

Programsko inženjerstvo

Ak. god. 2022./2023.

Group Fitness Planner

Dokumentacija, Rev. 2

Grupa: *BourbonTech*

Voditelj: *Luka Vukelić*

Datum predaje: *13.1.2023.*

Nastavnik: *Ivana Lulić*

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	3
3 Specifikacija programske potpore	6
3.1 Funkcionalni zahtjevi	6
3.1.1 Obrasci uporabe	7
3.1.2 Sekvencijski dijagrami	14
3.2 Ostali zahtjevi	18
4 Arhitektura i dizajn sustava	19
4.1 Baza podataka	19
4.1.1 Opis tablica	20
4.1.2 Dijagram baze podataka	22
4.2 Dijagram razreda	22
4.3 Dijagram stanja	26
4.4 Dijagram aktivnosti	27
4.5 Dijagram komponenti	28
5 Implementacija i korisničko sučelje	29
5.1 Korištene tehnologije i alati	29
5.2 Ispitivanje programskog rješenja	30
5.2.1 Ispitivanje komponenti	30
5.2.2 Ispitivanje sustava	33
5.3 Dijagram razmještaja	40
5.4 Upute za puštanje u pogon	41
6 Zaključak i budući rad	43
Popis literature	44
Indeks slika i dijagrama	45
Dodatak: Prikaz aktivnosti grupe	46

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Luka Vukelić	2.11.2022.
0.2	Dodan opis projektnog zadatka	Jelena Kulišić	4.11.2022.
0.3	Dodana specifikacija programske potpore	Luka Vukelić	6.11.2022.
0.4	Nadograđen opis projektnog zadatka	Jelena Kulišić	11.11.2022.
0.5	Nadograđena specifikacija programske potpore	Jelena Kulišić	12.11.2022.
0.6	Dodan opis baze podataka i opis tablica baze podataka	Jelena Kulišić	14.11.2022.
0.7	Dodani ostali zahtjevi	Jelena Kulišić	14.11.2022.
0.8	Dodana arhitektura sustava	Jelena Kulišić	14.11.2022.
1.1	Dodan korištene tehnologije i alati	Tin Pavletić	3.1.2023.
1.2	Dodan dijagram stanja	Jelena Kulišić	5.1.2023.
1.3	Dodan dijagram aktivnosti	Jelena Kulišić	5.1.2023.
1.4	Dodan dijagram razmještaja	Tin Pavletić	6.1.2023.
1.5	Dodan dijagram komponenti	Jelena Kulišić	10.1.2023.
1.6	Dodan zaključak	Jelena Kulišić	10.1.2023.

2. Opis projektnog zadatka

Zdravlje i fizička aktivnost prioritet su svakoj osobi. U današnje vrijeme kada većina živi u urbanim, zaposlenim životima, teško je u vlastiti raspored uključiti i tjelovježbu. Svakodnevne obaveze razlikuju se iz dana u dan te se ponekad čini nemogućim pohađati striktno određene termine treninga. Rješenje je tih problema web aplikacija "Group Fitness Planner".

Cilj je ovoga projekta razviti programsku podršku za navedenu aplikaciju. Ona će korisniku omogućiti da vrijeme treninga prilagođava svom slobodnom vremenu u skladu s osobnim planom vježbanja.

Prilikom pokretanja aplikacije prikazuje se naslovnica web aplikacije. Na njoj, prikazane su informacije o aplikaciji, u gornjem lijevom kutu nalazi se logo aplikacije, a u gornjem desnom kutu nalazi se poveznica za prijavu korisnika. Klikom na poveznicu, prikazuje se okvir u koji se upisuju podaci za prijavu korisnika, to jest korisničko ime i lozinka. U slučaju da korisnik nema račun, na dnu okvira također se nalazi poveznica koja vodi do stranice za registraciju korisnika. Za kreiranje novoga računa potrebni su idući podaci:

- ime
- prezime
- korisničko ime
- e-mail adresa
- lozinka
- cilj korisnika

Registracijom u sustav korisniku se dodjeljuju prava klijenta. Registrirani korisnik može pregledati osobne podatke, mijenjati ih te izbrisati korisnički račun.

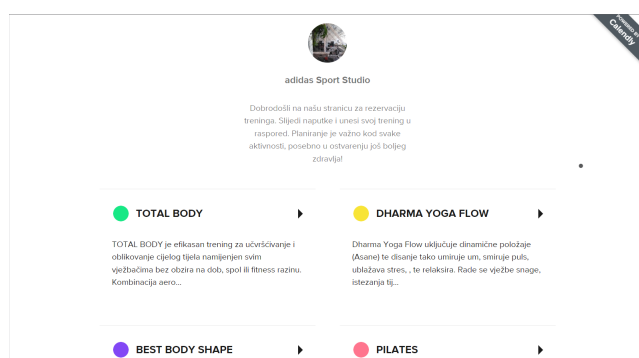
Korisnik treninga prilikom registracije odabire ciljeve koje želi postići vježbanjem. Sukladno odabranim ciljevima, korisniku se dodjeljuju vrste treninga koji dovode do ostvarenja tih ciljeva. Registracijom ili prijavom, korisniku se prikazuje kalendar tekućeg mjeseca i poruka „Molimo Vas pričekajte da Vam trener dodijeli vježbe.“ Nakon što su vježbe dodijeljene, korisniku su prikazani datumi u kalendaru u kojima se odvijaju treninzi koji sadržavaju te vježbe. Odabirom određenog termina, prikazuje se naziv treninga s opisanim vježbama koje su uključene u trening i imenom trenera. Korisnik odabire vrijeme koje mu odgovara te klikom na gumb rezervira trening. Odabrani cilj moguće je mijenjati početkom svakoga mjeseca. Ako korisnik želi na početku novog mjeseca promijeniti cilj odlazi na stranicu s informacijama o svom korisničkom profilu. Na toj stranici nalaze se opći podaci o korisniku kao što su ime, prezime, korisničko ime, email adresa, fond preostalih sati i odabrani ciljevi. Tamo korisnik može promijeniti jedan ili više ciljeva ili nadodati nove ciljeve. Promjenom ciljeva, mijenjaju s i preporučene vježbe, a time i izbor treninga.

Trener ima pristup profilima registriranih korisnika te u skladu s ciljevima koje je korisnik odabrao dodjeljuje korisniku vježbe. Prijavom u aplikaciju prikazuje mu se popis imena i ciljeva svih registriranih korisnika. Pored imena onih korisnika koji nemaju dodijeljene vrste vježbi, prikazuje

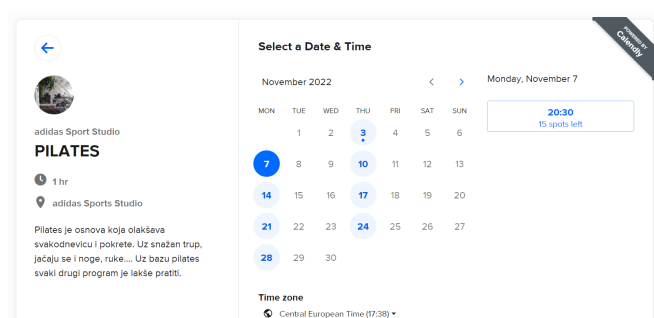
mu se gumb "Dodijeli vježbe". Također, trener na početku svakog mjeseca, u aplikaciju postavlja nove termine treninga te maksimalan kapacitet korisnika određenog termina. Osim toga, treneri za svaki trening odlučuju koje su vrste vježbi sadržane u treningu te se prema tome korisnicima dodjeljuju termini treninga koji sadrže vježbe preporučene za njih. Dodatno, treneri određuju pravila po kojima korisnici smiju rezervirati treninge. Ta su pravila da korisnik može rezervirati maksimalno pet treninga tjedno, smije rezervirati samo jedan trening tjedno intenzivnog karaktera te maksimalno dva treninga dnevno.

Administrator sustava ima najviše ovlasti. On ima pristup bazi podataka s popisom registriranih korisnika i njihovim podacima. Također, administrator trenerima dodjeljuje njihov status. Uz to, ažurira sve podatke u aplikaciji.

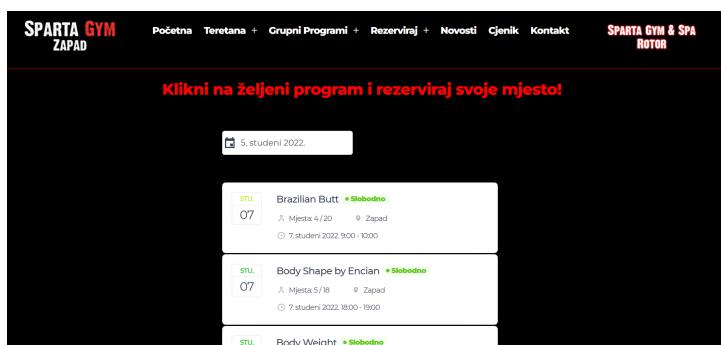
Web aplikacije slične našoj već postoje na tržištu. Neke od njih su web aplikacija za rezervaciju treninga u teretani "Adidas Sports Studio" (prikaz na slici 2.1 i 2.2) i web aplikacija za teretanu "Sparta Gym" (slika 2.3). Ono što našu aplikaciju razlikuje od već prisutnih, personalizirani je pristup korisniku. Na navedenim aplikacijama, svim korisnicima prikazuje se cjelokupni raspored treninga koji se odvijaju u navedenim teretanama. S druge strane, u našoj aplikaciji ovisno o cilju korisnika, trener dodjeljuje vježbe te se prema tim vježbama prikazuju određeni termini treninga. To dodatno motivira korisnike treninga jer znaju kako su baš ti trenizi namijenjeni njima te će ih dovesti do željenih rezultata. Također, korisnici prije dolaze do napretka, a to ih dodatno motivira za nastavak.



Slika 2.1: "Adidas Sports Studio" web aplikacija



Slika 2.2: "Adidas Sports Studio" web aplikacija



Slika 2.3: "Sparta Gym" web aplikacija

Upravo zbog tog personaliziranog pristupa, smatramo da će aplikacija biti korisna svima onima koji nemaju vremena istražiti i informirati se o tome koje vrste vježbi bi najviše odgovarale njihovim potrebama.

Također, postoji puno prostora za nadogradnju aplikacije. jedan primjer je povezivanje nekoliko sportskih objekata u aplikaciji. Veliki broj tvrtki svojim zaposlenicima nudi korištenje "Multisport" kartice. To je kartica kojom je omogućen pristup velikom broju sportskih objekata u cijeloj Hrvatskoj. Tako bi aplikacija mogla nuditi termine treninga u različitim teretanama i sportskim prostorima. Time bi korisnici koji često putuju mogli rezervirati termine i u drugim gradovima te tako biti još redovitiji u pohađanju treninga. Drugi je primjer proširenja aplikacije da se uz teretane, uključe i stručnjaci na nekim drugim područjima osim fitnessa. Jedan primjer su nutricionisti. Ovisno o ciljevima koje korisnik odabire, nutricionist mu preporuča prehrambene namirnice koje bi bilo poželjno da uvrsti u vlastitu prehranu. Korisniku bi se tako uz termine treninga u kalendaru prikazivao i jelovnik koji uključuje obroke koji sadrže preporučene namirnice.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnici treninga
2. Treneri
3. Administrator
4. Neregistrirani korisnici

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) registrirati se
 - (b) gumbom izbrisati sve unesene podatke pri registraciji
 - (c) prijaviti se
 - (d) pregledati početnu stranicu aplikacije
2. Korisnik treninga (inicijator) može:
 - (a) prijaviti se u sustav korisničkim imenom i lozinkom
 - (b) pregledavati i mijenjati svoje osobne podatke (lozinku)
 - (c) vidjeti kroz kalendar kojim treninzima može prisustvovati ovisno o vježbama koje mora odrađivati
 - (d) rezervirati treninge na koje želi ići ovisno o fondu sati kojeg ima
 - (e) otkazati rezervaciju treninga
3. Trener (inicijator) može:
 - (a) prijaviti se u sustav koristeći ime i lozinku
 - (b) pregledavati sve registrirane korisničke račune
 - (c) odabrati registriranog korisnika treninga kojem može dodijeliti vježbe ovisno o ciljevima osobe
 - (d) stvarati treninge na mjesečnoj bazi (definirati vježbe i maksimalan broj polaznika) te ih unositi u kalendar
 - (e) upisuje pravila o rezervaciji
4. Administrator (inicijator) može:
 - (a) prijaviti se
 - (b) administrirati korisničke račune te račune trenera
 - (c) ažurirati sve podatke u aplikaciji
 - (d) registrirati trenere

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1-Registracija

- **Glavni sudionik:** Neregistrirani korisnik treninga
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** UC15
- **Opis osnovnog tijeka:**
 1. Neregistrirani korisnik odabire akciju "Registration" na početnoj stranici
 2. Sustav otvara formu za upis podataka
 3. Neregistrirani korisnik upisuje potrebne podatke (ime, prezime, korisničko ime, email, lozinku i ciljeve)
 4. Neregistrirani korisnik odabire opciju "Submit"
 5. Sustav provjerava ispravnost unesenih podataka, sprema ih u bazu, prijavljuje korisnika, postavlja vrijeme isteka sesije na 30 minuta i otvara stranicu s osobnim podacima
- *** Opis mogućih odstupanja:** cx 5 m+cij9 jji66 +
 - 4.a Neregistrirani korisnik odabire akciju Reset, a sustav na formi registracije briše sve unesene podatke. Sustav nastavlja izvođenje scenarija u koraku 3.
 - 5.a Sustav provjerava i utvrđuje da uneseni podaci nisu ispravni. Obavještava korisnika odgovarajućom porukom te nastavlja izvođenje scenarija u koraku 3

UC2-Prijava

- **Glavni sudionik:** Korisnik treninga, trener, administrator
- **Cilj:** Prijaviti se u aplikaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** UC15
- **Opis osnovnog tijeka:**
 1. Neprijavljeni korisnik odabire akciju "Log in" na početnoj stranici
 2. Sustav otvara formu za upis podataka
 3. Neprijavljeni korisnik upisuje potrebne podatke u formu (korisničko ime i lozinku)
 4. Sustav provjerava ispravnost podataka, prijavljuje korisnika, postavlja vrijeme isteka sesije na 30 minuta i otvara stranicu s osobnim podacima
- **Opis mogućih odstupanja:**
 - 1.a Ako neprijavljeni korisnik nema račun može izabrati akciju "Nemaš račun?" te sustav otvara formu za Registraciju
 - 3.a Sustav provjerava i utvrđuje da uneseni podaci nisu ispravni. Obavještava korisnika odgovarajućom porukom te nastavlja izvođenje scenarija u koraku 2

UC3-Pregled osobnih podataka

- **Glavni sudionik:** Korisnik treninga, trener, administrator

- **Cilj:** Pregled osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Sustav prijavljenog korisnik nakon prijave/registracije automatski odvodi na stranicu osobnih podataka ili prijavljeni korisnik odabire akciju "Moji podaci" u zaglavlju aplikacije
 2. Korisnik ima uvid u svoje podatke (ime, prezime, korisničko ime, email, avatar)
- **Opis mogućih odstupanja:**
 1. /

UC4-Promjena osobnih podataka

- **Glavni sudionik:** Korisnik treninga
- **Cilj:** Promjena svojih osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** UC3
- **Opis osnovnog tijeka:**
 1. Korisnik odabire akciju "Izmijeni"
 2. Sustav otvara formu za izmjenu lozinke
 3. Korisnik Nakon promjene odabire akciju "Save changes"
 4. Sustav provjerava ispravnost podataka, sprema nove podatke u bazu i otvara stranicu s osobnim podacima
- **Opis mogućih odstupanja:**
 - 4.a Ako je korisnik mijenjao lozinku, sustav provjerava i utvrđuje da unesena stara lozinka nije ispravna. Obavještava korisnika odgovarajućom porukom te nastavlja izvođenje scenarija u koraku 2

UC5-Odabir ciljeva

- **Glavni sudionik:** Korisnik treninga
- **Cilj:** Odabir ciljeva za taj mjesec
- **Sudionici:** Baza podataka, trener
- **Preduvjet:** UC3
- **Opis osnovnog tijeka:**
 1. Korisnik na stranici osobnih podataka iz padajućih izbornika odabire ciljeve
 2. Korisnik odabire akciju "Promijeni ciljeve"
 3. Sustav provjerava ima li korisnik pravo na promjenu ciljeva, mijenja ciljeve u bazi i otvara ponovno stranicu osobnih podataka
- **Opis mogućih odstupanja:**
 - 1.a Sustav je pri prijavi utvrdio da je korisnik taj mjesec već promijenio ciljeve i onemogućuje padajuće izbornike i gumb "Promijeni ciljeve".

UC6-Pregled termina treninga

- **Glavni sudionik:** Korisnik treninga

- **Cilj:** Pregled datuma treninga
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Korisnik odabire akciju "Rezervacije" u zaglavlju aplikacije
 2. Sustav otvara kalendar s terminima treninga koje taj korisnik može rezervirati i naznačenim preostalim satima
 3. Korisnik odabire jedan od dostupnih treninga
 4. Sustav prikazuje u popup prozoru detalje o terminu (vrijeme početka, vrijeme kraja, ime i prezime trenera, listu vježbi i broj preostalih mjesta)
- **Opis mogućih odstupanja:**
 - 3.a Sustav utvrđuje da korisniku nisu dodijeljene vježbe pa time ni treninzi te obavještava korisnika porukom na sredini ekrana.

UC7-Rezervacija termina treninga

- **Glavni sudionik:** Korisnik treninga
- **Cilj:** Rezervacija termina
- **Sudionici:** Baza podataka
- **Preduvjet:** UC6
- **Opis osnovnog tijeka:**
 1. Korisnik nakon odabira treninga odabire akciju "Reserve"
 2. Sustav provjerava fond sati, smanjuje ih za 1, prema podatke za korisnika i termin u bazu te otvara stranicu Rezervacije
 3. Sustav rezrviranom treningu nadodaje akciju "Cancel reservation"
- **Opis mogućih odstupanja:**
 - 1.a Sustav utvrđuje da u terminu nema više slobodnih mjesta te onemogućuje gumb Reserve za taj termin
 - 1.b Sustav utvrđuje da korisnik nema više preostalih sati, onemogućuje gumb Reserve i obavještava korisnika porukom iznad kalendara

UC8-Pregled datuma treninga i termina po danu 2

- **Glavni sudionik:** Trener
- **Cilj:** Pregled datuma treninga
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Trener odabire akciju "Termini" u zaglavlju aplikacije
 2. Sustav otvara kalendar sa svim terminima drugih trenera i njegovim, posebno obojanim, terminima treninga.
 3. Trener odabire jedan od termina
 4. Sustav prikazuje u popup prozoru detalje o terminu (vrijeme početka, vrijeme kraja, ime i prezime trenera, listu vježbi i broj preostalih mjesta)
- **Opis mogućih odstupanja:**

1. /

UC9-Unos treninga

- **Glavni sudionik:** Trener
- **Cilj:** Unos novog termina
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Trener odabire akciju "Termini" u zaglavlju aplikacije
 2. Sustav otvara kalendar sa svim terminima drugih trenera i njegovim, posebno obojanim, terminima treninga.
 3. Trener odabire akciju dodaj termin
 4. Sustav otvara formu za unos podataka
 5. Trener unosi podatke (vrijeme početka treninga, vrijeme kraja, vrstu treninga i broj mjesta) i odabire akciju "Create"
 6. Sustav provjerava unesene podatke, sprema ih u bazu i otvara kalendar s terminima
- **Opis mogućih odstupanja:**
 - 6.a Sustav utvrđuje da postoji preklapanje između tog termina i već postojećeg ili da je prevelik broj polaznika te obavještava korisnika odgovarajućom porukom i anstavlja izvođenje od koraka 4

UC10-Pregled liste registriranih korisnika

- **Glavni sudionik:** Trener
- **Cilj:** Pregled korisnika treninga
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Trener odabire akciju "Korisnici" u zaglavlju aplikacije
 2. Sustav otvara stranicu s listom registriranih korisnika treninga, oni bez dodijeljenih vježbi su obojani drugačije i imaju uz sebe gumb "Dodijeli vježbu"
- **Opis mogućih odstupanja:**
 - 2.a Sustav utvrđuje da nema registriranih korisnika te obavještava trenera odgovarajućom porukom

UC11-Dodjela vježbi

- **Glavni sudionik:** Trener
- **Cilj:** Dodjela vježbi korisniku treninga shodno njihovom cilju
- **Sudionici:** Baza podataka, usluga e-pošte
- **Preduvjet:** UC10
- **Opis osnovnog tijeka:**
 1. Trener odabire akciju "Dodijeli vježbe"
 2. Sustav otvara popup prozor s opisom ciljeva tog korisnika i 3 padajuća izbornika za 3 vježbe

3. Trener bira vježbe i odabire akciju "Save"
 4. Sustav provjerava vježbe, sprema ih u bazu i otvara nanovo popis korisnika
- **Opis mogućih odstupanja:**
 - 4.a Sustav utvrđuje da je trener odabrao više istih vježbi, obavještava ga porukom i nastavlja od koraka 2

UC12-Brisanje termina treninga 3

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled termina
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Administrator odabire akciju "Termini" u zaglavlju aplikacije
 2. Sustav otvara stranicu s kalendarom sa svim postojećim terminima treninga
 3. Administrator odabire termin te odabire opciju "Delete"
 4. Sustav briše termin iz baze podataka i otvara nanovo kalendar s terminima
- **Opis mogućih odstupanja:**
 1. /

UC13-Registracija trenera

- **Glavni sudionik:** Administrator
- **Cilj:** Registracija trenera
- **Sudionici:** Baza podataka
- **Preduvjet:** UC2
- **Opis osnovnog tijeka:**
 1. Administrator odabire akciju "Register Trainer" u zaglavlju aplikacije
 2. Sustav otvara formom za upis podataka o novom treneru (ime, prezime, korisničko ime, lozinka, email)
 3. Administrator upisuje podatke i odabire akciju "Register"
 4. Sustav provjerava podatke sprema trenera u bazu i otvara početnu stranicu
- **Opis mogućih odstupanja:**
 - 4.a Sustav utvrđuje da su uneseni krivi podaci(npr. korisničko ime već postoji), obavještava administratora odgovarajućom porukom i nastavlja od koraka 2

UC14-Odjava

- **Glavni sudionik:** Prijavljeni korisnik
- **Cilj:**Odjava s računa
- **Sudionici:** Baza podataka
- **Preduvjet:** UC3
- **Opis osnovnog tijeka:**
 1. Korisnik odabire akciju "Log out"

2. Sustav odjavljuje korisnika i otvara početnu stranicu

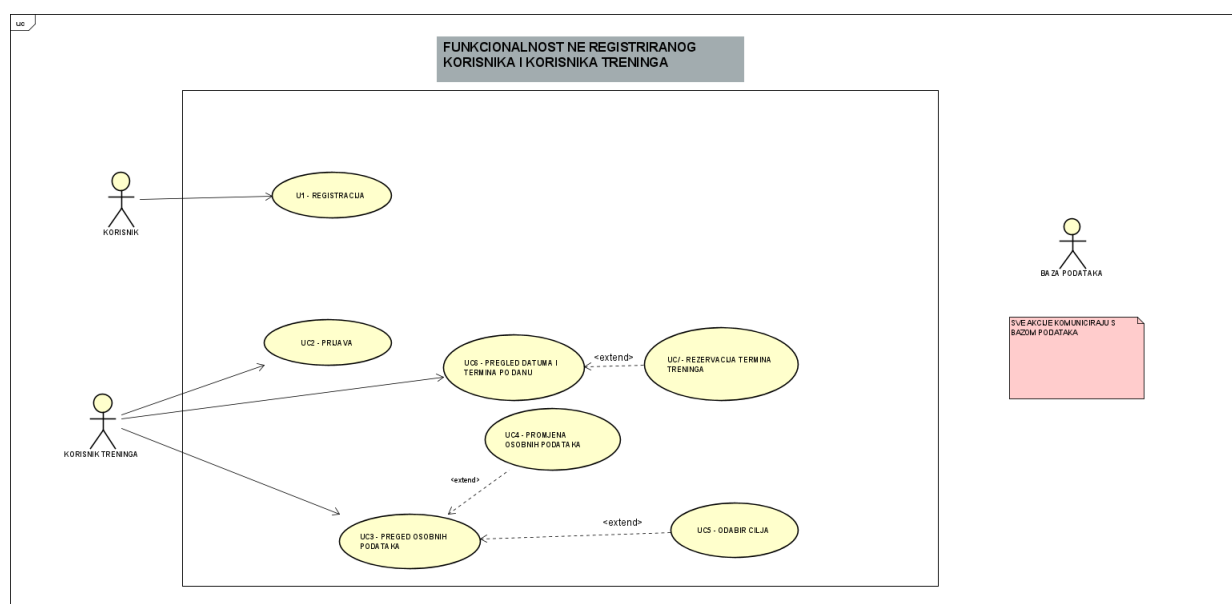
- Opis mogućih odstupanja:

1. /

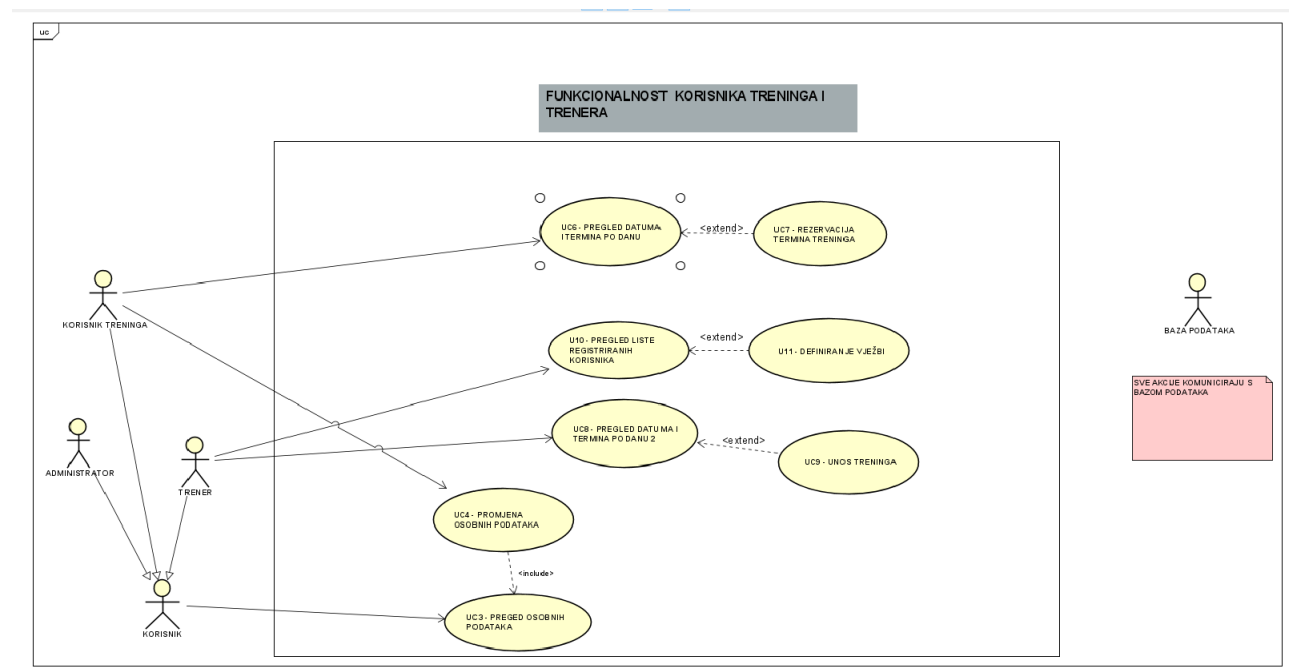
UC15-Pregled početne stranice

- **Glavni sudionik:** Korisnik (bilo tko)
- **Cilj:** Pregled početne stranice aplikacije na kojoj pobliže opisujemo te reklamiramo našu aplikaciju koja koristi za upravljanje grupnih treninga
- **Sudionici:** /
- **Preduvjet:** /
- **Opis osnovnog tijeka:**
 1. Upis adrese naše aplikacije u internet tražilicu
- **Opis mogućih odstupanja:**
 1. /

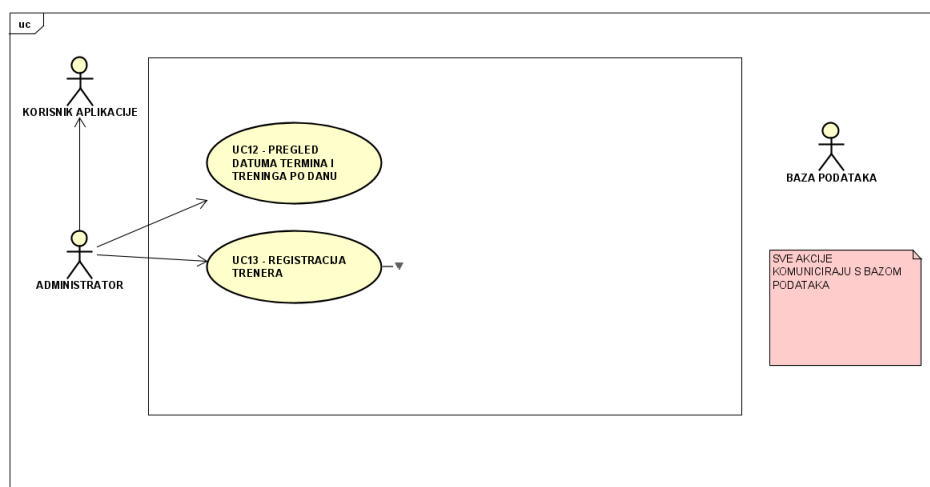
Dijagrami obrazaca uporabe



Slika 3.1: "Dijagram funkcionalnosti neregistriranog korisnika i korisnika treninga"



Slika 3.2: "Dijagram funkcionalnosti korisnika treninga i trenera"

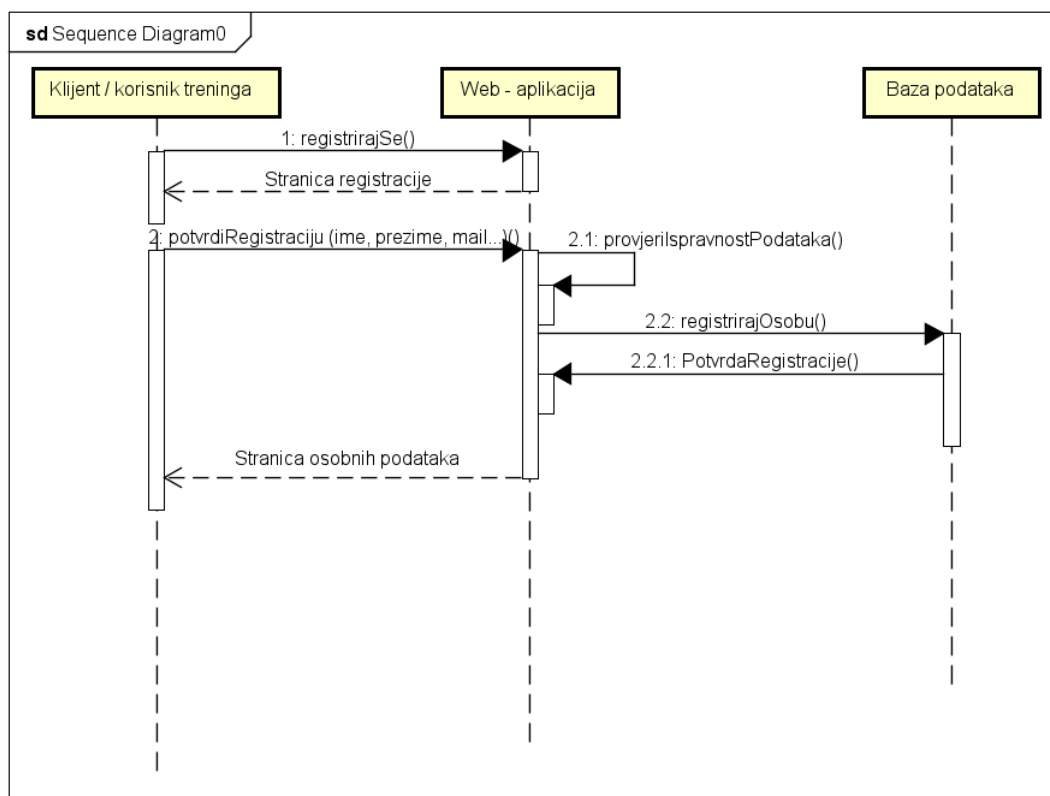


Slika 3.3: "Dijagram funkcionalnosti korisnika aplikacije i administratora"

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC1 - Registracija

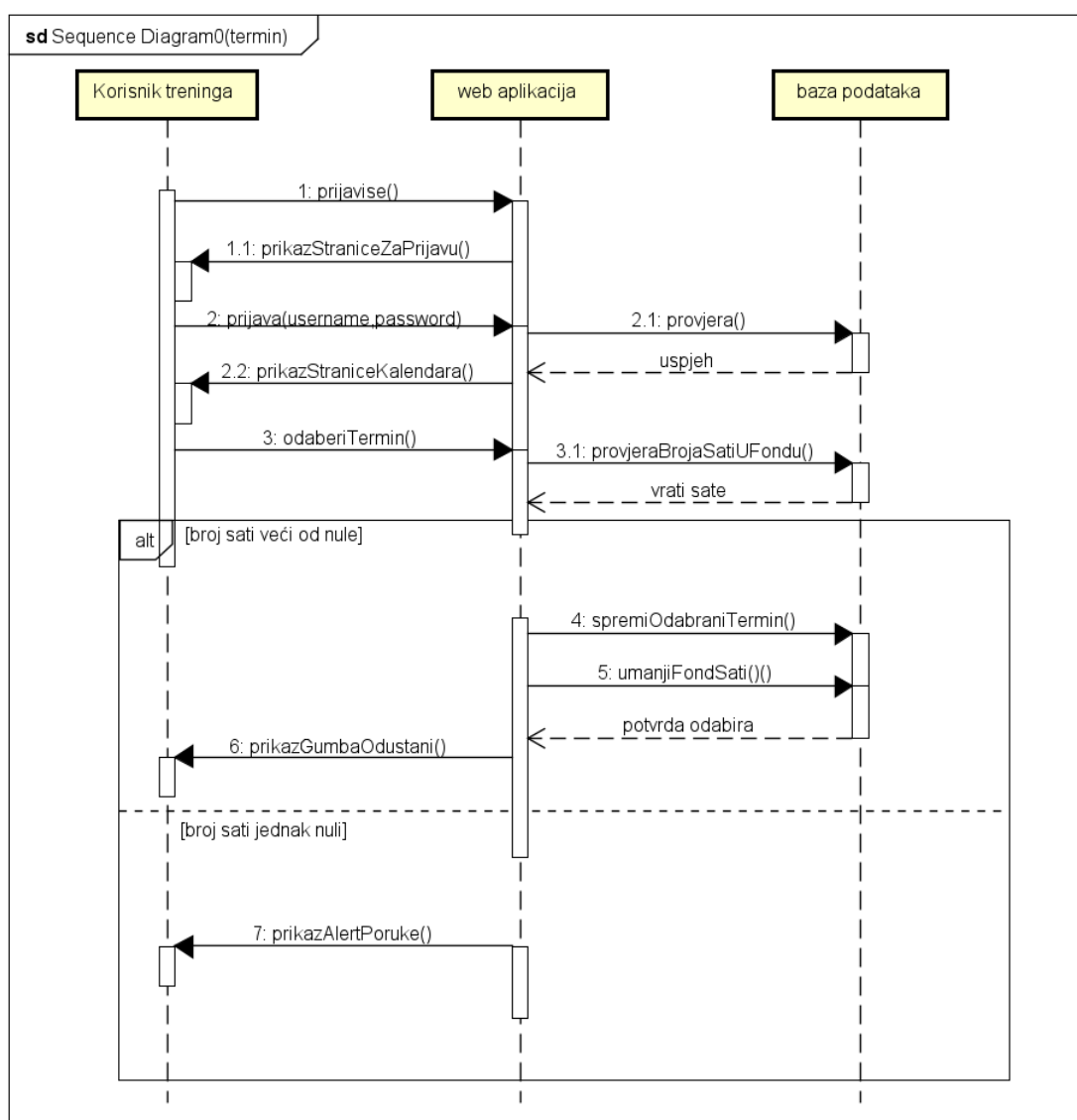
Klijent šalje zahtjev za stranicu registracije kako bih se mogao registrirati. Poslužitelj dohvaća stranicu za registraciju te ju prikazuje. Zatim klijent ispunjava formu za registraciju te ju šalje poslužitelju. Poslužitelj provjerava ispravnost podataka te ukoliko su ispravni zapisuje podatke u bazu podataka. Nakon uspješne registracije poslužitelj klijentu prikazuje stranicu s osobnim podacima.



Slika 3.4: "Registracija"

Obrazac uporabe UC7 - Rezervacija termina treninga

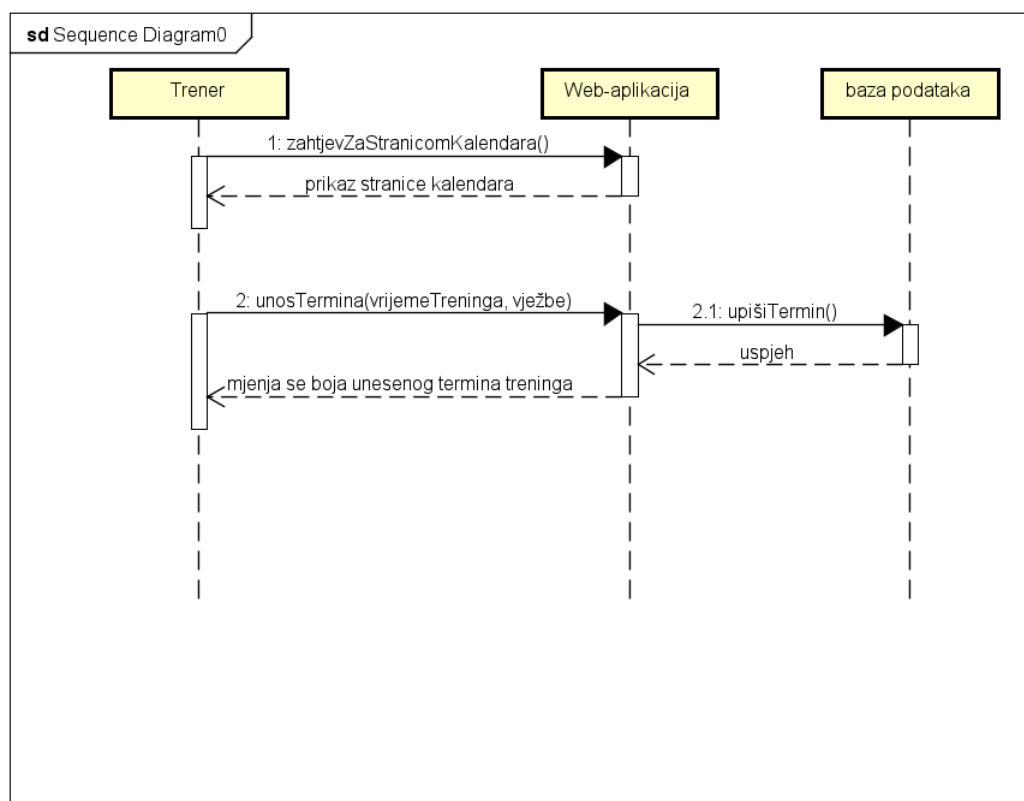
Klijent šale zahtjev za prikaz stranice za prijavu. Poslužitelj vraća stranicu za prijavu te klijent nastavlja s prijavom unoseći svoje podatke(username i password). Nakon unosa podataka poslužitelj provjerava ispravnost podataka sa bazom podataka te ukoliko je uspješno poslužitelj klijentu prikazuje stranicu kalendara. Zatim klijent bira željene termine te poslužitelj provjerava u bazi podataka ima li klijent dovoljno sati u vlastitom fondu. Ukoliko nema prikazuje se alert poruka u suprotnom poslužitelj sprema odabrane termine klijenta u bazu podataka te umanjuje preostali fond sati klijenta. Nakon obavljenih operacija poslužitelj klijentu prikazuje tipku "odustani" ukoliko klijent želi odustati od rezerviranih termina.



Slika 3.5: "Rezervacija termina treninga"

Obrazac uporabe UC9 - Unos treninga

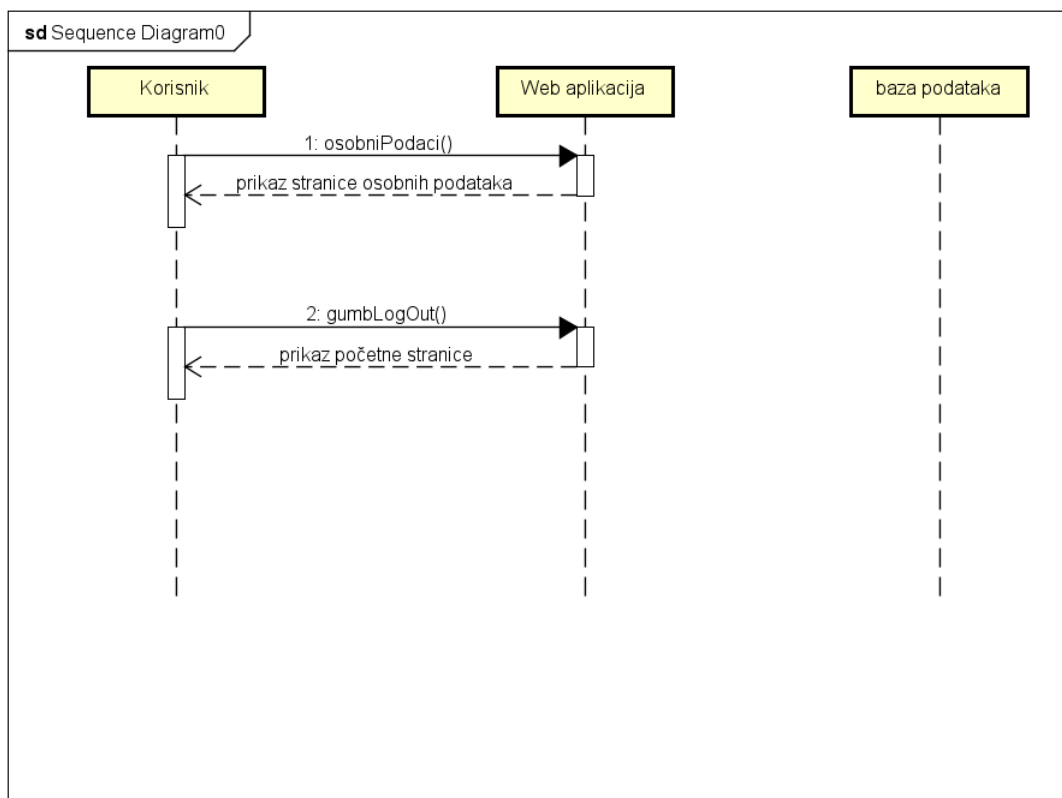
Klijent u ovom slučaju trener šalje zahtjev za prikaz stranice kalendara. Poslužitelj vraća stranicu kalendara te trener nastavlja s unosom termina treninga birajući na kalendaru u kojem terminu želi da se trening odvija. Nakon odabira termina poslužitelj sprema odabrani termin u bazu podataka zajedno s vježbama koje će se obavljati tokom treninga. Nakon spremanja u bazu podataka poslužitelj mjenja boju termina kako bih znali da je termin popunjen.



Slika 3.6: "Unos treninga"

Obrazac uporabe UC14 - Odjava

Klijent zatraži od poslužitelja stranicu osobnih podataka. Poslužitelj vraća klijentu stranicu poslužitelja te zatim klijent klikne gumb "logout". Poslužitelj zatim odjavljenom korisniku prikazuje početnu stranicu web aplikacije.



Slika 3.7: "Odjava"

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
- Aplikacija treba biti prilagođena za rad na različitim uređajima (mobilni uređaj, tablet, PC)
- Aplikacija mora osigurati sigurnost baze podataka i otpornost na vanjske prijetnje
- Aplikacija mora omogućiti rad više korisnika istovremeno
- Korisničko sučelje treba podržavati englesku abecedu pri unosu i prikazu sadržaja

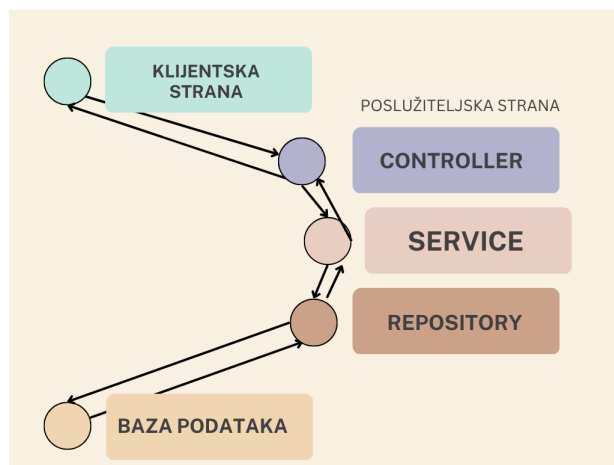
4. Arhitektura i dizajn sustava

Arhitektura sustava naše aplikacije implementirana je u arhitekturi klijent-poslužitelj.

Na klijentskoj strani koristili smo programski jezik *JavaScript*, radni okvir *React* i biblioteku komponenta *Material UI* u uređivaču *Visual Studio Code*.

Na poslužiteljskoj strani koristili smo programski jezik *Java* i radni okvir *Spring Boot* u uređivaču *IntelliJ IDEA*. Poslužiteljska strana organizirana je u tri sloja. To su nadglednik (controller), usluga (service) i repozitorij (repository). Nadglednik povezuje korisničku stranu s poslužiteljskom stranom. Sloj usluga ostvaruje temeljnu funkcionalnost web aplikacije i definira jednu ili više usluga web aplikacije. Repozitorij osigurava pristup podacima.

Podatke smo spremali u relacijsku bazu podataka *PostgreSQL* koristeći *Java Persistence API*.



Slika 4.1: Povezanost slojeva u radnom okviru Spring

Osim toga postoji i razrađeni model podataka domene (domenski objekti) kojeg koriste svi slojevi po potrebi. Razredi koji su sadržani u modelu domene imaju oznaku `@Entity` i sadrže podatke koji se postavljaju i čitaju putem metoda `get` i `set`. Nadglednik koristi domenske objekte za automatsko postavljanje vrijednosti objekata iz tijela zahtjeva (`@RequestBody`) od klijentske strane, sloj usluge koristi ih za ostvarenje odgovarajuće funkcionalnosti aplikacije, a repozitorij za automatsko-preslikavanje objekta u relaciju u bazi podataka i obratno.

4.1 Baza podataka

U implementaciji naše web aplikacije koristimo relacijsku bazu podataka. Objekti u relacijskoj bazi podataka su relacije, tj. dvodimenzionalne tablice koje su definirane imenom, a atributi su imenovani stupci relacije. Glavna zadaća baze podataka je pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- User
- Updates
- TrainingSession
- TrainingType
- Exercise
- Goal

4.1.1 Opis tablica

Entitet **User** sadrži sve podatke o korisniku aplikacije. Sadrži attribute korisničko ime, ime korisnika, prezime, email adresu, lozinku, ulogu koju korisnik ima, prvi i drugi cilj, preostali broj termina treninga koje korisnik ima pravo rezervirati ovaj mjesec i novi cilj korisnika. Ovaj entitet u vezi je *One-to-One* s entitetom *Updates* preko atributa username i u vezi je s *Many-to-One* s *TrainingSession* preko atributa username.

User		
username	VARCHAR	jedinstveno korisničko ime
firstName	VARCHAR	ime korisnika
lastName	VARCHAR	prezime korisnika
email	VARCHAR	e-mail korisnika
password	VARCHAR	lozinka korisnika
role	VARCHAR	uloga korisnika
goal1	VARCHAR	prvi cilj korisnika
goal2	VARCHAR	drugi cilj korisnika
remainingTrainingSessions	INT	preostali termini korisnika
newGoal	INT	novi cilj

Entitet **Updates** identifikacijski je slabi entitet koji ovisi o entitetu *User*. Sadrži podatke o zadnjem ažuriranju preostalih termina korisnika i posljednjoj promjeni ciljeva korisnika. Sadrži attribute korisničko ime, oznaku kada je ažuriran fond sati i oznaku kada je omogućena posljednja promjena cilja. Ovaj entitet u vezi je *One-to-One* s entitetom *User* preko atributa username.

Updates		
username	VARCHAR	jedinstveno korisničko ime
sessionsUpdate	TIMESTAMP	oznaka kad je ažuriran fond sati
goalsUpdate	TIMESTAMP	oznaka kad je omogućena posljednja promjena cilja

Entitet **TrainingSession** sadrži podatke o terminima treninga. Sadrži attribute identifikator ter-

mina treninga, početno vrijeme treninga, vrijeme kad završava trening, korisničko ime i identifikator vrste treninga. Ovaj entitet u vezi je *One-to-Many* s entitetom *User* preko atributa username., u vezi *One-to-One* s entitetom *TrainingType* preko atributa trainingTypeId te u vezi *One-to-One* s entitetom *Exercise* preko atributa exerciseId i trainingSessionId te se na vezi nalazi atribut TrainingExerciseId.

TrainingSession		
trainingSessionId	INT	jedinstveni identifikator termina treninga
startDateTime	TIMESTAMP	početak treninga
endDateTime	TIMESTAMP	kraj treninga
username	VARCHAR	jedinstveno korisničko ime
trainingTypeId	INT	jedinstveni identifikator treninga

Entitet **TrainingType** sadrži podatke o vrsti treninga. Sadrži attribute jedinstveni identifikator treninga i naziv vrste treninga. Ovaj entitet u vezi je *One-to-One* s entitetom *User* preko atributa username.

TrainingType		
trainingTypeId	INT	jedinstveni identifikator treninga
trainingType	VARCHAR	vrsta treninga

Entitet **Exercise** sadrži podatke o vježbama. Sadrži attribute jedinstveni identifikator vježbe i naziv vježbe. Ovaj entitet u vezi je *One-to-One* s entitetom *TrainingSession* preko atributa trainingSessionId i exerciseId te se na vezi nalazi atribut trainingExerciseId.

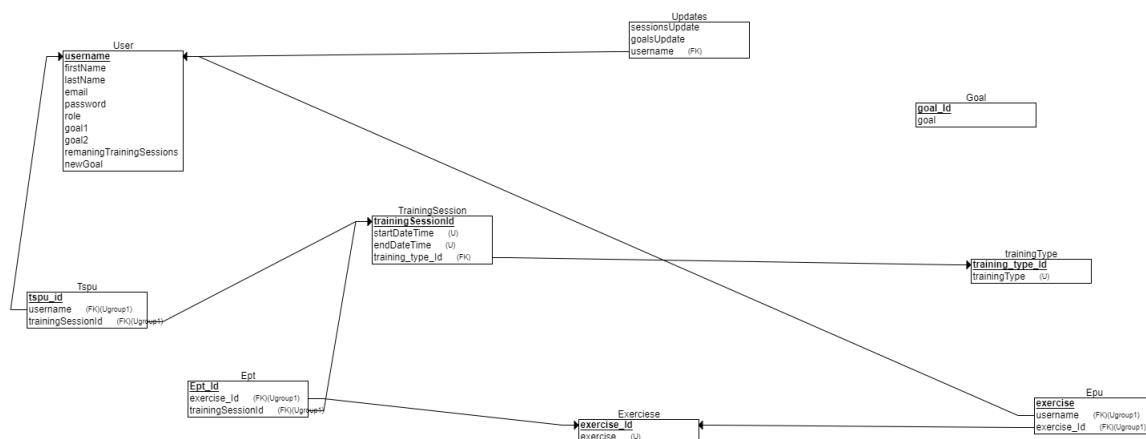
Exercise		
exerciseId	INT	jedinstveni identifikator vježbe
exercise	VARCHAR	vježba

TrainingExercise		
trainingExerciseId	INT	jedinstveni identifikator vježbe treninga
exerciseId	INT	jedinstveni identifikator vježbe
trainingSessionId	INT	jedinstveni identifikator termina treninga

Entitet **Goal** sadrži podatke o ciljevima korisnika. Sadrži attribute jedinstveni identifikator cilja i naziv cilja.

Goal		
goalId	INT	jedinstveni identifikator cilja
goal	VARCHAR	cilj

4.1.2 Dijagram baze podataka

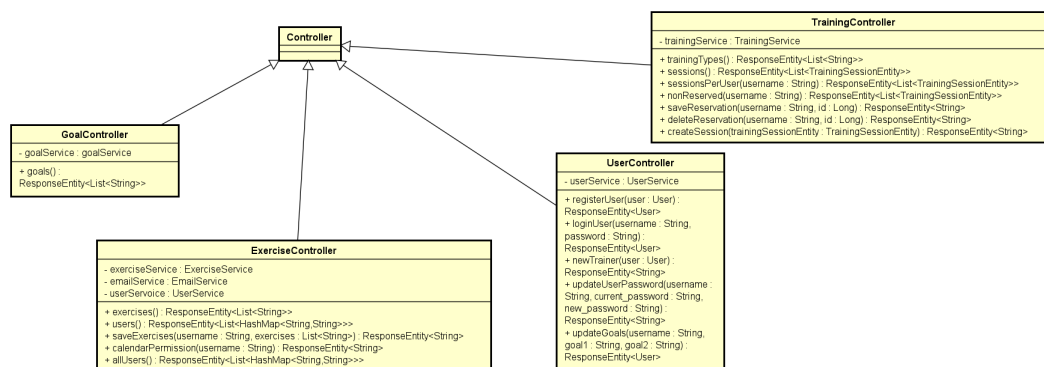


Slika 4.2: dijagram baze podataka

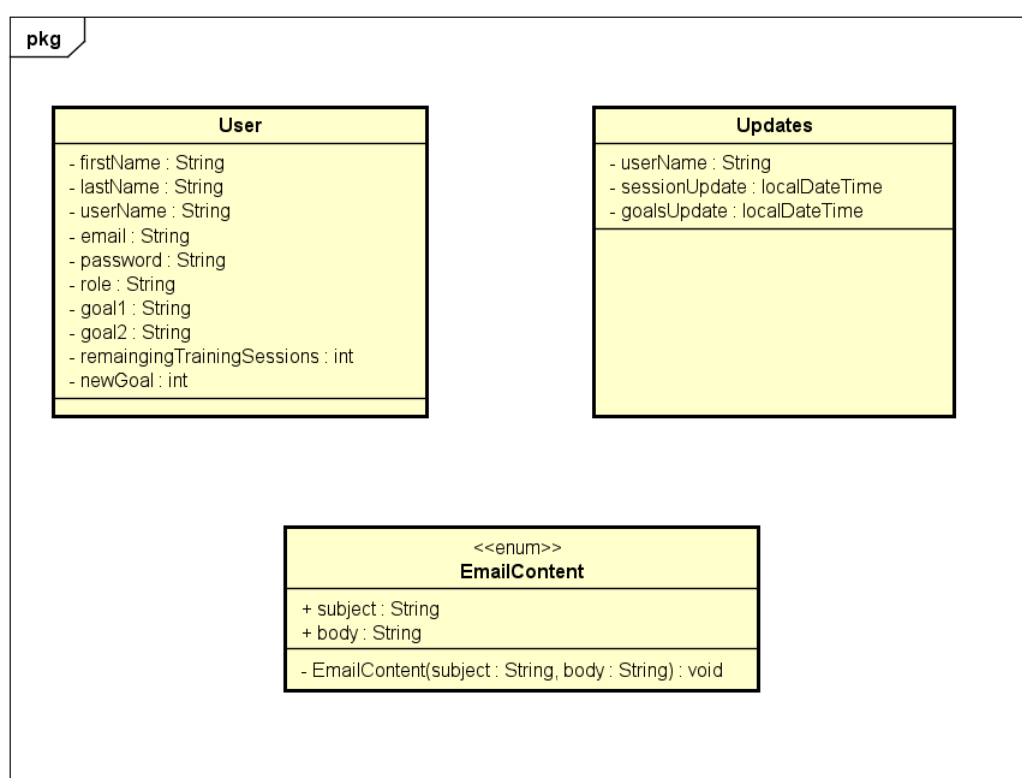
4.2 Dijagram razreda

Na sljedećij nekoliko fotografija(4.3, 4.4, 4.5, 4.6 te 4.7) prikazani su razredi backenda koji pripadaju spring arhitektur točnije troslojnom modelu. Konkretno na fotografiji 4.3 prikazani su controller razredi, na fotografiji 4.4 razredi modela, na fotografiji 4.5 razredi repozitorija, na fotografiji 4.6 razredi pripadnih entiteta te na fotografiji 4.7 razredi servisa .

Nadalje dijagrami su organizirani tako da zrcale komponente backend infrastrukture kako bih dijagrami bili pregledniji te logički smisleni.

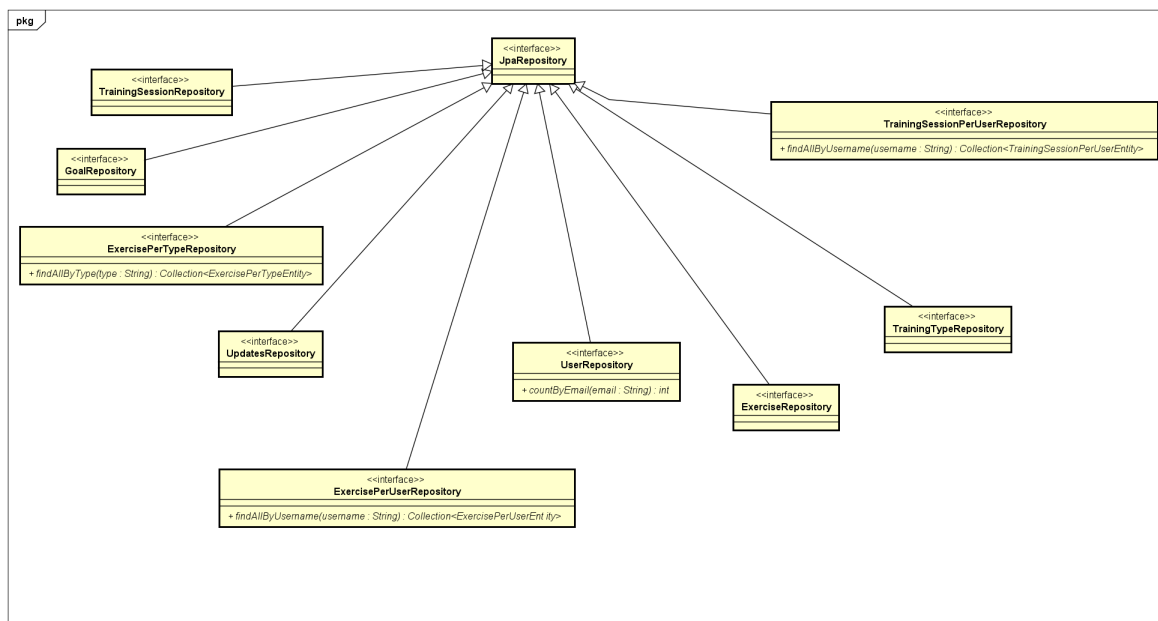


Slika 4.3: dijagram razreda - Dijagram kontrolera

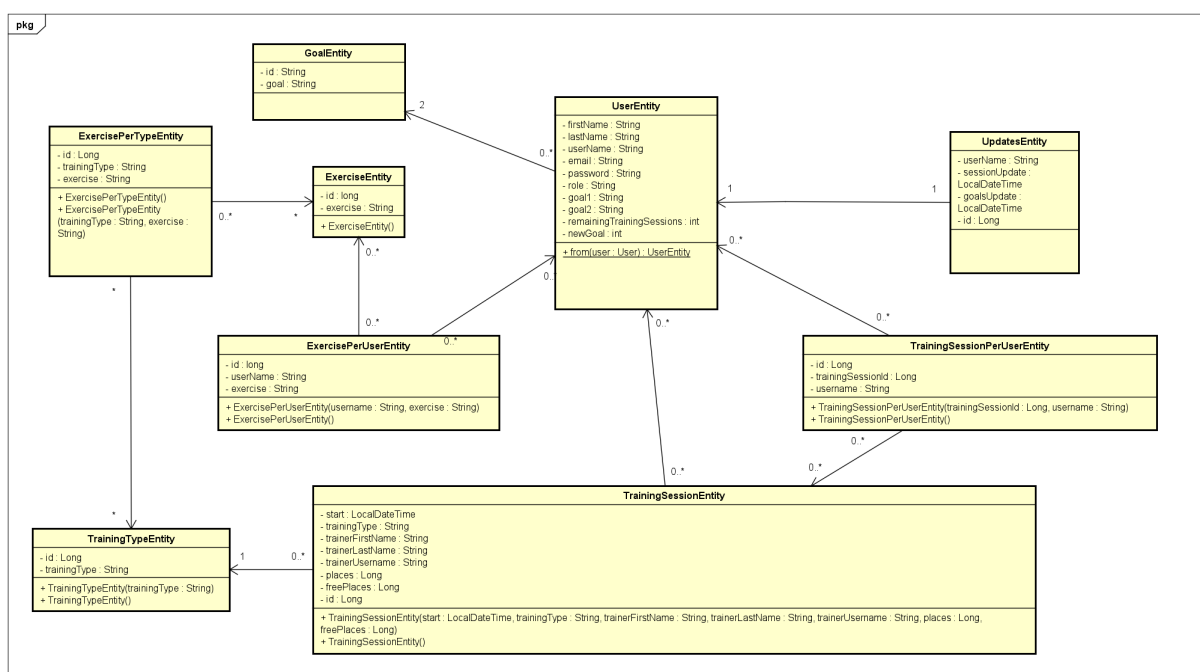


Slika 4.4: dijagram razreda - Dijagram modela

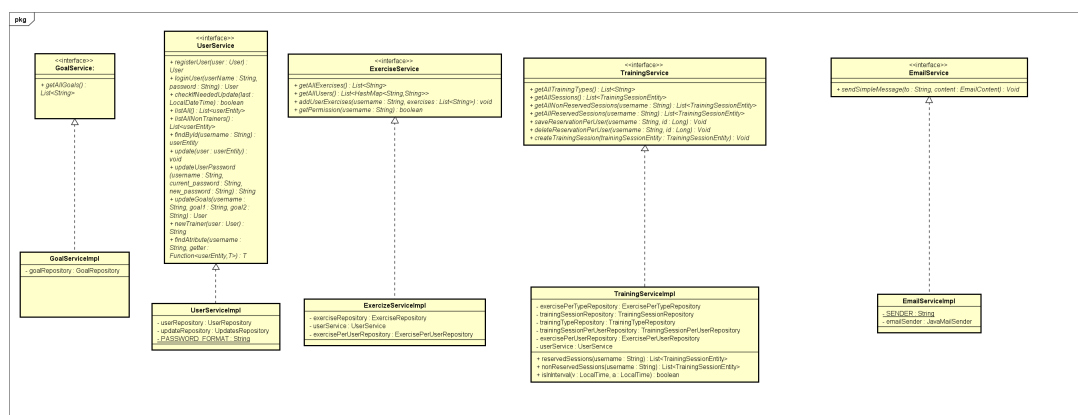
Na fotografiji 4.4 prikazan je dijagram razreda modela koji preslikavaju entitete i attribute istih na temelju modela baze podataka.



Slika 4.5: dijagram razreda - Dijagram repozitorija



Slika 4.6: dijagram razreda - dijagram entiteta



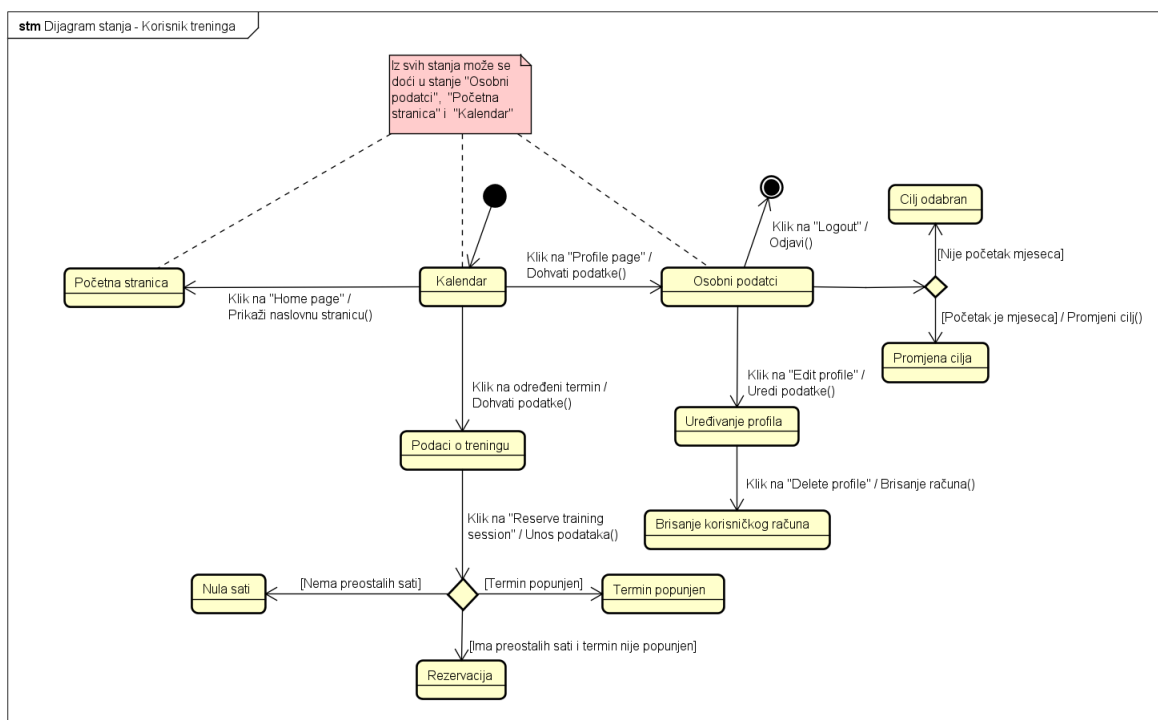
Slika 4.7: dijagram razreda - dijagram servisa

dio 2. revizije

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

4.3 Dijagram stanja

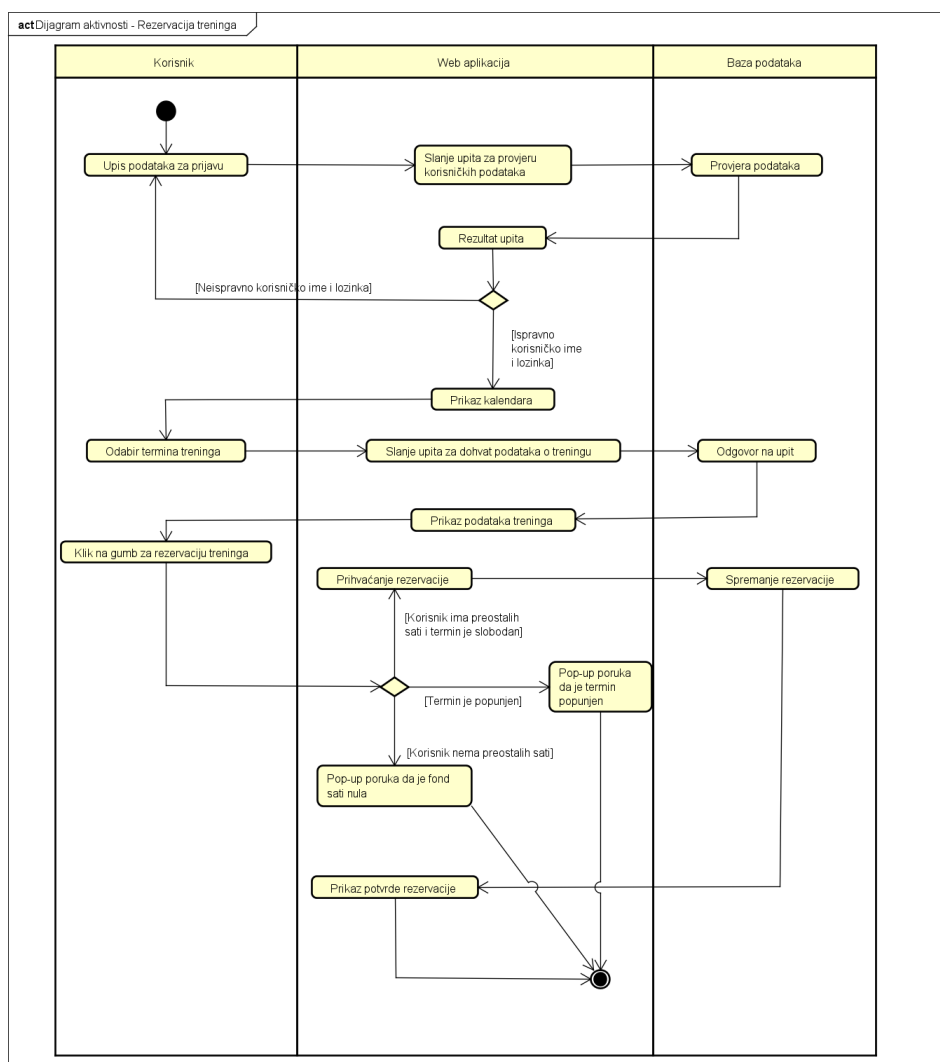
Dijagram stanja opisuje dinamičko ponašanje dijela sustava u vremenu, prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.8 prikazan je dijagram stanja za registriranog korisnika treninga. Nakon prijave, korisniku se prikazuje njegov personalizirani kalendar. Odabirom određenog termina prikazuju mu se podatci o treningu, to jest od kojih se vježbi sastoji i koliko traje. Ako korisnik ima preostalih sati te određeni termin nije popunjen, korisniku je omogućen gumb "Reserve training session" te klikom na gumb rezervira trening. U slučaju da je korisnikov fond sati nula i/ili da je termin popunjen korisniku je gumb onemogućen te klikom na njega pojavljuje mu se pop-up poruka da ne može rezervirati trening. Također, u zaglavlju stranice korisnik može odabrati neku od ponuđenih mogućnosti. Klikom na "Home page", korisniku se prikazuje naslovna stranica, a klikom na "Profile page" prikazuje mu se stranica s osobni podacima. Tamo se nalaze osobni podatci korisnika te njegov odabrani cilj. Ukoliko je početak mjeseca, utoliko je korisniku omogućen padajući izbornik za promjenu cilja. Također, klikom na gumb za uređivanje profila korisniku je omogućena izmjena podataka te mu je prikazan gumb za brisanje korisničkog profila.



Slika 4.8: Dijagram stanja - Korisnik treninga

4.4 Dijagram aktivnosti

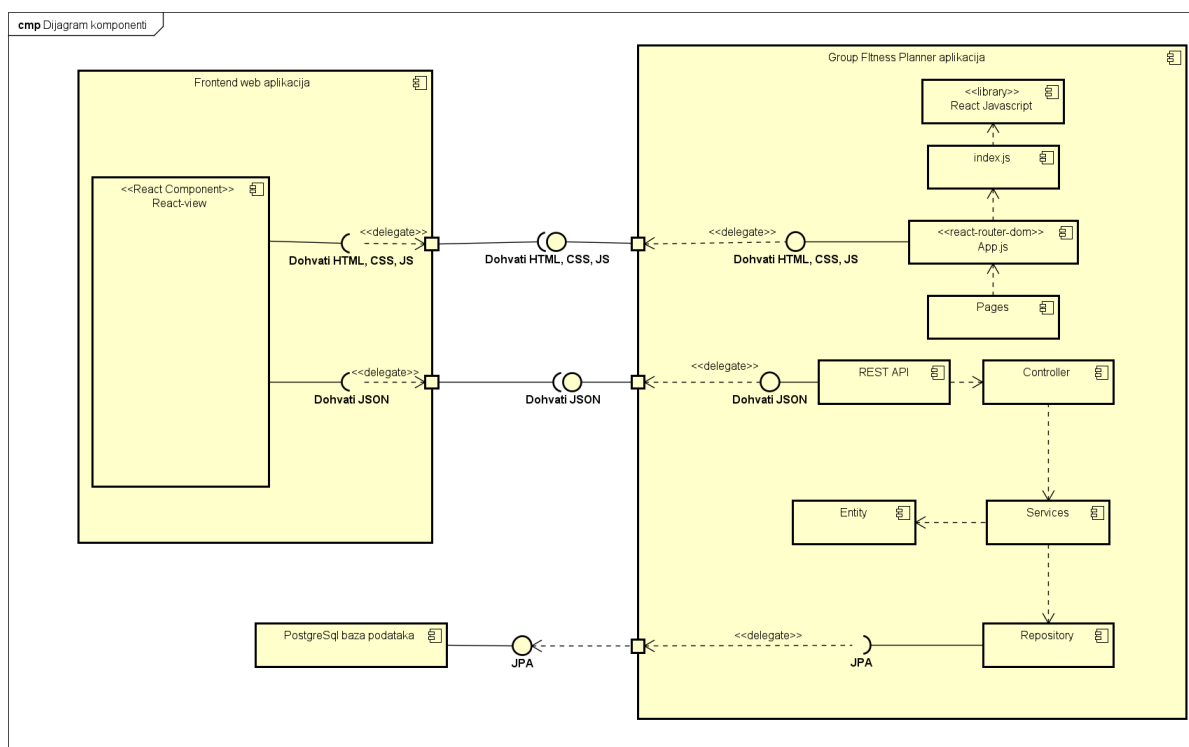
Dijagrami aktivnosti primjenjuju se za modeliranje poslovnih procesa i upravljačkog i podatkovnog toka. Na slici 4.9 prikazan je proces rezervacije treninga. Korisnik se prijavi u sustav, prikaže mu se personalizirani kalendar i odabire termin treninga za koji želi napraviti rezervaciju. U slučaju da korisnik ima preostalih sati te da termin nije popunjen, rezervacija se sprema te se prikazuje potvrda rezervacije. Ako je korisnikov fond sati nula ili je termin popunjen, korisniku je onemogućen gumb za rezervaciju te se klikom na njega prikazuje pop-up poruka da ne može rezervirati trening.



Slika 4.9: Dijagram aktivnosti - Rezervacija treninga

4.5 Dijagram komponenti

Dijagram komponenti statički je UML dijagram. Slika 4.10 opisuje organizaciju i međuovisnost komponenti, interne strukture i odnose prema okolini. Razlikujemo dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju klijentskom dijelu aplikacije. App.js komponenta je koja na upit s url određuje koja će se datoteka poslužiti na sučelje. Klijentski dio aplikacije se sastoji od niza JavaScript datoteka koje su sakupljene u komponentu Pages. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente kao što su gumbi i forme. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju poslužiteljskom dijelu aplikacije. Repository komponenta služi za dohvaćanje tablica iz baze podataka pomoću SQL upita. React-view komponenta preko dostupnih sučelja komunicira s aplikacijom te ovisno o akcijama korisnika osvježava prikaz i dohvaća nove podatke.



Slika 4.10: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Organizacija te raspodjela i komunikacija tima odvijala se preko aplikacije WhatsApp¹. Nadalje UML dijagrami su su napravljeni u alatu Astah Professional² te je za upravljanje kodom korišten Git³, a cijeli projekt je dostupan na udaljenom repozitoriju u sklopu web aplikacije GitLab⁴.

Pri konstrukciji front end-a korišten je Microsoft Visual studio code⁵ - uređivač koda tvrtke Microsoft. Primarno se koristi kao uređivač koda te sadrži podršku za razvojne operacije kao što su debuggiranje, izvođenje zadataka te verzioniranje. Nadalje pri konstrukciji back end-a korišten je IntelliJ IDEA⁶ integrirano radno okruženje (IDE) tvrtke JetBrains. Primarno se koristi za razvoj softvera napisanog u Javi, Kotlinu te drugim jezicima temeljenim na JVM-u (Java virtual machine).

Web aplikacija je napisana koristeći radni okvir Spring boot⁷ i jezik Java⁸ za izradu back end-a te za izradu front end-a korišten je jezik JavaScript⁹ odnosno njegova biblioteka otvorenog tipa React¹⁰ održavana od strane tvrtke Meta te neovisni programera. React se najčešće koristi kao osnova u razvoju web aplikacija. Složene aplikacije u Reactu zahtijevaju korištenje dodatnih biblioteka za interakciju s API-jem. Radni okvir Spring boot je ekstenzija Spring radnog okvira te eliminira konfiguraciju okvirne ploče potrebne za postavljanje spring aplikacije.

Baza podataka pisana je u PostgreSQL¹¹ SQL jeziku te je za njeno kreiranje korišten radna okolina PgAdmin¹². Nadalje aplikacija je podignuta na web pomoću oblaka Render¹³.

¹<https://www.whatsapp.com/>

²<https://astah.net/products/astah-professional/>

³<https://git-scm.com/>

⁴<https://about.gitlab.com/>

⁵<https://code.visualstudio.com/>

⁶<https://www.jetbrains.com/idea/>

⁷<https://spring.io/projects/spring-boot>

⁸<https://www.java.com/en/>

⁹<https://www.javascript.com/>

¹⁰<https://reactjs.org/>

¹¹<https://www.postgresql.org/>

¹²<https://www.pgadmin.org/>

¹³<https://render.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitni slučaj 1: Uspješno stvaranje trenera

```
@Test
public void testCreatingNewTrainerSuccessfully() throws Exception{

    User mockUser = UserGeneratingUtil.createMockUser();

    when(userService.newTrainer(mockUser)).thenReturn("success")

    mvc.perform(post("/new-trainer").contentType(MediaType.APPLICATION_JSON).content(objectMapper.writeValueAsString(mockUser)))
        .andExpect(status().isOk())
        .andExpect(content().string("success"));

}
```

Ispitni slučaj 2: Neuspješno stvaranje trenera

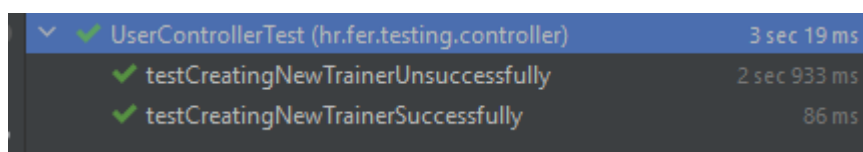
```
@Test
public void testCreatingNewTrainerUnsuccessfully() throws Exception{

    User mockUser = UserGeneratingUtil.createMockUser();

    when(userService.newTrainer(mockUser)).thenThrow(new RequestDeniedException("Username: " + mockUser.getUsername()));

    mvc.perform(post("/new-trainer").contentType(MediaType.APPLICATION_JSON).content(objectMapper.writeValueAsString(mockUser)))
        .andExpect(status().is4xxClientError());

}
```



✓	UserControllerTest (hr.fer.testing.controller)	3 sec 19 ms
✓	testCreatingNewTrainerUnsuccessfully	2 sec 933 ms
✓	testCreatingNewTrainerSuccessfully	86 ms

Slika 5.1: Rezultati ispitnih slučajeva 1 i 2

Ispitni slučaj 3: dohvaćanje treninga

```
@Test
public void testGetAllSessions() {

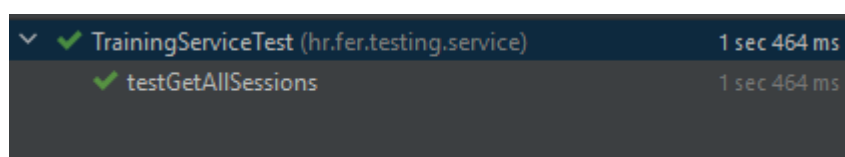
    TrainingSessionEntity mockTs = TrainingGeneratingUtil.createMockSession();

    when(trainingSessionRepository.findAll()).thenReturn(List.of(mockTs));

    verify(trainingSessionRepository, times(1)).findAll();

    Assert.assertEquals(found.size(), 1);

}
```



Slika 5.2: Rezultat ispitnog slučaja 3

Ispitni slučaj 4: Prikaz svih ne trenera

```
@Test
public void testListingAllNonTrainers() {

    User mockUser = UserGeneratingUtil.createMockUser();

    UserEntity mockUserEntity = UserEntity.from(mockUser);

    when(userRepository.findAll()).thenReturn(List.of(mockUserEntity));

    List<UserEntity> found = userService.listAllNonTrainers();

    verify(userRepository, times(1)).findAll();

    Assert.assertEquals(found.size(), 0)

}
```

Ispitni slučaj 5: Neuspješno dodavanje novog trenera


```
@Test
public void testAddingNewTrainerUnsuccessfully() {

    User mockUser = UserGeneratingUtil.createMockUser();

    UserEntity mockUserEntity = UserEntity.from(mockUser);

    when(userRepository.findById(mockUser.getUsername())).thenReturn(Optional.of(mockUserEntity));

    when(userRepository.countByEmail(mockUser.getEmail())).thenReturn(0);

    assertThrows(requestDeniedException.class, () -> userService.newTrainer(mockUser));

    verify(userRepository, times(1)).findById(mockUser.getUsername());

    verify(userRepository, times(1)).countByEmail(mockUser.getEmail());

}
```

Ispitni slučaj 6: Neuspješno pronalaženja korisnika po ID-u

```
@Test
public void testFindingUserByIdUnsuccessfully() {

    User mockUser = UserGeneratingUtil.createMockUser();

    UserEntity mockUserEntity = UserEntity.from(mockUser);

    String mockUsername= "stella";

    when(userRepository.findById(mockUsername)).thenReturn(Optional.empty());

    UserEntity found = userService.findById(mockUsername);

    verify(userRepository, times(1)).findById(mockUsername);

    Assert.assertEquals(found, null);

}
```

✓ UserServiceTest (hr.fer.testing.service)	1 sec 805 ms
✓ testFindingUserByldUnsuccessfully	1 sec 787 ms
✓ testAddingNewTrainerUnsuccessfully	17 ms
✓ testListingAllNonTrainers	1 ms

Slika 5.3: Rezultati ispitnih slučajeva 4, 5 i 6

5.2.2 Ispitivanje sustava

Ispitni slučaj 1: Uspješan pristup kalendaru

Ulaz:

1. Otvaranje početne stranice web aplikacije
2. pritisak na gumb "login"
3. unošenje podataka (korisnik posjeduje postojeći račun)
4. redirect na stranicu kalendar

Izlaz:

1. prikazuje se početna stranica
2. gumb nas vodi na stranicu login
3. uspješna prijava
4. prikaz stranice kalendar

Rezultati: Aplikacija je uspješno izvela sve korake testa. **Aplikacija je prošla test.**

```

testipy > -
1 from time import sleep
2 from selenium import webdriver
3 from selenium.common.exceptions import NoSuchElementException
4 from selenium.webdriver.common.by import By
5 from webdriver_manager.chrome import ChromeDriverManager
6 from selenium.webdriver.support.ui import WebDriverWait
7 from selenium.webdriver.support import expected_conditions as ec
8
9 options = webdriver.ChromeOptions()
10 options.add_argument([
11     '--trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org'])
12 browser = webdriver.Chrome(ChromeDriverManager().install())
13 browser.implicitly_wait(10)
14 browser.get('https://localhost:3000/')
15 sleep(20)
16 browser.get('https://www.bourbontech.onrender.com/login')
17 wait = WebDriverWait(browser, 300)
18
19 username= "marko"
20 password="1234az"
21 browser.find_element(By.ID, "username").send_keys(username)
22 browser.find_element(By.ID, "password").send_keys(password)
23 sleep(3)
24 browser.find_element(By.XPATH, "//button").click()
25 sleep(3)
26 assertEquals(browser.current_url, 'https://www.bourbontech.onrender.com/calendar')
27 browser.close()

```

Slika 5.4: Izvorni kod ispitnog slučaja 1

Ispitni slučaj 2: Neuspješan pristup kalendaru

Ulaz:

1. Otvaranje početne stranice web aplikacije
2. pritisak na gumb "login"
3. unošenje podataka (korisnik ne posjeduje postojeći račun)
4. redirect na stranicu neuspjele prijave

Izlaz:

1. prikazuje se početna stranica
2. gumb nas vodi na stranicu login
3. uneseni podaci ne odgovaraju ni jednom postojećem računu
4. prikaz stranice neuspjele prijave

Rezultati: Sve komponente testa su uspješno zadovoljene od strane aplikacije. Aplikacija je prošla test.

```
test2.py > ...
1 from time import sleep
2 from selenium import webdriver
3 from selenium.common.exceptions import NoSuchElementException
4 from selenium.webdriver.common.by import By
5 from webdriver_manager.chrome import ChromeDriverManager
6 from selenium.webdriver.support.ui import WebDriverWait
7 from selenium.webdriver.support import expected_conditions as ec
8
9 options = webdriver.ChromeOptions()
10 options.add_argument(
11     '--trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org')
12 browser = webdriver.Chrome(ChromeDriverManager().install())
13 browser.implicitly_wait(10)
14 browser.get('https://bourbontech.onrender.com/')
15 sleep(3)
16 browser.get('https://bourbontech.onrender.com/login')
17 sleep(3)
18 username="luka"
19 password="1234az"
20 browser.find_element(By.ID, "username").send_keys(username)
21 browser.find_element(By.ID, "password").send_keys(password)
22 sleep(3)
23 browser.find_element(By.XPATH, "//button").click()
24 element = browser.find_element(By.XPATH, "//*[contains(text(), 'Please')]")
25 assertNull(browser.element)
26 browser.close()
```

Slika 5.5: Izvorni kod ispitnog slučaja 2

Ispitni slučaj 3: Promjena lozinke**Ulaz:**

1. Otvaranje početne stranice web aplikacije
2. pritisak na gumb "login"
3. unošenje podataka (korisnik posjeduje postojeći račun)
4. redirect na stranicu kalendara
5. otvaranje stranice profila
6. pritisak na gumb "change password"
7. unos stare pa zatim nove lozinke

8. pritisak na gumb "save"
9. pritisak na gumb "logout"
10. pritisak na gumb "login"
11. prijava s promjenjenim podacima
12. redirect na stranicu kalendara

Izlaz:

1. prikazuje se početna stranica
2. gumb nas vodi na stranicu login
3. uspješna prijava
4. prikaz stranice kalendara
5. prikaz stranice profila
6. prikaz forme za promjenu lozinke
7. nova lozinka spremljena
8. korisnik odjavljen
9. gumb nas vodi na stranicu login
10. uspješna prijava
11. prikaz stranice kalendara

Rezultati: Aplikacija je pravilno i efikasno izvela zahtjeve testa. **Aplikacija je prošla test.**

```
testipy > ...
1 from time import sleep
2 from selenium import webdriver
3 from selenium.webdriver.common.by import By
4 from selenium.webdriver.remote.webelement import WebElement
5 from webdriver_manager.chrome import ChromeDriverManager
6 from selenium.webdriver.support.ui import WebDriverWait
7 from selenium.webdriver.support import expected_conditions as ec
8
9 options = webdriver.ChromeOptions()
10 options.add_argument(
11     '--trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org')
12 browser = webdriver.Chrome(ChromeDriverManager().install())
13 browser.implicitly_wait(10)
14 browser.get("https://bourbontech.onrender.com/")
15 sleep(3)
16 browser.get("https://bourbontech.onrender.com/login")
17 username = "stella"
18 oldPassword = "1234az"
19 newPassword = "4321az"
20 browser.find_element(By.ID, "username").send_keys(username)
21 sleep(3)
22 browser.find_element(By.ID, "password").send_keys(oldPassword)
23 sleep(3)
24 browser.find_element(By.XPATH, "//button").click()
25 sleep(3)
26 browser.find_element(By.XPATH, "//*[contains(text(), 'CHANGE CURRENT PASSWORD')]").click()
27 sleep(1)
28 browser.find_element(By.XPATH, "//*[@placeholder='Enter your old password']").send_keys(oldPassword)
29 browser.find_element(By.XPATH, "//*[@placeholder='Set your new password']").send_keys(newPassword)
30 browser.find_element(By.XPATH, "//*[contains(text(), 'SAVE NEW PASSWORD')]").click()
31 sleep(3)
32 browser.get("https://bourbontech.onrender.com/profilepage")
33 sleep(3)
34 browser.find_element(By.XPATH, "//*[contains(text(), 'Log out')]").click()
35 sleep(3)
36 browser.get("https://bourbontech.onrender.com/login")
37 sleep(3)
38 browser.find_element(By.ID, "username").send_keys(username)
39 sleep(3)
40 browser.find_element(By.ID, "password").send_keys(newPassword)
41 sleep(3)
42 browser.find_element(By.XPATH, "//button").click()
43 sleep(3)
44 assertEqual(browser.current_url, "https://www.bourbontech.onrender.com/calendar")
45 browser.close()
```

Slika 5.6: Izvorni kod ispitnog slučaja 3

Ispitni slučaj 4: Rezervacija treninga

Ulaz:

1. Otvaranje početne stranice web aplikacije
2. pritisak na gumb "login"
3. unošenje podataka (korisnik posjeduje postojeći račun)
4. redirect na stranicu kalendara
5. klik na gumb "reservation"

Izlaz:

1. prikazuje se početna stranica
2. gumb nas vodi na stranicu login
3. uspješna prijava
4. prikaz stranice kalendara
5. smanjivanje fonda sati za jedan

Rezultati: Testiranje aplikacije je završilo uspješno. Aplikacija je prošla test.

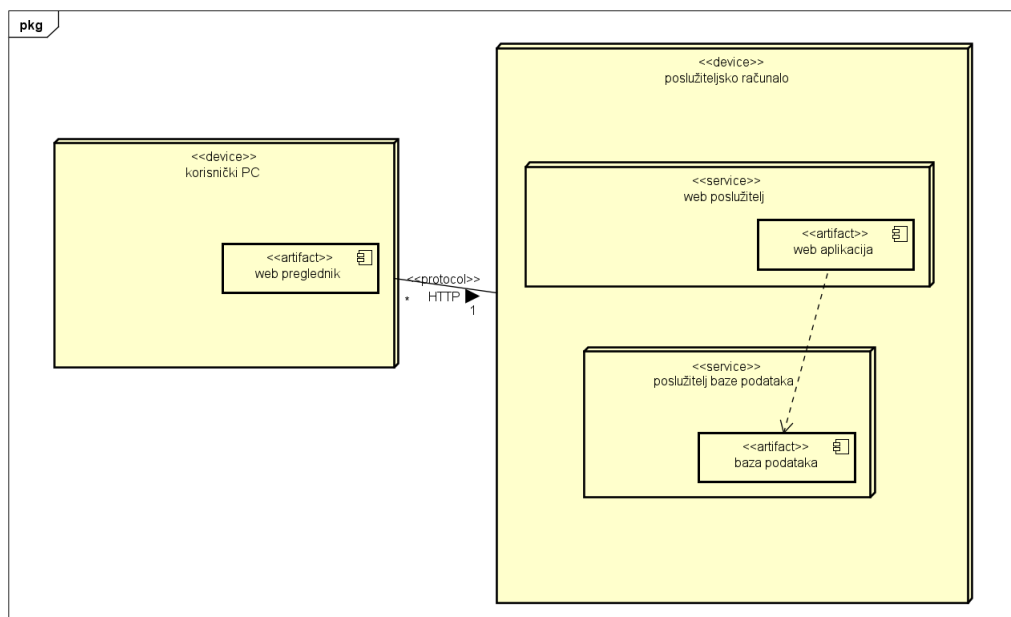
```
from time import sleep
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec

options = webdriver.ChromeOptions()
options.add_argument(
    '--trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org')
browser = webdriver.Chrome(ChromeDriverManager().install())
browser.implicitly_wait(10)
browser.get('https://burbontech.onrender.com/')
sleep(3)
browser.get('https://burbontech.onrender.com/login')
sleep(3)
username = "marko"
password = "1234aZ"
browser.find_element(By.ID, "username").send_keys(username)
browser.find_element(By.ID, "password").send_keys(password)
sleep(3)
browser.find_element(By.XPATH, "//button").click()
sleep(3)
trainings_left_before = browser.find_element(By.XPATH, "//*[contains(text(), 'trainings left')]").text
num = ""
for c in trainings_left_before:
    if c.isdigit():
        num = num + c
trainings_left_before = int(num)
VerticalAppointment-textContainer
browser.find_element(By.XPATH, "//*[[@class='VerticalAppointment-textContainer']]").click()
sleep(1)
browser.find_element(By.XPATH,
    "//*[[@class='MuiButtonBase-root MuiButton-root MuiButton-contained MuiButton-containedSuccess MuiButton-sizeLarge MuiButton-containedSizeLarge MuiButton-root MuiButton-contained MuiBut
    ").click()
trainings_left_after = browser.find_element(By.XPATH, "//*[contains(text(), 'trainings left')]").text
num = ""
for c in trainings_left_after:
    if c.isdigit():
        num = num + c
trainings_left_after = int(num)
assertEqual(trainings_left_before-1, trainings_left_after)
browser.close()
```

Slika 5.7: Izvorni kod ispitnog slučaja 4

5.3 Dijagram razmjestaja

Dijagrami razmjestaja su strukturni statički UML dijagrami koji opisuju topologiju sustava i usredotočeni su na odnos sklopovskih i programskih dijelova. Na strani poslužitelja se nalazi web poslužitelj te poslužitelj baze podataka. Dok se na strani klijenta nalazi web preglednik kojim korisnik pristupa aplikaciji. Sustav je koncipiran na arhitekturi "klijent-poslužitelj" te se komunikacija odvija putem HTTP veze.



Slika 5.8: dijagram razmještaja

5.4 Upute za puštanje u pogon

Priprema backend-a za deploy na Render

Po potrebi dodati env. varijable u run konfiguraciju vašeg IDE-a. Zatim dodati Dockerfile uz napomenu da u direktoriju docker postoje dvije verzije, za Maven i Gradle. Nadalje ukoliko se mijenja lokacija DockerFile-a pripaziti na putanju unutra "COPY" naredbi u DockerFile skripti. U application.properties postaviti property server.servlet.context-path na /api kao prefiks svim zahtjevima na backend. .

Priprema frontend-a za deploy na Render

U package.json potrebno je dodati dependency-e neophodne za deploy to jest primarno http-proxy-middleware, dontev te express. Zatim potrebno je dodati /src/setupProxy.js koji služi kao proxy server za lokalni development (preusmjerava api pozive na localhost:8080) odnosno kada se koristi "react-scripts start" skripta. Nadalje potrebno je dodati app.js u kojem se nalazi express server za produkcijski proxy te posluživanje frontend-a. Također potrebno je u package.json izmijeniti

```
"build": "yarn install && react-scripts build"
```

te dodati

```
"start-prod": "node app.js"
```

.

Deploy

Kreiranje baze podataka

Potrebno je u Render dashboard-u izvesti sljedeće komande :

- New ⇒ PostgreSQL
- Postaviti ime baze i opcionalno username za korisnika baze (password je automatski generiran)
- Region Frankfurt
- Create Database
- Free plan baza podataka ima max pohranu od 1GB

.

Kreiranje backend-a

Potrebno je u Render dashboard-u izvesti sljedeće komande :

- New ⇒ Web Service

- Povezati Gitlab racun, nakon čega su za odaabir dostupni svi projekti na koje imate prava pristupa
- Stisnuti connect pored odgovarajućeg projekta
- Postaviti ime za servis(postaje dio web adrese)
- Rot directory postaviti na prog-i-be
- Environment Docker
- Region Frankfurt
- Na dnu proširiti advanced
- Dodati potrebne environment varijable te kopirati vrijednosti iz postavki baze podataka na Renderu.
- Ako je dodan Spring Boot Actuator postaviti /api/actuator/health kao Healt Check Path
- Postaviti putanju za DOckerfile ovisno o pacage menageru
- Stisnuti Create Web Service

Kreiranje frontend-a

Potrebno je u Render dashboard-u izvesti sljedeće komande :

- New ⇒ Web Service
- Povezati Gitlab racun, nakon čega su za odaabir dostupni svi projekti na koje imate prava pristupa
- Stisnuti connect pored odgovarajućeg projekta
- Postaviti ime za servis(postaje dio web adrese)
- Rot directory postaviti na prog-i-fe
- Environment Node
- Region Frankfurt
- Build command postaviti na yarn build, a Start Command yarn start-prod
- Na dnu proširiti advanced
- Dodati potrebne environment varijable -API_BASE_URL postaviti na adresu deployanog backend-a aplikacije dostupne na Render dashboardu
- Stisnuti Create Web Service

6. Zaključak i budući rad

Naš projektni zadatak bio je razvoj web aplikacije za rezervaciju treninga uz dodatak personalizirane ponude treninga ovisno o odabranim ciljevima korisnika. Provedba projekta bila je podijeljena u dvije faze.

Prva faza projekta odvijala se u prvom ciklusu semestra. Glavni fokus bio je na okupljanju tima, raspodjeli poslova te dokumentaciji projekta. Prva revizija dokumentacija temeljila se na opisu projektnog zadatka, obrascima uporabe, dijagramima obrazaca uporabe, sekvencijskim dijagramima i dijagramima razreda. Definiranje funkcionalnih zahtjeva uvelike je olakšalo daljnji rad na aplikaciji. Također, izrada vizualnih prikaza idejnih rješenja aplikacije pomogla je svim članovima tima kod implementacije.

Druga faza projekta odvijala se u drugom ciklusu semestra te je glavni fokus bio na programskom dijelu aplikacije. U odnosu na prvu fazu, članovi tima puno su više radili samostalno te su sastanci bili puno rjeđi. Osim realizacije same aplikacije, u drugoj fazi potrebno je bilo dokumentirati ostatak UML dijagrama i provesti ispitivanje programskog rješenja.

Jedina funkcionalnost koju nismo implementirali u potpunosti je ta da treneri unose pravila koje se odnose na rezervaciju termina svih korisnika. Naime, odlučili smo kako nam je puno jednostavnije da su ta pravila već određena u aplikaciji, a ne da ih trener sam unosi.

Sudjelovanje na ovom projektu svim članovima tima bilo je jako korisno iskustvo. Imali smo priliku raditi u timu te smo na taj način naučili surađivati s drugim članovima i zajednički dolaziti do optimalnih rješenja. Zaključili smo da je najvažnija dobra komunikacija među članovima. Komunikacija među članovima tima uglavnom se odvijala preko Whatsappa te su tako svi članovi tima bili informirani o napretku projekta. Smatramo da smo vrlo dobro odradili zadatak, iako postoji prostora za napredak.

Popis literature

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

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Baze podataka, FER https://www.fer.unizg.hr/predmet/bazpod_c
3. Procesi programskog inženjerstva, FER [https://www.fer.unizg.hr/_download/repository/Procesi_programskog_inzenjerstva_3_izdanje\[1\].pdf](https://www.fer.unizg.hr/_download/repository/Procesi_programskog_inzenjerstva_3_izdanje[1].pdf)

Indeks slika i dijagrama

2.1	"Adidas Sports Studio" web aplikacija	4
2.2	"Adidas Sports Studio" web aplikacija	4
2.3	"Sparta Gym" web aplikacija	5
3.1	"Dijagram funkcionalnosti neregistriranog korisnika i korisnika treninga"	12
3.2	"Dijagram funkcionalnosti korisnika treninga i trenera"	13
3.3	"Dijagram funkcionalnosti korisnika aplikacije i administratora"	13
3.4	"Registracija"	14
3.5	"Rezervacija termina treninga"	15
3.6	"Unos treninga"	16
3.7	"Odjava"	17
4.1	Povezanost slojeva u radnom okviru Spring	19
4.2	dijagram baze podataka	22
4.3	dijagram razreda - Dijagram kontrolera	23
4.4	dijagram razreda - Dijagram modela	23
4.5	dijagram razreda - Dijagram repozitorija	24
4.6	dijagram razreda - dijagram entiteta	24
4.7	dijagram razreda - dijagram servisa	25
4.8	Dijagram stanja - Korisnik treninga	26
4.9	Dijagram aktivnosti - Rezervacija treninga	27
4.10	Dijagram komponenti	28
5.1	Rezultati ispitnih slučajeva 1 i 2	30
5.2	Rezultat ispitnog slučaja 3	31
5.3	Rezultati ispitnih slučajeva 4, 5 i 6	33
5.4	Izvorni kod ispitnog slučaja 1	34
5.5	Izvorni kod ispitnog slučaja 2	35
5.6	Izvorni kod ispitnog slučaja 3	37
5.7	Izvorni kod ispitnog slučaja 4	39
5.8	dijagram razmjesta	40

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 20.10.2022.
- Prisustvovali: Luka Vukelić, Stella Balić, Jelena Kulišić, Matko Nikolić, Barbara Pašalić, Tin Pavletić, Jure Rajčić
- Teme sastanka:
 - sastanak s asistentom i demonstratorom
 - upoznavanje s temom projekta

2. sastanak

- Datum: 26.10.2022.
- Prisustvovali: Luka Vukelić, Stella Balić, Jelena Kulišić, Matko Nikolić, Barbara Pašalić, Tin Pavletić, Jure Rajčić
- Teme sastanka:
 - raspodjela poslova po članovima tema
 - detaljna razrada teme

3. sastanak

- Datum: 5.11.2022.
- Prisustvovali: Luka Vukelić, Stella Balić, Jelena Kulišić, Matko Nikolić, Barbara Pašalić, Tin Pavletić, Jure Rajčić
- Teme sastanka:
 - komentari dosadašnjeg rada
 - daljnja podjela rada

4. sastanak

- Datum: 10.11.2022.
- Prisustvovali: Luka Vukelić, Stella Balić, Jelena Kulišić, Matko Nikolić, Barbara Pašalić, Tin Pavletić, Jure Rajčić
- Teme sastanka:
 - dogovor za popravak dokumentacije
 - dizajn stranice, komunikacija backenda i frontenda, baza podataka

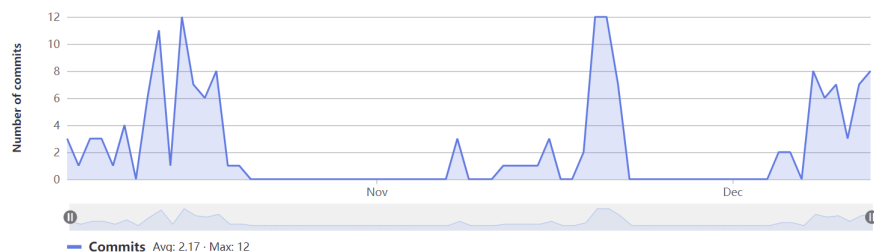
Tablica aktivnosti

	Luka Vukelić	Stella Balić	Jelena Kulišić	Matko Nikolić	Barbara Pašalić	Tin Pavletić	Jure Rajčić
Upravljanje projektom	5						
Opis projektnog zadatka			7				
Funkcionalni zahtjevi	5						
Opis pojedinih obrazaca	5						
Dijagram obrazaca	3					3	
Sekvencijski dijagrami	3					5	
Opis ostalih zahtjeva			1				
Arhitektura i dizajn sustava			4				
Baza podataka		3	3			3	
Dijagram razreda						7	
Dijagram stanja			7				
Dijagram aktivnosti			7				
Dijagram komponenti			6				
Korištene tehnologije i alati						5	
Ispitivanje programskog rješenja						10	
Dijagram razmještaja						5	
Upute za puštanje u pogon						3	
Dnevnik sastajanja			1				
Zaključak i budući rad			1				
Popis literature			1				
Izrada početne stranice					5		
Back end		40					40
Front end	25			40	35		

Dijagrami pregleda promjena

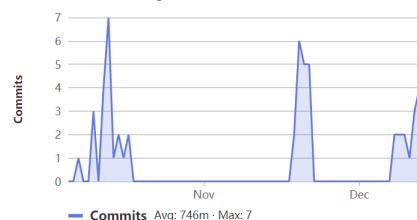
Commits to main

Excluding merge commits. Limited to 6,000 commits.



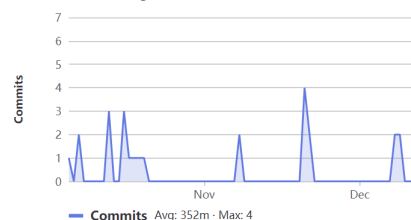
stella_balic1405

53 commits (stella.balic@fer.hr)



Luka Vukelić

25 commits (lv53072@fer.hr)



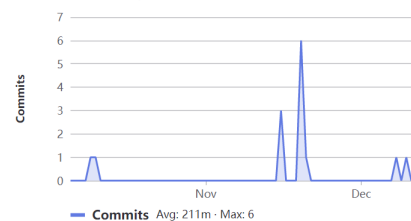
jelenakuliscic

22 commits (jelena.kuliscic@fer.hr)



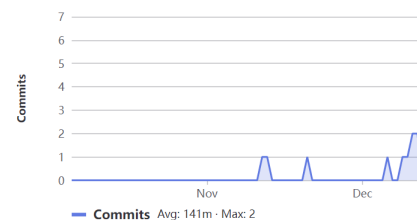
unknown

15 commits (barbara.pasalic@fer.hr)



matkonikolic

10 commits (matko.nikolic@fer.hr)



Tinko55

9 commits (tin.pavletic@fer.hr)

