Oblikovanje programske potpore

Ak. god. 2018./2019.

*Moj Kvart*

Dokumentacija, Rev. *1*

Grupa: *Javoljupci*

Voditelj: *Mate Gašparini*

Datum predaje: *28. studenoga 2018.*

Nastavnik: *Nikolina Frid*

Sadržaj

[1. Dnevnik promjena dokumentacije 3](#_Toc531196556)

[2. Opis projektnog zadatka 5](#_Toc531196557)

[3. Pojmovnik 7](#_Toc531196558)

[4. Funkcionalni zahtjevi 8](#_Toc531196559)

[Dijagrami obrazaca uporabe 14](#_Toc531196560)

[Sekvencijski dijagrami 17](#_Toc531196561)

[5. Ostali zahtjevi 24](#_Toc531196562)

[6. Arhitektura i dizajn sustava 25](#_Toc531196563)

[6.1. Svrha, opći prioriteti i skica sustava 25](#_Toc531196564)

[6.2. Dijagram razreda s opisom 29](#_Toc531196565)

[6.3. Dijagram objekata 31](#_Toc531196566)

[6.4. Ostali UML dijagrami 32](#_Toc531196567)

[7. Implementacija i korisničko sučelje 33](#_Toc531196568)

[7.1. Dijagram razmještaja 33](#_Toc531196569)

[7.2. Korištene tehnologije i alati 34](#_Toc531196570)

[7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 35](#_Toc531196571)

[7.4. Ispitivanje programskog rješenja 36](#_Toc531196572)

[7.5. Upute za instalaciju 37](#_Toc531196573)

[7.6. Korisničke upute 38](#_Toc531196574)

[8. Zaključak i budući rad 39](#_Toc531196575)

[9. Popis literature 40](#_Toc531196576)

[Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 41](#_Toc531196577)

[Dodatak B: Dnevnik sastajanja 42](#_Toc531196578)

[Dodatak C: Prikaz aktivnosti grupe 43](#_Toc531196579)

[Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 45](#_Toc531196580)

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.1 | Popisani dionici, razrađeni aktori i njihovi funkcionalni zahtjevi te napravljeno prvih sedam use-caseova. | Assouline, Dukić | 22.10.2018. |
| 0.2 | Napisan opis projektnog zadatka. | Dodigović | 22.10.2018. |
| 0.21 | Dodani use-caseovi od osmog do dvadesetog. | Dukić, Assouline | 23.10.2018. |
| 0.22 | Poboljšani obrasci uporabe. | Dukić, Assouline | 23.10.2018. |
| 0.23 | Sistematizirani obrasci uporabe. | Dukić | 26.10.2018. |
| 0.3 | Dodani dijagrami obrazaca uporabe. | Kršić, Kurtović | 27.10.2018. |
| 0.31 | Dodani sekvencijski dijagrami. | Kršić, Kurtović | 27.10.2018. |
| 0.32 | Obrisane funkcionalnosti vezane uz sustav poruka kod obrazaca uporabe. | Dukić | 28.10.2018. |
| 0.33 | Izmijenjeni dijagrami obrazaca uporabe i dodan sekvencijski dijagram. | Kršić | 3.11.2018. |
| 0.34 | Dodani dionici, poboljšane definicije funkcionalnih zahtjeva. | Assouline, Dukić | 3.11.2018. |
| 0.35 | Kombiniranje sekvencijskih dijagramima. | Kurtović | 5.11.2018. |
| 0.4 | Dodani ostali zahtjevi. | Gašparini | 5.11.2018. |
| 0.5 | Dodana skica arhitekture sustava. | Dodigović | 5.11.2018. |
| 0.6 | Dodana skica baze podataka. | Gašparini | 15.11.2018. |
| 0.61 | Dodan opis baze podataka. | Kurtović | 22.11.2018. |
| 0.7 | Dodan dijagram razreda. | Dodigović | 23.11.2018. |
| 0.71 | Dodan dijagram objekata. | Kršić | 24.11.2018. |
| 0.8 | Sitne izmjene baze podataka i oblikovanja. | Gašparini | 24.11.2018. |
| 0.81 | Izmjena dijagrama razreda. | Gašparini | 26.11.2018. |
| 0.9 | Dodani pojmovi u pojmovnik. | Dukić | 27.11.2018. |
| 0.91 | Dodan indeks. | Assouline | 27.11.2018. |
| 0.92 | Sitne pravopisne izmjene. | Assouline | 27.11.2018. |
| 0.93 | Izmjena dijagrama objekata. | Kršić | 27.11.2018. |
| 0.94 | Dodan opis sekvencijskih dijagrama. | Kršić | 27.11.2018. |
| 1.0 | Konačna verzija prve revizije. | Gašparini | 28.11.2018. |
| 1.1 | Dodani dijagrami stanja i razmještaja | Kurtović | 9.1.2018. |
| 1.2 | Dodani dijagrami komunikacije i aktivnosti | Kršić | 12.1.2019. |
| 1.3 | Dodani dijagrami komponenti, instalacijske upute i kratak opis važnih dijelova koda | Assouline | 16.1.2019. |

# Opis projektnog zadatka

Cilj ovog projekta je izrada web aplikacije koja će stanovnicima unutar odabrane gradske četvrti omogućiti bolju komunikaciju, lakšu razmjenu informacija te učinkovitiju organizaciju događaja.

Iako ljudi na dnevnoj razini prolaze pokraj svojih susjeda često ih niti ne upoznaju te time propuštaju priliku za moguću suradnju, druženje ili razmjenu korisnih informacija. Osim društvenog kontakta, stanovnike četvrti vežu stvari koje se tiču svakog pojedinačno. Samo neki od primjera su izvješće sa sjednice Vijeća četvrti, uređenje kvartovske okoline, mogući problemi oko zbrinjavanje otpada ili slično. Također, web aplikacija je prikladno mjesto za početak ostvarivanja odnosa na prethodno opisanim razinama zato što se ljudi zainteresirani za ostvarivanje komunikacije ujedinjuju na jednom mjestu i time si daju mogućnost za daljnji napredak u odnosima u stvarnom životu.

Stanovnik četvrti pri otvaranju računa upisuje osnovne informacije na temelju kojih se stvara njegov korisnički račun. Informacije potrebne za registraciju su:

* Ime
* Prezime
* Lozinka
* Adresa stanovanja
* Adresa e-pošte

Na temelju tih informacija aplikacija prema adresi stanovanja automatski određuje kojoj četvrti korisnik pripada i dodjeljuje mu se uloga stanovnika četvrti. Ostali korisnici kojima je dodijeljena viša uloga se pri otvaranju računa ne registriraju nego za njih račune stvaraju postojeći administratori sustava. Registracija je za sve korisnike, neovisno o ulozi, besplatna.

Da bi aplikacija zadovoljila sve funkcionalne zahtjeve koje četvrt zahtjeva, ona je sastavljena od tri osnovne cjeline. Prva cjelina je forum čija je svrha održavanje rasprava koje su tematski vezane uz aktualna događanja i sam život stanovnika u četvrti, sljedeća cjelina su događanja koja najavljuju buduće događaje u četvrti te posljednja, vijeće četvrti gdje se u kratkim natuknicama objavljuje sadržaj sjednica Vijeća gradske četvrti uz najavu novih tema.

Aplikacija razlikuje četiri vrste korisnika:

* Stanovnik četvrti
* Vijećnik
* Moderator četvrti
* Administrator aplikacije

Stanovniku četvrti je omogućeno vođenje rasprava unutar foruma i predlaganje najave događaja u četvrti. Pri definiranju prijedloga potrebno je definirati naziv, mjesto, vrijeme trajanja te kratak opis događaja.

Vijećnik uz mogućnosti redovnog stanovnika ima ovlasti pisanja objava u cjelini "Vijeće četvrti" gdje piše kratko izvješće sa zadnje sjednice.

Moderatori četvrti također imaju mogućnosti redovnih stanovnika, ali uz to pregledavaju najavljene događaje nakon čega ih odbijaju ili potvrđuju. Događaji se odbijaju ako sadrže vrijeđanje ili diskriminaciju na bilo kojoj razini. Potvrđene događaje objavljuju u rubrici "Događaji". Na raspolaganju im je i uređivanje najava zbog mogućih jezičnih pogrešaka. Osim toga, unutar foruma im je omogućeno brisanje objava neprikladnog sadržaja.

Administratori se ne smatraju stanovnicama niti jedne četvrti. Njihov je zadatak tehničke naravi te su oni zaduženi za upravljanje korisničkim računima, definiranje četvrti i njihovih granica.

Skup korisnika koji bi mogli biti zainteresirani za razvijeni sustav su stanovnici četvrti koji bi željeli uvesti neke promjene u svoju četvrt jer im se na ovaj način pruža lakši i učinkovitiji način diskusije, oni koji bi jednostavno htjeli poboljšati društvene odnose na razini kvarta i na kraju, oni koje zanima brži i lakši protok informacija vezanih uz život u kvartu.

# Pojmovnik

**BCrypt** – BCrypt je funkcija za sažimanje lozinke koju su 1999. dizajnirali Niels Provos i

David Mazières. Zasnovana je na Blowfish algoritmu. Funkcija se izvršava tako što se sažimanje izvršava više puta unutar petlje.

**HTTPS** – Internetski protokol nastao kombinacijom protokola HTTP i

protokola [SSL](https://hr.wikipedia.org/wiki/SSL)/[TLS](https://hr.wikipedia.org/wiki/TLS). HTTPS omogućava [kriptiranu](https://hr.wikipedia.org/w/index.php?title=Enkripcija&action=edit&redlink=1) komunikaciju i sigurnu identifikaciju web poslužitelja mreže. HTTPS veze se nerijetko koriste za novčane transakcije na [World Wide Webu](https://hr.wikipedia.org/wiki/World_Wide_Web) i ostale povjerljive transakcije u korporativnim informacijskim sustavima.

# Funkcionalni zahtjevi

**Dionici:**

* Stanovnik četvrti
* Vijećnik
* Razvojni programer
* Naručitelj proizvoda (FER)

**Aktori i njihovi funkcionalni zahtjevi:**

Aktori mogu biti inicijatori i sudionici. Inicijatori su stanovnici četvrti, vijećnici, moderatori četvrti i administratori aplikacije. Sudionik je baza podataka.

* **Stanovnik četvrti**
  + Može se registrirati
  + Može diskutirati o temama na forumu
  + Može predlagati najave budućih događanja u susjedstvu
* **Vijećnik**
  + Ima sve ovlasti kao i stanovnik četvrti (osim registracije)
  + Ima ovlast pisanja objava u cjelini „Vijeće četvrti“
* **Moderator četvrti**
  + Ima sve ovlasti kao i stanovnik četvrti (osim registracije)
  + Može pregledavati najave događanja koje su predložili stanovnici četvrti i objaviti ih u cjelini „Događanja“
  + Može uređivati prijedloge stanovnika četvrti u cjelini „Događanja“
  + Može odbaciti prijedloge neprikladnog sadržaja iz cjeline „Događanja“
  + Može ukloniti objave neprikladnog sadržaja s foruma
* **Administrator aplikacije**
  + Otvara račune za vijećnike, moderatore i druge administratore
  + Upravlja korisničkim računima
  + Definira četvrti i područja koja one obuhvaćaju
* **Baza podataka**
  + Pohranjuje podatke o svim korisnicima
  + Pohranjuje podatke o četvrtima i ulicama od kojih se one sastoje
  + Pohranjuje objave s foruma
  + Pohranjuje objave vijećnika gradske četvrti iz cjeline „Vijeće četvrti“
  + Pohranjuje najave iz cjeline „Događanja“

**UC1** Registracija

**Glavni sudionik** Stanovnik četvrti

**Rezultat** Stvaranje novog korisničkog računa za stanovnika četvrti

**Ostali sudionici** Baza podataka

**Željeni scenarij** 1. Korisnik upisuje svoje ime, prezime, željeno korisničko ime,

lozinku i adresu stanovanja (ulica u kojoj stanuje)

                    2. Koristeći unesene podatke, sustav stvara novi korisnički račun

3. Podaci o novostvorenom korisniku dodaju se u bazu podataka

**Mogući scenarij** 1. Za unesenu e-mail adresu već postoji korisnički račun

2. Sustav odbacuje pokušaj registracije i javlja grešku korisniku

**UC2**                Otvaranje povlaštenih korisničkih računa

**Glavni sudionik** Administrator

**Rezultat** Administrator stvara korisnički račun za vijećnika, moderatora ili

drugog administratora; ovaj postupak ima isti rezultat kao i registracija kod stanovnika četvrti

**Ostali sudionici** Vijećnik, moderator, baza podataka

**Preduvjeti** 1. Administrator je prijavljen u sustav

**Željeni scenarij** 1. Administrator popunjava obrazac za dodavanje novog

korisničkog računa

2. Pritiskom na gumb „Stvori korisnika” novi korisnik se stvara u

bazi podataka s ulogom koju mu je dodijelio administrator

**Mogući scenarij** 1. Za unesenu e-mail adresu već postoji korisnički račun

2. Sustav odbacuje pokušaj stvaranja novog računa i javlja

grešku administratoru

**UC3** Prijava

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator, administrator

**Rezultat** Prijava u sustav čime se omogućuje korisniku rad u aplikaciji

**Ostali sudionici** Baza podataka

**Željeni scenarij** 1. Korisnik upisuje svoje korisničko ime i lozinku

2. Sustav prepoznaje važeću prijavu i daje korisniku

odgovarajuće ovlasti

**Mogući scenarij**1. Korisnik je upisao nepostojeće korisničko ime ili neispravnu

lozinku

                    2. Sustav odbacuje pokušaj prijave

3. Sustav o tome obavještava korisnika

**UC4** Odjava

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator, administrator

**Rezultat** Korisnik se odjavljuje iz sustava i vraća na početnu stranicu

foruma

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1.Korisnik pritišće gumb „Odjavi se”

                    2. Sustav završava korisničku sjednicu

**UC5**                        Objavljivanje na forumu

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Stvaranje nove objave

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1. Korisnik upisuje tekst objave i potvrđuje unos

2. Sustav prihvaća objavu i prikazuje ju u odabranoj temi na

forumu

**UC6**                     Brisanje vlastite objave s foruma

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Objava je obrisana iz baze podataka

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1. Korisnik potvrđuje svoj odabir

                    2. Sustav briše objavu iz baze podataka

**UC7** Uređivanje vlastitih objava s foruma

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Promjena sadržaja objave i pohranjivanje novog sadržaja u bazu podataka

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1. Korisnik upisuje novi sadržaj i potvrđuje ga

2. U bazi podataka se izmjenjuje sadržaj originalne objave

3. Sustav prikazuje novu verziju ostalim korisnicima

**UC8** Pregled objava na forumu

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Objave se prikazuju korisniku

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1. Korisnik pritišće gumb „Forum“

2. Korisnik bira temu koju želi pregledati

3. Objave u određenoj temi se prikazuju korisniku klikom na ime

teme

**UC9** Stvaranje teme na forumu

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Tema je otvorena na forumu i moguće je pisati objave u njoj

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen

**Željeni scenarij**   1. Korisnik pritišće gumb „Forum”

2. Korisnik stvara novu temu pritiskom na gumb „Stvori temu“ i

unosi ime nove teme

**UC10** Brisanje teme s foruma

**Glavni sudionik** Moderator

**Rezultat** Tema je obrisana iz baze podataka zajedno sa svim objavama

koje su bile u njoj

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Moderator je prijavljen

**Željeni scenarij**   1. Moderator pritišće gumb „Forum”

2. Moderator odabire temu i pritišće gumb „Obriši temu”

3. Tema se uklanja iz baze podataka i iz cjeline „Forum”

**UC11** Uklanjanje objava s foruma

**Glavni sudionik** Moderator

**Rezultat** Objava je obrisana iz baze podataka

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Moderator je prijavljen

**Željeni scenarij**   1. Moderator pritišće gumb „Obriši”

2. Objava se uklanja iz baze podataka

**UC12**                      Prijedlog najave događaja

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Najava događaja poslana je na razmatranje moderatoru koji je

zadužen za dotičnu četvrt

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1.Korisnik potvrđuje unos

                    2. Prijedlog se pohranjuje u bazu podataka

                    3. Uneseni prijedlog se prikazuje moderatoru

**UC13** Objavljivanje u cjelini „Događanja“

**Glavni sudionik** Moderator

**Rezultat** Moderator dobiva prijedlog objave od stanovnika četvrti i odlučuje hoće li ga odobriti ili odbaciti

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Moderator je prijavljen

**Mogući scenarij** a)

1. Moderator objavljuje najavu pritiskom na gumb „Objavi”

2. Odobrena najava se pohranjuje u bazu podataka

3. Odobrena najava postaje javna i vidljiva svim korisnicima

aplikacije

b)

1. Moderator pregledava prijedlog pritiskom na gumb „Uredi”

2. Moderator unosi potrebne izmjene

3. Moderator pohranjuje prijedlog u bazu podataka pritiskom na

gumb „Pohrani prijedlog” ili ga objavljuje odmah nakon uređivanja pritiskom na gumb „Objavi”

c)

1. Moderator pritišće gumb „Obriši”

2. Prijedlog se uklanja iz baze podataka

**UC14** Pregled odobrenih objava iz cjeline „Događanja“

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Odobrene objave se prikazuju korisniku

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1. Korisnik pritišće gumb „Događanja“

2. Objave se dohvaćaju iz baze podataka i prikazuju korisniku

**UC15** Objavljivanje u cjelini „Vijeće četvrti“

**Glavni sudionik** Vijećnik

**Rezultat** Objave (kratka izvješća i najave budućih tema) koje pišu vijećnici se stvaraju i moguće ih je pregledavati

**Ostali sudionici**Baza podataka

**Preduvjeti** 1. Vijećnik je prijavljen u sustav

**Željeni scenarij** 1. Vijećnik upisuje tekst objave i potvrđuje unos

2. Sustav prihvaća objavu i prikazuje ju u cjelini „Vijeće četvrti“

**UC16**                     Pregled objava vijećnika iz cjeline „Vijeće četvrti“

**Glavni sudionik** Stanovnik četvrti, vijećnik, moderator

**Rezultat** Kratka izvješća koja pišu vijećnici se prikazuju korisniku

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Korisnik je prijavljen u sustav

**Željeni scenarij** 1. Korisnik pritišće gumb „Vijeće četvrti“

2. Objave iz cjeline „Vijeće četvrti“ se dohvaćaju iz baze podataka i prikazuju korisniku

**UC17** Dodavanje četvrti

**Glavni sudionik** Administrator

**Rezultat** Nova četvrt je definirana ovom akcijom

**Ostali sudionici** Moderator, baza podataka

**Preduvjeti** 1. Administrator je prijavljen

**Željeni scenarij** 1. Administrator definira novu četvrt i unosi ulice koje se nalaze

u novodefiniranoj četvrti

2. Administrator odabire moderatora za četvrt

3. Podaci se pohranjuju u bazu podataka

**Mogući scenarij** 1. Četvrt koja se dodaje postoji u bazi podataka

2. Sustav odbacuje zahtjev za stvaranje nove četvrti

**UC18** Uređivanje četvrti

**Glavni sudionik** Administrator

**Rezultat** Administrator dodaje neku novu ulicu u sastavni dio određene četvrti

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Administrator je prijavljen

**Željeni scenarij** 1. Administrator nalazi četvrt koju treba urediti preko tražilice

koristeći ime četvrti

2. Administrator pritiskom na gumb „Dodaj ulicu” dobiva pristup

tekstualnom polju za upis ulice

3. Pri unosu imena ulice i pritiskom na gumb „Ok” ulica se dodaje

u sastav četvrti, a time i u bazu podataka

**UC19** Brisanje četvrti

**Glavni sudionik** Administrator

**Rezultat** Četvrt je obrisana iz baze podataka

**Ostali sudionici** Baza podataka

**Preduvjeti** 1. Administrator je prijavljen

2. Četvrt postoji u bazi podataka

**Željeni scenarij** 1. Administrator nalazi četvrt koju treba obrisati preko tražilice

koristeći ime četvrti

2. Administrator pritiskom na gumb „Obriši četvrt” nepovratno

briše četvrt iz baze podataka

# Dijagrami obrazaca uporabe

# Vršni dijagram obrazaca uporabe:



Slika 1: Vršni dijagram obrazaca uporabe

# Rad na forumu:



Slika 2: Dijagram obrazaca uporabe - Rad na forumu

# Cjelina "Događanja":



Slika 3: Dijagram obrazaca uporabe - cjelina "Događanja"

# Cjelina "Vijeće četvrti":



Slika 4: Dijagram obrazaca uporabe - cjelina "Vijeće četvrti"

# Rad sa četvrtima:



Slika 5: Dijagram obrazaca uporabe - Rad sa četvrtima

# Sekvencijski dijagrami

# UC2: Otvaranje povlaštenih korisničkih računa

Povlašteni korisnički račun može otvoriti samo administrator. Administrator šalje zahtjev za stvaranje povlaštenog korisničkog računa s odgovarajućim podacima aplikaciji "Moj Kvart" koja zatim provjerava dostupnost e-mail adrese. U slučaju da e-mail adresa nije dostupna, aplikacija administratoru prikazuje odgovarajuću poruku i time je završena radnja. Ukoliko je e-mail adresa dostupna, aplikacija šalje bazi podataka zahtjev za pohranom te dobiva njen odgovor o uspješnosti pohrane. Pohranom u bazu podataka je račun otvoren te se administratoru ispisuje poruka o uspjehu otvaranja računa.



Slika 6: UC2: Otvaranje povlaštenih korisničkih računa

# UC5 i UC8: Objavljivanje na forumu i Pregled objava na forumu

Korisnik pregledava forum posredno putem aplikacije koja dohvaća podatke iz baze podataka - odabirom teme šalje zahtjev aplikaciji koja šalje zahtjev za dohvat podataka bazi podataka. Na sličan način i objavljuje - unosi tekst objave kojeg zahtjevom šalje aplikaciji te ona zatim šalje zahtjev za pohranom podatka bazi podataka.



Slika 7: UC5 i UC8: Objavljivanje na forumu i Pregled objava na forumu

# UC9: Stvaranje teme na forumu

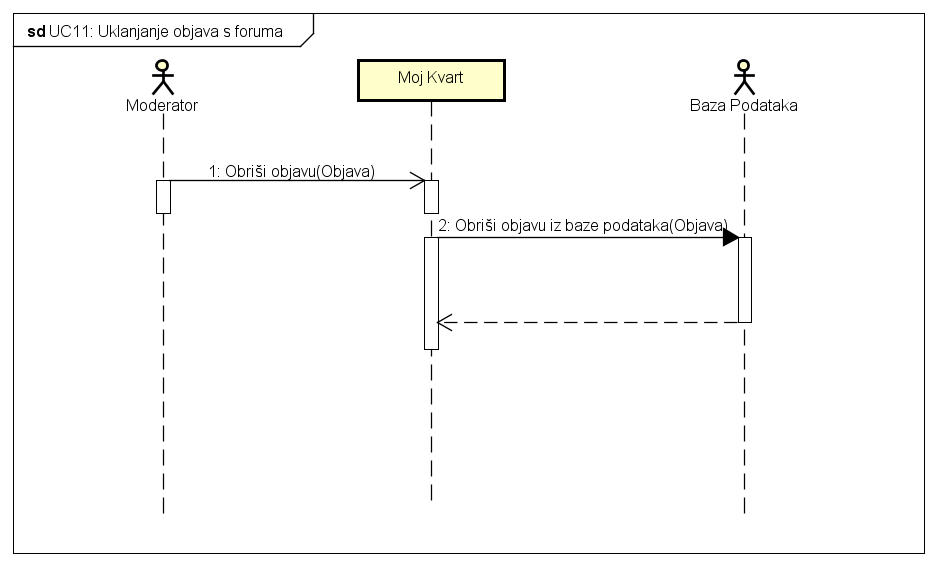
Korisnik odabire stvaranje nove teme te unosi ime nove teme - taj zahtjev se šalje aplikaciji koja pohranjuje novu temu u bazu podataka. Aplikacija nakon uspješnog stvaranja nove teme korisniku istu i prikazuje.



Slika 8: UC9: Stvaranje teme na forumu

# UC11: Uklanjanje objava s foruma

Moderator može obrisati objavu s foruma odabirom brisanja čime se šalje zahtjev aplikaciji koja zatim šalje zahtjev bazi podataka za uklanjanje odgovarajućih podataka.



Slika 9: UC11: Uklanjanje objava s foruma

# UC12 i UC13: Prijedlog najave događanja i Objavljivanje u cjelini "Događanja"

Stanovnik četvrti šalje aplikaciji zahtjev za predlaganjem najave događanja s odgovarajućim podacima. Aplikacija pohranjuje primljene podatke u bazu podataka te ispisuje prijedlog moderatoru koji odlučuje želi li prijedlog odobriti, dodatno urediti ili odbiti. U slučaju dodatnog uređivanja, uređeni prijedlog može dobriti ili pohraniti za daljnje uređivanje te se aplikaciji time šalje odgovarajući zahtjev kao i u slučaju inicijalnog prihvaćanja ili odbijanja. Aplikacija na temelju odgovora moderatora obavlja odgovarajuće izmjene u bazi podataka.



Slika 10: UC12 i UC13: Prijedlog najave događaja i Objavljivanje u cjelini "Događanja"

# UC18 i UC19: Uređivanje četvrti i brisanje četvrti

Administrator aplikaciji šalje zahtjev za dohvaćanjem željene četvrti. Aplikacija dohvaća željenu četvrt te prikazuje njene podatke administratoru. Administrator može urediti ili u potpunosti obrisati dohvaćenu četvrt. Dohvaćenoj četvrti može vršiti izmjene tako da dodaje ulice koje joj pripadaju nakon čega šalje zahtjev aplikaciji koja pohranjuje promjene u bazu podataka. Odabirom brisanja se šalje zahtjev aplikaciji koja zatim uklanja podatke te četvrti iz baze podataka.



Slika 11: UC18 i UC19: Uređivanje četvrti i Brisanje četvrti

# Ostali zahtjevi

* Korisničko sučelje mora biti oblikovano u skladu sa standardnim hrvatskim jezikom
* Sustav mora podržavati hrvatske dijakritičke znakove
* Sustav mora omogućiti paralelni rad više korisnika
* Vrijeme odziva u normalnim uvjetima ne smije biti veće od 10 sekundi
* Korisniku je ograničen broj znakova koje može unijeti u interakciji s pojedinim segmentima aplikacije
* Eventualne korisničke greške ne smiju utjecati na rad sustava i na stanje baze podataka
* Korisnik mora biti obaviješten o eventualnom neispravnom korištenju sustava
* Podaci se od poslužitelja do klijenta moraju prenositi protokolom HTTPS
* Korisničke lozinke ne smiju biti spremljene u bazu podataka kao jednostavan tekst, već se za njihovo šifriranje mora koristiti BCrypt funkcija za računanje sažetka
* Korisnik/gost ne smije imati uvid u dijelove sustava za koje nema ovlasti
* Web-aplikacija mora biti napisana koristeći se objektno orijentiranim jezikom
* Web-aplikacija i pripadna dokumentacija moraju biti u cijelosti dovršene do siječnja 2019. godine

# Arhitektura i dizajn sustava

## Svrha, opći prioriteti i skica sustava

Razvoj programskog produkta se sastoji od četiri generičke aktivnosti: specifikacija, razvoj i oblikovanje, validacija te evaluacija. Fazi razvoja i oblikovanja kao prva, odnosno, početna aktivnost pripada izbor i oblikovanje arhitekture. Arhitektura sustava podrazumijeva osnovna pravila stvaranja i strukturiranja cijelog sistema te je ona, kada je jednom osmišljena i provedena, nešto što se ne može tek tako mijenjati pa je zbog toga prije same implementacije potrebno arhitekturi pridati mnogo pažnje i vremena.

Pravilan odabir arhitekture ima niz prednosti, od kojih su samo neke niža cijena razvoja i održavanja programskog produkta, ponovna uporaba, veća razumljivost te uočavanje i analiza ranih pogrešaka u oblikovanju. Iz navedenih razloga je vidljivo da je odabir arhitekture ključan za razvoj programskog produkta pa je u svrhu izrade web aplikacije čija će implementacija biti opisana u kasnijim poglavljima odabrana objektno orijentirana arhitektura.

Osnovni elementi koje svaka objektno orijentirana arhitektura uključuje su:

• **Objekt** (*engl. Object)* – osnovna jedinica, primjerak razreda. U objektno orijentiranoj paradigmi teži se uporabi članskih atributa. Isto tako metode pripadaju nekom konkretnom razredu.

• **Učahurivanje** (*engl. Encapsulation*) – ili enkapsulacija. Pošto ne postoje globalne varijable, pristup varijablama razreda je moguć jedino preko neke metode iz te klase. Time je onemogućeno nepredviđeno mijenjanja parametara.

• **Nasljeđivanje** (*engl. Inheritance*) – razredi mogu naslijediti svojstva svojih roditelja te ih nadopuniti svojstvima specifičnim samo za taj razred.

• **Višeobličje** (*engl. Polymorphism*) – mogućnost definiranje metoda istog imena, a različitih parametara.

Objektno orijentirana paradigma općenito je način modeliranja u kojem se svijet promatra kroz objekte. Takav način oblikovanja je već ustaljen u svijetu programiranja, modularan, razumljiv i kod je oblikovan u smislene cjeline koje nazivamo razredima.

Ovaj projekt će prilikom oblikovanja pratiti MVC (“Model-View-Controller”, odnosno model – pogled – nadglednik) obrazac, a sam sustav se bazira na arhitekturi klijent – poslužitelj.

Razvoj web aplikacije bit će podijeljen u tri mehanizma:

1. **Model** – objektni model podataka. Služi skladištenju struktura podataka korisničkog sučelja.
2. **View** – prikaz grafičkog korisničkog sučelja.
3. **Controller** – logika interakcije / promjene.



Slika 12: MVC obrazac

Arhitektura sustava ovog projekta sastoji se od:

* Klijentska aplikacija
* Administratorska aplikacija
* Središnji poslužitelj
* Baza podataka

**Klijentska aplikacija**

Klijentska aplikacija je namijenjena krajnjem korisniku (u vidu ovog projekta to je stanovnik četvrti). Ona nudi grafičko korisničko sučelje koje je razumljivo i intuitivno za korištenje.

**Administratorska aplikacija**

Administrator sustava upravlja računima korisnika, otvara povlaštene račune, definira četvrti i područja koja ona obuhvaćaju . Administrator je neovisan od stanovnika četvrti i ne preuzima njegove uloge te on navedenim ovlastima pristupa preko administratorske aplikacije koja je neovisna o klijentskoj aplikaciji.

**Središnji poslužitelj**

Središnji poslužitelj vrši obradu zahtjeva klijentske i administratorske aplikacije. Svaki zadatak koji uključuje slanje i primanje podataka komunicira sa središnjim poslužiteljem čiji je zadatak obrada i slanje povratnih informacija klijentu. Središnji poslužitelj pri obavljanju rada komunicira s bazom podataka.

**Baza podataka**

Baza podataka je važan dio sustava jer su u njoj pohranjeni svi podaci koje koristi web aplikacija. Korištena baza podataka je relacijskog tipa. Takav model organizira podatke u jednu ili više tablica (relacija) između kojih su definirane veze.



Slika 13: Skica baze podataka

Baza ovog sustava sadrži tablice s navedenim atributima:

* **Korisnik** (*user*)
* šifra korisnika
* korisničko ime
* sažetak lozinke
* datum kreiranja računa
* **Administrator** (*administrator*)
* **Stanovnik** (*resident*)
* ime stanovnika
* prezime stanovnika
* email adresa stanovnika
* šifra ulice u kojoj stanovnik živi
* **Moderator** (*moderator*)
* **Vijećnik** (*councilor*)
* **Objava** (*forum\_post*)
* šifra objave
* sadržaj objave
* vrijeme objave
* šifra autora objave
* šifra teme kojoj objava pripada
* **Tema** (*forum\_thread*)
* šifra teme
* naslov teme
* vrijeme otvaranja teme
* autor teme
* šifra četvrti za koju je vezana tema
* **Četvrt** (*neighborhood*)
* šifra četvrti
* ime četvrti
* **Događaj** (*event*)
* šifra događaja
* ime događaja
* lokacija događaja
* vrijeme početka događaja
* vrijeme kraja događaja
* **Predložen događaj** (*suggested\_event*)
* šifra autora prijedloga događaja
* vrijeme predlaganja događaja
* **Prihvaćen događaj** (*accepted\_event*)
* vrijeme prihvaćanja događaja
* **Ulica** (*street*)
* šifra ulice
* ime ulice
* šifra četvrti u kojoj je ulica

## Dijagram razreda s opisom



Slika 14: Dijagram razreda

Razred **User** predstavlja registriranog korisnika. Sadrži korisničko ime, sažetak lozinke i vrijeme otvaranja računa.

**Administrator** je registrirani korisnik s posebnim ovlastima.

**Resident** je uobičajeni registrirani korisnik koji stanuje u nekoj četvrti. Sadrži puno ime, e-mail adresu i ulicu u kojoj stanuje.

**Resident** može dodatno biti **Councilor** (vijećnik četvrti) ili **Moderator**.

Razred **Street** predstavlja imenovanu ulicu, a **Neighborhood** skup ulica, odnosno imenovanu četvrt.

**Neighborhood** sadrži kolekcije instanci razreda **ForumThread** (tema na forumu), **AcceptedEvent** (prihvaćeni događaj), **SuggestedEvent** (predloženi događaj) i **CouncilReport** (izvješće vijeća četvrti).

**ForumThread** sadrži naslov, trenutak otvaranja i stanovnika (**Resident**) koji ju je otvorio te se sastoji od više postova na forumu (**ForumPost**).

**ForumPost** sadrži tekst objave i vrijeme objave.

Razred **Event** predstavlja općenite informacije o nekom događaju, a sadrži naslov, lokaciju, trenutak početka i trenutak završetka (koji može biti i nepoznat).

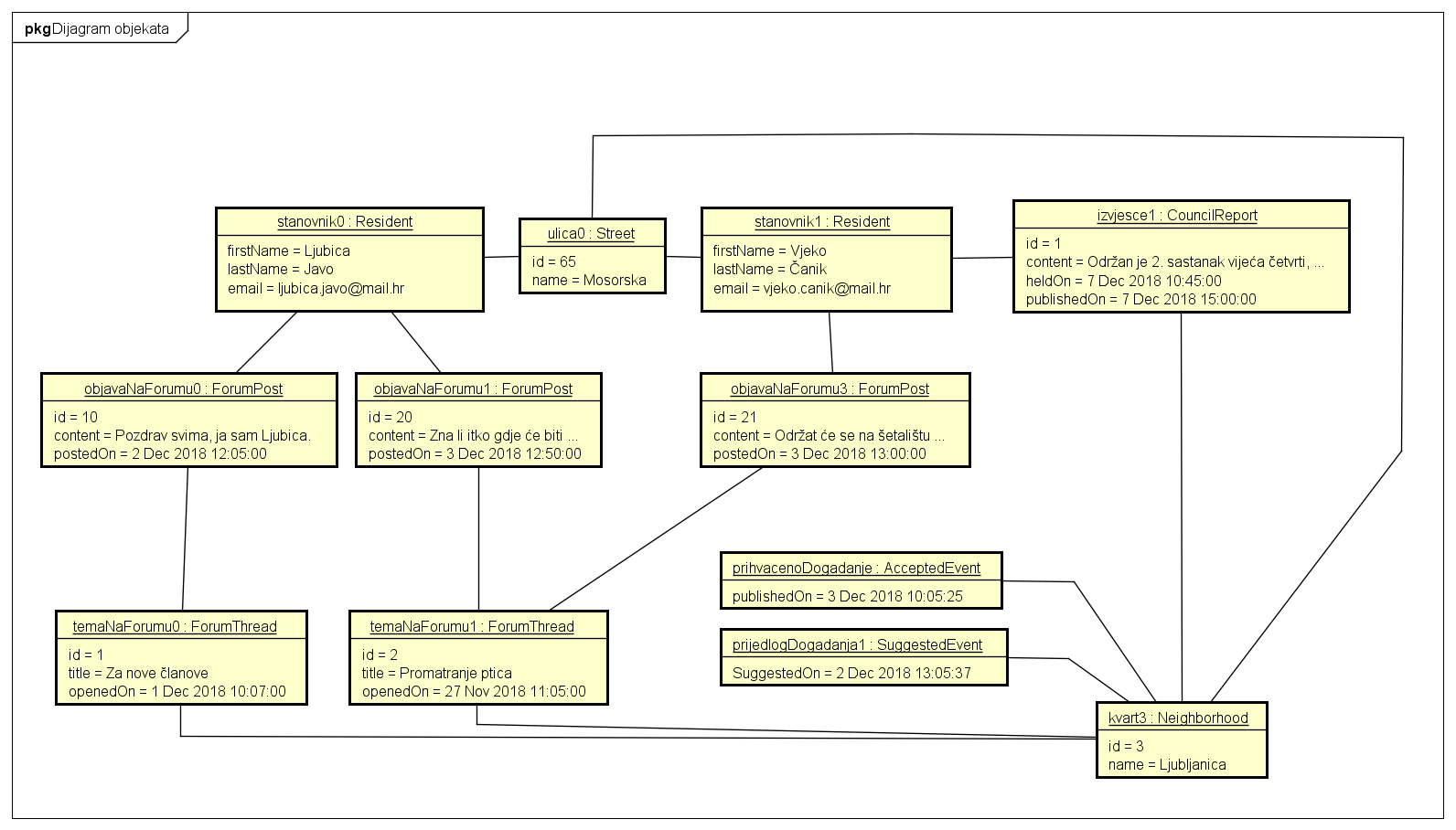
**SuggestedEvent** je **Event** kojeg je neki stanovnik predložio. Osim reference na tog stanovnika (**Resident**), sadrži i trenutak kad ga je predložio.

**AcceptedEvent** je **Event** koji je prihvaćen i objavljen. Sadrži trenutak objavljivanja.

**CouncilReport** predstavlja izvješće s vijeća četvrti. Sadrži tekst izvješća, vrijeme održavanja, vrijeme objavljivanja i referencu na vijećnika (**Councilor**) koji ga je objavio.

## Dijagram objekata

Sljedeći dijagram objekata prikazuje djelomično stanje sustava nastalo registracijom i radnjama dvaju korisnika - redovnog stanovnika i vijećnika. Oba korisnika su ostavili objave na forumu. Dodatno, vijećnik je objavio izvješće vijeća četvrti, a redovni stanovnik predložio dva događanja od kojih je jedno prihvaćeno.



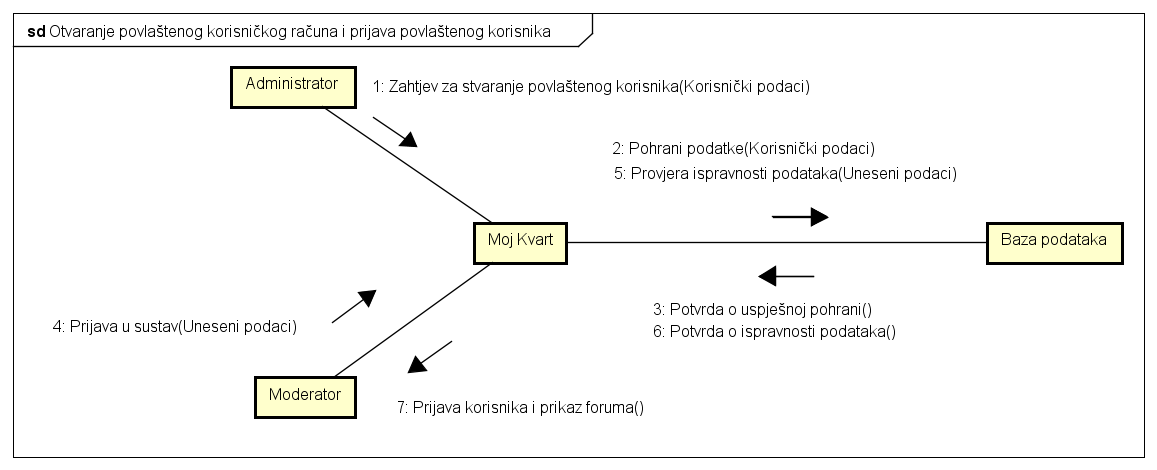
Slika 15: Dijagram objekata

## Ostali UML dijagrami

*Ovdje počinju sadržaji Revizije 2.*

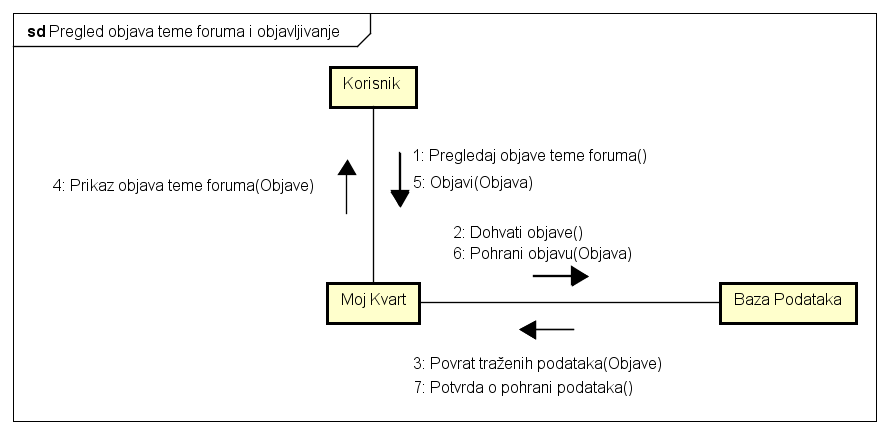
*U ovom potpoglavlju potrebno je uvrstiti dodatna četiri dijagrama koji prikazuju arhitekturu sustava i to: komunikacijski dijagram, dijagram stanja, dijagam aktivnosti i dijagram komponenti. Potrebno je dati barem po jedan primjerak svakog od dijagrama. Potrebno je da svaki od dijagrama prikazuje (jedan) bitan dio funkcionalnosti sustava. Dijagram komponenti treba prikazivati sve komponente sustava. Prema potrebi, ovdje se mogu dodati i neki dodatni sekvencijski dijagrami za objekte.*

**Dijagrami komunikacije**

****

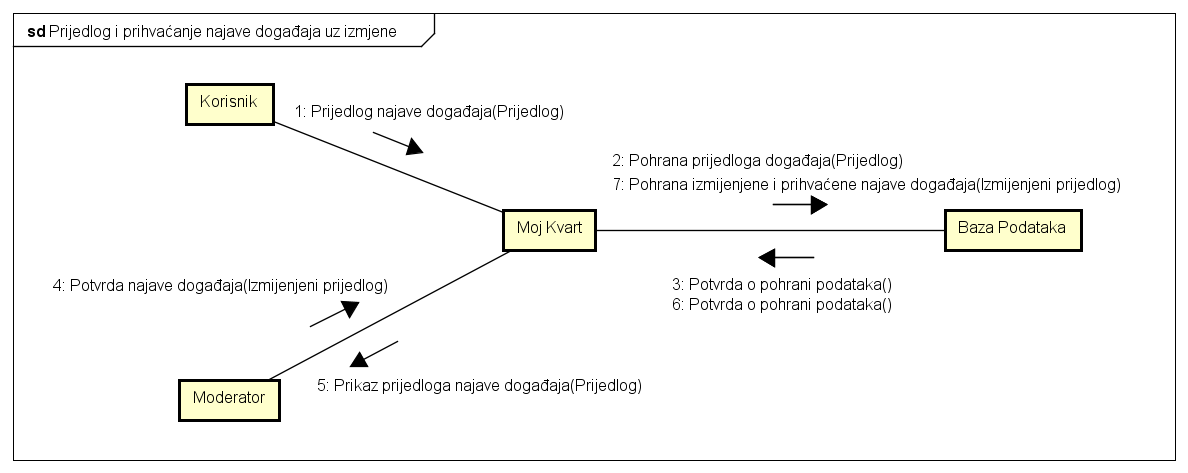
Slika 16: Dijagram komunikacije otvaranja korisničkog povlaštenog računa i prijave na isti

Slika 16 prikazuje dijagram komunikacije za slučaj gdje je potrebno stvoriti povlašteni korisnički račun na kojeg se potom vlasnik prijavljuje. Takav račun može otvoriti samo administrator. Administrator unosi podatke željenog korisničkog računa (ovdje s dostupnom e-mail adresom) te aplikacija pohranjuje te podatke u bazu podataka.Vlasnik računa se prijavljuje na sustav unoseći ispravne podatke čiju ispravnost aplikacija provjerava u bazi podataka te korisnika prijavljuje na sustav.

****

Slika 17: Dijagram komunikacije pregledavanja objava i objavljivanja na forumu

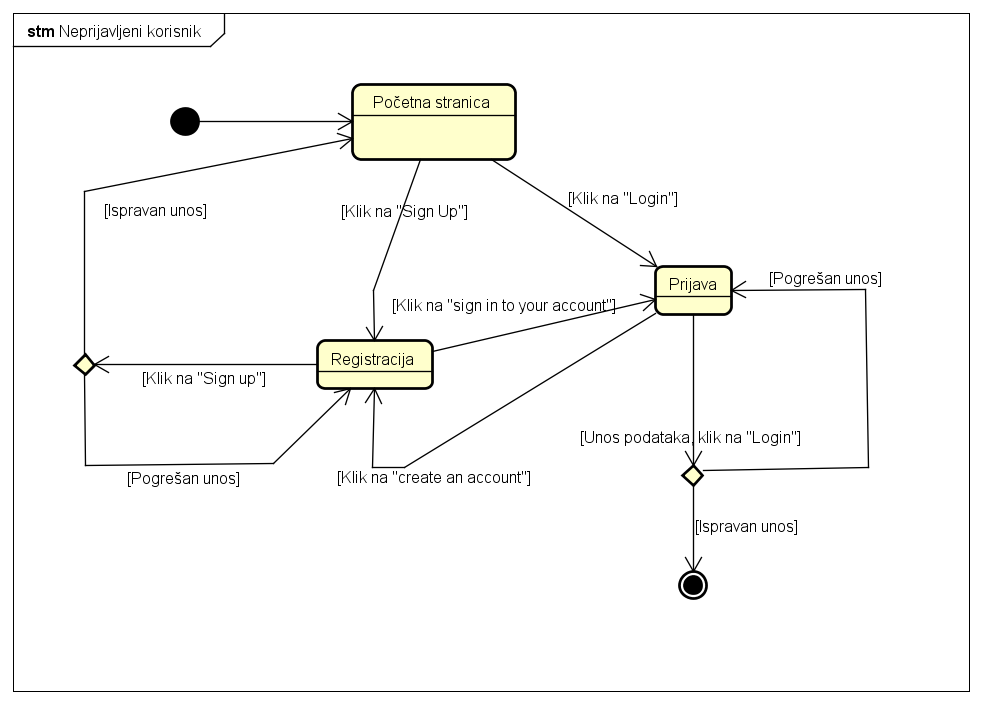
Slika 17 prikazuje dijagram komunikacije korisničkog pregledavanja foruma i objavljivanje. Korisnik otvara temu foruma kojeg aplikacija dohvaća iz baze podataka. Zatim unosi svoju objavu koju aplikacija pohranjuje u bazu podataka.

****

Slika 18: Dijagram komunikacije predlaganja i prihvaćanja događaja uz izmjene

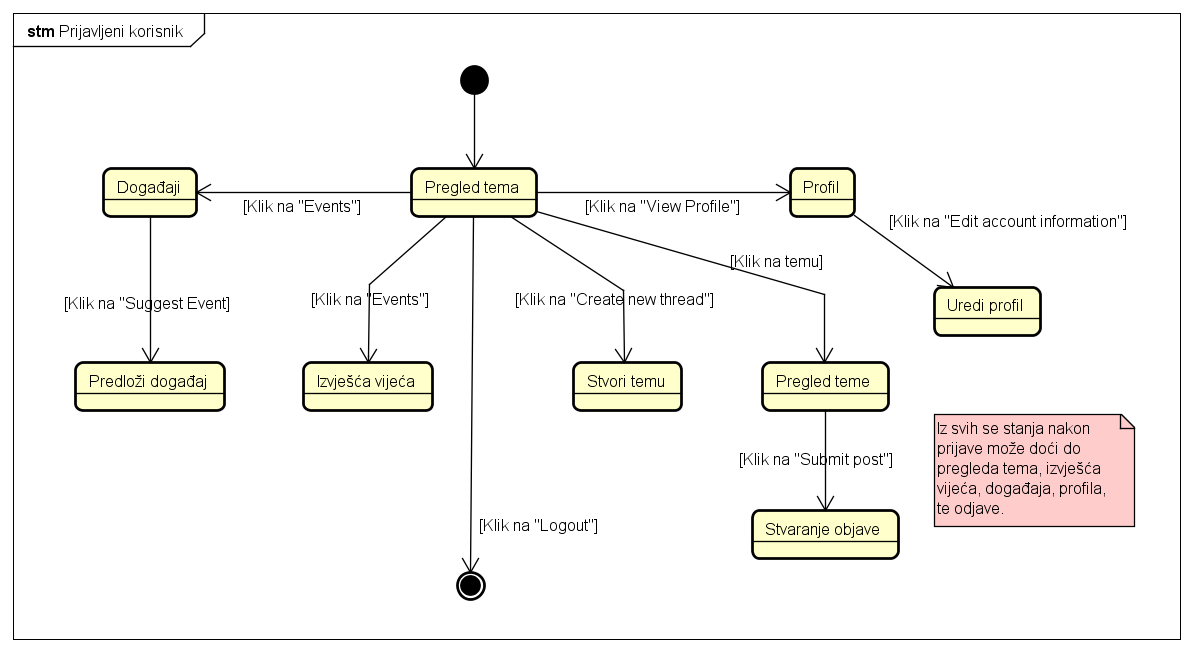
Slika 18 prikazuje dijagram komunikacije predlaganja najave događaja objavljenog uz izmjene. Korisnik unosi prijedlog događaja kojeg aplikacija pohranjuje u bazu podataka. Nakon pohrane, aplikacija prikazuje prijedlog moderatoru koji nakon pregledavanja uz unos željenih izmjena objavljuje prijedlog događaja - aplikacija ga unosi u bazu podataka.

**Dijagram stanja**

****

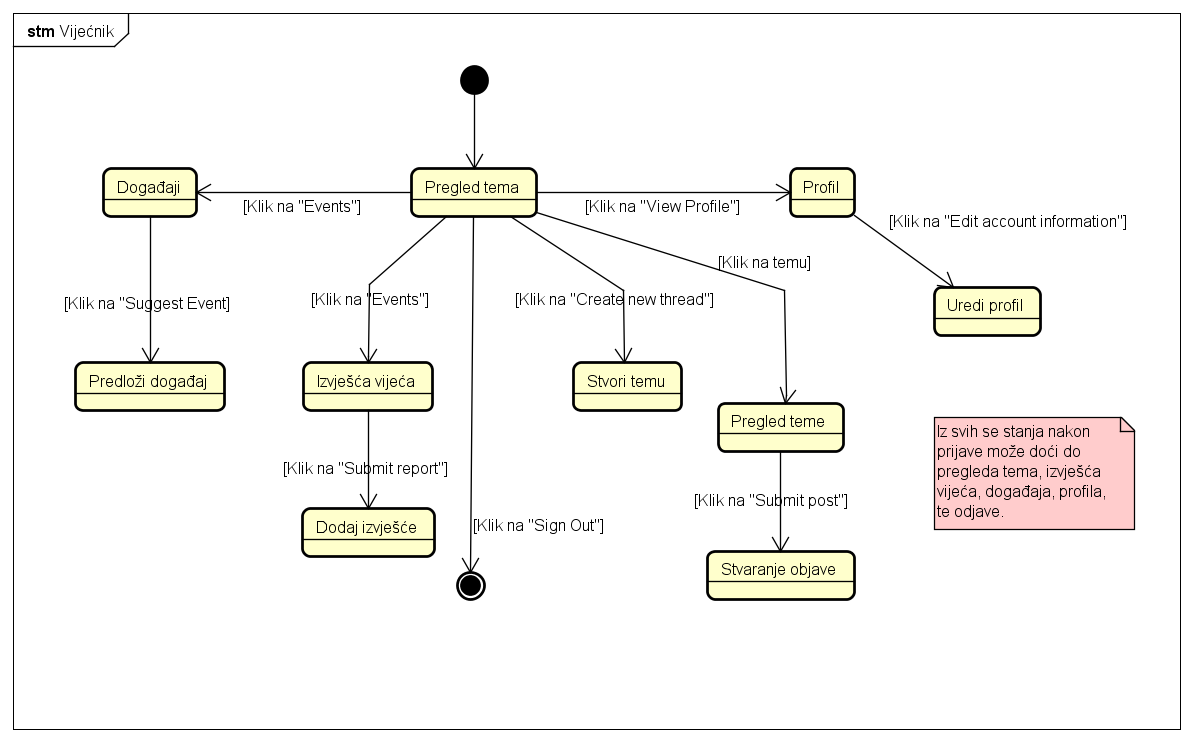
Slika 15: dijagram stanja za neprijavljenog korisnika

Neprijavljeni korisnik dolazi na početnu stranicu. Ako ima račun, može se prijaviti. Ako nema, može se registrirati isključivo kao stanovnik. Registracija i prijava se prihvaćaju tek kada je unos ispravan.



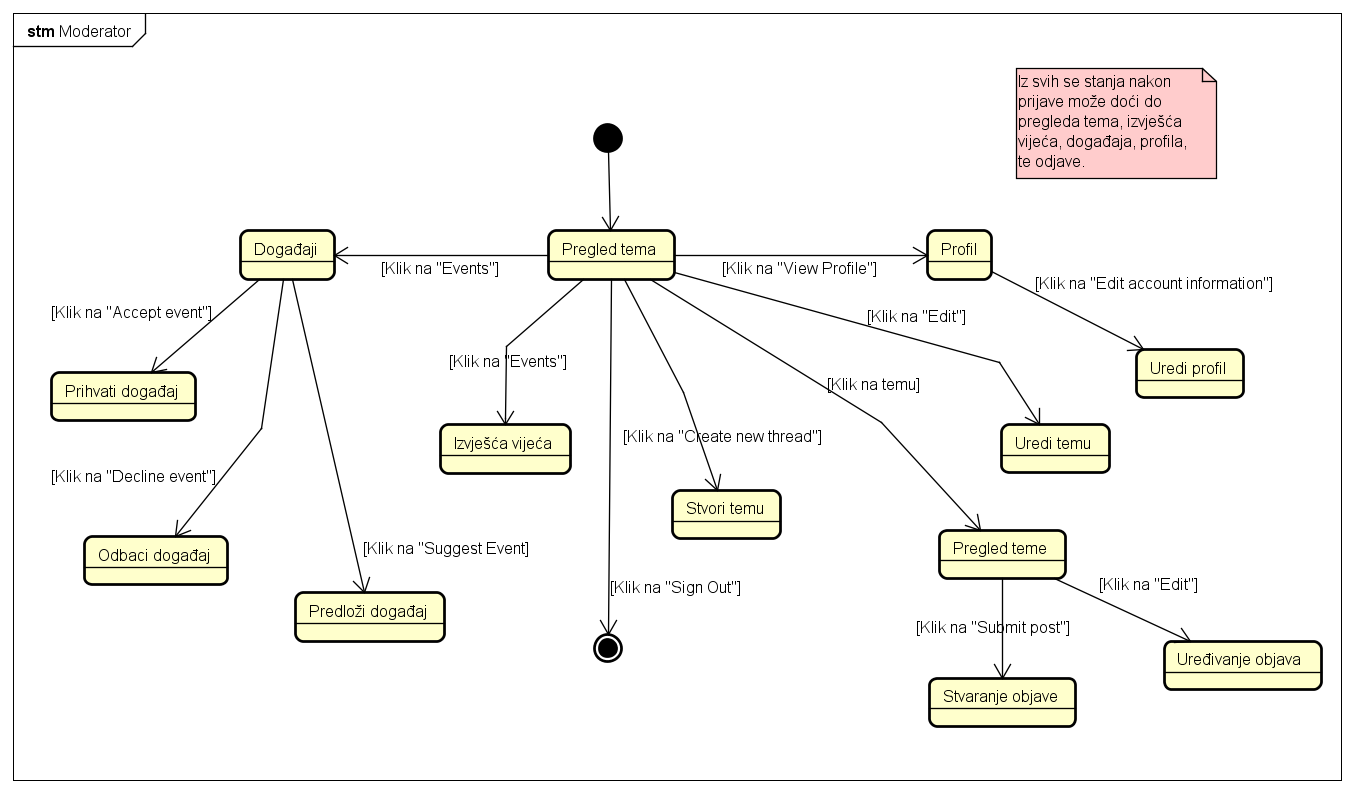
Slika 16:dijagram stanja za prijavljenog korisnika

Kada se korisnik prijavi, prikaže se pregled tema. Korisnik može stvarati teme na stranici „Threads“, predlagati događaje na stranici „Events“, pregledavati izvješća vijeća na stranici „Council“, pregledati svoj profil na stranici „Profile“ te se odjaviti sa „Logout“. Također može dodavati objave u temu, jednom kad ju otvori sa stranice „Threads“.



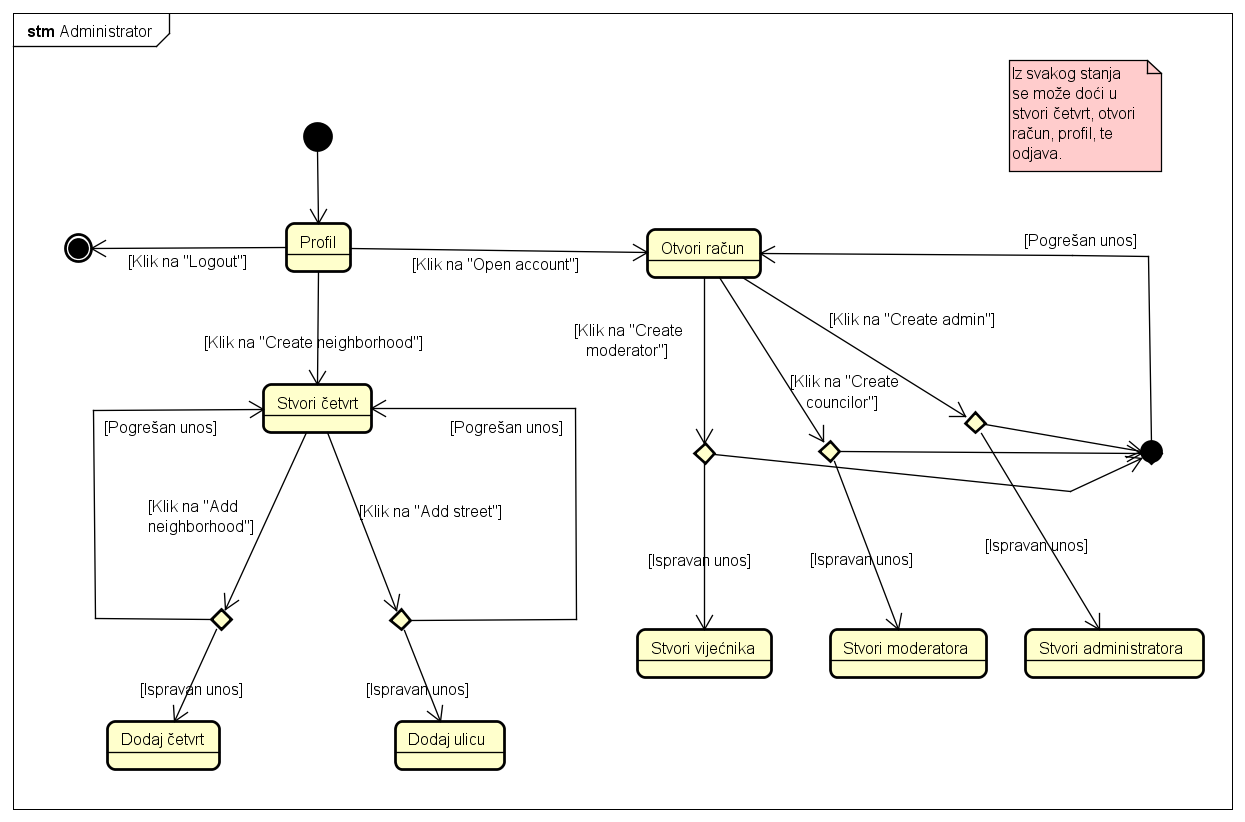
Slika 17: dijagram stanja za vijećnika

Vijećnik može sve što može i korisnik, i dodati izvješća vijeća na stranici „Events“.



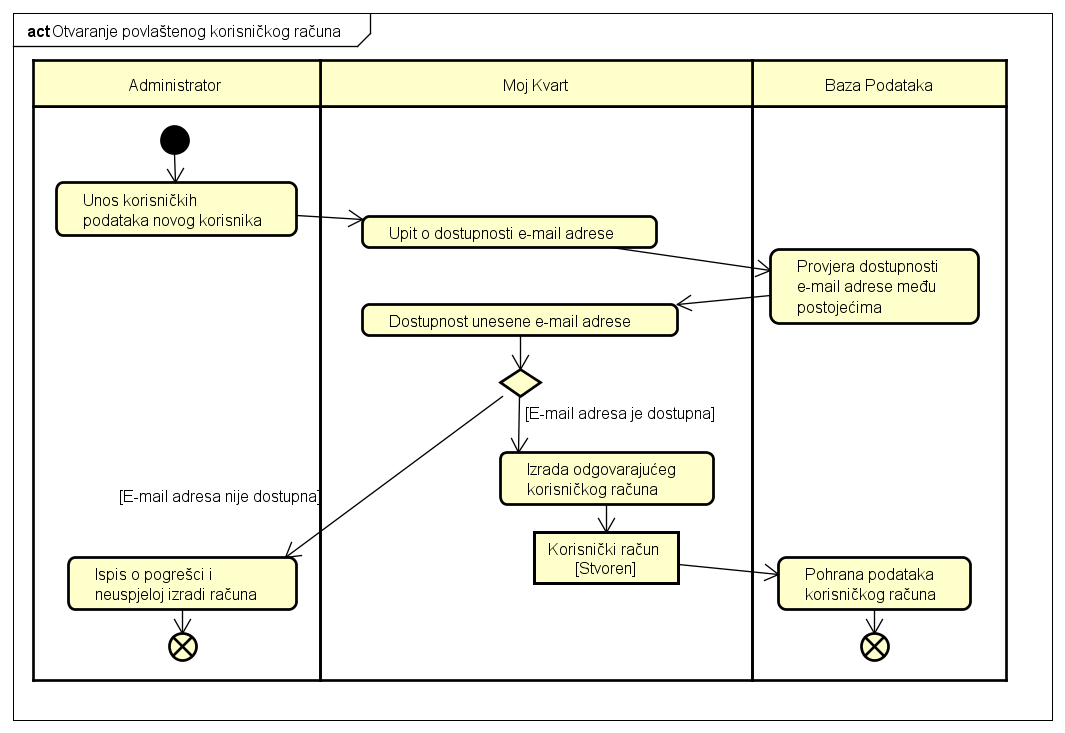
Slika 18: dijagram stanja za moderatora

Moderator može sve što može i korisnik, ali i uređivati teme na stranici „Threads“ te prihvaćati i odbacivati predložene događaje na stranici „Događaji“. Također može uređivati objave u temama.

Administrator nakon prijave dolazi na stranicu svog profila. Može otvarati račune za vijećnike, moderatore i administratore. Također može stvarati četvrti i ulice.

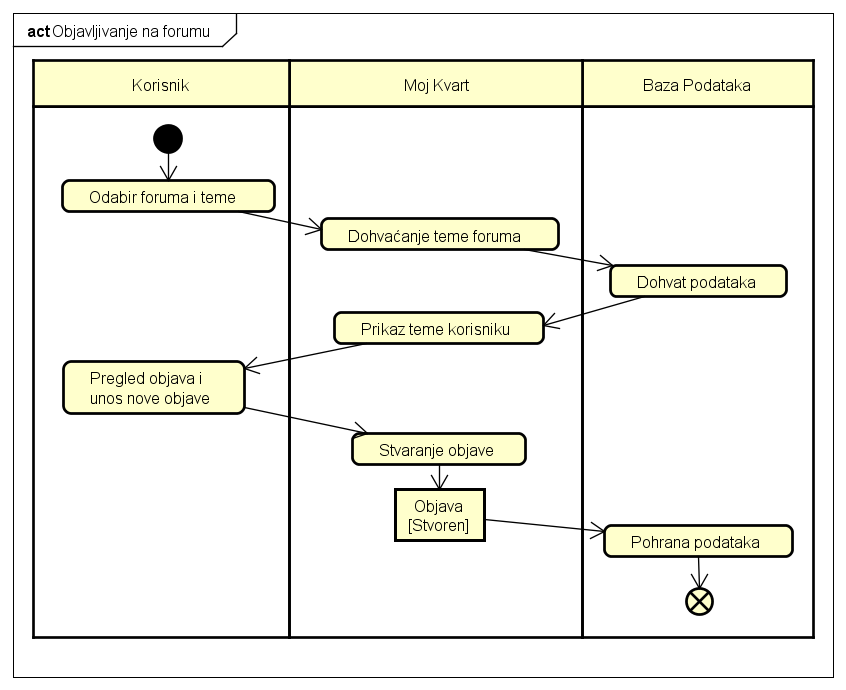
Slika 19: dijagram stanja za administratora

**Dijagrami aktivnosti**



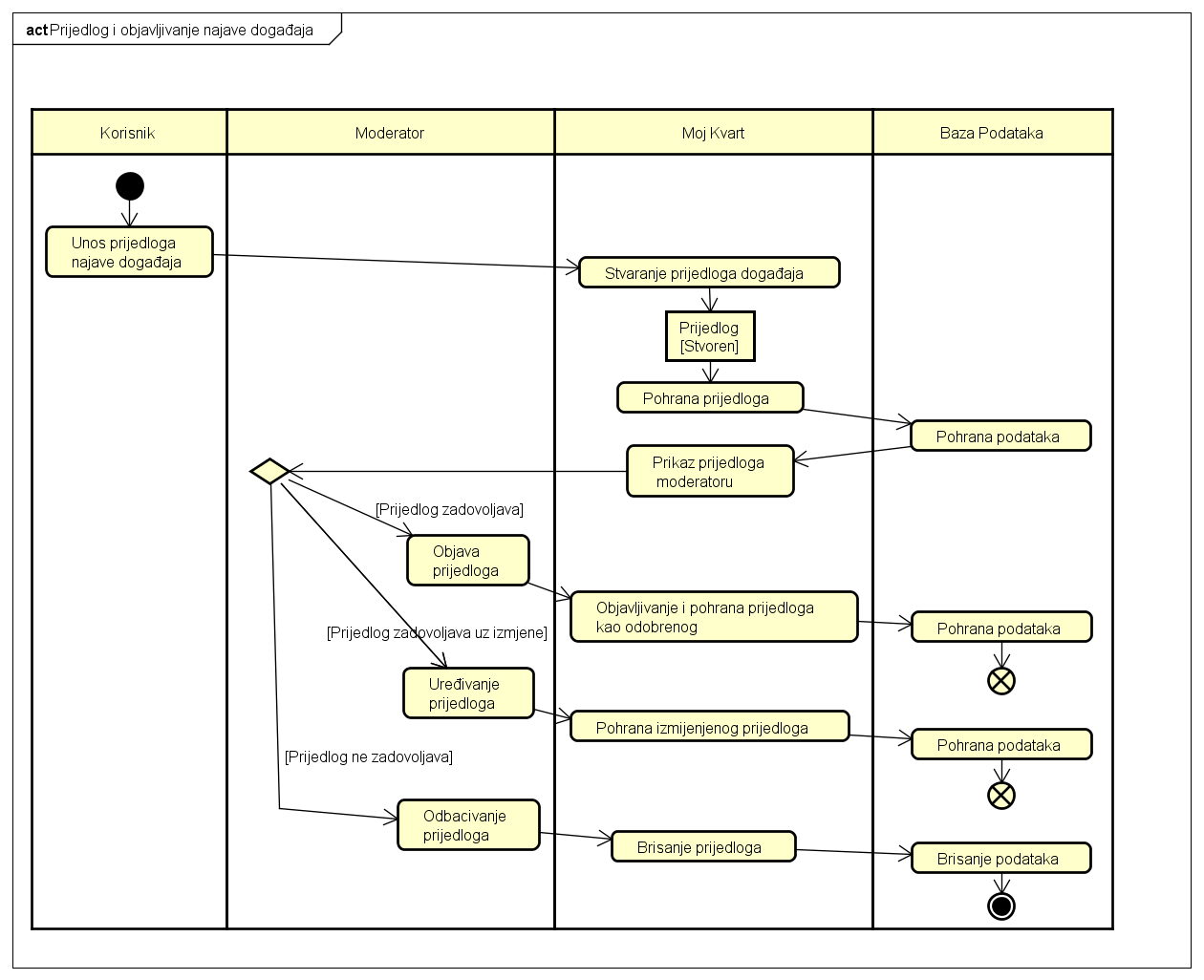
Slika 19: Dijagram aktivnosti otvaranja povlaštenog korisničkog računa

Slika 19 prikazuje dijagram aktivnosti za otvaranje povlaštenih korisničkih računa. Povlaštene korisničke račune može otvarati samo administrator te on upravo i u ovom slučaju započinje aktivnost. On unosi korisničke podatke za novi račun te aplikacija provjerava dostupnost unesene e-mail adrese pomoću baze podataka. Ako je e-mail adresa u uporabi - administratoru se ispisuje pogreška te novi račun nije izrađen. U drugom slučaju, ako je e-mail adresa dostupna, aplikacija izrađuje korisnički račun na temelju unesenih podataka te ga pohranjuje u bazu podataka.



Slika 20: Dijagram aktivnosti objavljivanja na forumu

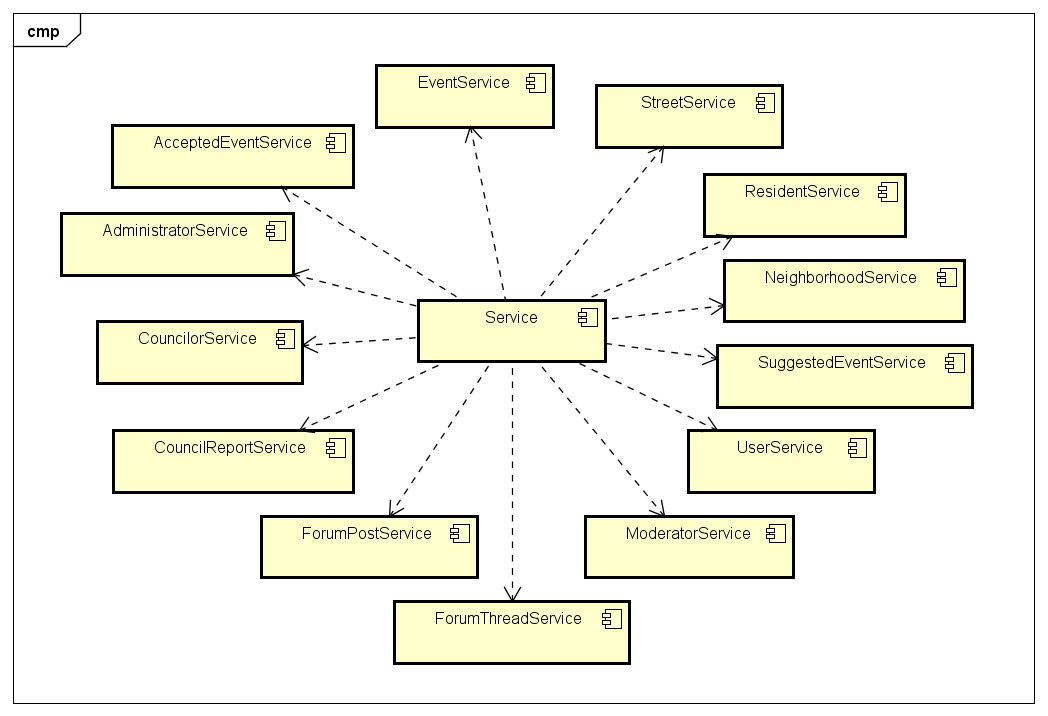
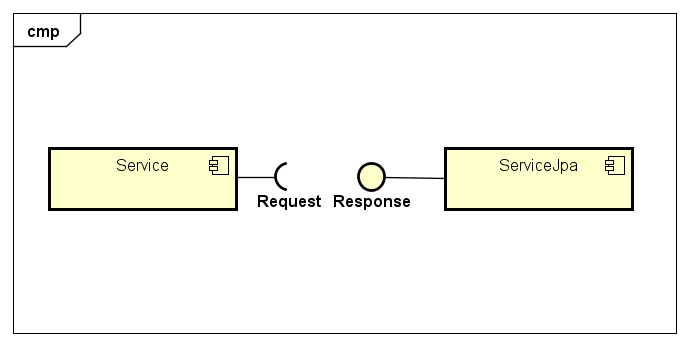
Slika 20 prikazuje dijagram aktivnosti za objavljivanje na forumu. Radnju započinje korisnik koji odabire temu na forumu. Aplikacija dohvaća temu iz baze podataka te prikazuje objave korisniku. Korisnik unosi objavu te ju aplikacija obradi i pohrani u bazu podataka.



Slika 21: Dijagram aktivnosti predlaganja i objavljivanja najave događaja

Slika 21 prikazuje dijagram aktivnosti predlaganja i objavljivanja najave događaja od strane korisnika. Proces započinje korisnik unosom prijedloga najave događaja. Na temelju korisnikova unosa aplikacija izrađuje prijedlog događaja te ga pohranjuje u bazu podataka. Aplikacija nakon pohrane prikazuje prijedlog događanja moderatoru na daljnje aktivnosti. Moderator, s obzirom na zadovoljstvo sadržajem prijedloga, može objaviti prijedlog kakav jest, izmijeniti ga ili u potpunosti odbaciti. Na posljetku aplikacija, na temelju moderatorove odluke, unosi odgovarajuće promjene u bazu podataka.

**Dijagram komponenti:**



# Implementacija i korisničko sučelje

*U ovom poglavlju potrebno je:*

*dati dijagram razmještaja (deployment dijagram)*

*navesti koje su tehnologije i alati korišteni u razvoju sustava*

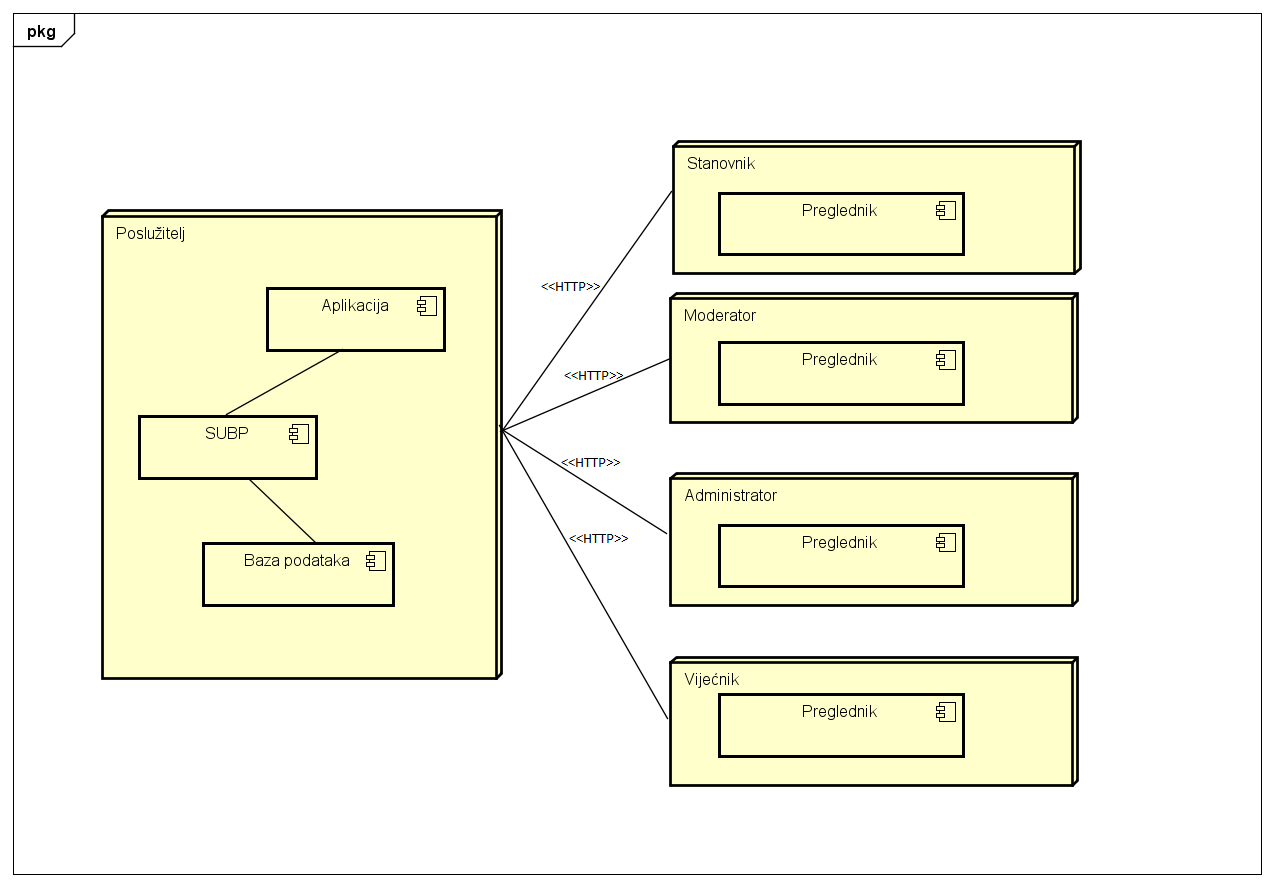
*dati isječak programskog koda koji implementira neku od temeljnih funkcionalnosti u sustavu*

*objasniti kako je ispitano implementirano rješenje i pokazati bar 4 ispitna scenarija*

*dati upute za instalaciju*

*dati upute za korištenje*

## Dijagram razmještaja



Slika 19: dijagram razmještaja

Sustav se sastoji od poslužitelja i klijenata, koji mogu biti stanovnici, moderatori, administratori i vijećnici. Klijenti komuniciraju s poslužiteljem putem HTTP-a, koristeći svoje web preglednike. Aplikacija na poslužitelju komunicira s bazom podataka preko sustava za upravljanje bazom podataka.

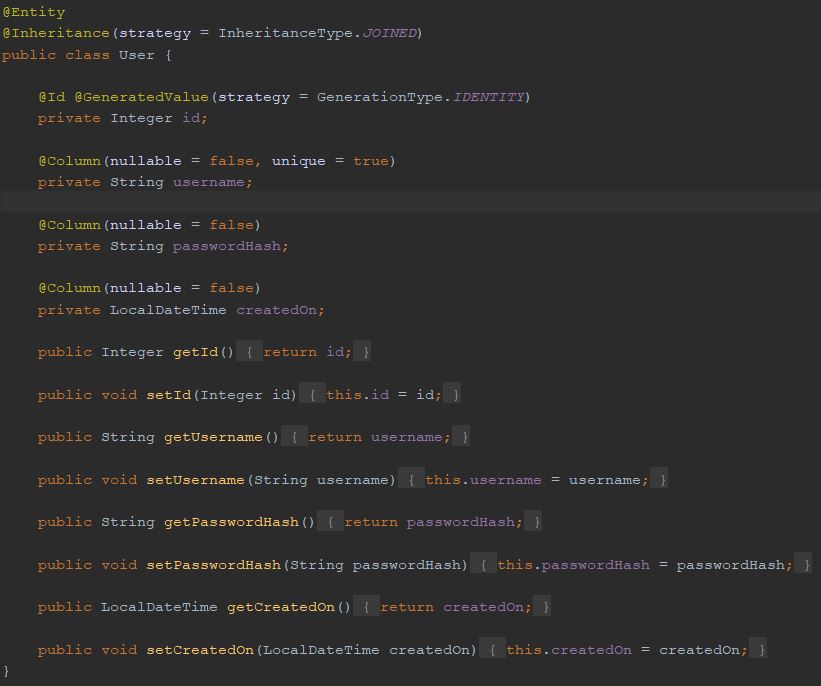
## Korištene tehnologije i alati

*Navesti sve tehnologije i alate koji su primijenjeni pri izradi projekta te ih ukratko opisati; njihovo značenje i mjesto i način primjene.*

## Isječak programskog koda vezan za temeljnu funkcionalnost sustava

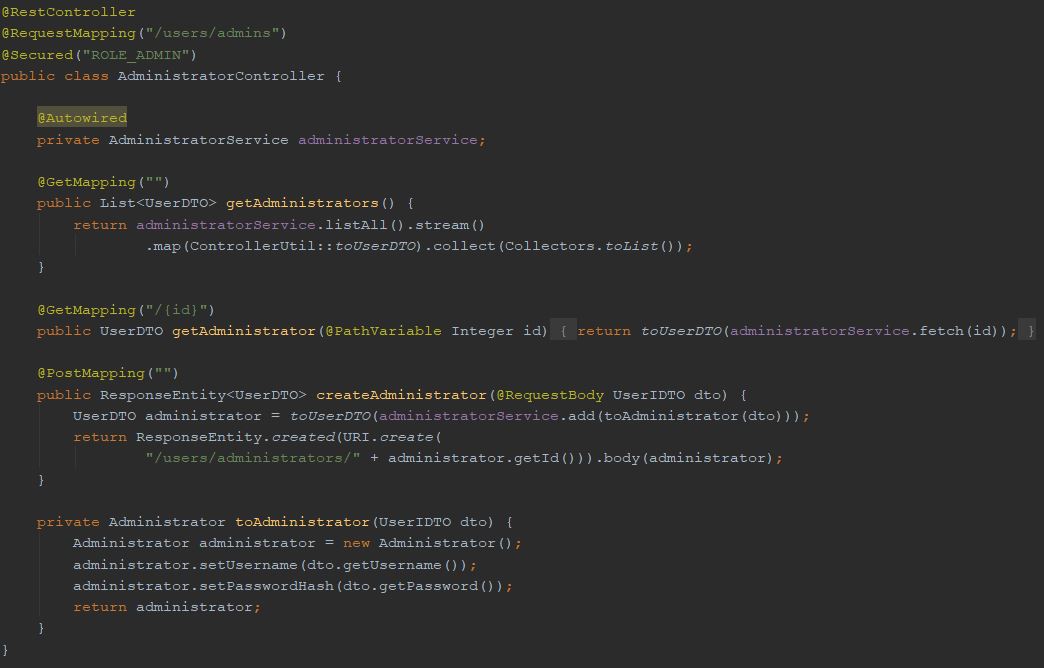
Pozadina foruma *Moj Kvart* pisana je u *Spring* radnom okviru. Spring radni okvir radi na MVC obrascu i ovo su isječci koda koji implementiraju temeljne funkcionalnosti:

Tablice u bazi podataka označene su @Entity anotacijama. Svi stupci su reprezentirani varijablama željenog tipa, a ograničenja se mogu specificirati parametrom u @Column anotaciji. Uz varijable, u razredima su napisane i pomoćne metode koje određuju na koji način se pristupa podatcima u bazi podataka. *Spring* radni okvir omogućuje korisniku brz i jednostavan odabir korištene baze podataka, u ovom slučaju H2.



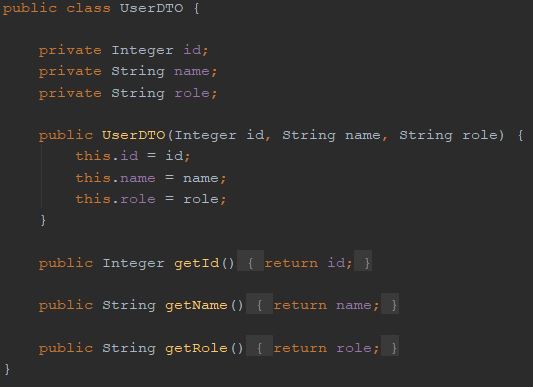
Slika 22: Primjer tablice u bazi podataka

Anotacija @RestController označava da je razred *Controller*. Controlleri upravljaju korisničkim zahtjevima i pomažu ostatku sustava obrađivati i mapirati REST zahtjeve na prave lokacije.



Slika 23: Primjer Controllera

Sufiks „DTO“ označava da je razred „Data transfer object“ tj. da prenosi podatke između procesa. DTO-i se često koriste jer uvelike olakšavaju komunikaciju između različitih dijelova sustava (u ovom slučaju front-enda i back-enda) jer na uniforman način dostavljaju informacije između njih.



Slika 24: Primjer DTO-a

## Ispitivanje programskog rješenja

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti s prikazom odabira ispitnih slučajeva.*

*Prilikom prezentacije svojih Ispitnih scenarija (minimalno četiri) studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete. Poželjno je da se napravi i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane te da se vidi na koji način sustav reagira kada nešto nije u potpunosti ostvareno.*

*Različiti ulazi za ispitne scenarije trebaju pokrivati temeljnu funkcionalnost nekog modula i nekoliko rubnih uvjeta.*

## Upute za instalaciju

**Instalacija JDK:**

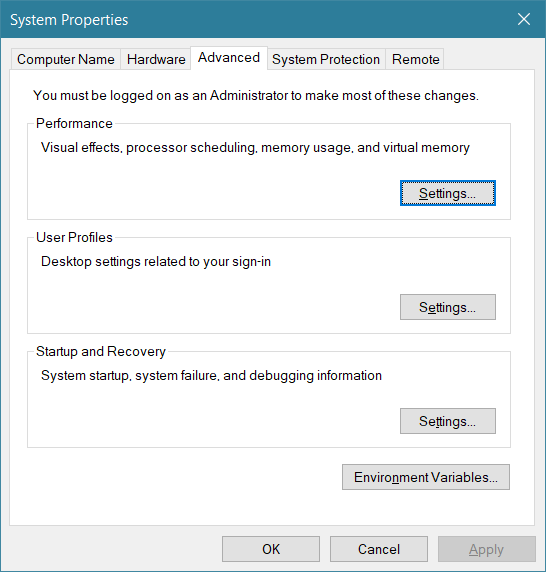
Sa web-lokacije <https://www.oracle.com/technetwork/java/javase/downloads/index.html> potrebno je skinuti odgovarajuću verziju Java Developers kit-a (ovisno o računalu 32 ili 64-bitnu verziju) i instalirati je.

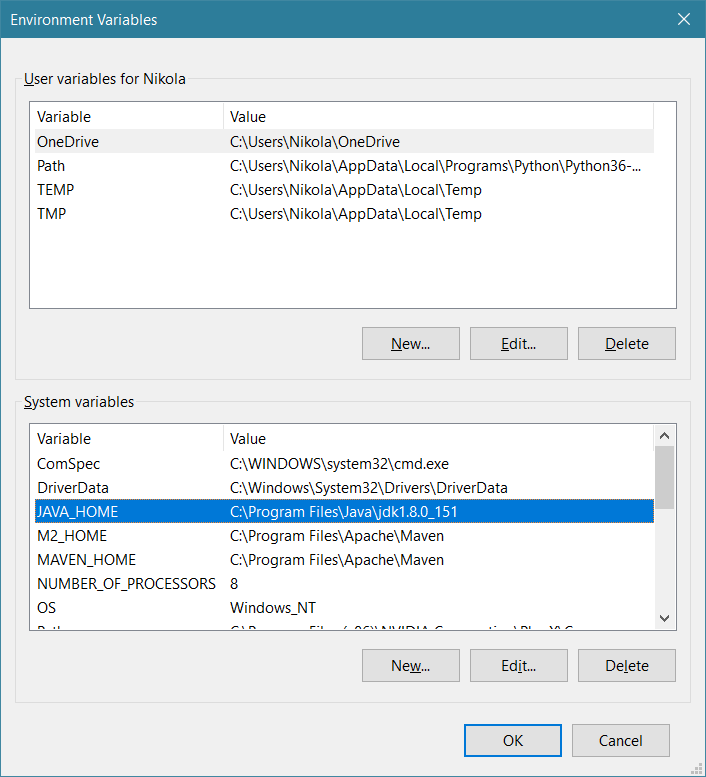


Da bi se dovršila instalacija, potrebno je otići u izbornik *System Properties* i u

podizborniku *Environment Variables* dodati varijablu JAVA\_HOME koja pokazuje na

direktorij u kojem je instalirana Java.





Također, u varijablu PATH treba dodati „%JAVA\_HOME%\bin“ i „%JAVA\_HOME%\jre\bin“.

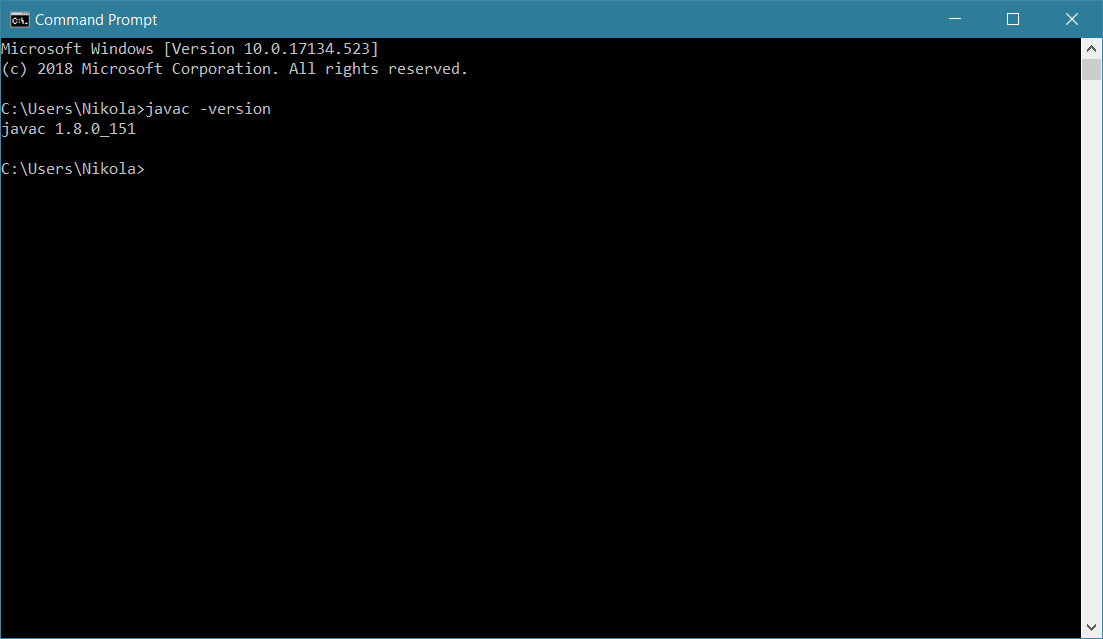
Varijabla okruženja PATH operacijskom sustavu govori gdje da traži programe koje

korisnik želi pokrenuti iz naredbenog retka

Za provjeru ispravnosti instalacije, dovoljno je otvoriti *Command prompt* (izbornik

*Start* -> cmd) i upisati naredbu „javac -version“.

Ispis bi trebao biti sličan ovome:

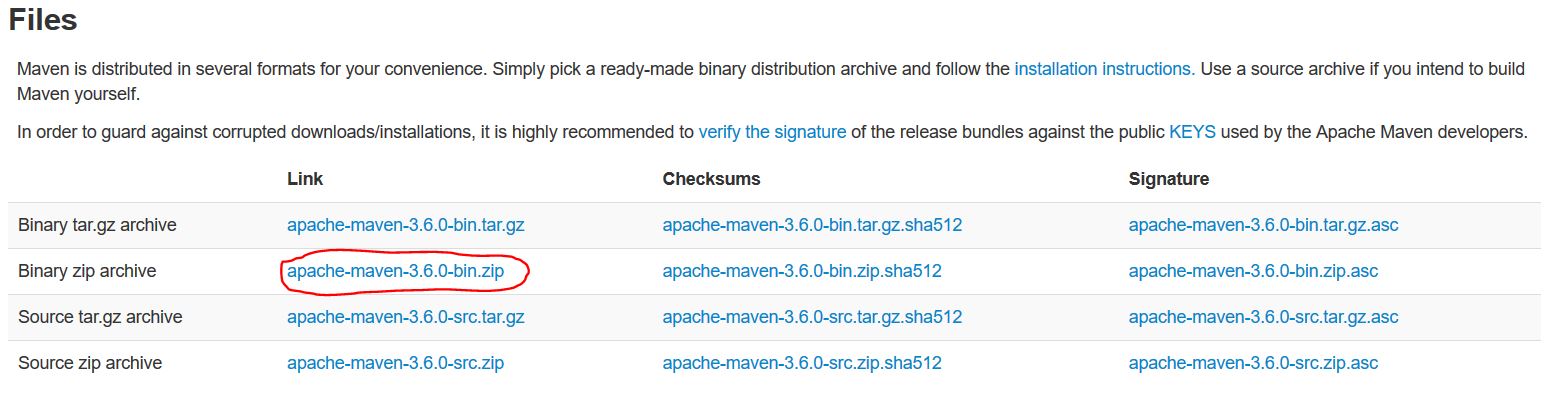


**Instalacija Maven-a:**

Za instalaciju alata za upravljanje paketima *Maven*, dovoljno je skinuti instalacijski

paket s web-lokacije <https://maven.apache.org/download.cgi>, raspakirati ga pomoću

alata *WinZip* ili *WinRAR* i instalirati ga duplim klikom na .bin datoteku.



Pri instalaciji, korisno je provjeriti je li varijabla okruženja %JAVA\_HOME% ispravno

podešena.

Instalacija se može provjeriti naredbom „mvn -v“ u ljusci operacijskog sustava.

Jednom kad je instaliran, *Maven* će se pobrinuti za instalaciju preostalih paketa

potrebnih za rad foruma *Moj Kvart .*

## Korisničke upute

*Korisničke upute ovisit će o količini implementirane funkcionalnosti. Očekuje se da upute budu na oko pet A4 stranica koje će dati potpuni opis funkcionalnosti sustava sa stajališta krajnjeg korisnika.*

# Zaključak i budući rad

*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.*

# Popis literature

*Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.*

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
3. I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
4. T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
5. Software engineering ,Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
6. I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
7. Concepts: Requirements, <http://www.upedu.org/upedu/process/gcncpt/co_req.htm>
8. UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
9. Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
10. Astah Community, <http://astah.net/editions/community/>

# Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

**Slike:**

[Slika 1: Vršni dijagram obrazaca uporabe 14](#_Toc535338181)

[Slika 2: Dijagram obrazaca uporabe - Rad na forumu 15](#_Toc535338182)

[Slika 3: Dijagram obrazaca uporabe - cjelina "Događanja" 16](#_Toc535338183)

[Slika 4: Dijagram obrazaca uporabe - cjelina "Vijeće četvrti" 16](#_Toc535338184)

[Slika 5: Dijagram obrazaca uporabe - Rad sa četvrtima 17](#_Toc535338185)

[Slika 6: UC2: Otvaranje povlaštenih korisničkih računa 18](#_Toc535338186)

[Slika 7: UC5 i UC8: Objavljivanje na forumu i Pregled objava na forumu 19](#_Toc535338187)

[Slika 8: UC9: Stvaranje teme na forumu 20](#_Toc535338188)

[Slika 9: UC11: Uklanjanje objava s foruma 21](#_Toc535338189)

[Slika 10: UC12 i UC13: Prijedlog najave događaja i Objavljivanje u cjelini "Događanja" 23](#_Toc535338190)

[Slika 11: UC18 i UC19: Uređivanje četvrti i Brisanje četvrti 24](#_Toc535338191)

[Slika 12: MVC obrazac 28](file:///C:\Users\Nikola\Desktop\FAKS\5.%20SEMESTAR\OPP\MOJ%20KVART\Dokumentacija\OPP_2018_Javoljupci_v1_2.docx#_Toc535338192)

[Slika 13: Skica baze podataka 29](#_Toc535338193)

[Slika 14: Dijagram razreda 31](file:///C:\Users\Nikola\Desktop\FAKS\5.%20SEMESTAR\OPP\MOJ%20KVART\Dokumentacija\OPP_2018_Javoljupci_v1_2.docx#_Toc535338194)

[Slika 15: Dijagram objekata 33](#_Toc535338195)

[Slika 16: Dijagram komunikacije otvaranja korisničkog povlaštenog računa i prijave na isti 34](#_Toc535338196)

[Slika 17: Dijagram komunikacije pregledavanja objava i objavljivanja na forumu 35](#_Toc535338197)

[Slika 18: Dijagram komunikacije predlaganja i prihvaćanja događaja uz izmjene 35](#_Toc535338198)

[Slika 19: Dijagram aktivnosti otvaranja povlaštenog korisničkog računa 40](#_Toc535338199)

[Slika 20: Dijagram aktivnosti objavljivanja na forumu 40](#_Toc535338200)

[Slika 21: Dijagram aktivnosti predlaganja i objavljivanja najave događaja 41](#_Toc535338201)

[Slika 22: Primjer tablice u bazi podataka 44](#_Toc535338202)

[Slika 23: Primjer Controllera 45](#_Toc535338203)

[Slika 24: Primjer DTO-a 45](#_Toc535338204)

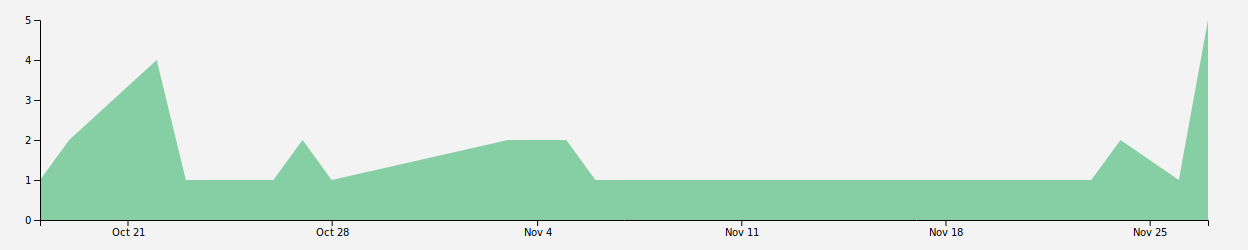
# Dodatak B: Dnevnik sastajanja

|  |  |  |  |
| --- | --- | --- | --- |
| # | Datum | Prisutni | Sadržaj |
| 01 | 19. listopada 2018. | * Mate Gašparini * Nicolas Assouline * Dana Dodigović * David Dukić * Marko Kršić * Marino Kurtović | * Podjela poslova - opis projekta i funkcionalni zahtjevi * Rasprava o odabiru tehnologija |
| 02 | 23. listopada 2018. | * Mate Gašparini * Nicolas Assouline * Dana Dodigović * David Dukić * Marko Kršić * Marino Kurtović | * Provjera točnosti funkcionalnih zahtjev * Podjela poslova - dijagrami obrazaca uporabe i sekvencijski dijagrami |
| 03 | 30. listopada 2018. | * Mate Gašparini * Marko Kršić * Marino Kurtović | * Provjera točnosti napravljenih dijagrama i eventualni ispravci * Podjela poslova - nefunkcionalni zahtjevi |
| 04 | 21. studenoga 2018. | * Mate Gašparini * Nicolas Assouline * Dana Dodigović * David Dukić * Marko Kršić * Marino Kurtović | * Općenita rasprava o nadolazećoj implementaciji * Podjela poslova - pismeni opis baze, dijagram razreda |

# Dodatak C: Prikaz aktivnosti grupe

|  |
| --- |
| **Popis aktivnosti** |
| Mate Gašparini | Nicolas Assouline | Dana Dodigović | David Dukić | Marko Kršić | Marino Kurtović |
| **Upravljanje projektom** | 100% |  |  |  |  |  |
| **Opis projektnog zadatka** |  |  | 100% |  |  |  |
| **Rječnik pojmova** |  |  |  | 100% |  |  |
| **Opis funkcionalnih zahtjeva** |  | 25% |  | 25% | 25% | 25% |
| **Opis ostalih zahtjeva** | 100% |  |  |  |  |  |
| **Arhitektura i dizajn sustava** |
| Svrha, opći prioriteti i skica sustava | 20% |  | 80% |  |  |  |
| Dijagram razreda s opisom | 50% |  | 50% |  |  |  |
| Dijagram objekata |  |  |  |  | 100% |  |
| Ostali UML dijagrami |  |  |  |  |  |  |
| **Implementacija i korisničko sučelje** |
| Dijagram razmještaja |  |  |  |  |  |  |
| Korištene tehnologije i alati |  |  |  |  |  |  |
| Isječak programskog kôda |  | 100% |  |  |  |  |
| Ispitivanje programskog rješenja |  |  |  |  |  |  |
| Upute za instalaciju |  | 100% |  |  |  |  |
| Korisničke upute |  |  |  |  |  |  |
| **Plan rada** |  |  |  |  |  |  |
| **Pregled rada i stanje ostvarenja** |  |  |  |  |  |  |
| **Zaključak i budući rad** | 33% |  |  | 33% |  | 33% |
| **Popis literature** |  |  |  |  |  |  |
| **Dodaci** |
| Indeks |  | 100% |  |  |  |  |
| Dnevnik sastajanja | 100% |  |  |  |  |  |
|  |  |  |  |  |  |  |

Pregled pohrana kroz vrijeme trajanja projekta (primjer):



# Dodatak D: Plan rada / Pregled rada i stanje ostvarenja

Plan rada uključuje implementaciju svih navedenih funkcionalnosti. Potrebno je dobro organizirati vrijeme i nastaviti tempom kojim smo do sada radili. Nastavak rada na projektu sastoji se od izrade i testiranja svih mogućnosti web-aplikacije. Kao i do sada, rad na projektu pratit će vođenje detaljne dokumentacije.