

# Programsko inženjerstvo

Ak. god. 2023./2024.

## *Ozdravi*

Dokumentacija, Rev. 1.9

Grupa: *Welebyte*

Voditelj: *Dorian Matić*

Datum predaje: *17. studenoga 2023.*

Nastavnik: *Ivana Lulić*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>6</b>
<b>3 Specifikacija programske potpore</b>	<b>11</b>
3.1 Funkcionalni zahtjevi . . . . .	11
3.1.1 Obrasci uporabe . . . . .	13
3.1.2 Sekvencijski dijagrami . . . . .	28
3.2 Ostali zahtjevi . . . . .	31
<b>4 Arhitektura i dizajn sustava</b>	<b>32</b>
4.1 Baza podataka . . . . .	35
4.1.1 Opis tablica . . . . .	35
4.1.2 Dijagram baze podataka . . . . .	40
4.2 Dijagram razreda . . . . .	41
4.3 Dijagram stanja . . . . .	44
4.4 Dijagram aktivnosti . . . . .	46
4.5 Dijagram komponenti . . . . .	48
<b>5 Implementacija i korisničko sučelje</b>	<b>49</b>
5.1 Korištene tehnologije i alati . . . . .	49
5.2 Ispitivanje programskog rješenja . . . . .	51
5.2.1 Ispitivanje komponenti . . . . .	51
5.2.2 Ispitivanje sustava . . . . .	57
5.3 Dijagram razmještaja . . . . .	60
5.4 Upute za puštanje u pogon . . . . .	61
<b>6 Zaključak i budući rad</b>	<b>64</b>
<b>Popis literature</b>	<b>66</b>
<b>Indeks slika i dijagrama</b>	<b>67</b>

**Dodatak: Prikaz aktivnosti grupe**

**68**

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	L. Crvelin	24.10.2023.
0.2	Napisan opis projektnog zadatka	L. Crvelin	27.10.2023.
0.3.1	Napisani funkcionalni zahtjevi	L. Crvelin	27.10.2023.
0.3.2	Izmijenjeni funkcionalni zahtjevi	L. Crvelin	31.10.2023.
0.4.1	Dodani opisi obrazaca uporabe	L. Crvelin	2.11.2023.
0.4.2	Izmijenjeni opisi obrazaca uporabe	L. Crvelin	2.11.2023.
0.5	Opis baze, opis tablica	L. Crvelin	2.11.2023.
0.6	Arhitektura i dizajn sustava	D. Matić	6.11.2023.
0.7	Dijagram baze podataka, dopisan opis tablica	L. Crvelin	6.11.2023.
0.8	Dopunjeni opisi obrazaca uporabe	L. Crvelin	7.11.2023.
0.9	Promijenjen dijagram baze podataka	L. Crvelin	8.11.2023.
0.10.1	Izmjena opisa obrazaca uporabe	L. Crvelin	8.11.2023.
0.10.2	Izmjena opisa obrazaca uporabe i tablica entiteta	L. Crvelin	9.11.2023.
0.11	Dodani dijagrami obrazaca uporabe	L. Crvelin	11.11.2023.
0.12.1	Započeti sekvencijski dijagrami	L. Crvelin	11.11.2023.
0.12.2	Završeni sekvencijski dijagrami	L. Crvelin	12.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.13	Opis ostalih zahtjeva	L. Crvelin	12.11.2023.
0.14.1	Izmjena i dopuna odlomka "Arhitektura i dizajn sustava"	D.Matić	13.11.2023.
0.14.2	Uređivanje dokumenta	L. Crvelin	13.11.2023.
0.14.3	Ispunjena tablica aktivnosti	L. Crvelin	16.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	L. Crvelin	17.11.2023.
1.1	Izmijenjeni opisi obrazaca uporabe i dijagrami obrazaca uporabe	L. Crvelin	7.12.2023.
1.2	Napisane upute za puštanje u pogon	V. Kumarnović	8.12.2023.
1.3	Dijagram aktivnosti	L. Crvelin	13.12.2023.
1.4	Dijagram razmještaja	V. Kumarnović	14.12.2023.
1.5	Korišteni alati i tehnologije	L. Crvelin	18.12.2023.
1.6	Zaključak i budući rad	L. Crvelin	21.12.2023.
1.7	Izmjena funkcionalnih zahtjeva i opisa obraza uporabe	L. Crvelin	21.12.2023.
1.7.1	Izmjena UC dijagrama, sekvencijskih dijagrama i dijagrama aktivnosti	L. Crvelin	23.12.2023.
1.8	Dijagram stanja i komponenti	V. Kumarnović	28.12.2023.
1.9	Dijagram razreda II	V. Kumarnović	8.1.2024.
1.9.1	Izmjena u dijagramima razreda	V. Kumarnović	12.1.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.10	Dokumentacija testiranja	D. Matić	12.1.2024.

## 2. Opis projektnog zadatka

Projekt "Ozdravi" predstavlja inovativno rješenje razvijeno kako bi se značajno olakšao svakodnevni život roditelja s djecom koja često zahtijevaju medicinsku skrb. U današnjem užurbanom svijetu, roditelji, osobito oni s više djece, suočavaju se s izazovom istovremenog angažmana na radnom mjestu i brige o zdravstvenim potrebama svoje djece. To često dovodi do gubitka vremena zbog administrativnih procedura koje se odnose na posjete liječnicima, izdavanje doznaka za bolovanje i ispričnica za školu ili vrtić. Projekt "Ozdravi" nastao je kako bi riješio ove izazove i omogućio bolju koordinaciju između roditelja, pedijatara i liječnika obiteljske medicine. Ova aplikacija omogućuje roditeljima, pedijatrima i liječnicima obiteljske medicine učinkovitiju komunikaciju, brže izdavanje doznaka za bolovanje i ispričnica za školu i vrtić, te praćenje medicinskih informacija djece.

Ciljevi projekta "Ozdravi" obuhvaćaju niz aspekata s ciljem poboljšanja kvalitete brige o zdravlju djece i olakšavanja svakodnevnih izazova roditelja.

Prvenstveno, svrha aplikacije je olakšati koordinaciju i poboljšati komunikaciju između roditelja, pedijatara i liječnika obiteljske medicine tijekom procesa brige o zdravstvenim potrebama djece. Time je omogućena bolja suradnja i razmjena informacija između ključnih dionika.

Drugi cilj je automatizacija procesa izdavanja doznaka za bolovanje i ispričnica za školu i vrtić. Ovime se smanjuju administrativne prepreke i olakšava roditeljima dobivanje potrebnih dokumenata, čime se značajno štedi vrijeme.

Treći cilj projekta je pružiti roditeljima brz pristup medicinskim informacijama i rezultatima pregleda njihove djece. Ova dostupnost informacija omogućuje roditeljima dobru informiranost o zdravstvenom stanju svoje djece.

Naposljetku, projekt "Ozdravi" također ima cilj smanjiti administrativni teret i stres povezan s upravljanjem medicinskim potrebama djece. Rezultat su manji administrativni zadaci čime je omogućen veći fokus na brigu o zdravlju djece.

Svi ovi ciljevi zajedno čine projekt "Ozdravi" vrijednim inovativnim rješenjem koje ima potencijal značajno unaprijediti kvalitetu života roditelja i skrb o zdravlju djece.

Projekt "Ozdravi" transformira način na koji se obavlja komunikacija i administracija u kontekstu zdravstvene skrbi za djecu. Aplikacija korisnicima omogućuje registraciju i prijavu, nakon čega administrator priprema registre djece, uključujući osnovne podatke i OIB. Roditelji se mogu povezati sa svojom djecom putem OIB-a. Svaki roditelj može pregledavati svoj profil i profil svoje djece. Pedijatrima je omogućeno prijavljivanje djece putem OIB-a, pregled popisa prijavljene djece i njihovih kartona te evidentiranje pregleda i događaja na istima. U slučaju potrebe za izdavanjem preporuke za bolovanje roditelju, pedijatar će to izdati, a liječnik obiteljske medicine će je odobriti i poslati doznaku poslodavcu roditelja. Aplikacija omogućuje roditeljima učitavanje medicinskih nalaza dobivenih u privatnim ustanovama i potraživanje povratnih informacija od liječnika ili pedijatra. Roditelji će primati obavijesti i upute od strane pedijatra ili liječnika obiteljske medicine nakon što stignu nalazi iz laboratorija. Liječnici obiteljske medicine i pedijatri mogu naručiti pacijenta na specijalistički pregled ili postupak, nakon čega pacijent dobiva poruku s potvrdom naručivanja i prikazom lokacija na mapi gdje može obaviti pregled.

Projekt obuhvaća četiri ključne korisničke uloge:

1. Administrator: Osoba odgovorna za upravljanje registrima djece i korisničkim računima
2. Pedijatar: Liječnik specijaliziran za dječju medicinu koji će unositi medicinske podatke i izdavati preporuke za bolovanje te ispričnice za školu i vrtić
3. Liječnik obiteljske medicine: Liječnik koji će odobravati izdane preporuke i slati ih poslodavcima roditelja
4. Roditelji: Korisnici koji će pratiti zdravstvene podatke svoje djece i komunicirati s liječnicima

Projekt "Ozdravi" nudi niz ključnih funkcionalnosti:

- Registracija i prijava korisnika.
- Administracija registra djece i korisničkih računa.



- Praćenje profila za roditelje i njihovu djecu.
- Unos medicinskih podataka, izdavanje preporuka za bolovanje i ispričnica od strane pedijatara.
- Odobranje preporuka i slanje doznaka za bolovanje od strane liječnika obiteljske medicine.
- Praćenje medicinskih nalaza i upita od strane roditelja.
- Specijalistički pregledi i usmjerenja prema lokacijama na mapi.
- Povijest posjeta i dijagnoza.
- Administracija korisničkih računa i ažuriranje podataka.

Postojeće slično rješenje Projektu "Ozdravi" je Portal zdravlja na e-Građani. Obje aplikacije su usmjerene na poboljšanje pristupa i upravljanja zdravstvenim informacijama i uslugama, ali imaju različite svrhe, korisnike i dosege.

#### 1. Svrha projekta:

- Ozdravi: Osnovna svrha projekta "Ozdravi" je olakšati koordinaciju između roditelja, pedijatara i liječnika obiteljske medicine te omogućiti brže izdavanje doznaka za bolovanje i ispričnica za školu i vrtić, s posebnim fokusom na djecu.
- Portal zdravlja na e-Građani: Portal zdravlja e-Građani je integrirana online platforma koja omogućuje građanima pristup vlastitim zdravstvenim podacima, elektroničkim receptima, laboratorijskim nalazima i drugim relevantnim informacijama.

#### 2. Korisnici:

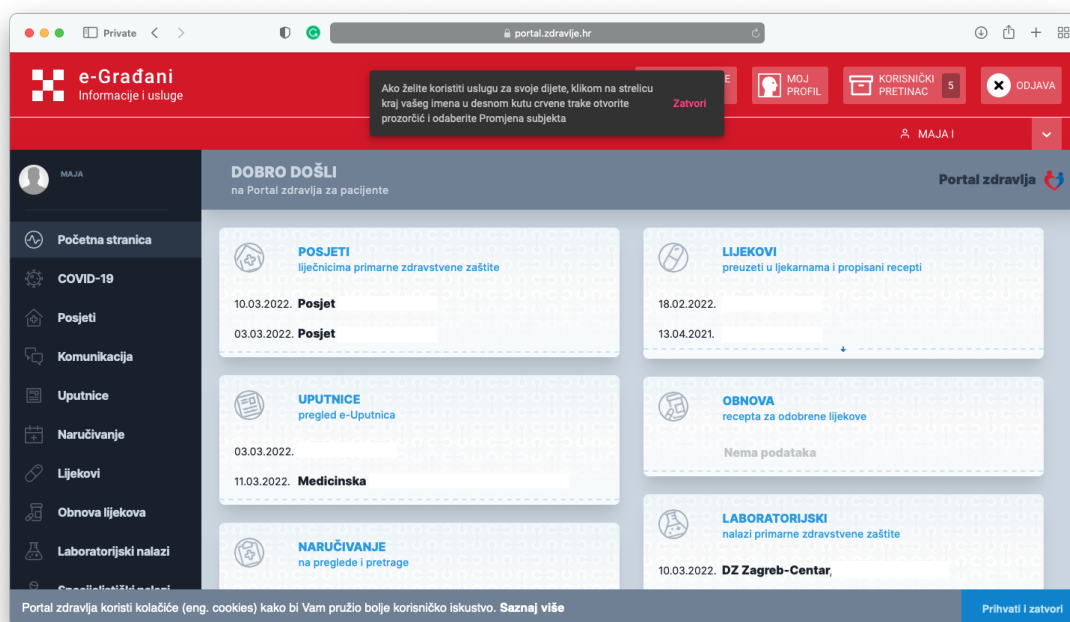
- Ozdravi: Projekt "Ozdravi" fokusira se na roditelje djece koja često zatražuju medicinsku skrb, pedijatre i liječnike obiteljske medicine. Osnovna svrha je poboljšati skrb o zdravlju djece i olakšati roditeljima administrativne postupke.
- Portal zdravlja na e-Građani: Portal zdravlja na e-Građani je dostupan svim građanima koji su korisnici javnog zdravstvenog sustava u Republici Hrvatskoj. Omogućuje im pristup vlastitim zdravstvenim podacima, receptima i drugim informacijama.

#### 3. Funkcionalnosti:

- Ozdravi: Osnovne funkcionalnosti projekta "Ozdravi" uključuju registraciju korisnika, upravljanje profilima djece, izdavanje preporuka i doz-

naka, praćenje medicinskih nalaza, te komunikaciju između korisnika i zdravstvenih stručnjaka.

- Portal zdravlja na e-Građani: Portal zdravlja na e-Građani omogućuje pristup elektroničkim zdravstvenim karticama, povijesti bolesti, laboratorijskim nalazima, eReceptima te pristup raznim zdravstvenim uslugama kao što su naručivanje na preglede.



Slika 2.1: Početna stranica "Portal zdravlje"

U konačnici, i "Ozdravi" i Portal zdravlja na e-Građani predstavljaju korisna digitalna rješenja za bolje upravljanje zdravstvenim informacijama, ali se razlikuju u svojoj svrsi, opsegu i ciljanoj publici. Oba projekta doprinose unaprjeđenju zdravstvene skrbi, svaki u svojem specifičnom kontekstu.

Projekt "Ozdravi" je izveden kao web aplikacija, prilagođena različitim uređajima, uključujući mobilne uređaje, tablete i računalima. Osim toga, aplikacija je skalabilna i prilagodljiva za različite regije i zdravstvene ustanove.

Opseg projektnog zadatka "Ozdravi" obuhvaća sve navedene funkcionalnosti i složenu koordinaciju između korisnika. Aplikacija omogućava brzu i preciznu komunikaciju između svih relevantnih strana. Kroz jednostavno i intuitivno sučelje,

aplikacija omogućuje roditeljima, pedijatrima i liječnicima učinkovitije upravljanje zdravstvenim potrebama djece.

Aplikacija pruža mogućnost za daljnji razvoj i proširenje, što će dodatno poboljšati iskustvo korisnika i olakšati njihovu svakodnevicu. Ova inovacija ima potencijal da unaprijedi kvalitetu života roditelja i omogući bolju skrb za djecu. Budući razvoj projekta "Ozdravi" može uključivati niz nadogradnji, uključujući integraciju s telemedicinom za daljinske konzultacije, razvoj mobilne aplikacija za veću dostupnost, proširenje podrške za različite jezike i regionalne prakse te poboljšanje analitičkih alata za praćenje medicinskih podataka.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Roditelj
2. Pedijatar
3. Liječnik obiteljske medicine
4. Administrator

#### Aktori i njihovi funkcionalni zahtjevi:

##### 1. Roditelj (inicijator) može:

- (a) registrirati se i prijaviti u sustav
- (b) vidjeti profil za sebe i svoju djecu
- (c) pregledati medicinske nalaze, dijagnoze i povijesti pregleda za sebe i svoju djecu
- (d) unijeti nalaz dobiven temeljem određenog pregleda ili postupka u privatnoj ustanovi te zatražiti drugo mišljenje liječnika ili pedijatra
- (e) vidjeti zahtjeve za drugim mišljenjima koje je stvorio
- (f) vidjeti upute koje su on ili njegova djeca dobili od liječnika/pedijatra

##### 2. Pedijatar (inicijator) može:

- (a) prijaviti se u sustav
- (b) prijaviti djecu po identifikatoru (OIB)
- (c) pregledati popis sve prijavljene djece i njihove kartone
- (d) unijeti medicinske podatke (dijagnoze, preglede, terapije)
- (e) evidentirati preporuku za bolovanje za roditelja u slučaju bolesti djeteta
- (f) utvrditi bolest djeteta nakon čega se šalje ispričnica u vrtić
- (g) naručiti djecu na specijalističke preglede/postupke
- (h) izdati uputu pacijentu
- (i) dati drugo mišljenje

##### 3. Liječnik obiteljske medicine (inicijator) može:

- (a) prijaviti se u sustav
- (b) pregledati i odobriti preporuke za bolovanje izdane od strane pedijatra
- (c) slati doznake za bolovanje poslodavcu roditelja
- (d) izdati uputu pacijentu
- (e) dati drugo mišljenje
- (f) naručiti pacijenta na specijalistički pregled/postupak
- (g) unositi podatke o pregledima
- (h) pristupiti medicinskim nalazima, povijesti pregleda i upitima

4. Administrator (inicijator) može:

- (a) prijaviti se u sustav kao administrator
- (b) pripremiti registre djece (ime, prezime, OIB)
- (c) registrirati liječnika obiteljske medicine i pedijatra
- (d) po registraciji roditelja, povezati iste s djecom (preko OIB-a)
- (e) administrirati korisničke račune
- (f) ažurirati sve podatke u aplikaciji

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - registriraj se

- **Glavni sudionik:** korisnik
- **Cilj:** stvoriti korisnički račun za pristup sustavu
- **Sudionici:** baza podataka
- **Preduvjet:** korisnik je otvorio početnu stranicu za registraciju u aplikaciju
- **Opis osnovnog tijeka:**
  1. korisnik odabire opciju registriraj se
  2. sustav otvara ekran registracije
  3. korisnik unosi potrebne korisničke podatke
  4. korisnik potvrđuje unos podataka odabirom akcije potvrdi
  5. sustav provjerava i utvrđuje da je unos uspješan
  6. sustav obavještava korisnika o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 4.a Korisnik odustaje od unosa podatka (odabir akcije odustani)
    1. sustav zatvara trenutni ekran
    2. izvođenje scenarija je završeno
  - 5.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkih podatka u nedozvoljenom formatu
    1. sustav obavještava korisnika o neuspjelom upisu
    2. sustav nastavlja izvođenje u koraku 3

##### UC2 - prijavi se

- **Glavni sudionik:** korisnik
- **Cilj:** dobiti pristup korisničkom sučelju
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC1
- **Opis osnovnog tijeka:**
  1. korisniku se po upisu web adrese aplikacije, otvara početna stranica
  2. korisnik odabire akciju prijavi se
  3. sustav otvara ekran prijave
  4. korisnik unosi email i lozinku
  5. korisnik odabire akciju potvrde unosa

6. sustav provjerava i utvrđuje da je unos uspješan
7. korisniku se prikazuje početni zaslon aplikacije
- **Opis mogućih odstupanja:**
  - 5.a Korisnik odabire akciju Odustani
    1. sustav zatvara ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija
  - 6.a sustav provjerava i utvrđuje da unos podataka nije uspješan
    1. sustav obavještava korisnika porukom na modalnom ekranu o unosu pogrešne lozinke
    2. sustav nastavlja izvođenje u koraku 4. (korisnik ispravlja unesenu lozinku i dalje scenarij ide svojim tijekom)

### UC3 - pregledaj profil djeteta

- **Glavni sudionik:** korisnik
- **Cilj:** pregledati djetetovu povijest pregleda, nalaze i dijagnoze
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC22, korisnik ima dijete
- **Opis osnovnog tijeka:**
  1. roditelj odabire opciju 'Moja djeca' na početnoj stranici
  2. sustav prikazuje popis djece koja su povezana s roditeljem
  3. roditelj odabire dijete čije podatke želi pregledati
  4. sustav prikazuje početnu stranicu djetetovog profila

### UC4 - pregledaj povijest pregleda svojeg djeteta

- **Glavni sudionik:** korisnik
- **Cilj:** pristupiti povijesti pregleda djeteta, za svaki pregled su vidljivi nalazi i dijagnoze ustanovljene na pregledu
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2, UC23
- **Opis osnovnog tijeka:**
  1. roditelj odabire opciju 'Moji pregledi' s početne stranice
  2. sustav prikazuje popis svih obavljenih i zakazanih pregleda korisnika i njegove djece
  3. korisnik filtrira popis pregleda tako da vidi samo djetetove preglede
  4. sustav prikazuje filtrirani popis pregleda
  5. korisnik odabire pregled za koji želi vidjeti podatke

6. sustav prikazuje podatke o odabranom pregledu
- **Opis mogućih odstupanja:**
  - 2.a korisnik nema nijedan zabilježeni pregled
    1. prikazuje se poruka 'Ne postoji nijedan zabilježeni pregled'
    2. završava izvođenje ovog scenarija

#### UC5 - pregledaj vlastiti profil

- **Glavni sudionik:** korisnik
- **Cilj:** pregledati vlastiti profil
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. korisnik odabire opciju 'Moj profil' s početne stranice
  2. sustav prikazuje osobne podatke korisnika

#### UC6 - ažuriraj e-mail adresu poslodavca

- **Glavni sudionik:** korisnik
- **Cilj:** ažurirati e-mail adresu svog poslodavca
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. korisnik odabire opciju 'Profil' s početne stranice
  2. sustav prikazuje osobne podatke korisnika
  3. korisnik pronalazi polje za unos e-mail adrese poslodavca
  4. korisnik unosi e-mail adresu poslodavca
  5. korisnik odabire opciju 'Spremi promjene'
  6. sustav provjerava i utvrđuje da je unos uspješan
  7. sustav obavještava korisnika da je unos uspješan
- **Opis mogućih odstupanja:**
  - 6.a korisnik je unio pogrešnu e-mail adresu
    1. prikazuje se poruka 'Pogrešan unos e-mail adrese'
    2. sustav vraća korisnika na ponovno upisivanje e-mail adrese
    3. scenarij se nastavlja od koraka 4

#### UC7 - ažuriraj e-mail adresu vrtića/škole

- **Glavni sudionik:** roditelj



- **Cilj:** ažurirati e-mail adresu djetetovog vrtića/škole
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC22
- **Opis osnovnog tijeka:**
  1. roditelj odabire opciju 'Moja djeca' s početne stranice
  2. sustav prikazuje popis djece koja su povezana s roditeljem
  3. roditelj odabire dijete čije podatke želi ažurirati
  4. sustav prikazuje osobne podatke o djetetu
  5. roditelj pronalazi polje za unos e-mail adrese vrtića/škole
  6. roditelj unosi e-mail adresu vrtića/škole
  7. roditelj odabire opciju 'Spremi promjene'
  8. sustav provjerava i utvrđuje da je unos uspješan
  9. sustav obavještava roditelja da je unos uspješan
- **Opis mogućih odstupanja:**
  - 8.a roditelj je unio pogrešnu e-mail adresu
    1. prikazuje se poruka 'Pogrešan unos e-mail adrese'
    2. sustav vraća roditelja na ponovno upisivanje e-mail adrese
    3. scenarij se nastavlja od koraka 6

#### UC8 - pregledaj medicinske podatke

- **Glavni sudionik:** korisnik
- **Cilj:** pristupiti povijesti pregleda, za svaki pregled su vidljivi nalazi i dijagnoze ustanovljene na pregledu
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. roditelj odabire opciju 'Moji pregledi' s početne stranice
  2. sustav prikazuje popis svih obavljenih i zakazanih pregleda
  3. korisnik odabire pregled za koji želi vidjeti podatke
  4. sustav prikazuje podatke o odabranom pregledu
- **Opis mogućih odstupanja:**
  - 2.a korisnik nema nijedan zabilježeni pregled
    1. prikazuje se poruka 'Ne postoji nijedan zabilježeni pregled'
    2. završava izvođenje ovog scenarija

#### UC9 - stvori zahtjev za drugim mišljenjem

- **Glavni sudionik:** korisnik
- **Cilj:** zatražiti drugo mišljenje liječnika ili pedijatra o nalazu dobivenom na temelju određene usluge u privatnoj ustanovi
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2, pacijent ima dodijeljenog liječnika ili pedijatra
- **Opis osnovnog tijeka:**
  1. korisnik odabire opciju 'Zatraži drugo mišljenje'
  2. sustav otvara modalni okvir unosa zahtjeva za drugim mišljenjem
  3. roditelj prenosi nalaz dobiven u privatnoj ustanovi
  4. roditelj odabire liječnika/pedijatra čije drugo mišljenje želi dobiti
  5. roditelj odabire akciju 'Pošalji zahtjev'
  6. sustav provjerava i utvrđuje da je unos uspješan
  7. prikazuje se poruka 'Zahtjev uspješno poslan'
- **Opis mogućih odstupanja:**
  - 5.a neispravno popunjen zahtjev
    1. sustav utvrđuje da je zahtjev neispravno popunjen
    2. sustav usmjerava korisnika na ponovno ispunjavanje zahtjeva za drugim mišljenjem
    3. nakon ispravnog unosa, korisnik odabire akciju 'Pošalji zahtjev' te se scenarij nastavlja od koraka 6
  - 5.b korisnik odabire akciju 'Odustani'
    1. sustav zatvara modalni ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### UC10 - Pregledaj zahtjev za drugim mišljenjem

- **Glavni sudionik:** korisnik
- **Cilj:** pregledati predani zahtjev za drugim mišljenjem te liječnikovu ili pedijatrovu povratnu informaciju
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC9
- **Opis osnovnog tijeka:**
  1. korisnik odabire opciju 'Moji zahtjevi' s početne stranice
  2. sustav prikazuje popis svih korisnikovih zahtjeva za drugim mišljenjem
  3. korisnik odabire zahtjev koji želi pregledati
  4. sustav prikazuje korisnikov zahtjev za drugim mišljenjem
  5. korisnik pregledava liječnikov odgovor na zahtjev

- **Opis mogućih odstupanja:**

5.a liječnik nije poslao svoje drugo mišljenje

1. roditelj može vidjeti samo svoj predani zahtjev
2. sustav prikazuje poruku 'Liječnik nije poslao svoje drugo mišljenje!' u polju za odgovor liječnika
3. završava se izvođenje ovog scenarija

### UC11 - prijavi pacijenta u svoju ordinaciju

- **Glavni sudionik:** pedijatar, liječnik obiteljske medicine

- **Cilj:** prijaviti pacijenta u svoju ordinaciju

- **Sudionici:** baza podataka

- **Preduvjet:** izvođenje UC2

- **Opis osnovnog tijeka:**

1. liječnik odabire opciju 'Pacijenti' s početne stranice
2. liječnik odabire opciju 'Prijavi pacijenta' s početne stranice
3. sustav prikazuje popis pacijenata koji nemaju pridijeljenog liječnika
4. liječnik odabire pacijenta koji postaje njegov pacijent odabirom akcije 'Odaberi pacijenta'
5. sustav obavještava liječnika da je pacijent uspješno odabran i prijavljen u njegovu ordinaciju

- **Opis mogućih odstupanja:**

3.a ne postoji nijedan pacijent koji nema pridijeljenog liječnika

1. sustav umjesto popisa pacijenata, prikazuje poruku 'Svi pacijenti imaju pridijeljenog liječnika'

4.a liječnik odustaje od prijave pacijenta

1. sustav zatvara trenutni ekran
2. završava se izvođenje ovog scenarija

### UC12 - pregledaj prijavljene pacijente i njihove kartone

- **Glavni sudionik:** pedijatar, liječnik obiteljske medicine

- **Cilj:** pregledati popis prijavljenih pacijenata i njihovih kartona

- **Sudionici:** baza podataka

- **Preduvjet:** izvođenje UC2

- **Opis osnovnog tijeka:**

1. liječnik odabire opciju 'Pacijenti' s početne stranice

2. sustav prikazuje popis pacijenata koji su prijavljeni u liječnikovoj ordinaciji
  3. liječnik odabire pacijenta čiji karton želi pregledati i odabire opciju 'Pregledaj karton'
  4. sustav prikazuje medicinske podatke o odabranom pacijentu
- **Opis mogućih odstupanja:**
    - 2.a liječnik nema nijednog pacijenta
      1. sustav prikazuje poruku 'Nemate nijednog prijavljenog pacijenta'
      2. završava se izvođenje ovog scenarija

### UC13 - unesi pregled

- **Glavni sudionik:** pedijatar, liječnik obiteljske medicine
- **Cilj:** unijeti pregled te medicinske podatke vezane uz pregled: dijagnoza, nalaz, terapija
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC11
- **Opis osnovnog tijeka:**
  1. liječnik odabire opciju 'Pregledi' s početne stranice
  2. liječnik odabire opciju 'Novi pregled' s početne stranice
  3. sustav prikazuje modalni ekran za unos pregleda
  4. liječnik odabire pacijenta iz padajućeg izbornika za kojega želi unijeti novi pregled
  5. liječnik unosi podatke o pregledu
  6. liječnik odabire opciju 'Unesi pregled'
  7. sustav provjerava i utvrđuje da je unos uspješan
  8. sustav obavještava liječnika da je pregled uspješno unesen
- **Opis mogućih odstupanja:**
  - 7.a liječnik je pogrešno unio podatke o pregledu
    1. sustav prikazuje poruku 'Neispravan unos podataka o pregledu'
    2. sustav vraća liječnika na izmjenju pogrešnih podataka o pregledu
    3. scenarij se nastavlja od koraka 5
  - 7.b liječnik odabire opciju 'Odustani'
    1. sustav zatvara modalni ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

### UC14 - evidentiraj preporuku za bolovanje

- **Glavni sudionik:** pedijatar
- **Cilj:** evidentirati preporuku za bolovanje roditelja u slučaju bolesti djeteta
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC13
- **Opis osnovnog tijeka:**
  1. pedijatar odabire opciju 'Preporuke za bolovanje' s početne stranice
  2. pedijatar odabire opciju 'Nova preporuka za bolovanje' s početne stranice
  3. sustav prikazuje modalni okvir za unos preporuke za bolovanje
  4. pedijatar odabire pregled na temelju kojeg je potrebna preporuka za bolovanje
  5. pedijatar unosi podatke o preporuci za bolovanje
  6. pedijatar odabire opciju 'Unesi preporuku'
  7. sustav provjerava i potvrđuje da je unos uspješan
  8. sustav obavještava pedijatra da je preporuka uspješno unesena
- **Opis mogućih odstupanja:**
  - 6.a liječnik odabire opciju 'Odustani'
    1. sustav zatvara modalni ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### UC15 - pošalji ispričnicu u vrtić/školu

- **Glavni sudionik:** pedijatar
- **Cilj:** poslati ispričnicu zbog bolesti djeteta u djetetov vrtić/školu
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC13
- **Opis osnovnog tijeka:**
  1. pedijatar odabire opciju 'Pregledi' s početne stranice
  2. pedijatar odabire pregled na temelju kojega je potrebno poslati ispričnicu
  3. pedijatar unosi podatke o ispričnici
  4. pedijatar odabire opciju 'Pošalji ispričnicu'
  5. sustav provjerava i potvrđuje da je unos uspješan
  6. sustav obavještava pedijatra da je ispričnica uspješno poslana
- **Opis mogućih odstupanja:**
  - 5.a pedijatar je neispravno popunio ispričnicu
    1. sustav prikazuje poruku 'Neispravan unos podataka o ispričnici'
    2. pedijatra se vraća na ponovno popunjavanje ispričnice

3. scenarij se nastavlja od koraka 3
- 5.b liječnik odabire opciju 'Odustani'
  1. sustav zatvara modalni ekran
  2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### **UC16 - naruči pacijenta na specijalistički pregled/postupak**

- **Glavni sudionik:** pedijatar, liječnik obiteljske medicine
- **Cilj:** naručiti pacijenta na specijalistički pregled/postupak
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC11
- **Opis osnovnog tijeka:**
  1. liječnik pregledava popis svojih pacijenata
  2. liječnik odabire pacijenta kojeg želi naručiti na specijalistički pregled/postupak
  3. liječnik stvara novi specijalistički pregled i zakaže vrijeme u budućnosti
  4. vrijeme i lokacija se šalju pacijenti te je vidljiv pod 'Moji pregledi'
- **Opis mogućih odstupanja:**
  - 5.a liječnik odabire opciju 'Odustani'
    1. sustav zatvara modalni ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### **UC17 - prihvati preporuku za bolovanje**

- **Glavni sudionik:** liječnik obiteljske medicine
- **Cilj:** pregledati preporuku za bolovanje roditelja izdanu od strane pedijatra
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC14
- **Opis osnovnog tijeka:**
  1. liječnik odabire opciju 'Preporuke za bolovanje' s početne stranice
  2. liječnik pregledava listu preporuka za bolovanje njegovim pacijentima
  3. liječnik odabire preporuku koju želi pregledati
  4. liječnik odabire opciju 'Prihvati' i prihvaća preporuku za bolovanje
  5. doznaka za bolovanje se šalje poslodavcu pacijenta

#### **UC18 - odbij odluku o preporuci za bolovanje**

- **Glavni sudionik:** liječnik obiteljske medicine
- **Cilj:** pregledati preporuku za bolovanje roditelja izdanu od strane pedijatra

- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2 i UC14
- **Opis osnovnog tijeka:**
  1. liječnik odabire opciju 'Preporuke za bolovanje' s početne stranice
  2. liječnik pregledava listu preporuka za bolovanje njegovim pacijentima
  3. liječnik odabire preporuku koju želi pregledati
  4. liječnik odabire opciju 'Odbij'
  5. preporuka je odbijena te se ne šalje doznaka za bolovanje poslodavcu pacijenta

#### UC19 - odjavi se iz sustava

- **Glavni sudionik:** korisnik
- **Cilj:** odjaviti se iz sustava
- **Sudionici:**
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. korisnik odabire opciju 'Odjava' s početne stranice
  2. sustav prikazuje modalni okvir za potvrdu odjave
  3. korisnik odabire opciju 'Odjavi se'
  4. sustav odjavljuje korisnika iz sustava i prikazuje početnu stranicu za prijavu
- **Opis mogućih odstupanja:**
  - 3.a korisnik odabire opciju 'Odustani'
    1. sustav zatvara modalni ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### UC20 - pregledaj korisnika

- **Glavni sudionik:** administrator
- **Cilj:** pregledati registrirane korisnike
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. administrator odabire opciju 'Korisnici' s početne stranice
  2. sustav prikazuje listu svih korisnika
  3. administrator odabire korisnika čije podatke želi pregledati

#### UC21 - promijeni prava pristupa

- **Glavni sudionik:** administrator
- **Cilj:** promijeniti razinu pristupa korisnika
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. administrator odabire opciju 'Pregled korisnika' s početne stranice
  2. sustav prikazuje listu svih ispravno registriranih korisnika
  3. administrator pronalazi željenog korisnika
  4. administrator mijenja razinu pristupa željenom korisniku odabirom opcije 'Promijeni razinu pristupa'
  5. sustav obavještava administratora da je razina pristupa uspješno promijenjena

#### UC22 - administriraj korisničke račune

- **Glavni sudionik:** administrator
- **Cilj:** upravljati registrima i korisničkim računima u sustavu
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC2
- **Opis osnovnog tijeka:**
  1. administrator odabire opciju 'Korisnici' s početne stranice
  2. administrator odabire opciju 'Novi korisnik'
  3. sustav prikazuje modalni okvir za stvaranje novog korisnika
  4. administrator unosi podatke o korisniku: OIB, ime, prezime, adresu
  5. administrator odabire ulogu korisnika (liječnik, pedijatar ili dijete)
  6. administrator odabire opciju 'Stvori korisnika'
  7. sustav provjerava unos i obavještava administratora da je registar uspješno stvoren
- **Opis mogućih odstupanja:**
  - 7.a neispravno uneseni podaci
    1. sustav utvrđuje da su podatci neispravni
    2. sustav usmjerava administratora na ponovno ispunjavanje podataka
    3. nakon ispravnog unosa, scenarij se nastavlja od koraka 4
  - 8.a administrator odabire opciju 'Odustani'
    1. sustav zatvara modalni ekran
    2. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### UC23 - poveži dijete s roditeljem



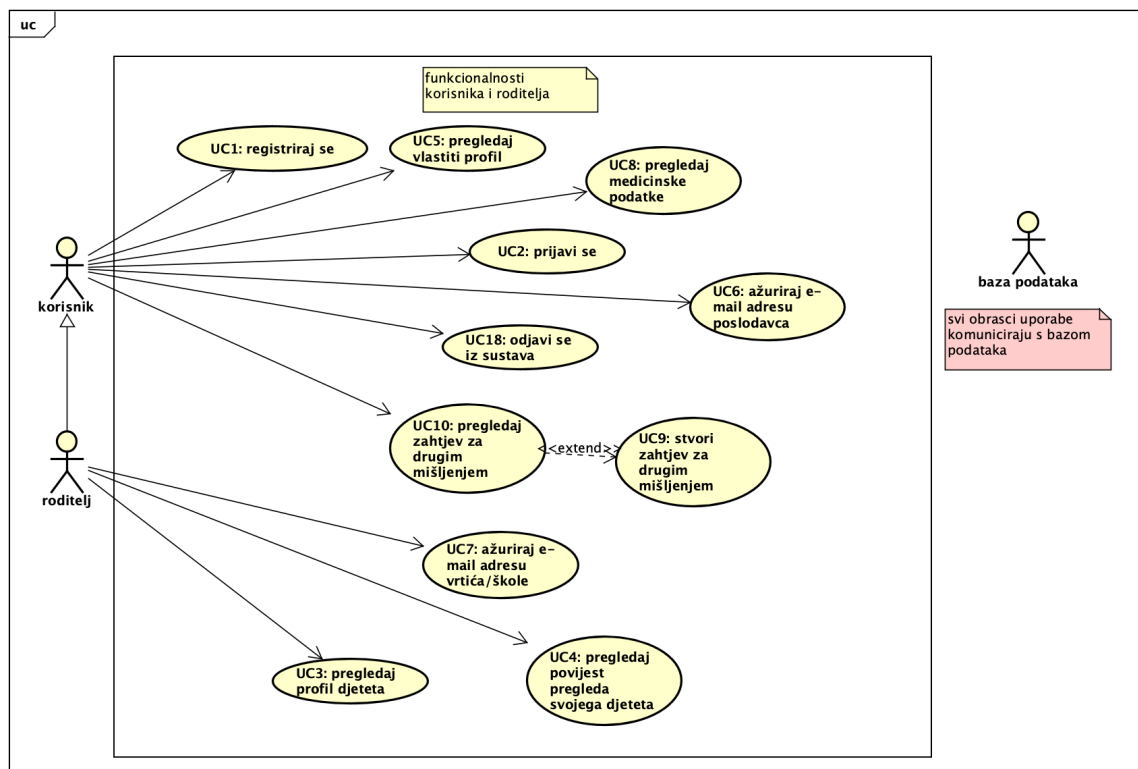
- **Glavni sudionik:** administrator
- **Cilj:** povezati profil djeteta s njegovim roditeljem u sustavu
- **Sudionici:** baza podataka
- **Preduvjet:** izvođenje UC1, UC2, UC22
- **Opis osnovnog tijeka:**
  1. administrator odabire opciju 'Korisnici' s početne stranice
  2. sustav prikazuje listu svih korisnika
  3. administrator filtrira listu svih korisnika tako da se prikazuju samo djeca
  4. administrator pronalazi željeno dijete
  5. administrator odabire profil djeteta s kojim želi povezati roditelja
  6. sustav prikazuje dropdown izbornik s popisom svih korisnika
  7. administrator odabire roditelja s kojim želi povezati dijete
  8. administrator odabire opciju 'Spremi'
  9. sustav obavještava administratora da je roditelj uspješno povezan s djetetom
- **Opis mogućih odstupanja:**
  - 8.a administrator odabire opciju 'Odustani'
    1. prikazuje se prethodni ekran i završava izvođenje ovog scenarija

#### UC24 - ažuriraj podatke korisnika aplikacije

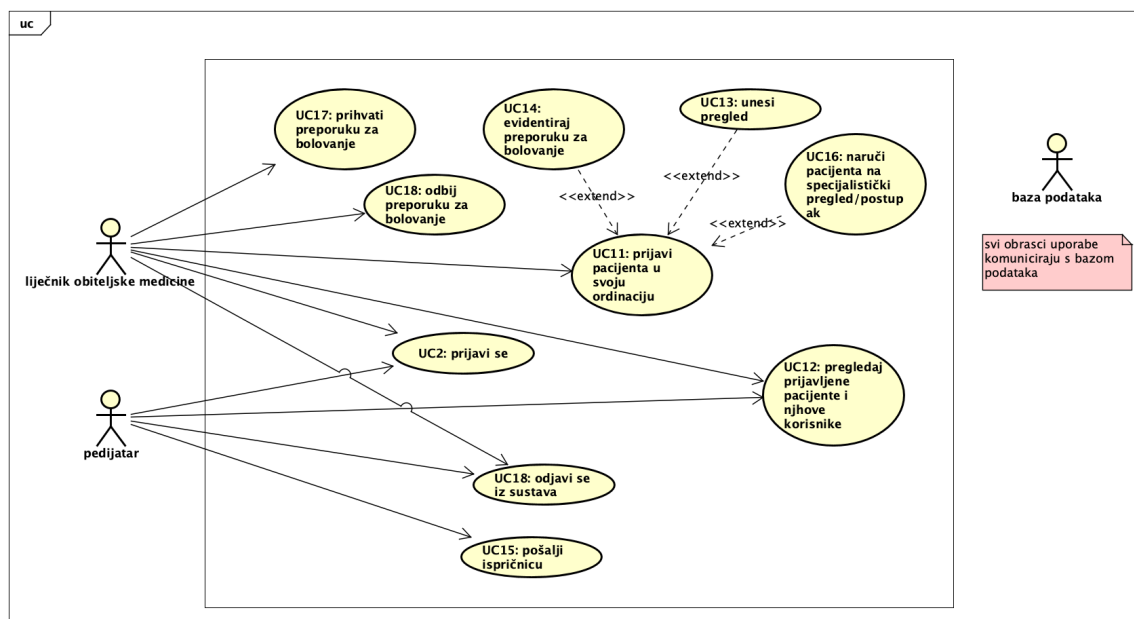
- **Glavni sudionik:** administrator
- **Cilj:** ažurirati sve podatke u aplikaciji, uključujući podatke o korisnicima i njihovim ovlastima
- **Sudionici:** baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
  1. administrator odabire opciju 'Korisnici' s početne stranice
  2. administrator pronalazi i odabire željenog korisnika
  3. sustav prikazuje profil odabranog korisnika
  4. administrator izmjenjuje osobne podatke o korisniku i njegove medicinske podatke
  5. administrator odabire opciju 'Spremi'
  6. sustav provjerava unos i obavještava administratora da su podaci uspješno ažurirani
- **Opis mogućih odstupanja:**
  - 6.a neispravno uneseni podaci

1. sustav utvrđuje da su podatci neispravni
2. sustav usmjerava administratora na ponovno ispunjavanje podataka
3. nakon ispravnog unosa, scenarij se nastavlja od koraka 4

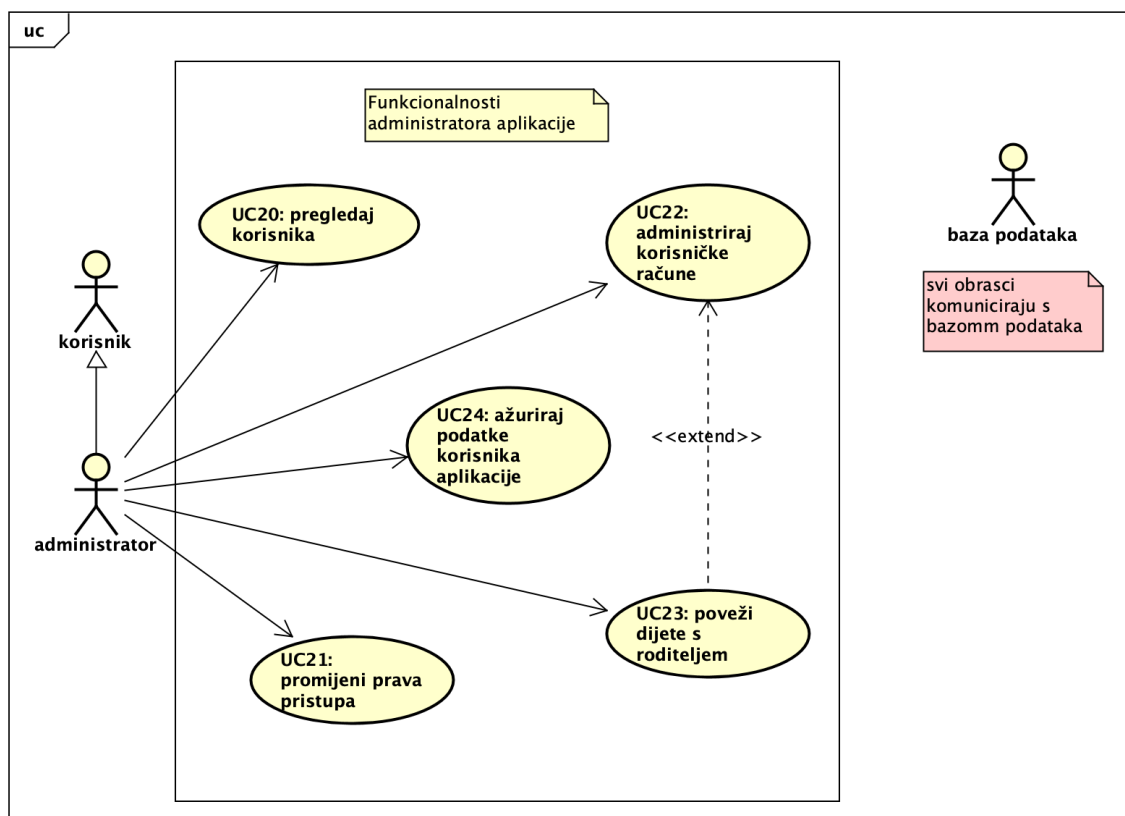
## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnosti korisnika i roditelja



Slika 3.2: Dijagram obrasca uporabe, funkcionalnosti liječnika obiteljske medicine i pedijatra

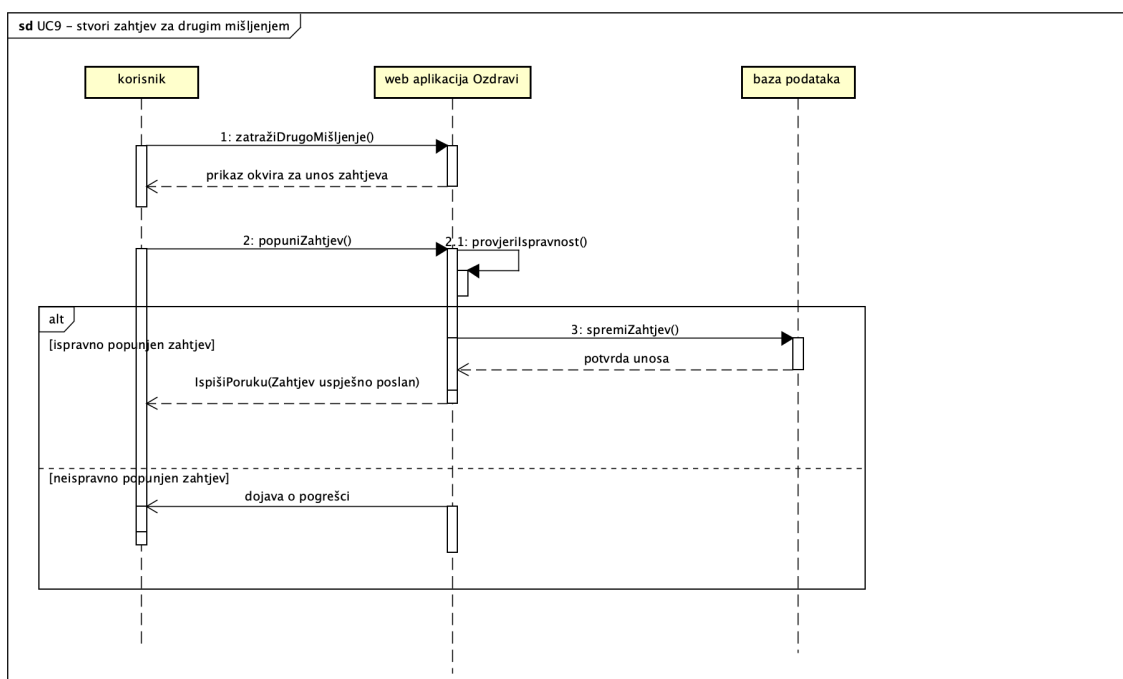


Slika 3.3: Dijagram obrasca uporabe, funkcionalnosti administratora

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC9 - stvori zahtjev za drugim mišljenjem

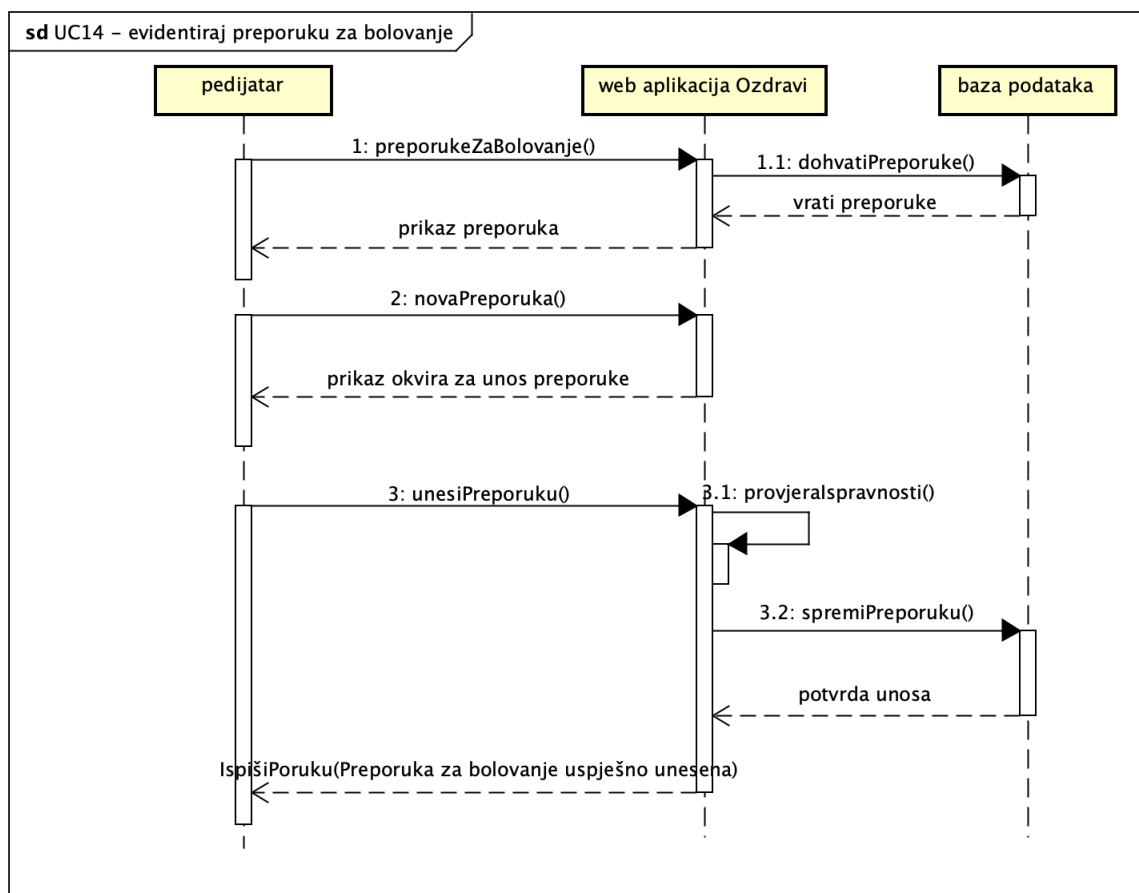
Kroz proces potraživanja drugog mišljenja, korisnik inicira postupak odabirom opcije 'Zatraži drugo mišljenje'. Sustav odgovara otvaranjem modalnog okvira za unos zahtjeva, gdje korisnik, u ovom slučaju roditelj, prenosi medicinski nalaz. Nakon toga, korisnik bira specifičnog liječnika ili pedijatra čije drugo mišljenje želi dobiti i potvrđuje zahtjev odabirom akcije 'Pošalji zahtjev'. Sustav potom provjerava i potvrđuje valjanost unosa, prikazujući korisniku poruku o uspješnom slanju zahtjeva. U slučaju neispravno popunjenog zahtjeva, sustav usmjerava korisnika na ponovni unos, osiguravajući točnost informacija prije nastavka postupka.



Slika 3.4: Sekvencijski dijagram za UC9

**Obrazac uporabe UC14 - evidentiraj preporuku za bolovanje**

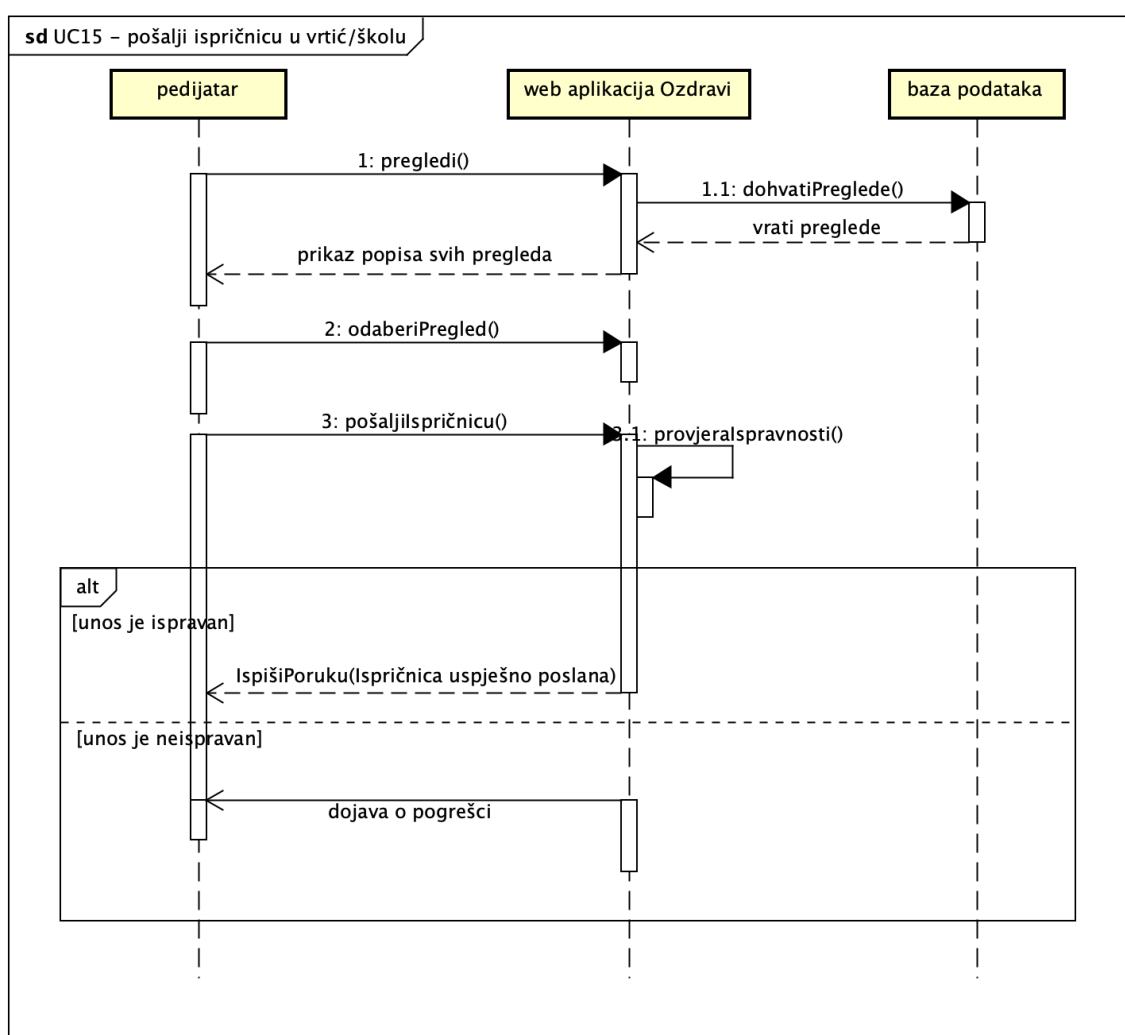
U postupku stvaranja nove preporuke za bolovanje, pedijatar započinje odabirom opcije 'Preporuke za bolovanje'. Zatim odabire opciju 'Nova preporuka za bolovanje'. Sustav reagira prikazivanjem modalnog okvira za unos preporuke za bolovanje. Pedijatar zatim odabire pregled na temelju kojeg je potrebna preporuka za bolovanje i unosi relevantne podatke o preporuci. Kada završi s unosom, pedijatar odabire opciju 'Unesi preporuku'. Sustav provjerava i potvrđuje uspješan unos, nakon čega obavještava pedijatra o uspješnom dodavanju preporuke.



Slika 3.5: Sekvencijski dijagram za UC14

**Obrazac uporabe UC15 - pošalji ispričnicu u vrtić/školu**

U procesu izdavanja nove ispričnice, pedijatar započinje odabirom opcije 'Pregledi' na početnoj stranici, što pokreće sustav da prikaže popis svih obavljenih pregleda. Pedijatar tada odabire pregled na temelju kojeg je potrebno poslati ispričnicu. Nakon što popuni potrebne podatke, pedijatar odabire opciju 'Pošalji ispričnicu'. Sustav izvršava provjeru i potvrđuje uspješan unos, nakon čega obavještava pedijatra da je ispričnica uspješno poslana. U slučaju neispravnog popunjavanja ispričnice, sustav prikazuje poruku o neispravnom unosu podataka, a pedijatar se vraća na ponovno popunjavanje ispričnice.



Slika 3.6: Sekvencijski dijagram za UC15

## 3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Sustav treba pružati brz odziv na korisničke zahtjeve
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, sučelje treba biti intuitivno te se korisnici moraju znati koristiti sučeljem bez opširnih uputa
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS protokola
- Svi osjetljivi korisnički podaci, uključujući medicinske informacije, moraju biti kriptirani u prijenosu i pohranjeni na način koji udovoljava standardima sigurnosti podataka u zdravstvenim aplikacijama
- Redovito automatsko izvođenje sigurnosnih kopija podataka s brzim i pouzdanim sustavom oporavka u slučaju neplaniranih događaja ili gubitka podataka



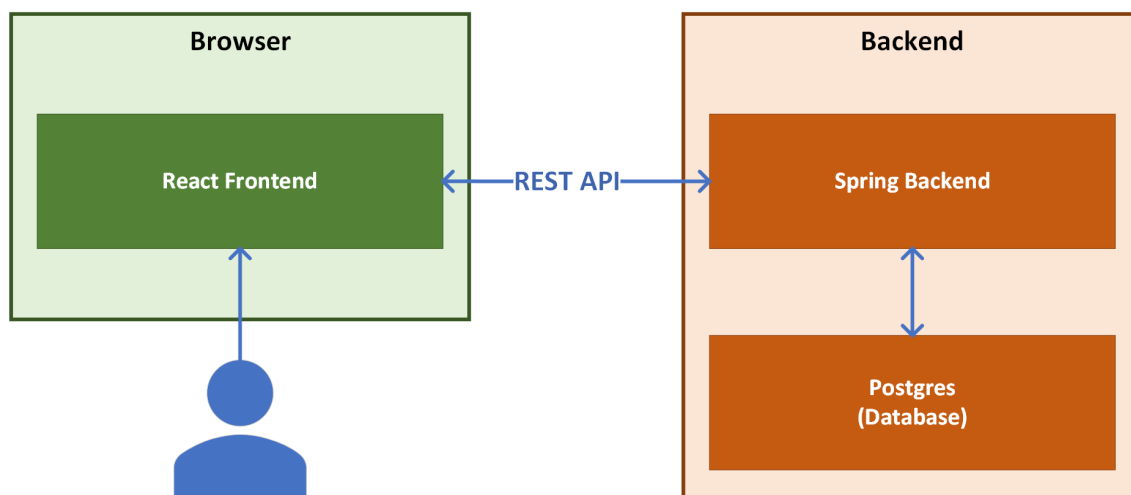
## 4. Arhitektura i dizajn sustava

Arhitektura sustava počiva na podjeli na korisničko sučelje (*en. frontend*) i na pozadinsku aplikaciju (*en. backend*) koja se spaja na relacijsku bazu podataka.

Glavna tehnologija izrade frontend dijela sustava je React.js razvojni okvir, a samo korisničko sučelje je izgrađeno od gotovih komponenti iz paketa "Bootstrap". Korištenje Bootstrapa omogućuje brzo i lako razvijanje sučelja koja izgledaju moderno i koja su responzivna, što je jedan od važnijih zahtjeva.

Backend dio je izrađen u Spring radnom okviru koji omogućava razvoj MVC (Model-View-Controller) aplikacija, ali i još puno toga, poput rada s bazama podataka i autorizacije korisnika (Spring Security). Za postavljanje aplikacije u skladu sa Springovim konvencijama koristili smo Spring Boot ekstenziju.

Kao relacijsku bazu podataka odabrali smo Postgres. Radi se o bazi otvorenog koda (*en. open-source*), visokih performansi koja se koristi za širok raspon zadataka i u raznim okruženjima.



Slika 4.1: Dijagram arhitekture sustava "Ozdravi".

Kao što je prikazano na slici 4.1, središnji je dio sustava Spring backend koji ostvaruje komunikaciju s Postgres bazom, ali i s frontend aplikacijom. Komunikacija s frontendom se ostvaruje primjenom HTTP (HyperText Transfer Protokola) proto-

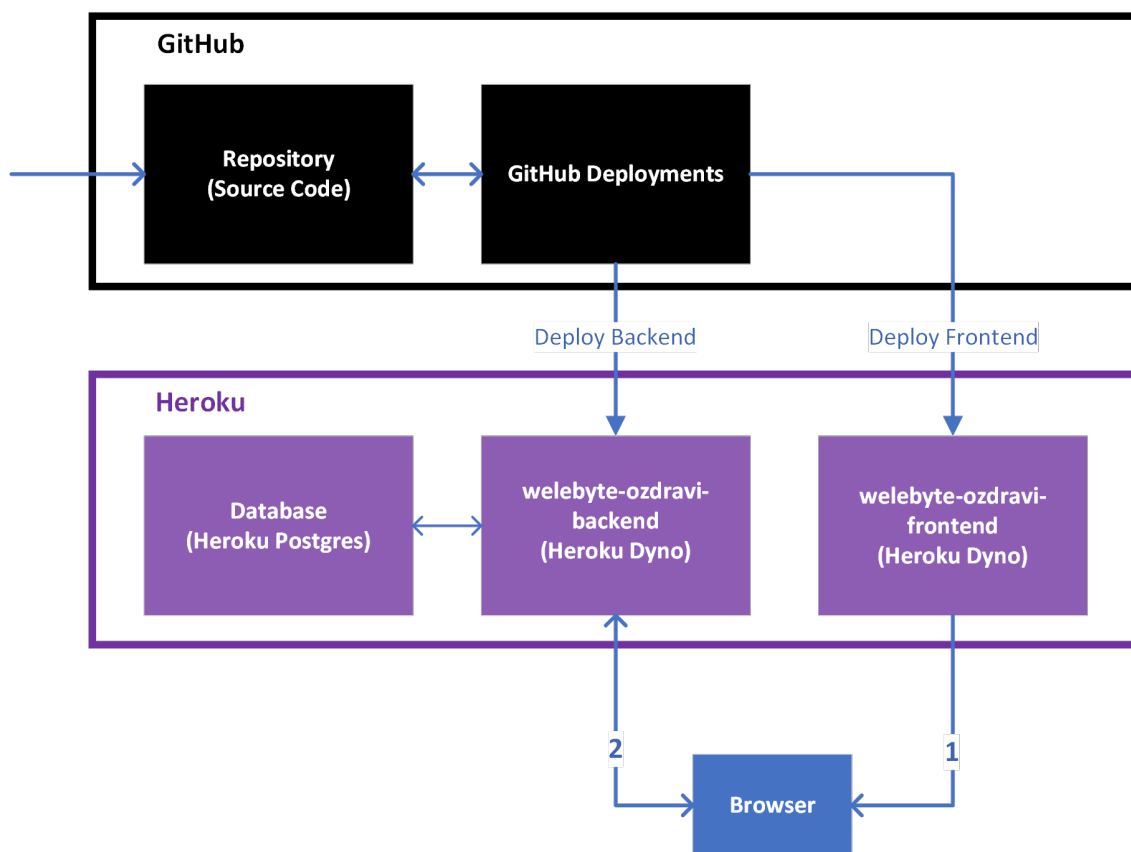
kola koji je ključni protokol web-a. Za potrebe komunikacije backenda i frontenda definiran je API (Application Programming Interface) u skladu s pravilima REST-a (Representational State Transfer). Neka od njih su:

- odvajanje poslužitelja i klijenta - klijent i poslužitelj su u potpunosti neovisni, a sve što klijent "zna" dolazi od poslužitelja
- "statelessness" - zahtjevi klijenta prema poslužitelju su međusobno neovisni i sadrže sve informacije potrebne za svoje izvođenje

U kontekstu Spring razvojnog okvira, aplikacija koristi tri arhitekturna sloja (Controller, Service i Repository) kako bi se organizirao kod i razdvojile odgovornosti:

- Controller - Sloj koji obrađuje HTTP zahtjeve i upravlja komunikacijom između korisničkog sučelja (frontend) i ostatka aplikacije. Kontroleri su odgovorni za prihvatanje HTTP zahtjeva, obradu korisničkih ulaznih podataka, pozivanje odgovarajućih metoda u Service sloju i pripremu odgovora za korisnika.
- Service - Sloj koji sadrži poslovnu logiku aplikacije. Servisi se koriste kako bi se odvojila poslovna logika od kontrolera i omogućila ponovna upotreba iste logike na različitim dijelovima aplikacije.
- Repository - Sloj koji je odgovoran za pristup podacima. To uključuje komunikaciju s bazom podataka. Repository sloj omogućava izvršavanje osnovnih operacija poput spremanja, dohvaćanja, ažuriranja i brisanja podataka.

Zahtjevi prema backendu dolaze iz React aplikacije koja se izvršava unutar korisničkog preglednika, a ne na nekom vanjskom poslužitelju. Ipak, kako bi ju pokrenuo, korisnički preglednik tu aplikaciju mora preuzeti od nekuda. To je prikazano na slici 4.2.



Slika 4.2: Dijagram infrastrukture sustava "Ozdravi".

U produkcijskom okruženju, sustav koristi dva poslužitelja - jedan pokreće Spring aplikaciju (backend), a drugi omogućava preglednicima preuzimanje React aplikacije (frontend).

Kao infrastrukturnu platformu, na kojoj ćemo postaviti produkcijsko okruženje, izabrali smo Heroku zbog ova tri razloga:

- jednostavnost postavljanja potrebnih resursa (Heroku Dyno, Heroku Postgres)
- fleksibilan model naplate koji nam omogućuje besplatno korištenje platforme
- jednostavno integriranje s GitHub repozitorijem za lagan "deployment" kad je to potrebno

Korisnici se u sustav prijavljuju unoseći svoje korisničko ime i lozinku. Na temelju tih vjerodajnica, a nakon provjere s podacima u bazi, backend korisniku izdaje token koji se koristi za autentikaciju narednih zahtjeva s klijenta. Koristimo "Bearer" autentikacijsku shemu koja podrazumijeva da svaki HTTP zahtjev koji

dolazi od klijenta ima token u ispravnom polju u zaglavlju. odvija putem HTTPS protokola, osiguravajući sigurnu komunikaciju između klijenta i poslužitelja putem TLS/SSL enkripcije. Ovakav pristup pridonosi zaštiti osjetljivih podataka i održava sigurnost komunikacije u sustavu.

## 4.1 Baza podataka

Baza podataka koja se koristi u projektu "Ozdravi" ključna je komponenta sustava koja omogućava pohranu, upravljanje i praćenje svih relevantnih informacija o korisnicima i njihovim zdravstvenim podacima. Izgrađena je relacijska baza podataka koja se sastoji od sljedećih entiteta:

- User
- Examination
- SecondOpinion
- Instruction
- SickLeaveRecommendation
- Address
- Role
- UserRoles

Baza podataka "Ozdravi" omogućava integraciju svih informacija kako bi podržala procese komunikacije, pregleda i dijagnoza u kontekstu brige o zdravstvenim potrebama djece. Ova struktura omogućava učinkovito praćenje i upravljanje svim relevantnim medicinskim podacima i podržava glavne funkcionalnosti sustava.

### 4.1.1 Opis tablica

**Korisnik** Ovaj entitet sadržava osnovne informacije o svim korisnicima sustava, uključujući njihovu ulogu, identifikacijske brojeve, ime, prezime, e-mail i lozinku te e-mail korisnikove institucije gdje je potrebno slati ispričnicu ili doznaku za bolovanje. Ova tablica je ključna za autentikaciju korisnika i definira njihove uloge unutar sustava.

User (korisnik)		
id	INT	jedinstveni brojčani identifikator
parent_id	INT	ID osobe koja je roditelj osobe [Parent]
doctor_id	INT	ID osobe koja je doktor osobe [Doctor/Pediatrician]
address_id	INT	ID Adrese osobe
OIB	VARCHAR	Osobni identifikacijski broj
first_name	VARCHAR	Ime osobe (NOT NULL)
last_name	VARCHAR	Prezime osobe (NOT NULL)
role	VARCHAR	Uloga korisnika (NOT NULL)
password	VARCHAR	Lozinka za prijavu u sustav
email	VARCHAR	e-mail za prijavu u sustav
institution_email	VARCHAR	službeni e-mail ustanove (vrtić/škola/poslodovac)

**Pregled** Ovaj entitet sadržava podatke o pacijentima, uključujući informacije o njihovim pregledima, dijagnozama i zapisnicima. Sadrži attribute: identifikator pregleda, identifikator pacijenta, identifikator liječnika, identifikator liječnika koji je zakazao pregled, identifikator adrese na kojoj se održava pregled, zapisnik s pregleda te datum i vrijeme pregleda. Pohranjuje veze između pacijenata, doktora i pregleda, omogućavajući praćenje medicinskih podataka.

Examination (Pregled)		
id	INT	jedinstveni brojčani identifikator
patient_id	INT	ID osobe koja je pacijent [Parent/Child] (NOT NULL)
doctor_id	INT	ID osobe koja vrši pregled [Doctor/Pediatrician] (NOT NULL)
scheduler_id	INT	ID osobe koja je zakazala pregled [Doctor/Pediatrician] (NOT NULL)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Examination (Pregled)		
address_id	INT	ID adrese na kojoj se održava pregled
report	TEXT	Dijagnoza/zapisnik s pregleda
date	DATETIME	Datum i vrijeme pregleda (NOT NULL)

**Drugo mišljenje** Ovaj entitet služi za praćenje detalja o pregledima pacijenata. Sadrži attribute: identifikator drugog mišljenja, identifikator osobe koja ga je zatražila, identifikator liječnika koji daje drugo mišljenje te nalaz iz privatne ustanove na temelju kojeg se traži drugo mišljenje. Ovaj entitet omogućava praćenje medicinskih konzultacija.

SecondOpinion (DrugoMišljenje)		
id	INT	jedinstveni brožčani identifikator
requester_id	INT	ID osobe koja je zatražila drugo mišljenje [Parent] (NOT NULL)
doctor_id	INT	ID osobe koja je dala drugo mišljenje [Doctor/Pediatrician] (NOT NULL)
content	TEXT	nalaz iz privatne ustanove koji unosi roditelj (NOT NULL)
opinion	TEXT	mišljenje koje liječnik/pedijatar daje o nalazu iz privatne ustanove

**Uputa** Ovaj entitet sadržava informacije o izdanim uputama, uključujući informacije o pacijentima, doktorima koji stvaraju upute i statusu upute. Sadrži attribute: identifikator upute, identifikator liječnika koji je izdao uputu, identifikator pacijenta kojemu je upućena, datum te sadržaj upute. Omogućava praćenje i upravljanje medicinskim preporukama.

Instruction (Uputa)		
id	INT	jedinstveni brožčani identifikator

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Instruction (Uputa)		
doctor_id	INT	ID osobe koja je izdala uputu [Doctor/Pediatrician] (NOT NULL)
patient_id	INT	ID osobe za koju je uputa [Parent] (NOT NULL)
date	DATETIME	Datum i vrijeme izdavanja upute (NOT NULL)
content	TEXT	Sadržaj upute (NOT NULL)

**Preporuka za bolovanje** Ovaj entitet sadržava informacije o izdanim preporukama za bolovanje, tko ih je izdao, za koga i tko ih treba odobriti. Sadrži attribute: identifikator preporuke, identifikator pacijenta, identifikator pedijatra koji ju je stvorio, identifikatora liječnika koji ju je odobrio, identifikator pregleda na kojem se izdaje te status preporuke.

SickLeaveRecommendation (Preporuka za bolovanje)		
id	INT	jedinstveni brožčani identifikator
patient_id	INT	ID osobe za koju se traži bolovanje [Parent] (NOT NULL)
creator_id	INT	ID osobe koja je stvorila preporuku [Pediatrician] (NOT NULL)
approver_id	INT	ID osobe koja mora odobriti preporuku [Doctor] (NOT NULL)
examination_id	INT	ID pregleda na temelju kojeg se izdaje preporuka (NOT NULL)
status	BOOLEAN	Je li preporuka odobrena? (default false)

**Adresa** Ovaj entitet sadržava informacije o adresi održavanja pregleda. Sadrži attribute: identifikator ulice, ulicu, kućni broj, grad, državu te geografsku širinu i dužinu.

Address (Adresa)		
id	INT	jedinstveni brojčani identifikator
street	VARCHAR	ulica
number	VARCHAR	kućni broj
city	VARCHAR	grad
country	VARCHAR	država
latitude	NUMERIC	geografska širina
longitude	NUMERIC	geografska dužina

**Uloga** Ovaj entitet sadržava informacije o mogućim ulogama u sustavu. Sadrži attribute: identifikator uloge i naziv uloge.

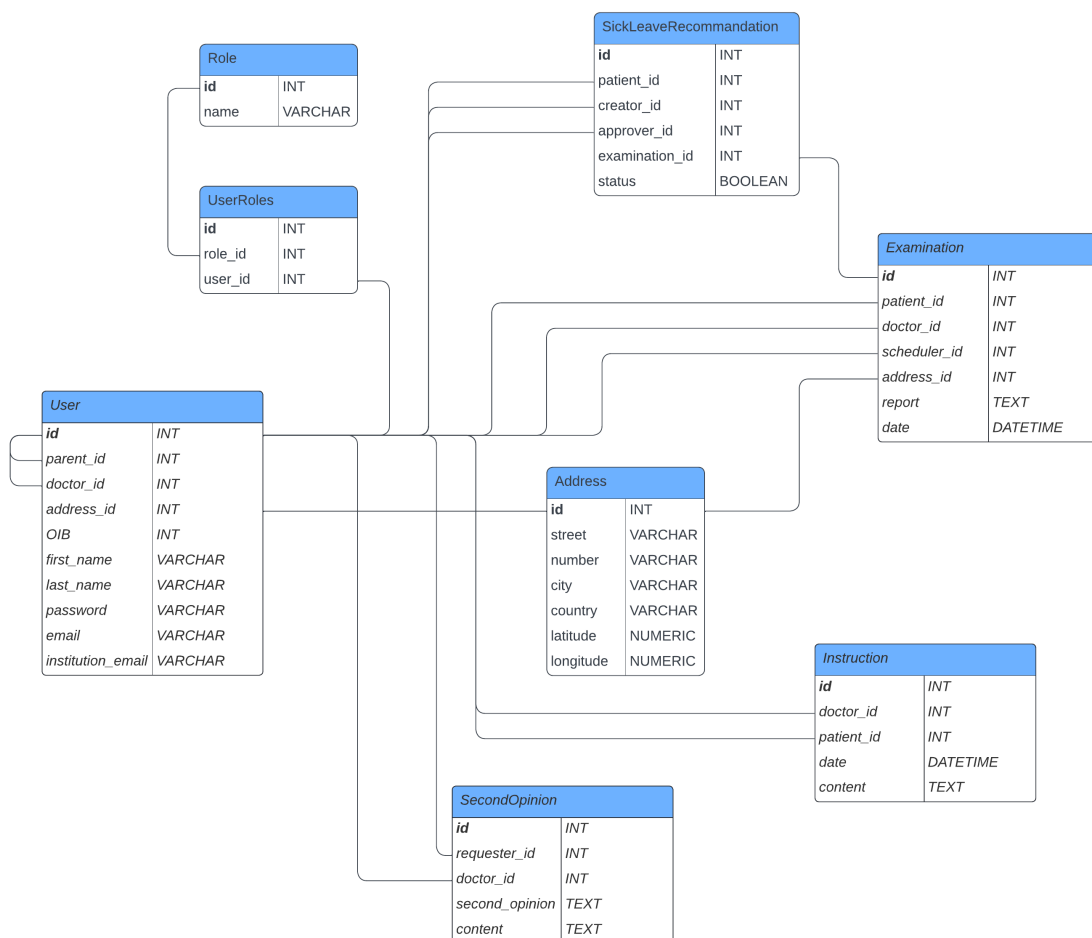
Role (Uloga)		
id	INT	jedinstveni brojčani identifikator
name	VARCHAR	naziv uloge

**UserRoles** Ovaj entitet sadržava informacije o ulogama korisnika u sustavu. Prikazuje koji korisnik ima koju ulogu. Sadrži attribute: identifikator, identifikator uloge i naziv uloge.

UserRoles		
id	INT	jedinstveni brojčani identifikator
role_id	VARCHAR	identifikator uloge
user_id	VARCHAR	identifikator osobe

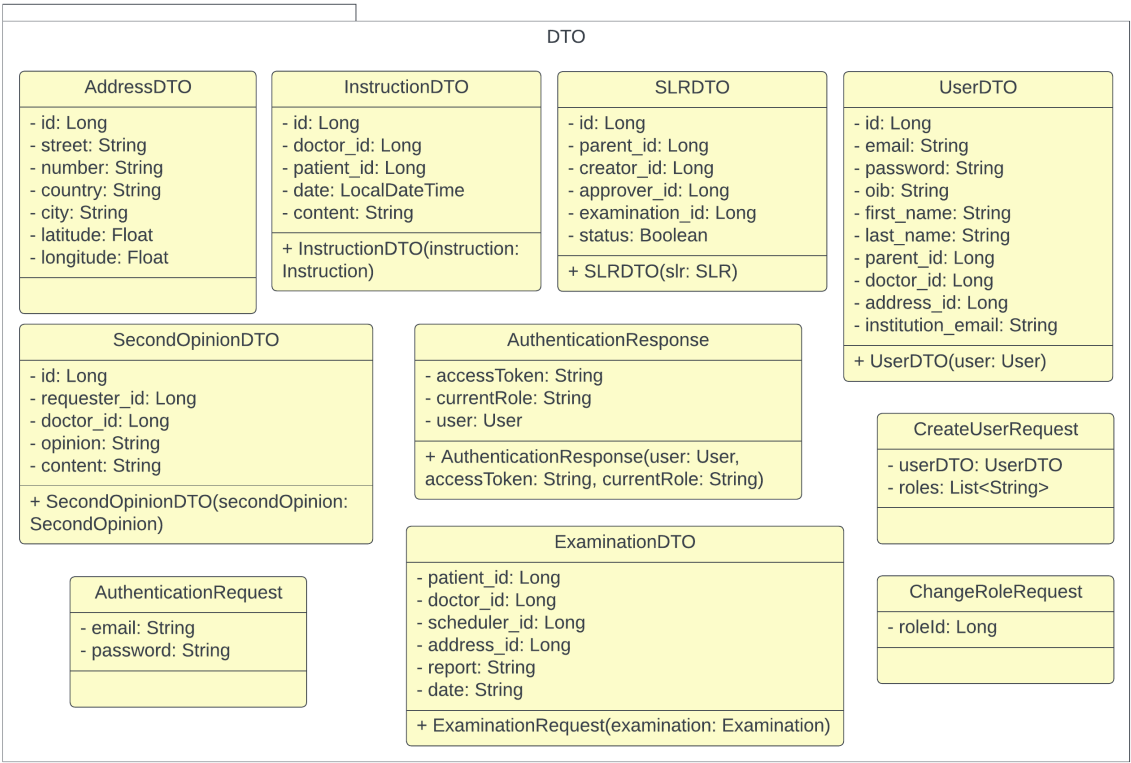


## 4.1.2 Dijagram baze podataka



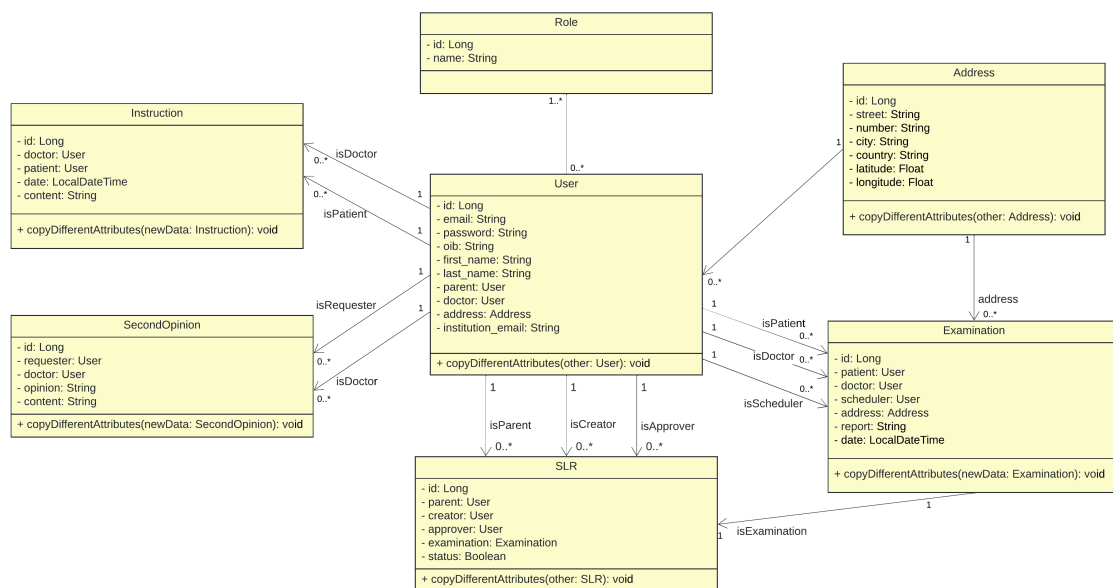
Slika 4.3: ER dijagram baze podataka





Slika 4.5: Dijagram razreda - dio Data transfer objects

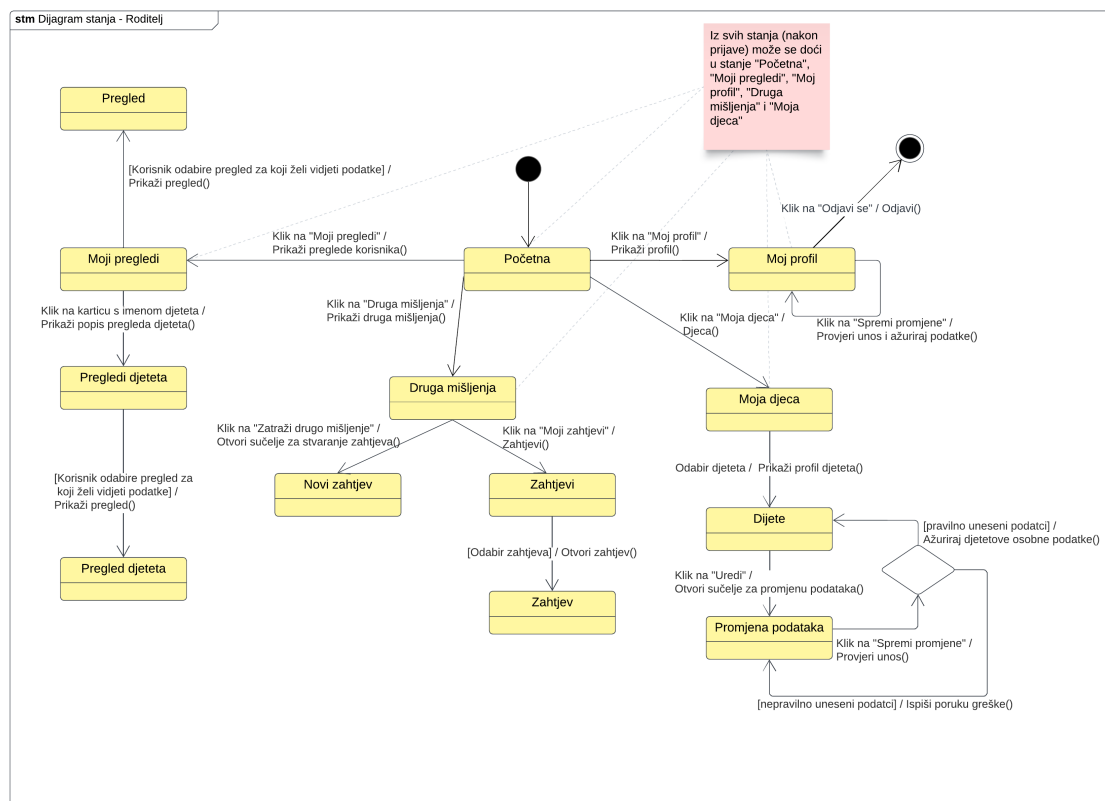
Domain razredi preslikavaju strukturu baze podataka u aplikaciji. Razred User predstavlja prijavljenog korisnika. Razred Address predstavlja adresu koja može biti povezana s pregledom ili sa samim korisnikom. Razred Examination sadrži podatke o pregledu pacijenta koje unosi korisnik(s ulogom liječnika). Razred SLR sadrži podatke vezane uz potvrdu za bolovanje koju pedijatar zatraži, a liječnik ju može odobriti. Razred Instruction predstavlja uputu koju izdaje liječnik, a prihvaća pacijent.



Slika 4.6: Dijagram razreda - domain razredi

## 4.3 Dijagram stanja

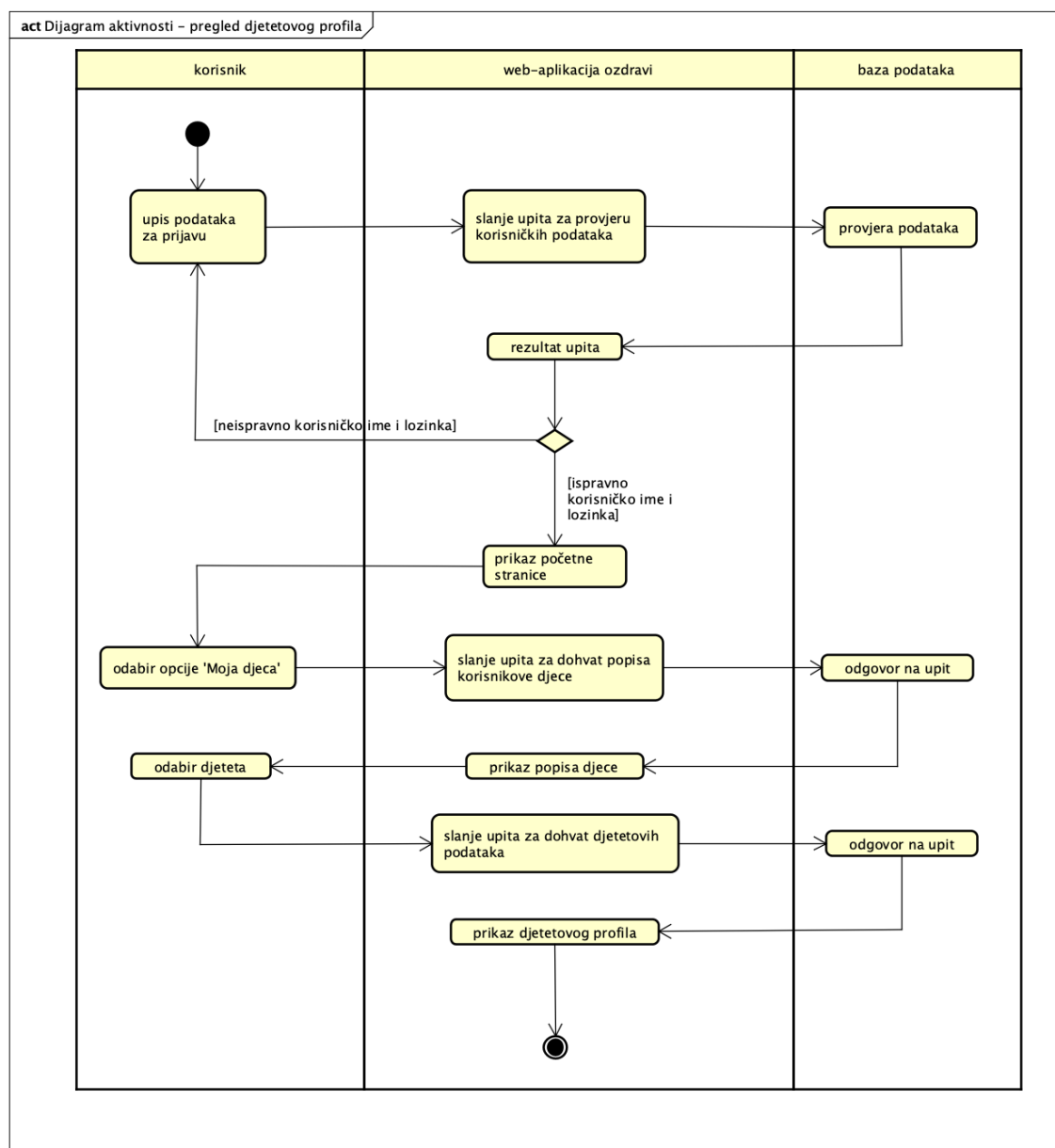
UML dijagram stanja je vrsta dijagrama koja prikazuje različita stanja kroz koje objekt ili sustav prolazi tokom svog životnog ciklusa. Na slici 4.6 prikazan je dijagram stanja za registriranog korisnika koji ima ulogu roditelja. Nakon što se korisnik prijavi, prikazuje mu se početna stranica. Korisnik u Izborniku može odabrati neku od ponuđenih opcija. Klikom na „Moj profil“ prikazuju se podatci korisnika koje može urediti. Također, korisnik ima mogućnost odjave klikom na „Odjavi se“. Odabirom „Moja djeca“ prikazuje se lista korisnikove djece, a klikom na neko od ponuđenih imena djece otvara se profil odabranog djeteta. Klikom na „Uredi“ moguće je izmijeniti djetetove podatke. U izborniku, klikom na „ Druga mišljenja“ moguće je stvoriti zahtjev za novim mišljenjem klikom na „Zatraži drugo mišljenje“ ili pregledati listu zahtjeva klikom na „Moji zahtjevi“ te vidjeti detalje zahtjeva odabirom željenog zahtjeva. Odabirom „Moji pregledi“ otvaramo povijest svih pregleda korisnika, klikom na određeni pregled prikazuju se detalji istog. Klikom na karticu s imenom djeteta, moguće je vidjeti povijest pregleda odabranog djeteta te kao i kod osobnih pregleda vidjeti detalje pregleda odabirom specifičnog pregleda.



Slika 4.7: Dijagram stanja

## 4.4 Dijagram aktivnosti

Dijagram aktivnosti u okviru projekta koristi se kako bi opisao tokove upravljanja ili podataka u sustavu. Za razliku od modeliranja događajima poticanog ponašanja, dijagram aktivnosti fokusira se na prikaz toka upravljanja, gdje svaki sljedeći korak slijedi nakon završenog prethodnog. Osnovna načela modeliranja dijagrama aktivnosti uključuju jednostavnost i jasnoću, čime olakšava razumijevanje procesa unutar sustava. Na dijagramu aktivnosti 4.7 prikazana je aktivnost pregledavanja djetetovog profila. Aktivnost započinje korisnikovom prijavom u sustav, nakon koje korisnik odabire dijete čiji profil želi pregledati. Nakon odabira, sustav prikazuje profil odabranog djeteta te korisnik ima mogućnost pregledati povijest pregleda. Odabirom opcije 'Povijest pregleda', sustav nudi popis svih djetetovih pregleda. Korisnik može odabrati pregled čiju povijest želi pregledati, nakon čega sustav prikazuje pojedinosti odabranog pregleda.

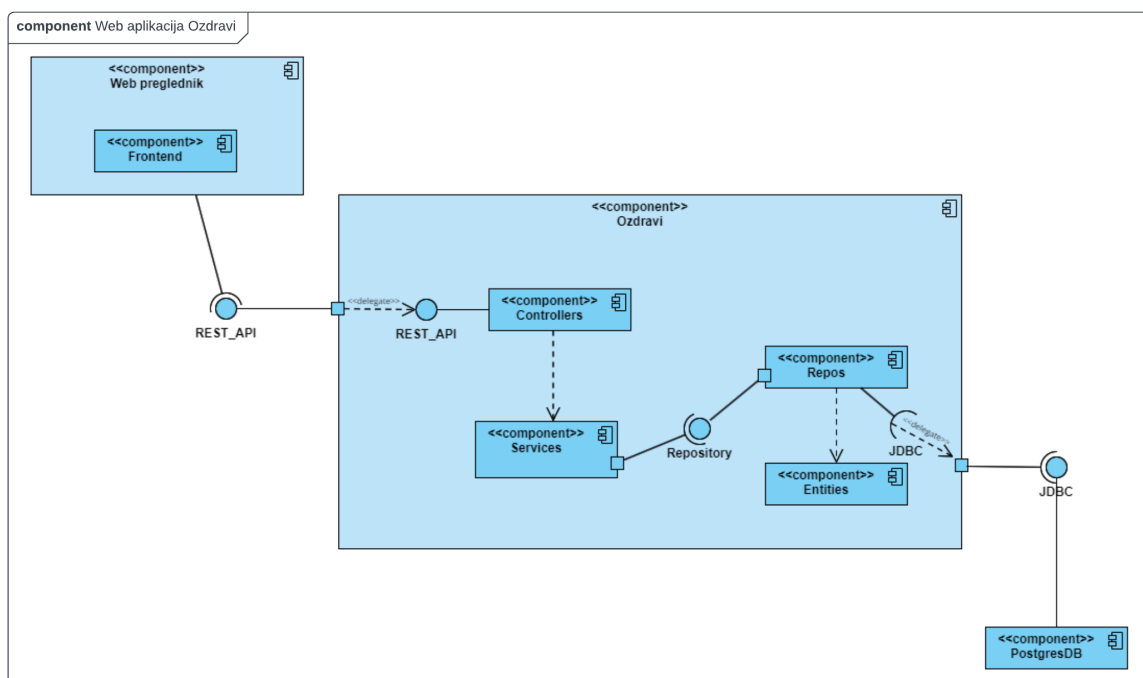


Slika 4.8: Dijagram aktivnosti



## 4.5 Dijagram komponenti

UML dijagrami komponenata pružaju vizualni prikaz modularne strukture i međusobnih odnosa između komponenata u programskoj potpori. Na slici 4.8. prikazan je dijagram komponenti web-aplikacije Ozdravi. Sastoji se od 4 komponente: Controllers, Services, Repos i Entities. Controllers pružaju REST\_API sučelje na koje vanjski web-preglednik može slati zahtjeve. Komponenta Controllers je zadužena za obradu HTTPS zahtjeva na temelju kojih poziva operacije komponente Services. Komponenta Services za izvođenje pojedinih operacija treba pristup podacima iz baze, za što koristi operacije komponente Repos koja ostvaruje sučelje JDBC. Repos upisuju podatke iz baze podataka u komponentu Entities.



Slika 4.9: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

U tijeku razvoja projekta "Ozdravi", pažljivo su odabrane i implementirane različite tehnologije i alati kako bi se postigla efikasna izrada dokumentacije i aplikacije za zdravstveni sustav. Za unapređenje timskih komunikacija korištene su aplikacije *WhatsApp*<sup>1</sup> i *Microsoft Teams*<sup>2</sup>. WhatsApp, popularna platforma za brzu razmjenu poruka, olakšala je neposrednu interakciju među članovima tima, dok je Microsoft Teams pružio šire mogućnosti za suradnju, uključujući organizaciju sastanaka, održavanje sastanaka i dijeljenje datoteka. U praćenju zadataka i upravljanju projektom koristila se *Jira*<sup>3</sup>, snažan alat koji je omogućio stvaranje strukturirane i pregledne radne okoline. Jira je poslužila kao središnje mjesto za praćenje i planiranje aktivnosti, olakšavajući koordinaciju među članovima tima. *GitHub*<sup>4</sup> se istaknuo kao ključni alat za upravljanje izvornim kodom i omogućavanje učinkovite suradnje među članovima tima. GitHub, kao web-bazirana platforma, pružio je mogućnost distribuiranog upravljanja verzijama korištenjem Git sustava. Repozi torij na GitHubu služio je kao centralno mjesto za čitav izvorni kod, omogućavajući članovima tima sinkroniziranje svojih lokalnih kopija koda, praćenje promjena te učinkovito upravljanje raznim granama razvoja. *Astah*<sup>5</sup> je korišten za izradu UML dijagrama, pružajući jasan prikaz strukture sustava i odnosa između njegovih dijelova. Osim Astaha, korišteni su i drugi alati za izradu dijagrama, uključujući *Visual Paradigm*<sup>6</sup> i *Lucidchart*<sup>7</sup>. Za razvoj frontend dijela aplikacije korišten je *React*<sup>8</sup>, jedna od najpopularnijih *JavaScript*<sup>9</sup> biblioteka koja omogućava dinamičko i responzivno korisničko sučelje. React je pridonosio brzom izgradnji modernog sučelja

---

<sup>1</sup><https://www.whatsapp.com>

<sup>2</sup><https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>

<sup>3</sup><https://www.atlassian.com/software/jira>

<sup>4</sup><https://github.com>

<sup>5</sup><http://astah.net/editions/professional>

<sup>6</sup><https://online.visual-paradigm.com>

<sup>7</sup><https://www.lucidchart.com>

<sup>8</sup><https://reactjs.org/>

<sup>9</sup><https://www.javascript.com/>

s visokom interaktivnošću. Korišten je i *Bootstrap*<sup>10</sup>, popularni okvir za izradu responzivnih web stranica, koji je omogućio brzo i jednostavno oblikovanje sučelja te moderan i funkcionalan dizajn. Bootstrap pruža jednostavne i učinkovite klase za organizaciju elemenata, responzivno oblikovanje pomoću tzv. grid sustava te unaprijed definirane stilove za tipografiju, forme, gumbiće i mnoge druge komponente. Za backend je odabran *Spring*<sup>11</sup>, robustan okvir za razvoj *Java*<sup>12</sup> aplikacija. Spring je pružio strukturu i podršku za izradu pouzdanih, sigurnih i skalabilnih backend rješenja. API sustav je strukturiran i dokumentiran korištenjem alata *Swagger*<sup>13</sup>, pri čemu je implementiran sukladno OpenAPI konvencijama. OpenAPI pristup omogućava jednostavno definiranje, dokumentiranje i testiranje RESTful API-ja te definiranje svih dostupnih endpointa, parametara, odgovora i modela podataka. U procesu pisanja koda korišteni su *IntelliJ IDEA*<sup>14</sup> i *Visual Studio Code*<sup>15</sup>. IntelliJ IDEA, s bogatim skupom značajki, podržavao je razvoj Java aplikacija, dok je Visual Studio Code, lagan i svestran, omogućavao efikasno pisanje koda u različitim jezicima, uključujući *JavaScript*<sup>16</sup>. Odabir ovih alata bio je prepušten osobnim preferencijama članova tima. Za postavljanje produkcijskog okruženja odabran je Heroku kao infrastrukturna platforma. *Heroku*<sup>17</sup> pruža stabilnost i skalabilnost, čineći implementaciju i održavanje aplikacije jednostavnim i učinkovitim. Baza podataka *PostgreSQL*<sup>18</sup> koristila se za trajno pohranjivanje podataka. PostgreSQL je open-source sustav za upravljanje bazama podataka koji pruža pouzdano i sigurno okruženje za pohranu informacija. U konačnici, ovom kombinacijom tehnologija postignuto je stvaranje moderne zdravstvene aplikacije koja je intuitivna za korisnike, prilagodljiva potrebama zdravstvenog sustava i unapređuje ukupno iskustvo krajnjih korisnika.

---

<sup>10</sup><https://getbootstrap.com>

<sup>11</sup><https://spring.io>

<sup>12</sup><https://www.java.com/en/>

<sup>13</sup><https://swagger.io>

<sup>14</sup><https://www.jetbrains.com/idea/>

<sup>15</sup><https://visualstudio.microsoft.com/>

<sup>16</sup><https://www.javascript.com/>

<sup>17</sup><https://www.heroku.com>

<sup>18</sup><https://www.postgresql.org>

## 5.2 Ispitivanje programskog rješenja

Ispitivanje programskog rješenja predstavlja ključan korak u procesu razvoja programske podrške s ciljem ostvarivanja nekoliko ključnih prednosti:

- olakšava dodavanje značajki - ispitivanje osigurava da nove promjene ne narušavaju funkcionalnost postojećeg koda
- sprječava unošenje grešaka u kod - kvalitetni testovi detektiraju neispravan kod prije nego što postane dio projekta čime se smanjuje mogućnost pojave grešaka

U sklopu ovog projekta koristili smo dvije vrste ispitivanja:

- Ispitivanje komponenti (eng. Unit Testing)
- Ispitivanje sustava (eng. Integration Testing)

Ispitivanjem komponenti koristeći JUnit biblioteke osigurali smo ispravan rad ključnih dijelovi backenda. Ispitivanjem sustava provjerili smo ispravnost povezanosti backenda i frontenda iz perspektive krajnjeg korisnika. Ispitivanje sustava je provedeno koristeći Selenium biblioteke.

### 5.2.1 Ispitivanje komponenti

#### Test 1: Ispitivanje ispravnog OIB-a

Komponenta: ValidityUtil

Metoda: public static boolean isValidOib(String oib);

Ulazni Podaci:

- String oib : "39751670659"

Očekivani rezultat:

- Metoda vraća istinu (true) jer je OIB ispravan prema standardu za OIB-e.

```
1 @Test
2 @DisplayName("Good oib test")
3 void shouldReturnTrue() {
4     assertTrue(ValidityUtil.isValidOib("39751670659"));
5 }
```

Listing 5.1: Test 1

## Test 2: Ispitivanje neispravnog OIB-a

Komponenta: ValidityUtil

Metoda: public static boolean isValidOib(String oib);

Ulazni Podaci:

- String oib : "397516706.9"

Očekivani rezultat:

- Metoda vraća laž (false) jer OIB sadrži '.', odnosno ne zadovoljava standardni oblik.

```
1 @Test
2 @DisplayName("Bad oib test")
3 void shouldReturnFalse() {
4     assertFalse(ValidityUtil.isValidOib("397516706.9"));
5 }
```

Listing 5.2: Test 2

U testovima namijenjenima testiranju kontrolera, želimo ispitati isključivo ponašanje kontrolera, bez ispitivanja servisa, baze i drugih dijelova aplikacije. Budući da kontoleri ne mogu raditi bez tih dijelova, nužno je stvoriti "mock" objekte koji ih mijenjaju i simuliraju njihovo ponašanje.

Ovaj postupak uključuje ručno stvaranje entiteta koji predstavljaju očekivane rezultate servisa.

## Test 3: Dohvaćanje popisa Uputa.

Komponenta: InstructionsController

Zahtjev: HTTP GET /instructions

Mock Podaci:

- Korisnik
- Pregledi kojima korisnik može pristupiti

Očekivani rezultat:

- Kontroler u JSON formatu vraća popis svih Uputa koje trenutni korisnik može vidjeti, HTTP odgovor ima status 200 (OK).

```
1 @Test
2 @WithMockUser
3 void shouldReturnAllInstructions() throws Exception {
4     when(instructionService.list()).thenReturn(allInstructions);
5     mock.perform(get("/instructions"))
6         .andExpect(content().contentTypeCompatibleWith(MediaType.APPLICATION_JSON))
7         .andExpect(status().isOk())
8         .andExpect(jsonPath("$", hasSize(1)));
9 }
```

Listing 5.3: Test 3

#### Test 4: Stvaranje Uputa.

Komponenta: InstructionsController

Zahtjev: HTTP POST /instructions

Mock Podaci:

- Korisnik (ADMIN)

Očekivani rezultat:

- Kontroler u JSON formatu vraća atribut novostvorene Upute, HTTP odgovor ima status 201 (Created).

```
1 @Test
2 @WithMockUser(roles = {"ADMIN"})
3 void shouldAddInstruction() throws Exception {
4     Instruction instruction = new Instruction(2L, allUsers.get(0),
5         allUsers.get(1), LocalDateTime.now(), "bla");
6     InstructionDTO instructionDTO = new InstructionDTO(instruction);
7     when(instructionService.createInstruction(any(InstructionDTO.class))
8         ).thenReturn(instruction);
9     MvcResult result = mock.perform(post("/instructions")
10         .contentType(MediaType.APPLICATION_JSON)
11         .content(asJsonString(instructionDTO)))
12         .andExpect(status().isCreated())
13         .andReturn();
14     assertThat(result).isNotNull();
15     assertThat(result.getResponse().getContentAsString()).isNotEmpty();
16     assertThat(result.getResponse().getContentAsString()).isEqualTo(
17         asJsonString(instruction));
18 }
```

15 | }

Listing 5.4: Test 4

**Test 5: Dohvaćanje nepostojeće Upute.**

Komponenta: InstructionsController

Zahtjev: HTTP GET /instruction/{id}

Mock Podaci:

- Korisnik (ADMIN)

Očekivani rezultat:

- HTTP odgovor ima status 404 (Not Found).

```
1 @Test
2 @WithMockUser
3 void shouldHandleEntityMissingException() throws Exception {
4     Long id = 45L;
5     when(instructionService.findById(id)).thenReturn(new
6         EntityMissingException("No instruction with such id"));
7     mock.perform(get("/instruction/{id}", id)).andExpect(status().
8         isNotFound());
9 }
```

Listing 5.5: Test 5

**Test 6: Dohvaćanje svih doktora u aplikaciji.**

Komponenta: UsersController

Zahtjev: HTTP GET /users/doctors

Mock Podaci:

- Korisnik (PARENT)

Očekivani rezultat:

- HTTP odgovor ima status 403 (Forbidden), jer roditelj ne smije dobiti popis svih liječnika u aplikaciji.

```
1 @Test
2 @WithMockUser(roles = {"PARENT"})
3 void shouldReturnForbidden() throws Exception {
4     mock.perform(get("/users/doctors")).andExpect(status().isForbidden());
5 }
```

Listing 5.6: Test 6

### Test 7: Dodavanje korisnika.

Komponenta: UsersController

Zahtjev: HTTP POST /users

Mock Podaci:

- Korisnik (ADMIN)

Očekivani rezultat:

- Kontroler u JSON formatu vraća attribute novostvorenog Korisnika, HTTP odgovor ima status 201 (Created).

```
1 @Test
2 @WithMockUser(roles = {"ADMIN"})
3 void shouldCreateNewUser() throws Exception {
4     UserDTO userDTO = new UserDTO();
5     userDTO.setId(3L);
6     userDTO.setEmail("user@gmail.com");
7     userDTO.setPassword("pass");
8     userDTO.setFirst_name("Karla");
9     userDTO.setLast_name("Tinic");
10    userDTO.setOib("25768672773");
11    User user = dtoManager.userDTOToUser(userDTO);
12
13    List<String> roles = List.of("doctor");
14    CreateUserRequest createUserRequest = new CreateUserRequest();
15    createUserRequest.setUserDTO(userDTO);
16    createUserRequest.setRoles(roles);
17
18    when(userService.createUser(userDTO, roles)).thenReturn(user);
19    MvcResult result = mock.perform(post("/users").contentType(MediaType
        .APPLICATION_JSON))
```



```
20         .content( asJsonString( createUserRequest() ) )
21         .andExpect( status().isCreated() )
22         .andReturn();
23     assertThat( result.getResponse().getContentAsString() ).isEqualTo(
24         asJsonString( user ) );
25 }
```

Listing 5.7: Test 7

### Test 8: Registracija korisnika.

Komponenta: AuthenticationController

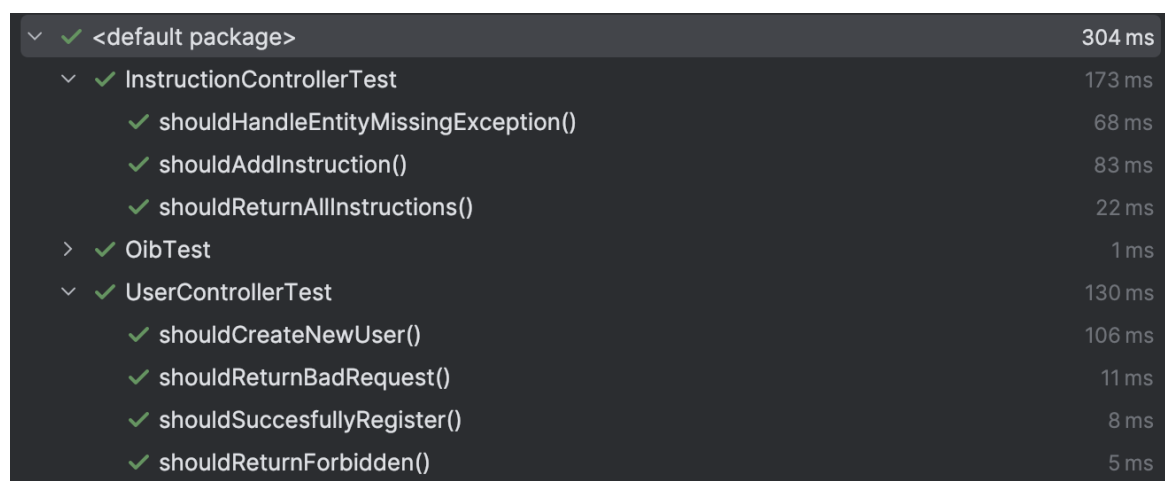
Zahtjev: HTTP POST /register

Očekivani rezultat:

- Kontroler u JSON formatu vraća attribute novostvorenog Korisnika, HTTP odgovor ima status 201 (Created).

```
1  @Test
2  void shouldSuccesfullyRegister() throws Exception{
3      User user = new User();
4      user.setOib("25768672773");
5      user.setEmail("bla@mail.com");
6      user.setFirst_name("Janko");
7      user.setLast_name("Bananko");
8      user.setPassword("pass");
9      UserDTO userDTO = dtoManager.userToUserDTO(user);
10     when(userService.createUser(userDTO, List.of("parent"))).thenReturn(
11         user);
12     mock.perform(post("/register").contentType(MediaType.
13         APPLICATION_JSON)
14         .content( asJsonString( user ) )
15         .andExpect( status().isCreated() )
16         .andExpect( content().string( equalTo( "Successfully registered
17         " ) ) ) );
18 }
```

Listing 5.8: Test 8



✓ <default package>	304 ms
✓ InstructionControllerTest	173 ms
✓ shouldHandleEntityMissingException()	68 ms
✓ shouldAddInstruction()	83 ms
✓ shouldReturnAllInstructions()	22 ms
> OibTest	1 ms
✓ UserControllerTest	130 ms
✓ shouldCreateNewUser()	106 ms
✓ shouldReturnBadRequest()	11 ms
✓ shouldSuccesfullyRegister()	8 ms
✓ shouldReturnForbidden()	5 ms

Slika 5.1: Rezultati testiranja na lokalnom stroju.

### 5.2.2 Ispitivanje sustava

Testovi ispitivanja sustava su napisani u Selenium IDE dodatku za preglednik i prate nekoliko obrazaca uporabe, ali su vezani uz jednu cjelinu - rad s preporukama za bolovanje. U ovom scenariju testiranja sudjeluje više uloga (roditelj, pedijatar, liječnik) i svi koraci koje jedna uloga izvodi su grupirani u zasebnu "grupu koraka". Također, za svaki korak je naveden i pripadajući očekivani rezultat.

#### Grupa koraka 1: Dodavanje preporuke za bolovanje

Koraci:

1. Prijava u sustav kao pedijatar ("pedijatar@mail.com")
2. Odabir opcije "Bolovanja" iz alatne trake.
3. Odabir opcije "Dodaj preporuku"
4. Odabir pregleda na temelju kojeg se izdaje preporuka ("Tamara Stanic 2023-12-25") unutar modalnog ekrana.
5. Odabir opcije "Spremi".

Očekivani rezultati:

1. Pedijatar se može prijaviti u sustav.
2. Postoji opcija "Bolovanja" u alatnoj traci.
3. Dostupna je opcija "Dodaj preporuku".
4. Postoji izbornik pregleda unutar modalnog ekrana u kojem je moguće izabrati traženi pregled.

5. Postoji opcija "Spremi" unutar modalnog ekrana.

## **Grupa koraka 2: Pregled neobrađene preporuke za bolovanje**

Koraci:

1. Prijava u sustav kao roditelj ("roditelj@mail.com")
2. Odabir opcije "Bolovanja" iz alatne trake.
3. Odabir preporuke iz popisa preporuke ("Tamara Stanic 2023-12-25").

Očekivani rezultati:

1. Roditelj se može prijaviti u sustav.
2. Postoji opcija "Bolovanja" u alatnoj traci.
3. Postoji tražena preporuka za bolovanje u popisu preporuka.
4. Pod "Status" piše "Čeka odobrenje." unutar modalnog ekrana.

## **Grupa koraka 3: Odobravanje preporuke za bolovanje**

Koraci:

1. Prijava u sustav kao liječnik obiteljske medicine ("doktor@mail.com")
2. Odabir opcije "Bolovanja" iz alatne trake.
3. Odabir preporuke iz popisa preporuke ("Tamara Stanic 2023-12-25").
4. Odabir opcije "Odobri" iz modalnog ekrana.

Očekivani rezultati:

1. Liječnik obiteljske medicine se može prijaviti u sustav.
2. Postoji opcija "Bolovanja" u alatnoj traci.
3. Postoji tražena preporuka za bolovanje u popisu preporuka.
4. Postoji opcija "Odobri" unutar modalnog okvira.

## **Grupa koraka 4: Pregled odobrene preporuke za bolovanje**

Koraci:

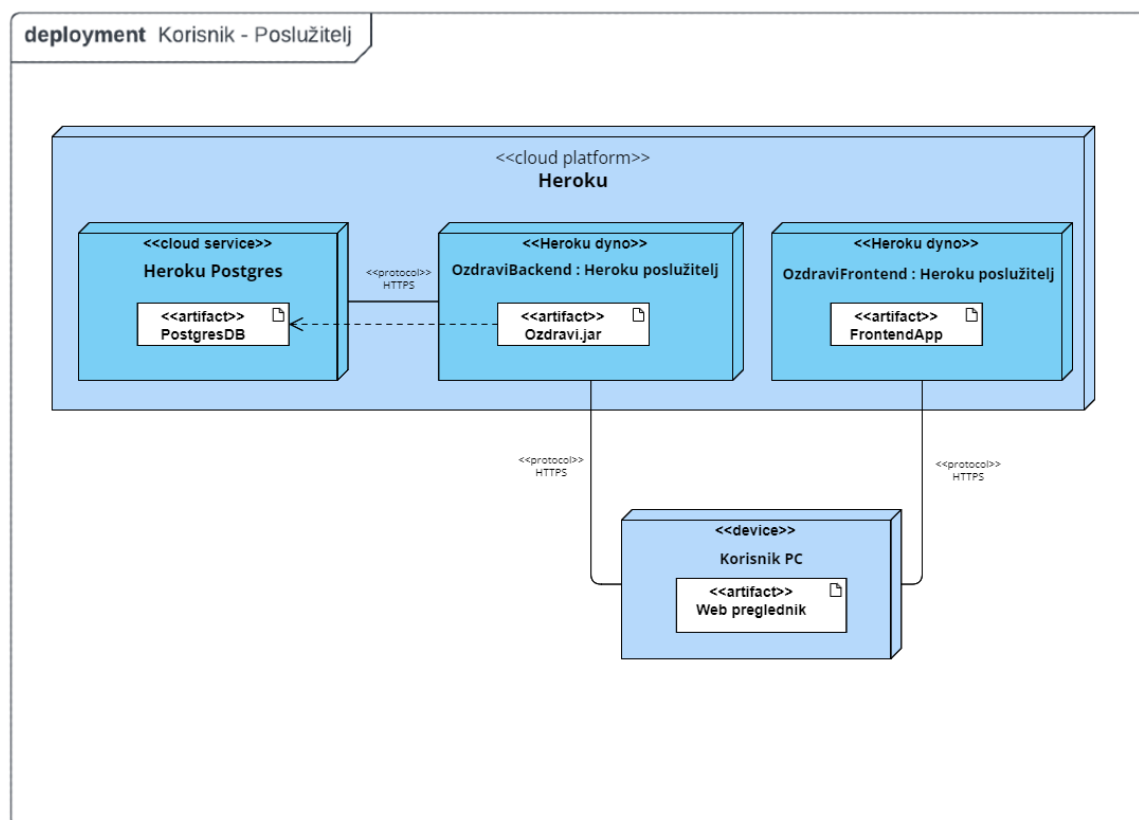
1. Prijava u sustav kao roditelj ("roditelj@mail.com")
2. Odabir opcije "Bolovanja" iz alatne trake.
3. Odabir preporuke iz popisa preporuke ("Tamara Stanic 2023-12-25").

Očekivani rezultati:

1. Roditelj se može prijaviti u sustav.
2. Postoji opcija "Bolovanja" u alatnoj traci.
3. Postoji tražena preporuka za bolovanje u popisu preporuka.
4. Pod "Status" piše "Odobreno." u modalnom ekranu preporuke.

## 5.3 Dijagram razmještaja

Dijagrami razmještaja prikazuju konfiguraciju sklopovlja i programske podrške koja se koristi za implementaciju sustava u njegovom operativnom okruženju. Na cloud platformi smješten je poslužitelj baze podataka te web poslužitelji za backend i frontend. Korisnici pristupaju web aplikaciji putem svog web preglednika. Sustav je temeljen na arhitekturi "klijent - poslužitelj" s komunikacijom između korisničkih računala (roditelj, pedijatar, doktor, administrator) i poslužitelja putem HTTPS veze.



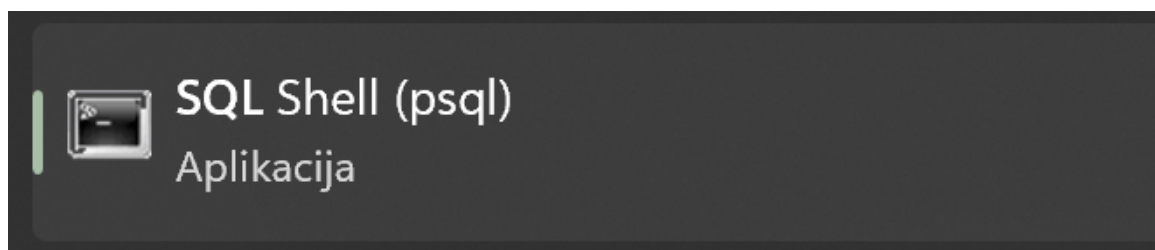
Slika 5.2: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### Postavljanje baze podataka

U procesu postavljanja baze podataka za aplikaciju "Ozdravi", prvi korak je preuzimanje *PostgreSQL-a* s ovdje dostupne poveznice za operacijski sustav Windows. Ovaj instalacijski paket obuhvaća PostgreSQL poslužitelj, pgAdmin (grafički alat za upravljanje i razvoj baza podataka) te StackBuilder (upravitelj paketa koji olakšava preuzimanje i instalaciju dodatnih PostgreSQL alata i upravljačkih programa). Nakon završetka instalacije, slijedi standardni postupak postavljanja korisničkog imena i lozinke (neobavezno).

Nakon uspješne instalacije, pokreće se SQL Shell aplikacija.



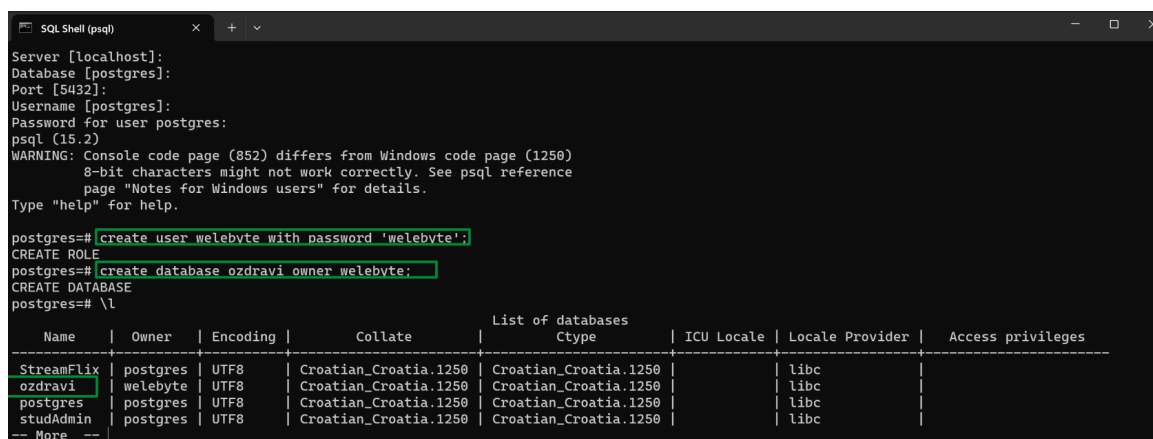
Slika 5.3: SQL Shell aplikacija

Kako bi se pristupilo komandnoj liniji, potrebno je pritisnuti tipku Enter dok linija ne započne s *postgres=#* (ako je postavljena lozinka tijekom konfiguracije PostgreSQL-a, onda ju treba unijeti kada sustav zatraži: *Password for user postgres:*). Nakon toga, izvršavaju se sljedeće naredbe:

```
create user welebyte with password 'welebyte';
```

```
create database ozdravi owner welebyte;
```

Sada bi trebala biti stvorena baza podataka "ozdravi", čiji je vlasnik korisnik "welebyte". Naredbom `\l` može se provjeriti lista svih postojećih baza podataka i njihovih vlasnika.



```
SQL Shell (psql)
Server [localhost]:
Database [postgres]:
Port [5432]:
Username [postgres]:
Password for user postgres:
psql (15.2)
WARNING: Console code page (852) differs from Windows code page (1250)
8-bit characters might not work correctly. See psql reference
page "Notes for Windows users" for details.
Type "help" for help.

postgres=# create user welebyte with password 'welebyte';
CREATE ROLE
postgres=# create database ozdravi owner welebyte;
CREATE DATABASE
postgres=# \l

   Name | Owner | Encoding | Collate | List of databases | ICU Locale | Locale Provider | Access privileges
-----|-----|-----|-----|-----|-----|-----|-----
 StreamFlix | postgres | UTF8 | Croatian_Croatia.1250 | Croatian_Croatia.1250 |  | libc | 
 ozdravi | welebyte | UTF8 | Croatian_Croatia.1250 | Croatian_Croatia.1250 |  | libc | 
 postgres | postgres | UTF8 | Croatian_Croatia.1250 | Croatian_Croatia.1250 |  | libc | 
 studAdmin | postgres | UTF8 | Croatian_Croatia.1250 | Croatian_Croatia.1250 |  | libc | 
-- More --
```

Slika 5.4: Postavljanje baze podataka

## Pokretanje aplikacije koristeći IntelliJ

Nakon uspješne konfiguracije baze podataka, prelazi se na postavljanje aplikacije koristeći IntelliJ IDE. Potrebno je otvoriti mapu *IzvorniKod\ozdravi-backend* kao novi projekt u IntelliJ-u, a aplikacija se dalje pokreće jednostavnim pokretanjem datoteke *OzdraviBackendApplication.java*. Aplikacija je sada dostupna na adresi: <http://localhost:8000>.

## Postavljanje frontenda

Sljedeći korak uključuje postavljanje frontenda. Prvotno se preuzima okruženje za izvršavanje JavaScript koda, Node.js, s ovdje dostupne poveznice, sljedeći standardne korake instalacije. Zatim, u naredbenom retku (terminalu), potrebno se pozicionirati u direktorij u kojem je spremljen izvorni kod za frontend: *...IzvorniKod\ozdravi-frontend*. Prilikom prvog pokretanja, izvršava se naredba *npm install* kako bi se preuzeli svi potrebni paketi, a zatim se aplikacija pokreće naredbom *npm start*.



```
C:\ . . . \IzvorniKod\ozdravi-frontend>npm install
```

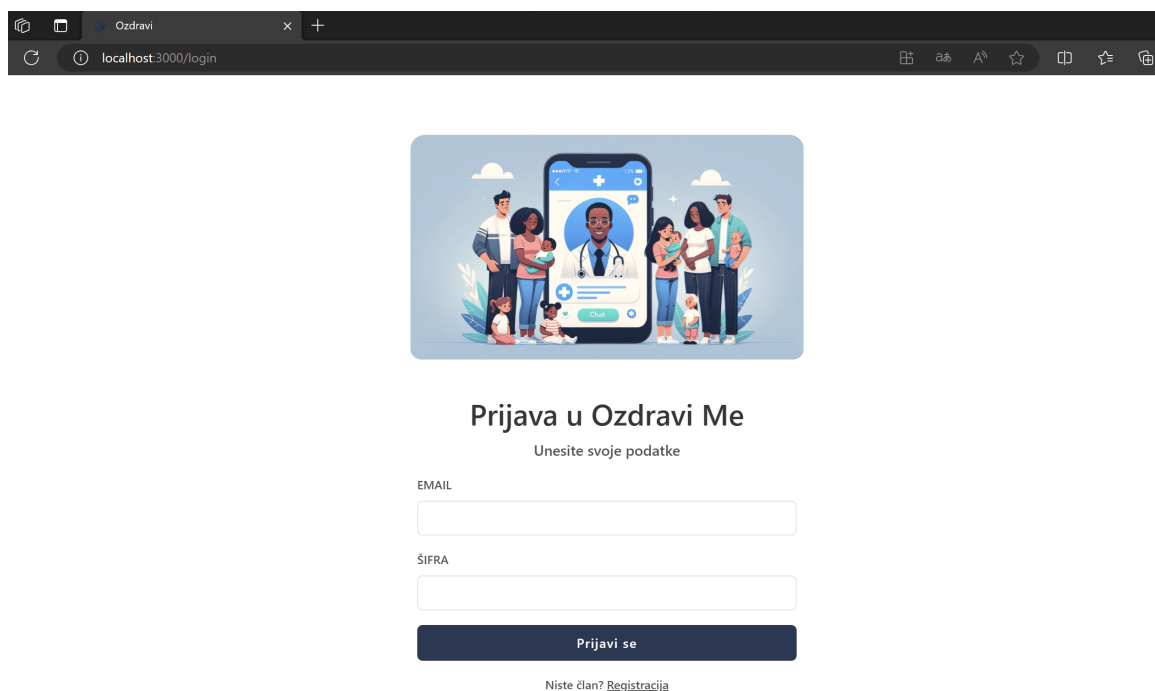
Slika 5.5: Preuzimanje paketa



```
C:\ . . . \IzvorniKod\ozdravi-frontend>npm start
```

Slika 5.6: Pokretanje aplikacije

Nakon pokretanja, aplikaciju je moguće vidjeti na adresi: `http://localhost:3000`.



Slika 5.7: Aplikacija na portu 3000

Nakon svake promjene i spremanja koda, stranica se automatski ponovno učitava. Osim toga, postoji mogućnost promjene porta aplikacije uređivanjem datoteke *package.json*, gdje se u liniji koda "start": *set PORT=3006 && react-scripts start* može izmijeniti broj porta prema želji.



## 6. Zaključak i budući rad

Zadatak grupe bio je razvoj web-aplikacije Ozdravi. Projekt "Ozdravi" predstavlja značajan korak prema modernizaciji zdravstvenog sustava, omogućujući brži pristup informacijama, efikasnije upravljanje resursima te bolje praćenje zdravstvenih podataka. Tijekom 13 tjedana rada, tim je uspješno implementirao zadanu web aplikaciju. Provedba projekta bila je podijeljena u tri faze.

Prva faza, poznata kao faza inicijacije (*en. initiation*), predstavlja početak cijelog procesa razvoja. Provedeno je formiranje tima, nakon čega je uslijedilo detaljno upoznavanje sa zadatkom. Time su članovi tima stekli jasnu sliku o svrsi aplikacije, funkcionalnostima i korisničkim zahtjevima. U ovoj fazi odabrane su odgovarajuće tehnologije i alati.

Druga faza projekta, odnosno faza otkrivanja (*engl. discovery*), obilježena je strukturiranim pristupom definiranja zahtjeva i dokumentacijom, što je značajno olakšalo implementaciju u drugoj fazi. Prvi samostalni zadaci članova tima uključivali su usvajanje znanja o odabranim tehnologijama i alatima. Izrada vizualnih prikaza idejnih rješenja problemskog zadatka pomogla je u rješavanju nedoumica oko implementacije rješenja. Obrasci uporabe, sekvencijski dijagrami te modeli baze podataka bili su ključni u usmjeravanju podtimova zaduženih za razvoj backenda i frontenda.

U trećoj fazi, poznata kao faza isporuke (*engl. delivery*), naglasak je bio na samostalnom radu članova tima na implementaciji i dokumentaciji. Kvalitetna dokumentacija olakšat će buduće prilagodbe sustava te omogućiti korisnicima lakše korištenje aplikacije. Osim realizacije rješenja, u ovoj fazi bilo je potrebno izraditi ostale UML dijagrame te popratnu dokumentaciju kako bi se budućim korisnicima omogućilo jednostavnije korištenje te izvršenje potrebnih preinaka sustava. Redoviti sastanci tima pridonijeli su informiranosti o napretku projekta, izvršnoj organizaciji te učinkovitoj podjeli zadataka.

Nakon uspješne implementacije, projekt "Ozdravi" otvara prostor za daljnje poboljšanje i proširenje. U skladu s time, moguće je izvesti refaktorizaciju određenih dijelova koda kako bi se smanjio tehnički dug i poboljšala održivost aplikacije. Također, razmatra se unapređenje deployment okruženja kako bi se postigle bo-

lje performanse i povećala dostupnost sustava. Nužno je i uvesti CI/CD (Continuous Integration/Continuous Delivery) sustava koji bi mogao automatski napraviti deploy aplikacije na javni poslužitelj u slučaju commita u granu "main" i pritom osigurati kvalitetu isporučenog koda izvršavanjem testova. Potencijalno proširenje funkcionalnosti sustava obuhvaća implementaciju mobilne aplikacije, pružajući dodatnu fleksibilnost korisnicima. Nadogradnja dodatnim funkcionalnostima ključna je u zadovoljavanju potreba korisnika, omogućavajući dodavanje novih značajki koje će obogatiti korisničko iskustvo. Uz to, postoji mogućnost unapređenja korisničkog sučelja (UI) i korisničkog iskustva (UX), čime će se poboljšati funkcionalnost sustava. S kontinuiranim praćenjem povratnih informacija korisnika, osigurat će se da aplikacija "Ozdravi" uvijek bude u skladu s najvišim standardima i potrebama moderne zdravstvene tehnologije.

Sudjelovanje u projektu "Ozdravi" pružilo je vrijedno iskustvo članovima tima, naglašavajući važnost suradnje, organizacije te kontinuiranog učenja u radu na složenim projektima.

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.

# Indeks slika i dijagrama

2.1	Početna stranica "Portal zdravlje" . . . . .	9
3.1	Dijagram obrasca uporabe, funkcionalnosti korisnika i roditelja . . .	25
3.2	Dijagram obrasca uporabe, funkcionalnosti liječnika obiteljske medicine i pedijatra . . . . .	26
3.3	Dijagram obrasca uporabe, funkcionalnosti administratora . . . . .	27
3.4	Sekvencijski dijagram za UC9 . . . . .	28
3.5	Sekvencijski dijagram za UC14 . . . . .	29
3.6	Sekvencijski dijagram za UC15 . . . . .	30
4.1	Dijagram arhitekture sustava "Ozdravi". . . . .	32
4.2	Dijagram infrastrukture sustava "Ozdravi". . . . .	34
4.3	ER dijagram baze podataka . . . . .	40
4.4	Dijagram razreda - controller i service razredi . . . . .	41
4.5	Dijagram razreda - dio Data transfer objects . . . . .	42
4.6	Dijagram razreda - domain razredi . . . . .	43
4.7	Dijagram stanja . . . . .	45
4.8	Dijagram aktivnosti . . . . .	47
4.9	Dijagram komponenti . . . . .	48
5.1	Rezultati testiranja na lokalnom stroju. . . . .	57
5.2	Dijagram razmještaja . . . . .	60
5.3	SQL Shell aplikacija . . . . .	61
5.4	Postavljanje baze podataka . . . . .	62
5.5	Preuzimanje paketa . . . . .	62
5.6	Pokretanje aplikacije . . . . .	62
5.7	Aplikacija na portu 3000 . . . . .	63

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: D. Matić, L. Crvelin, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 35min
- Teme sastanka:
  - izbor tehnologije
  - izbor teme
  - ime tima
  - osnovna podjela posla
  - osiguravanje pristupa na sve servisa svim članovima tima
- Zaključak:

Dogovoreni su taskovi za prvi sprint (do četvrtka 26.10). Oni za većinu ljudi uključuju učenje gita i tehnologija koje su odabrali (backend/frontend), a L. Crvelin treba i napraviti prve korake u organizaciji dokumentacije. Voditelj D. Matić će postaviti repozitorij na GitHubu i osigurati svima pristup. Dogovorena je i radionica na temu značajki (featurea) aplikacije sutra u 12 sati, gdje ćemo pokušati prioritzirati posao i složiti Jiru.

### 2. sastanak

- Datum: 20. listopada 2023.
- Prisustvovali: D. Matić, L. Crvelin, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 1h 45min
- Teme sastanka:
  - analiziranje značajki aplikacije i napisati iz njih epice i storije u Jiri
  - stvaranje dijagrama
  - daljnja podjela posla

- Zaključak:

Detaljno su analizirane značajke aplikacije i stvoreni dijagrami iz kojih se može stvoriti definicija baze podataka što će do kraja sprinta učiniti V. Kumanović i prioritizirani popis zadataka koji će u Jiri stvoriti D. Matić. Također, identificirali smo nejasnoće u samoj definiciji zadatka koje ćemo prenijeti asistentici. M. Vidaković će do kraja sprinta istražiti kako se pravilno postavlja Spring projekt, a K. Valečić će isto učiniti za React u čemu će mu pomoći M. Lešković.

Dogovorene su još dvije radionice sljedeći tjedan, u četvrtak i petak na temu postavljanja Reacta i Springa.

### 3. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 1h 30min
- Teme sastanka:

- planiranje i praćenje napretka
- postavljanje backend-a prema svim pravilima i omogućavanje nesmetanog razvoja u budućnosti.

- Zaključak:

Potvrdili smo da svi imaju potreban pristup na JIRU i GitHub repozitoriju.

L. Crvelin je izložila napredak u dokumentiranju i predstavila način prikupljanja podataka za pojedine dijelove dokumentacije. Podaci o napretku će se prikupljati kroz komentare na JIRA taskovima i kroz bilješke sa sastanaka. Sljedeći koraci vezani uz izradu dokumentacije su raspisivanje obrazaca uporabe i pisanje opisa projektnog zadatka. To će preuzeti L. Crvelin i V. Kumanović. Također, L. Crvelin je naglasila važnost praćenja potrošenog vremena za svaki zadatak koji radimo. To će u JIRI postaviti D. Matić.

Predstavljena je i schema baze podataka koju smo komentirali i utvrdili koje su potrebne promjene. Te će promjene u ovom sprintu učiniti V. Kumanović i M. Vidaković te će ih predstaviti timu na sljedećem planiranju sprinta u četvrtak.

M. Vidaković je prezentirala napredak vezan uz postavljanje Spring bac-

kenda i preuzela zadatak integracije backenda s Postgresom što se može događati usporedno uz razvijanje scheme baze.

K. Valečić i M. Lešković su potvrdili da su spremni za nadolazeću radionicu vezanu uz postavljanje Reacta, koja će se održati sutra (petak, 27.10. u 14 sati). Nakon te radionice bit ćemo spremni za početak razvoja na frontendu, što će preuzeti L. Cvetkovski i K. Valečić.

Komentirali smo rad s gitom te je M. Lešković pokazao workflow u GitHub Desktop alatu. Naglašena je važnost da više ljudi pogleda PR prije mergea s glavnom granom.

#### 4. sastanak

- Datum: 27. listopada 2023.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić
- Trajanje: 45min
- Teme sastanka:
  - planiranje i praćenje napretka
  - postavljanje frontend-a prema svim pravilima i omogućavanje nesmetanog razvoja u budućnosti

- Zaključak:

M. Lešković je postavio temelje React projekta i detaljno objasnio što i kako radi. Dogovorili smo se da ćemo za izgradnju korisničkog sučelja koristiti Bootstrap, a integraciju Bootstrapa s projektom će napraviti L. Cvetkovski. Za to vrijeme će K. Valečić naučiti kako se Bootstrap koristi i onda s L. Cvetkovski izgraditi prvih nekoliko stranica aplikacije (login/register, homepage, profile edit page) do kraja ovog sprinta. M. Lešković će od sad raditi na backend dijelu.

Dogovorili smo se da ćemo dodati README na projekt s uputama kako pokrenuti backend / frontend. To je bitno da svi mogu jednostavno postaviti projekt na svojim računalima.

#### 5. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 45min
- Teme sastanka:

- praćenje napretka na frontendu
- pregled funkcionalnih zahtjeva
- Zaključak:  
L. Crvelin je pokazala dosadašnji napredak vezan uz definiranje funkcionalnih zahtjeva i izložila nejasnoće koje imamo u ovom trenutku. Kroz raspravu smo razjasnili sve i omogućili daljnji razvoj dokumentacije. L. Cvetkovski je pokazao napredak vezan uz razvoj frontenda. Dogovoreno je da će frontend tim imati zasebne sastanke u terminu u kojem se dogovore kako bi odredili sljedeće korake.

#### 6. sastanak

- Datum: 1. studenoga 2023.
- Prisustvovali: L. Crvelin, L. Cvetkovski, M. Lešković
- Trajanje: 1h
- Teme sastanka:
  - Razdvajanje uloga
  - Određivanje podstranica za svaku ulogu
  - Pregled dosadašnjeg napretka
  - podjela zadatka
- Zaključak:  
Dogovoreno je da trebamo razdvojiti uloge na platformi, odnosno definirati različite korisničke uloge: pedijatar, liječnik obiteljske medicine (OM), i roditelj. Razgovarali smo o potrebi definiranja specifičnih podstranica za svaku ulogu kako bismo omogućili prilagođeni sadržaj i funkcionalnosti za svaku kategoriju korisnika. Odlučeno je da će adminu biti omogućeno upravljanje svim računima korisnika s jedne centralne stranice nazvane "racuni.js". Ovo će pojednostaviti administraciju i omogućiti efikasnije upravljanje korisničkim računima. Razgovarali smo o raspodjeli zadatka za izradu osnovnih kostura stranica za svaku ulogu na frontendu. L. Cvetkovski će raditi na stranici za pedijatra i liječnika obiteljske medicine, a K. Valečić će raditi na stranici za roditelja.

#### 7. sastanak

- Datum: 1. studenoga 2023.
- Prisustvovali: D. Matić, M. Vidaković
- Trajanje: 45min
- Teme sastanka:



- projektiranje baze podataka
- Zaključak:

#### 8. sastanak

- Datum: 2. studenoga 2023.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 2h 45min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Dogovorili smo sljedeće korake koje treba napraviti vezano uz dokumentaciju kako bismo mogli tražiti povratnu informaciju na konzultacijama u utorak (7.11). L. Crvelin će prepisati definirane tablice baze iz Excela u dokumentaciju, a V. Kumanović će iz definiranih tablica nacrtati dijagram baze podataka koji je također dio dokumentacije. Prošli smo kroz sve obrasce uporabe koje smo do sad definirali i dogovorili se da će oni biti dovršeni do ponedjeljka, što će učiniti L. Crvelin.

D. Matić je pokazao napredak u izboru i postavljanju deploy platforme. Izabrali smo Heroku i napravili osnovno postavljenje dviju aplikacija (frontend i backend).

M. Vidaković je pokazala kako se Postgres baza spaja na Spring aplikaciju što nam je bio preduvjet da možemo nastaviti razvoj na backendu, ali i da možemo napraviti potpun deployment aplikacije. Nastavno na to će M. Lešković implementirati mehanizme za prijavu i registraciju korisnika pomoću Spring Security modula, a D. Matić će napraviti potrebne korake na Heroku platformi vezane uz deploy aplikacije. M. Vidaković će u sljedećem sprintu implementirati mehanizam uloga na entitetu korisnika.

L. Cvetkovski i K. Valečić su podijelili plan za razvoj frontenda s ostalima. Potvrdili smo daljnje korake vezane uz razvoj React aplikacije koji uključuju restrukturiranje i razvoj homescreena. Dogovoreno je da će

frontend tim razraditi najbolje načine kako nastaviti raditi s obzirom da backend još uvijek zaostaje za frontendom.

#### 9. sastanak

- Datum: 9. studenoga 2023.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 1h 30min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Analizirali smo povratne informacije koje smo dobili na konzultacijama u utorak i podijelili posao vezan uz to u ovom sprintu. L. Crvelin će napraviti sve potrebne promjene na UC-ovima, Vedran Kumanović će modificirati dijagram baze podataka, a Dorian Matić će dovršiti odlo-mak "Arhitektura sustava".

Leon Cvetkovski je predstavio napredak u refaktoriziranju frontenda koji su on i Krešimir Valečić napravili u ovom sprintu. Zaključili smo da je frontend sad spreman za spajanje s backendom.

Mihael Lešković je opisao probleme s kojima se susreo pri postavljanju Spring Security modula za backend aplikacije. Dogovoreno je da će on i Dorian Matić probati što prije to popraviti i omogućiti daljnju razvoj frontenda.

#### 10. sastanak

- Datum: 11. studenoga 2023.
- Prisustvovali: D. Matić, L. Cvetkovski, M. Lešković, K. Valečić
- Trajanje: 1h 30min
- Teme sastanka:
  - Omogućiti frontend timu da spoji React aplikaciju s backendom
- Zaključak:

Predstavljen je dio backenda vezan uz Spring Security. Frontend tim je naučio kako se pokreće backend aplikacija lokalno. Dogovoreni su sljedeći koraci.

#### 11. sastanak

- Datum: 16. studenoga 2023.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 45min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Timovi za dokumentaciju, backend i frontend su predstavili svoj napredak u ovom sprintu. Dogovorene su zadnje promjene u dokumentaciji prije verzije 1.0.

Dogovoreno je da sljedeći sprint zbog međuispita kreće tek u četvrtak, 30.11. Do tad, svi timovi trebaju smisliti prijedloge daljnjeg djelovanja. Svi članovi tima su pristali na cilj da cjelokupna funkcionalnost aplikacije bude gotova do 1.1.2024.

## 12. sastanak

- Datum: 30. studenoga 2023.
- Prisustvovali: D. Matić, L. Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 40min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Dogovoreni su daljni koraci za frontend, backend i dokumentaciju. Vedran Kumanović će istražiti koji su nam dijagrami potrebni u dokumentaciji i koji podaci nam trebaju kako bismo ih izradili. Analizirat će kako te podatke prikupiti i imamo li ih već prikupljene. Osim toga, napisat će i odlomak "Upute za puštanje u rad aplikacije" u dokumentaciji, na temelju README-ja.

Mihael Lešković će nastaviti razvijati podršku za usere tako što će implementirati sve atribute usera unutar aplikacije i također njihove validacije. Mia Vidaković će nastaviti raditi na implementaciji uloga unutar aplikacije.

Krešimir Valečić i Leon Cvetkovski će nadograditi ekran za registraciju s novim poljima koja su potrebna i napraviti će novi ekran za unos ili izmjenu atributa usera.

## 13. sastanak

- Datum: 4. prosinca 2023.
- Prisustvovali: D. Matić, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 40min
- Teme sastanka:
  -
- Zaključak:

## 14. sastanak

- Datum: 7. prosinca 2023.
- Prisustvovali: D. Matić, L. Crvelin, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 1h 5min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Dogovoreno je da će se tijekom sljedećeg sprinta napraviti radionica vezana uz preostale dijagrame koje treba nacrtati tim za dokumentaciju će predstaviti kakvi su to dijagrami i prikupiti informacije potrebne da ih se nacrtaju.

Vedran Kumanović mora završiti odlomak "Upute za puštanje u rad aplikacije", što mu je ostalo od prethodnog sprinta.

Leon Cvetkovski i Krešimir Valečić moraju završiti ekrane koji su im ostali od prošlog sprinta i nastaviti raditi na ekranima za preglede i spajanje doktora i pacijenata.

Mia Vidaković će nastaviti s implementacijama uloga, a Mihael Lešković će nastaviti rad na korisnicima i pregledima.

## 15. sastanak

- Datum: 11. prosinca 2023.
- Prisustvovali: D. Matić, L. Crvelin, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 1 h 30 min

- Teme sastanka:
  - koordinacija o napretku unutar jednog sprinta
- Zaključak:

Diskusija i rješavanje tehničkih problema vezanih uz dodjeljene zadatke na frontendu i backendu.

#### 16. sastanak

- Datum: 14. prosinca 2023.
- Prisustvovali: D. Matić, L. Crvelin, L.Cvetkovski, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 1 h 30 min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Tim za dokumentaciju će pripremiti radionicu vezanu uz dijagrame stanja i komponenti, a Lucia Crvelin će napisati i odlomak "Korištene tehnologije i alati".

Dorian Matić će provesti reviziju dokumentacije kako bismo utvrdili u koji dijelovi dokumentacije se razlikuju od implementacije.

Frontend tim će integrirati ekrane vezane u korisnike s backendom.

Backend tim će nastaviti rad na novim entitetima i otvarati API endpote koji su definirani u specifikaciji API-ja.

#### 17. sastanak

- Datum: 18. prosinca 2023.
- Prisustvovali: D. Matić, L. Crvelin, V. Kumanović, M. Lešković, K. Valečić, M. Vidaković
- Trajanje: 30min
- Teme sastanka:
  - mid-sprint sync
- Zaključak:

Diskusija i rješavanje tehničkih problema vezanih uz dodjeljene zadatke na frontendu i backendu.

#### 18. sastanak

- Datum: 21. prosinca 2023.
- Prisustvovali: D. Matić, L. Crvelin, M. Lešković, V. Kumanović, K. Valečić, M. Vidaković

- Trajanje: 1h 30min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

#### 19. sastanak

- Datum: 26. prosinca 2023.
- Prisustvovali:
- Trajanje: 1 h 30 min
- Teme sastanka:
  - koordinacija o napretku unutar jednog sprinta
- Zaključak:

Diskusija i rješavanje tehničkih problema vezanih uz dodjeljene zadatke na frontendu i backendu.

#### 20. sastanak

- Datum: 28. prosinca 2023.
- Prisustvovali: D. Matić, L. Crvelin, M. Lešković, M. Vidaković
- Trajanje: 1h
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

Tim za dokumentaciju će pripremiti radionicu vezanu uz dijagrame stanja i komponenti, a Lucia Crvelin će napisati i odlomak "Korištene tehnologije i alati".

Dorian Matić će provesti reviziju dokumentacije kako bismo utvrdili u koji dijelovi dokumentacije se razlikuju od implementacije.

Frontend tim će integrirati ekrane vezane u korisnike s backendom.

Backend tim će nastaviti rad na novim entitetima i otvarati API endpoin-  
te koji su definirani u specifikaciji API-ja.

#### 21. sastanak

- Datum: 2. siječnja 2024.
- Prisustvovali: D. Matić, L. Crvelin, M. Lešković
- Trajanje: 30 min
- Teme sastanka:
  - mid-sprint sync

- Zaključak:  
Diskusija i rješavanje tehničkih problema vezanih uz dodjeljene zadatke na frontendu i backendu.

## 22. sastanak

- Datum: 4. siječnja 2024.
- Prisustvovali: D. Matić, L. Crvelin, L. Cvetkovski, M. Lešković, M. Vidaković
- Trajanje: 1 h
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

## 23. sastanak

- Datum: 11. siječnja 2024.
- Prisustvovali:
- Trajanje: 1 h 30 min
- Teme sastanka:
  - planiranje i praćenje napretka
- Zaključak:

## Tablica aktivnosti

	Dorian Matic	Lucia Crvelin	Leon Cvetkovski	Vedran Kumanović	Mihael Lešković	Krešimir Valečić	Mia Vidaković
Upravljanje projektom	4.5						
Opis projektnog zadatka		3.5					
Funkcionalni zahtjevi		3					
Opis pojedinih obrazaca		8					
Dijagram obrazaca		5					
Sekvencijski dijagrami		4.5					
Opis ostalih zahtjeva		1					
Arhitektura i dizajn sustava	3						
Baza podataka	2			5			3
Dijagram razreda				2.5			1
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							4
Ispitivanje programskog rješenja						1	
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja	1						
Zaključak i budući rad							

Nastavljeno na idućoj stranici



Nastavljeno od prethodne stranice

	Dorian Matić	Lucia Crvelin	Leon Cvetkovski	Vedran Kumanović	Mihael Lešković	Krešimir Valečić	Mia Vidaković
Popis literature							
<i>Integracija baze podataka</i>							7
<i>Spring security</i>	4				15		
<i>Frontend inicijalizacija</i>			10		3	1	
<i>Deployment</i>	6						
<i>Backend inicijalizacija</i>							2
<i>Postavljanje, revizija dokumentacije i dr.</i>		5					
<i>funkcionalnosti frontenda</i>			4			7	
<i>Sastanci</i>	13	13.5	11	12.5	12.5	13.5	12.5

## Dijagrami pregleda promjena