

# Programsko inženjerstvo

Ak. god. 2020./2021.

## MISFLIP

Dokumentacija, Rev. 2.0

Grupa: *MISFLIP*

Voditelj: *Ivar Cmrečak*

Datum predaje: *14.1.2021.*

Nastavnik: *Igor Stančin*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>2</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
2.1 Motivacija za projekt . . . . .	9
<b>3 Specifikacija programske potpore</b>	<b>10</b>
3.1 Funkcionalni zahtjevi . . . . .	10
3.1.1 Obrasci uporabe . . . . .	13
3.1.2 Sekvencijski dijagrami . . . . .	26
3.2 Ostali zahtjevi . . . . .	34
<b>4 Arhitektura i dizajn sustava</b>	<b>35</b>
4.1 Baza podataka . . . . .	36
4.1.1 Opis tablica . . . . .	38
4.1.2 Dijagram baze podataka . . . . .	42
4.2 Dijagram razreda . . . . .	43
4.3 Dijagram stanja . . . . .	47
4.4 Dijagram aktivnosti . . . . .	48
4.5 Dijagram komponenti . . . . .	49
<b>5 Implementacija i korisničko sučelje</b>	<b>51</b>
5.1 Korištene tehnologije i alati . . . . .	51
5.2 Ispitivanje programskog rješenja . . . . .	52
5.2.1 Ispitivanje komponenti . . . . .	52
5.2.2 Ispitivanje sustava . . . . .	55
5.3 Dijagram razmještaja . . . . .	59
5.4 Upute za puštanje u pogon . . . . .	60
<b>6 Zaključak i budući rad</b>	<b>62</b>
<b>Popis literature</b>	<b>63</b>

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Klabučar	08.10.2020.
0.2	Napisani dionici i napravljeni funkcionalni zahtjevi za aktore	Kedmenec	10.10.2020.
0.2.1	Napisana motivacija za projekt	Klabučar	12.10.2020.
0.2.2	Napravljene dopune funkcionalnih zahtjeva	Paprskar	13.10.2020.
0.3	Napisani opisi obrazaca uporabe za neregistriranog korisnika, trenera i članove	Svetec, Klabučar, Paprskar	19.10.2020.
0.3.1	Napisani opisi obrazaca uporabe za administratora	Kedmenec	20.10.2020.
0.4	Napisani nefunkcionalni zahtjevi i zahtjevi domene	Kedmenec	24.10.2020.
0.4.1	Dodani sekvencijski dijagrami za revidiranje taktike i stvaranje treninga	Klabučar	25.10.2020.
0.4.2	Dodan sekvencijski dijagram za objavu taktike	Leontić	28.10.2020.
0.4.3	Dodan sekvencijski dijagram za rješavanje taktike	Žižić	28.10.2020.
0.4.4	Dodan dijagram obrasca uporabe za funkcionalnost trenera	Paprskar	29.10.2020.
0.4.5	Modificirani obrasci uporabe	Paprskar	29.10.2020.
0.4.6	Dodan dijagram obrasca uporabe za funkcionalnost neregistriranog korisnika i člana	Svetec	29.10.2020.
0.4.7	Dodan dijagram obrasca uporabe za funkcionalnost administratora	Cmrečak	30.10.2020.
0.4.8	Numerirani svi obrasci uporabe i dodan UC19	Svetec	2.11.2020.
0.4.9	Dodane tablice vezane uz treninge i turnire	Leontić	2.11.2020.

Rev.	Opis promjene/dodatka	Autori	Datum
0.4.10	Ispravljen dijagram za rješavanje taktike	Žižić	3.11.2020.
0.4.11	Ispravljen dijagram obrasca uporabe za funkcionalnost trenera	Paprskar	4.11.2020.
0.4.12	Dodane tablice baze podataka vezane uz taktike	Klabučar	5.11.2020.
0.5	Dodan opis projektnog zadatka	Leontić	5.11.2020
0.5.1	Ispravljen admin use case dijagram i povezana dokumentacija	Žižić	5.11.2020.
0.5.2	Dovršen opis projektnog zadatka	Paprskar	6.11.2020
0.5.3	Dodana tablica transakcija	Leontić	6.11.2020
0.5.4	Dodan dijagram baze podataka	Svetec	6.11.2020.
0.5.5	Dodani dijagrami razreda	Klabučar	7.11.2020.
0.5.6	Gramatički ispravljen dio dokumentacije	Paprskar	9.11.2020.
0.5.7	Ispravljena veza između UC2 i UC3	Svetec	10.11.2020.
0.5.8	Dodan opis korištenog susatava baza podataka	Kedmenec	10.11.2020.
0.6	Popunjena tablica aktivnosti	Paprskar, Klabučar	12.11.2020.
0.6.1	Dodan opis veza uz tablice za trening, turnir i transakcije	Leontić	12.11.2020
0.6.2	Dodana literatura	Kedmenec	12.11.2020
1.0	Korigiranje teksta i provjera dokumentacije	Kedmenec, Cmrečak, Svetec, Klabučar, Leontić	13.11.2020
1.0	Napravljeni dijagram stanja za neregistriranog korisnika i dijagram aktivnosti za revidiranje dojave o pogrešci na taktici	Paprskar	13.12.2020.
1.1.1	Zamijenjen dijagram stanja neregistriranog korisnika s dijagramom stanja člana	Paprskar	15.12.2020.
1.1	Ažuriranje sastanaka	Paprskar	3.1.2021.
1.2	Dodano poglavlje Zaključak i budući rad	Paprskar	3.1.2021.
1.3	Dodan dijagram razmještaja	Paprskar	3.1.2021.

Rev.	Opis promjene/dodatka	Autori	Datum
1.4.1	Dodano poglavlje Korištene tehnologije i alati	Paprskar	3.1.2021.
1.4.2	Ispravljen dijagram stanja	Paprskar	4.1.2021.
1.4.3	Ispravljen dijagram aktivnosti	Paprskar	4.1.2021.
1.4.4	Ispravljen tekst vezan uz dijagram aktivnosti	Paprskar	4.1.2021.
1.5	Dodano poglavlje Upute za puštanje u pogon	Paprskar	5.1.2021.
1.5.1	Nadopunjeno poglavlje Upute za puštanje u pogon	Paprskar	10.1.2021.
1.6	Dodano poglavlje Ispitivanje programskog rješenja	Paprskar	13.1.2021.
1.6.1	Pregledan/ažuriran/ispravljen dio dokumentacije	Paprskar	13.1.2021.
1.6.2	Ažuriran dijagram stanja i popratni opisni tekst	Paprskar	13.1.2021.
1.6.3	Ažurirani dijelovi dokumentacije	Paprskar	14.1.2021.
2.0	Dodano poglavlje Dijagrami pregleda promjena	Paprskar	14.1.2021.

## 2. Opis projektnog zadatka

Cilj projekta je razvijanje aplikacije za šahovski klub. Aplikacijom će se moći služiti:

- treneri
- članovi
- neregistrirani korisnici

Aplikacijom će upravljati pozz<sup>1</sup>. Jedna osoba može imati samo jednu od ove četiri uloge. Glavne koristi aplikacije su interaktivno učenje šaha, natjecanje protiv drugih članova kluba u turnirima i informiranje članova o aktivnostima i novostima u klubu. Članovi mogu preko aplikacije plaćati članarinu. Korisnici će učiti šah pomoću dnevnih šahovskih taktika, koje postavljaju treneri. Treneri mogu organizirati turnire, organizirati treninge i informirati članove pomoću stranice s novostima. Neregistrirani korisnici imat će ograničene mogućnosti kod korištenja aplikacije.

Korisnik koji prvi puta pokreće aplikaciju i nije registriran smjet će pristupiti novostima i informirati se o svim događanjima u klubu, rješavati dnevne šahovske taktike, ali neće biti dio rang lista, na kojima će biti popisani najuspješniji članovi koji su rješavali dnevne taktike. Status člana potreban je i za pristup organiziranim treninzima i turnirima. Ako je korisnik zainteresiran za učlanjenje u klub, smjet će napraviti korisnički račun, za koji su potrebni:

- korisničko ime
- e-mail
- lozinka.

Pri izradi računa korisnik se mora složiti s pravilima kluba (ne smije varati pomoću “chess engine”-a , ne smije koristiti tuđi korisnički račun ili dati svoj drugima na korištenje, ne smije se neprikladno ponašati prema trenerima ili drugim članovima...).

---

<sup>1</sup>administrator

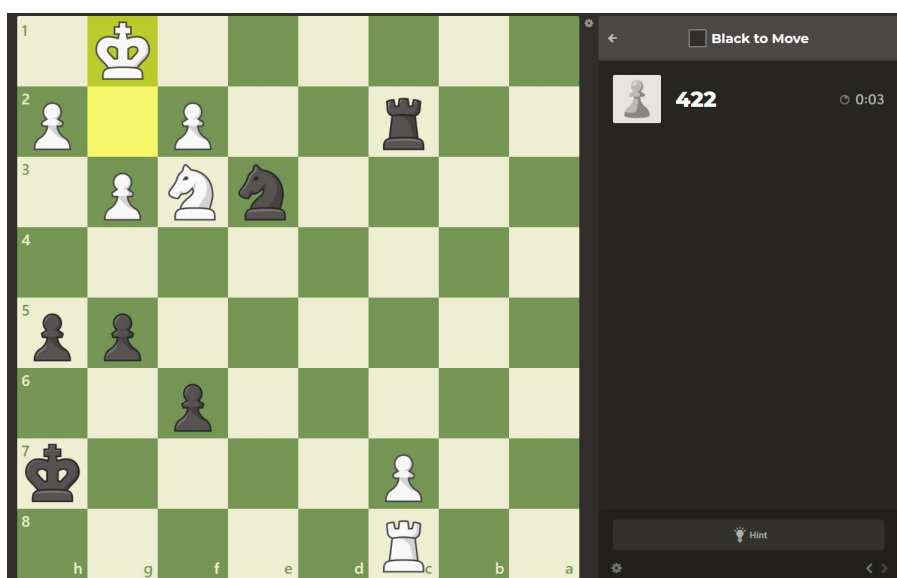
Nakon uspješne registracije, kako bi dobio status člana, registrirani korisnik mora platiti članarinu. Nakon ispunjavanja formulara u koji se upisuju ime i prezime člana, mjesec za koji se članarina plaća, dob, broj mobitela i e-mail adresa, ako je transakcija uspješno provedena administrator korisniku dodjeljuje status člana. Status člana bit će maknut ako korisnik prestane plaćati članarinu, tj. transakcija za određeni mjesec ne bude provedena i upisana u bazu podataka. Članovi uz rješavanje taktike imaju mogućnost dodijeliti ocjenu taktici ovisno o njenoj težini (1-5), a u slučaju da misle da postoji pogreška u taktici ili vide bolje rješenje, mogu prijaviti pogrešku. Rang liste za taktike rade se ovisno o vremenu koje je članu potrebno da točno riješi taktiku i težini taktike. U aplikaciji postoje rang liste koje prikazuju najbolje rješavače taktika. Nove rang liste postavljaju se svaki mjesec. U taktici je dozvoljen samo jedan poredak poteza, a u slučaju pogrešnog poteza rješavaču se dodaje vremenska kazna (1 minuta) za svaki pogrešan potez. Ako član misli da je uočio pogrešku u taktici, može je prijaviti tako da nakon pritiska na gumb za prijavu pogreške u otvorenu simulaciju taktike unese poteze za koje misli da su ispravni i dodatno obrazloži poteze u tekstualnoj kućici ako smatra da je potrebno. Ako član želi dodatno vježbati, može se prijaviti na organizirani trening kod trenera. Trener u opisu treninga navodi što će sve biti uključeno u njegov trening. Nakon vježbanja član može testirati svoju vještinu protiv drugih igrača u sklopu turnira. Na stranici turnira član može vidjeti raspored nadolazećih turnira i prijaviti se ako ima slobodnih mjesta. Turniri se održavaju u točno određeno vrijeme. Tada članovi koji su se prijavili igraju jedan protiv drugog sve dok se ne proglasi pobjednik (igre se ne odvijaju u sklopu aplikacije). Turniri se održavaju u različitim formatima:

- *rapid* (25 minuta po igraču)
- *blitz* (5 minuta po igraču)
- *bullet* (2 minute po igraču)

Na stranici turnira postavlja se očekivano ukupno vrijeme trajanja turnira. Treneri uz mogućnost organiziranja treninga i turnira mogu postavljati dnevne taktike ili objavljivati zanimljivosti i novosti. Ako član prijavi pogrešku u taktici, da bi došlo do promjene taktike, trener ju mora odobriti. Ako trener misli da je žalba valjana, on mijenja taktiku i automatski se oduzimaju bodovi za članove koji su riješili pogrešnu verziju taktike, te se omogućava ponovno rješavanje ispravljene

taktike. Najveće ovlasti na stranici ima administrator. On smije dodavati bilo kakav sadržaj, brisati sadržaj iz svih rubrika i zabraniti pristup bilo kojem članu i treneru. Prilikom brisanja sadržaja, on ostaje u bazi podataka, ali se ne prikazuje u aplikaciji. Administrator jedini može dodijeliti uloge drugim korisnicima.

Postoje aplikacije sa sličnim mogućnostima poput *chess.com* ili *lichess.org*. Tamo je moguće napraviti korisnički račun i vježbati na šahovskim taktikama na sličan način kao u našoj aplikaciji, ali te stranice nisu toliko fokusirane na rad kluba, nego na učenje i igranje šaha. Ovakve stranice pružaju razne mogućnosti poput igre jedan na jedan, učenja šahovskih otvaranja, igranja protiv „chess engine“-a, što naša aplikacija neće sadržavati. Naša aplikacija fokusira se na jedan šahovski klub i njegovu organizaciju.



Slika 2.1: Šahovska ploča preuzeta s druge stranice kao primjer drugačijeg rješenja

Za ovakvu aplikaciju mogli bi biti zainteresirani vlasnici šahovskog kluba koji žele svojim članovima pružiti više mogućnosti učenja šaha i lakši način praćenja novosti u klubu. Postojanje ovakve aplikacije je i u interesu svim šahistima koji žele interaktivnije načine sudjelovanja u klupskim aktivnostima. Vrlo je vjerojatno da bi šahisti odabrali klub koji pruža i online usluge umjesto kluba koji djeluje isključivo uživo.

Uz sve koristi koje aplikacija pruža, bitno je spomenuti i dosege. Projekt omogućava korištenje šahovske ploče za rješavanje dnevnih taktika, ali se ne omogućava online igra protiv drugih igrača. Dio aplikacije koji se bavi turnirima također ne omogućava



1 na 1 igranje, nego samo bilježi raspored turnira i omogućava članovima da se prijave na turnir. Organizirani treninzi ne omogućavaju treneru da direktno pomoću aplikacije igračima na ploči prikazuje lekcije, nego funkcioniraju poput rasporeda gdje članovi mogu vidjeti kada će se treninzi održati i kako će izgledati. Dnevne taktike ne omogućavaju igračima slobodno postavljanje figurica, nego se koncentriraju na točno određenu konfiguraciju ploče. Rješenja dnevnih taktika provjeravaju se protiv zadanog slijeda poteza, a ne "*chess engine*"-a (zbog toga se i omogućava prijava pogreške u taktici). Prijava pogreške ne sadrži dvosmjernu komunikaciju između člana koji se žali i trenera, nego se sastoji isključivo od žalbe i mogućeg odobrenja promjene. Aplikacija također ne omogućava dvosmjernu komunikaciju (chat) između članova ili trenera.

Moguće je nadograditi i proširiti razne aspekte naše aplikacije. Uz učenje na dnevnim taktikama moguće je dodati i analiziranje odigranih igara ili učenje šahovskih otvaranja. Aplikaciji je moguće dodati neki oblik komunikacije između korisnika (forum ili chat). Turniri bi se mogli održavati putem aplikacije ako se uz organizacijski aspekt doda i mogućnost igranja i bilježenja ratinga, koji bi se računao prema uspjehu u igranju protiv drugih članova kluba, a ne samo po uspjehu u šahovskim taktikama. Aplikacija bi mogla bilježiti statistike korisnika i prikazivati ih na njihovoj osobnoj stranici. Stranici s obavijestima bi bilo korisno dodati mogućnost komentiranja na objave. Ako se aplikacija želi značajnije proširiti, mogla bi obuhvaćati više šahovskih klubova.

Potencijal za nadogradnju ovog programa u budućnosti svakako bi trebao sadržavati mogućnost testiranja budućih članova i njihovo kategoriziranje u budućem članstvu kluba. To znači da bi budući član osim uplate članarine, koja bi mu omogućila ulaz i korištenje aplikacija šahovskog kluba, trebao proći kroz testne šahovske zadatke koji bi ga prema stupnju uspješnosti stavile u A, B ili C grupu. Primarno grupe ne bi „razbijale“ homogenost šahovskog kluba već bi svakom pojedincu ukazale na stupanj njegovog trenutnog znanja i omogućile kroz treninge i takmičenja njegov rast i napredak. Time bi se statistički moglo pratiti svaki pojedinac. Grupe bi bile određena interna kategorija članova unutar kluba. Samim članovima bio bi izazov pratiti svoju uzlaznu krivulju.

Osim redovne mjesečne članarine trebalo bi omogućiti, kao jednu od budućih nadogradnji kod uplate, opciju da članovi uplaćuju unaprijed za 6 ili 12 mjeseci. Time bi se osim određenih financijskih sredstava koje se korisno mogu utrošiti za razvoj kluba (materijalni i intelektualni troškovi), dobila i jedna stabilna baza

igrača na kojoj bi se mogao temeljiti razvoj, ali i opstojnost kluba. Sigurno članstvo osigurava budućnost kluba. Takvim članovima koji se odluče na dugoročnije uplate članarine treba osigurati popuste kod uplate, osigurati određeni reklamni materijal s logom kluba, omogućiti više drugih aktivnosti u samom programu.

Za pomoć pogledati reference navedene u poglavlju "Popis literature", a po potrebi konzultirati sadržaj na internetu koji nudi dobre smjernice u tom pogledu.

## 2.1 Motivacija za projekt

Ideja ovog je projekta olakšati šahovskom klubu obavljanje administrativnih poslova i komunikaciju s članovima. To bi se postiglo web aplikacijom s funkcionalnostima kao što su online uplaćivanje članarine, objavljivanje novosti, te prijavljivanje članova za treninge i turnire. Web aplikacija bila bi intuitivna i jednostavna za korištenje kako članovima, tako i zaposlenicima kluba. Osim toga, članovi bi se putem web aplikacije mogli nadmetati u rješavanju dnevnih taktika, te tako sudjelovati u šahovskoj zajednici čak i van fizičkog prostora kluba.

Još jedan bitan aspekt ove aplikacije bio bi njen potencijal za nadogradnju. Jasno je da će u budućnosti potrebe šahovskog kluba rasti i da će se trebati uvesti nove funkcionalnosti u web aplikaciju. Tu činjenicu mora odražavati kvaliteta programskog rješenja i dokumentacije kako bi se pružio dobar temelj za nova proširenja.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Vlasnik (naručitelj)
2. Zaposlenici šahovskog kluba
  - (a) Treneri
3. Članovi šahovskog kluba
4. Neregistrirani korisnici aplikacije
5. Administrator
6. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Trener (inicijator) može:
  - (a) prijaviti se u aplikaciju putem korisničkog imena i lozinke
  - (b) odjaviti se iz aplikacije
  - (c) postavljati dnevne šahovske taktike
  - (d) revidirati prijavljene pogrešne taktike
  - (e) slagati raspored vlastitih treninga
  - (f) organizirati turnire
  - (g) vidjeti rang liste članova
  - (h) pristupiti i objavljivati novi sadržaj na stranici novosti
  - (i) vidjeti osobne podatke i aktivnosti svojeg profila
2. Član šahovskog kluba (inicijator) može:
  - (a) prijaviti se u aplikaciju putem korisničkog imena i lozinke
  - (b) odjaviti se iz aplikacije
  - (c) platiti članarinu putem aplikacije
  - (d) prijavljivati se na treninge kod pojedinih trenera
  - (e) prijavljivati se na turnire

- (f) rješavati dnevne šahovske taktike
  - i. nakon rješavanja dnevne šahovske taktike mogu joj dodijeliti ocjenu
  - ii. nakon rješavanja dnevne šahovske taktike mogu prijaviti grešku u taktici
- (g) vidjeti rang liste članova
- (h) pristupiti stranici novosti
- (i) vidjeti osobne podatke i aktivnosti svojeg profila

3. Administrator (inicijator) može:

- (a) prijaviti se u aplikaciju putem korisničkog imena i lozinke
- (b) odjaviti se iz aplikacije
- (c) potpuno zabraniti pristup bilo kojem članu ili treneru
- (d) zabraniti pristup bilo čemu **osim** uplate članarine bilo kojem članu
- (e) vidjeti rang liste članova
- (f) objavlјivati i skidati sadržaj na stranici novosti
- (g) mijenjati raspored treninga bilo kojem treneru
- (h) postavljati i skidati dnevne šahovske taktike
  - i. nakon rješavanja dnevne šahovske taktike mogu joj dodijeliti ocjenu
  - ii. nakon rješavanja dnevne šahovske taktike mogu prijaviti grešku u taktici
- (i) dodavati i skidati turnire
- (j) pregledavati transakcije
- (k) vidjeti osobne podatke i aktivnosti svojeg profila

4. Neregistrirani korisnik (inicijator) može:

- (a) registrirati se
- (b) rješavati dnevne šahovske taktike
- (c) vidjeti rang liste članova
- (d) pristupiti novostima

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve dnevne šahovske taktike
- (c) pohranjuje rang listu članova
- (d) pohranjuje povijest svih transakcija
- (e) pohranjuje termine svih treninga

- (f) pohranjuje termine svih turnira
- (g) pohranjuje svaku stavku na stranici novosti

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik otvara sučelje za registraciju
  2. Korisnik unosi sve potrebne podatke za registraciju
  3. Korisnik prima obavijest o uspješnosti registracije
- **Opis mogućih odstupanja:**
  - 2.a Unos već postojećeg korisničkog imena i/ili e-maila, unos nepostojećeg e-maila, unos podatka u nedozvoljenom formatu.
    1. Sučelje obavještava korisnika o pogrešci u registraciji i vraća ga na stranicu za registraciju.
    2. Korisnik mijenja podatke i pokušava ponovo ili odustaje od registracije.

##### UC2 - Pregled dnevnih taktika

- **Glavni sudionik:** Neregistrirani korisnik, član, administrator
- **Cilj:** Pregledati dostupne taktike za rješavanje tog dana
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik otvara stranicu s popisom dnevnih taktika
  2. Korisnik odabire taktiku koju želi rješavati
  3. Pokreće se simulacija šahovske ploče s odabranom taktikom

##### UC3 - Rješavanje dnevne taktike

- **Glavni sudionik:** Neregistrirani korisnik, član, administrator
- **Cilj:** Uspješno riješiti odabranu dnevnu taktiku
- **Sudionici:** -
- **Preduvjet:** -

- **Opis osnovnog tijeka:**
  1. Korisniku se pokreće simulacija šahovske ploče
  2. Korisnik rješava dnevnu taktiku
  3. Simulacija se zatvara po uspješnom rješavanju
- **Opis mogućih odstupanja:**
  - 2.a Korisnik napravi pogrešan potez
    1. Simulacija obavještava korisnika o pogrešnom potezu i vraća ploču u stanje prije pogrešnog poteza.
    2. Korisnik pokušava s novim potezom.
  - 3.a Korisnik želi ocijeniti dnevnu taktiku
    1. Korisnik odabire opciju "Unesi ocjenu"
    2. Korisnik potvrđuje odabranu ocjenu ili odustaje od davanja ocjene
  - 3.b Korisnik želi prijaviti pogrešku u taktici
    1. Član odabire opciju "Prijavi pogrešku u taktici"
    2. Korisnik unosi nove poteze i opis novih poteza
    3. Korisnik potvrđuje unesene informacije ili odustaje od unosa novih poteza i njihovih opisa

#### UC4 - Pregled rang liste

- **Glavni sudionik:** Korisnik, član, trener, administrator
- **Cilj:** Pregledati rang listu članova kluba
- **Sudionici:** Baza podataka
- **Preduvjet:** Ako je korisnik prijavljen kao član mora imati tekuću članarinu
- **Opis osnovnog tijeka:**
  1. Korisnik otvara stranicu s rang listom
  2. Prikazuje se rang lista članova kluba

#### UC5 - Odjava

- **Glavni sudionik:** Administrator, član, trener
- **Cilj:** Okončati aktivnu sjednicu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Klik na gumb 'Odjavi se'
  2. Učitavanje stranice novosti

### UC6 - Objava dnevne šahovske taktike

- **Glavni sudionik:** Trener, administrator
- **Cilj:** Stvoriti novu dnevnu šahovsku taktiku
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Otvaranje stranice za unos nove taktike
  2. Trener unosi podatke o taktici poput imena, težine itd.
  3. Trener namješta inicijalnu konfiguraciju ploče na interaktivnoj simulaciji
  4. Trener potvrđuje inicijalnu konfiguraciju
  5. Trener naizmjenice pomiče crne i bijele šahovske figurice i tako unosi poteze taktike
  6. Trener potvrđuje korektan unos poteza i objavljuje dnevnu taktiku
- **Opis mogućih odstupanja:**
  - 3.a Zabuna u namještanju inicijalne konfiguracije ploče
    1. klik na gumb 'reset'
    2. ploča se vraća na startnu konfiguraciju
  - 5.a Zabuna u unosu poteza taktike
    1. klik na gumb 'reset'
    2. ploča se vraća na inicijalnu konfiguraciju unesenu u prošlom koraku

### UC7 - Pregled vlastitih treninga

- **Glavni sudionik:** Trener
- **Cilj:** Prikazati sve buduće treninge prijavljenog trenera
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Klik na karticu 'Treninzi'
  2. Kronološki prikaz svih budućih treninga

### UC8 - Brisanje treninga

- **Glavni sudionik:** Trener, Administrator
- **Cilj:** Obrisati postojeći trening
- **Sudionici:** Baza podataka



- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. klik na gumb za brisanje jednog od prikazanih treninga
  2. prikaz dijaloškog okvira koji traži potvrdu od korisnika da uistinu želi obrisati trening
  3. nakon davanja potvrde trening nestaje iz prikaza
- **Opis mogućih odstupanja:**
  - 3.a Korisnik se predomisli o brisanju treninga
    1. pritisak na gumb 'odustani' u dijaloškom okviru za potvrdu

#### **UC9 - Stvaranje novog treninga**

- **Glavni sudionik:** Trener, Administrator
- **Cilj:** Stvoriti novi trening
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. klik na gumb za dodavanje novog treninga
  2. prikaz dijaloškog okvira u koji korisnik unosi podatke o treningu kao vrijeme, naziv, izvoditelj itd.
  3. klik na gumb za stvaranje treninga
  4. zatvara se dijaloški okvir i novi trening dodaje se u prikaz treninga
- **Opis mogućih odstupanja:**
  - 2.a Trener dodaj podatke
    1. trener ne može birati izvoditelja nego je on sam zaključan kao izvoditelj prilikom stvaranja
  - 3.a Odustajanje od stvaranja novog treninga
    1. pritisak na gumb 'odustani' u dijaloškom okviru za unos podataka o treningu
    2. zatvaranje dijaloškog okvira i odbacivanje unesenih podataka

#### **UC10 - Pregled turnira**

- **Glavni sudionik:** Trener, administrator, član
- **Cilj:** Prikazati sve buduće turnire
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**

1. Klik na karticu 'Turniri'
2. Kronološki prikaz svih budućih turnira

### **UC11 - Brisanje turnira**

- **Glavni sudionik:** Trener, administrator
- **Cilj:** Obrisati postojeći turnir čiji je organizator prijavljeni korisnik
- **Sudionici:** Baza podataka
- **Preduvjet:** Ako je korisnik prijavljen kao trener, može brisati samo svoje turnire
- **Opis osnovnog tijeka:**
  1. klik na gumb za brisanje jednog od prikazanih turnira
  2. prikaz dijaloškog okvira koji traži potvrdu od korisnika da uistinu želi obrisati turnir
  3. nakon davanja potvrde turnir nestaje iz prikaza
- **Opis mogućih odstupanja:**
  - 3.a Korisnik se predomisli o brisanju turnira
    1. pritisak na gumb 'odustani' u dijaloškom okviru za potvrdu

### **UC12 - Stvaranje novog turnira**

- **Glavni sudionik:** Trener, administrator
- **Cilj:** Stvoriti novi turnir
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. klik na gumb za dodavanje novog turnira
  2. prikaz dijaloškog okvira u koji korisnik unosi podatke o turniru kao vrijeme, naziv, izvoditelj itd.
  3. klik na gumb za stvaranje turnira
  4. zatvara se dijaloški okvir i novi turnir dodaje se na stranicu
- **Opis mogućih odstupanja:**
  - 2.a Trener dodaj podatke
    1. trener ne može birati izvoditelja nego je on sam zaključan kao izvoditelj prilikom stvaranja
  - 3.a Odustajanje od stvaranja novog turnira
    1. pritisak na gumb 'odustani' u dijaloškom okviru za unos podataka
    2. zatvaranje dijaloškog okvira i odbacivanje unesenih podataka

**UC13 - Pristup stranici novosti**

- **Glavni sudionik:** Član, trener, administrator, neregistrirani korisnik
- **Cilj:** Vidjeti sve nedavne novosti
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. klik na karticu 'novosti'
  2. prikaz svih nedavnih novosti

**UC14 - Objava novosti**

- **Glavni sudionik:** Trener, administrator
- **Cilj:** Objaviti novost na stranici novosti
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. klik na gumb za objavu novosti
  2. ispunjavanje dijaloškog okvira za stvaranje novosti
  3. klik na gumb za objavu
  4. novost se prikazuje na vrhu stranice novosti
- **Opis mogućih odstupanja:**
  - 3.a Odustajanje od stvaranja novosti
    1. pritisak na gumb 'odustani' u dijaloškom okviru za unos podataka
    2. zatvaranje dijaloškog okvira i odbacivanje unesenih podataka

**UC15 - Revidiranje pogreške u taktici**

- **Glavni sudionik:** Trener
- **Cilj:** Prihvatiti ili odbaciti dojavu o pogrešci u nekoj taktici
- **Sudionici:** Baza podataka
- **Preduvjet:** Član je prijavio pogrešku u taktici
- **Opis osnovnog tijeka:**
  1. otvara se pregled s detaljima o dojavi pogreške na taktici
  2. Trener klikom na gumb potvrđuje da je dojava o pogrešci valjana
  3. taktika u pitanju automatski se revidira te se ažuriraju i rang liste članova
- **Opis mogućih odstupanja:**
  - 2.a Odbijanje dojave

1. pritiskom na gumb 'odbaci' dojava se zanemaruje

### UC16 - Pregled profila

- **Glavni sudionik:** Administrator, član, trener
- **Cilj:** pregled informacija vlastitog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. klik na karticu 'profil'
  2. učitava se prikaz osobnih podataka i svih aktivnosti na aplikaciji

### UC17 - Prijava u korisnički račun

- **Glavni sudionik:** Član, trener, administrator
- **Cilj:** Prijaviti se u korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik otvara sučelje za prijavu u korisnički račun
  2. Korisnik upisuje korisničko ime i lozinku
  3. Korisniku se otvara stranica s novostima
- **Opis mogućih odstupanja:**
  - 3.a Korisnik koji ima status člana nema plaćenu članarinu
    1. Korisniku se otvara stranica za uplatu članarine

### UC18 - Uplata članarine

- **Glavni sudionik:** Član
- **Cilj:** Uplatiti članarinu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Član odabire opciju "Uplata članarine"
  2. Članu se otvara sučelje s mogućnošću upisa potrebnih podataka za uplatu članarine
  3. Član upisuje potrebne podatke za uplatu i potvrđuje svoj unos
  4. Član dobiva obavijest da je transakcija uspješna
- **Opis mogućih odstupanja:**

3.a Član je unio netočne podatke

1. Članu dolazi obavijest o netočnom unosu podataka te mu se omogućuje da ponovno unese podatke

#### **UC19 - Pregled dostupnih treninga**

- **Glavni sudionik:** Član
- **Cilj:** Pregledati treninge na koje se može prijaviti
- **Sudionici:** Baza podataka
- **Preduvjet:** Član ima tekuću članarinu
- **Opis osnovnog tijeka:**
  1. Član odabire opciju "Pregledaj treninge"
  2. Članu se prikazuje popis dostupnih treninga

#### **UC20 - Prijava na trening**

- **Glavni sudionik:** Član
- **Cilj:** Prijaviti se na trening kod odabranog trenera
- **Sudionici:** Baza podataka
- **Preduvjet:** Član ima tekuću članarinu
- **Opis osnovnog tijeka:**
  1. Član odabire opciju "Prijavite trening"
  2. Članu se prikazuje popis raspoloživih trenera
  3. Član odabire jednog od ponuđenih trenera
  4. Član odabire jedan od raspoloživih termina treninga
  5. Član dobiva obavijest da je željeni termin treninga kod odabranog trenera zabilježen

#### **UC21 - Prijava na turnir**

- **Glavni sudionik:** Član
- **Cilj:** Prijaviti se na turnir
- **Sudionici:** Baza podataka
- **Preduvjet:** Član ima tekuću članarinu
- **Opis osnovnog tijeka:**
  1. Član odabire opciju "Prijava na turnir"
  2. Članu se javlja obavijest o terminu sljedećeg turnira te poruka o nužnosti potvrde za prijavu na turnir uz mogućnost odabira opcije "Potvrđujem prijavu" ili opcije "Odustajem od prijave"

3. Član odabire opciju "Potvrđujem prijavu"
4. Član dobiva obavijest da je prijava zabilježena
- **Opis mogućih odstupanja:**
  - 2.a Član odustaje od prijave
    1. Član odabire opciju "Odustajem od prijave"

#### UC22 - Brisanje novosti

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati novost sa stranice novosti
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Administrator se postavlja na listu novosti
  2. Administrator bira opciju "Obriši obavijest" pored obavijesti koju želi obrisati
  3. Administrator odabire opciju "Da" u dijaloškom okviru za potvrdu
- **Opis mogućih odstupanja:**
  - 3.a Administrator odustaje od brisanja novosti
    1. Administrator odabire opciju "Ne" u dijaloškom okviru za potvrdu

#### UC23 - Zabrana pristupa treneru ili članu

- **Glavni sudionik:** Administrator
- **Cilj:** Zabraniti pristup treneru ili članu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Administrator se postavlja na listu registriranih korisnika
  2. Administrator bira opciju "Zabrani pristup svemu" pored člana ili trenera kojem želi zabraniti pristup
  3. Administrator odabire opciju "Da" u dijaloškom okviru za potvrdu
- **Opis mogućih odstupanja:**
  - 3.a Administrator odustaje od zabrane pristupa
    1. Administrator odabire opciju "Ne" u dijaloškom okviru za potvrdu

#### UC24 - Pregled transakcija

- **Glavni sudionik:** Administrator

- **Cilj:** Pregledati obavljene transakcije
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Administrator klikne na karticu "Transakcije"
  2. Na stranici se prikažu sve transakcije

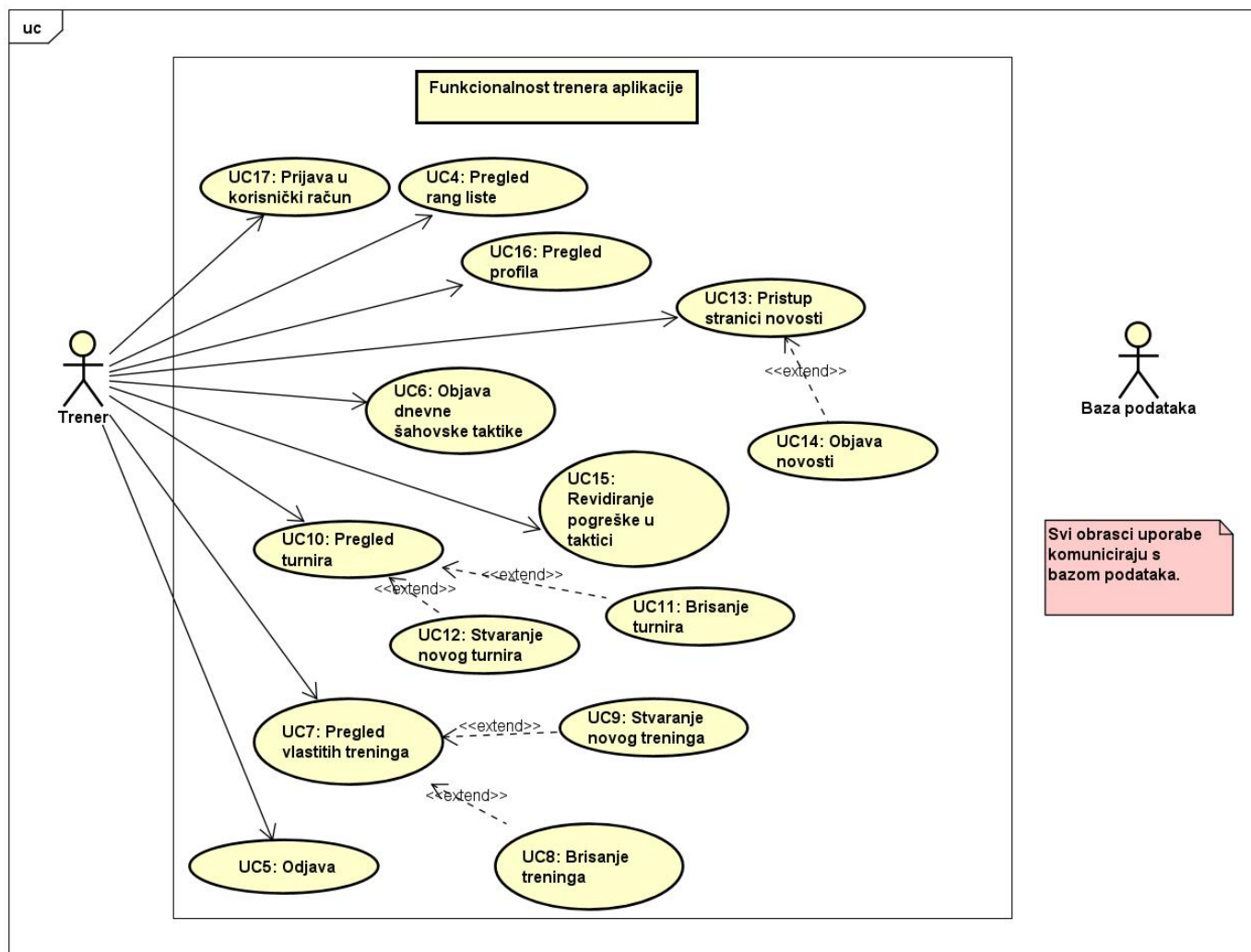
#### UC25 - Zabrana pristupa članu osim plaćanja članarine

- **Glavni sudionik:** Administrator
- **Cilj:** Zabraniti pristup članu svemu osim plaćanju članarine
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Administrator se postavlja na listu registriranih korisnika
  2. Administrator bira opciju "Zabrani pristup svemu osim plaćanju članarine" pored člana kojem želi zabraniti pristup
  3. Administrator odabire opciju "Da" u dijaloškom okviru za potvrdu
- **Opis mogućih odstupanja:**
  - 3.a Administrator odustaje od zabrane pristupa svemu osim plaćanju članarine
    1. Administrator odabire opciju "Ne" u dijaloškom okviru za potvrdu

#### UC26 - Brisanje dnevne šahovske taktike

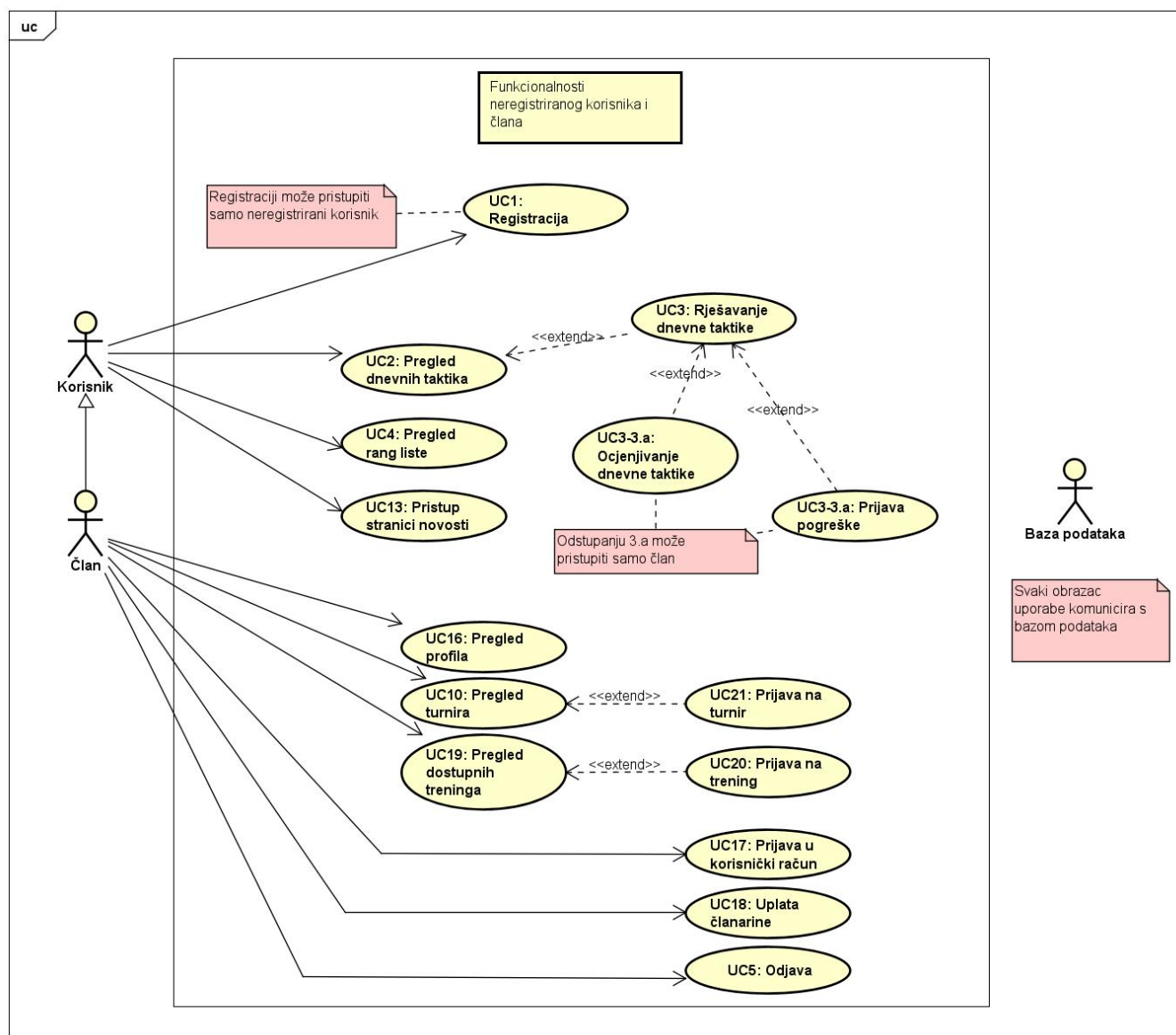
- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati dnevnu šahovsku taktiku
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Administrator se postavlja na listu dnevnih šahovskih taktika
  2. Administrator bira opciju "Obriši dnevnu šahovsku taktiku" pored dnevne šahovske taktike koju želi izbrisati
  3. Administrator odabire opciju "Da" u dijaloškom okviru za potvrdu
- **Opis mogućih odstupanja:**
  - 3.a Administrator odustaje od brisanja dnevne šahovske taktike
    1. Administrator odabire opciju "Ne" u dijaloškom okviru za potvrdu

## Dijagrami obrazaca uporabe

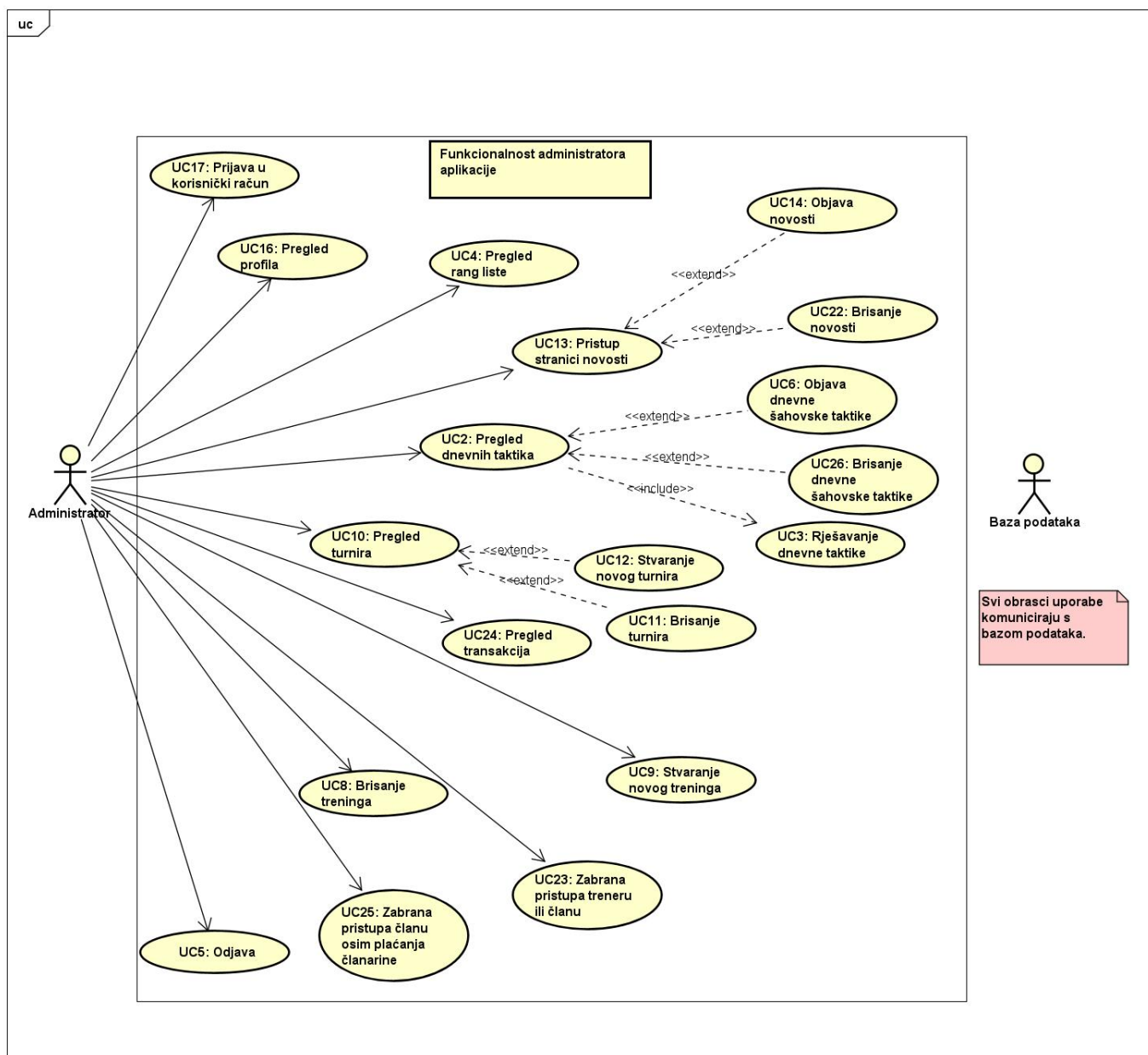


Slika 3.1: Dijagram obrazaca uporabe za trenera





Slika 3.2: Dijagram obrazaca uporabe za neregistriranog korisnika i člana

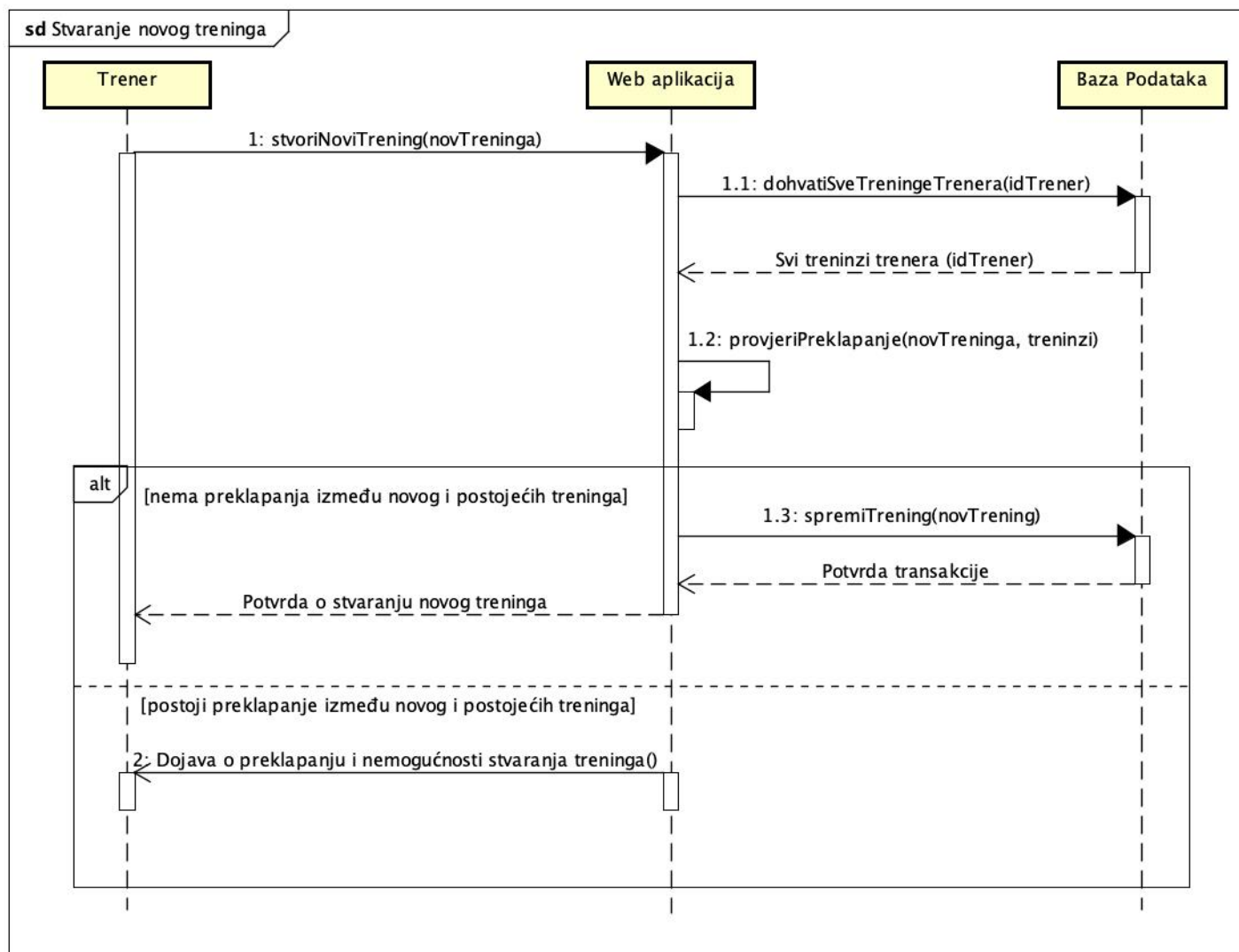


Slika 3.3: Dijagram obrazaca uporabe za administratora

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC9 - Stvaranje novog treninga

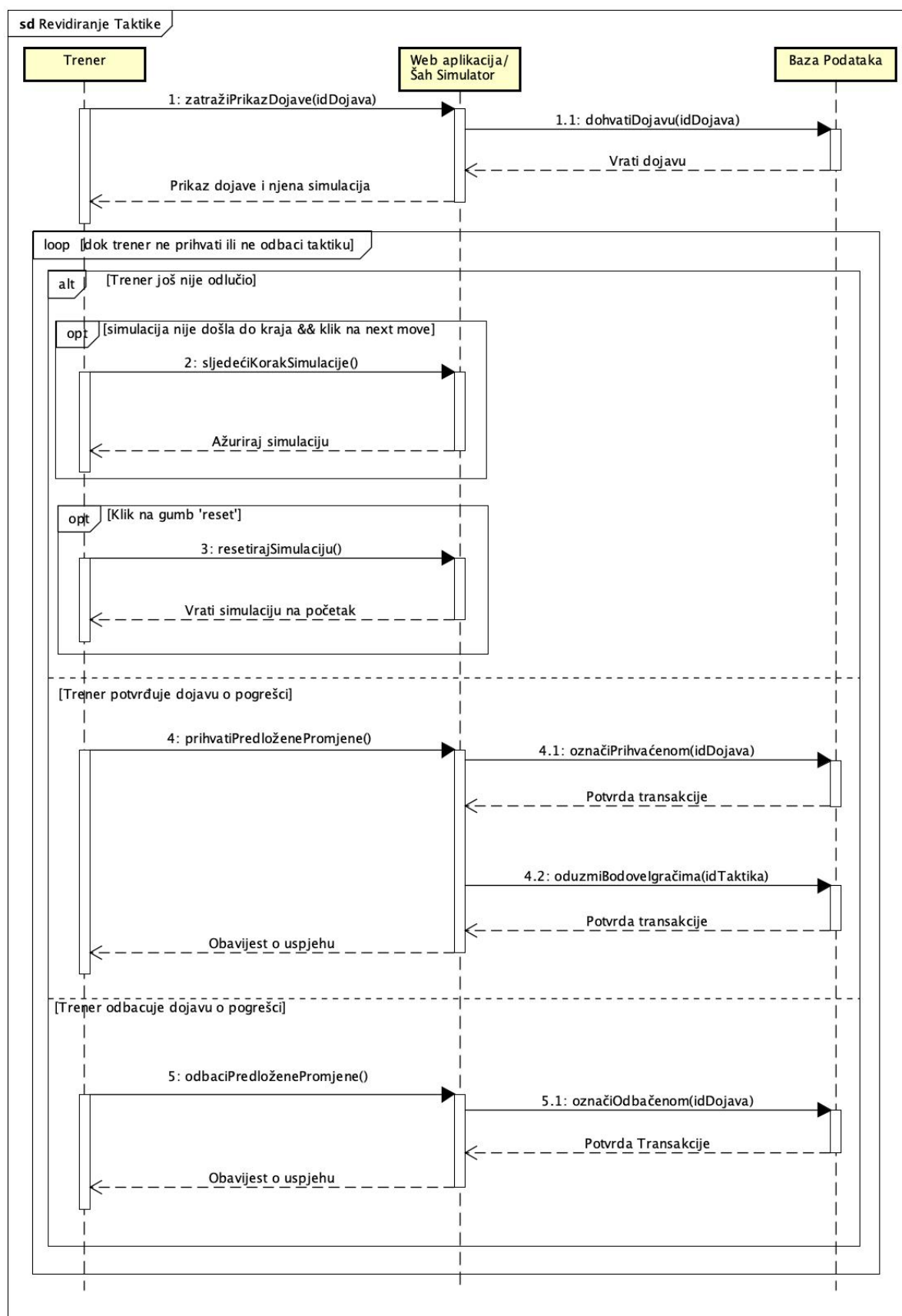
Trener šalje zahtjev web aplikaciji za stvaranje novog treninga. Web aplikacija dohvaća iz baze podataka sve ostale treninge tog trenera i provjerava da se novi trening ne preklapa ni s jednim postojećim treningom. Ako uistinu ne postoji preklapanje, novi trening se zapisuje u bazu podataka te se treneru šalje obavijest o uspjehu kreiranja novog treninga. Ako postoji preklapanje s već postojećim treningom, treneru web aplikacija javlja informacije o preklapanju i nemogućnosti stvaranja zatraženog treninga.



Slika 3.4: Sekvencijski dijagram za UC9

**Obrazac uporabe UC15 - Revidiranje pogreške u taktici**

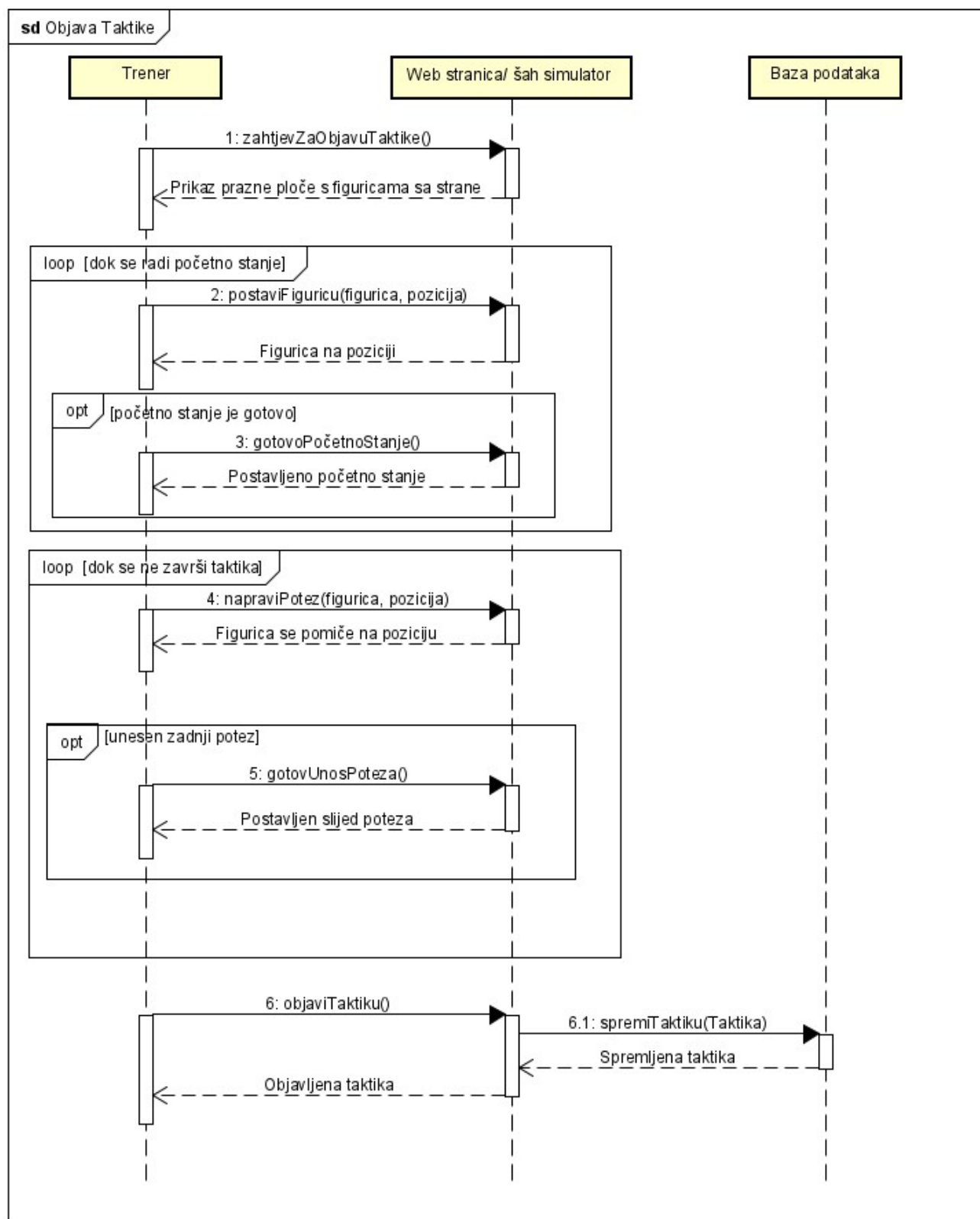
Trener šalje zahtjev za prikaz određene dojave o pogrešci taktike kako bi ju mogao pobliže proučiti i odlučiti je li primjedba valjana. Poslužitelj zatraženu dojavu vadi iz baze podataka te ju prikazuje. Taj prikaz sastoji se od osnovnih informacija o dojavi poput ime prijavljene taktike, težine prijavljene taktike itd., ali se sastoji i od simulacije dvije šahovske ploče. Na prvoj ploča simulira se trenutni tijek taktike, a na drugoj se simulira novi tijek taktike koji je predložen u dojavi o pogrešci. Trener pritiskom na odgovarajuće gumbе šalje zahtjeve za sljedeći korak u simulaciji. Trener također može i poslati zahtjev za resetiranje simulacije i tako ju vratiti u početni položaj. Nakon što je proučio simulaciju trener šalje poslužitelju zahtjev za potvrdu ili odbacivanje dojave o pogrešci. Ako je trener potvrdio dojavu, poslužitelj ju tako mora obilježiti u bazi podataka, te zatim oduzeti odgovarajući broj bodova svim igračima koji su uspješno riješili 'zastarjelu' verziju taktike. Ako je trener odbacio dojavu, poslužitelj ju samo mora evidentirati kao odbačenu u bazi podataka.



Slika 3.5: Sekvencijski dijagram za UC15

**Obrazac uporabe UC6 - Objava taktike**

Trener odabire kreaciju (objavu) nove taktike te mu web aplikacija vraća prikaz prazne ploče s figuricama sa strane. Trener zatim kreće s postavljanjem početnog stanja ploče. Mišem može povući željene figurice na njihovo mjesto te mu aplikacija to prikazuje. Postupak se ponavlja sve dok trener ne odluči da je početno stanje pripremljeno. To se javlja aplikaciji te se kreće u stanje rješavanja taktike. Trener odabire točne poteze bijelog i crnog igrača te se ti potezi prikazuju. Nakon što je unesen posljednji potez, trener zaključava unos poteza te objavljuje taktiku. Aplikacija šalje bazi podataka zahtjev za spremanjem taktike te prikazuje treneru poruku da je taktika objavljena.

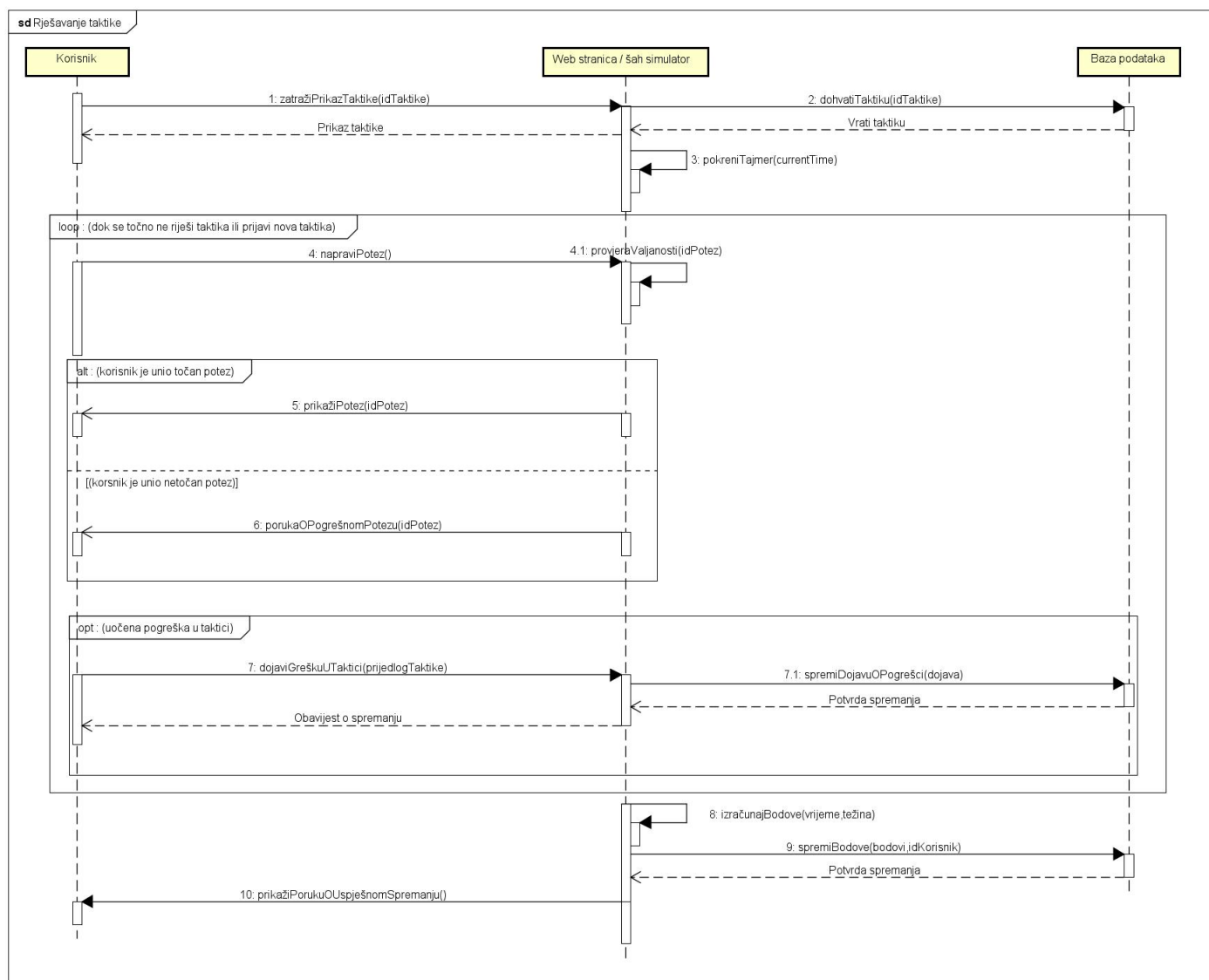


Slika 3.6: Sekvencijski dijagram za UC6



**Obrazac uporabe UC3 - Rješavanje taktike**

Korisnik otvara taktiku koja se dohvaća iz baze podataka. Korisnik zatim rješava tu taktiku te se pokreće tajmer. U slučaju krivog poteza, korisnik je obaviješten. Ako korisnik smatra da je taktika pogrešna, može spremiti svoju taktiku koja se šalje u bazu podataka. Nakon točnog rješavanja taktike, aplikacija računa postignute bodove na temelju vremena rješavanja i težine taktike. Bodovi zajedno s korisnikovim ID-om se spremaju u bazu podataka.



Slika 3.7: Sekvencijski dijagram za UC3

## 3.2 Ostali zahtjevi

Sustav mora osigurati da odgovor na svaki zahtjev dođe unutar 5 sekundi. Sustav mora isto tako osiguravati da korisnici imaju sigurne šifre za svoje korisničke račune kako bi što bolje poboljšao sigurnost sustava. Sustav će koristiti protokol HTTP za komunikaciju između korisnika i servera. Sustav mora osigurati da svaka osoba ima samo jednu ulogu u aplikaciji. Sustav mora osigurati da korisnici ne mogu raditi ilegalne poteze u šahu.

Sustav mora osigurati da korisnik mora unijeti sve potrebne podatke.

Treneri prilikom stvaranja novog turnira mora unijeti naziv, datum i vrijeme početka te opis turnira. Treneri prilikom stvaranja novog termina treninga ne smije unijeti vrijeme koje mu je već zauzeto drugim terminom. Trener prilikom unosa nove novosti mora osigurati da novost ima naslov i sadržaj.

Administrator prilikom stvaranja novog turnira mora unijeti naziv, datum i vrijeme početka, opis turnira te mora navesti trenera kojeg će sustav prepoznati kao organizatora turnira, pošto će tom treneru biti dane organizatorske ovlasti nad turnirom. Administrator prilikom stvaranja novog termina treninga mora navesti trenera kojem stvara termin te ne smije unijeti vrijeme koje je tom treneru već zauzeto drugim terminom.

Član mora prilikom prijave pogreške u dnevnoj taktici mora unijeti poteze koje smatra ispravnim te opis tih poteza.

Sustav mora mjeriti svakom članu vrijeme potrebno za rješavanje dnevne taktike te koristeći izmjereno vrijeme i ocjenu težine taktike dodijeliti članu bodove za tu taktiku. Sustav na bazi zbroja svih bodova, kreira težinsku listu rang listu članova. Korisničko sučelje će podržavati samo engleski jezik. Korisničko sučelje mora biti intuitivno za korištenje.

## 4. Arhitektura i dizajn sustava

Izbor odgovarajuće arhitekture programske poruke jedan je od najbitnijih koraka u oblikovanju sustava jer ona predstavlja poveznicu između zahtjeva na sustav i same implementacije sustava. Dobra arhitektura povlači dobru fleksibilnost sustava, jednostavnu mogućnost nadogradnje i jeftino održavanje.

S obzirom na to da je zadatak ovog projekta napraviti aplikaciju za šahovski klub, logičan izbor je web aplikacija. Primarni razlog je to što web aplikacija ne ovisi o platformi, već radi na svakom sustavu koji ima web preglednik što znatno smanjuje vrijeme i troškove potrebne za razvoj za više platformi

Radni okvir koji smo odabrali za arhitekturu je Django (pisan u i za jezik Python). S njime smo arhitekturu sustava odlučili temeljiti na MVT (Model-View-Template) konceptu kojeg on nativno podržava. Jedina bitna razlika između njega i poznatog MVC (Model-View-Controller) koncepta je to što u Django sam po sebi sređuje "controller" dio (programski kod koji kontrolira interakciju između modela i viewa) i daje nam na raspolaganje Template.

Dakle aplikacija se sastoji od 3 dijela:

- Model - Sređuje apstrakciju podataka, služi kao sučelje za podatke spremjene u bazu i dopušta upravljanje njima bez prevelikog razumijevanja kompleksnosti same baze
- View - funkcionalnosti slične controlleru u MVC konceptu. Sređuje svu logiku koja se treba prikazati na templateu i služi kao "most" između modela i templatea
- Template - funkcionalnosti slične Viewu u MVC konceptu. Služi kao sloj prikaza sadržaja i zadužen je za to kako i što će biti prikazano korisniku. Specifično za Django, Template je vrsta HTML datoteke koja može koristiti Django Template Language (DTL) što nam znatno olakšava komunikaciju između frontenda i backenda aplikacije.

## 4.1 Baza podataka

Za sustav baza podataka za ovaj projekt smo odabrali PostgreSQL.

PostgreSQL je objektno-relacijski sustav upravljanja bazama podataka (ORD-BMS) zasnovan na POSTGRES-u, inačica 4.2, razvijen na Kalifornijskom sveučilištu na Odjelu za računalne znanosti Berkeley. POSTGRES je pionir mnogih koncepata koji su tek kasnije postali dostupni u nekim komercijalnim sustavima baza podataka.

PostgreSQL je "open-source" potomak ovog izvornog Berkeley koda. Podržava velik dio SQL standarda i nudi brojne moderne značajke poput:

- složenih upita,
- stranih ključeva,
- okidača,
- obnovljivih pogleda,
- transakcijski integritet te
- multiverzijski protokol kontrole istodobnog pristupa

Također, korisnik može proširiti PostgreSQL na mnogo načina, na primjer dodavanjem novih:

- vrsta podataka,
- funkcija,
- operatora,
- agregatnih funkcija,
- indeksnih metoda te
- proceduralnih jezika.

A zbog liberalne licence, PostgreSQL može koristiti, mijenjati i distribuirati bilo tko u bilo koju svrhu, bilo privatnu, komercijalnu ili akademsku.<sup>1</sup>

U relacijskim bazama podataka, osnovne gradivne jedinice su tablice. Tablice koje grade bazu podataka ovog projekta su:

- taktika
- rjesenjeTaktika
- dojavaPogreske
- view - rangLista

---

<sup>1</sup><https://www.postgresql.org/docs/current/intro-what-is.html>

- Korisnik
- Grupa
- Novost
- Trening
- turnir
- PrijavaTrening
- PrijavaTurnir
- Transakcija

### 4.1.1 Opis tablica

Primarni ključevi tablice označeni su sa slovom 'P', a strani ključevi sa slovom 'F'.

Entitet **Taktika** u odnosu je *One to Many* s entitetima **Rješenje taktika** i **Dojava pogreške**.

taktika		
idTaktika P	INT	identifikacijski broj taktike
createdAt	TIMESTAMP	vrijeme stvaranja taktike, služi za kronološko sortiranje taktika
idUser F	INT	identifikator trenera ili admina koji je stvorio taktiku
initConfig	VARCHAR (100)	string koji opisuje inicijalno stanje šahovske ploče
movesWhite	VARCHAR (3000)	string koji opisuje poteze bijelih figurica
movesBlack	VARCHAR (3000)	string koji opisuje poteze crnih figurica
tezina	DECIMAL	težina taktike, koristi se u računanju bodova, u rasponu od 1 do 3
brojGlasova	INT	broj korisnika koji su glasali za neku težinu taktike

Entitet **Rješenje taktika** u odnosu je *One to One* s entitetnom **Taktika**, te u odnosu *One to One* s entitetnom **Korisnik**.

rjesenjeTaktika		
idTaktika PF	INT	identifikacijski broj taktike koja je riješena, foreign key
idUser PF	INT	identifikator člana koji je riješio taktiku, foreign key
vrijeme	DECIMAL	vrijeme rješavanja taktike u minutama

Entitet **Dojava pogreške** u odnosu je *One to One* s entitetnom **Taktika**, te u odnosu *One to One* s entitetnom **Korisnik** koji je prijavio taktiku. Također je u odnosu *One to One* s entitetnom **Korisnik** koji predstavlja trenera koji je zadužen za revidiranje dojava.

dojavaPogreske		
idDojava P	INT	identifikacijski broj dojava o pogrešci
idTaktika F	INT	identifikacijski broj prijavljene taktike
idUserDojavio F	INT	identifikator člana koji je dojavio pogrešku na taktiku
idUserRevidira F	INT	identifikator trenera koji revidira dojavu
prihvacena	BOOLEAN	je li dojava prihvaćena ili odbačena, NULL znači da čeka na revidiranje
predlozeniTijek	VARCHAR (6000)	string koji opisuje poteze koje je predložio član koji je dojavio pogrešku
opisPoteza	TEXT	opis poteza koje je član unio

Entiteti u tablici **rangLista** u odnosu su *One to One* s entitetnom **Korisnik**.

view - rangLista		
idUser PF	INT	identifikacijski broj člana
username	VARCHAR	korisničko ime člana
bodovi	INT	broj bodova koje je član sveukupno osvojio

Entitet **Korisnik** u odnosu je *One to Many* s entitetima **Rješenje taktika**, **Grupa** i **Dojava pogreške**, te u odnosu *One to One* s entitetima iz tablice **rangLista**.

Korisnik		
idUser P	INT	identifikacijski broj člana
idGrupa F	INT	identifikacijski broj grupe
username	VARCHAR (20)	korisničko ime člana
password	VARCHAR (20)	lozinka člana
email	VARCHAR (20)	mail adresa člana
ime	VARCHAR (20)	ime člana
prezime	VARCHAR (20)	prezime člana



Grupa		
idGrupa P	INT	identifikacijski broj grupe
imeGrupe	VARCHAR (20)	ime grupe

Entitet **Novost** u odnosu je *Many to One* s entitetom **Korisnik**

Novost		
idUser PF	INT	identifikacijski broj autora
vrijemeObjavljivanja P	DATETIME	vrijeme i datum objavljivanja
naslov	VARCHAR (50)	naslov novosti
tekst	TEXT	tekst novosti

Entitet **Trening** u odnosu je *One to Many* s entitetom **PrijavaTrening**, te u odnosu *Many to One* s entitetom **Korisnik**.

Trening		
idTreninga P	INT	identifikacijski broj treninga
idOrganizatora F	INT	identifikacijski broj organizatora treninga
vrijemePocetka	DATETIME	vrijeme i datum početka treninga
vrijemeZavrsetka	DATETIME	vrijeme i datum završetka treninga
opisTreninga	TEXT	opisan sadržaj treninga

Entitet **Turnir** u odnosu je *One to Many* s entitetom **PrijavaTurnir**.

Turnir		
idTurnira P	INT	identifikacijski broj turnira
formatTurnira	TEXT	opisan format turnira
vrijemePocetka	DATETIME	vrijeme i datum početka turnira
vrijemeZavrsetka	DATETIME	vrijeme i datum završetka turnira
brojSudionika	INT	maksimalan broj sudionika na turniru

Entitet **PrijavaTrening** u odnosu je *Many to One* s entitetom **Trening**, te u odnosu *One to One* s entitetom **Korisnik**.

PrijavaTrening		
idClana PF	INT	identifikacijski broj člana
idTreninga PF	INT	identifikacijski broj treninga

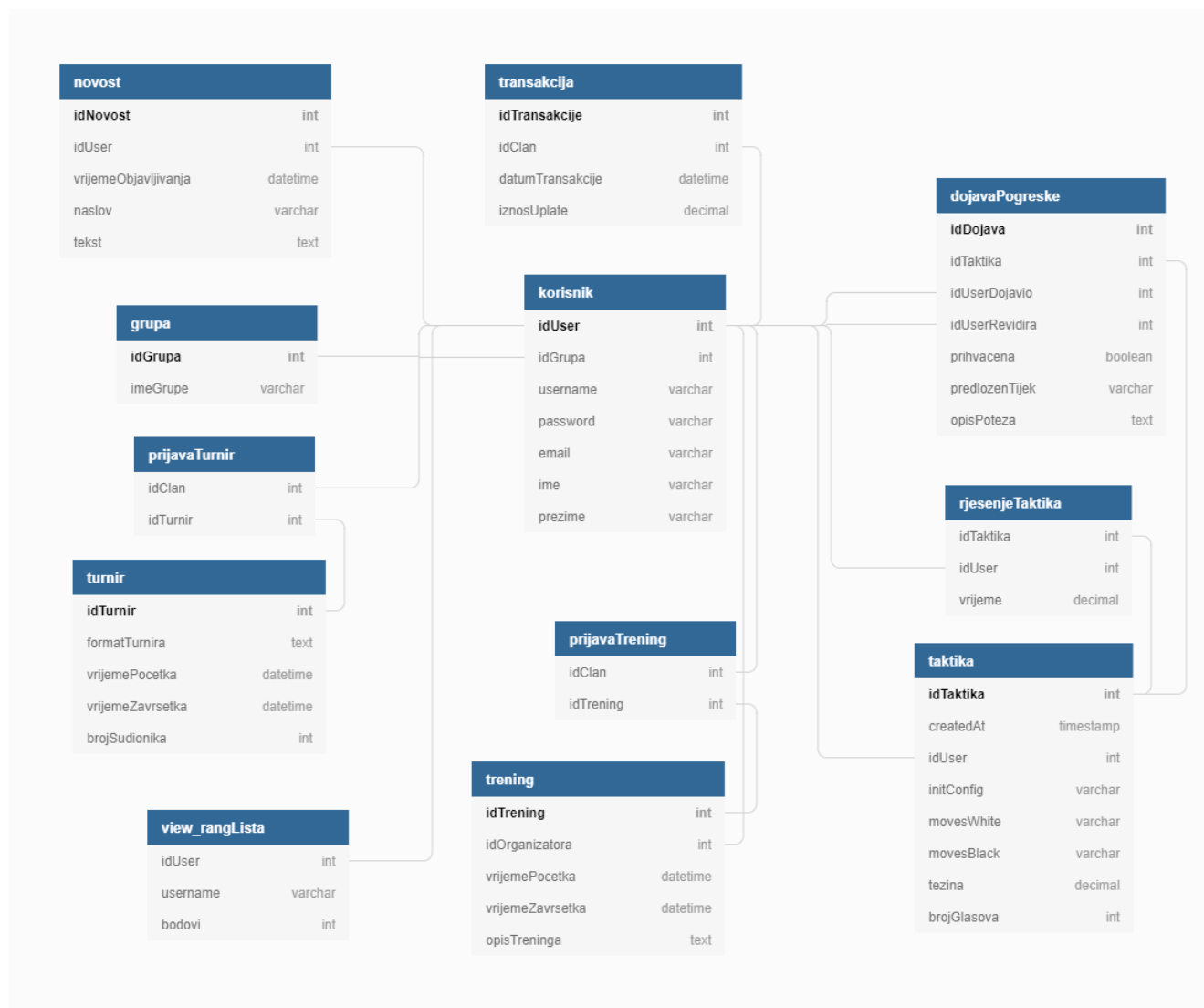
Entitet **PrijavaTurnir** u odnosu je *Many to One* s entitetom **Turnir**, te u odnosu *One to One* s entitetom **Korisnik**.

PrijavaTurnir		
idClana PF	INT	identifikacijski broj člana
idTurnira PF	INT	identifikacijski broj turnira

Entitet **Transakcija** u odnosu je *Many to One* s entitetom **Korisnik**.

Transakcija		
idTransakcije P	INT	identifikacijski broj transakcije
idClana F	INT	identifikacijski broj člana
datumTransakcije	DATETIME	vrijeme i datum izvršavanja transakcije
iznosUplate	DECIMAL	uplaćen iznos

## 4.1.2 Dijagram baze podataka

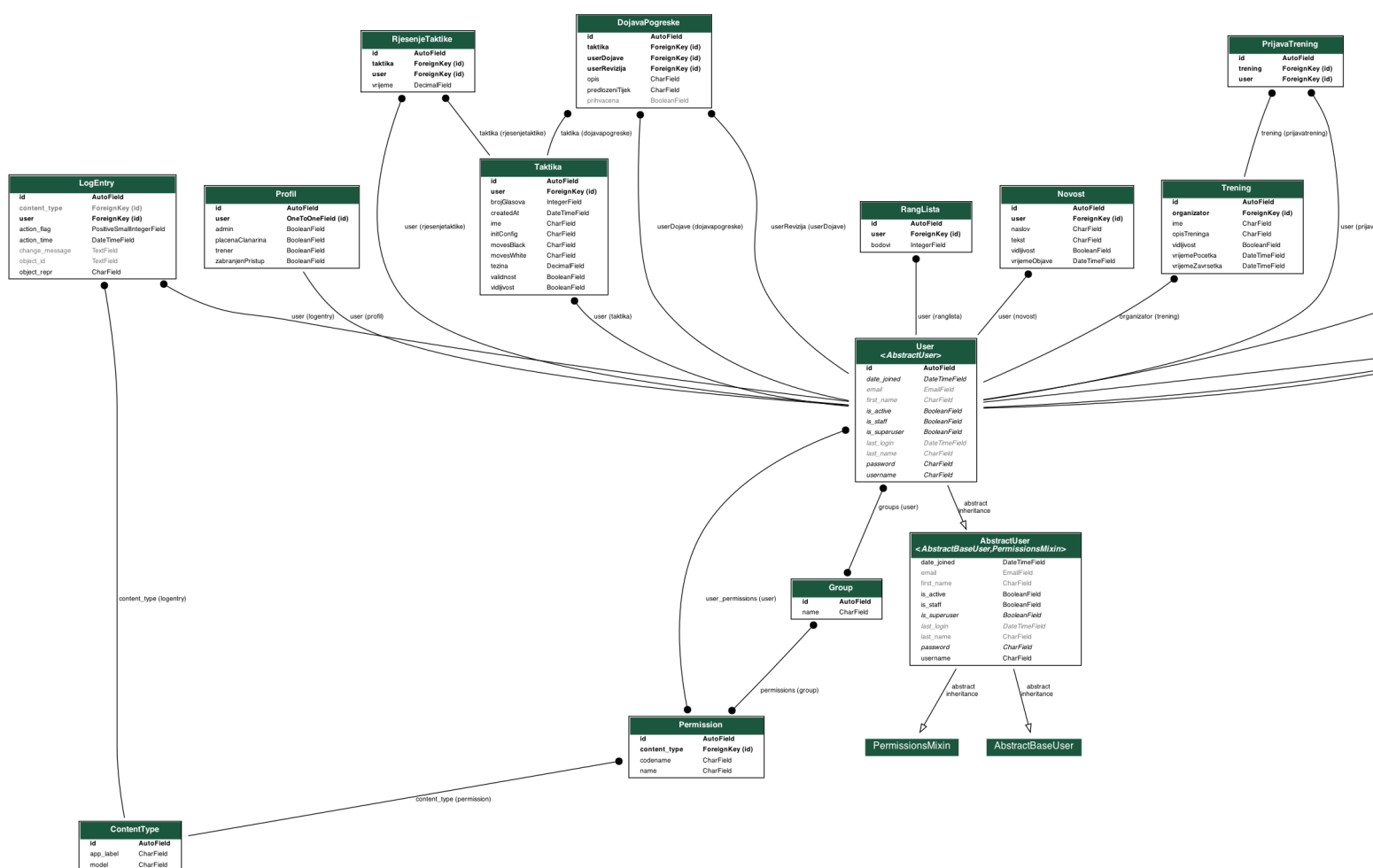


Slika 4.1: E-R dijagram baze podataka

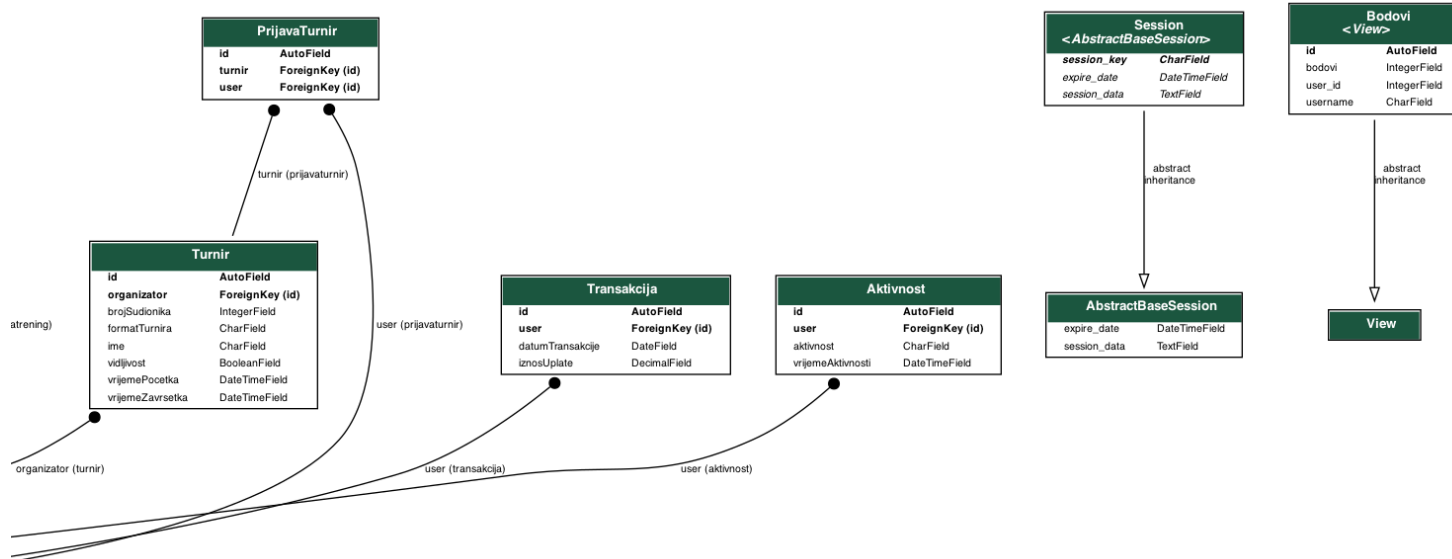
## 4.2 Dijagram razreda

Na sljedećim dijagramima prikazani su razredi koji pripadaju backend dijelu aplikacije. Na slikama 4.2 i 4.3 prikazano je stanje razreda modela koji modeliraju stanje objekata u bazi. Dijagram 4.4 prikazuje sve razrede koje nasljeđuju View. Svaki od ovih razreda zadužen je za obradu upita za određenu putanju. Jedinu podržani tipovi zahtjeva su GET i POST koje implementiraju odgovarajućim metodama. Te metode vraćaju HTTP odgovore na upite najčešće u obliku HTML i JSON datoteka.

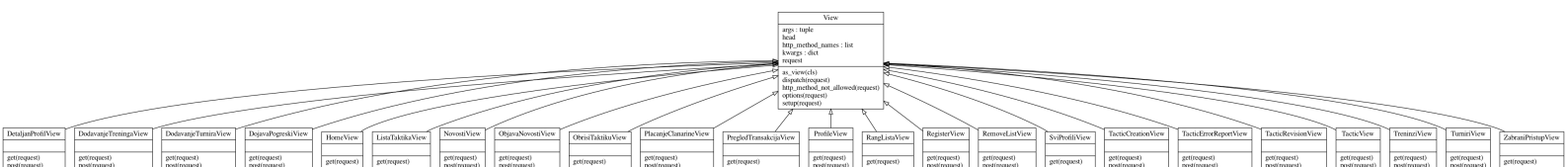
## Dijagrami modela



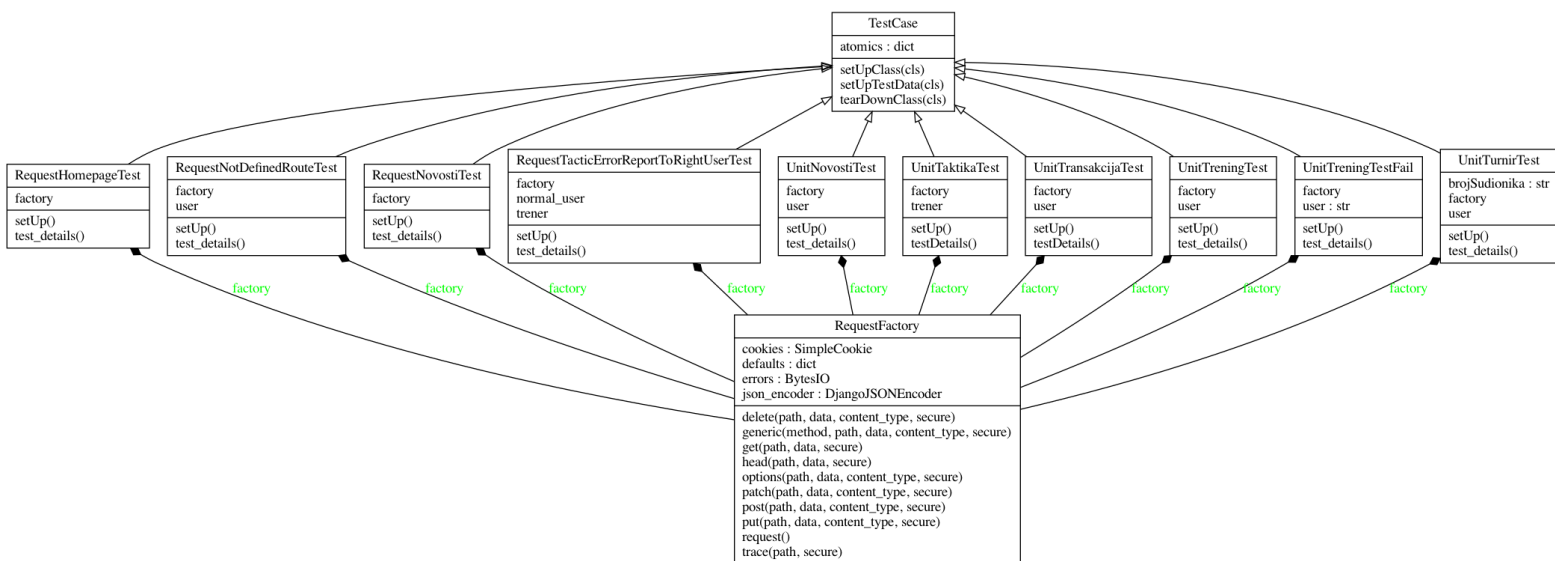
Slika 4.2: Dijagram razreda modela 1. dio



Slika 4.3: Dijagram razreda modela 2. dio



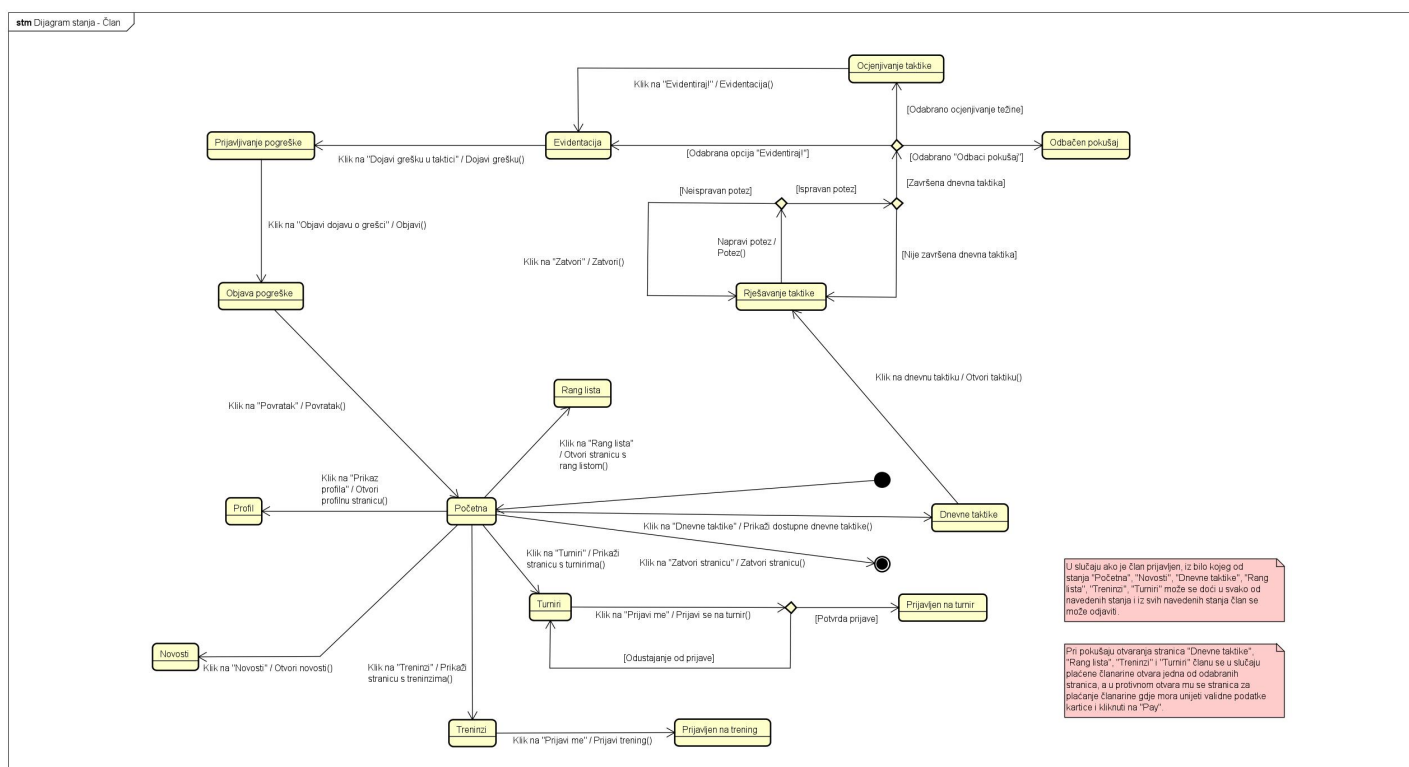
Slika 4.4: Razredni dijagram View-ova



Slika 4.5: Razredni dijagram testnih razreda

### 4.3 Dijagram stanja

Na slici 4.5 prikazan je dijagram stanja za člana. Pri pokretanju aplikacije i prijavljivanju u korisnički račun, članu se otvara početna stranica. Iz nje može otvoriti stranice "Početna", "Novosti", "Dnevne taktike", "Rang lista", "Treninzi", "Turniri" i "Prikaz profila". Pritiskom na "Početna" članu se otvara stranica s osnovnim informacijama o web stranici Šahovski klub MISFLIP. Na stranici "Novosti" mogu se pročitati sve objavljene novosti. Pritiskom na "Prikaz profila" članu se otvara stranica s pregledom vlastitog profila. U slučaju otvaranja stranica "Dnevne taktike", "Rang lista", "Treninzi" i "Turniri" članu se u slučaju neplaćene članarine otvara stranica za plaćanje članarine, a u protivnom odabrana stranica. Uz uvjet da je članarina plaćena, pri otvaranju stranice "Treninzi" član se može prijaviti na trening, a slično je i sa stranicom "Turniri" gdje se može prijaviti na turnir. Stranica "Rang lista" nudi prikaz rang liste, a odlaskom na stranicu "Dnevne taktike" moguće je odabrati jednu od ponuđenih dnevnih taktika, riješiti ju, odabrati težinu za nju i eventualno prijaviti primijećenu pogrešku u postupku rješavanje taktike.

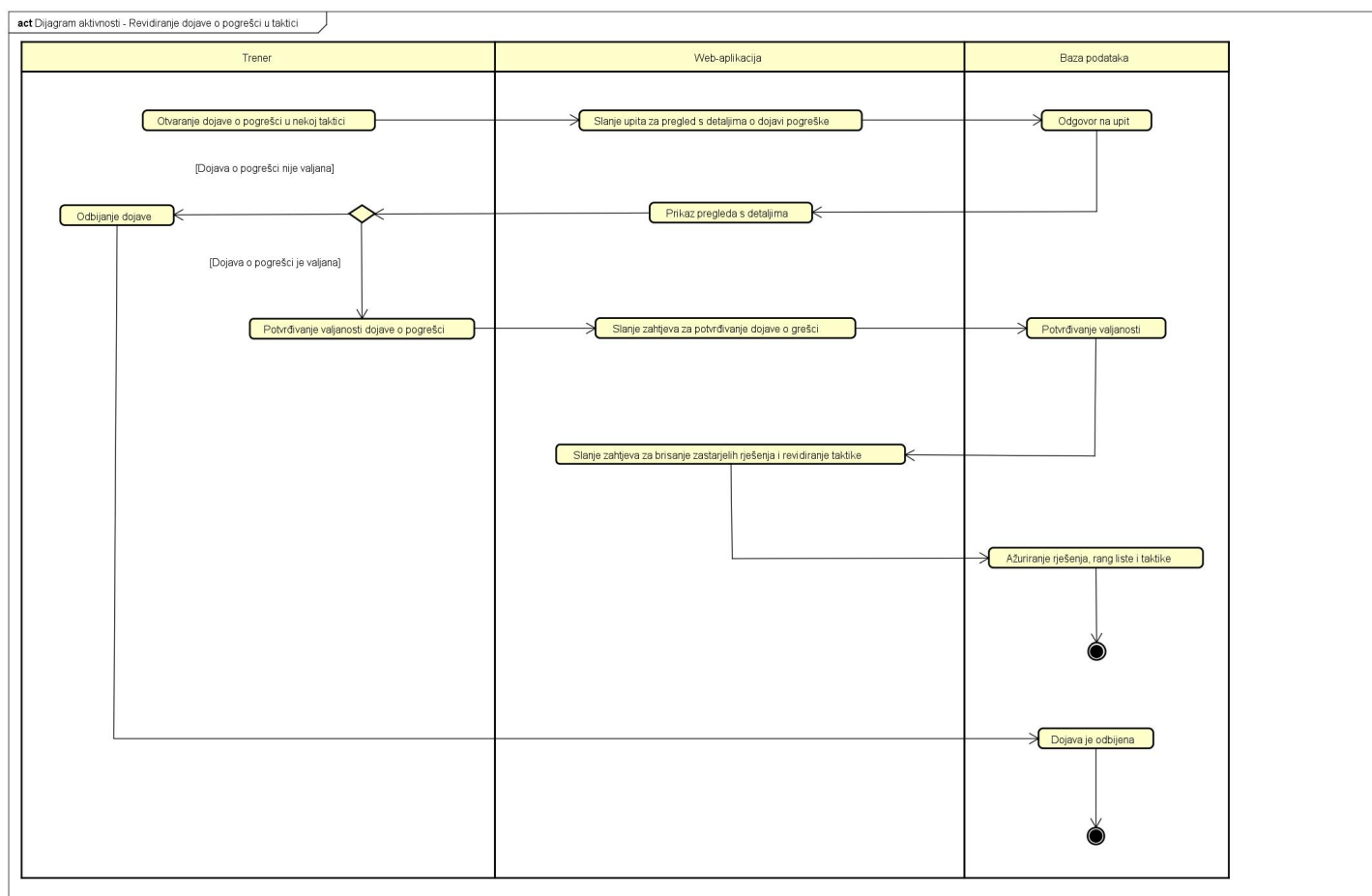


Slika 4.6: Dijagram stanja za člana



## 4.4 Dijagram aktivnosti

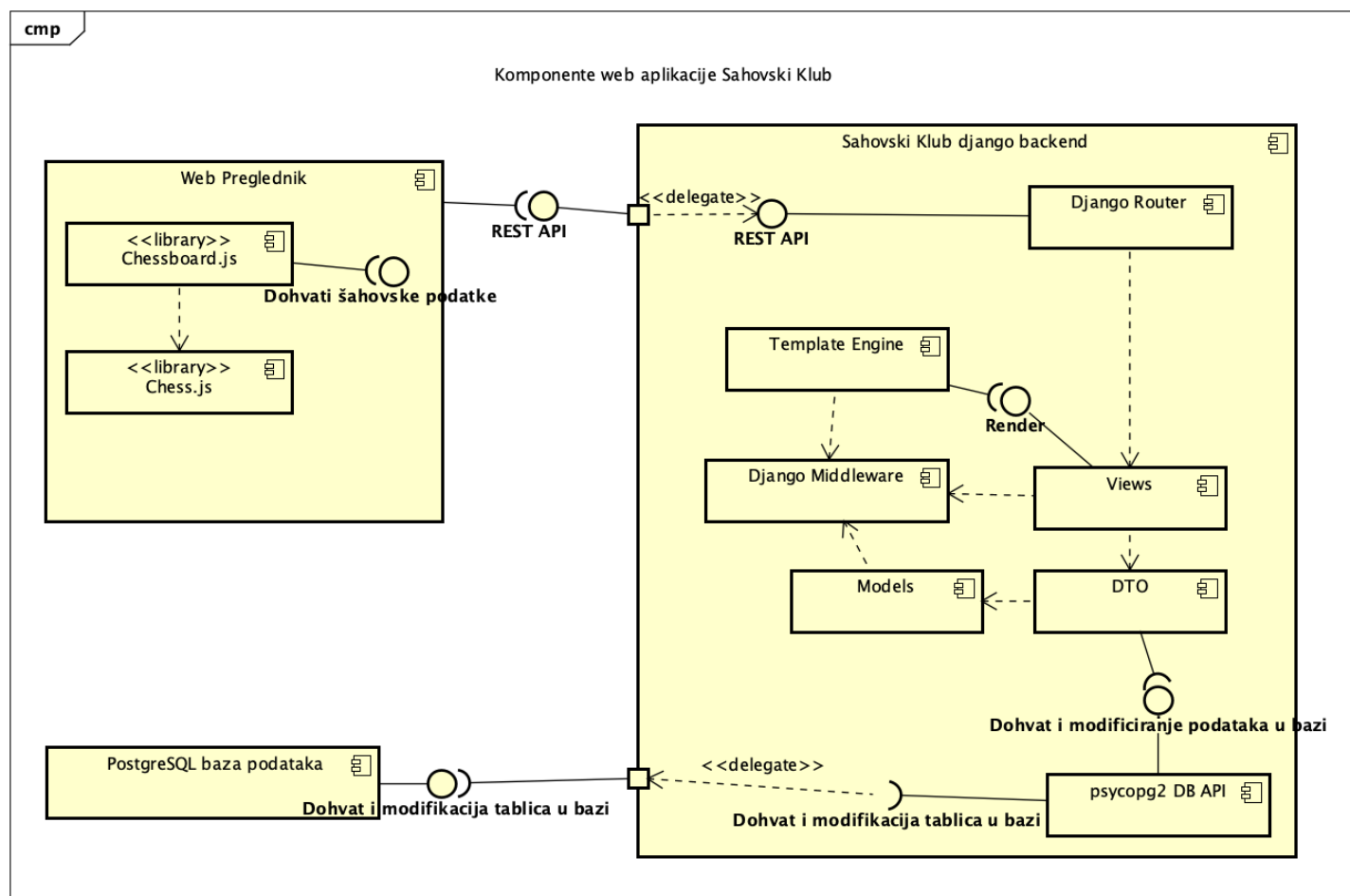
Na slici 4.6 prikazan je proces revidiranja pogreške u taktici. Trener otvara pregled s detaljima o dojadi pogreške u taktici i u ovisnosti o tome je li dojava o pogrešci valjana potvrđuje ili odbija dojavu. U slučaju da prihvati dojavu, šalje se zahtjev za brisanje zastarjelih rješenja revidirane taktike te baza automatski ažurira rang listu kad se promijeni neko od rješenja.



Slika 4.7: Dijagram aktivnosti za revidiranje dojava o pogrešci na taktici

## 4.5 Dijagram komponenti

Dijagram komponenti prikazan dolje ilustrira međuodnose i ovisnosti komponenta aplikacije Sahovski Klub. Korisnici sustavu pristupaju u web pregledniku putem standardnog REST api-ja. HTTP REST zahtjeve djangov interni ruter preusmjerava na odgovarajući view koji onda izvršava potrebnu logiku. Kada za izvršavanje te logike view mora raditi sa podacima iz baze podataka on to obavlja preko DTO (data transfer objects). DTO su python objekti koji opisuju Modele objekata u bazi, a komunikaciju s bazom ostvaruju preko psycopg2 drivera koji Django aplikaciji omogućava komunikaciju sa PostgreSQL bazom podataka. Django middleware moduli pružaju razne funkcionalnosti o kojima su ovisne gotovo sve ostale komponente. Na primjer, Modeli koriste gotova rješenja za autentikaciju Usera kao sigurno spremanje šifri (hash, salt itd.), Template-ovi koriste CSRF tokene za sprečavanje cross site request forgery-ja, te View-ovi koriste session middleware za obavljanje logike o sesijama. Na kraju, view predaje sve potrebne informacije odgovarajućem template-u na renderiranje kojim se generiraju svi potrebni HTML dokumenti koji se zajedno sa CSS i Javascript datotekama šalju kao odgovor na REST upit web preglednika. Također trebaju se istaknuti Chessboard.js i Chess.js Javascript biblioteke koje omogućuju igranje šaha unutar korisnikovog web preglednika. Chessboard.js zadužen je za izgled šahovske ploče a za svu potrebnu šahovsku logiku oslanja se na Chess.js biblioteku. Za početak rada Chessboard.js biblioteci potrebne su određene šahovske informacije poput početnog stanja ploče koje je web preglednik dobio u sklopu odgovora na svoj REST upit.



Slika 4.8: Dijagram Komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Komunikacija u timu je ostvarena putem aplikacija WhatsApp<sup>1</sup> i Microsoft Teams<sup>2</sup>. Aplikacija WhatsApp korištena je isključivo za slikovno-tekstualnu komunikaciju, dok je Microsoft Teams korišten za telekonferencijski kontakt. Kao sustav za upravljanje izvornim kodom korišten je Git<sup>3</sup>, dok se udaljeni repozitorij projekta nalazi na web platformi Git Lab<sup>4</sup>. Za izradu UML dijagrama korišten je Astah UML<sup>5</sup>.

Od alata za razvojno okruženje korišteni su PyCharm<sup>6</sup> i Notepad++<sup>7</sup>. PyCharm je razvojno sučelje koje se koristi za programiranje u Pythonu. Od velike je pomoći kod pisanja koda s laganom navigacijom unutar programa. Notepad++ je dobar za uređenje teksta i izvornog koda za upotrebu u sustavu Microsoft Windows.

Korištena baza podataka je PostgreSQL<sup>8</sup>, a nalazi se na poslužitelju u oblaku Heroku<sup>9</sup>. Za izradu backenda korišteni su Python<sup>10</sup> i Django<sup>11</sup>, a za frontend Bootstrap<sup>12</sup>. Python (Guido van Rossum, 1990.) programski je jezik opće namjene koji dopušta nekoliko stilova programiranja (objektno orijentirano, strukturno i aspektno orijentirano programiranje).

Django je besplatan i open-source web framework koji se temelji na Pythonu. Olakšava stvaranje složenih web stanica.

Bootstrap je frontend framework i besplatna open-source kolekcija HTML CSS i java script alata za kreiranje web stranica i web aplikacija.

Za pisanje dokumentacije korišten je alat TeXstudio.

---

<sup>1</sup><https://www.whatsapp.com/>

<sup>2</sup><https://www.microsoft.com/hr-hr/microsoft-365/microsoft-teams/group-chat-software>

<sup>3</sup><https://git-scm.com/>

<sup>4</sup><https://about.gitlab.com/>

<sup>5</sup><https://astah.net/>

<sup>6</sup><https://www.jetbrains.com/pycharm/>

<sup>7</sup><https://notepad-plus-plus.org/downloads/>

<sup>8</sup><https://www.postgresql.org/>

<sup>9</sup><https://www.heroku.com/>

<sup>10</sup><https://www.python.org/>

<sup>11</sup><https://www.djangoproject.com/>

<sup>12</sup><https://getbootstrap.com/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

U nastavku se nalazi 6 unit testova (izvorni kod i rezultat ispitivanja) numeriranih od 1 do 6.

```
class UnitNovostiTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = User.objects.create_user(
            username='trener', email='trener@fer.hr', password='top_secret')

    def test_details(self):
        vrijeme = datetime.datetime.now(tz=timezone.utc)
        novost = Novost(user_id=self.user.id, vrijemeObjave=vrijeme)
        novost.save()
        self.assertEqual(novost.user.username, self.user.username)
        self.assertEqual(novost.vrijemeObjave, vrijeme)
        self.assertEqual(novost.naslov, '')
        self.assertEqual(novost.tekst, '')
```

Slika 5.1: Izvorni kod unit testa 1

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.UnitNovostiTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 0.138s

OK
Destroying test database for alias 'default'...
```

Slika 5.2: Rezultat ispitivanja unit testa 1

```
class UnitTaktikaTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.trener = User.objects.create_user(username='trener', email='trener@hotmail.com', password='lozinka')
        self.trener.profil.trener = True

    def testDetails(self):
        taktika = Taktika(name="test_taktika", createdAt=datetime.datetime.now(tz=timezone.utc), user=self.trener,
            initConfig="K7/8/8/8/5n2/8/8/1K3Q2 w KQkq - 0 1", movesWhite="Qh1+,Qe4,Qxf4", movesBlack="Ka7,Ka6", tezina=1, brojGlasova=1, validnost=True, vidljivost=True)
        taktika.save()
```

Slika 5.3: Izvorni kod unit testa 2

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.UnitTaktikaTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 0.138s

OK
Destroying test database for alias 'default'...
```

Slika 5.4: Rezultat ispitivanja unit testa 2

```
class UnitTransakcijaTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = User.objects.create_user(
            username='clan', email='clan@fer.hr', password='top_secret')

    def testDetails(self):
        transakcija = Transakcija(user=self.user, datumTransakcije=datetime.datetime.now(tz=timezone.utc), iznosUpdate=100.00)
        transakcija.save()
```

Slika 5.5: Izvorni kod unit testa 3

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.UnitTransakcijaTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 0.139s

OK
Destroying test database for alias 'default'...
```

Slika 5.6: Rezultat ispitivanja unit testa 3

```
class UnitTreningTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = User.objects.create_user(
            username='trener', email='trener@fer.hr', password='top_secret')

    def test_details(self):
        vrijeme = datetime.datetime.now(tz=timezone.utc)
        trening = Trening(organizador=self.user, vrijemePocetka=vrijeme, vrijemeZavrsetka=vrijeme)
        trening.save()
        self.assertEqual(trening.organizador.username, self.user.username)
        self.assertEqual(trening.vrijemePocetka, vrijeme)
        self.assertEqual(trening.vrijemeZavrsetka, vrijeme)
```

Slika 5.7: Izvorni kod unit testa 4

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.UnitTreningTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
-----
Ran 1 test in 0.138s

OK
Destroying test database for alias 'default'...
```

Slika 5.8: Rezultat ispitivanja unit testa 4

```
class UnitTreningTestFail(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = "korisnik123"

    def test_details(self):
        vrijeme = datetime.datetime.now(tz=timezone.utc)
        trening = Trening(organizador=self.user)
        trening.save()
```

Slika 5.9: Izvorni kod unit testa 5

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.UnitTreningTestFail
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
E
=====
ERROR: test_details (SahovskiKlub.tests.UnitTreningTestFail)
-----
Traceback (most recent call last):
  File "C:\MISFLIPdevelop\misflip\IzvorniKod\SahovskiKlub\tests.py", line 49, in test_details
    trening = Trening(organizador=self.user)
  File "C:\Python38\lib\site-packages\django\db\models\base.py", line 483, in __init__
    _setattr(self, field.name, rel_obj)
  File "C:\Python38\lib\site-packages\django\db\models\fields\related_descriptors.py", line 215, in __set__
    raise ValueError(
ValueError: Cannot assign "'korisnik123'": "Trening.organizador" must be a "User" instance.

-----
Ran 1 test in 0.020s

FAILED (errors=1)
Destroying test database for alias 'default'...
```

Slika 5.10: Rezultat ispitivanja unit testa 5

```
class UnitTurnirTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = User.objects.create_user(
            username='trener', email='trener@fer.hr', password='top_secret')
        self.brojSudionika = 'deset'

    def test_details(self):
        turnir = Turnir(organizador=self.user, brojSudionika=self.brojSudionika)
        turnir.save()
```

Slika 5.11: Izvorni kod unit testa 6

```
C:\MISFLIPdevelop\misflip\Izvornikod>python manage.py test SahovskiKlub.tests.UnitTurnirTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
E
=====
ERROR: test_details (SahovskiKlub.tests.UnitTurnirTest)
-----
Traceback (most recent call last):
  File "C:\Python38\lib\site-packages\django\db\models\fields\__init__.py", line 1774, in get_prep_value
    return int(value)
ValueError: invalid literal for int() with base 10: 'deset'

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "C:\MISFLIPdevelop\misflip\Izvornikod\SahovskiKlub\tests.py", line 94, in test_details
    turnir.save()
  File "C:\Python38\lib\site-packages\django\db\models\base.py", line 753, in save
    self.save_base(using=using, force_insert=force_insert,
  File "C:\Python38\lib\site-packages\django\db\models\base.py", line 790, in save_base
    updated = self._save_table(
  File "C:\Python38\lib\site-packages\django\db\models\base.py", line 895, in _save_table
    results = self._do_insert(cls._base_manager, using, fields, returning_fields, raw)
  File "C:\Python38\lib\site-packages\django\db\models\base.py", line 933, in _do_insert
    return manager._insert(
  File "C:\Python38\lib\site-packages\django\db\models\manager.py", line 85, in manager_method
    return getattr(self.get_queryset(), name)(*args, **kwargs)
  File "C:\Python38\lib\site-packages\django\db\models\query.py", line 1254, in _insert
    return query.get_compiler(using=using).execute_sql(returning_fields)
  File "C:\Python38\lib\site-packages\django\db\models\sql\compiler.py", line 1396, in execute_sql
    for sql, params in self.as_sql():
  File "C:\Python38\lib\site-packages\django\db\models\sql\compiler.py", line 1339, in as_sql
    value_rows = [
  File "C:\Python38\lib\site-packages\django\db\models\sql\compiler.py", line 1340, in <listcomp>
    [self.prepare_value(field, self.pre_save_val(field, obj)) for field in fields]
```

Slika 5.12: Rezultat ispitivanja unit testa 6 - 1. dio

```
File "C:\Python38\lib\site-packages\django\db\models\sql\compiler.py", line 1340, in <listcomp>
    [self.prepare_value(field, self.pre_save_val(field, obj)) for field in fields]
File "C:\Python38\lib\site-packages\django\db\models\sql\compiler.py", line 1281, in prepare_value
    value = field.get_db_prep_save(value, connection=self.connection)
File "C:\Python38\lib\site-packages\django\db\models\fields\__init__.py", line 823, in get_db_prep_save
    return self.get_db_prep_value(value, connection=connection, prepared=False)
File "C:\Python38\lib\site-packages\django\db\models\fields\__init__.py", line 818, in get_db_prep_value
    value = self.get_prep_value(value)
File "C:\Python38\lib\site-packages\django\db\models\fields\__init__.py", line 1776, in get_prep_value
    raise e.__class__(
ValueError: Field 'brojSudionika' expected a number but got 'deset'.

-----
Ran 1 test in 0.146s

FAILED (errors=1)
Destroying test database for alias 'default'...
```

Slika 5.13: Rezultat ispitivanja unit testa 6 - 2. dio

## 5.2.2 Ispitivanje sustava

U nastavku se nalaze 4 slučajeve ispitivanja sustava (izvorni kod i rezultat ispitivanja) numeriranih od 1 do 4.



```
class RequestHomepageTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()

    def test_details(self):
        print("Upisite korisnicko ime")
        username = input()
        print("Upisite lozinku")
        lozinka = input()
        request = self.factory.get('/')
        try:
            request.user = User.objects.create_user(
                username=username, email=username+'@fer.hr', password=lozinka)
        except ValueError:
            pass
        response = HomeView.as_view()(request)
        self.assertEqual(response.status_code, 200)
```

Slika 5.14: Izvorni kod ispitivanja sustava 1

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.RequestHomepageTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Upisite korisnicko ime
Ime
Upisite lozinku
Lozinka
.
-----
Ran 1 test in 9.222s

OK
Destroying test database for alias 'default'...
```

Slika 5.15: Rezultat ispitivanja sustava 1

```
class RequestNotDefinedRouteTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = User.objects.create_user(
            username='clan', email='clan@fer.hr', password='password')

    def test_details(self):
        c = Client()
        response = c.get('/nedefiniranaRuta')
        self.assertEqual(response.status_code, 404)
```

Slika 5.16: Izvorni kod ispitivanja sustava 2

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.RequestHomepageTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Upisite korisnicko ime
Ime
Upisite lozinku
Lozinka
.
-----
Ran 1 test in 9.222s

OK
Destroying test database for alias 'default'...
```

Slika 5.17: Rezultat ispitivanja sustava 2

```
class RequestNovostiTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()

    def test_details(self):
        print("Upisite korisnicko ime")
        username = input()
        print("Upisite lozinku")
        lozinka = input()
        request = self.factory.get('/novosti')
        try:
            request.user = User.objects.create_user(
                username=username, email=username + '@fer.hr', password=lozinka)
        except ValueError:
            pass
        response = NovostiView.as_view()(request)
        self.assertEqual(response.status_code, 200)
```

Slika 5.18: Izvorni kod ispitivanja sustava 3

```
C:\MISFLIPdevelop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.RequestNovostiTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Upisite korisnicko ime
ime
Upisite lozinku
lozinka
.
-----
Ran 1 test in 5.862s

OK
Destroying test database for alias 'default'...
```

Slika 5.19: Rezultat ispitivanja sustava 3

```
class RequestTacticErrorReportToRightUserTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.normal_user = User.objects.create_user(
            username='John', email='john@hotmail.com', password='top_secret')
        self.trainer = User.objects.create_user(
            username='trainer', email='trainer@hotmail.com', password='top_secret')
        self.trainer.profil.trainer = True

    def test_details(self):
        vrijeme = datetime.datetime.now(tz=timezone.utc)
        taktika = Taktika(ime='laka_taktika', createdAt=vrijeme, user=self.trainer,
            initConfig="k7/8/8/8/5n2/8/8/1K3Q2 = KQkq - 0 1", movesWhite="Qh1+,Qe4,Qxf4", movesBlack="Ka7,Ka6", tezina=1, brojGlasova=1, validnost=True, vidljivost=True)
        taktika.save()
        dojava_pogreske = DojavaPogreske(taktika=taktika, userDojava=self.normal_user, userRevizija=self.trainer, predlozeniTijek="Qf3+,Qxf4<W|B>Ka7", opis="vrlo informativan opis")
        dojava_pogreske.save()
        request = self.factory.get(f'/reviziranjeTaktike?id={dojava_pogreske.id}')
        request.user = self.trainer
        response = TacticRevisionView.as_view()(request)
        self.assertEqual(response.status_code, 200)
```

Slika 5.20: Izvorni kod ispitivanja sustava 4

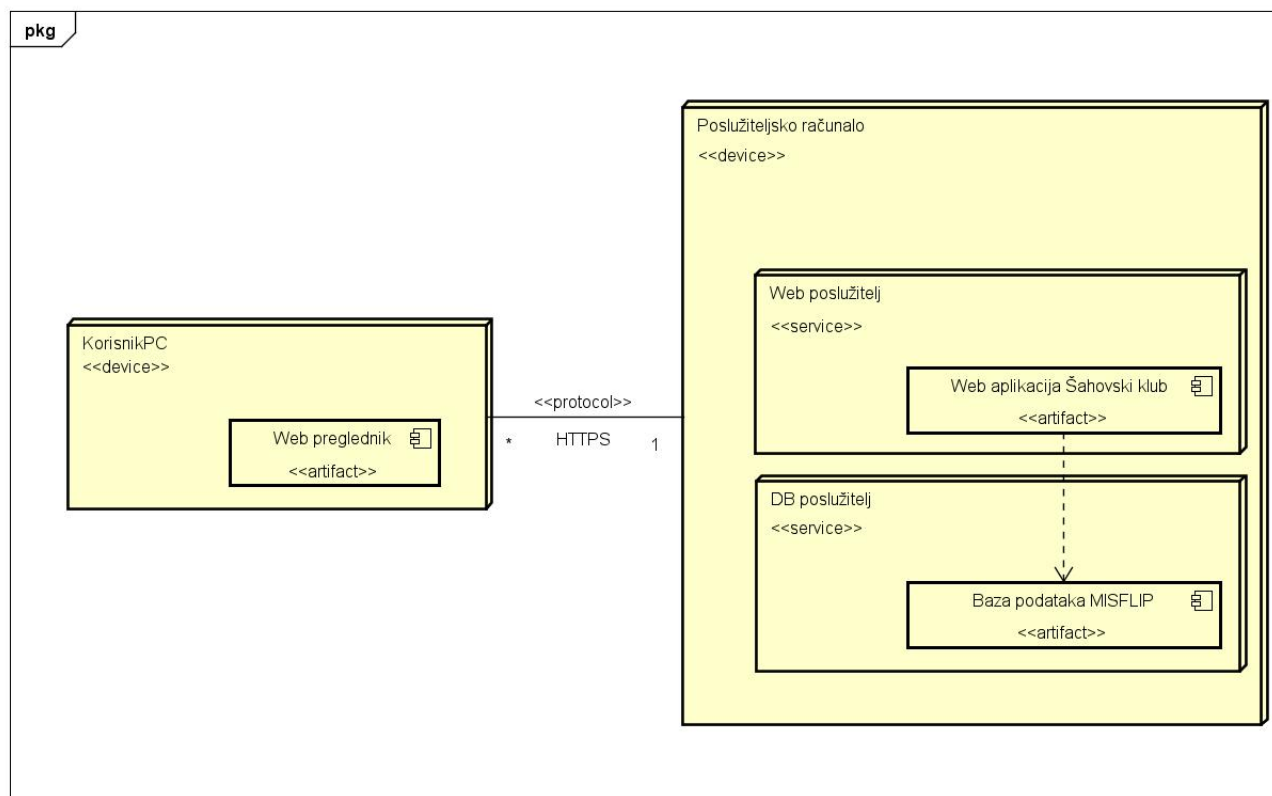
```
C:\MISFLIP\develop\misflip\IzvorniKod>python manage.py test SahovskiKlub.tests.RequestNovostiTest
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Upisite korisnicko ime
ime
Upisite lozinku
lozinka
:
-----
Ran 1 test in 5.862s

OK
Destroying test database for alias 'default'...
```

Slika 5.21: Rezultat ispitivanja sustava 4

## 5.3 Dijagram razmještaja

Na slici 5.22 prikazan je dijagram razmještaja. Na poslužiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka MISFLIP. Za pristup web aplikaciji Šahovski klub, klijent se služi web preglednikom. Sustav je baziran na arhitekturi klijent-poslužitelj. Komunikacija između korisnikovog računala (neregistrirani korisnik, član, trener, administrator) i poslužitelja odvija se preko HTTPS veze.



Slika 5.22: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

Za početak potrebno je otići na stranicu <https://www.heroku.com/> i registrirati se.

U sljedećem koraku treba otvoriti Command Prompt i pozicionirati se u mapu IzvorniKod unutar projekta.

Treba upisati naredbu "heroku login" i potom pritisnuti bilo koji gumb izuzevši q pri čemu se otvara stranica na web pregledniku gdje je potrebno prijaviti se s podacima koji su kreirani putem registracije.

Potom se u Command Promptu upisuje naredba „heroku create“ pri čijem izvršavanju se dobije ime servera.

```
E:\Programiranje\heroku\novo>heroku login
heroku: Press any key to open up the browser to login or q to exit:
Opening browser to https://cli-auth.heroku.com/auth/cli/browser/8b263df1-be79-4f1a-883b-34f1fc91766?requestor=SPHYNTY_g2gD0QAAAAxNDEuMTM4LjM3LjQwbGYPAP3y5nYBYq4BUYA_pXAszICRcyR-WKYBELUdgIdrjc4pu7LbmF04yP_SRZE
Logging in... done
Logged in as serene-stream-43293

E:\Programiranje\heroku\novo>heroku create
Creating app... done, ● serene-stream-43293
https://serene-stream-43293.herokuapp.com/ | https://git.heroku.com/serene-stream-43293.git
```

Slika 5.23: Podešavanje u Command Promptu 1

Sljedeće treba upisat "git init", a nakon toga "heroku git:remote -a", i uz ovu naredbu se navodi ime heroku servera koji je dobiven u prethodnom koraku npr. „serene-stream-43293“.

Nakon toga u datoteku settings.py u ALLOWED\_HOSTS treba staviti ime heroku servera i završiti sa ".herokuapp.com" npr. "serene-stream-43293.herokuapp.com". Nakon što je to napravljeno u Command Promptu upisuje se naredba "git add .".

```
E:\Programiranje\heroku\novo>git init
Initialized empty Git repository in E:/Programiranje/heroku/novo/.git/

E:\Programiranje\heroku\novo>heroku git:remote -a serene-stream-43293
set git remote heroku to https://git.heroku.com/serene-stream-43293.git

E:\Programiranje\heroku\novo>git add .
warning: LF will be replaced by CRLF in .idea/inspectionProfiles/profiles_settings.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in static/css/chessboard-1.0.0.css.
The file will have its original line endings in your working directory
```

Slika 5.24: Podešavanje u Command Promptu 2

Zatim se upisuje "git commit -m "message"" (pri čemu message može biti bilo koji niz znakova npr. "initial commit").

```
E:\Programiranje\heroku\novo>git commit -m "initial commit"
[master (root-commit) a4efacd] initial commit
114 files changed, 7560 insertions(+)
create mode 100644 .idea/.gitignore
create mode 100644 .idea/inspectionProfiles/profiles_settings.xml
create mode 100644 .idea/misc.xml
create mode 100644 .idea/modules.xml
create mode 100644 .idea/novo.iml
create mode 100644 MISFLIP/__init__.py
create mode 100644 MISFLIP/__pycache__/__init__.cpython-38.pyc
create mode 100644 MISFLIP/__pycache__/settings.cpython-38.pyc
create mode 100644 MISFLIP/__pycache__/urls.cpython-38.pyc
create mode 100644 MISFLIP/__pycache__/wsgi.cpython-38.pyc
create mode 100644 MISFLIP/asgi.py
create mode 100644 MISFLIP/settings.py
create mode 100644 MISFLIP/urls.py
create mode 100644 MISFLIP/wsgi.py
create mode 100644 SahovskiKlub/__init__.py
```

Slika 5.25: Podešavanje u Command Promptu 3

Na kraju se upisuje "git push heroku master", te je time aplikacija namještena.

```
E:\Programiranje\heroku\novo>git push heroku master
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 6 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 357 bytes | 357.00 KiB/s, done.
Total 4 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
remote: -----> No change in requirements detected, installing from cache
remote: -----> Installing pip 20.1.1, setuptools 47.1.1 and wheel 0.34.2
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: -----> $ python manage.py collectstatic --noinput
remote:      164 static files copied to '/tmp/build_8ccda8c9/staticfiles'.
remote:
remote: -----> Discovering process types
remote:      Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:      Done: 53.4M
remote: -----> Launching...
remote:      Released v11
remote:      https://serene-stream-43293.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/serene-stream-43293.git
 0fbb1d7..278fcab master -> master
```

Slika 5.26: Podešavanje u Command Promptu 4

## 6. Zaključak i budući rad

Zadatak grupe MISFLIP bio je izrada web aplikacije koja omogućuje poslovanje šahovskom klubu s naglašenom mogućnošću rješavanja dnevnih taktika kako neregistriranih korisnika tako i članova, te prijava članova na treninge i turnire. Grupa od sedam članova zadatak je izvršila uspješno kroz tri faze u zadanom vremenskom roku od 17 tjedana.

Prva faza uključivala je formiranje tima i uvodni sastanak putem MS Teamsa s asistentom gdje smo dobili uvodne upute i savjete vezane uz izvršavanje projekta. Druga faza odnosila se na period gdje smo podijelili zaduženja prema afinitetima a sve probleme i tijek izrade dodatno komentirali na tjednim sastancima članova preko MS Teamsa. Dio komunikacije odvijao se i preko WhatsAppa. Glavnina ove faze bila je vezana uz izradu dokumentacije, a pri samom kraju uz navedeno počelo se raditi na programskom rješenju.

U trećoj fazi intenzitet rada i komunikacije znatno je povećan. Paralelno se radilo na dokumentaciji i programiranju. U ovoj fazi grupa je imala određenih poteškoća zbog dodatnih napora učenja i savladavanja i gradiva predmeta i programske tehnologije.

Grupa je uložila velik trud i napor, ali i ostavila prostor za proširenje i unapređenje. Sama provedba koliko god bila zahtjevna, ujedinila je grupu u jednu cjelinu i pokazala kako kolektiv savladava zadani im projekt. Naučili smo puno, ali i imali značajnu podršku asistenta.

# Popis literature

## *Kontinuirano osvježavanje*

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

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. PostgreSQL Documentation, <https://www.postgresql.org/docs/>
8. W3Schools Online Web Tutorials, <https://www.w3schools.com/>
9. Django documentation, <https://docs.djangoproject.com/en/3.1/>
10. chessboard.js documentation, <https://chessboardjs.com/docs>
11. chess.js documentation, <https://github.com/jhlywa/chess.js/blob/master/README.md>
12. Bootstrap 4.0 Introduction, <https://getbootstrap.com/docs/4.0/getting-started/introduction/>



# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 5. listopada 2020.
- Prisustvovali: Igor Stančin, svi članovi tima
- Teme sastanka:
  - Uvodni sastanak s asistentom

### 2. sastanak

- Datum: 7. listopada 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - upoznavanje
  - dogovor o tehnologijama

### 3. sastanak

- Datum: 12. listopada 2020.
- Prisustvovali: Igor Stančin, Ivan Klabučar, Petar Gabrijel Kedmenec
- Teme sastanka:
  - pitanja oko funkcionalnih zahtjeva

### 4. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - raspodjela poslova oko obrazaca uporabe

### 5. sastanak

- Datum: 21. listopada 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - raspodjela poslova oko UML dijagrama
  - rasprava o prirodi uloge Admin

## 6. sastanak

- Datum: 24. listopada 2020.
- Prisustvovali: Ivan Klabučar, Fran Leontić, Luka Žižić
- Teme sastanka:
  - raspravljanje o implementaciji dnevnih taktika

## 7. sastanak

- Datum: 24. listopada 2020.
- Prisustvovali: Ivan Klabučar, Petar Gabrijel Kedmenec
- Teme sastanka:
  - rješavanje konflikata na GitLabu

## 8. sastanak

- Datum: 26. listopada 2020.
- Prisustvovali: Igor Stančin, svi članovi tima
- Teme sastanka:
  - razjašnjavanje dijagrama obrazaca i sekvencijskih dijagrama

## 9. sastanak

- Datum: 29. listopada 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - raspodjela kodiranja
- Zaključci:
  - Ivar Cmrečak, Fran Leontić i Ivan Klabučar rade na bazi
  - Petar Gabrijel Kedmenec i Luka Žižić rade na uspostavljanju templatea za listanje dnevnih taktika
  - Mihael Svetec i Sven Paprskar rade na uspostavljanju templatea za listanje novosti

## 10. sastanak

- Datum: 31. listopada 2020.
- Prisustvovali: Ivan Klabučar, Fran Leontić, Luka Žižić, Ivar Cmrečak
- Teme sastanka:
  - raspravljanje o tablicama u bazi podataka

## 11. sastanak

- Datum: 4. studenoga 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - raspravljanje o implementaciji potrebnih funkcija za demonstraciju generičke funkcionalnosti

## 12. sastanak

- Datum: 9. studenoga 2020.
- Prisustvovali: Igor Stančin, svi članovi tima
- Teme sastanka:
  - pokazivanje generičke funkcionalnosti aplikacije
  - pregled dokumentacije

## 13. sastanak

- Datum: 11. studenoga 2020.
- Prisustvovali: Ivan Klabučar, Ivar Cmrečak, Petar Gabrijel Kedmenec, Fran Leontić, Mihael Svetec, Sven Paprskar
- Teme sastanka:
  - raspodjela uloga prije pauze za međuispite
- Zaključci:
  - Petar Gabrijel Kedmenec dovršava uređenje dokumentacije za prvu predaju
  - Sven Paprskar izrađuje template za prikaz transakcija
  - Luka Žižić izrađuje template za dodavanje novih obavijesti
  - Mihael Svetec izrađuje template za dodavanje, listanje i prijavljivanje na treninge
  - Fran Leontić izrađuje template za prikaz korisničkog profila
  - Ivan Klabučar izrađuje template za objavu treninga
  - Ivar Cmrečak uspostavlja bazu podataka

## 14. sastanak

- Datum: 18. studenoga 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - pregled napravljenog i komentiranje o potrebnim ispravcima programskog koda

## 15. sastanak

- Datum: 25. studenoga 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - daljnja podjela rada vezana uz kodiranje i dokumentaciju

## 16. sastanak

- Datum: 2. prosinca 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - razrješavanje problema vezanog uz programiranje

## 17. sastanak

- Datum: 9. prosinca 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - analiza dosadašnjeg rada, dogovor o nastavku rada

## 18. sastanak

- Datum: 14. prosinca 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - dokumentacija, raspravljanje parova koji rade na zajedničkom problemu

## 19. sastanak

- Datum: 18. prosinca 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - dogovor o nastavku rada na dokumentaciji i programskim rješenjima

## 20. sastanak

- Datum: 23. prosinca 2020.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - podjela posla preko zimskih praznika

## 21. sastanak

- Datum: 4. siječnja 2021.
- Prisustvovali: svi članovi tima

- Teme sastanka:
  - dogovori oko završavanja programiranja vezanog uz alfa verziju aplikacije

## 22. sastanak

- Datum: 4. siječnja 2021.
- Prisustvovali: Igor Stančin, svi članovi tima
- Teme sastanka:
  - prezentacija alfa inačice aplikacije asistentu Igoru Stančinu

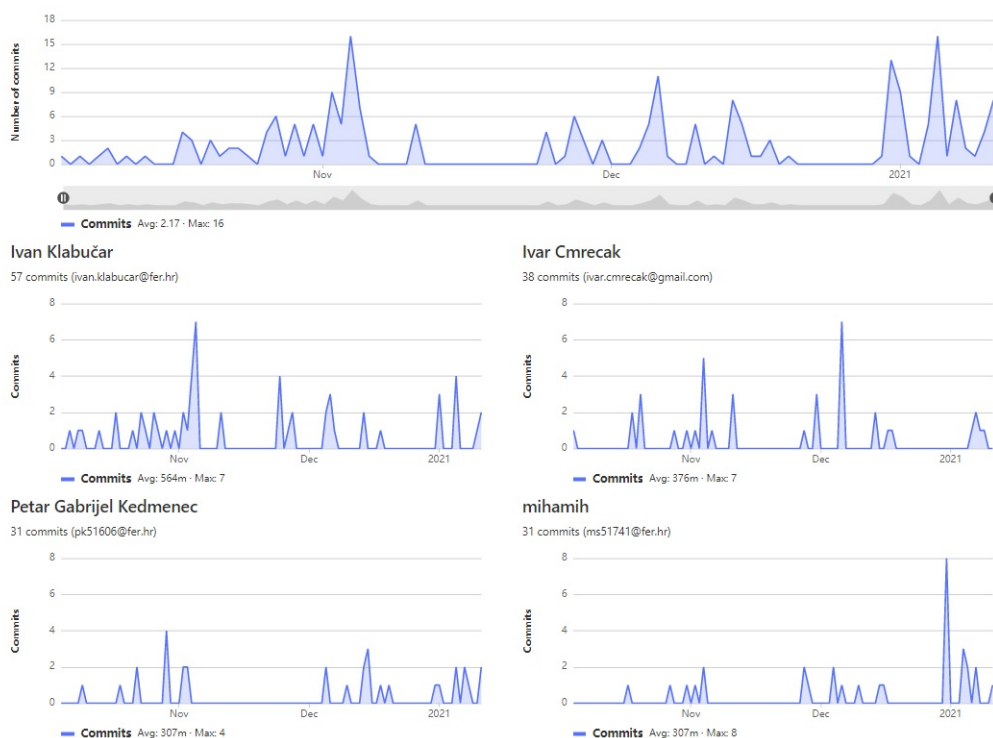
## Tablica aktivnosti

### Kontinuirano osvježavanje

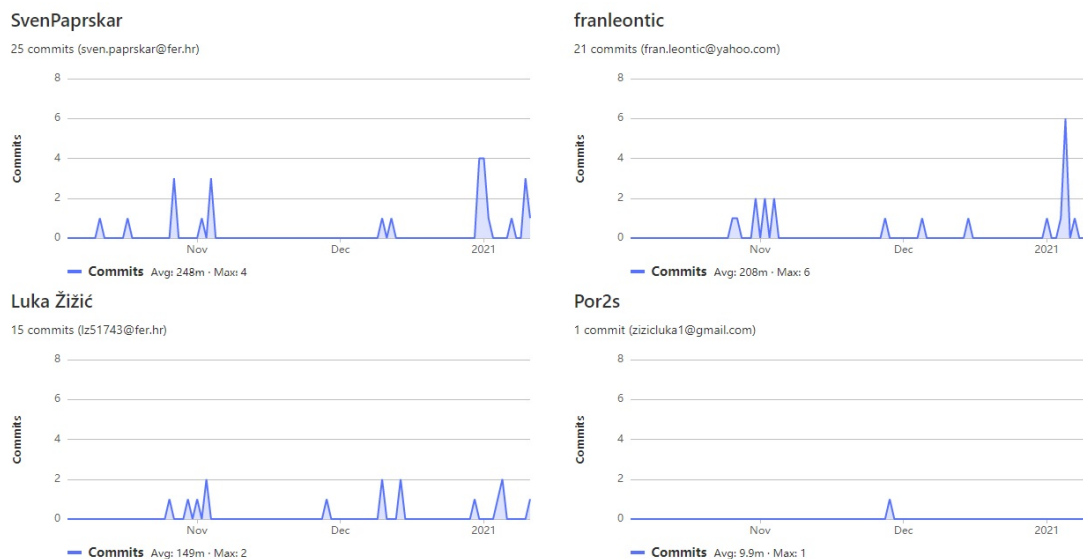
	Ivar Cmrečak	Sven Paprskar	Ivan Klabučar	Mihael Svetec	Fran Leontić	Petar Gabrijel Kedmenec	Luka Žižić
Upravljanje projektom	10		12				
Opis projektnog zadatka		1	1		2.5		2
Funkcionalni zahtjevi		1				3	
Opis pojedinih obrazaca	1	2	1	3		3	
Dijagram obrazaca		1		2			
Sekvencijski dijagrami			3		1.5		2.5
Opis ostalih zahtjeva						1.5	
Arhitektura i dizajn sustava	1						
Baza podataka	1		2	1	2	1	1
Dijagram razreda			5		1		
Dijagram stanja		5					
Dijagram aktivnosti		2					
Dijagram komponenti							
Korištene tehnologije i alati	2	2					
Ispitivanje programskog rješenja		2	1	1			
Dijagram razmještaja		1.5					
Upute za puštanje u pogon		1.5					
Dnevnik sastajanja		1	.5			1	
Zaključak i budući rad		1					
Popis literature						.5	
Izrada baze podataka	3		3				

	Ivar Cmrečak	Sven Paprskar	Ivan Klabučar	Mihael Svetec	Fran Leontić	Petar Gabrijel Kedmenec	Luka Žižić
Spajanje s bazom podataka	3						
Izrada frontenda	3		6		2		
Izrada autorizacije	3						
Izrada stranice s novostima		2		2			5
Izrada stranice s dnevnim taktikama			6			2	3
Izrada stranice s treninzima				5			
Izrada stranice s turnirima				5			
Izrada početne stranice i nav-bara				2			0.5
Izrada stranice s dojavama pogreški				3			
Izrada stranice profila					5		
Izrada liste profila i detaljnog prikaza					4		
Izrada stranice za plaćanje članarine							3
Izrada adminove liste neplaćenih članarina					2	1	
Izrada adminove liste svih transakcija						4	
Izrada rang liste			3			1	
Deployment na remote server	4						

## Dijagrami pregleda promjena



Slika 6.1: Prikaz aktivnosti na repozitoriju 1. dio



Slika 6.2: Prikaz aktivnosti na repozitoriju 2. dio