

Oblikovanje programske potpore
Ak. god. 2019./2020.

Zabava.NET
Dokumentacija, Rev. 1



Grupa: *WebMeisters*
Voditelj: *Antonio Filipović*
Datum predaje: *15.11.2019.*

Nastavnik: *Nikolina Frid*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	26
3.2 Ostali zahtjevi	32
4 Arhitektura i dizajn sustava	33
4.1 Baza podataka	37
4.1.1 Opis tablica	37
4.1.2 Dijagram baze podataka	43
4.2 Dijagram razreda	44
4.3 Dijagram stanja	53
4.4 Dijagram aktivnosti	55
4.5 Dijagram komponenti	57
5 Implementacija i korisničko sučelje	58
5.1 Korištene tehnologije i alati	58
5.2 Ispitivanje programskog rješenja	59
5.2.1 Ispitivanje komponenti	59
5.2.2 Ispitivanje sustava	63
5.3 Dijagram razmještaja	67
5.4 Upute za puštanje u pogon	68
6 Zaključak i budući rad	73
Popis literature	74
Indeks slika i dijagonama	76

Dodatak: Prikaz aktivnosti grupe

77

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Filipović	02.11.2019.
0.2	Dodani oblikovni obrasci	Marić	04.11.2019.
0.5	Dodani neki sekvencijski dijagrami	Tišma	04.11.2019.
0.6	Dodani neki UML dijagrami Izjene u oblikovnim obrascima	Marić	05.11.2019.
0.8	Naslovna stranica Dodan opis projektnog zadatka	Pavelić	07.11.2019.
0.9	Dodani funkcionalni zahtjevi	Duvnjak	07.11.2019.
0.10	Dodani <i>Use Case</i> dijagrami	Marić	10.11.2019.
0.11	Dodano još sekvencijskih dijagrama	Tišma	12.11.2019.
0.12	Dodano arhitektura i dizajn Dodani ostali zahtjevi	Pavelić	12.11.2019.
0.13	Dodana baza podataka Ispravak funkcionalnih zahtjeva	Duvnjak	12.11.2019.
0.14	Uređivanje <i>Use Case</i> dijagrama	Marić	14.11.2019.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Filipović Vusić Marić Tišma Duvnjak Pavelić	15.11.2019.
1.1	Izmjene <i>Use Case</i> dijagrama	Marić	8.1.2020.
1.2	Dodani dijagrami razmještaja	Duvnjak	14.1.2020.
1.3	Dijagram aktivnosti Dijagram komponenti Dijagram stanja Programsko rješenje	Pavelić Marić	15.1.2020.

Rev.	Opis promjene/dodataka	Autori	Datum
1.4	Ispitivanje komponenti Tehnologije i alati Dijagram razreda	Pavelić	16.1.2020.
1.5	Popravak baze podataka	Duvnjak	16.1.2020.
1.6	Popravak dijagrama komponenti Zaključak i budući rad Upute za puštanje u pogon Dnevnik sastajanja i tablica aktivnosti	Marić Pavelić	16.1.2020.
2.0	Konačni tekst predloška dokumentacije	Filipović Vusić Marić Tišma Duvnjak Pavelić	16.1.2020.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti novu personaliziranu platformu u obliku web aplikacije koja će se koristiti za promociju zabavnih događanja u gradu. Za korištenje aplikacije potrebno je registrirati se kao organizator ili posjetitelj. Organizatori zabavnih događanja postavljaju najave za događanja koja organiziraju (koncerti, događanje u klubovima, kazališne predstave, ...) i obvezni su plaćati mjesecnu članarinu putem PayPala ili kreditne kartice, dok zainteresirani posjetitelji besplatno koriste aplikaciju te mogu najaviti svoj dolazak na neki događaj i pisati recenzije. Organizatori imaju svoje javne profile na kojima se nalaze osnovni podaci:

- naziv
- adresa
- poveznice na vlastite web ili Facebook stranice
- popis svih događanja koja su oglašena putem aplikacije u zadnje 2 godine

Za svako događanje dostupni su detalji:

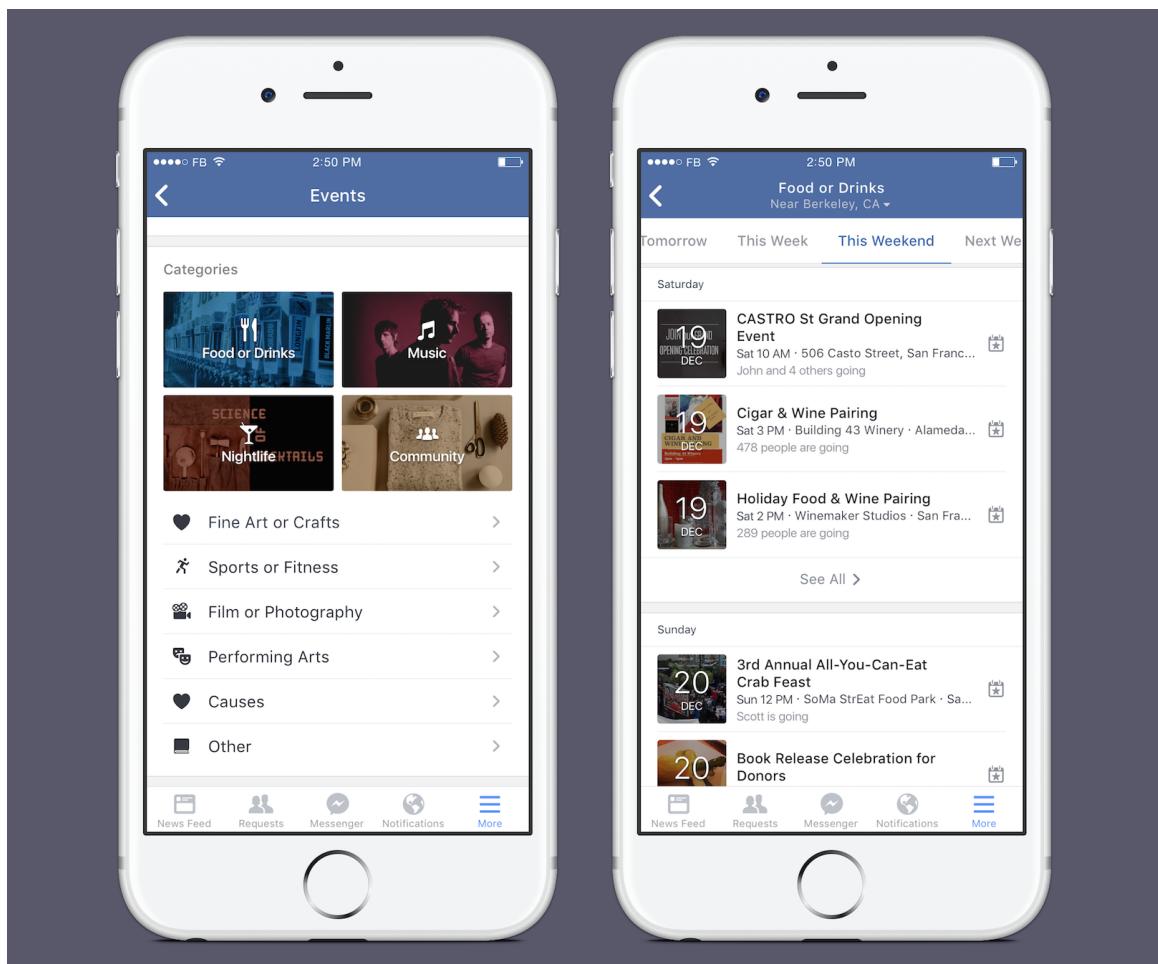
- naziv
- vrsta
- lokacija
- vrijeme početka i trajanje
- foto/video galerija
- popis svih recenzija vezanih uz to događanje

(Podatke o događanju, fotografije i video unose organizatori.)

Posjetitelji u aplikaciji mogu vidjeti popis aktualnih događanja u odabranom vremenskom razdoblju (24 sata, 7 dana, 30 dana) te mogu za svako događanje izraziti interes („sigurno dolazim“, „možda dolazim“, „ne dolazim“) koji se može promijeniti ukoliko se predomisle. Na taj način organizatori vide kolika je zainteresiranost za pojedino događanje, a pošto je broj zainteresiranih javno vodljiv uz događanje na profilu organizatora, posjetitelji mogu vidjeti koliko će ljudi otprilike prisustvovati događanju. Posjetitelji također mogu pisati recenzije za događanja koja su završila u proteklih 48 sati te mogu namjestiti postavke da im aplikacija automatski šalje

obavijesti o najnovijim događanjima prema zadanim kriterijima: vrsta događanja i područje.

U sustavu osim organizatora i posjetitelja postoje i administratori koji postavljaju cijenu članstva te upravljaju korisnicima. Ukoliko organizator nije podmirio svoju mjesecnu članarinu ne može pristupiti svom računu te s obzirom na to nije u mogućnosti vidjeti ili mijenjati već postojeća događanja niti kreirati nova. Prilikom prijave na svoj račun jedino što organizator može napraviti je podmiriti svoje dugove.



Slika 2.1: Facebook događaji

Slično postojeće rješenje su *Facebook događaji* gdje se mogu vidjeti najave za razna zabavna događanja, izraziti interes za njih, pisati recenzije i vidjeti fotografije. Razlika je u tome što ne postoje organizatori i posjetitelji, već se svi jednako i besplatno registriraju. Pošto nema razlike među korisnicima svatko može napraviti najavu za događanje koja nam daje kratki opis događanja i informacije o lokaciji,

njegovom početku i trajanju, broju ljudi koji su zainteresirani za njega, itd. Pošto je ovo slično rješenje dio društvene mreže *Facebook* postoji još dodatnih mogućnosti kao što su: slanje poziva za neko događanje prijateljima, komentiranje tuđeg interesa za neko događanje, i slično. Društvenu mrežu *Facebook* koristi jako puno ljudi koji pripadaju raznim skupinama, a svi koji ga koriste služe se i njegovom opcijom za organizaciju događanja (*Facebook događaji*). Neki ga koriste isključivo zbog toga što na jednom mjestu mogu imati organizirani pregled događanja za koja su zainteresirani. Stoga smatram da bi i ova naša platforma bila popularna unutar više različitih skupina ljudi.

Platforma koju smo napravili jedna je od mogućih verzija koja zadovoljava uvjete zadane projektnim zadatkom, no to ne znači da se nije moguća nadogradnja. Neki od mogućih dodataka su: ocjene od 1 do 5 kojima bi posjetitelji mogli ocijeniti događanje bez pisanja recenzija, mogućnost kupovine ulaznica i reklamnih materijala putem aplikacije, forum za korisnike na kojem bi mogli davati ideje kako unaprijediti aplikaciju, ...

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Naručitelj
2. Korisnici sustava
 - (a) Organizatori
 - (b) Posjetitelji
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) Pregledati najnovije događaje naslovne stranice
 - (b) registrirati se kao posjetitelj(nema članarine), stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, lozinka, ime, prezime i e-mail adresa
 - (c) registrirati se kao organizator(mjesečna članarina), stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, lozinka, ime, prezime, e-mail adresa i broj kreditne kartice
2. Posjetitelj (inicijator) može:
 - (a) pregledavati sva događanja
 - (b) pregledavati događanja na koje je pretplaćen
 - (c) izraziti interes s jednom od 3 sljedeće opcije:
 - i. *sigurno dolazim*
 - ii. *možda dolazim*
 - iii. *ne dolazim*
 - (d) promijeniti interes

- (e) pregledati recenzije
- (f) pisati recenzije za događanja koja su završila u proteklih 48h
- (g) brisati recenzije
- (h) vidjeti popis aktualnih događanja u odabranom vremenskom razdoblju:
 - i. *24h*
 - ii. *7 dana*
 - iii. *30 dana*
- (i) pretplatiti se na obavijesti o najnovijim događanjima prema zadanim kriterijima:
 - i. *vrsta događanja*
 - ii. *područje*
- (j) na profilu organizatora vidjeti koliki je broj zainteresiranih posjetitelja za pojedini događaj

3. Organizator (inicijator) može:

- (a) postavljati najave za događanja koja organiziraju
 - i. *koncerti*
 - ii. *kazališne predstave*
 - iii. *događanja u klubovima*
 - iv. *itd.*
- (b) uređivati i obrisati događaj
- (c) unositi podatke, fotografije i videozapise sa događanja kojeg su organizirali
- (d) uređivati javni profil
- (e) vidjeti kolika je zainteresiranost za pojedini događaj
- (f) pregledavati recenzije korisnika
- (g) reaktivirati vlastiti profil
- (h) dodati način plaćanja članarine

4. Administrator (inicijator) može:

- (a) postaviti cijenu članarine
- (b) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (c) brisati korisnike
- (d) brisati recenzije koje su u suprotnosti s pravilima korištenja aplikacije
- (e) dodavanje, uređivanje i brisanje kategorije događaja

5. Baza podataka (sudionik) može:

- (a) pohraniti sve podatke o korisnicima i njihovim ovlastima
- (b) pohraniti sve podatke o događanjima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Upravljanje korisnicima

- **Glavni sudionik:** Administrator
- **Cilj:** Pregled registriranih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je logiran i korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Administrator odlazi na svoj profil
 2. Odabire opciju upravljanja korisnicima
 3. Prikaže se lista registriranih korisnika sa njihovim osobnim podacima

UC1.1 - Brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati korisnika i njegovih podataka iz baze
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je logiran i korisnik postoji u bazi podataka
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregled korisnika
 2. Odabire opciju brisanje korisnika
- **Opis mogućih odstupanja:**
 - 2.a Korisnik se ne može izbrisati iz baze
 1. Sustav javlja administratoru da ne može izvršiti akciju

UC2 - Postavljanje cijena članstva

- **Glavni sudionik:** Administrator
- **Cilj:** Postavljanje ili promjena cijene članstva
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Administrator odlazi na svoj profil
 2. Administrator odabire opciju postavljanja cijene članstva i postavlja cijenu
- **Opis mogućih odstupanja:**

- 2.a Administrator pokušava postaviti neispravnu cijenu
 1. Akcija se odbija i vraća administratora na početak

UC3 - Provjera valjanosti računa

- **Glavni sudionik:** Baza podataka
- **Cilj:** Provjeriti valjanost računa organizatora
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator koji se želi prijaviti u sustav postoji
- **Opis osnovnog tijeka:**
 1. Prilikom prijave organizatora provjerava se valjanost računa po tome do kada mu je valjana članarina
 2. Provjerom članarine u bazi se postavlja odgovarajuća zastavica vezano za korisnikov račun

UC4 - Prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijava u sustav
- **Sudionici:** Baza podatka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Korisnik unosi sve potrebne podatke u web obrascu i potvrđuje svoj unos
 2. Provjera ispravnosti unesenih podataka te postoji li korisnik u bazi podataka.
 3. Ako korisnik postoji u bazi podataka, prijavljuje se u sustav i pristupa korisničkim funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime/e-mail/lozinka
 1. Sustav obavještava korisnika o neuspjeloj prijavi i vraća ga na stranicu za prijavu

UC5 - Registracija

- **Glavni sudionik:** Neprijavljeni korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju za registraciju kao posjetitelj ili organizator
 2. Unosi potrebne korisničke podatke
 3. Aplikacija provjerava valjanost unesenih podataka, te provjerava da li korisnik već postoji u bazi podataka
 4. Ako korisnik ne postoji u bazi podataka, aplikacija upisuje u bazu podatke o novom korisniku
- **Opis mogućih odstupanja:**
 - 3.a Unos podatka u nedozvoljenom formatu
 1. Sustav obaveštava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke ili odustaje od prijave
 - 3.b Odabir zauzetog korisničkog imena ili e-maila
 1. Aplikacija odbija registraciju korisnika i vraća ga na početak registracije

UC6 - Pregled najnovijih događaja naslovne stranice

- **Glavni sudionik:** Prijavljeni i neprijavljeni korisnici
- **Cilj:** Prikazati svima nove događaje
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisniku se prikazuje lista događaja koje može posjetiti
 2. Korisnik odabire događaj i prikazuju se informacije o njemu
 3. Predstavljanje korisniku mogućnosti postanka trajnog korisnika stranice nuđenjem mogućnosti prijave kao posjetitelj ili da postane jedan od organizatora

UC7 - Pregled informacija profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled informacija profila
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Ima dostupan pregled svojih informacija

UC7.1 - Uređivanje javnog profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Unos ili izmjena osnovnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregled osnovnih podataka
 2. Korisnik mijenja podatke
 3. Korisnik spremi podatke i baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik je promijenio podatke, ali nije odabrao opciju "Spremi promjenu"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora
 - 3.a Korisnik želi spremiti promjenu, ali podaci su promijenjeni u neispravne podatke
 1. Sustav upozorava korisnika da mora unijeti ispravne podatke kao i kod registracije

UC8 - Pregled događaja za koje je korisnik zainteresiran

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Pregledati sve događaje za koje je posjetitelj izrazio zainteresiranost
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled događaja za koje sam izrazio interes

UC8.1 - Promjena interesa

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Promjena interesa za događaj koji zanima posjetitelja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pregledava događaj
 2. Odabire opciju pregled događaja za koje je izrazio interes

3. Pokraj događaja za koji je izrazio interes u padajućem izborniku odabire interes koji želi

UC8.2 - Brisanje interesa

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Obrisati interes za određeni događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen i već ima izraženu zainteresiranost za događaj
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled događaja za koje je izrazio interes
 3. Pored događaja za koji je izrazio interes bira opciju "Obrisni interes"

UC9 - Pregled napisanih recenzija

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Pregled napisanih recenzija
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju napisane recenzije
 3. Prikazuje mu se lista napisanih recenzija

UC9.1 - Uređivanje recenzije

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Uređivanje recenzije
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i od završetka događaja nije prošlo više od 48 sati
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil i odabire opciju „Napisane recenzije“
 2. Ispisuje mu se lista događaja i recenzije koje su napisane
 3. Korisnik odabire opciju uredi recenziju pokraj recenzije koju želi uredit
 4. Izvršava promjene i sprema recenziju
- **Opis mogućih odstupanja:**

4.a Korisnik želi ostaviti prazan obrazac recenzije i kliknuti spremi promjene

1. Sustav odbija ostavljanje praznog obrasca

4.b Korisnik želi izaći, a nije kliknuo spremi

1. Sustav ga upozorava da je napisao recenziju a nije kliknuo „Spremi“

UC9.2 - Brisanje recenzije

• **Glavni sudionik:** Posjetitelj

• **Cilj:** Brisanje recenzije za događaj

• **Sudionici:** Baza podataka

• **Preduvjet:** Korisnik je prijavljen i od završetka događaja nije prošlo više od 48 sati

• **Opis osnovnog tijeka:**

1. Korisnik odlazi na svoj profil

2. Odabire opciju napisane recenzije

3. Pored događaja i napisane recenzije odabire opciju brisanje recenzije

UC10 - Dodavanje novog događaja

• **Glavni sudionik:** Organizator

• **Cilj:** Dodavanje novog događaja na listu

• **Sudionici:** Baza podataka

• **Preduvjet:** Korisnik je prijavljen

• **Opis osnovnog tijeka:**

1. Korisnik odlazi na svoj profil i odabire opciju dodavanje događaja

2. Korisnik unosi sve potrebne informacije za taj događaj i sprema ga

• **Opis mogućih odstupanja:**

3.a Korisnik je upisao podatke, ali nije odabrao opciju "Spremi promjenu"

1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

3.b Korisnik nije unio potrebne informacije o događaju

1. Sustav obavještava korisnika da određene informacije o događaju moraju biti unesene i vraća ga na obrazac za unos

UC11 - Pregled vlastitih događaja

• **Glavni sudionik:** Organizator

• **Cilj:** Pregled vlastitih organiziranih događaja

- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Organizator odabire svoj profil
 2. Bira pregled svih svojih događaja

UC11.1 - Brisanje događaja

- **Glavni sudionik:** Organizator
- **Cilj:** Obrisati događaj iz baze podataka te obavijestiti posjetitelje
- **Sudionici:** Baza podataka
- **Preduvjet:** Događaj postoji u bazi podataka
- **Opis osnovnog tijeka:**
 1. Organizator odlazi na svoj profil
 2. Odabire opciju pregled događaja
 3. Pored određenog događaja bira opciju brisanje događaja

UC11.2 - Pregled zainteresiranih posjetitelja

- **Glavni sudionik:** Organizator
- **Cilj:** Pregled posjetitelja koji su izrazili zainteresiranost za događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i događaj je postavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled događaja
 3. Pokraj određenog događaja bira na opciju pregled zainteresiranih posjetitelja

UC12 - Pregled načina plaćanja

- **Glavni sudionik:** Organizator
- **Cilj:** Pregled dosadašnjih načina plaćanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled plaćanja

UC12.1 - Dodavanje načina plaćanja

- **Glavni sudioñik:** Organizator
- **Cilj:** Dodati način plaćanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled plaćanja
 3. Može dodati karticu ili paypal račun

UC12.2 - Brisanje načina plaćanja

- **Glavni sudioñik:** Organizator
- **Cilj:** Obrisati način plaćanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled plaćanja
 3. Briše željeni način plaćanja

UC13 - Pregled događaja

- **Glavni sudioñik:** Posjetitelj
- **Cilj:** Pregled svih događaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na naslovnu stranicu
 2. Prikazuje se popis svih događanja

UC13.1 - Izražavanje interesa

- **Glavni sudioñik:** Posjetitelj
- **Cilj:** Izražavanje interesa za događaj koji zanima posjetitelja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pregledava događaj
 2. Odabire opciju izrazi interes za događaj

3. Bira željeni interes između ponuđenih

UC14 - Pregled svih informacija o događaju

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Pregledati više informacija o događaju te njegovu zainteresiranosti
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao posjetitelj
- **Opis osnovnog tijeka:**
 1. Korisnik pregledava događaje
 2. Pokraj događaja odabire opciju pregled vise informacija o događaju

UC14.1 - Pisanje recenzije

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Pisanje recenzije nakon završenog događaja
- **Sudionici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen i od završetka događaja je prošlo manje od 48 sati
- **Opis osnovnog tijeka:**
 1. Korisnik odabire pregled događaja u proteklih 48 sati na naslovnicu
 2. Pored događaja odabire opciju napiši recenziju
 3. Korisnik piše recenziju i spremi ju
- **Opis mogućih odstupanja:**
 - 2.a Korisnik želi dodati recenziju za događaj za koji je već napisao recenziju
 1. Sustav odbija akciju i obavještava ga da je već napisao recenziju za događaj i da može samo nju uredit
 - 3.a Posjetitelj je napisao recenziju ali nije odabrao opciju spremi.
 1. Sustav ga upozorava da je recenzija napisana, ali nije odabrao opciju spremi
 - 3.b Posjetitelj nije ispunio obazac za recenziju, a želi kliknut spremi
 1. Sustav odbija dodavanje recenzije i upozorava korisnika da ne može ne napisati ništa

UC15 - Plaćanje članarine

- **Glavni sudionik:** Organizator
- **Cilj:** Uplata mjesecne članarine
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju plati članarinu
 3. Korisnik odabire opciju plaćanja mjesecne članarine i izvršava akciju
- **Opis mogućih odstupanja:**
 - 3.a Transakcija nije uspjela
 1. Sustav javlja korisniku da se uplata nije dogodila i vraća ga na mjesto izvršavanja transakcije

UC15.1 - Reaktivacija profila

- **Glavni sudiočnik:** Baza podataka
- **Cilj:** Ponovna aktivacija organizatorovog profila
- **Sudiočnici:** Organizator
- **Preduvjet:** Organizator je platio članarinu
- **Opis osnovnog tijeka:**
 1. Nakon plaćanja članarine u bazi podataka se bilježi da je članarina plaćena
 2. Organizatoru se omogućuje pristup svim mogućnostima sustava

UC16 - Pregled organizatora

- **Glavni sudiočnik:** Posjetitelj
- **Cilj:** Prikaz liste svih organizatora
- **Sudiočnici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled svih organizatora

UC16.1 - Prikaz profila organizatora

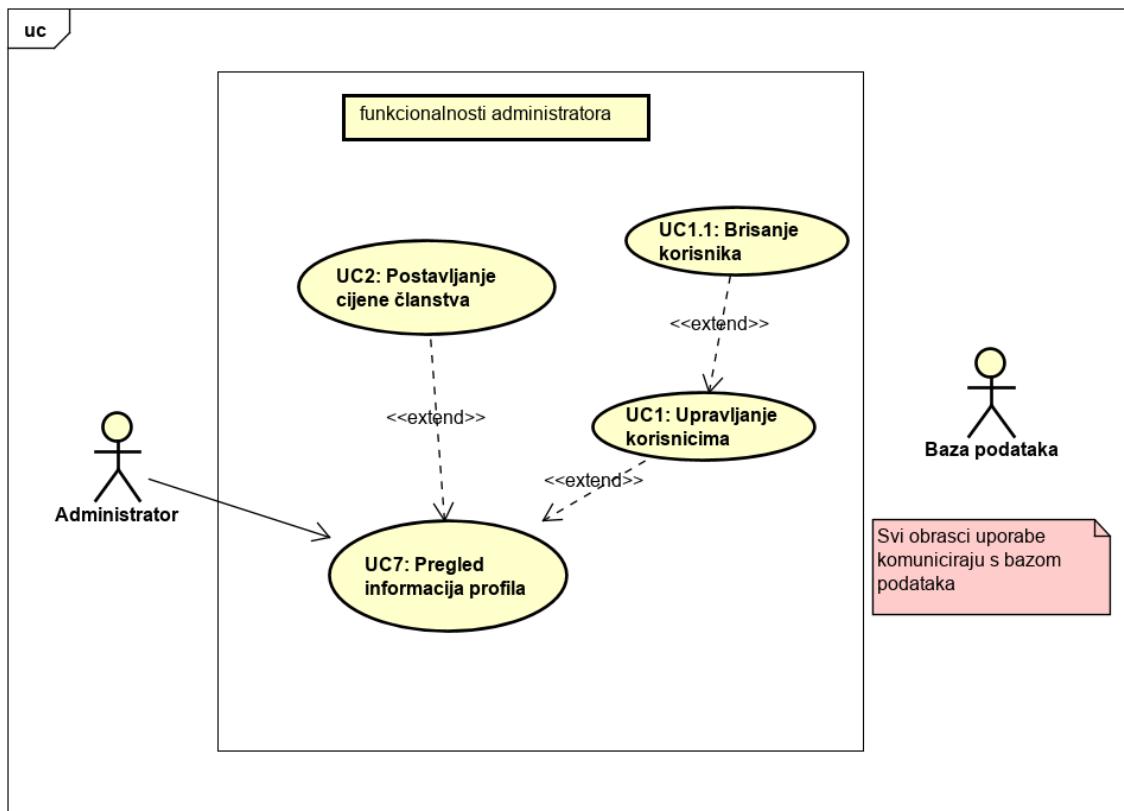
- **Glavni sudiočnik:** Posjetitelj
- **Cilj:** Prikaz detalja o profilu organizatora
- **Sudiočnici:** Baza podataka
- **Preduvjet:** Posjetitelj je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju pregled svih organizatora

3. Pored organizatora bira opciju za prikaz više detalja

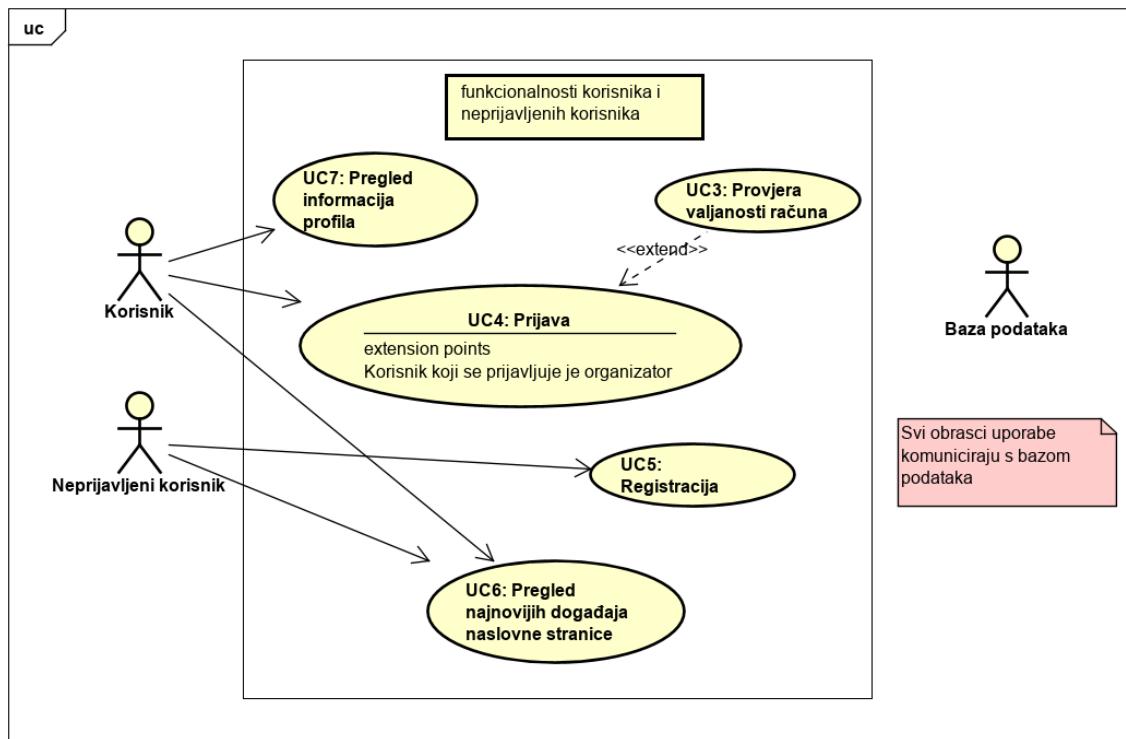
UC17 - Odabir opcije za slanje automatskih obavijesti o najnovijim događajima

- **Glavni sudionik:** Posjetitelj
- **Cilj:** Odabir postavki za slanje automatskih obavijesti o najnovijim događanjima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odlazi na svoj profil
 2. Odabire opciju za postavke obavijesti
 3. Odabire kriterije prema kojima će dobivati obavijesti
- **Opis mogućih odstupanja:**
 - 3.a Korisnik je odabrao postavke, ali nije odabrao opciju "Spremi promjenu"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

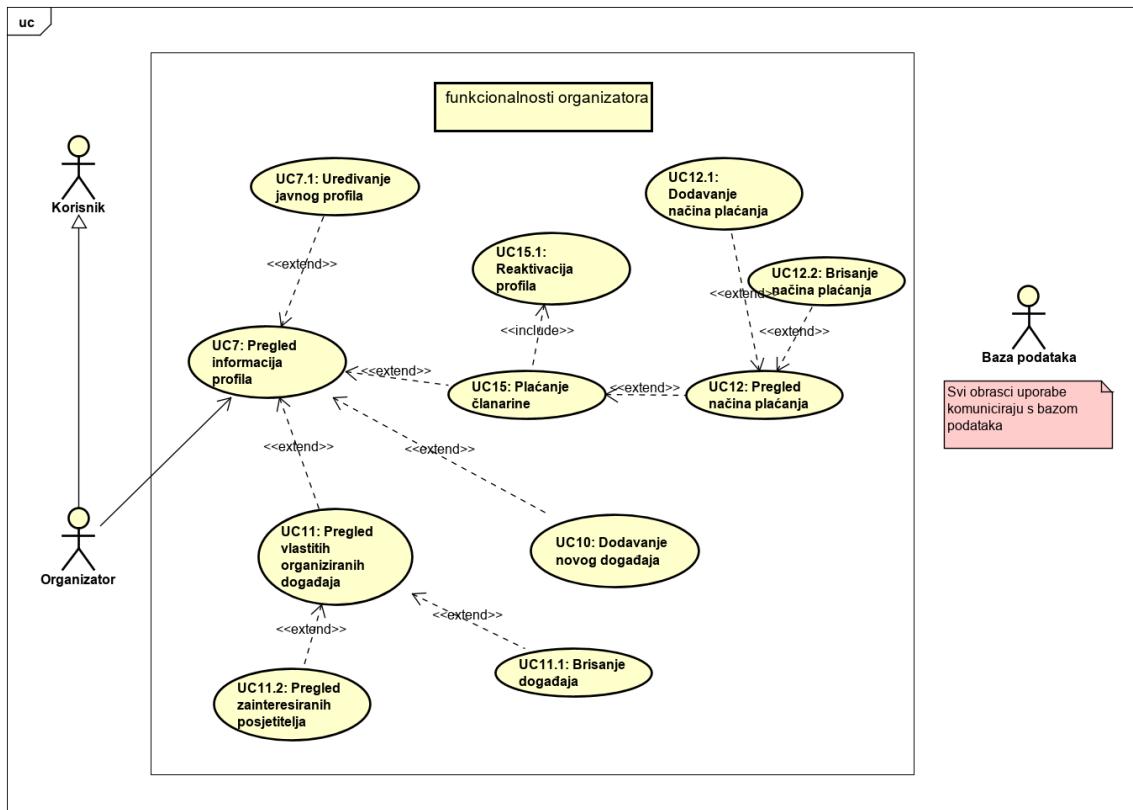
Dijagrami obrazaca uporabe



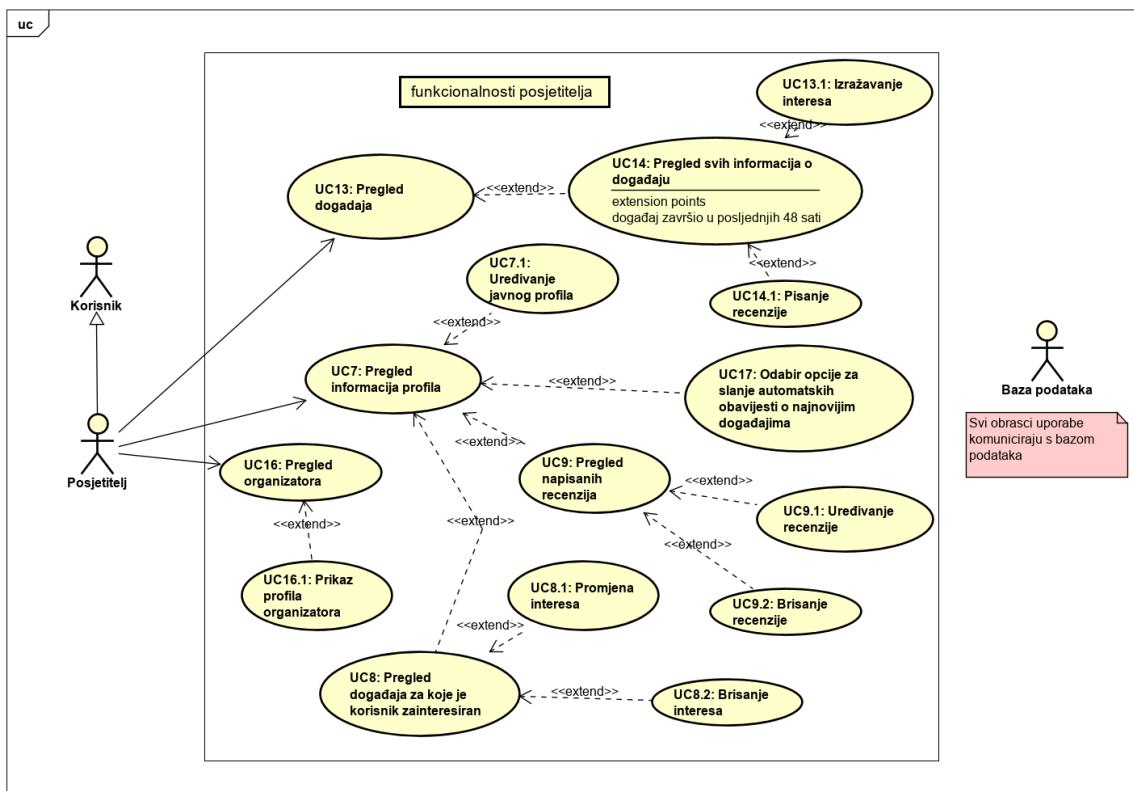
Slika 3.1: Dijagram obrasca uporabe, funkcionalnosti administratora



Slika 3.2: Dijagram obrasca uporabe, funkcionalnosti prijavljenih i neprijavljenih korisnika



Slika 3.3: Dijagram obrasca uporabe, funkcionalnosti organizatora

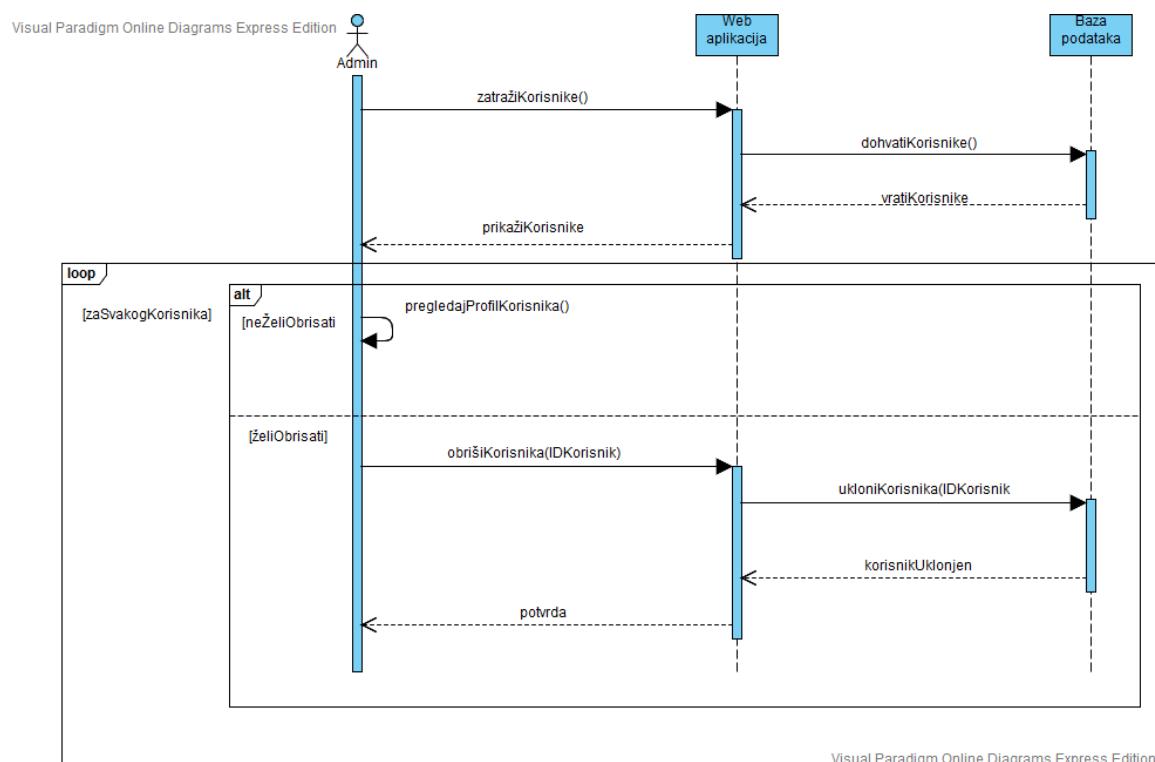


Slika 3.4: Dijagram obrasca uporabe, funkcionalnosti posjetitelja

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC1 – Pregled informacija profila

Administrator šalje zahtjev za prikaz korisnika. Poslužitelj dohvaća sve korisnike i prikazuje ih. Administrator prolazi kroz sve korisnike, pregledava sve informacije njihovog korisničkog profila i ako želi odabereti opciju brisanje korisnika. Odabiranjem opcije brisanja korisnika šalje zahtjev poslužitelju za brisanje korisničkog profila navedenog korisnika. Poslužitelj zatim šalje zahtjev bazi podataka te uklanja korisnika. Nakon uklanjanja poslužitelj obavještava o obavljenoj operaciji.



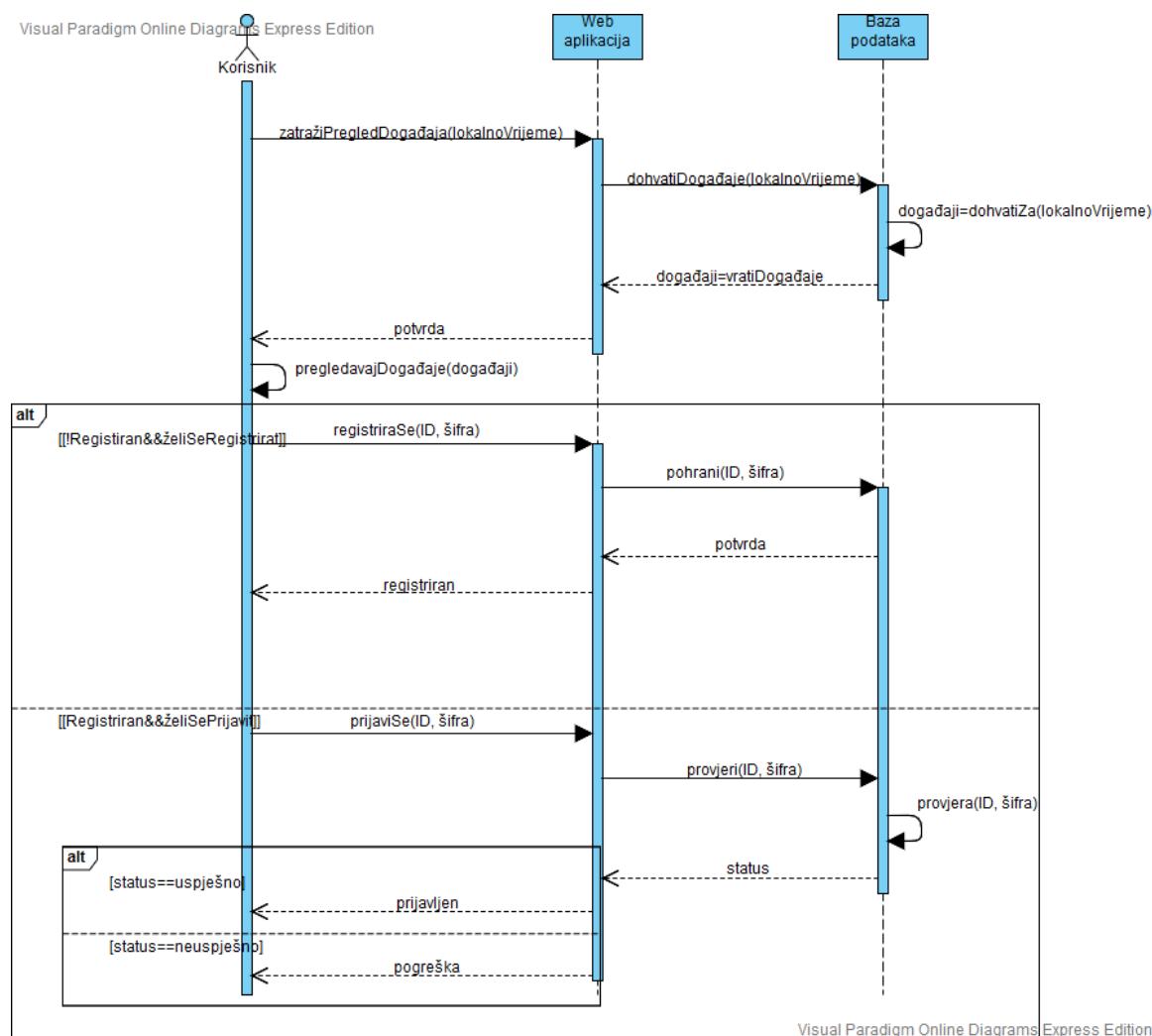
Slika 3.5: Sekvencijski dijagram za UC1

Obrazac uporabe UC5 – Prijava

Korisnik koristi aplikaciju te pregledava najnovije događaje. Zahtjev se šalje poslužitelju koji zatim dohvaća najnovije događaja iz baze podataka s obzirom na lokalno vrijeme upita korisnikovog zahtjeva.

Poslužitelj prikazuje događaja te ih po volji korisnik pojedinačno pregledava.

Korisnik se može registrirati te u tom slučaju ispunjava svoje podatke te šalje zahtjev poslužitelju koji zatim stvara registraciju novog korisnika u bazi podataka. Ukoliko je korisnik već registriran ispunjava svoje podatke koje zatim poslužitelj prosljeđuje bazi podataka te se tamo vrši provjera njihove točnosti i zatim se vraća potvrda korisniku o uspješnoj ili neuspješnoj prijavi.



Slika 3.6: Sekvencijski dijagram za UC5

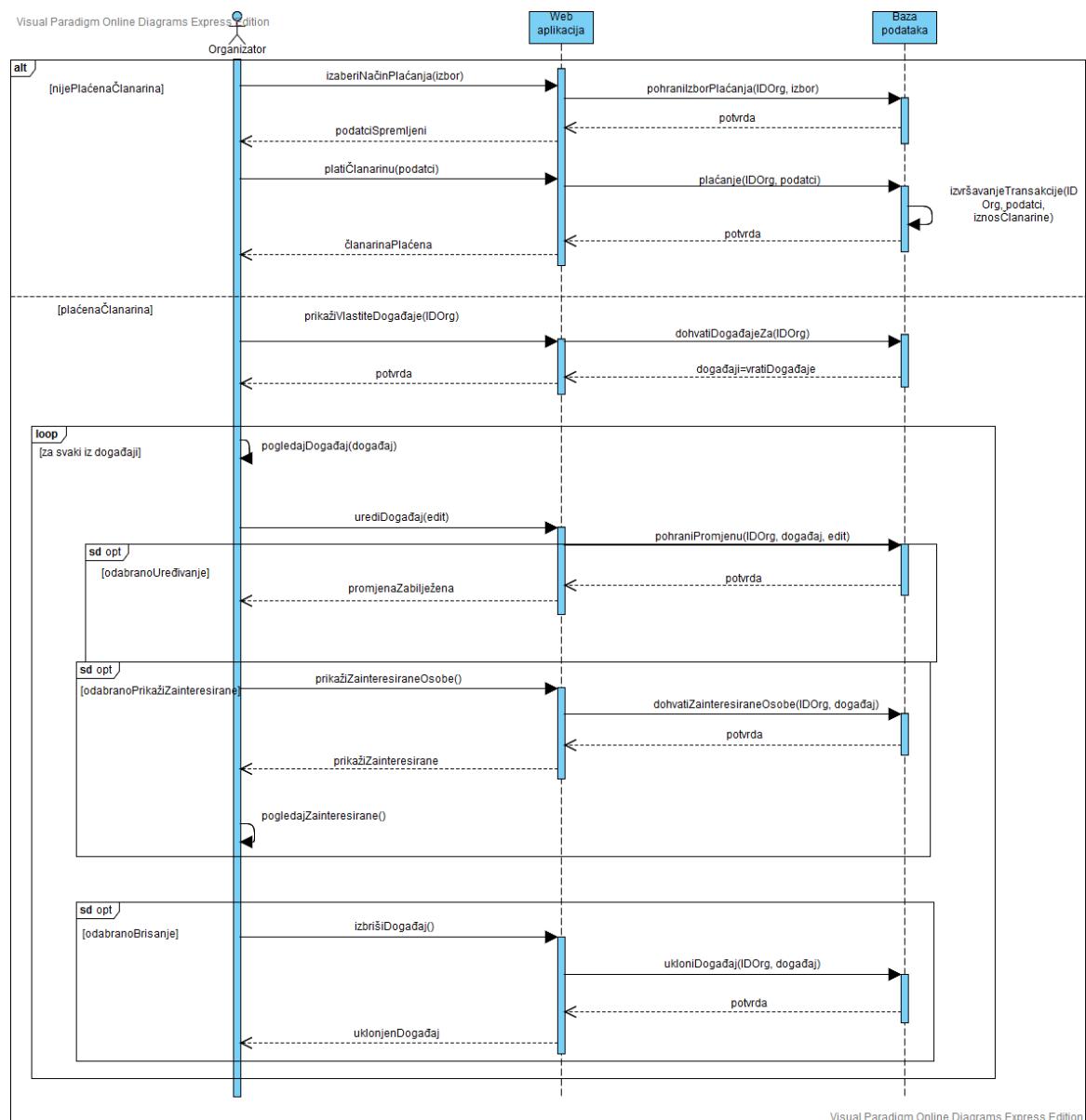
Obrazac uporabe UC10 – Pregled vlastitih organiziranih događaja

Uvjet za korištenje aplikacije kao organizator je plaćanje članarine. U slučaju neplaćene članarine, organizator šalje zahtjeve poslužitelju te odabire opciju plaćanja i ispunjava svoje podatke. Poslužitelj te podatke spremi u bazu podataka te ukoliko je članarina plaćena obavještava organizatora o uspješnoj transakciji.

Organizator zatim može pregledavati sve informacije o vlastitim događajima te po izboru ima opcije uređivanja informacija, prikaz svih zainteresiranih korisnika za događaj te brisanje događaja.

U slučaju uređivanja informacija i brisanja događa organizator šalje zahtjeve poslužitelju koji te promjene pohranjuje u bazi podataka za navedeni događaj.

Ako koristi prikaz svih zainteresiranih korisnika, poslužitelj mu vraća listu svih profila korisnika koji su izrazili interes za događaj iz baze podataka te po izboru pregledava pojedinačni profil.

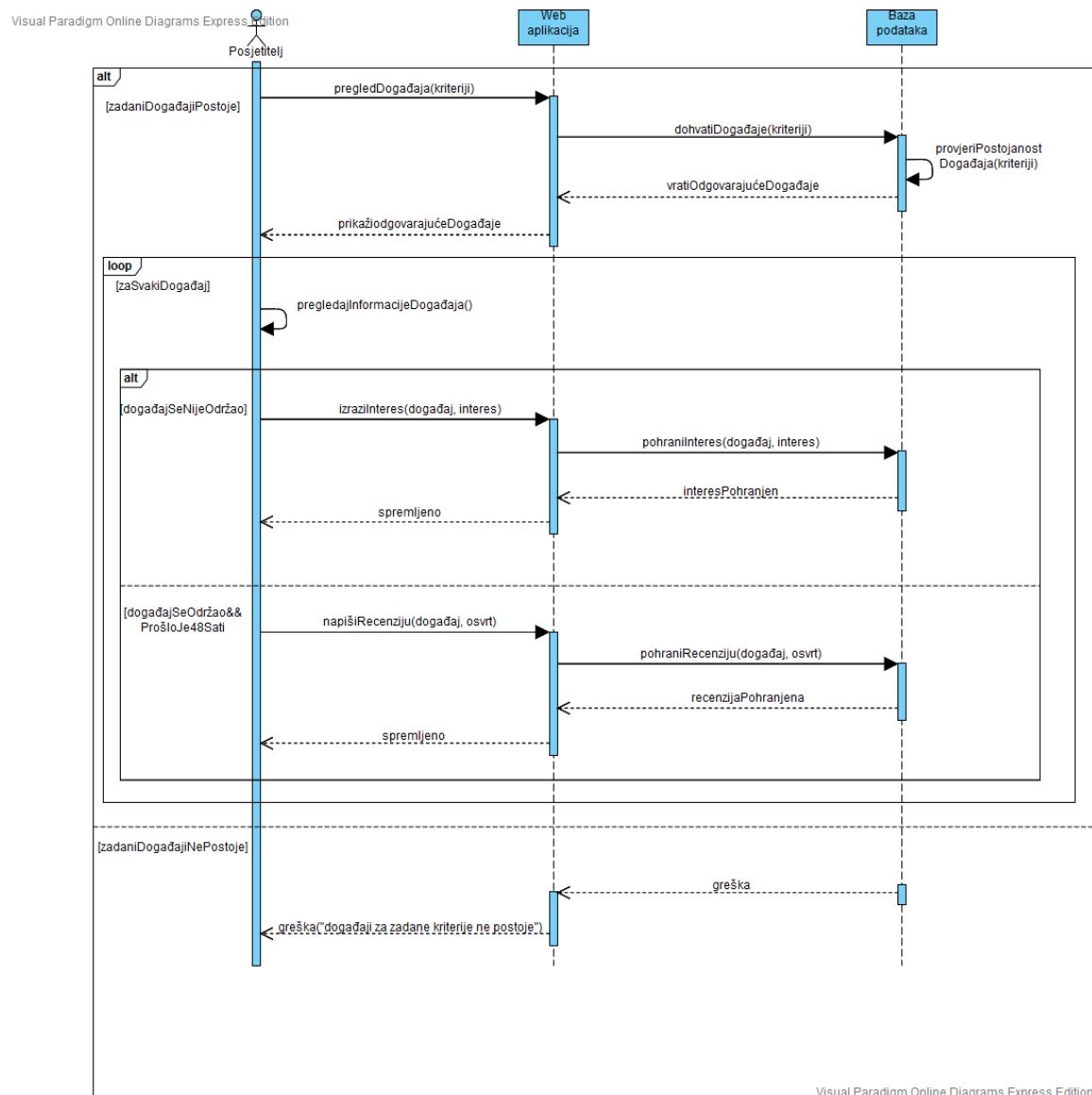


Slika 3.7: Sekvencijski dijagram za UC10

Obrazac uporabe UC13- Pregled događaja prema kriteriju

Posjetitelj unosi kriterije za pregled događaja te šalje zahtjev poslužitelju. Poslužitelj dohvaća događaje sa navedenim kriterijima iz baze podataka te ih prikazuje, inače ako ne postoji javlja grešku posjetitelju. Posjetitelj prolazi kroz svaki događaj i pregledava informacije o svakome. Ako se događaj nije još održao posjetitelj izražava svoj interes te šalje zahtjev poslužitelju. Poslužitelj za navedeni događaj pohranjuje posjetiteljev interes u bazi podataka te ga obavještava da je njegov interes spremlijen.

Ukoliko se događaj već odvio te je prošlo više od 48 sati od njegovog kraja, posjetitelj može napisati svoju recenziju navedenog događaja te poslati zahtjev poslužitelju. U tom slučaju poslužitelj spremi recenziju događaja posjetitelja u bazi podataka te obavještava posjetitelja da je recenzija spremljena.



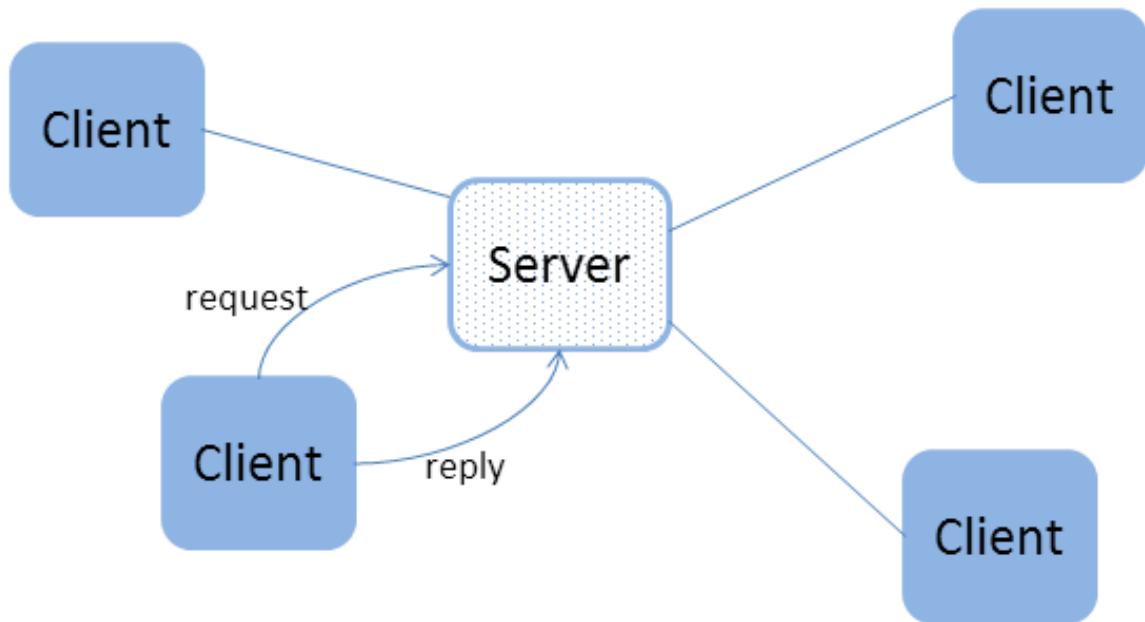
Slika 3.8: Sekvencijski dijagram za UC13

3.2 Ostali zahtjevi

- Mora biti podržan rad aplikacije ukoliko ga više korisnika koristi istovremeno u stvarnom vremenu
- Moraju biti podržani hrvatski dijakritici
- Izvršavanje učitavanja stranice od slanja zahtjeva sa klijentske strane pa sve do generiranja stranice ne smije trajati dulje od par sekundi
- Korisnički podatci moraju biti zaštićeni od strane drugih korisnika
- Recenzije i događaji moraju se ažurirati kako ih korisnici izmjenjuju/dodaju/brišu
- Osigurati da prije registracije korisnik unese potrebne podatke
- Lozinke se ne spremaju u unesenom obliku, već kriptiranom

4. Arhitektura i dizajn sustava

Prilikom izrade naše aplikacije morali smo odlučiti kako ćemo najbolje organizirati aplikaciju tako da više klijenata može istodobno pristupati našoj stranici. Odabrali smo model mrežne aplikacije klijent – poslužitelj(server). Poslužitelj je aplikacija koja nudi uslugu, u našem slučaju nudi mogućnost pretraživanje zabavnih sadržaja. Kada stigne zahtjev, ona šalje odgovor. Klijent je aplikacija koja traži uslugu, u našem slučaju to su svi korisnici. Ona pošalje zahtjev i čeka na odgovor. Dakle, server ne pohranjuje podatke o klijentu, nego samo odgovara na upite klijenta. Protokol kojim se provodi komunikacija je HTTP.



Slika 4.1: Klijent-Server model

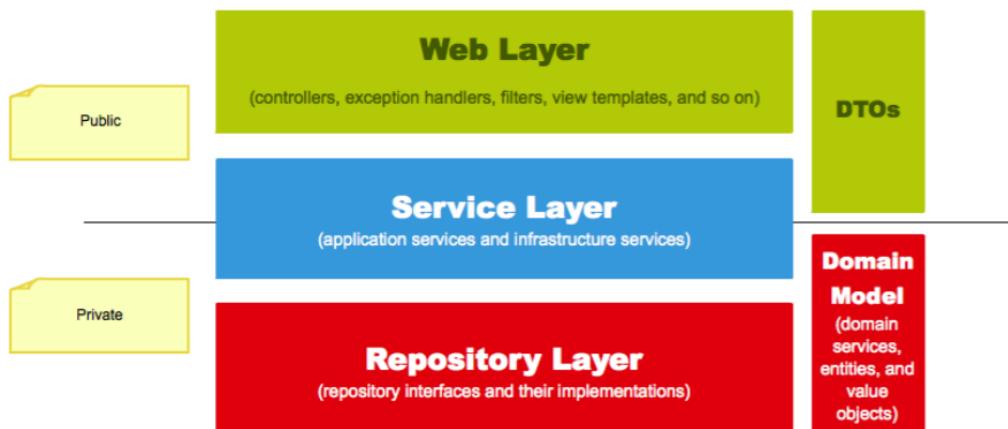
Backend dio aplikacije

Zbog dobre upoznatosti sa Javom, ali i velikog izbora mogućnosti koji nam nudi odabrali smo nju kao jezik za backend dio aplikacije. Uz to koristit ćemo i Spring Framework.

Web sloj će biti gornji sloj aplikacije. On je odgovoran za procesiranje korisnikovog unosa i vraćanja točnih informacija. On se također brine za greške koje bace drugi dijelovi aplikacije. Također to će nam biti prvi sloj da blokiramo korisnike koji žele napasti našu aplikaciju.

Servisni sloj sjedi iza web sloja. On se ponaša kao transakcijska granica te sadrži i servise aplikacije i infrastrukture. Odgovoran je za autorizaciju. On sadrži kod koji komunicira sa vanjskim resursima kao što su datotečni sustav, bazom podataka ili email servisi..

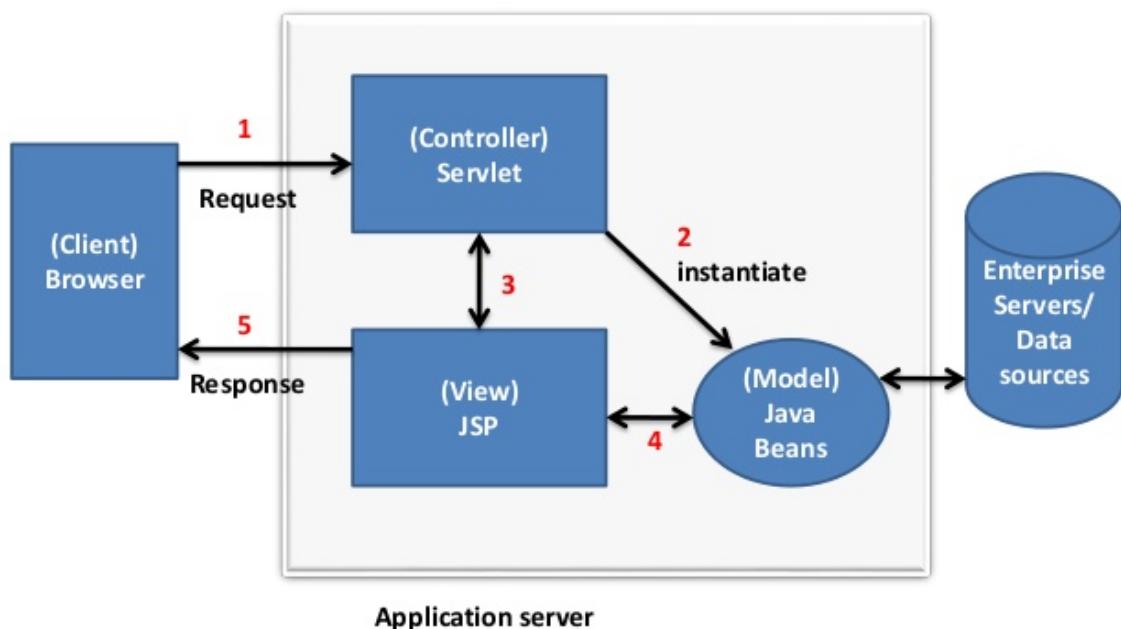
Repozitorjni sloj služi za komunikaciju sa bazom podataka.



Slika 4.2: Spring arhitektura

Ovdje je slika kako bi to sve trebalo biti povezano u cjelinu. Koristimo poznatu dakle Spring MVC arhitekturu. MVC je skraćenica od Model, View, Controller. U kraćim crtama, nakon što dobijemo zahtjev koji dođe do našeg kontrolera, stvaramo model na osnovu podataka iz baze te osnovu njega i odgovora generiranog iz kontrolera stvara se odgovor, VIEW, kod nas JSP datoteke stvara se odgovor. JSP označava Java Server Page, stranicu koja generira html te se na taj način korisniku stvara odgovor.

MVC Architecture

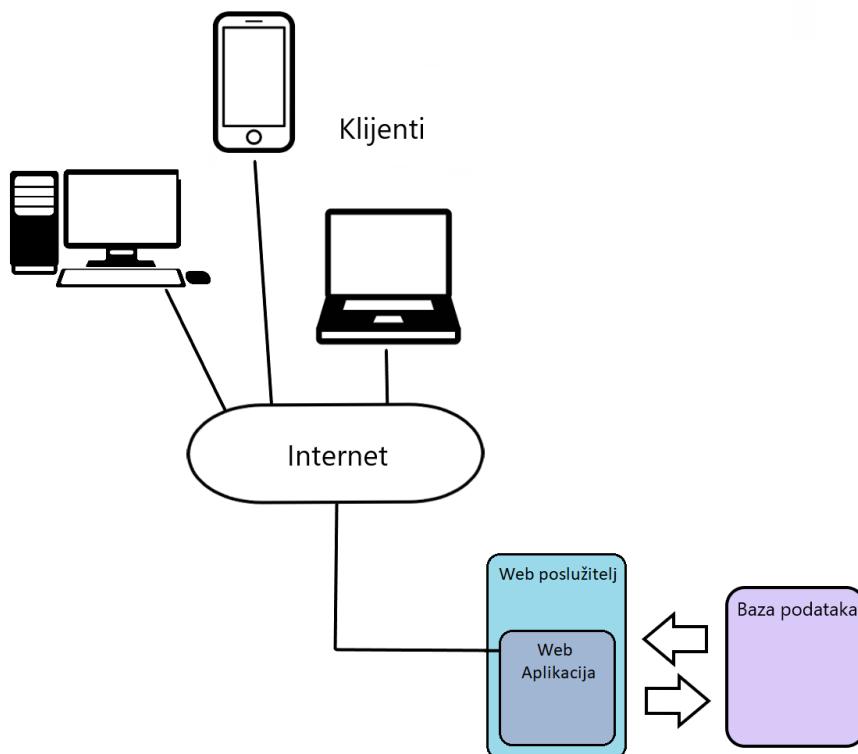


Slika 4.3: MVC arhitektura

Frontend dio aplikacije

Kako je spomenuto već za frontend dio aplikacije koristimo se JSP tehnologijom. U JSP datoteci mi dobivamo model na osnovu kojeg se generiraju komponente i dakle cijela stranica. Ona stvara odgovor i generira HTML, a izgled komponente je realiziran uporabom CSS jezika.

Za dizajn ćemo iskoristiti Bootstrap modul koji sadrži niz već gotovih dizajniranih komponenti kako bi to bilo što je moguće oku ljepše. U web poslužitelju nalaze se naše JSP datoteke.



Slika 4.4: JSP arhitektura

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvata podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Posjetitelj
- Kartica
- PayPal
- Organizator
- Događaj
- Kategorija
- Kategorija i događaj
- Slika
- Video
- Recenzija
- Interes
- Vrsta interesa
- Područje obavijesti
- Kategorija obavijesti
- Područje

4.1.1 Opis tablica

Korisnik Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži atribute: korisnik_id, email, password_hash, full_name te vrsta_korisnika. Ovaj entitet preko user_id u vezi je *One-to-One* s entitetima Posjetitelj, Organizator i Administrator.

Korisnik		
Korisnik_ID	INT	jedinstveni identifikator korisnika
Email	VARCHAR	e-mail adresa korisnika
Password_Hash	VARCHAR	hash lozinke
Full_Name	VARCHAR	ime i prezime korisnika

Korisnik		
Vrsta_korisnika	VARCHAR	razina ovlasti korisnika

Posjetitelj Ovaj entitet sadržava sve važne informacije o posjetitelju. Sadrži atribute: user_id, nick i obavijest_id. Ovaj entitet preko user_id u vezi je *One-to-One* s entitetom Korisnik, *One-to-Many* s entitetom Recenzija, *One-to-Many* s entitetom Interes, *One-to-Many* s entitetom ObavijestPodručje, *One-to-Many* s entitetom ObavijestKategorija.

Posjetitelj		
Korisnik_ID	INT	jedinstveni identifikator korisnika
Nick	VARCHAR	korisničko ime posjetitelja
Obavijest_ID	VARCHAR	jedinstveni identifikator obavijesti

Administrator Ovaj entitet sadržava sve važne informacije o administratoru. Sadrži atribute: user_id i pretplata. Ovaj entitet preko korisnik_id u vezi je *One-to-One* s entitetom Korisnik.

Administrator		
Korisnik_ID	INT	jedinstveni identifikator korisnika
Pretplata	VARCHAR	pretplaćeni događaji

Organizator Ovaj entitet sadržava sve važne informacije o organizatoru. Sadrži atribute: korisnik_id, fizička_adresa, web_adresa, kartica_id, račun_vrijedi_do i paypal_id. Ovaj entitet preko user_id u vezi je *One-to-One* s entitetom Korisnik, *One-to-Many* s entitetom Dogadaj, preko kartica_id i korisnik_id u vezi je *Many-to-One* s entitetom Kartica te preko paypal_id u vezi je *Many-to-One* s entitetom PayPal.

Organizator		
Korisnik_ID	INT	jedinstveni identifikator korisnika
Račun_vrijedi_do	TIMESTAMP	rok trajanja korisničkog računa
Fizicka_adresa	VARCHAR	adresa stanovanja
Web_adresa	VARCHAR	direktna poveznica na vlastitu web stranicu
Kartica_ID	INT	jedinstveni identifikator kreditne kartice
PayPal_ID	INT	jedinstveni identifikator PayPal računa

Kartica Ovaj entitet sadržava sve važne informacije o kartici. Sadrži atribute: kartica_id, korisnik_id i broj_kartice. Ovaj entitet preko kartica_id u vezi je *One-to-Many* s entitetom Organizator.

Kartica		
Kartica_ID	INT	jedinstveni identifikator kartice
Broj_Kartice	VARCHAR	broj kartice
Korisnik_ID	INT	jedinstveni identifikator korisnika

PayPal Ovaj entitet sadržava sve važne informacije o PayPal-u. Sadrži atribute: paypal_id i paypal_email. Ovaj entitet preko paypal_id u vezi je *One-to-Many* s entitetom Organizator.

PayPal		
PayPal_ID	INT	jedinstveni identifikator PayPal računa
PayPal_Mail	VARCHAR	PayPal adresa

Događaj Ovaj entitet sadržava sve važne informacije o događajima. Sadrži atribute: događaj_id, naziv_događaja, lokacija, područje_id, vrijeme_početka, vrijeme_završetka, kategorija_id, opis_događaja i korisnik_id. Ovaj entitet preko kategorija_id u vezi je *One-to-Many* s entitetom Kategorija, *One-to-Many* s entitetom SlikaDogađaj, *One-to-Many* s entitetom VideoDogađaj, *One-to-Many* s entitetom Recenzija, *One-to-Many* s entitetom Interes te preko korisnik_id *Many-to-One* s entitetom Organizator.

Događaj		
Događaj_ID	INT	jedinstveni identifikator događaja
Naziv_događaja	VARCHAR	naziv događaja
Lokacija	VARCHAR	lokacija događaja
Podrucje_ID	INT	jedinstveni identifikator područja
Vrijeme_početka	TIMESTAMP	vrijeme početka događaja
Vrijeme_završetka	TIMESTAMP	vrijeme završetka događaja
Korisnik_ID	INT	jedinstveni identifikator organizatora
Kategorija_ID	INT	jedinstveni identifikator kategorije
Opis_događaja	Varchar	opis događaja

Kategorija Ovaj entitet sadržava sve važne informacije o događajima. Sadrži atribute: kategorija_id i naziv_kategorije. Ovaj entitet preko kategorija_id u vezi je *One-to-Many* s entitetom Događaj i u vezi *One-to-Many* s entitetom ObavijestKategorija.

Kategorija		
Kategorija_ID	INT	jedinstveni identifikator kategorije
Naziv_Kategorije	VARCHAR	naziv kategorije

Slika Ovaj entitet sadržava sve važne informacije o slikama. Sadrži atribute: slika_id i događaj_id. Ovaj entitet preko dogadaj_id u vezi je *Many-to-One* s entitetom SlikaDogađaj.

Slika		
Slika_ID	INT	jedinstveni identifikator slike
Naziv_slika	VARCHAR	naziv slike

SlikaDogađaj Ovaj entitet sadržava sve važne informacije o slikama. Sadrži atribute: slika_id i događaj_id. Ovaj entitet preko dogadaj_id u vezi je *Many-to-One* s entitetom Događaj te *Many-to-One* vezi s entitetom Slika.

SlikaDogađaj		
Slika_ID	INT	jedinstveni identifikator slike
Dogadaj_ID	INT	jedinstveni identifikator događaja

Video Ovaj entitet sadržava sve važne informacije o videima. Sadrži atribute: video_id i događaj_id. Ovaj entitet preko dogadaj_id u vezi je *Many-to-One* s entitetom VideoDogađaj.

Video		
Video_ID	INT	jedinstveni identifikator videa
Naziv_videa	VARCHAR	naziv videa

VideoDogađaj Ovaj entitet sadržava sve važne informacije o videima. Sadrži atribute: video_id i događaj_id. Ovaj entitet preko dogadaj_id u vezi je *Many-to-One* s entitetom Događaj te *Many-to-One* s entitetom Video.

VideoDogadaj		
Video_ID	INT	jedinstveni identifikator videa
Dogadaj_ID	INT	jedinstveni identifikator događaja

Recenzija Ovaj entitet sadržava sve važne informacije o recenzijama. Sadrži atribute: dogadjaj_id, korisnik_id, tekst, datum_stvaranja i naslov_recenzije. Ovaj entitet preko dogadjaj_id u vezi je *Many-to-One* s entitetom Dogadaj te preko korisnik_id u vezi je *Many-to-One* s entitetom Posjetitelj.

Recenzija		
Dogadaj_ID	INT	jedinstveni identifikator događaja
Korisnik_ID	INT	jedinstveni identifikator korisnika
Tekst	VARCHAR	tekst recenzije
Datum_Stvaranja	TIMESTAP	vrijeme objavljivanja recenzije
Naslov_recenzije	VARCHAR	naslov recenzije

Interes Ovaj entitet sadržava sve važne informacije o interesima. Sadrži atribute: dogadjaj_id, korisnik_id i interes_id. Ovaj entitet preko dogadjaj_id u vezi je *Many-to-One* s entitetom Dogadaj, preko korisnik_id u vezi je *Many-to-One* s entitetom Posjetitelj te preko interes_id u vezi je *Many-to-One* s entitetom VrstaInteresa.

Interes		
Korisnik_ID	INT	jedinstveni identifikator korisnika
Dogadaj_ID	INT	jedinstveni identifikator događaja
Interes_ID	INT	jedinstveni identifikator interesa

Vrsta interesa Ovaj entitet sadržava sve važne informacije o vrstama interesa. Sadrži atribute: interes_id i naziv_interesa. Ovaj entitet preko interes_id u vezi je *One-to-Many* s entitetom Interes.

VrstaInteresa		
Interes_ID	INT	jedinstveni identifikator interesta
Naziv_interesa	VARCHAR	naziv interesa

Obavijest Ovaj entitet sadržava sve važne informacije o obavijestima. Sadrži atribute: obavijest_id i korisnik_id. Ovaj entitet preko dogadjaj_id u vezi je *Many-*

to-One s entitetom Dogadaj.

Obavijest		
Obavijest_ID	INT	jedinstveni identifikator slike
Korisnik_ID	INT	jedinstveni identifikator događaja

ObavijestPodručje Ovaj entitet sadržava sve važne informacije o obavijestima sa različitih područja. Sadrži atribute: korisnik_id i područje_id. Ovaj entitet preko korisnik_id u vezi je *Many-to-One* s entitetom Posjetitelj te preko podrucje_id u vezi je *Many-to-One* s entitetom Podrucje.

ObavijestPodručje		
Korisnik_ID	INT	jedinstveni identifikator korisnika
Podrucje_ID	INT	jedinstveni identifikator područja

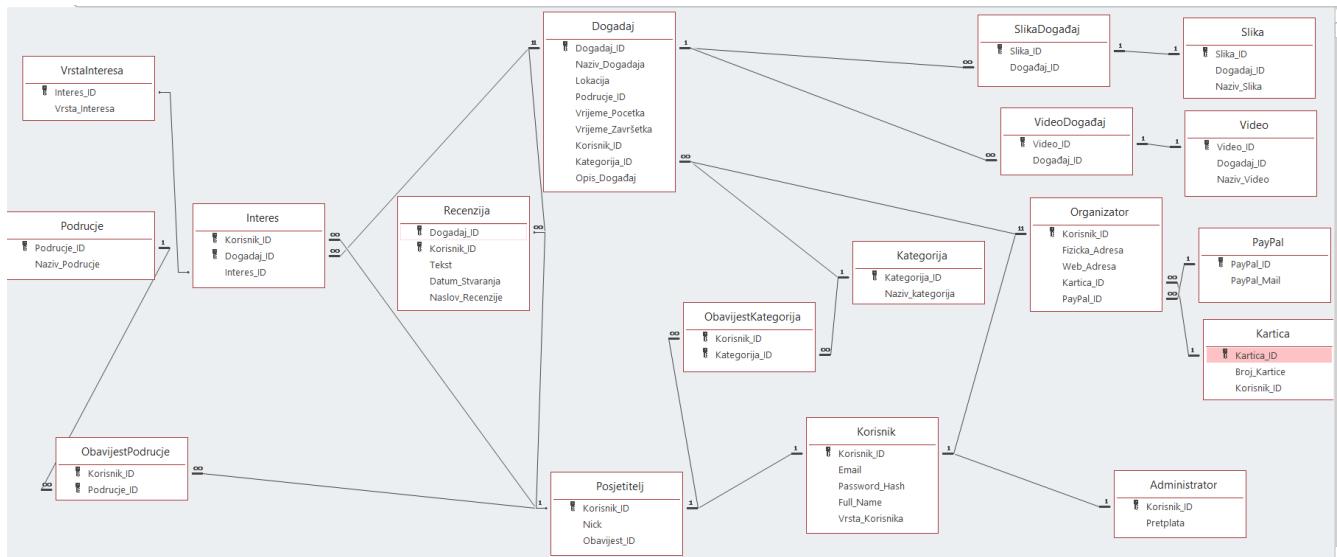
ObavijestKategorija Ovaj entitet sadržava sve važne informacije o obavijestima kategorija. Sadrži atribute: korisnik_id, kategorija_id i obavijest_id. Ovaj entitet preko korisnik_id u vezi je *Many-to-One* s entitetom Posjetitelj te preko kategorija_id u vezi je *Many-to-One* s entitetom Kategorija.

ObavijestKategorija		
Kategorija_ID	INT	jedinstveni identifikator kategorije
Obavijest_ID	INT	jedinstveni identifikator obavijesti

Područje Ovaj entitet sadržava sve važne informacije o područjima. Sadrži atribute: podrucje_id i naziv_podrucje_id. Ovaj entitet preko podrucje_id u vezi je *One-to-Many* s entitetom ObavijestPodručje i u vezi *One-to-Many* s entitetom Dogadaj.

Područje		
Podrucje_ID	INT	jedinstveni identifikator korisnika
Naziv_podrucja	VARCHAR	jedinstveni identifikator područja

4.1.2 Dijagram baze podataka



Slika 4.5: E-R dijagram baze podataka

4.2 Dijagram razreda

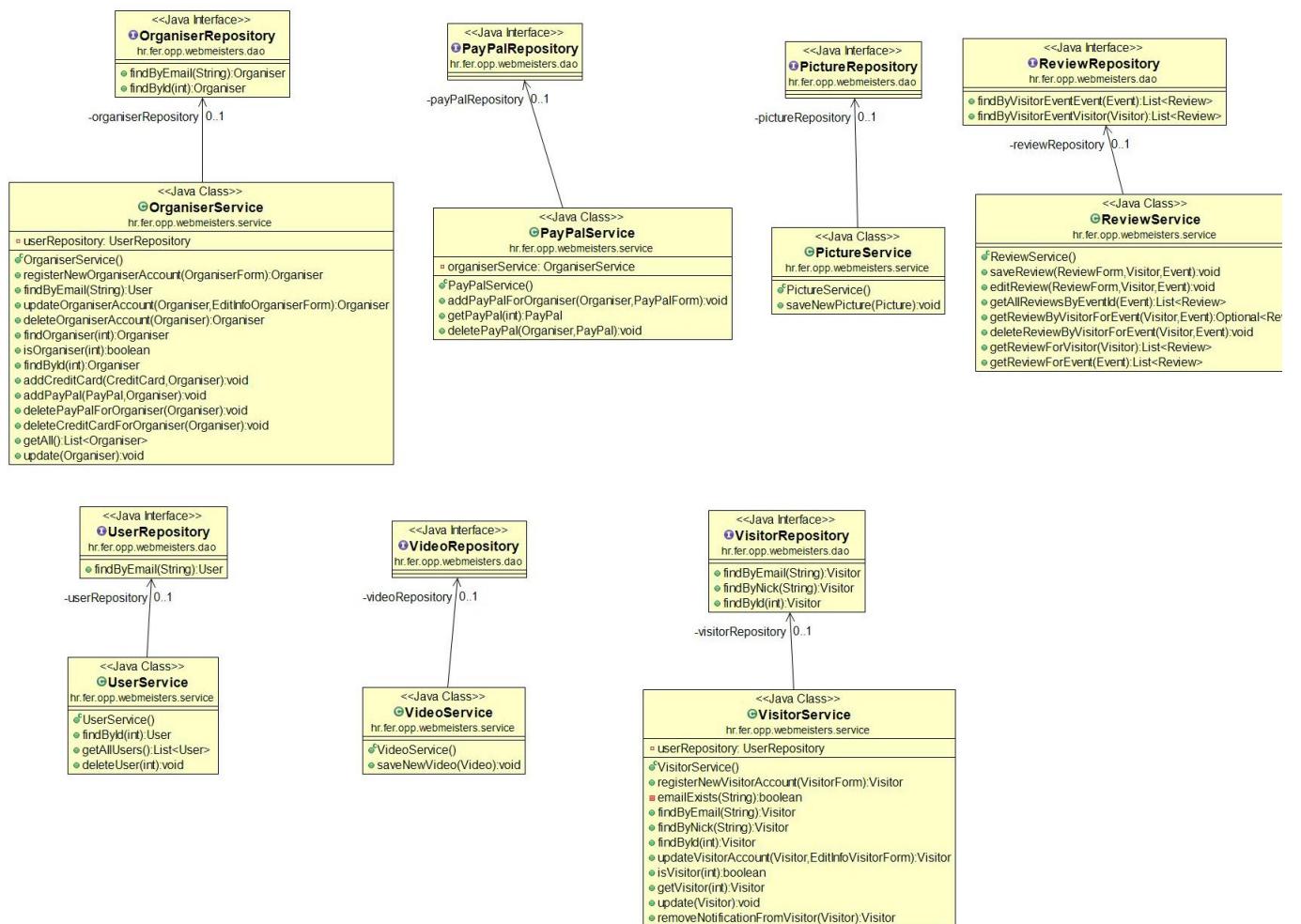
U našem sustavu postoje tri vrste korisnika: Admin, Organizator (Organiser) te Posjetitelj(Visitor).

Svaki od tih korisnika predstavlja jedan objekt koji nasljeđuje objekt User-a. U sustavu postoje događaji (Event) koje može dodavati organizator, postoje recenzije (Review) koje može pisati prijavljeni posjetitelj, postoje interesi(Interest) koje može izraziti posjetitelj za određeni događaj. Kako postoje razne vrste interesa, dodali smo tu InterestType koji nam služi kao jedan od interesa. Uz to na svaki događaj mogu se dodati videi(Video) i slike (Picture) te događaj pripada određenoj kategoriji (Category) i području (Area). Posjetitelji mogu postaviti obavijest (Notification) vezano za više područja te kategorija.

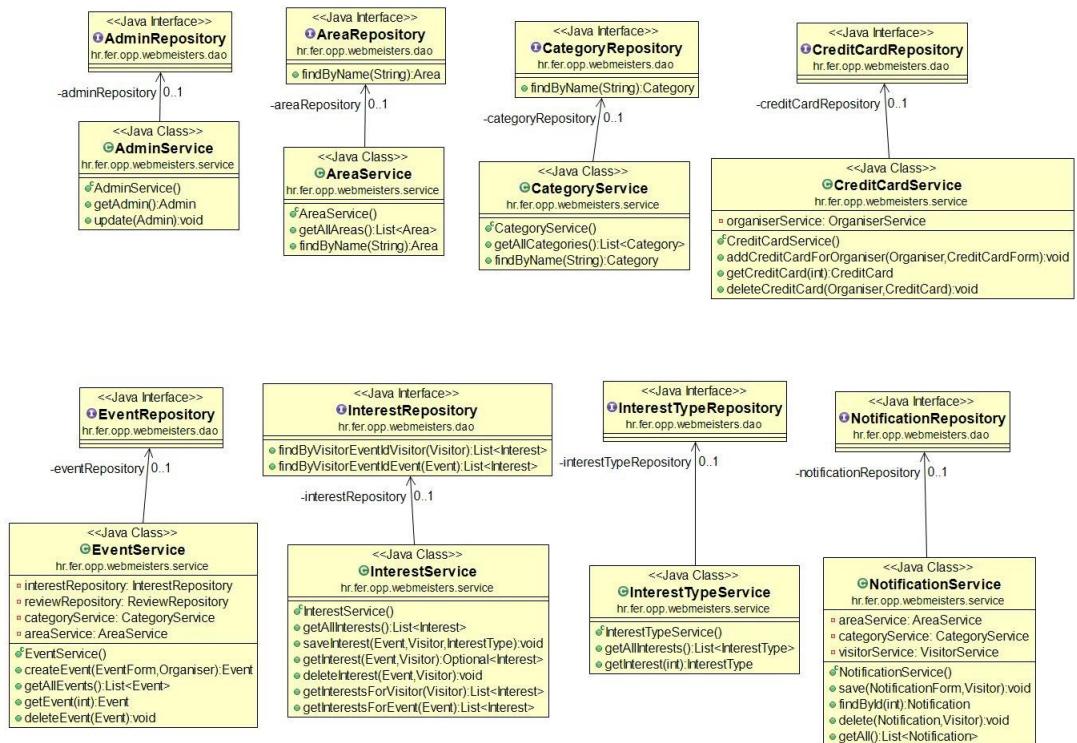
Organizator mora moći platiti članarinu koju postavlja admin, te ima mogućnost plaćanja Paypal-om (PayPal) ili kreditnom karticom(Credit Card).

našoj web aplikaciji poveznicu sa bazom podataka ostvarili smo preko Repositoryja. Svaki objekt ima svoj Repository, a svaki Service ima poveznicu na Repository. Service nam služi kao objekt poveznica između Repositoryja i Controllera tako da imamo između dodatni nivo provjere podataka koji dolaze prema bazi podataka i odlaze prema Controllerima.

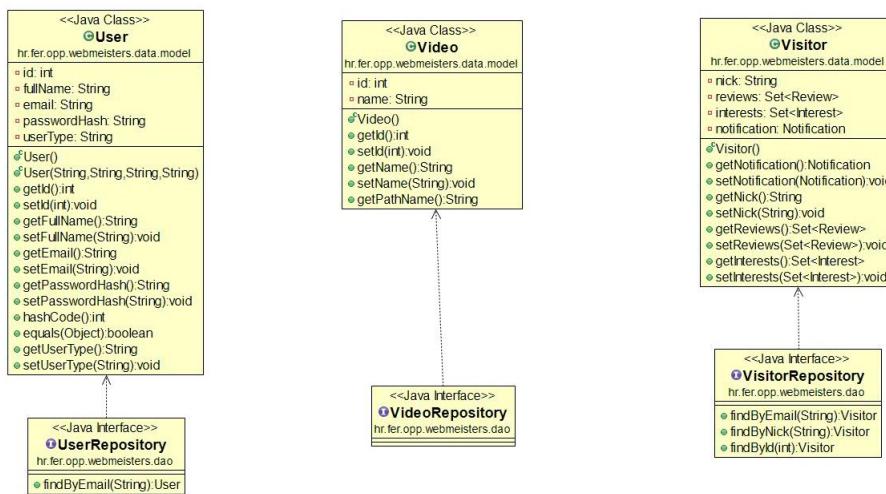
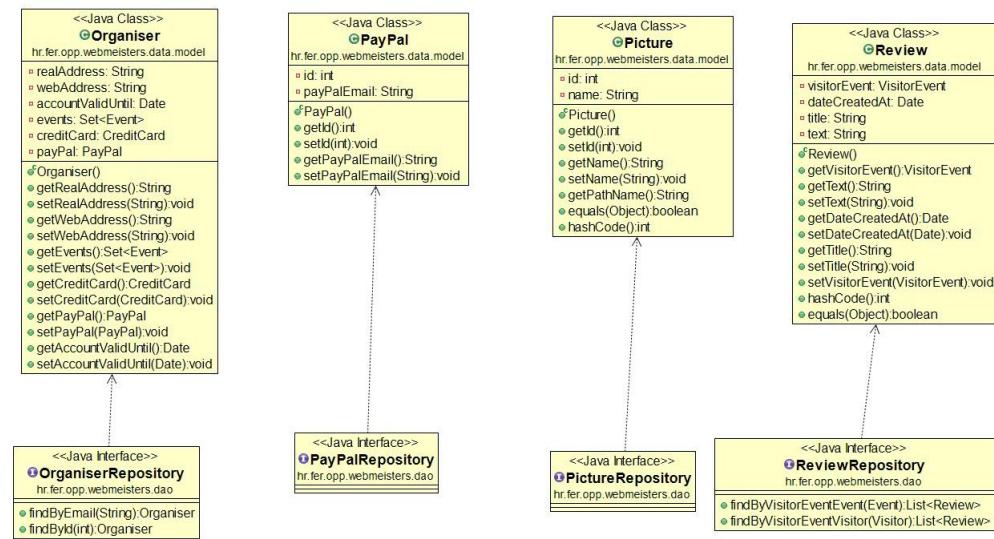
Klijentske zahtjeve primaju Controlleri i oni obavljaju posao koji je potreban kako bi se prikazao sav sadržaj na stranici. U slučaju da je potrebno, oni vrše preusmjeravanje korisnika. Controlleri pripreme ono što će dalje JSP koristiti kako bi izgenerirao HTML sadržaj.



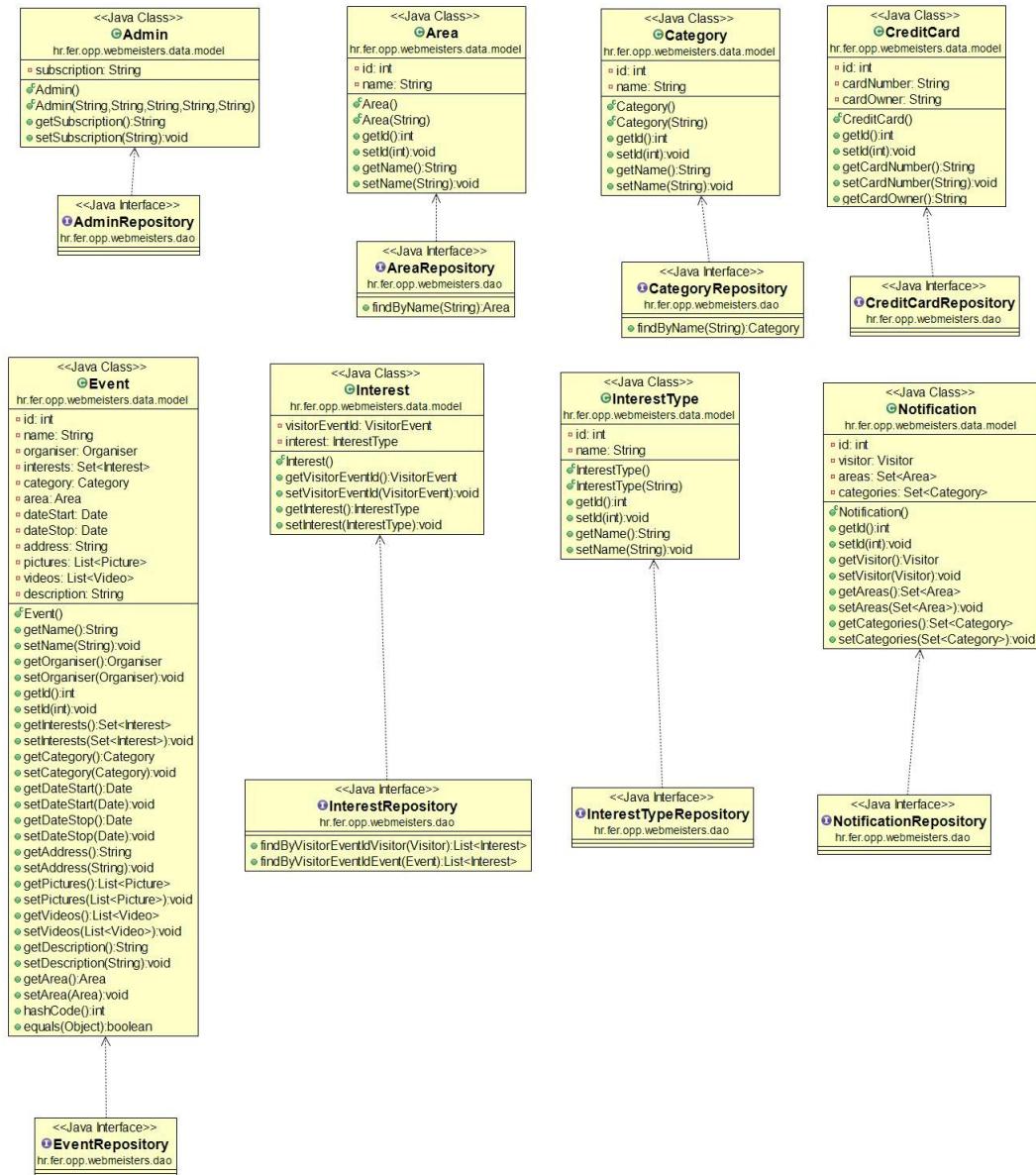
Slika 4.6: Prvi dio slike poveznice Servicea i Repositoryja



Slika 4.7: Drugi dio poveznica Servicea i Repositoryja



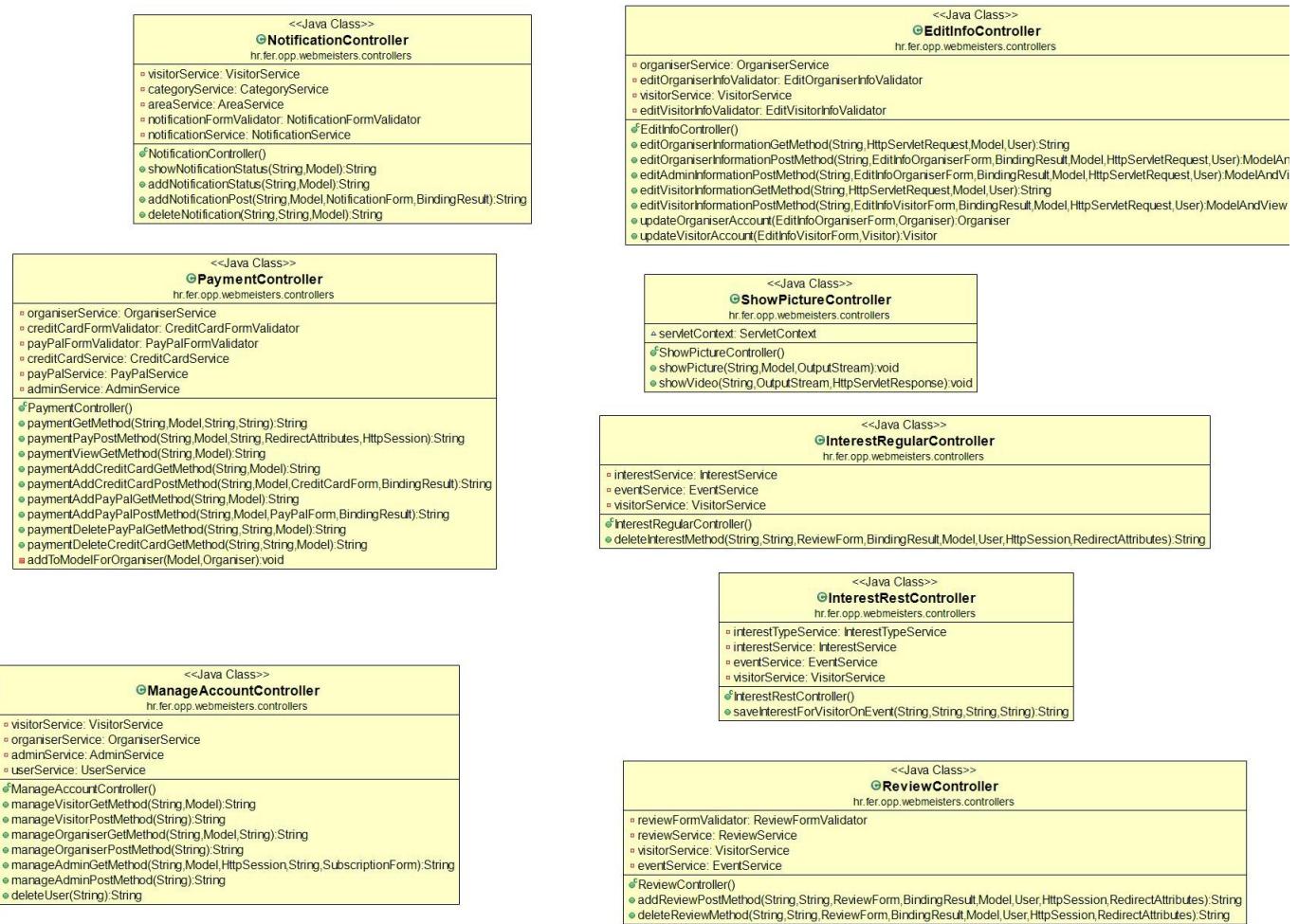
Slika 4.8: Poveznica objekta i Repositoryja



Slika 4.9: Nastavak poveznica objekta i Repositoryja

U nastavkulijedi prikaz svih Controllera koji primaju zahtjeve, obrađuju ih i dalje zovu odgovarajuću JSP datoteku koja generira HTML sadržaj.

Zbog kompleksne situacije nisu prikazane poveznice svakog Controllera sa svim Serviceima.



Slika 4.10: Controlleri prvi dio

NotificationController – služi za dodavanje notifikacija i brisanje notifikacija te prikaz notifikacija ako su postavljene

EditInfoController – služi za uređivanje korisničkih informacija

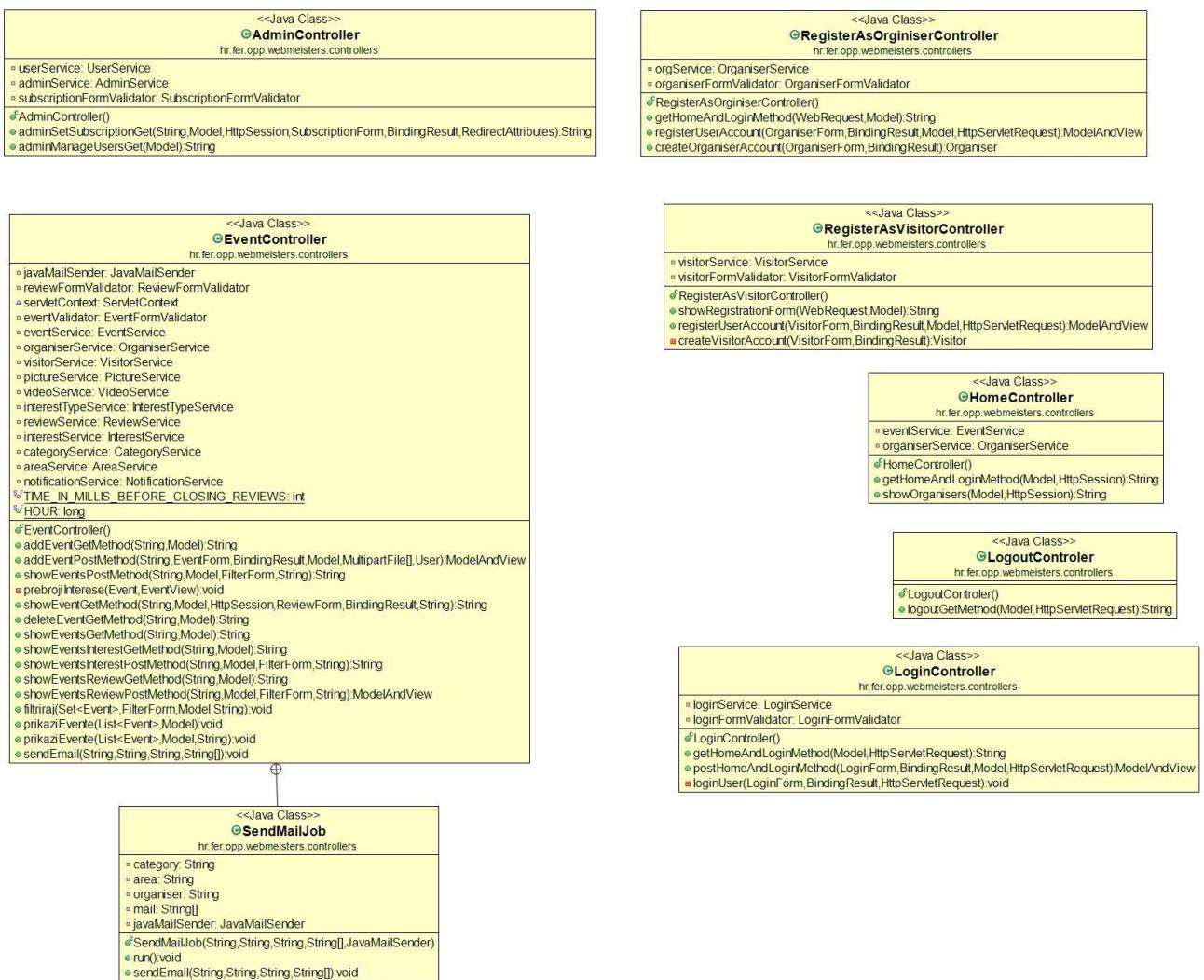
PaymentController – služi za plaćanje, dodavanje i brisanje kreditne kartice te dodavanje i brisanje Paypal računa kod organizatora

ShowPictureController – služi za prikaz fotografija na web stranici

InterestRegularController, InterestRestController – dodavanje, brisanje interesa kod događaja

ManageAccountController – služi za prikaz informacija svih korisnika

ReviewController – služi za dodavanje recenzija



Slika 4.11: Controlleri drugi dio

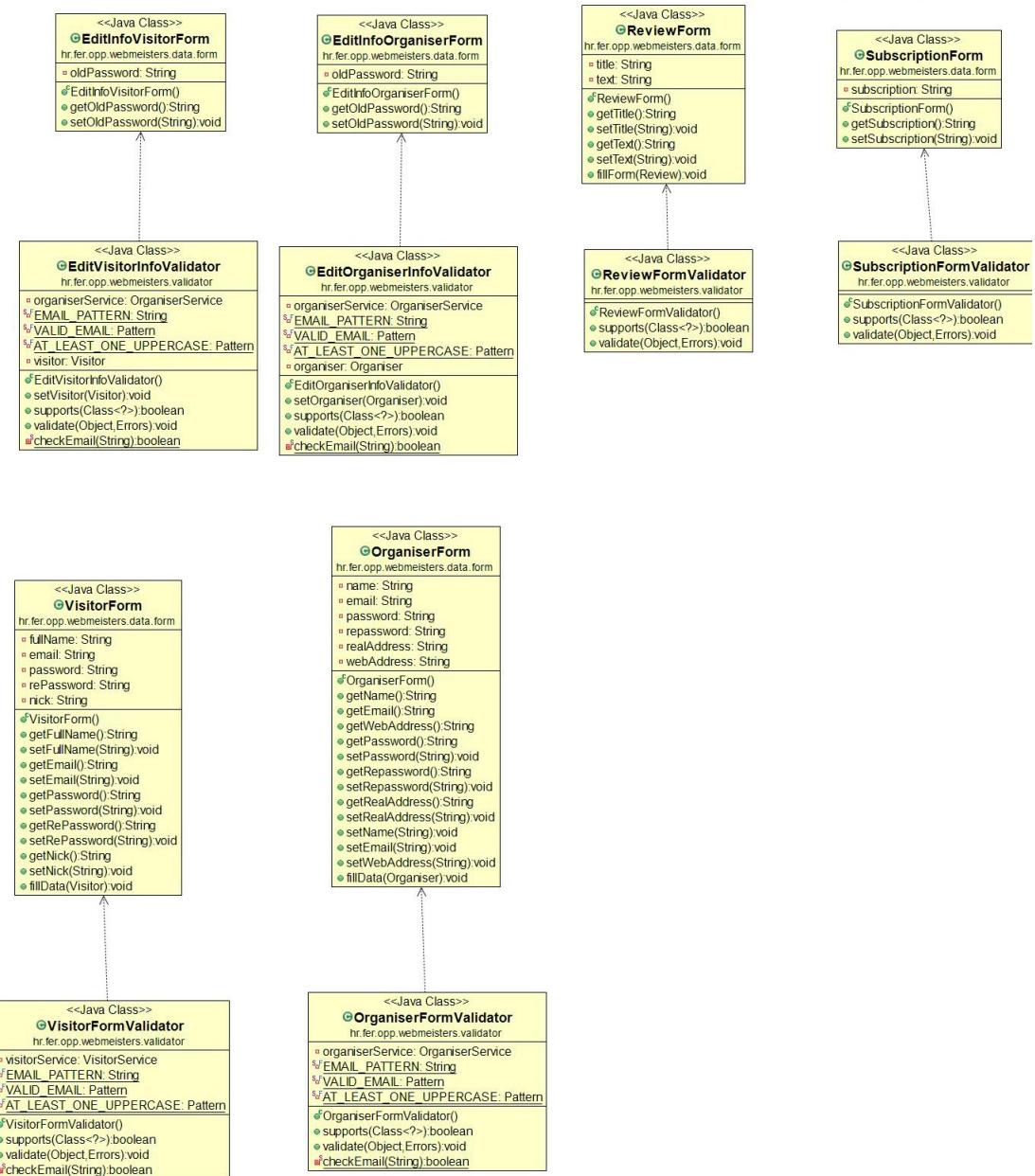
AdminController – služi svim adminovim funkcionalnostima

RegisterAsOrganiser, RegisterAsVisitor – registracija korisnika

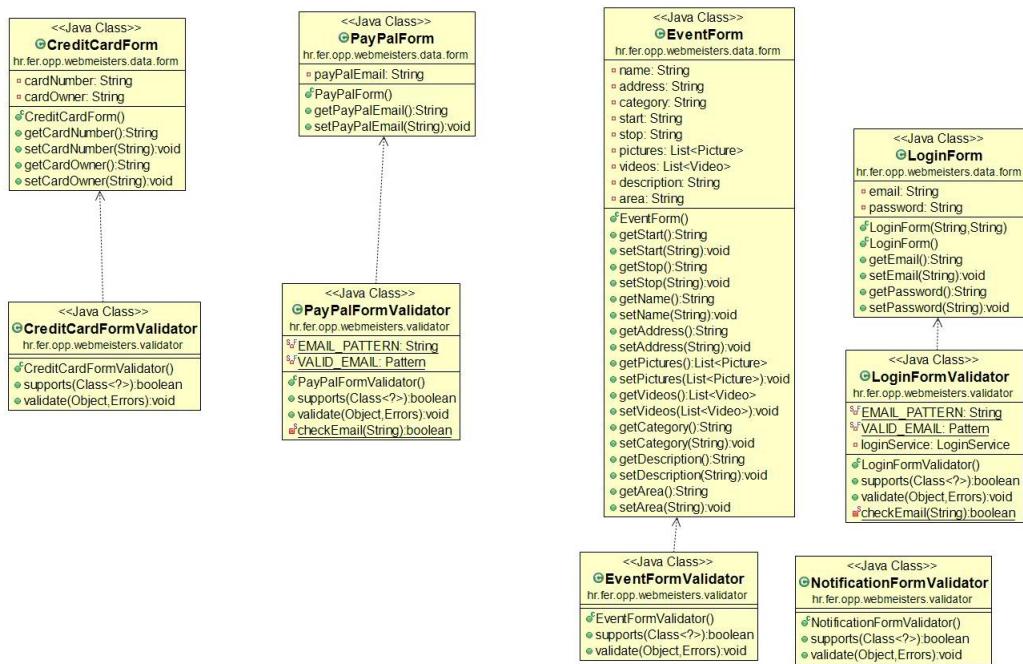
Login, LogoutController – prijava i odjava korisnika

EventController – služi za dodavanja i brisanje događaja, filtriranje događaja, slanje maila kod dodavanja događaja ako postoje postavke obavijesti za nekog korisnika, prikazivanje jednog događaja ili više njih

Kako bi bili sigurni da je svaki formular ispravno ispunjen, to provjeravamo na način pošaljemo Formu koja se ispuni, a odgovarajući Validator provjeri je li to dobro ispunjeno. Tek kad je ispunjeno kako treba biti, upišemo podatke u bazu.



Slika 4.12: Validatori i Forme prvi dio



Slika 4.13: Validatori i Forme drugi dio

4.3 Dijagram stanja

Dijagram stanja se koristi kao grafički prikaz prijelaz iz jednog stanja u drugo na temelju događaja.

Na slici ?? prikazan je dijagram stanja registriranog organizatora.

Nakon prijave, organizatoru se prikazuje početna stranica sa aktualnim događajima i notifikacija o plaćanju članarine ovisno jeli ona plaćena ili nije sa poveznicom koja ga vodi do plaćanja članarine.

Na vrhu web stranice nalazi se traka sa tri gumba, "Naslovница", "Ime profila organizatora" I "Odjavi se", koji su vidljivi tokom cijelog rada na web aplikaciji.

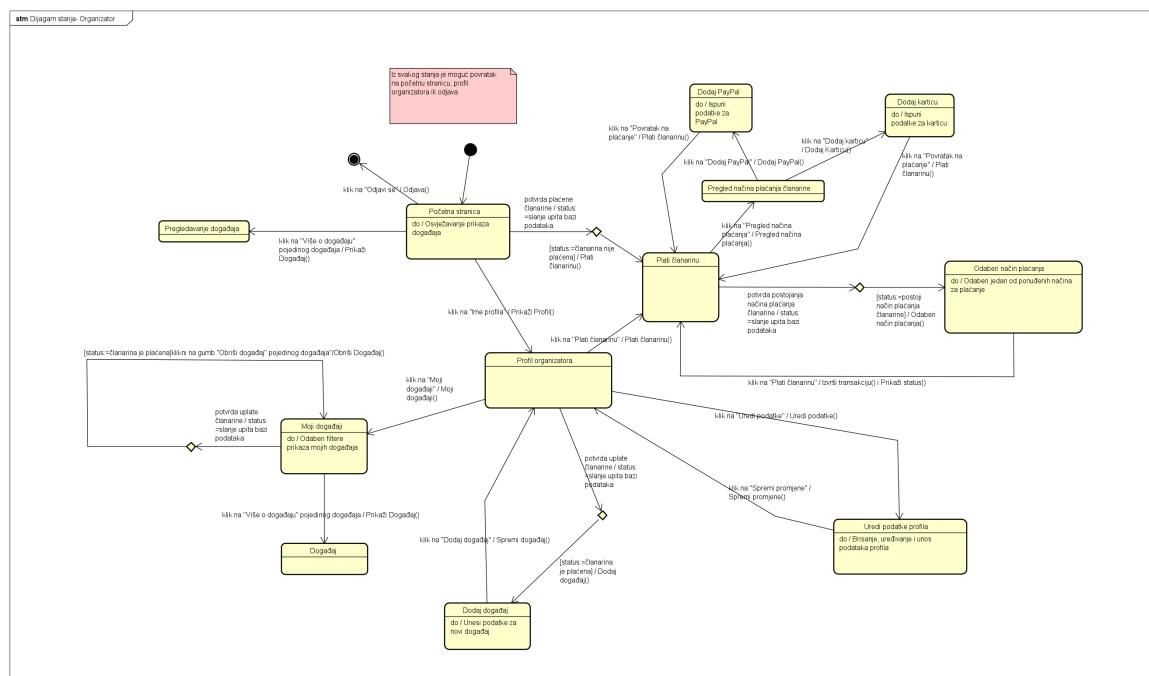
Organizator pored svakog događaja može kliknuti na gumb "Više o događaju" koji ga vodi na stranicu tog događaja sa dodatnim informacijama.

Klikanjem gumba "Ime profila organizatora" otvara se profil organizatora sa svim informacijama. Organizator ima opciju uređivanja podataka profila i pregledavanja vlastitih događaja.

Također ima opciju i plaćanja članarine kojom se otvara nova stranica na kojoj unosi načine plaćanja, Paypal ili karticu, te odabire način plaćanja i plaća članarinu.

Ukoliko je članarina plaćena organizatoru se prikazuje gumb "Dodaj događaj" kojim može dodavati nove događaje što uključuje dodavanje opisa događaja, odabir kategorije događaja, mjesto održavanja te različiti medijski sadržaji uključujući video zapise i slike.

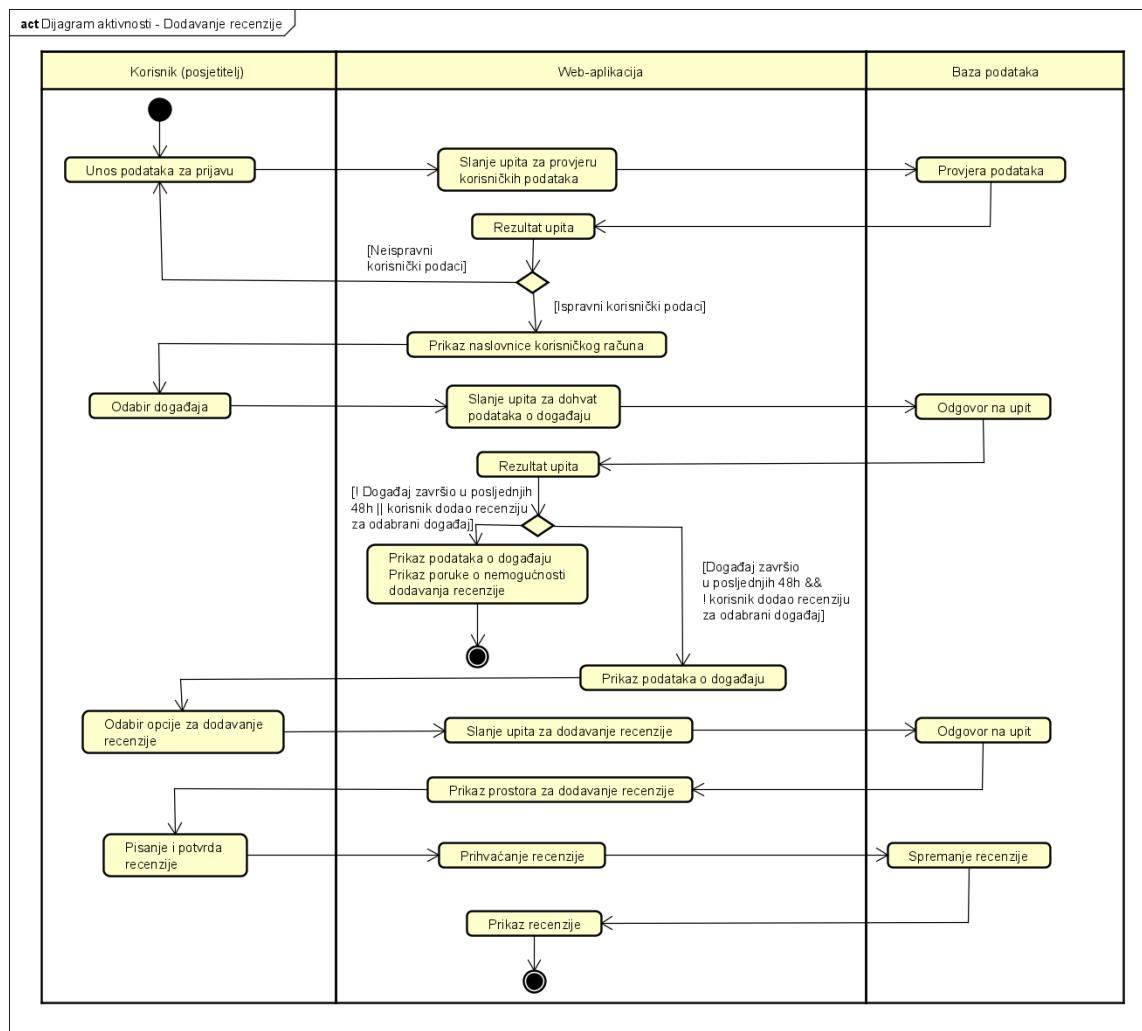
Također ako je članarina plaćena organizator u pregledavanju vlastitih događaja ima opciju i brisanja događaja.



Slika 4.14: Dijagram stanja - Organizator

4.4 Dijagram aktivnosti

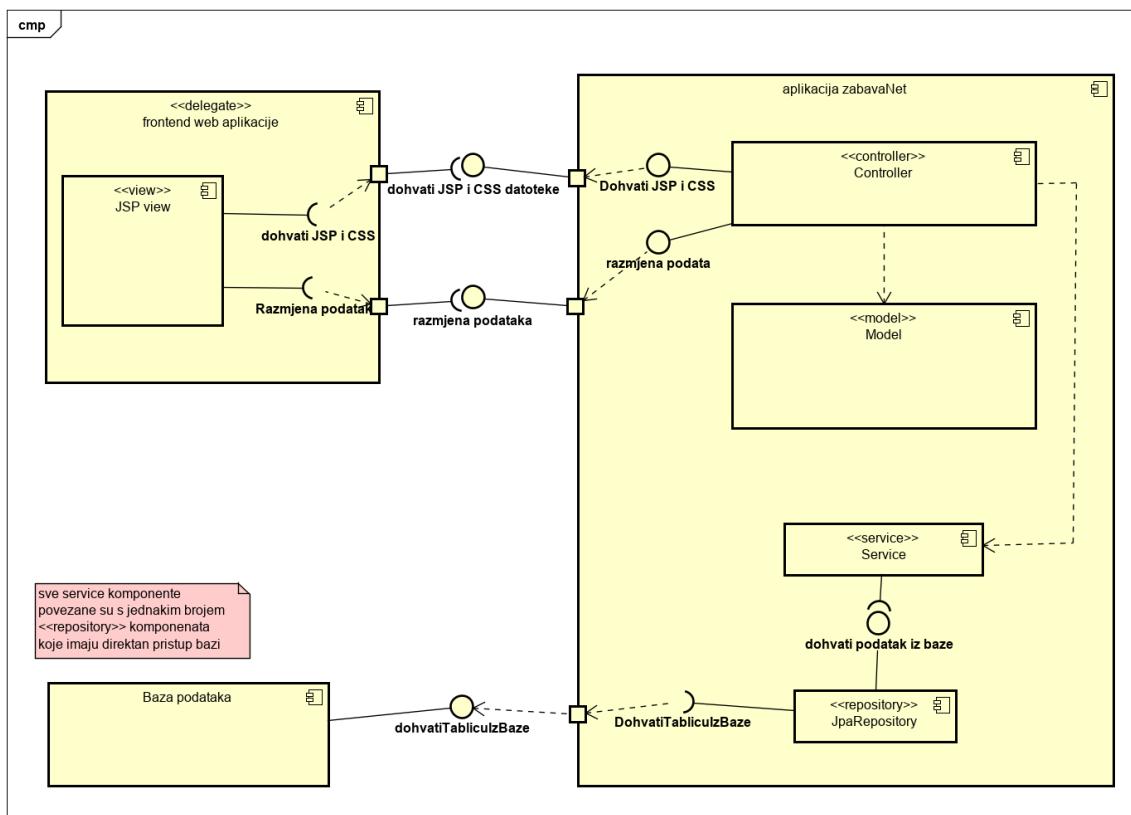
Dijagram na slici prikazuje slijedne aktivnosti koje rezultiraju dodavanjem recenzije za neko događanje. Preduvjet za dodavanje recenzije je taj da prijavljeni korisnik bude posjetitelj. Nakon što se korisnik prijavi kao posjetitelj prikaže mu se početna stranica na kojoj se nalazi popis događanja. Potom posjetitelj odabere događanje za koje želi dodati svoju recenziju te, ukoliko je događanje završilo unutar posljednjih 48h i posjetitelj nije već dodao svoju recenziju za njega, prilikom prikaza detalja o događanju posjetitelju će biti prikazana i opcija dodavanja recenzije. Ukoliko barem jedan od ta dva uvjeta nije zadovoljen posjetitelju neće biti prikazana opcija za dodavanje recenzije već će mu pisati prikladna poruka o nemogućnosti recenziranja. Po završetku pisanja teksta svoje recenzije posjetitelj odabirom opcije za spremanje označi kraj željene akcije te se njegova recenzija pošrani u bazu podataka i prikaže na ekranu.



Slika 4.15: Dijagram aktivnosti - Dodavanje recenzije

4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici 4.9 opisuje organizaciju komponenti, međuvisnosti i odnose prema okolini. Sustavu se pristupa preko dva sučelja. Preko sučelja za dohvati JSP datoteka u kojima je html kod i CSS datoteka se poslužuju datoteke koje pripadaju frontend dijelu aplikacije. Korisnik komunicira sa web aplikacijom pomoću JSP view komponenata te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke. Controller komponente u REST sloju komuniciraju sa Service komponentama u REST sloju, a zadaća Service komponenti jest usklađivanje i „uljepšavanje“ podataka iz baze (SQL upiti) s podacima iz Controllera (forme modela). Za dohvati podataka iz baze koristi se JpaRepository koji se spaja na bazu.



Slika 4.16: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Aplikacija je napisana kao Maven Projekt. Maven je alat korišten primarno za java projekte zbog lakšeg i preglednijeg upravljanja projektom. Backend dio aplikacije pisan je u jeziku Java i korišten je radni okvir Spring Boot dok je frontend pisan pomoću Bootstrap radnog okvira. Spring Boot omogućava brži i efikasniji pristup izgradnje Spring aplikacije i Spring web aplikacije. Pruža široki spektar opcija te olakšava posao korisniku tako da se ne treba brinuti o raznim implementacijskim detaljima.

Za razvojno okruženje korišten je IntelliJ Idea Ultimate. IntelliJ je razvojno okruženje koje je razvila tvrtka JetBrains i služi za razvijanje i pisanje računalnog softvera u programskom jeziku javi.

Kao web poslužitelj uzet je Apache Tomcat, a za rad s bazom podataka koristio se H2 Database. Apache Tomcat je open source web poslužitelj koji implementira nekoliko Java EE specifikacija. Također je korišten iz razloga što je direktno ugrađen u Spring Boot framework. H2 je open source baza podataka koja omogućuje korištenje tako da podaci ne ostaju na disku.

Komunikacija u timu realizirana je pomoću aplikacije Slack i aplikacije WhatsApp. Sustav za upravljanje kodom korišten je Git a repozitorij projekta je dostupan na web platformi GitLab.

<https://maven.apache.org/>
<https://spring.io/projects/spring-boot>
<https://www.oracle.com/java/technologies/>
<https://getbootstrap.com/>
<http://tomcat.apache.org/>
<https://www.h2database.com/html/main.html>
<https://slack.com/intl/en-hr/>
<https://web.whatsapp.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Testovi za ispitivanje pojedinih komponenti backend dijela aplikacije izvršeni su pomoću JUnit alata. Cilj ovih testova je provjera ispravnosti rada u programu za osnovne funkcionalnosti sustava, te za lakše otkrivanje pogrešaka kod nekih nespecifičnih slučajeva.

Jedan od testiranih komponenti je dodavanje događaja od strane organizatora. Sustav svaki pokušaj unosa događaja provjerava pomoću validatora. Validator metodom „validate“ prima EventForm objekt koji je ispunjen informacijama unesenih od strane organizatora i BindingResults pomoću kojeg vraća greške. Na slici 5.1 prikazano je dva testa, prvi simulira ispravan unos informacija za događaj a drugi neispravan tako što fali ime i opis događaja.

```
@Test
void ispravnUnosEventa(){
    EventForm form = new EventForm();
    // ispravna inicializacija informacija o eventu
    form.setName("party");
    form.setAddress("ulica 9");
    form.setArea("Zagreb");
    form.setCategory("Diskoteka");
    form.setDescription("bit će super, dodi i ti");
    form.setStart("01-04-2020 17:44");
    form.setStop("01-05-2020 17:44");
    EventFormValidator validator = new EventFormValidator();
    validator.validate(form, results);
    assertFalse(results.hasErrors());
}

@Test
void dodavanjeEventaBezImenaIOpisa(){
    EventForm form = new EventForm();
    // neispravna inicializacija informacija o eventu
    form.setAddress("ulica 9");
    form.setArea("Zagreb");
    form.setCategory("Diskoteka");
    form.setStart("01-04-2020 17:44");
    form.setStop("01-05-2020 17:44");
    EventFormValidator validator = new EventFormValidator();
    validator.validate(form, results);
    assertTrue(results.hasErrors());
    assertEquals( expected: 2, results.getErrorCount());
}
```

Slika 5.1: Ispitivanje ispravnog i neispravnog unosa događaja

Osim ne upisanih informacija o događaju, sustav provjerava jesu li te informacije smislene. Npr na slici 5.2 prikazana su dva testa kod kojih su krivo uneseni datumi početka i kraja događaja. Prvi test provjerava baca li validator pogrešku ako je unesen datum završetka događaja prije datuma početka. Drugi test provjejava također jeli validator bacio pogrešku kada je datum početka događaja prije današnjeg datuma.

```
@Test
void dodavanjeEventaNaseljenoUlica9() {
    EventForm form = new EventForm();
    form.setName("party");
    form.setAddress("ulica 9");
    form.setArea("Zagreb");
    form.setCategory("Diskoteka");
    form.setDescription("bit će super dodijeli ti");
    form.setStart("01-05-2020 17:44");
    form.setStop("01-04-2020 17:44");
    EventFormValidator validator = new EventFormValidator();
    validator.validate(form, results);
    assertEquals( expected: true, results.hasErrors());
    assertEquals( expected: 1,results.getErrorCount());
}

@Test
void dodavanjeEventaNaseljenoUlica9() {

    EventForm form = new EventForm();
    form.setName("party");
    form.setAddress("ulica 9");
    form.setArea("Zagreb");
    form.setCategory("Diskoteka");
    form.setDescription("bit će super dodijeli ti");
    form.setStart("10-01-2019 23:09");
    form.setStop("10-01-2020 17:44");
    EventFormValidator validator = new EventFormValidator();
    validator.validate(form, results);
    assertEquals( expected: true, results.hasErrors());
    assertEquals( expected: 1,results.getErrorCount());
}
```

Slika 5.2: Ispitivanje pogrešnog unosa datuma

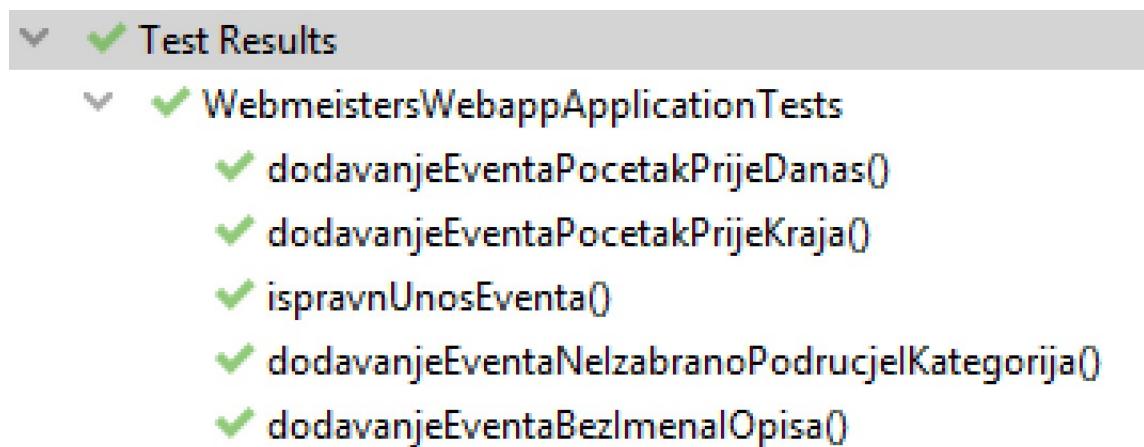
Uz ove već navedene informacije, korisnik također mora odabrati kategoriju i mjesto događaja pomoću „drop down“ liste. Ako je korisnik zaboravio odabrati primjerice kategoriju, u EventForm objekt će biti zapisano „Izaberite kategoriju“. Validator to provjerava i javlja grešku. Zadnji test na slici 5.3 ispituje jeli korisnik izabrao kategoriju i mjesto događaja. U ovom slučaju se simulira kao da korisnik nije naveo ni jedno ni drugo tako da validator baca dvije pogreške što se može

iščitati iz BindingResults objekta.

```
@Test  
void dodavanjeEventaNelzabranopodručjelKategorija(){  
    EventForm form = new EventForm();  
    form.setName("party");  
    form.setAddress("ulica 9");  
    form.setArea("Izaberipodručje");  
    form.setCategory("Izaberikategoriju");  
    form.setDescription("bit će super dođi i ti");  
    form.setStart("10-01-2020 23:09");  
    form.setStop("01-05-2020 17:44");  
    EventFormValidator validator = new EventFormValidator();  
    validator.validate(form, results);  
    assertEquals(expected: true, results.hasErrors());  
    assertEquals(expected: 2, results.getErrorCount());  
}
```

Slika 5.3: Ispitivanje kategorije i mesta događaja

Slika 5.4 pokazuje ispravni rezultat svih testova što znači da je validator dobro implementiran.



Slika 5.4: Rezultati testiranja

5.2.2 Ispitivanje sustava

Svi testovi su izvršeni preko Seleniuma i testiraju se neke funkcionalosti organizatora.

Na slici 5.5 nalazi se test dodavanja događaja kod organizatora. Također na slikama 5.6 i 5.7 testiraju se notifikacije tijekom izostajanja unosa podataka kod dodavanja događaja i krivim postavljanjem datuma.

```
/*
 * Dodavanje novog događaja
 */
@Test
public void addEvent(){
    this.driver.findElement(By.xpath("//a[.='Dodaj događaj']")).click();
    // input box za unos naziva događaja
    this.driver.findElement(By.id("name")).sendKeys("SELENIUM EVENT");
    //inputbox za unos adrese događaja
    this.driver.findElement(By.id("address")).sendKeys("SELENIUM ADDRESS");
    //početak događaja, current timestamp + 1 dan
    Date date= new Date();
    Calendar c1 = Calendar.getInstance();
    c1.setTime(date);
    c1.add(Calendar.DATE, 1);
    Date currentDatePlusOne = c1.getTime();
    SimpleDateFormat sdf= new SimpleDateFormat( pattern: "dd-MM-yyyy h:mm");
    String formatedDate = sdf.format(currentDatePlusOne);
    this.driver.findElement(By.id("start")).sendKeys(formatedDate);
    //kraj događaja, current timestamp + 2 dana
    Calendar c = Calendar.getInstance();
    c.setTime(date);
    c.add(Calendar.DATE, 2);
    Date currentDatePlusTwo = c.getTime();
    WebElement stop = this.driver.findElement(By.id("stop"));
    SimpleDateFormat sdf2= new SimpleDateFormat( pattern: "dd-MM-yyyy HH:mm");
    String formatedDate2 = sdf2.format(currentDatePlusTwo);
    stop.sendKeys(formatedDate2);
    //izabrane kategorije događaja: Diskoteka
    this.driver.findElement(By.xpath("//select[@name='category']")).click();
    this.driver.findElement(By.xpath("//option[@value='Diskoteka '])).click();
    //izabrane mesta događaja, Zagreb
    this.driver.findElement(By.xpath("//select[@name='area']")).click();
    this.driver.findElement(By.xpath("//option[@value='Zagreb '])).click();
    //unos opisa za događaj u input
    this.driver.findElement(By.id("description")).sendKeys("EDITED BY SELENIUM");
    //Klikanje dodavanja događaja
    this.driver.findElement(By.xpath("//input[@value='Dodaj događaj']")).click();
    //provjera redirekcije u slučaju ispravnog unosa događaja
    Assert.assertTrue(this.driver.getCurrentUrl().contains("http://zabavonet.herokuapp.com/showEvents/"));
}
```

Slika 5.5: Dodavanje novog događaja

```
/**
 * Provjeravanje prikaza svih errora kod ne unosa dodavanja dogadaja
 */
@Test
public void addEventErrorsNoInput(){
    this.driver.findElement(By.xpath("//a[.='Dodaj dogadaj']")).click();
    //klikanje gumba dodaj dogadaj dok nije uneseno
    this.driver.findElement(By.xpath("//input[@value='Dodaj dogadaj']")).click();
    //error nije unesen opis
    WebElement descriptionError = this.driver.findElement(By.id("description.errors"));
    //error nije uneseno ime
    WebElement nameError = this.driver.findElement(By.id("name.errors"));
    //error nije unesena adresa
    WebElement addressError = this.driver.findElement(By.id("address.errors"));
    //error nije unesen pocetak
    WebElement beginError = this.driver.findElement(By.id("start.errors"));
    //error nije unesen kraj
    WebElement stopError = this.driver.findElement(By.id("stop.errors"));
    //error nije unesena kategorija
    WebElement categoryError = this.driver.findElement(By.id("category.errors"));
    //error nije unesen podrucje
    WebElement areaError = this.driver.findElement(By.id("area.errors"));
    //provera display svih errora
    Assert.assertTrue( condition: descriptionError.isDisplayed() && nameError.isDisplayed() &&
        addressError.isDisplayed() && beginError.isDisplayed() && stopError.isDisplayed() &&
        categoryError.isDisplayed() && areaError.isDisplayed());
}
}
```

Slika 5.6: Provjeravanje prikaza svih errora kod ne unosa dodavanja događja

```
/**
 * Prikaz errora kad je vrijeme završetka prije nego vrijeme početka
 */
@Test
public void addEventWrongDateEnding(){
    this.driver.findElement(By.xpath("//a[.='Dodaj dogadaj']")).click();
    //ispunjavanje ostalih polja
    this.driver.findElement(By.id("name")).sendKeys( ...charSequences: "SELENIUM EVENT");
    this.driver.findElement(By.id("address")).sendKeys( ...charSequences: "SELENIUM ADDRESS");
    this.driver.findElement(By.xpath("//select[@name='category_1']")).click();
    this.driver.findElement(By.xpath("//option[@value='Diskoteka']")).click();
    this.driver.findElement(By.xpath("//select[@name='area']")).click();
    this.driver.findElement(By.xpath("//option[@value='Zagreb']")).click();
    this.driver.findElement(By.id("description")).sendKeys( ...charSequences: "EDITED BY SELENIUM");
    //početak dogadaja current timestamp + 1 dan
    Date date= new Date();
    Calendar c1 = Calendar.getInstance();
    c1.setTime(date);
    c1.add(Calendar.DATE, ii: 1);
    Date currentDatePlusOne = c1.getTime();
    SimpleDateFormat sdf= new SimpleDateFormat( pattern: "dd-MM-yyyy h:mm");
    String formatedDate = sdf.format(currentDatePlusOne);
    this.driver.findElement(By.id("start")).sendKeys(formatedDate);
    //kraj dogadaja current timestamp
    Date currentDate = new Date();
    SimpleDateFormat sdf2= new SimpleDateFormat( pattern: "dd-MM-yyyy HH:mm");
    String formatedDate2 = sdf2.format(currentDate);
    this.driver.findElement(By.id("stop")).sendKeys(formatedDate2);
    //klikanje dodavanja dogadaja
    this.driver.findElement(By.xpath("//input[@value='Dodaj dogadaj']")).click();
    //provjera errora
    Assert.assertTrue(this.driver.findElement(By.xpath("//span[.='Datum zavrsetka ne moze biti prije datuma pocetka dogadaja']"))
        .isDisplayed());
}
}
```

Slika 5.7: Prikaz errora kada je vrijeme završetka prije noego vrijeme početka

Na slici 5.8 testira se funkcionalost promjene imena organizatora.

```
/** 
 * Uređivanje korisničkog imena
 */
@Test
public void changeName(){
    this.driver.findElement(By.xpath("//a[starts-with(@href, '/editInfo')]")).click();
    this.driver.findElement(By.xpath("//div[@class='row']//div[1]/div[1]/input[1]")).sendKeys(" EDITED");
    this.driver.findElement(By.xpath("//input[@value='Spremi promjene']")).click();
    String editedName = this.driver.findElement(By.xpath("//body/div[@class='container']/div[@class='row']/div[@class='col-md-6 polja']/div[1]")).getText()
        .split( regex: "\n" )[1];
    Assert.assertTrue(editedName.contains("EDITED"));
}
```

Slika 5.8: Uređivanje korisničkog imena

Na slici 5.10 nalazi se metoda kojom se ulogirava u organizatorov račun, koja se poziva svaki put u metodi sa slike 5.11 prije svakog testa.

```
/** 
 * Logiranje na stranicu kao organizator
 */
@Test
public void loginOrganizer(){
    this.logMeIn();
    Assert.assertEquals(this.driver.getCurrentUrl(), "http://zabavanet.herokuapp.com/home");
}
```

Slika 5.9: Registracija kao organizator

```
public void logMeIn() {
    //this.driver = new PhantomJSdriver();
    System.setProperty("webdriver.gecko.driver", "/home/petar/Desktop/geckodriver");
    this.driver = new FirefoxDriver();
    this.driver.manage().timeouts().implicitlyWait(DEFAULT_TIMEOUT_TIME, TimeUnit.SECONDS);
    this.driver.get(this.baseUrl);
    WebElement loginButton = this.driver.findElement(By.xpath("//a[.='Prijavi se']"));
    loginButton.click();

    this.userNameInput = this.driver.findElement(By.id("email"));
    this.userNameInput.sendKeys(this.userName);

    this.passwordInput = this.driver.findElement(By.xpath("//input[@type='password']"));
    this.passwordInput.sendKeys(this.password);

    this.driver.findElement(By.xpath("//input[@type='submit']")).click();
}
```

Slika 5.10: Metoda pri registraciji

```
@BeforeMethod
public void loginVisitProfile(){
    this.logMeIn();
    //klik na moj profil
    this.driver.findElement(By.xpath("//a[starts-with(@href, '/manageAccount')]")).click();
}
```

Slika 5.11: Registracija

Na slici 5.12 nalaze se rezultati testiranja.

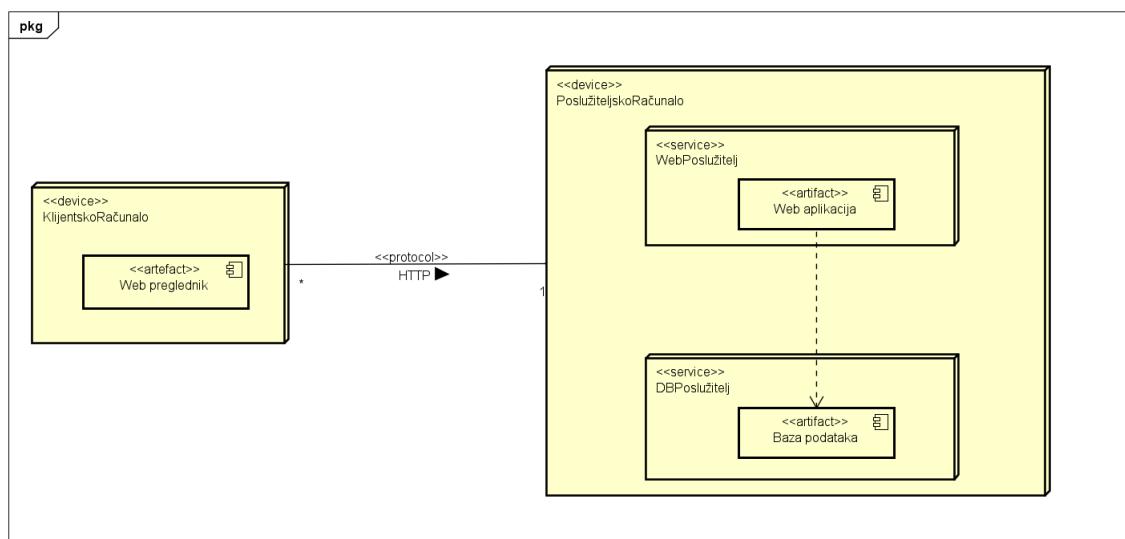
✓ addEvent	3 s 169 ms
✓ addEventErrorsNoInput	3 s 281 ms
✓ addEventWrongDateEnding	2 s 510 ms
✓ changeName	610 ms

Slika 5.12: Rezultati testiranja

5.3 Dijagram razmještaja

Dijagram razmještaja prikazuje generalnu topologiju sustava koji se koristi kako bi resursi bili optimalno raspoređeni. Predstavlja statički pogled na razmještaj sklopoških i programskih komponenata.

Na poslužiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka. Klijenti koriste web preglednik kako bi pristupili web aplikaciji te se komunikacija između računala korisnika i poslužitelja odvija preko HTTP veze.

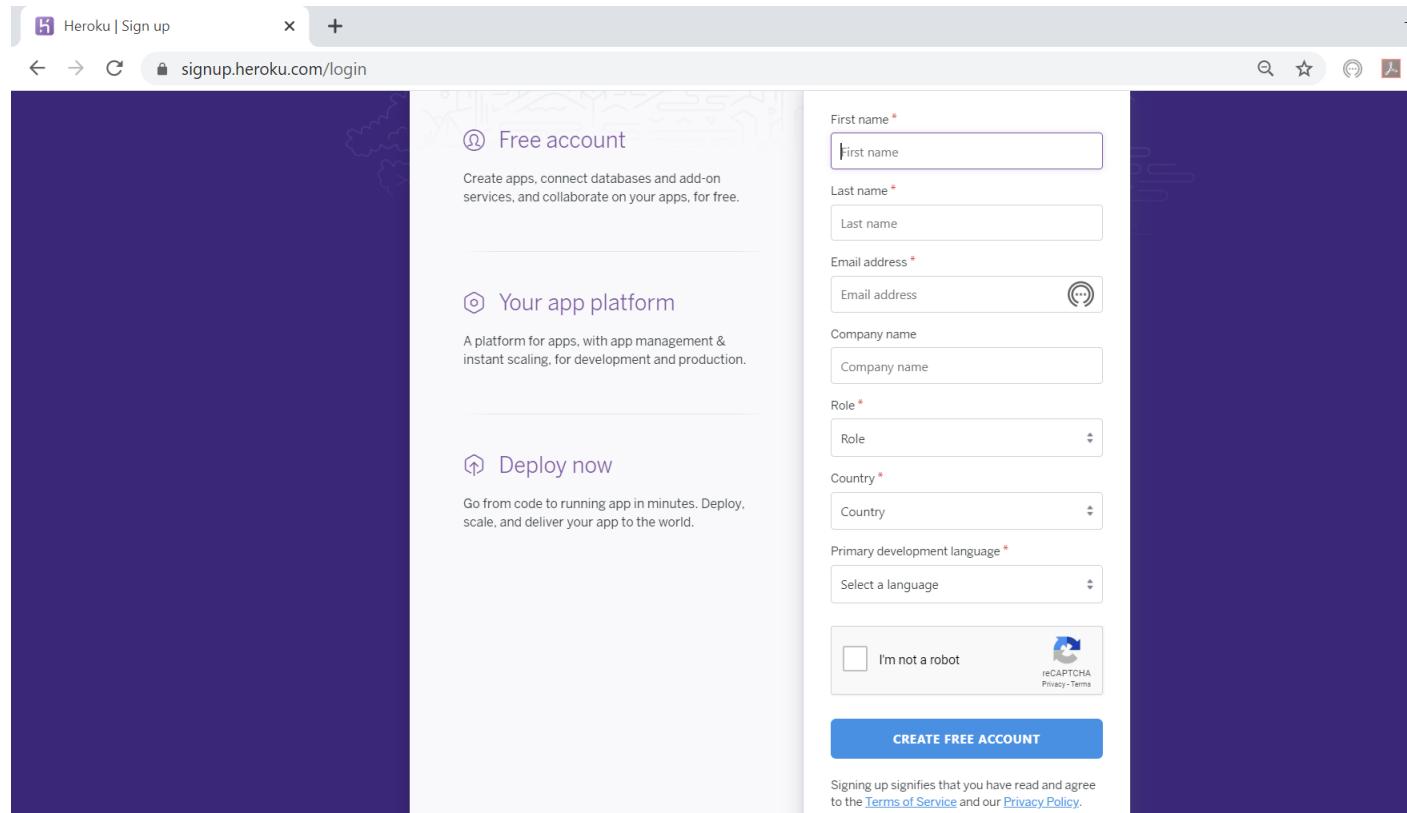


Slika 5.13: Dijagram razmještaja

5.4 Upute za puštanje u pogon

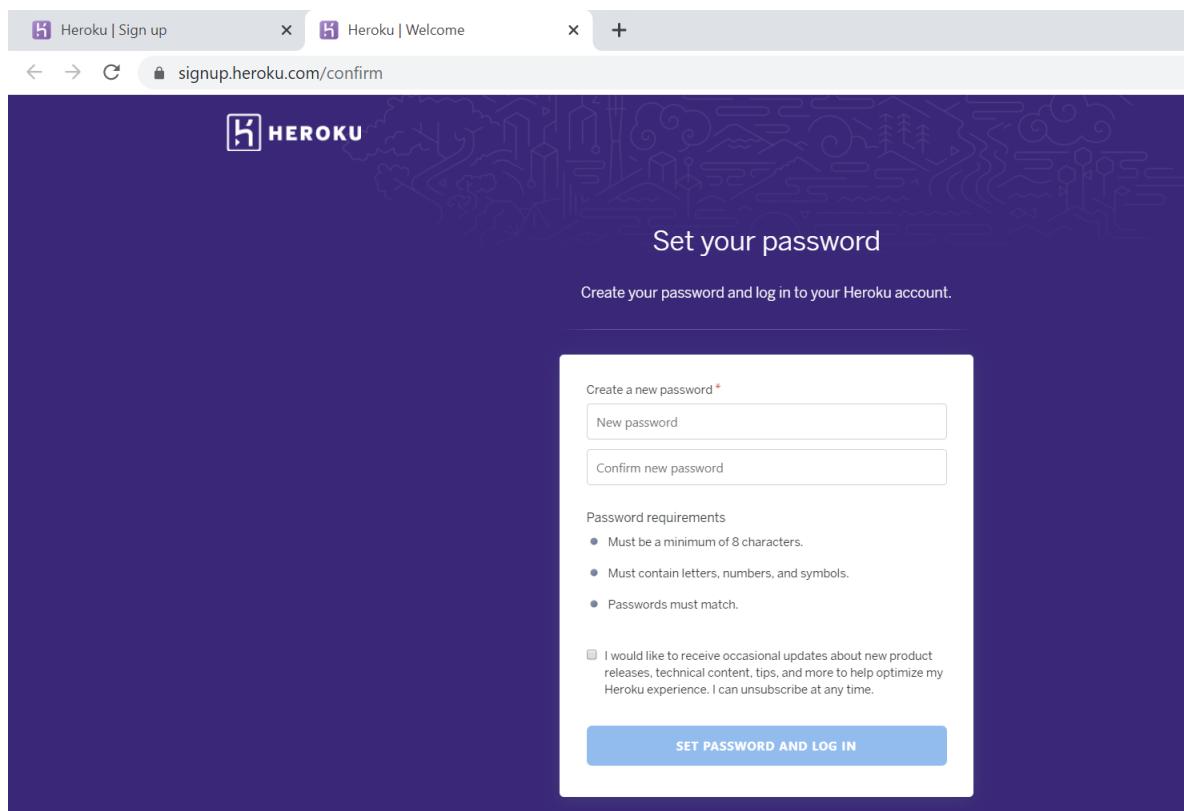
Kako bi pustili aplikaciju u pogon, potrebno je napraviti deploy. U našem slučaju taj deploy smo napravili na Heroku.

Kako bi napravili deploy aplikacije na Heroku potrebno je napraviti korisnički račun.



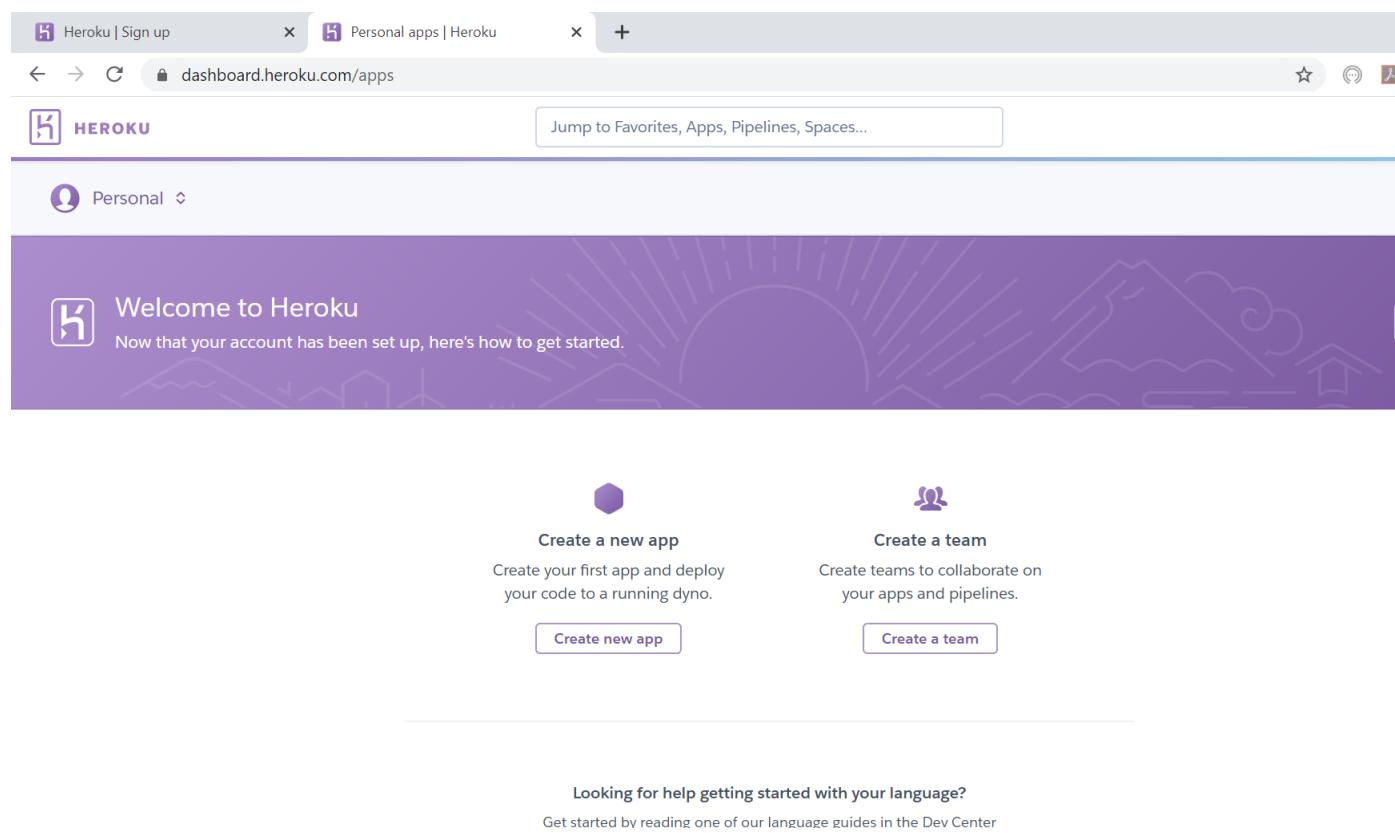
Slika 5.14: Izrada korisničkog računa

Nakon što se napravi korisnički račun koji je besplatan, na mail adresu koja je upisana dobije se link na koji se mora kliknuti kako bi se potvrdio račun i postavila lozinka.



Slika 5.15: Potvrda lozinke

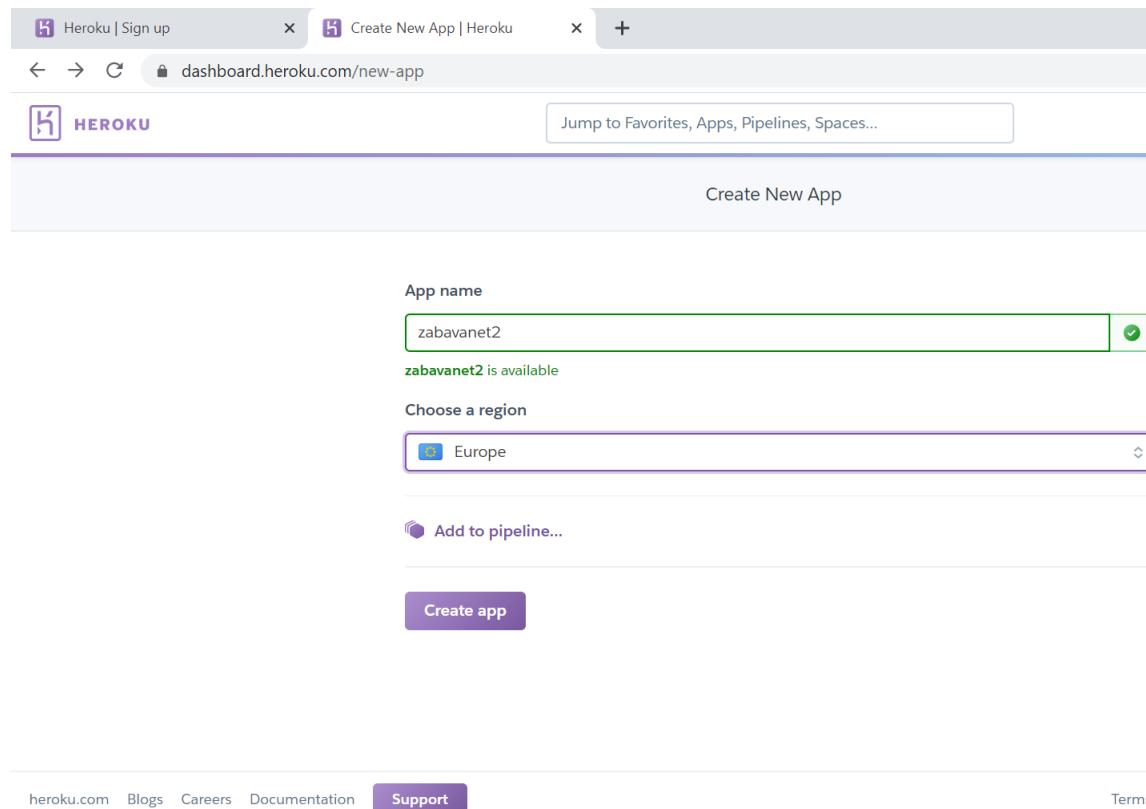
Nakon što se postavi lozinka, moguće je dodati aplikaciju.



Slika 5.16: Dodavanje aplikacije

Tu se odabere opcija „Create a new app“ te se nakon toga odabere ime aplikacije i regija.

Mi smo odabrali zabavanet2 i „Europe“



Slika 5.17: Odabir imena i regije

Dalje sve što treba napraviti je slijediti ove upute i aplikacija je postavljena.

The screenshot shows the Heroku dashboard for the app 'zabavanet2'. It features a navigation bar with tabs for 'zabavanet2 · Heroku-git | Heroku' and 'The Heroku CLI | Heroku Dev Cen'. Below the navigation is a search bar with the URL 'dashboard.heroku.com/apps/zabavanet2/deploy/heroku-git'. The main content area has a purple header 'HEROKU' and a search bar 'Jump to Favorites, Apps, Pipelines, Spaces...'. Below this are three links: 'Heroku Git Use Heroku CLI', 'GITHUB Connect to GitHub', and 'Container Registry Use Heroku CLI'. The central part of the page contains two columns of deployment instructions:

Deploy using Heroku Git	Install the Heroku CLI
Use git in the command line or a GUI tool to deploy this app.	Download and install the Heroku CLI . If you haven't already, log in to your Heroku account and follow the prompts to create a new SSH public key.
<pre>\$ heroku login</pre>	
Create a new Git repository	Initialize a git repository in a new or existing directory
<pre>\$ cd my-project/ \$ git init \$ heroku git:remote -a zabavanet2</pre>	
Deploy your application	Commit your code to the repository and deploy it to Heroku using Git.
<pre>\$ git add . \$ git commit -am "make it better" \$ git push heroku master</pre>	

Slika 5.18: Instalacija

1. Treba instalirati Heroku CLI, link je u uputam Herokua.
2. Napravimo vlastitu mapu na računalu „herokuDeploy2“ u koju se kopira sav potreban sadržaj aplikacije.
3. Nakon što je to učinjeno, u Command promptu na Windowsima možemo izvršiti ove naredbe.
4. Aplikacija bi ukoliko ne bude nikakvih problema trebala biti deployjana i dobit ćemo ovaku poruku : „Build success“.

To bi značilo da je sve uspješno napravljeno i možemo pristupiti aplikaciji na doivenom linku.

6. Zaključak i budući rad

Kada smo tek krenuli izrađivati projekt, najveći blagoslov, ali i teret na neki način predstavljalo je što za izradu imamo otvorene sve mogućnosti, ali trebalo je na neki način ukomponirati da izrada ne bude previše komplikirana, a da tehnologije koje koristimo nam omoguće laganu nadogradnju za kasnije. Uz to trebalo je i riješiti sve zadatke zadane. Odabrali smo Spring Boot s obzirom da doista uvelike olakšava izradu web aplikacija jer toliko stvari rješava u pozadini da je bilo malo i zastrašujuće, s obzirom da je nas nekolicina navikla bila pisati sve samostalno. Sljedeći izazov bio je frontend, točnije koju tehnologiju odabrat. Sad kad je aplikacija gotova mišljenja sam da ono što smo mi odabrali, a to je JSP, tehnologija je kod koje je užasno teško za ukomponirat više stvari na stranici. Bilo bi lakše da smo odabrali React iako ga nitko ne zna u grupi. Mnogo bi naučili, ali ostavit ćemo ga za sljedeći put.

Također trebalo je osmisiliti kako ćemo u bazi čuvati podatke te kako ćemo to dajte slati prema JSP stranicama koje te podatke koriste. Organiziranje baze nije bio lagan posao, možda smo nešto mogli drugačije, ali kako trenutno stvari stoje, baza se lagano može nadogradit ako zatrebaju nove funkcionalnosti, tako da smatram da je taj dio uspješno riješen.

Znanja koja su potrebna za bržu izradu projekta je definitivno poznavanje objektno orjenitrane paradigme te općenita znanja za baze podataka, te definitivno poznavanje HTML-a te Javascripta. Jako bi nam bilo otežano da smo sve stvari pisali kad god nam je u kodu bilo potrebno. No i taj dio smo dobro riješili s obzirom da smo napravili dobar reusability u kodu.

Također kod se može isto lagano nadograditi za nove funkcionalnosti, tipa novu vrstu korisnika, ali to bi vjerojatno zakompliciralo stvari na frontendu pa zbog toga React bi bio bolja opcija.

Uz to svi smo naučili koristiti se Git-om. U početku je to predstavljalo problem, ali kad smo savladali par puta što se može desiti, lako smo kasnije rješavali probleme. Još jednom, sve funkcionalnosti su implemenitrane.

Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
3. I. Sommerville, "Software engineering", 8th ed. Addison Wesley, 2007.
4. P. Clements, D. Garlan, P. Merson, "Documenting Software Architectures", 2nd ed. Addison-Wesley, 2010.

Indeks slika i dijagrama

2.1 Facebook događaji	6
3.1 Dijagram obrasca uporabe, funkcionalnosti administratora	22
3.2 Dijagram obrasca uporabe, funkcionalnosti prijavljenih i neprijavljenih korisnika	23
3.3 Dijagram obrasca uporabe, funkcionalnosti organizatora	24
3.4 Dijagram obrasca uporabe, funkcionalnosti posjetitelja	25
3.5 Sekvencijski dijagram za UC1	26
3.6 Sekvencijski dijagram za UC5	27
3.7 Sekvencijski dijagram za UC10	29
3.8 Sekvencijski dijagram za UC13	31
4.1 Klijent-Server model	33
4.2 Spring arhitektura	34
4.3 MVC arhitektura	35
4.4 JSP arhitektura	36
4.5 E-R dijagram baze podataka	43
4.6 Prvi dio slike poveznice Servicea i Repositoryja	45
4.7 Drugi dio poveznica Servicea i Repositoryja	46
4.8 Poveznica objekta i Repositoryja	47
4.9 Nastavak poveznica objekta i Repositoryja	48
4.10 Controlleri prvi dio	49
4.11 Controlleri drugi dio	50
4.12 Validatori i Forme prvi dio	51
4.13 Validatori i Forme drugi dio	52
4.14 Dijagram stanja - Organizator	54
4.15 Dijagram aktivnosti - Dodavanje recenzije	56
4.16 Dijagram komponenti	57
5.1 Ispitivanje ispravnog i neispravnog unosa događaja	60
5.2 Ispitivanje pogrešnog unosa datuma	61

5.3 Ispitivanje kategorije i mesta događaja	62
5.4 Rezultati testiranja	62
5.5 Dodavanje novog događaja	63
5.6 Provjeravanje prikaza svih errora kod ne unosa dodavanja događja .	64
5.7 Prikaz errora kada je vrijeme završetka prije noego vrijeme početka	64
5.8 Uređivanje korisničkog imena	65
5.9 Registracija kao organizator	65
5.10 Metoda pri registraciji	65
5.11 Registracija	65
5.12 Rezultati testiranja	66
5.13 Dijagram razmještaja	67
5.14 Izrada korisničkog računa	68
5.15 Potvrda lozinke	69
5.16 Dodavanje aplikacije	70
5.17 Odabir imena i regije	71
5.18 Instalacija	72
6.1 Dijagram aktivnosti na grani <i>master</i>	81
6.2 Dijagram aktivnosti na grani <i>devdoc</i>	81

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

1. sastanak

- Datum: 17. listopada 2019.
- Prisustvovali: A. Filipović, P. Tišma, D. Vusić, D. Marić, L. Duvnjak, D. Pavelić
- Teme sastanka:
 - sastanak s asistenticom
 - analiza zadatka
 - odabir alata

2. sastanak

- Datum: 7. studenog 2019.
- Prisustvovali: A. Filipović, P. Tišma, D. Vusić, D. Marić, L. Duvnjak, D. Pavelić
- Teme sastanka:
 - sastanak s mentoricom
 - komentiranje dosadašnjeg rada (uglavnom registracije, prijave i brišanje računa)
 - odabir UC dijagrama

3. sastanak

- Datum: 14. studenog 2019.
- Prisustvovali: A. Filipović, D. Vusić, D. Marić, L. Duvnjak, D. Pavelić
- Teme sastanka:
 - sastanak s demonstratoricom
 - komentiranje dokumentacije
 - komentiranje obaveznih poglavlja za 1. reviziju

4. sastanak

- Datum: 19. prosinca 2019.
- Prisustvovali: A. Filipović, P. Tišma, D. Vusić, D. Marić, L. Duvnjak,
- Teme sastanka:
 - sastanak s mentoricom
 - komentiranje novonapravljenog posla
 - odgovori na pitanja o dalnjem radu

5. sastanak

- Datum: 9. siječnja 2020.
- Prisustvovali: A. Filipović, P. Tišma, D. Vusić, D. Marić, L. Duvnjak
- Teme sastanka:
 - sastanak s demonstratoricom
 - demonstracija alfa verzije
 - komentiranje alfa verzije
 - odgovori na pitanja o izgledu finalnog proizvoda

6. sastanak

- Datum: 16. siječnja 2020.
- Prisustvovali: A. Filipović, P. Tišma, D. Vusić, D. Marić, L. Duvnjak, D. Pavelić
- Teme sastanka:
 - sastanak s mentoricom
 - komentiranje dokumentacije
 - komentiranje načina prikazivanja završnog proizvoda

Tablica aktivnosti

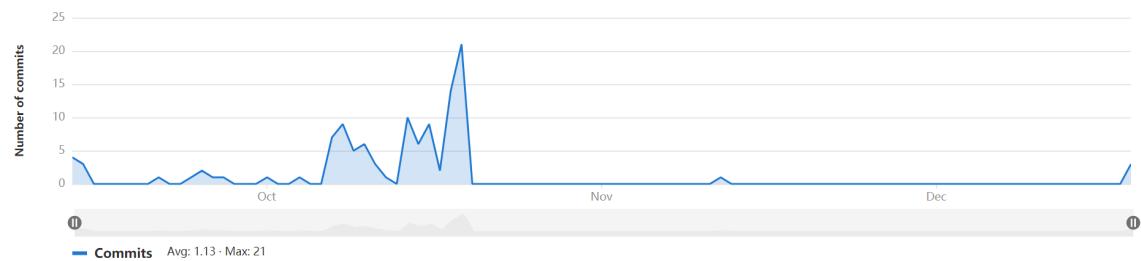
Kontinuirano osvježavanje

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

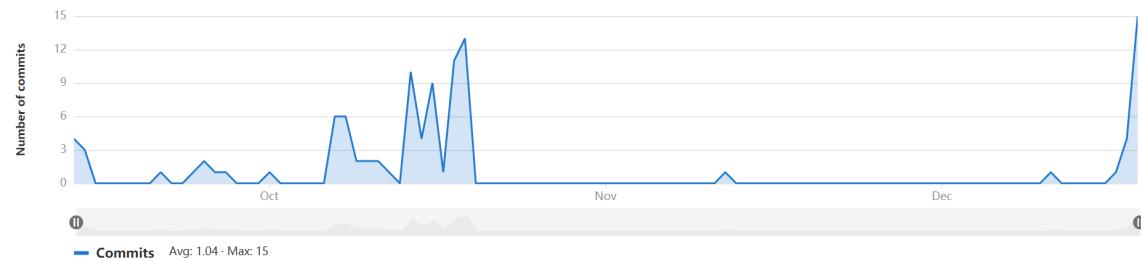
	Antonio Filipović	Petar Tišma	Daniel Vusić	Daniel Marić	Luka Duvnjak	Dora Pavelić
Upravljanje projektom	5					
Opis projektnog zadatka						3
Funkcionalni zahtjevi				1	3	
Opis pojedinih obrazaca		1				
Dijagram obrazaca	4		4	5		
Sekvencijski dijagrami		5				
Opis ostalih zahtjeva	1					1
Arhitektura i dizajn sustava	1		1			1
Baza podataka	1				3	
Dijagram razreda	4					2
Dijagram stanja		3				1
Dijagram aktivnosti						2
Dijagram komponenti				3		
Korištene tehnologije i alati			2			1
Ispitivanje programskog rješenja		3	3			1
Dijagram razmještaja						
Upute za puštanje u pogon	2					1
Dnevnik sastajanja						2
Zaključak i budući rad				2		
Popis literature						1
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>						

	Antonio Filipović	Petar Tišma	Daniel Vusić	Daniel Marić	Luka Duvnjak	Dora Pavelić
<i>npr. izrada početne stranice</i>	1		1	1		
<i>izrada baze podataka</i>	3		1			
<i>spajanje s bazom podataka</i>	3					
<i>back end</i>	5		5			
<i>front end</i>	2		2	3		

Dijagrami pregleda promjena



Slika 6.1: Dijagram promjena na grani *master*



Slika 6.2: Dijagram promjena na grani *devdoc*