

# Programsko inženjerstvo

Ak. god. 2022./2023.

## *Sinappsa*

Dokumentacija, Rev. 2

Grupa: *TurbulentTech*

Voditelj: *Mislav Domlija*

Datum predaje: 13. *siječnja* 2023.

Nastavnik: *Laura Majer*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>6</b>
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	10
3.1.2 Sekvencijski dijagrami . . . . .	18
3.2 Ostali zahtjevi . . . . .	21
<b>4 Arhitektura i dizajn sustava</b>	<b>22</b>
4.1 Baza podataka . . . . .	23
4.1.1 Opis tablica . . . . .	23
4.1.2 Dijagram baze podataka . . . . .	28
4.2 Dijagram razreda . . . . .	29
4.3 Dijagram stanja . . . . .	30
4.4 Dijagram aktivnosti . . . . .	31
4.5 Dijagram komponenti . . . . .	32
<b>5 Implementacija i korisničko sučelje</b>	<b>33</b>
5.1 Korištene tehnologije i alati . . . . .	33
5.2 Ispitivanje programskog rješenja . . . . .	34
5.2.1 Ispitivanje komponenti . . . . .	34
5.2.2 Ispitivanje sustava . . . . .	39
5.3 Dijagram razmještaja . . . . .	41
5.4 Upute za puštanje u pogon . . . . .	42
<b>6 Zaključak i budući rad</b>	<b>43</b>
<b>Popis literature</b>	<b>44</b>
<b>Indeks slika i dijagrama</b>	<b>46</b>

**Dodatak: Prikaz aktivnosti grupe**

**47**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Mislav Đomlija	3.11.2022.
0.2	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	Maksim Kos	12.11.2022.
0.3	Arhitektura i dizajn sustava, algoritmi i strukture podataka	Mislav Đomlija, Maksim Kos	25.10.2022.
0.4	Opisi obrazaca uporabe	Maksim Kos	07.11.2022.
0.5	Preveden uvod	Maksim Kos	12.11.2022.
0.6	Sekvencijski dijagrami	Maksim Kos	07.11.2022.
0.7.1	Započeo dijagrame razreda	David Šupančić	10.11.2022.
0.7.2	Nastavak dijagrama razreda	David Šupančić	12.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Mislav Đomlija, Maksim Kos, David Supančić	17.11.2022.
1.1	Popravljeni dijagrami obrazaca uporabe	Maksim Kos	19.12.2022.
1.2	Popravljeni sekvencijski dijagrami	Maksim Kos	19.12.2022.
1.3	Dijagram stanja	David Supančić	22.12.2022.
1.4	Dijagram aktivnosti	David Supančić	30.12.2022.
1.5	Dijagram razmještaja	David Supančić	5.1.2023.
1.6	Dijagram komponenti	David Supančić	6.1.2023.
1.7	Dijagram razreda završen	David Supančić	8.1.2022.
1.8	Opisane korištene tehnologije i alati	David Supančić	8.1.2022.
1.9	Upute za puštanje u pogon	Mislav Đomlija	10.1.2022.
1.10	Ispitivanje programskog rješenja	Maksim Kos	10.1.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.11	Zaključak rada	David Supančić	11.1.2022.
2.0	Konačni tekst predložka dokumentacije	Mislav Đomlija, Maksim Kos, David Supančić	13.1.2023.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku potporu za stvaranje Web-aplikacije "Sinappsa" koja će omogućiti studentima Fakulteta Elektrotehnike i Računastva da ponude pomoć ili zatraže pomoć oko specifičnog gradiva, labaratorijske vježbe, ispita ili kolegija općenito. Nakon nekoliko godina iskustva jasno je vidljivo da ferovke i ferovci imaju želju pomoći svojim kolegama, ali i neke kolege imaju potrebu zatražiti pomoć kako bi savladali gradivo koje ne razumiju što nas je potaknulno na izradu ove aplikacije. Aplikacija održava rang listu najuspješnijih studenata-pomagača što bi moglo pomoći korisnicima koji traže pomoć u odabiru što boljeg studenta-pomagača.

Prilikom pokretanja aplikacije, prikazuje se lista najnovijih objavljenih oglasa te rang lista najbolje ocjenjenih studenata-pomagača.

Aplikaciji mogu pristupiti tri vrste korisnika:

- neregistrirani
- registrirani
- moderator

Neregistriranom korisniku dostupna je lista trenutno objavljenih oglasa kao i rejting-lista studenata-pomagača. Neregistrirani korisnik može oglase filtrirati po smjeru(racunastvo ili elektrotehnika), kolegiju(ViS, UPRO, ARH2, PPJ, UTR) i kategoriji(labaratorijska vježba, blic, zadaća, ispitni rok). Neregistriranom korisniku omogućeno je prijavljivanje u sustav s postojećim računom(potrebno je upisati korisničko ime i lozinku) ili kreirati novi račun(kojim će postati registrirani korisnik). Prilikom kreiranja novog računa potreban je unos sljedećih podataka:

- ime
- prezime
- korisničko ime
- lozinka
- email adresa
- avatar(opcionalno/inače default)

Korisnik mora potvrditi registraciju odlaskom na poveznicu koja će mu biti poslana na adresu e-pošte koju je unio tokom registracije. Korisnik mora paziti na domenu vlastite e-mail adrese prilikom registracije (domena e-mail adrese mora biti @fer.hr). Registracijom u sustav korisniku se dodjeljuju sva prava koja ima registrirani korisnik.

Registrirani korisnik uz mogućnosti koje ima i neregistrirani korisnik može objavljivati i odgovarati na oglase. Također registrirani korisnik ima mogućnost i ocijeniti studenta-pomagača koji mu je održao instrukcije (na temelju statusa upita koji mora biti *prihvaćen*). Prikazi aktivnih i neaktivnih oglasa, trenutnih upita i informacija o profilu korisnika nalaze se grupirani u poseban izbornik „*Moj profil*“, gdje je korisniku omogućeno naknadno promijeniti lozinku, korisničko ime ili avatar te biti ukratko informiran o vlastitoj aktivnosti u samoj aplikaciji.

Moderator može uz sva prava registriranog korisnika uklanjati nepravilne ili neprikladne oglase ako smatra da je to potrebno. Prilikom uklanjanja oglasa, kreatora se obaviještava putem mail-a o razlogu brisanja njegovog oglasa.

Oglasi mogu biti statusa *aktivan* ili *neaktivan*. Prilikom stvaranja oglasa navodi se naslov, opis, kolegij te kategorija oglasa. Kategoriju definira program (npr. Računarstvo), predmet (npr. MATAN2) i komponenta kolegija (npr. MI). Zainteresirani korisnici postavljaju upit na oglas te se putem maila-a obaviještava kreatora oglasa da je postavljen upit na njegov oglas. Daljna komunikacija između kreatora oglasa i kreatora upita odvija se *izvan aplikacije*.

Ocjene se daju studentima koji su održali instrukcije. Ocjene su u rasponu od jedan do pet te se na temelju njih definira rang lista studenata-pomagača. Osoba koja je slušala instrukcije ocijenjuje studenta-pomagača kojem se ocjena pribraja u dosadašnje. Cilj ocjenjivanja studenata-pomagača je da drugi korisnici kojima je potrebna pomoć u vezi nekog kolegija imaju uvid o tome koliko je neki korisnik dobar instruktor.

Rang lista se sastoji od studenata pomagača poredanih po ukupnom zbroju ocjena. Kako se pomagači s puno ocjena 1 nebi podizali na ljestvici od svake ocjene se prije zbrajanja oduzima 2. Tako bi pomagač s ocjenama 5,5,3,2 imao ukupan zbroj 7, a pomagač s ocjenama 5,5,1,2 imao ukupan zbroj 5.



## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Studenti
  - (a) Neregistrirani
  - (b) Registrirani
2. Moderator
3. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) Pregledati listu trenutno objavljenih oglasa
  - (b) Pregledati rejting listu studenata-pomagača
  - (c) Filtirati oglase po smjeru, kolegiju i kategoriji
  - (d) Se registrirati u sustav, stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, lozinka, ime, prezime, e-mail adresa i opcionalno profilna slika
2. Registrirani korisnik (inicijator) može:
  - (a) Objavljivati i odgovarati na oglase
  - (b) Pregledavati svoje aktivne i neaktivne oglase
  - (c) Uređivati ili obrisati vlastite oglase
  - (d) Pregledavati sve svoje trenutne upite
  - (e) Ocjeniti studetna-pomagača
  - (f) Urediti profil
  - (g) Odjaviti se iz aplikacije
  - (h) Izbrisati korisnički račun
  - (i) Obrisati svoj oglas

3. Moderator (inicijator) može:

- (a) Izbrisati korisnički račun
- (b) Ukloniti tuđe oglase
- (c) Dodati kolegije, koje označava s obzirom na smjer

4. Baza podataka (sudionik) može:

- (a) Pohraniti podatke o korisnicima i njihovim ovlastima
- (b) Pohraniti podatke o oglasima i upitima
- (c) Pohraniti obrisane oglase (neprikladne ili obrisane od strane korisnika)

### 3.1.1 Obrasci uporabe

#### UC1 -Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvaranje korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Neregistrirani korisnik odabire opciju za registraciju
  2. Neregistrirani korisnik unosi potrebne korisničke podatke
  3. Šalje se poruka na e-mail adresu za potvrdu registracije
  4. Korisnik klikom na link u poruci potvrđuje registraciju
  5. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog korisničkog imena ili neispravne e-mail adrese, unos korisničkog podatka u nedozvoljenom formatu
    1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

#### UC2 -Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijava korisnika u sustav s vlastitim računom
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu
  2. Unos korisničkog imena i lozinke
  3. Potvrda o ispravnosti unesenih podataka
  4. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
  - 2.a Neispravno korisničko ime/lozinka
    1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu u aplikaciju

#### UC3 -Odjava registriranog korisnika iz sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjava korisnika iz sustav
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za odjavu
  2. Korisnik se odjavljuje i preusmjerava ga se na početnu stranicu

#### UC4 -Pregled oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled aktivnih oglasa
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Lista svih aktivnih oglasa je prikazana pilikom učitavanja aplikacije

#### UC5 -Pretraga oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Filtrirati oglase po vlastitim kriterijima
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik filtrira oglase po kriteriju kolegija, smjera i kategorije
  2. Izlistaju se oglasi koji zadovoljavaju tražene kriterije
- **Opis mogućih odstupanja:**
  - 2.a Niti jedan oglas ne zadovoljava tražene kriterije
    1. Korisnika se obaviještava da nema traženog oglasa

#### UC6 -Pregled rang liste

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati rang listu najbolje ocjenjenih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Rang lista je prikazana pilikom učitavanja aplikacije

#### UC7 -Uređivanje vlastitih oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Preudrediti objavljeni oglas
- **Sudionici:** Baza podataka
- **Preduvjet:**
  1. Korisnik je prijavljen u sustav
  2. Oglas je aktivan
- **Opis osnovnog tijeka:**
  1. Korisnik odabire izbornik "Moj profil"
  2. Korisnik odabire opcije preuredi
  3. Preuređuje naslov/opis oglasa
  4. Odabire opciju pohrani
  5. Promjene se pohranjuju u bazu podataka

#### UC8 -Odgovaranje na oglas

- **Glavni sudionik:** Korisnik
- **Cilj:** Postaviti upit
- **Sudionici:** Baza podataka, kreator oglasa
- **Preduvjet:**
  1. Korisnik je prijavljen u sustav
  2. Oglas je aktivan
- **Opis osnovnog tijeka:**
  1. Korisniku su prikazani oglasi
  2. Korisnik odabire opciju "Odgovori"
  3. Korisnik unosi proizvoljnu poruku
  4. Korisniku koji je objavio oglas se šalje e-mail s upitom i kontakt podacima zainteresiranog korisnika
  5. Upit se dodaje u izbornik "Moj profil" i kreatoru upita i kreatoru oglasa
  6. Upit se pohranjuje u bazu podataka

#### UC9 -Objava oglas

- **Glavni sudionik:** Korisnik
- **Cilj:** Objaviti oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Stvori"
2. Korisnik unosi traženi naslov, opis, kolegij i kategoriju
3. Korisnik odabire opciju "Objavi oglas"
4. Oglas se pohranjuje u bazu podataka

#### UC10 -Ocjenjivanje studenta-pomagača

- **Glavni sudionik:** Korisnik
- **Cilj:** Ocjeniti studenta-pomagača
- **Sudionici:** Baza podataka
- **Preduvjet:**
  1. Korisnik je prijavljen u sustav
  2. Oglas je aktivan
  3. Reply je statusa *prihvaćen*
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju ocjeni
  2. Korisnik unosi ocjenu izvrsnosti studenta-pomagača
  3. Ocjena se pribraja u dosadašnje ocjene studenta-pomagača
  4. Ocjena se pohranjuje u bazu podataka
- **Opis mogućih odstupanja:**
  - 2.a Oglas je izbrisan
    1. Korisnika se obavještava da nema mogućnost ocjene studenta-pomagača

#### UC11 -Pregled osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Osobni podatci"
  2. Aplikacija prikazuje osobne podatke korisnika

#### UC12 -Promjena osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabere opciju za promjenu podataka
  2. Korisnik mijenja svoje osobne podatke
  3. Korisnik sprema promjene
  4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 2.a Korisnik promijeni svoje osobne podatke, ali ne odabere opciju "Spremi promjenu"
    1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora
  - 2.b Korisnik unosi već zauzeto korisničko ime
    1. Sustav obavještava korisnika da je unio već zauzeto korisničko ime

### UC13 -Brisanje korisničkog računa

- **Glavni sudionik:** Korisnik
- **Cilj:** izbrisati svoj korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik pregledava osobne podatke
  2. Otvara se stranica s osobnim podacima korisnika
  3. Korisnik briše račun
  4. Korisnički račun se izbriše iz baze podataka
  5. Otvara se početna stranica

### UC14 -Brisanje neprikladnih oglasa

- **Glavni sudionik:** Moderator
- **Cilj:** Ukloniti neprikladne oglase
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik ima ulogu moderatora
- **Opis osnovnog tijeka:**
  1. Moderator primjećuje da je određenih aktivni oglas neprikladan
  2. Odabire opciju obriši te daje objašnjenje kreatoru oglasa
  3. Kreator oglasa je obaviješten putem e-mail adrese o uklanjanju oglasa
- **Opis mogućih odstupanja:**

2.a Oglas je već uklonjen ili obrisao od strane kreatora, ali je idalje vidljiv u aplikaciji

1. Obavijest moderatoru da je dani oglas uklonjen

#### UC15 -Dodavanje kolegija

- **Glavni sudionik:** Moderator
- **Cilj:** Dodavanje novog kolegija
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik ima ulogu moderatora
- **Opis osnovnog tijeka:**
  1. Moderator odabire opciju "Dodaj kolegij"
  2. Navodi ime kolegija i smjer kojem kolegij pripada
  3. Odabirom opciju "Dodaj" kolegij se dodaje
  4. Kolegij se pohranjuje u bazu podataka
- **Opis mogućih odstupanja:**
  - 3.a Kolegij je već dodan u sustavu
    1. Obavijest moderatoru da je kolegij već u sustavu

#### UC16 -Pregled vlastitih oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz svih korisnikovih oglasa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moj profil"
  2. Otvara se stranica s osobnim podacima korisnika
  3. Korisniku se prikazuju svi njegovi aktivni i neaktivni oglasi

#### UC17 -Brisanje vlastitih oglasa

- **Glavni sudionik:** Korisnik
- **Cilj:** Korisnikovih želi obrisati svoje aktivne oglase
- **Sudionici:** Baza podataka
- **Preduvjet:**
  1. Korisnik je prijavljen
  2. Oglas je statusa aktivan
- **Opis osnovnog tijeka:**



1. Korisnik odabire opciju "Moj profil"
  2. Otvara se stranica s osobnim podacima korisnika
  3. Korisniku se prikazuju svi njegovi aktivni i neaktivni oglasi
  4. Korisnik odabire opciju brisanja oglasa
  5. Oglas se briše iz baze podataka
- **Opis mogućih odstupanja:**
    - 4.a Oglas je već obrisao od strane moderatora
      1. Obavijest korisniku da je oglas obrisao

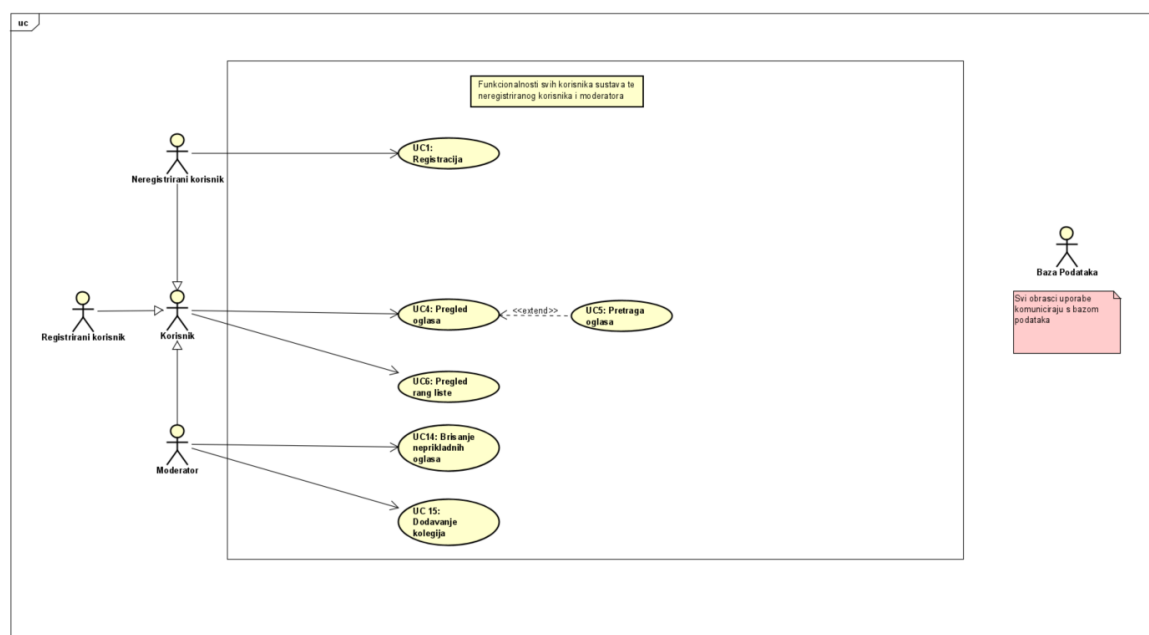
#### UC18 -Pregled trenutnih korisnikovih upita

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz svih korisnikovih upita
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moj profil"
  2. Otvara se stranica s osobnim podacima korisnika
  3. Korisnik odabire opciju za prikaz svojih upita
  4. Korisniku se prikazuju svi njegovi upiti na oglase

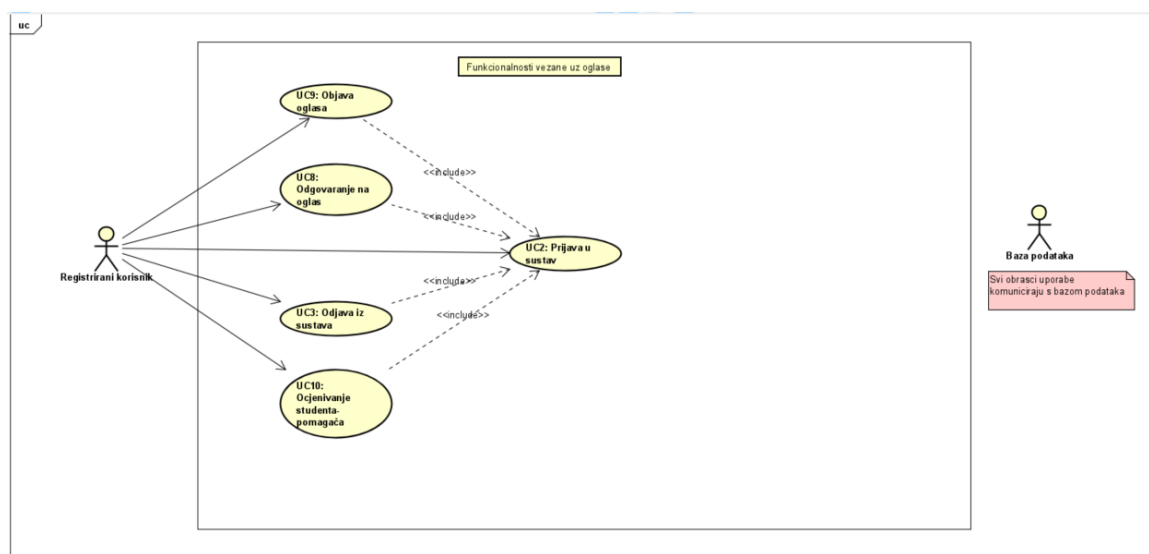
#### UC19 -Pregled upita korisniku

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz upita na korisnikov oglas
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Moj profil"
  2. Otvara se stranica s osobnim podacima korisnika
  3. Korisnik odabire opciju za prikaz upita na svoje
  4. Korisniku se prikazuju svi njegovi upiti na njegove oglase

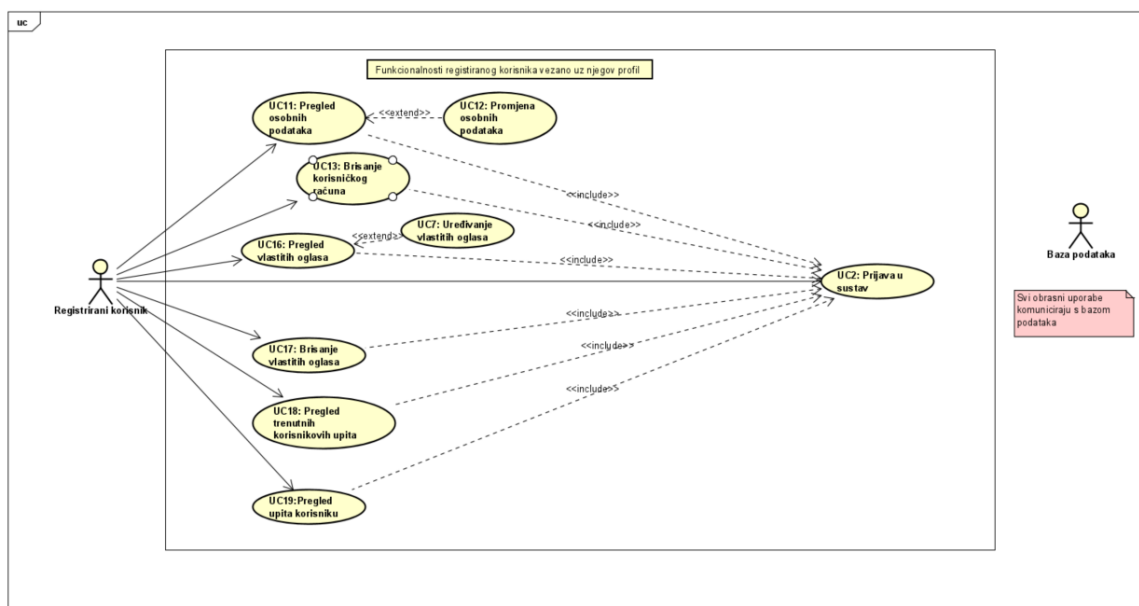
## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe funkcionalnosti svih korisnika, neregistriranog korisnika, moderatora i baze podataka



Slika 3.2: Dijagram obrasca uporabe registriranog korisnika vezano uz oglase



Slika 3.3: Dijagram obrasca uporabe registriranog korisnika vezane uz profil

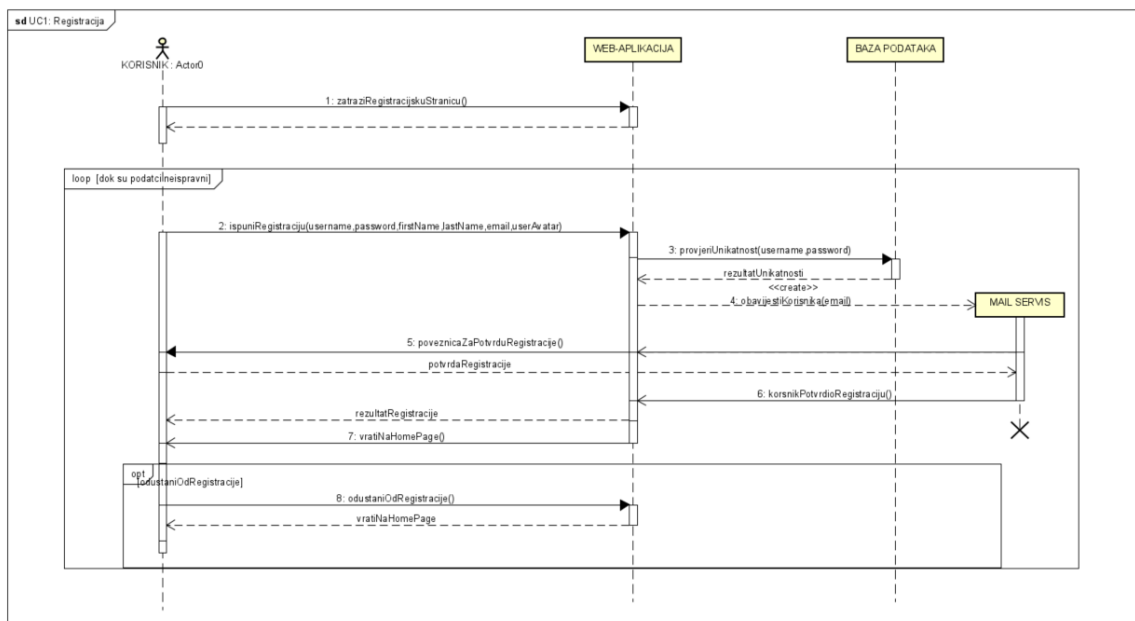
### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1-Registracija

Neregistrirani korisnik zatraži stranicu za registraciju kako bi se registrirao i imao mogućnosti koje ima registrirani korisnik. Web-aplikacija mu prikazuje stranicu za registraciju. Korisnik unosi sve potrebne podatke za registraciju koje Web-aplikacija zaprima i provjerava unikatnost unesenih podataka s bazom podataka točnije unikatnost e-mail adrese i username-a. Nakon rezultata unikatnosti Web-aplikacija uspostavlja kontakt s Mail-servisom koji korisniku na njegovu e-mail adresu šalje link za registraciju. Korisnik potvrđuje registraciju klikom na poveznicu, te se rezultat registracije dojavljuje Web-aplikaciji te se korisnika vraća na početnu stranicu. Ako je registracija neuspjela, vraća se korisnika na početnu formu za registraciju. Korisnik tokom registracijskog procesa može odustati od registracije te ga Web-aplikacija vraća na početnu stranicu.

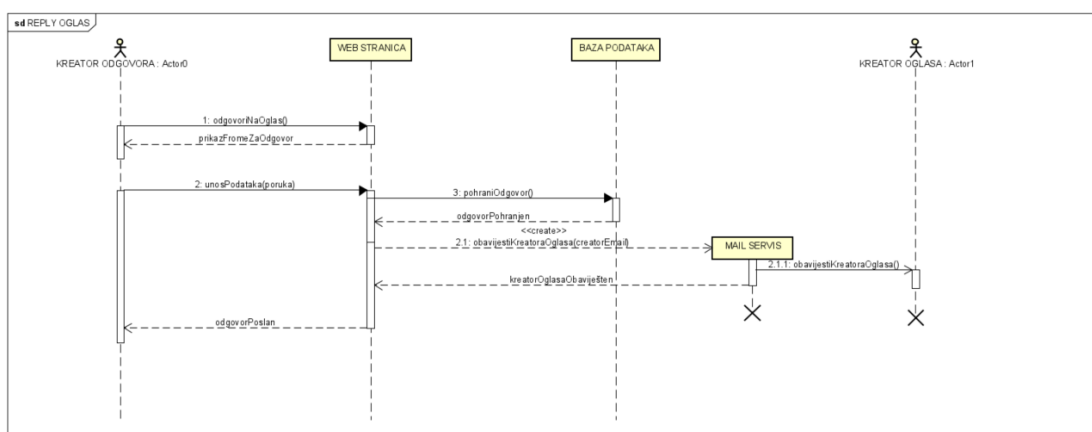
#### Obrazac uporabe UC6-Odgovor na oglas

Korisnik koji želi postaviti upit na traženi oglas odabire opciju reply te mu Web-aplikacija dostavlja formu za reply. Korisnik unosi proizvoljni tekst koji se predaje Web-aplikaciji, a Web-aplikacija prosljeđuje bazi podataka kako bi baza pohranila



Slika 3.4: Sekvencijski dijagram za UC1

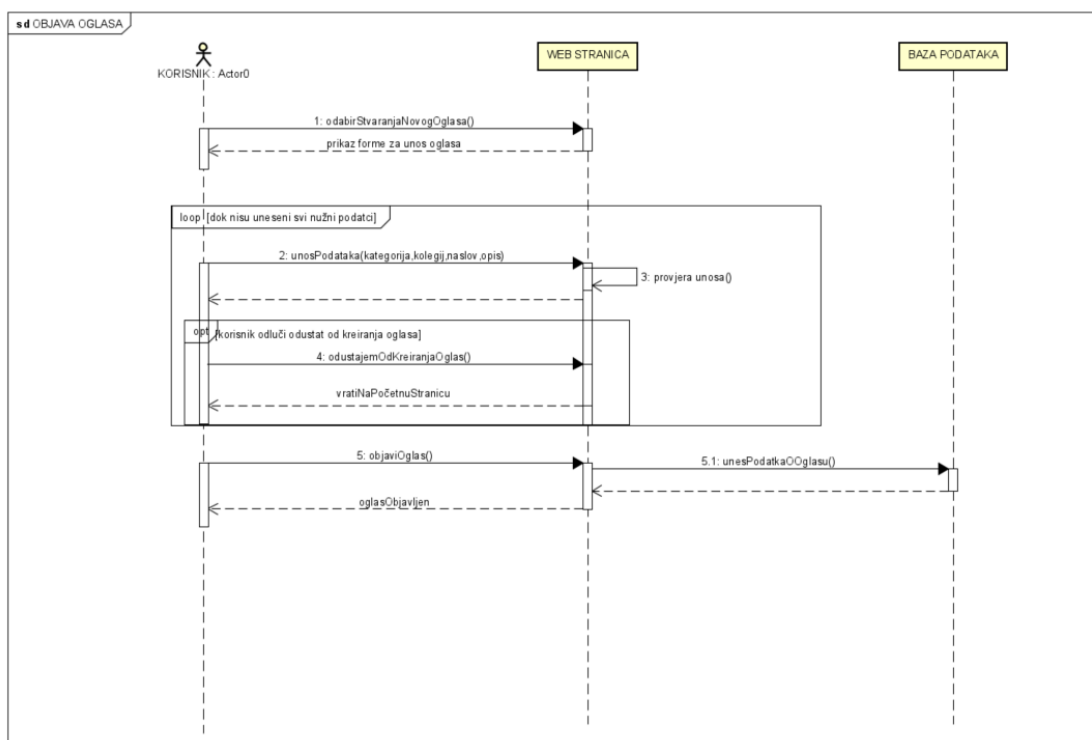
zadani upit. Web-aplikacija nakon pohrane upita u bazu podatak uspostavlja vezu s Mail-servisom kako bi obavijestila kreatora oglasa putem mail-a da mu je neki korisnik odgovorio na oglas. Web-aplikacija na kraju kreatoru upita daje doznanja da je mail poslan kreatoru oglasa.



Slika 3.5: Sekvencijski dijagram za UC6

### Obrazac uporabe UC7-Objava oglasa

Korisnik želi objaviti oglas. Odabire opciju stvaranje novog oglasa te mu Web-aplikacija vraća formu za stvaranje oglasa. Korisnik unosi podatke za oglas te unos predaje Web-aplikaciji koja provjerava valjanost unosa. Ako je unos nevaljan Web-aplikacija vraća korisnika ponovno na formu za stvaranje oglasa ili korisnik odlučuje odustati od kreiranja oglasa te ga Web-aplikacija vraća na početnu stranicu. Kada su uneseni podatci valjani korisnik odabire opciju objave oglasa te Web-aplikacija oglas pohranjuje u bazu podataka. Na kraju Web-aplikacija nakon uspješnog pohranjivanja oglasa u bazu podataka obavještava korisnika da je oglas objavljen.



Slika 3.6: Sekvencijski dijagram za UC7

## 3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške

## 4. Arhitektura i dizajn sustava

Sustav je predviđen za pristup više korisnika te je za implementaciju odabrana arhitektura Web-aplikacije. Pristup aplikaciji je moguć preko Web-preglednika, tj. nije potrebna nikakva instalacija posebne programske potpore.

Za rad aplikacije neophodna je baza podataka u kojoj se pohranjuju svi podatci potrebni za funkcioniranje sustava.

Arhitektura je podijeljena na tri podsustava:

- Web aplikacija
- Web poslužitelj
- Baza podataka

*Web poslužitelj* je obično računalo koje služi za dijeljenje sadržaja te izgradnju internetskih stranica. Poslužitelji su otvoreni preko domene (internetske adrese) ili IP adrese te svatko može s bilo kojega računala pristupiti nekome poslužitelju i nekoj internetskoj stranici. Poslužitelj procesira zahtjeve korisnika putem HTTP/HTTPS protokola pomoću kojeg komunicira s web preglednikom.

*Web aplikacija* omogućava korisniku komunikaciju sa sustavom. Osnovna zadaća Web-aplikacije je obrada korisničkih zahtjeva. Također prima podatke od strane korisnika te ih obrađuje i pohranjuje. Web-aplikacija komunicira s bazom podataka te može upisivati nove podatke, mijenjati postojeće te čitati postojeće podatke.

Backend naše aplikacije smo razvili pomoću jezika *Javascript* točnije NodeJS-a i jednostavnog razvojnog okvira *Express*. Za sustav upravljanja baze podataka odabrali smo *postgresql* zbog kvalitetne integracije s Expressom. Frontend naše aplikacije razvili smo također pomoću jezika *Javascript* i reaktivnog radnog okvira *vueJS*.

*Arhitektura sustava* se temelji na MVC arhitekturi. Odabrali smo taj koncept jer se odvaja korisničko sučelje od ostatka sustava, što čini razvoj i nadogradnju komponenata jednostavnijima. Sastoji se od modela, pogleda i upravitelja.

- Model - dojavljuje sebi pridruženim pogledima i upravitelju kada je došlo do promjene u njegovom stanju. Ove dojave omogućuju pogledu da prikaže obnovljeno stanje modela, a upravitelju promjenu dostupnog skupa naredbi

- View(Pogled) - od modela dobija informacije koje su mu potrebne za prikaz korisniku
- Controller(Upravitelj) - šalje naloge modelu koji ažurira svoje stanje i naredbe pogledima kojima mijenja prikaz modela

## 4.1 Baza podataka

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

- Registered
- Moderator
- Post
- Replies
- Grade
- DeletedPost
- Category
- Course
- StudyProgramme

### 4.1.1 Opis tablica

**Registered** Ovaj entitet sadržava sve važne informacije o registriranom korisniku aplikacije. Sadrži attribute: userID, firstName, lastName, userName, userAvatar, password, email i created. Ovaj entitet je generalizacija entiteta Moderator. Ovaj entitet u vezi je  $1:N$  s entitetom Post preko userID, u vezi je  $1:N$  s entitetom Replies preko userID i u dvosturkoj vezi je  $0..1:N$  s entitetom Grade preko userID-a.

Registered		
userID	INT	jedinstveni identifikator korisnika

Nastavljeno na idućoj stranici



Nastavljeno od prethodne stranice

Registered		
FirstName	VARCHAR	ime korisnika
LastName	VARCHAR	prezime korisnika
UserName	VARCHAR	korisničko ime korisnika
UserAvatar	VARCHAR	profilna slika korisničkog računa
Password	VARCHAR	hash lozinke
Email	VARCHAR	email adresa korisnika
Created	DATE	datum stvaranja korisničkog računa

**Moderator** Ovaj entitet je specijalizacija entiteta Registered. Entitet je namijenjen korisnicima koji imaju ulogu moderatora i njegove funkcionalnosti. Sadrži atribut userID. Ovaj entitet u vezi je 1:N s entitetom DeletedPost preko userID.

Moderator		
userID	INT	jedinstveni identifikator korisnika

**DeletedPost** Ovaj entitet sadržava informacije o moderatoru aplikacije i oglasu koji je taj moderator izbrisao uz dano objašnjenje. Sadrži attribute: postID, deleto-  
rUserID i explanation. Ovaj entitet u vezi je 1:N s entitetom Post preko postID-a.

DeletedPost		
postID	INT	jedinstveni identifikator oglasa
deletorUserID	INT	jedinstveni identifikator moderatora
explanation	VARCHAR	objašnjenje brisanja oglasa

**Post** Ovaj entitet je identifikacijski slab entitet kojeg entitet Registered identificira. Ovaj entitet sadržava sve važne informacije o objavljenom oglasu. Sadrži attribute: postID, postTitle, postDescription, timeOfCreation, creatorUserID i categoryID. Ovaj entitet u vezi je 1:N s entitetom Replies preko postID-a, u vezi je N:1 s entite-  
tom Category preko categoryID, u vezi je 0..1:N s entitetom Grade preko postID-a, u vezi je 1:N s entitetom DeletedPost preko postID-a i u vezi je N:1 s entitetom Registered preko userID-a.

Post		
postID	INT	jedinstveni identifikator oglasa
postTitle	VARCHAR	naslov oglasa
postDescription	VARCHAR	opis oglasa
timeOfCreation	DATE	vrijeme kreiranja oglasa
creatorUserID	INT	jedinstveni identifikator kreatora oglasa
categoryID	INT	jedinstveni identifikator kategorije

**Replies** Ovaj entitet je identifikacijski slab entitet kojeg entitet Registered identificira. Ovaj entitet sadržava sve važne informacije o odgovoru tj. upitu za oglas. Sadrži attribute: replyID, replyText, replyCreated, statusValue, replyCreatorID i postID. Ovaj entitet u vezi je *N:1* s entitetom Post preko postID-a i u vezi je *N:1* s entitetom Registered preko replyCreatorID-a.

Replies		
replyID	INT	jedinstveni identifikator upita
replyText	VARCHAR	tekst upita
replyCreated	DATE	vrijeme kreiranja upita
statusValue	VARCHAR	status upita
replyCreatorID	INT	jedinstveni identifikator kreatora upita
postID	INT	jedinstveni identifikator oglasa

**Grade** Ovaj entitet je identifikacijski slab entitet kojeg entitet Registered i entitet Post identificiraju. Ovaj entitet sadržava informacije o ocjeni studenta-pomagača. Sadrži attribute: instructorID, learnerID, postID i grade. Ovaj entitet u vezi je *N:1* s entitetom Post preko postID-a i dvostruko je vezan *N:1* s entitetom Registered preko instructorID-a i learnerID-a.

Grade		
instructorID	INT	jedinstveni identifikator studenta-pomagača

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Grade		
learnerID	INT	jedinstveni identifikator studenta koji traži pomoć
postID	INT	jedinstveni identifikator oglasa
grade	INT	ocjena dana studentu-pomagaču za dane instrukcije

**Category** Ovaj entitet sadržava informacije o kategoriji oglasa. Sadrži attribute: categoryID, categoryName i courseID. Ovaj entitet u vezi je 1:N s entitetom Post preko categoryID i u vezi je N:1 s entitetom Course preko courseID-a.

Category		
categoryID	INT	jedinstveni identifikator kategorije(MI, blic, Ispitni rok)
categoryName	VARCHAR	ime kategorije
courseID	INT	jedinstveni identifikator kolegija

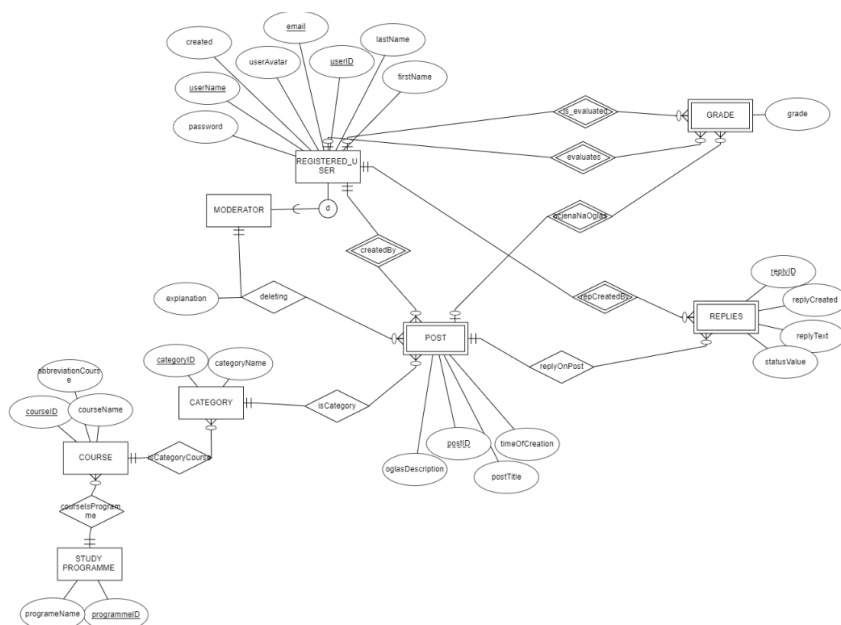
**Course** Ovaj entitet sadržava informacije o kolegiju. Sadrži attribute: courseID, courseName i abbreviationCourse i programmeID. Ovaj entitet u vezi je 1:N s entitetom Category preko categoryID i u vezi je N:1 s entitetom Study Programme preko programmeID-a.

Course		
courseID	INT	jedinstveni identifikator kolegija
courseName	VARCHAR	ime kolegija
abbreviationCourse	VARCHAR	kratica kolegija
programmeID	INT	jedinstveni identifikator smjera(E ili R)

**Study Programme** Ovaj entitet sadržava informacije o smjer kojem kolegij pripada. Sadrži attribute: programmeID i programmeName. Ovaj entitet u vezi je 1:N s entitetom Course preko programmeID-a.

Study Programme		
programmeID	INT	jedinstveni identifikator kolegija
programmeName	VARCHAR	ime smjera

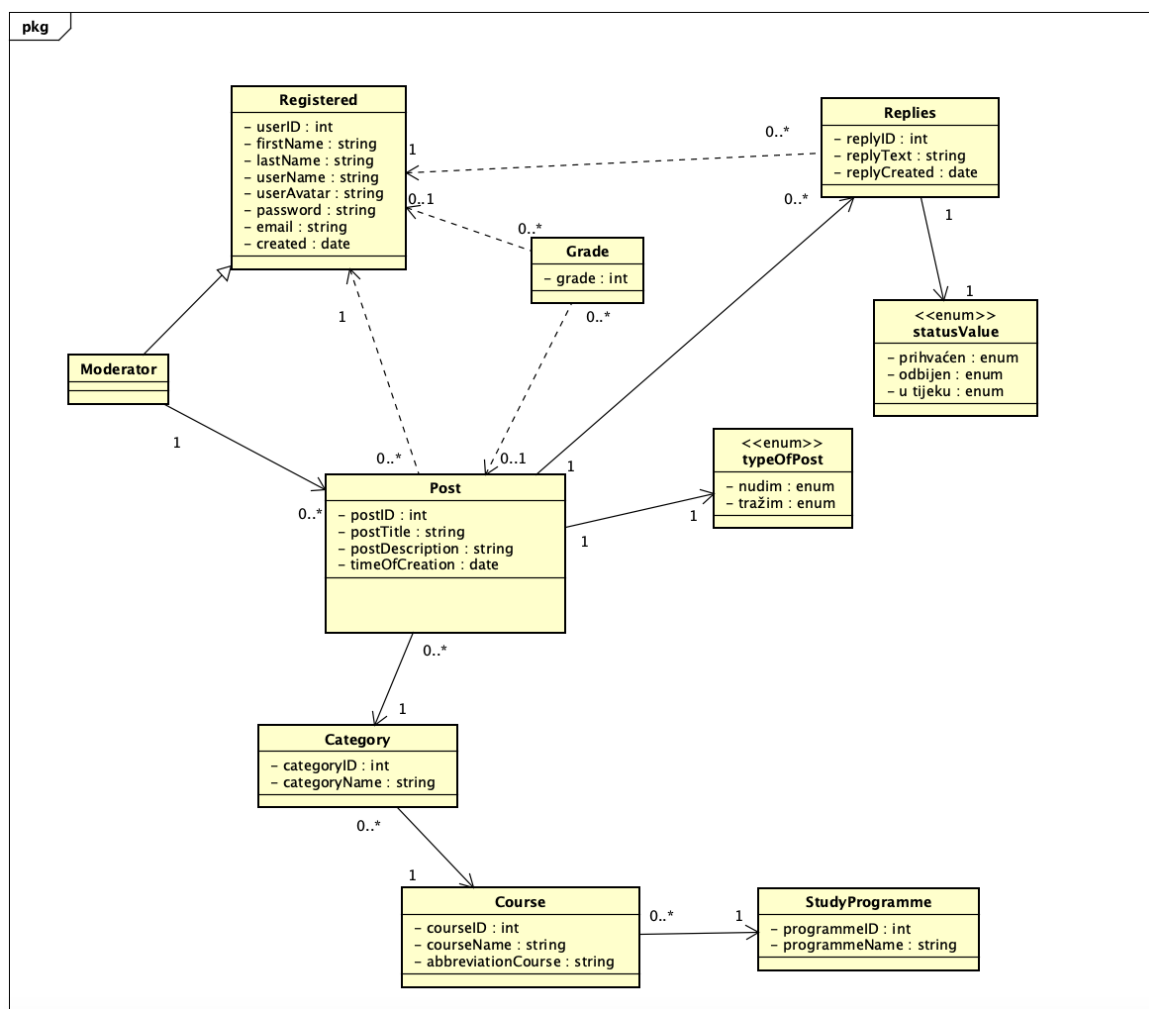
### 4.1.2 Dijagram baze podataka



Slika 4.1: E-R dijagram baze podataka.

## 4.2 Dijagram razreda

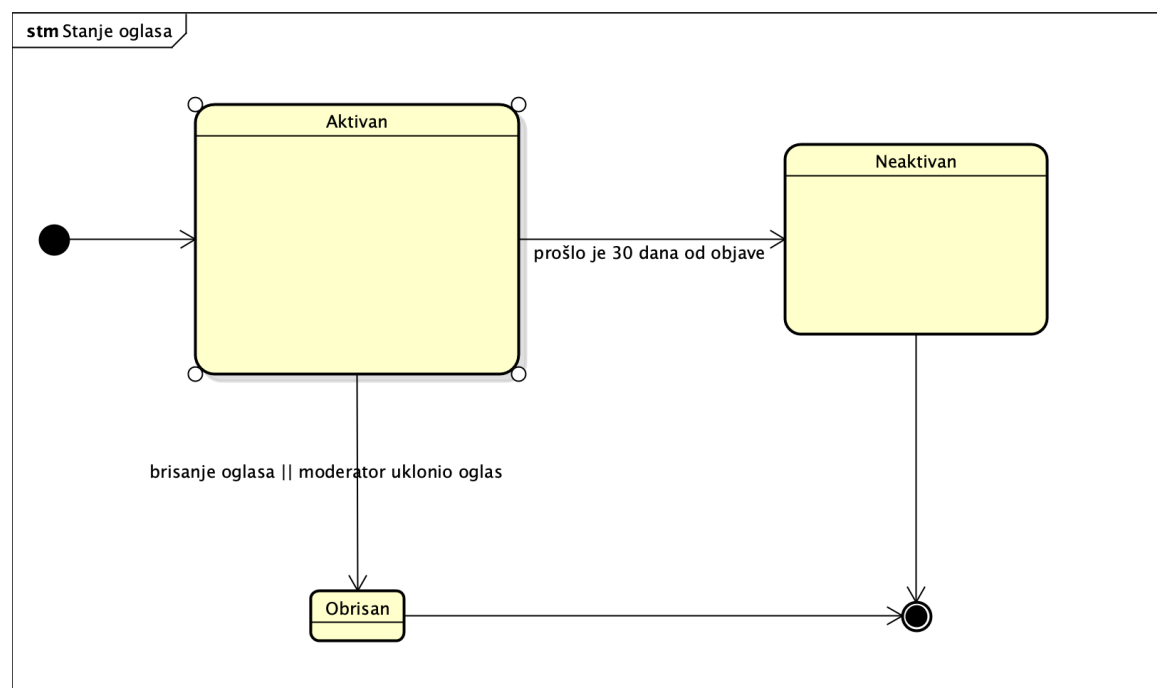
Na slici je prikazan dijagram razreda. S obzirom da smo koristili NodeJS na backendu, nemamo implementirane metode u klasama. Na ovom se dijagramu nalaze središnji objekti naše aplikacije: *Registered*, *Post*, *Category*... Navedeni su atributi svake klase i njihova vrsta (npr. *int*), odnosi međusobnih relacija i enumeracije.



Slika 4.2: Dijagram razreda.

## 4.3 Dijagram stanja

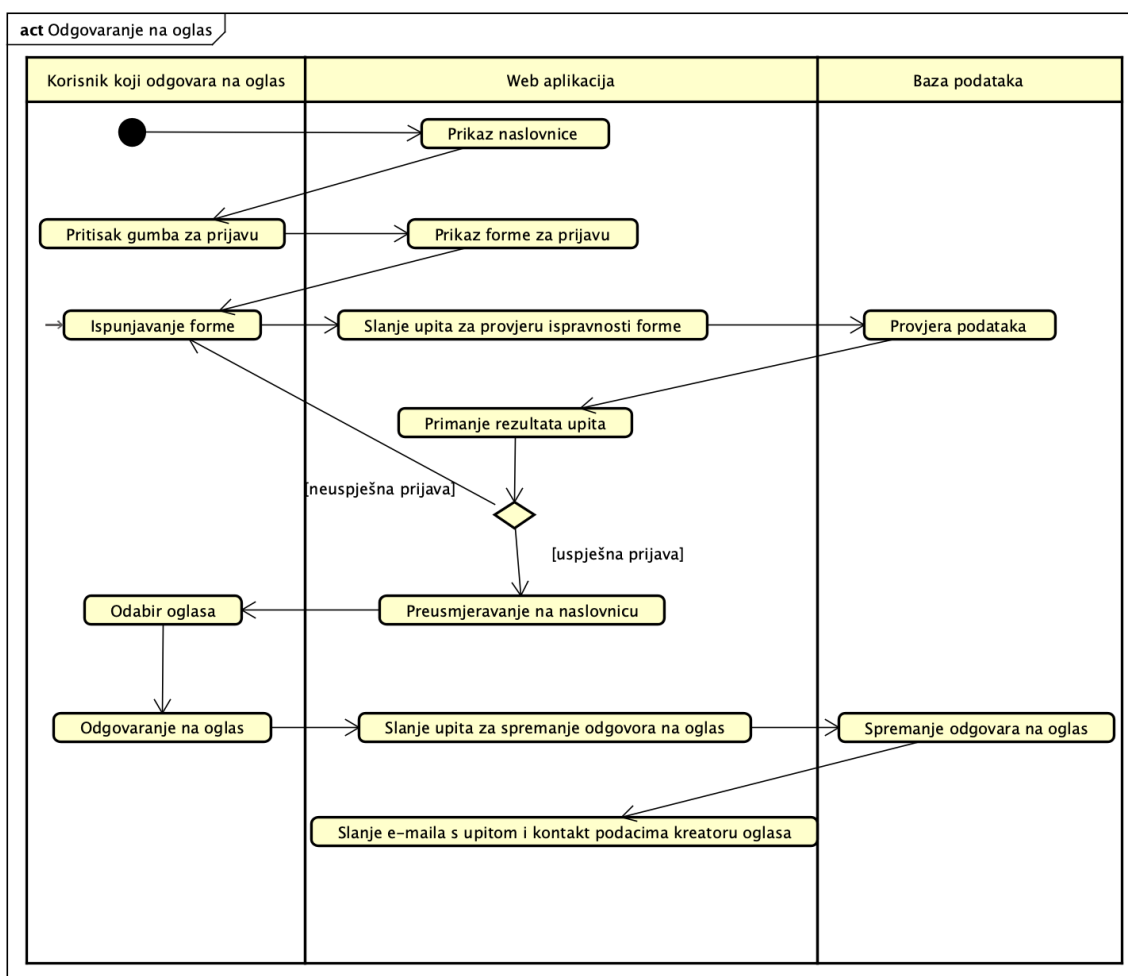
*U trenutku objave oglas postaje aktivan i njegovo se vrijeme objave zapisuje u bazu podataka. Nakon 30 dana oglas postaje neaktivan. Korisnik koji je objavio oglas može isti i obrisati, isto pravo ima i moderator u slučaju nepravilnog ili neprikladnog oglasa.*



Slika 4.3: Dijagram stanja oglasa.

## 4.4 Dijagram aktivnosti

U dijagramu aktivnosti je prikazano korisnikovo odgovaranje na oglas. Korisnik otvara web aplikaciju te bira prijava i ispunjuje formu. Ukoliko je web aplikacija pomoću baze podataka verificirala njegovu prijavu, korisnik se prijavljen vraća nazad na naslovnicu, a ako prijava nije uspješna vraća ga na ispunjavanje forme. Nakon toga korisnik odabire oglas i odgovara na njega slanjem poruke koju web aplikacija sprema u bazu podataka i šalje kreatoru oglasa.

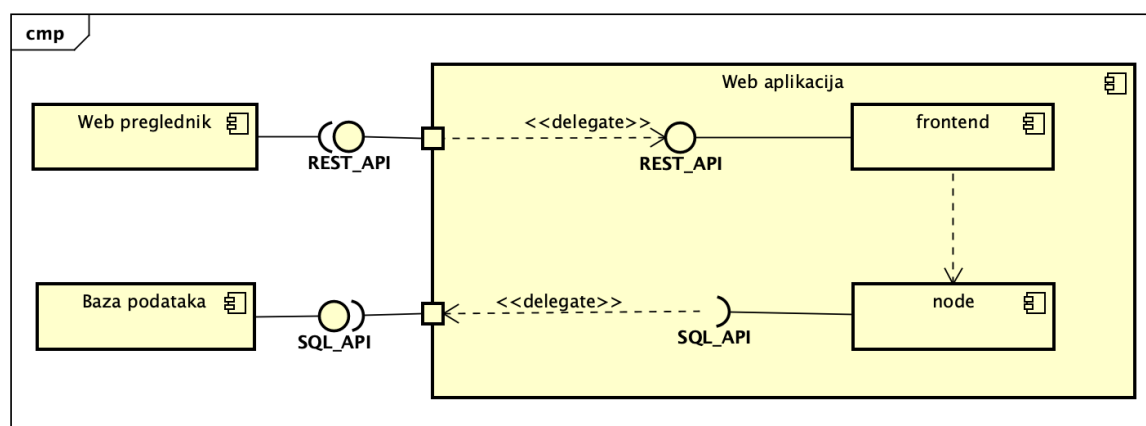


Slika 4.4: Dijagram aktivnosti, odgovaranje na oglas



## 4.5 Dijagram komponenti

Web aplikacija se sastoji od dvije komponente: frontend i node. Frontend pruža REST\_API sučelje na koje vanjski web preglednik može slati zahtjeve. Node preko SQL\_API sučelja uzima podatke iz baze podataka i servira ih frontendu kako bi korisnik uspješno koristio web aplikaciju.



Slika 4.5: Dijagram komponenti.

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je putem Discord-a<sup>1</sup>. Za izradu UML dijagrama koristili smo Astah Professional<sup>2</sup>, a za izradu dokumentacije LaTeX Overleaf<sup>3</sup>. Kako bi upravljali izvornim kodom i međusobno ga nadopunjavali, koristili smo Git<sup>4</sup>, a za spremanje udaljenog repozitorija služi nam web platforma GitLab<sup>5</sup>.

Za implementaciju frontenda web aplikacije koristili smo prezentacijski jezik HTML<sup>6</sup> te radne okvire Vue.js<sup>7</sup> (reaktivnost, velika količina materijala za učenje) i Bootstrap<sup>8</sup> (jednostavan razvoj). Za izradu backenda smo koristili Node.js<sup>9</sup> (velika količina biblioteka, brz i jednostavan razvoj). Bazu podataka smo implementirali putem PostgreSQL-a<sup>10</sup>.

---

<sup>1</sup><https://discord.com/>

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

<sup>3</sup><https://www.overleaf.com/>

<sup>4</sup><https://git-scm.com/>

<sup>5</sup><https://gitlab.com/>

<sup>6</sup><https://html.com/>

<sup>7</sup><https://vuejs.org/>

<sup>8</sup><https://getbootstrap.com/>

<sup>9</sup><https://nodejs.org/en/>

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

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

Napravljena su 5 testa koristeći *Mochu*, JavaScript test runner te smo ispitali funkcioniranje registracije korisnika. Ispitali su se slučajevi:

1. Sva polja su popunjena
2. Nije predan dokument u png formatu
3. Nije uneseno prezime korisnika
4. Nije zadovoljen format email adrese
5. Korisničko ime je zauzeto

```
(base) C:\Users\Maks\Desktop\PROGIprojektBAZA\sinapsa\node>npm run test
Debugger attached.

> node@1.0.0 test
> mocha

Waiting for the debugger to disconnect...
Debugger attached.

Register
  ✓ should return 201 when all required fields are filled and email is valid
  ✓ should return 500 when file upload fails
  ✓ should return 400 when a required field is missing (lastname)
  ✓ should return 400 when username is already taken
  ✓ should return 400 when email is invalid

5 passing (13ms)

Waiting for the debugger to disconnect...
Waiting for the debugger to disconnect...
```

Slika 5.1: Rezultati testova - RegisterTest

```
it('should return 201 when all required fields are filled and email is valid', function(done) {
  request(app)
    .post('/register')
    .field('firstname', 'John')
    .field('lastname', 'Doe')
    .field('username', 'johndoe1')
    .attach('picture', './public/sinisa.png') // attach the file
    .field('password', 'password')
    .field('email', 'mk53370@fer.hr')
    .expect(201)
    .then(async () => {
      await new Promise((resolve) => setTimeout(resolve, 50000));
      console.log("ide gas")
      done();
    });
});

it('should return 500 when file upload fails', function(done) {
  request(app)
    .post('/register')
    .field('firstname', 'John')
    .field('lastname', 'Doe')
    .field('username', 'johndoe2')
    .attach('picture', './public/sinisa14.png') // attach the file
    .field('password', 'password')
    .field('email', 'mk53370@fer.hr')
    .expect(500)
    .then(async () => {
      await new Promise((resolve) => setTimeout(resolve, 50000));
      console.log("ide gas")
      done();
    });
});
```

Slika 5.2: Isječak koda - Testiranje ispravnosti

```
it('should return 400 when a required field is missing (lastname)', function(done) {
  request(app)
    .post('/register')
    .field('firstname', 'John')
    .field('username', 'johndoe3')
    .attach('picture', './public/sinisa.png') // attach the file
    .field('password', 'password')
    .field('email', 'mk53370@fer.hr')
    .expect(400)
    .then(async () => {
      await new Promise((resolve) => setTimeout(resolve, 50000));
      console.log("ide gas")
      done();
    });
});

it('should return 400 when email is invalid', function(done) {
  request(app)
    .post('/register')
    .field('firstname', 'John')
    .field('lastname', 'Doe')
    .field('username', 'johndoe4')
    .attach('picture', './public/sinisa.png') // attach the file
    .field('password', 'password')
    .field('email', 'johndoe@gmail.com')
    .expect(400)
    .then(async () => {
      await new Promise((resolve) => setTimeout(resolve, 50000));
      console.log("ide gas")
      done();
    });
});
```

Slika 5.3: Isječak koda - Testiranje ispravnosti

```
it('should return 400 when username is already taken', function(done) {
  request(app)
    .post('/register')
    .field('firstname', 'John')
    .field('lastname', 'Doe')
    .field('username', 'gazda') //gazda already exists as user
    .attach('picture', './public/sinisa.png') // attach the file
    .field('password', 'password')
    .field('email', 'mk53370@fer.hr')
    .expect(400)
    .then(async () => {
      await new Promise((resolve) => setTimeout(resolve, 50000));
      console.log("ide gas")
      done();
    });
});
```

Slika 5.4: Isječak koda - Testiranje ispravnosti

Napravljen je 1 test za ispitivanje funkcionalnosti login-a tj. može li se postojeći korisnik prijaviti u sustav ako upiše krivu lozinku

```
describe("Login", () => {
  let stub;
  beforeEach(() => {
    stub = sinon.stub(db, 'query');
  });
  afterEach(() => {
    stub.restore();
  });
  it("should return a 401 status code if username and password do not match a registered user", (done) => {
    stub.returns(Promise.resolve({rowCount:0}));
    request(login)
      .post("/login")
      .send({
        username: "gazda",
        password: "kriviPassword"
      })
      .expect(401)
      .end((err, res) => {
        assert.deepEqual(res.body, { status: "failed" });
        done();
      });
  });
});
```

Slika 5.5: Rezultat testa - LoginTest

```
(base) C:\Users\Maks\Desktop\PROGIprojektBAZA\sinapsa\node>npm run test
Debugger attached.

> node@1.0.0 test
> mocha

Debugger attached.

Register
  ✓ should return a 401 status code if username and password do not match a registered user

1 passing (7ms)

Waiting for the debugger to disconnect...
Waiting for the debugger to disconnect...
```

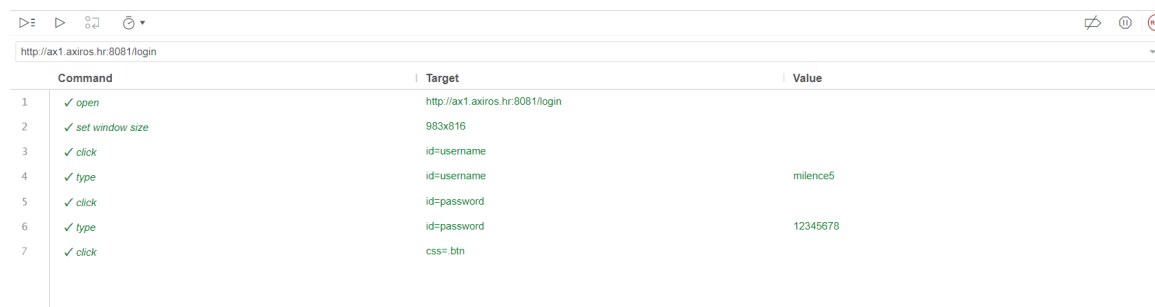
Slika 5.6: Isječak testa - LoginTest

## 5.2.2 Ispitivanje sustava

Svaki dio sustava ispitan je ručno po obrascima uporabe koristeći *Selenium IDE* kako bi se pronašla neočekivana ponašanja aplikacije. Prikazan je dio ispitivanja sustava zbog jednostavnosti.

### Ispitni slučaj 1: UC-2 prijava u sustav

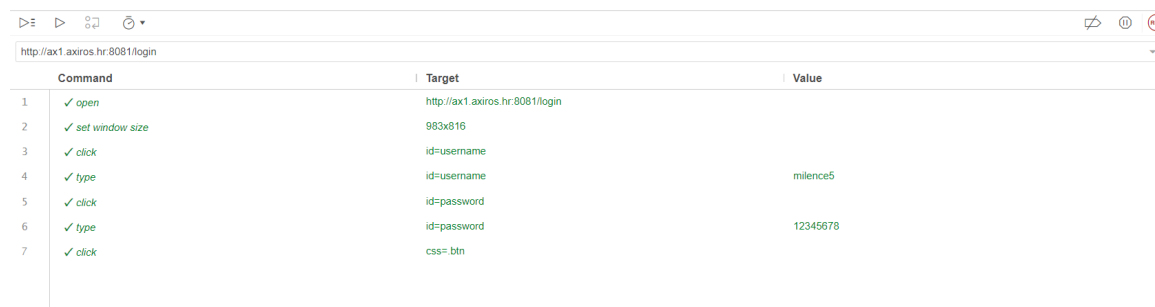
Pokušavamo se prijaviti u sustav koristeći valjani username, ali krivu lozinku



	Command	Target	Value
1	✓ open	http://ax1.axiros.hr:8081/login	
2	✓ set window size	983x816	
3	✓ click	id=username	
4	✓ type	id=username	milence5
5	✓ click	id=password	
6	✓ type	id=password	12345678
7	✓ click	css=.btn	

Slika 5.7: Isječak testa - LoginTest neuspjeh

Sustav je prošao test i obavijestio korisnika da je nevaljani unos  
Prijavljujemo se u sustav unoseći ispravan username i lozinku



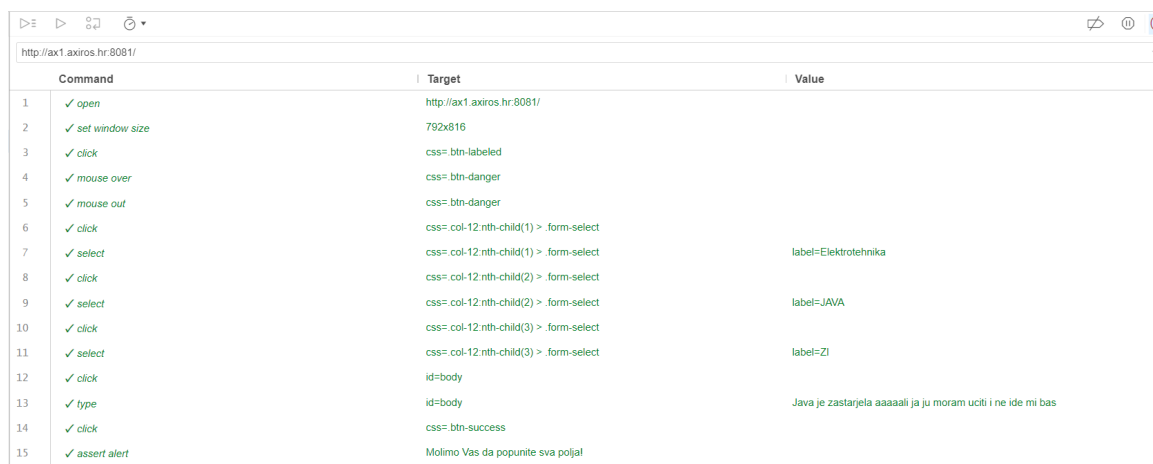
	Command	Target	Value
1	✓ open	http://ax1.axiros.hr:8081/login	
2	✓ set window size	983x816	
3	✓ click	id=username	
4	✓ type	id=username	milence5
5	✓ click	id=password	
6	✓ type	id=password	12345678
7	✓ click	css=.btn	

Slika 5.8: Isječak testa - LoginTest uspjeh

### Ispitni slučaj 2: UC-9 objava oglasa

Pokušavamo objaviti oglas, pritom ne unosimo naslov samog oglasa, ali nas sustav o tome obavještava.

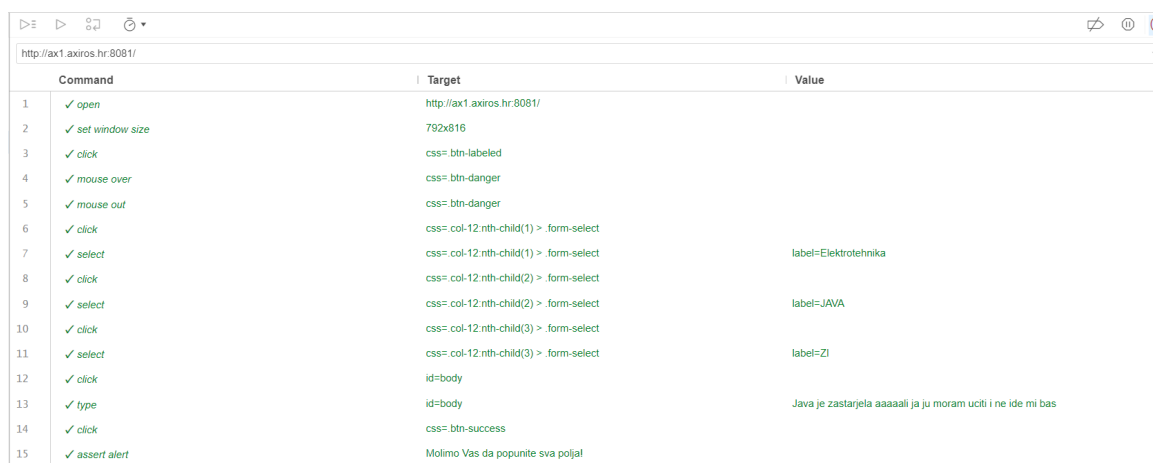




	Command	Target	Value
1	✓ open	http://ax1.axiros.hr/8081/	
2	✓ set window size	792x816	
3	✓ click	css= btn-labeled	
4	✓ mouse over	css= btn-danger	
5	✓ mouse out	css= btn-danger	
6	✓ click	css= col-12.nth-child(1) > .form-select	
7	✓ select	css= col-12.nth-child(1) > .form-select	label=Elektrotehnika
8	✓ click	css= col-12.nth-child(2) > .form-select	
9	✓ select	css= col-12.nth-child(2) > .form-select	label=JAVA
10	✓ click	css= col-12.nth-child(3) > .form-select	
11	✓ select	css= col-12.nth-child(3) > .form-select	label=ZI
12	✓ click	id=body	
13	✓ type	id=body	Java je zastarjela aaaaaa! ja ju moram uci i ne ide mi bas
14	✓ click	css= btn-success	
15	✓ assert alert	Molimo Vas da popunite sva polja!	

Slika 5.9: Isječak testa - PostTest neuspjeh

Objavljujemo oglas nakon što smo unijeli sve potrebne podatke, te nas sustav vraća na početnu stranicu

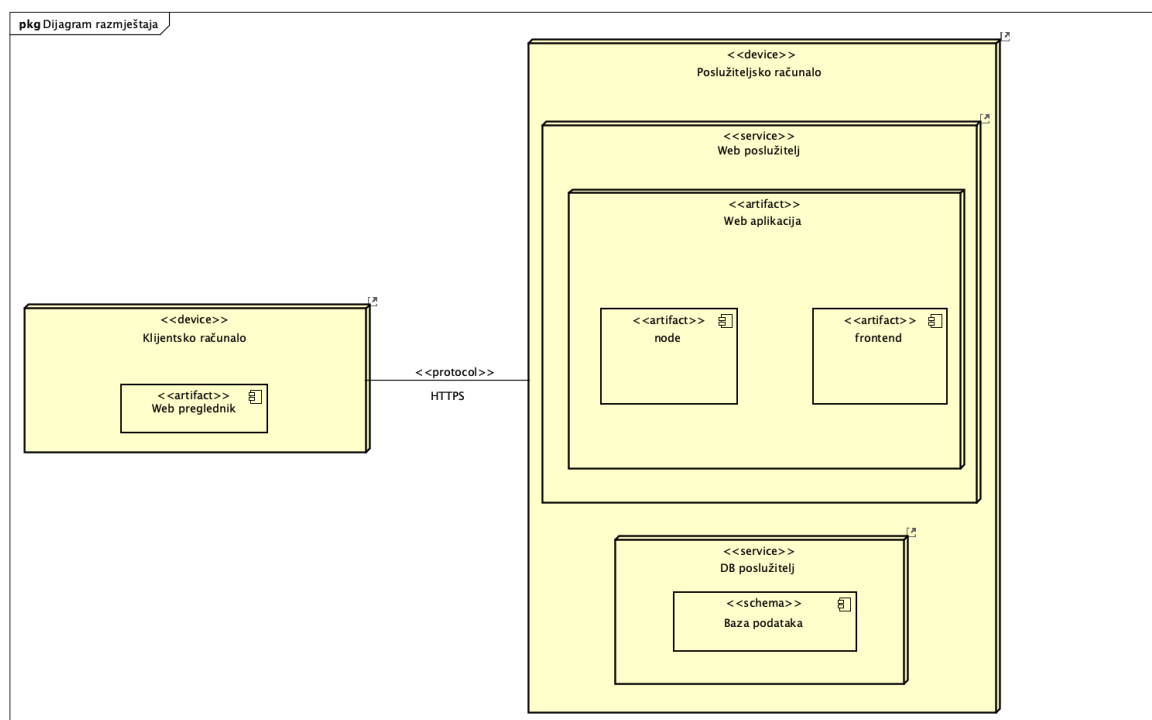


	Command	Target	Value
1	✓ open	http://ax1.axiros.hr/8081/	
2	✓ set window size	792x816	
3	✓ click	css= btn-labeled	
4	✓ mouse over	css= btn-danger	
5	✓ mouse out	css= btn-danger	
6	✓ click	css= col-12.nth-child(1) > .form-select	
7	✓ select	css= col-12.nth-child(1) > .form-select	label=Elektrotehnika
8	✓ click	css= col-12.nth-child(2) > .form-select	
9	✓ select	css= col-12.nth-child(2) > .form-select	label=JAVA
10	✓ click	css= col-12.nth-child(3) > .form-select	
11	✓ select	css= col-12.nth-child(3) > .form-select	label=ZI
12	✓ click	id=body	
13	✓ type	id=body	Java je zastarjela aaaaaa! ja ju moram uci i ne ide mi bas
14	✓ click	css= btn-success	
15	✓ assert alert	Molimo Vas da popunite sva polja!	

Slika 5.10: Isječak testa - PostTest uspjeh

## 5.3 Dijagram razmještaja

Dijagram razmještaja prikazuje topologiju sustava prilikom puštanja aplikacije u pogon. Klijentsko i poslužiteljsko računalo komuniciraju putem HTTPS protokola. Na poslužiteljskom računalu se nalaze web poslužitelj i DB poslužitelj.



Slika 5.11: Specifikacijski dijagram razmještaja.

## 5.4 Upute za puštanje u pogon

Proces puštanja aplikacije u pogon je sljedeći. Prvo je potrebno pokrenuti sustav "NGINX" koji radi kao poslužitelj. Treba ga konfigurirati tako da poslužuje statične resurse aplikacije (VueJS kod nakon što ga se pripremi za produkciju) kad se pristupa portu 80. Za pripremu VueJS koda za produkciju potrebno se pozicionirati u direktorij "frontend" i pokrenuti naredbu "npm run build". Ta naredba osigurava optimizaciju i smanjivanje izvornog koda radi poboljšanja performansi. Pošto je aplikacija rađena po principu aplikacija od jedne stranice ("single page application") potrebno je zasebno pokrenuti stražnji kraj aplikacije ("backend"). Pošto je "backend" pisan u programskom jeziku JavaScript i izvodi se u okolini izvođenja ("runtime environment") NodeJS potrebno je instalirati program "pm2". Taj program koristimo za nadziranje i pokretanje NodeJS poslužitelja. Nakon instalacije potrebno se pozicionirati u direktorij "node" i izvršiti naredbu "pm2 start server.js". Također je potrebno instalirati sustav upravljanja bazama podataka PostgreSQL i stvoriti bazu imena "sinapsa". Po potrebi treba promijeniti lozinku i korisničko ime za spajanje na bazu u datoteci node/db/index.js.

## 6. Zaključak i budući rad

*Zadatak naše grupe bio je razvoj web aplikacije Sinappsa putem koje studenti mogu ponuditi ili zatražiti pomoć od svojih kolega. Za taj nam je projekt trebalo 17 tjedana podijeljenih u dvije faze.*

*U prvoj fazi smo dogovorili način i alate implementacije te rasporedili uloge članovima tima. Izradili smo obrasce i dijagrame kako bi mogli započeti s razvojem naše aplikacije. Dokumentacija nam je pomogla da nemamo pogreške radi nejasnoći ili nesporazuma i uspješno krenemo s izradom aplikacije. U prvoj fazi implementirali smo registraciju i prijavu te povezali web aplikaciju s bazom podataka.*

*Druga faza je bila nešto drugačija jer smo se ovdje više usredotočili na izradu aplikacije, ali je bilo i posla s dokumentacijom koja omogućuje budućim korisnicima da vrše preinaku sustava. Kvalitetna podjela poslova prema znanjima i mogućnostima članova nam je omogućila da svaki član bude zadovoljan zadatkom koji je dobio i pokaže svoje kvalitete u istome. Ipak je mnoge dočekalo učenje alata s kojima se nikad nisu susreli. Uspješno smo odgovorili na sve probleme i implementirali sve željene funkcionalnosti.*

*Komunikacija tima bila je putem Discorda gdje smo raspravljali nedoumice i pratili napredak.*

*Uspješno smo implementirali sve funkcionalnosti osim izmjene avatara registriranog korisnika.*

*Projekt Sinappsa je bio vrijedno iskustvo svim članovima od kojih se mnogi prvi put susreću s timskim radom. U ovom semestru smo se upoznali s prednostima i povremenim problemima timskog rada te se naučili organizirati veliki projekt i vrijeme uz sve ostale fakultetske obaveze. Prostora za napredak ima puno, ali zadovoljni smo radom i uspjehom nad ovim projektom.*

# Popis literature

## *Kontinuirano osvježavanje*

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

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

# Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe funkcionalnosti svih korisnika, neregistriranog korisnika, moderatora i baze podataka . . . . .	17
3.2	Dijagram obrasca uporabe registriranog korisnika vezano uz oglase	17
3.3	Dijagram obrasca uporabe registriranog korisnika vezane uz profil .	18
3.4	Sekvencijski dijagram za UC1 . . . . .	19
3.5	Sekvencijski dijagram za UC6 . . . . .	19
3.6	Sekvencijski dijagram za UC7 . . . . .	20
4.1	E-R dijagram baze podataka. . . . .	28
4.2	Dijagram razreda. . . . .	29
4.3	Dijagram stanja oglasa. . . . .	30
4.4	Dijagram aktivnosti, odgovaranje na oglas . . . . .	31
4.5	Dijagram komponenti. . . . .	32
5.1	Rezultati testova - RegisterTest . . . . .	34
5.2	Isječak koda - Testiranje ispravnosti . . . . .	35
5.3	Isječak koda - Testiranje ispravnosti . . . . .	36
5.4	Isječak koda - Testiranje ispravnosti . . . . .	37
5.5	Rezultat testa - LoginTest . . . . .	37
5.6	Isječak testa - LoginTest . . . . .	38
5.7	Isječak testa - LoginTest neuspjeh . . . . .	39
5.8	Isječak testa - LoginTest uspjeh . . . . .	39
5.9	Isječak testa - PostTest neuspjeh . . . . .	40
5.10	Isječak testa - PostTest uspjeh . . . . .	40
5.11	Specifikacijski dijagram razmještaja. . . . .	41
6.1	Prikaz aktivnosti 1 na repozitoriju. . . . .	51
6.2	Prikaz aktivnosti 2 na repozitoriju. . . . .	51
6.3	Prikaz aktivnosti 3 na repozitoriju. . . . .	52
6.4	Prikaz aktivnosti 4 na repozitoriju. . . . .	52
6.5	Prikaz aktivnosti 5 na repozitoriju. . . . .	53

6.6	Prikaz aktivnosti 6 na repozitoriju. . . . .	53
6.7	Prikaz aktivnosti 7 na repozitoriju. . . . .	54
6.8	Prikaz aktivnosti 8 na repozitoriju. . . . .	54
6.9	Prikaz aktivnosti 9 na repozitoriju. . . . .	55
6.10	Prikaz aktivnosti 10 na repozitoriju. . . . .	55
6.11	Prikaz aktivnosti 11 na repozitoriju. . . . .	56
6.12	Prikaz aktivnosti 12 na repozitoriju. . . . .	56
6.13	Prikaz aktivnosti 13 na repozitoriju. . . . .	57
6.14	Prikaz aktivnosti 14 na repozitoriju. . . . .	57
6.15	Prikaz aktivnosti 15 na repozitoriju. . . . .	58
6.16	Prikaz aktivnosti 16 na repozitoriju. . . . .	58
6.17	Prikaz aktivnosti 17 na repozitoriju. . . . .	59
6.18	Prikaz aktivnosti 18 na repozitoriju. . . . .	59

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

*U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.*

#### 1. sastanak

- Datum: 20. listopada 2022.
- Prisustvovali: M.Đomlija, D.Supančić, J.Šegvić, D.Ljubić, M.Kos, I.Inkret, F.Belac
- Teme sastanka:
  - upoznavanje s projektom i mentorom

#### 2. sastanak

- Datum: 22. listopada 2022.
- Prisustvovali: M.Đomlija, D.Supančić, J.Šegvić, D.Ljubić, M.Kos, I.Inkret, F.Belac
- Teme sastanka:
  - planiranje projekta
  - podjela uloga u timu

#### 3. sastanak

- Datum: 12. studenoga 2022.
- Prisustvovali: M.Đomlija, D.Supančić, J.Šegvić, D.Ljubić, M.Kos, I.Inkret, F.Belac
- Teme sastanka:
  - razrada implementacije web aplikacije
  - podjela poslova članovima tima

#### 4. sastanak

- Datum: 15. studenoga 2022.
- Prisustvovali: M.Đomlija, J.Šegvić, M.Kos, F.Belac
- Teme sastanka:



- zajednički rad na projektu

5. sastanak

- Datum: 3. prosinca 2022.
- Prisustvovali: M.Đomlija, D.Supančić, J.Šegvić, D.Ljubić, M.Kos, I.Inkret, F.Belac
- Teme sastanka:
  - provjera napredka

6. sastanak

- Datum: 17. prosinca 2022.
- Prisustvovali: M.Đomlija, J.Šegvić, D.Ljubić, M.Kos, I.Inkret, F.Belac
- Teme sastanka:
  - zajednički rad na projektu

7. sastanak

- Datum: 7. siječnja 2023.
- Prisustvovali: M.Đomlija, J.Šegvić, M.Kos, D. Ljubić, I.Inkret
- Teme sastanka:
  - završetak implementacije

## Tablica aktivnosti

### Kontinuirano osvježavanje

*Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.*

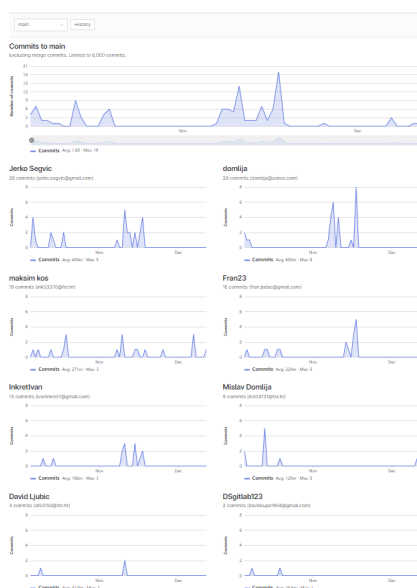
	Mislav Domlija	David Supančić	Jerko Šegvić	David Ljubić	Maksim Kos	Ivan Inkret	Fran Belac
Upravljanje projektom	3						
Opis projektnog zadatka	1				1		
Funkcionalni zahtjevi					1		
Opis pojedinih obrazaca					0.5		
Dijagram obrazaca					1		
Sekvencijski dijagrami					1		
Opis ostalih zahtjeva					0.5		
Arhitektura i dizajn sustava					0.5		
Baza podataka		2			3		
Dijagram razreda		2					
Dijagram stanja		1					
Dijagram aktivnosti		3					
Dijagram komponenti		3					
Korištene tehnologije i alati		1					
Ispitivanje programskog rješenja					6		
Dijagram razmještaja		1					
Upute za puštanje u pogon	1						

Nastavljeno na idućoj stranici

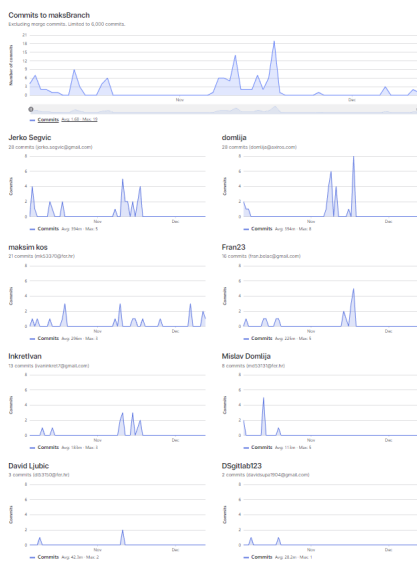
Nastavljeno od prethodne stranice

	Mislav Domlija	David Supančić	Jerko Šegvić	David Ljubić	Maksim Kos	Ivan Inkret	Fran Belac
Dnevnik sastajanja		1					
Zaključak i budući rad		0.5					
Popis literature	0.5						
<i>frontend</i>	12			14	0.5		25
<i>izrada baze podataka</i>		2			4		
<i>backend</i>	12	2	25			17	
<i>ostale izmjene dokumentacije</i>		3					

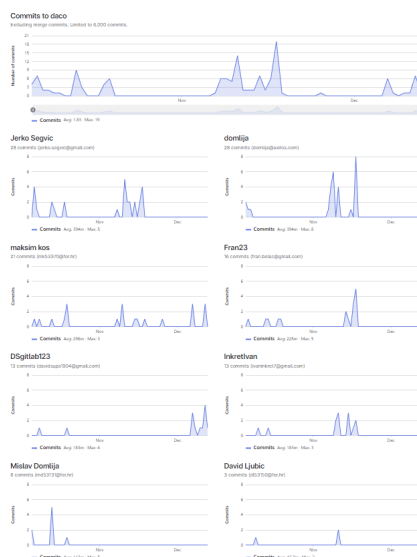
## Dijagrami pregleda promjena



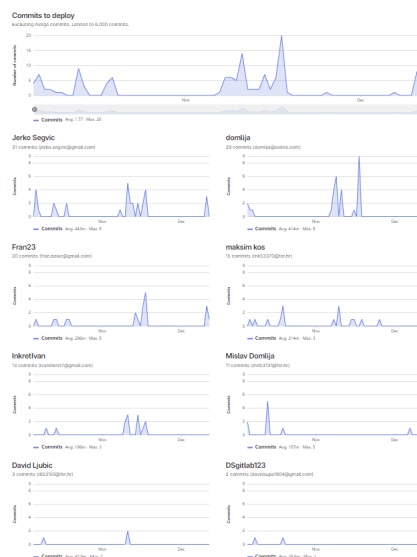
Slika 6.1: Prikaz aktivnosti 1 na repozitoriju.



Slika 6.2: Prikaz aktivnosti 2 na repozitoriju.



Slika 6.3: Prikaz aktivnosti 3 na repozitoriju.



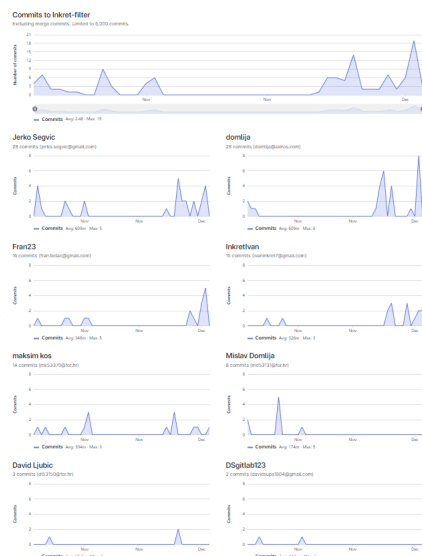
Slika 6.4: Prikaz aktivnosti 4 na repozitoriju.



Slika 6.5: Prikaz aktivnosti 5 na repozitoriju.



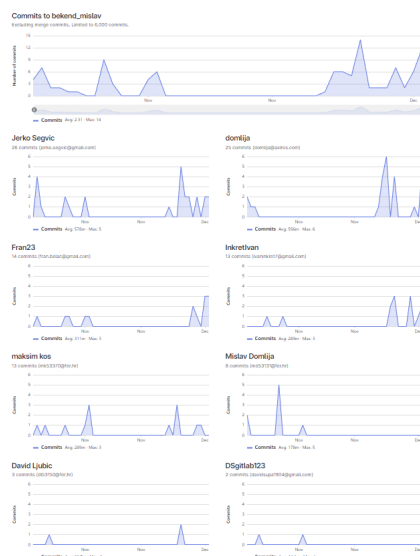
Slika 6.6: Prikaz aktivnosti 6 na repozitoriju.



Slika 6.7: Prikaz aktivnosti 7 na repozitoriju.



Slika 6.8: Prikaz aktivnosti 8 na repozitoriju.

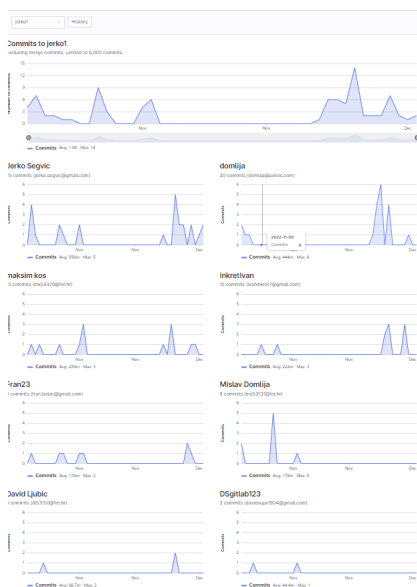


Slika 6.9: Prikaz aktivnosti 9 na repozitoriju.



Slika 6.10: Prikaz aktivnosti 10 na repozitoriju.

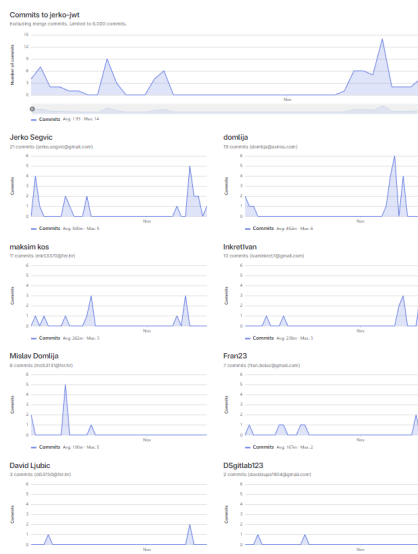




Slika 6.11: Prikaz aktivnosti 11 na repozitoriju.



Slika 6.12: Prikaz aktivnosti 12 na repozitoriju.



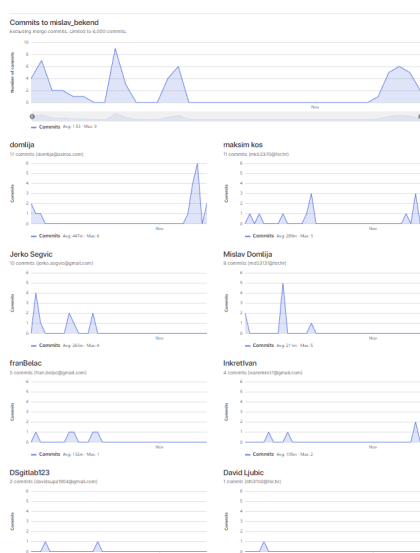
Slika 6.13: Prikaz aktivnosti 13 na repozitoriju.



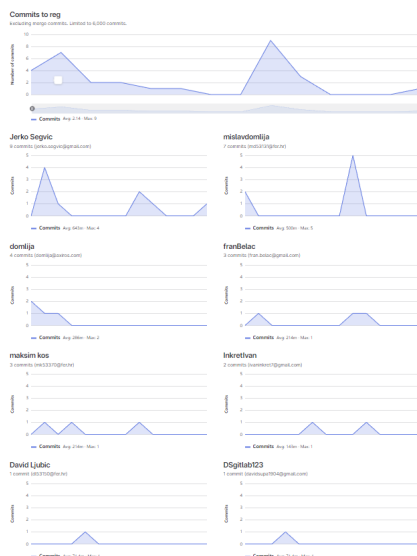
Slika 6.14: Prikaz aktivnosti 14 na repozitoriju.



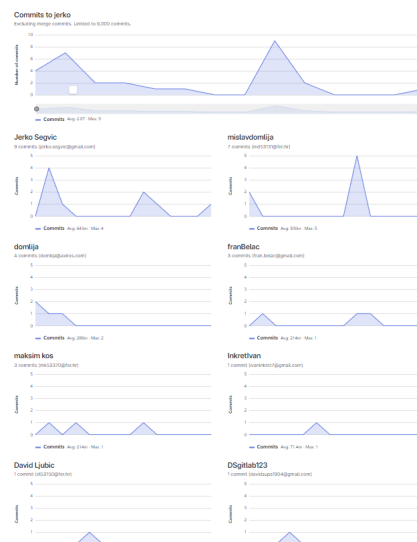
Slika 6.15: Prikaz aktivnosti 15 na repozitoriju.



Slika 6.16: Prikaz aktivnosti 16 na repozitoriju.



Slika 6.17: Prikaz aktivnosti 17 na repozitoriju.



Slika 6.18: Prikaz aktivnosti 18 na repozitoriju.