

Programsko inženjerstvo

Ak. god. 2023./2024.

# Digitalni poster

Dokumentacija, Rev. 1

Grupa: *Poster Rangers*

Voditelj: *Marko Ćurković*

Datum predaje: 17. 11. 2023.

Nastavnik: *Miljenko Krhen*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>13</b>
3.1 Funkcionalni zahtjevi . . . . .	13
3.1.1 Obrasci uporabe . . . . .	15
3.1.2 Sekvencijski dijagrami . . . . .	28
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 . . . . .	36
4.1.2 Dijagram baze podataka . . . . .	41
4.2 Dijagram razreda . . . . .	42
4.3 Dijagram stanja . . . . .	45
4.4 Dijagram aktivnosti . . . . .	46
4.5 Dijagram komponenti . . . . .	48
<b>5 Implementacija i korisničko sučelje</b>	<b>49</b>
5.1 Korištene tehnologije i alati . . . . .	49
5.2 Ispitivanje programskog rješenja . . . . .	51
5.2.1 Ispitivanje komponenti . . . . .	51
5.2.2 Ispitivanje sustava . . . . .	54
5.3 Dijagram razmještaja . . . . .	59
5.4 Upute za puštanje u pogon . . . . .	60
<b>6 Zaključak i budući rad</b>	<b>63</b>
<b>Popis literature</b>	<b>65</b>
<b>Indeks slika i dijagonama</b>	<b>67</b>

**Dodatak: Prikaz aktivnosti grupe**

**68**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Luka Miličević	30.10.2023.
0.2	Napisan opis projektnog zadatka.	Korina Jurić	2.11.2023.
0.2.1	Dodane slike u opis projektnog zadatka.	Korina Jurić	3.11.2023.
0.2.2	Popravljen opis projektnog zadatka.	Korina Jurić	4.11.2023.
0.3	Definirani obrasci uporabe	Marko Ćurković, Luka Lulić	10.11.2023.
0.3.1	Izmijenjeni obrasci uporabe	Luka Lulić	12.11.2023.
0.3.2	Dodani novo-definirani obrasci uporabe	Marko Ćurković	12.11.2023.
0.4	Dodani <i>Use Case</i> dijagrami	Andrija Dumančić	31.10.2023.
0.4.1	Dodani Sekvencijski dijagrami	Luka Miličević, Andrija Dumančić	5.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
0.4.2	Popravljeni <i>Use Case</i> i sekvencijski dijagrami	Luka Miličević	15.11.2023.
0.5	Arhitektura i dizajn sustava, algoritmi i strukture podataka	Daria Bevanda, Fran Talan	1.11.2023.
0.5.1	Izmjene u strukturi baze podataka	Daria Bevanda	3.11.2023.
0.5.2	Promjena dijagrama baze podataka	Daria Bevanda	12.11.2023.
0.6	Dodani dijagrami razreda	Luka Miličević	10.11.2023.
0.6.1	Promijenjeni opisi dijagrama razreda	Luka Miličević	15.11.2023.
<b>1.0</b>	Verzija samo s bitnim dijelovima za 1. ciklus	Korina Jurić	17.11.2023.

## 2. Opis projektnog zadatka

Cilj je projektnog zadatka razviti web-aplikaciju koja bi sudionicima stručne konferencije poboljšala ukupan doživljaj konferencije te im raznim funkcionalnostima olakšala prisustvovanje. Funkcionalnosti koje bi omogućile izvrsno korisničko iskustvo jesu mogućnost direktnog videopraćenja konferencije u glavnoj konferencijskoj dvorani, dio s informacijama o mjestu održavanja konferencije koji između ostalog sadrži podatke o trenutnim vremenskim uvjetima i vremenskoj prognozi za navedenu lokaciju, pregled fotografija s konferencije i lokalno pohranjivanje na vlastiti uređaj te pregled i korištenje promotivnih materijala pokroviteљa. Među istaknutim je funkcionalnostima pregled radova - postera - ostalih sudionika i mogućnost ocjenjivanja onog rada za kojeg smatraju da je najbolji. Budući da organizatori stručnih konferencija ciljaju na što veći broj sudionika, razvojem aplikacije svakako bi postigli veći doseg jer bi se konferenciji moglo priključiti s bilo koje lokacije i u bilo kojem vremenu unutar ukupnog trajanja konferencije.

Superadministrator definira sve potrebne parametre za rad sustava te dodaje druge administratore u sustav koji onda mogu stvarati konferencije. Superadministratoru su dostupni podaci svih konferencija te ima sve ovlasti administratora. Administrator je zadužen za proces prijave odnosno on prijavlje autore i njihove radove nakon što mu autori koji sudjeluju na stručnoj konferenciji elektroničkom poštom pošalju sve potrebne materijale te bira fotografije koje će biti dostupne za pregledavanje i preuzimanje na aplikaciji i postavlja promotivne materijale. Superadministrator i drugi administratori prijavljuju se u sustav na početnoj stranici web-aplikacije.

Sudionici prilikom dolaska na stručnu konferenciju prijavljuju svoj dolazak te pritom dobivaju lozinku za pristup toj određenoj stručnoj konferenciji na web-aplikaciji. Ukoliko sudionik odluči koristiti web-aplikaciju, pri njenom se otvaranju može prvo registrirati u sustav pa onda lozinkom ući u konferenciju ili odmah bez registracije ući u nju. Ako se sudionik prije unosa lozinke konferencije nije registrovao u sustav prikazuju mu se svi stručni radovi, ali nema mogućnost glasanja,

no ako se prethodno registrirao može pristupiti svim funkcionalnostima i pogodnostima aplikacije.

**Za samostalnu registraciju u sustav potrebno je sljedeće:**

- *Ime*
- *Prezime*
- *Adresa elektroničke pošte*
- *Lozinka*

**Za prijavu u sustav potrebno je sljedeće:**

- *Adresa elektroničke pošte*
- *Lozinka*

**Nakon registracije u sustav korisniku je dostupno sljedeće:**

- *Direktno videopraćenje trenutnih događanja u glavnoj konferencijskoj dvorani*
- *Prikaz svih prijavljenih radova*
- *Glasanje za najbolji rad po vlastitom izboru unutar vremenskog okvira*
- *Izabrane fotografije s konferencije koje je moguće preuzeti na vlastiti uređaj*
- *Promotivni materijali pokrovitelja konferencije*
- *Prikaz informacija o mjestu održavanja konferencije i podatci o trenutnim vremenskim uvjetima i vremenskoj prognozi za navedenu lokaciju*
- *Rezultati glasovanja za najbolji poster (nakon završetka glasovanja)*

Sudionici stručne konferencije tijekom određenog vremenskog razdoblja koje je određeno danima i vremenom održavanja konferencije mogu glasovati za rad koji predstavlja pojedinog predavača na konferenciji. Svi radovi prikazani su u web-aplikaciji. Svaki sudionik može glasovati samo jednom odnosno za samo jedan rad za kojeg smatra da je najbolji. Ukoliko se sudionik predomisli te želi dati glas drugom radu to može učiniti ako to odluči dok glasovanje još traje. Po završetku glasovanja objavljaju se rezultati koji su dostupni svim registriranim korisnicima i šalje

se obavijest autorima o rangu njihovog rada prema glasovima sudionika. Porukom elektroničke pošte na dodjelu nagrade poziva se autore prva tri rada s najviše glasova, a sve se ostale sudionike također elektroničkom poštrom obavještava o mjestu i vremenu dodjele nagrade. Nakon završetka glasovanja rezultati se objavljuju na web-aplikaciji u obliku rang tablice.

Aplikacija bi bila od velikog značaja za sve organizatore stručnih konferencija, znanstveno-istraživačkih skupova i raznih drugih javnih događanja jer uvelike olakšava organizaciju i komunikaciju sa sudionicima. Sve informacije sudionici bi mogli dobivati preko aplikacije, a ukoliko ne bi mogli prisustvovati uživo mogli bi se uključiti u direktne videoprijenosne čime bi se omogućio puno veći broj posjetitelja konferencije te bi njen značaj mogao brzo rasti među željenim krugovima ljudi.

Budući da bi aplikaciju koristio vrlo velik broj ljudi te bi ona uključivala videoprijenos uživo potrebno je osigurati stabilnost i brzinu aplikacije kako bi svaki pojedinac mogao u trenutku pristupiti željenom sadržaju i ostao zadovoljan korisničkim iskustvom. Razvojni programeri stoga moraju računati na to da bi moglo biti problema s prekapacitiranošću i unaprijed takve probleme pokušati suzbiti kako ne bi došlo do rušenja aplikacije. Osim toga, aplikacija sadrži bazu s puno osobnih podataka te treba osigurati odgovarajuću zaštitu kako podaci ne bi iscurili pa da se ne naruši privatnost korisnika ili još opasnije - da ne bi bili zloupotabljeni. Zadatak organizatora bio bi osigurati stabilnu mrežu odnosno Wi-Fi na cijelom području održavanja konferencije kako bi se aplikacija mogla nesmetano koristiti.

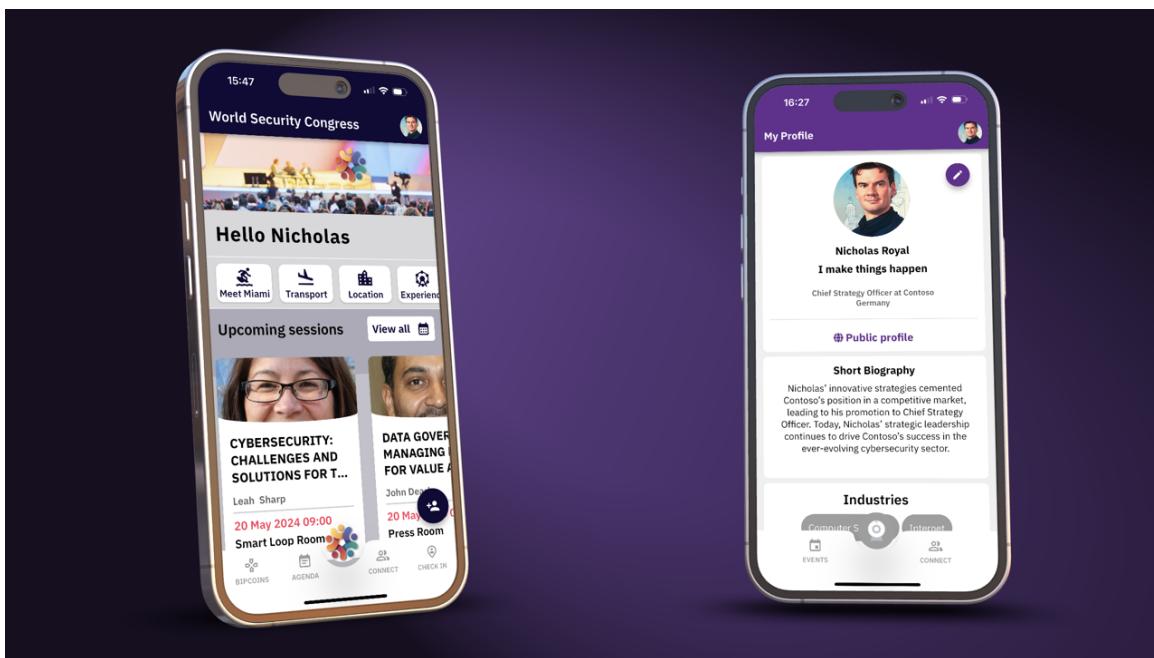
**Aplikacija bi se u budućnosti mogla nadograditi dodavanjem još korisnih funkcionalnosti kao što su na primjer:**

- Karta konferencije - *sudionici bi se mogli lakše snalaziti i pronaći određene sektore i konferencijske dvorane*
- Integracija s društvenim mrežama - *sudionici bi na društvenim mrežama mogli dijeliti informacije o konferenciji i svojim iskustvima čime bi se povećala vidljivost konferencije i time bi se privukli novi sudionici*
- Automatizirane obavijesti - *sustav obavijesti automatski bi podsjećao sudionike o vremenski određenim događajima, poput početka predavanja ili skorog završetka glasovanja*

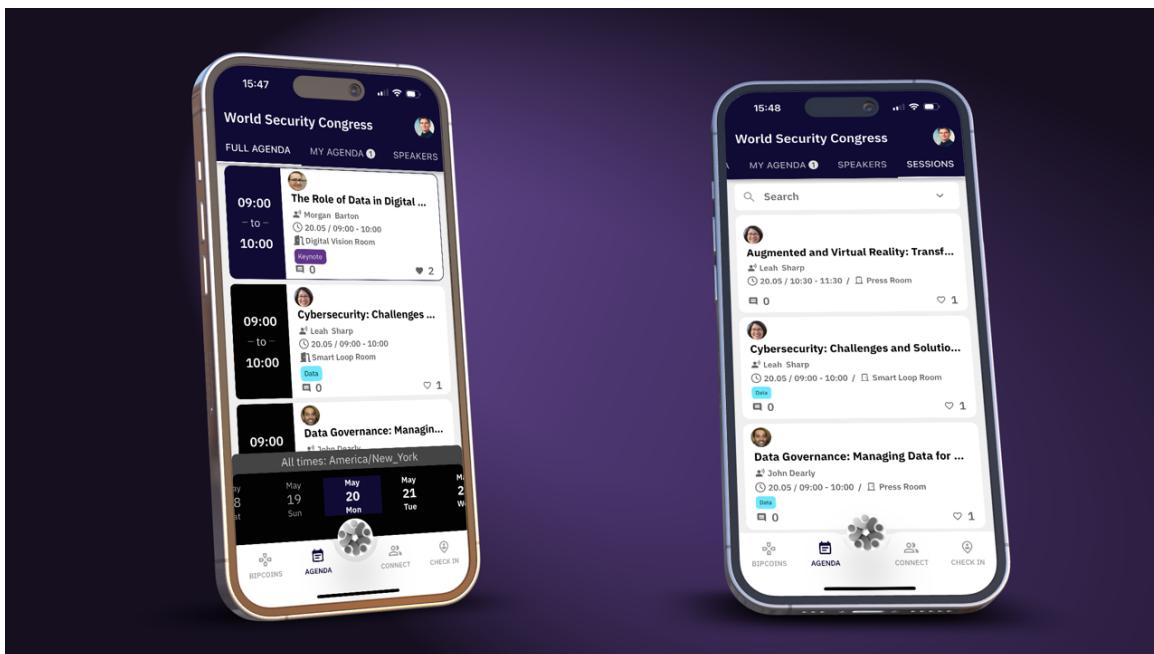
- Sadržaji u blizini pod pokroviteljstvom sponzora - *sudionici bi mogli saznati informacije o lokalnim atrakcijama, znamenitostima i restoranima koje bi mogli posjetiti u slobodno vrijeme, a za koje bi imali popust kao sudionici konferencije*
- Povratna informacija - *sudionici bi mogli dati povratnu informaciju na konferenciju u cijelosti ili određena predavanja čime bi pomogli organizatorima da poboljšaju kvalitetu konferencije u budućnosti, a autorima da unaprijede predavanja.*

Postojeće slično rješenje bila bi *run.events* aplikacija koja za sebe kaže ovako: *Mobilna aplikacija za događanja run.events vaš je vjerni pratitelj na događanju. Posjetitelji na prvi pogled mogu vidjeti što će se sljedeće događati, pregledati sve govornike i pregledavati sesije. Također, mogu koristiti aplikaciju za povezivanje sa sponzorima, izlagačima i drugim sudionicima. Mobilna aplikacija za događanja run.events ističe se modernim izgledom i osjećajem te uključuje značajke komunikacije, orientacije, umrežavanja i gamifikacije, sve prilagođene vašem događaju.* *run-events* nudi mogućnost kupnje ulaznica pa čak i integraciju u web-stranice svojih korisnika putem koda koji je dostupan na njihovim stranicama. Osim mogućnosti korištenja aplikacije na webu, potiče se instalacija njihove mobilne aplikacije u kojoj se nalazi popis nadolazećih događanja i opcija *connect* u kojoj sudionici imaju mogućnost brzog povezivanja s drugim sudionicima s kojima onda mogu razgovarati ili se natjecati u skupljanju novčića. Nakon odabira događaja kojem korisnik želi prisustvovati nudi se kupnja ulaznica ukoliko je to potrebno, a nakon prijave na početnom ekranu mobilne aplikacije mogu se vidjeti nadolazeće sesije i događanja.

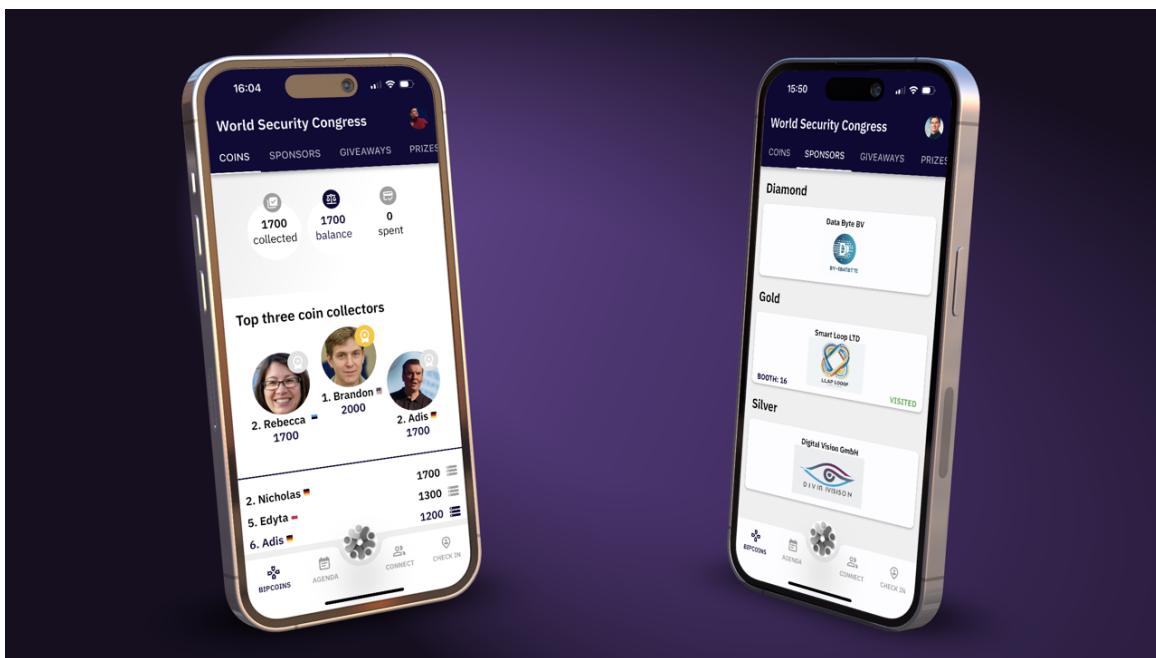
Sličnost s našom aplikacijom jest prvotno u cilju - olakšati organizatorima organizaciju i sudionicima sudjelovanje na konferenciji. Na aplikaciji *run.events* sudionici mogu vidjeti koje je predavanje trenutno u tijeku, a u našoj se aplikaciji može pratiti uživo. U obje aplikacije sudionicima su dostupni promotivni sadržaji. *run.events* nudi više dodatnih mogućnosti korisnicima koje uistinu pružaju izvrsno korisničko iskustvo te bi se u budućnosti i u našu aplikaciju moglo implementirati slične. U nastavku slijede fotografije izgleda mobilne aplikacije *run.events* i snimke zaslona jedne konferencije koja je njihov kod implementirala u svoju stranicu.



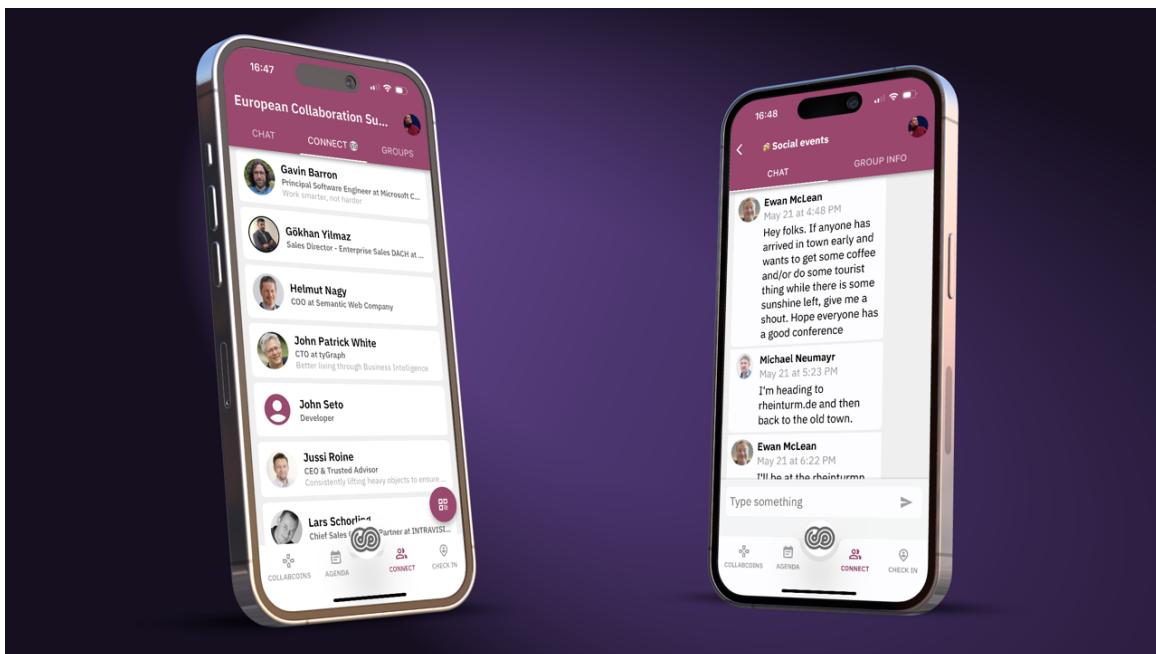
Slika 2.1: Primjer početne stranice i stranice profila



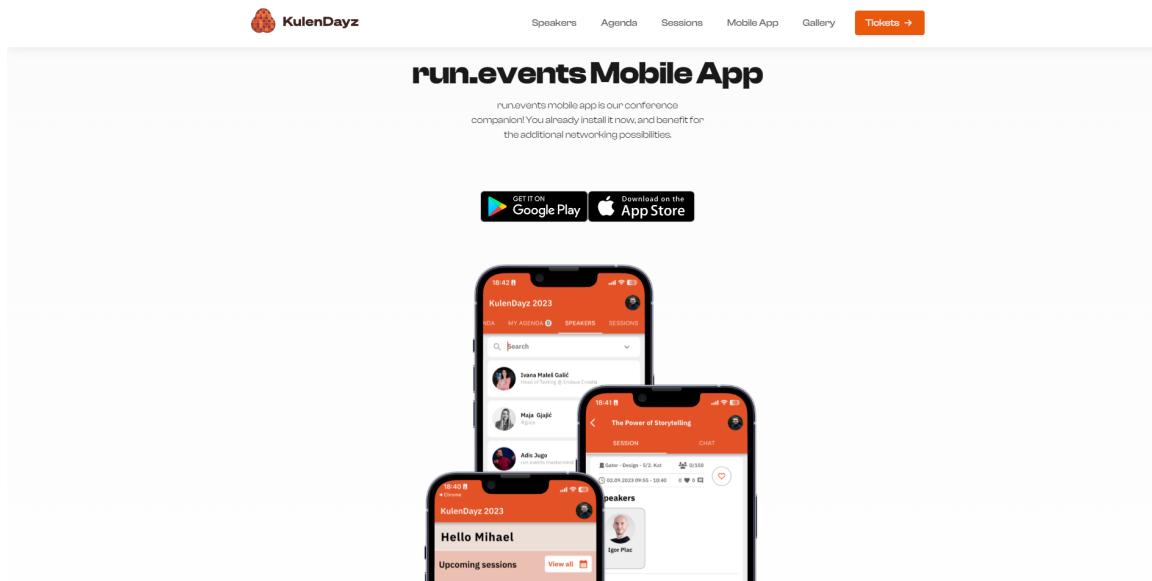
Slika 2.2: Primjer stranice za agendu i sesije



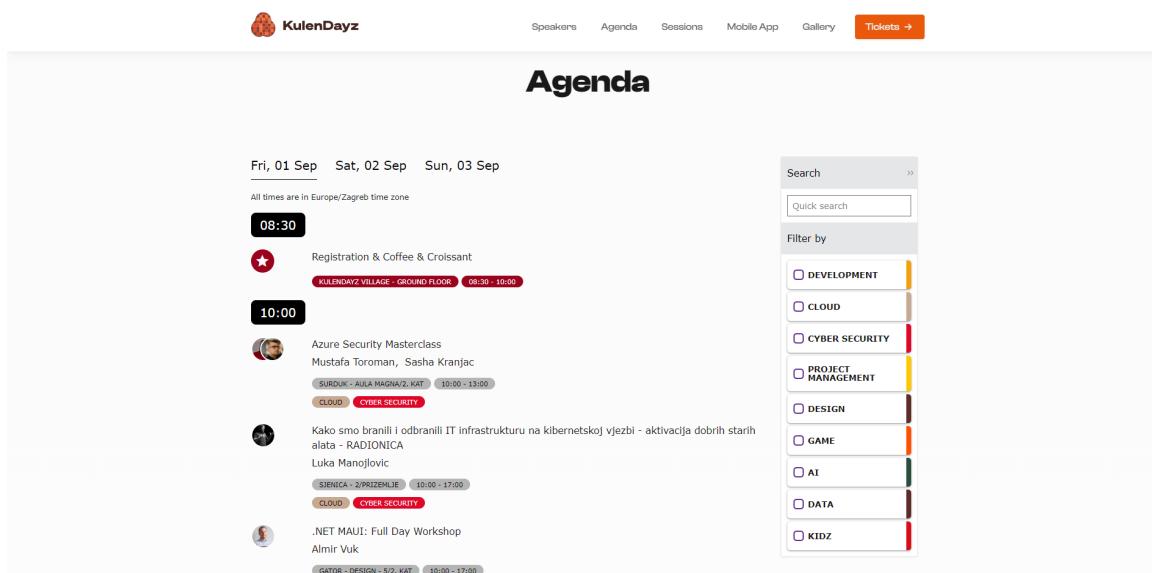
Slika 2.3: Primjer stranice za gamifikaciju i stranice sponzora



Slika 2.4: Primjer stranice za povezivanje i razgovor



Slika 2.5: Primjer stranice za instalaciju mobilne aplikacije *run.events* u sklopu konferencije *KulenDayz*



Slika 2.6: Primjer stranice za agendu implementirane iz *run.events* koda

The screenshot shows the 'Sessions' section of the digital poster. It features a grid of session cards. Each card includes a speaker's profile picture, name, title, and category tags. A sidebar on the right contains a search bar, a 'Filter by' section with various categories, and a 'Tickets' button.

Session Title	Speaker	Category
Bernardin Katić - Artificial Intelligence - Real Life Scenarios	Bernardin Katić	DEVELOPMENT, AI
Adis Jugo - Using Azure Logic Apps for automation of all approval and back-office processes ...	Adis Jugo	DEVELOPMENT, CLOUD
Albert Gajšak - Keynote	Albert Gajšak	
Damir Matešić - Great SQL functions for developers	Damir Matešić	DATA
Mustafa Toroman, Sasha Kranjac - Azure Security Masterclass	Mustafa Toroman, Sasha Kranjac	CLOUD, CYBER SECURITY
Mustafa Toroman, Sasha Kranjac - Cloud Adoption Framework for Azure	Mustafa Toroman, Sasha Kranjac	CLOUD

Slika 2.7: Primjer stranice za sesije implementirane iz *run.events* koda

The screenshot shows the 'Speakers' section of the digital poster. It features a grid of speaker cards. Each card includes a speaker's profile picture, name, title, and a brief description. A sidebar on the right contains a search bar, a 'Filter by' section, and a 'Tickets' button.

Speaker Name	Title	Description
Ivana Maleš Galic	Head of Testing @ Endava Croatia	
Maja Gajšić	#gajšić	
Adis Jugo	run.events mastermind	
Albert Gajšak		
Bernardin Katić	Make code not war	
Damir Dizdarevic		
Damir Matešić	Microsoft Data Platform MVP   Senior Database Developer @Span   SQL Server UG Cr...	
Dejan Sarka	Data platform MVP and MCT veteran	
Luka Manojlović	Moderate tech progress enthusiast	

Slika 2.8: Primjer stranice za govornike implementirane iz *run.events* koda

# 3. Specifikacija programske potpore

## 3.1 Funkcionalni zahtjevi

### Dionici:

#### 1. Primarni dionici:

- (a) Naručitelj
- (b) Razvojni tim
- (c) Rukovoditelji razvoja
- (d) Strateški klijenti
- (e) Superadministrator
- (f) Administrator

#### 2. Sekundarni dionici:

- (a) Sudionici konferencije
- (b) Zaposlenici konferencije
- (c) Autori

### Akteri i njihovi funkcionalni zahtjevi:

#### 1. Neregistrirani posjetitelji konferencije

Neregistrirani posjetitelji konferencije imaju mogućnost pregledavanja stručnih postera svih prijavljenih sudionika pomoću lozinke dobivene na konferenciji i imaju mogućnost dodatne registracije u sustav po želji.

#### 2. Registrirani posjetitelji konferencije

Registrirani posjetitelji konferencije imaju dodatnu mogućnost glasanja za stručni poster za koji smatraju da je najbolji i pristup rezultatima glasovanja kako bi vidjeli koji su posteri osvojili najviše glasova. Također mogu pratiti trenutna događanja u glavnoj konferencijskoj dvorani putem direktnog videoprijenosu. Nadalje, imaju

priliku vidjeti promotivne materijale pokrovitelja konferencije te podatke o trenutnim vremenskim uvjetima i vremenskoj prognozi za navedenu lokaciju. Za vrijeme trajanja konferencije registrirani posjetitelji moći će pregledavati i preuzimati objavljene fotografije s konferencije.

### 3. Superadministrator

Superadministrator ima mogućnost davanja administratorskih uloga već postojećim računima.

### 4. Administrator

Administrator ima mogućnost kreiranja nove konferencije. Prilikom kreiranja konferencije, administrator može odabrati njezin naziv, opis konferencije, mjesto održavanja, njezin početak kao i njezin kraj. Inicijalnu lozinku konferencije sustav automatski generira.

### 5. Autori

Autori su glavni sudionici konferencije koji će prezentirati svoje radove. Oni prije početka konferencije elektroničkom poštom dostavljaju sve potrebne materijale (postere) administratoru. Također, nakon završetka konferencije elektroničkom poštom dobivaju informaciju o rangu njihovih radova kao i pozivnicu za dodjelu nagrada.

### 6. Pokrovitelji

Pokrovitelji financiraju samu konferenciju i dostavljaju promotivne materijale administratoru koji će omogućiti njihov prikaz registriranim posjetiteljima.

### 7. Aplikacija i baza podataka

Pasivni faktori same funkcionalnosti aplikacije kao i spremanja podataka.

### 3.1.1 Obrasci uporabe

#### UC1 - Prijava u sustav odgovarajućom lozinkom

- **Glavni sudionik:** Svi posjetitelji konferencije
- **Cilj:** Prijava u sustav
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Odgovarajuća lozinka dostupna svim posjetiteljima konferencije
- **Opis osnovnog tijeka:**
  1. Posjetitelj upisuje lozinku dobivenu na konferenciji
  2. Sustav provjerava odgovara li kriptografski sažetak lozinke upisane od strane korisnika onoj u bazi podataka
  3. Ukoliko je lozinka točna, posjetitelj se uspješno prijavljuje u aplikaciju
  4. Prijavljenom korisniku otvara se nova stranica
- **Opis mogućih odstupanja:**
  - 1.a Posjetitelj je upisao pogrešnu lozinku
    1. Posjetitelj dobiva obavijest od strane aplikacije kako je njegova lozinka neispravna
    2. Od posjetitelja se traži da opet upiše lozinku

#### UC2 - Odjava prijavljenog korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjava prijavljenog korisnika
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Posjetitelj se uspješno prijavio u sustav odgovarajućom lozinkom
  2. Korisnik je pritisnuo gumb za odjavu prijavljenih korisnika iz sustava
  3. Sustav ga uspješno odjavljuje i vraća na početnu stranicu
- **Opis mogućih odstupanja:**
  - 1.a Korisnik se nije uspješno prijavio u sustav
    1. Od strane korisnika zahtijeva se da se prijavi u sustav odgovarajućom lozinkom

### UC3 - Pregledavanje stručnih postera prijavljenih korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledavanje stručnih postera
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Aplikacija dohvaća potrebne materijale i postere iz baze podataka
  2. Korisnik ima mogućnost pregledavanja stručnih postera

### UC4 - Registracija u sustav prijavljenih korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Registracija u sustav već prijavljenih korisnika
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik je pritisnuo gumb za registraciju u sustav
  2. Korisniku se otvara nova stranica s poljem za upis adrese elektroničke pošte i nove lozinke
  3. Sustav provjerava valjanost upisane adrese elektroničke pošte kao i lozinke
  4. Adresa elektroničke pošte registriranog korisnika sprema se u bazu podataka dok se upisana lozinka kriptira te se njezin sažetak sprema u bazu
- **Opis mogućih odstupanja:**
  - 3.a Pogrešan unos elektroničke pošte
    1. Sustav provjerava ispravnost formata unesene adrese elektroničke pošte te ukoliko je kriva, obavještava korisnika o traženom formatu
    2. Korisnik upisuje ispravan format adrese elektroničke pošte
  - 3.b Pogrešan unos lozinke
    1. Sustav provjerava ispravnost formata unesene lozinke te ukoliko je kriva, obavještava korisnika o traženom formatu
    2. Korisnik upisuje ispravan format lozinke
  - 4.a Korisnik je već prijavljen u sustav
    1. Sustav provjerava postoji li već registriran korisnik s istom adresom elektroničke pošte

2. Sustav odbija registraciju korisnika te ga obavještava kako je korisnik već registriran

### UC5 - Odjava registriranog korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjava iz sustava registriranog korisnika
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Registrirani korisnik prijavljen je u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik je pritisnuo gumb za odjavu registriranih korisnika iz sustava
  2. Sustav ga uspješno odjavljuje i vraća na početnu stranicu prijavljenih korisnika

### UC6 - Glasanje za najbolji stručni poster konferencije

- **Glavni sudionik:** Korisnik
- **Cilj:** Glasanje za najbolji stručni poster konferencije
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Glasovanje je moguće samo registriranim korisnicima tijekom određenog vremenskog razdoblja koje je određeno danima i vremenom održavanja konferencije
- **Opis osnovnog tijeka:**
  1. Registrirani korisnik pregledava objavljene stručne postere
  2. Nakon što registrirani korisnik odabere njemu najbolji poster, ima mogućnost glasanja za taj poster
  3. Registrirani korisnik pritišće gumb za glasanje za određeni poster
  4. Aplikacija spremi u bazu podataka taj glas i označava da je taj korisnik glasao i da više nema mogućnost glasanja
- **Opis mogućih odstupanja:**
  - 3.a Registrirani korisnik pritišće gumb za glasanje za određeni poster van vremenskog razdoblja predviđenog za glasanje
    1. Registrirani korisnik pritišće gumb za glasanje za određeni poster
    2. Sustav ga obavještava kako ne može glasati budući da glasovanje još nije započeto te mu javlja točno vrijeme početka glasanja ili da je glasovanje završilo

3.b Registrirani korisnik pritišće gumb za glasanje za drugi poster

1. Registrirani korisnik pritišće gumb za glasanje za poster nakon što je već jednom glasao za neki drugi poster
2. Sustav ga obavještava kako ne može glasati budući da je već jednom glasao

#### **UC7 - Povlačenje korisnikovog glasa stručnog postera**

- **Glavni sudionik:** Korisnik
- **Cilj:** Povlačenje korisnikovog glasa za stručni poster
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Povlačenje glasa moguće je samo registriranim korisnicima koji su već glasali tijekom određenog vremenskog razdoblja koje je određeno danima i vremenom održavanja konferencije
- **Opis osnovnog tijeka:**
  1. Korisnik je odlučio promijeniti svoje mišljenje te želi povući svoj glas za već odabrani stručni poster
  2. Registrirani korisnik pritišće gumb za povlačenje glasa za određeni poster
  3. Aplikacija briše iz baze podataka taj glas i označava da taj korisnik može ponovno glasati

#### **UC8 - Gledanje direktnog videoprijenos konferencije**

- **Glavni sudionik:** Korisnik
- **Cilj:** Gledanje direktnog videoprijenos konferencije
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Gledanje direktnog videoprijenos konferencije moguće je samo registriranim korisnicima tijekom određenog vremenskog razdoblja koje je određeno danima i vremenom održavanja konferencije
- **Opis osnovnog tijeka:**
  1. Registrirani korisnik pritišće gumb za direktno praćenje videoprijenos konferencije
  2. Aplikacija dohvaća iz baze podataka potreban videoprijenos te se korisniku prikazuje videoprozor

- **Opis mogućih odstupanja:**

- 1.a Registrirani korisnik pritišće gumb za direktno praćenje videoprijenos konferencije van vremenskog razdoblja predviđenog za direktni videoprijenos konferencije
  1. Registrirani korisnik pritišće gumb za direktno praćenje videoprijenos konferencije
  2. Sustav ga obavještava kako ne može gledati direktan videoprijenos konferencije budući da on još nije započeo te mu javlja točno vrijeme početka konferencije

### **UC9 - Zatvaranje direktnog videoprijenos konferencije**

- **Glavni sudionik:** Korisnik

- **Cilj:** Zatvaranje direktnog videoprijenos konferencije

- **Sudionici:** Aplikacija i baza podataka

- **Preduvjet:** Zatvaranje direktnog videoprijenos konferencije moguće je samo registriranim korisnicima koji gledaju direktan videoprijenos konferencije tijekom određenog vremenskog razdoblja koje je određeno danima i vremenom održavanja konferencije

- **Opis osnovnog tijeka:**

1. Registrirani korisnik pritišće gumb za zatvaranje direktnog videoprijenos konferencije
2. Aplikacija zatvara videoprozor

### **UC10 - Pregled poretku stručnih postera**

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregled poretku stručnih postera

- **Sudionici:** Aplikacija i baza podataka

- **Preduvjet:** Pregled poretku stručnih postera dostupan je svim registriranim korisnicima po završetku konferencije i glasanja

- **Opis osnovnog tijeka:**

1. Konferencija je kao i njeno glasanje završila
2. Aplikacija iz baze podataka dohvata glasove za stručne postere te prikazuje poretek najboljih stručnih postera svim registriranim korisnicima

- **Opis mogućih odstupanja:**

2.a Poredak nije dostupan

1. Prostor za poredak stručnih postera jest prazan
2. Glasanje kao i sama konferencija nisu još završeni te sustav obaveštava korisnika o vremenu objave poretku

### **UC11 - Gledanje promotivnih materijala pokrovitelja**

- **Glavni sudionik:** Korisnik

- **Cilj:** Gledanje promotivnih materijala pokrovitelja

- **Sudionici:** Aplikacija i baza podataka

- **Preduvjet:** Mogućnost gledanja promotivnih materijala pokrovitelja imaju registrirani korisnici

- **Opis osnovnog tijeka:**

1. Aplikacija iz baze podataka dohvaća promotivne materijale pokrovitelja konferencije
2. Na početnoj stranici dostupni su promotivni materijali pokrovitelja koje registrirani korisnik može gledati

### **UC12 - Prikaz dodatnih informacija konferencije kao što su vremenski uvjeti i lokacija**

- **Glavni sudionik:** Korisnik

- **Cilj:** Prikaz dodatnih informacija konferencije kao što su vremenski uvjeti i lokacija

- **Sudionici:** Aplikacija i baza podataka

- **Preduvjet:** Korisnik mora biti registriran u sustav

- **Opis osnovnog tijeka:**

1. Aplikacija iz baze podataka dohvaća dodatne podatke o konferenciji kao što su vremenski uvjeti i lokacija
2. Na početnoj stranici dostupne su dodatne informacije o konferenciji kao što su vremenski uvjeti i lokacija koje registrirani korisnik može gledati

### **UC13 - Prikaz objavljenih fotografija s konferencije**

- **Glavni sudionik:** Korisnik

- **Cilj:** Prikaz objavljenih fotografija

- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik mora biti registriran u sustav te su fotografije vidljive samo za vrijeme trajanja konferencije
- **Opis osnovnog tijeka:**
  1. Aplikacija iz baze podataka dohvata odabrane fotografije s konferencije
  2. Na početnoj stranici dostupne su odabrane fotografije s konferencije
- **Opis mogućih odstupanja:**
  - 2.a Korisnik ne vidi fotografije
    1. Korisnik na početnoj stranici ne vidi fotografije s konferencije
    2. Sustav obaveštava korisnika kako fotografije s konferencije nisu dostupne budući da je konferencija završena

#### UC14 - Preuzimanje objavljenih fotografija s konferencije

- **Glavni sudionik:** Korisnik
- **Cilj:** Preuzimanje objavljenih fotografija s konferencije
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik mora biti registriran u sustav te su fotografije dostupne za preuzimanje samo za vrijeme trajanja konferencije
- **Opis osnovnog tijeka:**
  1. Aplikacija iz baze podataka dohvata odabrane fotografije s konferencije
  2. Korisnik pritiše gumb za preuzimanje fotografije na svoj uređaj

#### UC15 - Dodavanje autora i njihovih stručnih postera u bazu podataka

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje autora i njihovih stručnih postera u bazu podataka
- **Sudionici:** Baza podataka i aplikacija
- **Preduvjet:** Administratoru su dostavljeni svi potrebni materijali za dodavanje autora kao i njihovih stručnih postera
- **Opis osnovnog tijeka:**
  1. Autor elektroničkom poštom šalje administratoru svoje podatke i stručni poster
  2. Administrator provjera valjanost podataka
  3. Ukoliko su svi podaci ispravni i valjani, administrator otvara sučelje za dodavanje autora i njegovog postera gdje upisuje sve potrebne informacije vezane za autora i poster te prilaže sam poster

4. Nakon pritiska na gumb za spremanje, baza podataka spremi informacije o posteru kao i o njegovom autoru

#### **UC16 - Definiranje svih potrebnih parametara za rad sustava**

- **Glavni sudionik:** Administrator
- **Cilj:** Definiranje svih potrebnih parametara za rad sustava kao što su vrijeme početka i završetka konferencije, otvaranje i zatvaranje glasanja...
- **Sudionici:** Baza podataka
- **Preduvjet:** Administratoru su dostupne sve potrebne informacije o konferenciji
- **Opis osnovnog tijeka:**
  1. Administrator dobiva sve potrebne informacije o konferenciji
  2. Dobivene informacije administrator pohranjuje u bazu podataka

#### **UC17 - Obavještavanje autora o njihovom uspjehu i mjestu održavanja dodjele nagrada**

- **Glavni sudionik:** Aplikacija
- **Cilj:** Obavještavanje autora o njihovom uspjehu i mjestu održavanja dodjele nagrada
- **Sudionici:** Autori
- **Preduvjet:** Glasanje je kao i sama konferencija završeno
- **Opis osnovnog tijeka:**
  1. Aplikacija iz baze podataka dohvata podatke o poretku radova
  2. Autori prva tri najbolja rada dobivaju posebnu čestitku i pozivnicu za dodjelu nagrade dok ostali autori dobivaju samo pozivnicu za dolazak na dodjelu nagrada

#### **UC18 - Dodavanje administratorske uloge postojećim registriranim korisnicima**

- **Glavni sudionik:** Superadministrator
- **Cilj:** Dodavanje administratorske uloge postojećim registriranim korisnicima
- **Sudionici:** Registrirani korisnici, baza podataka
- **Preduvjet:** Korisnik se mora registrirati

- **Opis osnovnog tijeka:**
  1. Nakon prijavljivanja u sustav superadministratoru se otvara lista svih već postojećih administratora
  2. Superadministrator po želji može kreirati novog administratora
- **Opis mogućih odstupanja:**
  - 1.a Superadministrator ne vidi niti jednog administratora
    1. U bazi podataka nema niti jednog administratora
    2. Superadministrator mora kreirati barem jednog administratora da bi mu se pojavila lista

### UC19 - Kreiranje konferencije

- **Glavni sudionik:** Administrator
- **Cilj:** Kreiranje konferencije
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik mora imati ulogu administratora
- **Opis osnovnog tijeka:**
  1. Nakon prijave u sustav administrator ima mogućnost kreiranja nove konferencije
  2. Administratoru se otvara sučelje za kreiranje nove konferencije s praznim poljima za naziv konferencije, opis konferencije, mjesto održavanja, datum početka kao i datum završetka konferencije koje administrator može popuniti po želji
  3. Nakon popunjavanja sustav će automatski generirati inicijalnu lozinku konferencije te će se čitava konferencija spremiti u bazu podataka

### UC20 - Dodavanje fotografija za konferenciju

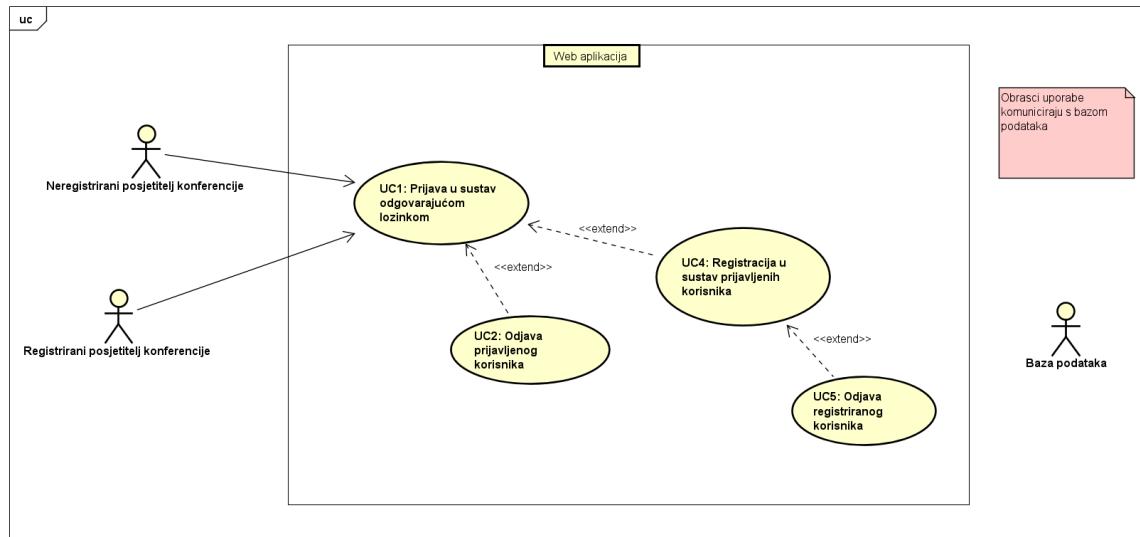
- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje fotografija za konferenciju
- **Sudionici:** Aplikacija i baza podataka
- **Preduvjet:** Korisnik mora imati ulogu administratora
- **Opis osnovnog tijeka:**
  1. Nakon prijave u sustav administrator ima mogućnost dodavanja fotografija u bazu podataka

2. Administrator otvara sučelje za dodavanje novih fotografija gdje upisuje potrebne informacije vezane za fotografije te prilaže fotografiju
  3. Nakon pritiska na gumb za spremanje, baza podatka spremi informacije o fotografiji
- **Opis mogućih odstupanja:**
    - 2.a Unos krivih podataka
      1. Administrator je unio neke krive podatke
      2. Sustav ga upozorava na krive podatke te mu javlja kako da ispravi pogrešku

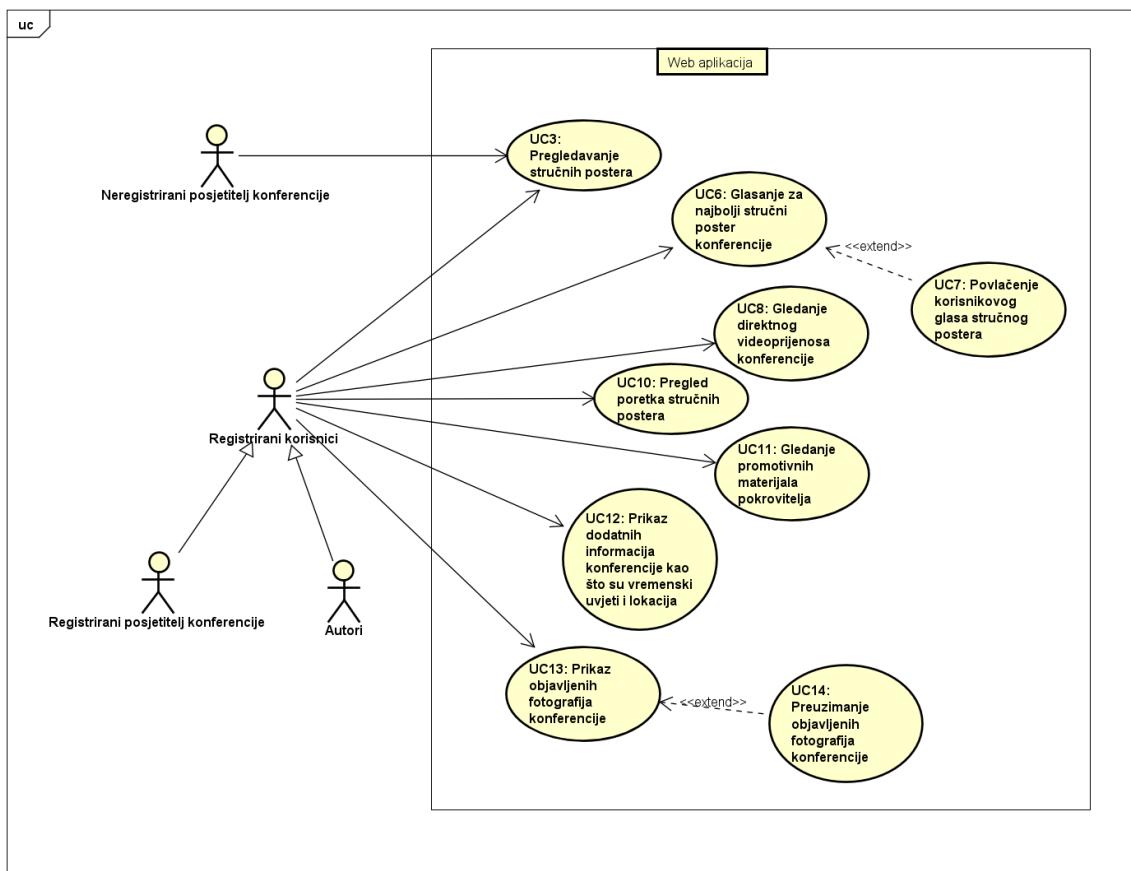
### UC21 - Dodavanje promotivnih materijala za konferenciju

- **Glavni sudionik:** Administrator
  - **Cilj:** Dodavanje promotivnih materijala za konferenciju
  - **Sudionici:** Aplikacija i baza podataka
  - **Preduvjet:** Korisnik mora imati ulogu administratora
  - **Opis osnovnog tijeka:**
    1. Nakon prijave u sustav administrator ima mogućnost dodavanja promotivnih materijala u bazu podataka
    2. Administrator otvara sučelje za dodavanje novih promotivnih materijala gdje upisuje potrebne informacije vezane za promotivne materijale te prilaže promotivni materijal
    3. Nakon pritiska na gumb za spremanje, baza podatka spremi informacije o promotivnom materijalu
- **Opis mogućih odstupanja:**
    - 2.a Unos krivih podataka
      1. Administrator je unio neke krive podatke
      2. Sustav ga upozorava na krive podatke te mu javlja kako da ispravi pogrešku

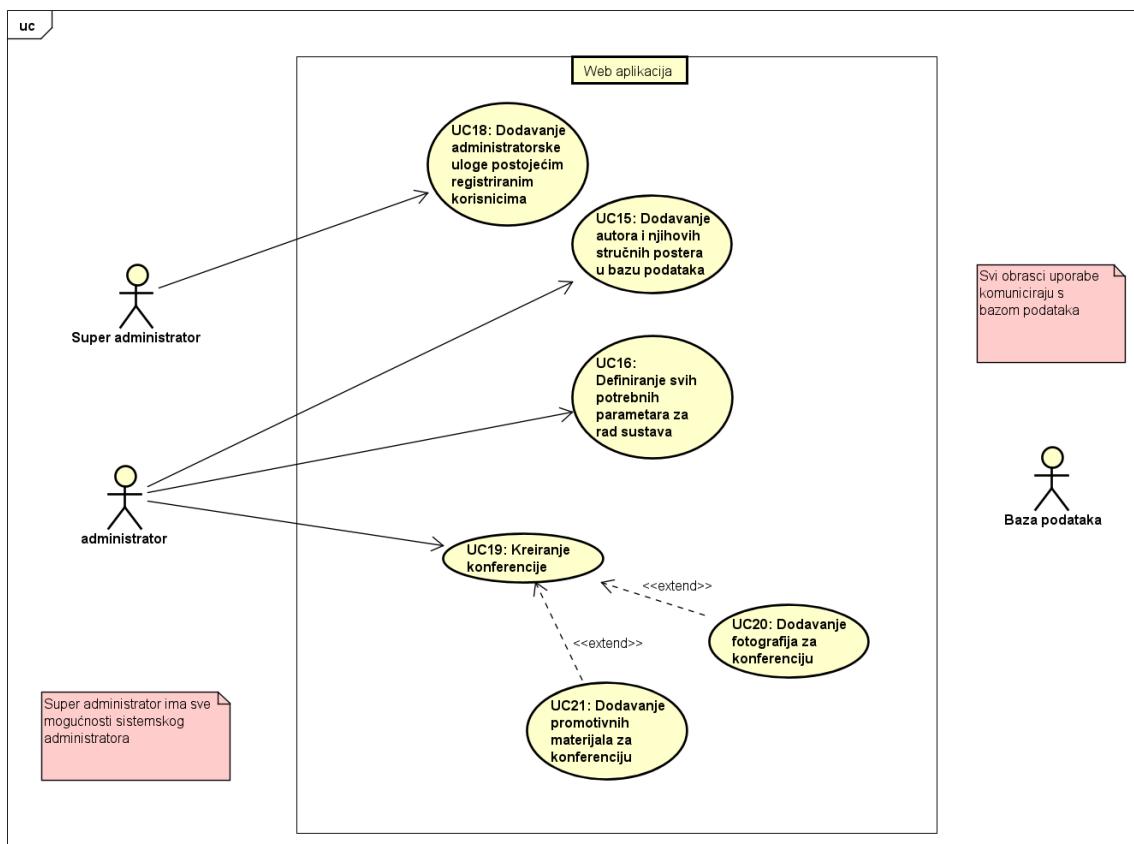
## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, prijava u sustav



Slika 3.2: Dijagram obrazaca uporabe, funkcionalnosti korisnika

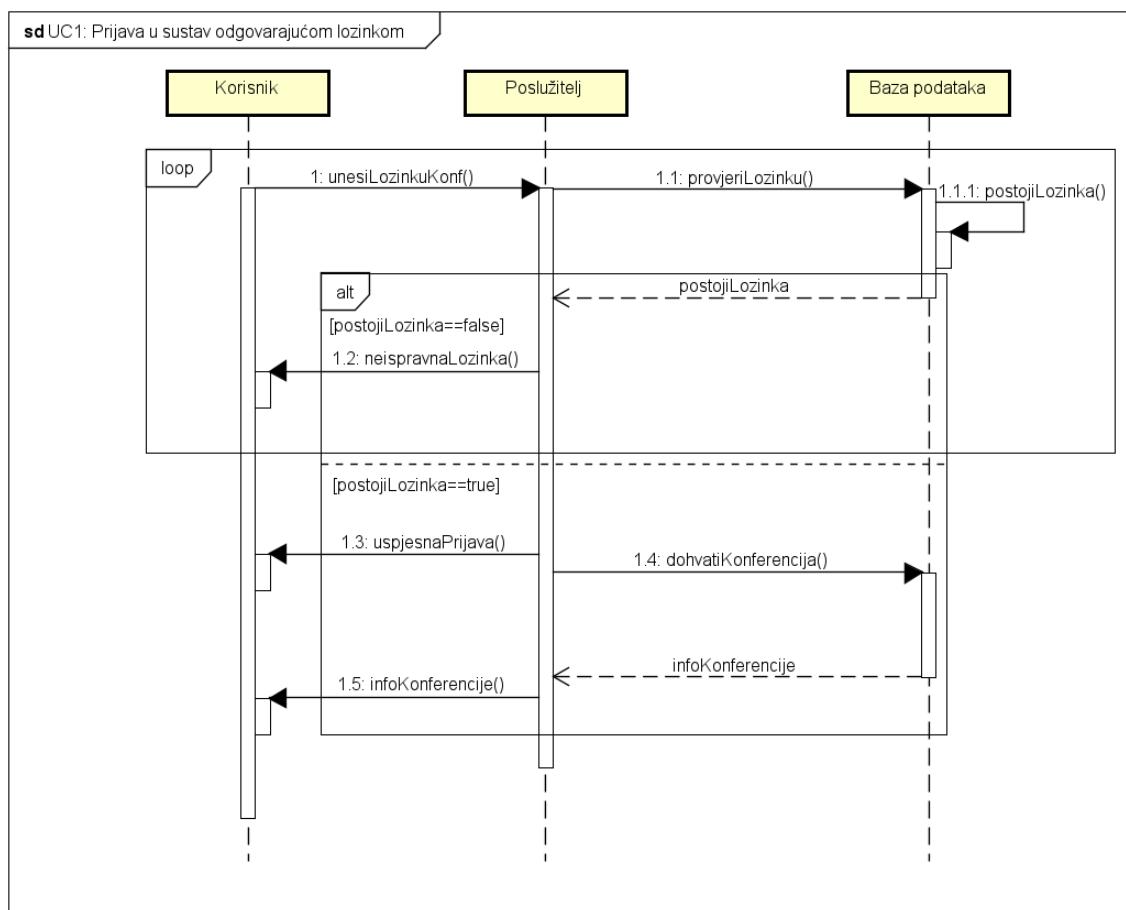


Slika 3.3: Dijagram obrazaca uporabe, funkcionalnosti administratora

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1: Prijava u sustav odgovarajućom lozinkom

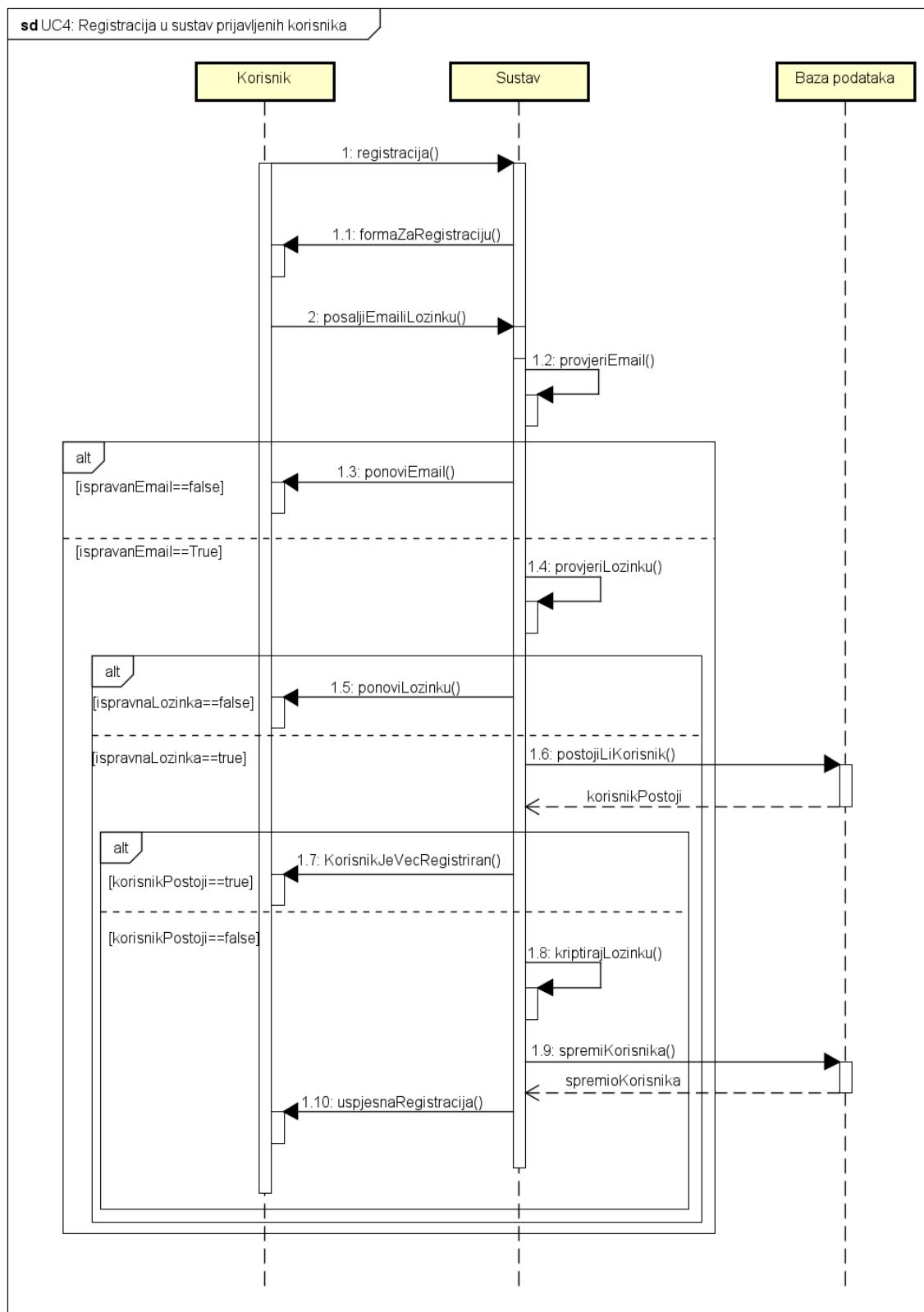
Korisnik na početnom zaslonu upisuje lozinku konferencije te ju šalje poslužitelju. Poslužitelj šalje upit bazi podataka i provjerava je li lozinka ispravna, ako je neispravna ovaj se postupak ponavlja do upisa točne lozinke. Ako je unesena lozinka ispravna, korisniku se potvrđuje da je unio ispravnu lozinku te se od baze podataka zatražuju podaci o konferenciji i oni se šalju korisniku.



Slika 3.4: Sekvencijski dijagram obrasca UC1: Prijava u sustav odgovarajućom lozinkom

**Obrazac uporabe UC4: Registracija u sustav prijavljenih korisnika**

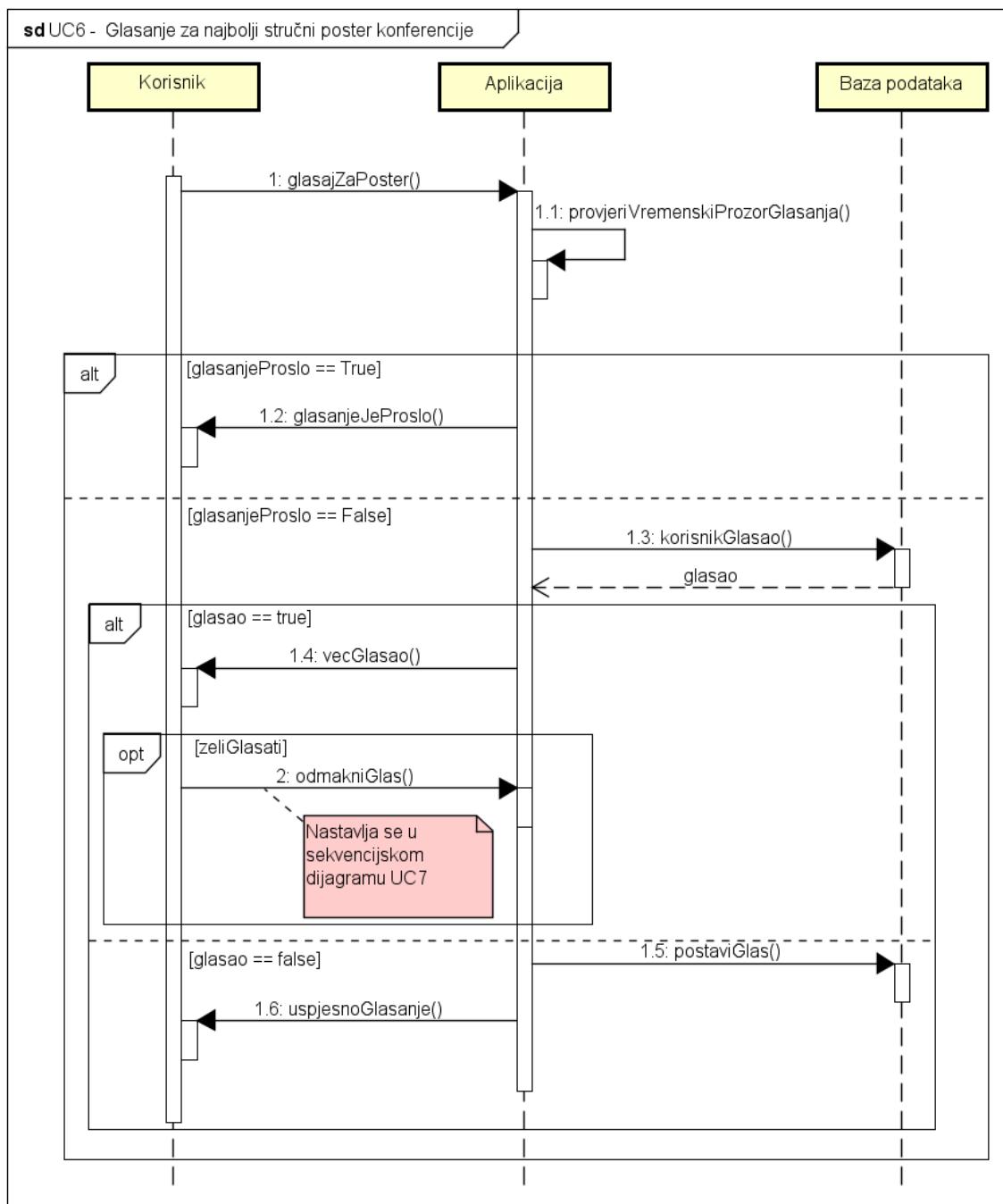
Korisnik klikom na gumb za registraciju pokreće postupak registracije. Sustav mu šalje nazad formu za registraciju u koju upisuje adresu elektroničke pošte i lozinku te ih šalje sustavu. Sustav zatim provjerava format adrese elektroničke pošte pa ako nije valjan obavještava korisnika o tome. Ukoliko je valjan, nastavlja dalje i provjerava ispunjava li lozinka traženi format te u slučaju da ne ispunjava također obavještava korisnika. Na kraju se upitom nad bazom provjerava postoji li registrirani korisnik s tom adresom elektroničke pošte. U slučaju da postoji obavijestit će korisnika o tome, a ako ne, lozinka će se kriptirati i spremiti zajedno s adresom elektroničke pošte u bazu podataka i korisnika će obavijestiti da je registracija bila uspješna.



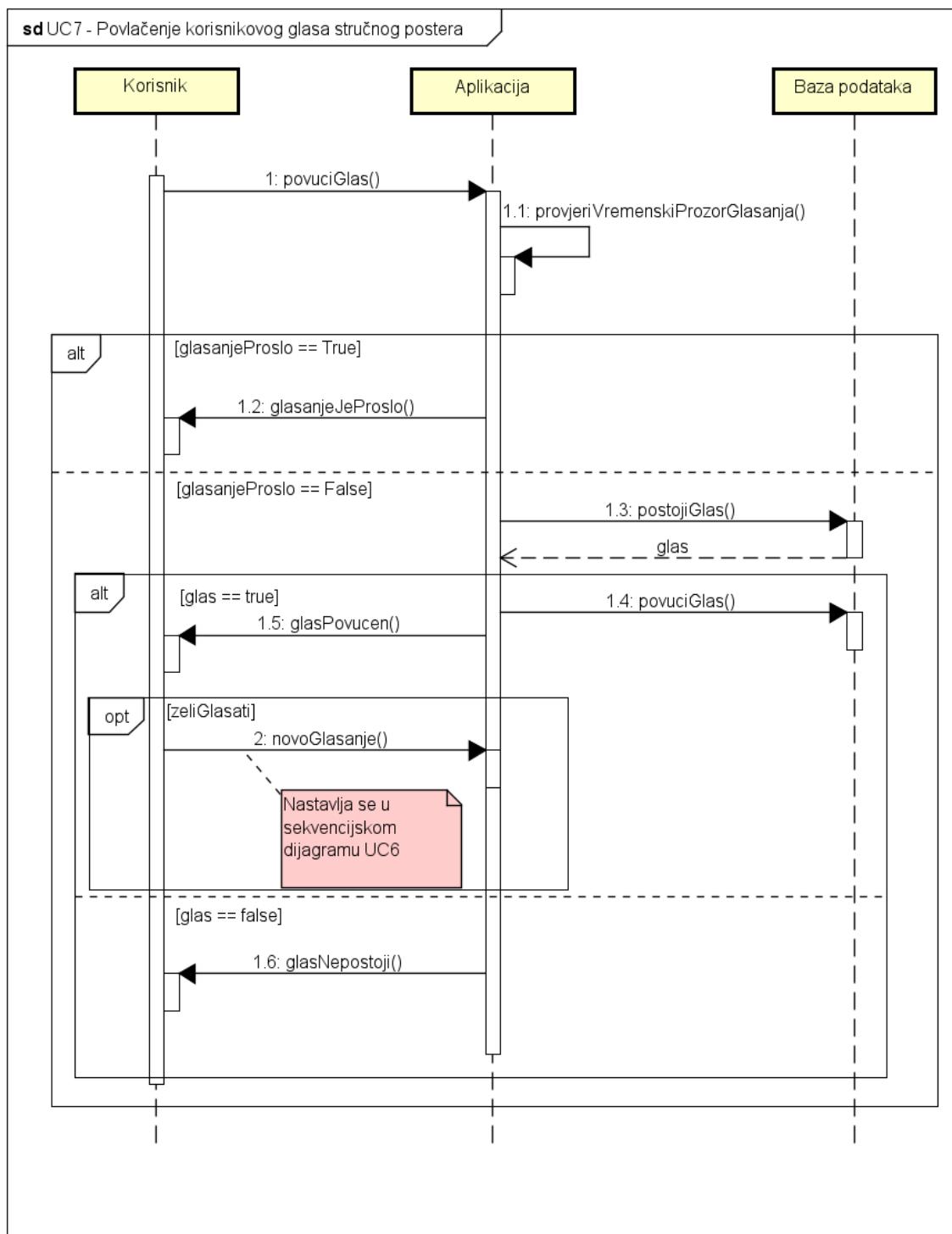
Slika 3.5: Sekvencijski dijagram obrasca UC4: Registracija u sustav prijavljenih korisnika

### Obrazci uporabe UC6 i UC7: Glasanje

Obrasci UC6 i UC7 opisuju rad sustava glasanja. Postupci glasanja i povlačenja glasa započinju zahtjevom korisnika te se za oba prvo provjerava vremenski prozor glasanja (glasanje i promjena glasa onemogućena je ako je vremenski prozor glasanja prošao). Ako je glasanje još uvijek dozvoljeno, sljedeći je korak provjera je li korisnik već glasao pa s obzirom na odgovor baze podataka imamo grananje. Kod UC6, ako je korisnik već glasao, javlja mu se greška i nudi opcija da makne svoj glas. U suprotnom, u bazu podataka sprema se glas te se korisniku dojavljuje da je glasanje provedeno uspješno. Kod UC7, ako je korisnik već glasao, onda se glas uspješno briše iz baze podataka, inače se korisniku javlja greška da nije glasao prije.



Slika 3.6: Sekvencijski dijagram obrasca UC6: Glasanje za najbolji stručni poster konferencije



Slika 3.7: Sekvencijski dijagram obrasca UC7: Povlačenje korisnikovog glasa stručnog postera

### 3.2 Ostali zahtjevi

- Unutar sustava mora biti omogućen istovremeni rad svih korisnika samog sustava
- Aplikacija se mora moći koristiti unutar bilo kojeg web-preglednika
- Aplikacija se mora ostvariti objektno orijentiranim pristupom programiranja
- Aplikacija mora biti jednostavna i intuitivna za korištenje svim korisnicima - svih uzrasta i predznanja
- Aplikacija mora biti javno dostupna
- Unutar aplikacije mora se provoditi autentifikacija te prikazani sadržaj mora odgovarati dozvoljenom sadržaju za trenutnog korisnika
- Nadogradnjom postojeće aplikacije ne smije doći do urušavanja funkcionalnosti i kvalitete već postojeće aplikacije
- Unutar sustava mora biti omogućen unos svih posebnih slova referentnih za odabrani jezik aplikacija (za slučaj hrvatskog jezika, mora biti moguć unos dijakritičkih znakova)

## 4. Arhitektura i dizajn sustava

Arhitektura sustava je hijerarhijska, dakle svaki pojedini sloj komunicira isključivo sa slojevima koji su neposredno ispred i iza njega. Slojevi sustava koje mi implementiramo jesu:

- *Korisničko sučelje*
- *Kontroler*
- *Servis*
- *Repozitorij*
- *Baza podataka*

Korisničko sučelje (eng. User interface, UI) predstavlja interaktivno područje između korisnika i računala. Njegov je glavni cilj omogućiti korisnicima učinkovito korištenje i upravljanje računalom te osigurati da računalo pruži korisniku potrebne informacije.

Za izradu korisničkog sučelja u ovom slučaju korišten je React, JavaScript biblioteka koja omogućava brzo i jednostavno stvaranje interaktivnih korisničkih sučelja. Kroz korisničko sučelje, korisnik šalje zahtjeve kontroleru, a kontroler potrebne podatke prosljeđuje pomoću JSON (JavaScript Object Notation) datoteka.

JSON datoteke služe za pohranu i prijenos podataka u obliku ključ-vrijednost. Nakon što korisničko sučelje preda JSON datoteku, ono očekuje odgovor od kontrolera koji će također biti u JSON formatu. Kontroler u ovom slučaju predstavlja REST API (representational state transfer) te obrađuje zahtjeve vanjskih potrošača.

Servis je odgovoran za obradu podataka koje prima od korisničkog sučelja putem kontrolera i baze podataka putem repozitorija. Osim toga, servis obuhvaća poslovne odluke, autorizaciju te provjeru valjanosti identiteta korisnika.

Repozitorij ima ulogu komunikacije s bazom podataka te uključuje funkcije za pronalaženje određenih objekata ili skupina objekata iz baze podataka. Ove funkcije obično vraćaju popis objekata koji zadovoljavaju određeni uvjet.

Baza podataka koristi se za pohranu i upravljanje podacima te predstavlja ključni dio sustava koji omogućuje trajno čuvanje informacija.

Kontroler, servis i repozitorij implementirani su pomoću Java Spring Boota, te su pisani u jeziku JavaScript.

## 4.1 Baza podataka

Za potrebe sustava koji implementiramo koristit ćemo se relacijskom bazom podataka koja kao svoju glavnu namjenu ima olakšano modeliranje stvarnog svijeta oko nas. Gradivna jedinica baze jest relacija (tablica) koja je definirana svojim imenom te skupom atributa. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- *Konferencija*
- *Mjesto*
- *FotoMaterijal*
- *Korisnik*
- *Uloge*
- *Poster*
- *PromoMaterijal*
- *Glasanje*

### 4.1.1 Opis tablica

Konferencija - centralni entitet koji definira događaj kojemu pristupaju sudionici, bilo autori ili ne. Atributi koje posjeduje su *konfID*, *kod* (za pristup posterima koji su u natjecanju), *datPocetak*, *datKraj*, *nazivKonf*, *mjestoKonf*, *opisKonf*, *aktivna*, *livesream*

Konferencija		
konfID	INT	Jedinstveni identifikator konferencije
kod	INT	Pristupna lozinka za posjetitelje
vrijemePoc	LOCALDATETIME	Vrijeme početka konferencije
vrijemeKraj	LOCALDATETIME	Vrijeme kraja konferencije
nazivKonf	VARCHAR	Naziv konferencije
opisKonf	VARCHAR	Opis konferencije
aktivna	BOOLEAN	Aktivnost konferencije
livestream	VARCHAR	WEB stranica prijenosa uživo
pbr	INT	Poštanski broj mjesta održavanja konferencije

Mjesto - entitet koji definira mjesto u kojemu se održava konferencija. Atributi koje posjeduje su *kucBroj* i *ulica* kao glavni ključ, *nazivMjesta* i *pbr*. Mjesto je u vezi jedan-prema-više s entitetom Konferencija. Odnosno jedna konferencija ima jedno mjesto, dok jedno mjesto ima više konferencija.

Mjesto		
kucBroj	INT	Kućanski broj ulice u kojoj će se održavati konferencija
ulica	VARCHAR	Ulica u kojoj će se održavati konferencija
nazivMjesta	VARCHAR	Naziv mesta gdje se održava konferencija
pbr	INT	Poštanski broj mjesta održavanja konferencije

FotoMaterijal - entitet FotoMaterijal koristi nam kako bismo mogli pohraniti fotografije nastale tijekom konferencije te ih kasnije prikazati korisnicima. Sadrži atrribute: *fotoID*, *nazivFoto*, *fotoPath* te strani ključ *konfID* pomoću kojega možemo upariti kojoj konferenciji pripada koja fotografija (veza jedan-prema-više). Također sadrži i *fotobyte* za pohranu fotografije u bazu kao i *fototype* za sami format fotografije.

<b>FotoMaterijal</b>		
fotoID	INT	Jedinstveni identifikator fotografije
nazivFoto	VARCHAR	Naziv fotografije
fotoPath	VARCHAR	Putanja do izvora fotografije
fototype	VARCHAR	Format fotografije
fotobyte	BYTEA	Pohrana fotografije u bazu
konfID	INT	Jedinstveni identifikator konferencije

Korisnik - entitet kojim se pokriva registrirani korisnik, neregistrirani korisnik te autor. Sadrži atributе *email* te *lozinka* kojima se kasnije može pristupiti posebnom sadržaju specifičnom za registrirane korisnike. Također sadrži *ime* i  *prezime* i *resetPassTok* za obnovu lozinke.

<b>Korisnik</b>		
email	VARCHAR	Elektronička pošta korisnika
lozinka	VARCHAR	Lozinka za prijavu korisnika
ime	VARCHAR	Ime korisnika
prezime	VARCHAR	Prezime korisnika
resetPassTok	VARCHAR	Token za obnovu lozinke

Uloge - entitet kojim se pokrivaju koje uloge sve postoje kao i koje uloge imaju korisnici. Atributi koje sadrži entitet Uloge su: *ulogaID* koji je glavni ključ i *uloga* koja opisuje o kojoj je ulozi riječ.

<b>Uloge</b>		
ulogaID	ID	Jedinstveni identifikator uloge
uloga	VARCHAR	Uloga korisnika

Poster - ovim entitetom kontrolirat će se radovi koje na konferenciju dostavljaju autori te se na njih izlažu i za njih se može glasati. Njegovi atributi su: *posterID*, *nazivPoster*, *posterPath*, *email*, *imeAutor*,  *prezimeAutor* i *konfID* koji je ujedno i strani ključ pomoću kojega možemo upariti kojoj konferenciji pripada koji poster (veza jedan-prema-više). Također imamo i *imageType* kojim opisujemo format postera

kao i *imageByte* kojim spremamo sami poster u bazu.

<b>Poster</b>		
posterID	INT	Jedinstveni identifikator postera
nazivPoster	VARCHAR	Naziv postera
posterPath	VARCHAR	Putanja do izvora postera
email	VARCHAR	Elektronička pošta korisnika
imeAutor	VARCHAR	Ime autora postera
prezimeAutor	VARCHAR	Prezime autora postera
imageByte	BYTEA	Pohrana postera u bazu
imageType	VARCHAR	Format postera
konfID	INT	Jedinstveni identifikator konferencije

PromoMaterijal - entitet PromoMaterijal pokriva konkretan ponuđeni materijal kojemu registrirani korisnik može pristupiti. Atributi koje ima su *promoId*, *nazivPromo*, *url*, *promoPath* i *konfID* koji je ujedno i strani ključ pomoću kojega možemo upariti kojoj konferenciji pripada koji promotivni materijal (veza jedan-prema-više). Također imamo i *promoType* kojim opisujemo format postera kao i *promoByte* kojim spremamo sami poster u bazu.

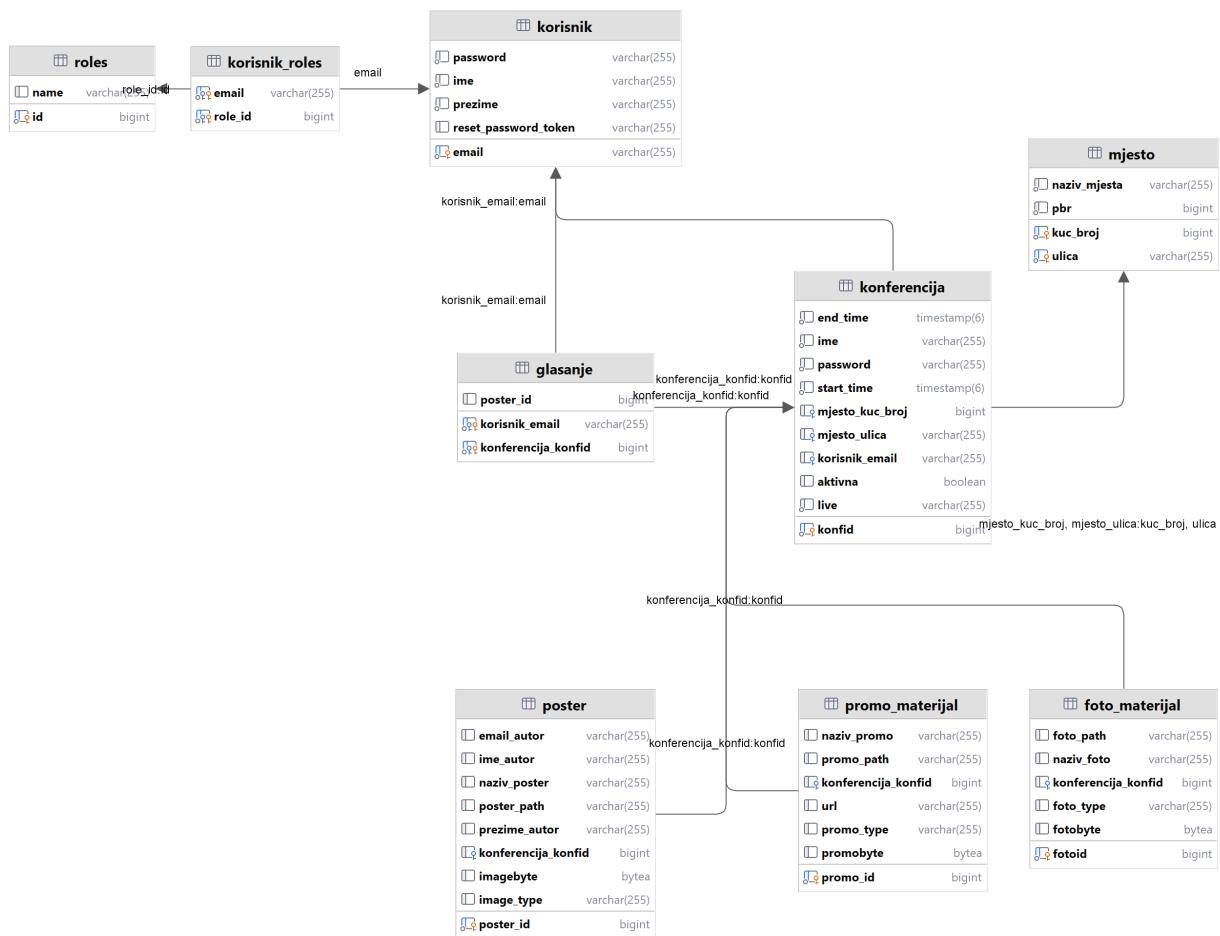
<b>PromoMaterijal</b>		
promoId	INT	Jedinstveni identifikator promotivnog materijala
nazivPromo	VARCHAR	Naziv promotivnog materijala
promoPath	VARCHAR	Putanja do izvora sadržaja promotivnog materijala
url	VARCHAR	WEB stranica pokrovitelja
promoType	VARCHAR	Format promotivnog materijala
promoByte	BYTEA	Pohrana promotivnog materijala u bazu
konfID	INT	Jedinstveni identifikator konferencije

Glasanje - entitet Glasanje služi za prebrojavanje broja glasova postera te također

služi za prevenciju glasovanja jednog registriranog korisnika za više postera. Atributi koji sadrži entitet Glasanje su: *posterID* koji je jedinstveni identifikator postera te je također i primarni ključ, *konfID* koji je jedinstveni identifikator konferencije te je također i strani ključ *email* koji je elektronička pošta korisnika, ali je isto tako i strani ključ. Obje veze od entiteta Glasanje prema entitetima Korisnik i Konferencija su jedan-prema-više.

Glasanje		
posterID	INT	Jedinstveni identifikator postera
konfID	INT	Jedinstveni identifikator konferencije
email	VARCHAR	Elektronička pošta korisnika

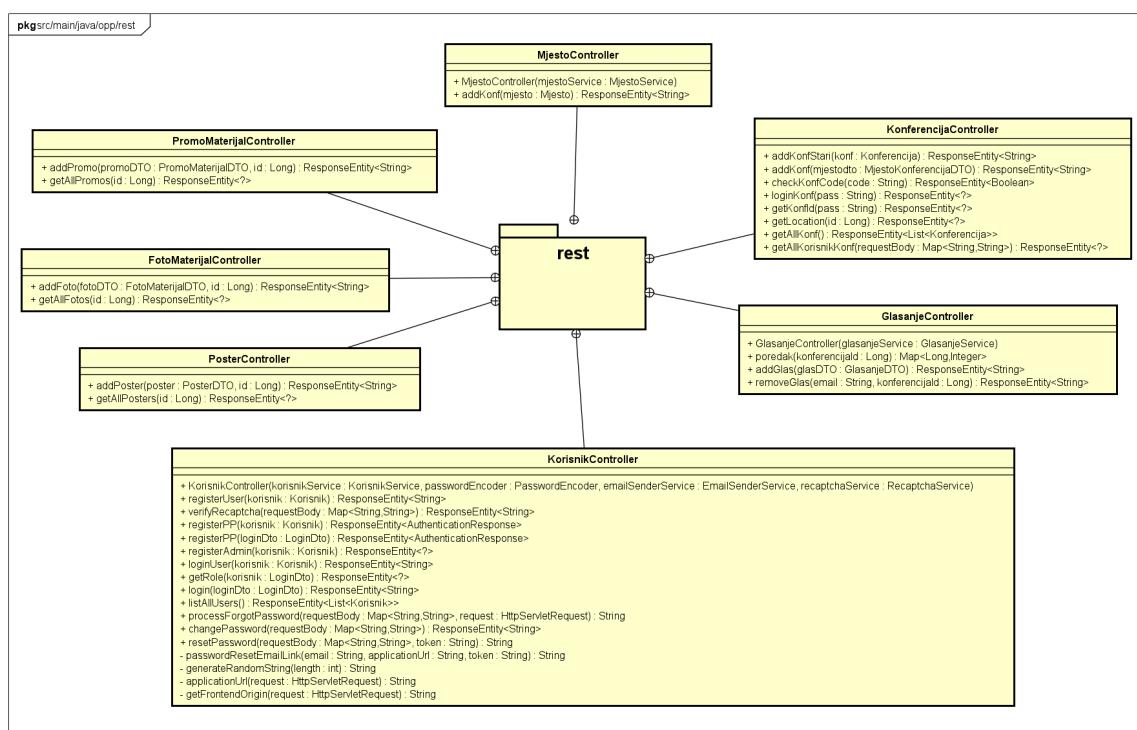
### 4.1.2 Dijagram baze podataka



Slika 4.1: Dijagram baze podataka

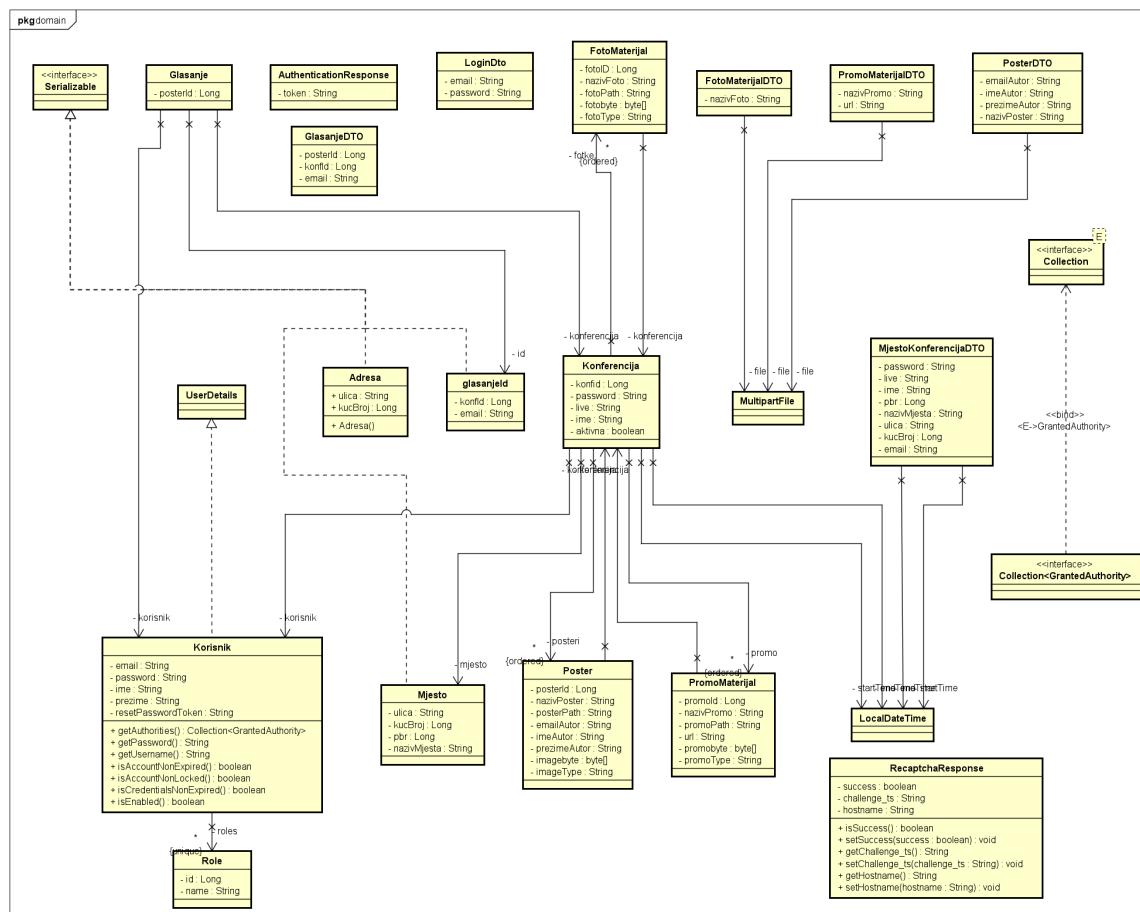
## 4.2 Dijagram razreda

Zbog lakše organizacije projekta, svi kontroler razredi koji su zaduženi za upravljanje krajnjim točkama svrstani su u paket *src/main/java/opp/rest*. Kontroleri igraju ključnu ulogu u upravljanju HTTP zahtjevima te su odgovorni za primanje zahtjeva od klijenta, obradu tih zahtjeva i generiranje odgovora. U našoj implementaciji imamo više kontrolera kako bi lakše organizirali i razdvojili sve dostupne krajnje točke. Npr. razred *KonferencijaController* zadužen je za obradu zahtjeva poput stvaranja konferencija, ulaska u konferenciju i dohvaćanja osnovnih informacija o konferenciji.



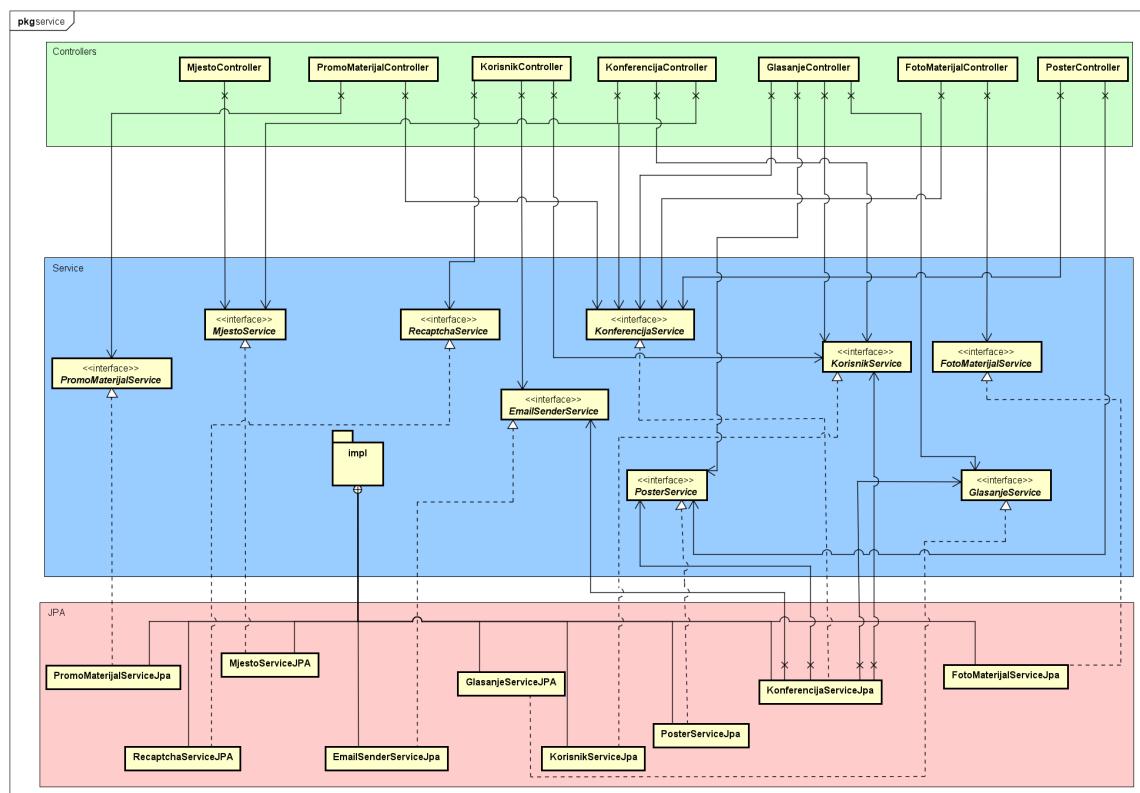
Slika 4.2: Dijagram razreda - Controllers

Svi razredi zaduženi za entitete i DTO (Data Transfer Objects) smješteni su u `src/main/java/opp/domain` paket. Razredi entiteta poput Konferencija, Poster, Mjesto itd. objekti su koji predstavljaju podatke u bazi podataka te su odgovorni za mapiranje objekata u redove u tablici baze podataka i obrnuto. Također, sadrže načine upravljanja podatcima i njihovu validaciju (osiguravaju integritet baze podataka). Razredi DTO također su objekti koji se koriste za razmjenu podataka između aplikacija, a u našem je slučaju to komunikacija između backend i frontend dijela.



Slika 4.3: Dijagram razreda - Domain

Sljedeći dijagram podijeljen je u 3 sloja: Controllers, Service i JPA koji sadrže pripadajuće razrede. Podjela je napravljena kako bi se postigla dobra organizacija i razdvajanje dijelova aplikacije. Sloj Controllers sadrži sve kontroler razrede te je njihova glavna uloga obrada HTTP zahtjeva. Sloj Service sadrži sučelja koja implementiraju razredi kontrolera. Oni pružaju apstrakciju između kontrolera i sloja podataka (JPA). Sloj Java Persistence API (JPA) zadužen je za pristup podacima u relacijskoj bazi podataka i za mapiranje java objekata u odgovarajući tip podatka u bazi podataka.

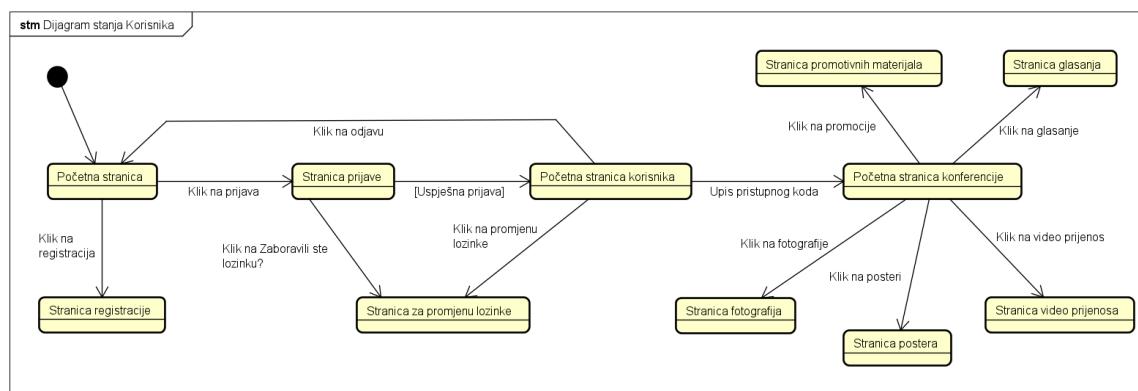


Slika 4.4: Dijagram razreda - Service

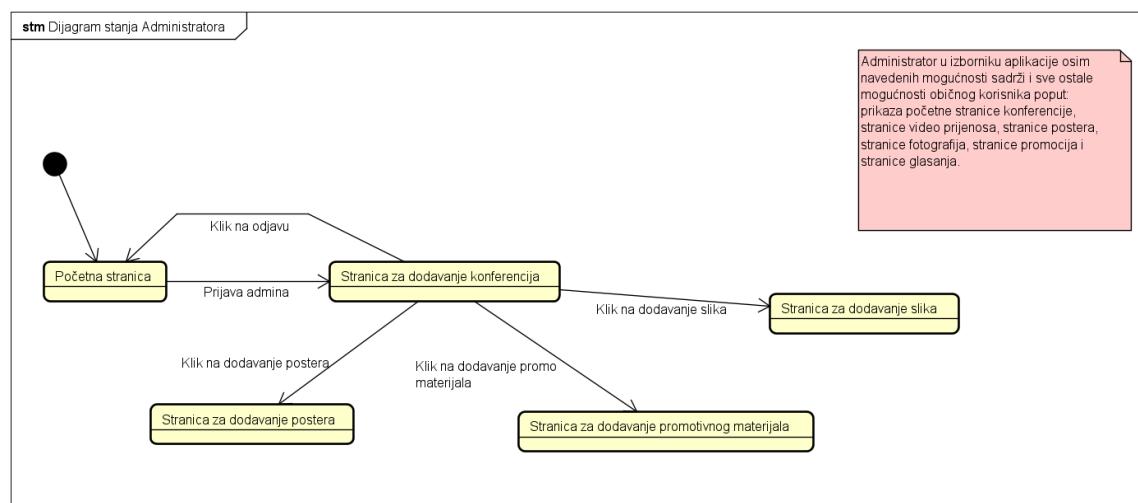
## 4.3 Dijagram stanja

Naredni dijagrami stanja opisuju prijelaze na druge dijelove (stranice) aplikacije. Izdvojena su dva dijagrama za korisnike 4.5 te administratore 4.6. Važno je napomenuti da administrator ima sve mogućnosti običnog korisnika.

Na dijagramima je opisan jedan od više mogućih načina prijelaza, jer korisnik ili administrator nakon prijave pomoću "Hamburger" izbornika koji se nalazi u gornjem lijevom uglu stranice može izravno odabratи gdje želi "otići". Naravno izbornik pokazuje samo prijelaze koji su odobreni korisniku/administratoru.



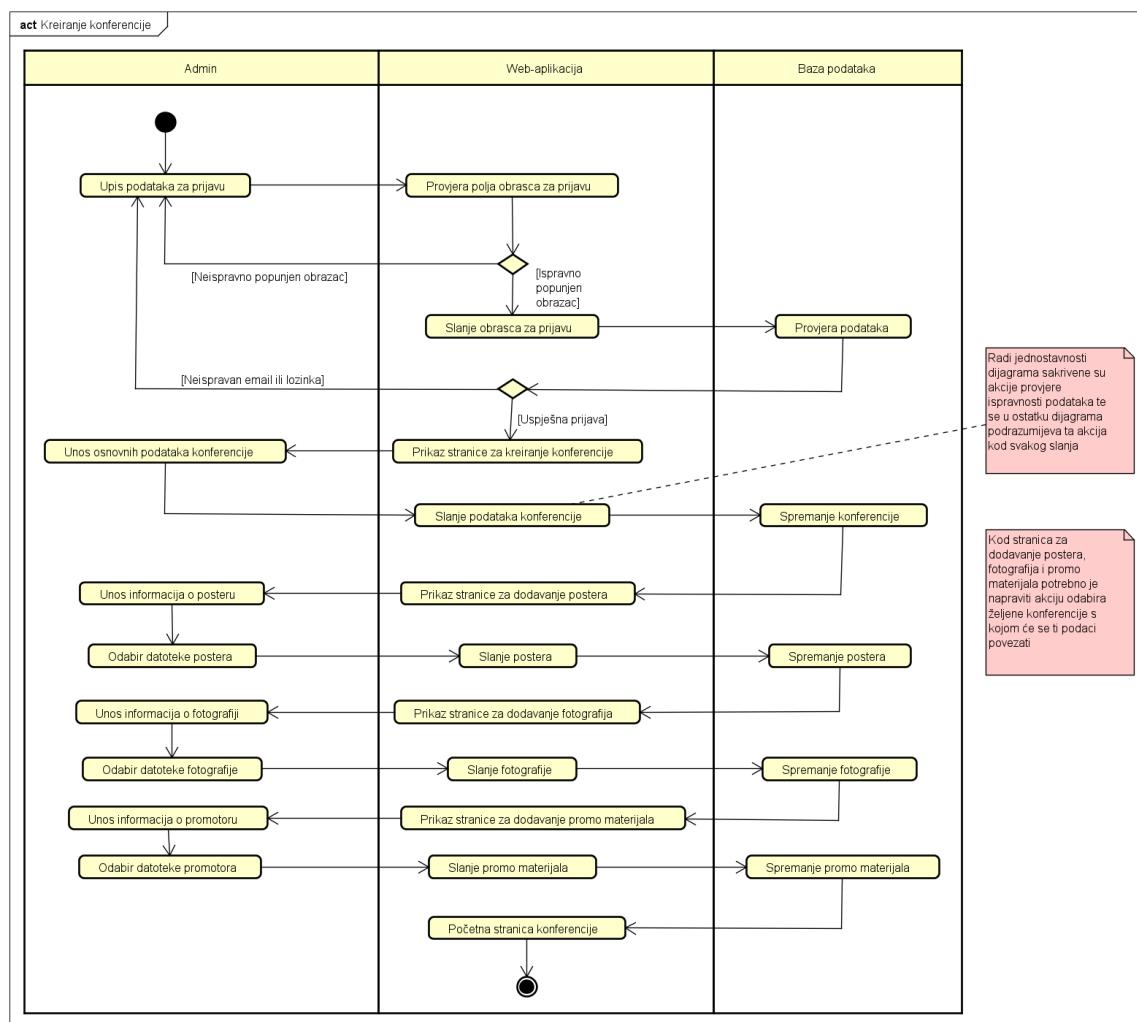
Slika 4.5: Dijagram stanja korisnika



Slika 4.6: Dijagram stanja administratora

## 4.4 Dijagram aktivnosti

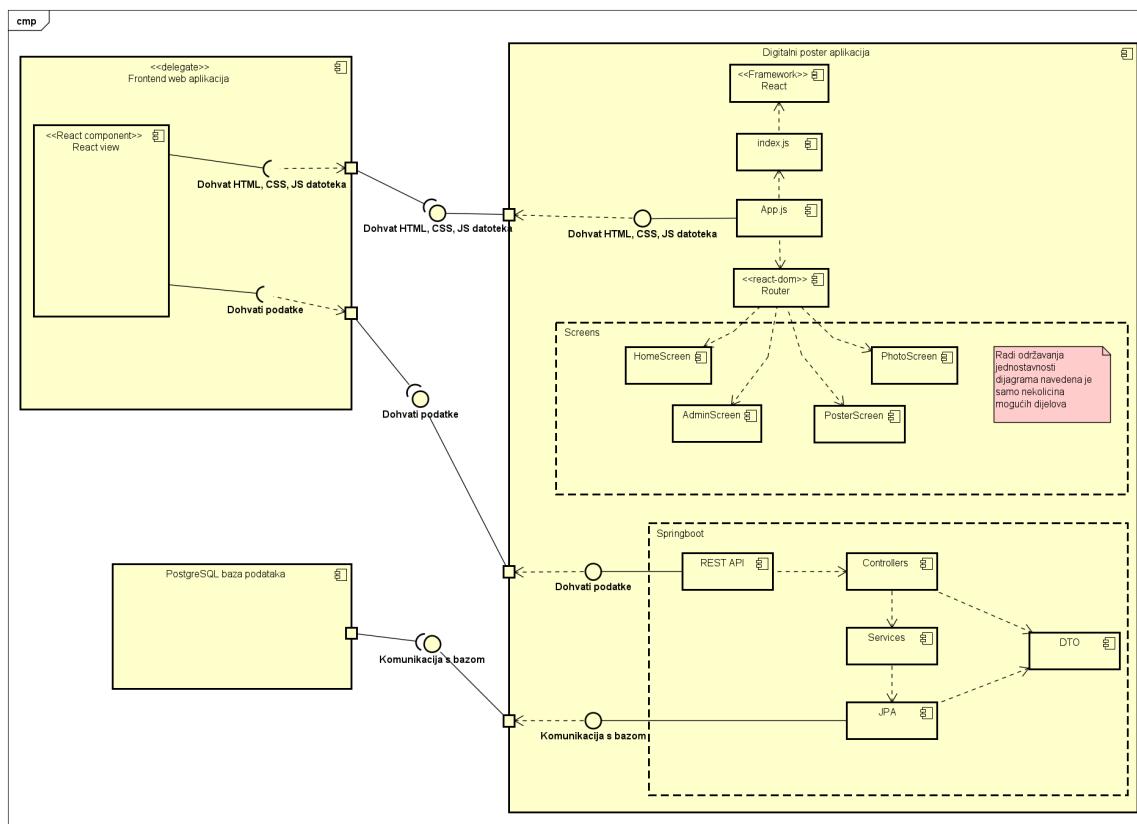
Dijagram aktivnosti na slici 4.7 prikazuje niz akcija potrebnih za stvaranja nove konferencije i konfiguriranje iste. Prvi korak je prijava korisnika koji ima administrativne mogućnosti koje su potrebne za kreiranje konferencije. Prijava se obavlja putem obrasca te se na strani aplikacije provjeravaju polja za unos. Ako je provjera uspješna provodi se provjera podataka i ovisno o ishodu ona je uspješna i možemo nastaviti dalje ili vraćamo korisnika na početak prijave. Daljnji koraci za dodavanja postera, fotografija i promotivnih materijala se svi mogu pojednostaviti u tri glavna koraka: 1. Unos podataka u obrazac, 2. Provjera unesenih podataka te slanje podataka i 3. Spremanje podataka. Važno je napomenuti da tijekom drugog koraka kod provjere unesenih podataka u slučaju neispravnog unosa, korisniku će se javiti greška i vratit će ga se na prijašnji korak. Tijek izvođenja akcija dodavanja postera, fotografija i promotivnih materijala je proizvoljan i može se ponavljati koliko god puta administrator želi. Na kraju administrator na početnoj stranici konferencije može vidjeti rezultat svog rada.



Slika 4.7: Dijagram aktivnosti: kreiranje konferencije

## 4.5 Dijagram komponenti

Dijagram komponenti opisuje organizaciju i međuovisnost komponenti programske potpore, odnosno njenu internu strukturu. Definirana su 3 glavna sučelja. Sučelje za dohvat HTML, CSS i JS datoteka zaduženo je za komunikaciju s frontend dijelom aplikacije, gdje uz pomoć definirane logike i kontrolera korisniku šalju odgovarajuće datoteke za prikaz pojedine stranice. Sučelje za dohvata podataka zaduženo je za komunikaciju frontend dijela aplikacije (React) s backend dijelom (Springboot) u cilju razmjene potrebnih podataka (konferencija, postera, fotografija...) te treće sučelje za komunikaciju s bazom podataka koje povezuje PostgreSQL bazu podataka i Springboot aplikaciju koristeći Java Persistence API kojim se omogućuje trajno spremanje podataka aplikacije.



Slika 4.8: Dijagram komponenti

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

Za izradu projekta koristili smo različite tehnologije, alate i platforme kako bismo učinkovito surađivali i razvili visokokvalitetnu aplikaciju. Za međusobnu komunikaciju unutar tima koristili smo aplikacije WhatsApp<sup>1</sup> i Discord<sup>2</sup> što nam je omogućilo brz i jednostavan način dijeljenja informacija i dogovaranja.

Sustav za upravljanje izvornim kodom bio je Git<sup>3</sup>, a repozitorij projekta smješten je na web platformi GitHub<sup>4</sup>. Ovo nam je omogućilo sinkronizaciju rada, praćenje promjena i suradnju na kodu na učinkovit način. Za izradu UML dijagrama korišten je alat Astah UML<sup>5</sup>, pružajući jasnu vizualizaciju strukture i odnosa unutar projekta.

Integrirano razvojno okruženje (IDE) koje smo koristili bilo je IntelliJ<sup>6</sup>, razvijeno u tvrtki JetBrains. IntelliJ je posebno prilagođen za rad s računalnim softverom napisanim u Javi, Kotlinu i Groovyju te pruža potporu za druge popularne jezike kao što su Python, JavaScript i TypeScript. To osigurava dosljedno iskustvo rada na različitim operativnim sustavima - Windows, macOS i Linux.

Za backend aplikacije koristili smo Spring Boot<sup>7</sup>, radni okvir baziran na Javi koji nudi autokonfiguraciju za olakšavanje početka razvoja, ali istovremeno omogućuje programerima da prilagode konfiguraciju prema potrebama.

Frontend je implementiran pomoću Reacta<sup>8</sup>, biblioteke u JavaScriptu<sup>9</sup> za izgradnju korisničkih sučelja. React se temelji na komponentama što je omogućilo razvoj složenih aplikacija s jasnom strukturom.

---

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

<sup>2</sup><https://discord.com/>

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

<sup>4</sup><https://github.com/>

<sup>5</sup><https://astah.net/products/astah-uml/>

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

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

<sup>8</sup><https://react.dev/>

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

Baza podataka projekta bila je PostgreSQL<sup>10</sup>, otvoreni sustav za upravljanje relacijskom bazom podataka. Konačno, aplikacija se nalazi na Renderu<sup>11</sup>, oblak platformi kao usluzi, koja podržava različite programske jezike i omogućuje laku implementaciju i upravljanje modernim aplikacijama.

Za ispitivanje sustava korišten je radni okvir Selenium<sup>12</sup> koji pruža automatsko testiranje kako bi se osigurala funkcionalnost i stabilnost sustava, a za ispitivanje komponenti JUnit<sup>13</sup>.

Kombinacija ovih tehnologija i alata omogućila je timu učinkovit rad i uspješan razvoj aplikacije.

---

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

<sup>11</sup><https://render.com/>

<sup>12</sup><https://www.seleniumhq.org/>

<sup>13</sup><https://junit.org/junit5/>

## 5.2 Ispitivanje programskog rješenja

Ispitivanje programskog riješena proveli smo na dva načina:

- *Ispitivanje komponenti - ostvareno pomoću JUnit testova u programskom jeziku Java*
- *Ispitivanje sustava - ostvareno pomoću Selenium alata za automatizirano testiranje web aplikacija*

### 5.2.1 Ispitivanje komponenti

Ispitivanje komponenti bazira se na ispitivanjima funkcija programa kao što su broj poziva određene funkcije, što točno vraćaju funkcije nakon što se izvedu, izaziva li se neka iznimka itd.

#### 1. Prvo testiranje testira kreiranje korisnika

Provjeravamo funkcionalnost funkcija za kreiranje korisnika te koliko puta se koja funkcija pozvala.

```
▲ II54152 *
@Test
void KorisnikCreateTest(){
    Korisnik korisnik = new Korisnik();
    korisnik.setEmail("test@example.com");
    korisnik.setPassword("password");

    when(passwordEncoder.encode(any())).thenReturn( value: "encodedPassword");
    when(korisnikRepo.save(any())).thenReturn(korisnik);

    Korisnik savedKorisnik = korisnikService.save(korisnik);

    verify(roleRepo, times( wantedNumberOfInvocations: 1)).findByName("ROLE_KORISNIK");
    verify(passwordEncoder, times( wantedNumberOfInvocations: 1)).encode( rawPassword: "password");
    verify(korisnikRepo, times( wantedNumberOfInvocations: 1)).save(any());
}
```

Slika 5.1: Prvo testiranje

#### 2. Drugo testiranje testira ulazak u konferenciju

U drugom testiranju provjeravamo ulazak u konferenciju kao i sve funkcionalnosti vezane uz nju.

```
@Test
public void testLoginKonf() {

    String password = "testPassword";
    Konferencija expectedKonferencija = new Konferencija();
    when(konferencijaService.findByPassword(password)).thenReturn(expectedKonferencija);

    try {
        ResponseEntity<?> responseEntity = konferencijaController.loginKonf( pass: "{\"password\":\"" + password + "\"}");
    } catch (JsonProcessingException e) {
        throw new RuntimeException(e);
    }

    verify(konferencijaService).findByPassword(password);
}
```

Slika 5.2: Drugo testiranje

### 3. Treće testiranje testira neuspješan ulazak u konferenciju

U trećem testiranju provjeravamo ponašanje naše aplikacije u slučaju neuspjelog ulaska u konferenciju kao i sve funkcionalnosti vezane uz nju.

```
@Test
public void testUnsuccessfulLoginKonf() {

    String password = "incorrectPassword";
    when(konferencijaService.findByPassword(password)).thenReturn( value: null);

    ResponseEntity<?> responseEntity = null;
    try {
        responseEntity = konferencijaController.loginKonf( pass: "{\"password\":\"" + password + "\"}");
    } catch (JsonProcessingException e) {
        throw new RuntimeException(e);
    }

    verify(konferencijaService).findByPassword(password);
    assertEquals(HttpStatus.UNAUTHORIZED, responseEntity.getStatusCode());
}
```

Slika 5.3: Treće testiranje

### 4. Četvrto testiranje testira dohvati svih konferencija

U četvrtom testiranju testiramo funkcionalnost dohvaćanja svih konferencija budući da je to jedna od ključnih funkcija prilikom kreiranja konferencije.

```
@Test
public void testGetAllKonf() {

    List<Konferencija> expectedKonferencije = Arrays.asList(new Konferencija(), new Konferencija());
    when(konferencijaService.listAll()).thenReturn(expectedKonferencije);

    ResponseEntity<List<Konferencija>> responseEntity = konferencijaController.getAllKonf();

    verify(konferencijaService, times(wantedNumberOfInvocations: 2)).listAll();
}

}
```

Slika 5.4: Četvrto testiranje

## 5. Peto testiranje testira dohvat svih lokacija

U petom testiranju testiramo funkcionalnost dohvaćanja lokacije koja je ključna za prikaz karte i vremenske prognoze.

```
@Test
public void testGetLocation() {

    long konferencijaId = 1L;

    Konferencija expectedKonferencija = new Konferencija();
    Mjesto expectedMjesto = new Mjesto();
    expectedKonferencija.setMjesto(expectedMjesto);
    when(konferencijaService.findByKonfid(konferencijaId)).thenReturn(expectedKonferencija);

    ResponseEntity<?> responseEntity = konferencijaController.getLocation(konferencijaId);
    verify(konferencijaService).findByKonfid(konferencijaId);
}

}
```

Slika 5.5: Peto testiranje

## 6. Šesto testiranje testira kreiranje mjesta uz konferenciju

U šestom testiranju testiramo funkcionalnost kreiranja i dodavanja mjesta nekoj konferenciji.

```

    @Test
    void testAddKonf() {
        Mjesto mjesto = new Mjesto();
        ResponseEntity<String> responseEntity = mjestoController.addKonf(mjesto);

        Mockito.verify(mjestoService).addKonferencija(any(Mjesto.class));
        assert responseEntity.getStatusCode() == HttpStatus.CREATED;
    }
}

```

Slika 5.6: Šesto testiranje

```

    @Test
    void testAddKonf() {
        Mjesto mjesto = new Mjesto();
        ResponseEntity<String> responseEntity = mjestoController.addKonf(mjesto);

        Mockito.verify(mjestoService).addKonferencija(any(Mjesto.class));
        assert responseEntity.getStatusCode() == HttpStatus.CREATED;
    }

    @Test
    public void testGetLocation() {
        long konferencijaId = 1L;

        Konferencija expectedKonferencija = new Konferencija();
        Mjesto expectedMjesto = new Mjesto();
        expectedKonferencija.setMjesto(expectedMjesto);
        when(konferencijaService.findById(konferencijaId)).thenReturn(expectedKonferencija);

        ResponseEntity<String> responseEntity = konferencijaController.getLocation(konferencijaId);
        verify(konferencijaService).findById(konferencijaId);
    }
}

[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 10.265 s - in opp.TeamsBackendApplicationTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  14.855 s
[INFO] Finished at: 2024-01-19T20:49:20+01:00
[INFO] -----

```

Slika 5.7: Rezultati testova

## 5.2.2 Ispitivanje sustava

Ispitivanje sustava proveli smo koristeći Selenium WebDriver i zadnju verziju Firefox web-preglednika. Selenium smo konfigurirali pomoću programskog jezika Java te je onda Selenium automatizirano testirao našu aplikaciju bazirajući se na naše testove. Za Selenium testiranje isključili smo reCAPTCHA.

### 1. Prvo testiranje testira registraciju korisnika

Provjeravamo funkcionalnost registracije korisnika. U polja upisujemo podatke te se pritišće gumb nakon upisa. Ukoliko je test prošao, korisnik je uspješno stvoren i prijavljen u aplikaciju.

```
@Test
private static boolean testRegKorisnik() {
    System.setProperty("webdriver.chrome.driver", "C:\\\\Program Files\\\\Firefox WebDriver\\\\geckodriver.exe");
    WebDriver driver = new FirefoxDriver();
    driver.manage().timeouts().implicitlyWait( time: 10, TimeUnit.SECONDS);
    driver.get("localhost:3000/");

    WebElement element3 = driver.findElement(By.className("dropdown-toggle"));
    element3.click();

    WebElement submitButton = driver.findElement(By.id("registerButton"));
    submitButton.click();

    WebElement imeInput = driver.findElement(By.id("ime"));
    imeInput.sendKeys( ...keysToSend: "Vaše ime");

    WebElement prezimeInput = driver.findElement(By.id("prezime"));
    prezimeInput.sendKeys( ...keysToSend: "Vaše prezime");

    WebElement emailInput = driver.findElement(By.id("Email"));
    emailInput.sendKeys( ...keysToSend: "vaš_email5@example.com");

    WebElement lozinkaInput = driver.findElement(By.id("password"));
    lozinkaInput.sendKeys( ...keysToSend: "Pa$$w0rd");

    WebElement potvrdaLozinkeInput = driver.findElement(By.id("confirmPassword"));
    potvrdaLozinkeInput.sendKeys( ...keysToSend: "Pa$$w0rd");

    WebElement regclick = driver.findElement(By.className("submit-button"));
    regclick.click();

    regclick.click();

    String redirURL = driver.getCurrentUrl();
    System.out.println("Current URL is: " + redirURL );
    driver.quit();
}
```

Slika 5.8: Prvo testiranje

## 2. Drugo testiranje testira dodavanje konferencije

U drugom testiranju provjeravamo dodavanje konferencije. Prije samog dodavanja konferencije administrator se mora prijaviti u sustav te nakon prijave administrator upisuje podatke za konferenciju te pritišće gumb za dodavanje. Ukoliko je test prošao, konferencija će biti dodana te će se prikazati poruka o uspješnosti našeg dodavanja.

```
@Test  
private static boolean addKonf() {  
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files\\Firefox WebDriver\\geckodriver.exe");  
    WebDriver driver = new FirefoxDriver();  
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
    driver.get("localhost:3000/");  
  
    WebElement element3 = driver.findElement(By.className("dropdown-toggle"));  
    element3.click();  
    WebElement submitButton = driver.findElement(By.id("loginButton"));  
    submitButton.click();  
    WebElement element = driver.findElement(By.className("input-field"));  
    element.sendKeys("ivana@horvat.hr");  
    WebElement element2 = driver.findElement(By.id("password"));  
    element2.sendKeys("Pa$$w0rd");  
    WebElement submitButton2 = driver.findElement(By.className("submit-button"));  
    submitButton2.click();  
  
    WebElement imeField = driver.findElement(By.name("ime"));  
    imeField.sendKeys("Naziv konferencije");  
  
    WebElement startTimeField = driver.findElement(By.name("startTime"));  
    startTimeField.sendKeys("2024-01-20T10:00");  
    WebElement endTimeField = driver.findElement(By.name("endTime"));  
    endTimeField.sendKeys("2024-01-21T18:00");  
  
    WebElement passwordField = driver.findElement(By.name("password"));  
    passwordField.sendKeys("tajnaLozinka");  
  
    WebElement liveLinkField = driver.findElement(By.name("liveLink"));  
    liveLinkField.sendKeys("http://www.live-link.com");  
  
    WebElement locationField = driver.findElement(By.name("location"));  
    locationField.sendKeys("Zagreb");  
    WebElement streetField = driver.findElement(By.name("street"));  
    streetField.sendKeys("Unska");  
    WebElement streetNumberField = driver.findElement(By.name("streetNumber"));  
    streetNumberField.sendKeys("3");  
    WebElement pbrField = driver.findElement(By.name("pbr"));  
    pbrField.sendKeys("10000");  
    WebElement dodajkonficgumbic = driver.findElement(By.id("dodajKonf"));
```

Slika 5.9: Drugo testiranje

### 3. Treće testiranje testira prijavu korisnika

U trećem testiranju provjeravamo prijavu korisnika. U prazna polja upisujemo potrebne podatke te na kraju pritišćemo gumb za prijavu. Ukoliko je test uspješan, vratit će nas na početnu stranicu.

```

@Test
private static boolean testLogInKorisnik() {
    System.setProperty("webdriver.chrome.driver", "C:\\\\Program Files\\\\Firefox WebDriver\\\\geckodriver.exe");
    WebDriver driver = new FirefoxDriver();
    driver.manage().timeouts().implicitlyWait( time: 10, TimeUnit.SECONDS);
    driver.get("localhost:3000/");

    WebElement element3 = driver.findElement(By.className("dropdown-toggle"));
    element3.click();

    WebElement submitButton = driver.findElement(By.id("LoginButton"));
    submitButton.click();

    WebElement element = driver.findElement(By.className("input-field"));
    element.sendKeys( ...keysToSend: "ivana@horvat.hr");

    WebElement element2 = driver.findElement(By.id("password"));
    element2.sendKeys( ...keysToSend: "Pa$$w0rd");

    WebElement submitButton2 = driver.findElement(By.className("submit-button"));
    submitButton2.click();

    String redirectURL = driver.getCurrentUrl();
    driver.quit();
}

```

Slika 5.10: Treće testiranje

#### 4. Četvrto testiranje testira prijavu u konferenciju

U četvrtom testiranju testiramo prijavu u konferenciju. Unutar polja za unos lozinke, upisujemo lozinku i pritišćemo gumb za prijavu. Ukoliko je test uspješan, aplikacija će nas vratiti na početnu stranicu konferencije.

```

▲ II54152 *
@Test
public static boolean testLogInKonf() {
    System.setProperty("webdriver.chrome.driver", "C:\\\\Program Files\\\\Firefox WebDriver\\\\geckodriver.exe");
    WebDriver driver = new FirefoxDriver();
    driver.manage().timeouts().implicitlyWait( time: 10, TimeUnit.SECONDS);
    driver.get("localhost:3000/");

    WebElement element = driver.findElement(By.className("codeInput"));
    element.sendKeys( ...keysToSend: "123");
    WebElement submitButton = driver.findElement(By.className("submitButton"));
    submitButton.click();

    String redirectURL = driver.getCurrentUrl();
    driver.quit();
}

```

Slika 5.11: Četvrto testiranje

The screenshot shows an IDE interface with several panes:

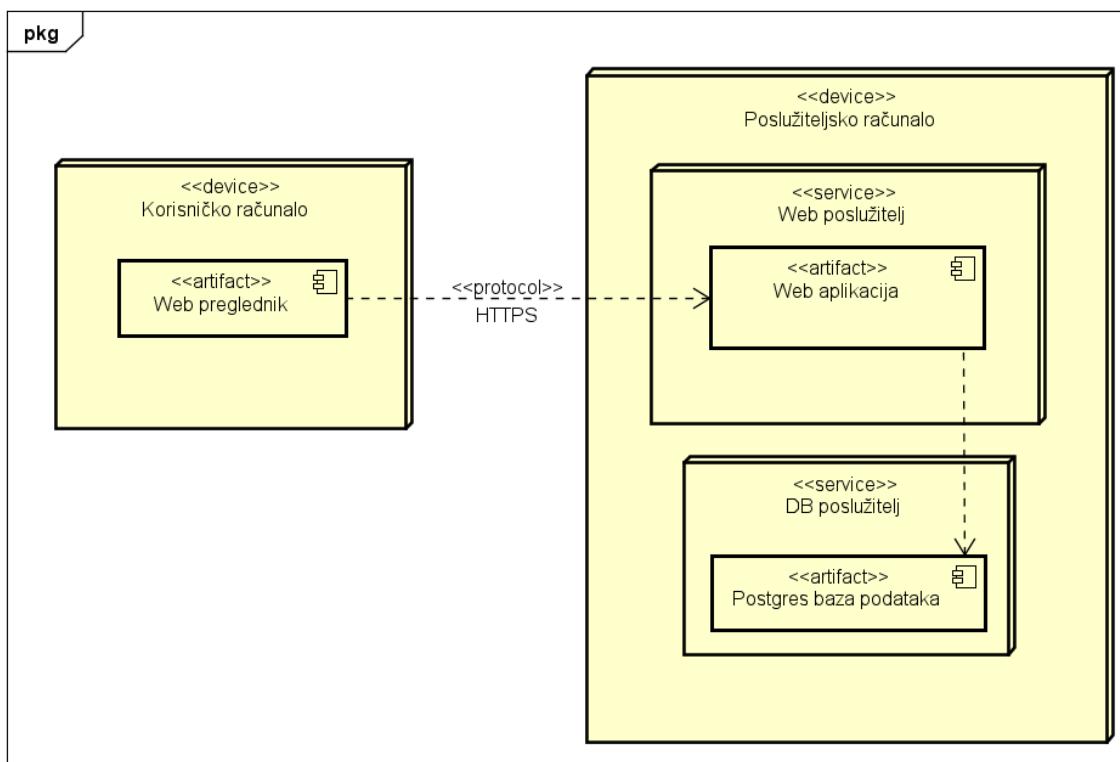
- Left pane:** A code editor window titled "TeamsBackendApplicationTests.java" containing Java test code. The code includes imports for Selenium and Mockito, and defines a test method named "testRegKorisnik".
- Middle pane:** A terminal or log window displaying Maven build output. It shows the test results and the build success message.
- Right pane:** A "Maven" tool window showing the project structure and available goals. The "Lifecycle" section is expanded, with "test" selected.

```
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 31.718 s - in opp.TeamsBackendApplicationTests
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 37.180 s
[INFO] Finished at: 2024-01-19T20:59:26+01:00
[INFO] -----
```

Slika 5.12: Rezultati testova

### 5.3 Dijagram razmještaja

Dijagram razmještaja 5.13 opisuje topologiju sklopolja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom okruženju. Sustav je baziran na arhitekturi "klijent-poslužitelj". Poslužiteljsko računalo (Render) sadrži web poslužitelj na kojem se nalazi naša web aplikacija te sadrži poslužitelj baze podataka na kojemu je naša PostgreSQL baza podataka. Komunikacija između klijenta i poslužitelja obavlja se preko sigurnog kanala ostvarenog protokolom HTTPS.



Slika 5.13: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

Za puštanje aplikacije u pogon (eng. deployment) koristili smo web stranicu Render. Prema uputama danim na GitHub repozitoriju tvrtke CROZ (<https://github.com/progi-devops/>) pripremili smo naš sustav za puštanje u pogon. Kako bismo mogli koristiti usluge servisa Render, prvotno smo se morali prijaviti u sustav, a to smo učinili pomoću istih podataka s kojima smo stvorili GitHub repozitorij na kojem držimo i održavamo naš projekt te njegov kod. Prvotno se podiže virtualna baza, a nakon nje potrebno je namjestiti skripte (navedene u package.json dokumentu i Dockerfile dokumentu) koje se izvode prilikom izgradnje rješenja projekta te prilikom samog pokretanja tog izgrađenog rješenja. To vrijedi i za pripremu "backend" i "frontend" dijela. Također, za "frontend" dio, u uputama se mogu pronaći koraci kako osigurati neometano lokalno pokretanje aplikacije bez obzira na novododane datoteke. Pri podizanju baze nije potrebno unositi gotovo nikakve podatke, već samo vrstu bazu kojom se koristimo (u našem slučaju PostgreSQL) te vremensku zonu kojoj pripadamo. Nakon toga Render nam povratno pruža sve potrebne informacije koje nam trebaju za lokalno spajanje na virtualnu bazu te povezivanje "backenda" na nju. Nakon toga moramo pustiti u pogon "backend" dio aplikacije kako bismo dobili URL na koji će naš "frontend", pušten u pogon, slati zahtjeve i s kojega će primati odgovore. Dockerfile koji nam je potreban za puštanje u pogon kreirali smo po uzoru na dani Dockerfile sa već spomenutog GitHub repozitorija uz manje preinake kod podataka vezanih za povezivanje na bazu. Isto vrijedi za package.json dokument te app.js. Također, bitno je istaknuti da Render prilikom puštanja u pogon koristi prethodno zadalu granu s odabranog repozitorija s koje potom uzima kod koji konkretno pušta u pogon. Tako prilikom svakog novog postavljanja promjena (eng. commit and push) na odabranu granu na Render se servisu automatski ponovno pušta u pogon kompletna aplikacija (osim baze podataka) s novim promjenama.

**Connections**

Hostname: dpg-cl9speto7jlc73fiu4qg-a

Port: 5432

Database: progi\_baza\_9iho

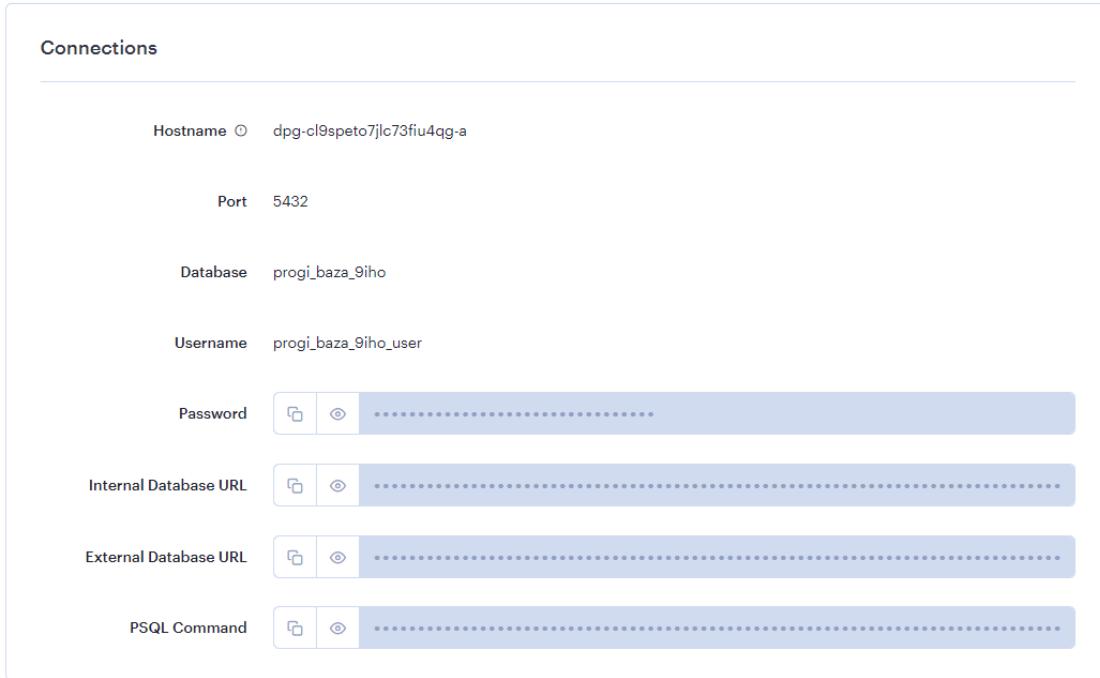
Username: progi\_baza\_9iho\_user

Password:  ..... (redacted)

Internal Database URL:  ..... (redacted)

External Database URL:  ..... (redacted)

PSQL Command:  ..... (redacted)



Slika 5.14: Generirani podatci za povezivanje na bazu

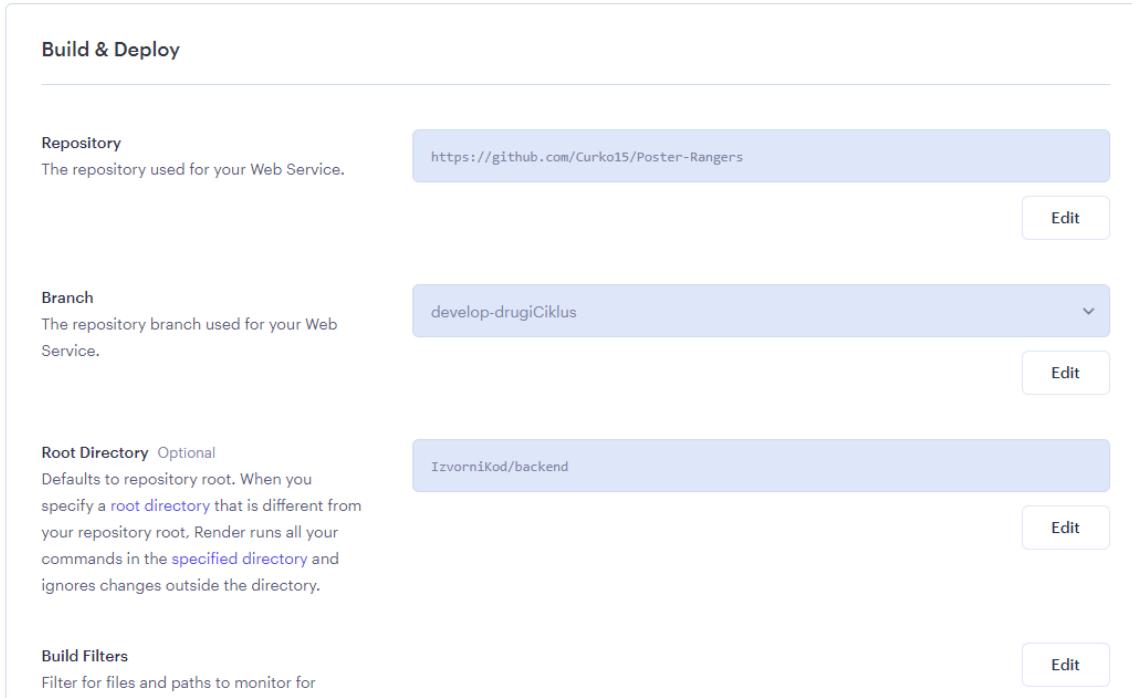
**Build & Deploy**

**Repository**  
The repository used for your Web Service.  
 Edit

**Branch**  
The repository branch used for your Web Service.  
 Edit

**Root Directory** Optional  
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.  
 Edit

**Build Filters**  
Filter for files and paths to monitor for  
 Edit



Slika 5.15: Postavljanje repozitorija, grane te direktorija za "backend"

**Build & Deploy**

**Repository**  
The repository used for your Web Service.

<https://github.com/Curko15/Poster-Rangers>

**Edit**

**Branch**  
The repository branch used for your Web Service.

develop-drugiCiklus

**Edit**

**Root Directory** Optional  
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

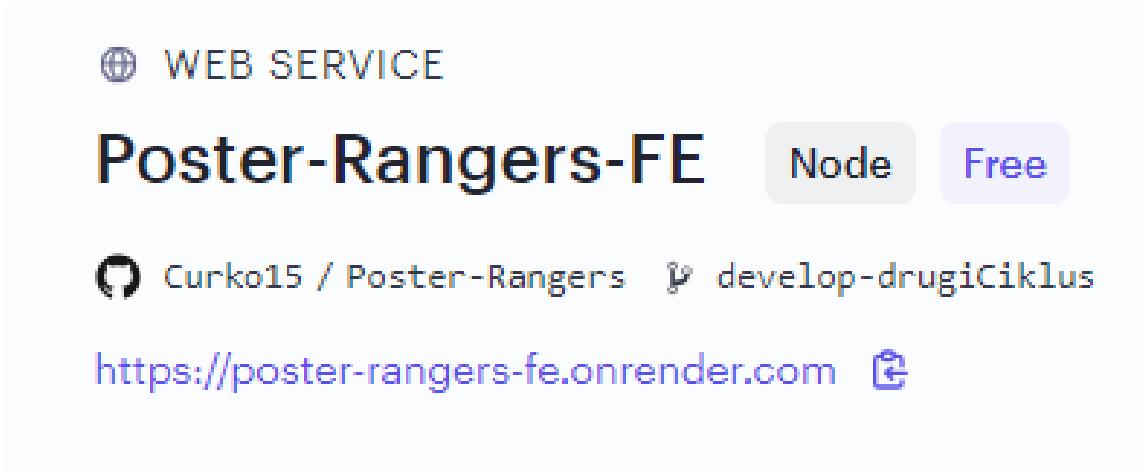
IzvorniKod/frontend

**Edit**

**Build Filters**  
Filter for files and paths to monitor for changes when building the service.

**Edit**

Slika 5.16: Postavljanje repozitorija, grane te direktorija za "frontend"



Slika 5.17: generirani URL za pristup puštenoj aplikaciji

## 6. Zaključak i budući rad

Zadatak našeg tima bio je razviti web-aplikaciju za stručne konferencije koja bi organizatorima omogućila veći doseg ljudi budući da bi se na konferenciji moglo sudjelovati i online te bi se u sklopu svake konferencije mogli pregledavati stručni radovi (posteri) drugih ljudi i za te rade glasati. Na projektu smo radili 3 mjeseca i napisljeku ga uspješno završili i ostvarili sve potrebne funkcionalnosti.

U prvoj fazi projekta, prije samog početka izrade projektnog zadatka, diskutirali smo o njegovoj temi i razmišljali kako najbolje organizirati i razviti aplikaciju. Nakon uspostave plana, izradili smo bazu podataka koja je ključni dio svake aplikacije, krenuli s izradom dokumentacije pa napisljeku i sa samom implementacijom. U ovoj smo fazi implementirali većinu potrebnih funkcionalnosti poput prijave, registracije, stvaranja konferencije i admina. Pojedini su članovi imali nešto više programerskog iskustva što je zasigurno pomoglo bržem napretku projekta i drugim članovima tima dalo dovoljno vremena da se uhodaju u svoje zadatke. Prije predaje prve revizije aplikaciju smo postavili na Render što nam je bio jedan od brojnih tehničkih izazova ove faze.

U drugoj smo fazi implementirali ostale potrebne funkcionalnosti među kojima su dodavanje i prikaz postera, fotografija i promotivnih materijala. U toj smo se fazi susreli s izazovima oko prikazivanja fotografija, no i taj smo problem riješili i stekli potrebna znanja za ubuduće. Dorađivanjem finesa, testiranjem aplikacije i dovršetkom dokumentacije završili smo sa svojim projektnim zadatkom vrlo zadovoljni učinjenim.

Za cijelo vrijeme trajanja izrade projekta, vrlo se bitnom stavkom pokazala međusobna komunikacija i pomoć među članovima tima, redoviti online sastanci i zajedništvo u rješavanju izazova te uključenost našeg voditelja u sve korake procesa.

Moguća nadogradnja aplikacije donijela bi korisnicima mogućnost primanja obavijesti o događanjima na konferenciji i davanje feedbacka na cjelokupnu konferenciju, a tijekom vremena zasigurno bi se moglo nadodati još novih funkcionalnosti koje bi korisnicima poboljšale iskustvo prisustvovanja na konferenciji.

Sudjelovanje u ovom projektu bilo je intenzivno iskustvo koje nam je pružilo dublji uvid u kompleksnosti timskog rada programera. Razvili smo vještine organizacije, suradnje i komunikacije unutar tima. Rad na projektu pružio nam je priliku primijeniti teorijsko znanje u stvarnom okruženju praktičnom primjenom principa programskog inženjerstva. Shvatili smo važnost timskog duha, prilagodbe te jasne komunikacije, a istovremeno smo svjesni prostora za osobni i timski napredak. Za brže i kvalitetnije izvođenje projekata u budućnosti, prepoznajemo potrebu za usavršavanjem u područjima agilnog upravljanja projektima, testiranja softvera te optimizacije koda. Ovo iskustvo postat će temelj našeg profesionalnog razvoja, osiguravajući da budući projekti budu izvedeni s još većim uspjehom.

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
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. Dodatno predavanje Spring Boot, <https://gitlab.com/hrvojesimic/progi-project-teams-spring-boot>
8. Dodatno predavanje React, <https://gitlab.com/jtomic/opp-project-teams-frontend>
9. Dodatno predavanje Deploy Render, <https://github.com/progi-devops/deploy-render>

# Indeks slika i dijagrama

2.1	Primjer početne stranice i stranice profila . . . . .	9
2.2	Primjer stranice za agendu i sesije . . . . .	9
2.3	Primjer stranice za gamifikaciju i stranice sponzora . . . . .	10
2.4	Primjer stranice za povezivanje i razgovor . . . . .	10
2.5	Primjer stranice za instalaciju mobilne aplikacije <i>run.events</i> u sklopu konferencije <i>KulenDayz</i> . . . . .	11
2.6	Primjer stranice za agendu implementirane iz <i>run.events</i> koda . . . .	11
2.7	Primjer stranice za sesije implementirane iz <i>run.events</i> koda . . . .	12
2.8	Primjer stranice za govornike implementirane iz <i>run.events</i> koda . .	12
3.1	Dijagram obrazaca uporabe, prijava u sustav . . . . .	25
3.2	Dijagram obrazaca uporabe, funkcionalnosti korisnika . . . . .	26
3.3	Dijagram obrazaca uporabe, funkcionalnosti administratora . . . . .	27
3.4	Sekvencijski dijagram obrasca UC1: Prijava u sustav odgovarajućom lozinkom . . . . .	28
3.5	Sekvencijski dijagram obrasca UC4: Registracija u sustav prijavljenih korisnika . . . . .	30
3.6	Sekvencijski dijagram obrasca UC6: Glasanje za najbolji stručni poster konferencije . . . . .	32
3.7	Sekvencijski dijagram obrasca UC7: Povlačenje korisnikovog glasa stručnog postera . . . . .	33
4.1	Dijagram baze podataka . . . . .	41
4.2	Dijagram razreda - Controllers . . . . .	42
4.3	Dijagram razreda - Domain . . . . .	43
4.4	Dijagram razreda - Service . . . . .	44
4.5	Dijagram stanja korisnika . . . . .	45
4.6	Dijagram stanja administratora . . . . .	45
4.7	Dijagram aktivnosti: kreiranje konferencije . . . . .	47
4.8	Dijagram komponenti . . . . .	48

5.1	Prvo testiranje . . . . .	51
5.2	Drugo testiranje . . . . .	52
5.3	Treće testiranje . . . . .	52
5.4	Četvrto testiranje . . . . .	53
5.5	Peto testiranje . . . . .	53
5.6	Šesto testiranje . . . . .	54
5.7	Rezultati testova . . . . .	54
5.8	Prvo testiranje . . . . .	55
5.9	Drugo testiranje . . . . .	56
5.10	Treće testiranje . . . . .	57
5.11	Četvrto testiranje . . . . .	57
5.12	Rezultati testova . . . . .	58
5.13	Dijagram razmještaja . . . . .	59
5.14	Generirani podatci za povezivanje na bazu . . . . .	61
5.15	Postavljanje repozitorija, grane te direktorija za "backend" . . . . .	61
5.16	Postavljanje repozitorija, grane te direktorija za "frontend" . . . . .	62
5.17	generirani URL za pristup puštenoj aplikaciji . . . . .	62
6.1	Dijagram pregleda promjena: cijela grupa . . . . .	73
6.2	Dijagram pregleda promjena: Marko Ćurković . . . . .	73
6.3	Dijagram pregleda promjena: Daria Bevanda . . . . .	74
6.4	Dijagram pregleda promjena: Andrija Dumančić . . . . .	74
6.5	Dijagram pregleda promjena: Korina Jurić . . . . .	75
6.6	Dijagram pregleda promjena: Luka Lulić . . . . .	75
6.7	Dijagram pregleda promjena: Luka Miličević . . . . .	76
6.8	Dijagram pregleda promjena: Fran Talan . . . . .	76

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 18. listopada 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - rasprava o projektnom zadatku (brainstorming)
  - okvirna podjela poslova

### 2. sastanak

- Datum: 20. listopada 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - detaljno proučavanje zadatka
  - okviran dogovor što bi trebalo implementirati

### 3. sastanak

- Datum: 22. listopada 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, L. Lulić i L. Miličević
- Teme sastanka:
  - dogovor i skica baze podataka

### 4. sastanak

- Datum: 23. listopada 2023.
- Prisustvovali: A. Dumančić, K. Jurić i F. Talan
- Teme sastanka:
  - dogovor oko izgleda korisničkog sučelja i potrebnih funkcionalnosti

### 5. sastanak

- Datum: 28. listopada 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić,

L. Miličević i F. Talan

- Teme sastanka:

- izrada konceptualnog modela baze podataka

#### 6. sastanak

- Datum: 4. studenoga 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - pregled dotad napravljene dokumentacije
  - podjela daljnjih poslova

#### 7. sastanak

- Datum: 8. studenoga 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - rasprava oko promjene baze i obrazaca
  - dodavanje funkcija superadmina i admina
  - rasprava oko promjene frontenda

#### 8. sastanak

- Datum: 11. studenoga 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - spajanje frontenda i backenda

#### 9. sastanak

- Datum: 13. studenoga 2023.
- Prisustvovali: M. Ćurković, A. Dumančić, L. Lulić i F. Talan
- Teme sastanka:
  - popravljanje komunikacije između backend i frontend dijela

#### 10. sastanak

- Datum: 15. studenoga 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, K. Jurić, L. Miličević
- Teme sastanka:
  - pokušaj postavljanja aplikacije na javni poslužitelj

**11. sastanak**

- Datum: 15. studenoga 2023.
- Prisustvovali: A. Dumančić, L. Lulić i F. Talan
- Teme sastanka:
  - popravljanje frontenda

**12. sastanak**

- Datum: 16. studenoga 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - postavljanje web-aplikacije na poslužitelj za prvu reviziju

**13. sastanak**

- Datum: 18. prosinca 2023.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - rasprava o dalnjim zadacima i potrebnim funkcionalnostima

**14. sastanak**

- Datum: 27. prosinca 2023.
- Prisustvovali: M. Ćurković, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - diskusija o backendu

**15. sastanak**

- Datum: 27. prosinca 2023.
- Prisustvovali: D. Bevanda, A. Dumančić, K. Jurić i F. Talan
- Teme sastanka:
  - diskusija o frontendu

**16. sastanak**

- Datum: 17. siječnja 2024.
- Prisustvovali: D. Bevanda, M. Ćurković, A. Dumančić, K. Jurić, L. Lulić, L. Miličević i F. Talan
- Teme sastanka:
  - pregled aplikacije i dogovor oko dalnjih koraka prije konačne predaje i završetka projekta

## Tablica aktivnosti

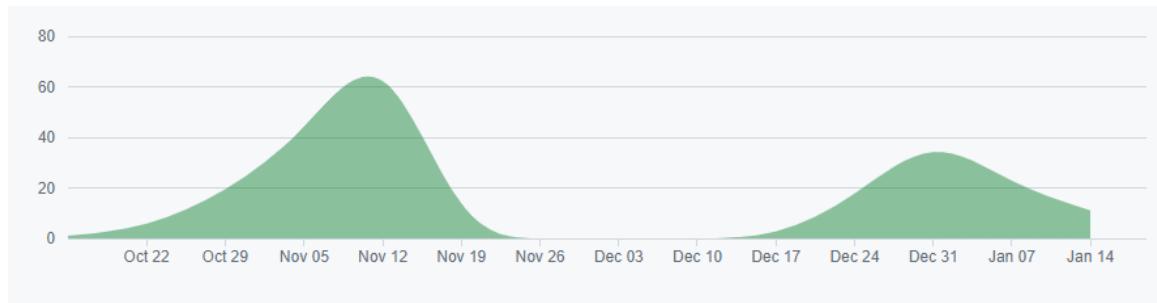
	Marko Ćurković	Daria Bevanda	Andrija Dumaničić	Korina Juric	Luka Lulić	Luka Miličević	Fran Talan
Upravljanje projektom	14	10	7	7	7	7	8
Opis projektnog zadatka				12			
Funkcionalni zahtjevi	2				3		
Opis pojedinih obrazaca	5				6		
Dijagram obrazaca			5			7	
Sekvencijski dijagrami			3			4	
Opis ostalih zahtjeva							2
Arhitektura i dizajn sustava		8	2	2		2	5
Baza podataka	3	8					
Dijagram razreda						8	
Dijagram stanja						2	
Dijagram aktivnosti						3	
Dijagram komponenti						3	
Korištene tehnologije i alati				2			
Ispitivanje programskog rješenja	3				2		
Dijagram razmještaja						2	
Upute za puštanje u pogon		4					
Dnevnik sastajanja				2			
Zaključak i budući rad				3			

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Marko Ćurković	Daria Bevanda	Andrija Dumančić	Korina Jurić	Luka Lulić	Luka Miličević	Fran Talan
Popis literature				1			
<i>dijagram pregleda promjena</i>				1			
<i>provjera pravopisa i čitljivosti</i>				5		4	
<i>izrada baze podataka</i>	2	4	5	2	2	3	2
<i>spajanje s bazom podataka</i>	6	4			5		
<i>back end</i>	48		25		52	7	8
<i>front end</i>		28	23	21		3	50
<i>postavljanje web-aplikacije na poslužitelj</i>		10		4		5	

## Dijagrami pregleda promjena



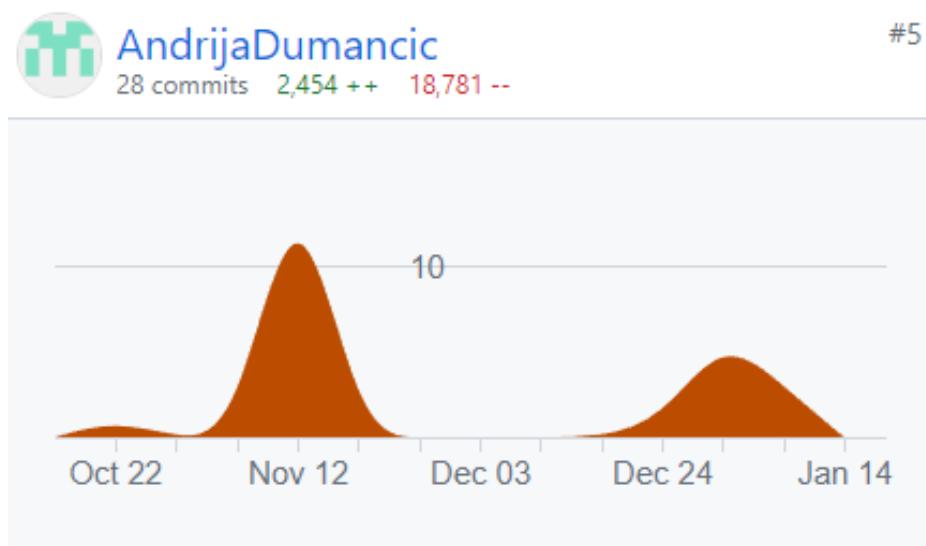
Slika 6.1: Dijagram pregleda promjena: cijela grupa



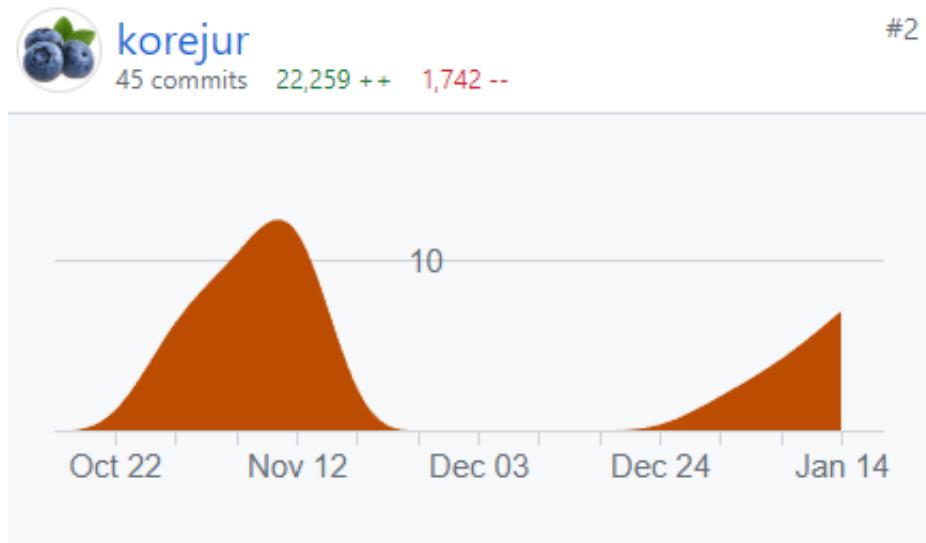
Slika 6.2: Dijagram pregleda promjena: Marko Ćurković



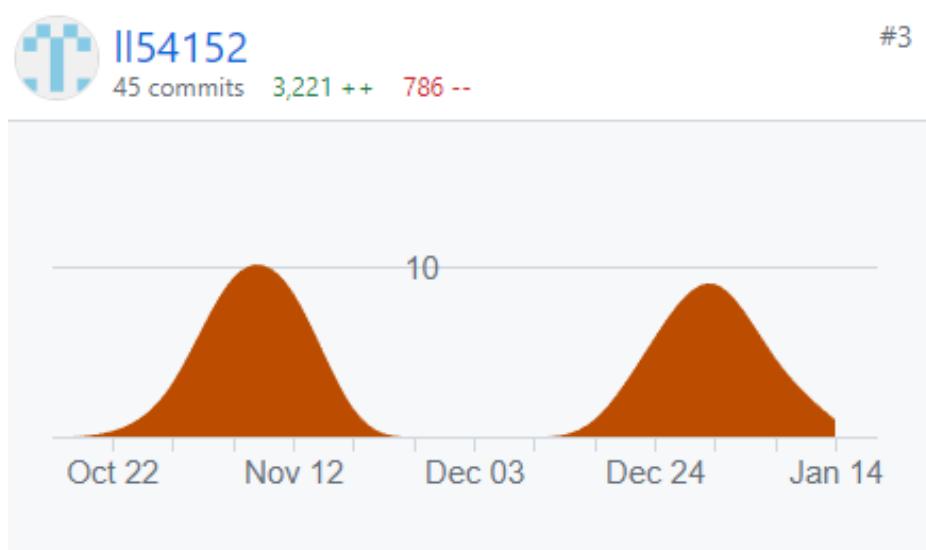
Slika 6.3: Dijagram pregleda promjena: Daria Bevanda



Slika 6.4: Dijagram pregleda promjena: Andrija Dumančić



Slika 6.5: Dijagram pregleda promjena: Korina Jurić



Slika 6.6: Dijagram pregleda promjena: Luka Lulić



Slika 6.7: Dijagram pregleda promjena: Luka Miličević



Slika 6.8: Dijagram pregleda promjena: Fran Talan