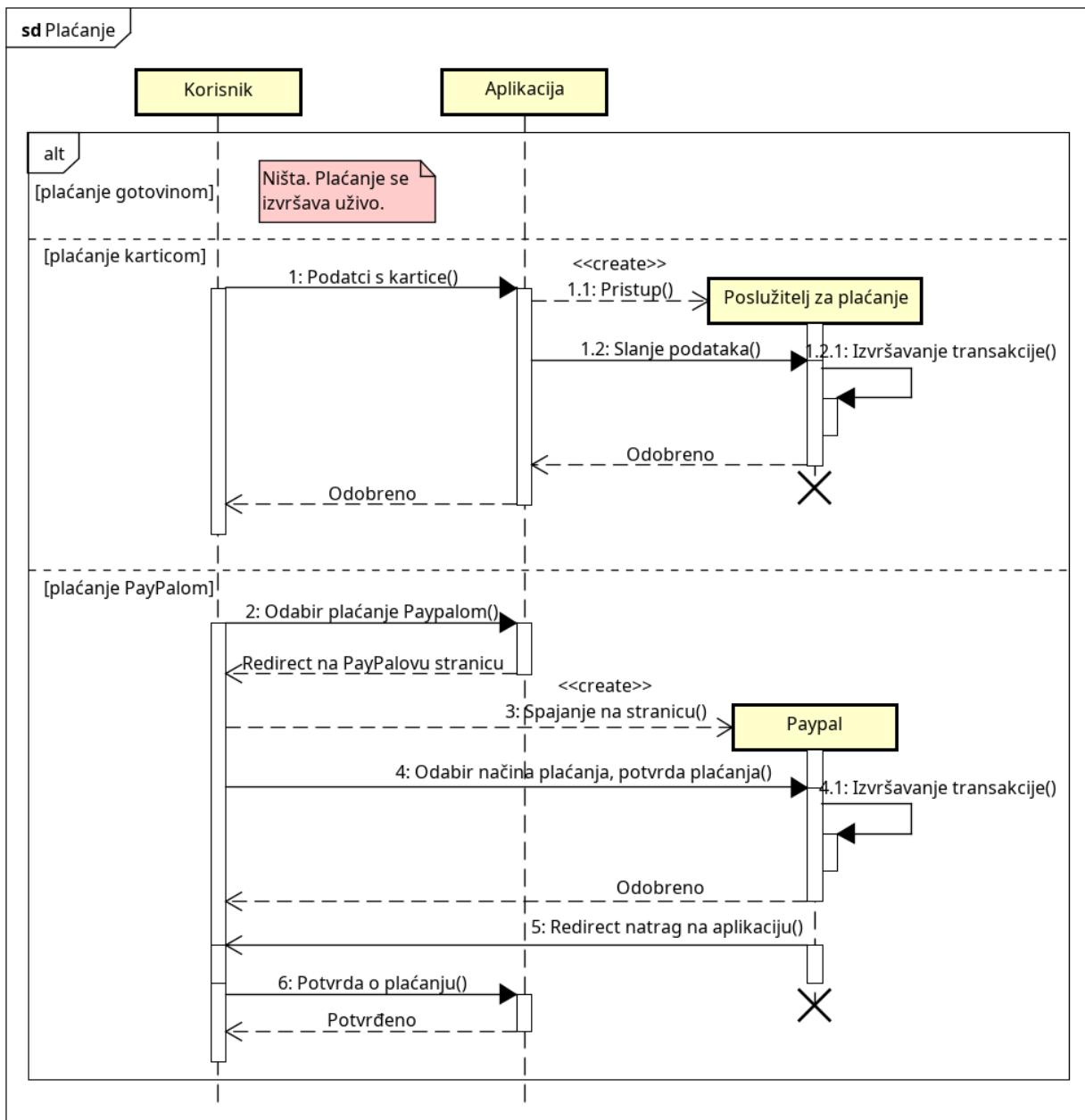
Dijagram prikazuje proces rezervacije termina šetnje. Šetač najprije unosi i sprema svoje dostupne termine u sustav. Kada vlasnik zatraži novu šetnju, aplikacija dohvaća odgovarajuće termine iz baze i prikazuje ih korisniku, koji zatim odabire željeni termin. Nakon što vlasnik pošalje zahtjev, šetač prima obavijest o novoj rezervaciji te ju može prihvati ili odbiti. Ako šetač prihvati, rezervacija postaje važeća; ako odbije, zahtjev se odbacuje. Vlasnik može otakzati termin, ali samo ako je više od 24 sata prije početka šetnje.

Profil



Kada korisnik otvorí postavke svog profila, aplikacija učitava postojeće podatke i prikazuje ih korisniku. Ako korisnik odlučí uređiti informacije – poput imena, prezimena ili dodatnih podataka o svojim psima – aplikacija šalje ažurirane podatke bazi podataka, koja ih pohranjuje i vraća potvrdu o uspješno izvršenoj izmjeni. Nakon potvrde ažuriranja aplikacija korisniku prikazuje poruku o uspjehu. Diagram prikazuje cijeli tijek komunikacije između korisnika, aplikacije i baze podataka tijekom uređivanja profila.

Plaćanje



Dijagram prikazuje tijek plaćanja u aplikaciji ovisno o odabranom načinu. Ako korisnik plaća gotovinom, aplikacija ne komunicira s vanjskim servisima i plaćanje se odvija uživo. Kod plaćanja karticom, korisnik unosi podatke, aplikacija ih šalje vanjskom platnom poslužitelju, izvršava se transakcija i rezultat se vraća korisniku. Ako se odabere PayPal, korisnik se preusmjerava na PayPalovu stranicu, tamo potvrđuje plaćanje, PayPal izvršava transakciju te korisnika vraća natrag u aplikaciju s potvrdom.

| izvorene kodove dijagrama pohranujte u svom GIT-u

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Obrazac	Funkcionalni zahtjevi
UC1 – Prijava / registracija	F-000, F-001
UC1.1 – Registracija	F-000
UC1.2 – Prijava	F-001
UC1.3 – Odabir uloge	F-002
UC2 – Pregled vlastitog profila	F-004
UC3 – Pregled tuđih profila	F-006
UC4 – Upravljanje korisnicima	F-007, F-008
UC4.1 – Brisanje korisnika	F-008
UC4.2 – Pregled svih korisnika	F-007
UC5 – Uključivanje obavijesti	F-009
UC6 – Plaćanje	F-010, F-012
UC7 – Rezervacija termina	F-013
UC7.2 – Slanje obavijesti šetaču	F-015
UC7.3 – Prihvatanje ili odbijanje šetnje	F-016
UC8 – Plaćanje članarine	F-010
UC9 – Upravljanje cijenom članarine	F-011
UC10 – Plaćanje šetnje	F-012
UC11 – Dopisivanje u chatu	F-017
UC12 – Objava termina šetnje	F-025
UC13 – Uređivanje vlastitih podataka	F-005
UC14 – Filtriranje šetača	F-018
UC16 – Dodavanje profila psa	F-019
UC18 – Pregled svih šetača	F-021
UC19 – Odabir specifikacija šetnje	F-022

Obrazac	Funkcionalni zahtjevi
UC20 – Otkazivanje šetnje	F-023
UC21 – Ostavljanje recenzije	F-024
UC22 – Pregled "Moje šetnje"	F-026

Arhitektura sustava

Arhitektura sustava

Cilj ovog poglavlja je pružiti jasan i strukturiran pregled arhitekture razvijenog programskog sustava. U nastavku su opisane ključne komponente, njihova međusobna povezanost te način na koji sustav ostvaruje zadane funkcionalne i niefunkcionalne zahtjeve.

1. Opis arhitekture

1.1 Stil arhitekture

Sustav koristi **MVC (Model–View–Controller)** arhitekturu, temeljenu na **Node.js (Express)** okviru za backend i **React** biblioteci za frontend.

Ovaj pristup omogućuje:

- **Odvajanje odgovornosti** između modela, prikaza i kontrolera
- **Fleksibilnost i proširivost** aplikacije
- **Jednostavno testiranje i održavanje**
- **Brzu integraciju novih funkcionalnosti**

Arhitektura slijedi **klijent–poslužitelj** model, gdje React frontend komunicira s Node.js backendom putem REST API-ja.

1.2 Podsustavi

Frontend (React)

- Implementira **View** sloj u MVC obrascu.
- Omogućava dinamičko i responzivno korisničko sučelje.
- Komunicira s backendom putem **HTTP/HTTPS REST API** poziva u **JSON** formatu.

Backend (Node.js + JAVAScript + Express)

Sastoji se od nekoliko slojeva:

- **Controller sloj** – prima i obrađuje HTTP zahtjeve od frontend-a.
- **Service sloj** – implementira poslovnu logiku i upravlja podacima.
- **Model sloj** – definira entitete i upravlja podacima u bazi.

Backend izlaže REST API sučelje prema frontendu i osigurava autentifikaciju, autorizaciju te rukovanje pogreškama.

Baza podataka (PostgreSQL)

- Koristi se **PostgreSQL** kao relacijska baza podataka.
 - Node.js backend komunicira s bazom pomoću **node-postgres**.
 - Entiteti uključuju korisnike, objave i druge podatke povezane s aplikacijom.
-

1.3 Preslikavanje na radnu platformu

Sustav je trenutno implementiran i testiran **lokalno**, s planom implementacije putem **Docker** kontejnera radi lakše distribucije, skalabilnosti i održavanja.

U budućnosti se može implementirati na **cloud platformi** (npr. AWS, Azure, Render, Vercel).

1.4 Spremišta podataka

Podaci se pohranjuju u **PostgreSQL** relacijsku bazu podataka.

Pohranjuju se:

- Korisnički podaci (autentifikacija, profili, autorizacija)
- Podaci o objavama, komentarima ili drugim sadržajima generiranim od strane korisnika

Slike se pohranjuju u **Microsoft Azure Blob Storage**.

1.5 Mrežni protokoli

Komunikacija između frontend-a i backend-a odvija se putem:

- **HTTP/HTTPS protokola**
 - HTTPS omogućuje **šifriranje i sigurnost** tijekom prijenosa osjetljivih podataka (npr. prijave korisnika).
-

1.6 Globalni upravljački tok

1. **Korisnik** putem React sučelja šalje zahtjev backendu (npr. prijava, dohvata podataka).
 2. **Backend (Controller)** prima zahtjev, prosljeđuje ga **Service** sloju.
 3. **Service** sloj obrađuje zahtjev i prema potrebi komunicira s **Model** slojem radi pristupa podacima.
 4. **Model** sloj dohvata ili pohranjuje podatke u **PostgreSQL** bazu.
 5. **Backend** vraća odgovor u JSON formatu, koji React frontend prikazuje korisniku.
-

1.7 Sklopoško-programske zahtjevi

Komponenta Zahtjev

Server / Hosting Render cloud platforma za hostanje Node.js aplikacije i PostgreSQL baze

OS Render podržava Linux-based okruženje za aplikacije, nema lokalnih zahtjeva

Backend Node.js (Express)

Frontend React, Node.js, npm

Baza PostgreSQL (hostano na Renderu)

S obzirom da se aplikacija hosta na Renderu, lokalni hardverski zahtjevi nisu strogo definirani. Razvojni tim može koristiti bilo koje računalo s podrškom za Node.js i npm za lokalni razvoj i testiranje.

2. Obrazloženje odabira arhitekture

2.1 Principi oblikovanja

Odabir MVC arhitekture temelji se na sljedećim principima:

- **Visoka kohezija** – svaki sloj ima jasno definiranu odgovornost
- **Niska povezanost** – slojevi komuniciraju preko REST API-ja
- **Modularnost i održivost** – sustav je jednostavno nadograđivati i testirati
- **Sigurnost** – omogućuje centralizirano upravljanje autentifikacijom i autorizacijom
- **Skalabilnost** – backend i frontend mogu se razvijati i skalirati neovisno

2.2 Razmatrane alternative

Razmatrana je **mikrouslužna arhitektura**, ali je zbog jednostavnosti i manjeg opsega projekta odabrana **monolitna MVC arhitektura** temeljena na **Node.js + React**.

Ovaj pristup omogućuje bržu implementaciju i jednostavnije održavanje.

3. Organizacija sustava na visokoj razini

3.1 Klijent–poslužitelj

- **Klijent (React)**: prikazuje korisničko sučelje i šalje REST API pozive.
- **Poslužitelj (Node.js + Express)**: obrađuje zahtjeve, primjenjuje poslovnu logiku i komunicira s bazom podataka.

3.2 Baza podataka

- Centralizirana relacijska baza **PostgreSQL**.
- Node package **pg** omogućuje jednostavno upravljanje relacijama i transakcijama.

3.3 Grafičko sučelje

- Izgrađeno u **Reactu**, s responzivnim dizajnom i dinamičkim prikazom podataka.
 - Komunicira s backendom putem REST API poziva u JSON formatu.
-

4. Organizacija aplikacije

Aplikacija je podijeljena u dvije glavne cjeline:

Frontend (React)

- Implementira **View** sloj MVC obrasca.
- Odgovoran za prikaz podataka i korisničke interakcije.
- Komunicira s backendom putem REST API poziva.

Backend (Node.js + Express)

- Implementira **Model** i **Controller** slojeve.
 - Obradjuje HTTP zahtjeve, primjenjuje poslovnu logiku i pristupa bazi podataka.
 - Omogućuje autentifikaciju, autorizaciju i rukovanje sesijama.
-

5. Reference

- [Architect modern web applications with ASP.NET Core and Azure – .NET | Microsoft Learn](#) (7. studeni 2024.)
 - Fowler, M. & Lewis, J. (2014). *Microservices: a definition of this new architectural term*.
 - Newman, S. (2015). *Building Microservices*. O'Reilly Media.
-

Baza podataka

Opis tablica

KORISNIK

Atribut	Tip podatka	Opis varijable
---------	-------------	----------------

idKorisnik (PK) INT	Jedinstveni identifikator korisnika
---------------------	-------------------------------------

Atribut	Tip podatka	Opis varijable
imeKorisnik	VARCHAR(50)	Ime korisnika
prezKorisnik	VARCHAR(50)	Prezime korisnika
email (U)	VARCHAR(60)	Elektronička pošta korisnika
telefon (U)	VARCHAR(20)	Broj mobilnog telefona korisnika

CLANARINA

Atribut	Tip podatka	Opis varijable
clanarina (PK)	VARCHAR(50)	Tip članarine za koju administrator postavlja cijenu (cijenaMjClanarina/cijenaGodClanarina)
iznos	NUMERIC(5, 2)	Cijena članarine koju administrator postavlja (iznos > 0)

SETAC

Atribut	Tip podatka	Opis varijable
idKorisnik (PK)(FK) INT		Jedinstveni identifikator šetača
tipClanarina	VARCHAR(15)	Članarina za koju se šetač odluči (mjesečna/godišnja)
profilFoto	VARCHAR(200)	Profilna fotografija šetača (pohranjuje se kao URL/lokalni path)
lokDjelovanja	VARCHAR(200)	Lokacija djelovanja šetača (npr. koji kvartovi)

VLASNIK

Atribut	Tip podatka	Opis varijable
idKorisnik (PK)(FK) INT		Jedinstveni identifikator vlasnika
primanjeObavijesti BOOLEAN		Je li vlasnik pretplaćen na primanje obavijesti o novim šetačima (default: FALSE)

PAS

Atribut	Tip podatka	Opis varijable
idPas (PK)	INT	Jedinstveni identifikator psa
imePas	VARCHAR(50)	Ime psa

Atribut	Tip podatka	Opis varijable
pasmina	VARCHAR(50)	Pasmina psa
socijalizacija	INT	Razina socijalizacije psa s drugim psima (1-5)
razinaEnergije	INT	Razina energije psa (1-5)
starost	INT	Starost psa (1-20)
zdravNapomene (O)	VARCHAR(500)	Opcionalne dodatne zdravstvene napomene za psa (npr. alergije, lijekovi)
idKorisnik (FK)	INT	Jedinstveni identifikator vlasnika

SETNJA

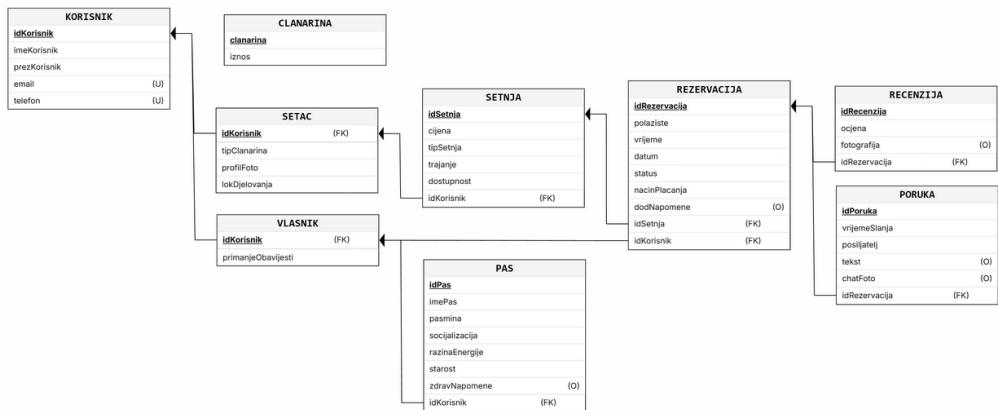
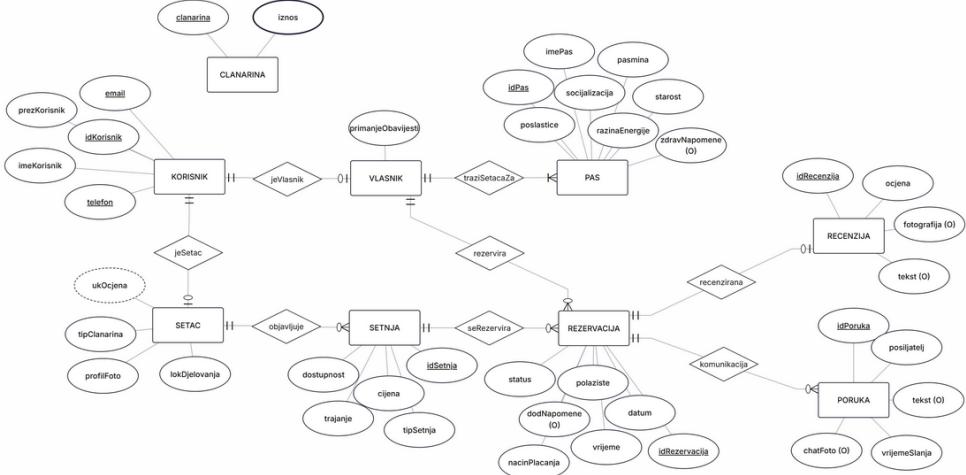
Atribut	Tip podatka	Opis varijable
idSetnja (PK)	INT	Jedinstveni identifikator šetnje
cijena	NUMERIC(4, 2)	Cijena (veća od 0) određene šetnje
tipSetnja	VARCHAR(15)	Tip šetnje (grupna/individualna)
trajanje	INT	Trajanje šetnje u minutama
dostupnost	BOOLEAN	Je li šetnju još uvijek moguće rezervirati (default: TRUE)
idKorisnik (FK)	INT	Jedinstveni identifikator šetača

REZERVACIJA

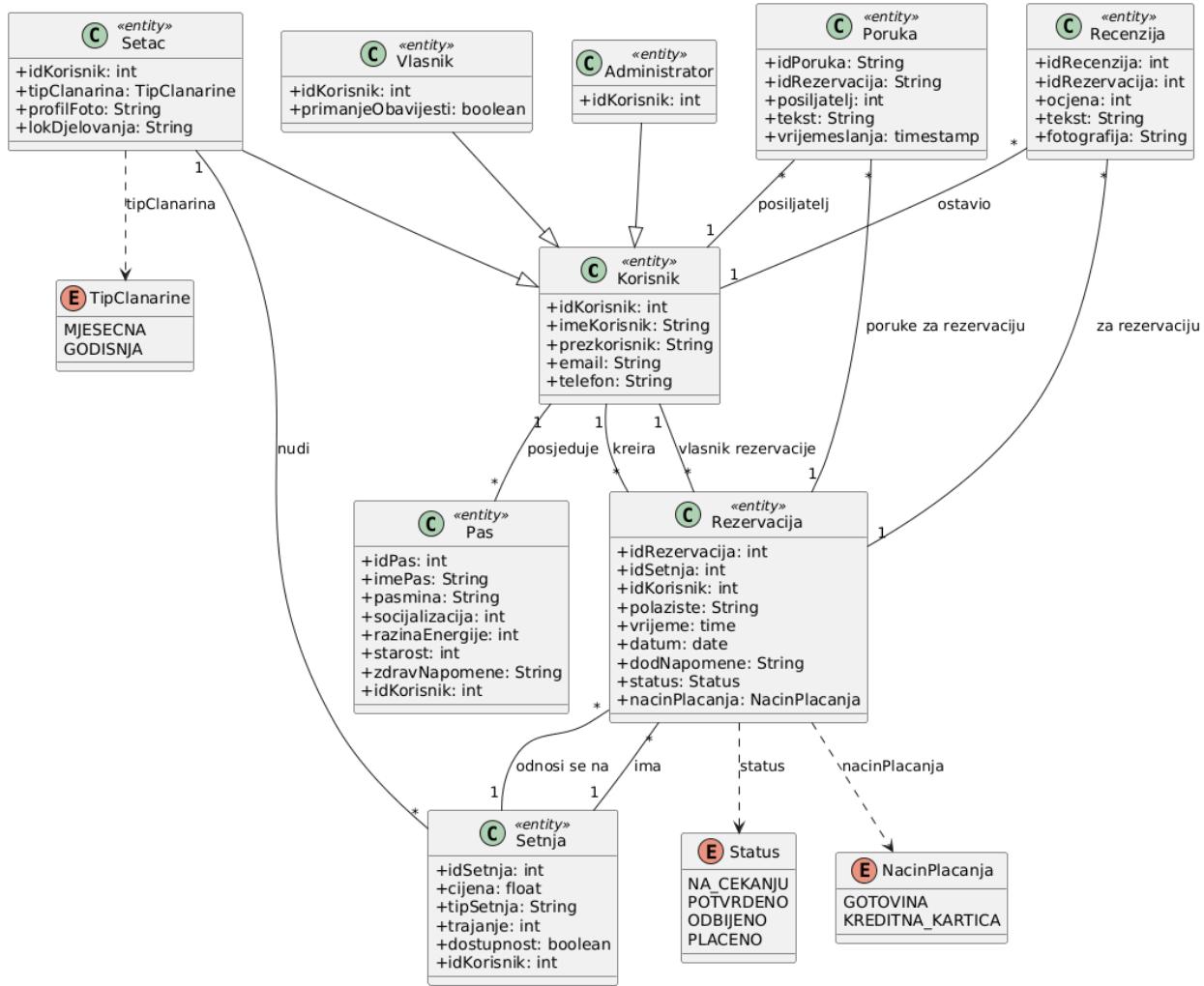
Atribut	Tip podatka	Opis varijable
idRezervacija (PK)	INT	Jedinstveni identifikator rezervacije
polaziste	VARCHAR(60)	Polazišna adresa šetnje
vrijeme	TIME	U koje doba dana treba izvesti psa u šetnju
datum	DATE	Dan na koji treba izvesti psa u šetnju
status	VARCHAR(15)	Status rezervacije (potvrdeno/odbijeno/na cekanju/placeno)
nacinPlacanja	VARCHAR(20)	Način plaćanja šetnje (gotovina/PayPal/kreditna kartica)
dodNapomene	VARCHAR(300)	Opcionalne dodatne napomene vezane uz šetnju

Atribut	Tip podatka	Opis varijable
idSetnja (FK)	INT	Jedinstveni identifikator šetnje
idKorisnik (FK)	INT	Jedinstveni identifikator vlasnika
RECENZIJA		
Atribut	Tip podatka	Opis varijable
idRecenzija (PK)	INT	Jedinstveni identifikator recenzije
ocjena	INT	Ocjena s kojom vlasnik želi ocjeniti obavljenu šetnju (1-5)
tekst (O)	VARCHAR(500)	Opcionalna tekstualna recenzija šetnje
fotografija (O)	VARCHAR(200)	Opcionalna fotografija uz recenziju (pohranjuje se kao URL/lokálni path)
idRezervacija (FK)	INT	Jedinstveni identifikator rezervacije
PORUKA		
Atribut	Tip podatka	Opis varijable
idPoruka (PK)	INT	Jedinstveni identifikator poruke
vrijemeSlanja	TIMESTAMP	Vrijeme slanja određene poruke
posiljatelj	VARCHAR(100)	Korisnik koji je poslao poruku
tekst (O)	VARCHAR(500)	Opcionalno poslani tekst u poruci
chatFoto (O)	VARCHAR(200)	Opcionalno poslana fotografija u poruci (pohranjuje se kao URL/lokálni path)
idRezervacija (FK)	INT	Jedinstveni identifikator rezervacije

Dijagram baze podataka

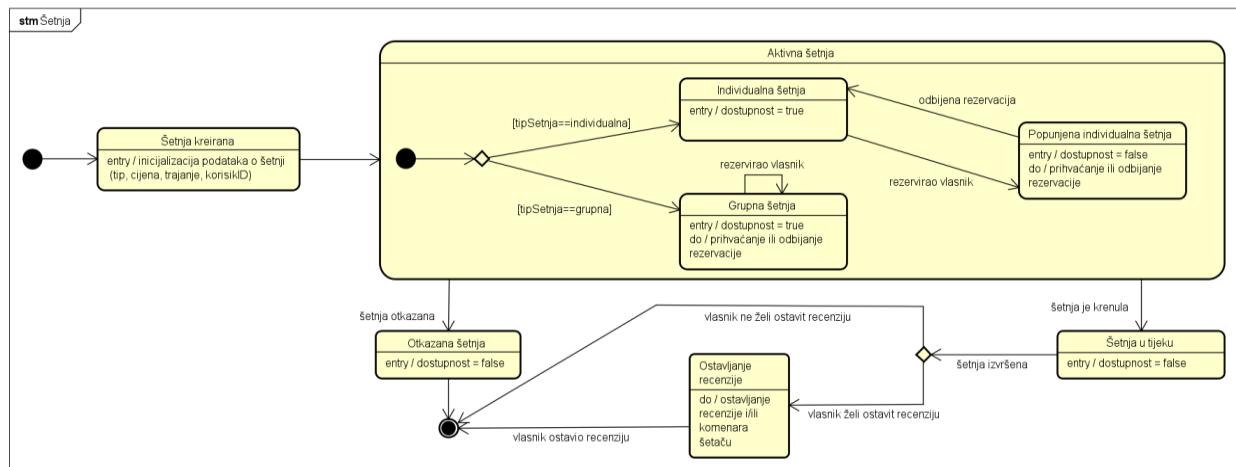


Dijagram razreda



UML dijagrami stanja

Dijagram stanja – Šetnja



Dijagram stanja prikazuje stanja **šetnje**, od izrade, rezervacije, otkazivanje te ostavljanje recenzije.

Proces započinje kreiranjem nove šetnje. U ovom stanju izvršava se inicijalizacija podataka o šetnji, kao što su tip šetnje (individualna ili grupna), cijena, trajanje i identifikator korisnika (šetača).

Nakon kreiranja, šetnja prelazi u stanje **Aktivna šetnja**, unutar kojeg se definiraju različita stanja ovisno o tipu šetnje.

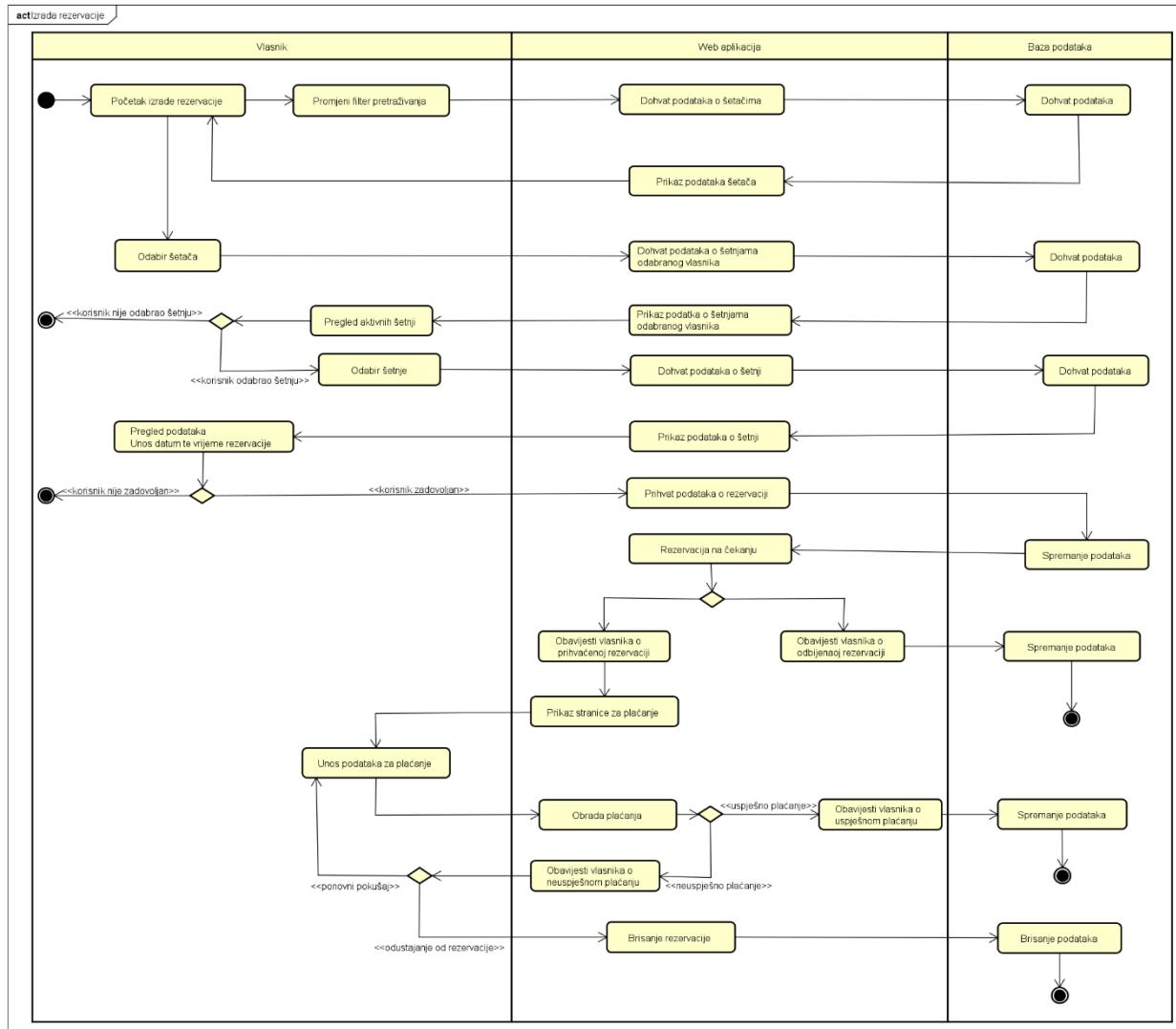
- Ako je tip šetnje individualna, šetnja prelazi u stanje **Individualna šetnja**, gdje je postaje dostupna za rezerviranje. Nakon što vlasnik rezervira šetnju, ona prelazi u stanje **Popunjena individualna šetnja**, u kojem se onemogućuje rezervacija te je omogućeno prihvatanje ili odbijanje već pristigle rezervacije. U slučaju odbijanja rezervacije, šetnja se ponovno vraća u početno stanje.
- Ako je tip šetnje grupna, šetnja prelazi u stanje **Grupna šetnja**, gdje je također označena kao dostupna. U ovom stanju moguće je prihvatanje ili odbijanje više rezervacija, budući da grupna šetnja može imati više sudionika.

Iz stanja Aktivna šetnja moguće je otkazivanje šetnje, čime ona prelazi u stanje **Otkazana šetnja**. U tom stanju šetnja više nije dostupna za rezervacije te se proces završava.

Kada započne dogovoren termin, šetnja prelazi u stanje **Šetnja u tijeku**, u kojem se onemogućuju daljnje rezervacije. Nakon izvršenja šetnje, vlasnik ima mogućnost **ostaviti recenziju i/ili komentar šetaču**. Ako vlasnik odluči ostaviti recenziju, šetnja prelazi u stanje **Ostavljanje recenzije**, nakon čega proces završava. U slučaju da vlasnik ne želi ostaviti recenziju, proces se također završava.

UML dijagram aktivnosti

Dijagram aktivnosti - Izrada rezervacije



Dijagram aktivnosti prikazuje tijek **izrade rezervacije**. Proces je prikazan kroz interakciju tri sudionika: **vlasnika, web aplikacije i baze podataka**.

Proces započinje kada vlasnik pokrene **izradu rezervacije**. Tada vlasnik može **promijeniti filter pretraživanja** kako bi pronašao šetača. Web aplikacija tada dohvaća podatke o dostupnim šetačima iz baze podataka te ih prikazuje vlasniku.

Nakon pregleda šetača, vlasnik odabire jednog. Web aplikacija potom dohvaća podatke o **aktivnim šetnjama odabranog šetača** iz baze podataka i prikazuje ih vlasniku.

Nakon što vlasnik pogleda i **odabere konkretnu šetnju** web aplikacija dohvaća detaljne podatke o odabranoj šetnji iz baze podataka te ih prikazuje vlasniku.

Vlasnik zatim pregledava podatke o šetnji te unosi **datum i vrijeme rezervacije**. Tada vlasnik može odlučiti da nije zadovoljan odabirom, čime se proces završava bez spremanja podataka, ili može potvrditi nastavak izrade rezervacije.

Ako je vlasnik zadovoljan, podatci o rezervaciji se zatim spremaju u bazu podataka. Tada se rezervacija stavlja u stanje čekanja.

Nakon toga, web aplikacija obavještava vlasnika o ishodu rezervacije:

- u slučaju **odobrene rezervacije**, vlasnik se obavještava o prihvaćanju te mu se daje opcija za plaćanje,
- u slučaju **odbijene rezervacije**, vlasnik se obavještava o odbijanju, promjena se spremi u bazu i proces završava.

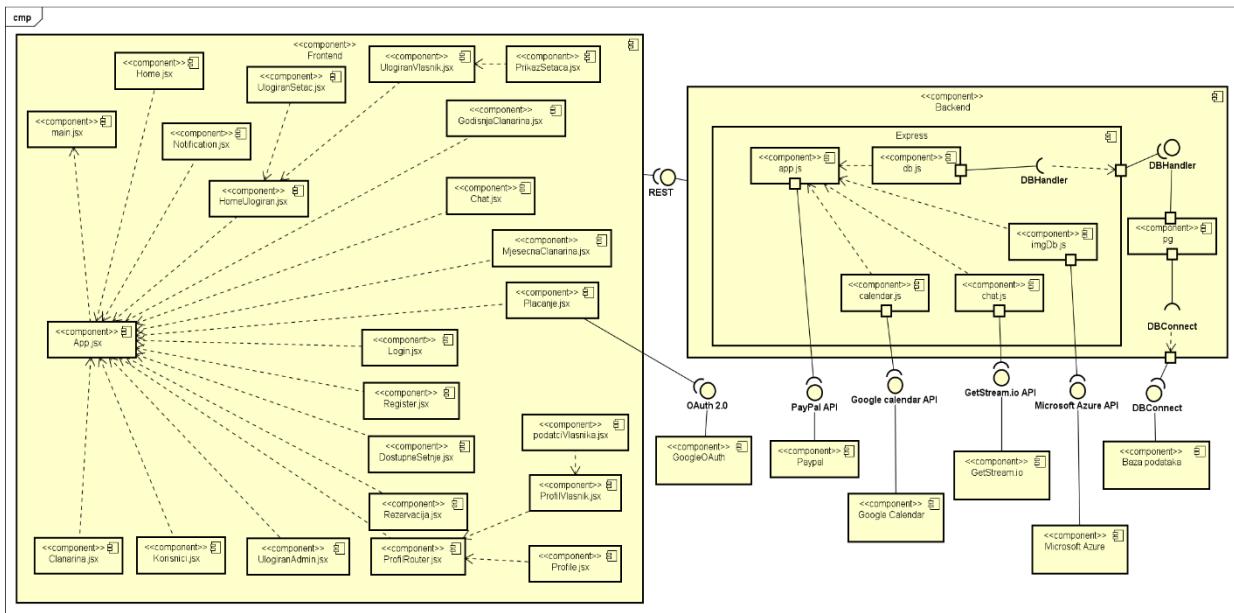
Ako je rezervacija prihvaćena, vlasnik unosi **podatke za plaćanje**. Web aplikacija provodi **obradu plaćanja** te ovisno o ishodu:

- u slučaju **uspješnog plaćanja**, vlasnik se obavještava o uspješnom plaćanju,
- u slučaju **neuspješnog plaćanja**, vlasnik se obavještava o neuspjehu te ima mogućnost ponovnog unosa podataka za plaćanje ili odustajanja od rezervacije.

U oba slučaja promjene o rezervaciji se spremaju u bazu podataka i time se proces završava.

Arhitektura komponenata i razmještaja

Dijagram komponenti



Dijagram komponenti prikazuje strukturu web aplikacije te međusobne odnose između klijentske i poslužiteljske strane sustava. Arhitektura je podijeljena na **Frontend** i **Backend**, koji međusobno komuniciraju putem **REST API** sučelja.

Frontend

Frontend predstavlja klijentsku stranu aplikacije i implementiran je pomoću React radnog okvira. Središnja komponenta frontend-a je **App.jsx**, koja povezuje sve ostale funkcionalne komponente aplikacije. Uzlazna točka aplikacije je **main.jsx**.

Frontend sadrži velik broj komponenti:

- Autentifikacija i korisnički računi:** Login.jsx, Register.jsx, Profile.jsx, ProfilRouter.jsx, ProfilVlasnik.jsx, podaciVlasnika.jsx
- Prikaz sadržaja:** HomeUlogiran.jsx, UlogiranSetac.jsx, UlogiranVlasnik.jsx, UlogiranAdmin.jsx, PrikazSetaca.jsx
- Šetnje i rezervacije:** DostupneSetnje.jsx, Rezervacija.jsx
- Plaćanja i članarine:** Plaćanje.jsx, MjesecnaClanarina.jsx, GodisnjaClanarina.jsx, Clanarina.jsx
- Komunikacija i obavijesti:** Chat.jsx, Notification.jsx
- Administracija:** Korisnici.jsx

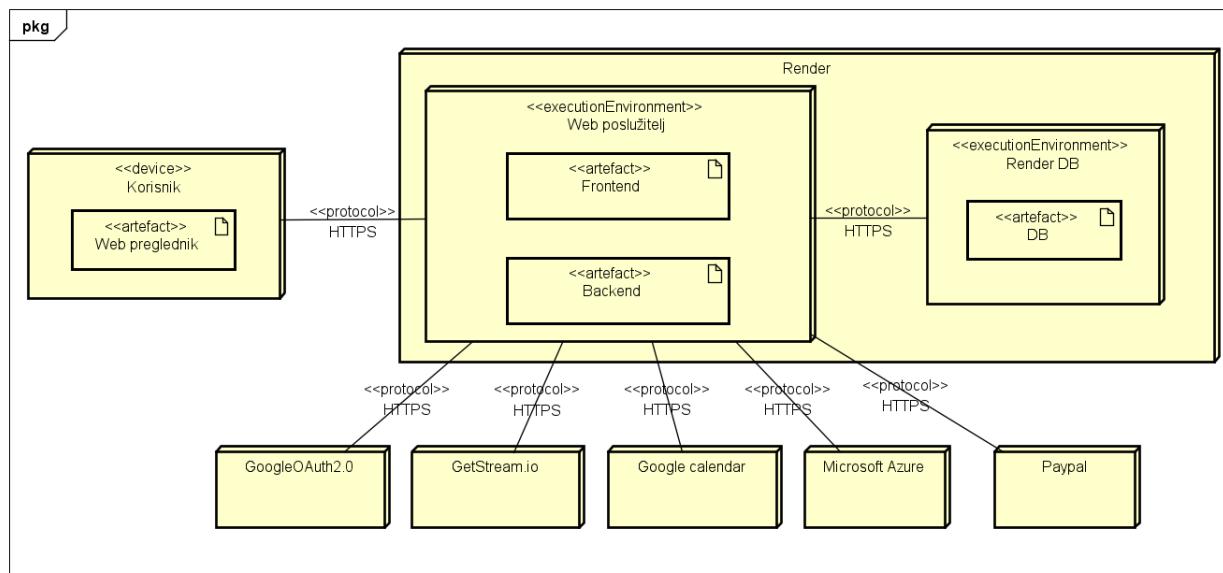
Backend

Backend je implementiran korištenjem **Express.js** radnog okvira. Glavna komponenta backenda je **app.js**, koja obrađuje REST zahtjeve pristigne s frontenda i koordinira rad ostalih komponenti.

Backend se sastoji od sljedećih komponenti:

- **db.js** – zadužen za komunikaciju s bazom podataka putem **pg** modula
- **imgDb.js** – spremanje i dobavljanje slika iz baze
- **calendar.js** – integracija s Google Calendar API-jem
- **chat.js** – chat funkcionalnosti

Dijagram razmještaja



Dijagram razmještaja prikazuje fizičku arhitekturu sustava, odnosno način na koji su pojedine komponente raspoređene na uređaje te kako međusobno komuniciraju.

Na strani korisnika nalazi se **uredaj korisnika** na kojem se izvršava **web preglednik**. Preko web preglednika korisnik pristupa aplikaciji. Komunikacija između web preglednika i poslužitelja odvija se korištenjem **HTTPS protokola**, čime je osigurana sigurna razmjena podataka.

Središnji dio sustava smješten je unutar okruženja **Render**, koje služi kao hosting platforma. Unutar tog okruženja nalaze se dva izvršna okruženja:

- **Web poslužitelj**

U ovom izvršnom okruženju nalaze se dva artefakta:

- **Frontend**, koji je odgovoran za prikaz korisničkog sučelja i interakciju s korisnikom
- **Backend**, koji sadrži poslovnu logiku aplikacije i obrađuje zahtjeve pristigne s frontenda

Frontend i backend međusobno komuniciraju unutar istog poslužiteljskog okruženja, dok prema vanjskim servisima i bazi podataka komuniciraju putem HTTPS protokola.

- **Render DB**

U zasebnom izvršnom okruženju nalazi se **baza podataka (DB)**, koja je zadužena za trajnu pohranu podataka aplikacije.

Backend aplikacija integrira nekoliko vanjskih servisa, s kojima komunicira putem HTTPS protokola:

- **Google OAuth 2.0** – koristi se za autentifikaciju i prijavu korisnika
- **Microsoft Azure** – koristi se za pohranu slika
- **Google Calendar** – koristi se za upravljanje terminima i događajima
- **GetStream.io** – koristi se za komunikaciju korisnika
- **PayPal** - koristi se za plaćanje

Ispitivanje programskog rješenja

Dokumentacija integracijskih ispitanja - Backend API

Ovaj dokument opisuje integracijske testove provedene nad backend sustavom. Testovi koriste supertest za simulaciju HTTP zahtjeva i jest za lažiranje vanjskih ovisnosti poput baze podataka i pohrane datoteka.

Ispitivanje komponenti (Integracijski testovi)

Test 1: test_dobioSveUsere

Vrati listu svih korisnika sa statusom 200.

Ispitni slučaj:

- **Ulazni podaci:** GET /api/users
 - **Očekivani rezultati:**
 - HTTP status je **200**.
 - Odgovor sadrži listu mockiranih korisnika (Marko, Iva).
 - Metoda db.getAllUsers pozvana je točno jednom.
 - **Dobiveni rezultati:** Test prolazi.
-

Test 2: test_neuspjeliUpitBaze

Vrati 500 ako upit bazi podataka ne uspije.

Ispitni slučaj:

- **Priprema:** db.getAllUsers lažiran je da baca grešku "Database connection failed".
 - **Očekivani rezultati:**
 - HTTP status je **500**.
 - Odgovor sadrži grešku "Internal server error".
 - **Dobiveni rezultati:** Test prolazi.
-

Test 3: test_jesamLiAutentificiran

Vrati 401 ako korisnik nije autentificiran.

Ispitni slučaj:

- **Ulazni podaci:** GET /api/me bez aktivne sesije.
 - **Očekivani rezultati:**
 - HTTP status je **401**.
 - **Dobiveni rezultati:** Test prolazi.
-

Test 4: testVlasnici_greškaBazePodataka

Vrati 500 i zabilježi pogrešku pri neuspjehu baze podataka.

Ispitni slučaj:

- **Ulazni podaci:** GET /api/vlasnici
 - **Očekivani rezultati:**
 - HTTP status je **500**.
 - console.error je pozvan kako bi zabilježio pogrešku.
 - **Dobiveni rezultati:** Test prolazi.
-

Test 5: testSetnje_dohvatiŠetačalŠetnje

Vrati 200 s podacima o šetaču i njegovim šetnjama.

Ispitni slučaj:

- **Ulazni podaci:** ID šetača u URL-u (npr. /api/setnje/5).
 - **Očekivani rezultati:**
 - HTTP status je **200**.
 - Objekt šetača sadrži točno ime i niz dostupnih šetnji.
 - **Dobiveni rezultati:** Test prolazi.
-

Test 6: test_uspješnoAžuriranje

Treba ažurirati šetnju i vratiti ažurirani objekt.

Ispitni slučaj:

- **Ulazni podaci:** PUT /api/setnje/101 s novim podacima (cijena, tip, trajanje).
- **Očekivani rezultati:**
 - HTTP status je **200**.

- Vraćeni objekt odgovara poslanim podacima.
 - Metoda db.updateSetnja pozvana s ispravnim parametrima.
 - **Dobiveni rezultati:** Test prolazi.
-

Test 7: test_izbrišiProfil

Vrati 401 ako korisnik u sesiji ne postoji.

Ispitni slučaj:

- **Ulazni podaci:** DELETE /api/delete-profile bez session objekta.
 - **Očekivani rezultati:**
 - HTTP status je **401**.
 - Greška: "Not authenticated".
 - **Dobiveni rezultati:** Test prolazi.
-

Test 8: test_uspješnoBrisanjeProfila

Treba obrisati korisnika i uništiti sesiju nakon uspjeha.

Ispitni slučaj:

- **Priprema:** Korisnik s ID-jem 42 postoji u sesiji i bazi.
 - **Očekivani rezultati:**
 - HTTP status je **200**.
 - Metoda db.deleteUserWithId pozvana s ID-jem 42.
 - Poruka: "Profile deleted successfully".
 - **Dobiveni rezultati:** Test prolazi.
-

Test 9: testIzbrišiProfil_korisnikNijePronađen

Treba vratiti 404 ako korisnik iz sesije ne postoji u bazi podataka.

Ispitni slučaj:

- **Priprema:** Sesija postoji, ali db.getUserWithEmail vraća null.
- **Očekivani rezultati:**
 - HTTP status je **404**.

- Poruka: "User not found".
 - **Dobiveni rezultati:** Test prolazi.
-

Test 10: testIzbriđiProfil_greškaBaze

Treba vratiti 500 ako brisanje iz baze podataka ne uspije.

Ispitni slučaj:

- **Priprema:** db.deleteUserWithId mockiran da baci iznimku.
- **Očekivani rezultati:**
 - HTTP status je **500**.
 - Poruk: "Internal server error".
- **Dobiveni rezultati:** Test prolazi.

Korištene tehnologije i alati

U ovom poglavlju opisane su tehnologije korištene u razvoju naše web aplikacije. Cilj je olakšati održavanje, proširenje i suradnju u timu, te jasno prikazati svrhu i verzije korištenih alata.

Tablica 7.01: Korištene tehnologije i alati

Redni broj	Ime tehnologije/ alata	Verzija	Kategorija
01	JavaScript	ES2023	programski jezik
02	Node.js	22.10.5	radni okvir
03	Express.js	4.18.2	radni okvir
04	React	18	radni okvir
05	HTML5	-	frontend markup
06	CSS3	-	stiliziranje
07	PostgreSQL	16	baza podataka
08	Git	2.43	razvojni alat
09	Visual Studio Code	1.96.3	razvojni alat
10	Render	-	alat za razmještavanje

Programski jezici

JavaScript (ES2023)

JavaScript je univerzalni jezik za web razvoj, korišten za razvoj frontend i backend dijela aplikacije. Verzija ES2023 koristi moderne sintakse i značajke za efikasan razvoj i održavanje.

Radni okviri i biblioteke

Node.js (verzija 22.10.5)

Node.js je runtime okruženje za izvođenje JavaScript koda izvan preglednika. Koristi se za backend razvoj i upravljanje ovisnostima preko npm-a.

Express.js (verzija 4.18.2)

Express je minimalistički Node.js framework za razvoj REST API-ja i server-side logike. Omogućava brzo postavljanje ruta i obradu zahtjeva.

PostgreSQL (verzija 16)

PostgreSQL je moćan sustav za upravljanje relacijskim bazama podataka. Koristi se kao glavna baza podataka zbog svoje pouzdanosti, skalabilnosti i podrške za složene upite.

React (verzija 18)

React je JavaScript biblioteka za razvoj interaktivnih korisničkih sučelja. Omogućuje izradu komponenti koje se mogu ponovno koristiti, čime se povećava učinkovitost frontend razvoja.

HTML5 i CSS3

HTML5 se koristi za strukturiranje sadržaja web stranica, a CSS3 za stiliziranje i responzivni dizajn sučelja.

Razvojni alati

Visual Studio Code (verzija 1.96.3)

IDE za razvoj frontend i backend dijela aplikacije, s podrškom za proširenja i debugging.

Git (verzija 2.43)

Alat za verzioniranje koda, omogućava suradnju tima i praćenje promjena.

Alati za razmještavanje

Render platforma za razmještaj aplikacija i baza podataka. Koristi se za hostanje Node.js aplikacije i PostgreSQL baze, pružajući jednostavan deployment, SSL certifikate i upravljanje okruženjem bez potrebe za Docker konfiguracijom.

Dodatni alati za suradnju i planiranje

- WhatsApp
- Discord

Upute za puštanje u pogon

1. Instalacija

- **Preduvjeti:** Node.js (22.10.5), Git (2.43), Express.js (4.18.2)
- **Preuzimanje:** Preuzmite kod s main brancha (instalirajte zip ili ga klonirajte)

Primjer:

```
git clone https://github.com/EmaZidar/PsiciUOphodnji.git
```

```
cd PsiciUOphodnji
```

U server i client folderu moramo instalirati ovisnosti pomoću naredbe

```
npm install
```

2. Postavke

Detaljne upute za konfiguraciju aplikacije:

.env file treba se nalaziti u server folderu i treba imati zadane sljedeće attribute:

PORT=8000

GOOGLE_CLIENT_ID=

GOOGLE_CLIENT_SECRET=

GOOGLE_OAUTH_URL=<https://accounts.google.com/o/oauth2/v2/auth>

GOOGLE_ACCESS_TOKEN_URL=<https://oauth2.googleapis.com/token>

GOOGLE_TOKEN_INFO_URL=<https://oauth2.googleapis.com/tokeninfo>

DATABASE_URL=

AZURE_STORAGE_CONNECTION_STRING=

STREAM_API_KEY=

STREAM_SECRET_KEY=

ADMIN_EMAIL=

VITE_PAYPAL_CLIENT_ID=

3. Pokretanje aplikacije

Razvojno okruženje:

- Frontend aplikacije pokrećemo tako što u client folderu pokrenemo:

```
npm run dev
```

- Backend aplikacije pokrećemo tako što u server folderu pokrenemo:

```
npm run start
```

- **Provjera rada:** Upišite "localhost:5173" u web preglednik.

4. Upute za administratore

Smjernice za administratore aplikacije nakon puštanja u pogon:

Pristup administratorskom sučelju

- Administratorskom sučelju pristupa se dodavanjem **/admin** na URL aplikacije
- Pristup je moguć **samo ako je korisnik prijavljen sa email adresom koje je navedena u .env datoteci (ADMIN_EMAIL)**

Redovito održavanje:

- Osvježavanje baze podataka
- Ažuriranje aplikacije
- Ažuriranje mjesecnih i godišnjih članarina za korisnike

5. Primjer za Render platformu (Cloud Deploy)

U client folderu nalaze se render.yaml datoteka za automatsku izgradnju aplikacije.

- Postavljanje na Render
 - Prijavite se na Render.
 - Kreirajte novi Web Service i povežite ga s vašim GitHub repozitorijem pomoću Deploy Hooka koji osigurava da će se aplikacija postaviti na Render nakon izgradnje.
 - Dodajte environmental varijable.
- Pokretanje aplikacije Render će automatski preuzeti repozitorij, instalirati ovisnosti i pokrenuti aplikaciju. Nakon deploja, aplikaciji možete pristupiti putem generiranog URL-a.

Opis prisutpa aplikaciji na javnom poslužitelju

Korisnici mogu pristupiti aplikaciji unosom URL-a aplikacije (<https://psiciuophodnji-1-zrvs.onrender.com/>) u web preglednik.

Za korištenje svih funkcionalnosti aplikacije potrebno je **prijaviti se u sustav**. Nakon prijave korisnik, ovisno o ulozi (vlasnik ili šetač), dobiva pristup odgovarajućim opcijama:

- vlasnici mogu pregledavati dostupne šetnje i izrađivati rezervacije
- šetači mogu kreirati šetnje te prihvati ili odbijati pristigne rezervacije

Ograničenja

- Neprijavljeni korisnici imaju ograničen pristup funkcionalnostima (samo pregled početne stranice).
- Administratorske funkcije nisu dostupne običnim korisnicima.

Pristup administratorskom sučelju

Administratorskom sučelju pristupa se dodavanjem nastavka **/admin** na osnovni URL aplikacije.

Zaključak i budući rad

Ciljevi projekta

Izrada projektnog zadatka *Psići U Ophodnji* bila je zahtjevan, ali vrlo koristan proces kroz koji su stečena brojna praktična znanja. Cilj projekta bio je razviti aplikaciju koja je namijenjena vlasnicima pasa i šetačima te im omogućuje jednostavno povezivanje kroz intuitivno sučelje. Vlasnici pasa mogu pronaći dostupne šetače, pregledati njihove šetnje i napraviti rezervaciju, dok šetači imaju mogućnost objavljivanja šetnji i upravljanja pristiglim rezervacijama. Tako aplikacija olakšava organizaciju šetanja pasa te omogućuje šetačima da za pruženu uslugu budu adekvatno plaćeni. Dodatno, sustav uključuje integraciju s vanjskim servisima kako bi se proširile funkcionalnosti i unaprijedilo korisničko iskustvo.

Izazovi tijekom razvoja

Tijekom izrade projekta kao najveći izazov pokazala se **organizacija vremena**. Razvoj aplikacije odvijao se paralelno s ostalim akademskim obvezama, zbog čega je bilo potrebno prilagođavati plan rada. Pojedine funkcionalnosti implementirane su postupno, što je omogućilo kontinuirani napredak projekta, ali je ujedno istaknuto važnost dobre početne pripreme i realne procjene opsega posla. Iako je osnovna funkcionalnost sustava postignuta, određeni elementi mogli su biti implementirani brže uz veću početnu pripremu ili detaljniju analizu potencijalnih problema.

Medu ostalim izazovima, najzahtjevnijima bila je **implementacija prijave korisnika putem Google OAuth 2.0**, koja je zahtjevala razumijevanje autentifikacijskih protokola te rada s vanjskim servisima, pošto je to bio prvi vanjski servis s kojim smo se susreli pri izradi aplikacije. Također, deployment aplikacije na Render predstavlja je težak korak u razvoju, jer je uključivao konfiguraciju produkcijskog okruženja te prilagodbu aplikacije za web poslužitelja. Također, implementacija **chat funkcionalnosti** odrađena je pred kraj razvoja projekta, što je zahtjevalo dodatno povezivanje s postojećim dijelovima sustava i bazom podataka.

Funkcionalnosti koje nismo napravili su:

- Postavljanje slika u chatu
- Napredne funkcionalnosti kalendar

Dodatna funkcionalnost koju imamo je brisanje i odjavu profila.

Stečena znanja

Tijekom rada na projektu stečena su brojna praktična znanja i vještine,

- razvoj web aplikacija korištenjem tehnologija poput Reacta, Node.js-a,
- rad s bazom podataka PostgreSQL,
- korištenje alata za kolaboraciju poput GitHub-a
- integraciju vanjskih servisa Osim tehničkih znanja, projekt je doprinio razvoju vještina timskog rada, komunikacije, rješavanja problema i upravljanja s vremenom.

Nastavak rada na aplikaciji

Iako su ostvareni svi ključni ciljevi projekta, dio planiranih funkcionalnosti nije implementiran zbog vremenskih ograničenja. Ti nedostatci predstavljaju priliku za buduće nadogradnje.

Literatura

Kontinuirano osvježavanje Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

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>
7. MDN dokumentacija, <https://developer.mozilla.org/en-US/docs/Web>
8. Online web tutorials, <https://www.w3schools.com/>
9. React dokumentacija, <https://react.dev/reference/react>
10. Bro Code React Course, <https://www.youtube.com/watch?v=CgkZ7MvWUAA&t=13411s>
11. Jest testovi, <https://jestjs.io/docs/getting-started>
12. Jest testovi, <https://www.geeksforgeeks.org/javascript/testing-with-jest/>
13. Jest testovi, <https://www.testim.io/blog/jest-testing-a-helpful-introductory-tutorial/>
14. Supertest, <https://www.npmjs.com/package/supertest>

Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatak	Autori	Datum
0.1	napravljen predložak	Nina Zamberlin	24.10.2025.
0.2	dodani funkcijski zahtjevi	Nina Zamberlin, Lana Šapić, Luka Zadro	24.10.2025.
0.3	usklađivanje imena funkcijskih zahtjeva s imenima obrazaca uporabe	Ema Zidar, Lana Šapić	1.11.2025.
0.4	dodani dionici, analiza zahtjeva	Ema Zidar	1.11.2025.
0.5	UML dijagrami	Lana Šapić	1.11.2025.
0.6	prvi dio opisa obrazaca uporabe	Lana Šapić	2.11.2025.
0.7	povezivanje broja obrazaca uporabe s funkcionalnih zahtjevima po nazivu (tablica)	Lana Šapić	2.11.2025.
0.8	opisi tablica baze podataka + slike dijagrama i sheme	Nina Zamberlin	3.11.2025.
0.9	napisan detaljnije opis projekta i home page te popis sastanaka	Ema Zidar	5.11.2025.
1.0	dodana arhitektura	Ema Zidar	5.11.2025.
1.1	dodan plan aktivnosti i tablica rada	Ema Zidar	5.11.2025.
1.2	napisane tehnologije za implementaciju aplikacije	Ema Zidar	5.11.2025.
1.3	izbacen token iz tablica i slika baze podataka	Nina Zamberlin	5.11.2025.
1.4	promijenjeni nefunkcionalni zahtjevi, podjela na kategorije, uređeni funkcionalni zahtjevi, promjena razine prioriteta	Ema Zidar	5.11.2025.
1.5	dovrseni opisi obrazaca uporabe, dodani F024-026	Lana Šapić	6.11.2025.
1.6	dodan F-003 za main page	Lana Šapić i Nina Zamberlin	13.11.2025.
1.7	dodan dijagram razreda	Viktor Wilder i Lana Šapić	13.11.2025.
1.8	dodani opisi za sekvencijske dijagrame	Nina Zamberlin	14.11.2025.

Opis promjene/dodataka	Autori	Datum
Rev.		
1.9 dodan izvanredni sastanak i uredeni nefunkcionalni zahtjevi	Ema Zidar	14.11.2025.
4.1 dodan dijagram stanja	Jakov Ramljak	12.1.2026.
4.2 dodan dijagram aktivnosti	Jakov Ramljak	15.1.2026.
4.3 popravljen dijagram stanja i aktivnosti	Jakov Ramljak	18.1.2026.
5.1 dodan dijagram razmještaja	Jakov Ramljak	21.1.2026.
C.1 Dodani sastanci drugog ciklusa	Jakov Ramljak	22.1.2026.
C.2 Tablica aktivnosti	Jakov Ramljak	22.1.2026.
5.2 dodan dijagram komponenti	Jakov Ramljak	22.1.2026.
8.1 dodane upute za korištenje	Jakov Ramljak	22.1.2026.
9.1 Dodan zaključak	Jakov Ramljak	23.12.2026.
6.1 dodano ispitivanje programskog rješenja	Viktor Wilder	23.1.2026.

Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

Ovdje se redovito bilježe sastanci i tjedni izvještaji aktivnosti. Cilj je prikazati ključne teme, donesene odluke i zaduženja članova tima.

Sastanak 1

Datum: 10. listopada 2025.

Prisustvovali: Ema Zidar, Jakov Ramljak, Nina Zamberlin, Lana Šapić, Luka Zadro, Leonard Zadro, Viktor Wilder

Teme sastanka:

- Organizacija tima i raspodjela uloga
 - LOŠE: nejasna raspodjela zadataka
 - BOLJE: definirani voditelji modula i odgovornosti
- Postavljanje GitHub repozitorija i osnovnih informacija o samome projektu (README, LICENSE, ČLANOVI)

Donesene odluke i zaduženja:

- Ema Zidar – koordinacija tima
 - Jakov & Viktor – frontend inicijalni setup
 - Nina – baza podataka, kreiranje schema
-

Sastanak 2

Datum: 17. listopada 2025.

Prisustvovali: svi članovi tima

Teme sastanka:

- Razrada funkcionalnih zahtjeva
 - Dokumentacija zahtjeva

Donesene odluke i zaduženja:

- Ema – front page izrada
- Nina – definicija baze podataka s obzirom na zahtjeve
- Leo & Luka – osnovni API endpointi

Sastanak 3

Datum: 24. listopada 2025.

Prisustvovali: svi članovi tima

Teme sastanka:

- Pregled napretka backend razvoja
- Integracija baze podataka i backend API-ja
- Diskusija o autentikaciji i OAuth 2.0

Donesene odluke i zaduženja:

- Leonard – implementacija OAuth 2.0 nakon dorade front page-a s Reactom
 - Ema - dokumentacija
-

Sastanak 4

Datum: 1. studenog 2025.

Prisustvovali: svi članovi tima

Teme sastanka:

- Pregled frontend dizajna
- Pregled i dorada funkcionalnih zahtjeva

Donesene odluke i zaduženja:

- Lana za izradu UML dijagrama, obrasca uporabe
 - Odluka o korištenju Sequelize-a za lakši flow između baza, backenda i frontenda
-

Sastanak 5

Datum: 8. studenog 2025.

Prisustvovali: svi članovi tima

Teme sastanka:

- Postavljanje frontenda pomoću React-a
 - Pregled dokumentacije
-

Sastanak 6

Datum: 14. studenog 2025.

Prisustvovali: svi članovi tima

Teme sastanka:

- odbijanje jedne funkcionalnosti
- izvedba druge u kratkom period
- druga funkcionalnost pregled profila

Donesene odluke i zaduženja:

- Žrtvovali deployment jer inače ne bi stigli imat dvije funkcionalnosti
-

Sastanak 7

Datum: 12. prosinca 2025.

Prisustvovali: svi članovi tima

Teme sastanka:

- Pregled dosadašnjeg napretka
- Planiranje funkcionalnosti profila korisnika(što je za vlasnika, a što za šetača)

Donesene odluke i zaduženja:

- Ema & Jakov – UI profila
 - Nina – Backend za profile vlasnika i šetača
-

Sastanak 8

Datum: 5. siječnja 2026.

Prisustvovali: svi članovi tima

Teme sastanka:

- Funkcionalnost stvaranja šetnje
- Pisanje dokumentacije - dijagrami stanja i aktivnosti

Donesene odluke i zaduženja:

- Backend endpointi za šetnje
 - Luka & Leonard – Backend logika
 - Jakov - Izrada dijagrami stanja i aktivnosti
-

Sastanak 9

Datum: 10. siječnja 2026.

Prisustvovali: svi članovi tima

Teme sastanka:

- Stvaranje i upravljanje rezervacijama
- Povezivanje korisnika, šetnji i termina

Donesene odluke i zaduženja:

- Implementacija osnovnog flowa rezervacije
 - Nina – Tablica rezervacija
 - Viktor – Frontend prikaz termina
-

Sastanak 10

Datum: 15. siječnja 2026.

Prisustvovali: svi članovi tima

Teme sastanka:

- Chat funkcionalnost
- Spremanje i dohvata slika u drugu bazu(Microsoft Azure)
- Pisanje dokumentacije, dijagrami komponenata i razmještaja

Donesene odluke i zaduženja:

- Luka - Backend spremanja slika
 - Leonard – Backend chat-a
 - Nina – Baza slika(Microsoft Azure)
 - Ema – UI chat-a
 - Viktor - Ispitivanje
 - Jakov - Dokumentacija
-

Sastanak 11

Datum: 18. siječnja 2026.

Prisustvovali: svi članovi tima

Teme sastanka:

- Plaćanje i kalendar
- Procjena realnog opsega projekta

Donesene odluke i zaduženja:

- Plaćanje lažna funkcionalnost
 - Lana – Plaćanje
 - Jakov – Frontend kalendar ako ostane vremena
 - Viktor - Ispitivanje i dokumentacija
-

Plan rada

Zadatak	Rok	Zaduženi	Status
Tema projekta	17.10.2025	Cijeli tim	Odrađeno
Dijagrami obrazaca uporabe	1.11.2025	E. Zidar, J. Ramljak, N. Zamberlin	Odrađeno
Nabranje funkcijskih zahtjeva	28.10.2025	Cijeli tim	Odrađeno
Planiranje dizajna	17.10.2025	Frontend podtim (E. Zidar, J. Ramljak, V. Wilder)	Odrađeno
Sekvencijski dijagrami	6.11.2025	L. Šapić, L. Zadro, L. Zadro	Odrađeno
Priprema za prezentiranje napretka	6.11.2025	Cijeli tim	Odrađeno
Izrada početne stranice	29.10.2025	Frontend podtim (E. Zidar, J. Ramljak, V. Wilder)	Odrađeno
Spajanje s bazom podataka	5.11.2025	N. Zamberlin, L. Šapić	Odrađeno
Dizajn i razvoj registracije korisnika	21.12.2025	Backend podtim (L. Zadro, L. Zadro)	Odrađeno
Moderatorski sustav	13.1.2026	Backend podtim (L. Zadro, L. Zadro)	Odrađeno
Testiranje alfa verzije aplikacije	14.1.2026	Cijeli tim	Odrađeno
Implementacija osnovnih funkcionalnosti alfa verzije	14.1.2026	Cijeli tim	Odrađeno
Implementacija sustava za pretragu objava po tagovima	9.1.2026	Backend podtim (L. Šapić, L. Zadro)	Odrađeno
Implementacija sustava notifikacija	17.1.2026	Frontend & Backend podtimovi	Odrađeno
Arhitektura sustava - dijagrami	15.1.2026	Jakov Ramljak	Odrađeno

Zadatak	Rok	Zaduženi	Status
Implementacija chat funkcionalnosti	18.1.2026	L. Zadro, E. Zidar	Odrađeno
Uređivanje i ažuriranje profila korisnika	9.1.2026	Frontend podtim (E. Zidar, J. Ramljak)	Odrađeno
Implementacija sustava plaćanja	23.1.2026	Frontend & Backend podtimovi	Odrađeno
Implementacija baze za spremanje slika	21.1.2026	Luka Zadro	Odrađeno
Arhitektura komponenata - dijagrami	22.1.2026	Jakov Ramljak	Odrađeno
Implementacija pretraživanja šetnji po lokaciji	23.1.2026	Frontend & Backend podtimovi	Odrađeno
Postavaljanje slika u chat		Frontend & Backend podtimovi	–
Implementacija naprednih mogućnosti kalendara		Frontend & Backend podtimovi	–
Testiranje finalne verzije aplikacije	23.1.2026	Cijeli tim	Odrađeno
Provjera finalne verzije projekta	23.1.2026	Cijeli tim	Odrađeno

Napomena:

- Frontend podtim: E. Zidar, J. Ramljak, V. Wilder
- Backend podtim: L. Šapić, L. Zadro, L. Zadro
- Cijeli tim: svi članovi

Tablica aktivnosti

Članovi	E. Zidar	J. Ramljak	N. Zamberlin	L. Šapić	Leonard Zadro	Luka Zadro	V. Wilder
Upravljanje projektom	8	0	0	0	0	0	0
Opis projektnog zadatka	2	2	2	2	2	2	2
Funkcionalni zahtjevi	2	1	2	2	2	2	1
Dijagram obrazaca	2	0	0	8	0	0	0
Sekvencijski dijagrami	0	1	0	0	3	3	1

Članovi	E. Zidar	J. Ramljak	N. Zamberlin	L. Šapić	Leonard Zadro	Luka Zadro	V. Wilder
Opis ostalih zahtjeva	2	0	2	0	0	0	0
Arhitektura i dizajn sustava	3	0	0	1	0	0	1
Baza podataka	0	1	3	3	0	1	0
Dijagram razreda	0	3	2	0	1	0	1
Dijagram stanja	0	4	0	0	1	1	1
Dijagram aktivnosti	0	4	0	0	1	0	0
Dijagram komponenti	0	3	0	0	1	0	1
Korištene tehnologije i alati	1	0	0	0	0	2	3
Ispitivanje programskog rješenja	0	1	0	1	1	0	10
Dijagram razmještaja	0	4	0	0	1	1	0
Upute za puštanje u pogon	0	5	0	1	5	5	0
Dnevnik sastajanja	0	1	0	0	0	0	0
Zaključak i budući rad	0	3	0	0	0	0	0
Popis literature	1	0	1	0	0	1	0
Izrada aplikacije	60	30	60	65	65	70	30
Izrada baze podataka	0	0	10	10	0	0	0
Spajanje s bazom podataka	0	0	0	0	1	5	0
Dokumentiranje plana projekta i uključenosti	5	4	1	1	1	1	1
Dokumentacija specifikacije obrazaca uporabe	2	1	0	1	0	0	0