Programsko inženjerstvo

Ak. god. 2023./2024.

Štete javnih površina

Dokumentacija, Rev. *1*

Grupa: *CestaFix*

Voditelj: *Fran Fodor*

Datum predaje: **17. 11. 2023.**

Nastavnik: *izv. prof. dr. sc. Vlado Sruk*

Sadržaj

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Preuzet predložak.	Sara Podvorec	28.10.2023.
0.2	Dodane osnovne informacije. Napisan početak opisa projekta. Crvenom bojom označeni dijelovi teksta koje treba izmijeniti.	Jan Murić	28.10.2023.
0.3	Nadopuna opisa projekta.	Sara Podvorec	29.10.2023.
0.4	Nadopuna osnovnih informacija. Nadupuna dnevnika sastajanja.	Mateo Jakšić	30.10.2023.
0.5	Stiliziranje opisa projekta.	Sara Podvorec	30.10.2023.
0.6	Nadopuna dnevnika sastajanja.	Mateo Jakšić	31.10.2023.
0.7	Nadopuna opisa projektnog zadatka i funkcionalnih zahtjeva.	Sara Podvorec	31.10.2023.
0.8	Nadopuna literature i funkcionalnih zahtjeva.	Mateo Jakšić	1.11.2023.
0.9	Napisan dio obrazaca uporabe	Mateo Jakšić	1.11.2023.
0.10	Nadopuna obrazaca uporabe	Mateo Jakšić	2.11.2023.
0.11	Nadopuna obrazaca uporabe. Dodani dijagrami obrazaca uporabe.	Mateo Jakšić	6.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.12	Dodani sekvencijski dijagrami. Dodani ostali zahtjevi. Nadopuna dnevnika sastajanja.	Mateo Jakšić	10.11.2023.
0.13	Dodane osnovne informacije o arhitekturi. Nadopuna opisa baze podataka.	Sara Podvorec	12.11.2023.
0.14	Dodan opis pojedinih entiteta baze podataka.	Sara Podvorec	13.11.2023.
0.15	Nadopuna dnevnika sastajanja. Nadopuna opisa i kreiranje ERD dijagrama baze podataka.	Mateo Jakšić	14.11.2023.
0.16	Dodatan ER dijagram baze podataka	Mateo Jakšić	15.11.2023.
0.17	Dodan opis arhitekture. Dodan vizualni prikaz arhitekture.	Sara Podvorec	16.11.2023.
0.18	Dodani dijagrami razreda i njihovi opisi. Nadopuna literature.	Mateo Jakšić	16.11.2023.

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

2.1 Opis projektnog zadatka 'Štete javnih površina'

Svakodnevno se suočavamo s brojnim problemima na javnim površinama i cestama u našim gradovima. Problemi kao što su vandalizam, oštećenje pločnika, udarne rupe na cestama, smeće i slično predstavljaju potencijalnu opasnost za građane te poprilično narušavaju kvalitetu njihovog svakodnevnog života.
Neadekvatna briga o oštećenjima i njihovoj sanaciji često ostavlja građane u vrlo nepovoljnoj situaciji kada iste žele prijaviti vlastima. Kako bi suzbili osjećaj nemoći građana i poboljšali kvalitetu života cjelokupne zajednice, potrebno je omogućiti adekvatnu prijavu oštećenja na javnim površinama.
Glavni je cilj ovog projekta razviti programsku podršku za stvaranje web aplikacije CestaFix za dojavu oštećenja i drugih problema na cestama, parkovima, javnim ustanovama i ostalim javnim mjestima kako bi se olakšala dojava, kategorizacija te u konačnici rješenja prijavljenih problema. Ideja je omogućiti građanima da na što jednostavniji način prijavljuju oštećenja i probleme na javnim površinama i cestama svojih gradova te tako pomognu lokalnim vlastima da pravodobno reagiraju na nastale probleme.
S obzirom na velik broj javih ustanova i njihovih odjeljaka koji se bave različitim problemima, teško je pratiti tko je nadležan za koju vrstu problema i nad kojim područjem. Sustav bi automatski odredio nadležno tijelo prema kategoriji prijave i drugim parametrima te proslijedio prijavu na obradu. Ključno je da proces prijave bude jednostavan i brz. Prilikom pokretanja sustava prikazuje se početna stranica na kojoj se nalazi karta s označenim lokacijama već podnesenih prijava raznovrsnih oštećenja javnih površina i navigacijski izbornik putem kojeg se pristupa registraciji/prijavi u sustav, podnošenju prijava, provjeri statusa već podnesenih prijava, često postavljanim pitanjima te informacijama o samoj aplikaciji. Svakom neregistriranom korisniku omogućeno je kreiranje novog računa prilikom čega je potrebno unijeti:

- *korisničko ime*
- *lozinku*

- *email adresu*

Ako korisnik već ima postojeći račun, prilikom prijave u sustav unosi:

- *email adresu*
- *lozinku*

Registrirani korisnik može pregledavati i mijenjati svoje osobne podatke i izbrisati svoj korisnički račun te postoji mogućnost naknadne dodjele prava službenika ili administratora stupanjem u kontakt naveden pri dnu stranice.
Također, postoji opcija anonimnog podnošenja prijave na temelju koje se status podnesene prijave prati putem jedinstvenog broja prijave bez iznošenja osobnih podataka i potrebe za registracijom odnosno prijavom.

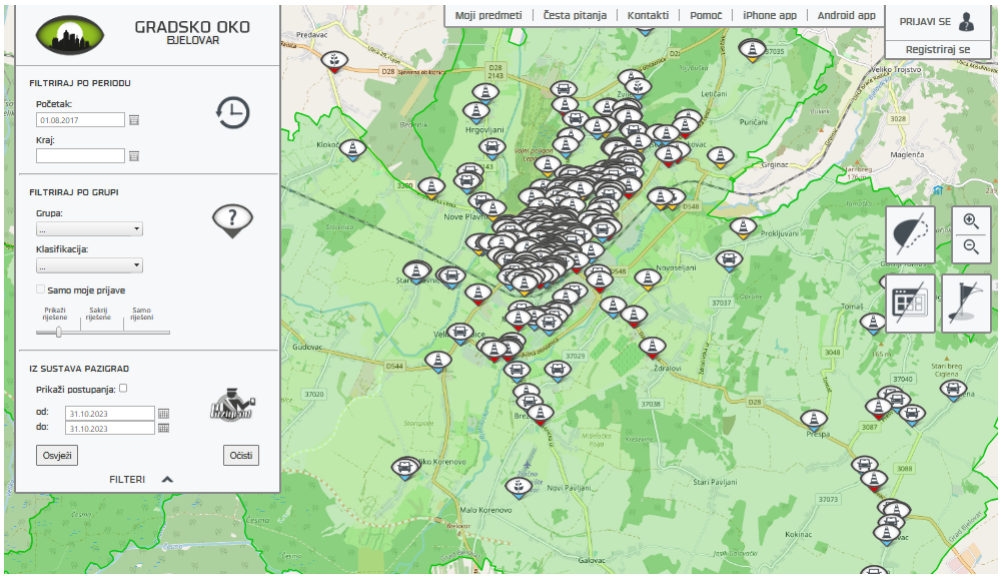
Korisnik s pravom službenika ima mogućnost ažuriranja statusa prijava i pregled profila prijavitelja (ili jedinstvenog broja prijave u slučaju anonimnog prijavitelja) među kojima odabire relevantne prijave. Nakon što sustav proslijedi prijavu u nadležni gradski ured, službenik tog ureda može izabrati problem i krenuti raditi na njegovu rješenju.

Administrator sustava ima najveće ovlasti među koje pripada i pristup bazi s popisom registriranih korisnika i njihovim podacima te mogućnost brisanja korisnika kao i dodjeljivanje administratorskih prava i prava službenika drugim korisnicima.
Kako bi olakšala proces podnošenja prijave, aplikacija omogućuje građanima da brzo i precizno dokumentiraju probleme. Svaka prijava uključuje naziv problema, kratki opis, geografske koordinate problema, jedinstveni broj prijave te opcionalno fotografije kako bi se problem što bolje opisao. Sustav također olakšava identifikaciju vremenski bliskih prijava na istoj lokaciji kako bi se građani mogli povezati na postojeće prijave sličnih problema, čime se smanjuje dupliciranje prijava i ubrzava proces njihovog rješavanja. Podnesene prijave obrađuju određeni gradski uredi, a svaka promjena u statusu prijave vidljiva je prijavitelju i svim ostalim korisnicima. Gradski uredi također imaju mogućnost objediniti nepovezane prijave na istoj lokaciji, što je vidljivo prijaviteljima. Kako bi gradskim uredima olakšali razvrstavanje i obrađivanje prijava, sustav dopušta prijave za različite kategorije problema. Osnovne kategorije problema su:

- *oštećenja na cesti*
- *oštećenja na vodovodnoj infrastrukturi*

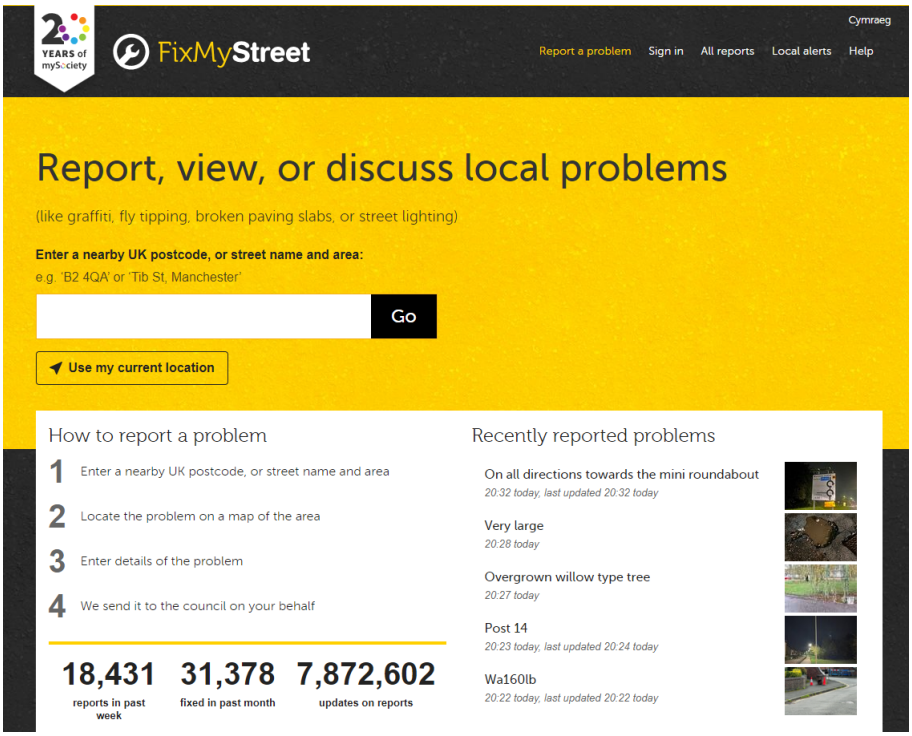
- *oštećenja na zelenim površinama*
- *oštećenja na elektroenergetskoj infrastrukturi*

Sve podnesene prijave su javno dostupne i mogu se grupirati prema temi i lokaciji, što omogućuje građanima i vlastima da prate i analiziraju probleme u stvarnom vremenu.
Statistika o statusima prijava, poput vremena potrebnog za rješavanje problema, također se obrađuje u stvarnom vremenu kako bi poboljšala učinkovitost sustava i pomogla gradskim vlastima u praćenju učinkovitosti njihovih usluga.
Posebna je pozornost posvećena zaštiti svih osobnih podataka te osiguranju korisničkog iskustva koje će biti jednostavno i intuitivno, kako bi se potaknulo građane na aktivno sudjelovanje u poboljšanju svojih zajednica.
Skup korisnika koji bi mogao biti zainteresiran za ovu aplikaciju je vrlo širok. To uključuje građane koji žele prijaviti probleme na javnim površinama, lokalne gradove i njihove urede koji će obraditi prijave, službenike koji će nadzirati i rješavati prijave te administratore koji će upravljati sustavom i osigurati sigurnost podataka.
Jedan od primjera sličnog sustava koji se aktivno upotrebljava u Hrvatskoj je "Gradsko Oko". Projekt je pokrenut u kolovozu 2017. godine u svrhu prijave komunalnih problema na području grada Bjelovara, a od tad se proširio na još 11 gradova i općina te na prijavu problema na moru.



Slika 2.1: Početna stranica sustava "Gradsko Oko"

Sustav “FixMyStreet” koji se koristi na području Ujedinjenog Kraljevstva također omogućava građanima da prijave probleme na javnim površinama i cestama u svojim lokalnim zajednicama kao i prikaz statističkih podataka o prijavljenim problemima, vremenskim okvirima za rješavanje i druge korisne informacije.



Slika 2.2: Početna stranica sustava ”FixMyStreet”

Postojeća slična rješenja, kao što su ”Gradsko Oko” i “FixMyStreet”, pružaju mogućnost prijave komunalnih problema, ali se razlikuju u nekoliko ključnih aspekata. Naša aplikacija ima određene značajke koje ju čine jedinstvenom, posebno u pogledu anonimnih prijava, statistike i praćenja rada gradskih ureda te povezivanja vremenski bliskih prijava.
Ova web aplikacija ima potencijal za prilagodbu i nadogradnje u budućnosti. Moguće nadogradnje uključuju implementaciju mobilne aplikacije kako bi se olakšao proces prijave, dodatne značajke za praćenje rada gradskih ureda, uvođenje sustava za nagrađivanje korisnika koji aktivno sudjeluju u rješavanju problema te proširenje na druge regije.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

*Navesti **dionike** koji imaju **interes u ovom sustavu** ili **su nositelji odgovornosti**. To su prije svega korisnici, ali i administratori sustava, naručitelji, razvojni tim.*

Dionici:

1. Neregistrirani korisnik
2. Registrirani korisnik
   (a) Građanin
   (b) Službenik gradskog ureda
3. Administrator
4. Razvojni tim
5. Suradnici
   (a) Nastavnik
   (b) Demonstrator

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
   (a) podnositi anonimne prijave
   (b) pratiti status prijave jedinstvenim brojem prijave
   (c) imati uvid u postojeće prijave
   (d) imati uvid u statistiku statusa prijava
   (e) registrirati korisnički račun
2. Građanin (inicijator) može:
   (a) prijaviti se na korisnički račun

(b) odjaviti se s korisničkog računa
(c) pratiti status vlastitih prijava
(d) imati uvid u postojeće prijave
(e) imati uvid u statistiku statusa prijava
(f) izmjeniti podatke korisničkog računa

3. Službenik gradskog ureda (inicijator) može:

(a) obrađivati podnesene prijave
(b) objediniti nepovezane prijave po temi i lokaciji
(c) promijeniti status prijave
(d) pratiti i analizirati probleme u stvarnom vremenu
(e) izmjeniti podatke korisničkog računa

4. Administrator (inicijator) može:

(a) uređivati i brisati korisničke račune
(b) pristupiti bazi podataka
(c) kreirati i ukloniti gradski ured
(d) dodijeliti i ukloniti ulogu službenika gradskog ureda registriranom korisniku
(e) dodijeliti i ukloniti ulogu administratora registriranom korisniku

5. Baza podataka (sudionik) može:

(a) pohraniti sve podatke o korisnicima i njihovim ulogama
(b) pohraniti sve podatke o prijavljenim oštećenjima
(c) pohraniti sve podatke o kategorijama problema i gradskim uredima za njihovo obrađivanje

3.1.1 Obrasci uporabe

dio 1. revizije

Opis obrazaca uporabe

Funkcionalne zahtjeve razraditi u obliku obrazaca uporabe. Svaki obrazac je potrebno razraditi prema donjem predlošku. Ukoliko u nekom koraku može doći do odstupanja, potrebno je to odstupanje opisati i po mogućnosti ponuditi rješenje kojim bi se tijek obrasca vratio na osnovni tijek.

UC01 - Registracija korisničkog računa

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoranje korisničkog račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu
  2. Korisnik odabire opciju da nema korisnički račun
  3. Korisnik unosi potrebne podatke i potvrđuje unos
  4. Korisnik se upisuje u bazu podataka i korisnik se preusmjerava na početnu stranicu
- **Opis mogućih odstupanja:**

  3.a Odabir već zauzetog korisničkog imena i/ili e-mail adrese, unos korisničkih podataka u neispravnom formatu, nepodudaranje lozinki
  1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za registraciju
  2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC02 - Prijava na korisnički račun

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Dobivanje pristupa korisničkom sučelju s ovlastima registriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je obavio registraciju i njegovi podaci se nalaze u bazi podataka

- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu
  2. Korisnik unosi korisničko ime i lozinku te potvrđuje unos
  3. Korisnik se preuspjerava na početnu stranicu
- **Opis mogućih odstupanja:**

  2.a Neispravni unos korisničkog imena i/ili lozinke
  1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za prijavu
  2. Korisnik unosi ispravne podatke te završava unos ili odustaje od prijave

UC03 - Odjava s korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Izlazak iz korisničkog sučelja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je obavio prijavu na korisnički račun
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za odjavu
  2. Korisnik se preusmjerava na početnu stranicu

UC04 - Kreiranje gradskog ureda

- **Glavni sudionik:** Administratori
- **Cilj:** Kreiranje gradskog ureda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima administrator ovlasti
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za kreiranje gradskog ureda
  3. Korisnik unosi potrebne podatke i potvrđuje unos
  4. Gradski ured se upisuje u bazu podataka i korisnik se preusmjerava na početnu stranicu
- **Opis mogućih odstupanja:**

  3.a Odabir već zauzetog imena gradskog ureda
  1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za kreiranje gradskog ureda

2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od kreiranja gradskog ureda

3.b Nije odabrana nijedna kategorija oštećenja kojom će gradski ured upravljati

1. Sustav obavještava korisnika o obveznom unosu barem jedne kategorije oštećenja kojom će ured upravljati
2. Korisnik odabire barem jednu kategoriju oštećenja te završava unos ili odustaje od kreiranja gradskog ureda

UC05 - Uklanjanje gradskog ureda

- **Glavni sudionik:** Administrator
- **Cilj:** Uklanjanje gradskog ureda
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima administrator ovlasti, gradski ured postoji
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled gradskih ureda
  3. Korisnik odabire gradski ured
  4. Korisnik odabire opciju za uklanjanje gradskog ureda
  5. Gradski ured se uklanja iz baze podataka i korisnik se preusmjerava na stranicu za pregled gradskih ureda

UC06 - Dodjela uloge službenika gradskog ureda

- **Glavni sudionik:** Administrator
- **Cilj:** Dodjeljavanje uloge službenika gradskog ureda registriranom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima administratorske ovlasti, postoji odgovarajući gradski ured za registriranog korisnika
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled popisa korisničkih računa
  3. Korisnik odabire korisnički račun
  4. Korisnik odabire opciju za dodjelu uloge službenika gradskog ureda
  5. Registrirani korisnik dobiva ulogu službenika gradskog ureda i korisnik se preusmjerava na pregled korisničkog računa

UC07 - Pregled korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregledavanje korisničkog račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je obavio prijavu na korisnički račun
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled vlastitog korisničkog računa

UC08 - Uređivanje podataka korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Promjena podataka na vlastitom korisničkom računu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je obavio prijavu na korisnički račun
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled vlastitog korisničkog računa
  3. Korisnik odabire opciju za uređivanje korisničkih podataka
  4. Korisnik uređuje korisničke podatke i potvrđuje promjene
  5. Korisnički podaci se ažuriraju u bazi podataka i korisnik se preusmjerava na pregled vlastitog korisničkog računa
- **Opis mogućih odstupanja:**

  4.a Odabir već zauzetog korisničkog imena i/ili e-mail adrese, unos korisničkog podatka u neispravnom formatu, nepodudaranje lozinki
  1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na uređivanje korisničkih podataka
  2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od uređivanja korisničkih podataka

UC09 - Brisanje korisničkog računa

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Uklanjanje vlastitog korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je obavio prijavu na korisnički račun
- **Opis osnovnog tijeka:**

1. Korisnik otvara padajući izbornik
2. Korisnik odabire opciju za pregled vlastitog korisničkog računa
3. Korisnik odabire opciju za brisanje korisničkog računa
4. Korisnik povrđuje odluku i korisnik se preusmjerava na početnu stranicu

UC10 - Kreiranje anonimne prijave oštećenja

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Prijava oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu oštećenja
  2. Korisnik unosi potrebne podatke i potvrđuje unos
  3. Prijava oštećenja se upisuje u bazu podataka i korisnik dobiva jedinstveni broj prijave
- **Opis mogućih odstupanja:**

  2.a Neuneseni obvezni podaci i/ili podaci uneseni u neispravnom formatu
  1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za prijavu oštećenja
  2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od prijave oštećenja

UC11 - Kreiranje prijave oštećenja

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prijava oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu oštećenja
  2. Korisnik unosi potrebne podatke i potvrđuje unos
  3. Prijava oštećenja se upisuje u bazu podataka i korisnik se preusmjerava na početnu stranicu
- **Opis mogućih odstupanja:**

  2.a Neuneseni obvezni podaci i/ili podaci uneseni u neispravnom formatu

1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za prijavu oštećenja
2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od prijave oštećenja

UC12 - Provjera statusa anonimne prijave oštećenja

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Provjera statusa anonimne prijave oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Podnesena prijava oštećenja
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za provjeru statusa prijave oštećenja
  2. Korisnik unosi jedinstveni broj prijave i potvrđuje unos
- **Opis mogućih odstupanja:**

  2.a Jedinstveni broj prijave ne postoji ili je unesen u neispravnom formatu
  1. Sustav obavještava korisnika o neuspjelom unosu i vraća ga na stranicu za provjeru statusa prijave oštećenja
  2. Korisnik unosi ispravni jedinstveni broj prijave i završava unos ili odustaje od provjere statusa prijave oštećenja

UC13 - Provjera statusa prijave oštećenja

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Provjera statusa vlastite prijave oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Podnesena prijava oštećenja, korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za provjeru statusa prijave oštećenja
  2. Korisnik odabire prijavu oštećenja za koju želi provjeriti status oštećenja

UC14 - Pregled prijave oštećenja

- **Glavni sudionici:** Registrirani korisnik, Neregistrirani korisnik
- **Cilj:** Pregled prijave oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik na karti odabire oznaku prijave oštećenja

2. Korisniku se otvara pregled prijavljeno oštećenje

UC15 - Promjena statusa prijave oštećenja

- **Glavni sudionik:** Službenik gradskog ureda
- **Cilj:** Promjena statusa prijave oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** Podnesena prijava oštećenja, korisnik ima ulogu službenika gradskog ureda
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz popisa prijavljenih oštećenja
  2. Korisnik odabire prijavu oštećenja
  3. Korisnik odabire opciju za promjenu statusa prijave i potvrđuje promjenu
  4. Status prijave oštećenja se ažurira u bazi podataka i korisnik se preusmjerava na pregled prijave oštećenja

UC16 - Provjera statistike statusa prijave oštećenja

- **Glavni sudionici:** Neregistrirani korisnik, Registrirani korisnik
- **Cilj:** Provjera statistike statusa prijave oštećenja
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za provjeru statistike statusa prijava oštećenja
  2. Korisniku se otvara pregled statistike statusa prijava oštećenja

UC17 - Objedinjavanje nepovezanih prijava oštećenja

- **Glavni sudionik:** Službenik gradskog ureda
- **Cilj:** Objediniti nepovezane prijave oštećenja po temi i lokaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoje prijave oštećenja s istom temom i lokacijom, korisnik ima ulogu službenika gradskog ureda
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prikaz popisa prijavljenih oštećenja
  2. Korisnik odabire opciju za objedinjavanje prijava oštećenja
  3. Korisnik odabire dvije ili više prijava oštećenja i potvrđuje odabir

4. Prijave oštećenja se objedinjuju u bazi podataka i korisnik se preusmjerava na početnu stranicu

- **Opis mogućih odstupanja:**

3.a Korisnik je odabrao manje od dvije prijave oštećenja i pokušao napraviti objedinjavanje

1. Sustav obavještava korisnika o neuspjelom pokušaju objedinjavanja i vraća ga na odabir prijava oštećenja
2. Korisnik odabire više od dvije prijave oštećenja i potvrđuje odabir ili odustaje od objedinjavanja prijava oštećenja

UC18 - Dodjela kategorije za upravljanje gradskom uredu

- **Glavni sudionik:** Administrator
- **Cilj:** Dodjela kategorije oštećenja kojom će određeni gradski ured upravljati
- **Sudionici:** Baza podataka
- **Preduvjet:** Gradski ured postoji, korisnik ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Korisnik odabire padajući izbornik
  2. Korisnik odabire opciju pregleda gradskih ureda
  3. Korisnik odabire gradski ured
  4. Korisnik odabire opciju da dodavanje kategorija oštećenja,
  5. Korisnik radi odabir kategorija i potvđuje odabir
  6. Ažurirani popis kategorija oštećenja za koje je zadužen gradski ured se upisuje u bazu podataka i korisnik se preusmjerava na pregled profila gradskog ureda
- **Opis mogućih odstupanja:**

4.a Gradski ured već upravlja svim kategorijama oštećenja, ne postoji kategorija oštećenja kojom već ne upravlja neki gradski ured u istom mjestu

1. Sustav obavještava korisnika da ne postoji kategorija oštećenja kojom gradski ured već ne upravlja ili da ne postoji kategorija kojom već ne upravlja neki gradski ured u istom mjestu
2. Korisnik odustaje od dodjele kategorije za upravljanje gradskom uredu

UC19 - Uklanjanje kategorije za upravljanje iz gradskog ureda

- **Glavni sudionik:** Administrator
- **Cilj:** Uklanjanje kategorije oštećenja kojom trenutno upravlja gradski ured
- **Sudionici:** Baza podataka

- **Preduvjet:** Gradski ured postoji, korisnik ima ulogu administratora
- **Opis osnovnog tijeka:**
  1. Korisnik odabire padajući izbornik
  2. Korisnik odabire opciju pregleda gradskih ureda
  3. Korisnik odabire gradski ured
  4. Korisnik odabire opciju da uklanjanje kategorija oštećenja
  5. Korisnik radi odabir kategorija i potvđuje ga
  6. Ažurirani popis kategorija oštećenja za koje je zadužen gradski ured se upisuje u bazu podataka i korisnik se preusmjerava na pregled profila gradskog ureda
- **Opis mogućih odstupanja:**

  5.a Korisnik je odabrao sve kategorije oštećenja
  1. Sustav obavještava korisnika da je odabrao sve kategorije oštećenja i vraća ga na stranicu za uređivanje gradskog ureda s otvorenom opcijom odabira kategorija
  2. Korisnik odabire kategorije pazeći da ostavi barem jednu kategoriju i potvrđuje odabir ili odustaje od uklanjanja kategorija oštećenja kojima upravlja gradski ured

UC20 - Ponovno postavljanje zaboravljene lozinke

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Stvaranje nove lozinke za registriranog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i njegovi podaci se nalaze u bazi podataka
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za prijavu
  2. Korisnik odabire opciju za ponovno postavljanje zaboravljene lozinke
  3. Korisnik unosi e-mail adresu korisničkog računa i potvrđuje unos
  4. Korisnik dobiva personalizirani link na stranicu gdje mijenja lozinku
  5. Korisnik unosi novu lozinku i potvrđuje odabir
  6. Nova lozinka sprema se u bazu podataka i korisnik se preusmjerava na početnu stranicu
- **Opis mogućih odstupanja:**

  3.a Unesena e-mail adresa ne postoji ili je u neispravnom formatu
  1. Sustav obavještava korisnika da e-mail adresa ne postoji ili je u neispravnom formatu

2. Korisnik unosi ispravnu e-mail adresu ili odustaje od ponovnog postavljanja zaboravljene lozinke

4.a Uneseni identifikacijski broj ne postoji ili je u neispravnom formatu
1. Sustav obavještava korisnika da identifikacijski broj ne postoji ili je u neispravnom formatu
2. Korisnik unosi ispravni identifikacijski broj ili odustaje od ponovnog postavljanja zaboravljene lozinke

5.a Unesena lozinka je u neispravnom formatu, lozinke se ne podudaraju
1. Sustav obavještava korisnika da je unesena lozinka u neispravnom formatu ili da se lozinke ne podudaraju
2. Korisnik unosi ispravno lozinku ili odustaje od ponovnog postavljanja zaboravljene lozinke

UC21 - Dodjela uloge administratora

- **Glavni sudionik:** Administrator
- **Cilj:** Dodjeljavanje uloge administratora registriranom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima administratorske ovlasti
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled popisa korisničkih računa
  3. Korisnik odabire korisnički račun
  4. Korisnik odabire opciju za dodjelu uloge administratora
  5. Registrirani korisnik dobiva ulogu administratora i korisnik se preusmjerava na pregled korisničkog računa

UC22 - Uklanjanje uloge službenika gradskog ureda

- **Glavni sudionik:** Administrator
- **Cilj:** Uklanjanje uloge službenika gradskog ureda registriranom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima administratorske ovlasti
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled popisa korisničkih računa
  3. Korisnik odabire korisnički račun
  4. Korisnik odabire opciju za uklanjanje uloge administratora

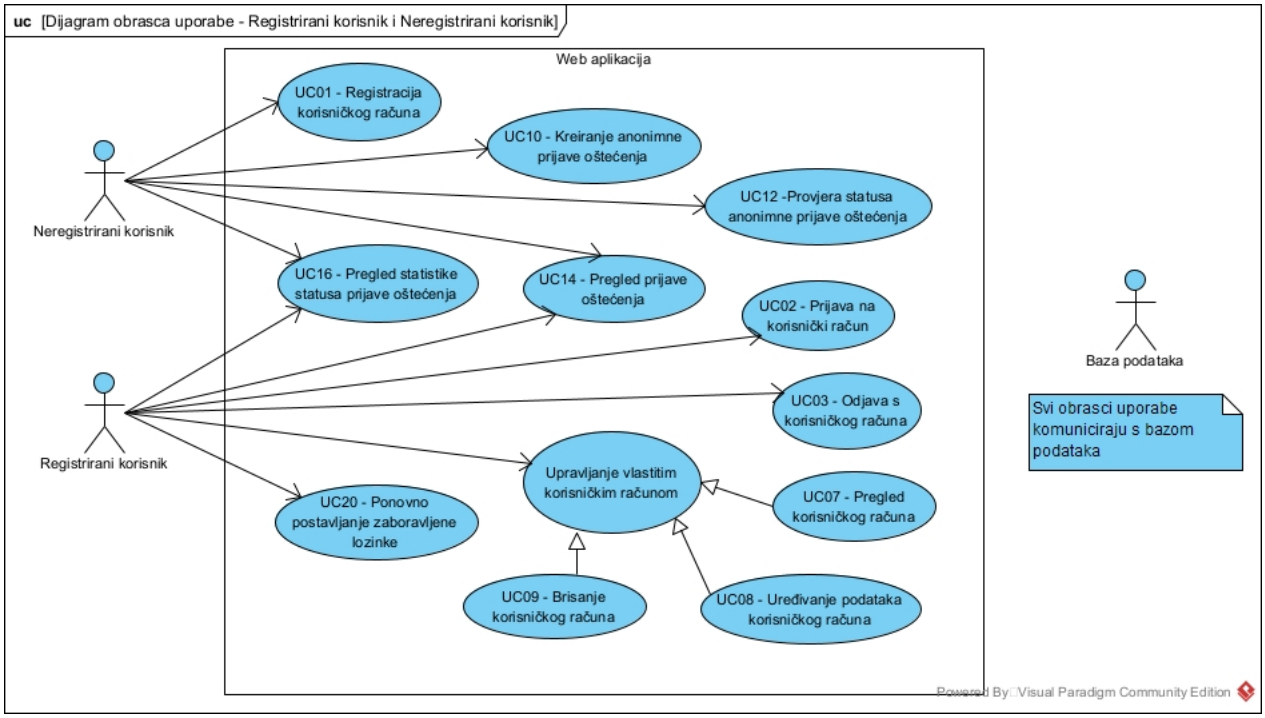
5. Registrirani korisnik ostaje bez uloge administratora i korisnik se preusmjerava na pregled korisničkog računa

UC23 - Uklanjanje uloge administratora

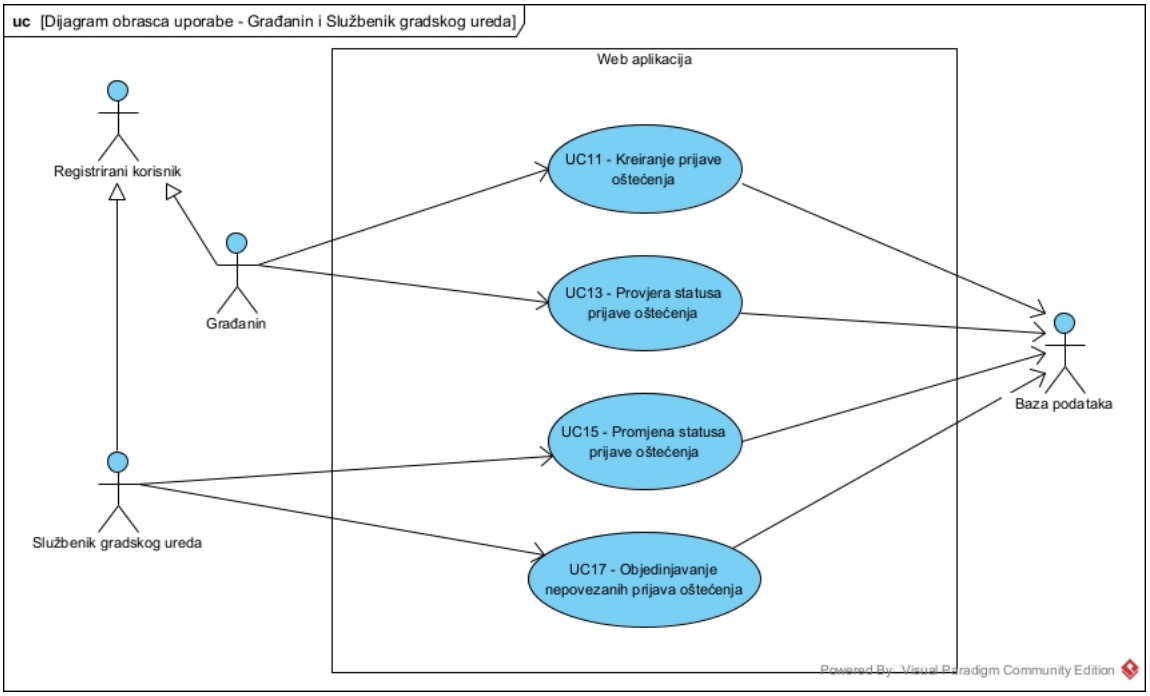
- **Glavni sudionik:** Administrator
- **Cilj:** Uklanjanje uloge administratora registriranom korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i ima administratorske ovlasti
- **Opis osnovnog tijeka:**
  1. Korisnik otvara padajući izbornik
  2. Korisnik odabire opciju za pregled popisa korisničkih računa
  3. Korisnik odabire korisnički račun
  4. Korisnik odabire opciju za uklanjanje uloge administratora
  5. Registrirani korisnik ostaje bez uloge administratora i korisnik se preusmjerava na pregled korisničkog računa

Dijagrami obrazaca uporabe

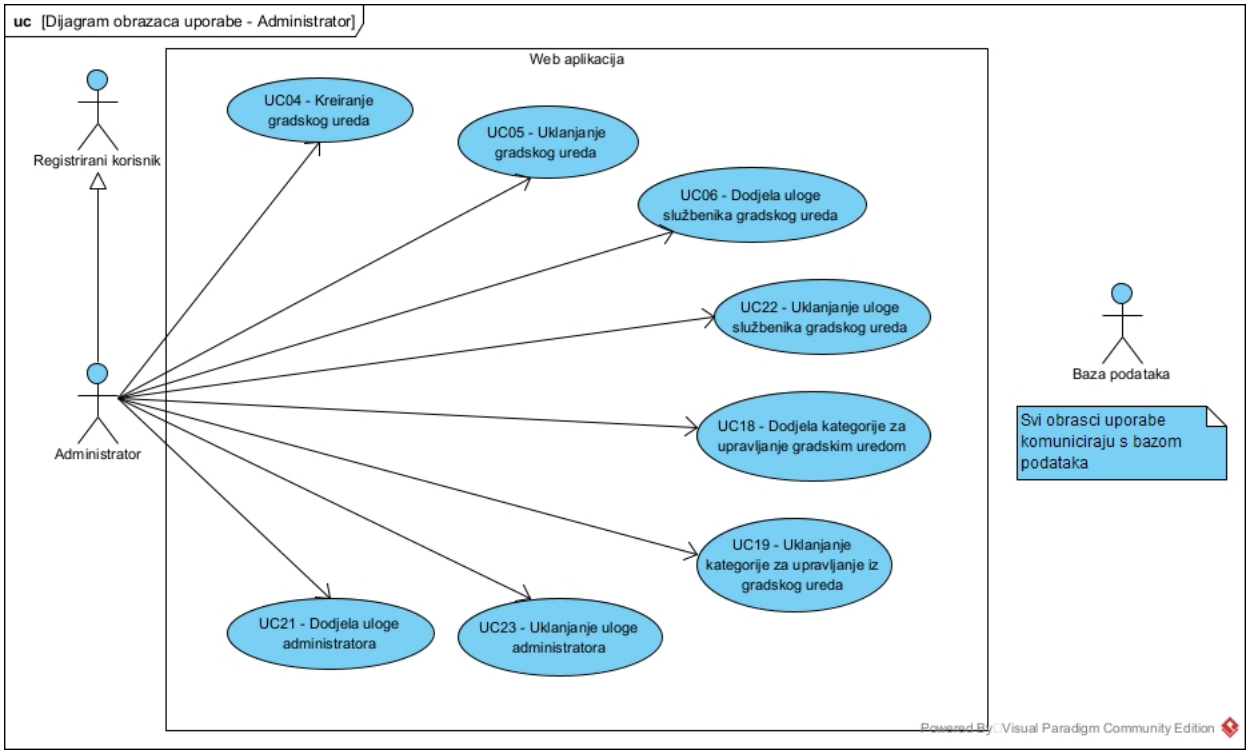
Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost registriranog i neregistriranog korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost građanina i službenika gradskog ureda



Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

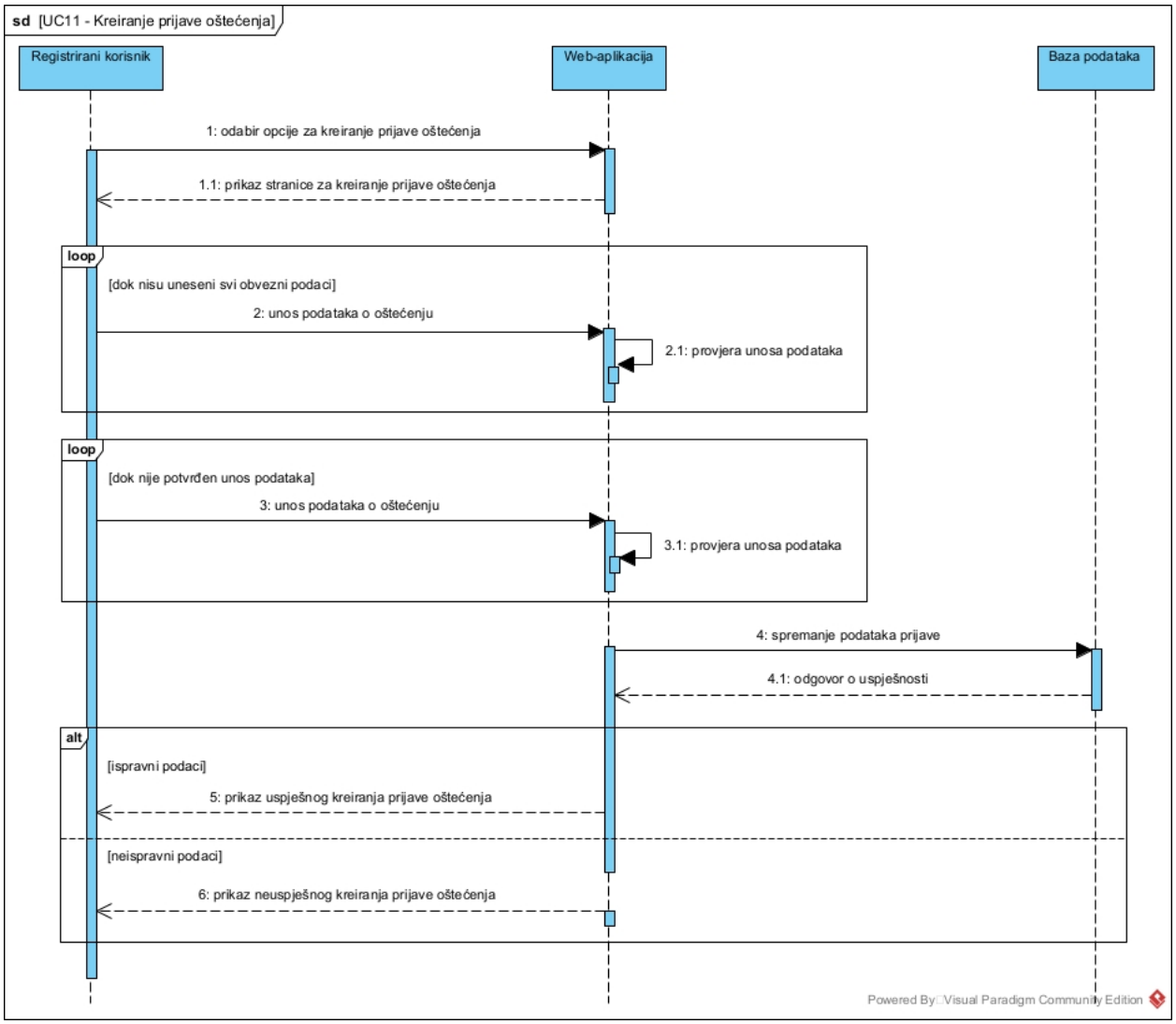
3.1.2 Sekvencijski dijagrami

dio 1. revizije

Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava (max. 4 dijagrama). Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.

Obrazac uporabe UC11 - Kreiranje prijave oštećenja

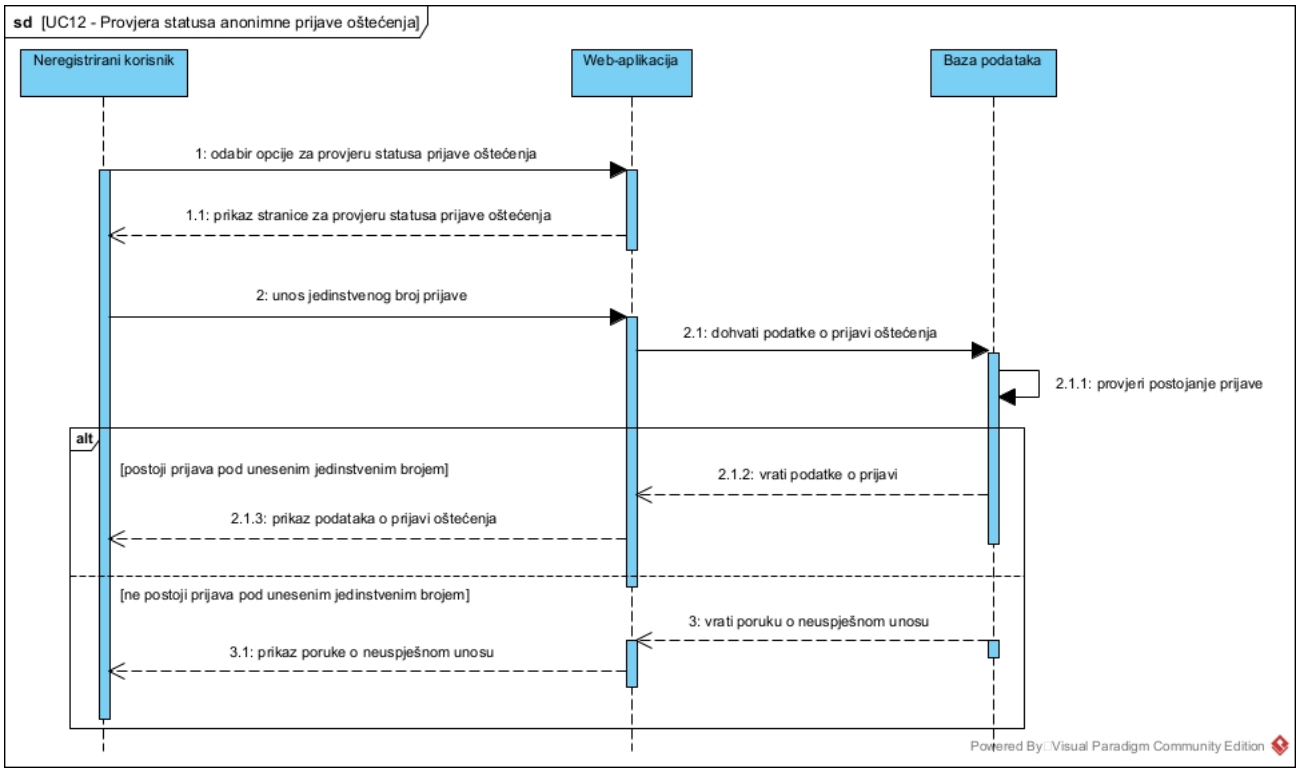
Registrirani korisnik šalje zahtjev za prikaz opcije kreiranja prijave oštećenja. Poslužitelj prikazuje stranica za kreiranje prijave oštećenja na kojoj korisnik unosi podatke o oštećenju. Postoje obvezni i neobvezni podaci. Dok korisnik ne unese sve obvezne podatke ne prikazuje se opcija za potvrdu unosa podataka. Kada unese sve obvezne podatke prikaže se opcija za potvrđivanja unosa podataka. Korisnik odlučuje hoće li unijeti neobvezne podatke poput fotografije oštećenja. Potvrđeni podaci spremaju se u bazu podataka. Ukoliko su podaci ispravni, poslužitelj prikazuje prikaz uspješnog kreiranja prijave oštećenja, a ukoliko nisu, poslužitelj prikazuje prikaz neuspješnog kreiranje prijave oštećenja s obrazloženjem razloga.



Slika 3.4: Sekvencijski dijagram za UC11

Obrazac uporabe UC12 - Provjera statusa anonimne prijave oštećenja

Neregistrirani korisnik šalje zahtjev za prikaz opcije za provjeru statusa anonimne prijave oštećenja. Poslužitelj prikazuje stranicu za pregled statusa anonimne prijave oštećenja. Korisnik unosi jedinstveni broj prijave. Poslužitelj provjerava postoji li prijava oštećenja s tim jedinstvenim brojem u bazi podataka. Ukoliko postoji, poslužitelj prikazuje podatke o oštećenju, a ukoliko ne postoji, poslužitelj prikazuje poruku o neuspješnom unosu jedinstvenog broja prijave.

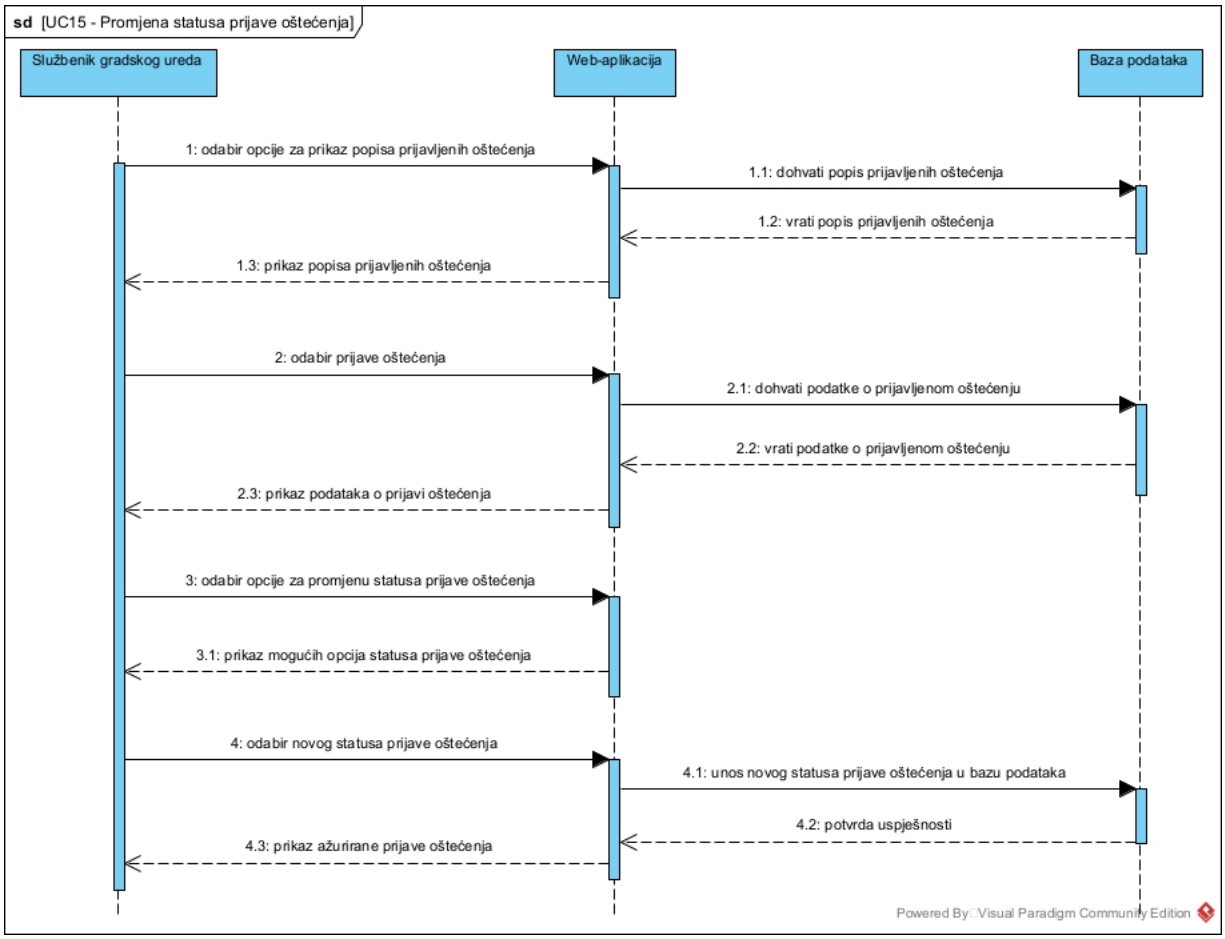


Slika 3.5: Sekvencijski dijagram za UC12

Obrazac uporabe UC15 - Promjena statusa prijave oštećenja

Službenik gradskog ureda šalje zahtjev za prikaz popisa prijavljenih oštećenja. Poslužitelj dohvaća popis prijavljenih oštećenja iz baze podataka te ih prikazuje službeniku. Službenik odabire prijavu oštećenja te time šalje zahtjev za prikaz podataka o toj prijavi oštećenja. Poslužitelj dohvaća podatke o prijavi oštećenja te ih prikazuje službeniku. Službenik šalje zahtjev za prikaz opcije za promjenu statusa prijave oštećenja. Poslužitelj prikazuje opciju za promjenu statusa prijave oštećenja s navedenim mogućim opcijama stanja prijave oštećenja. Službenik odabire novi status prijave oštećenja te šalje zahtjev za promjenu statusa poslužitelju. Poslužitelj ažurira status prijave oštećenja u bazi podataka. Poslužitelj prikazuje službeniku

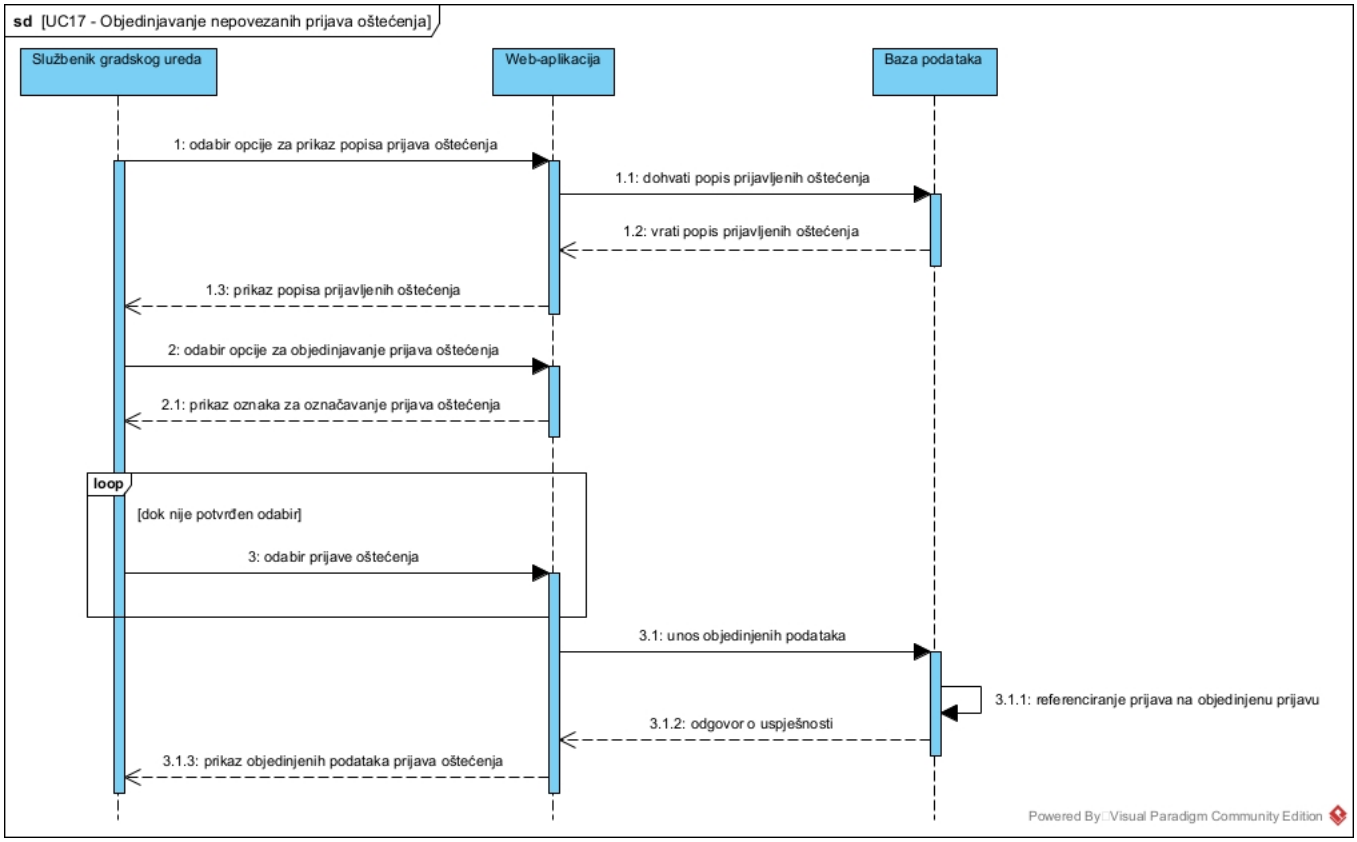
potvrdu o uspješnosti promjena statusa prijave oštećenja.



Slika 3.6: Sekvencijski dijagram za UC15

Obrazac uporabe UC17 - Objedinjavanje nepovezanih prijava oštećenja

Službenik gradskog ureda šalje zahtjev za prikaz popisa prijavljenih oštećenja. Poslužitelj dohvaća popis prijavljenih oštećenja iz baze podataka te ih prikazuje službeniku. Službenik odabire opciju za objedinjavanje prijava oštećenja. Poslužitelj prikazuje oznake za označavanje prijava oštećenja. Službenik odabire prijave oštećenja. Službenik potvrđuje odabir prijava oštećenja za objedinjavanje. Poslužitelj unosi podatke o objedinjenim prijavama oštećenja u bazu podataka. Događa se referenciranje objedinjene prijave oštećenja na postojećih prijava oštećenja. Po završetku, poslužitelj prikazuje službeniku gradskog ureda podatke objedinjene prijave oštećenja.



Slika 3.7: Sekvencijski dijagram za UC17

3.2 Ostali zahtjevi

dio 1. revizije

Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju ***kako se sustav treba ponašati*** *i koja* ***ograničenja*** *treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.*

1. Sustav treba podržavati rad više korisnika u stvarnom vremenu
2. Sustav treba ispravno funkcionirati na svim web preglednicima
3. Sustav treba biti izveden kao web aplikacija prilagođena mobilnom uređaju i tablet računalu

4. Korisničko sučelje i sustav podržavaju hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
5. Sustav koristi hrvatski jezik i srednjoeuropsko standardno vrijeme, GMT+1
6. Korisničko sučelje treba biti jednostavno za korištenje bez opširnih uputa
7. Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
8. Sustav ne smije omogućiti registraciju korisnika i promjenu lozinke dok nije unesena dovoljno jaka lozinka duljine od barem 8 znakove te barem jedno malo slovo, veliko slovo, znamenku i specijalni znak
9. Sustav sprema lozinku u sigurnom obliku različitom od običnog tekstualnog formata, koristeći bcrypt hash
10. Sustavu se pristupa s javne mreže pomoću HTTPS
11. Pristupanje bazi podataka ne smije trajati duže od nekoliko sekundi
12. Sustav ne smije javno prikazivati ime i prezime podnositelja prijave oštećenja
13. Buduće nadogradnje sustava ne smiju narušiti postojeće funkcionalnosti sustava

4. Arhitektura i dizajn sustava

dio 1. revizije

Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremišta podataka, mrežne protokole, globalni upravljački tok i sklopovsko-programske 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)*

Pri analizi projektnih zahtjeva i detaljnom razmatranju uloga dionika te njihovih interakcija unutar aplikacije, odlučili smo strukturirati naš sustav na tri ključne razine: razinu klijenta, razinu web aplikacije i razinu baze podataka. Unutar ove podjele, nužno je uključiti slojeve korisničkog sučelja, aplikacijske logike i pristupa podacima. Kao model arhitekture, odabrali smo višeslojnu strukturu sličnu MVC (Model-View-Controller) stilu. MVC je oblik arhitekture softvera koji organizira aplikaciju u tri komponente:

- *Model (poslovna logika i podaci)*
- *View (korisničko sučelje)*
- *Controller (upravljač, posrednik između Modela i Viewa)*

Razdvajanje logike, prezentacije i upravljanja omogućuje jednostavno održavanje i razvoj aplikacije, a promjene u jednoj komponenti ne bi trebale značajno utjecati na druge. Ova arhitektura, bazirana na klijent-poslužitelj odnosu, omogućuje jasno definiranu organizaciju slojeva, a s ciljem maksimalne ponovne uporabivosti, integrirali smo i različite programske biblioteke i radne okvire. Razvojni tim je pisao

kod u razvojnom okruženju IntelliJ IDEA i Visual Studio Code, a za pokretanje, konfiguraciju i puštanje u pogon, kao i neovisnost o računalu na kojem se kod izvršava, odabrali smo platforme Docker, Render i Node.JS.

Arhitektura web aplikacije "CestaFix" može se podijeliti na tri podsustava:

- *Web poslužitelj*
- *Web aplikacija*
- *Baza podataka*

Web poslužitelj je komponenta koja pruža podršku za backend sustav aplikacije i čija je ključna uloga omogućiti komunikaciju između klijenta i aplikacije. Ova interakcija odvija se putem HTTP (Hyper Text Transfer Protocol) protokola, standardnog protokola za prijenos informacija na webu. Pokretanje web aplikacije i prosljeđivanje zahtjeva aplikaciji radi daljnje obrade, inicira se upravo pomoću web poslužitelja. Za implementaciju web poslužitelja korišten je Spring Boot, Java framework, koji se ističe brzim konfiguriranjem i implementacijom web aplikacija temeljenih na Javi. Sastavljen od Model i Controller slojeva prema MVC arhitekturi, gdje se poslovna logika, kao što je upravljanje prijavama šteta, nalazi u Modelu. Web poslužitelj također omogućuje definiranje i implementaciju RESTful API-ja, što omogućuje komunikaciju između frontenda (klijenta) i backenda (poslužitelja) putem HTTP/HTTPS mrežnih protokola. Primarni zadatak Controllera unutar web poslužitelja je obrađivanje HTTP zahtjeva koji pristižu iz frontend dijela aplikacije, izvršavanje odgovarajuće funkcionalnosti te slanje odgovora. Dodatno, Spring Boot omogućuje učinkovito upravljanje stanjem aplikacije, uključujući praćenje stanja sesija za registrirane korisnike.

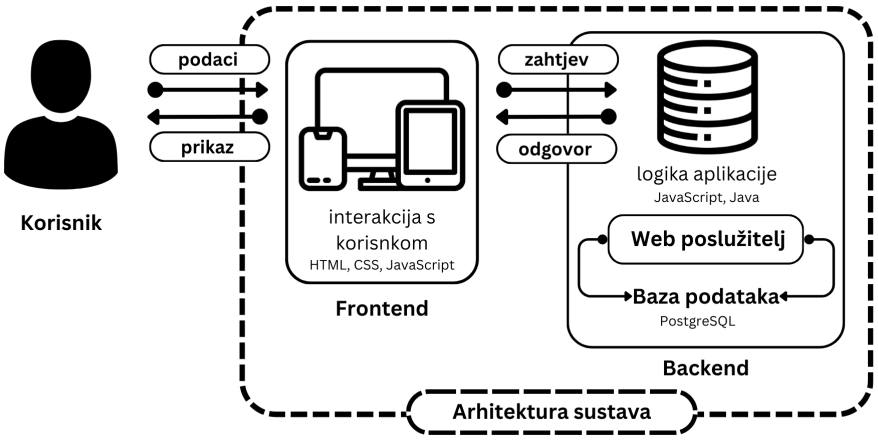
Web preglednik je softver koji djeluje kao posrednik između poslužitelja i klijenta, omogućujući korisniku prikaz web sadržaja. Osnovna funkcionalnost web preglednika ostvarena je pri slanju HTTP zahtjeva poslužitelju te prijemu i interpretaciji HTTP odgovora. Web preglednici djeluju kao prevoditelji, omogućavajući korisnicima vizualizaciju web sadržaja kroz sučelje preglednika, dekodiranjem informacija dobivenih iz HTTP odgovora.

Web aplikacija Web aplikacija je kompleksni softverski sustav koji se sastoji od frontend i backend dijelova:

- **Frontend** (Klijent) predstavlja korisničko sučelje putem kojeg korisnici ostvaruju interakciju sa sustavom. Ostvaren je korištenjem JavaScript programskog jezika, za potrebu upravljanja događajima na korisničkom sučelju, i React radnog okvira, što omogućuje kreiranje dinamičnog i intuitivnog korisničkog sučelja, čime se ostvaruje ugodno korisničko iskustvo. Frontend je strukturiran u komponente, što olakšava održavanje i ponovnu uporabu koda. Komunicira s backendom kroz HTTPS zahtjeve te pritom omogućuje prijenos podataka i ažuriranje informacija o prijavama šteta. Kroz klijentsku logiku, frontend provodi validaciju unesenih podataka kako bi osigurao ispravnost prije slanja na backend. Responsivni dizajn osigurava konzistentno korisničko iskustvo na različitim uređajima.

- **Backend** (Poslužitelj) je dio sustava unutar kojeg se obrađuju zahtjevi i izvršavaju daljnje radnje. Kako bi se postiglo ”razdvajanje zabrinutosti”, organiziran je na kontrolere, servise i repozitorije. Controlleri imaju ključnu ulogu u obradi ulaznih zahtjeva (HTTP zahtjeva) omogučujući organizaciju i upravljanje tokom rukovanja zahtjevima u backend dijelu aplikacije kao i pozivanje odgovarajućih metoda u Modelu te slanje odgovora klijentskom dijelu. Servisi u backendu na učinkovit i organiziran način obavljaju poslovnu logiku i specifične funkcionalnosti koje su potrebne za obradu zahtjeva, koji dolaze s frontend dijela aplikacije. Repozitoriji imaju ključnu ulogu u komunikaciji s bazom podataka, odnosno omogućuju servisima da abstrahiraju detalje interakcije s bazom podataka, pružajući im jednostavan i konzistentan način komunikacije s podacima. Backend je ostvaren korištenjem Java programskog jezika i Spring Boot radnog okvira. Također, pruža RESTful API-je koji omogućuju komunikaciju između klijenta i servera te definiraju kako resursi (poput prijava šteta) mogu biti stvoreni, ažurirani i dohvaćeni. Osim toga backend sadrži i DTO-e (Data Transfer Objects) za prijenos podataka između različitih dijelova sustava odnosno slojeva aplikacije.

Baza podataka je podatkovni sloj koji se koristi za sigurnu pohranu podataka te je detaljnije opisana u sljedećem poglavlju.



Slika 4.1: Arhitektura sustava

4.1 Baza podataka

dio 1. revizije

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

Sustav je temeljen na uporabi relacijske baze podataka implementirane u PostgreSQL-u gdje su entiteti modelirani kao tablice koje posjeduju svoje jedinstveno ime i skup atributa. Odabir relacijske baze podataka proizlazi iz potrebe za lakšim ostvarenjem naših potreba za upravljanjem podacima pri prijavljivanju oštećenja i njihovoj sanaciji odnosno kako bismo što jednostavnije modelirali sustav prema stvarnom svijetu. Ova baza podataka ključna je za sigurnost podataka i brz pristup, pohranu, umetanje, izmjenu te dohvat podataka koje sustav koristi za daljnju obradu. Baza podataka ove aplikacija sadrži sljedeće entitete:

- *Users*
- *Reports*
- *CityDep*
- *Category*
- *Problems*
- *CitydepCategory*

4.1.1 Opis tablica

Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ
Users je entitet koji sadrži sve bitne informacije o korisnicima i njihovim ulogama unutar aplikacije. Sastoji se od atributa: user_id, name, email, password i role. Povezan je vezom *Many-to-One* s entitetom CityDep preko atributa citydep_id i vezom *One-to-Many* s entitetom Reports preko atributa user_id.

Users		
user_id	INT	jedinstveni identifikator korisnika
username	VARCHAR	ime korisnika
email	VARCHAR	e-mail adresa korisnika
password	VARCHAR	hash lozinke korisnika
role	VARCHAR	uloga korisnika
citydep_id	INT	jedinstveni identifikator gradskog ureda

Reports je entitet koji sadrži sve bitne informacije o prijavama oštećenja. Sastoji se od atributa: report_id, user_id, title, description, address, photo, report_time, status i problem_id. Povezan je vezom *Many-to-One* s entitetom Problems preko atributa problem_id i vezom *Many-to-One* s entitetom Users preko atributa user_id.

Reports		
report_id	INT	jedinstveni identifikator prijave
user_id	INT	jedinstveni identifikator korisnika (users.user_id)
title	VARCHAR	naziv prijave/oštećenja
description	TEXT	opis oštećenja
address	VARCHAR	adresa oštećenja
photo	BYTEA	slika oštećenja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Reports		
report_time	TIMESTAMP	vrijeme podnošenja prijave
status	VARCHAR	status prijave
problem_id	INT	jedinstveni identifikator oštećenja (problems.problem_id)

Citydep je entitet koji sadrži sve bitne informacije o gradskim uredima. Sastoji se od atributa: citydep_id i citydep_name. Povezan je vezom *One-to-Many* s entitetom Users preko atributa citydep_id i vezom *One-to-Many* s entitetom CityDepCategory preko atributa citydep_id.

Citydep		
citydep_id	INT	jedinstveni identifikator gradskog ureda
citydep_name	VARCHAR	naziv gradskog ureda

Category je entitet koji sadrži sve bitne informacije o kategoriji oštećenja. Sastoji se od atributa: category_id i category_name. Povezan je vezom *One-to-Many* s entitetom Problems preko atributa category_id i vezom *One-to-Many* s entitetom CityDepCategory preko atributa category_id.

Category		
category_id	INT	jedinstveni identifikator kategorije oštećenja
category_name	VARCHAR	naziv kategorije oštećenja

Problems je entitet koji sadrži sve bitne informacije o prijavljenom oštećenju. Sastoji se od atributa: problem_id, longitude, latitude, status i category_id. Povezan je vezom One-to-Many s entitetom Reports preko atributa problem_id i vezom *Many-to-One* s entitetom Category preko atributa category_id.

Problems		
problem_id	INT	jedinstveni identifikator oštećenja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

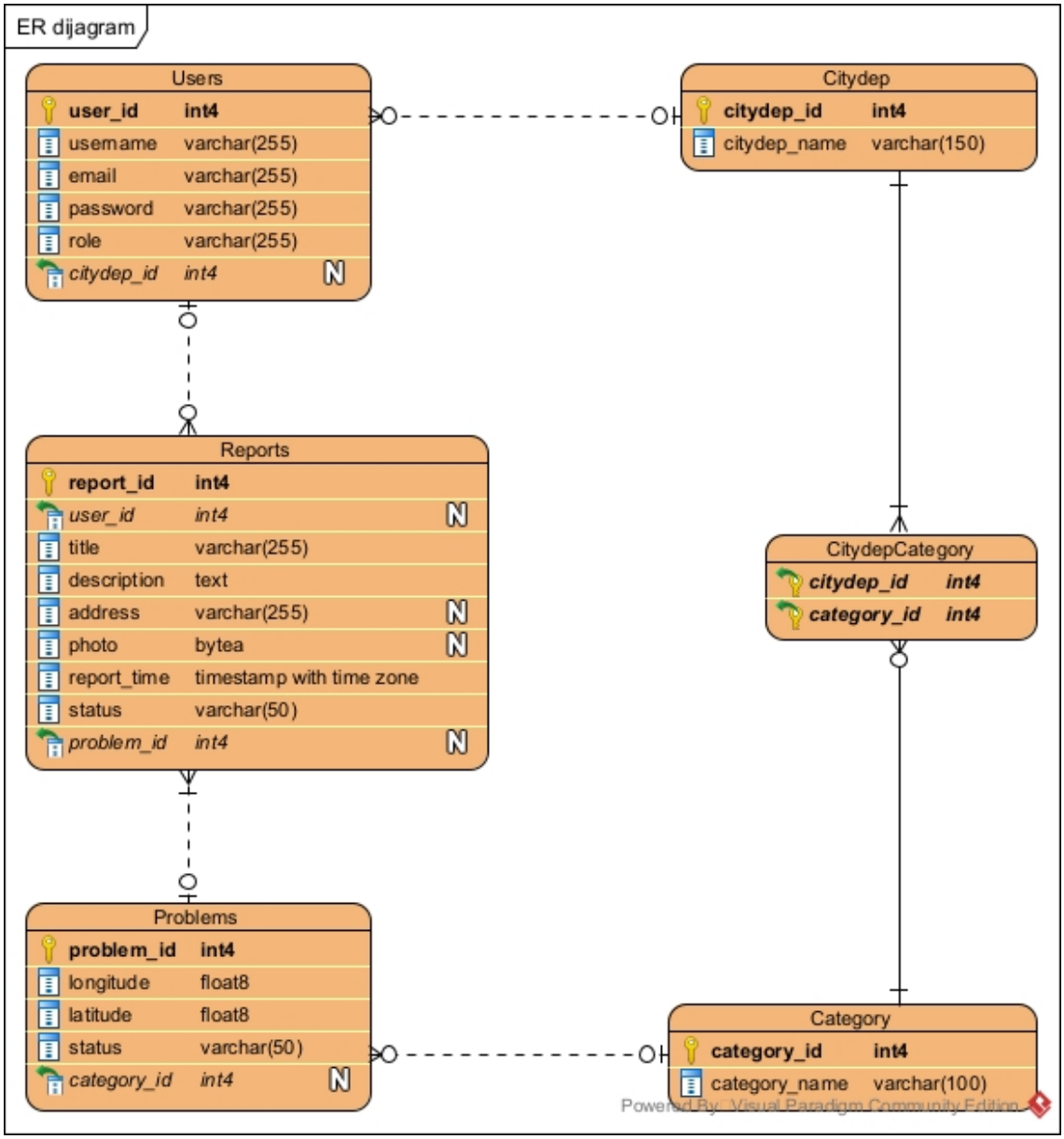
Problems		
longitude	DOUBLE	geografska dužina lokacije oštećenja
latitude	DOUBLE	geografska širina lokacije oštećenja
status	VARCHAR	status oštećenja
category_id	INT	jedinstveni identifikator kategorije oštećenja (category.category_id)

CitydepCategory je entitet koji sadrži sve bitne informacije vezane uz kategoriju oštećenja kojom se bavi određeni gradski ured. Sastoji se od atributa: citydep_id i category_id. Povezan je vezom *Many-to-One* s entitetom CityDep preko atributa citydep_id i vezom *Many-to-One* s entitetom Category preko atributa category_id.

CitydepCategory		
citydep_id	INT	jedinstveni identifikator gradskog ureda (citydep.citydep_id)
category_id	INT	jedinstveni identifikator kategorije oštećenja (category.category_id)

4.1.2 Dijagram baze podataka

U ovom potpoglavlju potrebno je umetnuti dijagram baze podataka. Primarni i strani ključevi moraju biti označeni, a tablice povezane. Bazu podataka je potrebno normalizirati. Podsjetite se kolegija "Baze podataka".



Slika 4.2: E-R dijagram baze podataka

4.2 Dijagram razreda

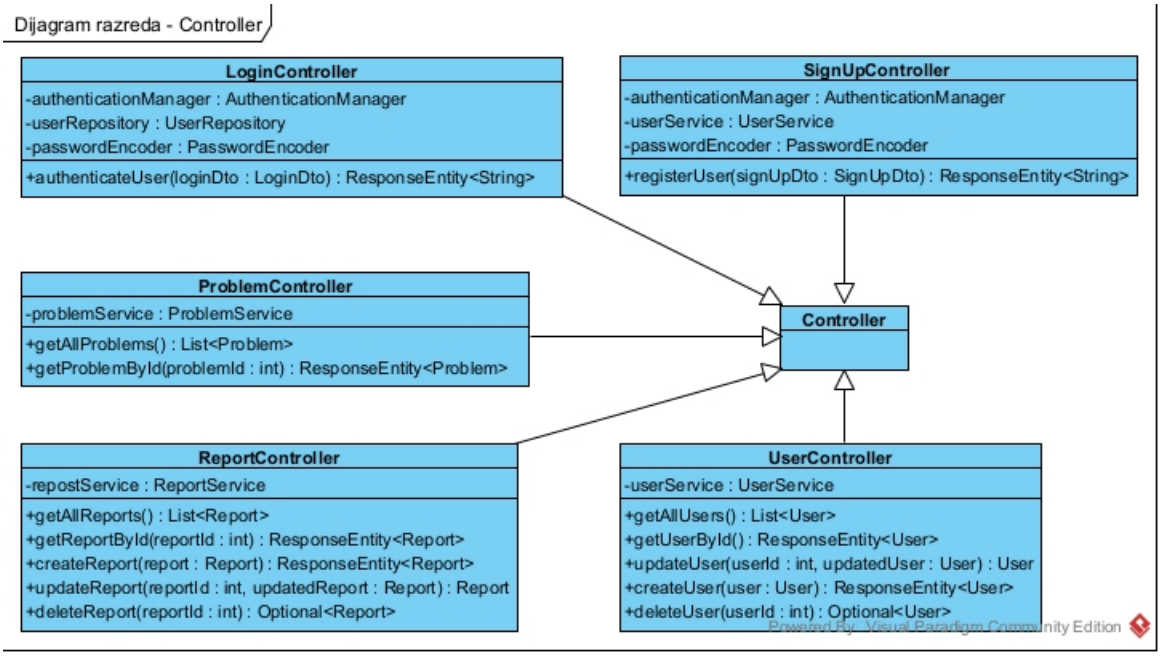
Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.

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.*

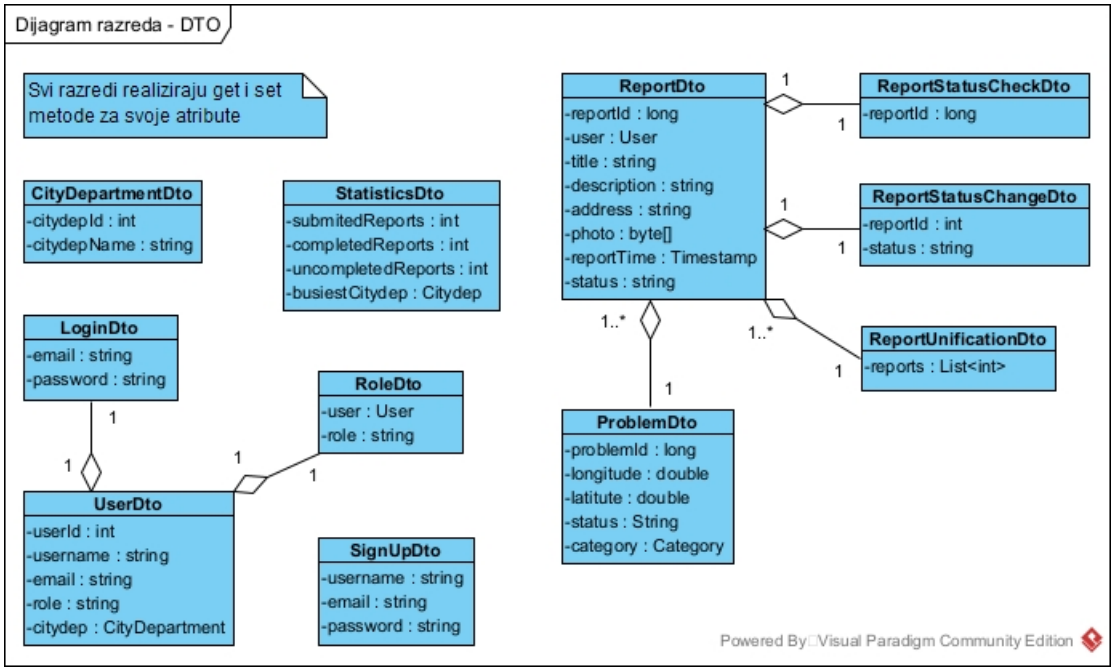
Na slikama 4.3, 4.4 i 4.5 su prikazani razredi koji pripadaju *backend* dijelu s arhitekturom podijeljenom na kontrolere, repozitorije i servise te uključuje DTO (Data Transfer Object) i modele.

Na slici 4.3 prikazani su razredi koji nasljeđuju Controller razred. Metode implementirane u tim razredima upravljaju i rukuju DTO-ima koji su dohvaćeni pomoću metoda implementiranih u Model razredima. Omogućuju logiku interakcije i promjene. Kontroler LoginController omogućuje prijavu registriranog korisniak u sustav, dok kontroler SignUpController omogućuje registraniju neregistriranog korisnika. Kontroler ReportController omogućuje interakcije s prijavom oštećenja, dok kontroler ProblemController omogućuje interakcije s objedinjenim problemima oštećenja s istom temom i lokacijom. Kontroler UserController omogućuje interakcije i promjene povezane s registriranim korisnicima.



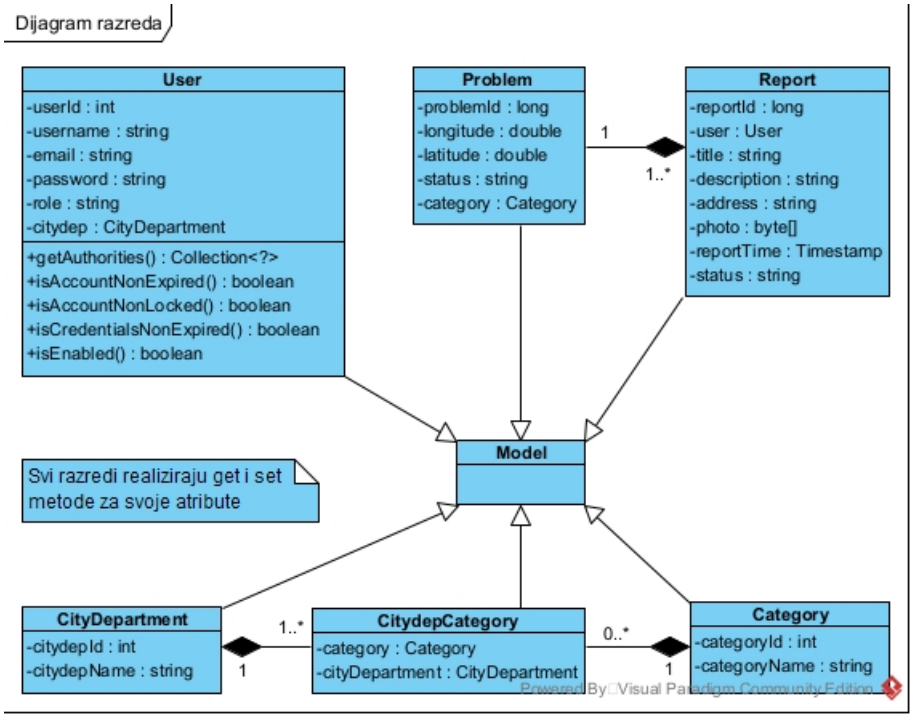
Slika 4.3: Dijagram razreda - dio Controllers

Na slici 4.4 prikazani su razredi koji pripadaju DTO. Data Transfer Objects omogućavaju razmjenu podataka između procesa i slojeva. LoginDto i SignUpDto služe za stvaranje novog objekta za prijavu, odnosno registraciju korisnika. ReportDto i ProblemDto služe za stvaranje novog objekta prijave oštećenja, odnosno problema. ReportStatusCheckDto i ReportStatusChangeDto služe za stvaranje objekta po kojem će se omogućiti provjera statusa prijave, odnosno promjena statusa prijave oštećenja. ReportStatusChangeDto služi za stvaranje objekta po kojem će se izvesti objedinjavanje prijava oštećenja. StatisticsDto služi za dogvaćanje podataka o statistici. UserDto služi da dohvaćanje podataka o korisničkom računu, dok RoleDto služi za stvaranje objekta po kojem će se izvesti promjena uloge registriranog korisnika. CityDepartmentDto služi za stvaranje objekta gradskog ureda.



Slika 4.4: Dijagram razreda - dio DTO

Na slici 4.5 prikazi su razredi koji pripadaju Model razredima. Oni preslikavaju strukturu baze podataka u aplikaciji. Komunicijaru s bazom podataka te vraćaju podatke iz nje. Razred User predstavlja registriranog korisnika koji je unio nužne podatke u sustav. Razred Report predstavlja prijavu oštećenja za koju je korisnik unio potrebne i opcionalne podatke. Razred Problem predstavlja prijavu problema koja predstavlja jednu ili više prijava oštećenja objedinjene u prijavu problema sa zajedničkom temom i lokacijom. Razred Category predstavlja kategoriju oštećenja koju korisnik može odabrati za prijavljeno oštećenje, odnosno koju gradski ured može dobiti na upravljanje oštećenjima. Razred CityDepartment predstavlja gradski ured koji upravlja jednom ili više kategorija oštećenja koje im je administrator dodijelio. Razred CitydepCategory predstavlja vezu između razreda Category i razreda CityDepartment, odnosno predstavlja zapis kategorija i gradskih ureda u parovima.



Slika 4.5: Dijagram razreda - dio DTO

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 tijek 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 Selenium[1]*. 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. Programsko inženjerstvo, FER ZEMRIS, http://www.fer.hr/predmet/proinz
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, "Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, http://www.ece.rutgers.edu/~marsic/books/SE
5. N. Frid, A. Jović, "Modeliranje programske potpore UML-dijagramima", Fakultet elektrotehnike i računarstva Sveučilišta u Zagrebu, https://www.fer.unizg.hr/_download/repository/UML_zadaci_za_vjezbu_-_nadopuna_sveucilisnog_prirucnika%5B1%5D.pdf
6. Visual Paradigm, https://www.visual-paradigm.com/
7. Tehničko predavanje o backendu, https://gitlab.com/hrvojesimic/progi-project-teams-backend/
8. Tehničko predavanje o frontendu, https://gitlab.com/jtomic/opp-project-teams-frontend
9. Tehničko predavanje o deploymentu, https://github.com/progi-devops

Indeks slika i dijagrama

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak
   - Datum: 17. listopada 2023.
   - Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
   - Teme sastanka:
     - formiranje tima
     - dogovor načina komunikacije
     - dogovor oko korištenih tehnologija
2. sastanak
   - Datum: 19. listopada 2023.
   - Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
   - Teme sastanka:
     - dogovor podjele poslova u početnoj fazi
     - analiza projektnog zadatka
3. sastanak
   - Datum: 24. listopada 2023.
   - Prisustvovali: Fran Fodor, Vedran Knežević, Ante Prkačin
   - Teme sastanka:
     - sastanak s nastavnikom
     - upoznavanje s temom projekta
4. sastanak
   - Datum: 27. listopada 2023.

- Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
- Teme sastanka:
  - definiranje funkcijskih zahtjeva
  - definiranje baze podataka

5. sastanak
   - Datum: 31. listopada 2023.
   - Prisustvovali: Fran Fodor, Mateo Jakšić, Sara Podvorec, Ante Prkačin
   - Teme sastanka:
     - sastanak s nastavnikom
     - pojašnjenje zadatka
6. sastanak
   - Datum: 6. studenoga 2023.
   - Prisustvovali: Fran Fodor, Vedran Knežević, Ante Prkačin
   - Teme sastanka:
     - razrada baze podataka
7. sastanak
   - Datum: 7. studenoga 2023.
   - Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc
   - Teme sastanka:
     - sastanak s nastavnikom
     - pojašnjenje dokumentacije
8. sastanak
   - Datum: 13. studenoga 2023.
   - Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
   - Teme sastanka:
     - analiza dosadašnjeg napretka
     - postavljanje daljnjih ciljeva
9. sastanak
   - Datum: 14. studenoga 2023.
   - Prisustvovali: Fran Fodor, Mateo Jakšić, Sara Podvorec
   - Teme sastanka:
     - sastanak s nastavnikom

– pregled ostvarenog

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Fran Fodor	**Mateo Jakšić**	**Vedran Kneževi**	**Leon Sattvik Kolenc**	**Jan Murić**	**Sara Podvorec**	**Ante Prkačin**
Upravljanje projektom	15	13	12	11	10	12	12
Opis projektnog zadatka							
Funkcionalni zahtjevi	1	2	1	1	1	1	1
Opis pojedinih obrazaca		8					
Dijagram obrazaca		6					
Sekvencijski dijagrami		6					
Opis ostalih zahtjeva		2					
Arhitektura i dizajn sustava							
Baza podataka		5					
Dijagram razreda		8					
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Fran Fodor	Mateo Jakšić	Vedran Knežević	Leon Sattvik Kolenc	Jan Murić	Sara Podvorec	Ante Prkačin
Upute za puštanje u pogon							
Dnevnik sastajanja		1					
Zaključak i budući rad							
Popis literature		1					
Učenje Springa			5				12
Učenje Reacta	3			5			
Učenje Tailwinda	3						
Izrada dizajna stranice	2			4			
Pisanje README.md	1	1		1	1	1	1
Rad na arhitekturi baze	4	1	4	1	1	1	4
Implementacije arhitekture baze			2				
Prikaz mape i funckionalnost markera	4			3			
Izrada prijave i registracije korisnika				7			
Prikazivanje prijava oštećenja				5			
Organizacija backenda							5
Sigurnost - Spring Security							6
Testiranje API-a							4

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Fran Fodor	**Mateo Jakšić**	**Vedran Knežević**	**Leon Sattvik Kolenc**	**Jan Murić**	**Sara Podvorec**	**Ante Prkačin**
Priprema za deployment i deployment	3			4			2

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.