

Oblikovanje programske potpore
Ak. god. 2019./2020.

Poliklinika za rehabilitaciju
Dokumentacija

Grupa: *Flow*
Voditelj: *Marko Malkoč*

Datum predaje: *16.1.2020.*

Nastavnik: *Hrvoje Nuić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	11
3.1 Funkcionalni zahtjevi	11
3.1.1 Obrasci uporabe	13
3.1.2 Sekvencijski dijagrami	30
3.2 Ostali zahtjevi	37
4 Arhitektura i dizajn sustava	38
4.1 Baza podataka	39
4.1.1 Opis tablica	40
4.1.2 Dijagram baze podataka	50
4.2 Dijagram razreda	51
4.3 Dijagram stanja	55
4.4 Dijagram aktivnosti	57
4.5 Dijagram komponenti	59
5 Implementacija i korisničko sučelje	61
5.1 Korištene tehnologije i alati	61
5.2 Ispitivanje programskog rješenja	61
5.2.1 Ispitivanje komponenti	61
5.2.2 Ispitivanje sustava	61
5.3 Dijagram razmještaja	63
5.4 Upute za puštanje u pogon	64
6 Zaključak i budući rad	70
Popis literature	71
Indeks slika i dijagonama	73

Dodatak: Prikaz aktivnosti grupe

74

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Malkoč	14.10.2019.
0.2	Započeo nefunkcionalne zahtjeve, nefunkcionalne dijelove, opis arhitekture <u>sustava</u>	Juričić	28.10.2019.
0.3	Napisani dionici, aktori i njihovi opisi	Čižmešija	28.10.2019.
0.4	Dorađeni funkcionalni zahtjevi	Malkoč	28.10.2019.
0.5	Dodan prvi obrazac uporabe	Malkoč	28.10.2019.
0.6	Reorganizacija datoteka i uređeni funkcionalni zahtjevi	Malkoč	02.11.2019.
0.7	Dodan opis baze	Bakula	03.11.2019.
0.8	Dodani obrasci uporabe i reformatirani nefunkcionalni zahtjevi	Malkoč	05.11.2019.
0.9	Uređene tablice baze podataka	Bakula	10.11.2019.
0.10	Napisani dijagrami razreda	Ereš, Zanetti	14.11.2019.
0.11	Ubačene slike obrazaca uporabe	Čižmešija	15.11.2019.
0.12	Dodan opis projektnog zadatka	Omrčen	15.11.2019.
0.13	Uređeno poglavlje Specifikacija programske potpore	Malkoč	15.11.2019.
0.14	Dodan dijagram baza i ključevi	Bakula	15.11.2019.
0.15	Dodani sekvencijski dijagrami i opis dijagrama razreda	Zanetti, Ereš	15.11.2019.
0.16	Izmijenjen opis arhitekture sustava	Juričić	15.11.2019.
0.17	Dnevnik sastanaka	Ereš	15.11.2019.
0.18	Uređena tablica aktivnosti	Malkoč	15.11.2019.
0.19	Dnevnik promjena dokumentacije	Zanetti	15.11.2019.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Malkoč	15.11.2019.
1.1	Dijagram razmještaja	Ereš	9.1.2020.
1.2	Dijagram komponenti	Čižmešija	11.1.2020.

Rev.	Opis promjene/dodataka	Autori	Datum
1.3	Dijagram stanja	Zanetti i Omrčen	12.1.2020.
1.4	Upute za puštanje u pogon	Malkoč	12.1.2020.
1.5	Zaključak i budući rad	Čižmešija	13.1.2020.
1.6	Ispitivanje programskog rješenja	Juričić	14.1.2020.
1.7	Dijagram aktivnosti	Bakula	14.1.2020.
2.0	Završne promjene	Malkoč	16.1.2020.

2. Opis projektnog zadatka

Cilj ovog projektnog zadatka je razviti programsku podršku za stvaranje web aplikacije „Poliklinika za rehabilitaciju“. Rehabilitacija od bolesti, ozljeda i operacija je složen i osjetljiv proces koji zahtjeva disciplinu i detaljan plan kojeg se pacijent mora držati. Svaki pacijent je drugačiji i zahtjeva specifičnu njegu koju mu bolnica mora osmisliti i omogućiti. Danas se nalazimo u modernom dobu gdje tehnologija uvelike olakšava mnoge aspekte ljudskog života pa tako i oporavak pacijenata te motrenje njihovog napretka i prevencija potencijalnih novih ozljeda.

U procesu oporavka određenim pacijentima je potreban doktor koji će propisati prikladnu dijetu, a nekim pacijentima je potreban samo trener koji će propisivati vježbe za svaki dan i motriti napredak te ponovno prilagođavati intenzitet vježbe s obzirom na trenutno stanje pacijenta. Na kraju postoje pacijenti kojima treba i doktor i trener koji će mu pomoći.

Svakodnevni odlazak u bolnicu mnogim pacijentima nije najbolji put do oporavka. Većina pacijenata nije u dovoljno dobrom fizičkom stanju za putovanje u bolnicu i čekanje u redu. Također za bolnicu nije efikasno da primaju toliko pacijenata ako postoji lakši način kojim bi mogli komunicirati sa svojim pacijentima i dati im najbolje savjete bez stresa i požurivanja.

Zbog ovih razloga stvorili smo aplikaciju koja bi omogućila stalnu komunikaciju između pacijenata, doktora i trenera. Doktor i trener ne mogu stalno biti s pacijentom i motriti ga, ali pomoći ove aplikacije pacijenti bi uvijek znali što moraju raditi, koliko i što smiju jest i imali bi osjećaj da je netko uvijek uz njih, a psihološka podrška je također vrlo bitna za pacijente.

U aplikaciji, doktor bi imao mogućnost propisivanja dijete pacijentu, a trener bi mu zadavao točno definirane treninge za svaki dan.

REGISTRACIJA

Kako bi se mogla koristiti ova aplikacija budući korisnik se mora prvo registrirati tako da pošalje zahtjev za registraciju prilikom koje bira ulogu koju želi imati, a one su slijedeće:

- **Klijent**
- **Doktor**
- **Trener**

Prilikom svake registracije potrebno je unijeti:

- **KorisničkoIme**
- **Ime**
- **Prezime**
- **Loziku**

Ukoliko se korisnik želi registrirati kao doktor ili trener potrebno je također dodatno priložiti:

- **Sliku**
- **Mail**
- **Maskimalan broj korisnika(koje želi nadgledati)**

Da bi neregistrirani korisnik dobio prava doktora i trenera, administrator ga mora potvrditi. Svi registrirani korisnici imaju mogućnost mijenjanja podataka na svom profilu.

PROIZVODI

Svaka dijeta sastoji se od određenih proizvoda koje klijenti mogu konzumirati. Bitno je da i doktor i pacijent znaju što se točno nalazi u kojem proizvodu kako bi doktor što bolje osmislio dijetu i kako bi ju klijent što bolje mogao pratiti. Popis proizvoda sa svim njihovim karakteristikama nalazit će se u bazi podataka. Svaki proizvod sadrži informacije o nutritivnim vrijednostima koje su definirane na 100 g, a to su:

- **Energija**
- **Masnoće**
- **Zasićene masne kiseline**
- **Ugljikohidrati**

- Šećeri
- Bjelančevine
- Sol

Dodatno, svaki proizvod sadrži sliku, masu i prisutne alergene. Također, oni će biti raspoređeni u kategorije koje sadrže više različitih proizvoda sličnih karakteristika, kao što su na primjer tjestenina, proizvodi od mlijeka, meso, voće, povrće...

TRENINZI

Trening se sastoji od niza vježbi koje imaju određeno trajanje. Vježba je definirana sa:

- **Slikom**
- **Opisom**
- **Informacijama o broju potrošenih kalorija u sat vremena u ovisnosti o 3 razine intenziteta vježbanja (lagano, normalno, teško)**

Klijent ima mogućnost traženja i pronađazaka dostupnih(onih koji još nemaju popunjeno maksimalan broj klijenata) doktora i trenera. Klijent može ostaviti recenziju s ocjenom i komentarom svom doktoru i treneru na koju oni mogu odgovoriti. Klijent, doktor i trener imaju mogućnost prekida suradnje. Sve kombinacije su moguće; da pacijent prekine suradnju s doktorom i trenerom ili samo jednim od njih. Kad pacijent prekine suradnju ima mogućnost biranja novog doktora i trenera, ali dok ih ne izabere on je u mogućnosti i dalje pratiti dijetu i trening koji su mu prije bili dodjeljeni.

Klijent ima mogućnost unošenja slike bar koda proizvoda koje planira konzumirati, a aplikacija mu javlja(uzimajući u obzir i ostale konzumirane namirnice u tom danu) uklapa li se taj proizvod u dijetu za taj dan. Uz unošenje bar koda, klijent može i ručno odabrati proizvod i postaviti mu masu.

Doktor svojim klijentima definira dijetu unoseći sve potrebne parametre. Dijeta se može definirati s ograničenjima na:

- **određene proizvode**
- **kategorije proizvoda**
- **proizvode s nedopuštenim količinama sastojaka**
- **dnevnim limitom za određene nutritivne vrijednosti proizvoda**

Doktor ima mogućnost dodavanja novih proizvoda i kategorija proizvoda u bazu podataka.

Trener svojim klijentima definira vježbe koje trebaju odraditi. Definira točno koliko puta se koja vježba treba napraviti i kojim redoslijedom ih mora raditi. Klijent će moći vidjeti sliku te vježbe i detaljan opis načina izvođenja vježbe. Trener određuje intenzitet vježbanja sukladno mogućnostima klijenta. S vremenom trener će mijenjati intenzitet kako bi oporavak prošao sa što manje ozljeda i najboljim rezultatima. Trener također ima pristup bazi podataka u koju može dodati nove vježbe.

Administrator ima najveće ovlasti. On ima pristup bazi s popisom kategorija proizvoda i vježbi i ima mogućnost uređivanja i brisanja. Također jedino on može potvrditi račun za doktora i trenera.

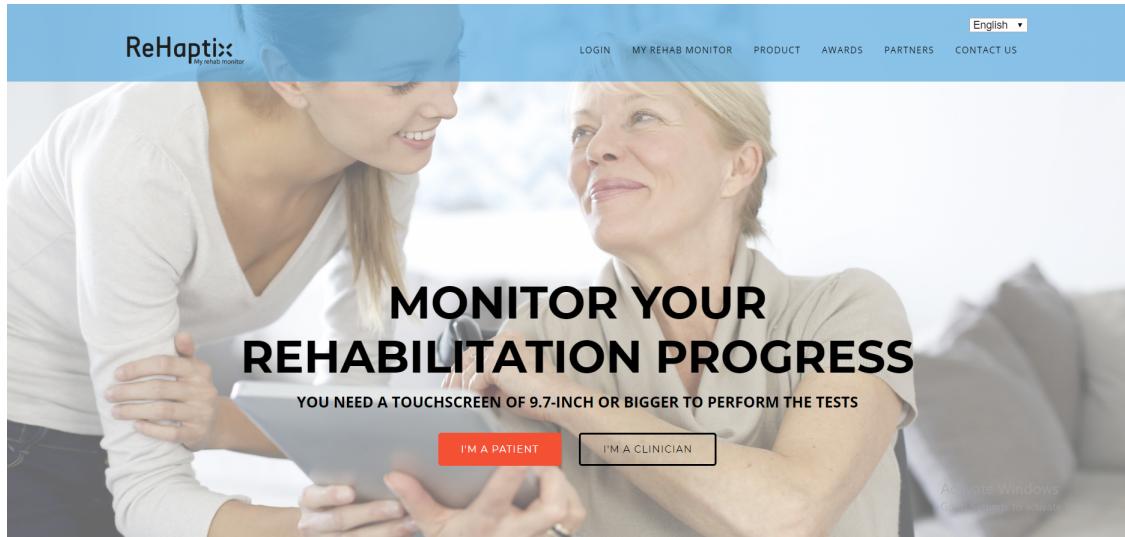
Nadogradnja

Aplikacija „Poliklinika za rehabilitaciju“ ostavlja mnogo prostora i mogućnosti za nadogradnju i proširenje. Korisno proširenje bilo bi omogućavanje slanja poruka između klijenata i doktora/trenera. Tako bi klijent mogao razriješiti postojeće nedoumice što prije, a ne bi ometao doktora/trenera slanjem poruka na privatni mobitel. Još jedno dobro proširenje bilo bi omogućavanje dogovora za pregled kod doktora, odnosno doktor bi mogao napomenuti klijentu da je vrijeme za pregled u bolnici i onda bi klijent u rasporedu izabrao slobodan termin koji mu najviše paše. Na taj način izbjegla bi se gužva i olakšao cijeli proces naručivanja kod doktora.

Ova aplikacija je namijenjena širokom spektru ljudi. Napravljena je tako da bude laka za korištenje, intuitivna i jednostavna. Nema nepotrebnih funkcionalnosti koje bi dodatno zbumile korisnike, tako da ju svi mogu koristiti, od mlađih do starih.

Slične aplikacije koje već postoje:

REHAPTIX



Slika 2.1: Početna stranica

Ova aplikacija funkcioniрајући слично као и „Poliklinika за rehabilitaciju“. Moguће је изабрати између две опције приликом пријаве:

- Ja sam pacijent(„I'm a patient“)
- Ja sam radnik u bolnici(„I'm a clinician“)

Login:

This site uses cookies. By continuing to browse the site, you are agreeing to our use of cookies.
 x

Already a member?

Email

Password

Forgot password?

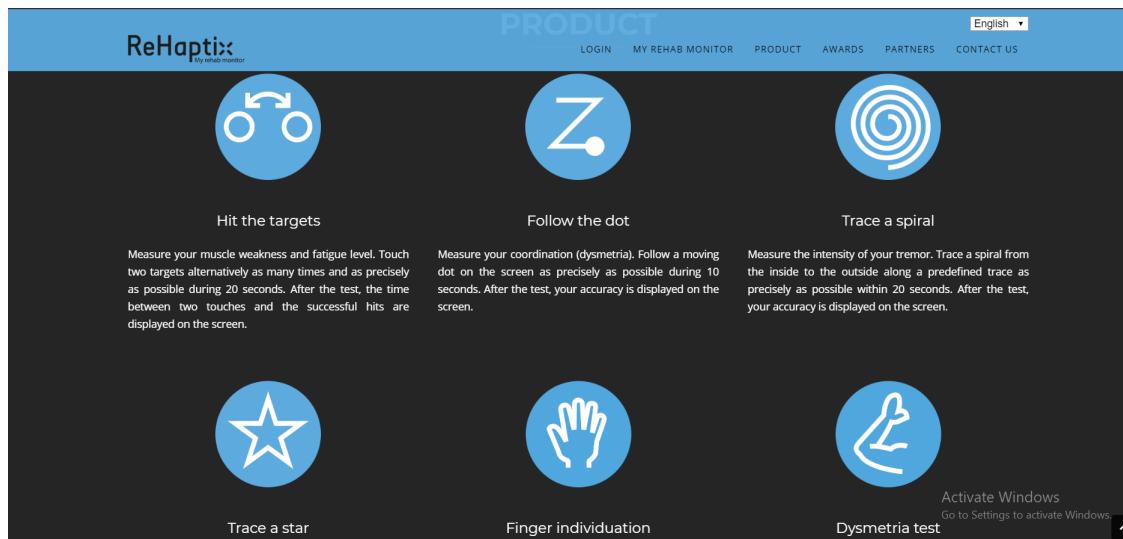
Not yet a member?

Register

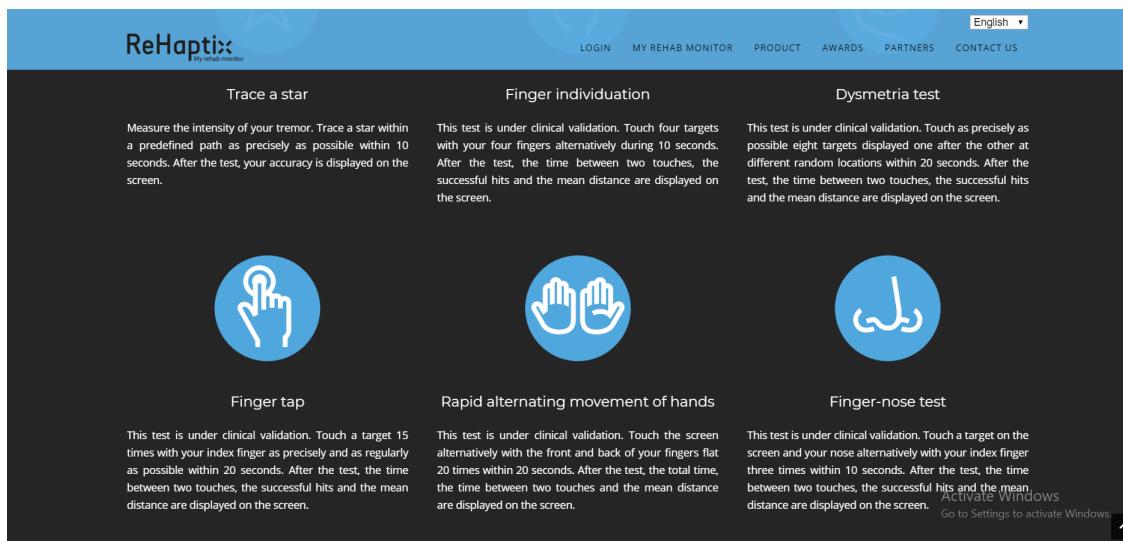
Copyright © 2015 - All Rights Reserved - Rehaptix.com
[General Terms and Conditions](#) and [Privacy Policy](#)

Slika 2.2: Login

Kao i naša aplikacija ona omogućava komunikaciju između klijenta i bolnice, prilagođavanje programa rehabilitacije klijentima na temelju njihovog napretka i trenutnog stanja. Dodatna funkcionalnost koju ova aplikacija nudi je testiranje vlastitih sposobnosti.



Slika 2.3: Testiranja



Slika 2.4: Testiranja

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Klijenti
2. Zaposlenici klinike:
 - (a) Doktori
 - (b) Treneri
3. Razvojni tim (grupa Flow)
4. Naručitelj (asistent Hrvoje Nuić)
5. Administratori aplikacije

Aktori i njihovi funkcionalni zahtjevi:

1. Administrator (inicijator) može:
 - (a) potvrditi ili odbiti željenu ulogu neregistriranom korisniku
 - (b) dodavati proizvode i vježbe u kategorije
 - (c) uređivati i brisati
 - i. kategorije
 - ii. proizvode
 - iii. vježbe
2. Doktor (inicijator) može:
 - (a) odgovarati na recenzije svojih klijenata
 - (b) prekinuti suradnju s klijentom
 - (c) definirati dijetu svome klijentu

- (d) pregledati statistiku svog klijenta
- (e) dodavati proizvode i vježbe u kategorije

3. Trener (inicijator) može:

- (a) odgovarati na recenzije svojih klijenata
- (b) prekinuti suradnju s klijentom
- (c) definirati treninge svome klijentu
- (d) pregledati statistiku svog klijenta
- (e) dodavati proizvode i vježbe u kategorije

4. Klijent (inicijator) može:

- (a) tražiti i pregledavati profile svih dostupnih doktora i trenera
- (b) ostaviti recenziju svom doktoru ili treneru
- (c) prekinuti suradnju sa svojim doktorom ili trenerom
- (d) pregledati vlastitu statistiku
- (e) provjeriti uklapa li se neki proizvod u dijetu skeniranjem bar koda ili ručnim unosom

5. Neregistrirani korisnik (inicijator) može:

- (a) registrirati se kao klijent
- (b) poslati zahtjev za registraciju sa željenom ulogom
 - i. trener
 - ii. doktor

6. Baza podataka (sudionik) može:

- (a) pohranjuje podatke o svim:
 - i. korisnicima i njihovim ovlastima
 - ii. recenzijama
 - iii. dijetama i proizvodima
 - iv. treninzima i vježbama

3.1.1 Obrasci uporabe

UC1 - Registracija klijenta

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu kao klijent
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju klijenata
 2. Korisnik unosi potrebne podatke
 3. Korisnik potvrđuje podatke
 4. Sustav stvara novi korisnički račun i obavještava korisnika da je registracija uspješna
- **Opis mogućih odstupanja:**
 - 3.a Uneseni podatci nisu ispravni
 1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja/dodaje potrebne podatke te završava unos ili odustaje od registracije

UC2 - Registracija zaposlenika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu kao zaposlenik
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju zaposlenika
 2. Korisnik unosi potrebne podatke
 3. Korisnik potvrđuje podatke

4. Administrator potvrđuje zahtjev za registracijom
 5. Sustav stvara novi korisnički račun
- **Opis mogućih odstupanja:**
 - 3.a Uneseni podatci nisu ispravni
 1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja/dodaje potrebne podatke te završava unos ili odustaje od registracije
 - 4.a Administrator odbija zahtjev
 1. Korisnik ne dobiva pristup sustavu

UC3 - Prijava u sustav

- **Glavni sudionik:** Neprijavljeni korisnik
 - **Cilj:** Dobiti pristup korisničkom sučelju
 - **Sudionici:** Baza podataka
 - **Preduvjet:** Korisnik je registriran
 - **Opis osnovnog tijeka:**
 1. Korisnik unosi podatke potrebne za prijavu
 2. Korisnik potvrđuje unos
 3. Sustav prijavljuje korisnika i prikazuje korisničko sučelje
- **Opis mogućih odstupanja:**
 - 2.a Uneseni podatci nisu ispravni
 1. Sustav obavještava korisnika o neuspjeloj prijavi i vraća ga na stranicu za prijavu
 - 3.a Administrator nije još potvrdio korisnikov zahtjev za registraciju
 1. Sustav obavještava korisnika da njegov zahtjev još nije prihvaćen

UC4 - Potvrda zahtjeva neregistriranog korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi zahtjev neregistriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz zahtjeva za registraciju
 2. Korisnik potvrđuje željeni zahtjev
 3. Sustav stvara novi korisnički račun za osobu koja je poslala zahtjev
- **Opis mogućih odstupanja:**
 - 2.a Zahtjev nije valjan
 1. Korisnik odbija zahtjev
 2. Sustav briše odbijeni zahtjev
 3. Sustav ponovo prikazuje zahtjeve za registracijom

UC5 - Slanje zahtjeva za suradnju

- **Glavni sudionik:** Klijent
- **Cilj:** Započeti suradnju sa željenim zaposlenikom
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled zaposlenika
 2. Korisnik odabire opciju za slanje zahtjeva željenom zaposleniku
 3. Zaposlenik kojem je slan zahtjev potvrđuje korisnika
 4. Korisnik i zaposlenik sada su u suradnji i sustav pohranjuje promjene u bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Zaposlenik odbija zahtjev

(a) Sustav obavještava korisnika da je njegov zahtjev odbijen

UC6 - Potvrda zahtjeva za suradnju

- **Glavni sudionik:** Doktor, trener
- **Cilj:** Potvrditi zahtjev za suradnju koji je poslao klijent
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled zahtjeva za suradnju
 2. Korisnik potvrđuje željeni zahtjev
 3. Korisnik i klijent sada su u suradnji i sustav pohranjuje promjene u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Zahtjev nije valjan
 1. Korisnik odbija zahtjev
 2. Sustav javlja klijentu da je njegov zahtjev odbijen
 3. Sustav ponovno prikazuje pregled zahtjeva za suradnjom

UC7 - Prekid suradnje

- **Glavni sudionik:** Klijent, doktor, trener
- **Cilj:** Prekinuti suradnju s osobom
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i u suradnji s barem jednom osobom
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled svojih suradnika
 2. Korisnik odabire opciju za prekid suradnje s željenom osobom
 3. Korisnik i odabrana osoba više nisu u suradnji i sustav pohranjuje promjene u bazu podataka

UC8 - Ocjenjivanje zaposlenika

- **Glavni sudionik:** Klijent
- **Cilj:** Napisati recenziju zaposlenika s kojim je klijent u suradnji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i u suradnji je s barem jednom osobom
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz osoba s kojima je u suradnji
 2. Korisnik odabire kojeg zaposlenika želi ocijeniti
 3. Korisnik izabire ocjenu i piše komentar
 4. Korisnik potvrđuje unos
 5. Sustav pohranjuje recenziju u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Nisu uneseni svi potrebni podatci
 1. Sustav obavještava korisnika da treba ispuniti sve potrebne podatke i vraća ga na stranicu za pisanje recenzije

UC9 - Odgovor na vlastitu recenziju

- **Glavni sudionik:** Doktor, trener
- **Cilj:** Odgovoriti na klijentovu recenziju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i barem jedan klijent je napisao recenziju za njega
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled svojih recenzija
 2. Korisnik odabire opciju za odgovoriti na željenu recenziju
 3. Korisnik unosi potrebne podatke
 4. Korisnik potvrđuje unos
 5. Sustav pohranjuje odgovor u bazu podataka

- **Opis mogućih odstupanja:**

4.a Nisu uneseni svi potrebni podatci

1. Sustav obavještava korisnika da treba ispuniti sve potrebne podatke i vraća ga na stranicu za odgovor na recenziju

UC10 - Definiranje dijete

- **Glavni sudionik:** Doktor

- **Cilj:** Definirati dijetu svome klijentu

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen i u suradnji je s barem jednim klijentom

- **Opis osnovnog tijeka:**

1. Korisnik odabire pregled svojih klijenata

2. Korisnik odabire opciju za definiranje dijete željenom klijentu

3. Korisnik unosi potrebne podatke

4. Korisnik potvrđuje unos

5. Sustav pohranjuje promjene u bazu podataka

- **Opis mogućih odstupanja:**

4.a Nisu uneseni svi potrebni podatci

1. Sustav obavještava korisnika da treba ispuniti sve potrebne podatke i vraća ga na stranicu za definiranje dijete

UC11 - Definiranje treninga

- **Glavni sudionik:** Trener

- **Cilj:** Definirati trening svome klijentu

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen i u suradnji s barem jednim klijentom
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz svojih klijenata
 2. Korisnik odabire opciju za definiranje treninga željenom klijentu
 3. Korisnik unosi potrebne podatke
 4. Korisnik potvrđuje unos
 5. Sustav pohranjuje promjene u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Nisu uneseni svi potrebni podatci
 1. Sustav obavještava korisnika da treba ispuniti sve potrebne podatke i vraća ga na stranicu za definiranje treninga

UC12 - Pregled statistike klijenta

- **Glavni sudionik:** Doktor, trener
- **Cilj:** Pregledati statustiku klijenta po danu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i u suradnji je s barem jednim klijentom
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled svojih klijenata
 2. Korisnik odabire opciju za pregled statistike željenog klijenta
 3. Korisnik odabire za koji dan se prikazuje statistika
 4. Sustav prikazuje statistiku klijenta za taj dan
- **Opis mogućih odstupanja:**
 - 4.a Klijent nema upisanih podataka

1. Sustav obavještava da klijent nema upisanih podataka za taj dan i nudi opciju za povratak

UC13 - Pregled svoje statistike

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati svoju statistiku po danu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prikaz statistike
 2. Korisnik odabire za koji dan se prikazuje statistika
 3. Sustav prikazuje statistiku za taj dan
- **Opis mogućih odstupanja:**
 - 3.a Klijent nema upisanih podataka
 1. Sustav obavještava da nema upisanih podataka i nudi opciju za povratak

UC14 - Dodavanje proizvoda u kategoriju

- **Glavni sudionik:** Doktor, trener, administrator
- **Cilj:** Dodati proizvod u željenu kategoriju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz kategorija proizvoda
 2. Korisnik odabire željenu kategoriju
 3. Korisnik dodaje željene proizvode u kategoriju
 4. Sustav pohranjuje promjene bazu podataka

UC15 - Dodavanje vježbe u kategoriju

- **Glavni sudionik:** Doktor, trener, administrator

- **Cilj:** Dodati vježbu u željenu kategoriju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz kategorija vježbi
 2. Korisnik odabire željenu kategoriju
 3. Korisnik dodaje željene vježbe u kategoriju
 4. Sustav pohranjuje promjene u bazu podataka

UC16 - Stvaranje kategorije

- **Glavni sudionik:** Administrator
- **Cilj:** Stvoriti novu kategoriju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz kategorija
 2. Korisnik odabire opciju za kreiranje nove kategorije
 3. Korisnik unosi potrebne podatke o kategoriji i njen sadržaj
 4. Korisnik potvrđuje unos
 5. Sustav pohranjuje novu kategoriju u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Nisu uneseni svi potrebni podaci
 1. Sustav obavještava korisnika da treba unijeti sve potrebne podatke i vraća ga na stranicu za stvaranje kategorije

UC17 - Uređivanje kategorija

- **Glavni sudionik:** Administrator
- **Cilj:** Izmijenti sadržaj kategorije i podatke o kategoriji
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i postoji barem jedna kategorija

- **Opis osnovnog tijeka:**

1. Korisnik odabire prikaz kategorija
2. Korisnik odabire kategoriju koju želi izmijeniti
3. Korisnik mijenja podatke o kategoriji i njen sadržaj
4. Korisnik potvrđuje izmjenu
5. Promjene se pohranjuju u bazu podataka

- **Opis mogućih odstupanja:**

- 4.a Izmijenjeni podatci nisu ispravni
 1. Sustav obavještava korisnika da mora unijeti ispravne podatke i vraća ga na stranicu za uređivanje kategorije

UC18 - Brisanje kategorije

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati kategoriju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i postoji barem jedna kategorija
- **Opis osnovnog tijeka:**

1. Korisnik odabire prikaz kategorija
2. Korisnik odabire opciju za brisanje željene kategorije
3. Sustav briše kategoriju i pohranjuje promjene u bazu podataka

UC19 - Dodavanje proizvoda

- **Glavni sudionik:** Administrator
 - **Cilj:** Dodati novi proizvod u sustav
 - **Sudionici:** Baza podataka
 - **Preduvjet:** Korisnik je prijavljen
 - **Opis osnovnog tijeka:**
1. Korisnik odabire prikaz proizvoda
 2. Korisnik odabire opciju za dodavanje novog proizvoda

3. Korisnik unosi sve potrebne podatke
 4. Korisnik potvrđuje unos
 5. Sustav dodaje proizvod i pohranjuje ga u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Nisu uneseni svi potrebni podatci
 1. Sustav obavještava korisnika da treba unijeti sve potrebne podatke i vraća ga na stranicu za dodavanje proizvoda

UC20 - Uređivanje proizvoda

- **Glavni sudionik:** Administrator
- **Cilj:** Izmijeniti podatke o željenom proizvodu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodan je barem jedan proizvod
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz proizvoda
 2. Korisnik odabire opciju za uređivanje željenog proizvoda
 3. Korisnik mijenja podatke o proizvodu
 4. Korisnik potvrđuje izmjenu
 5. Sustav pohranjuje promjenu u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Izmijenjeni podatci nisu ispravni
 1. Sustav obavještava korisnika da mora unijeti ispravne podatke i vraća ga na stranicu za uređivanje proizvoda

UC21 - Brisanje proizvoda

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati proizvod
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen i dodan je barem jedan proizvod
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz proizvoda
 2. Korisnik odabire opciju za brisanje željenog proizvoda
 3. Sustav briše proizvod i pohranjuje promjene u bazu podataka

UC22 - Dodavanje vježbe

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati novu vježbu u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prikaz vježbi
 2. Korisnik odabire opciju za dodavanje vježbe
 3. Korisnik unosi sve potrebne podatke
 4. Korisnik potvrđuje unos
 5. Sustav stvara novu vježbu i pohranjuje izmijene u bazu podataka
- **Opis mogućih odstupanja:**
 - 4.a Nisu uneseni svi potrebni podaci
 1. Sustav obavještava korisnika da treba unijeti sve potrebne podatke i vraća ga na stranicu za dodavanje vježbe

UC23 - Uređivanje vježbe

- **Glavni sudionik:** Administrator
- **Cilj:** Izmijeniti podatke o željenoj vježbi
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodana je barem jedna vježba

- **Opis osnovnog tijeka:**

1. Korisnik odabire prikaz vježbi
2. Korisnik odabire opciju za uređivanje željenog proizvoda
3. Korisnik mijenja podatke o vježbi
4. Korisnik potvrđuje izmjenu
5. Sustav pohranjuje promjenu u bazu podataka

- **Opis mogućih odstupanja:**

- 4.a Izmijenjeni podatci nisu ispravni
 1. Sustav obavještava korisnika da mora unijeti ispravne podatke i vraća ga na stranicu za uređivanje vježbe

UC24 - Brisanje vježbe

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati vježbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodana je barem jedna vježba
- **Opis osnovnog tijeka:**

1. Korisnik odabire prikaz vježbi
2. Korisnik odabire opciju za brisanje željene vježbe
3. Sustav briše vježbu i promijene se pohranjuju u bazu podataka

UC25 - Provjera proizvoda skeniranjem

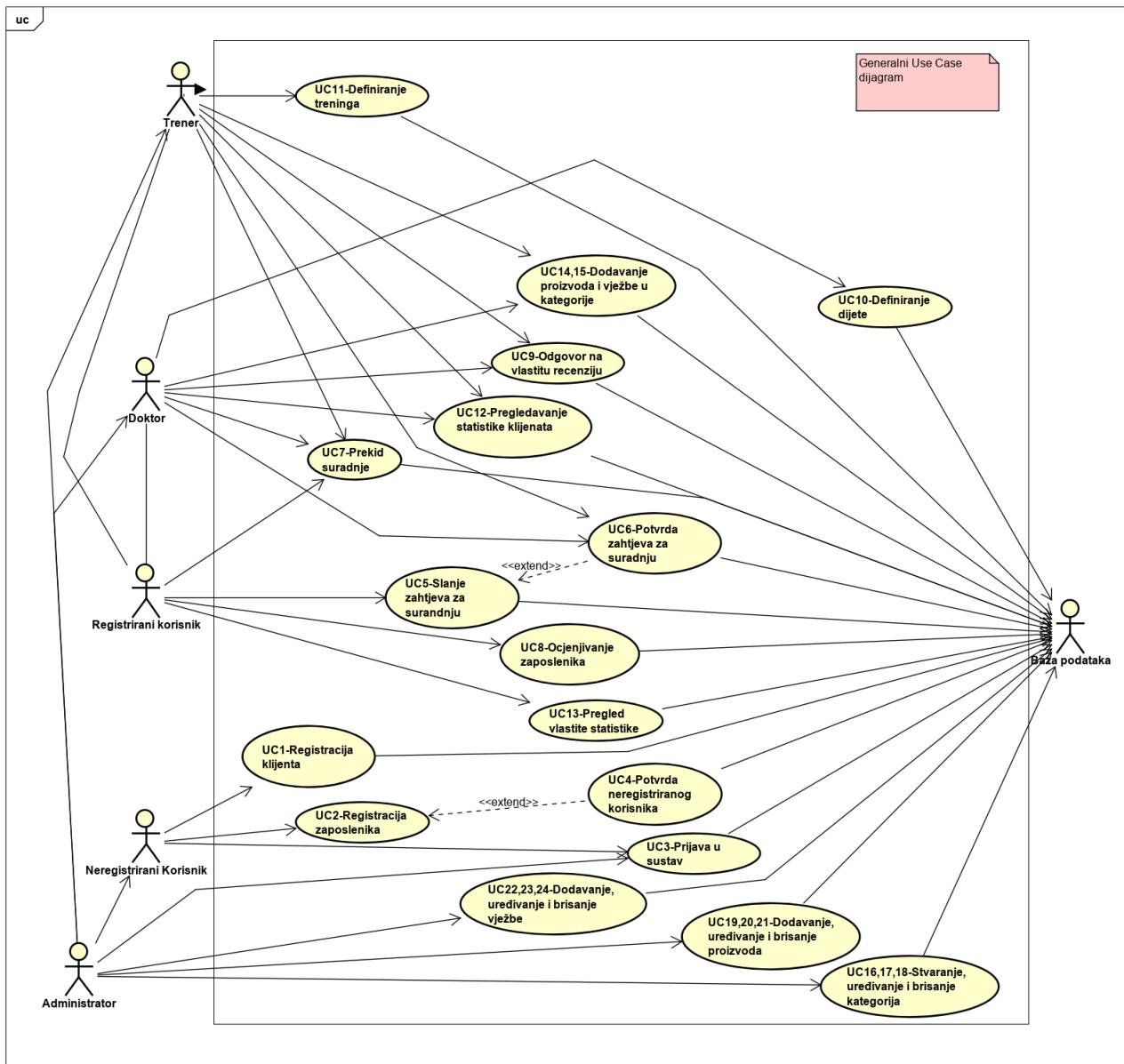
- **Glavni sudionik:** Klijent
 - **Cilj:** Skenirati proizvod i odrediti uklapa li se u njegovu dijetu
 - **Sudionici:** Baza podataka
 - **Preduvjet:** Korisnik je prijavljen i pripisana mu je dijeta
 - **Opis osnovnog tijeka:**
1. Korisnik odabire opciju za provjeru proizvoda
 2. Korisnik odabire skeniranje bar koda kao način provjere

3. Korisnik fotografira bar kod proizvoda
 4. Sustav javlja korisniku uklapa li se proizvod u dijetu za taj dan
- **Opis mogućih odstupanja:**
 - 2.a Bar kod nije prepoznat u slici
 1. Sustav javlja da je potrebno jasnije skenirati bar kod
 2. Korisnik ponovo skenira proizvod
 - 2.b Sustav ne prepoznae proizvod
 1. Sustav javlja da nema podatke o traženom proizvodu
 2. Sustav vraća korisnika na početno sučelje

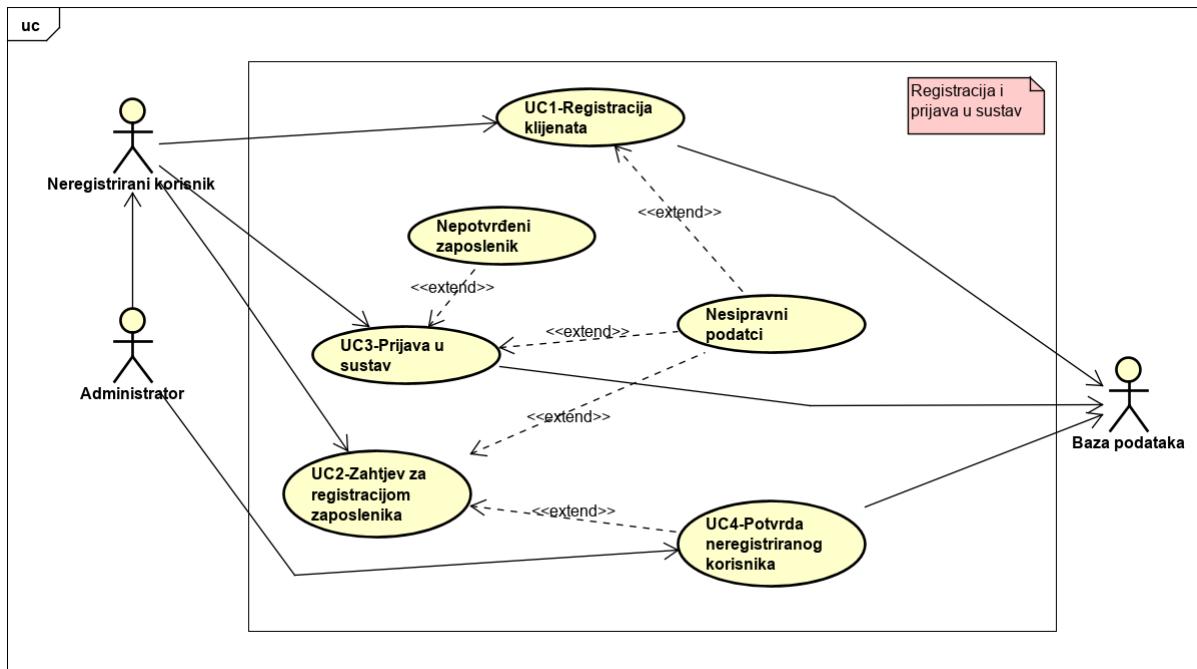
UC26 - Provjera proizvoda ručnim unosom

- **Glavni sudionik:** Klijent
- **Cilj:** Unijeti masu proizvoda i odrediti uklapa li se u njegovu dijetu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i pripisana mu je dijeta
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju provjere proizvoda
 2. Korisnik odabire ručni unos kao način provjere
 3. Korisnik odabire željeni proizvod i unosi količinu(masu) koju želi konzumirati
 4. Sustav javlja korisniku uklapa li se proizvod u dijetu za taj dan

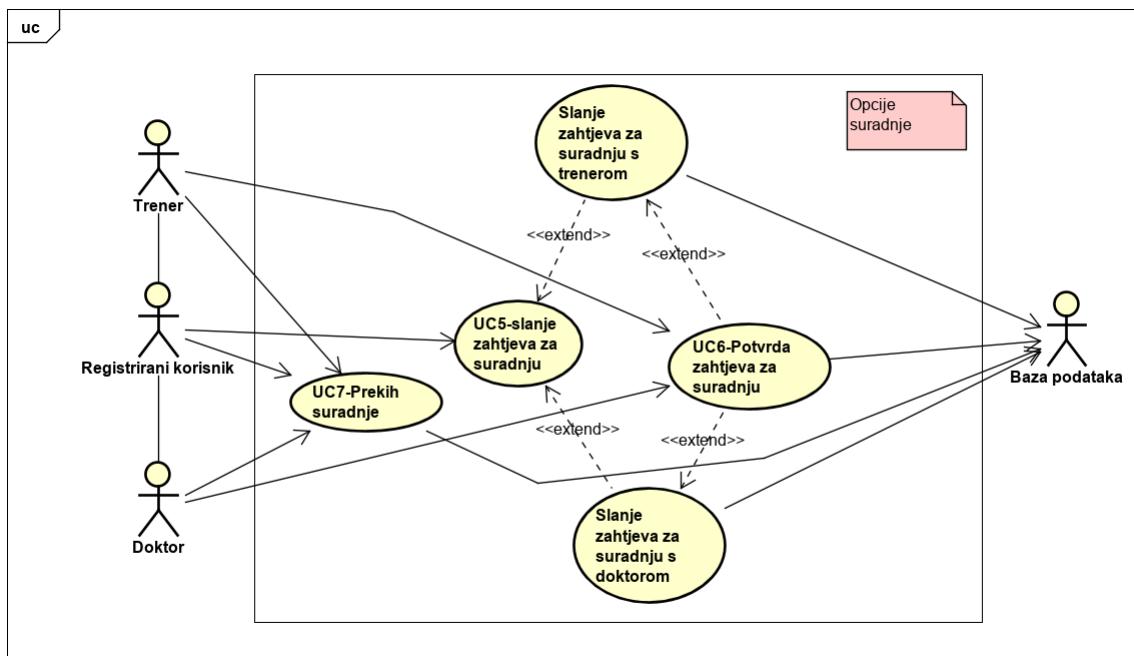
Dijagrami obrazaca uporabe



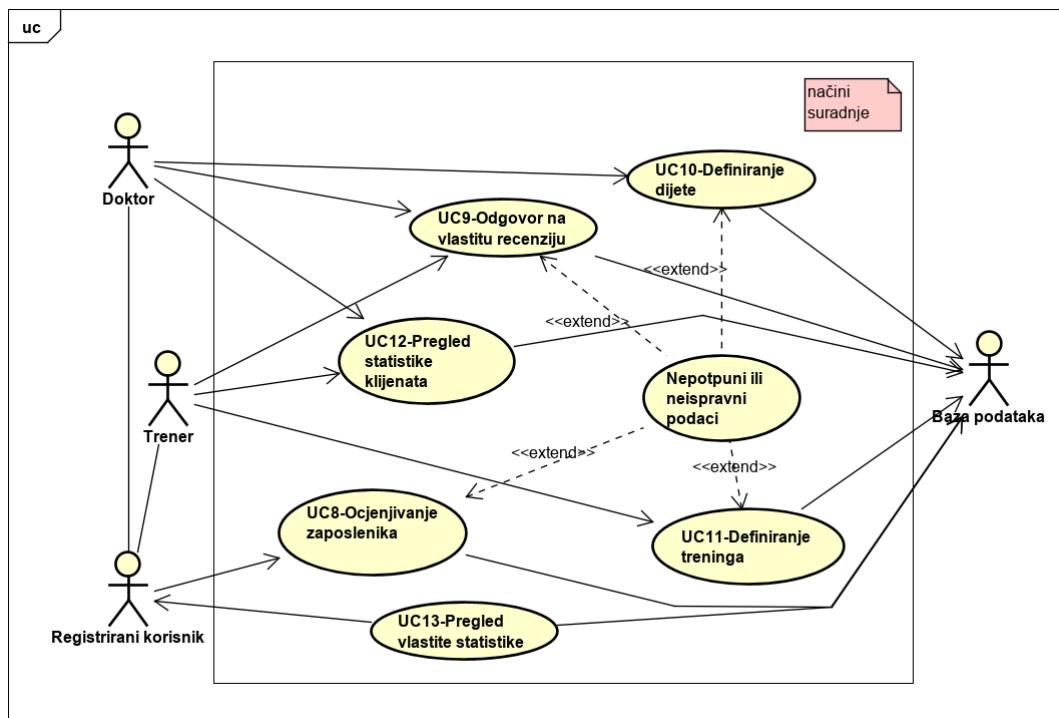
Slika 3.1: Generalni dijagram obrasca uporabe



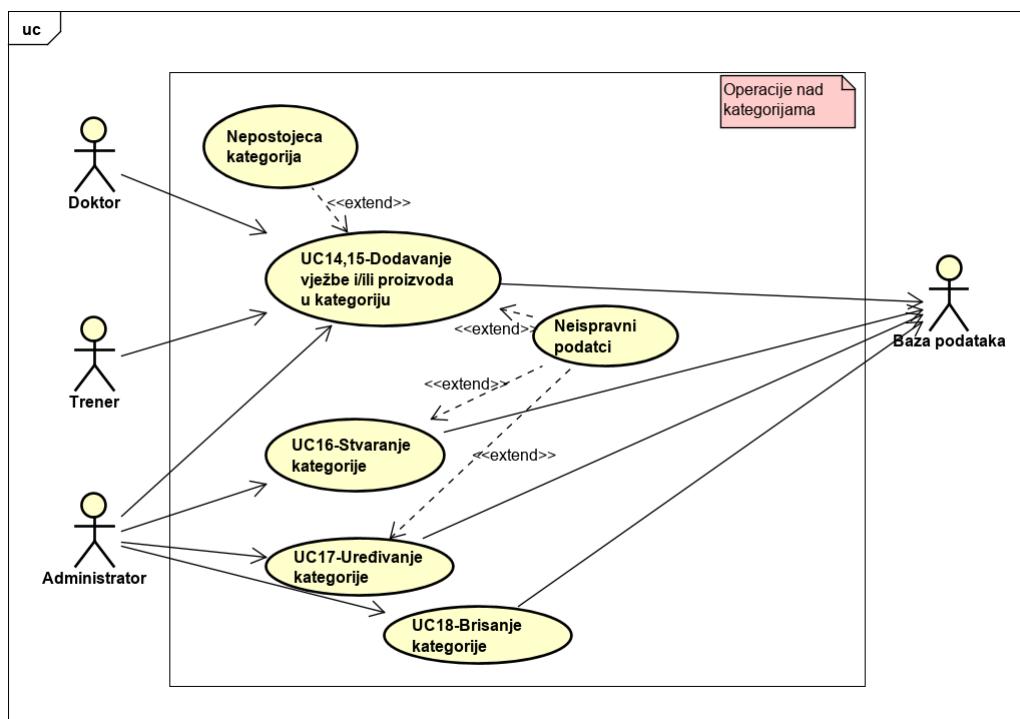
Slika 3.2: Dijagram obrasca uporabe - registracija i prijava



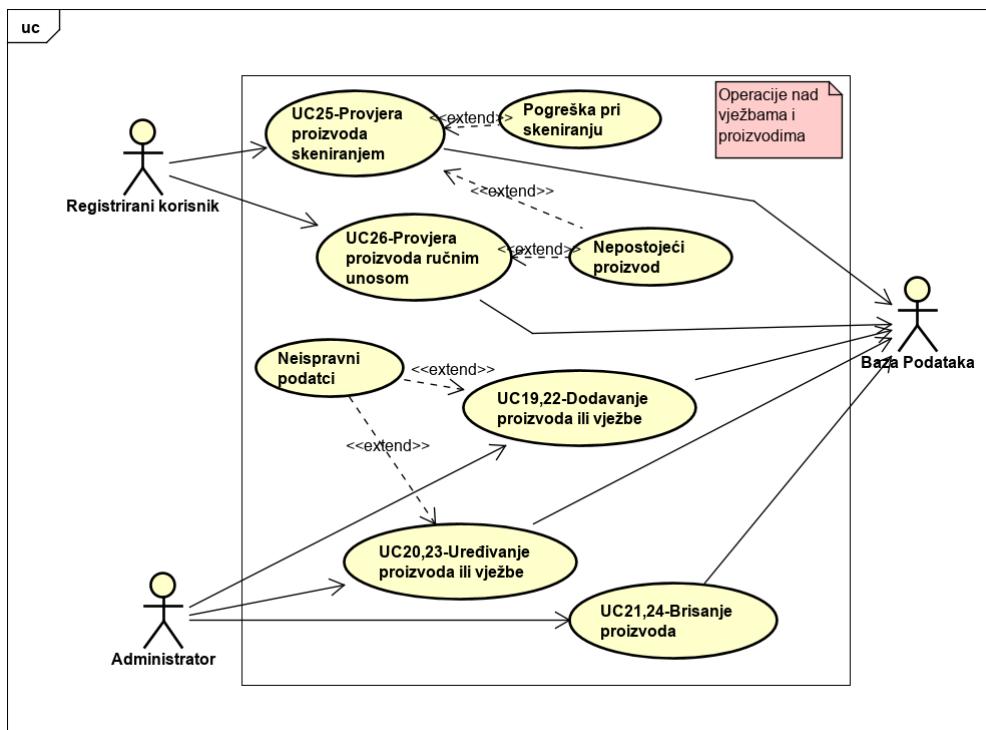
Slika 3.3: Dijagram obrasca uporabe - opcije suradnje



Slika 3.4: Dijagram obrasca uporabe - načini suradnje



Slika 3.5: Dijagram obrasca uporabe - operacije nad kategorijama

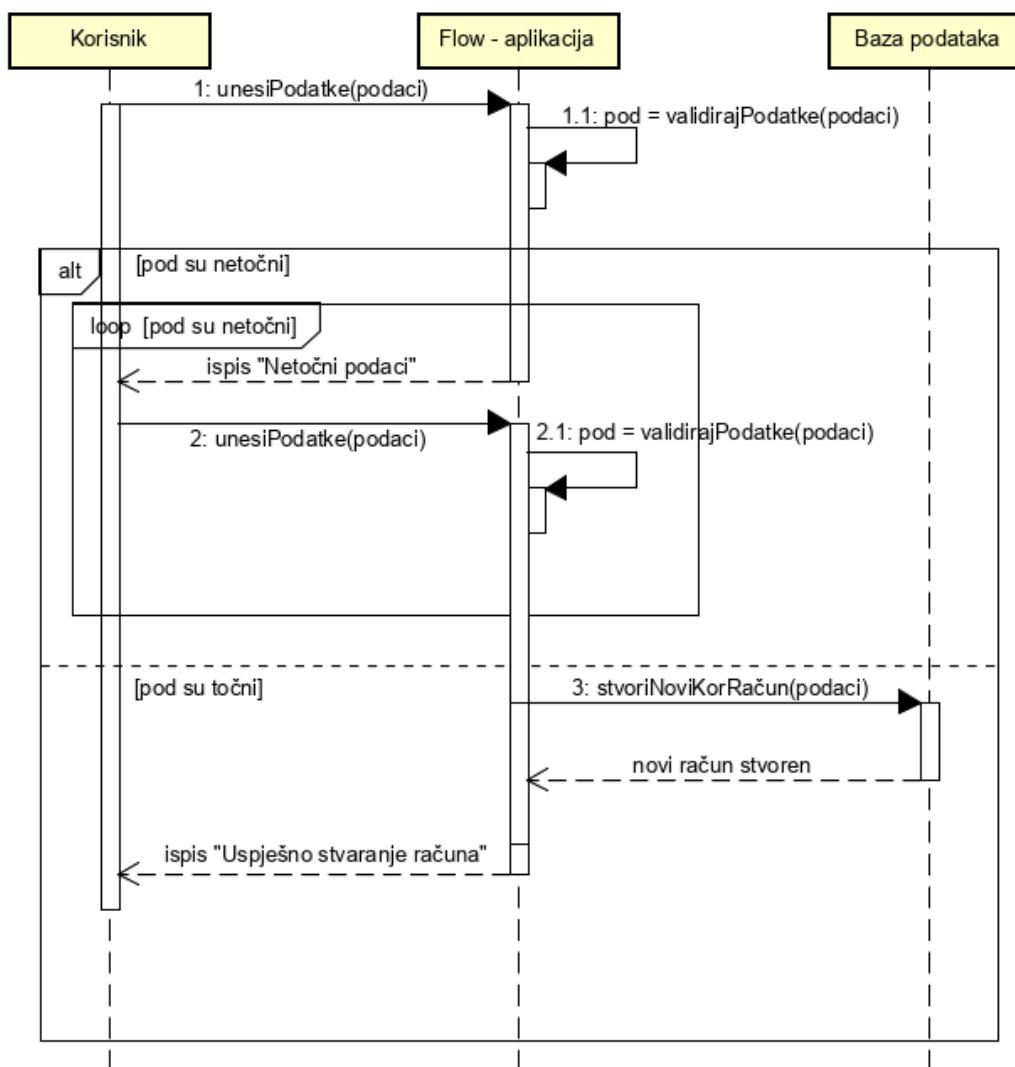


Slika 3.6: Dijagram obrasca uporabe - operacije nad vježbama i proizvodima

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC1 - Registracija klijenta

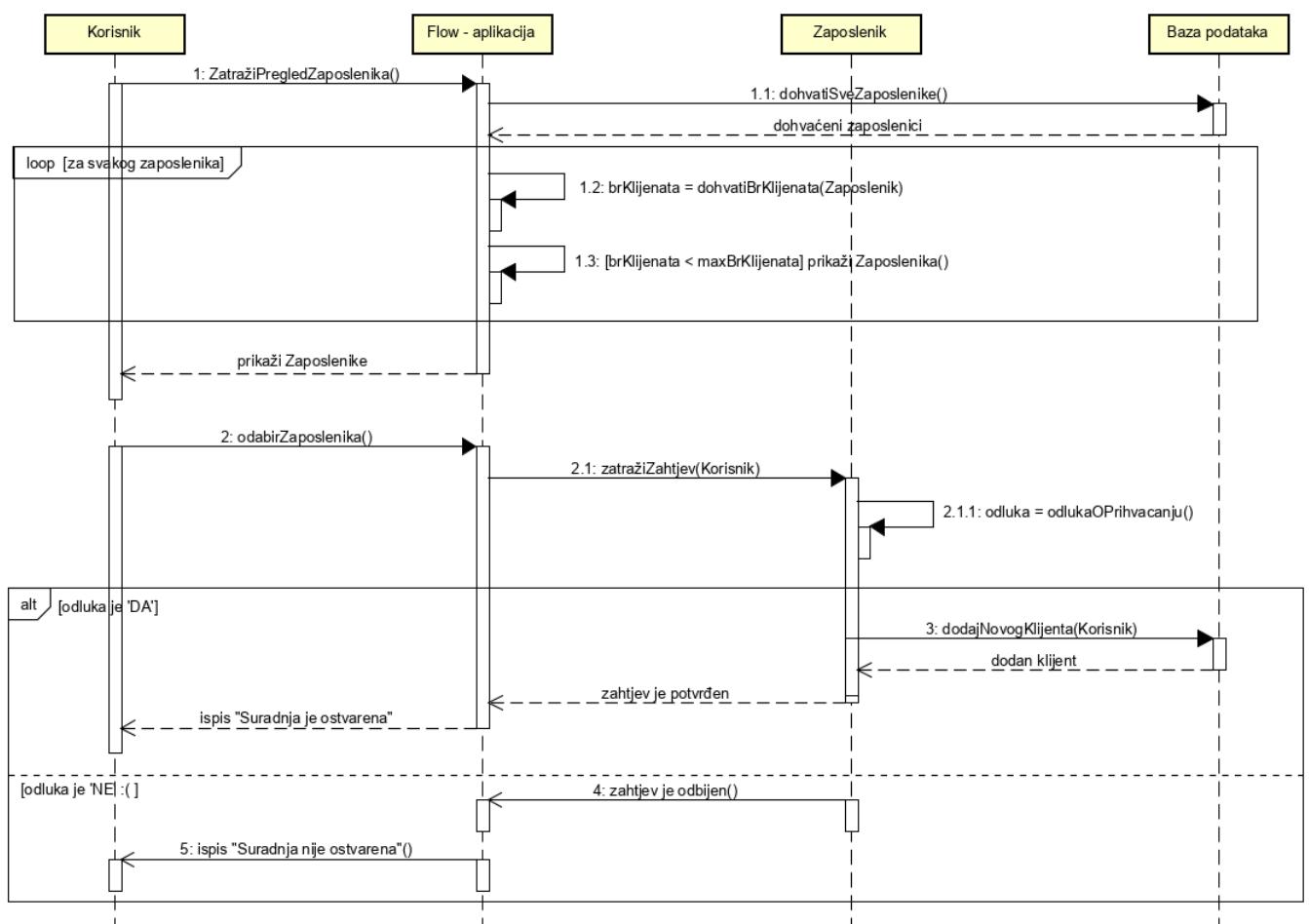
Neregistrirani korisnik šalje podatke za registraciju. Aplikacija ih zatim validira i ustanavlja jesu li ispravni ili ne. Ako nisu, aplikacija obavještava korisnika i traži ponovan unos sve dok ne unese ispravne podatke. Za unos ispravnih podataka stvara se novi korisnički račun u bazi te se korisnika obavještava o uspješnoj registraciji.



Slika 3.7: Sekvencijski dijagram za UC1

Obrazac uporabe UC5 - Slanje zahtjeva za suradnju i UC6 - Potvrda zahtjeva za suradnju

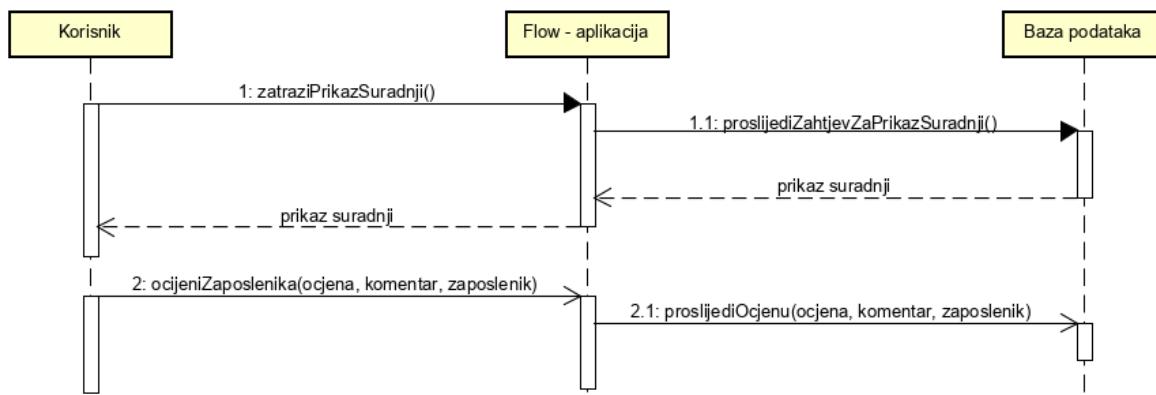
Korisnik traži pregled svih zaposlenika. Aplikacija zaposlenicima koji su dohvaćeni iz baze provjerava broj klijenata. Ukoliko je on manji od njegovog maksimalnog broja klijenata, aplikacija ga prikazuje. Potom korisnik odabere zaposlenika kojem želi poslati zahtjev za suradnju. Aplikacija mu taj zahtjev prosljeđuje. On ima mogućnost prihvaćanja ili odbijanja suradnje. Ako ju prihvati, podaci o novom klijentu zaposlenika se spremaju u bazu te korisnik dobiva poruku o prihvaćanju suradnje. Nadalje, ako zaposlenik odbije suradnju, poruka s prikladnim opisom se prikazuje korisniku.



Slika 3.8: Sekvencijski dijagram za UC5 i UC6

Obrazac uporabe UC8 - Ocjenjivanje zaposlenika

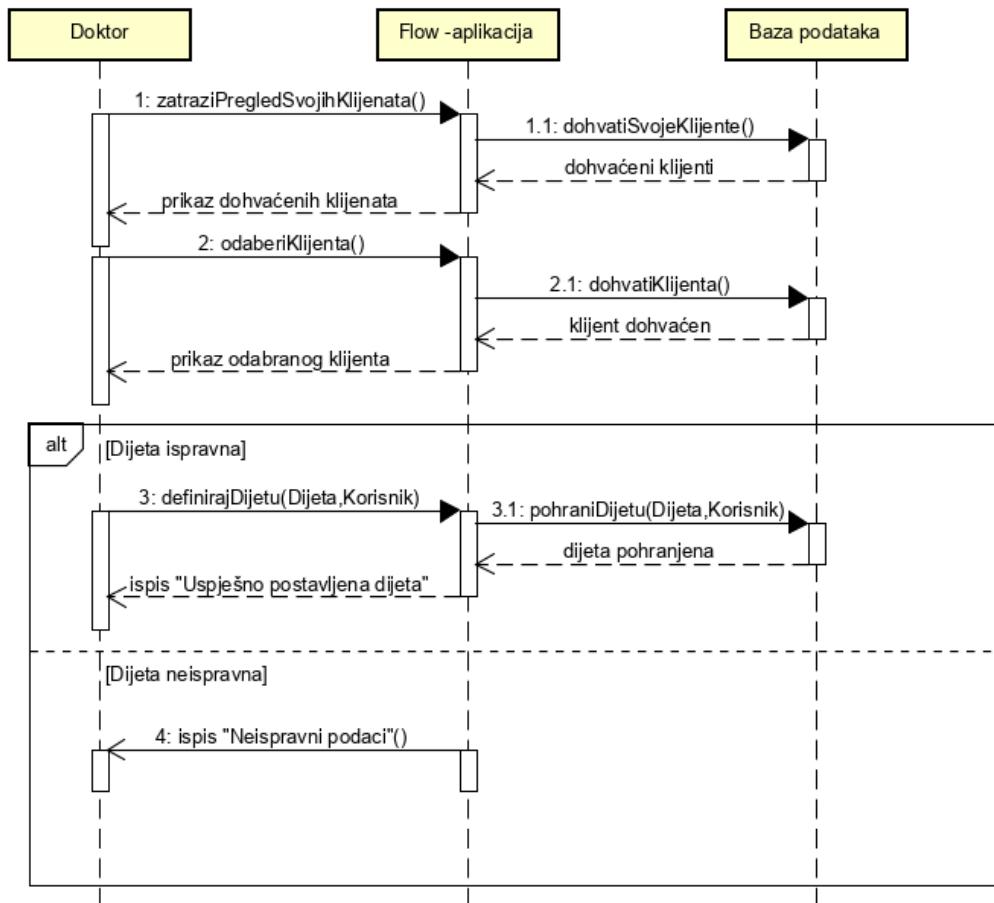
Korisnik traži prikaz svojih suradnji te nakon dobivenog prikaza oda-bire zaposlenika kojem želi ostaviti recenziju. U recenziji ostavlja komentar i daje ocjenu. Ti podaci se potom spremaju u bazu.



Slika 3.9: Sekvencijski dijagram za UC8

Obrazac uporabe UC10 - Definiranje dijete

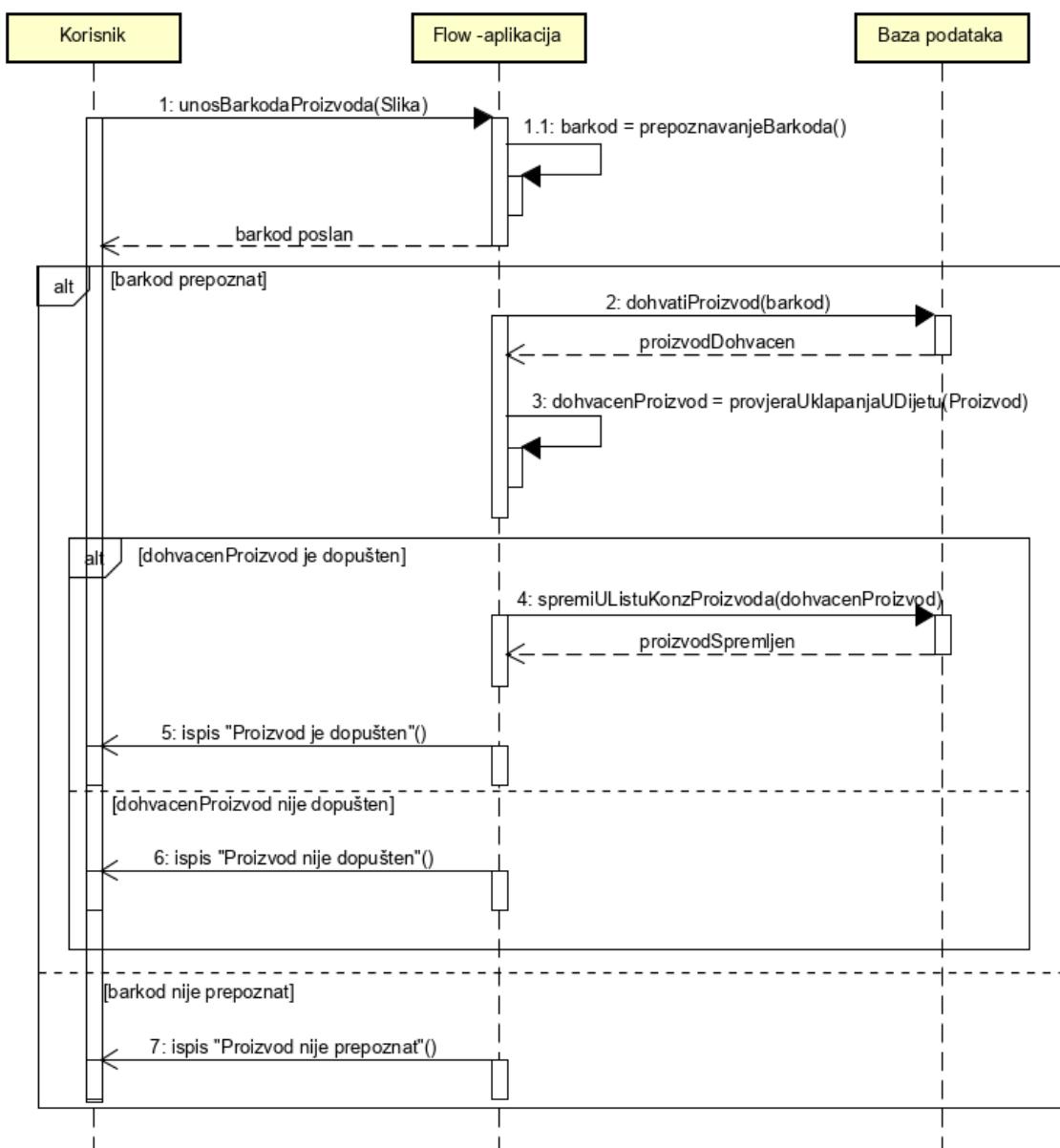
Zaposlenik traži pregled svih svojih klijenata. Nakon njihovog prikaza odabere onog kojem želi postaviti dijetu. On tu dijetu definira. Ako je dijeta dobro definirana ti podaci se proslijede u bazu i ispisuje se poruka o uspješnom postavljanju dijete. Ako dijeta nije dobro definirana ispisuje se samo prikladna poruka.



Slika 3.10: Sekvencijski dijagram za UC10

Obrazac uporabe UC25 - Provjera proizvoda skeniranjem

Korisnik unese sliku bar koda. Aplikacija provjerava je li bar kod uspješno očitan. Ako je, proizvod se dohvaća iz baze te provjerava je li korisniku dopušteno njegovo konzumiranje.U slučaju da bar kod nije uspješno učitan, ispisuje se prikladna poruka. Ako je proizvod dopušten, ispisuje se prikladna poruka i proizvod se sprema u listu konzumiranih proizvoda. Ako korisniku nije dopušteno konzumiranje proizvoda, samo mu se ispisuje poruka o tome.



Slika 3.11: Sekvencijski dijagram za UC25

3.2 Ostali zahtjevi

- Pristup aplikaciji mora biti omogućen iz javne mreže pomoću HTTPS-a.
- Veza s bazom podataka mora biti sigurna i kvalitetna. Također veza mora biti brza.
- Nadogradnja sustava mora biti moguća.
- Sustav mora biti jednostavan za korištenje.
- Kratke upute o korištenju moraju biti omogućene.
- Obavijestiti korisnika o neispravnom korištenju sučelja.
- Neispravno korištenje sučelja ne smije narušiti rad aplikacije.
- Sustav mora biti implementiran kao web-aplikacija pomoću objektno orijentiranih jezika.
- Sustav mora podržavati rad više korisnika u istome trenutku.
- Aplikacija mora podržavati hrvatsku abecedu.

4. Arhitektura i dizajn sustava

Arhitektura se sastoji od 3 podsustava:

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

Web poslužitelj omogućuje komunikaciju klijenta s aplikacijom, pomoću HTTP (Hyper Text Transfer Protocol) protokola. HTTP protokol je protokol aplikacijskog sloja koji služi za razmjenu svih podataka (HTML stranica, slika, itd.) na Internetu. Dvije glavne metode HTTP protokola su GET i POST metode. GET (HTTP GET) metoda se koristi kao zahtjev za podacima na serveru, a POST (HTTP POST) metoda se koristi za kreiranje ili ažuriranje podataka na serveru. Uspostava veze se odvija tako da se klijent spoji na port 80 i web poslužitelju uputi zahtjev za traženom stranicom, a on mu vraća HTML kod tj. HTML stranicu ili ju izgenerira. Web poslužitelj zapravo pokreće aplikaciju te joj proslijedi zahtjev. Klijent prima HTML kod te ga prikazuje u obliku stranice. Sam korisnik upućuje zahtjev preko web preglednika. Web aplikacija služi za obrađivanje željenih zahtjeva. Kroz HTTP zahtjeve ćemo dohvaćati podatke u formatu JSON, a u web aplikaciji ih obrađivati te preslikavati u bazu podataka. Bazu podataka ćemo napravili u PostgreSQLu, te smo tablice iz baze, pomoću paketa JPA i Hibernate (paketi Springa), povezali sa klasama u aplikaciji. Za izradu backend dijela aplikacije ćemo koristiti Java Spring Boot, a frontend ćemo izraditi u Reactu. Za razvojno

okruženje smo odabrali IntelliJ. Arhitektura aplikacije će se temeljiti na MVC (Model-View-Controller) konceptu, koji Spring podržava.

MVC koncept sastoji se od:

- Model- Središnja komponenta, koja prima ulazne podatke od Controllera. Model predstavlja dinamičke strukture podataka, neovisne o korisničkom sučelju. Upravlja podacima, logikom i pravilima aplikacije.
- View- Bilo kakav prikaz podataka.
- Controller- Prima ulaze i prilagođava ih za prosljeđivanje Modelu i Viewu.

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvata podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Doktori-Treneri
- Recenzije
- Klijent-Doktor
- Klijent-Trener
- KategorijeProizvoda

- Proizvod
- NutritivneVrijednosti
- Vježba
- Intenzitet
- BrPotrošenihKalorija
- Trening
- NizVježbi
- Dijeta
- OgraničenjaNaProizvode
- OgraničenjaNaKategorije
- DnevniLimit
- UneseneNutrVrijednosti
- KonzumiraniProizvodi

4.1.1 Opis tablica

Korisnik Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Svaki korisnik ima korisničko ime, lozinku, ime, prezime i naziv uloge. Entitet Korisnik je preko atributa KorisnickoIme u *One-to-One* vezi s DoktoriTreneri, *One-to-One* vezi s KlijentDoktor, *One-to-One* vezi s KlijentTrener, *One-to-Many* vezi s entitetom Training, *One-to-Many* vezi s UneseneNutrVrijednosti, *One-to-Many* vezi s KonzumiraniProizvodi, *One-to-Many* vezi s entitetom Dijeta te *One-to-Many* vezi s entitetom Recenzija.

Korisnik		
KorisnickoIme	VARCHAR	jedinstveni identifikator korisnika
Ime	VARCHAR	ime korisnika
Prezime	VARCHAR	prezime korisnika
Lozinka	VARCHAR	lozinka korisnika
Uloga	VARCHAR	Uloga korisnika: doktor, trener ili klijent.

DoktoriTreneri Da bi neregistrirani korisnik dobio prava doktora i trenera, administrator ga mora potvrditi. Potrebno je dodatno priložiti sliku, email i maksimalni broj korisnika koje želi nadgledati. Navedeni atributi se spremaju u entitet DoktoriTreneri. Entitet DoktoriTreneri je preko atributa KorisnickoIme u *One-to-One* vezi s entitetom Korisnik, *One-to-Many* vezi s KlijentDoktor, *One-to-Many* vezi s KlijentTrener te *One-to-Many* vezi s entitetom Recenzije.

DoktoriTreneri		
korisnickoIme	VARCHAR	jedinstveni identifikator korisnika
Mail	VARCHAR	e-mail adresa doktora/trenera
Slika	BYTEA	fotografija doktora/trenera
BrojKlijenata	INT	trenutni broj prijavljenih klijenata
MaxBrKlijenata	INT	maksimalan broj prijavljenih klijenata

Recenzije Klijent svom doktoru i treneru na profilu može ostaviti recenziju s ocjenom i komentarom, a doktor i trener mogu odgovoriti na vlastitu recenziju. Entitet Recenzije sadrži attribute: korisnickoIme, korImeKlijenta, ocjena, komentar, odgovor i ID komentara. Ovaj entitet je preko atributa KorImeKlijenta u *Many-to-One* vezi s

Korisnik, te preko atributa KorisnickoIme u *Many-to-One* vezi s entitetom DoktoriTreneri.

Recenzije		
ID	INT	jedinstveni identifikator recenzije
KorImeKlijenta	VARCHAR	jedinstveni identifikator klijenta
KorisnickoIme	VARCHAR	jedinstveni identifikator doktora/trenera
Ocjena	INT	ocjena od 1 do 5
Komentar	VARCHAR	recenzija doktora/trenera
Odgovor	VARCHAR	odgovor doktora/trenera na recenziju

KlijentDoktor Entitet sadrži informacije o trenutnoj suradnji klijenta i nekog doktora. Atributi su: korImeKlijenta i korImeDoktora. Ovaj entitet je preko atributa KorImeKlijenta u *One-to-One* vezi s Korisnik te preko atributa korImeDoktora u *Many-to-One* vezi s entitetom DoktoriTreneri.

KlijentDoktor		
KorImeKlijenta	VARCHAR	jedinstveni identifikator klijenta
KorImeDoktora	VARCHAR	jedinstveni identifikator doktora

KlijentTrener Entitet sadrži informacije o trenutnoj suradnji klijenta i nekog trenera. Atributi su: korImeKlijenta i korImeTrenera. Ovaj entitet je preko atributa KorImeKlijenta u *One-to-One* vezi s Korisnik te preko atributa korImeTrenera u *Many-to-One* vezi s entitetom DoktoriTreneri.

KlijentTrener		
KorImeKlijenta	VARCHAR	jedinstveni identifikator klijenta

KlijentTrener		
KorImeTrenera	VARCHAR	jedinstveni identifikator trenera

KategorijeProizvoda Kategorija proizvoda sadrži više različitih proizvoda sličnih karakteristika, kao što su na primjer tjestenina, proizvodi od mlijeka, meso itd. Entitet KategorijeProizvoda sadrži atribute: ID kategorije i ime kategorije. Ovaj entitet je preko atributa ID u *Many-to-Many* vezi s entitetom Proizvod te *Many-to-Many* vezi s entitetom OgranicenjaNaKategorije.

KategorijeProizvoda		
ID	INT	jedinstveni identifikator kategorije
Ime	VARCHAR	ime kategorije

Proizvod Svaki proizvod sadrži sliku, masu i prisutne alergene. Svaki proizvod pripada nekoj kategoriji proizvoda. Entitet Proizvod sadrži atribute: ID proizvoda, masa, slika, barkod, prisutni alergeni, ime te ID kategorije. Ovaj entitet je preko ID atributa u *Many-to-Many* vezi s KategorijeProizvoda, *Many-to-Many* vezi s Ogranicenja-NaProizvode, *One-to-One* vezi s NutritivneVrijednosti te *Many-to-Many* vezi s KonzumiraniProizvodi.

Proizvod		
ID	INT	jedinstveni identifikator proizvoda
Ime	VARCHAR	ime proizvoda
Masa	DOUBLE	masa proizvoda u gramima
Slika	BYTEA	slika proizvoda
Barkod	BYTEA	barkod proizvoda
Alergeni	VARCHAR	popis prisutnih alergena

Proizvod		
IDkategorije	INT	ID kategorije proizvoda

NutritivneVrijednosti Informacije o nutritivnim vrijednostima proizvoda su definirane na 100g, a to su energija, masnoće, zasićene masne kiseline, ugljikohidrati, šećeri, bjelančevine i sol. Entitet NutritivneVrijednosti sadrži istoimene atribute te dodatno ID proizvoda za koji su vrijednosti definirane. Ovaj entitet je preko atributa IDproizvoda u *One-to-One* vezi s entitetom Proizvod.

NutritivneVrijednosti		
IDproizvoda	INT	jedinstveni identifikator proizvoda
Energija	DOUBLE	energija u kJ
Masnoća	DOUBLE	masnoća u sastavu proizvoda
ZasMasneKiseline	DOUBLE	zasićene masne kiseline u sastavu proizvoda
Ugljikohidrati	DOUBLE	ugljikohidrati u sastavu proizvoda
Šećeri	DOUBLE	šećeri u sastavu proizvoda
Bjelančevine	DOUBLE	bjelančevine u sastavu proizvoda
Sol	DOUBLE	soli u sastavu proizvoda

Vježba Vježba je definirana sa slikom, opisom i informacijama o broju potrošenih kalorija u sat vremena u ovisnosti o 3 razine intenziteta vježbanja (lagano, normalno, teško). Entitet Vježba dodatno sadrži atribut ID, tj. jedinstveni identifikator vježbe. Ovaj entitet je preko atributa ID u *One-to-Many* vezi s entitetom NizVjezbi te *One-to-Many* vezi s entitetom BrPotrosenihKalorija.

Vježba		
ID	INT	jedinstveni identifikator vježbe
Slika	BYTEA	slika vježbe
Opis	VARCHAR	opis vježbe

Intenzitet Entitet Intenzitet se sastoji od atributa: razina intenziteta i šifra intenziteta. Intenzitet može biti: lagan, normalan i težak. Intenzitet je preko atributa Razina u *Many-to-Many* vezi s BrPotrošenihKalorija te *One-to-Many* vezi s entitetom NizVjezbi.

Intenzitet		
Šifra	INTEGER	šifra intenziteta
Razina	VARCHAR	intenzitet: lagano, normalno ili teško

BrPotrošenihKalorija Za svaku vježbu je definiran broj potrošenih kalorija u sat vremena u ovisnosti o 3 razine intenziteta vježbanja (lagano, normalno, teško). Ovaj entitet je preko atributa IDvjezbe u *Many-to-One* vezi s entitetom Vježba te preko atributa Intenzitet u *Many-to-Many* vezi s entitetom Intenzitet.

BrPotrošenihKalorija		
IDvježbe	INT	jedinstveni identifikator vježbe
Intenzitet	VARCHAR	jedinstveni identifikator intenziteta
PotrošeneKalorije	DOUBLE	broj potrošenih kalorija u sat vremena

Trening Trener zadaje trening klijentu svaki dan. Entitet Trening se sastoji od atributa: ID treninga, ID klijenta te datuma. Ovaj entitet

je preko atributa IDklijenta u *Many-to-One* vezi s entitetom Korisnik te preko atributa ID u *One-to-Many* vezi s entitetom NizVjezbi.

Trening		
ID	INT	jedinstveni identifikator treninga
IDklijenta	VARCHAR	jedinstveni identifikator klijenta
Datum	DATETIME	datum treninga
Odrađen	BOOLEAN	klijent mora potvrditi da je odradio trening

NizVježbi Trening se sastoji od niza vježbi koji imaju određeno trajanje i intenzitet. Entitet NizVježbi se sastoji od atributa: ID treninga, ID vježbe, intenzitet vježbe, trajanje vježbe te redni broj vježbe. Ovaj entitet je preko atributa IDtreninga u *Many-to-One* vezi s entitetom Trening, preko atributa IDvježbe u *Many-to-One* vezi s entitetom Vjezba te u *Many-to-One* vezi s entitetom Intenzitet preko atributa Intenzitet.

NizVježbi		
IDtreninga	INT	jedinstveni identifikator treninga
RedniBroj	INT	redni broj vježbe
IDvježbe	INT	jedinstveni identifikator vježbe
Intenzitet	VARCHAR	intenzitet vježbe (lagano, normalno ili teško)
Trajanje	INTERVAL	trajanje vježbe

Dijeta Entitet Dijeta se sastoji od atributa: ID dijete, ID klijenta te opisa dijete. Dijetu i potrebne parametre unosi doktor. Entitet Dijeta je preko atributa ID u *Many-to-One* vezi s entitetom Korisnik,a preko atributa ID u *One-to-One* vezi s entitetom DnevniLimit,

One-to-Many vezi s OgranicenjaNaKategorije te *One-to-Many* vezi s entitetom OgranicenjaNaProizvode.

Dijeta		
ID	INT	jedinstveni identifikator dijete
IDklijenta	VARCHAR	jedinstveni identifikator klijenta
Opis	VARCHAR	opis dijete

OgraničenjaNaProizvode Dijeta se može definirati s ograničenjima na određene proizvode. Entitet Ograničenja se sastoji od atributa: ID dijete i ID proizvoda. Ovaj entitet je preko atributa IDdijete u *Many-to-One* vezi s entitetom Dijeta te preko atributa IDproizvoda u *Many-to-Many* vezi s entitetom Proizvod.

OgraničenjaNaProizvode		
IDdijete	INT	jedinstveni identifikator dijete
IDproizvoda	INT	jedinstveni identifikator proizvoda

OgraničenjaNaKategorije Dijeta se može definirati s ograničenjima na određene kategorije proizvoda. Entitet Ograničenja se sastoji od atributa: ID dijete i ID kategorije. Ovaj entitet je preko atributa IDdijete u *Many-to-One* vezi s entitetom Dijeta, a preko atributa IDkategorije u *Many-to-Many* vezi s entitetom KategorijeProizvoda.

OgraničenjaNaKategorije		
IDdijete	INT	jedinstveni identifikator dijete
IDkategorije	INT	jedinstveni identifikator kategorije

DnevniLimit Dijeta se može definirati dnevnim limitom za određene nutritivne vrijednosti proizvoda. Entitet DnevniLimit se sastoji od atributa: ID dijete, limit masnoće, limit ugljikohidrata, limit šećera,

limit bjelančevina i limit soli. Ovaj entitet je preko atributa IDdijete u *One-to-One* vezi s entitetom Dijeta.

DnevniLimit		
IDdijete	INT	jedinstveni identifikator dijete
limitEnergije	DOUBLE	limit energije u kJ
LimitMasnoće	DOUBLE	limit masnoća u sastavu proizvoda
LimitZasMasneKiselina	DOUBLE	limit zasićenih masnih kiselina
LimitUgljikohidrata	DOUBLE	limit ugljikohidrata u sastavu proizvoda
LimitŠećera	DOUBLE	limit šećera u sastavu proizvoda
LimitBjelančevina	DOUBLE	limit bjelančevina u sastavu proizvoda
LimitSoli	DOUBLE	limit soli u sastavu proizvoda

UneseneNutrVrijednosti Dijeta se može definirati dnevnim limitom za određene nutritivne vrijednosti proizvoda. Entitet Unesene-NutrVrijednosti prati dnevni unos pojedinih nutritivnih vrijednosti. Atributi su: ID korisnika, datum, te unesena energija, masnoće, zasićene masne kiseline, ugljikohidrati, šećeri, bjelančevine i sol. Ovaj entitet je preko atributa IDklijenta u *Many-to-One* vezi s entitetom Korisnik.

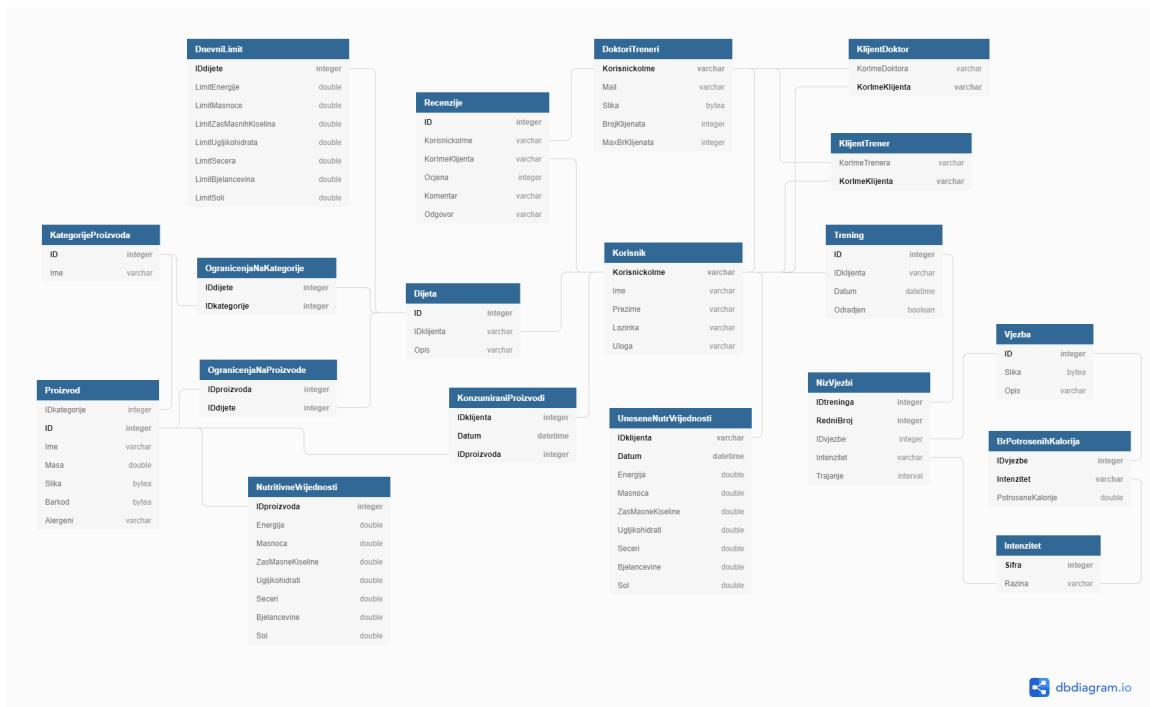
UneseneNutrVrijednosti		
IDklijenta	VARCHAR	jedinstveni identifikator klijenta
Datum	DATETIME	datum
Energija	DOUBLE	unesena energija u kJ
Masnoća	DOUBLE	unesena masnoća

Unesene NutrVrijednosti		
ZasMasneKiseline	DOUBLE	unesene zasićene masne kiseline
Ugljikohidrati	DOUBLE	uneseni ugljikohidrati
Šećeri	DOUBLE	unesena šećeri
Bjelančevine	DOUBLE	unesene bjelančevine
Soli	DOUBLE	unesene soli

KonzumiraniProizvodi Entitet KonzumiraniProizvodi se sastoji od atributa: ID klijenta, datum te ID proizvoda. Ovaj entitet je preko atributa IDproizvoda u *Many-to-Many* vezi s entitetom Proizvod te preko atributa IDklijenta u *Many-to-One* vezi s entitetom Korisnik.

KonzumiraniProizvodi		
IDklijenta	INT	jedinstveni identifikator klijenta
Datum	DATETIME	datum
IDproizvoda	INT	jedinstveni identifikator proizvoda

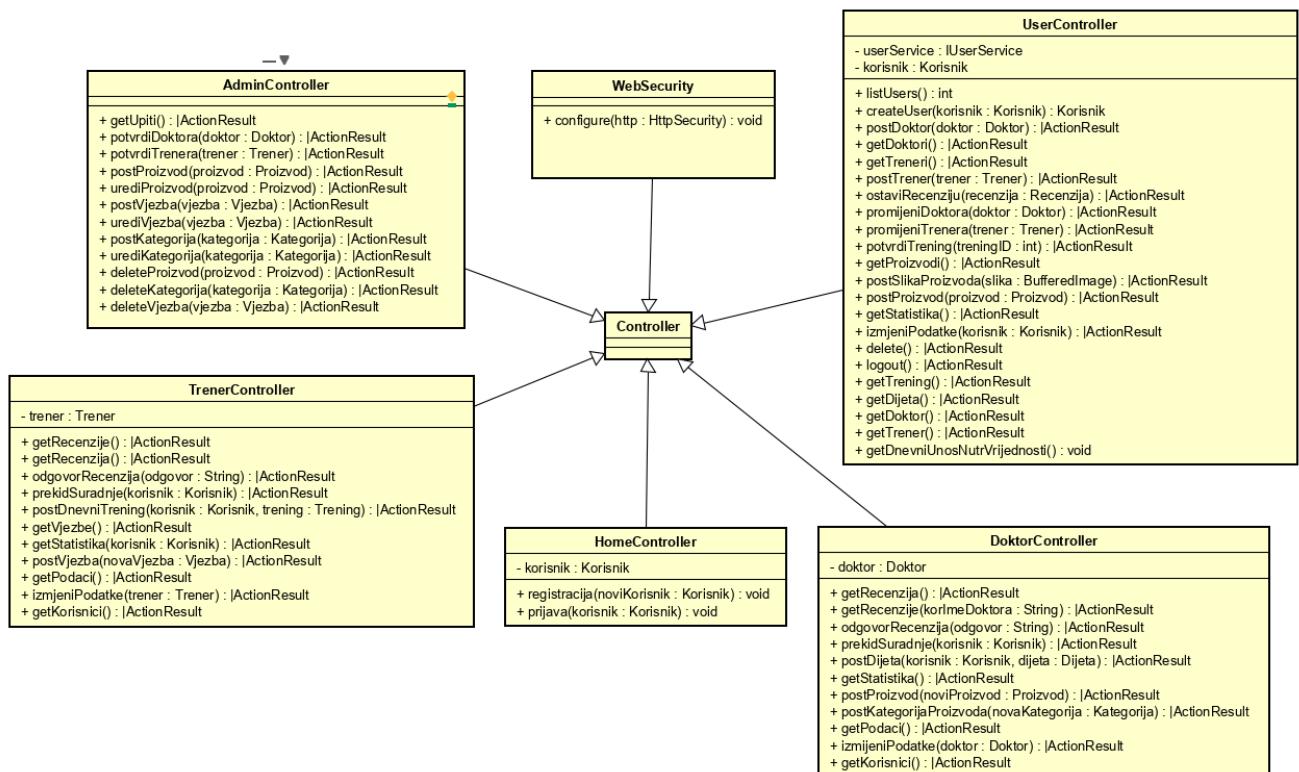
4.1.2 Dijagram baze podataka



Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda

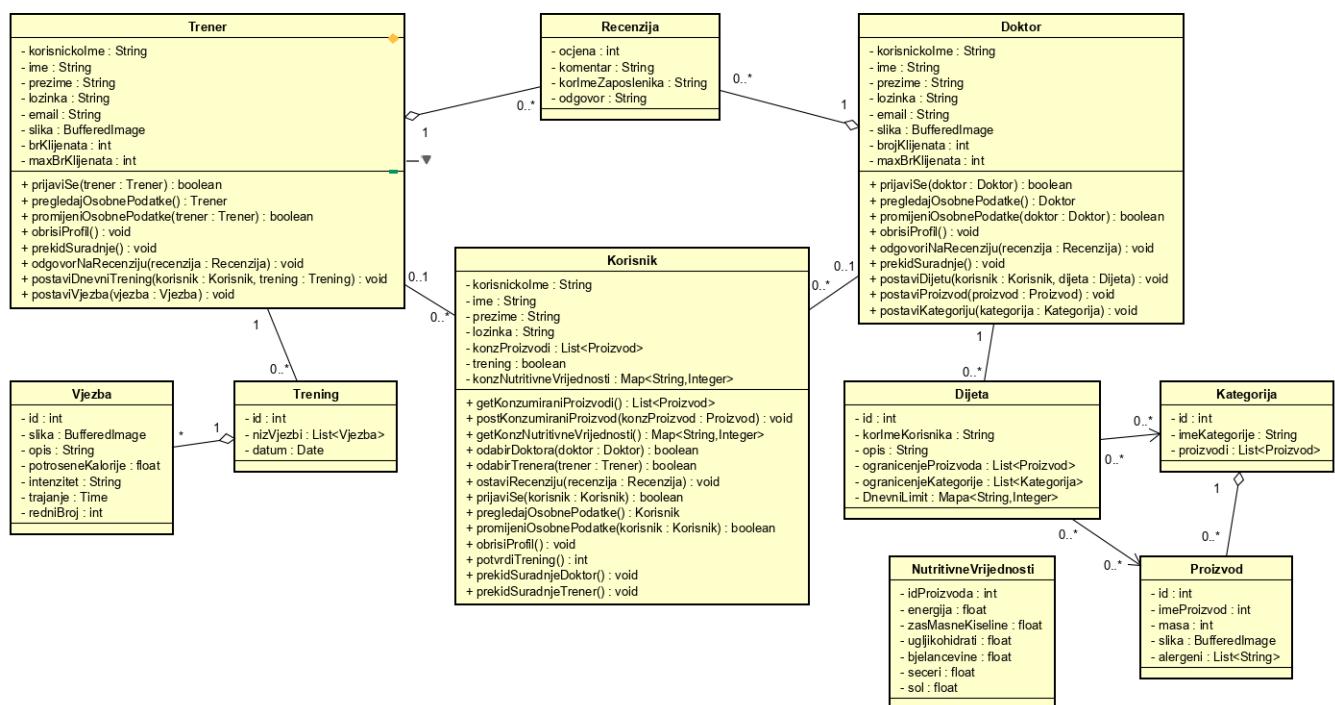
Na slikama 4.1 i 4.2 su prikazani razredi koji pripadaju backend dijelu MVC arhitekture. Razredi prikazani na slici 4.1 nasljeđuju Controller razred. Metode implementirane u Controller razredima vraćaju JSON datoteke s html status kodom.



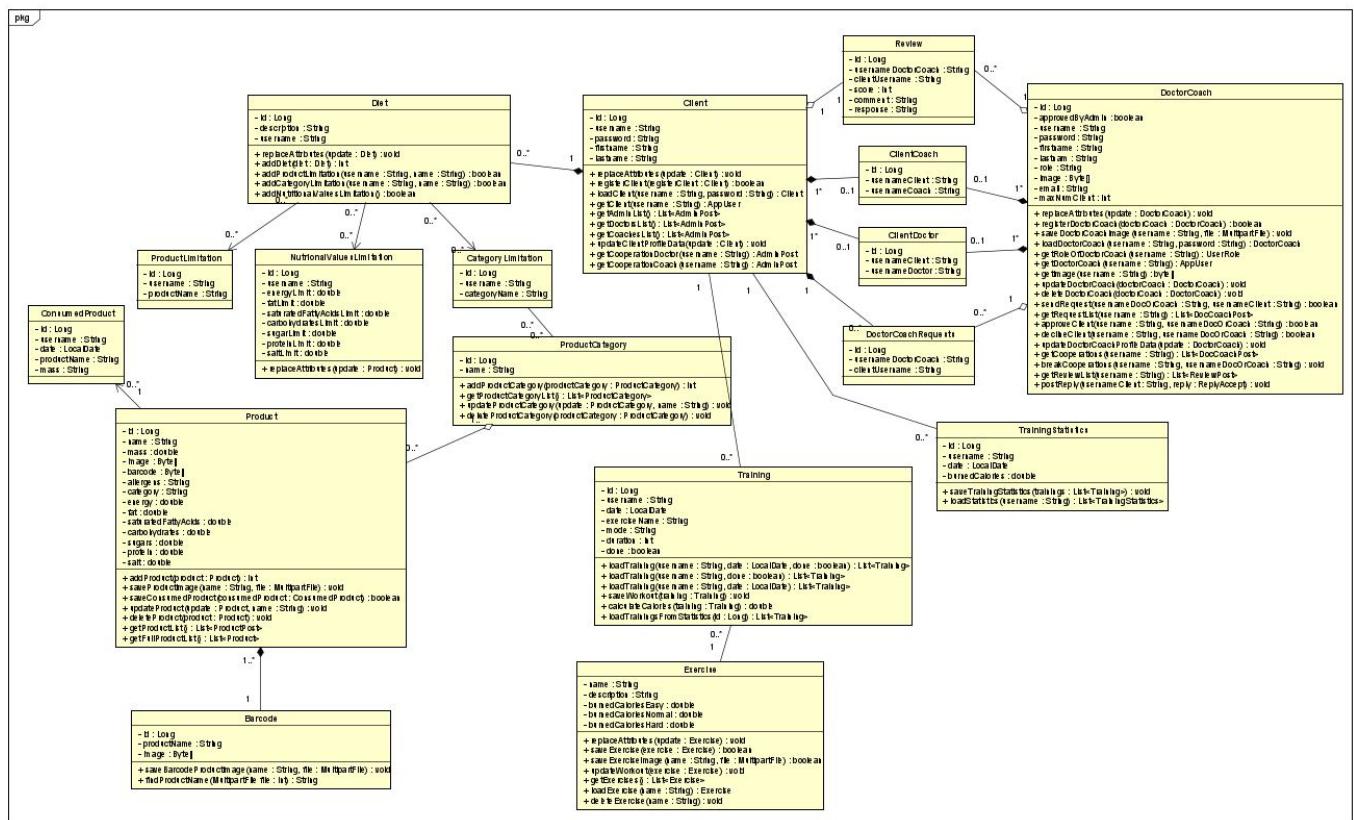
Slika 4.2: Dijagram razreda - dio Controlleri

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vracaju tražene podatke. Razred Korisnik predstavlja registriranu osobu koja može stupiti u suradnju s doktorom i trenerom. Razred Trener predstavlja registrirnu osobu koju je dodatno potvrdio administrator. Njegova glavna uloga je osmišljavanje treninga, koji se sastoje od niza vježbi, svojim klijentima. Vježbe i treninzi su razredi sa svojim atributima. Razred Doktor predstavlja registriranu osobu koju je također dodatno potvrdio administrator. Njegova glavna uloga je osmišljjanje dijete koja se može sastojati od dnevnog limita pojedinih nutritivnih vrijednosti, specifičnih proizvoda ili kategorija proizvoda. Dijeta, Proizvod, Kategorija i Nutritivne Vrijednosti su razredi sa svojim atributima. Razred Recenzija predstavlja osvrt klijenta na svog doktora ili trenera.

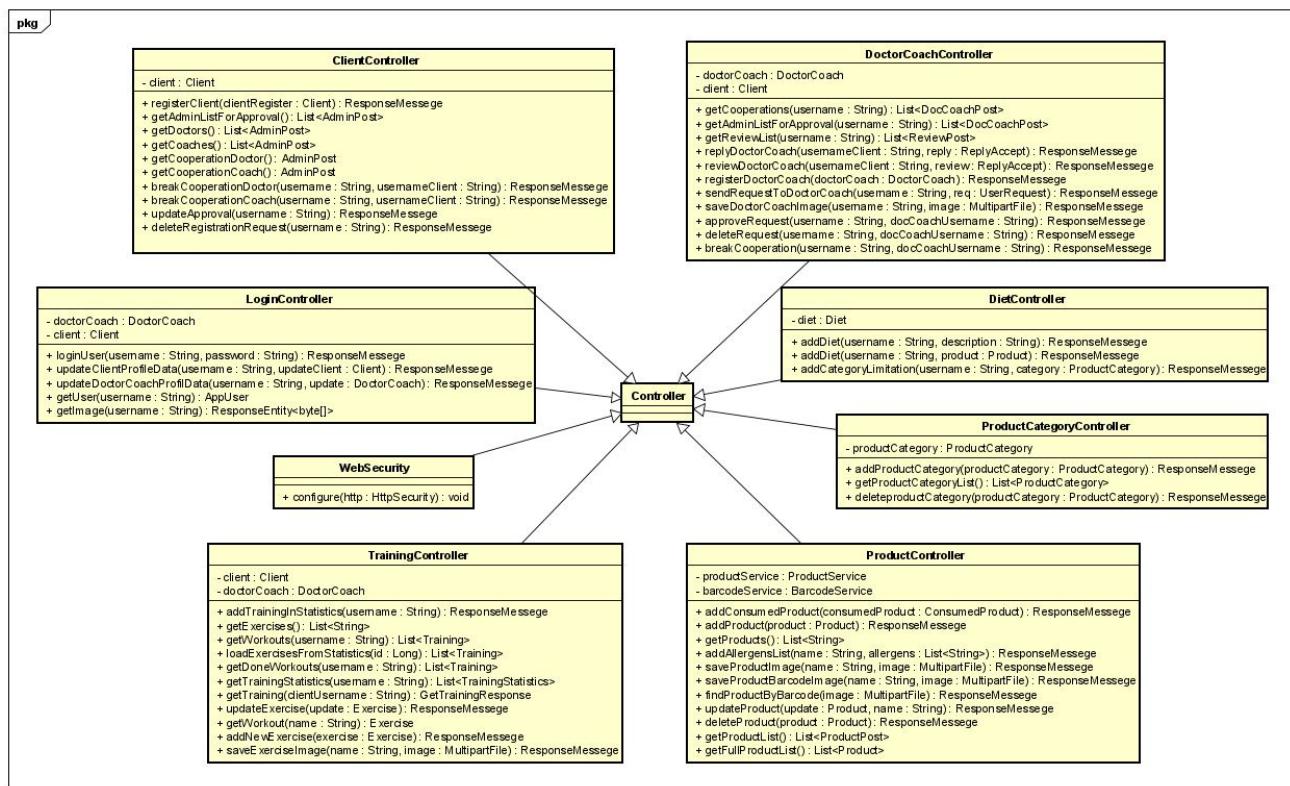
Svi razredi realiziraju get i set metode za svoje atribute.



Slika 4.3: Dijagram razreda - dio Modeli



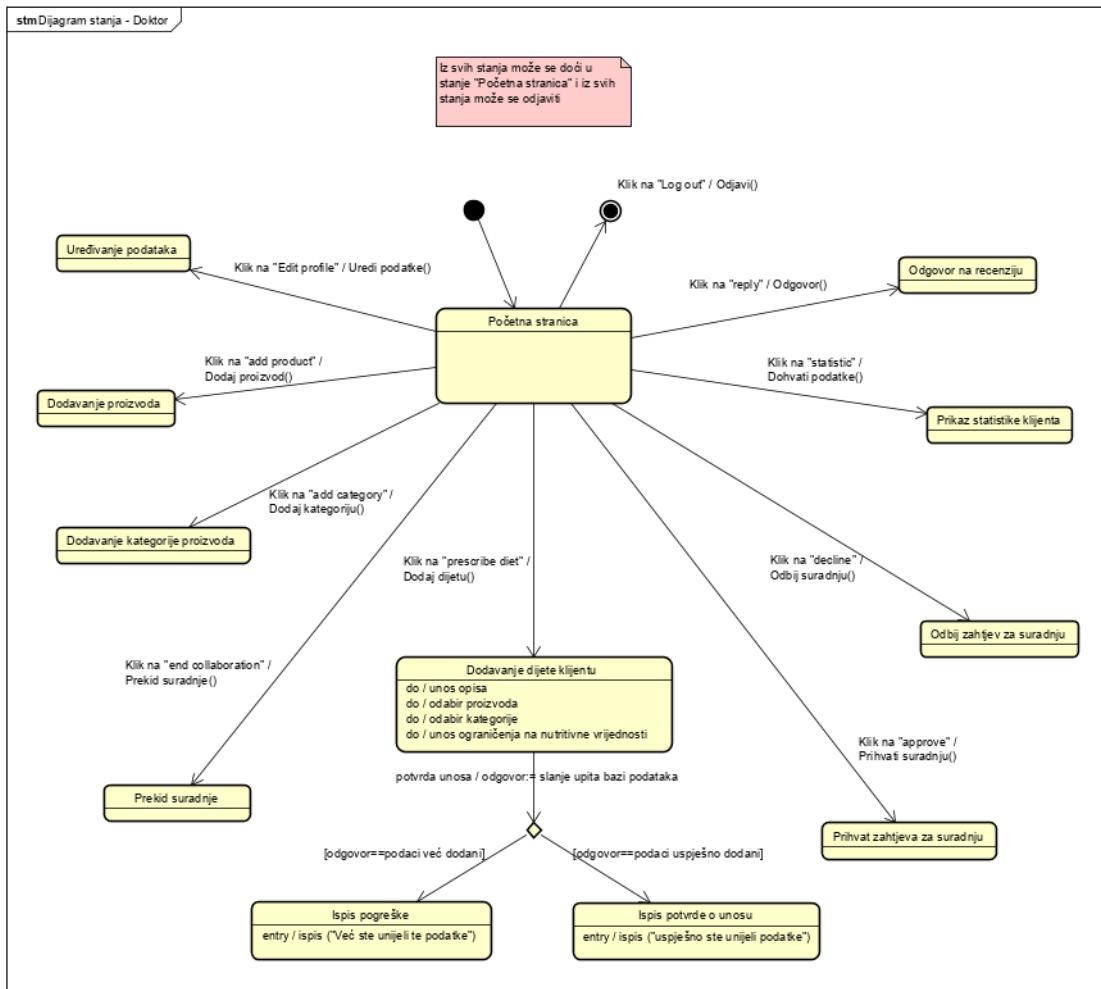
Slika 4.4: Dijagram razreda - dio Model



Slika 4.5: Dijagram razreda - dio Controlleri

4.3 Dijagram stanja

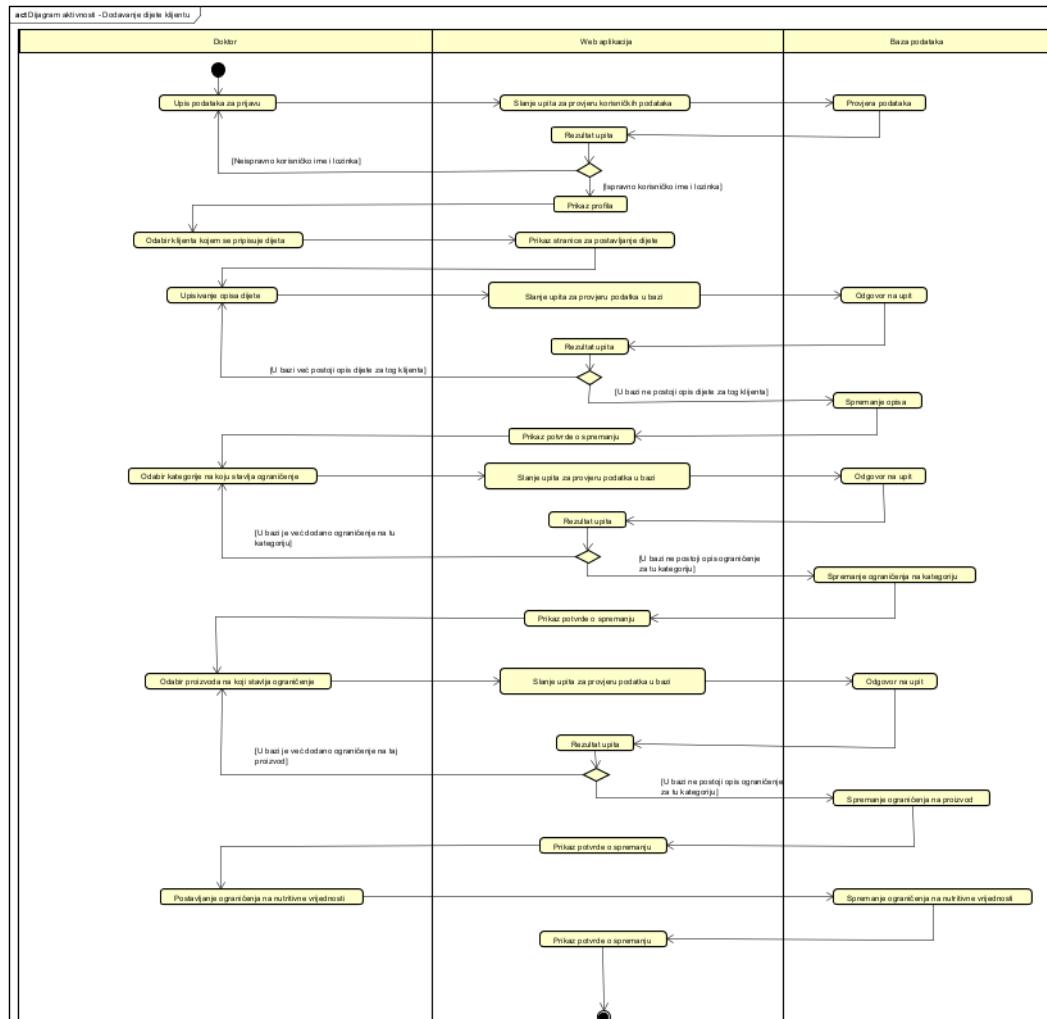
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na dogadajima. Prikazan je dijagram stanja za registriranog doktora. Nakon prijave, doktoru se prikazuje početna stranica. Na toj stanici dokor može prihvati ili odbiti zahtjev za suradnju koju mu je poslao klijent, također ima mogućnost prekida suradnje s klijentima koje je prihvatio prije toga. Svojim klijentima doktor ima mogućnost pripisivanja dijete, nakon klika na pripadajući gumb pojavljuje se mogućnost unosa opisa dijete te ograničenja na proizvode,kategorije i nutritivne vrijednosti. Doktor ima mogućnost dodavanja novih proizvoda i kategorija proizvoda u bazu podataka. Kliko na gumb "Edit Profile" doktor može uređivati vlastiti profil i pohraniti svoje promjene, a klikom na gumb "Reply" može odgoviriti na recenziju koju mu je dao njegov klijent.



Slika 4.6: Dijagram stanja

4.4 Dijagram aktivnosti

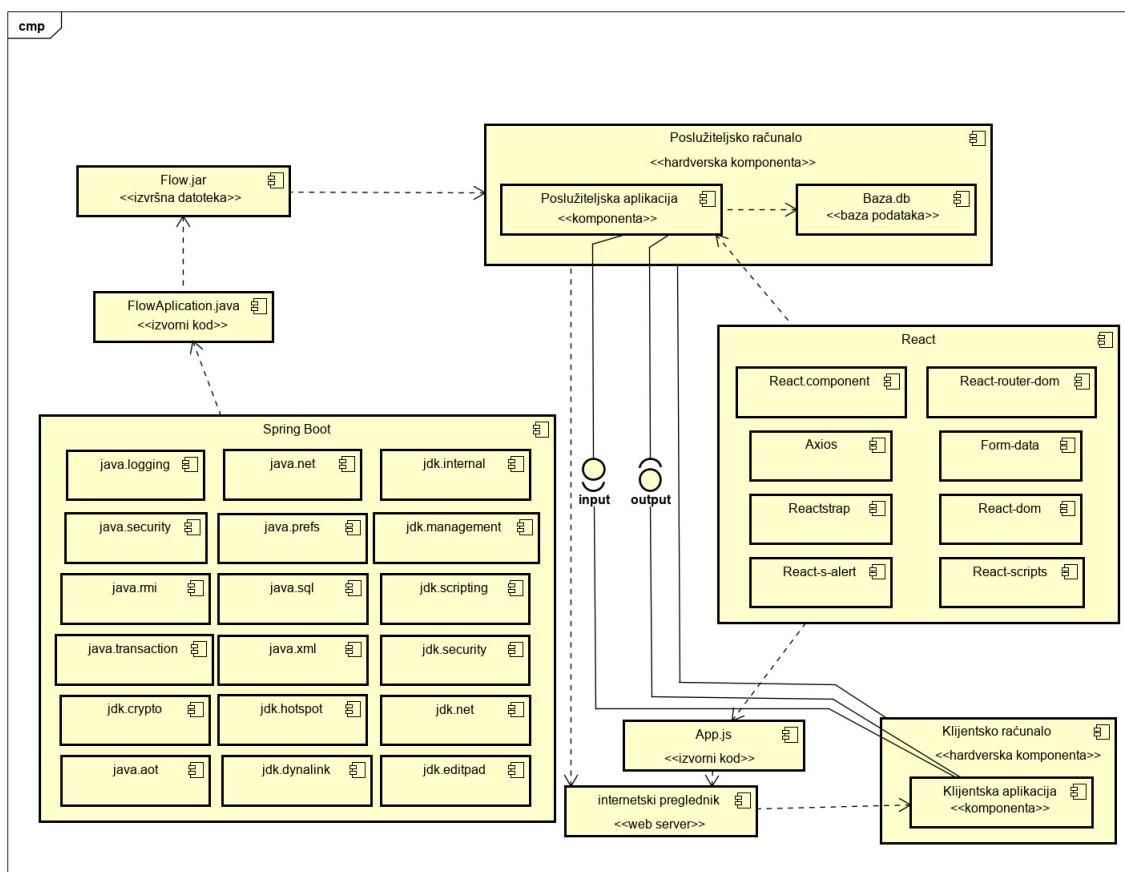
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Na dijagramu aktivnosti prikazan je proces pripisivanja dijete klijentu. Doktor se prijavi u sustav i odabire kojem klijentu želi dodijeliti dijetu. Zatim mu se prikazuje stranica u kojoj ima mogućnost unost unosa opisa dijete i to može učiniti samo jednom. Ispod toga se nalazi padajući izbornik s kategorijama koje doktor može odabrati kao ograničenje u dijeti. Ekvivalentno tome ispod se nalazi mogućnost stavljanja kategorija na proizvode. Može se dodati više ograničenja na proizvode i kategorije te se svaki put prikaže poruka o uspješnosti dodavanja, nije dopušteno dva puta dodati isti proizvod/kategoriju. Naposlijetku ima mogućnost stavljanja ograničenja na nutritivne vrijednosti ručnim upisivanjem i potvrdom s gumbom nakon čega se i to sprema u bazu.



Slika 4.7: Dijagram aktivnosti

4.5 Dijagram komponenti

Na slici je predstavljen okvirni komponentni dijagram sustava. U dijagramu uzimamo da su poslužiteljsko i klijentsko računalo generička te da za njih ne postoje nikakvi posebni zahtjevi. U našem primjeru, klijentsko računalo preko web preglednika komunicira sa poslužiteljskom računalom. U web pregledniku klijentskog računala pokreće se klijentska aplikacija koja je temeljena na razvojnem okružju za javaScript zvanom React pripadajućim knjižnicama, a vrti se pomoću svog izvornog koda App.js. Klijentska aplikacija nudi korisničko sučelje poslužiteljskoj aplikaciji. Na poslužiteljskom računalu s druge strane, pokreće se poslužiteljska aplikacija koja za svoje pokretanje koristi datoteku Flow.jar, koja koristi skupinu knjižnica iz razvojnog okružja Spring Boot. Poslužiteljska aplikacija komunicira sa bazom podataka te preko poslužiteljskog računala šalje podatke na klijentsko računalo.



Slika 4.8: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Pri izradi aplikacije koristili smo bazu podataka PostgreSQL, s kojom aplikacija komunicira preko platforme za ravoj aplikacija java Spring Boot. U njoj su isprogramirane same funkcionalnosti aplikacije. Njih smo tijekom razvoja poloću slanja http zahtjeva provjeravali preko razvojnog alata Postman koji je služio kao sučelje. Samo sučelje aplikacije je pisano u JavaScript biblioteci React. Za samo programiranje smo koristili razvojna okružja Intelij i Eclipse. Za pisanje dokumentacije smo koristili uređivački program TeXstudio, a potrebne dijagramne smo radili u UML alatu za modeliranje zvanom Astah. Za spremanje koda i dokumentacije i praćenje verzija aplikacija smo koristili internetski alat gitLab.

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Komponente su ispitane kroz njihov razvoj.

5.2.2 Ispitivanje sustava

1. Ispitivanje promjene podataka profila

- ulaz : korisničko ime i lozinka korisnika
- očekivani rezultat : promjenjene vrijednosti na profilu
- koraci :

1. ulogiramo se kao korisnik
2. ou kartici gdje su zapisani podaci stisnemo edit profile
3. unesemo nove podatke i stisnemo save
4. aplikacija nas baci na pocetnu stranicu gdje su u kartici promjenjeni

2. Ispitivanje slanje suradnje između klijenta i doktora ili trenera

-ulaz : korisničko ime i lozinka korisnika
-očekivani rezultat : prikaz suranje na profilima klijenta i doktora ili trenera

-koraci :

1. ulogiramo se kao klijent
2. u karticama Doctors ili Coaches izaberemo doktora ili trenera s kojim bi htjeli zapoceti suradnju
3. pritisnemo gumb send request
4. ulogiramo se kao doktor ili trener
5. prihvativmo ili odbijemo suranju pritiskom na gumb accept ili decline

2. Ispitivanje prekida suranje

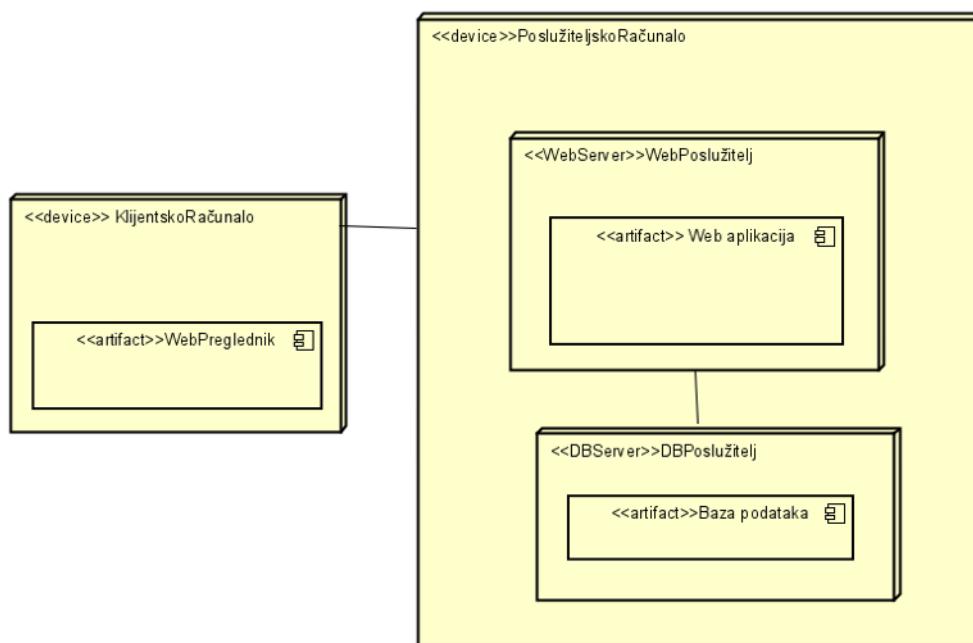
-ulaz : korisničko ime i lozinka korisnik
-očekivani rezultat : prekinuta suradnja

-koraci :

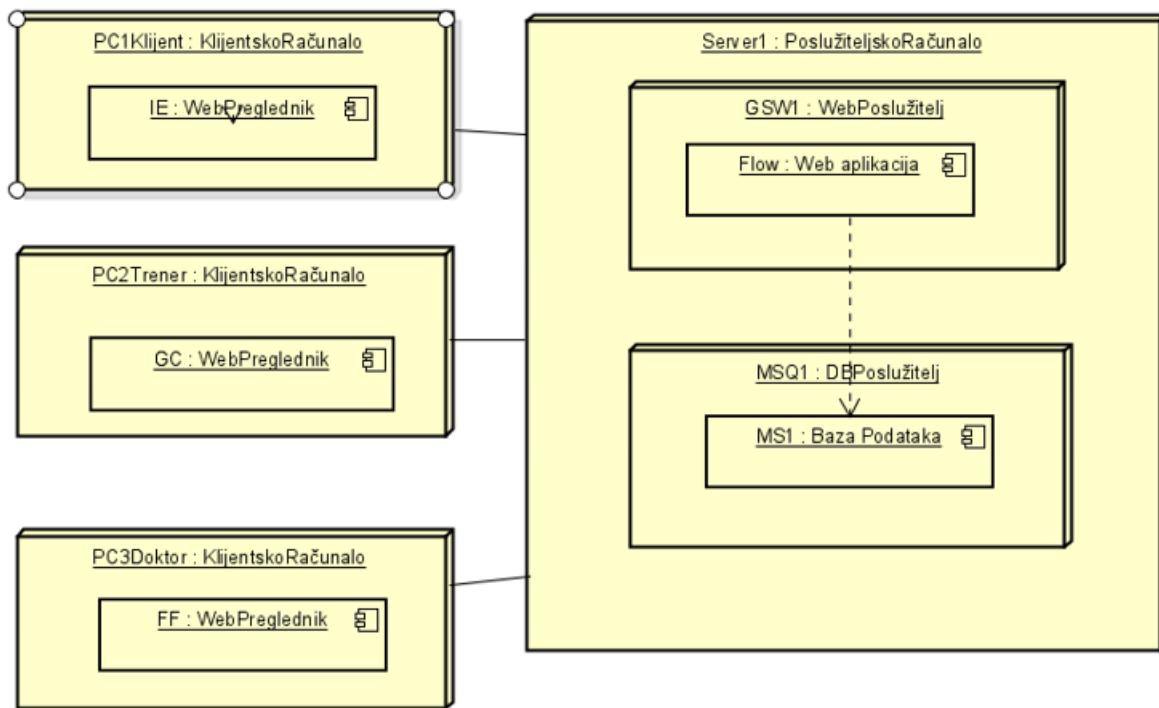
1. ulogiramo se kao klijent
2. u karticama my doctor ili my coach pritisnemo end collaboration

5.3 Dijagram razmještaja

Dijagramima razmještaja prikazana je programska potpora i sklo-povlje korišteno prilikom implementacije sustava. Baza podataka i web poslužitelj nalaze se na poslužiteljskom računalu. Web preglednik, pomoću kojega se pristupa aplikaciji, nalazi se na klijentskom računalu. Komunikacija između korisnika (klijent,doktor,trener,admin) odvija se preko HTTP veze. Specifikacijski dijagram razmještaja pokazuje da je sustav baziran na arhitekturi “klijent-poslužitelj”. Dijagram razmještaja instanci prikazuje sustav tijekom suradnje jednog klijenta,doktora i trenera.



Slika 5.1: Specifikacijski dijagram razmještaja



Slika 5.2: Dijagram razmještaja instanci

5.4 Upute za puštanje u pogon

Ove upute su pisane za operacijski sustav Windows 10, aplikaciju je moguće pokrenuti i na drugim operacijskim sustavima, ali treba se voditi računa da se programi drukčije instaliraju na njima i da su naredbe različite.

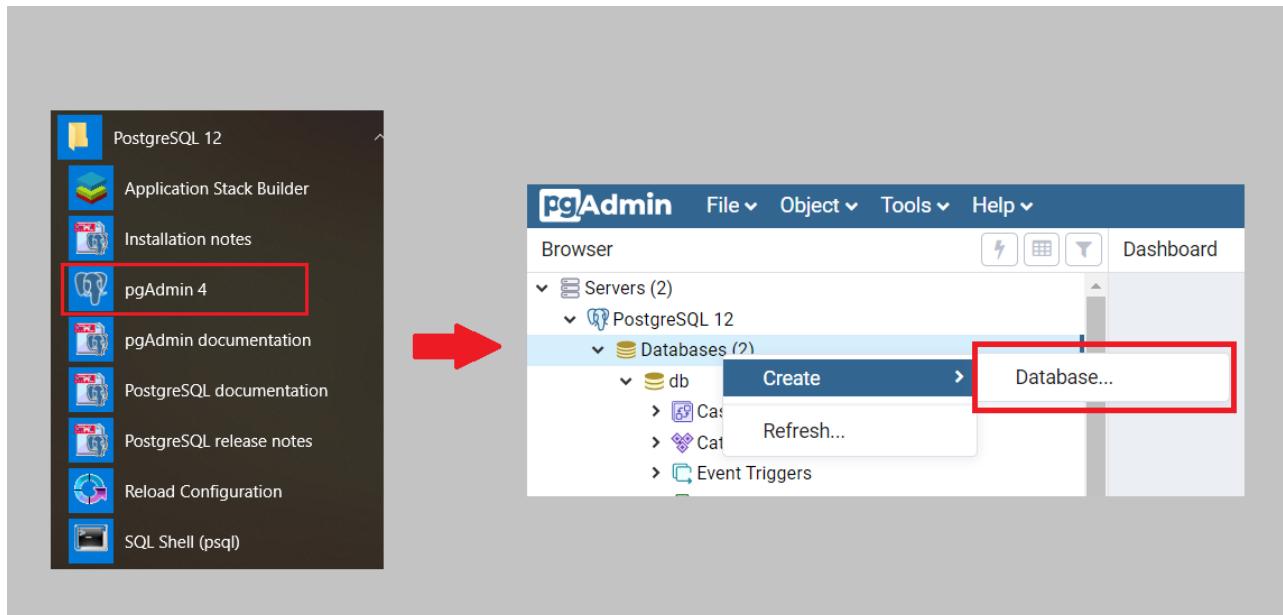
Potrebno je preuzeti installer datoteku s web stranice

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

i provesti standardnu instalaciju PostgreSQL baze podataka na računalo. Bitno je zapamtiti korisničko ime i lozinku koja se postavlja u instalaciji jer će biti potrebne za kasnije spajanje na bazu podataka. Nakon instalacije potrebno je pokrenuti *pgAdmin.exe* koji će otvoriti sučelje za upravljanje bazama podataka.

Zatim treba stvoriti novu bazu podataka tako da se na izborniku

koji se prikaže desnim klikom na 'databases' odabere opcija 'Create->Database...' (ime je proizvoljno).



Slika 5.3: Stvaranje baze podataka

Backend aplikacije pisan je u programskom jeziku Java, stoga računalo mora na sebi imati instaliran Java Development Kit(JDK) verzija 13.

Preuzeti ga je moguće sa službene web stranice:

<https://www.oracle.com/technetwork/java/javase/downloads/index.html>

Provesti standardnu instalaciju i provjeriti je li sve ispravno instalirano tako da se otvori Command Prompt i upiše 'java --version', ako je sve uredu ispisati će se verzija instalirane java.

A screenshot of a Windows Command Prompt window titled 'cmd C:\WINDOWS\System32\cmd.exe'. The command 'java --version' is entered, and the output shows the Java version: 'java 13.0.1 2019-10-15 Java(TM) SE Runtime Environment (build 13.0.1+9) Java HotSpot(TM) 64-Bit Server VM (build 13.0.1+9, mixed mode, sharing)'.

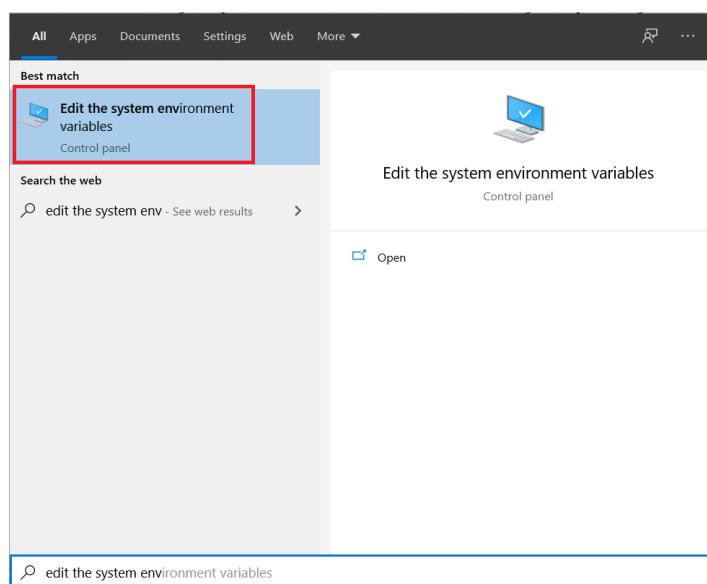
Slika 5.4: Provjera je li JDK ispravno instaliran

U pisanju backenda korišten je Spring framework te zato računalo mora imati instaliran Maven koji vodi računa o svim zavisnostima

(dependencies) koje projekt ima. Potrebno je preuzeti zip arhivu s web stranice:

<https://maven.apache.org/download.cgi>

te ju raspakirati u željeni direktorij. Potom, podesiti environment variable operacijskog sustava tako da otvorite '*System Properties*' (to je moguće na par načina, ali najlakše je tako da se u polje za pretraživanje upiše: '*edit the system en...*') i odaberete '*Edit the system environment variables*'.



Slika 5.5: Otvaranje System Properties

U prozoru koji se otvorio odaberite '*Environment Variables*', nakon toga odabratи opciju '*New*' u dijelu sučelja '*System Variables*', u polje '*Variable name*' upisati '*MAVEN_HOME*', a u polje '*Variable value*' upisati putanju direktorija koji je nastao raspakiravanjem zip arhive. Potom je potrebno odabratи opciju '*Edit*' za varijablu '*Path*' u polju '*System Variables*' i kada se otvori novi prozor odabratи opciju '*New*' te upisati '*%MAVEN_HOME%\bin*'. Maven bi sada trebao biti uspješno instaliran, provjeriti tako da se u Command Prompt upiše naredba '*mvn --version*'.



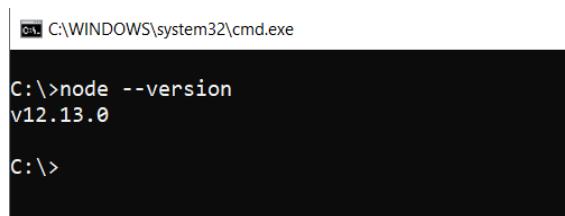
```
C:\>mvn --version
C:\
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: C:\Programs\apache-maven-3.6.3\bin\..
Java version: 13.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-13.0.1
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
C:\>
```

Slika 5.6: Provjera je li Maven ispravno instaliran

Frontend aplikacije napisan je koristeći React Javascript library, stoga računalo mora imati instaliran Node.js kako bi ga mogli pokrenuti. S web stranice:

<https://nodejs.org/en/>

preuzeti i pokrenuti installer datoteku. Provesti standardnu instalaciju. Provjeriti je li instalacija uspješna tako da se u Command Prompt upiše naredba '`node --version`'.



```
C:\>node --version
v12.13.0
C:\>
```

Slika 5.7: Provjera je li Node.js ispravno instaliran

Sada kada su svi potrebni programi instalirani možemo početi s puštanjem aplikacije u pogon. Preuzeti direktorij s GitLab stranice:

<https://gitlab.com/fermark/flow>

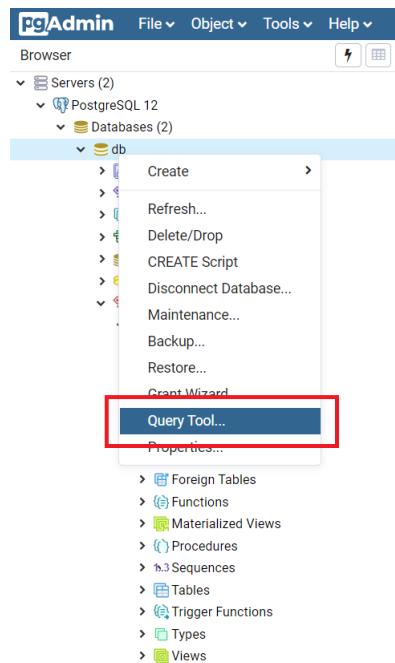
tako da se odabere opcija `Download->zip`. Raspakirati zip arhiv te navigirati u direktorij `flow\src\main\resources` te otvoriti datoteku `application.properties`. Potrebno je podesiti nekoliko parametara kako bi se moglo ispravno spojiti na bazu podataka. Za vrijednost parametra `<ime baze podataka>` u polju `spring.datasource.url` potrebno je upisati ime vaše baze podataka. U polja `username` i `password` upisati korisničko ime i lozinku koju ste odabrali kada ste instalirali PostgreSQL bazu podataka. Spremiti izmijene u datoteci.

```
spring.datasource.url=jdbc:postgresql://localhost:5432/<ime baze podataka>
spring.datasource.username=
spring.datasource.password=
spring.datasource.driver-class-name=org.postgresql.Driver
```

Slika 5.8: Izmjena datoteke 'application.properties'

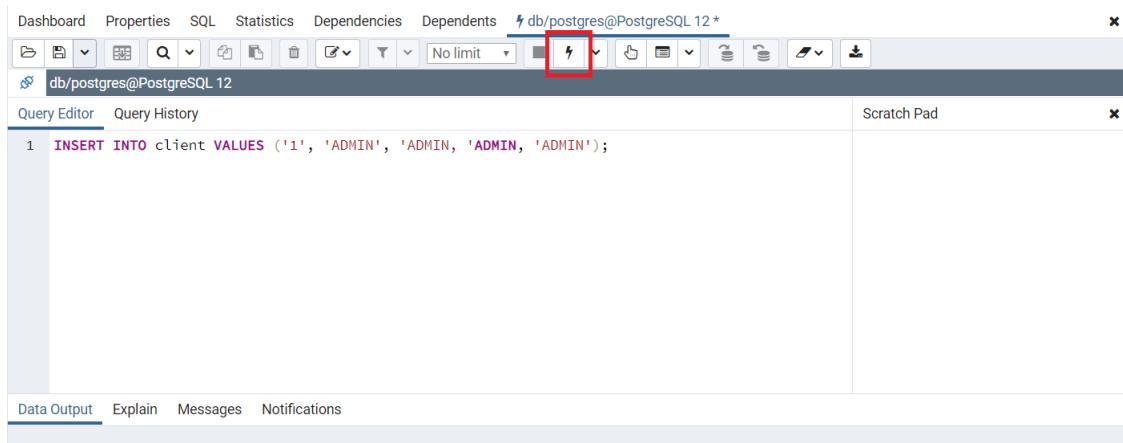
Otvoriti Command Prompt i navigirati u direktorij `\f1ow` te tamo izvršiti naredbu `'mvn install'`. Time se stvorila izvršna datoteka tipa 'jar' u poddirektoriju `target`. Sada je potrebno postaviti i frontend dio aplikacije. U Command Promptu navigirati u direktorij `\f1ow_front` te tamo izvršiti naredbu `'npm install'` koja će preuzeti biblioteku React. Potom u istome direktoriju je potrebno izvršiti naredbe `'npm install react-router-dom --save'` i `'npm install axios --save'`, a te dvije naredbe će preuzeti potrebne zavisnosti (dependencies) projekta.

Sada još jedino preostaje pokrenuti aplikaciju. U direktoriju `\f1ow_front` izvršiti naredbu `'npm start'`, frontend dio aplikacije će se pokrenuti. Zatim je potrebno otvoriti novu instancu Command Prompta i u direktoriju `\f1ow\target` izvršiti naredbu `'java -jar f1ow-0.0.1-SNAPSHOT.jar'`. Ta će naredba pokrenuti backend dio aplikacije koji automatski stvara sve potrebne tablice u bazi podataka, jedino što je još potrebno napraviti je ubaciti u bazu podataka podatke za prijavu administratora. Ponovo pokrenuti sučelje za upravljanje bazom podataka `pgAdmin` i pritisnuti desni klik na vašoj bazi podataka te odabrati opciju `Query Tool`....



Slika 5.9: Otvaranje Query Toola u pgAdmin sučelju

U polje Query Editor upisati naredbu `'INSERT INTO client VALUES ('1', 'ADMIN', 'ADMIN', 'ADMIN', 'ADMIN');'` i izvršiti ju pritiskom na gumb s ikonom munje.



Slika 5.10: Izvršavanje naredbe u sučelju pgAdmin

Aplikacija je pokrenuta i moguće joj je pristupiti na URL

<http://localhost:3000/>

6. Zaključak i budući rad

Nakon što je aplikacija razvijena i postala javno dostupna stavljanjem na server, slijedi razdoblje u kojem bi se trebalo raditi na kontinuiranom poboljšavanju i optimiziranju performansi same aplikacije. To uključuje samo održavanje zdravljia izvorne aplikacije i njezinih funkcionalnosti, kao i proširivanje same aplikacije prema zahtjevima i potrebama njihovih korisnika. Rad na projektu aplikacije ne završava u trenutku njena puštenja u javnost, jer se tek tada treba krenuti u njenu prilagodbu zahtjevima korisnika

Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <https://www.fer.unizg.hr/predmet/opp>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. The Unified Modeling Language, <https://www.uml-diagrams.org/>
4. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Početna stranica	9
2.2	Login	9
2.3	Testiranja	10
2.4	Testiranja	10
3.1	Generalni dijagram obrasca uporabe	27
3.2	Dijagram obrasca uporabe - registracija i prijava . . .	28
3.3	Dijagram obrasca uporabe - opcije suradnje	28
3.4	Dijagram obrasca uporabe - načini suradnje	29
3.5	Dijagram obrasca uporabe - operacije nad kategorijama	29
3.6	Dijagram obrasca uporabe - operacije nad vježbama i proizvoidma	30
3.7	Sekvencijski dijagram za UC1	31
3.8	Sekvencijski dijagram za UC5 i UC6	32
3.9	Sekvencijski dijagram za UC8	33
3.10	Sekvencijski dijagram za UC10	34
3.11	Sekvencijski dijagram za UC25	36
4.1	Dijagram baze podataka	50
4.2	Dijagram razreda - dio Controlleri	51
4.3	Dijagram razreda - dio Modeli	52
4.4	Dijagram razreda - dio Model	53
4.5	Dijagram razreda - dio Controlleri	54
4.6	Dijagram stanja	56
4.7	Dijagram aktivnosti	58

4.8 Dijagram komponenti	60
5.1 Specifikacijski dijagram razmještaja	63
5.2 Dijagram razmještaja instanci	64
5.3 Stvaranje baze podataka	65
5.4 Provjera je li JDK ispravno instaliran	65
5.5 Otvaranje System Properties	66
5.6 Provjera je li Maven ispravno instaliran	67
5.7 Provjera je li Node.js ispravno instaliran	67
5.8 Izmjena datoteke 'application.properties'	68
5.9 Otvaranje Query Toola u pgAdmin sučelju	69
5.10 Izvršavanje naredbe u sučelju pgAdmin	69
6.1 Dijagrami aktivnosti - 1. dio	79
6.2 Dijagram aktivnosti - 2. dio	79

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 05. listopada 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - upoznavanje tima
 - okvirna podjela posla (backend, frontend, baza podataka)
 - komentiranje teme dobivenog zadatka

2. sastanak

- Datum: 10. listopada 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - okvirna razrada baze podataka
 - dogovor o korištenju alata i tehnologija kojima će se ostvariti aplikacija
 - dogovor o upoznavanju alata i tehnologija koje se koriste za ostvarivanje aplikacije

3. sastanak

- Datum: 17. listopada 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - dijeljenje stečenog znanja o pojedinim alatima i tehnolo-

gijama

- dogovor da se baza podataka radi u PostgreSQLu i backend u Java Springu

4. sastanak

- Datum: 22. listopada 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - konkretna razrada baze podataka
 - podjela pisanja dokumentacije

5. sastanak

- Datum: 28. listopada 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - implementacija prijave
 - isprobavanje funkcionalnosti backenda pomoću alata Postman

6. sastanak

- Datum: 07. studeni 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - spajanje backenda s bazom podataka

7. sastanak

- Datum: 11. studeni 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - isprobavanje generičke funkcionalnosti
 - spajanje s frontendom
- Datum: 8. siječanj 2019.

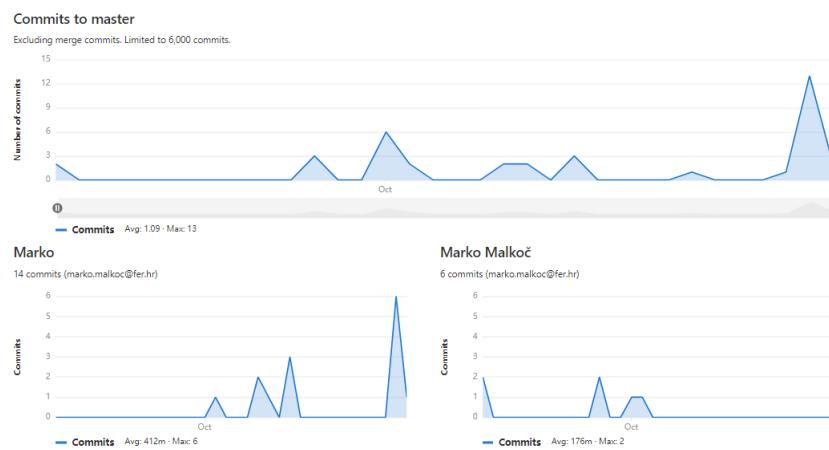
- Prisustvovali: cijeli tim
- Teme sastanka:
 - dogovor oko alfa inačice
- Datum: 13. siječanj 2019.
- Prisustvovali: cijeli tim
- Teme sastanka:
 - dogovor oko završetka aplikacije

Tablica aktivnosti

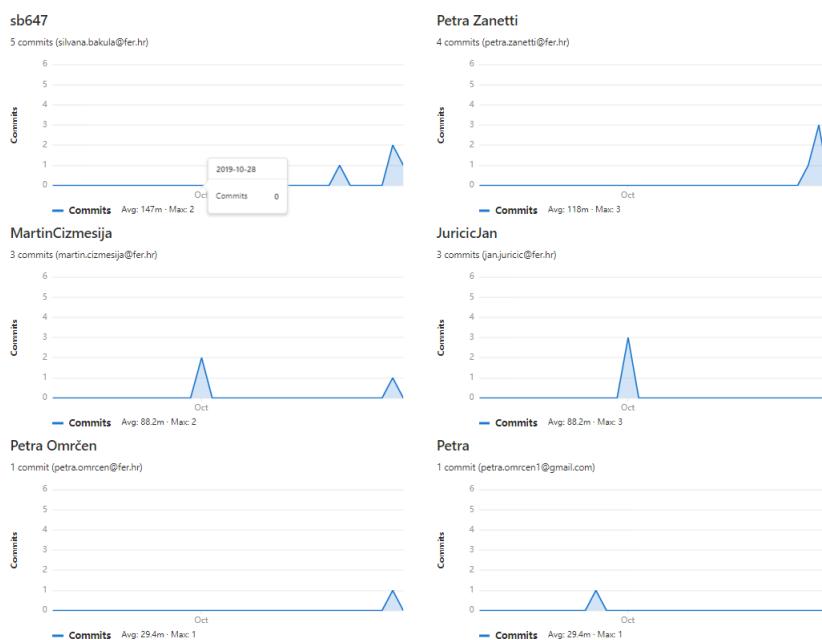
	Marko Malkoč	Petra Zanetti	Ana Marija Ereš	Jan Juričić	Petra Omrčen	Martin Čižmešija	Silvana Bakula
Upravljanje projektom	x						
Opis projektnog zadatka					x		
Funkcionalni zahtjevi						x	
Opis pojedinih obrazaca	x						
Dijagram obrazaca						x	
Sekvencijski dijagrami	x	x					
Opis ostalih zahtjeva				x			
Arhitektura i dizajn sustava				x			
Baza podataka							x
Dijagram razreda	x	x					
Dijagram stanja	x				x		
Dijagram aktivnosti			x				
Dijagram komponenti						x	
Korištene tehnologije i alati						x	
Ispitivanje programskog rješenja	x	x	x	x	x	x	x
Dijagram razmještaja		x					
Upute za puštanje u pogon	x						

	Marko Malkoč	Petra Zanetti	Ana Marija Ereš	Jan Jurčić	Petra Omrčen	Martin Čižmešija	Silvana Bakula
Dnevnik sastajanja		x					
Zaključak i budući rad					x		
Popis literature	x						
<i>front end</i>	x					x	
<i>izrada baze podataka</i>		x			x		x
<i>spajanje s bazom podataka</i>	x	x	x	x	x	x	x
<i>back end</i>			x	x			

Dijagrami pregleda promjena



Slika 6.1: Dijagrami aktivnosti - 1. dio



Slika 6.2: Dijagram aktivnosti - 2. dio