

Programsko inženjerstvo

Ak. god. 2021./2022.

Moj Kvart

Dokumentacija, Rev. 2

Grupa: *Ronins*

Voditelj: *Mihail Miličević*

Datum predaje: *14.1.2022.*

Nastavnik: *Nikolina Frid*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	6
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	13
3.1.2 Sekvencijski dijagrami	41
3.2 Ostali zahtjevi	47
4 Arhitektura i dizajn sustava	48
4.1 Baza podataka	49
4.1.1 Opis tablica	49
4.1.2 Dijagram baze podataka	55
4.2 Dijagram razreda	56
4.3 Dijagram stanja	63
4.4 Dijagram aktivnosti	65
4.5 Dijagram komponenti	67
5 Implementacija i korisničko sučelje	68
5.1 Korištene tehnologije i alati	68
5.2 Ispitivanje programskog rješenja	69
5.2.1 Ispitivanje komponenti	69
5.2.2 Ispitivanje sustava	75
5.3 Dijagram razmještaja	79
5.4 Upute za puštanje u pogon	80
6 Zaključak i budući rad	84
Popis literature	86
Indeks slika i dijagonama	88

Dodatak: Prikaz aktivnosti grupe

89

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Napravljen predložak.	Mihael Miličević	19.10.2021.
0.2	Dodani dionici u projektu, te aktori i njihovi funkcionalni zahtjevi.	Mihael Miličević	20.10.2021.
0.3	Dodani obrasci uporabe.	Mihael Miličević, Tomislav Žiger	24.10.2021.
0.4	Dodan opis projektnog zadatka.	Mihael Miličević	24.10.2021.
0.4.1	Dodani ostali zahtjevi.	Tomislav Žiger	27.10.2021.
0.4.2	Ispravak obrazaca uporabe.	Mihael Miličević	28.10.2021.
0.4.3	Dodani dijagrami obrazaca uporabe.	Mihael Miličević, Tomislav Žiger	3.11.2021.
0.4.4	Dodani sekvencijski dijagrami.	Mihael Miličević, Tomislav Žiger	3.11.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
0.5	Dodana specifikacija baze podataka.	Andrija Banić	3.11.2021.
0.5.1	Dodan opis arhitekture sustava.	Mihael Miličević	8.11.2021.
0.5.2	Dodan dijagram razreda.	Danijel Barišić, Mihael Miličević, Tomislav Žiger	11.11.2021.
0.5.3	Doradjeni dijagrami razreda.	Mihael Miličević, Tomislav Žiger	14.11.2021.
1.0	Verzija prve predaje.	Mihael Miličević	14.11.2021.
1.1	Dodan dijagram razmještaja.	Mihael Miličević	29.12.2021.
1.2	Dodan dijagram komponenti.	Mihael Miličević	29.12.2021.
1.3	Dodan dijagram aktivnosti.	Mihael Miličević	29.12.2021.
1.4	Dodan dijagram stanja.	Mihael Miličević	29.12.2021.
1.5	Dodano ispitivanje sustava.	Mihael Miličević	31.12.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.5.1	Dodano ispitivanje komponenti.	Danijel Barišić	5.1.2022.
1.5.2	Dodane korištene tehnologije i alati.	Mihael Miličević	8.1.2022.
1.5.3	Ažuriran opis baze podataka.	Mihael Miličević	9.1.2022.
1.6	Dodane upute za puštanje u pogon.	Danko Čurlin, Mihael Miličević	10.1.2022.
1.7	Dodan zaključak projekta.	Mihael Miličević	10.1.2022.
1.7.1	Ažurirano poglavlje o dijagramima razreda.	Mihael Miličević	11.1.2022.
1.7.2	Dodani grafovi aktivnosti u repozitoriju.	Mihael Miličević	12.1.2022.
2.0	Verzija konačne predaje.	Mihael Miličević	12.1.2022.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti web aplikaciju pod nazivom "MojKvart" koja će omogućiti poboljšanje komunikacije i razmjenu informacija među stanovnicima jednog kvarta. Sama web aplikacija se sastoji od tri glavna dijela: "Forum", "Događaji" i "Vijeće četvrti", te postoje tri primarne vrste korisnika: "Obični stanovnik", "Viječnik" i "Moderator". Dodatno postoji i uloga "Administrator", iako administratori nisu stanovnici niti jedne četvrti, nego je njihova uloga održavanje sustava.

Svaki stanovnik četvrti koji poželi koristiti aplikaciju morat će prvo stvoriti korisnički račun. Kada otvori aplikaciju, prikazat će mu se stranica "Prijava". S obzirom da on trenutno nema korisnički račun, odabrat će opciju "Registracija", i to će ga preusmjeriti na stranicu na kojoj će morati ispuniti obrazac za stvaranje korisničkog računa. U obrascu će morati unijeti svoje ime, prezime, adresu stanovanja i adresu e-pošte, i pritom će ga se u sustavu raspoznavati primarno po njegovoj adresi e-pošte. Dodatno morat će odabrati i lozinku. Ukoliko pri registraciji korisnik odabere e-mail s kojim je već povezan neki korisnički račun u sustavu, ili ne unese sve podatke, ili unese neki podatak u krivom formatu (npr. napiše ime ili prezime malim slovom, ili za adresu unese adresu koja nije validna), dobit će prikladnu poruku o pogrešci, i tada će moći ispraviti te podatke i ponovno pokušati dovršiti registraciju. Ako se uspješno registrira, sustav će automatski po njegovoj adresi prepoznati kojem kvartu pripada, i preusmjerit će ga na početnu stranicu njegovog kvarta.

Svaki korisnik koji ima korisnički račun će se na stranici "Prijava" moći prijaviti u sustav i time dobiti pristup korisničkom dijelu sustava tako da unese svoj e-mail i lozinku. Ukoliko korisnik napravi neku pogrešku u ovom koraku, recimo unese e-mail s kojim nije povezan niti jedan korisnički račun u sustavu, ili unese lozinku koja nije ispravna, dobit će prikladnu poruku o pogrešci, i tada će moći ispraviti tražene podatke i ponovno se pokušati prijaviti. Ako se uspješno prijavi, bit će preusmjeren na početnu stranicu svog kvarta.

Na početnoj stranici korisnik vidi poveznice na tri glavne cjeline sustava za svoj kvart: "Forum", "Događaji" i "Vijeće četvrti". Dodatno, korisnik vidi i poveznice "Osobni podaci" i "Odjava". Odabirom opcije "Odjava" korisnik gubi pristup ko-

risničkom dijelu sustava i biva preusmjeren na stranicu "Prijava". Odabirom poveznice "Osobni podaci" korisnik biva preusmjeren na stranicu na kojoj vidi svoje osobne podatke - ime, prezime, adresa stanovanja, adresa e-pošte i lozinka. Korisnik može mijenjati sve osobne podatke osim e-pošte. Ako korisnik promijeni adresu stanovanja, i nova adresa se ne nalazi u istom kvartu kao i stara, korisnik će izgubiti uloge "Vijećnik" i "Moderator", ako ih ima. Dodatno, korisnik ovdje ima i opciju "Zahtjev za ulogom", kojom može administratorima sustava poslati zahtjev da mu se dodijeli uloga vijećnika ili moderatora. Konačno, korisnik ovdje ima opciju "Obriši korisnički račun", kojom može obrisati svoj korisnički račun, i tada gubi pristup korisničkom dijelu sustava i biva preusmjeren na stranicu "Prijava".

Na početnoj stranici, odabirom poveznice "Događaji", korisnik biva preusmjeren na stranicu sa svim potvrđenim događajima u svojoj četvrti. Ovdje može vidjeti vidjeti te događaje, te predložiti vlastite događaje, za što mora unijeti naziv događaja, mjesto, vrijeme, trajanje i kratak opis. Prijedloge događaja odobravaju moderatori pojedinih četvrti.

Odabirom poveznice "Forum" na početnoj stranici, korisnik biva preusmjeren na stranicu "Forum" gdje dobije prikaz svih tema na forumu za svoju četvrt. Korisnik može čitati teme i objave u njima, stvarati vlastite teme, komentirati objave drugih korisnika, uređivati i brisati svoje odgovore. U slučaju da se u temi na forumu obrišu sve objave, tema se automatski obriše. Pri stvaranju teme, korisnik mora napisati prvu objavu u njoj.

Konačno, odabirom poveznice "Vijeće četvrti" na početnoj stranici, korisnik biva preusmjeren na stranicu vijeća, gdje može vidjeti popis izvješća s prethodnih sjednica. Korisnik može odabratи pojedino izvješće za više detalja, a može i automatski otvoriti temu na forumu vezanu za pojedino izvješće. Takve teme je moguće otvoriti jednom po izvješću, i vizualno se razlikuju od ostalih tema na forumu, te u njima prva objava bude tekst izvješća. Tada se u tekstu izvješća ugradi poveznica na raspravu na temu za to izvješće.

Uz sve navedene mogućnosti, korisnici s ulogom "Vijećnik" mogu stvarati nova izvješća u cjelini "Vijeće četvrti", uređivati izvješća kojima su oni autori, i mogu brisati izvješća.

Moderatori mogu pregledavati prijedloge događaja za svoj kvart, i te prijedloge mogu objaviti ili obrisati. Također mogu uređivati prijedloge koji ne zadovoljavaju jezični standard. Moderatori također mogu brisati objave u temama na forumu, te cijele teme.

Kada se korisnici s ulogom "Administrator" prijave u sustav, dočeka ih drukčija početna stranica od one koju imaju korisnici koji su stanovnici nekog kvarta. Administratori na svojoj početnoj stranici imaju tri glavne cjeline, a to su "Korisnici", "Kvartovi", i "Zahtjevi uloga", te dodatno imaju opcije "Osobni podaci" i "Odjava". Odjava funkcionira isto kao i kod ostalih korisnika, administratori tada gube pristup korisničkom dijelu sučelja i bivaju preusmjereni na stranicu "Prijava". Pregled osobnih podataka za administratore je sličan pregledu osobnih podataka za ostale korisnike, a razlike su te da administratori nemaju adresu, odnosno nisu stanovnici niti jednog kvarta, te dodatno, administratori ne mogu obrisati svoj korisnički račun.

Odabirom opcije "Korisnici", administratori bivaju preusmjereni na stranicu na kojoj vide popis svih registriranih korisnika u sustavu. Prikazani korisnici su sortirani po kvartovima, a unutar kvartova po ulogama, a administratori sustava su također prikazani u svojoj cjelini. Ovdje administratori mogu odabrati bilo kojeg korisnika i vidjeti njegove osobne podatke. Ovdje također mogu korisniku dodijeliti ili oduzeti ulogu "Vijećnik" ili "Administrator". Administratori ovdje mogu svakog pojedinog korisnika koji nema ulogu "Administrator" privremeno blokirati ili deblokirati, a mogu i trajno obrisati korisnički račun svakog korisnika koji nema ulogu "Administrator". Također, klikom na poveznicu "Blokirani korisnici", administratori mogu vidjeti popis svih blokiranih korisnika.

Odabirom opcije "Zahtjevi uloga", administratori mogu vidjeti sve nerazriješene zahtjeve uloga. Ovdje mogu odabrati pojedini zahtjev, i prihvati ga ili odbiti.

Odabirom opcije "Kvartovi", administratori mogu vidjeti popis svih kvartova. Ovdje sada mogu odabrati pojedini kvart i pregledati podatke o tom kvartu, uključujući ulice koje mu pripadaju i korisnike aplikacije koji su stanovnici tog kvarta.

Dodatno, pod opcijom "Kvartovi", administratori mogu odabrati opciju "Dodaj kvart", čime definiraju novi kvart u sustavu. Za kvart moraju odabrati ime, i to ime kvarta mora biti jedinstveno među imenima kvartova. Dodatno, moraju odabrati koje ulice pripadaju tom kvartu, te imena ulica također moraju biti jedinstvena, a za svaku ulicu moraju odabrati i raspon dozvoljenih brojeva.

Konačno, pri pregledu kvartova, administratori mogu uređivati podatke za pojedine kvartove, te ih mogu i obrisati. Ukoliko promjenom podataka kvarta adresa nekog korisnika prestane biti validna, ili ukoliko se njegov kvart i time sve ulice u tom kvartu obrišu, pri idućoj prijavi korisnik će morati promijeniti svoju adresu prije nastavka korištenja sustava.

Za ovu svrhu su stanovnici pojedinih kvartova mogli koristiti razne aplikacije općenitije namjene, npr. Facebook grupe ili Discord serveri, ali prednost ovakvog sustava je upravo prilagođenost toj specifičnoj namjeni spajanja ljudi pojedinog kvarta, te naglasak i jasnija definicija upravo onih funkcionalnosti koje su bitnije za neki kvart.

Sustav bi se mogao prilagoditi i za mobilne uređaje, no svrha ovog projekta je napraviti web aplikaciju, a ideja izrade mobilne aplikacije je jedna od mogućnosti za kasnije faze projekta. Postoje također razne ideje manje važnosti koje nisu nužne za osnovno funkcioniranje projekta, ali bi poboljšale cjelokupni sustav, a neke od njih su:

- mogućnost da svaki korisnik može pregledati osobne podatke svakog drugog korisnika
- mogućnost da korisnici biraju svoje nadimke, i da se, umjesto njihovih imena i prezimena, ti nadimci prikazuju u sustavu
- 2FA prilikom prijave
- primanje obavijesti na mail
- mogućnost pregleda svih objava koje je korisnik imao na forumu
- mogućnost pregleda raznih statistika na forumu

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik (naručitelj)
2. Stanovnik četvrti
 - (a) Obični stanovnik
 - (b) Vijećnik
 - (c) Moderator
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) se registrirati u sustav stvaranjem novog korisničkog računa, za što su mu potrebni ime, prezime, adresa stanovanja, adresa e-pošte i lozinka
2. Neprijavljeni korisnik (inicijator) može:
 - (a) se prijaviti u sustav, za što su mu potrebni adresa e-pošte i lozinka
3. Stanovnik četvrti (inicijator) može:
 - (a) čitati teme na "Forumu"
 - (b) otvarati teme na "Forumu"
 - (c) odgovarati u temama na "Forumu"
 - (d) uređivati svoje odgovore na "Forumu"
 - (e) ukloniti svoje odgovore na "Forumu"
 - (f) otvoriti temu na "Forumu" vezanu za objavu na "Vijeću četvrti", ako za tu objavu već nije otvorena tema
 - (g) čitati objave na "Vijeću četvrti"
 - (h) vidjeti najave događaja u cjelini "Događaji"

- (i) predlagati najave budućih događaja u cjelini "Događaji", za što je potrebno navesti naziv, mjesto, vrijeme, trajanje i kratak opis
- (j) poslati zahtjev administratorima za promjenu uloge (u "Vijećnika" ili "Moderatora")
- (k) promijeniti osobne podatke
- (l) odjaviti se iz sustava
- (m) obrisati svoj korisnički račun

4. Vijećnik (inicijator) može:

- (a) stvoriti objavu u cjelini "Vijeće četvrти"
- (b) urediti objavu u cjelini "Vijeće četvrти"
- (c) obrisati objavu u cjelini "Vijeće četvrти"

5. Moderator (inicijator) može:

- (a) pregledati najave događaja koje predlažu stanovnici
- (b) objaviti najave događaja u cjelini "Događaji"
- (c) odbaciti prijedloge događaja
- (d) urediti prijedloge događaja
- (e) ukloniti odgovore u temama na "Forumu"
- (f) ukloniti teme na Forumu

6. Administrator (inicijator) može:

- (a) vidjeti popis svih registriranih korisnika i njihove osobne podatke
- (b) definirati četvrt i područje koje ta četvrt obuhvaća
- (c) obrisati četvrti
- (d) urediti podatke o četvrti
- (e) vidjeti popis svih četvrti u sustavu
- (f) odabrati pojedinu četvrt s popisa četvrti i pristupiti svom korisničkom sadržaju te četvrti
- (g) pristupiti svim zahtjevima za uloge "Vijećnik" ili "Moderator"
- (h) odbiti zahtjev za dodjelu uloge
- (i) prihvati zahtjev za dodjelu uloge
- (j) dodijeliti ulogu "Moderator" ili "Vijećnik" korisniku
- (k) oduzeti ulogu "Moderator" ili "Vijećnik" korisniku
- (l) privremeno blokirati korisnika
- (m) pristupiti popisu blokiranih korisnika
- (n) deblokirati korisnika

(o) trajno obrisati profil korisnika

7. Baza podataka (sudionik):

(a) pohranjuje podatke o korisnicima i njihovim ovlastima

(b) pohranjuje podatke o četvrtima

i. ulice koje im pripadaju

ii. teme na "Forumu"

iii. izvješća s "Vijeća četvrti"

iv. događaje iz sekcije "Događaji"

3.1.1 Obrasci uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Registracija"
 2. Korisnik unosi tražene podatke - ime, prezime, adresa stanovanja, adresa e-pošte i lozinka
 3. Korisnik odabire opciju "Podnesi"
 4. Korisnik dobiva obavijest o uspješnoj registraciji
 5. Korisnik biva preusmjerjen na početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odabire e-mail s kojim je već povezan korisnički račun u sustavu, ili unese tražene podatke u krivom formatu
 1. Korisnik dobiva obavijest o pogrešci
 2. Korisnik mijenja podatke i ponovno odabire opciju "Podnesi", ili odustaje od registracije

UC2 - Prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Dobiti pristup korisničkom dijelu sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prijava"
 2. Korisnik unosi e-mail adresu i lozinku
 3. Korisnik odabire opciju "Podnesi"
 4. Korisnik dobiva pristup korisničkom dijelu sustava
 5. Korisnik biva preusmjerjen na početnu stranicu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik unosi e-mail s kojim nije povezan niti jedan korisnički račun, ili unosi krivu lozinku, ili unosi tražene podatke u krivom formatu
 1. Korisnik dobiva obavijest o pogrešci

2. Korisnik mijenja podatke i ponovno odabire opciju "Podnesi", ili odustaje od prijave

UC3 - Odjava

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** -
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Odjava"
 2. Korisnik gubi pristup korisničkom dijelu sustava
 3. Korisnik biva preusmјeren na stranicu "Prijava"

UC4 - Pregled osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Vidjeti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Osobni podaci"
 2. Korisnik biva preusmјeren na stranicu "Osobni podaci" gdje dobije prikaz svojih osobnih podataka

UC5 - Promjena osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava svoje osobne podatke (UC4)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Promjena osobnih podataka"
 2. Korisnik mijenja svoje podatke
 3. Korisnik odabire opciju "Spremi promjene"
 4. Baza podataka se ažurira
 5. Korisnik biva preusmјeren na stranicu "Osobni podaci"

- **Opis mogućih odstupanja:**

- 3.a Korisnik mijenja svoje osobne podatke, ali ne spremi promjene
 1. Sustav upozori korisnika da nije spremio promjene
 2. Korisnik spremi promjene, ili odustane od njih
- 3.b Korisnik pokuša promijeniti e-mail adresu
 1. Sustav upozori korisnika da ne može promijeniti e-mail adresu
 2. Korisnik promijeni neke druge osobne podatke, ili odustane od promjena
- 3.c Korisnik pokuša promijeniti neki osobni podatak u nedozvoljeni format (npr. ime koje počinje malim slovom), ili pokuša promijeniti adresu na neku adresu koja ne postoji u sustavu
 1. Sustav upozori korisnika o pogrešci
 2. Korisnik pokuša promijeniti svoje osobne podatke u ispravan format, ili odustane od promjena
- 3.d Korisnik je promijenio adresu, i nova adresa ne pripada kvartu kojem je pripadala njegova stara adresa
 1. Sustav dodijeli korisnika novom kvartu
 2. Sustav korisniku oduzima uloge "Vijećnik" i "Moderator", ako ih ima
 3. Baza podataka se ažurira
 4. Korisnik biva preusmjeren na stranicu "Osobni podaci"

UC6 - Brisanje korisničkog računa

- **Glavni sudionik:** Korisnik

- **Cilj:** Obrisati korisnički račun

- **Sudionici:** Baza podataka

- **Preduvjeti:**

1. Korisnik je prijavljen u sustav
2. Korisnik pregledava svoje osobne podatke (UC4)
3. Korisnik nema ulogu "Administrator"

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Obriši korisnički račun"
2. Sustav upozori korisnika da je brisanje korisničkog računa trajna i ne-povratna akcija
3. Korisnik potvrđuje svoj odabir
4. Korisnički račun se izbriše iz baze podataka

5. Korisnik biva preusmjeren na stranicu "Prijava"
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustane od brisanja svog računa
 1. Korisnik biva preusmjeren na stranicu "Osobni podaci"

UC7 - Promjena adrese radi ispravljanja nevažećeg statusa stanovanja

- **Glavni sudionik:** Korisnik
- **Cilj:** Promijeniti adresu i omogućiti korisniku pristup korisničkom dijelu sustava
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnikov status stanovanja je "Nevažeći"
- **Opis osnovnog tijeka:**
 1. Korisnik ispunjava obrazac u kojem unosi novu adresu
 2. Korisnik odabire opciju "Spremi promjene"
 3. Baza podataka se ažurira
 4. Korisnik biva preusmjeren na početnu stranicu
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nije unio novu adresu, ili je adresa koju je korisnik unio nevažeća
 1. Sustav upozorava korisnika o pogrešci
 2. Korisnik ispravlja pogrešku i ponovno odabire opciju "Spremi promjene", ili odustaje od ispravljanja adrese

UC8 - Pregled događaja u cjelini "Događaji"

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz potvrđenih događaja u korisnikovom kvartu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabere opciju "Događaji"
 2. Korisnik biva preusmjeren na stranicu "Događaji" gdje dobije prikaz potvrđenih događaja u njegovom kvartu

UC9 - Stvaranje prijedloga događaja

- **Glavni sudionik:** Korisnik

- **Cilj:** Stvoriti prijedlog događaja
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava događaje (UC8)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Novi prijedlog događaja"
 2. Korisnik ispunjava polja "naziv", "mjesto", "vrijeme", "trajanje" i "kratki opis"
 3. Korisnik odabire opciju "Objavi prijedlog događaja"
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije ispunio sva polja prilikom stvaranja prijedloga događaja
 1. Sustav upozori korisnika da sva polja nisu ispunjena
 2. Korisnik ispuni preostala polja i odabire opciju "Objavi prijedlog događaja", ili odustane od prijedloga
 - 3.b Korisnik pokuša napustiti stranicu prije biranja opcije "Objavi prijedlog događaja"
 1. Sustav upozori korisnika da se skica prijedloga događaja neće spremiti
 2. Korisnik odabire opciju "Objavi prijedlog događaja", ili odustaje od objave prijedloga događaja

UC10 - Pregled prijedloga događaja

- **Glavni sudionik:** Moderator
- **Cilj:** Pregled prijedloga događaja u cjelini "Događaji"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Moderator"
 3. Korisnik pregledava događaje (UC8)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prijedlozi događaja"
 2. Korisnik biva preusmjerena na stranicu "Prijedlozi događaja" gdje dobije prikaz svih prijedloga događaja za svoj kvart

UC11 - Uređivanje prijedloga događaja

- **Glavni sudionik:** Moderator
- **Cilj:** Uređivanje prijedloga događaja u cjelini "Događaji"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Moderator"
 3. Korisnik pregledava prijedloge događaja (UC10)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prijedlog događaja
 2. Korisnik odabire opciju "Uredi prijedlog događaja"
 3. Korisnik uređuje polja za odabrani prijedlog događaja
 4. Korisnik odabire opciju "Spremi promjene"
 5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 4.a Korisnik nije spremio promjene
 1. Sustav upozori korisnika da promjene nisu spremljene
 2. Korisnik odabire opciju "Spremi promjene", ili odustane od promjena
 - 4.b Korisnik obriše neko polje, tj. ostavi ga praznim nakon unošenja promjena
 1. Sustav upozori korisnika da niti jedno polje u formularu ne može biti prazno
 2. Korisnik ispuni to polje i odabire opciju "Spremi promjene", ili odustane od promjena

UC12 - Objava prijedloga događaja

- **Glavni sudionik:** Moderator
- **Cilj:** Objava predloženog događaja u cjelini "Događaji"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Moderator"
 3. Korisnik pregledava prijedloge događaja (UC10)
- **Opis osnovnog tijeka:**

1. Korisnik odabire prijedlog događaja
2. Korisnik odabire opciju "Objavi događaj"
3. Baza podataka se ažurira

UC13 - Brisanje prijedloga događaja

- **Glavni sudionik:** Moderator
- **Cilj:** Brisanje prijedloga događaja u cjelini "Događaji"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Moderator"
 3. Korisnik pregledava prijedloge događaja (UC10)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire prijedlog događaja
 2. Korisnik odabire opciju "Obriši prijedlog događaja"
 3. Baza podataka se ažurira

UC14 - Pregled tema na "Forumu"

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz tema na "Forumu"
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire cjelinu "Forum"
 2. Korisnik biva preusmjeren na stranicu "Forum" gdje gdje dobije prikaz tema na "Forumu", poredanih po vremenu zadnjeg odgovora

UC15 - Prikaz teme na "Forumu"

- **Glavni sudionik:** Korisnik
- **Cilj:** Odabrati temu na "Forumu" i vidjeti sve objave u njoj
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava "Forum" (UC14)
- **Opis osnovnog tijeka:**
 1. Korisnik u cjelini "Forum" odabire željenu temu

2. Prikazuje se odabrana tema

UC16 - Otvaranje teme na "Forumu"

- **Glavni sudionik:** Korisnik
- **Cilj:** Otvoriti temu na "Forumu"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava "Forum" (UC14)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Otvori novu temu"
 2. Korisnik ispunjava obrazac s imenom novootvorene teme i prvom objavom u toj temi
 3. Korisnik odabire opciju "Otvori temu"
 4. Baza podataka se ažurira
 5. Korisnik biva preusmjeren na stranicu koja odgovara novootvorenoj temi
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije ispunio sva polja u obrascu s imenom i prvom objavom u temi
 1. Sustav upozorava korisnika da mora ispuniti sva polja u obrascu
 2. Korisnik ispunjava polja i ponovno odabire opciju "Otvori temu", ili odustaje od otvaranja teme

UC17 - Odgovaranje na objavu u temi na "Forumu"

- **Glavni sudionik:** Korisnik
- **Cilj:** Odgovoriti na željenu objavu u temi na "Forumu"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava temu na "Forumu" (UC15)
- **Opis osnovnog tijeka:**
 1. Korisnik unutar teme na "Forumu" na odabranoj objavi odabire opciju "Odgovori"
 2. Otvara se polje za odgovor
 3. Korisnik ispunjava polje za odgovor tekstom
 4. Korisnik bira opciju "Objavi odgovor"

5. Baza podataka se ažurira
 6. Korisnik biva preusmjeren na stranicu koja odgovara toj temi na "Forumu"
- **Opis mogućih odstupanja:**
 - 3.a Korisnik pokuša objaviti odgovor bez teksta
 1. Sustav upozorava korisnika da odgovor na "Forumu" ne može biti prazan tekst
 2. Korisnik doda tekst u polje za tekst odgovora, ili odustane od objave odgovora

UC18 - Uređivanje odgovora u temi na "Forumu"

- **Glavni sudionik:** Korisnik
- **Cilj:** Uređivanje odgovora u temi na "Forumu"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava temu na "Forumu" (UC15)
 3. Korisnik je autor odgovora kojeg odabire
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Uredi odgovor"
 2. Otvara se polje za odgovor koje sadrži trenutnu verziju teksta odgovora
 3. Korisnik uređuje tekst
 4. Korisnik odabire opciju "Spremi promjene"
 5. Baza podataka se ažurira
 6. Korisnik biva preusmjeren na stranicu koja odgovara toj temi na "Forumu"
- **Opis mogućih odstupanja:**
 - 4.a Korisnik napušta stranicu iako uređeni odgovor nije objavio
 1. Sustav upozorava korisnika da uređeni tekst neće biti spremišten ako napusti stranicu
 2. Korisnik odabire opciju "Spremi promjene", ili odustane od promjena
 - 4.b Korisnik je obrisao sav tekst odgovora i odabrao opciju "Spremi promjene"
 1. Sustav upozorava korisnika da tekst objave ne smije biti prazan
 2. Korisnik u polje za tekst odgovora doda neki tekst, ili odustane od

promjene odgovora

UC19 - Brisanje odgovora u temi na "Forumu"

- **Glavni sudionik:** Korisnik
- **Cilj:** Obrisati odgovor u temi na "Forumu"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava temu na "Forumu" (UC15)
 3. Korisnik odabire odgovor čiji autor je on, ili korisnik ima ulogu "Moderator"
- **Opis osnovnog tijeka:**
 1. Korisnik pronađe odgovor koji želi obrisati
 2. Korisnik odabire opciju "Izbriši odgovor"
 3. Sustav upozorava korisnika da je brisanje odgovora trajna i nepovratna akcija
 4. Korisnik potvrđuje svoj odabir
 5. Baza podataka se ažurira
 6. Korisnik biva preusmjeren na stranicu koja odgovara toj temi na "Forumu"
- **Opis mogućih odstupanja:**
 - 4.a Korisnik odustane od brisanja odgovora
 1. Prozor za brisanje odgovora se ugasi
 - 4.b Korisnik je obrisao posljednji odgovor u odgovarajućoj temi
 1. Sustav automatski obriše tu temu na "Forumu"
 2. Baza podataka se ažurira
 3. Korisnik biva preusmjeren na stranicu "Forum"

UC20 - Brisanje teme na "Forumu"

- **Glavni sudionik:** Moderator
- **Cilj:** Obrisati temu na "Forumu"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava "Forum" (UC14)
 3. Korisnik ima ulogu "Moderator"

- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Obriši temu"
 2. Sustav upozorava korisnika da je brisanje teme trajna i nepovratna akcija
 3. Korisnik potvrđuje svoju odluku i tema se obriše
 4. Baza podataka se ažurira
 5. Korisnik biva preusmjeren na stranicu "Forum"
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustane od brisanja teme
 1. Korisnik biva preusmjeren na stranicu "Forum"

UC21 - Pregled cjeline "Vijeće četvrti"

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled svih izvješća u cjelini "Vijeće četvrti"
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Vijeće četvrti"
 2. Korisnik biva preusmjeren na stranicu "Vijeće četvrti" gdje dobije prikaz svih izvješća, poredanih kronološki od najsvježijeg

UC22 - Prikaz izvješća u "Vijeću četvrti"

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz jednog izvješća u cjelini "Vijeće četvrti"
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik pregledava "Vijeće četvrti" (UC21)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire izvješće koje želi pregledati
 2. Korisnik biva preusmjeren na stranicu koja odgovara izvješću koje je odabrao

UC23 - Objava novog izvješća na "Vijeću četvrti"

- **Glavni sudionik:** Vijećnik
- **Cilj:** Objaviti novo izvješće na "Vijeću četvrti"
- **Sudionici:** Baza podataka

- **Preduvjeti:**

1. Korisnik je prijavljen u sustav
2. Korisnik pregledava "Vijeće četvrti" (UC21)
3. Korisnik ima ulogu "Vijećnik"

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Stvaranje novog izvješća"
2. Otvara se polje za pisanje izvješća
3. Korisnik ispunjava polje tekstrom izvješća
4. Korisnik odabire opciju "Objavi izvješće"
5. Baza podataka se ažurira
6. Korisnik biva preusmjeren na stranicu "Vijeća četvrti"

- **Opis mogućih odstupanja:**

- 4.a Korisnik pokuša objaviti prazno izvješće bez teksta
 1. Sustav upozorava korisnika da izvješće s "Vijeća četvrti" ne može biti prazan tekst
 2. Korisnik doda tekst u polje za tekst izvješća, ili odustane od objave izvješća
- 4.b Korisnik pokuša napustiti stranicu bez biranja opcije "Objavi izvješće"
 1. Sustav upozorava korisnika da skica izvješća neće biti objavljena
 2. Korisnik odabire opciju "Objavi izvješće", ili odustaje od objave

UC24 - Uređivanje izvješća na "Vijeću četvrti"

- **Glavni sudionik:** Vijećnik

- **Cilj:** Urediti izvješće

- **Sudionici:** Baza podataka

- **Preduvjeti:**

1. Korisnik je prijavljen u sustav
2. Korisnik pregledava izvješće s "Vijeća četvrti" (UC22)
3. Korisnik ima ulogu "Vijećnik"
4. Korisnik je autor izvješća kojeg pregledava

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Uredi izvješće"
2. Otvara se tekstualno polje koje sadrži trenutnu verziju izvješća
3. Korisnik uređuje tekst
4. Korisnik odabire opciju "Spremi promjene"
5. Baza podataka se ažurira

6. Korisnik biva preusmjeren na stranicu koja odgovara tom izvješću na "Vijeću četvrti"

- **Opis mogućih odstupanja:**

4.a Korisnik napušta stranicu iako uređeno izvješće nije objavio

1. Sustav upozorava korisnika da uređeni tekst neće biti spremjen ako napusti stranicu

2. Korisnik odabire opciju "Spremi promjene", ili odustane od promjena

4.b Korisnik je obrasio sav tekst izvješća i odabrao opciju "Spremi promjene"

1. Sustav upozorava korisnika da tekst izvješća ne smije biti prazan

2. Korisnik u polje za tekst izvješća doda neki tekst, ili odustane od promjene izvješća

UC25 - Brisanje izvješća na "Vijeću četvrti"

- **Glavni sudionik:** Vijećnik

- **Cilj:** Obrisati izvješće u cjelini "Vijeće četvrti"

- **Sudionici:** Baza podataka

- **Preduvjeti:**

1. Korisnik je prijavljen u sustav

2. Korisnik pregledava izvješće s "Vijeća četvrti" (UC22)

3. Korisnik ima ulogu "Vijećnik"

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Obriši izvješće"

2. Sustav upozorava korisnika da je brisanje izvješća trajna i nepovratna akcija

3. Korisnik potvrđuje svoju odluku i izvješće se obriše

4. Baza podataka se ažurira

5. Korisnik biva preusmjeren na stranicu "Vijeće četvrti"

- **Opis mogućih odstupanja:**

3.a Korisnik odustane od brisanja izvješća

1. Korisnik biva preusmjeren na stranicu koja odgovara tom izvješću

UC26 - Otvaranje teme na "Forumu" vezanu za izvješće s "Vijeća četvrti"

- **Glavni sudionik:** Korisnik

- **Cilj:** Na "Forumu" stvoriti temu vezanu uz izvješće na "Vijeću četvrti"

- **Sudionici:** Baza podataka

- **Preduvjeti:**

1. Korisnik je prijavljen u sustav
2. Korisnik pregledava izvješće s "Vijeća četvrти" (UC22)
3. Za odabranu izvješće nije već otvorena tema na "Forumu"

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Otvori temu na 'Forumu'"
2. Korisnik ispunjava obrazac s prvom objavom u novootvorenoj temi, za ime teme se automatski postavlja naslov izvješća, a kao nulta objava u novootvorenoj temi se postavlja tekst izvješća s autorom "Vijeće četvrти"
3. Korisnik odabire opciju "Otvori temu"
4. Otvara se nova tema na cjelini "Forum"
5. Baza podataka se ažurira
6. Korisnik biva preusmјeren na novootvorenu temu na "Forumu"

- **Opis mogućih odstupanja:**

- 3.a Korisnik nije ispunio obrazac s prvom objavom u temi
 1. Sustav upozorava korisnika da mora ispuniti sva polja u obrascu
 2. Korisnik ispunjava polja i ponovno odabire opciju "Otvori temu", ili odustaje od otvaranja teme

UC27 - Podnošenje zahtjeva za dodjelu uloge

- **Glavni sudionik:** Korisnik

- **Cilj:** Priložiti zahtjev za promjenu uloge

- **Sudionici:** Baza podataka

- **Preduvjeti:**

1. Korisnik je prijavljen u sustav
2. Korisnik nema ulogu za koju šalje zahtjev
3. Korisnik pregledava svoje osobne podatke (UC4)
4. Korisnik već nije predao zahtjev za ulogom koju sad traži čiji je status trenutno "Nerazriješen"
5. Korisnik nema ulogu "Administrator"

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Novi zahtjev za ulogom"
2. Korisnik odabire ulogu za koju šalje zahtjev
3. Korisnik pošalje zahtjev
4. Baza podataka se ažurira

- **Opis mogućih odstupanja:**

- 3.a Korisnik pokuša poslati zahtjev bez da je odabrao ulogu
 1. Sustav upozori korisnika da nije odabrao ulogu
 2. Korisnik odabere ulogu, ili odustane od zahtjeva
- 3.b Korisnik pokuša ugasiti prozor za zahtjev za ulogu bez da je poslao zah-tjev
 1. Sustav upozori korisnika da nije poslao zahtjev
 2. Korisnik pošalje zahtjev, ili odustane od njega

UC28 - Pregled svih korisnika u sustavu

- **Glavni sudionik:** Administrator
- **Cilj:** Pristupiti popisu svih korisnika u sustavu
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Korisnici"
 2. Korisnik biva preusmjeren na stranicu "Korisnici" gdje dobije popis svih korisnika sustava

UC29 - Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati osobne podatke korisnika (radi jednoznačnosti, u dalnjem tekstu: objekt)
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava sve korisnike u sustavu (UC28)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire objekta čije osobne podatke želi pregledati
 2. Korisnik biva preusmjeren na stranicu s osobnim podacima objekta

UC30 - Dodjela uloge

- **Glavni sudionik:** Administrator
- **Cilj:** Dodijeliti ulogu korisniku (radi jednoznačnosti, u dalnjem tekstu: objekt)

- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava osobne podatke objekta (UC29)
 4. Objekt nema ulogu koju mu korisnik želi dodijeliti
 5. Objekt nema ulogu "Administrator"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire ulogu koju želi dodijeliti objektu
 2. Baza podataka se ažurira
 3. Korisnik biva preusmјeren na stranicu s osobnim podacima objekta
- **Opis mogućih odstupanja:**
 - 1.a Postoji zahtjev za ulogom sa statusom "Nerazriješen" čiji je autor objekt i kojem je uloga jednaka ulozi koji je korisnik manualno dodijelio objektu
 1. Status zahtjeva za ulogu se automatski promijeni u "Prihvaćen"
 2. Baza podataka se ažurira
 3. Korisnik biva preusmјeren na stranicu s osobnim podacima objekta

UC31 - Oduzimanje uloge

- **Glavni sudionik:** Administrator
- **Cilj:** Oduzeti ulogu korisniku (radi jednoznačnosti, u dalnjem tekstu: objekt)
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava osobne podatke objekta (UC29)
 4. Objekt ima ulogu koju mu korisnik želi oduzeti
 5. Objekt nema ulogu "Administrator"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire ulogu koju želi oduzeti objektu
 2. Baza podataka se ažurira
 3. Korisnik biva preusmјeren na stranicu s osobnim podacima objekta

UC32 - Pregled zahtjeva za ulogu

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati zahtjeve korisnika za uloge

- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Zahtjevi uloga"
 2. Korisnik biva preusmjeren na stranicu "Zahtjevi uloga" gdje dobije pregled svih nerazriješenih zahtjeva uloga

UC33 - Prihvatanje zahtjeva za ulogu

- **Glavni sudionik:** Administrator
- **Cilj:** Prihvati zahtjev korisnika za ulogu
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava zahtjeve uloga (UC32)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire zahtjev
 2. Korisnik odabire opciju "Prihvati"
 3. Baza podataka se ažurira

UC34 - Odbijanje zahtjeva za ulogu

- **Glavni sudionik:** Administrator
- **Cilj:** Odbiti zahtjev korisnika za ulogu
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava zahtjeve uloga (UC32)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire zahtjev
 2. Korisnik odabire opciju "Odbij"
 3. Baza podataka se ažurira

UC35 - Pregled blokiranih korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati popis blokiranih korisnika
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava sve korisnike u sustavu (UC28)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Blokirani korisnici"
 2. Korisnik biva preusmjerena na stranicu gdje vidi popis blokiranih korisnika

UC36 - Blokiranje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Blokirati korisnika (radi jednoznačnosti, u dalnjem tekstu: objekt)
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava osobne podatke objekta (UC29)
 4. Objekt trenutno nije blokiran
 5. Objekt nema ulogu "Administrator"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Blokiraj korisnika"
 2. Baza podataka se ažurira
 3. Korisnik biva preusmjerena na stranicu s osobnim podacima objekta

UC37 - Deblokiranje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Deblokirati korisnika (radi jednoznačnosti, u dalnjem tekstu: objekt)
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"

3. Korisnik pregledava osobne podatke objekta (UC29)
 4. Objekt je trenutno blokiran
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Deblokiraj korisnika"
 2. Baza podataka se ažurira
 3. Korisnik biva preusmjerena na stranicu s osobnim podacima objekta

UC38 - Brisanje korisničkog računa drugog korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati korisnički račun drugog korisnika (radi jednoznačnosti, u dalnjem tekstu: objekt)
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava osobne podatke objekta (UC29)
 4. Objekt nema ulogu "Administrator"
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Obrisи korisnički račun"
 2. Sustav upozorava korisnika da je brisanje korisničkog računa trajna i nepovratna akcija
 3. Korisnik potvrđuje svoj odabir i briše korisnički račun drugog korisnika
 4. Baza podataka se ažurira
 5. Korisnik biva preusmjerena na stranicu s popisom svih korisnika u sustavu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustane od brisanja korisničkog računa
 1. Korisnik biva preusmjerena na pregled osobnih podataka objekta

UC39 - Pregled popisa kvartova

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati popis kvartova
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"

- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Kvartovi"
 2. Korisnik biva preusmjerena na stranicu "Kvartovi" gdje vidi popis kvartova

UC40 - Pregled kvarta

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati podatke o odabranom kvartu
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava popis kvartova (UC39)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire kvart koju želi pregledati
 2. Korisnik dobiva prikaz informacija o odabranom kvartu

UC41 - Definiranje kvarta

- **Glavni sudionik:** Administrator
- **Cilj:** Definirati novi kvart
- **Sudionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava popis kvartova (UC39)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Dodaj kvart"
 2. Korisnik ispunjava sve potrebne podatke - ime kvarta, imena ulica, za svaku ulicu raspon brojeva
 3. Korisnik odabire opciju "Spremi promjene"
 4. Baza podataka se ažurira
 5. Korisnik biva preusmjerena na stranicu "Kvartovi"
- **Opis mogućih odstupanja:**
 - 3.a Korisnik unese ime kvarta koje je već zauzeto, ili ime ulice koje je već zauzeto, ili za kvart ne definira niti jednu ulicu, ili za neku ulicu ne definira raspon brojeva, ili za neku ulicu definira mogućnost postojanja negativnih brojeva

1. Sustav obavještava korisnika o pogrešci
2. Korisnik ispravlja pogrešku i odabire opciju "Spremi promjene", ili odustane od definiranja četvrти

UC42 - Uređivanje kvarta

- **Glavni sudsionik:** Administrator
- **Cilj:** Promijeniti podatke o nekom kvartu
- **Sudsionici:** Baza podataka
- **Preduvjeti:**
 1. Korisnik je prijavljen u sustav
 2. Korisnik ima ulogu "Administrator"
 3. Korisnik pregledava podatke o kvartu (UC40)
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Uredi kvart"
 2. Korisnik stvara promjene
 3. Korisnik odabire opciju "Spremi promjene"
 4. Baza podataka se ažurira
 5. Korisnik biva preusmјeren na stranicu koja sadrži podatke o tom kvartu
- **Opis mogućih odstupanja:**
 - 3.a Korisnik odustane od promjena
 1. Korisnik biva preusmјeren na stranicu koja sadrži podatke o tom kvartu
 - 3.b Postoje korisnici sustava čije adrese su zbog promjena podataka o kvartu prestale biti važeće
 1. Sustav automatski postavi status stanovanja na "Nevažeći" svim korisnicima čije adrese su zbog promjena podataka kvarta prestale biti važeće
 2. Baza podataka se ažurira
 3. Korisnik biva preusmјeren na stranicu koja sadrži podatke o tom kvartu

UC43 - Brisanje kvarta

- **Glavni sudsionik:** Administrator
- **Cilj:** Obrisati kvart
- **Sudsionici:** Baza podataka
- **Preduvjeti:**

1. Korisnik je prijavljen u sustav
2. Korisnik ima ulogu "Administrator"
3. Korisnik pregledava podatke o kvartu (UC40)

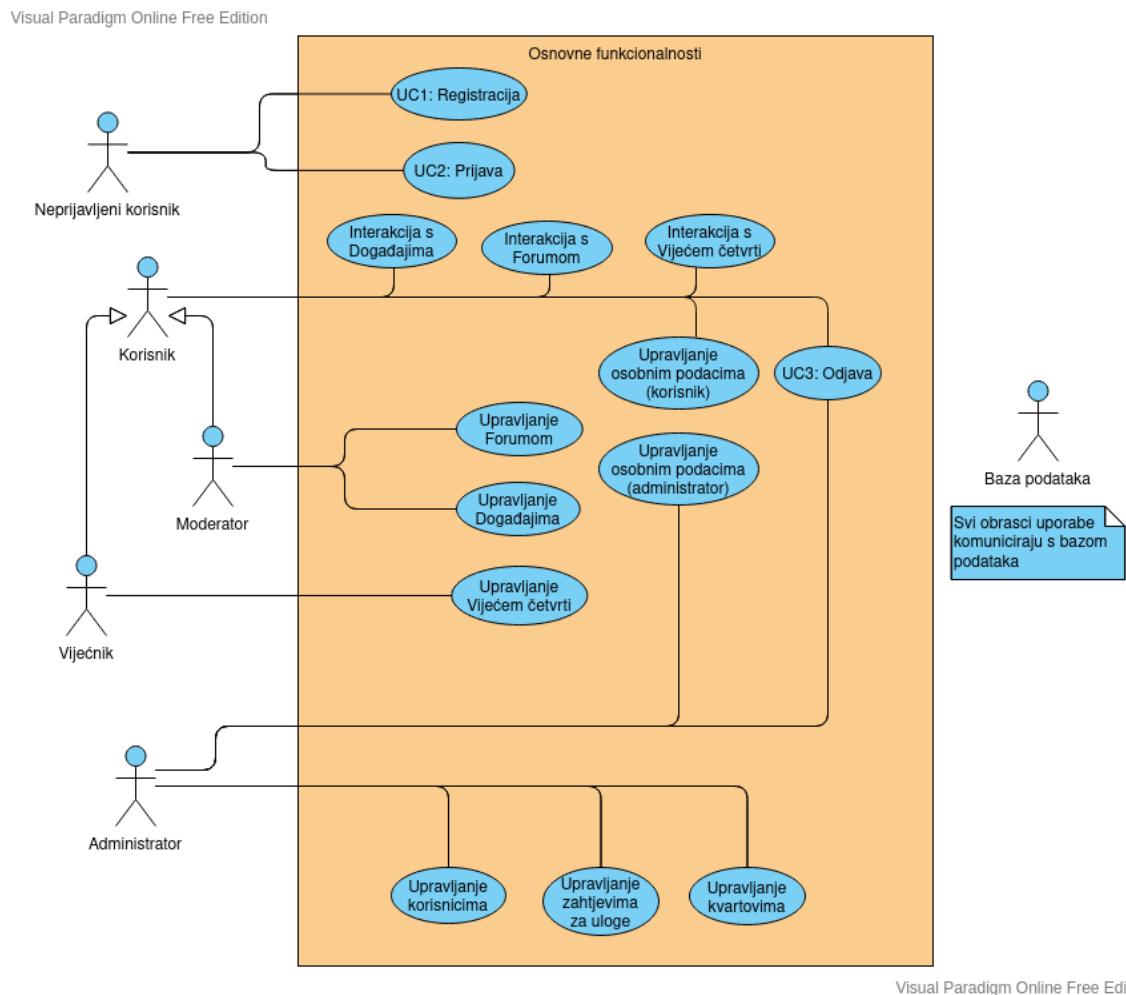
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Obriši kvart"
2. Sustav upozorava korisnika da je brisanje kvarta trajna i nepovratna akcija
3. Korisnik potvrđuje svoju odluku i obriše kvart
4. Iz baze podataka se obrišu svi podaci vezani za taj kvart - sve teme na "Forumu", sva izvješća s "Vijeća četvrti", svi događaji na "Dogadajima"
5. Baza podataka se ažurira
6. Korisnik biva preusmјeren na stranicu "Kvartovi"

- **Opis mogućih odstupanja:**

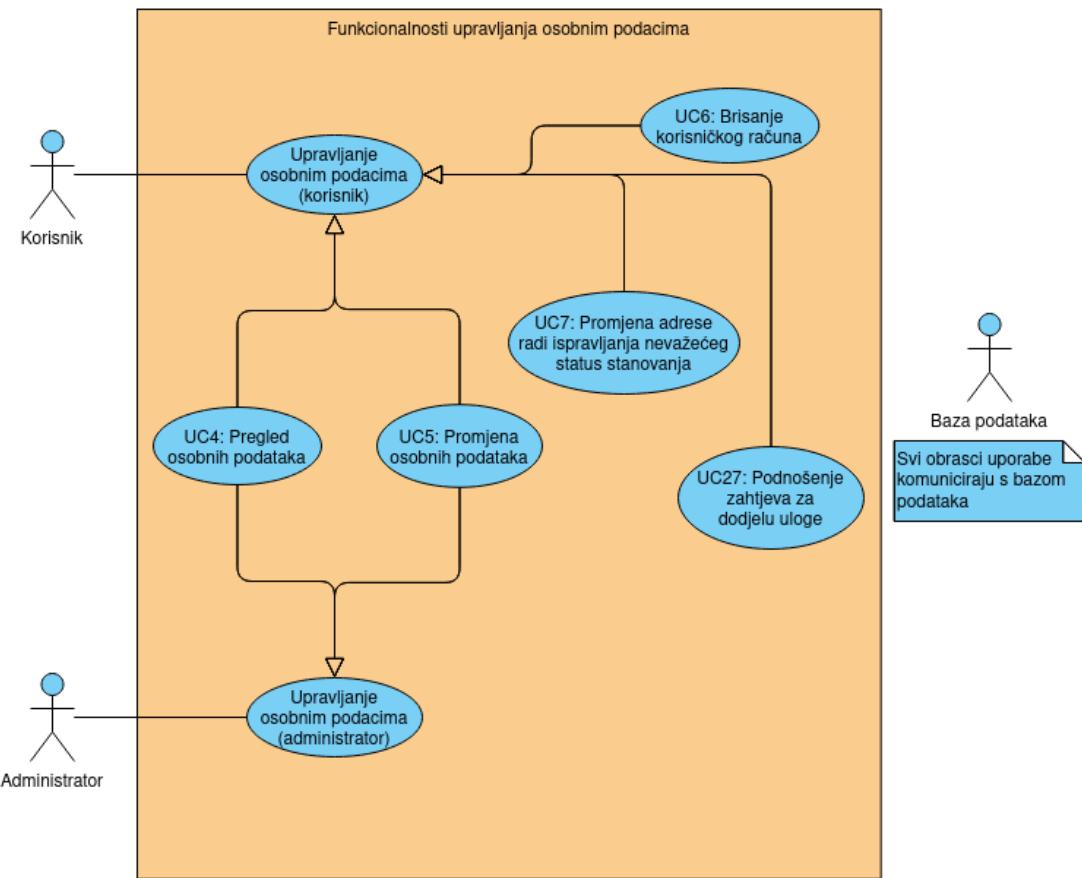
- 3.a Korisnik odustane od brisanja kvarta
 1. Korisnik biva preusmјeren na stranicu koja sadrži podatke o tom kvartu
- 4.a Postoje korisnici sustava čije adrese su zbog brisanja kvarta prestale biti važeće
 1. Sustav automatski postavi status stanovanja na "Nevažeći" svim korisnicima čije adrese su zbog brisanja kvarta prestale biti važeće
 2. Baza podataka se ažurira
 3. Korisnik biva preusmјeren na stranicu "Kvartovi"

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, osnovne funkcionalnosti

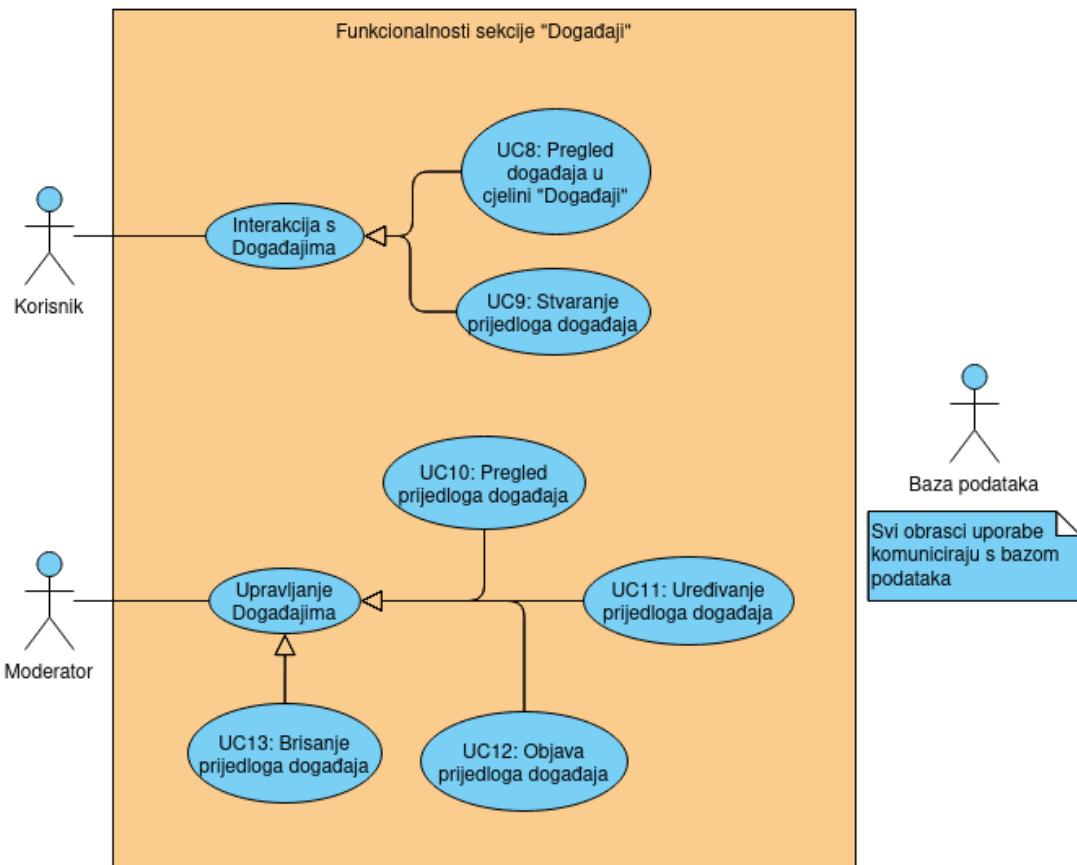
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 3.2: Dijagram obrazaca uporabe, funkcionalnosti upravljanja osobnim podacima

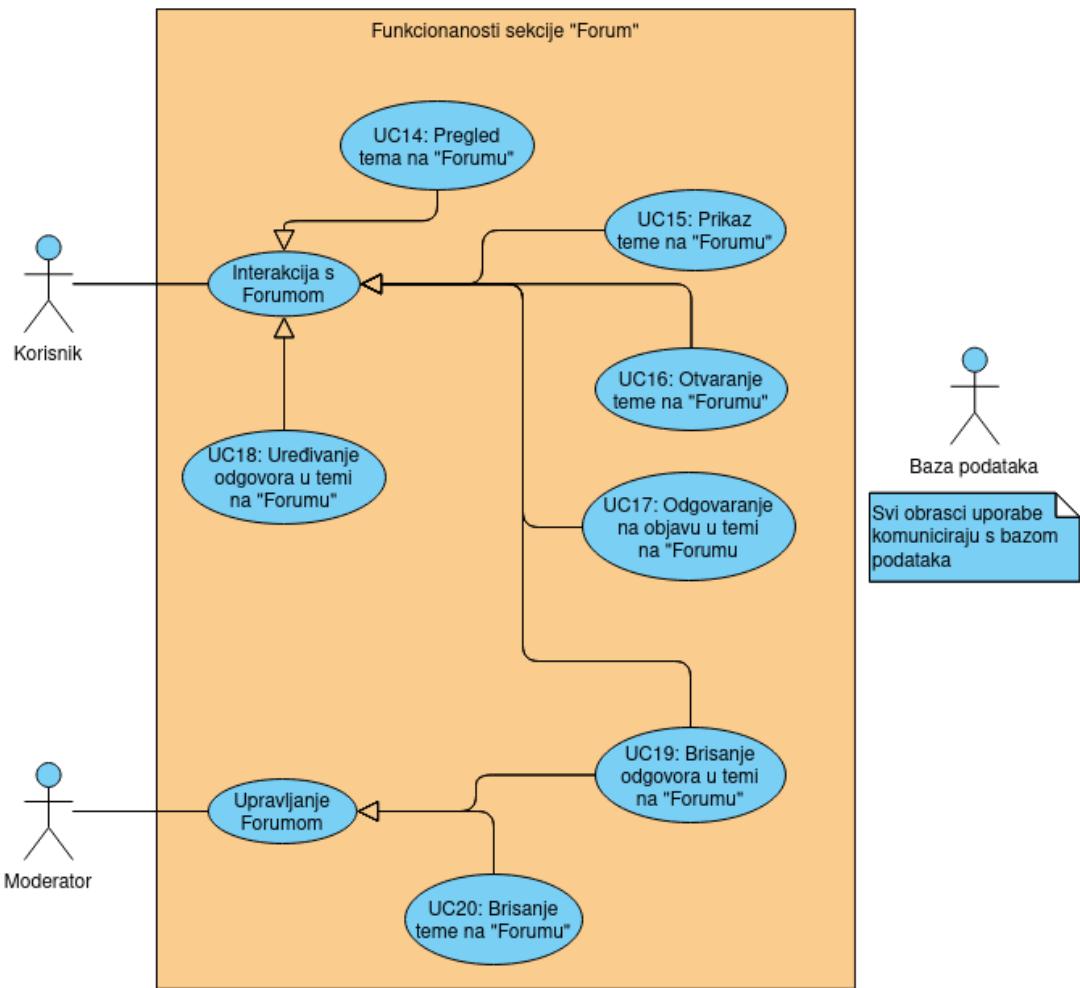
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 3.3: Dijagram obrazaca uporabe, funkcionalnosti sekcije "Dogadaji"

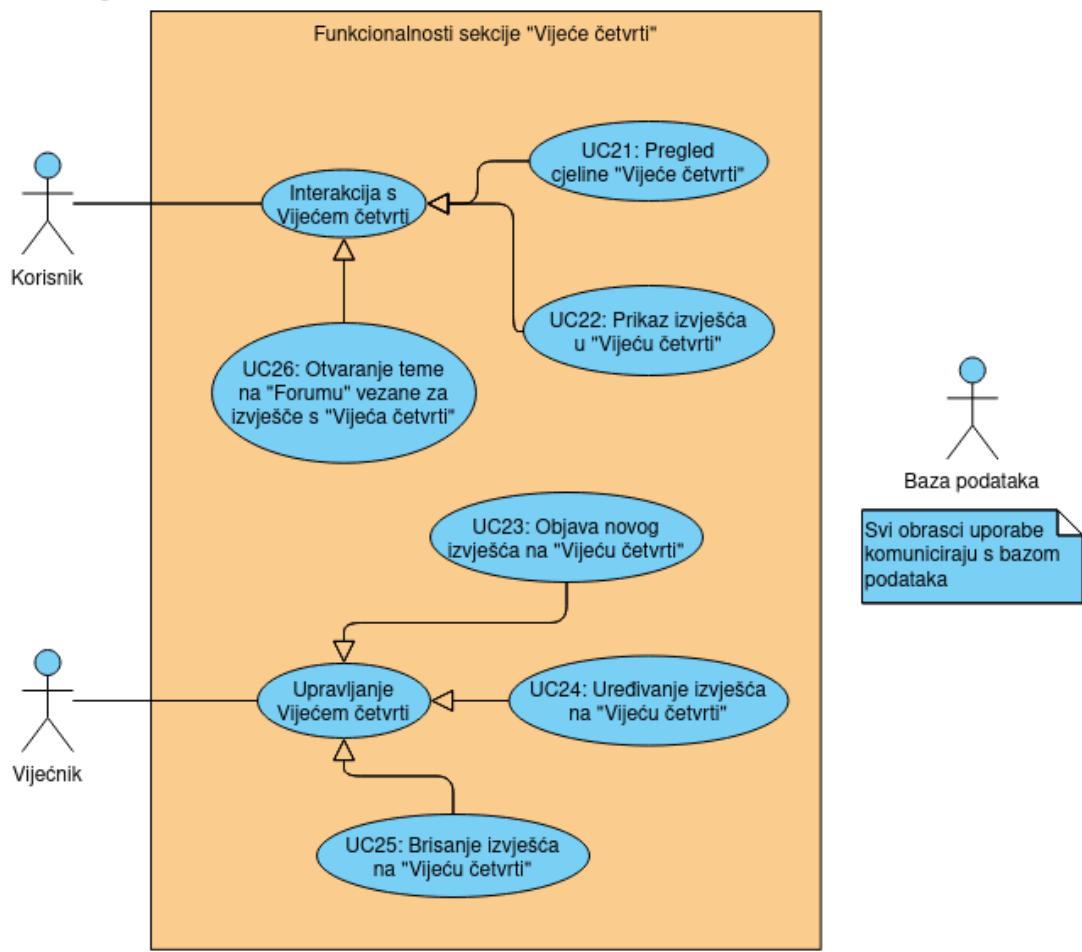
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 3.4: Dijagram obrazaca uporabe, funkcionalnosti sekcije "Forum"

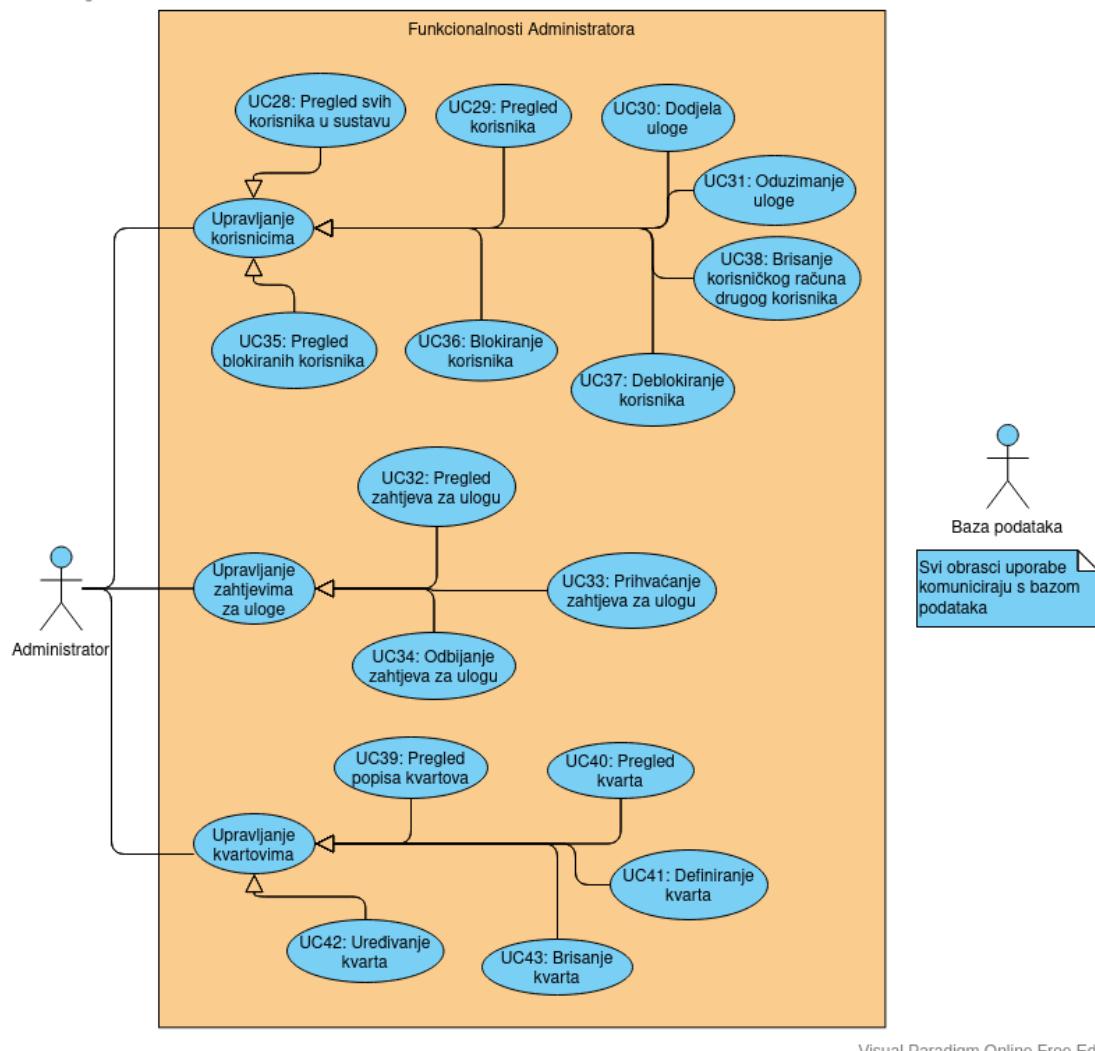
Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Slika 3.5: Dijagram obrazaca uporabe, funkcionalnosti sekcije "Vijeće četvrti"

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

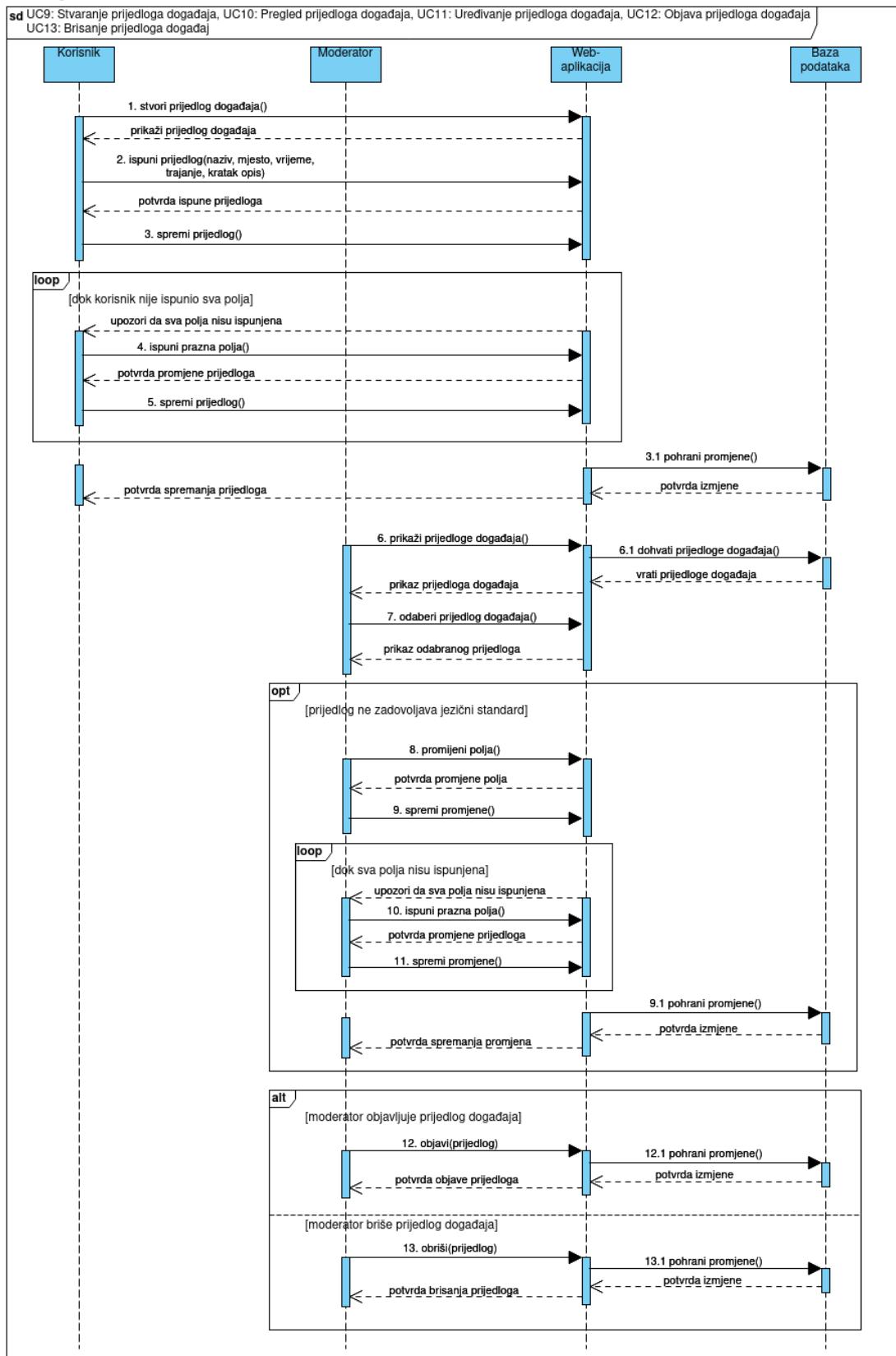
Slika 3.6: Dijagram obrazaca uporabe, funkcionalnosti Administratora

3.1.2 Sekvencijski dijagrami

Stvaranje, uređivanje, prihvatanje i brisanje prijedloga događaja - obrasci uporabe UC9, UC10, UC11, UC12, UC13

Korisnik šalje zahtjev za stvaranjem prijedloga događaja. Poslužitelj prikazuje obrazac korisniku sa poljima naziv, mjesto, vrijeme, trajanje, kratak opis. Nakon što korisnik ispuni polja, poslužitelj traži potvrdu ispunе prijedloga te korisnik potvrđuje spremanje prijedloga. Dok sva polja u obrascu nisu ispunjena poslužitelj upozorava korisnika. Ispunom praznih polja od strane korisnika, poslužitelj traži potvrdu promjene prijedloga te korisnik potvrđuje spremanje prijedloga poslužitelju. Baza podataka prima prijedlog od poslužitelja i pohranjuje promjene, vraća potvrdu izmjene poslužitelju, a poslužitelj potvrđuje spremanje prijedloga korisniku. Moderator započinje pregled prijedloga događaja, slanjem zahtjeva za prikaz prijedloga događaja poslužitelju. Poslužitelj dohvata prijedloge događaja iz baze podataka te ih prikazuje moderatoru. Moderator odabire prijedlog događaja te poslužitelj prikazuje odabrani prijedlog. Dok sva polja nisu ispunjena poslužitelj upozorava moderatora da polja nisu ispunjena, moderator ispunjava prazna polja, poslužitelj traži potvrdu promjene te moderator sprema promjene. Ako prijedlog ne zadovoljava jezični standard, moderator mijenja polja, poslužitelj traži potvrdu promjene te moderator šalje zahtjev za spremanjem promjena poslužitelju. Poslužitelj nakon provedenih provjera i izmjena pohranjuje promjene na bazi podataka i vraća potvrdu spremanja moderatoru. Moderator šalje zahtjev za objavu prijedloga, poslužitelj pohranjuje promjene na bazi podataka i vraća potvrdu objave. Ukoliko Moderator šalje zahtjev za brisanje prijedloga, poslužitelj pohranjuje promjene na bazi podataka i vraća potvrdu brisanja.

Visual Paradigm Online Free Edition

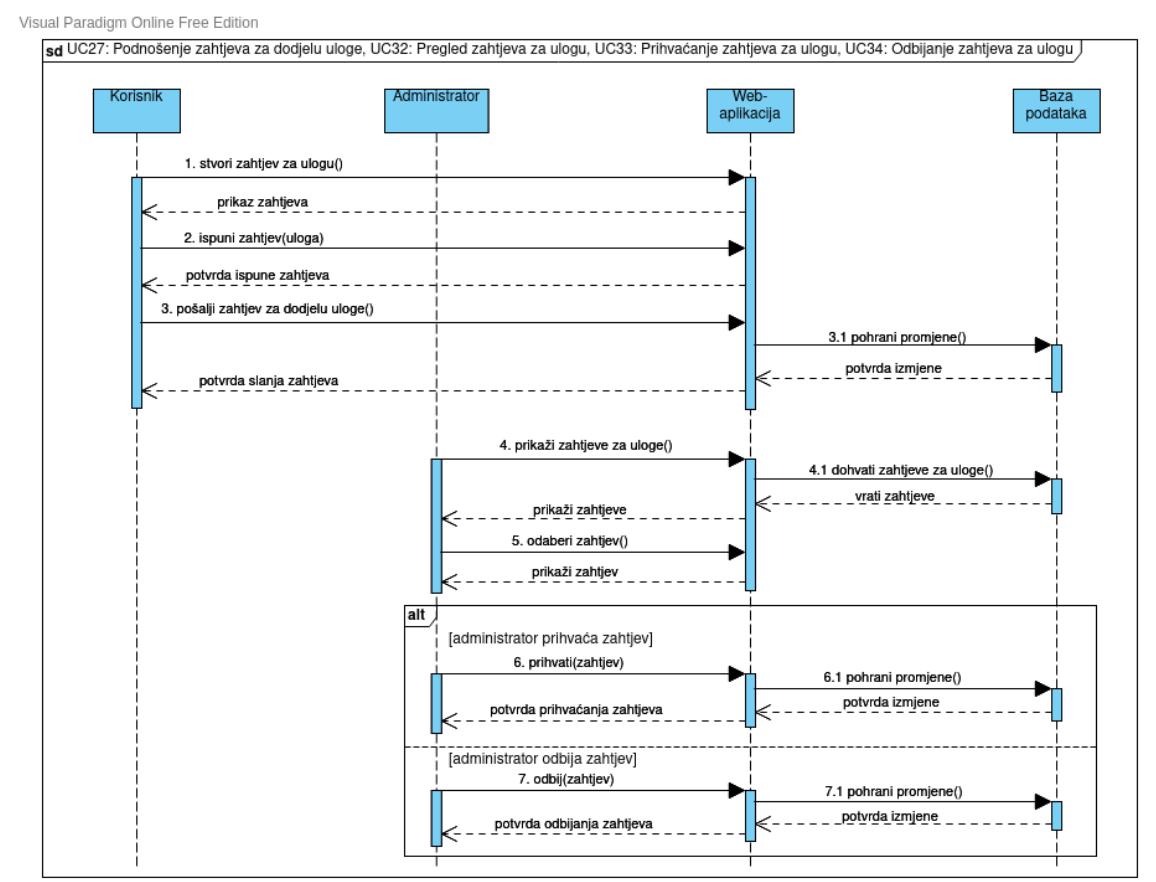


Visual Paradigm Online Free Edition

Slika 3.7: Sekvencijski dijagram za UC9, UC10, UC11, UC12, UC13

Stvaranje, prihvatanje i odbijanje zahtjeva za dodjelu uloge - obrasci uporabe UC27, UC32, UC33, UC34

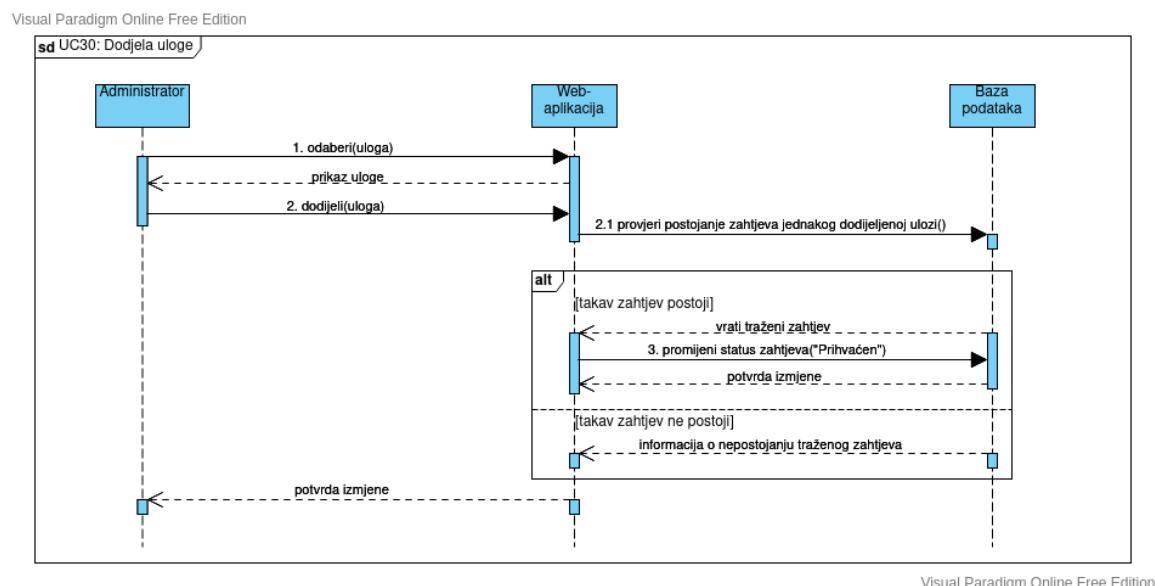
Korisnik šalje poslužitelju zahtjev za stvaranje uloge, poslužitelj vraća prikaz zahtjeva. Nakon ispune zahtjeva poslužitelj traži potvrdu ispune zahtjeva te korisnik šalje zahtjev za dodjelu uloge. Poslužitelj pohranjuje promjene na bazi podataka i nakon što baza potvrdi izmjene, šalje potvrdu slanja zahtjeva korisniku. Administrator šalje zahtjev za prikaz zahtjeva za uloge. Poslužitelj dohvaća zahtjeve iz baze podataka te ih prikazuje administratoru. Administrator odabire željeni zahtjev te ga poslužitelj prikazuje. Administrator prihvata odabrani zahtjev, poslužitelj pohranjuje promjene na bazi podataka i vraća potvrdu prihvata administratoru. Ukoliko administrator odbije zahtjev, poslužitelj pohranjuje promjene na bazi podataka te vraća potvrdu odbijanja zahtjeva administratoru.



Slika 3.8: Sekvencijski dijagram za UC27, UC32, UC33, UC34

Dodjela uloge - obrazac uporabe UC30

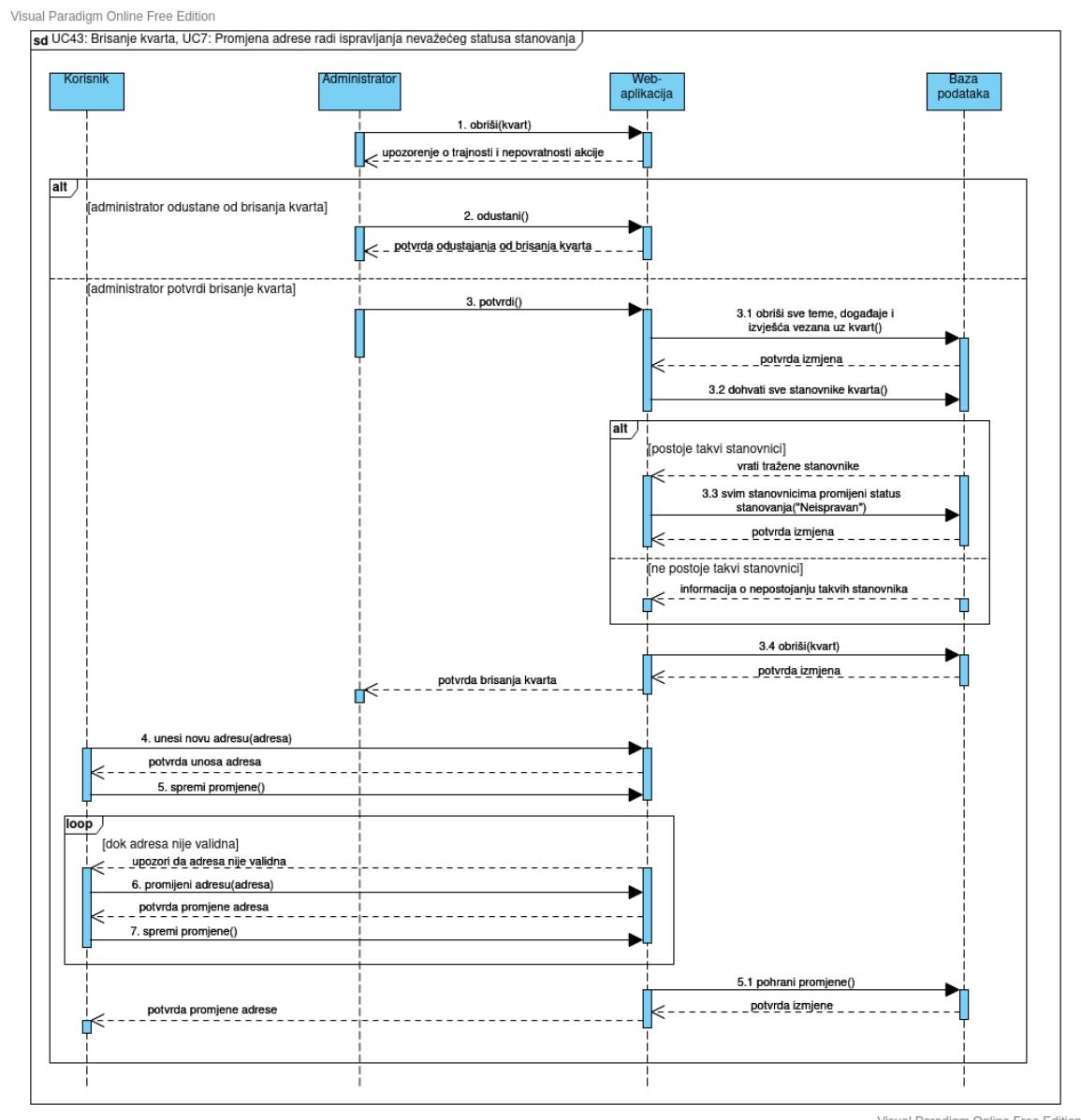
Administrator odabire željenu ulogu, poslužitelj vraća prikaz uloge te administrator šalje zahtjev za dodijelu uloge. Poslužitelj provjerava postojanje zahtjeva jednakog dodijeljenoj ulozi u bazi podataka. Ako zahtjev postoji baza podataka vraća traženi zahtjev, poslužitelj mijenja status zahtjeva u "prihvачen" te baza podataka vraća potvrdu izmjene. Ukoliko takav zahtjev ne postoji baza podataka vraća informaciju o nepostojanju traženog zahtjeva. Poslužitelj vraća administratoru potvrdu izmjene.



Slika 3.9: Sekvencijski dijagram za UC30

Brisanje kvarta i promjena nevažećih adresa - obrasci uporabe UC43, UC7

Administrator šalje zahtjev poslužitelju za brisanjem kvarta. Poslužitelj odgovara sa upozorenjem o trajnosti i nepovratnosti odabrane akcije. Ako administrator odabere odustati od brisanja kvarta, šalje zahtjev za odustajanju poslužitelju te poslužitelj odgovara s potvrdom odustajanja od brisanja kvarta. Ukoliko administrator odabere opciju potvrdi, šalje potvrdu za brisanjem poslužitelju. Poslužitelj šalje zahtjev za brisanjem svih tema, događaja i izvješća vezanih uz kvart na bazi podataka, baza vraća potvrdu izmjene te poslužitelj šalje zahtjev za dohvat svih stanovnika obrisanog kvarta. Ako postoje takvi stanovnici, baza vraća tražene stanovnike, poslužitelj mijenja svim stanovnicima status stanovanja u neispravan, baza podataka šalje potvrdu izmjene poslužitelju. Ukoliko ne postoje takvi stanovnici, baza podataka šalje poslužitelju informaciju o nepostojanju takvih stanovnika. Poslužitelj briše kvart, a baza podataka vraća potvrdu izmjene koju poslužitelj prosljeđuje do administratora. Korisnik koji ispravlja status nevažećeg statusa stanovanja unosi novu adresu, poslužitelj šalje potvrdu te korisnik potvrđuje promjene. Dok unesena adresa nije validna poslužitelj upozorava korisnika da adresa nije valjana. Nakon što korisnik promijeni adresu, poslužitelj šalje potvrdu promjene adrese te korisnik spremi promjene. Poslužitelj pohranjuje promjene na bazi podataka, te nakon primitka potvrde izmjene, šalje potvrdu promjene adrese korisniku.



Slika 3.10: Sekvencijski dijagram za UC43, UC7

3.2 Ostali zahtjevi

- Sustav treba koristiti hrvatski jezik te podržava hrvatsku abecedu
- Sustav treba biti sposoban izvršavati sve zadatke u kratkom vremenu od najviše nekoliko sekundi
- Sustav treba biti izrađen korištenjem objektno orijentirane paradigme u obliku web-aplikacije
- Sustavu se pristupa iz javne mreže pomoću protokola HTTPS
- Sustav omogućava istovremeni rad više korisnika
- Korisnikovo neispravno korištenje ne smije poremetiti rad sustava
- Sustav mora biti jednostavan za korištenje tako da ga je moguće koristiti bez uputa
- Rad na sustavu ne smije narušavati funkcionalnosti sustava
- Svi privatni podaci na sustavu su zaštićeni

4. Arhitektura i dizajn sustava

Arhitektura sustava se sastoji od tri glavna podsustava, a to su web preglednik, web poslužitelj i baza podataka.

- **Web preglednik** je program s pomoću kojeg korisnik pristupa sustavu, odnosno sva korisnička interakcija se odvija preko web-preglednika. Korisnik putem web preglednika šalje zahtjeve za resursima, koje web preglednik dohvaća od web poslužitelja, i onda se ti resursi na ispravan način interpretiraju i prikazuju. Korisnik također može i slati podatke preko web aplikacije, najčešće korištenjem formi.
- **Web poslužitelj** je centralni dio web aplikacije. Zasniva se na protokolu HTTP, komunicira i s korisnikom i s bazom podataka, te na zahtjeve korisnika dohvaća resurse ili obrađuje podatke poslane korištenjem forme i ažurira bazu podataka.
- **Baza podataka** služi za pohranu podataka sustava. Gotovo svi scenariji korištenja web aplikacije podrazumijevaju dohvaćanje podataka iz baze, i spremanje podataka u bazu.

Aplikacija je izgrađena u stilu arhitekture zasnovane na događajima, i shodno tome, aplikacija se temelji na MVC konceptu. MVC se sastoji od tri komponente:

- **Model** sadrži razrede čiji objekti se obrađuju.
- **View** (hrv. pogled) sadrži razrede čiji objekti služe za prikaz podataka.
- **Controller** (hrv. nadglednik) sadrži razrede koji upravljaju i rukuju korisničkom interakcijom s pogledom i modelom.

Aplikacija je izgradena korištenjem objektno orijentirane paradigme. Za backend je korišten radni okvir Java Spring. Za frontend je korištena biblioteka React.

4.1 Baza podataka

Sustav koristi relacijsku bazu podataka, koja je odabrana jer su podaci kojima web aplikacija barata vrlo povezani, i zbog toga ima najviše smisla koristiti relacije. Relacija, odnosno tablica, je srž i glavna komponenta baze koja se sastoji od imena i skupa atributa. Baza podataka mora biti brza i efikasna u svojoj zadaći, to jest u pohranjivanju, dohvaćanju i izmjeni podataka.

Baza podataka sastoji se od deset entiteta:

- Account
- Event
- Thread
- Post
- District
- Street
- Home
- Role
- RoleRequest
- Meeting

4.1.1 Opis tablica

Account Entitet Account sadržava informacije o korisniku. Atributi koje sadrži entitet su: Identifikacijski ključ korisnika, email, ime, prezime, password, "isBlocked" što pokazuje da li je korisnik blokiran, "isValidAddress" što pokazuje ima li korisnik ispravnu adresu na kojoj stanuje, šifru kuće u kojoj stanuje, šifru sastanaka ako je korisnik vijećnik te je prisustvovao sastanku vijeća te šifra četvrte kojoj korisnik pripada. Ovaj entitet u vezi je One-to-Many s entitetom Post preko atributa IDAccount korisnika, u vezi Many-to-One s Meeting preko IDMeeting, u vezi One-to-Many s entitetom Event preko atributa IDAccount. Također je u vezi One-to-Many s RoleRequest preko IDAccount, u vezi Many-to-One s entitetom District preko atributa IDDistrict. U vezi je Many-to-One s entitetom Home preko IDHome te napisljeku je u vezi Many-to-Many s Role preko atributa IDAccount.

Account		
IDAccount	INT	Identifikacijski ključ korisnika
email	VARCHAR	Email korisnika
firstName	VARCHAR	Ime korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Account		
lastName	VARCHAR	Prezime korisnika
password	VARCHAR	Lozinka korisnika
isBlocked	BOOLEAN	Oznaka je li korisnik blokiran
isAddressValid	BOOLEAN	Oznaka je li adresa korisnika valjana
IDHome	INT	Identifikacijski ključ korisnikove kuće
IDMeeting	INT	Identifikacijski ključ sastanka
IDDistrict	INT	Identifikacijski ključ četvrti kojoj korisnik pripada

Event Ovaj entitet sadržava sve informacije vezane uz događaj. Atributi koje sadrži entitet su: Identifikacijski ključ događaja, naziv događaja, opis, trajanje, vrijeme, lokaciju te status događaja. Entitet je u samo jednoj i to Many-to-One vezi s entitetom Account preko atributa IDAccount.

Event		
IDEVENT	INT	Identifikacijski ključ događaja
eventName	VARCHAR	Naziv događaja
eventDatetime	TIMESTAMP	Vrijeme događaja
eventDate	DATE	Datum događaja
eventTime	TIME	Vrijeme početka događaja
eventLocation	VARCHAR	Lokacija događaja
eventDuration	INTERVAL	Trajanje događaja
eventDescription	VARCHAR	Opis događaja
eventStatus	VARCHAR	Status događaja
IDAccount	INT	Identifikacijski ključ korisnika

Thread Ovaj entitet sadržava sve informacije vezane uz dretvu na forumu. Sadrži atribute: Identifikacijski ključ dretve, ime dretve te šifru četvrte kako bi se

utvrdilo kojem forumu pripada ta dretva, odnosno četvrти. Ovaj entitet je u One-to-One vezi s entitetom Meeting preko atributa IDThread te je u vezi One-to-Many s District preko šifre četvrти.

Thread		
IDThread	INT	Identifikacijski ključ dretve
threadName	VARCHAR	Ime dretve
IDDistrict	INT	Identifikacijski ključ četvrти

Post Ovaj entitet sadržava sve informacije vezane uz objavu na forumu. Sadrži atribute: Identifikacijski ključ objave, vrijeme objave, sadržaj objave, šifra odgovora na objavu(ako postoji), šifra dretve kojoj objava pripada te šifra korisnika koji je objavio objavu. Ovaj entitet je u dvije Many-to-One veze i to s entitetima Account i Thread preko njihovih odgovarajućih identifikacijskih šifri(IDAccount i IDThread). Također je u refleksivnoj vezi zbog mogućnosti odgovora na objavu.

Post		
IDPost	INT	Identifikacijski ključ objave
postDatetime	TIMESTAMP	Vrijeme postavljanja objave
postContent	VARCHAR	Sadržaj objave
IdThread	INT	Identifikacijski ključ pripadajuće dretve
IDAccount	INT	Identifikacijski ključ korisnika koji je objavio objavu
IDPostReply	INT	Identifikacijski ključ odgovora na objavu ako postoji

District Ovaj entitet sadržava sve informacije vezane uz četvrt. Sadrži atribute: Identifikacijski ključ četvrti te naziv četvrti. Ovaj entitet je u tri One-to-Many veze i to s entitetima Account, Post i Street preko vlastite identifikacijske šifre(IDDistrict). Također je u One-to-Many vezi sa slabim entitetom Meeting preko atributa IDDistrict.

District		
IDDistrict	INT	Identifikacijski ključ četvrti
districtName	VARCHAR	Naziv četvrti

Street Ovaj entitet sadržava sve informacije vezane uz ulicu. Sadrži atribute: Identifikacijski ključ ulice, naziv ulice, najmanji kućanski broj u ulici, najveći kućanski broj u ulici te šifra četvrti u kojoj se ulica nalazi. Ovaj entitet je u Many-to-One vezi s entitetom District preko šifre četvrti. Također je u One-to-Many vezi s entitetom Home preko atributa IDStreet.

Street		
IDStreet	INT	Identifikacijski ključ ulice
streetName	VARCHAR	Naziv ulice
minStreetNumber	INT	Najmanji kućanski broj ulice
maxStreetNumber	INT	Najveći kućanski broj ulice
IDDistrict	INT	Identifikacijski ključ pripadajuće četvrti

Home Ovaj entitet sadržava sve informacije vezane uz kuću korisnika. Sadrži atribute: Identifikacijski ključ kuće, kućni broj te šifra ulice u kojoj se kuća nalazi. Ovaj entitet je u Many-to-One vezi s entitetom Street preko šifre ulice. Također je u One-to-Many vezi s entitetom Account preko atributa IDAccount.

Home		
IDHome	INT	Identifikacijski ključ kuće
homeNumber	INT	Kućanski broj
IDStreet	INT	Identifikacijski ključ pripadajuće ulice

Role Ovaj entitet sadržava sve informacije vezane uz ulogu korisnika. Sadrži atribute: Identifikacijski ključ uloge te naziv uloge. Ovaj entitet je u Many-to-Many vezi s entitetom Account preko šifre uloge te u One-to-Many vezi s entitetom RoleRequest preko šifre uloge.

Role		
IDRole	INT	Identifikacijski ključ uloge
roleName	VARCHAR	Naziv uloge

RoleRequest Ovaj entitet sadržava sve informacije vezane uz zahtjev korisnika za ulogom. Sadrži atribute: Identifikacijski ključ zahtjeva, status zahtjeva te šifra korisnika koji je podnio zahtjev i šifra uloge koju korisnik zahtjeva. Ovaj entitet je u Many-to-One vezi s entitetom Role preko šifre uloge. Također je u Many-to-One vezi s entitetom Account preko atributa IDAccount.

RoleRequest		
IDRoleRequest	INT	Identifikacijski ključ zahtjeva za ulogu
roleRequestStatus	VARCHAR	Status zahtjeva za ulogu
IDAccount	INT	Identifikacijski ključ korisnika
IDRole	INT	Identifikacijski ključ uloge

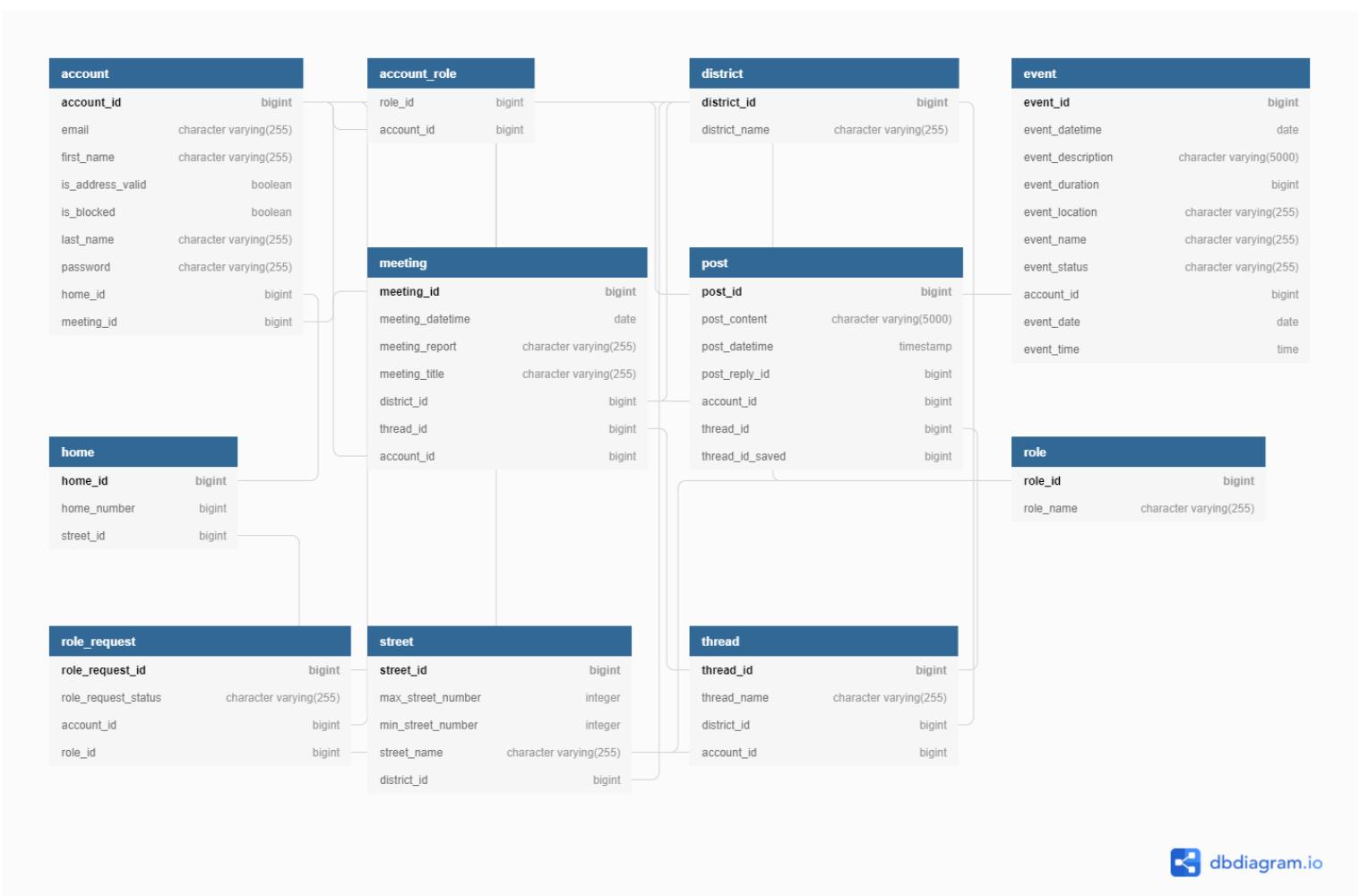
AccountRole Ova join tablica koja je posljedično nastala spajanjem entiteta Account s entitetom Role vezom Many-to-Many sadržava informacije o ulozi korisnika. Sadrži 2 atribute koja su ključ entiteta Account i ključ entiteta Role.

AccountRole		
IDAccount	INT	Identifikacijski ključ korisnika
IDRole	INT	Identifikacijski ključ uloge

Meeting Ovaj slabi entitet sadržava sve informacije vezane uz sastanak vijeća. Sadrži atribute: Identifikacijski ključ sastanka, naziv sastanka, vrijeme sastanka, izvještaj te šifru autora izvještaja. Također sadrži šifru četvrte te šifru dretve na forumu ukoliko je otvorena na temu izvještaja. Ovaj entitet je u One-to-One vezi s entitetom Thread preko šifre dretve. Također je u Many-to-One vezi s entitetom District preko šifre četvrte te u vezi One-to-Many s entitetom Account preko šifre sastanka.

Meeting		
IDMeeting	INT	Identifikacijski ključ sastanka
IDDistrict	INT	Identifikacijski ključ četvrti
IDThread	INT	Identifikacijski ključ dretve
meetingReport	VARCHAR	Izvješće sastanka
reportAuthorId	INT	Identifikacijski broj korisnika koji je sastavio izvješće
meetingTitle	VARCHAR	Naziv sastanka
meetingDatetime	TIMESTAMP	Vrijeme sastanka

4.1.2 Dijagram baze podataka



Slika 4.1: Relacijska shema baze podataka

4.2 Dijagram razreda

Backend aplikacije je ostvaren korištenjem radnog okvira Spring Boot, i stoga aplikacija ima nekoliko slojeva.

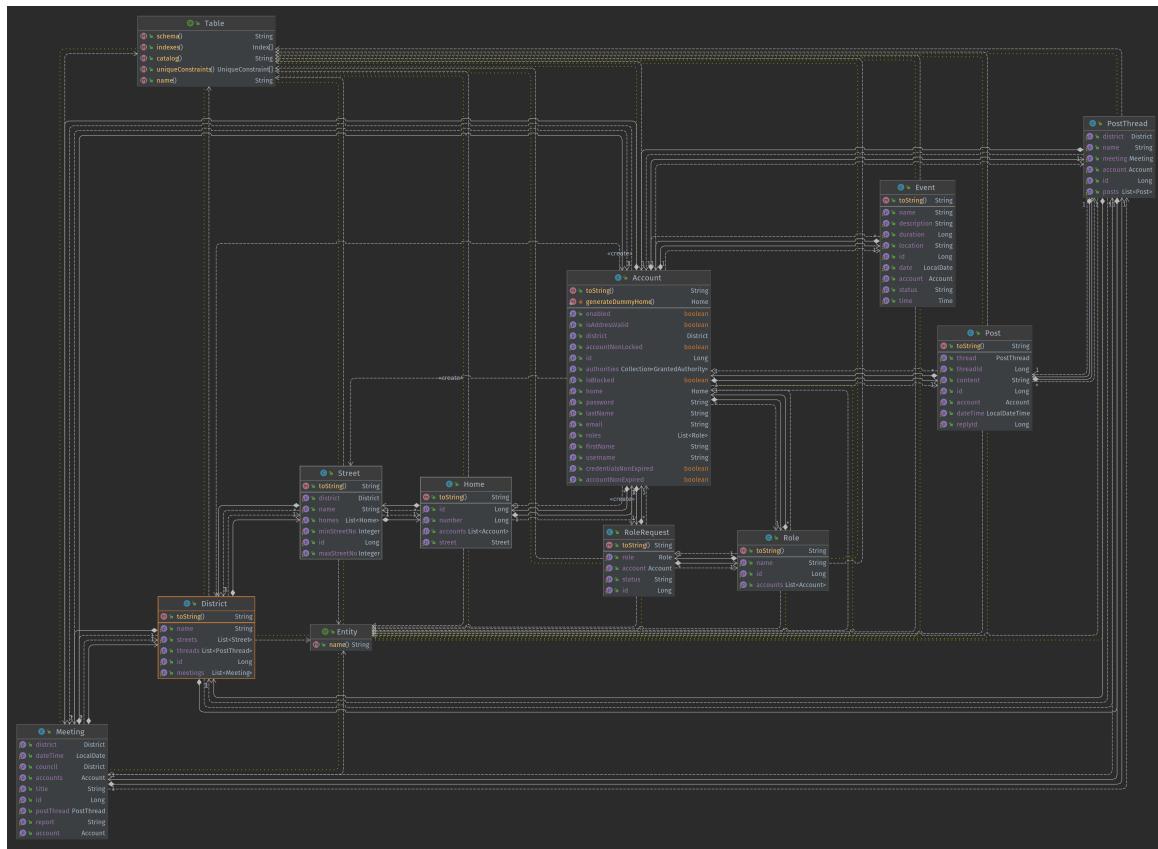
Iako se ne smatraju posebnim slojem, na slici 4.2 prikazani su entiteti koji odgovaraju relacijama u bazi podataka, opisanoj u poglavlju 4.1. Entiteti ne sadrže nikakvu proceduralnu logiku, nego isključivo članske varijable i njihove gettere i setttere. Entiteti u našoj aplikaciji su redom: Account, District, Event, Home, Meeting, Post, Role, RoleRequest, Street i PostThread.

Na slici 4.3 prikazan je sloj Repository. Razredi u njemu se definiraju kao sučelja koja nude metode dohvaćanja elemenata iz baze, te stvaranja promjena u bazi. Njihov kod u pravilu programer ne piše eksplisitno, nego te metode generira Spring na temelju njihovog imena.

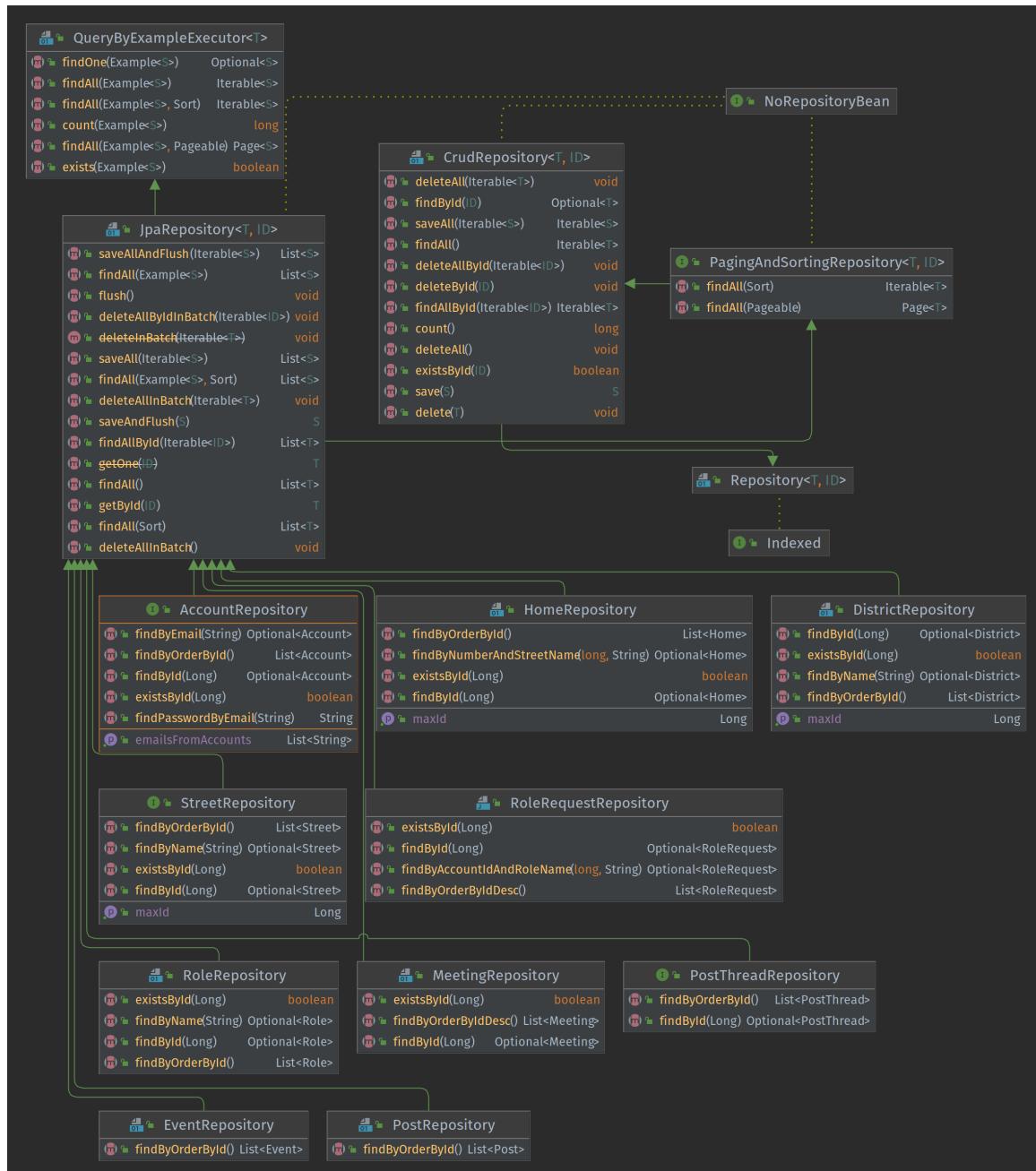
Na slici 4.4 prikazan je sloj Service. Taj sloj sadrži centralnu logiku na backendu, manipulira entitetima i poziva metode koje nudi prethodni sloj, Repository.

Na slici 4.5 prikazan je sloj Controller. Taj je sloj najniži u hijerarhiji slojeva, i on je jedini sloj s kojim komunicira frontend dio aplikacije. Uloga ovog sloja je da obrađuje HTTP zahtjeve, poziva metode koje nudi prethodni sloj Service, i potom šalje odgovore na primljene HTTP zahtjeve.

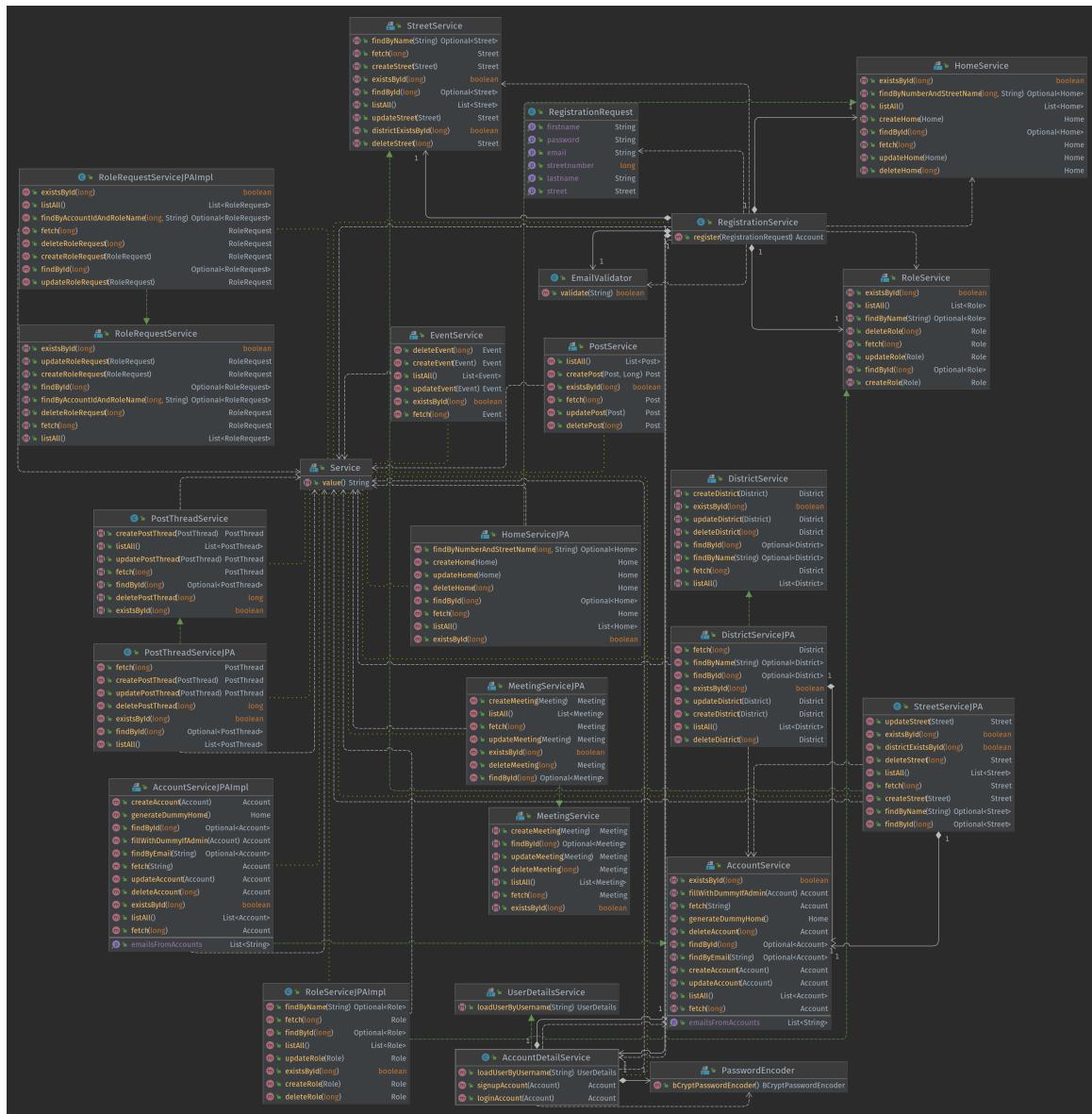
Zbog preglednosti, na jednom dijagramu nije bilo moguće prikazati sve razrede. Umjesto toga, na sljedeći način su prikazani odnosi među slojevima: na slici 4.6 prikazan je odnos između slojeva Controller i Service, na slici 4.7 prikazan je odnos između slojeva Service i Repository, na slici 4.8 prikazan je odnos između sloja Service i entiteta, te je na slici 4.9 prikazan odnos između sloja Repository i entiteta.



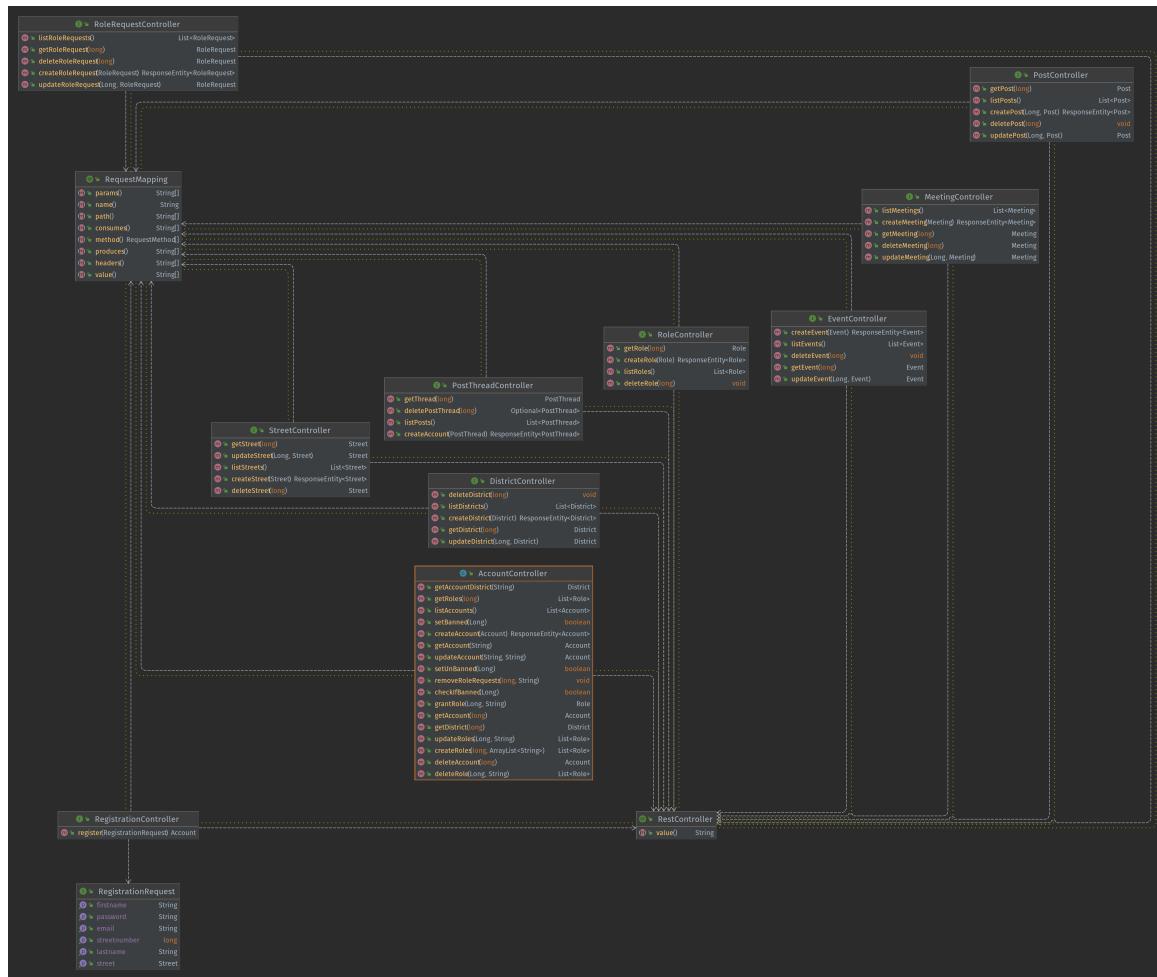
Slika 4.2: Dijagram razreda - entiteti



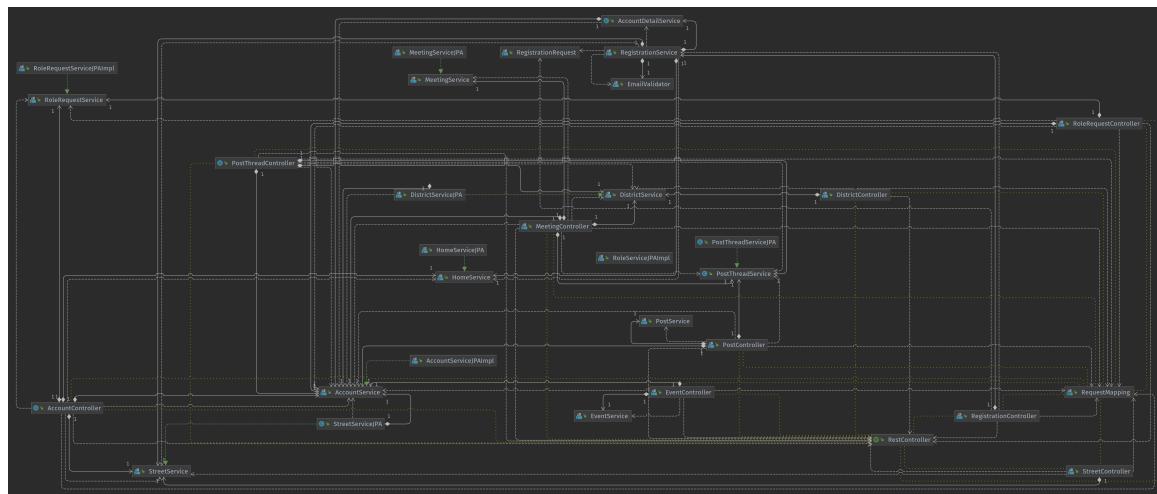
Slika 4.3: Dijagram razreda - sloj Repository



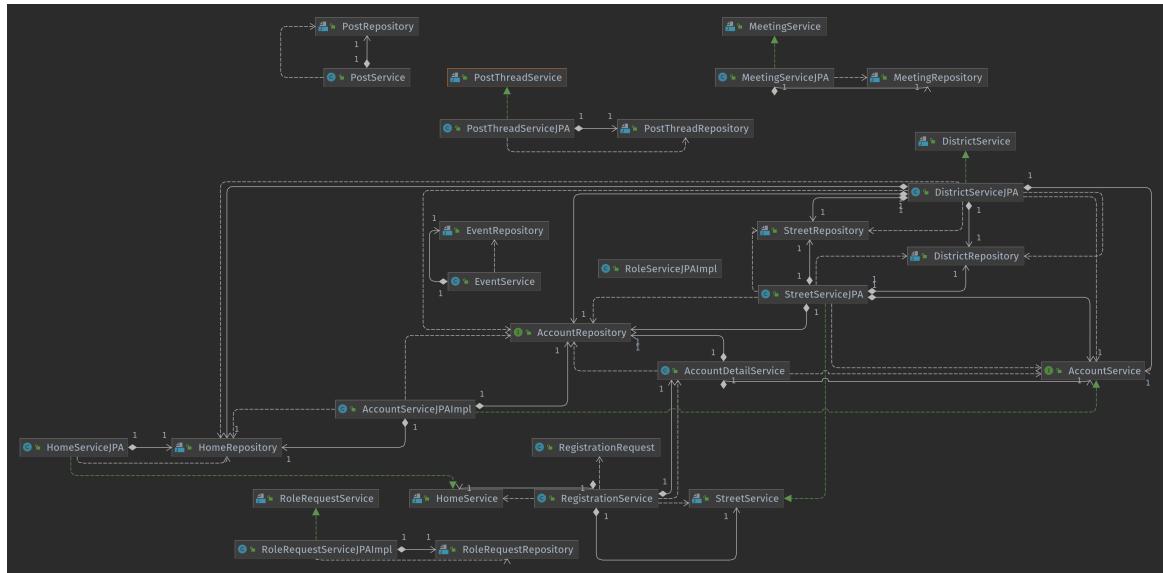
Slika 4.4: Dijagram razreda - sloj Service



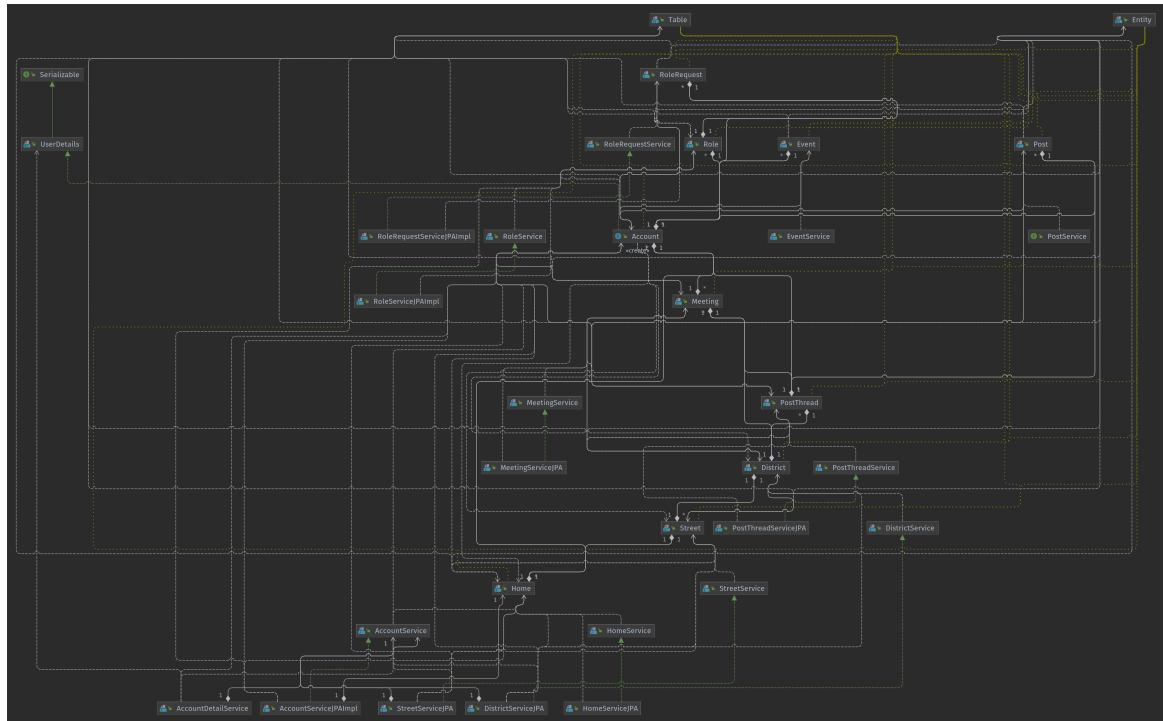
Slika 4.5: Dijagram razreda - sloj Controller



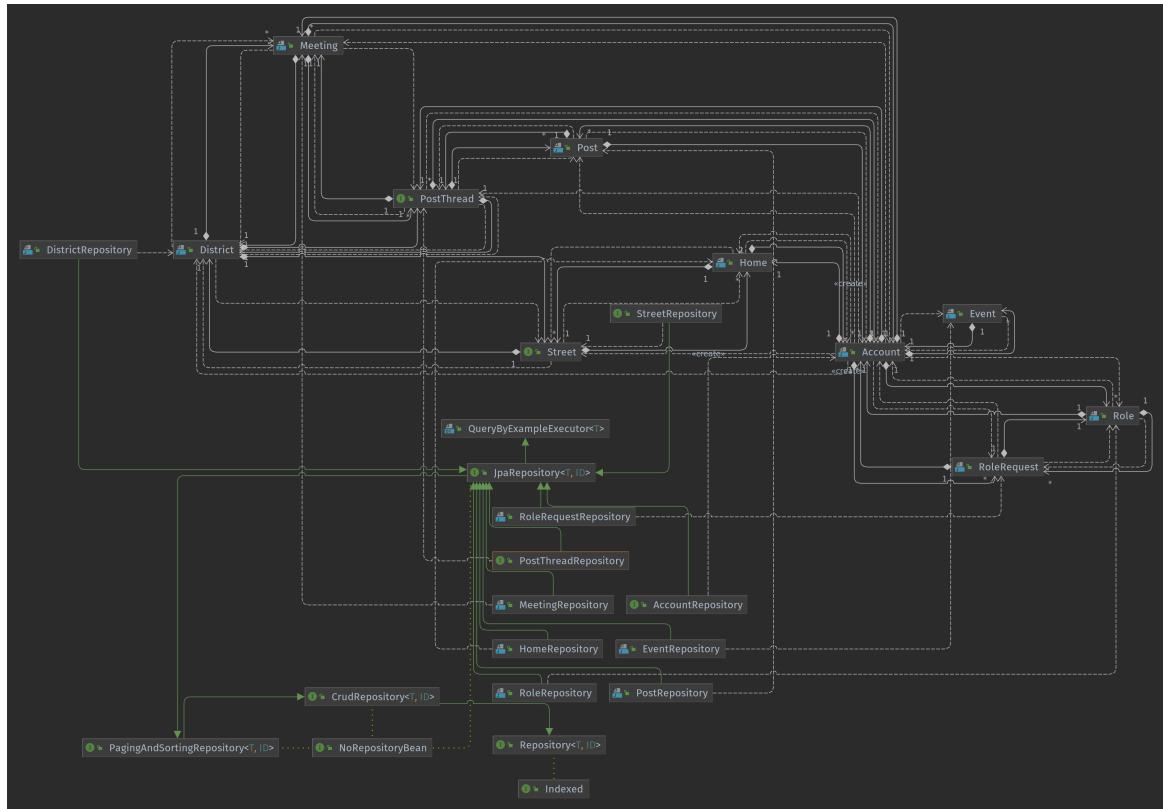
Slika 4.6: Dijagram razreda - odnos slojeva Controller i Service



Slika 4.7: Dijagram razreda - odnos slojeva Service i Repository



Slika 4.8: Dijagram razreda - odnos sloja Service i entiteta



Slika 4.9: Dijagram razreda - odnos sloja Repository i entiteta

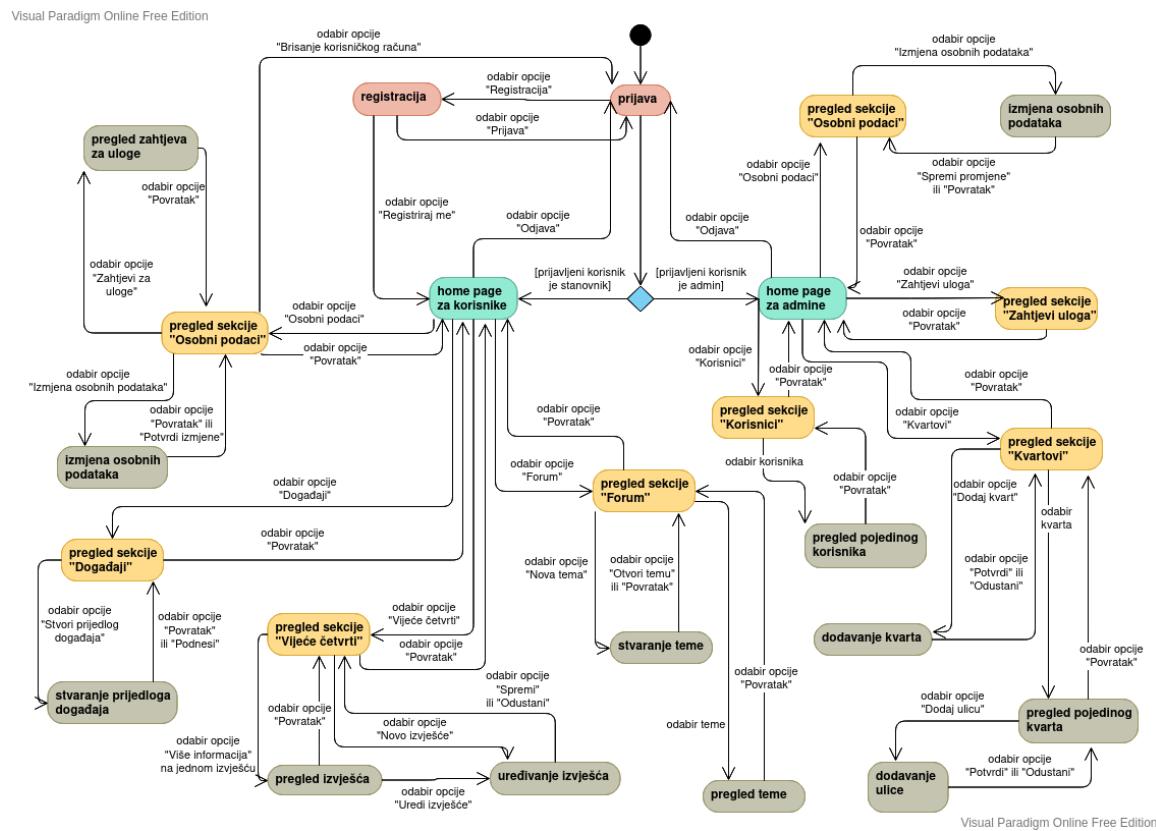
4.3 Dijagram stanja

Na slici 4.10 prikazan je dijagram stanja korisničkog sučelja. Stranica na koju korisnik uvijek prvo dođe je "Prijava". S te stranice korisnik može birati opciju "Registracija" kako bi stvorio novi račun, ili se može prijaviti s postojećim računom. Ovisno o tome je li korisnik stanovnik ili administrator, dočeka ga prikladna početna stranica.

Ako je korisnik stanovnik, sa svoje početne stranice može birati opcije "Osobni podaci", "Događaji", "Vijeće četvrti" i "Forum". Pri pregledu svojih osobnih podataka, korisnik može slati zahtjeve za dodatne uloge ili mijenjati svoje osobne podatke. Pri pregledu sekcije "Događaji", korisnik može stvarati prijedloge događaja. Pri pregledu sekcije "Vijeće četvrti", korisnik klikom na pojedino izvješće može dobiti dodatne informacije o tom izvješću, a ako ima ulogu "Vijećnik", onda može i stvarati nova izvješća. Konačno, pregledom sekcije "Forum" korisnik može otvarati nove teme i pregledavati postojeće.

Ako je korisnik administrator, sa svoje početne stranice može birati opcije "Osobni podaci", "Zahtjevi za uloge", "Korisnici" i "Kvartovi". Prilikom pregleda svojih osobnih podataka, administrator može neke od tih podataka može mijenjati. Prilikom pregleda zahtjeva za uloge administrator može odabrati pojedini zahtjev te ga prihvati ili odbiti. Prilikom pregleda svih korisnika sustava, administrator može odabrati pojedinog korisnika i dobiti više informacija o njemu. Konačno, pri pregledu kvartova, administrator može dodati novi kvart, a može i za postojeći kvart odabrati opciju unosa nove ulice.

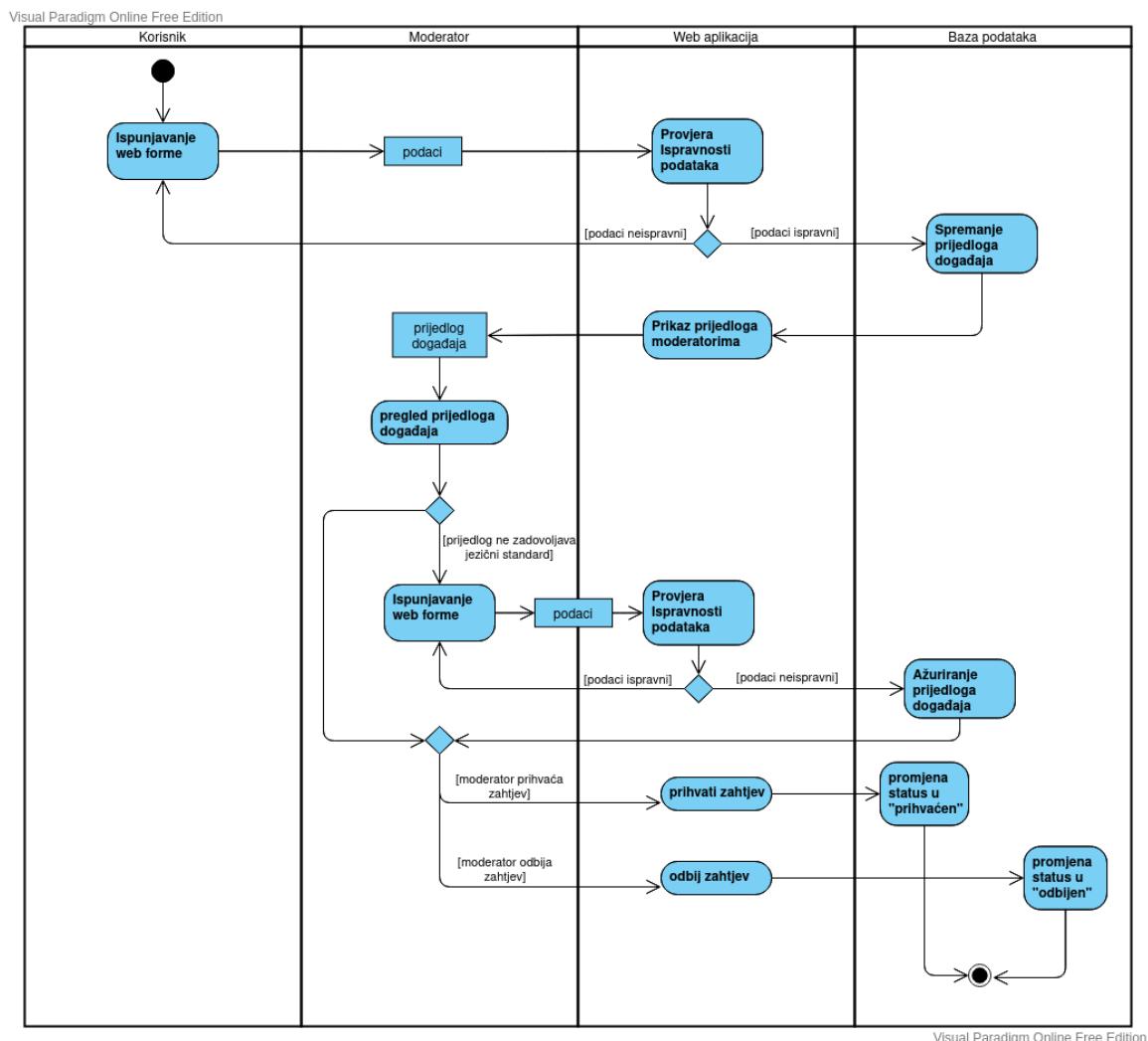
Iako zbog preglednosti to nije navedeno na dijagramu, stanovnici iz svih stanja mogu doći u stanja "Početna stranica", "Događaji", "Vijeće četvrti", "Forum" i "Osobni podaci", te administratori analogno mogu iz svih stanja doći u stanja "Početna stranica", "Korisnici", "Kvartovi", "Zahtjevi uloga" i "Osobni podaci". Također, svi korisnici iz svih stanja mogu birati opciju "Odjava" i preći u stanje "Prijava".



Slika 4.10: Dijagram stanja

4.4 Dijagram aktivnosti

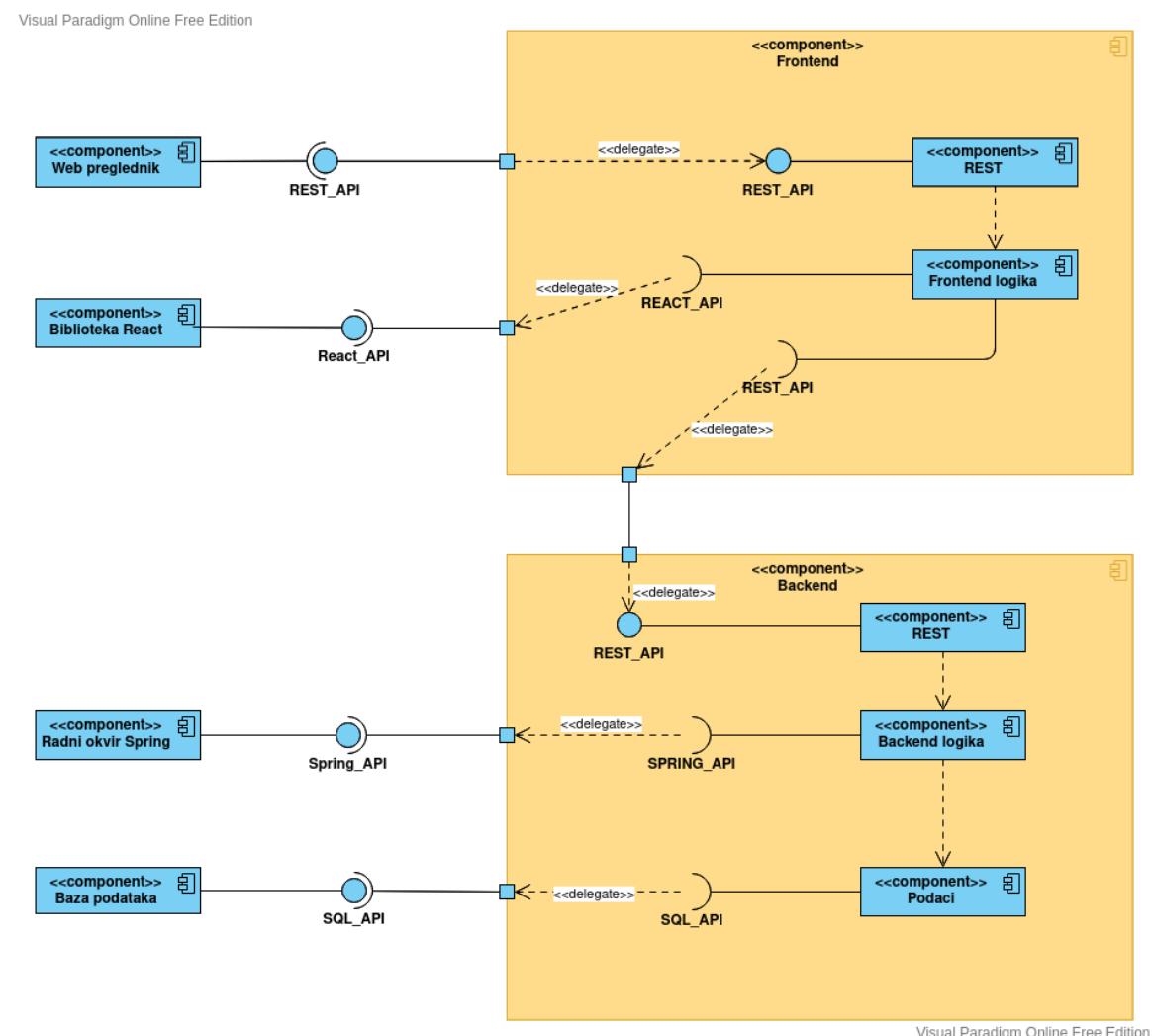
Na slici 4.11 prikazan je dijagram aktivnosti prilikom stvaranja novih događaja. Korisnik koji želi predložiti događaj ispunjava formu u koju unosi naziv, mjesto, vrijeme, trajanje i kratki opis. Kada ju je ispunio, odabire opciju spremanja prijedloga. Web aplikacija tada provjerava jesu li svi traženi podaci u ispravnom formatu. Ako nisu, upozorava korisnika na pogreške i omogućuje mu da ih ispravi, a ako su podaci ispravni, onda spremi prijedlog u bazu podataka. Nakon što je prijedlog spremišten, moderator ga može pregledati. Ako moderator procijeni da prijedlog ne zadovoljava jezični standard, može ga uređivati. Kada je ispravio sve eventualne pogreške, moderator odabire opciju spremanja promjena. Tada web aplikacija provjerava jesu li svi podaci u ispravnom formatu, te ako jesu, spremi promjene, a ako nisu, upozorava moderatora na pogreške i omogućuje mu da ih ispravi. Nakon što je završio s eventualnim uređivanjem prijedloga, moderator može odabrati opciju prihvatanja ili odbijanja prijedloga, i u oba slučaja se ažurira status prijedloga događaja u bazi podataka.



Slika 4.11: Dijagram aktivnosti

4.5 Dijagram komponenti

Na slici 4.12 prikazan je dijagram komponenti. Korisnik iz web preglednika pristupa aplikaciji korištenjem REST API-ja. Sama aplikacija se sastoji od dvije komponente. Prva komponenta odgovara frontendu i izgrađena je korištenjem React biblioteke. Druga komponenta odgovara backendu i izgrađena je korištenjem radnog okvira Spring Boot. Frontend i backend komuniciraju korištenjem REST API-ja. Baza podataka je relacijska i backend joj pristupa slanjem SQL upita.



Slika 4.12: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu je ostvarena korištenjem aplikacije Discord¹. Komunikacija s asistenticom grupе je ostvarena korištenjem aplikacije Microsoft Teams². Za izradu UML dijagrama korišten je alat Visual Paradigm Online³. Za upravljanje izvornim kodom korišten je alat Git⁴. Udaljeni repozitorij projekta nalazi se na platformi Gitlab⁵. Za upravljanje zadacima na projektu korištena je aplikacija Trello⁶.

Kao razvojno okruženje na backendu je koršten IntelliJ⁷, a na frontendu je korišten Visual Studio Code⁸. Tehnologije korištene za razvoj backenda su radni okvir Spring Boot⁹ i programski jezik Java¹⁰. Tehnologije korištene za razvoj frontenda su biblioteka React¹¹ i programski jezik JavaScript¹². Za automatizirano testiranje korišten je alat Selenium WebDriver¹³ te programski jezici Java i Python¹⁴. Za puštanje aplikacije u pogon korištena je platforma Heroku¹⁵.

¹<https://discord.com/>

²<https://www.microsoft.com/hr-hr/microsoft-teams/group-chat-software/>

³<https://online.visual-paradigm.com/>

⁴<https://git-scm.com/>

⁵<https://gitlab.com/>

⁶<https://trello.com/>

⁷<https://www.jetbrains.com/idea/>

⁸<https://code.visualstudio.com/>

⁹<https://spring.io/projects/spring-boot>

¹⁰<https://www.java.com/en/>

¹¹<https://reactjs.org/>

¹²<https://www.javascript.com/>

¹³<https://www.selenium.dev/documentation/webdriver/>

¹⁴<https://www.python.org/>

¹⁵<https://www.heroku.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Ispitivanje komponenti ostvareno je korištenjem Spring Boot i JUnit alata za ispitivanje. U nastavku su opisani provedeni testovi, te su priloženi kodovi.

Testovi 1-4 testiraju Account kontroler. Inicijalizacijska metoda za testove 1-4 izgleda ovako:

```
@BeforeEach  
public void init() {  
    Street street = new Street(1L, "Testna_ulica", 1, 5)  
        ;  
    street.setDistrict(new District(1L, "Testni_kvart"))  
        ;  
    home = new Home(1L, 1L, street);  
  
    account = new Account(1L, "John", "Doe", "  
        johndoe@gmail.com", "pass123",  
        home, null, false);  
    account.setRoles(new ArrayList<>());  
}
```

Testovi 5 i 6 testiraju Post kontroler. Inicijalizacijska metoda za testove 5 i 6 izgleda ovako:

```
@BeforeEach  
public void init() {  
    District district = new District(1L, "Testni_kvart")  
        ;  
    Street street = new Street(1L, "Testna_ulica", 1, 5)  
        ;  
    street.setDistrict(district);  
    home = new Home(1L, 1L, street);  
  
    account = new Account(1L, "John", "Doe", "  
        johndoe@gmail.com", "pass123",  
        home, null, false);
```

```
postThread = new PostThread(1L, "Example", new
    ArrayList<>(), null, district);

post = new Post(1L, "First-post",
    null,
    null, postThread, account);

post.setThreadId(postThread.getId());
}
```

U prvom ispitnom slučaju ispitani je pokušaj dobavljanja liste svih korisničkih računa. Očekivani rezultat je uspješno dobavljanje liste u JSON obliku te HTTP status 200 (OK).

```
@Test
public void
givenAccountsList_whenGetAccountsList_ThenReturnJsonArray
() throws Exception {
    List<Account> accounts = Collections.singletonList(
        account);

    given(accountService.listAll()).willReturn(accounts)
        ;

    mvc.perform(get("/accounts")
        .contentType(MediaType.APPLICATION_JSON).
        accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$", hasSize(1)))
        .andExpect(jsonPath("$.first_name").value(
            "John"));
}
```

U drugom ispitnom slučaju testiran je rubni slučaj u kojem ne postoji niti jedan korisnički račun, a lista korisničkih računa se pokušava dohvatiti. Očekivani rezultat je uspješno dobavljanje prazne liste u JSON obliku te HTTP status 200 (OK).

```
@Test
public void givenEmptyAccountsList_whenGetAccountsList_thenReturnEmptyJsonArray()
    throws Exception {
    List<Account> accounts = Collections.emptyList();

    given(accountService.listAll()).willReturn(accounts);

    mvc.perform(get("/accounts")
        .contentType(MediaType.APPLICATION_JSON)
        .accept(MediaType.APPLICATION_JSON))
        .andExpect(status().isOk())
        .andExpect(jsonPath("$.empty()"));

}
```

U trećem ispitnom slučaju ispitana je pokušaj izrade novog korisničkog računa, dok je pritom priložena email adresa koju neki postojeći korisnik već koristi. Očekivani rezultat je neuspješna izrada korisničkog računa te HTTP status 400 (Bad Request).

```
@Test
public void givenEmailList_whenCreateAccountWithExistingEmail_thenCauseError400()
    throws Exception {
    given(accountService.getEmailsFromAccounts())
        .willReturn(Collections.singletonList(
            "johndoe@gmail.com"));

    mvc.perform(post("/accounts")
        .content(new ObjectMapper()
            .writeValueAsString(new Account("Johnny", "Doe",
                "johndoe@gmail.com", "pass123", home, null,
                false)))
        .contentType(MediaType.APPLICATION_JSON)).
```

```

        accept(MediaType.APPLICATION_JSON)
    ) . andExpect(status().isBadRequest());
}

```

U četvrtom ispitnom slučaju ispitan je pokušaj dodjele uloge "Stanovnik" postojećem korisniku preko "/accounts/grantRole/" endpointa. Očekivani rezultat je uspješna dodjela uloge "Stanovnik" korisniku te HTTP status 200 (OK).

```

@Test
public void whenGrantRole_thenReturnRole() throws
Exception {
    given(accountService.fetch(any(Long.class))).
        willReturn(account);
    given(accountService.existsById(eq(1L))).willReturn(
        true);
    given(roleService.findByName(any(String.class))).
        willReturn(java.util.Optional.of(new Role(""
        Stanovnik)));
}

mvc.perform(put("/accounts/grantRole/{id}", 1)
    .content("Stanovnik") .contentType(MediaType.
        APPLICATION_JSON) .accept(MediaType.
        APPLICATION_JSON)
) . andExpect(status().isOk())
    . andExpect(jsonPath("$.name").value(""
        Stanovnik));
}

```

U petom ispitnom slučaju ispitan je pokušaj izrade nove objave uz priložen tekst objave i priložen id postojeće dretve. Očekivani rezultat je uspješna izrada nove objave koja sadrži priložen tekst, te HTTP status 201 (Created).

```

@Test
public void
givenPost_whenCreatePost_thenReturnPostAsJson()
throws Exception {
    Post post2 = new Post(2L, "Second_post",
        null,
        null, postThread, account);
}

```

```

post2.setThreadId(postThread.getId());
given(accountService.fetch(any(Long.class))).
    willReturn(account);
given(postService.createPost(any(Post.class)), eq(
    postThread.getId())).willReturn(post2);
given(postService.existsById(eq(post2.getId()))).
    willReturn(false);
given(postThreadService.existsById(eq(postThread.
    getId()))).willReturn(true);

mvc.perform(post("/posts/{id}", 1)
    .content(new ObjectMapper().
        writeValueAsString(post2))
    .contentType(MediaType.APPLICATION_JSON).
        accept(MediaType.APPLICATION_JSON))
    .andExpect(status().isCreated())
    .andExpect(jsonPath("$.id").value(2))
    .andExpect(jsonPath("$.content").value(""
        "Second-post")));
}

```

U šestom ispitnom slučaju ispitan je pokušaj dohvaćanja objave preko nepostojećeg id-a. Očekivani rezultat je neuspješno dohvaćanje objave te HTTP status 400 (Bad Request).

```

@Test
public void
givenNonexistentPostId_whenGetPost_thenCauseError400
() throws Exception {
    given(postService.existsById(any(Long.class))).
        willReturn(false);

    mvc.perform(get("/posts/{id}", 5)
        .contentType(MediaType.APPLICATION_JSON)
        .accept(MediaType.APPLICATION_JSON)
    ).andExpect(status().isBadRequest());
}

```

{}

Na slici 5.1 prikazana je snimka zaslona terminala u IntelliJ IDE-u, kao rezultat izvođenja prethodno navedenih šest ispitnih slučajeva.

```
[INFO] Results:  
[INFO]  
[INFO] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0  
[INFO]  
[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 01:13 min  
[INFO] Finished at: 2022-01-04T23:16:51+01:00  
[INFO] -----  
  
Process finished with exit code 0
```

Slika 5.1: Rezultati ispitivanja pomoću Spring-a i JUnit-a

5.2.2 Ispitivanje sustava

Ispitivanje sustava ostvareno je korištenjem Selenium WebDrivera i programskog jezika Python. U nastavku su opisani provedeni testovi, te su priloženi kodovi.

U prvom ispitnom slučaju ispitan je pokušaj prijave s emailom s kojim nije povezan niti jedan račun, tj. emailom koji ne postoji u sustavu. Očekivani izlaz je neuspjeh pri pokušaju prijave.

```
def test1 () -> bool:
    driver = webdriver.Firefox()
    driver.get("http://localhost:3000/login")
    driver.find_element_by_name("username").send_keys(
        "stanovnik@stanovnik.com")
    driver.find_element_by_name("password").send_keys(
        "stanovnik")
    driver.find_element_by_css_selector("button[type='submit']").click()
    try:
        driver.find_element_by_class_name("logout")
    except NoSuchElementException:
        driver.close()
        return True
    driver.close()
    return False
```

U drugom ispitnom slučaju ispitan je pokušaj registracije s neispravnim podacima, konkretno s emailom koji je zadan u krivom formatu. Očekivani izlaz je neuspjeh pri pokušaju registracije.

```
def test2 () -> bool:
    driver = webdriver.Firefox()
    driver.get("http://localhost:3000/login")
    driver.find_element_by_css_selector("button[type='button']").click()
    driver.find_element_by_name("firstname").send_keys(
        "Stanovnik")
    driver.find_element_by_name("lastname").send_keys(
        "Stanovnikic")
```

```
driver.find_element_by_name("username").send_keys("stanovnik")
driver.find_element_by_name("password").send_keys("stanovnik")
dropdown = driver.find_element_by_class_name("css-tlfecz-indicatorContainer")
dropdown.click()
actions = ActionChains(driver)
actions.move_to_element(dropdown).send_keys("Ulica_1_Kvarta_1").key_down(Keys.ENTER).key_up(Keys.ENTER).perform()
driver.find_element_by_name("streetnumber").send_keys("7")
driver.find_element_by_css_selector("button[type='submit']").click()
try:
    driver.find_element_by_class_name("logout")
except NoSuchElementException:
    driver.close()
    return True
driver.close()
return False
```

U trećem ispitnom slučaju ispitana je pokušaj registracije gdje su svi uneseni podaci valjani. Očekivani izlaz je uspjeh pri pokušaju registracije i preusmjeravanje na početnu stranicu korisnikovog kvarta.

```
def test3() -> bool:
    driver = webdriver.Firefox()
    driver.get("http://localhost:3000/login")
    driver.find_element_by_css_selector("button[type='button']").click()
    driver.find_element_by_name("firstname").send_keys("Stanovnik")
    driver.find_element_by_name("lastname").send_keys("Stanovnikic")
    driver.find_element_by_name("username").send_keys("
```

```
stanovnik@stanovnik.com")
driver.find_element_by_name("password").send_keys(
    "stanovnik")
dropdown = driver.find_element_by_class_name("css-tlfecz
-indicatorContainer")
dropdown.click()
actions = ActionChains(driver)
actions.move_to_element(dropdown).send_keys("Ulica_1_
Kvarta_1").key_down(Keys.ENTER).key_up(Keys.ENTER).per
form()
driver.find_element_by_name("streetnumber").send_keys("7
")
driver.find_element_by_css_selector("button[type='submit
']").click()
try:
    driver.find_element_by_class_name("logout")
except NoSuchElementException:
    driver.close()
    return False
driver.close()
return True
```

U četvrtom ispitnom slučaju ispitan je pokušaj prijave s emailom za koji postoji korisnički račun, ali s neispravnom lozinkom. Očekivani izlaz je neuspjeh pri pokušaju prijave.

```
def test4() -> bool:
    driver = webdriver.Firefox()
    driver.get("http://localhost:3000/login")
    driver.find_element_by_name("username").send_keys(
        "stanovnik@stanovnik.com")
    driver.find_element_by_name("password").send_keys(
        "stanovni")
    driver.find_element_by_css_selector("button[type='submit
']").click()
    try:
        driver.find_element_by_class_name("logout")
```

```
except NoSuchElementException:  
    driver.close()  
    return True  
driver.close()  
return False
```

U petom ispitnom slučaju ispitan je pokušaj prijave gdje su svi uneseni podaci ispravni. Očekivani izlaz je uspjeh pri pokušaju prijave i preusmjeravanje na početnu stranicu korisnikovog kvarta.

```
def test5 () -> bool:  
    driver = webdriver.Firefox()  
    driver.get("http://localhost:3000/login")  
    driver.find_element_by_name("username").send_keys("stanovnik@stanovnik.com")  
    driver.find_element_by_name("password").send_keys("stanovnik")  
    driver.find_element_by_css_selector("button[type='submit']").click()  
    try:  
        driver.find_element_by_class_name("logout")  
    except NoSuchElementException:  
        driver.close()  
        return False  
    driver.close()  
    return True
```

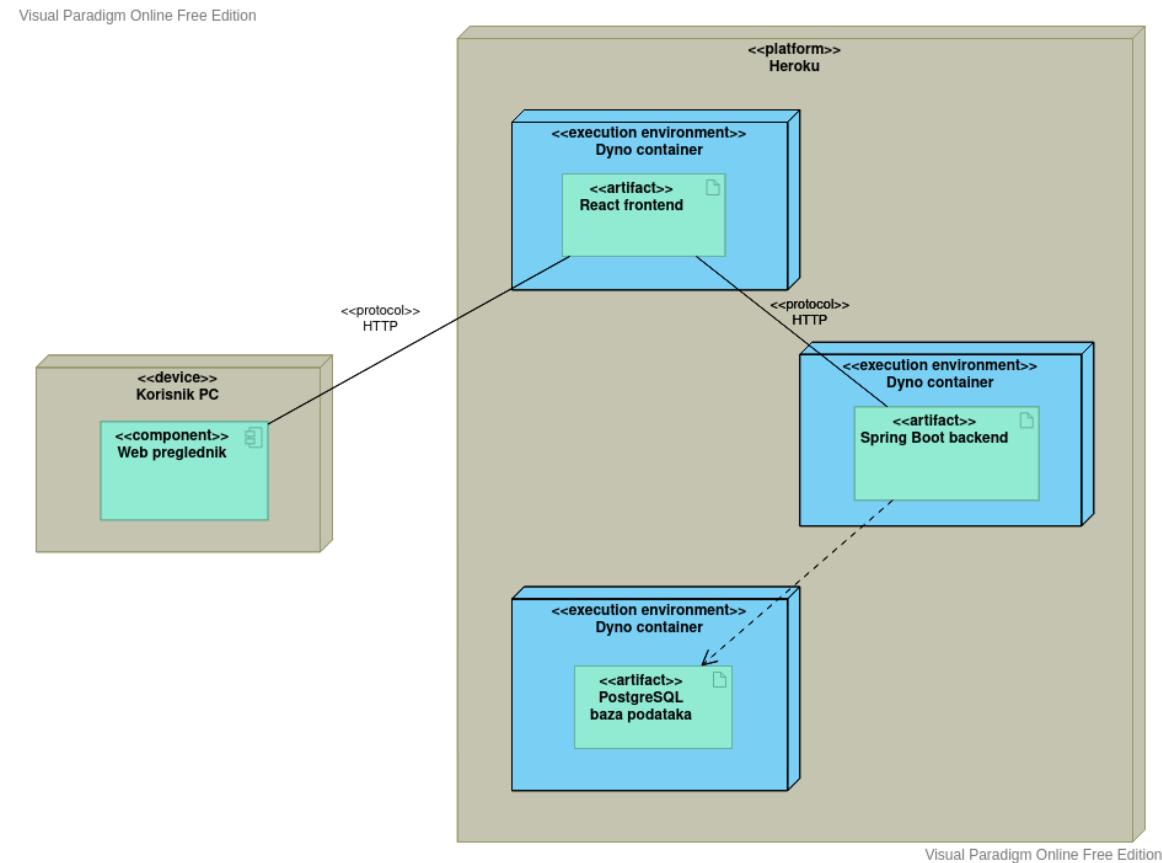
Na slici 5.2 prikazana je snimka zaslona terminala kao rezultat izvođenja prethodno navedenih pet ispitnih slučajeva.

```
mihael@pop-os:~/Desktop/Selenium$ python3 testiranje_sustava.py  
Rezultati testiranja sustava:  
Test 1 - USPJEŠAN!  
Test 2 - USPJEŠAN!  
Test 3 - USPJEŠAN!  
Test 4 - USPJEŠAN!  
Test 5 - USPJEŠAN!
```

Slika 5.2: Rezultati ispitivanja Selenium WebDriverom

5.3 Dijagram razmještaja

Na slici 5.3 prikazan je dijagram razmještaja. Sustav je baziran na arhitekturi "klijent-poslužitelj". Korisnici pristupaju aplikaciji korištenjem web preglednika. Na platformi Heroku se nalaze poslužitelji za frontend, backend i bazu podataka. Komunikacija između korisnika i poslužitelja za frontend, te poslužitelja za frontend i poslužitelja za backend, ostvaruje se korištenjem protokola HTTP.



Slika 5.3: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Potrebno je napraviti korisnički račun na Heroku i zatim odabirom opcije "New app" stvoriti dvije aplikacije, jednu za frontend i jednu za backend.

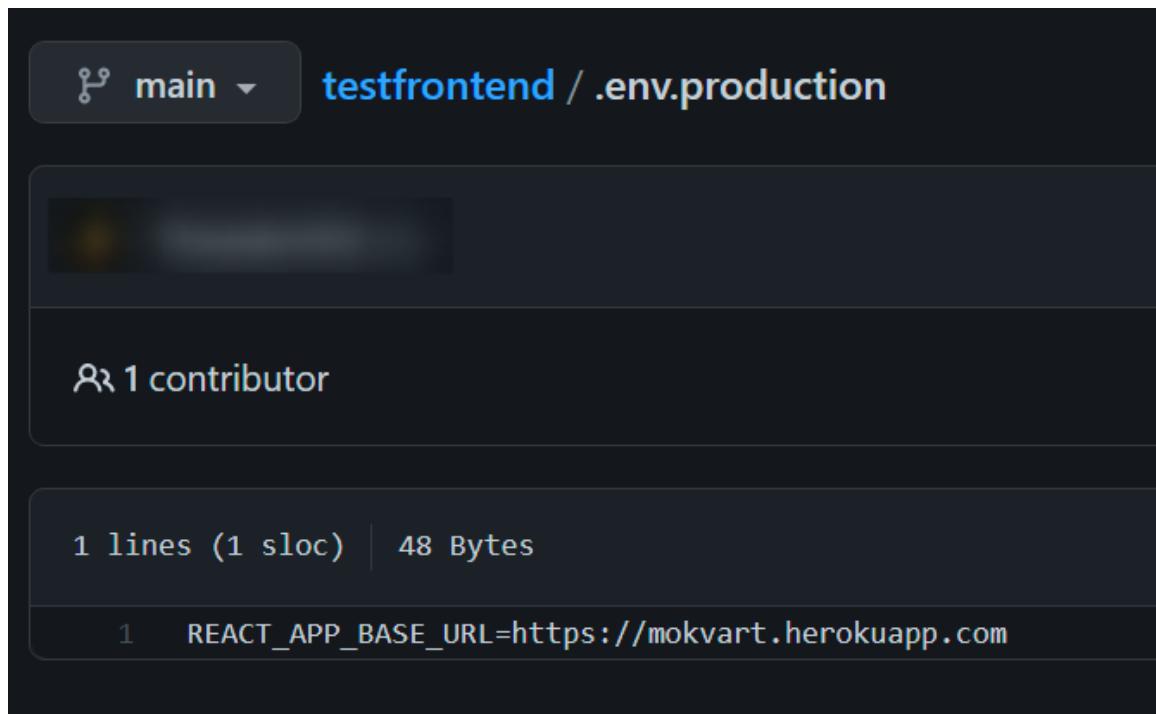
S obzirom da je puštanje u pogon značajno jednostavnije s Githuba nego s Gitlaba, potrebno je neki, npr. osobni Github račun, povezati s Heroku računom i potom klonirati projekt dva puta. U prvom kloniranom projektu je potrebno u root direktorij premjestiti sadržaj poddirektorija IzvorniKod/backend-spring-boot, a ostatak sadržaja projekta obrisati. U drugom je projektu potrebno učiniti istu stvar za sadržaj poddirektorija IzvorniKod/frontend-react. Zatim je u svakom projektu na main grani potrebno odabrati opciju "Deploy to Heroku".

Na Heroku Vas dočeka stranica poput one na slici 5.4. Kada ste u pogon pustili frontend i backend, potrebno je u datoteku .env.production na frontend projektu napisati točan URL na kojem se nalazi backend. Primjer je pokazan na slici 5.5.

Konačni je korak napuniti bazu s nekom .sql skriptom. Pri puštanju u pogon backenda, automatski se stvari i baza podataka. Na slici 5.4 vidljiva je poveznica "Resources". Potrebno je odabrat ju, zatim je potrebno pod "add-ons" odabrat "heroku postgres" i dočekaju Vas podaci slični onima na slici 5.6. Zatim je potrebno otvoriti pgAdmin, i odabrat opciju "Create Server", kao što je prikazano na slici 5.7. Dalje je potrebno slijediti upute i unijeti podatke poput onih na slici 5.6, i konačno, kada je baza podataka povezana lokalno s pgAdminom, potrebno je odabirom na "Query tool" unijeti željenu .sql skriptu.

The screenshot shows the Heroku Platform interface. At the top, there's a navigation bar with links for Overview, Resources, Deploy, Metrics, Activity, Access, and Settings. A search bar says "Jump to Favorites, Apps, Pipelines, Spaces...". On the left, there's a sidebar with "Personal" and "GitHub" sections. The main area has sections for "Add this app to a pipeline" (with instructions to create a new pipeline or choose an existing one), "Deployment method" (with options for Heroku Git, GitHub Connected, and Container Registry), and "App connected to GitHub" (with a message that code diffs, manual and auto deploys are available). Below that is a section for "Automatic deploys" where users can enable automatic deployments from GitHub. The final section is "Manual deploy" where users can deploy a GitHub branch. At the bottom, there are links for heroku.com, Blogs, Careers, Documentation, Support, Terms of Service, Privacy, Cookies, and a copyright notice for © 2022 Salesforce.com.

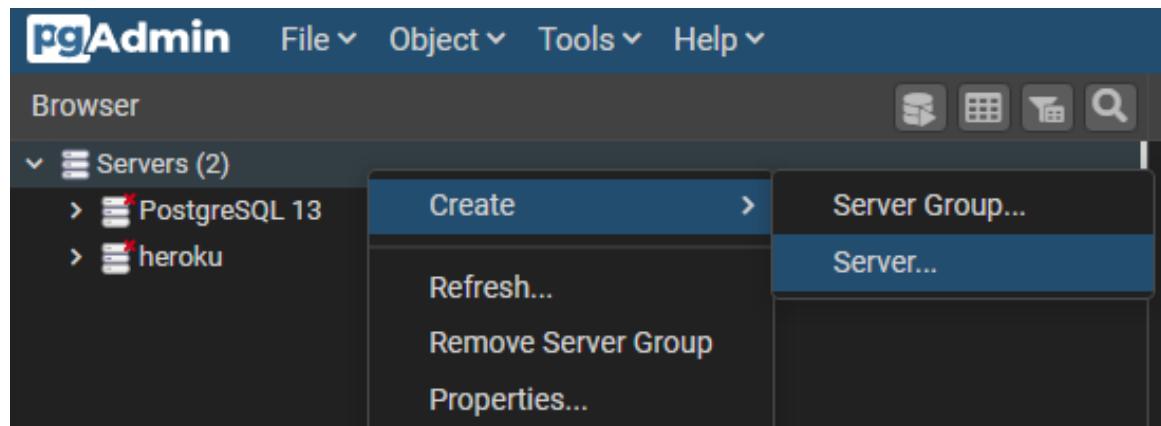
Slika 5.4: Prikaz aplikacije na poslužitelju Heroku



Slika 5.5: Povezivanje frontenda i backenda

A screenshot of the Heroku Datastore settings page for a PostgreSQL database named 'postgresql-flat-08117'. The page shows basic information like the service, plan, and app it's attached to. It includes tabs for Overview, Durability, Settings, and Dataclips. The 'Settings' tab is active. Under 'ADMINISTRATION', there's a section for 'Database Credentials' with fields for Host, Database, User, Port, and URI. The 'URI' field contains a complex PostgreSQL connection string. There are also sections for 'Reset Database' and 'Destroy Database'. Buttons for 'Cancel', 'Reset Database...', and 'Destroy Database...' are visible.

Slika 5.6: Podaci za pristup bazi



Slika 5.7: Pristup bazi lokalno u pgAdminu

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije pod nazivom "Moj kvart". Ideja aplikacije je da bude svojevrsna društvena mreža preko koje stanovnici istih kvartova mogu komunicirati, komentirati neke teme na forumu, predlagati grupne događaje i slično. Nakon 12 tjedana timskog rada, ostvarili smo zadani cilj i projekt je završen. Projekt je imao tri faze.

Prva faza je uključivala okupljanje tima, razgovor o generalnim idejama za aplikaciju, izražavanje pojedinačnih interesa i želja za radom u prvom ciklusu predaje, te konačno podjela zadataka za prvi ciklus. Formirala su se dva podtima. U prvom podtimu su članovi radili na dokumentiranju zahtjeva, obrazaca uporabe, UML dijagrama te bazi podataka. U drugom podtimu su članovi istraživali, proučavali i eksperimentirali s tehnologijama u kojima će kasnije biti implementirana aplikacija. Prva je faza projekta trajala do kolokviranja prvog ciklusa projekta.

U drugoj je fazi naglasak bio na implementaciji aplikacije, i ovdje nije bilo podtima nego su svi članovi zajedno radili na programskom ostvarenju aplikacije. U ovoj je fazi implementirana većina aplikacije, a trajala je do demonstracije alfa inačice aplikacije.

Treća i konačna faza je uključivala izradu raznih UML dijagrama, ispitivanje sustava, pronalazak i ispravak grešaka, rad na izgledu aplikacije i implementacija preostalih funkcionalnosti. U ovoj fazi je postojalo dosta manjih zadataka koje je trebalo napraviti, pa su članovi tima uglavnom samostalno preuzimali te zadatke i obavljali ih. Ova je faza trajala do završetka projekta, prije konačne predaje i kolokviranja drugog ciklusa.

Izgrađenu aplikaciju je moguće proširiti na mnogo načina. Jedna od ideja je u Forum dodati više funkcionalnosti, tako da korisnici mogu pregledati druge korisnike i sve njihove objave, te da dobivaju obavijesti kada im netko odgovori na objavu. Drugo proširenje koje bi doprinijelo kvaliteti aplikacije je mogućnost da korisnici šalju izravno poruke drugim korisnicima.

Zbog problema s Google Calendar API-jem, nije implementirana sinkronizacija događaja unutar aplikacije s Google Kalendrom.

Sudjelovanje u ovom projektu je bilo vrijedno iskustvo za sve članove tima.

Svima nama je ovo bio prvi ozbiljniji grupni projekt, i snašli smo se jako dobro. Konflikata u timu gotovo da nije bilo, a suradnja i komunikacija su bili na iznimno zadovoljavajućoj razini. Većini nas je ovaj projekt bio prvi ozbiljniji dodir s tehnologijama poput Gita i Latex. Naučili smo koristiti neke moderne radne okvire pri izradi web aplikacija. Iznimno smo zadovoljni postignutim rezultatima i timskim radom koji je do tih rezultata doveo.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T. C. Lethbridge, R. Laganiere, "Object-Oriented Software Engineering", 2nd ed, McGraw-Hill, 2005.
4. G. Booch, J. Rumbaugh, I. Jacobson, "Unified Modeling Language User Guide", 2nd ed, Addison Wesley, 2005.
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. OMG UML, <https://www.omg.org/spec/UML/>
7. Visual Paradigm Online, <https://online.visual-paradigm.com/>

Indeks slika i dijagrama

3.1	Dijagram obrazaca uporabe, osnovne funkcionalnosti	35
3.2	Dijagram obrazaca uporabe, funkcionalnosti upravljanja osobnim podacima	36
3.3	Dijagram obrazaca uporabe, funkcionalnosti sekcije "Događaji" . . .	37
3.4	Dijagram obrazaca uporabe, funkcionalnosti sekcije "Forum"	38
3.5	Dijagram obrazaca uporabe, funkcionalnosti sekcije "Vijeće četvrti"	39
3.6	Dijagram obrazaca uporabe, funkcionalnosti Administratora	40
3.7	Sekvencijski dijagram za UC9, UC10, UC11, UC12, UC13	42
3.8	Sekvencijski dijagram za UC27, UC32, UC33, UC34	43
3.9	Sekvencijski dijagram za UC30	44
3.10	Sekvencijski dijagram za UC43, UC7	46
4.1	Relacijska shema baze podataka	55
4.2	Dijagram razreda - entiteti	57
4.3	Dijagram razreda - sloj Repository	58
4.4	Dijagram razreda - sloj Service	59
4.5	Dijagram razreda - sloj Controller	60
4.6	Dijagram razreda - odnos slojeva Controller i Service	60
4.7	Dijagram razreda - odnos slojeva Service i Repository	61
4.8	Dijagram razreda - odnos sloja Service i entiteta	61
4.9	Dijagram razreda - odnos sloja Repository i entiteta	62
4.10	Dijagram stanja	64
4.11	Dijagram aktivnosti	66
4.12	Dijagram komponenti	67
5.1	Rezultati ispitivanja pomoću Spring-a i JUnit-a	74
5.2	Rezultati ispitivanja Selenium WebDriverom	78
5.3	Dijagram razmještaja	79
5.4	Prikaz aplikacije na poslužitelju Heroku	81
5.5	Povezivanje frontenda i backenda	82
5.6	Podaci za pristup bazi	82

5.7 Pristup bazi lokalno u pgAdminu	83
6.1 Prikaz aktivnosti u repozitoriju	94

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 3.10.2021.
- Prisustvovali: Andrija Banić, Danijel Barišić, Danko Čurlin, Bartol Hrg, Mihael Miličević, Dario Pavlović, Tomislav Žiger
- Teme sastanka:
 - formiranje tima
 - dogovor o imenu tima
 - dogovor o korištenim tehnologijama

2. sastanak

- Datum: 13.10.2021.
- Prisustvovali: Andrija Banić, Danijel Barišić, Danko Čurlin, Dario Pavlović
- Teme sastanka:
 - s asistenticom grupe raspravljali o osnovnim pitanjima na projektu

3. sastanak

- Datum: 14.10.2021.
- Prisustvovali: Andrija Banić, Danijel Barišić, Danko Čurlin, Bartol Hrg, Mihael Miličević, Dario Pavlović, Tomislav Žiger
- Teme sastanka:
 - okvirna podjela zadataka na projektu u prvom ciklusu predaje

4. sastanak

- Datum: 15.10.2021.
- Prisustvovali: Andrija Banić, Danijel Barišić, Mihael Miličević
- Teme sastanka:
 - izrada prve inačice ER modela baze podataka

5. sastanak

- Datum: 18.10.2021.

- Prisustvovali: Mihael Miličević, Tomislav Žiger
- Teme sastanka:
 - s asistenticom grupe raspravljali o izrađenom ER modelu baze podataka
 - s asistenticom grupe raspravljali o aktivnostima na projektu do prve predaje

6. sastanak

- Datum: 27.10.2021.
- Prisustvovali: Mihael Miličević, Dario Pavlović
- Teme sastanka:
 - s asistenticom grupe raspravljali o izrađenim obrascima uporabe
 - s asistenticom grupe komentirali izborne tehnologije na projektu

7. sastanak

- Datum: 30.10.2021.
- Prisustvovali: Andrija Banić, Danijel Barišić, Danko Čurlin, Bartol Hrg, Mihael Miličević, Dario Pavlović, Tomislav Žiger
- Teme sastanka:
 - raspravljali o aktivnostima na projektu do prve predaje

8. sastanak

- Datum: 3.11.2021.
- Prisustvovali: Bartol Hrg, Mihael Miličević
- Teme sastanka:
 - s asistenticom grupe raspravljali o izrađenim UML dijagramima obrazaca uporabe i sekveničijskim dijagramima

9. sastanak

- Datum: 10.11.2021.
- Prisustvovali: Danijel Barišić, Mihael Miličević, Tomislav Žiger
- Teme sastanka:
 - izrada dijagrama razreda

10. sastanak

- Datum: 8.12.2021.
- Prisustvovali: Andrija Banić, Danijel Barišić, Danko Čurlin, Bartol Hrg, Mihael Miličević, Dario Pavlović, Tomislav Žiger
- Teme sastanka:

- rasprava o aktivnostima na projektu i zadacima do prezentacije alfa inačice

11. sastanak

- Datum: 1.1.2022.
- Prisustvovali: Andrija Banić, Danijel Barišić, Danko Čurlin, Bartol Hrg, Mihael Miličević, Dario Pavlović, Tomislav Žiger
- Teme sastanka:
 - rasprava o aktivnostima na projektu i zadacima do konačne predaje projekta

Tablica aktivnosti

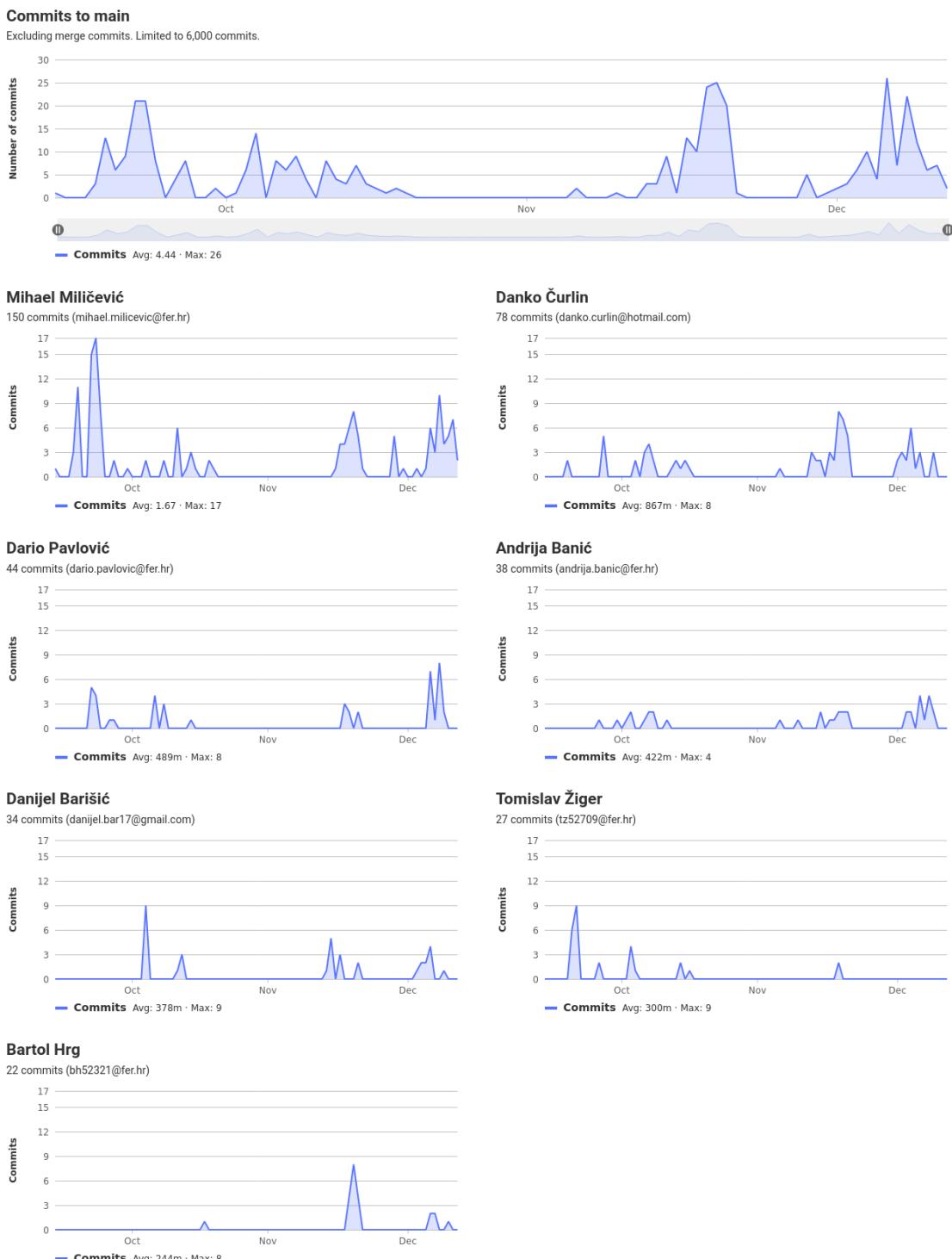
	Mihael Miličević	Andrija Banić	Danijel Barišić	Danko Čurin	Bartol Hrg	Dario Pavlović	Tomislav Žiger
Upravljanje projektom	10						
Opis projektnog zadatka	2						
Funkcionalni zahtjevi	3						
Opis pojedinih obrazaca	20						10
Dijagram obrazaca	5						1
Sekvencijski dijagrami	5						3
Opis ostalih zahtjeva							1
Arhitektura i dizajn sustava	1						
Baza podataka	2	5					
Dijagram razreda	8		3				5
Dijagram stanja	3						
Dijagram aktivnosti	2						
Dijagram komponenti	2						
Korištene tehnologije i alati	1						
Ispitivanje programskog rješenja	5		8				
Dijagram razmještaja	2						
Upute za puštanje u pogon	2			2			
Dnevnik sastajanja	1						
Zaključak i budući rad	1						
Popis literature	1						

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Mihael Miličević	Andrija Banić	Danijel Barišić	Danko Čurlin	Bartol Hrg	Dario Pavlović	Tomislav Žiger
Izrada baze podataka	1	20	1	3		3	
Backend		5	100	60	20	40	
Frontend	60	60		100	15	15	10
Izgled aplikacije	20	15			2	30	
Puštanje u pogon				10		5	

Dijagrami pregleda promjena



Slika 6.1: Prikaz aktivnosti u repozitoriju