

Programsko inženjerstvo

Ak. god. 2023./2024.

Ozdravi – olakšava život kad imate
bolesnu djecu

Dokumentacija, Rev. 1

Grupa: *Proggy i Žohari*

Voditelj: *Jan Komericcki*

Datum predaje: 17. studenog. 2023.

Nastavnik: *Ivana Lulić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
2.1 Korisnički zahtjevi	5
2.2 Opis korištenja aplikacije	6
2.3 Postojeća slična rješenja	8
2.4 Opseg i prilagodljivost projektnog zadatka	9
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Dijagrami obrazaca uporabe	29
3.1.3 Sekvencijski dijagrami	31
3.2 Ostali zahtjevi	35
4 Arhitektura i dizajn sustava	36
4.1 Baza podataka	38
4.1.1 Opis tablica	38
4.1.2 Dijagrami baze podataka	44
4.2 Dijagram razreda	45
4.3 Dijagram stanja	49
4.4 Dijagram aktivnosti	50
4.5 Dijagram komponenti	52
5 Implementacija i korisničko sučelje	54
5.1 Korištene tehnologije i alati	54
5.2 Ispitivanje programskog rješenja	56
5.2.1 Ispitivanje komponenti	56
5.2.2 Ispitivanje sustava	60
5.3 Dijagram razmještaja	61
5.4 Upute za puštanje u pogon	62

5.4.1	<i>Deploy</i> na Render - Backend	62
5.4.2	<i>Deploy</i> na Render - Frontend	65
6	Zaključak i budući rad	66
	Popis literature	68
	Indeks slika i dijagrama	70
	Dodatak: Prikaz aktivnosti grupe	71

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Napravljen predložak.	J.K.	18.10.2023.
0.2.1	Napisan dio opisa projektnog zadatka i funkcionalni zahtjevi	J.K.	26.10.2023.
0.2.2	Napisani <i>use cases</i> za neregistriranog korisnika i roditelja.	J.K.	28.10.2023.
0.2.3	Završeni <i>use cases</i> i završen Opis projektnog zadatka	J.K.	2.11.2023.
0.3.1	Napisan opis baze podataka, ispravljene greške u <i>use cases</i>	J.K.	11.11.2023.
0.3.2	Napisan opis arhitekture sustava, uneseni dijagrami za istog i za bazu podataka, popravljene sitne pogreške	J.K.	14.11.2023.
0.4.1	Dodana odstupanja zbog greške u sustavu (nedovršeno), napravljeni i dodani UML dijagrami za <i>use cases</i>	J.K.	15.11.2023.
0.4.2	Dovršeni <i>use cases</i> , proširena baza podataka, napravljeni sekvencijski dijagrami	J.K.	16.11.2023.
0.4.3	Dodan posljednji sekvencijski dijagram, dodani dijagrami razreda i opisi	J.K.	17.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	17.11.2023.
1.1	Ispravljene pogreške iz prve verzije dokumentacije	J.K.	13.1.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.2	Napravljen UML dijagram stanja i opis, dodane korištene tehnologije	J.K.	14.1.2024.
1.3	Napravljen UML dijagram aktivnosti i opis	J.K.	15.1.2024.
1.4	Napravljen UML dijagram razmještaja i opis, uređena literatura	J.K.	16.1.2024.
1.5	Napravljen UML dijagram komponenti i opis, dodano poglavlje o ispitivanju komponenti, napisane upute za <i>deploy</i>	J.K.	18.1.2024.

2. Opis projektnog zadatka

Cilj ovog projekta razviti je programsku podršku za web aplikaciju *Ozdravi* koja će korisnicima omogućiti olakšanu komunikaciju s pedijatrom i liječnikom obiteljske medicine, te lakši pregled podataka o pregledima sebe i svojeg djeteta. Uz to, aplikacija će automatizirati slanje preporuka za bolovanje i ispričnica, uvelike štедеći vrijeme roditeljima koji zbog toga neće morati naknadno ići po imenovane potvrde.

Do sada, kada bi dijete obolilo, roditelji bi trebali gubiti mnogo vremena u komunikaciji s pedijatrom, te kontinuiranim preuzimanjem i prenošenjem raznih potvrda, bilo ispričnica ili preporuka i doznaka za bolovanje. Naum ove aplikacije jest automatizacija slanja tih potvrda, te olakšani pregled medicinskih podataka djece.

2.1 Korisnički zahtjevi

Korisnički zahtjevi su sljedeći:

- mogućnost samostalne registracije i prijave roditelja
- roditeljski pristup profilima sebe i sve svoje djece, u kojima je omogućen pristup pripadnim medicinskim kartonima
- primanje obavijesti od pedijatra vezano uz pojedino dijete
- mogućnost učitavanja privatnih nalaza i primanja povratnih informacija od pedijatra
- pedijatri/liječnici moraju imati mogućnost unosa novog pacijenta
- pedijatri/liječnici obiteljske medicine imaju pristup popisu svih svojih pacijenata, njihovim informacijama i mogućnost upisivanja novog pregleda
- pedijatri moraju imati mogućnost generiranja i automatskog slanja ispričnica i preporuka za bolovanje
- pedijatri i liječnici moraju imati mogućnost pisanja uputnica za specijalističke preglede, te slanje potencijalnih lokacija za iste roditeljima
- liječnici obiteljske medicine moraju imati mogućnost potvrđivanja preporuka za bolovanje i slanja doznaka o istom

- administratori moraju imati mogućnost brisanja i uređivanja korisničkih računa, stvaranja računa liječnika i pedijatra, te stvaranja i povezivanja profila djece s roditeljem

Analizom korisničkih zahtjeva, razvojni tim došao je do zaključka da je stvaranje profila djece od strane administratora nepraktično, te da postoji efikasnije rješenje. Ono je stvaranje profila djece od strane pedijatra pri prvoj pregledu, s obzirom na to da pedijatri imaju pristup osobnim podacima djece.

2.2 Opis korištenja aplikacije

Prilikom pokretanja aplikacije neprijavljenom korisniku prikazat će se naslovna web stranica s opisom funkcionalnosti aplikacije, katalogom usluga, te opcijama za registraciju ili prijavu.

Prilikom registracije korisnik unosi sljedeće podatke:

- korisničko ime
- email adresa
- lozinka
- ime
- prezime
- OIB
- mjesto prebivališta
- poštanski broj prebivališta
- email adresa poslodavca
- broj telefona

Registracijom u sustav, korisnik dobiva prava roditelja. Naknadna promjena prava je nemoguća. Ostale uloge se ne registriraju, već su dodane u sustav od strane administratora.

Roditelj prijavom u sustav koristeći svoje korisničko ime i lozinku dolazi do uvodne stranice za roditelje. Na uvodnoj stranici nalaze se obavijesti od liječnika obiteljske medicine i pedijatra, te izbornik mogućih profila kojima roditelj može pristupiti. Roditelj ima omogućen pristup svojem profilu i profilima svoje djece. Nakon biranja profila, otvara se nova stranica na kojoj se nalazi izbornik mogućnosti, te prostor za prikaz.

Unutar profila djeteta, izbornik sadrži sljedeće opcije:

- Obavijesti - otvara prikaz svih obavijesti od pedijatra zaduženog za navedeno dijete
- Povijest liječničkih pregleda - otvara svojevrstan medicinski karton djeteta
- Generirane ispričnice - otvara pregled generiranih ispričnica koje pedijatar izdaje djetetu
- Nalazi iz laboratorija - otvara pregled nalaza djeteta koji su naknadno dobiveni iz laboratorija
- Specijalistički pregledi - otvara stranicu na kojoj pedijatar šalje potvrdu za specijalistički pregled i lokacije na kojima je moguće izvršiti navedeni pregled
- Učitavanje nalaza - opcija koja omogućuje *upload* nalaza koji je dobiven pri eventualnom pregledu kod privatnika

Unutar profila roditelja, izbornik sadrži sljedeće opcije:

- Obavijesti - otvara prikaz svih obavijesti od liječnika zaduženog za roditelja
- Povijest liječničkih pregleda - otvara svojevrstan medicinski karton roditelja
- Potvrđena bolovanja - otvara pregled bolovanja koja su odobrena od strane liječnika
- Specijalistički pregledi - otvara stranicu na kojoj liječnik šalje potvrdu za specijalistički pregled i lokacije na kojima je moguće izvršiti navedeni pregled
- Učitavanje nalaza - opcija koja omogućuje *upload* nalaza koji je dobiven pri eventualnom pregledu kod privatnika

Osim roditelja, postoje još tri vrste korisnika:

- pedijatar
- liječnik obiteljske medicine
- administrator

Pedijatar/liječnik obiteljske medicine prijavom u sustav, koristeći korisničko ime i lozinku, ulazi na stranicu na kojoj je otvoren popis pacijenata. Moguć je odabir pojedinog pacijenta, čime se otvara profil tog pacijenta. Na profilu su vidljivi osnovni osobni podaci, te nekoliko gumba s opcijama. Te opcije su:

- Medicinski karton - otvara medicinski karton odabranog pacijenta
- Novi pregled - otvara sučelje za unos novog pregleda odabranog pacijenta
- Generiranje ispričnice (opcija samo za pedijatra) - otvara sučelje za generiranje ispričnice za odabranog pacijenta

- Generiranje preporuke za bolovanje (opcija samo za pedijatra) - otvara sučelje za generiranje preporuke za bolovanje za roditelja odabranog pacijenta
- Nalazi iz privatnih ustanova - otvara popis nalaza iz privatnih ustanova za odabranog pacijenta
- Specijalistički pregled - otvara sučelje za naručivanje odabranog pacijenta na specijalistički pregled
- Potvrda bolovanja (opcija samo za liječnika obiteljske medicine) - otvara sučelje za potvrdu preporuka za bolovanje/otvaranje bolovanja

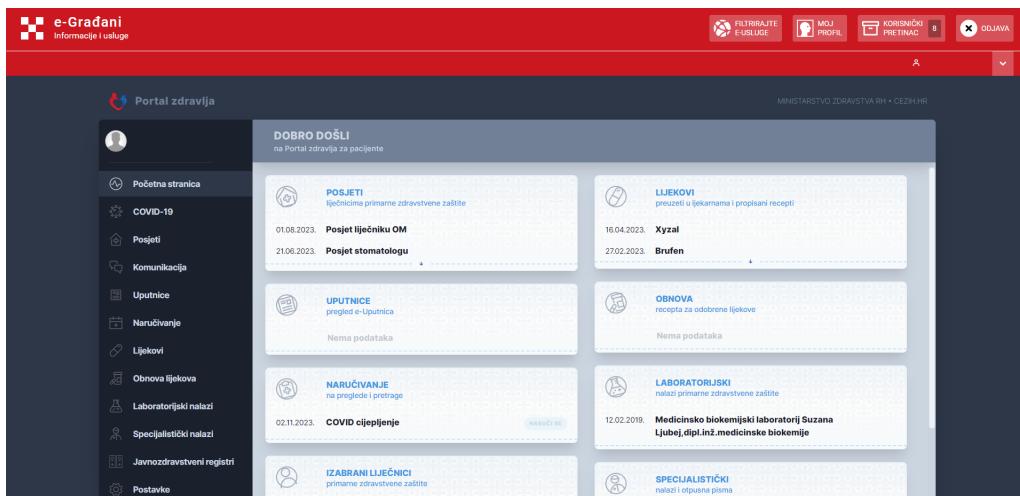
S iznad glavnog sučelja nalazi se izbornik sa dvije opcije:

- Popis pacijenata - otvara već spomenuti popis pacijenata
- Novi pacijent - otvara sučelje za unos novog pacijenta prilikom prvog pregleda
- Preporuke za bolovanje (opcija samo za LOM-a) - otvara sučelje za potvrđivanje preporuka za bolovanje koje dobije od pedijatra

Administrator sustava posjeduje najveće ovlasti od svih vrsta korisnika. On ima pristup bazi s popisom svih korisnika. Administrator ima mogućnost te korisnike brisati i mijenjati im podatke i međusobne veze (npr. promijeniti zaduženog pedijatra djetetu). Uz to, administrator je zadužen za stvaranje korisničkih računa za pedijatre i liječnike obiteljske medicine.

2.3 Postojeća slična rješenja

Već postoji web-aplikacija koja vrši sličan posao kao ovaj projekt. Radi se o portalu na platformi *e-Građani*, pod nazivom *Portal zdravlja*. Ipak, to rješenje vrlo je općenito. Imenovana aplikacija ne specijalizira se za dječju medicinsku brigu, već je platforma gdje punoljetni građani imaju pristup raznim uslugama hrvatskog zdravstva. Na primjer, građani imaju mogućnost naručivanja na pregledе i cijepljenje, pregleda svojih posjeta liječniku, te pristup popisu lijekova koji su im pripisani, te još mnoge mogućnosti. Jedna sličnost koju valja primijetiti jest sekcija za specijalističke pregledе. Ipak, na *Portalu zdravlja* može se pristupiti samo prijašnjim nalazima, i ne nudi se mogućnost prikaza lokacija za pregled na karti.



Slika 2.1: Izgled početnog izbornika platforme *Portal zdravlja*

2.4 Opseg i prilagodljivost projektnog zadatka

Opisan rad aplikacije ostvaren je korištenjem nekoliko alata. Najprije, za razvoj *backend-a* koristili smo alat *Spring Boot*, gdje smo programirali u programskom jeziku Java. Za bazu podataka koristili smo *H2* koji je također prilagođen radu u Javi, a za razvoj *frontend-a* koristili smo *React*, biblioteku programskog jezika *JavaScript*.

Aplikacija se može naknadno proširiti za širu uporabu, na primjer dodavanjem funkcionalnosti za naručivanje lijekova ili dodavanjem *chat* funkcije za komunikaciju sa zdravstvenim djelatnicima.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik (naručitelj)
2. Neregistrirani korisnik
3. Roditelji
4. Liječnici i pedijatri
5. Administrator
6. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) pregledati katalog usluga aplikacije
 - (b) vidjeti koje sve vrste korisnika aplikacija podržava
 - (c) registrirati se u sustav, koristeći korisničko ime, email adresu, lozinku, ime, prezime i OIB, čime stvara osobni korisnički račun
 - (d) prijaviti se u sustav putem korisničkog imena i lozinke
2. Roditelj (inicijator) može:
 - (a) pristupiti profilima svoje djece ili svojem profilu
 - (b) otvarati i čitati obavijesti vezane uz svaki profil, poslane od liječnika ili pedijatra
 - (c) vidjeti svoj ili djetetov medicinski karton te povijest pregleda
 - (d) pristupiti ispričnicama generiranim za pojedino dijete te potvrdoma o bolovanju
 - (e) pristupiti naknadnim nalazima laboratorijskih pretraga
 - (f) primiti informacije i narudžbe na specijalističke preglede, te vidjeti lokacije na kojima je moguće izvesti navedeni pregled
 - (g) učitati (*uploadati*) nalaze koje dobije prilikom pregleda u privatnoj ordinaciji

3. Pedijatar (inicijator) može:

- (a) pristupiti profilima djece kojima je dedicirani pedijatar
- (b) pristupiti liječničkim kartonima svih svojih pacijenata
- (c) prijaviti novo dijete prilikom prvog pregleda, koristeći osobne podatke djeteta (ime, prezime, OIB, datum rođenja), te OIB roditelja
- (d) za svakog pacijenta unijeti novi pregled
- (e) za svakog pacijenta generirati ispričnicu
- (f) za roditelje djece preporučiti bolovanje
- (g) naručiti dijete na specijalistički pregled, te preporučiti lokacije za izvedbu istog

4. Liječnik obiteljske medicine(inicijator) može:

- (a) pristupiti profilima roditelja kojima je liječnik
- (b) pristupiti liječničkim kartonima svih svojih pacijenata
- (c) za svakog pacijenta unijeti novi pregled
- (d) potvrditi ili odbiti preporuke za bolovanje roditelja
- (e) naručiti roditelja na specijalistički pregled, te preporučiti lokacije za izvedbu istog

5. Administrator(inicijator) može:

- (a) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (b) brisati korisnike
- (c) mijenjati veze između korisnika, npr. premjestiti dijete s profila jednog roditelja na profil drugog
- (d) stvarati profile liječnika i pedijatara

6. Baza podataka(sudionik):

- (a) pohranjuje sve podatke o korisnicima, njihove međusobne povezanosti i uloge
- (b) pohranjuje liječničke kartone i dijagnoze roditelja i djece

3.1.1 Obrasci uporabe

UC1 - Pregledaj usluge aplikacije

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Upoznati se s mogućnostima aplikacije
- **Sudionici:-**
- **Preduvjet:-**
- **Opis osnovnog tijeka:**
 1. Korisnik otvara web stranicu aplikacije
 2. Korisnik pregledava sadržaj stranice
- **Opis mogućih odstupanja:**
 - 1.a Neuspjeh učitavanja stranice, zbog greške u pristupu serverima
 1. Korisnika se obavještava o neuspjehu učitavanja stranice putem ispisane poruke
 2. Korisnik provjerava svoj pristup internetu, te pokušava ponovno

UC2-Registriraj se

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun roditelja za prijavu u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Otvorena početna stranica aplikacije
- **Opis osnovnog tijeka:**
 1. Na početnoj stranici aplikacije korisnik odabire opciju "Registriraj se"
 2. Sustav otvara ekran registracije
 3. Korisnik unosi osobne i korisničke podatke, te potvrđuje da se želi registrirati
 4. Sustav ažurira bazu podataka i vraća korisnika na početnu stranicu
 5. Korisnik prima vizualnu obavijest o registraciji
- **Opis mogućih odstupanja:**
 - 3.a Odabir već korištenog korisničkog imena, email-a ili OIB-a/nepravilan format unosa
 1. Korisnika se obavještava o neuspjehu registracije, i vraća ga se na stranicu za registraciju
 2. Korisnik ispravlja nepravilno unesene podatke, te ponovno potvrđuje unos
 - 3.b Korisnik odabire opciju "Odustani"

1. Sustav korisnika vraća na početnu stranicu (korak 1)

UC3 - Prijavi se u sustav

- **Glavni sudionik:** Roditelj/Pedijatar/Liječnik obiteljske medicine
- **Cilj:** Prijava u sustav čime se pristupa korisničkom profilu
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija roditelja, postojanje profila pedijatra i liječnika u sustavu
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prijava" na početnoj stranici aplikacije
 2. Sustav otvara ekran prijave
 3. Korisnik unosi svoje korisničko ime i lozinku, te odabire opciju "Prijavi se"
 4. Nakon provjere unesenih podataka u bazi podataka, sustav korisniku otvara početna stranica profila
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odabire opciju "Odustani"
 1. Sustav korisnika vraća na početnu stranicu (korak 1)
 - 4.a Nepravilan unos imena ili lozinke
 1. Korisnik dobiva informaciju o neuspjeloj prijavi, i vraća ga se na stranicu za prijavu
 2. Korisnik ispravlja nepravilno unesene podatke

UC4 - Pregledaj obavijesti (1)

- **Glavni sudionik:** Roditelj
- **Cilj:** Informiranje roditelja o svim novostima vezanim uz profile njih i djece
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava roditelja u sustav
- **Opis osnovnog tijeka:**
 1. Sustav uzima popis obavijesti iz baze podataka i prikazuje korisniku
 2. Roditelj čita popis obavijesti (vezanih uz sve profile računa roditelja, dakle osobni profil i sva djeca) koji mu se prikazuje na lijevoj strani sučelja nakon uspješne prijave
- **Opis mogućih odstupanja:**
 - 1.a Greška pri pristupu obavijestima u bazi podataka

1. Roditelja se obavještava o neuspjehu dohvaćanja obavijesti, putem ispisane poruke
2. Roditelja se moli da pokuša kasnije, ili da kontaktira administratora

UC5 - Odaberi profil

- **Glavni sudsionik:** Roditelj
- **Cilj:** Pristup podacima i funkcijama pojedinog profila roditelja ili djeteta
- **Sudsionici:** Baza podataka
- **Preduvjet:** Prijava roditelja u sustav
- **Opis osnovnog tijeka:**
 1. Sustav filtrira popis profila vezanih uz roditeljski račun iz baze podataka, te ih prikazuje
 2. Roditelj odabire jedan od profila koji mu se prikazuju na desnoj strani sučelja nakon uspješne prijave
 3. Sustav pronađe podatke o kliknutom profilu, te otvara stranicu odabranog profila
- **Opis mogućih odstupanja:**
 - 1.a Greška u sustavu pri pristupu odabranom profilu
 1. Roditelja se obavještava o neuspjehu dohvaćanja podataka o profilu, putem ispisane poruke
 2. Roditelja se moli da pokuša kasnije, ili da kontaktira administratora

UC6 - Pregledaj obavijesti (2)

- **Glavni sudsionik:** Roditelj
- **Cilj:** Informiranje roditelja o novostima vezanim uz odabrani profil
- **Sudsionici:** Baza podataka
- **Preduvjet:** Odabir profila djeteta ili roditelja
- **Opis osnovnog tijeka:**
 1. Sustav uzima popis obavijesti iz baze podataka te ih prikazuje
 2. Roditelj čita popis obavijesti (isključivo vezanih za odabrani profil) koji mu se prikazuje na desnoj strani sučelja nakon uspješne odabira profila
- **Opis mogućih odstupanja:**
 - 1.a Greška pri pristupu obavijestima u bazi podataka
 1. Roditelja se obavještava o neuspjehu dohvaćanja obavijesti, putem ispisane poruke

2. Roditelja se moli da pokuša kasnije, ili da kontaktira administratora

UC7 - Pregledaj povijest liječničkih pregleda

- **Glavni sudionik:** Roditelj
- **Cilj:** Pregled povijesti liječničkih pregleda osobe čijem je profilu roditelj pristupio
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabir profila djeteta ili roditelja
- **Opis osnovnog tijeka:**
 1. Na izborniku s lijeve strane sučelja roditelj odabire opciju "Povijest liječničkih pregleda"
 2. Sustav pristupa bazi podataka, iz koje prenosi popis obavljenih liječničkih pregleda
 3. Otvara se prikaz povijesti liječničkih pregleda na desnoj strani sučelja pored izbornika
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu pregledima u bazi podataka
 1. Roditelja se obavještava o neuspjehu dohvaćanja popisa pregleda, putem ispisane poruke
 2. Roditelja se moli da pokuša kasnije, ili da kontaktira administratora

UC8 - Pregledaj generirane ispričnice

- **Glavni sudionik:** Roditelj
- **Cilj:** Pregled generiranih ispričnica za dijete čijem je profilu roditelj pristupio
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabir profila djeteta
- **Opis osnovnog tijeka:**
 1. Na izborniku s lijeve strane sučelja roditelj odabire opciju "Generirane ispričnice"
 2. Sustav pristupa bazi podataka, iz koje prenosi popis generiranih ispričnica
 3. Otvara se prikaz generiranih ispričnica na desnoj strani sučelja pored izbornika
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu ispričnicama u bazi podataka
 1. Roditelja se obavještava o neuspjehu dohvaćanja popisa ispričnica, putem ispisane poruke

2. Roditelja se moli da pokuša kasnije, ili da kontaktira administratora

UC9 - Odjavi se

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjava iz aplikacije
- **Sudionici:** -
- **Preduvjet:** Prijava korisnika
- **Opis osnovnog tijeka:**
 1. Korisnik pritišće na tipku "Odjavi se", kojom se odjavljuje iz aplikacije
- **Opis mogućih odstupanja:**
 - 1.a Greška pri odjavi
 1. Roditelja se obavještava o greški, te ga se moli za ponovni pokušaj

UC10 - Pregledaj specijalističke preglede

- **Glavni sudionik:** Roditelj
- **Cilj:** Prikaz narudžbi na specijalističke preglede i lokacija za iste
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabir profila djeteta ili roditelja
- **Opis osnovnog tijeka:**
 1. Na izborniku s lijeve strane sučelja roditelj odabire opciju "Specijalistički pregledi"
 2. Sustav pristupa bazi podataka, iz koje prenosi podatke o specijalističkim pregledima, i lokacije za iste
 3. Otvara se prikaz popisa specijalističkih pregleda koji se moraju izvršiti, na desnoj strani sučelja pored izbornika
 4. Roditelj odabire jedan
 5. Otvara se prikaz s nazivom pregleda, uputama od pedijatra ili liječnika te karta s lokacijama na kojima je moguće izvesti imenovani pregled
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu spec. pregledima u bazi podataka
 1. Roditelja se obavještava o neuspjehu dohvaćanja popisa pregleda, putem ispisane poruke
 2. Roditelja se moli da pokuša kasnije, ili da kontaktira administratora

UC11 - Učitaj nalaz

- **Glavni sudionik:** Roditelj

- **Cilj:** *Upload* nalaza dobivenog pri privatnom pregledu na sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabir profila djeteta ili roditelja
- **Opis osnovnog tijeka:**
 1. Na izborniku s lijeve strane sučelja roditelj odabire opciju "Učitavanje nalaza"
 2. Otvara se novo sučelje s desne strane izbornika, na kojem se nalazi polje za poruku liječniku/pedijatru, gumb za odabir dokumenta, te gumb "Učitaj"
 3. Roditelj piše poruku, odabire dokument i stišće gumb "Učitaj", čime se vrši *upload* dokumenta i poruke
 4. Sustav dokument sprema u bazu podataka, vraća korisnika na početni prikaz (korak 2), i obavještava ga o uspjehu *uploada*
- **Opis mogućih odstupanja:**
 - 3.a Neuspjeh učitavanja dokumenta, zbog krivog formata dokumenta
 1. U slučaju pogrešnog formata dokumenta, ispisati će se upozorenje korisniku
 2. Korisnik pokušava učitati drugu vrstu dokumenta
 - 3.b Neuspjeh učitavanja dokumenta, zbog greške u pristupu serverima
 1. U slučaju pogreške u mreži, ispisuje se poruka korisniku da provjeri svoju mrežnu povezanost

UC12 - Pregledaj pacijente u sustavu

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Mogućnost pregleda popisa svih pacijenata koji su prijavljeni kod liječnika/pedijatra
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao liječnik/pedijatar
- **Opis osnovnog tijeka:**
 1. Sustav nakon prijave automatski pristupa popisu pacijenata u bazi podataka
 2. Nakon prijave odmah se otvara sučelje s popisom svih pacijenata
- **Opis mogućih odstupanja:**
 - 1.a Greška pri pristupu pacijentima u bazi podataka
 1. L.O.M./Pedijatra se obavještava o neuspjehu dohvaćanja popisa pacijenata, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora
 - 2.a Otvoren je neki drugi prikaz
 1. Klikom na opciju na izborniku s lijeve strane sučelja korisnik pristupa popisu pacijenata
 2. Sustav otvara prikaz

UC13 - Pregledaj podatke pacijenta

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Otvaranje profila pacijenta na pregled
- **Sudionici:** Baza podataka
- **Preduvjet:** Otvoren pregled popisa svih pacijenata
- **Opis osnovnog tijeka:**
 1. Liječnik/pedijatar unutar popisa odabire pacijenta
 2. Sustav pristupa bazi podataka, te iz nje prenosi podatke o odabranom pacijentu
 3. Otvara se prikaz profila tog pacijenta s osobnim podacima, te nekoliko opcija navedenih kasnije
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu podacima pacijenta u bazi podataka
 1. L.O.M./Pedijatra se obavještava o neuspjehu dohvaćanja podataka pacijenta, putem ispisane poruke

2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC14 - Otvori karton pacijenta

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Otvaranje medicinskog kartona pacijenta na pregled
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**
 1. Liječnik/pedijatar na profilu odabranog pacijenta pritišće opciju za pregled kartona
 2. Sustav pristupa bazi podataka, te iz nje prenosi medicinski karton odabranog pacijenta
 3. Otvara se prikaz medicinskog kartona pacijenta
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu podacima pacijenta u bazi podataka
 1. L.O.M./Pedijatra se obavještava o neuspjehu dohvaćanja kartona pacijenta, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC15 - Unesi novi pregled

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Unos novog pregleda za odabranog pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**
 1. Liječnik/pedijatar na profilu odabranog pacijenta pritišće opciju za unos novog pregleda
 2. Otvara se prikaz za unos podataka o pregledu, s poljima vezanim uz podatke pacijenta već ispunjenima
 3. Liječnik/pedijatar unosi podatke o pregledu, te pritišće opciju za unos
 4. Sustav navedene podatke o pregledu koristi da bi stvorio objekt pregleda, te njega sprema u bazu podataka i vraća liječnika/pedijatra na profil pacijenta (korak 1)
- **Opis mogućih odstupanja:**

- 3.a Liječnik/Pedijatar odabire opciju "Odustani"
 1. Sustav korisnika vraća na početnu stranicu (korak 1)
- 4.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC16 - Generiraj ispričnicu

- **Glavni sudionik:** Pedijatar
- **Cilj:** Generiranje i slanje ispričnice za pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**
 1. Pedijatar na profilu pacijenta bira opciju za generiranje ispričnice
 2. Sustav pristupa bazi podataka, te iz nje prenosi podatke o odabranom pacijentu
 3. Otvara se sučelje s unesenim podacima pacijenta, te poljima za unos imena bolesti i vremena trajanja ispričnice
 4. Pedijatar unosi podatke i pritišće opciju za slanje
 5. Sustav stvara novi objekt ispričnice, te ga sprema u bazu podataka i vraća pedijatra na profil pacijenta (korak 1)
- **Opis mogućih odstupanja:**
 - 2/5.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora
 - 4.a Liječnik/pedijatar odabire opciju "Odustani"
 1. Sustav liječnika/pedijatra vraća na prikaz profila pacijenta(korak 1)

UC17 - Generiraj preporuku o bolovanju

- **Glavni sudionik:** Pedijatar
- **Cilj:** Generiranje i slanje preporuke o bolovanju za roditelja pacijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**

1. Pedijatar na profilu pacijenta bira opciju za generiranje preporuke o bolovanju
 2. Sustav pristupa bazi podataka, u kojoj pronalazi OIB roditelja, kojim dohvaća podatke roditelja pacijenta
 3. Otvara se sučelje s unesenim podacima roditelja pacijenta, te poljima za unos razloga za bolovanjem i vremena trajanja bolovanja
 4. Pedijatar unosi podatke i pritišće opciju za slanje
 5. Sustav stvara novi objekt preporuke o bolovanju kojeg sprema u bazu podataka i vraća pedijatra na profil pacijenta (korak 1)
- **Opis mogućih odstupanja:**
 - 2/5.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora
 - 4.a Liječnik/pedijatar odabire opciju "Odustani"
 1. Sustav liječnika/pedijatra vraća na prikaz profila pacijenta(korak 1)

UC18 - Pošalji obavijest roditelju

- **Glavni sudionik:** Pedijatar
- **Cilj:** Slanje obavijesti roditelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**
 1. Pedijatar na profilu pacijenta bira opciju za slanje obavijesti roditelju pacijenta
 2. Otvara se sučelje s poljima za unos naslova obavijesti, te poljem za unos dužeg teksta/poruke obavijesti
 3. Pedijatar unosi podatke i pritišće opciju za slanje
 4. Sustav pristupa bazi podataka u kojoj pronalazi OIB roditelja, s kojim tvori novi objekt obavijesti koji se šalje i vraća pedijatra na profil pacijenta (korak 1)
- **Opis mogućih odstupanja:**
 - 3.a Liječnik/pedijatar odabire opciju "Odustani"
 1. Sustav liječnika/pedijatra vraća na prikaz profila pacijenta(korak 1)
 - 4.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke

2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC19 - Zakaži specijalistički pregled

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Slanje potvrde i lokacija za specijalistički pregled
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**
 1. Liječnik/pedijatar na profilu pacijenta bira opciju "Specijalistički pregled"
 2. Sustav pristupa bazi podataka iz koje prenosi podatke pacijenta
 3. Otvara se sučelje s unesenim podacima pacijenta, te poljima za unos vrste specijalističkog pregleda i za unos mogućih lokacija na kojima se može izvesti pregled
 4. Liječnik/pedijatar unosi tražene podatke te pritišće opciju za slanje
 5. Sustav stvara novi objekt specijalističkog pregleda i sprema ga u bazu, te vraća liječnika/pedijatra na profil pacijenta (korak 1)
- **Opis mogućih odstupanja:**
 - 2/5.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora
 - 4.a Liječnik/pedijatar odabire opciju "Odustani"
 1. Sustav liječnika/pedijatra vraća na prikaz profila pacijenta(korak 1)

UC20 - Pregledaj nalaze iz privatnih ustanova

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Pregled nalaza iz privatnih ustanova
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran pacijent
- **Opis osnovnog tijeka:**
 1. Liječnik/pedijatar na profilu pacijenta bira opciju "Nalazi iz privatnih ustanova"
 2. Sustav pristupa bazi podataka iz koje prenosi popis nalaza

3. Otvara se prikaz s popisom nalaza, na kojima piše datum te ime i prezime pacijenta
 4. Pritisom na jedan nalaz otvara se detaljan prikaz
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC21 - Pošalji povratnu informaciju

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Slanje povratne informacije vezane uz nalaz iz privatne ustanove
- **Sudionici:** Baza podataka
- **Preduvjet:** Otvoren nalaz iz privatne ustanove
- **Opis osnovnog tijeka:**
 1. Liječnik/pedijatar pritišće na opciju "Povratna informacija" koja se nalazi na dnu nalaza
 2. Otvara se sučelje s poljima za unos naslova obavijesti, te poljem za unos dužeg teksta poruke
 3. Liječnik/pedijatar unosi podatke i pritišće opciju za slanje
 4. Sustav pristupa bazi podataka u kojoj pronalazi OIB roditelja, s kojim tvori novi objekt obavijesti koji se šalje, te vraća liječnika/pedijatra na profil pacijenta
- **Opis mogućih odstupanja:**
 - 3.a Liječnik/pedijatar odabire opciju "Odustani"
 1. Sustav liječnika/pedijatra vraća na prikaz nalaza (korak 1)
 - 4.a Greška pri pristupu bazi podataka
 1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
 2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC22 - Unesi novog pacijenta

- **Glavni sudionik:** Liječnik/pedijatar
- **Cilj:** Unos novog pacijenta pri prvom pregledu
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao liječnik/pedijatar

- **Opis osnovnog tijeka:**

1. Liječnik/pedijatar na izborniku s lijeve strane sučelja bira opciju "Novi pacijent"
2. Otvara se sučelje s poljima za osobne podatke pacijenta (u slučaju roditelja samo OIB, jer roditelj sam registrira ostale podatke), s poljem za OIB roditelja (u slučaju kada je novi pacijent dijete), te s poljem za kontakt(email) škole/vrtića
3. Liječnik/pedijatar unosi podatke u polja
4. Liječnik/pedijatar pritišće gumb "Unesi"
5. Sustav stvara novi objekt pacijenta i spremi ga u bazu podataka, te vraća liječnika/pedijatra na popis pacijenata

- **Opis mogućih odstupanja:**

- 3.a Neispravan OIB

1. U slučaju pogrešno unesenog OIB-a (OIB ne postoji), liječnik/pedijatar dobiva obavijest o pogrešci
2. Liječnik/pedijatar ispravlja grešku

- 5.a Greška pri pristupu bazi podataka

1. L.O.M./Pedijatra se obavještava o grešci, putem ispisane poruke
2. L.O.M./Pedijatra se moli da pokuša kasnije, ili da kontaktira administratora

UC23 - Potvrdi preporuku za bolovanje

- **Glavni sudionik:** Liječnik obiteljske medicine

- **Cilj:** Potvrda preporuka za bolovanje roditelja koje je poslao pedijatar

- **Sudionici:** Baza podataka

- **Preduvjet:** Prijava u sustav kao liječnik/pedijatar

- **Opis osnovnog tijeka:**

1. Liječnik/pedijatar na izborniku s lijeve strane sučelja bira opciju "Preporuke za bolovanje"
2. Sustav pristupa bazi podataka iz koje povlači popis preporuka za bolovanje
3. Otvara se sučelje s popisom preporuka za bolovanje, gdje se pritiskom na jednu otvara detaljan prikaz podataka pacijenta, trajanja bolovanja i opis razloga za preporukom
4. Liječnik pritišće gumb za prihvatanje

- **Opis mogućih odstupanja:**

- 2.a Greška pri pristupu bazi podataka
 1. Liječnika se obavještava o grešci, putem ispisane poruke
 2. Liječnika se moli da pokuša kasnije, ili da kontaktira administratora
- 4.a Greška pri generiranju/slanju doznake o bolovanju poslodavcu
 1. Liječnika se obavještava o grešci, putem ispisane poruke
 2. Liječnika se moli da pokuša kasnije, ili da kontaktira administratora
- 4.b Liječnik odbija preporuku o bolovanju
 1. Sustav preporuku o bolovanju briše iz baze podataka, o čemu obavještava liječnika

UC24 - Stvori profil liječnika/pedijatra

- **Glavni sudsionik:** Administrator
- **Cilj:** Stvaranje profila liječnika obiteljske medicine i pedijatra
- **Sudsionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao administrator
- **Opis osnovnog tijeka:**
 1. Administrator bira opciju za stvaranje novog liječnika ili pedijatra
 2. Otvara se sučelje s poljima za unos osobnih i korisničkih podataka
 3. Administrator ispunjava polja i pritišće opciju "Stvori profil"
 4. Sustav stvara profil te ga sprema u bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Administrator odabire opciju "Odustani"
 1. Sustav administratora preusmjerava na popis pacijenata
 - 4.a Greška pri pristupu bazi podataka
 1. Administratora se obavještava o grešci i ispisuje se *Stack trace* greške

UC25 - Pregledaj korisnike

- **Glavni sudsionik:** Administrator
- **Cilj:** Pregled svih korisnika aplikacije, po kategorijama
- **Sudsionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao administrator
- **Opis osnovnog tijeka:**
 1. Administrator bira opciju za prikaz korisnika
 2. Sustav pristupa bazi podataka te vraća popis svih korisnika
 3. Otvara se lista ispravno registriranih korisnika, po kategorijama
- **Opis mogućih odstupanja:**

2.a Greška pri pristupu bazi podataka

1. Administratora se obavještava o grešci i ispisuje se *Stack trace* greške

UC26 - Odaberi korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled podataka korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Otvoren pregled korisnika
- **Opis osnovnog tijeka:**
 1. Administrator bira profil korisnika kojem želi pristupiti
 2. Sustav pristupa bazi podataka i vraća podatke o korisniku
 3. Otvara se prikaz osobnih, korisničkih i medicinskih (npr. zaduženi pedijatar) podataka nekog korisnika
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu bazi podataka
 1. Administratora se obavještava o grešci i ispisuje se *Stack trace* greške

UC27 - Obriši korisnika

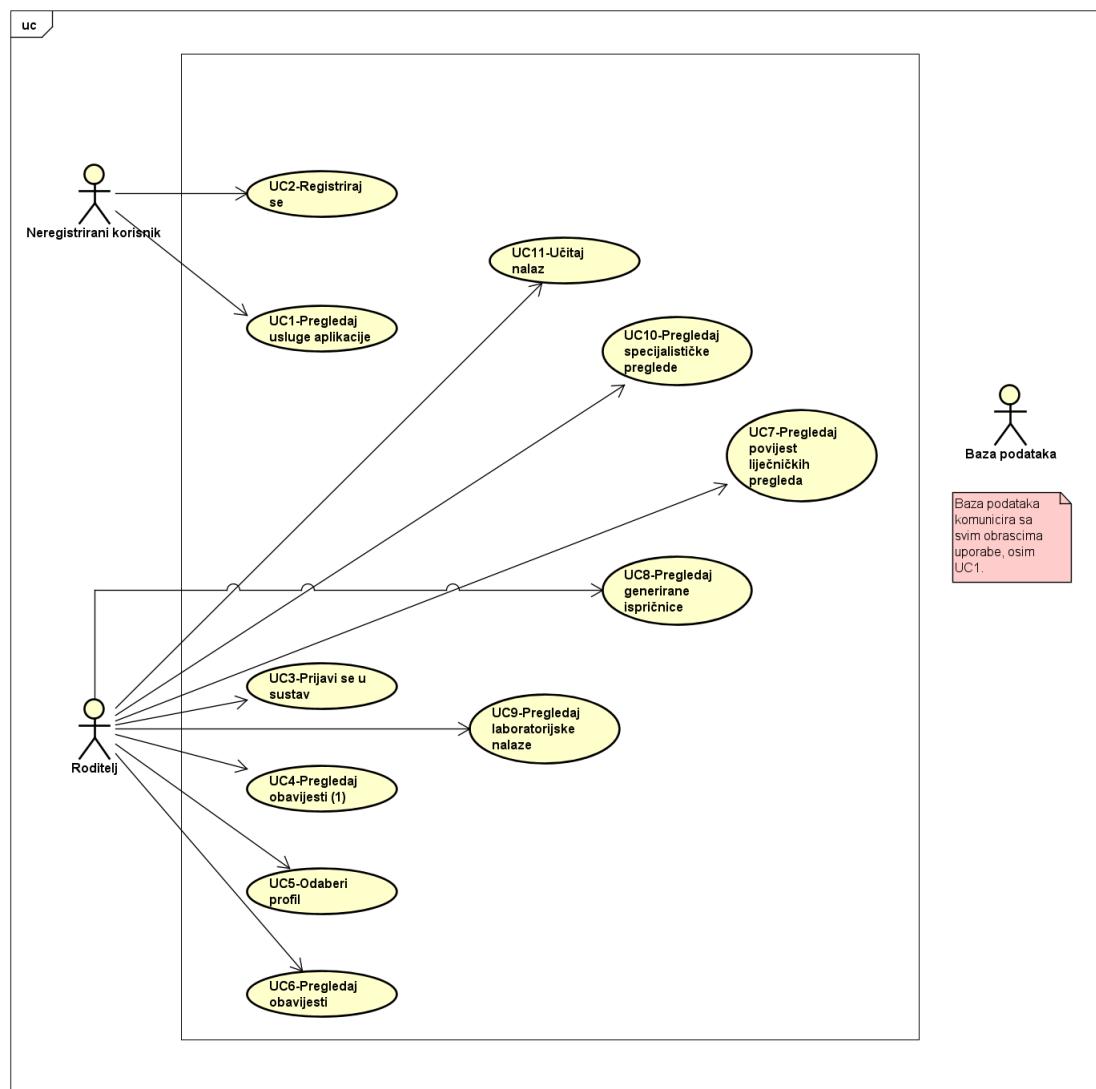
- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje računa korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Odabran profil korisnika
- **Opis osnovnog tijeka:**
 1. Administrator na profilu pritiše opciju "Izbriši profil"
 2. Sustav pristupa bazi podataka u kojoj briše profil
 3. Administratora se vraća na prikaz svih korisnika
- **Opis mogućih odstupanja:**
 - 2.a Greška pri pristupu bazi podataka
 1. Administratora se obavještava o grešci i ispisuje se *Stack trace* greške

UC28 - Promjeni podatke korisnika

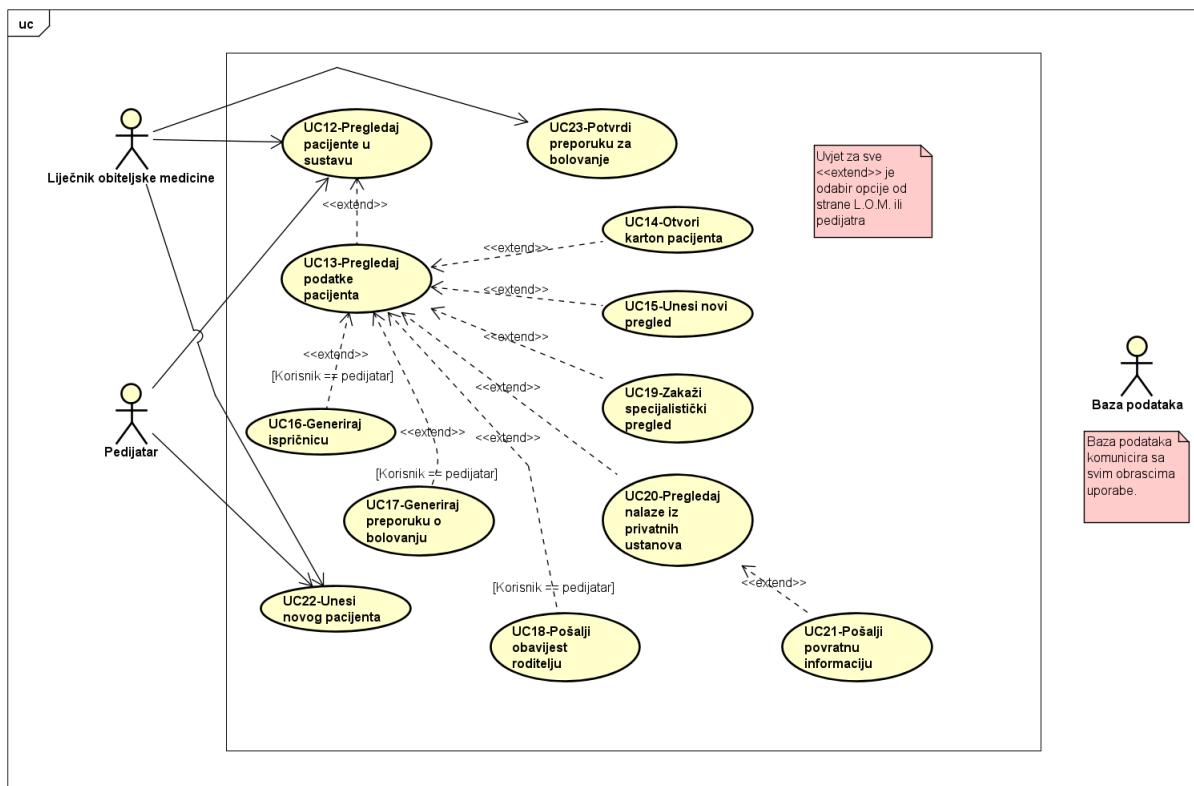
- **Glavni sudionik:** Administrator
- **Cilj:** Promjena osobnih, korisničkih i medicinskih (npr. zaduženi pedijatar) podataka nekog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Otvoren pregled korisnika
- **Opis osnovnog tijeka:**
 1. Administrator na profilu pritiše opciju "Uredi profil"

2. Sustav pristupa bazi podataka gdje dobavlja podatke o korisniku
 3. Otvara se sučelje s ispunjenim poljima za tekst, gdje se nalaze pripadni podaci o korisniku
 4. Administrator uređuje podatke, i na kraju pritišće opciju "Spremi promjene"
 5. Sustav unosi promjene u bazu podataka
- **Opis mogućih odstupanja:**
 - 2/5.a Greška pri pristupu bazi podataka
 1. Administratora se obavještava o grešci i ispisuje se *Stack trace* greške

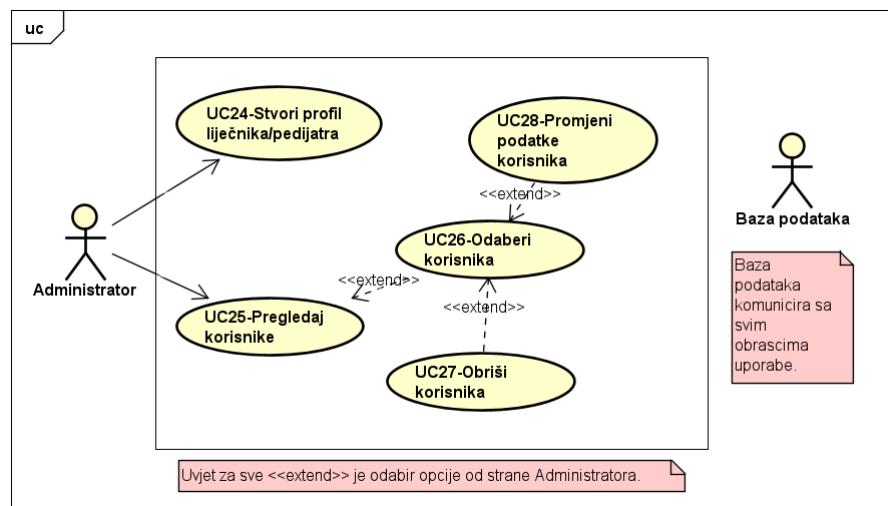
3.1.2 Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, funkcionalnost nereg. korisnika i Roditelja



Slika 3.2: Dijagram obrazaca uporabe, funkcionalnost Pedijatra i L.O.M.

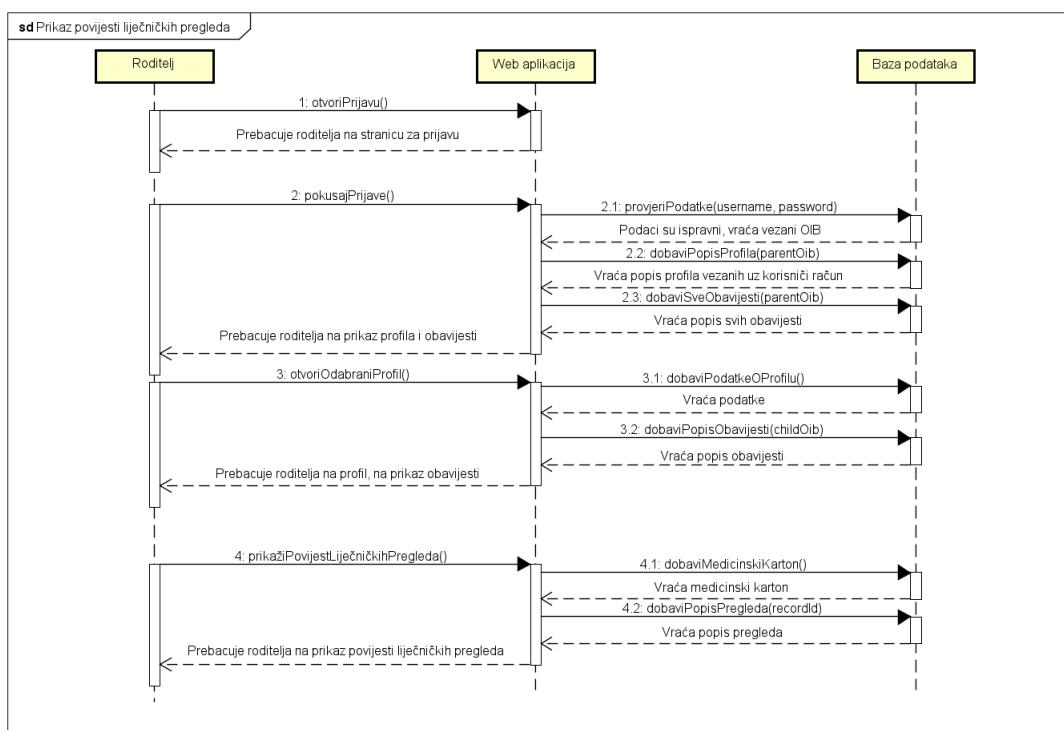


Slika 3.3: Dijagram obrazaca uporabe, funkcionalnost Administratora

3.1.3 Sekvencijski dijagrami

Sekvencijski dijagram - Obrasci uporabe UC3, UC4, UC5, UC6, UC7 (Od prijave do prikaza povijesti liječničkih pregleda)

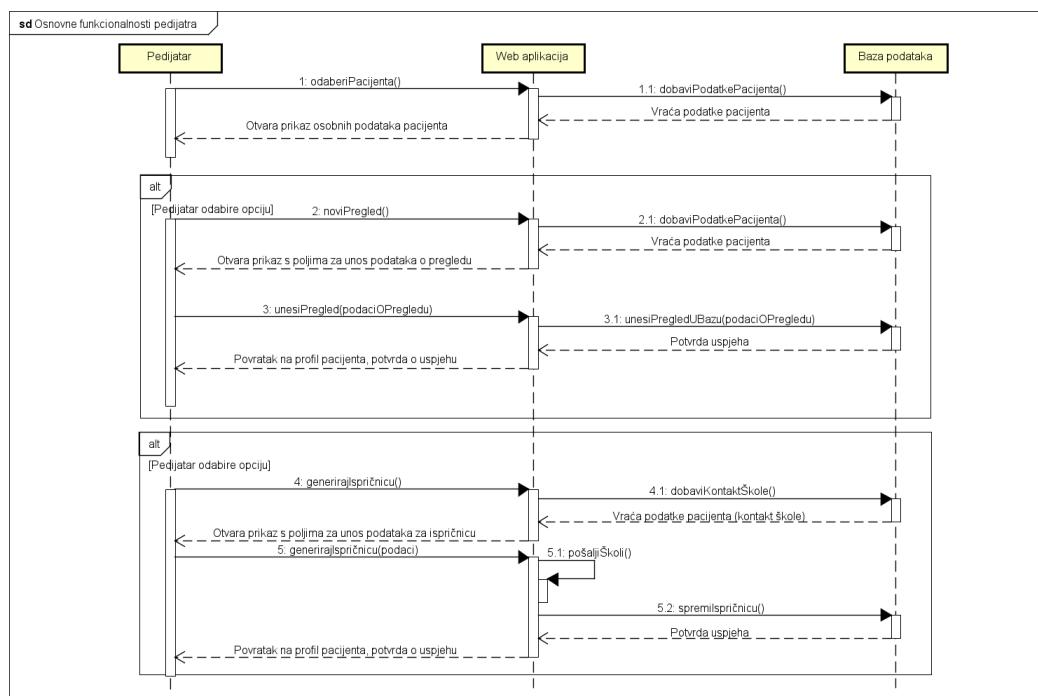
Roditelj prvo šalje zahtjev za prijavom, nakon čega ga sustav prebacuje na ekran za prijavu. Roditelj unosi podatke, koje sustav uzima i provjerava u bazi. Ako su ispravni, sustav dobavlja popis profila vezanih uz korisnički račun i popis svih obavijesti vezanih uz račun roditelja, nakon čega sustav otvara prikaz profila i obavijesti. Kada roditelj odabere jedan od računa, dobavljuju se podaci o tom računu, te popis obavijesti vezanih samo uz taj račun, i zatim se roditelju otvara profil, s prikazanim obavijestima. Roditelj odabire opciju za prikaz povijesti liječničkih pregleda. Sustav nakon toga dobavlja medicinski karton tog profila, nakon čega pomoću identifikatora kartona dobavlja popis obavljenih pregleda. Korisniku se otvara prikaz povijesti liječničkih pregleda.



Slika 3.4: Sekvencijski dijagram prikaza pregleda

Sekvencijski dijagram - Obrasci uporabe UC13, UC15, UC16 (Odabir pacijenta, novi pregled i ispričnica)

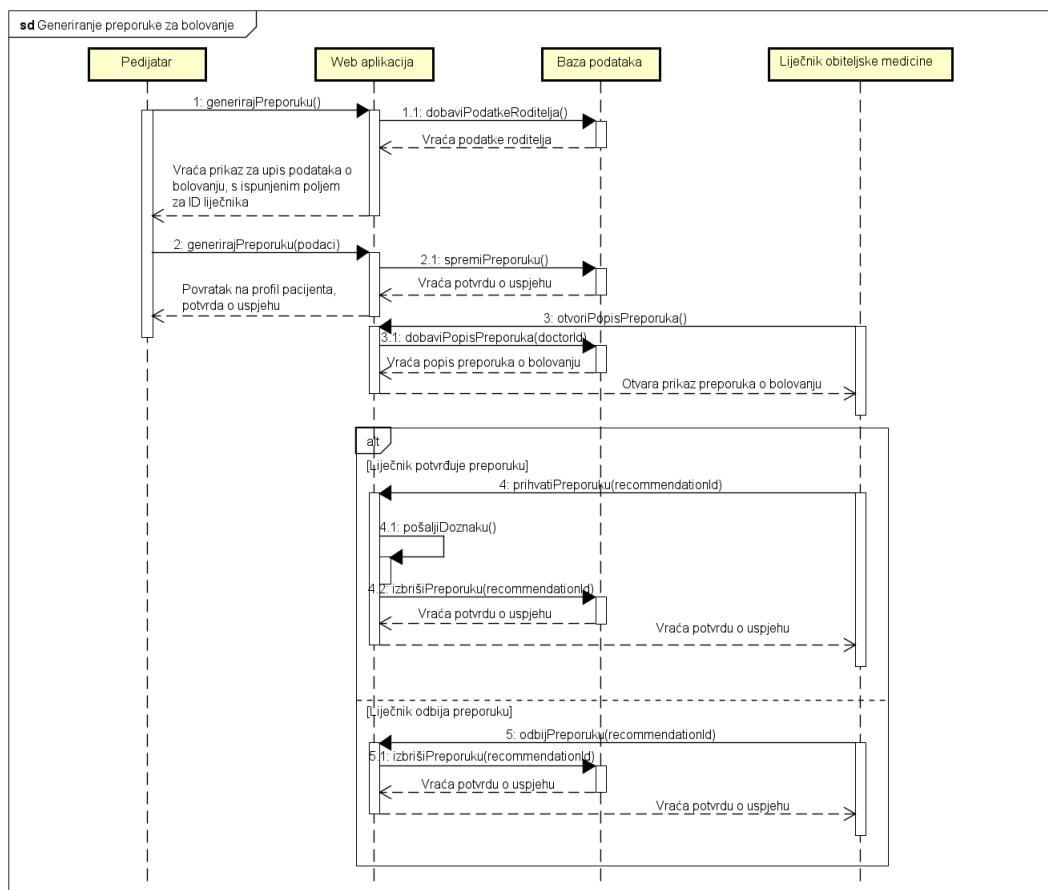
Na popisu pacijenata pedijatar odabire jednog. Sustav pristupa bazi podataka iz koje vraća podatke o pacijentu, koji se zatim koriste pri stvaranju prikaza osobnih podataka pacijenta. Ako na profilu pacijenta pedijatar odabere opciju za unos novog pregleda, sustav dobavlja podatke o pacijentu ponovno iz baze, te ih zapisuje u polja za podatke pacijenta u prikazu koji stvara za pedijatra. U taj prikaz pedijatar, u za to određena polja, unosi podatke o pregledu. Ti podaci, kao i podaci pacijenta, koriste se u stvaranju novog objekta pregleda koji se sprema u bazu podataka. Pedijatra se nakon potvrde uspješnog spremanja u bazu vraća na profil pacijenta. [Bitno! UC13 i UC15 na identičan način funkcioniraju i za L.O.M.] Ako pedijatar odabere opciju za generiranje ispričnice, sustav iz baze prenosi podatak o kontaktu škole, koji se zapisuje u polje u prikazu za stvaranje nove ispričnice. U tom prikazu pedijatar ispunjava ostala polja, primarno opis razloga izdavanja ispričnice. Ispričnica se generira, prilikom čega se prvo šalje školi, a zatim sprema u bazu. Nakon primljene potvrde o uspješnom spremanju u bazu, sustav vraća pedijatra na profil pacijenta.



Slika 3.5: Sekvencijski dijagram osnovnih funkcionalnosti pedijatra

Sekvencijski dijagram - Obrasci uporabe UC17 i UC23 (Generiranje i potvrda preporuke o bolovanju)

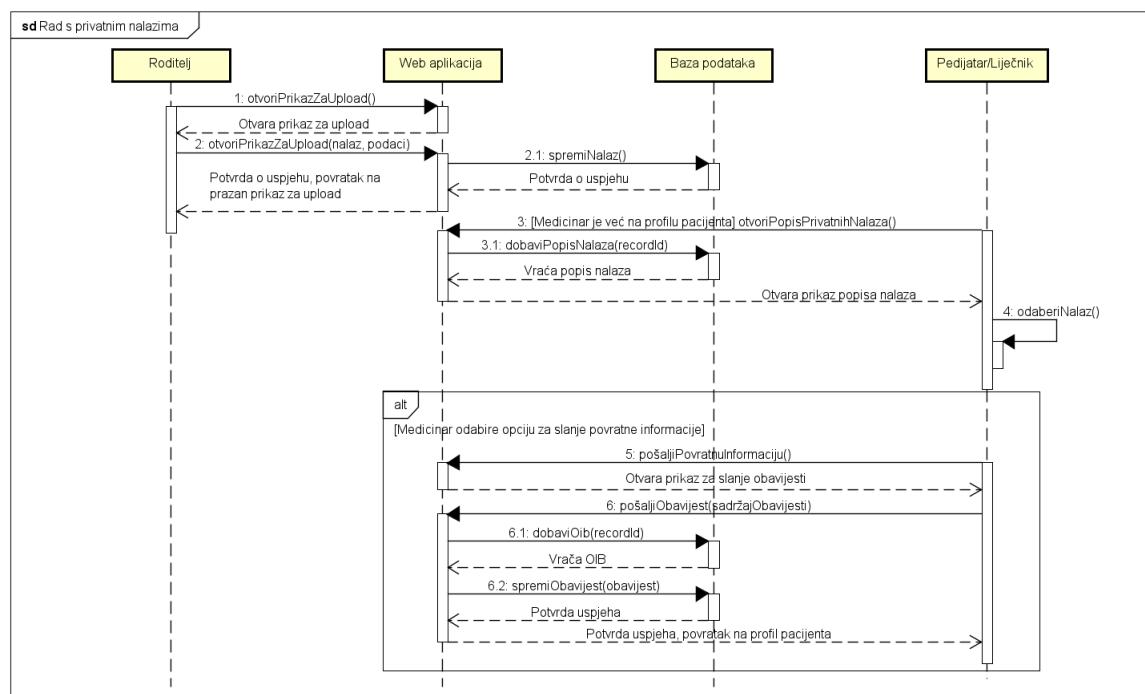
Na profilu pacijenta pedijatar odabire opciju generiraj preporuku za bolovanje. Sustav iz baze podataka dohvaća podatke o roditelju pacijenta. Iz njih se čuvaju email adresa poslodavca i identifikator liječnika. Pedijatu se otvara prikaz u kojem ispunjava podatke o preporuci o bolovanju (npr. razlog i trajanje, te ime i prezime roditelja). Nakon potvrde generiranja preporuke, ona se spremi u bazu podataka. Kada liječnik želi pristupiti preporukama o bolovanju, on odabire opciju na svom izborniku. Kada odabere tu opciju, iz baze se dohvaćaju sve preporuke o bolovanju koje su povezane na doktora putem njegovog identifikatora. Za svaku doktor može odabrati opciju "Potvrdi" ili "Odbij". Prilikom potvrde se generira i šalje doznaka o bolovanju poslodavcu, nakon čega se preporuka briše iz baze. U slučaju odbijanja, preporuka se isto briše iz baze, bez slanja.



Slika 3.6: Sekvencijski dijagram stvaranja i potvrde preporuke o bolovanju

Sekvencijski dijagram - Obrasci uporabe UC11, UC20 i UC21 (Upload nalaza, pregled svih nalaza i optionalno slanje povratne informacije)

Roditelj može na meniju otvorenog profila odabrati opciju za učitavanje nalaza od privatnika. Sustav roditelju otvara prikaz za *upload* s poljima za dodatne informacije. Nakon što roditelj odabire opciju "Učitaj", stvara se novi objekt privatnog nalaza, koji se povezuje na medicinski karton pacijenta, čiji je identifikator jedno od polja u objektu Roditelj/Dijete. Pedijatar/liječnik može na otvorenom profilu pacijenta odabrati opciju za prikaz nalaza od privatnika. U tom slučaju sustav iz baze podataka dobavlja popis, i prikazuje ga pedijatru/liječniku, nakon čega pedijatar/liječnik može otvoriti svaki pojedinačno. Nakon otvaranja detaljnog prikaza nalaza, pedijatar može odabrati opciju za slanje povratne informacije. Nakon toga mu sustav otvara prikaz za slanje obavijesti. Nakon ispunjavanja podataka obavijesti i pritiska opcije za slanje, sustav dobavlja OIB koji je potreban za stvaranje objekta obavijesti iz baze, nakon čega sprema obavijest. [Važno: postupak čitanja obavijesti već je opisan u sekvencijskom dijagramu (Slika 3.4)]



Slika 3.7: Sekvencijski dijagram rada s privatnim nalazima

3.2 Ostali zahtjevi

Aplikacija treba biti prilagođena radu na različitim uređajima, specifično na računalima, tabletima i mobitelima.

Aplikacija treba podržavati više korisnika u isto vrijeme. Maksimalan broj podržanih korisnika nije definiran.

Aplikacija treba biti izvedena kao web aplikacija. Siguran pristup mora biti osiguran korisničkim imenom i lozinkom, no nisu definirani zahtjevi dodatne zaštite.

4. Arhitektura i dizajn sustava

Arhitektura našeg projekta podijeljena je na tri generalna dijela:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Dijagram arhitekture projekta

Web preglednik je program koji omogućava korisnicima pregledavanje web-stranica i multimedijskih sadržaja, te interakciju s njima. Svaki web preglednik također je i svojevrsni prevoditelj, koji programski kod otvorenih stranica interpretira u razumljivi format za korisnike. Kada korisnik koristi web aplikacije, njegovi zahtjevi šalju se putem web preglednika prema web poslužitelju.

Web poslužitelj glavni je dio zadužen za rad aplikacije. S njime korisnik komunicira s web aplikacijom. Ta komunikacija vrši se putem HTTP protokola (engl. *Hyper Text Transfer Protokol*). Web poslužitelj također je zadužen za pokretanje web aplikacije, nakon čega joj šalje korisničke zahtjeve.

Web aplikacija obrađuje korisničke zahtjeve, prilikom čega pristupa bazi podataka. Obrađene podatke vraća korisniku putem poslužitelja, koji zatim navedene podatke prikazuje u web pregledniku.

Prilikom razvoja ovog projekta, koristili smo programski jezik Java i alat *Spring*

Boot za razvoj web aplikacije. Za razvoj izgleda kakav se prikazuje u web pregledniku koristili smo programski jezik Java-Script i Java-Script biblioteku *React*. Razvoj projekta vršio se u razvojnem okruženju *IntelliJ IDEA*.

Za razvoj našeg projekta odlučili smo se ugledati na MVC [Model-Pogled-Nadglednik] (engl. *Model-View-Controller*) koncept arhitekture sustava. Glavna karakteristika imenovanog koncepta jest smanjena međuvisnost korisničkog sučelja i ostatka sustava. Takav koncept omogućava lakše testiranje i daljnji razvoj sustava.

- **Model** - Središnji dio cijelog sustava. Sadrži razrede čiji se objekti/podaci obrađuju. Izravno upravlja s podacima i pravilima aplikacije.
- **Pogled** - Sadrži razrede čiji objekti služe za prikaz podataka, u bilo kojem obliku.
- **Nadglednik** - Sadrži razrede koji upravljaju i rukuju korisničkom interakcijom s pogledom i modelom.

Također smo se u našem kodu odlučili koristiti *Controller-Service-Repository* uzorkom koji je dosta čest u aplikacijama koje koriste *Spring Boot* alat. On se, očito iz naziva, sastoji od tri dijela:

- **Controller** - "Najviši" dio. Služi za bilo kakvu interakciju između okoline aplikacije i njene unutarnje logike, na primjer za primanje i obradu zahtjeva poput *POST* i *GET*.
- **Service** - "Središnji" dio. Unutar njega se implementiraju sve funkcije i sva logika aplikacije. Služi za pisanje kompleksne interakcije s bazom podataka i podacima u njoj.
- **Repository** - "Najniži" dio. Esencijalno se radi o bazi podataka. Služi za spremanje entiteta unutar sustava. Može imati vlastite funkcije, poput *Save*, ali njih može pozivati i *Service*.

4.1 Baza podataka

Za potrebe našeg projekta koristili smo relacijsku bazu podataka napisanu u programskom jeziku Java, pokrenutu *engine-om* H2. Prednost te vrste baze podataka jest jednostavnost modeliranja pravog svijeta na temelju relacija. Zadaća baze podataka u projektu jest pohrana, izmjena i dohvata podataka koji se zatim obrađuju. Baza podataka sastoji se od entiteta:

- Roditelj
- Dijete
- Pedijatar
- Liječnik obiteljske medicine
- Medicinski karton
- Pregled
- Nalaz
- Obavijest
- Specijalistički pregled
- Preporuka o bolovanju
- Ispričnica

4.1.1 Opis tablica

Roditelj - Entitet sadržava sve podatke koje roditelj unosi prilikom registracije. Oni su: OIB, ime i prezime, datum rođenja, korisničko ime, lozinka, broj telefona, e-mail, poštanski broj, mjesto prebivališta, e-mail poslodavca te šifra liječnika. Entitet roditelj u odnosu je *Many-to-One* s entitetom Liječnik obiteljske medicine preko šifre tog liječnika, u odnosu *One-to-Many* sa entitetom Dijete, putem vlastitog OIB-a, te u odnosu *One-to-One* s entitetom Medicinski karton, putem jedinstvenog identifikatora kartona.

Parent/Roditelj		
OIB	CHAR(11)	Jedinstveni identifikator roditelja
nameParent	VARCHAR	Ime roditelja
lastNameParent	VARCHAR	Prezime roditelja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Parent/Roditelj		
dateOfBirthParent	DATE	Datum rođenja roditelja
userNameParent	VARCHAR	Korisničko ime računa
passwordParent	VARCHAR	Zaporka računa
phoneNumberParent	VARCHAR	Broj telefona roditelja
emailParent	VARCHAR	E-mail roditelja
postalCode	INT	Poštanski broj mjesta stanovanja
placeOfResidence	VARCHAR	Mjesto stanovanja roditelja
employerEmail	VARCHAR	E-mail poslodavca roditelja
doctorId	INT	Identifikator L.O.M.-a roditelja
recordId	INT	Identifikator medicinskog kartona roditelja

Dijete - Entitet sadržava sve podatke koje unosi pedijatar prilikom prvog pregleda djeteta. Oni su: OIB, OIB roditelja, šifra pedijatra, ime i prezime, datum rođenja, naziv škole/vrtića te e-mail škole/vrtića. Entitet dijete u odnosu je *Many-to-One* sa entitetom Roditelj, putem roditeljevog OIB-a, u odnosu *Many-to-One* s entitetom Pedijatar putem pedijatrovog ID-a, te u odnosu *One-to-One* s entitetom Medicinski karton, putem jedinstvenog identifikatora kartona.

Child/Dijete		
OIB	CHAR(11)	Jedinstveni identifikator djeteta
nameChild	VARCHAR	Ime djeteta
lastNameChild	VARCHAR	Prezime djeteta
dateOfBirthChild	DATE	Datum rođenja djeteta
educationalInstitution	VARCHAR	Ime škole/vrtića djeteta
emailEduInstitution	VARCHAR	E-mail škole/vrtića
pediatricianId	INT	Jedinstveni identifikator pedijatra

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Child/Dijete		
recordId	INT	Identifikator medicinskog kartona djeteta
parentOib	CHAR(11)	Jedinstveni identifikator roditelja

Pedijatar - Entitet sadržava sve podatke koje unosi administrator prilikom registracije pedijatra u sustav. Oni su: šifra pedijatra, ime i prezime, datum rođenja, korisničko ime, lozinka, broj telefona, e-mail. Entitet pedijatar u odnosu je *One-to-Many* s entitetom Dijete putem vlastite šifre.

Pediatrician/Pedijatar		
pediatricianId	INT	Jedinstveni identifikator pedijatra
namePediatirican	VARCHAR	Ime pedijatra
lastNamePediatirican	VARCHAR	Prezime pedijatra
dateOfBirthPediatirican	DATE	Datum rođenja pedijatra
userNamePediatirican	VARCHAR	Korisničko ime računa
passwordPediatrician	VARCHAR	Zaporka računa
phoneNumPediatrician	VARCHAR	Broj telefona pedijatra
emailPediatrician	VARCHAR	E-mail pedijatra

Liječnik obiteljske medicine - Entitet sadržava sve podatke koje unosi administrator prilikom registracije LOM-a u sustav. Oni su: šifra doktora, ime i prezime, datum rođenja, korisničko ime, lozinka, broj telefona i e-mail. Entitet Liječnik obiteljske medicine u odnosu je *One-to-Many* s entitetom Roditelj putem vlastite šifre.

Doctor/Liječnik obiteljske medicine		
doctorId	INT	Jedinstveni identifikator liječnika
nameDoctor	VARCHAR	Ime liječnika
lastNameDoctor	VARCHAR	Prezime liječnika
dateOfBirthDoctor	DATE	Datum rođenja liječnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Doctor/Liječnik obiteljske medicine		
userNameDoctor	VARCHAR	Korisničko ime računa
passwordDoctor	VARCHAR	Zaporka računa
phoneNumDoctor	VARCHAR	Broj telefona liječnika
emailDoctor	VARCHAR	E-mail liječnika

Medicinski karton - Entitet sadržava podatke za opis medicinskog kartona pojedinog pacijenta (roditelj/dijete). Oni su: šifra kartona, OIB, trenutna dijagnoza i popis alergija. Entitet Medicinski karton u odnosu je *One-to-One* s entitetom Roditelj i s entitetom Dijete, putem vlastitog jedinstvenog identifikatora, te u odnosima *One-to-Many* s entitetima Nalaz i Pregled.

MedicalRecord/Medicinski karton		
redordId	INT	Jedinstveni identifikator kartona
currentDiagnosis	VARCHAR	Aktivna ili zadnja dijagnoza djeteta
allergyList	VARCHAR	Popis alergija djeteta

Pregled - Entitet sadržava podatke pregleda kojeg održava pedijatar ili LOM. Oni su: šifra pregleda, šifra kartona kojem pripada, dijagnoza i datum pregleda. Entitet Pregled u odnosu je *Many-to-One* s entitetom Medicinski karton.

Examination/Pregled		
examinationId	INT	Jedinstveni identifikator pregleda
recordId	INT	Jedinstveni identifikator kartona
diagnosis	VARCHAR	Opis dijagnoze
dateOfExamination	DATE	Datum pregleda

Nalaz - Entitet sadržava podatke koji opisuju nalaz dobiven prilikom privatnog pregleda, kojeg *uploada* roditelj. Oni su: šifra nalaza, šifra kartona, datum nalaza, podaci nalaza. Entitet Nalaz u odnosu je *Many-to-One* s entitetom Medicinski karton.

MedicalReport/Nalaz		
reportId	INT	Jedinstveni identifikator nalaza
recordId	INT	Jedinstveni identifikator kartona
reportInformation	VARCHAR	Podaci iz nalaza
dateOfReport	DATE	Datum pregleda

Obavijest - Entitet sadržava podatke koji opisuju obavijest posлану родитељу, коју piše личник/педијатар. Oni су: шифра обавијести, OIB родитеља, OIB дјетета, адреса и текст обавијести. Entitet Obavijest у односу је *Many-to-One* са ентитетима Родитељ и Дјете.

Notification/Obavijest		
notificationId	INT	Jedinstveni identifikator обавијести
parentOib	CHAR(11)	Jedinstveni identifikator родитеља
childOib	CHAR(11)	Jedinstveni identifikator дјетета
notificationInformation	VARCHAR	Подаци обавијести
notificationTitle	VARCHAR	Наслов обавијести

Specijalistički pregled - Entitet sadržava podatke koji opisuju specijalistički pregled на којег личник/педијатар пријављује пацијента. Oni су: шифра pregleda, шифра pacijenta, назив pregleda, могуће lokacije pregleda. Entitet Spec. pregled у односу је *Many-to-One* са ентитетом Medicinski karton, преко чега му приступају ентитети Родитељ и Дјете.

SpecialistExamination/Specijalistički pregled		
examId	INT	Jedinstveni identifikator pregleda
recordId	INT	Jedinstveni identifikator kartona
examTitle	VARCHAR	Naziv pregleda
examLocations	VARCHAR	Moguće lokacije pregleda

Preporuka o bolovanju - Entitet sadržava податке о болovanју којег педијатар препоручује за родитеља дјетета. Oni су: шифра preporuke, шифра личника обiteljske

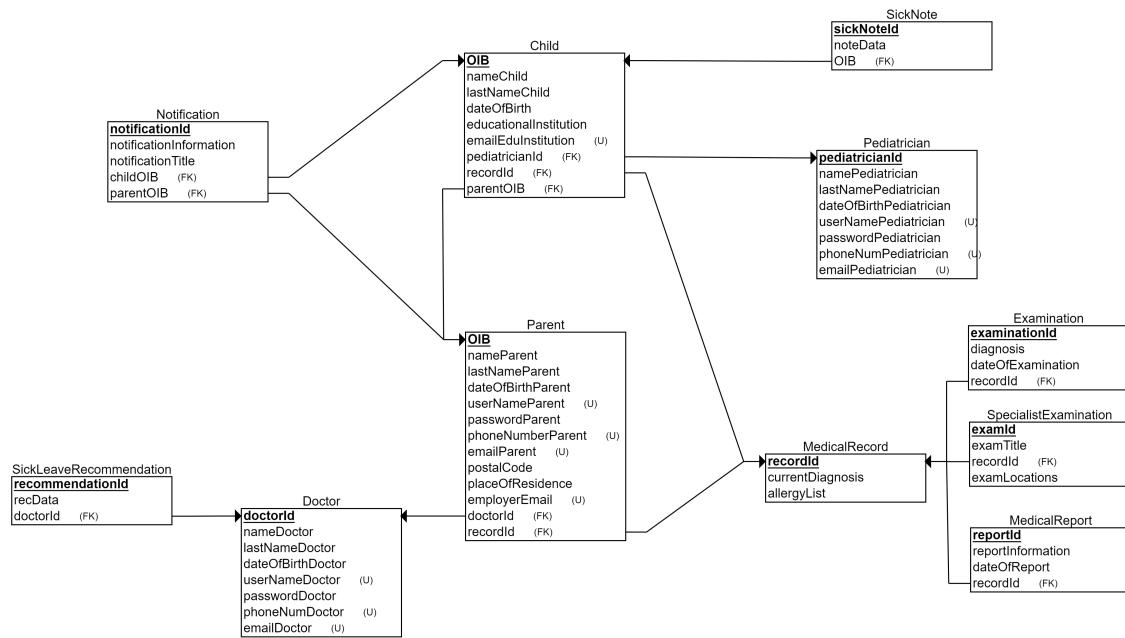
medicine, opis razloga za bolovanje i trajanja, te email poslodavca. Entitet Preporuka o bolovanju u odnosu je *Many-to-One* s entitetom Liječnik obiteljske medicine.

SickLeaveRecommendation/Preporuka o bolovanju		
recommendationId	INT	Jedinstveni identifikator preporuke
doctorId	INT	Jedinstveni identifikator liječnika
recData	VARCHAR	Sadržaj preporuke
employerEmail	VARCHAR	email poslodavca kojem se šalje doznaka

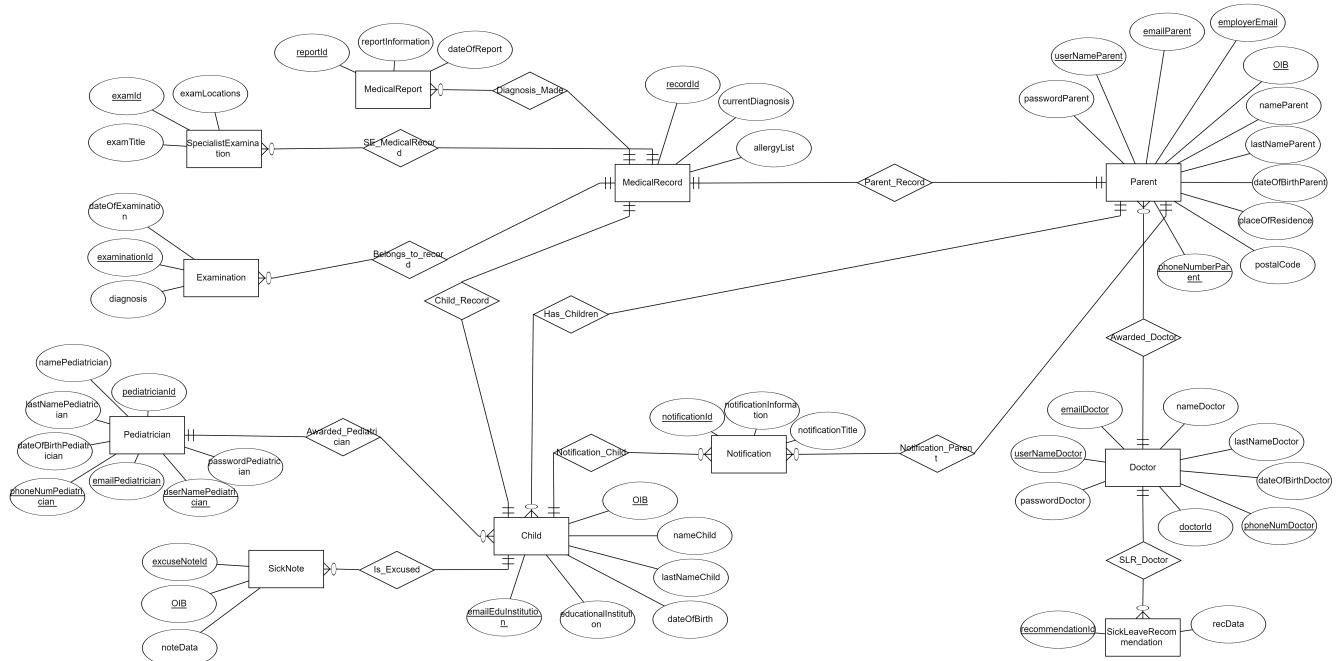
Ispričnica - Entitet sadržava podatke o ispričnici koju pedijatar piše za dijete. Oni su: šifra ispričnice, OIB djeteta, opis razloga za ispričnicom i trajanja. Entitet Ispričnica u odnosu je *Many-to-One* s entitetom Dijete.

SickNote/Ispričnica		
sicknoteId	INT	Jedinstveni identifikator ispričnice
childOib	CHAR(11)	Jedinstveni identifikator djeteta
noteData	VARCHAR	Sadržaj ispričnice

4.1.2 Dijagrami baze podataka



Slika 4.2: Dijagram arhitekture baze podataka

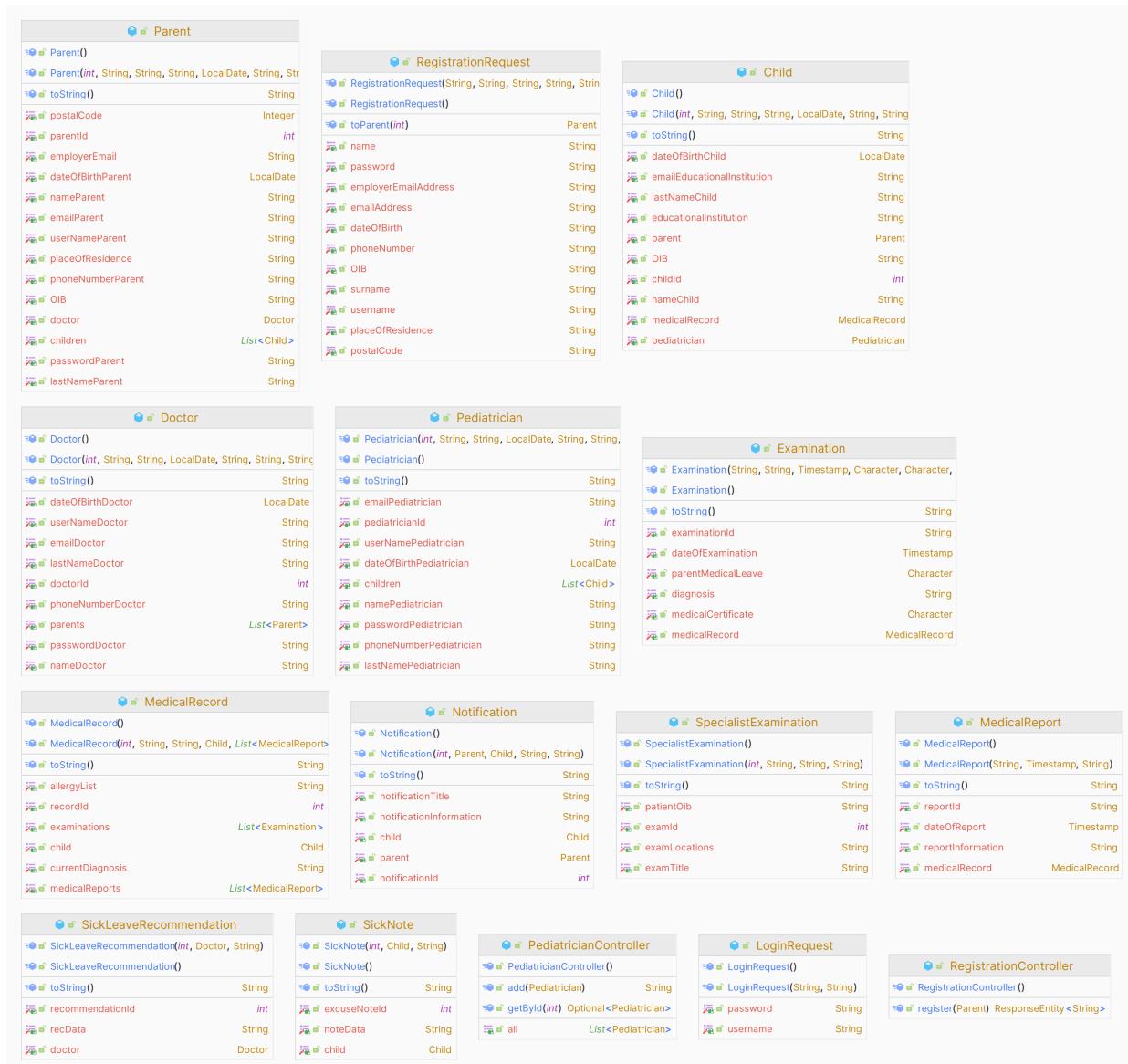


Slika 4.3: Dijagram arhitekture baze podataka

4.2 Dijagram razreda

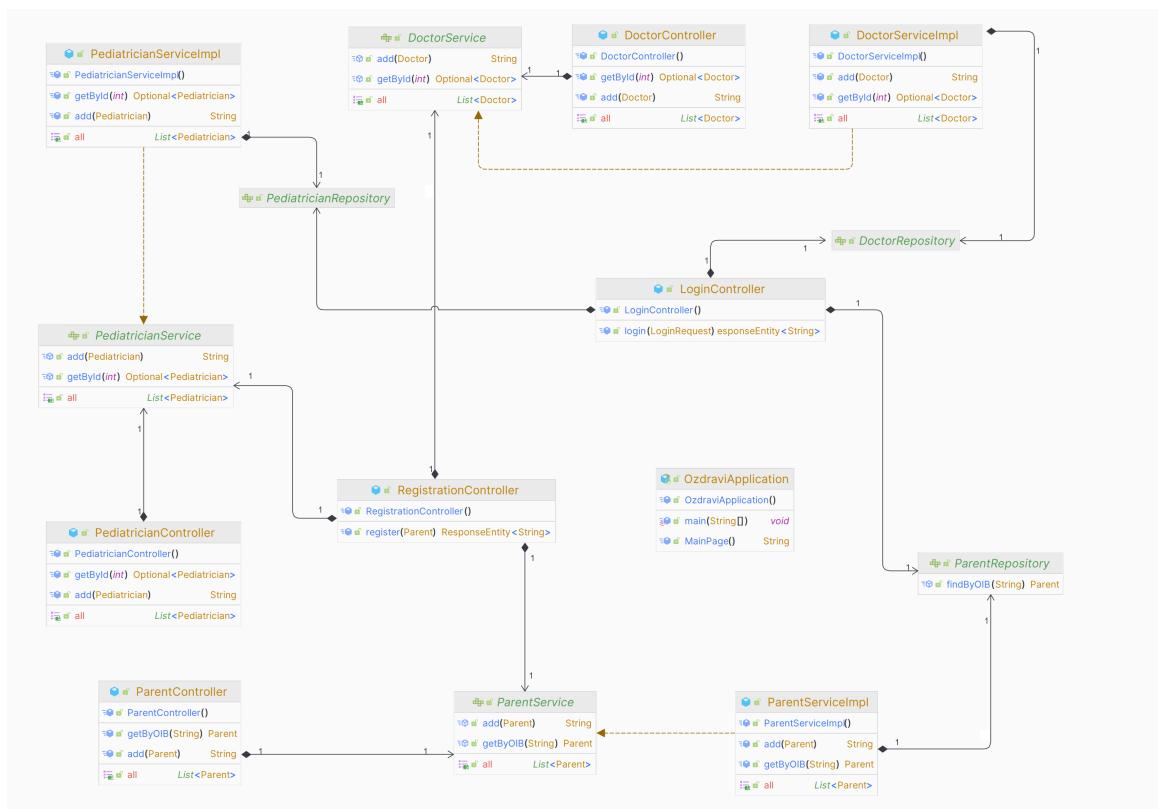
Na sljedećim slikama prikazani su dijagrami razreda ovog projekta. Zbog lakšeg snalaženja, razredi su raspodijeljeni na nekoliko dijagrama.

Postoje četiri razreda koji označavaju osobu: liječnik obiteljske medicine (Doctor), pedijatar (Pediatrician), roditelj (Parent) i njegovo dijete (Child). Razred Parent predstavlja roditelja koji se registrira u sustav te povezuje svoju djecu radi komunikacije s pedijatrom i liječnikom. Razred Doctor predstavlja liječnika obiteljske medicine kojeg registrira administrator i ima komunikaciju s roditeljem u vezi bolevanja, te ostalim zadaćama liječnika i pacijenta. Razred Pediatrician predstavlja pedijatra koji je zadužen za liječenje djeteta i izdavanje ispričnica i medicinskih dokumentacija. Razred Child predstavlja dijete registriranog roditelja koje se liječi kod pedijatra. Za dijete i roditelja kao pacijente vežu se tri razreda koji označavaju vrstu medicinske dokumentacije: medicinski kartoni (MedicalRecord), pregledi (Examination), specijalistički pregledi (SpecialistExamination) i nalazi (MedicalReport). Pregledi i nalazi su s djetetom povezani preko medicinskog kartona. Liječnik je povezan s roditeljem, a pedijatar je povezan s roditeljem preko djeteta. Razredi RegistrationRequest i LoginRequest služe za registraciju i prijavu korisnika u sustav.

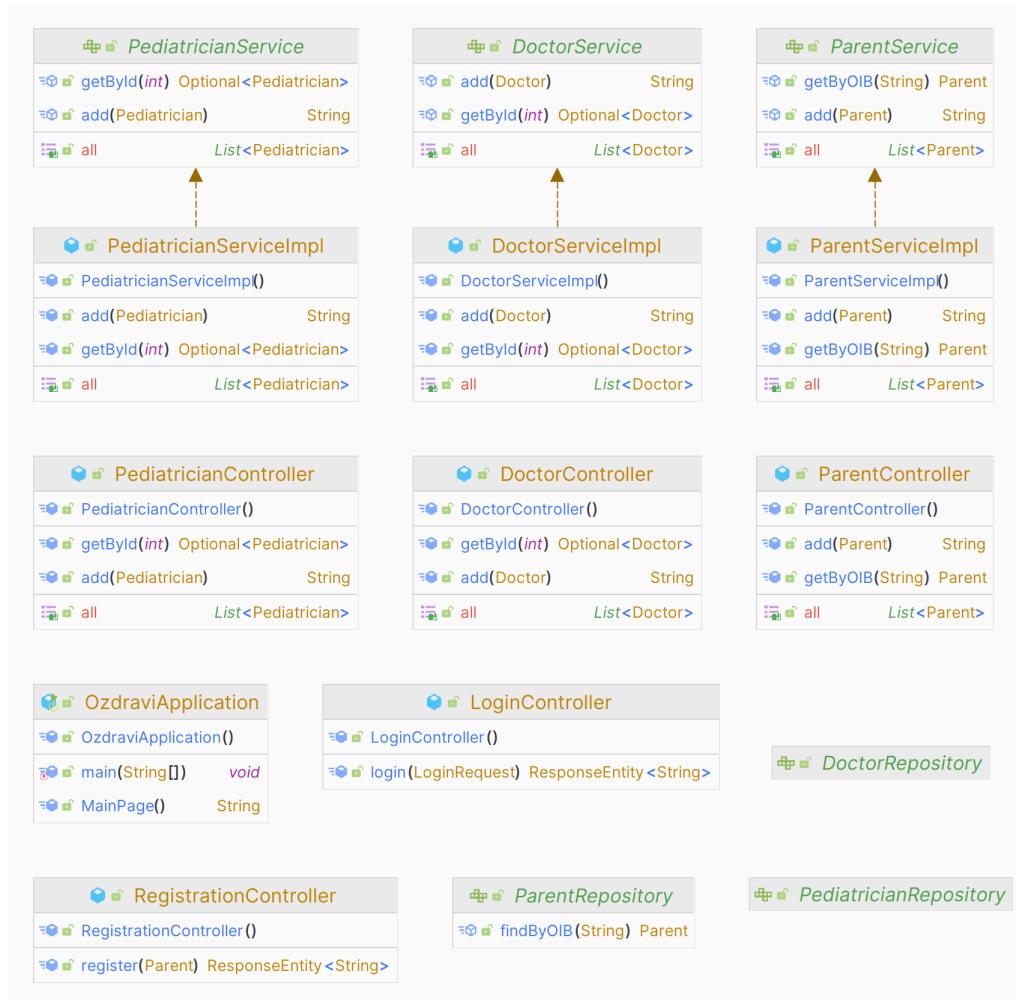


Slika 4.4: Dijagram razreda 1

Postoje tri razreda koji služe kao kontroleri za manipulaciju objekta i njihovo obrađivanje u bazi podataka za liječnika, pedijatra i roditelja. To su DoctorController, PediatricianController i ParentController. Postoje dva razreda RegistrationController i LoginController koji služe za rukovanje registracijama i prijavama korisnika. Za prijenos podataka između controllera i baze podataka postoje sučelja DoctorService, PediatricianService i ParentService te razredi DoctorServiceImpl, PediatricianServiceImpl i ParentServiceImpl koji implementiraju metode iz tih sučelja i osiguravaju prijenos podataka s bazom podataka preko tri sučelja koji predstavljaju rezervorije: DoctorRepository, PediatricianRepository i ParentRepository.



Slika 4.5: Dijagram razreda 2



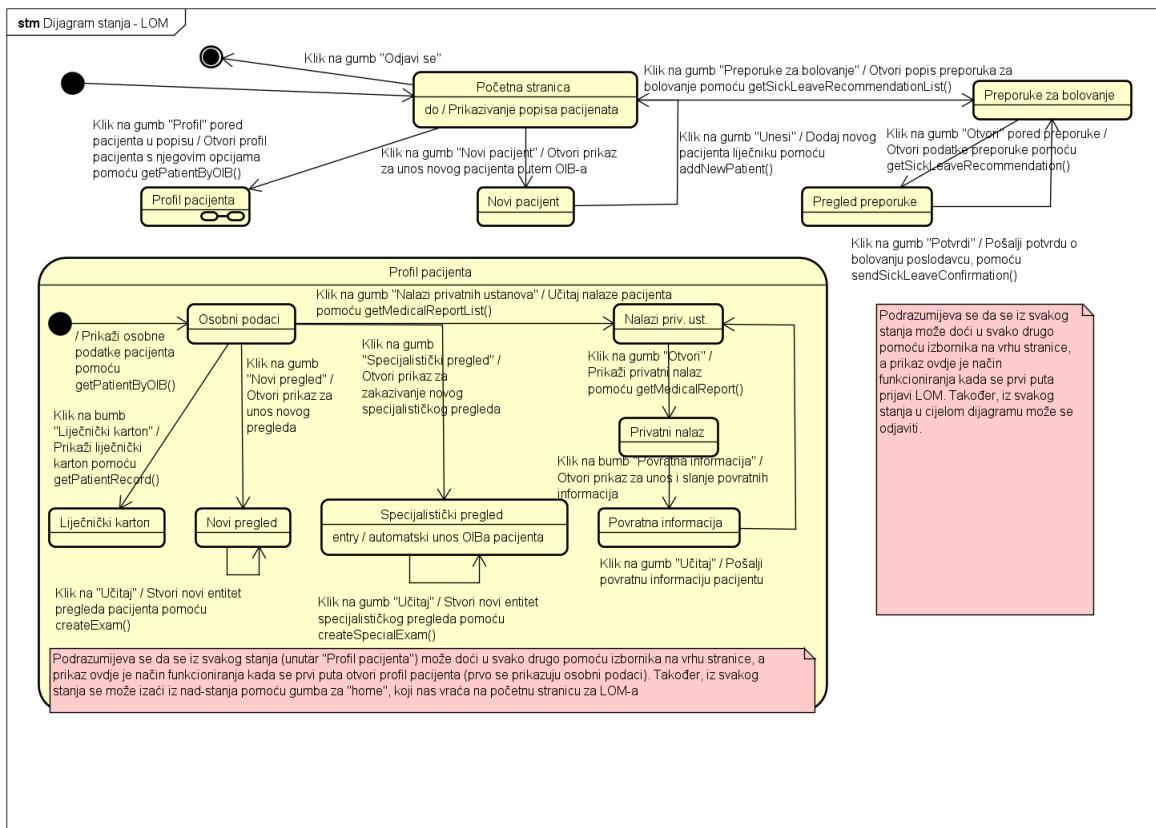
Slika 4.6: Dijagram razreda 3

dio 2. revizije

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

4.3 Dijagram stanja

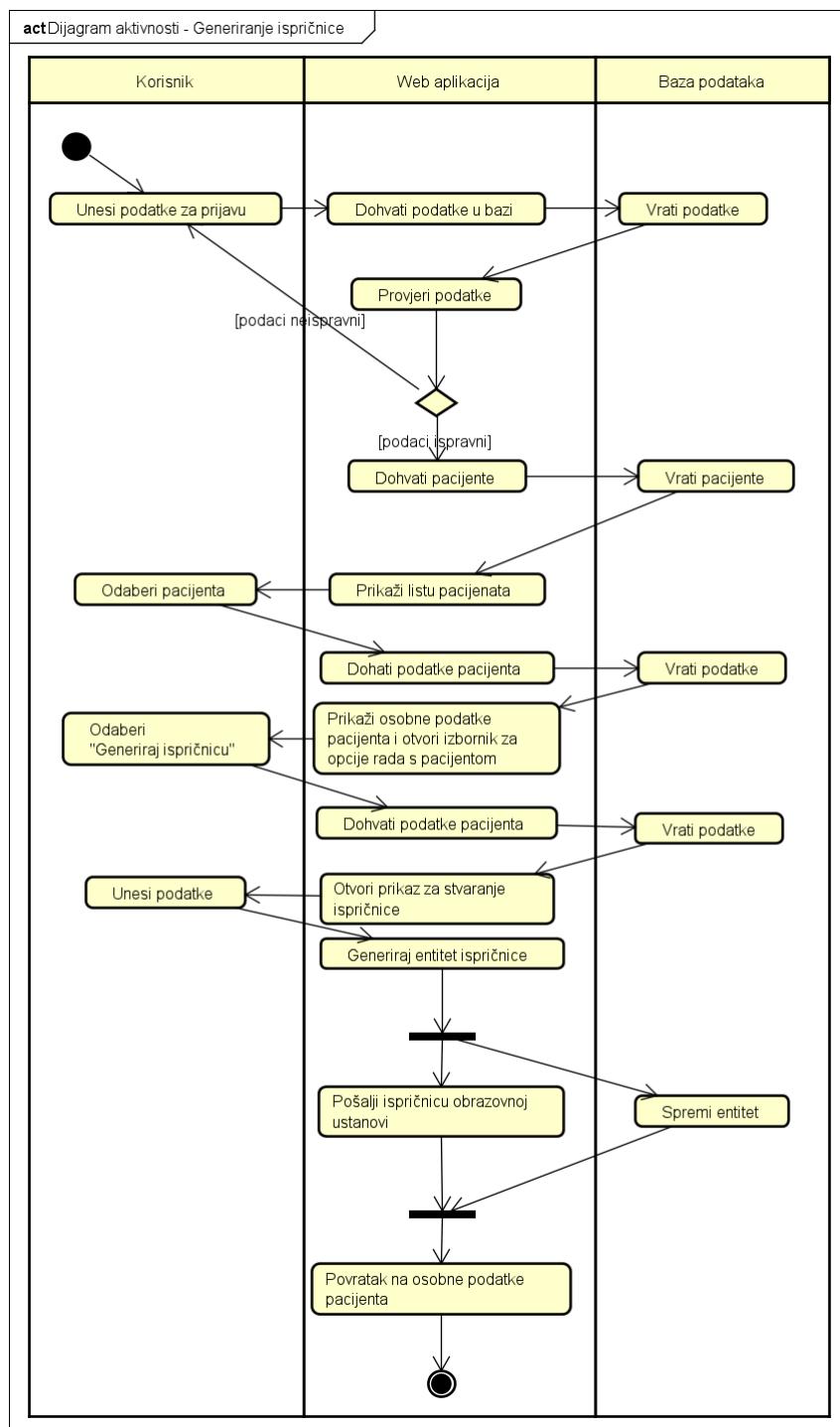
Dijagrami stanja pripadaju ponašajnim dijagramima u UML-u. Služe za opis diskretnih stanja sustava i prijelaza između istih. Na slici 4.7 prikazan je dijagram stanja za liječnika obiteljske medicine. Nakon prijave LOM-u se prikazuje popis njegovih pacijenata. Uz taj popis, ima opcije za prikaz preporuka za bolovanje koje je primio od raznih pedijatara, te opcija za unos novog pacijenta pomoću OIB-a istog. Uz svakog pacijenta u popisu ima opciju za otvaranje profila tog pacijenta. Nakon otvaranja profila, prvo se prikazuju osobni podaci pacijenta. Uz to, LOM ima opcije za prikaz liječničkog kartona, za unos novog pregleda odabranom pacijentu, za zakazivanje specijalističkog pregleda odabranom pacijentu, te za prikaz svih *uploadanih* privatnih nalaza tog pacijenta. Svaki od tih nalaza moguće je pojedinačno otvoriti, te na svaki poslati povratnu informaciju.



Slika 4.7: Dijagram stanja

4.4 Dijagram aktivnosti

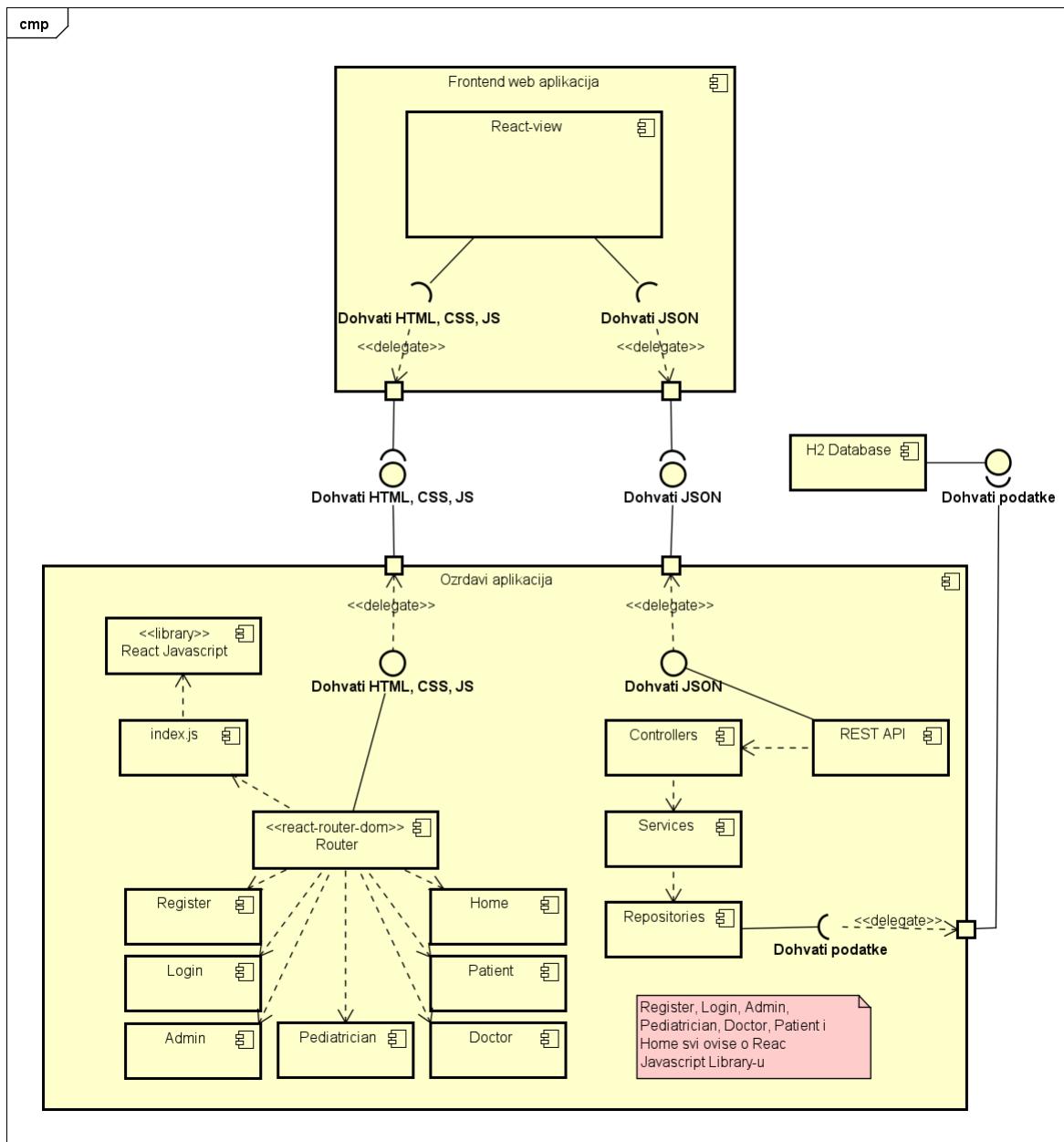
Dijagrami aktivnosti prikazuju radni tok aktivnosti koje se obavljaju, korak po korak, unutar nekog sustava. Dakle, modelira se ponašanje nizom akcija. Izričit je naglasak na jednostavnosti i čitljivosti. Na slici 4.8 prikazan je dijagram aktivnosti pedijatra, web aplikacije i baze podataka prilikom generiranja ispričnice za pacijenta. Pedijatar se prijavljuje u sustav. Potom odabire pacijenta iz liste, te pritišće opciju "Generiraj ispričnicu". Web aplikacija otvara prikaz s unesenim osobnim podacima pacijenta, te poljima za unos podataka o ispričnici/bolesti. Nakon što ih pedijatar unese, stvara se entitet ispričnice, koji se šalje obrazovnoj ustanovi i sprema u bazu podataka.



Slika 4.8: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti strukturni je i statički UML dijagram koji služi za specifikaciju arhitekture programske potpore. Vrlo je koristan za prikaz organizacije i međuvisnosti unutar građe sustava. Na slici 4.9 nalazi se dijagram komponenti koji opisuje strukturu naše aplikacije. Korisnik ima interakciju s *frontend* dijelom aplikacije, unutar kojeg se nalazi vizualno sučelje *React-view*. Ondje se nalaze dva sučelja za interakciju s aplikacijom. Sučelje za dohvrat HTML, CSS i JavaScript datoteka poslužuje *frontend* dio aplikacije i omogućuje njegov rad. Router komponenta odgovara na URL upite, te kao odgovor šalje različite React/JS datoteke. Te datoteke grupirane su radi jednostavnosti u logičke cjeline, koje su nazvane prema glavnim prikazima u aplikaciji, te po korisnicima iste. Sve JavaScript datoteke ovise o React biblioteci, jer iz nje dohvaćaju gotove komponente. Drugo sučelje služi za prijenos JSON podataka. REST API poslužuje te podatke. Svi podaci unutar aplikacije nalaze se u H2 bazi podataka. Ona je putem sučelja povezana s grupom komponenata Repositories, koje služe kao svojevrsni most između aplikacije i baze podataka. Services grupa komponenata vuče podatke iz repozitorija, te ih šalje upravljačima.



Slika 4.9: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu izvedena je putem dva servisa. Korištenjem aplikacije WhatsApp¹, te na serveru u aplikaciji Discord². Oboje imaju mogućnosti grupnih poziva, dijeljena ekrana te slanja poruka u grupni *chat*. Za izradu UML dijagrama u dokumentaciji korištena je aplikacija Astah³ UML kompanije ChangeVision⁴. Za kontrolu verzija izvornog koda korišten je Git⁵, a kao udaljeni repozitorij svog koda i dokumentacije korišten je direktorij na web platformi GitHub⁶. Kao razvojno okruženje koristili smo IntelliJ IDEA⁷, integrirano razvojno okruženje (IDE) proizvedeno od strane JetBrains⁸. IntelliJ IDEA koristi se za razvoj aplikacija pisanih u Javi⁹, Kotlin-u¹⁰, Groovy-u¹¹ i drugim JVM baziranim programskim jezicima. Aplikacija je napisana u Java programskom jeziku koristeći alat Spring Boot¹² za razvoj *backenda*, a za razvoj *frontenda* korišten je React¹³. React je biblioteka programskog jezika JavaScript¹⁴, koja se koristi za razvoj programske sučelja, a stvorio ju je i održava Meta¹⁵. Spring Boot alat je Spring¹⁶ Framework-a, koji je aplikacijski *framework* za razvoj Java aplikacija, najčešće onih baziranih na rad na web-u. Za bazu podataka koristili smo H2¹⁷, koji je prilagođen radu u Javi i baziran na poslužitelju, za kojeg smo odabrali Render¹⁸. Prilikom testiranja POST i GET me-

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://astah.net/>

⁴<https://www.change-vision.com/>

⁵<https://git-scm.com/>

⁶<https://github.com/>

⁷<https://www.jetbrains.com/idea/>

⁸<https://www.jetbrains.com/>

⁹<https://www.java.com/en/>

¹⁰<https://kotlinlang.org/>

¹¹<https://groovy-lang.org/>

¹²<https://spring.io/projects/spring-boot/>

¹³<https://react.dev/>

¹⁴<https://www.javascript.com/>

¹⁵<https://about.meta.com/>

¹⁶<https://spring.io/>

¹⁷<https://www.h2database.com/html/main.html>

¹⁸<https://render.com/>

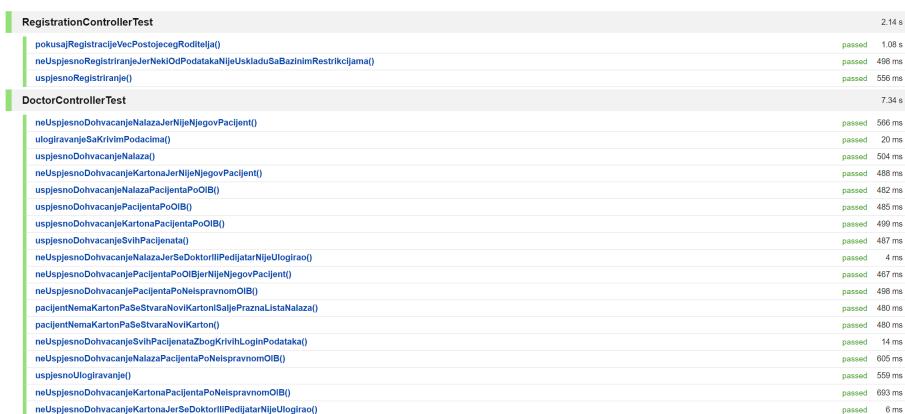
toda koristili smo se alatom Postman¹⁹.

¹⁹<https://www.postman.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Prilikom ispitivanja komponenti izведен je 21 test nad razredima u izvornom kodu. Ovo potpoglavlje biti će ograničeno na svega 6, no ostale je moguće vidjeti na GitHub repozitoriju. Poveznica na isti nalazi se u literaturi. Također, bitno je spomenuti da se sekcija testova naziva *DoctorControllerTest* može gledati kao skupina koja se odnosi i na korisnike tipa Pedijatar i korisnike tipa Liječnik obiteljske medicine, zbog sličnosti izvedbe razreda.



Slika 5.1: Prikaz izvršenih testova

- Test registracije postojećeg roditelja:** U navedenom testu šalje se zahtjev za stvaranjem novog entiteta roditelja koji ima već korišteni OIB unutar aplikacije. Očekuje se odgovor tipa "400 Bad Request", kakav odgovor je i dobiven.

```
@Test
void pokusajRegistracijeVecPostojecegRoditelja() throws Exception {
    String jsonPayload = "{\n" +
        "    \"name\": \"Petra\", \n" +
        "    \"surname\": \"Horvat\", \n" +
        "    \"oib\": \"12345678901\", \n" +
        "    \"dateOfBirth\": \"1980-12-04\", \n" +
        "    \"placeOfResidence\": \"Zagreb\", \n" +
        "    \"postalCode\": \"10000\", \n" +
        "    \"phoneNumber\": \"0971234567\", \n" +
        "    \"emailAddress\": \"petra.horvat@gmail.com\", \n" +
        "    \"username\": \"petraHorvat\", \n" +
        "    \"password\": \"petraHorvat\", \n" +
        "    \"employerEmailAddress\": \"telekom@t-com.hr\"\n" +
    "}";
    mockMvc.perform(MockMvcRequestBuilders.post("/register")
        .contentType(MediaType.APPLICATION_JSON)
        .content(jsonPayload))
        .andExpect(MockMvcResultMatchers.status().isBadRequest());
}
```

Slika 5.2: Test registracije postojećeg OIB-a

2. Test registracije roditelja: U navedenom testu šalje se zahtjev za stvaranjem novog entiteta roditelja, i kojemu su svi podaci ispravni. Očekuje se odgovor tipa "200 OK", kakav je i dobiven.

```
@Test
void uspjesnoRegistriranje() throws Exception {
    String jsonPayload = "{\n" +
        "    \"name\": \"Andrej\",\\n\" +
        "    \"surname\": \"Panković\",\\n\" +
        "    \"oib\": \"12345678910\",\\n\" +
        "    \"dateOfBirth\": \"1970-04-08\",\\n\" +
        "    \"placeOfResidence\": \"Zagreb\",\\n\" +
        "    \"postalCode\": \"10000\",\\n\" +
        "    \"phoneNumber\": \"0991234567\",\\n\" +
        "    \"emailAddress\": \"andrey.pankovic@yahoo.hr\",\\n\" +
        "    \"username\": \"najOsoba\",\\n\" +
        "    \"password\": \"najOsoba\",\\n\" +
        "    \"employerEmailAddress\": \"kebabShop@gmail.hr\"\n" +
    "}";
    mockMvc.perform(MockMvcRequestBuilders.post("/register")
        .contentType(MediaType.APPLICATION_JSON)
        .content(jsonPayload))
        .andExpect(MockMvcResultMatchers.status().isOk());
}
```

Slika 5.3: Test registracije roditelja

3. Test prijave i dobavljanja svih pacijenata: U navedenom testu šalju se ispravni podaci za prijavu, nakon čega se šalje zahtjev za popisom svih pacijenata prijavljenog doktora. Očekuje se odgovor tipa "200 OK", kakav je i dobiven.

```
@Test
void uspjesnoDohvacanjeSvihPacijenata() throws Exception {
    MockHttpSession session = new MockHttpSession();

    String jsonPayload = "{\n" +
        "    \"username\": \"anaIvanovic\",\\n\" +
        \"    \"password\": \"anaIvanovic\"\\n\" +
    \"}";

    mockMvc.perform(MockMvcRequestBuilders.post("/login")
        .session(session)
        .contentType(MediaType.APPLICATION_JSON)
        .content(jsonPayload));

    mockMvc.perform(MockMvcRequestBuilders.get("/doctor/getAllPatients")
        .session(session))
        .andExpect(MockMvcResultMatchers.status().isOk());
}
```

Slika 5.4: Test dobavljanja pacijenata

4. Test dobavljanja kartona: U navedenom testu šalju se ispravni podaci za prijavu, nakon čega se šalje zahtjev za kartonom određenog pacijenta. Taj pacijent nema postojeći karton, te je potrebno stvoriti novi. Očekuje se odgovor tipa "200 OK", kakav je i dobiven.

```

@Test
void pacijentNemaKartonPaSeStvaraNoviKarton() throws Exception{
    MockHttpSession session = new MockHttpSession();

    String jsonPayload = "{\n" +
        "    \"username\": \"jelenaKovac\",\\n\" +
        "    \"password\": \"jelenaKovac\"\\n\" +
        \"}\";

    mockMvc.perform(MockMvcRequestBuilders.post("/login")
        .session(session)
        .contentType(MediaType.APPLICATION_JSON)
        .content(jsonPayload));

    mockMvc.perform(MockMvcRequestBuilders.get("/doctor/getPatientRecord/48249825043")
        .session(session)
        .andExpect(MockMvcResultMatchers.status().isOk()));
}

```

Slika 5.5: Test dobavljanja kartona

5. **Test dobavljanja pacijenta po neispravnom OIB-u:** U navedenom testu šalju se ispravni podaci za prijavu, nakon čega se šalje zahtjev za profilom pacijenta čiji OIB nije unesen u sustav. Očekuje se odgovor tipa "400 Bad Request", kakav je i dobiven.

```

@Test
void neUspjesnoDohvacanjeKartonaPacijentaPoNeispravnomOIB() throws Exception{
    MockHttpSession session = new MockHttpSession();

    String jsonPayload = "{\n" +
        "    \"username\": \"anaIvanovic\",\\n\" +
        \"    \"password\": \"anaIvanovic\"\\n\" +
        \"}\";

    mockMvc.perform(MockMvcRequestBuilders.post("/login")
        .session(session)
        .contentType(MediaType.APPLICATION_JSON)
        .content(jsonPayload));

    mockMvc.perform(MockMvcRequestBuilders.get("/doctor/getPatientRecord/0000000000")
        .session(session)
        .andExpect(MockMvcResultMatchers.status().isBadRequest()));
}

```

Slika 5.6: Test dobavljanja pacijenata s neispravnim OIB-om

6. **Test dobavljanja nalaza pacijenta:** U navedenom testu šalju se ispravni podaci za prijavu, nakon čega se šalje zahtjev za nalazima pacijenta prijavljenog doktora. Očekuje se odgovor tipa "200 OK", kakav je i dobiven.

```
@Test
void uspjesnoDohvacanjeNalaza() throws Exception{
    MockHttpSession session = new MockHttpSession();

    String jsonPayload = "{\n        \"username\": \"anaIvanovic\",\\n        \"password\": \"anaIvanovic\"\\n    }";

    mockMvc.perform(MockMvcRequestBuilders.post("/login")
        .session(session)
        .contentType(MediaType.APPLICATION_JSON)
        .content(jsonPayload));

    mockMvc.perform(MockMvcRequestBuilders.get("/doctor/getPatientReports/12345678901")
        .session(session)
        .andExpect(MockMvcResultMatchers.status().isOk()));
}
```

Slika 5.7: Test dobavljanja nalaza

5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium²⁰. Razraditi minimalno 4 ispitna slučaja u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.

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

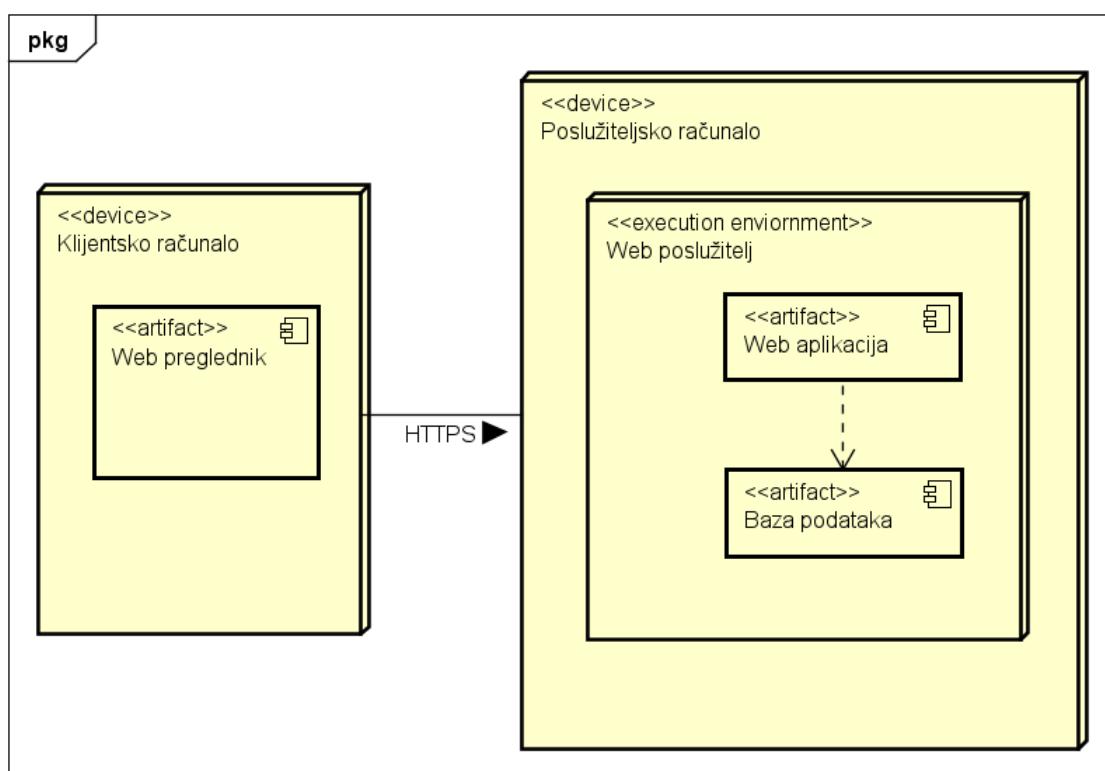
- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsко sučelje.

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

²⁰<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

Dijagrami razmještaja spadaju pod strukturne i statičke UML dijagrame koji opisuju topologiju sustava, te su usredotočeni na odnos sklopovskih i programskih dijelova. Na slici 5.X prikazan je dijagram razmještaja naše aplikacije. Na klijentskom se računalu nalazi web preglednik kojim se putem HTTPS veze pristupa aplikaciji. Web aplikacija pokrenuta je unutar web poslužitelja, koji se nalazi na poslužiteljskom računalu. Unutar okoline izvođenja web poslužitelja također se nalazi i baza podataka. Razlog manjku zasebnog poslužitelja za bazu podataka je korištenje baze H2, koja je integrirana u *backend*.

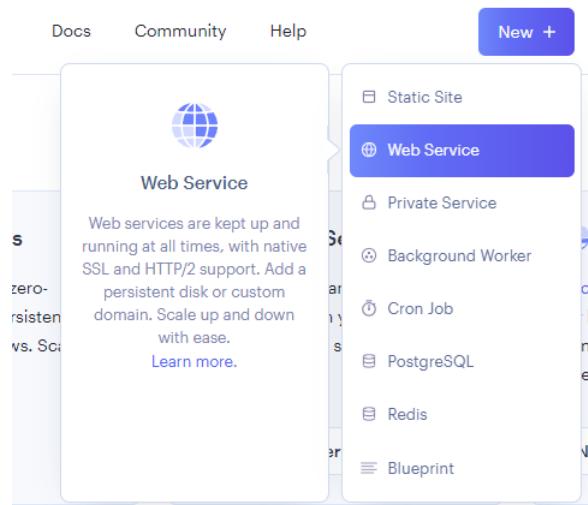


Slika 5.8: Dijagram razmještaja

5.4 Upute za puštanje u pogon

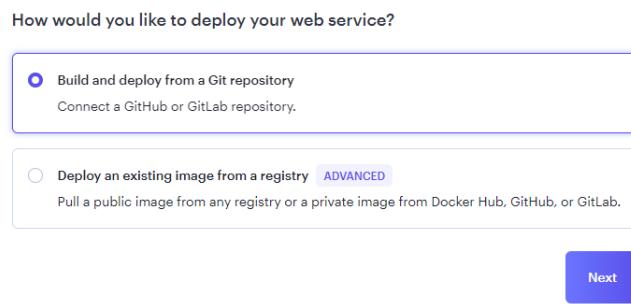
5.4.1 Deploy na Render - Backend

Potrebno je postaviti izvorni kod aplikacije na vlastiti repozitorij na servisu GitHub. Izvorni kod moguće je preuzeti s GitHub repozitorija projekta, čija se poveznica nalazi u literaturi. Jednom kada je cijeli izvorni kod na repozitoriju, potrebno je otvoriti stranicu servisa Render²¹, te se prijaviti ili kreirati račun. Kada je prijava provedena, potrebno je odabrati opciju "New", te odabrati opciju *Web service*.



Slika 5.9: Opcija *Web service*

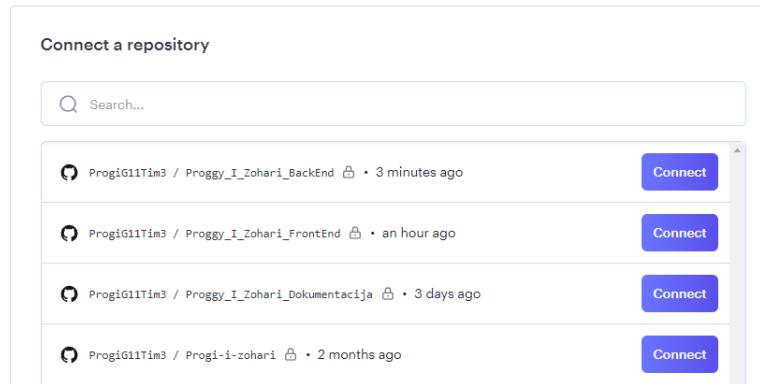
Nakon toga, potrebno je odabrati prvu opciju na sljedećem izborniku, *Build and deploy from a Git repository*.



Slika 5.10: Izbornik za *deploy*

²¹<https://dashboard.render.com/>

Na idućem ekranu potrebno je prijaviti se pomoću svog GitHub ili GitLab računa. Nakon što se korisnik uspješno prijavi, nudi mu se lista mogućih repozitorija za dodavanje na Render.



Slika 5.11: Izbornik repozitorija

Na sljedećem ekranu potrebno je unijeti ime servisa, regiju za korištenje, izvorni/korijenski direktorij te *Runtime* opciju. Pod prepostavkom da je odabran direktorij za *backend*, unesene opcije trebaju biti sljedeće:

Name	Ozdravi
Region	Frankfurt (EU Central)
Branch	main
Root Directory	e.g. src
Runtime	Docker

Slika 5.12: Izbornik opcija za stvaranje *backenda*

Dodatno, potrebno je otvoriti opciju *Advanced* te unijeti putanju do *Dockerfile*, što se u slučaju aplikacije ozdravi nalazi na lokaciji `./docker/Dockerfile`:

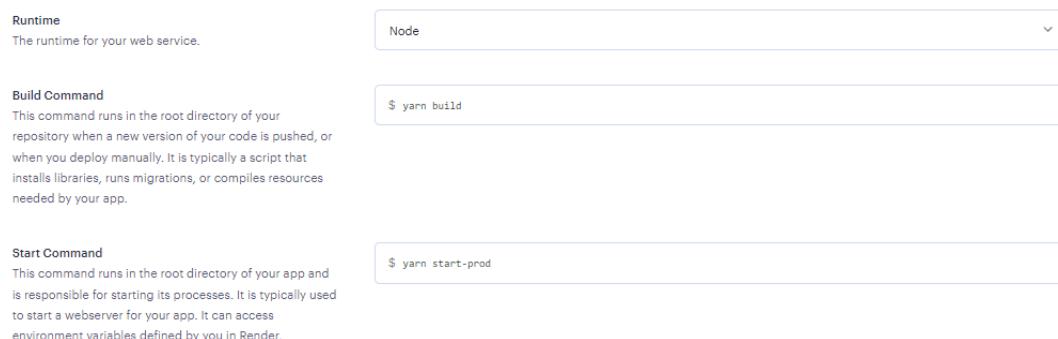


Slika 5.13: Unos lokacije Dockerfilea

Potrebno je još samo stisnuti gumb *Create Web Service*.

5.4.2 Deploy na Render - Frontend

Za *deploy frontend* dijela aplikacije, potrebno je ponovno odabratи opcije **New - Web service**. Prilikom unosa opcija, ovaj puta se postupa malо drugačije. Potrebno je unijeti naziv, te dodati **frontend** kao *root directory*, a zatim odabratи opciju *Node* unutar opcije *Runtime*. Uz to, potrebno je unijeti komandu **yarn build** unutar opcije *Build Command* i **yarn start-prod** unutar *Start Command*.



Slika 5.14: Unos opcija za *frontend*

Potrebno je još samo otvoriti opciju *Advanced*, te u dodatne *environment* varijable dodati varijablu imena **API_BASE_URL**, te postaviti *value* na adresu deploynaog backenda aplikacije dostupnu na Render dashboardu.



Slika 5.15: Unos opcije za povezivanje

Ponovno, potrebno je još samo pritisnuti *Create Web Service*.

6. Zaključak i budući rad

Zadatak našeg tima na ovome projektu bio je razvoj web aplikacije naziva "Ozdravi". Glavna ideja iza aplikacije jest olakšanje komunikacije između roditelja djece, pedijatara, liječnika obiteljske medicine, škola i poslodavaca. Razvoj aplikacije trajao je 13 tjedana, za vrijeme kojih se dinamika tima i njegova efikasnost drastično mijenjala. Ipak, na kraju je postignut cilj projekta, razvijena je aplikacija. Podijelio bih rad na projektu u tri faze.

Prva faza sastojala se od početne organizacije te nekoliko uzastopnih sastanaka na kojima je razvijana ideja i arhitektura aplikacije, te na kojima su se članovi tima međusobno upoznali. U ovoj fazi izglasan je i voditelj tima, čiju ulogu je preuzeo Jan Komerički. Također, u ovoj fazi su određena 4 pod-tima, svaki sa svojim zadaćima. Marko Žaja i Lovro Matić oformili su tim za rad s bazom podataka, koji se u kasnijoj fazi projekta stopio u tim za razvoj *backend* potpore. Taj tim oformili su Dino Dubinović i Luka Bračun. Treći tim sastojao se od Kristine Čavlović i Ante Prolića, koji su odlučili raditi na razvoju potpore za *frontend* dio aplikacije. Zadnji, jednočlani tim sastojao se od Jana Komeričkog, čiji je zadatak postao pisanje kompletne dokumentacije projekta i koordinacija rada ostala tri tima.

Druga faza rada tima bio je razvoj inicijalne verzije aplikacije, a krajnji rok ove faze bio je 17.11.2023. U ovoj fazi su se svi članovi tima upoznali sa tehnologijama koje su bile potrebne za razvoj aplikacije. Fazu je okarakterizirao spor i metodičan rad tima, dijelom zbog još nezrele suradnje među članovima, a dijelom zbog manjka znanja o tehnologijama. Velika većina dokumentacije je također završena u ovoj fazi, te većina UML dijagrama koji su potrebni za prikaz rada aplikacije. Ipak, faza je završila vrlo uspješno, s vrlo pozitivnim povratnim informacijama dionika.

Treća faza rada sastojala se od raspodijele preostalog posla i nastavak rada na razvoju aplikacije. Zadnji tjedan rada bio je pomalo kaotičan zbog ponekad nejasne komunikacije između pod-timova, no projekt i dokumentacija uspješno su završeni.

Ovaj projekt bio je prvi ozbiljni i opsežni projekt razvoja programske potpore za većinu nas. Upoznali smo nove tehnologije, pogotovo *Spring Boot* i *React*, te smo naučili o važnosti dokumentacije. Kvalitetna dokumentacija jedan je od najkoris-

nijih pomoćnika pri razvoju, jer se po njoj mogu orijentirati svi članovi tima. Sve u svemu, naučili smo kako raditi i komunicirati u timu, te kako se koristiti modernim tehnologijama za programske potpore. Mislim da svi odlazimo od ovog projekta iznimno zadovoljni postignutim, usprkos velikoj količini mogućih unaprijeđenja koja se mogu dodati aplikaciji u budućnosti. ODređeni članovi tima čak su izrazili želju za dalnjim radom unutar ovog tima, ali na nekom vlastitom projektu, tako da je moguće da će *Proggy i Žohari* nastaviti razvijati *software*, bar još neko vrijeme. :)

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <https://www.fer.unizg.hr/predmet/proinzh>
2. I. Sommerville, "Engineering software products : an introduction to modern software engineering", Global ed, Pearson 2021.
3. A. Jović, M. Horvat, I. Grudenić, "UML-dijagrami: Zbirka primjera i riješenih zadataka", Graphis d.o.o. 2014.
4. The Unified Modeling Language, <https://www.uml-diagrams.org/>
5. Repozitorij izvornog koda, <https://github.com/ProgiG11Tim3/Progi-i-zohari>

Indeks slika i dijagrama

2.1 Izgled početnog izbornika platforme <i>Portal zdravlja</i>	9
3.1 Dijagram obrazaca uporabe, funkcionalnost nereg. korisnika i Roditelja	29
3.2 Dijagram obrazaca uporabe, funkcionalnost Pedijatra i L.O.M.	30
3.3 Dijagram obrazaca uporabe, funkcionalnost Administratora	30
3.4 Sekvencijski dijagram prikaza pregleda	31
3.5 Sekvencijski dijagram osnovnih funkcionalnosti pedijatra	32
3.6 Sekvencijski dijagram stvaranja i potvrde preporuke o bolovanju	33
3.7 Sekvencijski dijagram rada s privatnim nalazima	34
4.1 Dijagram arhitekture projekta	36
4.2 Dijagram arhitekture baze podataka	44
4.3 Dijagram arhitekture baze podataka	44
4.4 Dijagram razreda 1	46
4.5 Dijagram razreda 2	47
4.6 Dijagram razreda 3	48
4.7 Dijagram stanja	49
4.8 Dijagram aktivnosti	51
4.9 Dijagram komponenti	53
5.1 Prikaz izvršenih testova	56
5.2 Test registracije postojećeg OIB-a	56
5.3 Test registracije roditelja	57
5.4 Test dobavljanja pacijenata	57
5.5 Test dobavljanja kartona	58
5.6 Test dobavljanja pacijenata s neispravnim OIB-om	58
5.7 Test dobavljanja nalaza	59
5.8 Dijagram razmještaja	61
5.9 Opcija <i>Web service</i>	62
5.10 Izbornik za <i>deploy</i>	62

5.11 Izbornik repozitorija	63
5.12 Izbornik opcija za stvaranje <i>backenda</i>	63
5.13 Unos lokacije Dockerfilea	64
5.14 Unos opcija za <i>frontend</i>	65
5.15 Unos opcije za povezivanje	65

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 17. listopada 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - prvi sastanak s asistenticom i demonstratorom, te predstavnikom CROZ-a
 - kratko predavanje o projektu, generalan plan izvedbe

2. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - prvi samostalan sastanak
 - grupna analiza zadatka, raščišćavanje osnovnih dilema funkcionalnosti
 - raspodijela odgovornosti
 - odabir alata i tehnologija

3. sastanak

- Datum: 24. listopada 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - drugi sastanak s asistenticom i demonstratorom
 - prolazak kroz pitanja vezana uz funkcionalnost i izgled aplikacije
 - razriješene dileme vezane uz uporabu GitHub-a

4. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: svi članovi tima

- Teme sastanka:
 - drugi samostalni sastanak
 - donesene konačne odluke o arhitekturi i izgledu aplikacije

5. sastanak

- Datum: 6. studenog 2023.
- Prisustvovali: Luka Bračun, Dino Dubinović
- Teme sastanka:
 - sastanak pod-tima za razvoj *backend-a*
 - grupno programiranje

6. sastanak

- Datum: 7. studenog 2023.
- Prisustvovali: Jan Komerički
- Teme sastanka:
 - treći sastanak s asistenticom
 - pojašnjen ispravan način pisanja dokumentacije, specifično *use cases*

7. sastanak

- Datum: 9. studenog 2023.
- Prisustvovali: Luka Bračun, Dino Dubinović, Luka Žaja, Lovro Matić
- Teme sastanka:
 - virtualan sastanak pod-timova za razvoj baze podataka i *backend-a*
 - objašnjenje arhitekture baze podataka, te napravljen plan integracije iste s *backend-om*
 - vježba korištenja sustava *Postman*

8. sastanak

- Datum: 13. studenog 2023.
- Prisustvovali: Luka Bračun, Dino Dubinović, Kristina Čavlović, Ante Prolić
- Teme sastanka:
 - virtualan sastanak pod-timova za razvoj *frontend-a* i *backend-a*
 - rad s login i registracija funkcionalnostima

9. sastanak

- Datum: 15. studenog 2023.
- Prisustvovali: Dino Dubinović, Kristina Čavlović, Ante Prolić, Jan Komerički
- Teme sastanka:

- virtualan sastanak pod-timova za razvoj *frontend-a* i *backend-a*, te voditelja tima
- rad s login i registracija funkcionalnostima

10. sastanak

- Datum: 16. studenog 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - virtualan sastanak cijelog tima
 - prikaz rada aplikacije, *deployment*

11. sastanak

- Datum: 6. siječnja 2023.
- Prisustvovali: Jan Komerički, Dino Dubinović, Luka Žaja, Lovro Matić, Luka Bračun
- Teme sastanka:
 - virtualan sastanak voditelja i *backend* tima
 - planiranje i raspodjela posla

12. sastanak

- Datum: 11. siječnja 2023.
- Prisustvovali: Dino Dubinović, Luka Žaja, Lovro Matić, Luka Bračun
- Teme sastanka:
 - virtualan sastanak *backend* tima
 - planiranje i raspodjela posla

13. sastanak

- Datum: 14. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - virtualan sastanak cijelog tima
 - testiranje i spajanje određenih komponenti

14. sastanak

- Datum: 15. siječnja 2023.
- Prisustvovali: Dino Dubinović, Kristina Čavlović, Lovro Matić, Luka Žaja, Jan Komerički
- Teme sastanka:
 - virtualan sastanak tima
 - razvoj aplikacije

15. sastanak

- Datum: 16. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - virtualan sastanak tima
 - razvoj aplikacije

16. sastanak

- Datum: 17. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - virtualan sastanak tima
 - razvoj aplikacije i spajanje komponenti

17. sastanak

- Datum: 18. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - virtualan sastanak tima
 - razvoj aplikacije i spajanje komponenti

18. sastanak

- Datum: 19. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - virtualan sastanak tima
 - razvoj aplikacije, spajanje komponenti i testiranje aplikacije

Tablica aktivnosti

	Jan Komerički	Luka Bračun	Kristina Čavlović	Dino Dublinović	Lovro Matić	Ante Prolić	Luka Žaja
Upravljanje projektom	15						
Opis projektnog zadatka	4						
Funkcionalni zahtjevi	3						
Opis pojedinih obrazaca	15						
Dijagram obrazaca	3						
Sekvencijski dijagrami	8						
Opis ostalih zahtjeva	0.5						
Arhitektura i dizajn sustava	5	5	5	5	5	5	5
Opis baza podataka	8.5						
Dijagrami baza podataka					3		3
Dijagram razreda		4					
Dijagram stanja	3						
Dijagram aktivnosti	3						
Dijagram komponenti	4						
Korištene tehnologije i alati	.5						
Ispitivanje programskog rješenja	3						4
Dijagram razmještaja	1.5						
Upute za puštanje u pogon	1						
Dnevnik sastajanja	1						

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Jan Komerički	Luka Bračun	Kristina Čavlović	Dino Dublinović	Lovro Matić	Ante Prolić	Luka Žaja
Zaključak i budući rad	1						
Popis literature	.25						
<i>vizualni dizajn aplikacije</i>			15				
<i>projektiranje baze podataka</i>							4
<i>izrada baze podataka</i>					15		
<i>spajanje s bazom podataka</i>							
<i>back end</i>		26		24			15
<i>front end i spajanje</i>			42			36	
<i>puštanje u pogon</i>							

Dijagrami pregleda promjena

dio 2. revizije

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