

Programsko inženjerstvo

Ak. god. 2022./2023.

Kokeferencije

Dokumentacija, Rev. 2.

Grupa: *Koke*

Voditeljica: *Nikoleta Benić*

Datum predaje: *13. siječnja 2023.*

Nastavnik: *Miljenko Krhen*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	7
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	20
3.2 Nefunkcionalni zahtjevi	26
4 Arhitektura i dizajn sustava	27
4.1 Baza podataka	28
4.1.1 Opis tablica	29
4.1.2 Dijagram baze podataka	35
4.2 Dijagram razreda	36
4.3 Dijagram stanja	41
4.4 Dijagram aktivnosti	43
4.5 Dijagram komponenti	47
5 Implementacija i korisničko sučelje	49
5.1 Korištene tehnologije i alati	49
5.2 Ispitivanje programskog rješenja	51
5.2.1 Ispitivanje komponenti	51
5.2.2 Ispitivanje sustava	57
5.3 Dijagram razmještaja	61
5.4 Upute za puštanje u pogon	62
6 Zaključak i budući rad	66
Popis literature	67
Indeks slika i dijagrama	69

Dodatak: Prikaz aktivnosti grupe

70

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Andrea Kaselj	31.10.2022.
0.2	Zapisani dotadašnji sastanci u dodatku.	Andrea Kaselj	31.10.2022.
0.3	Dodana 4 Use Case i 2 sekvencijska dijagrama	Ana Čepić	12.11.2022.
0.4	Dodana 4 Use Case dijagrama	Iva Ursić	14.11.2022.
0.5.	Dodani opisi tablica baze podataka	Josipa Markić	14.11.2022.
0.5.1	Dijagram baze podataka	Dorotea Dragojević	15.11.2022.
0.5.2	Popravljeni opisi tablica baze podataka	Josipa Markić	16.11.2022.
0.6	Opis projektnog zadatka	Valentina Valić	16.11.2022.
0.7	Dodana 4 Use case dijagrama	Nikoleta Benić	16.11.2022.
0.8	Dodana 4 Use case dijagrama	Andrea Kaselj	16.11.2022.
0.9	Opisani nefunkcionalni zahtjevi	Andrea Kaselj	16.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.9.1	Ispravljeni <i>Use Case</i>	Nikoleta Benić	17.11.2022.
0.9.2	Ispravljeni Use Case dijagrami	Ana Čepić	17.11.2022.
0.9.3	Ispravljena naslovna strana	Josipa Markić	17.11.2022.
0.9.4	Popunjena tablica aktivnosti	Dorotea Dragojević	17.11.2022.
0.9.5	Opisana arhitektura sustava	Andrea Kaselj	18.11.2022.
0.9.6	Dodan dijagram razreda	Nikoleta Benić	18.11.2022.
1.0	Popravak i provjera dokumentacije	Nikoleta Benić	18.11.2022.
1.1	Dodan dijagram aktivnosti za system ownera	Dorotea Dragojević	3.1.2023.
1.1.1	Dodan dijagram aktivnosti za glavnog admina, operativnog admina i korisnika	Dorotea Dragojević	7.1.2023.
1.1.2	Dodano ispitivanje komponenti	Dorotea Dragojević	7.1.2023.
1.1.3	Aktivnosti grupe	Dorotea Dragojević	7.1.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.1.4	Grafovi promjena datoteka	Dorotea Dragojević	10.1.2023.
1.2	Ispravak tablica baze podataka	Josipa Markić	10.1.2023.
1.2.1	Ispravljena naslovna strana	Josipa Markić	10.1.2023.
1.2.2	Relacijski model baze	Josipa Markić	10.1.2023.
1.3	Dodano ispitivanje komponenti	Iva Ursić	10.1.2023.
1.3.1	Dodan zaključak	Iva Ursić	10.1.2023.
1.3.2	Nadopunjeni nefunkcionalni zahtjevi	Iva Ursić	10.1.2023.
1.3.3	Dodani novi sekvencijski dijagrami	Josipa Markić	10.1.2023.
1.3.4	Dodane korištene tehnologije i alati	Valentina Valić	10.1.2023.
1.3.5	Dodana literatura	Valentina Valić	10.1.2023.
1.4	Ažurirani dijagrami razreda	Nikoleta Benić	10.1.2023.
1.4.1	Dodane upute za puštanje u pogon	Nikoleta Benić	11.1.2023.
1.5	Dodani novi sekvencijski dijagrami	Ana Čepić	11.1.2023.
1.5.1	Uređeni obrasci uporabe	Ana Čepić	11.1.2023.
1.6	Dodani dijagram komponenti	Ana Čepić	11.1.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.7	Dodan dijagram stanja	Nikoleta Benić	11.1.2023.
1.8	Dodan dijagram razmještaja	Josipa Markić	11.1.2023.
1.8.1	Dodani use case dijagrami	Iva Ursić	11.1.2023.
1.9	Ispravci i provjera	Josipa Markić	11.1.2023.
1.9.1	Ispravak use case dijagrama	Iva Ursić	11.1.2023.
1.9.2	Ispitivanje sustava	Nikoleta Benić	12.1.2023.
1.9.3	Ispravak opisa projektnog zadatka	Valentina Valić	12.1.2023.
2.0	Pregled dokumentacije	Iva Ursić	13.1.2023.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije “Program Konferencije” koja pruža učinkovit informacijski sustav koji omogućuje praćenje dolazaka sudionika na konferenciju, sudjelovanje sudionika u glavnim i popratnim događanjima na konferenciji i davanje svih potrebnih informacija sudionicima. Registriranim korisnicima konferencije informacijski sustav služi kao temeljno mjesto za dobivanje informacije, dok organizator pomoću njega ima mogućnost distribucije raznih materijala sudionicima, davanja svih potrebnih informacija te praćenja aktivnosti sudionika tijekom konferencije. Ako korisnik nije registriran, na web stranici može vidjeti samo temeljni opis događaja i tema o kojima se raspravlja na konferenciji.

Četiri su vrste korisnika u informacijskom sustavu:

Vlasnik sustava određuje administratora za svaku pojedinu konferenciju te kreira generalne podatke o konferencijama čiji se rad prati.

Glavni administrator konferencije imenuje operativne administratore, te upisuje konkretne i detaljne podatke o konferenciji. Sustav mu mora omogućiti pregled prijavljenih sudionika, pregled po državama od kuda sudionici dolaze, pregled po njihovim statusima te broj prijavljenih sudionika na pojedinim posebnim događanjima. Glavni administrator ima mogućnost upisa do najviše 15 grupa podataka o nekoj konferenciji, a pritom su obavezni dijelovi za svaku konferenciju:

- Raspored predavanja
- Prezentacije predavanja u pdf formatu dostupne svim sudionicima za preuzimanje
- Zbornik radova u pdf formatu dostupan svim sudionicima za preuzimanje
- Mjesto događanja

Također ima mogućnost postaviti sve multimedijske materijale na poslužitelj kako bi registrirani korisnici imali pristup tim materijalima u obliku pregleda te može preuzeti pdf s podacima o konferenciji.

Operativni administrator koji prilikom dolaska na konferenciju svakom sudioniku dodjeljuje, tj. aktivira korisnički račun koji sudionik potom mora potvrditi putem elektroničke pošte. Za svakog korisnika treba upisati:

- Ime i prezime
- Broj telefona
- Adresu elektroničke pošte pod pretpostavkom da je svi sudionici koriste
- Adresu
- Državu
- Naziv institucije ili poduzeća iz kojeg dolazi
- Svojstvo sudjelovanja na konferenciji (gost, predavač, pozvani predavač, sudionik s objavljenim radom u zborniku, sudionik s plaćenom kotizacijom bez predavanja, pratnja sudionika)

Sudionik konferencije čija je funkcija da pregledava podatke namijenjene sudionicima. Može se registrirati tek nakon početka i dolaska na konferenciju. Kako bi registracija bila uspješna potrebno je provjeriti zadovoljava li sudionik sve zahtjeve za registracijom, a ti su podaci dostupni u drugom, tj. nezavisnom informacijskom sustavu. Registrirani sudionik može izdati zahtjev o potvrdi za sudjelovanjem u obliku pdf datoteke koja će biti izrađena na službenom dokumentu, tj. memorandumu konferencije. Na njoj će biti navedeno da je sudionik (imenom i prezime-nom) sudjelovao na određenoj konferenciji u navedenom vremenskom terminu.

Nakon kreiranja konferencije u sustavu upisuju se njeni podatci, a dostupni su do 30 dana nakon njenog završetka nakon čega se korisnicima zabranjuje pristup, osim glavnom administratoru koji ima period od 40 dana nakon završetka konferencije za spremanje svih podataka i zaključivanje rada s aplikacijom. Također se treba omogućiti prikaz podataka o trenutnim vremenskim uvjetima i prognozi vremena za lokaciju održavanja konferencije sudionicima što se preuzima od nekog javno dostupnog servisa. Obzirom da se sva događanja tijekom rada konferencije snimaju i slikaju od strane ovlaštenog fotografa na stranici postoji obavijest da se događaji snimaju. Potrebno je omogućiti pregled konferencija putem javno nedostupnog Youtube kanala kroz aplikaciju. Kanalu je moguće pristupiti preko poveznice koja se nalazi u aplikaciji. Pregled multimedijских materijala napravljen

je po danima, a pristupa im se zasebno za svaki pojedini dan. Po želji se pojedini materijali mogu skinuti i na lokalno računalo.

Tijekom konferencije su predviđena i posebna događaja te je broj mjesta na njima ograničen. Zbog toga se svaki sudionik treba prijaviti ukoliko želi na njima sudjelovati, a dozvoljeni broj sudionika je definiran od strane glavnog organizatora konferencije. Ukoliko nakon prijave sudionika sustav ustanovi da nema više slobodnih mjesta, glavnog administratora se obavještava putem elektroničke pošte o dostignutom najvećem broju sudionika na nekom događaju te ga se upućuje na potrebu rješavanja prekomjernih zahtjeva. Glavni administrator naknadno može povećati broj mogućih sudionika na nekom posebnom događaju, u tome slučaju se sudionici koji su na listi čekanja automatski prijavljuju na posebni događaj te dobivaju obavijest na mail da mogu pohađati taj posebni događaj.

Sustav mora omogućiti istovremeni rad svih korisnika sustava i unos hrvatskih diakritičkih znakova.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik sustava
2. Glavni administrator konferencije
3. Operativni administrator
4. Sudionik konferencije
5. Razvojni tim
6. Baza podataka
7. Cloudinary
8. Poslužitelj za vremensku prognozu

Aktori i njihovi funkcionalni zahtjevi:

1. Neprijavljeni korisnik može:
 - (a) pregledati aktivne konferencije i temeljne podatka o svakoj od tih konferencija
2. Vlasnik sustava može:
 - (a) Prijaviti se u sustav
 - (b) Dodati konferenciju
 - (c) Kreirati korisnički račun glavnog admina svake konferencije
3. Glavni administrator može:
 - (a) Prijaviti se u sustav
 - (b) Unijeti grupe podataka za konferenciju
 - (c) Kreirati korisničke račune za operativne admina konferencije
 - (d) Spremiti podatke o konferenciji nakon njenog završetka u obliku PDFa
 - (e) Postaviti multimedijske materijale sa konferencije i skinuti ih
 - (f) Pregledati sudionike konferencije po ulozi i državi iz koje dolaze

- (g) Kreirati poseban događaj i po potrebi povećati kapacitet
- (h) Pregledati detaljne podatke o svojoj konferenciji

4. Operativni administrator može:

- (a) Prijaviti se u sustav
- (b) Kreirati korisničke račune sudionika konferencije
- (c) Pregledati detaljne podatke o svojoj konferenciji

5. Sudionik konferencije može:

- (a) Prijaviti se u sustav
- (b) Pregledati multimedijske materijale i skinuti ih
- (c) Prijaviti se na posebna događanja
- (d) Skinuti PDF potvrdu o sudjelovanju
- (e) Pregledati detaljne podatke o svojoj konferenciji

6. Baza podataka može:

- (a) Pohranjivati podatke o konferencijama
- (b) Pohranjivati podatke o registriranim korisnicima
- (c) Pohranjivati podatke o posebnim događajima
- (d) Pohranjivati podatke o multimediji
- (e) Pohranjivati podatke o grupama podataka

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 -Pregled aktivnih konferencija

- **Glavni sudionik:** neprijavljeni korisnik, prijavljeni korisnik(sudionik, glavni admin, vlasnik sustava, operativni admin)
- **Cilj:** Pregledati aktivne konferencije
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Popis konferencija vidljiv je odmah nakon pokretanja aplikacije
 2. Prikazuju se osnovne informacije o konferenciji: grad održavanja, opis konferencije i teme

UC2 - Registracija glavnog admina

- **Glavni sudionik:** vlasnik sustava
- **Cilj:** Stvoriti korisnički račun za glavnog admina konferencije
- **Sudionici:** Baza podatka, glavni admin
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Vlasnik sustava odabere opciju *Registriraj*
 2. Vlasnik sustava unosi potrebne podatke: ime, prezime, broj telefona, e-mail, adresu, državu, naziv poduzeća iz koje dolazi, svojstvo sudjelovanja na konferenciji
 3. Povratak na meni
- **Opis mogućih odstupanja:**
 1. Unos već zauzetog korisničkog imena, unos podataka u nedozvoljenom formatu ili pružanje neispravnog e-maila:
 1. Sustav obavještava glavnog sudionika o neuspjelom upisu
 2. Glavni sudionik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC3 - Registracija operativnog admina

- **Glavni sudionik:** glavni admin
- **Cilj:** Stvoriti korisnički račun za operativnog admina konferencije

- **Sudionici:** Baza podataka, operativni admin
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Glavni admin odabere opciju *Registriraj*
 2. Glavni admin unosi potrebne podatke: ime, prezime, broj telefona, e-mail, adresu, državu, naziv poduzeća iz koje dolazi, svojstvo sudjelovanja na konferenciji
 3. Povratak na meni
- **Opis mogućih odstupanja:**
 1. Unos već zauzetog korisničkog imena, unos podataka u nedozvoljenom formatu ili pružanje neispravnog e-maila:
 1. Sustav obavještava glavnog sudionika o neuspjelom upisu
 2. Glavni sudionik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC4 - Registracija sudionika

- **Glavni sudionik:** operativni admin
- **Cilj:** Stvoriti korisnički račun za sudionika konferencije
- **Sudionici:** Baza podataka, operativni admin
- **Preduvjet:** Plaćena kotizacija
- **Opis osnovnog tijeka:**
 1. Operativni admin odabere opciju *Registriraj*
 2. Operativni admin unosi potrebne podatke: ime, prezime, broj telefona, e-mail, adresu, državu, naziv poduzeća iz koje dolazi, svojstvo sudjelovanja na konferenciji
 3. Sustav šalje mail s linkom za potvrdu registracije sudioniku
 4. Povratak na meni
- **Opis mogućih odstupanja:**
 1. Sudionik nije platio kotizaciju
 1. Odustaje se od registracije
 2. Unos već zauzetog korisničkog imena, unos podataka u nedozvoljenom formatu ili pružanje neispravnog e-maila:
 - (a) Sustav obavještava glavnog sudionika o neuspjelom upisu
 - (b) Glavni sudionik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC5 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija uspješna
- **Opis osnovnog tijeka:**
 1. Korisnik odabere opciju *Prijavi se*
 2. Korisnik unosi korisničko ime i lozinku
 3. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
 1. Neispravno korisničko ime i lozinka
 1. Sustav obavještava korisnika o neuspjeloj prijavi porukom "Login failed"

UC6 - Stvaranje konferencije

- **Glavni sudionik:** vlasnik sustava
- **Cilj:** Unijeti temeljne podatke o konferenciji u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Vlasnik sustava odabere opciju *Dodaj konferenciju*
 2. Vlasnik unosi temeljne podatke: naziv, mjesto i vrijeme, teme, događaji, glavnog administratora konferencije
 3. Konferencija se prikazuje u sustavu
- **Opis mogućih odstupanja:**
 1. Nisu uneseni svi podatci
 1. Sustav obavještava korisnika da mora unijeti sve podatke za konferenciju

UC7 - Unos grupa podataka za konferenciju

- **Glavni sudionik:** glavni administrator
- **Cilj:** unijeti grupe podataka za konferenciju
- **Sudionici:** baza podataka
- **Preduvjet:** konferencija stvorena
- **Opis osnovnog tijeka:**
 1. Glavni admin najprije unosi glavne grupe podataka
 2. Glavni admin odabere opciju *Unos podataka*

3. Glavni admin unosi do najviše 15 grupa podataka među kojima su obavezni: raspored predavanja, zbornik radova, prezentacije predavanja i mjesto događaja
4. Detaljni podaci dostupni registriranim korisnicima na pregled
- **Opis mogućih odstupanja:**
 1. Nepotpuno ispunjavanje obveznih grupa podataka
 1. Admin ne može unijeti ostale grupe podataka
 2. Nepotpuno ispunjavanje forme za kreiranje neobavezne grupe podataka
 1. Sustav obavještava admina da mora unijeti sve podatke
 3. Admin pokušava unijeti više od 15 grupa podataka
 1. Sustav obavještava admina da ne može unijeti sve podatke

UC8 - Spremanje podataka o konferenciji u obliku PDF-a

- **Glavni sudionik:** Glavni administrator
- **Cilj:** spremi sve podatke prije zaključenja rada s aplikacijom
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Glavni admin odabire opciju *Spremi konferenciju*
 2. Podaci se spremaju na lokalno računalo u obliku PDF-a

UC9 - Postavljanje multimedijских sadržaja

- **Glavni sudionik:** Glavni administrator
- **Cilj:** ponuditi pregled i skidanje multimedijskog sadržaja registriranim korisnicima
- **Sudionici:** poslužitelj Cloudinary
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Glavni admin odabire opciju *Dodaj multimedijски sadržaj*
 2. Glavni admin dodaje direktorij za pojedini dan ili odabire postojeći direktorij
 3. Glavni admin postavlja multimedijски sadržaj za pojedini dan
- **Opis mogućih odstupanja:**
 1. Pokušava se kreirati direktorij s datumom koji nije unutar datuma održavanja konferencije
 1. Sustav obavještava korisnika da uneseni datum nije dobar

UC10 - Pregled sudionika konferencije

- **Glavni sudionik:** Glavni administrator
- **Cilj:** Pregled prijavljenih sudionika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Glavni admin odabere opciju *Pregled prijavljenih sudionika*
 2. Prikazuju se podaci o prijavljenim sudionicima (status i država iz koje dolaze)

UC11 - Pregled multimedijских sadržaja

- **Glavni sudionik:** Korisnik(sudionik, glavni admin ili operativni admin)
- **Cilj:** Pregled multimedijских materijala po danima i mogućnost preuzimanja na lokalno računalo
- **Sudionici:** Poslužitelj Cloudinary
- **Preduvjet:** Glavni administrator konferencije je postavio multimedijске materijale na poslužitelj
- **Opis osnovnog tijeka:**
 1. Korisnik odabere opciju *Pregled multimedijского sadržaja*
 2. Korisniku se prikazuju multimedijски materijali po danima
 3. Korisnik ih može preuzeti klikom na gumb "Preuzmi"

UC12 - Prijava na poseban događaj

- **Glavni sudionik:** Sudionik konferencije
- **Cilj:** Prijava na posebni događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabere opciju *Prijavi se na posebni događaj*
 2. Sustav ispisuje potvrdu o uspješnoj prijavi
- **Opis mogućih odstupanja:**
 1. Sva mjesta za željeni događaj su zauzeta
 1. Sudionik se stavlja u listu čekanja za događaj
 2. Sustav obavještava glavnog administratora o prekobrojnoj potražnji za već popunjeni događaj i mogućnosti povećanja broja sudionika

UC13 - Generiranje PDF potvrde o sudjelovanju

- **Glavni sudionik:** Korisnik
- **Cilj:** Generiranje PDF potvrde
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju *Potvrda o sudjelovanju*
 2. Sustav generira PDF dokument o sudjelovanju
 3. Korisnik može preuzeti pdf potvrdu

UC14 - Povećanje kapaciteta posebnog događaja

- **Glavni sudionik:** Glavni admin
- **Cilj:** Povećanje kapaciteta posebnog događaja
- **Sudionici:** baza podataka
- **Preduvjet:** Popunjenost kapaciteta posebnog događaja, postoji poseban događaj
- **Opis osnovnog tijeka:**
 1. Glavni admin bira opciju *Povećaj kapacitet*
 2. Sustav ažurira broj mjesta na posebnom događaju
 3. Sudionik koji je bio u listi čekanja dobiva mail obavijest da se mjesto otvorilo

UC15 - Stvaranje posebnog događaja

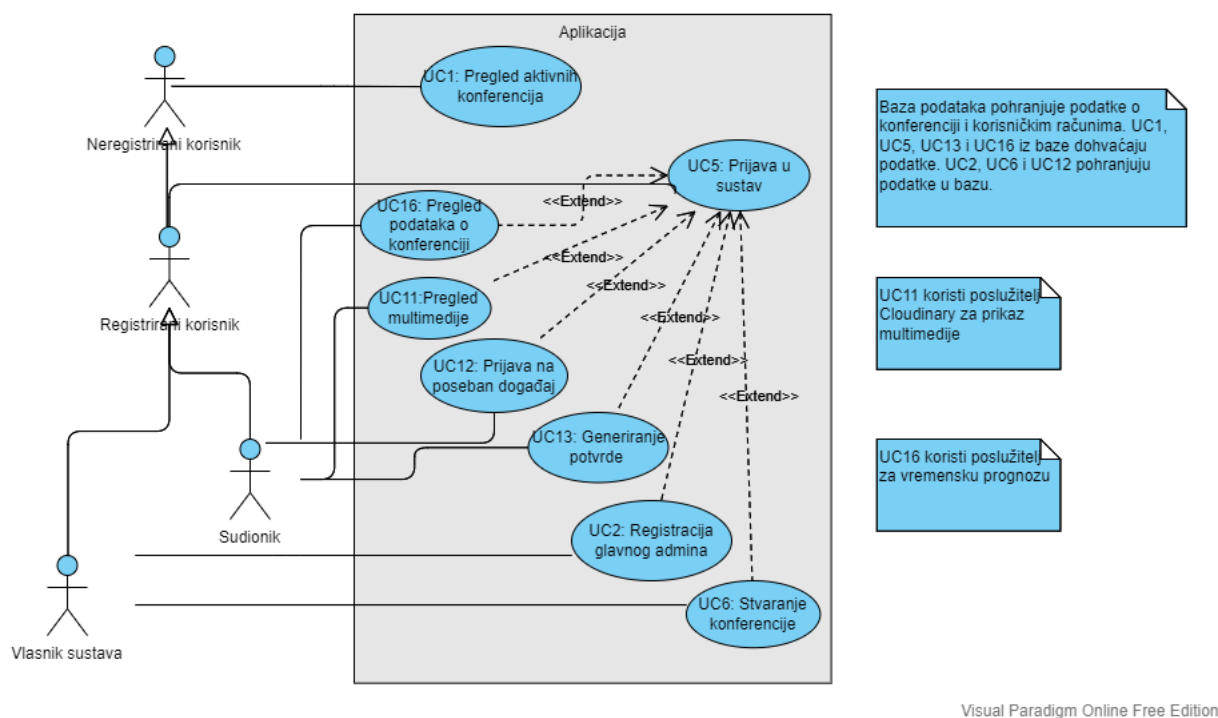
- **Glavni sudionik:** Glavni admin
- **Cilj:** Stvoriti poseban događaj na konferenciji
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Glavni admin bira opciju *Kreiraj posebni događaj*
 2. Glavni admin unosi kapacitet, tip posebnog događaja i poruku sudionicima
 3. Poseban događaj se dodaje kao grupa podataka konferencije
- **Opis mogućih odstupanja:**
 1. Nisu uneseni svi podatci
 1. Sustav obavještava korisnika da mora unijeti sve podatke

UC16 - Pregled podataka o konferenciji

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledati detaljne podatke o konferenciji
- **Sudionici:** baza podataka, poslužitelj za vremensku prognozu
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju *Moja konferencija*
 2. Prikazuje se stranica sa svim podacima o konferenciji

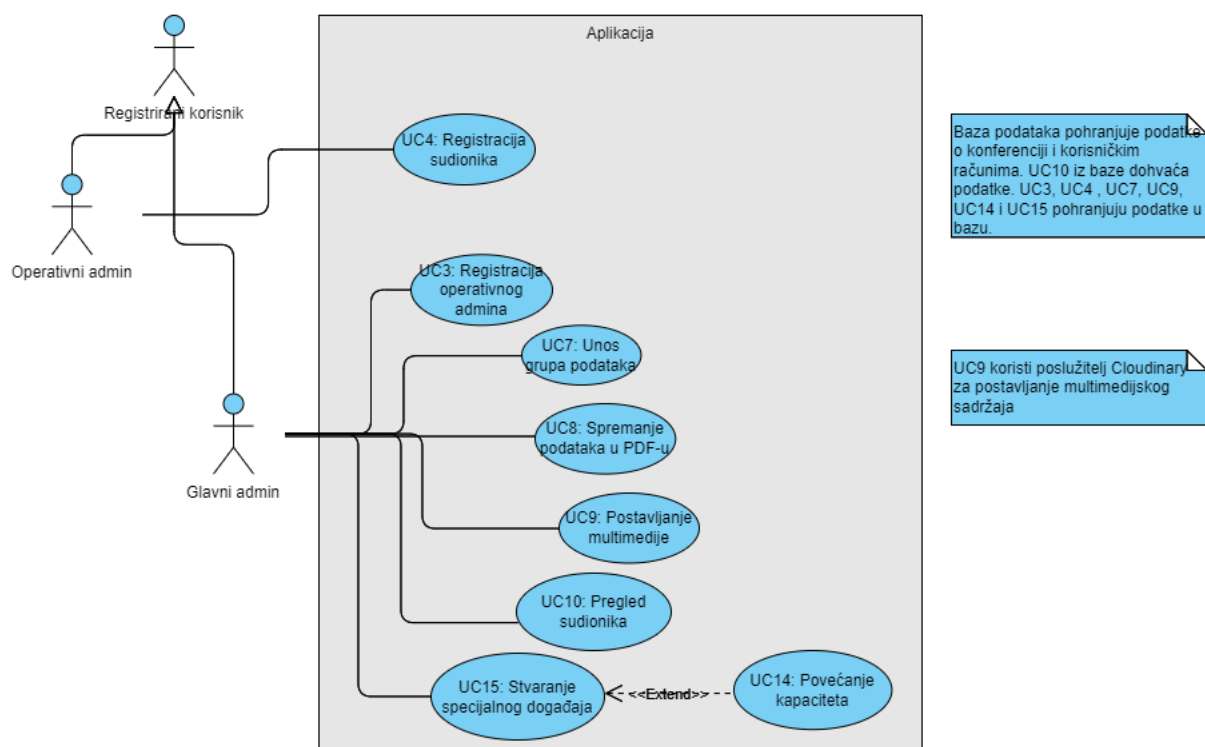
Dijagrami obrazaca uporabe

Visual Paradigm Online Free Edition



Slika 3.1: Funkcionalnosti za sudionika i vlasnika sustava

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

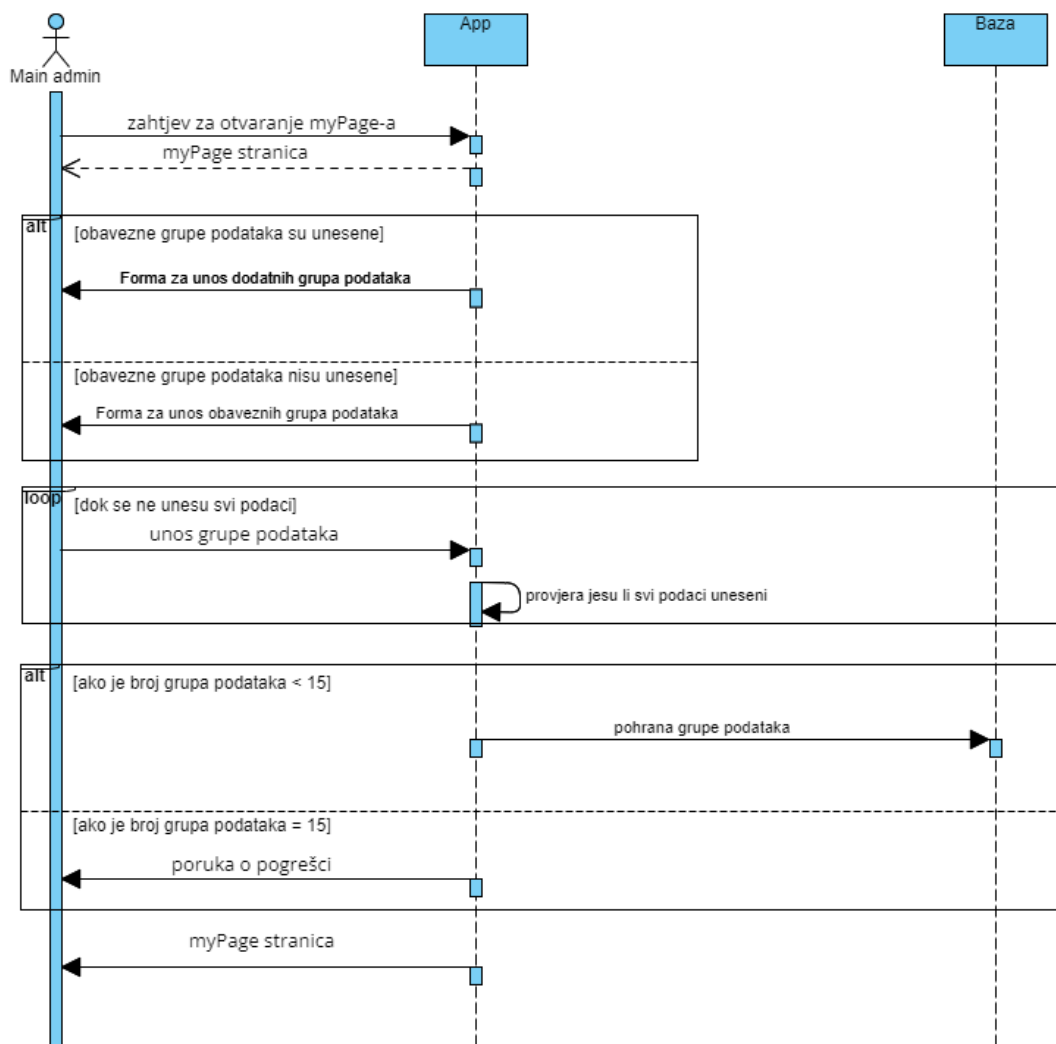
Slika 3.2: Funkcionalnosti za glavnog admina i operativnog admina

3.1.2 Sekvencijski dijagrami

UC7: Unos grupa podataka za konferenciju

Glavni admin konferencije unosi grupe podataka za svoju konferenciju. Postoje 4 obavezne grupe podataka: raspored predavanja, zbornik radova, prezentacije predavanja i lokacija događanja te jedna grupa podataka koja je obavezna samo ako postoji posebni događaj - obavijest o posebnom događaju. Glavni admin mora prvo unijeti obavezne grupe podataka, a onda tek smije unijeti dodatne podatke. Moguće je unijeti maksimalno 15 grupa podataka za neku konferenciju. Ukoliko je već uneseno 15 grupa podataka aplikacija obaviještava admina porukom.

Visual Paradigm Online Free Edition



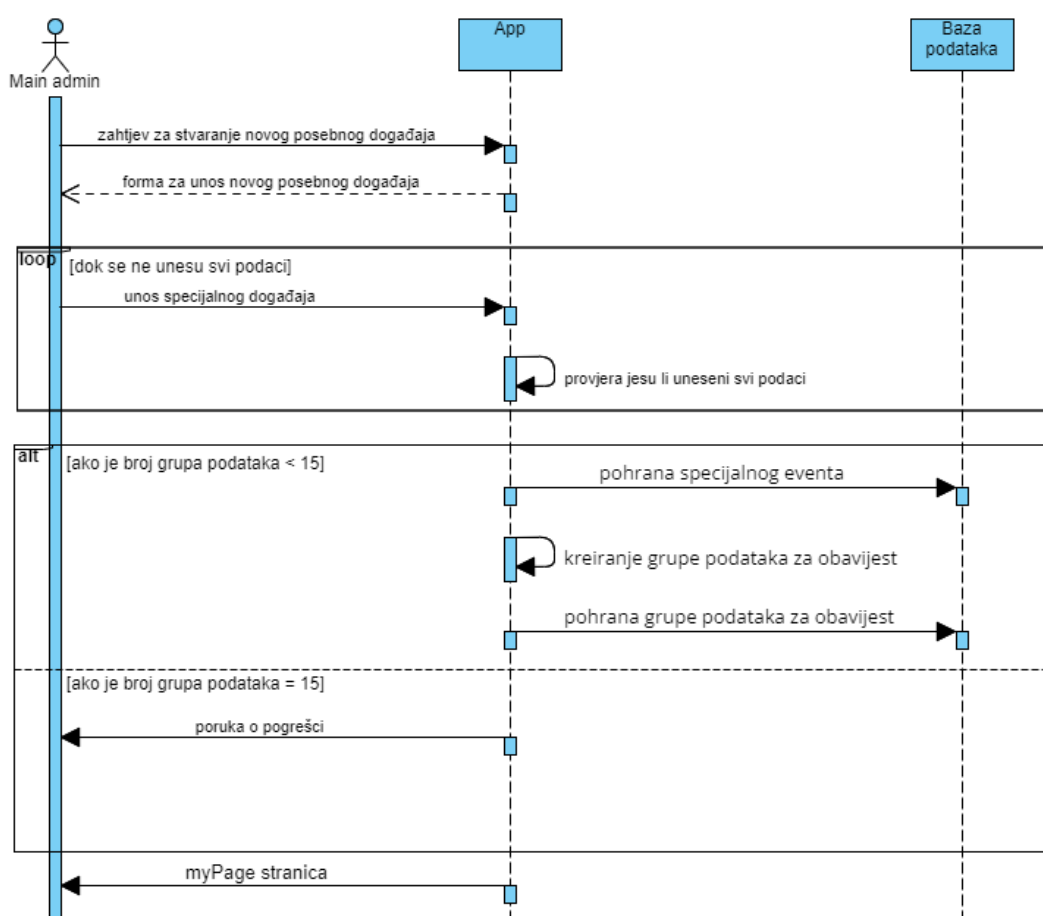
Visual Paradigm Online Free Edition

Slika 3.3: Unos grupe podataka

UC15: Stvaranje posebnog događaja

Glavni admin stvara posebna događanja za konferenciju. S obzirom da se stvaranjem posebnog događaja stvara i grupa podataka tj. obavijest, provjerava se je li dosegnut maksimalan broj grupa podataka. Ukoliko nije, stvara se posebni događaj i obavijest o tom posebnom događaju, u suprotnom se obavještava admina o pogrešci.

Visual Paradigm Online Free Edition



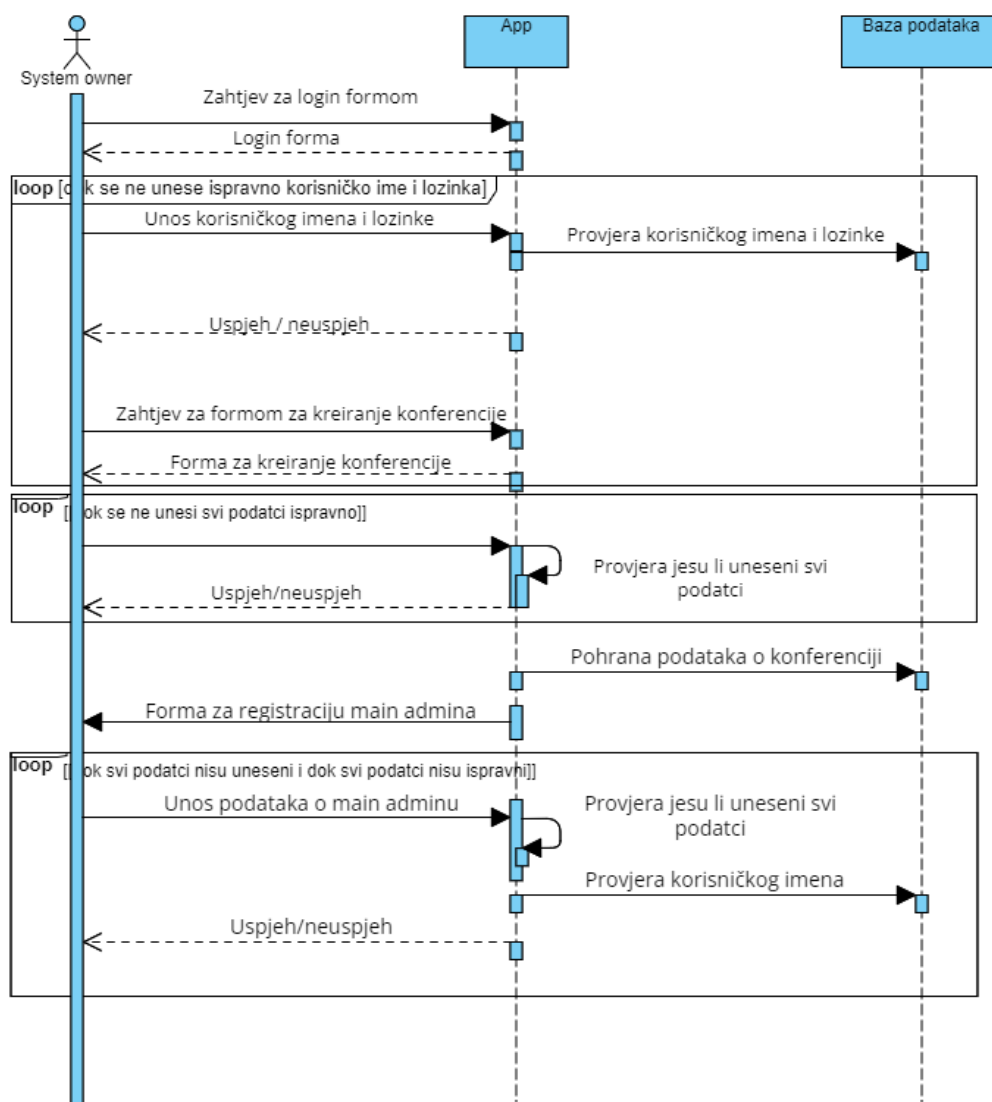
Visual Paradigm Online Free Edition

Slika 3.4: Unos posebnog događaja

UC5: Prijava u sustav, UC6: Stvaranje konferencije, UC2: Registracija glavnog admina

Vlasnik sustava se prijavljuje u aplikaciju. Ako je prijava uspješna, vlasnik sustava dobiva pristup aplikaciji, inače dobiva poruku o neuspjeloj prijavi i može pokušati opet. Vlasnik sustava odabire opciju kreiranja konferencije i unosi glavne podatke o konferenciji. Ako nisu uneseni svi podatci, sustav to dojavljuje vlasniku. Ako je sve ispravno uneseno, stvara se nova konferencija i vlasnik sustava je preusmjeren na formu za registraciju glavnog admina te konferencije. Unosi podatke o glavnom adminu i pridružuje ga upravo napravljenom konferenciji. Ako je sve ispravno uneseno, stvoren je račun glavnog admina i vlasnik sustava je preusmjeren na menu page, inače dobiva poruku o netočno unesenim podacima.

Visual Paradigm Online Free Edition



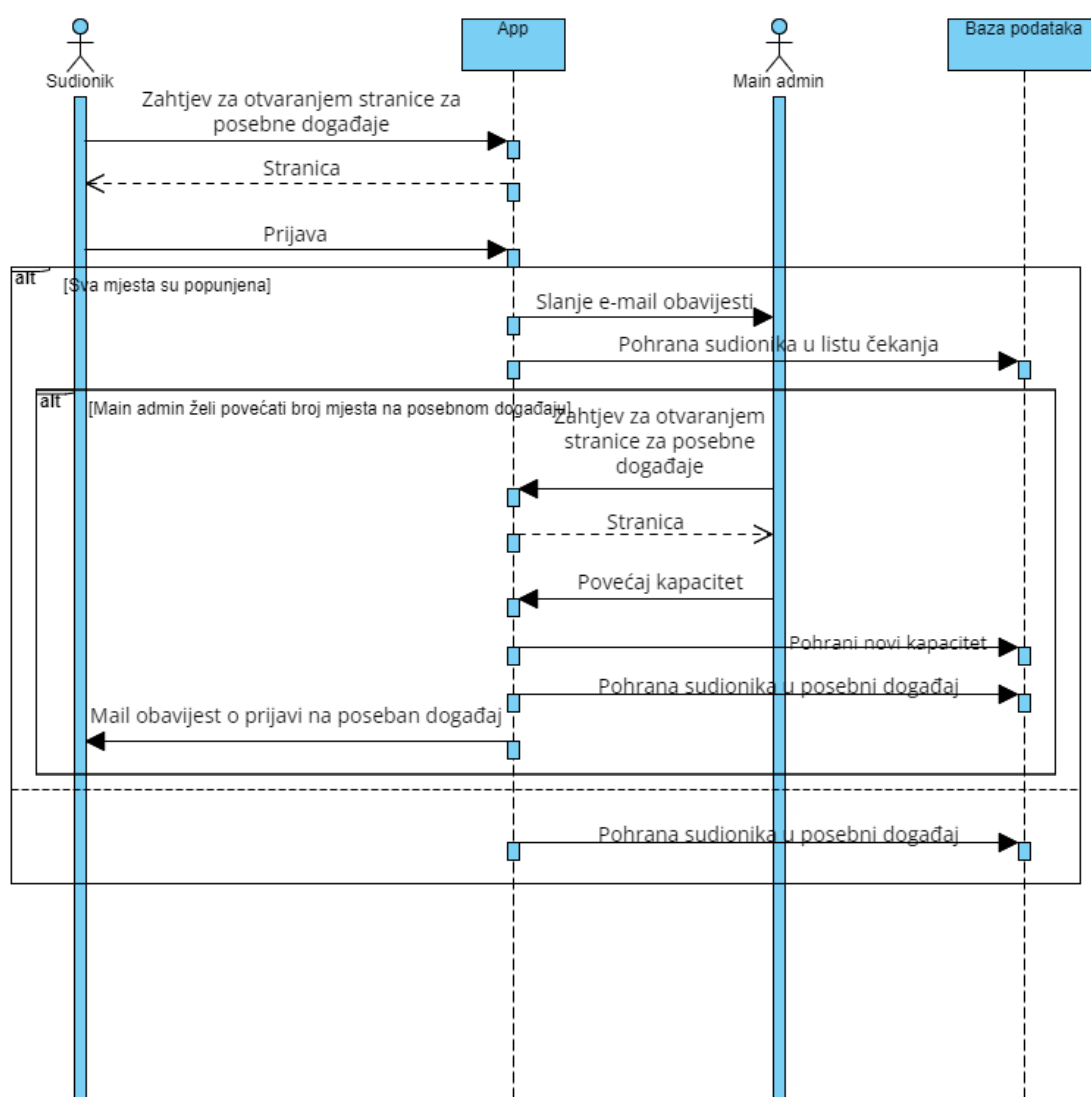
Visual Paradigm Online Free Edition

Slika 3.5: Stvaranje konferencije i main admina konferencije

UC12: Prijava na poseban događaj

Sudionik konferencije odabire opciju prijave na poseban događaj. Ako ima još slobodnih mjesta, korisnik postaje sudionik tog posebnog događaja, a broj slobodnih mjesta se umanjuje za jedan. Ako nema slobodnih mjesta, sudionik se stavlja u listu čekanja, a sustav šalje mail obavijest glavnom adminu konferencije da su sva mjesta na specijalnom događaju popunjena. Nakon toga, glavni admin može povećati broj mjesta. Ako to učini, sustav šalje mail obavijest korisniku da je primljen na poseban događaj te ažurira broj slobodnih mjesta.

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 3.6: Prijava na poseban događaj

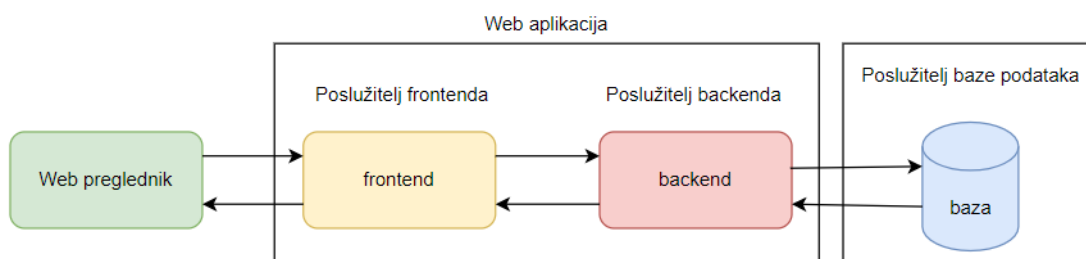
3.2 Nefunkcionalni zahtjevi

- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS
- Sustav treba omogućiti rad više korisnika u stvarnom vremenu, preko 500
- Sustav treba biti izdržljiv na neispravno korištenje
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Veza s bazom podataka mora biti zaštićena i otporna na vanjske greške
- Učitavanje prikaza ne smije trajati duže od nekoliko sekundi
- Pristup sustavu treba biti omogućen samo registriranim korisnicima i zaštićen od vanjskih sudionika
- Tokeni za potvrdu e-mail adrese traju 24 sata
- Tokeni za autentifikaciju traju 1 sat
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Korisnički podaci trebaju biti sigurno pohranjeni i odgovarajuće enkriptirani
- Korisničko sučelje treba biti jednostavno, intuitivno i pregledno, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Korisnicima su podatci o njihovoj konferenciji dostupni do 30 dana nakon njenog završetka
- Glavni administrator do 40 dana nakon završetka konferencije može spremati podatke
- Glavnom administratoru zabranjeno je stvaranje specijalnog događaja ili grupe podataka za konferenciju koja već ima 15 grupa podataka

4. Arhitektura i dizajn sustava

Budući da je glavna namjera sustava da funkcionira putem interneta i ima što jednostavniju uporabu od strane korisnika odlučile smo se za samostalnu jednos-traničnu web aplikaciju. Arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijских sadržaja vezanih uz njih. Svaki internetski preglednik je prevoditelj. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj osnova je rada web aplikacije. Njegova primarna zadaća je komunikacija s aplikacijom koja se odvija preko HTTPS (engl. Hyper Text Transfer Protocol Secure) protokola, što je protokol u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjev.

Korisnik koristi web aplikaciju za obrađivanje željenih zahtjeva. Web aplikacija obrađuje zahtjev te ovisno o zahtjevu, pristupa bazi podataka nakon čega preko poslužitelja vraća korisniku odgovor u web pregledniku.

Programski jezik kojeg smo odabrale za izradu naše web aplikacije je Java u okviru Spring Boot te programski jezik JavaScript u React razvojnoj biblioteci. Oda-brana razvojna okruženja su IntelliJ, Eclipse i VSCode. Arhitektura web aplikacije

je slojevita što omogućava nezavisan razvoj pojedinog dijela aplikacije, lakše ispitivanje i održavanje aplikacije te vrlo jednostavno dodavanje novih značajki u sustav. Sastoji se od pristupne i pozadinske aplikacije.

Pristupna aplikacija (frontend) preko Rest API komunicira s pozadinskom aplikacijom.

Pozadinska aplikacija (backend) je troslojno ustrojena:

- Repository
- Service
- Controller

Controller predstavlja sloj aplikacije koji obrađuje HTTP zahtjeve i priprema podatke za prikaz u JSON datoteke. Service je središnji sloj koji izravno upravlja podacima, logikom i pravilima aplikacije te obavlja autentikaciju i validaciju. Repository šalje SQL upite bazi podataka, odnosno obavlja CRUD (Create, Retrieve, Update and Delete) operacije.

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka čija je gradivna jedinka relacija, odnosno dvodimenzionalna imenovana tablica sa skupom atributa - imenovanim stupcima tablice. Relacijska baza podataka omogućava jednostavno pohranjivanje i upravljanje podacima. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- UserAccount
- Conference
- Multimedia
- SpecialEvent
- DataGroup
- conferenceDataGroup
- operationalAdmins

- pending
- attendees
- users

4.1.1 Opis tablica

UserAccount Ovaj entitet opisuje korisnika konferencije i njegov račun s kojim se prijavljuje u aplikaciju. Sadrži attribute idUserAccount, address, companyName, country, detailsOfParticipation, email, enabled, firstAndLastName, isMainAdmin, isOperativeAdmin, isParticipant, isSystemOwner, phoneNumber, username, password i idConference. Postoje četiri vrste korisnika, vlasnik sustava, glavni admin, operativni admin te sudionik konferencije.

UserAccount		
idUserAccount	INT	jedinstveni identifikator korisnika
address	VARCHAR	adresa
companyName	VARCHAR	ime tvrtke
country	VARCHAR	država
detailsOfParticipation	VARCHAR	uloga
email	VARCHAR	email
enabled	BOOLEAN	potvrđen račun
firstAndLastName	VARCHAR	ime i prezime
isMainAdmin	BOOLEAN	je li glavni admin
isOperativeAdmin	BOOLEAN	je li operativni admin
isParticipant	BOOLEAN	je li participant
isSystemOwner	BOOLEAN	je li vlasnik sustava
phoneNumber	VARCHAR	broj mobitela
username	VARCHAR	korisničko ime
password	VARCHAR	lozinka

Conference Ovaj entitet opisuje konferenciju. Sadrži attribute idConference,

city, country, dateStart, dateEnd, description, topics, name, mainAdminIdUserAccount i systemOwnerIdUserAccount. Ovaj entitet je u dvije veze *One-to-Many* sa entitetom UserAccount preko atributa idConference, jedna veza predstavlja operativne admina, a druga participante konferencije, također je u vezi *One-to-Many* sa entitetom dataGroup preko atributa idConference, *One-to-Many* sa entitetom SpecialEvent preko atributa idConference te je u vezi *One-to-Many* sa entitetom Multimedia preko atributa idConference.

Conference		
idConference	INT	jedinstveni identifikator konferencije
city	VARCHAR	grad u kojem se održava
country	VARCHAR	država u kojoj se održava
dateStart	DATE	datum početka konferencije
dateEnd	DATE	datum završetka konferencije
description	VARCHAR	kratak opis konferencije
topics	VARCHAR	teme
name	VARCHAR	naziv grupe podataka
mainAdminIdUserAccount	INT	id glavnog admina sustava
systemOwnerIdUserAccount	INT	id vlasnika sustava

Multimedia Ovaj entitet opisuje multimedijske sadržaje snimane od strane ovlaštenog fotografa za određenu konferenciju. Sadrži attribute idMultimedia, url, date i idConference. Ovaj entitet je u vezi *Many-to-One* sa entitetom Conference preko atributa idConference.

Multimedia		
idMultimedia	INT	jedinstveni identifikator multimedijskog sadržaja
url	VARCHAR	link na drive gdje je pohranjena multimedia

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Multimedia		
date	DATETIME	datum nastanka
idConference	INT	id konferencije kojoj pripada

SpecialEvent Ovaj entitet opisuje posebna događanja tijekom neke konferencije. Sadrži attribute idSpecialEvent, capacity, type, message i idConference. Ovaj entitet je u vezi *Many-to-One* sa entitetom Conference preko atributa idSpecialEvent te je u dvije veze *Many-to-Many* sa entitetom UserAccount preko atributa idSpecialEvent. Jedna veza opisuje korisnike koji su prijavljeni za sudjelovanje na posebnom događaju, a druga opisuje korisnike koji su na čekanju za sudjelovanje zbog popunjenih kapaciteta.

SpecialEvent		
idSpecialEvent	INT	jedinstveni identifikator posebnog događaja
capacity	INT	maksimalan broj sudionika
type	VARCHAR	tip događaja
message	VARCHAR	kratka poruka o događaju
idConference	INT	id konferencije kojoj pripada

DataGroup Ovaj entitet opisuje grupu podataka o nekoj konferenciji (raspored, zbornik radova i sl.). Sadrži attribute idDataGroup, groupName, data i idConference. Ovaj entitet je u vezi *Many-to-One* sa entitetom Conference preko atributa idDataGroup.

DataGroup		
idDataGroup	INT	jedinstveni identifikator grupe podataka
data	VARCHAR	podatak za određenu grupu podataka (može biti tekst ili datoteka)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

DataGroup		
groupName	VARCHAR	naziv

ConferenceDataGroup Ovaj entitet opisuje odnos između entiteta Conference i DataGroup. Sadrži attribute idDataGroup i idConference. Ovaj entitet je u vezi *Many-to-One* sa entitetom Conference preko atributa idConference te u vezi *Many-to-One* sa entitetom DataGroup preko atributa idDataGroup.

ConferenceDataGroup		
idDataGroup	INT	jedinstveni identifikator grupe podataka
idConference	INT	jedinstveni identifikator konferencije

OperationalAdmins Ovaj entitet opisuje odnos između entiteta Conference i UserAccount, a predstavlja operativne admina. Sadrži attribute idUserAccount i idConference. Ovaj entitet je u vezi *Many-to-One* sa entitetom Conference preko atributa idConference te u vezi *Many-to-One* sa entitetom UserAccount preko atributa idUserAccount.

OperationalAdmins		
idUserAccount	INT	jedinstveni identifikator operativnog admina
idConference	INT	jedinstveni identifikator konferencije

Users Ovaj entitet opisuje odnos između entiteta Conference i UserAccount, a predstavlja participante konferencije. Sadrži attribute idUserAccount i idConference. Ovaj entitet je u vezi *Many-to-One* sa entitetom Conference preko atributa idConference te u vezi *Many-to-One* sa entitetom UserAccount preko atributa idUserAccount.

Users		
idUserAccount	INT	jedinstveni identifikator participanta konferencije
icConference	INT	jedinstveni identifikator konferencije

VerificationToken Ovaj entitet opisuje tokene koji se koriste pri potvrdi računa mailom. Ovaj entitet je u vezi *Many-to-One* sa entitetom UserAccount preko atributa idUserAccount.

VerificationToken		
idToken	INT	jedinstveni identifikator tokena
createdDate	DATE	datum i vrijeme nastanka tokena
expiryDate	DATE	datum i vrijeme prestanka valjanosti tokena
token	VARCHAR	token
idUserAccount	INT	id korisničkog računa za kojeg je token generiran

SpecialEventAttendees Ovaj entitet opisuje odnos između entiteta SpecialEvent i UserAccount, a predstavlja participante koji su prijavljeni na neki specijalni događaj. Sadrži attribute idUserAccount i idSpecialEvent. Ovaj entitet je u vezi *Many-to-One* sa entitetom SpecialEvent preko atributa idSpecialEvent te u vezi *Many-to-One* sa entitetom UserAccount preko atributa idUserAccount.

SpecialEventAttendees		
idUserAccount	INT	jedinstveni identifikator participanta konferencije

Nastavljeno na idućoj stranici

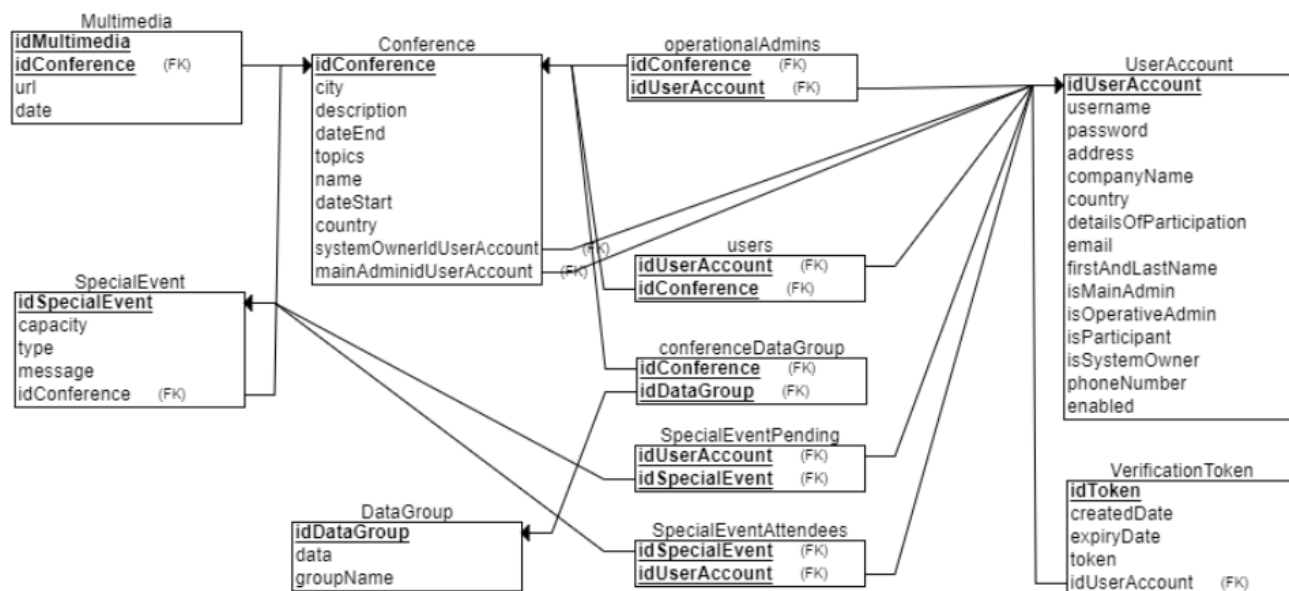
Nastavljeno od prethodne stranice

SpecialEventAttendees		
idSpecialEvent	INT	jedinstveni identifikator specijalnog događaja

SpecialEventPending Ovaj entitet opisuje odnos između entiteta SpecialEvent i UserAccount, a predstavlja participante koji su se htjeli prijaviti na specijalni događaj, ali su stavljeni na čekanje zbog popunjenih kapaciteta na tom događaju. Sadrži attribute idUserAccount i idSpecialEvent. Ovaj entitet je u vezi *Many-to-One* sa entitetom SpecialEvent preko atributa idSpecialEvent te u vezi *Many-to-One* sa entitetom UserAccount preko atributa idUserAccount.

SpecialEventPending		
idUserAccount	INT	jedinstveni identifikator participanta konferencije
idSpecialEvent	INT	jedinstveni identifikator specijalnog događaja

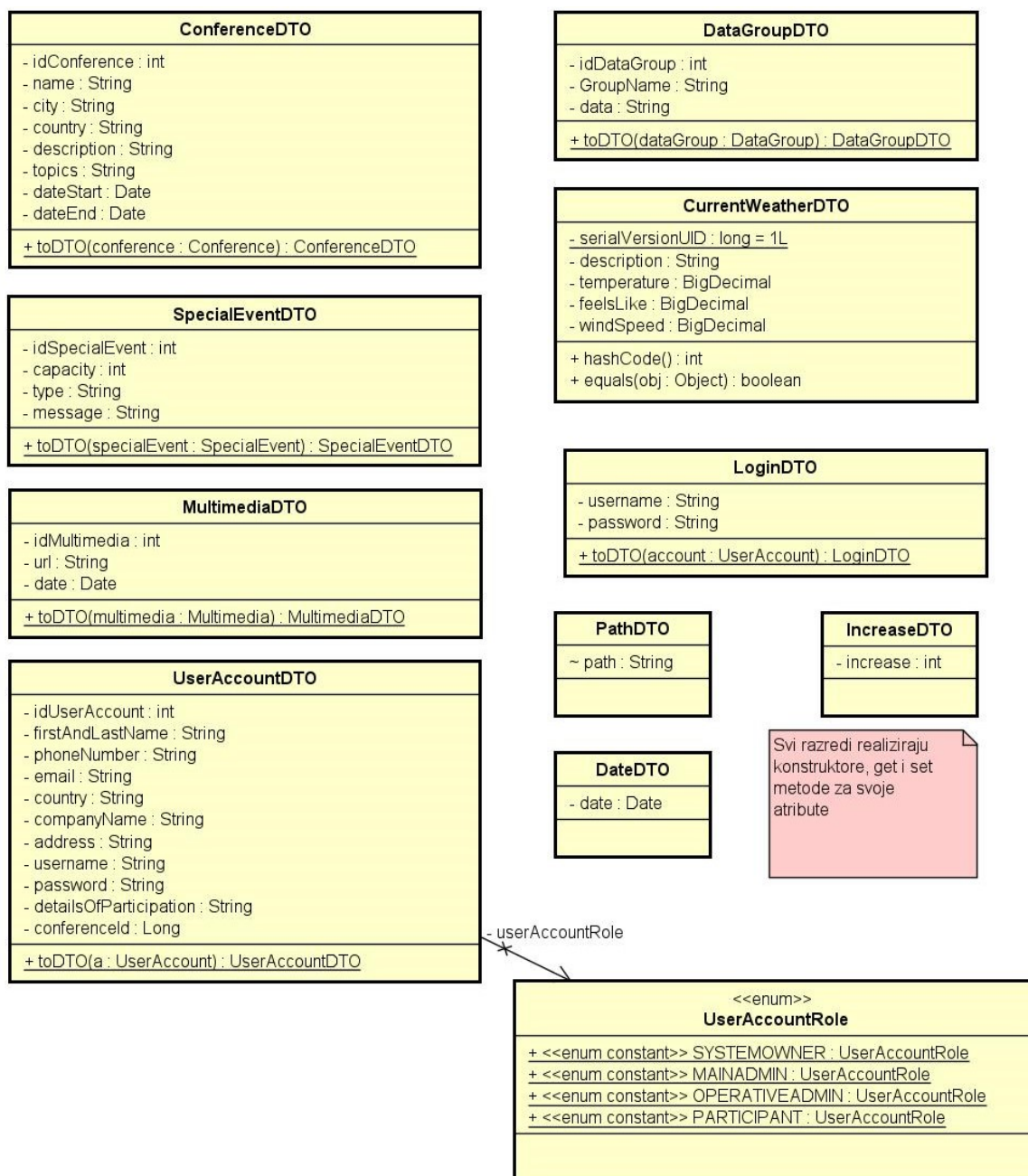
4.1.2 Dijagram baze podataka



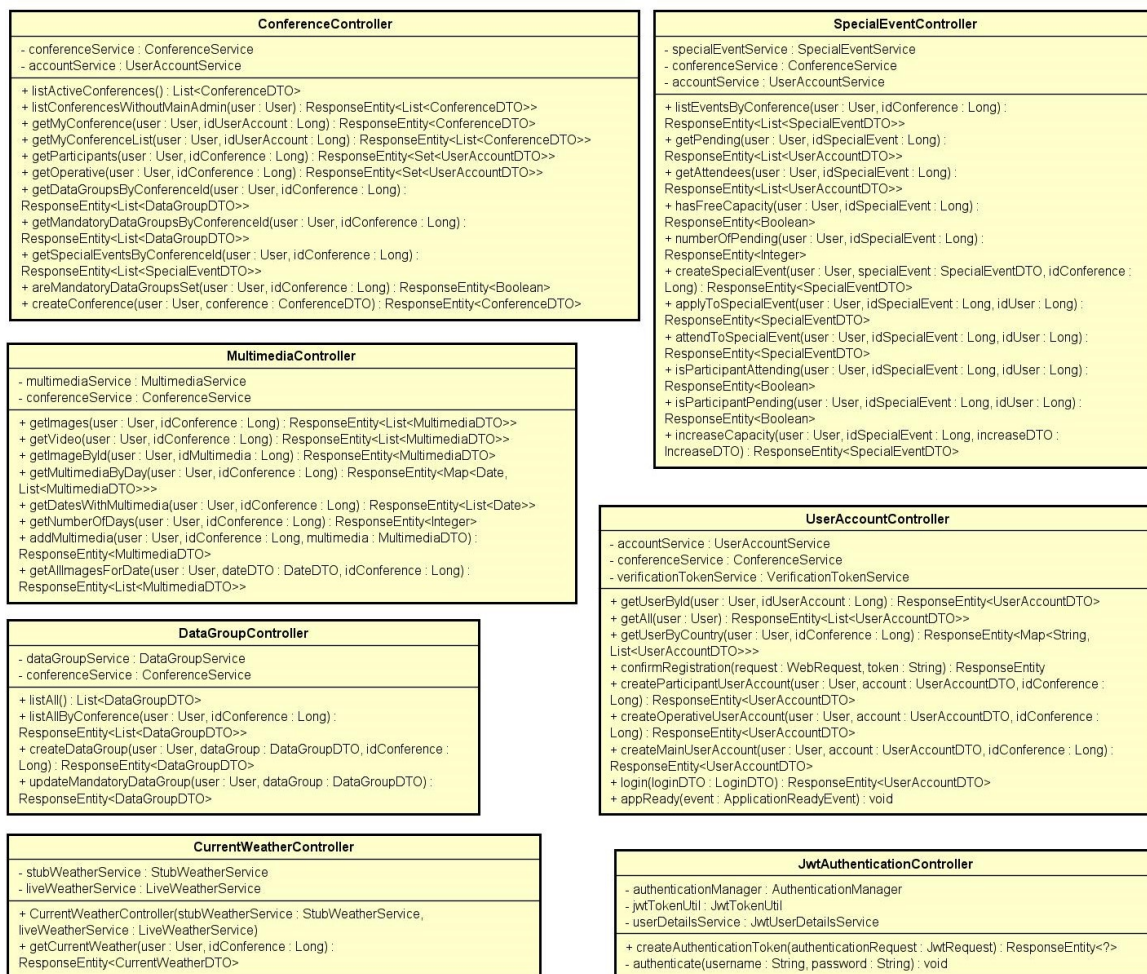
Slika 4.2: Relacijski model baze

4.2 Dijagram razreda

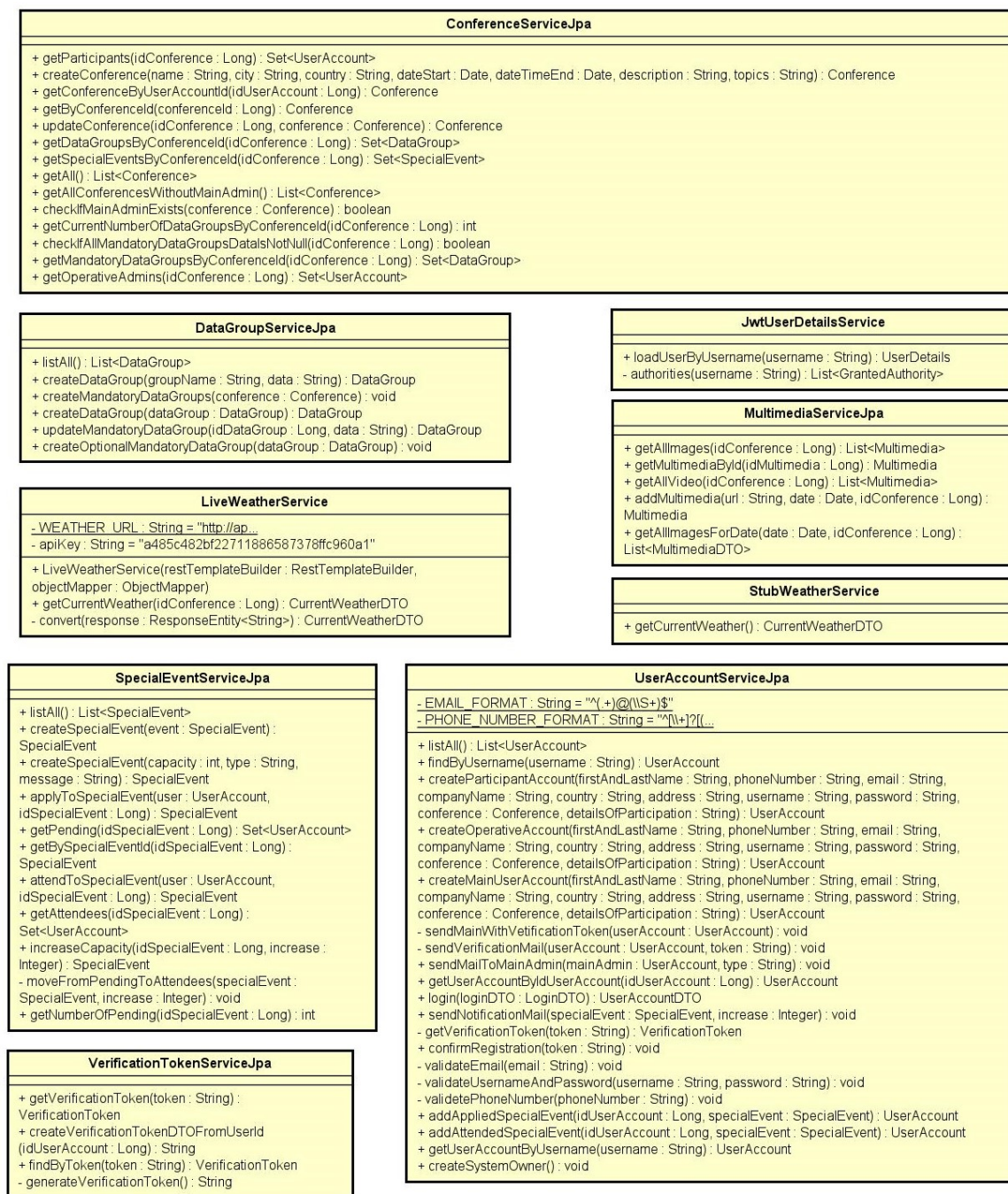
Na slikama 4.3, 4.4, 4.5 i 4.6 su prikazani razredi koji pripadaju backend dijelu arhitekture. Slika 4.3 prikazuje razred DTO (*Data Transfer Object*) razrede. Na slici 4.4. prikazan je dijagram Modela, oni predstavljaju entitete naše baze podataka, na tom dijagramu također možemo vidjeti i međusobnu ovisnost entiteta. Zatim slike 4.5 i 4.6 prikazuju Kontrolere i Service, metode u Kontrolerima manipuliraju s DTO podacima, primaju zahtjeve s frontend-a koje dalje šalje na Service gdje se zahtjev obrađuje. Povratna informacija koja se generira u Service se ponovno vraća preko Kontrolera koji vraća JSON datoteku na frontend.



Slika 4.3: Dijagram razreda - DTO



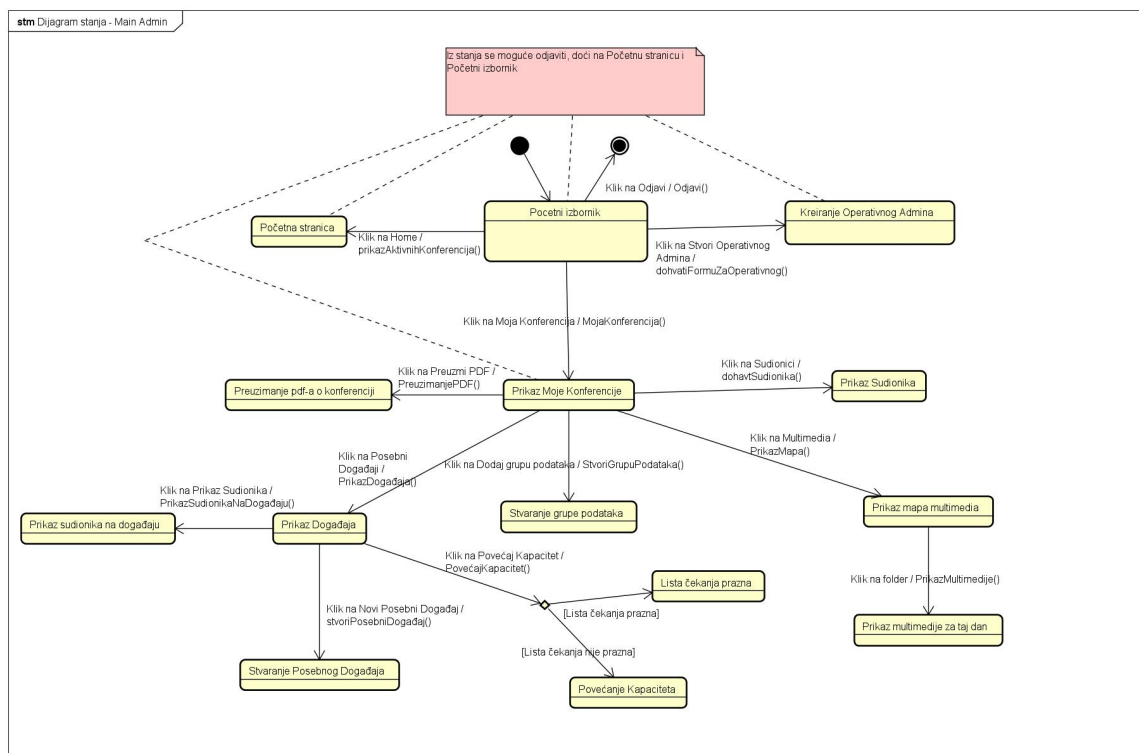
Slika 4.5: Dijagram razreda - Kontroleri



Slika 4.6: Dijagram razreda - Service

4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.7 prikazan je dijagram stanja za registriranog main admina. Nakon prijave prvo se prikazuje početni izbornik koji nudi tri opcije : *Home*, *My Conference* i *Create Operative Admin*. Klikom na "Home" poziva se početna stranica na kojoj se prikazuju aktivne konferencije. Klikom na "Create Operative Admin" prikaže se forma za kreiranje operativnog admina. Klikom na "My Conference" prikažu se podatci o konferenciji. Main admin tu ima mogućnosti stvoriti novu grupu podataka klikom na "New Data Group". Ima mogućnost pregleda svih sudionika i operativnih admina klikom na *Show Attendees*. Klikom na "Special Events" main admin dobiva prikaz svih događaja za tu konferenciju gdje može odabrati stvaranje novog događaja klikom na "New Special Event". Također može povećati kapacitet za određeni događaj. Ako je lista čekanja prazna, povećanje kapaciteta će biti odbijeno, ako lista čekanja nije prazna kapacitet će se povećati. Iz stanja *Prikaz Moje Konferencije* može pristupiti i multimediji klikom na "See Multimedia", dobiva prikaz direktorija raspoređeni po danima, klikom na direktorij prikažu se slike tog dana.

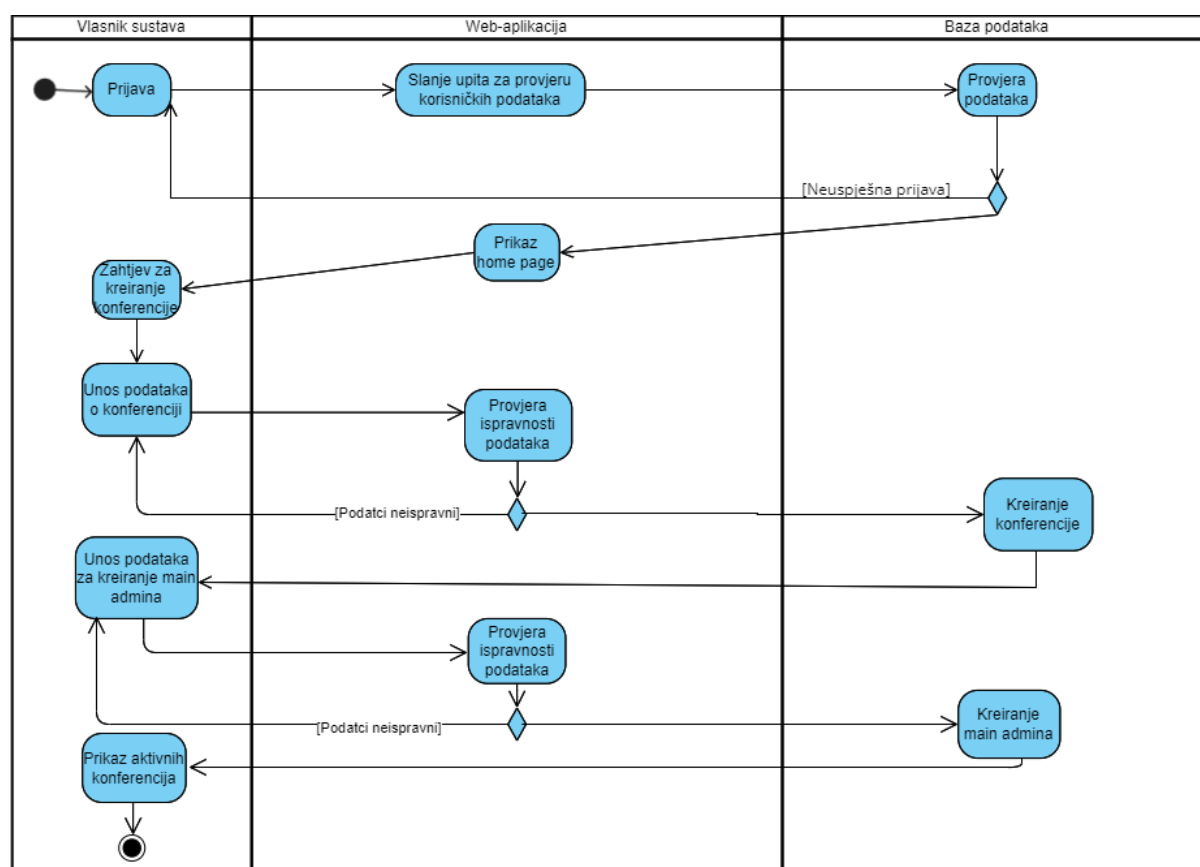


Slika 4.7: Dijagram stanja - Main Admin

4.4 Dijagram aktivnosti

Vlasnik sustava - kreiranje konferencije i glavnog admina

Na dijagramu aktivnosti prikazan je proces stvaranja konferencije i main admina. Vlasnik sustava prijavljuje se u sustav, odabire gumb za kreiranje konferencije, te unosi podatke o konferenciji. Nakon toga otvara se obrazac za unos podataka o main adminu.

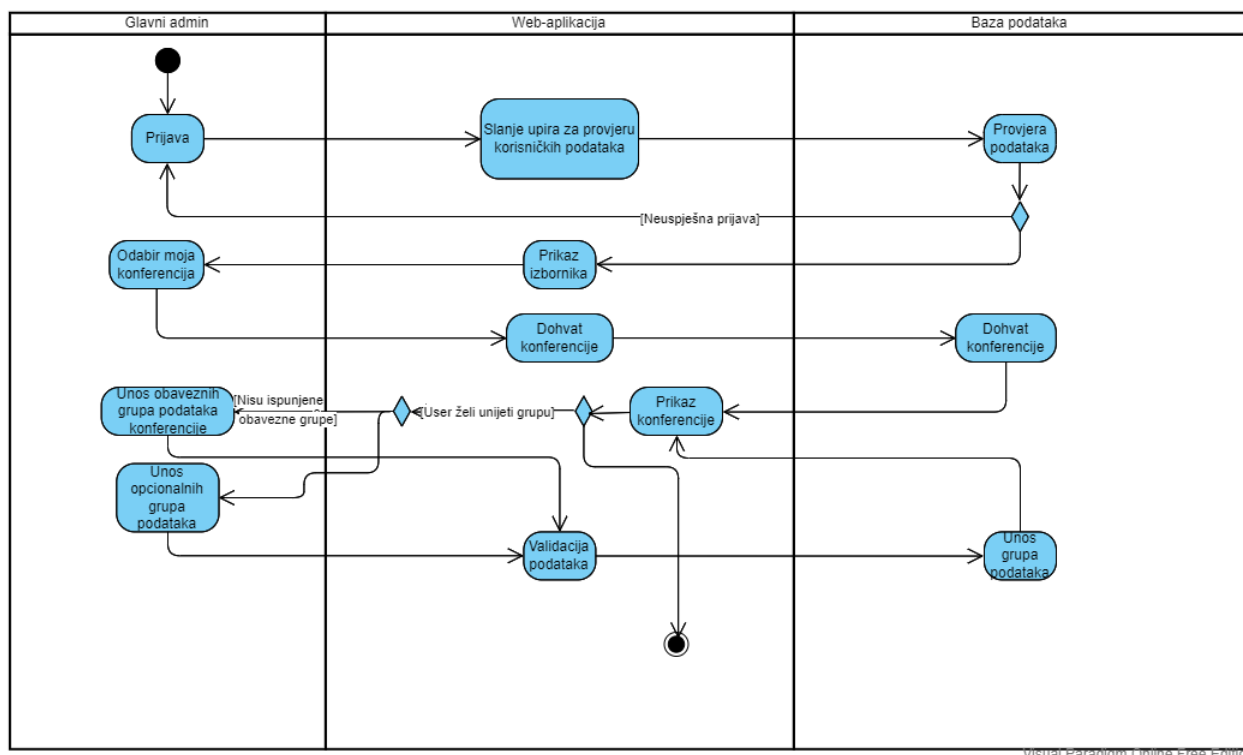


Slika 4.8: Kreiranje konferencije i glavnog admina

Glavni admin - unos grupa podataka

Na dijagramu aktivnosti prikazan je proces unosa grupa podataka. Glavni admin prijavljuje se u sustav, odabire gumb "My conference", te unosi obavezne grupe podataka. Nakon toga može stvarati i opcionalne grupe po dataka.

Visual Paradigm Online Free Edition

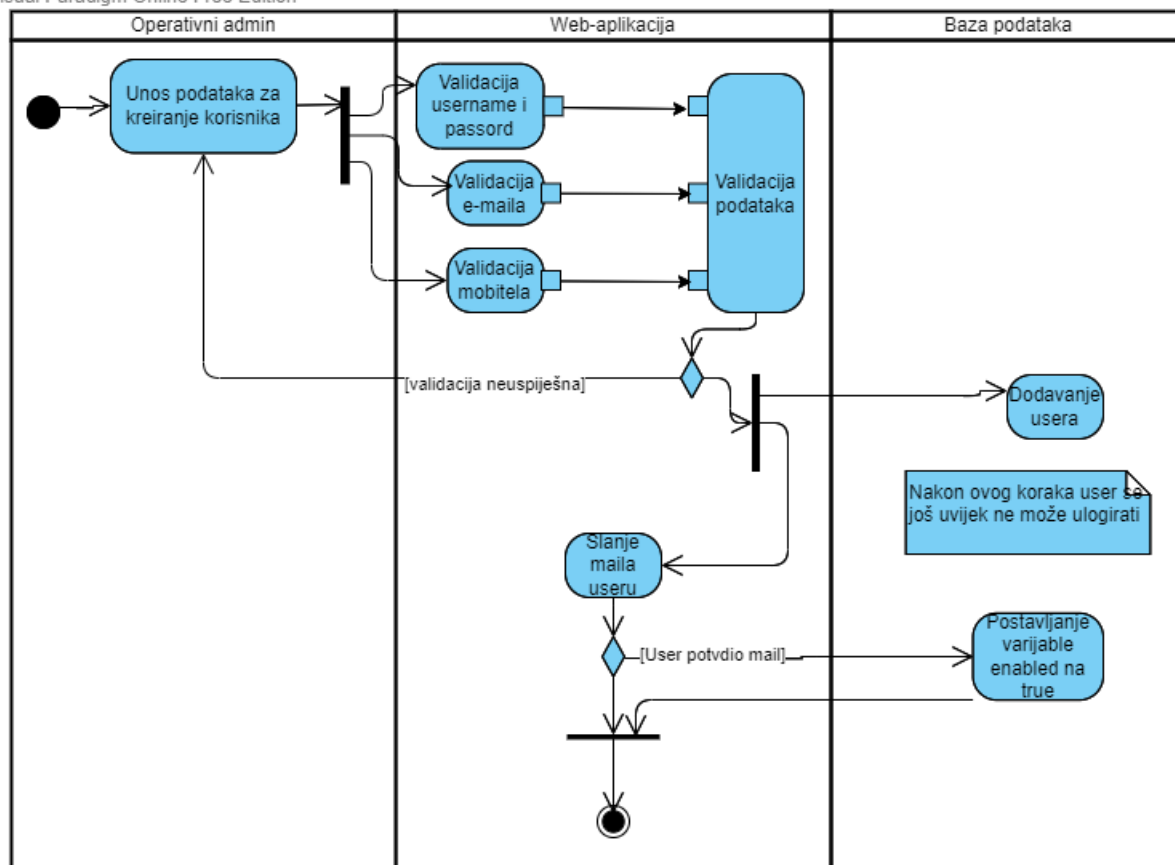


Slika 4.9: Unos grupa podataka

Operativni admin - kreiranje korisnika

Na dijagramu aktivnosti prikazan je proces kreiranja korisnika. Operativni admin prijavljuje se u sustav, odabire gumb "Create user", te mu se otvara obrazac za popunjavanje podataka o korisniku. Nakon uspješnog unosa korisnika, korisnik mora potvrditi svoj korisnički račun mailom.

Visual Paradigm Online Free Edition

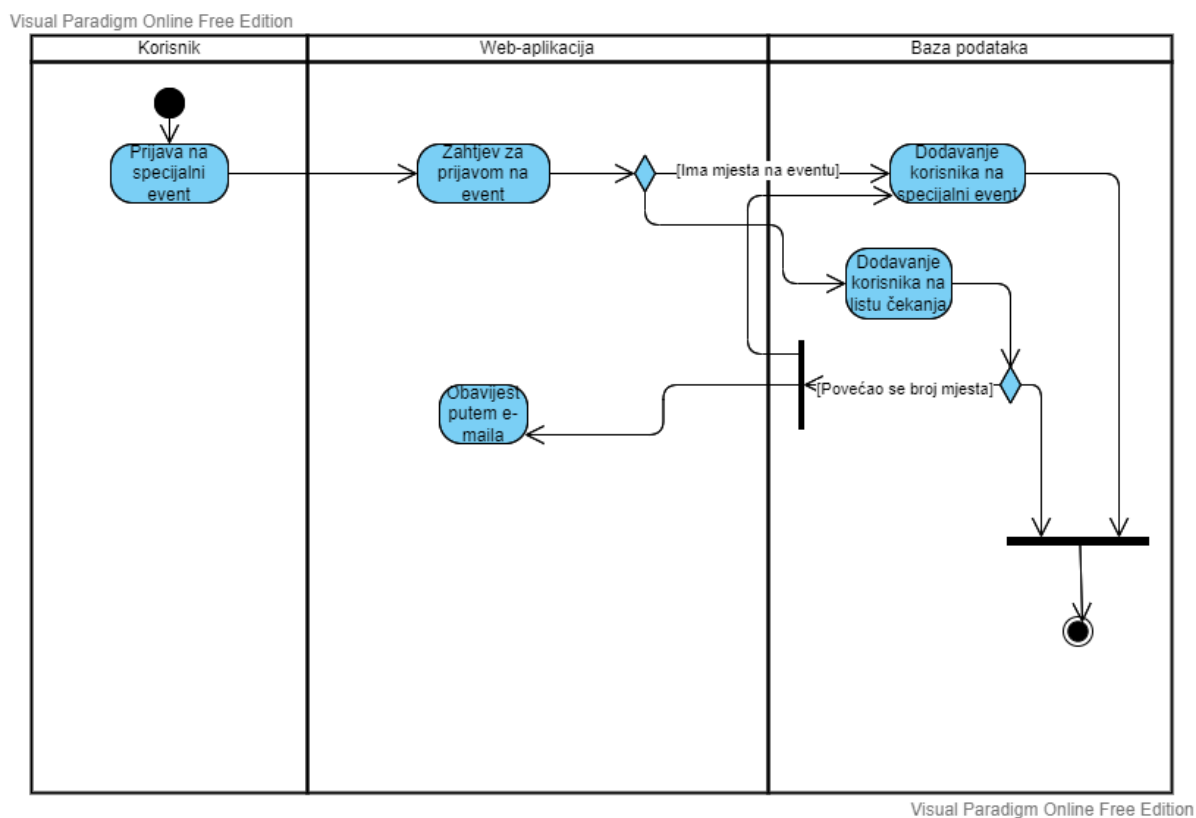


Visual Paradigm Online Free Edition

Slika 4.10: Kreiranje korisnika

Korisnik - prijava na specijalni event

Na dijagramu aktivnosti prikazan je proces prijave korisnika na specijalni event. Kada korisnik odabere gumb "Special event", otvara mu se lista specijalnih eventova na koje se može prijaviti. Ako na specijalnom eventu ima mjesta, korisnik se automatski dodaje na listu prijavljenih korisnika za specijalni event. U slučaju da su sva mjesta na specijalnom eventu popunjena, korisnika se o tome obavještava. U slučaju da se naknadno poveća broj slobodnih mjesta na specijalnom eventu, korisnika se mailom obavještava da je sada na listi prijavljenih korisnika.



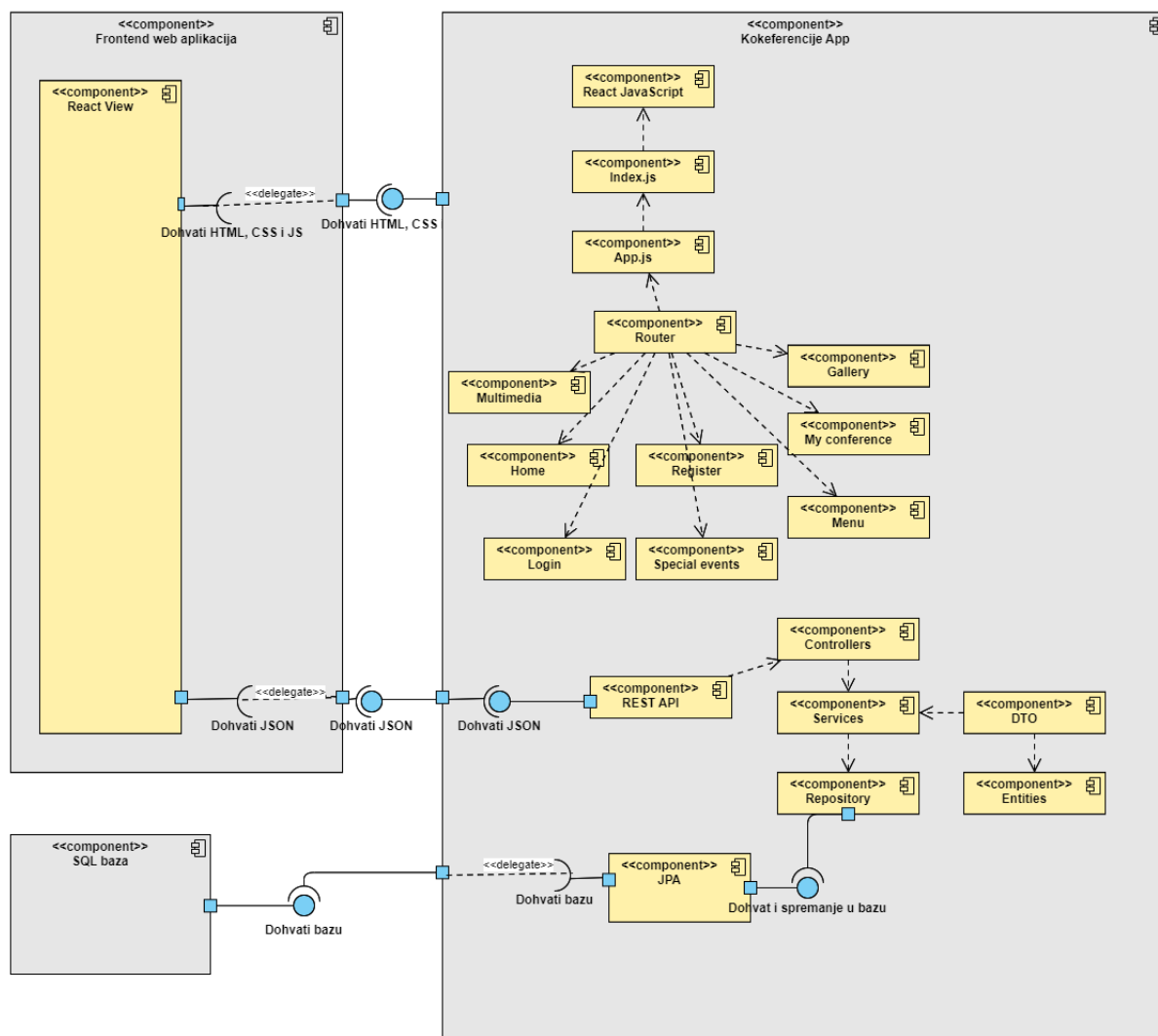
Slika 4.11: Prijava na specijalni event

4.5 Dijagram komponenti

Dijagram komponenti

Prikazani dijagram komponenti prikazuje organizaciju i međuovisnost komponenti i odnose prema okolini. Sustavu se pristupa preko dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju frontend dijelu aplikacije. To je sučelje zahtjevano sučelje za React View komponentu budući da preko njega ostvaruje prikaz i osvježavanje frontenda. Router je komponenta koja na upit s url-om određuje koja datoteka će se poslužiti na sučelje. Svaka datoteka predstavlja jednu stranicu aplikacije, tj. JavaScript kod te datoteke. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente kao što su gumbi, forme i slično. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju backend dijelu aplikacije. JPA(Java persistence API) je zadužen za dohvaćanje tablica iz baze podataka pomoću SQL upita. Sučelje za komunikaciju s JPA koristi Repository razred Swing aplikacije kako bi na zahtjev iz Service komponente mogla poslati upit bazi podataka. Podatci koji su pristigli iz baze se šalju dalje arhitekturi u obliku DTO (Data transfer object) iz Service prema Controlleru i dalje prema frontend dijelu aplikacije. Reactview komponenta preko dostupnih sučelja komunicira sa Kokeferencije aplikacijom te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke.

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 4.12: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Tim je komunicirao putem aplikacija WhatsApp¹ te Discord². WhatsApp je besplatna aplikacija za dopisivanje te je prvenstveno služila za međusobne dogovore kao i razrješavanje pojedinih nejasnoća. Discord je aplikacija i platforma za digitalnu distribuciju, dizajnirana prvenstveno za zajednice koje se bave igranjem video igara. Specijalizirana je za tekstualnu, slikovnu, video i audio komunikaciju između korisnika što je uvelike olakšalo komunikaciju prilikom procesa izrade aplikacije. Za izradu UML dijagrama korišten je alat Astah Professional³ te Visual Paradigm⁴. Kao sustav za upavljanje izvornim kodom koristio se Git⁵, a udaljeni repozitorij projekta je dostupan na web platformi Gitlab⁶.

Kao razvojna okruženja korišteni su IntelliJ IDEA⁷, Eclipse⁸ te Microsoft Visual Studio⁹. IntelliJ IDEA je integrirano razvojno okruženje (IDE) napisano u Javi za razvoj računalnog softvera napisanog u Javi, Kotlinu, Groovyju i drugim jezicima koji se temelje na JVM-u. Razvio ga je JetBrains (ranije poznat kao IntelliJ) i dostupan je kao Apache 2 licencirano izdanje zajednice i u vlasničkom komercijalnom izdanju. Oba se izdanja mogu koristiti za komercijalni razvoj. Eclipse je integrirano razvojno okruženje koje se koristi u računalnom programiranju. Sadrži osnovni radni prostor i proširivi sustav za prilagođavanje okruženja. To je drugi najpopularniji IDE za Java razvoj, a do 2016. bilo je najpopularniji. Visual Studio je integrirano razvojno okruženje tvrtke Microsoft. Koristi se za razvoj računalnih programa uključujući web stranice, web aplikacije, web usluge i mobilne aplikacije.

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://astah.net/products/astah-professional/>

⁴<https://www.visual-paradigm.com/>

⁵<https://git-scm.com/>

⁶<https://gitlab.com/>

⁷<https://www.jetbrains.com/idea/>

⁸<https://www.eclipse.org/>

⁹<https://visualstudio.microsoft.com/>

Aplikacija je napisana pomoću radnog okvira Spring framework¹⁰ i jezika Java¹¹ za izradu *backenda*, a za izradu *frontenda* koristio se React¹² te jezik Javascript¹³. Spring framework omogućava jednostavan razvoj modernih aplikacija u Javi. Nudi brojne značajke koje su organizirane u module. Središnji modul je Spring spremnik čija je dužnost povezivanje, konfiguriranje, stvaranje i upravljanje životnim ciklusom pojedinog objekta, tj. Spring bean-a. Za razvoj web aplikacija veliku važnost ima Spring MVC modul. On kombinira sve prednosti *Model-View-Controller* obrasca i Spring okvira. Osim Spring modula postoje i Spring projekti, a oni nude sve značajke za određenu funkcionalnost. React je Javascript biblioteka otvorenog koda koja se temelji na lakšem dizajniranju sučelja. Odgovoran je za prikaz slojeva u aplikaciji, a napravio ga je Facebook. React omogućuje programiranje i razvoj aplikacija te je važan dio za *frontend development*.

Za razvoj i testiranje HTTP REST API-ja korišten je alat Postman¹⁴, a baza podataka se prije procesa *deploymenta* nalazila lokalno što je ostvareno pomoću PostgreSQL-a¹⁵, sustava za upravljanje relacijskom bazom podataka otvorenog koda. Nakon *deploymenta* baza podataka se nalazi na poslužitelju u oblaku Render¹⁶.

¹⁰<https://spring.io/projects/spring-framework>

¹¹<https://www.java.com/en/>

¹²<https://reactjs.org/>

¹³<https://www.javascript.com/>

¹⁴<https://www.postman.com/>

¹⁵<https://www.postgresql.org/>

¹⁶<https://render.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitni slučaj 1: Login sa postojećim i nepostojećim korisnikom, te korisnikom čija konferencija više nije dostupna.

Ulaz:

1. Username i password postojećeg korisnika
2. Username i password nepostojećeg korisnika
3. Username i password korisnika čija konferencija više nije dostupna

Očekivani rezultat:

1. Login je uspješan za postojećeg korisnika i kao odgovor se vraća korisnički račun usera.
2. Login je neuspješan za nepostojećeg korisnika i kao odgovor se vraća Unauthorised.
3. Login je neuspješan za korisnika čija je konferencija nedostupna i odgovor je Forbidden.

Rezultat: Svaki očekivani rezultat je ostvaren. **Aplikacija je prošla test.**

```
@Test
public void loginTest() throws JSONException {
    RestTemplate restTemplate = new RestTemplate();
    String url = "http://localhost:8080/users/login";
    // nepostojeći user
    Assertions.assertThrows(HttpClientErrorException.Unauthorized.class, () -> {
        LoginDTO a = new LoginDTO("nepostojeći", "password");
        restTemplate.postForEntity(url, a, UserAccountDTO.class);
    });
    // postojeći user
    LoginDTO authorisedLogin = new LoginDTO("systemowner", "password");
    ResponseEntity<UserAccountDTO> account = restTemplate.postForEntity(url, authorisedLogin, UserAccountDTO.class);
    Assertions.assertNotNull(account);
    // user kojem je istekla konferencija
    Assertions.assertThrows(HttpClientErrorException.Forbidden.class, () -> {
        LoginDTO a = new LoginDTO("participant3", "pass123456");
        restTemplate.postForEntity(url, a, UserAccountDTO.class);
    });
}
```

Slika 5.1: Testiranje prijave

Ispitni slučaj 2: Registracija korisnika

Ulaz:

1. Stvaranje konferencije te unos ispravnih podataka o korisniku.

2. Stvaranje konferencije, te unos neispravnih podataka za korisnika (neispravni format e-maila, neispravan format broja mobitela, nedovoljan broj znakova za username, nedovoljan broj znakova za password itd.).

Očekivani rezultat:

1. Registracija je uspješna i kao odgovor se vraća stvoreni korisnički račun.
2. Registracija je neuspješna te se kao odgovor vraća BadRequest.

Rezultat: Svaki očekivani rezultat je ostvaren. **Aplikacija je prošla test.**

```
@Test
public void registerTest() {
    RestTemplate restTemplate = new RestTemplate();
    String tokenUrl = "http://localhost:8080/authenticate";
    JwtRequest req = new JwtRequest("systemowner", "password");
    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_JSON);
    String accessToken = restTemplate.postForObject(tokenUrl, req, String.class);
    headers.set("Authorization", "Bearer " + accessToken.substring(10, accessToken.length() - 2));
    ConferenceDTO conference = new ConferenceDTO(1, "konfa", "grad", "drzava", new Date(2023, 5, 1),
        new Date(2023, 6, 1), "konfa", "neke teme");
    String createKonfa = "http://localhost:8080/";
    HttpEntity<ConferenceDTO> req1 = new HttpEntity<>(conference, headers);
    ResponseEntity<ConferenceDTO> konfa = restTemplate.postForEntity(createKonfa, req1, ConferenceDTO.class, 20);

    UserAccountDTO acc = new UserAccountDTO(1, "Mato matic", "0987654321", "mato@matic.hr", "Cro", "infinum",
        "kbbjd", "matic4", "12345678", "main admin", UserAccountRole.MAINADMIN,
        (long) konfa.getBody().getIdConference());
    headers.setContentType(MediaType.APPLICATION_JSON);
    accessToken = restTemplate.postForObject(tokenUrl, req, String.class);
    headers.set("Authorization", "Bearer " + accessToken.substring(10, accessToken.length() - 2));
    HttpEntity<UserAccountDTO> req2 = new HttpEntity<>(acc, headers);
    String url = "http://localhost:8080/users/createMainAdmin/" + konfa.getBody().getIdConference();
    ResponseEntity<UserAccountDTO> a = restTemplate.postForEntity(url, req2, UserAccountDTO.class,
        Long.valueOf(konfa.getBody().getIdConference()));
    Assertions.assertNotNull(a);
}
```

Slika 5.2: Test ispravne registracije korisnika

```
@Test
public void wrongRegisterTest() {
    RestTemplate restTemplate = new RestTemplate();
    String tokenUrl = "http://localhost:8080/authenticate";
    JwtRequest req = new JwtRequest("systemowner", "password");
    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_JSON);
    String accessToken = restTemplate.postForObject(tokenUrl, req, String.class);
    headers.set("Authorization", "Bearer " + accessToken.substring(10, accessToken.length() - 2));
    ConferenceDTO conference = new ConferenceDTO(1, "konfa", "grad", "drzava", new Date(2023, 5, 1),
        new Date(2023, 6, 1), "konfa", "neke teme");
    String createKonfa = "http://localhost:8080/";
    HttpEntity<ConferenceDTO> req1 = new HttpEntity<>(conference, headers);
    ResponseEntity<ConferenceDTO> konfa = restTemplate.postForEntity(createKonfa, req1, ConferenceDTO.class, 20);

    UserAccountDTO acc = new UserAccountDTO(1, "Mato matic", "0987651", "mato", "Cro", "infinum", "kbbjd", "mat",
        "1234", "main admin", UserAccountRole.MAINADMIN, (long) konfa.getBody().getIdConference());
    headers.setContentType(MediaType.APPLICATION_JSON);
    accessToken = restTemplate.postForObject(tokenUrl, req, String.class);
    headers.set("Authorization", "Bearer " + accessToken.substring(10, accessToken.length() - 2));
    HttpEntity<UserAccountDTO> req2 = new HttpEntity<>(acc, headers);
    String url = "http://localhost:8080/users/createMainAdmin/" + konfa.getBody().getIdConference();
    Assertions.assertThrows(HttpClientErrorException.BadRequest.class, () -> {
        restTemplate.postForEntity(url, req2, UserAccountDTO.class,
            Long.valueOf(konfa.getBody().getIdConference()));
    });
}
```

Slika 5.3: Neispravna registracija korisnika

Ispitni slučaj 3: Kreiranje konferencije

Ulaz:

1. Stvaranje konferencije sa svim ispravnim podacima.
2. Kreiranje konferencije s neispravnim podacima (npr. datum početka je nakon datuma završetka).

Očekivani rezultat:

1. Kreiranje konferencije je uspješno. Kao odgovor se vraća kreirana konferencija.
2. Kreiranje konferencije je neuspješno i kao odgovor se vraća BadRequest.

Rezultat: Svaki očekivani rezultat je ostvaren. **Aplikacija je prošla test.**

```
@Test
public void createConferenceRight() {
    RestTemplate restTemplate = new RestTemplate();
    String tokenUrl = "http://localhost:8080/authenticate";
    JwtRequest req = new JwtRequest("systemowner", "password");
    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_JSON);
    String accessToken = restTemplate.postForObject(tokenUrl, req, String.class);
    headers.set("Authorization", "Bearer " + accessToken.substring(10, accessToken.length() - 2));
    ConferenceDTO conference = new ConferenceDTO(1, "konfa", "grad", "drzava", new Date(2023, 5, 1),
        new Date(2023, 6, 1), "konfa", "neke teme");
    String createKonfa = "http://localhost:8080/";
    HttpEntity<ConferenceDTO> req1 = new HttpEntity<>(conference, headers);
    ResponseEntity<ConferenceDTO> konfa = restTemplate.postForEntity(createKonfa, req1, ConferenceDTO.class, 20);
    Assertions.assertNotNull(konfa);
}

@Test
public void wrongCreateConference() {
    RestTemplate restTemplate = new RestTemplate();
    String tokenUrl = "http://localhost:8080/authenticate";
    JwtRequest req = new JwtRequest("systemowner", "password");
    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_JSON);
    String accessToken = restTemplate.postForObject(tokenUrl, req, String.class);
    headers.set("Authorization", "Bearer " + accessToken.substring(10, accessToken.length() - 2));
    ConferenceDTO conference = new ConferenceDTO(1, "konfaKriva", "grad", "drzava", new Date(2023, 7, 1),
        new Date(2023, 6, 1), "konfa", "neke teme");
    String createKonfa = "http://localhost:8080/";
    HttpEntity<ConferenceDTO> req1 = new HttpEntity<>(conference, headers);
    Assertions.assertThrows(HttpClientErrorException.BadRequest.class, () -> {
        restTemplate.postForEntity(createKonfa, req1, ConferenceDTO.class, 20);
    });
}
```

Slika 5.4: Testiranje kreiranja konferencije

Ispitni slučaj 4: Stvaranje grupe podataka

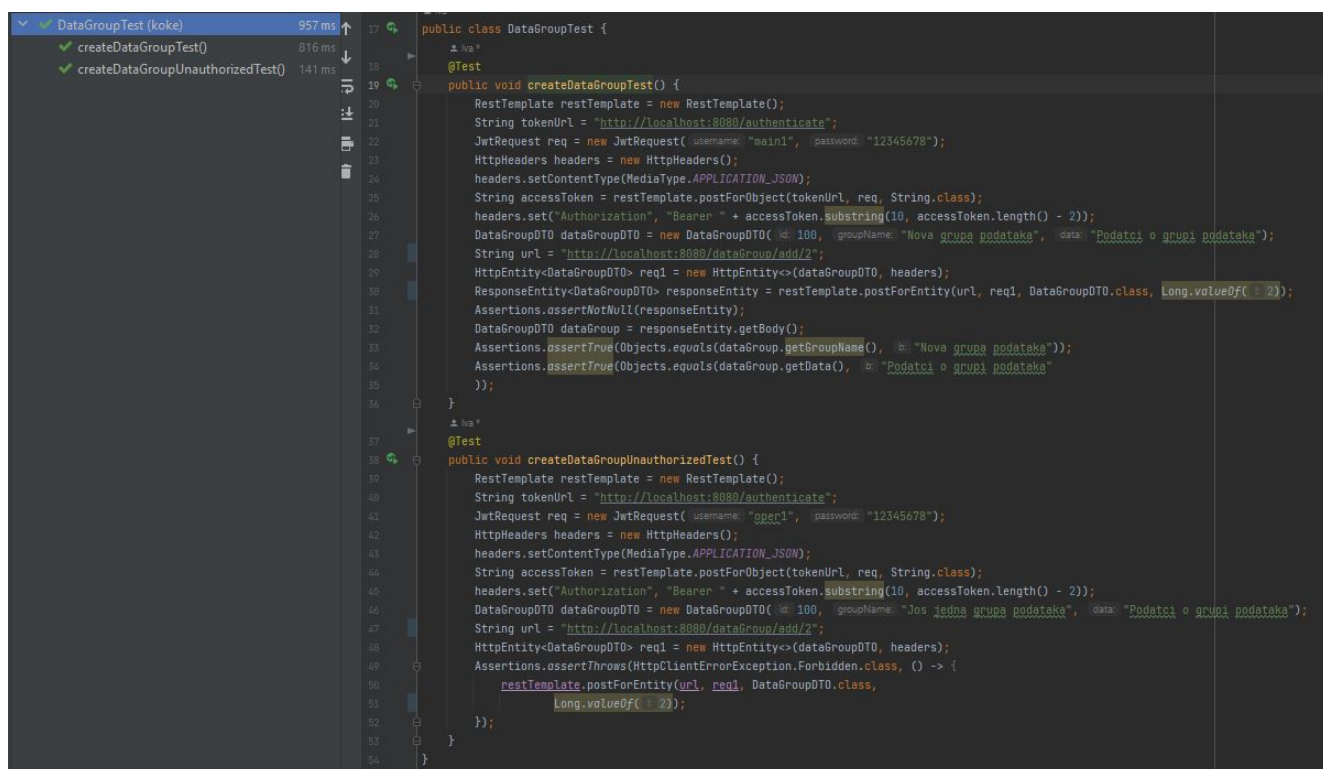
Ulaz:

1. Podatci o grupi podataka koju pokušava stvoriti ovlašteni glavni admin.
2. Podatci o grupi podataka koju pokušava stvoriti neovlašteni operativni admin.

Očekivani rezultat:

1. Kreiranje grupe podataka je uspješno i kao odgovor se vraća stvorena grupa podataka.
2. Kreiranje grupe podataka je neuspješno i kao odgovor se vraća Forbidden.

Rezultat: Svaki očekivani rezultat je ostvaren. **Aplikacija je prošla test.**



Slika 5.5: Testiranje kreiranja grupe podataka

Ispitni slučaj 5: Stvaranje specijalnog događaja za konferenciju**Ulaz:**

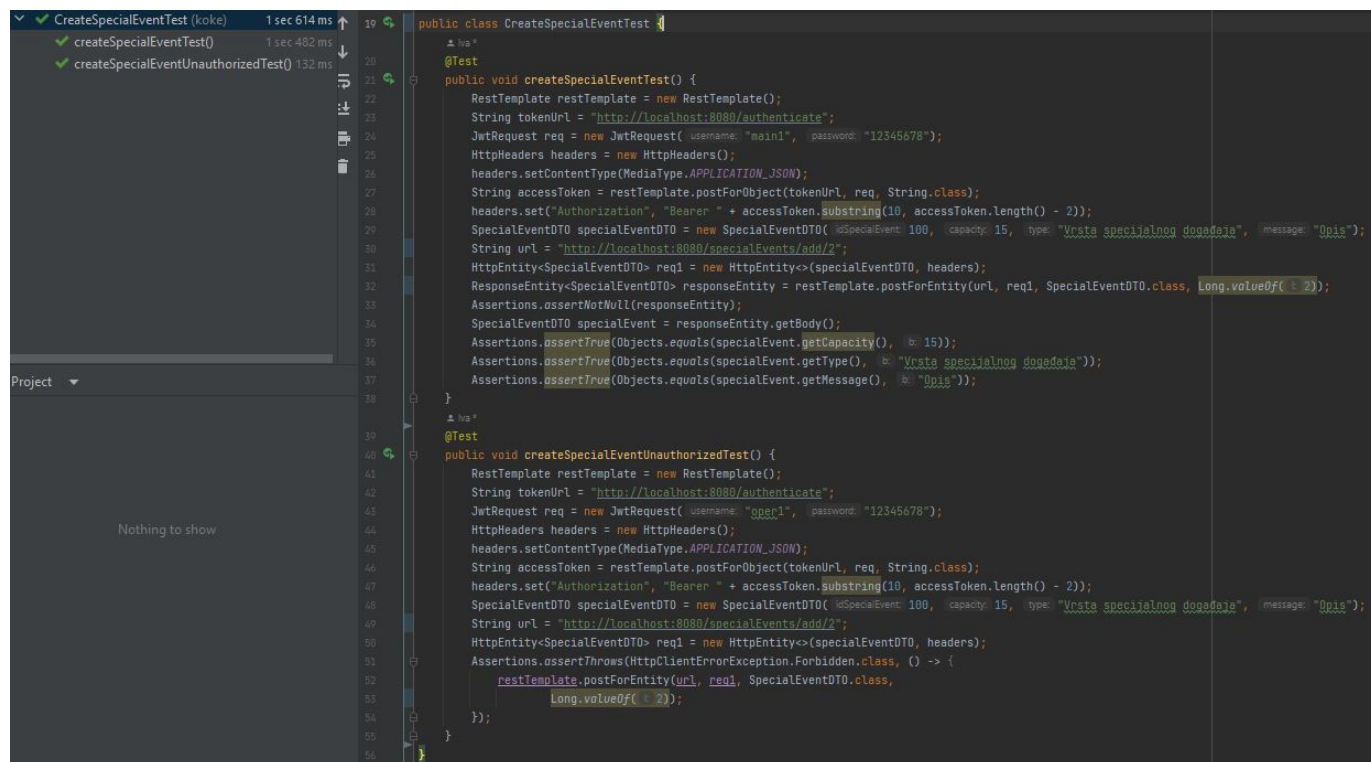
1. Podatci o specijalnom događaju kojeg pokušava stvoriti ovlašteni glavni admin.
2. Podatci o specijalnom događaju kojeg pokušava stvoriti neovlašteni operativni admin.

Očekivani rezultat:

1. Kreiranje specijalnog događaja je uspješno i kao odgovor se vraća stvoreni specijalni događaj.

2. Kreiranje specijalnog događaja je neuspješno i kao odgovor se vraća Forbidden.

Rezultat: Svaki očekivani rezultat je ostvaren. **Aplikacija je prošla test.**



Slika 5.6: Testiranje kreiranja specijalnog događaja

Ispitni slučaj 6: Prijava sudionika na specijalni događaj

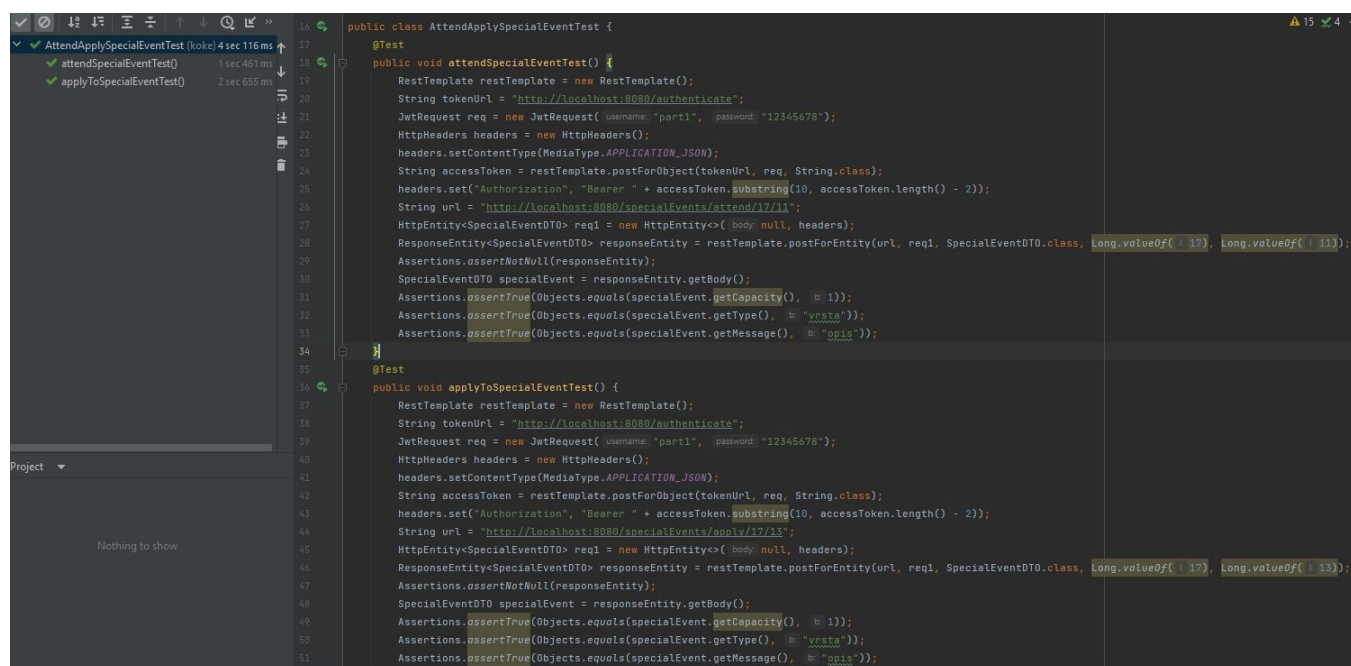
Ulaz:

1. Sudionik se prijavljuje na događaj kojem kapacitet nije popunjen.
2. Sudionik se prijavljuje na događaj kojem kapacitet je popunjen.

Očekivani rezultat:

1. Sudionik se uspješno prijavio na događaj i kao odgovor se vraća taj specijalni događaj.
2. Sudionik se nije prijavio na događaj, već je dodan u listu čekanja za taj događaj i na njegov email mu dolazi obavijest, a ovlaštenom glavnom adminu šalje se mail da lista čekanja nije prazna te da, ako je moguće, poveća kapacitet. Kao odgovor se vraća isti specijalni događaj.

Rezultat: Svaki očekivani rezultat je ostvaren. **Aplikacija je prošla test.**



Slika 5.7: Testiranje prijave na specijalni događaj

Ispitni slučaj 7: Povećanje kapaciteta specijalnog događaja

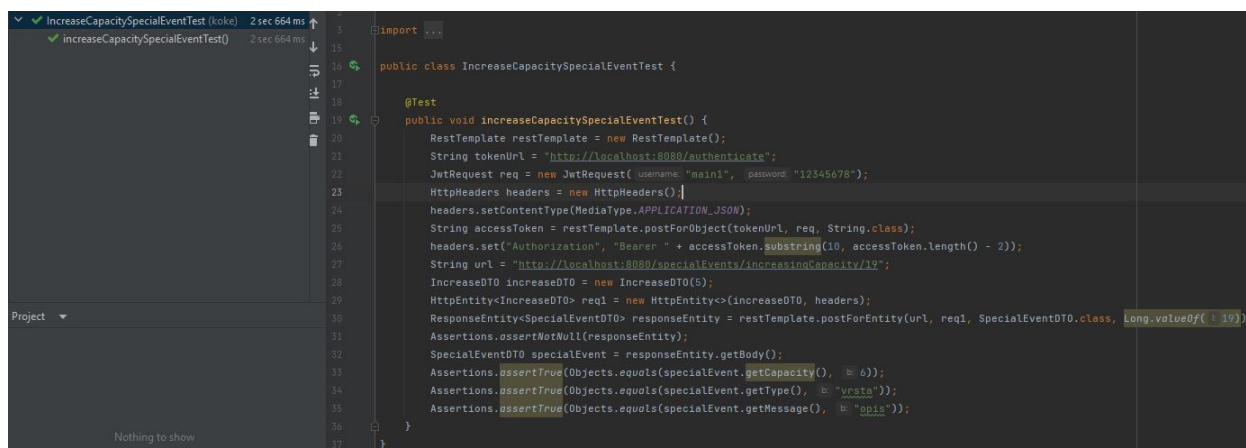
Ulaz:

1. Ovlašteni glavni admin unosi za koliko želi povećati kapacitet specijalnog događaja.

Očekivani rezultat:

1. Kapacitet specijalnog događaja je povećan. Šalje se mail sudionicima na listi čekanja da su sada prijavljeni na specijalni događaj. Kao odgovor se vraća isti specijalni događaj.

Rezultat: Očekivani rezultat je ostvaren. **Aplikacija je prošla test.**



Slika 5.8: Testiranje povećanja kapaciteta specijalnog događaja

5.2.2 Ispitivanje sustava

Ispitni slučaj 1: Uspješan login s postojećim korisnikom

Ulaz:

1. Korisničko ime i lozinka postojećeg korisnika

Očekivani rezultat:

1. Login je uspješan i preusmjerava nas na početni izbornik.

Rezultat: Očekivani rezultat je ostvaren. **Aplikacija je prošla test.**



Slika 5.9: Uspješan login - Selenium

Ispitni slučaj 2: Neuspješan login s nepostojećim korisnikom

Ulaz:

1. Korisničko ime i lozinka nepostojećeg korisnika

Očekivani rezultat:

1. Login je neuspješan i ne preusmjerava nas na početni izbornik.

Rezultat: Očekivani rezultat je ostvaren. **Aplikacija je prošla test.**

```
@Test
void loginTestFailed() {
    System.setProperty("webdriver.chrome.driver", "C:\\Program Files (x86)\\ChromeDriver\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();

    driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

    driver.get("https://kokeferencija.onrender.com/login");
    WebElement username = driver.findElement(By.name("username"));
    username.sendKeys("barcelona");
    WebElement password = driver.findElement(By.name("password"));
    password.sendKeys("jenaiboljklub");
    driver.findElement(By.xpath("//button[normalize-space()='Login']")).click();
    try {
        Thread.sleep(5000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    String redirectUrl = driver.getCurrentUrl();
    boolean result = redirectUrl.contains("menu");
    assertFalse(result);
}
```

Slika 5.10: Neuspješan login - Selenium

Ispitni slučaj 3: Otvaranje stranice MyConference

Ulaz:

- Korisničko ime i lozinka postojećeg korisnika

Očekivani rezultat:

1. Uspješno se dohvaćaju svi resursi za prikaz stranice MyConference i stranica se uspješno prikazuje.

Rezultat: Očekivani rezultat je ostvaren. **Aplikacija je prošla test.**

✓ Otvaranje MyConference	Command	Target	Value
1	✓ open	/	
2	✓ set window size	1106x864	
3	✓ click	css=.ui	
4	✓ click	name=username	
5	✓ type	name=username	mainadmin
6	✓ click	name=password	
7	✓ type	name=password	12345678
8	✓ click	css=.ui:nth-child(3)	
9	✓ click	css=.ui:nth-child(3) p	

Slika 5.11: Otvaranje MyConference test

Running 'Otvaranje MyConference stranice'

1. open on / OK
 2. setWindowSize on 1106x864 OK
 3. click on css=.ui OK
 4. click on name=username OK
 5. type on name=username with value mainadmin OK
 6. click on name=password OK
 7. type on name=password with value 12345678 OK
 8. click on css=.ui:nth-child(3) OK
 9. Trying to find css=.ui:nth-child(3) p... OK
- 'Otvaranje MyConference stranice' completed successfully

Slika 5.12: Otvaranje MyConference rezultat

Ispitni slučaj 4: Otvaranje galerije

Ulaz:

- Korisničko ime i lozinka postojećeg korisnika

Očekivani rezultat:

1. Uspješno se dohvaćaju svi resursi za prikaz galerije i stranica se uspješno prikazuje.

Rezultat: Očekivani rezultat je ostvaren. **Aplikacija je prošla test.**

✓ Otvaranje MyConference		Command	Target	Value
✓ Otvaranje galerije	1	✓ open	/	
	2	✓ set window size	1552x880	
	3	✓ click	css=.ui	
	4	✓ click	name=username	
	5	✓ type	name=username	mainadmin
	6	✓ click	name=password	
	7	✓ type	name=password	12345678
	8	✓ click	css=.ui:nth-child(3)	
	9	✓ click	css=.tablet:nth-child(3) span	
	10	✓ click	css=.button-container-bottom > div:nth-child(2) .ui	
	11	✓ click	css=.icong-container:nth-child(1) img	

Slika 5.13: Otvaranje galerije test

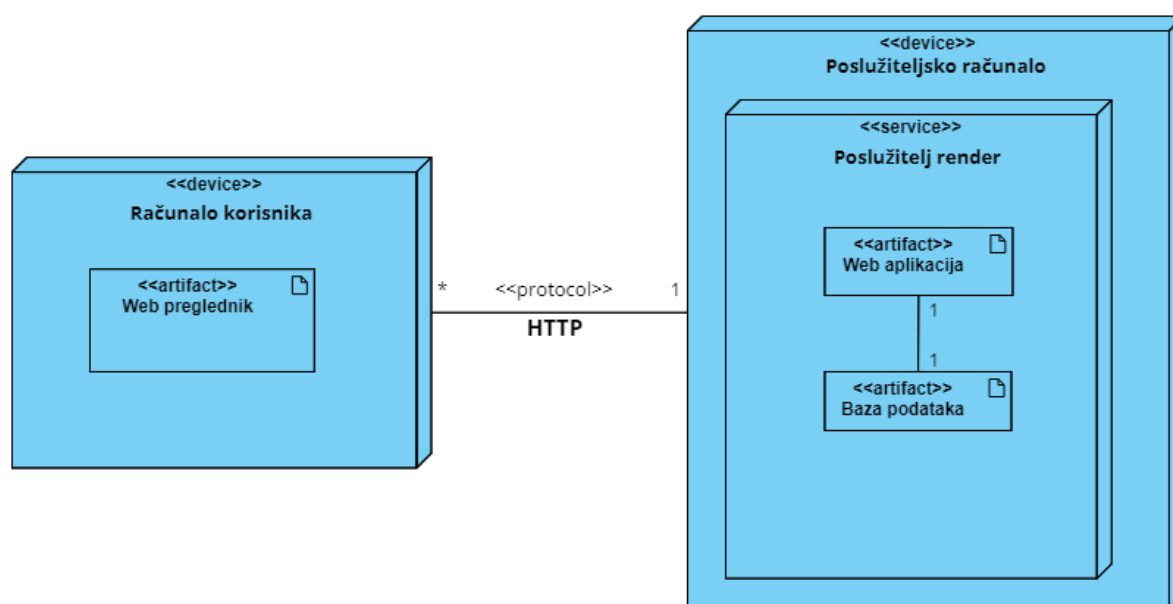
Running 'Otvaranje galerije'
1. open on / OK
2. setWindowSize on 1552x880 OK
3. click on css=.ui OK
4. click on name=username OK
5. type on name=username with value mainadmin OK
6. click on name=password OK
7. type on name=password with value 12345678 OK
8. click on css=.ui:nth-child(3) OK
9. Trying to find css=.tablet:nth-child(3) span... OK
10. click on css=.button-container-bottom > div:nth-child(2) .ui OK
11. Trying to find css=.icong-container:nth-child(1) img... OK
'Otvaranje galerije' completed successfully

Slika 5.14: Otvaranje galerije rezultat

5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Arhitektura sustava aplikacije *Kokeferencija* je "klijent - poslužitelj" gdje se poslužitelj Render, na kojem se nalaze web stranica i baza podataka, nalazi na poslužiteljskom računalu, a klijenti koriste web preglednik kako bi pristupili web aplikaciji. Komunikacija između računala korisnika i poslužitelja odvija se preko HTTP veze.

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 5.15: dijagram razmještaja

5.4 Upute za puštanje u pogon

Postavljanje računa na Render¹⁷

Potrebno je stvoriti korisnički račun na Render-u, te povezati ga s GitLab korisničkim računom kako bi mogli pristup projektu.

Konfiguracija i pokretanje poslužitelja baze podataka

Na nadzornoj ploči Rendera odabрати opciju New, zatim PostgreSQL. U polje *Name* upisujemo ime baze koja se nalazi lokalno na našem računali, zatim u polja *Database* i *Username* opcionalno upisujemo proizvoljno ime baze i korisničko ime. Za regiju odabiremo Frankfurt, te odgovarajuću PostgreSQL verziju. Za kraj odredimo besplatni plan.

Spajanje backend-a na poslužitelja baze podataka

U datoteku *application.properties* upisujemo potrebne varijable. Kao *spring.datasource.password* postavljamo lozinku koja se izgenerirala pri puštanju u pogon. Zatim postavljamo *spring.datasource.username*. Url koji postavljamo ima oblik *jdbc:postgresql://hostname/database*. Za driver postavljamo *org.postgresql.Driver*. Svi potrebi podatci o bazi koja je puštena u pogon nalaze se na Renderu (Dashboard zatim naziv baze koji smo postavili pri pogonu). U *application.properties* također dodamo liniju ***server.servlet.context-path=/api*** koja određuje prefiks svih zahtjeva na backend.

```
spring.datasource.password=${DB_PASS:F4bs6XxH8n6Aacm4tRvKATdQKRTnp8gy}
spring.datasource.username=${DB_USERNAME:systemowner}
spring.datasource.url=${DB_URL:jdbc:postgresql://dpg-cesa0nqrk0dan7jr7og-a:5432/kokedatabase_olpj}
spring.datasource.driverClassName=${DB_DRIVER:org.postgresql.Driver}
server.servlet.context-path=/api
```

Slika 5.16: application.properties

Priprema backend-a

Stvaramo Dockerfile u mapi u kojoj se nalazi *sourcecode*. Konfiguracije datoteke jer prikazana na slici 5.10.

¹⁷<https://dashboard.render.com/>

```
# Container za izgradnju (build) aplikacije
FROM openjdk:17-alpine AS builder

# Kopiranje izvornog koda u container
COPY ../../mvn .mvn
COPY ../../mvnw .
COPY ../../pom.xml .
COPY ../../src src
RUN chmod +x mvnw

# Pokretanje builda
RUN ./mvnw clean package

# Stvaranje containera u kojem ce se vrtiti aplikacija
FROM openjdk:17-alpine

## Ovdje je moguće instalirati alate potrebne za rad aplikacije. Vjerojatno vam neće trebati, no dobro je znati.
## Linux distro koji se koristi je Alpine, stoga se kao package manager koristi apk
#RUN apk install <nesto>

# Kopiranje izvršnog JAR-a iz build containera u izvršni container
COPY --from=builder target/*.jar /app.jar

# Izlaganje porta
EXPOSE 8080

# Naredba kojom se pokrene aplikacija
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

Slika 5.17: Dockerfile

Puštanje backenda u pogon na Renderu

Na nadzornoj ploči Rendera izaberemo *New* zatim *Web Service*. Prikazat će se projekti povezanog GitLab računa. Pored odgovarajućeg projekta izaberemo *Connect*. Postavljamo jedinstveno ime za servis (kokeferencije). Za regiju postavljamo Frankfurt. Zatim postavljamo granu na kojoj se nalazi kod i mapu u kojoj se nalazi izvorni kod. Za varijablu *Environment* odabiremo *Docker*.

Name A unique name for your web service.	<input type="text" value="kokeferencije"/>
Region The region where your web service runs. Services must be in the same region to communicate privately and you currently have services running in Frankfurt .	<input type="text" value="Frankfurt (EU Central)"/>
Branch The repository branch used for your web service.	<input type="text" value="main"/>
Root Directory <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.	<input type="text" value="IzvorniKod/Backend"/>
Environment The runtime environment for your web service.	<input type="text" value="Docker"/>

Slika 5.18: Postavljanje osnovnih varijabli za backend

Na dnu stranice odabiremo gumb *Advanced*, zatim *Add environment variable* te postavimo putanju za Dockerfile. Kada smo sve popunili odabiremo *Create Web Service*.

Priprema frontenda

U package.json datoteku dodajemo ovisnosti potrebne za puštanje u pogon(http-proxy-middleware, dotenv, express)

```
{
  "name": "kokeapp",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@artsy/fresnel": "^6.1.0",
    "@react-pdf/font": "2.2.0",
    "@react-pdf/renderer": "2.1.0",
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^1.1.3",
    "clone": "^2.1.2",
    "cloudinary-react": "^1.8.1",
    "dotenv": "^16.0.3",
    "express": "^4.18.2",
    "file-saver": "^2.0.5",
    "history": "^5.3.0",
    "http-proxy-middleware": "^2.0.6",
    "i18n-iso-countries": "^7.5.0",
    "is-file-download": "^0.4.12"
  }
}
```

Slika 5.19: package.json

U package.json izmijeniti naredbu **"build": "yarn install react-scripts build"**

i dodati naredbu **"start-prod": "node app.js"**.

Puštanje frontenda u pogon na Renderu

Na nadzornoj ploči Rendera izaberemo *New* zatim *Web Service*. Pored odgovarajućeg projekta odaberemo *Connect*. Postavljamo ime za frontend servis koje će biti dio web-adrese (kokeferencija). Za regiju odabiremo Frankfurt, zatim odabiremo grane i mapu u kojoj se nalazi kod. Za varijablu *Environment* biramo *Node*. Za naredbu *Build Command* postavljamo **yarn build**, a za *Start Command* postavljamo **yarn start-prod**. Na dnu stranice odabiremo gumb *Advanced* zatim *Add Environment Variable*. Kao ključ postavljamo **APIBASEURL** a vrijednost postavljamo na adresu backend-a koji je pušten u pogon. Pritisnemo *Create Web Service*.

Pokretanje web-stranice

Aplikaciji se može pristupiti preko adrese <https://kokeferencija.onrender.com/>¹⁸.

¹⁸<https://kokeferencija.onrender.com/>

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije za konferencije. Omogućeno je prećenje dolaska sudionika na konferenciju, sudjelovanja sudionika u događanjima na konferenciji i davanje svih potrebnih informacija sudionicima. Rad na izradi aplikaciji trajao je 16 tjedana. Provedba projekta je bila kroz dvije faze.

Prva faza projekta uključivala je okupljanje tima za razvoj aplikacije, dodjelu projektnog zadatka i dokumentiranje zahtjeva. Kvalitetna provedba prve faze uvelike je olakšala daljnji rad pri realizaciji osmišljenog sustava. Izrađeni obrasci i dijagrami (obraci uporabe, sekvencijski dijagrami, model baze podataka, dijagram razreda) bili su od pomoći podtimovima zaduženima za razvoj *backend*a i *frontend*a.

Dok se u prvoj fazi najvećim dijelom fokusiralo na grupnom pronalaženju optimalnog rješenja, druga faza projekta bila je puno intenzivnija po pitanju samostalnog rada članova. Članovi tima bili su bez iskustva rada na sličnim projektima te su dosta vremena potrošili na upoznavanje i učenje odabranih alata za izradu aplikacije. Također puno vremena potrošeno je na puštanje aplikacije u pogon. Osim realizacije rješenja i *deploy*-a, u drugoj fazi je bilo potrebno dokumentirati ostale UML dijagrame i izraditi popratnu dokumentaciju kako bi budući korisnici mogli lakše koristiti. Dokumentiranje iz prve faze uštedjelo nam je na vremenu tijekom konkretne izrade aplikacije, iako je bilo manjih izmjena i dorada kako se aplikacija razvijala.

Komunikacija između članova tima bila je većinom putem WhatsAppa, a manjim dijelom i Discorda. Moguće proširenje projekta bila bi izrada mobile aplikacije za lakše i efikasnije korištenje stalnim korisnicima.

Rad na ovom projektu bilo je jako vrijedno te ujedno i novo iskustvo članovima tima. Naučili smo puno o radu u timu te smo proširili znanje u smislu poznavanja novih alata. Svjesni smo da aplikacija ima puno prostora za poboljšanje, no iznimno smo zadovoljni postignutim rezultatom projekta upravo zbog neiskustva svih članova tima.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. Stack Overflow, <https://stackoverflow.com/>
3. Geeks for Geeks, <https://www.geeksforgeeks.org/>
4. W3 Schools, <https://www.w3schools.com/>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Javatpoint, <https://www.javatpoint.com/>
7. Astah Community, <http://astah.net/editions/uml-new>
8. Visual Paradigm, <https://www.visual-paradigm.com/>

Indeks slika i dijagrama

3.1	Funkcionalnosti za sudionika i vlasnika sustava	18
3.2	Funkcionalnosti za glavnog admina i operativnog admina	19
3.3	Unos grupe podataka	21
3.4	Unos posebnog događaja	22
3.5	Stvaranje konferencije i main admina konferencije	24
3.6	Prijava na poseban događaj	25
4.1	Arhitektura sustava	27
4.2	Relacijski model baze	35
4.3	Dijagram razreda - DTO	37
4.4	Dijagram razreda - Modeli	38
4.5	Dijagram razreda - Kontroleri	39
4.6	Dijagram razreda - Service	40
4.7	Dijagram stanja - Main Admin	42
4.8	Kreiranje konferencije i glavnog admina	43
4.9	Unos grupa podataka	44
4.10	Kreiranje korisnika	45
4.11	Prijava na specijalni event	46
4.12	Dijagram komponenti	48
5.1	Testiranje prijave	51
5.2	Test ispravne registracije korisnika	52
5.3	Neispravna registracija korisnika	52
5.4	Testiranje kreiranja konferencije	53
5.5	Testiranje kreiranja grupe podataka	54
5.6	Testiranje kreiranja specijalnog događaja	55
5.7	Testiranje prijave na specijalni događaj	56
5.8	Testiranje povećanja kapaciteta specijalnog događaja	57
5.9	Uspješan login - Selenium	57
5.10	Neuspješan login - Selenium	58
5.11	Otvaranje MyConference test	59

5.12	Otvaranje MyConference rezultat	59
5.13	Otvaranje galerije test	60
5.14	Otvaranje galerije rezultat	60
5.15	dijagram razmještaja	61
5.16	application.properties	62
5.17	Dockerfile	63
5.18	Postavljanje osnovnih varijabli za backend	64
5.19	package.json	64
6.1	Commitovi na main grani	76
6.2	Iva Ursić - commits	76
6.3	Dorotea Dragojević - commits	77
6.4	Josipa Markić - commits	77
6.5	Ana Čepić - commits	78
6.6	Nikoleta Benić - commits	78
6.7	Valentina Valić - commits	79
6.8	Andrea Kaselj - commits	79

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 20. listopada 2022.
- Prisustvovali: N.Benić, J.Markić, A.Ćepić, D.Dragojević, A.Kaselj, I.Ursić, V.Valić
- Teme sastanka:
 - sastanak s asistentom i demonstratorom
 - upute i analiza zadatka

2. sastanak

- Datum: 24. listopada 2022.
- Prisustvovali: N.Benić, J.Markić, A.Ćepić, D.Dragojević, A.Kaselj, I.Ursić, V.Valić
- Teme sastanka:
 - dogovor odabira alata i tehnologija
 - pravljenje kanban ploče na gitu
 - izrada osnovnog plana po tjednima

3. sastanak

- Datum: 25. listopada 2022.
- Prisustvovali: N.Benić, J.Markić, A.Ćepić, D.Dragojević, A.Kaselj, I.Ursić, V.Valić
- Teme sastanka:
 - razrada inicijalnog ER modela
 - diskusija funkcionalnosti
 - podjela poslova na relacijski model i obrasce uporabe

4. sastanak

- Datum: 27. listopada 2022.
- Prisustvovali: N.Benić, D.Dragojević, A.Kaselj, I.Ursić
- Teme sastanka:

- sastanak s asistentom i demonstratorom - diskusija dosadašnjeg rada
- konačan odabir alata i tehnologija
- raščišćavanje dilema o funkcionalnostima

5. sastanak

- Datum: 31. listopada 2022.
- Prisustvovali: N.Benić, J.Markić, A.Ćepić, D.Dragojević, A.Kaselj, I.Ursić, V.Valić
- Teme sastanka:
 - podjela poslova za izradu baze i backenda
 - dogovor za pisanje dokumentacije

6. sastanak

- Datum: 2. studenog 2022.
- Prisustvovali: N.Benić, A.Ćepić, A.Kaselj, I.Ursić
- Teme sastanka:
 - pregled backenda
 - popis klasa i metoda

7. sastanak

- Datum: 3. studenog 2022.
- Prisustvovali: N.Benić, I.Ursić
- Teme sastanka:
 - sastanak s asistentom i demonstratorom- diskusija dosadašnjeg rada
 - razrada vrsta korisnika
 - ispravke u bazi

8. sastanak

- Datum: 8. studenog 2022.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić, A.Kaselj
- Teme sastanka:
 - podjela ostatka posla
 - dogovor oko daljnje implementacije

9. sastanak

- Datum: 15. studenog 2022.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić, A.Kaselj
- Teme sastanka:
 - čišćenje gita, rad na deploymentu aplikacije
 - završavanje dokumentacije

10. sastanak

- Datum: 6. prosinca 2022.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić, A.Kaselj
- Teme sastanka:
 - Dokumentacija
 - određivanje novih zadataka

11. sastanak

- Datum: 7. prosinca 2022.
- Prisustvovali: I.Ursić, D.Dragojević, J.Markić, A.Kaselj
- Teme sastanka:
 - određivanje novih zadataka za backend

12. sastanak

- Datum: 8. prosinca 2022.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić, A.Kaselj
- Teme sastanka:
 - određivanje novih zadataka za frontend i backend

13. sastanak

- Datum: 9. prosinca 2022.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić, A.Kaselj
- Teme sastanka:
 - određivanje novih zadataka za frontend i backend

14. sastanak

- Datum: 2. siječnja 2023.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić, A.Kaselj
- Teme sastanka:
 - dogovor o pisanju dokumentacije

15. sastanak

- Datum: 11. siječnja 2023.
- Prisustvovali: N.Benić, I.Ursić, A.Ćepić, D.Dragojević, J.Markić, V.Valić
- Teme sastanka:
 - ispravak i provjera dokumentacije

Tablica aktivnosti

	Nikoleta Benić	Josipa Markić	Iva Ursić	Valentina Valić	Ana Čepić	Andrea Kaselj	Dorotea Dragojević
Upravljanje projektom	6	6	6	6	10	8	10
Opis projektnog zadatka	0	0	0	4	0	0	0
Funkcionalni zahtjevi	0	0	0	0	3	0	0
Opis pojedinih obrazaca	4	0	4	0	2	0	0
Dijagram obrazaca	0	0	1.5	0	2	3	0
Sekvencijski dijagrami	0	1.5	0	0	4	0	0
Opis ostalih zahtjeva	0	0	0.5	0	0	1	0
Arhitektura i dizajn sustava	0	0	0	0	0	4	0
Baza podataka	0	5.5	0	0	0	0	0.5
Dijagram razreda	5	0	0	0	0	0	0
Dijagram stanja	1.5	0	0	0	0	0	0
Dijagram aktivnosti	0	0	0	0	0	0	3
Dijagram komponenti	0	0	0	0	1	0	0
Korištene tehnologije i alati	0	0	0	1	0	0	0
Ispitivanje programskog rješenja	1	0	1	0	0	0	0.75
Dijagram razmještaja	0	1.5	0	0	0	0	0
Upute za puštanje u pogon	1	0	0	0	0	0	0
Dnevnik sastajanja	0	0	0	0	0	1	2
Zaključak i budući rad	0	0	1	0	0	0	0

Nastavljeno na idućoj stranici

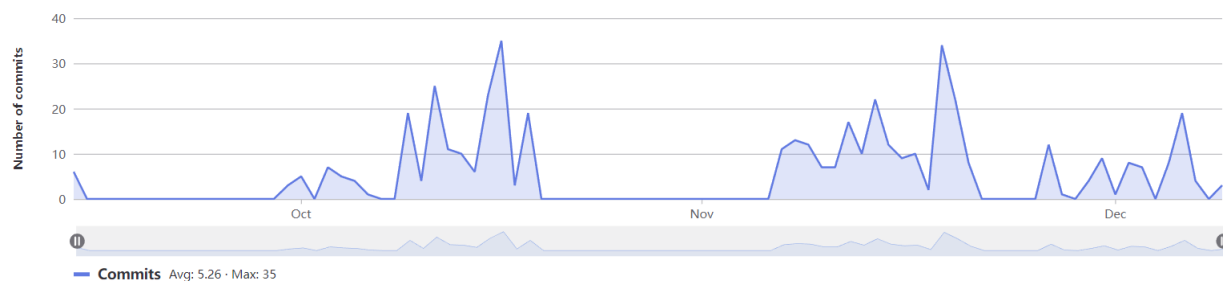
Nastavljeno od prethodne stranice

	Nikoleta Benić	Josipa Markić	Iva Ursić	Valentina Valić	Ana Čepić	Andrea Kaselj	Dorotea Dragojević
Popis literature	0	0	0	0.2	0	0	0
Frontend	40	20	0	38	55	0	0
Izrada baze podataka	3	8	3	8	3	3	13
Spajanje s bazom podataka	2	0	2	4	0	0	2
Backend	15	45	55	20	25	0	51.5
Puštanje aplikacije u pogon	2	20	22	4	11	5	14
Testiranje	0	0	3	0	3	0	5

Dijagrami pregleda promjena

Commits to main

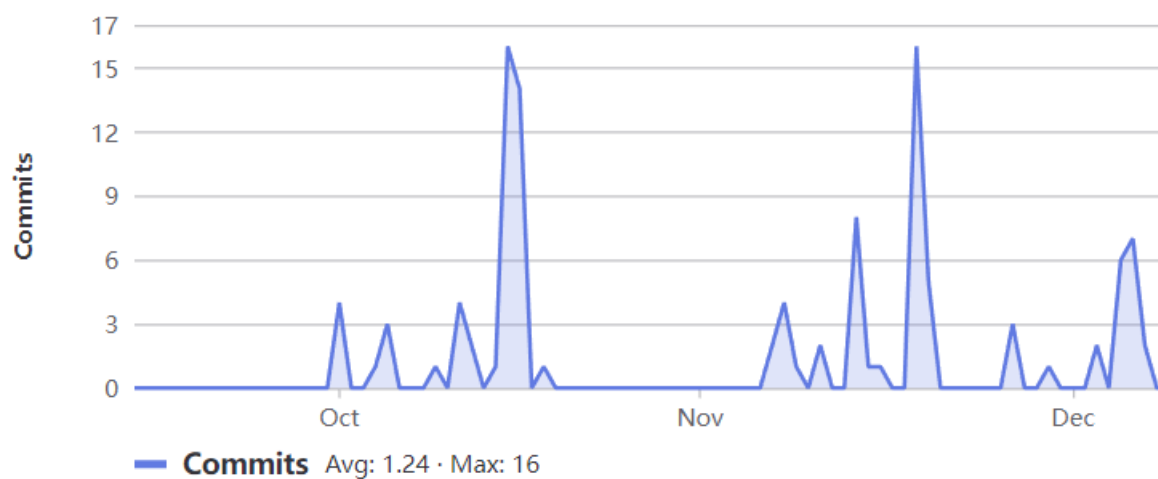
Excluding merge commits. Limited to 6,000 commits.



Slika 6.1: Commitovi na main grani

Iva

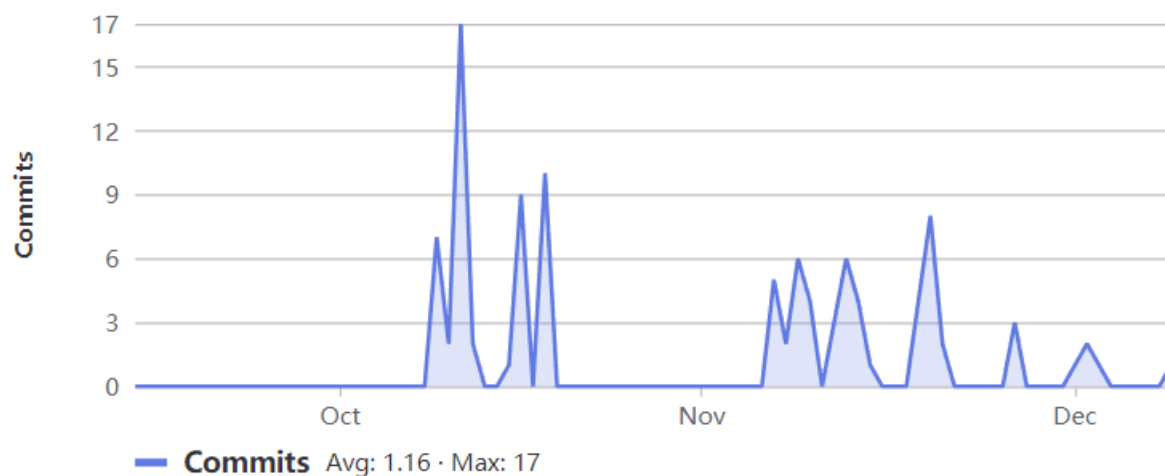
108 commits (iva.ursic@fer.hr)



Slika 6.2: Iva Ursić - commits

Dorotea Dragojevic

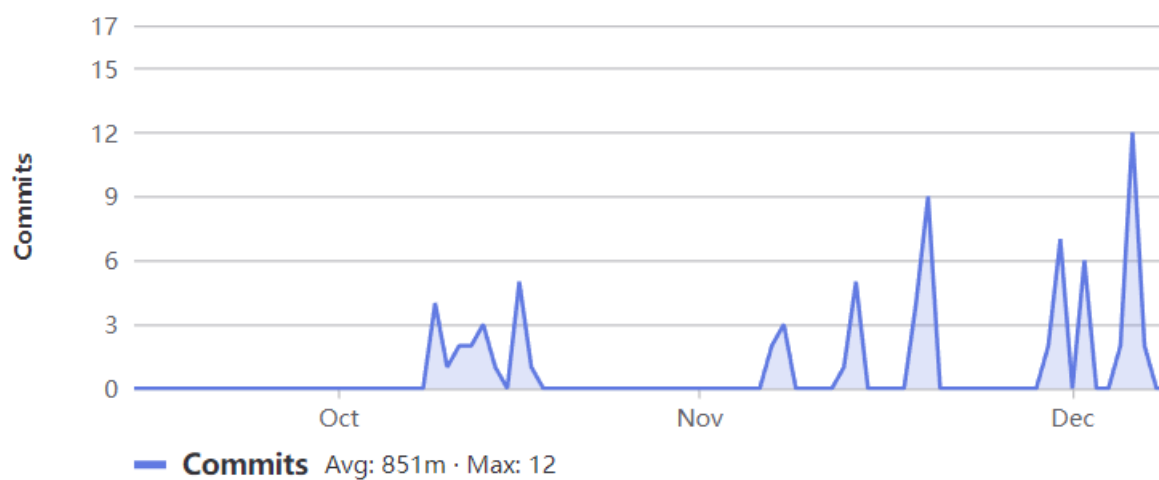
101 commits (dorotea.dragojevic@fer.hr)



Slika 6.3: Dorotea Dragojević - commits

josipamarkic

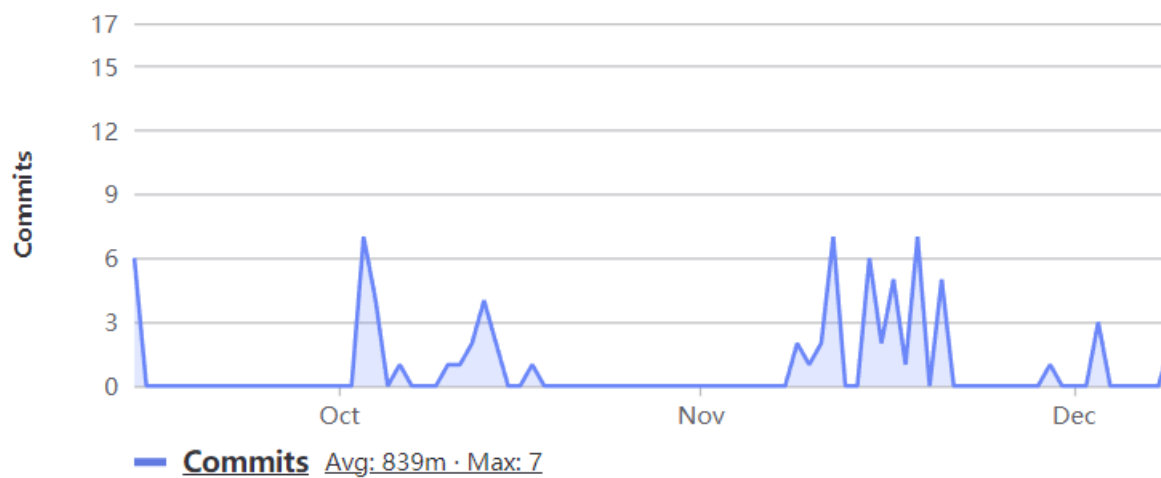
74 commits (jm533@fer.hr)



Slika 6.4: Josipa Markić - commits

AnaČepić

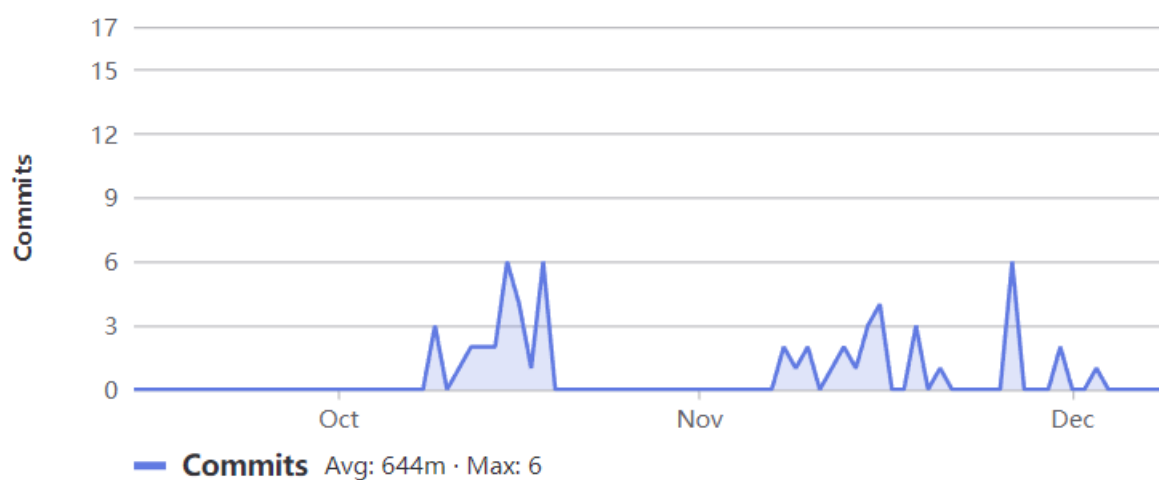
73 commits (ac52963@fer.hr)



Slika 6.5: Ana Čepić - commits

Nikoleta Benić

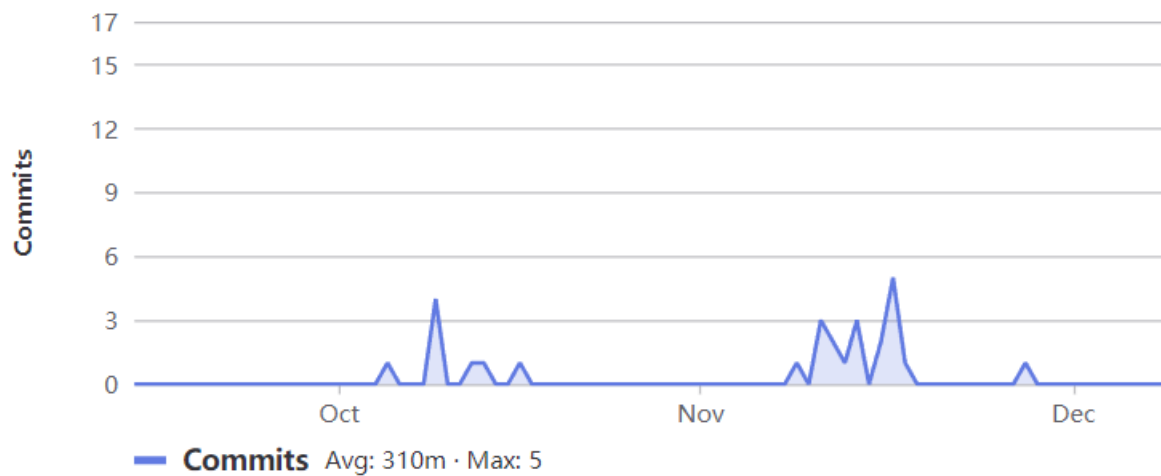
56 commits (nb53112@fer.hr)



Slika 6.6: Nikoleta Benić - commits

Valentina

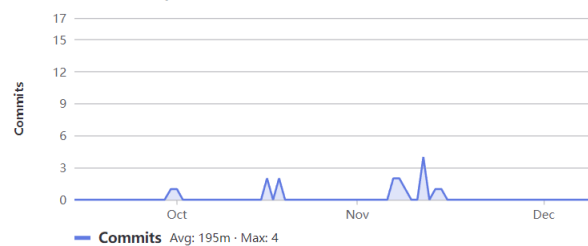
27 commits (valentina.valic13@example.com)



Slika 6.7: Valentina Valić - commits

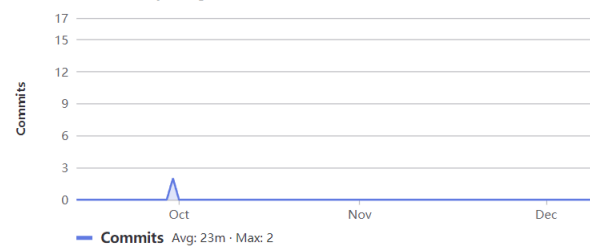
Andrea-Kaselj

17 commits (andrea.kaselj@fer.hr)



Andrea-Kaselj

2 commits (andreaskaselj844@gmail.com)



Slika 6.8: Andrea Kaselj - commits