

# Programsko inženjerstvo

Ak. god. 2023./2024.

## Medicinska rehabilitacija

Dokumentacija, Rev. <1 ili 2>

Grupa: *Proginator*

Voditelj: *EmaBadurina*

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

Nastavnik: <Ime i prezime nastavnika zaduženog za vašu grupu>

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
2.1	Primjeri u $\text{\LaTeX}$ u . . . . .	8
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>11</b>
3.1	Funkcionalni zahtjevi . . . . .	11
3.1.1	Obrasci uporabe . . . . .	12
3.1.2	Sekvencijski dijagrami . . . . .	13
3.2	Ostali zahtjevi . . . . .	14
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>15</b>
4.1	Baza podataka . . . . .	15
4.1.1	Opis tablica . . . . .	16
4.1.2	Dijagram baze podataka . . . . .	21
4.2	Dijagram razreda . . . . .	22
4.3	Dijagram stanja . . . . .	23
4.4	Dijagram aktivnosti . . . . .	24
4.5	Dijagram komponenti . . . . .	25
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>26</b>
5.1	Korištene tehnologije i alati . . . . .	26
5.2	Ispitivanje programskog rješenja . . . . .	27
5.2.1	Ispitivanje komponenti . . . . .	27
5.2.2	Ispitivanje sustava . . . . .	27
5.3	Dijagram razmještaja . . . . .	28
5.4	Upute za puštanje u pogon . . . . .	29
<b>6</b>	<b>Zaključak i budući rad</b>	<b>30</b>
	<b>Popis literature</b>	<b>31</b>

<b>Indeks slika i dijagrama</b>	<b>32</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>33</b>

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Dovršen opis projektnog zadatka	Ema Badurina	13.11.2023.
0.2	Dovršen opis arhitekture	Lucija Lasović	13.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene: Timer - Brojilo vremena	*	15.09.2013.
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.
2.0	Konačni tekst predložka dokumentacije	*	28.09.2013.

*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao*

revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.

## 2. Opis projektnog zadatka

### *dio 1. revizije*

Cilj ovog projekta je izgradnja web aplikacije koja će omogućiti ljudima/bolesnicima 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 porođenja rehabilitacije je svakim radnim danom od ponedjeljka do petka od 8:00 do 20:00 sati.

Aplikacija će razlikovati tri vrste korisnika:

- pacijenta/bolesnika
- liječnika - djelatnika zdravstvene ustanove
- administratora

Prilikom pokretanja web aplikacije svaki korisnik unosi svoj e-mail i lozinku. Ovisno o vrsti korisnika biti će preusmjereni na različite stranice.

*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 ili registracije korisnik je preusmjeren na stranicu gdje se prikazuju njegovi termini i zahtjevi. 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. Zahtjev se sastoji od:

- vremena predaje
- željenog termina
- vrste/tipa rehabilitacije
- liječnika koji je uputio pacijenta na terapiju
- reference na prošlu terapiju
- statusa

Status može biti odobreno, odbijeno, čeka na odobrenje ili isteklo u slučaju da nitko nije odobrio niti odbio termin do željenog vremena terapije. Klikom na opciju *naruči se* pacijent može napraviti novi zahtjev za terminom. Zahtjev popunjava unoseći vrstu rehabilitacije, liječnika koji ga je uputio, ukoliko se radi o ponovljenoj terapiji odabrati će i termin već obavljenog postupka terapije u ustanovi, također u komentar za liječnika dodaje opis svojih oboljenja. Status liječnika koji je izdao uputnicu provjerava se u imeniku liječnika. U odabiru termina biti će prikazano prvih nekoliko slobodnih termina za odabrani tip rehabilitacije i opcije da se unese željeni datum i vrijeme. Nakon odabira željenog termina pacijent predaje zahtjev. Pacijent će o odobrenju ili odbijanju zahtjeva, kao i svim mogućim promjenama biti obaviješten mailom.

*Liječnik* nakon prijavljivanja u sustav ima pregled svih pacijenata i njihovih podataka, 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. Također svaki djelatnik ustanove imati će mogućnost prihvatanja ili odbijanja termina ovisno o raspoloživosti prostorija i opreme. Liječnik koji prihvati zahtjev automatski će se dodati kao liječnik koji će tu terapiju izvoditi.

*Administrator* ima pregled svih pacijenata i djelatnika. Uz ovlasti koje imaju djelatnici 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 broj opreme, dostupne prostorije.

Ovim projektom smanjio bi se opseg posla djelatnika ustanove —

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, prostorije, oprema i djelatnici vidljivi su adminu koji im može mijenjati status za određeno vrijeme, na primjer kada liječnik ode na godišnji odmor može promije-

niti 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.

---

- *potencijalna korist ovog projekta*
- *postojeća slična rješenja (istražiti i ukratko opisati razlike u odnosu na zadani zadatak). Dodajte slike koja predočavaju slična rješenja.*
- *skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje.*
- *mogućnost prilagodbe rješenja*
- *opseg projektnog zadatka*
- *moguće nadogradnje projektnog zadatka*

*Za pomoć pogledati reference navedene u poglavlju „Popis literature“, a po potrebi konzultirati sadržaj na internetu koji nudi dobre smjernice u tom pogledu.*



## 2.1 Primjeri u L<sup>A</sup>T<sub>E</sub>Xu

*Ovo potpoglavlje izbrisati.*

U nastavku se nalaze različiti primjeri kako koristiti osnovne funkcionalnosti L<sup>A</sup>T<sub>E</sub>Xa koje su potrebne za izradu dokumentacije. Za dodatnu pomoć obratiti se asistentu na projektu ili potražiti upute na sljedećim web sjedištima:

- Upute za izradu diplomskog rada u L<sup>A</sup>T<sub>E</sub>Xu - [https://www.fer.unizg.hr/\\_download/repository/LaTeX-upute.pdf](https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf)
- L<sup>A</sup>T<sub>E</sub>X projekt - <https://www.latex-project.org/help/>
- StackExchange za Tex - <https://tex.stackexchange.com/>

podcrtani tekst, **podebljani tekst**, *nagnuti tekst*

primjer primjer primjer primjer primjer primjer

- primjer
- primjer
- primjer
  1. primjer
    - 1.a primjer
    - b primjer
  2. primjer

primjer url-a: <https://www.fer.unizg.hr/predmet/proinz/projekt>

posebni znakovi: # \$ % & { } \_ | < > ^ ~ \

naslov unutar tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

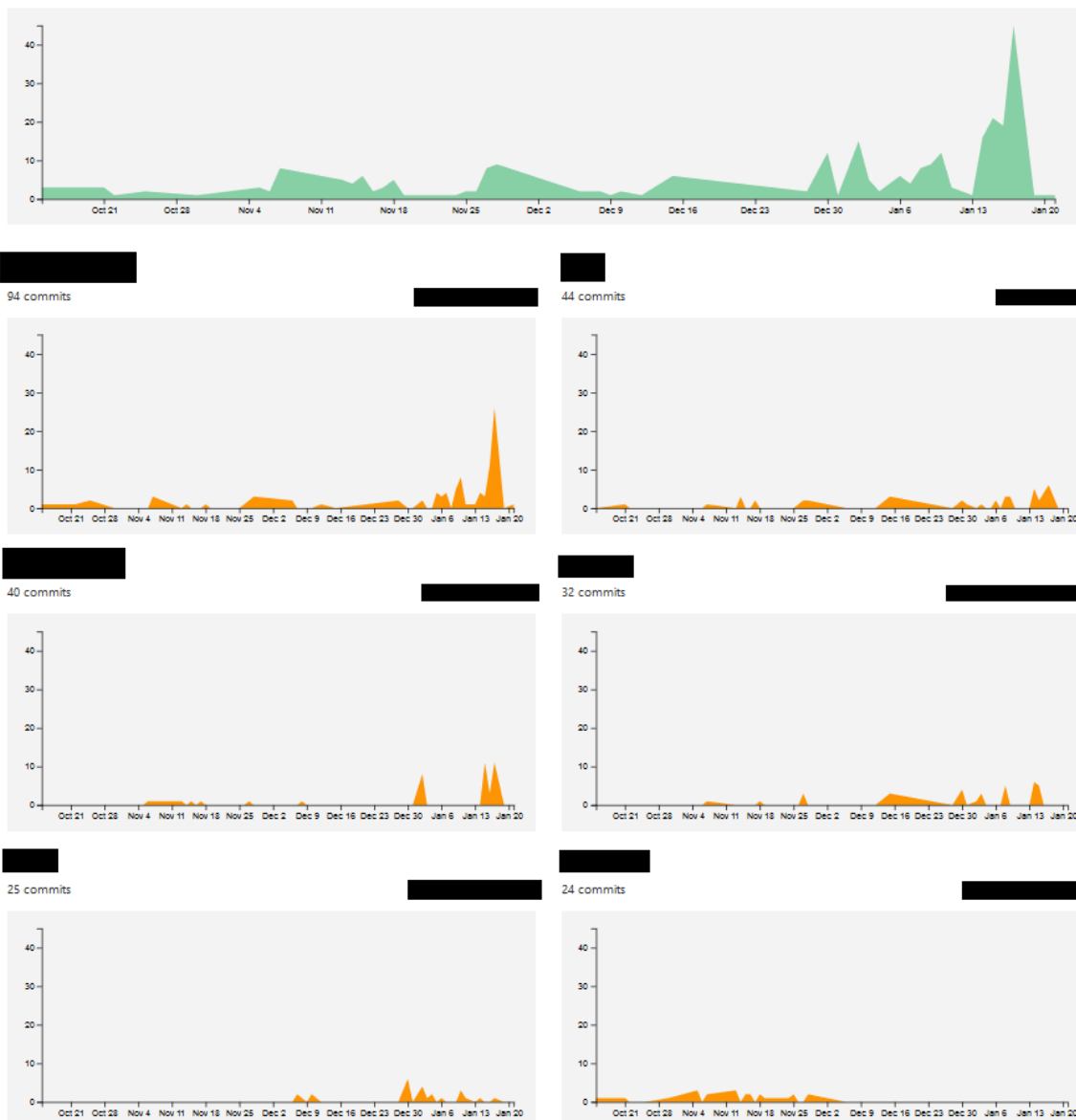
naslov unutar tablice		
ime	VARCHAR	
primjer	VARCHAR	

Tablica 2.1: Naslov s referencom izvan tablice

IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	



Slika 2.1: Primjer slike s potpisom



Slika 2.2: Primjer slike s potpisom 2

Referenciranje slike 2.2 u tekstu.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### *dio 1. revizije*

Navesti **dionike** koji imaju **interes u ovom sustavu** ili **su nositelji odgovornosti**. To su prije svega korisnici, ali i administratori sustava, naručitelji, razvojni tim.

Navesti **aktore** koji izravno **koriste** ili **komuniciraju sa sustavom**. Oni mogu imati inicijatorsku ulogu, tj. započinju određene procese u sustavu ili samo sudioničku ulogu, tj. obavljaju određeni posao. Za svakog aktora navesti funkcionalne zahtjeve koji se na njega odnose.

#### **Dionici:**

1. Pacijent
2. Djelatnik
3. Administrator
4. ...?

#### **Aktori i njihovi funkcionalni zahtjevi:**

1. Aktor 1 (inicijator) može:
  - (a) funkcionalnost 1
  - (b) funkcionalnost 2
    - i. podfunkcionalnost 1
    - ii. podfunkcionalnost 2
  - (c) funkcionalnost 3
2. Aktor 2 (sudionik) može:
  - (a) funkcionalnost 1
  - (b) funkcionalnost 2

### 3.1.1 Obrasci uporabe

#### *dio 1. revizije*

#### **Opis obrazaca uporabe**

*Funkcionalne zahtjeve razraditi u obliku obrazaca uporabe. Svaki obrazac je potrebno razraditi prema donjem predlošku. Ukoliko u nekom koraku može doći do odstupanja, potrebno je to odstupanje opisati i po mogućnosti ponuditi rješenje kojim bi se tijekom obrasca vratio na osnovni tijek.*

#### UC<broj obrasca> -<ime obrasca>

- **Glavni sudionik:** <sudionik>
- **Cilj:** <cilj>
- **Sudionici:** <sudionici>
- **Preduvjet:** <preduvjet>
- **Opis osnovnog tijeka:**
  1. <opis korak jedan>
  2. <opis korak dva>
  3. <opis korak tri>
  4. <opis korak četiri>
  5. <opis korak pet>
- **Opis mogućih odstupanja:**
  - 2.a <opis mogućeg scenarija odstupanja u koraku 2>
    1. <opis rješenja mogućeg scenarija korak 1>
    2. <opis rješenja mogućeg scenarija korak 2>
  - 2.b <opis mogućeg scenarija odstupanja u koraku 2>
  - 3.a <opis mogućeg scenarija odstupanja u koraku 3>

#### **Dijagrami obrazaca uporabe**

*Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.*

### 3.1.2 Sekvencijski dijagrami

#### *dio 1. revizije*

*Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava (max. 4 dijagrama). Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.*

## 3.2 Ostali zahtjevi

### *dio 1. revizije*

Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju **kako se sustav treba ponašati** i koja **ograničenja** treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.

## 4. Arhitektura i dizajn sustava

### *dio 1. revizije*

*Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremišta podataka, mrežne protokole, globalni upravljački tok i sklopovsko-programске zahtjeve. Po točkama razraditi i popratiti odgovarajućim skicama:*

- *izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zašto ste baš odabrali takvu arhitekturu)*
- *organizaciju sustava s najviše razine apstrakcije (npr. klijent-poslužitelj, baza podataka, datotečni sustav, grafičko sučelje)*
- *organizaciju aplikacije (npr. slojevi frontend i backend, MVC arhitektura)*

### 4.1 Baza podataka

#### *dio 1. revizije*

*Potrebno je opisati koju vrstu i implementaciju baze podataka ste odabrali, glavne komponente od kojih se sastoji i slično. 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
- Uloga



- Pacijent
- Terapija
- VrstaTerapije
- Termin
- Status
- Oprema
- ZauzetaOprema
- Soba
- Uredaj
- VrstaUredaja

#### 4.1.1 Opis tablica

##### Korisnik:

Entitet Korisnik sadrži sve bitne informacije o korisniku aplikacije. Entitet sadrži attribute: idKorisnika, ime, prezime, datumRodnja, adresa, email, lozinka. 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
datumRodnja	DATE	datum rođenja korisnika
adresa	VARCHAR(100)	adresa mjesta stanovanja korisnika
email	VARCHAR(50)	e-mail adresa korisnika
lozinka	VARCHAR(50)	hash lozinke za prijavu u aplikaciju

##### Djelatnik:

Entitet Djelatnik je ekskluzivna specijalizacija entiteta Korisnik. Entitet sadrži attribute: idKorisnika, aktivan, idUloge.

Djelatnik		
idKorisnika	INT	jedinstveni identifikator korisnika (korisnik.idKorisnika)
aktivan	CHAR(2)	označava radni odnos liječnika i zdravstvene ustanove
idUloge	INT	jedinstveni identifikator uloge (uloga.idUloge)

**Uloga:**

Entitet Uloga služi za razlikovanje djelatnika zdravstvene ustanove. U aplikaciji razlikujemo liječnike od administratora. Entitet sadrži attribute: idUloge, imeUloge.

Uloga		
idUloge	INT	jedinstveni identifikator uloge
imeUloge	VARCHAR(50)	naziv uloge (liječnik ili administrator)

**Pacijent:**

Entitet Pacijent je ekskluzivna specijalizacija entiteta Korisnik. Entitet sadrži attribute: idKorisnika, MBO.

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, idDjelatnika, idVrste.

Terapija		
idTerapije	INT	jedinstveni identifikator terapije
idLijecnika	INT	jedinstveni identifikator liječnika
opisOboljenja	VARCHAR(200)	opis oboljenja pacijenta
zahtPostLijec	VARCHAR(200)	zahtjev za postupkom liječenja (terapijom)
datumPoc	DATE	datum početka terapije
datumZavrs	DATE	datum završetka terapije
idPacijenta	INT	jedinstveni identifikator pacijenta (korisnik.idKorisnika)
idDjelatnika	INT	jedinstveni identifikator djelatnika (korisnik.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.

VrstaTerapije		
idVrste	INT	jedinstveni identifikator vrste terapije
imeVrste	VARCHAR(50)	naziv vrste terapije
opisVrste	VARCHAR(200)	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.)

Termin		
idTermina	INT	jedinstveni identifikator termina terapije

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Termin		
idTerapije	INT	jedinstveni identifikator terapije (terapija.idTerapije)
od	DATETIME	datum i vrijeme početka termina
do	DATETIME	datum i vrijeme završetka termina
komentar	VARCHAR(200)	komentar liječnika o napretku terapije
brSobe	VARCHAR(10)	jedinstveni identifikator sobe (soba.brSobe)
idStatus	INT	jedinstveni identifikator statusa (status.idStatus)
idKorisnika	INT	jedinstveni identifikator korisnika (korisnik.idKorisnika)

#### Soba:

Entitet Soba sadrži sve bitne informacije o sobi u kojoj se provodi neka terapija. Entitet sadrži atribute: brSobe, kapacitet.

Soba		
brSobe	VRACHAR(10)	jedinstveni identifikator sobe
kapacitet	INT	kapacitet sobe, broj pacijenata koji istovremeno mogu biti na terapiji u nekoj sobi

#### Uredaj:

Entitet Uredaj sadrži sve bitne informacije o uređaju koji se koristi u nekoj terapiji. Entitet sadrži atribute: idUredaja, brSobe, 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 koja se primjenjuje u nekoj terapiji. Entitet sadrži attribute: idVrste, imeVrste, opisVrste.

VrstaUredaja		
idVrste	INT	jedinstveni identifikator vrste uređaja
imeVrste	VARCHAR(50)	naziv vrste uređaja
opisVrste	VARCHAR(200)	opis vrste uređaja

**Oprema:**

Entitet Oprema sadrži sve bitne informacije o opremi koja je na raspolaganju u zdravstvenoj ustanovi. Entitet sadrži attribute: idOpreme, imeOpreme, opisOpreme.

Oprema		
idOpreme	INT	jedinstveni identifikator opreme
imeOpreme	VARCHAR(50)	naziv opreme (npr. štake)
opisOpreme	VARCHAR(200)	opis namjene opreme

**ZauzetaOprema:**

Entitet ZauzetaOprema sadrži sve bitne informacije o alokaciji opreme po terminima. Entitet sadrži attribute: idTermina, idTerapije, idOpreme.

ZauzetaOprema		
idTermina	INT	jedinstveni identifikator termina (termin.idTermina)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

ZauzetaOprema		
idTerapije	INT	jedinstveni identifikator terapije (terapija.idTerapije)
idOpreme	INT	jedinstveni identifikator opreme (oprema.idOpreme)

**Status:**

Entitet Status sadrži sve bitne informacije o statusu termina terapija koji je dodijeljen nekom pacijentu. Entitet sadrži atribute: idStatus, imeStatus.

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

*Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ*

#### 4.1.2 Dijagram baze podataka

*U ovom potpoglavlju potrebno je umetnuti dijagram baze podataka. Primarni i strani ključevi moraju biti označeni, a tablice povezane. Bazu podataka je potrebno normalizirati. Podsjetite se kolegija "Baze podataka".*

## 4.2 Dijagram razreda

*Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.*

### **dio 1. revizije**

*Prilikom prve predaje projekta, potrebno je priložiti potpuno razrađen dijagram razreda vezan uz **generičku funkcionalnost** sustava. Ostale funkcionalnosti trebaju biti idejno razrađene u dijagramu sa sljedećim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazivi atributa razreda, veze i odnosi između razreda.*

### **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 Selenium<sup>1</sup>. 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.*

---

<sup>1</sup><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 slike s potpisom . . . . .	9
2.2	Primjer slike s potpisom 2 . . . . .	10

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

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

#### 1. sastanak

- Datum: u ovom formatu: 13. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

#### 2. sastanak

- Datum: u ovom formatu: 13. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

## Tablica aktivnosti

### Kontinuirano osvježavanje

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

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<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](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*