

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	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	23
3.2 Ostali zahtjevi	26
4 Arhitektura i dizajn sustava	27
4.1 Baza podataka	28
4.1.1 Opis tablica	28
4.1.2 Dijagram baze podataka	33
4.2 Dijagram razreda	34
4.3 Dijagram stanja	38
4.4 Dijagram aktivnosti	39
4.5 Dijagram komponenti	40
5 Implementacija i korisničko sučelje	41
5.1 Korištene tehnologije i alati	41
5.2 Ispitivanje programskog rješenja	42
5.2.1 Ispitivanje komponenti	42
5.2.2 Ispitivanje sustava	42
5.3 Dijagram razmještaja	43
5.4 Upute za puštanje u pogon	44
6 Zaključak i budući rad	45
Popis literature	46
Indeks slika i dijagrama	47

Dodatak: Prikaz aktivnosti grupe

48

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	*

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.1	*	* *	*
1.2	*	*	*
2.0	*	*	*

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

2. Opis projektnog zadatka

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 odbijaju ili prihvaćaju termine ovisno o raspoloživosti prostorija, opreme i osoblja pritom imajući mogućnost uvida u pacijentove 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 mijenjanja lozinke.

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 stranicu gdje se prikazuju njegovi termini i zahtjevi, ukoliko ih ima. 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 će moći napraviti zahtjev za terminom. Zahtjev se sastoji od:

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

Pacijent odabire željeni termin tako što prvo odabire tip rehabilitacije i referencu na neku od njegovih terapija, nakon toga unosi datum u tražilicu, na temelju kojeg će mu se prikazati slobodni termini. Pacijent će odabrati jedan, opcionalno dodati komentar i poslati zahtjev. Pacijent će o odobrenju ili odbijanju zahtjeva, te o svim mogućim promjenama biti obaviješten e-mailom.

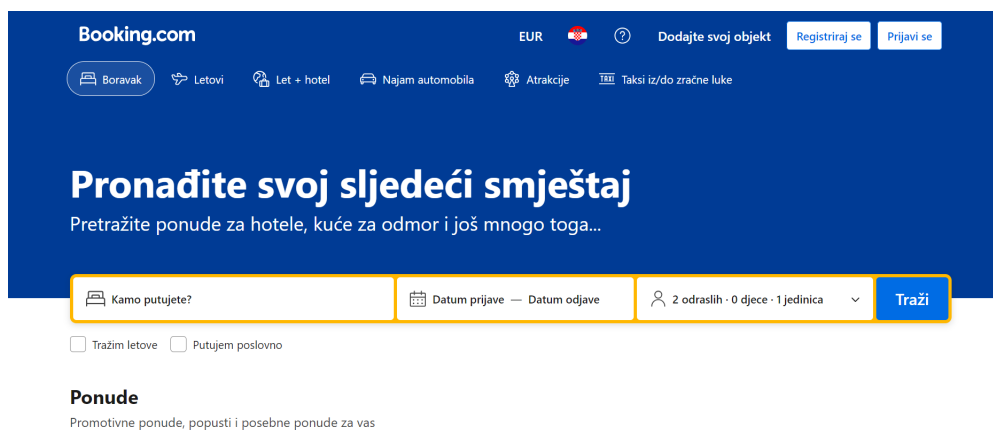
Liječnik nakon prijavljivanja u sustav ima pregled svih pacijenata i njihovih podataka. Imati će opciju pretraživanja pacijenta. Klikom na prikaz terapije bit će mu prikazani svi termini odabranog pacijenta i detalji o njima, tu će imati opciju evidentirati dolazak pacijenta i zabilježiti komentare vezane uz napredak ili pregledati napredak i komentare iz prošlih termina. 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 prihvatanja ili odbijanja termina ovisno o raspoloživosti prostorija i opreme. Kod odbijanja zahtjeva liječnik će morati dodijeliti pacijentu novi termin u kojem su oprema i prostorije raspoloživi.

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, liječnik će imati mogućnost obavješćavanja pacijenta o otkazanom terminu te mogućnost izdavanja novog termina.

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 prijavljivanje, a djelatnicima odbijanje ili prihvaćanje termina rehabilitacije u samo par klikova.

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 ponudene termine/opcije ili može samostalno odabrati vrijeme željenog termina/rezervacije.



Slika 2.1: Primjer slične aplikacije

Aplikacija će se moći proširiti da djelatnici uopće ne moraju prihvaćati ili odbijati prijave, već će to sustav raditi automatski na temelju raspoloživih podataka o opremi, prostorijama, djelatnicima i već zauzetim terminima. Svi termini, prostori, 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. Ostala bi mogućnost upućivanja maila ili telefonskog poziva pacijentu u slučaju ikakvih promjena.

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
3. Pacijent-prijavljeni korisnik (inicijator) može:
 - (a) prikazati i mijenjati svoje osobne podatke i lozinku
 - (b) vidjeti svoje zahtjeve za terminom terapije (vrijeme predaje zahtjeva, vrijeme željenog termina, tip rehabilitacije, liječnika koji ga je uputio na rehabilitaciju, referencu na prošlu terapiju i status)
 - (c) vidjeti svoje termine (vrijeme termina, prostoriju, tip rehabilitacije, liječnika, napomene i ishod termina)
 - (d) filtrirati svoje termine
 - (e) prikazati nalaz (detalji i komentari) s termina odrađene terapije
 - (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 terapiju između priloženih ili unijeti datum i odabrati neki na temelju toga ponuđeni 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) prihvatiti/odbiti zahtjev za terminom na temelju raspoloživih prostorija i opreme

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/registirati novog djelatnika (ime, prezime, e-mail adresa, OIB, lozinka)
- (e) mijenjati raspoloživost soba i opreme
- (f) dodavati opremu i sobe
- (g) otkazivati i mijenjati termine, te o tome obavještavati pacijenta

6. Baza podataka (sudionik) može:

- (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:**
 - 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, MBO, lozinka)
 2. Provjera ispravnosti podataka verifikacijom iz baze podataka sustava zdravstvene zaštite
 3. 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 "Profil"
 2. Sustav prikazuje korisnikove osobne podatke

UC4 - Promjena osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Uredi"
 2. Korisnik unosi/mijenja podatke
 3. Korisnik odabire opciju "Spremi promjene"

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 potvrđuje staru lozinku
 3. Korisnik unosi novu lozinku
 4. Korisnik potvrđuje novu lozinku
 5. Sustav korisnika prebacuje na prijavu
 6. Korisnik se prijavljuje s novom lozinkom
- **Opis mogućih odstupanja:**
 - 2.a korisnik unosi neispravnu staru lozinku
 1. Sustav upozorava korisnika o neispravnosti lozinke
 2. Korisnik ponovno unosi 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 "Kreiraj terapiju"
 2. Otvara se stranica za prijavljivanje terapije
 3. Pacijent unosi ime i prezime liječnika, opis oboljenja i zahtijevani postupak liječenja, te odabire vrstu terapije
 4. Pacijent odabire opciju "Pošalji"
 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. Pacijent unosi datum i na temelju njega bira ponuđene slobodne termine
 3. Pacijent opcionalno dodaje napomenu za liječnika
 4. Pacijent šalje zahtjev klikom na "Pošalji"
 5. Spremanje podataka u bazu i slanje maila s podacima o terminu
- **Opis mogućih odstupanja:**
 - 2.a Pacijentu ne odgovara niti jedan termin
 1. Pacijent odustaje od prijave na termin

UC8 - Prikaz nalaza termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Pacijent vidi svoje nalaze

- **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 i prezimena u tražilicu
 2. Djelatnik odabire opciju "Prikaži termine" pored željenog pacijenta
 3. Sustav otvara tablicu termina odabranog pacijenta
 4. Djelatnik evidentira(UC10)/prikazuje evidenciju(UC8)
 5. Djelatnik odabire opciju povratka na popis pacijenata
- **Opis mogućih odstupanja:**
 - 1.a Upisani pacijent ne postoji u sustavu
 1. Prikaz teksta o ne postojanju pacijenta
 2. Djelatnik mijenja upisano ime i prezime ili odustaje od pretraživanja
 - 3.a Odabrani pacijent još nije bio ni na jednoj terapiji
 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 opciju evidentiraj pored termina odabranog pacijenta

2. Otvara se obrazac za evidenciju dolaska na termin
 3. Djelatnik evidentira dolazak odabirom jedne od tri ponuđene opcije(nije došao, došao je)
 4. Djelatnik unosi komentar o odrađenoj terapiji
 5. Djelatnik predaje evidenciju
 6. Povratak na popis termina odabranog pacijenta
- **Opis mogućih odstupanja:**
 - 3.a Djelatnik odustaje od evidentiranja termina
 1. Djelatnik odabire opciju povratka na popis termina odabranog pacijenta

UC11 - Registracija djelatnika

- **Glavni sudionik:** Administrator
- **Cilj:** Registrirati novog liječnika
- **Sudionici:** Baza podataka
- **Preduvjet:**
 - Djelatnik je prijavljen u sustav
 - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
 1. Djelatnik odabire opciju "Dodaj djelatnika"
 2. Prikazuje se stranica za registraciju novog djelatnika
 3. Administrator unosi podatke o djelatniku(ime, prezime, OIB, e-mail adresu i inicijalnu lozinku)
 4. Administrator odabire opciju "Dodaj račun"
 5. Podatci se unose 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

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)
 4. Administrator odabire opciju "Spremi"
 5. Podatci se mijenjaju u bazi podataka
 6. Prikazuje se popis djelatnika
- **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 odabire opciju "Promijeni ovlasti"
 4. Otvara se skočni prozor
 5. Administrator pridjeljuje/uklanja administratorske ovlasti djelatnika
 6. Administrator odabire opciju "Spremi"
 7. Otvara se skočni prozor s upozorenjem i zahtjevom za unos lozinke administratora

8. Administrator unosi lozinku
 9. Administrator odabire opcije "potvrdi"
 10. Sustav potvrđuje lozinku i sprema promjene u bazu podataka
 11. 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 mijenja vrijednost atributa jeAktivan odabranog djelatnika
 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. Unosi podatke o opremi (ime i opis)
 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 pretražuje 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 "Dodaj prostoriju"
 4. Prikazuje se skočni prozor s prostorom za unos podataka o prostoriji
 5. Administrator unosi broj i kapacitet prostorije
 6. Administrator odabire opciju "spremi"
 7. Podatci se spremaju u bazu podataka
 8. Prikazuje se stranica s popisom opreme
- **Opis mogućih odstupanja:**
 - 6.a Administrator želi odustati od unosa prostorije
 1. Administrator odabire opciju "Odustani"

UC18 - Promjena dostupnosti prostorije

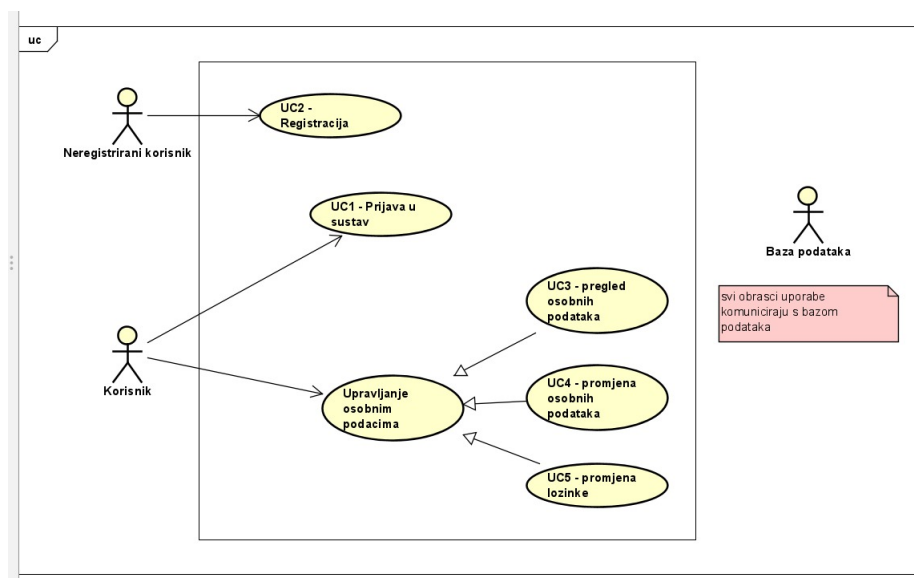
- **Glavni sudionik:** Administrator
- **Cilj:** Promjeniti dostupnost prostorije
- **Sudionici:** Baza podataka
- **Preduvjet:**
 - Djelatnik je prijavljen u sustav
 - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Prostorije"
 2. Otvara se stranica s popisom prostorija
 3. Administrator pretražuje prostorije
 4. Administrator mijenja dostupnost željene prostorije odabirom opcije "Dostupno"/"Nedostupno" koja se nalazi pored podataka o prostoriji
 5. Promjena dostupnosti odabrane prostorije u bazi podataka

UC19 - Obavijest o promjeni termina

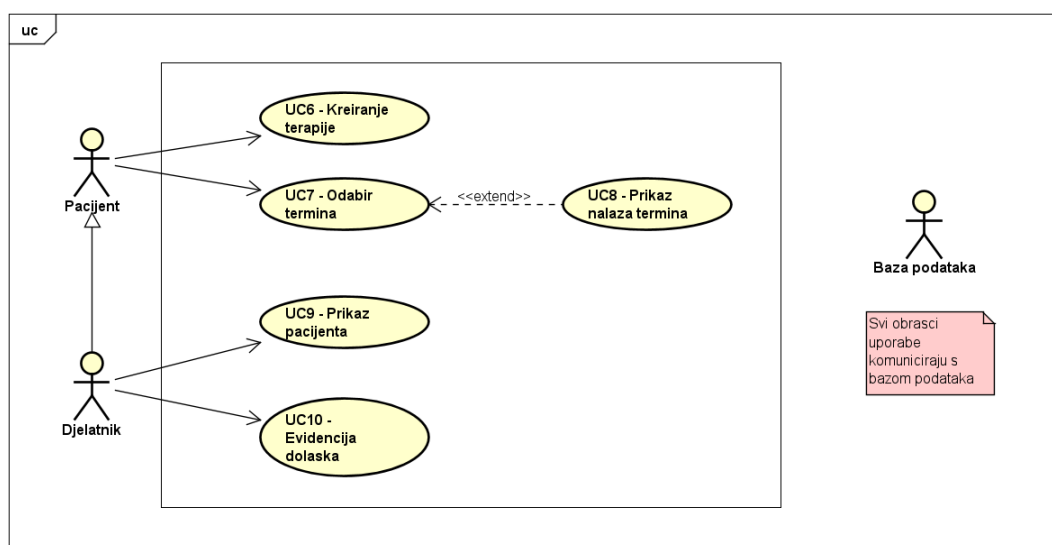
- **Glavni sudionik:** Administrator

- **Cilj:** Unijeti podatke o novom terminu i obavijestiti pacijenta o tome putem e-maila
- **Sudionici:** Baza podataka
- **Preduvjet:**
 - Djelatnik je prijavljen u sustav
 - Djelatnik ima ovlasti administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Promjena termina"
 2. Otvara se stranica s prostorom za podatke o pacijentu i terminu
 3. Administrator pretražuje i odabire pacijenta
 4. Prikazuju se termini odabranog pacijenta koji još nisu evidentirani
 5. Administrator odabire termin koji želi promijeniti
 6. Administrator unosi željene podatke o novom terminu
 7. Prikaz mogućih termina sa željenim podacima
 8. Odabir termina
 9. Administrator opcionalno dodaje komentar
 10. Administrator odabire opciju "Pošalji"
 11. Podaci o terminu se spremaju u bazu podataka
 12. Odabranom pacijentu se šalje e-mail o promjenama i informacijama o novom terminu
- **Opis mogućih odstupanja:**
 - 3.a Administrator unosi nepostojećeg pacijenta/krivo uneseno ime/prezime/MBO pacijenta
 1. Sustav upozorava administratora o nepostojanju pacijenta u sustavu
 2. Administrator mijenja svoj unos ili odustaje od promjene termina

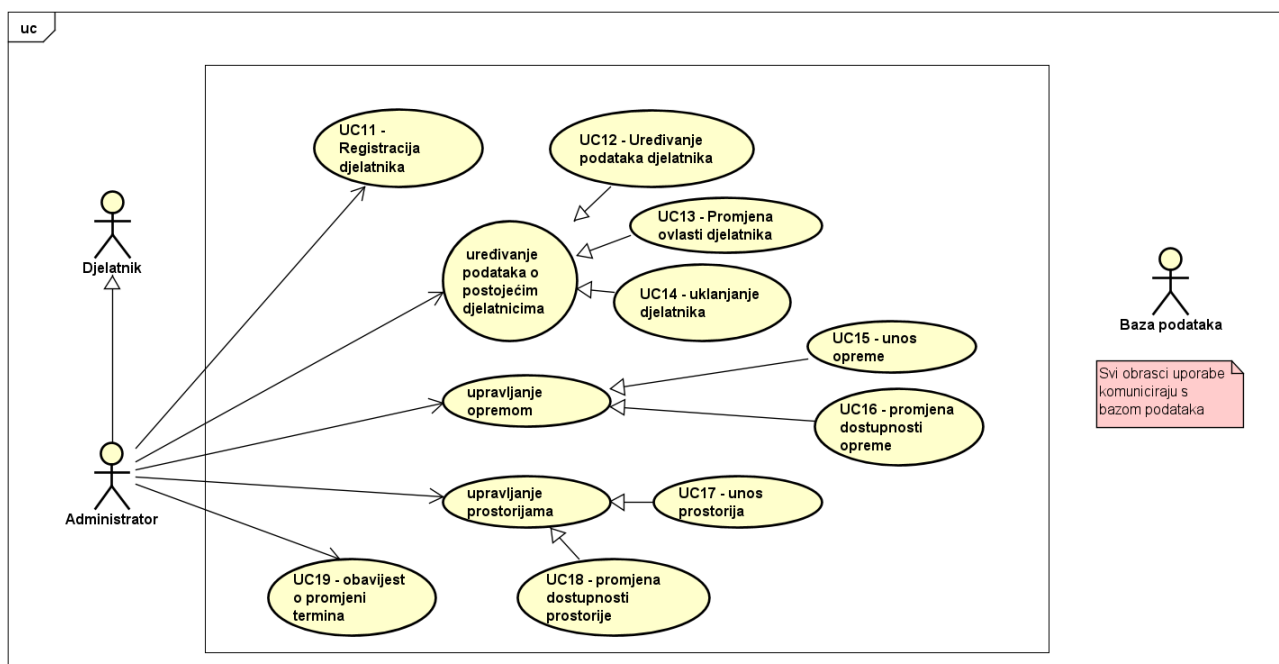
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe - Korisnik i neregistrirani korisnik



Slika 3.2: Dijagram obrasca uporabe - Djelatnik i pacijent

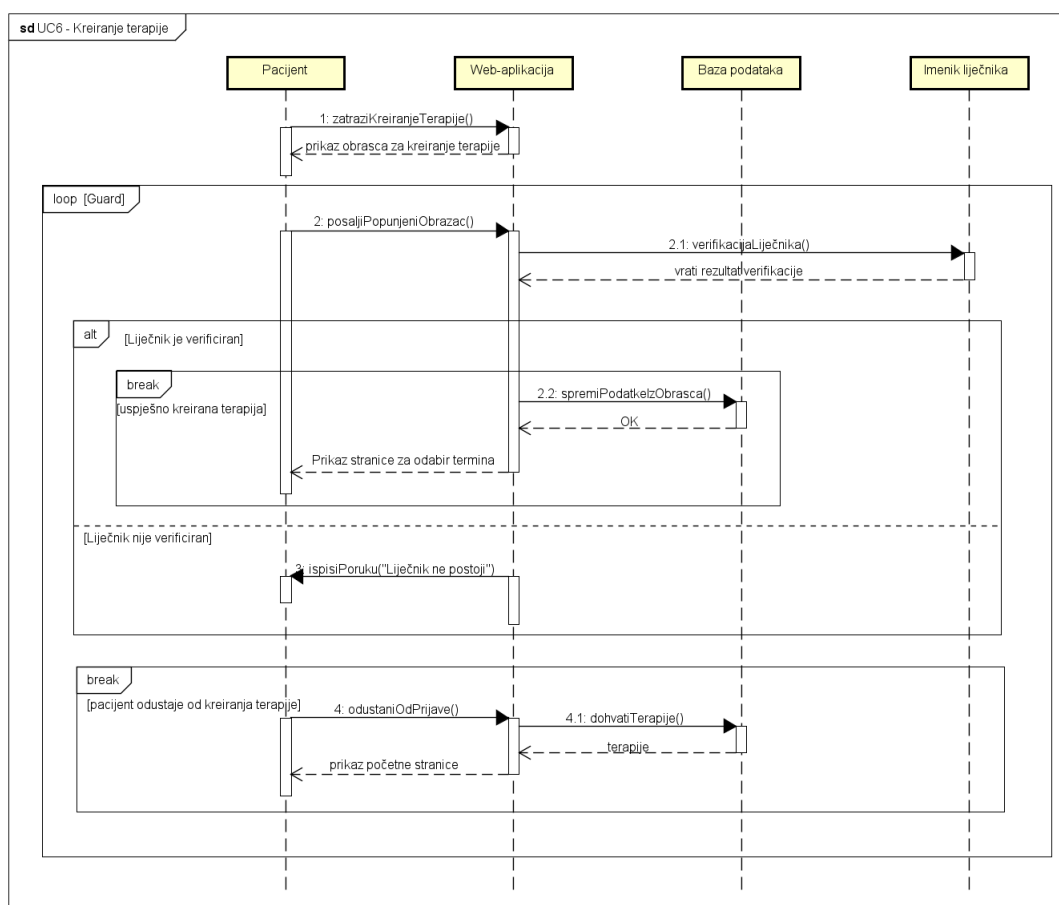


Slika 3.3: Dijagram obrasca uporabe - Administrator

3.1.2 Sekvencijski dijagrami

UC6 - Kreiranje terapije

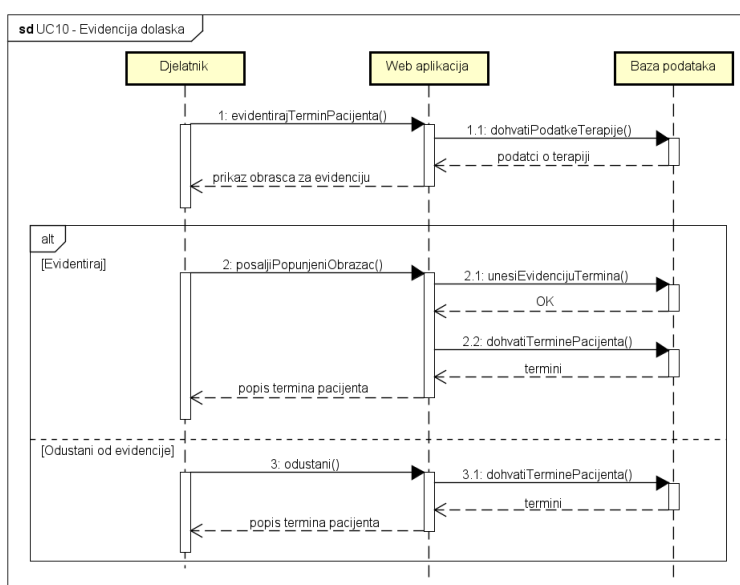
Pacijent šalje zahtjev poslužitelju za naručivanje na terapiju. Web aplikacija mu prikazuje obrazac koji mora ispuniti kako i se naručio na terapiju. Pacijent unosi potrebne podatke (ime i prezime liječnika koji ga je uputio na rehabilitaciju, vrstu terapije, opis oboljenja i zahtijevani postupak liječenja). 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 prikazuje 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 pored termina odabranog pacijenta. Web aplikacija iz baze podataka dohvaća podatke o terapiji kojoj pripada odabrani termin, te nakon toga prikazuje obrazac za evidenciju s dohvaćenim podacima i prostorom za označavanje dolaska i unosom komentara. Djelatnik odabire status(odrađeno/nije odrađeno) 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

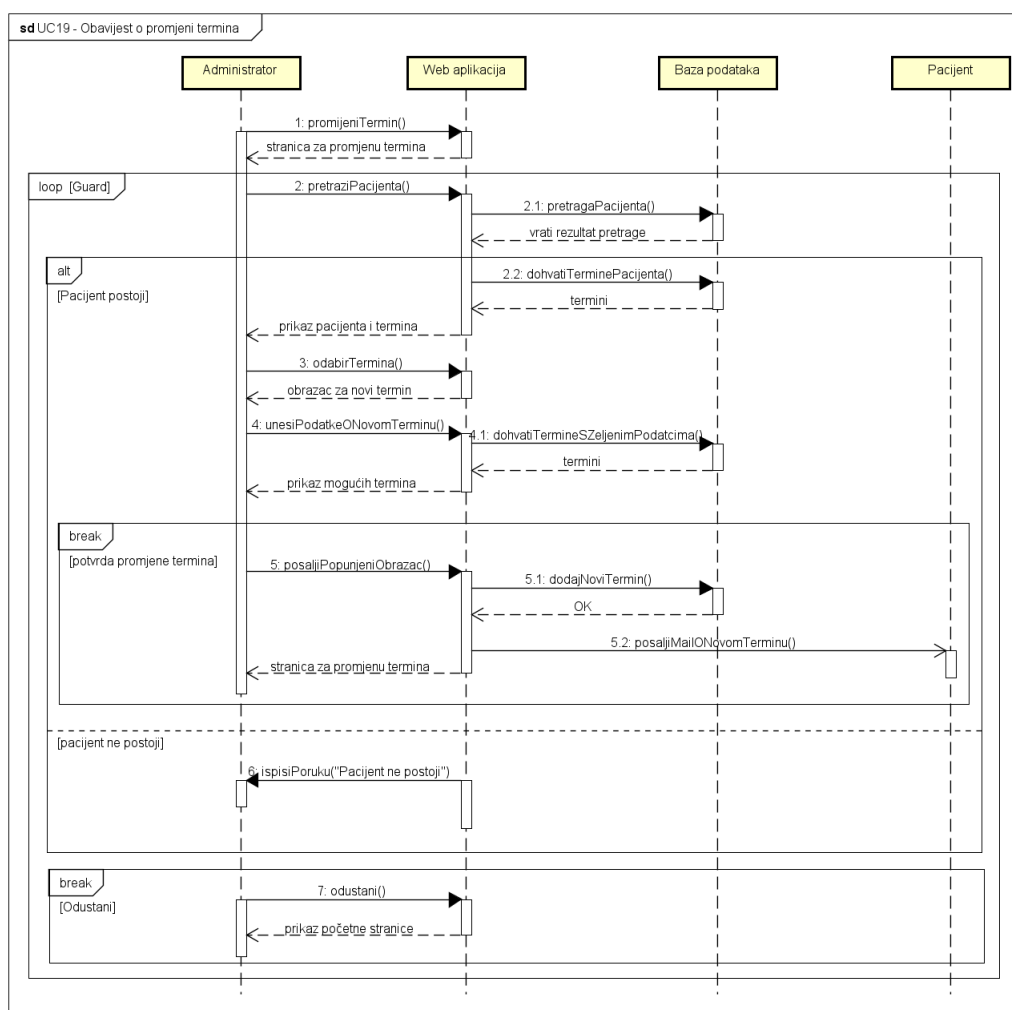
UC19 - Obavijest o promjeni termina

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 potvrđi 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

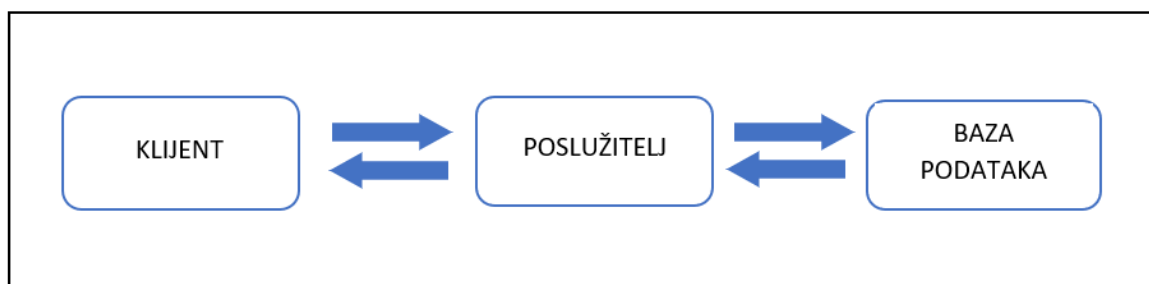
3.2 Ostali zahtjevi

- Sustav mora omogućiti istovremeni pristup više korisnika (100 korisnika istovremeno) uz očuvanje performansi.
- Poslužitelj mora vratiti odgovor na korisnički zahtjev unutar 5 sekundi. iiiiii
HEAD
- 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 attribute: idKorisnika, ime, prezime, datumRodjenja, telefon, email, lozinka. 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
datumRodjenja	DATE	datum rođenja korisnika
telefon	VARCHAR(50)	broj mobilnog telefona korisnika
email	VARCHAR(100)	e-mail adresa korisnika
lozinka	VARCHAR(100)	hash lozinke za prijavu u aplikaciju

Djelatnik:

Entitet Djelatnik je ekskluzivna specijalizacija entiteta Korisnik. Entitet sadrži attribute: 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 attribute: idKorisnika, MBO. Pacijent je povezan s Korisnikom preko atributa idKorisnika (korisnik.idKorisnika). 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 attribute: 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 attribute: 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 attribute: 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 attribute 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 attribute: 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)
idVrste	INT	jedinstveni identifikator vrste uređaja (vrstaUredaja.idVrste)

VrstaUredaja:

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

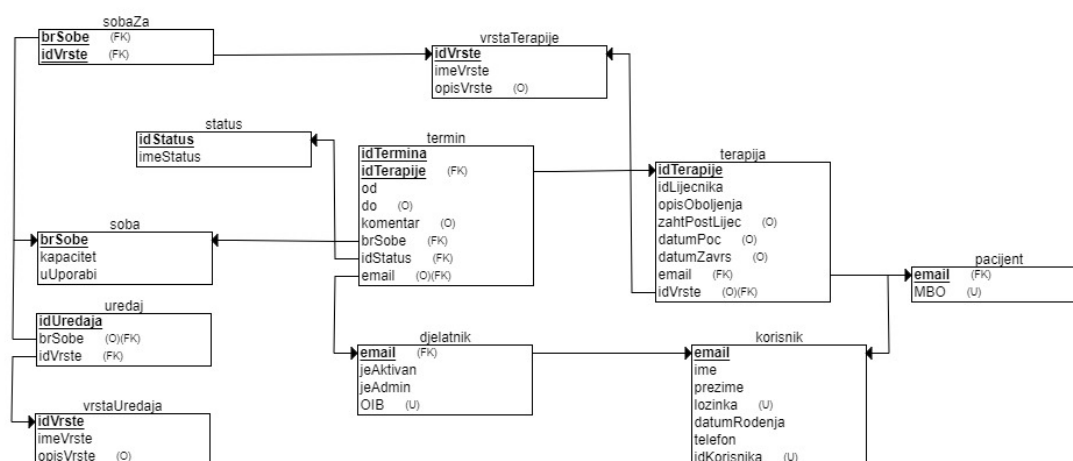
VrstaUredaja		
idVrste	INT	jedinstveni identifikator vrste uređaja
imeVrste	VARCHAR(50)	naziv vrste uređaja
opisVrste	VARCHAR(300)	opis vrste uređaja

Status:

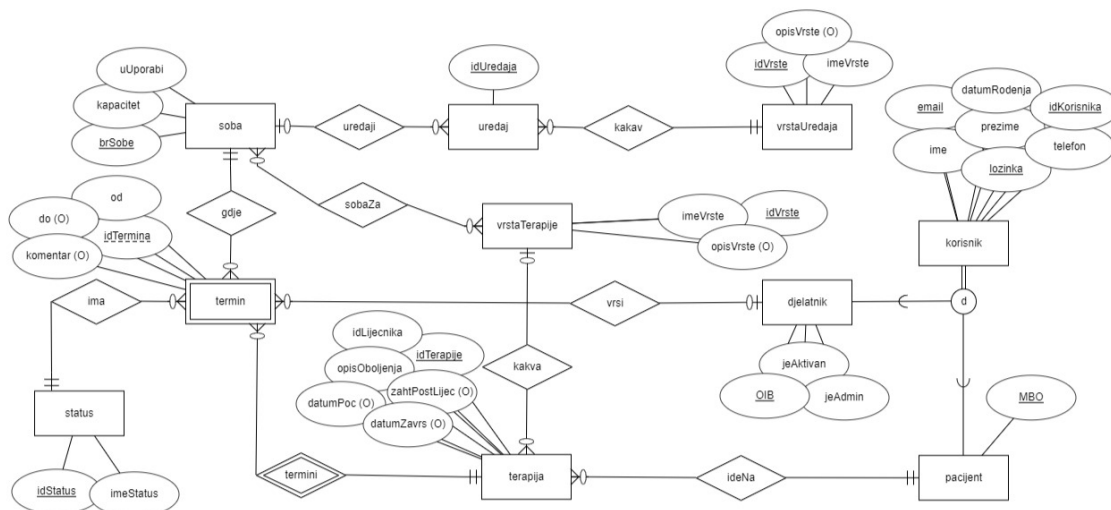
Entitet Status sadrži sve bitne informacije o statusu termina terapije koji je dodijeljen pacijentu. Entitet sadrži attribute: idStatus, imeStatus. Atribut idStatus je primarni ključ.

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

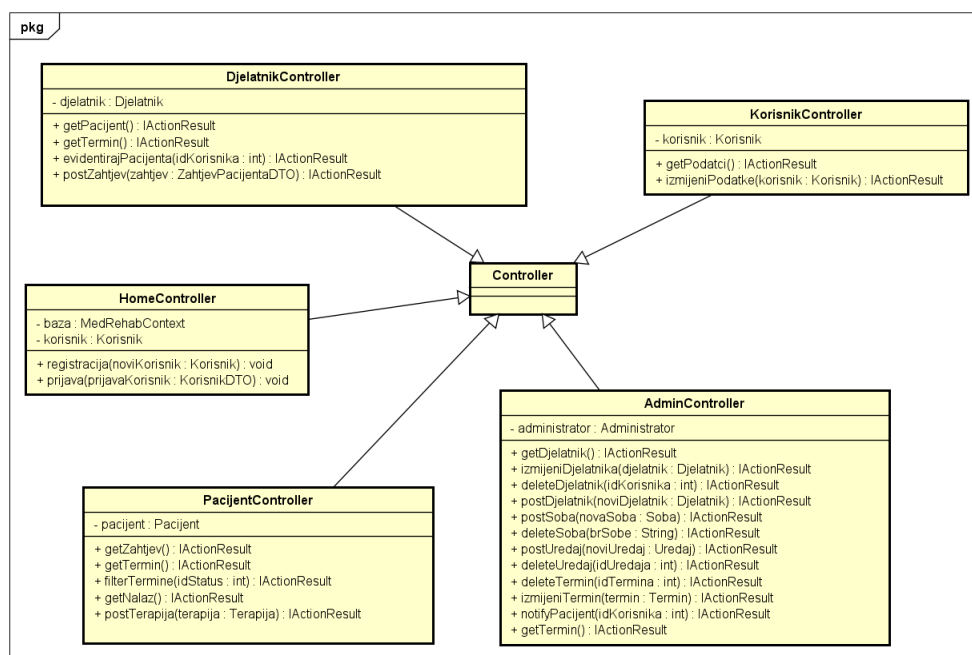


Slika 4.3: ER model baze podataka

4.2 Dijagram razreda

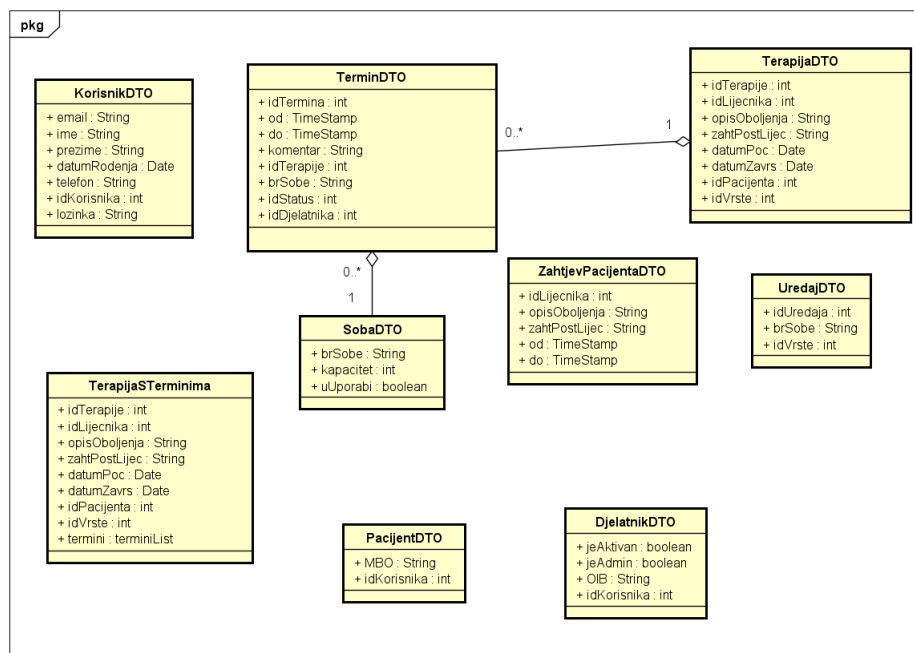
Dijagrami razreda rađeni su po uzoru na MVC obrazac po kojem je naša aplikacija organizirana. Podijeljeni su u 3 dijela: Nadglednik(*Controller*) (4.4), DTO (4.5) i Model (4.6).

Nadglednik 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* treba imati funkcije dobavljanja podataka i izmjene osobnih podataka. *DjelatnikController* mora imati funkcije dobavljanja pacijenta i termina, evidencije pacijenta i dodavanje zahtjeva. *PacijentController* mora imati funkcije dobavljanja zahtjeva, termina i nalaza, filtraciju termina i postavljanje terapije. *AdminController* treba moći dobavljati djelatnike i termine, mijenjati djelatnike i termine, brisati djelatnike, sobe, uređaje i termine, dohvaćati djelatnike, sobe i uređaje i obavještavati pacijente. *HomeController* treba imati pristup bazi i treba imati funkcije registracije i prijave.



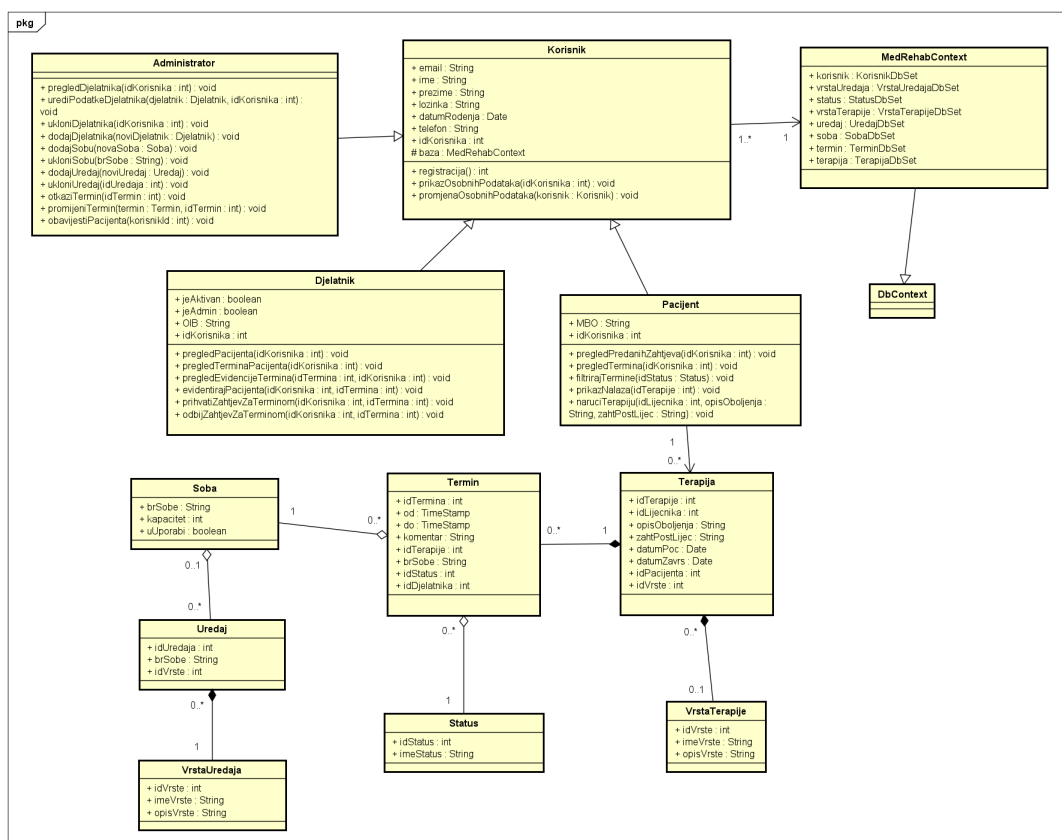
Slika 4.4: Dijagram razreda za dio Controller

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 prihvatanje 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

dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

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.*

5.2 Ispitivanje programskog rješenja

dio 2. revizije

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

5.2.1 Ispitivanje komponenti

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

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

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

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

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

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

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

5.4 Upute za puštanje u pogon

dio 2. revizije

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

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

6. Zaključak i budući rad

dio 2. revizije

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

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

Popis literature

Kontinuirano osvježavanje

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

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Primjer slične aplikacije	7
3.1	Dijagram obrasca uporabe - Korisnik i neregistrirani korisnik	21
3.2	Dijagram obrasca uporabe - Djelatnik i pacijent	21
3.3	Dijagram obrasca uporabe - Administrator	22
3.4	Sekvencijski dijagram za UC6	23
3.5	Sekvencijski dijagram za UC10	24
3.6	Sekvencijski dijagram za UC19	25
4.1	Organizacija sustava	27
4.2	Relacijska shema baze podataka	33
4.3	ER model baze podataka	34
4.4	Dijagram razreda za dio Controller	35
4.5	Dijagram razreda za dio DTO	36
4.6	Dijagram razreda za dio Model	37

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.Zlutar
- 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

Tablica aktivnosti

Kontinuirano osvježavanje

Napomena: Doprinosi 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	5			5			
Opis pojedinih obrazaca	5			5			
Dijagram obrazaca			3				
Sekvencijski dijagrami	4			4			
Opis ostalih zahtjeva	1			1			
Arhitektura i dizajn sustava				2			
Baza podataka	2	2	2	6	2	3	3
Dijagram razreda		4					
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							

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							
Dnevnik sastajanja	0.2						
Zaključak i budući rad							
Popis literature							
UI dizajn					6		
<i>Frontend</i> (prijava i registracija)					30		
Izrada baze podataka							
<i>Backend</i> (prijava i registracija)						10	10
<i>Deploy</i>							
						10	10
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

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.