

Programsko inženjerstvo

Ak. god. 2023./2024.

DentAll

Dokumentacija, Rev. 1

Grupa: *ZuBit*

Voditelj: *Danko Delimar*

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

Nastavnik: *Goran Rajić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Sekvencijski dijagrami	20
3.2 Ostali zahtjevi	23
4 Arhitektura i dizajn sustava	24
4.1 Baza podataka	24
4.1.1 Opis tablica	25
4.1.2 Dijagram baze podataka	28
4.2 Dijagram razreda	29
4.3 Dijagram stanja	33
4.4 Dijagram aktivnosti	35
4.5 Dijagram komponenti	36
5 Implementacija i korisničko sučelje	37
5.1 Korištene tehnologije i alati	37
5.2 Ispitivanje programskog rješenja	38
5.2.1 Ispitivanje komponenti	38
5.2.2 Ispitivanje sustava	38
5.3 Dijagram razmještaja	39
5.4 Upute za puštanje u pogon	40
6 Zaključak i budući rad	41
Popis literature	42
Indeks slika i dijagrama	43

Dodatak: Prikaz aktivnosti grupe

44

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	*	20.10.2023.
0.2	Dodan opis projekta, dionici, funkcionalni zahtjevi i neki use casevi	*	25.10.2023.
0.3	Dodan ostatak <i>Use Caseva</i>	*	29.10.2023.
0.6		*	26.08.2013.
0.8	Povijest rada i trenutni status implementacije, Zaključci i plan daljnjeg rada	*	28.08.2013.
0.9	Opisi obrazaca uporabe	*	07.09.2013.
0.10	Preveden uvod	*	08.09.2013.
0.11	Sekvencijski dijagrami	*	09.09.2013.
0.12.1	Započeo dijagrame razreda	*	10.09.2013.
0.12.2	Nastavak dijagrama razreda	*	11.09.2013.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene:Timer - Brojilo vremena	*	15.09.2013.
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
2.0	Konačni tekst predložka dokumentacije	*	28.09.2013.

Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.

2. Opis projektnog zadatka

Ideja ovog projekta je napraviti aplikaciju koja olakšava dentalnim klinikama proces ponude potpunog plana liječenja koji uključuje medicinske podatke, smještaj u kojem će se pacijent nalaziti i prijevoz od klinike do smještaja i obratno. Ovo je važna aplikacija za klinike koje žele svoje usluge ponuditi pacijentima iz drugih gradova ili država, tzv. "dentalni turizam", zato što ih mogućnost takve ponude čini puno privlačnijima potencijalnim klijentima koji nisu upoznati s lokacijom na kojoj se klinika nalazi. Time se klijenti, a i klinike mogu fokusirati samo na posao, a naša aplikacija obavlja svu logistiku.

Aplikaciju mogu koristiti sve dentalne klinike kojima je potrebna organizacija smještaja i prijevoza na relaciji smještaj-klinika. Posebice ciljamo na klinike koje žele proširiti posao oglašavanjem u inozemstvu i žele ponuditi potpuni paket potencijalnim klijentima kako bi se istaknule naspram konkurencije. Takvim klinikama je ovakva aplikacija jako potrebna zato što je jedan od glavnih razloga zašto dentalni turizam nije popularan koliko bi mogao biti činjenica da je potencijalnim klijentima težak zadatak samostalno organizirati ovakvo putovanje u stranu državu o kojoj možda ništa ne znaju.

U trenutnom obliku aplikacija cilja dentalne klinike jer su zahvati koje dentalne klinike nude relativno jednostavni za definirati u smislu termina i jer taj sektor medicine ima veliki potencijal za privući ljude iz stranih država zbog značajno jeftinije usluge. Ukoliko je potrebno aplikacija se jednostavno može promijeniti i nadograditi kako bi uslugu mogla pružati cijelom medicinskom sektoru ili bilo kakvoj djelatnosti koja želi ponuditi takav potpuni paket smještaja, prijevoza i usluge.

Aplikacija se može koristiti iz perspektive tri administratora koji predstavljaju tri dijela paketa koje aplikacija spaja, a to su smještajni administrator, prijevozni administrator i korisnički administrator. Krajnji korisnik, tj. pacijent klinike, nema interakciju s aplikacijom nego samo na račun elektroničke pošte koji mu je definiran od strane korisničkog administratora prima informaciju o cijelom paketu smještaja, prijevoza, kliničkih usluga i datuma kada treba stići u državu.

Smještajni administrator ima najveće ovlasti u sustavu i može dodavati nove korisnike i dodjeljivati im uloge. Glavna uloga smještajnog administratora je dodavanje u sustav novih smještaja i popunjavanje osnovnih podataka o tim smještajima. Smještajnim administratorima je omogućen pregled lokacije stana na karti uz pomoć Google Maps API-a. Svi osnovni podatci se mogu mijenjati, a cijeli smještaj se može izbrisati. Osnovni podatci uključuju:

- Tip stana (u vlasništvu klinike ili ne)
- Veličina stana
- Ocjena stana
- Adresa
- Vremenska dostupnost

Prijevozni administrator ima glavnu ulogu u unosu osnovnih podataka o prijevoznicima koji su dostupni u aplikaciji. Osnovni osobni podatci prijevoznika se ne mogu mijenjati, svi ostali se mogu, a cijeli prijevoznik se može izbrisati. Osnovni podatci uključuju:

- Ime i prezime prijevoznika
- Email
- Broj telefona
- Tip prijevoznog sredstva
- Kapacitet prijevoznog sredstva
- Radno vrijeme

Korisnički administrator ima glavnu ulogu u unosu osnovnih podataka o pacijentima klinike. Detalje tretmana ne unose korisnički administratori već se oni dohvaćaju iz postojećeg sustava klinike. Moguće je mijenjati podatke o pacijentima i brisati pacijente. Osnovi podatci o pacijentima su:

- Ime i prezime pacijenta
- Email

- Broj telefona
- Vrijeme dolaska u državu
- Mjesto dolaska u državu
- Preferencija o veličini i ocjeni stana

Svaki korisnički račun može imati više administratorskih uloga. Svaki administrator ima web sučelje unutar kojeg može obavljati potrebne operacije dodavanja, izmjenjivanja i brisanja podataka. Takvo jednostavno web sučelje olakšava posao administratora kada moraju raditi potrebne izmjene.

Kada korisnički administrator unese novog korisnika aplikacija mora provjeriti postoji li dostupan smještaj u danom terminu. Kada se pronade termin u kojem postoji slobodan smještaj čeka se potvrda od interne aplikacije klinike da je zaključan plan tretmana. Kada se plan tretmana zaključa aplikacija automatski šalje poruku elektroničke pošte klijentu sa cijelim paketom i šalje dostupnim prijevoznicima informacije o klijentu i smještaju.

Aplikacija ne prati podatke o tretmanu pacijenta već podatke mora pratiti interna aplikacija klinike te se tada oni uključuju u kompletan plan iz spomenute aplikacije. Također, aplikacija ne dozvoljava nikakav pristup ili pregled neregistriranim korisnicima s obzirom da je zamišljena za internu uporabu unutra klinike, a ne za vanjsku uporabu od strane korisnika. Sva komunikacija s vanjskim korisnicima kao što su pacijenti ili prijevoznici odvija se automatski putem elektroničke pošte.

S obzirom na to da je glavna funkcionalnost aplikacije smanjenje mentalnog napora vezanog za logistiku, moguća proširenja aplikacije većinom su vezana za dodatno olakšanje logistike. Moguća je implementacija povezanosti s internacionalnim prijevoznicima kako bi klijenti klinika morali učiniti što je manje moguće. Također je moguća integracija sa sustavima vlakova ili avioprijevoznika kako bi se klijentu pružio potpuno organizirani plan puta do destinacije. Druga vrsta mogućeg proširenja je automatsko izračunavanje troška ovakvog plana i automatsko uračunavanje tog troška u cjelokupni trošak tretmana. U taj trošak bili bi uključeni troškovi poput troškova prijevoza i troškova čišćenja stanova u kojima klijenti borave za vrijeme boravka u državi. Povezano s time, u aplikaciju se mogu uključiti i usluge čistača kojima se automatski šalje poruka kada određena smještajna jedinica treba čišćenje.

U našem kratkom istraživanju sličnih aplikacija nismo pronašli aplikaciju koja rješava isti problem. Većina se aplikacija u sektoru dentalnog turizma bavi prikazom tretmana pacijentima, ali ne zadiru u pitanja kako doći do neke klinike ili gdje stanovati za vrijeme obavljanja usluge. Mislimo da zato ova aplikacija ima dobru priliku iskoristiti nišu u tržištu koja trenutno nije jako zastupljena. Također je moguće da za ovakvu funkcionalnost klinike trenutno imaju interna rješenja, no naša aplikacija će omogućiti da svaka zainteresirana klinika ima ovakve mogućnosti uz jednostavan oblik pretplate.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Dentalne klinike
2. Vlasnici smještaja
3. Prijevoznici
4. Administratori koji koriste aplikaciju
5. Klijenti, tj. pacijenti
6. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Smještajni administrator (inicijator) može:
 - (a) Pregledavati podatke o smještajima (tip, ocjena, adresa, period dostupnosti)
 - (b) Dodavati nove korisnike (ne klijente nego korisnike sustava)
 - (c) Korisnicima dodavati nove uloge
 - (d) Dodavati, izmjenjivati i brisati podatke o smještaju
 - (e) Vidjeti prikaz smještaja na karti

2. Prijevoznički administrator (inicijator) može:

- (a) Pregledavati podatke o prijevoznicima
- (b) Dodati osnovne osobne podatke prijevoznika
- (c) Dodati kontaktne podatke i podatke o vrsti i kapacitetu vozila
- (d) Izmjenjivati neosnovne podatke (kontakt, vrsta i kapacitet vozila)
- (e) Izbrisati prijevoznika

3. Korisnički administrator (inicijator) može:

- (a) Dodati podatke o klijentima (osobni podatci, kontakt, preferencije o smještaju)

4. Baza podataka (sudionik) može:

- (a) Pohranjuje sve podatke o prijevoznicima
- (b) Pohranjuje sve podatke o smještaju
- (c) Pohranjuje nemedicinske podatke o klijentima

UC1 - Prijava smještajnog administratora

- **Glavni sudionik:** Smještajni administrator
- **Cilj:** Prijaviti se u sustav kao smještajni administrator
- **Sudionici:** Baza podataka
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke te odabir uloge smještajnog administratora
 2. Potvrda o postojanju računa i ispravnosti podataka
 3. Pristup funkcijama smještajnog administratora
- **Opis mogućih odstupanja:**
 - 2.a Uneseni podatci nisu u točnom formatu
 1. Obavijestiti korisnika koji podatci nisu u točnom formatu
 2. Korisnik mijenja potrebne podatke i pokušava opet
 - 2.b Korisnički račun ne postoji, lozinka nije točna ili korisničko ime nije točno
 1. Obavijestiti korisnika da su korisničko ime ili lozinka netočni
 2. Korisnik mijenja potrebne podatke i pokušava opet

UC2 - Prijava prijevoznog administratora

- **Glavni sudionik:** Prijevozni administrator
- **Cilj:** Prijaviti se u sustav kao prijevozni administrator
- **Sudionici:** Baza podataka
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke te odabir uloge prijevoznog administratora
 2. Potvrda o postojanju računa i ispravnosti podataka
 3. Pristup funkcijama prijevoznog administratora
- **Opis mogućih odstupanja:**
 - 2.a Uneseni podatci nisu u točnom formatu
 1. Obavijestiti korisnika koji podatci nisu u točnom formatu
 2. Korisnik mijenja potrebne podatke i pokušava opet
 - 2.b Korisnički račun ne postoji, lozinka nije točna ili korisničko ime nije točno
 1. Obavijestiti korisnika da su korisničko ime ili lozinka netočni
 2. Korisnik mijenja potrebne podatke i pokušava opet

UC3 - Prijava korisničkog administratora

- **Glavni sudionik:** Korisnički administrator
- **Cilj:** Prijaviti se u sustav kao korisnički administrator
- **Sudionici:** Baza podataka
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke te odabir uloge korisničkog administratora
 2. Potvrda o postojanju računa i ispravnosti podataka
 3. Pristup funkcijama korisničkog administratora
- **Opis mogućih odstupanja:**
 - 2.a Uneseni podatci nisu u točnom formatu
 1. Obavijestiti korisnika koji podatci nisu u točnom formatu
 2. Korisnik mijenja potrebne podatke i pokušava opet
 - 2.b Korisnički račun ne postoji, lozinka nije točna ili korisničko ime nije točno
 1. Obavijestiti korisnika da su korisničko ime ili lozinka netočni
 2. Korisnik mijenja potrebne podatke i pokušava opet

UC4 - Dodavanje novog korisnika

- **Glavni sudionik:** Smještajni administrator
- **Cilj:** Dodati novog korisnika sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava smještajnog administratora
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke novog korisnika te odabir svih uloga koje će korisnik imati
 2. Provjera postoji li već korisnik s takvim imenom
 3. Upis novog korisnika u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Polja korisničko ime i/ili lozinka su prazni
 1. Obavijestiti korisnika koji su podatci prazni
 2. Korisnik mijenja potrebne podatke i pokušava opet
 - 2.b Postoji korisnički račun s tim imenom
 1. Obavijestiti korisnika da postoji račun s tim imenom
 2. Korisnik mijenja korisničko ime i pokušava opet

2.c Nije odabrana niti jedna uloga

1. Obavijestiti korisnika da račun mora imati neku ulogu
2. Korisnik dodaje jednu ili više uloga i pokušava opet

UC5 - Pregled postojećih smještaja

- **Glavni sudionik:** Smještajni administrator
- **Cilj:** Pregledati postojeće smještaje
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava smještajnog administratora
- **Opis osnovnog tijeka:**
 1. Smještajni administrator odabire opcije "Postojeći smještaji"
 2. Aplikacija u obliku liste prikazuje sve postojeće smještaje i kartu s njihovim adresama

UC6 - Dodavanje novog smještaja

- **Glavni sudionik:** Smještajni administrator
- **Cilj:** Dodati novi smještaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava smještajnog administratora
- **Opis osnovnog tijeka:**
 1. Unos potrebnih podataka za dodavanje novog smještaja
 2. Provjera jesu li uneseni svi podatci
 3. Upis novog smještaja u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Neki od podataka nije unesen
 1. Obavijestiti korisnika koji podatci su prazni
 2. Korisnik mijenja potrebne podatke i pokušava opet

UC7 - Promjena podataka smještaja

- **Glavni sudionik:** Smještajni administrator
- **Cilj:** Promijeniti podatke smještaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava smještajnog administratora
- **Opis osnovnog tijeka:**
 1. Smještajni administrator odabire "Promijeni" opciju na smještaju
 2. Smještajni administrator mijenja podatke o smještaju

3. Smještajni administrator sprema promjene
 4. Pregledava se ispravnost podataka
 5. Mijenjaju se podatci u bazi podataka
- **Opis mogućih odstupanja:**
 - 3.a Smještajni administrator izlazi bez spremanja
 1. Podatci se ne spremaju u bazu podataka
 - 4.a Neki od podataka nije unesen
 1. Obavijestiti smještajnog administratora koji podatci su prazni
 2. Smještajni administrator mijenja potrebne podatke i pokušava opet

UC8 - Brisanje smještaja

- **Glavni sudionik:** Smještajni administrator
- **Cilj:** Promijeniti podatke smještaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava smještajnog administratora
- **Opis osnovnog tijeka:**
 1. Smještajni administrator odabire "Izbriši" opciju na smještaju
 2. Aplikacija šalje "jeste li sigurni?" upit smještajnom administratoru
 3. Ako je, smještaj se briše iz baze, u protivnom se ne desi ništa

UC9 - Pregled postojećih prijevoznika

- **Glavni sudionik:** Prijevozni administrator
- **Cilj:** Pregledati postojeće prijevoznike
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava prijevoznog administratora
- **Opis osnovnog tijeka:**
 1. Prijevozni administrator odabire opcije "Postojeći prijevoznici"
 2. Aplikacija u obliku liste prikazuje sve postojeće prijevoznike

UC10 - Dodavanje novog prijevoznika

- **Glavni sudionik:** Prijevozni administrator
- **Cilj:** Dodati novog prijevoznika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava prijevoznog administratora
- **Opis osnovnog tijeka:**
 1. Unos potrebnih podataka za dodavanje novog prijevoznika

2. Provjera jesu li uneseni svi podatci
3. Upis novog prijevoznika u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Neki od podataka nije unesen
 1. Obavijestiti prijevoznog administratora koji podatci su prazni
 2. Prijevozni administrator mijenja potrebne podatke i pokušava opet

UC11 - Promjena podataka o prijevozniku

- **Glavni sudionik:** Prijevozni administrator
- **Cilj:** Promijeniti podatke o prijevozniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava prijevoznog administratora
- **Opis osnovnog tijeka:**
 1. Prijevozni administrator odabire "Promijeni" opciju na postojećem prijevozniku
 2. Prijevozni administrator mijenja podatke o prijevozniku
 3. Prijevozni administrator sprema promijene
 4. Pregledava se ispravnost podataka
 5. Mijenjaju se podatci u bazi podataka
- **Opis mogućih odstupanja:**
 - 3.a Prijevozni administrator izlazi bez spremanja
 1. Podatci se ne spremaju u bazu podataka
 - 4.a Neki od podataka nije unesen
 1. Obavijestiti prijevoznog administratora koji podatci su prazni
 2. Prijevozni administrator mijenja potrebne podatke i pokušava opet

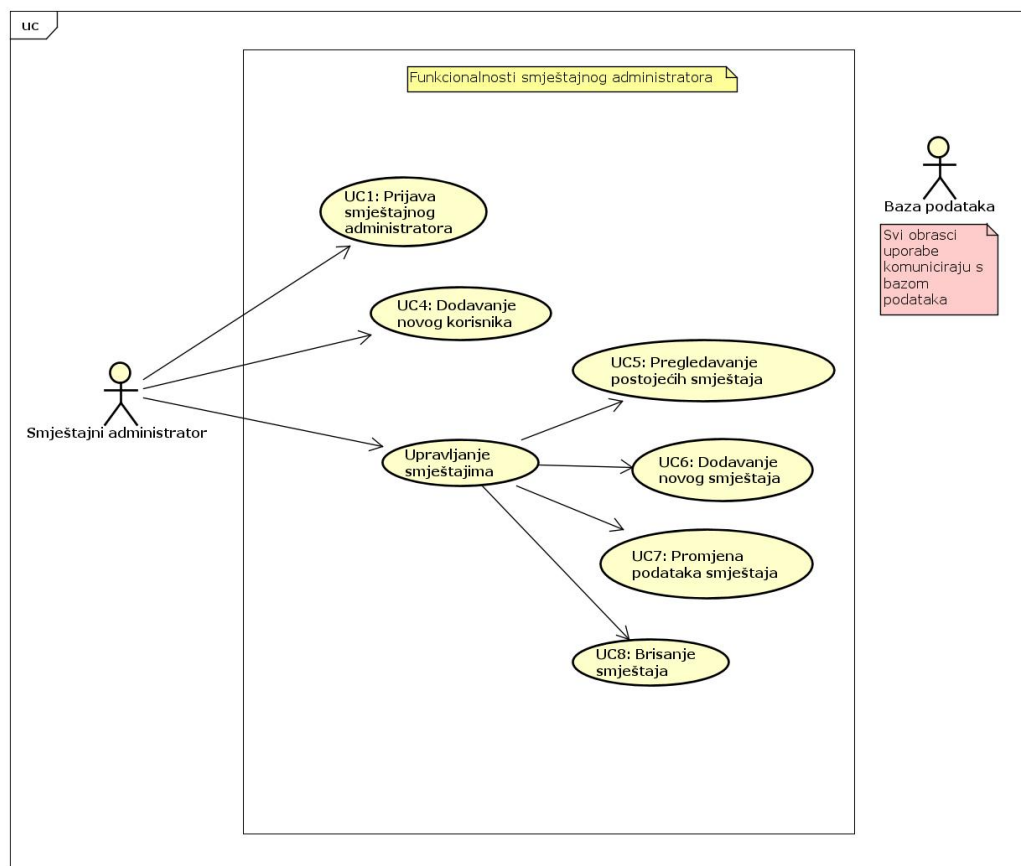
UC12 - Brisanje prijevoznika

- **Glavni sudionik:** Prijevozni administrator
- **Cilj:** Promijeniti podatke smještaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava prijevoznog administratora
- **Opis osnovnog tijeka:**
 1. Prijevozni administrator odabire "Izbriši" opciju na prijevozniku
 2. Aplikacija šalje "jeste li sigurni?" upit prijevoznom administratoru
 3. Ako je, prijevoznik se briše iz baze, u protivnom se ne desi ništa

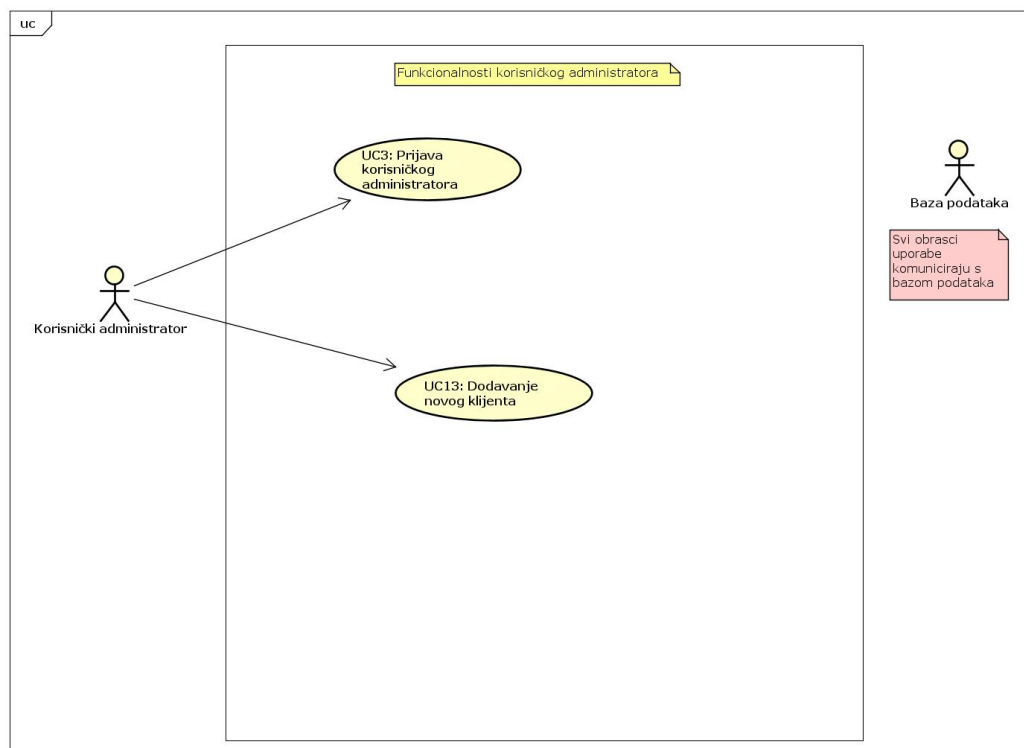
UC13 - Dodavanje novog klijenta

- **Glavni sudionik:** Korisnički administrator
- **Cilj:** Dodati novog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava korisničkog administratora
- **Opis osnovnog tijeka:**
 1. Unos potrebnih podataka za dodavanje novog korisnika
 2. Dohvat medicinskih podataka o korisniku
 3. Provjera jesu li uneseni svi podatci
 4. Dodjela smještaja klijentu
 5. Slanje poruke elektroničke pošte klijentu i prijevozniku o zaključenom planu
- **Opis mogućih odstupanja:**
 - 2.b Ne mogu se dohvatiti medicinski podatci
 1. Aplikacija ponovno pokušava dohvatiti podatke
 2. Ako je podatke nemoguće dohvatiti, nemoguće je unijeti novog korisnika
 - 3.a Neki od podataka nije unesen
 1. Obavijestiti korisničkog administratora koji podatci su prazni
 2. Korisnički administrator mijenja potrebne podatke i pokušava opet

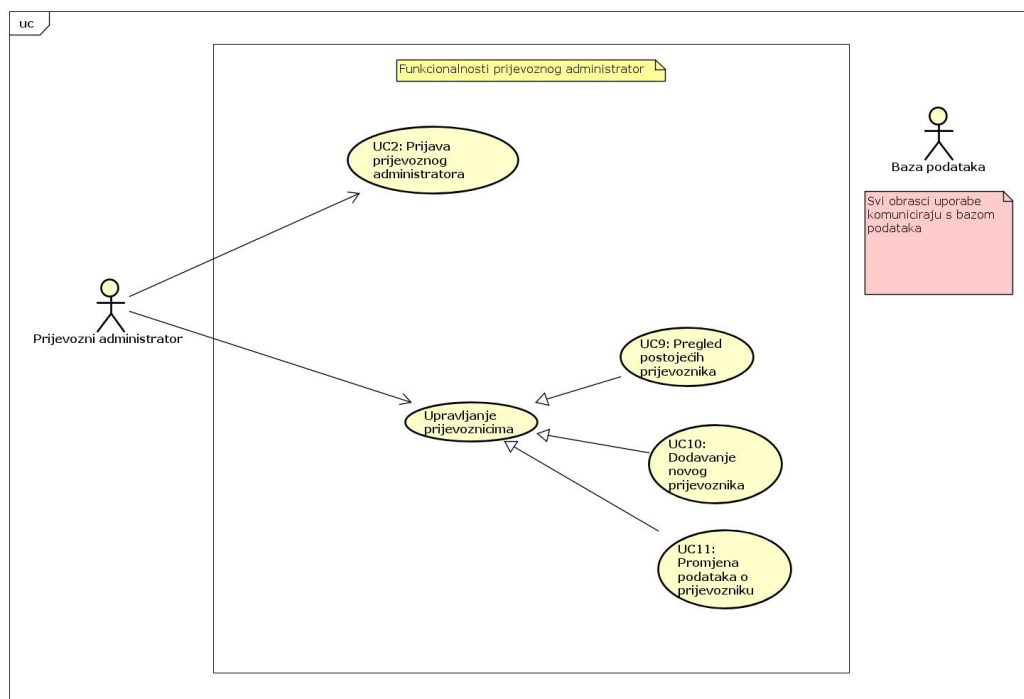
Dijagrami obrazaca uporabe



Slika 3.1: Funkcijski zahtjevi smještajnog administratora



Slika 3.2: Funkcijski zahtjevi transportnog administratora

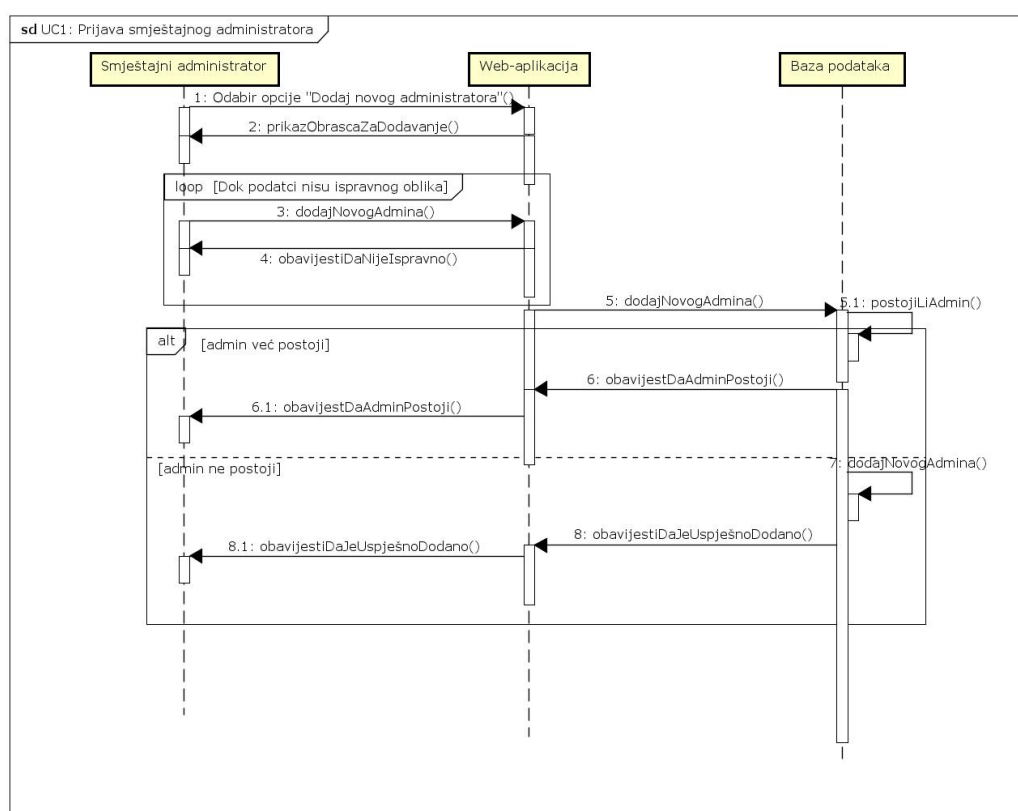


Slika 3.3: Funkcijski zahtjevi korisničkog administratora

3.1.1 Sekvencijski dijagrami

Obrazac uporabe UC4: Dodavanje novog korisnika

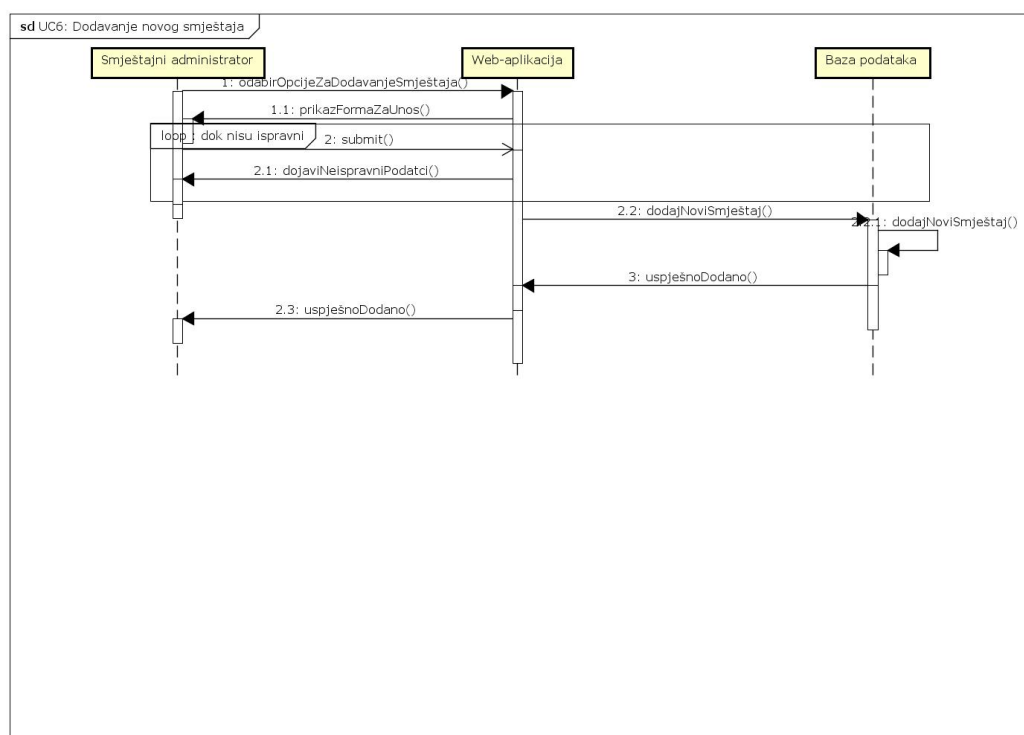
Smještajni administrator odabire opciju dodaj novog korisnika. Web-aplikacija otvara obrazac za dodavanje novog korisnika. Smještajni administrator upisuje korisničko ime, lozinku i odabire uloge koje će novi administrator imati te šalje zahtjev aplikaciji. Aplikacija radi provjeru ispravnosti tih podataka i ako su ispravni šalje ih bazi podataka. Baza podataka provjerava postoji li već administrator s tim korisničkim imenom i ako ne postoji dodaje novog administratora. Smještajnom administratoru aplikacija javlja da je unos uspješno izvršen.



Slika 3.4: Sekvencijski dijagram za UC4

Obrazac uporabe UC6: Dodavanje novog smještaja

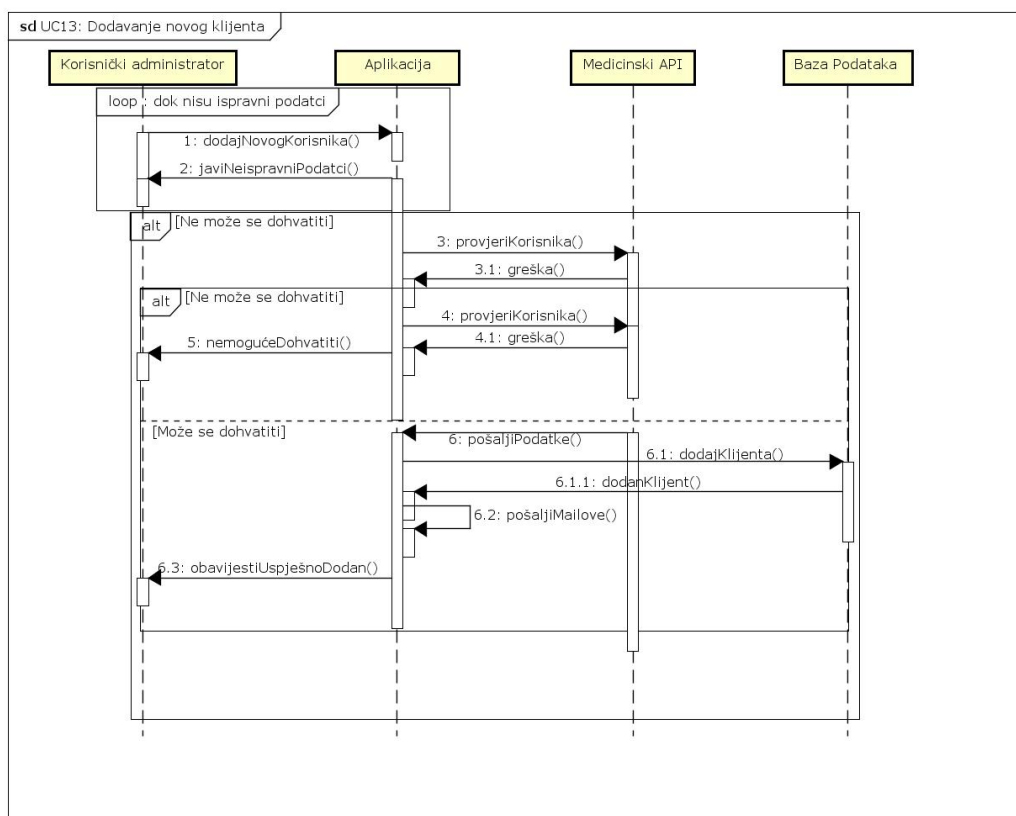
Smještajni administrator odabire opciju dodaj novi smještaj. Web-aplikacija otvara obrazac za dodavanje novog smještaja. Smještajni administrator upisuje sve potrebne informacije za dodavanje smještaja. Ako su svi potrebni podatci ispunjeni aplikacija šalje podatke bazi podataka. Baza podataka registrira novi smještaj i vraća informaciju da je smještaj ispravno unesen.



Slika 3.5: Sekvencijski dijagram za UC6

Obrazac uporabe UC13: Dodavanje novog klijenta

Korisnički administrator odabire opciju dodaj novog korisnika. Aplikacija mu pokazuje obrazac za dodavanje novog klijenta. Nakon što upiše sve potrebne podatke aplikacija pokušava dohvatiti podatke od klijenta iz API-a klinike. Ako prvi put ne uspije pokušava opet, a ako i drugi put ne uspije javlja administratoru da trenutno unos nije moguć. Ako je dohvat uspješan klijent se dodaje u bazu podataka i aplikacija šalje poruku elektroničke pošte klijentu i prijevozniku odgovornom za klijenta, a administratoru se javlja da je unos uspješan.



Slika 3.6: Sekvencijski dijagram za UC13

3.2 Ostali zahtjevi

- Sustav treba omogućiti udaljeni pristup administratorima
- Sustav treba omogućiti istovremeni rad više korisnika
- Sve operacije s bazom podataka moraju biti sigurne, a lozinke zaštićene
- Sustav bilo kakve greške treba dojaviti administratorima na pregledan način umjesto da sruši sustav
- Sustav mora poslati pravilno formatirane poruke elektroničke pošte kako one ne bi završile u spam-u
- Sustav mora biti dovoljno općenito implementiran kako bi bio lagano nadogradiv
- Sustav mora biti izgrađen koristeći principe objektno orijentiranog programiranja
- Sustav mora biti jednostavno izgrađen i svi obavezni podatci za unos moraju biti jasno naznačeni
- Greška u unosu ne smije srušiti sustav

4. Arhitektura i dizajn sustava

Za arhitekturu sustava odabrali smo klasičan klijent-server pristup.

Klijent

Strana klijenta je web stranica izgrađena u programskom jeziku JavaScript uz pomoć biblioteke React. Odabrali smo ovu tehnologiju jer je React danas najkorištenija biblioteka za razvoj web stranica i kao takva nudi najbolji ekosustav funkcionalnosti i podrške. Korišteno razvojno okruženje je VScode. Zadatak klijenta je slanje zahtjeva prema serveru koji ih zatim obrađuje. Svi zahtjevi se šalju pomoću HTTP POST metode i šalju se u JSON formatu prema serveru.

Server

Za server stranu odabrali smo programski jezik Javu i razvojno okruženje *Spring Boot*. *Spring Boot* smo odabrali jer je standardno razvojno okruženje za jezik Javu, a Javu smo odabrali kako bismo na prirodan način mogli sustav implementirati koristeći objektno orijentiranu paradigmu. Za razvoj serverskog koda korišten je alata IntelliJ IDEA. *Spring Boot* nam također nudi neke dodatne pogodnosti kao što je proširenje *Spring Security* koje znatno olakšava proces implementacije sigurnog i točnog procesa prijave i registracije korisnika. Server prima zahtjeve od klijenta u JSON formatu i pretvara te zahtjeve u Java objekte nad kojima izvršava daljnje operacije. Kada server obradi zahtjev šalje natrag HTTP odgovor s odgovarajućim statusnim kodom kako bi klijent znao je li operacija uspjela ili nije.

4.1 Baza podataka

Kao sustav za upravljanje bazama podataka odabrali smo PostgreSQL. Implementacija naše PostgreSQL baze podataka obuhvaća nekoliko ključnih elemenata, uključujući organizaciju podataka u tablicama i uspostavljanje veza između tablica radi složenih upita. Baza podatka sastoji se od slijedećih entiteta:

- KLINIKA
- SMJEŠTAJ

- PRIJEVOZNIK
- VOZILO
- KORISNIK
- PUTOVANJE

4.1.1 Opis tablica

KLINIKA Entitet KLINIKA sadrži informacije o ID-u klinike, nazivu i adresi. Prema tome, entitet KLINIKA posjeduje sljedeće attribute: IDKlinika, naziv i adresa. Entitet KLINIKA je u vezi *One-to-Many* s entitetom SMJESTAJ preko atributa IDKlinika i u vezi *One-to-Many* s entitetom PUTOVANJE preko atributa IDKlinika. Također je u *One-to-Many* vezi s entitetom PUTOVANJE preko atributa adresa.

KLINIKA		
IDKlinika	INT	Identifikacijski ključ klinike
naziv	VARCHAR	Naziv klinike
adresa	VARCHAR	Adresa klinike

SMJESTAJ Entitet SMJESTAJ sadrži podatke o ID-u smještaja, tipu stana, kategoriji opremljenosti, adresi kao i vremenskom periodu dostupnosti za korištenje. Sukladno tome, entitet SMJESTAJ posjeduje sljedeće attribute: IDSmjestaj, tip, kategorija, adresa i dostupnost. Entitet SMJESTAJ je u vezi *Many-to-One* s entitetom KLINIKA preko atributa IDKlinika i u vezi *One-to-Many* s entitetom PUTOVANJE preko atributa IDPutovanje. Također je u *One-to-Many* vezi s entitetom PUTOVANJE preko atributa adresa.

SMJESTAJ		
IDSmjestaj	INT	Identifikacijski ključ smještaja
tip	VARCHAR	Tip stana
kategorija	VARCHAR	Kategorija opremljenosti
adresa	VARCHAR	Adresa smještaja
dostupnost	INTERVAL	Vremenski period dostupnosti za korištenje

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

SMJESTAJ		
IDKlinika	INT	Identifikacijski ključ klinike

PRIJEVOZNIK Entitet PRIJEVOZNIK sadrži informacije o ID-u prijevoznika, kontaktnim podacima i o radnom vremenu u kojem je prijevoznik raspoloživ. Prema tome, entitet PRIJEVOZNIK posjeduje sljedeće attribute: IDPrijevoznik, kontakt i radnoVrijeme. Entitet PRIJEVOZNIK je u vezi *One-to-Many* s entitetom VOZILO preko atributa IDPrijevoznik i u vezi *One-to-Many* s entitetom PUTOVANJE preko atributa IDPrijevoznik.

PRIJEVOZNIK		
IDPrijevoznik	INT	Identifikacijski ključ prijevoznika
kontakt	VARCHAR	Kontaktni podatci prijevoznika
radnoVrijeme	TIME	Radno vrijeme u kojem su prijevoznici raspoloživi

VOZILO Entitet VOZILO sadrži informacije o ID-u vozila, vrsti i kapacitetu prijevoznog sredstva. Prema tome, entitet VOZILO posjeduje sljedeće attribute: IDVozilo, vrsta i kapacitet. Entitet VOZILO je u vezi *Many-to-One* s entitetom PRIJEVOZNIK preko atributa IDPrijevoznik.

VOZILO		
IDVozilo	INT	Identifikacijski ključ vozila
vrsta	VARCHAR	Vrsta vozila
kapacitet	VARCHAR	Kapacitet vozila
IDPrijevoznik	INT	Identifikacijski ključ prijevoznika

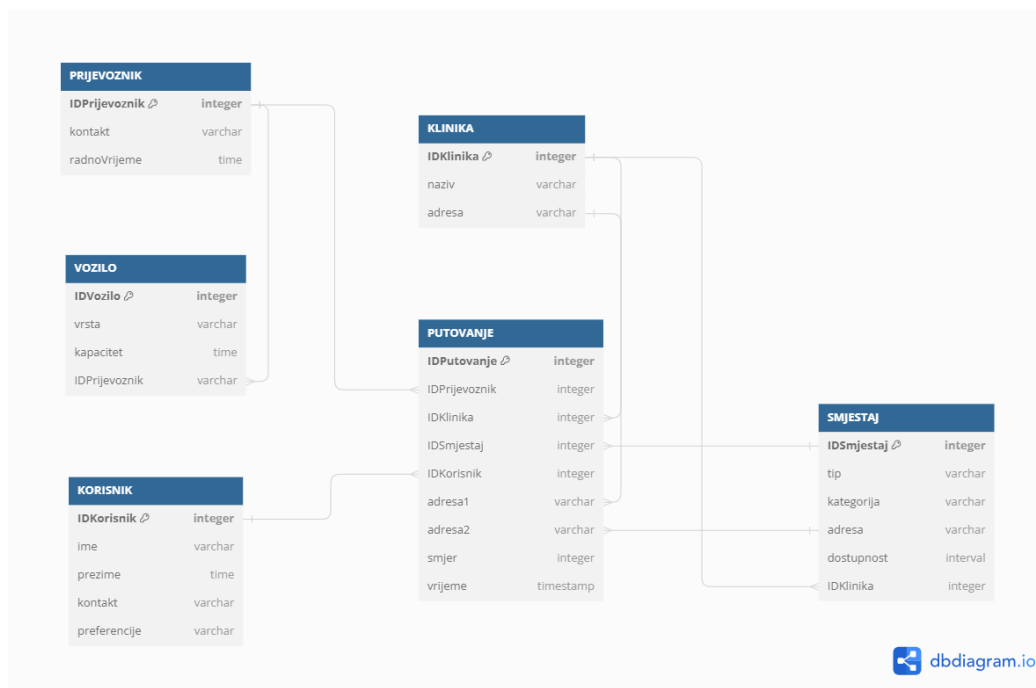
KORISNIK Entitet KORISNIK sadrži informacije o ID-u korisnika, imenu, prezimenu, kontaktnim podacima i preferencijama vezanim uz veličinu i kvalitetu smještaja. Prema tome, entitet KORISNIK posjeduje sljedeće attribute: IDKorisnik, ime, prezime, kontakt i preferencije. Entitet KORISNIK je u vezi *One-to-Many* s entitetom PUTOVANJE preko atributa IDKorisnik.

KORISNIK		
IDKorisnik	INT	Identifikacijski ključ korisnika
ime	VARCHAR	Ime korisnika
prezime	VARCHAR	Prezime korisnika
kontakt	VARCHAR	Kontakt korisnika
preferencije	VARCHAR	Preferencije vezane uz veličinu i kvalitetu smještaja

PUTOVANJE Entitet PUTOVANJE sadrži informacije o ID-u putovanja, vremenu i smjeru putovanja. Prema tome, entitet PUTOVANJE posjeduje sljedeće attribute: IDPutovanje, vrijeme i smjer. Entitet PUTOVANJE je u vezi *Many-to-One* s entitetom KLINIKA preko atributa IDKorisnik, u vezi *Many-to-One* s entitetom SMJESTAJ preko atributa IDSmjestaj, u vezi *Many-to-One* s entitetom KORISNIK preko atributa IDKorisnik, u vezi *Many-to-One* s entitetom PRIJEVOZNIK preko atributa IDPrijevoznik. Atributi adresa1 i adresa2 su atributi iz kojih saznajemo adresu polaska ili dolaska u ovisnosti o smjeru koji može biti 1 ili 0. Entitet PUTOVANJE u vezi je *Many-to-One* s entitetom KLINIKA preko atributa adresa1, u vezi *Many-to-One* s entitetom SMJESTAJ preko atributa adresa2.

PUTOVANJE		
IDPutovanje	INT	Identifikacijski ključ putovanja
vrijeme	TIME	Vrijeme putovanja
smjer	INT	Smjer u kojem se putovanje izvodi
adresa1	VARCHAR	Adresa klinike
adresa2	VARCHAR	Adresa smještaja
IDKorisnik	INT	Identifikacijski ključ korisnika
IDKlinika	INT	Identifikacijski ključ klinike
IDPrijevoznik	INT	Identifikacijski ključ prijevoznika
IDSmjestaj	INT	Identifikacijski ključ smještaja

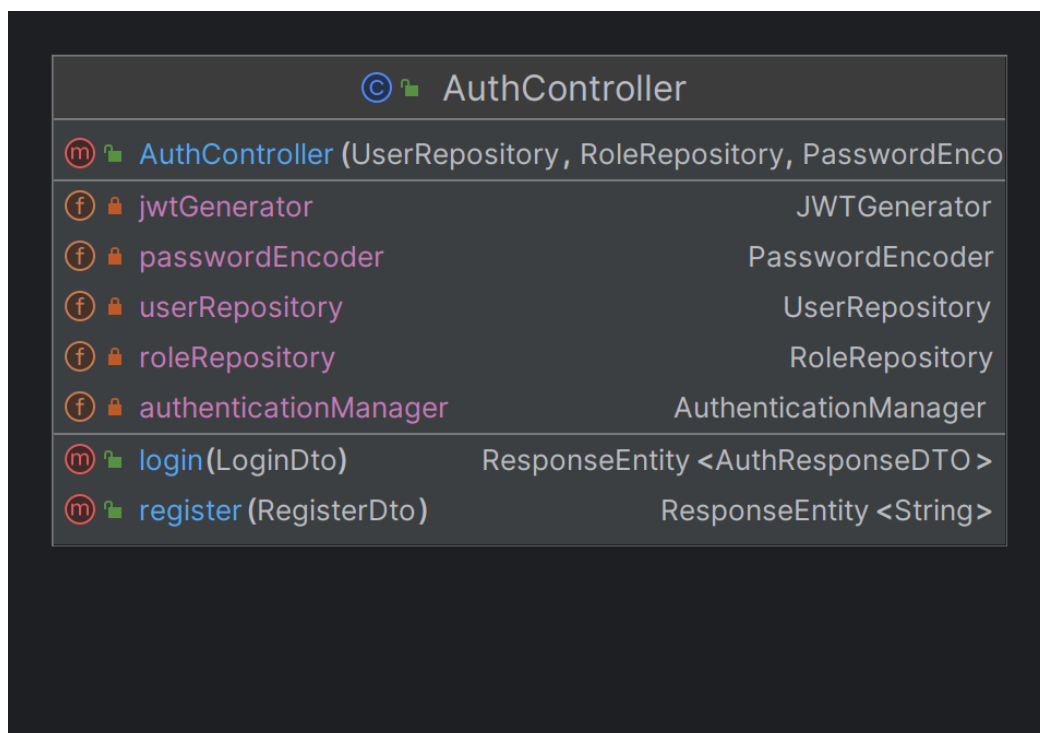
4.1.2 Dijagram baze podataka



Slika 4.1: Relacijska shema baze podataka

4.2 Dijagram razreda

Na slikama 4.2, 4.3, 4.4 i 4.5 prikazani su razredi za implementaciju funkcionalnosti prijave i registracije korisnika. Na slici 4.2 prikazan je razred *AuthController* koji služi prihvatanju HTTP zahtjeva od strane klijenta i to specifično za URL */auth/***. Metode *login()* i *register()* služe kao URL-ovi */auth/login* i */auth/register* na koje se šalju JSON objekti za prijavu administratora i registraciju novog administratora.

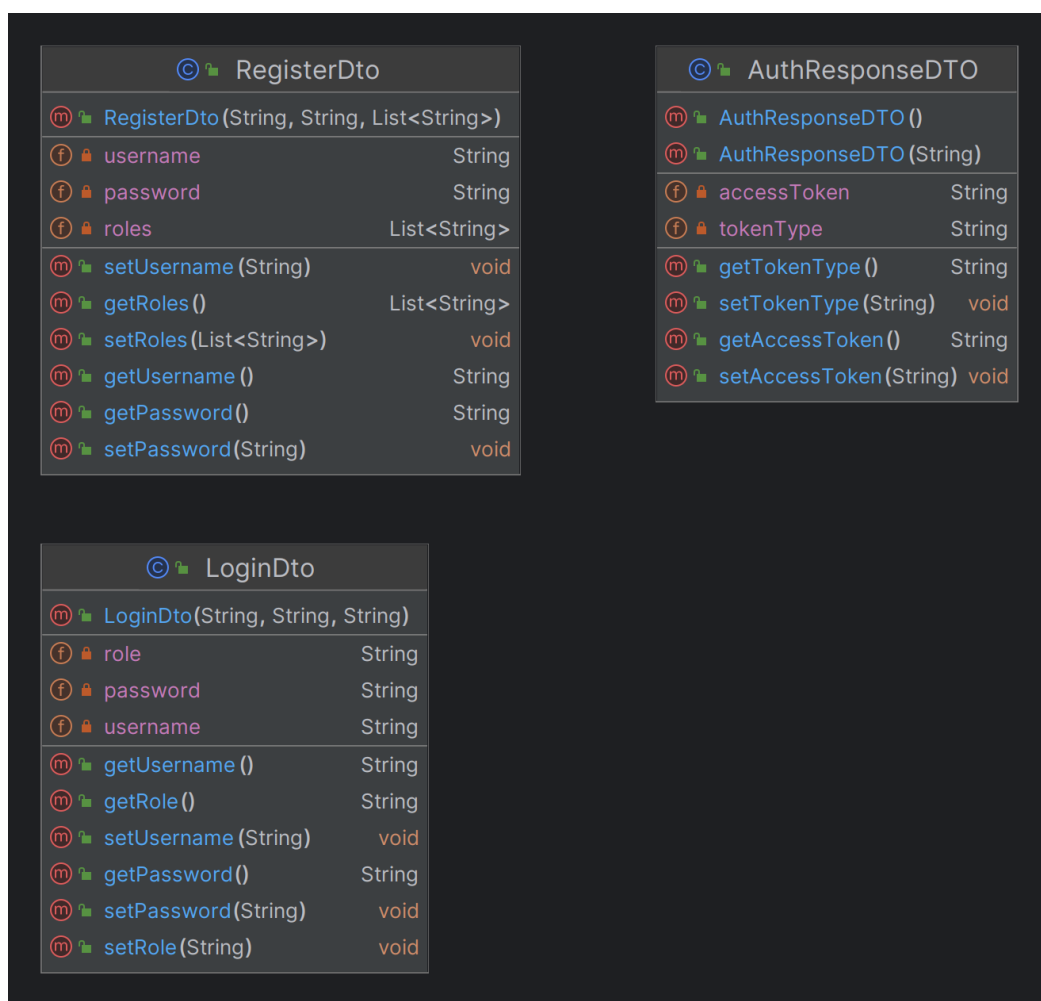


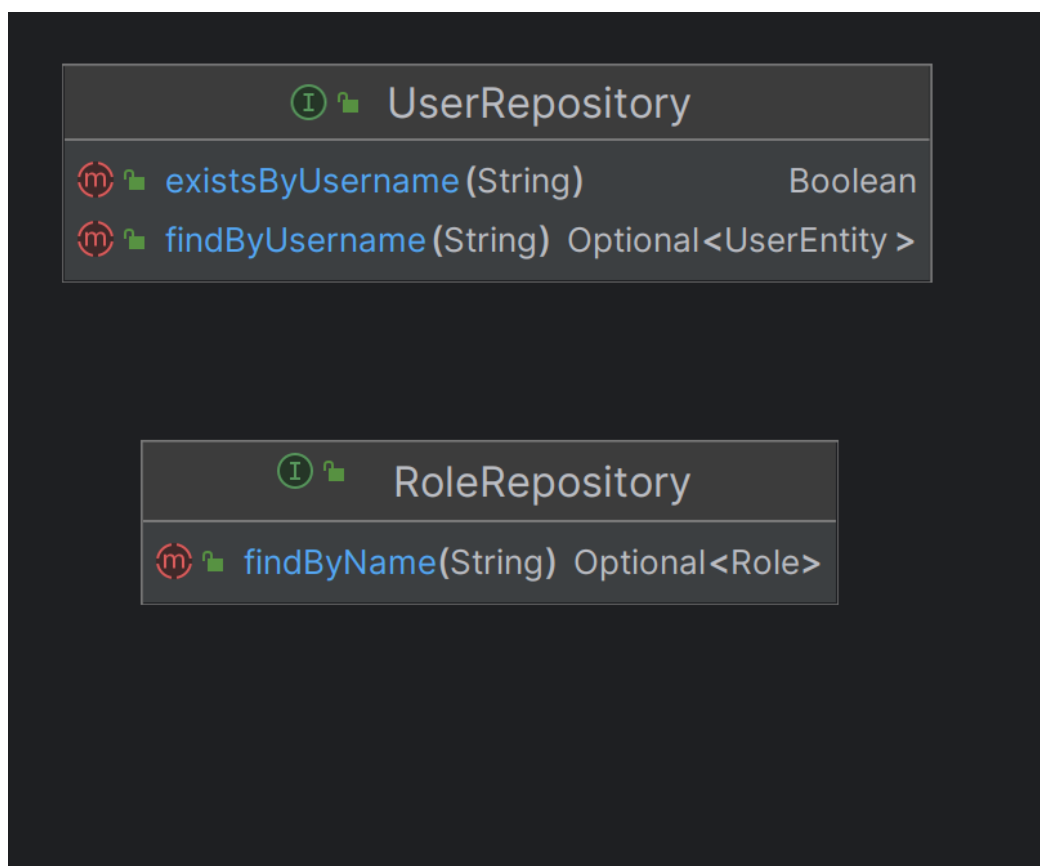
Slika 4.2: UML dijagram paketa *Controllers*

Slika 4.3 prikazuje paket DTO koji služi za pretvaranje JSON objekata koji stižu na određenu rutu i Java objekt i za pretvaranje Java objekata u JSON odgovore koje klijent razumije.

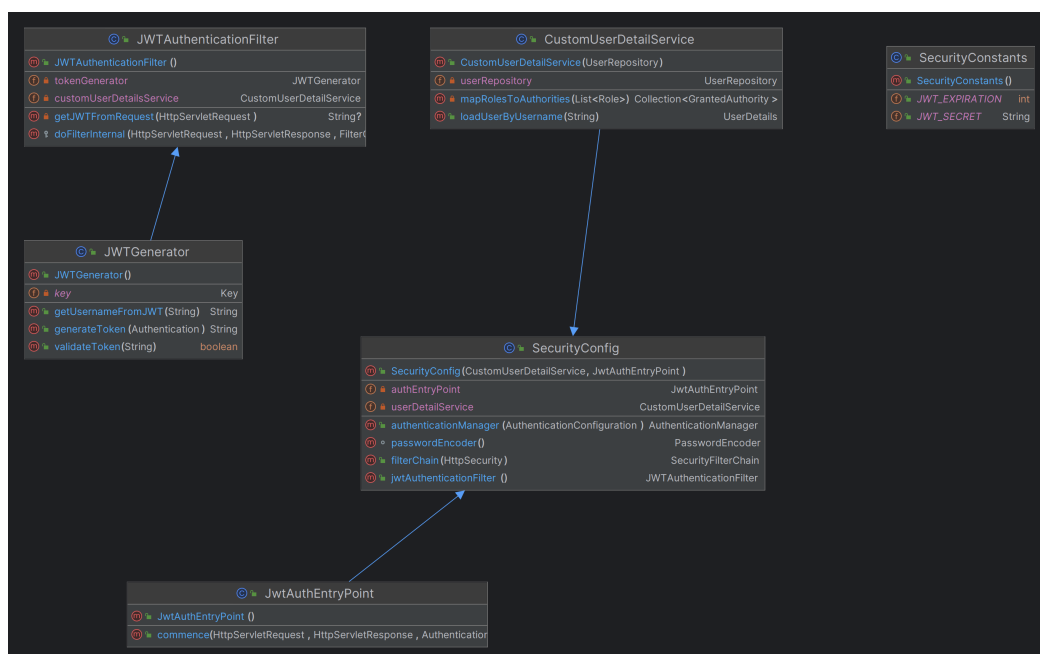
Slika 4.4 prikazuje paket *Repository* koji preko JPA-a (*Java Persistence API*) pristupa bazi podataka. Za stvaranje SQL upita koristi se *Spring Data* koji omogućava kreiranje metoda s posebnim imenima i iz njih stvara SQL upite. Primjerice *findByUsername()* *Spring Data* pretvara u SQL upit koji pretražuje tablicu *Users* za određeno korisničko ime.

Slika 4.5 prikazuje paket *Security* koji je zadužen za obradu svakog zahtjeva koji stiže na server i prosuditi ima li trenutni korisnik pravo pristupa. Glavna klasa za

Slika 4.3: UML dijagram paketa *DTO*

Slika 4.4: UML dijagram paketa *Repository*

to je klasa *SecurityConfig* koja preko metode *filterChain()* primjenjuje filtre na svaki zahtjev da odredi pravo pristupa. Također klasa *SecurityConfig* pomoću klasa *JWT-Generator*, *JWTAuthenticationFilter* i *JwtAuthEntryPoint* za svaku uspješnu prijavu generira JWT token koji se zatim u svim zahtjevima tog korisnika koristi za autentifikaciju tog korisnika.



Slika 4.5: UML dijagram paketa *Security*

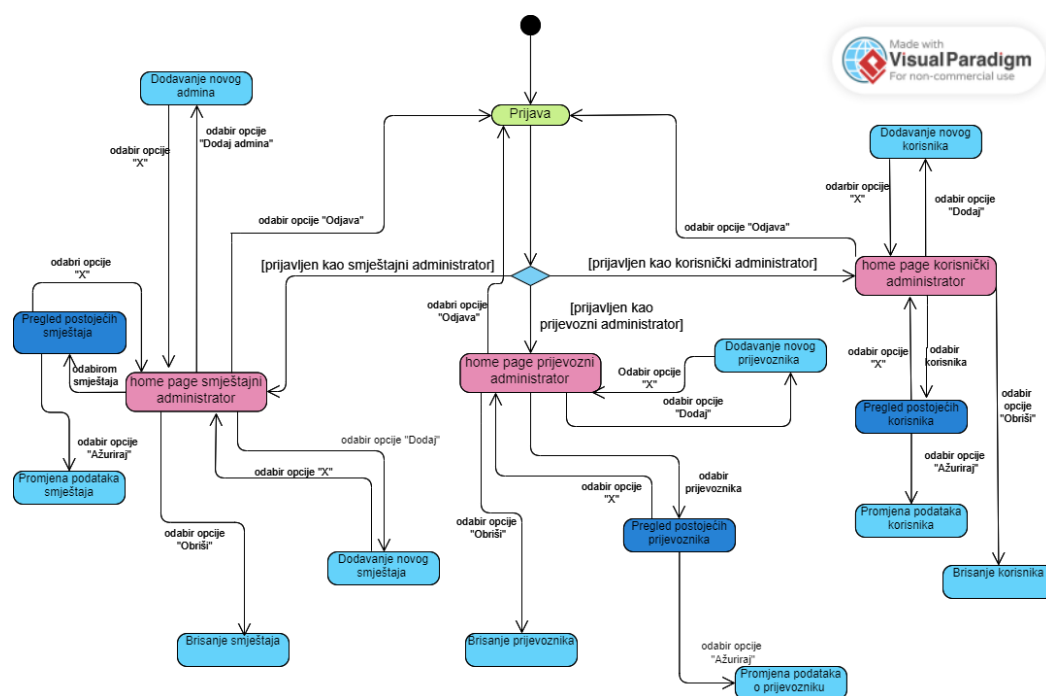
4.3 Dijagram stanja

Na slici 4.6 prikazan je dijagram stanja. Dijagram stanja prikazuje u kojem se stanju korisnik aplikacije može nalaziti. Početno stanje je prijava. Korisnik se može prijaviti kao jedan od administratora: Korisnički administrator, Smještajni administrator ili Prijevozni administrator. Ovisno o tome koji administrator se prijavi, odlazi na odgovarajuću početnu stranicu (homepage) gdje se nalaze opcije za korištenje ovisno o ulozi. Svi administratori mogu se vratiti na prijavu tako što se prethodno odjave, klikom "Odjava".

Ako je prijavljeni administrator smještajni administrator, ima opciju dodavanja novih administratora odabirom opcije "Dodaj admina". Također, ima mogućnost pregleda smještaja i ažuriranja istog pomoću opcije "Ažuriraj". Već postojeći smještaj administrator može obrisati opcijom "Obriši". Novi smještaj dodaje opcijom "Dodaj".

Ako je prijavljeni administrator prijevozni administrator, ima opciju pregleda već postojećih prijevoznika te ažuriranja istih opcijom "Ažuriraj". Također, prijevozni administrator može dodavati nove prijevoznike opcijom "Dodaj" i brisati već postojeće opcijom "Obriši".

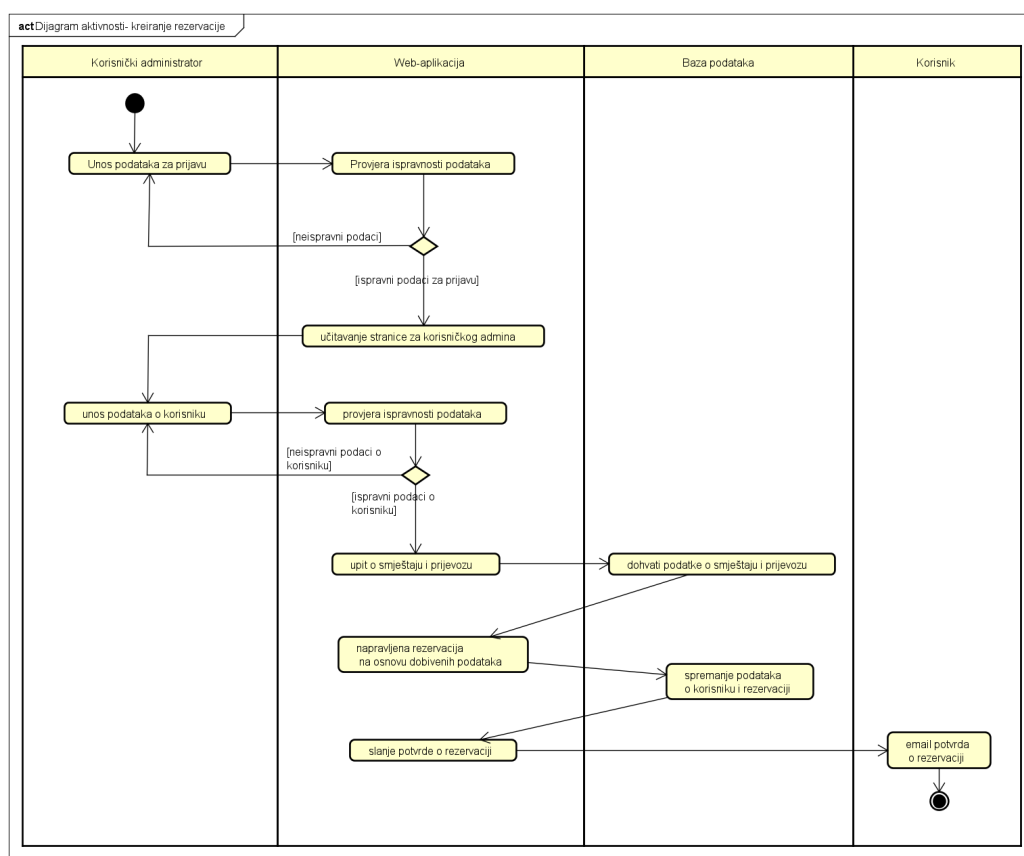
Ako je prijavljeni administrator korisnički administrator, ima opciju dodavanja novog korisnika (klijenta) opcijom "Dodaj". Također, odabirom korisnika mogu se pregledavati njegovi podaci te odabirom opcije "Ažuriraj" ažurirati. Korisnike se može brisati opcijom "Obriši".



Slika 4.6: Dijagram stanja

4.4 Dijagram aktivnosti

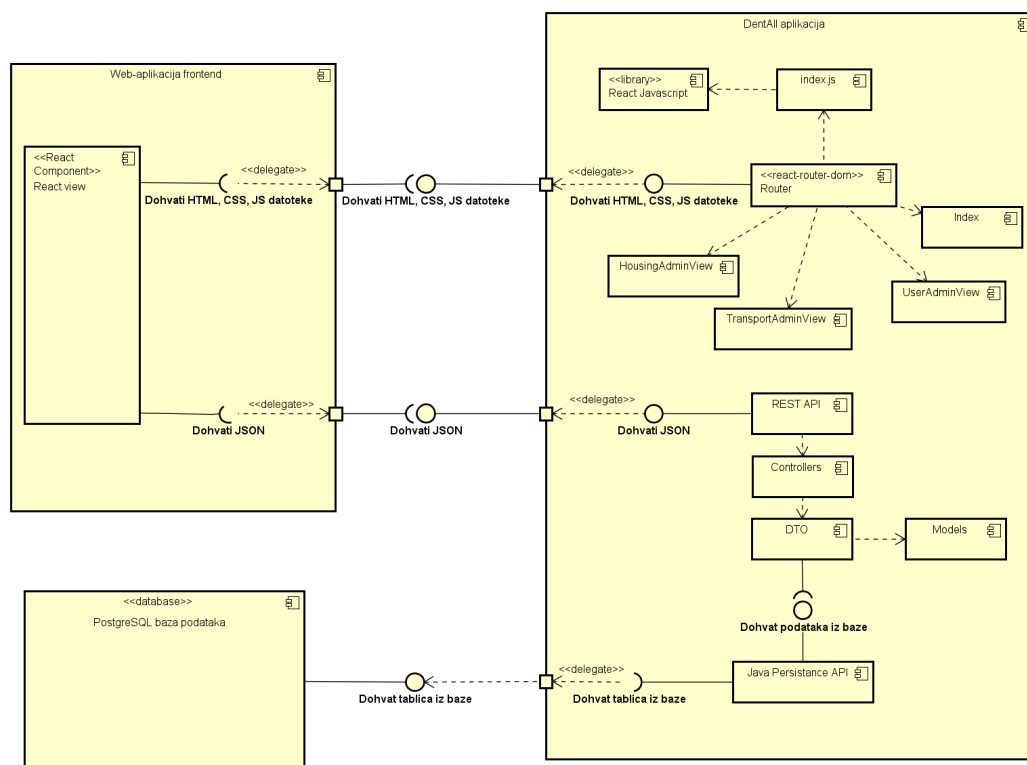
Na slici 4.7 prikazan je dijagram aktivnosti. Dijagram aktivnosti ilustrira detalje procesa kreiranja rezervacije za korisnika putem web-aplikacije. Korisnički administrator započinje postupak unosom ispravnih podataka za prijavu u web-aplikaciju. Ukoliko se uneseni podaci pokažu neispravnima, administrator će ponovno unijeti potrebne informacije kako bi uspješno pristupio aplikaciji. Nakon uspješne prijave administratoru se učitava stranica, a zatim korisnički administrator unosi podatke o korisniku za kojeg želi napraviti rezervaciju. Web-aplikacija provodi provjeru ispravnosti unesenih podataka, osiguravajući da su svi potrebni detalji upisani. Kada su svi podaci uneseni ispravno, aplikacija nastavlja s provjerom dostupnosti smještaja i prijevoza za odabranog korisnika. Na temelju rezultata provjere dostupnosti, web-aplikacija generira rezervaciju za korisnika. Baza podataka se ažurira označavajući odabrani smještaj i prijevoz kao zauzet u određenom razdoblju rezerviranom za korisnika. Nakon spremanja podataka, Web-aplikacija šalje e-mail potvrdu korisniku o potvrdi rezervacije.



Slika 4.7: Dijagram aktivnosti

4.5 Dijagram komponenti

Na slici 4.8 prikazan je dijagram komponenti. Za dohvaćanje HTML, CSS i JS datoteka na klijentsku stranu (frontend) koristi se prvo od dva sučelja. Ovisno o potrebnom prikazu (HousingAdminView, TransportAdminView, UserAdminView, Index), Router komponenta određuje koje se HTML, CSS i JS datoteke poslužuju na prvom sučelju. Sučelju za primanje JSON podataka pristupa se putem REST API komponenti. REST API pruža podatke koji pripadaju serverskoj strani aplikacije (backend). Java Persistence API omogućava dohvaćanje podataka iz baze generirajući SQL upite te upisivanje u bazu. Na serverskoj strani implementiramo kontrolere (Controllers) kako bismo prenijeli modele, pretvorene u DTO (Data Transfer Object), prema klijentskoj strani dijela aplikacije.



Slika 4.8: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija unutar tima ostvarena je korištenjem aplikacije Whatsapp¹, dok je za direktnu interakciju s asistentom grupe odabran Microsoft Teams². Za upravljanje verzijama koda i suradnju u razvoju, korišteni su Git³ i GitHub⁴ na kojemu se nalazi udaljeni repozitorij. U izradi UML dijagrama korišteni su alati Astah UML⁵ i Visual Paradigm Online⁶. Za oblikovanje dijagrama baze podataka korišten je alat DBDiagram⁷.

Stranica klijenta je razvijena korištenjem programskog jezika JavaScript⁸ i React⁹ biblioteke, a korišteno je razvojno okruženje Visual Studio Code¹⁰. Za server stranu korišteni su programski jezik Java¹¹ i radni okvir Spring Boot¹², a korišteno je razvojno okruženje IntelliJ IDEA¹³. Za puštanje aplikacije u pogon korištena je platforma Heroku¹⁴. Ispitivanje sustava provedeno je koristeći radni okvir Selenium¹⁵.

¹<https://www.whatsapp.com/>

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

³<https://git-scm.com/>

⁴<https://github.com/>

⁵<https://astah.net/products/astah-uml/>

⁶<https://online.visual-paradigm.com/>

⁷<https://dbdiagram.io/home>

⁸<https://www.javascript.com/>

⁹<https://react.dev/>

¹⁰<https://code.visualstudio.com/>

¹¹<https://www.java.com/en/>

¹²<https://spring.io/projects/spring-boot/>

¹³<https://www.jetbrains.com/idea/>

¹⁴<https://www.heroku.com/>

¹⁵<https://www.selenium.dev/>

5.2 Ispitivanje programskog rješenja

dio 2. revizije

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

5.2.1 Ispitivanje komponenti

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

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹⁶. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

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

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

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

¹⁶<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

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

5.4 Upute za puštanje u pogon

dio 2. revizije

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

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

6. Zaključak i budući rad

dio 2. revizije

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

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

Popis literature

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>

Indeks slika i dijagrama

3.1	Funkcijski zahtjevi smještajnog administratora	18
3.2	Funkcijski zahtjevi transportnog administratora	19
3.3	Funkcijski zahtjevi korisničkog administratora	19
3.4	Sekvencijski dijagram za UC4	20
3.5	Sekvencijski dijagram za UC6	21
3.6	Sekvencijski dijagram za UC13	22
4.1	Relacijska shema baze podataka	28
4.2	UML dijagram paketa <i>Controllers</i>	29
4.3	UML dijagram paketa <i>DTO</i>	30
4.4	UML dijagram paketa <i>Repository</i>	31
4.5	UML dijagram paketa <i>Security</i>	32
4.6	Dijagram stanja	34
4.7	Dijagram aktivnosti	35
4.8	Dijagram komponenti	36

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.

1. sastanak

- Datum: 16.10.2023
- Prisustvovali: Vedran Lončar, Jan Kozina, Filip Stilinović, Lorena Jakić, Viktorija Štrulić-Tupek, Stela Dermit, Danko Delimar
- Teme sastanka:
 - upoznavanje članova tima
 - upoznavanje s temom
 - okvirno definiranje baze podataka

2. sastanak

- Datum: 23.10.2023
- Prisustvovali: Vedran Lončar, Jan Kozina, Filip Stilinović, Lorena Jakić, Viktorija Štrulić-Tupek, Stela Dermit, Danko Delimar
- Teme sastanka:
 - završena baza podataka
 - podjela uloga na poslužiteljsku i klijentsku stranu

3. sastanak

- Datum: 30.10.2023
- Prisustvovali: Vedran Lončar, Jan Kozina, Filip Stilinović, Lorena Jakić, Viktorija Štrulić-Tupek, Stela Dermit, Danko Delimar
- Teme sastanka:
 - podjela zadataka unutar klijentske i poslužiteljske strane
 - početak izrade generičke funkcionalnosti klijentske i poslužiteljske strane

4. sastanak

- Datum: 6.11.2023
- Prisustvovali: Vedran Lončar, Jan Kozina, Filip Stilinović, Lorena Jakić, Viktorija Štrulić-Tupek, Stela Dermit, Danko Delimar
- Teme sastanka:
 - završena klijentska strana
 - završena autentifikacija
 - završeno modeliranje baze podataka unutar spring boota

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Danko Delimar	Stela Dermić	Lorena Jakić	Jan Kozina	Vedran Lončar	Filip Stilić	Viktorija Štrulić-Tupek
Upravljanje projektom	30	2					
Opis projektnog zadatka	2	1					
Funkcionalni zahtjevi	3						
Opis pojedinih obrazaca	4						
Dijagram obrazaca	1						
Sekvencijski dijagrami	1						
Opis ostalih zahtjeva	1						
Arhitektura i dizajn sustava	2						
Baza podataka			16				
Dijagram razreda	1						
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati				4	8	8	
Ispitivanje programskog rješenja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Danko Delimar	Stela Dermitt	Lorena Jakić	Jan Kozina	Vedran Lončar	Filip Stilinović	Viktorija Štrulić-Tupek
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>					8	8	
<i>izrada baze podataka</i>			16				
<i>spajanje s bazom podataka</i>		8					
<i>back end</i>		12		2			

Dijagrami pregleda promjena

dio 2. revizije

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