Programsko inženjerstvo

Ak. god. 2023./2024.

Medicinska rehabilitacija

Dokumentacija, Rev.1

Grupa: Proginator

Voditelj: Ema Badurina

Datum predaje: 17. studenog 2023.

Nastavnik: Miljenko Krhen

Sadržaj

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	**Opis promjene/dodatka**	**Autori**	**Datum**
0.1	Dovršen opis projektnog zadatka	E.Badurina	13.11.2023.
0.2	Dovršen opis arhitekture	L.Lasović	13.11.2023.
0.2.1	Dodan dio obrazaca uporabe	L.Lasović, E.Badurina	13.11.2023.
0.3	Dodani svi obrasci uporabe	L.Lasović, E.Badurina	14.11.2023.
0.3.1	Dodani dijagrami obrazaca uporabe	A.Jakovčević	14.11.2023.
0.3.2	Revizija i promjene na obrascima uporabe	A.Jakovčević, E.Badurina	15.11.2023.
0.4	Dodani sekvencijski dijagrami	L.Lasović, E.Badurina	15.11.2023.
0.4.1	Dodan opis baze	L.Lasović	15.11.2023.
0.4.1	Ispravak sekvencijskih dijagrama i ostali zahtjevi	E.Badurina, L.Lasović	16.11.2023.
0.5	Dijagrami razreda	L.Akmačić	17.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	17.11.2023.
1.0.1	Ispravak grešaka iz prvog ciklusa	E.Badurina	21.12.2023
1.1	Dijagram stanja	L.Lasović	18.1.2024

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	**Opis promjene/dodatka**	**Autori**	**Datum**
1.2	Dijagram komponenti	L.Lasović	18.1.2024
1.3	Dijagram aktivnosti	L.Lasović	19.1.2024
1.4	Korištene tehnologije i alati	E.Badurina	19.1.2024
1.5	Ispitivanje komponenti	L.Lasović, K.Vrdoljak	19.1.2024
1.6	Ispitivanje sustava	L.Lasović, K.Vrdoljak	19.1.2024
1.7	Upute za puštanje u pogon	L.Lasović, F.Zlatar	19.1.2024
1.8	Zaključak i budući rad	E.Badurina	19.1.2024
1.8.1	Konačne promjene na dokumentaciji	E.Badurina, L.Lasović	19.1.2024
2.0	Finalna verzija	*	19.1.2024

2. Opis projektnog zadatka

Cilj ovog projekta je izgradnja web aplikacije koja će omogućiti ljudima/pacijentima lakše prijavljivanje na slobodne termine za medicinsku rehabilitaciju i fizikalnu terapiju te praćenje njihovog zdravstvenog napretka. Također će omogućiti zaposlenicima ustanove da otkazuju ili mijenjaju termine ovisno o raspoloživosti prostorija, opreme i osoblja pritom imajući mogućnost uvida u pacjentove prošle terapije i napredak. Vrijeme provođenja rehabilitacije je svakim radnim danom od ponedjeljka do petka od 8:00 do 20:00 sati.
Aplikacija će razlikovati tri vrste korisnika:

- pacijenta
- liječnika - djelatnika zdravstvene ustanove
- administratora - djelatnika zdravstvene ustanove

Prilikom pokretanja web aplikacije svaki korisnik unosi svoju e-mail adresu i lozinku. Ovisno o vrsti korisnika biti će preusmjereni na različite stranice. Svaki korisnik imati će mogućnost mijenjanja nekih osobnih podataka i lozinke. Kao i promjene lozinke u slučaju da je zaboravljena.

Pacijent se samostalno prijavljuje u sustav. U slučaju da još ne postoji imati će opciju registracije. Za registraciju mora unijeti:

- ime
- prezime
- e-mail adresu
- MBO - Matični Broj Osiguranika
- broj telefona
- lozinku

Prilikom registracije pomoću MBO-a provjerava se ako korisnik postoji u središnjem informacijskom sustavu zdravstvene zaštite. Nakon prijave korisnik je preusmjeren na početnu stranicu gdje može prikazati svoje termine, terapije i prijavljivati iste. U terminima se prikazuju protekli i budući termini, za protekle termine piše ako je pacijent došao te komentari djelatnika ustanove o napretku. Termin se dobiva nakon što se odobri zahtjev za jednim. No prije nego pacijent može birati

termin mora kreirati terapiju, u slučaju da nema ni jednu aktivnu. Terapiju kreira unoseći podatke:

- ime i prezime liječnika koji ga je uputio na terapiju
- tip rehabilitacije
- opis oboljenja
- opis postupka liječenja

Nakon što preda podatke o terapiji, provjerava se status liječnika u imeniku liječnika. U slučaju da uneseni liječnik ne postoji, zbog krivo unesenih podataka sustav će upozoriti pacijenta o tome i imati će priliku ispraviti podatke. Nakon uspješno kreirane terapije pacijent može kreirati termin. Termin se sastoji od:

- željenog vremena
- vrste/tipa rehabilitacije
- reference na terapiju
- komentara za liječnika

Pacijent, nakon što je kreirao terapiju, unutar nje kreira termin odabirući pritom prostoriju i željeno vrijeme. Ukoliko se željeni termin nije popunio, termin će automatski biti odobren. Pacijentu će o odobrenom terminu, te o svim mogućim promjenama biti obaviješten e-mailom.

Liječnik nakon prijavljivanja u sustav ima pregled svih pacijenata i njihovih podataka. Imati će opciju pretraživanja pacijenta. Klikom na prikaz bit će mu prikazani svi termini odabranog pacijenta i detalji o njima, tu će imati opciju evidentirati dolazak pacijenta i zabilježiti komentare vezane uz napredak. Liječnik koji je evidentirao pacijenta dodati će se u sustav kao liječnik koji je vodio taj termin terapije. Također svaki djelatnik ustanove imati će mogućnost promjene termina, o čemu će pacijent biti obaviješten e-mailom.

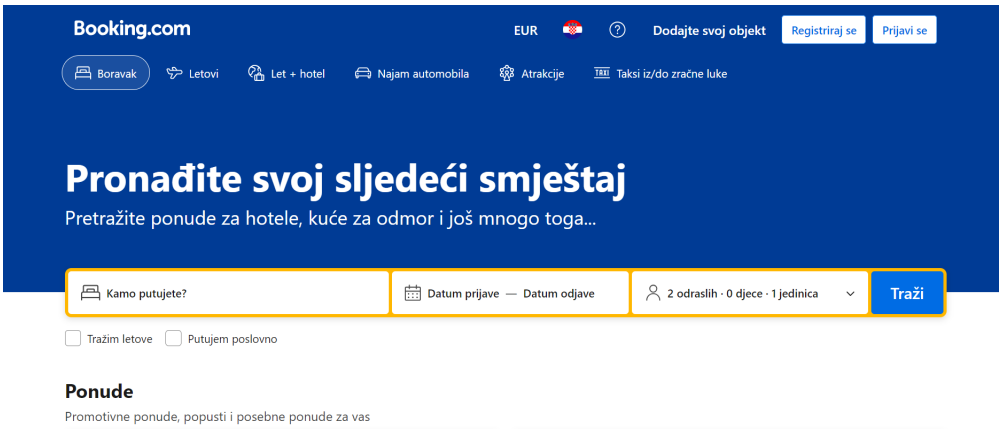
Administrator ima pregled svih pacijenata i djelatnika. Uz ovlasti koje imaju liječnici, administrator pri zaposlenju novog liječnika izrađuje korisnički račun za njega s inicijalnom lozinkom. Također nakon prestanka radnog odnosa administrator može ukloniti tog liječnika. Administrator definira sve što je potrebno za ispravan rad sustava, dakle može mijenjati dostupnost opreme i unositi novu opremu, mijenjati dostupnost prostorija i unositi nove. Također ako dođe do promjena u dostupnosti prostorija ili opreme.

—

Ovim projektom smanjio bi se opseg posla djelatnika ustanove i olakšao proces prijave na rehabilitaciju pacijentima, umjesto prijavljivanja u živo i rješavanja papirologije pacijentima će biti omogućeno automatsko prijavljivanje na slobodne termine, a liječnicima evidentiranje istih.

—

Odgovarajući primjer slične aplikacije na istu temu nije pronađen, moguće, zbog zatvorenosti ovakvog tipa aplikacije prema javnosti, ali primjer sa sličnim funkcijama iako ne u istu svrhu je booking.com (2.1). Na početnoj stranici *booking*-a nailazimo na opcije prijave i registracije, iako za početak korištenja same stranice to nije nužno kao kod ove aplikacije. Korisnik *booking*-a rezervira apartman, dok korisnik aplikacije za rehabilitaciju rezervira svoj termin za terapiju. Kao što korisnik na *booking*-u unosi grad u kojem želi rezervirati apartman, korisnik u ovoj aplikaciji odabire tip rehabilitacije za koju se želi prijaviti. U oba slučaja kod prijave na željeni termin ili rezervacije apartmana korisnik dobiva neke ponuđene termine/opcije ili može samostalno odabrati vrijeme željenog termina/rezervacije.



Slika 2.1: Primjer slične aplikacije

—

Aplikacija će se moći proširiti na način da su svi termini, prostorije, oprema i djelatnici vidljivi su administratoru koji im može mijenjati status za određeno vrijeme, na primjer kada liječnik ode na godišnji odmor može promijeniti njegov status u neaktivan za to razdoblje ili kada cijela ustanova ima sastanak onemogućiti sve termine za vrijeme tog sastanka. Također može se proširiti uvođenjem statistike koja bi ustanovi omogućila bolje korištenje resursa i nabavku novih. Primjerice ako je veća potražnja za kardiovaskularnim terapijama da povećaju broj opreme i liječnika specijaliziranih za taj tip terapije.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Pacijent
2. Djelatnik
   - Liječnik
   - Administrator
3. Razvojni tim

Pacijenti i djelatnici zajedničkim imenom su korisnici.

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
   (a) registrirati se u sustav, unijeti potrebne podatke(ime, prezime, e-mail adresu, MBO, broj telefona, lozinku)
2. Neprijavljeni korisnik (inicijator) može:
   (a) prijaviti se u sustav koristeći svoju e-mail adresu i lozinku
   (b) promijeniti zaboravljenu lozinku
3. Pacijent-*prijavljeni korisnik* (inicijator) može:
   (a) prikazati i mijenjati svoje osobne podatke i lozinku
   (b) prikazati svoje terapije (tip rehabilitacije, liječnika koji ga je uputio na rehabilitaciju, referencu na prošlu terapiju i status)
   (c) prikazati svoje termine (vrijeme termina, prostoriju, tip rehabilitacije, liječnika, napomene i ishod termina)
   (d) filtrirati svoje termine
   (e) prikazati nalaz (detalji i komentari) s odrađenog termina
   (f) naručiti se na terapiju, unijeti tip rehabilitacije, liječnika koji ga je uputio na terapiju, opis oboljenja i zahtijevani postupak liječenja

(g) odabrati termin za neku od aktivnih terapija
(h) promijeniti termin????

4. Djelatnik-*prijavljeni korisnik* (inicijator) može:
   (a) prikazati i mijenjati svoje osobne podatke i lozinku
   (b) vidjeti popis svih pacijenata (ime, prezime, MBO, e-mail adresa, broj telefona)
   (c) prikazati termine pojedinog pacijenta i prikazati evidenciju termina pojedinog pacijenta
   (d) evidentirati dolazak pacijenta na termin, dodati komentar o odrađenom terminu i vidjeti informacije o terapiji
   (e) promijeniti i otkazati zakazani termin
5. Administrator - *prijavljeni korisnik* (inicijator) može:
   (a) vidjeti popis svih djelatnika (ime, prezime, OIB, e-mail adresa)
   (b) urediti podatke djelatnika
   (c) ukloniti djelatnika
   (d) dodati/registrirati novog djelatnika (ime, prezime, e-mail adresa, OIB, lozinka)
   (e) mijenjati raspoloživost soba i opreme
   (f) dodavati opremu i sobe
6. Baza podataka (sudionik):
   (a) pohranjuje podatke i ovlasti korisnika
   (b) pohranjuje podatke o opremi i prostorijama
   (c) pohranjuje podatke o terapijama i terminima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijaviti se u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik unosi svoju e-mail adresu i lozinku
  2. Provjera ispravnosti podataka
  3. Prikaz početne stranice ovisno o korisniku
- **Opis mogućih odstupanja:**

  1.a Korisnik je zaboravio lozinku
  1. Korisnik odabire opciju ”Zaboravili ste lozinku?”
  2. Korisnik unosi e-mail adresu
  3. Nakon potvrde korisniku se šalje e-mail s novom lozinkom
  4. Korisnik ponovno unosi podatke

  2.a Neispravna e-mail adresa ili lozinka
  1. obavijest o neispravnosti unesenih podataka

UC2 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Registrirati se u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik unosi svoje podatke (ime, prezime, e-mail adresu, broj telefona, datum rođenja, MBO, lozinka)
  2. Provjera ispravnosti podataka verifikacijom iz baze podataka sustava zdravstvene zaštite
  3. Slanje maila za verifikaciju e-mail adrese
  4. Korisnik klikom na link verificira svoju e-mail adresu
  5. Prikaz stranice za prijavu u sustav
- **Opis mogućih odstupanja:**

2.a Korisnik ne postoji u sustavu zdravstvene zaštite
1. Sustav obavještava korisnika o neispravnosti podataka
2. Korisnik mijenja podatke ili odustaje od registracije

UC3 - pregled osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju ”Korisnički račun”
  2. Sustav prikazuje korisnikove osobne podatke

UC4 - Promjena broja telefona

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti broj telefona
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju ”Promijeni broj telefona”
  2. Korisnik unosi/mijenja broj telefona
  3. Korisnik odabire opciju ”Spremi”
  4. Povratak na prikaz osobnih podataka

UC5 - Promjena lozinke

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti lozinku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju ”Promijeni lozinku”
  2. Korisnik unosi staru lozinku
  3. Korisnik unosi novu lozinku
  4. Korisnik potvrđuje novu lozinku
  5. Korisnik odabire opciju ”Potvrdi”
  6. Lozinka se mijenja u bazi podataka

7. Povratak na prikaz podataka

- **Opis mogućih odstupanja:**

2.a korisnik unosi neispravnu staru lozinku

1. Sustav upozorava korisnika o neispravnosti lozinke
2. Korisnik ponovno mijenja lozinku ili odustaje od promjene

3.a Nova lozinka je jednaka staroj

1. Sustav upozorava korisnika da su mu lozinke iste
2. Korisnik mijenja novu lozinku ili odustaje od promjene

UC6 - Kreiranje terapije(3.4)

- **Glavni sudionik:** Pacijent
- **Cilj:** Kreiranje procesa terapije
- **Sudionici:** Baza podataka
- **Preduvjet:** Pacijent je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Pacijent odabire opciju ”Nova terapija”
  2. Otvara se stranica za kreiranje terapije
  3. Pacijent unosi ime i prezime liječnika, opis oboljenja i zahtijevani postupak liječenja, te odabire vrstu terapije
  4. Pacijent odabire opciju ”Potvrdi”
  5. Provjera ispravnosti podataka u imeniku liječnika
  6. Prikaz stranice za odabir termina
- **Opis mogućih odstupanja:**

3.a Liječnik ne postoji u imeniku liječnika

1. Sustav obavještava pacijenta o neispravnim podacima(ime i prezime liječnika)
2. Pacijent mijenja podatke ili odustaje od kreiranja terapije

UC7 - Odabir termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Poslati zahtjev za željenim terminom
- **Sudionici:** Baza podataka
- **Preduvjet:** Pacijent je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Pacijent odabire terapiju za koju se prijavljuje na termin

2. Otvara se stranica s popisom termina te terapije i gumbom za novi termin
3. Pacijent odabire opciju ”Novi termin”
4. Otvara se stranica za prijavu termina
5. Pacijent unosi prostoriju, datum i vrijeme
6. Pacijent šalje zahtjev klikom na ”Potvrdi”
7. Spremanje podataka u bazu i slanje maila s podatcima o terminu

- **Opis mogućih odstupanja:**

5.a Pacijentu ne odgovara niti jedan termin
1. Pacijent odustaje od prijave na termin

7.a Termin koji je pacijent odabrao se u međuvremenu popunio
1. Sustav ne sprema podatke u bazu i obaviještava korisnika o zauzetom terminu
2. Pacijent mijenja podatke ili odustaje od prijave termina

UC8 - Prikaz nalaza termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Pacijent vidi svoje termine
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen u sustav, postoje prošli termini
- **Opis osnovnog tijeka:**
  1. Pacijent odabire opciju prikaži nalaz
  2. Otvara se skočni prozor s informacijama i komentarom liječnika (nalazom)
  3. Pacijent odabire opciju zatvori u skočnom prozoru

UC9 - Prikaz pacijenta

- **Glavni sudionik:** Djelatnik
- **Cilj:** Prikazati termine pojedinog pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Djelatnik pretražuje pacijenta upisom imena ili prezimena u tražilicu
  2. Djelatnik odabire pacijenta
  3. Sustav otvara tablicu termina odabranog pacijenta
  4. Djelatnik odabire termin

5. Prikaz stranice termina
6. Djelatnik evidentira(UC10)/prikazuje evidenciju(UC8)/otkazuje termin
7. Djelatnik odabire opciju povratka

- **Opis mogućih odstupanja:**

1.a Upisani pacijent ne postoji u sustavu
   1. Prikaz teksta o ne postojanju pacijenta
   2. Djelatnik mijenja upisano ime ili prezime ili odustaje od pretraživanja

3.a Odabrani pacijent još nije prijavio ni jedan termin
   1. Prikaz teksta o ne postojanju termina za odabranog pacijenta
   2. Djelatnik odabire opciju povratka na popis pacijenata

UC10 - Evidencija dolaska(3.5)

- **Glavni sudionik:** Djelatnik
- **Cilj:** Evidentirati dolazak pacijenta na termin
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Pacijent mora imati termin koji nije evidentiran
- **Opis osnovnog tijeka:**
  1. Djelatnik odabire termin koji će evidentirati
  2. Otvara se stranica termina
  3. Djelatnik odabire opciju ”Evidentiraj”
  4. Otvara se obrazac za evidenciju dolaska na termin
  5. Djelatnik evidentira dolazak odabirom jedne od tri ponuđene opcije(nije došao, došao je, otkazano)
  6. Djelatnik unosi komentar o odrađenom terminu i napredku
  7. Djelatnik opcionalno završava terapiju
  8. Djelatnik predaje evidenciju
  9. Povratak na stranicu termina
- **Opis mogućih odstupanja:**

3.a Djelatnik odustaje od evidentiranja termina
   1. Djelatnik odabire opciju povratka na popis termina odabranog pacijenta

7.a Djelatnik predaje evidenciju bez popunjene opcije ili komentara
   1. Sustav upozorava djelatnika o nepopunjenim poljima
   2. Djelatnik popunjava obrazac ili odustaje od evidencije

UC11 - Registracija korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Registrirati novog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Djelatnik odabire opciju ”Novi račun”
  2. Prikazuje se stranica za registraciju novog korisnika
  3. Administrator unosi podatke o korisniku(ime, prezime, broj telefona, datum rođenja, OIB/MBO, e-mail adresu i inicijalnu lozinku)
  4. Administrator odabire opciju ”Dodaj korisnika”
  5. Provjera podataka i unos u bazu podataka
- **Opis mogućih odstupanja:**

  3.a Podatci ne zadovoljavaju željeni format
  1. Sustav upozorava administratora o neispravnom formatu podataka
  2. Administrator mijenja podatke ili odustaje od registracije djelatnika

  5.a Korisnik već postoji
  1. Sustav upozorava administratora o već zauzetim podacima
  2. Administrator mijenja podatke ili odustaje od registracije korisnika

UC12 - Uređivanje podataka djelatnika

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti podatke djelatnika
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
  - Postoji barem jedan liječnik/djelatnik
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Uredi” pokraj imena djelatnika
  2. Prikazuje se stranica s podacima odabranog djelatnika
  3. Administrator mijenja podatke o djelatniku(ime, prezime, OIB, e-mail adresu, broj telefona, godinu rođenja)

4. Administrator odabire opciju ”Potvrdi”
5. Podatci se mijenjaju u bazi podataka

- **Opis mogućih odstupanja:**

3.a Podatci ne zadovoljavaju željeni format
   1. Sustav upozorava administratora o neispravnom formatu podataka
   2. Administrator mijenja podatke ili odustaje od promjene podataka

4.a Administrator ne želi promijeniti podatke
   1. Administrator odabire opciju ”Odustani”
   2. Podatci se ne mijenjaju u bazi podataka
   3. Prikazuje se popis djelatnika

UC13 - Promjena ovlasti djelatnika

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti ovlasti djelatnika
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
  - Postoji barem jedan liječnik/djelatnik
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Uredi” pokraj imena djelatnika
  2. Prikazuje se stranica s podacima odabranog djelatnika
  3. Administrator pridjeljuje/uklanja administratorske ovlasti djelatnika
  4. Administrator odabire opciju ”Potvrdi”
  5. Otvara se skočni prozor s upozorenjem i zahtjevom za unos lozinke administratora
  6. Administrator unosi lozinku
  7. Administrator odabire opciju ”potvrdi”
  8. Sustav potvrđuje lozinku i sprema promjene u bazu podataka
  9. Vraća se na podatke o djelatniku
- **Opis mogućih odstupanja:**

10.a Administrator je unio neispravnu lozinku
   1. Sustav upozorava administratora o unosu neispravne lozinke
   2. Administrator ponovno unosi lozinku ili odustaje od promjene ovlasti

UC14 - Uklanjanje djelatnika

- **Glavni sudionik:** Administrator
- **Cilj:** Ukloniti djelatnika
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
  - Postoji barem jedan liječnik/djelatnik
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Ukloni” pokraj imena djelatnika
  2. Otvara se skočni prozor s upozorenjem i zahtjevom za unos lozinke administratora
  3. Administrator unosi lozinku
  4. Administrator odabire opciju ”Potvrdi”
  5. Sustav potvrđuje lozinku
  6. U bazi podataka se briše djelatnik
  7. prikazuje se popis djelatnika bez upravo uklonjenog djelatnika
- **Opis mogućih odstupanja:**

  5.a Administrator je unio neispravnu lozinku
  1. Sustav upozorava administratora o unosu neispravne lozinke
  2. Administrator ponovno unosi lozinku ili odustaje od uklanjanja djelatnika

UC15 - Unos opreme

- **Glavni sudionik:** Administrator
- **Cilj:** Unos opreme
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Inventar”
  2. Otvara se stranica s popisom opreme
  3. Administrator odabire opciju ”Dodaj opremu”
  4. Otvara se skočni prozor s prostorom za unos podataka o opremi
  5. Administrator unosi podatke o opremi (vrstu i prostoriju u kojoj se nalazi)

6. Odabire prostoriju u kojoj će se oprema nalaziti
7. Administrator odabire opciju ”Spremi”
8. Podatci se spremaju u bazu podataka i generira se jedinstveni identifikator opreme
9. Prikazuje se stranica s popisom opreme

- **Opis mogućih odstupanja:**

6.a Administrator želi odustati od unosa opreme

1. Administrator odabire opciju ”odustani”

UC16 - Promjena dostupnosti opreme

- **Glavni sudionik:** Administrator
- **Cilj:** Promjeniti dostupnost opreme
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Inventar”
  2. Otvara se stranica s popisom opreme
  3. Administrator odabire opremu
  4. Administrator mijenja dostupnost opreme odabirom opcije ”Dostupno”/”Nedostupno”
  5. Promjena dostupnosti odabrane opreme u bazi podataka

UC17 - Unos prostorija

- **Glavni sudionik:** Administrator
- **Cilj:** Unijeti nove prostorije
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Prostorije”
  2. Otvara se stranica s popisom prostorija
  3. Administrator odabire opciju ”Nova soba”
  4. Prikazuje se stranica s prostorom za unos podataka o prostoriji

5. Administrator unosi broj i kapacitet prostorije, te vrstu terapija za koje je vezana
6. Administrator odabire opciju ”Potvrdi”
7. Podatci se spremaju u bazu podataka
8. Prikazuje se stranica s popisom opreme

- **Opis mogućih odstupanja:**

6.a Administrator je unio već postojeći broj prostorije
1. Sustav upozorava administratora o neispravnosti podataka
2. Administrator mijenja podatke ili odustaje od unosa prostorije

6.b Administrator želi odustati od unosa prostorije
1. Administrator odabire opciju ”Odustani”

UC18 - Promjena podataka prostorije

- **Glavni sudionik:** Administrator
- **Cilj:** Promjeniti podatke o prostoriji
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju ”Prostorije”
  2. Otvara se stranica s popisom prostorija
  3. Administrator pretražuje prostorije
  4. Administrator odabire prostoriju
  5. Administrator odabire opciju ”Uredi”
  6. Administrator mijenja dostupnost i vrste terapije za koje je prostorija vezana
  7. Administrator odabire opciju ”Potvrdi”
  8. Promjena podataka odabrane prostorije u bazi podataka

UC19 - Obavijest o promjeni termina

- **Glavni sudionik:** Djelatnik
- **Cilj:** Unijeti podatke o novom terminu i obavijestiti pacijenta o tome putem e-maila
- **Sudionici:** Baza podataka
- **Preduvjet:**

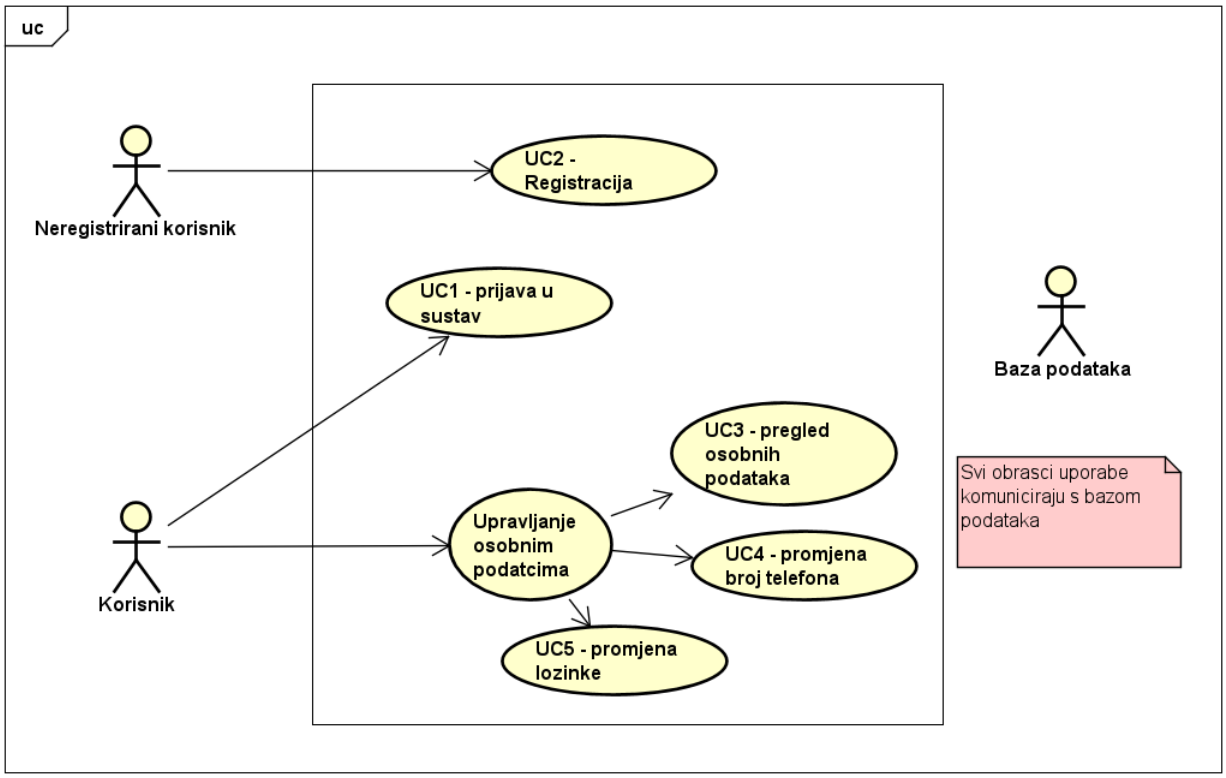
- Djelatnik je prijavljen u sustav
- Djelatnik je na stranici odabranog zakazanog termina

- **Opis osnovnog tijeka:**
  1. Djelatnik odabire opciju ”Premjesti termin”
  2. Otvara se obrazac za premještanje termina
  3. Djelatnik unosi vrijeme novog termina
  4. Administrator odabire opciju ”Potvrdi”
  5. Podaci o terminu se spremaju u bazu podataka
  6. Odabranom pacijentu se šalje e-mail s informacijama o novom terminu
- **Opis mogućih odstupanja:**
  4.a Djelatnik je odabrao već zauzeti termin
    1. Sustav upozorava djelatnika o zauzetom terminu
    2. Djelatnik mijenja termin ili odustaje od promjene termina

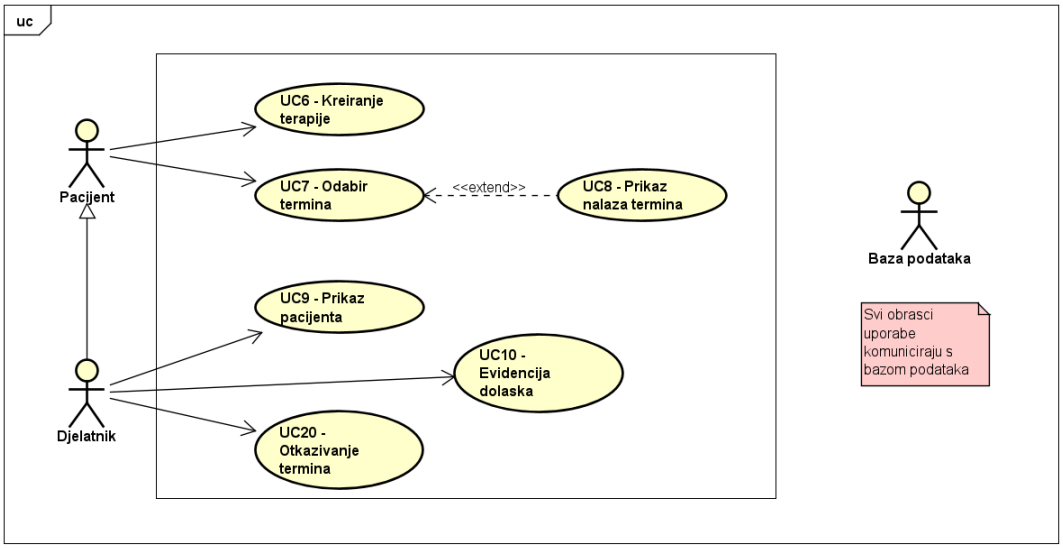
UC20 - Otkazivanje termina

- **Glavni sudionik:** Djelatnik
- **Cilj:** Otkazati termin
- **Sudionici:** Baza podataka
- **Preduvjet:**
  - Djelatnik je prijavljen u sustav
  - Djelatnik je na stranici odabranog zakazanog termina
- **Opis osnovnog tijeka:**
  1. Djelatnik odabire opciju ”Otkaži termin”
  2. Otvara se obrazac za otkazivanje termina
  3. Djelatnik unosi komentar za pacijenta
  4. Djelatnik odabire opciju ”Potvrdi”
  5. Podaci o terminu se spremaju u bazu podataka
  6. Odabranom pacijentu se šalje e-mail s informacijama o otkazanom terminu
- **Opis mogućih odstupanja:**
  4.a Djelatnik nije napisao komentar
    1. Sustav upozorava djelatnika o nenapisanom komentaru
    2. Djelatnik dodaje komentar ili odustaje od otkazivanja termina

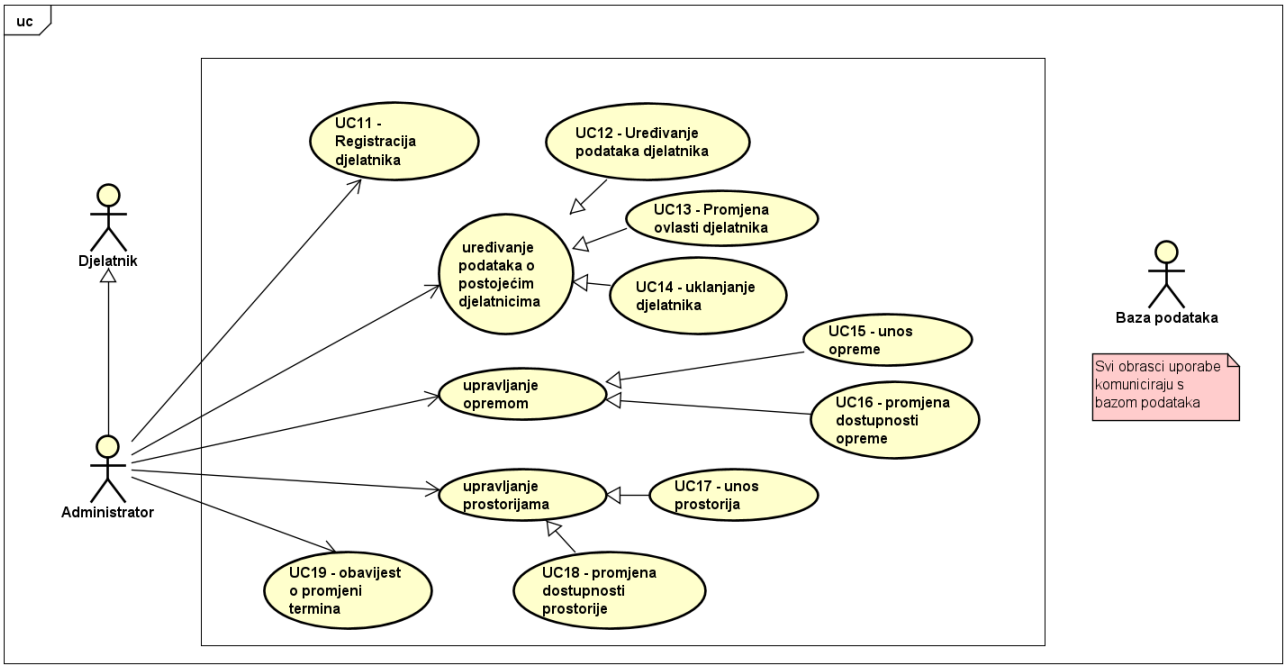
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe - Korisnik i neregistrirani korisnik



Slika 3.2: Dijagram obrasca uporabe - Djelatnik i pacijent

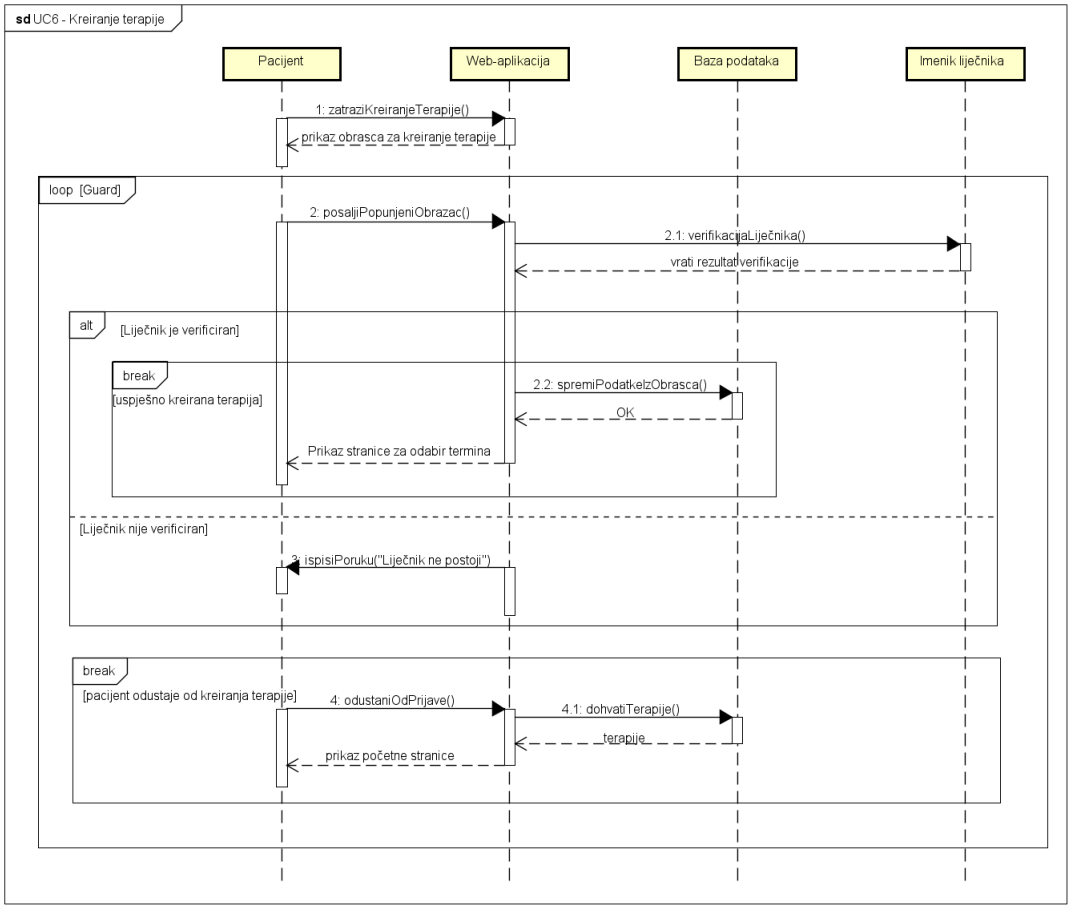


Slika 3.3: Dijagram obrasca uporabe - Administrator

3.1.2 Sekvencijski dijagrami

UC6 - Kreiranje terapije

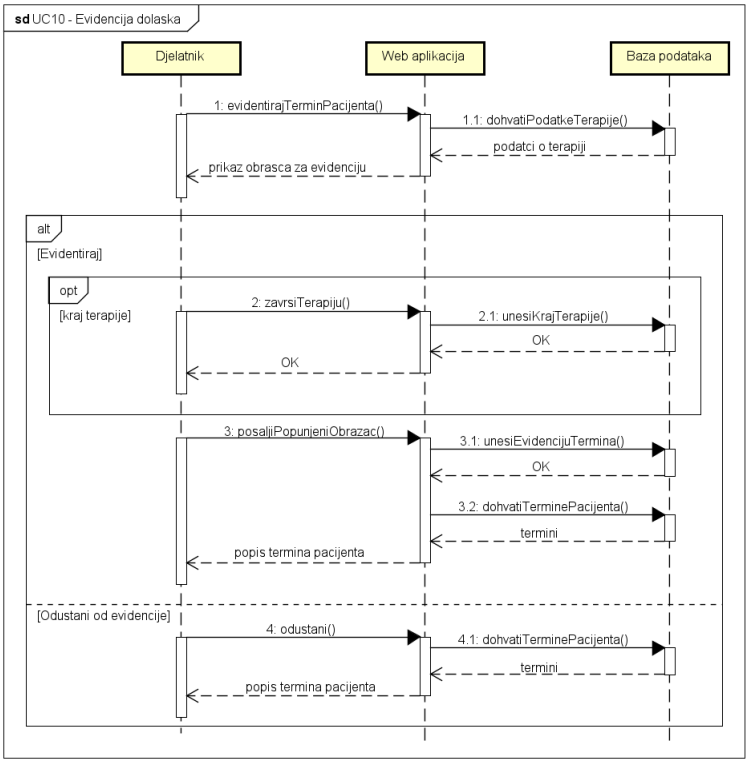
Pacijent šalje zahtjev za kreiranjem terapije. Web aplikacija mu prikazuje obrazac koji mora ispuniti kako i se naručio na terapiju. Pacijent unosi potrebne podatke(ime i prezime liječnika koji ga je uputio na rehabilitaciju, datum početka terapije, vrstu terapije, opis oboljenja i zahtijevani postupak liječenja). Pacijent šalje popunjeni obrazac. Web aplikacija na temelju primljenih podataka provjerava kredibilitet liječnika u imeniku liječnika. Ovisno o rezultatu promjene web aplikacija, ako liječnik postoji u imeniku liječnika, sprema sve podatke u bazu podataka i prikazje stranicu za odabir termina. U slučaju da liječnik ne postoji u imeniku liječnika aplikacija ispisuje poruku "Liječnik ne postoji", pacijent u tom slučaju ponovno unosi to jest mijenja podatke i ponovno šalje obrazac. Pacijent može odustati od kreiranja terapije, pri čemu će mu se prikazati početna stranica.



Slika 3.4: Sekvencijski dijagram za UC6

UC10 - Evidencija dolaska

Djelatnik odabire opciju evidentiraj na stranici odabranog termina. Dohvaća podatke o terapiji, te nakon toga prikazuje obrazac za evidenciju s dohvaćenim podacima, prostorom za označavanje dolaska i unosom komentara. Djelatnik odabire status(odrađen, propušten ili otkazan) i piše komentar, nakon toga potvrđuje evidenciju i šalje popunjeni obrazac, web aplikacija potom unesene podatke sprema u bazu podataka izmjenjujući tako podatke o odabranom terminu. Nakon što su podaci uspješno uneseni web aplikacija traži od baze termine odabranog pacijenta i prikazuje stranicu s popisom termina tog pacijenta. U slučaju da je djelatnik odabrao krivi termin za evidenciju može odustati od evidentiranja, web aplikacija će dohvatiti podatke o terminima i prikazati stranicu s popisom termina odabranog pacijenta.



Slika 3.5: Sekvencijski dijagram za UC10

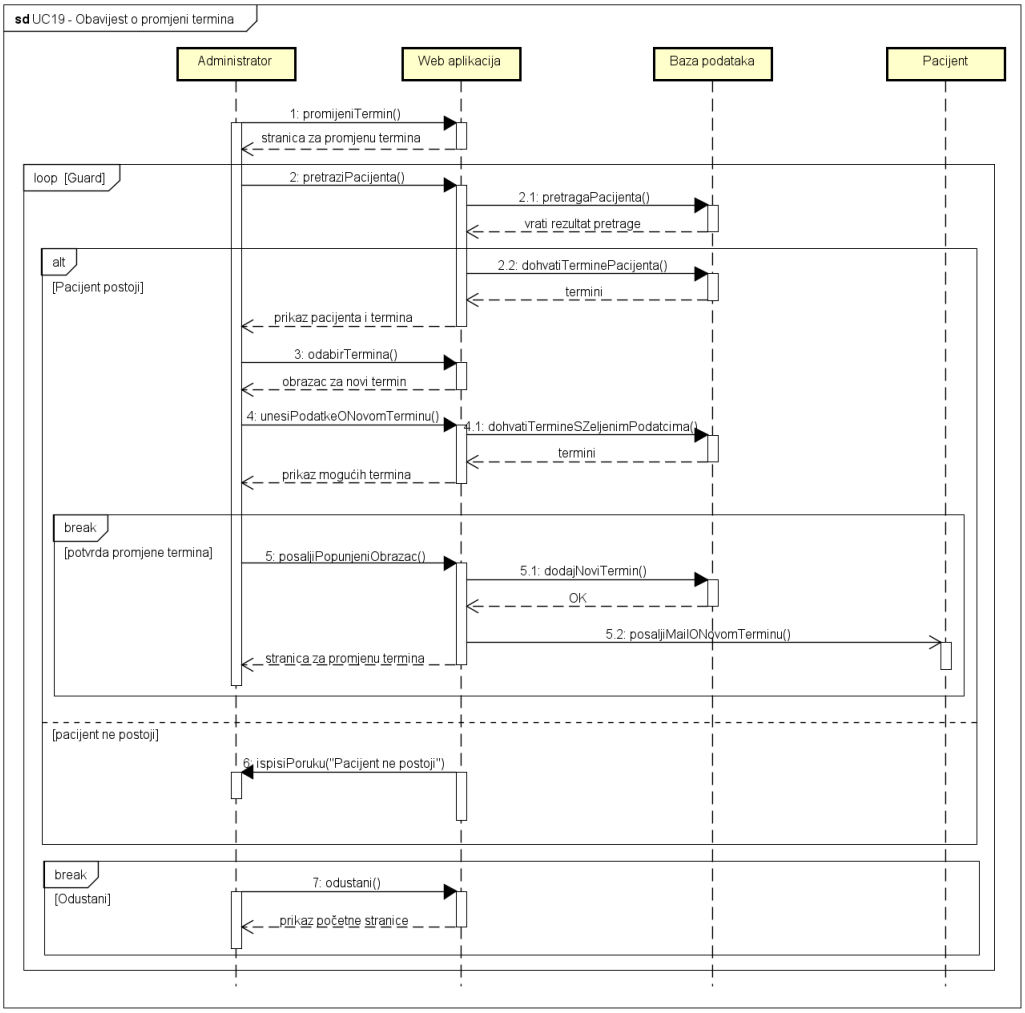
UC19 - Obavijest o promjeni termina

Administrator na svojoj stranici odabire opciju ”Promjeni termin” te mu nakon toga web aplikacija otvara stranicu za promjenu termina. Administrator unosi pacijenta i pretražuje. Web aplikacija prima podatak i provjerava u bazi podataka ako pacijent postoji. Ako postoji web aplikacija odmah dohvaća neevidentirane termine odabranog pacijenta, te ih prikazuje administratoru. Administrator

potom odabire termin koji želi promijeniti i web aplikacija mu pokazuje obrazac za dodjelu novog termina. Administrator unosi podatke o novom terminu (datum/soba/oprema) i aplikacija na temelju toga dohvaća moguće termine iz baze podataka. Administrator odabire neki od mogućih termina i opcionalno dodaje komentar. Nakon što administrator potvrdi odabrani termin web aplikacija dodaje termin u bazu podataka i šalje e-mail pacijentu s informacijama o promjeni termina, a administratoru se ponovno prikazuje stranica za promjenu termina.

U slučaju da pacijent kojeg je administrator unio ne postoji u bazi podataka aplikacija će ga obavijestiti o nepostojanju pacijenta. Administrator može ponovno unijeti pacijenta i ponoviti pretragu.

Administrator može odustati od promjene termina nakon čega mu se prikazuje početna stranica.



Slika 3.6: Sekvencijski dijagram za UC19

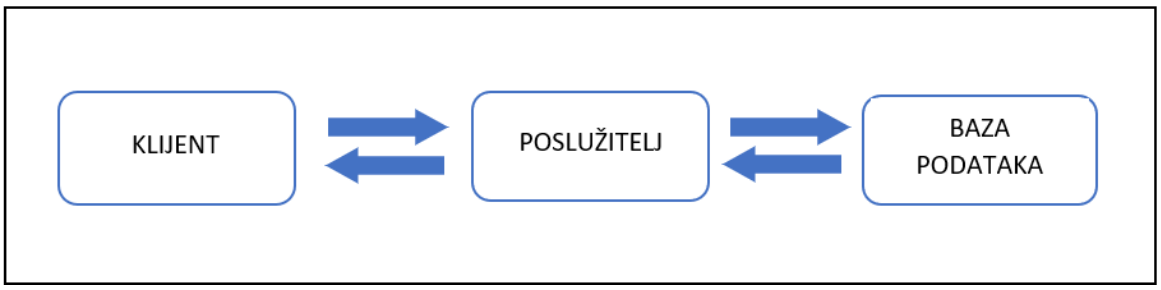
3.2 Ostali zahtjevi

- Sustav mora omogućiti istovremeni pristup više korisnika (100 korisnika istovremeno) uz očuvanje performansi.
- Poslužitelj mora vratiti odgovor na korisnički zahtjev unutar 5 sekundi.
- Sustav mora biti dostupan korisnicima barem 99,99% vremena unutar jedne godine.
- Lozinka prije spremanja u bazu podataka mora biti kriptirana.
- Veza s bazom podataka mora biti brza i otporna na greške.
- Mora biti omogućen pristup sustavu iz javne mreže uz određenu razinu sigurnosti (HTTPS).
- Klijentska aplikacija mora raditi u pregledniku Google Chrome.
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost aplikacije.
- Korisničko sučelje mora podržavati hrvatski jezik.
- Korisničko sučelje mora biti jednostavno za upotrebu (intuitivno).

4. Arhitektura i dizajn sustava

Sustav se može podijeliti na podsustave:

- Klijent
- Poslužitelj
- Baza podataka



Slika 4.1: Organizacija sustava

Klijent je web preglednik pomoću kojeg korisnici pristupaju našoj web aplikaciji. Često korišteni web preglednici su: Google Chrome, Apple Safari, Mozilla Firefox. Kada korisnik pristupa web aplikaciji, web preglednik šalje HTTP (*engl. Hypertext Transfer Protocol*) zahtjeve za preuzimanje statičkih datoteka web poslužitelju. Statičke datoteke mogu biti HTML, CSS i JavaScript (React.js) datoteke. Nakon preuzimanja datoteka, preglednik ih koristi za izgradnju i prikaz korisničkog sučelja te izvršavanje funkcija unutar aplikacije.

Poslužitelj je Python aplikacija pisana unutar web mikrookvira Flask. On služi kao posrednik između klijenta (korisničkog sučelja) i baze podataka. U Python aplikaciji (Flask aplikaciji) definirani su RESTful API-ji (kao rute ili krajnje točke) koji omogućuju klijentima da šalju HTTP zahtjeve za određenim podacima. Prilikom obrade tih zahtjeva, Python aplikacija šalje upite bazi podataka kako bi dohvatila, promijenila ili dodala željene podatke.

Kada baza podataka zaprimi upite od poslužitelja, ona ih izvršava i vraća rezultate poslužitelju. Vraćeni rezultati mogu biti u obliku potvrde o izvršenom upitu ili u obliku podataka koje je poslužitelj zatražio od baze podataka. Baza podataka ostaje pasivna sve dok ne zaprimi nove upite od poslužitelja.

Aplikacija je organizirana po MVC (*engl. Model-View-Controller, hrv. Model-Pogled-Nadglednik*) obrascu.

Općenito, *Model* definira podatke, njihovu strukturu i operacije koje se mogu izvršavati nad tim podacima. *Pogled* predstavlja prikaz podataka korisniku (npr. korisničko sučelje). *Nadglednik* predstavlja posrednike između Modela i Pogleda. Oni obrađuju zahtjeve korisnika, vrše operacije nad Modelom i ažuriraju Pogled. Organizacija po ovom obrascu olakšava proširivanje i održavanje aplikacije.

Naša aplikacija ne sadrži doslovno sve tri navedene komponente. Ovdje Python objedinjuje Model i dio logike Nadglednika. Komponente u React-u odgovaraju Pogledu te je i ovdje sadržan dio logike Nadglednika.

4.1 Baza podataka

Za rješavanje projektnog zadatka odabrana je relacijska baza podataka. Implementirali smo je korištenjem sustava PostgreSQL. To je besplatan sustav za upravljanje bazom podataka otvorenog koda. Objekti relacijske baze podataka su relacije. Neformalno, i u PostgreSQL-u, to su dvodimenzionalne imenovane tablice gdje imenovani stupac predstavlja atribut, a redak zapis relacije. Izbor ovakve baze podataka omogućio nam je lakše strukturiranje podatka i definiranje veza između njih, normalizaciju, skalabilnost te sigurnost kod pohrane i dohvata podataka. Zbog potreba naše aplikacije, baza podataka sadrži entitete:

- Korisnik
- Djelatnik
- Pacijent
- Terapija
- VrstaTerapije
- Termin
- Status
- Soba
- Uredaj
- VrstaUredaja
- SobaZa

4.1.1 Opis tablica

Korisnik:

Entitet Korisnik sadrži sve bitne informacije o korisniku aplikacije. Entitet sadrži atribute: idKorisnika, ime, prezime, datumRodenja, telefon, email, lozinka, potvrden, potvrdenNa. Atribut idKorisnika je primarni ključ. Atributi email i telefon su alternativni ključevi. Korisnik može biti ili djelatnik zdravstvene ustanove ili pacijent koji se želi naručiti na terapiju u zdravstvenoj ustanovi.

Korisnik		
idKorisnika	INT	jedinstveni identifikator korisnika
ime	VARCHAR(50)	ime korisnika
prezime	VARCHAR(50)	prezime korisnika
datumRodenja	DATE	datum rođenja korisnika
telefon	VARCHAR(20)	broj mobilnog telefona korisnika
email	VARCHAR(100)	e-mail adresa korisnika
lozinka	VARCHAR(100)	hash lozinke za prijavu u aplikaciju
potvrden	BOOLEAN	status potvrde e-maila korisnika (e-mail je ili nije potvrđen)
potvrdenNa	TIMESTAMP	datum i vrijeme potvrde e-maila korisnika

Djelatnik:

Entitet Djelatnik je ekskluzivna specijalizacija entiteta Korisnik. Entitet sadrži atribute: idKorisnika, jeAktivan, jeAdmin, OIB. Djelatnik je povezan s Korisnikom preko atributa idKorisnika. Atribut OIB je alternativni ključ.

Djelatnik		
idKorisnika	INT	jedinstveni identifikator korisnika (korisnik.idKorisnika)
OIB	CHAR(11)	osobni identifikacijski broj pacijenta
jeAktivan	BOOLEAN	označava radni odnos liječnika i zdravstvene ustanove (radi = true, ne radi = false)
jeAdmin	BOOLEAN	označava je li djelatnik administrator (true = je, false = nije)

Pacijent:

Entitet Pacijent je ekskluzivna specijalizacija entiteta Korisnik. Entitet sadrži atribute: idKorisnika, MBO. Pacijent je povezan s Korisnikom preko atributa idKorisnika (korisnik.idKorisika). Atribut MBO je alternativni ključ.

Pacijent		
idKorisnika	INT	jedinstveni identifikator korisnika (korisnik.idKorisnika)
MBO	CHAR(9)	matični broj osiguranika (pacijenta)

Terapija:

Entitet Terapija sadrži sve bitne informacije o terapiji na koju je pacijent naručen. Entitet sadrži atribute: idTerapije, idLijecnika, opisOboljenja, zahtPostLijec, datumPoc, datumZavrs, idPacijenta, idVrste. Atributi zahtPostLijec, datumPoc, datumZavrs i idVrste su opcionalni. Entitet Terapija je povezan binarnom N:1 vezom s entitetom Pacijentom preko atributa idPacijenta. Entitet Terapija je povezan binarnom N:1 vezom s entitetom VrstaTerapije preko atributa idVrste.

Terapija		
idTerapije	INT	jedinstveni identifikator terapije
idLijecnika	INT	jedinstveni identifikator liječnika
opisOboljenja	VARCHAR(300)	opis oboljenja pacijenta
zahtPostLijec	VARCHAR(300)	zahtjev za postupkom liječenja (terapijom)
datumPoc	DATETIME	datum početka terapije
datumZavrs	DATETIME	datum završetka terapije
idPacijenta	INT	jedinstveni identifikator pacijenta (pacijent.idKorisnika)
idVrste	INT	jedinstveni identifikator vrste uređaja (vrstaTerapije.idVrste)

VrstaTerapije:

Entitet VrstaTerapije sadrži sve bitne informacije o vrsti terapije na koju je pacijent naručen. Entitet sadrži atribute: idVrste, imeVrste, opisVrste. Atribut idVrste je primarni ključ. Atribut opisVrste je opcionalan. Entitet VrstaTerapije povezan je

binarnom N:N vezom s entitetom Soba.

VrstaTerapije		
idVrste	INT	jedinstveni identifikator vrste terapije
imeVrste	VARCHAR(50)	naziv vrste terapije
opisVrste	VARCHAR(300)	opis vrste terapije

Termin:

Entitet Termin sadrži sve bitne informacije o terminu terapije na koji je pacijent naručen. Entitet Termin ne postoji bez entiteta vlasnika, entiteta Terapija (Termin je egzistencijalno slab entitet). Atribut idTermina je primarni ključ, dok su idTerapije, brSobe, idStatus i idDjelatnika strani ključevi. Atributi do, komentar i idDjelatnika su opcionalni. Entitet Termin povezan je binarnom N:1 vezom s entitetom Terapija preko atributa idTerapije. Entitet Termin povezan je binarnom N:1 vezom s entitetom Soba preko atributa brSobe. Entitet Termin povezan je binarnom N:1 vezom s entitetom Status preko atributa idStatus. Entitet Termin povezan je binarnom N:0..1 vezom s entitetom Djelatnik preko atributa idKorisnika. *Napomena: Primarni ključ entiteta Termin je kompozitni ključ (idTermina, idTerapije).*

Termin		
idTermina	INT	jedinstveni identifikator termina terapije
idTerapije	INT	jedinstveni identifikator terapije (terapija.idTerapije)
idDjelatnika	INT	jedinstveni identifikator djelatnika (djelatnik.idKorisnika)
od	TIMESTAMP	datum i vrijeme početka termina
do	TIMESTAMP	datum i vrijeme završetka termina
komentar	VARCHAR(300)	komentar liječnika o napretku terapije
brSobe	VARCHAR(10)	jedinstveni identifikator sobe (soba.brSobe)
idStatus	INT	jedinstveni identifikator statusa (status.idStatus)

Soba:

Entitet Soba sadrži sve bitne informacije o sobi u kojoj se provodi neka terapija. Entitet sadrži atribute: brSobe, kapacitet, uUporabi. Atribut brSobe je primarni ključ. Entitet Soba povezan je binarnom N:N vezom s entitetom VrstaTerapije.

Soba		
brSobe	VARCHAR(10)	jedinstveni identifikator sobe
kapacitet	INT	kapacitet sobe, broj pacijenata koji istovremeno mogu biti na terapiji u nekoj sobi
uUporabi	BOOLEAN	označava je li soba popunjena (true = popunjena, false = nije popunjena)

SobaZa:

Entitet SobaZa sadrži sve bitne informacije o sobi u kojoj se odvija točno određena vrsta terapije. Entitet SobaZa sadrži atribute brSobe i idVrste. Atributi brSobe i idVrste su strani ključevi.

SobaZa		
brSobe	INT	jedinstveni identifikator sobe (soba.brSobe)
idVrste	INT	jedinstveni identifikator vrste terapije (vrstaTerapije.idVrste)

Uredaj:

Entitet Uredaj sadrži sve bitne informacije o uređaju. Entitet sadrži atribute: idUredaja, brSobe, idVrste. Atribut idUredaja je primarni ključ, dok su atributi brSobe i idVrste strani ključevi. Atribut brSobe je opcionalan. Entitet Uredaj povezan je binarnom N:0..1 vezom s entitetom Soba preko atributa brSobe. Entitet Uredaj povezan je binarnom N:1 vezom s entitetom VrstaUredaja preko atributa idVrste.

Uredaj		
idUredaja	INT	jedinstveni identifikator uređaja
brSobe	VARCHAR(10)	jedinstveni identifikator sobe (soba.brSobe)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Uredaj		
idVrste	INT	jedinstveni identifikator vrste uređaja (vrstaUredaja.idVrste)

VrstaUredaja:

Entitet VrstaUredaja sadrži sve bitne informacije o vrsti uređaja. Entitet sadrži atribute: idVrste, imeVrste, opisVrste. Atribut idVrste je primarni ključ. Atribut opisVrste je opcionalan.

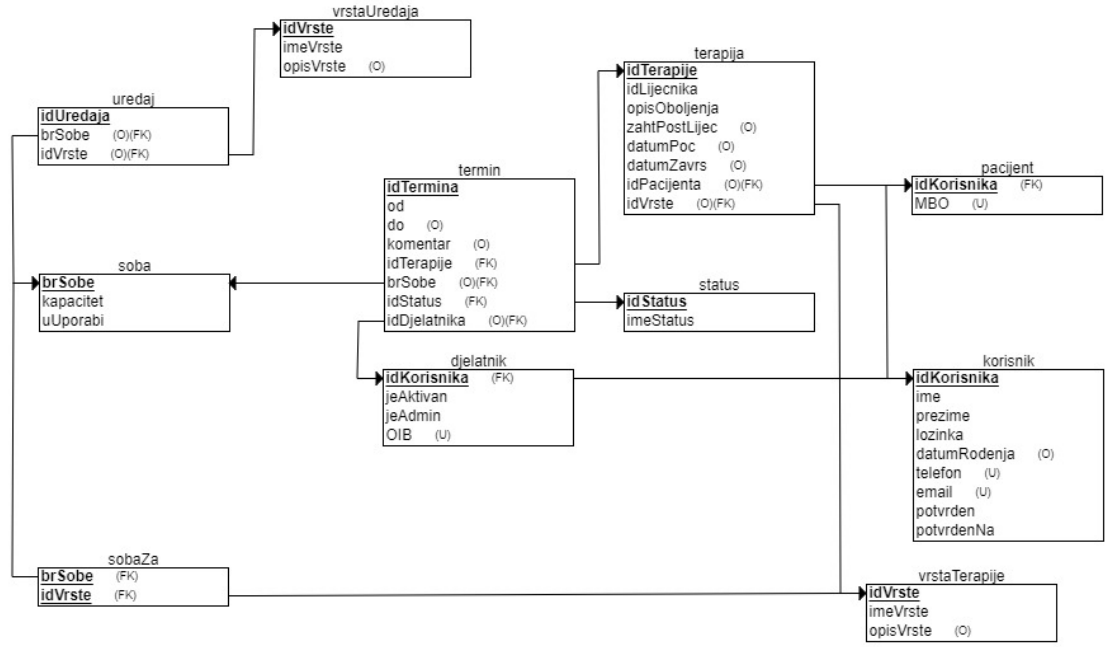
VrstaUredaja		
idVrste	INT	jedinstveni identifikator vrste uređaja
imeVrste	VARCHAR(50)	naziv vrste uređaja
opisVrste	VARCHAR(300)	opis vrste uređaja

Status:

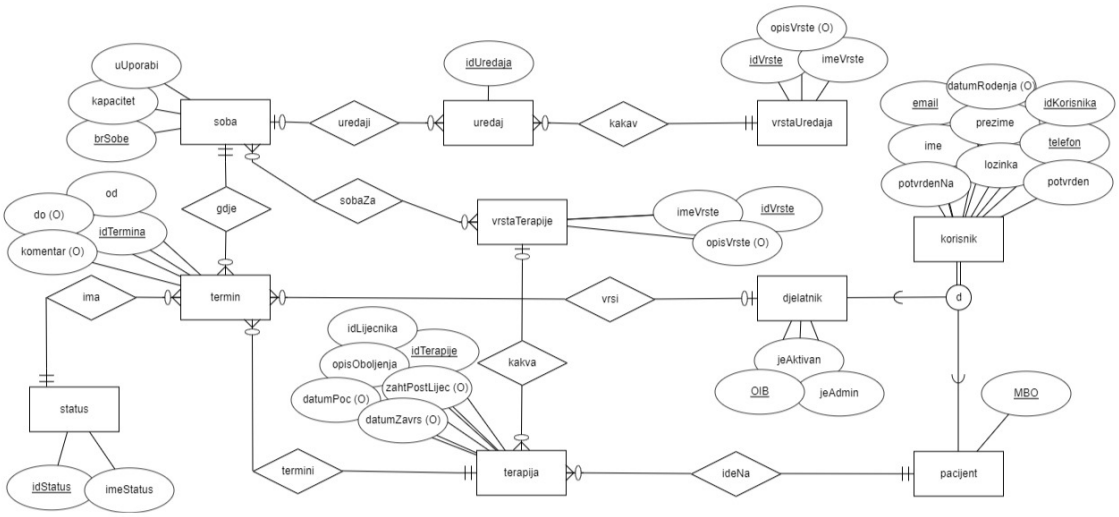
Entitet Status sadrži sve bitne informacije o statusu termina terapije koji je dodijeljen pacijentu. Entitet sadrži atribute: idStatus, imeStatus. Atribut idStatus je primarni ključ.

Status		
idStatus	INT	jedinstveni identifikator uređaja
imeStatus	VARCHAR(50)	naziv statusa (npr. u tijeku, završen)

4.1.2 Dijagram baze podataka



Slika 4.2: Relacijska shema baze podataka

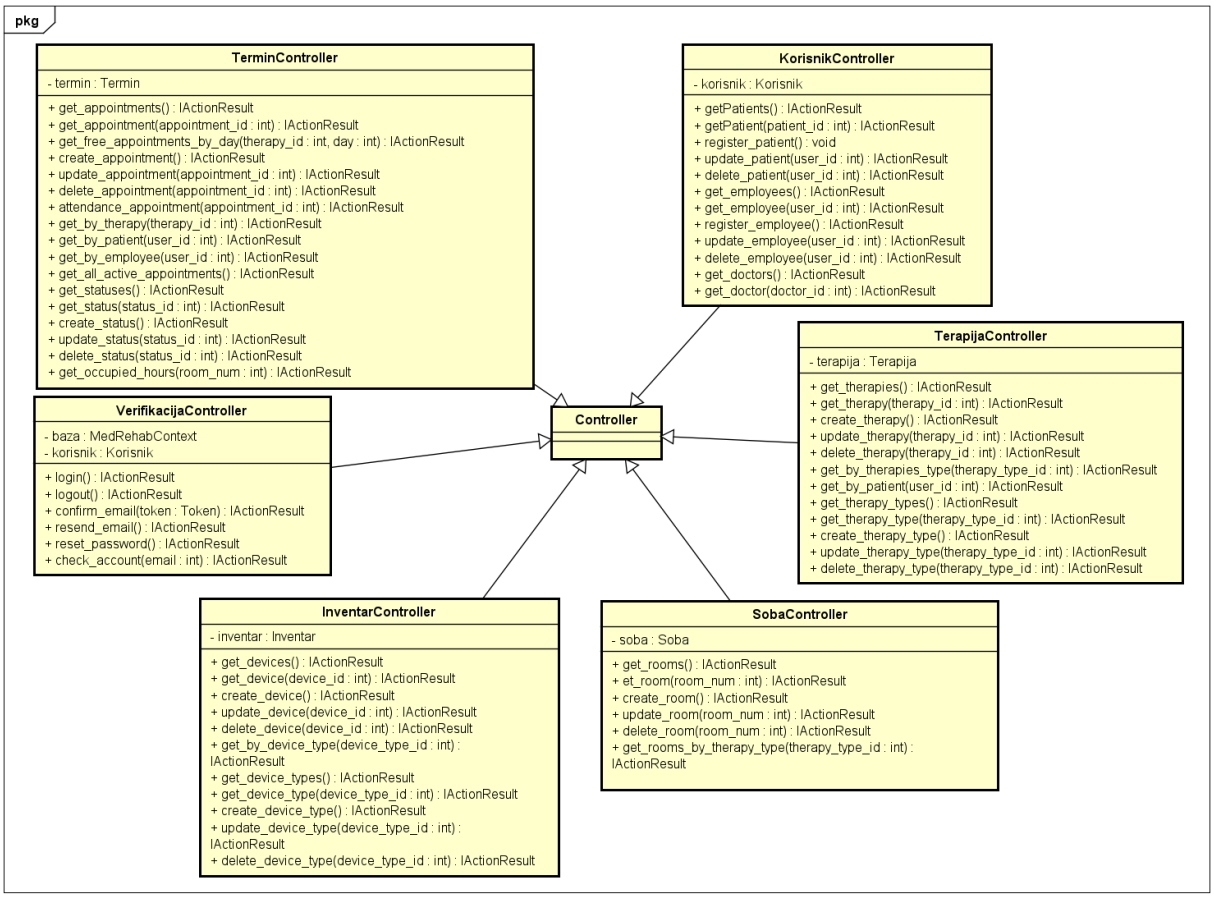


Slika 4.3: ER model baze podataka

4.2 Dijagram razreda

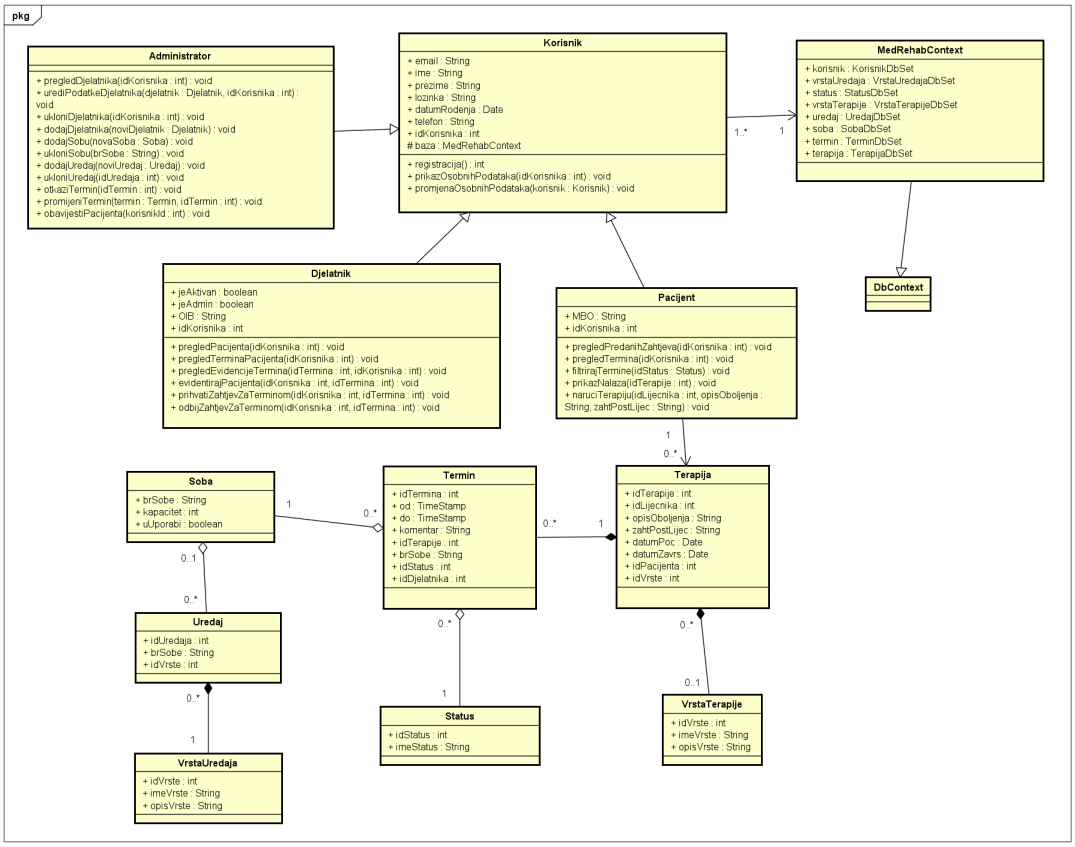
Dijagrami razreda rađeni su po uzoru na MVC obrazac po kojem je naša aplikacija organizirana. Podijeljeni su u 3 dijela: Nadglednik(*Controller*) (4.4), DTO (4.5) i Model (4.6).

Nadglednik (*Controller*) obrađuje zahtjeve korisnika i preko DTO-a(*Data transfer object*) vrši operacije nad Modelom. Zamišljeno je da funkcije implementirane u Nadgledniku vraćaju html status kod. *KorisnikController* ima funkcije za dohvaćanje, stvaranje i izmjenjivanje korisnika(pacijenta i djelatnika). *TerminController* ima funkcije za dohvaćanje, izmjenjivanje i dodavanje termina i statusa koji je povezan za njega. *VerifikacijaController* ima funkcije za verifikaciju i slanje e-maila, promjenu lozinke, prijavu i registraciju korisnika. *InventarController* sadrži funkcije za dodavanje, brisanje i izmjenjivanje inventara/uređaja. *SobaController* ima funkcije za dohvaćanje, uređivanje i stvaranje soba. *TerapijaController* ima funkcije za stvaranje, dohvaćanje i uređivanje prostorija.



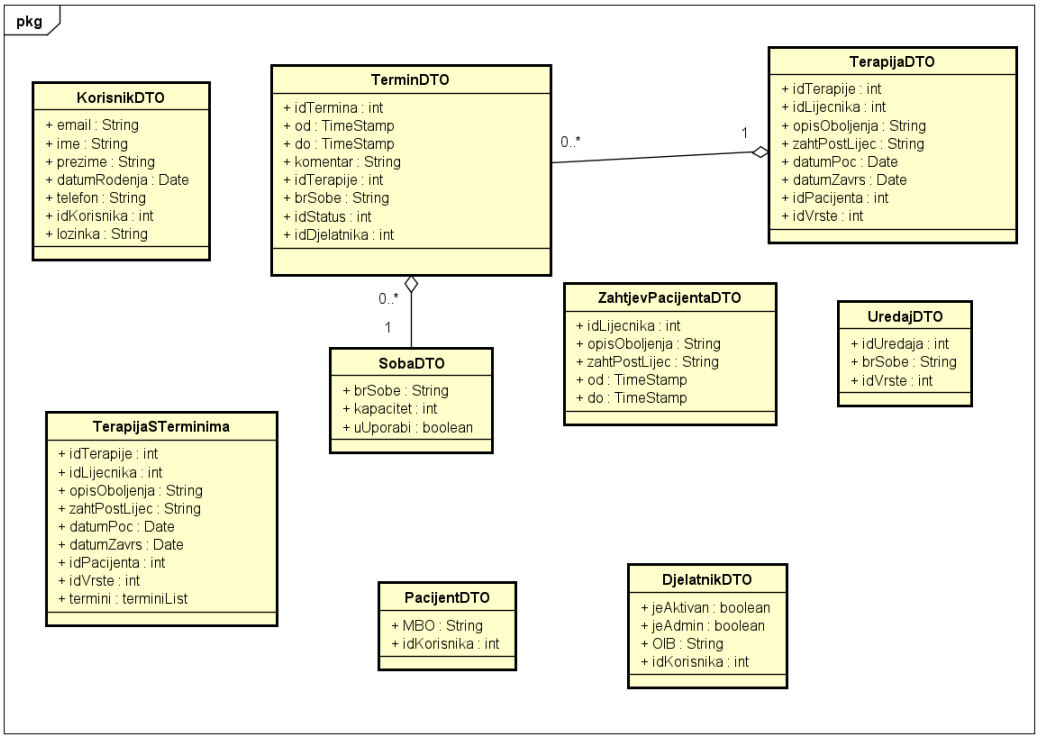
Slika 4.4: Dijagram razreda za dio Controller

Dio s DTO-om sadrži jednostavne klase i služi isključivo za prenošenje podataka kako bi Nadglednik mogao vršiti operacije nad Modelom. Sadrži klase slične dijagramu razreda za dio Model uz nekoliko iznimaka. *TerapijaSTerminima* sadrži atribute terapije s listom termina koji su vezanu uz pojedinu terapiju. *ZahtjevPacijentaDTO* sadrži atribute Terapije i Termina koji su potrebni da bi Pacijent mogao napraviti Zahtjev.



Slika 4.5: Dijagram razreda za dio DTO

Model sadrži atribute i metode koje su potrebne radi komunikacije s bazom podataka. Razred *Korisnik* predstavlja korisnika koji, ako je neprijavljen, ima funkciju registracije, a ako je prijavljen, ima funkciju prikaza i promjene osobnih podataka. *Administrator* predstavlja korisnika koji upravlja djelatnicima, sobama i uređajima i obavještava pacijenta ukoliko je potrebno. *Djelatnik* je liječnik koji ima opciju pregleda pacijenta, njegovih termina i evidencije njegovih termina, evidencije pacijenta i prihvaćanje ili odbijanje zahtjeva za terminom. Razred *Pacijent* je pacijent koji može pregledati predane zahtjeve i termine, filtrirati termine, pregledati nalaze terapije i naručiti novu terapiju. Prisutne su i klase Terapija, vrstaTerapije, Termin, Status, Soba, Uredaj i VrstaUredaja koje za sada nemaju predviđene nikakve funkcije. Svaki razred sadrži atribute bitne za taj razred.

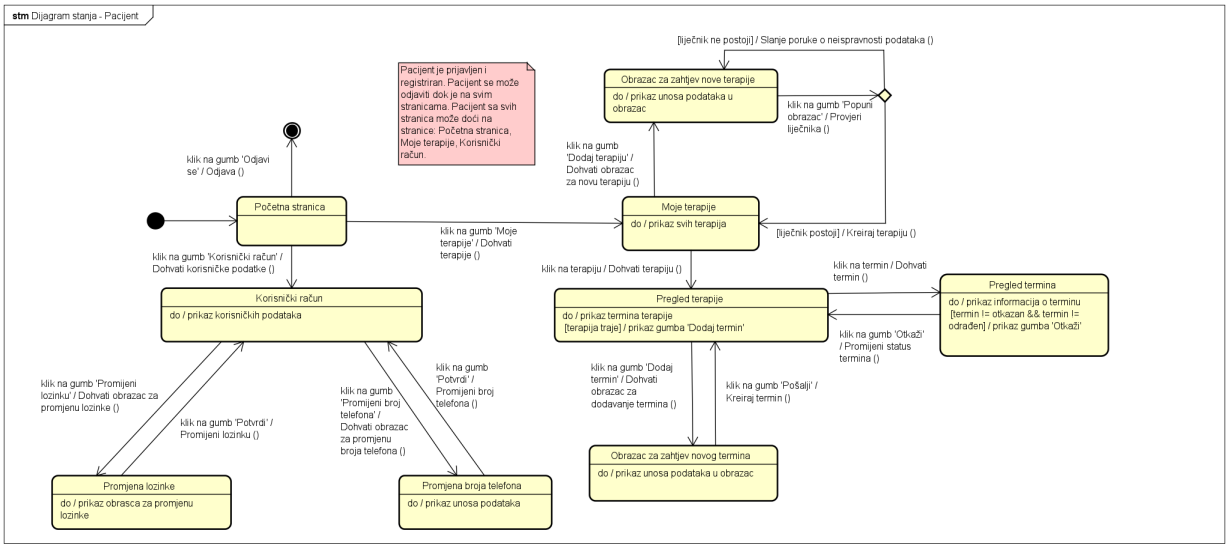


Slika 4.6: Dijagram razreda za dio Model

4.3 Dijagram stanja

Dijagram stanja (4.7) prikazuje stanja i prijelaze između stanja u kojima se nalazi korisničko sučelje kada korisnik - pacijent koristi aplikaciju. Stanje u dijagramu predstavlja ekran (*engl. screen*) u aplikaciji, a prijelaz predstavlja radnju korisnika kada koristi aplikaciju (klik na gumb ili ikonu). Kada se korisnik registrira i prijavi u aplikaciju, prvo mu se prikaže početni ekran. Klikom na gumb 'Moje terapije', korisniku se prikazuje ekran s listom terapija. Na ekranu s listom terapija korisnik vidi svoje terapije i pripadajuće informacije te može dodati novu terapiju klikom na gumb 'Dodaj terapiju'. Kada korisnik dodaje novu terapiju, otvara mu se ekran s obrascem u koji mora unijeti informacije o terapiji. Ako su uneseni podaci ispravni, korisnika se vraća na ekran s listom terapija u koju je dodana i nova terapija. Ako uneseni podaci nisu ispravni, korisnika se vraća na ekran s obrascem te on može ponovno unijeti podatke ili odustati od dodavanja nove terapije. Klikom na terapiju s liste terapija, korisniku se prikazuje novi ekran s informacijama o terapiji i listom termina te njihovim informacijama (dalje: ekran s listom termina). Ako terapija još uvijek traje, korisnik može poslati zahtjev za novim terminom klikom na gumb 'Dodaj termin'. Kada korisnik dodaje novi termin, prikaže mu se ekran s obrascem u koji mora unijeti podatke o terminu. Kada korisnik ispuni obrazac i klikne na gumb 'Pošalji', vraća ga se na ekran s listom termina (na listi je prikazan i novi termin). Klikom na termin s liste termina, korisniku se prikazuje ekran s detaljnim informacijama o terminu te mogućnost otkazivanja termina klikom na gumb 'Otkaži', ako je status termina 'zakazan'. U slučaju otkazivanja termina, korisnika se vraća na ekran s listom termina - otkazani termin se i dalje nalazi u listi termina, ali s promijenjenim statusom ('otkazan'). Klikom na gumb 'Korisnički račun', korisniku se prikazuje ekran s korisničkim informacijama i mogućnošću promjene broja telefona ili lozinke. Klikom na gumb 'Promijeni lozinku', korisniku se prikazuje ekran s obrascem za promjenu lozinke. Ako su podaci uneseni u obrazac ispravni, korisnik na ekranu dobiva potvrdu o uspješnoj promjeni lozinke te ga se vraća na ekran s ažuriranim korisničkim informacijama. Ako uneseni podaci nisu ispravni, korisnika se vraća na obrazac te on ima mogućnost ponovnog unosa podataka ili odustajanja od promjene podataka i vraćanja na ekran s korisničkim informacijama. Isti proces se odvija i u slučaju promjene broja telefona - klikom na gumb 'Promijeni broj telefona'. Korisnik se može odjaviti iz aplikacije klikom na gumb 'Odjavi se', bez obzira na kojem se ekranu nalazi. Također, korisnik u

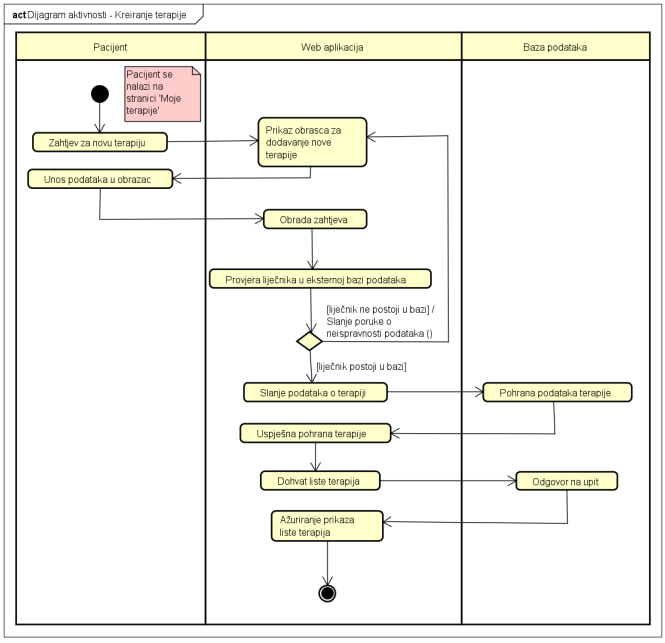
svakom trenutku ima mogućnost povratka na početni ekran, ekran s prikazom terapija ili ekran s korisničkim informacija klikom na odgovarajući gumb.



Slika 4.7: Dijagram stanja

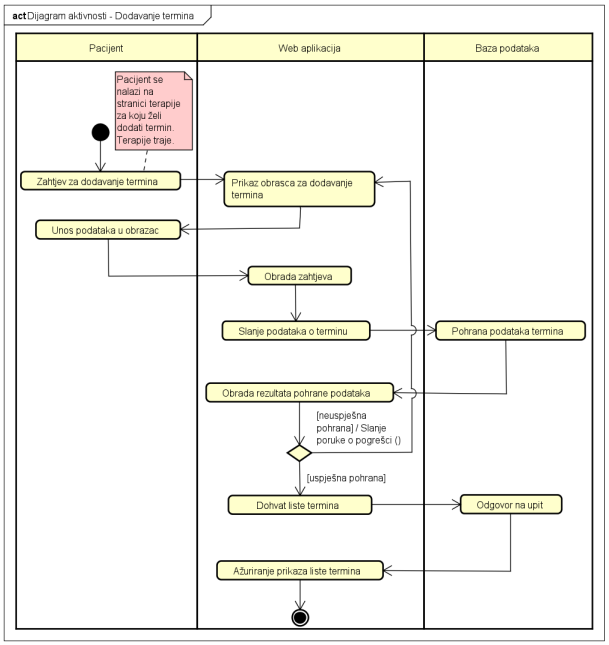
4.4 Dijagram aktivnosti

Dijagram aktivnosti - Kreiranje terapije (4.8) prikazuje proces prijave korisnika (pacijenta) na novu terapiju. Kada korisnik zatraži novu terapiju, prikazuje mu se obrazac koji popunjava s podacima o terapiji. Ako su podaci neispravni, sustav mu dojavljuje grešku i vraća ga na obrazac. Ako su podaci ispravni, nova terapija mu se prikaže na listi terapija.



Slika 4.8: Dijagram aktivnosti za kreiranje terapije

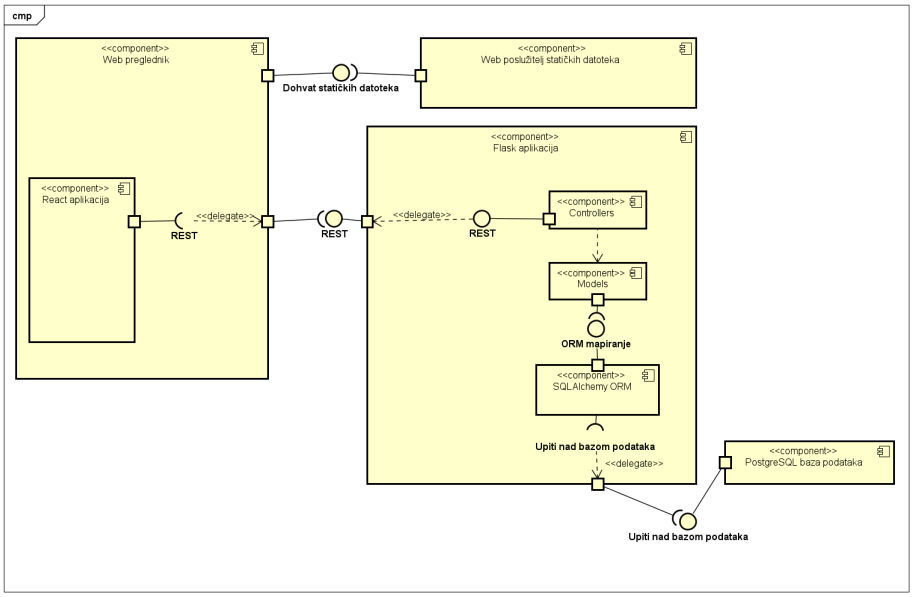
Dijagram aktivnosti - Dodavanje termina (4.9) prikazuje proces naručivanja korisnika (pacijenta) na novi termin odabrane terapije. Kada korisnik zatraži novi termin, prikazuje mu se obrazac koji popunjava s podacima o terminu. Ako su podaci neispravni, sustav mu dojavljuje grešku i vraća ga na obrazac. Ako su podaci ispravni, novi termin se prikaže u listi termina.



Slika 4.9: Dijagram stanja

4.5 Dijagram komponenti

Dijagram komponenti (4.7) služi za vizualizaciju arhitekture sustava - sustav se dijeli na komponente koje međusobno komuniciraju putem sučelja. Ključni dijelovi (komponente) ove web aplikacije su: Web preglednik, React aplikacija, Web poslužitelj statičkih datoteka, Flask aplikacija i SQL baza podataka. Kada web preglednik putem sučelja preuzme statičke datoteke (HTML, CSS, JS datoteke) s web poslužitelja statičkih datoteka, u web pregledniku se izvršava React aplikacija (JavaScript kod). React aplikacija izgrađuje korisničko sučelje te omogućuje interakciju korisnika s aplikacijom. Kada korisnik traži, mijenja ili unosi podatke u aplikaciju, React aplikacija šalje HTTP zahtjeve Flask aplikaciji putem REST sučelja, koje je implementirano unutar Flask aplikacije. Flask aplikacija obrađuje zahtjev te komunicira s bazom podataka s pomoću SQLAlchemy integracije. Kada obradi zahtjeve, Flask aplikacija šalje odgovore React aplikaciji putem REST sučelja.



Slika 4.10: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

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

Komunikacija u timu realizirana je korištenjem aplikacija WhatsApp[1] i Discord[2]. Za izradu dijagrama korištena je aplikacija Astah Proffesional[3], a za pisanje dokumenatcije korišten je opisni jezik LaTeX[4] u aplikaciji TeXstudio[5]. kao sustav za upravljanje izvornim kodom koristo se sustav Git[6]. Udaljeni repozitorij projekta je dostupan na web platformi GitHub[7].

Kao razvojno okruženje koristili smo Visual Studio Code[8]. Visual Studio Code je uređivač koda kojeg je razvio Microsoft za Windows, Linux i MacOs operacijske sustave. Olakšava rad s Gitom i omogućava pokretanje programa. Omogućava i instaliranje exstenzija za lakši rad s različitim programskim jezicima i radnim okvirima.

Aplikacija je pisana koristeći radni okvir Flask[9], SQLAlchemy[10] i jezik Python[11] za izradu *backenda*, te React[12] i jezik JavaScript[13]. Flask je mikro radni okvir koji pruža osnovne funkcionalnosti za razvoj web-aplikacija, ali i omogućava programeru određenu razinu slobode. React, poznatiji kao ReactJS ili React.js je bibli-

[1] https://www.whatsapp.com/
[2] https://discord.com/
[3] https://astah.net/products/astah-professional/
[4] https://www.latex-project.org/
[5] https://www.texstudio.org/
[6] https://git-scm.com/
[7] https://github.com/
[8] https://code.visualstudio.com/
[9] https://flask.palletsprojects.com/
[10] https://www.sqlalchemy.org/
[11] https://www.python.org/
[12] https://react.dev/
[13] https://www.javascript.com/

oteka u javascriptu koja omogućava izgradnju korisničkih sučelja, održavana od strane Facebooka.

Baza podataka nalazi se na Renderu[14].

[14] https://render.com/

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

U ovom odjeljku je opisano izvršavanje testova za testiranje ispravnosti sučelja poslužitelja.

Incijalizacija

Kako bi postavili testno okruženje i pokrenuli tesnog poslužitelja vršimo kod:

```
import pytest
import sys, os
from posixpath import abspath, dirname
sys.path.append(abspath(dirname(__file__)))
sys.path.append('../controllers')
sys.path.append('../models')
from models import Therapy, Appointment, TherapyType, Room
from db import db
from app import build_app
from mail import mail

@pytest.fixture
def client():
    app = build_app()
    app.config['TESTING'] = True
    app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///' +
os.path.join(os.getcwd(), 'temp.db')
    db.init_app(app)
    mail.init_app(app)
    with app.app_context():
        db.create_all()
        yield app.test_client()
        db.drop_all()
```

Slika 5.1: Inicijalizacija testnog okruženja

U prvom dijelu se definiraju korišteni moduli, a drugom dijelu se definira funkcija *client* koja vraća instancu poslužitelja koju je onda moguće koristiti za testiranje. U funkciji se stvara prazna privremena baza podataka na istoj memorijskoj lokaciji.

S naredbama `db.init_app(app)` će se inicijalizirati baza podataka po uzoru na db objekt koji sadrži sve entitete u poslužitelju. `mail.init_app(app)` služi kako bi se omogućilo slanje obavijesti na e-poštu. Iako se ono ne provjerava tijekom testiranja, potrebno je za ispravan rad aplikacije.

Ova funkcija će se pokrenuti prije svakog testa kako bi se baza podataka očistila i time testovi ne bi utjecali jedni na druge.

Testovi

U testovima će se uglavnom testirati manipulacija terminima jer je to najsloženija procedura.

```python
def test_create_appointment_missing_fields(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1
    response = client.post('/appointments', json={})
    print(response.json)
    assert response.status_code == 400
    assert response.json['message'] == "Nedostajuća polja: date_from, therapy_id"

def test_create_appointment_therapy_not_found(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1
    response = client.post('/appointments', json={"date_from": "2022-01-01
10:00", "therapy_id": 1})
    assert response.status_code == 404
    assert response.json['message'] == "Terapija ne postoji"

def test_create_appointment_forbidden(client):
    with client.session_transaction() as session:
        session['role'] = 'doctor'
        session['user_id'] = 1

    response = client.post('/appointments', json={"date_from": "2022-01-01
10:00", "therapy_id": 1})
    assert response.status_code == 401
    assert response.json['message'] == "You don't have any of the required roles:
patient, admin"

def test_create_appointment_invalid_date_format(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1

    # create therapy
    therapy = Therapy(patient_id=1, doctor_id=1, date_from="2024-02-02",
date_to="2024-03-02",
                      disease_descr="desc")
    db.session.add(therapy)
    db.session.commit()

    response = client.post('/appointments', json={"date_from": "2022-01-01",
"therapy_id": 1})
    assert response.status_code == 400
    assert response.json['message'] == "Pogrešan format datuma"
```

Slika 5.2: Testovi 1, 2, 3 i 4

```python
def test_create_appointment_valid(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1

    # create therapy_type
    therapy_type = TherapyType(therapy_type_name="test", description="desc")
    db.session.add(therapy_type)
    db.session.commit()

    # create therapy
    therapy = Therapy(patient_id=1, doctor_id=1, date_from="2024-02-02",
date_to="2024-03-02",
                      disease_descr="desc", therapy_type_id=1)
    db.session.add(therapy)
    db.session.commit()

    # create room
    room = Room(room_num=1, capacity=1, in_use=True)
    room.therapy_types.append(therapy_type)
    db.session.add(room)
    db.session.commit()

    response = client.post('/appointments', json={"date_from": "2024-02-02
10:00", "therapy_id": 1})
    assert response.status_code == 201

def test_create_appointment_no_room(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1

    # create therapy_type
    therapy_type = TherapyType(therapy_type_name="test", description="desc")
    db.session.add(therapy_type)
    db.session.commit()

    # create therapy
    therapy = Therapy(patient_id=1, doctor_id=1, date_from="2024-02-02",
date_to="2024-03-02",
                      disease_descr="desc", therapy_type_id=1)
    db.session.add(therapy)
    db.session.commit()
```

Slika 5.3: Testovi 5 i 6a

```
    # create room
    room = Room(room_num=1, capacity=0, in_use=True)
    room.therapy_types.append(therapy_type)
    db.session.add(room)
    db.session.commit()

    response = client.post('/appointments', json={"date_from": "2024-02-02
10:00", "therapy_id": 1})
    assert response.status_code == 400
    assert response.json['message'] == "Nema slobodnih soba za ovaj termin"

def test_get_available_hours_by_day_therapy_not_found(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1

    response = client.get('/free-appointments/therapy/1/date/2022-01-01')
    assert response.status_code == 404
    assert response.json['error'] == "Terapija ne postoji"

def test_get_available_hours_by_day(client):
    with client.session_transaction() as session:
        session['role'] = 'patient'
        session['user_id'] = 1

    # create therapy
    therapy = Therapy(patient_id=1, doctor_id=1, date_from="2024-01-01",
date_to="2024-01-02",
                      disease_descr="desc", therapy_type_id=1)
    db.session.add(therapy)
    db.session.commit()

    # create therapy_type
    therapy_type = TherapyType(therapy_type_name="test", description="desc")
    db.session.add(therapy_type)

    # create room
    room = Room(room_num=1, capacity=1, in_use=True)
    room.therapy_types.append(therapy_type)
    db.session.add(room)
    db.session.commit()

    # create appointment
    appointment = Appointment(therapy_id=1, room_num=1, date_from="2024-02-01
10:00")
```

Slika 5.4: Testovi 6b, 7 i 8a

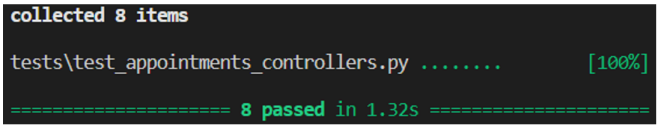
```
db.session.add(appointment)
db.session.commit()

response = client.get('/free-appointments/therapy/1/date/2024-02-01')
assert response.status_code == 200
assert '10:00' not in response.json['data']['appointments']
```

Slika 5.5: Test 8b

- `test_create_appointment_missing_fields(client)`
  Testira hoće li funkcija za stvaranje termina (putem HTTP POST zahtjeva) vratiti status 400 i odgovarajuću poruku ako nedostaju polja `date_from` i `therapy_id`.
- `test_create_appointment_therapy_not_found(client)`
  Provjerava hoće li funkcija vratiti status 404 i odgovarajuću poruku ako terapija s navedenim ID-jem ne postoji.
- `test_create_appointment_forbidden(client)`
  Testira hoće li funkcija vratiti status 401 i odgovarajuću poruku ako korisnik koji pokušava stvoriti termin nema ulogu pacijenta ili administratora.
- `test_create_appointment_invalid_date_format(client)`
  Provjerava hoće li funkcija vratiti status 400 i odgovarajuću poruku ako je format datuma neispravan prilikom stvaranja termina.
- `test_create_appointment_valid(client)`
  Testira uspješno stvaranje termina (status 201) kada su svi uvjeti zadovoljeni, uključujući postojanje terapije, tipa terapije, sobe i slobodnih termina.

- `test_create_appointment_no_room(client)`
  Provjerava hoće li funkcija vratiti status 400 i odgovarajuću poruku ako nema slobodnih soba za navedeni termin.
- `test_get_available_hours_by_day_therapy_not_found(client)`
  Testira hoće li funkcija za dobivanje slobodnih termina po danu vratiti status 404 i odgovarajuću poruku ako terapija s navedenim ID-jem ne postoji.
- texttttest_get_available_hours_by_day(client)
  Provjerava ispravno dobivanje slobodnih termina (status 200) za određeni dan, uz pretpostavku da postoji terapija, tip terapije, soba i već stvoren termin na taj dan.

Rezultati
Svi testovi su usješno riješeni.



Slika 5.6: Rezultati testova

5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium[15]*. 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:

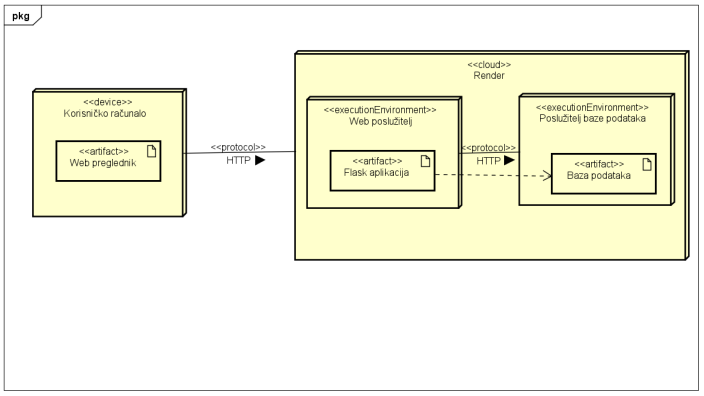
- *dodatak za preglednik* ***Selenium IDE*** *- snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver*** *- podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

[15] `https://www.seleniumhq.org/`

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

5.3 Dijagram razmještaja

Specifikacijski dijagram razmještaja (5.7) prikazuje kako se ključni dijelovi sustava razmještaju na fizičke i virtualne komponente te kako ti dijelovi međusobno komuniciraju. Korisnik na svom računalu može s pomoću web preglednika pristupiti web aplikaciji. Oblak pruža poslužitelje za aplikaciju i bazu podataka. Korisničko računalo koje djeluje kao klijent komunicira s oblakom koje djeluje kao poslužitelj putem HTTP protokola. Web aplikacija i baza podataka također komuniciraju putem HTTP protokola.



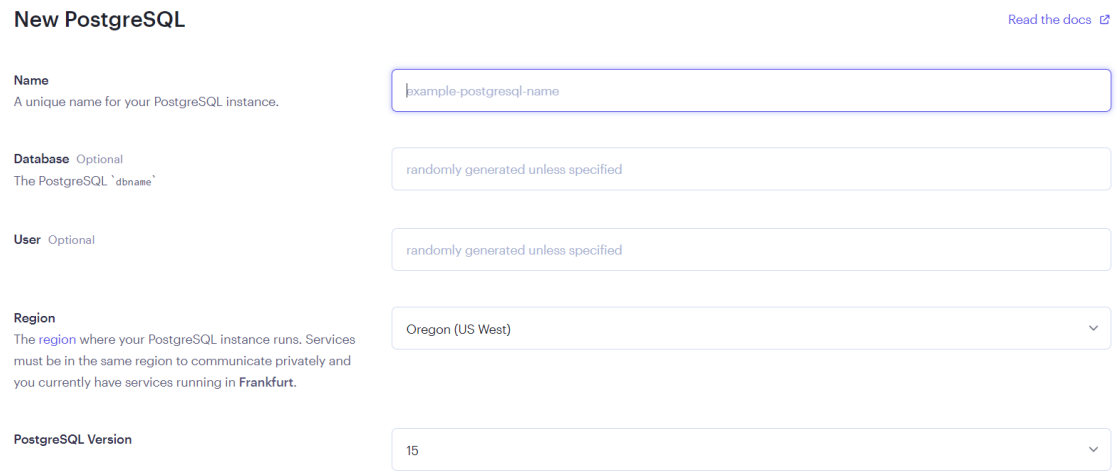
Slika 5.7: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Postupak postavljanja baze podataka

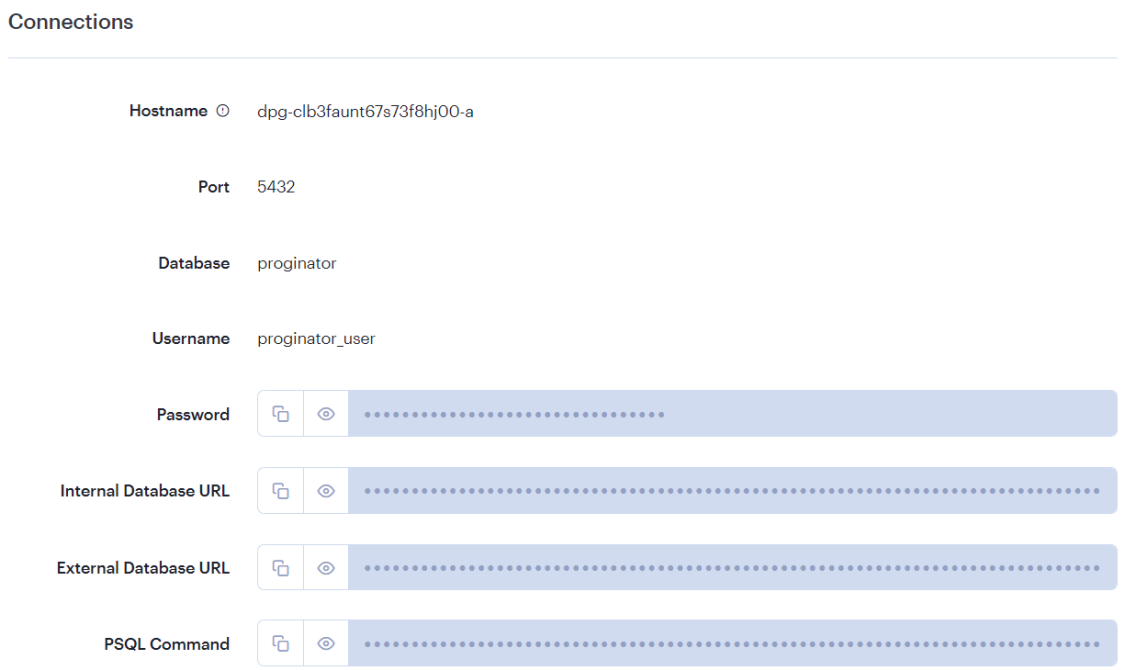
- Postavljanje i konfiguracija PostgreSQL baze podataka unutar platforme Render

  U polje *Name* potrebno je upisati ime instance PostgreSQL baze podataka. Polje *Database* i *User* Render popunjava s nasumično generiranim vrijednostima. Umjesto tih vrijednosti definirali smo vlastite, proginator za Database i proginator_user za User. Nakon toga je potrebno odabrati regiju. Odabrali smo regiju Frankfurt (EU central) zato što je najbliža klijentu.



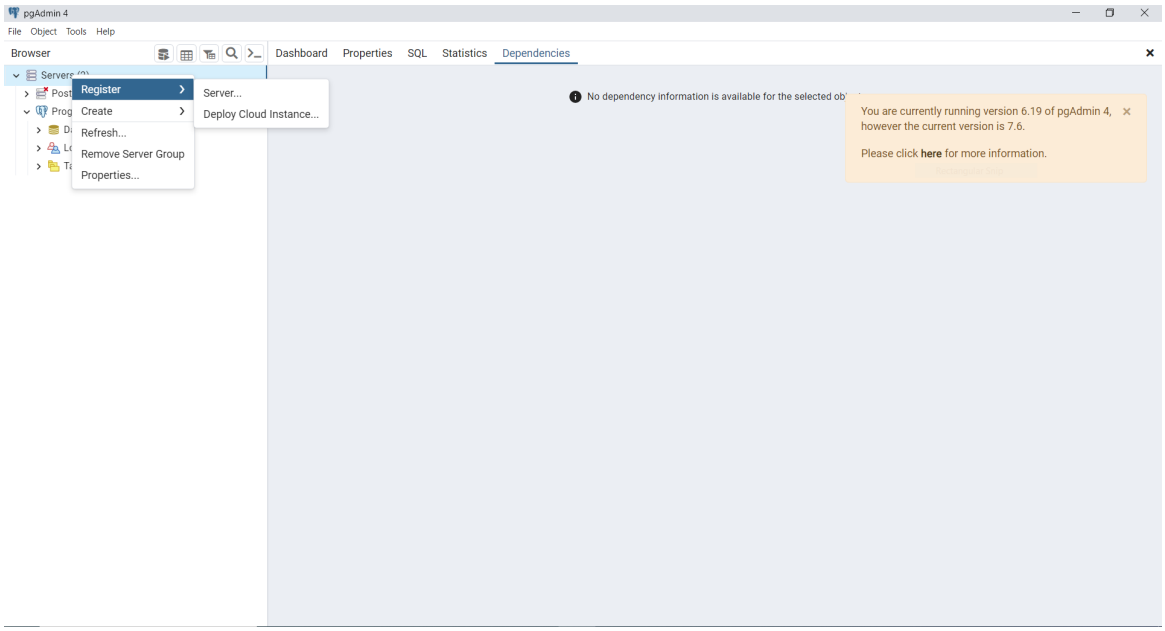
Slika 5.8: Postavljanje i konfiguracija PostgreSQL baze podataka unutar platforme

  Nakon toga Render prikazuje informacije o bazi podataka: ime poslužitelja (*Hostname*), vrata (*Port*), ime baze podataka (*Database*), korisničko ime PostgreSQL korisnika (*username*), lozinku korisnika (*password*) i URL za povezivanje na bazu podataka.
- Konfiguracija pristupa bazi podataka iz aplikacije pgAdmin
  Unutar aplikacije pgAdmin potrebno je u izborniku odabrati Servers -¿ Register.



Slika 5.9: Konfiguracija pristupa bazi podataka iz aplikacije pgAdmin

Nakon toga, potrebno je ispuniti formu s podacima za povezivanje na bazu podataka koje smo dobili u platformi Render.



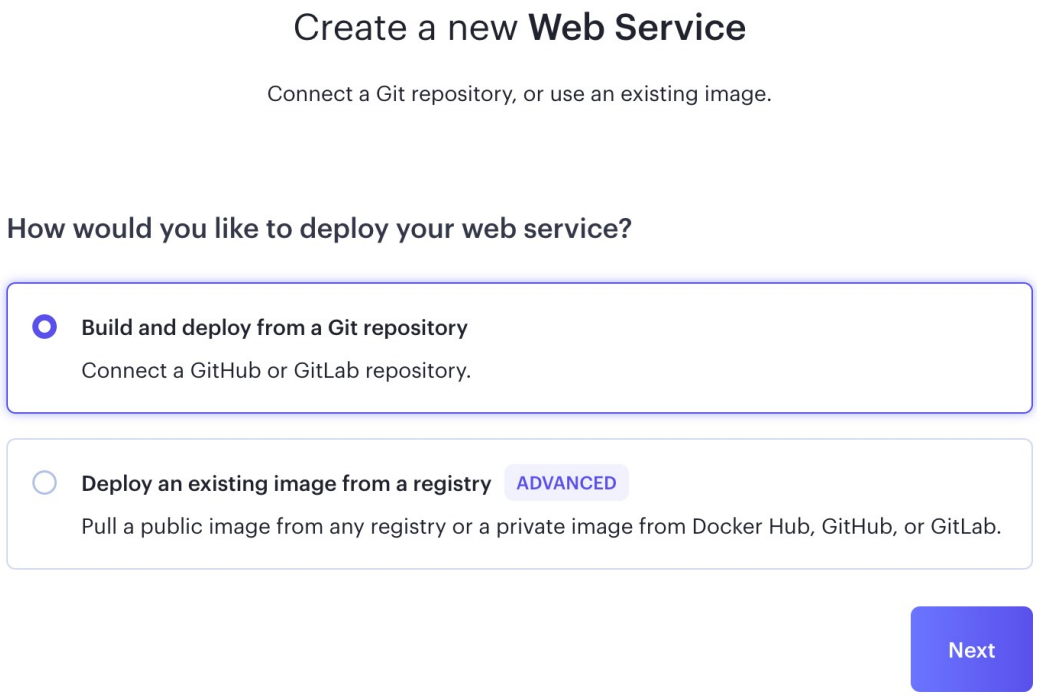
Slika 5.10: Ispunjavanje forme

- Incijalizacija baze podataka i stvaranje tablica

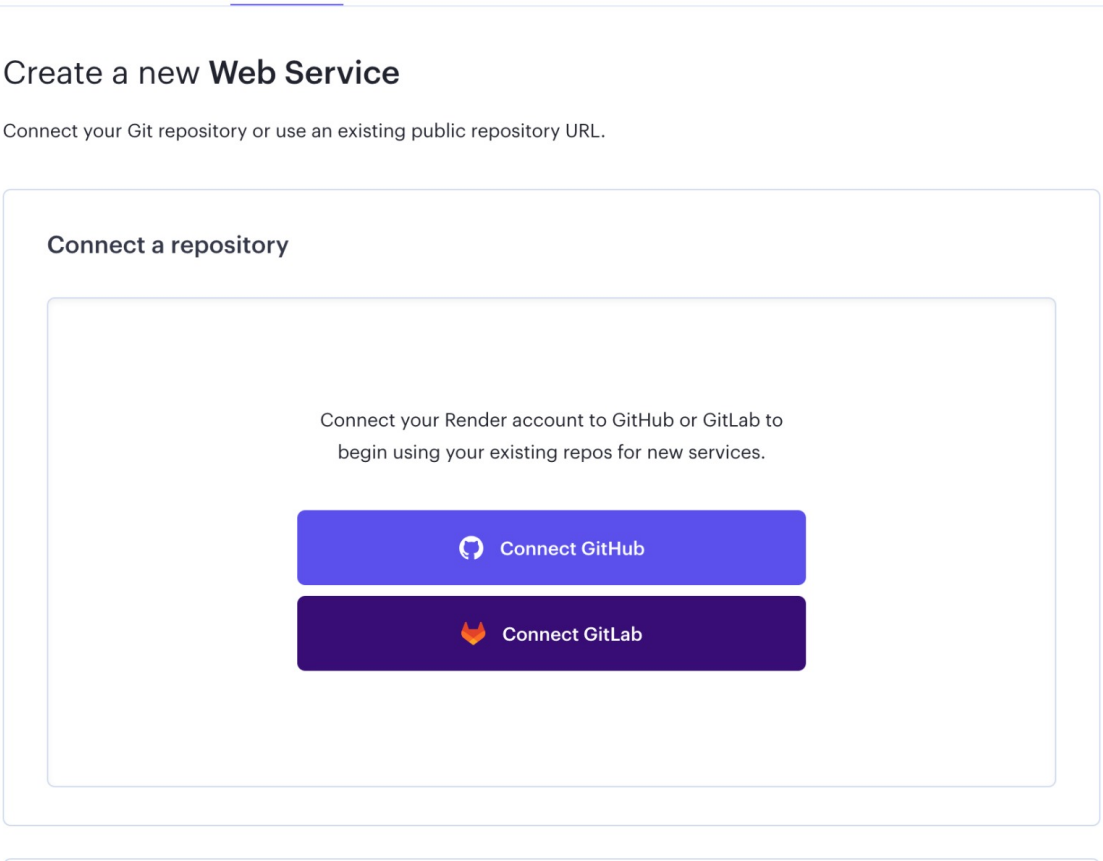
Potrebno je pokrenuti inicijalizacijsku skriptu db_init.sql koja stvara tablice u bazi podataka.

Postupak postavljanja backend aplikacije

- Povezivanje Render web servisa s GitHub repozitorijem



Slika 5.11: Povezivanje Render web servisa s GitHub repozitorijem



Slika 5.12: Povezivanje Render web servisa s GitHub repozitorijem

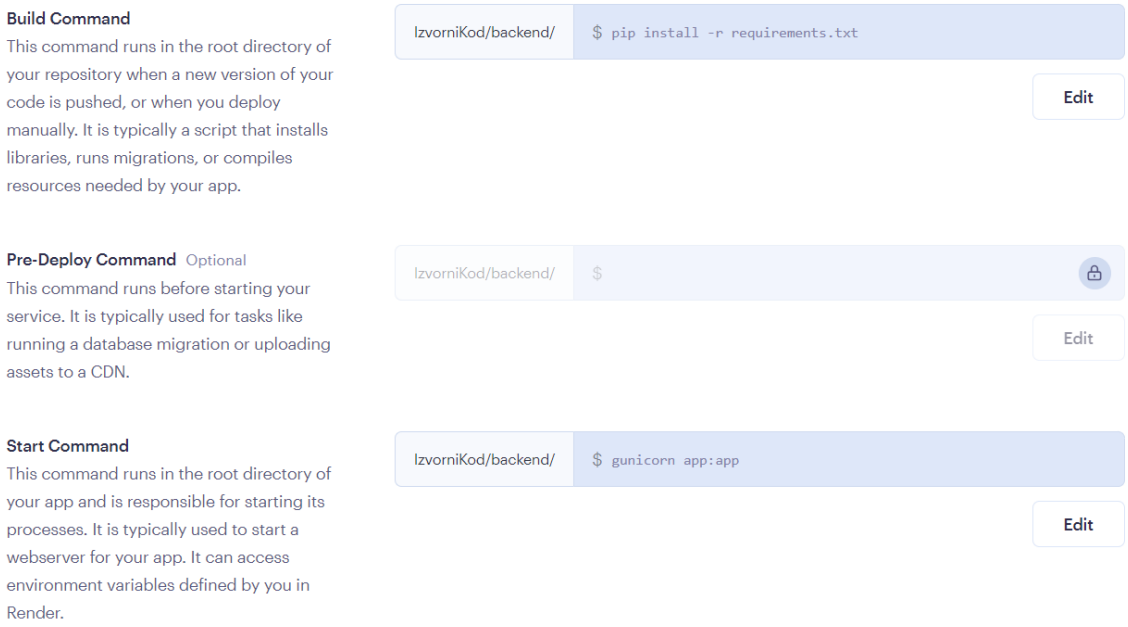
Render web servis povezujemo s GitHub repozitorijem kako bi Render platforma imala pristup izvornom kodu.

- Konfiguracija web servisa

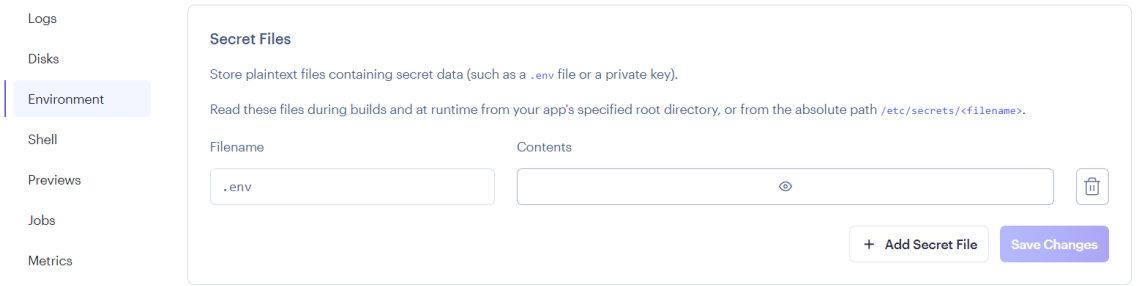
- Postavljanje varijabli okruženja

Postupak postavljanja frontend aplikacije

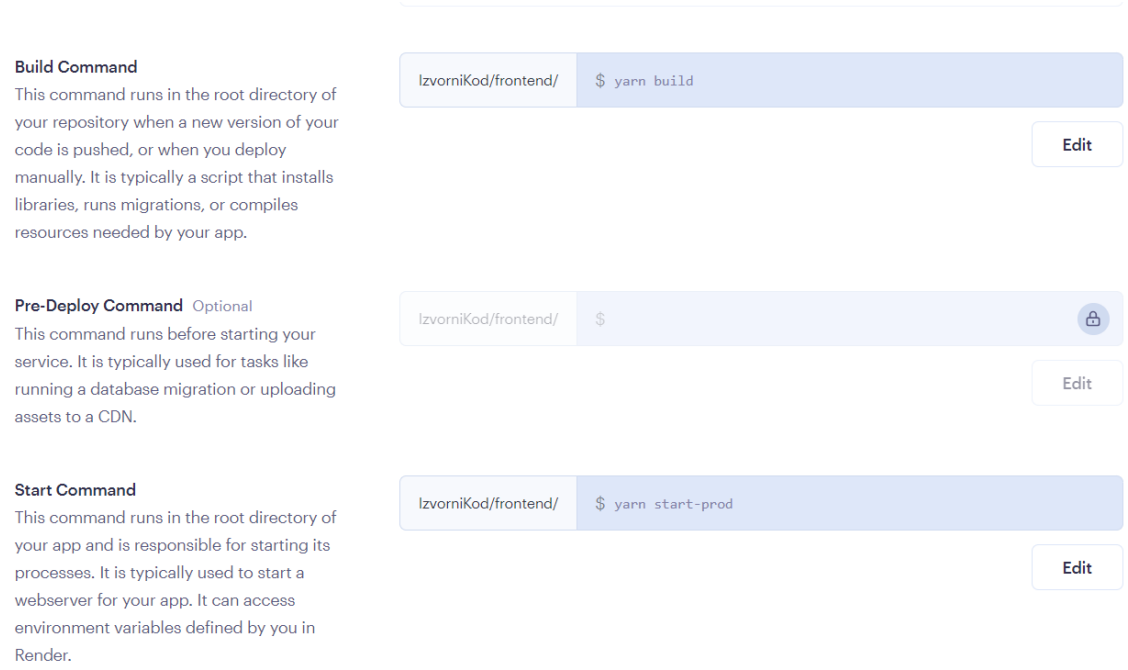
- Konfiguracija web servisa



Slika 5.13: Konfiguracija web servisa



Slika 5.14: Postavljanje varijabli okruženja



Slika 5.15: Konfiguracija web servisa

Ostali koraci jednaki su koracima postupka postavljanja backend aplikacije.

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

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak
    - Datum: 18. listopada 2023.
    - Prisustvovali: svi
    - Teme sastanka:
        - razjašnjavanje dobivene teme projekta - Medicinska Rehabilitacija
        - odabir tehnologija koje će biti korištene u izradi projekta
2. sastanak
    - Datum: 23. listopada 2023.
    - Prisustvovali: E.Badurina, L.Lasović, V.Rohak, K.Vrdoljak, F.Zlatar
    - Teme sastanka:
        - razjašnjavanje teme projekta
        - zapisivanje konkretnih pitanja vezano uz projekt
3. sastanak
    - Datum: 25. listopada 2023.
    - Prisustvovali: svi
    - Teme sastanka:
        - podjela uloga
        - razrada baze podataka
4. sastanak
    - Datum: 6. studenoga 2023.
    - Prisustvovali: svi
    - Teme sastanka:
        - komentiranje predloška UI-dizajna
        - razrada i dovršavanje baze podataka
        - razjašnjavanje dodatnih nejasnoća, pitanja za laboratorijsku vježbu
5. sastanak
    - Datum: 8. studenoga 2023.

- Prisustvovali: L.Akmačić, E.Badurina, L.Lasović
- Teme sastanka:
  - komentar do sad napravljenog

6. sastanak
   - Datum: 15. studenoga 2023.
   - Prisustvovali: svi
   - Teme sastanka:
     - revizija do sada napravljenog
     - baza podataka
     - funkcionalnost prijave i registracije
7. sastanak
   - Datum: 6. prosinca 2024.
   - Prisustvovali: svi
   - Teme sastanka:
     - evaluacija prve verzije projekta
     - podjela bodova
     - podjela poslova za drugi ciklus
8. sastanak
   - Datum: 4. siječnja 2024.
   - Prisustvovali: svi
   - Teme sastanka:
     - predstavljanje do sad napravljenog
     - komentiranje dijelova stranice
9. sastanak
   - Datum: 10. siječnja 2024.
   - Prisustvovali: svi
   - Teme sastanka:
     - predstavljanje alfa inačice
     - podjela preostalog posla
     - dogovaranje rokova

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Ema Badurina	**Lovro Akmačić**	**Andrea Jakovčević**	**Lucija Lasović**	**Vlatko Rohak**	**Karlo Vrdoljak**	**Fran Zlatar**
Upravljanje projektom							
Opis projektnog zadatka	4	2	2	2	2	2	2
Funkcionalni zahtjevi	15			5			
Opis pojedinih obrazaca	5			5			
Dijagram obrazaca	2		3				
Sekvencijski dijagrami	12			4			
Opis ostalih zahtjeva	1			1			
Arhitektura i dizajn sustava				2			
Baza podataka	2	2	2	6	2	3	3
Dijagram razreda	3	4					
Dijagram stanja				5			
Dijagram aktivnosti				7			
Dijagram komponenti				4			
Korištene tehnologije i alati				3			
Ispitivanje programskog rješenja						3	
Dijagram razmještaja				2			

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ema Badurina	**Lovro Akmačić**	**Andrea Jakovčević**	**Lucija Lasović**	**Vlatko Rohak**	**Karlo Vrdoljak**	**Fran Zlatar**
Upute za puštanje u pogon							0.5
Dnevnik sastajanja	0.3						
Zaključak i budući rad	2						
Popis literature							
UI dizajn					6		
Frontend (prijava i registracija)					30		
Frontend (stranice pacijenta)			50				
Frontend (stranice djelatnika)	2	60					
Frontend (stranice administratora)					30		
Frontend (komponenta za tablice)					30		
Izrada baze podataka							10
Spajanje s bazom podataka						5	5
Backend (prijava i registracija/ računi(accounts))						10	10
Backend (termini)	3					17	4
Backend (terapije)						8	5
Backend (verifikacija)							20
Backend (sobe)						5	5
Backend (inventar)						5	5

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ema Badurina	**Lovro Akmačić**	**Andrea Jakovčević**	**Lucija Lasović**	**Vlatko Rohak**	**Karlo Vrdoljak**	**Fran Zlatar**
Deploy					4	4	14
Ispitivanje aplikacije	3	3	3	3	3	3	3
Testiranje aplikacije						6	

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.