

Programsko inženjerstvo

Ak. god. 2023./2024.

<Naziv projekta>

Dokumentacija, Rev. <1 ili 2>

Grupa: *SourceresOfTheNorth*

Voditelj: *Petar Belošević*

Datum predaje: <dan>. <mjesec>. <godina>.

Nastavnik: *Vlado Sruk*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
2.1	Primjeri u \LaTeX u	10
3	Specifikacija programske potpore	13
3.1	Funkcionalni zahtjevi	13
3.1.1	Obrasci uporabe	15
3.1.2	Sekvencijski dijagrami	24
3.2	Ostali zahtjevi	30
4	Arhitektura i dizajn sustava	31
4.1	Baza podataka	34
4.1.1	Opis tablica	34
4.1.2	Dijagram baze podataka	38
4.2	Dijagram razreda	39
4.3	Dijagram stanja	40
4.4	Dijagram aktivnosti	41
4.5	Dijagram komponenti	42
5	Implementacija i korisničko sučelje	43
5.1	Korištene tehnologije i alati	43
5.2	Ispitivanje programskog rješenja	44
5.2.1	Ispitivanje komponenti	44
5.2.2	Ispitivanje sustava	44
5.3	Dijagram razmještaja	45
5.4	Upute za puštanje u pogon	46
6	Zaključak i budući rad	47
	Popis literature	48

Indeks slika i dijagrama	49
Dodatak: Prikaz aktivnosti grupe	50

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak. Započet opis zadatka.	Petar Belošević	23.10.2023.
0.2	Dovršen opis projektnog zadatka.	Petar Belošević	28.10.2023.
0.3	Dodani dionici i aktori.	Petar Belošević	29.10.2023.
0.4	Napravljeni obrasci uporabe i dijagrami obrazaca uporabe	Petar Belošević	4.11.2023.
0.5	Dodani sekvencijski dijagrami s opisima, napisani ostali zahtjevi	Petar Belošević	7.11.2023.
0.9	Opisi obrazaca uporabe	*	07.09.2013.
0.10	Preveden uvod	*	08.09.2013.
0.11	Sekvencijski dijagrami	*	09.09.2013.
0.12.1	Započeo dijagrame razreda	*	10.09.2013.
0.12.2	Nastavak dijagrama razreda	*	11.09.2013.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene: Timer - Brojilo vremena	*	15.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.
2.0	Konačni tekst predloška dokumentacije	*	28.09.2013.

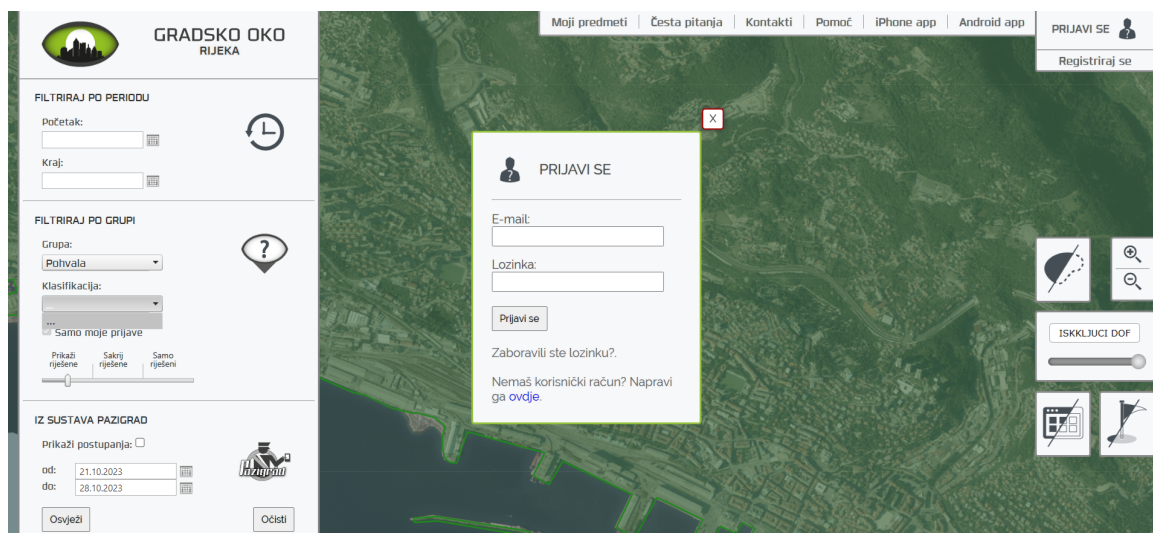
Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "IME APLIKACIJE". Ideja aplikacije je ponuditi običnim građanima jednostavan i efikasan način prijave oštećenja javnih površina i cesta u gradu te dobivanje povratnih informacija konkretnom oštećenju i njihovoj prijavi.

Građani će svojim djelovanjem putem aplikacije unaprjeđivati kvalitetu života u svojoj zajednici. Svojim prijavama, koje će biti javno dostupne, građani će poticati gradske urede da saniraju prijavljena oštećenja. Aplikacija će također pomoći gradskim uredima u obavljanju svojih zadaća. Naime, preko aplikacije uredi će imati puno bolji uvid u probleme koji postoje u zajednici. Zahvaljujući aplikaciji uredi će moći brže i kvalitetnije reagirati na probleme u zajednici.

Slične aplikacije već postoje u Hrvatskoj. Tako primjerice Grad Rijeka na svojim stranicama nudi mogućnost prijave raznih prekršaja i oštećenja (<https://gov.rijecka.hr/zahtjevi-i-obrasci/komunalna-djelatnost-i-prijava-komunalnog-nereda/prijave-ostecenja-nereda-ili-nepropisnog-parkiranja/383>). Njihova stranica nudi mogućnost prijave u 4 kategorije. Neke prijave se podnose putem obrazaca, neke samo telefonskim putem, a za neke su ponuđene i specijalne stranice. Tako primjerice za prijavu većine problema, ali i za podnošenje pohvala, Grad Rijeka nudi web aplikaciju *Gradsko oko* (<http://rijecka.oko.hr/>) dostupnu i u mobilnoj verziji. Aplikacija nudi mogućnost podnošenja i pregleda prijava putem interaktivne karte. Prijave su kategorizirane po grupama i klasifikacijama. Prilikom pregleda se mogu filtrirati po vremenu i kategorijama. Prijave i pregled prijava su dostupne samo prijavljenim korisnicima te svaka prijava mora biti odobrena od strane administratora.

Slika 2.1: Aplikacija *Gradsko oko* Grada Rijeke

Građani aplikaciju mogu koristiti kao registrirani i kao neregistrirani korisnici. Na naslovnoj stranici građanima su ponuđene opcije podnošenja nove prijave, pregleda postojećih prijava i pregled statistike prijava. Registrirani građani također imaju opciju pregleda svojeg profila dok neregistrirani građani imaju mogućnost registracije, tj. izrade profila odnosno prijave, ako već imaju izrađeni profil.

Neregistrirani korisnik se može registrirati te prilikom registracije mora navesti sljedeće informacije:

- ime
- prezime
- datum rođenja
- email adresa
- lozinka

Korisnik također može priložiti profilnu sliku, ali ona nije nužna za izradu profila. Prijavljeni korisnici prilikom pregleda svojeg profila mogu vidjeti prije navedene podatke o sebi koje mogu i izmjeniti. Također imaju pregled nad svim prijavama koje su podnesli kroz aplikaciju. Prikazana im je i kratka statistika prijava (broj prijava, broj riješenih prijava, broj prijava na čekanju i u procesu rješavanja).

Građanima se prilikom podnošenja nove prijave otvara obrazac koji je potrebno ispuniti. Prijava sadrži naslov koji definira oštećenje. Slijedi kratak opis u kojem korisnik detaljnije opisuje oštećenje u slobodnoj formi. Korisnik dalje prilaže geografski položaj oštećenja koje se prijavljuje. Geografski položaj se prilaže oda-

birom položaja na karti koja će se moći otvoriti prilikom ispunjavanja obrasca ili upisivanjem adrese na kojoj se nalazi oštećenje. Također korisnik će moći priložiti geografski položaj automatski preko svoje trenutne lokacije. Korisniku se nudi opcija da uz prijavu priloži fotografije oštećenja. Ukoliko korisnik priloži fotografije, aplikacija će pokušati iz njih izvući podatke o lokaciji te ih ponuditi korisniku kao lokaciju oštećenja. Korisnik u prijavi mora odabrati gradski ured nadležan za konkretan problem. Prilikom opisivanja oštećenja aplikacija će pokušati sama zaključiti koji ured bi mogao biti nadležan za problem koji se prijavljuje te će isti predložiti korisniku. Također, ako u sustavu već postoji slična nedavna prijava na istoj lokaciji aplikacija će predložiti korisniku već postojeću prijavu i ponuditi mu da svoju prijavu nadoveže na već postojeću. Nakon podnošenja prijave aplikacija korisniku daje jedinstveni kod podnesene prijave koja služi za praćenje te prijave preko same aplikacije. Prijavljeni korisnici će također primati obavijesti o svojoj prijavi preko emaila.

Prilikom pregleda postojećih prijava korisniku se otvara lista svih aktivnih prijava. Lista se može filtrirati po vremenu podnošenja prijave, lokaciji, statusu, nadležnom gradskom uredu. Korisnik također može pretraživati prijave po jedinstvenom kodu prijave. Na taj način svaki korisnik može pregledavati svoje prijave na aplikaciji. Za svaku prijavu se prikazuju informacije iz podnesenog obrasca. Također se prikazuje status prijave (*na čekanju*, *u procesu rješavanja*), broj prijava koje su vezane za ovu prijavu, odnosno prijava za koju je vezana ova prijava. Prijavljeni korisnici također vide svoje prijave na pregledu svojeg profila. Pregled prijava je također moguć preko karte. Prijave koje su riješene mogu se pregledati jedino preko njihovog koda ili preko pregleda korisničkog profila kod registriranih korisnika. U drugim slučajevima takve prijave nisu više dostupne za pregled.

Prikaz statistike prijava nudi filtriranje podataka po lokaciji prijave, vremenu podnošenja prijave i nadležnom uredu. Za odabrane parametre aplikacija prikazuje:

- ukupan broj podnešenih prijava
- broj prijava sa statusom *na čekanju* i njihov udio
- broj prijava sa statusom *u procesu rješavanja* i njihov udio
- broj prijava sa statusom *riješena* i njihov udio
- prosječan broj podnesenih prijava u danu
- prosječan broj dana koji prijava provede na čekanju
- prosječan broj dana za vrijeme kojih je prijava u procesu rješavanja

Djelatnici gradskog ureda se obavezno u sustav prijavljuju preko profila gradskog ureda. Djelatnici na svojoj naslovnoj stranici imaju 3 liste prijava, jedna za svaku vrstu statusa: novopristigle, one u procesu rješavanja i riješene prijave. Djelatnici ureda imaju isključivo pristup prijavama za probleme za koje je nadležan njihov ured. Ostale prijave djelatnicima nisu vidljive.

Djelatnici pregledavaju pristigle prijave i stavljaju ih u proces rješavanja. Time prijave prelaze u listu prijava u procesu rješavanja i mijenjaju svoj status. Aplikacija u tom slučaju automatski šalje mail prijavitelju ako se radi o registriranom korisniku.

Kada neka prijava bude riješena, djelatnik prijavi mijenja status u *riješena*. Prijava prelazi u za to predviđenu listu te aplikacija šalje mail prijavitelju, ako se radi o registriranom korisniku.

Djelatnik ima mogućnost objedinjavanja prijava. Naime, ako se pristigla prijava referira na problem koji je već ranije prijavljen ili je prijavljen u još nekoj pristigloj prijavi, djelatnik može grupirati te prijave. U tom slučaju se najstarija od tih prijava tretira kao glavna prijava, a ostale prijave postaju nadovezane na tu prijavu. Aplikacija u tom slučaju šalje prikladnu obavijest registriranim korisnicima čije su prijave sada nadovezane na glavnu prijavu.

Kao i kod običnih korisnika, djelatnici ureda mogu pregledavati prijave preko karte. Također mogu pregledavati statistiku prijava kao i obični korisnici, uz restrikciju na prijave vezane na njihov ured, kao što je prije navedeno.

Aplikacija je napravljena tako da podržava višejezičnost. Aplikacija podržava hrvatski i engleski jezik.

Aplikacija je napravljena na primjeru Grada Zagreba, prvenstveno u smislu da će aplikacija predviđati postojanje gradskih ureda relevantnih za ovu problematiku kakvi su ustrojeni u Zagrebu. Konkretno, aplikacija predviđa postojanje sljedećih ureda:

- Gradski zavod za zaštitu spomenika kulture i prirode - za oštećenja na spomenicima kulture i prirode
- Gradski ured za mjesnu samoupravu, promet, civilnu zaštitu i sigurnost - za oštećenja na komunalnoj infrastrukturi vezanoj za oborinske vode, površinama javne namjene, u gradskim parkovima i zelenim površinama, cestama
- Gradski ured za obnovu, izgradnju, prostorno uređenje, graditeljstvo i komunalne poslove - za oštećenja na zgradama, komunalne probleme

Također, sve karte na aplikaciji će inicijalno prikazivati područje Grada Zagreba.

Prijavljivanja oštećenja izvan tog područja neće biti moguća, tj. takva lokacija će se smatrati nevažećom i takva prijava se neće moći podnesti.

Uz male preinake moguće je prilagoditi aplikaciju da pruža istu potporu građanima nekog drugog grada ili općine, bilo u Hrvatskoj ili nekoj drugoj zemlji. Aplikacija bi se mogla skalirati i na veće jedinice lokalne samouprave kao što su županije, s obzirom da obično imaju sličan ustroj ureda i slične nadležnosti kao i gradovi. Ako se ideja skalira još više, aplikacija bi mogla pružati uslugu korisnicima cijele regije (npr. Slavonija, Dalmacija, Središnja Hrvatska) ili čak čitave države. U tom slučaju u aplikaciji bi bilo potrebno dodati mogućnost prepoznavanja nadležne jedinice lokalne samouprave prije predlaganja nadležnog ureda za konkretan problem. Takvo ponašanje bi se moglo implementirati uz malo prerađivanja i korištenje već postojećih funkcionalnosti aplikacije. Generalno gledano, aplikacija bi se mogla skalirati tako da pruža uslugu korisnicima nadnacionalnoj razini, primjerice na području cijele Europske Unije. Ponašanje aplikacije bi zapravo bilo isto uz potrebe skaliranja funkcionalnosti prepoznavanja nadležnih administrativnih jedinica na nadnacionalnoj razini. U tom slučaju aplikacija svakako mora pružati mogućnost prikaza na jeziku svake države u kojoj pruža uslugu. Glavna prepreka u ovakvoj implementaciji bi bila potreba za značajnijom infrastrukturnom podrškom, prvenstveno u obliku velikih poslužitelja koji mogu obrađivati puno zahtjeva i pohranjivati velike količine podataka. Također, aplikacija bi na toj razini zahtjevala dobru suranju i koordinaciju velikog broja institucija u više različitih država što može biti zahtjevno.

2.1 Primjeri u L^AT_EXu

Ovo potpoglavlje izbrisati.

U nastavku se nalaze različiti primjeri kako koristiti osnovne funkcionalnosti L^AT_EXa koje su potrebne za izradu dokumentacije. Za dodatnu pomoć obratiti se asistentu na projektu ili potražiti upute na sljedećim web sjedištima:

- Upute za izradu diplomskog rada u L^AT_EXu - https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf
- L^AT_EX projekt - <https://www.latex-project.org/help/>
- StackExchange za Tex - <https://tex.stackexchange.com/>

podcrtani tekst, **podebljani tekst**, *nagnuti tekst*

primjer primjer primjer primjer primjer primjer

- primjer
 - primjer
 - primjer
1. primjer
 - 1.a primjer
 - b primjer
 2. primjer

primjer url-a: <https://www.fer.unizg.hr/predmet/proinz/projekt>

posebni znakovi: # \$ % & { } _ | < > ^ ~ \

naslov unutar tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

naslov unutar tablice		
ime	VARCHAR	
primjer	VARCHAR	

Tablica 2.1: Naslov s referencom izvan tablice

IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	



Slika 2.2: Primjer slike s potpisom



Slika 2.3: Primjer slike s potpisom 2

Referenciranje slike 2.3 u tekstu.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

Dionici:

1. Gradska uprava
2. Djelatnici gradskih ureda
3. Građani grada
 - Registrirani korisnici
 - Neregistrirani korisnici
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) napraviti svoj profil za koji mu je potrebno ime, prezime, datum rođenja, email adresa, lozinka i opcionalno profilna slika
 - (b) podnositi anonimnu prijavu koja sadrži naslov, opis, lokaciju i nadležni gradski ured, a opcionalno i fotografiju i prijavu na koju se nadovezuje
 - (c) pregledavati postojeće prijave preko liste ili karte te ih filtrirati po lokaciji, vremenu, statusu i nadležnom gradskom uredu
 - (d) pregledavati statistiku postojećih prijava za odabran period, lokaciju i nadležan gradski ured
2. Registrirani korisnik (inicijator) može:
 - (a) pregledavati i mijenjati osobne podatke
 - (b) obrisati svoj profil
 - (c) pregledavati svoje prošle prijave
 - (d) podnositi prijavu koja sadrži naslov, opis, lokaciju i nadležni gradski ured, a opcionalno i fotografiju i prijavu na koju se nadovezuje

- (e) pregledavati postojeće prijave preko liste ili karte te ih filtrirati po lokaciji, vremenu, statusu i nadležnom gradskom uredu
- (f) pregledavati statistiku postojećih prijava za odabran period, lokaciju i nadležan gradski ured

3. Djelatnik gradskog ureda (inicijator) može:

- (a) pregledavati prijave pristigle u njihov ured prema njihovom statusu
- (b) prijavama sa statusom *na čekanju* mijenjati status u *u procesu rješavanja*
- (c) prijavama sa statusom *u procesu rješavanja* mijenjati status u *riješena*
- (d) objediniti prijave ako se referiraju na isti problem
- (e) pregledavati prijave na karti
- (f) pregledavati statistiku prijava pristiglih u njihov ured

4. Baza podataka (sudionik) može:

- (a) spremati sve podatke o korisnicima
- (b) spremati sve podatke o prijavama

3.1.1 Obrasci uporabe

dio 1. revizije

UC1 - Prijava oštećenja

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijaviti oštećenje nadležnom gradskom uredu
- **Sudionici:** Gradski ured, Baza podataka
- **Preduvjet:** Korisnik je pronašao oštećenje koje želi prijaviti
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prijavljivanje oštećenja
 2. Korisnik popunjava podatke za prijavu oštećenja i podnosi prijavu
 3. Prijava se sprema u bazu podataka
 4. Korisnik dobiva jedinstveni kod svoje prijave.
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nije unio sve obavezne podatke u prijavu.
 1. Korisnik dobiva obavijest.
 2. Korisnik popunjava preostala obavezna polja ili odustaje od podnošenja prijave

UC2 - Pregled svih prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled aktivnih prijava u sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregledavanje prijava
 2. Aplikacija prikazuje prijave
 3. Korisnik odabire prijavu
 4. Prikazuju se podaci o prijavi

UC3 - Filtriranje prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Filtriranje prijava prilikom pregleda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je odabrao opciju pregleda svih prijava
- **Opis osnovnog tijeka:**

1. Korisnik odabire opcije kod filtriranja
2. Aplikacija prikazuje filtrirane prijave
- **Opis mogućih odstupanja:**
 - 1.a Korisnik ne želi filtrirati prijave
 1. Aplikacija prikazuje nefiltrirane prijave

UC4 - Pregled statistike prijava

- **Glavni sudionik:** Korisnik ili Djelatnik gradskog ureda
- **Cilj:** Pregled statistike prijava u sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za pregledavanje statistike prijava
 2. Aplikacija prikazuje razne podatke za prijave u sustavu

UC5 - Filtriranje statistike prijava

- **Glavni sudionik:** Korisnik ili Djelatnik Gradskog ureda
- **Cilj:** Filtriranje prijava prilikom pregleda statistike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je odabrao opciju pregleda statistike prijava
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opcije za filtriranje
 2. Aplikacija prikazuje statistiku za filtrirane prijave

UC6 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik ili Djelatnik gradskog ureda
- **Cilj:** Izrada korisničkog računa za korisnika ili novi gradski ured
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju registracije
 2. Korisnik popunjava podatke za izradu korisničkog računa i potvrđuje registraciju
- **Opis mogućih odstupanja:**
 - 2.a Unesena email adresa je već zauzeta

1. Aplikacija prikazuje prikladnu obavijest i vraća korisnika na stranicu za registraciju
2. Korisnik unosi drugu email adresu ili odustaje od registracije

UC7 - Pregled vlastitih prijava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled prijava koje je korisnik do sada podnio
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Registrirani korisnik odabire ikonu svojeg profila
 2. Registrirani korisnik na ponuđenom izborniku odabire opciju pregleda vlastitih prijava
 3. Aplikacija prikazuje listu dosadašnjih prijava Registriranog korisnika
 4. Registrirani korisnik odabire prijavu
 5. Prikazuju se podaci o prijavi
- **Opis mogućih odstupanja:**
 - 4.a Registrirani korisnik nema podnesenih prijava
 1. Aplikacija prikazuje praznu listu
 2. Registrirani korisnik odustaje od pregleda vlastitih prijava

UC8 - Pregled podataka o računu

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled podataka o korisničkom računu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Registrirani korisnik odabire ikonu svojeg profila
 2. Registrirani korisnik na ponuđenom izborniku odabire opciju pregleda podataka o računu
 3. Aplikacija prikazuje podatke o korisničkom računu

UC9 - Promjena podataka o računu

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promjena podataka o korisničkom računu
- **Sudionici:** Baza podataka

- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Registrirani korisnik odabire ikonu svojeg profila
 2. Registrirani korisnik na ponuđenom izborniku odabire opciju pregleda podataka o računu
 3. Aplikacija prikazuje podatke o korisničkom računu
 4. Registrirani korisnik odabire opciju uređivanja korisničkog računa
 5. Registrirani korisnik mijenja željene podatke o računu i potvrđuje izmjene
- **Opis mogućih odstupanja:**
 - 5.a Registrirani korisnik mijenja email adresu u već zauzetu email adresu
 1. Aplikacija javlja pogrešku s odgovarajućom porukom
 2. Registrirani korisnik unosi drugu email adresu ili odustaje od promjene email adrese ili uređivanja podataka o korisničkom računu

UC10 - Brisanje korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Brisanje korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Registrirani korisnik odabire ikonu svojeg profila
 2. Registrirani korisnik na ponuđenom izborniku odabire opciju brisanja korisničkog računa
 3. Korisnički račun se briše, korisnik postaje Neregistrirani korisnik i aplikacija ga vraća na početnu stranicu

UC11 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik ili Djelatnik gradskog ureda
- **Cilj:** Prijava korisnika u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju prijave
 2. Korisnik unosi svoju email adresu i lozinku
 3. Aplikacija otvara početnu stranicu

- **Opis mogućih odstupanja:**

2.a Registrirani korisnik unosi nevažeću kombinaciju email adrese i lozinke

1. Aplikacija javlja pogrešku s odgovarajućom porukom i traži ponovnu prijavu
2. Korisnik unosi ispravnu kombinaciju ili odustaje od prijave i koristi aplikaciju u anonimnom načinu

UC12 - Odjava iz sustava

- **Glavni sudionik:** Registrirani korisnik ili Djelatnik gradskog ureda
- **Cilj:** Odjava korisnika iz sustav
- **Sudionici:** -
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. korisnik odabire ikonu svojeg profila
 2. Korisnik na ponuđenom izborniku odabire opciju odjave iz sustava
 3. Aplikacija otvara početnu stranicu u anonimnom načinu

UC13 - Obrada prijava

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Obrada prijave u gradskom uredu
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik pregledava prijave ureda
- **Opis osnovnog tijeka:**
 1. Djelatnik gradskog ureda odabire prijavu
 2. Djelatnik mijenja status prijave

UC14 - Odbacivanje pristigle prijave

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Odbacivanje nevaljale prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik gradskog ureda obrađuje prijavu, prijava ima status *na čekanju*
- **Opis osnovnog tijeka:**
 1. Djelatnik procjenjuje da prijava nije utemeljena
 2. Djelatnik odabire opciju za odbacivanje prijave

UC15 - Prebacivanje prijave na drugi ured

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Prosljeđivanje prijave nadležnom gradskom uredu
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik gradskog ureda obrađuje prijavu, prijava ima status *na čekanju*
- **Opis osnovnog tijeka:**
 1. Djelatnik procjenjuje da je neki drugi ured nadležan za ovu prijavu
 2. Djelatnik odabire opciju prebacivanja prijave na drugi ured
 3. Djelatnik odabire gradski ured kojemu želi proslijediti prijavu i potvrđuje unos

UC16 - Objedinjenje nepovezanih prijava

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Objedinjavanje prijava istog oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik gradskog ureda obrađuje prijavu, prijava ima status *na čekanju*
- **Opis osnovnog tijeka:**
 1. Djelatnik procjenjuje da postoji više prijava istog oštećenja
 2. Djelatnik odabire opciju objedinjavanja prijava
 3. Djelatnik odabire prijave koje želi objediniti i potvrđuje odabir

UC17 - Slanje povratnih informacija

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Informiranje prijavitelja o promjenama u njegovoj prijavi
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik gradskog ureda je prilikom obrade prijave izvršio neku radnju/promjenu nad prijavom
- **Opis osnovnog tijeka:**
 1. Aplikacija šalje email kojim obavještava prijavitelja o promjenama vezanim uz prijavu
- **Opis mogućih odstupanja:**
 - 1.a Prijava je anonimna
 1. Aplikacija ne radi ništa

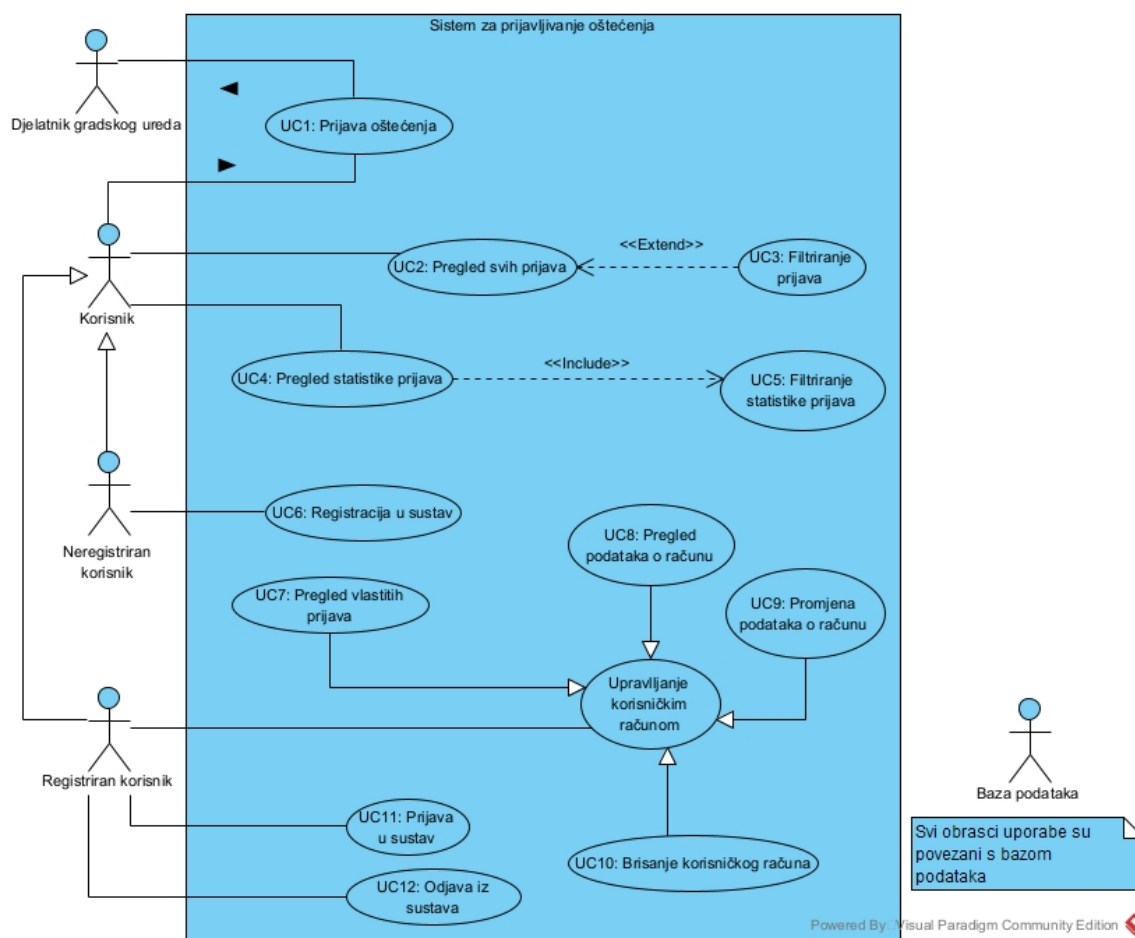
UC18 - Pregled prijava gradskog ureda

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Pregledavanje prijava koje gradski ured obrađuje/treba obraditi/je obradio
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Djelatnik gradskog ureda je na glavnoj stranici za Djelatnike gradskog ureda
 2. Aplikacija prikazuje prijave iz liste novopristiglih prijava

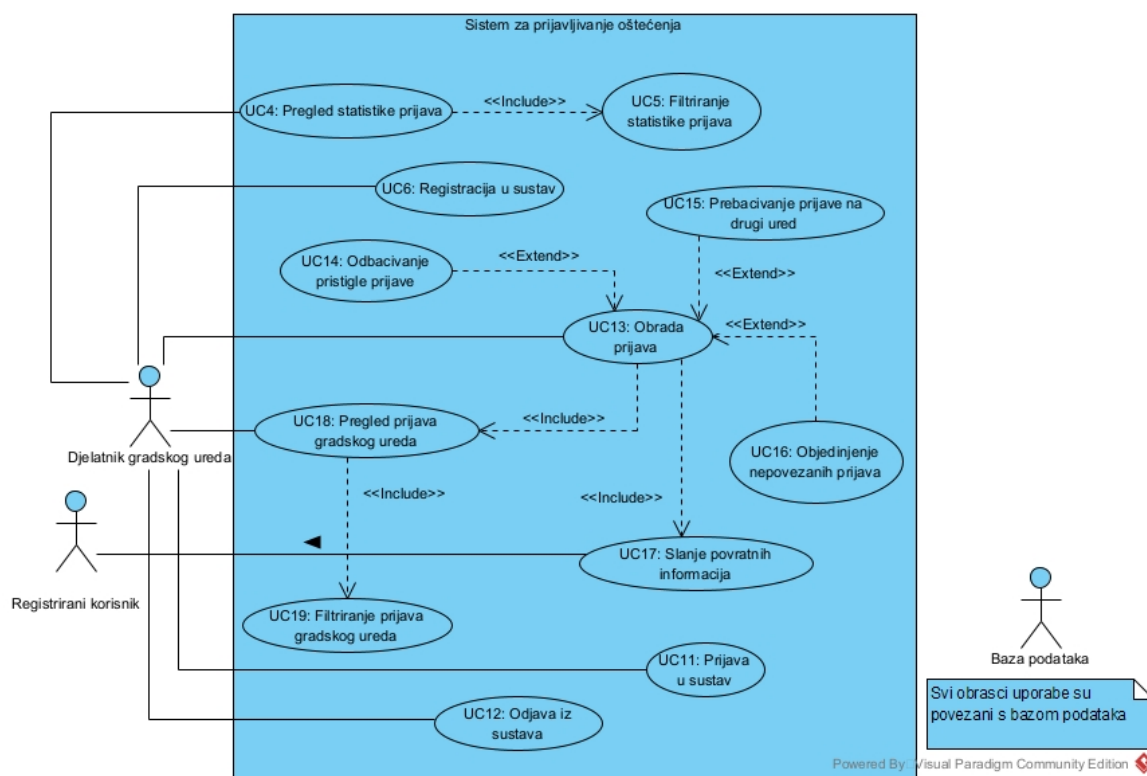
UC19 - Filtriranje prijava gradskog ureda

- **Glavni sudionik:** Djelatnik gradskog ureda
- **Cilj:** Filtriranje prijava koje gradski ured mora obraditi/obrađuje/je obradio
- **Sudionici:** Baza podataka
- **Preduvjet:** Djelatnik gradskog ureda pregledava prijave
- **Opis osnovnog tijeka:**
 1. Djelatnik gradskog ureda odabire listu prijava prema statusu prijava
 2. Aplikacija prikazuje odabranu listu prijava
 3. Djelatnik unosi dodatne parametre za filtriranje
 4. Aplikacija prikazuje filtriranu listu prijava
- **Opis mogućih odstupanja:**
 - 3.a Djelatnik ne želi dodatno filtrirati prijave
 1. Preskače se dodatno filtriranje

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnosti korisnika



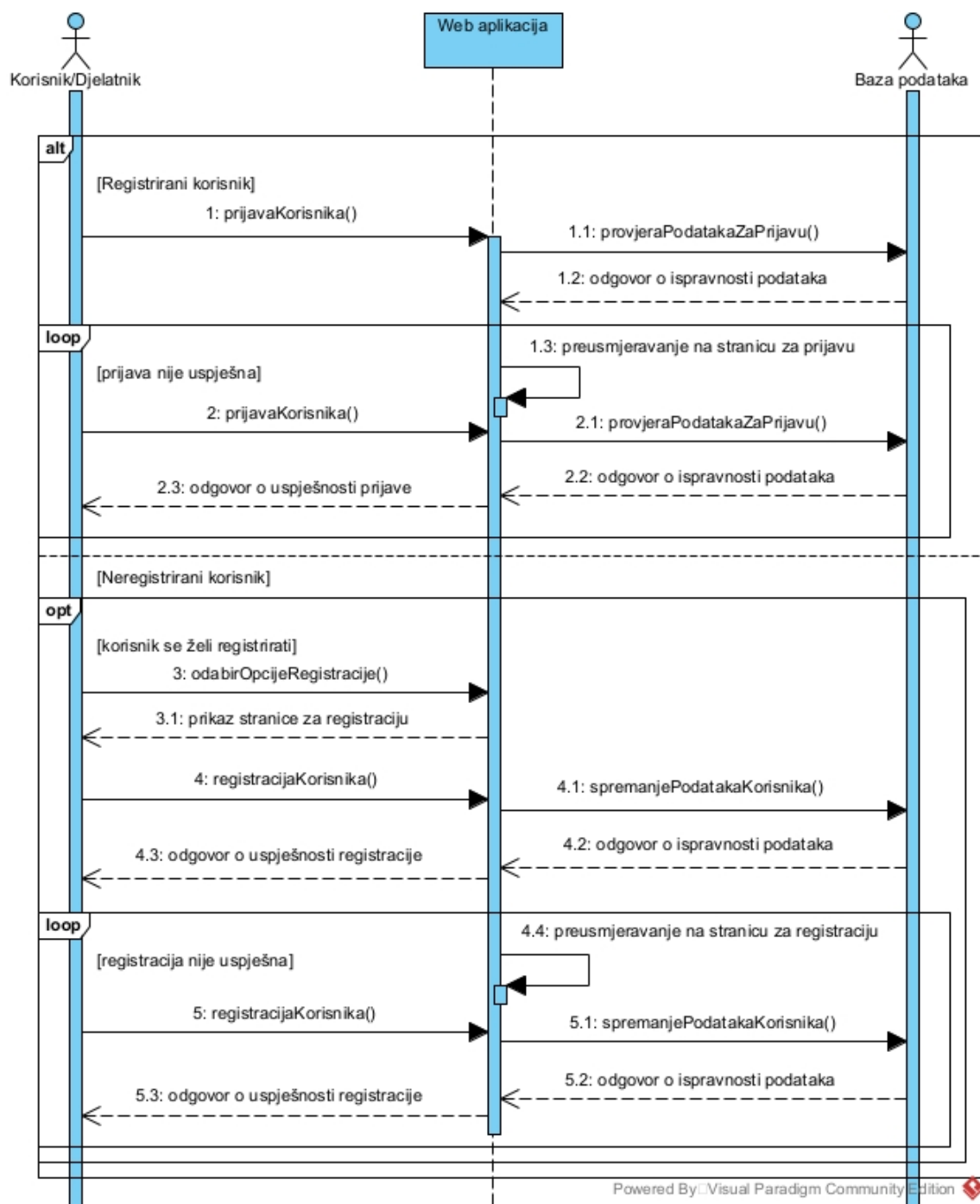
Slika 3.2: Dijagram obrasca uporabe, obrada prijava

3.1.2 Sekvencijski dijagrami

dio 1. revizije

Obrasci uporabe - UC06, UC11 - Registracija u sustav, Prijava u sustav

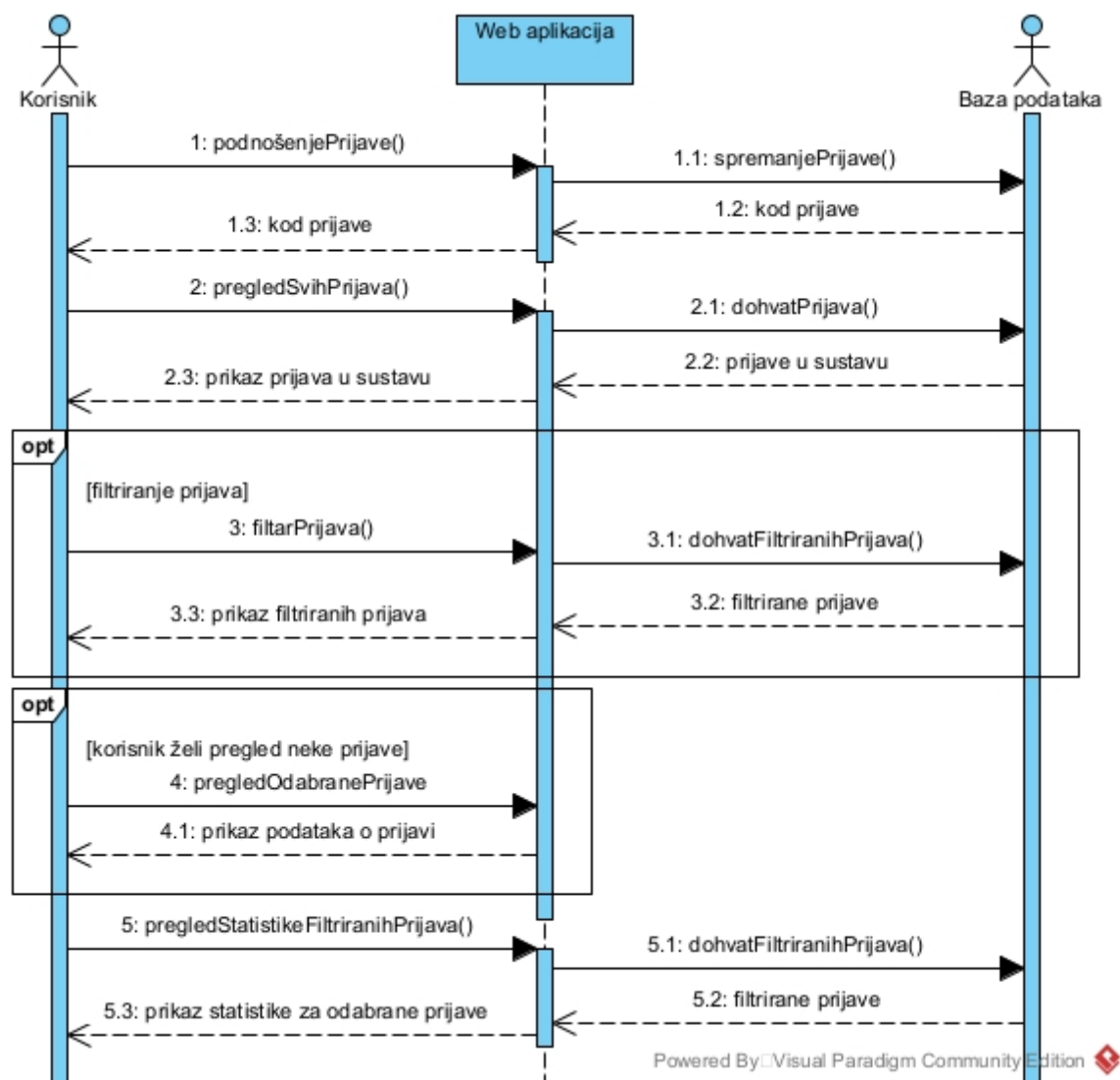
Korisnik(može biti i djelatnik) na naslovnoj stranici odabire opciju prijave ili registracije te mu se zatim otvara prikladna stranica. Ako se radi o stranici za prijavu, korisnik unosi svoj email i zaporku i potvrđuje unos. Ako su uneseni podaci validni korisniku se prikazuje naslovna stranica za prijavljenog korisnika, odnosno za djelatnika gradskog ureda. U suprotnom se korisnika obavještava o grešci i ponovno traži unos podataka za prijavu. Ako se radi o stranici za registraciju, korisnik mora unijeti svoje ime, prezime, datum rođenja, email adresu i lozinku. Ako se registrira novi gradski ured, potrebno je označiti tu opciju. U tom slučaju se traži unos imena ureda, email adresa i lozinka. Sustav provjerava u bazi ako za danu email adresu već postoji korisnički račun. Ako račun već postoji, sustav korisniku javlja grešku i ponovno traži unos podataka za registraciju. Ako je unos dobar, korisniku se prikazuje naslovna stranica za prijavljenog korisnika, odnosno za djelatnika gradskog ureda. Korisnik ne mora odabrati opciju prijave ili registracije. U tom slučaju koristi stranicu kao anonimni neregistrirani korisnik. Djelatnik se mora prijaviti ili registrirati ured da bi obavljao svoju funkcionalnost.



Slika 3.3: Sekvencijski dijagram, prijava/registracija

Obrasci uporabe - UC1, UC2, UC3, UC4, UC5 - Prijava oštećenja, Pregled svih prijava, Filtriranje prijava, Pregled statistike prijava, Filtriranje statistike prijava

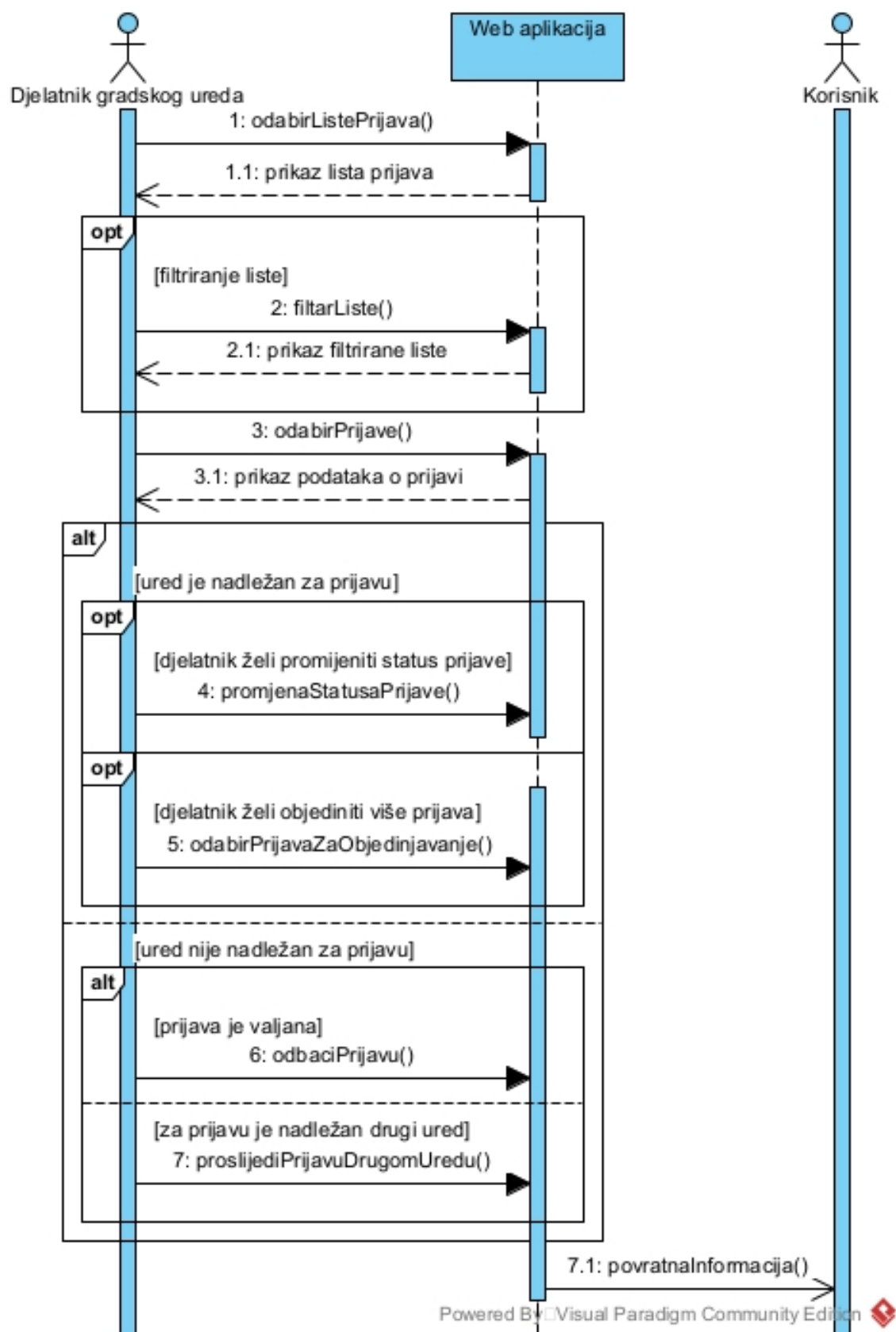
Korisnik odabire opciju podnošenja nove prijave oštećenja. U prijavi korisnik upisuje kratak naslov, opis oštećenja, lokaciju i kategoriju. Korisnik također može priložiti sliku oštećenja. Lokaciju korisnik može priložiti preko interaktivne karte, upisom adrese ili se lokacija može izvući iz metapodataka slike. Aplikacija će sama pokušati prepoznati kategoriju oštećenja i na temelju toga i ponuditi nadležan ured. Nakon podnošenja prijave korisnik dobiva kod prijave preko koje može pratiti njezino stanje. Korisnik odabirom pregleda prijava može pregledavati postojeće aktive prijave u sustavu preko interaktivne karte. Prijave može dodatno filtrirati po vremenu, gradskom uredu i kategoriji. Odabirom neke prijave na karti korisniku se prikazuju podaci o toj prijavi. Korisnik odabirom statistike prijava također može pregledavati statistiku za prijave filtrirane po lokaciji, vremenu i gradskom uredu. U statistici se nalaze podaci o broju prijava svakog statusa, prosječno vrijeme koje je prijava imala neki status, prosječan broj prijava u danu i ukupan broj prijava.



Slika 3.4: Sekvencijski dijagram, funkcionalnosti korisnika

Obrasci uporabe - UC13, UC14, UC15, UC16, UC17, UC18, UC19 - Obrada prijava, Odbacivanje pristigle prijave, Prebacivanje prijave na drugi ured, Objedinjenje nepovezanih prijava, Slanje povratnih informacija, Pregled prijava gradskog ureda, filtriranje prijava gradskog ureda

Djelatnik gradskog ureda odabire jednu od 3 lista prijava, jedna za svaki status. Djelatnik može dodatno filtrirati prijave u listi po vremenu i kategoriji. Odabirom jedne od prijava djelatniku se prikazuju podaci o toj prijavi. Djelatnik dalje može promijeniti status prijave. Ako vidi da je oštećenje u toj prijavi već prijavljeno, djelatnik može odabrati sve takve prijave i objediniti ih. Ako utvrdi da njegov ured nije nadležan za ovu prijavu, djelatnik odabire nadležan ured i prosljeđuje mu prijavu. Ako je prijava u potpunosti nevaljana djelatnik ju može odbaciti. Bilo koja od ovih radnji nad prijavom rezultirati će slanjem povratne informacije korisniku koji je podnio prijavu, ako je taj korisnik registriran.



Slika 3.5: Sekvencijski dijagram, obrada prijava

3.2 Ostali zahtjevi

dio 1. revizije

- Sustav treba podržati rad više korisnika u stvarnom vremenu
- Aplikacija podržava korisničko sučelje na hrvatskom i engleskom jeziku
- Sustav treba biti implementiran kao web aplikacija sa responzivnim dizajnom
- Rad i veza s bazom podataka moraju biti sigurni i pouzdani
- Aplikacija mora koristiti HTTPS protokol
- Aplikacija mora biti jednostavna za korištenje
- Neispravno korištenje korisničkog sučelja ne smije narušiti cijeloukupan rad sustava
- Izvršavanje bilo koje akcije na aplikaciji ne smije trajati dulje od 5 sekundi
- Osjetljivi podaci, kao što su lozinke korisnika, se u bazi moraju spremati u kriptiranom obliku

4. Arhitektura i dizajn sustava

dio 1. revizije

Arhitektura ovog sustava je bazirana na arhitekturi klijent-poslužitelj. Sustav se sastoji od 3 sloja: sloj korisničkog sučelja, sloj aplikacijske logike, sloj podataka.

Korisnik koristi aplikaciju preko web preglednika. Web preglednik je program koji omogućuje klijentu komunikaciju s web poslužiteljem aplikacije. Preglednik nam omogućava prikaz sloja korisničkog sučelja sustava. Korisnik preko web preglednika komunicira s web poslužiteljem slanjem i primanjem HTTP zahtjeva. Primljene podatke i datoteke web preglednik zna interpretirati i prikazati korisniku tako da sama interakcija s aplikacijom bude jednostavna.

Web poslužitelj je računalo na kojem se aplikacija pokreće. Na njemu se dakle nalazi sloj aplikacijske logike. Web poslužitelj aplikaciji proslijeđuje zahtjeve na obradu i njezine odgovore proslijeđuje natrag klijentima. Web aplikacija na poslužitelju obrađuje zaprimljene zahtjeve. Ako obrada zahtjeva to zahtjeva, web aplikacija dodatno komunicira sa slojem podataka.

Sloj podataka je predstavljen bazom podataka. U bazi podataka se na siguran način spremaju svi podaci koje je potrebno trajno čuvati. To podrazumijeva podatke o prijavama, korisničkim računima i slično. Takvi podaci moraju ostati očuvani i ako je rad aplikacije prekinut iz bilo kojeg razloga. Baza podataka je detaljnije opisana u poglavlju 4.1.

Web aplikacija je podijeljena na frontend i backend.

Frontend je zapravo prezentacijski dio aplikacije. On zapravo oblikuje korisničko sučelje i samim time definira što će korisnik vidjeti na web pregledniku kada koristi aplikaciju.

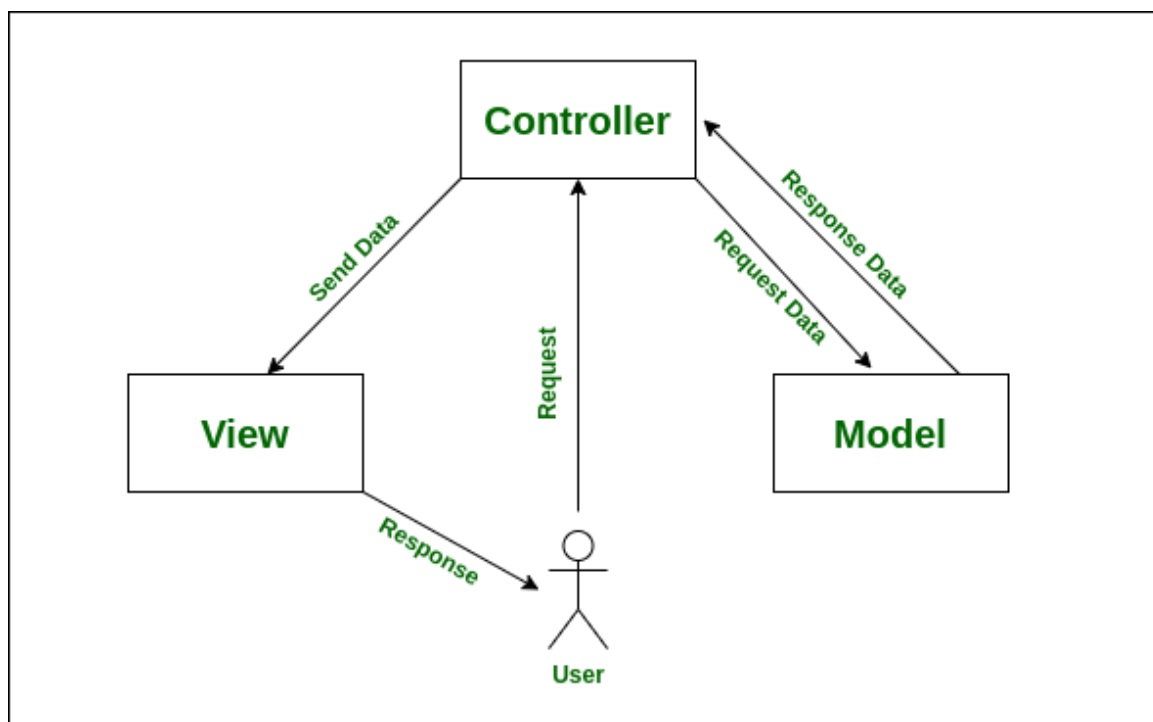
Backend je dio aplikacije koji obrađuje zahtjeve i kontrolira rad ostalih dijelova sustava.

Arhitektura same web aplikacije je odrađena u stilu MVC arhitekture. U ovoj arhitekturi aplikacija se dijeli na 3 sloja, svaki sa svojom ulogom. Ovime se postiže podjela odgovornosti pojedinih komponenti sustava. Sama struktura aplikacije je preglednija i smislenije primjenom ove arhitekture. Također, razvoj i održavanje

sustava je znatno olakšan primjenom ove arhitekture. MVC arhitektura je izuzetno popularna u razvoju web aplikacija te je stoga dobro razrađena i mnogi radni okviri podržavaju njezinu implementaciju. MVC arhitektura se sastoji od slojeva:

- **Model** - predstavlja komponentu aplikacije koja modelira aplikacijsku logiku. Ovaj model preslikava podatke dobivene od korisnika u bazu podataka i obrnuto. Na temelju zaprimljenih podataka vrši obradu zahtjeva.
- **Pogled(View)** - predstavlja dio aplikacije zadužen za prikaz podataka korisniku. Također omogućava korisniku interakciju s aplikacijom kroz prikaz korisničkog sučelja. Usko je povezan s frontend dijelom aplikacije.
- **Upravljač(Controller)** - upravlja nadolazećim zahtjevima. Ova komponenta proslijeđuje zahtjeve modelu na obradu. Od modela zaprima odgovore koje proslijeđuje komponenti za pogled na formiranje grafičkog prikaza koje se zatim prikazuje korisniku.

Kroz međusobnu interakciju ova 3 modela omogućuju normalno funkcioniranje aplikacije.



Slika 4.1: prikaz komponenti u MVC arhitekturi

Za razvoj frontend dijela aplikacije odlučili smo koristiti programski jezik JavaScript uz radni okvir React. Backend dio aplikacije je realiziran u programskom

jeziku Java uz radni okvir Spring Boot.

Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremišta podataka, mrežne protokole, globalni upravljački tok i sklopovsko-programске zahtjeve. Po točkama razraditi i popratiti odgovarajućim skicama:

- *izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zašto ste baš odabrali takvu arhitekturu)*
- *organizaciju sustava s najviše razine apstrakcije (npr. klijent-poslužitelj, baza podataka, datotečni sustav, grafičko sučelje)*
- *organizaciju aplikacije (npr. slojevi frontend i backend, MVC arhitektura)*

4.1 Baza podataka

dio 1. revizije

Ovaj sustav će za svoje potrebe koristiti relacijsku bazu podataka implementirana u PostgreSQL-u. Ovakav tip baze podataka svojom strukturom omogućuje jednostavno modeliranje elemenata iz stvarnog svijeta i zbog toga je široko primjenjiv. Odabrali smo implementaciju u PostgreSQL-u jer smo s ovom implementacijom već dobro upoznati. Relacijske baze podataka se sastoje od relacija. Relacije su predstavljene tablicama koje su definirane svojim nazivom i skupom atributima. Bazu podataka koristimo za jednostavnu i sigurnu pohranu, izmjenu, umećanje i dohvat podataka potrebnih za rad sustava. Naša baza podataka se sastoji od sljedećih relacija, odnosno tablica:

- Users
- Report
- Image
- Category
- CategoryKeywords
- ReportGroup
- Feedback
- CityOffice

Potrebno je opisati koju vrstu i implementaciju baze podataka ste odabrali, glavne komponente od kojih se sastoji i slično.

4.1.1 Opis tablica

Users tablica čuva podatke o korisničkim računima koje korisnici izrađuju tijekom registracije. Tablica je povezana vezom *One-to-Many* s tablicom *Report* preko atributa *userID*.

Tablica 4.1:

Users		
userID	INT	jedinstveni identifikator korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Users		
email	VARCHAR	email adresa korisnika
firstName	VARCHAR	ime korisnika
lastName	VARCHAR	prezime korisnika
password	VARCHAR	kriptirana lozinka za korisnički račun

Report tablica čuva podatke o pojedinim prijavama oštećenja koje korisnici podnose. Tablica je povezana vezom *Many-to-One* s tablicom *Users* preko atributa *userID*, *Many-to-One* vezom s tablicom *Category* preko atributa *categoryID* i *Many-to-One* vezom s tablicom *ReportGroup* preko atributa *groupID*.

Tablica 4.2:

Report		
reportID	INT	jedinstveni identifikator prijave
location	VARCHAR	lokacija oštećenja
description	VARCHAR	opis oštećenja
reportTS	TIMESTAMP	vrijeme prijave
reportHeadline	VARCHAR	naslov prijave
userID	INT	jedinstveni identifikator korisnika koji je podnio prijavu
categoryID	INT	jedinstveni identifikator kategorije oštećenja
groupID	INT	jedinstveni identifikator grupe prijave

Image tablica čuva podatke o slikama koje se prilažu u prijavama oštećenja. Tablica je povezana *Many-to-One* vezom s tablicom *Report* preko atributa *reportID*.

Tablica 4.3:

Image		
imageID	INT	jedinstveni identifikator slike

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Image		
reportID	INT	jedinstveni identifikator prijave kojoj slika pripada
URL	VARCHAR	URL slike

Category tablica čuva podatke o kategorijama kojima oštećenje može pripadati. Tablica je povezana *One-to-Many* vezom s tablicom *Report* preko atributa *categoryID*, *Many-to-One* vezom s tablicom *CityOffice* preko atributa *cityOfficeID* i *One-to-Many* vezom s tablicom *CategoryKeywords* preko atributa *categoryID*.

Tablica 4.4:

Category		
categoryID	INT	jedinstveni identifikator kategorije
categoryName	VARCHAR	naziv kategorije
cityOfficeID	INT	jedinstveni identifikator gradskog ureda koji obrađuje prijave oštećenja iz ove kategorije

CategoryKeywords tablica čuva podatke o ključnim riječima po kojima se može identificirati kategorija oštećenja. Tablica je povezana *Many-to-One* vezom s tablicom *Category* preko atributa *categoryID*.

Tablica 4.5:

CategoryKeywords		
keywordID	INT	jedinstveni identifikator ključne riječi
keyword	VARCHAR	ključna riječ
categoryID	INT	jedinstveni identifikator kategorije kojoj ključna riječ pripada.

ReportGroup tablica predstavlja grupu prijava koje su objedinjene. Tablica je povezana *One-to-Many* vezom s tablicom *Report* preko atributa *groupID* i *One-to-Many* vezom s tablicom *Feedback* preko atributa *groupID*.

Tablica 4.6:

ReportGroup		
groupID	INT	jedinstveni identifikator grupe prijava

Feedback tablica čuva podatke o statusu pojedine grupe prijava. Također čuva podatke potrebne za slanje povratnih informacija korisnicima i računanje nekih statistika. Tablica je povezana *Many-to-One* vezom s tablicom *ReportGroup* preko atributa *groupID* i *Many-to-One* vezom s tablicom *CityOffice* preko atributa *cityOfficeID*.

Tablica 4.7:

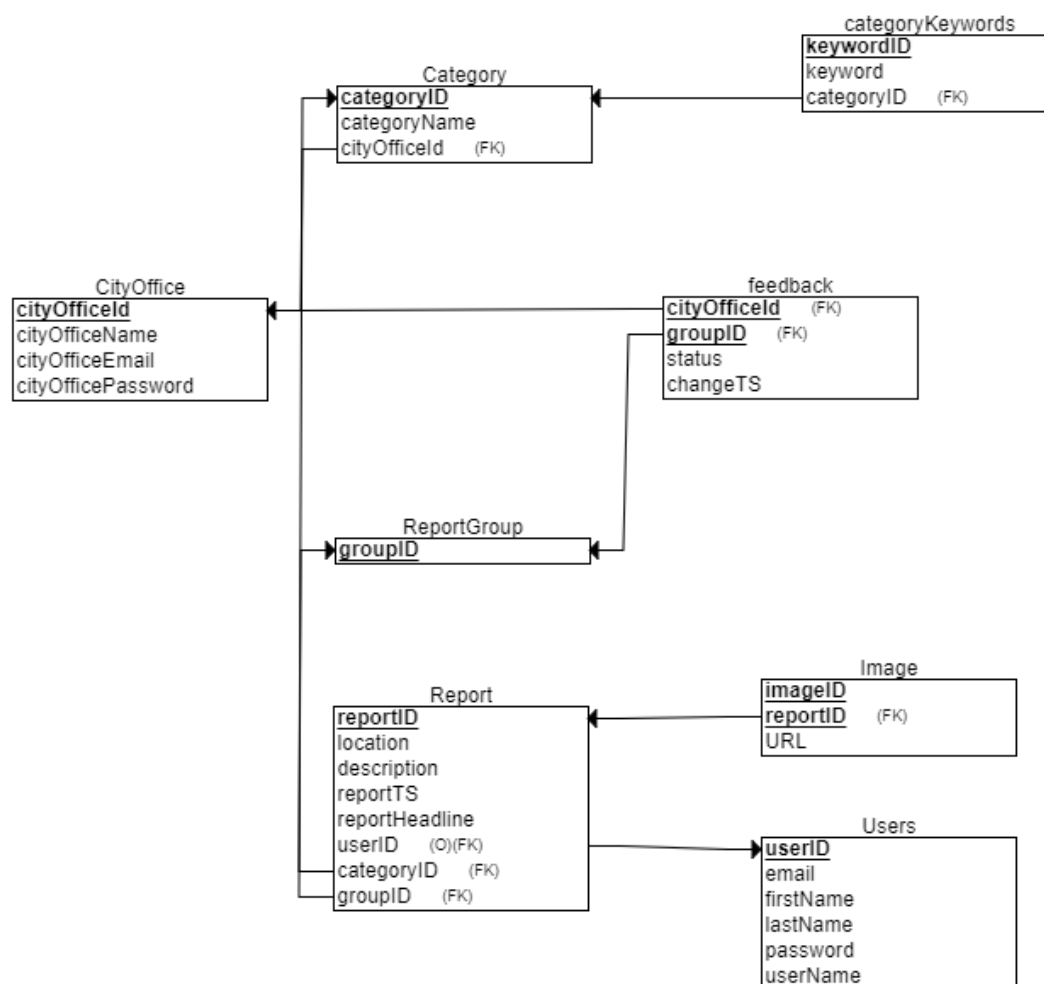
Feedback		
cityOfficeID	INT	jedinstveni identifikator gradskog ureda
groupID	INT	jedinstveni identifikator grupe prijava na koju se podaci referiraju
status	VARCHAR	status prijava u referiranoj grupi
changeTS	TIMESTAMP	vrijeme kada se postavio status prijava iz referirane grupe

CityOffice tablica čuva podatke o računima gradskih ureda. Tablica je povezana *One-to-Many* vezom s tablicom *Feedback* preko atributa *cityOfficeID* i *One-to-Many* vezom s tablicom *Category* preko atributa *cityOfficeID*.

Tablica 4.8:

CityOffice		
cityOfficeID	INT	jedinstveni identifikator gradskog ureda
cityOfficeName	VARCHAR	naziv gradskog ureda
cityOfficeEmail	VARCHAR	email adresa gradskog ureda
cityOfficePassword	VARCHAR	kriptirana lozinka za račun gradskog ureda

4.1.2 Dijagram baze podataka



Slika 4.2: relacijski dijagram baze podataka

4.2 Dijagram razreda

Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.

dio 1. revizije

*Prilikom prve predaje projekta, potrebno je priložiti potpuno razrađen dijagram razreda vezan uz **generičku funkcionalnost** sustava. Ostale funkcionalnosti trebaju biti idejno razrađene u dijagramu sa sljedećim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazivi atributa razreda, veze i odnosi između razreda.*

dio 2. revizije

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

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

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

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

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

5.2 Ispitivanje programskog rješenja

dio 2. revizije

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

5.2.1 Ispitivanje komponenti

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

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

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

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

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

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

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

5.4 Upute za puštanje u pogon

dio 2. revizije

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

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

6. Zaključak i budući rad

dio 2. revizije

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

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

Popis literature

Kontinuirano osvježavanje

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

1. GeeksForGeeks, MVC Design Pattern, <https://www.geeksforgeeks.org/mvc-design-pattern/>
2. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
3. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
4. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
5. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
6. The Unified Modeling Language, <https://www.uml-diagrams.org/>
7. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Aplikacija <i>Gradsko oko</i> Grada Rijeke	6
2.2	Primjer slike s potpisom	11
2.3	Primjer slike s potpisom 2	12
3.1	Dijagram obrasca uporabe, funkcionalnosti korisnika	22
3.2	Dijagram obrasca uporabe, obrada prijava	23
3.3	Sekvencijski dijagram, prijava/registracija	25
3.4	Sekvencijski dijagram, funkcionalnosti korisnika	27
3.5	Sekvencijski dijagram, obrada prijava	29
4.1	prikaz komponenti u MVC arhitekturi	32
4.2	relacijski dijagram baze podataka	38

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.

1. sastanak

- Datum: u ovom formatu: 23. listopada 2023.
- Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglič, Eno Peršić, Bruno Mikulan, Filip Vučenik
- Teme sastanka:
 - napravljen git repozitorij
 - rasprava o zadatku - definirani ključni funkcijski i nefunkcijski zahtjevi, razmjena ideja oko aplikacije
 - podjela poslova (pisanje opisa zadatka, izrada UML dijagrama, zapisivanje zahtjeva, raspodjela timova)

2. sastanak

- Datum: u ovom formatu: 30. listopada 2023.
- Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglič, Eno Peršić, Bruno Mikulan, Filip Vučenik
- Teme sastanka:
 - dizajn baze podataka
 - dizajn korisničkog sučelja aplikacije
 - detaljnija rasprava o svojstvima aplikacije

3. sastanak

- Datum: u ovom formatu: 6. studenoga 2023.
- Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglič, Eno Peršić, Bruno Mikulan, Filip Vučenik
- Teme sastanka:
 - izvještaj svakog dijela tima o svojem napretku

– rad na dizajnu aplikacije

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Petar Belošević	Vinko Brkić	Tomislav Grudić	Fran Meglić	Bruno Mikulan	Eno Peršić	Filip Vučenik
Upravljanje projektom	2						
Opis projektnog zadatka	4						1
Funkcionalni zahtjevi	1		1				
Opis pojedinih obrazaca	1						
Dijagram obrazaca	1				2		
Sekvencijski dijagrami	1	1		1			
Opis ostalih zahtjeva	1					1	
Arhitektura i dizajn sustava							
Baza podataka			2				
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Petar Belošević	Vinko Brkić	Tomislav Grudić	Fran Meglić	Bruno Mikulan	Eno Peršić	Filip Vučenik
Dnevnik sastajanja	1						
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>rad na frontendu</i>		2			2	2	
<i>rad na backendu</i>			2	2			2
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>usvajanje korištenih tehnologija</i>	3	3	3	3	3	3	3

Dijagrami pregleda promjena

dio 2. revizije

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