

Programsko inženjerstvo

Ak. god. 2022./2023.

Šahisti

Dokumentacija, Rev. 2

Grupa: *AlmostByte*

Voditelj: *Ivan Bilobrk*

Datum predaje: 13. 1. 2023.

Nastavnik: *Igor Stančin*

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	20
3.2 Ostali zahtjevi	26
4 Arhitektura i dizajn sustava	27
4.1 Opis arhitekture sustava	27
4.2 Baza podataka	28
4.3 Dijagrami razreda	34
4.4 Dijagram stanja	37
4.5 Dijagram aktivnosti	38
4.6 Dijagram komponenti	39
5 Implementacija i korisničko sučelje	40
5.1 Korištene tehnologije i alati	40
5.2 Ispitivanje programskog rješenja	41
5.2.1 Ispitivanje komponenti	41
5.2.2 Ispitivanje sustava	47
5.3 Dijagram razmještaja	57
5.4 Upute za puštanje u pogon	58
6 Zaključak i budući rad	62
7 Popis literature	64
8 Dodatak: Prikaz aktivnosti grupe	65
8.1 Dnevnik sastajanja	65
8.2 Tablica aktivnosti	68

1 Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Preuzet predložak i izrađena gruba kopija predloška dokumentacije.	Anteo Vukasović	18.10.2022.
0.2	Napisan i dodan opis use caseova.	Anteo Vukasović	29.10.2022.
0.3	Napisani i dodani funkcionalni zahtjevi.	Anteo Vukasović	03.11.2022.
0.4	Dodani use case dijagrami.	Mijo Rajić	05.11.2022.
0.5	Napisan i dodan opis projektnog zadatka.	Anteo Vukasović	06.11.2022.
0.6	Dodani sekvencijski dijagrami te njihovi opisi.	Altea Božić, Tina Jureško i Lara Mahalec	09.11.2022.
0.7	Napisan i dodan opis baze podataka.	Lara Mahalec	11.11.2022.
0.8	Napisan i dodan opis arhitekture sustava.	Anteo Vukasović	17.11.2022.
0.9	Napisani i dodani dijagrami razreda te odgovarajući opisi.	Altea Božić, Tina Jureško i Mijo Rajić	18.11.2022.
0.10	Dodan popis sastanaka sa glavnim temama te ažurirana tablica aktivnosti prvog ciklusa	Anteo Vukasović	18.11.2022.
0.11	Kompletirana tablica promjena dokumentacije za prvi ciklus.	Anteo Vukasović	18.11.2022.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	<i>verificirao cijeli tim</i>	18.11.2022.
1.1	Dijagram stanja	Tina Jureško	11.1.2023.
1.2	Dijagram aktivnosti	Tina Jureško	11.1.2023.
1.3	Dijagram komponenti	Altea Božić	12.1.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.4	Dijagram razmještaja	Altea Božić	12.1.2023.
1.5	Korištene tehnologije i alati	Anteo Vukasović	13.1.2023.
1.6	Ispitivanje programskog rješenja	Lara Mahalec	13.1.2023.
1.7	Upute za puštanje u pogon	Ivan Bilobrk	13.1.2023.
1.8	Zaključak i daljnji rad	Anteo Vukasović	13.1.2023.
1.9	Literatura i ažuriranje poglavlja aktivnosti grupe	Anteo Vukasović	13.1.2023.
2.0	Konačna verzija	<i>verificirao cijeli tim</i>	13.1.2023.

2 Opis projektnog zadatka

Cilj ovog projektnog zadatka je izrada funkcionalne web aplikacije koju će koristiti članovi šahovskog kluba. Ova web aplikacija olakšala bi trenerima u šahovskom klubu organizaciju i vođenje treninga i natjecanja svojih učenika, a učenicima bi pružila platformu na kojoj bi mogli okušati svoja znanja i vještine putem dnevnih taktika. Na taj način svi korisnici mogu doraditi i unaprijediti svoja znanja i vještine u divnoj šahovskoj igri.

Aplikacija je dizajnirana primarno za članove šahovskog kluba te će samo oni imati pristup punoj funkcionalnosti aplikacije. No to ne sprječava sve one zainteresirane da dobiju uvid u svijet šaha putem ove aplikacije.

Prilikom pokretanja same aplikacije, korisnika se vodi na stranicu s 3 opcije:

- Registracija
- Prijava u sustav
- Nastavi kao gost

Ukoliko korisnik želi nastaviti bez ikakve prijave ili registracije te odabere posljednju opciju, vodi ga se na početnu stranicu koja ne nudi sve opcije kao što bi u normalnom načinu rada, no ipak nudi dosta zabavnog sadržaja. Korisnik u ovom načinu rada može:

- Riješiti dnevnu taktiku
- Pročitati novosti i zanimljivosti
- Pregledati rang listu članova

Novosti i zanimljivosti je rubrika koju treneri ureduju po svojoj volji. Koncipirana je kao svojevrsna lista članaka čiji sadržaj bi bile vijesti o radu i rezultatima kluba, vijesti iz svijeta šaha i šahovskih natjecanja diljem svijeta, crtice iz povijesti, analize legendarnih šahovskih partija i slično.

Dnevna taktika je kratka šahovska zagonetka, najčešće od 1 do 5 poteza, u kojoj je cilj naći optimalnu kombinaciju poteza u danom scenariju šahovske igre. Primjer jedne takve zagonetke možemo vidjeti na raznim šahovskim platformama kao što su lichess(link), chess.com(link) i slični. U našoj aplikaciji, dnevna taktika bilježi vrijeme rješavanja, te točnost ponuđenog rješenja. Rješavanjem dnevne taktike, neregistriranog korisnika se obavijesti o ispravnosti njegova rješenja, no njegovo rješenje se nigdje ne pohranjuje, niti taj korisnik sam može biti uvršten u rang listu. Za te funkcionalnosti korisnik mora biti član, a član može postati registracijom.



Slika 2.1: Primjer šahovske zagonetke

Registracija nudi korisnicima koji žele postati članovi šahovskog kluba mogućnost da tu želju ostvare. Odabirom te opcije korisnika se vodi na formular u kojem korisnik unosi svoje vlastite podatke. Ukoliko su svi podatci ispravno uneseni, korisnik se pohranjuje u bazu podataka te postaje punopravni član šahovskog kluba. Čestitke!

Prijava u sustav je slična registraciji, jedina razlika je ta što njoj pristupaju postojeći članovi kluba, stoga moraju unijeti ispravne podatke koji već postoje u bazi podataka šahovskog kluba kako bi im se omogućio pristup aplikaciji.

Ukoliko je korisnik prijavljen u sustav te se radi o članu šahovskog kluba koji nije trener, korisnik u aplikaciji može:

- Pregledati osobne podatke
- Riješiti dnevnu taktiku
- Pročitati novosti i zanimljivosti
- Pregledati rang listu članova
- Prijaviti se na treninge
- Prijaviti se na turnire
- Biti dio rang liste svih članova
- Prijaviti rješenja dnevne taktike
- Platiti članarinu i pregledati svoja prijašnja plaćanja

Treća i četvrta točka prethodne liste se ne razlikuju prema funkcionalnosti od funkcionalnosti koju za iste komponente imaju neregistrirani članovi.

Rješavanje dnevne taktike je isto kao i kod neregistriranih korisnika, no ono što se odvija nakon samog rješavanja jest drugačije. Kod registriranih članova, rješenja se pohranjuju u bazu podataka.

Na temelju toga kreira se rang lista svih članova kluba. Rang lista uzima u obzir sve riješene taktike te 3 parametra tih rješenja; točnost, brzinu i težinu same taktike.

Osim rješavanja dnevne taktike članovi imaju opciju prijavljivanja rješenja. Ukoliko smatraju kako rješenje nije ispravno, korisnici odabiru opciju za prijavu dnevne taktike te obavljaju proces podnošenja prijave koji se sastoji od 3 koraka:

1. Unos novih poteza za koje član smatra da su ispravni
2. Tekstualni opis tih rješenja, iznošenje svojih misli iza priloženih novih rješenja
3. Odabir trenera kojem se šalje zahtjev

Ukoliko trener prihvati novo rješenje, ono se uzima kao ispravno te se rang liste ažuriraju prema ovim rješenjima.

Član se unutar naše aplikacije može prijavljivati na treninge i turnire objavljene na aplikaciju od strane trenera. Odabirom opcije za prijavu na turnir/trening, član odlazi na stranicu gdje vidi popis svih dostupnih treninga i turnira. Ukoliko nema nikakvih konflikata ili prekršenih ograničenja o maksimalnom broju dopuštenih članova u treningu/turniru, člana se prijavljuje na dani trening/turnir.

Odabirom opcije za pregled osobnih podataka, član na jednom mjestu vidi svoje osnovne osobne podatke koje je unio u trenutku registracije te svu svoju aktivnost na aplikaciji.

Svaki član mora platiti članarinu kako bi mogao i dalje biti punopravni član, a naša aplikacija nudi opciju plaćanja. Odabirom opcije za plaćanje članarine, člana se vodi na popis svih svojih računa. Ukoliko na tom popisu ima neplaćenih, člana se o tome obavijesti te odabirom pojedinog neplaćenog računa korisnik može platiti taj račun unošenjem osnovnih podataka potrebnih za plaćanje. Za člana je bitno da uredno plaća svoje račune jer je moguće da mu administrator(više o njemu kasnije) ograniči pristup aplikaciji sve dok ne plati sve neplaćene članarine.

Osim članova, u šahovskom klubu postoje i treneri. Oni su zaduženi za objavljivanje sadržaja i aktivnosti na aplikaciju koje članovi kasnije mogu koristiti. Treneri na aplikaciji mogu:

- Pregledati osobne podatke
- Objavljivati i ažurirati dnevne taktike
- Objavljivati novosti i zanimljivosti
- Objavljivati treninge
- Organizirati i objaviti turnire
- Revidirati zaprimljene prijave o neispravnim rješenjima dnevnih taktika

Prva točka je identična kao kod članova, iste funkcionalnosti imaju i jedni i drugi.

Treneri mogu objavljivati novosti i zanimljivosti na aplikaciju te postavljati nove dnevne taktike. Kod svake od navedenih rubrika u aplikaciji, za trenere će biti ponuđena opcija dodavanja novog sadržaja. Primjerice, kod unosa novosti, treneru će se otvoriti prostor za unošenje teksta u kojem trener upisuje što želi. Nakon što je gotov, trener potvrdi svoju namjeru objavljivanja odabirom prikladnog gumba te se taj tekst dodaje u aplikaciju.

Treninge i turnire trener dodaje odabirom opcije za dodavanje novog treninga/turnira. Odabire termin početka i vrijeme trajanja te potencijalno ograničenje maksimalnog broja sudionika. Ukoliko su svi kriteriji zadovoljeni, aktivnost se dodaje u bazu podataka i prikazuje se unutar aplikacije. Ukoliko termin nije ispravan te dolazi do kolizije s nekom drugom aktivnosti, trenera se vraća na ponovni odabir ispravnog termina.

U prijašnjem dijelu teksta smo vidjeli kako članovi mogu prijavljivati dnevne taktike za koje misle da imaju bolja rješenja od onih koja su ponuđena. Odabirom na opciju pristigle prijave, trener može pregledati imaju li kakvih prijava od članova. Ukoliko imaju, mogu pogledati cijeli tekst prijave i proučiti ponovno dnevnu taktiku. Smatraju li kako je prijava neutemeljena treneri je samo odbacuju. Na ako je prijava valjana, trener može promijeniti rješenje dnevne taktike te ju na taj način ažurirati. Tim činom se ujedno i ažuriraju rang liste.

Osim do sada navedenih korisnika aplikacije, postoji i još jedna vrsta korisnika a to je administrator. Njegove ovlasti su slične kao kod trenera, no još dodatno proširene. On može vidjeti sav sadržaj aplikacije, ali može ga i ukloniti i mijenjati po volji, bez obzira je li on autor tog sadržaja. Uz to, administrator ima opciju zabrane pristupa korisnicima. Ova akcija se može dogoditi iz proizvoljnih razloga no istaknimo jedan poseban scenarij; član uredno ne plaća svoju članarinu. Administrator može vidjeti pregled svih plaćanja svakog člana te ako utvrdi da pojedini član ima previše neplaćenih računa može mu ograničiti pristup aplikaciji. Taj član će od funkcionalnosti jedino moći vidjeti svoja plaćanja i platiti svoje neplaćene račune, i tako sve dok ne plati račune. U tom slučaju, administrator može maknuti restrikcije tom članu.

3 Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici ovog sustava su predsjednik šahovskog kluba, administrator sustava, razvojni tim te članovi kluba

Aktori ovog sustava su neregistrirani korisnici, registrirani članovi, registrirani treneri te administratori kao inicijatori, te baza podataka kao sudionik.

Dionici:

1. Predsjednik kluba (naručitelj)
2. Članovi kluba
 - (a) Treneri
 - (b) Polaznici
3. Administrator
4. Razvojni tim

Aktori:

1. Neregistrirani korisnik(inicijator) može:
 - (a) Obaviti registraciju i postati članom kluba
 - (b) Pregledati novosti postavljene od strane trenera/administratora na aplikaciju
 - (c) Rješavati dnevnu taktiku
 - (d) Vidjeti rang listu uspješnosti rješavanja dnevnih taktika svih članova
2. Registrirani član(inicijator) može:
 - (a) Pregledati svoje osobne podatke
 - (b) Obaviti plaćanje i pregled računa(članarine)
 - (c) Prijaviti se na treninge postavljene na stranicu od strane trenera
 - (d) Prijaviti se na turnire postavljene na stranicu od strane trenera
 - (e) Rješavati dnevne taktike uz pohranjivanje rješenja u bazu podataka te ažuriranje rang liste svih članova na temelju točnosti i brzine rješavanja taktike
 - (f) Prijaviti greške u dnevnoj taktici na način da unese i opiše nove poteze za koje smatra da su pravo rješenje te pošalje svoja zapažanja proizvoljnom treneru na reviziju
3. Trener(inicijator) može:
 - (a) Pregledati svoje osobne podatke
 - (b) Postavljati nove dnevne taktike na aplikaciju
 - (c) Postavljati novosti na aplikaciju
 - (d) Postavljati raspored svojih treninga na aplikaciju te omogućiti članovima prijavu na iste
 - (e) Organizirati turnire za članove te omogućiti članovima prijavu na iste
 - (f) Revidirati zaprimljene prijave pogreški u dnevnim taktikama te ukoliko je potrebno ažurirati ih
4. Administrator(inicijator) može:

- (a) Imati uvid u sve podatke na aplikaciji te opciju uređivanja i dodavanja novih sadržaja
- (b) Zabraniti pristup aplikaciji određenim članovima/trenerima
- (c) Pregledati plaćanja svih članova te ukoliko ima neplaćenih računa određenim članovima zabraniti pristup dok ne obavi plaćanja

5. Baza podataka(sudionik):

- (a) Pohranjuje sve osobne podatke o registriranim članovima
- (b) Pohranjuje sve akcije registriranih članova
- (c) Pohranjuje status o potencijalnoj zabrani pristupa nekom od članova

3.1.1 Obrasci uporabe

Opis obrazaca uporabe:

UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvaranje korisničkog računa za neregistriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Odabir gumba za registraciju
 2. Ispunjavanje ponuđenog formulara
 3. Pohranjivanje podataka u bazu podataka
 4. Otvaranje početne stranice
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o grešci i vraća ga na registraciju
 2. Korisnik ispravlja grešku i uspješno obavlja registraciju ili ne uspijeva dok ne odustane

UC2 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Verifikacija postojanja korisničkog računa i mogućnost pristupa većem broju sadržaja na stranici
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Odabir gumba za prijavu u sustav
 2. Popunjavanje danog formulara
 3. Provjera postojanja danih podataka u samoj bazi podataka
 4. Otvaranje početne stranice
- **Opis mogućih odstupanja:**
 - 2.a Korisniku je onemogućen pristup aplikaciji
 - 2.b Unos neispravnog korisničkog imena ili lozinke
 1. Sustav obavještava korisnika o grešci i vraća ga na prijavu u sustav
 2. Korisnik ispravlja grešku i uspješno obavlja registraciju ili ne uspijeva dok ne odustane

UC3 - Pregled novosti

- **Glavni sudionik:** Korisnik
- **Cilj:** Mogućnost pregledavanja novosti postavljenih na stranicu od strane ovlaštenih korisnika (trener/administrator)

- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik otvaranjem glavne stranice vidi sve novosti postavljene na stranici
 2. Klikom na novost dobiva potpuni pregled same novosti

UC4 - Rješavanje dnevne taktike

- **Glavni sudionik:** Korisnik
- **Cilj:** Otvaranje postavljene dnevne taktike i prijavljivanje rješenja
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Otvaranje dnevne taktike i prikaz prosječne ocjene taktike
 2. Prijava rješenja i njegova pohrana u bazu podataka te ažuriranje rang listi(pohrana i ažuriranje samo ako je korisnik registriran)
 3. Nakon predaje rješenja korisnik dobije obavijest o točnosti rješenja te točno rješenje ukoliko ga sam nije ponudio

UC5 - Pregled rang-liste

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled težinske liste temeljene na točnosti i brzini rješavanja dnevnih taktika. Rang-lista sastavljena je od svih članova.
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik pritiskom na gumb odlazi na stranicu rang-liste
 2. Otvara mu se popis svih članova sortiranih na temelju točnosti i brzini rješavanja dnevnih taktika

UC6 - Pregled osobnih podataka

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire gumb za prikaz osobnih podataka i aktivnosti na stranici
 2. Podatci se prikazuju u aplikaciji

UC7 - Plaćanje članarine

- **Glavni sudionik:** Član
- **Cilj:** Član plaća svoj mjesečni iznos članarine
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav

- **Opis osnovnog tijeka:**

1. Korisnik odabire gumb za plaćanje članarine
2. Na otvorenoj stranici se prikazuju neplaćeni računi(ukoliko ih ima)
3. Korisnik odabere račun te potvrdi plaćanje

UC8 - Prijava na trening

- **Glavni sudionik:** Član

- **Cilj:** Prijavljivanje korisnika na odabrani trening

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav

- **Opis osnovnog tijeka:**

1. Korisnik odabire gumb za prijave na treninge
2. Aplikacija prikazuje popis dostupnih treninga iz baze podataka
3. Korisnik odabire trening i njegov odabir se pohranjuje u bazu podataka

UC9 - Prijava na turnir

- **Glavni sudionik:** Član

- **Cilj:** Prijavljivanje korisnika na odabrani turnir

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav

- **Opis osnovnog tijeka:**

1. Korisnik odabire gumb za prijave na turnire
2. Aplikacija prikazuje popis dostupnih turnira iz baze podataka
3. Korisnik odabire turnir i njegov odabir se pohranjuje u bazu podataka

UC10 - Dodjeljivanje ocjene dnevnoj taktici

- **Glavni sudionik:** Član

- **Cilj:** Dodjeljivanje ocjene dnevnoj taktici od strane člana

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav te je riješio dnevnu taktiku

- **Opis osnovnog tijeka:**

1. Nakon rješavanja dnevne taktike korisniku se otvara opcija ocjenjivanja taktike
2. Korisnik unosi ocjenu koja se zatim pohranjuje u bazu podataka

UC11 - Prijava pogreške

- **Glavni sudionik:** Član

- **Cilj:** Član prijavljuje pogrešku u taktici ukoliko smatra da postoji optimalnije rješenje

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen u sustav te je riješio dnevnu taktiku

- **Opis osnovnog tijeka:**

1. Korisniku se nakon rješavanja taktike nudi opcija prijavljivanja pogreške
2. Korisnik odabire gumb te se otvara forma za prijavljivanje pogreške
3. Nakon ispunjavanja forme, korisnik podnosi zahtjev za promjenom taktike koji se može odobriti ili odbiti

UC12 - Unos novih poteza

- **Glavni sudionik:** Član
- **Cilj:** Korisnik predlaže nove poteze kao rješenje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav te je riješio dnevnu taktiku
- **Opis osnovnog tijeka:**
 1. Nakon odabira prijavljivanja greške korisnik unosi svoje nove poteze kao rješenja

UC13 - Opis novih poteza

- **Glavni sudionik:** Član
- **Cilj:** Korisnik daje obrazloženje predloženog rješenja taktike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav te je riješio dnevnu taktiku
- **Opis osnovnog tijeka:**
 1. Nakon unesenih novih poteza korisnik daje pismeno objašnjenje tih poteza

UC14 - Odabir trenera za revidiranje

- **Glavni sudionik:** Član
- **Cilj:** Korisnik odabire trenera koji će pregledati njegov prijedlog o promjeni taktike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav te je riješio dnevnu taktiku
- **Opis osnovnog tijeka:**
 1. Nakon predanog opisa poteza korisniku se otvara opcija odabira među trenerima u bazi
 2. Korisnik odabire trenera kojem zatim dolazi taj isti zahtjev za revizijom

UC15 - Postavljanje dnevne taktike

- **Glavni sudionik:** Trener
- **Cilj:** Trener postavlja dnevnu taktiku koju ostali korisnici zatim mogu rješavati
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni trener
- **Opis osnovnog tijeka:**
 1. Trener kreira dnevnu taktiku s njenim zadanim rješenjima
 2. Postavljanje taktike na stranicu te njeno pohranjivanje u bazu podataka

UC16 - Slaganje rasporeda vlastitih treninga

- **Glavni sudionik:** Trener
- **Cilj:** Trener odabire termine treninga i postavlja ih na aplikaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni trener
- **Opis osnovnog tijeka:**
 1. Trener odabire opciju postavljanja novog treninga
 2. Trener postavlja vrijeme početka i trajanja samog treninga, te njegov kratki opis

3. Nakon što je termin potvrđen kao slobodan, trening se postavlja na aplikaciju i pohranjuje u bazu podataka

- **Opis mogućih odstupanja:**

- 2.a Termin treninga je u preklapanju s drugim aktivnostima

1. Trenera se vraća na ponovno postavljanje termina dok ne unese dostupni termin ili odustane od postavljanja treninga

UC17 - Organiziranje turnira

- **Glavni sudionik:** Trener
- **Cilj:** Trener odabire osnovne parametre turnira i postavlja ga na aplikaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni trener
- **Opis osnovnog tijeka:**
 1. Trener odabire opciju postavljanja novog termina
 2. Trener postavlja vrijeme početka i trajanja turnira, njegov opis i maksimalni broj sudionika
 3. Nakon što je termin potvrđen kao slobodan, turnir se postavlja na aplikaciju i pohranjuje u bazu podataka

UC18 - Objavljivanje novosti

- **Glavni sudionik:** Trener
- **Cilj:** Trener postavlja novosti/zanimljivosti/crtice iz povijesti na aplikaciju drugim korisnicima na čitanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni trener
- **Opis osnovnog tijeka:**
 1. Trener odabire opciju dodavanja novosti na aplikaciji
 2. Trener unosi novost te potvrđuje svoje postavljanje
 3. Novost se objavljuje na aplikaciji te pohranjuje u bazu podataka

UC19 - Revidiranje pogreški

- **Glavni sudionik:** Trener
- **Cilj:** Trener nakon zaprimanja zahtjeva za revizijom dnevne taktike, pregledava taktiku i donosi odluku o potencijalnoj promjeni
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni trener te ima pristigli zahtjev za revizijom taktika
- **Opis osnovnog tijeka:**
 1. Trener odabire gumb za pristigle zahtjeve te prelazi na novu stranicu
 2. Trener odabire jedan od pristiglih zahtjeva te mu se otvara cijeli sadržaj zahtjeva
 3. Nakon revizije trener odbacuje zahtjev ili ga prihvaća

UC20 - Mijenjanje dnevne taktike

- **Glavni sudionik:** Trener

- **Cilj:** Trener zaključuje da je zahtjev ispravan te mijenja dnevnu taktiku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni trener te je obavio reviziju taktike
- **Opis osnovnog tijeka:**
 1. Nakon potvrde da se radi o valjanom zahtjevu, trener mijenja rješenja taktike
 2. Ažurirana taktika se objavljuje na aplikaciju i pohranjuje u bazu podataka
 3. Ažuriraju se rang-liste na temelju novih rješenja

UC21 - Dodavanje svih sadržaja

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator može dodavati sve sadržaje na aplikaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni administrator
- **Opis osnovnog tijeka:**
 1. Administrator ima opciju dodavati bilo koji od prije navedenih sadržaja
 2. Administratoru su kod svih prije navedenih sadržaja dostupne opcije za dodavanje koje on može koristiti kao i prije navedeni korisnici

UC22 - Brisanje svih sadržaja

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator može ukloniti bilo kakav sadržaj sa stranice
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni administrator
- **Opis osnovnog tijeka:**
 1. Administrator kod bilo kojeg sadržaja vidi opciju za uklanjanje istog
 2. Ukoliko odabere opciju uklanjanja, sadržaj se miče sa prikaza aplikacije, no ostaje pohranjen u bazi podataka

UC23 - Zabrana pristupa

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator može zabraniti pristup bilo kojem registriranom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni administrator
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za zabranu pristupa
 2. Uz prikazani popis zabranjenih korisnika, administrator može dodati novog korisnika pritiskom na gumb za dodavanje novih osoba u popis
 3. Pritiskom na gumb otvori se formular u kojeg administrator unosi osobne podatke korisnika kojem želi zabraniti pristup
 4. Ukoliko uneseni korisnik zapravo postoji, njegovo ime se dodaje na popis zabranjenih korisnika u bazi podataka te mu se zabranjuje pristup stranici
- **Opis mogućih odstupanja:**
 - 3.a Administrator je unio korisničke podatke koji ne postoje u bazi svih korisnika

1. Administratora se vraća na ponovni upis podataka u formular sve dok ne unese postojeće podatke ili dok ne odustane

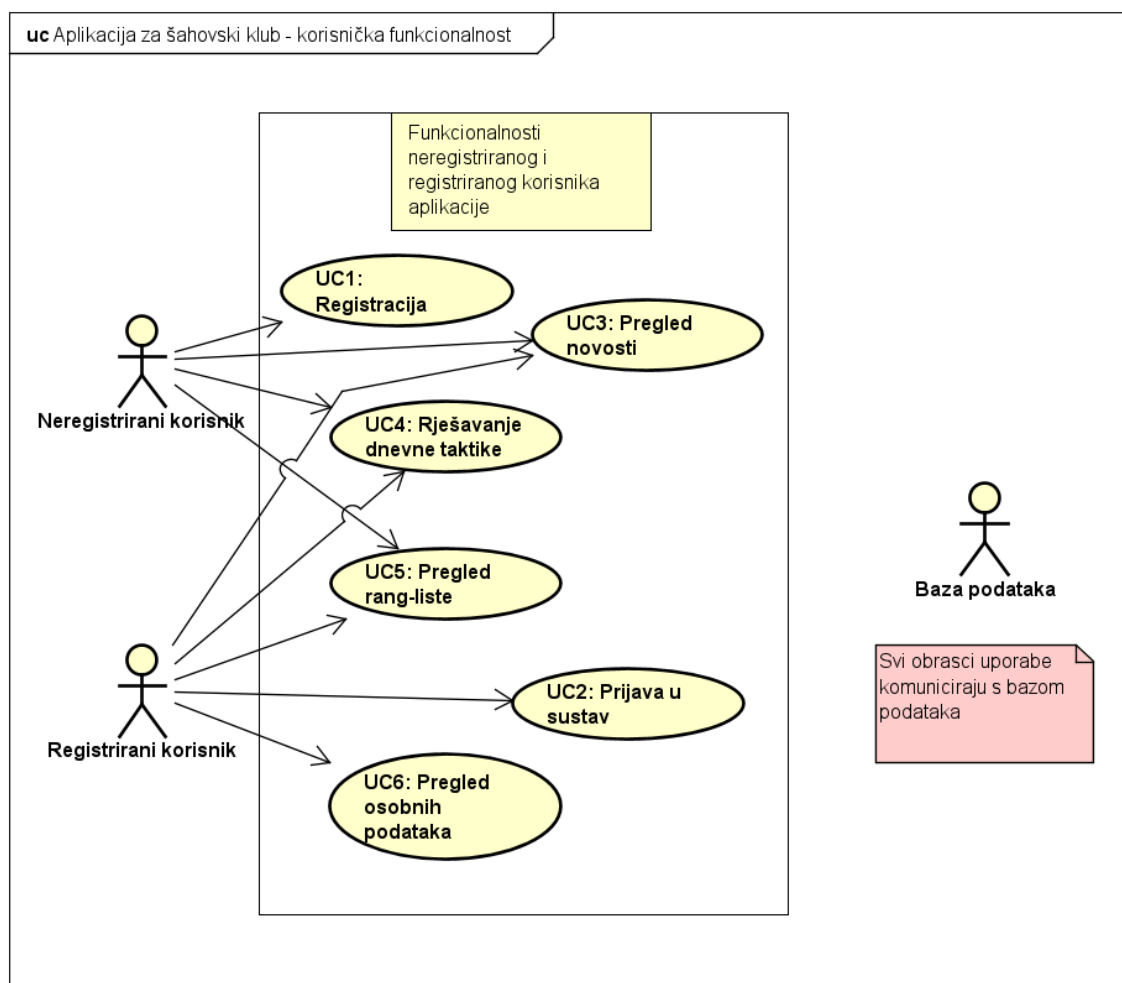
UC24 - Pregled svih plaćanja

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator pregledava sva plaćanja članarina svih članova
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljeni administrator
- **Opis osnovnog tijeka:**
 1. Korisnik za svakog člana ima opciju pregleda računa
 2. Odabirom te opcije otvara se popis svih računa te napomena uz svaki račun je li on plaćen ili ne

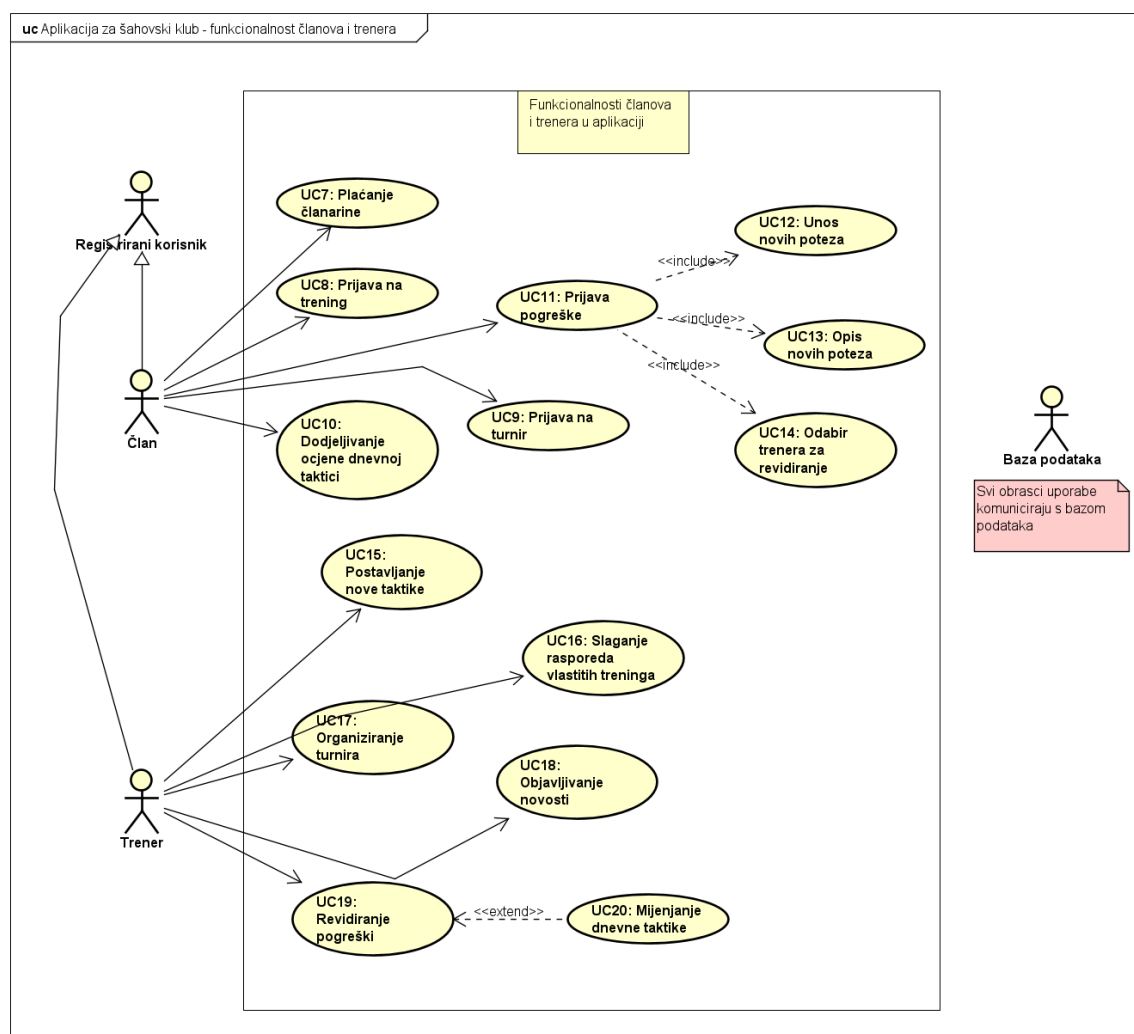
UC25 - Zabrana svih funkcionalnosti na temelju neplaćenih računa

- **Glavni sudionik:** Administrator
- **Cilj:** Administrator zabranjuje funkcionalnosti članovima koji nisu platili članarinu dok ona nije plaćena
- **Sudionici:** Baza podataka, prijavljeni korisnik
- **Preduvjet:** Korisnik je prijavljeni administrator
- **Opis osnovnog tijeka:**
 1. Korisnik otvara pregled plaćanja člana
 2. Ukoliko član nema sve plaćene račune omogućen je gumb za zabranu pristupa člana na temelju neplaćenih računa
 3. Ukoliko je gumb pritisnut, tom članu se dopušta prijava u sustav, no sve funkcionalnosti osim plaćanja računa su mu uskraćene
 4. Nakon plaćanja računa, funkcionalnosti se vraćaju

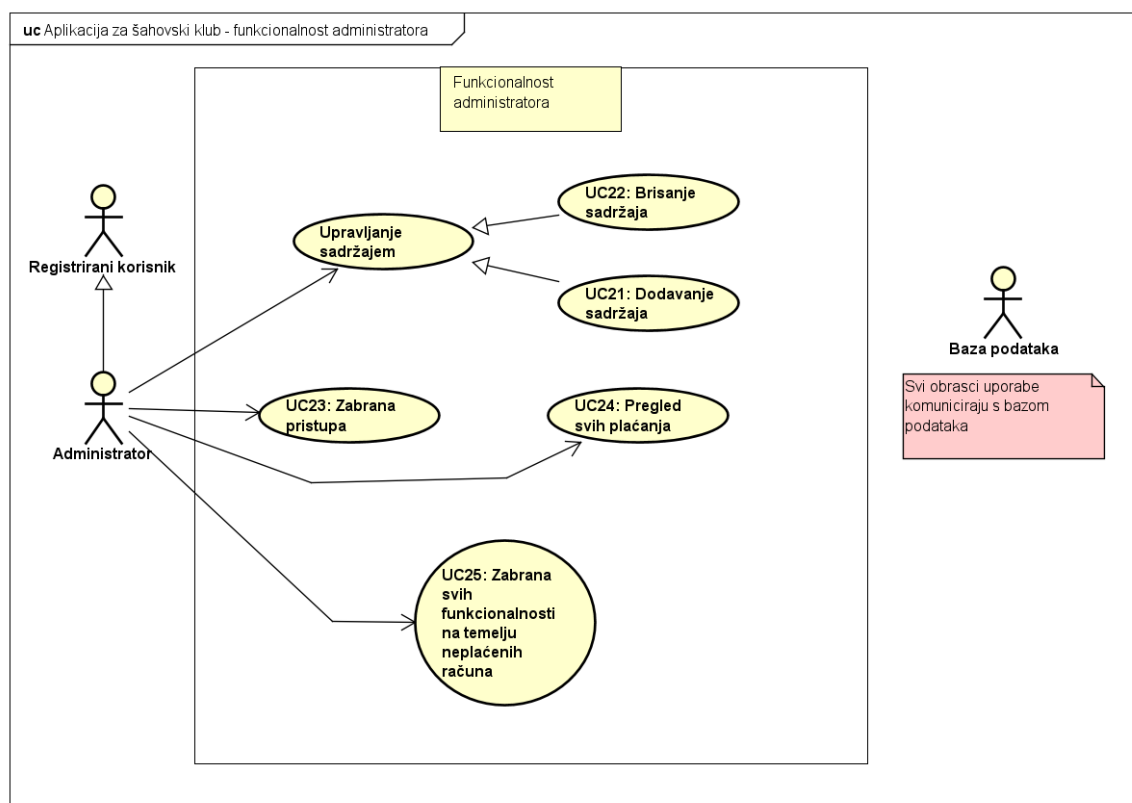
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, korisnička funkcionalnost



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost člana i trenera

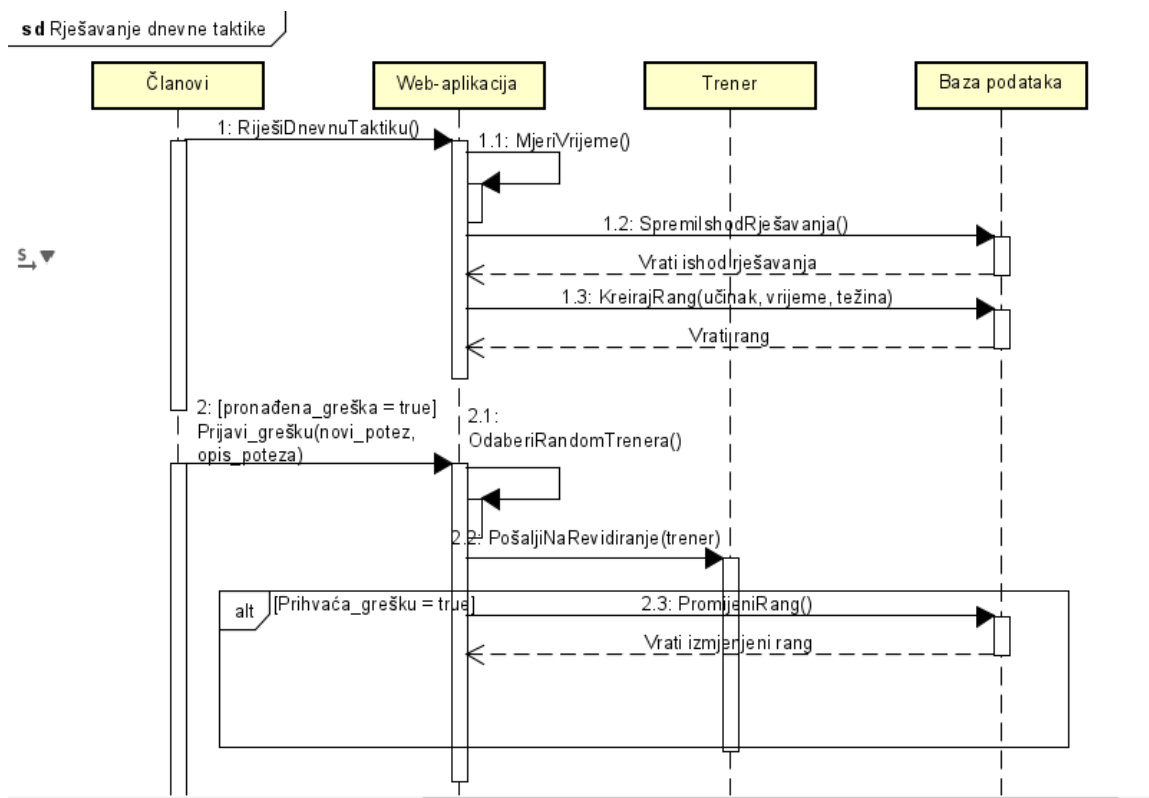


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC4 - Rješavanje dnevne taktike

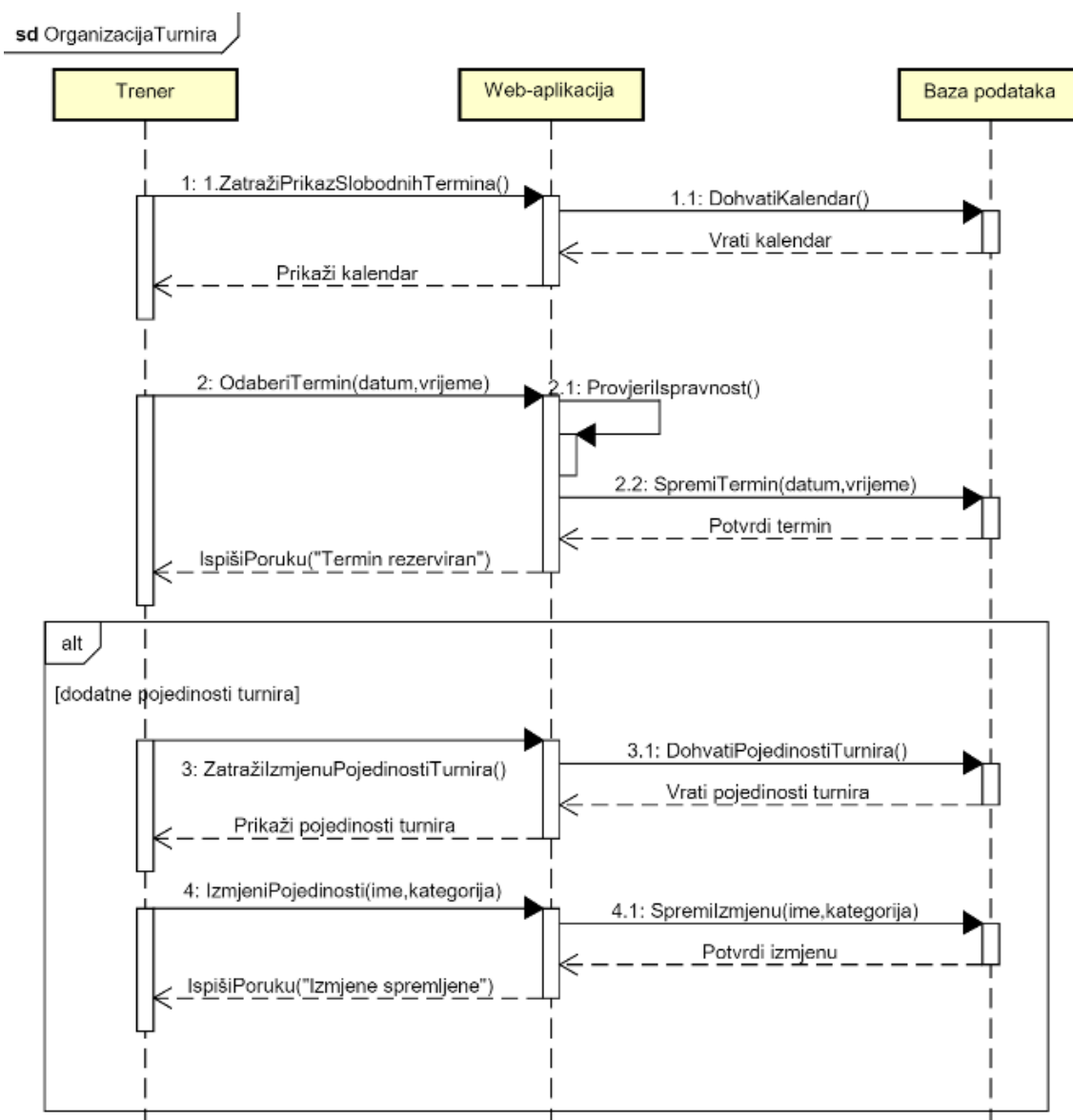
Kad član pokrene rješavanje dnevne taktike, poslužitelj započinje mjeriti vrijeme, a nakon toga u bazu podataka proslijeđuje izmjereno vrijeme odnosno ishod rješavanja dnevne taktike. Aplikacija uz to kreira težinsku rang listu članova na temelju povijesnog učinka u rješavanju taktika, izmjerenog vremena za pojedini član te težine dnevne taktike. Ako član zamijeti pogrešku u dnevnoj taktici to prijavljuje unoseći nove poteze i opis novih poteza, zatim svaka prijavljena pogreška odlazi na revidiranje proizvoljno odabranom treneru. Ako se trener složi s novim rješenjem, ono se mora promijeniti na navedenoj taktici i automatski se moraju revidirati rang liste pri čemu se dodjeljuju bodovi onim članovima koji su ponudili točno rješenje, a uklanjaju onima koji su ponudili staro rješenje.



Slika 3.4: Sekvencijski dijagram za UC4

Obrazac uporabe UC17 - Organiziranje turnira

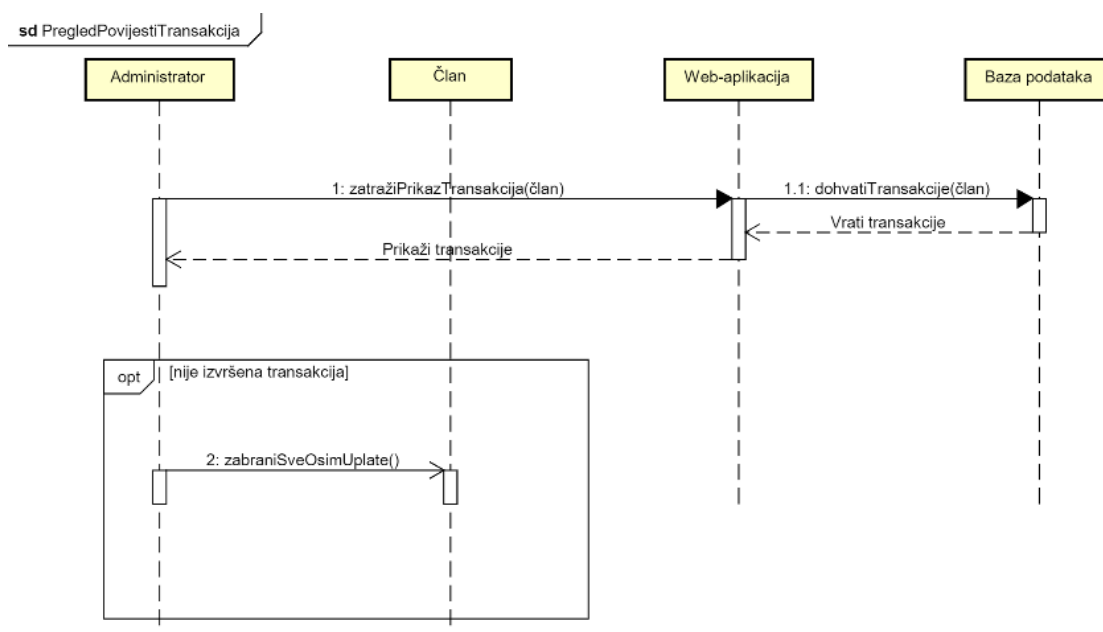
Trener šalje zahtjev za pregled slobodnih termina kako bi mogao odabrati termin kada će se održati turnir. Poslužitelj dohvaća slobodne termine i prikazuje ih. Nakon što trener odabere termin, provjerava se ispravnost te se sprema odabir u bazu podataka nakon čega se prikazuje poruka koja ukazuje na uspješan odabir i rezervaciju termina. Trener može opcionalno promijeniti određene pojedinosti turnira. Ukoliko se na to odluči, trener šalje zahtjev za izmjenu pojedinosti. Poslužitelj dohvaća dosad određene pojedinosti treneru na izmjenu. Nakon što trener izmjeni željene pojedinosti, one se spremaju u bazu podataka. Na kraju se prikazuje poruka koja ukazuje na uspješnu izmjenu pojedinosti.



Slika 3.5: Sekvencijski dijagram za UC17

Obrazac uporabe UC24 - Pregled svih plaćanja

Administrator šalje zahtjev poslužitelju za prikaz transakcija pojedinog člana. Poslužitelj pristupa bazi podataka, dohvaća tražene transakcije i prikazuje ih. Ako transakcije ne postoje ili nisu provedene u cijelosti, administrator može članu zabraniti pristup svim funkcionalnostima aplikacije osim uplate članarine.



Slika 3.6: Sekvencijski dijagram za UC24

3.2 Ostali zahtjevi

1. Sustav mora omogućiti paralelno korištenje aplikacije više korisnika u isto vrijeme bez ikakvih međusobnih interferencija
2. Aplikacija mora biti maksimalno responzivna, korisnik ne smije puno čekati na odgovor ili prikaz sadržaja od strane sustava
3. Sustav treba biti intuitivan i jednostavan za korištenje
4. Neispravno korištenje ne smije rezultirati narušavanjem funkcionalnosti
5. Korisnik treba biti spriječen pristupu određenih dijelova aplikacije ukoliko nije ovlašten
6. Valute unutar sustava prikazane su u Eurima
7. Nadogradnje sustava ne smiju narušavati postojeće funkcionalnosti sustava
8. Pristup aplikaciji mora biti omogućen iz javne mreže
9. Ukoliko je korisniku zabranjen pristup odlukom administratora, ne smije imati nikakve mogućnosti korištenja i treba biti obaviješten o tome, a ukoliko je zabranjen pristup zbog neplaćenih računa, treba mu biti sve ograničeno osim plaćanja računa

4 Arhitektura i dizajn sustava

4.1 Opis arhitekture sustava

Arhitektura sustava može se podijeliti na 3 dijela:

1. Prikaz web aplikacije korisniku (front-end)
2. Web poslužitelj (back-end)
3. Baza podataka

Našem proizvodu pristupa se putem web preglednika. Putem web preglednika, naša web aplikacija napisana u svojim programskim jezicima se prevodi u oblik koji je korisniku jasniji i prirodniji. To korisniku omogućuje korištenje svih funkcionalnosti naše aplikacije.

Web poslužitelj temelj je rada naše aplikacije. On prima zahtjeve poslane od strane korisnika te vraća informacije koje je korisnik zatražio. Također, poslužitelj ima pristup bazi podataka u kojoj su pohranjeni svi podaci relevantni za rad naše aplikacije. Više o bazi podataka možete pročitati u poglavlju 4.2 (Baza podataka). Komunikacija između korisnika i poslužitelja odvija se pomoću HTTP protokola.

Web aplikacija je korisniku prikazana na način da mu se omogući što lakši rad i interakcija s mogućnostima same aplikacije. Odabirom jedne od ponuđenih opcija na stranici, web preglednik šalje zahtjeve za potrebnim podacima i/ili resursima s web poslužitelja/baze podataka na što mu poslužitelj odgovara. Ukoliko nije došlo do nikakvih nepredviđenih poteškoća, korisniku se prikazuje zatraženi sadržaj.

Front-end naše aplikacije razvijen je u JavaScriptu ili točnije Reactu, open-source JavaScript biblioteci koja se koristi za jednostavnije kreiranje korisničkih sučelja. Kao dodatnu pomoć u nekim dijelovima korišten je i CSS.

Back-end naše aplikacije razvijen je u JavaScriptu.

Baza podataka razvijena je u pgAdmin 4, GUI-u razvijen za upravljanje i korištenje PostgreSQL sustavom upravljanja bazi podataka.

Razvojno okruženje korišteno za razvoj back-end i front-end dijela aplikacije je Visual Studio Code. Arhitektura sustava modelirana je prema MVC konceptu. Taj koncept se sastoji od 3 dijela; modela koji kontrolira radom i pravilima aplikacije, prima podatke od controllera te upravlja podacima, viewa koji predstavlja bilo kakav grafički ili drugi prikaz podataka te controllera, komponente koja prima zahtjeve korisnika i proslijeđuje ih drugim dijelovima sustava.

Osim do sada navedenih značajki, naš sustav koristi JWT tehnologiju prenošenja podataka u obliku JSON objekata. JWT je skraćeno od JSON Web Token, a u našoj aplikaciji se koriste 2 vrste tokena, access token i refresh token. Access token služi za pristup resursima, a vremenski je ograničen. Refresh token ima dulje vrijeme trajanja od access tokena te služi za pridruživanje novog access tokena korisniku nakon isteka prijašnjeg access tokena. Na taj način korisnik se ne mora ponovno prijavljivati u sustav, no nakon eventualnog isteka refresh tokena, korisnik će ipak morati obaviti ponovnu prijavu u sustav.

4.2 Baza podataka

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

- Users
- Training
- Tournament
- Scheduled training
- Scheduled tournament
- News
- Daily tactics
- Score
- Reported mistake
- Membership

Opis tablica

Users Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: userID, role, name, surname, username, email i pwdHash. Ovaj entitet u vezi je *One-to-Many* s entitetom Training, Tournament, scheduledTraining, scheduledTournament, News, dailyTactics, Membership, reportedMistake i Score preko jedinstvenog identifikatora korisnika.

Users		
userID	INT	jedinstveni identifikator korisnika
role	VARCHAR	uloga korisnika (trainer, member, admin)
name	VARCHAR	ime korisnika
surname	VARCHAR	prezime korisnika
email	VARCHAR	e-mail adresa korisnika
pwdHash	INT	hash lozinke

Training Ovaj entitet sadržava sve važne informacije o šahovskom treningu pojedinog trenera. Sadrži attribute: trainingID, trainerID, trainingStartTimeDate i trainingDurationMin. Ovaj entitet u vezi je *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika i *One-to-Many* sa entitetom scheduledTraining.

Training		
trainingID	INT	jedinstveni identifikator treninga
trainerID	INT	jedinstveni identifikator trenera
trainingStartTimeDate	TIMESTAMP	vrijeme početka treninga
trainingDurationMin	INT	trajanje treninga

Tournament Ovaj entitet sadržava sve važne informacije o šahovskom turniru. Sadrži attribute: tournamentID, trainerID, tournamentStartTimeDate, tournamentDurationMin i participantsNo. Ovaj entitet u vezi je *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika i *One-to-Many* sa entitetom scheduledTournament.

Tournament		
tournamentID	INT	jedinstveni identifikator turnira
trainerID	INT	identifikator trenera
tournamentStartTimeDate	TIMESTAMP	vrijeme početka turnira
tournamentDurationMin	INT	trajanje turnira
participantsNo	INT	broj natjecatelja

scheduledTournament Pomoću ovog entiteta zapisujemo koji član je prijavljen na koji turnir. Sadrži attribute: memberID i tournamentID. Ovaj entitet u vezi je *Many-to-One* sa entitetom Tournament preko jedinstvenog identifikatora turnira i *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika.

scheduledTournament		
memberID	INT	jedinstveni identifikator člana
tournamentID	INT	jedinstveni identifikator turnira

scheduledTraining Pomoću ovog entiteta zapisujemo koji član je prijavljen na koje treninge. Sadrži attribute: memberID i trainingID. Ovaj entitet u vezi je *Many-to-One* sa entitetom Training preko jedinstvenog identifikatora treninga i *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika.

scheduledTraining		
memberID	INT	jedinstveni identifikator člana
tournamentID	INT	jedinstveni identifikator trninga

News Ovaj entitet sadržava sve važne informacije o objavljenoj novosti. Sadrži attribute: newsID, trainerID i content. Ovaj entitet u vezi je *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika.

News		
newsID	INT	jedinstveni identifikator novosti
trainerID	INT	jedinstveni identifikator trenera
content	VARCHAR	sadržaj novosti

dailyTactics Ovaj entitet sadržava sve važne informacije o dnevnoj taktici. Sadrži attribute: tacticID, trainerID, content i solution. Ovaj entitet u vezi je *Many-to-One*

sa entitetom Users preko jedinstvenog identifikatora korisnika, *One-to-Many* sa entitetima reportedMistake i Score.

dailyTactics		
tacticID	INT	jedinstveni identifikator dnevne taktike
trainerID	INT	jedinstveni identifikator trenera
content	VARCHAR	opis dnevnog zadatka
solution	VARCHAR	opis rješenja

Score Pomoću ovog entiteta zapisujemo uspjeh rješavanja dnevne taktike. Sadrži attribute: memberID, tacticID, solvingTime i accuracy. Ovaj entitet u vezi je *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika i *Many-to-One* sa entitetom dailyTactics preko jedinstvenog identifikatora dnevne taktike.

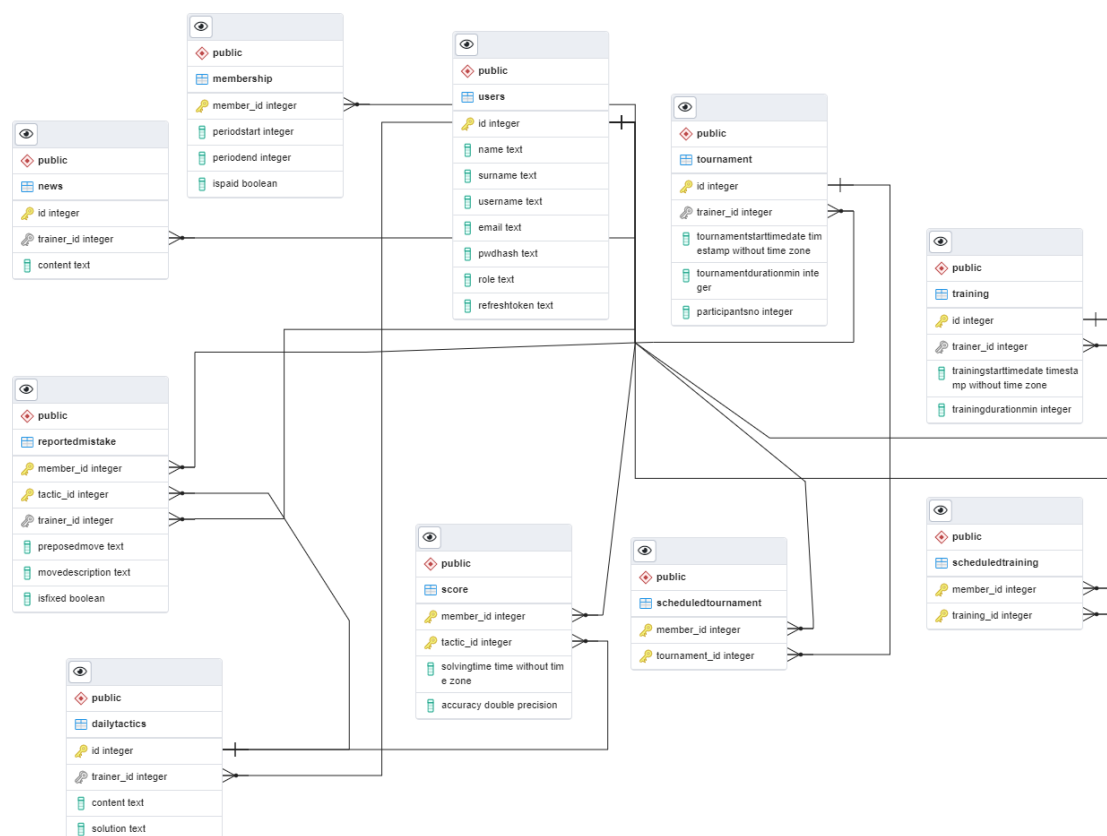
Score		
memberID	INT	jedinstveni identifikator člana
tacticID	INT	jedinstveni identifikator dnevne taktike
solvingTime	TIME	vrijeme rješavanja
accuracy	FLOAT	točnost rješenja

reportedMistake Pomoću ovog entiteta zapisujemo prijavljenu pogrešku dnevne taktike. Sadrži attribute: memberID, tacticID, trainerID, preposedMove, moveDescription i isFixed. Ovaj entitet u vezi je *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika i *Many-to-One* sa entitetom dailyTactics preko jedinstvenog identifikatora dnevne taktike.

reportedMistake		
memberID	INT	jedinstveni identifikator člana
tacticID	INT	jedinstveni identifikator dnevne taktike
trainerID	INT	jedinstveni identifikator trenera
preposedMove	VARCHAR	predloženo rješenje
moveDescription	VARCHAR	opis predloženog rješenja
isFixed	BOOLEAN	oznaka je li greška riješena

Membership Pomoću ovog entiteta zapisujemo plaćanje članarine članova. Sadrži attribute: memberID, periodStart, periodEnd i isPaid. Ovaj entitet u vezi je *Many-to-One* sa entitetom Users preko jedinstvenog identifikatora korisnika.

Membership		
memberID	INT	jedinstveni identifikator člana
periodStart	INT	početak perioda za koji se plaća članarina
periodEnd	INT	kraj perioda za koji se plaća članarina
isPaid	BOOLEAN	oznaka je li članarina plaćena

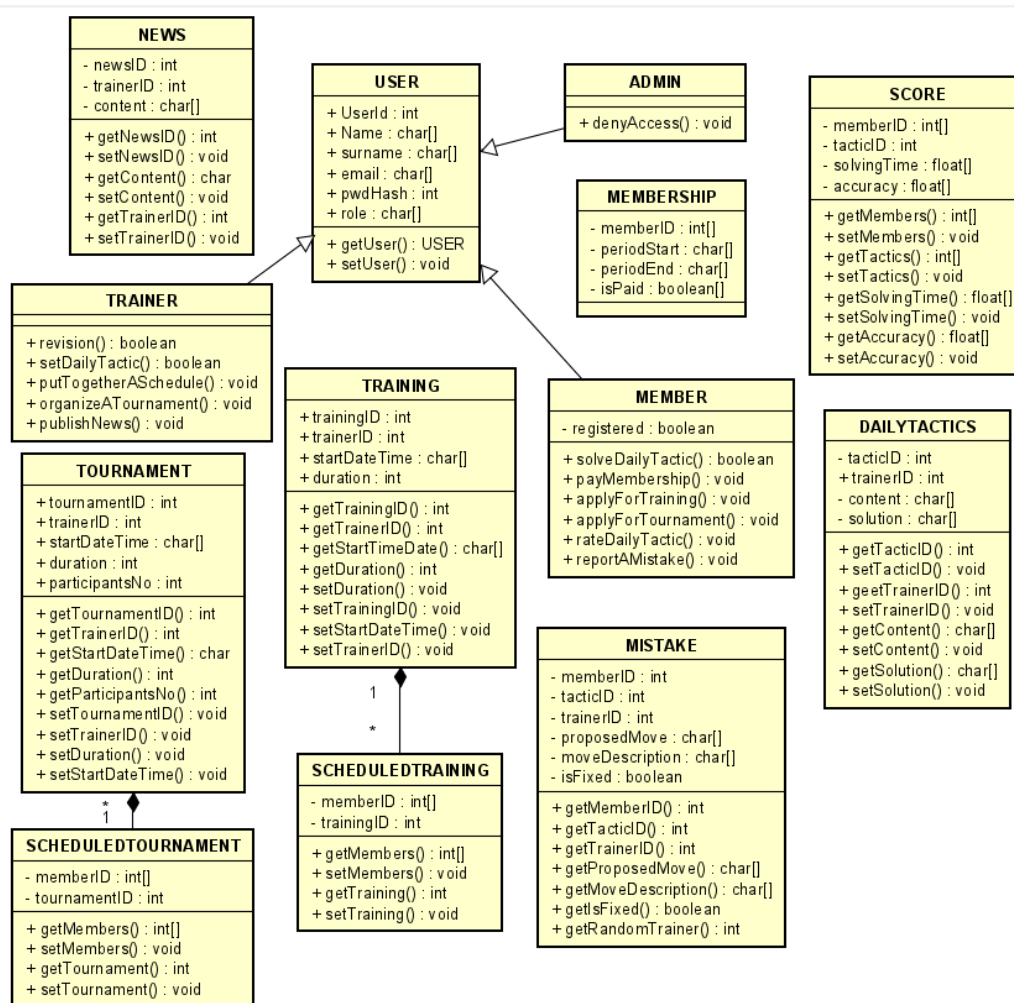


Slika 4.1: E-R dijagram baze podataka

4.3 Dijagrami razreda

Back-end arhitektura našeg sustava prikazana je u sljedećim dijagramima označenim brojevima 4.2, 4.3 i 4.4.

Prikazani dijagrami razreda sa slike 4.2 preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke poput novosti, rang liste ili dnevnih taktika.



Slika 4.2: Dijagram razreda

Razredi *Member*, *Admin* i *Trainer* specifikacija su razreda *User* pa s tom činjenicom nasljeđuju njegove public attribute i metode.

Razred *Member* predstavlja neregistriranog ili registriranog korisnika koji može rješavati dnevne taktike, a ako je atribut *registered* postavljen na *true*, može se prijaviti na turnire, treninge, ocjeniti dnevne taktike, prijaviti pogreške na istima ako ih ima te plaćati članarinu.

Razred *Admin* predstavlja administratora koji jedini ima dozvolu zabraniti pris-

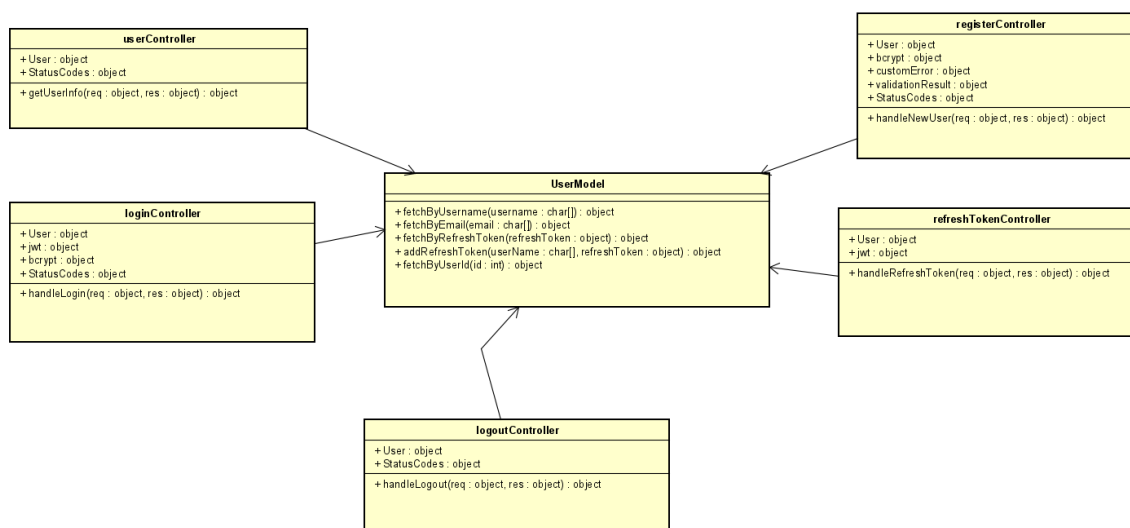
tup ostalim ulogama.

Razred *Trainer* predstavlja trenera koji sastavlja dnevne taktike, provodi provjeru prijavljenih pogrešaka dnevne taktike, sastavlja rasporede za treninge i organizira turnire te sastavlja novosti.

Razred *Training* i razred *ScheduledTraining* u vezi su "cjelina-dio" pri čemu je razred *Training* agregat. Isti slučaj vrijedi i za razrede *Tournament* i *ScheduledTournament*.

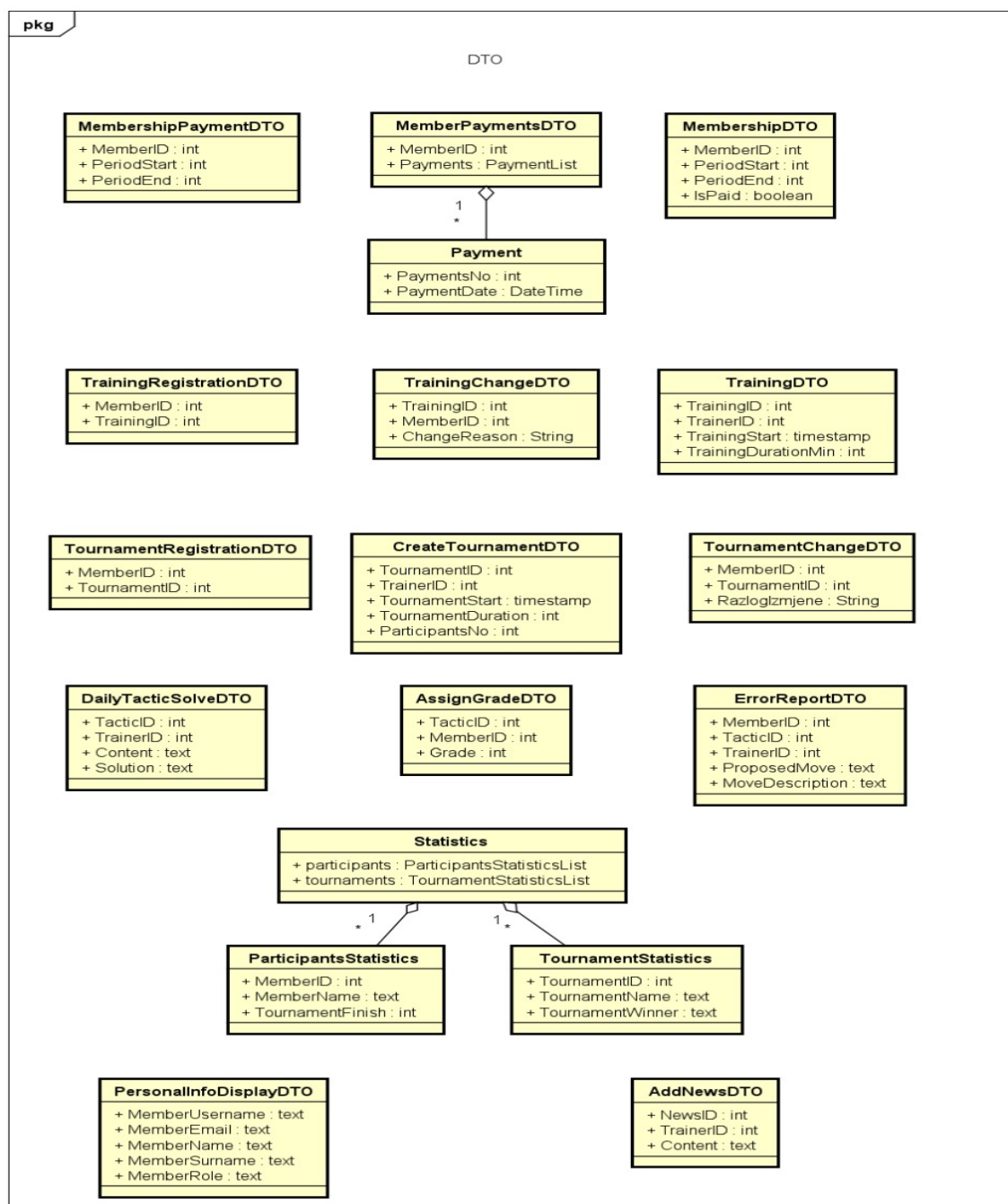
Razred *Membership* predstavlja skup atributa potrebnih za uvid o placenim članarinama, a razred *Mistake* ima potrebne attribute i metode kojima se upravlja postupkom prijave pogreške, provjere i prihvatanja ili odbijanja novog poteza.

Na slici 4.3 prikazani su razredi koji reprezentiraju controller dio našeg sustava.



Slika 4.3: Dijagram razreda - Controllers dio

Na slici 4.4 prikazan je dijagram razreda koji reprezentiraju Data transfer objects.



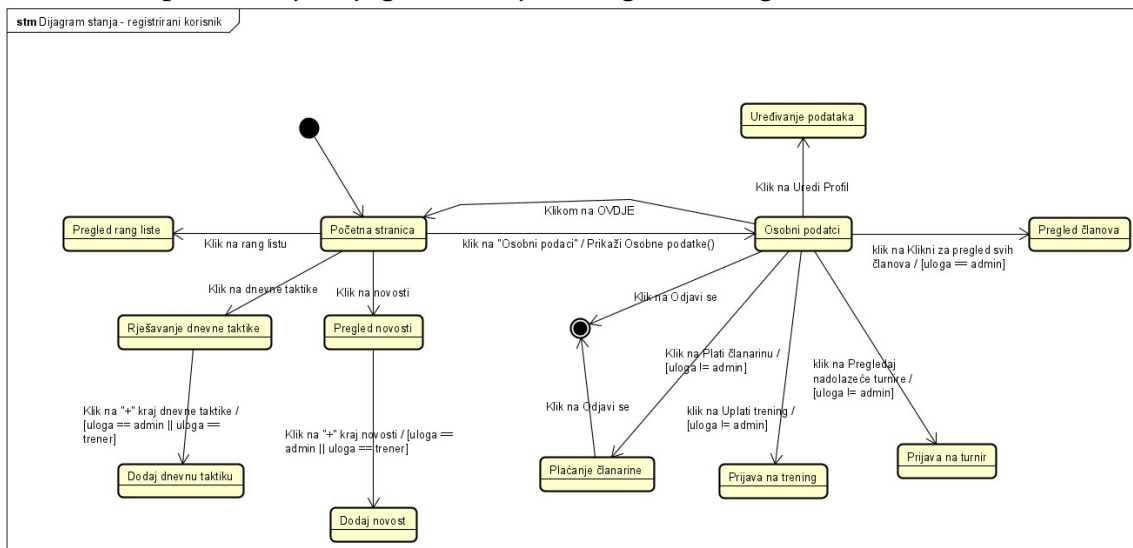
Slika 4.4: Dijagram razreda - Data transfer objects

DTO služi za razmjenu raznih zahtjeva i podataka između različitih dijelova sustava.

4.4 Dijagram stanja

Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima.

Na slici 4.5 prikazan je dijagram stanja za registriranog korisnika.



Slika 4.5: Dijagram stanja

Nakon prijave, klijentu se prikazuje početna stranica na kojoj može pregledati novosti, rang listu te odigrati dnevnu taktiku.

Klikom na "Osobni podatci" prikazuju mu se njegovi podatci koje može uređivati po želji.

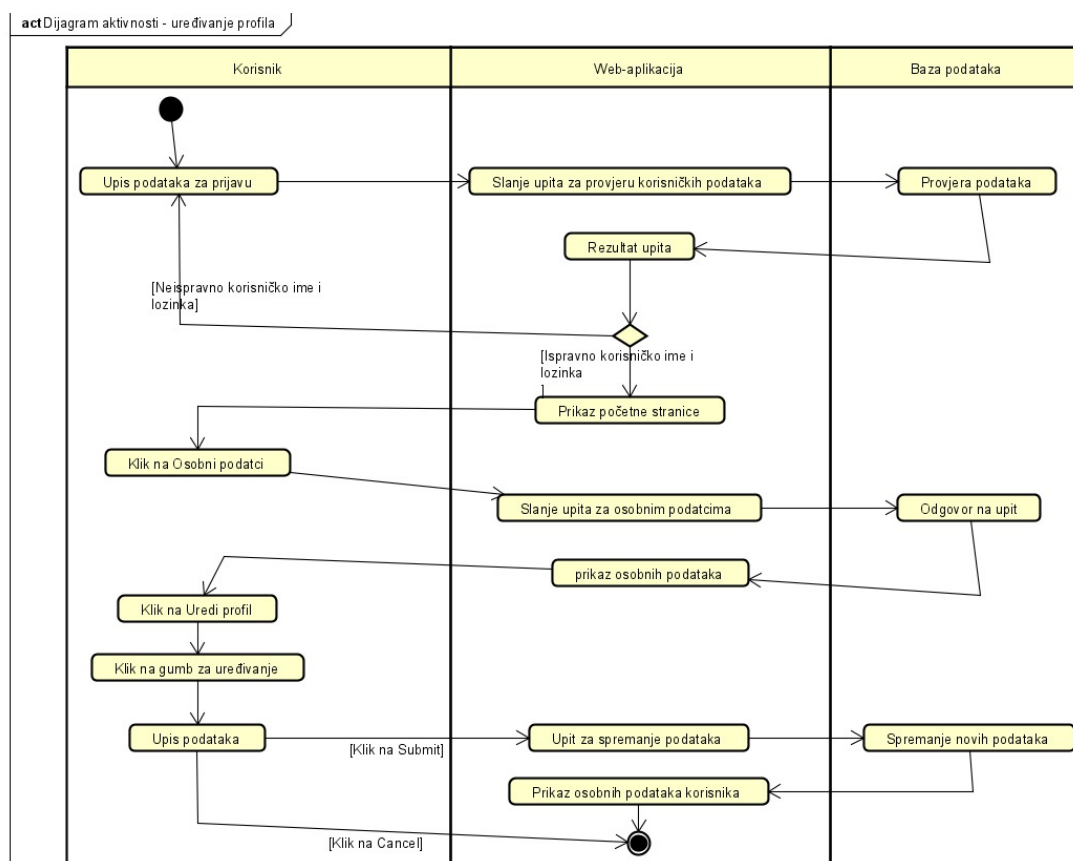
Registrirani korisnik ima i opcije prijave na treninge klikom na "Uplati trening", kao i prijavu na turnir klikom na "Pregledaj nadolazeće turnire". Korisnik uplatu izvršava klikom na "Plati članarinu".

U slučaju da je registrirani korisnik administrator, ima opciju pregleda svih članova kluba uz pomoć gumba "Klikni za pregled svih članova".

4.5 Dijagram aktivnosti

Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja.

U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti sa slike 4.6 prikazan je proces uređivanja profila.



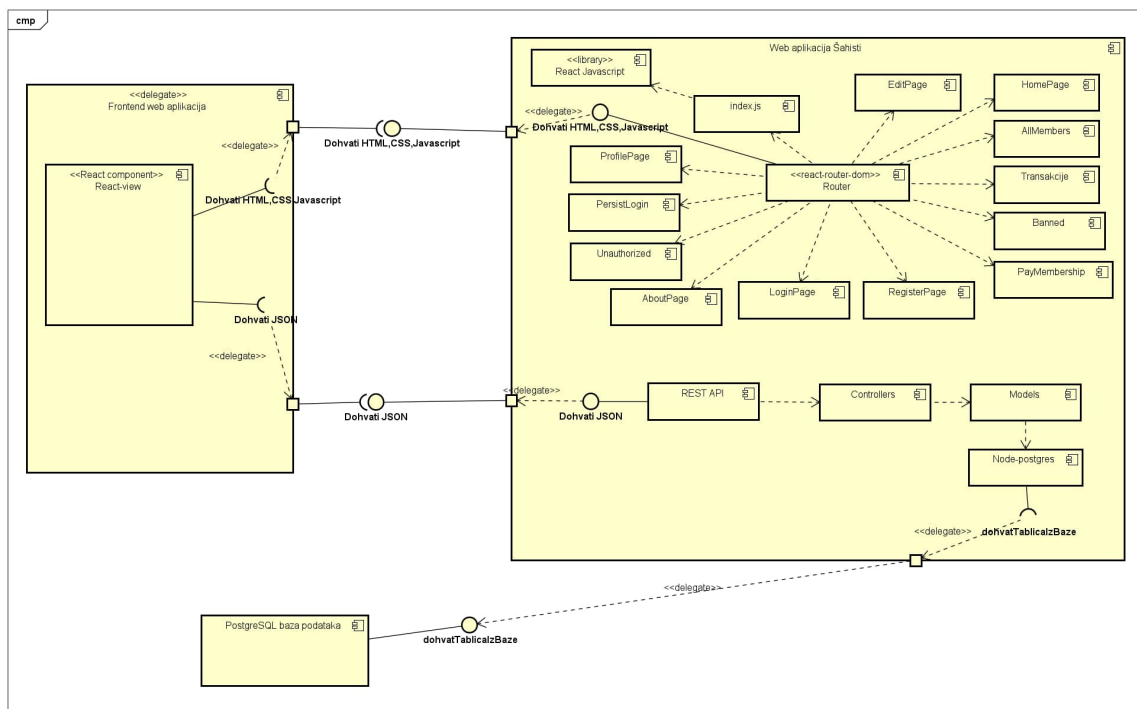
Slika 4.6: Dijagram aktivnosti

Korisnik nakon uspješne prijave u sustav sa svojim korisničkim podacima klikom na gumb "Uredi profil" započinje uređivanje nakon čega se pojavljuje prozorčić za upis novih podataka te klikom na "Submit" web-aplikacija šalje upit za spremanjem podataka bazi.

Ako je uređivanje uspješno izvršeno, korisniku se prikazuju njegovi novi osobni podatci.

4.6 Dijagram komponenti

Dijagram komponenti Slika 4.7 prikazuje dijagram komponenti kojim je vizualizirana organizacija te međuovisnost između implementacijskih komponenata te odnos programske potpore prema okolini.



Slika 4.7: Dijagram komponenti

Postoje dva različita sučelja. Sučelje za dohvat HTML, CSS te Javascript datoteka služi za posluživanje datoteka s frontend dijela aplikacije, a koje su prikazane kao komponente koje su potrebne komponenti Router za njezino djelovanje (ovisnost). Router je komponenta koja određuje koja će datoteka biti poslužena na sučelju. Sučelje za JSON podatke omogućuje pristup REST API-u koji je zadužen za backend podatke aplikacije. React-view komponenta preko sučelja komunicira s web aplikacijom Šahisti te djeluje ovisno o korisnikovim akcijama.

5 Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

U radu na našem projektu za međusobnu komunikaciju korištene su 3 aplikacije; Discord¹, MS Teams² i WhatsApp³.

Upravljanje izvornim kodom ostvareno je sustavom Git⁴, a udaljeni repozitorij našeg projekta dostupan nam je bio na web platformi GitLab⁵.

Kao razvojno okruženje korišten je Visual Studio Code⁶, uređivač izvornog koda razvijen od strane Microsofta s Frameworkom, dostupan za Windowse, Linux te macOS. Visual Studio Code podržava rad u brojnim jezicima te nudi mogućnost preuzimanja mnoštva ekstenzija za omogućavanje i olakšavanje rada programerima stoga je nama pružio dobru platformu za rad.

Backend dio stranice napisan je u jeziku JavaScript⁷, a frontend dio stranice napisan je u Reactu⁸, biblioteci u JavaScriptu koja se koristi za izradu korisničkih sučelja.

Baza podataka izrađena je pomoću PostgreSQL⁹ sustava za upravljanja bazom podataka, a za lakšu vizualizaciju podatka korišten je i pgAdmin¹⁰.

¹<https://discord.com/>

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

³<https://www.whatsapp.com/>

⁴<https://git-scm.com/>

⁵<https://gitlab.com/>

⁶<https://code.visualstudio.com/>

⁷<https://www.javascript.com/>

⁸<https://reactjs.org/>

⁹<https://www.postgresql.org/>

¹⁰<https://www.pgadmin.org/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Test 1: Ispitivanje registracije

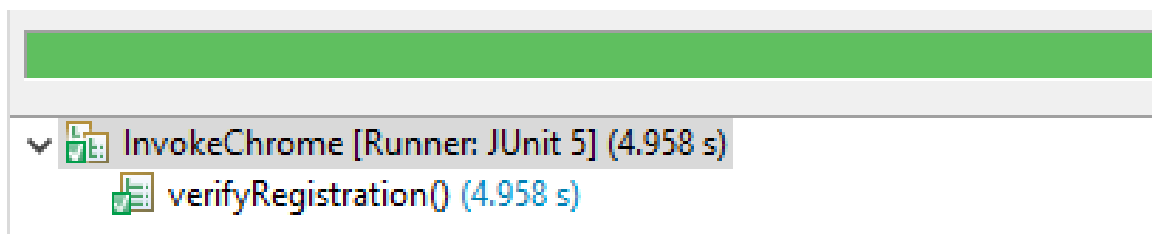
Provedeno je ispitivanje uspješnosti registracije stvaranjem novog korisnika sa podacima koji ne postoje u bazi podataka. Provodi se test koji nakon klika gumba za registraciju provjera pokazuje li se obavijest "Uspješna registracija". Ukoliko se korisniku prikazuje obavijest, registracija je uspješno izvršena.

Prikaz koda ispitivanja:

```
@Test
public void verifyRegistration() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\Chrome Driver\\chromedriver.exe");
    driver.get("https://almostbyte.onrender.com/register");
    driver.findElement(By.id("name")).sendKeys("Korisnik");
    driver.findElement(By.id("surName")).sendKeys("Korisnik");
    driver.findElement(By.id("userName")).sendKeys("KorisnikP");
    driver.findElement(By.id("email")).sendKeys("korisnikp@mail.com");
    driver.findElement(By.id("pwd")).sendKeys("Prvisifra!12");
    driver.findElement(By.id("matchpassword")).sendKeys("Prvisifra!12");
    driver.findElement(By.xpath("/html/body/div/main/main/div/form/button")).click();
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(2));
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/main/div[2]/div[2]")).getText(), "Uspješna registracija");
    driver.quit();
}
```

Slika 5.1

Prikaz rezultata ispitivanja:



Slika 5.2

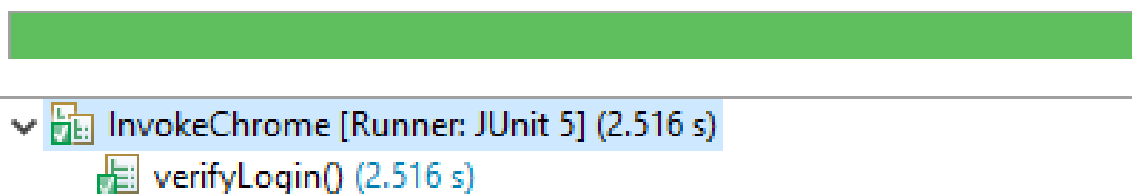
Test 2: Ispitivanje prijave Provedeno je ispitivanje uspješnosti prijave u sustav upisivanjem podataka korisnika koji već postoje u sustavu. Provodi se test koji nakon klika gumba za prijavu provjerava trenutni URL. Ako se korisniku prikazuje profil, prijava je uspješno izvršena.

Prikaz koda ispitivanja:

```
@Test
public void verifyLogin() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\Chrome Driver\\chromedriver.exe");
    driver.get("https://almostbyte.onrender.com");
    driver.get("https://almostbyte.onrender.com/login");
    driver.findElement(By.id("userNameEmail")).sendKeys("PrviiTest");
    driver.findElement(By.id("pwd")).sendKeys("Prvisifra!12");
    driver.findElement(By.xpath("/html/body/div/main/main/div/form/button")).click();
    driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(20));
    System.out.println(driver.getCurrentUrl());
    assertEquals(driver.getCurrentUrl(), "https://almostbyte.onrender.com/profile");
    driver.quit();
}
```

Slika 5.3

Prikaz rezultata ispitivanja:



Slika 5.4

Test 3: Ispitivanje pogreške prilikom prijave

Ispituje se hoće li se prikazati obavijest o pogrešnom unosu prilikom unosa nepostojećih podataka. Nakon pogrešnog unosa provjerava se prikazuje li se obavijest sa tekstom "Pogreška User ne postoji". Ukoliko je obavijest pronađena, test je ispravan.

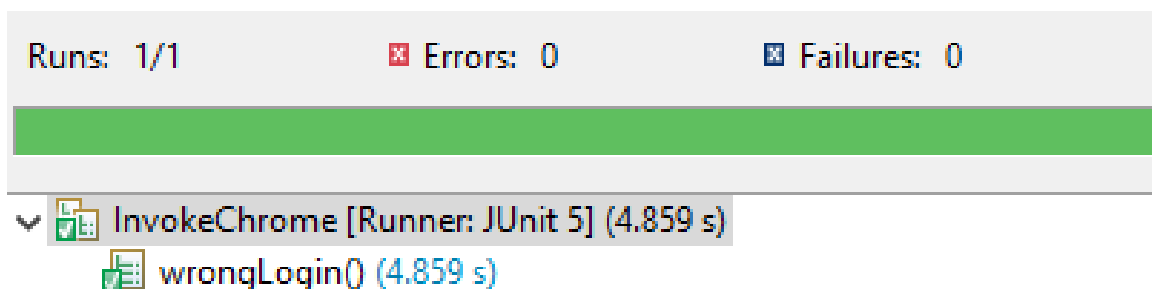
Prikaz koda ispitivanja:

```
@Test
public void wrongLogin() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\Chrome Driver\\chromedriver.exe");
    driver.get("https://almostbyte.onrender.com");
    driver.get("https://almostbyte.onrender.com/login");
    driver.findElement(By.id("userNameEmail")).sendKeys("Nepostoji");
    driver.findElement(By.id("pwd")).sendKeys("Sifra!12");
    driver.findElement(By.xpath("/html/body/div/main/main/div/form/button")).click();
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));
    assertTrue(driver.findElement(By.xpath("/html/body/div/main/main/div/div[2]")).isDisplayed());

    driver.quit();
}
```

Slika 5.5

Prikaz rezultata ispitivanja:



Slika 5.6

Test 4: Ispitivanje dohvata točnih podataka o korisniku

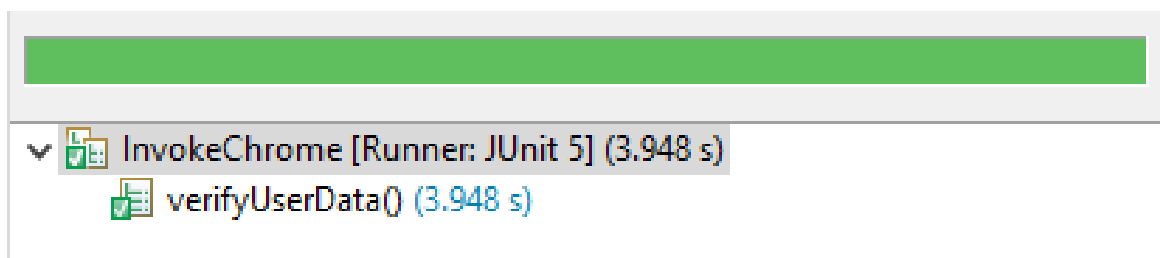
Ispituje se prikaz podataka korisniku nakon prijave, ako svi pročitani podatci odgovaraju očekivanim vrijednostima, test je ispravan.

Prikaz koda ispitivanja:

```
@Test
public void verifyUserData() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\Chrome Driver\\chromedriver.exe");
    driver.get("https://almostbyte.onrender.com/login");
    driver.findElement(By.id("userNameEmail")).sendKeys("PrviiTest");
    driver.findElement(By.id("pwd")).sendKeys("Prvisifra!12");
    driver.findElement(By.xpath("/html/body/div/main/main/div/form/button")).click();
    System.out.println(driver.getCurrentUrl());
    driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(20));
    driver.navigate().to("https://almostbyte.onrender.com/profile");
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/div[3]/div[1]/div/ul/li[1]/div/span")).getText(), "PrviiTest");
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/div[3]/div[1]/div/ul/li[2]/div/span")).getText(), "prviiTest@gmail.com");
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/div[3]/div[1]/div/ul/li[3]/div/span")).getText(), "Prvi");
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/div[3]/div[1]/div/ul/li[4]/div/span")).getText(), "Test");
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/div[3]/div[1]/div/ul/li[5]/div/span")).getText(), "user");
    driver.quit();
}
```

Slika 5.7

Prikaz rezultata ispitivanja:



Slika 5.8

Test 5: Ispitivanje dohvata točnih podataka o plaćanju

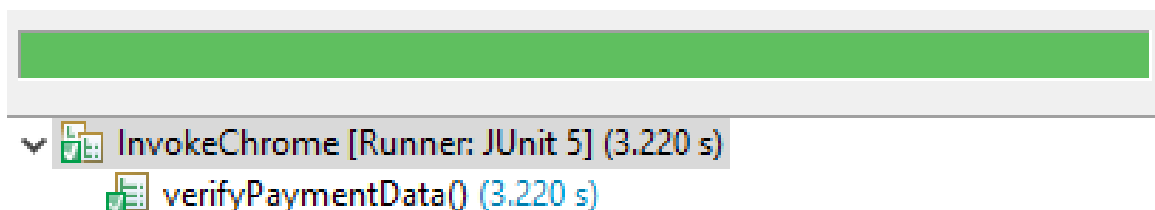
Ispituje se prikaz podataka o plaćanju korisnika, nakon prijave prelazimo na stranicu za plaćanje i provjeravamo ima li upisanih transakcija. S obzirom da testiramo novog korisnika, očekivani rezultat je “Nema uplata”. Ukoliko pročitano odgovara očekivanom rezultatu, test je ispravan.

Prikaz koda ispitivanja:

```
@Test
public void verifyPaymentData() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\Chrome Driver\\chromedriver.exe");
    driver.get("https://almostbyte.onrender.com");
    driver.get("https://almostbyte.onrender.com/login");
    driver.findElement(By.id("userNameEmail")).sendKeys("PrviiTest");
    driver.findElement(By.id("pwd")).sendKeys("Prvisifra!12");
    driver.findElement(By.xpath("/html/body/div/main/main/div/form/button")).click();
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));
    driver.navigate().to("https://almostbyte.onrender.com/PayMembership");
    driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));
    driver.findElement(By.xpath("/html/body/div/main/div[2]/div[1]/div/p[2]"));
    assertEquals(driver.findElement(By.xpath("/html/body/div/main/div[2]/div[1]/div/p[2]")).getText(), "Nema uplata");
    driver.quit();
}
```

Slika 5.9

Prikaz rezultata ispitivanja:



Slika 5.10

Test 6: Ispitivanje pogreške prikikom rješavanja dnevne taktike
Ispituje se hoće li se prikazati obavijes o pogrešnom rješenju dnevne taktike. Očekivani rezultat je “ Krivo rješenje. Pokušajte ponovo!”. Ukoliko izlaz odgovara oekivanom rješenju, test je ispravan.

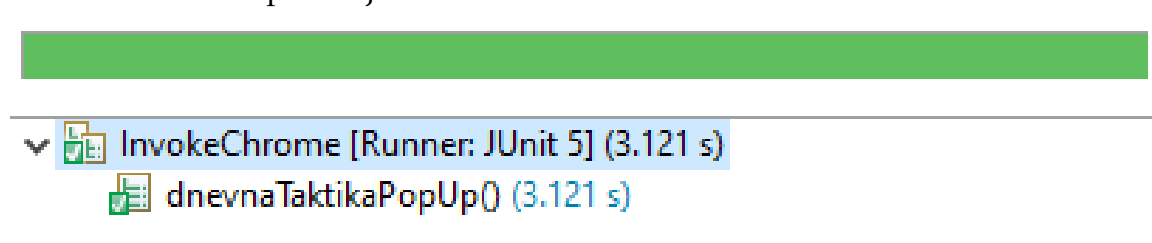
Prikaz koda ispitivanja:

```
@Test
public void dnevnaTaktikaPopUp() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\Chrome Driver\\chromedriver.exe");
    driver.get("https://almostbyte.onrender.com");

    driver.manage().timeouts().pageLoadTimeout(Duration.ofSeconds(20));
    driver.findElement(By.xpath("/html/body/div/main/div[2]/div[2]/div[2]/div/div/div[1]")).click();
    driver.findElement(By.xpath("/html/body/div/main/div[2]/div[2]/div[2]/div/div/div[2]/div/div/div/p[2]/button")).click();
    driver.findElement(By.xpath("/html/body/div/main/div[2]/div[2]/div[2]/div/div/div[2]/div/div/div/div/div/button")).click();
    driver.switchTo().alert().getText();
    assertEquals(driver.switchTo().alert().getText(), "Krivo rješenje. Pokušajte ponovo!");
    driver.quit();
}
```

Slika 5.11

Prikaz rezultata ispitivanja:



Slika 5.12

5.2.2 Ispitivanje sustava

Test 1: ispitivanje registracije

Preduvjeti: User name i email ne postoje u bazi

Podatci:

1. Name: Prvi
2. Surname: Test
3. Username: PrviiTest
4. Email Adress: prviitest@gmail.com
5. Password:****
6. Confirm Password: ****

Koraci:

1. Kliknuti na link za registraciju
2. Unijeti točne podatke u predviđena polja
3. Kliknuti na gumb “Registriraj se!”
4. Kliknuti na link za prijavu

Očekivani rezultat: Registracija je uspješna, nakon unosa podataka korisnik se može prijaviti na svoj račun

Stvarni izlaz: Jednak očekivanom rezultatu

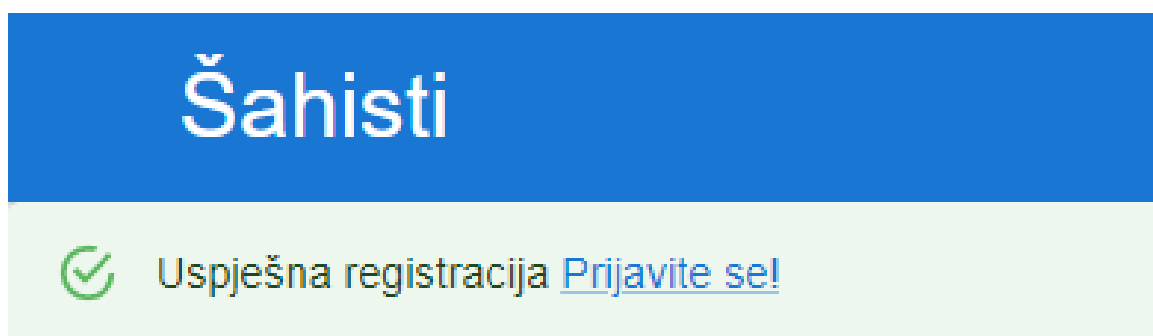
Prikaz rezultata za registraciju u Selenium IDE-u

https://almostbyte.onrender.com			
	Command	Target	Value
5	✓ mouse out	css=.MuiButton-text	
6	✓ click	linkText=Nemaš profil? Registriraj se!	
7	✓ click	id=name	
8	✓ type	id=name	Prvi
9	✓ click	id=surName	
10	✓ type	id=surName	Test
11	✓ click	id=username	
12	✓ type	id=username	PrviiTest
13	✓ click	id=email	
14	✓ type	id=email	prviitest@gmail.com
15	✓ click	id=pwd	
16	✓ type	id=pwd	****
17	✓ click	id=matchpassword	
18	✓ type	id=matchpassword	****
19	✓ click	css=.MuiButton-contained	
20	✓ click	linkText=Prijavite se!	

Slika 5.13

Prikaz stranice prije i nakon registracije:

Slika 5.14



Slika 5.15

Test 2: Ispitivanje prijave

Preduvjeti: Korisnik je registriran i zapisan u bazi podataka

Podatci:

1.Username: PrviiTest

2.Password: ****

Koraci:

1. Kliknuti na link za prijavu
2. Unijeti točne podatke u predviđena polja
3. Kliknuti na gumb "Prijava se!"

Očekivani rezultat: Prijava je uspješna, nakon unosa podataka korisnik može vidjeti svoje podatke

Stvarni izlaz: Jednak očekivanom rezultatu

Prikaz rezultata za registraciju u Selenium IDE-u:

https://almostbyte.onrender.com		
Command	Target	Value
1 ✓ open	/	
2 ✓ set window size	958x1039	
3 ✓ click	css= MuButton-textinherit	
4 ✓ mouse over	css= MuButton-text	
5 ✓ mouse out	css= MuButton-text	
6 ✓ click	id=usernameEmail	
7 ✓ type	id=usernameEmail	PrviiTest
8 ✓ click	id=pwd	
9 ✓ mouse over	css= MuButton-contained	
10 ✓ type	id=pwd	****
11 ✓ click	css= MuButton-contained	
12 ✓ mouse out	css= MuButton-contained	

Slika 5.16

Prikaz stranice prije i nakon prijave:

Šahisti

OSOBNI PODACI

Prijava

Unesite svoj username *

Password *

PRIJAVI SE!

☐ Zapani me!

[Nemaš profil? Registriraj se!](#)

Copyright © Šahisti 2023.

Slika 5.17

Šahisti

OSOBNI PODACI

Dobrodošli na svoj profil! Ovdje možete pregledati svoje osobne podatke.

Ukoliko se želite vratiti na home page, kliknite [OVDJE](#)

Vaš profil

OSOBNI PODACI

PriviTest
username

privtest@gmail.com
email

Privi
name

Test
surname

User
role

[UREDİ PROFIL](#)

POVIJEST PLAĆANJA

Nema uplata

[PLATI ČLANARINU](#)

TRENING

26. Listopad - trening održan

2. Studeni - trening održan

9. Studeni - trening održan

[UPLATI TRENING!](#)

TURNIRI

29. Listopad - 3. mjesto

5. Studeni - 7. mjesto

12. Studeni - 1. mjesto

[PREGLEDAJ NADOLAZEĆE TURNIRE](#)

ODJAVI SE

Slika 5.18

AlmostByte

50

13. siječnja 2023.

Test 3: ispitivanje registracije s pogrešnim unosom

Preduvjeti: User name ili email već postoje u bazi

Podatci:

7. Name: Prvi

8. Surname: Test

9. Username: PrviiTest

10. Email Adress: prviitest@gmail.com

11. Password:****

12. Confirm Password: ****

Koraci:

1. Kliknuti na link za registraciju
2. Unijeti točne podatke u predviđena polja
3. Kliknuti na gumb “Registriraj se!”
4. Izmjeniti već postojeće podatke

Očekivani rezultat: Registracija nije uspješna, nakon unosa podataka javlja se greška “User name je već iskorišten!”

Stvarni izlaz: Jednak očekivanom rezultatu Prikaz rezultata za registraciju u Selenium IDE-u

https://almostbyte.onrender.com			
	Command	Target	Value
4	✓ click	linkText=Nemaš profil? Registriraj se!	
5	✓ click	id=name	
6	✓ type	id=name	Prvi
7	✓ click	id=surName	
8	✓ type	id=surName	Test
9	✓ click	id=userName	
10	✓ type	id=userName	PrviiTest
11	✓ click	id=email	
12	✓ type	id=email	prviitest@mail.com
13	✓ click	id=pwd	
14	✓ type	id=pwd	****
15	✓ click	id=matchpassword	
16	✓ type	id=matchpassword	****
17	✓ click	css= MatButtonModule-contained	
18	✓ click	css= MuTypography-body1	Pogreška: User name je već korišten!
19	✓ click	css= MatButtonModule-text	
20	✓ mouse over	css= MatButtonModule-text	
21	✓ mouse out	css= MatButtonModule-text	

Slika 5.19

Prikaz stranice:

Šahisti

OSOBNI PODACI

Registracija

Username: ✓

Password: ✓

Email: ✓

Confirm Password: ✓

[Već imate račun? Prijava](#)

Copyright © Šahisti 2023.

Slika 5.20

Test 4: ispitivanje stvaranja nove novosti

Preduvjeti: Korisnik je admin ili trener

Koraci:

1. Kliknuti na ikonu za dodavanje novosti
2. Unijeti točne podatke u predviđena polja
3. Kliknuti na gumb “Spremi”

Očekivani rezultat: Nova novost je unešena u bazu podataka

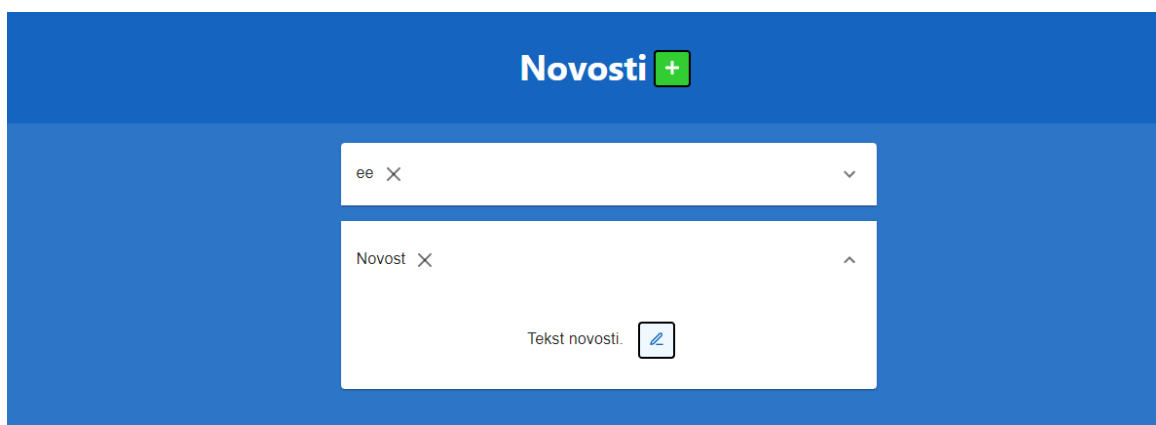
Stvarni izlaz: Jednak očekivanom rezultatu

Prikaz rezultata za registraciju u Selenium IDE-u:

https://almostbyte.onrender.com		
	Command	Target Value
1	✓ open	/
2	✓ set window size	945x1020
3	✓ run script	window.scrollTo(0,1107)
4	✓ click	css=paragraf:nth-child(4).addButton
5	✓ type	id=name Nova novost
6	✓ click	css=.Mui-focused > #name
7	✓ mouse over	css=.MuiPaper-root > .MuiDialogActions-root > .MuiButtonBase-root:nth-child(2)
8	✓ type	css=.Mui-focused > #name Tekst nove novosti.
9	✓ click	css=.MuiPaper-root > .MuiDialogActions-root > .MuiButtonBase-root:nth-child(2)
10	✓ click	css=.MuiPaper-root:nth-child(3) > #panel1a-header .MuiSvgIcon-root
11	✓ close	

Slika 5.21

Prikaz stranice:



Slika 5.22

Test 5: ispitivanje funkcionalnosti dnevnih taktika

Preduvjeti: Nema

Koraci:

1. Kliknuti na gumb za početak dnevne taktike
2. Pomaknuti figurice po želji
3. Kliknuti na gumb za završetak

Očekivani rezultat: Ako je korisnik prijavljen te je točno riješio taktiku, bit će stavljen na rang listu

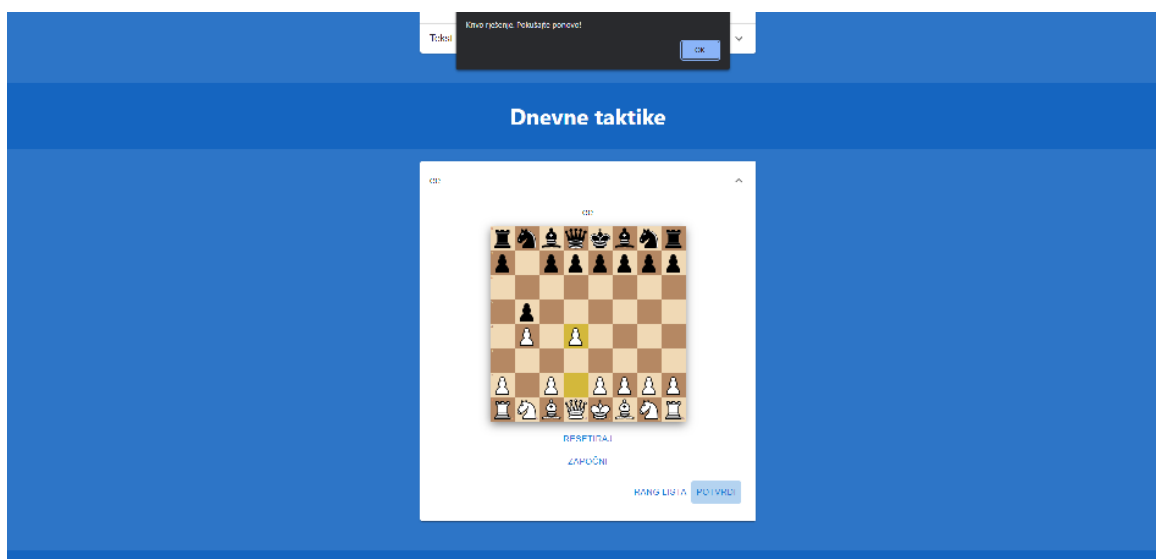
Stvarni izlaz: Taktika je pogrešno riješena

Prikaz rezultata za registraciju u Selenium IDE-u:

https://almostbyte.onrender.com			
	Command	Target	Value
1	✓ open	/	
2	✓ set window size	958x1039	
3	✓ click	css=.news:nth-child(8) > .MuiAccordionSummary-content	
4	✓ click	css=.MuiTypography-root > .MuiButtonBase-root	
5	✓ drag and drop to object	css=div:nth-child(7) > div:nth-child(4) > div > div	css=div:nth-child(5) > div:nth-child(4) > div
6	✓ drag and drop to object	css=div:nth-child(1) > div:nth-child(2) > div:nth-child(5) > div > div	css=div:nth-child(4) > div:nth-child(5) > div
7	✓ click	css=div:nth-child(2) > div:nth-child(4) > div > svg	
8	✓ drag and drop to object	css=div:nth-child(2) > div:nth-child(4) > div > div	css=div:nth-child(1) > div:nth-child(2) > div:nth-child(5) > div
9	✓ click	css=div:nth-child(7) > div:nth-child(5) > path	
10	✓ mouse down at	css=div:nth-child(5) > div:nth-child(5) > div	26.5, 37.625
11	✓ mouse move at	css=div:nth-child(5) > div:nth-child(5) > div	26.5, 37.625
12	✓ mouse up at	css=div:nth-child(5) > div:nth-child(5) > div	26.5, 37.625
13	✓ click	css=.MuiTypography-root > div > div > div > div	
14	✓ double click	css=.MuiTypography-root > div > div > div > div	
15	✓ drag and drop to object	css=div:nth-child(7) > div:nth-child(5) > div > div	css=div:nth-child(5) > div:nth-child(5) > div
16	✓ drag and drop to object	css=div:nth-child(2) > div:nth-child(7) > div > div	css=div:nth-child(4) > div:nth-child(7) > div
17	✓ drag and drop to object	css=div:nth-child(8) > div:nth-child(6) > div > div:nth-child(1)	css=div:nth-child(5) > div:nth-child(3) > div
18	✓ click	css=.MuiDialogActions-root > .MuiButtonBase-root	
19	✓ assert alert	Krivo rješenje. Pokušajte ponovo!	

Slika 5.23

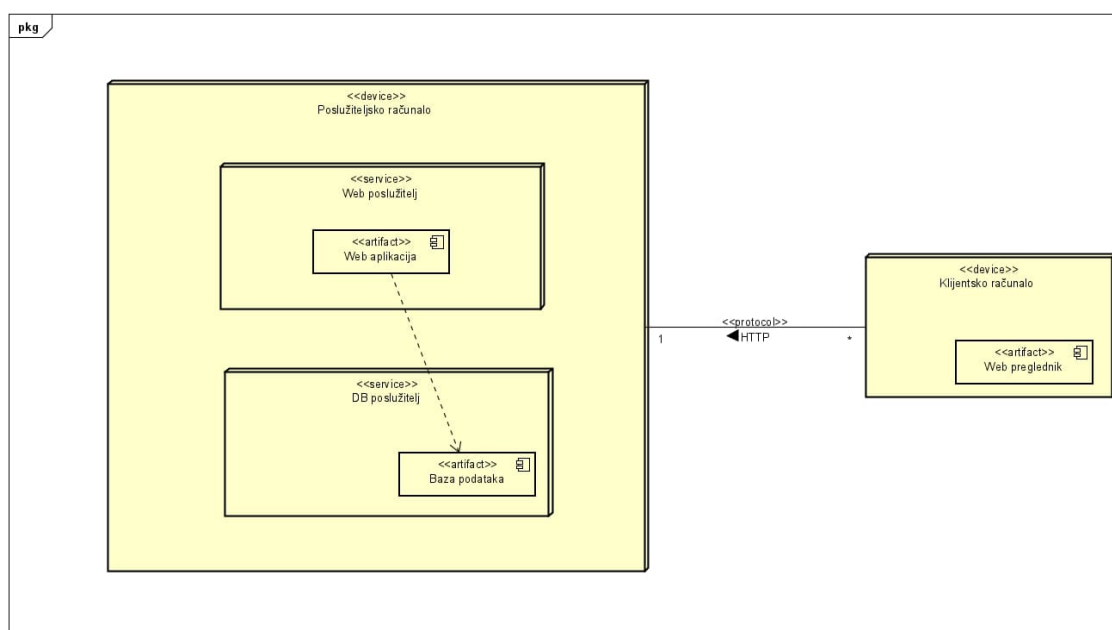
Prikaz stranice:



Slika 5.24

5.3 Dijagram razmještaja

Na slici 5.25 prikazan je dijagram razmještaja koji opisuje topologiju sustava. Sustav se sastoji od poslužiteljskog računala na kojem se nalaze web poslužitelj te poslužitelj baze podataka. Na dijagramu je prikazano i klijentsko računalo, a klijent pristupa aplikaciji preko web preglednika. Komunikacija klijent – poslužitelj odvija se preko HTTP protokola.



Slika 5.25: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Web aplikaciju koju smo razvili u okviru ovog projekta pustili smo u pogon koristeći Render¹¹ web stranicu koja nudi besplatne usluge dovoljne za funkcionalnost cijele naše aplikacije.

Na Render-u smo postavili PostgreSQL bazu podataka, Node poslužitelj koji odgovara na upite korisnika te statičku stranicu koja služi za prikaz React web aplikacije.

NAME	STATUS	TYPE	ENVIRONMENT	REGION	▼ LAST DEPLOYED
BackendSahisti	Deploy succeeded	Web Service	Node	Oregon	3 days ago
AlmostByte	Deploy succeeded	Static Site	Static	Global	4 days ago
SahistiBaza	Available	PostgreSQL	PostgreSQL 15	Oregon	2 months ago

Slika 5.26

Postavljanje baze podataka

Postavljanje baze podataka kroz Render web sučelje je jednostavno. Render nam nudi biranje određenih parametara baze poput imena, korisnika baze, lozinke te vrata baze putem kojih se Node poslužitelj spaja na bazu. Nakon što smo napravili bazu podataka ona je prazna te u nju moramo dodati tablice prilagođene našoj web aplikaciji. Render nam za to daje PSQL naredbu putem koje se možemo izvana spo-

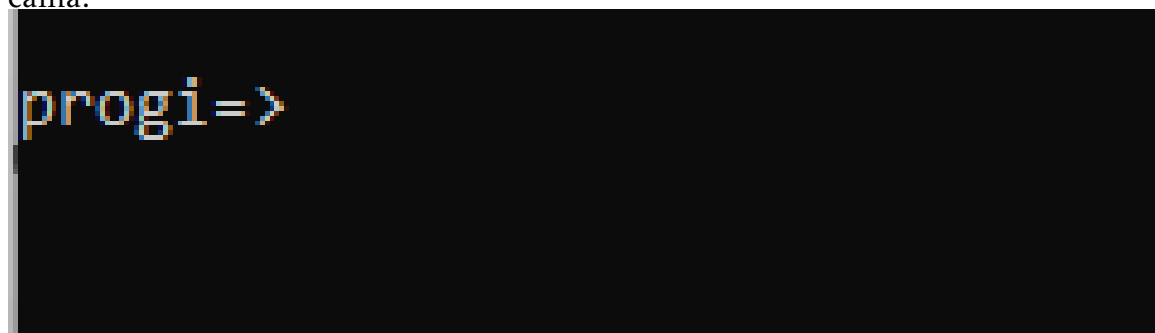
жити na bazu podataka te ju onda napuniti podatcima.

PSQL Command



Slika 5.27

Iduća slika prikazuje naredbeni redak putem kojeg punimo bazu željenim tablicama.



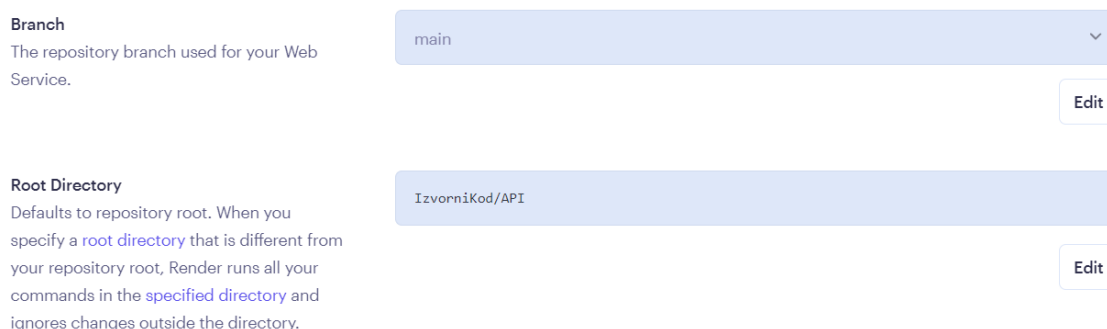
Slika 5.28

¹¹<https://render.com/>

Podatci o svim tablicama baze podataka nalaze se u datoteci `seed.js`. Ta datoteka sadrži SQL naredbe putem kojih direktno inicijaliziramo bazu. Pomoću PSQL naredbe se iz naredbenog retka izvana spajamo na bazu te kopiramo sve naredbe iz `seed.js`. Nakon ovog koraka u bazi se nalaze sve tablice potrebne za rad web aplikacije.

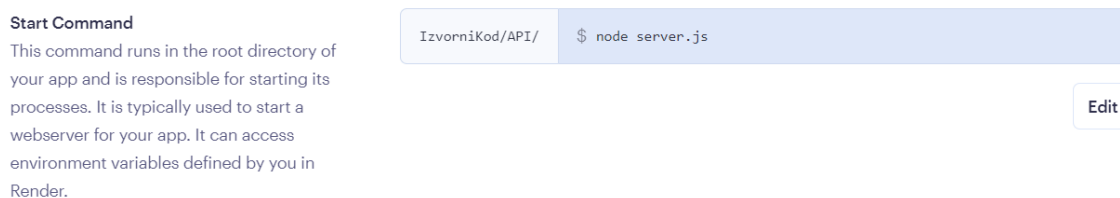
Postavljanje Node poslužitelja

Kroz Render web sučelje stvaramo poslužitelj koji će odgovarati na upite korisnika iz React web aplikacije te koji komunicira sa bazom podataka. Renderova usluga Web service je pravi izbor za ovo. Render nam nudi odabir GitLab grane koja će služiti za deploy poslužitelja te konfiguriranje putanje u kojoj se nalaze datoteke poslužitelja. Odabrali smo main granu našeg projekta te sada svaki puta kada se dogodi neki push na tu granu Node poslužitelj na Renderu će se automatski ažurirati.



Slika 5.29

Također, moramo postaviti i naredbu putem koje će Render pokretati poslužitelj. Kako je riječ o Node poslužitelju to radimo sa naredbom: `node server.js`.



Slika 5.30

Kako bi sve radilo moramo konfigurirati i varijable okoline koje se ne objavljuju na GitLab već je svaki član tima imao vlastitu `.env` datoteku koju je konfigurirao prema vlastitim potrebama. U `.env` datoteci sada se moraju nalaziti parametri koji

će omogućiti Node poslužitelju spajanje na prethodno stvorenu bazu i stvaranje Json web tokena.

Key	Value	
ACCESS_TOKEN_SECRET	
CORS	
DB_HOST	
DB_NAME	
DB_PASSWORD	
DB_PORT	
DB_USER	
REFRESH_TOKEN_SECRET	
TOKEN_EXPIRATION	

Slika 5.31

Postavljanje React web aplikacije

Stvaranje React web aplikacije kroz Render je jednostavno. React web aplikaciju kroz Render postavljamo tako da koristimo static site uslugu. Kao i kod kofiguiranja Node poslužitelja biramo GitLab granu te putanju do datoteka potrebnih za React web aplikaciju.

Branch The repository branch used for your Static Site.	main			Edit
Root Directory Defaults to repository root. When you specify a root directory that is different from your repository root, Render runs all your commands in the specified directory and ignores changes outside the directory.	Izvornikod/client			Edit

Slika 5.32

Kako bi Render znao stvoriti build verziju React web aplikacije moramo napisati ispravnu naredbu.

Build Command

This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.

IzvorniKod/client/ \$ npm run build

Edit

Slika 5.33

Ovom naredbom React stvara optimalnu verziju web aplikacije koju će Render na kraju i prikazivati u pregledniku. Optimalnu verziju čini jedan html dokument te nekoliko Javascript datoteka potrebnih za rad aplikacije. Datoteke optimalne verzije tj. builda nalaze se u direktoriju build.

Publish directory

The [relative](#) path of the directory containing built assets to publish. Examples: `./`, `./build`, `dist` and `frontend/build`.

IzvorniKod/client/ build

Edit

Slika 5.34

Ostalo je još konfigurirati varijable okoline. Sada imamo jednu naredbu koja predstavlja URL kojim React web aplikacija pristupa Node poslužitelju.

Key

Value

REACT_APP_BACKEND_HOST

.....



Slika 5.35

Nakon svih ovih koraka aplikaciji pristupamo putem URL-a koji nam je dao Render za static site uslugu. Ovim URL-om pristupamo React web aplikaciji koja šalje upite na Node poslužitelj koji dalje može komunicirati sa bazom.

6 Zaključak i budući rad

Nakon 17 tjedana postignut je konačni cilj ovog projekta a to je izrada funkcionalne web aplikacije šahovskog kluba. U ostvarenoj aplikaciji omogućeno je registriranje i prijava korisnika, rješavanje dnevnih taktika, pregled rang listi, prijavljivanje pogrešaka u dnevnoj taktici te revizija istih, plaćanje članarine šahovskog kluba, prijavljivanje na treninge i turnire postavljenih na aplikaciju od strane ovlaštenih osoba te pregled osobnih podataka.

Sama izrada aplikacije tekla je u dvije faze koje su se otprilike podudarale s ciklusima akademskog semestra.

Prva faza bila je slabijeg intenziteta od druge no nije bila bez svojih izazova. U prvoj fazi bilo je ključno definirati potrebne stvari za izradu aplikaciju te konceptualno izgraditi "kostur" prema kojem će se graditi aplikacija. U tu svrhu trebalo je dobro razumjeti i dokumentirati sve zahtjeve naše aplikacije, izraditi use case-ove te njihove dijagrame, izraditi dijagrame razreda te osmisлити bazu podataka. Sve je to bilo potrebno lijepo dokumentirati u jedan čitki i smisleni dokument čije poglavlje upravo čitate. Osim dokumentacije, potrebno je bilo implementirati generičke funkcionalnosti stranicu kao što su registracija, prijava i slično. A uz sve to, svi smo se morali prilagoditi radu u grupi sa osobama koje smo uglavnom netom upoznali.

Nakon uspješno i kvalitetno odrađene prve faze, uslijedila je druga, užurbanija i zahtjevnija faza. U ovoj fazi glavni fokus bio je na izradi implementacijskih rješenja. Na trenutke je ovo iziskivalo mnoštvo napora s obzirom da je većini članova ovo prvi susret s radom na projektu ovakvog obujma. No ipak sve je uglavnom uspješno savladano te smo na kraju dobili aplikaciju koja je spremna za korištenje, sve što joj je potrebno je stvarni šahovski klub. Osim same implementacije, bilo je i posla oko ispitivanje programskog rješenja te puštanja aplikacije u pogon(deployment). Također, na dokumentaciji je preostalo posla, kao što je izrada dijagrama aktivnosti, stanja, komponenti, razmještaja i slično.

Iako je aplikacija izrađena i puštena u pogon prostora za daljnji rad ima. Sitne greške se mogu pronaći te bi ih se moglo još dodatno "ispeglati". Na dizajnu stranice bi se također dalo poraditi, no i ovakvim minimalističkim stilom i dalje izgleda pristojno. Jedan od mogućih daljnjih razvoja je i omogućavanje igranja partija tzv. dopisnog šaha u kojem članovi međusobno naizmjenice šalju poteze jedan drugome unutar nekog definiranog vremenskog okvira te na taj način simuliraju "over the board" partiju šaha. Također bi se moglo poraditi i na dodavanju video sadržaja od strane trener, na primjer lekcije o otvaranjima, analize legendarnih partija i slično.

Ovaj projekt te naš rad i sudjelovanje vrijedno je iskustvo za sve članove tima. Nije bilo lako prilagoditi se na ovakav način rada te obujam posla, no to je ono što nas sve čeka u budućnosti. Timski rad, podrška drugima, organiziranje vremena, samoedukacija i korištenje do sada nepoznatih tehnologija; sve su to vrline koje smo trebali usvojiti kako bi odradili ovaj projekt. Uzimajući u obzir sve navedeno, ali i ostale obveze koje individualno imamo na nekim drugim područjima, možemo biti zadovoljni s obavljenim poslom na ovom projektu.

7 Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. The Unified Modelling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>
4. React MUI, <https://mui.com/>

8 Dodatak: Prikaz aktivnosti grupe

8.1 Dnevnik sastajanja

1. sastanak

- Datum: 17. listopada 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Upoznavanje tima
 - Okvirna podjela poslova

2. sastanak

- Datum: 24. listopada 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Diskutirana izrada funkcionalnih zahtjeva, use case zahtjeva, sekvencijskih dijagrama te ostalih zahtjeva
 - Obavljena konkretna podjela posla

3. sastanak

- Datum: 4. studenog 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec i Mijo Rajič
- Teme sastanka:
 - Provjera obavljenog posla
 - Razrada baze podataka i početak njene implementacije
 - Podjela daljnjeg rada u dokumentaciji

4. sastanak

- Datum: 9. studenog 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič i Anteo Vukasović
- Teme sastanka:
 - Prikaz i testiranje do sada obavljenog posla

5. sastanak

- Datum: 18. studenog 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Provjera i potvrđivanje napisane dokumentacije za prvi ciklus

6. sastanak

- Datum: 9. prosinca 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Podjela poslova implementacije među članovima tima

7. sastanak

- Datum: 14. prosinca 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Konzultacije s asistentom

8. sastanak

- Datum: 22. prosinca 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Prikaz i testiranje implementiranog sadržaja
 - Podjela daljnjih poslova među članovima tima

9. sastanak

- Datum: 4. siječnja 2023.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič i Anteo Vukasović
- Teme sastanka:
 - Prikaz i testiranje do sada obavljenog posla
 - Planiranje rada završne faze implementacija
 - Podjela daljnjeg rada u dokumentaciji
 - Podjela posla za ispitivanje programskog rješenja

10. sastanak

- Datum: 13. siječnja 2022.
- Prisustvovali: Ivan Bilobrk, Altea Božić, Tina Jureško, Lara Mahalec, Mijo Rajič, Ilan Vezmarović i Anteo Vukasović
- Teme sastanka:
 - Konačna verifikacija dokumentacije i programskog rješenja
 - Podjela posla oko prezentacije projekta

8.2 Tablica aktivnosti

U nastavku je prikazana tablica aktivnosti na projektu. Sve vrijednosti tablice predstavljaju sate rada člana tima na pojedinoj aktivnosti.

	Ivan Bilobrk	Altea Božić	Tina Jureško	Lara Mahalec	Mijo Rajič	Ilan Vezmarović	Anteo Vukasović
Upravljanje projektom	8						
Opis projektnog zadatka							1
Funkcionalni zahtjevi	1	1	1	1	1	1	3
Opis pojedinih obrazaca	1	1	1	1	1	1	3
Dijagram obrazaca	1	1	1	1	3	1	1
Sekvencijski dijagrami		3	3	2			
Opis ostalih zahtjeva							1
Arhitektura i dizajn sustava	1	1	1	1	1	1	2
Baza podataka	1	1	1	10	1	1	
Dijagram razreda		3	2		3		1
Dijagram stanja		4					
Dijagram aktivnosti			4				
Dijagram komponenti		4					
Korištene tehnologije i alati							1
Ispitivanje programskog rješenja				8			
Dijagram razmještaja			4				
Upute za puštanje u pogon	1						
Zaključak i budući rad							1
Popis literature							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ivan Bilobrk	Altea Božić	Tina Jureško	Lara Mahalec	Mijo Rajič	Ilan Vezmarović	Anteo Vukasović
Front-end	14	10	10	10	9	14	16
Back-end	24			16	12	18	2
Deployment	4						