Programsko inženjerstvo

Ak. god. 2023./2024.

Prijava oštećenja

Dokumentacija, Rev. *1*

Grupa: *SourceresOfTheNorth*

Voditelj: *Petar Bešević*

Datum predaje: *17.11.2023.*

Nastavnik: *Vlado Sruk*

Sadržaj

1. Dnevnik promjena dokumentacije

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.
0.6	Dodan opis arhitekture sustava.	Petar Belošević	8.11.2023.
0.7	Opis baze podataka	Petar Belošević	9.11.2023.
0.8	Dodan dijagram razreda s opisom, izmjenjen opis arhitekture sustava.	Petar Belošević	16.11.2023.
1.0	Završna verzija za prvu kontrolnu točku.	Petar Belošević	17.11.2023.
1.1	Popravci grešaka iz prve kontrolne točke.	Petar Belošević	10.12.2023.
1.2	Dodan dijagram stanja s pripadnim opisom.	Petar Belošević	16.12.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.3	Dodan dijagram aktivnosti s pripadnim opisom.	Petar Belošević	18.12.2023.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije *"StreetPatrol"*. Ideja aplikacije je ponuditi običnim građanima jednostavan i efikasan način prijave oštećenja javnih površina i cesta u gradu te dobivanje povratnih informacija o 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.

Građani aplikaciju mogu koristiti kao registrirani i kao neregistrirani korisnici. Građani imaju mogućnost 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
- email adresa
- lozinka

Prijavljeni korisnici prilikom pregleda svojeg profila mogu vidjeti prije navedene podatke o sebi koje mogu i izmijeniti. 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, kratak opis, lokaciju i kategoriju oštećenja. Opcionalno se u prijavi može priložiti fotografija oštećenja. Lokacija oštećenja se prilaže odabirom položaja na karti ili upisivanjem adrese na kojoj se nalazi oštećenje. Ako korisnik priloži fotografiju, aplikacija će pokušati iz nje izvući podatke o lokaciji

te ih ponuditi korisniku kao lokaciju oštećenja. Odabirom kategorije oštećenja korisnik indirektno odabire gradski ured nadležan za konkretan problem. Prilikom opisivanja oštećenja aplikacija će pokušati sama zaključiti kojoj kategoriji oštećenje pripada te će istu predložiti korisniku. Također, ako u sustavu već postoji slična nedavna prijava na istoj lokaciji aplikacija će predložiti korisniku postojeću prijavu i ponuditi mu da svoju prijavu nadoveže na 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.

Za pregled postojećih prijava korisnik koristi interaktivnu kartu s označenim prijavama na njoj. Prijave se mogu dodatno filtrirati po vremenu podnošenja prijave, statusu, kategoriji oštećenja i nadležnom gradskom uredu. Korisnik također može pretraživati prijave po jedinstvenom kodu prijave. Na taj način svaki korisnik može pregledavati svoje prijave na aplikaciji. Odabirom neke prijave prikazuju se podaci iz podnesenog obrasca o odabranoj prijavi. Također se prikazuje status prijave (*na čekanju*, *u procesu rješavanja*, *riješena*), 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. 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 vremenu podnošenja prijave, kategoriji oštećenja i nadležnom gradskom uredu. Za odabrane parametre aplikacija prikazuje:

- ukupan broj podnesenih 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 pregledavaju prijave kroz 3 liste, jedna za svaku vrstu statusa: novopristigle, one u procesu rješavanja i riješene prijave. Djelatnici ureda imaju isključivo pristup prijavama za oštećenja za koja je nadležan njihov ured.

Ostale prijave djelatnicima nisu vidljive.

Sustav nudi mogućnost registracije novih gradskih ureda. U tom slučaju kod stranice za registraciju potrebno je odabrati opciju registracije gradskog ureda. Aplikacija tada prikazuje obrazac za koji je potrebno navesti:

- ime gradskog ureda
- email adresa gradskog ureda
- lozinka za račun gradskog ureda
- jedna ili više kategorija koje će novi gradski ured pokrivati

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 obavijest 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 obavijest prijavitelju ako se radi o registriranom korisniku.

Prethodne dvije radnje bi trebale sadržavati komunikaciju s nadležnim gradskim službama. Dakle, po primitku prijave djelatnik promjenom statusa prijave delegira prijavu nadležnoj gradskoj službi. Kada nadležna služba riješi problem šalje informaciju uredu koji onda može prijavu označiti kao riješenu. Ova funkcionalnost nije implementirana jer zahtjeva komunikaciju s vanjskim sustavima što nije u opsegu ovog projekta.

Djelatnik ima mogućnost objedinjavanja prijava. Naime, ako se pristigla prijava referira na problem za koji već postoji prijava, djelatnik može grupirati te prijave. Aplikacija u tom slučaju šalje prikladnu obavijest registriranim korisnicima čije su prijave zahvaćene ovom radnjom.

Ako procijeni da je pristigla prijava poslana na krivi ured, djelatnik ima mogućnost proslijediti prijavu uredu koji je zapravo nadležan za pristiglu prijavu. Korisniku koji je podnio prijavu se u tom slučaju šalje prikladna obavijest putem maila ako je taj korisnik registriran.

Slično, ako procijeni da pristigla prijava uopće nije valjana, djelatnik ima mogućnost odbaciti prijavu. U tom slučaju se prijava arhivira u bazi podataka ali se više ne koristi. Korisnik koji je podnio prijavu dobiva prikladnu obavijest putem maila ako je registriran.

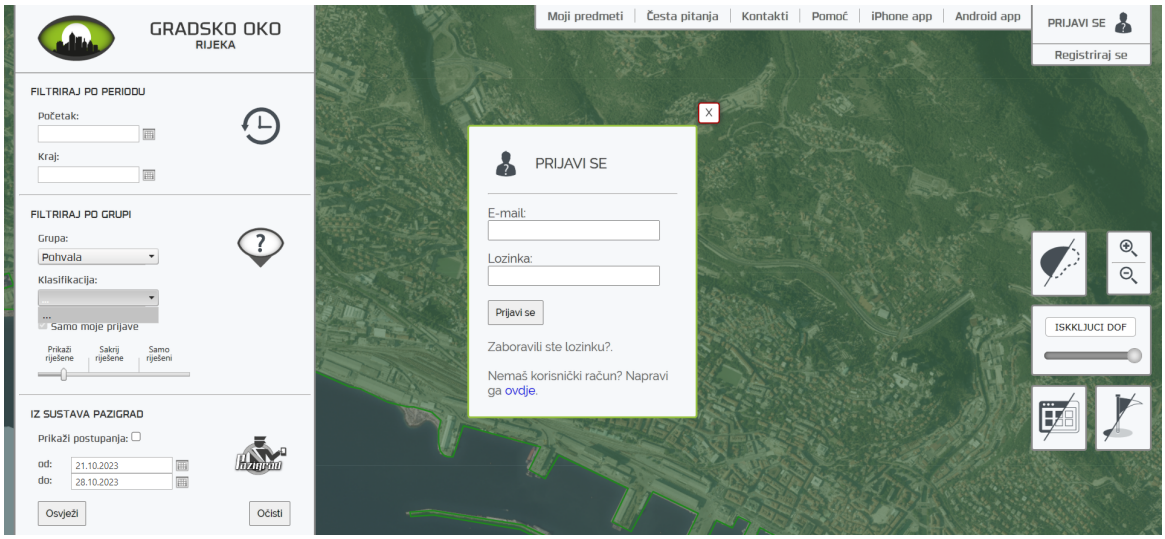
Djelatnici gradskih ureda mogu pregledavati statistiku prijava kao i obični korisnici, uz restrikciju na prijave vezane na njihov ured, kao što je prije navedeno.

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

Aplikacija je napravljena na primjeru Grada Zagreba. Karte u aplikaciji će prikazivati područje Grada Zagreba i prijave će se moći podnositi samo za oštećenja u tom području. Aplikacija nema fiksirani broj gradskih ureda, već je omogućeno dodavanje novih ureda, Kao što je već i navedeno. Time je aplikaciju lakše prilagoditi promjenama u organizaciji gradskih ureda. Također, na ovaj način je lakše konfigurirati aplikaciju za implementaciju u nekom drugom području osim Grada Zagreba.

2.1 Postojeća rješenja za slični problem

Slične aplikacije već postoje u Hrvatskoj. Tako primjerice Grad Rijeka na svojim stranicama nudi mogućnost prijave raznih prekršaja i oštećenja (`https://gov.rijeka.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://rijeka.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

2.2 Mogućnosti nadogradnje i skaliranja rješenja

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 na to 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 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 infrastrukturalnom podrškom, prvenstveno u obliku velikih poslužitelja koji mogu obrađivati puno zahtjeva i pohranjivati velike količine podataka. Također, aplikacija bi na toj razini zahtijevala dobru suradnju i koordinaciju velikog broja institucija u više različitih država što može biti zahtjevno.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Pobrojani funkcionalni zahtjevi:

- F01: registracija novog korisnika
- F02: prijava korisnika
- F03: podnošenje prijave oštećenja (naslov, opis, lokacija, slika - opcionalno, kategorija)
- F04: automatsko prepoznavanje kategorije prijave
- F05: mogućnost automatskog predlaganja nadovezivanja prijave na postojeću
- F06: pregledavanje podataka o korisničkom računu prijavljenog korisnika
- F07: uređivanje podataka o korisničkom računu prijavljenog korisnika
- F08: pregledavanje prijava prijavljenog korisnika
- F09: pregledavanje svih aktivnih prijava u sustavu preko interaktivne karte
- F10: mogućnost filtriranja prijava u sustavu prilikom pregledavanja
- F11: pregled statistike prijava
- F12: filtriranje statistike prijava
- F13: djelatnik gradskog ureda može odbaciti nevažeću prijavu
- F14: djelatnik gradskog ureda može proslijediti prijavu drugom uredu
- F15: djelatnik gradskog ureda može mijenjati status prijave
- F16: djelatnik gradskog ureda može objediniti prijave istog oštećenja
- F17: povratna informacija korisniku (ako je registriran) za svaku promjenu nad prijavom
- F18: pregled prijava ureda kroz 3 liste (jedna za svaki status) za djelatnike gradskog ureda
- F19: dodatno filtriranje liste prijava za djelatnike gradskog ureda
- F20: odjava iz sustava
- F21: brisanje korisničkog računa

Dionici:

1. Gradska uprava
2. Djelatnici gradskih ureda
3. Gradske službe
4. Građani
   - Registrirani korisnici
   - Neregistrirani korisnici
5. 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 i lozinka
   (b) podnositi anonimnu prijavu koja sadrži naslov, opis, lokaciju i kategoriju oštećenja, a opcionalno i fotografiju i prijavu na koju se nadovezuje
   (c) pregledavati postojeće prijave preko karte te ih filtrirati po vremenu, statusu, kategoriji oštećenja i nadležnom gradskom uredu
   (d) pregledavati statistiku postojećih prijava za odabran period, kategoriju oštećenja 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 kategoriju oštećenja, a opcionalno i fotografiju i prijavu na koju se nadovezuje
   (e) pregledavati postojeće prijave preko karte te ih filtrirati po vremenu, statusu, kategoriji oštećenja i nadležnom gradskom uredu
   (f) pregledavati statistiku postojećih prijava za odabran period, kategoriju oštećenja i nadležan gradski ured
3. Djelatnik gradskog ureda (inicijator) može:
   (a) pregledavati prijave pristigle u njihov ured prema njihovom statusu
   (b) dodatno filtrirati prijave prilikom pregleda
   (c) prijavama sa statusom *na čekanju* mijenjati status u *u procesu rješavanja*
   (d) prijavama sa statusom *u procesu rješavanja* mijenjati status u *riješena*

(e) objediniti prijave ako se referiraju na isti problem
(f) pregledavati statistiku prijava pristiglih u njihov ured
(g) dodatno filtrirati za koje prijave želi pregledati statistiku

4. Baza podataka (sudionik) može:
   (a) spremati sve podatke o korisnicima
   (b) spremati sve podatke o prijavama

3.1.1 Obrasci uporabe

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
  3. Podaci o novom računu se spremaju u bazu podataka
- **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:** Korisnik pregledava svoj korisnički račun
- **Opis osnovnog tijeka:**
  1. Registrirani korisnik prilikom pregleda korisničkog računa odabire opciju pregleda vlastitih prijava
  2. Aplikacija prikazuje listu dosadašnjih prijava Registriranog korisnika
  3. Registrirani korisnik odabire prijavu
  4. Prikazuju se podaci o prijavi
- **Opis mogućih odstupanja:**

  3.a Registrirani korisnik ne želi pregledati konkretnu prijavu
  1. Korisnik nakon pregleda liste prijava izlazi iz opcije pregleda vlastitih prijava

UC8 - Pregled korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled vlastitog korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Registrirani korisnik odabire ikonu svojeg profila
  2. Aplikacija prikazuje podatke o korisničkom računu i dodatne opcije
  3. Registrirani korisnik pregledava podatke o svojem korisničkom računu

UC9 - Promjena podataka o računu

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promjena podataka o korisničkom računu
- **Sudionici:** Baza podataka
- **Preduvjet:** Registrirani korisnik pregledava svoj korisnički račun
- **Opis osnovnog tijeka:**

1. Registrirani korisnik odabire opciju uređivanja korisničkog računa
2. Registrirani korisnik mijenja željene podatke o računu
3. Registrirani korisnik potvrđuje izmjene i vraća se u pregled korisničkog računa
4. Izmijenjeni podaci se spremaju u bazu podataka

- **Opis mogućih odstupanja:**

4.a Registrirani korisnik je promijenio email adresu u već zauzetu email adresu

1. Aplikacija javlja pogrešku s odgovarajućom porukom i vraća korisnika na stranicu za registraciju
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:** Registrirani korisnik pregledava svoj korisnički račun
- **Opis osnovnog tijeka:**
  1. Registrirani korisnik odabire ikonu svojeg profila
  2. Registrirani korisnik prilikom pregledavanja svojeg korisničkog računa 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:** Korisnik je prijavljen u sustav
- **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, Gradska služba
- **Preduvjet:** Djelatnik pregledava prijave ureda
- **Opis osnovnog tijeka:**
  1. Djelatnik gradskog ureda odabire prijavu
  2. Djelatnik mijenja status prijave
  3. Prijava se delegira nadležnoj gradskoj službi ako joj je status promijenjen u *u procesu rješavanja*

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 obavijest prijavitelju 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

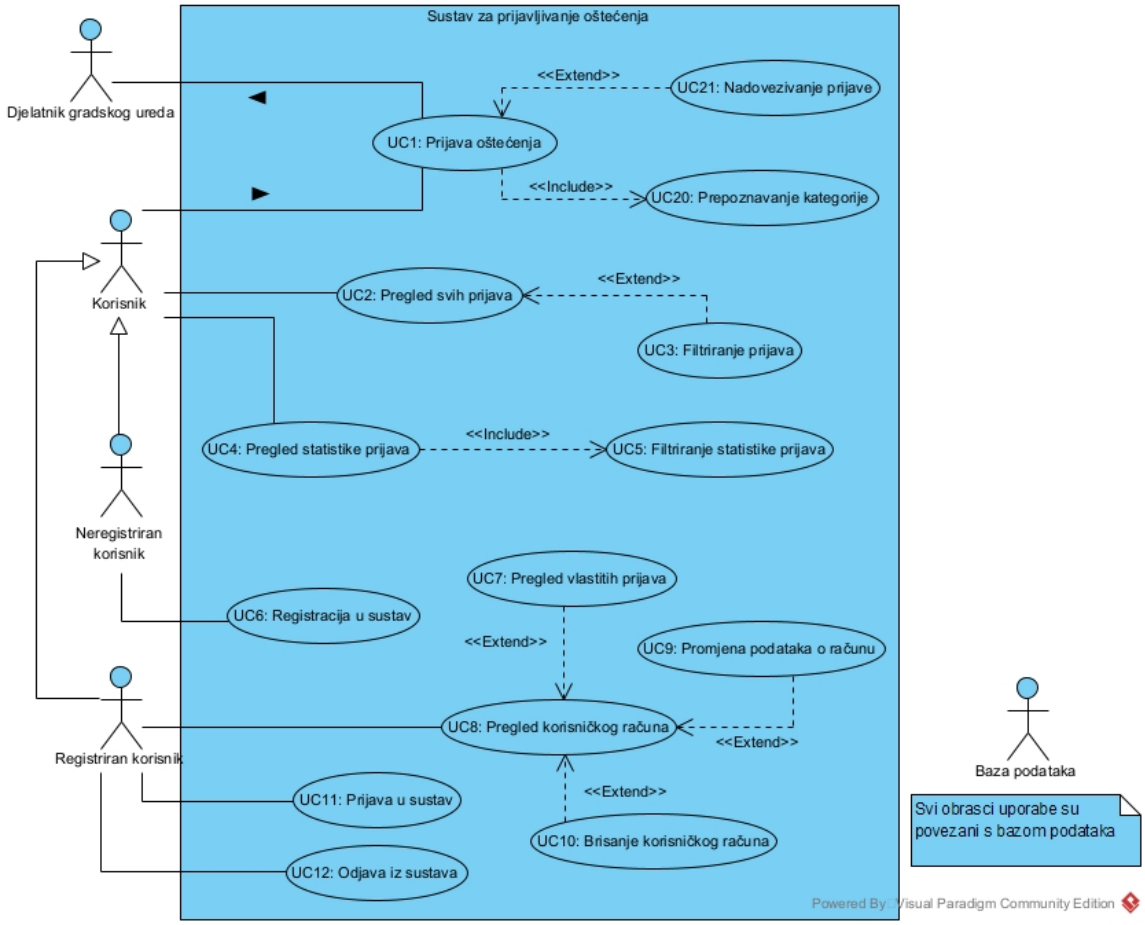
UC20 - Prepoznavanje kategorije

- **Glavni sudionik:** Korisnik
- **Cilj:** Automatsko prepoznavanje kategorije oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik podnosi prijavu oštećenja
- **Opis osnovnog tijeka:**
    1. Aplikacija dohvaća popis ključnih riječi i njihovih kategorija
    2. Korisnik unosi opis oštećenja
    3. Aplikacija na temelju prepoznatih ključnih riječi predlaže kategoriju
    4. Korisnik ostavlja predloženu kategoriju ili ručno odabire ispravnu kategoriju

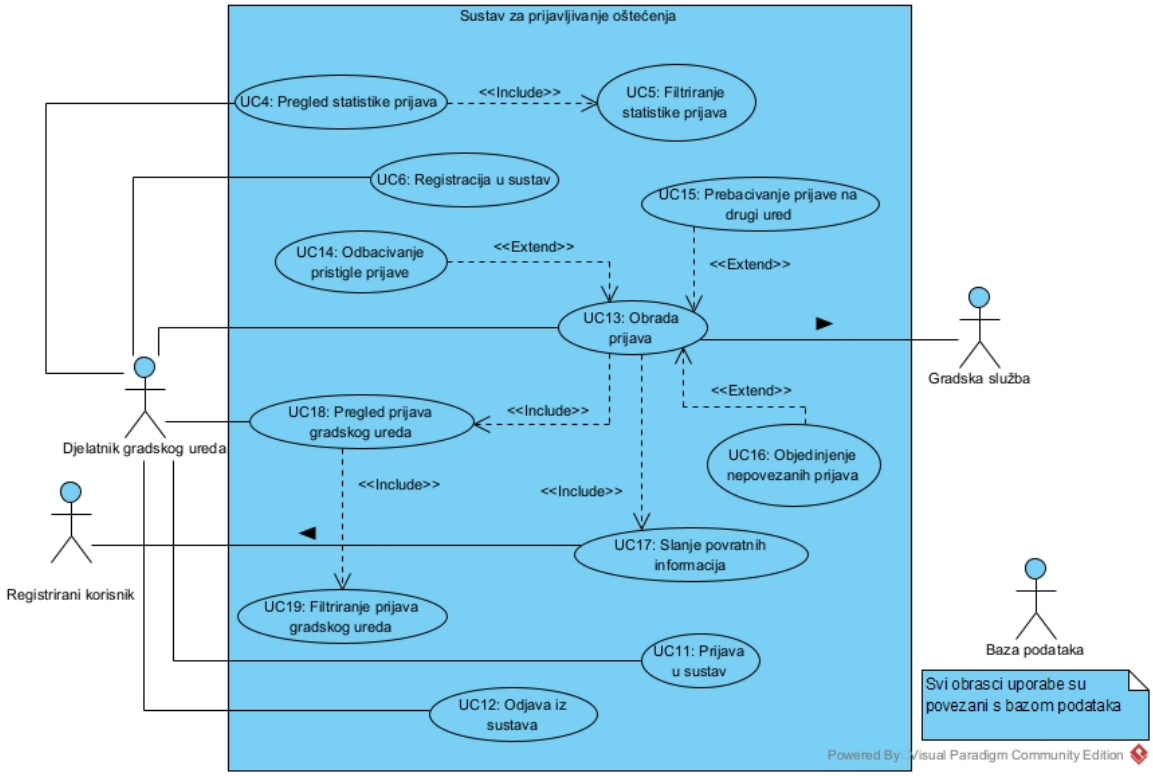
UC21 - Nadovezivanje prijave

- **Glavni sudionik:** Korisnik
- **Cilj:** Nadovezivanje prijave na već postojeću prijavu istog oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik podnosi prijavu oštećenja
- **Opis osnovnog tijeka:**
  1. Korisnik potvrđuje i podnosi prijavu
  2. Aplikacija provjerava ako u bazi podataka već postoji slična prijava
  3. Ako takva postoji, aplikacija nudi korisniku da se nadoveže na već postojeću prijavu
  4. Korisnik može nadovezati prijavu ili ju podnesti kao samostalnu

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, mogućnosti korisnika

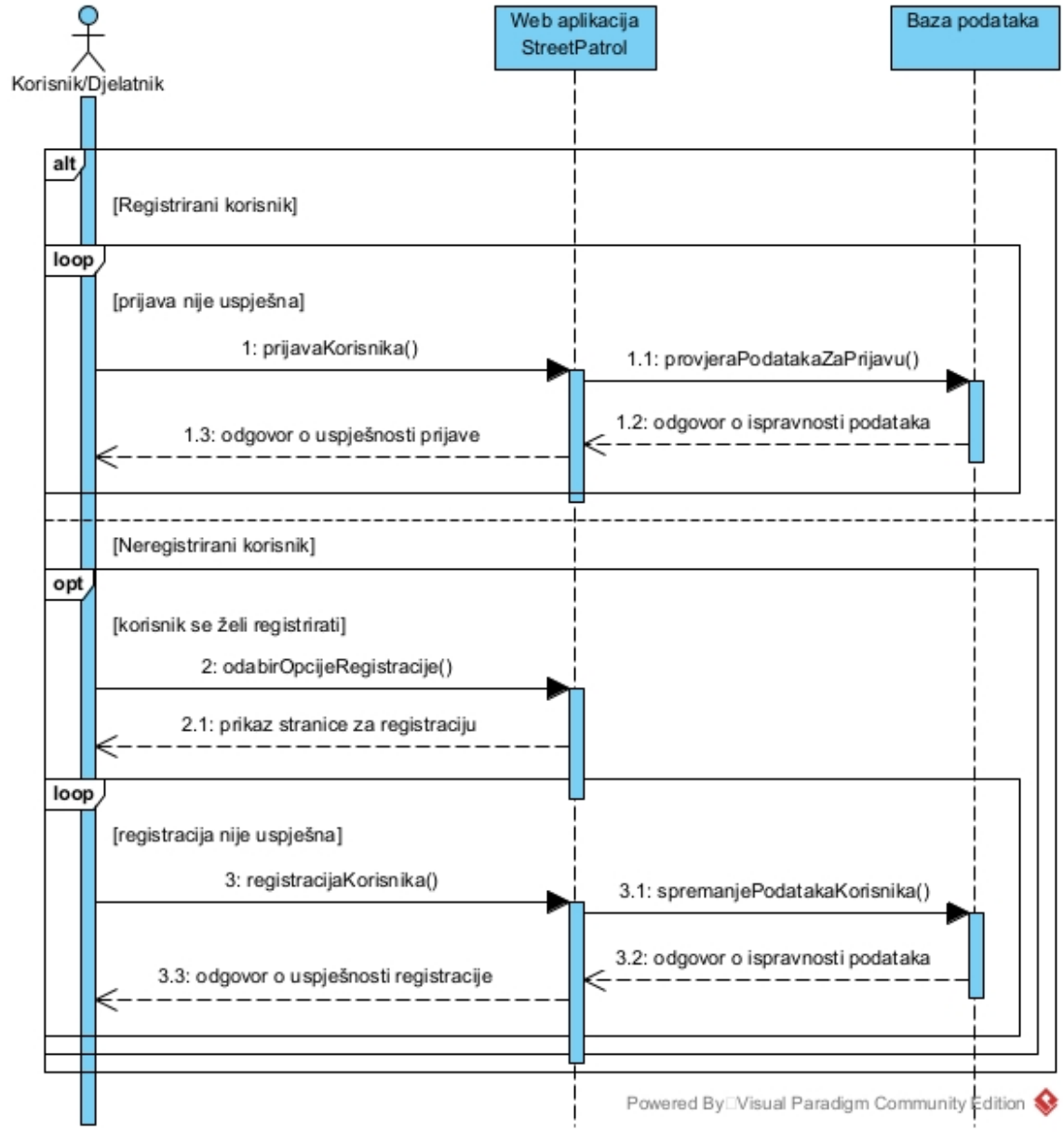


Slika 3.2: Dijagram obrasca uporabe, obrada prijava

3.1.2 Sekvencijski dijagrami

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

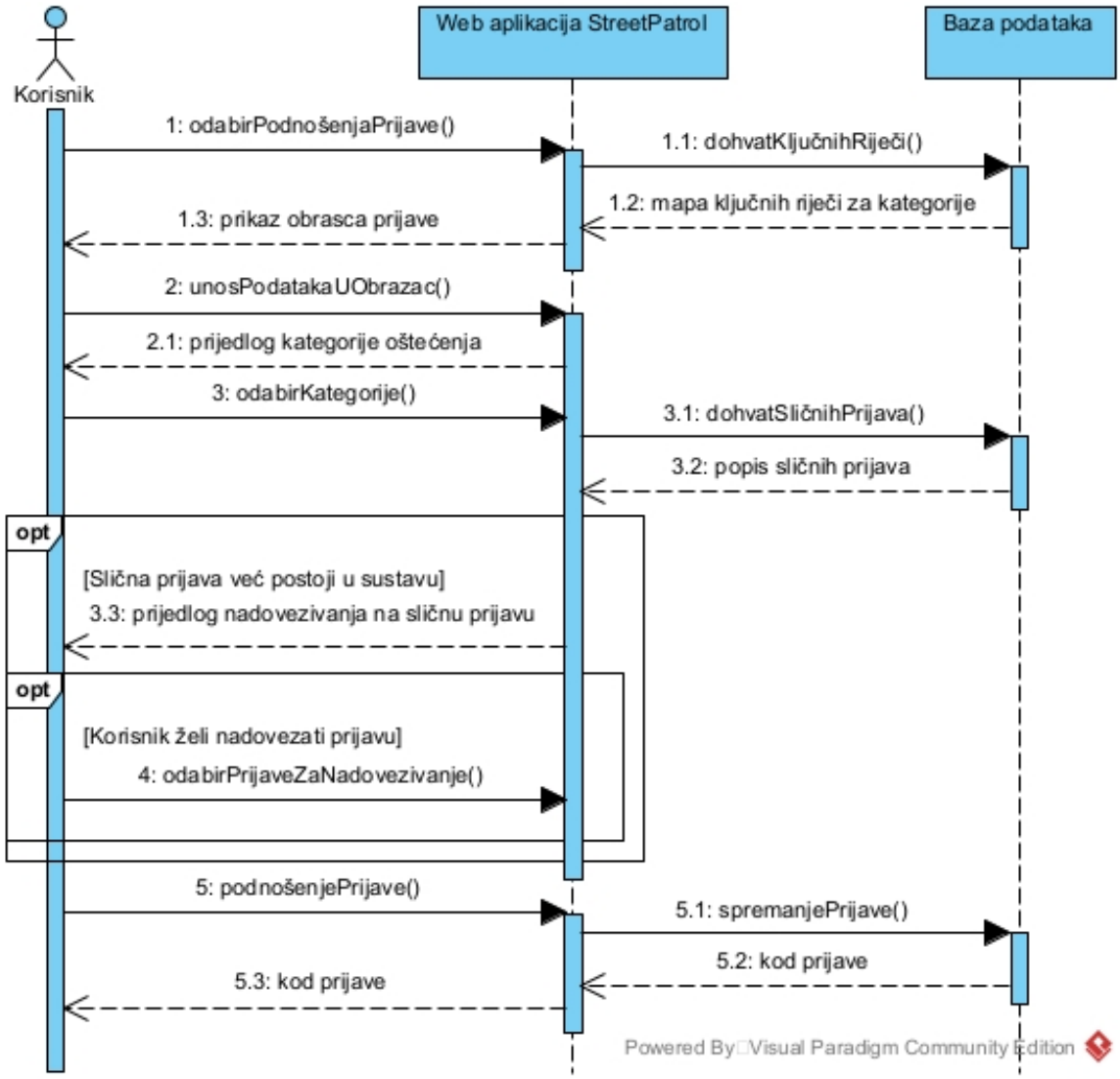
Korisnik(može biti i djelatnik) na naslovnoj stranici odabire opciju prijave ili registracije te mu se zatim otvara prikladna stranica. Ako se radi o stranici za prijavu, korisnik unosi svoj email i zaporku i potvrđuje unos. Ako su uneseni podaci validni korisniku se prikazuje naslovna stranica za prijavljenog 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, lozinka i barem jedna kategorija za koju će novi gradski ured biti nadležan. 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 anoniman neregistrirani korisnik. Djelatnik se mora prijaviti ili registrirati ured da bi obavljao svoju funkcionalnost.



Slika 3.3: Sekvencijski dijagram, prijava/registracija

Obrasci uporabe - UC1 - Prijava oštećenja

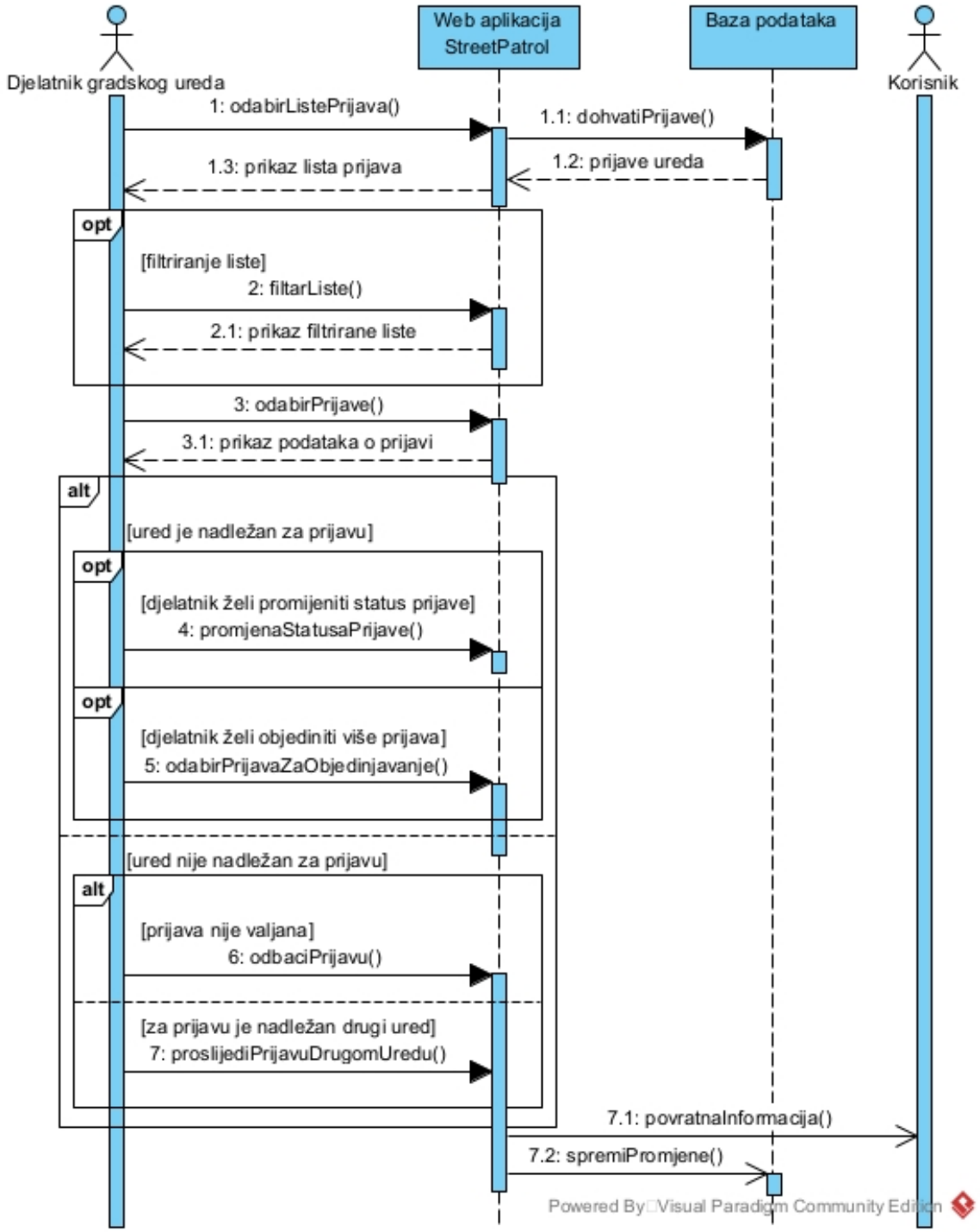
Korisnik odabire opciju podnošenja nove prijave oštećenja. U prijavi korisnik upisuje kratak naslov, opis oštećenja, lokaciju i kategoriju. Korisnik također može priložiti fotografiju oštećenja. Lokaciju korisnik može priložiti preko interaktivne karte, upisom adrese ili se lokacija može izvući iz metapodataka fotografija, ako je fotografija priložena. Aplikacija će sama pokušati prepoznati kategoriju oštećenja i na temelju toga i ponuditi nadležan ured. Za to su joj potrebne ključne riječi za svaku kategoriju koje je aplikacija dohvatila prilikom odabira podnošenja nove prijave. Aplikacija nakon unosa podataka dohvaća postojeće prijave koje imaju slična vremena, lokaciju i kategoriju. Ako takve prijave postoje, aplikacija će korisniku ponuditi da svoju prijavu nadoveže na jednu od ponuđenih prijava. Korisnik može, ali i ne mora nadovezati svoju prijavu kada mu je ta opcija ponuđena. Nakon podnošenja prijave korisnik dobiva kod prijave preko koje može pratiti njezino stanje.



Slika 3.4: Sekvencijski dijagram, podnošenje prijave

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 prijavu 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 rezultirat 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

- 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 s responzivnim dizajnom
- Aplikacija mora koristiti HTTPS protokol
- Aplikacija mora biti jednostavna za korištenje
- Neispravno korištenje korisničkog sučelja ne smije narušiti cjelokupan rad sustava
- Izvršavanje bilo koje akcije na aplikaciji ne smije trajati dulje od 5 sekundi
- Osjetljivi podaci, kao što su lozinke korisnika, se u bazu podataka moraju spremati u kriptiranom obliku koristeći *bcrypt* algoritam

4. Arhitektura i dizajn sustava

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

Web preglednik je program preko kojeg korisnik koristi aplikaciju. Preglednik 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. Neki od popularnih web preglednika su Crome, Safari i Edge.

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

Baza Podataka predstavlja sloj 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 i zadužen je za interakciju s korisnikom. On oblikuje korisničko sučelje i samim time definira što će korisnik vidjeti na web pregledniku kada koristi aplikaciju.

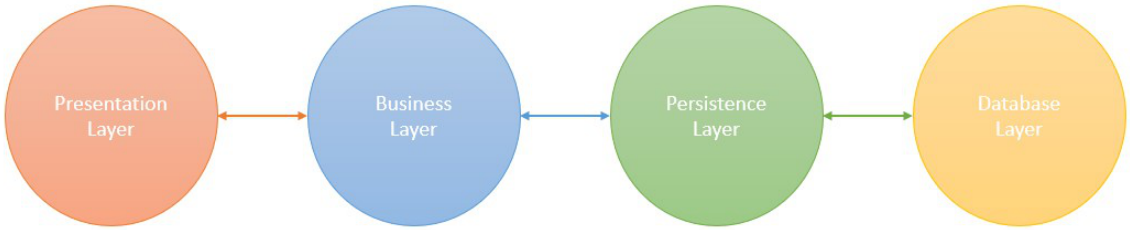
Backend je dio aplikacije koji obrađuje zahtjeve i kontrolira rad ostalih dijelova sustava. Baceknd je usko povezan s web poslužiteljem.

Za izradu backend dijela aplikacije odabrali smo programski jezik Java s radnim okvirom Spring Boot. Spring Boot radni okvir koristi višeslojnu arhitekturu u kojoj svaki sloj komunicira sa slojem koji se u hijerarhiji nalazi iznad ili ispod njega. Arhitektura Spring Boota se sastoji od četiri sloja:

- **Prezentacijski sloj** (*Presentation layer*) - Ovo je najviši sloj arhitekture i pred-

stavlja frontend dio aplikacije. Ovaj sloj prima HTTP zahtjeve i vrši autentifikaciju. Također vrši pretvorbu JSON objekata u objekte u Javi i obrnuto. Na kraju svoje obrade Prezentacijski sloj prosljeđuje HTTP zahtjev sljedećem sloju.

- **Poslovni sloj** (*Business layer*) - Ovaj sloj sadrži svu poslovnu logiku te je zadužen za validaciju i autorizaciju.
- **Sloj postojanosti** (*Persistence layer*) - Ovaj sloj sadrži logiku vezanu uz pohranu u bazu podataka. Sloj pretvara zapise iz baze podataka u objekte iz poslovnog sloja i obrnuto.
- **Sloj baze podataka** (*Database layer*) - Ovaj sloj sadrži bazu podataka ili više njih. Sloj je zadužen za obavljanje CRUD (copy, read, update, delete) operacija.



Slika 4.1: Prikaz Spring Boot arhitekture

Za razvoj frontend dijela aplikacije odlučili smo koristiti programski jezik JavaScript uz radni okvir React. React omogućuje jednostavnu izradu korisničkog sučelja preko JavaScripta, bez direktnog korištenja HTML-a.

Za razvojne okoline smo odlučili koristiti IntelliJ IDEA za rad u Spring Boot radnom okviru i Visual Studio Code za rad u React radnom okviru.

4.1 Baza podataka

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, umetanje 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

S obzirom na to da je još u fazi razvoja, aplikacija trenutno koristi H2 bazu podataka u radnoj memoriji. Ovu bazu podataka Spring Boot automatski inicijalizira i spaja s aplikacijom. Ova baza je pogodna za testiranje i razvoj te smo je zbog toga odlučili koristiti tijekom razvoja. Kasnije kada aplikacija bude gotova ćemo H2 bazu zamijeniti odvojenom PostgreSQL bazom podataka.

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
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*. Također postoji auto-refleksivna *Many-to-One* veza kojom se modelira nadovezivanje prijava.

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
isConnectedToID	INT	jedinstveni identifikator prijave na koju je ova prijava nadovezana

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
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.

Feedback tablica čuva podatke o statusu prijava koje su povezane. Također čuva podatke potrebne za slanje povratnih informacija korisnicima i računanje nekih statistika. Tablica je povezana *Many-to-One* vezom s tablicom *Report* preko atributa *reportID*.

Tablica 4.6:

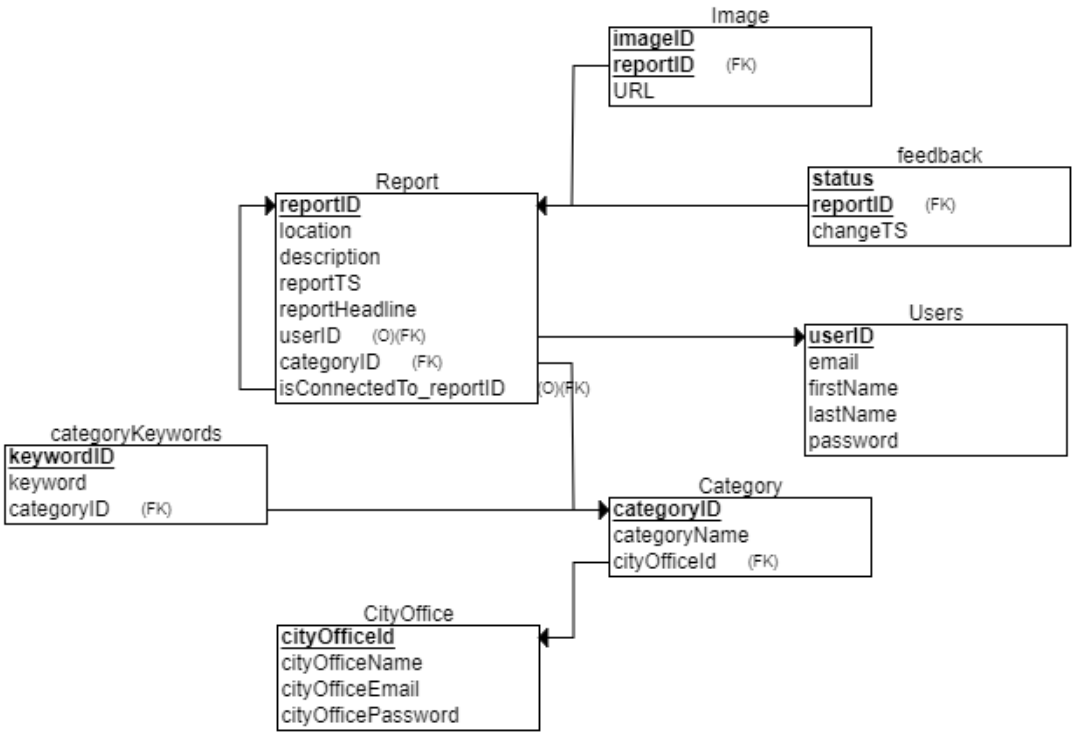
Feedback		
reportID	INT	jedinstveni identifikator grupe prijava na koju se podaci referiraju
status	VARCHAR	status prijava u referiranoj grupi
changeTS	TIMESTAMP	vrijeme kada se postavio status prijava iz referirane grupe

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 4.7:

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

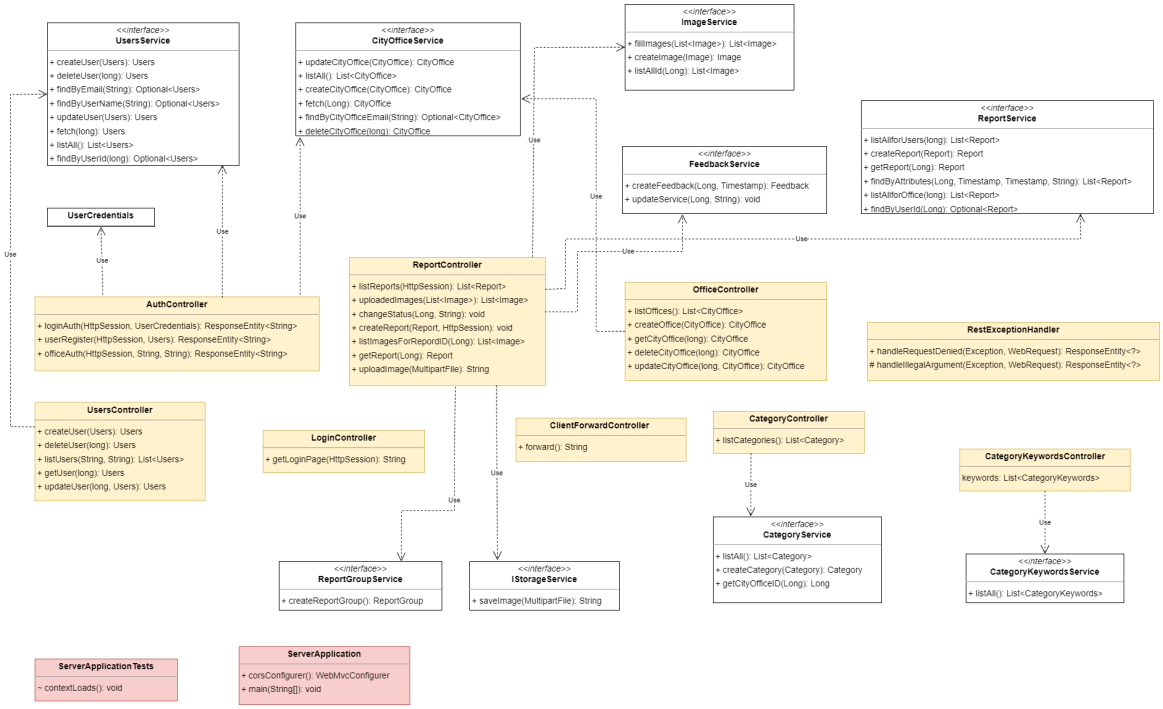


Slika 4.2: Relacijski dijagram baze podataka

4.2 Dijagram razreda

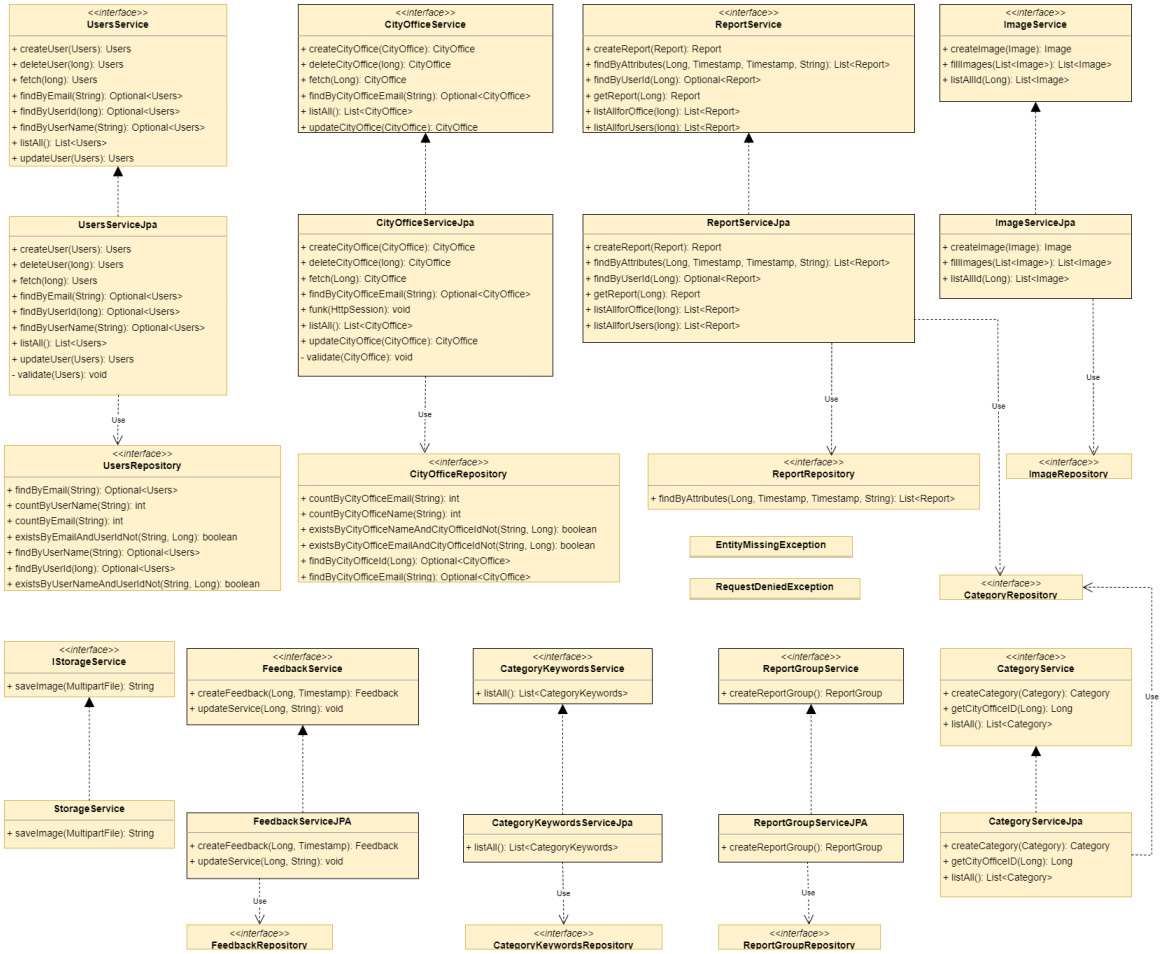
Na slikama 4.3, 4.4, 4.5, 4.6 su prikazani razredi koji pripadaju backend dijelu aplikacije. Razredi su arhitekturom podijeljeni na kontrolere, repozitorije i servise. Također postoje Domain i DTO (Data Transfer Object) modeli. Svi dijagrami su podložni promjenama, s obzirom na to da je aplikacija još uvijek u fazi razvoja.

Slika 4.3 prikazuje klase koje imaju ulogu kontrolera, tj. u kodu imaju anotaciju @RestController. Te klase definiraju endpointove i koriste se za dohvat i manipuliranje podacima iz baze podataka. ClientForwardController je kontroler koji povezuje Spring server React aplikacijom. Klase sa @RestController anotacijom koriste instance servisa. Servisi su klase koje imaju @Service anotaciju. Servisi implementiraju metode za upis (npr. createUser(Users);, createCitiyOffice(cityOffice), itd.), brisanje, dohvat i izmjenu podataka u bazi. Klasa ServerApplication je glavna klasa koja pokreće aplikaciju, a ServerApplicationTest služi za testiranje funkcionalnosti aplikacije, ali još nema implementiranu nikakvu funkcionalnost.



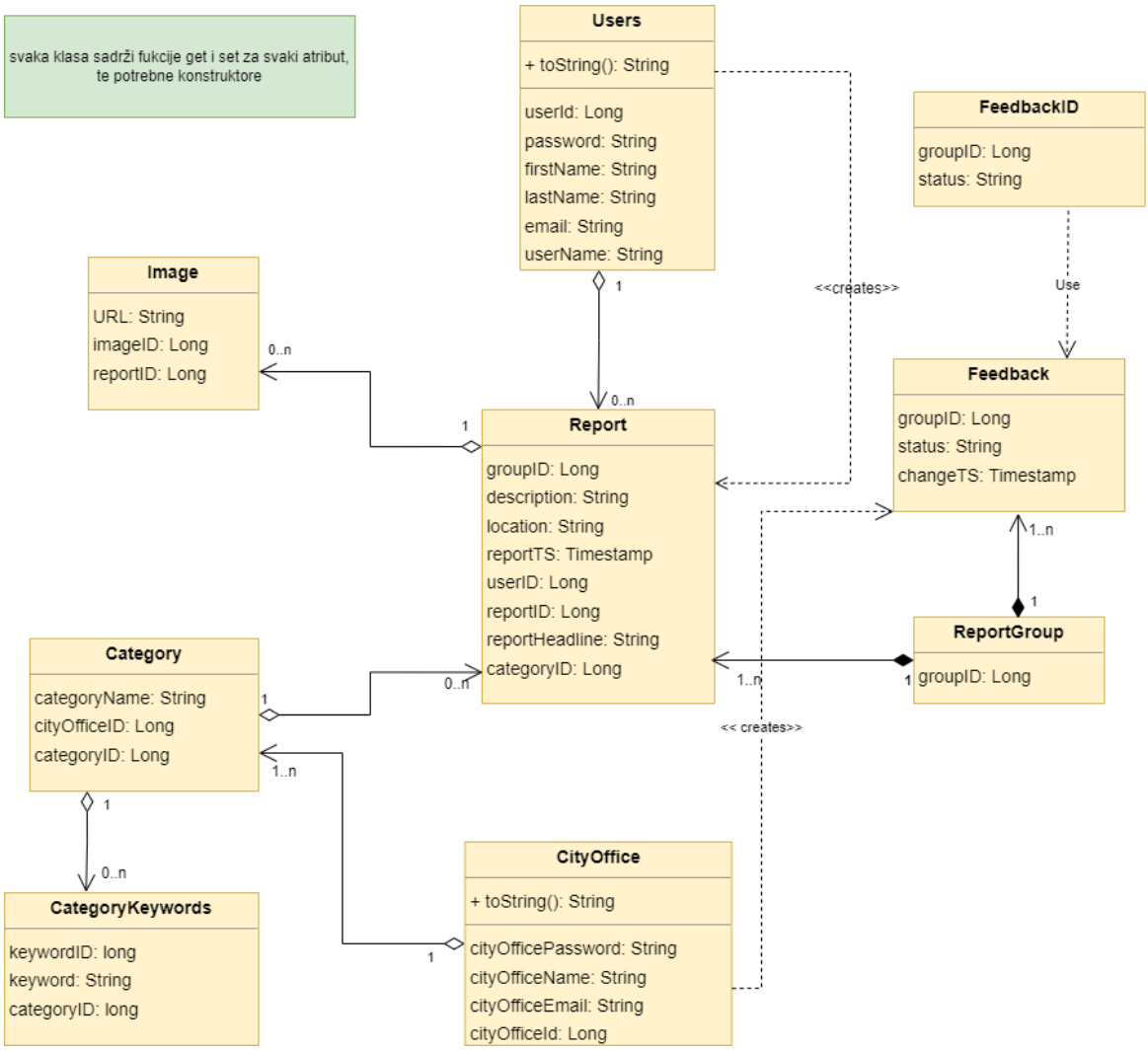
Slika 4.3: Kontroleri na backendu

Slika 4.4 prikazuje sučelja koja nasljeđuju sučelje JpaRepository. Sučelje JpaRepository definira razne CRUD (create, read, update, delete) metode za rad nad entitetima u bazi. Definirane metode izvršavaju razne vrste upite nad bazom podataka.



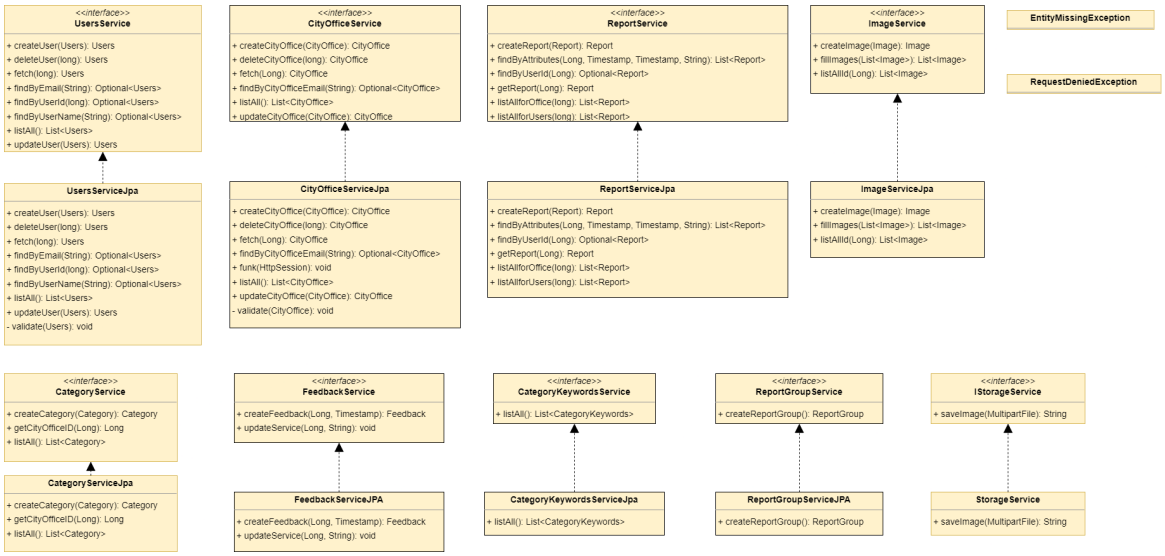
Slika 4.4: Data Access Objects

Slika 4.5 prikazuje klase u paketu repo koje reprezentiraju entitete u bazi podataka te njihove međusobne veze. Ove klase definiraju tipove atributa pojedinih entiteta, ograničenja nad atributima i primarne i strane ključeve. Svaka klasa ima implementirane metode get() i set() za svaki od atributa, kao i odgovarajuće konstruktore. Klasa FeedbackID služi kao pomoćna klasa za definiranje primarnog ključa feedback. Korisnici (Users) stvaraju prijave (Report), a gradski uredi (CityOffice) upravljaju prijavama i prilikom promjena automatski zapisuje nove povratne informacije (Feedback) u bazu podataka za tu grupu prijava.



Slika 4.5: Modeli

Slika 4.6 prikazuje klase servisa koje definiraju metode za CRUD operacije nad bazom podataka. EntityMissingException i RequestDeniedException su pomoćne klase za baratanje pogreškama. Klase servisa koriste instance Repository klasa iz slike 2. za poziv metoda definiranih u sklopu sučelja Repository odnosno JpaRepository.



Slika 4.6: Servisi

4.3 Dijagram stanja

Dijagram stanja je vrsta UML dijagrama koja prikazuje stanja objekta i prijelaze između stanja. Prijelazi se aktiviraju određenim događajima i opcionalno dodatnim uvjetima.

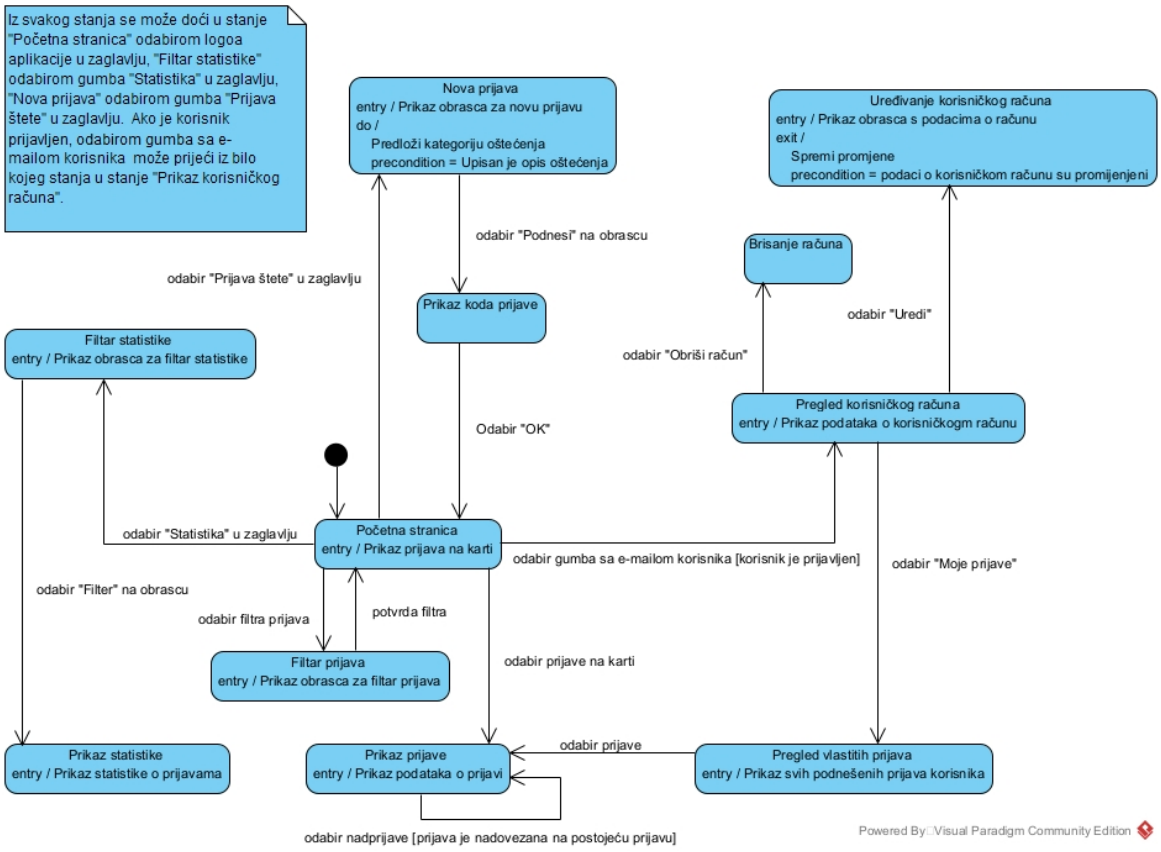
Slika 4.7 prikazuje dijagram stanja za korisnika koji može, ali i ne mora biti prijavljen u sustav. Korisniku je na početku prikazana glavna stranica na kojoj je karta sa prikazanim aktivnim prijavama u sustavu. Zaglavlje stranice sadrži gumbove za podnošenje nove prijave oštećenja (”Prijava štete”), prikaz statistike prijava (”Statistika”). Također, zaglavlje sadrži gumb sa e-mailom korisnika koji služi za upravljanje korisničkim računom ako je korisnik prijavljen u sustav. Ako korisnik nije prijavljen, zaglavlje sadrži gumbove za prijavu (”Log in”) i registraciju (”Register”) u sustav. S obzirom da je zaglavlje isto na svim dijelovima aplikacije, korisnik ima pristup ovim opcijama bilo gdje u aplikaciji.

Na glavnoj stranici korisnik može odabrati filtar prijava. Pomoću tog filtra korisnik može filtrirati aktivne prijave u sustavu koje će se prikazati na karti na početnoj stranici. Odabirom neke od prikazanih prijava na karti korisniku se otvara stranica odabrane prijave. Ako je ta prijava nadovezana na drugu prijavu oštećenja, korisnik može kliknuti na tu prijavu da mu se otvori stranica nadprijave.

Kod pregleda statistike prijava korisnik mora ispuniti obrazac filtra. Nakon potvrde obrasca korisniku se prikazuje statistika prijava koje su odabrane pomoću filtra.

Prilikom prijave novog oštećenja korisnik ispunjava obrazac prijave. Kada je u obrazac upisan opis oštećenja, sustav korisniku predlaže kategoriju oštećenja. Sustav korisniku prikazuje jedinstveni kod prijave kada korisnik podnese prijavu. Kada se vraća na početnu stranicu kada potvrdi da je vidio kod prijave.

Ako je korisnik prijavljen može pregledati podatke o svojem korisničkom računu. Prilikom pregleda korisničkog računa može dodatno pregledavati vlastite prijave. Kada želi pregledati neku od prijava, korisniku se prikazuje stranica te prijave, isto kao i kod pregleda svih prijava na početnoj stranici. Korisnik također može mijenjati podatke o svojem korisničkom računu. Kod potvrde se izmjene spremaju, ako je do njih došlo. Također, korisnik kod pregleda računa može izbrisati svoj račun čime on postaje neprijavljen i neregistriran.



Slika 4.7: Dijagram stanja

4.4 Dijagram aktivnosti

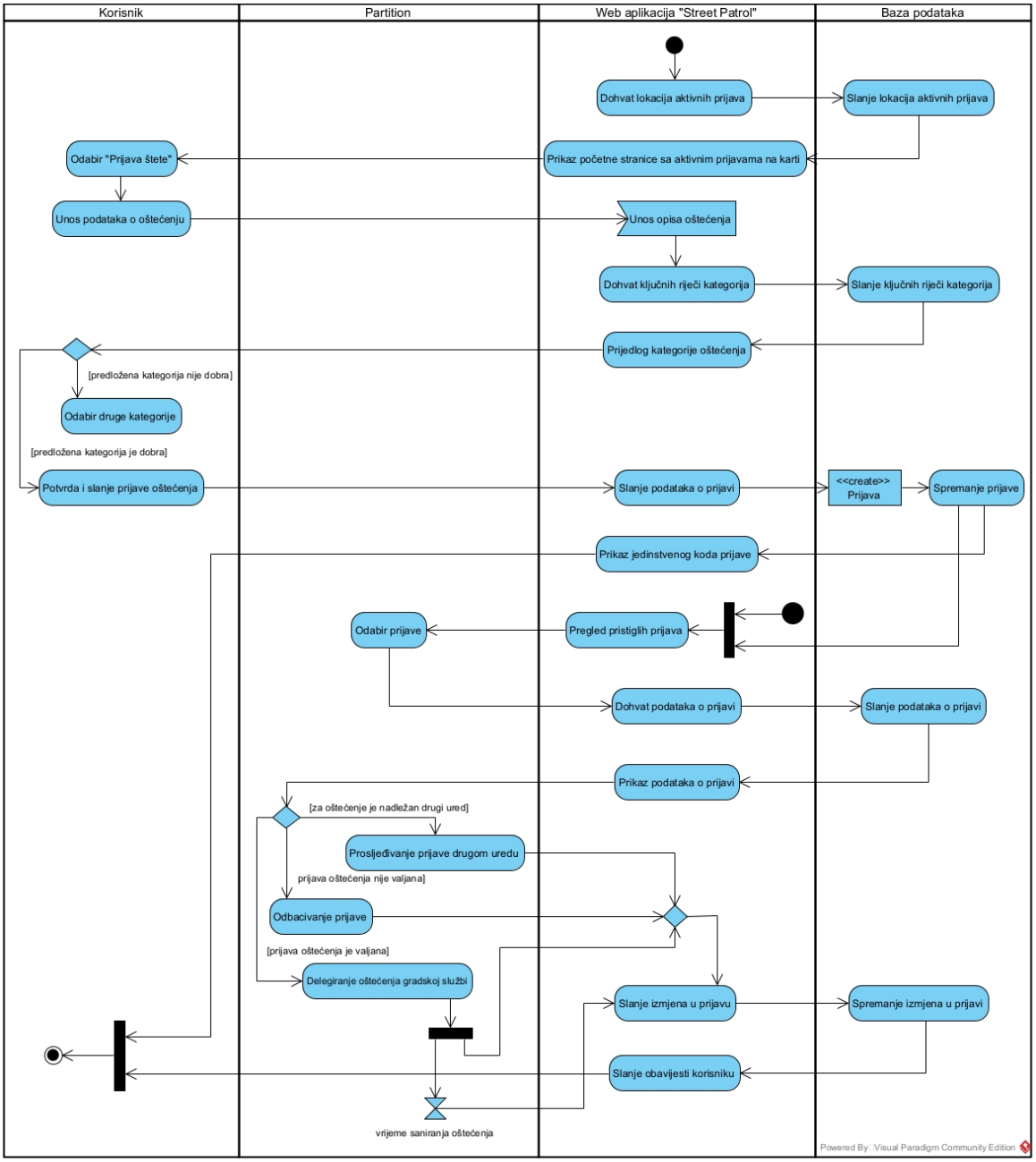
dio 2. revizije

Dijagram aktivnosti je ponašajni UML dijagram koji modelira ponašanje sustava nizom akcija koje zajedno čine aktivnost. Dijagram aktivnosti se primjenjuje za modeliranje upravljačkog i podatkovnog toka. Ovakav tip dijagrama nije preporučljiv za modeliranje sustava sa događajima poticanim ponašanjem. Kod dijagrama aktivnosti svaka akcija se izvršava nakon završavanja prethodne akcije. Općenito je u dijagramu aktivnosti naglasak na jednostavnosti prikaza.

Na dijagramu 4.8 prikazan je proces podnošenja prijave oštećenja i njezine obrade u gradskom uredu.

Korisnik odabire opciju ”Prijava štete” te ispunjava podatke o oštećenju na dobivenom obrascu. Kada je unesen opis prijave, aplikacija pomoću ključnih riječi pokušava prepoznati kategoriju oštećenja koju predlaže korisniku. Korisnik može prihvatiti predloženu kategoriju ili ručno odabrati drugu.

Nakon što je prijava podnesena i spremljena u bazu podataka, djelatnik gradskog ureda može obraditi tu prijavu. Ako smatra da je prijava nevaljala, djelatnik ju odbacuje. Ako je prijava podnesena krivom uredu, korisnik prijavu može proslijediti drugom uredu. Ako je prijava valjana, djelatnik tu prijavu šalje gradskoj službi za sanaciju oštećenja. U svakom od navedenih slučaja promjena u prijavi se sprema u bazu podataka i obavještava se korisnika koji je podnio prijavu, ako je korisnik registriran u sustavu. Na kraju, kada gradska služba sanira oštećenje iz valjane prijave, ta se promjena također evidentira u bazi podataka i obavještava se korisnik kao i u ostalim slučajevima.



Slika 4.8: Dijagram aktivnosti

4.5 Dijagram komponenti

dio 2. revizije

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

Popis literature

1. GeeksForGeeks, Spring Boot Architecture, https://www.geeksforgeeks.org/spring-boot-architecture/
2. Programsko inženjerstvo, FER ZEMRIS, http://www.fer.hr/predmet/proinz
3. Visual Paradigm, https://www.visual-paradigm.com/
4. draw.io, https://app.diagrams.net/
5. The Unified Modeling Language, https://www.uml-diagrams.org/
6. Baeldung, Spring Boot H2 database, https://www.baeldung.com/spring-boot-h2-database

Indeks slika i dijagrama

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak
    - Datum: 23. listopada 2023.
    - Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, 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: 30. listopada 2023.
    - Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, Eno Peršić, Bruno Mikulan, Filip Vučenik
    - Teme sastanka:
        - dizajn baze podataka
        - dizajn korisničkog sučelja aplikacije
        - detaljnija rasprava o svojstvima aplikacije
3. sastanak
    - Datum: 6. studenoga 2023.
    - Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, Eno Peršić, Bruno Mikulan, Filip Vučenik
    - Teme sastanka:
        - izvještaj svakog dijela tima o svojem napretku
        - rad na dizajnu aplikacije
4. sastanak
    - Datum: 13. studenoga 2023.

- Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, Eno Peršić, Bruno Mikulan, Filip Vučenik
- Teme sastanka:
  - izvještaj svakog dijela tima o svojem napretku
  - korekcije na pojedinim dijelovima aplikacije
  - rad na spajanju frontend i backend dijelova aplikacije

5. sastanak
   - Datum: 15. studenoga 2023.
   - Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, Eno Peršić, Bruno Mikulan, Filip Vučenik
   - Teme sastanka:
     - rad na spajanju frontend i backend dijelova aplikacije
6. sastanak
   - Datum: 16. studenoga 2023.
   - Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, Eno Peršić, Bruno Mikulan, Filip Vučenik
   - Teme sastanka:
     - rad na deploymentu aplikacije
     - definiranje završnih zadataka prije prve predaje
7. sastanak
   - Datum: 11. prosinca 2023.
   - Prisustvovali: Petar Belošević, Vinko Brkić, Tomislav Grudić, Fran Meglić, Eno Peršić, Bruno Mikulan, Filip Vučenik
   - Teme sastanka:
     - kratka evaulacija dosadašnjeg rada, dogovor o raspodijeli bodova
     - pregled nedovršenog posla, daljnjih obaveza i zadataka kod dokumentacije i same aplikacije

Tablica aktivnosti

	Petar Bošević	Vinko Brkić	Tomislav Grudić	Fran Meglić	Bruno Mikulan	Eno Peršić	Filip Vučenik
Upravljanje projektom	8h						
Opis projektnog zadatka	4h						1h
Funkcionalni zahtjevi	2h		1h				
Opis pojedinih obrazaca	3h						
Dijagram obrazaca	3h				2h		
Sekvencijski dijagrami	4h	2h		1h			
Opis ostalih zahtjeva	1h					1h	
Arhitektura i dizajn sustava	3h						
Opis baze podataka	2h		2h				
Dijagram razreda	1h		2h				
Dijagram stanja	3h						
Dijagram aktivnosti	3h						
Dijagram komponenti							
Korištene tehnologije i alati		2h			2h		
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja	1h						
Zaključak i budući rad							
Popis literature	0.5h						

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Petar Belošević	**Vinko Brkić**	**Tomislav Grudić**	**Fran Meglić**	**Bruno Mikulan**	**Eno Peršić**	**Filip Vučenik**
Povezivanje frontenda i backenda					3h	2h	2h
Puštanje u pogon							3h
rad na frontendu		14h			10h	14h	
rad na backendu			4h	10h			10h
izrada baze podataka			0.5h	1h			0.5h
spajanje s bazom podataka							
usvajanje korištenih tehnologija	2h	6h	5h	8h	3h	6h	10h