

# Programsko inženjerstvo

Ak. god. 2023./2024.

## Digitalni poster

Dokumentacija, Rev. 1.0

Grupa: *Posterheimer*

Voditelj: *Dominik Papeš*

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>8</b>
<b>3 Specifikacija programske potpore</b>	<b>12</b>
3.1 Funkcionalni zahtjevi . . . . .	12
3.1.1 Obrasci uporabe . . . . .	14
3.1.2 Sekvencijski dijagrami . . . . .	26
3.2 Ostali zahtjevi . . . . .	31
<b>4 Arhitektura i dizajn sustava</b>	<b>32</b>
4.1 Baza podataka . . . . .	34
4.1.1 Opis tablica . . . . .	35
4.1.2 Dijagram baze podataka . . . . .	41
4.2 Dijagram razreda . . . . .	42
4.3 Dijagram stanja . . . . .	46
4.4 Dijagram aktivnosti . . . . .	47
4.5 Dijagram komponenti . . . . .	49
<b>5 Implementacija i korisničko sučelje</b>	<b>51</b>
5.1 Korištene tehnologije i alati . . . . .	51
5.2 Ispitivanje programskog rješenja . . . . .	53
5.2.1 Ispitivanje komponenti . . . . .	53
5.2.2 Ispitivanje sustava . . . . .	63
5.3 Dijagram razmještaja . . . . .	71
5.4 Upute za puštanje u pogon . . . . .	72
<b>6 Zaključak i budući rad</b>	<b>88</b>
<b>Popis literature</b>	<b>89</b>
<b>Indeks slika i dijagrama</b>	<b>91</b>

**Dodatak: Prikaz aktivnosti grupe**

**92**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	V. Javornik	22.10.2023.
0.2	Dodan osnovni opis projektnog zadatka	V. Javornik	23.10.2023.
0.2.1	Manje promjene opisa projektnog zadatka	V. Javornik	24.10.2023.
0.2.2	Ažuriran dnevnik sastajanja	V. Javornik	24.10.2023.
0.3	Dodani aktori	D. Papeš	24.10.2023.
0.3.1	Dopunjeni aktori	D. Papeš	25.10.2023.
0.4	Pobrojani funkcionalni zahtjevi	D. Papeš	25.10.2023.
0.4.1	Opisani obrasci uporabe 2-5	F. Androić	26.10.2023.
0.4.2	Male jezične promjene, dopunjeni aktori	F. Androić	26.10.2023.
0.4.3	Dodan prvi dio opisa obrazaca uporabe	V.Javornik	26.10.2023.
0.4.4	Definirani obrasci uporabe 6-9, 14-17 i 30, dodani nefunkcionalni zahtjevi, ažuriran opis projektnog zadatka	M. Perhat	26.10.2023.
0.4.5	Definirani obrasci uporabe 22-26	D. Papeš	27.10.2023.
0.4.6	Definirani obrasci uporabe 12-14	D. Tomšić	27.10.2023.
0.4.7	Definirani obrasci uporabe 19 i 20, pobrisan obrazac uporabe 21 (Uređivanje digitalnog postera)	F. Androić	27.10.2023.
0.5	Dodan početni dijagram obrazaca uporabe	F. Androić	27.10.2023.
0.5.1	Ažuriran opis i ostali zahtjevi	D. Papeš	1.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.5.2	Ažurirani obrasci uporabe	D. Papeš	1.11.2023.
0.5.3	Ažurirani obrasci uporabe	D. Papeš	4.11.2023.
0.6	Dodan sekvencijski dijagram za upravljanje digitalnim posterima	V. Javornik	4.11.2023.
0.6.1	Ažuriran popis aktera, dodani dijagrami obrazaca uporabe za opis korisničkih računa i za administratorove i natkorisnikove mogućnosti, uklonjen početni dijagram obrazaca uporabe	F. Androić	4.11.2023.
0.6.2	Dodan dijagram obrazaca uporabe za opis funkcionalnosti aplikacije, manje jezične promjene	D. Tomšić	5.11.2023.
0.6.3	Dodan sekvencijski dijagram za slanje e-pošte	M. Perhat	5.11.2023.
0.6.4	Dodan sekvencijski dijagram za glasanje	D. Papeš	6.11.2023.
0.7	Dodan opis tablica i dijagram baze podataka	V. Javornik	9.11.2023.
0.7.1	Dopunjen opis projektnog zadatka, izbrisane smjernice za 1. reviziju, dodan dijagram razreda, gramatički ispravci	F. Androić	13.11.2023.
0.7.2	Napisan opis dijagrama razreda, dodan Dijagram razreda - dio Controllers	V. Javornik	14.11.2023.
0.7.3	Dodan dijagram razreda, ažuriran dnevnik sastajanja, male jezične promjene	D. Tomšić	14.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.7.4	Izmjene opisa projektnog zadatka, izmjene obrazaca uporabe (Upravljanje podacima o konferenciji i Poziv na dodjelu nagrade), izmjene UC dijagrama, izmjene opisa dijagrama razreda	F. Androić	14.11.2023.
0.8	Dodan opis arhitekture i dizajna sustava	A. Batić	14.11.2023.
0.8.1	Uređen opis arhitekture i dodana slika, ažuriran dnevnik sastanaka	F. Androić	14.11.2023.
0.8.2	Usklađivanje opisa baze podataka sa stvarnom implementacijom	V. Javornik	16.11.2023.
0.8.3	Usklađivanje dijagrama razreda sa stvarnom implementacijom te ažuriranje opisa dijagrama	D. Tomšić	16.11.2023.
0.9.0	Završne izmjene prije prve predaje	D. Papeš	17.11.2023.
0.9.1	Ažurirani dijagrami razreda	D. Tomšić	17.11.2023.
1.0	Posljednja revizija, male gramatičke izmjene i raspored prve verzije dokumentacije	F. Androić	17.11.2023.
1.1	Dodani dijagram stanja i dijagram aktivnosti	V. Javornik	20.12.2023.
1.2	Dodan dijagram komponenti	V. Javornik	24.12.2023.
1.3	Navedene korištene tehnologije i alati	V. Javornik	01.01.2024.
1.4	Usklađivanje opisa svih tablica baze podataka sa stvarnom implementacijom	V. Javornik	05.01.2024.
1.5	Dodan dijagram razmještaja i nedovršeni zaključak	F. Androić	06.01.2024.
1.5.1	Uređeni opisi dijagrama i korištenih alata	F. Androić	07.01.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.5.2	Postman dodan u korištene tehnologije i alate, manje jezične promjene	V.Javornik	08.01.2024.
1.6	Planiranje ispitivanja, usklađivanje opisa projekta sa stvarnom implementacijom, obrisani obrasci uporabe vezani za osobne podatke, ispravljeni dijagrami stanja i aktivnosti	V.Javornik	10.01.2024.
1.6.1	Ispravak baze podataka, ispitivanje programskog rješenja	V.Javornik	11.01.2024.
1.6.2	Velike izmjene u opisima obrazaca uporabe, promjena njihovog redoslijeda, brisanje onih koji nisu realizirani, dopuna opisa projektnog zadatka, jezične promjene	F. Androić	15.01.2024.
1.6.3	Svi dijagrami obrazaca uporabe prilagođeni realnom stanju koda i izmijenjenim obrascima uporabe	F. Androić	15.01.2024.
1.6.3	Ispitivanje	V. Javornik	15.01.2024.
1.6.4	Manje izmjene u prvom i veće u drugom sekvencijskom dijagramu i u njihovim opisima, jezične, stilističke izmjene opisa zahtjeva, arhitekture i baze	F. Androić	16.01.2024.
1.6.4	Izbacivanje suvišnog koda u ispitnim slučajevima	V. Javornik	16.01.2024.
1.6.5	Ispravak dijagrama stanja, ispitivanje	V. Javornik	18.01.2024.
1.6.6	Razredni dijagrami "DAO" i "Modeli" prilagođeni funkcionalnosti koda	F. Androić	18.01.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.6.7	Opis postavljanja baze podataka i backenda na server, ažuriranje tablice aktivnosti	M. Perhat	18.01.2024.



## 2. Opis projektnog zadatka

Cilj projekta je razviti programsku podršku za stvaranje web aplikacije „Posterheimer“ koja će olakšati pregled radova sudionicima stručnih konferencija prikazom svakog prijavljenog rada odgovarajućim posterom/prezentacijom unutar pregledne galerije. Omogućit će glasovanje putem aplikacije, automatsko rangiranje autora i njihovih radova, te pregled dodatnih informacija o konferenciji kao što su datum i vrijeme njezinog održavanja, prikaz mjesta održavanja pomoću karte i vremenske prognoze za tu lokaciju, video prijenos konferencije, pregled fotografija s konferencije i pregled pokrovitelja konferencije.

Prilikom pokretanja aplikacije prikazuje se popis aktivnih konferencija. Prije pristupa određenoj konferenciji prikazuje se zahtjev za prijavu u sustav. Inicijalni pristup moguće je ostvariti pomoću generičkog korisničkog računa. Za prijavu tog tipa potrebno je generičko korisničko ime (adresa e-pošte predviđena za određenu konferenciju) i odgovarajuća generička lozinka. Spomenuti podaci bit će dostupni samo sudionicima konferencije koji će ih dobiti prilikom dolaska i prijave na samu konferenciju. Podaci vezani uz generički račun zajednički su za sve posjetitelje konferencije. Nakon što su pristupili konferenciji korištenjem generičkog korisničkog računa, korisnici mogu izraditi vlastiti korisnički račun pomoću opcije registracije. Nakon registracije, korisnici za pristup konferenciji (prijavu) mogu koristiti vlastiti korisnički račun.

U nastavku dokumenta, za korisnike koji su za pristup konferenciji koristili samo generički korisnički račun koristiti će se naziv "korisnici posjetitelji" ili kraće "posjetitelji". Korisnici koji još nisu posjetili konferenciju pomoću generičkog računa zvat će se "neregistrirani korisnici". Korisnici koji se registriraju, odnosno izrade svoj korisnički račun, zvat će se "registrirani korisnici".

Svaki posjetitelj ima pristup osnovnim podacima o samoj konferenciji (vrijeme i mjesto održavanja, kao i relevantnu vremensku prognozu, te prikaz navedene lokacije na karti). Osim toga, može pregledavati postere/prezentacije i izraditi vlastiti korisnički račun.

Za kreiranje novog računa potrebni su sljedeći podaci:

- adresa e-pošte

- lozinka
- ime
- prezime

Registracijom u sustav korisniku se dodjeljuju prava "registriranog korisnika". Registrirani korisnici nakon prijave zadržavaju sva prava koja ima korisnik posjetitelj te uz njih dobivaju i neke nove mogućnosti:

- glasovanje za jedan od ponuđenih postera
- praćenje trenutnih događanja u glavnoj konferencijskoj dvorani pomoću video prijenosa
- pregled i preuzimanje fotografija fotografiranih za vrijeme trajanja konferencije
- pregled pokrovitelja konferencije
- pregled konačnih rezultata glasovanja kada ono završi

Glasovanje je moguće samo tijekom određenog razdoblja koje je određeno danima i vremenom održavanja konferencije, odnosno ono će završiti dva dana prije završetka same konferencije. Nakon završenog postupka glasovanja, objavljuju se rezultati koji su dostupni svim registriranim korisnicima. Registrirani korisnici mogu dati glas samo jednom posteru te svoj glas ne mogu naknadno promijeniti. Glasovanje nije obavezno, dakle korisnici se mogu suzdržati od glasovanja.

Događanja na konferenciji će se prenositi uživo video prijenosom putem usluge "YouTube". Eventualne fotografije konferencije koje bi administrator objavio na stranici mogu pristupiti svi registrirani korisnici, koji ih zatim mogu pregledati u uvećanom izdanju ili preuzeti na svoj uređaj.

Autori postera ili prezentacija koji sudjeluju na stručnom skupu elektroničkom poštom dostavljaju sve potrebne materijale administratoru zaduženog za konferenciju proizvoljnom metodom izvan sklopa aplikacije (npr. elektroničkom poštom), a nakon završetka postupka glasovanja primaju obavijest o rangui svojeg rada prema glasovima posjetitelja. Porukom elektroničke pošte se prva tri nagrađena rada poziva na dodjelu nagrade. Također, sve se sudionike elektroničkom poštom obavještava o mjestu i vremenu dodjele nagrade. Rezultate glasanja mogu vidjeti samo registrirani korisnici pritiskom na prikladni gumb.

Uz korisnika posjetitelja i registriranog korisnika postoje još dvije vrste korisnika, a to su:

- administrator

- natkorisnik

Administrator konferencije zadužen je za upravljanje svim podacima vezanih uz konferenciju. Svaka konferencija ima vlastitog administratora. On ima pristup bazi s popisom registriranih korisnika i njihovim podacima te može brisati korisnike registrirane na konferenciji za koju je zadužen. Ne može upravljati registriranim korisnicima ostalih konferencija. Obavlja prijavu autora, radova i postera i ima mogućnost ažuriranja tih podataka. Dostupni su mu svi podaci te može definirati sve potrebne uvjete za ispravan rad sustava.

Natkorisnik je korisnik koji postoji u bazi podataka od samog postavljanja aplikacije na mrežu i ima ovlasti stvaranja i brisanja konferencija, te prilikom stvaranja nove konferencije registrira administratora zaduženog za nju. Prilikom stvaranja nove konferencije podaci koje je natkorisnik dužan unijeti koji služe kao njezin kontekst su:

- ime
- poveznica na video prijenos
- adresa lokala gdje se održava
- mjesto u kojem se održava
- poštanski broj navedenog mjesta
- vrijeme početka
- vrijeme završetka
- generička adresa e-pošte za prijavu posjetitelja
- generička lozinka za prijavu posjetitelja
- adresa e-pošte zaduženog administratora
- lozinka za zaduženog administratora

Aplikacija je građena za specifičnu uporabu stvaranja konferencija natjecateljskog tipa, čime je namjera zadovoljiti potrošačku nišu koju druge slične aplikacije poput "b2match.com", "Wix.com" i "monday.com" ne ciljaju. Također, cilj aplikacije je privući klijente iznimnom jednostavnošću dizajna i funkcionalnosti, što znači da ju je lako koristiti i pruža efikasan i brz način za organizirati željeni događaj.

Neke od mogućih nadogradnja aplikacije su:

- omogućiti administratoru estetsku konfiguraciju i personalizaciju stranice same konferencije, čime bi se olakšalo organizatorima da na vizualan način plasiraju svoju poruku, što bi bilo ostvareno jednostavnim sučeljem na kojem

bi administrator mogao birati boje elemenata stranice ili postavljati vlastite fotografije ili logotipe u prednji ili zadnji plan prozora

- prikaz više opcija o lokaciji održavanja konferencije, na primjer dodati interaktivnu kartu mjesta konferencije s označenim važnim lokacijama ili integrirati informacije o lokalnim restoranima, hotelima i prijevozu
- omogućiti registriranim korisnicima komentiranje postera
- implementirati dvofaktorsku ovjeru autentičnosti kako bi se dodatno osigurao pristup sustavu
- razvoj mobilne aplikacije za Android i iOS uređaje
- uvesti analitičke alate za praćenje korištenja platforme tijekom konferencije i generiranje izvještaja o učinkovitosti i angažmanu korisnika nakon završetka konferencije

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Organizator konferencije (naručitelj)
2. Pokrovitelji konferencije
3. Autori radova
4. Glasači
5. Administrator
6. Natkorisnik
7. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) prijaviti se pomoću generičkog korisničkog računa
2. Korisnik posjetitelj (inicijator) može:
  - (a) registrirati se, odnosno stvoriti novi korisnički račun
  - (b) pregledavati postere, pri čemu ne može glasati
  - (c) vidjeti informacije o vremenu i mjestu održavanja koje uključuju vremenske uvjete, vremensku prognozu i kartu
3. Registrirani/Prijavljeni korisnik (inicijator) može:
  - (a) pregledati i glasati za postere pri čemu svaki posjetitelj može glasati za najviše jedan poster
  - (b) pregledavati promotivne materijale pokrovitelja
  - (c) gledati video-prijenos
  - (d) pregledati i preuzimati fotografije
  - (e) pristupiti rezultatima jednom kad postanu dostupni
  - (f) odjaviti se

4. Korisnik (inicijator):

- (a) generalizacija korisnika posjetitelja, registriranog korisnika, administratora i natkorisnika
- (b) prijaviti se vlastitim podacima

5. Administrator (inicijator) može:

- (a) dodati ili ukloniti postere natjecatelja s potrebnim podacima
- (b) dodati ili ukloniti fotografije s događaja
- (c) dodati ili ukloniti logo pokrovitelja s potrebnim podacima
- (d) obrisati određene korisničke račune

6. Natkorisnik (inicijator) može:

- (a) stvoriti novu konferenciju sa svim podacima potrebnim za njenu funkciju
- (b) izbrisati konferenciju

7. Baza podataka (sudionik):

- (a) pohranjuje podatke o korisnicima
- (b) pohranjuje podatke o digitalnim posterima, broju njihovih glasova, njihovim autorima
- (c) pohranjuje fotografije uslikane za vrijeme trajanja konferencije
- (d) pohranjuje sve ostale potrebne podatke o konferenciji

8. Poslužitelj strujanja (sudionik):

- (a) pruža uslugu video prijenosa konferencije

9. Poslužitelj vremenske prognoze (sudionik):

- (a) poslužuje potrebne podatke vezane za prognozu vremena

10. Poslužitelj karte (sudionik):

- (a) poslužuje potrebne podatke za prikaz karte

11. Poslužitelj e-pošte (sudionik):

- (a) omogućuje aplikaciji da šalje e-poruke korisnicima

### 3.1.1 Obrasci uporabe

#### UC1 - Pristup konferenciji

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Pristupiti konferenciji kao korisnik posjetitelj
- **Sudionici:** Baza podataka
- **Preduvjet:** Nema preduvjeta
- **Opis osnovnog tijeka:**
  1. Korisnik odabire konferenciju kojoj želi pristupiti
  2. Upisuje generičko korisničko ime (predviđenu adresu e-pošte) i lozinku za odabranu konferenciju
  3. Otvara se pregled konferencije s mogućnostima korisnika posjetitelja
- **Opis mogućih odstupanja:**
  - 2.a Pogrešno generičko ime ili lozinka
    1. Sustav obavještava korisnika o upisu pogrešne lozinke ili imena

#### UC2 - Registracija

- **Glavni sudionik:** Korisnik posjetitelj
- **Cilj:** Stvoriti vlastiti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen s generičkim računom
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzete adrese e-pošte, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravne adrese e-pošte
    1. Sustav obavještava korisnika o neuspjelom upisu
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

#### UC3 - Prijava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pristup svim korisničkim funkcionalnostima aplikacije
- **Sudionici:** Baza podataka

- **Preduvjet:** Nema preduvjeta
- **Opis osnovnog tijeka:**
  1. Korisnik pritišće gumb za pristup konferenciji ili gumb za prijavu unutar konferencije
  2. Otvara se prozor u koji korisnik upisuje podatke
  3. Provjera postoji li korisnik u bazi podataka kao registrirani korisnik
  4. Ako korisnik postoji, otvara se prozor s podacima o konferenciji, a korisnik je prijavljen
- **Opis mogućih odstupanja:**
  - 1.a Korisnik pritišće gumb za prijavu natkorisnika
    1. Korisnik unosi natkorisničke podatke
    2. Otvara se natkorisnički prikaz stranice s popisom konferencija
  - 3.a Korisnik je Administrator
    1. Korisnik ulazi u aplikaciju s administratorskim ovlastima
  - 4.a Korisnik ne postoji u bazi podataka
    1. Sustav obavještava korisnika o upisu pogrešne lozinke ili adrese e-pošte

#### UC4 - Odjava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Odjaviti se iz konferencije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za odjavu
  2. Povratak na zaslon za pristup konferenciji

#### UC5 - Pregled informacija o konferenciji

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati informacije o mjestu održavanja konferencije, uključujući kartu i informacije o trenutnim vremenskim uvjetima i vremenskoj prognozi za navedenu lokaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik se upravo prijavljuje ili pritišće gumb "Konferencija"



2. Otvara se stranica s podacima o konferenciji

#### **UC6 - Prikaz video prijenosa**

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Gledati video prijenos konferencije
- **Sudionici:** Poslužitelj strujanja
- **Preduvjet:** Korisnik je registriran i prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik otvara prozor s podacima o konferenciji
  2. Pokreće se prijenos videa

#### **UC7 - Pregled galerije postera**

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled svih postavljenih postera
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik pritišće gumb "Posteri"
  2. Otvara se prozor s posterima

#### **UC8 - Pregled digitalnog postera/prezentacije**

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati poster
- **Sudionici:** Baza podataka
- **Preduvjet:** Pregled galerije postera
- **Opis osnovnog tijeka:**
  1. Korisnik odabire digitalni poster
  2. Korisnik pregledava poster/prezentaciju u .pdf formatu

#### **UC9 - Glasovanje**

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Glasovati za rad
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava, Pregled digitalnog postera
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju glasovanja

2. U sustavu se evidentira korisnikovo glasovanje
- **Opis mogućih odstupanja:**
  - 2.a Korisnik je već glasovao
    1. Korisnik je obaviješten o nemogućnosti daljnjeg glasovanja

#### UC10 - Pregled rezultata glasovanja

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Uvid u rezultate glasovanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava, registrirani korisnik
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju rezultati glasovanja u izborniku
  2. Otvara se zasebna stranica gdje je rang lista autora/radova
  3. Korisnik pregledava rang listu

#### UC11 - Pregled galerije fotografija

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati galeriju fotografija
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik pritišće gumb "Fotografije"
  2. Otvara se galerija fotografija

#### UC12 - Pregled fotografije

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati izabranu fotografiju iz galerije fotografija
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen te je otvorena galerija fotografija
- **Opis osnovnog tijeka:**
  1. Korisnik odabire fotografiju iz galerije
  2. Odabrana fotografije prikaže se uvećana

#### UC13 - Preuzimanje fotografije

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Preuzeti izabranu fotografiju iz galerije

- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen, otvoren je uvećani prikaz za pregled fotografije
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za preuzimanje fotografije
  2. Odabrana fotografija se preuzima na uređaj

#### **UC14 - Pregled promotivnih materijala**

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati promotivne materijale pokrovitelja konferencije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik pritišće gumb "Pokrovitelji"
  2. Otvara se stranica s promotivnim materijalima

#### **UC15 - Brisanje korisničkog računa**

- **Glavni sudionik:** Administrator
- **Cilj:** Izbrisati korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator otvori popis registriranih korisnika
  2. Administrator pritišće gumb "Ukloni"
  3. Podaci o odabranom korisniku se brišu iz baze podataka

#### **UC16 - Dodavanje digitalnog postera**

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje novog postera
- **Sudionici:** Baza Podataka
- **Preduvjet:** Administrator je prijavljen, otvoren je administratorov prikaz galerije postera
- **Opis osnovnog tijeka:**
  1. Administrator pritišće gumb za dodavanje novog postera
  2. Otvara se prozor s mogućnošću odabira lokalne datoteke i unosa podataka o posteru

3. Administrator odabire željenu datoteku za prijenos u bazu podataka aplikacije
  4. Administrator potvrđuje dodavanje novog postera
  5. Podaci o novom posteru se spremaju u bazu podataka
- **Opis mogućih odstupanja:**
    - 3.a Odabrana datoteka nije u .pdf formatu
      1. Sustav obavještava administratora o odabiru datoteke nepodržanog tipa
      2. Odbacivanje promjena

#### UC17 - Brisanje digitalnog postera

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje odabranog digitalnog postera
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen, otvoren je administratorov prikaz galerije postera
- **Opis osnovnog tijeka:**
  1. Administrator pritišće gumb "Obriši" ispod odabranog postera
  2. Podaci o odabranom posteru se brišu iz baze podataka, a on se više ne vidi u galeriji

#### UC18 - Dodavanje fotografije

- **Glavni sudionik:** Administrator
- **Cilj:** Dodavanje nove fotografije u galeriju fotografija
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je administrator, otvoren je administratorov prikaz galerije fotografija
- **Opis osnovnog tijeka:**
  1. Administrator pritišće gumb za dodavanje nove fotografije
  2. Otvara se prozor s poljem za unos lokalne datoteke
  3. Administrator odabire sliku s vlastitog računala
  4. Administrator potvrđuje postavljanje nove fotografije
  5. Podaci o novoj fotografiji se spremaju u bazu podataka
- **Opis mogućih odstupanja:**
  - 3.a Odabrana datoteka nije podržanog tipa

1. Sustav obavještava administratora o odabiru datoteke nepodržanog tipa
2. Odbacivanje promjena

### UC19 - Brisanje fotografije

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje odabrane fotografije
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je administrator, otvoren je administratorov prikaz galerije fotografija
- **Opis osnovnog tijeka:**
  1. Administrator odabire fotografiju
  2. Otvara se uvećani prikaz fotografije
  3. Administrator odabire gumb za brisanje fotografije
  4. Podaci o fotografiji se miču iz baze podataka i fotografija se miče iz galerije

### UC20 - Dodavanje promotivnog materijala

- **Glavni sudionik:** Administrator
- **Cilj:** Dodati novi promotivni materijal
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je administrator, otvoren je administratorov prikaz promotivnih materijala
- **Opis osnovnog tijeka:**
  1. Administrator pritišće gumb za dodavanje novog promotivnog materijala
  2. Administrator unosi potrebne podatke i lokalnu datoteku sa znakom pokrovitelja
  3. Administrator potvrđuje promjene
  4. Podaci o pokrovitelju se unose u bazu podataka
- **Opis mogućih odstupanja:**
  - 2.a Odabrana datoteka nije podržanog tipa
    1. Sustav obavještava administratora o odabiru datoteke nepodržanog tipa
    2. Odbacivanje promjena

### UC21 - Brisanje promotivnog materijala

- **Glavni sudionik:** Administrator
- **Cilj:** Brisanje odabranog promotivnog materijala
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je administrator, otvoren je administratorov prikaz promotivnih materijala
- **Opis osnovnog tijeka:**
  1. Administrator pritišće gumb "Obriši" ispod odabranog logotipa pokrovitelja
  2. Podaci o odabranom pokrovitelju se brišu iz baze podataka i logotip se više ne vidi na stranici
- **Opis mogućih odstupanja:**
  - 2.a Administrator odustaje od brisanja materijala
    1. Povratak na sučelje za upravljanje promotivnim materijalima

#### UC22 - Dohvat vremenske prognoze

- **Glavni sudionik:** Poslužitelj vremenske prognoze
- **Cilj:** Dohvatiti podatke o vremenskoj prognozi
- **Sudionici:** -
- **Preduvjet:** Prijavljen je registrirani korisnik, aplikacija je pokrenuta
- **Opis osnovnog tijeka:**
  1. Nakon prijave registriranog korisnika aplikacija šalje zahtjev za podacima poslužitelju vremenske prognoze

#### UC23 - Dohvat karte

- **Glavni sudionik:** Poslužitelj karte
- **Cilj:** Dohvatiti kartu
- **Sudionici:** -
- **Preduvjet:** Prijavljen je registrirani korisnik, aplikacija je pokrenuta
- **Opis osnovnog tijeka:**
  1. Nakon prijave registriranog korisnika aplikacija šalje zahtjev za podacima poslužitelju karte

#### UC24 - Slanje poruke e-pošte

- **Glavni sudionik:** Poslužitelj e-pošte
- **Cilj:** Poslati poruku e-pošte
- **Sudionici:** Baza podataka

- **Preduvjet:** Završen postupak glasovanja
- **Opis osnovnog tijeka:**
  1. Traženje i dohvat adresa e-pošte autora prvih triju radova po broju glasova iz baze podataka
  2. Slanje poziva na dodjelu nagrade autorima nagrađenih radova e-porukom i notifikacijom na aplikaciji
  3. Slanje obavijesti o mjestu i vremenu dodjele nagrade svim registriranim korisnicima e-porukom i notifikacijom na aplikaciji
  4. Dohvaćanje adresa e-pošte svih autora natjecateljskih radova
  5. Slanje obavijesti o rangui svim sudionicima natjecanja e-porukom i notifikacijom na aplikaciji

#### UC25 - Dodavanje konferencije

- **Glavni sudionik:** Natkorisnik
- **Cilj:** Stvoriti novu konferenciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je natkorisnik, otvoren je natkorisnički prikaz popisa konferencija
- **Opis osnovnog tijeka:**
  1. Natkorisnik odabire opciju za stvaranje nove konferencije
  2. Otvara se obrazac za unos podataka o potrebnih za stvaranje konferencije
  3. Natkorisnik unosi potrebne podatke
  4. Natkorisnik potvrđuje unos podataka
  5. Podaci o novostvorenoj konferenciji se spremaju u bazu podataka
- **Opis mogućih odstupanja:**
  - 4.a Natkorisnik odustaje od brisanja konferencije
    1. Povratak na popis konferencija

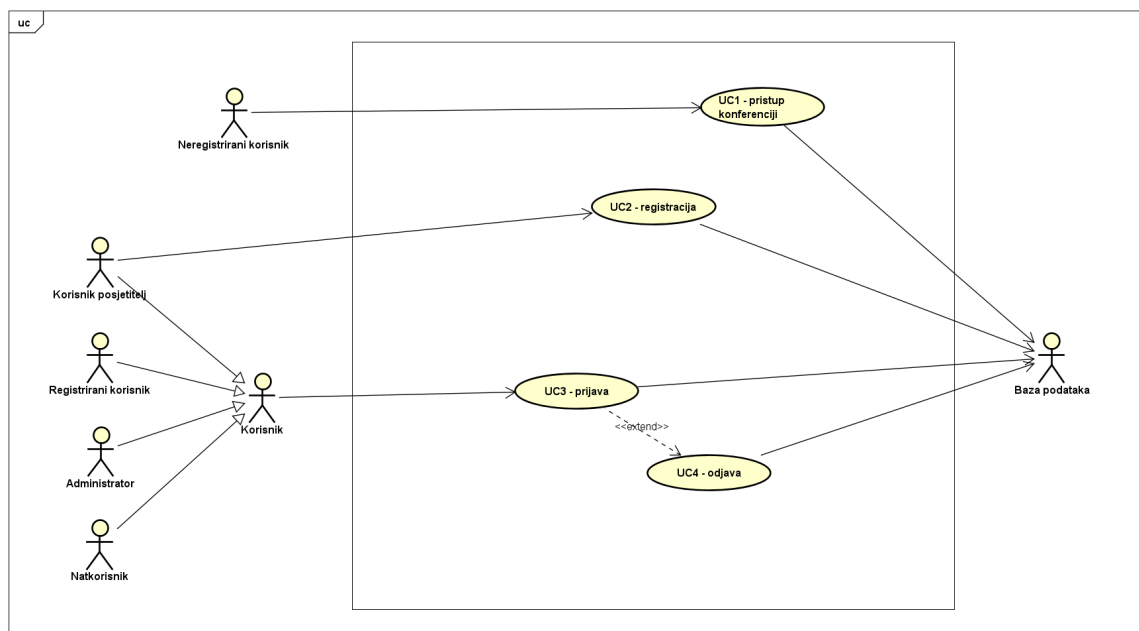
#### UC26 - Brisanje konferencije

- **Glavni sudionik:** Natkorisnik
- **Cilj:** Izbrisati konferenciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je natkorisnik, otvoren je natkorisnički prikaz popisa konferencija
- **Opis osnovnog tijeka:**

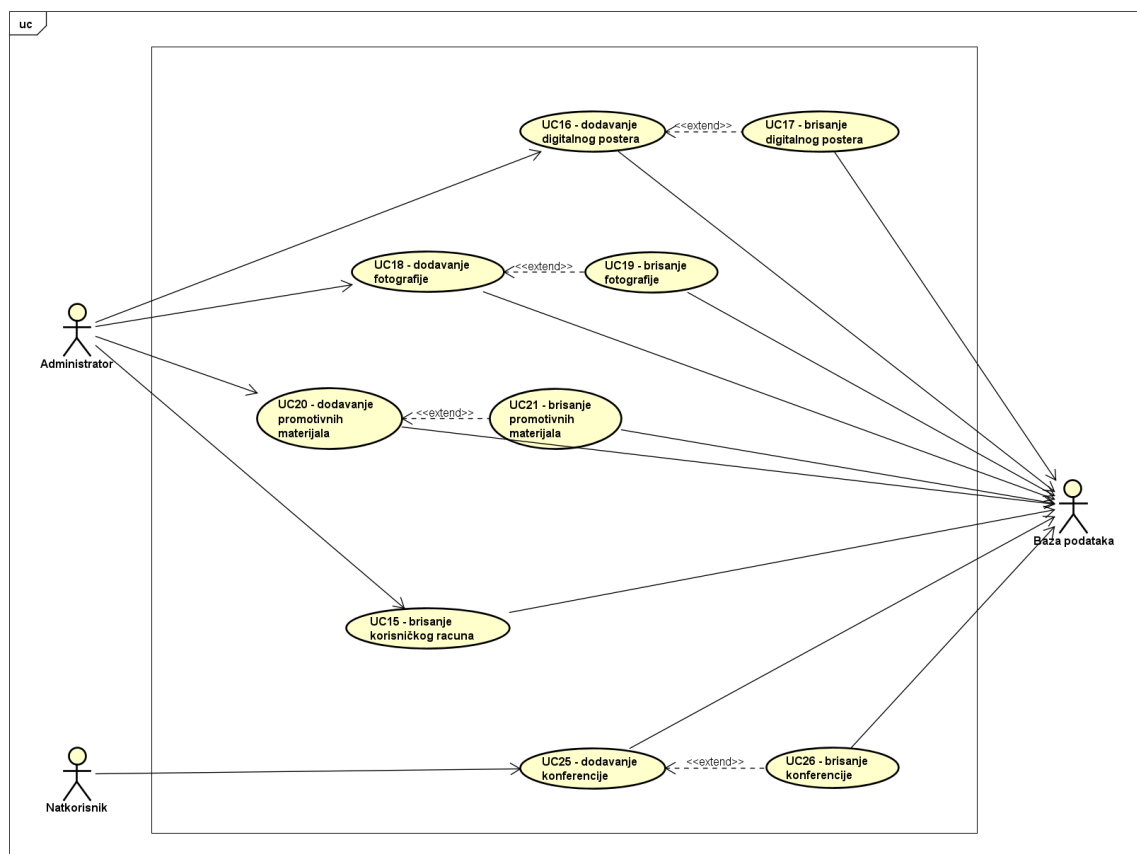
1. Korisnik pritisće gumb "Obriši" pored naziva konferencije koju želi obri-  
sati
2. Podaci o odabranoj konferenciji se brišu iz baze podataka i njezino ime  
više nije na popisu konferencija



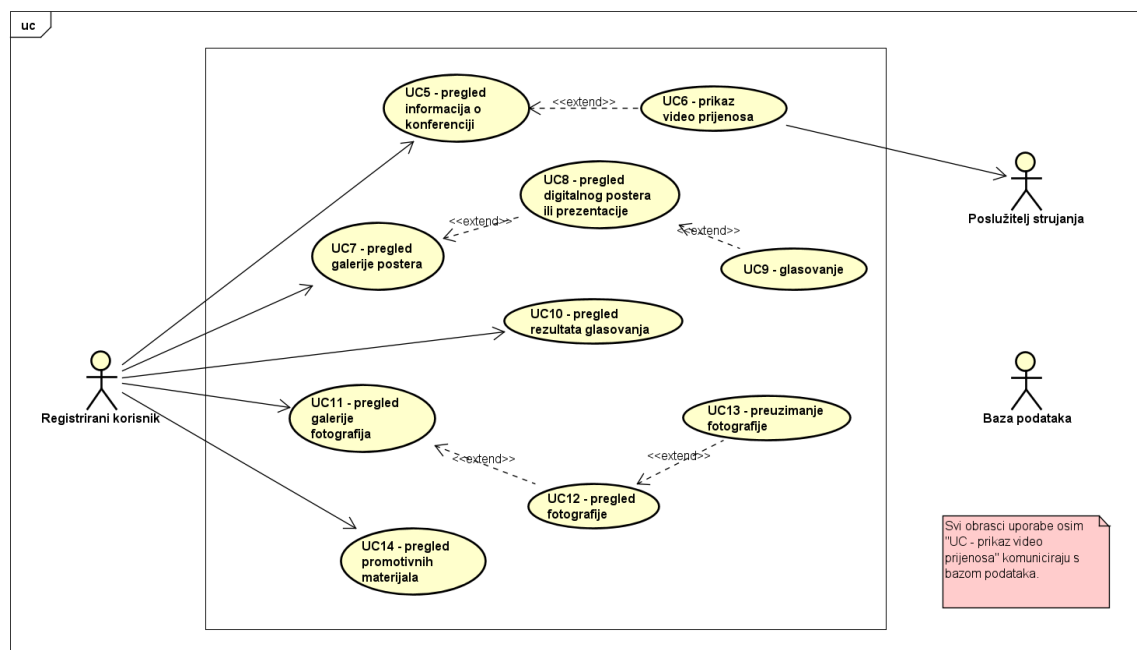
## Dijagrami obrazaca uporabe



Slika 3.1: UC dijagram za opis korisničkih računa



Slika 3.2: UC dijagram za opis administrativnih mogućnosti

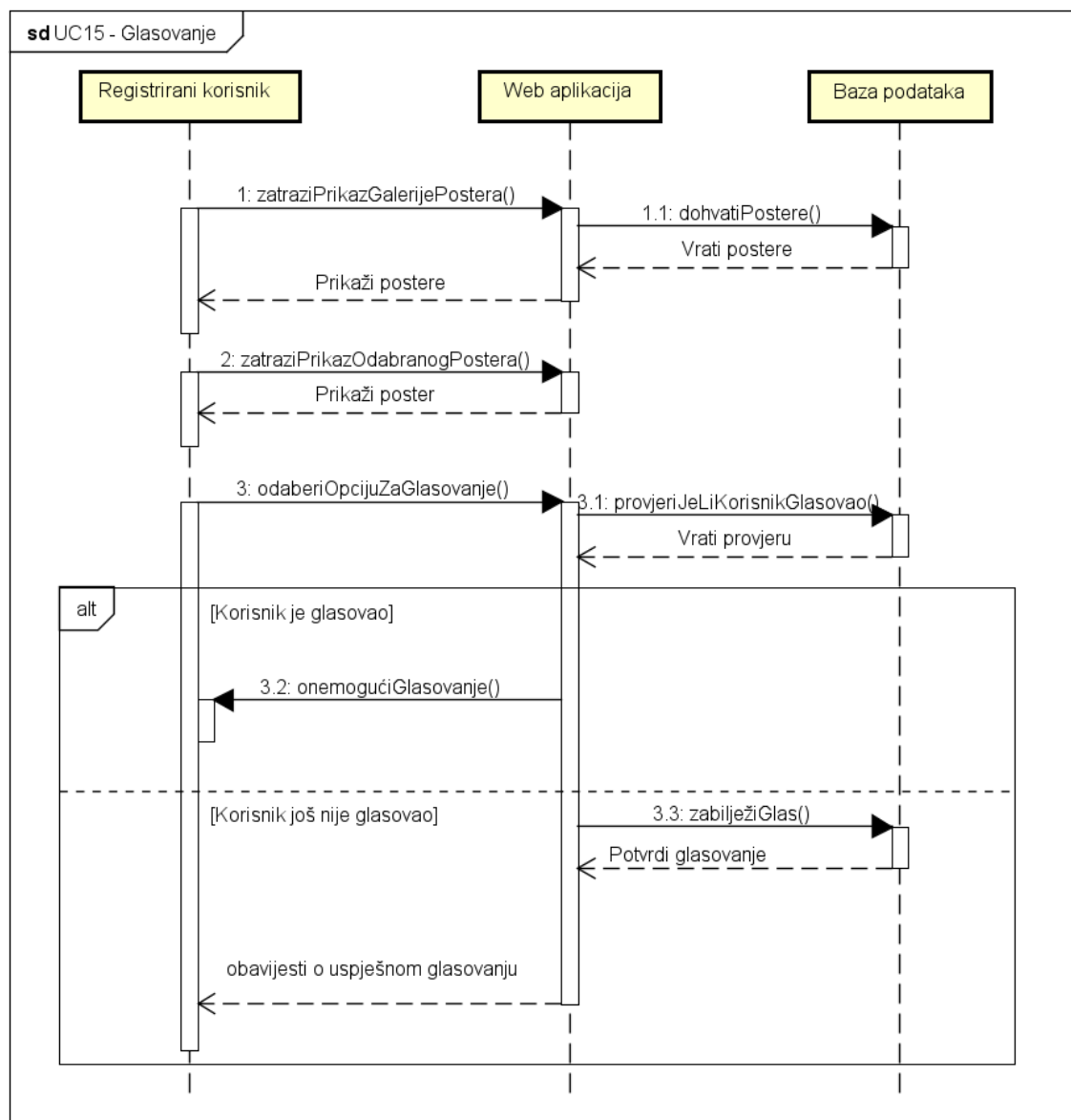


Slika 3.3: UC dijagram za opis funkcionalnosti aplikacije

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC9 - Glasovanje

Registrirani korisnik odabire opciju za pregled svih postera. U galeriji postera odabire jedan poster. Pojavljuje se opcija za glasovanje za odabrani poster. Klikom na tu opciju aplikacija šalje zahtjev bazi podataka kojim provjerava je li prijavljeni korisnik već glasao, te ako jest zabranjuje mu daljnje glasanje. Ukoliko korisnik nije glasao, baza podataka sprema glas. Aplikacija obavještava korisnika o uspješnom glasanju.



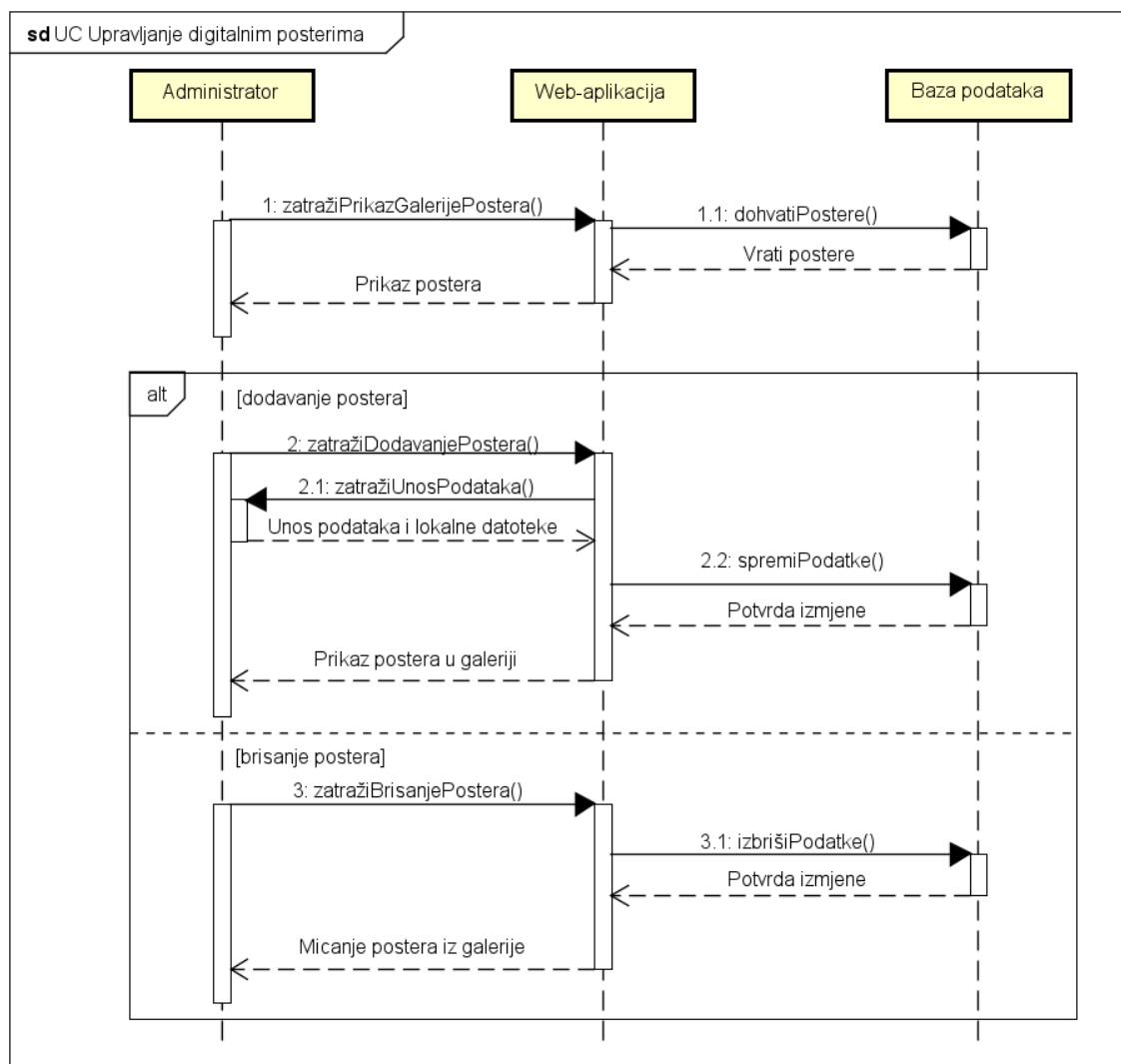
Slika 3.4: Sekvencijski dijagram za UC15 - Glasovanje

**Obrazac uporabe UC16 - Dodavanje digitalnog postera**

Administrator odabire opciju "Posteri". Poslužitelj dohvaća popis postera iz baze podataka te ih prikazuje administratoru, kao i gumb predviđen za dodavanje novog postera u galeriju. Pritiskom na njega otvara se obrazac za unos podataka o novom posteru i prijenos lokalne datoteke. Nakon što administrator pritiskom na gumb potvrdi unos podataka u bazu, web aplikacija to i učini, te kad baza pohrani podatak, preda natrag aplikaciji sve potrebne podatke za prikaz postera na sučelju stranice.

**Obrazac uporabe UC17 - Brisanje digitalnog postera**

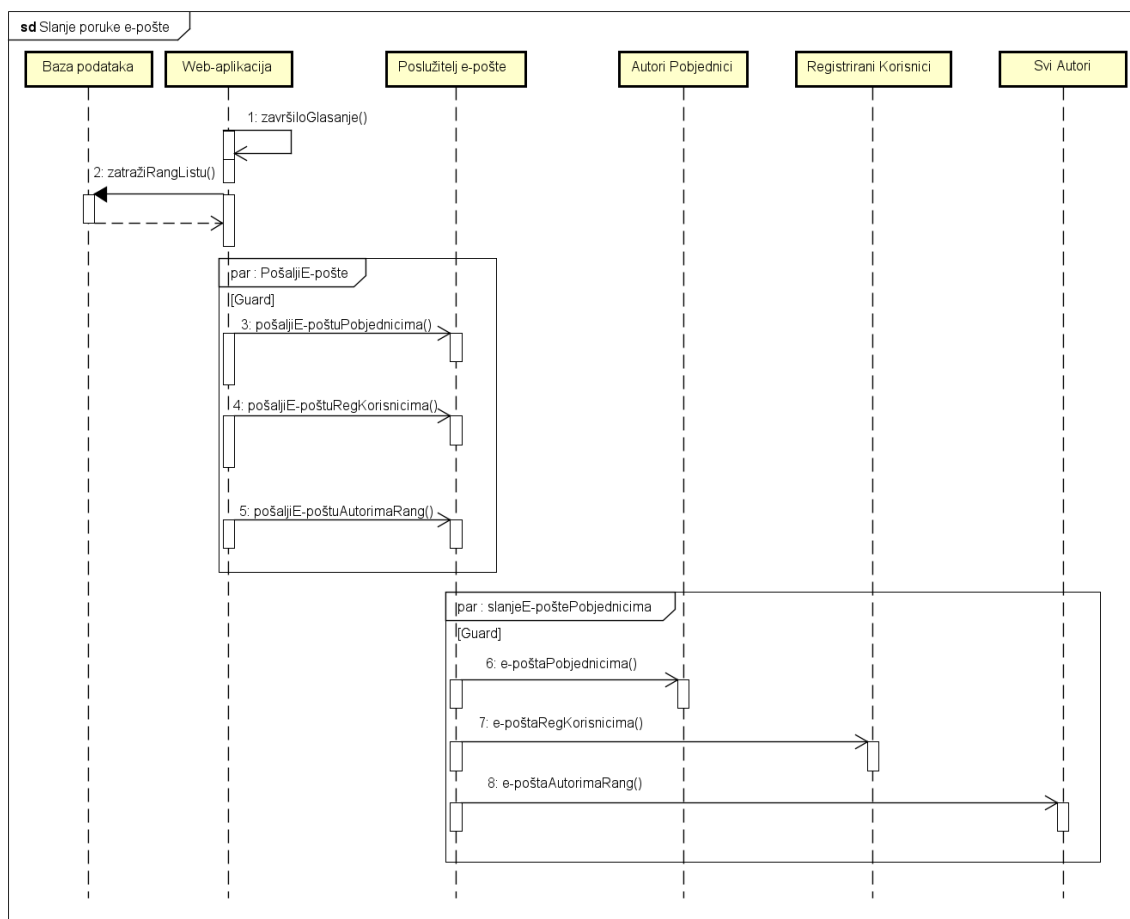
Administrator pritišće gumb za brisanje odabranog postera, na što aplikacija šalje zahtjev bazi da izbriše podatke o tom posteru. Baza odgovara potvrdom i izbrisani poster se miče iz otvorene galerije postera.



Slika 3.5: Sekvencijski dijagram za UC18 - Upravljanje posterima

## Obrazac uporabe UC24 - Slanje poruke e-pošte

Nakon što je glasovanje završeno šalju se tri vrste automatiziranih poruka e-pošte. Prvim trima autorima po broju glasova šalje se poziv na dodjelu nagrade. Svim autorima, bez obzira na njihov rang šalje se poruka o njihovom rang. Svim registriranim korisnicima šalje se obavijest o mjestu i vremenu dodjele nagrade.



Slika 3.6: Sekvencijski dijagram za UC31 - Slanje e-pošte

## 3.2 Ostali zahtjevi

- svaki korisnik može, ali ne mora dati najviše jedan glas
- glasovanje je moguće samo tijekom vremenskog razdoblja koje je predodređeno trajanjem konferencije
- sustav treba moći podržati rad više korisnika u stvarnom vremenu.
- korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati dulje od nekoliko sekundi
- sustav treba biti implementiran kao mrežna aplikacija služeći se objektno-orijentiranom paradigmom
- neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i ispravan rad sustava
- sustav treba biti jednostavan i intuitivan, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- potencijalna nadogradnja sustava ne smije narušiti njegovu postojeću funkcionalnost
- veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- pristup sustavu mora biti omogućen iz javne mreže protokolom HTTPS



## 4. Arhitektura i dizajn sustava

Arhitektura je podijeljena u tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

Internetski preglednik služi za pregled web-stranica i njihovog višemedijskog sadržaja, a one komuniciraju s poslužiteljima slanjem zahtjeva i primanjem odgovora. Preglednik ima sposobnost interpretacije koda kojim je pisana stranica u ljudski čitljiv oblik.

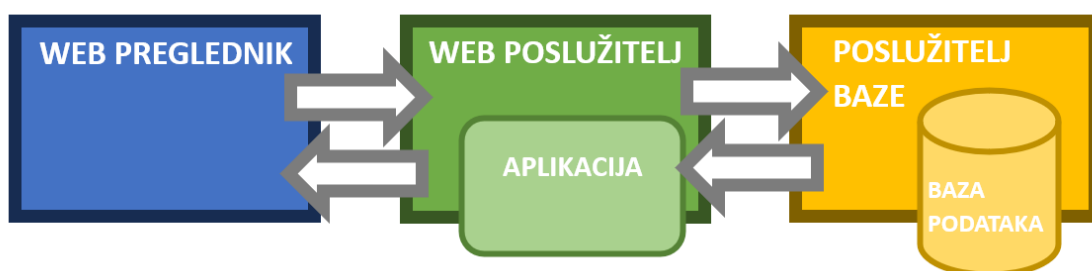
Web poslužitelj je posrednik između korisnika i aplikacije, te služi kao osnova rada aplikacije. Komunikacija se ostvaruje korisničkim HTTP (engl. *Hyper Text Transfer Protocol*) zahtjevima koji obično u zaglavlju imaju definiranu GET ili POST metodu za dohvaćanje odnosno predaju podataka. Poslužitelj na njih odgovara dostavom traženog sadržaja.

Web aplikacija po nalogu poslužitelja pristupa ili mijenja podatke iz baze podataka i vraća HTML dokument koji se potom prikazuje korisniku u sučelju preglednika. Arhitektura sustava će se temeljiti na stilističkoj varijaciji arhitekture zasnovane na događajima (engl. *event based architecture*) - MVC obrazcu. Spring - radni okvir korišten za razvoj pozadinskog dijela aplikacije, podržava MVC (Model-View-Controller) obrazac, te kao takav ima gotove predloške koji nam olakšavaju razvoj web aplikacije.

MVC obrazac omogućuje odvojen razvoj navedenih slojeva aplikacije što znatno olakšava ispitivanje, razvijanje i dodavanje novih svojstava u sustav.

Dijelovi MVC obrazca su:

- Model(*Model*) - rješava problem interakcije korisnika s bazom podataka, predstavlja bazu i služi za komunikaciju upravljača s bazom, te je na taj način zadužen za logiku vezanu za podatke i njihov prijenos
- Prikaz(*View*) - prikaz podataka i njihovih reprezentacija na korisničkom sučelju
- Upravljač(*Controller*) - povezuje komponente modela i prikaza i služi kao posrednik između tih komponenata i korisnika, a sam nije zadužen za obradu podataka, međudjeluje s modelom kako bi dohvatio podatke i s prikazom kako bi ih prikazao korisniku



Slika 4.1: Reprezentacija arhitekture sustava

## 4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava 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. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Konferencija
- Korisnik
- Poster
- Pokrovitelj
- Pokrovitelj-Konferencije
- Fotografija

### 4.1.1 Opis tablica

#### Konferencija

Ovaj entitet sadržava sve važne informacije o stručnoj konferenciji. Sadrži atribute: identifikator konferencije, ime konferencije, datum i vrijeme početka konferencije, datum i vrijeme završetka konferencije, mjesto održavanja konferencije, adresu održavanja konferencije, poštanski broj mjesta, generičko korisničko ime i pripadajuću lozinku koja se koristi za pristup konferenciji, poveznicu na video prijenos konferencije i oznaku o trajanju glasovanja. Ovaj entitet u vezi je *jedan na više* s entitetom Korisnik preko identifikatora konferencije, u vezi *jedan na više* s entitetom Poster preko identifikatora konferencije, u vezi *jedan na više* s entitetom Fotografija preko identifikatora konferencije i u vezi *više na više* s entitetom Pokrovitelj preko identifikatora konferencije.

Konferencija		
id_konferencija	INT	jedinstveni brojčani identifikator konferencije
ime_konferencija	VARCHAR	jedinstveno ime konferencije
datum_vrijeme_pocetka	TIMESTAMP	datum i vrijeme početka konferencije
datum_vrijeme_zavrsetka	TIMESTAMP	datum i vrijeme završetka konferencije
mjesto	VARCHAR	ime mjesta u kojem se održava konferencija
adresa	VARCHAR	adresa održavanja konferencije
zip_code	VARCHAR	poštanski broj mjesta održavanja konferencije
generic_username	VARCHAR	generičko korisničko ime koje koriste svi posjetitelji konferencije prije izrade vlastitog korisničkog računa
generic_password	VARCHAR	hash lozinka generičkog korisničkog računa
video_url	VARCHAR	poveznica na video prijenos konferencije

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Konferencija		
voting_re-minder_sent	BOOLEAN	oznaka pomoću koje aplikacija svakih 24 sata provjerava traje li glasanje za zadanu konferenciju

## Korisnik

Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži atribute: identifikator korisnika, e-mail adresu korisnika, lozinku, ime, prezime, oznaku ima li korisnik ovlasti administratora, oznaku radi li se o generičkom računu, oznaku je li korisnik glasao i identifikator konferencije na kojoj korisnik sudjeluje. Ovaj entitet je u vezi *više na jedan* s entitetom Konferencija preko identifikatora konferencije.

Korisnik		
id_korisnik	INT	jedinstveni brožčani identifikator korisnika
email	VARCHAR	e-mail adresa kojom korisnik izrađuje korisnički račun
lozinka	VARCHAR	hash lozinka korisničkog računa
ime	VARCHAR	ime korisnika
prezime	VARCHAR	prezime korisnika
admin	BOOLEAN	oznaka ima li ovlasti administratora
visitor	BOOLEAN	oznaka radi li se o generičkom računu (TRUE), kada posjetitelj napravi vlastiti korisnički račun vrijednost varijable postaje FALSE
voted	BOOLEAN	oznaka je li korisnik glasao
id_konferencija	INT	jedinstveni identifikator konferencije na kojoj korisnik sudjeluje (konferencija.id_konferencija)

**Poster**

Ovaj entitet sadržava sve važne informacije o radu prikazanom odgovarajućim posterom/prezentacijom. Sadrži attribute: identifikator postera, ime postera, datoteku postera, broj glasova za poster, ime i prezime autora, e-mail adresu kojom je autor poslao poster i identifikator konferencije na kojoj se rad izlaže. Ovaj entitet je u vezi *više na jedan* s entitetom Konferencija preko identifikatora konferencije.

<b>Poster</b>		
id_poster	INT	jedinstveni brojčani identifikator postera
ime_poster	VARCHAR	ime postera
file_path	LONGBLOB	PDF datoteka postera ili prezentacije
br_glasova	INT	broj glasova za određeni poster
ime_autor	VARCHAR	ime autora
prezime_autor	VARCHAR	prezime autora
poster_email	VARCHAR	e-mail adresa autora
id_konferencija	INT	jedinstveni brojčani identifikator konferencije kojoj poster pripada (konferencija.id_konferencija)

### Pokrovitelj

Ovaj entitet sadržava sve važne informacije o pokrovitelju. Sadrži attribute: identifikator pokrovitelja, ime pokrovitelja, promotivne materijale i poveznicu na stranicu pokrovitelja. Ovaj entitet je u vezi *više na više* s entitetom Konferencija.

Pokrovitelj		
id_pokrovitelj	INT	jedinstveni brojčani identifikator pokrovitelja
ime_pokrovitelja	VARCHAR	jedinstveni identifikator pokrovitelja
promotivni_materijal	LONGBLOB	datoteka, promotivni materijal je logotip pokrovitelja
url_promo	VARCHAR	poveznica na stranicu pokrovitelja

### Pokrovitelj-Konferencije

Ovaj entitet nastaje zbog *više na više* veze između entiteta Konferencija i Pokrovitelj. Sadržava sve informacije o odnosu konferencije i pokrovitelja odnosno kojoj konferenciji pokrovitelj pripada.

Pokrovitelj-Konferencije		
id_pokrovitelj	INT	jedinstveni brojčani identifikator pokrovitelja (pokrovitelj.id_pokrovitelj)
id_konferencija	INT	jedinstveni brojčani identifikator konferencije (konferencija.id_konferencija)

### Fotografija

Ovaj entitet sadržava sve informacije o fotografijama s konferencije. Sadrži attribute: identifikator fotografije, sliku fotografije i identifikator konferencije kojoj fotografija pripada. Ovaj entitet u vezi je *više na jedan* s entitetom Konferencija.

Fotografije		
id_fotografija	INT	jedinstveni brojčani identifikator fotografije
file_path	LONGBLOB	fotografije

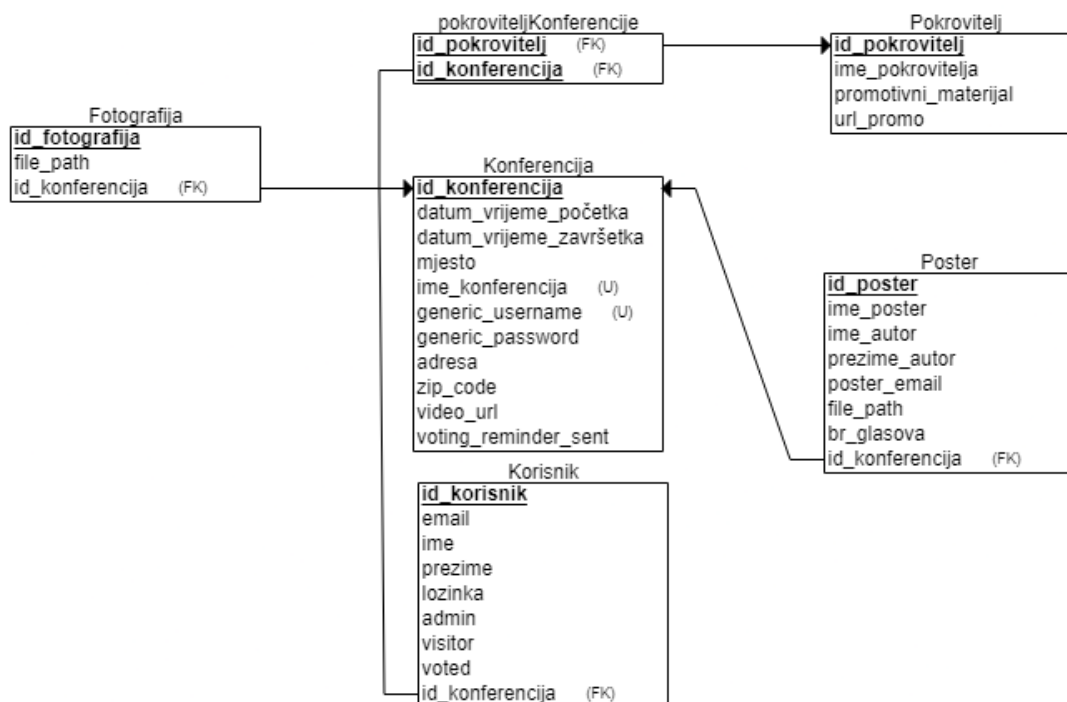
Nastavljeno na idućoj stranici



Nastavljeno od prethodne stranice

Fotografije		
id_konferencija	INT	jedinstveni brojčani identifikator konferencije kojoj fotografija pripada (konferencija.id_konferencija)

### 4.1.2 Dijagram baze podataka



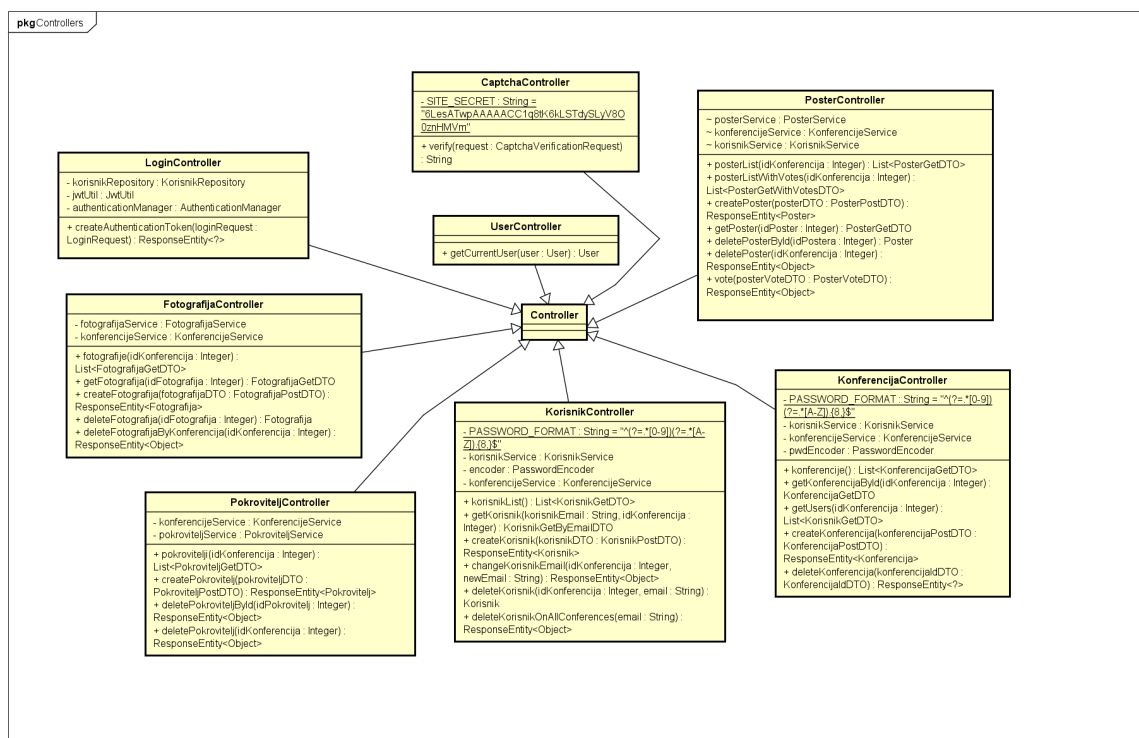
Slika 4.2: E-R dijagram baze podataka

## 4.2 Dijagram razreda

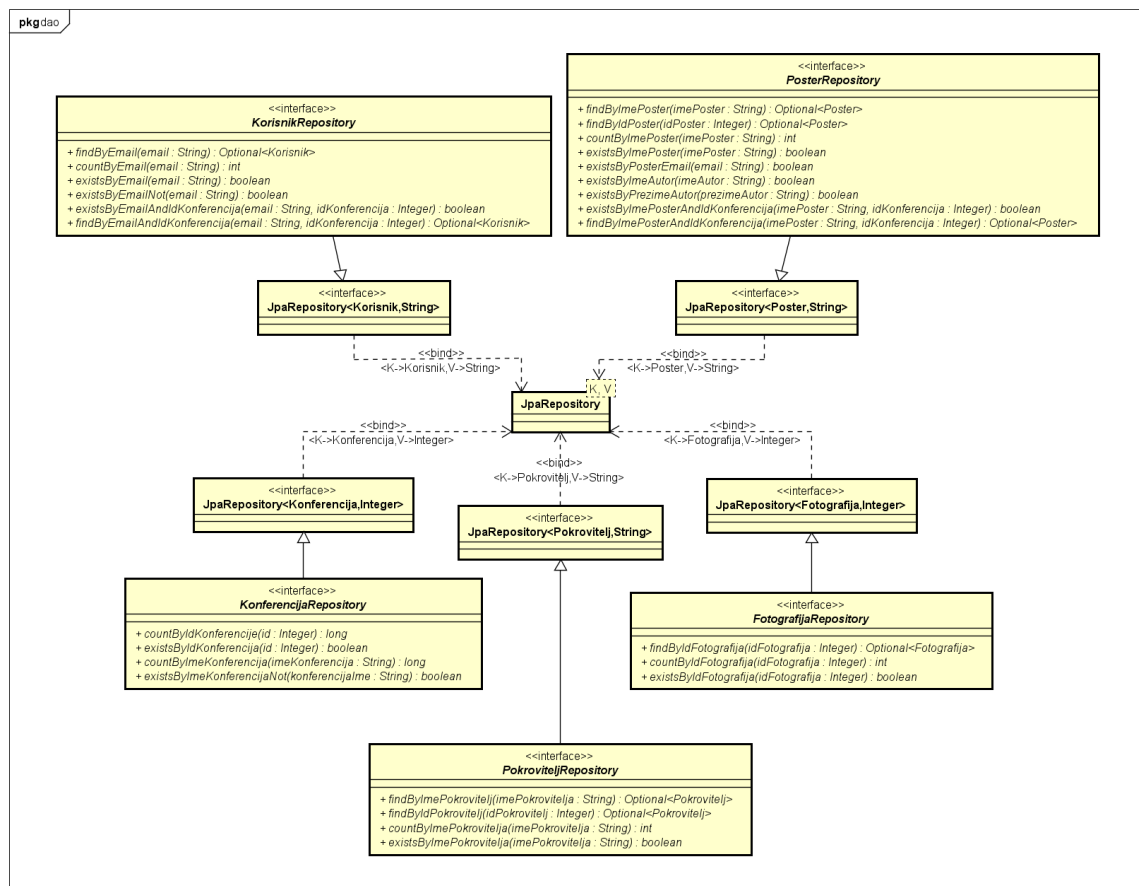
Slike 4.3, 4.4 i 4.5 prikazuju dijagrame razreda pozadinskog dijela MVC arhitekture. Slike 4.3 i 4.4 prikazuju upravljačke razrede, te DAO (*Data Access Object*) razrede.

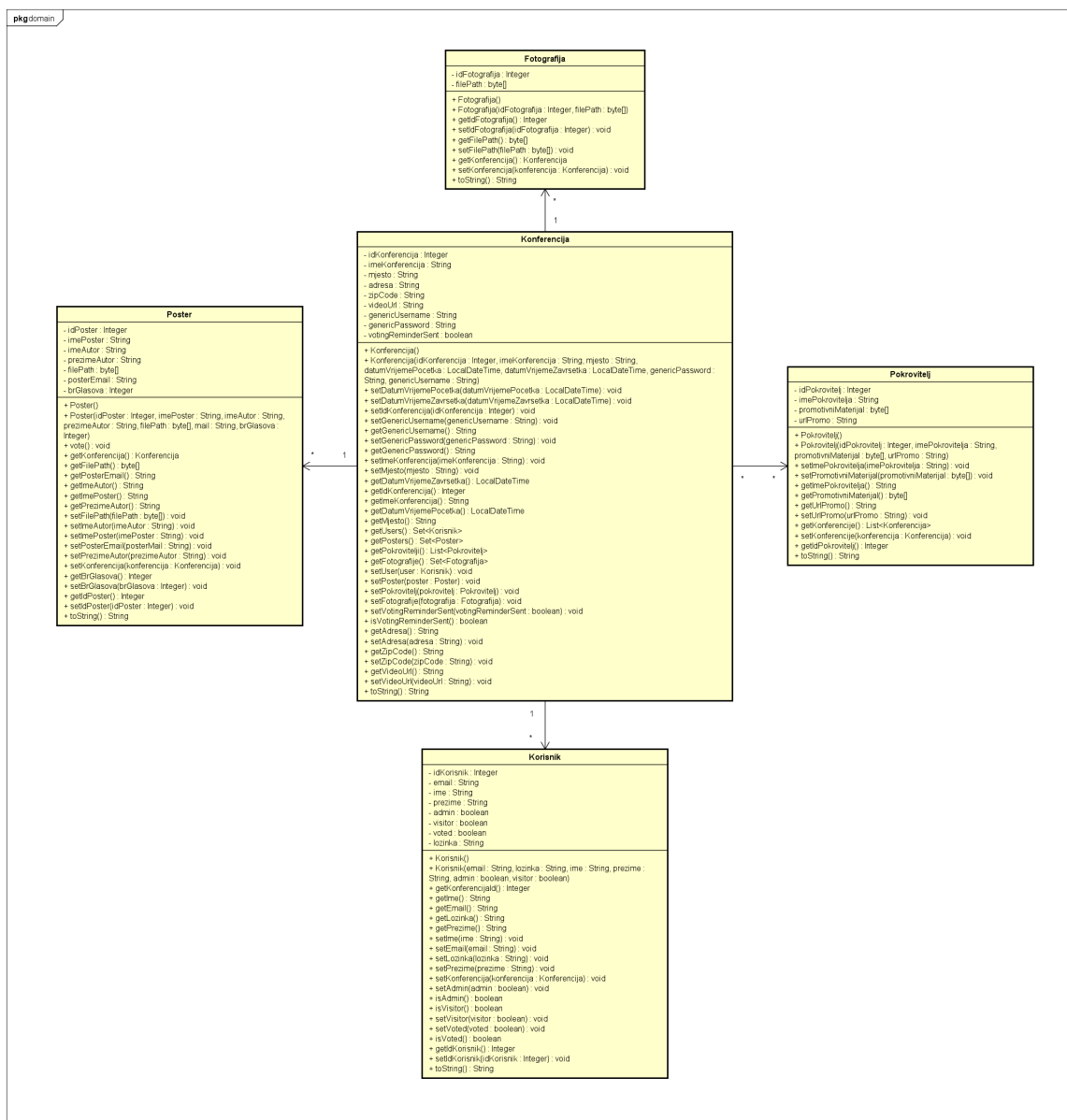
Dijagrami su zbog preglednosti podijeljeni po slojevima, odnosno na pojedinom dijagramu prikazane su samo ovisnosti između razreda istog MVC sloja, a prema nazivima samih razreda može se logički zaključiti koji su razredi različitih slojeva funkcionalno povezani.

Razredi dijagrama na slici 4.5 preslikavaju entitete i atribute baze podataka. Razred Korisnik predstavlja općeniti model korisnika aplikacije koji može biti posjetitelj konferencije ili administrator. Može se registrirati u sustav unoseći osnove informacije. Razred Konferencija predstavlja skup podataka koji su potrebni za registraciju konferencije i koji se prikazuju korisnicima. Razred Poster predstavlja skup podataka koji su potrebni za dodavanje postera i koji se prikazuju korisnicima. Razred Fotografija predstavlja skup podataka vezan uz pojedinačnu fotografiju konferencije. Razred Pokrovitelj predstavlja pokrovitelja konferencije.



Slika 4.3: Dijagram razreda - Upravljači





Slika 4.5: Dijagram razreda - Modeli

***dio 2. revizije***

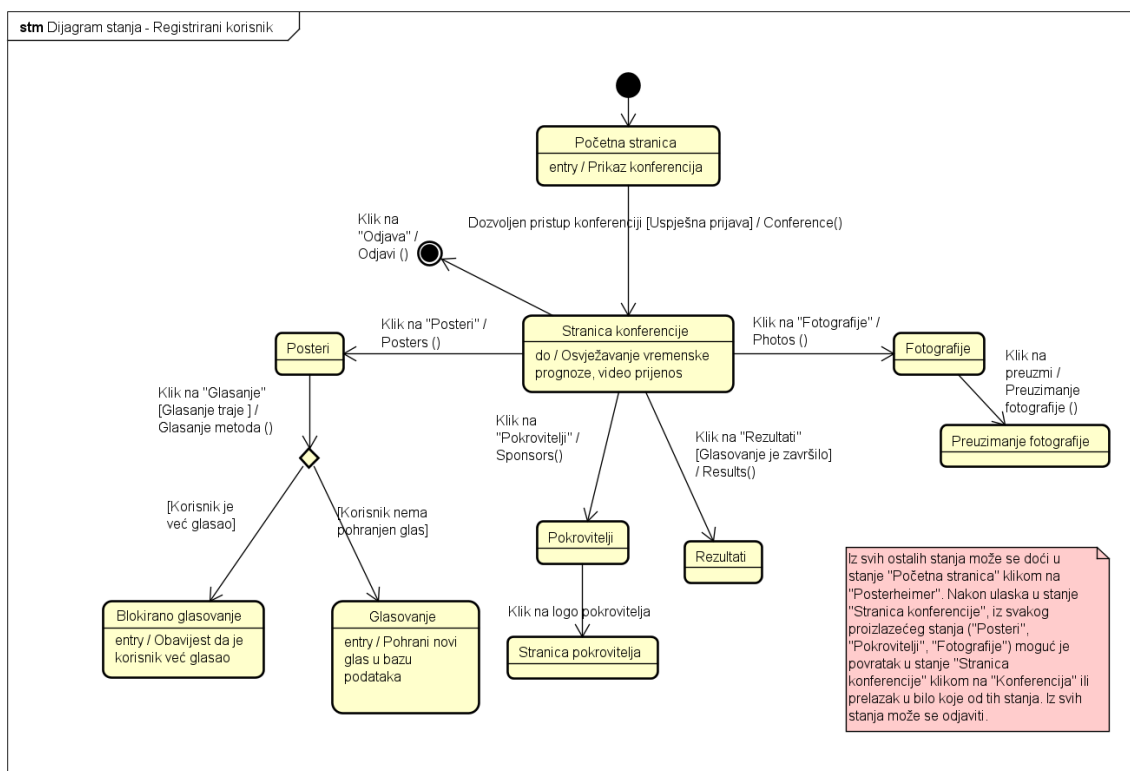
*Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije*

## 4.3 Dijagram stanja

### *dio 2. revizije*

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu **značajan dio** sustava, a registracija i prijava nisu.*

Dijagram stanja opisuje ponašanje sustava koristeći konačan broj apstraktnih diskretnih stanja u kojima se on može nalaziti i načine na koje se može prijeći iz jednog u drugo stanje, odnosno korisničke akcije koje potiču te prijelaze. Na slici 4.6 je prikazan dijagram stanja za registriranog korisnika koji je prije same registracije pristupio konferenciji pomoću generičke lozinke. Registriranom korisniku omogućen je pregled samo onih konferencija kojima je pristupio. Nakon uspješnog pristupa konferenciji, korisniku se prikazuje početna stranica konferencije na kojoj može pregledati vrijeme i mjesto događaja, vremensku prognozu za idućih 48 sati na toj lokaciji, te video prijenos konferencije, kao i popis pokrovitelja konferencije. Navigacijska traka korisniku omogućava pristup posterima, fotografijama, te pokroviteljima konferencije. Klikom na "Posteri" prikazuje mu se galerija postera gdje ima mogućnost glasanja za poster po izboru unutar zadannog vremenskog ograničenja. Klikom na "Fotografije" otvara se galerija fotografija, odakle se odabrane fotografije mogu preuzeti na računalo. Klikom na "Pokrovitelji", korisniku se prikazuju pokrovitelji konferencije. Klikom na logo određenog pokrovitelja otvoriti će se službena stranica pokrovitelja koja se nalazi izvan aplikacije. Nakon završetka glasanja korisnik će imati pristup rezultatima klikom na "Rezultati".



Slika 4.6: Dijagram stanja

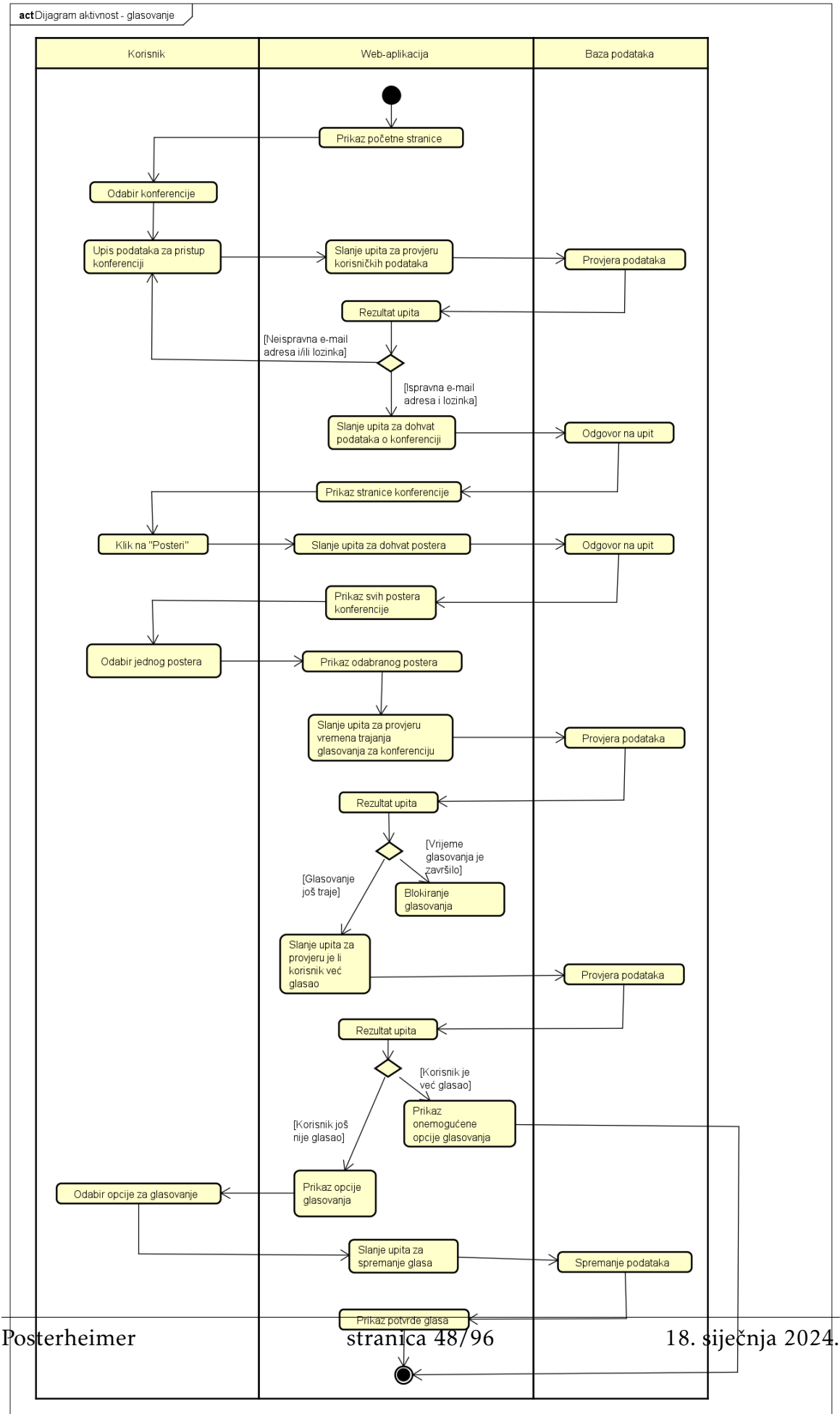
## 4.4 Dijagram aktivnosti

### dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

Dijagram aktivnosti opisuje ponašanje funkcionalnog dijela sustava, te interakciju više fizički ili programski odvojenih dijelova sustava, a posebno je pogodan za prikaz njihove sinkronizacije i konkurentnog djelovanja. Dijagram aktivnosti na slici 4.7 prikazuje tok podataka postupka glasovanja za poster. Korisnik pristupa aplikaciji i odabire konferenciju na kojoj sudjeluje. Klikom na opciju "Poster" korisniku će se prikazati galerija postera. Korisnik odabire jedan poster te za njega glasa.



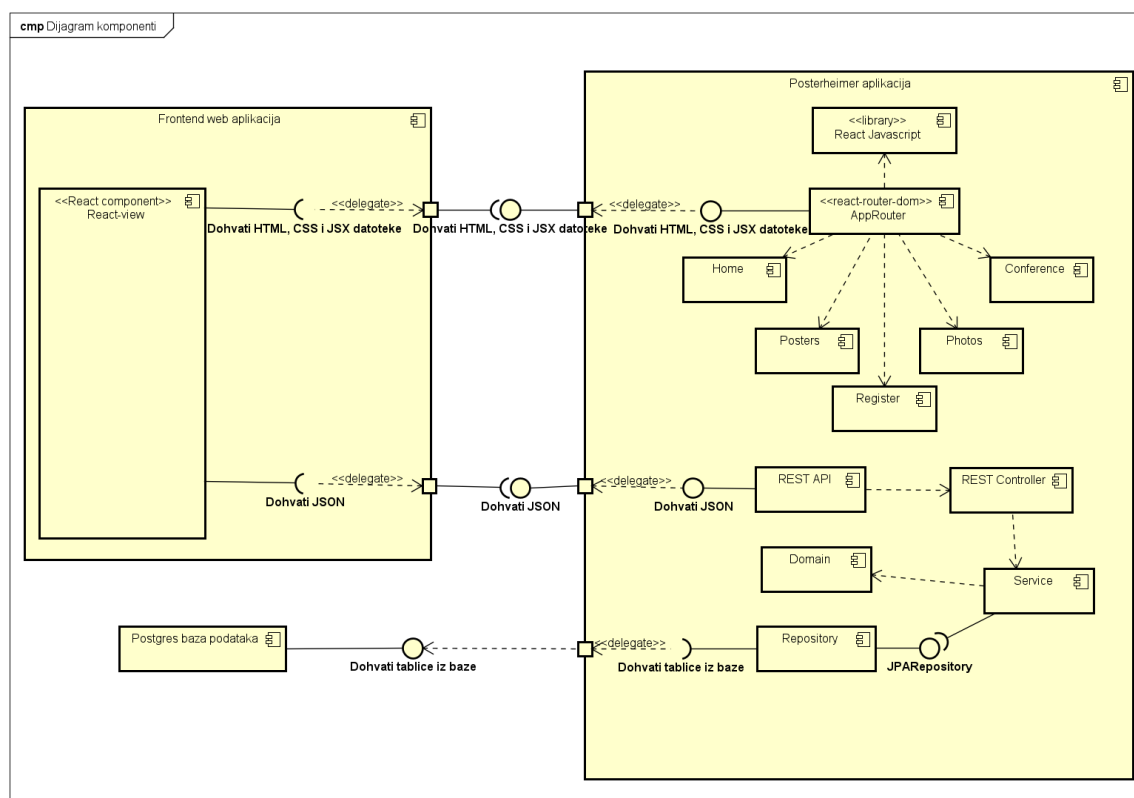


## 4.5 Dijagram komponenti

### *dio 2. revizije*

*Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.*

Dijagram komponenti prikazan na slici 4.8 vizualizira organizaciju programskih komponenti i cjelina, te njihovu međuovisnost i međusobnu interakciju preko nuđenih i traženih sučelja. Sustavu se pristupa preko dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JSX datoteka poslužuju se datoteke koje pripadaju prednjem *engl. frontend* dijelu aplikacije. AppRouter je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje. Prednji *engl. frontend* dio sastoji se od niza JavaScript XML datoteka koje su raspoređene u logičke cjeline nazvane po dijelu aplikacije za koji se koriste. Sve JavaScript XML datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju *backend* dijelu aplikacije. Repository je zadužen za dohvaćanje tablica iz baze podataka pomoću SQL upita. Podaci koji su pristigli iz baze se šalju dalje MVC arhitekturi u obliku DTO (*Data transfer object*). React-view komponenta preko dostupnih sučelja komunicira sa Posterheimer aplikacijom te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke.



Slika 4.8: Dijagram komponenti

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### *dio 2. revizije*

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

Pri izradi dokumentacije korišten je **LaTeX**<sup>1</sup> - označni jezik korišten za uređivanje tekstualnih dokumenata najčešće znanstvene publikacije. Dokument je sastavljen u programu **TeXstudio**<sup>2</sup> - uređivaču teksta prilagođen LaTeX-u. UML dijagrami nacrtani su uz pomoć alata **Astah UML**<sup>3</sup>. Dijagrami koji nisu UML tipa nacrtani su u uređivaču **MS Word**<sup>4</sup>

Udaljeni repozitorij projekta dostupan je na platformi **GitHub**<sup>5</sup>, korištenoj za pohranu svih datoteka potrebnih za rad na projektu.

Za izradu pozadinskog dijela aplikacije korišten je objektno orijentirani programski jezik **Java**<sup>6</sup> i radni okvir **Spring Boot**<sup>7</sup> - specijalizacija radnog okvira Spring, s ciljem jednostavnijeg i bržeg oblikovanja web aplikacije. Kao razvojno okruženje korišten je **IntelliJ IDEA**<sup>8</sup>. Tijekom izrade, pozadinski dio aplikacije testiran je pomoću platforme za testiranje API-ja **Postman**<sup>9</sup>.

Za izradu prednjeg dijela aplikacije korišten je **React**<sup>10</sup> i JavaScript ekstenzija **JavaScript XML**<sup>11</sup>. React, također poznat kao React.js ili ReactJS, je biblioteka u JavaScriptu za izgradnju korisničkih sučelja koju održava *Facebook*. React se najčešće

---

<sup>1</sup><https://www.latex-project.org>

<sup>2</sup><https://www.texstudio.org>

<sup>3</sup><https://astah.net/products/astah-uml>

<sup>4</sup><https://www.microsoft.com/en/microsoft-365/word>

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

<sup>6</sup><https://www.java.com/en>

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

<sup>8</sup><https://www.jetbrains.com/idea>

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

<sup>10</sup><https://react.dev>

<sup>11</sup><https://legacy.reactjs.org/docs/introducing-jsx.html>

koristi kao osnova u razvoju mrežnih ili mobilnih aplikacija. Složene aplikacije u *React*-u obično zahtijevaju korištenje dodatnih biblioteka za interakciju s API-jem.

Baza podataka izrađena je u sustavu za upravljanje bazama podataka **PostgreSQL**<sup>12</sup>. Za pristup PostgreSQL sustavu baze podataka korišten je **pgAdmin 4**<sup>13</sup> program. Za izradu ER dijagrama baze podataka korišten je **ERDPlus**<sup>14</sup> alat.

Aplikacija i baza podataka su postavljene na udaljene poslužitelje kao usluga aplikacije **Render**<sup>15</sup>, koja nudi besplatnu opciju pri iznajmljivanju sklopovlja, s ograničenim mogućnostima.

Komunikacija između članova tima ostvarena je putem aplikacija s tom svrhom: **WhatsApp**<sup>16</sup> i **Discord**<sup>17</sup>.

---

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

<sup>13</sup><https://www.pgadmin.org>

<sup>14</sup><https://erdplus.com>

<sup>15</sup><https://render.com>

<sup>16</sup><https://www.whatsapp.com>

<sup>17</sup><https://discord.com>

## 5.2 Ispitivanje programskog rješenja

### dio 2. revizije

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.*

### 5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

Ispitivanje komponenti provedeno je pomoću Selenium WebDrivera<sup>18</sup> unutar JUnit 4 testova kao podršku za pisanje ispita unutar programskog jezika Java. Ispitivanje je provedeno nad konferencijom naziva "TestKonferencija". U vrijeme izrade dokumentacije, konferencija se nalazila na drugom mjestu na listi doostupnih konferencija. Izrađeno je 7 ispitnih slučajeva. Uspješnost 6 slučajeva te izazivanje pogreške u jednom prikazni su slikom 5.1.

U **prvom ispitnom slučaju** provjerena je funkcionalnost gumba za pristup konferenciji pri incijalnom pristupu pomoću točnog generičkog korisničkog imena (email adrese predviđenu za generički račun) i odgovarajuće lozinke. Predviđen rezultat je dozvoljen pristup stranici konferencije. Ispitni slučaj je uspješan.

---

```
@Test
```

```
public void genericLogin() {
```

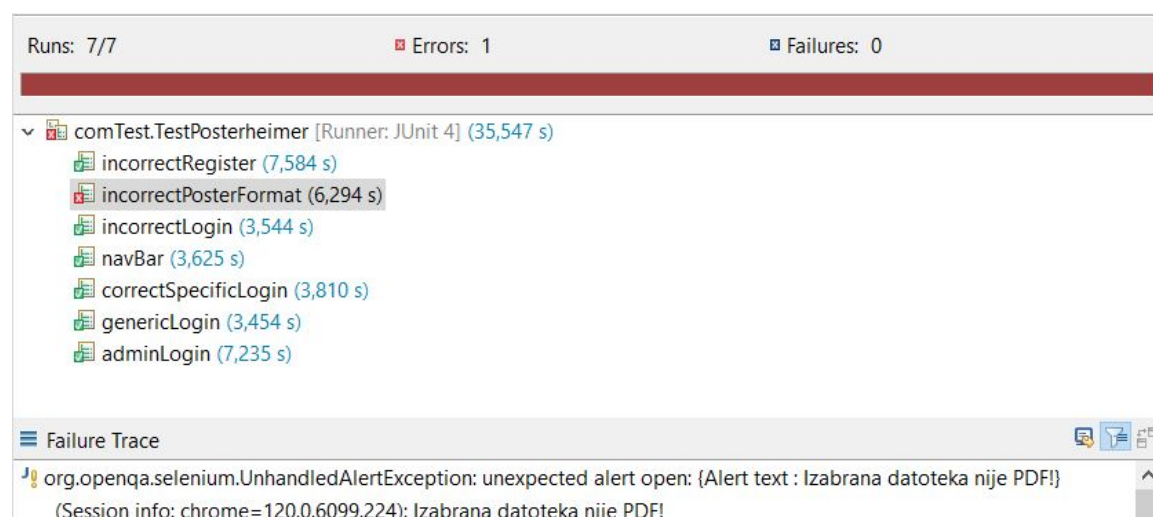
```
    System.setProperty("webdriver.chrome.driver",  
        "src\\test\\java\\chromedriver\\chromedriver.exe");
```

```
    WebDriver driver = new ChromeDriver();  
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```
    driver.get("https://posterheimer.onrender.com/");
```

---

<sup>18</sup><https://www.selenium.dev/documentation/webdriver/>



Slika 5.1: Prikaz uspješnosti ispitnih slučajeva

```
org.openqa.selenium.Dimension target = new
    org.openqa.selenium.Dimension(1552, 840);
driver.manage().window().setSize(target);

driver.findElement(By.xpath("(//button[contains(text(),'Pristupi')])[2]")).click();
// broj postaviti na mjesto konferencije u listi

driver.findElement(By.id("username")).click();
driver.findElement(By.id("username")).sendKeys("visitor.test@mail.hr");
driver.findElement(By.id("password")).click();
driver.findElement(By.id("password")).sendKeys("Lozinka1");
driver.findElement(By.cssSelector(".btn-primary")).click();

driver.findElement(By.linkText("Konferencija")).click();

String redirURL = driver.getCurrentUrl();
boolean comperRes = redirURL.contains("/conference");
if(comperRes == true) {
    System.out.println("Pristupljeno konferenciji");
} else {
    System.out.println("Error");
}
assertEquals(comperRes, true);
```

```
driver.quit();  
}
```

---

U **drugom ispitnom slučaju** provjerena je funkcionalnost gumba za pristup konferenciji pomoću jedinstvenog korisničkog računa te odjava. Potrebno je upisati točnu email adresu i odgovarajuću lozinku već stvorenog korisničkog računa. Nakon upisa podataka, korisnik dobiva pristup stranici konferencije. Nakon prijave, korisnik se odjavljuje. Predviđen rezultat je povratak na početnu stranicu aplikacije. Ispitni slučaj je uspješan.

---

@Test

```
public void correctSpecificLogin() {  
    System.setProperty("webdriver.chrome.driver",  
        "src\\test\\java\\chromedriver\\chromedriver.exe");  
    WebDriver driver = new ChromeDriver();  
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
  
    driver.get("https://posterheimer.onrender.com/");  
    driver.manage().window().setSize(new Dimension(1552, 840));  
    driver.findElement(By.cssSelector(".list-group-item:nth-child(4) >  
        .float-end")).click();  
    driver.findElement(By.id("username")).click();  
    driver.findElement(By.id("username")).sendKeys("register.email@mail.hr");  
    driver.findElement(By.id("password")).click();  
    driver.findElement(By.id("password")).sendKeys("Lozinka1");  
    driver.findElement(By.cssSelector(".btn-primary")).click();  
  
    driver.findElement(By.linkText("Konferencija")).click();  
  
    String redirURL = driver.getCurrentUrl();  
    boolean comperRes = redirURL.contains("/conference");  
    if(comperRes == true) {  
        System.out.println("Pristupljeno konferenciji");  
    } else {  
        System.out.println("Error: prijava");  
    }  
  
    driver.findElement(By.id("user-dropdown")).click();
```



```
driver.findElement(By.cssSelector(".fa-right-from-bracket")).click();

redirURL = driver.getCurrentUrl();
comperRes = redirURL.equals("https://posterheimer.onrender.com/");
if(comperRes == true) {
    System.out.println("Ostvarena odjava");
} else {
    System.out.println("Error: odjava");
}
driver.quit();
}
```

---

U **trećem ispitnom slučaju** provjerena je funkcionalnost gumba za pristup konferenciji pri neispravnoj prijavi. Predviđen rezultat je prikaz obavijesti "Pogrešan email ili lozinka!" te ostajanje na sučelju za prijavu unutar početne stranice. Ispitni slučaj je uspješan.

---

@Test

```
public void incorrectLogin() {
    System.setProperty("webdriver.chrome.driver",
        "src\\test\\java\\chromedriver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://posterheimer.onrender.com/");
    driver.manage().window().setSize(new Dimension(1552, 840));
    driver.findElement(By.xpath("//button[@type='button']")[2])).click();
    // vrijednost koju mijenjamo

    driver.findElement(By.cssSelector(".mb-3:nth-child(1)")).click();
    driver.findElement(By.id("username")).click();
    driver.findElement(By.id("username")).sendKeys("test");
    driver.findElement(By.id("password")).click();
    driver.findElement(By.id("password")).sendKeys("test");
    driver.findElement(By.cssSelector(".btn-primary")).click();

    driver.findElement(By.cssSelector(".alert")).click();
}
```

```

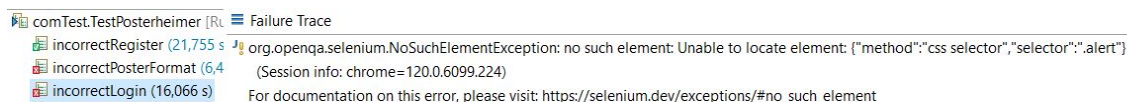
WebElement alert = driver.findElement(By.cssSelector(".alert"));
String errorMessage = alert.getText();
if(errorMessage.contains("email ili lozinka!")) {
    System.out.println("Prikaz prikladne obavijesti");
}

driver.findElement(By.cssSelector(".btn-close")).click();

String redirectURL = driver.getCurrentUrl();
boolean compareRes =
    redirectURL.equals("https://posterheimer.onrender.com/");
if(compareRes) {
    System.out.println("Ostvaren rezultat");
}
driver.quit();
}

```

Neispravnu prijavu ispitati ćemo i pomoću postojećeg korisničkog računa koji nije namijenjen odabranoj konferenciji. Upisat ćemo redni broj druge konferencije, kao *username* poslati "visitor.test@mail.hr" te "Lozinka1" za *password*. Predviđeni rezultat je ponovan prikaz obavijesti "Pogrešan email ili lozinka!". Ispitni slučaj nije uspješan - obavijest se ne prikazuje te je korisniku dozvoljen pristup konferenciji. Funkcionalnost bi u budućnosti trebalo implementirati.



Slika 5.2: Prikaz pada ispitnog slučaja

U **četvrtom ispitnom slučaju** provjerena je funkcionalnost gumba za pristup konferenciji pri prijavi administratora. Predviđen rezultat je dozvoljen pristup konferenciji te mogućnost dodavanja postera, fotografija i pokrovitelja. Ispitni slučaj je uspješan.

```

@Test
public void adminLogin() {
    System.setProperty("webdriver.chrome.driver",
        "src\\test\\java\\chromedriver\\chromedriver.exe");
}

```

```
WebDriver driver = new ChromeDriver();
driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
// Vrijeme povecano zbog učitavanja postera, koje je znatno sporije za
// administratora

driver.get("https://posterheimer.onrender.com/");
driver.manage().window().setSize(new Dimension(1552, 840));
driver.findElement(By.cssSelector(".list-group-item:nth-child(2) >
    .float-end")).click();
driver.findElement(By.id("username")).click();
driver.findElement(By.id("username")).sendKeys("admin.test@mail.hr");
driver.findElement(By.id("password")).click();
driver.findElement(By.id("password")).sendKeys("Lozinka1");
driver.findElement(By.cssSelector(".btn-primary")).click();
driver.findElement(By.cssSelector(".conference-content")).click();
driver.findElement(By.linkText("Posteri")).click();
driver.findElement(By.cssSelector(".fa-solid")).click();
driver.findElement(By.cssSelector(".btn-close")).click();
driver.findElement(By.linkText("Fotografije")).click();
driver.findElement(By.cssSelector(".image-container > img")).click();
driver.findElement(By.cssSelector(".modal-title")).click();
driver.findElement(By.cssSelector(".btn-close")).click();
driver.findElement(By.linkText("Pokrovitelji")).click();
driver.findElement(By.cssSelector(".fa-solid")).click();
driver.findElement(By.cssSelector(".modal-title")).click();
driver.findElement(By.cssSelector(".btn-close")).click();

String redirectURL = driver.getCurrentUrl();
boolean comperRes = redirectURL.contains("sponsors");
if(comperRes) {
    System.out.println("Ostvaren rezultat");
}
driver.quit();
}
```

---

U **petom ispitnom slučaju** provjerena je funkcionalnost navigacijske trake. Predviđen rezultat je otvaranje stranice konferencije nakon klika na "Konferencija", stranice postera nakon klika na "Posteri", stranice za fotografije nakon klika na "Fotogra-

fije" i stranice pokrovitelja nakon klika na "Pokrovitelji". Ispitni slučaj je uspješan.

```
@Test
public void navBar() {
    System.setProperty("webdriver.chrome.driver",
        "src\\test\\java\\chromedriver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://posterheimer.onrender.com/");
    org.openqa.selenium.Dimension target = new
        org.openqa.selenium.Dimension(1552, 840);
    driver.manage().window().setSize(target);

    driver.findElement(By.xpath("(//button[contains(text(),'Pristupi')])[2]").click();
    // broj postaviti na mjesto konferencije u listi

    driver.findElement(By.id("username")).click();
    driver.findElement(By.id("username")).sendKeys("visitor.test@mail.hr");
    driver.findElement(By.id("password")).click();
    driver.findElement(By.id("password")).sendKeys("Lozinka1");
    driver.findElement(By.cssSelector(".btn-primary")).click();

    driver.findElement(By.linkText("Konferencija")).click();
    String redirURL = driver.getCurrentUrl();
    boolean comperRes = redirURL.contains("/conference");
    if(comperRes == true) {
        System.out.println("Pristupljeno konferenciji");
    } else {
        System.out.println("Error: konferencije");
    }
}

driver.findElement(By.linkText("Posteri")).click();
redirURL = driver.getCurrentUrl();
comperRes = redirURL.contains("/posters");
if(comperRes == true) {
    System.out.println("Pristupljeno posterima");
} else {
```

```
        System.out.println("Error: posteri");
    }

    driver.findElement(By.linkText("Fotografije")).click();
    redirURL = driver.getCurrentUrl();
    comperRes = redirURL.contains("/photos");
    if(comperRes == true) {
        System.out.println("Pristupljeno fotografijama");
    } else {
        System.out.println("Error: fotografije");
    }
    driver.findElement(By.linkText("Pokrovitelji")).click();
    redirURL = driver.getCurrentUrl();
    comperRes = redirURL.contains("/sponsors");
    if(comperRes == true) {
        System.out.println("Pristupljeno pokroviteljima");
    } else {
        System.out.println("Error: pokrovitelji");
    }
    driver.quit();
}
```

---

U šestom ispitnom slučaju provjerena je funkcionalnost gumba za registraciju pri unosu email adrese netočnog formata. Sva ostala polja su točno ispunjena te je prođen reCAPTCHA test. Predviđen rezultat je neuspješna registracija te obavijest o netočnom unosu. Ispitni slučaj je uspješan. Ispitni slučaj je neuspješan u slučaju CAPTCHA Image testa

---

```
@Test
public void incorrectRegister() {
    System.setProperty("webdriver.chrome.driver",
        "src\\test\\java\\chromedriver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    JavascriptExecutor js = (JavascriptExecutor) driver;
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://posterheimer.onrender.com/");
    driver.manage().window().setSize(new Dimension(1552, 840));
}
```

```
driver.findElement(By.cssSelector(".list-group-item:nth-child(2) >
    .float-end")).click();
driver.findElement(By.cssSelector("form")).click();
driver.findElement(By.id("username")).click();
driver.findElement(By.id("username")).sendKeys("visitor.test@mail.hr");
driver.findElement(By.id("password")).click();
driver.findElement(By.id("password")).sendKeys("Lozinka1");
driver.findElement(By.cssSelector(".btn-primary")).click();
driver.findElement(By.linkText("Registracija")).click();
js.executeScript("window.scrollTo(0,0)");
driver.findElement(By.id("formIme")).click();
driver.findElement(By.id("formIme")).sendKeys("TestIme");
driver.findElement(By.id("formPrezime")).click();
driver.findElement(By.id("formPrezime")).sendKeys("TestPrezime");
driver.findElement(By.id("formBasicEmail")).click();
driver.findElement(By.id("formBasicEmail")).sendKeys("email");
driver.findElement(By.id("formBasicPassword")).click();
driver.findElement(By.id("formBasicPassword")).sendKeys("lozinka");
driver.findElement(By.cssSelector(".mb-3:nth-child(2) >
    #formBasicPassword")).click();
driver.findElement(By.cssSelector(".mb-3:nth-child(2) >
    #formBasicPassword")).sendKeys("lozinka");
driver.switchTo().frame(0);
driver.findElement(By.cssSelector(".recaptcha-checkbox-border")).click();
driver.switchTo().defaultContent();
driver.findElement(By.cssSelector(".ml-2")).click();
driver.findElement(By.cssSelector(".mx-2 >
    .mb-3:nth-child(1)")).click();
driver.findElement(By.cssSelector(".mx-2 >
    .mb-3:nth-child(1)")).click();

String redirURL = driver.getCurrentUrl();
boolean comperRes = redirURL.contains("/register");
if(comperRes == true) {
    System.out.println("Neuspjesna registracija");
} else {
    System.out.println("Error");
}
```

```
        driver.quit();  
    }
```

---

U **sedmon ispitnom slučaju** provjerena je funkcionalnost sučelja za dodavanje postera pri unosu datoteke netočnog formata. Prije dodavanje postera potrebna je prijava administratora. Predviđen rezultat je prikaz obavijesti "Izabrana datoteka nije PDF!". Rezultat je ostvaren.

---

@Test

```
public void incorrectPosterFormat() {  
    System.setProperty("webdriver.chrome.driver",  
        "src\\test\\java\\chromedriver\\chromedriver.exe");  
    WebDriver driver = new ChromeDriver();  
    JavascriptExecutor js = (JavascriptExecutor) driver;  
    File filepath = new File("src\\test\\java\\files\\example.pptx");  
  
    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);  
  
    driver.get("https://posterheimer.onrender.com/");  
    driver.manage().window().setSize(new Dimension(1552, 840));  
    driver.findElement(By.cssSelector(".list-group-item:nth-child(2) >  
        .float-end")).click();  
  
    driver.findElement(By.id("username")).click();  
    driver.findElement(By.id("username")).sendKeys("admin.test@mail.hr");  
    driver.findElement(By.id("password")).click();  
    driver.findElement(By.id("password")).sendKeys("Lozinka1");  
    driver.findElement(By.cssSelector(".btn-primary")).click();  
    driver.findElement(By.linkText("Posteri")).click();  
    driver.findElement(By.cssSelector(".fa-solid")).click();  
  
    WebElement fileInput =  
        driver.findElement(By.cssSelector("input[type=file]"));  
    fileInput.sendKeys(filepath.getAbsolutePath());  
}
```

```
driver.findElement(By.name("imePoster")).click();
driver.findElement(By.name("imePoster")).sendKeys("Krivi format");
driver.findElement(By.name("imeAutor")).click();
driver.findElement(By.name("imeAutor")).sendKeys("Ime");
driver.findElement(By.name("prezimeAutor")).click();
driver.findElement(By.name("prezimeAutor")).sendKeys("Prezime");
driver.findElement(By.cssSelector("form")).click();
driver.findElement(By.name("posterEmail")).click();
driver.findElement(By.name("posterEmail")).sendKeys("test.poster@posterheimer.hr");
driver.findElement(By.cssSelector(".btn")).click();

WebElement alert = driver.findElement(By.cssSelector(".alert"));
String errorMessage = alert.getText();
if(errorMessage.contains("Izabrana datoteka nije PDF!")) {
    System.out.println("Prikaz prikladne obavijesti");
} else {
    System.out.println("Error");
}

driver.findElement(By.cssSelector(".alert")).click();
driver.findElement(By.cssSelector(".btn-close")).click();

driver.quit();
}
```

---

### 5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>19</sup>. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.

*Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću*

---

<sup>19</sup><https://www.seleniumhq.org/>



jednog od sljedeća dva alata:

- dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita
- **Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

Ispitivanje sustava provedeno je pomoću dodatka za preglednik Selenium IDE<sup>20</sup>. JUnit ispitni slučajevi generirani su pomoću Selenium IDE ispitnih slučajeva.

**Prvi ispitni slučaj** provjerava funkcionalnost brisanja korisnika. Prije brisanja korisnika potrebna je prijava registratora. Administrator briše određenog korisnika te se odjavljuje. Uspješnost brisanja korisnika provjeravamo pokušajem prijave izbrisanog korisnika. Očekivani rezultat je neuspješna prijava. Ispitni slučaj je uspješan.

Log	Reference
19. click on linkText=Odjava OK	17:56:58
20. click on css= list-group-item:nth-child(5) > .float-end OK	17:56:59
21. mouseOver on css= list-group-item:nth-child(5) > .float-end OK	17:57:00
22. mouseOut on css= light > .float-end OK	17:57:01
23. click on id=username OK	17:57:02
24. type on id=username with value register2 OK	17:57:03
25. type on id=username with value register2_email@mail.hr OK	17:57:04
26. click on id=password OK	17:57:05
27. type on id=password with value pass OK	17:57:06
28. click on css= btn-primary OK	17:57:07
'DeleteUser' completed successfully	17:57:08

Slika 5.3: Prikaz uspješnosti ispitnog slučaja

@Test

```
public void deleteUser() {
    driver.get("https://posterheimer.onrender.com/");
    driver.manage().window().setSize(new Dimension(1552, 840));
    driver.findElement(By.cssSelector(".list-group-item:nth-child(5) >
        .float-end")).click();
    {
        WebElement element =
            driver.findElement(By.cssSelector(".list-group-item:nth-child(5)
                > .float-end"));
        Actions builder = new Actions(driver);
```

<sup>20</sup><https://www.selenium.dev/selenium-ide/>

```
        builder.moveToElement(element).perform();
    }
    {
        WebElement element = driver.findElement(By.tagName("body"));
        Actions builder = new Actions(driver);
        builder.moveToElement(element, 0, 0).perform();
    }
    driver.findElement(By.cssSelector(".mb-3:nth-child(1)")).click();
    driver.findElement(By.id("username")).click();
    driver.findElement(By.id("username")).sendKeys("admin.test@mail.hr");
    driver.findElement(By.cssSelector("form")).click();
    driver.findElement(By.id("password")).click();
    driver.findElement(By.id("password")).sendKeys("pass");
    driver.findElement(By.cssSelector(".btn-primary")).click();
    driver.findElement(By.id("user-dropdown")).click();
    driver.findElement(By.linkText("Korisnici")).click();
    driver.findElement(By.cssSelector(".list-group-item:nth-child(16) >
        .float-start")).click();
    driver.findElement(By.id("root")).click();
    driver.findElement(By.cssSelector(".list-group-item:nth-child(18) >
        .btn")).click();
    driver.findElement(By.id("user-dropdown")).click();
    driver.findElement(By.linkText("Odjava")).click();
    driver.findElement(By.cssSelector(".list-group-item:nth-child(5) >
        .float-end")).click();
    {
        WebElement element =
            driver.findElement(By.cssSelector(".list-group-item:nth-child(5)
                > .float-end"));
        Actions builder = new Actions(driver);
        builder.moveToElement(element).perform();
    }
    {
        WebElement element = driver.findElement(By.tagName("body"));
        Actions builder = new Actions(driver);
        builder.moveToElement(element, 0, 0).perform();
    }
    driver.findElement(By.id("username")).click();
```

```
driver.findElement(By.id("username")).sendKeys("register2.");
driver.findElement(By.id("username")).sendKeys("register2.email@mail.hr");
driver.findElement(By.id("password")).click();
driver.findElement(By.id("password")).sendKeys("pass");
driver.findElement(By.cssSelector(".btn-primary")).click();
}
```

**Drugi ispitni slučaj** provjerava funkcionalnost gumba za registraciju pri unosu podataka točnog formata. Tijekom registracije odabrana je opcija "Prikaz lozinke" te prođen CAPTCHA test. Rezultat ispitnog slučaja je uspješan.

Log	Reference
29. click on css= recaptcha-checkbox-border OK	17:48:34
30. selectFrame on relative=parent OK	17:48:35
31. click on css= ml-2 OK	17:48:36
32. assertAlert on Form submission successful OK	17:48:37
33. runScript on window.scrollTo(0,0) OK	17:48:40
34. click on css=h1 OK	17:48:41
35. click on css=h1 OK	17:48:42
36. doubleClick on css=h1 OK	17:48:43
37. click on css=h1 OK	17:48:44
38. click on css=h1 OK	17:48:44
"Register" completed successfully	

Slika 5.4: Prikaz uspješnosti ispitnog slučaja

@Test

```
public void register() {
    driver.get("https://posterheimer.onrender.com/");
    driver.manage().window().setSize(new Dimension(1552, 840));
    driver.findElement(By.cssSelector(".list-group-item:nth-child(5) >
        .float-end")).click();
    driver.findElement(By.id("username")).click();
    driver.findElement(By.id("username")).sendKeys("visitor.test@mail.hr");
    driver.findElement(By.id("password")).click();
    driver.findElement(By.id("password")).sendKeys("pass");
    driver.findElement(By.cssSelector(".btn-primary")).click();
    driver.findElement(By.cssSelector(".conference-content")).click();
    driver.findElement(By.cssSelector(".mx-auto:nth-child(1)")).click();
    driver.findElement(By.cssSelector(".card-text")).click();
    driver.findElement(By.linkText("Registracija")).click();
    js.executeScript("window.scrollTo(0,0)");
    driver.findElement(By.id("formIme")).click();
    driver.findElement(By.id("formIme")).sendKeys("TestIme");
    driver.findElement(By.id("formPrezime")).click();
}
```

```
driver.findElement(By.id("formPrezime")).sendKeys("TestPrezime");
driver.findElement(By.id("formBasicEmail")).click();
driver.findElement(By.id("formBasicEmail")).click();
driver.findElement(By.id("formBasicEmail")).sendKeys("register.email@mail.hr");
// promijeniti email za svaki test
driver.findElement(By.id("formBasicPassword")).click();
driver.findElement(By.id("formBasicPassword")).sendKeys("pass");
driver.findElement(By.cssSelector(".mb-3:nth-child(2) >
    #formBasicPassword")).click();
driver.findElement(By.cssSelector(".mb-3:nth-child(2) >
    #formBasicPassword")).sendKeys("pass");
driver.findElement(By.cssSelector(".form-check")).click();
driver.findElement(By.id("formShowPassword")).click();
driver.findElement(By.id("formShowPassword")).click();
driver.switchTo().frame(0);
driver.findElement(By.cssSelector(".recaptcha-checkbox-border")).click();
driver.switchTo().defaultContent();
driver.findElement(By.cssSelector(".ml-2")).click();
assertThat(driver.switchTo().alert().getText(), is("Form submission
    successful!"));
js.executeScript("window.scrollTo(0,0)");
driver.findElement(By.cssSelector("h1")).click();
driver.findElement(By.cssSelector("h1")).click();
{
    WebElement element = driver.findElement(By.cssSelector("h1"));
    Actions builder = new Actions(driver);
    builder.doubleClick(element).perform();
}
driver.findElement(By.cssSelector("h1")).click();
driver.findElement(By.cssSelector("h1")).click();
}
```

**Treći ispitni slučaj** provjerava funkcionalnost gumba za dodavanje konferencije. Prije dodavanja konferencije potrebna je prijava natkorisnika. Podaci vezani za korisnički račun natkorisnika uklonjeni su iz izvornog koda. Predviđen rezultat je stvaranje nove konferencije. Ispitni slučaj je uspješan.

@Test

Log	Reference
28. type on id=formBasicEmail with value visitor.test@mail.hr OK	18.05.03
29. click on id=formBasicPassword OK	18.05.03
30. type on id=formBasicPassword with value pass OK	18.05.03
31. click on css=mt-2 > .mb-3.rnh-child(2) OK	18.05.03
32. click on name=adminUsername OK	18.05.03
33. type on name=adminUsername with value admin.test@mail.hr OK	18.05.03
34. click on name=adminPassword OK	18.05.04
35. type on name=adminPassword with value pass OK	18.05.04
36. click on css=mt-2 OK	18.05.04
37. runScript on window.scrollTo(0,0) OK	18.05.04
'AddConferention' completed successfully	

Slika 5.5: Prikaz uspješnosti ispitnog slučaja

```
public void addConferention() {  
    driver.get("https://posterheimer.onrender.com/");  
    driver.manage().window().setSize(new Dimension(1552, 840));  
    driver.findElement(By.cssSelector(".fa-solid")).click();  
    driver.findElement(By.id("username")).click();  
    driver.findElement(By.id("username")).sendKeys("ime natkorisnika");  
    driver.findElement(By.id("password")).click();  
    driver.findElement(By.id("password")).sendKeys("lozinka  
        natkorisnika");  
    driver.findElement(By.cssSelector(".btn-primary")).click();  
    driver.findElement(By.cssSelector(".fa-square-plus")).click();  
    driver.findElement(By.id("formConferenceName")).click();  
    driver.findElement(By.id("formConferenceName")).sendKeys("AutomaticTestKonferencija");  
    driver.findElement(By.name("videoUrl")).click();  
    driver.findElement(By.name("videoUrl")).sendKeys("https://www.youtube.com/watch?v=ScM");  
    driver.findElement(By.id("formConferenceCity")).click();  
    driver.findElement(By.id("formConferenceCity")).sendKeys("Unska ");  
    driver.findElement(By.id("formConferenceCity")).sendKeys("Unska 3");  
    driver.findElement(By.id("formConferenceLocation")).click();  
    driver.findElement(By.id("formConferenceLocation")).sendKeys("Zagreb");  
    driver.findElement(By.id("formConferenceZipCode")).click();  
    driver.findElement(By.id("formConferenceZipCode")).sendKeys("10000");  
    driver.findElement(By.id("formDateStart")).click();  
    driver.findElement(By.id("formDateStart")).sendKeys("2024-01-11T09:13");  
    driver.findElement(By.id("formDateEnd")).click();  
    driver.findElement(By.id("formDateEnd")).sendKeys("2024-01-14T09:13");  
    driver.findElement(By.id("root")).click();  
    driver.findElement(By.id("formBasicEmail")).click();  
    driver.findElement(By.id("formBasicEmail")).sendKeys("visitor");  
    driver.findElement(By.id("formBasicEmail")).sendKeys("visitor.test@mail.hr");  
}
```

```

driver.findElement(By.id("formBasicPassword")).click();
driver.findElement(By.id("formBasicPassword")).sendKeys("pass");
driver.findElement(By.cssSelector(".mt-2 >
    .mb-3:nth-child(2)")).click();
driver.findElement(By.name("adminUsername")).click();
driver.findElement(By.name("adminUsername")).sendKeys("admin.test@mail.hr");
driver.findElement(By.name("adminPassword")).click();
driver.findElement(By.name("adminPassword")).sendKeys("pass");
driver.findElement(By.cssSelector(".ml-2")).click();
js.executeScript("window.scrollTo(0,0)");
}

```

**Četvrti ispitni slučaj** provjerava funkcionalnost glasovanja. Prije samog glasovanja potrebna je prijava registriranog korisnika te odlazak na postere. Korisnik odabire jedan od postera, glasuje za njega te potvrđuje svoj glas. Zatim otvara drugi poster te više nema mogućnost glasovanja. Ispitni slučaj je uspješan.

Log	Reference	
15. click on css= poster-grid-element:nth-child(2) .fa-solid OK		22:52:15
16. click on css= modal-footer OK		22:52:16
17. click on css= modal-footer OK		22:52:17
18. doubleClick on css= modal-footer OK		22:52:18
19. click on css= modal-footer OK		22:52:19
20. click on css= modal-footer OK		22:52:20
21. click on css= modal-header > btn-close OK		22:52:21
"Voting" completed successfully		22:52:21

Slika 5.6: Prikaz uspješnosti ispitnog slučaja

@Test

```

public void voting() {
    driver.get("https://posterheimer.onrender.com/");
    driver.manage().window().setSize(new Dimension(1536, 824));
    driver.findElement(By.cssSelector(".list-group-item:nth-child(4) >
        .float-end")).click();
    driver.findElement(By.cssSelector("form")).click();
    driver.findElement(By.id("username")).click();
    driver.findElement(By.id("username")).sendKeys("register2.email@mail.hr");
    driver.findElement(By.id("password")).click();
    driver.findElement(By.id("password")).sendKeys("Lozinka1");
    driver.findElement(By.cssSelector(".btn-primary")).click();
    driver.findElement(By.linkText("Posteri")).click();
    driver.findElement(By.cssSelector(".poster-grid-element:nth-child(1)

```

```
        .fa-solid")).click();
driver.findElement(By.cssSelector(".btn-success")).click();
driver.findElement(By.cssSelector(".mx-2")).click();
driver.findElement(By.cssSelector(".modal-header >
    .btn-close")).click();
driver.findElement(By.cssSelector(".poster-grid-element:nth-child(2)
    .fa-solid")).click();
driver.findElement(By.cssSelector(".modal-footer")).click();
driver.findElement(By.cssSelector(".modal-footer")).click();
{
    WebElement element =
        driver.findElement(By.cssSelector(".modal-footer"));
    Actions builder = new Actions(driver);
    builder.doubleClick(element).perform();
}
driver.findElement(By.cssSelector(".modal-footer")).click();
driver.findElement(By.cssSelector(".modal-footer")).click();
driver.findElement(By.cssSelector(".modal-header >
    .btn-close")).click();
}
```

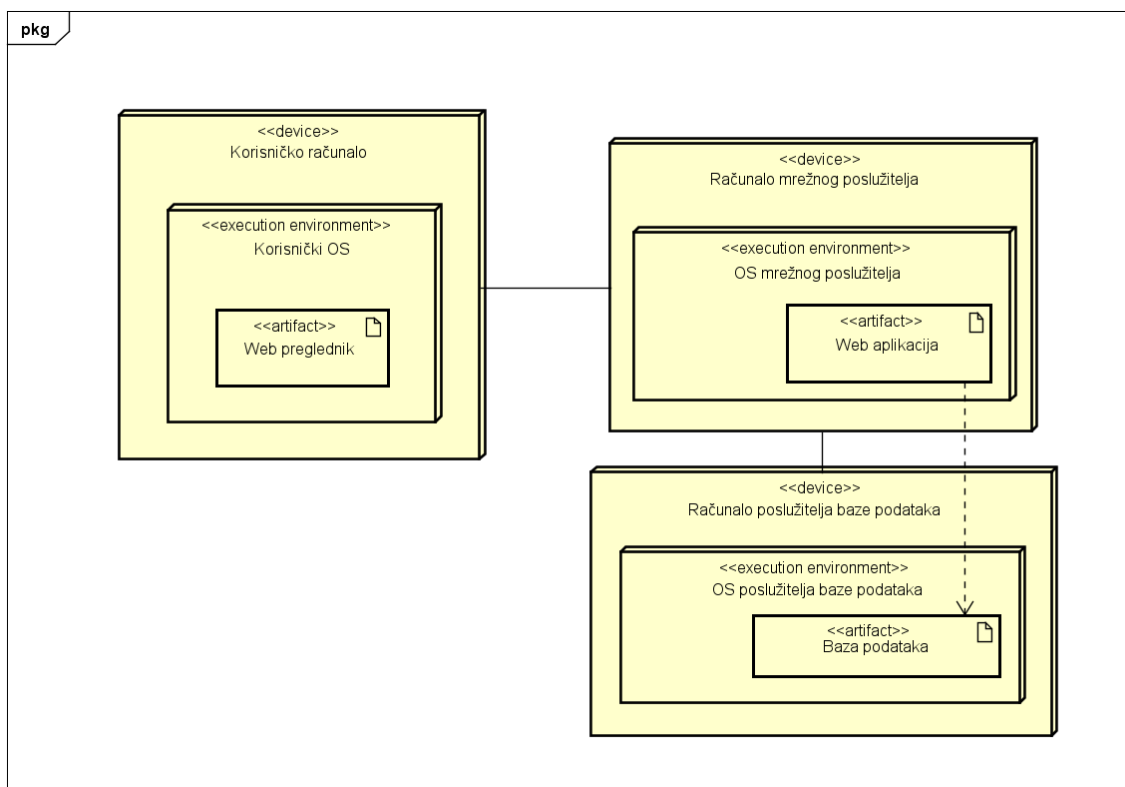
---

## 5.3 Dijagram razmještaja

### dio 2. revizije

Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.

Dijagram razmještaja prikazuje odnos sklopovskih dijelova sustava međusobno i s programskim rješenjima koja su potrebna za korisnikovu interakciju s aplikacijom. Kao dio udaljene poslužiteljske infrastrukture postoje dva poslužiteljska računala: mrežni poslužitelj i poslužitelj baze podataka. Na mrežnom poslužitelju je aktivan proces programa aplikacije koji komunicira s bazom podataka koja je aktivna na vlastitom poslužitelju. Predviđeno je da korisnik koristi mrežni preglednik na vlastitom računalu za komunikaciju s aplikacijom na mrežnom poslužitelju.



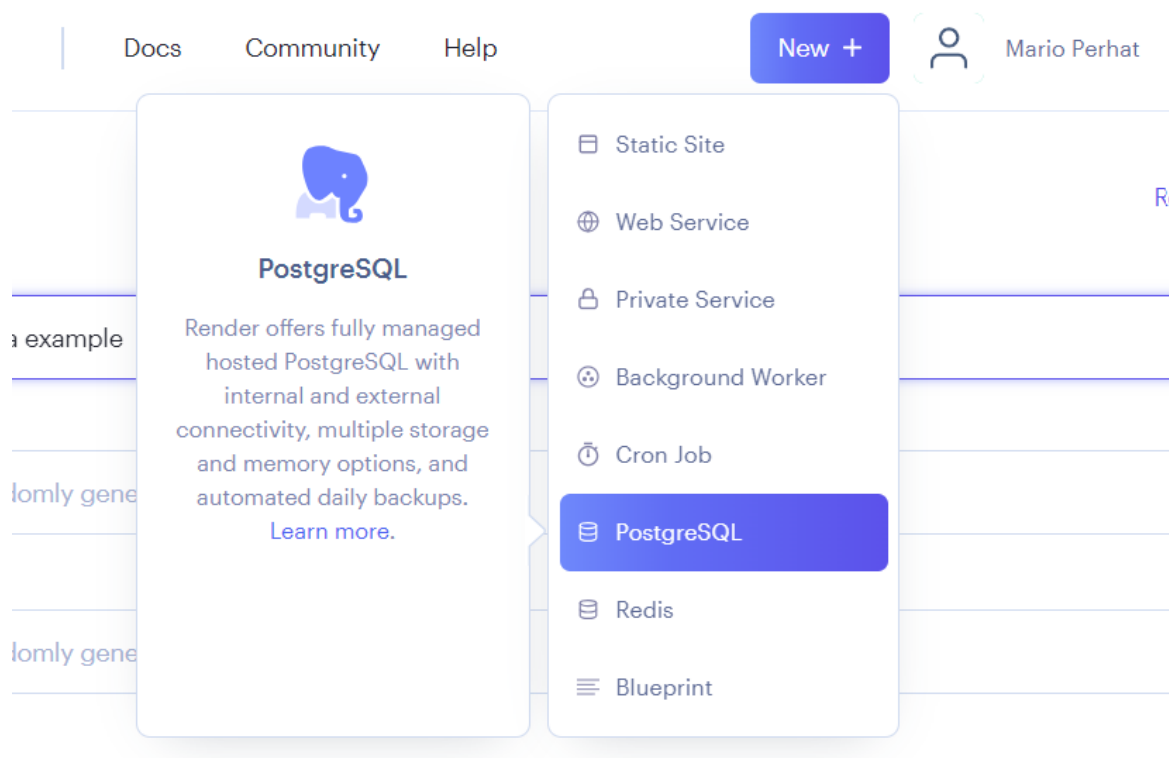
Slika 5.7: Dijagram razmještaja



## 5.4 Upute za puštanje u pogon

### Kreiranje baze podataka na serveru

U ovom projektu je korišten Render oblak. Nakon izrade korisničkog računa možemo pristupiti opciji Dashboard, zatim odabiremo opciju "New +" - PostgreSQL. Unese se ime baze, ostalo se autogenerira, odabire se plan plaćanja (besplatna verzija te različite plaćene) te zatim Create. Time je baza kreirana. Sada ulaskom u dashboard imamo opciju imebaze-db. Klikom na bazu ulazimo u prozor s informacijama o bazi od kojih su nam neke ključne za spajanje baze s backend servisom.



Slika 5.8: Dashboard-New-PostgreSQL

## New PostgreSQL

[Read the docs](#) 

## Name

A unique name for your PostgreSQL instance.

Baza example


Database OptionalThe PostgreSQL `dbname`.

randomly generated unless specified

User Optional

randomly generated unless specified

## Region

The [region](#) where your PostgreSQL instance runs. Services must be in the same region to communicate privately and you currently have services running in **Frankfurt**.Oregon (US West) 

## PostgreSQL Version

15 

Slika 5.9: Izrada Baze Podataka

## Connections

Hostname	ⓘ	dpg-clb1cg616hkc7381inn0-a
Port		5432
Database		dbposterheimer_lpw5
Username		dbposterheimer_lpw5_user
Password	ⓘ	.....
Internal Database URL	ⓘ	.....
External Database URL	ⓘ	.....
PSQL Command	ⓘ	.....

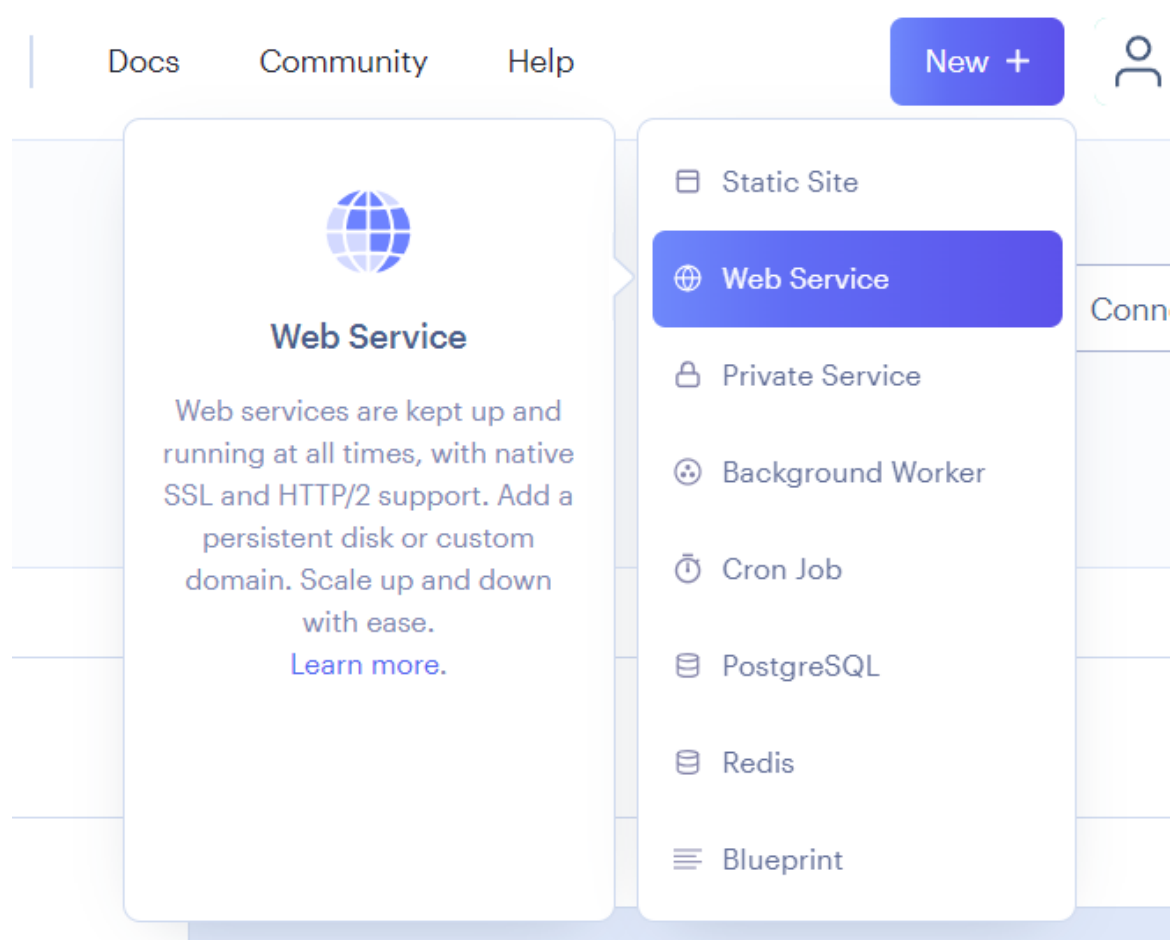
Slika 5.10: Info o Bazi Podataka

## Postavljanje backend servisa na server

Prije svega treba pripremiti backend za puštanje na server. Dodaje se Dockerfile u direktorij docker. Sličnim postupkom kao i za bazu, u Dashboard kliknemo "New +", zatim Web Service, nakon povezivanja git računa, dobije se pristup svim repozitorijima kojima račun ima pravo pristupa. Stisnuti connect kraj odgovarajućeg projekta.

- Postaviti ime servisa
- Root directory postaviti na IzvorniKod/posterheimer-backend
- Environment Docker
- Region postaviti na Frankfurt
- Na dnu proširiti advanced opciju
- Dodati potrebne environment varijable
- Stisnuti Create Web Service

Ako je sve dobro postavljeno, nakon što prođe neko vrijeme dok sve "deploya" na server, servis bi trebao biti "live".



Slika 5.11: Dashboard-New-Web Service

## Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

☒ Build and deploy from a Git repository

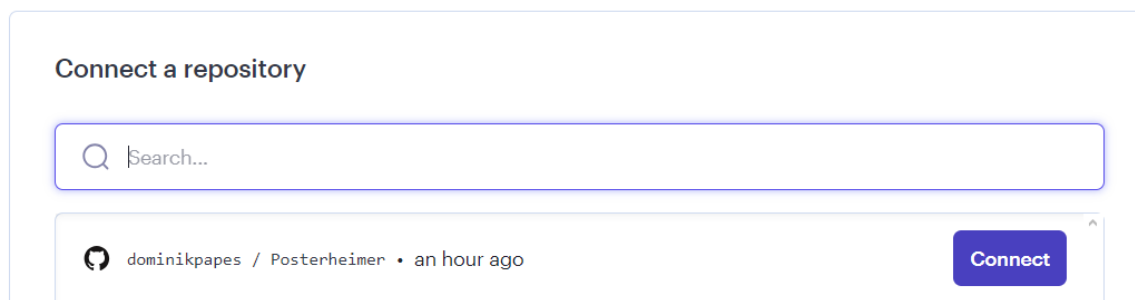
Connect a GitHub or GitLab repository.

☐ Deploy an existing image from a registry **ADVANCED**

Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.

Next

Slika 5.12: Stvaranje iz git repozitorija



Slika 5.13: Spajanje s repozitorijem



You are deploying a web service for [dominikpapes/Posterheimer](#).

<b>Name</b> A unique name for your web service.	<input type="text" value="example-service-name"/>
<b>Region</b> The <a href="#">region</a> where your web service runs. Services must be in the same region to communicate privately and you currently have services running in <b>Frankfurt</b> .	<div>Frankfurt (EU Central) ▾</div>
<b>Branch</b> The repository branch used for your web service.	<div>master ▾</div>
<b>Root Directory</b> <small>Optional</small> Defaults to repository root. When you specify a <a href="#">root directory</a> that is different from your repository root, Render runs all your commands in the <a href="#">specified directory</a> and ignores changes outside the directory.	<div>e.g., src</div>
<b>Runtime</b> The runtime for your web service.	<div>Docker ▾</div>

Slika 5.14: Polja za ispunjavanje

**Environment Variables** Optional  
Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

<input type="text" value="NAME_OF_VARIABLE"/>	<input type="text" value="value"/>	<input type="button" value="Generate"/>	
---	------------------------------------	---	--

▾

Slika 5.15: Dodavanje Environment varijabli

<b>Health Check Path</b> If you're running a server, enter the path where your server will always return a <code>200 OK</code> response. We use it to monitor your app and for <a href="#">zero downtime deploys</a> .	<input type="text" value="/healthz"/>
<b>Registry Credential</b> Use a credential to access private images. Manage credentials in <a href="#">Settings</a>	<input type="text" value="No credential"/>
<b>Docker Build Context Directory</b> <a href="#">Docker build context directory</a> . This is relative to your repository root. Defaults to the root.	<input type="text" value="."/>
<b>Dockerfile Path</b> Path to your Dockerfile relative to the repository root. This is <i>not</i> relative to your Docker build context. For example, <code>./subdir/Dockerfile</code> .	<input type="text" value="Docker/Dockerfile"/>
<b>Docker Command</b> Add an optional command to override the Docker <code>CMD</code> for this service. This will also override the <code>ENTRYPOINT</code> if defined in your Dockerfile. Examples: <code>./start.sh --</code> <code>python /app.py</code>	<input type="text"/>

Slika 5.16: Advance-Docker path

### Environment Variables

Set environment-specific config and secrets (such as API keys), then read those values from your code. [Learn more.](#)

Key	Value	
DB_DRIVER	.....	
DB_PASS	.....	
DB_URL	.....	
DB_USERNAME	.....	
PORT	.....	
SPRING_H2_CONSOLE_ENABLED	.....	
SPRING_PROFILES_ACTIVE	.....	

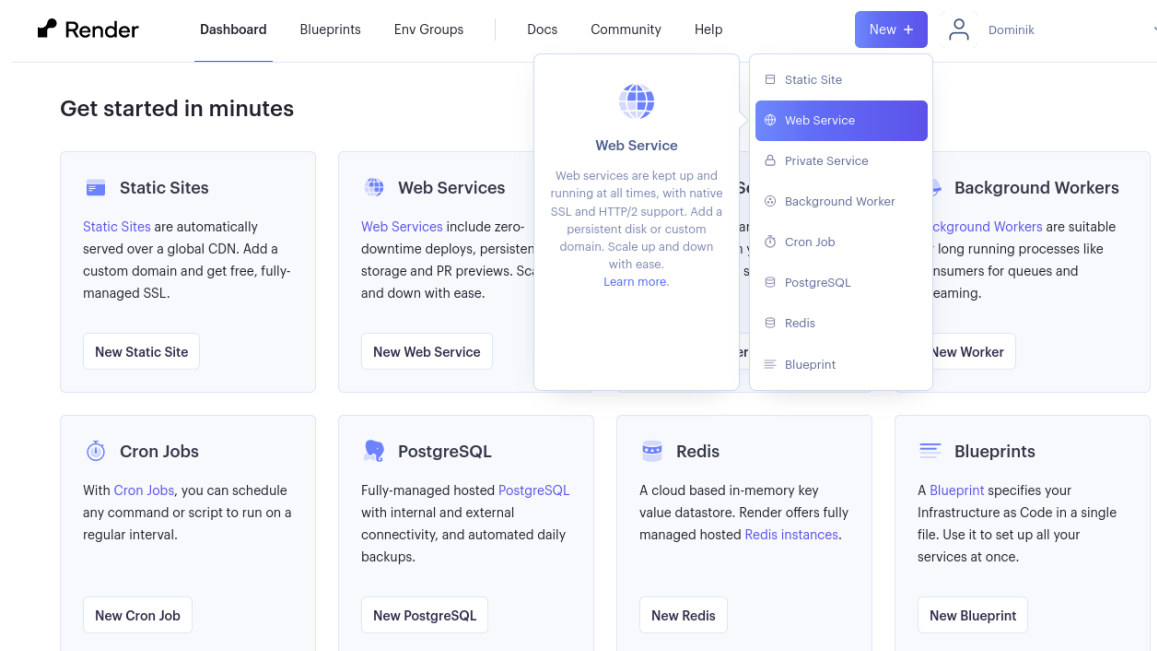
[Create Environment Group](#)

[+ Add Environment Variable](#)[Save Changes](#)

Slika 5.17: Environment varijable

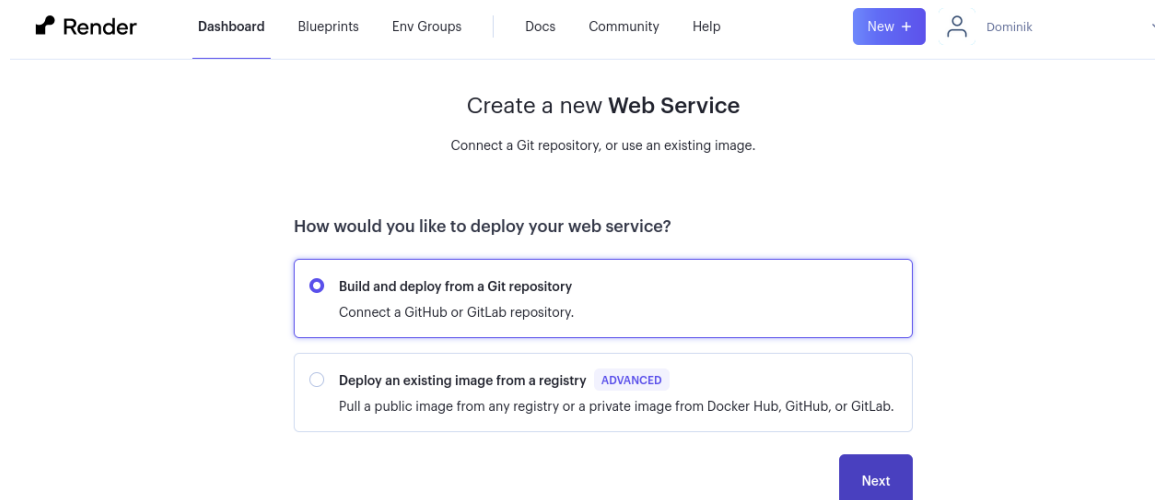
## Puštanje frontenda u pogon

Prije svega trebamo imati Github profil s repozitorijem koji sadrži naš kod. U ovom primjeru radi se o repozitoriju Posterheimer. Potom se trebamo prijaviti na Render s pripadajućim Github profilom. U Render Dashboardu odabiremo opciju "New" - "Web Service".



Slika 5.18: Render Dashboard

U sljedećem izborniku odabiremo opciju "Build and deploy from a Git repository"



Render

Dashboard Blueprints Env Groups Docs Community Help

New + Dominik

### Create a new Web Service

Connect a Git repository, or use an existing image.

How would you like to deploy your web service?

☒ Build and deploy from a Git repository  
Connect a GitHub or GitLab repository.

☐ Deploy an existing image from a registry **ADVANCED**  
Pull a public image from any registry or a private image from Docker Hub, GitHub, or GitLab.

Next

Slika 5.19: Build and deploy

Iz izbornika izabiremo željeni repozitorij i odaberemo opciju "Connect". U sljedećem izborniku odaberemo željeno ime našeg servisa, regiju u kojoj će se servis pokretati, za europsko tržište odaberemo Frankfurt. Odabiremo granu s koje će se servis pokretati, u našem slučaju to je grana master. Specifično za pokretanje frontend koda želimo da se sve komande pokreću unutar direktorija IzvorniKod/posterheimer-frontend, stoga ćemo to upisati u polje Root directory. Runtime za naš servis je Node. Komanda izgradnje mora biti `npm install && npm run build`. Kod samog pokretanja servisa treba se izvršiti naredba `npm run start` koju upisujemo u Start command polje.

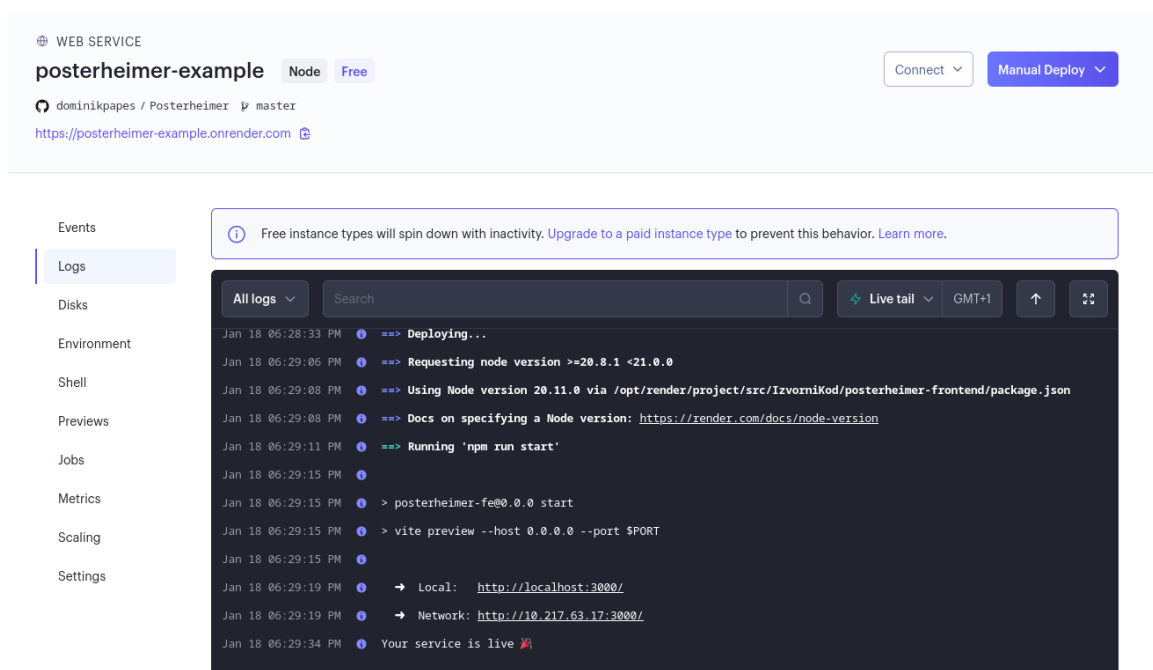
<b>Name</b> A unique name for your web service.	posterheimer-example
<b>Region</b> The region where your web service runs.	Frankfurt (EU Central) ▾
<b>Branch</b> The repository branch used for your web service.	master ▾
<b>Root Directory</b> <small>Optional</small> Defaults to repository root. When you 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/posterheimer-frontend
<b>Runtime</b> The runtime for your web service.	Node ▾
<b>Build Command</b> This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app.	Izvornikod/posterheimer-frontend/ \$ npm install && npm run build
<b>Start Command</b> This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render.	Izvornikod/posterheimer-frontend/ \$ npm run start

Slika 5.20: Komande za pokretanje

Možemo odabrati vrstu instance našeg servisa, postoji besplatna opcija koja nudi najlošije performanse te koju ćemo koristiti za potrebe ovog projekta. Potrebne su nam 3 varijable okoline, Environment variables.

- `API_BASE_URL` = `https://posterheimer-service.onrender.com` - središnja ruta za sve API pozive prema serveru
- `HOST` = `0.0.0.0` - IP računala na kojem se izvodi servis
- `PORT` = `3000` - port na kojem računalo sluša

Nakon što smo unijeli sve vrijednosti možemo kliknuti na "Create Web Service" gumb čime će se započeti puštanje aplikacije u pogon. Nakon kratkog čekanja trebali bismo vidjeti poruku "Your service is live!" nakon čega joj možemo pristupiti putem dodijeljenog linka, u ovom slučaju `https://posterheimer-example.onrender.com/`.



Slika 5.21: Aplikacija uspješno pokrenuta u pogon



## 6. Zaključak i budući rad

### *dio 2. revizije*

*U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.*

*Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.*

Cilj projekta bio je razviti upotrebljivu aplikaciju od koje bi korисти imali organizatori događanja kao što su skupovi ili konvencije natjecateljskog tipa, gdje natjecatelji predstavljaju svoj rad posterom ili prezentacijom, a oni prisutni na događanju svojim glasom doprinose odluci o pobjedniku. Tim koji je radio na aplikaciji sastojao se od šest članova, a rad na projektu trajao je sedamnaest tjedana, podijeljenih na dva dijela. Završetak prvog dijela obilježilo je postavljanje potrebnih poslužitelja, te na njima uspostava početne baze podataka i početne inačice aplikacije. Prije toga se provelo prikupljanje i analiza zahtjeva, osmišljavanje funkcionalnosti aplikacije i dizajn arhitekture sustava, što je uključivalo odabir radnih okruženja i alata. Glavnina vremena ove etape iskoristila se na detaljan ispis obraza uporabe i crtanje dijagrama na temelju kojih bi se kasnije napisao kod. S početkom drugog dijela razvoja, tim se podijelio u tri para, pri čemu su dva člana bila zadužena za prednji (*engl. frontend*), a dva za pozadinski (*engl. backend*) dio koda, te dva člana za dovršavanje dokumentacije. U ovoj fazi se uglavnom stjecalo potrebno znanje i usavršavale se vještine za rad s odabranim radnim okvirima i knjižnicama, koje se tada koristilo za oblikovanje koda prisutnog u konačnom rješenju. Po uzoru na to rješenje pisao se nastavak dokumentacija, odnosno njezini dijelovi koji ovaj put služe kako bi opisali funkcionalnost programirane aplikacije.

# Popis literature

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	UC dijagram za opis korisničkih računa . . . . .	24
3.2	UC dijagram za opis administrativnih mogućnosti . . . . .	25
3.3	UC dijagram za opis funkcionalnosti aplikacije . . . . .	25
3.4	Sekvencijski dijagram za UC15 - Glasovanje . . . . .	27
3.5	Sekvencijski dijagram za UC18 - Upravljanje posterima . . . . .	29
3.6	Sekvencijski dijagram za UC31 - Slanje e-pošte . . . . .	30
4.1	Reprezentacija arhitekture sustava . . . . .	33
4.2	E-R dijagram baze podataka . . . . .	41
4.3	Dijagram razreda - Upravljači . . . . .	42
4.4	Dijagram razreda - DAO . . . . .	43
4.5	Dijagram razreda - Modeli . . . . .	44
4.6	Dijagram stanja . . . . .	47
4.7	Dijagram aktivnosti - Glasovanje . . . . .	48
4.8	Dijagram komponenti . . . . .	50
5.1	Prikaz uspješnosti ispitnih slučajeva . . . . .	54
5.2	Prikaz pada ispitnog slučaja . . . . .	57
5.3	Prikaz uspješnosti ispitnog slučaja . . . . .	64
5.4	Prikaz uspješnosti ispitnog slučaja . . . . .	66
5.5	Prikaz uspješnosti ispitnog slučaja . . . . .	68
5.6	Prikaz uspješnosti ispitnog slučaja . . . . .	69
5.7	Dijagram razmještaja . . . . .	71
5.8	Dashboard-New-PostgreSQL . . . . .	73
5.9	Izrada Baze Podataka . . . . .	74
5.10	Info o Bazi Podataka . . . . .	75
5.11	Dashboard-New-Web Service . . . . .	77
5.12	Stvaranje iz git repozitorija . . . . .	78
5.13	Spajanje s repozitorijem . . . . .	79
5.14	Polja za ispunjavanje . . . . .	80
5.15	Dodavanje Environment varijabli . . . . .	81

5.16 Advance-Docker path . . . . .	82
5.17 Environment varijable . . . . .	83
5.18 Render Dashboard . . . . .	84
5.19 Build and deploy . . . . .	85
5.20 Komande za pokretanje . . . . .	86
5.21 Aplikacija uspješno pokrenuta u pogon . . . . .	87

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 18. listopada 2023.
- Prisustvovali: D.Papeš, F.Androić, A.Batić, V.Javornik, M.Perhat, D.Tomšić
- Teme sastanka:
  - upoznavanje tima
  - dogovor načina komunikacije

### 2. sastanak

- Datum: 23. listopada 2023.
- Prisustvovali: D.Papeš, F.Androić, A.Batić, V.Javornik, M.Perhat, D.Tomšić
- Teme sastanka:
  - podjela članova u dvije podgrupe, prva će napraviti funkcionalne zahtjeve i obrasce uporabe, druga sekvencijske dijagrame i ostale zahtjeve
  - razmjena pitanja koje će grupa postaviti na laboratorijskim vježbama

### 3. sastanak

- Datum: 1. studenoga 2023.
- Prisustvovali: D.Papeš, F.Androić, A.Batić, V.Javornik, M.Perhat, D.Tomšić
- Teme sastanka:
  - dovršavanje obrazaca uporabe
  - dogovaranje oko korisničkog sučelja aplikacije

### 4. sastanak

- Datum: 9. studenoga 2023.
- Prisustvovali: D.Papeš, A.Batić, V.Javornik, M.Perhat, D.Tomšić
- Teme sastanka:
  - dogovaranje oko podjele rada

### 5. sastanak

- Datum: 14. studenoga 2023.

- Prisustvovali: D.Papeš,F.Androić,V.Javornik,M.Perhat
- Teme sastanka:
  - dogovor oko dovršavanja dokumentacije
  - pregled napisanog koda i dogovor o daljnjoj implementaciji

#### 6. sastanak

- Datum: 3. siječnja 2024.
- Prisustvovali: D.Papeš,F.Androić,A.Batić,V.Javornik,M.Perhat,D.Tomšić
- Teme sastanka:
  - pregled napravljenog
  - konačni dogovori o daljnjoj implementaciji

## Tablica aktivnosti

	Dominik Papeš	Fran Androić	Ante Batić	Valeria Javornik	Mario Perhat	Dario Tomšić
Upravljanje projektom	3	2			3	
Opis projektnog zadatka	1	2		2	1	
Funkcionalni zahtjevi	2				2	
Opis pojedinih obrazaca	3	3	3	3	3	3
Dijagram obrazaca		2.5				1
Sekvencijski dijagrami			1	1	1	
Opis ostalih zahtjeva	0.5				1	
Arhitektura i dizajn sustava		0.5	1			
Baza podataka				3.5		
Dijagram razreda		2		0.5		2.5
Dijagram stanja						
Dijagram aktivnosti						
Dijagram komponenti						
Korištene tehnologije i alati						
Ispitivanje programskog rješenja						
Dijagram razmještaja						
Upute za puštanje u pogon					2	
Dnevnik sastajanja						
Zaključak i budući rad						
Popis literature						
Učenje novih tehnologija	8				16	

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	<b>Dominik Papeš</b>	<b>Fran Androić</b>	<b>Ante Batić</b>	<b>Valeria Javornik</b>	<b>Mario Perhat</b>	<b>Dario Tomšić</b>
Izrada korisničkog sučelja	8					
Povezivanje s bazom	8				10	
Izrada backend					35	
Izrada frontend						
Postavljanje na server	8	2			10	



## Dijagrami pregleda promjena

### *dio 2. revizije*

*Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s [gitlab.com](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*