

Programsko inženjerstvo

Ak. god. 2023./2024.

Medicinska rehabilitacija

Dokumentacija, Rev. 2

Grupa: *Seven Smurfs*

Voditelj: *Dora Kašik*

Datum predaje: 19. 1. 2024.

Nastavnik: *Miljenko Krhen*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka „Medicinska rehabilitacija“	6
2.1 Uvod	6
2.2 Izgled stranica aplikacije	6
2.2.1 Početna stranica	6
2.2.2 Stranica za prijavu	6
2.2.3 Zaboravljena lozinka	7
2.2.4 Registracija	7
2.2.5 Pomoć	7
2.2.6 Kontakt	7
2.2.7 Početna stranica Admina	7
2.2.8 Početna stranica Korisnika	7
2.2.9 Naručivanje na termin	7
2.2.10 Početna stranica Djelatnika	8
2.2.11 Dodjela termina	8
2.2.12 Stranica o terapiji	8
2.2.13 Korisničke postavke	8
2.3 Dodatno o aplikaciji	8
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	21
3.2 Ostali zahtjevi	25
4 Arhitektura i dizajn sustava	27
4.1 Baza podataka	29
4.1.1 Opis tablica	30
4.1.2 Dijagram baze podataka	37
4.2 Dijagram razreda	38

4.3 Dijagram stanja	46
4.4 Dijagram aktivnosti	48
4.5 Dijagram komponenti	50
5 Implementacija i korisničko sučelje	51
5.1 Korištene tehnologije i alati	51
5.2 Ispitivanje programskog rješenja	52
5.3 Dijagram razmještaja	58
5.4 Upute za puštanje u pogon	59
6 Zaključak i budući rad	65
Popis literature	66
Indeks slika i dijagrama	68
Dodatak: Prikaz aktivnosti grupe	69

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatak	Autori	Datum
1.1	Napravljen predložak	Marko Miletić	17.10.2023.
1.2	Dodani opis projektnog zadatka i funkcionalni zahtjevi	Marko Miletić	21.10.2023.
1.3	Kreiran dijagram arhitekture	Josip Begić	22.10.2023.
1.4	Početak rada na opisu baze podataka	Marko Miletić	27.10.2023.
1.5	Dodani opisi tablica baze podataka	Marko Miletić	02.11.2023.
1.6	Ažuriran dnevnik sastajanja	Marko Miletić	09.11.2023.
1.7	Ažurirana tablica aktivnosti	Marko Miletić	09.11.2023.
1.8	Kreiran dijagram baze podataka	Josip Begić	10.11.2023.
1.9	Kreiran dijagram komponenti	Josip Begić	10.11.2023.
1.10	Kreirani dijagrami razreda	Anton Vivedova	13.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.11	Dodani funkcionalni i ostali zahtjevi	Dora Kašik, Katarina Mikulić	14.11.2023.
1.12	Kreirani sekvencijski dijagrami i dijagrami obrazaca uporabe	Dora Kašik, Katarina Mikulić	14.11.2023.
1.13	Dodani svi dijagrami i opisi	Marko Miletić	14.11.2023.
1.14	Napisan opis arhitekture	Roko Krstičević	17.11.2023.
1.15	Ispravljene izmjene u bazi podataka i u izgledu stranice	Marko Miletić	17.11.2023.
2.1	Dodan dijagram stanja	Marko Miletić	03.01.2024.
2.2	Ažuriran dnevnik sastajanja	Marko Miletić	04.01.2024.
2.3	Dodan opis dijagrama komponenti	Marko Miletić	04.01.2024.
2.4	Napisan odlomak "Korištene tehnologije i alati"	Marko Miletić	08.01.2024.
2.5	Dodan dijagram razmještaja i njegov opis	Marko Miletić	08.01.2024.
2.6	Napisan zaključak	Marko Miletić	09.01.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
2.7	Napisano poglavlje o testiranju	Marko Miletić	17.01.2024.
2.8	Dodani dijagrami promjena	Marko Miletić	18.01.2024.
2.9	Dodani dijagrami stanja	Anton Vi-voda	19.01.2024.
2.10	Dodane upute za puštanje u pogon	Katarina Mikulić	19.01.2024.
2.11	Dodan dijagram aktivnosti	Katarina Mikulić	19.01.2024.
2.12	Dodani dijagrami razreda	Anton Vi-voda	19.01.2024.

2. Opis projektnog zadatka „Medicinska rehabilitacija“

2.1 Uvod

Cilj ovog projektnog zadatka je razviti programsku podršku za izradu web aplikacije „ReHub“ koja će olakšati prijavu i praćenje poboljšanja zdravstvenog stanja ljudi/bolesnika koji su imali lakše ili teže povrede nakon kojih je potrebno provesti fizikalnu terapiju i medicinsku rehabilitaciju. Razvijena kako bi omogućila transparentan, učinkovit, i siguran način za bolesnike da prate svoj put prema oporavku, „ReHub“ također pomaže zdravstvenim ustanovama bolje organizirati i pratiti rehabilitacijske procese.

2.2 Izgled stranica aplikacije

U nastavku se opisuju stranice web aplikacije te njihove funkcionalnosti.

2.2.1 Početna stranica

Na početnoj stranici korisnik može odabrat jednu od tri moguće opcije kako želi nastaviti rad u aplikaciji ili potražiti detaljnije informacije:

- Prijava
- Registracija
- Korisnik

2.2.2 Stranica za prijavu

Od korisnika se za uspješan nastavak rada u aplikaciji traži da unese valjano korisničko ime te ispravnu lozinku. Postoji i opcija „Zaboravljena lozinka“ kojom se korisniku na e-mail šalju upute za postavljanje nove lozinke u slučaju da je korisnik zaboravio lozinku.

2.2.3 Zaboravljena lozinka

Na ovoj stranici se od korisnika očekuje da unese e-mail adresu na koju želi dobiti daljnje upute o ponovnom postavljanju lozinke.

2.2.4 Registracija

Stranica sadrži polja u koje korisnik upisuje ime, prezime, e-mail adresu, broj telefona, OIB, lozinku, ponovno lozinku, MBO, datum rođenja te spol (opcionalno) ukoliko se želi uspješno registrirati.

2.2.5 Pomoć

Ova stranica sadrži odgovore na najčešće postavljena pitanja te sadrži detaljne upute kako se ispravno registrirati i prijaviti u aplikaciju.

2.2.6 Kontakt

Odabirom ove stranice korisnik može vidjeti e-mail adresu lokaciju te radno vrijeme organizacije.

2.2.7 Početna stranica Admina

Na početnoj stranici Admina mogu se vidjeti svi djelatnici i pacijenti u klinici. Gornji dio stranice sadrži opciju odjave te opciju biranja postavki.

2.2.8 Početna stranica Korisnika

Ovdje korisnik može vidjeti popis svojih termina te ima opciju kojom se naručuje na novi termin. Na vrhu stranice postoje opcije „Kontakt“, „Moj profil“ i „Odjava“.

2.2.9 Naručivanje na termin

Korisnik bira jednu od opcija „Prvi dolazak“ ili „Ponovna terapija“. Uz to navodi vrstu oboljenja, opis oboljenja, izdavatelja uputnice postupak liječenja i referencu na obavljenu terapiju.

2.2.10 Početna stranica Djelatnika

Djelatnik na svojoj početnoj stranici vidi popis prijava na termine, popis dostupnih i nedostupnih djelatnika, opciju za dodjeljivanje termina, popis slobodnih i nedostupnih soba i ordinacija te popis opreme.

2.2.11 Dodjela termina

Na ovoj stranici djelatnik ima pristup informacijama koje je korisnik naveo pri naručivanju termina. Djelatnik odabire ordinaciju, termin i djelatnika koji će raditi na rehabilitaciji.

2.2.12 Stranica o terapiji

Na ovoj stranici unose se podaci vezani uz terapiju.

2.2.13 Korisničke postavke

Na ovoj stranici korisnik ima pregled podataka i opcije postavki.

2.3 Dodatno o aplikaciji

Zamišljeno je da aplikacija bude što je više moguće lakša i intuitivnija za upotrebu, pošto računamo da bi pripadnici svih dobnih skupina mogli i htjeli koristiti ovu aplikaciju. U funkcionalnosti aplikacije vidimo veliku korist jer bi se njenom uporabom klijentima znatno olakšalo i ubrzalo naručivanje na medicinsku rehabilitaciju, a djelatnicima klinike bi se olakšala evidencija naručenih klijenata. Klinike bi također mogle doživjeti povećani interes klijenata zainteresiranih za njihove usluge. Uporabom aplikacije klijenti bi izbjegli dugotrajna čekanja u redovima bolnica i klinika. Aplikacija osigurava sigurnost i zaštitu osjetljivih zdravstvenih podataka te poboljšava komunikaciju između bolesnika, zdravstvenih ustanova i liječnika. Smatramo kako je ključno pratiti i analizirati statistiku rehabilitacije radi kontinuiranog poboljšanja procesa. Aplikacija podržava tamnu temu.

Ključne Funkcionalnosti

- **Registracija bolesnika:**

Bolesnici se registriraju putem aplikacije, pružajući svoje osobne podatke. Administrator ustanove provjerava i verificira podatke za osiguranje točnosti informacija.

- **Prijavljivanje za rehabilitaciju:**

Bolesnici unose detalje o svojim oboljenjima i zahtjevima za liječenjem. Liječnici su verificirani putem imenika liječnika.

- **Raspoređivanje termina:**

Djelatnici ustanove pregledavaju i odobravaju prijave bolesnika. Na temelju kapaciteta opreme, prostora i osoblja, dodjeljuju termine za rehabilitaciju.

- **Komunikacija s bolesnicima:**

Bolesnicima se automatski šalju informacije o njihovim terminima putem elektroničke pošte. Mogućnost direktnog kontakta putem elektroničke pošte za obavještavanje o promjenama.

- **Sigurnost podataka:**

Visoka razina sigurnosti osigurava zaštitu osobnih podataka bolesnika i drugih osjetljivih informacija.

- **Pregled i upisivanje rezultata:**

Djelatnici ustanove imaju pristup svim podacima o bolesnicima i njihovim tretmanima. Nakon svakog rehabilitacijskog zahvata, upisuju postignute rezultate.

- **Administrator:**

Sistemski administrator ima potpunu kontrolu nad aplikacijom. Može definirati postavke, pristupne razine i konfiguracije sustava.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Pacijent
2. Zaposlenik (medicinsko osoblje)
3. Administrator
4. Razvojni tim:
 - Dora Kašik (voditelj)
 - Josip Begić
 - Erik Greblo
 - Anton Vivoda
 - Roko Krstičević
 - Katarina Mikulić
 - Marko Miletić

Aktori i njihovi funkcionalni zahtjevi:

1. Pacijent (inicijator) može:
 - (a) Predati zahtjev za terapijom i dodjelom termina iste
 - i. Ako dolazi na ponovljenu terapiju, u zahtjevu prilaže referencu na prethodno obavljenu terapiju
 - (b) Izvršiti registraciju u sustav
2. Administrator (inicijator) može:
 - (a) Upravljati podacima o pacijentima i djelatnicima
 - (b) Upravljati podacima o dostupnoj opremi i prostorijama klinike
 - (c) Vršiti postupke za zaštitu podataka
 - (d) Dodavati nove djelatnike
3. Superadministrator (inicijator) može:

- (a) Dodijeliti djelatniku administratorskih ovlasti
- (b) Ukloniti administratorske ovlasti djelatnika
- (c) Verificirati registraciju korisnika
- (d) Upravljati podacima o pacijentima i djelatnicima
- (e) Upravljati podacima o dostupnoj opremi i prostorijama klinike
- (f) Vršiti postupke za zaštitu podataka
- (g) Dodavati nove djelatnike

4. Zaposlenik (inicijator) može:

- (a) Dodijeliti termin pacijentu
 - i. Nakon dodjele termina šalje pacijentu potrebne podatke o terminu putem elektroničke pošte
- (b) Pristupiti pregledu aktivnih prijava pacijenata
- (c) Pristupiti podatku o dostupnim djelatnicima
- (d) Pristupiti podatku o trajanju pojedinog zahvata
- (e) Pristupiti podacima o dostupnoj opremi i prostorijama klinike
- (f) Registrirati dolazak pacijenta na terapiju i rezultate nakon obavljene terapije
- (g) Prihvati ili odbiti zahtjev pacijenta za terminom terapije

5. Baza podataka (sudionik) može:

- (a) Pohranjuje podatke o korisnicima i njihovim ovlastima
- (b) Pohranjuje podatke o terapijama
- (c) Pohranjuje podatke o čestim pitanjima
- (d) Pohranjuje podatke o prostorijama i opremi

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija pacijenta

- **Glavni sudionik:** Pacijent
- **Cilj:** Registracija novog pacijenta u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Bolesnik nema prethodno registriran račun
- **Opis osnovnog tijeka:**
 1. Pacijent unosi osobne podatke, adresu elektroničke pošte i lozinku.
 2. Sustav provjerava ispravnost podataka.
 3. Sustav pohranjuje podatke i stvara korisnički račun.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava korisnika o potrebnim ispravkama.

UC2 - Prijava

- **Glavni sudionik:** Pacijent
- **Cilj:** Pristup stranici rehabilitacije nakon prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Pacijent je registriran i prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Pacijent šalje zahtjev za prijavom na sustav s korisničkim imenom i lozinkom.
 2. Sustav provjerava ispravnost podataka.
 3. Ako su podaci ispravni, pacijent je prijavljen u sustav.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava korisnika o potrebnim ispravkama. Ako korisnik nije registriran, sustav obavještava korisnika da nema važeći račun.

UC3 - Zahtjev za resetiranje lozinke

- **Glavni sudionik:** Bilo koji korisnik
- **Cilj:** Prijava u sustav kako bi pristupio stranici rehabilitacije
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima registriran račun u sustavu
- **Opis osnovnog tijeka:**

1. Pacijent šalje zahtjev za poveznicu za resetiranje lozinke
 2. Sustav šalje poveznicu za resetiranje lozinke na korisnikovu adresu elektroničke pošte.
- **Opis mogućih odstupanja:** Ako se ne može pronaći korisnik s unesenom adresom e-pošte, sustav obavještava korisnika o neuspjelom zahtjevu.

UC4 - Spremanje nove lozinke

- **Glavni sudionik:** Bilo koji korisnik
- **Cilj:** Spremiti novu lozinku nakon poveznice za resetiranje lozinke
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik je primio poveznicu za resetiranje lozinke i želi postaviti novu lozinku
- **Opis osnovnog tijeka:**
 1. Pacijent šalje novu lozinku putem poveznice za resetiranje lozinke.
 2. Sustav sprema novu lozinku za korisnički račun.
- **Opis mogućih odstupanja:** Ako poveznica za resetiranje lozinke nije ispravna ili je istekla, sustav obavještava korisnika o neuspjelom ažuriranju lozinke.

UC5 - Dodavanje zaposlenika u bazu

- **Glavni sudionik:** Superadministrator
- **Cilj:** Registrirati novog zaposlenika u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima superadministratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za registraciju zaposlenika s podacima o zaposleniku.
 2. Sustav provjerava ispravnost podataka.
 3. Sustav registrira zaposlenika u sustavu.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava superadministratore o potrebnim ispravkama.

UC6 - Brisanje zaposlenika

- **Glavni sudionik:** Superadministrator
- **Cilj:** Deaktivirati račun određenog zaposlenika u sustav
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima superadministratorske ovlasti i prijavljen je u sustav

- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za brisanje zaposlenika s određenim identifikatorom.
 2. Sustav označava zaposlenika kao neaktivnog u bazi podataka.

UC8 - Prikaz svih učestalih pitanja (FAQ-a)

- **Glavni sudionik:** Bilo koji korisnik
- **Cilj:** Prikazati popis svih FAQ-ova u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Opis osnovnog tijeka:**
 1. Korisnik šalje zahtjev za dohvaćanje svih FAQ-ova.
 2. Sustav dohvaća popis FAQ-ova iz baze podataka.
 3. Sustav vraća odgovor s popisom FAQ-ova.

UC10 - Kreiranje sobe

- **Glavni sudionik:** Superadministrator
- **Cilj:** Kreirati novu sobu u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima superadministratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za kreiranje nove sobe s potrebnim podacima.
 2. Sustav provjerava ispravnost podataka.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava uperadministratora o potrebnim ispravkama.

UC11 - Pregled svih soba

- **Glavni sudionik:** Zaposlenik, Administrator i Superadministrator
- **Cilj:** Prikazati sve sobe u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima administratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Zaposlenik šalje zahtjev za dohvaćanje svih soba.
 2. Sustav dohvaća sve sobe iz baze podataka.
 3. Sustav vraća odgovor s popisom soba.

UC13 - Brisanje sobe

- **Glavni sudionik:** Superadministrator
- **Cilj:** Obrisati sobu iz sustava
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima superadministratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za brisanje sobe s određenim identifikacijskim brojem.
 2. Sustav provjerava postoji li soba s navedenim identifikacijskim brojem.
 3. Ako postoji, sustav briše sobu.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava administratora o potrebnim ispravkama.

UC14 - Kreiranje opreme

- **Glavni sudionik:** Superadministrator
- **Cilj:** Kreirati novu opremu u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima superadministratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za kreiranje nove opreme s potrebnim podacima.
 2. Sustav provjerava ispravnost podataka.
 3. Sustav kreira novu opremu u sustav.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava superadministratora o potrebnim ispravkama.

UC15 - Pregled svih oprema

- **Glavni sudionik:** Zaposlenik, Administrator i Superadministrator
- **Cilj:** Prikazati sve sobe u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima administratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Zaposlenik šalje zahtjev za dohvaćanje svih opreme.
 2. Sustav dohvaća svu opremu iz baze podataka.
 3. Sustav vraća odgovor s popisom opreme.

UC17 - Brisanje opreme

- **Glavni sudionik:** Superadministrator
- **Cilj:** Obrisati opremu iz sustava
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima superadministratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za brisanje opreme s određenim identifikacijskim brojem.
 2. Sustav sustav provjerava postoji li oprema s navedenim identifikacijskim brojem.
 3. Ako postoji, sustav briše opremu.

UC18 - Prikaz svih pacijenata

- **Glavni sudionik:** Superadministrator
- **Cilj:** Prikazati popis svih pacijenata u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima administratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Administrator šalje zahtjev za dohvaćanje svih pacijenata.
 2. Sustav dohvača popis pacijenata iz baze podataka.
 3. Sustav vraća odgovor s popisom pacijenata.

UC19 - Deaktivacija pacijentovog računa

- **Glavni sudionik:** Pacijent
- **Cilj:** Deaktivirati račun određenog pacijenta u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik šalje zahtjev za deaktivaciju pacijentovog računa s određenim identifikatorom.
 2. Sustav označava pacijenta kao neaktivnog u bazi podataka.

UC20 - Kreiranje terapije za pacijenta

- **Glavni sudionik:** Pacijent, Zaposlenik
- **Cilj:** Dodati novu terapiju za određenog pacijenta

- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik i zaposlenik prijavljeni u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik šalje zahtjev za dodavanje nove terapije s podacima o terapiji.
 2. Sustav provjerava ispravnost unesenih podataka.
 3. Zaposlenik unosi dodatne informacije o terapiji poput datuma, vremena i sobe.
 4. Sustav provjerava valjanost unesenih podataka, uključujući dostupnost sobe i preklapanje termina.
 5. Ako su podaci valjani, terapija se odobrava zaposleniku i potvrđuje pacijentu.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava korisnika o potrebnim ispravkama.

UC21 - Prikaz svih terapija za pacijenta

- **Glavni sudionik:** Pacijent
- **Cilj:** Prikazati popis svih terapija za određenog pacijenta
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik šalje zahtjev za dohvaćanje svih terapija za određenog pacijenta.
 2. Sustav dohvaća popis terapija iz baze podataka za određenog pacijenta.
 3. Sustav vraća odgovor s popisom terapija.

UC22 - Unos rezultata terapije

- **Glavni sudionik:** Zaposlenik i Administrator
- **Cilj:** Unijeti rezultate terapije u sustav
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima ovlasti zaposlenika i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Zaposlenik odabire pacijenta iz dostupne liste.
 2. Sustav provjerava bazu podataka kako bi dohvatio relevantne podatke o pacijentu.
 3. U slučaju uspješnog pronađenja pacijenta, zaposlenik šalje zahtjev za unos rezultata terapije, pružajući relevantne podatke o rezultatima.

4. Sustav provjerava ispravnost unesenih podataka.
 5. Ako su podaci ispravni, sustav zabilježava rezultate terapije u bazu podataka.
 6. Nakon uspješnog zapisivanja, sustav prikazuje na zaslonu potvrdu o uspješno unesenom zapisu, pružajući zaposleniku vizualnu potvrdu.
 7. Pacijent ima mogućnost pregleda svojih rezultata terapije unutar web-aplikacije.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni, sustav obavještava zaposlenika o potrebnim ispravkama.

UC23 - Dodjela administratorskih ovlasti

- **Glavni sudionik:** Superadministrator
- **Cilj:** Dodjela administratorskih ovlasti radi lakšeg upravljanja sustavom
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima ovlasti superadministratora i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Superadministrator šalje zahtjev za dodjelu administratorskih ovlasti određenom korisniku.
 2. Sustav provjerava ispravnost zahtjeva i potvrđuje administratorske ovlasti korisniku.
 3. Sustav ažurira informacije o ovlastima korisnika u bazi podataka.
- **Opis mogućih odstupanja:** Ako podaci nisu ispravni ili korisnik već ima administratorske ovlasti, sustav obavještava superadministratora o potrebnim ispravkama ili prekoračenju ovlasti.

UC24 - Uklanjanje administratorskih ovlasti

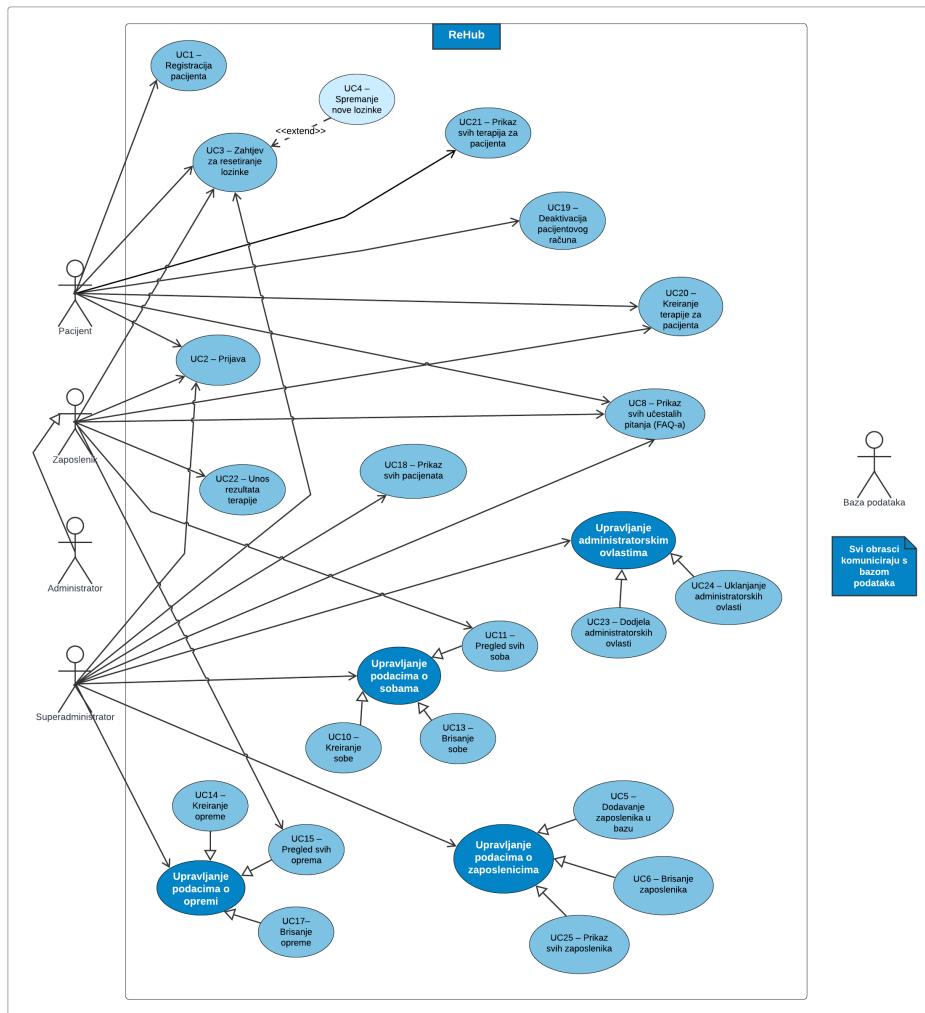
- **Glavni sudionik:** Superadministrator
- **Cilj:** Uklanjanje administratorskih ovlasti korisniku u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima ovlasti superadministratora i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Administrator šalje zahtjev za uklanjanje administratorskih ovlasti određenom korisniku.
 2. Sustav provjerava ispravnost zahtjeva i uklanja administratorske ovlasti korisniku.
 3. Sustav ažurira informacije o ovlastima korisnika u bazi podataka.

- **Opis mogućih odstupanja:** Ako podaci nisu ispravni ili korisnik nema administratorske ovlasti, sustav obavještava administratora o potrebnim ispravkama ili nedostatku ovlasti za uklanjanje.

UC25 - Prikaz svih zaposlenika

- **Glavni sudionik:** Superadministrator
- **Cilj:** Prikazati popis svih zaposlenika u sustavu
- **Sudionici:** Baza podataka/Sustav
- **Preduvjet:** Korisnik ima administratorske ovlasti i prijavljen je u sustav
- **Opis osnovnog tijeka:**
 1. Administrator šalje zahtjev za dohvaćanje svih djelatnika.
 2. Sustav dohvaća popis zaposlenika iz baze podataka.
 3. Sustav vraća odgovor s popisom zaposlenika.

Dijagrami obrazaca uporabe



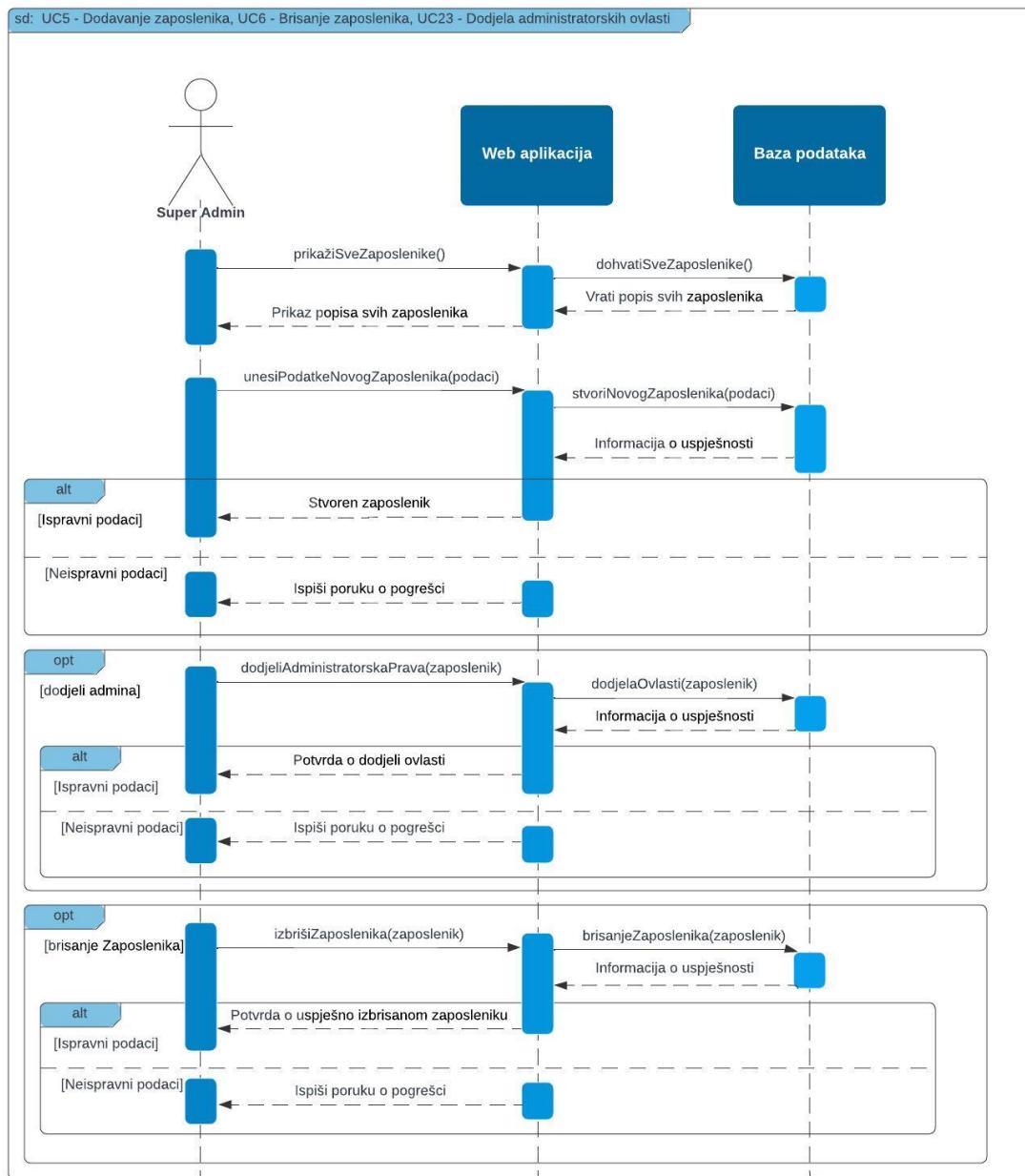
Slika 3.1: Dijagram obrazaca uporabe

3.1.2 Sekvencijski dijagrami

U sljedećem dijelu prikazani su sekvencijski dijagrami i njihovi opisi.

Obrazac uporabe UC5 – Dodavanje zaposlenika, UC6 - Brisanje zaposlenika i UC23 – Dodjela administratorskih ovlasti

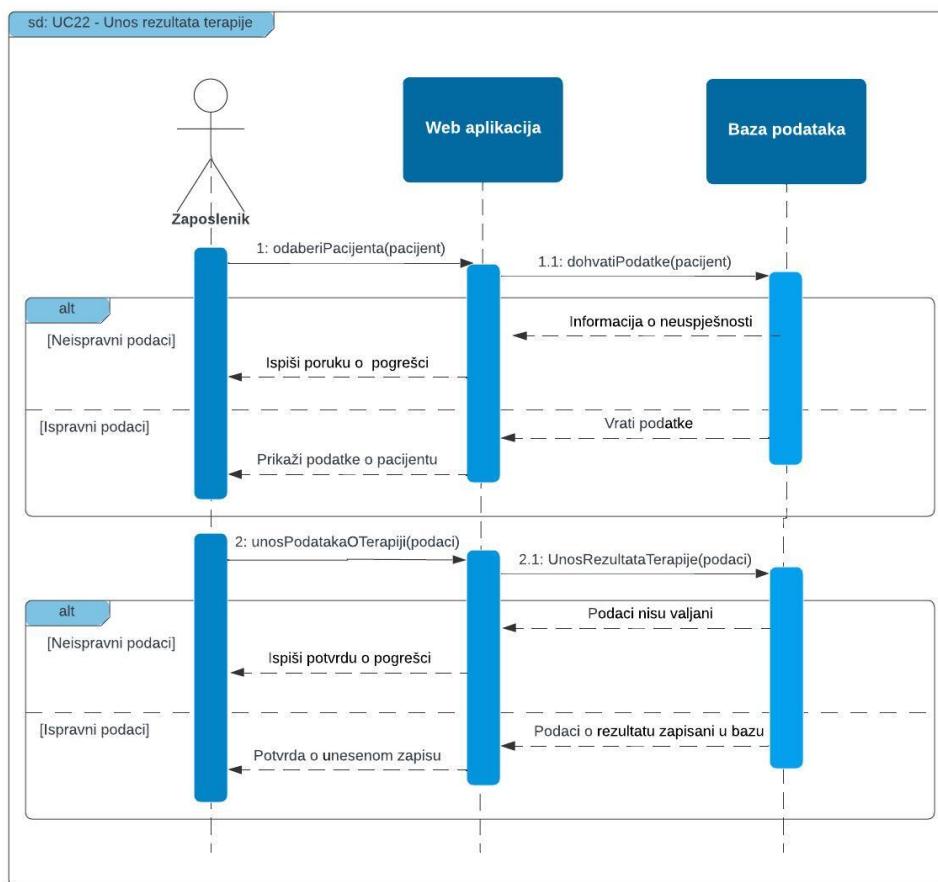
Kada korisnik odabere opciju "Dodavanje zaposlenika", web-aplikacija omogućuje korisniku unos svih relevantnih podataka o novom zaposleniku. Nakon završetka unosa, aplikacija automatski ažurira bazu podataka s novim informacijama. U slučaju odabira opcije "Brisanje zaposlenika", web-aplikacija pristupa bazi podataka kako bi prikazala popis trenutno zaposlenih osoba. Korisnik ima mogućnost odabira zaposlenika koje želi izbrisati, a nakon potvrde brisanja, aplikacija ažurira bazu podataka u skladu s korisnikovim zahtjevom. Kada korisnik odabere opciju "Dodjela administratorskih ovlasti", web-aplikacija pruža korisniku mogućnost odabira zaposlenika kojima želi dodijeliti administratorske ovlasti. Nakon što korisnik izabere zaposlenike, aplikacija ažurira bazu podataka s promijenjenim ovlastima za odabrane osobe. U situaciji kada baza podataka ne sadrži informacije o zaposlenicima u skladu s odabranim opcijama (primjerice, nema zaposlenika za brisanje ili dodjelu administratorskih ovlasti), web-aplikacija će prikazati odgovarajuću poruku, obavještavajući korisnika o nedostatku dostupnih podataka za izvršenje odabrane radnje.



Slika 3.2: Sekvencijski dijagram za UC5/6/23

Obrazac uporabe UC22 - Unos rezultata terapije

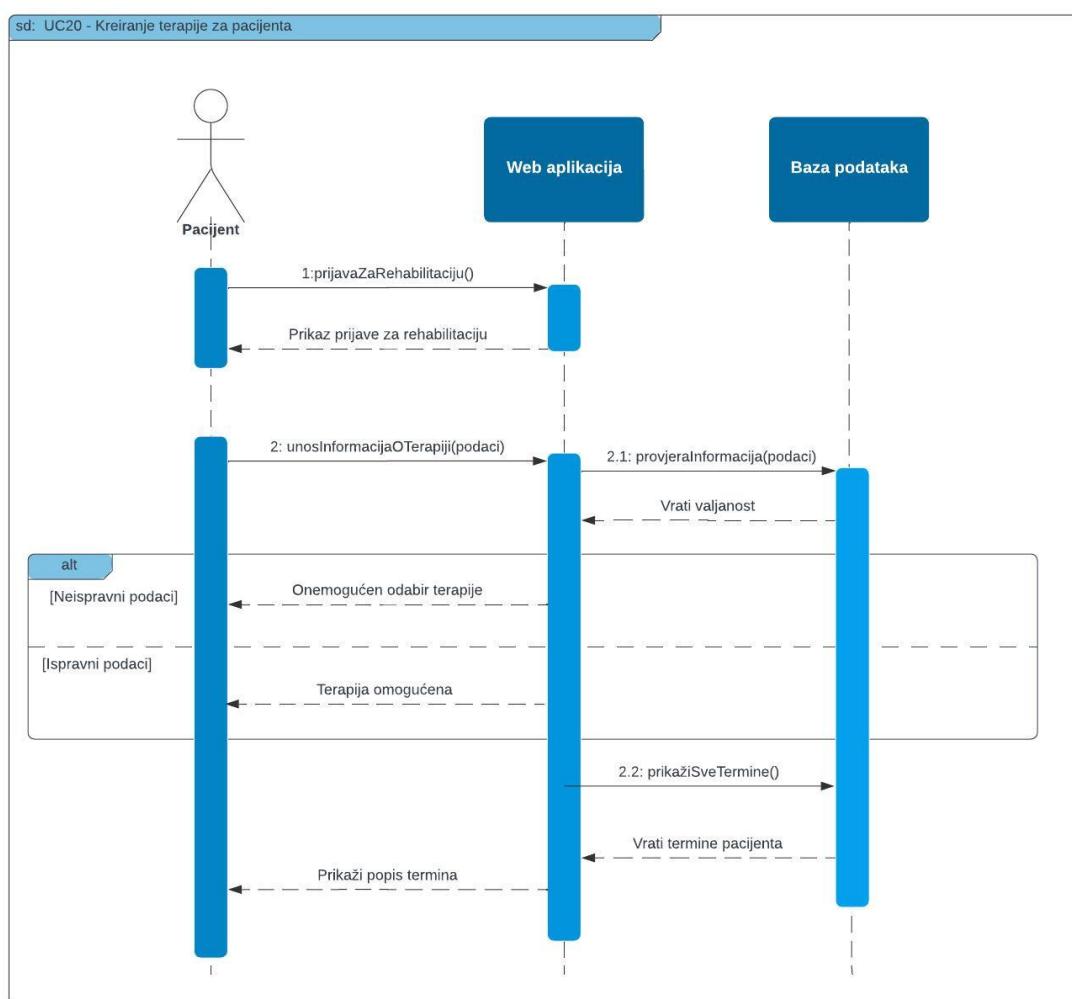
Kada zaposlenik pokreće proces unosa rezultata terapije, prvo odabire pacijenta iz dostupne liste. Nakon odabira pacijenta, web-aplikacija provjerava bazu podataka kako bi dohvatile relevantne podatke o tom pacijentu. Ako pacijent nije pronađen, sustav ispisuje poruku o pogrešci, obavještavajući zaposlenika o nedostatku informacija. U slučaju uspješnog pronalaska pacijenta, zaposlenik unosi rezultate terapije. Uneseni podaci o rezultatima terapije zatim se upisuju u bazu podataka. Nakon uspješnog zapisivanja, sustav prikazuje na zaslonu potvrdu o uspješno unesenom zapisu, pružajući zaposleniku vizualnu potvrdu. Dodatno, nakon unosa rezultata, pacijent ima mogućnost pregleda svojih rezultata terapije unutar web-aplikacije. To omogućuje pacijentu praćenje vlastitog napretka i informiranje o rezultatima terapije.



Slika 3.3: Sekvencijski dijagram za UC22

Obrazac uporabe UC20 - Kreiranje terapije za pacijenta

Kada pacijent zatraži obrazac za prijavu na rehabilitaciju, web-aplikacija generira i vraća prijavu. Pacijent unosi informacije o terapiji, a sustav provjerava ispravnost podataka u bazi. Ako su svi podaci ispravno uneseni, omogućuje se terapija, ali ostaje nepotvrđena. Liječniku se prikazuje zahtjev za terapijom, gdje unosi datum, vrijeme i sobu. Web-aplikacija provjerava valjanost unesenih podataka, uključujući dostupnost sobe i preklapanje termina. Ako su podaci valjni, terapija se potvrđuje, a pacijentu se odobrava. Pacijent prima potvrdu o odobrenju terapije, a dodatno može pregledavati svoje rezultate unutar aplikacije. Ovaj proces osigurava učinkovito upravljanje terapijama, pravovremenu potvrdu i omogućuje pacijentima praćenje vlastitog napretka.



Slika 3.4: Sekvencijski dijagram za UC20

3.2 Ostali zahtjevi

1. Sustav mora podržavati više različitih uloga korisnika
 - Sustav treba omogućiti definiranje različitih razina pristupa i funkcionalnosti ovisno o ulozi korisnika (Administrator, Liječnik, Pacijent, itd.).
2. Korisnički podaci moraju biti sigurno pohranjeni u sustavu
 - Sustav treba koristiti kriptiranje za pohranu i prijenos osjetljivih korisničkih podataka kako bi osigurao privatnost i sigurnost informacija.
3. Brza i pouzdana veza s bazom podataka
 - Sustav treba osigurati pouzdanu vezu s bazom podataka kako bi pristup podacima bio brz i otporan na vanjske pogreške.
4. Odgovarajuće vrijeme izvršavanja za upite prema bazi podataka
 - Izvršavanje upita koji pristupaju bazi podataka ne smije trajati dulje od definiranog vremena, kako bi se osigurala učinkovitost sustava.
5. Podrška za hrvatsku abecedu u korisničkom sučelju i pri unosu teksta
 - Korisničko sučelje i sustav trebaju podržavati hrvatsku abecedu pri unosu, prikazu i obradi tekstualnih podataka.
6. Web aplikacija implementirana korištenjem objektno-orientiranih jezika
 - Sustav treba biti implementiran kao web aplikacija koristeći suvremene objektno-orientirane jezike kako bi bio skalabilan i održiv.
7. Intuitivan i lagan za korištenje
 - Korisničko sučelje treba biti intuitivno i jednostavno za korištenje kako bi korisnici mogli efikasno koristiti sustav bez potrebe za dodatnim uputama.
8. Nadogradnja sustava bez narušavanja postojećih funkcionalnosti
 - Nadogradnje sustava trebaju biti provedene bez negativnog utjecaja na postojeće funkcionalnosti kako bi korisnici nastavili neometano koristiti sustav.
9. Zaštita od neispravnog korištenja korisničkog sučelja

- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnosti i rad sustava. Sustav treba biti otporan na potencijalne greške ili zloupotrebu od strane korisnika.

4. Arhitektura i dizajn sustava

Rehub predstavlja sofisticirano arhitektonsko rješenje koje integrira PostgreSQL bazu podataka, Spring Boot za backend logiku, React za korisničko sučelje, te Tailwind CSS za moderno i responzivno oblikovanje. Sustav je implementiran na cloud platformama Render za backend i bazu te Netlify za frontend, koristeći njihove usluge za brz, skalabilan i pouzdan deploying.

- **Baza podataka: PostgreSQL**

PostgreSQL, snažan sustav za upravljanje relacijskim bazama podataka, predstavlja ključnu komponentu u arhitekturi ove web aplikacije. Struktura baze podataka temeljito je dizajnirana kako bi zadovoljila specifične zahtjeve aplikacije. Kroz pažljivu integraciju s PostgreSQL-om, aplikacija ostvaruje brz i učinkovit pristup podacima, čime se osigurava optimalna performansa. Ova konfiguracija baze podataka predstavlja ključnu komponentu cjelokupnog arhitektonskog rješenja, pružajući čvrstu temeljnu strukturu za rad backend sustava.

- **Backend: Spring Boot**

Spring Boot, kao Java-based framework, pruža snažan temelj za razvoj backend logike. Ova web aplikacija koristi Spring Boot za stvaranje RESTful API-ja koji komunicira s PostgreSQL bazom podataka. Ovo rješenje omogućuje efikasno upravljanje podacima i pruža mogućnost proširivosti sustava kroz modularnost i lakoću integracije.

- **Frontend: React**

React, centralna JavaScript biblioteka, određuje arhitekturu frontend dijela

ove aplikacije. Modularne React komponente omogućuju čist i jednostavan kod, prilagodljiv specifičnim zahtjevima sučelja. Ovaj pristup omogućuje stvaranje brzog, dinamičkog korisničkog sučelja koje se lako održava. Kroz React, osigurava se fluidna navigacija kroz aplikaciju, pridonoseći ukupnom intuitivnom korisničkom iskustvu.

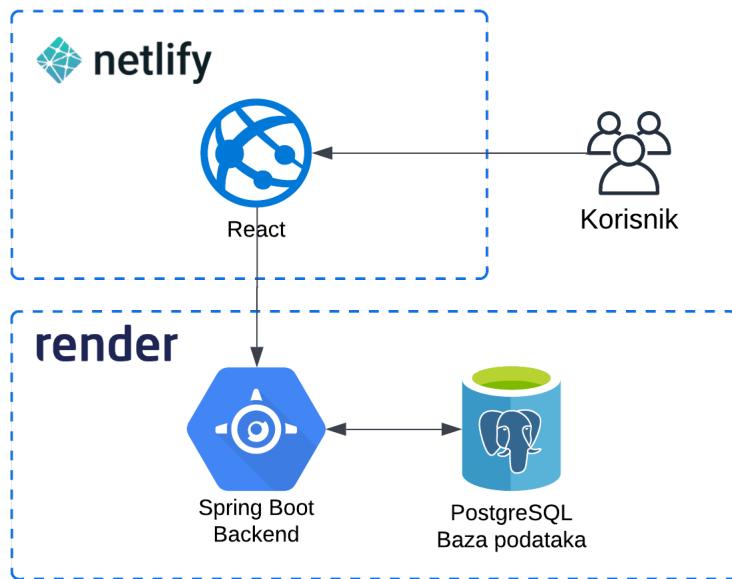
- **Frontend Integracija:** Tailwind CSS, Axios, React-Router-Dom, reCAPTCHA

Frontend aplikacije unaprijeđen je integracijom Tailwind CSS za brzo oblikovanje sučelja, Axios za efikasnu komunikaciju s backendom, React-Router-Dom za fluidnu navigaciju te reCAPTCHA sustava radi dodatne sigurnosti. Tailwind CSS pridonosi estetskom izgledu, Axios omogućuje dinamičku interakciju s backendom, React-Router-Dom olakšava upravljanje rutama, dok reCAPTCHA sprječava zlonamjerne aktivnosti. Kroz ovu integraciju postiže se uravnoteženo frontend rješenje koje kombinira funkcionalnost, estetiku i sigurnost korisničkog iskustva.

- **Deployment:** Render za backend i Netlify za frontend

Backend i baza podataka su postavljeni na Render platformu za optimalno upravljanje resursima. Render osigurava automatsko skaliranje i brzu dostupnost aplikacije koristeći njihove cloud usluge. Na renderu je također podešeno kontinuirano postavljanje koje omogućuje automatsko ažuriranje softvera. Frontend je zasebno postavljen na Netlify platformi, koja pruža brzo i globalno dostupno CDN (Content Delivery Network).

Rehub predstavlja integrirano arhitektonsko rješenje koje kombinira najbolje prakse u razvoju aplikacija, pružajući stabilnost, skalabilnost i visoku razinu korisničkog iskustva. Korištenje navedenih tehnologija i servisa osigurava da aplikacija bude spremna za izazove suvremenog web razvoja.



Slika 4.1: Dijagram arhitekture

4.1 Baza podataka

U bazi su stvorene relacije koje olakšavaju jednostavno rukovanje podacima, uključujući dodavanje, brisanje, izmjenu i dohvaćanje podataka. Svi entiteti i relacije su organizirani prema trećoj normalnoj formi kako bi se izbjegla redundancija podataka. Za bolje vizualiziranje strukture baze podataka, napravljen je relacijski dijagram koji je prikazan u nastavku.

Baza podataka se sastoji od tablica (relacija) koje su definirane svojim imenom i skupom atributa. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- rehub_user
- employee
- user_role
- role
- patient
- therapy
- therapy_result
- doctor
- appointment

- room
- equipment
- faq
- personal_data
- verification
- password_reset
- flyway_history_schema

4.1.1 Opis tablica

Prva čelija svake tablice označava njenu ime. U prvom stupcu navedeni su atributi tog entiteta, u drugom stupcu naveden je tip varijable, a u trećem opis svakog pojedinog atributa. Svjetlozelenom bojom označeni su primarni ključevi. Svjetlo plavom označeni su strani ključevi.

rehub_user		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR KORISNIKA
USERNAME	VARCHAR	KORISNIČKO IME S KOJIM SE KORISNIK PRIJAVLJUJE
PASSWORD	VARCHAR	LOZINKA S KOJOM SE KORISNIK PRIJAVLJUJE
STATUS	VARCHAR	STATUS KORISNIKA
PERSONAL_DATA_ID	VARCHAR	JEDINSTVENI IDENTIFIKATOR KORISNIKA

Entitet označava korisnika aplikacije te sadrži atribute: id, username, password, status, personal_data_id.

employee		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR ZAPOSLENIKA
FIRST_NAME	VARCHAR	IME ZAPOSLENIKA

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

employee		
LAST_NAME	VARCHAR	PREZIME ZAPOSLENIKA
PHONE_NUMBER	VARCHAR	BROJ MOBITELA ZAPOSLENIKA
PROFESSION	VARCHAR	STRUKA ZAPOSLENIKA
GENDER	VARCHAR	SPOL ZAPOSLENIKA
CREATED_AT	TIMESTAMP	TRENUTAK KREIRANJA
LAST_MODIFIED_AT	TIMESTAMP	TRENUTAK ZADNJE PROMJENE
USER_ID	VARCHAR	JEDINSTVENI IDENTIFIKATOR KORISNIKA

Entitet označava osobe zaposlene u klinici. Atributi ovoga entiteta su: id, first_name, last_name, phone_number, profession, gender, created_at, last_modified_at, user_id.

user_role		
USER_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR KORISNIKA
ROLE_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR VRSTE KORISNIKA

Entitet označava vrstu korisnika aplikacije, a njegovi atributi su: user_id, role_id.

role		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR ULOGE
NAME	VARCHAR	IME ULOGE

Entitet označava ulogu i sastoje se od atributa: id, name.

patient		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR PACIJENTA
FIRST_NAME	VARCHAR	IME PACIJENTA

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

patient		
LAST_NAME	VARCHAR	PREZIME PACIJENTA
GENDER	VARCHAR	SPOL PACIJENTA
PHONE_NUMBER	VARCHAR	BROJ MOBITELA PACIJENTA
DATE_OF_BIRTH	DATE	DATUM ROĐENJA PACIJENTA
CREATED_AT	TIMESTAMP	TRENUTAK KREIRANJA
LAST_MODIFIED_AT	TIMESTAMP	TRENUTAK ZADNJE PROMJENE
USER_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR KORISNIKA

Entitet predstavlja pacijenta koji je korisnik aplikacije. Atributi su: id, first_name, last_name, gender, phone_number, date_of_birth, created_at, last_modified_at, user_id.

therapy		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR TERAPIJE
TYPE	VARCHAR	VRSTA TERAPIJE
REQUEST	VARCHAR	ZAHTJEV ZA TERAPIJOM
STATUS	VARCHAR	STATUS TERAPIJE
REF_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR
CREATED_AT	TIMESTAMP	TRENUTAK KREIRANJA
LAST_MODIFIED_AT	TIMESTAMP	TRENUTAK ZADNJE PROMJENE
PATIENT_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR PACIJENTA
EMPLOYEE_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR ZAPOSLENIKA
ROOM_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR SOBE

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

therapy		
APPOINTMENT_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR TERMINA
THERAPY_RESULT_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR REZULTATA TERAPIJE
DOCTOR_FULL_NAME	VARCHAR	PUNO IME DOKTORA
THERAPY_SCAN	VARCHAR	NAZIV UPUTNICE ZA TERAPIJU

Entitet opisuje terapiju dodijeljenu pacijentu. Njeni atributi su: id, type, request, status, ref_id, created_at, last_modified_at, patient_id, employee_id, room_id, appointment_id, therapy_result_id, doctor_full_name, therapy_scan.

doctor		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR DOKTORA
FIRST_NAME	VARCHAR	IME DOKTORA
LAST_NAME	VARCHAR	PREZIME DOKTORA
PIN	VARCHAR	PIN DOKTORA
MEDICAL_NO	VARCHAR	MEDICINSKI BROJ
SPECIALITY	VARCHAR	DOKTOROVA SPECIJALIZACIJA
DATE_OF_EMPLOYMENT	DATE	DATUM ZAPOSLENJA
DATE_OF_REGISTRATION	DATE	DATUM REGISTRACIJE
DATE_OF_BIRTH	TIMESTAMP	DATUM ROĐENJA

Entitet predstavlja doktora koji je korisnik aplikacije. Atributi su: id, first_name,

last_name, pin, medical_no, speciality, date_of_employment, date_of_registration, date_of_birth.

appointment		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR TERMINA
START_AT	TIMESTAMP	POČETAK TERMINA
END_AT	TIMESTAMP	KRAJ TERMINA

Entitet predstavlja termin na koji se pacijent naručio, a atributi su: id, start_at, end_at.

therapy_result		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR REZULTATA TERAPIJE
STATUS	VARCHAR	STATUS TERAPIJE
RESULT	VARCHAR	KONAČNI REZULTAT TERAPIJE

Ovaj entitet predstavlja rezultat terapije, a definiran je sljedećim atributima: id, status, result.

room		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR SOBE
LABEL	VARCHAR	OZNAKA SOBE
CAPACITY	BIGSERIAL	KAPACITET SOBE
STATUS	VARCHAR	STATUS O KORIŠTENJU SOBE
SPECIAL_MESSAGE	VARCHAR	NAPOMENA

Entitet predstavlja sobu u kojoj se odvija terapija, a njeni atributi su: id, label, capacity, status, special_message.

equipment		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR OPREME
NAME	VARCHAR	IME OPREME
STATUS	VARCHAR	STATUS O KORIŠTENJU OPREME
SPECIAL_MESSAGE	VARCHAR	NAPOMENA
ROOM_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR SOBE

Ovaj entitet se odnosi na opremu koja se koristi prilikom terapije, a opisuju ju: id, name, status, special_message, room_id.

personal_data		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR KORISNIKA
FIRST_NAME	VARCHAR	IME KORISNIKA
LAST_NAME	VARCHAR	PREZIME KORISNIKA
PIN	VARCHAR	OIB KORISNIKA
PHIN	VARCHAR	MATIČNI BROJ KORISNIKA
DATE_OF_BIRTH	DATE	DATUM ROĐENJA KORISNIKA

Ovaj entitet sadrži osobne podatke o korisniku aplikacije, a atributi su mu: id, first_name, last_name, pin, phin i date_of_birth.

faq		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR PITANJA I ODGOVORA
QUESTION	VARCHAR	PITANJE
ANSWER	VARCHAR	ODGOVOR

Entitet se odnosi na često postavljena pitanja i sastoji se od sljedećih atributa: id, question te answer.

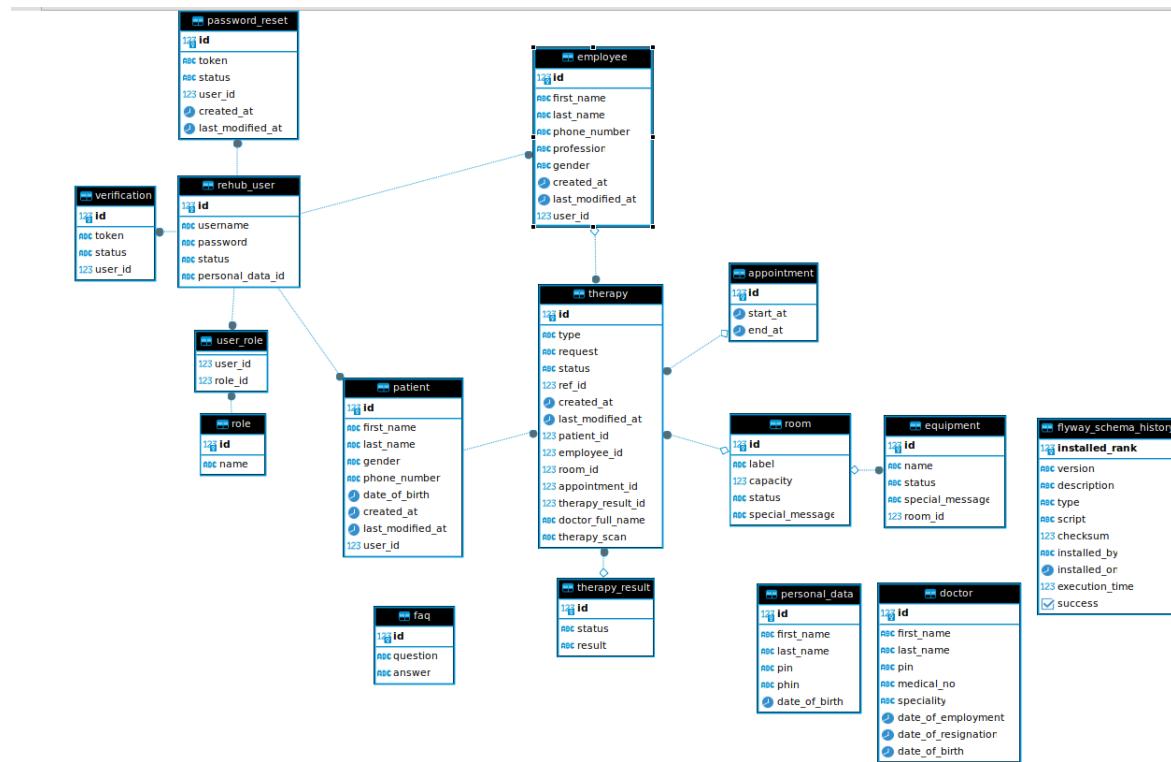
verification		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR VERIFIKACIJE
TOKEN	VARCHAR	TOKEN VERIFIKACIJE
STATUS	VARCHAR	STATUS VERIFIKACIJE
USER_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR KORISNIKA

Entitet se odnosi na verifikaciju i sastoji se od sljedećih atributa: id, token, status i user_id.

password_reset		
ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR PONOVNOG POSTAVLJANJA LOZINKE
TOKEN	VARCHAR	TOKEN PONOVNOG POSTAVLJANJA LOZINKE
STATUS	VARCHAR	STATUS PONOVNOG POSTAVLJANJA LOZINKE
USER_ID	BIGSERIAL	JEDINSTVENI IDENTIFIKATOR KORISNIKA
CREATED_AT	TIMESTAMP	TRENUTAK KREIRANJA
LAST_MODIFIED_AT	TIMESTAMP	ZADNJE MIJENJANO

Entitet se odnosi na ponovno postavljanje lozinke i sastoji se od sljedećih atributa: id, token, status, user_id, created_at i last_modified_at.

4.1.2 Dijagram baze podataka



Slika 4.2: Dijagram baze podataka

4.2 Dijagram razreda

Dijagramom razreda prikazujemo razrede u sustavu, njihove atributе i metode te veze između razreda koji se nasljeđuju ili međusobno komuniciraju. U nastavku slijede dijagrami čiji razredi imaju sličnu funkcionalnost i razinu apstrakcije

EntityClassDiagram - prikazuje razrede koji pripadaju sloju Model. Svaki od razreda se preslikava u odgovarajuću tablicu u bazi

RepositoryClassDiagram – prikazuje razrede koji pripadaju sloju Repository. Ovaj sloj komunicira sa bazom podataka i sa slojem Service.

ServiceClassDiagram – prikazuje razrede koji pripadaju sloju Service. Ovaj sloj komunicira sa slojem Repository i Controller.

ControllerClassDiagram – prikazuje razrede sloja Controller. Ovaj sloj komunicira sa slojem Service i s frontend dijelom aplikacije.

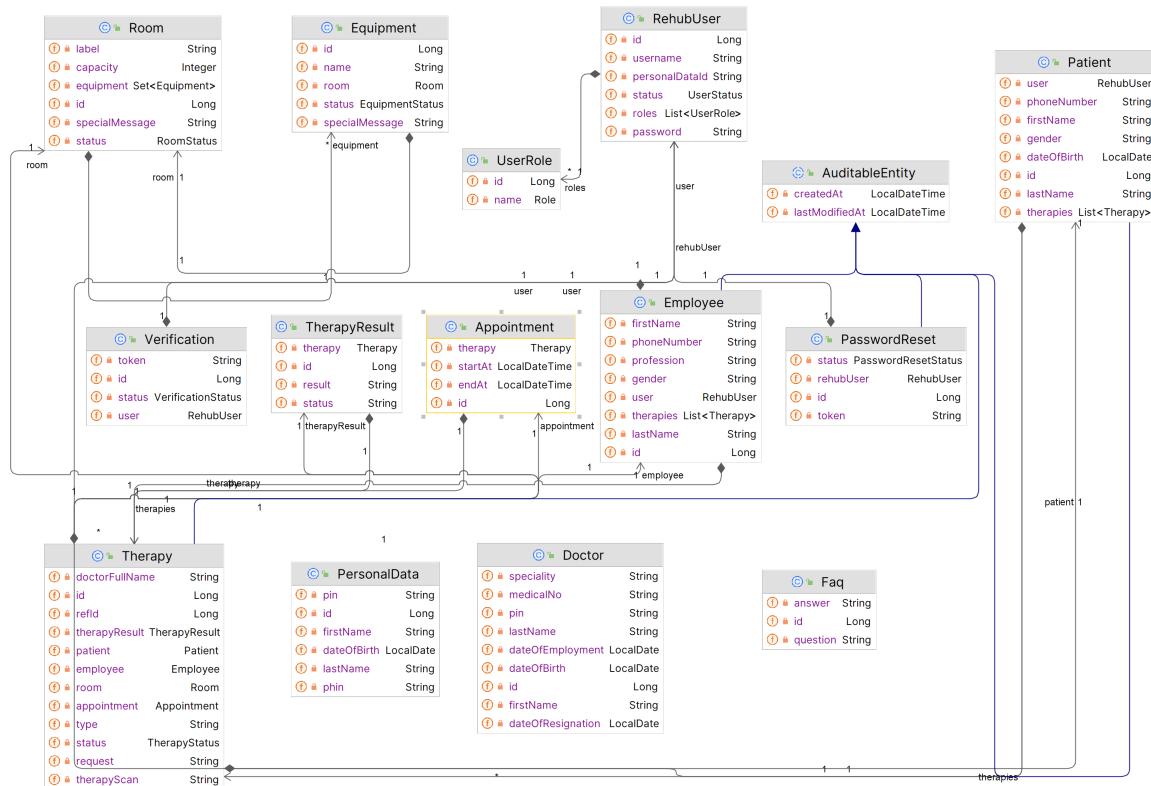
EnumClassDiagram – prikazuje enumeracije koje koristi sloj Model.

ConfigClassDiagram – prikazuje razrede sloja Config. Ovaj sloj služi za sigurnosne potrebe i komuniciranje putem e-maila.

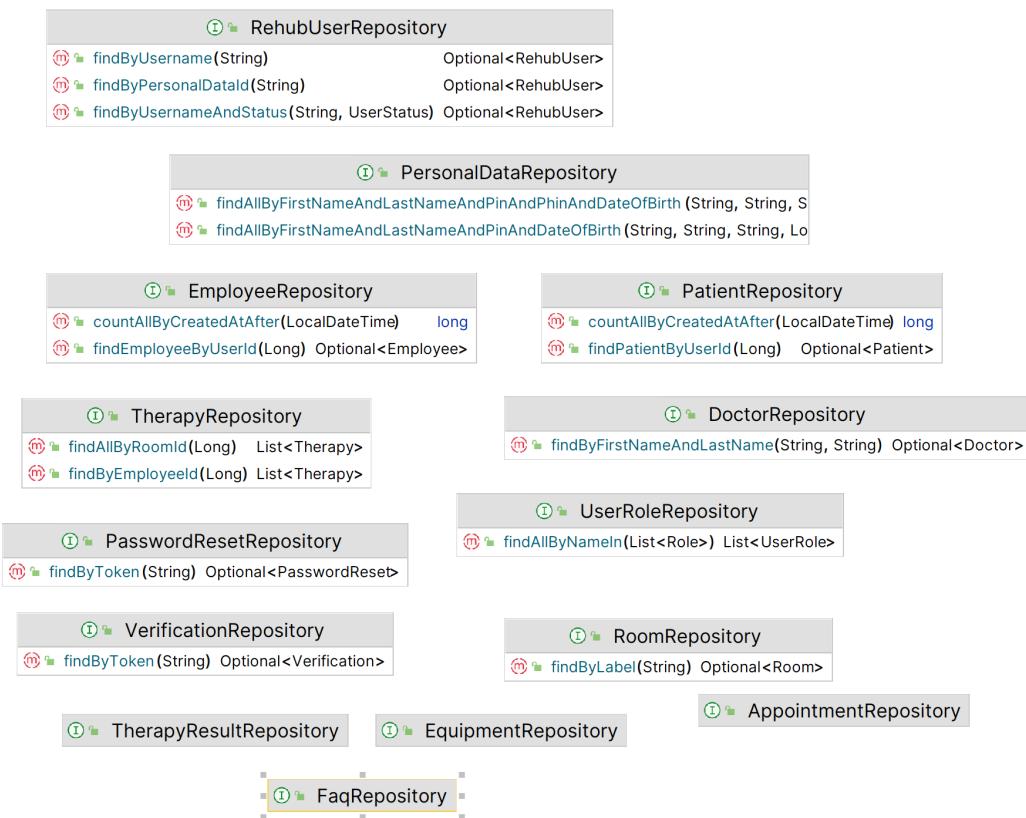
SecurityClassDiagram - prikazuje razrede sloja Security. Ovaj sloj služi za sigurnost i detalje o korisniku.

RequestClassDiagram – prikazuje razrede sloja Request. Ovaj sloj komunicira sa slojem Controller u svrhu validacije Post Request-a.

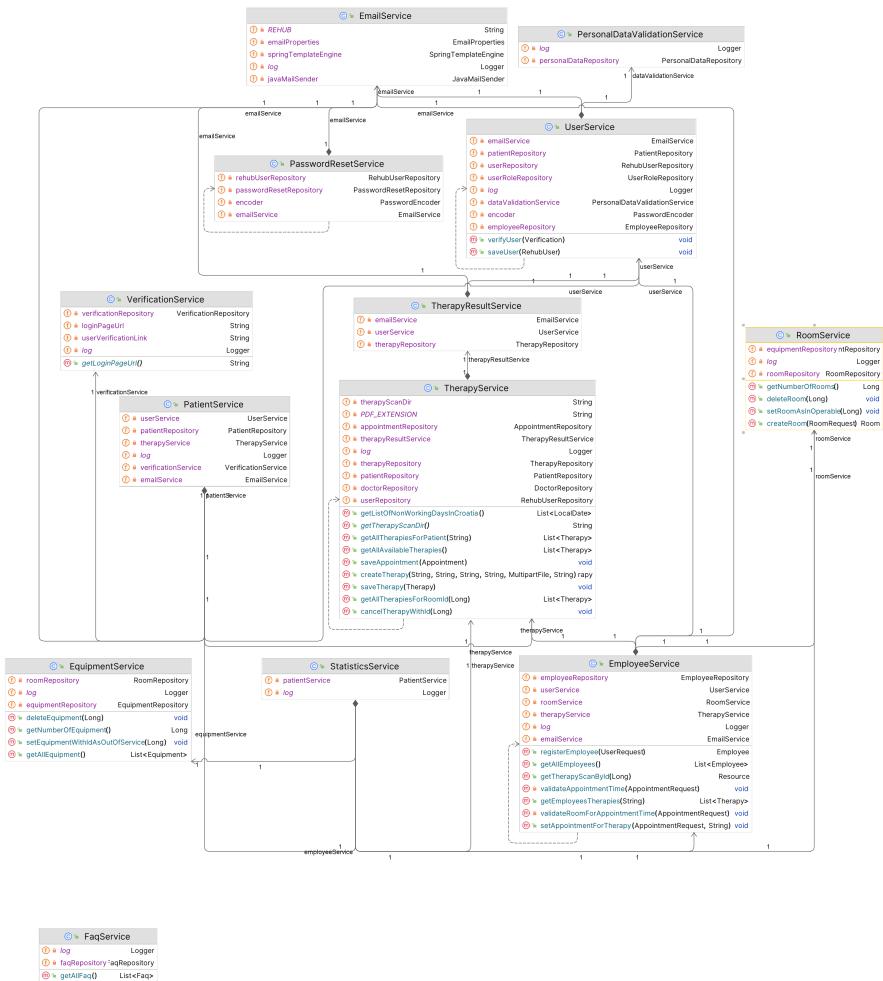
ResponseClassDiagram – prikazuje razrede sloja Response. Ovaj sloj komunicira sa slojem Controller u svrhu vraćanja Response poruke.



Slika 4.3: Dijagram razreda (Model)



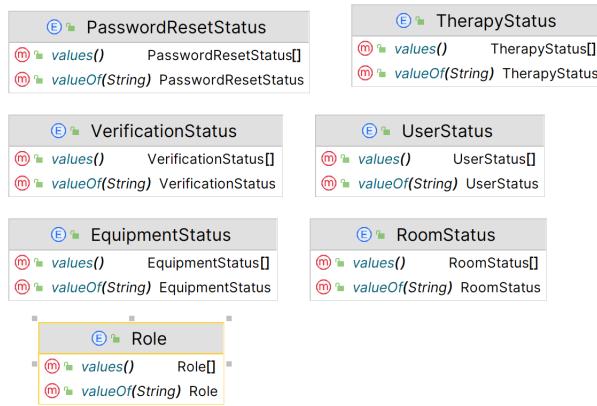
Slika 4.4: Dijagram razreda (Repository)



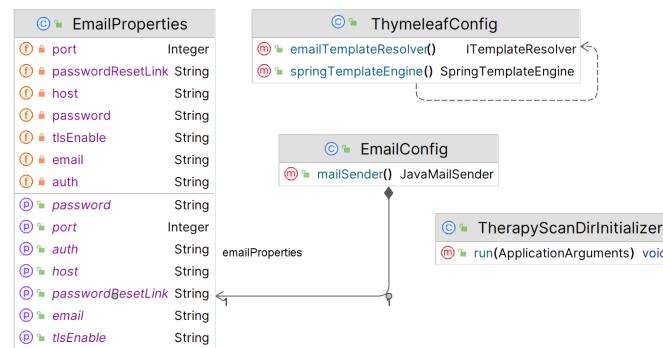
Slika 4.5: Dijagram razreda (Service)



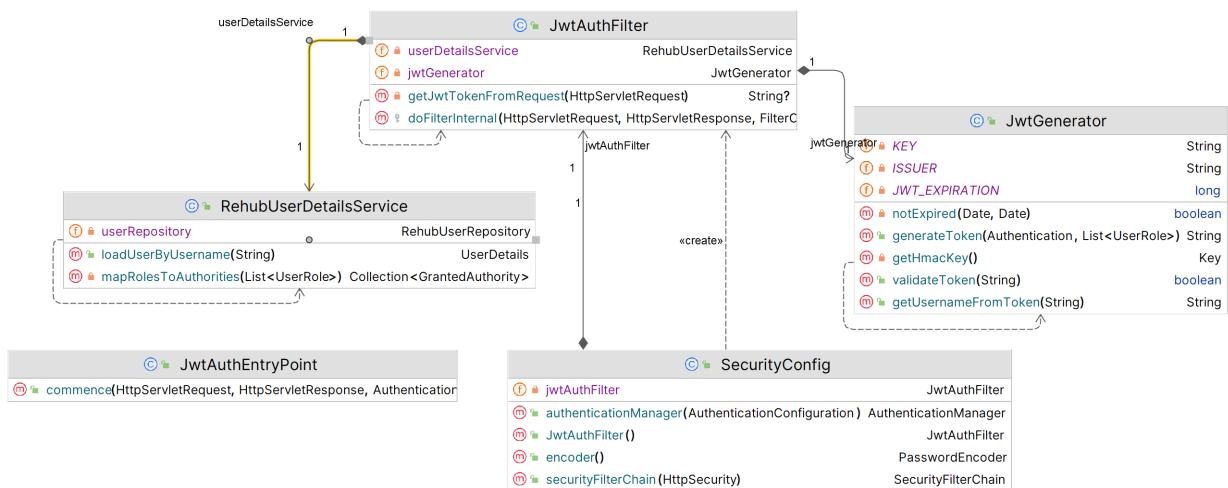
Slika 4.6: Dijagram razreda (Controller)



Slika 4.7: Dijagram enumeracija (Model)



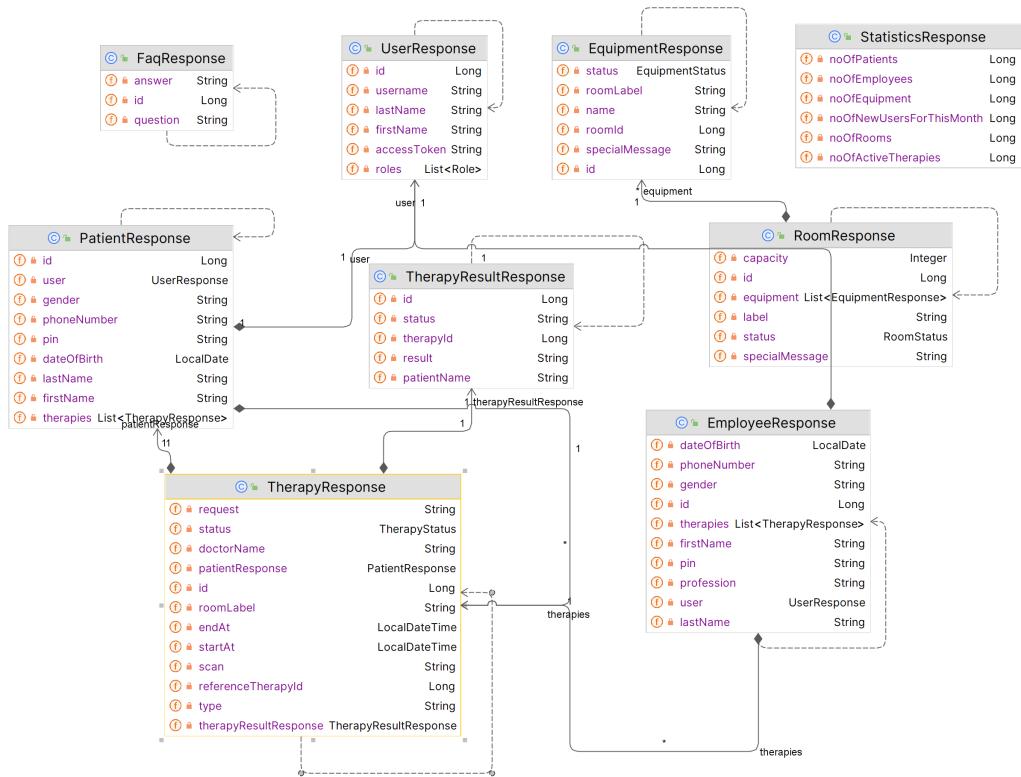
Slika 4.8: Dijagram razreda (Config)



Slika 4.9: Dijagram razreda (Security)



Slika 4.10: Dijagram razreda (Request)



Slika 4.11: Dijagram razreda (Response)

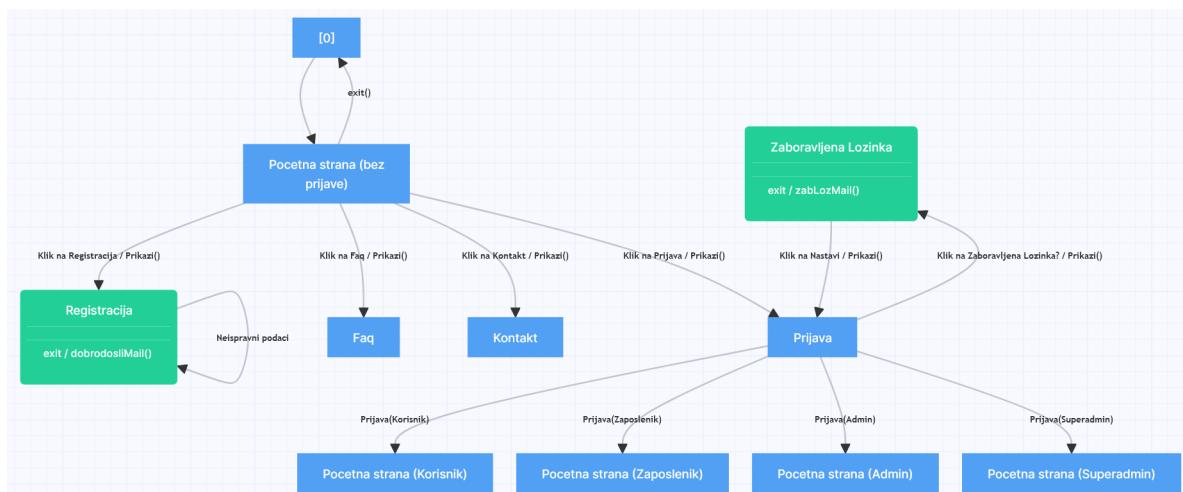
4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na dogadajima. Prilikom pokretanja aplikacije dolazi se na početnu stranicu (eng. Home page). Neregistrirani korisnik ima mogućnost registracije i pregled kontakta i učestalih pitanja bez registracije

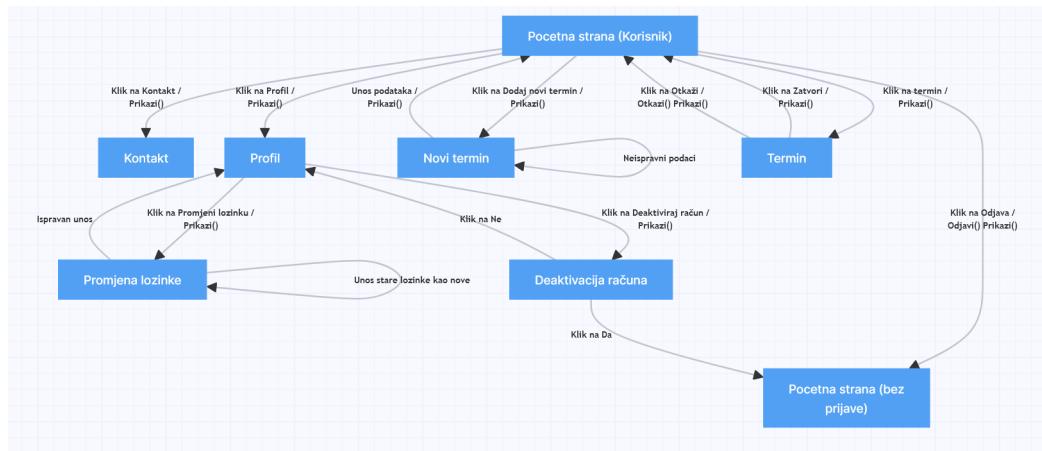
Registrirani korisnik na svojem početnom zaslonu nakon prijave može kliknuti na svoj profil gdje vidi svoje podatke, također na profilnoj stranici može deaktivirati račun i promjenit lozinku. Klikom na Dodaj novi termin korisnik se može prijaviti za termin koji može i naknadno otkazati.

Zaposlenik na svojem zaslonu može kao i korisnik kliknut na svoj profil gdje vidi svoje podatke i mjenja lozinku. Može davati termine i sobe pristiglim prijavama kao i pregledavati gotove termine te pisati rezultate terapije.

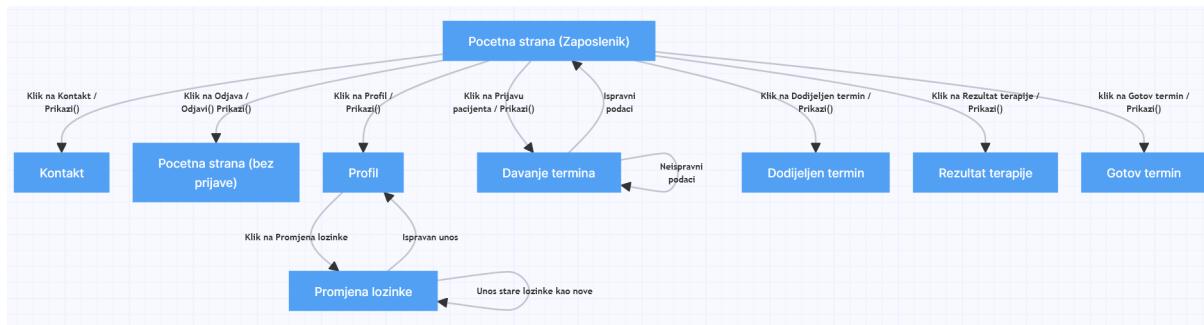
Superadmin ima uvid u statistiku, ima popis svih pacijenta i osoblja. Može dodavati, brisati osoblje te im dati uloge admina. Također ima popis svih soba i opreme koje može brisati i dodavati ili staviti da nisu u funkciji.



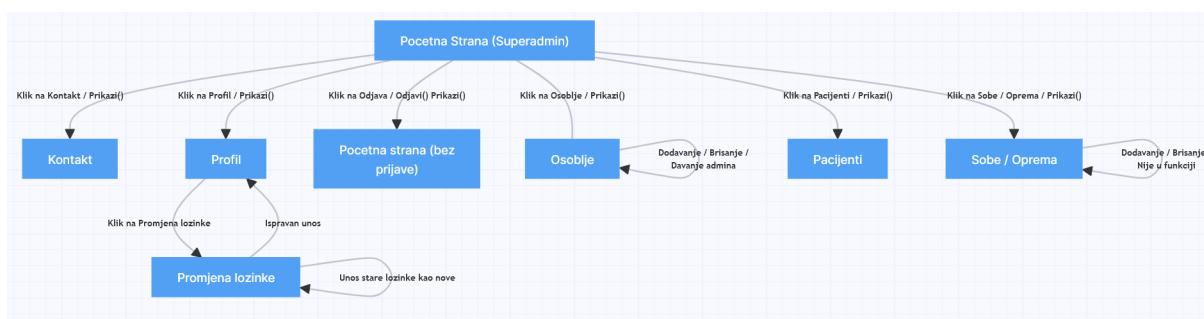
Slika 4.12: Dijagram stanja (neprijavljeni korisnik)



Slika 4.13: Dijagram stanja (prijavljeni korisnik)



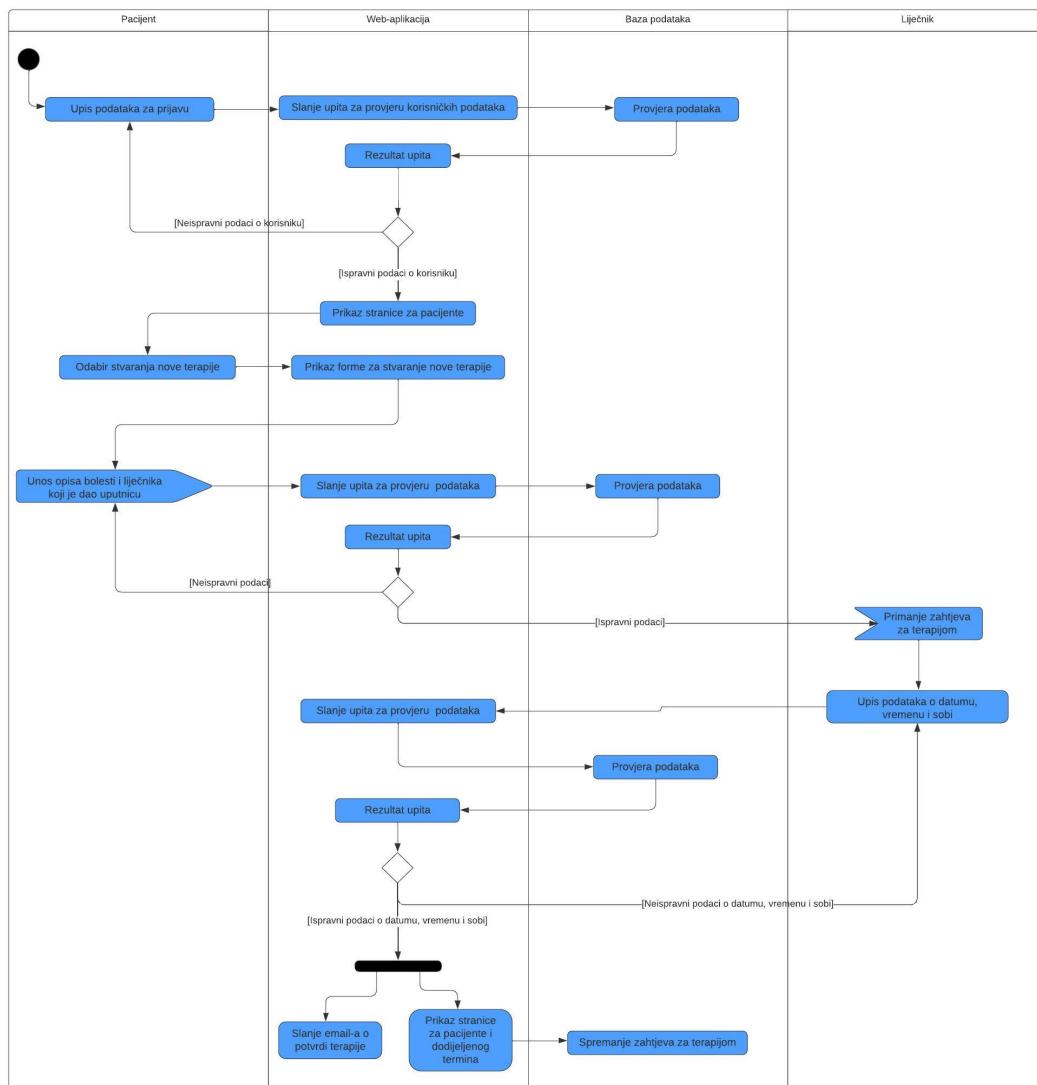
Slika 4.14: Dijagram stanja (zaposlenik)



Slika 4.15: Dijagram stanja (superadmin)

4.4 Dijagram aktivnosti

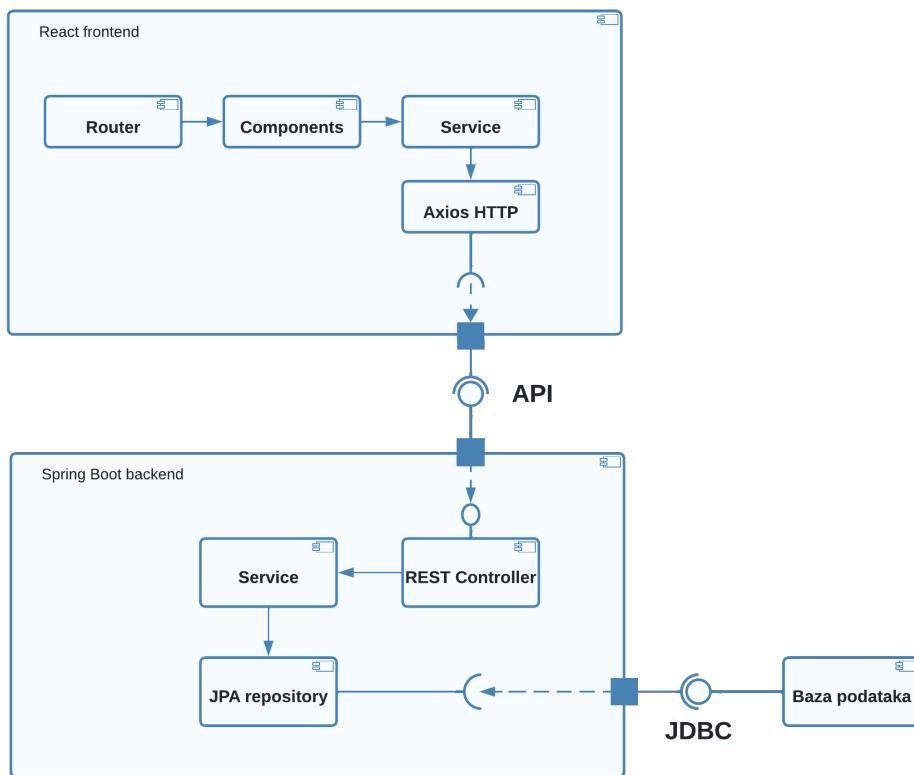
Pacijent započinje proces prijavom u sustav putem web-aplikacije, unoseći svoje korisničke podatke. Web-aplikacija šalje upit za provjeru tih podataka bazi podataka, koja zatim vraća rezultat. Ako su uneseni podaci neispravni, pacijent mora ponovno unijeti podatke, inače mu se prikazuje stranica za pacijente. Nakon uspješne prijave, pacijent odabire stvaranje nove terapije, što rezultira prikazom forme za unos podataka o bolesti i liječniku koji je dao uputnicu. Ponovno se šalje upit za provjeru tih podataka bazi, a ako su neispravni, pacijent ponovno unosi informacije; inače, liječnik prima zahtjev za terapijom i unosi podatke o datumu, vremenu i sobi za terapiju. Web-aplikacija ponovno šalje upit za provjeru podataka bazi, a u slučaju neispravnih podataka, liječnik ih ponovno unosi. U suprotnom, web-aplikacija šalje e-mail o potvrdi terapije i prikazuje stranicu za pacijenta s dodijeljenim terminom. Nakon toga, podaci o zahtjevu za terapijom spremaju se u sustav.



Slika 4.16: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti detaljno opisuje strukturu sustava, međusobne veze između komponenti te njihove odnose s okolinom. Prikazuje integraciju frontend i backend aplikacija putem REST API-ja. Backend aplikacija je složena od servisa, repozitorija i kontrolera. Repozitorij komunicira s bazom podataka putem SQL upita kako bi dohvatio potrebne podatke. Servis komunicira s repozitorijem i kontrolerom. Kontroler obrađuje dolazne zahtjeve s frontenda. Frontend aplikacija obuhvaća routere, komponente, servise te razne biblioteke. Router je komponenta koja, na temelju korisničkog zahtjeva za određeni URL, određuje koja datoteka (stranica) će biti isporučena. Komponente predstavljaju stranice koje se isporučuju kao JavaScript datoteke (HTML, CSS, JavaScript kod) ovisne o React biblioteci. Servisi imaju ulogu zahtjeva prema backendu, validacije i sličnih zadataka. Axios, kao biblioteka HTTP klijenta, pojednostavljuje slanje HTTP zahtjeva prema REST API-ju, odnosno prema backend aplikaciji.



Slika 4.17: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Frontend aplikacije realiziran je pomoću Reacta korištenjem programskog jezika Javascript. React je Javascript biblioteka otvorenog koda za izgradnju korisničkih sučelja temeljenih na komponentama korisničkog sučelja. Backend aplikacije ostvaren je pomoću Spring Boota zasnovanog na razvojnom okviru Spring, pisanog u programskom jeziku Java. Od razvojnih okolina korišteni su Visual Studio Code i IntelliJ IDEA. Pri uspostavi arhitekture su također korišteni Render, Netlify, Tailwind CSS, Docker Docs, Axios, React-Router-Dom i reCAPTCHA. Baza podataka pisana je u PostgreSQL. Za komunikaciju, dogovore i sastanke korišteni su WhatsApp, Microsoft Teams te Atlassian. Lucidchart je alat korišten za kreiranje dijagrama, a sama dokumentacija je pisana u markup jeziku LaTeX te uz pomoć alata Overleaf. Udaljeni repozitorij projekta smješten je na web platformi GitHub.

5.2 Ispitivanje programskog rješenja

U ovom poglavlju provedeno je ispitivanje implementiranih funkcionalnosti na razini komponenti, s fokusom na ispitivanje jedinica (unit testing). Razrađeno je 6 ispitnih slučajeva koji obuhvaćaju redovne slučajeve, rubne uvjete, izazivanje pogrešaka (exception throwing) te ispitivanje neispravnog formata podataka.

Ispitni slučaj 1: Redovni slučaj

Opis: provjera ispravnog izvođenja osnovne funkcionalnosti. Unosi se nova soba.

Slika 5.1: Test 1: Redovni slučaj

Nova soba je uspješno dodana.

D-237	Kapacitet: 32	X	-
A-111	Kapacitet: 23	X	-
D1	Kapacitet: 12	X	-
D2	Kapacitet: 2	X	-
A-112	Kapacitet: 20	X	-

Slika 5.2: Rezultat testa 1: Redovni slučaj

Ispitni slučaj 2: Rubni uvjet

Opis: ispitivanje ponašanja sustava kada su ulazni podaci na granici dopuštenih vrijednosti. Unosi se kapacitet sobe s minimalnom vrijednosti.

UNESI NOVU SOBU

Oznaka sobe: A-101 Kapacitet: 1

Dodatne informacije:

UNESI SOBU

Slika 5.3: Test 2: Rubni uvjet

Nova soba je uspješno dodana.

D-237	Kapacitet: 32	X	–
A-111	Kapacitet: 23	X	–
D1	Kapacitet: 12	X	–
D2	Kapacitet: 2	X	–
A-112	Kapacitet: 20	X	–
A-101	Kapacitet: 1	X	–

Slika 5.4: Rezultat testa 2: Rubni uvjet

Ispitni slučaj 3: Izazivanje pogrešaka

Opis: provjera kako sustav reagira na izazivanje pogreške. Dodaje se soba s kapacitetom 0.

UNESI NOVU SOBU

Oznaka sobe:	Kapacitet:
D-343	0
Dodatne informacije:	
UNESI SOBU	

Slika 5.5: Test 3: Izazivanje pogrešaka

Soba nije dodana i iskače poruka o pogrešci.

D-237	Kapacitet: 32
A-111	Kapacitet: 23
D1	Kapacitet: 12
D2	Kapacitet: 2
A-112	Kapacitet: 20
A-101	Kapacitet: 1

UNESI NOVU SOBU

Oznaka sobe:	Kapacitet:
D-343	0
Dodatne informacije:	
UNESI SOBU	

Postavljeno u vijeće. Projektno
četvrti poslovne potrebe. Microsoft Windows.
Windows 10 Pro

Slika 5.6: Rezultat testa 3: Izazivanje pogrešaka

Ispitni slučaj 4: Ispitivanje neispravnog formata podataka

Opis: unos podataka u pogrešnom formatu. Unosi se zaposlenik s OIB-om u pogrešnom formatu.

UNESI NOVOG ZAPOSLENIKA

Ime:	Prezime:
Jude	Bellingham
E-mail:	OIB:
jude.belligol@fer.hr	1234
Datum rođenja:	Profesija:
05/13/2002	Doktor
Broj mobitela:	Spol:
+385991234568	Muško

UNESI ZAPOSLENIKA

Slika 5.7: Test 4: Ispitivanje neispravnog formata podataka

Nije dodan zaposlenik i iskače poruka o pogrešci.

Zaposlenici

- Matija Fabić **ADMIN**
- Alida Kušen
- Petar Preradović

UNESI NOVOG ZAPOSLENIKA

Ime:	Prezime:
Jude	Bellingham
E-mail:	OIB:
jude.belligol@fer.hr	1234
Datum rođenja:	Profesija:
05/13/2002	Doktor
Broj mobitela:	Spol:
+385991234568	Muško

UNESI ZAPOSLENIKA

OIB nije ispravne duljine

Podaci nisu valjni. Provjerite konzolne podatke! Windows

Slika 5.8: Rezultat testa 4: Ispitivanje neispravnog formata podataka

Ispitni slučaj 5: Ispitivanje neispravnog formata podataka

Opis: unos podataka u pogrešnom formatu. Unosi se broj mobitela u pogrešnom formatu.

UNESI NOVOG ZAPOSLENIKA

Ime: Jude	Prezime: Bellingham
E-mail: jude.belligol@fer.hr	OIB: 12344321456
Datum rođenja: 05/13/2002	Profesija: Doktor
Broj mobitela: 0991234568	Spol: Muško

UNESI ZAPOSLENIKA

Slika 5.9: Test 5: Ispitivanje neispravnog formata podataka

Nije dodan zaposlenik i iskače poruka o pogrešci.

UNESI NOVOG ZAPOSLENIKA

Matija Fabić ADMIN	Doktor
Alida Kušen	Računarac
Petar Preradović	Pulmolog

Jude Bellingham

E-mail: jude.belligol@fer.hr OIB: 12344321456

Datum rođenja: 05/13/2002 Profesija: Doktor

Broj mobitela: 0991234568 Spol: Muško

UNESI ZAPOSLENIKA

Neispravan format broja. Koristite predbroj +385.

Proverite svoju vežbu: Proverite komunikacije posteljke Windows.

Slika 5.10: Rezultat testa 5: Ispitivanje neispravnog formata podataka

Ispitni slučaj 6: Ispitivanje neispravnog formata podataka

Opis: unos podataka u pogrešnom formatu. Kao kapacitet sobe unose se slova.

UNESI NOVU SOBU

Oznaka sobe:	Kapacitet:
D-343	pet
Dodatne informacije:	
UNESI SOBU	

Slika 5.11: Test 6: Ispitivanje neispravnog formata podataka

Nije dodana soba i iskače poruka o pogrešci.

D-237	Kapacitet: 32 X (x)
A-111	Kapacitet: 23 X (x)
D1	Kapacitet: 12 X (x)
D2	Kapacitet: 2 X (x)
A-112	Kapacitet: 20 X (x)
A-101	Kapacitet: 1 X (x)

UNESI NOVU SOBU

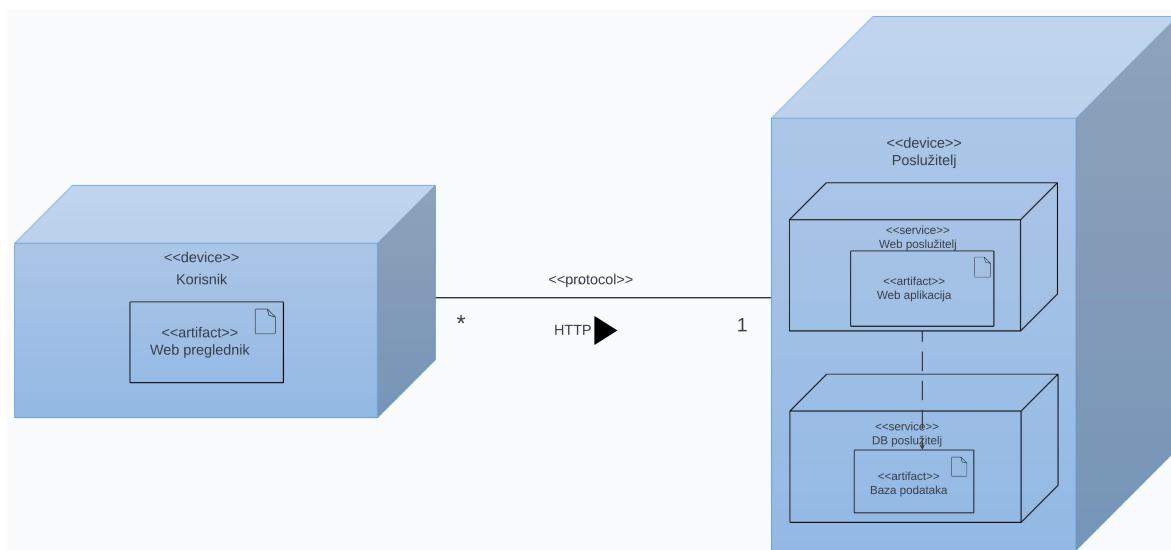
Oznaka sobe:	Kapacitet:
D-343	pet
Dodatne informacije:	
UNESI SOBU	

Copyright 2024, ReHub X
Podaci nisu validni. Proverite podatke sobe. Microsoft Windows, revija 1.6

Slika 5.12: Rezultat testa 6: Ispitivanje neispravnog formata podataka

5.3 Dijagram razmještaja

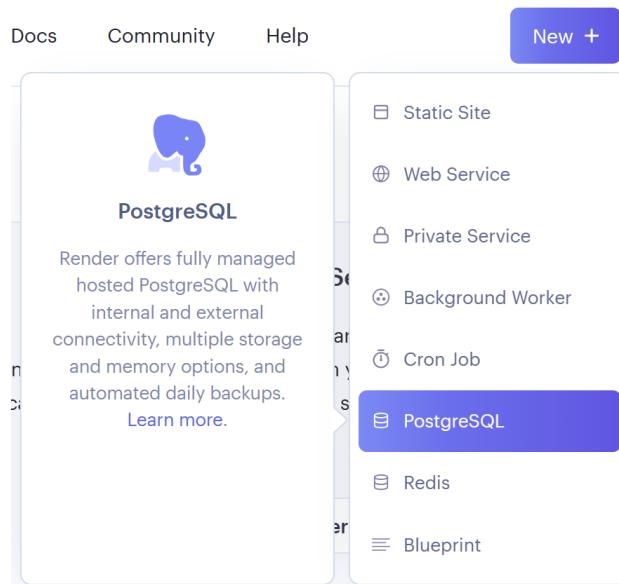
Dijagram razmještaja opisuje strukturu sustava, tj. odnose između sklopoških i programske dijelova. Donja slika predstavlja specifikacijski dijagram razmještaja, pri čemu se na poslužiteljskom računalu nalaze web aplikacija i baza podataka. Pristup aplikaciji klijenti ostvaruju putem web preglednika. Funkcioniranje sistema temelji se na modelu "klijent - poslužitelj", gdje klijenti zahtijevaju usluge od poslužitelja putem HTTP zahtjeva te očekuju odgovor. Poslužitelj obrađuje zahtjeve i šalje odgovore klijentima, a istovremeno upravlja komunikacijom s bazom podataka.



Slika 5.13: Specifikacijski dijagram razmještaja

5.4 Upute za puštanje u pogon

Aplikacija je uspješno implementirana i dostupna na platformi Netlify. Za poslužitelj backend-a koristi se alat Render. Kako bi iskoristili sve korisne značajke koje Render pruža, potrebno je stvoriti instance baze podataka, poslužitelja za backend, te poslužitelja za frontend aplikacije. Prvi korak je kreiranje PostgreSQL baze podataka.



Slika 5.14: Izbornik za stvaranje nove baze podataka

Potrebno je upisati ime baze, odabrati regiju poslužitelja instance i označiti tip instance koji će se koristiti.

New PostgreSQL

Read the docs [↗](#)

Name A unique name for your PostgreSQL instance. example-postgres-name

Database Optional
The PostgreSQL 'dbname' randomly generated unless specified

User Optional randomly generated unless specified

Region The region where your PostgreSQL instance runs. Frankfurt (EU Central)

PostgreSQL Version 15

Datadog API Key Optional
The API key to use for sending metrics to Datadog. Setting this will enable Datadog monitoring.

Instance Type

For hobby projects

Free \$0 / month	256 MB (RAM) 0.1 CPU
----------------------------	-------------------------

Slika 5.15: Stvaranje nove baze podataka

Nakon ovih koraka treba pokrenuti stvaranje pritiskom na tipku "Create". Prikazuje se prozor u kojemu su navedeni osnovni podaci o bazi kao ime, verzija, regija, prostor za pohranu, API ključ, itd. Kad je baza uspješno kreirana, potrebno je uzeti podatke koje treba dati backendu aplikacije. Ti podatci se nalaze u poljima *Hostname*, *Port*, *Database*, *Username*, *Password* i *External Database URL*.

Backend je implementiran na Render platformi, koristeći Docker slike koje se automatski generiraju i implementiraju prilikom svakog push-a na master granu. Proces implementacije upravlja se pomoću GitHub Actions radnog toka definiranog u datoteci *build-test-deploy.yaml*, koji obuhvaća korake poput preuzimanja koda, postavljanja JDK 17, izgradnje, testiranja, kopiranja JAR datoteke i prijenosa artefakta. Docker slika se označava tijekom procesa izgradnje.

```

1  name: Build Docker image
2
3  on:
4    push:
5      branches:
6        - master
7
8  jobs:
9    build-and-deploy:
10      name: Build docker image and deploy to hub
11      runs-on: ubuntu-latest
12
13    steps:
14      - name: Checkout code
15        uses: actions/checkout@v2
16
17      - name: Setup JDK 17
18        uses: actions/setup-java@v2
19        with:
20          distribution: adopt
21          java-version: 17
22
23      - name: Build
24        run: ./gradlew build
25
26      - name: Test
27        run: ./gradlew test
28
29      - name: Copy Jar file
30        run: mv build/libs/rehub-1.0.0.jar rehub-1.0.0.jar
31
32      - uses: actions/upload-artifact@master
33        with:
34          name: jar-file
35          path: rehub-1.0.0.jar
36
37      - name: Set up QEMU
38        uses: docker/setup-qemu-action@v2
39
40      - name: Set up Docker Buildx
41        uses: docker/setup-buildx-action@v2
42
43      - name: Login to Docker Hub
44        uses: docker/login-action@v2
45        with:
46          username: ${{ secrets.DOCKER_USERNAME }}
47          password: ${{ secrets.DOCKER_PASSWORD }}
48
49      - name: Set env
50        run: echo "DATE=$(date +%)\" >> $GITHUB_ENV
51
52      - name: Build and push
53        uses: docker/build-push-action@v3
54        with:
55          context: .
56          push: true
57          tags: begjosip/rehub-backend:latest

```

Slika 5.16: Datoteka *Datoteka build-test-deploy*

Konfiguracija backend-a upravlja se putem datoteke *application-prod.properties*, koja sadrži ključne parametre za konfiguraciju poslužitelja, prijenosa datoteka, podatkovnog izvora, e-pošte, ponovnog postavljanja lozinke, provjere korisnika, JPA i Flyway konfiguraciju te Jackson konfiguraciju. Svi ovi parametri su pažljivo prilagođeni kako bi podržali funkcionalnosti i integracije aplikacije, čime se osigurava glatki i učinkoviti razvojni i implementacijski proces backend-a.

```
application-prod.properties

1 # Server
2 server.port=8080
3 server.servlet.context-path=/api
4 spring.servlet.multipart.max-file-size=10MB
5 spring.servlet.multipart.max-request-size=10MB
6
7 therapy.scan.save.dir=/therapy_scans/
8
9 # Datasource
10 spring.datasource.driver-class-name=org.postgresql.Driver
11 spring.datasource.url=jdbc:postgresql://dpg-cmao87i1hb1s73cm7db0-a:5432/rehub_ocow
12 spring.datasource.username=rehub
13 spring.datasource.password=4a5f7seLiRKPRCoVun0TN5ETeG5QcvC
14
15 # Email
16 spring.mail.host=smtp.gmail.com
17 spring.mail.port=587
18 spring.mail.username=rehub.noreply@gmail.com
19 spring.mail.password=uviewxegrrthgbhf
20 spring.mail.properties.mail.smtp.auth=true
21 spring.mail.properties.mail.smtp.starttls.enable=true
22 spring.mail.test-connection=false
23
24 #Password Reset
25 password.reset.url=https://rehub.netlify.app/password/reset/
26
27 #User verification
28 user.verification.link=https://rehub-backend.onrender.com/api/v1/auth/user/verification/
29 login.page.url=https://rehub.netlify.app/login
30
31 # JPA
32 spring.jpa.show-sql=false
33 spring.jpa.open-in-view=false
34 spring.jpa.hibernate.ddl-auto=validate
35
36 #Flyway
37 spring.flyway.locations=classpath:db/migration,classpath:db/data
38
39 # Jackson
40 spring.jackson.serialization.write-dates-as-timestamps=false
41 spring.jackson.default-property-inclusion=non_null
```

Slika 5.17: Datoteka *application-prod.properties*

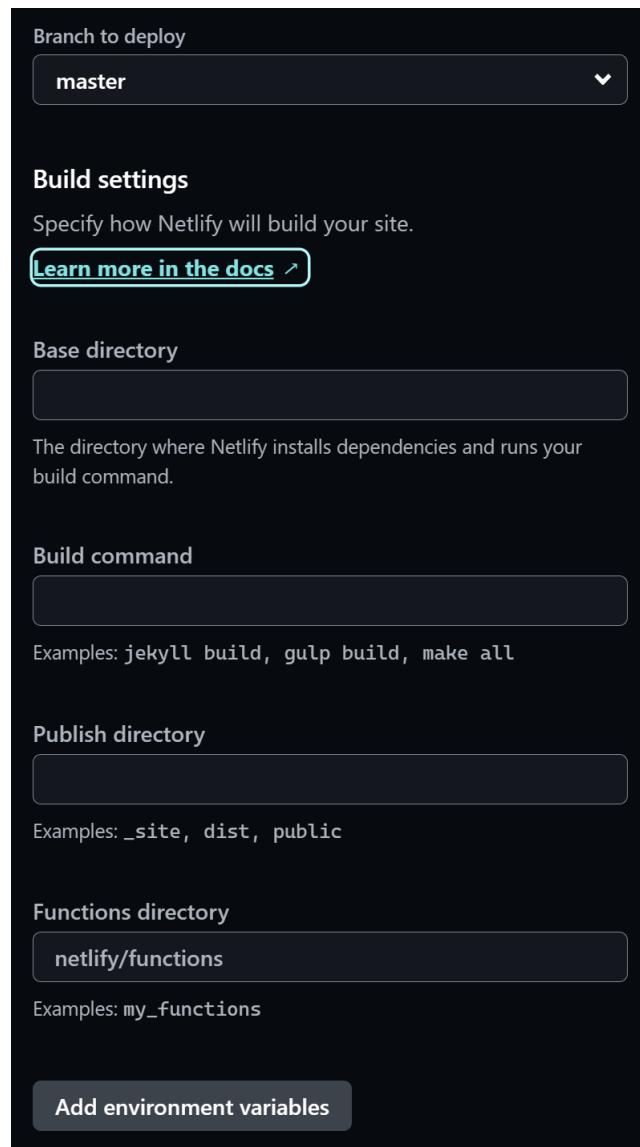
URL backend-a: <https://rehub-backend.onrender.com/>

Frontend aplikacija je postavljena za automatski deploy na Netlify prilikom push-a na master granu. U datoteci *build-deploy.yaml* specificiran je GitHub Actions radni tok za deploy na Netlify.

```
1   name: 'Netlify Deploy'
2
3   on:
4     push:
5       branches:
6         - master
7
8   jobs:
9     deploy:
10       name: 'Deploy to Netlify'
11       runs-on: ubuntu-latest
12
13   steps:
14     - uses: actions/checkout@v3
15     - uses: jsmrcaga/action-netlify-deploy@master
16       with:
17         NETLIFY_AUTH_TOKEN: ${{ secrets.NETLIFY_AUTH_TOKEN }}
18         NETLIFY_SITE_ID: ${{ secrets.NETLIFY_SITE_ID }}
19         NETLIFY_DEPLOY_TO_PROD: true
```

Slika 5.18: Datoteka *build-and-deploy.yaml*

U Netlify konzoli, u postavkama projekta, konfigurirajte *build* opcije.



Slika 5.19: Postavljanje konfiguracije

Base Directory: Postavite na direktorij u kojem se nalazi package.json. *Build Command:* Postavite na naredbu koja pokreće izgradnju projekta, u ovom slučaju npm run build. *Publish Directory:* Odredite direktorij koji sadrži izgrađene artefakte, u ovom slučaju build.

Aplikacija je puštena u pogon i spremna za uporabu. Pristupa se preko adrese koja je navedena za instancu frontenda. <https://rehub.netlify.app>

6. Zaključak i budući rad

Suočili smo se s izazovom razvoja web aplikacije za naručivanje medicinske rehabilitacije. Vrijednost koju smo najviše cijenili bila je suradnja u timu, koja je donijela raznolike izazove s kojima smo se uspješno nosili. Voditelj grupe imao je zahtjevan zadatak koordinacije sedmoro ljudi, što nije bilo jednostavno. Početni susret s mnogim novim konceptima zahtijevao je brzo učenje i intenzivan rad na aplikaciji i dokumentaciji.

Razumijevanje alata i tehnologija korištenih u izradi aplikacije i dokumentacije predstavljalo je dodatan izazov koji je zahtijevao značajan vremenski angažman. S vremenom proces je postao olakšan jer smo se bolje prilagodili timskom radu i aktivno si pružali podršku. Iako smo uvjereni da bi aplikacija mogla ostvariti još veći napredak s više vremena na raspolaganju, zadovoljni smo končanim rezultatom s obzirom na fakultetske i druge obaveze svakog člana tima.

Prijedlozi za budući rad obuhvaćaju proširenje funkcionalnosti aplikacije, kao što su dodatne opcije za korisnike, poboljšane analize statistike rehabilitacije te implementacija dodatnih sigurnosnih slojeva. Također, istraživanje mogućnosti integracije novih tehnologija i prilagodba sučelja prema povratnim informacijama korisnika predstavljaju ključne smjernice za daljnji razvoj "ReHub" aplikacije.

Kvalitetna komunikacija među članovima tima bila je ključna. Koristili smo alate poput WhatsAppa, Teamsa i Atlassian platforme. Redovito održavani sastanci služili su za raspravu o zadacima, pregled dosadašnjeg rada te dodjeljivanje novih zadataka. Sastanci su se održavali prema potrebi.

Na kraju, prepoznali smo da su opuštena atmosfera, međusobna pomoć i organizirano vođenje od strane voditelja ključni faktori za postizanje krajnjeg cilja.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>
4. Baeldung, <https://www.baeldung.com>
5. Tehnička predavanja tvrtke Croz, <https://gitlab.com/progi-deploy-demo>
6. React tutorial, <https://www.w3schools.com/REACT/DEFAULT.ASP>
7. Moodle - Programsko inženjertsvo, <https://moodle.fer.hr/course/view.php?id=527>

Indeks slika i dijagrama

3.1 Dijagram obrazaca uporabe	20
3.2 Sekvencijski dijagram za UC5/6/23	22
3.3 Sekvencijski dijagram za UC22	23
3.4 Sekvencijski dijagram za UC20	24
4.1 Dijagram arhitekture	29
4.2 Dijagram baze podataka	37
4.3 Dijagram razreda (Model)	39
4.4 Dijagram razreda (Repository)	40
4.5 Dijagram razreda (Service)	41
4.6 Dijagram razreda (Controller)	42
4.7 Dijagram enumeracija (Model)	42
4.8 Dijagram razreda (Config)	43
4.9 Dijagram razreda (Security)	43
4.10 Dijagram razreda (Request)	44
4.11 Dijagram razreda (Response)	45
4.12 Dijagram stanja (neprijavljeni korisnik)	46
4.13 Dijagram stanja (prijavljeni korisnik)	47
4.14 Dijagram stanja (zaposlenik)	47
4.15 Dijagram stanja (superadmin)	47
4.16 Dijagram aktivnosti	49
4.17 Dijagram komponenti	50
5.1 Test 1: Redovni slučaj	52
5.2 Rezultat testa 1: Redovni slučaj	52
5.3 Test 2: Rubni uvjet	53
5.4 Rezultat testa 2: Rubni uvjet	53
5.5 Test 3: Izazivanje pogrešaka	54
5.6 Rezultat testa 3: Izazivanje pogrešaka	54
5.7 Test 4: Ispitivanje neispravnog formata podataka	55
5.8 Rezultat testa 4: Ispitivanje neispravnog formata podataka	55

5.9 Test 5: Ispitivanje neispravnog formata podataka	56
5.10 Rezultat testa 5: Ispitivanje neispravnog formata podataka	56
5.11 Test 6: Ispitivanje neispravnog formata podataka	57
5.12 Rezultat testa 6: Ispitivanje neispravnog formata podataka	57
5.13 Specifikacijski dijagram razmještaja	58
5.14 Izbornik za stvaranje nove baze podataka	59
5.15 Stvaranje nove baze podataka	60
5.16 Datoteka <i>Datoteka build-test-deploy</i>	61
5.17 Datoteka <i>application-prod.properties</i>	62
5.18 Datoteka <i>build-and-deploy.yaml</i>	63
5.19 Postavljanje konfiguracije	64
6.1 Dijagram promjena na Frontendu	76
6.2 Dijagram promjena na Backendu	76
6.3 Dijagram promjena na dokumentaciji	76

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 16. listopada 2023.
- Prisustvovali: Kašik, Begić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Upoznavanje s kolegama

2. sastanak

- Datum: 18. listopada 2023.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Upoznavanje s projektnim zadatkom
 - Podjela zadataka
 - Dogovor oko načina daljnje komunikacije

3. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: Kašik, Krstičević
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

4. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: Begić, Greblo, Vivoda
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

5. sastanak

- Datum: 23. listopada 2023.
- Prisustvovali: Kašik, Krstičević
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

6. sastanak

- Datum: 24. listopada 2023.
- Prisustvovali: Begić, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

7. sastanak

- Datum: 25. listopada 2023.
- Prisustvovali: Kašik, Begić
- Teme sastanka:
 - Laboratorijska vježba

8. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Upoznavanje s pokretanjem backenda
 - Predstavljanje dizajna aplikacije

9. sastanak

- Datum: 30. listopada 2023.
- Prisustvovali: Kašik, Krstičević
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

10. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: Begić, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

11. sastanak

- Datum: 6. studeni 2023.
- Prisustvovali: Kašik, Krstičević
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

12. sastanak

- Datum: 7. studeni 2023.
- Prisustvovali: Begić, Mikulić, Vivoda, Greblo, Miletić

- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

13. sastanak

- Datum: 8. studeni 2023.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla
 - Dogovor vezan uz izradu dokumentacije
 - Laboratorijska vježba

14. sastanak

- Datum: 13. studeni 2023.
- Prisustvovali: Kašik, Krstičević
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

15. sastanak

- Datum: 14. studeni 2023.
- Prisustvovali: Begić, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla

16. sastanak

- Datum: 15. studeni 2023.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Dogovor oko daljnje raspodjele posla
 - Završni pregled dosad napravljenog posla

17. sastanak

- Datum: 6. prosinac 2023.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Laboratorijska vježba

18. sastanak

- Datum: 11. prosinac 2023.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:

- Dogovor oko daljnje raspodjele posla

19. sastanak

- Datum: 18. prosinac 2023.
- Prisustvovali: Kašik, Krstičević
- Teme sastanka:

- Dogovor oko daljnje raspodjele posla

20. sastanak

- Datum: 19. prosinac 2023.
- Prisustvovali: Begić, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:

- Dogovor oko daljnje raspodjele posla

21. sastanak

- Datum: 03. siječanj 2024.
 - Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
 - Teme sastanka:
- Dogovor oko daljnje raspodjele posla

22. sastanak

- Datum: 08. siječanj 2024.
 - Prisustvovali: Kašik, Krstičević
 - Teme sastanka:
- Dogovor oko daljnje raspodjele posla

23. sastanak

- Datum: 09. siječanj 2024.
 - Prisustvovali: Begić, Mikulić, Vivoda, Greblo, Miletić
 - Teme sastanka:
- Dogovor oko daljnje raspodjele posla

24. sastanak

- Datum: 10. siječanj 2024.
 - Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo
 - Teme sastanka:
- Dogovor oko daljnje raspodjele posla
 - Laboratorijska vježba

25. sastanak

- Datum: 17. siječanj 2024.

- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Laboratorijska vježba

26. sastanak

- Datum: 19. siječanj 2024.
- Prisustvovali: Kašik, Begić, Krstičević, Mikulić, Vivoda, Greblo, Miletić
- Teme sastanka:
 - Završni pregled napravljenog

Tablica aktivnosti

	Dora Kašik	Josip Begić	Roko Krstićević	Katarina Mikulić	Anton Vivoda	Erik Greblo	Marko Miletić
Upravljanje projektom	3	1					2
Opis projektnog zadatka							8
Funkcionalni zahtjevi				5			
Opis pojedinih obrazaca				2			
Dijagram obrazaca	2			3			
Sekvencijski dijagrami				4			
Opis ostalih zahtjeva				2			
Arhitektura i dizajn sustava		2	6				9
Baza podataka		5				7	
Dijagram razreda					2		
Dijagram stanja					5		
Dijagram aktivnosti				5			
Dijagram komponenti			3				
Korištene tehnologije i alati							5
Ispitivanje programskog rješenja	1	1	1	1	1	1	1
Dijagram razmještaja							6
Upute za puštanje u pogon				5			3
Dnevnik sastajanja							2
Zaključak i budući rad							2
Popis literature							2

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Dora Kašik	Josip Begić	Roko Krstičević	Katarina Mikulić	Anton Vivoda	Erik Greblo	Marko Miletić
<i>izrada plana stranica</i>	5						
<i>front end</i>	32	11	30		5	11	
<i>back end</i>		24		13	27	21	

Dijagrami pregleda promjena

U nastavku slijede dijagrami pregleda promjena nad datotekama izvornog koda i dokumentacije.



Slika 6.1: Dijagram promjena na Frontendu



Slika 6.2: Dijagram promjena na Backendu



Slika 6.3: Dijagram promjena na dokumentaciji