

Programsko inženjerstvo

Ak. god. 2023./2024.

Medicinska rehabilitacija

Dokumentacija, Rev. 1

Grupa: Teletabisi

Voditelj: Benjamin Gregov

Datum predaje: 17. 11. 2023.

Nastavnik: Miljenko Krhen

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	24
3.2 Ostali zahtjevi	29
4 Arhitektura i dizajn sustava	30
4.1 Baza podataka	32
4.1.1 Opis tablica	32
4.1.2 Dijagram baze podataka	36
4.2 Dijagram razreda	37
4.3 Dijagram stanja	41
4.4 Dijagram aktivnosti	44
4.5 Dijagram komponenti	47
5 Implementacija i korisničko sučelje	48
5.1 Korištene tehnologije i alati	48
5.2 Ispitivanje programskog rješenja	49
5.2.1 Ispitivanje komponenti	49
5.2.2 Ispitivanje sustava	49
5.3 Dijagram razmještaja	50
5.4 Upute za puštanje u pogon	51
6 Zaključak i budući rad	52
Popis literature	53
Indeks slika i dijagrama	55

Dodatak: Prikaz aktivnosti grupe

56

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Neven Pralas	30.10.2023.
0.2	Napisan opis projektnog zadatka	Neven Pralas	02.11.2023.
0.3	Opisani funkcionalni zahtjevi i svi aktori sustava	Neven Pralas	03.11.2023.
0.4	Dodan prvi dio obrazaca uporabe	Neven Pralas	05.11.2023.
0.4.1	Dodan preostali dio obrazaca uporabe	Neven Pralas	06.11.2023.
0.5	Dodani sekvencijski dijagrami i ostali zahtjevi	Tin Ogrizek	09.11.2023.
0.5.1	Popravljeni sekvencijski dijagrami	Tin Ogrizek	12.11.2023.
0.6	Izmijenjeni početni dijelovi dokumentacije	Neven Pralas	13.11.2023.
0.7	Opisana arhitektura računala	Filip Posavec i Neven Pralas	14.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.8	Opisana baza podataka i dijagrami baze podataka	Bruno Petković	15.11.2023.
0.9	Dodani dijagrami razreda	Tin Ogrizek	16.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Tin Ogrizek	17.11.2023.
1.1	Dodani dijagrami stanja	Neven Pralas	13.01.2024.
1.2	Dodani dijagrami aktivnosti	Neven Pralas	14.01.2024.
1.3	Dodan dijagram komponenti	Neven Pralas	14.01.2024.
1.4	Manja revizija (uvod+funkcionalni zahtjevi)	Neven Pralas	18.01.2024.
1.4	Ažuriranje dijagrama (sekvencijski i dijagram razreda)	Tin Ogrizek	19.01.2024.
2.0	Konačni tekst predložka dokumentacije	*	28.09.2013.

2. Opis projektnog zadatka

Uvod

Cilj projekta "Medicinska rehabilitacija" je razvoj programske podrške za stvaranje istoimene web aplikacije koja će omogućiti korisnicima jednostavno naručivanje za fizikalnu terapiju i medicinsku rehabilitaciju. Korisnici u našem slučaju su pacijenti koji su se ozlijedili te trebaju stručnu pomoć. Zadatak nam je omogućiti intuitivno korisničko sučelje kako bi olakšali korištenje aplikacije pacijentima, ali i zdravstvenim djelatnicima.

Glavna motivacija bila nam je modernizacija hrvatskog zdravstva. Vjerujemo kako pacijentima nije drago zvati mobitelom svoje doktore znajući da su pretrpani poslom, a također im je dosta čekanje na odgovor zdravstvenog djelatnika kada im se pošalje mail. S druge strane nije ni zdravstvenim djelatnicima lako odgovarati na sve te silne poruke. Ovom aplikacijom bismo omogućili pacijentima da biraju dostupan termin koji njima najbolje odgovara, a doktore poštedili dodatnog posla.

Između ostalog ova bi aplikacija omogućivala liječnicima praćenje poboljšanja zdravstvenog stanja pacijenata. Ako se pacijenti prvi puta naručuju za terapiju potrebna je registracija, a svaki sljedeći put je nužna prijava u sustav. Za registraciju potrebni su sljedeći podaci:

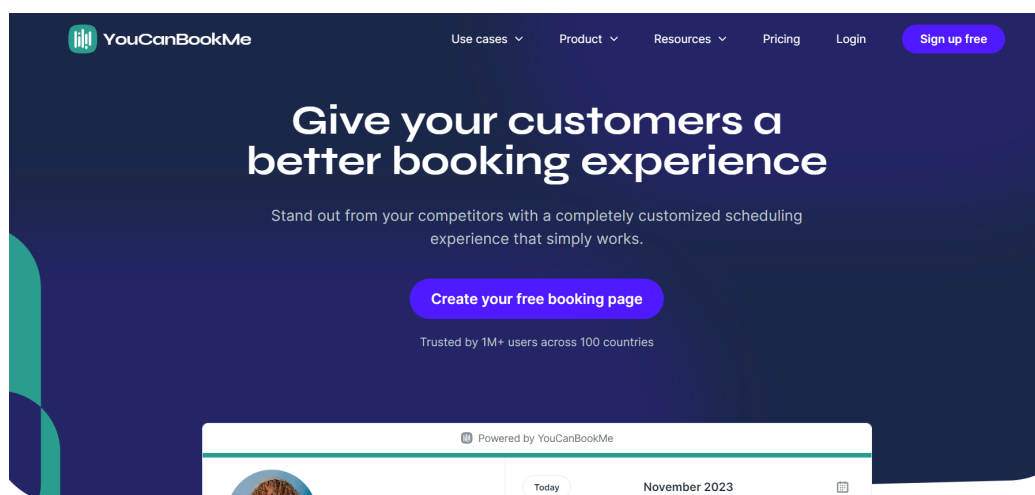
- *ime i prezime*
- *korisničko ime*
- *lozinka*
- *e-mail adresa*
- *OIB*
- *spol*
- *datum rođenja*

Administrator gornje podatke mora verificirati iz središnjeg informacijskog sustava zdravstvene zaštite. Pacijenti prilikom zakazivanja termina moraju navesti vrstu i opis svojih oboljenja, zahtijevani postupak liječenja te vrijeme koje bi im odgovaralo za termin. Ako se ne radi o prvom dolasku u ustanovu zahtijeva se i prikaz reference na već obavljeni postupak u toj ustanovi. Zdravstveni djelatnici

dodjeljuju termine tako da vode brigu o ukupnom kapacitetu opreme/uređaja, kapacitetu prostorija, osobnim mogućnostima te trajanju svakog zahvata. Nakon dobivenog termina pacijentu dolazi e-pošta sa terminom te ostalim dodatnim informacijama. U slučaju neočekivanih promjena djelatnik može kontaktirati bolesnika putem e-pošte. Radno vrijeme ustanove u kojoj se provodi rehabilitacija je svakim radnim danom od 8 do 20 sati. Ustanova ima pravo odrediti određene slobodne dane na temelju blagdana ili praznika.

Primjer sličnog rješenja

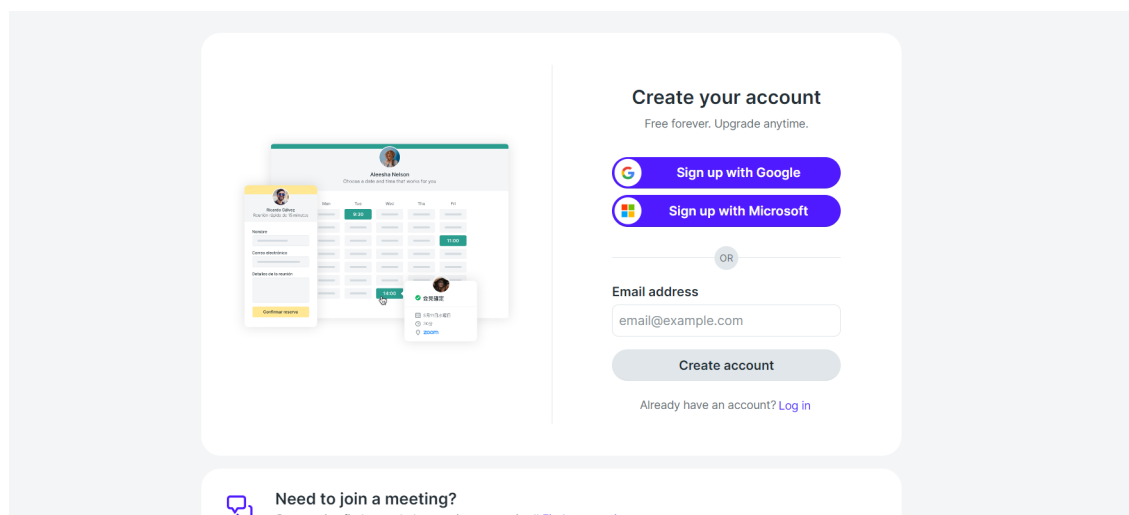
Primjer sličnog rješenja je web aplikacija "YouCanBookMe". To je aplikacija koja omogućuje čovjeku koji prima klijente da postavi svoje termine u kojima je dostupan, a klijentima odabir termina koji njima odgovaraju. Sličnost s našim sustavom je ta što će zdravstveni djelatnik moći određivati u kojim terminima je dostupan, a klijent će moći odabrati jedan od tih dostupnih termina. Ova web aplikacije je općenamjenska, dok naša ima primjenu u medicini.



Slika 2.1: Naslovna stranica "YouCanBookMe" aplikacije

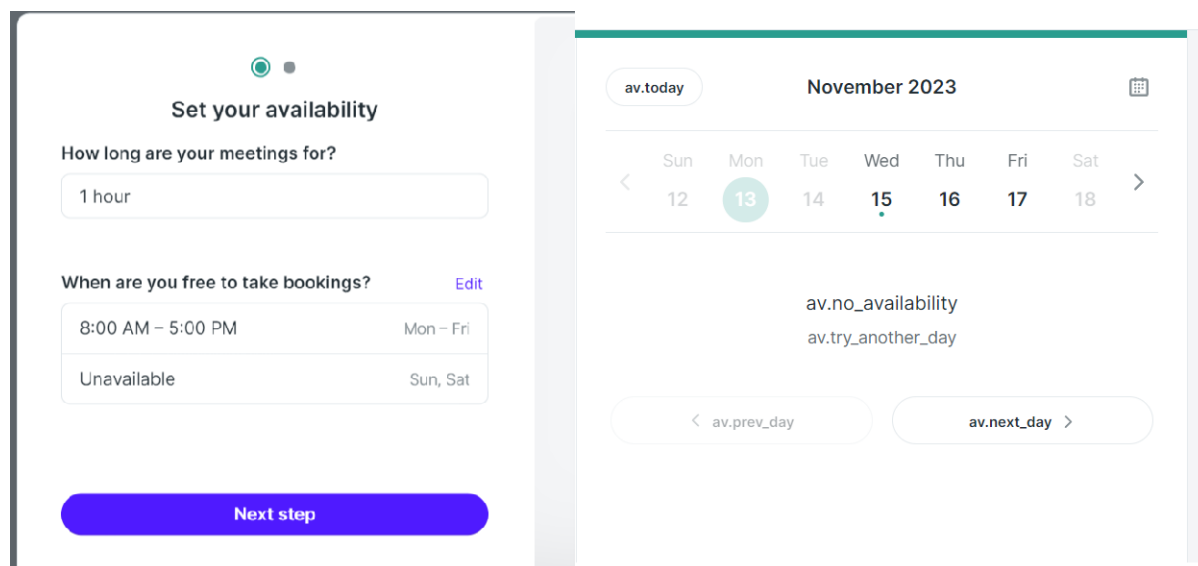
Registracija je omogućena putem upisivanja email adrese, lozinke i ostalih podataka. Neki od važnijih preostalih podataka su ime i prezime osobe. Sigurnost podataka ostvarena je tako da korisnik na svoj email dobiva sigurnosni kod koji treba upisati kako bi uspješno bio registriran.

Prijava u sustav je omogućena tek nakon registracije korisnika, a potrebni podaci koje trebamo upisati su email i lozinka. Izgled stranice za prijavu je vrlo sličan izgledu stranice za registraciju.



Slika 2.2: Primjer registracije, klikom na dugme "Create account" kreiramo korisnički račun

Korisnik koji se prijavio ima pravo postavljati termine u kojima je slobodan ili se prijavljivati na termine nekih drugih korisnika (zavisi o razlogu zašto koristi "YouCanBookMe" aplikaciju). U slučaju postavljanja termina odabire koliko mu traje pojedini sastanak sa klijentom te za pojedini dan određuje u kojem vremenu je slobodan ili zauzet za sastanak. Ovako slično će u našem primjeru moći raditi zdravstveni djelatnik.



Slika 2.3: Primjer postavljanja dostupnih termina

U slučaju da se radi o klijentu koji odabire termin, on ima opciju izabrati samo neke od dostupnih termina. Nakon odabira ispunjava svoje podatke još jedanput

kako bi organizator sastanka znao osnovne podatke o klijentu. U našem primjeru će na ovaj način funkcionirati pacijent uz iznimku da neće ponovno morati ispunjavati svoje podatke već će prilikom prijave za dostupni termin njegove osobne podatke moći vidjeti zdravstveni djelatnik.

The image shows a mobile application interface for patient registration. At the top, there is a teal header bar with a back button labeled 'cm.back' and a title 'bf.toolbar_title'. Below the header, there is a form with three input fields: 'First name (bf.required)', 'Last name', and 'Email (bf.required)'. Each field has a light gray border and a small 'x' icon on the right. At the bottom of the form is a large teal button labeled 'bf.confirm'.

Slika 2.4: Primjer ispunjavanja podataka nakon odabira jednog od dostupnih termina

Dodatno pojašnjenje

Korisnici sustava su:

- *pacijenti*
- *zdravstveni djelatnici*
- *administrator*

Svaki od te tri skupine korisnika ima različita dopuštenja. Djelatnici ustanove mogu vidjeti sve bolesnike i tretmane koji su im dodijeljeni. Nakon svakog obavljenog postupka djelatnik u sustav registrira pacijenta, a nakon završenog ciklusa oslobađa opremu. Pacijenti mogu vidjeti samo svoje tretmane. Administrator upravlja čitavim sustavom. Njemu su dostupni svi podaci i može definirati sve što je potrebno za ispravan rad sustava.

Za unesene podatke prilikom registracije u aplikaciju osigurat ćemo provjeru konzistentnosti s podacima iz baze podataka. Nakon toga korisnik postaje pacijent, izabire datum te ima mogućnost otkazati termin 1 dan prije početka tog termina.

Opreme, odnosno uređaji koje posjeduje ustanova se kategoriziraju. Spomenuli smo da slobodni termini ovise o ukupnom kapacitetu opreme/uređaja. Naravno

da nije isto ako je čovjeku oštećena ruka ili noga. Tu se radi o drugoj opremi koju će zdravstveni djelatnik morati upotrijebiti za liječenja pacijenta. Dakle, korisnik kojemu je ozljeđena ruka neće zauzimati opremu korisniku kojemu je ozljeđena noga. Očekujemo da će broj oprema/uređaja za određeni dio tijela ipak biti veći od jedan, no svakako će biti ograničen.

Korisnik se u sustav prijavljuje upisivanjem korisničkog imena te svoje lozinke koja mora zadovoljavati određena pravila. Lozinka mora imati minimalno 8 znakova, od čega mora biti minimalno jedno veliko slovo te jedna znamenka. Ako korisnik prilikom registracije i smišljanja lozinke nije zadovoljio navedena pravila, dobit će upozorenje što mu nedostaje za ispravnu lozinku.

Aplikacija će imati obavezno početnu stranicu u kojoj će moći odabrati želi li se prijaviti ili registrirati u sustav. Zatim stranicu za prijavu i stranicu za registraciju. Na kraju je omogućen odlazak (nakon prijave korisnika s određenom ulogom) na povlaštenu stranicu za pacijente, zdravstvene djelatnike ili za administratore.

Opseg projekta

Na projektu radi 7 osoba, studenata FER-a. Projekt se radi u edukativne svrhe provjere znanja studenata u sklopu predmeta "Programsko inženjerstvo". Vremensko ograničenje za prvu verziju projekta je 7 tjedana, a druga verzija (odnosno cijeli projekt) mora biti isporučen u roku od 14 tjedana od početka rada. Ukupan broj potrošenog vremena otići će najviše na sami rad oko projekta, ali značajan dio vremena otići će i na sastanke uživo gdje će tim redovito raspraviti o tjednom napretku.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik (naručitelj)
2. Pacijenti
3. Zdravstveni djelatnici
4. Administrator
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) se registrirati u sustav kao pacijent te tako stvoriti novi korisnički račun za koji su mu potrebni ime, prezime, korisničko ime, e-mail adresa, OIB, lozinka, spol i datum rođenja.
 - (b) pregledati naslovnu stranicu koja ga može uputiti na stranicu za registraciju ili stranicu za prijavu
2. Pacijent (inicijator) može:
 - (a) prijaviti se u sustav
 - (b) odabrati željeni termin dolazaka i opisati svoje zdravstvene poteškoće
 - (c) otkazati zakazani termin u roku od 24 sata prije samoga termina te potvrditi ili odbiti pomaknuti termin zbog novonastale okolnosti
 - (d) primiti elektroničku poštu o zakazanom terminu i novonastalim okolnostima
 - (e) pogledati popis svojih prihvaćenih termina te podatke vezane uz termin (doktor koji će ga liječiti, oprema koja će se koristiti, soba u kojoj će se termin odvijati i vrijeme termina)
3. Zdravstveni djelatnik (inicijator) može:

- (a) prijaviti se u sustav
- (b) pregledavati nepotvrđene termine pacijenata u svojoj smjeni
- (c) pregledavati potvrđene termine koje on mora odraditi
- (d) direktno kontaktirati pacijenta putem elektroničke pošte u slučaju neočekivanih promjena (i predložiti im novi termin)
- (e) vidjeti sve pacijente (uključujući njihove osobne podatke) i sve tretmane koji su im dodijeljeni
- (f) otkazati zakazani termin u roku od 24 sata prije samoga termina te potvrditi ili odbiti pomaknuti termin zbog novonastale okolnosti
- (g) obrisati termin ako smatra da termin nije potrebno izvršiti (čime taj termin postaje nevidljiv i ostalim zdravstvenim djelatnicima)
- (h) dodijeliti prilikom prihvatanja termina opremu prikladnu opisu zdravstvenih poteškoća samoga pacijenta (oprema je vezana uz sobu)

4. Administrator (inicijator) može:

- (a) verificirati podatke iz središnjeg informacijskog sustava zdravstvene zaštite
- (b) pristupiti popisu korisnika te njihovim osobnim podacima
- (c) stvoriti zdravstvenog djelatnika u sustavu
- (d) maknuti korisnika iz sustava (i pacijenta i zdravstvenog djelatnika)
- (e) pregledavati korisnike (pacijente i zdravstvene djelatnike) i filtrirati ih po nekim parametrima (npr. po spolu)

5. Baza podataka (sudionik) može:

- (a) pohraniti podatke o korisnicima i njihovim dopuštenjima
- (b) pohraniti podatke o terminima (slobodnim i zauzetim)
- (c) pohraniti podatke o opremama/uslugama (slobodnim i zauzetim)
- (d) pohraniti podatke o sobama (slobodnim i zauzetim)

3.1.1 Obrasci uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registriranje
 2. Korisnik unosi potrebne podatke
 3. Korisnik pritišće dugme da potvrdi registraciju
 4. Korisnik bude poslan na stranicu za prijavu u sustav
- **Opis mogućih odstupanja:**
 - 2.a Odabir zauzetog e-maila i/ili korisničkog imena i/ili OIBa, unos neispravnog e-maila ili unos korisničkog podatka u zabranjenom formatu
 1. Korisnik dobiva obavijest o neuspješnoj registraciji i vraća ga na stranicu za registraciju
 2. Korisnik mijenja krivo unešene podatke ili odustaje od registracije

UC2 - Prijava u sustav

- **Glavni sudionici:** Pacijent, Zdravstveni djelatnik, Administrator
- **Cilj:** Pristupiti uslugama koje pruža naš sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prijavu
 2. Korisnik unosi korisničko ime i lozinku
 3. Korisnik pritišće dugme da potvrdi prijavu
 4. Korisnik pristupa uslugama za koje ima dopuštenje (što uključuje odlazak na određenu stranicu - ili za pacijente ili za zdravstvene djelatnike ili za administratore)
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime i/ili lozinka
 1. Korisnik dobiva obavijest o neuspješnoj prijavi i vraća ga na stranicu za prijavu

2. Korisnik mijenja krivo unešene podatke ili odustaje od prijave (vraća se na naslovnu stranu)

UC3 - Promjena korisničkih podataka

- **Glavni sudionici:** Pacijent, Zdravstveni djelatnik, Administrator
- **Cilj:** Promijeniti određene korisničke podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za promjenu podataka
 2. Korisnik unosi podatak ili podatke koje želi promijeniti (email ili korisničko ime ili lozinku)
 3. Korisnik pritišće dugme da potvrdi promjenu podataka
 4. Korisnik pristupa uslugama za koje ima dopuštenje ali s novim podacima pri čemu stari podatci budu obrisani
- **Opis mogućih odstupanja:**
 - 2.a Neispravan oblik email-a ili neispravan oblik lozinke
 1. Korisnik dobiva obavijest o neuspješnoj promijeni podataka te bude vraćen na stranicu za promjenu podataka
 2. Korisnik mijenja krivo unešene podatke ili odustaje od promjene podataka (vraća se na određenu stranicu)

UC4 - Upravljanje terminima

Uključuje obrasce uporabe UC4.1 Zakazivanje termina, UC4.2 Otkazivanje termina i UC4.3 Pregled termina.

UC4.1 - Zakazivanje termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Mogućnost odabira termina za odlazak na fizikalnu terapiju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao pacijent
- **Opis osnovnog tijeka:**
 1. Pacijent odabire opciju za zakazivanje termina
 2. Pacijent na prikazanom kalendaru izabire poželjni slobodni termin
 3. Pacijent navodi vrstu i opis oboljenja
 4. Pacijent pritišće dugme za potvrdu odabranog termina

5. Pacijent dobiva potvrdu o uspješno poslanom zahjevu za termin te bude vraćen na stranicu za pacijente
- **Opis mogućih odstupanja:**
 - 2.a Pacijent je unio termin koji ima identično vrijeme održavanja kao i neki već prethodno potvrđeni njegov termin
 1. Pacijent dobiva obavijest da unese drugi termin
 - 2.b Pacijent je unio termin u roku od 10 sekundi od prethodno poslanog termina
 1. Pacijent dobiva obavijest da pričeka 10 sekundi
 - 3.a Pacijent nije unio opis oboljenja a kliknuo je na potvrdu termina
 1. Pacijent dobiva obavijest da unese opis oboljenja

UC4.2 - Otkazivanje termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Mogućnost otkazivanja termina
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao pacijent i ima zakazan određen termin
- **Opis osnovnog tijeka:**
 1. Pacijent na popisu svojih termina odabire zakazani termin koji želi otkazati
 2. Pacijent odabire opciju za otkazivanje odabranog termina
 3. Pacijent dobije obavijest o uspješno otkazanom terminu
- **Opis mogućih odstupanja:**
 - 2.a Pacijent prekasno pokušava otkazati termin (unutar 24 sata kada je zakazani termin)
 1. Sustav onemogućuje pacijentu otkazivanje tog termina te pacijent dobiva obavijest o nemogućnosti otkazivanja termina

UC4.3 - Pregled termina

- **Glavni sudionik:** Pacijent
- **Cilj:** Mogućnost prikaza zakazanih termina
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao pacijent
- **Opis osnovnog tijeka:**
 1. Pacijent na kalendaru može vidjeti svoje zakazane termine

UC5 - E-mail podsjetnik za termin

- **Glavni sudionik:** -
- **Cilj:** Slanje e-mail podsjetnika za zakazani termin (24 sata prije termina)
- **Sudionici:** Baza podataka, pacijent
- **Preduvjet:** Korisnik prijavljen kao pacijent i postoji zakazani termin
- **Opis osnovnog tijeka:**
 1. Pacijent dobiva e-mail obavijest o zakazanom terminu (koji je potvrđen od strane određenog zdravstvenog djelatnika)

UC6: Pregled svih neprihvaćenih termina u djelatnikovoj smjeni

- **Glavni sudionik:** Zdravstveni djelatnik
- **Cilj:** Omogućiti pregled svih neprihvaćenih termina, ali isključivo u djelatnikovoj smjeni
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik
- **Opis osnovnog tijeka:**
 1. Zdravstvenom djelatniku se na njegovoj stranici pokazuje tablica svih neprihvaćenih termina koje on ima pravo prihvatiti
 2. Zdravstveni djelatnik pritiskom na dostupno dugme šalje taj termin u daljnju konfiguraciju

UC7 - Pregled svojih rezerviranih termina

Uključuje obrasce uporabe UC7.1 Odabir opreme, UC7.2 Pomicanje svojih rezerviranih termina, UC7.3 Otkazivanje rezerviranih termina i UC7.4 Brisanje rezerviranih termina

UC7.1 - Odabir opreme

- **Glavni sudionik:** Zdravstveni djelatnik
- **Cilj:** Omogućiti zdravstvenom djelatniku da odabere opremu koja je pogodna za korištenje s obzirom na opis zdravstvenih problema pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik
- **Opis osnovnog tijeka:**
 1. Zdravstveni djelatnik odabire opremu kojom će liječiti pacijenta
 2. Oprema automatski bude povezana sa sobom
 3. Zdravstveni djelatnik pritiskom na dugme potvrđuje korištenje opreme u određenoj sobi za tretman

- **Opis mogućih odstupanja:**

- 1.a Svaka instanca te opreme je u to vrijeme zauzeta

- 1. Zdravstveni djelatnik će morati ili premjestiti termin ili pokušati liječiti pacijenta nekom drugom opremom

UC7.2 - Pomicanje svojih rezerviranih termina

- **Glavni sudionik:** Zdravstveni djelatnik

- **Cilj:** Omogućiti zdravstvenom djelatniku da pomakne rezervirane termine zbog novonastale okolnosti te čeka potvrdu ili odbijanje od strane pacijenta (to ne smije činiti u istom danu kada je taj termin zakazan, mora biti na vrijeme organiziran)

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik

- **Opis osnovnog tijeka:**

- 1. Zdravstveni djelatnik na kalendaru odabire zakazani termin koje želi pomaknuti

- 2. Zdravstveni djelatnik bira novo vrijeme kada želi održati taj termin

- 3. Zdravstveni djelatnik pritiskom na dugme potvrđuje novonastale promjene

- **Opis mogućih odstupanja:**

- 2.a Pacijent odbija novi predloženi termin

- 1. Zdravstveni djelatnik ponavlja postupak pomicanja svojih rezerviranih termina (dok se pacijent ne složi sa prijedlogom)

- 2.a Zdravstveni djelatnik pokušava pomaknuti zakazani termin u vremenu unutar 24 sata kada je zakazani termina

- 1. Sustav onemogućuje pomicanje termina i šalje zdravstvenom djelatniku odgovarajuću poruku (zdravstveni djelatnik mora odraditi zakazani termin zbog kasne reakcije)

UC7.3 - Otkazivanje rezerviranih termina

- **Glavni sudionik:** Zdravstveni djelatnik

- **Cilj:** Omogućiti zdravstvenom djelatniku otkazivanje rezerviranih termina

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik

- **Opis osnovnog tijeka:**

1. Zdravstveni djelatnik na popisu svojih termina odabire zakazani termin koji želi otkazati
 2. Zdravstveni djelatnik odabire opciju za otkazivanje odabranog termina
 3. Zdravstveni djelatnik dobije obavijest o uspješno otkazanom terminu
- **Opis mogućih odstupanja:**
 - 2.a Zdravstveni djelatnik prekasno pokušava otkazati termin (unutar 24 sata kada je zakazani termin)
 1. Sustav onemogućuje zdravstvenom djelatniku otkazivanje tog termina (prisiljen je odraditi taj termin) te dobije obavijest o prekasno pokušaju otkazivanja termina

UC7.4 - Brisanje rezerviranih termina

- **Glavni sudionik:** Zdravstveni djelatnik
- **Cilj:** Omogućiti zdravstvenom djelatniku brisanje rezerviranih termina
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik
- **Opis osnovnog tijeka:**
 1. Zdravstveni djelatnik na popisu svojih termina odabire zakazani termin koji želi obrisati (smatra da taj tretman nema smisla izvršiti niti on niti bilo koji drugi zdravstveni djelatnik)
 2. Zdravstveni djelatnik odabire opciju za brisanje odabranog termina
 3. Zdravstveni djelatnik dobije obavijest o uspješno obrisanoj terminu

UC8 - Pregled termina pacijenata

- **Glavni sudionik:** Zdravstveni djelatnik
- **Cilj:** Omogućiti zdravstvenom djelatniku da pregleda termina pacijenata
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik
- **Opis osnovnog tijeka:**
 1. Zdravstveni djelatnik upisuje ime, prezime i OIB pacijenta
 2. Zdravstveni djelatnik ima uvid u sve termine tog pacijenta (čak i one termine koji nisu pod njegovoj nadležnosti)

UC9 - Proizvoljno slanje emaila

- **Glavni sudionik:** Zdravstveni djelatnik

- **Cilj:** Omogućiti zdravstvenom djelatniku da komunicira s pacijentom preko elektroničke pošte
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao zdravstveni djelatnik
- **Opis osnovnog tijeka:**
 1. Zdravstveni djelatnik odabire opciju slanja emaila
 2. Zdravstveni djelatnik upisuje email adresu određenog pacijenta
 3. Zdravstveni djelatnik upisuje naslov i glavni dio teksta

UC10 - Upravljanje korisnicima

Uključuje obrasce uporabe UC10.1. Pregled korisnika, UC10.2. Dodavanje korisnika, UC10.3. Brisanje korisnika.

UC10.1. - Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati registrirane korisnike (pacijente i zdravstvene djelatnike)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Korisnik koji je dobio ulogu administratora odabire opciju pregledavanja korisnika
 2. Administratoru se prikazuje lista svih registriranih korisnika sa njihovim osobnim podacima (posebno odvojeni pacijenti od zdravstvenih djelatnika)

UC10.2. - Dodavanje zdravstvenih djelatnika

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru stvaranje posebno privilegiranog aktora naše aplikacije - zdravstvenog djelatnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Korisnik koji je dobio ulogu administratora odabire opciju za dodavanje zdravstvenog djelatnika
 2. Administrator u određenim poljima upisuje podatke o novom zdravstvenom djelatniku

3. Administrator pritiskom na dugme sprema novounesene promjene
- **Opis mogućih odstupanja:**
 - 2.a Administrator upisuje zabranjeni format e-maila, već zauzeti e-mail ili općenito unosi podatke u nedopuštenom formatu
 1. Sustav upozorava administratora o neuspjelom unosu i vraća ga na stranicu za upisivanje novih zdravstvenih djelatnika
 2. Administrator ponovno unosi podatke o zdravstvenom djelatniku (ovaj put u ispravnom formatu) ili odustaje od tog čina

UC10.3. - Brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru brisanje korisnika (pacijenta ili zdravstvenog djelatnika)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora i u bazi podataka imamo barem jednog korisnika
- **Opis osnovnog tijeka:**
 1. Korisnik koji je dobio ulogu administratora odabire opciju pregledavanja korisnika
 2. Administrator za određenog korisnika odabire opciju brisanja tog korisnika
 3. Administrator pritiskom na dugme sprema novounesene promjene
 4. Popis korisnika se administratoru osvježava

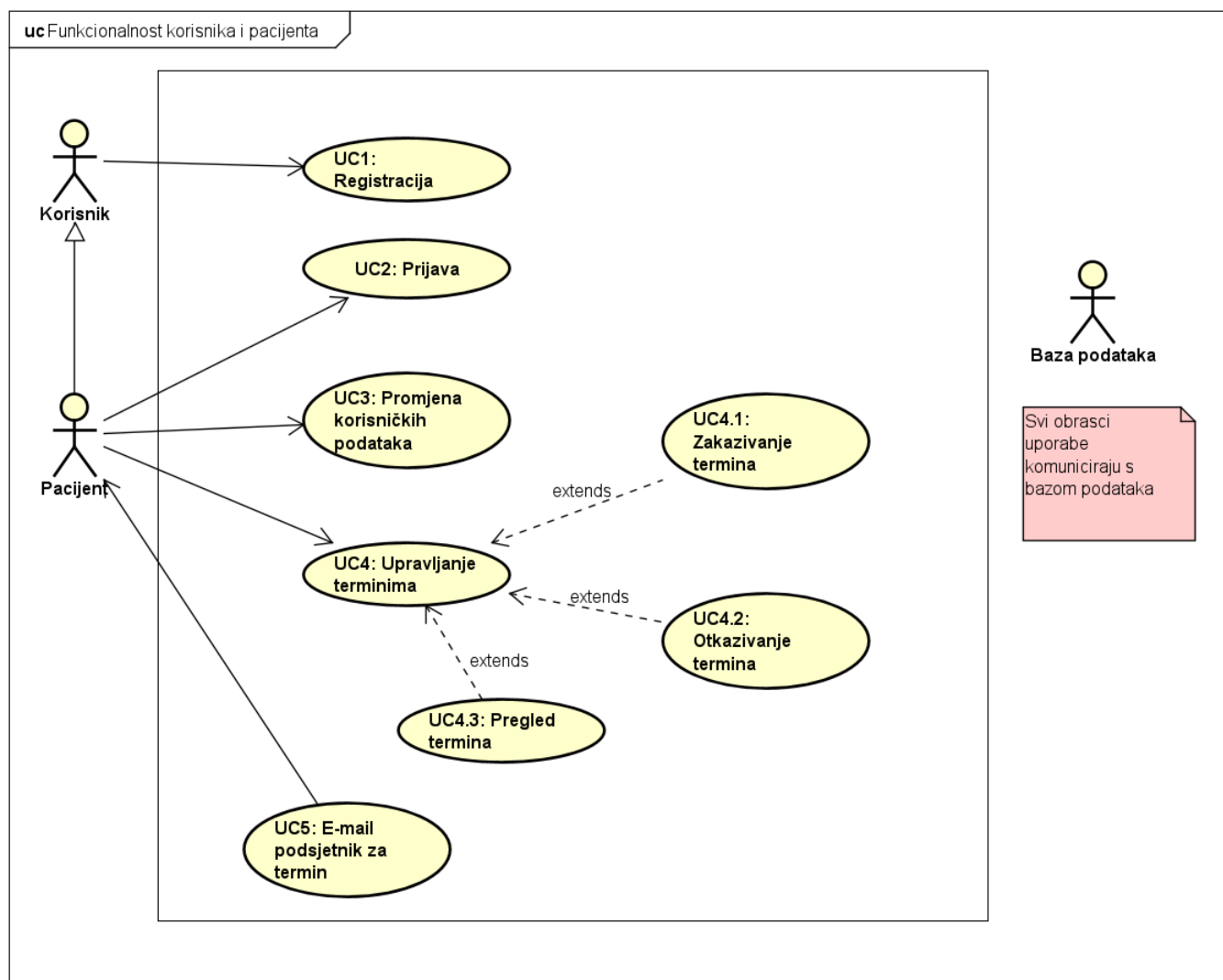
UC11 - Verifikacija podataka

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru verificiranje unesenih podataka od strane pacijenata
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora i u bazi podataka se određeni korisnik prijavio kao pacijent
- **Opis osnovnog tijeka:**
 1. Administrator dobiva obavijest o registraciji novog korisnika
 2. Administrator verificira podatke iz središnjeg informacijskog sustava zdravstvene zaštite

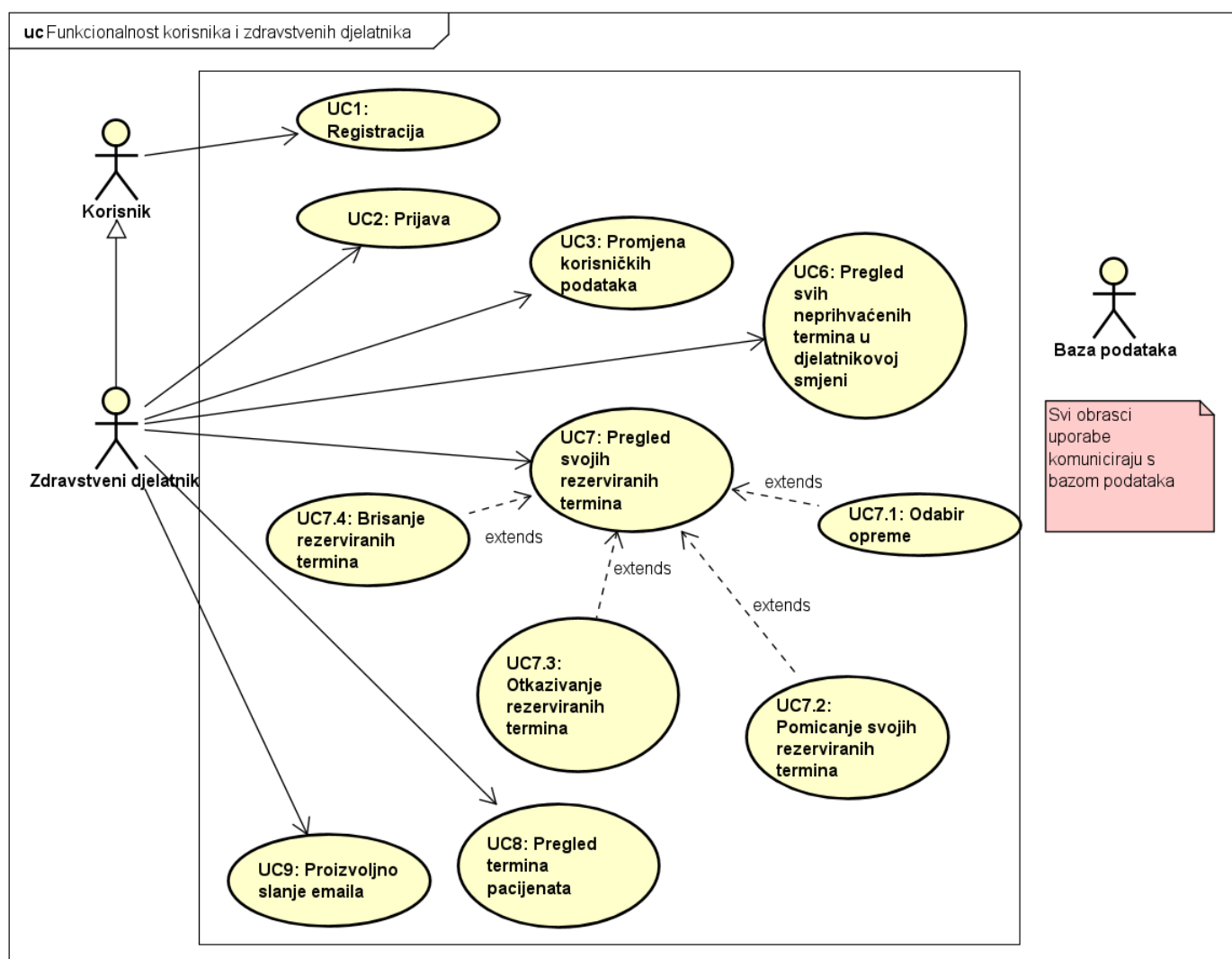
UC12 - Pregled statistike

- **Glavni sudionik:** Administrator
- **Cilj:** Omogućiti administratoru pregledati statistiku (npr. koji je postotak u sustavu pacijenata, a koji zdravstvenih djelatnika)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Statistika"
 2. Administrator se prikazuju razni statistički podaci (postotak pacijenata u sustavu, postotak zdravstvenih djelatnika u sustavu, postotak pacijenata kojemu treba određena oprema/usluga...)

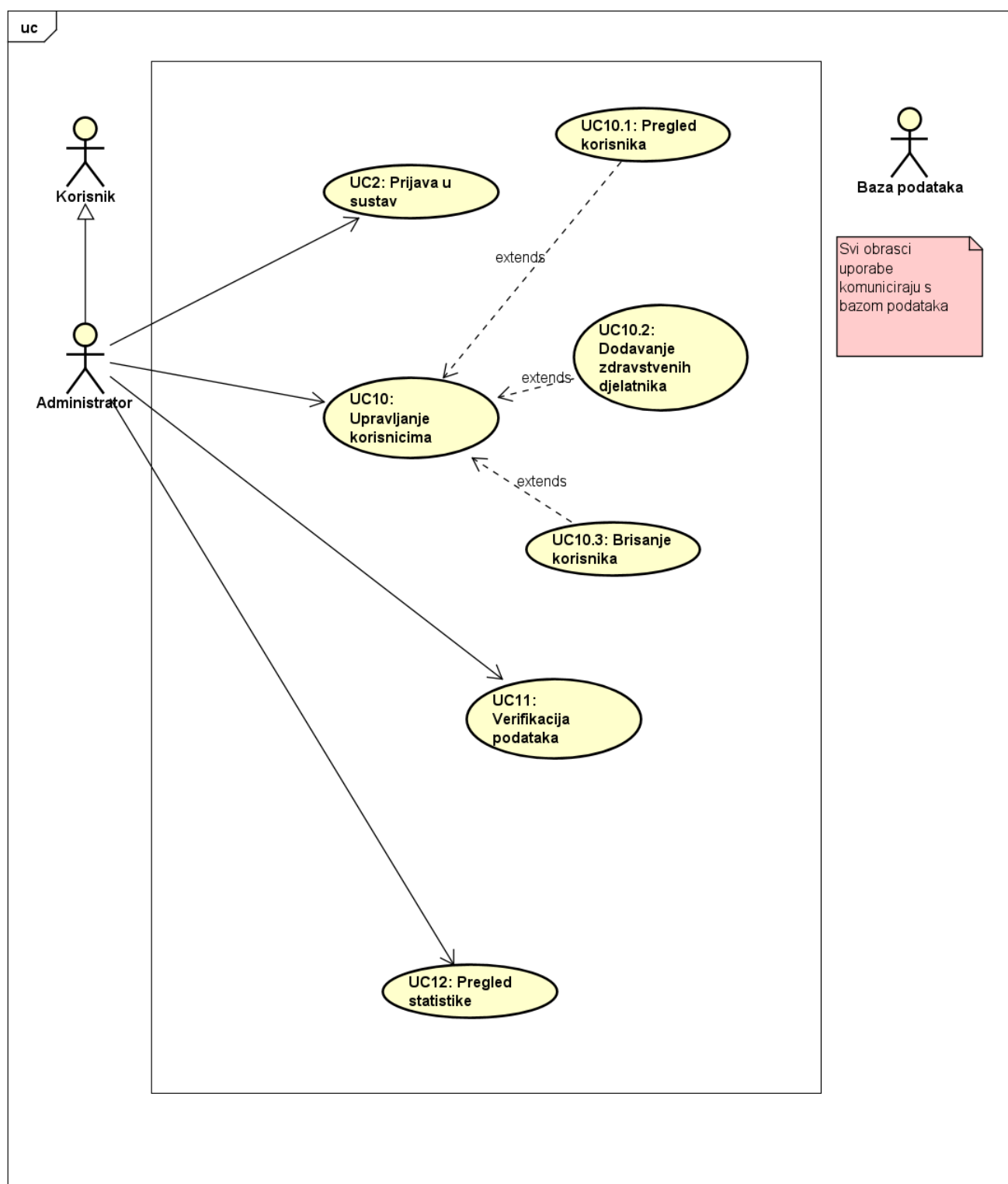
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe koji prikazuje funkcionalnost korisnika i pacijenta



Slika 3.2: Dijagram obrasca uporabe koji prikazuje funkcionalnost zdravstvenih djelatnika



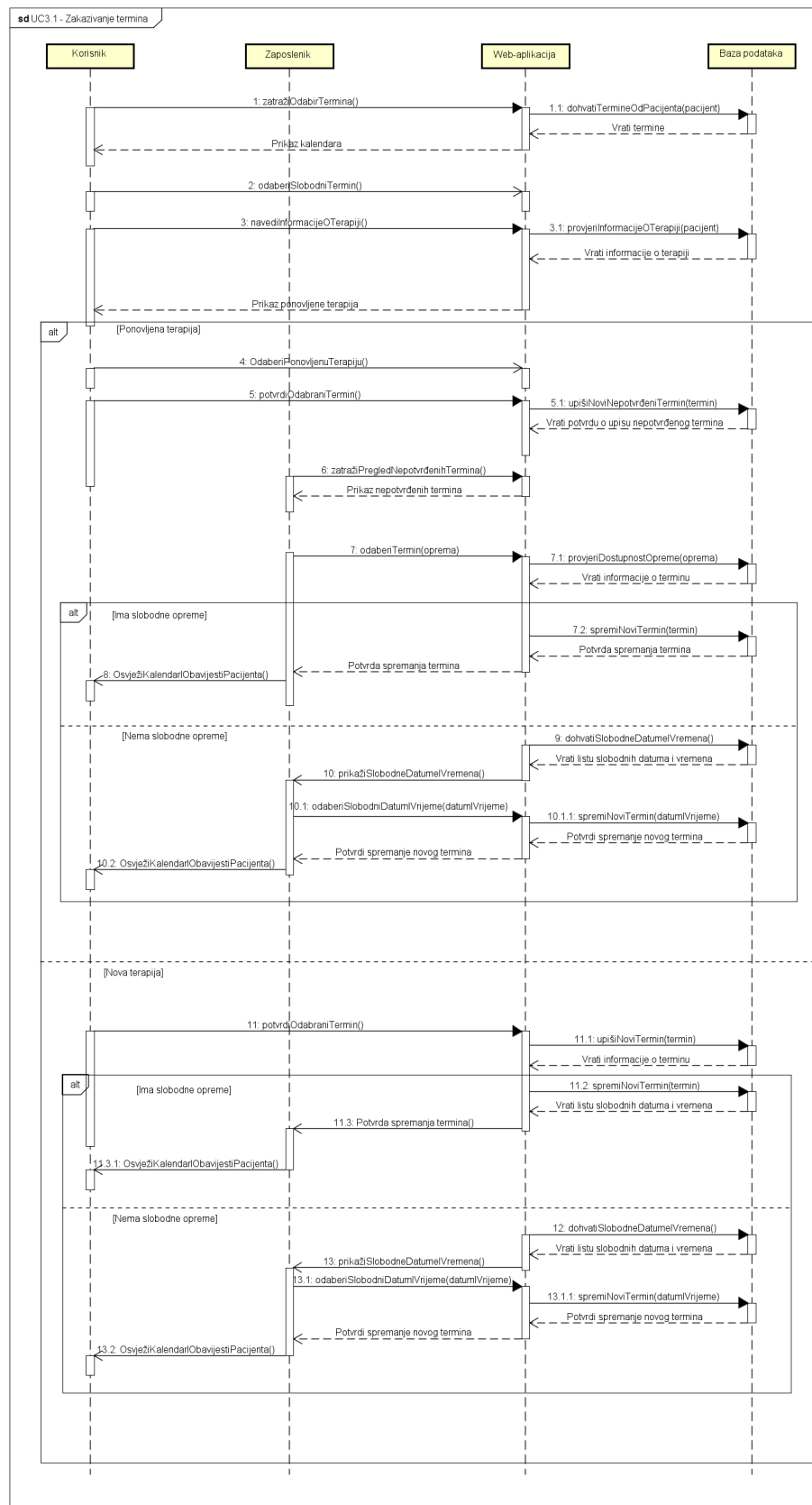
Slika 3.3: Dijagram obrasca uporabe koji prikazuje funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

UC3.1. - Zakazivanje termina

Korisnik, prijavljen kao pacijent, želi odabrati novi slobodni termin. Poslužitelj dohvaća trenutne termine te korisniku prikazuje kalendar. Pacijent navodi datum i vrijeme kada želi zakazati termin te od strane poslužitelja dobiva pregled svih vlastitih opisa zdravstvenih problema koje je unosio u prošlosti (ako ih ima). Pacijent odabire referencu na ponovljeni opis problema te potvrđuje odabrani termin. Termin se tada sprema u bazu podataka kao nepotvrđeni zahtjev za terminom. Djelatnik dohvaća sve zahtjeve koji odgovaraju njegovoj smjeni te odabire termin koji želi potvrditi. Djelatnik navodi opremu koja mu je potrebna za opisani zdravstveni problem te se vrši provjera dostupnosti opreme. Ako je oprema slobodna, termin se sprema u bazu podataka, djelatnik dobiva potvrdu o spremanju te se pacijentu šalje e-mail o uspješno zakazanom terminu. Ako nema slobodne opreme, djelatnik dobiva listu slobodnih termina u sljedećih tjedan dana te odabire neki od ponuđenih termina, nakon čega se termin sprema u bazu podataka, a djelatnik i pacijent dobivaju prikladnu obavijest.

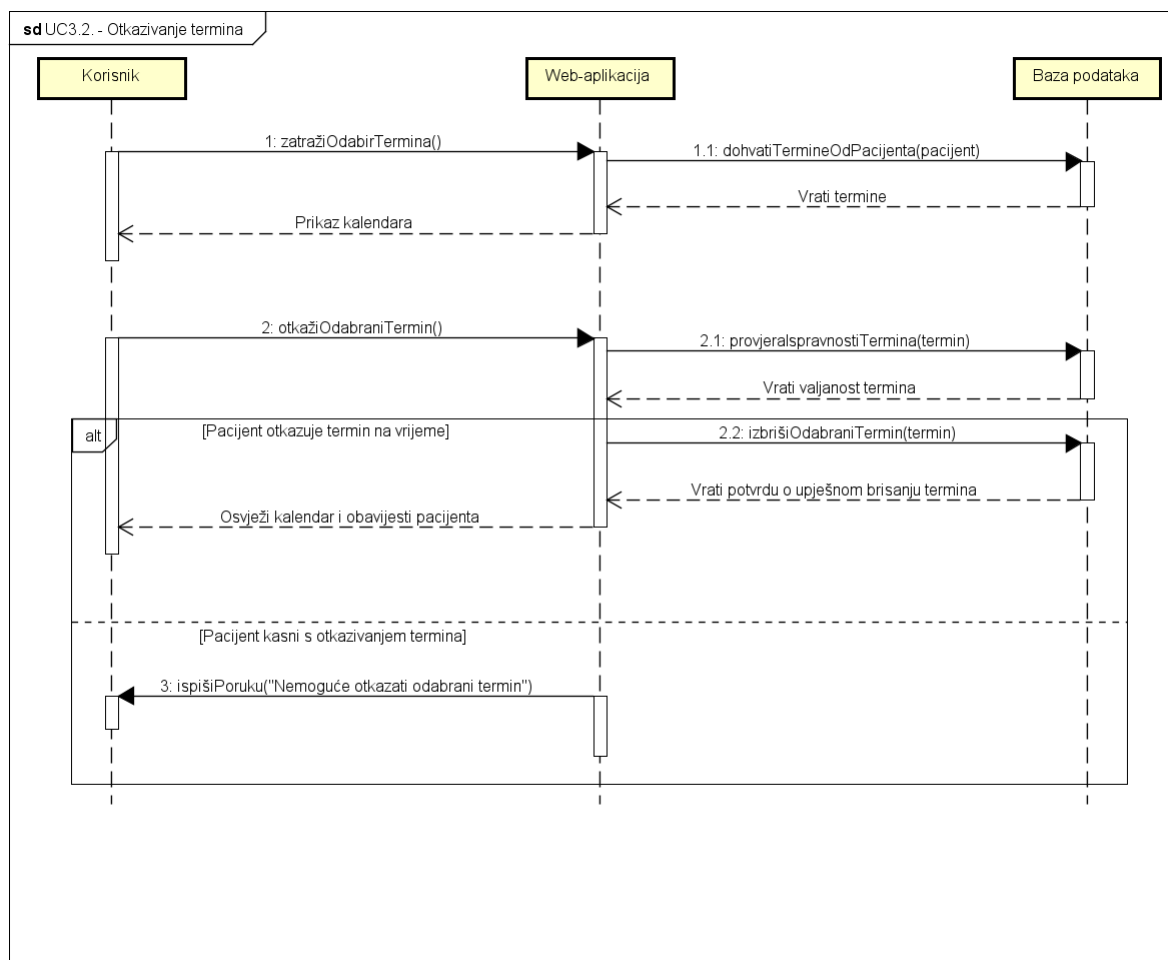
Ako je pacijent unio podatke o novom zdravstvenom problemu, poslužitelj te informacije sprema u bazu podataka. Pacijent potvrđuje odabrani termin te se on sprema kao nepotvrđeni zahtjev za terminom. Postupak je dalje jednak kao i u prijašnjem slučaju - Djelatnik dohvaća sve zahtjeve koji odgovaraju njegovoj smjeni te odabire termin koji želi potvrditi. Djelatnik navodi opremu koja mu je potrebna za opisani zdravstveni problem te se vrši provjera dostupnosti opreme. Ako je oprema slobodna, termin se sprema u bazu podataka, djelatnik dobiva potvrdu o spremanju te se pacijentu šalje e-mail o uspješno zakazanom terminu. Ako nema slobodne opreme, djelatnik dobiva listu slobodnih termina u sljedećih tjedan dana te odabire neki od ponuđenih termina, nakon čega se termin sprema u bazu podataka, a djelatnik i pacijent dobivaju prikladnu obavijest.



Slika 3.4: Sekvencijski dijagram za UC3.1.

UC3.2. - Otkazivanje termina

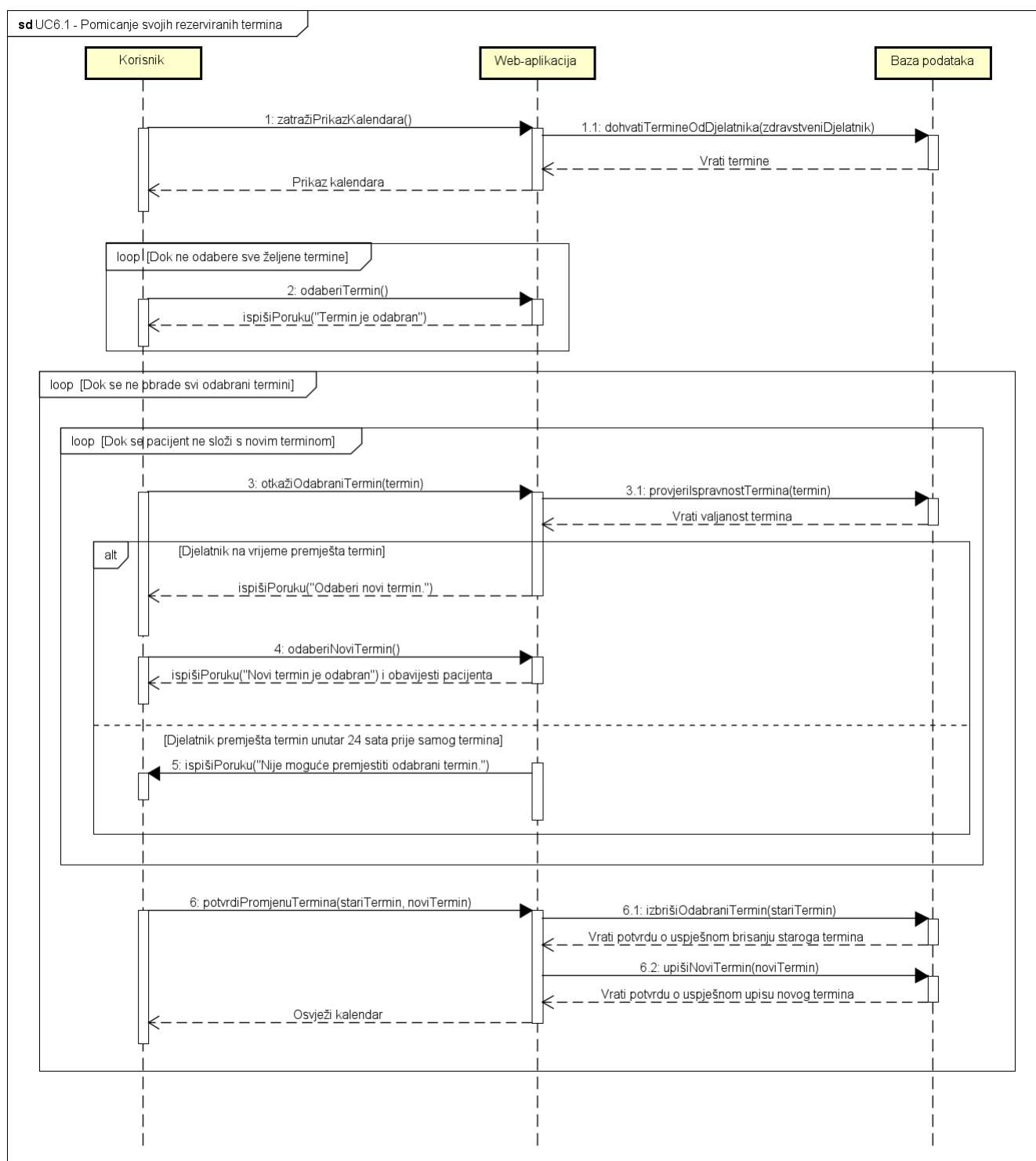
Korisnik, prijavljen kao pacijent, želi odabrati termin kako bi ga mogao otkazati. Poslužitelj dohvaća sve zakazane termine od tog pacijenta i prikazuje ih u kalendaru. Pacijent otkazuje odabrani termin. Poslužitelj ispituje ispravnost odabranog termina te ako je on ispravan, briše ga iz baze podataka, osvježava kalendar i šalje e-mail poruku pacijentu. Ako pacijent prekasno pokušava otkazati termin, odnosno ako to čini 24 sata prije zakazanog termina, poslužitelj ne briše termin i šalje odgovarajuću poruku pacijentu.



Slika 3.5: Sekvencijski dijagram za UC3.2.

UC6.1 - Pomicanje svojih rezerviranih termina

Korisnik, prijavljen kao zdravstveni djelatnik, želi premjestiti svoje rezervirane termine zbog nemogućnosti njihovog izvršavanja. Poslužitelj dohvaća sve rezervirane termine od zdravstvenog djelatnika te ih prikazuje na kalendaru. Zdravstveni djelatnik odabire sve termine koje bi želio premjestiti. Sljedeći koraci se izvršavaju sve dok se ne obrade svi odabrani termini. Dok se ne dobije potvrda pacijenta, zdravstveni djelatnik odabire opciju otkazivanja trenutno odabranog termina. Poslužitelj ispituje ispravnost odabranog termina te vraća njegovu valjanost. Ako je zdravstveni djelatnik na vrijeme odlučio pomaknuti odabrani termin, on to čini te se pacijentu šalje obavijest o mogućem novom terminu. Ako zdravstveni djelatnik pomiče odabrani termin unutar 24 sata od samoga odabranoga termina, poslužitelj ga u tome onemogućuje. Nakon što pacijent potvrdi suglasnost o novome terminu, zdravstveni djelatnik potvrđuje samu promjenu termina. Poslužitelj dobiva i stari i novi termin te se stari uklanja, a novi dodaje u bazu podataka. Nakon pomicanja termina, zdravstvenom se djelatniku osvježava kalendar.



Slika 3.6: Sekvencijski dijagram za UC6.1

3.2 Ostali zahtjevi

- Korisničko sučelje mora biti intuitivno i prilagođeno, olakšavajući korisnicima njenu upotrebu
- Sustav mora omogućiti istovremeni rad većem broju korisnika
- Sustav treba omogućiti brzu i učinkovitu prijavu pacijenata te raspoređivanje termina unutar radnog vremena
- Prilikom komunikacije s bazom podataka, svi osobni podaci moraju biti prikladno zaštićeni
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Aplikacija mora biti oblikovana poštujući načela objektno-orijentiranog programiranja
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba omogućiti obavješćavanje pacijenata putem elektroničke pošte o promjenama u rasporedu ili drugim važnim informacijama
- Sistemski administrator treba imati potpuni pristup svim podacima i mogućnost definiranja postavki sustava

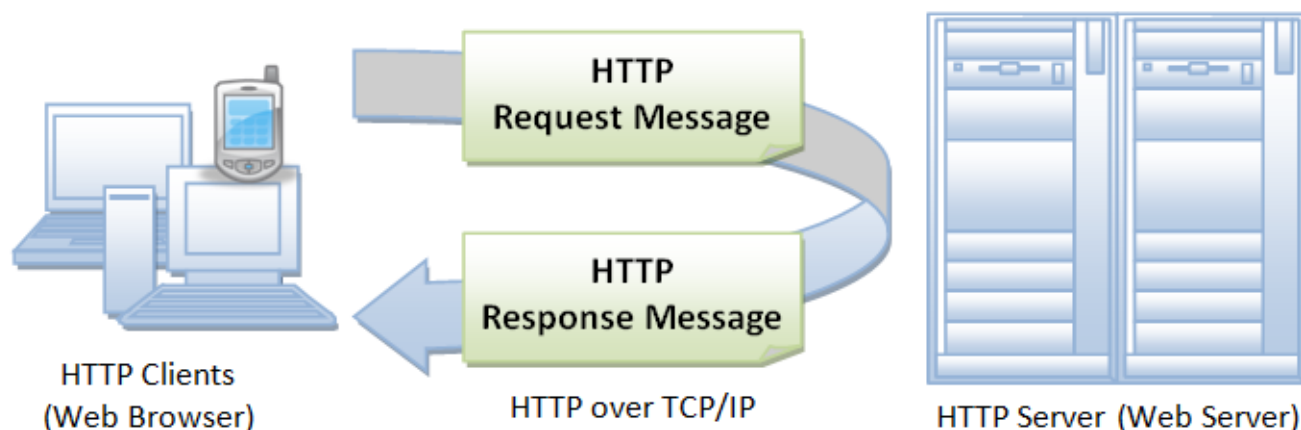
4. Arhitektura i dizajn sustava

Arhitekturu sustava općenito (pa tako i našeg) možemo podijeliti na tri dijela:

- *Web poslužitelj*
- *Web aplikacija*
- *Baza podataka*

Internetski preglednik (web preglednik, web browser) je dio arhitekture koji korisniku omogućuje pregled web-stranica pa tako i samoga sadržaja koji se nalazi na njoj. Korisnik putem web preglednika šalje HTTP zahtjev poslužitelju za dohvat željenog sadržaja i čeka HTTP odgovor. HTTP je protokol bez stanja što znači da primatelj ne smije zadržati stanje sesije iz prethodnih zahtjeva. Neki od popularnijih web preglednika su "Google Chrome" ili "Opera".

Web poslužitelj je osnova svake web aplikacije. On šalje klijentu HTTP odgovor na određeni HTTP zahtjev. Korisnik koristi web aplikaciju na način da web aplikacija obrađuje zahtjev te ovisno o potrebi pristupa bazi podataka. Poslužitelj vraća odgovor u obliku HTML dokumenta koji je vidljiv korisniku.



Slika 4.1: Prikaz HTTP komunikacije između klijenta i poslužitelja

Naša grupa je za projekt na predmetu Programsko inženjerstvo (ak.god. 2023./2024.) odabrala React sustav za potrebe frontenda. Odabrali smo React zbog njegove jednostavne i moćne arhitekture koja omogućuje brz i održiv razvoj web aplikacija. React se temelji na principu komponenata, što znači da aplikaciju gradimo kao skup neovisnih dijelova sučelja, svaki s vlastitom logikom i stilovima. Time svaka komponenta postaje ponovno upotrebljiva i lako zamjenjiva što olakšava razvoj aplikacije.

Za backend smo odabrali raditi u programskom jeziku Java i to u Spring Boot-u. Koristili smo Maven alat za izgradnju programskog koda. Spring Boot nam se svidio zbog dvije karakteristike: inverzija kontrole gdje sam Spring Boot kontrolira izvršavanje programskog koda te injektiranje objekata o kojima ovisi rad koda. Neke od karakteristika Spring Boot-a su: unaprijed pripremljene funkcionalnosti, nema generiranja klasa i koda već se koriste unaprijed definirane biblioteke. U svakom slučaju Spring Boot zna dosta olakšati programeru posao. Također olakšava i spajanje na bazu podataka o kojoj će biti više riječi u sljedećoj cjelini.

Razvojno okruženje koje smo koristili zavisilo je od člana do člana ekipe, neki su radili u Eclipse-u, neki u IntelliJ. Koristili smo GitHub sustav za upravljanje verzijama programske potpore te TeXstudio za pisanje dokumentacije.



Slika 4.2: Neke od koristi Spring Boot-a

4.1 Baza podataka

Za bazu podataka odabrali smo SQL vrstu baze podataka, točnije MySQL software zbog svoje jednostavnosti i kvalitetne usluge. MySQL je besplatni open-source sistem za upravljanje bazama podataka. Također ga podržava velika i aktivna zajednica korisnika što nam omogućava brz pronalazak potrebnih resursa, tutorijala i rješenja za moguće probleme. Što se performanse tiče, MySQL je poznat po brzini i efikasnosti te se koristi kako za manje web aplikacije tako i za velike poslovne sisteme. Lako se integrira u različite programske jezike i razvojne okoline.



Slika 4.3: MySQL

4.1.1 Opis tablica

Glavni cilj: Baza podataka osigurava pregled i pohranu podataka o svim faktorima procesa zauzimanja termina u medicinsko rehabilitacijskoj klinici. Kako bi se osiguralo pravilno rukovanje bazom podataka potrebne su pojedine tablice tj. entiteti od kojih svaki sadrži svoje atribute te odnose među pojedinim entitetima koji se u struci nazivaju „relationships“. Entiteti: Našu bazu podataka tvori nekoliko entiteta, preciznije: User (korisnik), Appointment (dogovoreni sastanak), Equipment (oprema) te Room (prostorija/radna sala).

Entitet User: To je najbitniji entitet jer sadrži podatke o svim djelatnicima

medicinsko rehabilitacijske klinike, pacijentima i ostalim djelatnicima. Za ostvarivanje funkcionalnosti potrebni su sljedeći atributi: idUser(primary key), imeUser, prezime, oib, username, password, email, datum rođenja, spol, start date(unosi se pri stvaranju tj. kad je napravljen korisnički račun) te na kraju enum koji specijalizira entitet Usera na jedno od dvije opcije. To su Patient ili Employee. Employee-u se dalje bilježi status, tj. gleda li se na njega kao aktivnog zaposlenika, neaktivnog ili admin.

Entitet Appointment: Entitet Appointment prikazuje sve dogovorene sastanke te omogućava uspješno dogovaranje novih bez da dođe do nekih problema kao što su kolizije ili nedostatak opreme. Atributi u entitetu Appointment su: idApp(primary key), vrijeme termina (od kad do kad je zakazan termin), user, opis termina, oprema i prostorija.

Entitet Equipment: Entitet Equipment sadrži podatke o svojoj opremi. Osigurava da ne dođe do problema rezerviranja više opreme nego što je dostupno, na primjer pacijent A je rezervirao zadnje štake te pacijent B u istom terminu nema pristup tom komadu opreme. Atributi koji čine entitet Equipment su: idEqu(primary key), ime, opis i status(radi li ili ne).

Entitet Prostorija: Entitet Prostorija prikazuje podatke o mogućim prostorijama za rehabilitaciju. Osigurava da određena prostorija ne može biti zauzeta više puta odjednom. Atributi koji sačinjavaju entitet Prostorija su: idPro(primary key), ime i kapacitet.

Veze (relationships) među entitetima: Veze među entitetima osiguravanju pravilnu i predviđenu funkcionalnost cijelog sustava (web aplikacije), povezuju sve entitete na način koji tvori logiku povezanosti. Uspostavljene su tri veze u našoj bazi podataka: User Appointment. Veza User - Appointment je one to many(jedan na više, 1 - N) tip veze. Jedan User može zatražiti više sastanaka. Appointment - Prostorija. Veza Appointment Prostorija je one to many(jedan na više, 1 - N) tip veze. Više sastanaka se može održavati u isto vrijeme u istoj prostoriji. Prostorija - Equipment. Veza Prostorija Equipment je one to many(jedan na više, 1 - N) tip veze. Više opreme se može držati u jednoj prostoriji.

korisnik - User tablica		
idUser	INT AUTO_INCREMENT	Jedinstveni identifikator korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

korisnik - User tablica		
imeUser	VARCHAR(255)	Ime korisnika
prezime	VARCHAR(255)	Prezime korisnika
oib	INT	OIB korisnika
username	VARCHAR(255)	Korisničko ime
password	VARCHAR(255)	Lozinka
datumRodnja	DATE	Datum rođenja
spol	VARCHAR(1)	Spol (M ili F)
startDate	DATE	Datum početka korisničkog računa
email	VARCHAR(255)	E-mail adresa

prostorija - Prostorija tablica		
idPro	INT AUTO_INCREMENT	Jedinstveni identifikator prostorije
imePro	VARCHAR(255)	Ime prostorije
kapacitet	INT	Kapacitet prostorije

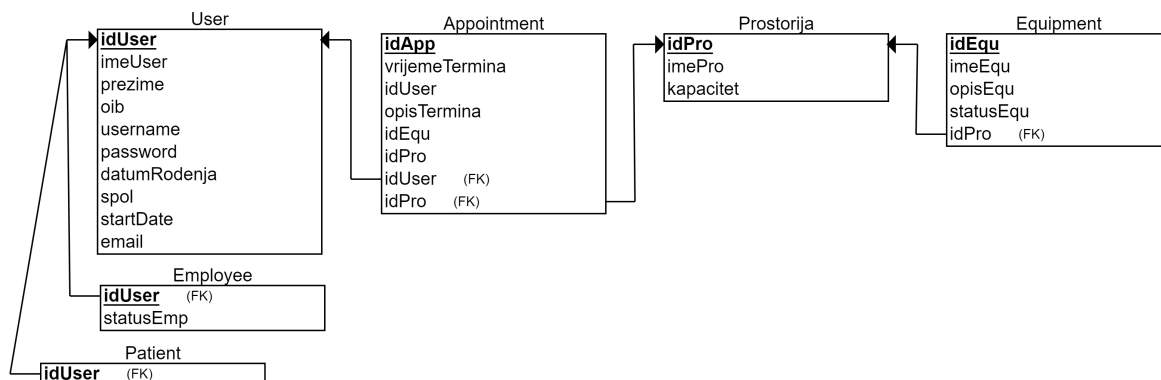
patient - Patient tablica		
idUser	INT AUTO_INCREMENT	Jedinstveni identifikator pacijenta

employee - Employee tablica		
statusEmp	VARCHAR(255)	Status zaposlenika
idUser	INT AUTO_INCREMENT	Jedinstveni identifikator zaposlenika

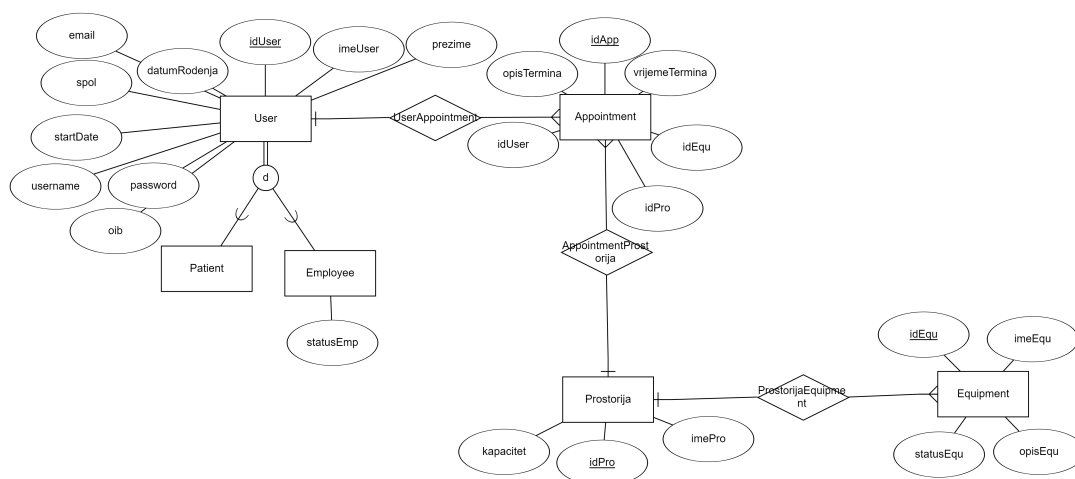
appointment - Appointment tablica		
idApp	INT AUTO_INCREMENT	Jedinstveni identifikator termina
vrijemeTermina	DATETIME	Vrijeme termina
idUser	INT	Jedinstveni identifikator korisnika
opisTermina	VARCHAR(255)	Opis termina
idEqu	INT	Jedinstveni identifikator opreme
idPro	INT	Jedinstveni identifikator prostorije

equipment - Equipment tablica		
idEqu	INT AUTO_INCREMENT	Jedinstveni identifikator opreme
imeEqu	VARCHAR(255)	Ime opreme
opisEqu	VARCHAR(255)	Opis opreme
statusEqu	VARCHAR(255)	Status opreme
idPro	INT	Jedinstveni identifikator prostorije

4.1.2 Dijagram baze podataka



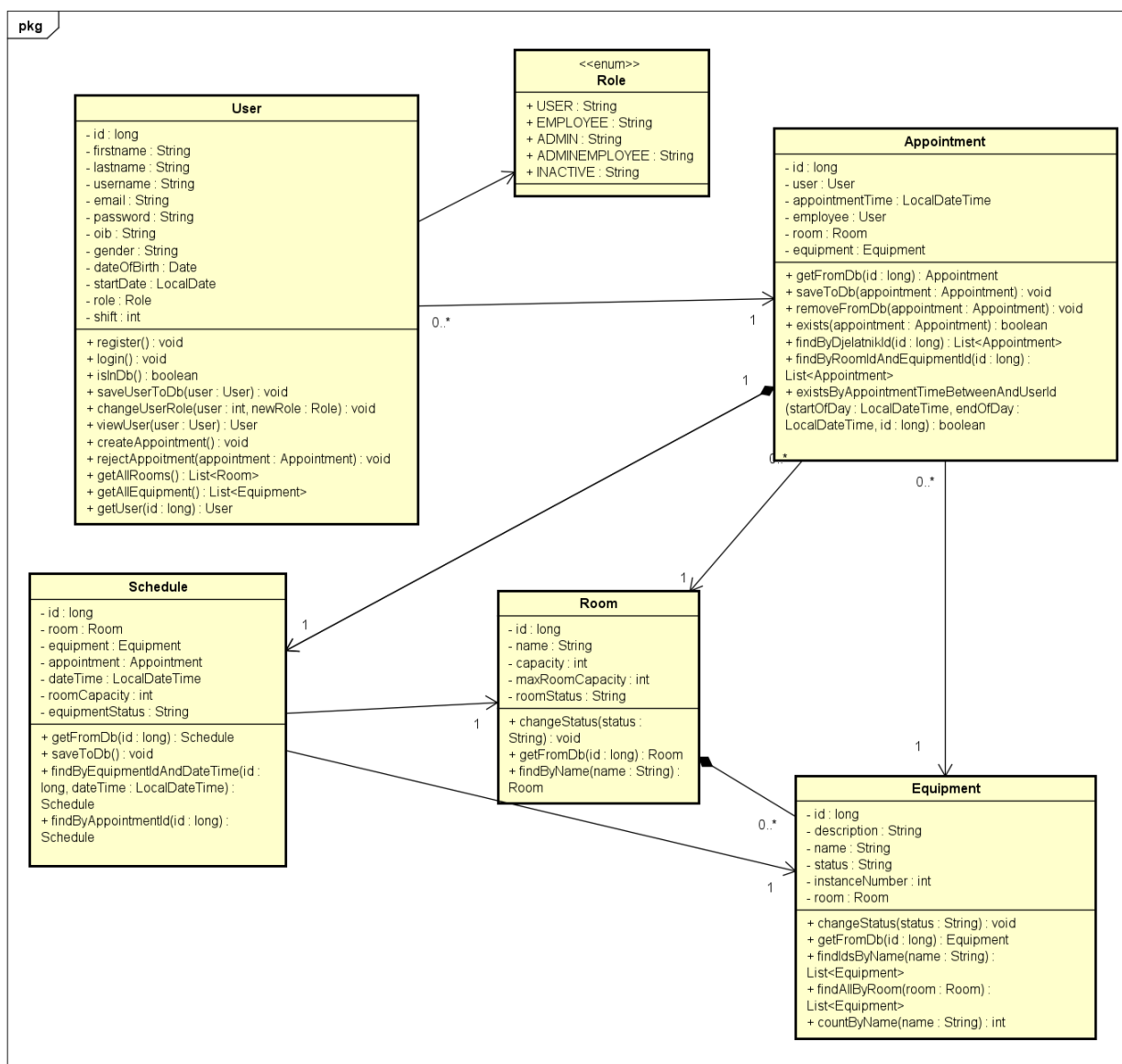
Slika 4.4: Relacijski model baze podataka



Slika 4.5: ER model baze podataka

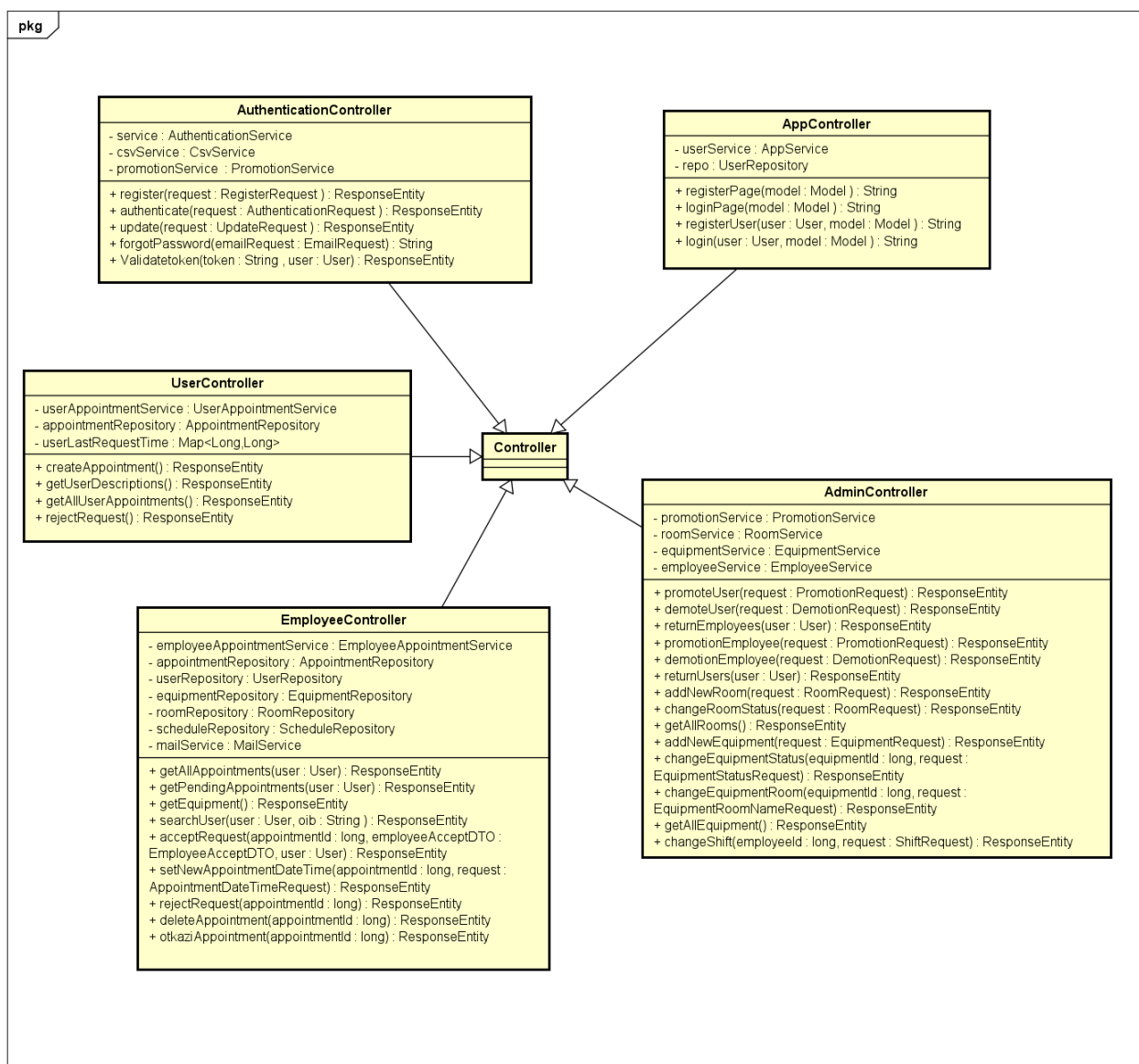
4.2 Dijagram razreda

Na slici 4.6 prikazani su razredi kojima se predstavlja struktura baze podataka u našoj aplikaciji te su razredi User, Appointment, Room, Schedule i Equipment entiteti u bazi podataka. Implementirane metode služe za komunikaciju s bazom podataka te po potrebi izvršavaju pisanje, brisanje ili dohvaćanje podataka iz same baze. Razred User predstavlja svakog korisnika aplikacije. Kako bi korisnik mogao upotrebljavati pogodnosti koje aplikacija nudi, on se najprije mora registrirati u sustav koristeći osobne podatke: ime, prezime, korisničko ime, lozinka, email, OIB, spol te datum rođenja. Razred User također koristi i enumeraciju Role koja predstavlja ulogu korisnika u sustavu. Razred User također može zakazati i otkazati termin terapije, koju predstavlja razred Appointment. Razred Appointment sadrži podatke o samome terminu terapije te ga samo korisnici s pripadajućom ulogom EMPLOYEE, ADMINEMPLOYEE mogu otkazati ili stvoriti. Entitet Schedule koristi se prilikom stvaranja, odnosno potvrde samoga termina. Svrha je zauzimanje termina po datumima i satima s pripadajućom prostorijom za pregled te opremom za pregled, što omogućava provjeru raspoloživosti određenog datuma, vremena, sobe te opreme. Entitet Room koristi se za spremanje podataka o svakoj prostoriji ustanove te njih u sustav dodaje korisnik s ulogom ADMIN. ADMIN također jedini barata funkcijama vezanim za uređivanje atributa prostorija. Entitet Equipment predstavlja opremu koja se koristi prilikom medicinske rehabilitacije. Oprema je vezana uz određenu prostoriju te ju također zadaje ADMIN.



Slika 4.6: Dijagram razreda - Modeli

Na slici 4.7 prikazani su razredi kojima se povezuje Backend i Frontend dio same aplikacije. Razred `AuthenticationController` bavi se adekvatnom provjerom informacija koje korisnik unosi te omogućava njegovu registraciju, autentikaciju, izmjenu osobnih podataka poput lozinke, korisničkog imena ili emaila ako ih je zaboravio. Razred `AdminController` omogućava Adminu da dodaje nove zaposlenike, uklanja ljude s kojima više nema radnog odnosa, dobiva popis svih zaposlenika, dodaje i dohvaća prostorija, opreme te izmjena njihovih parametara. Razred `AppController` služi za povezivanje s frontend dijelom aplikacije i prikazivanje odabranih stranica, npr. stranica za registraciju, stranica za prijavu te provjerava valjanost podataka prilikom prijave i registracije. `UserController` se bavi korisnikom koji ima ulogu `USER` te omogućava dohvat svih korisnikovih zakazanih termina, dohvat prijašnjih opisa fizikalnih problema, te slanje i poništavanje zahtjeva za terminom. `EmployeeController` koriste korisnici s ulogama `EMPLOYEE` i `ADMINEMPLOYEE` te on služi za obradu zahtjeva za terminom - njihovo prihvatanje, odbijanje, brisanje i izmjena određenih parametara.

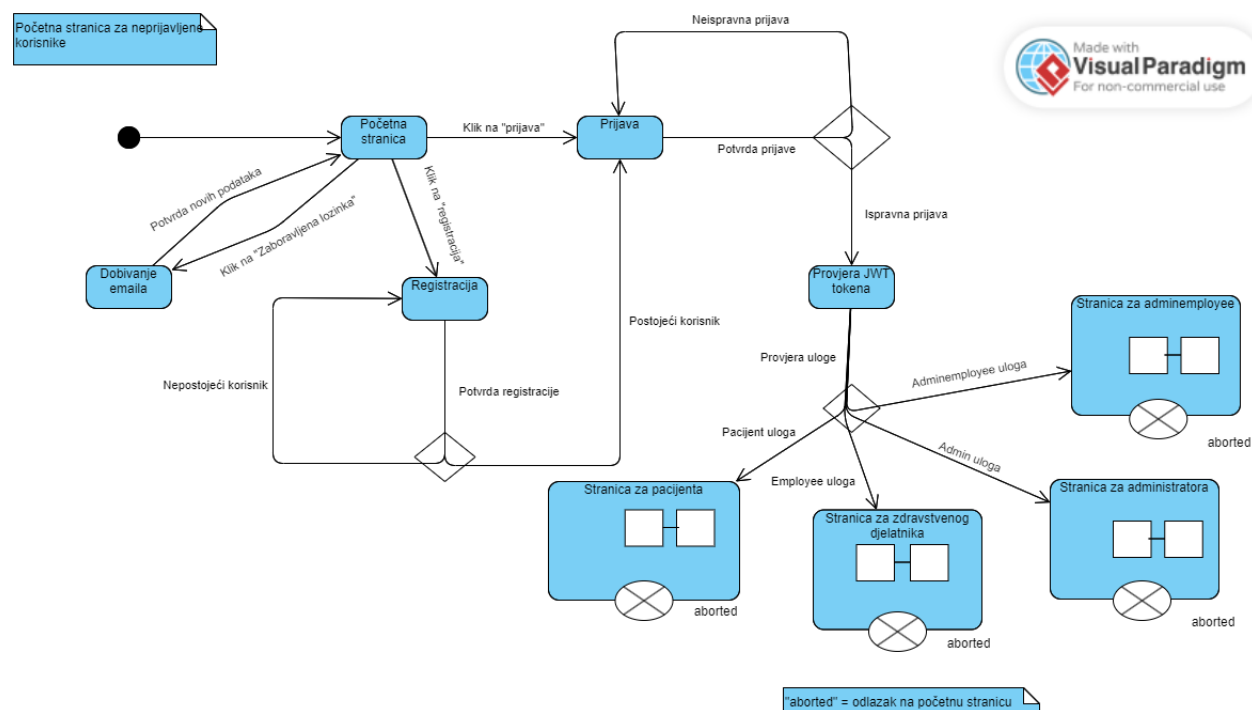


Slika 4.7: Dijagram razreda - Kontroleri

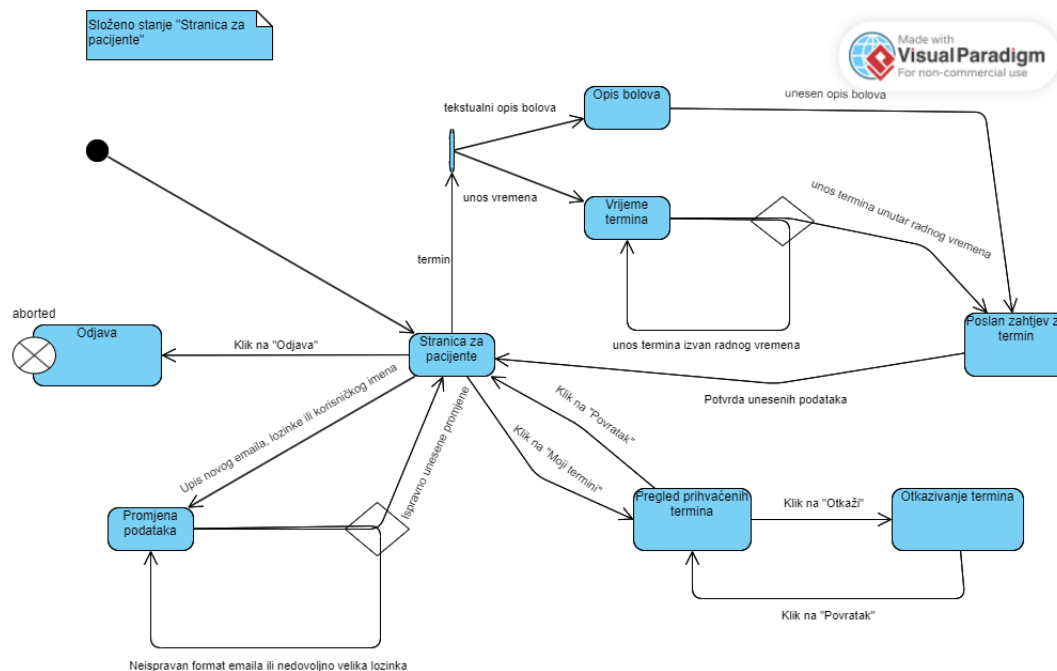
dio 2. revizije

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

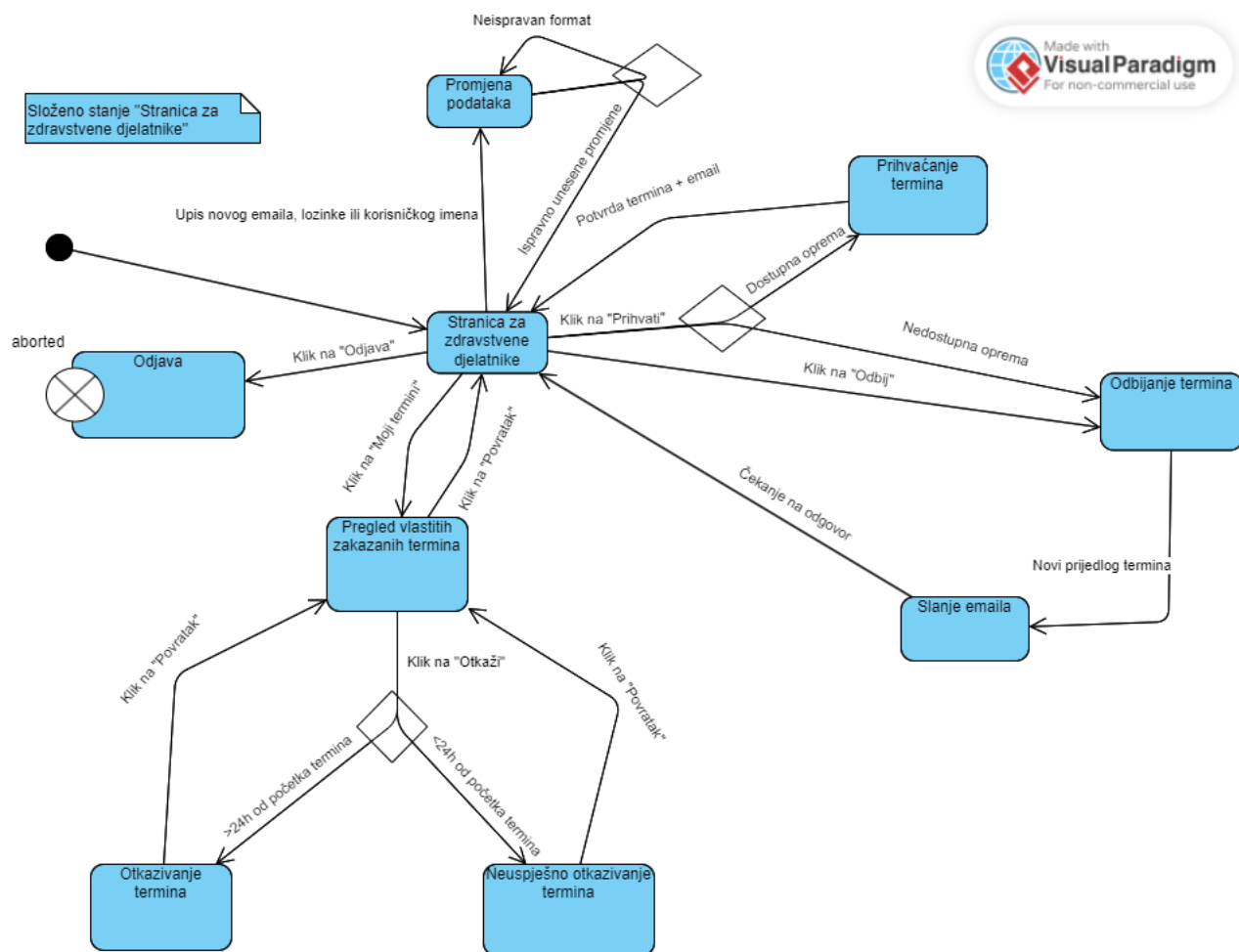
4.3 Dijagram stanja



Slika 4.8: Dijagram stanja početne stranice

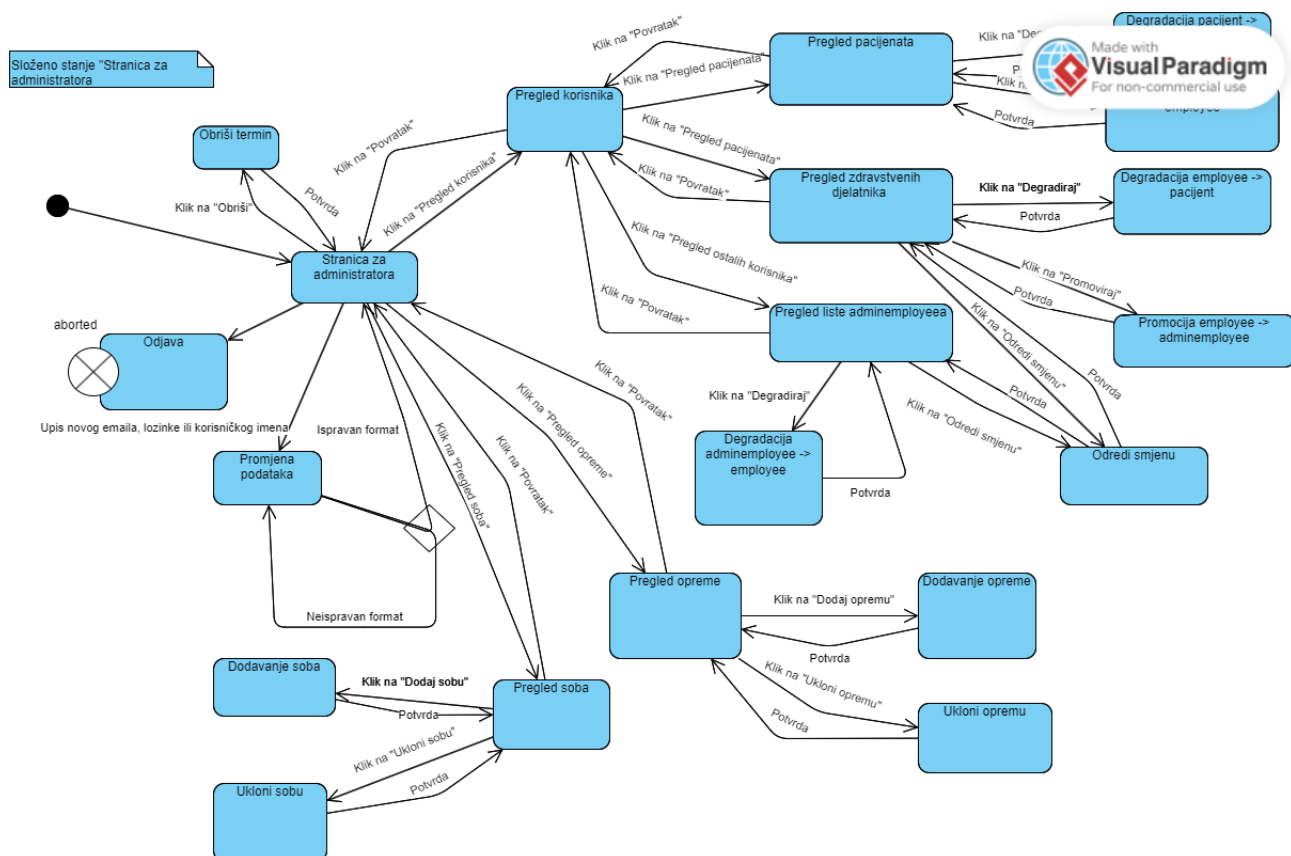


Slika 4.9: Složeno stanje "Stranica za pacijenta"



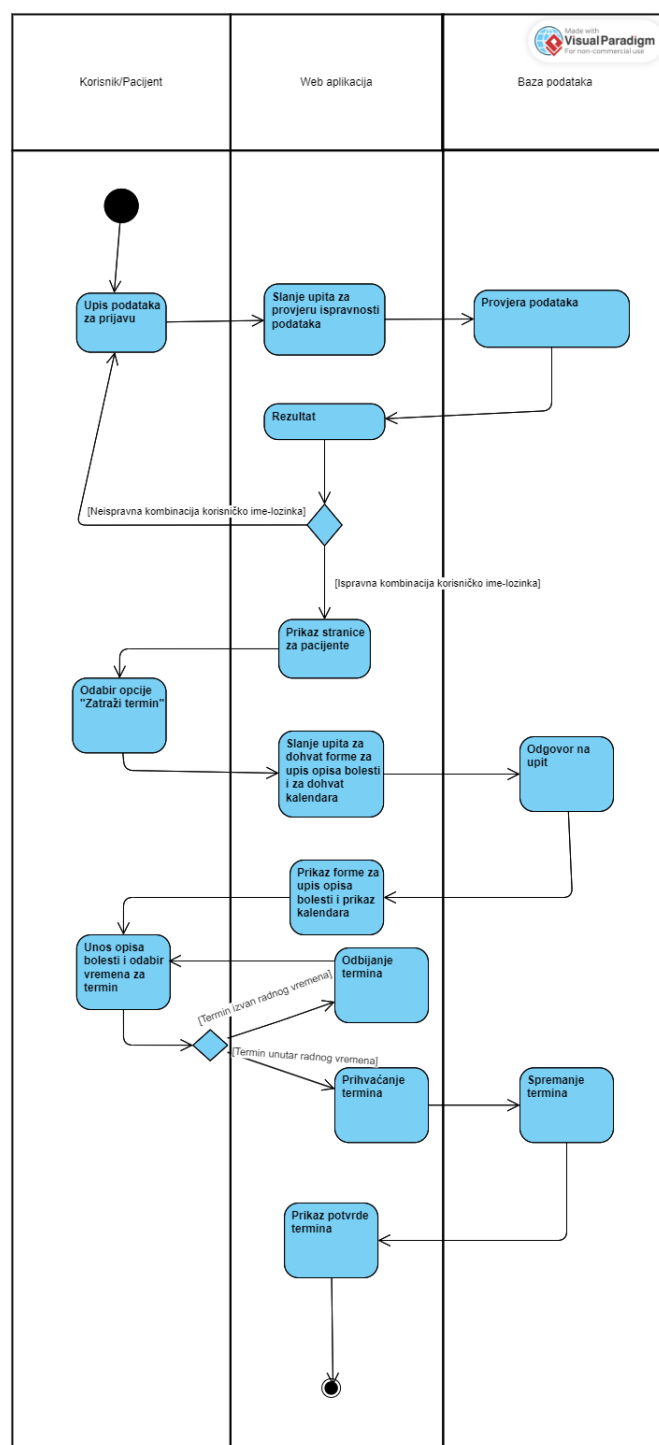
Slika 4.10: Složeno stanje "Stranica za zdravstvenog djelatnika"

Složeno stanje "Stranica za adminemployee" je vrlo slična složenom stanju "Stranica za zdravstvenog djelatnika" uz još dodatnu mogućnost dodavanja opreme i soba. Iz tog razloga smatramo da nije potrebno postaviti i taj dijagram stanja (jer je to kombinacija stanja iz složenog stanja "Stranica za zdravstvenog djelatnika" i "Stranica za administratora").

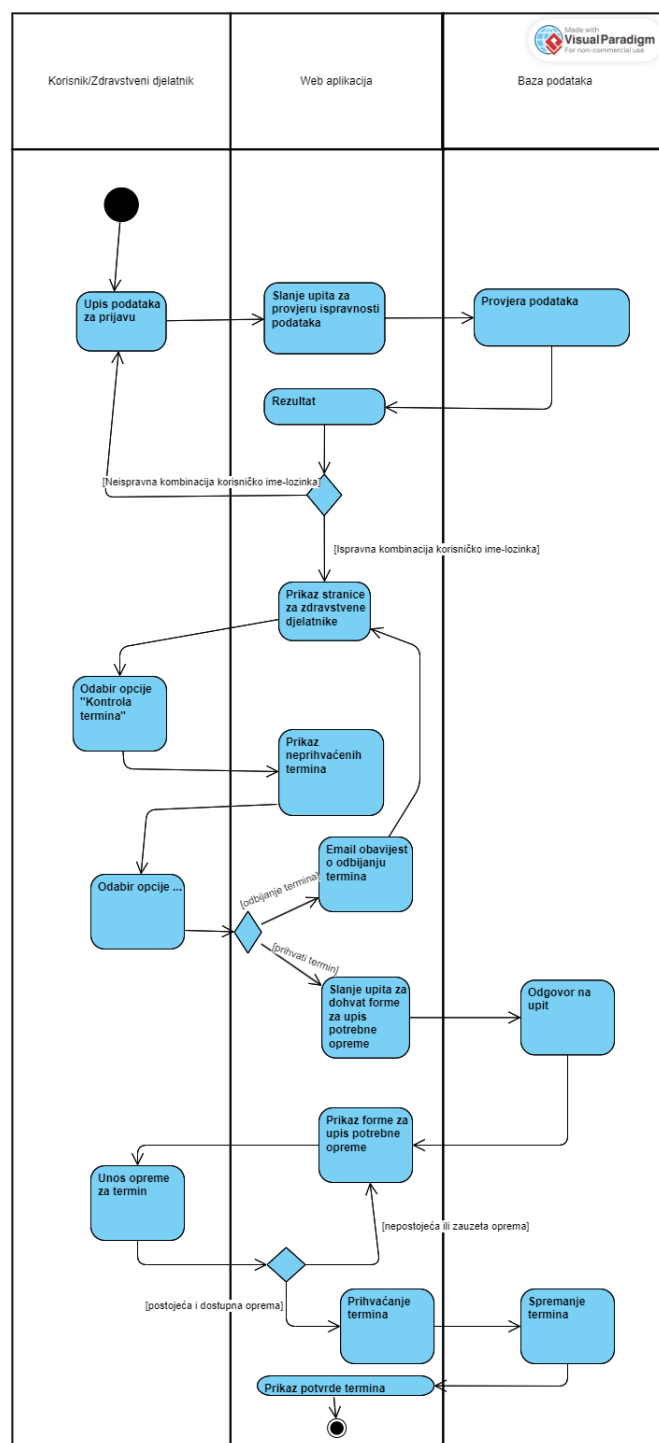


Slika 4.11: Složeno stanje "Stranica za administratora"

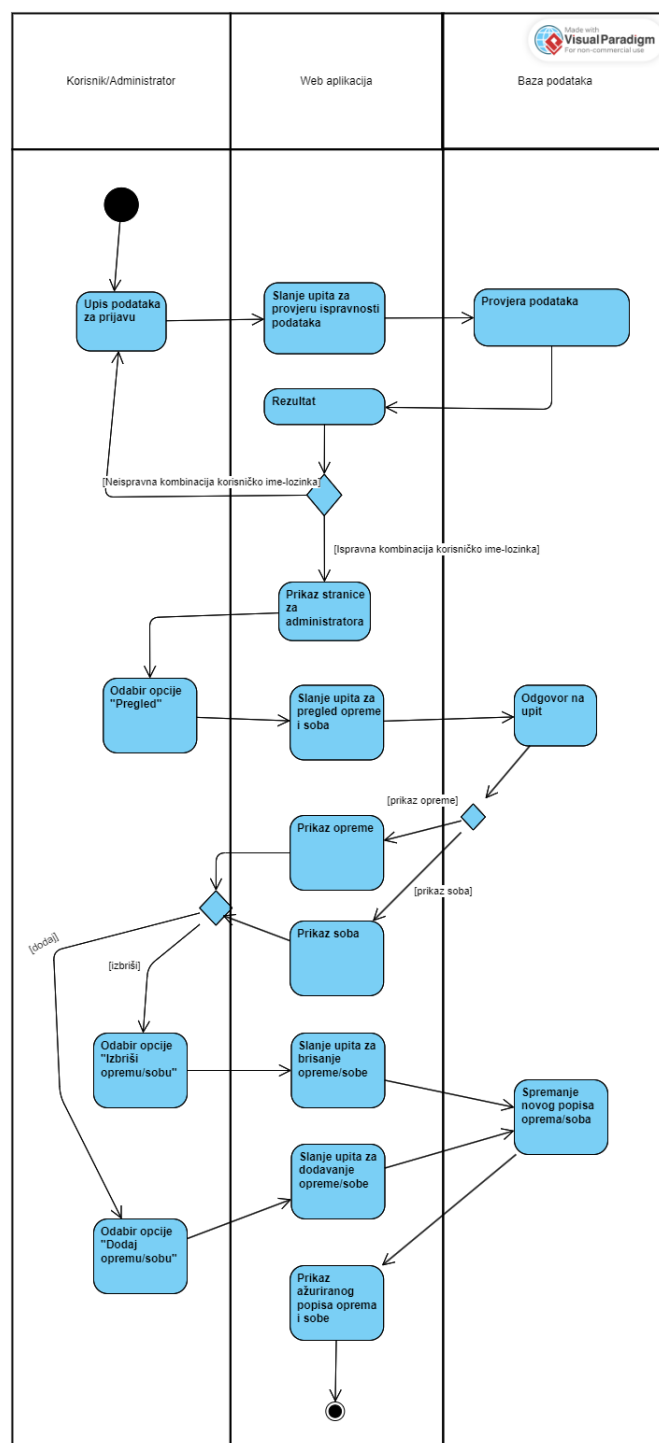
4.4 Dijagram aktivnosti



Slika 4.12: Prikaz dijagrama aktivnosti pri zahtjevu za novi termin

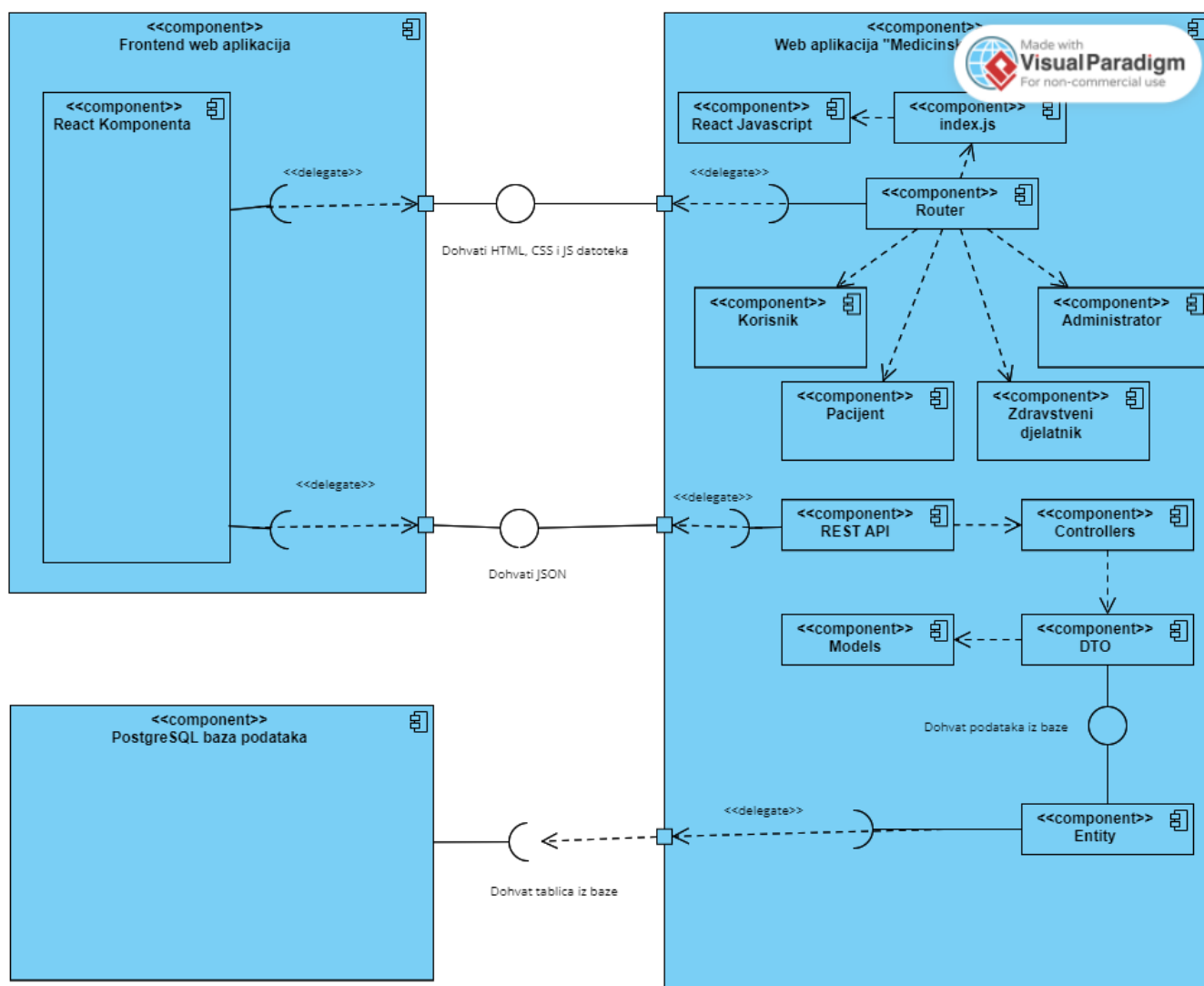


Slika 4.13: Prikaz dijagrama aktivnosti pri potvrđivanju/odbijanju termina



Slika 4.14: Prikaz dijagrama aktivnosti pri dodavanju/brisanju soba/opreme

4.5 Dijagram komponenti



Slika 4.15: Prikaz dijagrama komponenti naše web aplikacije

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu se obavljala pomoću aplikacija WhatsApp¹ i Discord². Dokumentacija je bila pisana pomoću alata TeXstudio i TexLive. Za izradu UML dijagrama korišten je alat Astah Professional³. Kao razvojno okruženje korišten je IntelliJ IDEA Ultimate⁴. IntelliJ IDEA je integrirano razvojno okruženje napisano u Javi za razvoj računalnog softvera napisanog u Javi, Kotlinu, Groovyju i drugim jezicima koji se temelje na JVM-u. Razvio ga je JetBrains i dostupan je u dva izdanja, Ultimate i Community. Aplikacija je napisana koristeći radni okvir Spring Boot⁵ i jezik Java⁶ za izradu backenda. Za izradu frontenda je korišten React⁷. Za izradu naše baze podataka koristili smo PostgreSQL⁸.

¹ Aplikacija za besplatnu razmjenu poruka, fotografija, videozapisa i drugih datoteka, te uspostavljanje glasovnih i videopoziva putem mobilnog interneta pametnim telefonima:
<https://www.whatsapp.com/>

² Besplatna aplikacija za komunikaciju putem tekstualnih poruka i glasovnih i video poziva:
<https://discord.com/>

³ <https://astah.net/products/astah-professional/>

⁴ <https://www.jetbrains.com/idea/>

⁵ <https://spring.io/projects/spring-boot>

⁶ <https://www.java.com/en/>

⁷ Besplatna JavaScript biblioteka otvorenog tipa za izgradnju korisničkih sučelja:
<https://reactjs.org/>

⁸ <https://www.postgresql.org/>

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 Selenium⁹. 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:

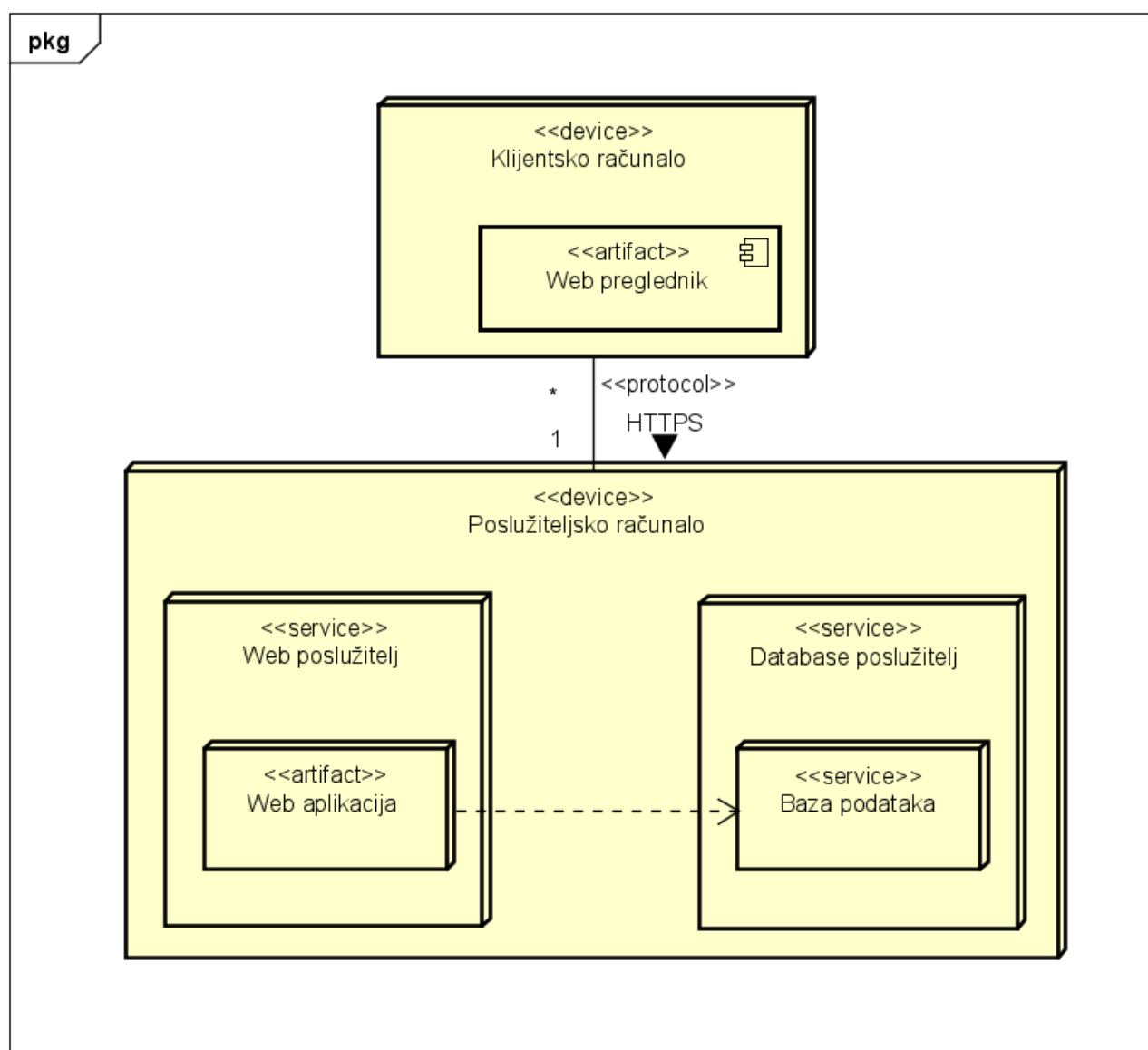
- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

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

⁹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

U ovom dijagramu razmještaja opisana je programska potpora i topologija sklopovlja korištena u radnom okruženju sustava. Sustav je izveden u arhitekturi "klijent-poslužitelj" i podijeljen je na dva dijela, na poslužiteljsko i korisničko računalo. Baza podataka i web poslužitelj nalaze se na poslužiteljskom računalo, klijent se koristi web preglednikom kako bi pristupio web aplikaciji, a sva komunikacija između klijenta i poslužitelja odvija se HTTPS vezom.



Slika 5.1: Specifikacijski dijagram razmještaja

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

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. Nikolina Frid, Alan Jović, "Modeliranje programske potpore UML-dijagramima", listopad 2023.
4. Andrea Danzante, "Korištenje razvojnog okvira Spring Boot za implementaciju servisa SOAP i REST", <https://repozitorij.foi.unizg.hr/islandora/object/foi%3A5416/datastream/PDF/view>
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	Naslovna stranica "YouCanBookMe" aplikacije	6
2.2	Primjer registracije, klikom na dugme "Create account" kreiramo korisnički račun	7
2.3	Primjer postavljanja dostupnih termina	7
2.4	Primjer ispunjavanja podataka nakon odabira jednog od dostupnih termina	8
3.1	Dijagram obrasca uporabe koji prikazuje funkcionalnost korisnika i pacijenta	21
3.2	Dijagram obrasca uporabe koji prikazuje funkcionalnost zdravstvenih djelatnika	22
3.3	Dijagram obrasca uporabe koji prikazuje funkcionalnost administratora	23
3.4	Sekvencijski dijagram za UC3.1.	25
3.5	Sekvencijski dijagram za UC3.2.	26
3.6	Sekvencijski dijagram za UC6.1	28
4.1	Prikaz HTTP komunikacije između klijenta i poslužitelja	30
4.2	Neke od koristi Spring Boot-a	31
4.3	MySQL	32
4.4	Relacijski model baze podataka	36
4.5	ER model baze podataka	36
4.6	Dijagram razreda - Modeli	38
4.7	Dijagram razreda - Kontroleri	40
4.8	Dijagram stanja početne stranice	41
4.9	Složeno stanje "Stranica za pacijenta"	41
4.10	Složeno stanje "Stranica za zdravstvenog djelatnika"	42
4.11	Složeno stanje "Stranica za administratora"	43
4.12	Prikaz dijagrama aktivnosti pri zahtjevu za novi termin	44
4.13	Prikaz dijagrama aktivnosti pri potvrđivanju/odbijanju termina	45
4.14	Prikaz dijagrama aktivnosti pri dodavanju/brisanju soba/opreme	46

4.15 Prikaz dijagrama komponenti naše web aplikacije	47
5.1 Specifikacijski dijagram razmještaja	50

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 23. listopad 2023.
- Prisustvovali: Benjamin Gregov, Filip Grgić, Marko Lipovac, Tin Ogri-zek, Bruno Petković, Filip Posavec, Neven Pralas
- Teme sastanka:
 - diskusija o zadatku, organiziran cijeli plan rada
 - podjela tima - tko će što raditi

2. sastanak

- Datum: u ovom formatu: 04. studenti 2023.
- Prisustvovali: Benjamin Gregov, Filip Grgić, Marko Lipovac, Tin Ogri-zek, Bruno Petković, Filip Posavec, Neven Pralas
- Teme sastanka:
 - diskusija o napretku svakog pojedinog člana tima, ali i tima zajedno
 - rješenje problema koji se pojavio jednom članu tima, pokušaj usmje-renja članova tima na pravi put

3. sastanak

- Datum: u ovom formatu: 13. studenti 2023.
- Prisustvovali: Benjamin Gregov, Filip Grgić, Marko Lipovac, Tin Ogri-zek, Bruno Petković, Filip Posavec, Neven Pralas
- Teme sastanka:
 - diskusija o napretku svakog pojedinog člana tima, ali i tima zajedno
 - diskusija o tome što smo dobro radili za prvo bodovanje, a što tre-bamo u sljedećem ciklusu popraviti
 - završavanje projekta za prvo bodovanje projekta

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Benjamin Gregov	Filip Grgić	Marko Lipovac	Bruno Petković	Filip Posavec	Tin Ogrizek	Neven Pralas
Upravljanje projektom	4						
Opis projektnog zadatka							3
Funkcionalni zahtjevi							1
Opis pojedinih obrazaca							4
Dijagram obrazaca							2
Sekvencijski dijagrami						4	
Opis ostalih zahtjeva						1	
Arhitektura i dizajn sustava					1		2
Baza podataka				6			
Dijagram razreda						3	
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Benjamin Gregov	Filip Grgić	Marko Lipovac	Bruno Petković	Filip Posavec	Tin Ogrizek	Neven Pralas
Dnevnik sastajanja							1
Zaključak i budući rad							
Popis literature							1
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>izrada početne stranice</i>		6			2	2	
<i>izrada pacijent stranice</i>					1		
<i>izrada djelatnik stranice</i>					1		
<i>izrada login stranice</i>	9	2					
<i>izrada registracijske stranice</i>			2		10		
<i>izrada admin stranice</i>			11			2	
<i>izrada baze podataka</i>				7			
<i>spajanje s bazom podataka</i>	2			3		1	
<i>back end</i>	7	7	3	4	2	8	7
<i>front end</i>	4	6	6	2	7	2	2

Dijagrami pregleda promjena

dio 2. revizije

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