

# Programsko inženjerstvo

Ak. god. 2022./2023.

## *Group Fitness Planner*

Dokumentacija, Rev. <2>

Grupa:

*DevUps*

Voditelj:

*Tomislav Kožul*

Datum predaje: 13. 1. 2023.

Nastavnik: *Ivana Lulić*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
2.1	Pregled konkurentskih rješenja i analiza tržišta .....	9
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>15</b>
3.1	Funkcionalni zahtjevi .....	15
3.1.1	Obrasci uporabe .....	17
3.1.2	Dijagrami obrazaca uporabe .....	26
3.1.3	Sekvencijski dijagrami .....	28
3.2	Ostali zahtjevi .....	33
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>34</b>
4.1	Baza podataka .....	36
4.1.1	Opis tablica .....	37
4.1.2	Dijagram baze podataka .....	41
4.2	Dijagram razreda .....	42
4.3	Dijagram stanja .....	46
4.4	Dijagram aktivnosti .....	47
4.5	Dijagram komponenti .....	49
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>51</b>
5.1	Korištene tehnologije i alati .....	51
5.1.1	Frontend tehnologije .....	51
5.1.2	Backend tehnologije .....	51
5.1.3	Deployment .....	51
5.1.4	Dokumentacija .....	51
5.1.5	Timska komunikacija .....	51
5.2	Ispitivanje programskog rješenja .....	52
5.2.1	Ispitivanje komponenti .....	52

5.2.2 Ispitivanje sustava . . . . .	54
5.3 Dijagram razmještaja . . . . .	58
5.4 Upute za puštanje u pogon . . . . .	60
5.4.1 Upute za puštanje u pogon na javnom poslužitelju . . . . .	60
5.4.2 Upute za puštanje u pogon lokalno . . . . .	61
<b>6 Zaključak i budući rad</b>	<b>63</b>
<b>Popis literature</b>	<b>65</b>
<b>Indeks slika i dijagrama</b>	<b>67</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>68</b>

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Opis projektnog zadatka	Tomislav Kožul	5.11.2022.
0.2	Dodavanje funkcionalnih i ne-funkcionalnih zahtjeva	Nika Šljubura	5.11.2022.
0.3	Dodane tablice i opisi baze podataka	Damir Numić-Meša, Bruna Kaštela	3.11.2022.
0.3.1	Izmjena tablica baze podataka. Dopisan dio ostalih zahtjeva.	Damir Numić-Meša	7.11.2022.
0.4	Opisi obrazaca uporabe	Bruna Kaštela, Damir Numić-Meša	10.11.2022.
0.5	Vođenje i dodavanje dnevnika	Rujana Perić	10.11.2022.
0.6	Arhitektura i dizajn sustava. Dijagrami razreda.	Damir Numić-Meša	14.11.2022.
0.6.1	Popravljeni dijagrami razreda. Dijagram baze podataka.	Damir Numić-Meša	15.11.2022.
0.7	Specifikacija programske potpore	Nika Šljubura	5.11.2022
0.8	Izmjena opisa projektnog zadatka	Tomislav Kožul	16.11.2022
0.9	Dodani sekvencijski dijagrami	Rujana Perić	18.11.2022
0.9	Dodani dijagrami obrasca uporabe	Petra Renić	18.11.2022
1.0	<b>Verzija samo s bitnim dijelovima za 1. ciklus</b>	Rujana Perić, Bruna Kaštela	18.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.1	Dodani dijagrami aktivnosti i komponenti	Rujana Perić	6.1.2023.
1.2	Ispravci dijagrama razreda	Damir Numić-Meša	5.1.2023.
1.3	Dodan dijagram razmještaja, opisivanje implementacija i korištenih alata	Damir Numić-Meša	6.1.2023.
1.4	Dodani testovi komponenti i odgovarajući opisi	Damir Numić-Meša	9.1.2023.
1.5	Dodan dijagram stanja i opis dijagrama	Bruna Kaštela	10.1.2023.
1.6	Dodan pregled konkurentskih rješenja i analiza tržišta	Rujana Perić	10.1.2023.
1.7	Dodani Selenium testovi sustava	Damir Numić-Meša	11.1.2023.
1.8	Dodane upute za puštanje aplikacije u pogon	Rujana Perić	11.1.2023.
1.9	Dodan zaključak	Rujana Perić	12.1.2023.
1.10	Popravljeni obrasci uporabe i dijagrami obrazaca uporabe	Rujana Perić, Petra Renić	13.1.2023.
2.0	<b>Konačna verzija dokumentacije</b>	Rujana Perić	13.1.2023.

## 2. Opis projektnog zadatka

Cilj ovog projektnog zadatka je napraviti web aplikaciju koja polaznicima treninga omogućuje rezervaciju odgovarajućeg termina treninga i postavljanje osobnih ciljeva koje žele ostvariti u nadolazećem mjesecu, a terenima daje mogućnost planiranja i treninga ovisno prema ciljevima korisnika i daje im mogućnosti postavljanja ograničenja za treninge.

Potreba za razvojem ovakve aplikacije iz klijentske perspektive javlja se zbog velikog broja obaveza i nemogućnosti praćenja fiksnog rasporeda treninga. Kako korisnici ne mogu pohađati svaki trening, javlja se nezadovoljstvo iz dva razloga:

- nemogućnost pohađanja usluge koju su platili
- usporen napredak zbog neredovitog treniranja

Kako treneri ne bi gubili klijente, nudimo im mogućnost pohađanja termina u vrijeme kada klijentima odgovara.

U aplikaciji, postoje dvije role koje korisnik može odabrati:

- klijent (u daljnjem tekstu vježbač)
- trener

Pri pokretanju aplikacije, ukoliko korisnik već nema postojeći račun, mora izvršiti registraciju. Pri registraciji, korisnik mora unijeti određene podatke:

- ime
- prezime
- datum rođenja
- email adresa
- korisničko ime
- kontakt broj
- lozinka
- tip korisnika (vježbač ili trener)

- ako je tip vježbač, potrebno je unijeti i cilj

Lozinka koju korisnik unese mora imati 8 – 20 znakova.

Pri registraciji, vježbač odabire vlastiti cilj na temelju kojeg mu trener određuje treninge. Postoje tri različita cilja koja korisnik može odabrati:

- povećanje mišićne mase
- izdržljivost
- gubitak težine

Kada se vježbač registrira, dobiva potvrdni mail za registraciju koji ga odvodi na stranicu za prijavu. Po dodjeli treninga vježbaču se unutar aplikacije prikazuju njemu dodijeljeni treninzi (i termini istih).

Kada netko od trenera vježbaču dodijeli fond sati koji vježbač može iskoristiti i kada vježbač dobije ponuđene treninge, vježbač dobiva obavijest na mail te može na kalendaru odabrati termine u kojima bi htio odraditi treninge. Treninzi se odrađuju u nekoliko termina svaki radni dan:

- 8:00 sati
- 10:00 sati
- 12:00 sati
- 14:00 sati
- 16:00 sati
- 18:00 sati
- 20:00 sati

Ako vježbač ipak ne može pohađati trening za koji se prijavio, moguće je poništiti rezervaciju bez gubitka fonda sati.

Vježbač može nakon nekog vremena, ili na samom početku, izgubiti volju ili motivaciju za određenom vrstom treninga i zbog toga mu moramo omogućiti određenu fleksibilnost. Zadnji tjedan u mjesecu, vježbaču ćemo omogućiti promjenu cilja.

Kada vježbač rezervira termin, njegov fond sati se umanjuje za duljinu trajanja treninga koja je u našem slučaju 2h. S druge strane, otkazivanjem rezervacije, njegov fond sati se uvećava za duljinu treninga, tako da rezervacija te otkaz iste

nemaju utjecaj na korisnikov fond sati.

Ako vježbač odluči ne doći na trening, ukoliko nije poništio rezervaciju, rezervirani trening se računa kao odrađeni, iako ga vježbač nije pohađao. Time vježbači gube mogućnost naknadnog odrađivanja rezerviranog treninga.

Radi ograničenog broja opreme, treneri moraju definirati i maksimalan broj polaznika za određenu vrstu treninga. Kada su sva slobodna mjesta za određeni trening popunjena, termin nestaje iz aplikacije i korisnici ga više ne mogu vidjeti.

Treneri trebaju napraviti raspored treninga prema određenim ciljevima vježbača kako bi vježbači mogli što prije ispuniti svoje ciljeve i biti zadovoljni pruženom uslugom.

Kao primjer, uzmimo vježbača koji želi povećati izdržljivost. Trener u skladu s ciljem vježbača treba uzeti u obzir fond sati koji bi odgovarali za postizanje navedenog cilja te na temelju oba faktora odrediti koliko različitih vrsta treninga vježbač treba odraditi.

Iako više treninga može biti "tipa" trening snage (što će često biti vidljivo u nazivu treninga), to ne znači da svi treninzi istog ili sličnog naziva nužno sadrže iste vježbe. Treneru je dano na vlastitu prosudbu da pri izradi treninga definira koje vježbe isti obuhvaća.

Za vježbače postoji ograničenje na broj mogućih rezervacija termina treninga tjedno. Ograničenje je skalirano prema fondu sati koje je vježbaču dodijeli trener, tj. vježbači sa velikim fondom sati imaju mogućnost većeg broja rezervacija tjedno u odnosu na manji fond sati.

Ako vježbač pokuša prekršiti ograničenje broja mogućih rezervacija, aplikacija ga o tome obavijesti porukom i ne dopušta mu izvršavanje rezervacije koja krši zadano pravilo.

Uz unaprijed definirano pravilo koje provjerava aplikacija, treneri mogu pri izradi treninga napisati vlastita pravila (u sklopu opisa treninga).



Administrator je uloga koja ima mogućnosti upravljanja korisničkim računima (brisanje računa) te verifikaciju trenera.

Pri registraciji, korisnik odabire ulogu trenera ili vježbača. Kako se bilo tko ne bi mogao neovlašteno registrirati kao trener, administrator mora potvrditi (verificirati) registraciju trenera.

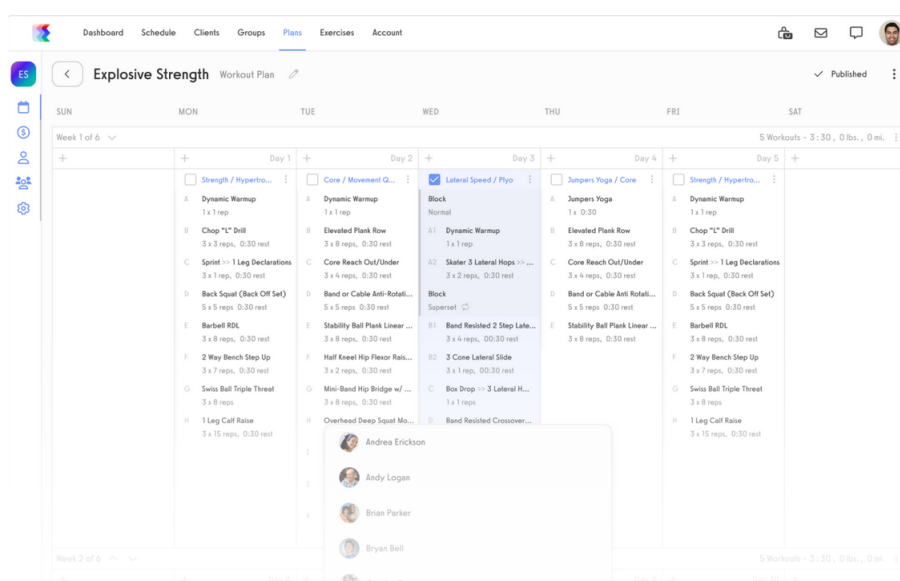
Nema mogućnost mijenjati raspored ili program koji su treneri definirali.

Aplikacija je dostupna na "Group Fitness Planner."

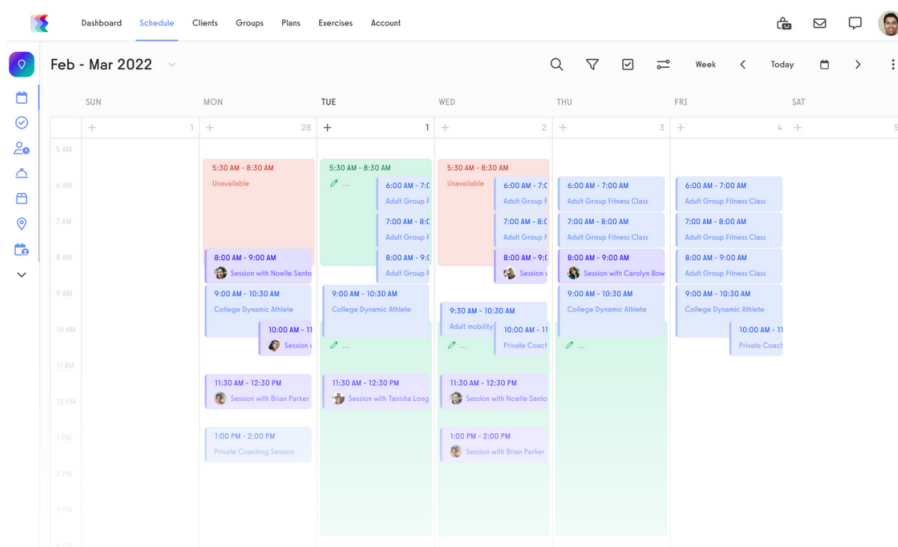
## 2.1 Pregled konkurentskih rješenja i analiza tržišta

Istražujući na internetu, naišli smo na nekolicinu aplikacija sa sličnim namjenama kao što je i naša. Kako se teži digitalizaciji i navika ljudi je glavnu informaciju o onome što ih zanima pronaći na internetu, tako i razna poduzeća teže imati kompletnu i interaktivnu internetsku uslugu preko koje se reklamiraju, organiziraju, a ponekad i djeluju raznim on-line sadržajima.

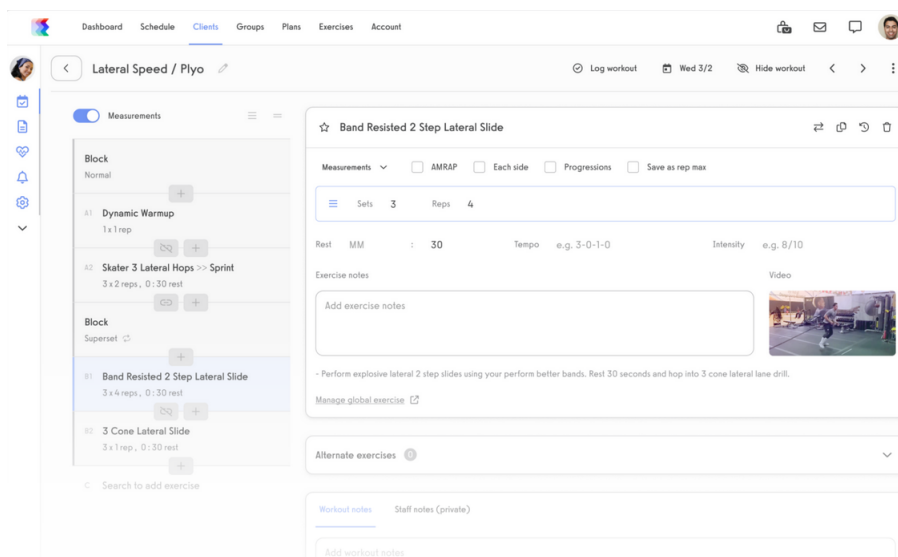
Primjerice, servis Exercise (<https://www.exercise.com/>) nudi jedinstvenu platformu za upravljanje fitness ustanovom, od mogućnosti registracije polaznika i pretplatnika na usluge, zaposlenika i voditelja, do registracije lokacija na kojima se odvijaju aktivnosti, e-potpisivanja ugovora i pretplata, kreiranja treninga, TV-treninga i biblioteke treninga i vježbi, statistike dolazaka na treninge, izvješća o napredovanju, online live prijenosa, izazova...



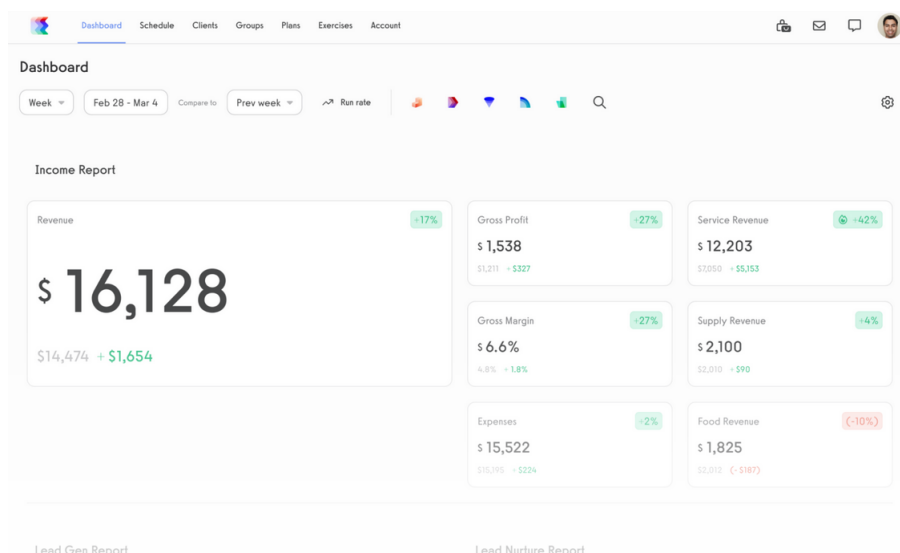
Slika 2.1: Exercise



Slika 2.2: Exercise

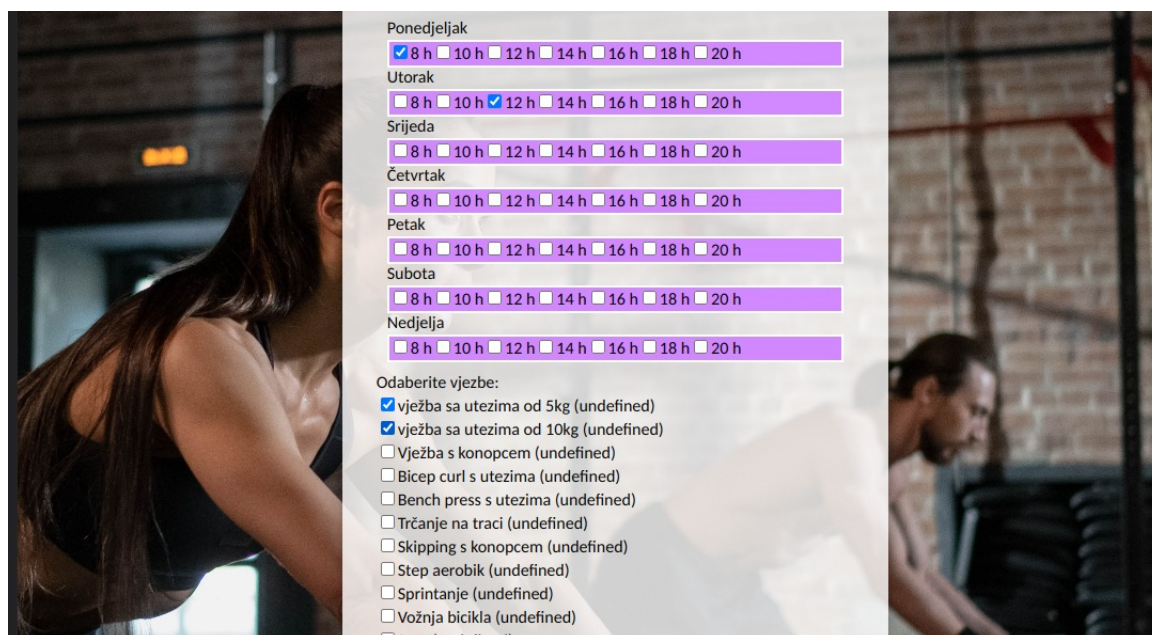


Slika 2.3: Exercise



Slika 2.4: Exercise

Exercise obuhvaća puno veći spektar usluga nego što obuhvaća naša „Group fitness planner“ aplikacija, pa je samim time naša aplikacija namijenjena užem krugu klijenata. Ciljani klijenti su manji fitness centri sa jednim prostorom za vježbanje, koji pomoću dobre organizacije i individualiziranog pristupa treninzima žele steći povjerenje kod svojih korisnika i imati maksimalnu učinkovitost u pogledu prilagođavanja vrste i termina treninga po želji i mogućnostima svojih korisnika. Aplikacija je jednostavna i intuitivna za korištenje, te sa samo nekoliko funkcionalnosti ispunjava svrhu opisanu opisom projektnog zadatka.



Slika 2.5: Group fitness planner

Osim opisanog servisa Excercise, na tržištu su dostupni i sljedeći servisi: Mind-body (<https://www.mindbodyonline.com/business/fitness-software>) aplikacija koja ima mnoge verzije, a svaka je prilagođena tipu poduzeća (za frizere, za fitness centre, za sportske klubove, za studije ljepote...) i ovisno o verziji sadržava funkcionalnosti potrebne za upravljanje tim poduzećem što intenzivno širi područja primjene aplikacije i skupinu ciljanih klijenata, Zenplanner (<https://zenplanner.com/>)...

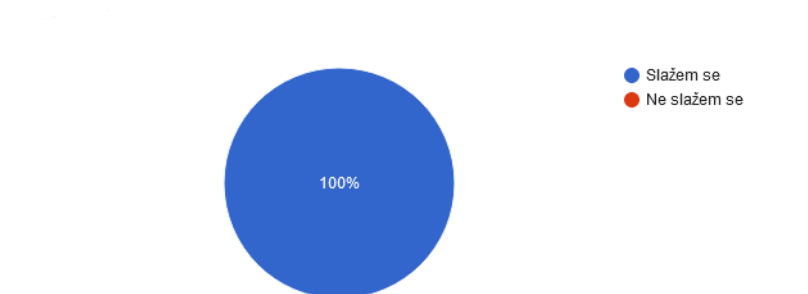
Da bismo dobili uvid u stvarni interes potencijalnih kupaca našeg proizvoda odlučili smo kontaktirati lokalni sportski klub i postaviti trenerima i polaznicima u klubu Google forms upitnik (<https://docs.google.com/forms>) .

Rezultati upitnika nalažu da su mišljenja oko aplikacije podijeljena. Dok su vježbači oduševljeno prihvatili osmišljeni sustav, treneri su rezerviraniji prema njemu i vide moguće nedostatke.

Vježbači se apsolutno slažu sa idejom o plaćanju treninga po terminima (umjesto mjesečno, unaprijed), te sa mogućnošću dolaska na različite tipove treninga, koji se poklapaju sa njihovim ciljevima.

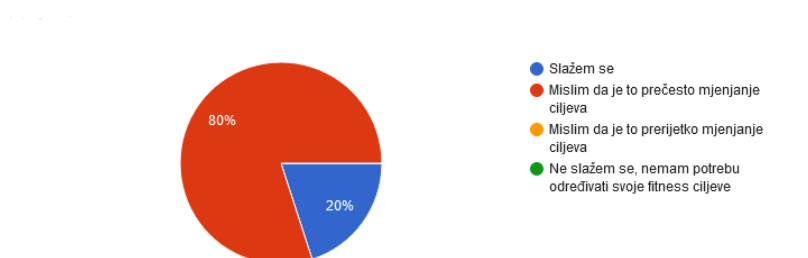
Zanimljivo je istaknuti kako vježbači smatraju da je određivanje fitness ciljeva na mjesečnoj bazi prečesto, te se sugerira timu da u budućem radu obrati pozornost na taj podatak.

Volio bih moći dolaziti na različite tipove treninga, kako mi koji dan odgovara radije nego da imam mjesečnu pretplatu na jednu aktivnost.



Slika 2.6: Rezultati ankete

Volio bih moći određivati svoje fitness ciljeve na mjesečnoj bazi.



Slika 2.7: Rezultati ankete

Također, vježbači su sugerirali da bi u ovakvoj aplikaciji voljeli vidjeti status svog napretka. Budućnost razvoja ovakve aplikacije mogla bi se usmjeriti na to, da uz postojeće funkcionalnosti, implementira i podatke o dolascima i redovitosti te napraviti stranicu sa statistikama konzistentnosti i napretka u postizanju željenih ciljeva. Također, voljeli bi imati „objašnjenja“ i neke osnovne edukativne materijale na stranici o tome koji su treninzi potrebni za koje ciljeve, koliko često se treba trenirati, koliko intenzivno... Iako smo mi na to vodili računa u implementaciji, korisnik iz korisničkog sučelja nema takav dojam, a aplikacija bi trebala biti jednostavna i intuitivna, te „voditi te“ kroz fitness iskustvo.

Treneri su skeptični prema ovakvoj aplikaciji jer smatraju da bi kvaliteta njihovog rada bila značajno umanjena, ponajviše time što ne bi imali stalne grupe polaznika na koje mogu računati, i koje već poznaju, pa time i pomažu i prilagođavaju određene zadatke. Smatraju kako bi pod izlikom „danas mi ne odgovara dani termin, pa ću odraditi neki drugi trening“ klijenti otežavali i usporavali dostizanje određenih ciljeva. Također, smatraju da u klasičnim grupama s vremenom mogu

zadavati sve teže i dinamičnije treninge, kako ljudi napreduju. Na ovakvim treninzima, uvijek bi bilo ljudi sa svim stupnjevima vještine, od početnika do iskusnih, te da bi bilo teško osmišljavati treninge takvim grupama.

Pri razvoju svakako treba poštovati mišljenja stručnjaka i naći rješenje koje će djelomično zadovoljavati obje strane. Treneri predlažu da se treninzi, uz obzir na odabir ciljeva, dijele i prema stupnju fizičke spremne. Da stari i novi polaznici imaju svoje zasebne termine, i da sustav prepozna da je neki vježbač prvih mjesec-dva po registraciji „početnik“, osim ako vježbač zbog svoje prethodne sportske podloge ne zatraži drugačije.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Vježbači
2. Treneri
3. Administrator
4. Baza podataka

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik može:
  - (a) se registrirati u sustav, stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, lozinka, ime, prezime, broj mobitela, e-mail adresa, datum rođenja, uloga, cilj (ovisno o ulozi)
2. Vježbač (primarni dionik) može:
  - (a) pregledavati i mijenjati osobne podatke
  - (b) birati i rezervirati određenu vrstu treninga u skladu sa postavljenim ciljevima
  - (c) otkazivati rezervacije
  - (d) odabrati željeni fitness cilj
  - (e) mijenjati ciljeve završetkom tekućeg mjeseca
3. Trener (primarni dionik) može:
  - (a) određivati vrstu treninga po terminima
  - (b) napraviti raspored treninga prema određenim ciljevima vježbača
  - (c) na temelju fonda sati koji je odabrao vježbač odrediti koliko različitih vrsta treninga vježbač mora odraditi
  - (d) izraditi trening
  - (e) definirati maksimalan broj polaznika za određenu vrstu treninga



4. Administrator (sekundarni dionik) može:

- (a) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (b) verificirati račune trenera
- (c) brisati korisničke račune

5. Baza podataka (sudionik) može:

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o treninzima, vježbama, terminima i njihovim opisima i ograničenjima

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registriraj korisnika

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Izrada korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Pristup aplikaciji putem web preglednika
- **Opis osnovnog tijeka:**
  1. Neregistrirani korisnik bira opciju "Registriraj se"
  2. Sustav otvara formu za upis podataka
  3. Neregistrirani korisnik unosi potrebne podatke (matični i kontakt podatci)
  4. Neregistrirani korisnik odabire tip registracije (klijent ili trener)
    - (a) Neregistrirani korisnik odabire tip registracije *trener* i sustav preskače korak 5.
    - (b) Neregistrirani korisnik odabire tip registracije *klijent*
  5. Neregistrirani korisnik odabire cilj
  6. Neregistrirani korisnik odabire gumb "Registriracija"
  7. Sustav provjerava i utvrđuje dostupnost korisničkog imena i maila te ispravnost ostalih unesenih podataka u skladu s zahtjevima sustava
  8. Sustav na upisani mail šalje se link za aktivaciju korisničkog računa
  9. Sustav pohranjuje potvrdu maila, aktivira korisnički račun i obavještava korisnika o uspješnoj registraciji
  10. Sustav preusmjerava registriranog korisnika na stranicu "Prijava"
- **Opis mogućih odstupanja:**
  - 6.a Neregistrirani korisnik odustaje od registracije
    1. Neregistriranog se korisnika preusmjerava na početnu stranicu
  - 7.a Sustav je provjerio i utvrdio da je neregistrirani korisnik unio neispravan oblik mail adrese, email već postoji u bazi podataka, unio je korisničko ime koje se već koristi ili je unio lozinku u nedozvoljene duljine
    1. Sustav obavještava neregistriranog korisnika o pogreškama te se izvođenje tijeka nastavlja u koraku 4.

##### UC2 - Prijavi korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijava u sustav
- **Sudionici:** Baza podataka, korisnik
- **Preduvjet:** Pristup aplikaciji putem web preglednika
- **Opis osnovnog tijeka:**
  1. Korisnik bira opciju "Prijava"
  2. Sustav otvara formu za upis podataka
  3. Korisnik upisuje svoje korisničko ime i lozinku
  4. Korisnik odabire potvrdu aplikacije
  5. Sustav provjerava i utvrđuje postojanje i ispravnost podataka unesenih u formu
  6. Sustav obavještava korisnika o uspješnoj prijavi
  7. Sustav preusmjerava korisnika na glavnu stranicu
  8. Korisnik dobiva pristup korisničkim funkcijama ovisnima o ulozi
- **Opis mogućih odstupanja:**
  - 4.a Korisnik odustaje od prijave
  - 5.a Sustav je provjerio i utvrdio da je korisnik upisao neispravno korisničko ime i/ili lozinku ili račun nije verificiran putem maila i/ili od strane administratora (za slučaj prijave trenera)
    1. Sustav obavještava korisnika o pogreškama te se izvođenje tijeka nastavlja u koraku 4.

### UC3 - Odjavi korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjavljivanje korisnika iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Prijavljeni korisnik bira opciju "Odjava"
  2. Sustav prijavljenom korisniku prikazuje prozor za potvrdu odjave
  3. Korisnik potvrđuje odjavu klikom
  4. Sustav odjavljuje korisnika
  5. Sustav preusmjerava korisnika na stranicu za prijavu
- **Opis mogućih odstupanja:**
  - 3.a Prijavljeni korisnik odustaje od odjave
    1. Sustav zatvara prozor za potvrdu odjave iz sustava

**UC4 - Pregledaj korisničke podatke**

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Prijavljeni korisnik bira opciju "Osobni podatci"
  2. Sustav preusjerava prijavljenog korisnika na stranicu "Osobni podatci"
  3. Sustav učitava iz baze podataka osobne podatke prijavljenog korisnika(matični i kontakt podatci)
  4. Sustav prikazuje učitane podatke na stranici
  5. Korisnik pregledava prikazane podatke

**UC5 - Promijeni korisničke podatke**

- **Glavni sudionik:** Korisnik
- **Cilj:** Promjena osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Promjena podataka"
  2. Sustav otvara formu za promjenu podataka
  3. Prijavljeni korisnik mijenja podatke po želji
  4. Prijavljeni korisnik odabire potvrdu aplikacije
  5. Sustav provjerava i utvrđuje jesu li novi podatci ispravnog oblika
  6. Sustav pohranjuje nove podatke u bazu podataka
  7. Sustav obavještava prijavljenog korisnika o uspješnoj promjeni
- **Opis mogućih odstupanja:**
  - 4.a Prijavljeni korisnik je odustao od promjene
    1. Sustav preusmjerava prijavljenog korisnika na stranicu "Osobni podatci"
  - 5.a Sustav je utvrdio da uneseni podatci nisu ispravnog formata
    1. Sustav obavještava prijavljenog korisnika o pogreškama te se izvođenje tijeka nastavlja u koraku 3.

**UC6 - Obriši korisnički račun**

- **Glavni sudionik:** Admin

- **Cilj:** Brisanje korisničkog računa iz baze podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator mora biti prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Administrator bira opciju "Obriši korisnički račun"
  2. Administratoru se prikazuje prozor u kojem potvrđuje brisanje korisničkog računa
  3. Administrator odabire potvrdu aplikacije
  4. Sustav briše račun iz baze podataka
  5. Sustav obavještava administratora o uspješnom brisanju računa
  6. Sustav obavještava korisnika o brisanju računa
- **Opis mogućih odstupanja:**
  - 3.a Administrator odustaje od brisanja računa
    1. Zatvaranje prozora za potvrdu brisanja računa iz sustava

#### UC7 - Promijeni cilj

- **Glavni sudionik:** Klijent
- **Cilj:** Odabir cilja sudjelovanja u programu vježbanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Zadnji je tjedan tekućeg mjeseca
- **Opis osnovnog tijeka:**
  1. Klijent bira opciju "Promjena cilja"
  2. Sustav prikazuje klijentu ciljeve koje može odabrati
  3. Klijent odabire cilj koji želi
  4. Klijent odabire potvrdu aplikacije
  5. Sustav pohranjuje odabrani cilj
  6. Sustav uklanja klijenta iz liste klijenata trenera te uklanja sve dodijeljene treninge klijentu zbog promjene cilja
  7. Sustav obavještava klijenta o uspješnoj promjeni cilja
  8. Klijenta se preusmjerava na glavnu stranicu
- **Opis mogućih odstupanja:**
  - 4.a Klijent odustaje od promjene cilja
    1. Sustav preusmjerava klijenta na glavnu stranicu

#### UC8 - Izradi trening

- **Glavni sudionik:** Trener

- **Cilj:** Izrada treninga
- **Sudionici:** Baza podataka
- **Preduvjet:** Trener mora biti prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Trener bira opciju "Izradi trening"
  2. Sustav otvara prozor za izradu treninga s formom za unos podataka o treningu
  3. Trener unosi glavne podatke o treningu
  4. Trener odabire dane u tjednu u kojima bi se izvodio trening
  5. Sustav za svaki dan nudi opciju odabira slobodnog termina
  6. Trener odabire po jedan termin za svaki odabrani dan
  7. Trener odabire opciju "Potvrdi" unutar forme
  8. Sustav provjerava jesu li podaci ispravnog oblika
  9. Sustav pohranjuje podatke o treningu i njegovim terminima u bazu podataka
  10. Sustav obavještava trenera o uspješno unesenom treningu
  11. Sustav preusmjerava trenera na stranicu svih njegovih treninga
- **Opis mogućih odstupanja:**
  - 8.a Trener odustaje od izrade treninga
    1. Trenera se preusmjerava na stranicu svih njegovih treninga
  - 9.a Sustav je utvrdio da je trener unio neispravan oblik podataka o treningu
    1. Sustav obavještava trenera o pogreškama te se izvođenje nastavlja od koraka 4.

#### UC9 - Pregledaj klijenate kojima nisu dodijeljeni treninzi

- **Glavni sudionik:** Trener
- **Cilj:** Dobiti uvid u klijente kojima nije dodijeljen trening
- **Sudionici:** Baza podataka
- **Preduvjet:** Trener je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Trener odabire opciju "Prikaži klijente bez dodijeljenog treninga"
  2. Sustav dohvaća podatke iz baze podataka o klijentima bez dodijeljenog treninga
  3. Sustav prikazuje dohvaćene podatke o klijentima na stranici
  4. Trener pregledava prikazane klijente

#### UC10 - Dodijeli treninge klijentu

- **Glavni sudionik:** Trener
- **Cilj:** Dodjela treninga klijentu
- **Sudionici:** Baza podataka, klijent
- **Preduvjeti:** Klijent je registriran u sustav, trener je priavljen u sustav
- **Opis osnovnog tijeka:**
  1. Trener odabire klijenta sa stranice "Prikaz klijenata bez dodijeljenog treninga"
  2. Treneru se za dodjelu prikazuju svi mogući treninzi sa opcijom "Dodijeli"
  3. Trener odabire treninge za dodjelu opcijom "Dodijeli"
  4. Trener unosi broj sati predviđenih za odrađivanje odabranih treninga klijenta na mjesečnoj bazi
  5. Trener potvrđuje dodijeljene treninge i unos odabirom opcije "Spremi"
  6. Sustav ispituje ispravnost spremljenih podataka
  7. Sustav sprema dodijeljene treninge i broj sati u bazu podataka
  8. Sustav obavještava trenera o uspješno dodijeljenim treninzima
  9. Sustav obavještava klijenta o dodijeljenim treninzima putem emaila
- **Opis mogućih odstupanja:**
  - 6.a Sustav je utvrdio da trener nije odabrao ni jedan trening i/ili unio broj sati
    1. Sustav ne sprema promjene, tj. ne evidentira klijenta kao onog kojemu su dodijeljeni treninzi te se izvođenje nastavlja u koraku 1.
  - 5.a Trener nije potvrdio dodijeljene treninge i/ili unio broj sati
    1. Sustav obavještava trenera o nespremljenim promjenama te se izvođenje nastavlja u koraku 1.

### UC11 - Pregledaj termine dodijeljenih treninga

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati dodijeljene treninge i termine
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je na stranici "Pregled dodijeljenih treninga"
- **Opis osnovnog tijeka:**
  1. Klijent odabire trening sa stranice "Pregled dodijeljenih treninga"
  2. Sustav dohvaća termine odabranog treninga iz baze podataka
  3. Sustav prikazuje klijentu termine odabranog treninga
  4. Klijent pregledava termine prikazanog treninga

**UC12 - Rezerviraj termin dodijeljenog treninga**

- **Glavni sudionik:** Klijent
- **Cilj:** Rezervirati termin treninga
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent mora imati dodijeljen trening
- **Opis osnovnog tijeka:**
  1. Klijent je na stranici "Moji treninzi"
  2. Klijent odabire termin među ponuđenima
  3. Klijent klikom na gumb "Rezerviraj" rezervira termin
  4. Sustav otvara prozor u kojem traži potvrdu rezervacije termina
  5. Klijent odabire potvrdu aplikacije
  6. Sustav provjerava ograničenja sustava
  7. Sustav unosi podatak o rezervaciji u bazu podataka
  8. Sustav smanjuje fond sati klijenta za jedan
  9. Sustav smanjuje broj raspoloživih mjesta u rezervaciji treninga za jedan
  10. Sustav obavještava klijenta o uspješnoj rezervaciji
- **Opis mogućih odstupanja:**
  - 5.a Klijent odustaje od rezervacije
    1. Klijent je preusmjeren na stranicu prikaza svih termina njemu dodijeljenih treninga
  - 6.a Sustav je utvrdio da rezervacija nije u skladu s ograničenjima sustava
    1. Sustav obavještava klijenta da rezervaciju nije moguće izvršiti zbog ograničenja sustava te se izvođenje nastavlja od koraka 1.

**UC13 - Otkazi rezervaciju termina treninga**

- **Glavni sudionik:** Klijent
- **Cilj:** Otkazati rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:**
- **Opis osnovnog tijeka:**
  1. Klijent je na stranici "Pregled dodijeljenih treninga"
  2. Klijent odabire trening koji želi otkazati
  3. Klijent klikom na gumb "Otkazi" otkazuje rezervaciju
  4. Sustav otvara prozor u kojem traži potvrdu otkaza rezervacije od klijenta
  5. Klijent odabire potvrdu aplikacije



6. Sustav briše rezervaciju iz baze podataka
  7. Sustav povećava fond sati klijenta za jedan
  8. Sustav povećava broj mjesta treninga u odabranom terminu za jedan
  9. Sustav obavještava klijenta o uspješno otkazanoj rezervaciji
- **Opis mogućih odstupanja:**
    - 5.a Klijent odustaje od otkaza rezervacije
      1. Klijent je preusmjeren na stranicu prikaza svojih rezervacija

#### UC14 - Pregledaj rezervacije

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati rezervirane termine treninga
- **Sudionici:** Baza podataka
- **Preduvjet:**
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju "Pregled rezervacija"
  2. Sustav dohvaća rezervacije termina trenutnog klijenta iz baze podataka
  3. Sustav prikazuje klijentu rezervacije termina treninga
  4. Klijent pregledava prikazane rezervacije

#### UC15 - Pregledaj sve korisničke račune

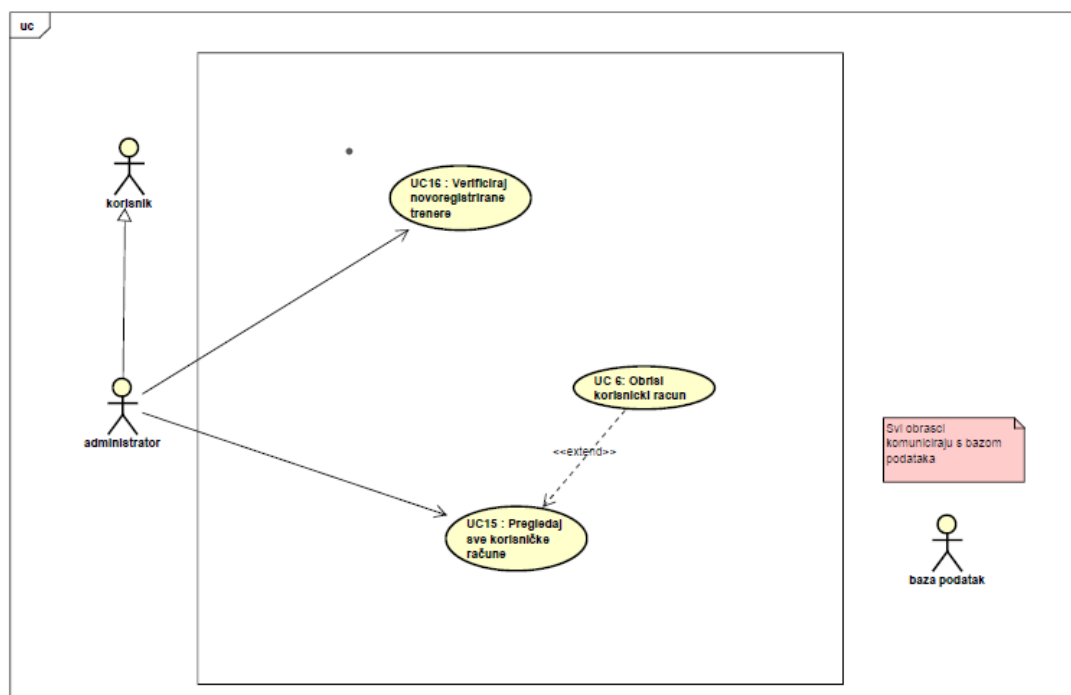
- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati sve korisničke račune
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator mora biti prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju "Pregled svih korisničkih računa"
  2. Sustav preusmjerava administratora na stranicu pregleda računa
  3. Sustav učitava sve račune iz baze podataka
  4. Sustav prikazuje administratoru račune s oznakom tipa računa (klijent, trener) i opcijom "Odobri" pored računa trenera
  5. Administrator pregledava prikazane račune

#### UC16 - Verificiraj novoregistrirane trenere

- **Glavni sudionik:** Administrator
- **Cilj:** Verifikacija novoregistriranih trenera
- **Sudionici:** Baza podataka

- **Preduvjet:** Administrator mora biti prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator je na stranici "Pregled svih korisničkih računa"
  2. Sustav učitava sve korisničke račune
  3. Administrator odabire trenera kojeg želi odobriti
  4. Administrator obabire gumb "Odobri"
  5. Sustav otvara prozor u kojem traži potvrdu administratora
  6. Administrator odabire potvrdu sustava
  7. Sustav ažurira bazu podataka
  8. Sustav obavještava administratora o uspješnoj verifikaciji
- **Opis mogućih odstupanja:**
  - 4.a Administrator odustaje od verifikacije trenera
    1. Administrator je preusmjeren na stranicu "Pregled svih korisničkih računa"

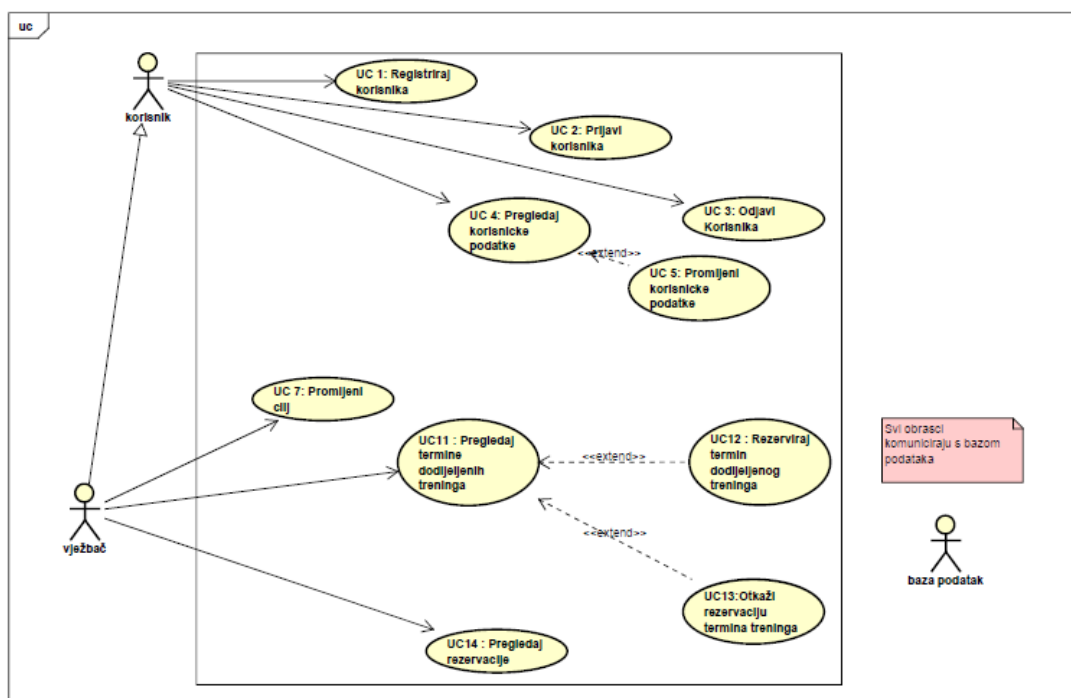
### 3.1.2 Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe sa funkcionalnostima administratora



Slika 3.2: Dijagram obrasca uporabe sa funkcionalnostima trenera

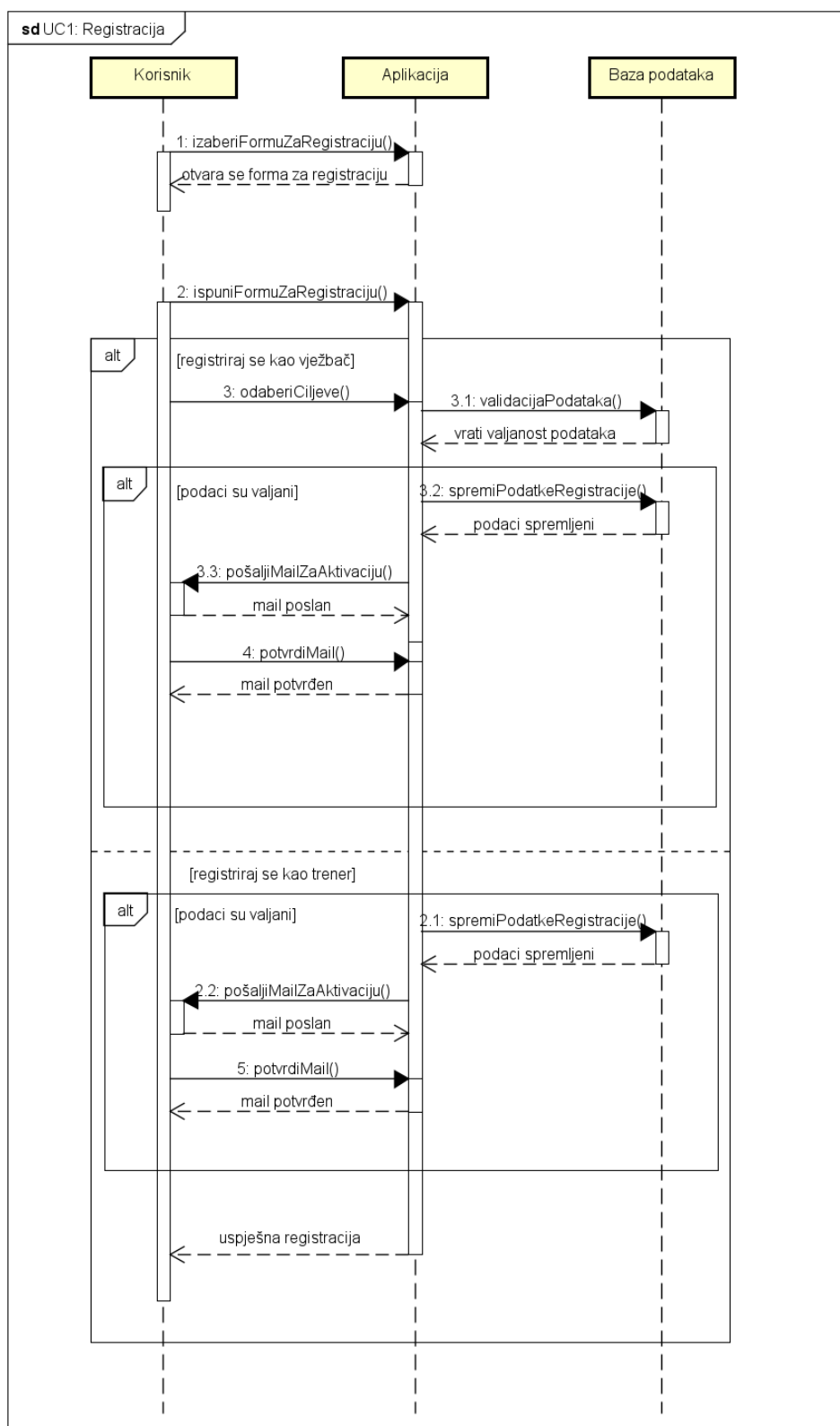


Slika 3.3: Dijagram obrasca uporabe sa funkcionalnostima vježbača

### 3.1.3 Sekvencijski dijagrami

#### Obrazac uporabe UC1: Registracija

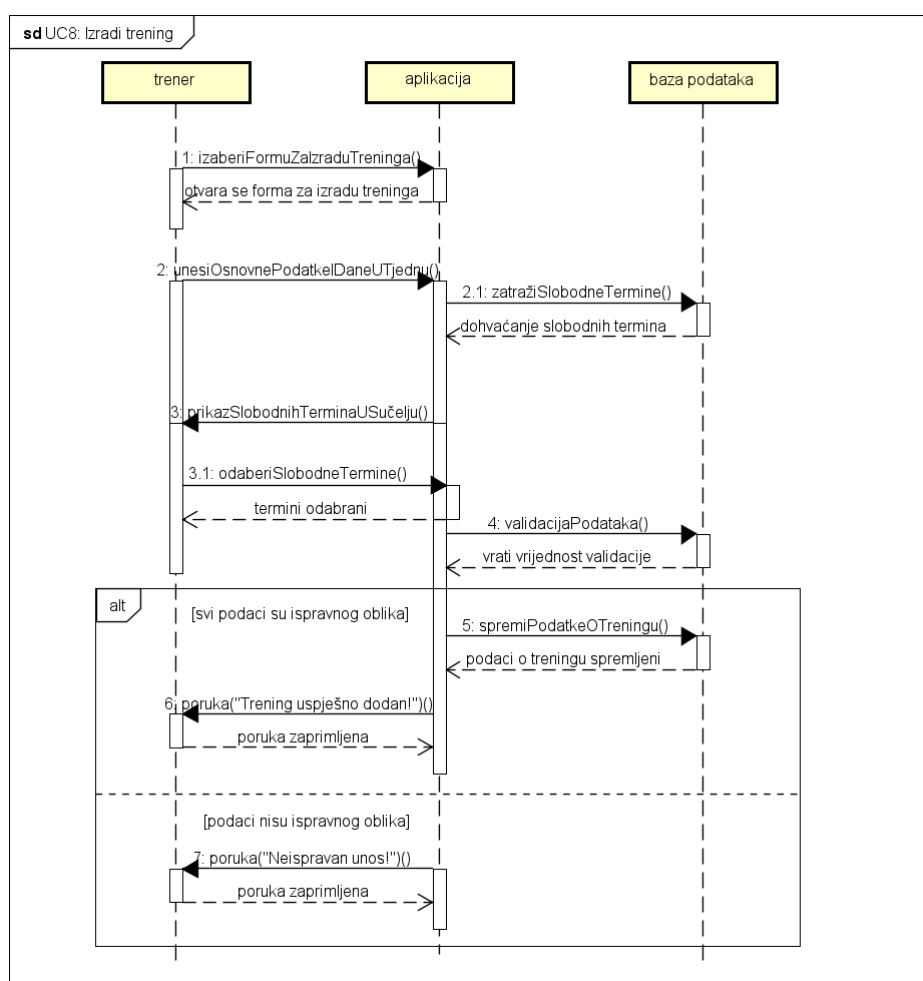
Korisnik bira opciju za registraciju, te poslužitelj prikazuje formu za registraciju. Zatim korisnik unosi svoje ime, prezime, datum rođenja, e-mail adresu i broj telefona. Osmišlja korisničko ime i lozinku za prijavu, te odabire opciju želi li se registrirati kao vježbač ili kao trener. U slučaju da korisnik odabire opciju vježbač, odabire cilj. Po završetku ispunjavanja forme korisnik klikne na gumb registriraj se. Sustav provjerava dostupnost korisničkog imena, e-mail adrese, te ispravnost ostalih unesenih podataka u skladu sa zahtjevima sustava. Ukoliko su podaci ispravni, sustav šalje link za aktivaciju korisničkog računa na upisanu e-mail adresu i obavještava korisnika o uspješnoj registraciji.



Slika 3.4: Sekvencijski dijagram registracije

## Obrazac uporabe UC8: Izradi trening

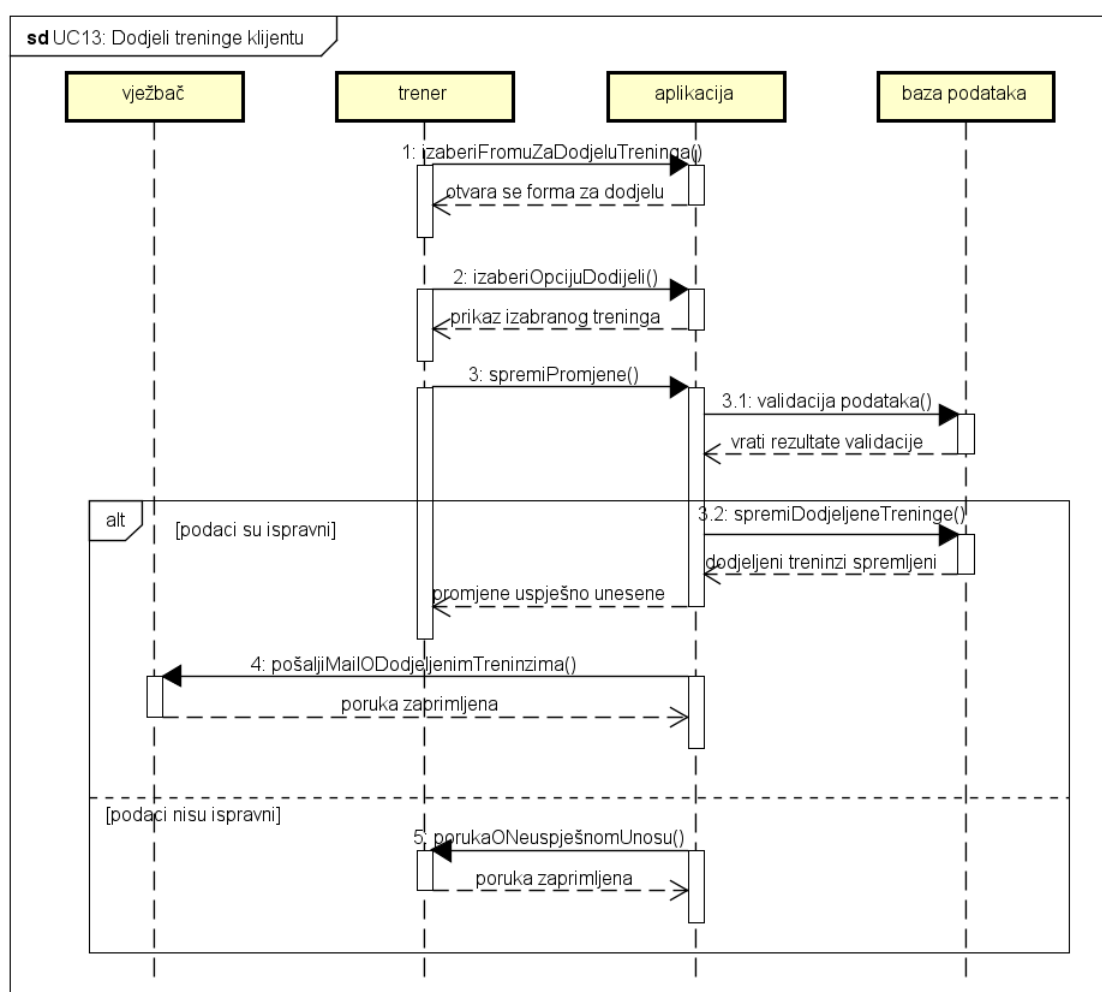
Trener bira opciju "Izradi trening". Sustav otvara prozor za izradu treninga s formom za unos podataka o treningu. Trener unosi glavne podatke o treningu i odabire dane u tjednu u kojima bi se izvodio trening. Sustav za svaki dan nudi slobodne termine treninga koje zatim trener odabire za terning u izradi. Na kraju trener odabire opciju "Potvrdi", a sustav provjerava jesu li svi podaci ispravnog oblika. Ukoliko jesu sustav pohranjuje podatke o kreiranom treningu i njegovim terminima te ih sprema u bazu podataka. Ukoliko je sustav utvrdio da je trener unio neispravan oblik podataka o treningu, na zaslonu se pojavi poruka "Došlo je do pogreške".



Slika 3.5: Sekvencijski dijagram izrade treninga

## Obrazac uporabe UC10: Dodjeli treninge klijentu

Trener klikom na gumb "Prikaz klijenata bez dodjeljenog treninga" otvara formu. Unutar te forme treneru se prikazuju svi mogući treninzi sa opcijom "Dodijeli". Treneru se odabirom opcije "Dodijeli" omogućuje i unos broja sati predviđenih za odrađivanje odabranih treninga. Trener potvrđuje odabrane treninge klikom na gumb "Spremi". Sustav ispituje ispravnost spremljenih podataka. Ukoliko su podaci ispravni, sustav sprema dodjeljene treninge i broj sati potrebnih da se oni odrade u bazu podataka. Zatim obavještava trenera o uspješno dodjeljenim treninzima iz sučelja aplikacije, a klijenta putem maila. Ukoliko podaci nisu ispravni aplikacija obavještava trenera o neuspješnom unosu treninga.

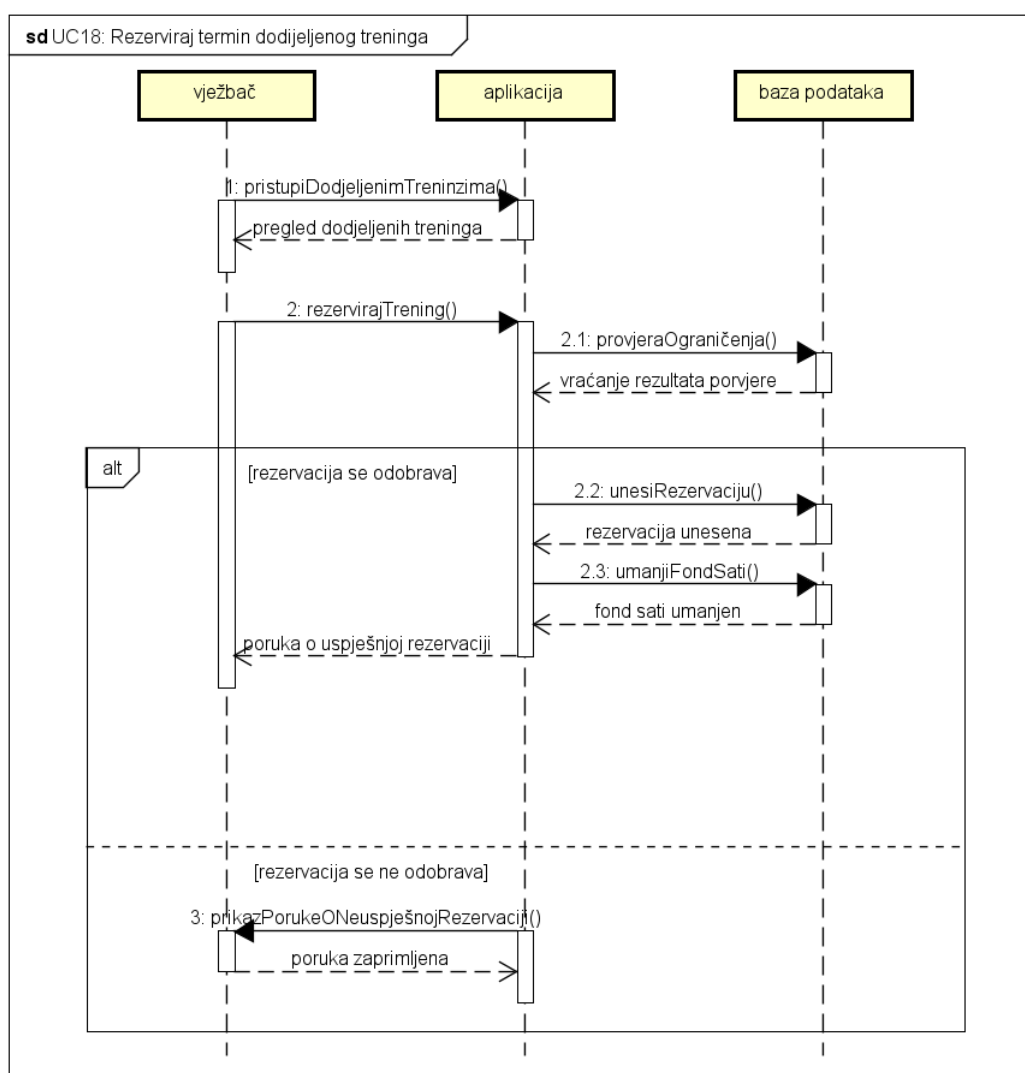


Slika 3.6: Sekvencijski dijagram dodjele treninga klijentima



**Obrazac uporabe UC12: Rezerviraj termin dodijeljenog treninga**

Klijent ima pristup formi "Pregled dodijeljenih treninga". Klijent odabire termin među ponuđenima, a klikom na gumb "Rezerviraj" rezervira termin treninga. Sustav provjerava ograničenja. Ukoliko provjera odobrava rezervaciju treninga, aplikacija unosi podatak o rezervaciji u bazu podataka, smanjuje fond sati klijenta za jedan, smanjuje broj raspoloživim mjesta u rezervaciji treninga, i obavještava klijenta o uspješnoj rezervaciji u sučelju aplikacije. U slučaju neuspješne provjere sustav u sučelju aplikacije izdaje obavijest da rezervaciju nije moguće izvršiti zbog ograničenja.



Slika 3.7: Sekvencijski dijagram rezervacije dodijeljenog termina

## 3.2 Ostali zahtjevi

Aplikacija treba biti izvedena kao web aplikacija kojoj će korisnici pristupati uz pomoć korisničkog imena i lozinke.

Aplikacija treba biti jednostavna za korištenje, a sučelje pregledno i intuitivno. Osim toga, aplikacija treba biti prilagođena za rad na različitim uređajima (mobilni uređaj, tablet, PC).

Aplikaciju treba implementirati u arhitekturi klijent-poslužitelj. Na poslužiteljskoj strani koristiti programski jezik Java i radni okvir Spring Boot, spremati podatke u relacijsku bazu podataka koristeći JPA, a potrebnu funkcionalnost izložiti kroz REST Web servise.

Na klijentskoj strani implementirati korisničko sučelje u Web pregledniku koristeći React ili Angular, koje se spaja na navedene servise.

## 4. Arhitektura i dizajn sustava

Komunikacija sa našim sustavom ostvaruje se korištenjem web preglednika koji prikazuje grafičko sučelje naše aplikacije. Korisnička interakcija sa grafičkim sučeljem rezultira slanjem zahtjeva web preglednika prema web poslužitelju na kojem se nalazi naša web aplikacija.

Iz navedenog možemo definirati **web preglednik** kao program koji krajnjem korisniku omogućava jednostavno pretraživanje i pregled podataka na Internetu, **web poslužitelj** kao program koji prima zahtjeve web preglednika, prosljeđuje ih web aplikaciji na obradu te odgovara web pregledniku podacima koji su mu prosljeđeni od strane web aplikacije. **Web aplikacija** program je koji obrađuje primljene zahtjeve, komunicira s bazom podataka te šalje podatke web poslužitelju za prikaz u web pregledniku.

U našem slučaju web poslužitelju se pristupa putem REST-a (engl. *Representational State Transfer*) te je odgovor poslužitelja u JSON obliku.

Troslojna arhitektura našeg sustava prezentirana je sljedećim slojevima

- Sloj interakcije
- Sloj servisa
- Sloj repozitorija

Interakcijski sloj predstavlja sučelje aplikacije pomoću kojeg krajnji korisnik iskorištava ostvarene funkcionalnosti aplikacije.

Sastoji se od dva podsloja:

- Prikazni sloj
- Kontrolerski sloj

Prikazni sloj dio je *frontend* dijela sustava. Za njegovu implementaciju odabran je React razvojni okvir (engl. *framework*).

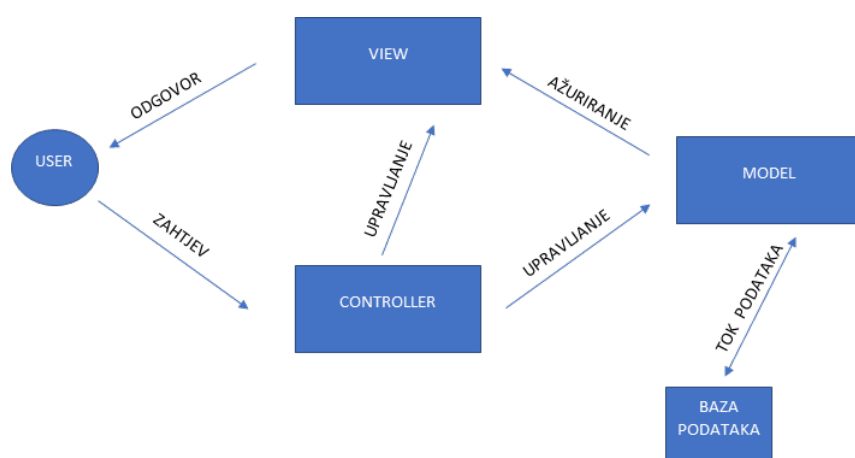
Kontrolerski sloj dio je *backend* dijela sustava. Zadužen je za posluživanje zahtjeva dobivenih od prikaznog sloja.

Servisni sloj naše aplikacije sadržan je unutar *backend* dijela sustava. Sadrži strukture i funkcije čija je zadaća ispunjavanje zamišljenih funkcionalnosti aplikacije.

Unutar njega se odrađuju izračuni i obrada podataka koji se onda predaju podatkovnom sloju.

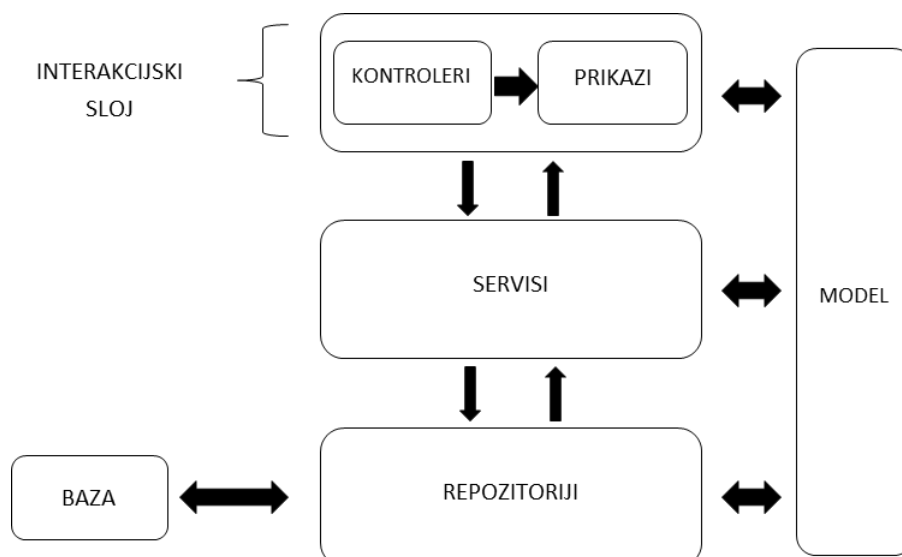
Repozitorski je sloj također sadržan unutar već navedenog *backend* dijela sustava. Zadužen je za komunikaciju s bazom podataka i razmjenom navedenih podataka sa servisnim slojem.

Navedeni opis slojeva odgovara MVC arhitekturi (engl. *Model-View-Controller*) čija je osnovna forma prikazana na sljedećoj slici.



Slika 4.1: Opći MVC obrazac

Korištenjem Spring razvojnog okvira (engl. *framework*) za razvoj aplikacije sloj prikaza i sloj kontrolera spojeni su unutar jednog sloja (koji možemo zvati sloj interakcije), dok su ostala dva sloja ostala odvojena.



Slika 4.2: Spring MVC obrazac

Svi navedeni slojevi ostvaruju svoje funkcionalnosti komunikacijom s modelima (engl. *models*) sustava. Oni su oblikovani na način da apstrahiraju korisnike i elemente sustava te enkapsuliraju attribute potrebne za ostvarenje zamišljenih funkcionalnosti sustava.

Spring razvojni okvir (engl. *framework*) pohranjuje strukturu modela unutar baze podataka korištenjem objektno-relacijskog preslikavanja (engl. *object-relational mapping, ORM*) koje pruža alat Hibernate ORM. Preslikavanje se odvija prema predodređenim Java specifikacijama unutar Java Persistence aplikacijskog sučelja (JPA) koje alat kao što je Hibernate ORM implementira. Rezultat je skup tablica unutar odabrane baze podataka koje sadrže attribute modela i attribute veza među modelima.

Sustav upravljanja bazama podataka (engl. *Relation Database Management System, RDBMS*) koji smo odabrali za izradu aplikacije je PostgreSQL. Jednostavno ga je integrirati unutar Spring razvojnog okvira (engl. *framework*) uključivanjem tzv. *dependecya*.

## 4.1 Baza podataka

Za potrebe sustava modelirana je relacijska baza podataka koja svojom strukturom modelira stvarni svijet, tj. odnose potrebne aplikaciji. Gradivna jedinica baze je relacija, tablica definirana svojim imenom i skupom atributa. Zadaća baze podataka brza je i jednostavna pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- User
- Client
- Coach
- Training
- Workout
- Time
- Goals
- Schedule
- Token
- Training\_workouts
- Client\_schedules
- Coach\_clients
- Client\_trainings

#### 4.1.1 Opis tablica

##### USER

Entitet USER sadržava informacije o korisniku aplikacije. Korisnik može biti trener, klijent i administrator. Sadrži attribute: userID, name, surname, dateOfBirth, username, email, password, contact, role.

USER		
userID	BIGINT	jedinstveni identifikator korisnika
name	VARCHAR	ime korisnika
surname	VARCHAR	prezime korisnika
dateOfBirth	DATE	datum rođenja korisnika
email	VARCHAR	e-mail adresa korisnika
username	VARCHAR	korisničko ime korisnika, alternativni ključ
password	VARCHAR	hash zapis lozinke
contact	VARCHAR	kontakt broj korisnika
role	ENUM	uloga korisnika u sustavu

##### CLIENT

Entitet CLIENT predstavlja specijalizaciju nad entitetom USER. Sadrži attribute en-

titete USER te currentGoal, nextGoal, hoursAvailable.

CLIENT		
currentGoal	VARCHAR	trenutni cilj
nextGoal	VARCHAR	sljedeći odabrani cilj
hoursAvailable	INT	fond sati treninga po mjesecu

### COACH

Entitet COACH predstavlja specijalizaciju nad entitetom USER. Sadrži atribute entiteta USER i atribut verified koji predstavlja odobrenje administratora

COACH		
verified	BOOLEAN	status verifikacije trenera

### TRAINING

Entitet TRAINING sadržava informacije o treningu. Sadrži atribute: trainingID, userID, trainingName, duration, trainingRules. Entitet je u vezi Many-to-Many s entitetom TRAINING\_WORKOUTS preko atributa trainingID.

TRAINING		
trainingID	INT	jedinstveni identifikator treninga
coachUserID	INT	jedinstveni identifikator trenera
trainingName	VARCHAR	naziv treninga
duration	TIME	duljina treninga
trainingRules	VARCHAR	pravila koja postavlja trener

### WORKOUT

Entitet WORKOUT sadržava sve moguće vježbe unutar aplikacije. Sadrži atribute: workoutID, workoutName, workoutTypeID. Entitet je u vezi Many-To-Many s TRAINING\_WORKOUTS preko atributa workoutID, te u vezi Many-To-One s WORKOUT\_TYPE preko atributa workoutTypeID;

WORKOUT		
workoutID	INT	jedinstveni identifikator vježbe
workoutName	VARCHAR	naziv vježbe
workoutTypeID	BIGINT	identifikator tipa vježbe

### TIME

Entitet TIME sadržava moguće termina treninga unutar dana. Sadrži atribut time-Ofday.

TIME		
timeOfDay	INT	termin treninga unutar dana

### GOALS

Entitet GOALS sadržava moguće ciljeve vježbanja za korisnika. Sadrži attribute goalName.

GOALS		
goalName	VARCHAR	jedinstveni identifikator i naziv cilja

### SCHEDULE

Entitet SCHEDULE sadržava informacije o održavanju treninga unutar dana, te broju mjesta slobodnih za određeni trening. Sadrži attribute trainingID, date, timeOfDay, spaceLeft. TrainingID strani je ključ entiteta TRAINING te s atributom date čini primarni ključ.

SCHEDULE		
trainingID	BIGINT	jedinstveni identifikator treninga
date	DATE	datum treninga
timeOfDay	INT	termin treninga u danu
spaceLeft	INT	broj slobodnih mjesta

### TRAINING\_WORKOUTS

TRAINING\_WORKOUTS predstavlja spojnu tablicu treninga i vježbi koja će se koristiti za određivanje vrste treninga. Sadrži attribute trainingID i workoutID koji su



strani ključevi redom tablica TRAINING i WORKOUT.

TRAINING_WORKOUTS		
trainingID	BIGINT	jedinstveni identifikator treninga
workoutID	BIGINT	jedinstveni identifikator vježbe

### CLIENT\_SCHEDULES

CLIENT\_SCHEDULES predstavlja spojnu tablicu koja povezuje klijenta s dodijeljenim treninzima i njihovim terminima. Sadrži attribute userID, trainingID, date, reservation. UserID i trainingID redom su strani ključevi iz tablica USER i TRAINING.

CLIENT_SCHEDULES		
userID	BIGINT	jedinstveni identifikator korisnika
trainingID	BIGINT	jedinstveni identifikator treninga
date	DATE	datum treninga
reservation	BOOLEAN	potvrda rezervacije termina treninga

### COACH\_CLIENTS

COACH\_CLIENTS predstavlja spojnu tablicu koja povezuje trenera sa svojim klijentima. Sadrži attribute coachID, clientID. CoachID i ClientID redom su strani ključevi iz tablica COACH i CLIENT u kojima se pojavljuju kao userID.

COACH_CLIENTS		
coachID	BIGINT	jedinstveni identifikator trenera
clientID	BIGINT	jedinstveni identifikator klijenta

### CLIENT\_TRAININGS

CLIENT\_TRAININGS predstavlja spojnu tablicu koja povezuje klijenta sa dodijeljenim treninzima. Sadrži attribute userID te trainingID. UserID i trainingID redom su strani ključevi iz tablica CLIENT i TRAINING.

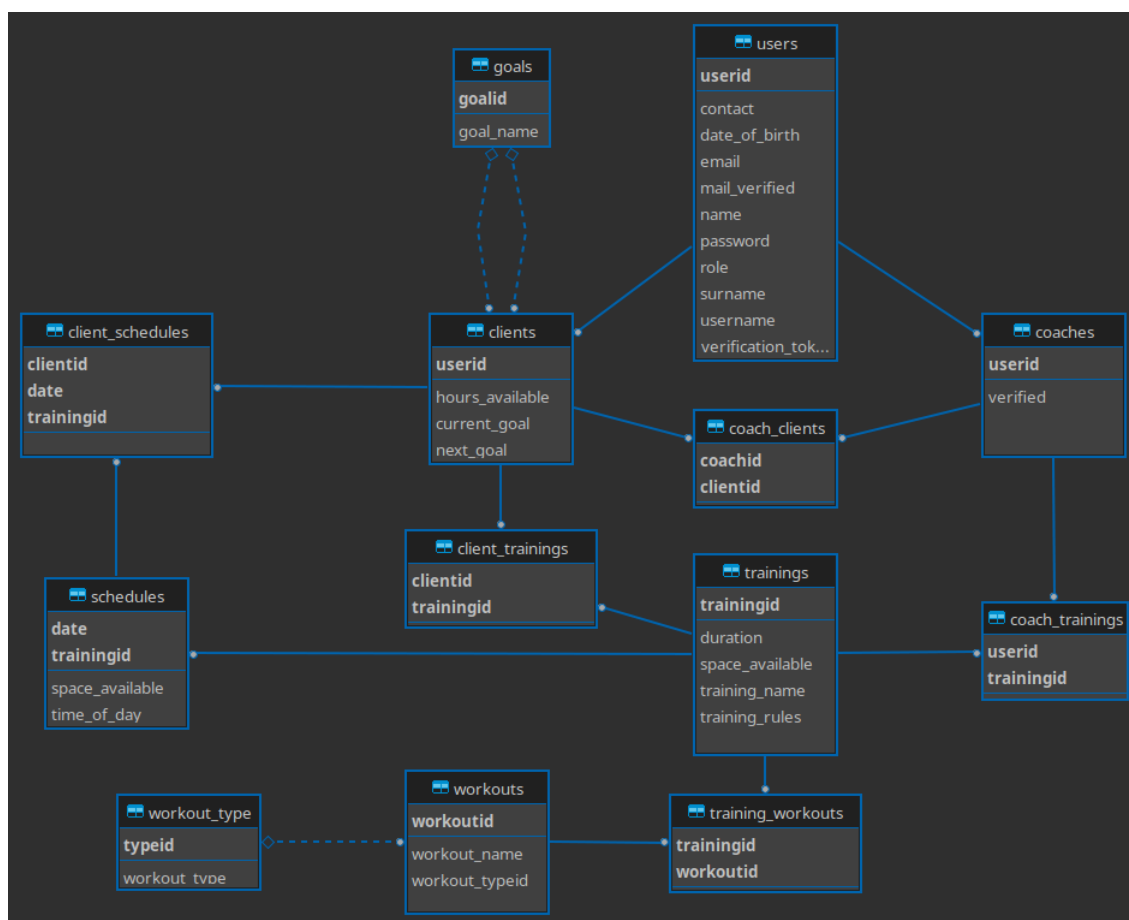
CLIENT_TRAININGS		
userID	BIGINT	jedinstveni identifikator klijenta
trainingID	BIGINT	jedinstveni identifikator treninga

### WORKOUT\_TYPE

WORKOUT\_TYPE predstavlja tablicu koja sadrži podatke o mogućim tipovima vježbi. Sadrži attribute typeID i workoutType.

WORKOUT_TYPE		
typeID	BIGINT	jedinstveni identifikator tipa vježbe
workoutType	VARCHAR	naziv tipa vježbe

## 4.1.2 Dijagram baze podataka



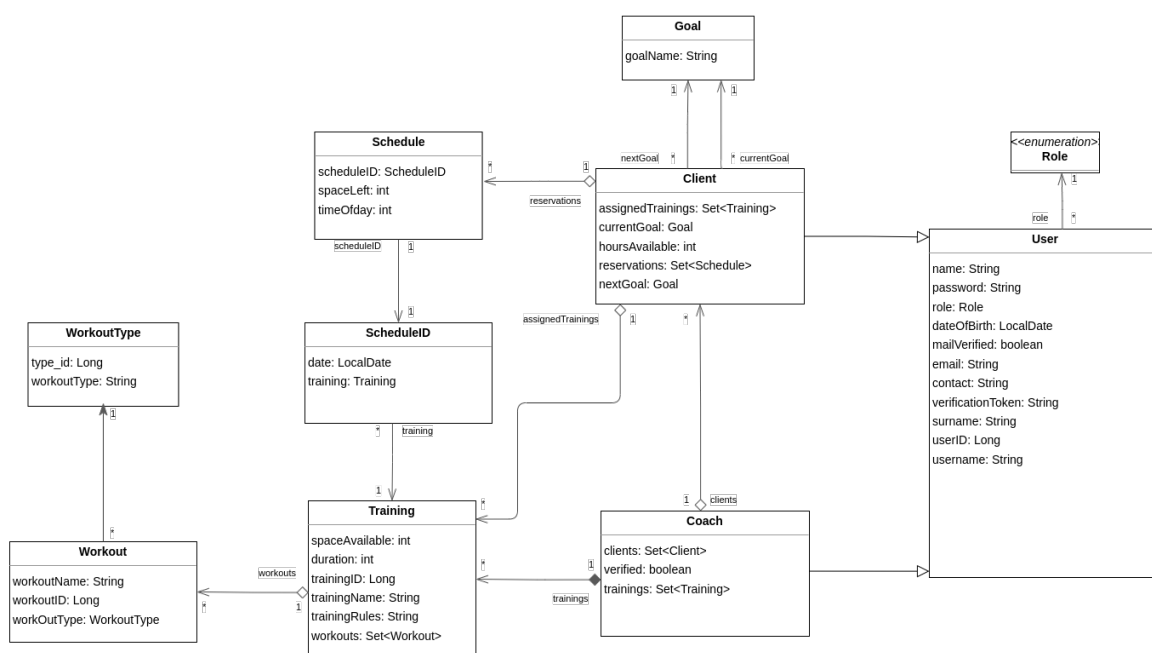
Slika 4.3: Dijagram tablica unutar baze podataka

## 4.2 Dijagram razreda

Dijagrami razreda pokazuju povezanost komponenti sustava te daju jasniji uvid u arhitekturu i dizajn aplikacije. Podijeljeni su u 4 podkategorije (sa dodatnim prikazom metoda servisa):

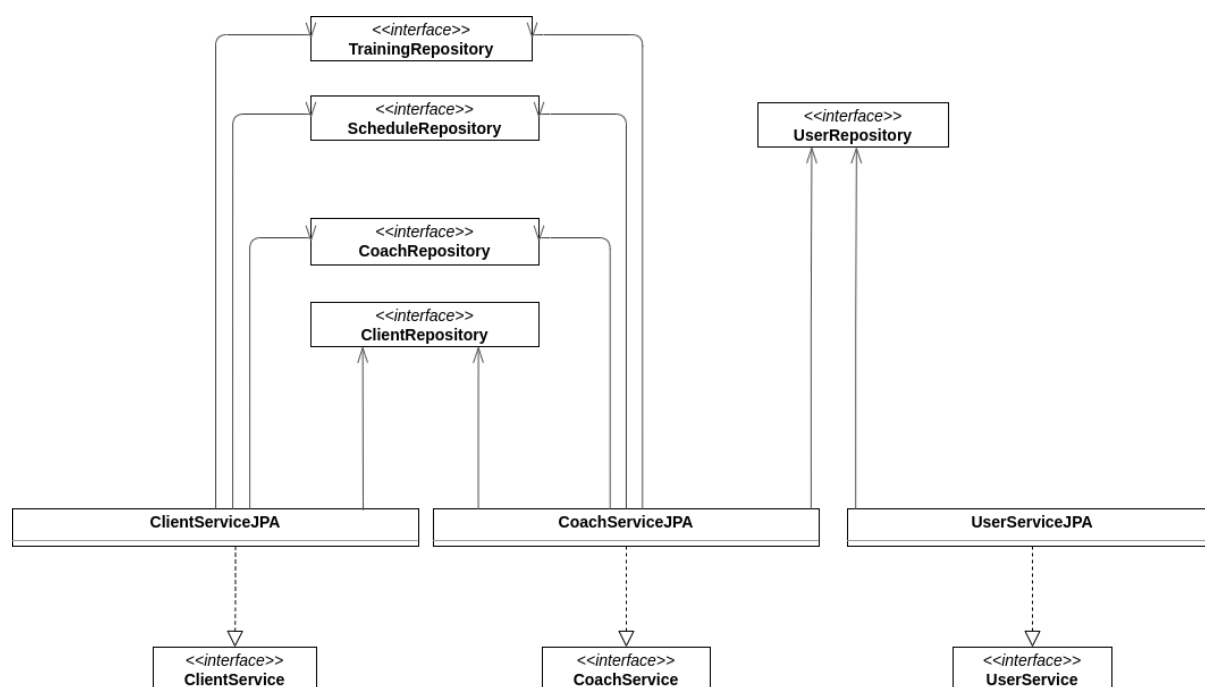
- Modeli - Slika 4.4
- Kontroleri u vezi sa servisima - Slika 4.5
- Servisi u vezi sa repozitorijima - Slika 4.6
- Repozitoriji - Slika 4.7
- Servisi - Slika 4.8

U modelima je važno istaknuti postojanje glavne klase za korisnike zvane "User". Iz nje su izvedene klase klijenta (vježbač) i trenera sa potrebnim atributima specijalizacije.



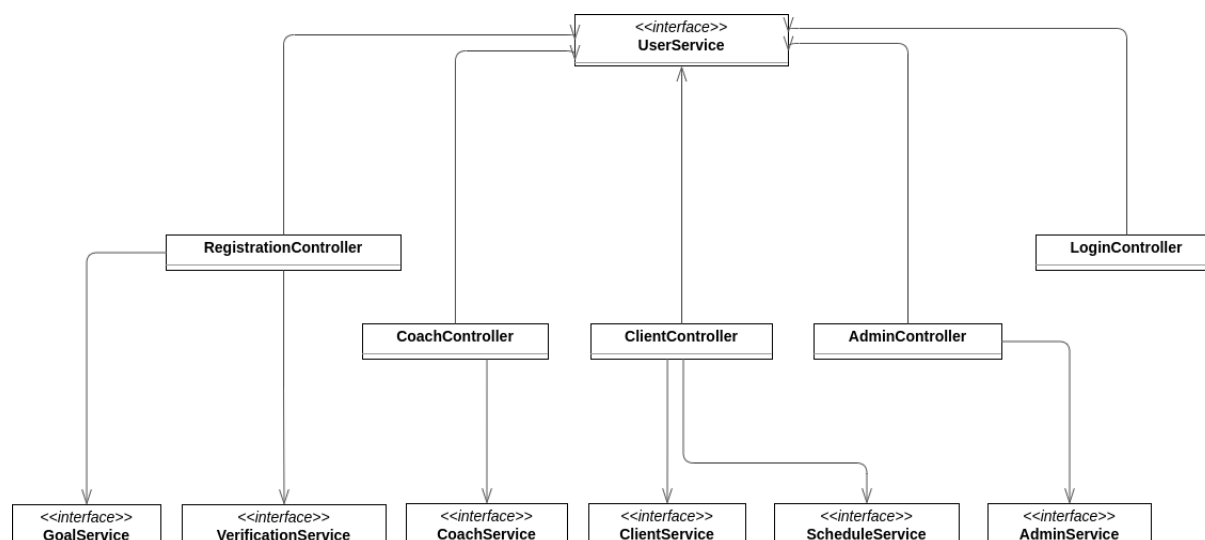
Slika 4.4: Modeli

Servisi ostvaruju funkcionalnosti sustava komunikacijom sa bazom podataka preko repozitorija. Prikazani su neki glavni servisi (i njihove implementacije) koji su zaduženi za opće korisnike (tj. funkcionalnosti koje imaju i klijenti i treneri) te pojedinačni servisi za svaku od navedenih uloga.



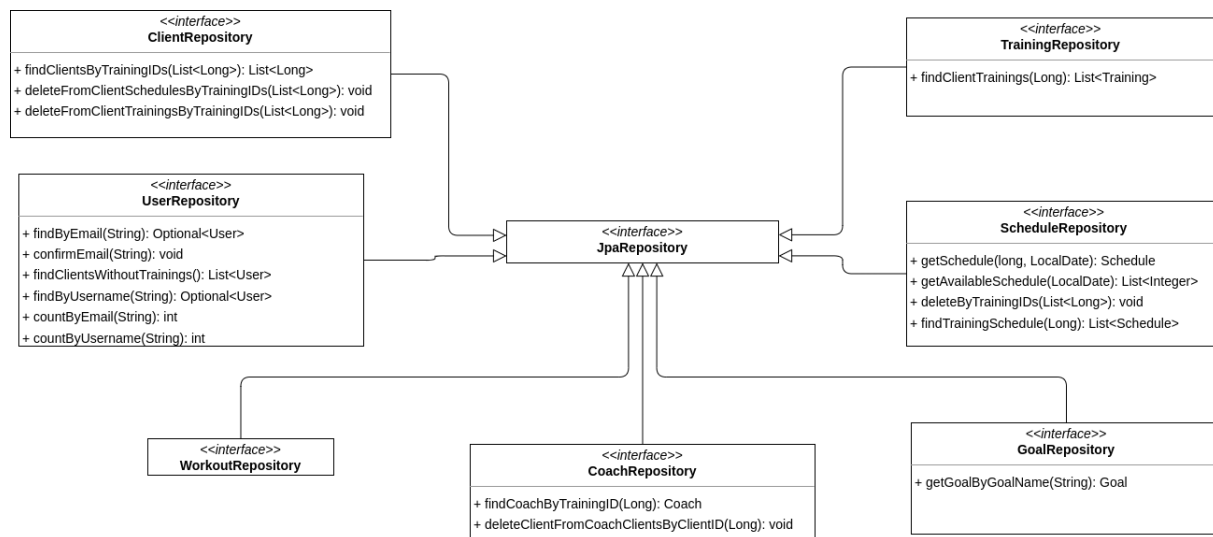
Slika 4.5: Servisi u vezi sa repozitorijima

Kontroleri i servisi čine najvažniji dio sustava jer sadržavaju logiku potrebnu za ostvarivanje zamišljenih funkcionalnosti. Dobivanjem zahtjeva sa *frontend* strane sustava, kontroleri prosleđuju iste odgovarajućim servisima (sa kojima su u vezi). Ovdje su prikazani najvažniji servisi (i odgovarajuće implementacije) i kontroleri za ostvarivanje funkcionalnosti trenera, klijenata i korisnika kao zajedničkog naziva istih.



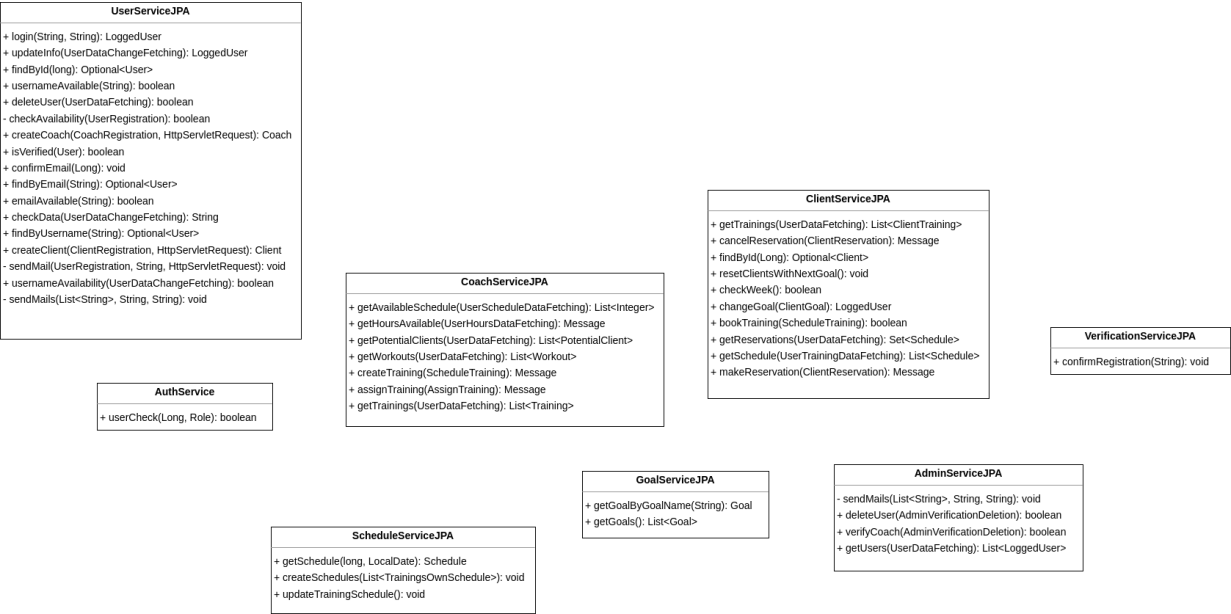
Slika 4.6: Kontroleri u vezi sa servisima

Repozitoriji nasljeđuju osnovni repozitorij unutar Spring Boot radnog okvira (Jpa-Repository). Navedeni omogućuje nekoliko standardnih operacija nad odgovarajućim entitetima (spremanje, brisanje, dohvaćanje...). Ostale funkcionalnosti se ostvaruju komunikaciju sa bazom podataka koristeći prilagođenih query upita (čija je funkcionalnost vidljiva u nazivu na prikazanoj slici).



Slika 4.7: Repoziitoriji

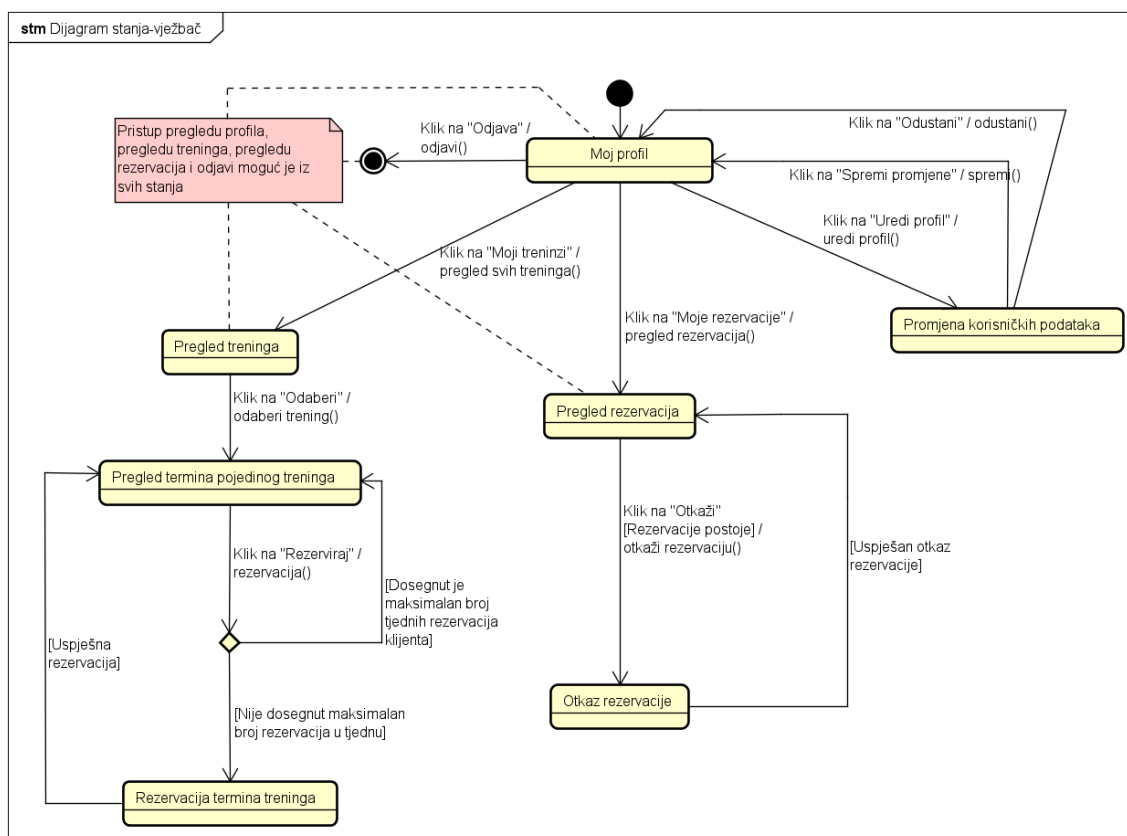
Radi izbjegavanja zbijenosti na dijagramu servisa i repozitorija izdvojili smo razrede servisa i prikazali njihove implementirane funkcije na sljedećem dijagramu.



Slika 4.8: Implementirane funkcije servisa

## 4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekata te prijelaze među stanjima ovisno o događajima. Slika 4.9 prikazuje dijagram stanja za registriranog korisnika, točnije, vježbača. Prijavom u sustav otvara se početna stranica "Moj profil" na kojoj se mogu pregledati osobni podatci i obrisati račun. Početna stranica je, uz pregled treninga, pregled rezervacija i odjavu, vidljiva i dostupna iz svakog stanja. Na stranici pregleda treninga prikazani su svi treninzi dodijeljeni klijentu te postoji mogućnost odabira jednog od njih. Time se dobiva uvid u sve termine odabranog treninga koji se mogu rezervirati. Na stranici "Pregled rezervacija" prikazani su svi trenutno rezervirani termini logiranog vježbača koji se mogu i otkazati. Na stranici promjene korisničkih podataka dostupna je forma u koju se upisuju podatci za promjenu.



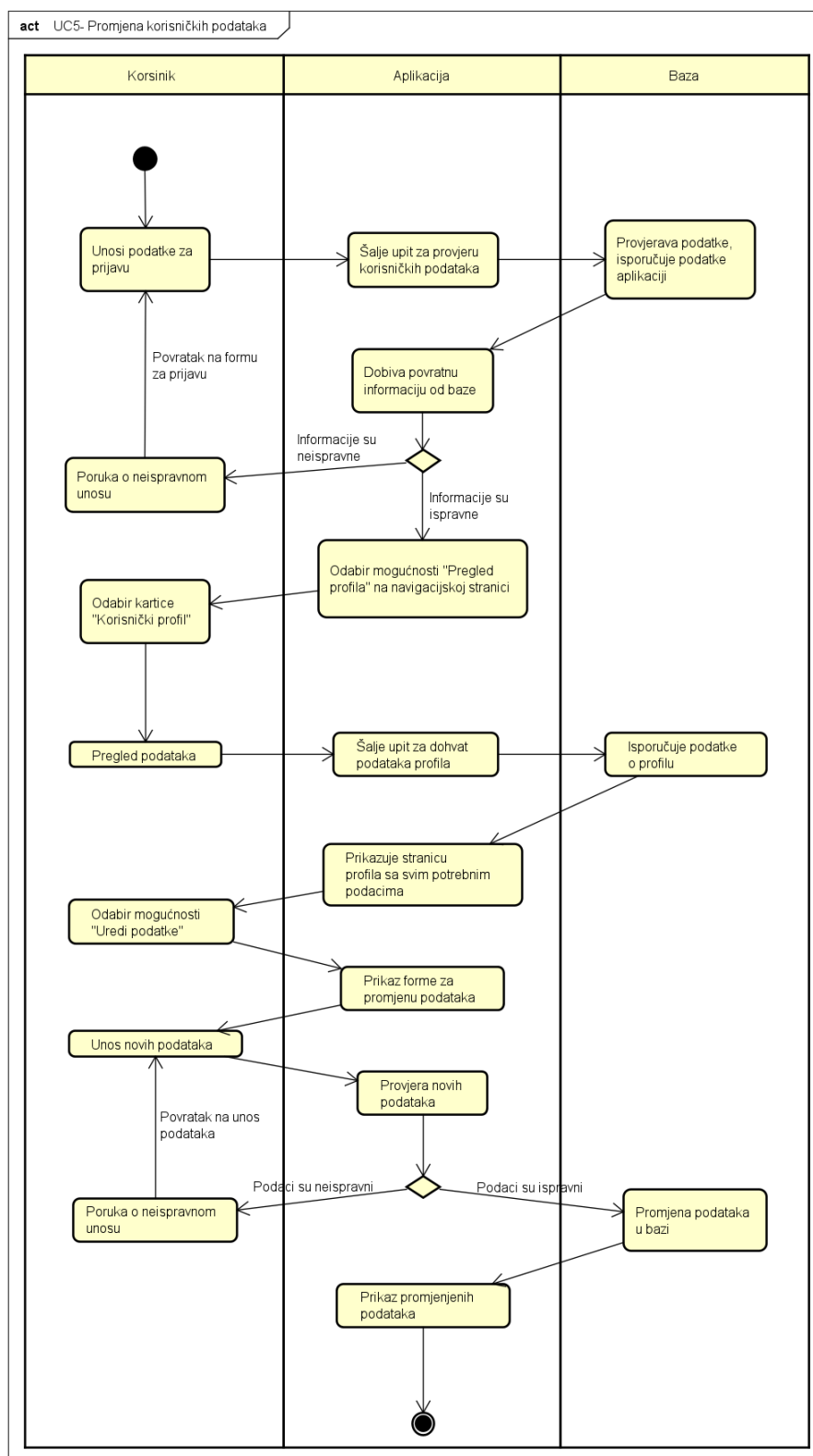
Slika 4.9: Dijagram stanja

## 4.4 Dijagram aktivnosti

Dijagram aktivnosti je ponašajni UML dijagram koji modelira ponašanje nizom akcija, a služi za detaljan prikaz upravljačkog i podatkovnog toga pojedinog segmenta aplikacije (najčešće pojedinih obrazaca uporabe). Dijagramom aktivnosti pogodno je opisivati sinkronizaciju i konkurentnost značajki.

Uz našu aplikaciju implementirali smo dijagram aktivnost za obrazac uporabe 5: „Promjeni korisničke podatke“ kako bismo prikazali podatkovni tok uređivanja postojećih korisničkih profila.





Slika 4.10: Dijagram aktivnosti za obrazac uporabe 5: "Promjena korisničkih podataka"

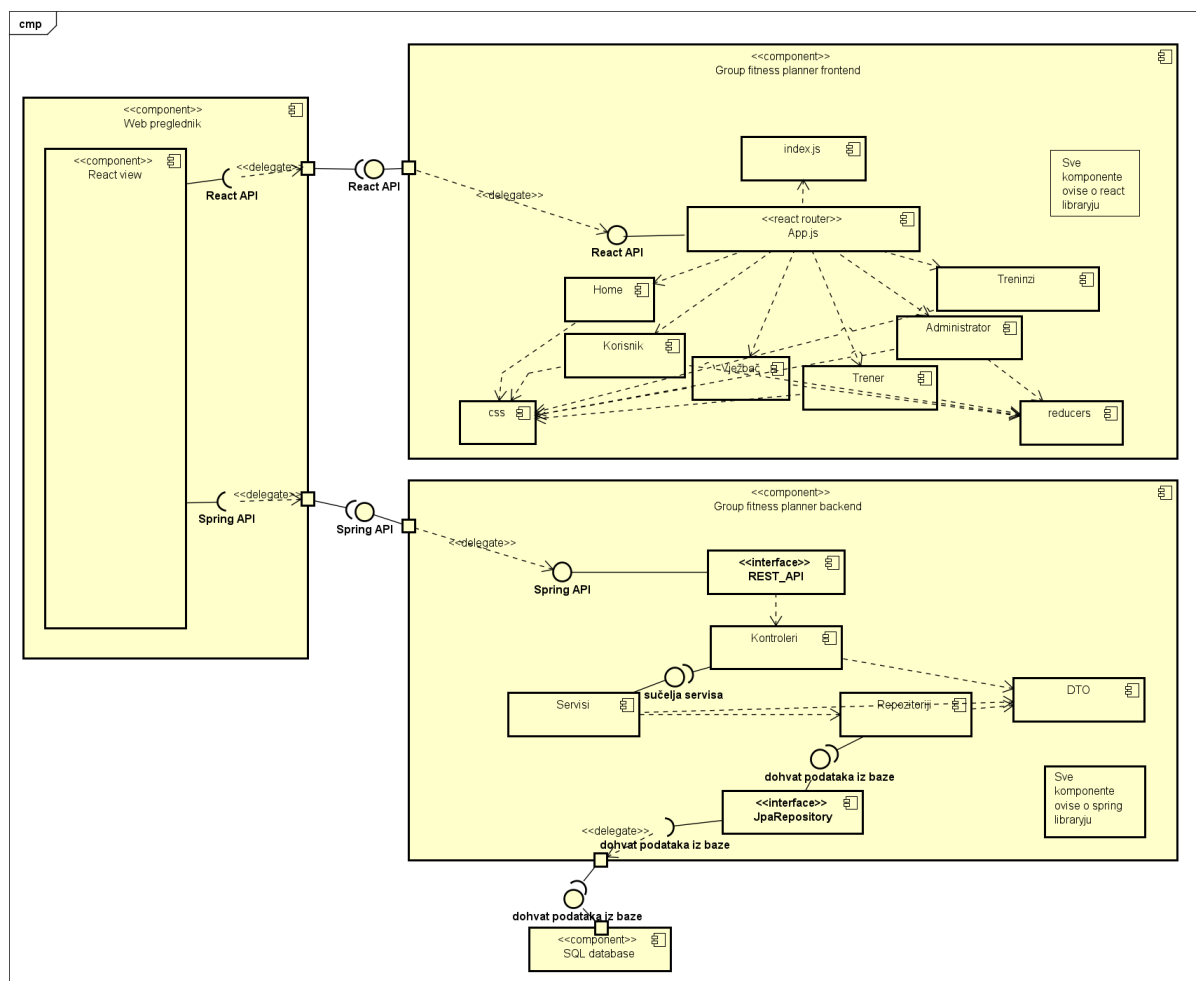
## 4.5 Dijagram komponenti

Dijagram komponenti je strukturni, statički UML dijagram koji vizualizira organizaciju i međuovisnost interne strukture implementacijskih komponenata, te odnos programske potpore prema okolini. Koristan je za stjecanje okvirne ideje o implementaciji sustava, bez ulaženja u prevelike detalje.

Naša aplikacija podijeljena je u četiri glavne komponente: web preglednik (ciljana platforma za pogon aplikacije) koji dohvaća odgovarajuću frontend i backend logiku poštujući internetske komunikacijske protokole, frontend koji dohvaća odgovarajuće HTML, CSS i .js datoteke koje sadrže kod korisničkog sučelja, backend zadužen da dohvat i obradu podataka iz korisničkog sučelja i baze podataka i posreduje komunikaciji navedenih komponenti, te samu SQL bazu podataka.

Komponenti frontend logike pristupa se preko sučelja za dohvat HTML-a, CSS-s i .js datoteka iz web preglednika. Centralni dio frontenda je router datoteka (u našem slučaju App.js) zadužena za usklađivanje rada i dohvaćanje željenih komponenti koje su niže u hijerarhiji frontend logike. Svaka komponenta unutar frontend komponente na grafu sadržava ugnježdene komponente u čiju strukturu radi razumljivosti grafa ne ulazimo, a to označava da svaka komponenta ovisi o React libraryju.

Komponenta backend logike sa web preglednikom povezana je preko sučelja REST (zathjevi GET, POST, DELETE). Zahtjevi se nakon primitka prvo prosljeđuju kontrolerima, subkomponentama backenda zaduženim za prihvrat i isporuku servisa. Servisi su subkomponente backenda koji pomoću svojih sučelja komuniciraju sa kontrolerima i prihvaćaju podatke koji se u njima nalaze, te ih pomoću JPA sučelja prosljeđuju repozitorijima. Repozitoriji omogućuju komunikaciju između servisa i baze podataka. Povezani su sučeljem s SQL bazom podataka. Prijenos podataka između komponenti backend logike ostvaren je „Data transfer object“ klasama pa sve komponente backenda imaju ovisnost prema njima. Svaka komponenta unutar backend komponente na grafu sadržava ugnježdene komponente u čiju strukturu radi razumljivosti grafa ne ulazimo, a to označava da svaka komponenta ovisi o Spring libraryju.



Slika 4.11: Dijagram komponenti aplikacije

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### 5.1.1 Frontend tehnologije

U izradi korisničkog sučelja, točnije *frontend* dijela aplikacije korišten je popularni razvojni okvir React. Korištenjem programskog jezika JavaScript React omogućava jednostavnu izradu interaktivnih sučelja temeljenu na React komponentama. Komponente je moguće strukturirati za ponovno korištenje i dijeljenje kroz cijeli projekt (a i druge projekte) što olakšava samu izradu i održavanje. Za uređivanje koda je korišten Visual Studio Code.

#### 5.1.2 Backend tehnologije

Programski jezik *backend* dijela aplikacije je Java. Korišten je razvojni okvir Spring Boot za čiji je *management* upotrijebljen alat Maven. Tijekom razvoja aplikacije korišten je H2 (in-memory) sustav za upravljanje bazama podataka, dok je za *deploy* verziju korišten sustav PostgreSQL. Razvojna okruženja korištena na *backendu* su IntelliJ IDEA i EclipseIDE.

#### 5.1.3 Deployment

Aplikacija je puštena u pogon na oblaku (engl. cloud) Render. Render omogućava povezivanje sa servisom GitLab te olakšava puštanje proizvoda u pogon.

#### 5.1.4 Dokumentacija

Za dokumentaciju programskog rješenja korišten je online latex editor. Za crtanje UML dijagrama korišten je Astah i IntelliJ IDEA IDE.

#### 5.1.5 Timska komunikacija

Za timsku komunikaciju korišteni su alati Whatsapp, Discord i Microsoft Teams.

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

#### 1. ispitni slučaj - pokušaj dodjele rezervacije za trening koji ne postoji

U ovom testu ispituje se mogućnost dodjele nepostojeće rezervacije, točnije rezervacije termina treninga koji ne postoji u bazi podataka. Očekuje se iznimka (engl. exception) na servisnom sloju klijenta te ispis poruke u skladu s navedenom iznimkom.

```
@Test
void addReservationForTrainingThatDoesntExist(){
    ClientReservation clientReservation = new ClientReservation( userID: 2L, trainingID: 1L, date: "11.01.2023");
    Exception ex = Assert.assertThrows(Exception.class, ()->clientService.makeReservation(clientReservation));
    Assert.assertEquals(ex.getMessage(), actual: "Rezervacija ne postoji!");
}
```

Slika 5.1: 1. test

#### 2. ispitni slučaj - pokušaj dodjele rezervacije za trening koji nije dodijeljen klijentu

U ovom testu ispituje se mogućnost dodjele rezervacije treninga koji nije u treninzima koji su dodijeljeni određenom klijentu, točnije onih koje mu je dodijelio trener. Očekuje se iznimka na servisnom sloju klijenta te ispis poruke u skladu s navedenom iznimkom.

```
@Test
void addReservationForNotAssignedTraining(){
    ClientReservation clientReservation = new ClientReservation( userID: 2L, trainingID: 1L, date: "11.01.2023");
    LocalDate date = LocalDate.parse( LocalDate.parse( "11.01.2023", DateTimeFormatter.ofPattern("dd.MM.yyyy")) );
    when(scheduleRepo.getSchedule( trainingID: 1L, LocalDate.parse( "11.01.2023", DateTimeFormatter.ofPattern("dd.MM.yyyy")) )).thenReturn(new Schedule(new Training(), date));
    Exception ex = Assert.assertThrows(Exception.class, ()->clientService.makeReservation(clientReservation));
    Assert.assertEquals(ex.getMessage(), actual: "Pokušavate dodati rezervaciju za trening koji vam nije dodijeljen!");
}
```

Slika 5.2: 2. test

#### 3. ispitni slučaj - pokušaj izrade treninga bez prethodne verifikacije od admina

U ovom testu ispituje se mogućnost izrade treninga bez prethodne verifikacije trenera od administratora aplikacije. Očekuje se iznimka na servisnom sloju trenera prilikom provjere verifikacije te ispis poruke u skladu s navedenom iznimkom.

```
@Test
void createTrainingWithoutBeingVerified(){
    ScheduleTraining training = new ScheduleTraining( trainingName: "test", trainingRules: "test", spaceAvailable: 25, new String[]{"1","2","3"}, coachID: 7L, new String[][]{{"1","2","3"}});
    Exception ex = Assert.assertThrows(Exception.class, ()->coachServiceForNotVerifiedTest.createTraining(training));
    Assert.assertEquals(ex.getMessage(), actual: "Još niste verificirani od strane admina!");
}
```

Slika 5.3: 3. test

#### 4. ispitni slučaj - pokušaj izrade treninga sa prethodnom verifikacijom od admina

U ovom testu ispituje se mogućnost izrade treninga nakon verifikacije trenera od administratora aplikacije. Očekuje se uspješna izrada treninga i povratna informacija o istome.

```
@Test
void createTrainingAfterBeingVerified() throws Exception {
    Coach coach = coachRepoReal.findById(7L).get();
    coach.setVerified(true);

    when(coachRepoMock.findById(7L)).thenReturn(Optional.of(coach));
    ScheduleTraining scheduleTraining = new ScheduleTraining("trainingName: test", "trainingRules: test", "spaceAvailable: 25", new String[]{"1", "2"}, "coachID: 7L", new String[][]{{"1", "2"}, {"1", "2"}}, "start: 2023-01-01T00:00:00", "end: 2023-01-01T00:00:00");

    when(scheduleRepo.saveAll(Mockito.any(List.class))).thenReturn(1 -> 1.getArguments()[0]);
    when(trainingRepo.save(Mockito.any(Training.class))).thenReturn(1 -> 1.getArguments()[0]);
    Message message = coachServiceForVerifiedTest.createTraining(scheduleTraining);
    Assert.assertEquals(message.getMessage(), "OK");
}
```

Slika 5.4: 4. test

#### 5. ispitni slučaj - pokušaja prijave pogrešnom lozinkom

U ovom testu ispituje se mogućnost prijave klijenta u aplikaciju pogrešnom lozinkom. Pretpostavka je da je klijent prethodno instanciran i spremljen u bazu podataka s korisničkim imenom kao na slici, a lozinkom password123 te je odgovor servisnog sloja zaduženog za provjeru podataka iznimka s prikladnom porukom o istoj.

```
@Test
void wrongPasswordAuth() throws Exception {
    when(userService.login("johnsmith", "password1234")).thenThrow(new Exception("Pogrešno korisničko ime i/ili lozinka!"));
    LoginForm loginForm = new LoginForm("johnsmith", "password1234");
    mockMvc.perform(post("/login/user")
        .contentType(MediaType.APPLICATION_JSON)
        .content(new ObjectMapper().writeValueAsString(loginForm)))
        .andExpect(status().isBadRequest());
}
```

Slika 5.5: 5. test

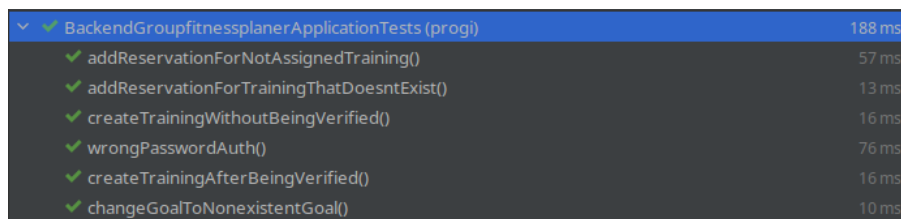
**6. ispitni slučaj - pokušaj promjene cilja u nepostojeći** U ovom testu ispituje se mogućnost promjene cilja klijenta u cilj koji nije definiran unutar baze podataka. Sustav neće dozvoliti takvu promjenu te će se dogoditi iznimka s prikladnom porukom o istoj.

```
@Test
void changeGoalToNonexistentGoal() {
    ClientGoal clientGoal = new ClientGoal("userID: 2L", "goalName: xyz-nepostoji");
    Exception ex = Assert.assertThrows(Exception.class, () -> clientService.changeGoal(clientGoal));
    Assert.assertEquals(ex.getMessage(), "Navedeni cilj ne postoji!");
}
```

Slika 5.6: 6. test

## Prikaz rezultata testova

Testovi komponenti su uspješno provedeni što je vidljivo na priloženoj slici.



✓ BackendGroupfitnessplanerApplicationTests (prog)	188 ms
✓ addReservationForNotAssignedTraining()	57 ms
✓ addReservationForTrainingThatDoesntExist()	13 ms
✓ createTrainingWithoutBeingVerified()	16 ms
✓ wrongPasswordAuth()	76 ms
✓ createTrainingAfterBeingVerified()	16 ms
✓ changeGoalToNonexistentGoal()	10 ms

Slika 5.7: Rezultati testova

## 5.2.2 Ispitivanje sustava

### 1. ispitni slučaj sustava: Verifikacija trenera (administrator)

#### Postupak:

1. Otvaranje početne stranice
2. Odabir veze "Pregled korisnika"
3. Odabir gumba "Potvrdi trenera"
4. Odabir gumba "OK" u obavještajnom prozoru

#### Očekivani rezultati:

1. Prikaz početne stranice
2. Prikaz stranice za pregled korisnika
3. Prikaz obavještajnog prozora s porukom "Jeste li sigurni da želite potvrditi trenera?"
4. Zatvaranje obavještajnog prozora i uklanjanje gumba "Potvrdi trenera"

**Running 'Verifikacija trenera'**

1. open on /home OK
  2. setWindowSize on 1440x778 OK
  3. click on id=13 OK
  4. chooseOkOnNextConfirmation OK
  5. click on css=.table-section:nth-child(9) .green-button OK
  6. webdriverChooseOkOnVisibleConfirmation OK
- 'Verifikacija trenera' completed successfully**

Slika 5.8: 1. Selenium test

**2. ispitni slučaj sustava: Verifikacija trenera (administrator)****Postupak:**

1. Otvaranje početne stranice
2. Odabir veze "Stvori treninga"
3. Unos podataka o treningu i terminima treninga
4. Odabir gumba "Stvori trening"
5. Odabir gumba "OK" u obavještajnom prozoru

**Očekivani rezultati:**

1. Prikaz početne stranice
2. Prikaz stranice za izradu treninga
3. Prikaz obavještajnog prozora s porukom "Uspješno stvoren trening" (nakon 4.)
4. Zatvaranje obavještajnog prozora



**Running 'Izrada treninga'**

1. open on /home OK
  2. setWindowSize on 1440x778 OK
  3. click on id=9 OK
  4. click on name=trainingName OK
  5. type on name=trainingName with value Trening test OK
  6. click on name=trainingRules OK
  7. type on name=trainingRules with value potrebni rekviziti OK
  8. click on name=spaceAvailable OK
  9. type on name=spaceAvailable with value 50 OK
  10. click on css=.nekaj:nth-child(2) > div:nth-child(1) input OK
  11. click on css=.nekaj:nth-child(4) > div:nth-child(2) input OK
  12. click on css=.nekaj:nth-child(6) > div:nth-child(3) input OK
  13. click on css=.nekaj:nth-child(8) > div:nth-child(4) input OK
  14. click on css=.nekaj:nth-child(10) > div:nth-child(5) input OK
  15. click on css=div:nth-child(9) input OK
  16. click on css=div:nth-child(12) > label > input OK
  17. click on css=div:nth-child(19) input OK
  18. click on css=form > div:nth-child(7) input OK
  19. click on css=.button-holder > input OK
- 'Izrada treninga' completed successfully**

Slika 5.9: 2. Selenium test

**3. ispitni slučaj sustava: Dodjela treninga klijentu (trener)****Postupak:**

1. Otvaranje početne stranice
2. Odabir veze "Moji klijenti"
3. Odabir gumba "Dodijeli trening"
4. Odabir gumba "Odaberi trening" (moguć višestruki odabir)
5. Unos fonda sati klijenta
6. Odabir gumba "Potvrdi odabir"
7. Odabir gumba "OK" u obavještajnom prozoru

**Očekivani rezultati:**

1. Prikaz početne stranice
2. Prikaz stranice za pregled klijenata bez treninga
3. Prikaz stranice za pregled mogućih treninga za dodjelu
4. Prikaz obavještajnog prozora s porukama "OK" (nakon 6.)
5. Zatvaranje obavještajnog prozora

```
Running 'Dodavanje treninga klijentu'  
1. open on /home OK  
2. setWindowSize on 1440x778 OK  
3. click on id=11 OK  
4. click on css=.table-section:nth-child(4) .table-button OK  
5. click on id=1 OK  
6. click on name=dodjelafonda OK  
7. type on name=dodjelafonda with value 25 OK  
8. click on css=.green-button:nth-child(2) OK  
'Dodavanje treninga klijentu' completed successfully
```

Slika 5.10: 3. Selenium test

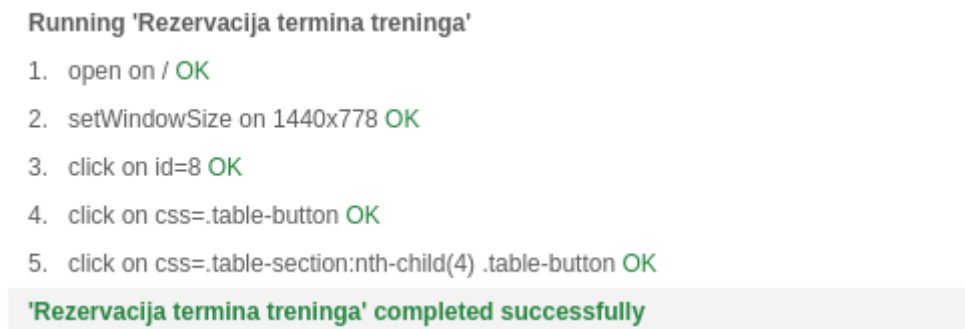
**4. ispitni slučaj sustava: Rezervacija termina treninga (klijent)****Postupak:**

1. Otvaranje početne stranice
2. Odabir veze "Moji treninzi"
3. Odabir gumba "Odaberi"
4. Odabir gumba "Rezerviraj"
5. Odabir gumba "OK" u obavještajnom prozoru

**Očekivani rezultati:**

1. Prikaz početne stranice
2. Prikaz stranice za pregled treninga

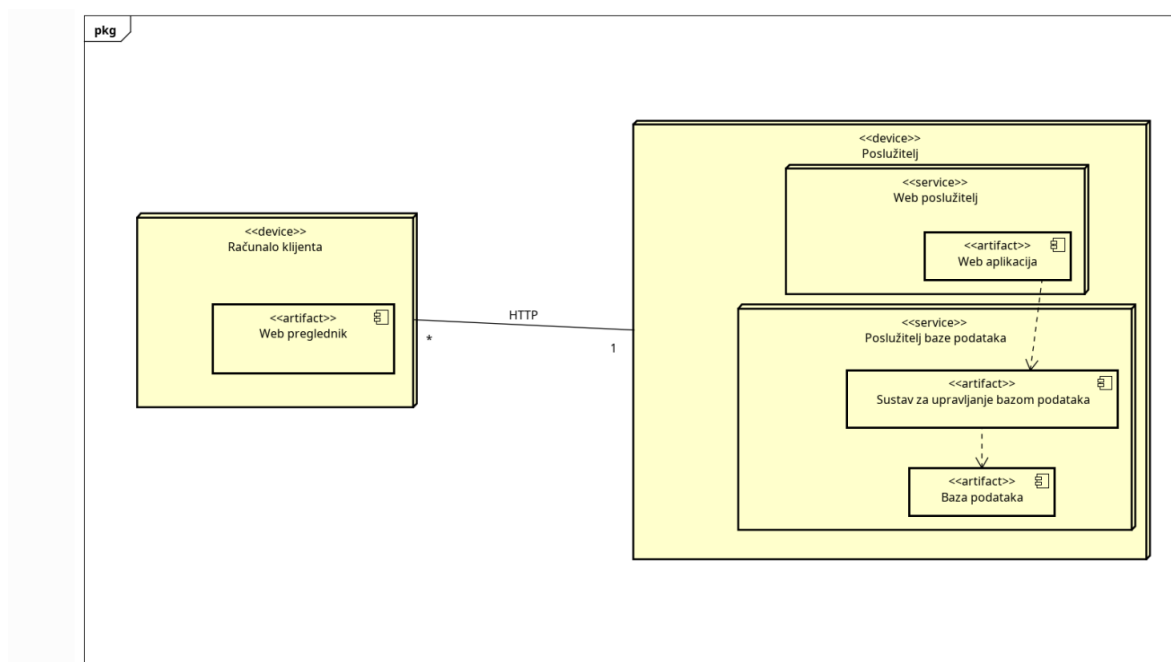
3. Prikaz stranice za pregled termina treninga
4. Prikaz obavještajnog prozora s porukom "Uspješno ste rezervirali trening!"
5. Zatvaranje obavještajnog prozora i zatamnjivanje gumba "Rezerviraj"



Slika 5.11: 4. Selenium test

## 5.3 Dijagram razmještaja

Specifikacijski dijagram razmještaja prikazuje raspodjelu komponenata sustava po lokacijama te odnos sklopovlja i programa. Uočavamo komunikaciju između računala klijenta putem web preglednika i udaljenog poslužitelja (koji sadrži web aplikaciju sa svojim *backend* i *frontend* elementima te poslužitelja baze podataka). Navedena arhitektura sustava naziva se *klijent-poslužitelj* te je u njoj izvedbi za komunikaciju između dviju navedenih strana korišten protokol HTTP.



Slika 5.12: Specifikacijski dijagram razmještaja

## 5.4 Upute za puštanje u pogon

### 5.4.1 Upute za puštanje u pogon na javnom poslužitelju

Aplikacija je puštena u pogon na javnom poslužitelju „Render“. On pruža mogućnost posluživanja web servisa, kao i PostgreSQL baze podataka.

Puštanje u pogon sadržava korake:

- kreiranje baze podataka
- puštanje backenda u pogon na javnom poslužitelju
- puštanje frontenda u pogon na javnom poslužitelju

Bazu podataka kreirali smo i njome upravljamo direktno iz servisa Render, jer on omogućuje tu opciju. Baza sadržava podatke potrebne za njeno povezivanje sa web servisima.

The image shows the 'Connections' section of the Render dashboard. It contains a list of database connections with the following fields:

- Hostname: dpg-cdpm1npgp3jr9p59f8b0-a
- Port: 5432
- Database: groupfitnessplanerdb
- Username: devups
- Password: [Redacted]
- Internal Database URL: [Redacted]
- External Database URL: [Redacted]
- PSQL Command: [Redacted]

Slika 5.13: Render sučelje za upravljanje bazom podataka

Kako bismo uspješno preneli backend aplikacije na web poslužitelj, bilo je potrebno dodati Dockerfile koji upravlja i posreduje komunikaciji backenda i Mavena, alata pomoću kojeg Java projekti postaju sinkronizirani sa Apache HTTP serverima. Također, bilo je potrebno povezati kreiranu bazu podataka sa backendom. Konfiguracija environment varijabli u application.properties datoteci svojstva nužno je da bismo mogli postaviti adresu, korisničko ime i lozinku baze podataka na produkciji. Također, opcija kreiranja web servisa u Renderu zahtjeva dodavanje potrebnih varijabli okruženja koje treba kopirati iz danih connections-a baze podataka, postavljanje putanje za Dockerfile.

```
spring.sql.init.mode=always
spring.jpa.hibernate.ddl-auto=update
spring.sql.init.platform=postgres
spring.datasource.password=
spring.datasource.username=devups
spring.datasource.url=
spring.datasource.driverClassName=org.postgresql.Driver
```

Slika 5.14: Konfiguracija postavki za povezivanje backenda sa bazom podataka

Kako bismo uspješno preneli frontend, bilo je potrebno u package.json dodati zavisnosti potrebne za puštanje u pogon, primarno proxy-middleware, dotnev, express. Također, bilo je potrebno dodati setupProxy.js koji služi kao server za lokalno razvijanje, app.js u kojem se nalazi express server za produkcijski proxy, kao i izmijeniti package.json dodavanjem "start-prod": "node app.js" skripte, koja navigira front da komunicira preko app.js filea. Opcija kreiranja web servisa u Renderu zahtjevala je postavljanje build komande i start komande, dodavanje varijabli okruženja koje pokazuju na adresu backenda koji je prethodno pušten u pogon na javom poslužitelju kako bi javni frontend bio povezan za backendom (koji je povezan sa bazom podataka).

URL na javnu verziju aplikacije: <https://group-fitness-planer-q3fc.onrender.com>

## 5.4.2 Upute za puštanje u pogon lokalno

Aplikaciju također možemo u pogon puštati lokalno, za potrebe razvoja. Taj postupak podijeljen je u dva glavna koraka:

- puštanje backenda lokalno
- puštanje frontenda lokalno

Backend se pokreće iz IDE-a (IntelliJ, Eclipse...) pokretanjem datoteke: GroupFitnessPlannerApplication.java (putanja glasi: /group-fitness-planer/Backend/src/main/java/progi/GroupFitnessPlannerApplication.java), koja je autogenerirani dokument svakog Spring boot projekta. Ono što omogućuje lokalno pokretanje su svojstva aplikacije u application.properties koja su postavljena kao na idućoj slici:

```
spring.sql.init.mode=always
spring.jpa.defer-datasource-initialization=true
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa

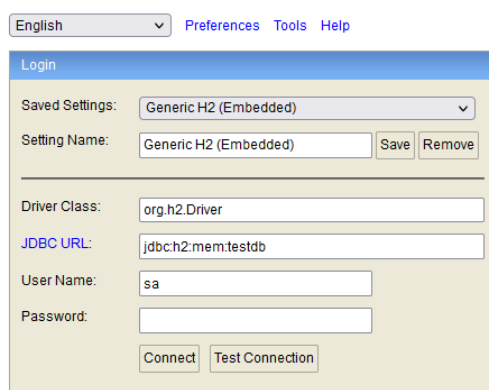
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true

spring.security.enabled=true

security.rule.coach = /coach/**, COACH
```

Slika 5.15: Konfiguracija postavki za pokretanje backenda lokalno

Za potrebe razvoja koristili smo se in-memory h2 bazom podataka. Aplikacija se pokreće na localhost:8080 portu. Ukoliko želimo pregleđavati stanje u bazi, koristimo URL: localhost:8080/h2-console.



Slika 5.16: Konzola za in-memory bazu

Pokretanje frontenda obavlja se iz naredbenog retka. Najprije se pozicioniramo u direktorij sa package.json fileom (/group-fitness-planer/Frontend/groupfitnessui) i u njemu izvedemo naredbe:

### **npm install**

koja instalira sve node module potrebne za pokretanje aplikacije, te

### **npm start**

koja pokreće aplikaciju na localhost:8080 (namješteno tako da se front i back pokreću na istom portu kako bi mogli komunicirati). Stranica se ponovno učitava svakom spremljenom promjenom u kodu.

## 6. Zaključak i budući rad

Primarni cilj ovog projekta bio je razviti aplikaciju koja omogućuje polaznicima treninga odabir treninga po terminu i tipu, koje korisnik može pohađati u skladu sa svojim slobodnim vremenom. Na taj način termini vježbanja prilagođavaju se slobodnom vremenu vježbača (polaznika), te su vježbači puno zadovoljniji jer mogu uložiti više pažnje vježbanju, i dobiti maksimalnu korist i dobrobit od usluga koje teretana, koja se koristi aplikacijom, pruža.

Iz projektnog zadatka bilo je potrebno izlučiti funkcionalne i nefunkcionalne zahtjeve, konceptualno osmisliti, dokumentirati, a zatim i implementirati osmišljeno. Mentori ovog projekta ujedno su „glumili“ i klijente, pa smo komunikacijom s njima definirali koje funkcionalnosti žele vidjeti u aplikaciji, što ona mora sadržavati.

Provedba projekta bila je podijeljena u dva ciklusa. U prvom ciklusu oformili smo projektni tim, uspostavili kanale komunikacije, upoznali se sa zadatkom i krenuli u osmišljavanje rješenja zadatka. Inicijalne funkcionalnosti koje je naša aplikacija imala bili su prijava i registracija korisnika u sustav. Redovitim sastancima i komunikacijom preko društvenih mreža podijelili smo se u timove, zadužili se za pojedine zadatke i funkcionalnosti, te jasno definirali što koja funkcionalnost radi kako bismo svi imali jednaku ideju i razumjevanje zadaka.

U drugom ciklusu veći naglasak je bio na implementaciji same aplikacije. Dok smo se svi u prvom ciklusu „upoznavali“ i po prvi put susreli sa programskim alatima koje trebamo koristiti, u drugom ciklusu naglasak je bio na individualnom radu i programskoj implementaciji. Sve funkcionalnosti opisane obrascima uporabe su implementiranje, a one koje nisu opisane, nisu niti implementirane.

Korist ovog projekta bila je prvenstveno edukacijska. Jako puno smo naučili o tehnologijama kojima smo implementirali aplikaciju, ali i o timskom radu, komunikaciji i dobili smo uvid od svojih mentora kako je raditi takve projekte u pravim firmama za stvarne klijente. Iako smo se trudili implementirati sve željene funk-



cionalnosti, svjesni smo kako je potrebno još mnogo rada i vještine kako bi ona zaista bila na nivou na kakav su korisnici danas navikli. Trenutna verzija simbolizira prototip kojeg bilo koji član tima jednog dana može unaprjeđivati i usavršavati ukoliko bude imao interesa. Smatramo da je aplikacija korisna i primjenjiva svakom od nas jer se, kao i većina ljudi, brinemo o svome zdravlju, nastojimo baviti fizičkom aktivnosti i povremeno odlaziti u teretane, te se lako vidimo kako aplikaciju u budućnosti zaista i upotrebljavamo.

# Popis literature

## *Kontinuirano osvježavanje*

*Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.*

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

# Indeks slika i dijagrama

2.1	Exercise . . . . .	9
2.2	Exercise . . . . .	10
2.3	Exercise . . . . .	10
2.4	Exercise . . . . .	11
2.5	Group fitness planner . . . . .	12
2.6	Rezultati ankete . . . . .	13
2.7	Rezultati ankete . . . . .	13
3.1	Dijagram obrasca uporabe sa funkcionalnostima administratora . .	26
3.2	Dijagram obrasca uporabe sa funkcionalnostima trenera . . . . .	26
3.3	Dijagram obrasca uporabe sa funkcionalnostima vježbača . . . . .	27
3.4	Sekvencijski dijagram registracije . . . . .	29
3.5	Sekvencijski dijagram izrade treninga . . . . .	30
3.6	Sekvencijski dijagram dodjele treninga klijentima . . . . .	31
3.7	Sekvencijski dijagram rezervacije dodijeljenog termina . . . . .	32
4.1	Opći MVC obrazac . . . . .	35
4.2	Spring MVC obrazac . . . . .	36
4.3	Dijagram tablica unutar baze podataka . . . . .	41
4.4	Modeli . . . . .	42
4.5	Servisi u vezi sa repozitorijima . . . . .	43
4.6	Kontroleri u vezi sa servisima . . . . .	43
4.7	Repozitoriji . . . . .	44
4.8	Implementirane funkcije servisa . . . . .	45
4.9	Dijagram stanja . . . . .	46
4.10	Dijagram aktivnosti za obrazac uporabe 5: "Promjena korisničkih podataka" . . . . .	48
4.11	Dijagram komponenti aplikacije . . . . .	50
5.1	1. test . . . . .	52
5.2	2. test . . . . .	52

5.3	3. test . . . . .	52
5.4	4. test . . . . .	53
5.5	5. test . . . . .	53
5.6	6. test . . . . .	53
5.7	Rezultati testova . . . . .	54
5.8	1. Selenium test . . . . .	55
5.9	2. Selenium test . . . . .	56
5.10	3. Selenium test . . . . .	57
5.11	4. Selenium test . . . . .	58
5.12	Specifikacijski dijagram razmještaja . . . . .	59
5.13	Render sučelje za upravljanje bazom podataka . . . . .	60
5.14	Konfiguracija postavki za povezivanje backenda sa bazom podataka	61
5.15	Konfiguracija postavki za pokretanje backenda lokalno . . . . .	62
5.16	Konzola za in-memory bazu . . . . .	62

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 24. listopada 2022.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: uvodni sastanak
  - upoznavanje
  - određivanje kanala komunikacije
  - odabir tehnologija
  - raspodjela zaduženja unutar tima

### 2. sastanak

- Datum: 3. studenoga 2022.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: daljnji rad na projektu
  - raspravljanje oko dodjeljenog zadatka
  - definiranje funkcionalnih i nefunkcionalnih zahtjeva
  - definiranje ostalih zahtjeva
  - raspodjela pisanja dokumentacije

### 3. sastanak

- Datum: 10. studenoga 2022.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: daljnji rad na projektu
  - kontrolna točka, prezentiranje dosad napravljenog
  - detaljna razrada svih Use caseova sustava
  - razmjena informacija o tehnologijama
  - raspodjela oko predstojećih zadataka

## 4. sastanak

- Datum: 17. studenoga 2022.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: revizija prilikom prvog kolokviranja
  - prezentiranje dosad napravljenog
  - testiranje postojećih funkcionalnosti
  - diskusija o predaji prve inačice aplikacije

## 5. sastanak

- Datum: 7. prosinca 2022.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: nastavak rada na projektu, ulazak u drugu fazu
  - definiranje novih zadataka
  - podjela oko novih zadataka
  - uvid u dosad napravljeno

## 6. sastanak

- Datum: 17. prosinca 2022.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: nastavak rada na projektu
  - kontrolna točka
  - uvid u dosad napravljeno

## 7. sastanak

- Datum: 3. siječnja 2023.
- Prisustvovali: Bruna Kaštela, Tomislav Kožul, Petar Lovrić, Damir Numić-Meša, Rujana Perić, Petra Renić, Nika Šljubura
- Teme sastanka: nastavak rada na projektu
  - kontrolna točka
  - uvid u dosad napravljeno
  - ulazak u završnu fazu projekta

## Tablica aktivnosti

	Tomislav Kožul	Damir Numić-Meša	Bruna Kaštela	Petar Lovrić	Nika Šljubura	Petra Renić	Rujana Perić
Upravljanje projektom		3					
Opis projektnog zadatka	4						4
Funkcionalni zahtjevi					2		
Opis pojedinih obrazaca		6	6	3			
Dijagram obrazaca						3	
Sekvencijski dijagrami							4
Opis ostalih zahtjeva	1						
Arhitektura i dizajn sustava		5					
Baza podataka		4	3				
Dijagram razreda		2					
Dijagram stanja			1				
Dijagram aktivnosti							2
Dijagram komponenti							2
Korištene tehnologije i alati		4	5				
Ispitivanje programskog rješenja	2	8	2	2	2	2	2
Dijagram razmještaja		1					
Upute za puštanje u pogon		3					1
Dnevnik sastajanja							2
Zaključak i budući rad							2

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

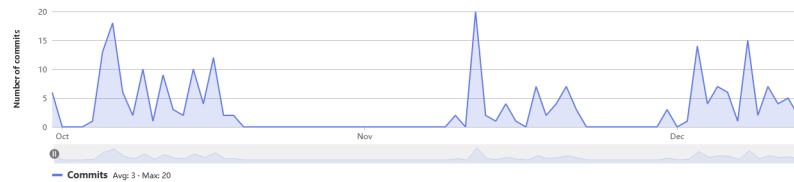
	Tomislav Kožul	Damir Numić-Meša	Bruna Kaštela	Petar Lovrić	Nika Šljubura	Petra Renić	Rujana Perić
Popis literature							1
<i>front end</i>	70			56	68	60	
<i>izrada baze podataka</i>		5	5				
<i>spajanje s bazom podataka</i>		1					2
<i>back end</i>		72	60				24



## Dijagrami pregleda promjena

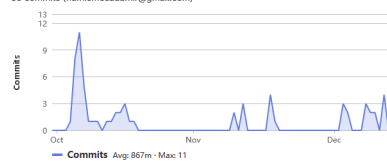
### Commits to develop

Excluding merge commits. Limited to 6,000 commits.



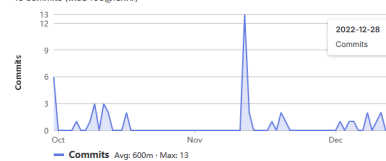
### Damir

65 commits (numicmesadamir@gmail.com)



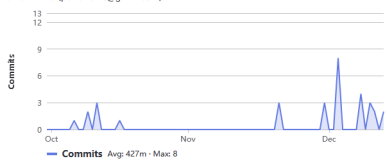
### TomislavKozul

45 commits (tk53406@fer.hr)



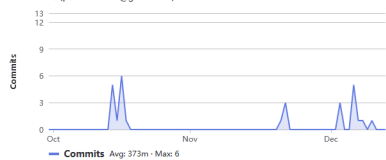
### Petra Renić

32 commits (petra.renic@gmail.com)



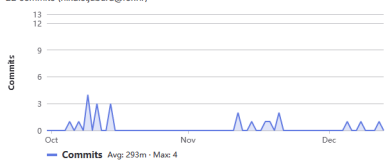
### Doktor-byte

28 commits (petar15.lovric@gmail.com)



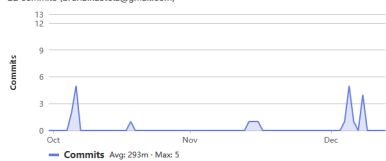
### nikas0808

22 commits (nika.sjubura@fer.hr)



### Bruna Kastela

22 commits (bruna.kastela@gmail.com)



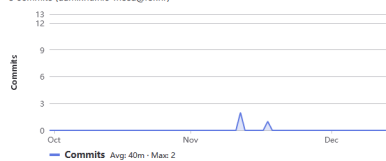
### rperic

7 commits (rujana.peric@fer.hr)



### Damir Numić-Meša

3 commits (damir.numic-mesa@fer.hr)



### dsmoljan

1 commit (dorian.smoljan@croz.net)



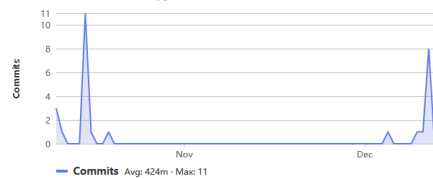
## Commits to devdocs

Excluding merge commits. Limited to 6,000 commits.



## Damir

28 commits (numicmesadamir@gmail.com)



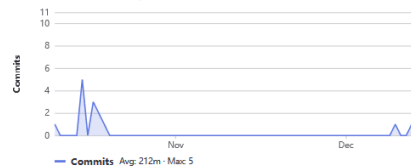
## Petra Renić

19 commits (petra.renic@gmail.com)



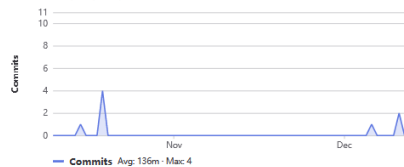
## Bruna Kastela

14 commits (bruna.kastela@gmail.com)



## rperic

9 commits (rujana.peric@fer.hr)



## TomislavKozul

8 commits (tk53406@fer.hr)



## Damir Numić-Meša

1 commit (damir.numic-mesa@fer.hr)



## nikas0808

1 commit (nika.sljubura@fer.hr)

