

Programsko inženjerstvo

Ak. god. 2022./2023.

# Interaktivni socijalni kalendar Eventko

Dokumentacija, Rev. 2

Grupa: *Vidoje*

Voditelj: *Velimir Kovačić*

Datum predaje: *13. 01. 2023.*

Nastavnik: *Miljenko Krhen*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>4</b>
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	10
3.1.2 Sekvencijski dijagrami . . . . .	17
3.2 Ostali zahtjevi . . . . .	21
<b>4 Arhitektura i dizajn sustava</b>	<b>22</b>
4.1 Baza podataka . . . . .	24
4.1.1 Opis tablica . . . . .	24
4.1.2 Dijagrami baze podataka . . . . .	29
4.2 Dijagram razreda . . . . .	31
4.3 Dijagram stanja . . . . .	34
4.4 Dijagram aktivnosti . . . . .	35
4.5 Dijagram komponenti . . . . .	37
<b>5 Implementacija i korisničko sučelje</b>	<b>38</b>
5.1 Korištene tehnologije i alati . . . . .	38
5.2 Ispitivanje programskog rješenja . . . . .	39
5.2.1 Ispitivanje komponenti . . . . .	39
5.2.2 Ispitivanje sustava . . . . .	39
5.3 Dijagram razmještaja . . . . .	40
5.4 Upute za puštanje u pogon . . . . .	41
<b>6 Zaključak i budući rad</b>	<b>48</b>
<b>Popis literature</b>	<b>49</b>
<b>Indeks slika i dijagrama</b>	<b>50</b>

**Dodatak: Prikaz aktivnosti grupe**

**51**

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Perković, Saganić	03.11.2022.
0.2	Dodan detaljan opis projekta. Popunjen dosadašnji dnevnik sastajanja. Dodani obrasci uporabe i funkcionalni zahtjevi.	Perković, Saganić	14.11.2022.
0.5	Dodan <i>Use Case</i> dijagram i dva sekvencijska dijagrama, uređeni funkcionalni zahtjevi	Kovačić, Perković, Saganić	15.11.2022.
0.6	Arhitektura sustava. Dijagrami razreda. Uređivanje slika. Baza podataka. Ostali zahtjevi	Perković, Saganić	16.11.2022.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	Kovačić, Perković, Saganić	18.11.2022.
1.1	Novi dijagrami razreda, stanja i aktivnosti. Opis puštanja u pogon	Perković	10.01.2023.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije *Eventko*. Ova aplikacija omogućit će korisnicima da kvalitetnije i jednostavnije stvaraju i upisuju se na događaje. Aplikacija će pružati uvid u sve trenutne javne događaje kojima se može pristupiti. Događaji će biti prikazani u obliku kalendara koji je inspiriran izgledom kalendara u aplikaciji Ferko.

Prilikom prvog pokretanja aplikacije korisnik se nalazi na stranici za prijavu na kojoj će se prikazati prozorčić s poljima za potrebne podatke:

- polje za unos korisničkog imena profila
- polje za unos lozinke korisničkog profila
- gumb za prijavu
- gumb za izradu novog računa (*register*)

Tijekom prvog pokretanja profil još nije napravljen tako da je potrebno stisnuti gumb *registrirajte se* kako bi se napravio novi korisnički profil. Nakon pritiska gumba aplikacija prebacuje korisnika na novu stranicu na kojoj je prikazan prozor s poljima za unos potrebnih podataka za izradu novog profila, kao i gumbi za registraciju ili za povratak na stranicu za prijavu. Za registraciju novog profila potrebni su sljedeći podaci:

- korisničko ime
- nadimak (neobavezno)
- e-mail adresa
- lozinka
- ponovljena lozinka

Nakon što su podaci uspješno uneseni potrebno je stisnuti gumb *Registrirajte se* kako bi se novi korisnički profil generirao. Po uspješnom stvaranju korisnik je prebačen natrag na stranicu za prijavu te se može prijaviti svojim novim korisničkim računom. Nakon uspješne prijave aplikacija vodi korisnika na početnu stranicu.

Na početnoj stranici prikazana je alatna traka na kojoj su redom nanizani sljedeći objekti:

- logotip aplikacije Eventko
- Moji Prijatelji
- Pohađani Eventi
- Upravljanje korisnicima (samo za moderatore i administratore)
- Korisničko ime ulogiranog korisnika
- Odjava

Ispod alatne trake prikazan je gumb *Dodaj u kalendar* kojim je moguće kreiranje novog javnog ili privatnog događaja te osobne obaveze. Odabir otvara novi prozorčić u koji je potrebno unijeti sljedeće podatke:

- Naziv događaja
- Mjesto događaja
- Datum i vrijeme početka
- Datum i vrijeme kraja
- Vrsta događaja
- Oznake događaja (npr. kava, učenje, kviz...)
- Opis događaja

*Vrsta događaja* padajući je izbornik s opcijama Obveza, Privatni događaj i javan događaj. Javni događaji vidljivi su svim korisnicima na desnoj strani početne stranice, privatni samo autorovim prijateljima, a obaveze samo njihovom autoru.

Ispod gumba *Dodaj u kalendar* nalaze se liste aktivnih korisnika te istaknutih događaja koje uređuju moderatori. Većinu prikaza stranice zauzima središnji kalendar na kojem su vidljive sve korisnikove obaveze te događaji kojima je odabrao prisustvovati ili koje sam organizira. Prikazuje se tekući tjedan, a moguće je listati i one prethodne i nadolazeće. Klikom na događaj u kalendaru vidljive su sve informacije i oznake te opcije brisanja, objavljivanja ili uređivanja.

S desne strane nalazi se sekcija *Dostupni eventi* s padajućim izbornikom koji navodi nadolazeće javne događaje u kojima je moguće sudjelovati. Odabir pojedinog događaja privremeno ga prikazuje u kalendaru gdje se vide njegove informacije i opcija prijave čime se trajno dodaje u korisnikov kalendar.

Klik na logotip aplikacije uvijek vraća korisnika na početnu stranicu.

Odabir kartice *Moji Prijatelji* otvara stranicu koja prikazuje listu trenutnih prijatelja korisnika uz opciju da ih se ukloni s liste. Ispod toga nalazi se gumb *Pretraži korisnike* koji otvara tražilicu korisnika po korisničkom imenu ili nadimku.

Odabir kartice *Pohađani Eventi* prebacuje korisnika na novu stranicu sa svim događajima na kojima je korisnik sudjelovao i bit će mu omogućeno da ih označi sa *Sviđa mi se* ili *Ne sviđa mi se*, što pozitivno ili negativno utječe na korisničku ocjenu njihovog autora, prikazanu kao broj u uglatim zagradama.

Odabir kartice s vlastitim korisničkim imenom otvara stranicu koja prikazuje korisnikove podatke:

- Nadimak, uz opciju izmjene
- Korisničko ime
- E-mail adresa
- Korisnička ocjena
- ukoliko je riječ o običnom korisniku, opcija pretplate za Premium račun

Odabir kartice *Odjava* vraća korisnika na stranicu za prijavu.

Korisnik aplikacije može imati jednu ili više od sljedećih uloga:

- Običan korisnik
- Premium korisnik
- Pregledavač
- Moderator
- Administrator

Običnom korisniku dostupne su sve funkcionalnosti prethodno opisane. Ostale uloge proširuju mogućnosti korisnika na način da mu svaka otključava nove značajke. Pregledavač je korisnik na uvodnoj stranici za prijavu koji trenutno nije prijavljen.

Premium korisnik ima sve mogućnosti običnoga uz dodatnu opciju promoviranja događaja koje organizira, čime se oni prikazuju na listi istaknutih događaja na

lijevoj strani stranice.

Moderator ima sve mogućnosti običnoga korisnika, uz to što ima dodatnu karticu na alatnoj traci imena *Upravljanje korisnicima*, koja ga prebacuje na novu stranicu na kojoj može suspendirati korisnike koji se ne ponašaju u skladu s internet-skim bontonom. Također ima opciju uređivanja oznaka na događajima kako bi mogao dodati ili ukloniti oznake po potrebi. Konačno, ima opciju *Obriši* ako je neki javni događaj neprimjeren.

Administrator ima sve mogućnosti moderatora i običnog korisnika uz to što može na stranici za upravljanje korisnicima ili promovirati posebno vrijedne članove zajednice u moderatore, ili obrisati račune korisnika koji uporno krše pravila i nakon privremene suspenzije.



## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Administrator
2. Moderator
3. Korisnici aplikacije
  - (a) Pregledavač
  - (b) Običan korisnik
  - (c) Premium korisnik
4. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Pregledavač/neprijavljen korisnik (inicijator) može:
  - (a) Otvoriti stranicu za prijavu
  - (b) Izvršiti prijavu
  - (c) Otvoriti stranicu za registraciju
  - (d) Izvršiti registraciju
2. Običan korisnik (inicijator) može:
  - (a) Dodati obveze, javne i privatne događaje
  - (b) Ocjenjivati posjećene događaje
  - (c) Prijavljivati se na događaje
  - (d) Preko nadimka dodati prijatelje
  - (e) Ukloniti prijatelje
  - (f) Blokirati korisnike
  - (g) Promijeniti nadimak profila
  - (h) Pretplatiti se na premium račun
3. Premium korisnik (inicijator) može:

- (a) Promovirati vlastiti događaj

4. Moderator (inicijator) može:

- (a) Suspendirati korisnika
- (b) Uređivati oznake javnih događaja
- (c) Brisati događaje

5. Administrator (inicijator) može:

- (a) Promovirati korisnika u moderatora
- (b) Brisati korisničke račune

6. Baza podataka (sudionik):

- (a) Pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) Pohranjuje sve podatke o događajima i njihovim karakteristikama

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registracija

- **Glavni sudionik:** Pregledavač
- **Cilj:** registracija
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik nije prijavljen u aplikaciji
- **Opis osnovnog tijeka:**
  1. Otvaranjem aplikacije otvara se stranica za prijavu
  2. Pritisne se gumb *registrirajte se*
  3. Unesu se potrebni podaci i pritisne gumb *Registrirajte se*
- **Opis mogućih odstupanja:**
  - 3.a Korisnik ne unese pravilno potrebne podatke

##### UC2 - Prijava

- **Glavni sudionik:** Pregledavač
- **Cilj:** prijava
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik nije prijavljen u aplikaciji, ali je registriran
- **Opis osnovnog tijeka:**
  1. Otvaranjem aplikacije otvara se stranica za prijavu
  2. Upišu se potrebni podaci
  3. Pritisne se gumb *Prijava*
- **Opis mogućih odstupanja:**
  - 2.a Korisnik ne unese pravilno potrebne podatke

##### UC3 - Promocija vlastitih događaja

- **Glavni sudionik:** Premium korisnik
- **Cilj:** isticanje svojih događaja drugim korisnicima
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen i kupljen je premium profil
- **Opis osnovnog tijeka:**
  1. Na Početnoj stranici pritisne se gumb *Dodaj u kalendar*
  2. Stvori se novi događaj i klikne se gumb *Promoviraj događaj*

#### UC4 - Blokiranje korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** blokiranje korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen
- **Opis osnovnog tijeka:**
  1. Pritiskom na karticu *Moji prijatelji* na alatnoj traci otvara se stranica za dodavanje prijatelja
  2. Na sekciji *Pretraži korisnike* unosi se željeno korisničko ime ili nadimak
  3. Pritiskom na opciju *Blokiraj* željeni korisnik je blokiran
- **Opis mogućih odstupanja:**
  - 2.a Uneseno nepostojeće korisničko ime

#### UC5 - Pretplata za premium

- **Glavni sudionik:** Korisnik
- **Cilj:** Pretplata na premium profil
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen kao običan korisnik
- **Opis osnovnog tijeka:**
  1. Pritiskom na karticu *Korisnik* na alatnoj traci otvara se profil korisnika
  2. Pritiskom na gumb *Kupnja premium računa* otvara se stranica za kupnju
  3. Unose se potrebni podaci i odabire gumb *Kupi*

#### UC6 - Izmjena nadimka

- **Glavni sudionik:** Korisnik
- **Cilj:** izmijeniti korisnički nadimak
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen
- **Opis osnovnog tijeka:**
  1. Klikom na vlastito korisničko ime na alatnoj traci korisnik je prebačen na stranicu svog profila
  2. Odabire polje *Nadimak* i upisuje novi željeni nadimak koji ne mora biti jedinstven

#### UC7 - Dodavanje prijatelja

- **Glavni sudionik:** Korisnik

- **Cilj:** dodavanje prijatelja
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen
- **Opis osnovnog tijeka:**
  1. Pritiskom na karticu *Moji Prijatelji* na alatnoj traci otvara se stranica za dodavanje prijatelja
  2. Pritiskom na *Pretraži korisnike* pokazuje se tražilica korisnika
  3. Prijatelja se može dodati upisivanjem korisničkog imena ili nadimka te klikom na *Dodaj*
- **Opis mogućih odstupanja:**
  - 2.a Unesen nepostojeći korisnik

#### UC8 - Uklanjanje prijatelja

- **Glavni sudionik:** Korisnik
- **Cilj:** uklanjanje prijatelja
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen
- **Opis osnovnog tijeka:**
  1. Pritiskom na karticu *Moji prijatelji* na alatnoj traci otvara se stranica za dodavanje prijatelja
  2. Prijatelja se može ukloniti s liste izborom *Ukloni prijatelja*

#### UC9 - Stvaranje događaja

- **Glavni sudionik:** Korisnik
- **Cilj:** stvaranje događaja u kalendaru
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen
- **Opis osnovnog tijeka:**
  1. Sa lijeve strane Početne stranice prikazan je gumb *Dodaj u kalendar*
  2. Otvara se prozor u koji se unose potrebni podaci i oznake
  3. Događaj se kreira pritiskom na *Dodaj događaj*
- **Opis mogućih odstupanja:**
  - 2.a Pogrešno uneseni podaci

#### UC10 - Prijava na događaj

- **Glavni sudionik:** Korisnik

- **Cilj:** prijavljivanje na događaj
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen
- **Opis osnovnog tijeka:**
  1. Na Početnoj stranici odabire se izbornik *Dostupni eventi*
  2. Otvara se padajući izbornik sa svim javnim događajima
  3. Pritiskom na događaj on se privremeno prikazuje u kalendaru
  4. Klikom na događaj u kalendaru otvara se prozor s njegovim opisom
  5. Klikom na gumb *Prijavi se* moguća je prijava na događaj

#### UC11 - Ocjenjivanje pohađanih javnih događaja

- **Glavni sudionik:** Korisnik
- **Cilj:** ocjenjivanje pohađanog događaja
- **Sudionici:** Baza podataka
- **Preduvjet:** : korisnik prijavljen i događaj dodan u kalendar
- **Opis osnovnog tijeka:**
  1. Na Početnoj stranici stisne se kartica *Pohađani događaji*
  2. Otvara se stranica sa svim pohađanim događajima i mogućnosti da se uz svaki odabere tipka *Sviđa mi se* ili *Ne sviđa mi se*

#### UC12 - Brisanje događaja

- **Glavni sudionik:** Moderator
- **Cilj:** brisanje događaja koji nisu u skladu sa pravilima aplikacije
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen i ima ovlasti moderatora
- **Opis osnovnog tijeka:**
  1. Na početnoj stranici moderator željeni događaj na kalendaru
  2. Otvara se prozorčić s informacijama o događaju uz opciju *Obriši*

#### UC13 - Uređivanje oznaka javnih događaja

- **Glavni sudionik:** Moderator
- **Cilj:** mijenjanje oznaka javnih događaja koje nisu adekvatno postavljene
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen i ima ovlasti moderatora
- **Opis osnovnog tijeka:**
  1. Na početnoj stranici moderator bira *Izbornik* i klikće željeni događaj

2. U kalendaru odabere događaj
3. Pritisne gumb *Izmijeni oznake* te ih proizvoljno dodaje i miče

#### UC14 - Suspendiranje korisnika

- **Glavni sudionik:** Moderator i korisnik
- **Cilj:** suspendiranje korisnika koji se nedolično ponašaju
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen i ima ovlasti moderatora
- **Opis osnovnog tijeka:**
  1. Na početnoj stranici moderator bira karticu *Upravljanje korisnicima* na alatnoj traci
  2. Otvara se stranica s pretragom korisnika po korisničkom imenu ili nadimku
  3. Moderator bira opciju *Suspendiraj* desno od korisničkog imena što ga blokira od korištenja profila na tjedan dana
- **Opis mogućih odstupanja:**
  - 2.a Pogrešan unos korisničkog imena

#### UC15 - Promocija korisnika u moderatora

- **Glavni sudionik:** Administrator i običan korisnik/premium korisnik
- **Cilj:** promidžba korisnika u moderatora
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen i ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Na početnoj stranici moderator bira karticu *Upravljanje korisnicima* na alatnoj traci
  2. Otvara se stranica s pretragom korisnika po korisničkom imenu
  3. Administrator bira opciju *Promoviraj* desno od korisničkog imena što mu dodaje ulogu i sposobnosti moderatora
- **Opis mogućih odstupanja:**
  - 2.a Pogrešan unos korisničkog imena
  - 3.a Korisnik je već moderator

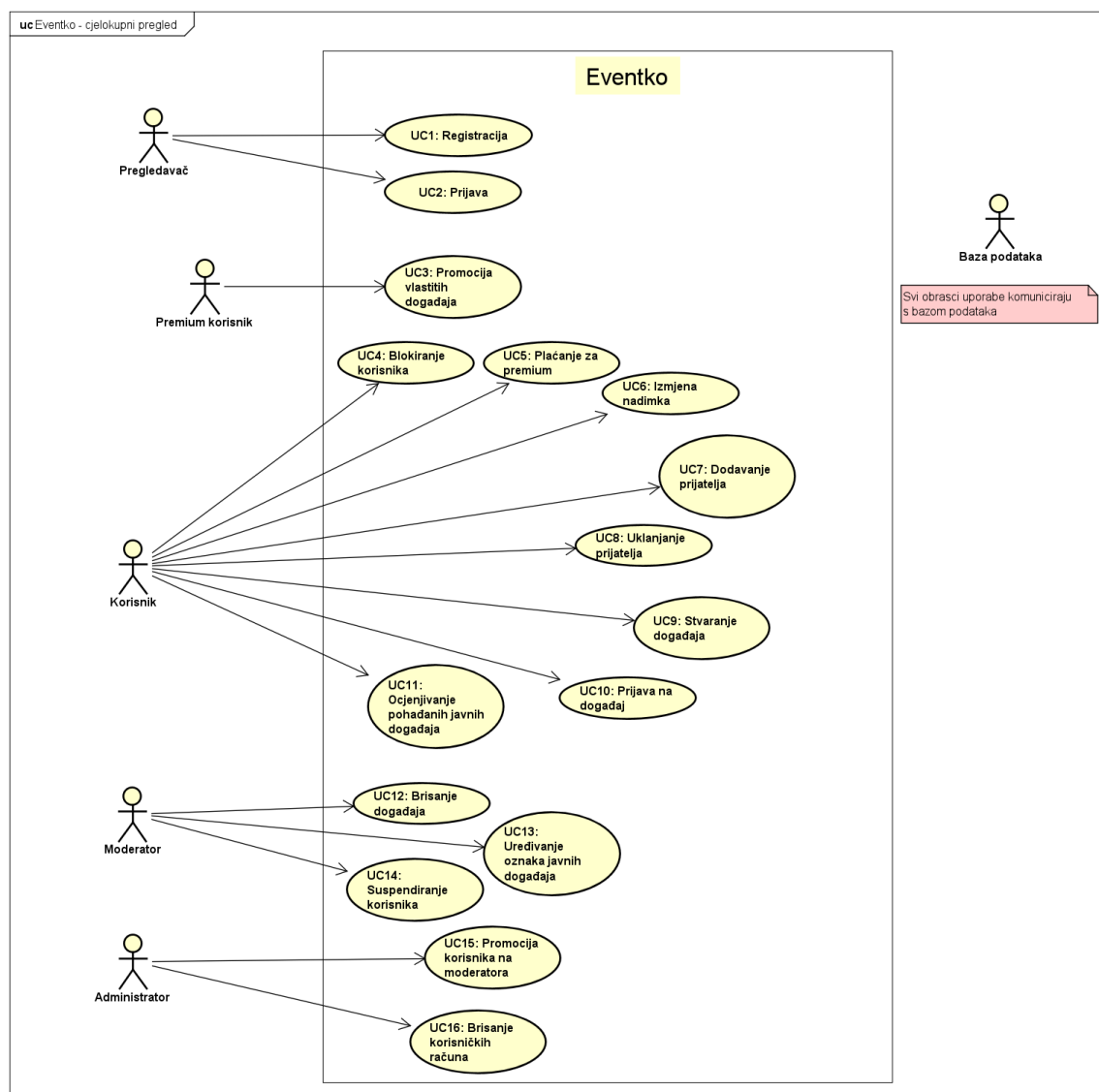
#### UC16 - Brisanje korisničkih računa

- **Glavni sudionik:** Administrator

- **Cilj:** Brisanje korisničkih računa osoba koje se neadekvatno ponašaju na aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** korisnik prijavljen i ima ovlasti administratora
- **Opis osnovnog tijeka:**
  1. Na početnoj stranici moderator bira karticu *Upravljanje korisnicima* na alatnoj traci
  2. Otvara se stranica s pretragom korisnika po korisničkom imenu
  3. Administrator bira opciju *Obriši* desno od korisničkog imena što briše račun iz baze podataka
- **Opis mogućih odstupanja:**
  - 2.a Pogrešan unos korisničkog imena



## Dijagram obrazaca uporabe



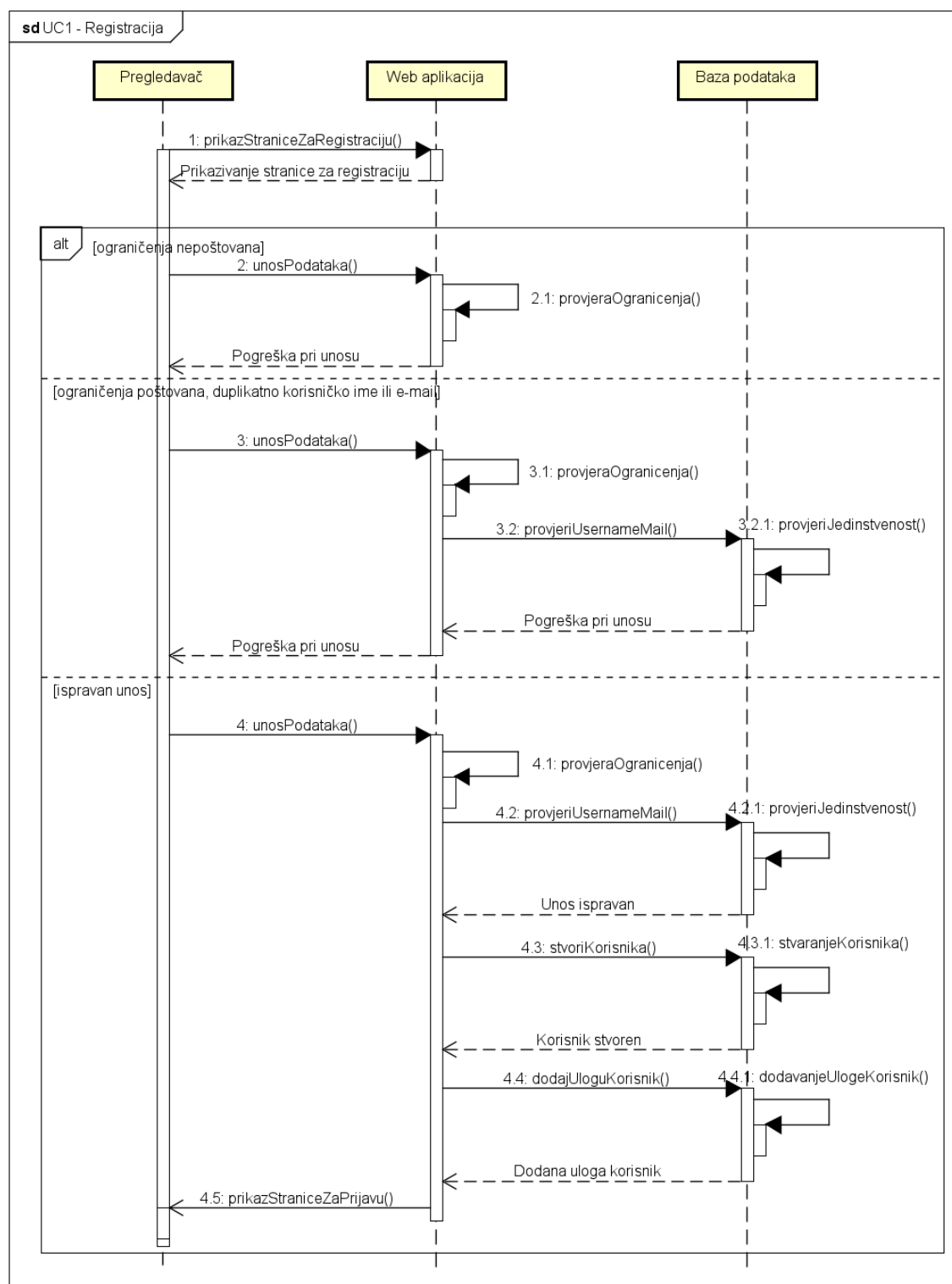
Slika 3.1: Dijagram obrazaca uporabe, funkcionalnosti različitih korisnika stranice

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1 - Registracija

Pregledavač klikom na izbor *Registriraj se* poziva prikaz stranice za registraciju. Upisuje podatke potrebne za izradu računa koje web aplikacija šalje bazi podataka te ona pomoću njih izrađuje novi korisnički profil. Baza vraća potvrdu nakon izrađenog profila i korisnik je vraćen na prikaz stranice za prijavu.

Ukoliko je došlo do greške pri upisu podataka, korisnik ostaje na stranici prijave i pojavljuje mu se poruka o pogrešci uz podatak koji je kriv. Moguće greške pri unosa podataka mogu se desiti ako za korisničko ime koristimo znakove koji nisu u engleskoj abecedi, znamenke 0-9, "\_" ili ako je kraće od 2 znaka. Greška se može desiti za nadimak ako sadrži manje od 2 ili više od 25 znakova. E-mail adresa mora biti u formatu *ime@domena.hr*. Lozinka mora sadržavati barem 4 znaka. Ta ograničenja provjeravaju se na web aplikaciji te samo ako su poštovana šalje se poruka bazi da pomoću njih stvori novi račun. Prije toga baza provjerava postoji li već korisnik s istim korisničkim imenom ili e-mail adresom.



Slika 3.2: Sekvencijski dijagram za UC1

**Obrazac uporabe UC2 - Prijava**

Korisnik šalje zahtjev za prikaz stranice za prijavu nakon kojeg upisuje potrebne podatke. Web aplikacija šalje podatke bazi podataka koja dohvaća korisnike i traži zadanog korisnika prema imenu profila. Baza podataka vraća profil korisnika nakon kojeg je korisniku prikazana početna stranica. Ukoliko dođe do greške korisnik ostaje na stranici Prijava te mu je prikazan error poruka. Greška se može desiti ako koristimo korisničko ime i lozinku koji nisu povezani jedno uz drugo ili ne postoje.

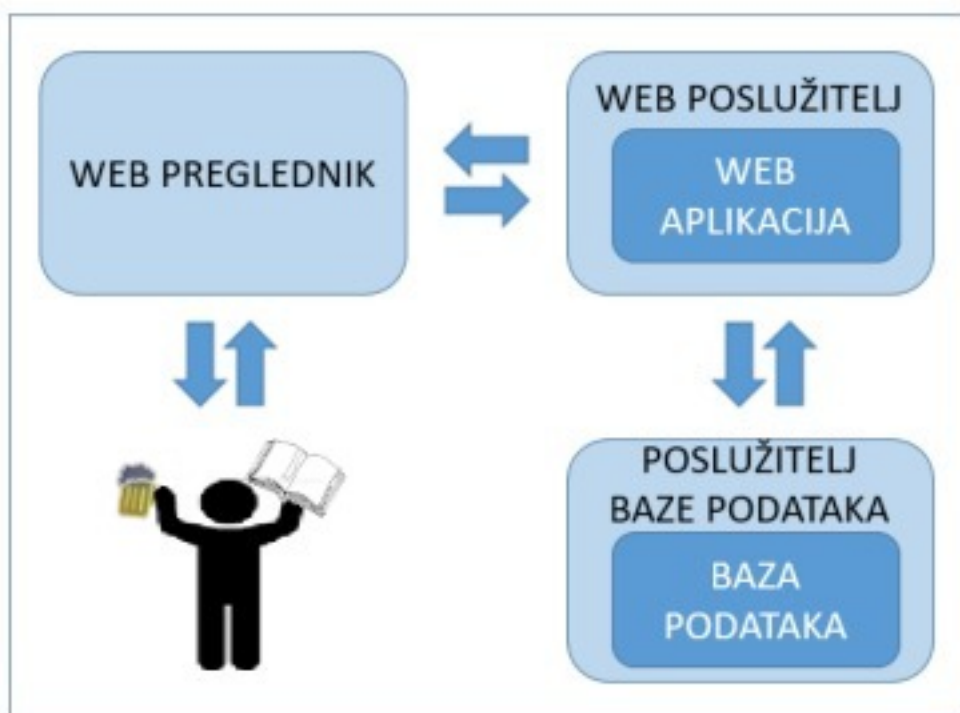


Slika 3.3: Sekvencijski dijagram za UC2

## 3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške

## 4. Arhitektura i dizajn sustava



Slika 4.1: Arhitektura sustava

Arhitekturu tvore tri podsustava: web poslužitelj, web aplikacija te baza podataka. *Web preglednik* je program za pregledavanje i navigaciju web-stranicama. Kada korisnik pošalje zahtjev za web-stranicom, preglednik dohvaća potrebne datoteke s *web poslužitelja* i prikazuje stranicu na korisnikovom ekranu u namijenjenom obliku. Poslužitelj omogućuje komunikaciju klijenta s *web aplikacijom* koja je na njemu pokrenuta, a prosljeđuje joj zahtjeve HTTP-om (engl. *Hyper Text Transfer Protocol*). Web aplikacija odgovara na zahtjeve klijenta pristupajući po potrebi bazi podataka i vraćajući HTML dokument čitljiv u web pregledniku.

Za izradu ovog projekta koristili smo se Spring Boot frameworkom u Javi kroz razvojno okruženje IntelliJ Community Edition, Javascriptom uz React u Visual Studio Code-u te nizom drugih programa za dizajn slika i grafova (GIMP, AstahUML itd.).

Arhitektura sustava prati MVC obrazac, odnosno Model-Pogled-Nadglednik (engl. *Model View Controller*), stilističku varijaciju arhitekture zasnovane na događajima. Takve arhitekture odlikuje to što se komponente međusobno ne pozivaju eksplicitno, već neke od njih generiraju signale (događaje) ne znajući koja druga "osluškuje" tj. očekuje takav signal i na njega reagira. Kod MVC-a pogodno je što smanjuje međuovisnost korisničkog sučelja i ostatka sustava, a omogućuje i nezavisan razvoj, nadogradnje i dodavanje različitih dijelova aplikacije. Sadrži različite gotove predloške za klase koji nam olakšavaju proces izrade.

MVC model sastoji se od komponenti:

- **Model** - Središnja komponenta sustava, sadrži razrede čiji se objekti obrađuju. Rukuje s podatkovnom logikom i bazom podataka. Prima podatke od nadglednika.
- **Pogled** - Predstavlja model korisniku na čitljiv način. Sadrži razrede čiji objekti služe za prikaz podataka. Dinamički se osvježava.
- **Nadglednik** - Razumije naputke korisnika i pretvara ih u upute ka modelu. Sadrži razrede koji upravljaju i rukuju korisničkom interakcijom s pogledom i modelom, poput poslovne logike i odgovora na događaje.



## 4.1 Baza podataka

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

- Korisnik
- Uloga
- Vrsta
- Oznaka
- ImaUlogu
- JePrijatelj
- JeBlokiranOd
- Dogadjaj
- Pohadja
- ImaOznaku

### 4.1.1 Opis tablica

**Korisnik** Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: id korisnika, nadimak, korisničko ime, email, salt, lozinka i suspendiran. Ovaj entitet u vezi je Many-to-Many s Uloga preko veze imaUlogu, u vezi Many-to-Many s Korisnik preko veze jePrijatelj, u vezi Many-to-Many s Korisnik preko veze JeBlokiranOd, te u vezi One-to-Many s entitetom Događaj preko i u vezi Many-to-Many s entitetom Dogadjaj preko veze Pohadja.

Korisnik	tip podataka	opis varijable
id korisnik	BIGINT NOT NULL	jedinstveni brojčani identifikator korisnika
nadimak	VARCHAR(25) NOT NULL	nadimak korisnika
korisnicko ime	VARCHAR(25) NOT NULL	ime korisnika
email	VARCHAR(255) NOT NULL	email korisnika
salt	BYTEA NOT NULL	salt za hashiranje lozinke
lozinka	BYTEA NOT NULL	hash lozinke
suspendiran	BOOLEAN NOT NULL	oznaka je li korisnik suspendiran

**Uloga** Ovaj entitet sadržava sve važne informacije o ulogama korisnika. Sadrži attribute: id uloga, naziv uloga i opis uloga. Ovaj entitet u vezi je Many-to-Many s Korisnik preko veze imaUlogu.

Uloga	tip podataka	opis varijable
id uloga	BIGINT NOT NULL	jedinstveni brojčani identifikator uloge korisnika
naziv uloga	VARCHAR(255) NOT NULL	naziv uloge korisnika
opis uloga	VARCHAR(255) NOT NULL	opis uloge korisnika

**Vrsta** Ovaj entitet sadržava sve važne informacije o vrstama događaja. Sadrži attribute: id vrsta, naziv vrsta i opis vrsta. Ovaj entitet u vezi je Many-to-On s entitetom Događaj preko identifikatora vrste u entitetu Događaj.

Vrsta	tip podataka	opis varijable
id vrsta	INT NOT NULL	jedinstveni brojčani identifikator vrste događaja
naziv vrsta	VARCHAR(255) NOT NULL	naziv vrste događaja
opis vrsta	VARCHAR(255) NOT NULL	opis vrste događaja

**Oznaka** Ovaj entitet sadržava sve važne informacije o oznakama događaja. Sadrži attribute: id oznaka, naziv oznaka i boja hex. Ovaj entitet u vezi je Many-to-Many s Oznaka preko veze imaOznaku.

Oznaka	tip podataka	opis varijable
id oznaka	INT NOT NULL	jedinstveni brojčani identifikator oznake događaja
naziv oznaka	VARCHAR(255) NOT NULL	naziv oznake događaja
boja hex	CHAR(7) NOT NULL	boja oznake

**ImaUlogu** Ova veza sadržava sve važne informacije po kojima saznajemo koji korisnik ima koju ulogu. Sadrži attribute: id korisnik i id uloga. Povezuje entitete Korisnik i Uloga.

ImaUlogu	tip podataka	opis varijable
id korisnik	BIGINT NOT NULL	jedinstveni brojčani identifikator korisnika (korisnik.id korisnik)
id uloga	BIGINT NOT NULL	jedinstveni brojčani identifikator uloga korisnika (uloga.id uloga)

**JePrijatelj** Ova veza sadržava sve važne informacije o tome koji je korisnik nekom drugom korisniku prijatelj. Sadrži attribute: id korisnik i id prijatelj. Povezuje entitet Korisnik sa samim sobom.

JePrijatelj	tip podataka	opis varijable
id korisnik	BIGINT NOT NULL	jedinstveni brojčani identifikator korisnika (korisnik.id korisnik)
id prijatelj	BIGINT NOT NULL	jedinstveni brojčani identifikator drugog korisnika prijatelja (korisnik.id korisnik)

**JeBlokiranOd** Ova veza sadržava sve važne informacije o tome koji je korisnik blokiran i od kojeg je korisnika blokiran. Sadrži attribute: id blokiran i id blokiran od. Povezuje entitet Korisnik sa samim sobom.

JeBlokiranOd	tip podataka	opis varijable
id blokiran	BIGINT NOT NULL	jedinstveni brojčani identifikator korisnika koji je blokiran (korisnik.id korisnik)
id blokiran od	BIGINT NOT NULL	jedinstveni brojčani identifikator korisnika koji blokira (korisnik.id korisnik)

**Događaj** Ovaj entitet sadržava sve važne informacije o događaju. Sadrži attribute: id događaj, naziv, mjesto, vrijeme početka, vrijeme kraja, opis, promoviran, koordinate, id organizator i id vrsta. Ovaj entitet u vezi je Many-to-One s entitetom Vrsta preko identifikatora vrste, u vezi Many-to-One s entitetom Korisnik preko identifikatora korisnika, u vezi Many-to-Many s entitetom Korisnik preko veze Pohadja i u vezi Many-to-Many s Oznaka preko veze imaOznaku.

Događaj	tip podataka	opis varijable
id događaj	BIGINT NOT NULL	jedinstveni brojčani identifikator događaja
naziv	VARCHAR(255) NOT NULL	naziv događaja
mjesto	VARCHAR(255) NOT NULL	mjesto zbivanja događaja
vrijeme poc	TIMESTAMP NOT NULL	vrijeme počinjanja događaja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Događaj	tip podataka	opis varijable
vrijeme kraj	TIMESTAMP NOT NULL	vrijeme završetka događaja
opis	VARCHAR(255) NOT NULL	opis događaja
promoviran	BOOLEAN NOT NULL	oznaka je li događaj promoviran
koordinate	VARCHAR(255) NOT NULL	koordinate događaja
id organizator	BIGINT NOT NULL	jedinstveni brojčani identifikator organizatora događaja (korisnik.id korisnik)
id vrsta	INT NOT NULL	jedinstveni brojčani identifikator vrste događaja (vrsta.id vrsta)

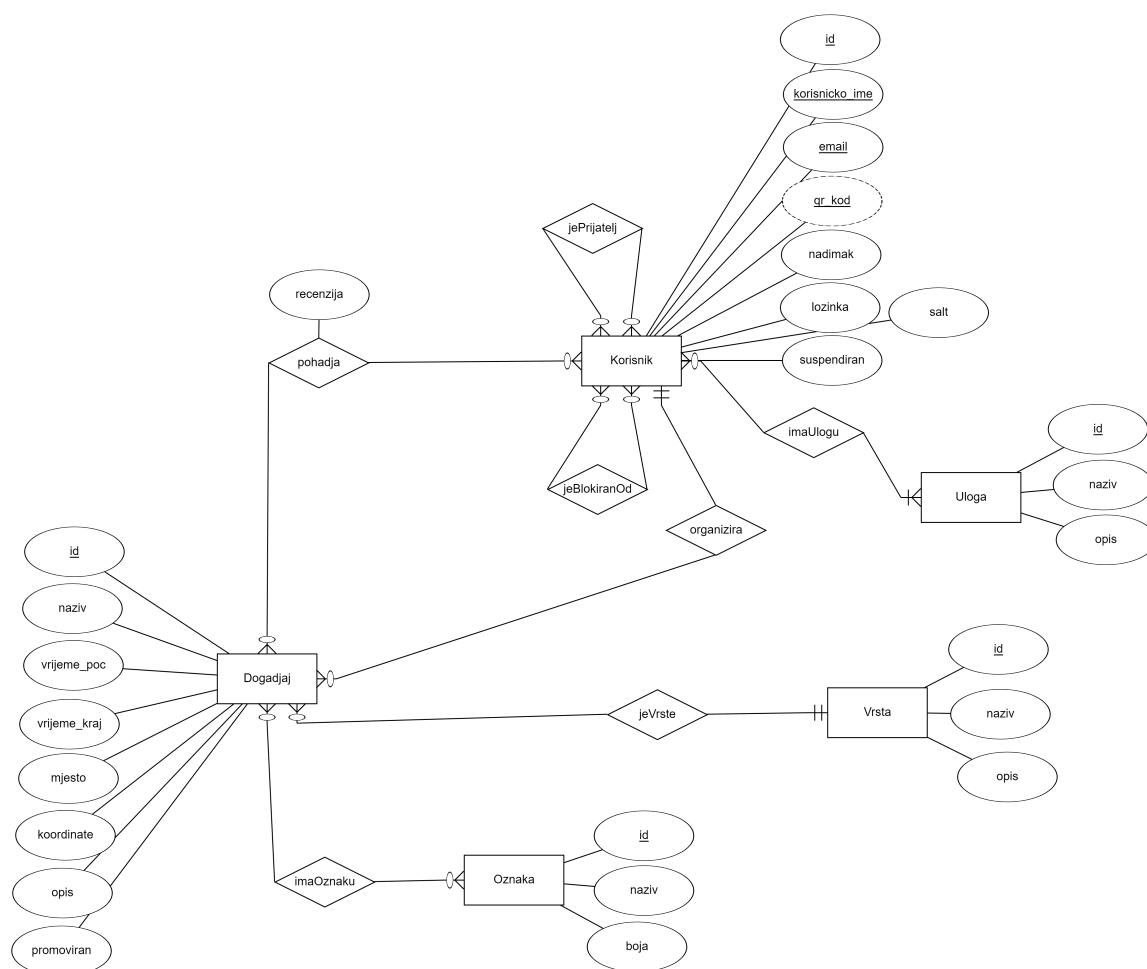
**Pohadja** Ova veza sadržava sve važne informacije o tome tko je pohađao koji događaj i kako ga je ocijenio. Sadrži attribute: recenzija, id polaznika i id događaja. Povezuje entitete Korisnik i Događaj.

Pohadja	tip podataka	opis varijable
recenzija	SMALLINT NOT NULL	recenzija korisnika za događaj
id pohađatelja	BIGINT NOT NULL	jedinstveni brojčani identifikator pohađatelja (korisnik.id korisnik)
id događaja	BIGINT NOT NULL	jedinstveni brojčani identifikator događaja (događaj.id događaj)

**ImaOznaku** Ova veza sadržava sve važne informacije o oznakama određenih događaja. Sadrži attribute: id događaj i id oznaka. Povezuje entitete Oznaka i Događaj.

ImaOznaku	tip podataka	opis varijable
id događaj	BIGINT NOT NULL	jedinstveni brojčani identifikator događaja (događaj.id događaj)
id oznaka	INT NOT NULL	jedinstveni brojčani identifikator oznake (oznaka.id oznaka)

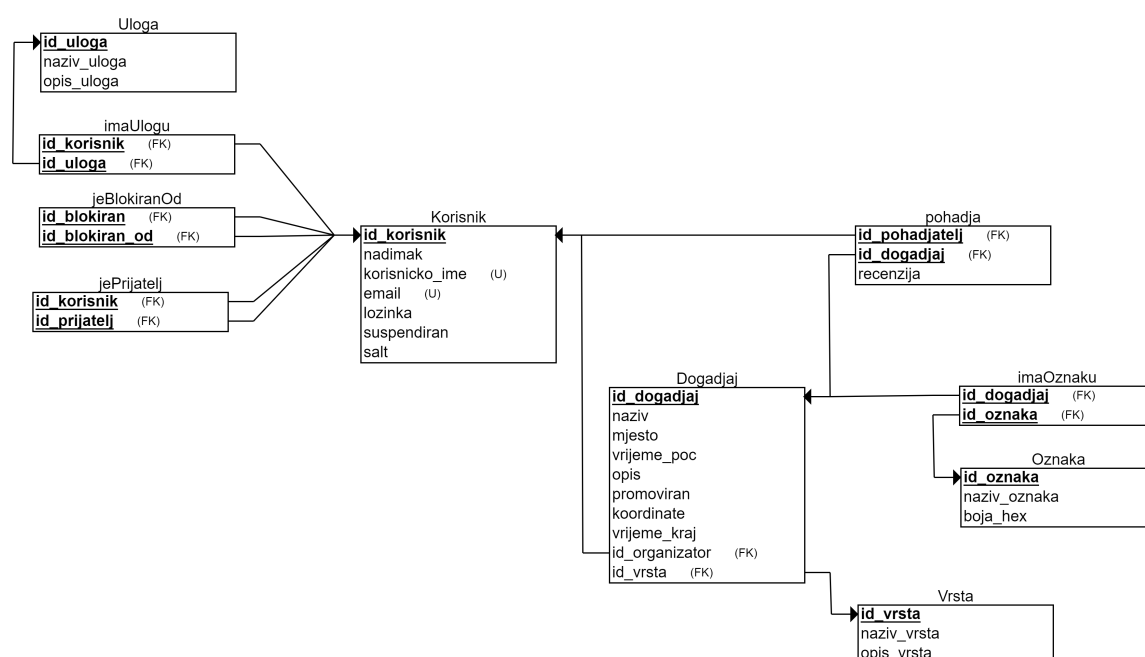
#### 4.1.2 Dijagrami baze podataka



Slika 4.2: ER dijagram baze podataka



Slika 4.3: ER dijagram baze podataka bez atributa



Slika 4.4: Relacijski dijagram baze podataka

## 4.2 Dijagram razreda

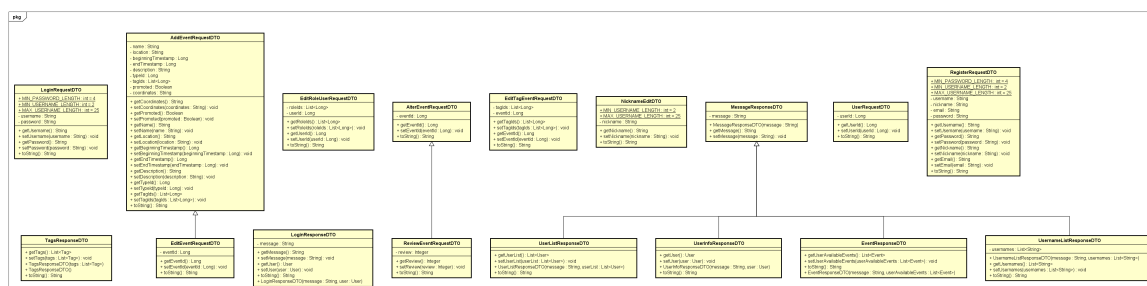
Na naredne tri slike nalaze se dijagrami razreda podijeljeni logički po srodnosti. Neki razredi povezani su i s onima na odvojenim slikama što se da zaključiti po nazivima njihovih metoda.

Razredi na slici 4.6 u gornjem redu su kontroleri. Njihove metode služe za primanje i slanje DTO-ova (*Data Transfer Objects*) prema frontendu u obliku JSON datoteka s html statusnim kodom. Pozivaju funkcije servisa. Razlikuju se kontroleri za korisnike, događaje te postupke prijave i registracije. Sami DTO razredi nalaze se na slici 4.5, a to su zahtjevi i odgovori za prijavu i registraciju.

Razredi na slici 4.6 u sredini su servisi. Služe za komunikaciju između repozitorija i kontrolera.

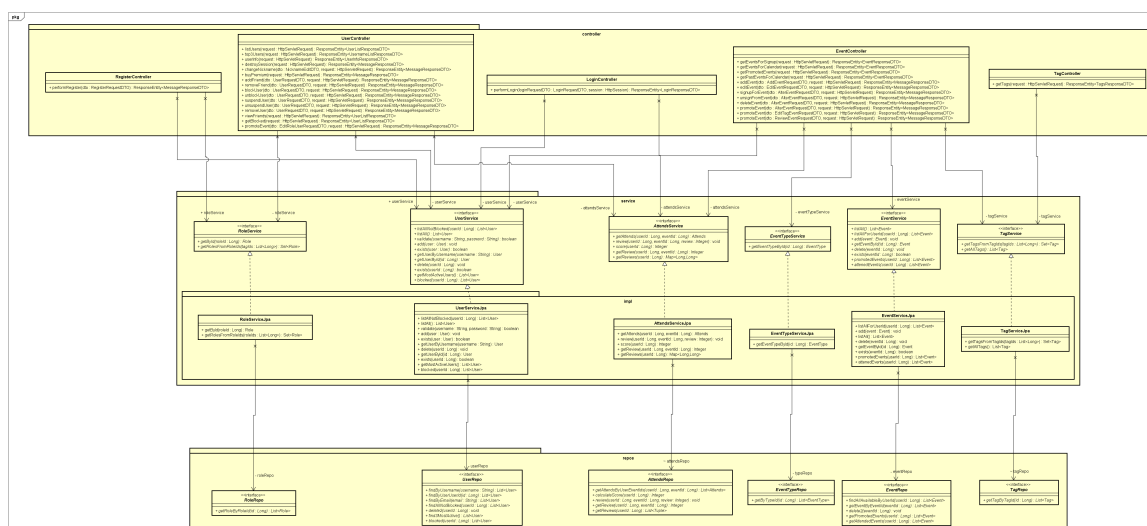
Razredi na slici 4.6 u donjem redu su repozitoriji. Služe za pozivanje SQL upita nad bazom podataka.

Razredi na slici 4.7 su modeli. Modeliraju potrebne razrede iz baze podataka. Razred *User* predstavlja korisnika aplikacije, *Role* predstavlja različite uloge, *Event* događaje, *EventType* vrste događaja i *Tag* oznake za događaje.

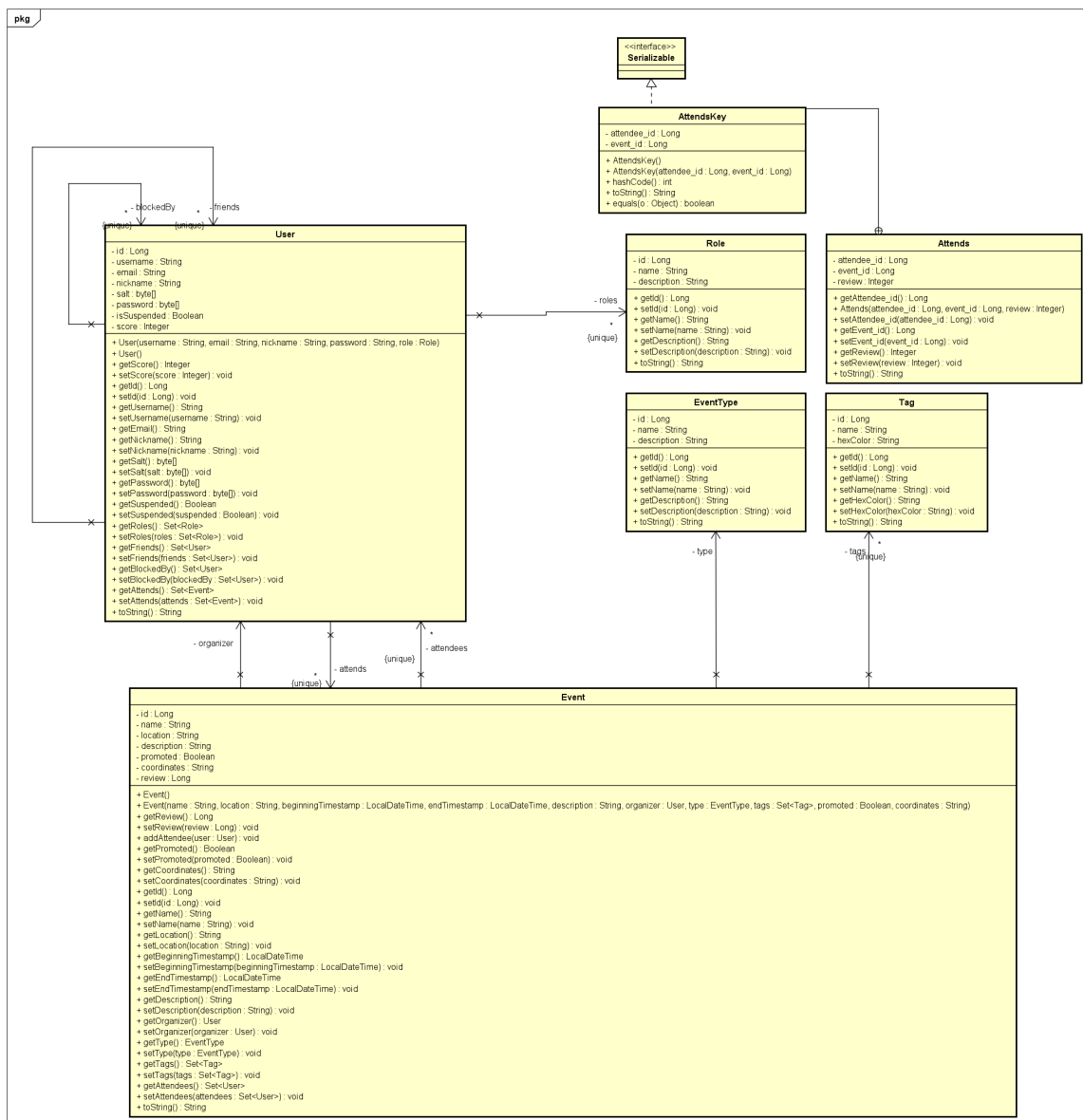


Slika 4.5: Dijagram razreda - DTO-ovi



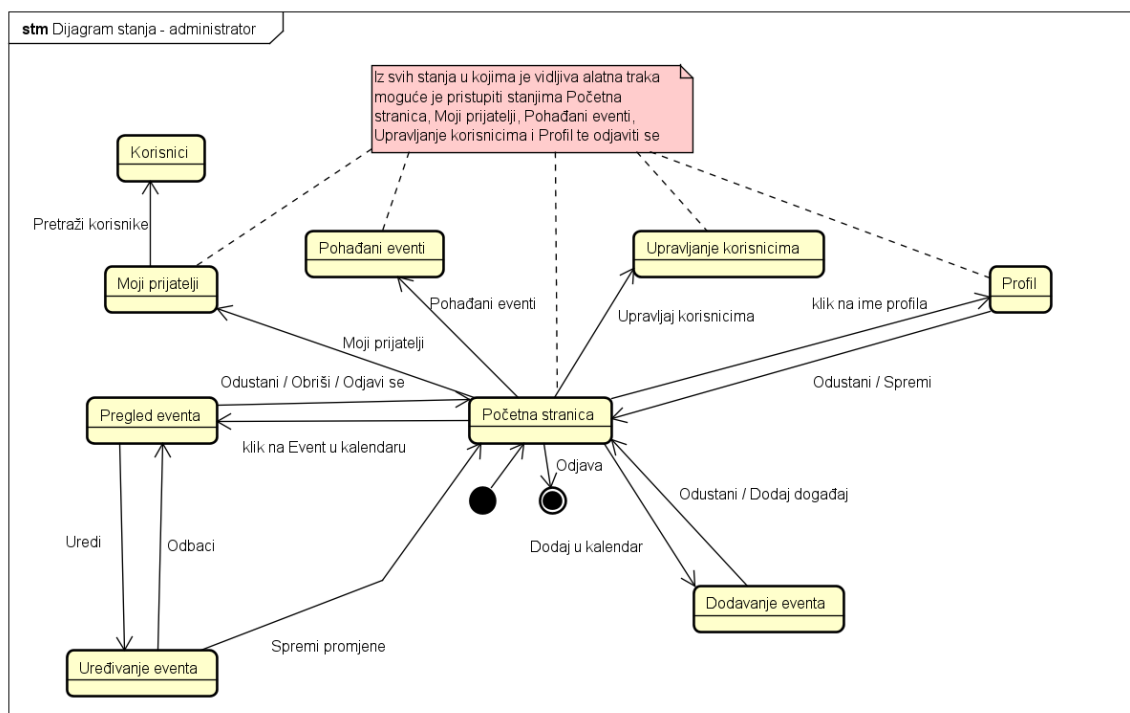


Slika 4.6: Dijagram razreda - kontroleri, servisi i repozitoriji



Slika 4.7: Dijagram razreda - modeli

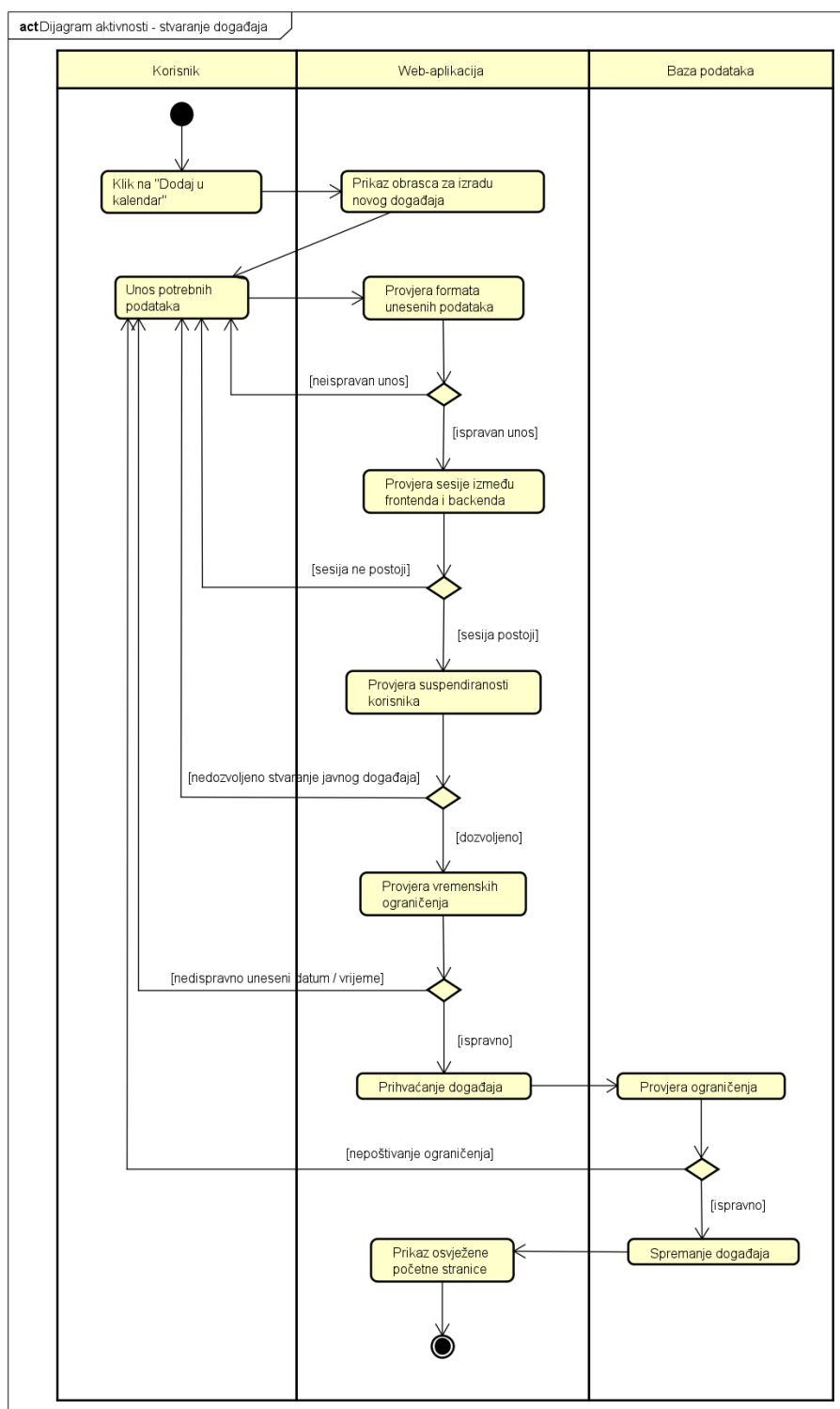
## 4.3 Dijagram stanja



Slika 4.8: Dijagram stanja

Dijagram stanja opisuje dinamičko ponašanje sustava uslijed raznih mogućih događaja. Uspješnom prijavom prikazana je početna stranica s kalendarom s koje preko alatne trake možemo pristupiti stranicama Moji prijatelji, Pohadani event, Upravljanje korisnicima (samo moderator i administrator) te vlastitom korisničkom profilu. Klik na *eventko* logotip uvijek vraća na početnu stranicu. Opcija *Dodaj u kalendar* otvara izbornik za stvaranje novog eventa, a postojeće evente moguće je klikom na njih u kalendaru pregledavati i uređivati.

## 4.4 Dijagram aktivnosti



Slika 4.9: Dijagram aktivnosti za stvaranje novog događaja

Dijagramom aktivnosti modeliran je tijek zadataka i poslova pri korisnikovom stvaranju novog događaja. Format podataka koji se provjerava na frontendu mora sadržavati stavke *name*, *location*, *beginningTimestamp*, *endTimeStamp*, *description*, *typeId*, *tagIds*, svi podaci moraju biti predviđenog tipa, a *typeId* mora biti 1, 2 ili 3. Ukoliko je korisnik trenutno suspendiran, a pokušava stvoriti javni događaj, to će mu biti onemogućeno. Vremenska ograničenja provjeravaju se na backendu i automatski na bazi podataka kako ne bi bilo moguće stvoriti događaj u prošlosti niti postaviti datum i vrijeme početka nakon datuma i vremena završetka. Ukoliko su sve provjere uspješne događaj se stvara u bazi podataka, a korisniku se pokazuje ažurirana početna stranica s njegovim novim događajem upisanim u kalendar.

## 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

#### *dio 2. revizije*

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

## 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:*

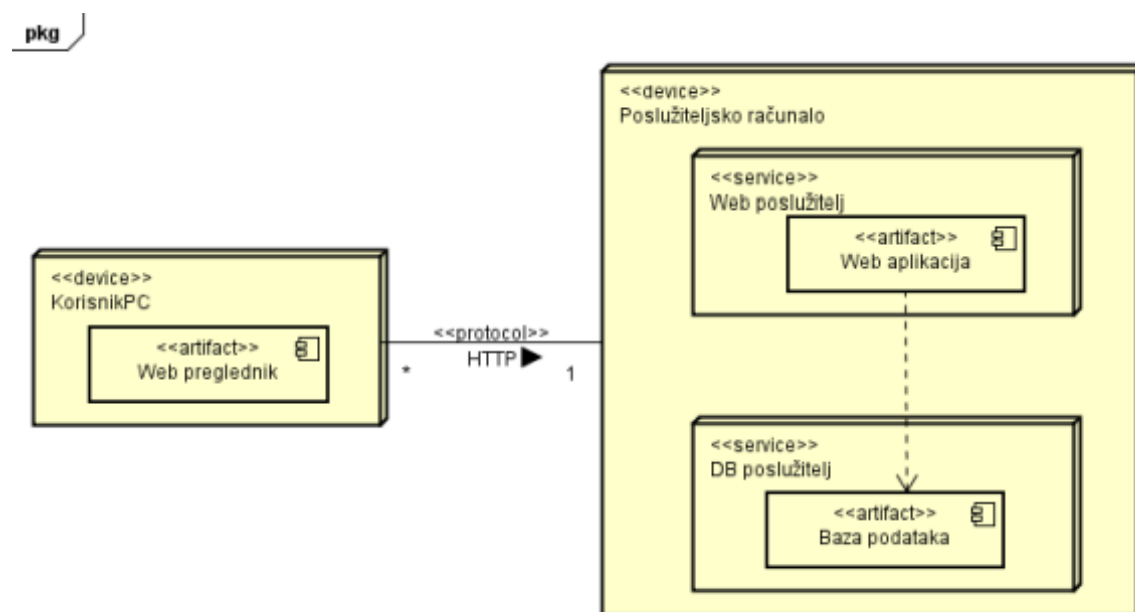
- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

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

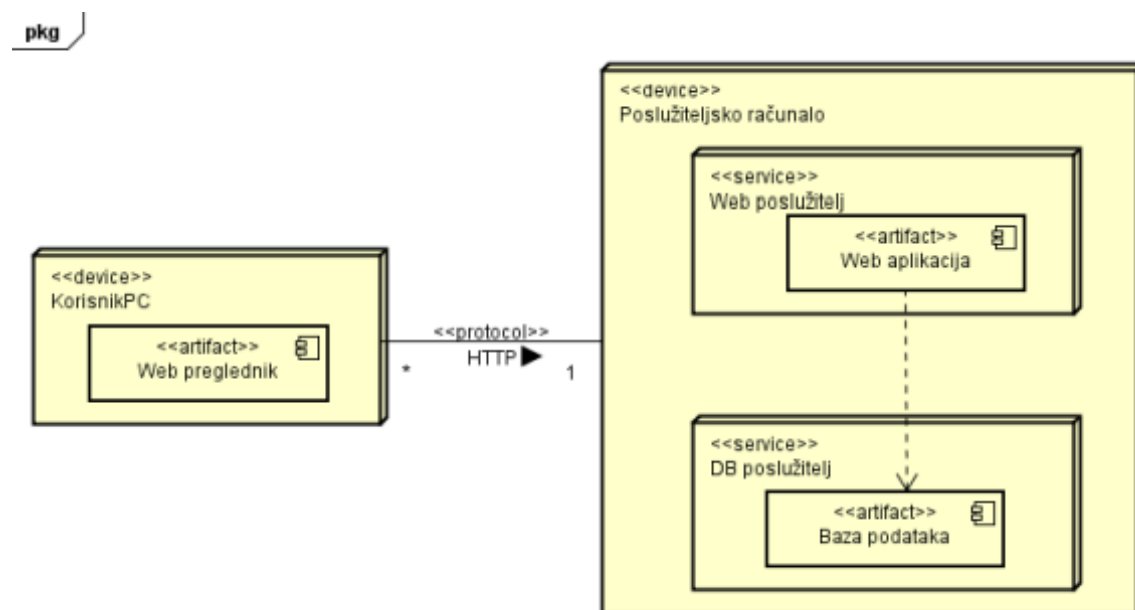
---

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

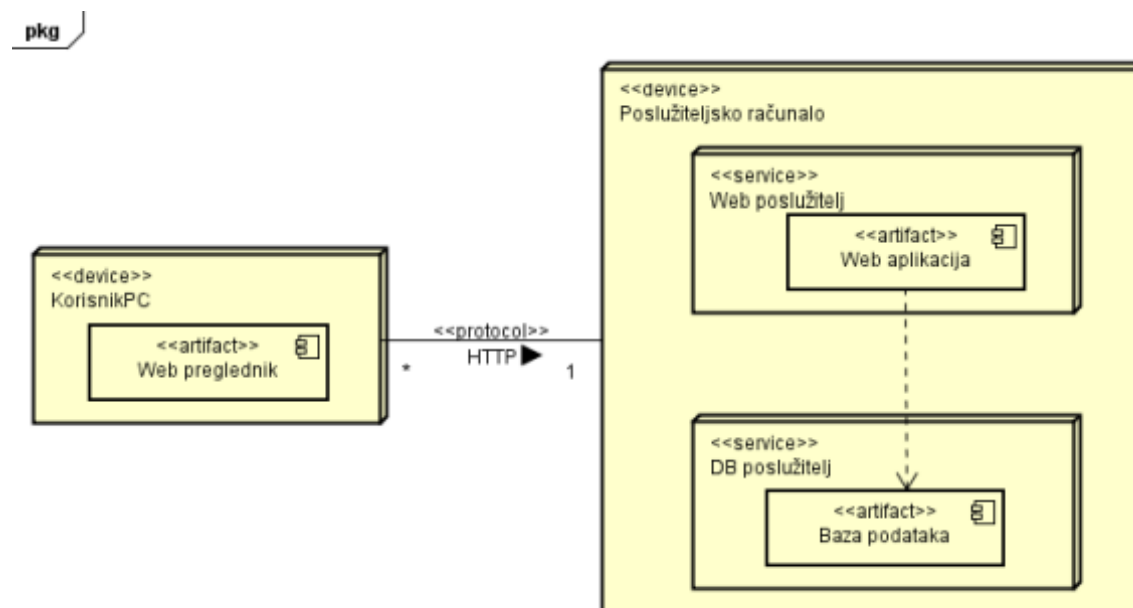




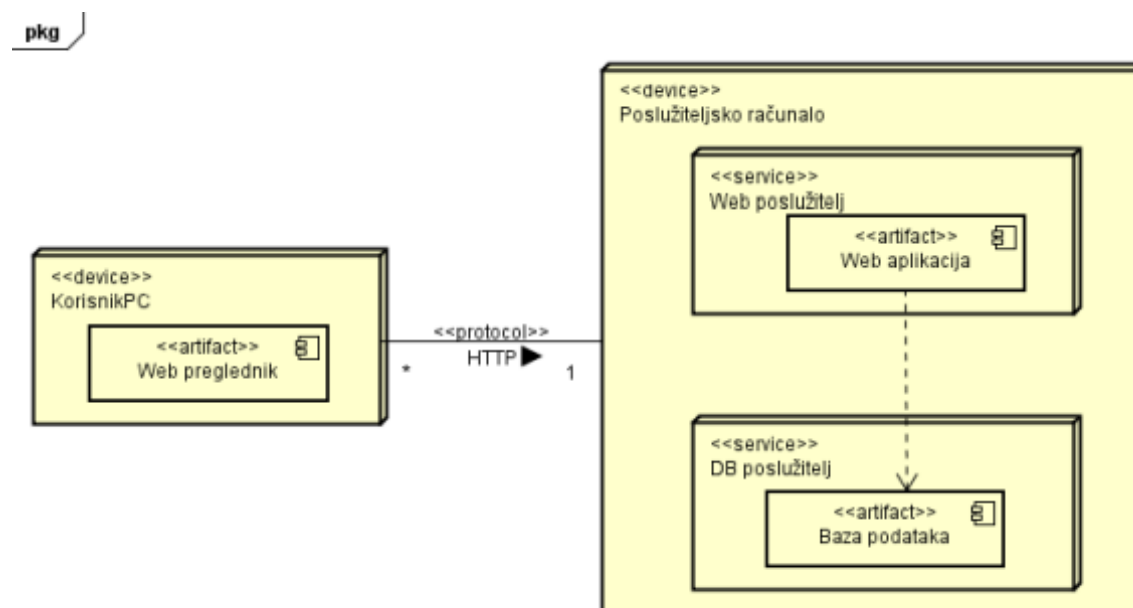
Slika 5.1: Dijagram razmještaja



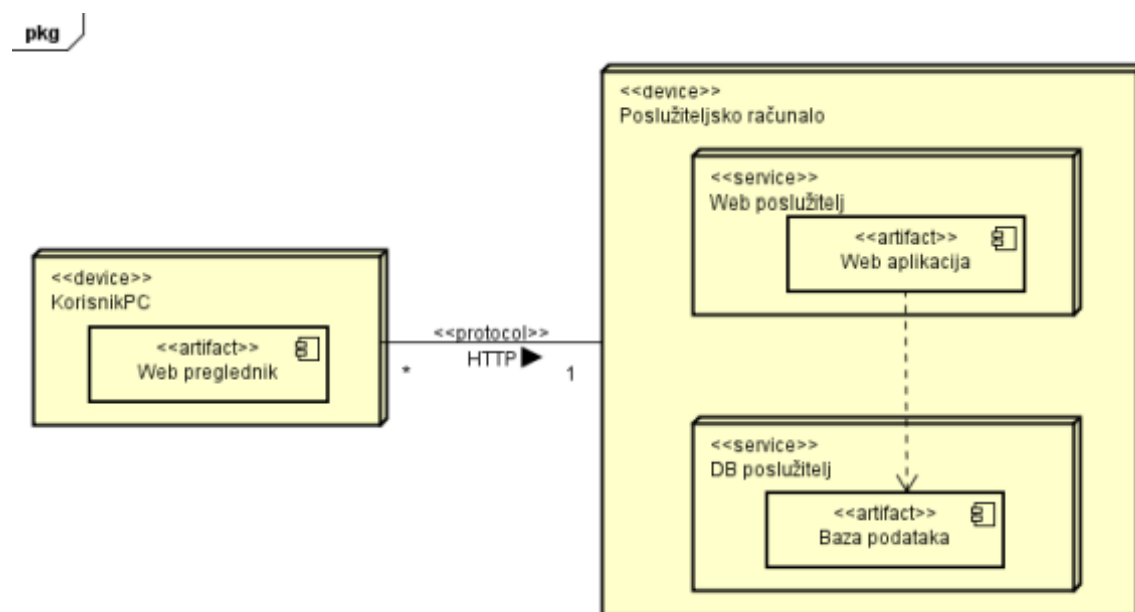
Slika 5.2: Dijagram razmještaja



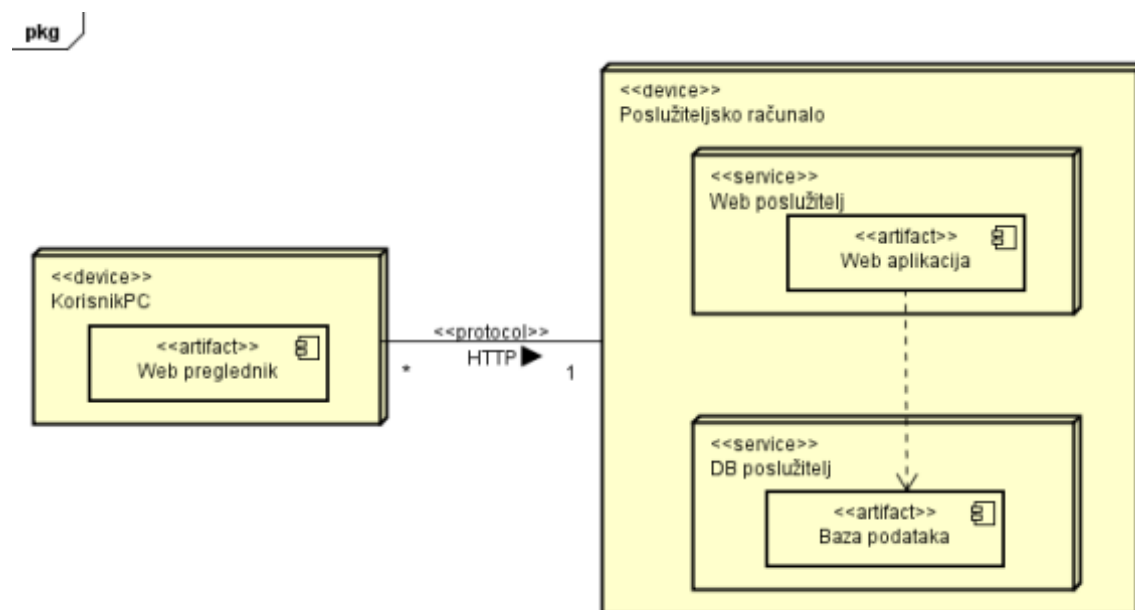
Slika 5.3: Dijagram razmjesta



Slika 5.4: Dijagram razmjesta



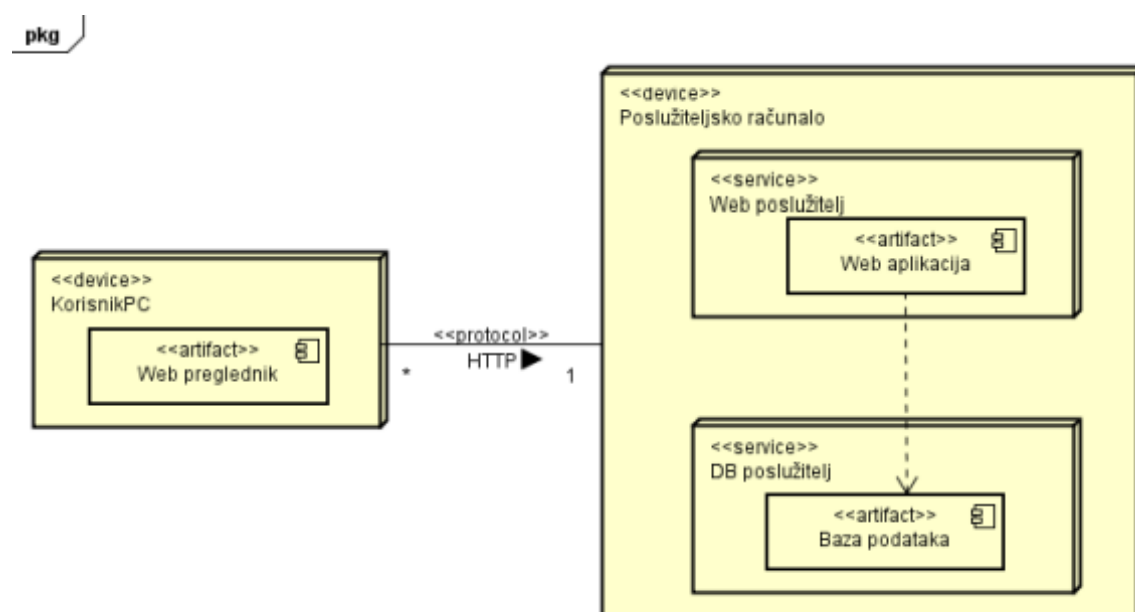
Slika 5.5: Dijagram razmjesta



Slika 5.6: Dijagram razmjesta

## 5.3 Dijagram razmještaja

Dijagram razmještaja opisuje topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom okruženju. Na poslužiteljskom računalu se nalaze web poslužitelj na renderu i poslužitelj baze podataka na vercelu. Sustav je baziran na arhitekturi "klijent - poslužitelj", a komunikacija između računala korisnika (običan korisnik, premium korisnik, pregledavač, moderator, administrator) i poslužitelja odvija se preko HTTP veze.



Slika 5.7: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

Puštanje web aplikacije u pogon sastoji se od tri segmenta:

- Stvaranje baze podataka
- Puštanje backenda u pogon
- Puštanje frontenda u pogon

### Stvaranje baze podataka

Baza podataka besplatno je spremljena na web-oblaku render.com. Ona se postavlja i osposobljava sljedećim koracima:

#### 1. Stvaranje nove baze

- Za početak je potrebno (nakon registracije na render.com) odabrati opciju *New* te na padajućem izborniku *PostgreSQL*
- Pojavljuje se izbornik koji ispunjavamo kao na slici ispod
- Biramo *Create Database*

**New PostgreSQL**

Name ○ Eventko - baza podataka

Database ○ eventko\_2nhy

User eventko\_admin

Region  
The region where your Database runs. Services must be in the same region to communicate privately and you currently have services running in Frankfurt. Frankfurt (EU Central) ▾

PostgreSQL Version 14 ▾

Datadog API Key ○

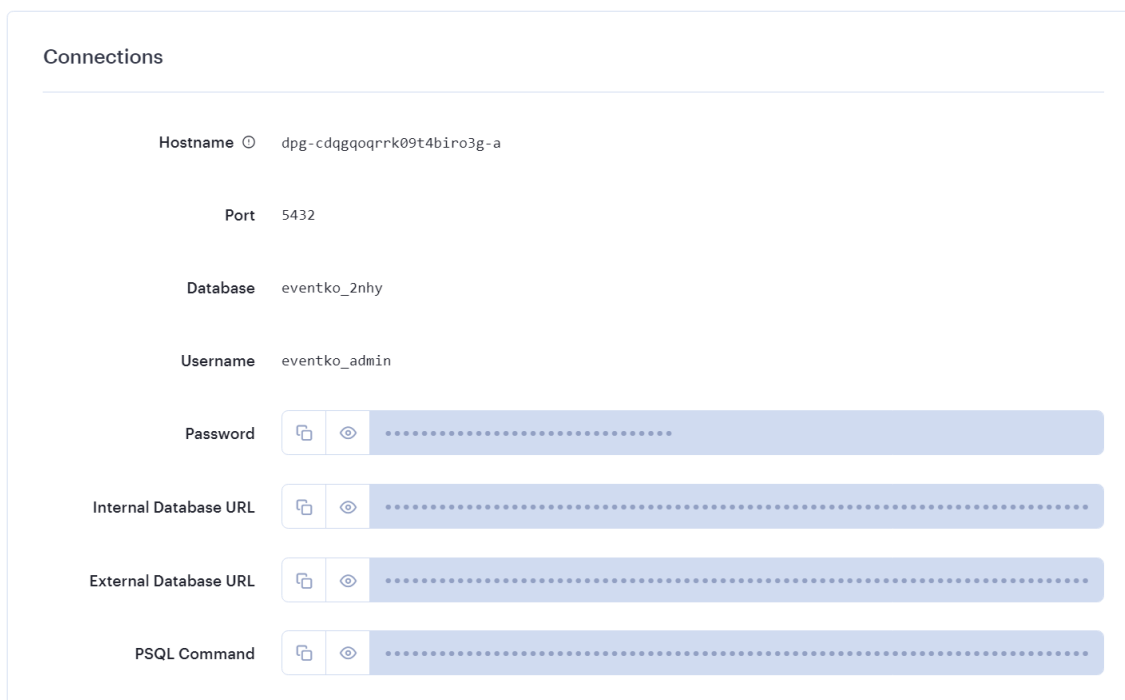
Please enter your payment information to select an instance type with higher limits.

Instance Type	RAM	CPU	Storage	Price
<input checked="" type="radio"/> Free	256 MB	Shared	1 GB	\$0 / month

Slika 5.8: Stvaranje nove PostgreSQL baze

## 2. Dohvat podataka za spajanje

- Redom biramo *Dashboard* -> *Eventko* - baza podataka -> *Info*
- Skrolamo do izbora *Connections* gdje vidimo informacije potrebne za spajanje na bazu (slika ispod).



Connections

Hostname ⓘ dpg-cdqgqorrk09t4biro3g-a

Port 5432

Database eventko\_2nhy

Username eventko\_admin

Password ⓘ [Masked]

Internal Database URL ⓘ [Masked]

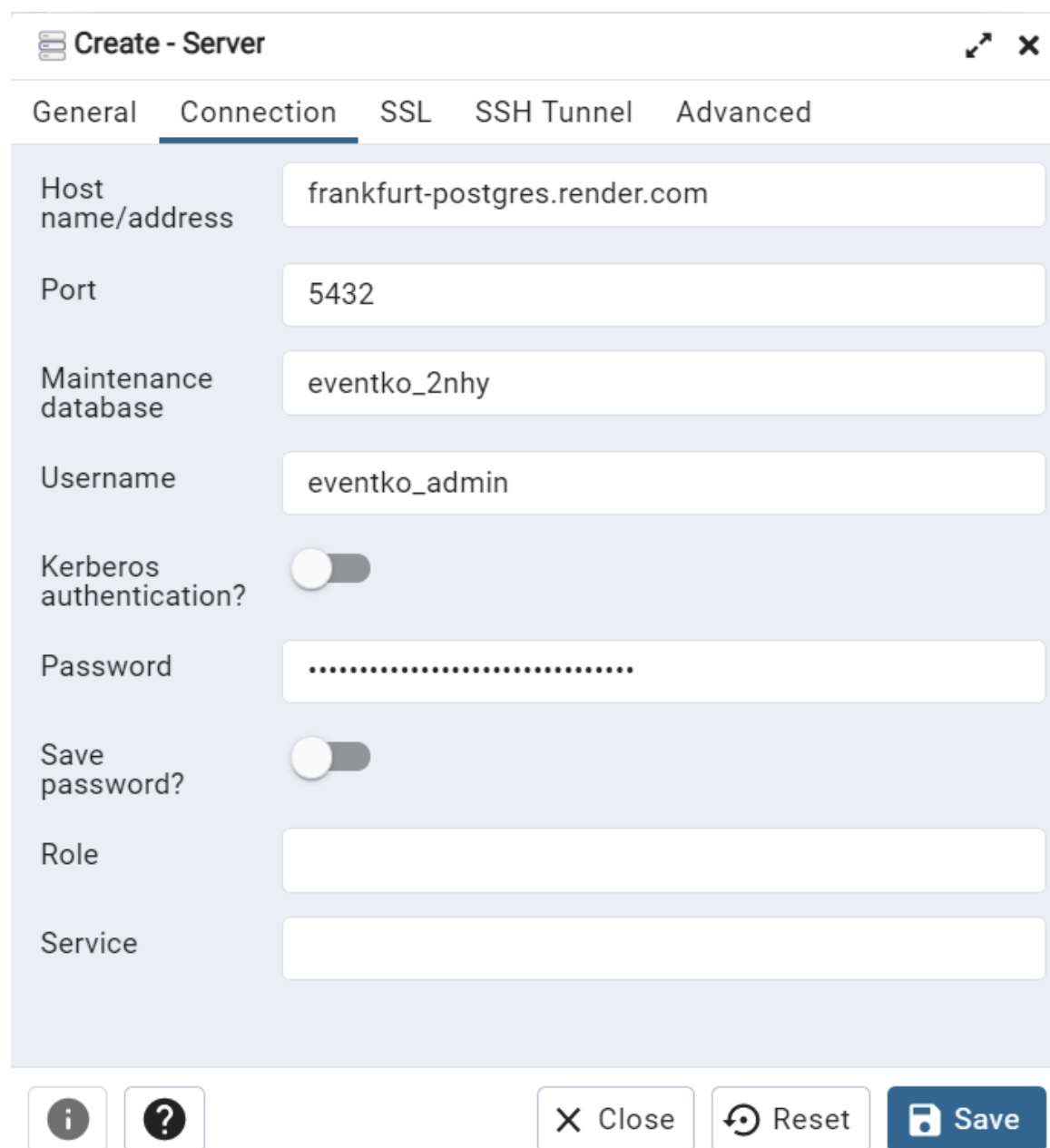
External Database URL ⓘ [Masked]

PSQL Command ⓘ [Masked]

Slika 5.9: Dohvat podataka za spajanje baze

## 3. Spajanje putem pgAdmina

- Izabiremo *Object* -> *Create* -> *Server*
- Na kartici *General* unosimo ime servera kojeg otvaramo u pgAdminu
- Na kartici *Connection* unosimo podatke kao na slici i spremamo ih sa *Save*



**Create - Server**

General **Connection** SSL SSH Tunnel Advanced

Host name/address: frankfurt-postgres.render.com

Port: 5432

Maintenance database: eventko\_2nhy

Username: eventko\_admin

Kerberos authentication? ☐

Password: .....

Save password? ☐

Role:

Service:

*[Info] [Help] [Close] [Reset] [Save]*

Slika 5.10: Unos podataka za spajanje baze

#### 4. Spajanje iz backenda

- U src/main/resources/application.properties upisujemo naredbe sa slike



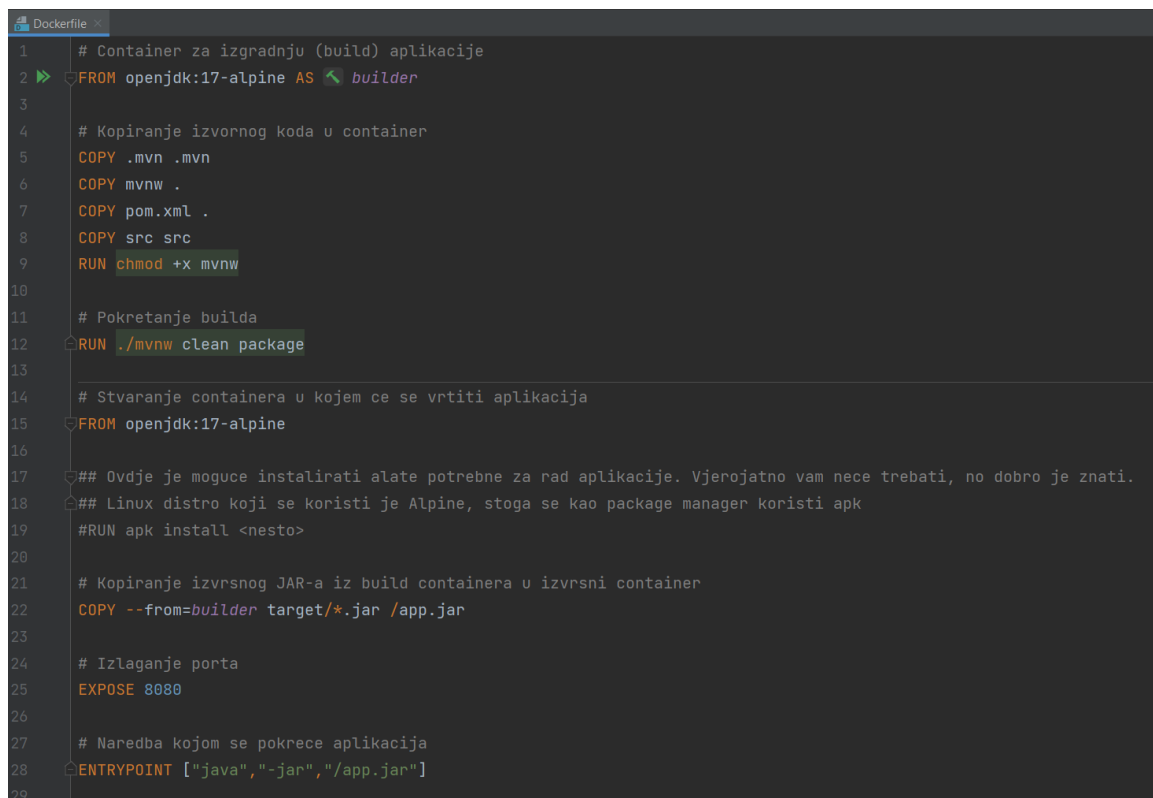
```
1 spring.datasource.url=jdbc:postgresql://frankfurt-postgres.render.com:5432/eventko_2nhy
2 spring.datasource.username=eventko_admin
3 spring.datasource.password=
4
```

Slika 5.11: Spajanje baze iz backenda

## Puštanje backenda u pogon

Backend dio projekta također koristi render.com za puštanje u pogon. Preduvjeti za to su:

- Dodavanje Dockerfilea prikazanog na slici ispod
- Povezivanje s GitLabom na renderu



```
1 # Container za izgradnju (build) aplikacije
2 FROM openjdk:17-alpine AS builder
3
4 # Kopiranje izvornog koda u container
5 COPY .mvn .mvn
6 COPY mvnw .
7 COPY pom.xml .
8 COPY src src
9 RUN chmod +x mvnw
10
11 # Pokretanje builda
12 RUN ./mvnw clean package
13
14 # Stvaranje containera u kojem ce se vrtiti aplikacija
15 FROM openjdk:17-alpine
16
17 ## Ovdje je moguće instalirati alate potrebne za rad aplikacije. Vjerojatno vam neće trebati, no dobro je znati.
18 ## Linux distro koji se koristi je Alpine, stoga se kao package manager koristi apk
19 #RUN apk install <nesto>
20
21 # Kopiranje izvršnog JAR-a iz build containera u izvršni container
22 COPY --from=builder target/*.jar /app.jar
23
24 # Izlaganje porta
25 EXPOSE 8080
26
27 # Naredba kojom se pokreće aplikacija
28 ENTRYPOINT ["java", "-jar", "/app.jar"]
29
```

Slika 5.12: Potrebni Dockerfile

Stvaranje backenda postićemo na sljedeći način:

- Odabiremo na render.com *New -> Web Service*
- Odabiremo željeni GitLab repozitorij
- Dalje unosimo podatke kao na slici ispod i potvrđujemo s *Create Web Service*



**Name**  
A unique name for your web service.

eventko-backend

**Region**  
The [region](#) where your web service runs. Services must be in the same region to communicate privately and you currently have services running in **Frankfurt**.

Frankfurt (EU Central)

**Branch**  
The repository branch used for your web service.

develop

**Root Directory** Optional  
Defaults to repository root. When you specify a [root directory](#) that is different from your repository root, Render runs all your commands in the [specified directory](#) and ignores changes outside the directory.

Izvornikod/backend

**Environment**  
The runtime environment for your web service.

Docker

Please [enter your payment information](#) to select an instance type with higher limits.

Instance Type	RAM	CPU	Price
<input checked="" type="radio"/> Free	512 MB	Shared	\$0 / month

Slika 5.13: Stvaranje novog web servisa

## Puštanje frontenda u pogon

Frontend dio projekta puštamo u pogon servisom [vercel.com](https://vercel.com). Preduvjeti za to su:

- Dodavanje `vercel.json` datoteke u projekt (slika ispod) radi mapiranja putanja prema backendu
- Povezivanje s GitLabom i stvaranje grupe na vercelu

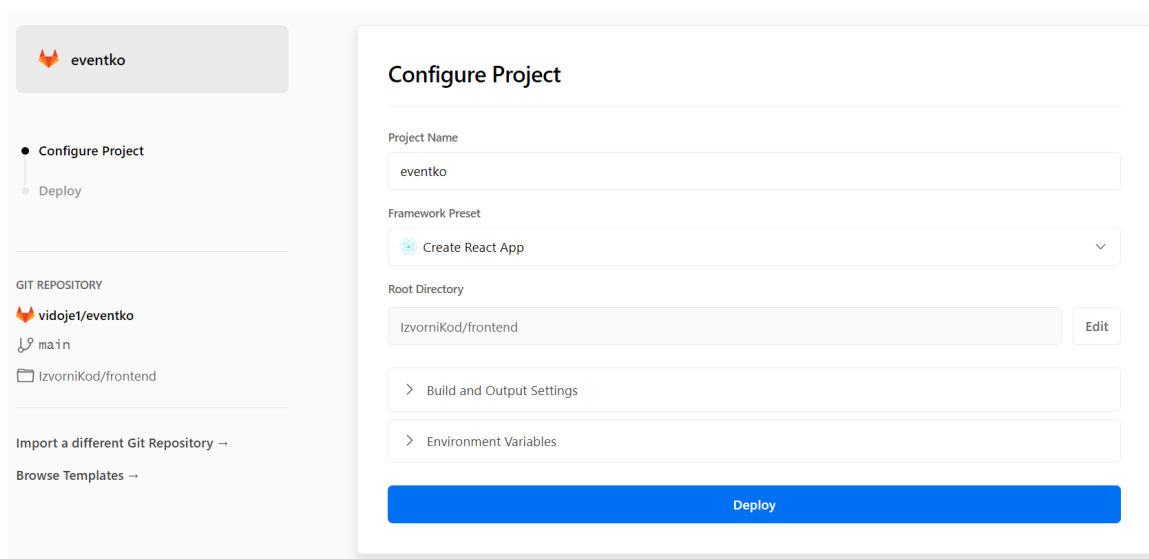
```
{} vercel.json X
{} vercel.json > ...
1  {
2    "rewrites": [
3      {
4        "source": "/api/:match*",
5        "destination": "https://eventko-backend.onrender.com/api/:match*"
6      },
7      {
8
9        "source": "/(.*)",
10       "destination": "/"
11     }
12   ]
13 }
```

Slika 5.14: Potrebna `vercel.json` datoteka

Za puštanje frontenda u pogon provode se sljedeći koraci:

## 1. Stvaranje

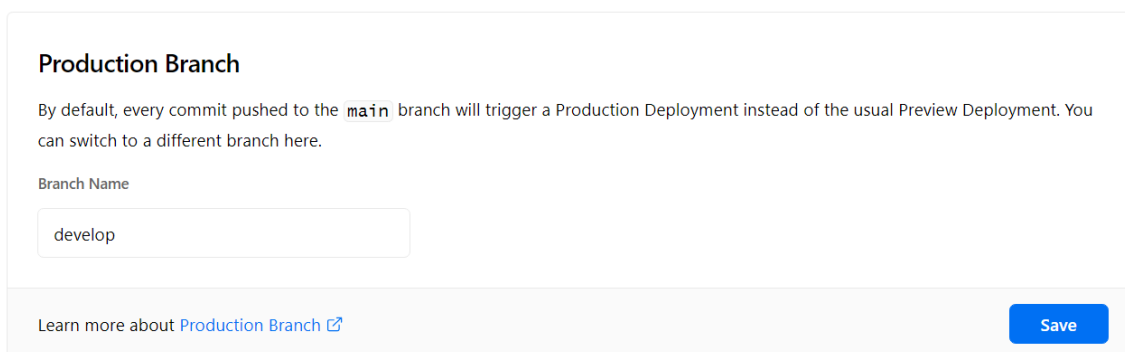
- Odabiremo grupu *Vidoje* od ponuđenih iz našeg GitLab računa
- Biramo redom *Add new* pa *Import git repository*, pri čemu biramo repozitorij Eventko
- Odabiremo *Configure Project*, ispunjavamo polja kao na slici ispod i završavamo s *Deploy*



Slika 5.15: Stvaranje React aplikacije i puštanje u pogon

## 2. Promjena grane

- Biramo redom *Overview* -> *Eventko* -> *Settings* -> *Git*
- Mijenjamo granu iz *main* u *develop* kao na slici ispod te spremamo sa *Save*



Slika 5.16: Promjena razvojne grane

### 3. Izmjena dodijeljene domene web-stranice (opcionalno)

- Biramo redom *Overview* -> *Eventko* -> *Settings* -> *Domains*
- Unosimo novu željenu domenu kao na slici ispod i spremamo sa *Save*

#### Domains

These domains are assigned to your Production Deployments. Optionally, a different [Git branch](#) or a [redirection](#) to another domain can be configured for each one.

Add

Domain

eventko.vercel.app

Redirect to ⓘ

No Redirect ▾

Git Branch ⓘ

develop

Remove

Cancel

Save

Slika 5.17: Promjena domene web-stranice

## 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. W3Schools, <https://www.w3schools.com/>
3. ReactJS, <https://reactjs.org>
4. Baledung, <https://www.baeldung.com>
5. Spring Boot, <https://spring.io/projects/spring-boot>
6. Latex Forum, <https://latex.org/forum/index.php>
7. Overleaf, <https://www.overleaf.com/learn>
8. The Unified Modeling Language, <https://www.uml-diagrams.org>
9. Astah Community, <http://astah.net/editions/uml-new>

# Indeks slika i dijagrama

3.1	Dijagram obrazaca uporabe, funkcionalnosti različitih korisnika stranice . . . . .	16
3.2	Sekvencijski dijagram za UC1 . . . . .	18
3.3	Sekvencijski dijagram za UC2 . . . . .	20
4.1	Arhitektura sustava . . . . .	22
4.2	ER dijagram baze podataka . . . . .	29
4.3	ER dijagram baze podataka bez atributa . . . . .	30
4.4	Relacijski dijagram baze podataka . . . . .	30
4.5	Dijagram razreda - DTO-ovi . . . . .	31
4.6	Dijagram razreda - kontroleri, servisi i repozitoriji . . . . .	32
4.7	Dijagram razreda - modeli . . . . .	33
4.8	Dijagram stanja . . . . .	34
4.9	Dijagram aktivnosti za stvaranje novog događaja . . . . .	35
5.1	Dijagram razmještaja . . . . .	40
5.2	Stvaranje nove PostgreSQL baze . . . . .	41
5.3	Dohvat podataka za spajanje baze . . . . .	42
5.4	Unos podataka za spajanje baze . . . . .	43
5.5	Spajanje baze iz backenda . . . . .	43
5.6	Potrebni Dockerfile . . . . .	44
5.7	Stvaranje novog web servisa . . . . .	45
5.8	Potrebna vercel.json datoteka . . . . .	45
5.9	Stvaranje React aplikacije i puštanje u pogon . . . . .	46
5.10	Promjena razvojne grane . . . . .	46
5.11	Promjena domene web-stranice . . . . .	47

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 20. listopada 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - sastanak s asistentom i demonstratorom
  - raščišćavanje elemenata projektnog zadatka
  - predaja alternativnog "Eventko" projekta

### 2. sastanak

- Datum: 24. listopada 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - razrada koncepta projekta
  - osmišljavanje dodatnih funkcionalnosti, aktera
  - prijedlog podjele uloga

### 3. sastanak

- Datum: 27. listopada 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - finalna podjela uloga
  - odabir radnog okruženja

### 4. sastanak

- Datum: 3. studenog 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - instalacija potrebne programske potpore
  - detaljna skica funkcionalnosti stranice
  - daljnja podjela rada

- dizajn originalnog logotipa

#### 5. sastanak

- Datum: 14. studenog 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - priprema za demonstraciju osnovnih funkcionalnosti
  - povezivanje backenda i frontenda
  - usklađivanje ograničenja

#### 6. sastanak

- Datum: 15. studenog 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - daljnja priprema za demonstraciju osnovnih funkcionalnosti
  - izrada UML, sekvencijskog te dijagrama razreda
  - dovršeno povezivanje frontenda i backenda
  - lokalna pohrana prijave

#### 7. sastanak

- Datum: 7. prosinca 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
  - priprema za prvo kolokviranje
  - izmjene dijagrama baze podataka



## Tablica aktivnosti

	Velimir Kovačić	Jakov Krčadinac	Ana Marić	Ema Nekić	Filip Perković	Fran Saganić	Luka Srića
Upravljanje projektom	7	5					
Opis projektnog zadatka	2	3			5	2	
Funkcionalni zahtjevi					2	1	
Opis pojedinih obrazaca		1			3		
Dijagram obrazaca	2				2	4	
Sekvencijski dijagrami						5	
Opis ostalih zahtjeva		1			2	1	
Arhitektura i dizajn sustava	2	1			3		
Baza podataka	3	2				5	
Dijagram razreda	2				2	3	
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati	7	5	5	5	7	7	4
Ispitivanje programskog rješenja	2	1	5	5			3
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja		2			3		
Zaključak i budući rad	1	1					
Popis literature	1						

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	<b>Velimir Kovačić</b>	<b>Jakov Krčadinac</b>	<b>Ana Marić</b>	<b>Ema Nekić</b>	<b>Filip Perković</b>	<b>Fran Saganić</b>	<b>Luka Srića</b>
Izrada početne stranice	1	1	4	3			3
Spajanje s bazom podataka	4						
Back end	12	2					
Deployment	8						
Front end			12	12			12

## Dijagrami pregleda promjena

### *dio 2. revizije*

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