

Programsko inženjerstvo  
Ak. god. 2020./2021.

Kamp Mlade Nade  
Dokumentacija, Rev. 2

Grupa: *Cozy*  
Voditelj: *Petar Cvitanović*

Datum predaje: *14.1.2020.*

Nastavnik: *Eugen Vušak*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	10
3.1.2 Sekvencijski dijagrami . . . . .	21
3.2 Ostali zahtjevi . . . . .	25
<b>4 Arhitektura i dizajn sustava</b>	<b>26</b>
4.1 Baza podataka . . . . .	28
4.1.1 Opis tablica . . . . .	28
4.1.2 Dijagram baze podataka . . . . .	33
4.2 Dijagram razreda . . . . .	34
4.3 Dijagram stanja . . . . .	37
4.4 Dijagram aktivnosti . . . . .	39
4.5 Dijagram komponenti . . . . .	44
<b>5 Implementacija i korisničko sučelje</b>	<b>45</b>
5.1 Korištene tehnologije i alati . . . . .	45
5.2 Ispitivanje programskog rješenja . . . . .	46
5.2.1 Ispitivanje komponenti . . . . .	46
5.2.2 Ispitivanje sustava . . . . .	51
5.3 Dijagram razmještaja . . . . .	56
5.4 Upute za puštanje u pogon . . . . .	57
<b>6 Zaključak i budući rad</b>	<b>59</b>
<b>Popis literature</b>	<b>61</b>
<b>Indeks slika i dijagonama</b>	<b>62</b>

**Dodatak: Prikaz aktivnosti grupe**

**63**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Opis zadatka i Funkcionalni zahtjevi iz Specifikacije programske potpore	Fabijan Kozina	25.10.2020.
0.2	Obrasci uporabe i dijagrami obrazaca uporabe	Fabijan Kozina	26.10.2020.
0.3	Sekvencijski dijagrami	Antonio Vencl	1.11.2020.
0.4	Ostali zahtjevi	Leo Li	3.11.2020.
0.5	Arhitektura i dizajn sustava - uvod	Petar Jukić	3.11.2020.
0.6	Baza podataka - uvod, opis tablica	Fabijan Kozina	5.11.2020.
0.7	ER i RS dijagrami baze podataka	Fabijan Kozina	7.11.2020.
0.8	Dijagrami razreda	Luka Škarica, Petar Jukić	8.11.2020.
1.0	Korigiranje teksta i provjera dokumentacije	Fabijan Kozina	.
1.2	Dijagram stanja, aktivnosti i komponenti	Petar Cvitanović, Fabijan Kozina	.
1.4	Selenium testiranje	Petar Jukić	. 3.1.2021
1.5	Unit testiranje	Luka Škarica, Fabijan Kozina	. 4.1.2021

Rev.	Opis promjene/dodatak	Autori	Datum
1.7	Upute za puštanje u pogon	Petar Jukić, Leo Li	. 8.1.2021
1.8	Ažuriranje starih dijagrama	Anotnio Vencl	10.1.2021
1.9	Dodatak	Leo Li, Petar Jukić, Petar Cvitanović, Antonio Vencl	. 12.1.2021
2.0	Završne promjene, korigiranje teksta i provjera dokumentacije	Fabijan Kozina	. 14.1.2021

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije u sklopu „Kampa mlade nade“. Kamp nudi svojim korisnicima mogućnost pregledavanja dnevnih i tjednih aktivnosti putem rasporeda koji je personaliziran za svaku pojedinu vrstu korisnika. Na taj način će kamp omogućiti korisnicima da bez brige obavljaju svakodnevne grupne aktivnosti koje su predviđene odgovarajućim animatorima, koji također imaju personalizirane rasporede s dnevnim obavezama. Prilikom pokretanja aplikacije prikazuje se stranica s osnovnim informacijama o kampu, na kojoj je također moguće kreirati račun s ulogama sudionika ili animatora. Jednom kada korisnik kreira račun, umjesto početne stranice pojavljuje se sat koji odbrojava vrijeme preostalo do početka kampa. Početkom kampa svaki od korisnika može vidjeti svoj raspored aktivnosti te pripadnost grupi, u koju su dodijeljeni od strane organizatora kampa. Tijekom kampa i na kraju, moguće je ostaviti ocjenu aktivnosti te cijelokupan dojam o kampu.

Kao što smo rekli, postoje 2 tipa korisnika. To su animatori i sudionici. Za stvaranje računa sudionike, potrebno je na početnoj stranici odabrati gumb „Sign up“ te predati sljedeće podatke:

- ime
- prezime
- broj mobitela
- email adresa
- datum i godina rođenja
- motivacijsko pismo
- broj mobitela odgovarajuće osobe (u slučaju da je sudionik mlađi od 18 godina)

Isto vrijedi i za animatore. Jednom kada je prijava zaprimljena, korisnik na mail dobiva mail uz pomoć kojega verificira račun i email adresu. Klikom na link koji se nalazi u mail-u, korisnik dolazi na novu stranicu web aplikacije na kojoj upisuje lozinku za zaštitu računa. Odabirom željene lozinke i klikom na gumb „Save“ završen je proces registracije i korisniku je stvoren račun unutar web aplikacije.

Registrirani korisnik (sudionik ili animator) može pregledati, mijenjati osobne podatke i izbrisati svoj korisnički račun. Osim toga, može u svakom trenutku za vrijeme trajanja kampa otvoriti i pregledati svoj personalizirani raspored aktivnosti.

Sudionik prije početka kampa vidi samo sat koji odbrojava vrijeme do početka kampa te kontakt organizatora, a tek nakon početka kampa mogu vidjeti raspored aktivnosti koji je ujedno i glavna značajka ove web aplikacije. Uz to mogu vidjeti popis aktivnosti na kojima su do sada sudjelovali, te iste aktivnosti ocijeniti ocjenama od 1 do 10. Osim rasporeda aktivnosti, mogu vidjeti informacije o grupi u koju su dodijeljeni te podatke o animatorima koji su zaposleni na kampu. Tijekom trajanja kampa moguće je također dobiti uvid o prošlim aktivnostima na kojima je sudionik prisustvovao, te ih ocijeniti ocjenama od 1 do 10. Po završetku kampa dodatno je omogućeno davanje ocjene cjelokupnog dojma.

Uz sudionika postoje još 2 vrste korisnika, a to su:

- organizator
- animator

Animator kampa kroz aplikaciju ima pristup popisu svih grupa koje je stvorio organizator, kao i popis svih članova svake grupe zajedno s njihovim kontakt podatcima. Isto tako, omogućene su iste akcije povezane s uvidom u prošle aktivnosti i ocjenjivanje istih, kao i kod sudionika.

Organizator sustava ima najveće ovlasti. Za početak organizator kampa određuje početak održavanja događaja. Za vrijeme registracije sudionika i animatora, organizator zaprima sve pokušaje prijave, te ako procijeni da je prijava valjana, prima korisnika u sustav te u bazi podataka stvara račun s podatcima predanim u obrascu zajedno sa šifrom koju je korisnik specificirao. Na početku kampa dužnost orga-

nizatora jest na temelju broja prijavljenih korisnika odrediti (subjektivno) optimalan broj grupa u koje će se podijeliti sudionici. Jednom kada organizator to učini, izvršava se nasumična raspodjela sudionika po grupama, ali su naravno moguće naknadne izmjene i premještanje sudionika iz grupe u grupu. Nakon formiranja grupa, slijedi popunjavanje rasporeda s aktivnostima koje također provodi organizator kampa. Taj proces se sastoji od određivanja vremenskog početka i kraja pojedine aktivnosti te pridruživanja odgovarajućeg animatora za aktivnost. Pri tome je bitno paziti da ne dođe do nekakvih preklapanja između grupa i animatora, o čemu će nas sustav upozoriti ili neće dopustiti da do takve situacije uopće i dođe. Prilikom završetka dodavanja aktivnosti treba dodatno provjeriti je jesu li sve grupe sudjelovale na svakoj aktivnosti točno jednom. Svaki dan postoje tri aktivnosti koje su fiksne, a to su doručak u 8 sati, ručak u 12 sati i večera u 18 sati. Na tim aktivnostima sudjeluju sve grupe i svi animatori, te traju 1 sat. Tijekom završetka kampa organizatori mogu vidjeti popis svih povratnih ocjena po aktivnostima te ih moraju moći pretraživati prema sljedećim atributima: korisnik, grupa i/ili aktivnost. Po završetku kampa dodatne je moguće pregledati i ocjene cjelokupnog dojma svih korisnika.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

Dionici:

1. Organizator (naručitelj)
2. Sudionici kampa
3. Animatori u kampu
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:

- (a) pregledati osnovne informacije o kampu
  - i. vrijeme održavanja
  - ii. mjesto održavanja
  - iii. kratak opis
  - iv. kontakt organizatora
  - v. popis aktivnosti
- (b) prijaviti se na kamp kao sudionik ili kao animator

2. Sudionik može:

- (a) pregledavati i mijenjati osobne podatke
- (b) izbrisati svoj korisnički račun
- (c) vidjeti svoj raspored aktivnosti
- (d) vidjeti informacije o grupi u koju je dodijeljen
- (e) vidjeti kontakt podatke o animatoru koji je zadužen za njegovu grupu
- (f) imati uvid u popis odrađenih aktivnosti

- (g) ocijeniti sve aktivnosti koje je odradio ocjenama od 1 do 10
- (h) dati ocjenu cjelokupnog dojma po završetku kampa

3. Animator može:

- (a) vidjeti raspored aktivnosti koje je dužan predvoditi
- (b) vidjeti popis svih grupa u kampu
- (c) vidjeti popis i informacije o svim sudionicima u kampu
- (d) vidjeti popis svih aktivnosti koje je odradio
- (e) ocijeniti sve aktivnosti koje je odradio ocjenama od 1 do 10

4. Organizator može:

- (a) odrediti vrijeme početka kampa
- (b) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (c) odbiti ili prihvati zaprimljene prijave od strane korisnika
- (d) odrediti broj grupa u koje će se rasporediti sudionici
- (e) mijenjati pripadnost grupi proizvoljnih sudionika
- (f) popuniti raspored aktivnostima u kojima sudjeluju grupe i animatori  
(pri tome paziti na pravila o preklapanju aktivnosti)
- (g) vidjeti popi ocjena za svaku aktivnost na kojoj su sudjelovali sudionici i animatori

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima
- (b) pohranjuje sve podatke i razinama ovlasti korisnika
- (c) pohranjuje sve podatke vezane uz sadržaje kampa (raspored, aktivnosti, grupe)

### 3.1.1 Obrasci uporabe

#### UC1 -Početna stranica

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Pregledati informacije o kampu te se po mogućnosti prijaviti
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik na stranici vidi osnovne informacije o kampu
  2. U slučaju zainteresiranosti, korisnik ima mogućnost kliknuti gumb za stvaranje računa kao sudionik ili animator

#### UC2 -Registracija

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest u obliku emaila o uspješnoj registraciji
  4. Klikom na link u mailu korisnik odabire šifru za svoj račun u sustavu
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u
    1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
    2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

### UC3 -Prijava u sustav (prije početka kampa)

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Dobiti informaciju o početku kampa (odbrojavanje u određenom vremenskom formatu)
- **Sudionici:** -
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena i lozinke
  2. Potvrda o ispravnosti unesenih podataka
  3. Pristup satu za odbrojavanje
- **Opis mogućih odstupanja:**
  - 2.a Neispravno korisničko ime ili lozinka
    1. Sustav obaveštava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

### UC4 -Odabir lozinke

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Odabrati lozinku koja će korisniku služiti za prijavu u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Unos željene lozinke
  2. Potvrda o uspješnom stvaranju korisničkog računa u bazi podataka
- **Opis mogućih odstupanja:**
  - 2.a Ne unošenje lozinke u predviđeni prostor te izlazak iz web aplikacije
    1. Sustav odbacuje prijavu

### UC5 -Prijava u sustav (nakon početka kampa) - sudionici

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Pregled korisničkog sučelja koje se sastoji od rasporeda aktivnosti, informacijama o dodijeljenoj grupi i animatoru te popisu odrađenih aktivnosti
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**

1. Unos korisničkog imena i lozinke
  2. Potvrda o ispravnosti unesenih podataka
  3. Pristup rasporedu aktivnosti te informacijama o dodijeljenoj grupi i animatoru
- **Opis mogućih odstupanja:**
    - 2.a Neispravno korisničko ime ili lozinka
      1. Sustav obaveštava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

#### **UC6 -Prijava u sustav (nakon početka kampa) - animatori**

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Pregled korisničkog sučelja koje se sastoji od rasporeda aktivnosti, informacija o svim grupama i sudionicima kampa te popisu odrđenih aktivnosti
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena i lozinke
  2. Potvrda o ispravnosti unesenih podataka
  3. Pristup informacijama o svim grupama i sudionicima kampa
- **Opis mogućih odstupanja:**
  - 2.a Neispravno korisničko ime ili lozinka
    1. Sustav obaveštava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

#### **UC7 -Pregled osobnih podataka**

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

### UC8 -Promjena osobnih podataka

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabere opciju za promjenu podataka
  2. Korisnik mijenja svoje osobne podatke
  3. Korisnik sprema promjene
  4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 2.a Korisnik promijeni svoje osobne podatke, ali ne odabere opciju "Spremi promjenu"
    1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

### UC9 -Brisanje korisničkog računa

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

### UC10 -Popis održanih aktivnosti

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Vidjeti popis održanih aktivnosti te ocijeniti ih ocjenom od 1 do 10
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**

1. Klijent vidi popis svih odrđenih aktivnosti
2. Klijent odabire aktivnost koju želi ocijeniti
3. Klijent ocjenjuje aktivnost željenom ocjenom
4. Klijent odabire opciju "Ocijeni", kojom završava proces ocjenjivanja

### **UC11 -Završetak kampa – sudionici i animatori**

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Ocjena cjelokupnog dojma kampa
- **Sudionici:** Klijent je prijavljen
- **Preduvjet:** <preduvjet>
- **Opis osnovnog tijeka:**
  1. Klijent ocjenjuje cjelokupan doživljaj na kampu

### **UC12 -Sučelje za organizatore**

- **Glavni sudionik:** Organizator
- **Cilj:** Uvid u sve operacije koje obavlja organizator za vrijeme i i nakon trajanja kampa.
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen i kamp je započeo
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena i lozinke
  2. Potvrda o ispravnosti unesenih podataka
  3. Pristup informacijama o svim grupama i sudionicima kampa
  4. Organizator se prijavljuje u sustav
  5. Prikaz gumbova i poveznica za sljedeće akcije:
    - (a) Definiranje aktivnosti
    - (b) Uvid u zaprimljene prijave korisnika i sudionika
    - (c) Određivanje broja grupa
    - (d) Stvaranje rasporeda za korisnike i animatore
  6. Organizator odabire gumb ili akciju koju želi izvršiti, te odlazi na novu stranicu

### UC13 -Definiranje aktivnosti

- **Glavni sudionik:** Organizator
- **Cilj:** Stvoriti novu aktivnost.
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen.
- **Opis osnovnog tijeka:**

Organizator vidi obrazac za stvaranje definiranje nove aktivnosti koji se sastoji od sljedećih polja:

- (a) Ime aktivnosti
  - (b) Kratki opis
  - (c) Vremensko trajanje aktivnosti
  - (d) Tip aktivnosti (4 tipa s obzirom na broj grupa koje sudjeluju u kampu)
- **Opis mogućih odstupanja:**

2.a

- Organizator ne ispuni barem jedno od potrebnih polja za unos
- 1. Vraćanje na sučelje za organizatore

### UC14 -Prihvatanje i odbijanje prijava

- **Glavni sudionik:** Organizator
- **Cilj:** Prihvati ili odbiti prijave sudionika i animatora
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen i barem jedan korisnik je napravio prijavu
- **Opis osnovnog tijeka:**
  1. Organizator odabire opciju uvida u prijave
  2. Prikazuje se lista svih prijava koje sadržavaju osobne podatke o korisnicima
  3. Prikaz dvaju gumbova, od kojih jedan služi za prihvatanje prijave i stvaranje korisničkog računa, dok drugi služi za odbijanje prijave

## UC15 -Određivanje broja grupa

- **Glavni sudionik:** Organizator
- **Cilj:** Odrediti broj grupa u koje će sudionici biti raspodijeljeni.
- **Sudionici:** Baza podataka
- **Preduvjet:**
  1. Organizator je prijavljen
  2. Proces prijava korisnika za kamp je završio
- **Opis osnovnog tijeka:**
  1. Organizator vidi popis svih uspješno prijavljenih sudionika
  2. U polju za unos podatka o broju grupa unosi željeni broj grupa u brojčanom obliku
    - (a) 3. Izvršava se automatska dodjela sudionika u grupe
- **Opis mogućih odstupanja:**
  - 2.a Organizator u polje za unos napiše nešto što nije broj ili negativni broj
    1. Sustav javlja poruku o pogrešnom unosu te organizatora vraća na stranicu za određivanje grupa (refresh)
  - 2.b Organizator u polje za unos, unosi broj veći od broja svih sudionika
    1. Sustav stvara broj grupa jednak broju sudionika, tako da svaki korisnik čini jednu grupu, a ostali broj grupa zanemaruje.

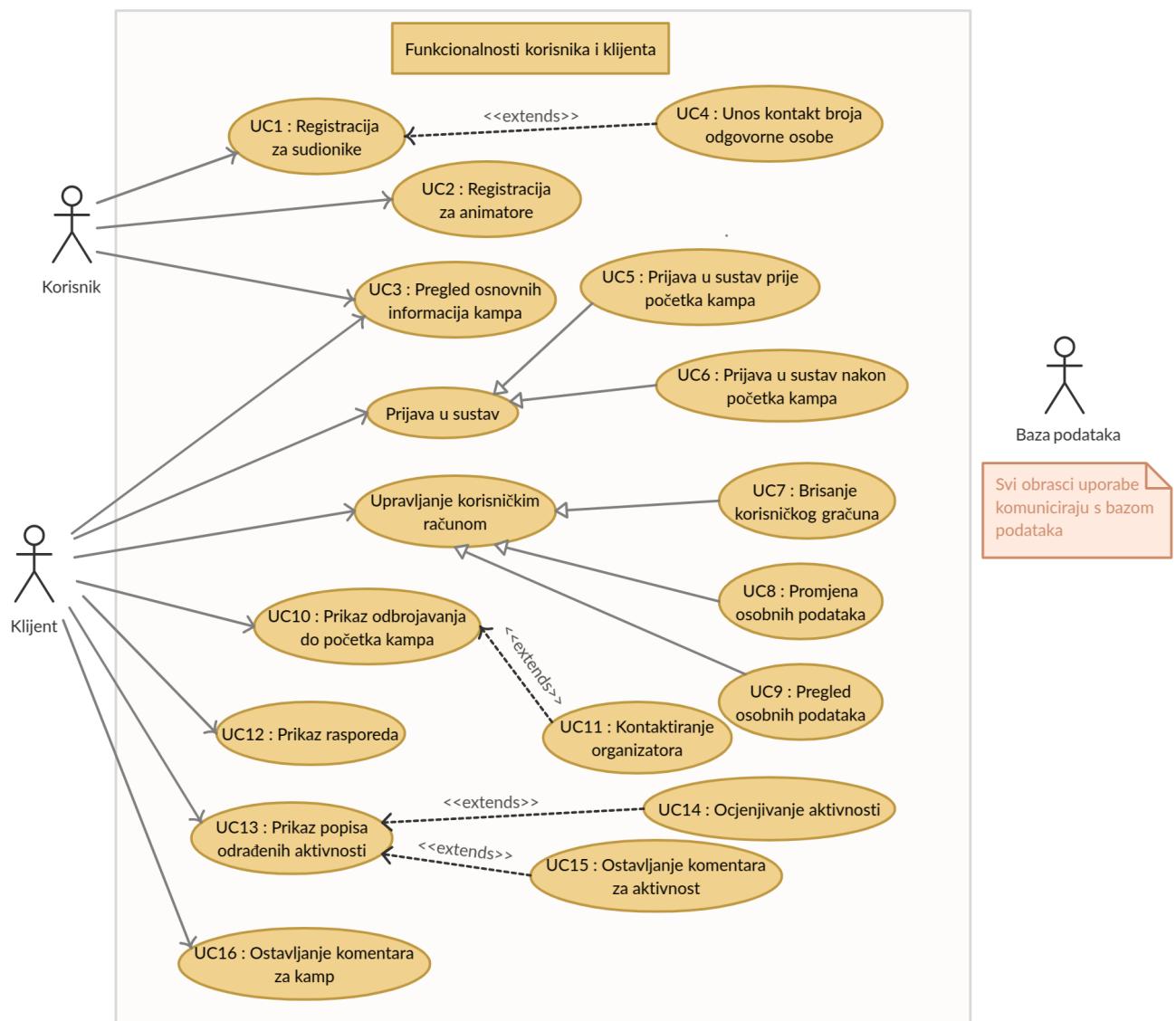
## UC16 -Kreiranje rasporeda

- **Glavni sudionik:** Organizator
- **Cilj:** Stvoriti raspored koji će koristiti sudionici i animatori
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen i svi korisnici su prijavljeni u sustav
- **Opis osnovnog tijeka:**
  1. Organizator odabire opciju stvaranja rasporeda
  2. Prikaz tjednog rasporeda po danima
  3. Organizator za svaku aktivnost određuje sljedeća svojstva:
    - (a) Vrijeme održavanja
    - (b) Grupe koje prisustvuju aktivnosti
    - (c) Animator/e koji predvodi/e aktivnost
  4. Onemogućeno je dodavanje aktivnosti koje ne poštuju pravila o preklapanju u rasporedu

- **Opis mogućih odstupanja:**

- 2.a Organizator u polje za unos napiše nešto što nije broj ili negativni broj
  1. Sustav javlja poruku o pogrešnom unosu te organizatora vraća na stranicu za određivanje grupa (refresh)
- 2.b Organizator u polje za unos, unosi broj veći od broja svih sudionika
  1. Sustav stvara broj grupa jednak broju sudionika, tako da svaki korisnik čini jednu grupu, a ostali broj grupa zanemaruje.

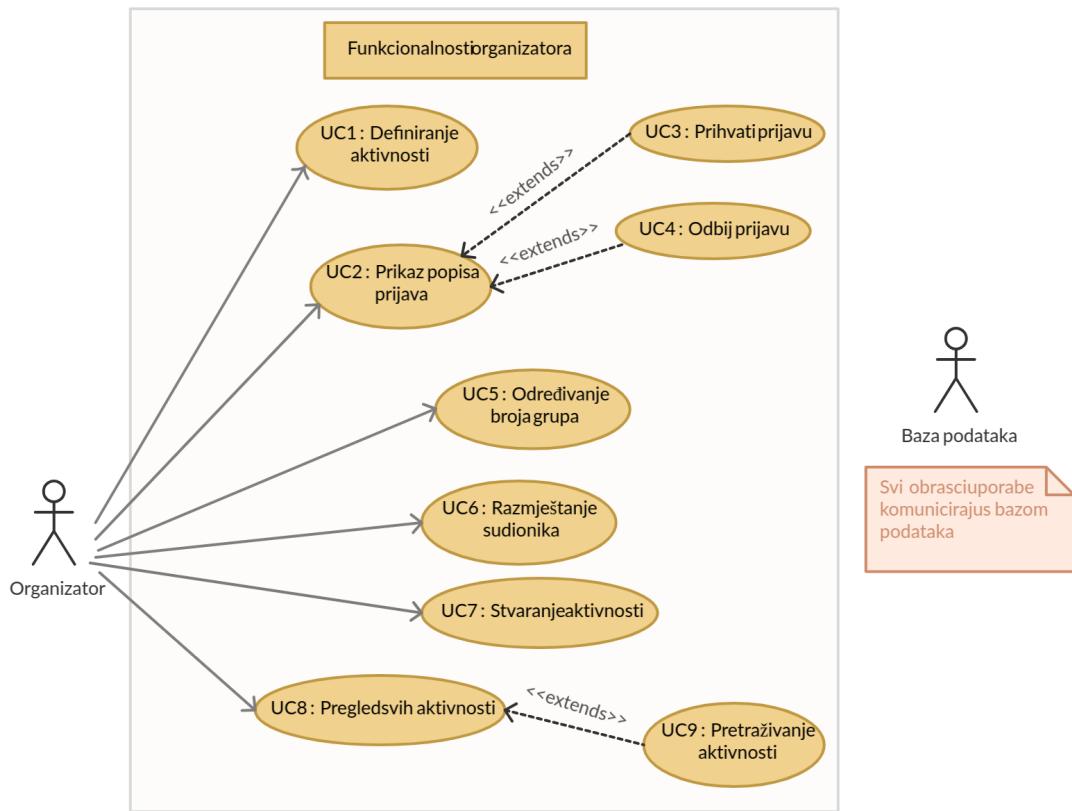
## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe korisnik-klijent



Slika 3.2: Dijagram obrasca uporabe animator-sudionik.

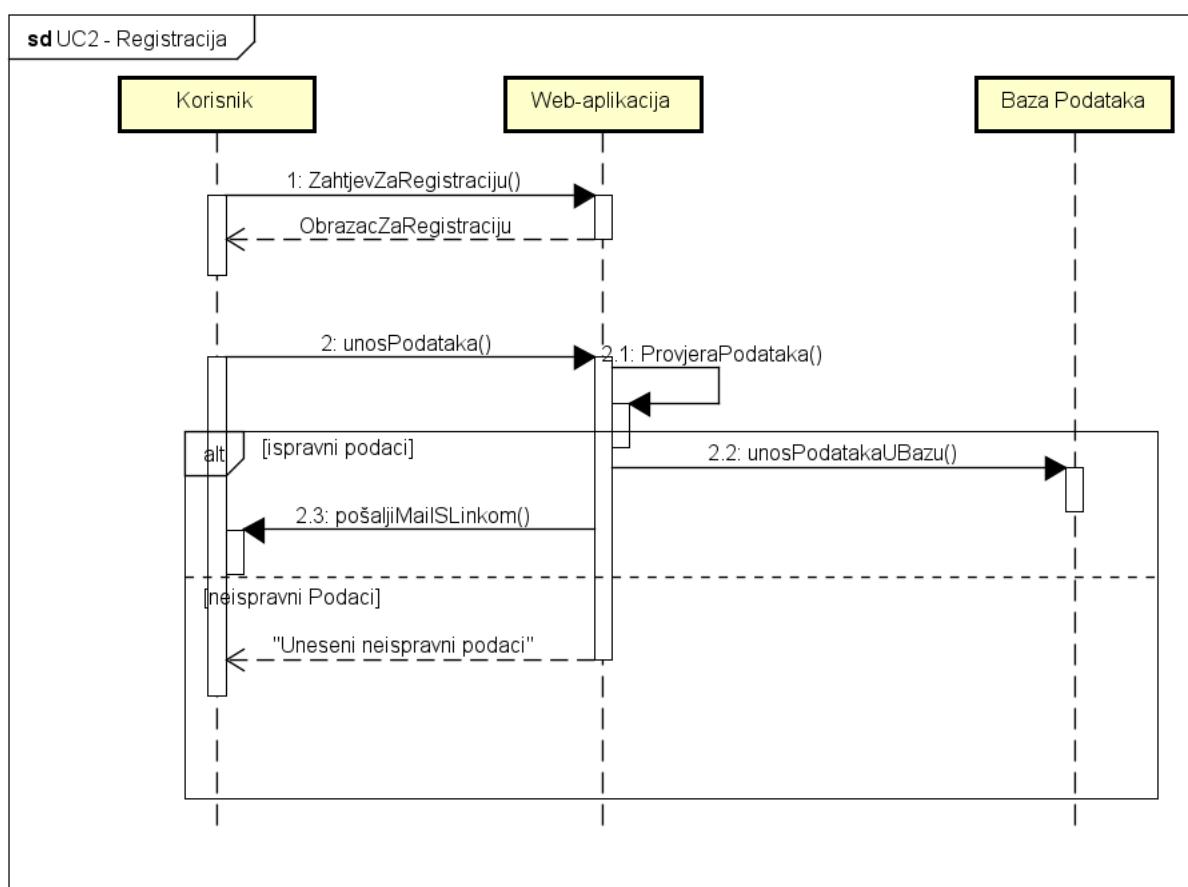


Slika 3.3: Dijagram obrasca uporabe organizatora.

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC2 - Registracija

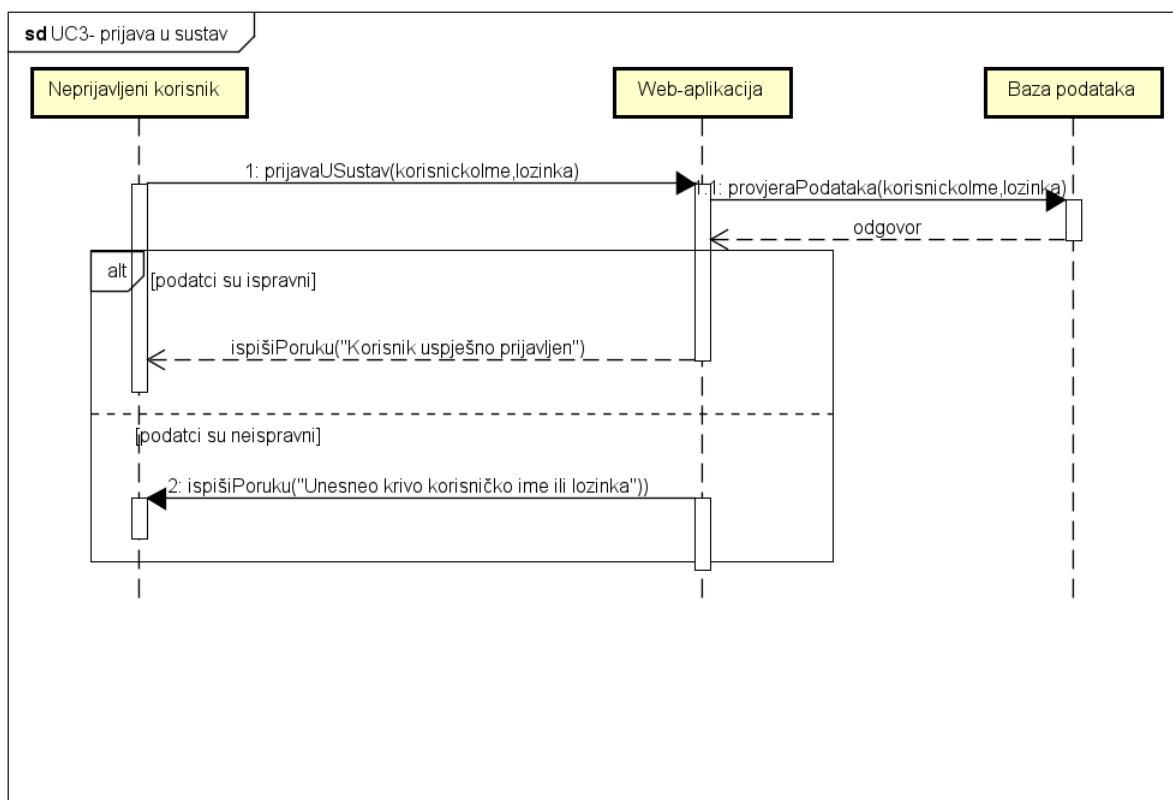
Neregistrirani korisnik odabire opciju za registraciju u određena polja upisuje osobne podatke kao što su ime, prezime, E-mail te datum rođenja. Zatim aplikacija provjerava ispravnost unesenih podataka te u slučaju neispravnih podataka obavještava korisnika. Ako su podaci ispravni unose se u bazu podataka i ukoliko je korisnik primljen u kamp na E-mail dobiva personalizirani link na kojem će unijeti svoju lozinku po vlastitom izboru.



Slika 3.4: Sekvencijski dijagram za UC2

### Obrazac uporabe UC5 - Prijava u sustav

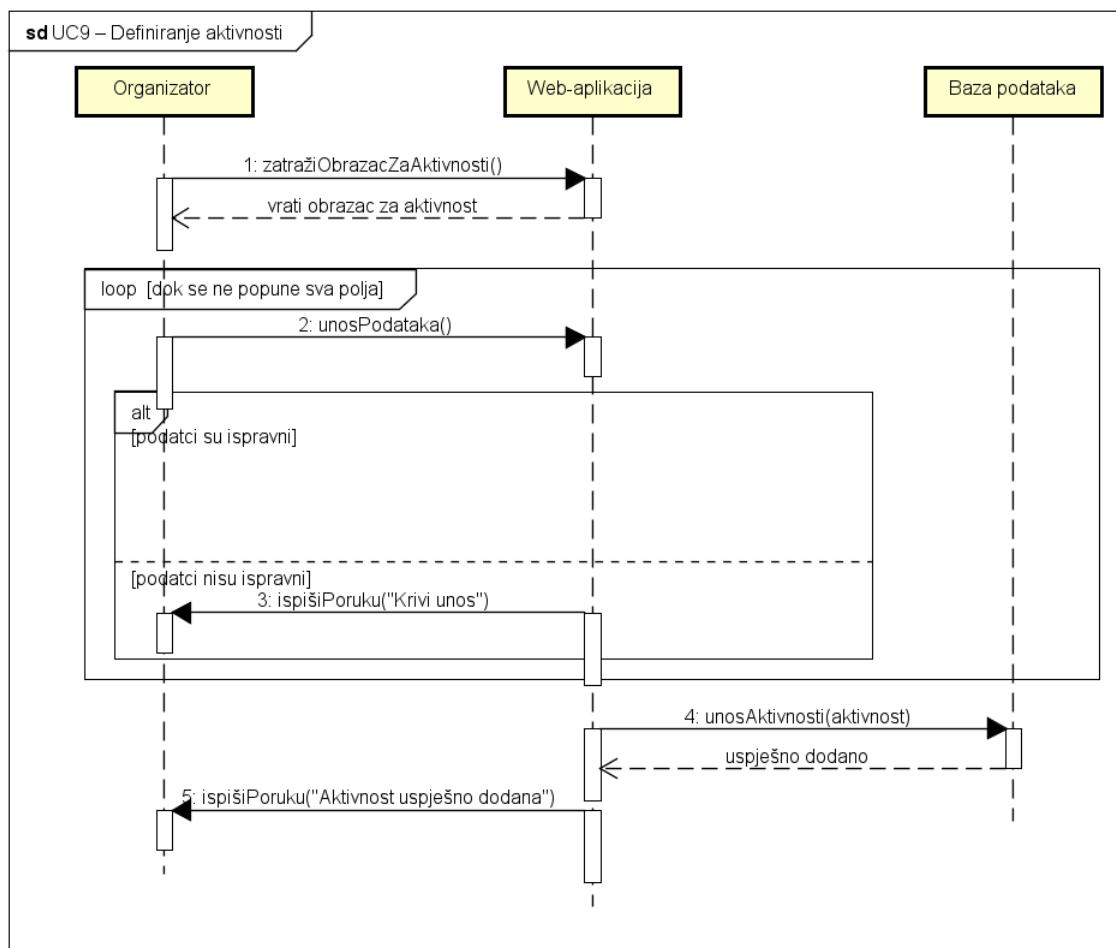
Neprijavljeni korisnik šalje zahtjev za prijavu u kojem unosi svoje korisničko ime i izabranu lozinku. Web-aplikacija provjerava ispravnost podataka tako da provjeri postoji li u bazi podataka predano korisničko ime, te ako postoji, je li tom korisničkom imenu pridjeljena upisana lozinka. U slučaju uspješne prijave neprijavljeni korisnik dobije ovlasti prijavljenog korisnika. U slučaju krivog unosa podataka korisnik dobije primjereni odgovor



Slika 3.5: Sekvencijski dijagram za UC5

### Obrazac uporabe UC13 - Definiranje aktivnosti

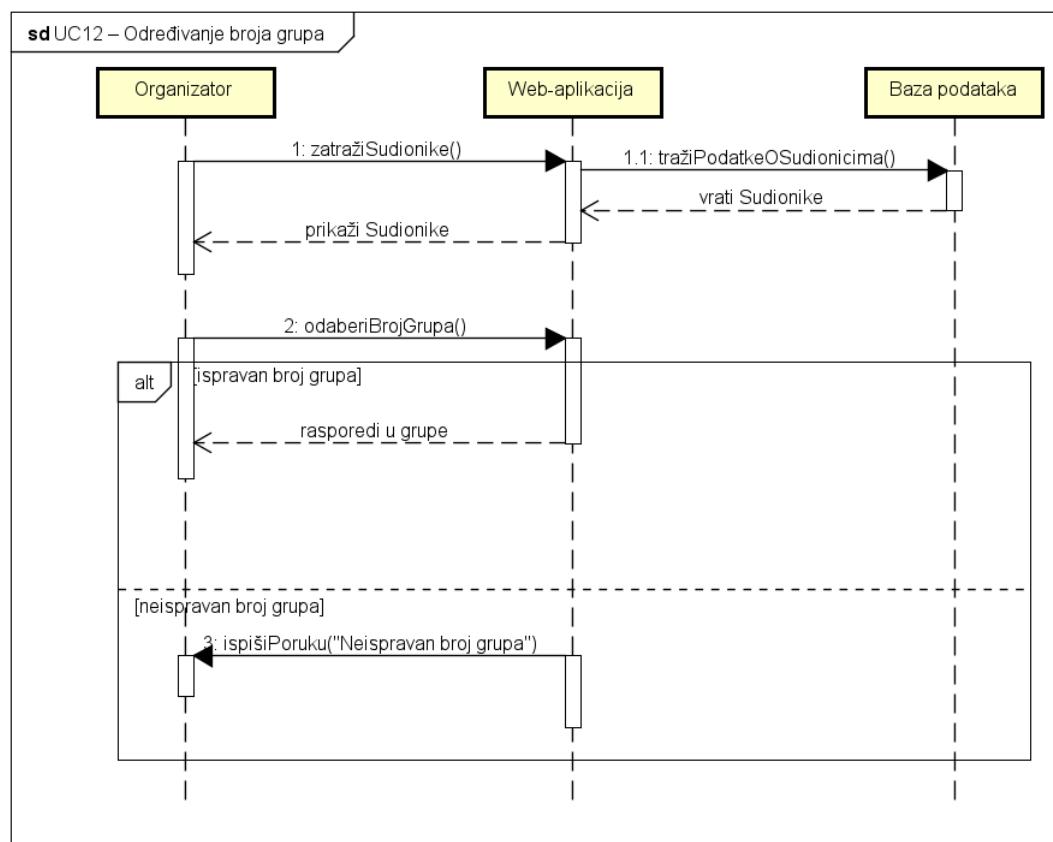
Organizator šalje zahtjev za definiranje nove aktivnosti. Organizator mora upisati ime aktivnosti, kratki opis, vremensko trajanje i tip aktivnosti. U slučaju da nije upisan bilo koji od podataka, ispisuje se poruka o grešci i organizatora se vraća na sučelje za organizatora. U slučaju da su uneseni svi parametri aktivnosti, nova aktivnost se upisuje u bazu podataka



Slika 3.6: Sekvencijski dijagram za UC13

### Obrazac uporabe U15 - Određivanje broja grupa

Nakon što je proces prijava korisnika za kamp završio, organizator vidi popis svih uspješno prijavljenih sudionika. Organizator u polje za unos podataka unosi broj koliko želi da bude grupa sudionika. U slučaju unosa tipa podatka koji nije broj ispisuje se poruka o grešci i omogućava se ponovan unos. U slučaju da je unesen broj grupa koji je veći od ukupnog broja sudionika ispisuje se poruka o grešci i omogućava se ponovan unos. U slučaju da je unesen ispravan broj grupa, formira se taj broj grupi a sudionici se razvrstavaju nasumično.



Slika 3.7: Sekvencijski dijagram za UC15

## 3.2 Ostali zahtjevi

- Sustav treba podržavati rad više korisnika u stvarnom vremenu
- Sustav treba podržavati hrvatsku abecedu bez gubitka funkcionalnosti
- Implementacija korisničkog sučelja mora biti korisniku jednostavna i razumljiva za korištenje
- Sustav treba prikazivati podatke na zathjev korisnika u prihvatljivom vremenu, zahtjevi ne smiju trajati dulje od nekoliko sekundi
- Sustav bi trebao garantirati točnost podataka koje prikazuje
- Sustav treba biti temeljen na načelima objektno orijentirane paradigme
- Ažuriranje sustava i naknadne promjene ne smiju narušavati postojeću funkcionalnost sustava
- Baza podataka mora biti sigurna i zaštićena od vanjskih utjecaja
- Korisnikov upit mora biti odgovoren u prihvatljivom vremenu od ne više od par sekundi
- Neispravno korištenje sustava od korisnika ne smije narušavati funkcionalost sustava
- Sustavu treba biti omogućen javni pristup preko sigurnog internetskog protokola HTTPS

## 4. Arhitektura i dizajn sustava

**Arhitektura se može podijeliti na tri podsustava:**

- *Web poslužitelj*
- *Web aplikacija*
- *Baza podataka*

**Web preglednik** je program koji korisniku omogucuje pregled web-stranica i multimedijalnih sadrzaja vezanih uz njih. Svaki internetski preglednik je prevoditelj. Dakle, stranica je pisana u kodu koji preglednik nakon toga interpretira kao nešto svakome razumljivo. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

**Web poslužitelj** osnova je rada web aplikacije. Njegova primarna zadaća je komunikacija klijenta s aplikacijom. Komunikacija se odvija preko HTTP (engl. Hyper Text Transfer Protocol) 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 cega preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku

Programski jezik kojeg smo odabrali za izradu naše web aplikacije je Java zajedno s Spring Boot radnim okvirom te JSP obrascima za frontend. Odabrano razvojno okruženje je IntelliJ IDEA. Arhitektura sustava temeljiti će se na MVC (Model-View-Controller) konceptu. MVC koncept podržan je od strane Spring Boot radnog okvira i kao takav ima gotove predloške koji nam olakšavaju razvoj web aplikacije.

Karakteristika MVC koncepta je nezavisan razvoj pojedinih dijelova aplikacije što za posljedicu ima jednostavnije ispitivanje kao i jednostavno razvijanje i dodavanje novih svojstava u sustav.

MVC koncept sastoji se od:

- **Model** - Središnja komponenta sustava. Predstavlja dinamičke strukture podataka, neovisne o korisnickom sučelju. Izravno upravlja podacima, logikom i pravilima aplikacije. Takoder prima ulazne podatke od Controllera
- **View** - Bilo kakav prikaz podataka, poput rasporeda. Mogući su različiti prikazi iste informacije poput grafičkog i tabličnog prikaza podataka.
- **Controller** - Prima ulaze i prilagođava ih za prosljeđivanje Modelu ili Viewu. Upravlja korisničkim zahtjevima i temeljem njih izvodi daljnju interakciju s ostalim elementima sustava.

## 4.1 Baza podataka

Kao implementaciju baze podataka u ovome projektu odabrali smo PostgreSQL RDMS. Baza podataka se sastoji od tablica koje predstavljaju entitete u Kampu mlade nade. Za potrebe naseg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olaksava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaca baze podataka je brza i jednostavna pohrana, izmjena i dohvata podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od entiteta koji su opisani u nastavku.

### 4.1.1 Opis tablica

#### Person

Ovaj entitet modelira osobu koja se prijavljuje na kamp, bilo kao sudionik ili kao animator. Sadrži atribute ID osobe, nickname, motivational letter, birthday, email, password, surname, name, telephone, accepted i nickhash. S entitetom Prijava u vezi je tipa One-to-One. Također služi kao nad-entitet za entitete Sudionik i Animator, koji nasljeđuju sve atribute. Osim toga, svaka osoba povezana je Many-to-One vezom sa entitetom Komentar.

Osoba		
ID osobe	VARCHAR	Jedinstveni identifikator sudionika.
motivational letter	VARCHAR	Motivacijsko pismo osobe.
birthday	DATE	Datum i godina rođenja osobe.
name	VARCHAR	Ime osobe.
surname	VARCHAR	Prezime osobe.
email	VARCHAR	Email osobe.
telephone	VARCHAR	Telefon osobe.
nickname	VARCHAR	Korisničko ime osobe.
nickHash	VARCHAR	Hash vrijednost korisničkog imena

### **Participant**

Ovaj entitet sadržava sve važne informacije o sudioniku kampa. Preuzima sve attribute nad-entiteta Osobe te uz to sadrži još jedan atribut - contact, koji predstavlja kontakt skrbnika za sudionike mlađe od 18 godina. Ovaj entitet je u vezi One-to-Many s entitetom Grupa, te ima dodatni vlastiti atribut Korisničko ime. Entitet modelira osobu koja prisustvuje kampu.

<b>Sudionik</b>		
contact	VARCHAR	Opcionalni atribut, broj mobitela skrbnika djeteta mlađeg od 18 godina.
ID osobe	INT	ID osobe koja se prijavila na kamp kao sudionik.
ID grupe	VARCHAR	ID grupe kojoj je osoba pridružena.

### **Animator**

Ovaj entitet sadržava sve važne informacije o animatoru kampa. Preuzima sve attribute nad-entiteta Osobe. Ovaj entitet je u vezi Many-to-Many s entitetom Aktivnost u vremenu, te ima dodatni vlastiti atribut Korisničko ime. Entitet modelira osobu koja je zadužena za provođenje aktivnosti u kampu.

<b>Animator</b>		
ID aktivnosti u vremenu	VARCHAR	Jedinstveni identifikator aktivnosti u vremenu kojoj je pridružen animator.
ID osobe	INT	ID osobe koja se prijavila na kamp kao animator.

### **Comment**

Ovaj entitet modelira komentare koje ostavljaju sudionici i animatori po završetku kampa. Sadrži atribute ID komentara, grade, description te ID osobe. Entitet je u vezi One-to-Many sa Osobom i One-to-Many s Aktivnosti.

<b>Komentar</b>		
ID komentara	VARCHAR	Jedinstveni identifikator komentara koji je osoba ostavila za aktivnost.
grade	INT	Ocjena aktivnosti.
description	VARCHAR	Komentar odnosno kratki dojam o aktivnosti.
ID osobe	VARCHAR	ID osobe koja je komentirala aktivnost.

### **Activity**

Ovaj entitet predstavlja aktivnosti koje kamp Mlade Nade nudi svojim polaznicima. Aktivnost se sastoji od sljedećih atributa: ID aktivnosti, name, description, duration, type, ID aktivnosti u vremenu, ID komentara. Povezana je Many-to-One vezom s entitetom Komentar, te vezom Many-to-One s Aktivnosti u vremenu.

<b>Aktivnost</b>		
ID aktivnosti	VARCHAR	Jedinstveni ID aktivnosti.
name	TIME	Ime aktivnosti.
description	VARCHAR	Jedinstveni opis aktivnosti.
duration	VARCHAR	Trajanje aktivnosti.
type	VARCHAR	Tip aktivnosti.
ID activity in time	VARCHAR	ID aktivnosti u vremenu, preciznije to je konkretna implementacija Aktivnosti.
ID comment	VARCHAR	ID komentara koji su osobe postavile na aktivnosti.

### Activity in time

Ovaj entitet predstavlja aktivnost u kampu koja se odvija u specifičnom vremenu, odnosno predstavlja aktivnost koju sudionici, animatori i organizatori vide u svojim rasporedima. Od atributa ima ID aktivnosti u vremenu, end i start. Primarni ključ ovog entiteta je ID aktivnosti u vremenu. Entitet je u One-to-Many vezi s Aktivnosti, Many-to-Many vezni sa Animatorom te Many-to-Many vezom sa Grupom.

Aktivnost u vremenu		
start	DATE	Početak aktivnosti.
end	DATE	Kraj aktivnosti.
ID aktivnosti u vremenu	VARCHAR	Ime aktivnosti.
ID grupe	VARCHAR	ID grupe koja sudjeluje na trenutnoj aktivnosti.
Korisničko ime	VARCHAR	Korisničko ime animatora koji je zadužen za tu aktivnost.

### Group

Ovaj entitet modelira grupe sudionika na kampu. O broju i sadržaju grupa brine se organizator na početku kampa. Entitet sadrži atribut ID grupe, ime sudionika i ID aktivnosti u vremenu. Povezan je Many-to-One vezom sa Sudionikom te Many-to-Many vezom sa Aktivnosti u vremenu.

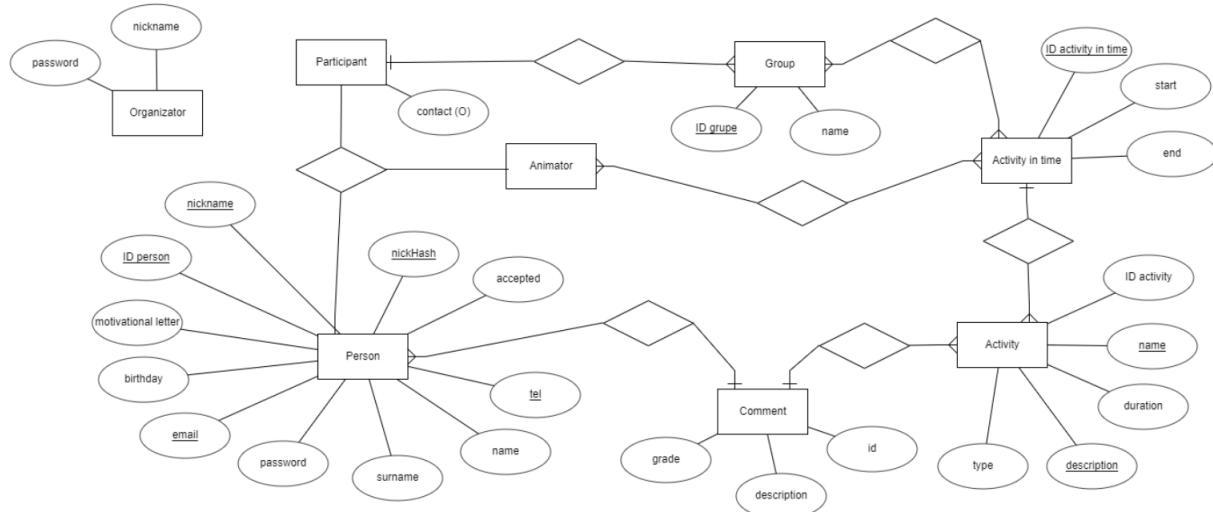
Grupa		
ID grupe	VARCHAR	Jedinstveni ID grupe.
Ime sudionika	VARCHAR	Ime sudionika koji pripada grupi.
ID aktivnosti u vremenu	VARCHAR	ID aktivnosti kojoj određena pripada u nekom vremenu.

**Organizator**

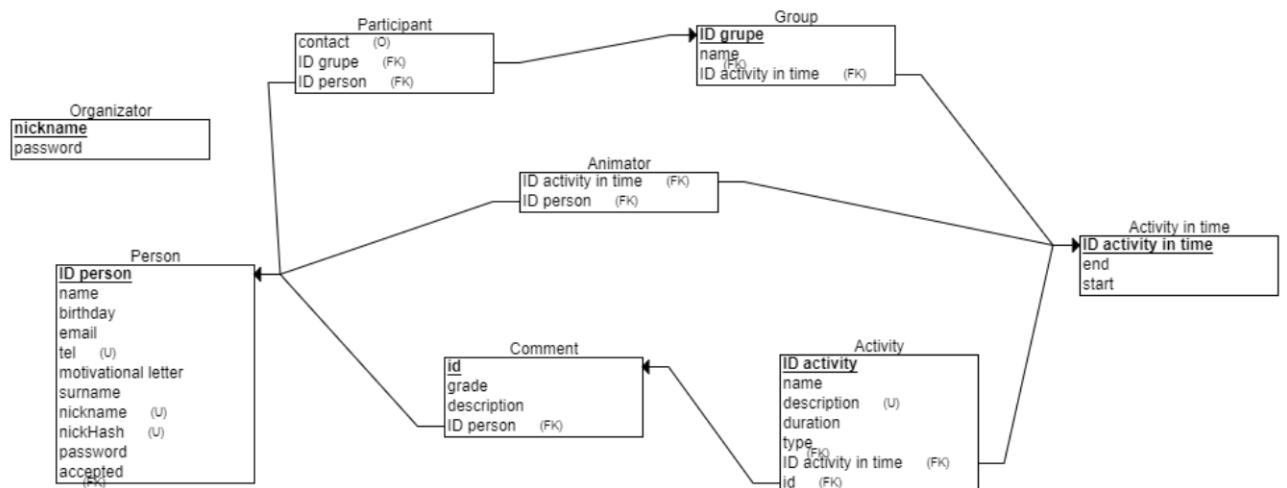
Ovaj entitet modelira organizatora kampa. Sadrži atribute nickname i password. On ima najveće ovlasti od svih te nije povezan niti s jednim od ostalih entiteta.

<b>Grupa</b>		
nickname	VARCHAR	Korisničko ime organizatora.
password	VARCHAR	Password organizatora.

#### 4.1.2 Dijagram baze podataka

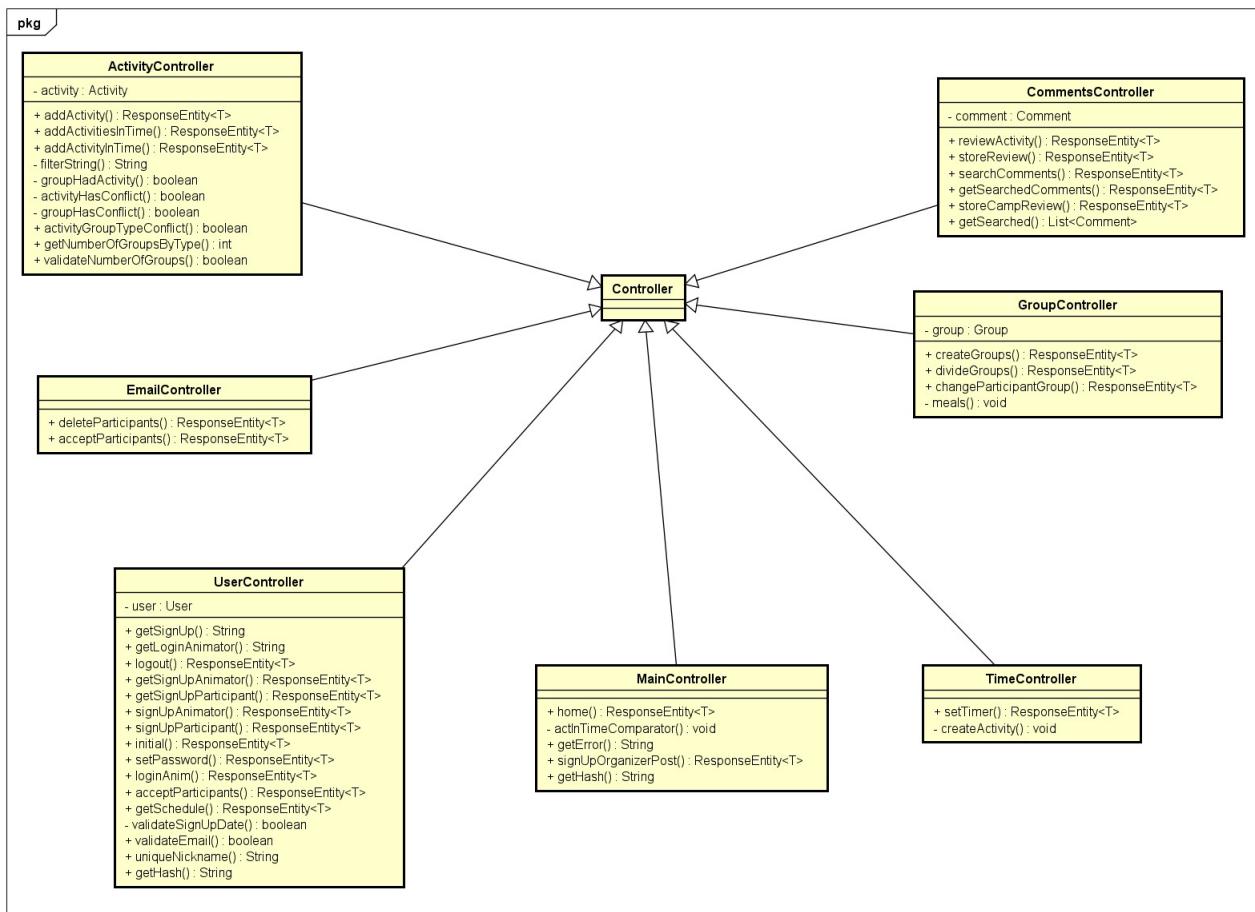


Slika 4.1: ER dijagram baze podataka

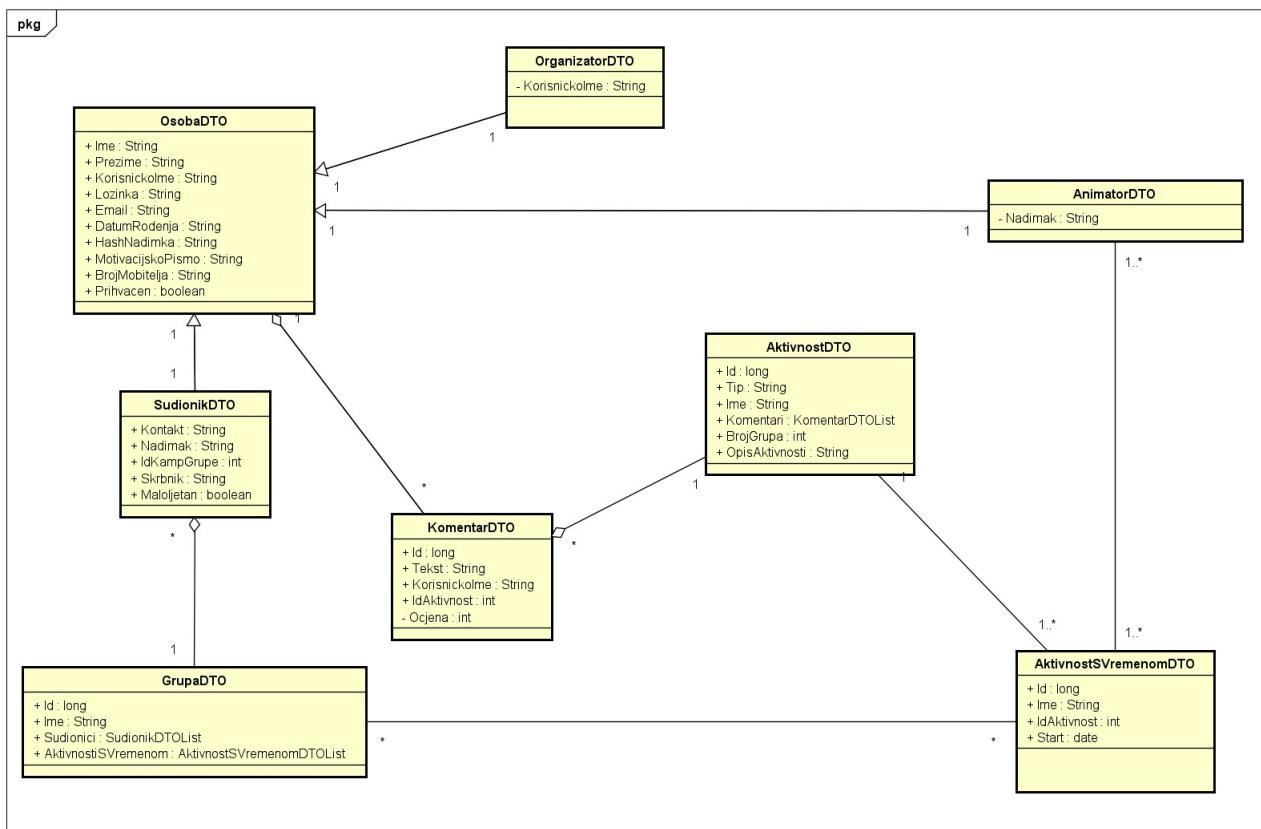


Slika 4.2: Relacijska shema baze podataka

## 4.2 Dijagram razreda



Slika 4.3: Controlleri

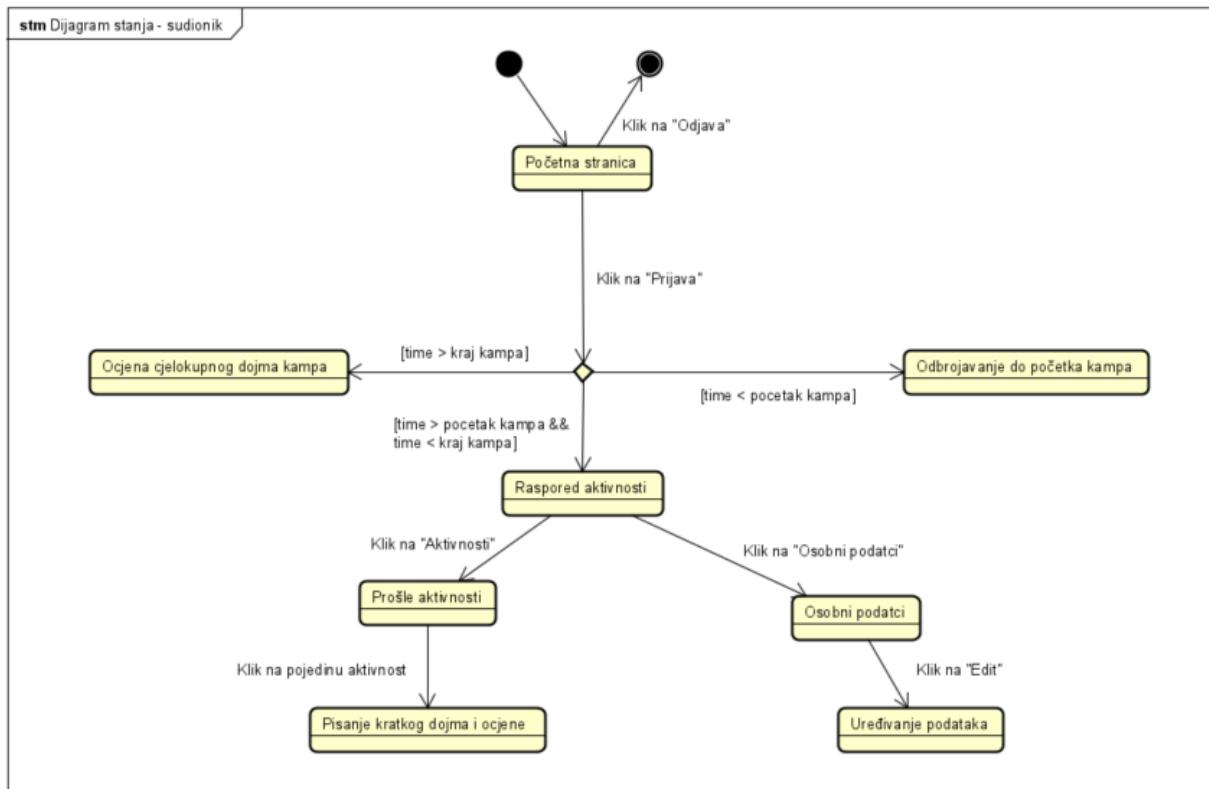


Slika 4.4: DTO

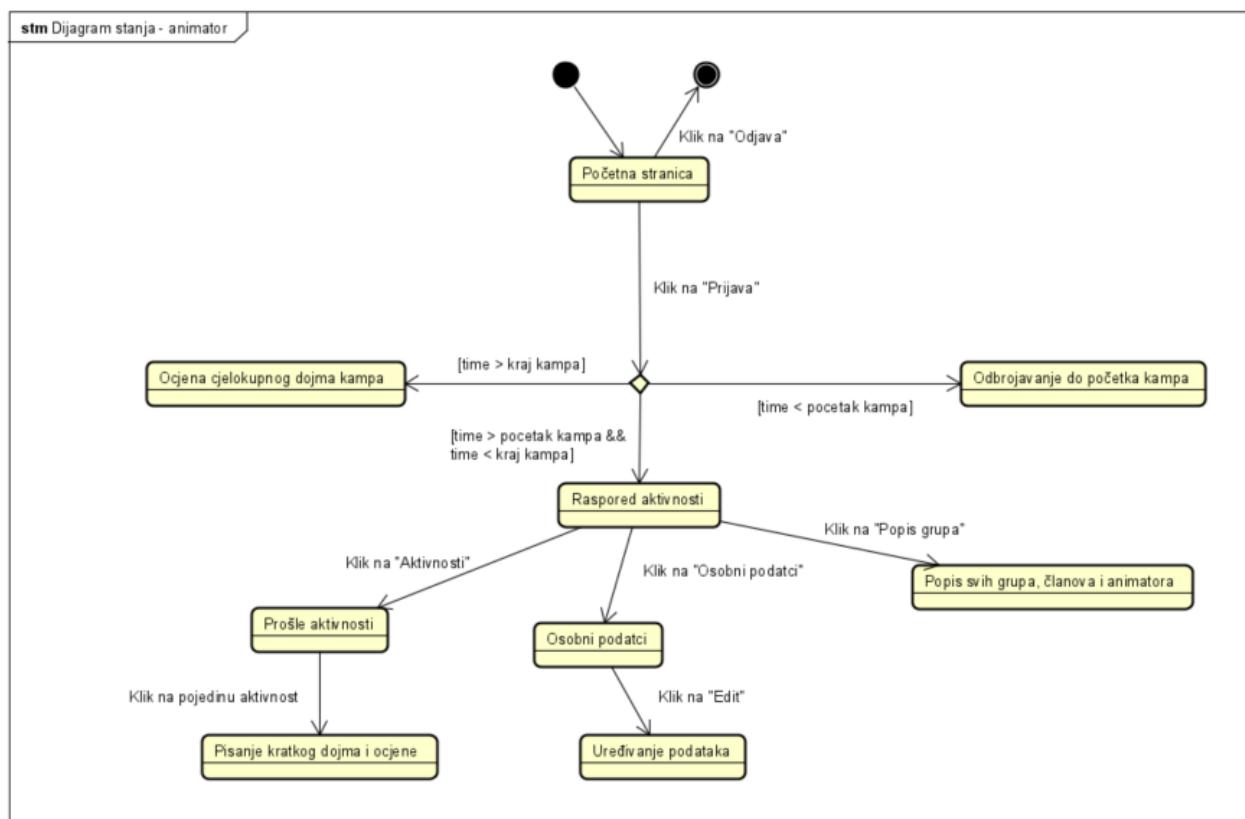


Slika 4.5: Dijagram razreda

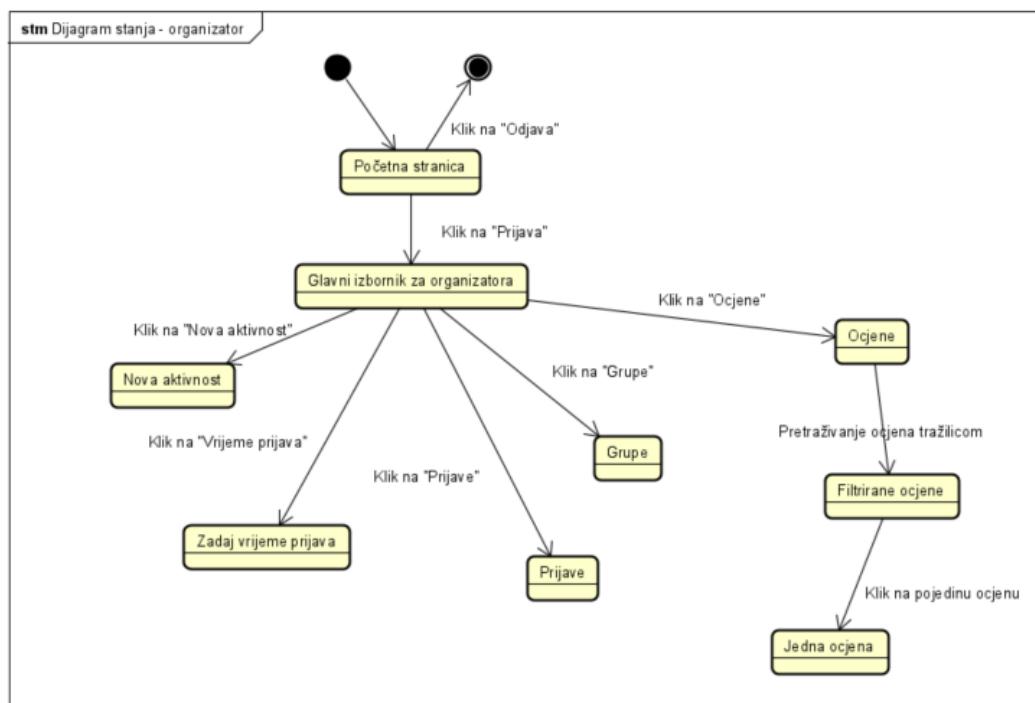
## 4.3 Dijagram stanja



Slika 4.6: DIjagram stanja sudionika

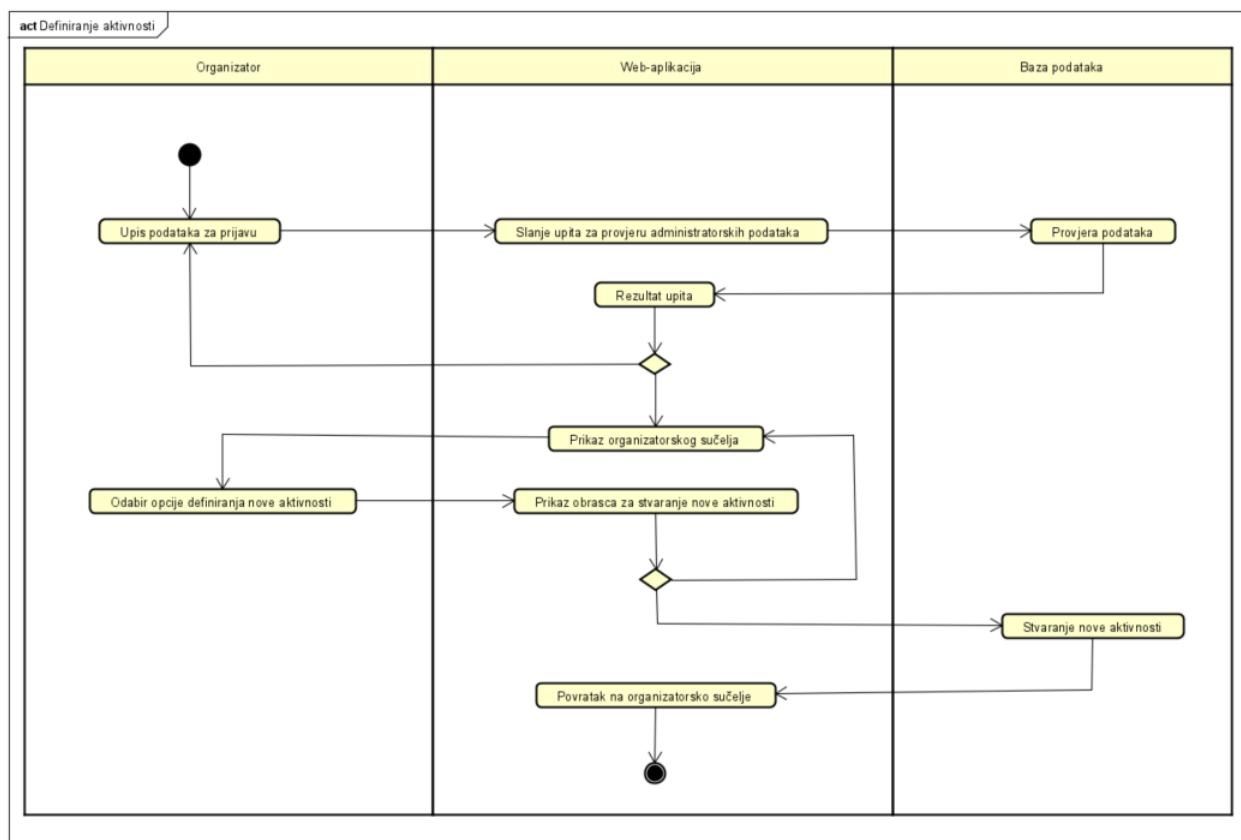


Slika 4.7: DIjagram stanja animatora

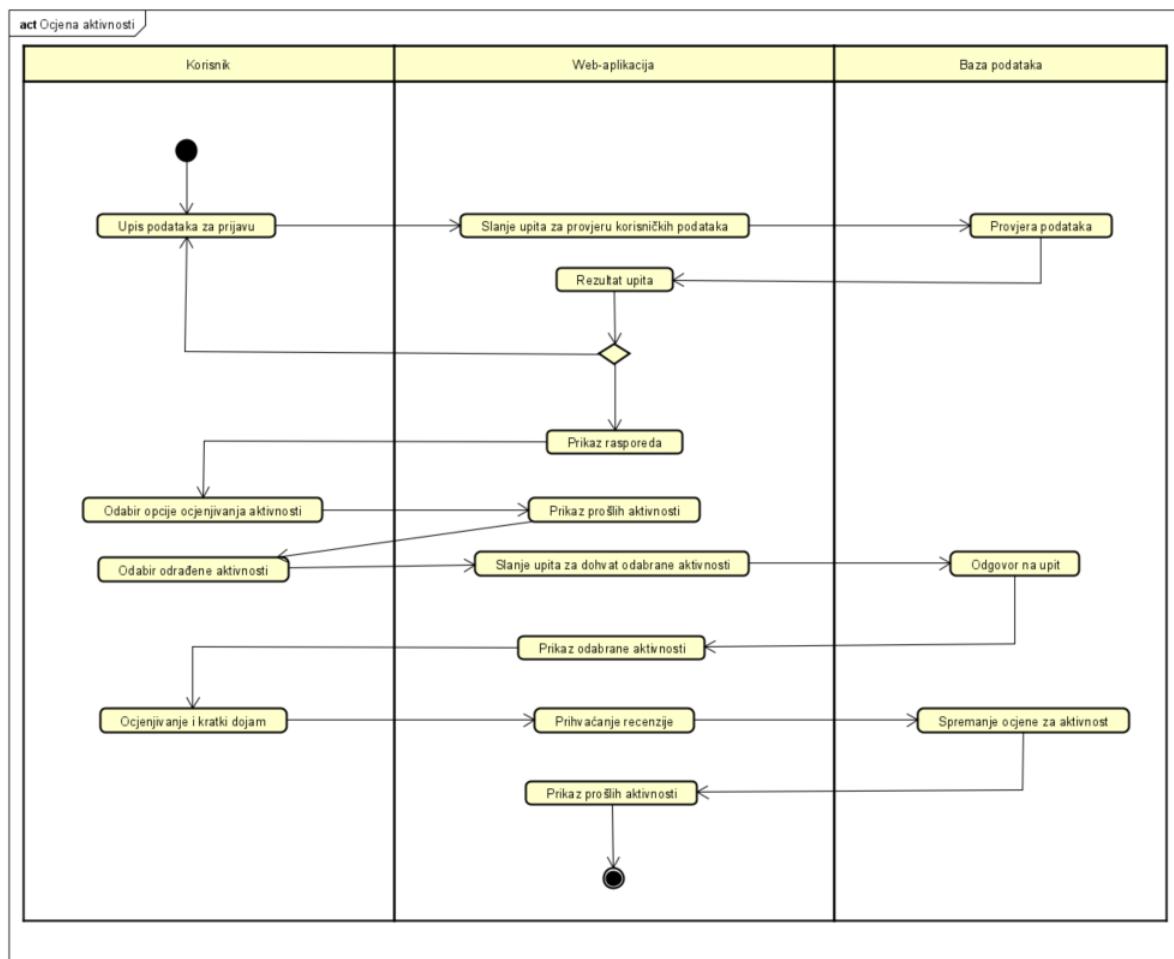


Slika 4.8: DIjagram stanja organizatora

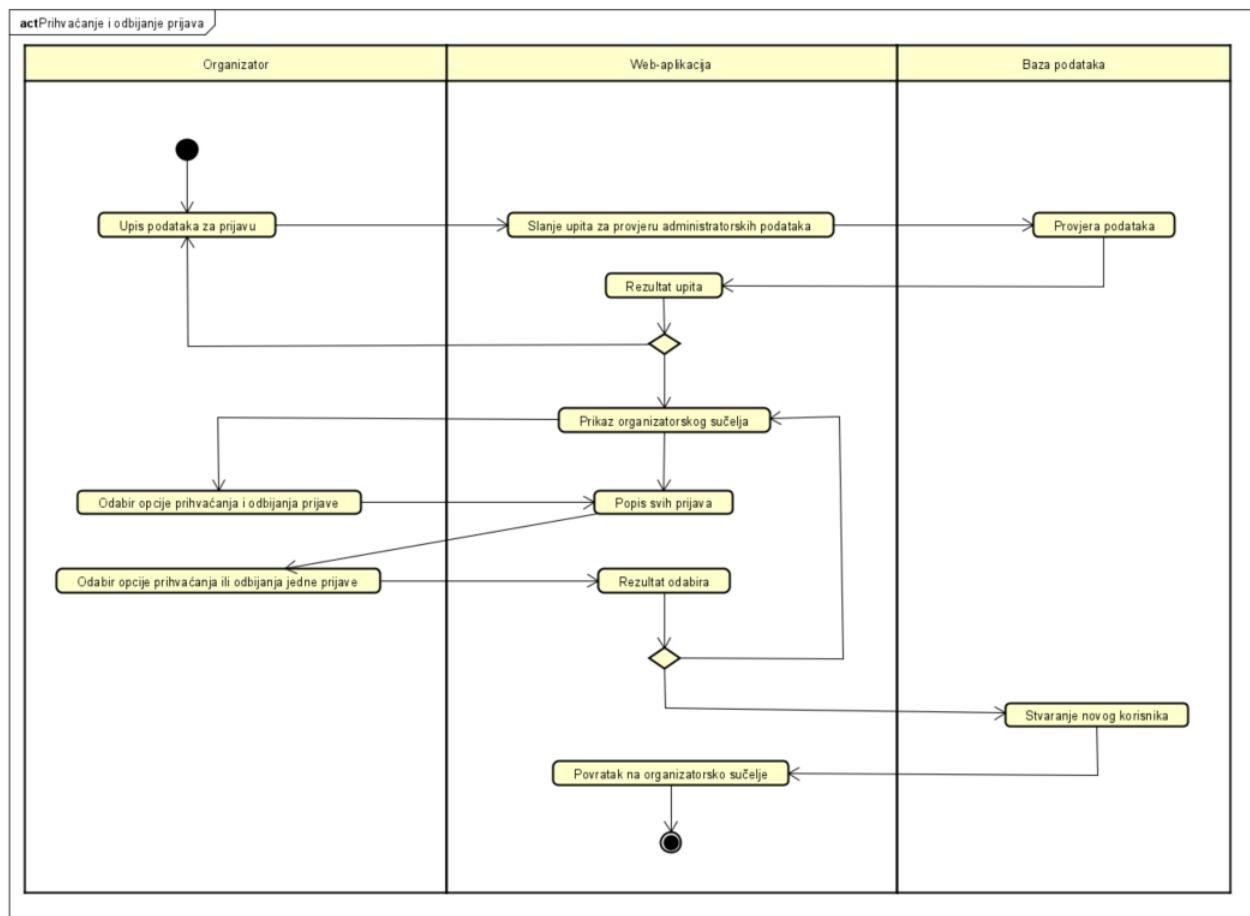
## 4.4 Dijagram aktivnosti



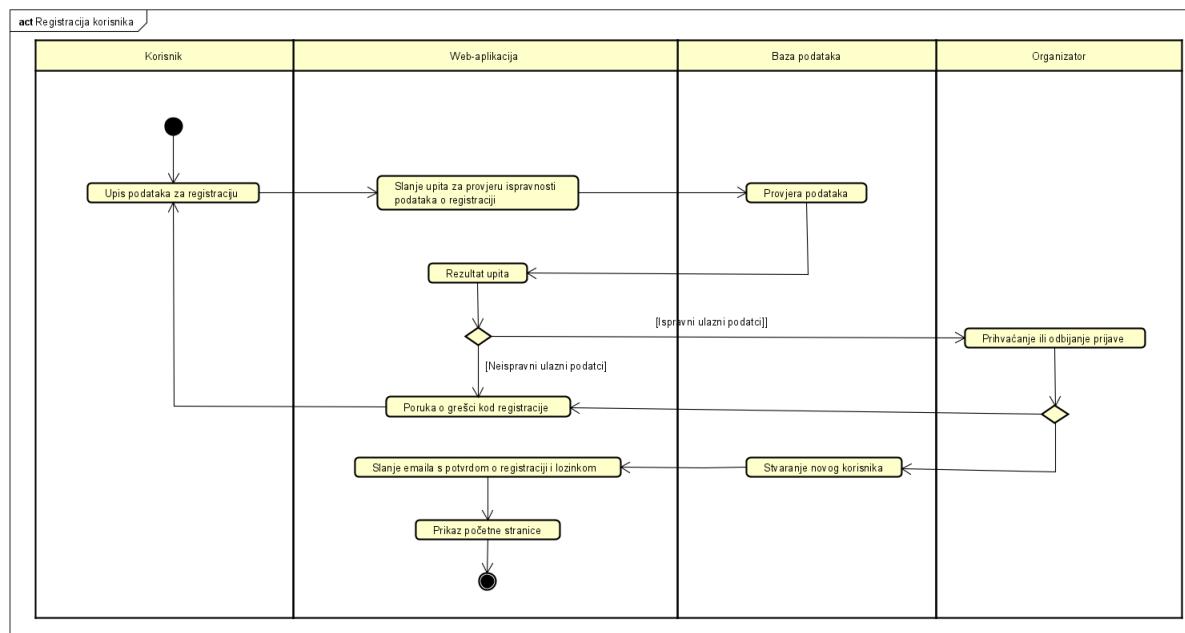
Slika 4.9: Definiranje aktivnosti



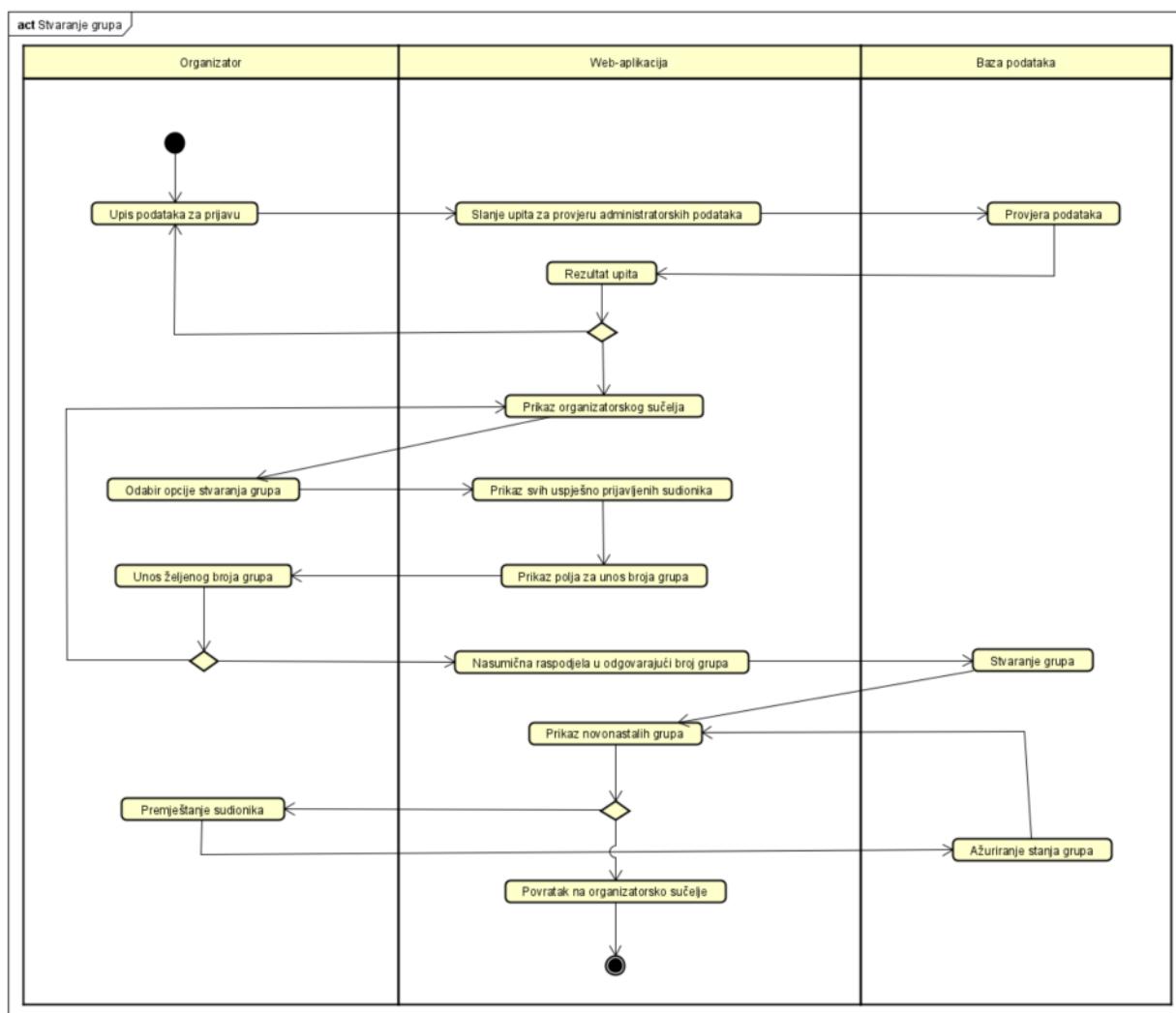
Slika 4.10: Ocjena aktivnosti



Slika 4.11: Prihvatanje i odbijanje prijava korisnika

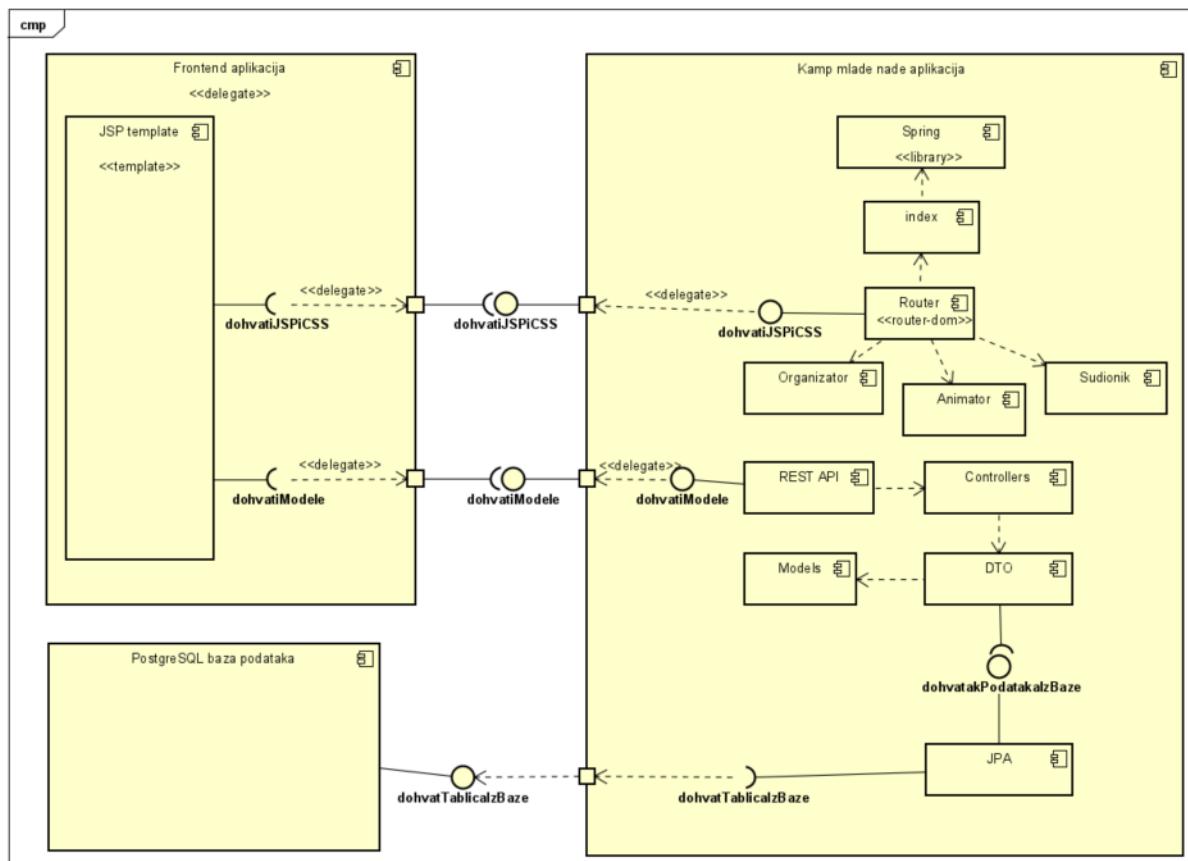


Slika 4.12: Registracija korisnika



Slika 4.13: Stvaranje grupe

## 4.5 Dijagram komponenti



Slika 4.14: Dijagram komponenti

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

Aplikacije korištene za komunikaciju među članovima tima su slack<sup>1</sup>, whatsapp<sup>2</sup> i microsoft teams<sup>3</sup>. Za izradu dijagrama obrazaca uporabe korišten je creately<sup>4</sup>, a za sekvencijske dijagrame i dijagrame razreda korišten je Astah Professional<sup>5</sup>. Korišten je distribuirani sustav za upravljanje izvornog koda git<sup>6</sup> čiji je udaljeni repozitorij dostupan na platformi gitlab<sup>7</sup>.

Za razvoj računalnog softvera korišten je Inetllij<sup>8</sup>, integrirano razvojno okruženje napisano na Javi. Pretežno se koristi u razvoju web-aplikacija, web-stranica i mobilnih aplikacija.

Aplikacija je pisana koristeći javni okvir Spring Boot<sup>9</sup> i jeziku Javu<sup>10</sup> za pisanje *backenda* te Java Server Pages<sup>11</sup> za prikaz stranica. Java Server Pages (JSP) je programerska tehnologija poslužitelja koja omogućuje kreiranje i prikaz dinamičkih web-aplikacija. JSP je vrlo koristan jer ima direktni pristup Java API-ima. Radni okvir Spring Boot omogućuje programerima pisanje manje koda za postizanje jednakih funkcionalnosti. Okvir je primarno namjenjen kako bi programerima olakšao i ubrzao posao nudeći već gotove funkcionalnosti.

Baza podataka se nalazi na poslužitelju Heroku<sup>12</sup>.

---

<sup>1</sup><https://slack.com/intl/en-hr/>

<sup>2</sup><https://www.whatsapp.com/>

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

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

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

<sup>6</sup><https://git-scm.com/>

<sup>7</sup><https://about.gitlab.com/>

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

<sup>9</sup><https://spring.io/>

<sup>10</sup><https://java.com/en/>

<sup>11</sup><https://www.oracle.com/java/technologies/jspt.html>

<sup>12</sup><https://www.heroku.com/>

## 5.2 Ispitivanje programskog rješenja

Programsku potporu nužno je ispitati budući da programi ne daju nikakvo jamstvo da će raditi pod svim mogućim okolnostima. Testiranje aplikacije je ključno jer poboljšava kvalitetu proizvoda, čini ga jednostavnijim za korištenje i osigurava zadovoljstvo korisnika.

### 5.2.1 Ispitivanje komponenti

#### Ispitni slučaj 1: testiranje funkcionalnosti metode getSearched()

U ovom testu cilj je provjeriti ispravnost rada metode koja pretražuje ocjene koje su sudionici i animatori ostavili za svaku pojedinu aktivnost. U slučaju da se za kategoriju odabere String vrijednost koja ne spada u predefiniran skup odgovarajućih vrijednosti, test bi trebao baciti IllegalArgumentException.

Ulaz:

1. String koji predstavlja naziv neke kategorije.
2. String koji predstavlja operaciju koju izvodimo, u ovom slučaju pretraživanje

Očekivani rezultat:

1. IllegalArgumentException

```
public class MyControllerTest {  
  
    @Test(expected = IllegalArgumentException.class)  
    public void testGetSearchedThrowsIllegalArgumentException(){  
        String category = "neka kategorija";  
        String search = "search";  
        MyController controller = new MyController();  
        controller.getSearched(category, search);  
    }  
}
```

Rezultat izvođenja testa jest javljanje IllegalArgumentException, te su uvjeti testa zadovoljeni, i aplikacija je prošla test.

**Ispitni slučaj 2: testiranje funkcionalnosti metode validateEmail()**

Cilj ovog testa jest provjeriti rad funkcije testEmailValidation na način da se za 2 String vrijednosti (od kojih je jedna ispravna a druga neispravna) pozove dotična metoda.

Ulaz:

1. String koji predstavlja email korisnika

Očekivani rezultat:

1. za ispravan format email adrese vraća true
2. za neispravan format email adrese vraća false

```
@Test  
public void testEmailValidation(){  
    String notEmail = "asfafsgsdga";  
    String email = "ivo.peric@gmail.com";  
    MyController controller = new MyController();  
    assert controller.validateEmail(notEmail) == false;  
    assert controller.validateEmail(email) == true;  
}
```

Rezultat izvođenja testa je odgovarajuć, te su uvjeti testa zadovoljeni, i aplikacija je prošla test.

**Ispitni slučaj 3: testiranje funkcionalnosti metode getHash()**

Rad metode testHashingFunction provjeravamo na način da kao ulaz u funkciju pripremimo proizvoljan String tekst, za koji unaprijed znamo koji mu je hash kod. Pozivom naše funkcije, promatramo rezultat.

Ulaz:

1. String koji predstavlja String tekst za koji provjeravamo hash kod.
2. String vrijednost unaprijed izračunatog hash koda.

Očekivani rezultat:

1. usporedba unaprijed izračunate hash vrijednosti i rezultata funkcije je jednak

```
@Test  
public void testHashingFunction(){  
    String str = "Abeceda";  
    String strHash = "3d356b73b7e08f0190ee6071c5b9b75f";  
    MyController controller = new MyController();  
    assert strHash.equals(controller.getHash(str));  
}
```

Rezultat izvođenja testa je odgovarajuć, te su uvjeti testa zadovoljeni, i aplikacija je prošla test.

#### Ispitni slučaj 4: testiranje dodatne funkcionalnosti metode getHash()

Rad metode testHashingFunction dodatno ćemo provjeriti tako što ćemo ponoviti postupak kao u prethodnom testu, samo što ćemo umjesto ispravne vrijednosti unaprijed izračunate hash vrijednosti, koristiti neispravnu vrijednost.

Ulaz:

1. String koji predstavlja String tekst za koji provjeravamo hash kod.
2. String vrijednost unaprijed izračunatog krivog hash koda.

Očekivani rezultat:

1. usporedba unaprijed izračunate hash vrijednosti i rezultata funkcije je različita

```
@Test  
public void testHashingFunction2(){  
    String str = "Zivotinje";  
    String strHash = "3d356b73b7e08f0190ee6071c5b9b75f";  
    MyController controller = new MyController();  
    assert !strHash.equals(controller.getHash(str));  
}
```

Rezultat izvođenja testa je odgovarajuć, te su uvjeti testa zadovoljeni, i aplikacija je prošla test.

**Ispitni slučaj 5: testiranje funkcionalnosti dodavanja aktivnosti u raspored**

Cilj ovog testa jest provjeriti ispravnost rada funkcije testActivityComparator. Konkretnije, ispitujemo je li redoslijed aktivnosti u rasporedu ispravan.

Ulaz:

1. Aktivnost u vremenu koja počinje 15.5.2020.
2. Aktivnost u vremenu koja počinje 1.1.2021.

Očekivani rezultat:

1. U prvom slučaju, pozivom metode after(), koja utvrđuje je li prva aktivnost nakon druge u rasporedu aktivnosti, rezultat očekujemo da bude false
2. U drugom slučaju, zamijenimo uloge aktivnosti te očekujemo rezultat true.

```
@Test
public void testActivityComparator(){
    ActivityInTime act1 = new ActivityInTime( name: "Odbojka", new Date( s: "15.5.2020."), activity: null, groups: null, animators: null);
    ActivityInTime act2 = new ActivityInTime( name: "Nogomet", new Date( s: "1.1.2021."), activity: null, groups: null, animators: null);

    assert !act1.getStart().after(act2.getStart());
    assert act1.getStart().after(act2.getStart());
}
```

Rezultat izvođenja testa je odgovarajuć, te su uvjeti testa zadovoljeni, i aplikacija je prošla test.

**Ispitni slučaj 6: testiranje funkcionalnosti metode validateNumberOfGroups()**

Cilj ovog testa jest provjeriti ispravnost rada funkcije testActivityComparator. Konkretnije, ispitujemo hoće li aplikacija baciti grešku ako kao tip aktivnosti pošaljemo neispravnu vrijednost.

Ulaz:

1. Aktivnost s neispravno postavljenim tipom aktivnosti

Očekivani rezultat:

1. Pozivom metode, aplikacija treba baciti IllegalArgumentException

```
@Test(expected = IllegalArgumentException.class)
public void testGroupNumberValidator(){
    Activity act = new Activity( name: "", activityDescription: "", durationInMinutes: 0, fer.ppj.model.ActivityType.valueOf("la"), numberOfGroups: 0);
    MyController controller = new MyController();
    controller.validateNumberOfGroups(act, new String[5]);
}
```

Rezultat izvođenja testa je odgovarajuć, te su uvjeti testa zadovoljeni, i aplikacija je prošla test.

## 5.2.2 Ispitivanje sustava

Svi testovi izvršeni su pomoću Selenium IDE-a. Ispitivanje sustava je provedeno po obrascima uporabe kako bi se provjerila osnova funkcionalnost sustava, ali i nasumičnim kretanjima po aplikaciji kako bi se pronašle neočekivane greške ili nepredviđena ponašanja. Prikazivanje ispitivanja UC2, UC3, UC 13, UC 15.

### Ispitni slučaj 1: Registracija

#### Ulaz:

1. Korisnik odabire opciju registracije
2. Korisnik unosi potrebne korisničke podatke
3. Korisnik prima email o uspješnoj registraciji i odabire šifru za svoj račun u sustavu

#### Očekivani rezultat:

1. Korisnik je primio email i registrirao se

#### Rezultat:

Command	Target	Value
open	https://kamp-mlade-nade-progi.herokuapp.com/	
set window size	1920x977	
click	linkText=Registracija za sudionika	
click	id=name	
type	id=name	Petar
click	id=surname	
type	id=surname	Jukić
click	id=email	
type	id=email	petar.jukic7@gmail.com
click	id=contact	
type	id=contact	0989632519
click	id=birthday	

Log

```

12. click on id=birthday OK
21:18:28
13. type on id=birthday with value 1999-06-20 OK
21:18:29
14. click on id=motivation OK
21:18:30
15. type on id=motivation with value Jako želim pristupiti vašem kampu! OK
21:18:31
16. click on css=.round-button OK
21:18:32
'registracija_sudionika' completed successfully
21:18:33

```

Korisnik je uspješno registriran i primio je mail.



Korisnik primi mail ako je uspješno registriran.

### Ispitni slučaj 2: Prijava u sustav (prije početka kampa) - sudionici

**Ulaz:**

1. Unos korisničkog imena i lozinke
2. Potvrda o ispravnosti unesenih podataka

**Očekivani rezultat:**

1. Korisniku se prikazuje sat koji odbrojava do početka kampa.

**Rezultat:**

Selenium IDE - progi\_tests

Project: progi\_tests

Tests + REC

Search tests... ...

https://kamp-mlade-nade-progi.herokuapp.com/

	Command	Target	Value
1	open	https://kamp-mlade-nade-pr ogi.herokuapp.com/	
2	set window size	1920x977	
3	click	linkText=Prijava	
4	click	id=name	pjukic
5	click	id=password	oc00123

Command: click Target: css=.round-button Value: Description:

Log Reference

5. click on id=password with value miocuu123 OK

6. click on css=.round-button OK

'prijava' completed successfully

Test je prošao.

Kamp počinje za  
1dana 23sati 57minuta 54sekundi

Odbrojavanje sata.

### Ispitni slučaj 3: Definiranje aktivnosti

#### Ulaz:

- Organizator unosi podatke za definiranje nove aktivnosti (ime, kratki opis, vremensko trajanje, tip aktivnosti)

#### Očekivani rezultat:

- Stvorena je nova aktivnost

#### Rezultat:

Command	Target	Value
open	https://kamp-mlade-nade-progi.herokuapp.com/	
set window size	1920x977	
click	linkText=dodaj aktivnost	
click	id=name	
type	id=name	Badminton
click	id=desc	
type	id=desc	Učenje osnova badmintona
click	id=duration	37
type	id=duration	
click	css=.container:nth-child(5) > .checkbox	
click	id=numberOfGroups	
type	id=numberOfGroups	3
click	css=.round-button	

Log Reference

```

12. click on id=birthday OK
13. type on id=birthday with value 1999-06-20 OK
14. click on id=motivation OK
15. type on id=motivation with value Jako želim pristupiti vašem kampu! OK
16. click on css=.round-button OK
'registracija_sudionika' completed successfully

```

Test je prošao.

KAMP MЛАДЕ НАДЕ		
Naziv	Opis	Trajanje
Badminton	Učenje osnova badmintona	37

Aktivnost je dodana.

#### Ispitni slučaj 4: Određivanje broja grupa

**Ulaz:**

1. Organizator vidi popis svih uspješno prijavljenih sudionika i odabire željeni broj grupa u brojčanom obliku

**Očekivani rezultat:**

1. Izvršava se automatska dodjela sudionika u grupe

## Rezultat:

Selenium IDE - progi\_tests

Project: progi\_tests

Tests + Search tests... REC

http://localhost:8080/

	Command	Target	Value
1	open	http://localhost:8080/	
2	set window size	1920x977	
3	click	linkText=stvori grupe	
4	click	id=numberOfGroups	
5	type	id=numberOfGroups	1
6	click	id=numberOfGroups	
7	type	id=numberOfGroups	2
8	click	id=numberOfGroups	
9	double click	id=numberOfGroups	
10	type	id=numberOfGroups	3
11	click	id=numberOfGroups	
12	click	css=.round-button	

Command: open | Target: http://localhost:8080/ | Value:   |   |  

Target: http://localhost:8080/ |   |   |  

Description:  

Log Reference

9. type on id=duration with value 37 OK 21:12:22

10. click on css=.container:nth-child(5) > .checkmark OK 21:12:23

11. click on id=numberOfGroups OK 21:12:24

12. type on id=numberOfGroups with value 3 OK 21:12:25

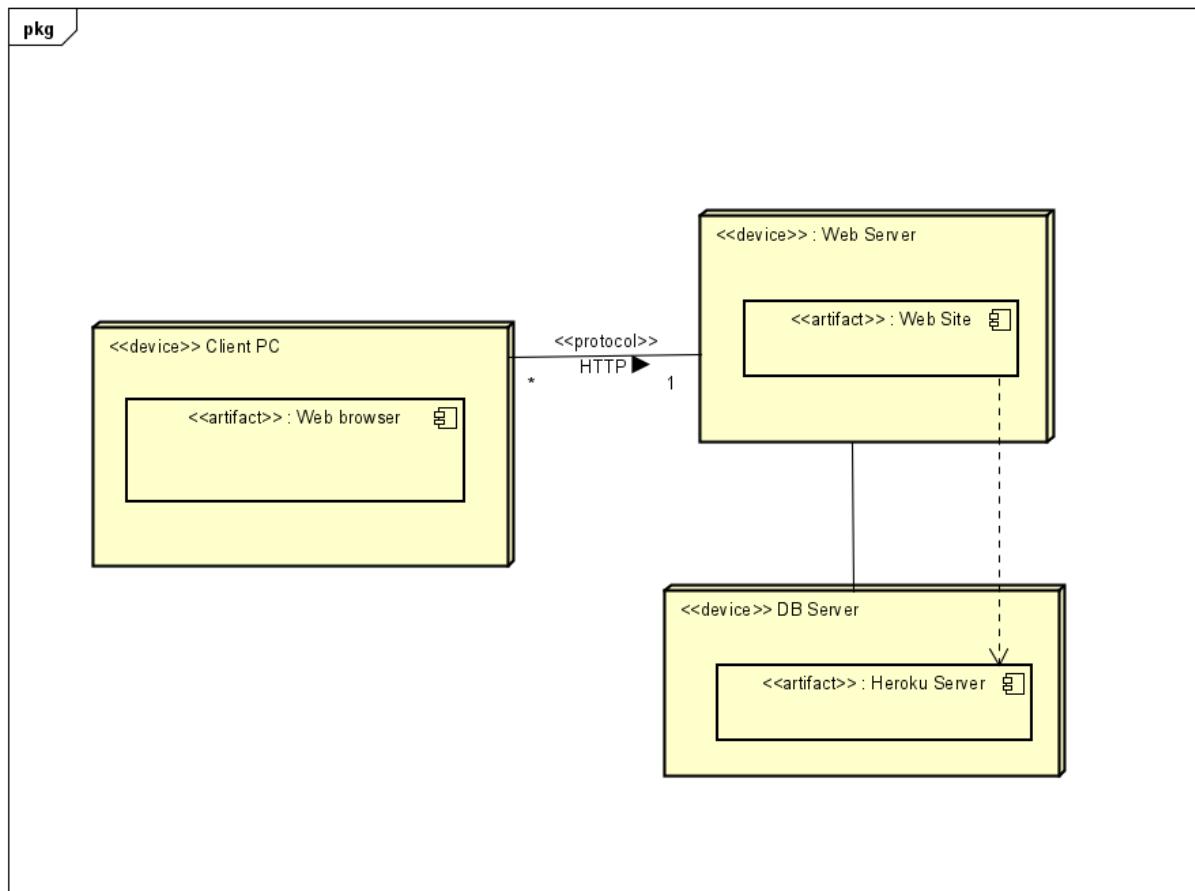
13. click on css=.round-button OK 21:12:26

**'dodaj\_aktivnost' completed successfully** 21:12:27

Test je prošao i dodane su grupe.

### 5.3 Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju programske potpore kako bi elementi sustava bili optimalno raspoređeni. Sustav je baziran na "klijent-poslužitelj" arhitekturi. Korisnička računala razmjenjuju podatke sa centraliziranim serverom, a Server sadrži sve potrebne datoteke i aplikacije koje šalje korisnicima na zahtjev. Komunikacija između korisnika i poslužitelja odvija se preko HTTPS veze.



## 5.4 Upute za puštanje u pogon

### Instalacija PostgreSQL sustava za upravljanje bazom podataka

Potrebno je preuzeti instalacijski program sa sljedeće poveznice <https://www.postgresql.org/download/>. Odabrat odgovarajući operacijski sustav i pritisnuti *Download installer*.

#### Konfiguracija i instalacija

1. Pokrenuti instalacijski program te zatim čarobnjak za konfiguraciju instalacije
2. Za direktorij je preporučeno ostaviti predloženu putanju
3. Kod odabira komponenti sustava koje će biti instalirane obavezno označiti komponente : **PostgreSQL Server** i **PgAdmin4**
4. Kod odabira direktorija za pohranu podataka, preporuka je ostaviti predloženu putanju
5. Unijeti lozinku za admin korisnika
6. Ostaviti predložena vrata (port) za pristup sustavu baze podataka : 5432
7. Odabrat lokalne postavke
8. Kliknuti Next, pričekati završetak instalacije te zatim kliknuti Finsih

#### Kreiranje baze podataka

Za interakciju s poslužiteljem PostgreSQL koristit ćemo *pgAdmin* program. Prvi prvom pokretanju traži da postavite "master" lozinku - upišite onu koju ste postavili u gornjim koracima. Nakon unosa lozinke otvoriti u prozoru "Browser" listu "Servers" i kliknite na znak strijelice pored "PostgreSQL 12". Ponovo unesite lozinku za korisnika *postgres*. Odabirom "Databases" u prozoru "Browser" desnim klikom odabrat "Create - Database". Unijeti ime baze podataka *projekt* i kliknuti na gumb Save.

### Instalacija alata Apache Maven

Za potrebe pokretanja naše aplikacije bitna stavka je instalacija alata Maven.

- Sa stranice <https://maven.apache.org/download.cgi> skinuti distribuciju archive u tar.gz ili zip formatu

- Provjeriti da je JAVA HOME varijabla pravilno podešena
- Ekstratirati distribuciju arhive na željenu lokaciju
- Dodati bin direktorij novostvorenog direktorija apache-maven-3.6.3 u PATH varijablu
- Provjeriti sa mvn -v uspješnost instalacije.

### Konfiguracija veze s bazom podataka

Da bi aplikacija uspješno koristila bazu podataka, potrebno je postaviti parametre konekcije s bazom podataka u datoteci **application.properties** u direktoriju **resources**. Potrebno je izmijeniti sljedeće linije:

spring.datasource.username=postgres

spring.datasource.password=*upisati lozinku odabrani u prethodnim koracima*

spring.datasource.url=jdbc:postgresql://localhost:5432/projekt

### Pokretanje web aplikacije

U terminalu se pozicionirati unutar vršnog direktorija u kojem je projekt. Pokrenuti naredbu mvn spring-boot:run.

Web aplikacija je sada dostupna na adresi <http://localhost:8080/>

## 6. Zaključak i budući rad

Naša grupa je kao zadatak dobila razvoj web aplikacije za Kamp mlade nade. Ta aplikacija trebala bi služiti organizatorima kampa, sudionicima i animatorima kampa, u svrhe bolje organizacije. Dvije ključne stavke bile su raspodjela sudiонika i animatora po grupama te dodavanje aktivnosti u raspored koji je uvijek na raspolaganju svim korisnicima sustava. Dodatno, trebali smo omogućiti davanje povratnih ocjena iskustva, kako bi organizatori dobili povratne informacije o uspješnosti kampa. Nakon 17 tjedana uspjeli smo razviti traženu aplikaciju. Sam razvoj projekta odvijao se kroz 2 faze.

U prvoj fazi obavili smo međusobno upoznavanje članova tima, podjelu zadataka te smo krenuli s proučavanjem zadatka. Usporedno s proučavanjem zadatka krenuli smo i s izradom dokumentacije koja nam je kasnije dobro poslužila i u rješavanju implementacijskih problema. Iako smo se prvi put susreli sa obrascima uporabe, sekvencijskim dijagramima i dijagramima razreda, ubrzo smo shvatili njihovu važnost u dalnjem procesu razvoja. Nadalje, međusobno smo se podijelili u 3 tima: backend, frontend i dokumentacija, jer smo shvatili da će veliki naglasak biti upravo na izradi dokumentacije, pogotovo u prvoj fazi.

Nakon inicijalne podjele rada i tima te rješavanja, krenuli smo u drugu fazu projekta. Iako nešto kraća od prve, bila je puno intezivnija po pitanju rada svih članova. Neki članovi koji su imali više prijašnjeg iskustva u tehnologijama koje smo koristili pomogli su manje iskusnim članovima pa nismo morali trošiti previše vremena na učenje novih tehnologija. Osim realizacije rješenja, u drugoj fazi je bilo potrebno dokumentirati ostale UML dijagrame i izraditi popratnu dokumentaciju kako bi budući korisnici mogli lakše koristiti ili vršiti preinake na sustavu. Dobro izrađen kostur projekta uštedio nam je mnogo vremena prilikom izrade aplikacije te smo izbjegli moguće pogreške u izradi koje bi bile vremenski skupe za ispravljanje u daljnjoj fazi projekta.

Komunikacija među članovima grupe je bila putem Slacka gdje je svaka podgrupa imala svoji kanal u kojima je mogla raspravljati detalje vezane uz svoj dio implementacije. Sastanke smo održivali svaki tjedan, prvo uživo, a kasnije preko servisa MS Teams. Svaki član obavještavao je o svojem napretku, a dodatno smo se

i dogovarali koji su nam zadaci za sljedeći tjedan.

Moguće proširenje postojeće inačice sustava bila bi izrada mobilne aplikacije čime bi aplikacija bila dostupna većem broju korisnika i daljnji rad na UI/UX dizajnu čime bi korištenje aplikacije bilo još intuitivnije.

Sudjelovanje na ovakvom projektu bilo je vrijedno iskustvo svim članovima tima jer smo kroz intenzivnih nekoliko tjedana rada iskusili zajednički rad na istom projektu. Također, osjetili smo važnost dobre vremenske organiziranosti i koordiniranosti između članova tima. Zadovoljni smo postignutim bez obzira na moguć prostor za usavršavanje aplikacije što je posljedica neiskustva na takvim i sličnim projektima.

Sve su funkcionalnosti implementirane u ostvarenoj aplikaciji.

# Popis literature

1. Programsko inženjerstvo, FER, <https://www.fer.unizg.hr/predmet/proinzh>
2. Creately, <https://creately.com/>
3. Astah, <https://astah.net/>
4. The Unified Modeling Language, <https://www.uml-diagrams.org/>
5. Spring Boot, <https://spring.io/projects/spring-boot>
6. Java Server Pages, <https://www.tutorialspoint.com/jsp/index.htm>
7. PostgreSQL, <https://www.postgresql.org/>

# Indeks slika i dijagrama

3.1 Dijagram obrasca uporabe korisnik-klijent . . . . .	18
3.2 Dijagram obrasca uporabe animator-sudionik. . . . .	19
3.3 Dijagram obrasca uporabe organizatora. . . . .	20
3.4 Sekvencijski dijagram za UC2 . . . . .	21
3.5 Sekvencijski dijagram za UC5 . . . . .	22
3.6 Sekvencijski dijagram za UC13 . . . . .	23
3.7 Sekvencijski dijagram za UC15 . . . . .	24
4.1 ER dijagram baze podataka . . . . .	33
4.2 Relacijska shema baze podataka . . . . .	33
4.3 Controlleri . . . . .	34
4.4 DTO . . . . .	35
4.5 Dijagram razreda . . . . .	36
4.6 DIjagram stanja sudionika . . . . .	37
4.7 DIjagram stanja animatora . . . . .	38
4.8 DIjagram stanja organizatora . . . . .	38
4.9 Definiranje aktivnosti . . . . .	39
4.10 Ocjena aktivnosti . . . . .	40
4.11 Prihvaćanje i odbijanje prijava korisnika . . . . .	41
4.12 Rregistacija korisnika . . . . .	42
4.13 Stvaranje grupa . . . . .	43
4.14 Dijagram komponenti . . . . .	44

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- 20.listopada.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - inicijalni sastanak, upoznavanje s timom

### 2. sastanak

- 29.listopada.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - dogovor oko korištene tehnologije

### 3. sastanak

- 18.studenoga.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - početak rada na frontend-u
  - spajanje baze s backend-om

### 4. sastanak

- 7.prosinca.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - početak rada na backend-u

5. sastanak

- 27.listopada.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin
- Teme sastanka:
  - implementacija rasporeda u aplikaciju

6. sastanak

- 3.siječnja.2021.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - dogovor završnih funkcionalnosti

7. sastanak

- 5.siječnja.2021.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - demonstracija rada aplikacije
  - bilježenje funkcionalnosti koje ne rade

8. sastanak

- 11.siječnja.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - finalna raspodjela posla

9. sastanak

- 12.listopada.2020.
- Prisustvovali: Petar Cvitanović,Luka Škarica, Jakov Gracin, Fabijan Kozina, Antonio Vencl, Petar Jukić, Leo Li
- Teme sastanka:
  - provjera ispravnosti rada aplikacije

## Tablica aktivnosti

### Kontinuirano osvježavanje

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

	Antonio Vencl	Petar Cvitanović	Luka Škarica	Jakov Gracin	Fabijan Kozina	Petar Jukić	Leo Li
Upravljanje projektom	2	8	2	1	2	1	1
Opis projektnog zadatka	3	0	1	0	0	0	0
Funkcionalni zahtjevi	0	0	0	3	2	0	1
Opis pojedinih obrazaca	0	0	0	0	4	0	0
Dijagram obrazaca	3	0	0	0	3	1	0
Sekvencijski dijagrami	0	0	4	0	0	0	0
Opis ostalih zahtjeva	0	2	0	0	0	1	1
Arhitektura i dizajn sustava	1	1	0	0	0	3	0
Baza podataka	3	2	0	2	3	0	2
Dijagram razreda	0	0	0	0	0	0	5
Dijagram stanja	0	0	0	0	0	5	0
Dijagram aktivnosti	0	0	0	0	5	0	0
Dijagram komponenti	0	0	0	5	0	0	0
Korištene tehnologije i alati	1	1	1	1	0	1	3
Ispitivanje programskog rješenja	0	0	5	0	5	5	0
Dijagram razmještaja	0	0	1	1	5	0	0
Upute za puštanje u pogon	1	1	1	1	1	5	1
Dnevnik sastajanja	0	0	0	0	0	1	0
Zaključak i budući rad	0	0	0	0	1	1	0
Popis literature	0	0	0	0	0	0	2
<i>Izrada početne stranice</i>	5	0	0	0	0	0	0

	Antonio Vencl	Petar Cvitanović	Luka Škarica	Jakov Gracin	Fabijan Kozina	Petar Jukić	Leo Li
Izrada baze podataka	1	6	1	1	1	0	2
Spajanje s bazom podataka	2	4	1	1	0	0	0
back end	8	8	8	8	4	4	4

## Dijagrami pregleda promjena

