

# Programsko inženjerstvo

Ak. god. 2023./2024.

## ConnectiNET

Dokumentacija, Rev. 2

Grupa: *Eventio*

Voditelj: *Gašpar Haramija*

Datum predaje: *18. siječanj 2024.*

Nastavnik: *Nikolina Frid*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>6</b>
<b>3 Specifikacija programske potpore</b>	<b>11</b>
3.1 Funkcionalni zahtjevi . . . . .	11
3.1.1 Obrasci uporabe . . . . .	13
3.1.2 Sekvencijski dijagrami . . . . .	27
3.2 Ostali zahtjevi . . . . .	30
<b>4 Arhitektura i dizajn sustava</b>	<b>31</b>
4.1 Baza podataka . . . . .	32
4.1.1 Opis tablica . . . . .	33
4.1.2 Dijagram baze podataka . . . . .	37
4.2 Dijagram razreda . . . . .	38
4.3 Dijagram stanja . . . . .	45
4.4 Dijagram aktivnosti . . . . .	46
4.5 Dijagram komponenti . . . . .	48
<b>5 Implementacija i korisničko sučelje</b>	<b>49</b>
5.1 Korištene tehnologije i alati . . . . .	49
5.2 Ispitivanje programskog rješenja . . . . .	51
5.2.1 Ispitivanje komponenti . . . . .	51
5.2.2 Ispitivanje sustava . . . . .	51
5.3 Dijagram razmještaja . . . . .	52
5.4 Upute za puštanje u pogon . . . . .	53
<b>6 Zaključak i budući rad</b>	<b>54</b>
<b>Popis literature</b>	<b>55</b>
<b>Indeks slika i dijagrama</b>	<b>56</b>

**Dodatak: Prikaz aktivnosti grupe**

**57**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Gašpar Haramija	24.10.2023.
0.2	Ažuriranje dnevnika sastajanja.	Filip Buhiniček	26.10.2023.
0.3	Dodavanje općeg UC dijagrama te pojedinačnih UC dijagrama. Dodane reference.	Sandro Boka	30.10.2023.
0.4	Izrada i dopuna funkcionalnih zahtjeva. Dodane reference.	Mateo Grdić	31.10.2023.
0.5	Dodani tekstualni opisi obrazaca uporabe	Marko Sršić, Jakov Šarolić	31.10.2023.
0.6	Dopisan opis projekta te ažuriran dnevnik promjena dokumentacije	Filip Buhiniček, Gašpar Haramija	01.11.2023.
0.7	Sekvencijski dijagrami	Gašpar Haramija	02.11.2023.
0.8.1	Prva verzija dijagrama razreda	Sandro Boka, Marko Sršić	03.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

<b>Rev.</b>	<b>Opis promjene/dodatka</b>	<b>Autori</b>	<b>Datum</b>
0.8.2	Nastavak i ažuriranje dijagrama razreda	Sandro Boka, Marko Sršić	15.11.2023.
0.9	Ažuriranje dokumentacije, brisanje template dijelova	Marko Sršić	16.11.2023.
<b>1.0</b>	Verzija samo s bitnim dijelovima za 1. ciklus	Gašpar Haramija	17.11.2023.
1.1	Ažuriranje obrazaca uporabe prema uputama iz 1. revizije	Sandro Boka, Marko Sršić	9.12.2023.
1.2	Brisanje template dijelova	Marko Sršić	18.12.2023.
1.3	Dijagram stanja	Marko Sršić	19.12.2023.
1.4	Dijagram aktivnosti	Marko Sršić	20.12.2023.
1.5.1	Razni ispravci (funkcionalni zahtjevi, obrasci uporabe, arhitektura i dizajn sustava, itd.)	Marko Sršić	4.1.2024.
1.5.2	Ažuriranje dnevnika sastajanja, tablice aktivnosti	Marko Sršić	11.1.2024.
1.6	Dijagram komponenti, dijagram razmještaja	Gašpar Haramija	13.1.2024.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.7	Korištene tehnologije, ispitivanje komponenti/sustava, opći ispravci i ažuriranja	Gašpar Haramija, Marko Sršić	17.1.2024.
1.8	Upute za puštanje u pogon	Gašpar Haramija	17.1.2024.
1.9	Dijagrami pregleda promjena, zaključak i budući rad	Gašpar Haramija	17.1.2024.
2.0	Konačni tekst predloška dokumentacije	Gašpar Haramija, Marko Sršić	18.1.2024.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "Eventio", koja će korisnicima omogućiti stvaranje, oglašavanje i sudjelovanje na različitim događanjima. Ova inovativna web aplikacija će korisnicima omogućiti lakši pristup društvenim događanjima koja odgovaraju njihovim interesima te olakšati organizaciju i promociju istih.

Postoje mnoge društvene platforme koje već imaju razvijene mreže korisnika i nude vlastite implementacije sličnih rješenja za događanja. Međutim, ono što "Eventio" čini drugačijim jest njegov pristup. Dok druga rješenja ovise o korisnicima koji stvaraju različite grupe i objavljuju događanja na koje se korisnici prijavljuju ili postaju članovi tih grupa (primjer toga je Facebook), "Eventio" nudi alternativni pristup. Korisnici će moći vidjeti sve dostupne događaje i filtrirati ih prema vlastitim preferencama, kao što su vrsta događaja, lokacija, datum i druge karakteristike. Na taj način, korisnici će imati veće šanse saznati za događanja koja najbolje odgovaraju njihovim interesima, čime se rješava glavni problem: pronalaženje događanja koja odgovaraju njihovim željama i potrebama.

Ova aplikacija je namijenjena svima - bez obzira na dob, spol ili interes. Svatko tko želi sudjelovati na događanjima bilo kojeg opsega i pronaći svoju zajednicu s istim interesima može pronaći korist u "Eventio". Osim što korisnicima pruža bolji pristup događanjima, aplikacija također nudi organizatorima priliku da promoviraju svoje događaje i privuku veći broj sudionika.

Unatoč prednostima, postoji nekoliko problema koji će se morati rješavati kako bi se projekt uspješno realizirao i u konačnici našao među krajnjim korisnicima. Aplikacija ovisi o velikom broju korisnika i organizatora kako bi postala funkcionalna i privukla interes. Potrebno je privući dovoljan broj organizatora s raznovrsnim događanjima kako bi se zadovoljili interesi široke publike. Također, potrebno je privući i potaknuti korisnike da koriste aplikaciju i aktivno sudjeluju na događanjima. Konkurencija s postojećim sličnim rješenjima kao što su Facebook, Meetup i Eventbrite ima prednost zbog svojih već razvijenih korisničkih baza koje mogu koristiti za testiranje i lansiranje ovakvog proizvoda.

U ovom kontekstu, "Eventio" se postavlja kao platforma koja će pružiti jed-

nostavan i učinkovit način povezivanja organizatora i posjetitelja događanja te olakšati promociju i sudjelovanje u raznovrsnim događanjima. Kroz inovativan pristup i kontinuirane nadogradnje, "Eventio" ima potencijal promijeniti način na koji korisnici pronalaze i sudjeluju na događanjima u svom okruženju. Primarni cilj ovog zadatka je razviti Minimalan vitalni proizvod (Minimum viable product, MVP), rješenje koje će na najjednostavniji i pristupačan način rješavati problem pronalaženja i sudjelovanja na događanjima prema osobnim preferencama. MVP će osigurati osnovne funkcionalnosti aplikacije kako bi se zadovoljile osnovne potrebe korisnika i omogućila provedba testiranja i daljnjeg razvoja.

U sljedećem dijelu opisa projekta bit će detaljno razrađena problematika zadatka. Također, bit će opisani korisnički zahtjevi i predložena moguća rješenja za svaku komponentu aplikacije. Sve ovo ima za cilj definirati temelje i smjernice za daljnji razvoj "Eventio" aplikacije.

Sljedeći dio opisa projekta detaljnije će razmotriti problematiku projektnog zadatka, identificirati glavne aktore i dionike te opisati korisničke zahtjeve i funkcionalnosti aplikacije "Eventio".

Glavni aktori uključuju:

- *Administrator: Osoba s najvišim ovlastima u sustavu, odgovorna za upravljanje korisnicima, događajima, i financijskim aspektima*
- *Korisnik (Organizator): Osoba koja može stvarati i oglašavati svoje događaje, upravljati svojim profilom, i platiti mjesečnu članarinu za objavu plaćenih događaja*
- *Korisnik (Posjetitelj): Osoba koja može pretraživati i iskazivati zainteresiranost za događaje, ostavljati recenzije i upravljati svojim profilom.*
- *Banka i PayPal: Pruzatelji usluga plaćanja koji omogućuju korisnicima plaćanje mjesečne članarine i ulaznica za događaje.*
- *Baza podataka: Skladište podataka o korisnicima, događajima, recenzijama, i financijskim transakcijama*

U nastavku su detaljno razmotreni korisnički zahtjevi i funkcionalnosti aplikacije "Eventio".

Prilikom pokretanja aplikacije, korisniku će se prikazati jedna od dvije mogućnosti.

- *Prozor sa pozdravom: Korisnik je već stvorio profil i ranije se prijavio u sustav*
- *Prozor sa formom za prijavu te gumbom koji vodi na kreiranje novog računa: Ovisno o tome posjeduje li korisnik račun će ili ispuniti prijavu u sustav ili pak kliknuti na gumb za stvaranje računa.*



Kako bi korisnik mogao uspješno kreirati račun potrebni su sljedeći podaci:

- *Korisničko ime*
- *Lozinka*
- *E-mail adresa*

Korisnik tijekom registracije isto tako može birati koju ulogu želi imati unutar aplikacije. Uloge su posjetitelj i organizator. Ako je korisnik odabrao ulogu organizatora, tada još dodatno uz navedene podatke treba upisati i sljedeće:

- *Naziv organizacije*
- *Adresa*
- *Link/Url svoje stranice*

Nakon kreiranja samog računa kao jedna uloga, neće biti omogućeno mijenjanje te uloge. Za prijavu će biti potrebno unijeti korisničko ime i lozinku postojećeg računa. Za odjavu će korisnici imati gumb odjavi koji je potrebno pritisnuti kako bi se korisnik odjavio sa računa te se nakon toga vraća na početni zaslon sa formom za prijavu.

Nakon uspješne prijave/registracije kao organizator događaja, korisnik može krenuti sa oglašavanjem. Prilikom unosa novog događaja, organizator mora unijeti podatke kao što su:

- *Naziv događaja*
- *Vrsta događaja*
- *Lokacija događaja*
- *Opis lokacije*
- *Vrijeme početka*
- *Trajanje*
- *Cijena ulaznice*
- *Opis događaja*

Gore navedeni podaci su obavezni, te uz obavezne podatke organizator može dodati još slike ili video snimke u galeriju samog događaja. Kada završi sa kreiranjem događaja, njega je moguće vidjeti na profilu organizatora zajedno sa svim njegovim podacima te događajima koji su se zbili unatrag dvije godine ili drugim događajima koji su stvoreni i tek dolaze.

Neki organizatori će organizirati događaje s besplatnim ulazom, no neki će vjerojatno htjeti naplatiti ulaz na događaj. Za mogućnost objavljivanja budućih

događaja za koje se plaća ulaz organizator mora biti pretplatnik mjesečne članarine na web aplikaciji. Prilikom objave prvog takvog događaja organizator je obvezan platiti članarinu.

Članarina se može platiti na dva načina:

- *PayPal*
- *Bankovna kartica*

Članarina će se obnavljati na mjesečnoj bazi sve dokle organizator sadrži neki događaj za koji je potrebno plaćanje ulaza. Također, prilikom objave novog događaja od strane organizatora za koji se naplaćuje ulaz, a organizator je u tom trenutku pretplatnik, slobodan je objaviti događaj normalnim procesom bez dodatnih obaveza.

Nakon uspješne prijave/registracije kao posjetitelj, korisnik može krenuti tražiti događaje koji mu odgovaraju te iskazivati svoju zainteresiranost za sami dolazak na te događaje. Zainteresiranost se dijeli u tri skupine:

- *Sigurno dolazim*
- *Možda dolazim*
- *Ne dolazim*

Kada korisnik stisne na bilo koju grupu, na samom događaju će biti vidljiva zainteresiranost. Ako je odabrana grupa „Sigurno dolazim“ ili „Možda dolazim“, tada se na profilu posjetitelja pojavi taj događaj.

Posjetitelj isto tako može ostaviti recenziju na događaju koja će biti dodana u listu svih recenzija pod tim događajem. Kako bi ju uspješno dodao u listu recenzija, svaka recenzija mora sadržavati kratki opis recenzije te opcionalno može sadržavati ocjenu od 1 do 10.

Posjetitelji unutar aplikacije mogu filtrirati sve događaje. Mogu ih filtrirati po nazivu ili vrsti događaja, njihovom vremenu početka, po lokaciji na kojoj se događaj odvija te po samom trajanju događaja. Na taj način posjetitelj može naći one događaje koji mu najviše odgovaraju i koji su mu najpogodniji. Kako bi se najvećim fanovima olakšala potražnja događaja njihovih najdražih organizatora, svaki posjetitelj moći će pretraživati isto tako i po korisničkom imenu organizatora. Kao što je navedeno, na profilu organizatora navedeni su svi prethodni događaji te svi događaji u budućnosti koji se također mogu filtrirati.

Svaki korisnik će isto tako imati mogućnosti osvježavanja i uređivanja vlastitog profila. Posjetitelji će moći izmijeniti svoje podatke te uvijek mogu promijeniti

zainteresiranost za neki event. Osim toga, posjetitelji mogu odabrati postavke da im aplikacija automatski šalje obavijesti o najnovijim događanjima prema zadanim kriterijima: vrsta događanja i područje. Organizatori uz promjene vlastitih podataka mogu isto tako promijeniti podatke događaja koji će se tek dogoditi, a svi oni već prošli događaji neće se moći mijenjati od strane organizatora.

Kako bi posjetiteljima dodatno olakšali korištenje aplikacije, događaji na kojima će zainteresiranost biti „Sigurno dolazim“ ili „Možda dolazim“ poslati će e-mail poruku na posjetiteljev račun 24 sata prije početka događaja. Isto tako, kada dođe do neke promjene na događaju, a koje nisu došle sa strane posjetitelja, ponovno stiže notifikacija u obliku e-mail poruke o samoj promjeni.

Uz dvije uloge korisnika, organizator i posjetitelj, unutar sustava postoji i uloga administratora. Administrator se kao i ostali korisnici može prijaviti te odjaviti iz sustava te unutar sustava on je korisnik s najvišim ovlastima.

Neće imati mogućnosti iskazati zainteresiranost za neki event te neće moći stvarati događaje kao što to organizator može, no ima druge ovlasti. On može pristupiti bazi podataka te može pregledati profil bilo kojeg drugog korisnika. Ukoliko smatra potrebnim, on može mijenjati podatke o događajima te može brisati recenzije ukoliko ih smatra nevaljanim. Isto tako može brisati same događaje, ali i korisnike ako se ne pridržavaju pravila unutar aplikacije. Zajedno sa time, on je taj koji određuje kolika će biti cijena mjesečne članarine koju plaćaju organizatori koji posjeduju neki događaj za koji se plaća upad.

Za ovakav tip proizvoda može se stalnim nadogradnjama i poboljšanjima privući još i šira publika. Ubrzo bi se na sami prototip aplikacije mogli dodati i chatovi unutar svakog događaja kako bi ljudi mogli komunicirati i prije samog događaja te na taj način zapravo započeli sami događaj i prije njegovog stvarnog početka. Uz ovaj dodatak mogli bi se na vrh svakog pretraživanja dodati i oni događaji za koje smatramo da su najpogodniji posjetitelju ovisno o njegovim prošlim izborima događaja.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Administrator
2. Korisnik/Posjetitelj
3. Organizator
4. Baza podataka
5. PayPal
6. Banka

#### Aktori i njihovi funkcionalni zahtjevi:

1. Administrator (inicijator) može:

- (a) prijaviti se i odjaviti sa stranice
- (b) pregledavati događaje
  - i. filtrirati događaje po vremenskom razdoblju
- (c) upravljati korisnicima
  - i. urediti ili izbrisati korisničke profile
  - ii. urediti ili izbrisati objave korisnika
- (d) postavljati cijene članstva

2. Korisnik/Posjetitelj (inicijator) može:

- (a) prijaviti se i odjaviti sa stranice
- (b) napraviti profil
- (c) prikazati profil i urediti ga
- (d) pregledavati događaje
  - i. filtrirati događaje po vremenskom razdoblju i ostalim parametrima
- (e) uključiti obavijesti o najnovijim događajima po kriterijima: vrsta događanja i područje

(f) iskazati interese za događaj („sigurno dolazim“, „možda dolazim“, „ne dolazim“)

(g) recenzirati događaj

3. Organizator (inicijator) može:

(a) postaviti događaj

i. urediti ili izbrisati događaj

ii. postaviti događaj sa plaćanjem ulaznice i platiti članarinu (Paypalom ili bankovnom karticom)

(b) prijaviti se i odjaviti sa stranice

(c) napraviti profil

(d) prikazati profil i urediti ga

(e) pregledavati događaje

i. filtrirati događaje po vremenskom razdoblju i ostalim parametrima

4. Baza podataka (sudionik):

(a) pohranjuje zapise o korisnicima i događajima

(b) pohranjuje iskazane interese za događaje

(c) pohranjuje recenzije događaja

(d) pohranjuje informacije o cijenama članstva

(e) omogućuje prijavu i odjavu sa stranice (token)

5. PayPal (sudionik):

(a) Odobrava transakcije potrebne za platiti članarinu

6. Banka (sudionik):

(a) Odobrava transakcije potrebne za platiti članarinu

### 3.1.1 Obrasci uporabe

#### UC1 - Registracija posjetiteljskog računa

- **Glavni sudionik:** Korisnik
- **Cilj:** Izrada korisničkog računa za potrebe pregledavanja i posjećivanja događaja
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik izabire opciju registracije
  2. Korisnik unosi korisničke podatke koji se od njega traže
  3. Sustav provjerava ispravnost unesenih podataka
  4. Sustav šalje korisnika na prozor za prijavu na stranicu
- **Opis mogućih odstupanja:**
  - 2.a Korisnik odabire već zauzeto korisničko ime ili e-mail
    1. Sustav obavještava korisnika o neuspješnoj registraciji i navodi razlog zašto registracija nije uspjela
    2. Korisnik mijenja potrebne podatke, završi registraciju ili odustane od registracije

#### UC2 - Registracija organizatorskog računa

- **Glavni sudionik:** Korisnik
- **Cilj:** Izrada korisničkog računa za potrebe stvaranja/organiziranja događaja
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik izabire opciju registracije uz odabir polja "Organizator"
  2. Korisnik unosi korisničke podatke koji se od njega traže
  3. Sustav provjerava ispravnost unesenih podataka
  4. Sustav šalje korisnika na prozor za prijavu na stranicu
- **Opis mogućih odstupanja:**
  - 2.a Korisnik odabire već zauzeto korisničko ime ili e-mail
    1. Sustav obavještava korisnika o neuspješnoj registraciji i navodi razlog zašto registracija nije uspjela
    2. Korisnik mijenja potrebne podatke, završi registraciju ili odustane od registracije

#### UC3 - Prijava na stranicu

- **Glavni sudionik:** Korisnik, Administrator
- **Cilj:** Dobiti pristup stranici
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik ima registrirani račun ili ima račun s pravima administratora
- **Opis osnovnog tijeka:**
  1. Korisnik/Administrator otvara stranicu
  2. Korisnik/Administrator unosi korisničke podatke za prijavu na stranicu
  3. Sustav prepoznaje je li prijavljeni račun korisnički ili administratorski te otvara odgovarajuću početnu stranicu
- **Opis mogućih odstupanja:**
  - 2.a Korisnik unosi pogrešnu kombinaciju imena i lozinke za prijavu
    1. Sustav javlja korisniku da je prijava neuspješna, nudi ponovnu prijavu ili stvaranje novog korisničkog računa

#### UC4 - Pregled događaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati dostupne događaje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik se prijavio na stranicu
- **Opis osnovnog tijeka:**
  1. Nakon prijave korisnika se prebacuje na stranicu za pregledavanje događaja - "Svi događaji"
  2. Korisniku su događaji izlistani jedan ispod drugog
  3. Kod svakog događaja prikazane su sve informacije o događaju (npr. datum održavanja, broj zainteresiranih itd.) te pripadajuća fotografija

#### UC5 - Filtriranje događaja

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati događaje prema željenim kriterijima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i nalazi se na stranici za pregled događaja
- **Opis osnovnog tijeka:**
  1. Korisnik na stranici za pregledavanje događaja bira između više ponuđenih opcija za filtriranje događaja
- **Opis mogućih odstupanja:**

1.a Ne postoji događaj koji odgovara korisnikovoj kombinaciji filtera

1. Sustav korisniku prikazuje stranicu bez događaja

#### **UC6 - Prikazivanje osobnog profila**

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osobni profil
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen na stranici
- **Opis osnovnog tijeka:**
  1. Korisnik klikne na ime svog računa te odabire opciju "Profil"
  2. Sustav prikazuje stranicu korisnikovog osobnog profila

#### **UC7 - Uređivanje profila**

- **Glavni sudionik:** Korisnik
- **Cilj:** Izmjena osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen na stranici i odabrao je opciju "Profil" (UC6)
- **Opis osnovnog tijeka:**
  1. Korisnik na stranici osobnog profila pored podatka koji želi promijeniti pritisne ikonu olovke
  2. Sustav omogućuje izmjenu podatka
  3. Korisnik mijenja podatak
  4. Korisnik odabire ikonu kvačice
  5. Korisnik odabire opciju "SPREMI"
  6. Sustav korisniku javlja da su promjene uspješno pohranjene
- **Opis mogućih odstupanja:**
  - 4.a Korisnik je promijenio korisničko ime ili e-mail u nešto što već postoji u sustavu
    1. Sustav javlja korisniku da je došlo do pogreške
    2. Korisnik ponovno upisuje ime/e-mail ili odustaje od promjena te ih briše pritiskom na ikonu križića

#### **UC8 - Odjava sa stranice**

- **Glavni sudionik:** Korisnik, Administrator
- **Cilj:** Odjaviti se sa stranice
- **Sudionici:** Baza podataka



- **Preduvjet:** Korisnik/Administrator je prijavljen na stranici
- **Opis osnovnog tijeka:**
  1. Korisnik/Administrator nakon klika na korisničko ime odabire opciju "Odjava"
  2. Sustav odjavljuje korisnika/administratora sa stranice

### UC9 - Izražavanje interesa

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Izraziti interes za neki ponuđeni događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen na stranici i njegov račun je posjetiteljski
- **Opis osnovnog tijeka:**
  1. Posjetitelj pregledava događaje (UC4) te dolazi do događaja koji ga zanima
  2. Posjetitelj odabire jednu od tri ponuđene opcije interesa: "DOLAZIM", "MOŽDA", "NE DOLAZIM"
  3. Sustav pohranjuje podatak o posjetiteljevom interesu, a u slučaju da je posjetitelj odabrao "DOLAZIM" ili "MOŽDA", ažurira odgovarajući brojač kod prikaza događaja te dodaje taj događaj na popis posjetiteljevih interesnih događaja
- **Opis mogućih odstupanja:**
  - 2.a Posjetitelj odabrao pogrešnu opciju ili želi promijeniti interes
    1. Posjetitelj odabire neku drugu ponuđenu opciju

### UC10 - Pregled posjetiteljevih interesnih događaja

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Pregledati sve događaje koje posjetitelja zanimaju
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen i na početnoj je stranici
- **Opis osnovnog tijeka:**
  1. Posjetitelj na početnoj stranici odabire opciju "MOJI DOGAĐAJI"
  2. Sustav posjetitelju otvara stranicu na kojoj je popis svih događaja za koje je postavio interes "DOLAZIM" ili "MOŽDA"
- **Opis mogućih odstupanja:**
  - 2.a Posjetitelj nije odabrao odgovarajuću opciju niti za jedan događaj
    1. Sustav korisniku prikazuje stranicu bez događaja

### UC11 - Recenziranje događaja

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Ostaviti recenziju na posjećen događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen na stranicu i nalazi se na stranici pregleda interesnih događaja (UC10)
- **Opis osnovnog tijeka:**
  1. Posjetitelj dolazi do događaja kojeg želi recenzirati
  2. Ako je događaj završio i nije prošlo 48 sati od njegovog završetka, posjetitelju je otvoren obrazac za recenziranje u kojem može odabrati ocjenu te napisati recenziju
  3. Posjetitelj završava recenziju i odabire opciju "Pohrani recenziju"
  4. Sustav pohranjuje recenziju
  5. Ako događaj nije završio, ili je prošlo više od 48 sati od njegovog završetka, posjetitelju nije otvoren obrazac za recenziranje
- **Opis mogućih odstupanja:**
  - 2.a Posjetitelj napisao opis recenzije, no nije izabrao ocjenu
    1. Sustav ne dozvoljava pohranjivanje recenzije dok ocjena nije odabrana
    2. Posjetitelj odabire ocjenu i pohranjuje recenziju, ili odustaje

### UC12 - Automatsko slanje obavijesti

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Kroz aplikaciju postaviti automatsko primanje obavijesti za najnovije događaje na temelju zadanih kriterija
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj prijavljen i nalazi se na stranici osobnog korisničkog računa
- **Opis osnovnog tijeka:**
  1. Korisnik ispod naznake "Pretplate na obavijesti" zadaje kriterij notifikacija - vrstu događaja te lokaciju
  2. Korisnik sprema pretplatu na obavijesti pritiskom na ikonu plusa te sustav pohranjuje korisnikov odabir
- **Opis mogućih odstupanja:**
  - 2.a Korisnik odabrao pretplatu na obavijesti, ali se predomislio i ne želi je

1. Sustav omogućuje korisniku brisanje pretplate klikom na ikonu koša za smeće

### UC13 - Postavljanje događaja

- **Glavni sudionik:** Organizator
- **Cilj:** Postaviti novi događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen na stranici i ima organizatorski račun
- **Opis osnovnog tijeka:**
  1. Organizator započinje proces izrade novog događaja pritiskom na ikonu plusa u desnom donjem kutu
  2. Organizatoru se otvara obrazac za dodavanje događaja
  3. Organizator popunjava sve potrebne podatke o događaju te postavlja događaj odabirom opcije "DODAJ"
  4. Sustav pohranjuje novi događaj te vraća organizatora na stranicu "Moji događaji"
- **Opis mogućih odstupanja:**
  - 4.a Organizator odustaje od objave događaja
    1. Organizator ima opciju "Odustani" čijim odabirom prekida proces stvaranja novog događaja

### UC14 - Postavljanje događaja sa plaćanjem ulaznice

- **Glavni sudionik:** Organizator
- **Cilj:** Postaviti događaj na kojem će biti naplaćivanje ulaznica
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen na stranicu, ima organizatorski korisnički račun i plaćenu pretplatu
- **Opis osnovnog tijeka:**
  1. Organizator pri ispunjavanju podataka o događaju kojeg želi postaviti stavlja željenu cijenu događaja (veću od 0)
  2. Organizator odabire opciju "DODAJ", sustav pohranjuje novi događaj te vraća organizatora na stranicu "Moji događaji"
- **Opis mogućih odstupanja:**
  - 2.a Organizator nema plaćenu članarinu
    1. Sustav javlja organizatoru kako nema plaćenu članarinu i ne objavljuje događaj

### UC15 - Plaćanje članarine

- **Glavni sudionik:** Organizator
- **Cilj:** Platiti članarinu kako bi se mogao postaviti događaj na kojem će se naplaćivati ulaznice
- **Sudionici:** Baza podataka, PayPal, Banka
- **Preduvjet:** Organizator se nalazi na stranici osobnog korisničkog računa
- **Opis osnovnog tijeka:**
  1. Organizator odabire opciju "PRETPLATA"
  2. Organizator odabire način plaćanja članarine - plaćanje PayPal-om ili plaćanje bankovnom karticom, ispunjava podatke i odabire opciju "Pretplati se"
  3. Sustav javlja organizatoru da je plaćanje članarine uspješno
- **Opis mogućih odstupanja:**
  - 2.a Transakcija je neuspješna
    1. Sustav javlja organizatoru kako transakcija nije prošla
    2. Organizator ponavlja postupak ili odustaje od plaćanja odabirom opcije "Povratak na profil"

### UC16 - Pregled vlastitih događaja

- **Glavni sudionik:** Organizator
- **Cilj:** Pregledati vlastite postavljene događaje
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator prijavljen na stranicu
- **Opis osnovnog tijeka:**
  1. Organizator na početnoj stranici odabire opciju "MOJI DOGAĐAJI"
  2. Sustav otvara organizatoru stranicu na kojoj prikazuje sve njegove postavljene događaje
- **Opis mogućih odstupanja:**
  - 2.a Organizator nema niti jedan postavljen događaj
    1. Sustav korisniku prikazuje stranicu bez događaja

### UC17 - Uređivanje/brisanje događaja

- **Glavni sudionik:** Organizator
- **Cilj:** Urediti ili obrisati postavljeni događaj
- **Sudionici:** Baza podataka

- **Preduvjet:** Organizator prijavljen, nalazi se na stranici "MOJI DOGAĐAJI" te ima postavljeni događaj
- **Opis osnovnog tijeka:**
  1. Organizator pronalazi događaj kojeg želi urediti ili izbrisati i pritisne na ikonu olovke, odnosno koša za smeće
  2. Ako je izabrana ikona olovke, sustav otvara organizatoru obrazac za postavljanje događaja ispunjen s njegovim podacima i dozvoljava uređivanje; organizator zatim uređuje podatke i bira opciju "Spremi"
  3. Ako je izabrana ikona koša za smeće, sustav briše događaj iz baze podataka
- **Opis mogućih odstupanja:**
  - 2.a Organizator greškom unio krive podatke
    1. Sustav dozvoljava izmjenu podataka neograničen broj puta

#### UC18 - Administrativni pregled događaja

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati događaje kao administrator
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran, ima prava administratora i uspješno se prijavio
- **Opis osnovnog tijeka:**
  1. Administrator na početnoj stranici odabire opciju "SVI DOGAĐAJI"
  2. Sustav administratora prebacuje na stranicu s popisom događaja

#### UC19 - Postavljanje cijene pretplate

- **Glavni sudionik:** Administrator
- **Cilj:** Postaviti ili promijeniti cijenu pretplate
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran, ima prava administratora i nalazi se na stranici osobnog profila
- **Opis osnovnog tijeka:**
  1. Administrator ispod naznake "Cijena pretplate" ažurira cijenu pretplate i odabire opciju "Spremi"

#### UC20 - Upravljanje korisnicima

- **Glavni sudionik:** Administrator

- **Cilj:** Pregledati korisničke račune i/ili njihove objave
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator na početnoj stranici odabire opciju "SVI KORISNICI"
  2. Sustav administratora prebacuje na stranicu s popisom korisnika i njihovim podacima

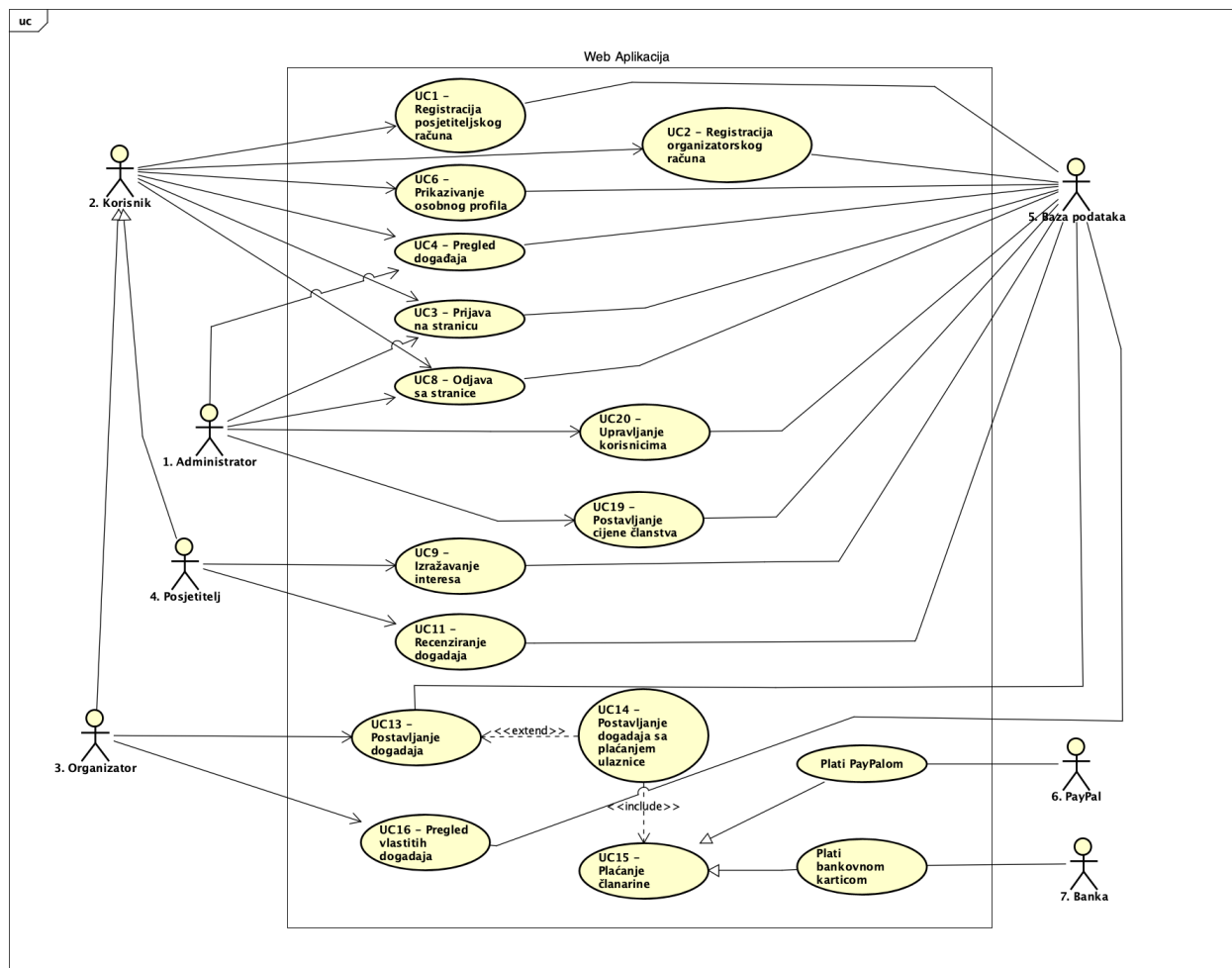
#### UC21 - Uređivanje/brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Urediti ili obrisati korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i ima prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator pronalazi željenog korisnika
  2. Administrator odabire ikonu koša za smeće te briše njegove podatke iz baze podataka

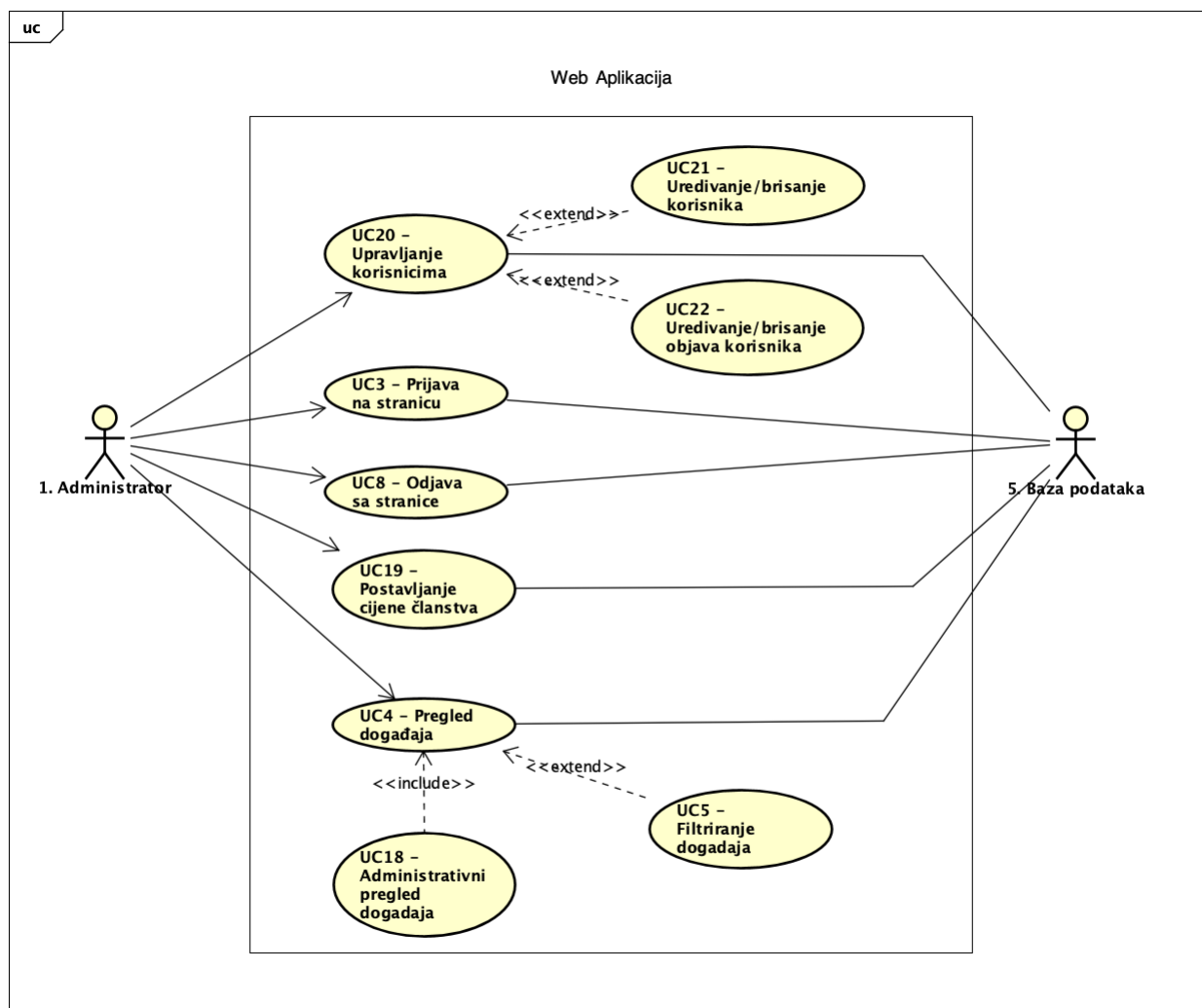
#### UC22 - Uređivanje/brisanje objava korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Urediti/obrisati objave korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran, ima prava administratora i pregledava sve događaje
- **Opis osnovnog tijeka:**
  1. Administrator pronalazi događaj kojeg želi urediti ili izbrisati te pritisne opciju na ikonu olovke ili ikonu koša za smeće
  2. Ako je odabrana ikona olovke, sustav otvara obrazac za postavljanje događaja ispunjen s njegovim podacima i dozvoljava uređivanje; administrator zatim uređuje podatke i bira opciju "Spremi"
  3. Ako je izabrana ikona koša za smeće, sustav briše događaj iz baze podataka

## Dijagrami obrazaca uporabe

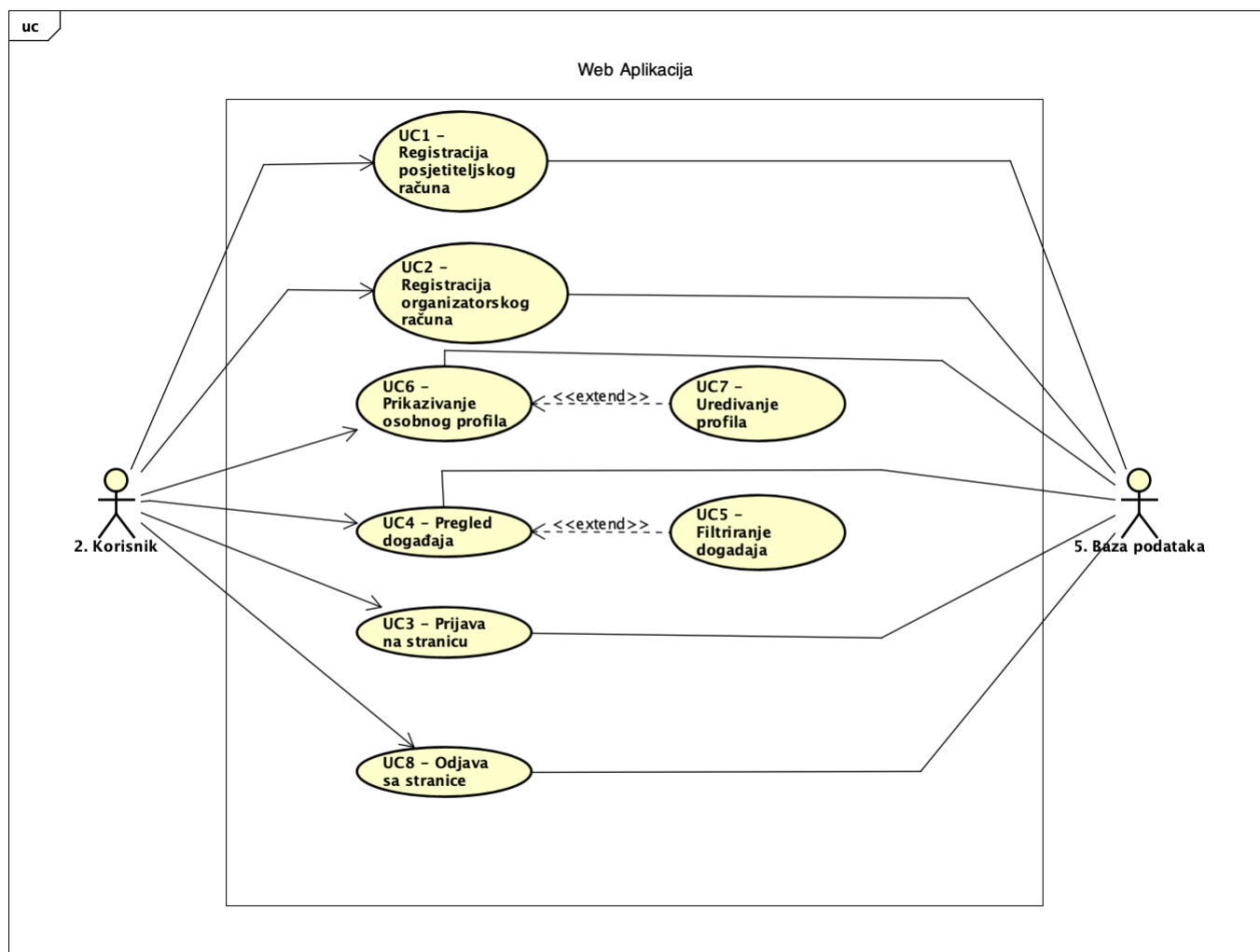


Slika 3.1: Dijagram obrasca uporabe, opća funkcionalnost aktora

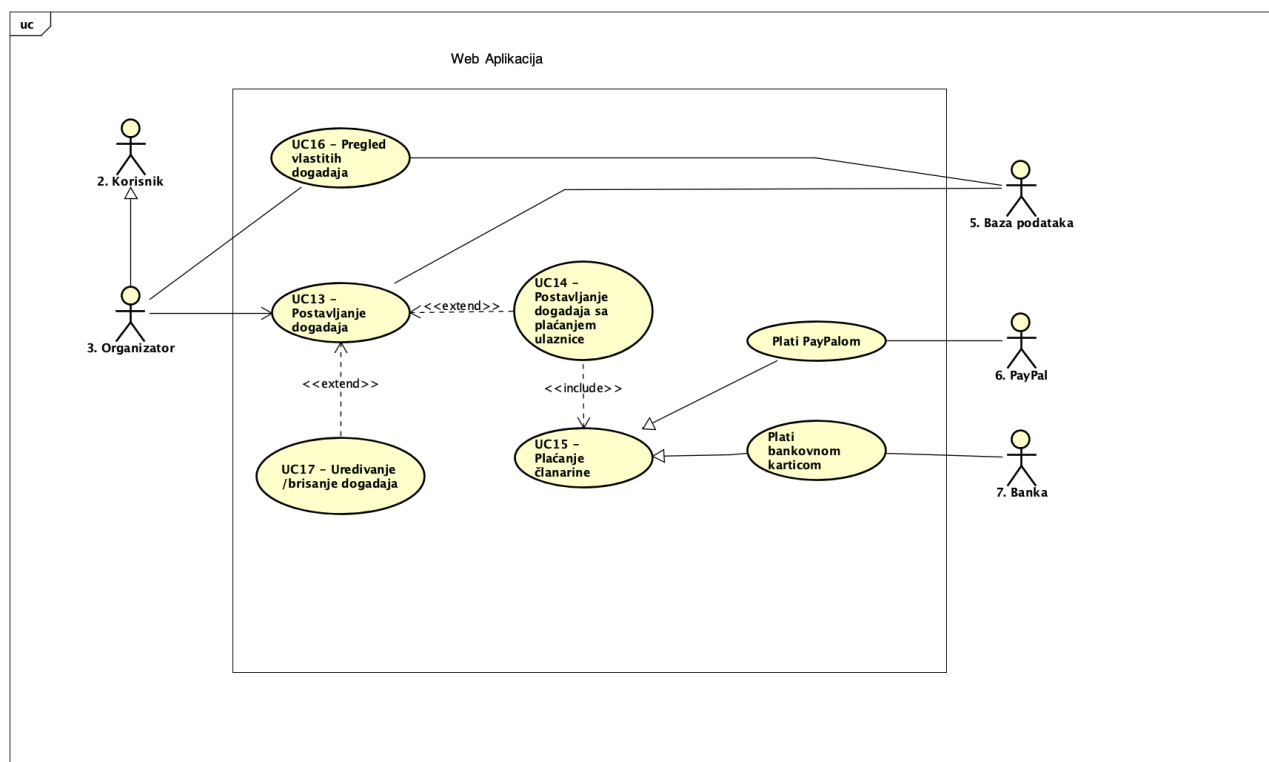


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost administratora

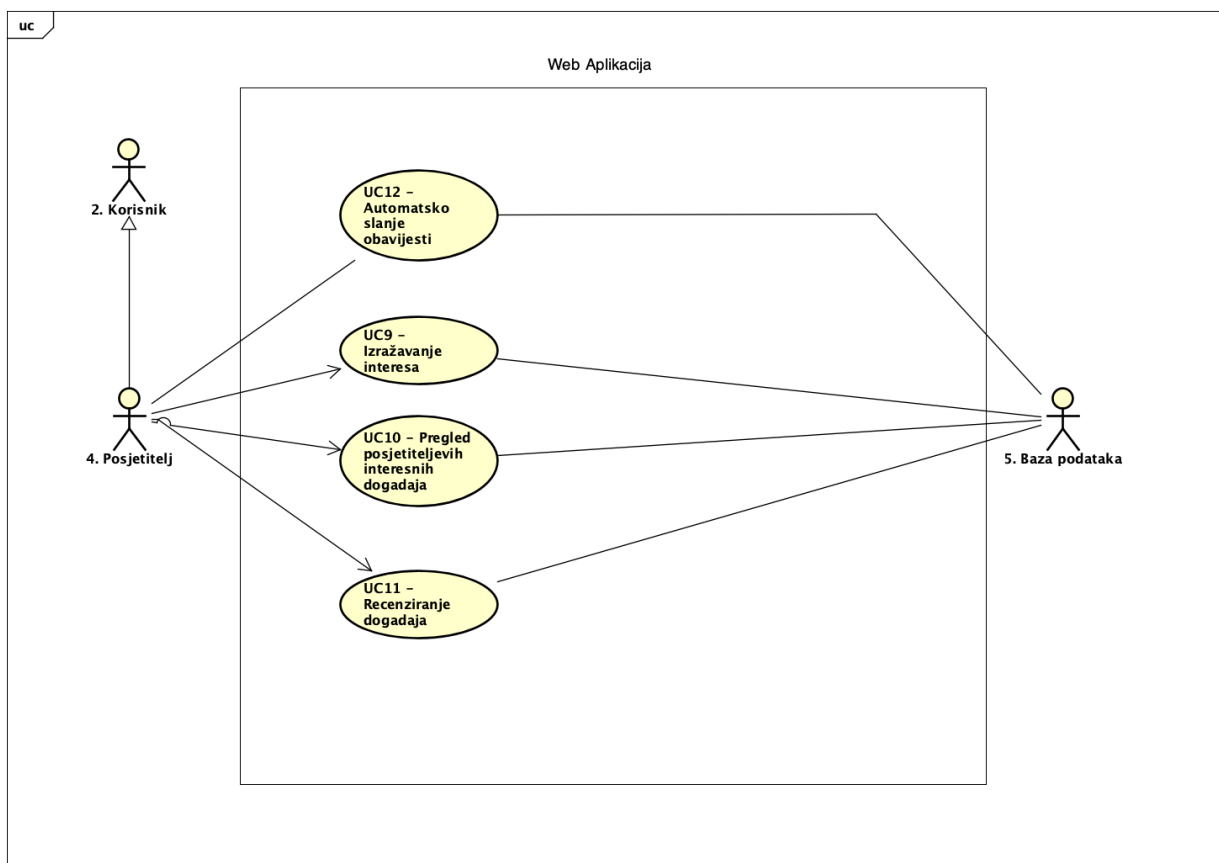




Slika 3.3: Dijagram obrasca uporabe, funkcionalnost korisnika



Slika 3.4: Dijagram obrasca uporabe, funkcionalnost organizatora



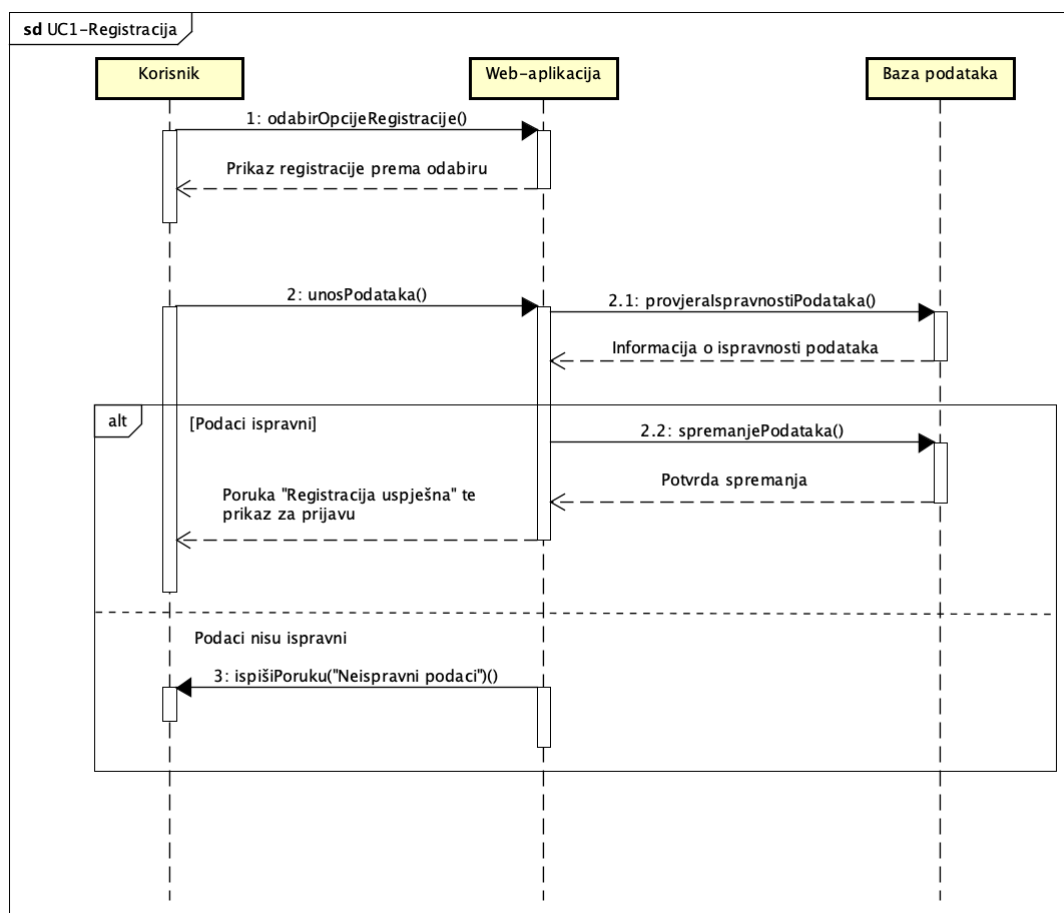
Slika 3.5: Dijagram obrasca uporabe, funkcionalnost posjetitelja

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1 - Registracija

Korisnik prilikom procesa registracije odabire opciju registracije: Posjetitelj ili Organizator. Na temelju odabira, korisnik dohvaća obrazac za registraciju. Nakon ispunjavanja obrasca, korisnik šalje zahtjev za registraciju. Nakon provjere ispravnosti unesenih podataka moguće su dvije opcije:

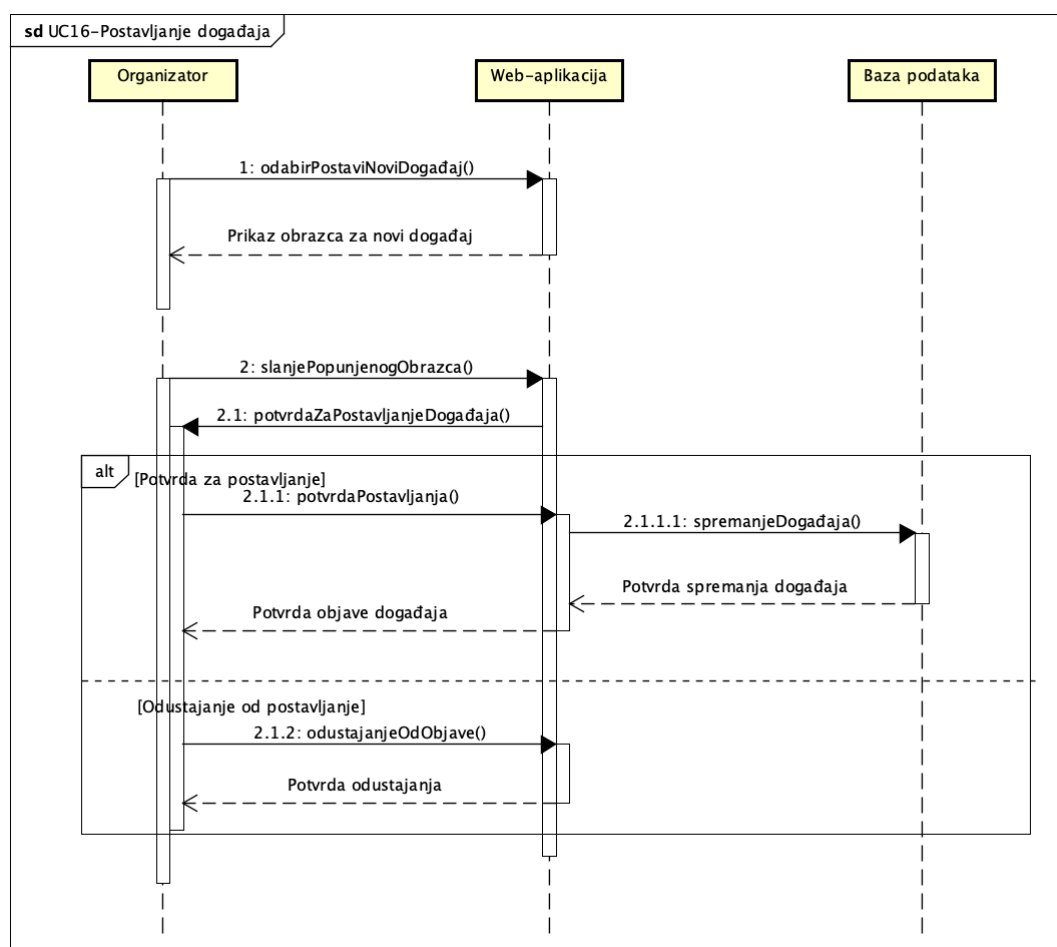
- Podaci ispravni: u slučaju ispravnih podataka vrši se spremanje korisničkih podataka te korisnik prima poruku "Registracija uspješna" te slijedi prikaz za prijavu
- Podaci neispravni: Korisnik prima poruku da su podaci neispravni te može unijeti podatke ispočetka



Slika 3.6: Sekvencijski dijagram za UC1

## Obrazac uporabe UC16 - Postavljanje događaja

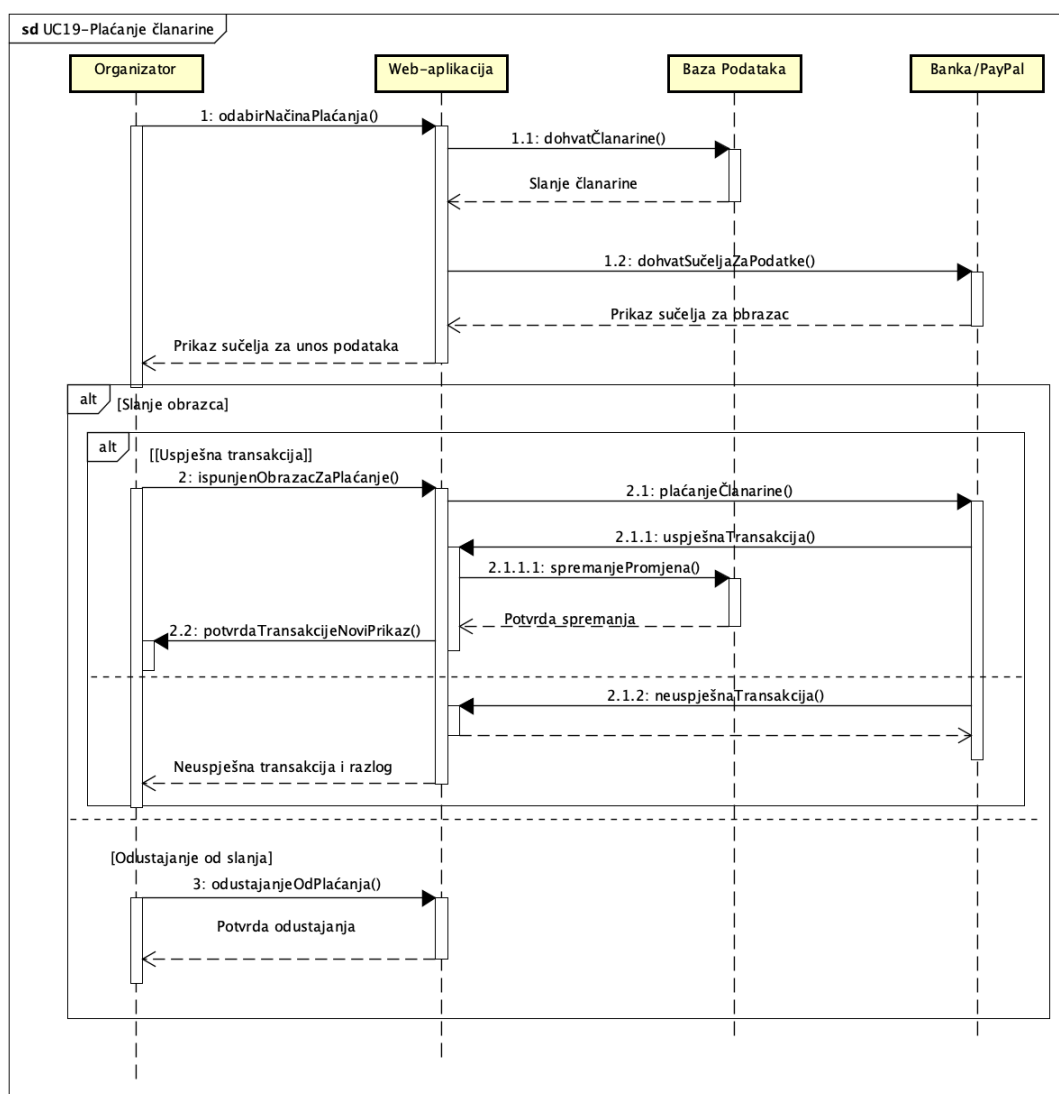
Tijekom procesa postavljanja događaja Organizator ima mogućnost izabrati objavu bez plaćanje ulaznice za događaj ili uz plaćanje. U slučaju postavljanja događaja bez plaćanja ulaznice odmah dohvaća obrazac za novi događaj. Nakon slanja popunjenog obrasca sa svim potrebnim informacija, Organizator zaprima potvrdu za postavljanje događaja te ima mogućnost potvrditi ili odustati od objave. Ukoliko dođe do potvrde, informacije o događaju se spremaju te se isti objavljuje, u suprotnom Organizator dobiva potvrdu odustajanja.



Slika 3.7: Sekvencijski dijagram za UC16

## Obrazac uporabe UC19 - Plaćanje članarine

Tijekom procesa objave događaja moguće je izabrati opciju za plaćanje ulaznice te u tom slučaju prije slanja obrasca za postavljanje novog događaja dolazi do plaćanje članarine ukoliko je potrebno. Organizator odabire način plaćanje te mu se prikazuje sučelje za unos podataka od izabranog izvora sa trenutnom cijenom članarine. U ovom trenutku Organizator može nastaviti sa procesom plaćanja ili odustati. U slučaju odustajanja dobiva potvrdu, a u suprotnom šalje ispunjen obrazac za plaćanje. Na temelju poslanog obrasca se izvršava plaćanje koje može biti uspješno ili neuspješno. Nakon uspješnog plaćanja se navedene promjene spremaju te Organizator dobiva potvrdu transakcije i novi prikaz. U slučaju neuspješne transakcije Organizator dobiva poruku i razlog neuspješne transakcije.



Slika 3.8: Sekvencijski dijagram za UC19

## 3.2 Ostali zahtjevi

- Sustav treba omogućiti brzo i učinkovito pregledavanje aktualnih događanja, uz minimalno vrijeme odziva aplikacije.
- Korisničko sučelje i sustav moraju podržavati različite vrste događanja i inicijativa, prilagođene različitim interesima zajednice.
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi kako bi se osigurala brza dostupnost informacija.
- Sustav treba biti implementiran kao web-aplikacija koristeći suvremene tehnologije i objektno-orijentirane jezike.
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava, a sustav treba pružiti jasne upute korisnicima o načinu korištenja.
- Sustav treba podržavati različite načine plaćanja članarine, uključujući PayPal i kreditne kartice, te osigurati sigurnost transakcija.
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške kako bi se očuvala integritet podataka.
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS kako bi se osigurala sigurna komunikacija između korisnika i sustava.
- Administratori sustava trebaju imati alate za postavljanje cijena članstva, upravljanje korisnicima i održavanje cjelokupnog sustava.

## 4. Arhitektura i dizajn sustava

Arhitektura našeg sustava temeljit će se na kombinaciji modernih tehnologija kako bismo ostvarili funkcionalan i skalabilan sustav. Sustav će se podijeliti na tri ključna podsustava:

- *Web poslužitelj*
- *Web klijent*
- *Baza podataka*



Slika 4.1: Arhitektura sustava

Organizacija će biti usklađena s MVC (Model-View-Controller) konceptom, što će omogućiti neovisnost između različitih dijelova sustava te olakšati razvoj, ispitivanje i dodavanje novih funkcionalnosti.

**Web preglednik**, kao program koji omogućuje korisnicima pregled web-stranica i multimedijalnih sadržaja, igra ključnu ulogu kao sučelje između web klijenta i web aplikacije. Klijenti, putem web preglednika, šalju zahtjeve web poslužitelju kako bi pristupili željenim resursima.

**Web poslužitelj** je osnova rada web aplikacije i odgovoran je za komunikaciju između klijenta i aplikacije. Komunikacija se odvija putem HTTP protokola, koji omogućuje prijenos informacija na webu. Poslužitelj pokreće web aplikaciju i prosljeđuje joj zahtjeve koje prima od klijenata.

**Web aplikacija**, smještena na poslužitelju, obrađuje zahtjeve korisnika. Ovisno o tim zahtjevima, pristupa poslužitelju baze podataka i na temelju dobivenih podataka, putem web poslužitelja, šalje odgovor korisnicima. Aplikacija se sastoji od backend i frontend dijela, koji se razvijaju koristeći različite tehnologije u skladu s



opisanom arhitekturom sustava. Konkretno, za backend dio koristit ćemo Spring, dok će za frontend biti korišten React, što je u skladu s MVC načelima.

Ključna karakteristika arhitekturnog obrasca MVC-a je nezavisan razvoj pojedinih dijelova aplikacije. Ova karakteristika rezultira jednostavnijim ispitivanjem, kao i olakšanim razvojem i dodavanjem novih svojstava u sustav.

MVC koncept sastoji se od tri osnovna dijela:

- **Model:** središnja komponenta sustava koja predstavlja dinamičke strukture podataka neovisne o korisničkom sučelju. Model upravlja podacima, logikom i pravilima aplikacije, primajući ulazne podatke od Controllera
- **View:** odgovara za prikaz podataka, poput grafičkih elemenata. Moguć je različiti prikaz istih informacija, kao što su grafički ili tablični prikazi podataka
- **Controller:** primarno prima ulazne podatke od korisnika i prilagođava ih za daljnju interakciju s Modelom ili Viewom. Kontrolira korisničke zahtjeve i izvodi daljnju interakciju s ostalim elementima sustava

Ovaj pristup omogućuje jasnu organizaciju sustava i olakšava daljnji razvoj i održavanje.

## 4.1 Baza podataka

Kao sustav za upravljanje bazama podataka za našu aplikaciju odabrali smo PostgreSQL. To je otvoren sustav za upravljanje bazama koji omogućuje pohranu, upravljanje i analizu podataka.

Sama baza podataka sadržavat će sedam tablica. One će služiti za pohranu podataka o korisnicima te njihovom korištenju aplikacije. Korištenje će se spremati u obliku događaja koje stvaraju ili posjećuju te recenzija koje pišu za određene događaje. Isto tako spremat će se sama zainteresiranost za događaje te notifikacije koje posjetitelji žele primati za događaje s određenim karakteristikama.

Sami odabir PostgreSQL-a je u njegovoj pouzdanosti, skalabilnosti i podrške za napredne SQL funkcionalnosti, što će nam omogućiti lakše te učinkovitije upravljanje podacima. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Organizator

- Događaj
- Recenzija
- Zainteresiranost
- Notifikacija
- Misc

#### 4.1.1 Opis tablica

**Korisnik** - ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: korisnikId koji je automatski dodijeljen, username, password, email i uloga u aplikaciji. Ovaj entitet u vezi je Many-to-One s entitetom organizator preko atributa organizatorId.

korisnik		
korisnikId	INT	Primarni ključ tablice, dodjeljuje se automatski
username	VARCHAR	Jedinstveno ime koje korisnik odabire tijekom registracije
password	VARCHAR	Lozinka koju korisnik odabire tijekom registracije
email	VARCHAR	Jedinstveni email korisnika
uloga	SMALL INT/INT	Uloga koju korisnik ima unutar aplikacije(admin, posjetitelj, organizator)

**Organizator** - Ovaj entitet sadržava informacije o organizatoru događaja. Sadrži attribute: organizatorId, nazivOrganizacije, adresa, poveznica, clanarina. Ovaj entitet u vezi je One-to-Many s entitetom događaj preko atributa organizatorId.

organizator		
organizatorId	INT	Primarni ključ tablice, ujedno i strani ključ iz tablice korisnik

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

organizator		
nazivOrganizacije	VARCHAR	Naziv organizacije kojoj pripada organizator
adresa	VARCHAR	Adresa organizacije
poveznica	VARCHAR	Poveznica na web stranicu organizacije
clanarina	BOOLEAN	Plaćena članarina

**Događaj** - ovaj entitet sadržava informacije o događaju. Sadrži attribute: događajId, organizatorId, nazivDogađaja, vrsta, lokacija, trajanje, vrijemePocetka, cijenaUlaznice, opis, galerija. Ovaj entitet u vezi je Many-to-One s entitetom Organizator preko atributa organizatorId te u vezi One-to-Many s entitetom recenzija preko atributa događajId.

događaj		
događajId	INT	Primarni ključ tablice, dodjeljuje se automatski
organizatorId	INT	Strani ključ iz tablice organizator koji se odnosi na njegov ID
nazivDogađaja	VARCHAR	Naziv događaja
vrsta	VARCHAR	Vrsta događaja(koncert, kazališna predstava ...)
lokacija	VARCHAR	Lokacija događaja
opisLokacije	VARCHAR	Kratki opis lokacije, adresa, kat ...
trajanje	INTERVAL	Trajanje događaja u danima
vrijemePocetka	TIMESTAMP	Datum i vrijeme početka događaja
cijenaUlaznice	NUMERIC	Cijena ulaznice na događaj
opis	VARCHAR	Kratki opis samog događaja
galerija	VARCHAR	Link do web stranice gdje se nalaze video snimke ili fotografije

**Recenzija** - ovaj entitet sadržava informacije o recenziji događaja. Sadrži atri-

bute: recenzijaId, korisnikId, događajId, tekst, ocjena. Ovaj entitet u vezi je Many-to-One s entitetom korisnik preko atributa korisnikId te u vezi Many-to-One s entitetom događaj preko atributa događajId.

recenzija		
recenzijaId	INT	Primarni ključ tablice, dodjeljuje se automatski
korisnikId	INT	Strani ključ iz tablice korisnik koji se odnosi na njegov ID
događajId	INT	Strani ključ iz tablice događaj koji se odnosi na njegov ID
tekst	VARCHAR	Kratki tekst recenzije
ocjena	SMALL INT/INT	Ocjena od 1-5, opcionalna

**Zainteresiranost** - ovaj entitet sadržava informacije o zainteresiranosti posjetitelja za određeni događaj. Sadrži attribute: zainteresiranostId, posjetiteljId, događajId, kategorija. Ovaj entitet u vezi je Many-to-One s entitetom korisnik preko atributa posjetiteljId te u vezi Many-to-One s entitetom događaj preko atributa događajId.

zainteresiranost		
zainteresiranostId	INT	Primarni ključ tablice, dodjeljuje se automatski
posjetiteljId	INT	Strani ključ iz tablice korisnik koji se odnosi na njegov ID
događajId	INT	Strani ključ iz tablice događaj koji se odnosi na njegov ID
kategorija	SMALL INT/INT	Jedna od tri kategorije: Sigurno dolazim, možda dolazim ili ne dolazim

**Notifikacija** - ovaj entitet sadržava informacije o notifikacijama koje će posjetitelji primati vezano uz tražene događaje filtrirane po vrsti ili lokaciji. Sadrži attribute: notifikacijaId, posjetiteljId, vrsta, lokacija. Ovaj entitet u vezi je Many-to-One

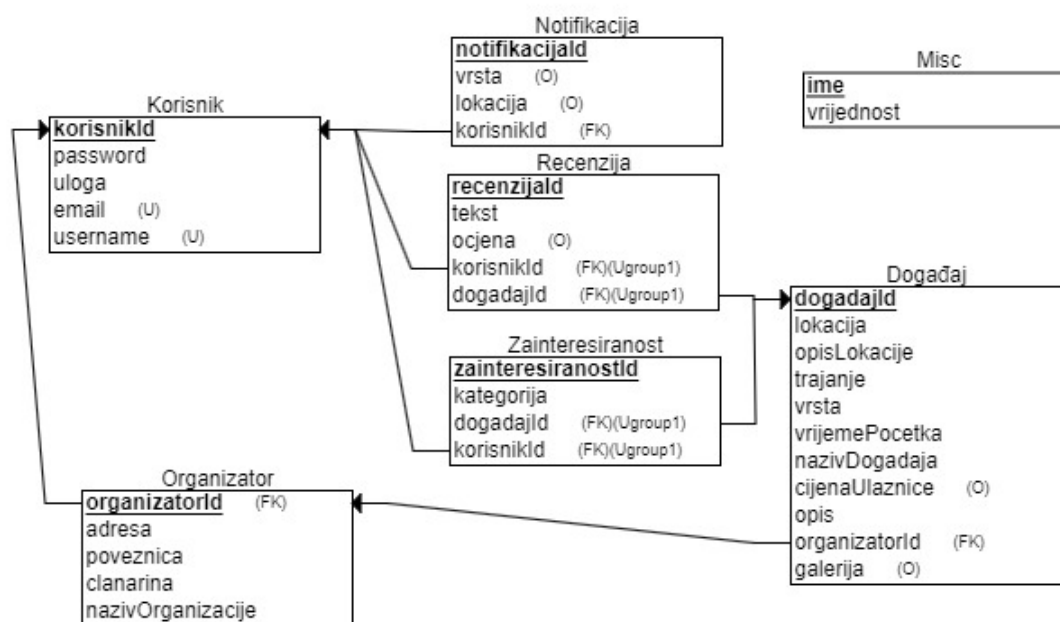
s entitetom korisnik preko atributa korisnikId te u vezi Many-to-One s entitetom događaj preko atributa vrsta i lokacija.

notifikacija		
notifikacijaId	INT	Primarni ključ tablice, dodjeljuje se automatski
posjetiteljId	INT	Strani ključ iz tablice korisnik koji se odnosi na njegov ID
vrsta	VARCHAR	Vrsta događaja za koji posjetitelj želi primiti obavijest o njegovu stvaranju, opcionalno
lokacija	VARCHAR	Lokacija događaja za koji posjetitelj želi primiti obavijest o njegovu stvaranju, opcionalno

**misc** - skraćeno od "miscellaneous", ovaj entitet služi za spremanje globalnih postavki aplikacija u obliku ključ-vrijednost. Sadrži attribute ime i vrijednost.

misc		
ime	VARCHAR	Ime (ključ) svojstva
vrijednost	VARCHAR	Vrijednost svojstva

### 4.1.2 Dijagram baze podataka



Slika 4.2: Dijagram baze podataka

## 4.2 Dijagram razreda

Na slikama 4.2, 4.3, 4.4 i 4.5 su prikazani razredi koji pripadaju *backend* dijelu projekta.

Razredi prikazani na slici 4.2 su razredi kontroleri. Ti razredi su ključni dijelovi koji upravljaju pristiglim zahtjevima, određuju rute te obrađuju logiku aplikacije. Obično se koriste za implementaciju HTTP metoda poput **GET**, **POST**, **PUT** i **DELETE** za obrađivanje zahtjeva koji dolaze od klijenata. Razredi koji imaju svoje kontrolere su **Korisnik**, **Organizator**, **Admin**, **Posjetitelj**, **Događaj**, **Notifikacija**, **Recenzija** i **Zainteresiranost**.

**KorisnikController** ima metode:

- `getAll` - vraća listu Korisnika
- `getKorisnikById` - ako postoji Korisnik sa tim id, njega vraća
- `getCurrentUser` - kao argument prima Korisnika, a vraća `KorisnikDTO`
- `register` - kao argument prima `KorisnikDTO`, a vraća `ResponseEntity` koji je `string`

Isto ima i člansku varijablu tipa `KorisnikService` preko koje se obavlja poslovna logika (operacije nad podacima, pristupanje bazi podataka ili komunikacija s vanjskim servisima).

**OrganizatorController** ima metode:

- `getDetails` - kao argument prima Korisnika, a vraća `OrganizatorDTO`
- `register` - kao argument prima `OrganizatorDTO`, a vraća `ResponseEntity` koji je `string`

Isto ima i članske varijable tipa `KorisnikService` i `OrganizatorService` preko koje se obavlja poslovna logika (operacije nad podacima, pristupanje bazi podataka ili komunikacija s vanjskim servisima).

Razredi prikazani na slici 4.3 su razredi servisi. Razredi servisi u Spring Bootu

predstavljaju komponente koje sadrže poslovnu logiku aplikacije, pristupaju podacima i obavljaju operacije koje nisu specifične za obradu HTTP zahtjeva. Oni često služe kao posrednici između kontrolera i sloja pristupa podacima (Repository sloj) te obavljaju ključne funkcije za obradu podataka, poslovnu logiku i vanjske integracije. Razredi koji imaju svoje servise su **Korisnik, Organizator, Admin, Posjetitelj, Događaj, Notifikacija, Recenzija i Zainteresiranost**.

**KorisnikService** ima metode:

- `listAll` - vraća listu Korisnika
- `findById` - prima id kao argument, a ako postoji Korisnik s tim id, njega vraća
- `findByUsername` - prima username kao argument, a ako postoji korisnik s tim usernamom njega vraća
- `findByEmail` - prima email kao argument, a ako postoji korisnik s tim emailom njega vraća
- `registerUser` - kao argument prima KorsnikDTO, a boolean ovisno o uspješnosti

Isto ima i člansku varijablu tipa `KorisnikRepository`. Razred repozitorija u Spring Bootu predstavlja sloj pristupa podacima (data access layer) i obično se koristi za komunikaciju s bazom podataka. Ovi repozitoriji pružaju apstrakciju za operacije nad podacima, kao što su dohvaćanje, spremanje, ažuriranje ili brisanje podataka.

**OrganizatorService** ima metode:

- `findById` - prima id kao argument, a ako postoji Organizator sa tim id, njega vraća
- `registerOrganizator` - kao argument prima OrganizatorDTO, a vraća boolean ovisno o uspješnosti



Razredi prikazani na slici 4.5. su razredi modeli. Struktura baze podataka u aplikaciji odražava se upravo model razredima; osim što imaju svoje pripadne atributa, razredi također implementiraju metode koje ostvaruju interakciju s bazom podataka.

Razred Korisnik reprezentira općeg korisnika aplikacije. Specificira ga atribut "uloga" koja određuje je li korisnik administrator, posjetitelj ili organizator.

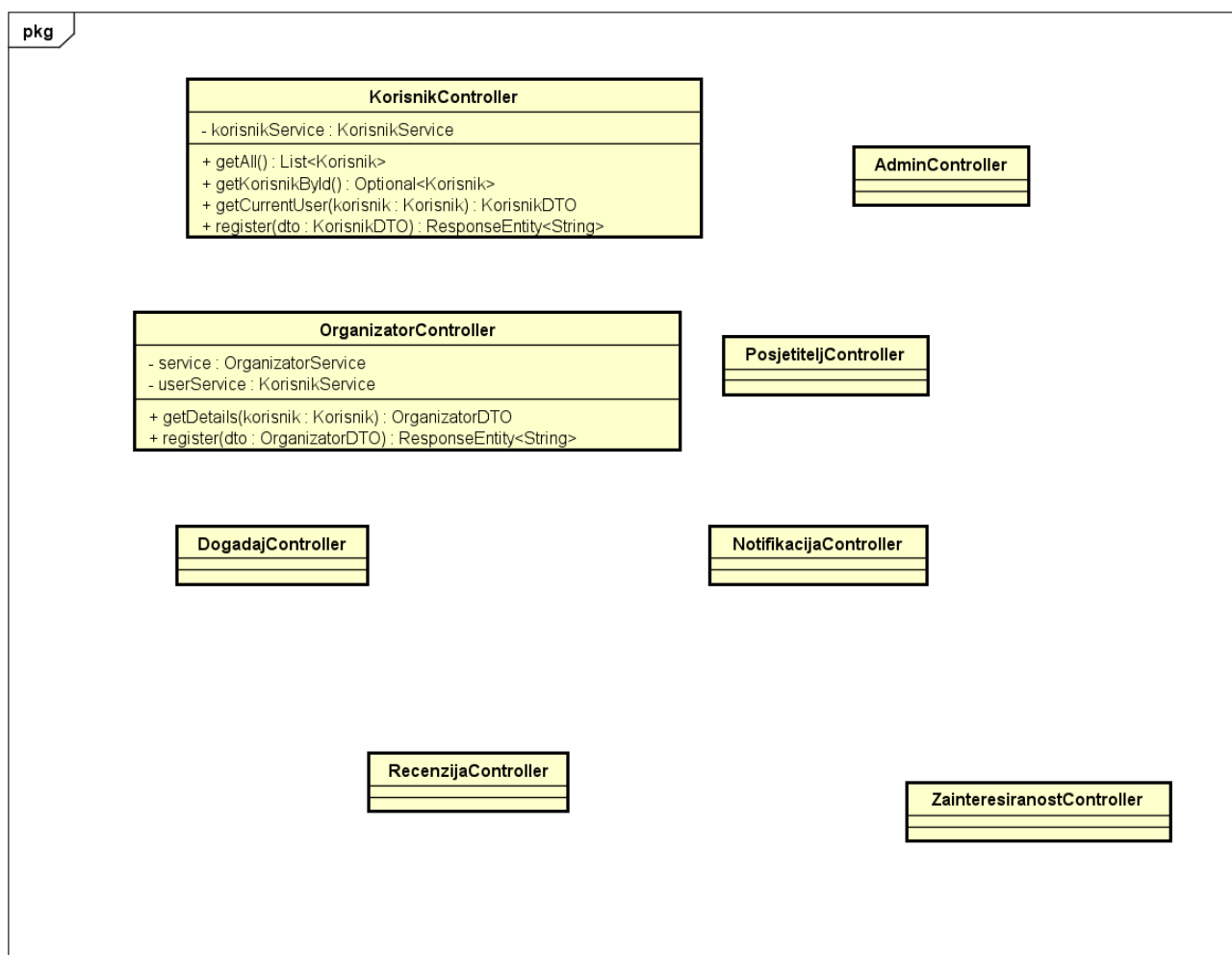
Razred Organizator reprezentira korisnika s organizatorskim računom. Organizatori mogu pregledavati svoje postojeće događaje i postavljati nove događaje.

Razred Recenzija reprezentira korisnikov osvrt na događaj u obliku ocjene i komentara.

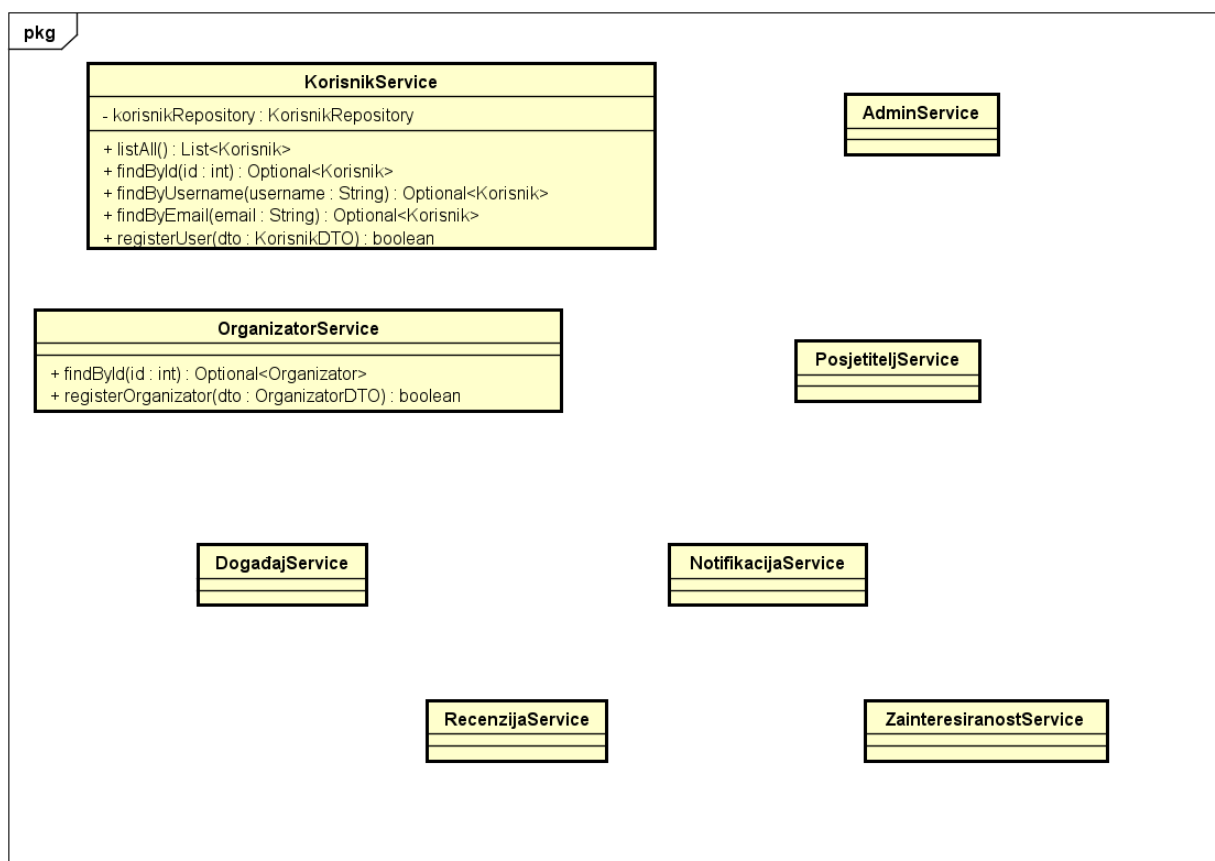
Razred Notifikacija reprezentira obavijest koja se šalje korisniku. Korisnik ima mogućnost uključiti notifikacije za određenu vrstu događaja te za lokaciju na kojoj se događaji organiziraju.

Razred Zainteresiranost reprezentira korisnikovu interakciju s postavljenim događajima. Korisnikova zainteresiranost ima tri razine: "sigurno dolazim", "možda dolazim", "ne dolazim".

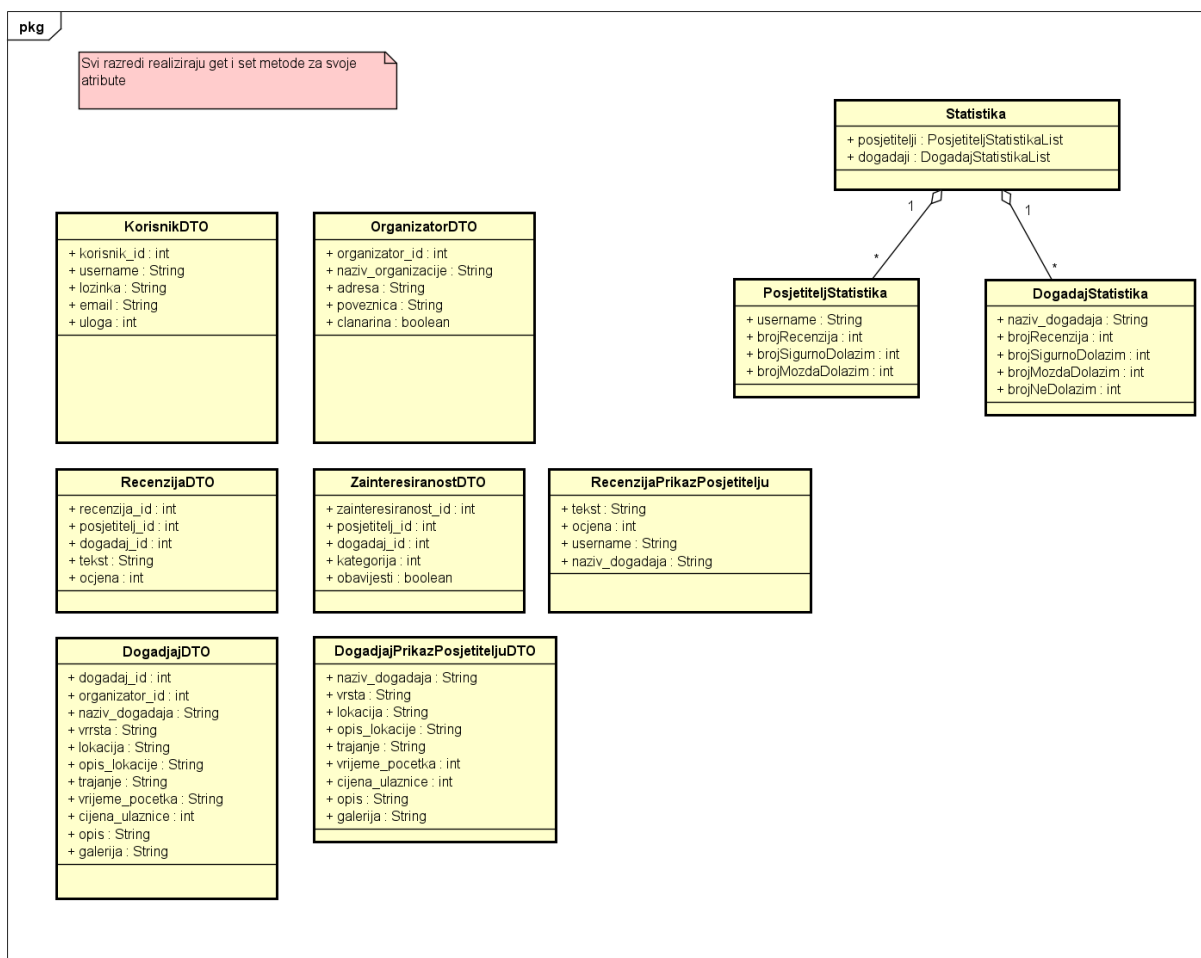
Razred Događaj reprezentira događaj koji je postavio organizator. Lokacija događaja predstavljena je gradskom četvrti - kvartom.



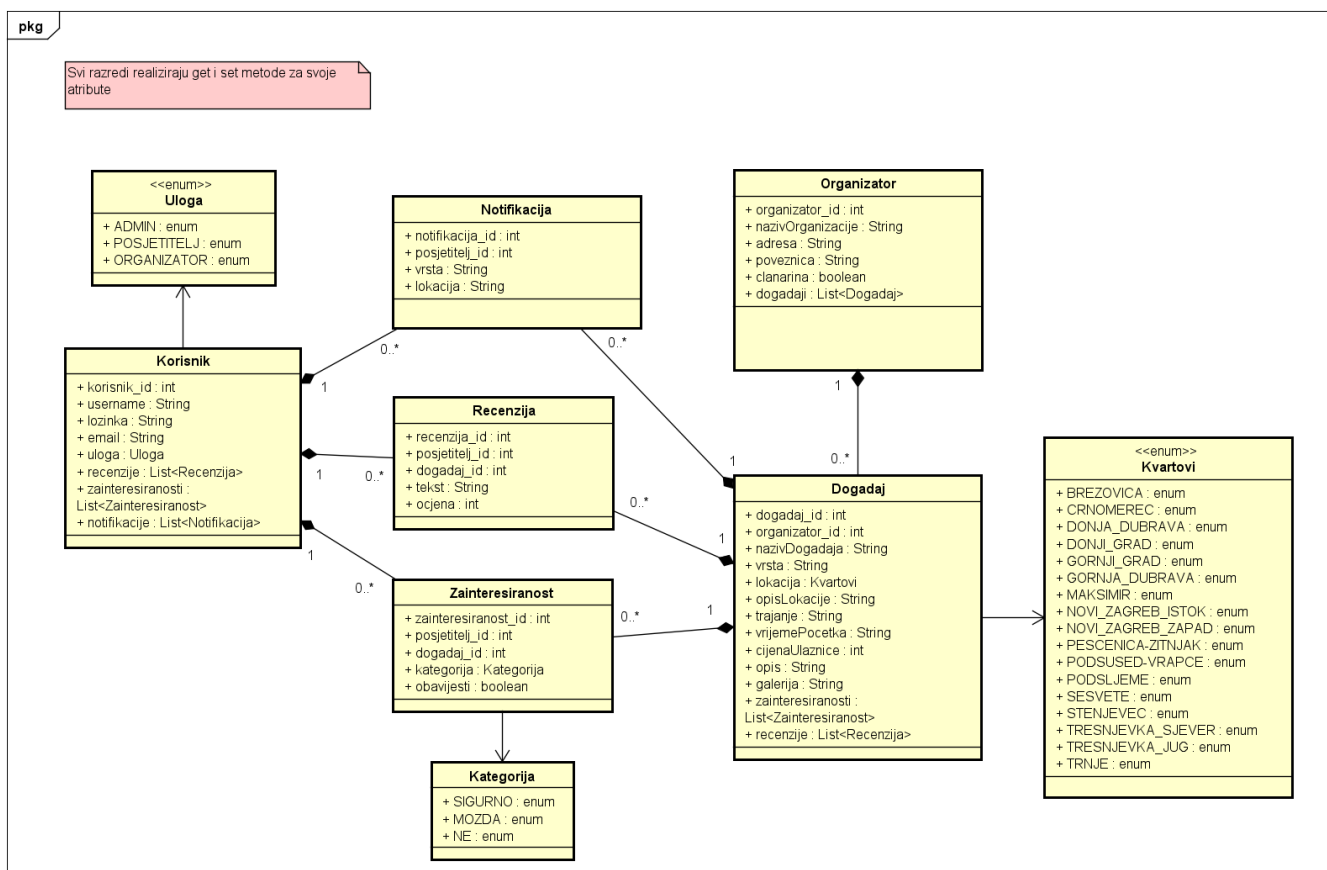
Slika 4.3: Dijagram razreda - Kontroleri



Slika 4.4: Dijagram razreda - Servisi



Slika 4.5: Dijagram razreda - Data Transfer Objects



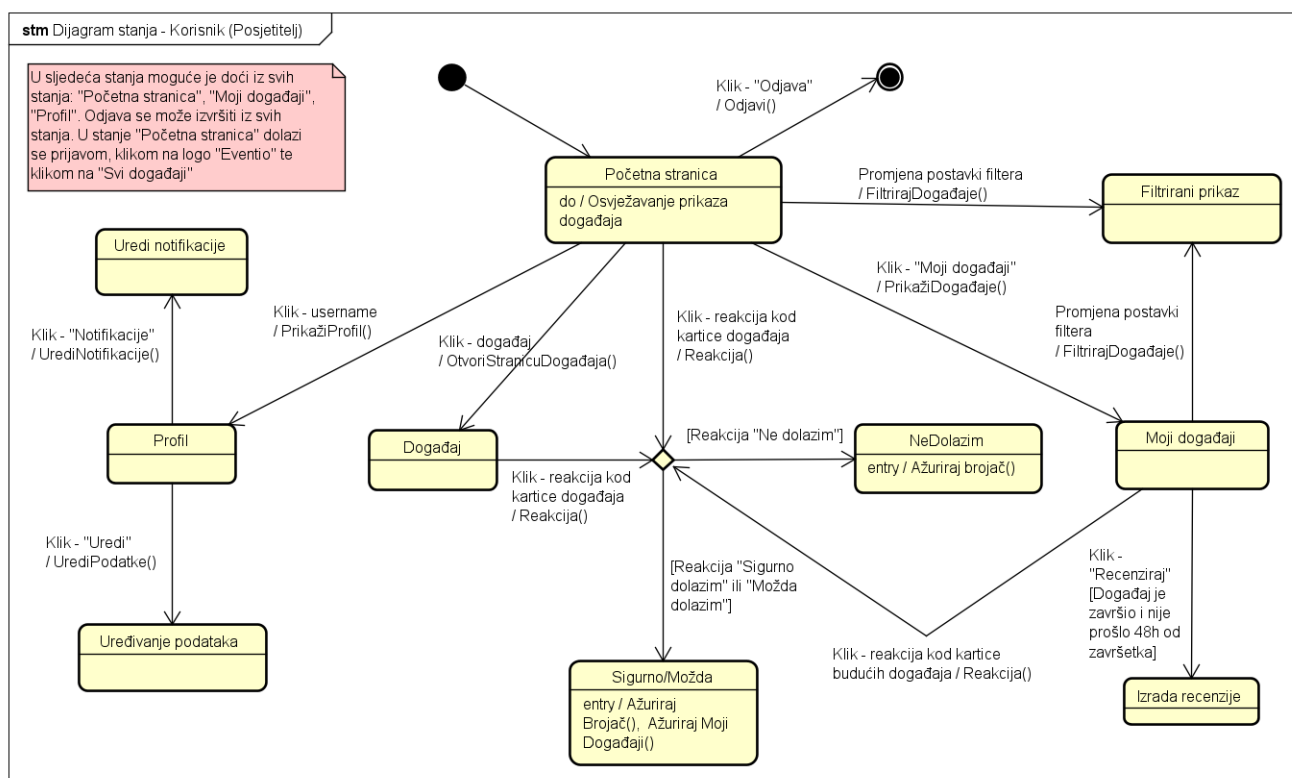
Slika 4.6: Dijagram razreda - Models

**dio 2. revizije**

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

### 4.3 Dijagram stanja

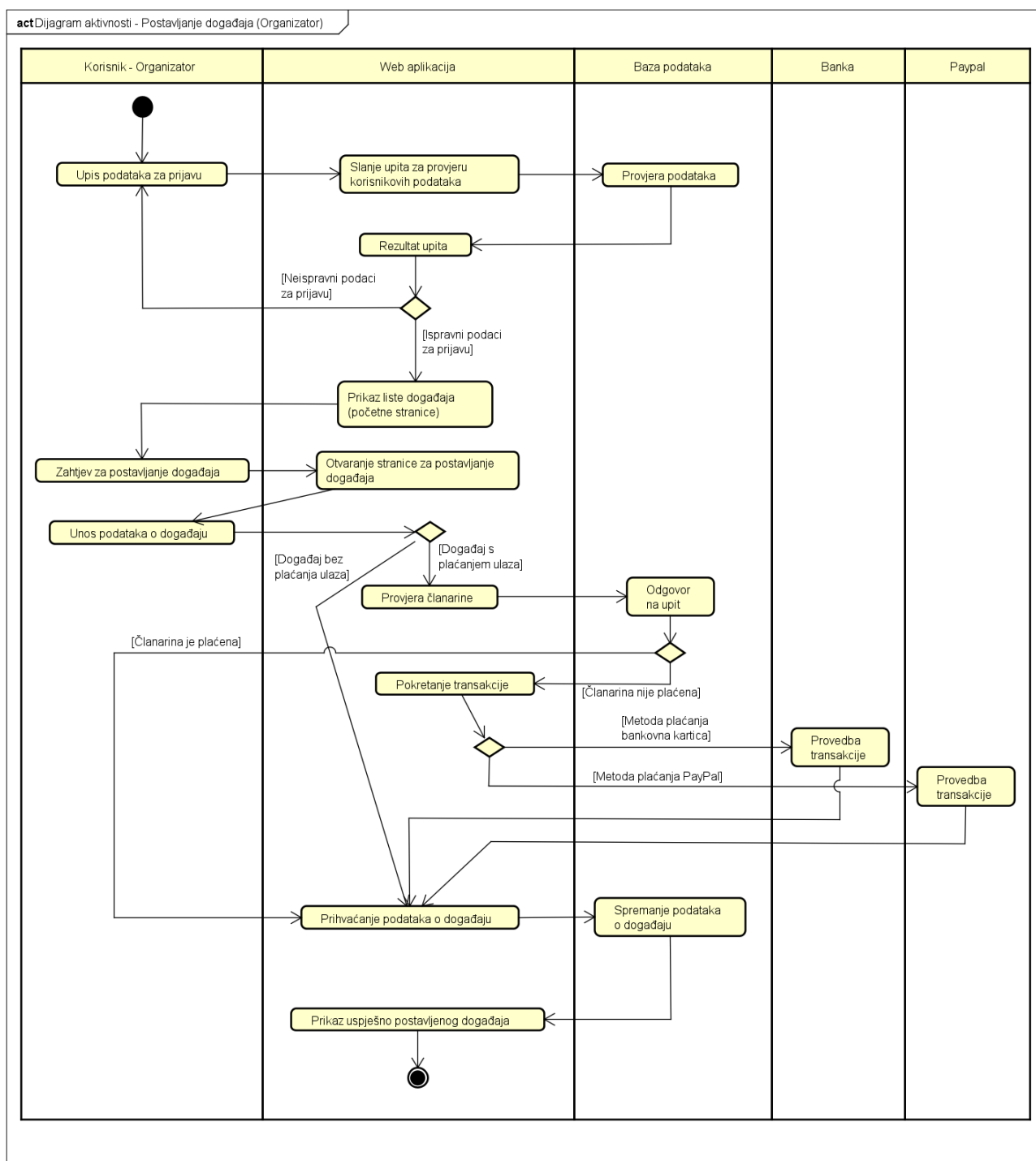
Dijagram stanja opisuje dinamičko ponašanje dijela sustava u vremenu; prikazuje stanja objekta i prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.6 prikazan je dijagram stanja za korisnika - posjetitelja. Kada se korisnik prijavi prikazuje mu se početna stranica na kojoj su vidljivi događaji, filter pomoću kojeg može odlučiti koje događaje želi pregledati, te navigacijska traka preko kojeg pristupa svim događajima, događajima za koje je označio "Sigurno dolazim" ili "Možda dolazim" te profilu. Klikom na događaj korisniku se prikazuje stranica događaja s dodatnim informacijama gdje također ima mogućnost reagirati na događaj. Klikom na "Moji događaji" prikazuju mu se događaji na koje je prethodno reagirao sa "Sigurno dolazim" ili "Možda dolazim". Ako među ovim događajima postoje događaji koji su završili i nije prošlo 48 sati od njihovog završetka, korisnik za njih može izraditi recenziju. Klikom na korisničko ime prikazuju mu se stranica profila na kojoj može promijeniti osobne podatke te urediti postavke notifikacija.



Slika 4.7: Dijagram stanja - Korisnik (Posjetitelj)

## 4.4 Dijagram aktivnosti

Dijagram aktivnosti primjenjuje se za modeliranje poslovnih procesa, upravljačkog i podatkovnog toka. Na slici 4.7 prikazan je dijagram aktivnosti za proces stvaranja novog događaja. Nakon prijave organizator na početnoj stranici odabire opciju postavljanja događaja. Otvara mu se stranica za postavljanje događaja na kojoj unosi sve potrebne podatke o događaju. Ukoliko organizator želi događaj bez plaćanja ulaza, podaci o događaju se odmah procesiraju, spremaju u bazu te se organizatoru prikazuje potvrda o uspješno postavljenom događaju. Ako organizator želi događaj koji će imati plaćanje ulaza, provodi se provjera članarine. Ako je članarina plaćena, proces se nastavlja jednako kao i u slučaju besplatnog događaja. Ako nije, pokreće se plaćanje članarine - transakcija koja se može izvršiti karticom ili PayPal-om. Nakon uspješne transakcije podaci se prihvataju, spremaju u bazu te organizator dobiva potvrdu.



Slika 4.8: Dijagram aktivnosti - Postavljanje događaja (Organizator)



## 4.5 Dijagram komponenti

### *dio 2. revizije*

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

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

Za međusobnu komunikaciju unutar tima koristili smo aplikacije WhatsApp i Discord. *WhatsApp* je popularna platforma koja omogućava korisnicima slanje poruka i medijskih datoteka te obavljanje glasovnih/video poziva. *Discord* je aplikacija za stvaranje servera na kojima članovi mogu komunicirati putem poruka i poziva.

Za modeliranje UML dijagrama koristili smo *Astah UML*, alat koji omogućava korisnicima da kreiraju različite vrste UML dijagrama. Za pripremu i uređivanje dokumentacije koristili smo *TeXstudio*, specifično prilagođeno okruženje za učinkovit rad s LaTeX sustavom.

Po pitanju baze podataka i svega vezanog uz nju koristili smo nekoliko alata. Atom *ERDplus*, popularnim online alatom za izradu ER dijagrama, modelirali smo ER dijagram. Sama baza podataka temeljena je na *PostgreSQL*-u, poznatom sustavu za upravljanje bazama podataka. *Firebase* je platforma koja nam služi za pohranu slika koje organizatori postavljaju uz svoje događaje. Alat *Liquibase* koristili smo za upravljanje promjenama u shemi baze podataka.

Za ispitivanje komponenti iskoristili smo *JUnit*, okvir za testiranje osmišljen za podršku automatskom testiranju Java aplikacija, i *Mockito*, koji nam je poslužio za stvaranje lažnih, mock objekata. Za ispitivanje sustava upotrijebili smo *Selenium IDE*, alat koji omogućuje snimanje, uređivanje i reprodukciju testova jednostavnim klikanjem i snimanjem korisničkih radnji na web stranici.

Za upravljanje izvornim kodom primijenili smo sustav *Git*, dok se sam kod nalazi na *GitHub* platformi unutar udaljenog repozitorija. GitHub je jedna od najpoznatijih web platformi koja pruži usluge za upravljanje projektima temeljenima na Git sustavu.

Naposljetku, za izradu aplikacije upotrijebili smo mnogo različitih alata. Za frontend smo koristili *React.js* uz *JavaScript*, *Vite* te *Material UI*. React.js je popularna JavaScript biblioteka koja se koristi za izgradnju korisničkih sučelja u web aplikacijama. Vite je alat čija je osnovna svrha pojednostaviti i ubrzati razvoj web

aplikacija. Material UI je poslužio za dizajniranje osnovnih komponenti frontenda. Za backend smo koristili **Java** i **Spring Boot**, okvir za izgradnju Java aplikacija temeljenih na Springu, a koji je posebno koristan jer omogućuje programerima da se fokusiraju na poslovnu logiku aplikacije umjesto na konfiguraciju i upravljanje okolinom

Deployment frontenda, backenda te baze podataka postigli smo uz pomoć platforme **Render** koja služi za deployment i hosting web aplikacija.

- 
- Whatsapp <https://www.whatsapp.com/>
  - Discord <https://discord.com/>
  - Astah UML <https://astah.net/products/astah-uml/>
  - TeXstudio <https://www.texstudio.org/>
  - ERDplus <https://erdplus.com/>
  - PostgreSQL <https://www.postgresql.org/>
  - Firebase <https://firebase.google.com/>
  - Liquibase <https://www.liquibase.org/>
  - JUnit <https://junit.org/junit5/>
  - Mockito <https://site.mockito.org/>
  - Selenium IDE <https://www.selenium.dev/selenium-ide/>
  - Git <https://git-scm.com/>
  - GitHub <https://github.com/>
  - React.js <https://reactjs.org/>
  - Vite <https://vitejs.dev/>
  - JavaScript <https://www.javascript.com/>
  - Material UI <https://mui.com/>
  - Java <https://www.java.com/en/>
  - SpringBoot <https://spring.io/projects/spring-boot/>
  - Render <https://render.com/>

## 5.2 Ispitivanje programskog rješenja

### *dio 2. revizije*

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

### 5.2.1 Ispitivanje komponenti

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

### 5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>1</sup>. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

*Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:*

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

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

---

<sup>1</sup><https://www.seleniumhq.org/>

## 5.3 Dijagram razmještaja

### *dio 2. revizije*

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

## 5.4 Upute za puštanje u pogon

### *dio 2. revizije*

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

*Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.*

## 6. Zaključak i budući rad

### *dio 2. revizije*

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

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

# Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Stack Overflow, <http://https://stackoverflow.com/>
8. React dokumentacija <https://legacy.reactjs.org/>
9. PostgreSQL dokumentacija <https://www.postgresql.org/>
10. W3Schools <https://www.w3schools.com/>
11. GitHub <https://github.com/>



# Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe, opća funkcionalnost aktora . . . . .	22
3.2	Dijagram obrasca uporabe, funkcionalnost administratora . . . . .	23
3.3	Dijagram obrasca uporabe, funkcionalnost korisnika . . . . .	24
3.4	Dijagram obrasca uporabe, funkcionalnost organizatora . . . . .	25
3.5	Dijagram obrasca uporabe, funkcionalnost posjetitelja . . . . .	26
3.6	Sekvencijski dijagram za UC1 . . . . .	27
3.7	Sekvencijski dijagram za UC16 . . . . .	28
3.8	Sekvencijski dijagram za UC19 . . . . .	29
4.1	Arhitektura sustava . . . . .	31
4.2	Dijagram baze podataka . . . . .	37
4.3	Dijagram razreda - Kontroleri . . . . .	41
4.4	Dijagram razreda - Servisi . . . . .	42
4.5	Dijagram razreda - Data Transfer Objects . . . . .	43
4.6	Dijagram razreda - Models . . . . .	44
4.7	Dijagram stanja - Korisnik (Posjetitelj) . . . . .	45
4.8	Dijagram aktivnosti - Postavljanje događaja (Organizator) . . . . .	47

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - upoznavanje sudionika
  - dogovaranje prvih koraka

### 2. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - podijelili dijelove dokumentacije članovima za rad (2,3.1,3.1.1 )
  - složen osnovni Spring Boot server

### 3. sastanak

- Datum: 2. studenoga 2023.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - podijelili dijelove dokumentacije (4, 4.1, 4.1.1, 4.1.2, 4.2)
  - započeta izrada registracije i login-a (backend, frontend)

### 4. sastanak

- Datum: 10. studenoga 2023.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - pregled napravljenog, dogovor za ispravke, poboljšanja, deploy

## 5. sastanak

- Datum: 13. prosinca 2023.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - pregled napravljenog
  - izrada skica screen-ova
  - podjela frontend komponenti

## 6. sastanak

- Datum: 22. prosinca 2023.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - pregled napravljenog
  - dogovor o radu za vrijeme praznika

## 7. sastanak

- Datum: 11. siječnja 2024.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - pregled napravljenog
  - podjela završnih zadataka

## 8. sastanak

- Datum: 18. siječnja 2024.
- Prisustvovali: F. Buhiniček, M. Grdić, M. Sršić, N. Kušen, J. Šarolić, S. Boka, G. Haramija
- Teme sastanka:
  - pregled i finalizacija aplikacije i dokumentacije
  - priprema za predaju

## Tablica aktivnosti

	Gašpar Haramija	Filip Buhiniček	Sandro Boka	Marko Sršić	Mateo Grdić	Nikola Kušen	Jakov Šarolić
Upravljanje projektom	40						
Opis projektnog zadatka	8	7					
Funkcionalni zahtjevi				2.5	5		
Opis pojedinih obrazaca				12			
Dijagram obrazaca			7				
Sekvencijski dijagrami	5			1			
Opis ostalih zahtjeva	2						
Arhitektura i dizajn sustava	3			2			
Baza podataka		9		1.5	10		
Dijagram razreda	2		6	8			
Dijagram stanja				5.5			
Dijagram aktivnosti				5.5			
Dijagram komponenti	5						
Korištene tehnologije i alati	7	5	3	5	5	5	5
Ispitivanje programskog rješenja	5	5			1.5	1.5	
Dijagram razmještaja	5						
Upute za puštanje u pogon	6.5	1.5	1.5	1.5	1.5	1.5	1.5
Dnevnik sastajanja	3	3		4			
Zaključak i budući rad	2						
Popis literature				1.5			

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	<b>Gašpar Haramija</b>	<b>Filip Buhiniček</b>	<b>Sandro Boka</b>	<b>Marko Sršić</b>	<b>Mateo Grdić</b>	<b>Nikola Kušen</b>	<b>Jakov Šarolić</b>
Front end			25		25	30	20
Back end	10	27			28	30	
Deploy	10.5	8			8	8	
Opće ažuriranje dokumentacije	9	1.5		10			
GitHub	10	8			5		
Readme, upute	2			1			

## Dijagrami pregleda promjena

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