Programsko inženjerstvo ak.god 2025. /2026.

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

PawPal

Tim: TG13.4

Članovi tima

- **Nikola Ivković**
- **Alan Miljak**
- **Bruno Cavor**
- **Antonio Stjepić**
- **Zvonimir Schönwald**
- **Vid Veselko**

Ime tima: Sarma

Nastavnik: Vlado Sruk

\newpage

1. Opis projektnog zadatka

1.1. Potencijalna korist projekta PawPal

PawPal je inovativna platforma koja povezuje vlasnike pasa sa šetačima pasa, pojednostavljujući i digitalizirajući proces pronalaženja, rezerviranja i plaćanja šetnji. Ovaj projekt donosi višestruke koristi kako za vlasnike pasa, tako i za šetače i administratore platforme.

1.1.1. Prednosti za vlasnike pasa

- **Jednostavno i brzo rezerviranje**
  Platforma omogućuje vlasnicima pasa da brzo pronađu dostupne šetače u svojoj blizini koristeći pretragu po lokaciji, cijeni i ocjenama. Rezervacija šetnje traje svega nekoliko minuta.
- **Transparentnost i sigurnost**
  Svaki šetač ima javan profil s ocjenama i recenzijama, što vlasnicima omogućuje da odaberu provjerene i pouzdane šetače.
- **Fleksibilnost i kontrola**
  Vlasnici mogu birati između individualnih i grupnih šetnji, trajanje, polazišnu adresu te dodati posebne napomene (npr. lijekovi, alergije, dopuštene poslastice). Mogu otkazati šetnju do 24 sata prije bez naknade.
- **Jednostavno plaćanje**
  PawPal podržava plaćanje gotovinom, PayPalom i kreditnim karticama putem sigurnog vanjskog servisa.
- **Komunikacija u stvarnom vremenu**
  Ugrađeni chat omogućuje izravnu komunikaciju i dijeljenje fotografija između vlasnika i šetača tijekom šetnje.

1.1.2. Prednosti za šetače pasa

- **Povećana vidljivost i prilika za zaradu**
  Šetači mogu kreirati javni profil, objavljivati termine i definirati cijene. Time dobivaju priliku za dodatnu zaradu i stvaranje baze klijenata.
- **Automatizacija i organizacija**
  Integracija s Google Calendarom omogućava jednostavno upravljanje terminima, automatske podsjetnike i izbjegavanje preklapanja.
- **Sustav članarina i recenzija**
  Šetači plaćaju mjesečnu ili godišnju članarinu, a bolja ocjena povećava vidljivost i broj rezervacija.

1.1.3. Prednosti za administratore

- **Centralizirano upravljanje sustavom**
  Administratori postavljaju cijenu članarine, nadgledaju korisnike i moderiraju sadržaj.
- **Sigurnost i podrška korisnicima**
  Administrator nadzire transakcije, sprječava zlouporabe i rješava sporove između korisnika.

1.2. Postojeća slična rješenja

Na tržištu postoje globalno poznate platforme poput **Rover** i **Wag!**, koje nude slične usluge povezivanja vlasnika i šetača pasa. Međutim, **PawPal** se izdvaja lokalnim pristupom i jednostavnošću korištenja.

1.2.1. Rover

Rover je jedna od najvećih svjetskih platformi za pronalazak šetača i čuvara pasa. Iako nudi širok spektar usluga, nije prilagođen hrvatskom tržištu, koristi isključivo američke metode plaćanja i ima visoke provizije.

1.2.2. Wag!

Wag! omogućuje rezervacije šetnji i praćenje pasa putem GPS-a, no ograničen je na SAD i Kanadu. Nema podršku za hrvatski jezik ni lokalne opcije plaćanja.

1.2.3. Razlike između PawPal-a i konkurencije

- Lokalizacija i podrška za hrvatski jezik i valutu
- Integracija s Google Calendarom
- Plaćanje gotovinom, PayPalom i karticama
- Chat i razmjena fotografija u stvarnom vremenu
- Otkazivanje rezervacije do 24 sata prije početka
- Jednostavan sustav članarina bez provizija po rezervaciji

1.3. Skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje

- **Vlasnici pasa**
  Primarna skupina korisnika. Osobe koje zbog posla ili obaveza nemaju dovoljno vremena za redovite šetnje. PawPal im omogućuje jednostavno pronalaženje provjerenih šetača i praćenje statusa rezervacija.
- **Šetači pasa**
  Sekundarna skupina. Ljubitelji životinja, studenti ili profesionalni treneri koji žele dodatni prihod putem oglašavanja svojih usluga.
- **Turisti**
  Putnici koji sa sobom vode ljubimce mogu koristiti PawPal za privremene šetnje tijekom boravka u novom gradu.
- **Starije osobe i osobe s invaliditetom**
  Omogućuje im lakšu brigu o ljubimcima uz pomoć pouzdanih šetača.
- **Administratori i organizacije**
  Mogu koristiti platformu za upravljanje korisnicima, moderaciju sadržaja i podršku sustavu.

---

s

1.4. Mogućnost prilagodbe rješenja

- Podrška za više jezika i valuta (hrvatski, engleski, njemački)
- Integracija dodatnih servisa (lokalni platni sustavi, e-mail obavijesti)
- Proširenje na druge vrste ljubimaca (mačke, zečevi i dr.)
- Implementacija personaliziranih preporuka pomoću AI analitike
- Razvoj mobilnih aplikacija (Android/iOS) za jednostavniji pristup

---

1.5. Opseg projektnog zadatka

- **Frontend sučelje** – React uz JavaScript, HTML i CSS. Omogućuju pretraživanje šetača, pregled profila i rezervaciju šetnji.
- **Backend poslovna logika** – C# s ASP.NET Core i PostgreSQL bazom podataka. Upravlja korisnicima, rezervacijama i plaćanjima.
- **Autentifikacija i plaćanja** – OAuth 2.0 (Google prijava), PayPal i kreditne kartice.
- **Integracija s Google Calendarom** – sinkronizacija termina između vlasnika i šetača.
- **Chat funkcionalnost** – komunikacija i dijeljenje fotografija (FreeChat).
- **Administratorski panel** – upravljanje članarinama, recenzijama i korisnicima.

1.6. Moguće nadogradnje projektnog zadatka

- Mobilna aplikacija s push obavijestima i GPS praćenjem šetnji
- Praćenje rute šetnje u stvarnom vremenu
- AI sustav preporuka šetača prema preferencijama korisnika
- Nagradni program za aktivne korisnike

- Integracija s veterinarima i trgovinama za kućne ljubimce
- Analitički panel s prikazom rezervacija, ocjena i prihoda

\newpage

2. Analiza zahtjeva

2.1. Funkcionalni zahtjevi

2.1.1. Autentifikacija i korisnički računi

Oznaka	Naziv	Opis	Prioritet	Kriterij Prihvaćanja
FR1	Registracija korisnika u sustav	Aplikacija omogućuje korisniku da se registrira pomoću Google računa kako bi mogao koristiti sustav.	Visoki	Sustav uspješno registrira korisnika u sustav.
FR2	Prijava korisnika u sustav	Aplikacija omogućuje korisnik da se prijavljuje u sustava kako bi ga mogao koristiti.	Visoki	Sustav uspješno prijavljuje korisnika u sustav.
FR3	Uređivanje profila korisnika	Aplikacija omogućuje korisnik da unosi i uređuje svoje podatke kroz web sučelje.	Srednji	Korisnik uspješno uređuje svoj korisnički profil.

2.1.2. Profil šetača i vlasnika

Oznaka	Naziv	Opis	Prioritet	Kriterij Prihvaćanja
FR4	Profil šetača	Aplikacija prikazuje osnovne informacije, ocjene, termine i cijene.	Visoki	Sustav točno prikazuje sve potrebne informacije.
FR5	Profil psa	Aplikacija omogućuje vlasniku dodavanje jednog ili više pasa s opisom i napomenama.	Visoki	Vlasnik uspješno dodaje ili uređuje profile svojih pasa.
FR6	Pretplata vlasnika na obavijesti o novim šetačima	Aplikacija omogućuje vlasniku da se pretplati na obavijesti o novim šetačima.	Srednji	Sustav obavještava pretplaćene vlasnike o svakom novoregistriranom šetaču.

2.1.3. Pretraživanje i rezervacija

Oznaka	Naziv	Opis	Prioritet	Kriterij Prihvaćanja
FR7	Pretraživanje šetača	Aplikacija omogućuje korisniku filtriranje po lokaciji, cijeni i ocjeni šetače putem web preglednika.	Visoki	Korisnik uspješno pretražuje sustav po odabranim stavkama.
FR8	Rezervacija termina	Aplikacija omogućuje korisniku rezervacije termina putem web sučelja pri čemu korisnik bira termin šetnje u kalendaru te odabire detalje, tj. datum šetnje, vrijeme šetnje, trajanje šetnje, tip šetnje (grupna/individulna), adresa šetnje i napomene te poslijetku korisnik to sve mora i potvrditi.	Visoki	Korisnik uspješno rezervira termin i odabire detalje šetnje.
FR9	Otkazivanje rezervacije	Aplikacija omogućuje korisniku mogućnost otkazivanja termina do 24 sata prije početka šetnje putem web forme.	Visoki	Korisnik uspješno otkazuje svoju rezervaciju.
FR10	Prihvaćanje i odbijanje rezervacije	Nakon što korisnik napravi rezervaciju termina šetnje, sustav obavještava šetača o rezervaciji termina šetnje te mu pruža mogućnost odbijanja ili prihvaćanja rezervacije.	Visoki	Sustav obavještava šetača o rezervaciji te šetač odbija ili prihvaća rezervaciju.

2.1.4. Komunikacija i recenzije

Oznaka	Naziv	Opis	Prioritet	Kriterij Prihvaćanja
FR11	Chat funkcionalnost	Aplikacija omogućuje komunikaciju između šetača i vlasnika putem chata nakon što šetač potvrdi rezervaciju.	Visoki	Komunikacija uspješno ostvarena putem chata.
FR12	Ocjene i recenzije	Aplikacija daje mogućnost ocjenjivanja(od 1 do 5) i recenziranja šetača.	Srednji	Vlasnici uspješno ostavljaju recenzije o šetačima nakon šetnje.

Oznaka	Naziv	Opis	Prioritet	Kriterij Prihvaćanja
FR13	Prikaz prosjeka ocjena na profilu šetača	Aplikacija omogućuje korisniku na profilu šetača da ima uvid u ocjene i recenzije.	Srednji	Uspješan prikaz ocjena na profilu šetača.

2.1.5. Plaćanja i članarine

Oznaka	Naziv	Opis	Prioritet	Kriterij Prihvaćanja
FR14	Plaćanje članarine	Aplikacija omogućuje šetaču plaćanje naknade za korištenje sustava putem elektronskog plaćanja, preciznije putem kartičnog plaćanja ili PayPala.	Visoki	Šetač uspješno plaća članarinu putem elektronskog plaćanja.
FR15	Plaćanje šetnje	Aplikacija daje mogućnost odabira načina plaćanja šetnje, odnosno odabir elektronskog plaćanja ili plaćanja gotovinom.	Visoki	Korisnik uspješno plaća uslugu šetnje.
FR16	Upravljanje plaćanjima, obavijestima i komentarima	Aplikacija omogućuje administratoru da unosi cijene članarina, izdaje obavijesti i upravlja komentarima.	Visoki	Administrator uspješno upravlja cijenama članarina, izdaje obavijesti i upravlja komentarima.

2.2. Nefunkcionalni zahtjevi

2.2.1. Zahtjevi performansi

Oznaka	Opis	Prioritet
NFR1	Aplikacija Aplikacija omogućuje posluživanje do 1000 istovremenih korisnika bez pogoršanja performansi.	Visoki

2.2.2. Zahtjevi sigurnosti

Oznaka	Opis	Prioritet
NFR2	Aplikacija ima HTTPS enkripciju, validacija inputa, CSRF i XSS zaštitu.	Visoki

2.2.3. Zahtjevi za održavanje

Oznaka	Opis	Prioritet
NFR3	Aplikacija zahtjeva da se kod za backend i frontend piše kao dvije zasebne cjeline kako bi se olakšalo održavanje.	Visoki

2.2.4. Zahtjevi za korisničko iskustvo

Oznaka	Opis	Prioritet
NFR4	Aplikacija osigurava da sučelje mora imati responzivan dizajn.	Visoki

2.2.5. Zahtjevi za oporavak

Oznaka	Opis	Prioritet
NFR5	Aplikacija omogućuje sigurnosne kopije podataka u sustavu svakih 60 minuta.	Visoki
NFR6	Aplikacija ima mogućnost oporavka za manje od 10 minuta.	Visoki

2.2.6. Domenski zahtjevi

Oznaka	Opis	Prioritet
NFR7	Aplikacija poštuje GDPR-a (prilikom prikupljanja podataka) i sličnih zakona vezanih za privatnost podataka.	Visoki

2.3. Dionici i njihovi funkcionalni zahtjevi

1. Korisnici sustava - krajnji korisnik koji koristi aplikaciju kako bi pronašao šetnju za psa
2. Šetači - osobe koje nude usluge šetanja pasa
3. Administratori - osobe koje nadziru rad samog sustava, moderiraju sadržaj u sustavu te brinu da sustav funkcionira
4. Razvojni tim - tim odgovoran za održavanje, unaprjeđenje i razvoj samog sustava

2.3.1. Korisnici sustava

- Napraviti korisnički račun koji sadrži osnovne informacije o njihovom psu pri čemu jedan korisnik može imati više pasa
- Pretražiti i filtrirati ponuđene šetnje
- Pretražiti i filtrirati ponuđene šetače

- Rezervirati željenu šetnju
- Otkazati šetnju 24 sati prije šetnje (u zadnjih 24 sata nije moguće otkazati šetnju)
- Odabrati način plaćanja šetnje
- Komunicirati sa šetačem putem chata
- Ocijeniti šetača
- Ostaviti povratnu informaciju nakon šetnje u obliku komentara ili slike
- Pretraživati i filtrirati šetače
- Pretplatiti na obavijesti o novim šetačima
- Platiti šetnju putem elektronskog plaćanja ako oblik plaćanja nije gotovina

2.3.2. Šetači

- Napraviti korisnički račun koji sadrži osnovne informacije o šetaču, sliku te popis oglašenih šetnja
- Oglasiti šetnju i osnove informacije o šetnji (tip šetnje, cijena i trajanje)
- Mogućnost prihvaćanja ili odbijanja rezervacije za šetnjom
- Komunikacija sa korisnikom putem chata
- Plaćanje mjesečne ili godišnje članarine elektronskim plaćanjem

2.3.3. Administratori

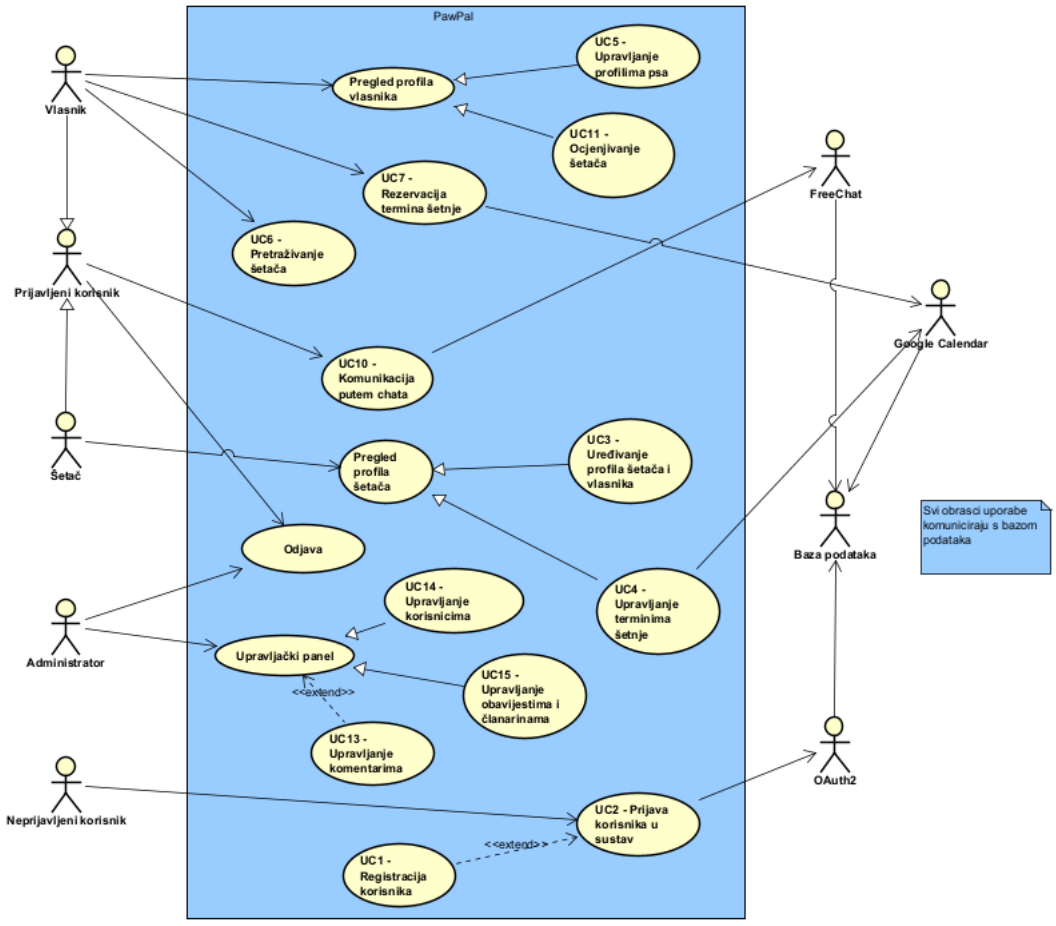
- Prijaviti se u upravljački panel sustava
- Obrisati i suspendirati račune korisnika ili šetača
- Moderirati komentare u sustavu
- Izdavati obavijesti u sustav
- Mijenjati iznos članarine

\newpage

3. Specifikacija zahtjeva sustava

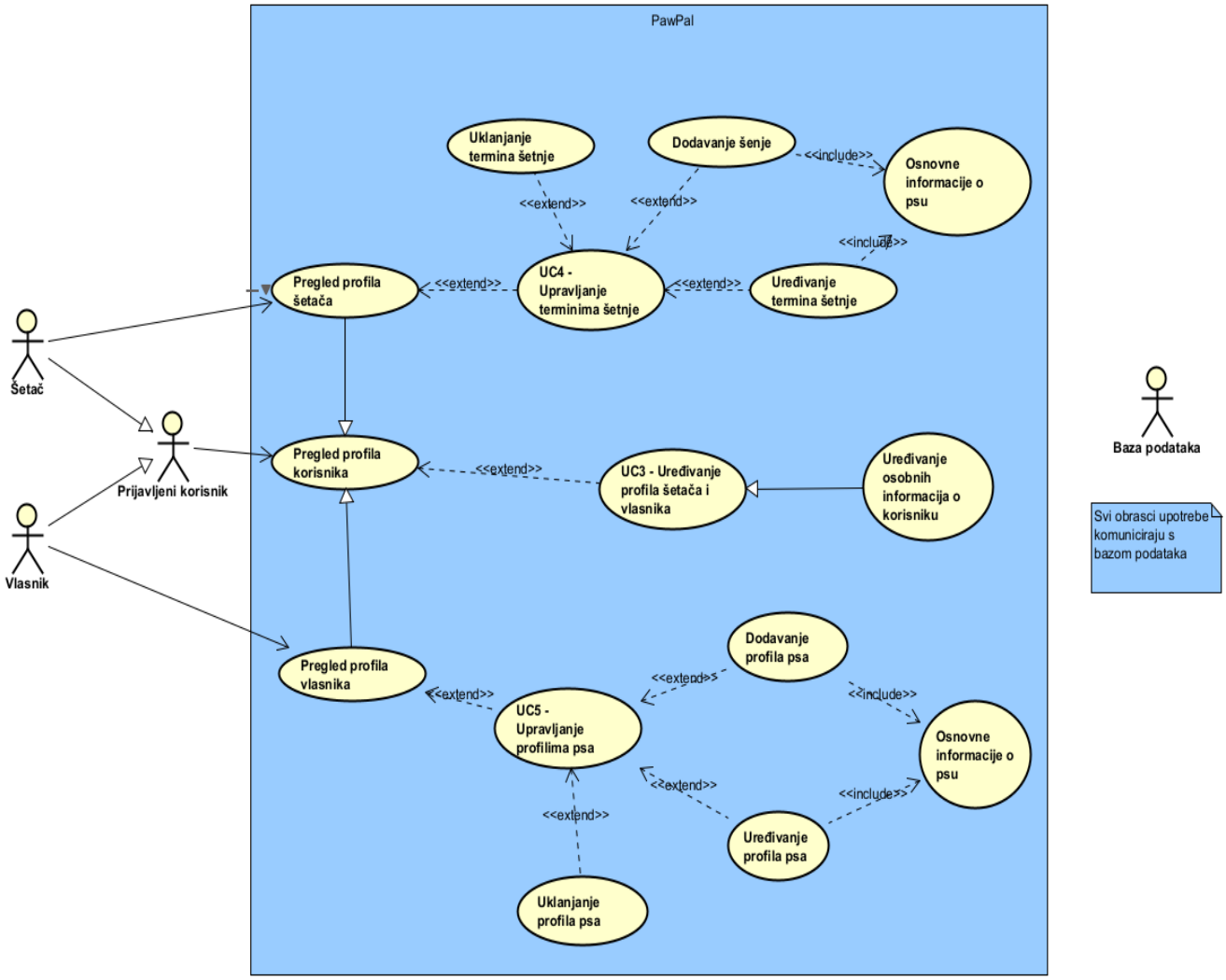
3.1. Obrasci uporabe

3.1.1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

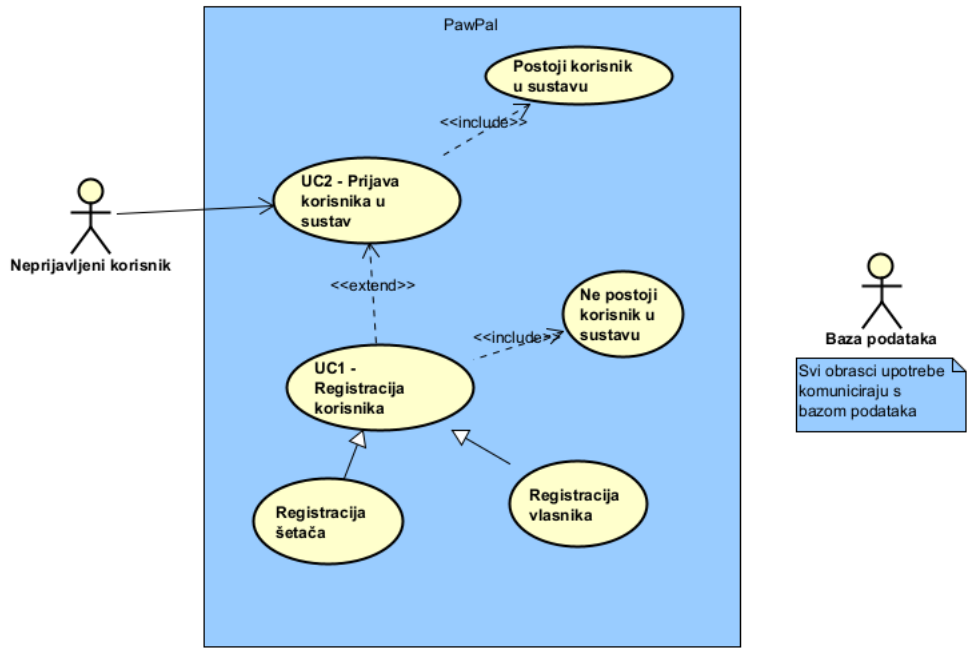


Slika 3.1. Slika dijagrama obrazaca upotrebe za uređivanje profila korisnika

3.1.2. Dijagram obrazaca uporabe za ključne značajke

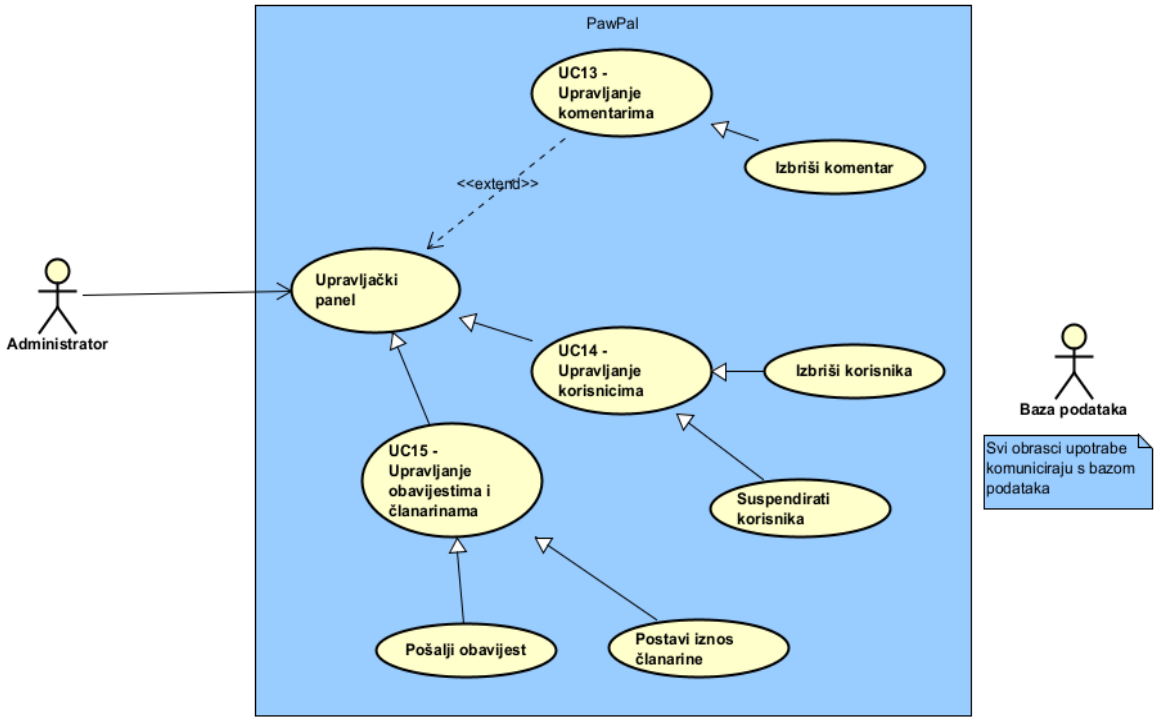


Slika 3.2. Slika dijagrama obrazaca upotrebe za uređivanje profila korisnika

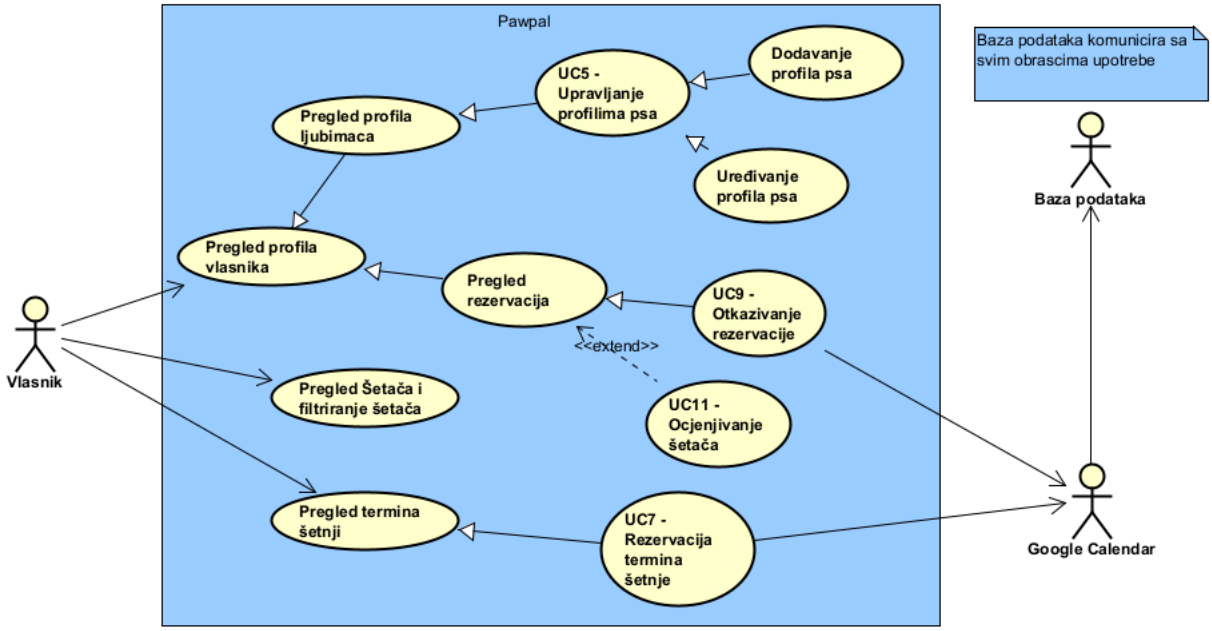


Slika 3.3. Slika dijagrama obrazaca upotrebe za prijavu korisnika

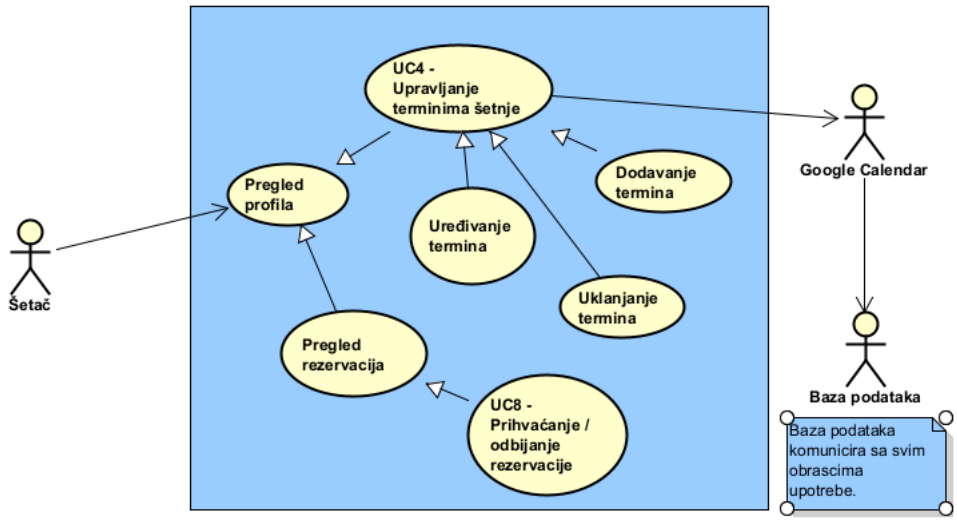
3.1.3. Dijagram obrazaca uporabe za korisničke uloge



Slika 3.4. Slika dijagrama obrazaca upotrebe za ulogu administratora

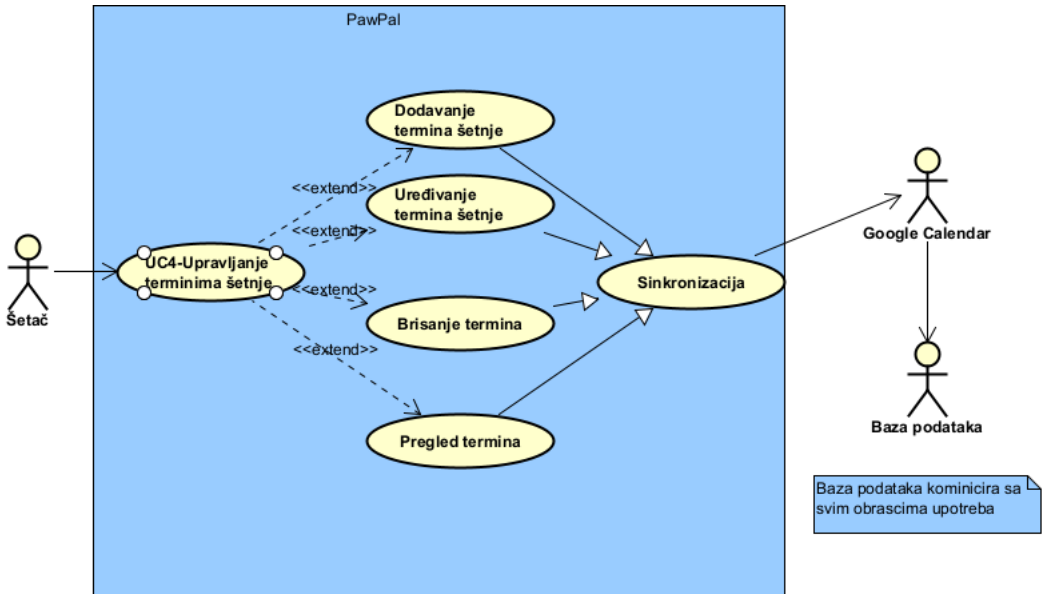


Slika 3.5. Slika dijagrama obrazaca upotrebe za ulogu vlasnika

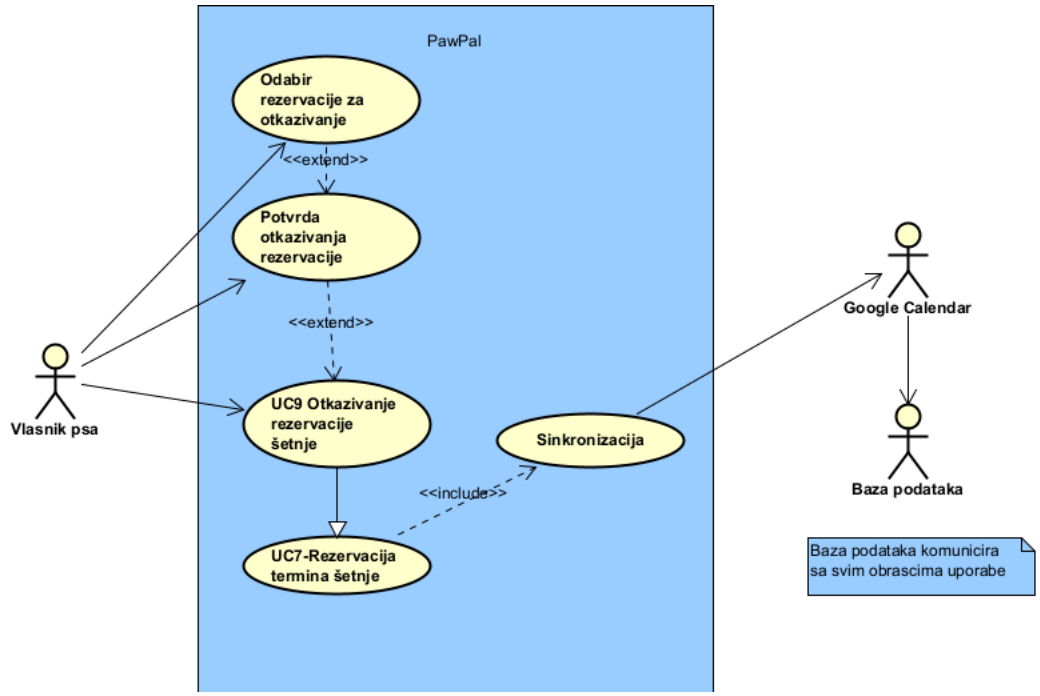


Slika 3.6. Slika dijagrama obrazaca upotrebe za ulogu šetača

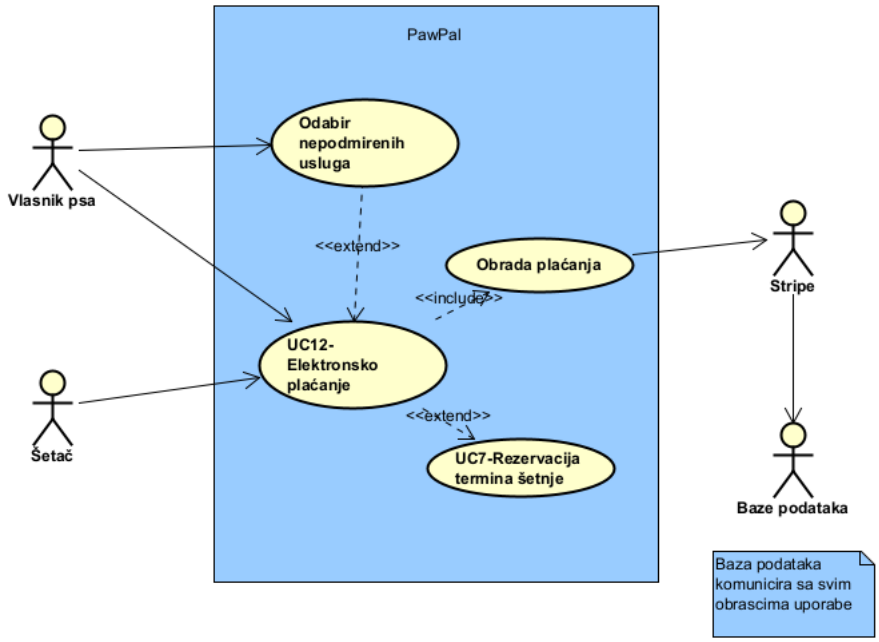
3.1.4. Dijagram obrazaca uporabe za osnovne poslovne procese



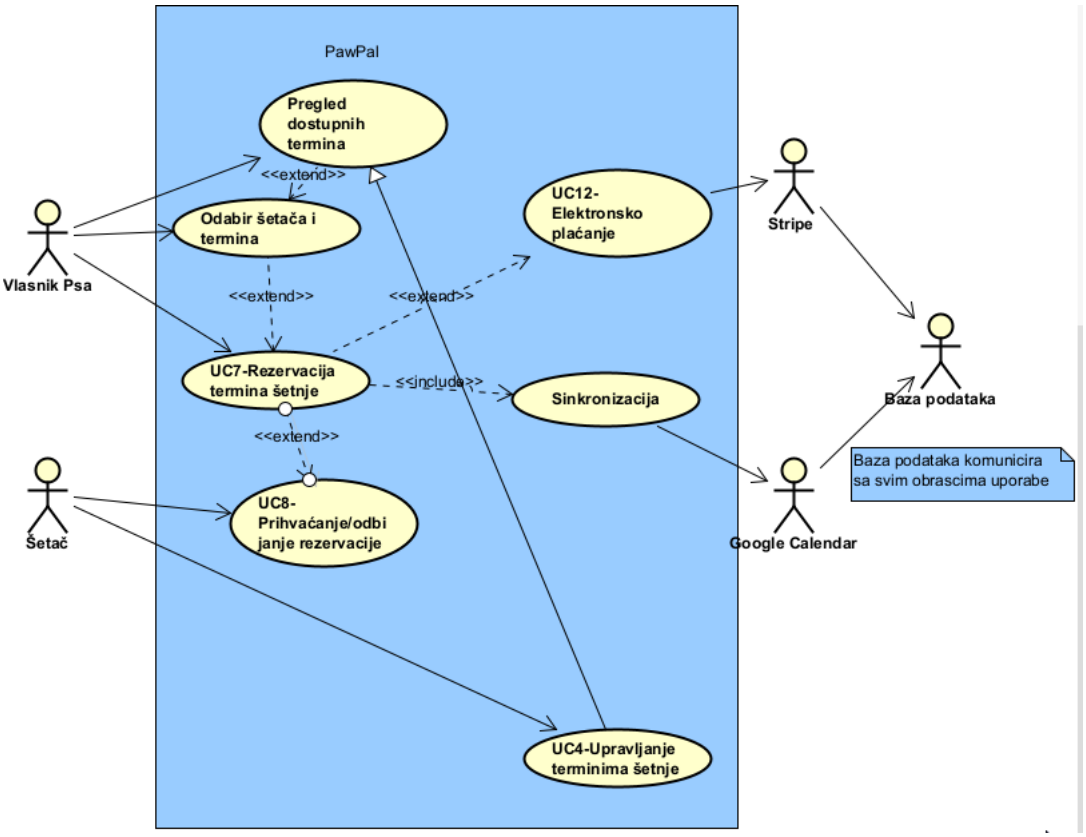
Slika 3.7. Slika dijagrama obrazaca upotrebe za upravljanje terminima



Slika 3.8. Slika dijagrama obrazaca upotrebe za otkazivanje rezervacije

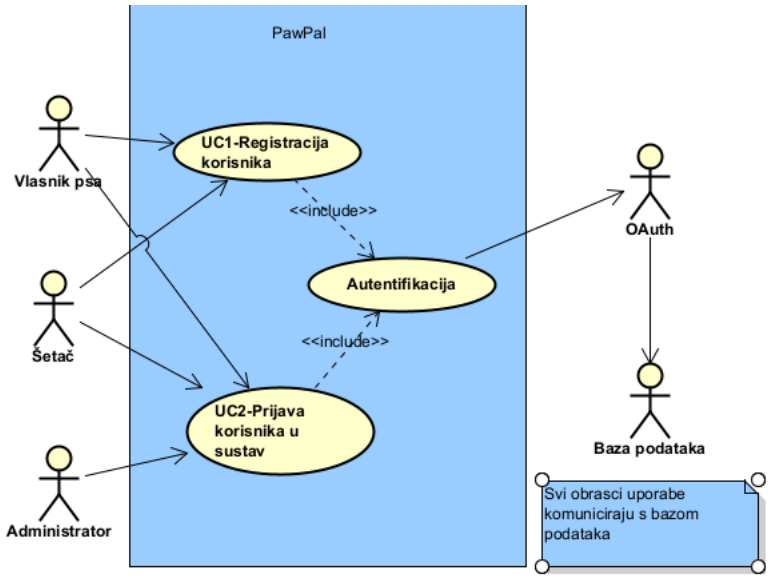


Slika 3.9. Slika dijagrama obrazaca upotrebe za plaćanje usluga u sustavu

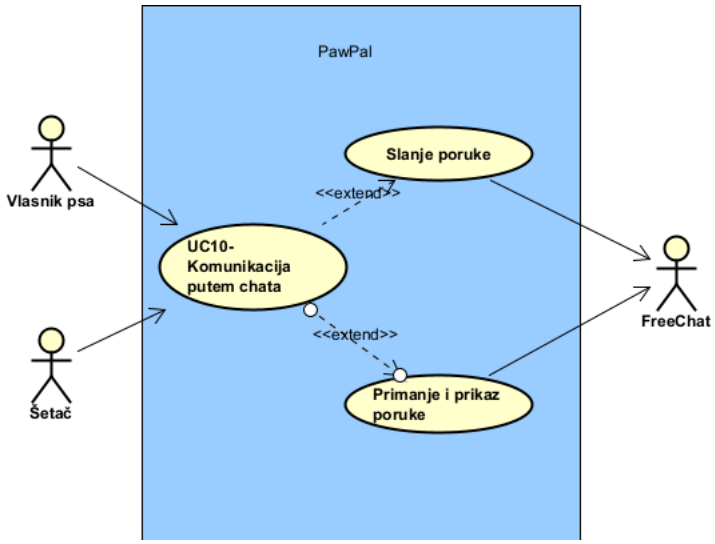


Slika 3.10. Slika dijagrama obrazaca upotrebe za rezervaciju termina

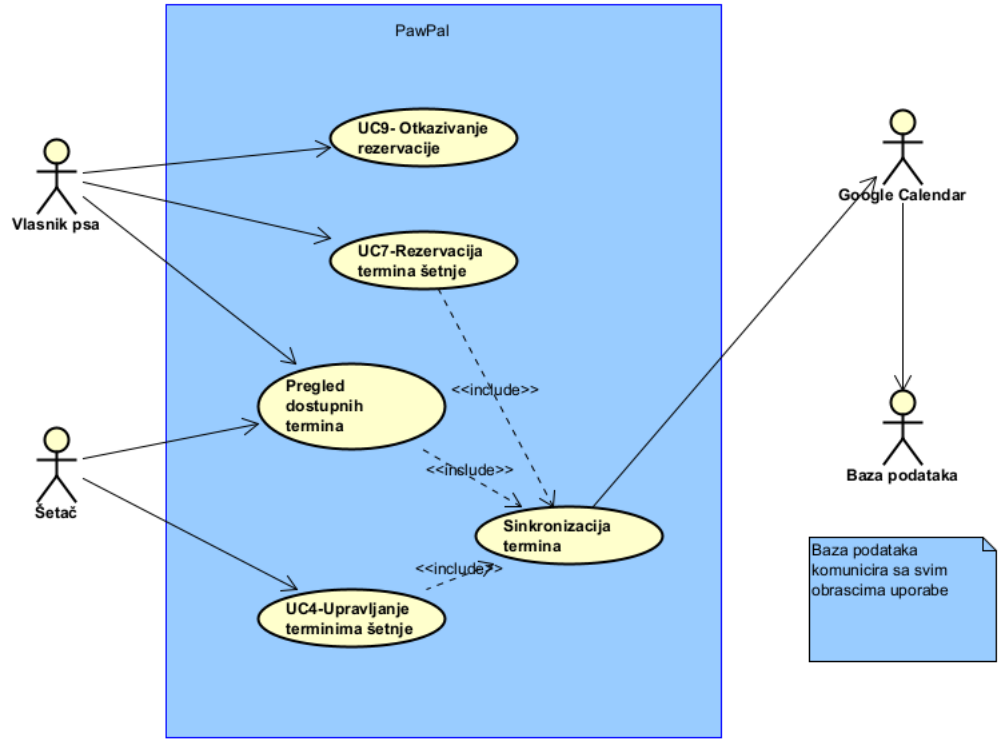
3.1.5. Dijagram obrazaca uporabe za kritične sustave i integracije



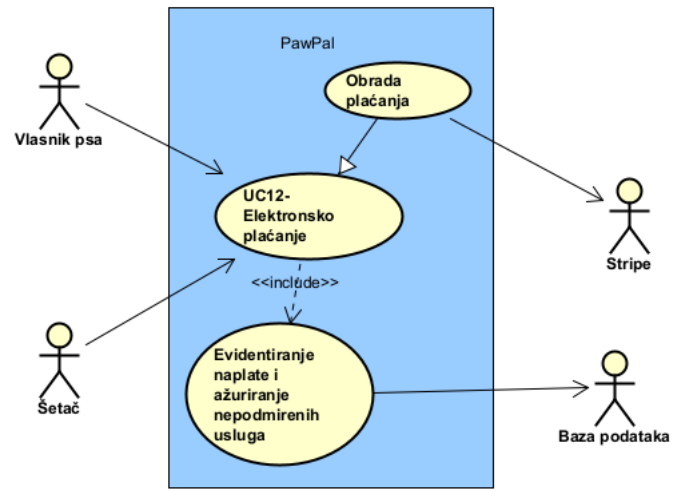
Slika 3.11. Slika dijagrama obrazaca upotrebe za OAuth servis



Slika 3.12. Slika dijagrama obrazaca upotrebe za FreeChat servis



Slika 3.13. Slika dijagrama obrazaca upotrebe za Google Calendar servis



Slika 3.14. Slika dijagrama obrazaca upotrebe za Stripe servis

3.2. Opis obrazaca uporabe

UC1 - Registracija korisnika

- Glavni sudionik: Vlasnik, Šetač
- Cilj: Korisnik(Vlasnik i Šetač) se registira u sustav kako bi mogao koristiti njegove značajke.
- Sudionici: Vlasnik, Šetač, Sustav
- Preduvjet: Korisnik nije prethodno registriran u sustav.
- Povezani funkcionalni zahtjevi: FR1
- Opis osnovnog tijeka:

1. Korisnik otvara početnu stranicu sustava.
2. Prelaskom miša preko ikone korisnika, pojavljuje se opcija prijave za korisnike. Alternativno, pritiskom na ikonu pristigle pošte ili na šetnje u izborniku, korisnika se preusmjerava na stranicu za prijavu.
3. Korisnik na stranici za prijavu odabire opciju za nove korisnike čime ga se preusmjerava na stranicu za registraciju.
4. Korisnik unosi tražene podatke i potvrđuje registraciju.
5. Sustav provjerava valjanost unesenih podataka.
6. Ako korisnik unese ispravne podatke, sustav vraća korisnika na početnu stranicu i omogućuje mu korištenje sustava.

• Opis mogućih odstupanja:

1. Ako je korisnik već registriran u sustavu, sustav obavještava korisnika te mu nudi opciju preusmjeravanja na stranicu za prijavu.
2. Ako korisnik unese neispravne podatke u sustav, sustav pokazuje korisniku greške koje je napravio prilikom unosa podataka i zahtjeva novu potvrdu registracije.

UC2 - Prijava korisnika u sustav

- Glavni sudionik: Vlasnik, Šetač, Admin
- Cilj: Korisnik(Vlasnik, Šetač i Administrator) se prijavljuje u sustav kako bi mogao koristiti njegove značajke.
- Sudionici: Vlasnik, Šetač, Admin, Sustav
- Preduvjet: Korisnik je prethodno registriran u sustav.
- Povezani funkcionalni zahtjevi: FR2
- Opis osnovnog tijeka:

1. Korisnik otvara početnu stranicu sustava.
2. Prelaskom miša preko ikone korisnika, pojavljuje se opcija prijave za korisnike. Alternativno, pritiskom na ikonu pristigle pošte ili na šetnje u izborniku, korisnika se preusmjerava na stranicu za prijavu.
3. Korisnik unosi e-mail i lozinku te potvrđuje prijavu.
4. Sustav provjerava valjanost unesenih podataka.
5. Ako korisnik unese ispravne podatke, sustav vraća korisnika na početnu stranicu i omogućuje mu korištenje sustava.
6. Ako je korisnik zaboravio lozinku, sustav mu nudi opciju promjene lozinke.

• Opis mogućih odstupanja:

1. Ako korisnik nije prethodno registriran u sustav, sustav obavještava korsnika te mu nudi opciju preusmjeravanja na stranicu za registraciju.
2. Ako korisnik unese neispravne podatke, sustav pokazuje korisniku greške koje je napravio prilikom unosa podataka i zahtjeva novu potvrdu prijave.

UC3 - Uređivanje profila šetača i vlasnika

- Glavni sudionik: Šetač, Vlasnik
- Cilj: Korisnik uređuje informacije na svom profilu.
- Sudionici: Šetač, Vlasnik, Sustav
- Preduvjet: Korisnik je prijavljen u sustav.
- Povezani funkcionalni zahtjevi: FR3
- Opis osnovnog tijeka:

1. Korisnik pritišće ikonu korisnika na početnoj stranici što ga preusmjeri na stranicu profila.
2. Na stranici profila korisnik u izborniku pritišće na "moje informacije" što ga preusmjerava na stranicu sa njegovim informacijama.
3. Na stranici "moje informacije" korisnik pritišće tipku uredi što ga preusmjerava na stranicu za uređivanje informacija.
4. Šetač uređuje informacije na svom profilu i potvrđuje izmjenu.
5. Sustav provjerava valjanost izmjena.
6. Ako je korisnik napravio ispravne izmjene, sustav vraća korisnika na stranicu "moje informacije".

• Opis mogućih odstupanja:

1. Ako korisnik unese izmjene koje nisu dopuštene, sustav pokazuje korisniku greške koje je napravio prilikom unosa podataka i zahtjeva novu potvrdu izmjena.

UC4 - Upravljanje terminima šetnje

- Glavni sudionik: Šetač

- Cilj: Šetač dodaje, uklanja i izmjenjuje termine šetnje.
- Sudionici: Šetač, Sustav
- Preduvjet: Šetač je prijavljen u sustav
- Povezani funkcionalni zahtjevi: FR4
- Opis osnovnog tijeka:

1. Šetač pritišće ikonu korisnika na početnoj stranici što ga preusmjeri na stranicu profila.
2. Na stranici profila šetač u izborniku odabire "moji termini" što ga preusmjerava na stranicu s njegovim terminima šetnje.
3. Na stranici "moji termini" šetač pritišće ikonu plusa za dodavanje termina što ga preusmjerava na stranicu za dodavanje termina.
4. Na stranici "moji termini" šetač pritišće ikonu olovke za uređivanje termina što ga preusmjerava na stranicu za uređivanje termina.
5. Na stranici "moji termini" šetač pritišće ikonu kantice za uklanjanje termina što uklanja termin.
6. Na stranici za dodavanje termina, šetač unosi podatke o šetnji i potvrđuje šetnju.
7. Na stranici za uređivanje termina, šetač izmjenjuje podatke o šetnji i potvrđuje izmjene.
8. Sustav provjerava valjanost unesenih podataka.
9. Ako je šetač unio ispravne podatke, sustav vraća šetača na stranicu "moji termini".

• Opis mogućih odstupanja:

1. Ako Šetač unese podatke koji nisu dopušteni, sustav pokazuje šetaču greške koje je napravio prilikom unosa podataka i zahtjeva novu potvrdu unosa podataka.
2. Prilikom uklanjanja termina može doći do uklanjanja krivoga termina, stoga se nakon svakog uklanjanja pojavljuje privremeni prozor za poništavanje uklanjanja.

UC5 - Upravljanje profilima psa

- Glavni sudionik: Vlasnik
- Cilj: Vlasnik dodaje, uklanja i izmjenjuje profile psa.
- Sudionici: Vlasnik, Sustav
- Preduvjet: Vlasnik je prijavljen u sustav.
- Povezani funkcionalni zahtjevi: FR5
- Opis osnovnog tijeka:

1. Vlasnik pritišće ikonu korisnika na početnoj stranici što ga preusmjeri na stranicu profila.
2. Na stranici profila vlasnik u izborniku odabire "moji psići" što ga preusmjerava na stranicu sa profilima njegovih psa.
3. Na stranici "moji psići" vlasnik pritišće ikonu plusa za dodavanje profila psa što ga preusmjerava na stranicu za dodavanje profila psa.
4. Na stranici "moji psići" vlasnik pritišće ikonu olovke za uređivanje profila psa što ga preusmjerava na stranicu za uređivanje profila psa.
5. Na stranici "moji psići" vlasnik pritišće ikonu kantice za uklanjanje profila psa što uklanja profil.
6. Na stranici za dodavanje profila psa, vlasnik unosi podatke o psu i potvrđuje izradu profila.
7. Na stranici za uređivanje profila psa, šetač izmjenjuje podatke o psu i potvrđuje izmjene.
8. Sustav provjerava valjanost unesenih podataka.
9. Ako je vlasnik unio ispravne podatke, sustav vraća vlasnika na stranicu "moji psići".

• Opis mogućih odstupanja:

1. Ako vlasnik unese podatke koji nisu dopušteni, sustav pokazuje vlasniku greške koje je napravio prilikom unosa podataka i zahtjeva novu potvrdu unosa podataka.
2. Prilikom uklanjanja profila psa može doći do uklanjanja krivoga profila, stoga se nakon svakog uklanjanja pojavljuje prozor za poništavanje uklanjanja.

UC6 - Pretraživanje šetača

- Glavni sudionik: Vlasnik
- Cilj: Pronalazak odgovarajućeg šetača.
- Sudionici: Vlasnik, Sustav
- Preduvjet: Vlasnik je prijavljen u sustav.
- Povezani funkcionalni zahtjevi: FR6, FR4
- Opis osnovnog tijeka:

1. Vlasnik pritišće "Naši šetači" u izborniku na početnoj stranici što ga preusmjeri na stranicu s registriranim šetačima.
2. Vlasnik filtrira šetače po odgovarajućim kriterijima te dobiva listu prigodnih šetača.
3. Vlasnik se pritiskom na odgovarajućeg šetača preusmjerava na stranicu sa detaljima o šetaču.

• Opis mogućih odstupanja:

1. Ako vlasnik odabere kriterije koje ne može zadovoljiti niti jedan šetač, sustav predlaže vlasniku da prilagodi kriterije.

UC7 - Rezervacija termina šetnje

- Glavni sudionik: Vlasnik
- Cilj: Rezervacija termina šetnje pomoću Google Calendar API.
- Sudionici: Vlasnik, Šetač i Sustav
- Preduvjet: Vlasnik je prijavljen i postoji minimalno jedan profil psa.
- Povezani funkcionalni zahtjevi: FR7
- Opis osnovnog tijeka:

1. Vlasnik pritišće "Pogledaj šetnje" ili "Dostupne šetnje" u izborniku na početnoj stranici što ga preusmjeri na stranicu s terminima šetnji.
2. Na stranici s terminima šetnji, vlasnik odabire slobodan termin iz Google Calendar integracije.
3. Vlasnik unosi datum, vrijeme, trajanje, tip šetnje, polazište i napomene te bira način plaćanja(gotovina ili elektronsko plaćanje) te potvrđuje rezervaciju.
4. Sustav provjerava valjanost rezervacije.
5. Sustav stvara zahtjev i šalje rezervaciju šetaču.

• Opis mogućih odstupanja:

1. Ako Vlasnik unese neispravne informacije, sustav pokazuje vlasniku greške koje je napravio prilikom unosa podataka i traži novu potvrdu rezervacije.

UC8 - Prihvaćanje / odbijanje rezervacije

- Glavni sudionik: Šetač
- Cilj: Prihvaćanje ili odbijanje rezervacije vlasnika.
- Sudionici: Vlasnik, Šetač i Sustav
- Preduvjet: Postoji rezervacija vlasnika.
- Povezani funkcionalni zahtjevi: FR7
- Opis osnovnog tijeka:

1. Šetač pritišće ikonu korisnika na početnoj stranici što ga preusmjeri na stranicu profila.
2. Na stranici profila šetač u izborniku odabire "Moje rezervacije" što ga preusmjerava na stranicu s rezervacijama.
3. Na stranici s rezervacijama prikazuje se lista pristiglih rezervacija šetnji.
4. Pritiskom na ikonu kvačice, odnosno ikone slova x, šetač prihvaća, odnosno odbija šetnju.

• Opis mogućih odstupanja:

1. Prilikom odbijanja rezervacije može doći do odbijanja krive rezervacije, stoga se nakon svakog odbijanja pojavljuje privremeni prozor za poništavanje odbijanja.

UC9 - Otkazivanje rezervacije

- Glavni sudionik: Vlasnik
- Cilj: Otkazivanje rezervacije.
- Sudionici: Vlasnik, Sustav
- Preduvjet: Vlasnik ima rezervacije i do početka šetnje ima više od 24 sata.
- Povezani funkcionalni zahtjevi: FR8
- Opis osnovnog tijeka:

1. Vlasnik pritišće ikonu korisnika na početnoj stranici što ga preusmjeri na stranicu profila.
2. Na stranici profila vlasnik u izborniku odabire "Moje rezervacije" što ga preusmjerava na stranicu s njegovim rezervacijama.
3. Na stranici sa rezervacijama, vlasniku se uz šetnje prikazuje ikona slova x ako do početka šetnje ima više od 24 sata.
4. Pritiskom na ikonu slova X, otkazuje se rezervacija.

• Opis mogućih odstupanja:

1. Prilikom otkazivanja rezervacije može doći do otkazivanja krive rezervacije, stoga se nakon svakog otkazivanja pojavljuje privremeni prozor za poništavanje otkazivanja.

UC10 - Komunikacija putem chata

- Glavni sudionik: Vlasnik, Šetač
- Cilj: Komunikacija vlasnika i šetača putem chata ostvarenog uz pomoć servisa FreeChat.
- Sudionici: Vlasnik, Šetač i Sustav
- Preduvjet: Postoji prihvaćena ponuda, odnosno rezervacija.
- Povezani funkcionalni zahtjevi: FR9
- Opis osnovnog tijeka:

1. Korisnik pritišće ikonu poruke na početnoj stranici što ga preusmjeri na stranicu sa razgovorima.
2. Na stranici sa razgovorima, korisnik upisuje poruku u prozor za poruke te pritiskom na gumb za slanje poruke, šalje poruku drugom korisniku.
3. Sustav šalje poruku željenom korisniku.

• Opis mogućih odstupanja:

1. Ako korisnik pošalje poruku neprimjerenog sadržaja, administrator je može izbrisati.

UC11 - Ocjenjivanje šetača

- Glavni sudionik: Vlasnik
- Cilj: Vlasnik ocjenjuje šetača.
- Sudionici: Vlasnik, Šetač i Sustav
- Preduvjet: Postoji završena rezervacija, odnosno šetnja.
- Povezani funkcionalni zahtjevi: FR10, FR11
- Opis osnovnog tijeka:

1. Vlasnik pod "moje rezervacije" pronalazi završene šetnje te pritišće gumb za ocjenjivanje šetača što ga prusmjerava na stranicu za ocjenjivanje.
2. Na stranici za ocjenjivanje, vlasnik odabire ocjenu(od 1 do 5), može ostaviti sliku i komentar te potvrđuje ocjenjivanje.
3. Sustav sprema ocjenu i ažurira prosjek šetača.

• Opis mogućih odstupanja:

1. Ako korisnik pošalje komentar ili sliku neprimjerenog sadržaja, administrator je može izbrisati.

UC12 - Elektronsko plaćanje

- Glavni sudionik: Vlasnik i Šetač
- Cilj: Korisnik plaća šetnje i članarinu putem elektronskog plaćanja(PayPal/kartično plaćanje) uz pomoć servisa Stripe.
- Sudionici: Vlasnik, Šetač i Sustav
- Preduvjet: U sustavu postoji članarina ili šetnja koju korisnik treba platiti.
- Povezani funkcionalni zahtjevi: FR12, FR13
- Opis osnovnog tijeka:

1. Korisnik na početnoj stranici pritišće ikonu korisnika koja ga preusmjerava na stranicu profila.
2. Na stranici profila, korisnik u izborniku pritišće "Plaćanje" što ga preusmjerava na stranicu s nepodmirenim uslugama.
3. Na stranici s nepodmirenim uslugama, korisnik pritišće na tipku za plaćanje što ga preusmjerava na platni servis.
4. Sustav bilježi plaćanje i uklanja nepodmirenu uslugu.

• Opis mogućih odstupanja:

1. Ako korisnik ne plati iskorištene usluge u roku od 14 dana od datuma naplaćivanja, sustav obavještava administratora koji poduzima daljnje korake.

UC13 - Upravljanje komentarima

- Glavni sudionik: Administrator
- Cilj: Administrator pregledava i uklanja komentare.
- Sudionici: Administrator, Sustav
- Preduvjet: Postoji komentar korisnika u sustavu.
- Povezani funkcionalni zahtjevi: FR10
- Opis osnovnog tijeka:

1. Administrator na početnoj stranici pritišće ikonu alata koja ga preusmjerava na stranicu za upravljanje sustavom.
2. Na stranici za upravljanje, administrator u izborniku pritišće "komentari" što ga preusmjerava na stranicu s komentarima.
3. Na stranici s komentarima, administrator uklanja komentare pritiskom na ikonu koša za smeće.
4. Sustav uklanja komentar.

• Opis mogućih odstupanja:

1. Pri pokušaju uklanjanja komentara, sustav traži potvrdu uklanjanja kako bi se sprječilo neželjeno uklanjanje komentara.

UC14 - Upravljanje korisnicima

- Glavni sudionik: Administrator
- Cilj: Administrator suspendira korisnike.
- Sudionici: Administrator, Sustav
- Preduvjet: Postoji korisnik u sustavu.
- Povezani funkcionalni zahtjevi: FR14
- Opis osnovnog tijeka:

1. Administrator na početnoj stranici pritišće ikonu alata koja ga preusmjerava na stranicu za upravljanje sustavom.
2. Na stranici za upravljanje, administrator u izborniku pritišće "korisnici" što ga preusmjerava na stranicu s korisnicima.
3. Na stranici s korisnicima, administrator upravlja (brisanje, suspendiranje) korisnicima.

• Opis mogućih odstupanja:

1. Pri pokušaju suspendiranja korisnika, sustav traži potvrdu suspendiranja kako bi se sprječilo neželjeno suspendiranje.

UC15 - Upravljanje obavijestima i članarinama

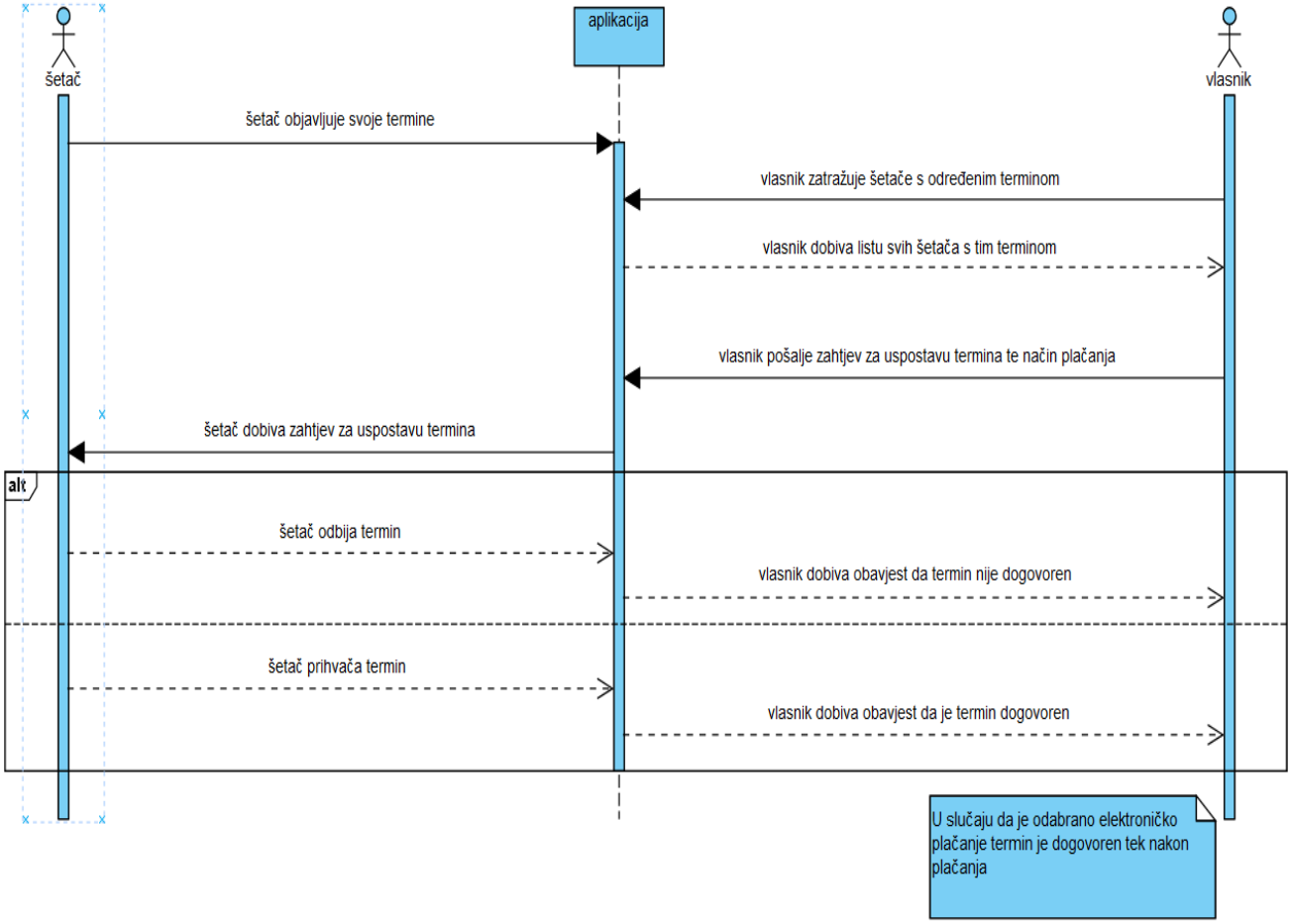
- Glavni sudionik: Administrator
- Cilj: Administrator izdaje obavijesti i upravlja članarinama.
- Sudionici: Administrator, Sustav
- Preduvjet: Administrator je prijavljen u sustav.
- Povezani funkcionalni zahtjevi: FR12, FR14
- Opis osnovnog tijeka:

1. Administrator na početnoj stranici pritišće ikonu alata koja ga preusmjerava na stranicu za upravljanje sustavom.
2. Na stranici za upravljanje, administrator u izborniku pritišće "generalno" što ga preusmjerava na stranicu s općim postavkama.
3. Na stranici s općim postavkama, administrator izdaje obavijesti i korigira članarinu.

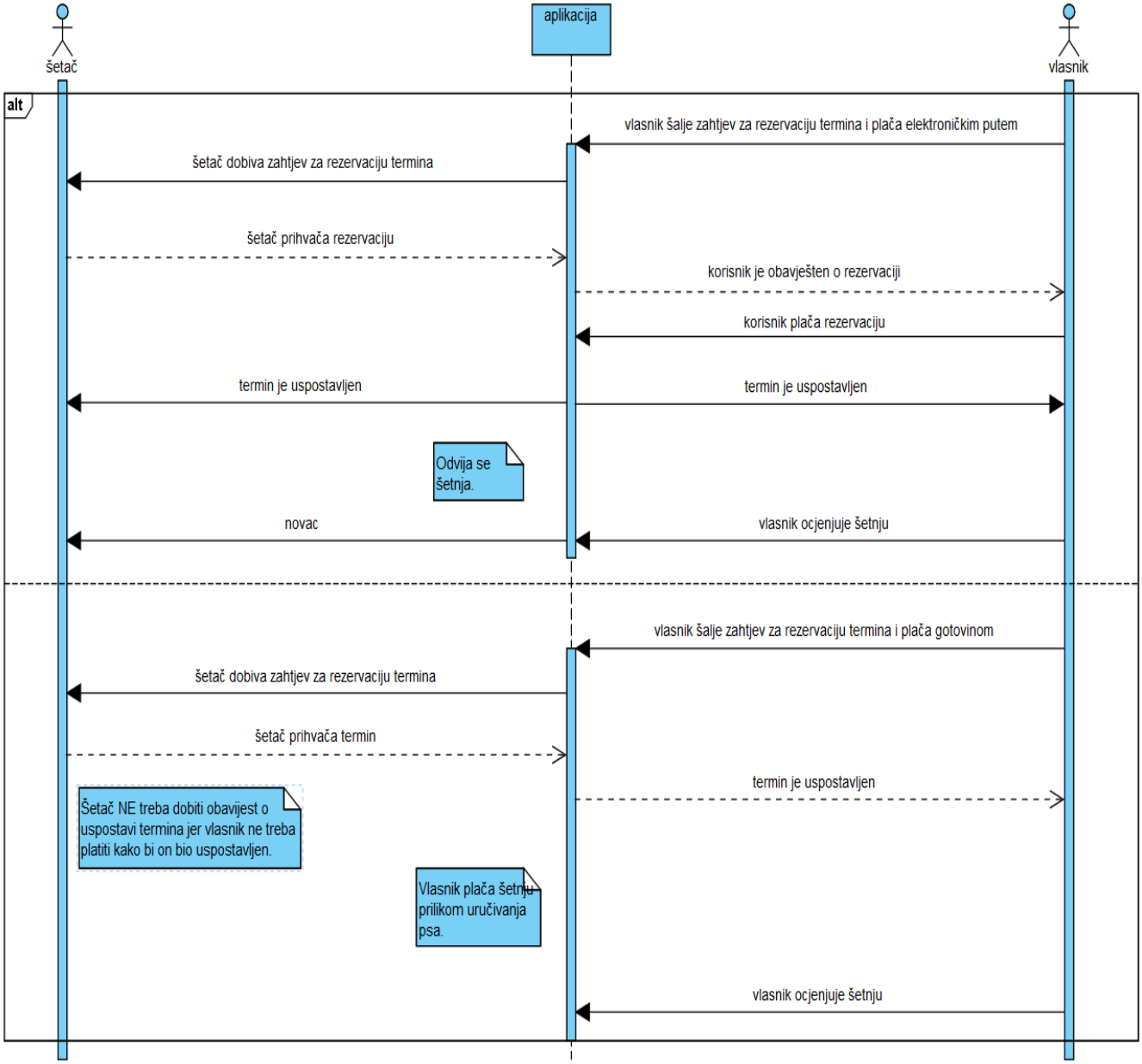
• Opis mogućih odstupanja:

1. Pri izdavanju obavijesti, sustav traži potvrdu kako bi se sprječilo neželjeno obaviještavanje.
2. Pri promjeni članarine, sustav traži potvrdu kako bi se sprječilo neželjeno mijenjanje članarine.

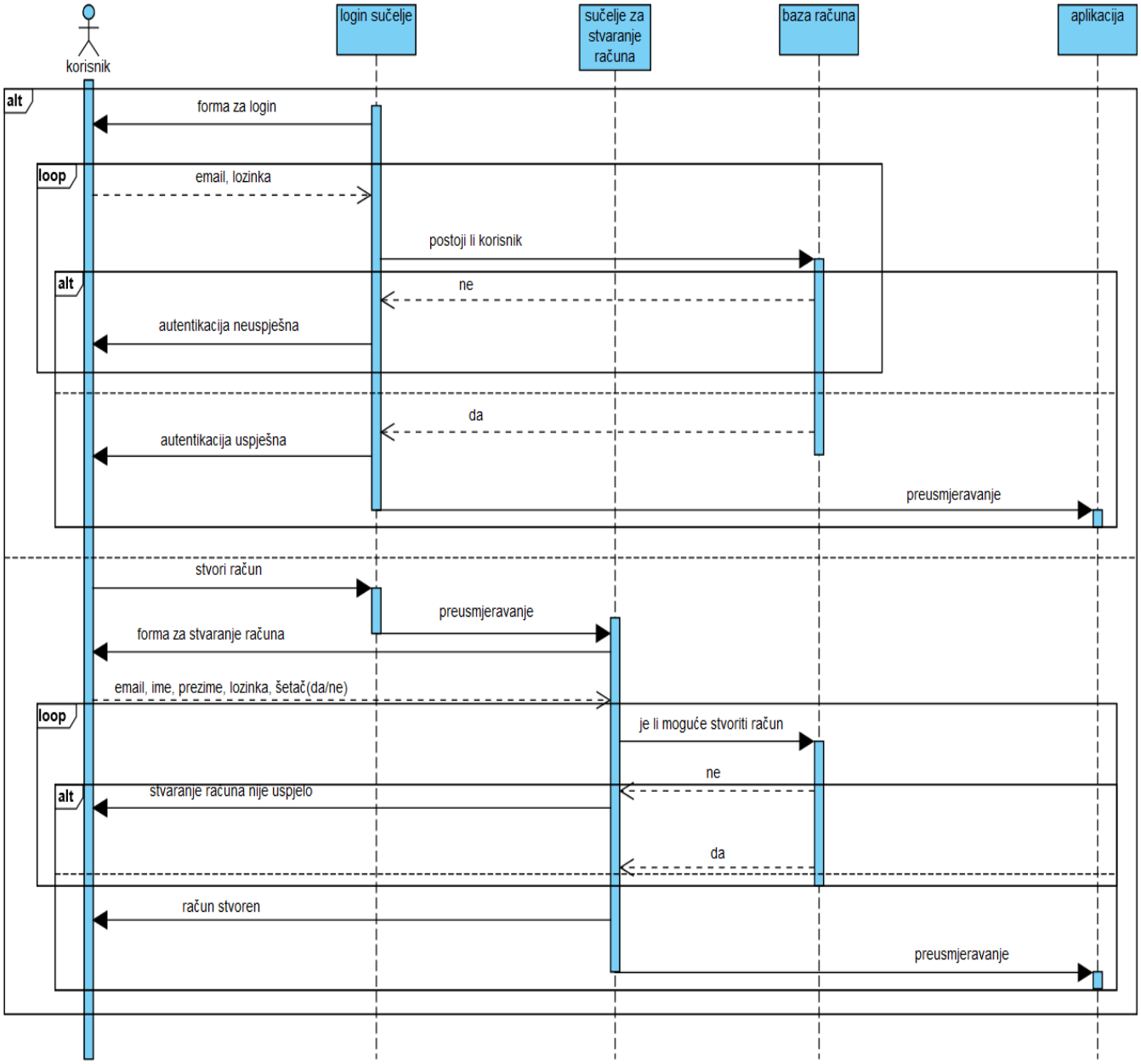
3.3. Sekvencijski dijagrami



Slika 3.15. Sekvencijski dijagram usluge plaćanja i rezervacije termina



Slika 3.16. Sekvencijski dijagram usluge plaćanja



Slika 3.17. Sekvencijski dijagram prijave korisnika

3.4. Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Funkcionalni zahtjev	Obrazac uporabe
FR1	UC1
FR2	UC2
FR3	UC3
FR4	UC4, UC8, UC3
FR5	UC5, UC3
FR6	UC3
FR7	UC6
FR8	UC7
FR9	UC9
FR10	UC8
FR11	UC10
FR12	UC11

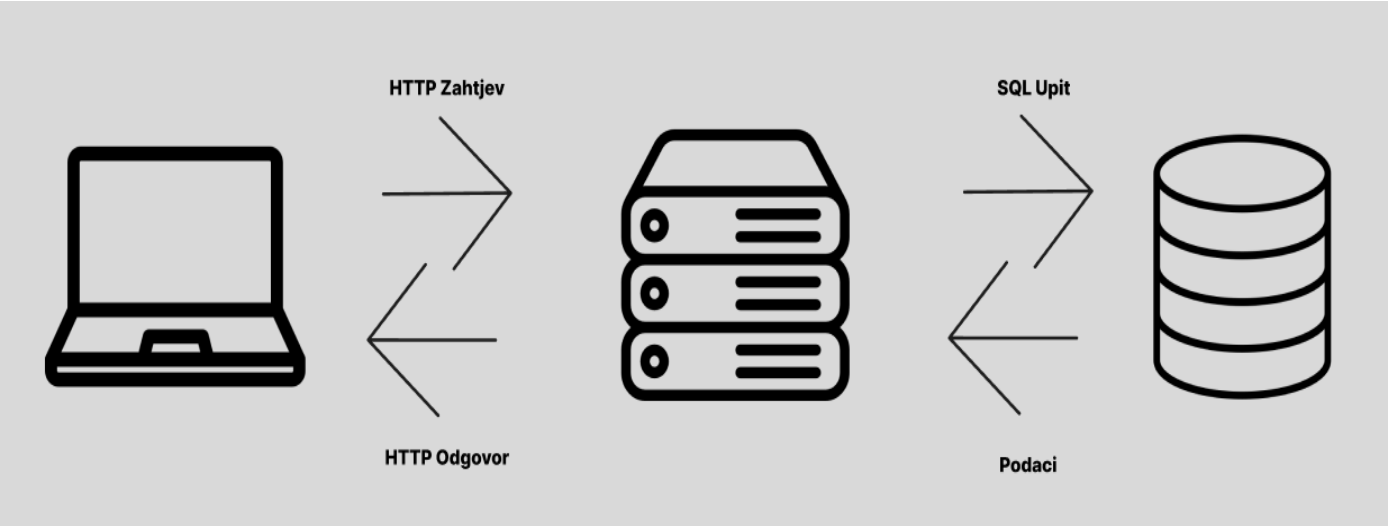
Funkcionalni zahtjev	Obrazac uporabe
FR13	UC6
FR14	UC12
FR15	UC12
FR16	UC15, UC13, UC14

\newpage

4. Arhitektura i dizajn sustava

4.1. Opis arhitekture

PawPal koristi klijent–poslužitelj arhitektonski stil s odvojenim frontendom i backendom koji komuniciraju preko REST API-ja. Frontend je React aplikacija (SPA) koja pruža korisničko sučelje i upravlja klijentskim stanjem, dok backend (Python Django + Django REST Framework) izlaže API krajnje točke, provodi poslovnu logiku te pristupa PostgreSQL bazi podataka. Autentikacija je putem OAuth 2.0, a autorizacija je vođena ulogama (Vlasnik/Šetač/Admin). Komunikacija se odvija preko HTTPS.



Slika 4.1. Dijagram sustava

4.1.2. Podsustavi

Korisnički podsustav čini frontend u Reactu, implementiran kao SPA s rutiranjem (React Router). On prikazuje profile šetača, omogućuje filtriranje po lokaciji, cijeni i ocjeni, vodi korisnika kroz forme za rezervacije i recenzije te provodi osnovnu klijentsku validaciju. Podsustav za logiku i podatke temelji se na Django/Django REST Frameworku i zadužen je za autentikaciju putem OAuth-a 2.0 te autorizaciju prema ulogama (Owner/Walker/Admin). Ovaj podsustav implementira poslovna pravila, izlaže REST krajnje točke te posreduju između frontenda i sloja pohrane podataka. Podsustav za pohranu podataka koristi PostgreSQL i organiziran je kao relacijski model s primarnim i stranim ključevima te odgovarajućim UNIQUE i CHECK ograničenjima. Nad često korištenim atributima postavljeni su indeksi kako bi se ubrzale pretrage i filtriranja. Za pregled i rezervacije termina koristi se integracija s Google Calendarom, koja omogućuje sinkronizaciju slobodnih i zauzetih slotova te lakše upravljanje rasporedom šetača. Podsustav za plaćanja oslanja se na vanjski payment gateway, prvenstveno Stripe/PayPal, preko kojeg se obrađuju transakcije i zaprima povratni status. Za notifikacije i chat koriste se e-mail obavijesti te integrirani vanjski chat-widget (FreeChat) kako bi komunikacija između vlasnika i šetača bila brza i centralizirana

4.1.3. Spremišta podataka

Primarno PostgreSQL zbog referencijalnog integriteta i potreba za transakcijama u rezervacijama (sprječavanje dvostrukog bookinga). Kreirani su indeksi po poljima pretrage(lokacija, cijena, dostupnost, ocjena). Slikama upravlja se izvan baze (objektna pohrana), dok se u bazi drže putanje i atributi.

4.1.4. Mrežni protokoli

Primarni način komunikacije između frontenda i backenda odvija se putem HTTPS-a uz REST-sučelje. Frontend(React) šalje zahtjeve prema backendu (Django/DRF), koji ih obrađuje i vraća odgovore u JSON formatu. Za autentikaciju se koriste OAuth 2.0 tokovi. Sustav prima webhook poruke iz platnog sustava.

4.1.5. Sklopovskoprogramski zahtjevi

Poslužiteljska strana nema visoke hardverske zahtjeve za očekivani studentski opseg: preporučuje se instanca s 2 vCPU i 2–4 GB RAM-a Korisnička strana je optimizirana za moderne preglednike te ne ovisi o korištenom operacijskom sustavu. Prikaz je prilagođen za različite vrste zaslona. Preporučuje se brza i stabilna internetska veza.

4.2. Obrazloženje odabira arhitekture

Odabrali smo klijent–poslužitelj s odvojenim React frontendom i Django/DRF backendom zbog jasne podjele odgovornosti, paralelnog rada podtimova i malog opsega projekta. React nudi bogat ekosustav i mnogo resursa, što ubrzava izradu UI-ja, dok Django/DRF donosi siguran i dosljedan REST API i odličnu integraciju s PostgreSQL-om. Tehnološki izbor usklađen je s vještinama tima pa je krivulja učenja niska i rizik manji. Arhitektura zadovoljava zahtjeve skalabilnosti i održivosti jer frontend i backend možemo neovisno razvijati i deployati, a domene su modularno razdvojene. Vodili smo se principima visoke kohezije, niske povezanosti i sigurnosti po slojevima Razmatrali smo SSR monolit (jednostavniji deploy, ali slabije SPA iskustvo) i mikrousluge (prevelik operativni overhead za naš opseg), kao i Spring/ASP.NET, no preferirali smo Django zbog kompetencija tima i bržeg razvoja.

4.3. Organizacija sustava na visokoj razini

PawPal koristi klijent–poslužitelj arhitekturu s odvojenim React frontendom i Django/DRF backendom. Korisnik kroz web sučelje šalje zahtjeve prema REST APIju, gdje backend obrađuje poslovnu logiku, pristupa podacima i vraća JSON odgovore koje frontend prikazuje u prikladnom obliku. Baza podataka je relacijska i služi kao centralno spremište za korisnike, ljubimce, šetače, termine, rezervacije, plaćanja i recenzije, uz transakcije i referencijalni integritet. Sučelje je responzivno i povezano s ostalim komponentama preko HTTPS/REST poziva, uz podršku za autentikaciju i notifikacije/webhookove gdje je potrebno.

4.4. Organizacija aplikacije

Aplikacija PawPal organizirana je kao dvoslojni web sustav: frontend (React SPA) i backend (Django + Django REST Framework) koji komuniciraju putem REST API-ja.

Frontend i Backend slojevi

Frontend (React + Vite) Frontend je implementiran u direktoriju frontend/ kao single-page aplikacija. Glavne rute su definirane u App.jsx (/, /login, /register), a korisničko sučelje je podijeljeno na stranice (src/pages) i komponente (src/components, npr. Navbar, GoogleLoginButton). Frontend ne pristupa bazi podataka direktno, nego sve podatke dobiva preko API poziva definiranih u src/api/auth.js i src/api/client.js (npr. /api/auth/csrf/, /api/auth/login/, /api/auth/logout/, /api/auth/me/). Autentifikacija se temelji na session cookie + CSRF tokenu, a za Google prijavu koristi se poseban endpoint za OAuth login.

Backend (Django + DRF) Backend se nalazi u direktoriju bekend/ i organiziran je kao Django projekt pawpal_backend s aplikacijom accounts. U pawpal_backend/urls.py backend izlaže:

/api/ – REST API, uključujući auth rute iz accounts/urls.py

/api/schema/ i /api/docs/ – automatska dokumentacija API-ja preko drf-spectacular

/accounts/ – Django Allauth rute za socijalnu prijavu (Google)

Aplikacija accounts sadrži modele, serijalizere i view-e za registraciju, prijavu, odjavu, dohvat trenutnog korisnika i upravljanje postavkama obavijesti. Backend centralizira poslovnu logiku (pravila registracije, autentifikacije, rad s profilima i notifikacijama) i komunikaciju s bazom podataka.

MVC / slojevita arhitektura

Backend slijedi tipičan Django MVC obrazac:

Model – Django modeli (npr. korisnici i profili) definiraju strukturu podataka i veze.

View – DRF view funkcije/CBV obrađuju HTTP zahtjeve, validiraju ulaz preko serijalizera i vraćaju JSON odgovore.

“Controller” – implicitno je sadržan u Django URL routeru i view sloju (urls.py usmjerava zahtjeve na odgovarajuće view-e).

Frontend preuzima ulogu View-a na klijentskoj strani (renderiranje sučelja, navigacija, local state), dok backend implementira poslovnu logiku i pristup podacima, što jasno odvaja odgovornosti i olakšava daljnje proširenje (dodavanje modula za termine, plaćanje, recenzije i chat kao novih Django aplikacija i React stranica).

Dijagram visoke razine (opis za sliku)

Na dijagramu visoke razine (koji može biti inicijalno skiciran) prikazuju se sljedeći podsustavi:

Korisnik (browser) – komunicira s frontend aplikacijom.

Frontend (React SPA) – šalje HTTP(S) zahtjeve na backend REST API (/api/...).

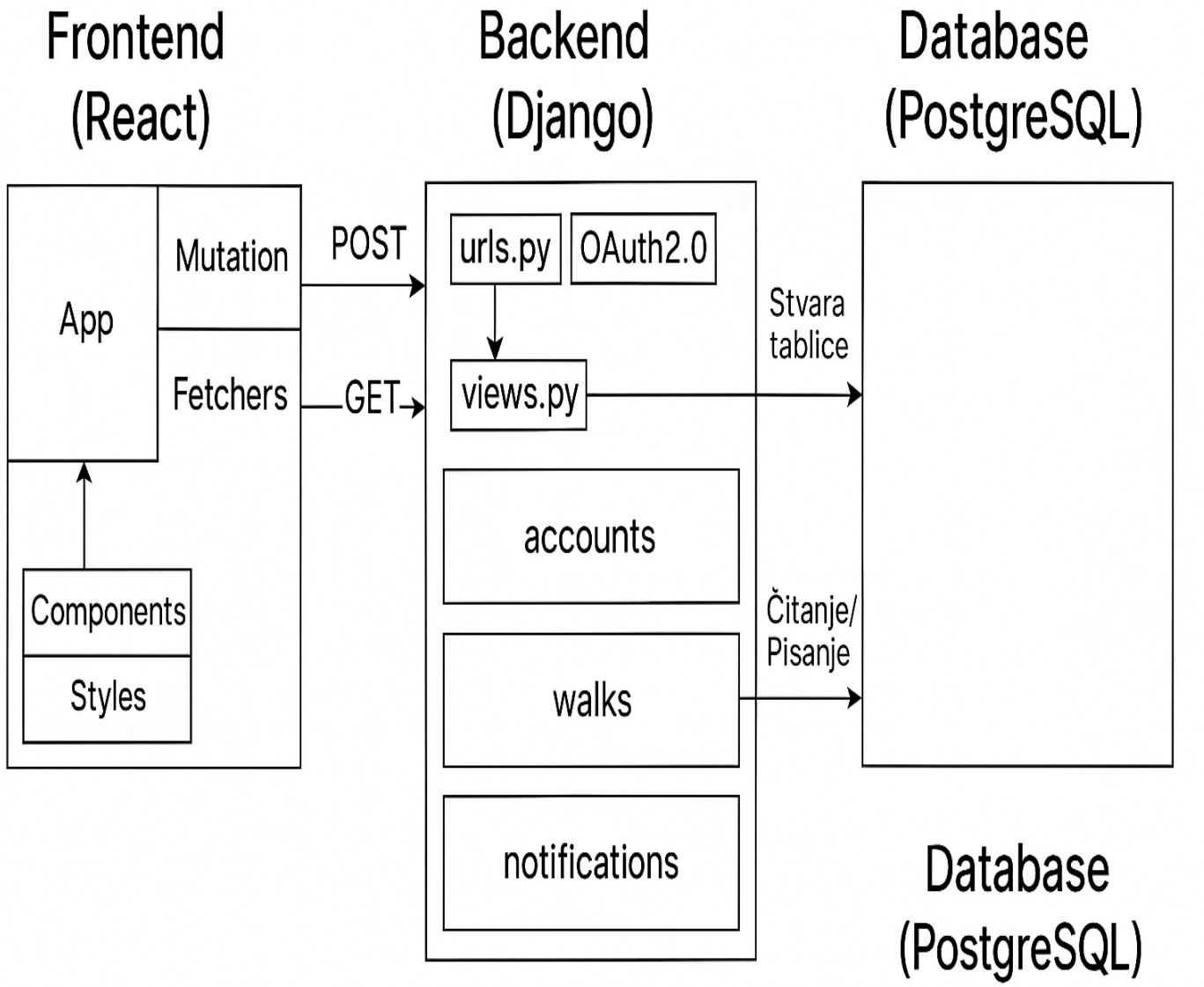
Backend (Django + DRF + Allauth) – obrađuje zahtjeve, primjenjuje poslovnu logiku, upravlja autentifikacijom (uključujući Google OAuth) i poziva vanjske servise.

Baza podataka – pohranjuje korisnike, profile, termine, recenzije i ostale domenske podatke.

Vanjski servisi (planirano) – OAuth provider (Google), servis za kalendar (Google Calendar), platni servisi (PayPal / kartice) i chat servis (npr. FreeChat).

U drugom ciklusu ovaj dijagram će se detaljnije razraditi u dijagram komponenti (razdvajanje pojedinih Django aplikacija i React modula).

Referenca Ovaj pristup je u skladu s principima mikrouslužne arhitekture kako ih definiraju Fowler i Lewis (2014) te s industrijskim praksama navedenim u "Building Microservices" (Newman, 2015), čiji su primjeri korišteni za detaljnije definiranje uloga i komunikacije među mikrouslugama.



Slika 4.2. Dijagram organizacije aplikacije

4.5. Baza podataka

Sustav koristi relacijsku bazu podataka – PostgreSQL. Odabrana je zbog transakcijske pouzdanosti, referencijalnog integriteta, podrške za složene upite i jednostavnog indeksiranja po poljima pretrage (lokacija/cijena/termin/ocjena). Podatci su normalizirani, a konzistentnost se osigurava PK/FK/UNIQUE/CHECK ograničenjima.

4.5.1. Opis tablica

Administrator

Primary Key: `idAdmin`
Unique: `emailAdmin`, `urnAdmin`

Atribut	Tip podatka	Opis varijable
`idAdmin` (PK)	BIGINT	Jedinstveni identifikator administratora.
`emailAdmin` (U)	VARCHAR(50)	E-mail administratora (jedinstven).
`urnAdmin` (U)	VARCHAR(50)	Korisničko ime (jedinstveno).

Atribut	Tip podatka	Opis varijable
passAdmin	VARCHAR(50)	Hash lozinke.

VlasnikPsa

Primary Key: idVlasnika
Unique: emailVlasnika, urnVlasnika, telefonVlasnika

Atribut	Tip podatka	Opis varijable
idVlasnika (PK)	BIGINT	Jedinstveni identifikator vlasnika.
imeVlasnika	VARCHAR(20)	Ime vlasnika.
prezimeVlasnika	VARCHAR(20)	Prezime vlasnika.
emailVlasnika (U)	VARCHAR(50)	E-mail (jedinstven).
telefonVlasnika (U)	VARCHAR(15)	Kontakt telefon (jedinstven).
urnVlasnika (U)	VARCHAR(20)	Korisničko ime (jedinstveno).
passVlasnika	VARCHAR(20)	Hash lozinke.
idPretplate	BIGINT	Jedinstveni identifikator pretplate.

Pretplata

Primary Key: idPretplate
Foreign Key: idVlasnika → VlasnikPsa.idVlasnika

Atribut	Tip podatka	Opis varijable
idPretplate (PK)	BIGINT	Jedinstveni identifikator pretplate.
idVlasnika (FK)	BIGINT	Poveznica na vlasnika psa.

> Jedan vlasnik može imati najviše jednu aktivnu pretplatu.

Pas

Primary Key: idPsa
Foreign Key: idVlasnika → VlasnikPsa.idVlasnika

Atribut	Tip podatka	Opis varijable
idPsa (PK)	BIGINT	Jedinstveni identifikator psa.
zdravPsa	VARCHAR(20)	Zdravstveno stanje/opis.
energijaPsa	VARCHAR(20)	Razina energije.
posPsa	VARCHAR(20)	Poslastice koje smije primati.
socPsa	VARCHAR(20)	Socijalnost psa.
imePsa	VARCHAR(20)	Ime psa.
starostPsa	INT	Starost u godinama.
pasminaPsa	VARCHAR(20)	Pasmina psa.
idVlasnika (FK)	INT	Vlasnik psa.

> Jedan vlasnik može imati više pasa (1:N).

Šetač

Primary Key: idSetac
Unique: emailSetac, urnSetac
Foreign Key: idClanarine → Clanarina.idClanarine

Atribut	Tip podatka	Opis varijable
idSetac (PK)	BIGINT	Jedinstveni identifikator šetača.
emailSetac (U)	VARCHAR(50)	E-mail šetača (jedinstven).
imeSetaca	VARCHAR(20)	Ime šetača.
prezimeSetaca	VARCHAR(20)	Prezime šetača.
urnSetac (U)	VARCHAR(20)	Korisničko ime (jedinstveno).

Atribut	Tip podatka	Opis varijable
passSetac	VARCHAR(20)	Hash lozinke.
telefonSetaca	VARCHAR(32)	Kontakt telefon.
idClanarine (FK)	BIGINT	Veza s članarinom.
datReg	DATE	Datum registracije.
avgOcjena	NUMERIC(3,2)	Prosječna ocjena (1–5).

> Šetač može imati više termina u tablici Setnja.

Članarina

Primary Key: idClanarine
Foreign Key: idSetac → Setac.idSetac

Atribut	Tip podatka	Opis varijable
idClanarine (PK)	BIGINT	Jedinstveni identifikator članarine.
idSetac (FK)	BIGINT	Šetač kojem članarina pripada.
tipClanarine	VARCHAR(20)	Tip članarine (npr. standard/premium).
datumUplate	DATE	Datum uplate članarine.
vrijediDo	DATE	Datum isteka članarine.

Šetnja

Primary Key: idSetnje
Foreign Keys: idSetac → Setac.idSetac, idPsa → Pas.idPsa

Atribut	Tip podatka	Opis varijable
idSetnje (PK)	BIGINT	Jedinstveni identifikator šetnje.
idSetac (FK)	BIGINT	Šetač koji vodi šetnju.
idPsa (FK)	BIGINT	Pas koji sudjeluje u šetnji.
datumSetnje	DATE	Datum šetnje.
vrijemePocetka	TIME	Vrijeme početka šetnje.
vrijemeZavrsetka	TIME	Vrijeme završetka šetnje.
trajanje	INTERVAL	Ukupno trajanje šetnje.
lokacija	VARCHAR(50)	Lokacija ili ruta šetnje.
napomena	TEXT	Dodatne napomene o šetnji.

Recenzija

Primary Key: idRecenzije
Foreign Keys: idVlasnika → VlasnikPsa.idVlasnika, idSetac → Setac.idSetac, idSetnje → Setnja.idSetnje

Atribut	Tip podatka	Opis varijable
idRecenzije (PK)	BIGINT	Jedinstveni identifikator recenzije.
idVlasnika (FK)	BIGINT	Vlasnik koji je ostavio recenziju.
idSetac (FK)	BIGINT	Šetač na kojeg se odnosi recenzija.
idSetnje (FK)	BIGINT	Šetnja na koju se odnosi recenzija.
ocjena	NUMERIC(2,1)	Ocjena šetača (1–5).
komentar	TEXT	Opisna recenzija vlasnika.
datumRecenzije	DATE	Datum kada je recenzija ostavljena.

> Svaka šetnja može imati samo jednu recenziju (1:1).

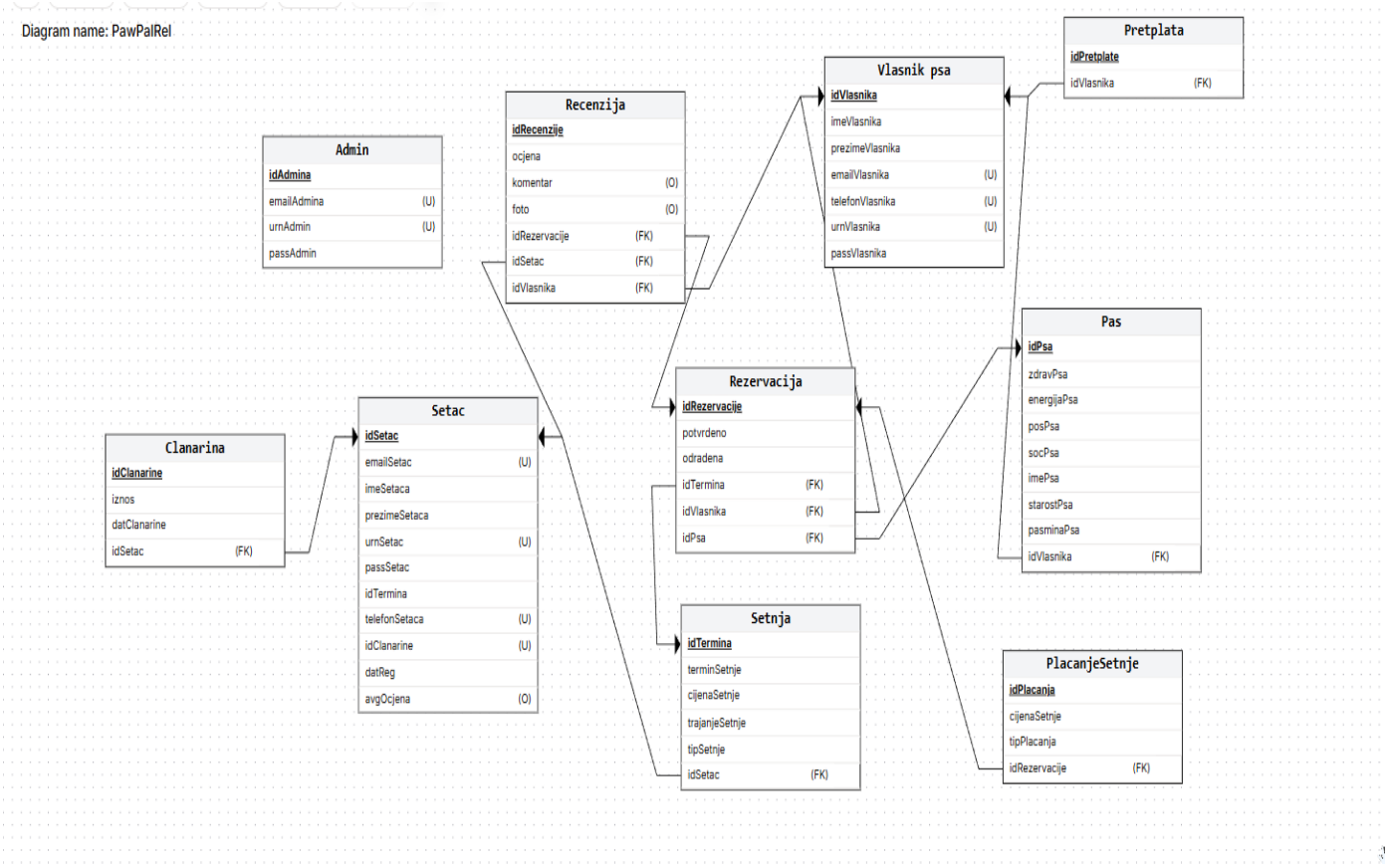
PasSetnja

Primary Keys: idPsa, idSetnje
Foreign Keys: idPsa → Pas.idPsa, idSetnje → Setnja.idSetnje

Atribut	Tip podatka	Opis varijable
`idPsa` (PK, FK)	BIGINT	Identifikator psa.
`idSetnje` (PK, FK)	BIGINT	Identifikator šetnje.

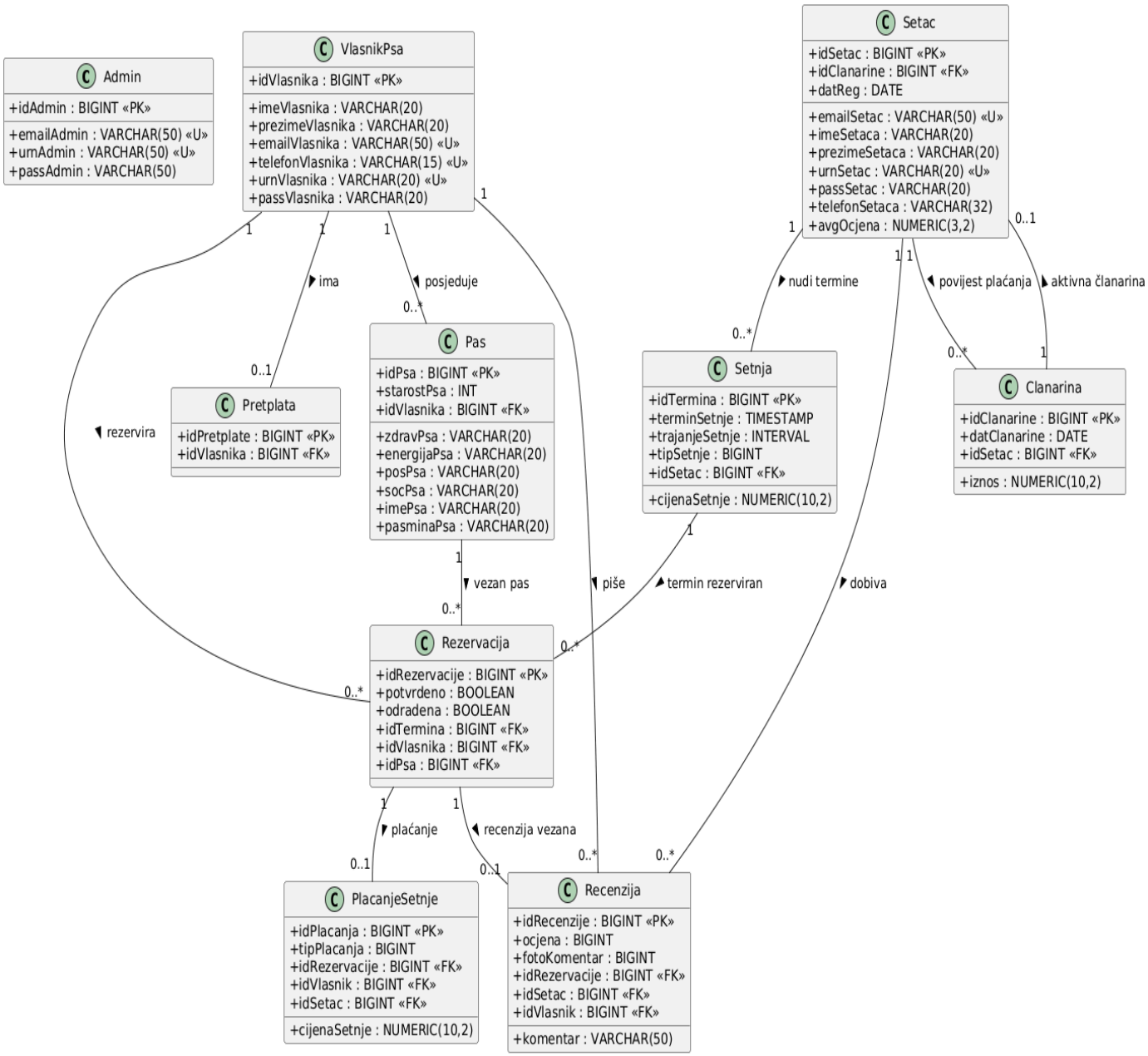
> Relacijska tablica koja omogućuje da više pasa sudjeluje u jednoj šetnji (M:N).

4.5.2. Dijagram baze podataka



> Slika 4.3. Relacijska shema baze podataka

4.6. Dijagram razreda



Slika 4.4. Dijagram razreda

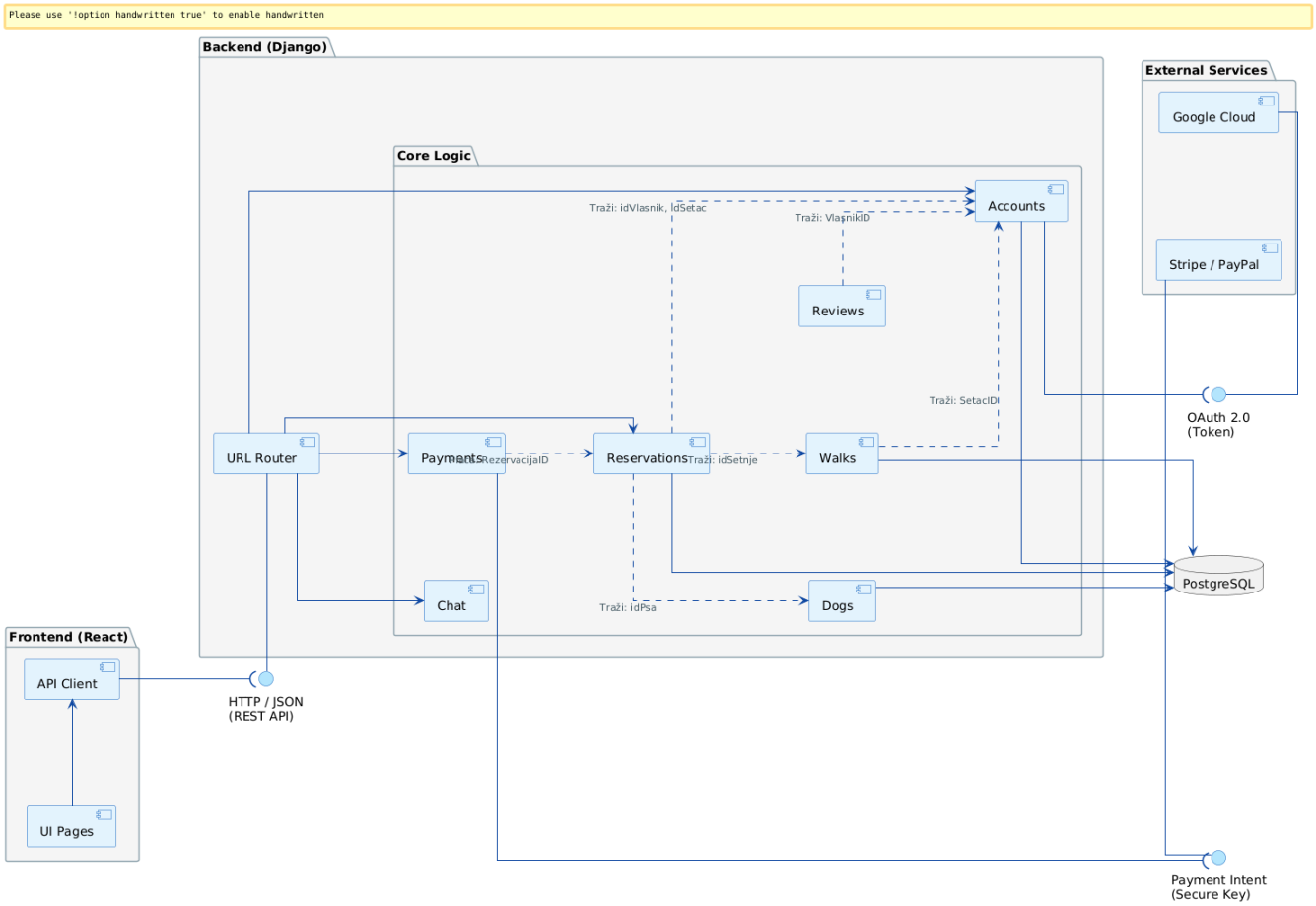
\newpage

5. Arhitektura komponenata i razmještaja

U ovom poglavlju dokumentirana je logička struktura programskog rješenja PawPal te fizička distribucija softverskih artefakata unutar produkcijskog okruženja. Arhitektura sustava projektirana je prema načelima modularnosti i odvajanja odgovornosti (*Separation of Concerns*), pri čemu su prezentacijski sloj i sloj poslovne logike implementirani kao zasebne cjeline koje komuniciraju putem mrežnog sučelja.

5.1. Dijagram komponenata

Sustav se sastoji od **Frontend aplikacije** i **Backend aplikacije** koje funkcioniraju neovisno. Komunikacija između njih odvija se isključivo putem asinkronih HTTP zahtjeva definiranih REST API specifikacijom, što osigurava labavu povezanost (*loose coupling*) između klijentske i poslužiteljske strane.



Slika 5.1. Dijagram komponenata sustava PawPal

5.1.1. Backend komponente

Poslužiteljska strana implementirana je koristeći **Django** okvir. Arhitektura backenda je modularna, gdje je svaka funkcionalna cjelina implementirana kao zasebna Django aplikacija (*app*), registrirana u `INSTALLED_APPS` konfiguraciji (`settings.py`). Uloge pojedinih komponenata su sljedeće:

- **`pawpal_backend`:** Predstavlja središnju konfiguracijsku točku sustava. Sadrži globalne postavke u `settings.py` te definira glavnu tablicu usmjeravanja (`urls.py`) koja delegira ulazne HTTP zahtjeve odgovarajućim funkcionalnim modulima.
- **`accounts`:** Modul zadužen za upravljanje identitetima i autorizaciju. Koristi biblioteku `django-allauth` za implementaciju registracije i prijave, uključujući integraciju s Google OAuth 2.0 protokolom definiranu u `SOCIALACCOUNT_PROVIDERS`.
- **`reservations`:** Implementira središnju poslovnu logiku rezerviranja usluga. Ovaj modul ne koristi stroge relacijske ključeve na razini baze podataka za sve entitete, već logički povezuje zapise pohranom identifikatora (`idVlasnik`, `idSetac`, `idPsa`, `idSetnje`), što je vidljivo u definiciji modela `Rezervacija`.
- **`walks`:** Modul odgovoran za upravljanje ponudom usluga, uključujući definiranje termina, cijena i vrsta šetnji.
- **`dogs`:** Komponenta za upravljanje podacima o ljubimcima, koja omogućuje kreiranje, čitanje, ažuriranje i brisanje (CRUD) profila pasa.
- **`reviews`:** Modul za upravljanje povratnim informacijama koji putem stranih ključeva (`ForeignKey`) izravno referencira korisničke modele (`Vlasnik`, `Setac`), omogućujući sustav ocjenjivanja sudionika.
- **`payments`:** Modul zadužen za financijske operacije. Sadrži logiku za komunikaciju s vanjskim procesorima plaćanja (Stripe i PayPal), koristeći API ključeve definirane u varijablama okruženja.
- **`chat`:** Modul koji omogućuje komunikaciju između korisnika platforme.

5.1.2. Frontend komponente

Klijentska aplikacija izgrađena je na **React** biblioteci i strukturirana je slojevito unutar `src` direktorija kako bi se odvojila logika prikaza od logike komunikacije:

- **API Layer (`src/api`):** Ovaj sloj centralizira mrežnu komunikaciju. Moduli poput `client.js` i `auth.js` sadrže definicije HTTP poziva prema backendu, čime se postiže apstrakcija API krajnjih točaka od ostatka aplikacije.
- **UI Components (`src/components`):** Sadrži ponovno iskoristive elemente sučelja, kao što su navigacija (`Navbar.jsx`) ili gumbi za prijavu (`GoogleLoginButton.jsx`), koji osiguravaju konzistentnost vizualnog identiteta.
- **Pages (`src/pages`):** Predstavljaju komponente visoke razine koje objedinjuju UI elemente u cjelovite ekrane (npr. `Home.jsx`, `Register.jsx`, `WalkerProfile.jsx`). Svaka stranica odgovara specifičnoj ruti definiranoj u `React Router` konfiguraciji.

5.2. Globalni upravljački tok

Sustav primjenjuje klijent-poslužitelj model komunikacije. Razmjena podataka odvija se putem **HTTP** protokola, pri čemu se podaci prenose u **JSON** formatu. Zbog sigurnosnih ograničenja web preglednika, na poslužiteljskoj strani konfiguriran je **CORS** (*Cross-Origin Resource Sharing*) koji dopušta zahtjeve s definiranih izvora (`CORS_ALLOWED_ORIGINS` u `settings.py`).

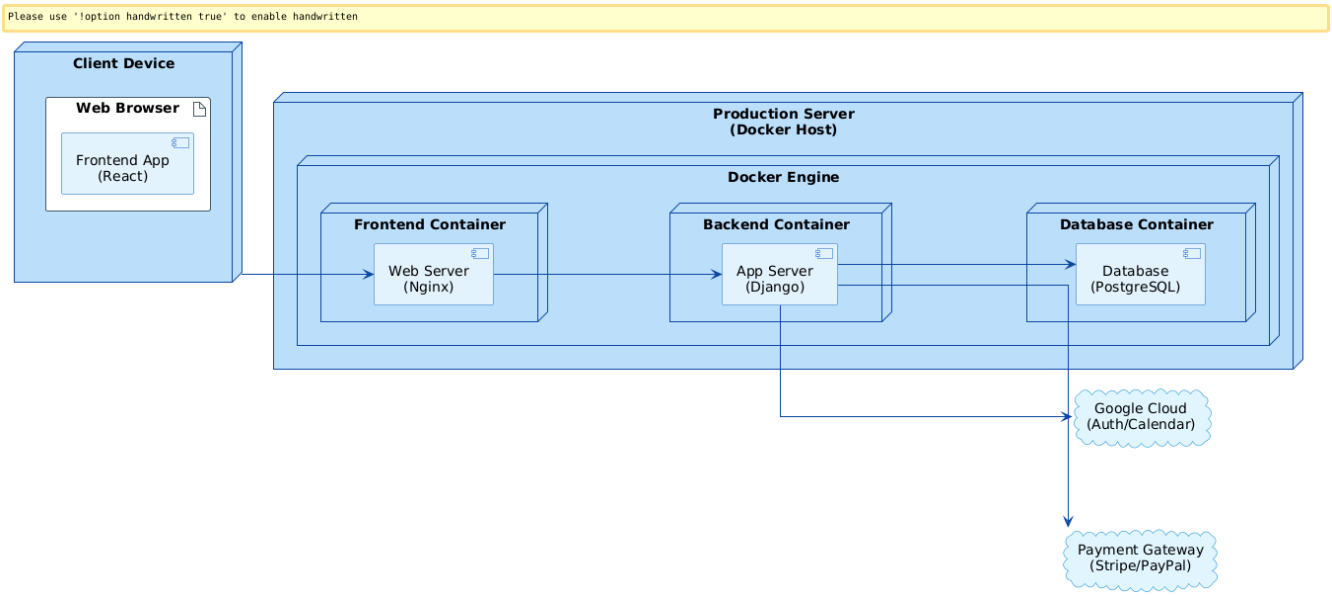
5.2.1. Primjer toka podataka: Proces rezervacije

Proces kreiranja rezervacije odvija se kroz sljedeće korake:

1. **Slanje zahtjeva:** Korisnik putem sučelja inicira akciju rezervacije. Frontend aplikacija prikuplja podatke (odabrani termin, pas, šetač) i šalje asinkroni `POST` zahtjev na odgovarajuću API točku.
2. **Usmjeravanje i Autentifikacija:** Django usmjerivač prosljeđuje zahtjev modulu `reservations`. Prije obrade, sustav provjerava valjanost autentifikacijskog tokena (Session/JWT).
3. **Poslovna logika:** Serijalizeri (`serializers.py`) validiraju ulazne podatke, provjeravajući postojanje referenciranih ID-ova (psa i termina) unutar baze podataka.
4. **Perzistencija:** Nakon uspješne validacije, podaci se trajno pohranjuju u tablicu `Rezervacija` unutar PostgreSQL baze podataka.
5. **Odgovor i Integracija:** Poslužitelj vraća status uspjeha (HTTP 201), nakon čega frontend aplikacija inicira preusmjeravanje na proces naplate, pozivajući PayPal ili Stripe integraciju definiranu u komponenti `PaymentModal.jsx`.

5.3. Dijagram razmještaja

Dijagram razmještaja prikazuje fizičku arhitekturu sustava u produkcijskom okruženju. Implementacija se oslanja na **Docker kontejnerizaciju**, što je potvrđeno prisutnošću `Dockerfile` definicija u izvornom kodu i `docker-compose.yml` datotekom za orkestraciju servisa.



Slika 5.2. Dijagram razmještaja sustava PawPal

5.3.1. Infrastruktura i Kontejneri

Sustav se sastoji od tri izolirana kontejnera koja komuniciraju unutar jedinstvene virtualne mreže:

1. **Frontend Kontejner:** Sadrži React aplikaciju serviranu putem web poslužitelja (Nginx ili Vite Preview). Kontejner izlaže port (standardno 80 ili 5173) prema vanjskoj mreži kako bi omogućio pristup korisnicima putem web preglednika.
2. **Backend Kontejner:** Izvršava Django aplikaciju pokrenutu putem Gunicorn WSGI poslužitelja na portu 8000. Ovaj kontejner nije izravno dostupan s interneta, već prihvaća zahtjeve isključivo od frontend kontejnera ili putem definiranog API prolaza. Konfiguriran je da bazu podataka dohvaća putem mrežnog imena `"db"`, kako je definirano u `settings.py` (`DATABASES` konfiguracija).
3. **Database Kontejner:** Izvršava PostgreSQL sustav za upravljanje bazom podataka. Podaci se pohranjuju na perzistentnom Docker volumenu radi trajnosti. Mrežni pristup ograničen je na internu Docker mrežu (port 5432), čime se onemogućuje neovlašteni vanjski pristup podacima.

5.3.2. Integracija s vanjskim servisima

Aplikacija je konfigurirana za sigurnu komunikaciju s vanjskim API servisima:

- **Google Cloud Platform:** Koristi se za autentifikaciju korisnika putem protokola OAuth 2.0. Integracija je izvedena pomoću biblioteke `django-allauth`, a parametri pružatelja usluge definirani su u `SOCIALACCOUNT_PROVIDERS`.
- **Procesori plaćanja (Stripe & PayPal):** Sustav komunicira s financijskim servisima putem HTTPS protokola. Autentifikacija prema ovim servisima vrši se pomoću tajnih ključeva (`STRIPE_SECRET_KEY`, `PAYPAL_CLIENT_ID`) koji se učitavaju iz varijabli okruženja, osiguravajući da osjetljivi podaci nisu tvrdo kodirani u izvornom kodu.

\newpage

6. Ispitivanje programskog rješenja

Ovo poglavlje treba opisati provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i

izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka, kao i upotrebu odgovarajućih alata za provedbu testiranja.

6.1. **Ispitivanje komponenti**

Za testiranje REST API-ja korišten je Swagger (OpenAPI 3.0).
Ova komponenta služi kao most između backend logike i frontend razvoja, pružajući jasan uvid u sve dostupne komponente sustava.

6.2. Ispitni slučajevi

Ispitni slučajevi raspoređeni su po kategorijama:

6.2.1. Admin

- `users_list`: dohvaća popis svih registriranih korisnika u sustavu, poredanih prema identifikacijskom broju, te njihove osnovne informacije.
- `user_disable`: onemogućuje pristup sustavu određenom korisniku.
- `user_enable`: ponovno aktivira prethodno isključen korisnički račun.
- `get_clanarina`: pomoćna funkcija koja dohvaća postojeći podatak o članarini iz baze ili stvara novi s početnim iznosom nula ako on ne postoji.
- `clanarina_get`: prikazuje trenutni iznos članarine i vrijeme njezine posljednje promjene putem administrativnog sučelja.
- `clanarina_update`: omogućuje administratoru da postavi novi iznos članarine za šetače, uz validaciju unesenih podataka.

6.2.2. Auth

- `csrf`: omogućuje klijentskoj aplikaciji da uspostavi važeći CSRF token.
- `google_login_url`: dohvaća URL za autentifikaciju putem Google računa, omogućujući sustavu da preusmjeri korisnika na vanjski servis za sigurnu prijavu bez ručnog unosa lozinke.
- `login_view`: omogućuje prijavu u testni račun, jer je za neke funkcije nužno biti prijavljen.
- `logout_view`: omogućuje odjavu iz testnog računa.
- `me`: vraća podatke o trenutno prijavljenom korisniku.
- `register`: stvara novi račun.
- `enable_walker`: pretvara vlasnika u ownera.
- `available_walkers`: pretražuje šetače prema ocjenama i lokaciji.
- `notification_events`: poll koji klijentu dostavlja informacije o novim događajima.

6.2.3. Chat

- `list_conversations`: dohvaća sve aktivne razgovore prijavljenog korisnika.
- `get_conversation`: dohvaća sve poruke iz odabranog razgovora.
- `get_or_create_conversation_from_reservation`: dohvaća ili stvara novi razgovor iz postojeće rezervacije.
- `get_or_create_conversation`: dohvaća ili stvara novi razgovor s drugim korisnikom.
- `send_message`: omogućuje korisniku slanje tekstualne poruke unutar odabranog razgovora.
- `list_users`: prikazuje listu ostalih korisnika platforme s kojima je moguće započeti komunikaciju.

6.2.4. Dogs

- `get_AllDogs`: vraća listu svih pasa prijavljenog korisnika.
- `dog_detail`: dohvaća podatke o psu s unesenim ID-jem.
- `dog_delete`: briše psa s unesenim ID-jem.
- `dog_update`: mijenja podatke o psu s unesenim ID-jem.
- `dogs_create`: stvara novog psa kojem je vlasnik prijavljeni korisnik.

6.2.5. Notifications

- `toggle_notifications`: uključuje ili isključuje obavijesti.

6.2.6. Payments

- `get_paypal_access_token`: pomoćna funkcija koja komunicira s PayPal API-jem radi dobivanja autorizacijskog tokena potrebnog za obavljanje transakcija.
- `create_payment_intent`: stvara zahtjev za plaćanje i vraća korisniku `approval_url` ili `client_secret`, ovisno o načinu plaćanja.
- `confirm_paypal_payment`: potvrđuje PayPal transakciju te ažurira status plaćanja u bazi podataka.
- `confirm_stripe_payment`: provjerava status Stripe transakcije putem njihovog SDK-a i, u slučaju uspjeha, označava plaćanje kao dovršeno u sustavu PawPal.
- `get_payment_status`: omogućuje klijentskoj aplikaciji provjeru trenutnog stanja određene transakcije.
- `get_user_payments`: dohvaća povijest svih plaćanja za prijavljenog vlasnika psa.

6.2.7. Reservations

- `accept_reservation`: (dostupno je samo vlasniku), prihvaća rezervaciju s unesenim ID-jem.
- `create_reservation`: (dostupno je samo šetaču), stvara rezervaciju.

- `create_reservation_from_walk`: (dostupno je samo šetaču), stvara rezervaciju iz postojeće šetnje.
- `delete_reservation`: briše rezervaciju.
- `mark_walk_done`: označava da je rezervacija sa zadanim ID-jem obavljena.
- `get_my_reservations`: dohvaća listu svih rezervacija prijavljenog korisnika.
- `reject_reservation`: (dostupno je samo šetaču), šetač ovom funkcijom može odbiti šetnju.

6.2.8. Reviews

- `get_walker_average`: dohvaća prosječnu ocjenu šetača.
- `create_review`: dodaje ocjenu šetaču.
- `delete_review`: (dostupno je samo vlasniku), briše ocjenu sa zadanim ID-jem.
- `get_my_reviews`: dohvaća recenzije prijavljenog korisnika.

6.2.9. Walker

- `enable`: pretvara prijavljenog vlasnika u šetača.

6.2.10. Walkers

- `walkers`: dohvaća listu svih šetača.

6.2.11. Walks

- `walks_list`: dohvaća sve planirane šetnje za trenutno prijavljenog šetača.
- `walks_create`: omogućuje šetaču otvaranje novog termina šetnje uz definiranje tipa (individualna/grupna), cijene, trajanja i lokacije.
- `walk_detail`: prikazuje detaljne informacije o jednoj specifičnoj šetnji prijavljenog šetača.
- `walk_update`: omogućuje šetaču izmjenu podataka postojećeg termina, poput promjene cijene ili vremena početka.
- `walk_delete`: trajno uklanja termin šetnje iz sustava.
- `get_AllWalks`: administrativna funkcija koja dohvaća cjelokupan popis svih šetnji u sustavu, bez obzira na šetača.
- `get_available_walks`: filtrira i prikazuje samo buduće šetnje koje su slobodne za rezervaciju.

\newpage

7. Tehnologije za implementaciju aplikacije

Ovo poglavlje pruža detaljan pregled tehnološkog stoga korištenog za razvoj, testiranje i implementaciju sustava PawPal. Odabir navedenih tehnologija temelji se na analizi zahtjeva za skalabilnošću, sigurnošću i modularnošću. Specifikacije verzija preuzete su iz konfiguracijskih datoteka `requirements.txt` i `package.json`.

7.1. Backend tehnologije

Razvoj poslužiteljske strane temelji se na programskom jeziku **Python (v3.12)**. Ovaj ekosustav odabran je zbog svoje zrelosti, čitljivosti koda i opsežne podrške biblioteka za web razvoj.

- **Django 5.1.2:** Glavni web radni okvir (*framework*) visoke razine. Odabran je zbog svoje arhitekture koja potiče brz razvoj, ugrađenih sigurnosnih mehanizama (zaštita od SQL injekcija, XSS, CSRF) i robusnog ORM sustava.
- **Django REST Framework 3.15.2:** Biblioteka korištena za izgradnju RESTful API sučelja. Omogućuje serijalizaciju složenih tipova podataka (poput Django modela) u JSON format, te upravljanje autentifikacijom i dozvolama na razini API točaka.
- **Gunicorn 23.0.0:** Produkcijski WSGI HTTP poslužitelj za UNIX sustave. Koristi se kao sučelje između Python aplikacije i web poslužitelja unutar Docker kontejnera, osiguravajući efikasno upravljanje paralelnim zahtjevima.
- **Psycopg2-binary 2.9.11:** PostgreSQL adapter za Python. Ova biblioteka omogućuje Python aplikaciji uspostavljanje sigurne i performantne veze s bazom podataka, podržavajući napredne značajke PostgreSQL-a.
- **Django-Allauth 65.1.0:** Sveobuhvatno rješenje za upravljanje autentifikacijom. U projektu se koristi za implementaciju standardne prijave te integraciju s Google računima putem OAuth 2.0 protokola.
- **PyJWT 2.10.1:** Biblioteka za kodiranje i dekodiranje JSON Web Tokena (JWT). Koristi se za implementaciju *stateless* autentifikacije, što je prikladno za SPA arhitekturu gdje poslužitelj ne mora pamtiti sesiju.
- **Pillow 11.1.0:** Biblioteka za obradu slika (fork originalnog PIL-a). Koristi se za validaciju, promjenu veličine i manipulaciju korisničkih avatara i fotografija ljubimaca prije pohrane.
- **Drf-spectacular 0.28.0:** Alat za automatsko generiranje API dokumentacije prema OpenAPI 3.0 standardu. Omogućuje vizualizaciju API sheme putem Swagger UI sučelja.

7.2. Frontend tehnologije

Klijentska strana aplikacije izvedena je kao **Single Page Application (SPA)** temeljena na modernom JavaScript stogu, što osigurava bogato korisničko iskustvo bez potrebe za osvježavanjem stranice.

- **React 19.1.1:** Biblioteka za izgradnju korisničkih sučelja. Njena deklarativna priroda i arhitektura temeljena na komponentama omogućuju efikasan razvoj složenih UI elemenata i upravljanje stanjem aplikacije.
- **React Router DOM 7.9.5:** Biblioteka za usmjeravanje na klijentskoj strani. Omogućuje navigaciju između različitih pogleda aplikacije manipulacijom URL-a preglednika, zadržavajući pritom stanje SPA aplikacije.

- **Vite 7.1.7:** Alat za izgradnju (*build tool*) nove generacije. Koristi se zbog iznimno brzog pokretanja razvojnog servera (koristeći native ES module) i optimiziranog pakiranja koda za produkciju.
- **Node.js (v20+):** Izvršno okruženje za JavaScript izvan preglednika. U projektu se koristi kao platforma za pokretanje razvojnih alata i upravljanje ovisnostima paketa putem `npm`-a.

7.3. Baza podataka

- **PostgreSQL:** Sustav koristi PostgreSQL kao primarnu relacijsku bazu podataka, što je definirano konfiguracijom `django.db.backends.postgresql` u `settings.py`. Odabran je zbog svoje pouzdanosti, sukladnosti s ACID principima i podrške za složene transakcije, što je nužno za integritet podataka o rezervacijama i korisnicima.

7.4. Razvojni alati i okruženje

Za osiguranje kvalitete, konzistentnosti i kolaboracije tijekom razvojnog procesa korišteni su sljedeći alati:

- **Visual Studio Code:** Napredno integrirano razvojno okruženje (IDE) s podrškom za ekstenzije specifične za Python, React i Docker sintaksu.
- **Git & GitHub:** Sustav za distribuiranu kontrolu verzija i platforma za pohranu koda. Omogućuje praćenje povijesti promjena, grananje koda (*branching*) i suradnju tima.
- **Docker & Docker Compose:** Platforma za kontejnerizaciju aplikacija. Omogućuje definiranje cjelokupnog okruženja (backend, frontend, baza) kroz kod (`docker-compose.yml`), osiguravajući da aplikacija radi identično na svim razvojnim i produkcijskim strojevima.
- **Discord:** Platforma za komunikaciju u stvarnom vremenu, korištena za koordinaciju tima, dnevne sastanke i dijeljenje resursa.

7.5. Integracije i vanjski servisi

Sustav se povezuje s vanjskim servisima radi proširenja funkcionalnosti bez povećanja kompleksnosti vlastite infrastrukture:

- **PayPal & Stripe API:** Integrirani su servisi za sigurnu obradu online plaćanja. Implementacija se nalazi u modulu `payments` na backendu te odgovarajućim React komponentama na frontendu.
- **Google OAuth 2.0:** Omogućuje korisnicima prijavu u sustav koristeći postojeće Google vjerodajnice, čime se pojednostavljuje proces registracije.

\newpage

8. Upute za puštanje u pogon

Ovaj odjeljak dokumentacije daje detaljne smjernice za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije PawPal. Sustav je dizajniran za rad u kontejneriziranom okruženju, što osigurava identično ponašanje na različitim operacijskim sustavima.

8.1. Instalacija

8.1.1. Preduvjeti

Za ispravno pokretanje sustava potrebno je osigurati sljedeće softverske preduvjete na računalu domaćina:

- **Docker Desktop:** Verzija **24.0+** (uključuje Docker Compose v2).
- **Git:** Verzija **2.30+**.
- **Python** (Opcionalno, za lokalno pokretanje skripti): Verzija **3.12**.

8.1.2. Preuzimanje izvornog koda

Otvorite terminal i klonirajte repozitorij na lokalno računalo:

```
git clone https://github.com/Nikolaaa777/Progi---13.4---PawPal.git
cd Progi---13.4---PawPal
```

8.2. Postavke

Sustav dolazi s pred-konfiguriranim postavkama za razvoj.

8.2.1. Konfiguracijska datoteka (.env)

Datoteka `.env` već se nalazi u mapi `bekend` i uključena je u repozitorij radi jednostavnijeg ocjenjivanja i postavljanja.

Sadržaj datoteke (`bekend/.env`):

- `DEBUG=True`: Omogućuje detaljan ispis grešaka.
- `ALLOWED_HOSTS`: Definira dozvoljene domene (localhost, 127.0.0.1, backend).
- `DATABASE_URL`: Connection string za PostgreSQL unutar Docker mreže.

- `GOOGLE_CLIENT_ID` / `SECRET`: Ključevi za Google OAuth integraciju.

8.2.2. Instalacija Python ovisnosti (Opcionalno)

Iako Docker automatski instalira potrebne pakete unutar kontejnera, za pregled koda ili lokalno izvršavanje skripti preporučuje se instalacija ovisnosti u virtualno okruženje:

```
cd bekend
pip install -r requirements.txt
cd ..
```

8.3. Pokretanje aplikacije (Lokalno)

Sustav se pokreće koristeći Docker Compose orkestraciju. Izvršite sljedeće naredbe u terminalu unutar korijenskog foldera projekta:

1. Priprema okruženja:

```
docker compose down -v
```

(Ova naredba zaustavlja i uklanja sve prethodne kontejnere i volumene kako bi se osigurao čist start.)

2. Podizanje sustava:

```
docker compose up
```

(Dodajte zastavicu `-d` za pokretanje u pozadini. Prvo pokretanje može potrajati nekoliko minuta dok se ne preuzmu Docker slike.)

3. Inicijalizacija baze podataka: Otvorite **novi terminal** i pokrenite migracije unutar aktivnog backend kontejnera:

```
# 1. Kreiranje migracija
docker compose exec backend python manage.py makemigrations

# 2. Primjena migracija
docker compose exec backend python manage.py migrate

# 3. Primjena specifične SQL sheme (Trigeri i procedure)
docker compose exec backend python scripts/manual_schema/apply_manual_schema.py
```

8.3.1. Provjera rada

- **Frontend:** http://localhost:5173
- **Backend API:** http://localhost:8000

\newpage

9. Zaključak i budući rad

9. Zaključak i budući rad

Projektni zadatak izrađivan je u ograničenom vremenskom razdoblju, što je bilo dovoljno za realizaciju osnovnih funkcionalnosti aplikacije. Zbog složenosti sustava i opsega zahtjeva, dio planiranih funkcionalnosti nije implementiran.

Tijekom izrade prepoznati su tehnički izazovi vezani uz integraciju vanjskih servisa (autentifikacija, plaćanje, kalendar) te komunikaciju u stvarnom vremenu. Neki izazovi nisu riješeni zbog vremenskih ograničenja, ali bi se mogli riješiti dodatnim istraživanjem i boljim planiranjem arhitekture.

Rad na projektu omogućio je stjecanje znanja iz analize zahtjeva, timskog rada i razvoja web aplikacija. Za bržu i kvalitetniju realizaciju bila bi korisna naprednija znanja o sigurnosti, vanjskim API-jima i bazama podataka.

Projekt ima dobru osnovu za nastavak rada i daljnje proširenje funkcionalnosti.

\newpage

A. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Kloniran službeni GitHub repozitorij dostupan u sklopu predmeta	Nikola Ivković	20.10.2025.
0.2	Napisani funkcijski i nefunkcijski zahtjevi te opis projektnog zadatka	Nikola Ivković, Antonio Stjepić i Bruno Cavor	23.10.2025.
0.3	Prepravljeni funkcijski i nefunkcijski zahtjevi	Bruno Cavor	27.10.2025.
0.4	Započet rad na cjelini 3. Specifikacija zahtjeva sustava	Bruno Cavor	12.11.2025.
0.5	Napravljena cjelina 4. Arhitektura i dizajn sustava osim organizacije aplikacije	Antonio Stjepić i Bruno Cavor	12.11.2025.
0.6	Nastavljen rad na cjelini 3. Specifikacija zahtjeva sustava	Bruno Cavor	13.11.2025.
0.7	Dovršena cjelina 4. Arhitektura i dizajn sustava	Nikola Ivković i Zvonimir Schönwald	13.11.2025.
0.8	Nastavljen rad na cjelini 3. Specifikacija zahtjeva sustava	Bruno Cavor	13.11.2025.
0.9	Dodatak svih vizualnih elemenata	Bruno Cavor	14.11.2025.
1.0	Preinake u indeksiranju cjelina, ispravak pisanih pogrešaka, formatiranje teksta	Bruno Cavor	14.11.2025.
1.1	Dovršen rad na cjelini 3. Specifikacija zahtjeva sustava	Bruno Cavor	14.11.2025.
1.2	Preinaka funkcijskih zahtjeva u 2. cjelini	Bruno Cavor	14.11.2025.
1.3	Izmjena popisa literature	Nikola Ivković i Bruno Cavor	14.11.2025.
1.4	Popunjavanje dnevnika promjena dokumentacije	Bruno Cavor	14.11.2025.
1.5	Izmjena početne stranice repozitorija	Bruno Cavor	13.01.2026.
1.6	Izmjena opisa projektnog zadatka	Bruno Cavor	13.01.2026.
1.7	Napravljena cjelina 5. Arhitektura komponenata i razmještaja	Vid Veselko	18.01.2026.
1.8	Napravljena cjelina 6. Ispitivanje programskog rješenja	Vid Veselko	21.01.2026.
1.9	Dodana cjelina C. Prikaz aktivnosti grupe	Bruno Cavor	23.01.2026.
2.0	uređena cjelina B. Popis literature	Bruno Cavor	23.01.2026.
2.1	uređena cjelina B. Popis literature	Bruno Cavor	23.01.2026.

\newpage

B. Popis literature

Kontinuirano osvježavanje Literatura korištena prilikom izrade 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. PlantUML, https://plantuml.com/
8. https://docs.djangoproject.com/en/5.2/topics/auth/
9. https://docs.djangoproject.com/en/5.2/topics/db/models/
10. https://www.geeksforgeeks.org/python/python-django-google-authentication-and-fetching-mails-from-scratch/

\newpage

C. Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak – Formiranje tima

- **Datum:** 10. listopada 2025.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Upoznavanje članova tima i očekivanja
    - **LOŠE:** nejasna očekivanja o obvezama
    - **BOLJE:** dogovorena osnovna pravila suradnje
  - Definiranje cilja projekta
  - Otvoren GitHub repozitorij
  - Kreiran Discord server

Donesene odluke: Projekt se razvija timski uz tjedne sastanke
Zaduženja: Svi članovi – pregledati opis projekta

---

2. sastanak – Podjela uloga

- **Datum:** 16. listopada 2025.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Analiza potrebnih uloga u projektu
  - Podjela odgovornosti:
    - Voditelj
    - Backend tim
    - Frontend tim
    - Tim za dokumentaciju i testiranje

Donesene odluke: Svaki član ima ulogu.
Zaduženja: Cijeli tim – izrada opisa projekta

---

3. sastanak – Planiranje funkcionalnosti

- **Datum:** 30. listopada 2025.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Definiranje glavnih funkcionalnosti aplikacije
  - Postavljanje prioriteta (MVP)

Donesene odluke: Prvo razviti osnovne funkcije
Zaduženja: Cijeli tim – izrada funkcijskih zahtjeva

---

4. sastanak – Tehnologije i alati

- **Datum:** 8. studenoga 2025.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Odabir programskog jezika i frameworka
  - Izrada baze podataka, frontenda i backenda

Donesene odluke: Izrada sustava za prvu predaju.
Zaduženja: Cijeli tim – izrada segmenata po podtimu

---

5. sastanak – Pregled napretka

- **Datum:** 9. siječnja 2026.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Pregled implementiranih funkcionalnosti
  - Identifikacija problema
  - Dogovor oko rokova

Donesene odluke: Prilagodba vremenskog plana
Zaduženja: Frontend – dorada UI-a

---

6. sastanak – Testiranje i ispravci

- **Datum:** 16. siječnja 2026.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Analiza bugova
  - Dogovor o načinu testiranja
  - Evidencija problema na GitHubu

Donesene odluke: Obavezno testiranje prije spajanja grana
Zaduženja: Tim za dokumentaciju – opis testnih scenarija

---

7. sastanak – Završna priprema

- **Datum:** 23. siječnja 2026.
- **Prisustvovali:** Nikola Ivković, Zvonimir Schonwald, Alan Miljak, Vid Veselko, Antonio Stjepić i Bruno Cavor
- **Teme sastanka:**
  - Završni pregled projekta

- Priprema prezentacije i dokumentacije
- Dogovor oko predaje projekta

Donesene odluke: Projekt spreman za predaju
Zaduženja: Svi – završni commit i provjera dokumentacije

Plan rada

- Gantogram

Aktivnost	30.9–6.10	7.10–13.10	14.10–20.10	21.10–27.10	28.10–3.11	4.11–10.11	11.11–17.11	18.11–24.11	25.11–1.12	2.12–8.12	9.12–15.12	1(2
Okupljanje tima	okupljanje tima											
Pisanje dokumentacije		dokumentacija	dokumentacija	dokumentacija	dokumentacija							
Razvoj frontenda			frontend	frontend	frontend	frontend					frontend	fr
Razvoj backenda				backend	backend	backend						
Razvoj baza podataka			baze podataka	baze podataka								
Deplozment projekta				deployment								

Tablica aktivnosti

- Stavke su navedene u krajnje lijevom stupcu.
- Imena članova su navedena u prvom redu.

Stavka	Nikola Ivković	Zvonimir Schonwald	Vid Veselko	Antonio Stjepić	Alan Miljak	Bruno Cavor
Upravljanje projektom	10	3				15
Opis projektnog zadatka	2					
Funkcionalni zahtjevi	2			2		10
Opis pojedinih obrazaca						10
Dijagram obrazaca				5		15
Sekvencijski dijagrami					5	
Opis ostalih zahtjeva						1
Arhitektura i dizajn sustava				5		
Baza podataka	5			16		16
Dijagram razreda						1
Dijagram stanja						
Dijagram aktivnosti						
Dijagram komponenti						
Korištene tehnologije i alati	0.5					0.5
Ispitivanje programskog rješenja					4	
Dijagram razmještaja						
Upute za puštanje u pogon						
Dnevnik sastajanja						1
Zaključak i budući rad						2
Popis literature						1
Dodatne stavke – podjela izrade aplikacije						
Izrada početne stranice		6				3
Izrada baze podataka	2			2		2

Stavka	Nikola Ivković	Zvonimir Schonwald	Vid Veselko	Antonio Stjepić	Alan Miljak	Bruno Cavor
Spajanje s bazom podataka	10			10	10	
Izrada prezentacije			2			
Backend	100	10		20	10	10
Frontend	10	80		120		100

Dijagram pregleda promjena

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

Ključni izazovi i rješenja

Prilikom izrade projekta tim se susreo s brojnim novim izazovima koji su itekako utjecali na razvoj projekta, no unatoč navedenim izazovima tim je implementirao veliku većinu zadanih funkcionalnosti.

Opis izazova

Izazovi koji su utjecali na razvoj projekta su:

- Kašnjenje u razvoju - Nedostatak znanja i iskustva u radu s odabranim tehnologijama znatno je usporio i otežao razvoj projekta. Osim prethodno navedenog, na razvoj su utjecale i ostale fakulteske obveze.
- Timska oragnizacija - Neadekvatna angažiranost određenih članova tima prilikom razvoja, posljedično i manjak timske komunikacije.
- Tehnički problemi - Integracija vanjskih servisa te povezivanje manjih cjelina u funkcionalni sustav.
- Dokumentacija - Osiguravanje kvalitetne i upotrebljive dokumentacije.
- Motivacija tima - Održavanja radne discipline na adekvatnoj razini tijekom zahtjevnih perioda.

Rješenja i naučene lekcije

Kako se navedene poteškoće minimizirale, a potencijalno i u potpunosti uklonile tim je primijenio sljedeće:

- Jasan plan rada - Jasno i pravovremeno definiranje ciljeva koje uključuje sve članove tima.
- Komunikacija - Redovita komunikacija između svih članova tima kako bi se osigurao neometani razvoj projekta.
- Tehnička rješenja - Izazovi implementacije vanjskih servisa svladani su proučavanjem tehničke dokumentacije dok je problem deploymenta riješen uz pomoć Rendera.
- Timski duh i motivacija - Razvoju projekta tim treba pristupati kao cjelina koja brine o svim svojim članovima i koja rješava problem kao cjelina.

Zaključak

Prilikom razvoja projekta tim se susreo s brojnim izazovima, no upravo izazovi su doveli do velikog napretka u svim područjima. Izazovi su osvijetstili tim o brojnim važnim stavkama u razvoju kojih tim nije niti bio svjestan poput toga koliko je komunikacija neizostavna.

U slučaju puštanja projekta u stvarnu upotrebu bilo bi potrebno osigurati dodatnu sigurnost podataka, optimizirati performanse, unaprijediti postojeće funkcionalnosti, poboljšati modularnost koda, itd.

Funkcionalnost koja nije implementirana je povezivanje sustava s Google Calendar API-jem kao niti sustav recenziranja.

Kao član tima, smatram da je tim pokazao izuzetnu sposobnost učenja i implementacije, no određeni izazovi su nanijeli popriličnu štetu koja nažalost nije u potpunosti sanirana. Unatoč tome što je tim poprilično upoznat s timskim radom, ovaj projekt je pokazao kako je timski rad neizostavna komponenta razvoja.

\newpage

_Footer

Sarma

\newpage