Programsko inženjerstvo

Ak. god. 2023./2024.

Gradska šteta

Dokumentacija, Rev. *1*

Grupa: *VelicanstveniTimRaketa*

Koordinator: *Ivan Šimunić*

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	Napravljen predložak.	Ivan Šimunić	26.10.2023.
0.2	Dodan opis projektnog zadatka.	Nino Ćurko	4.11.2023.
0.3	Dodani aktori i njihovi funkcijski zahtjevi	Nino Ćurko	7.11.2023.
0.3	Dodan dio obrazaca uporabe	Nino Ćurko	7.11.2023.
0.3.1	Dodani opisi obrazaca uporabe i ostali zahtjevi	Nino Ćurko	9.11.2023.
0.4	Dodani dijagrami obrazaca uporabe	Nino Ćurko	14.11.2023.
0.4.1	Dodani sekvencijski dijagram	Nino Ćurko, Josip Šare, Davor Najev, Dominik Zoričić	13.11.2023.
0.5	Dodan opis korištene arhitekture te opisa baze podataka i njezinih tablica	Nino Ćurko	16.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.5.1	Dodan prototip verzije dijagrama razreda	Nino Ćurko, Josip Šare	3.12.2023.

2. Opis projektnog zadatka

U ovo moderno doba i vrijeme, kada su tehnološka sredstva uvijek nadomak ruke, nekolicina entuzijasta se dosjetila kako doskočiti u pomoć svom gradu. Naime, štete po cestama i javnim provršinama se ne smanjuju, ali to je upravo cilj jednog ovakvog projekta.

Ono što ovaj projekt nastoji ponuditi jest upravo jedno pravo suvremeno i efikasno rješenje pristupačno svima,a usmjereno samo prema jednoj stvari - otklanjanju štete javnih površina u gradu. Cilj projekta je svim ljudima omogućiti jednostavni pristup aplikaciji preko koje će moći prijavljivati uočena oštećenja javnih površina i cesta u područjima gradova čime sveukupno kumulira poboljšanje života u društvu.

Ukratko, tijek zamišljenih događaja je idući: Korisnik uoči oštećenu gradsku imovinu ili cestu te istu želi prijaviti gradskom uredu nadležnom za taj tip štete. Korisnik učitava našu aplikacuju te šalje prijavu sa određenim popunjenim podacima. Gradski ured dobiva obavijest o prijavi te kad istu riješi korisnik koji ju je prijavio dobiva obavijest da je ona razriješena.

Pri učitavanju početne stranice aplikacije korisniku je vidljiva karta, opcija za registraciju (ili prijavu), te opcija za podnošenje prijave. Regsistririani kao i neregistrirani korisnici imaju mogućnost slanja prijave, ali razlika je u tome što neregistrirani korisnik dobiva jedinstveni broj pomoću kojeg prati status prijave, registrirani korisnik uvijek može samo ući u povijest svojih prijava i vidjeti status svake podnesene

Neregistrirani korisnik se u svakom trenutku može regitrirati, te pri registraciji trebe navesti iduće podatke:

- Ime
- Prezime
- Korisničko ime
- E-mail

- Lozinku

Registrirani korisnici se uvijek mogu prijaviti u sustav pomoću svog korisničkog imena i lozinke. Ako je korisnik prijavljen u sustav, ima mogućnost odjave ponuđenu u gornjem desnom kutu. Registrirani korisnici su podijeljeni na iduć uloge:

- Klijent
- Administrator
- Gradski ured

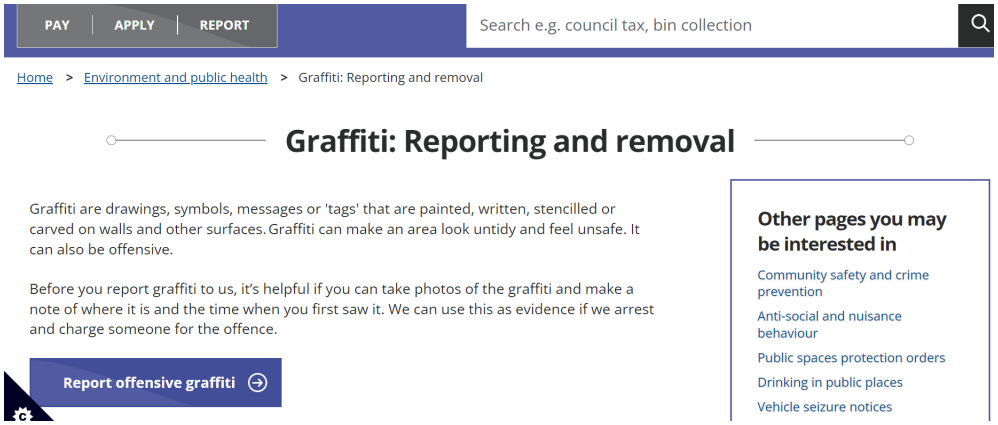
Klijent na karti ima označene sve aktivne prijave. Pri pregledu prijava, prijave može filtrirati po tematici (tipu) i lokaciji. Pri predaji prijave, korisnik mora unijeti odgovarajuće podatke. Također korisniku nije zabranjeno mijenjanje vlastitih osobnih podataka ako vidi potrebu za tim.

Prijava koja se šalje u sustav se sastoji od atributa:

- Naziv
- Kratki opis
- Geografske koordinate

U slučaju da korisnik šalje prijavu koja je na obližnjoj lokaciji neke već druge prijave poslane u sustav sustav ga o tome obavještava tako da mu izbacuje obližnju prijavu i ponudu mu opciju nadovezivanja na tu istu u slučaju da korisnik uoči kako je to upravo ta prijava koju je on htio poslati.Uz sve to, korisnik ocionalno može poslati i sliku viđenog i potom sustav, u slučaju da nisu unesene koordinate, langitudu i longitudu izvlači sa meta podataka fotografije.

Web stranica koja nudi sličnu uslugu, ali je bazirana na vandalizmu i isključivo grafitima u Londonu je:
`https://www.newham.gov.uk/publichealth-safety/graffiti-reporting-removal`,
a izgled je na slici 2.1.

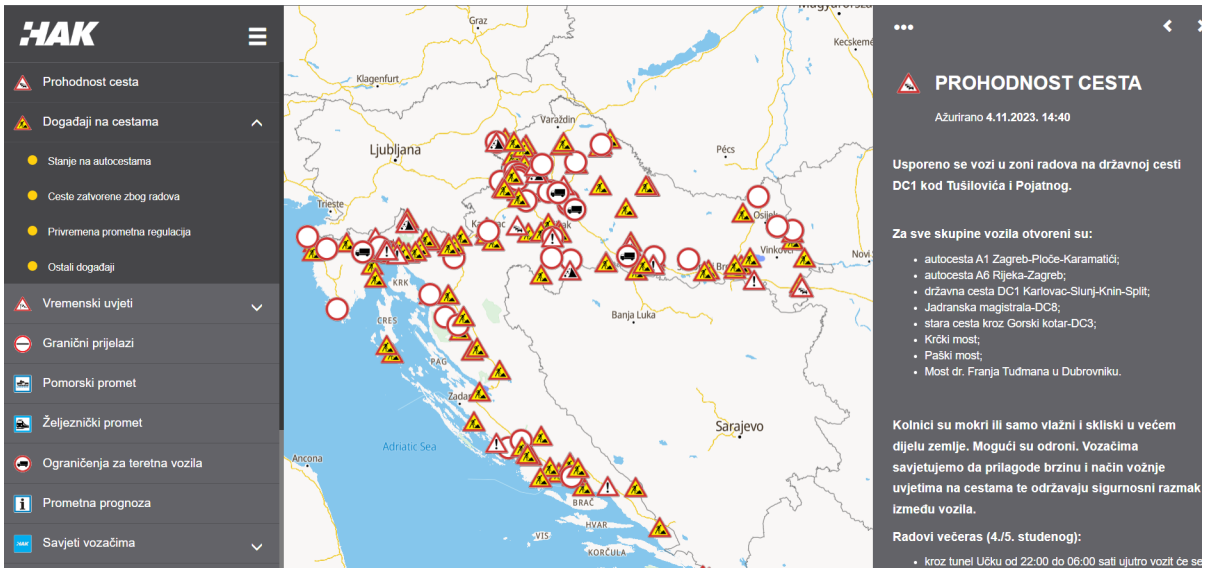


Slika 2.1: Izgled navedene stranice

Administrator je uloga koja ima najveće ovlasti. On je tu da kontrolira sve vezano za stranicu pa tako ima mogućnost uređivanja podataka pojedinih prijava kojen smatra nevaljalima kao i njihovo brisanje iz baze podatka. Pored toga, može brisati profile regsitriranih korsinika za koje procjeni da krše pravila ponašanja na aplikaciji, te potvrđuje ili opvrgava registraciju novih gradskih ureda za koje procijeni validnost.

Uz navedene korisnike sustava, još postoje i gradski uredi/vijeća koji primaju prijave na temelju tematike problema za koju su zaduženi. Tako će puknuće na cestama primati isključivo javni zavod za ceste, probleme sa zgradama će obrađivati ured za izgradnju i prostorno uređenje, probleme sa vodovodom ured za vodovod... Dodatno, svaki gradski ured uvijek ima mogućnost promjene statusa prijave koji će prijaviteljima omogućiti da jasno vide ukoliko je prijava razriješena ili nije. Uz to, ured može povezati one prijave koje korisnici nisu povezali ukoliko shvati da se radi o istom problemu iste tematike na konkretno bliskom području. Nadalje, ured prihavaća zahtjeve registriranih korisnika za ulazak u ured, to jest praticipaciju kako bi korisnik (koji je zaposlenik u tom uredu) dobio ovlasti gradskog ureda. Kreiranje (registracija) novih ureda u sustavu mora biti odobrena od strane administratora.

Sustav će sve prijave obrađivati u stvarnom vremenu pa će korisnici u bilo kojem trenutku iamti uvid da li je npr. neko puknuće na cesti razriješeno te da li se tom cestom promet uopće može odvijati. Sličnu mogućnost nudi stranica HAK-a na idućoj adresi : `https://www.hak.hr/info/stanje-na-cestama/#prohodnost-cesta` na slici refhak.



Slika 2.2: HAK - stanje na cestama

Ovakva aplikacije sama po sebi već ima klijentelu i predispozicije za uspješno korištenje i popularizaciju. Uz štete javnih površina i cesta moglo bi se dodati prijava za zastoj primjerice tramvaja na određenoj lokaciji (naravmo u gradovima gdje je tramvajski prijevoz omogućen) kako bi bilo vidljivo svim korisnicima aplikacije u stvarnom vremenu. Uz to moglo bi se područje aplikacije proširiti na prometne nesreće kako bi korisnici uz status da li je cesta zatvorena ili ne, mogli uz sliku procijeniti prohodnost iste. Neke vizualne stvari za korisnike koje se mogu još dodati bi bile gledanje tuđih profila, kao i dopisivanje porukama kako bi se doznale konkretnije informacije o prijavi od osobe koja je tu prijavu napravila. Besmisleno je izostaviti je Hrvatska turistička zemlja, te aplikaciji bi se još mogli dodati ostali osnovni jezici za inozemne korisnike. Kako bi se što više promoviralo prijavljivanje šteta, u aplikaciju bi se dala ugraditi neka online nagrađivanja; primjerice dostupnost povećanja levela i stjecanja points-a.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Regitrirani korisnik
   (a) Klijent
   (b) Administrator
2. Neregistrirani (anonimni) korisnik
3. Razvojni tim
4. Gradski ured

Aktori i njihovi funkcionalni zahtjevi:

1. Klijent (inicijator) može:
   (a) Prijaviti se u sustav koristeći svoj email i lozinku
   (b) Uređivati osobne podatke
   (c) Poslati prijavu u sustav
      - Unijeti naziv za prijavu, opis prijave, geografske koordinate, te opcionalno fotografiju
      - Odabrati koordinate preko karte ili unijeti najbližu adresu
      - Povezati svoju prijavu na postojeću (ako takva postoji)
   (d) Pregledati prijave i podatke vezane za njih
      - Odabrati aktivnu prijavu na karti
      - Pregledati povijest svojih prijava
      - Filtrirati pregled prijava po lokaciji i temi
   (e) Registrirati novi gradski ured
   (f) Zatražiti ulazak za participaciju u određeni gradski ured
   (g) Mijenjati podatke aktivne prijave

2. Administrator (inicijator) može:
   (a) Pregledati popis svih prijava ikad napravljenih u sustavu
      - Uređivati podatke prijava
      - Brisati prijave
   (b) Pregledati i uklanjati registrirane profile po potrebi
   (c) Odobravati kreaciju novih gradskih ureda
3. Gradski ured (inicijator) može:
   (a) Pregledati popis aktivnih zaprimljenih prijava
      - Prihvatiti (ili odbiti) određenu prijavu
      - Spojiti nepovezane prijave
      - Promijeniti status prihvaćene prijave ovisno o uspješnosti odrađene te iste prijave
   (b) Prihvatiti (ili odbiti) zahtjev korisnika za ulazak u taj ured
   (c) Poslati izvješće o odrađenoj prijavi na E-mail klijent
4. Neregistrirani korisnik (incijator) može:
   (a) Registrirati se u sustav koristeći ime, prezime, mail, username i lozinku
   (b) Poslati prijavu u sustav
      - Unijeti naziv za prijavu, opis prijave, geografske koordinate, te opcionalno fotografiju
      - Odabrati koordinate preko karte ili unijeti najbližu adresu
      - Povezati svoju prijavu na postojeću (ako takva postoji)
      - Zaprimiti jedistveni ID za podnesenu prijavu
5. Baza podataka (sudionik) može:
   (a) Pohraniti sve podatke o korisnicima i njihovim ovlastima
   (b) Pohraniti svaku prijavu sa koreliranim podacima za istu

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC01 - Registracija korisnika u sustav

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Klijent bira opciju ”registracija” na sučelju web aplikacicje
  2. Klijent unosi tažene podatke
  3. Korisnik je upisan u bazu podataka
- **Opis mogućih odstupanja:**

  2.a Klijent unosi neispravni/postojeći username ili email
  1. Sustav obavještava korisnika o problemu i briše mu unesena polja
  2. Korisnik mijenja podatke u ispravne i registracija uspješno se privede kraju

UC02 - Unos nove prijave u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Podnijeti novu prijavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik upisuje tražene podatke pri unosu prijave
  2. Sustav javlja ako postoji vremenski bliska prijava na toj lokaciji
  3. Korisnik može povezati svoju prijavu na postojeću (ako takva postoji)
  4. Prijava se predaje i zapisuje u sustav
- **Opis mogućih odstupanja:**

  1.a Korisnik nije naveo sve zahtijevane podatke
  1. Sustav obavještava korisnika o problemu i javlja mu da popuni tražena polja

UC03 - Pregled aktivnih prijava u sustavu

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled postojećih prijava
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
    1. Korisnik otvara pregled svih postojećih prijava
    2. Korisniku se nudi opcija filtriranja po temi i lokaciji
    3. Na sučelju se prikazuju filtritrane prijave

UC04 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prijaviti se svojim profilom u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
    1. Korisnik upisuje korisničko ime i lozinku
    2. Sustav javlja potvrdu ispravnosti unesinih podataka
    3. Korisniku se učitava njemu prilagođeno sučelje
- **Opis mogućih odstupanja:**

    1.a Korisnik krivo unio korisničko ime/lozinku

        1. Sustav obavještava korisnika o problemu i javlja mu da ispravi tražena polja

UC05 - Uređivanje podataka prijave

- **Glavni sudionik:** Administrator
- **Cilj:** Korigirati podatke vezane za odabranu prijavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
  1. Prikazuju se sve prijave
  2. Administrator može filtrirati prijave
  3. Administrator izmjenjuje podatke prijave
  4. Izmjenjena prijava se sprema u bazu podataka

UC06 - Brisanje prijave

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati određenu prijavu iz baze podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
  1. Administratoru se prikazuje pregled prijava
  2. Administrator bira prijavu koju želi izbrisati
  3. Prijava se uklanja iz baze podataka i više nije vidljiva u aplikaciji

UC07 - Pregled podataka registriranih profila korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Uvid u profile registriranih korisnika i njihovih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
  1. Korisnik bira opciju za pregled svih profila
  2. Otvara mu se lista svih registriranih profila zajedno sa njihovim osobnim podacima

UC08 - Pregled liste osobnih podataka vlastitog profila

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati osobne podatke svog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Korisnik ulazi u opis svog profila
  2. Sustav mu prikaže username, e-mail i lozinku

UC09 - Odjava iz sustava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Aktivna prijava
- **Opis osnovnog tijeka:**
  1. Korisnik bira opciju odjava
  2. Sustav ga vraća na početnu stranicu web aplikacije

UC10 - Brisanje korisničkog računa

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati određeni profil
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator bira profil koji želi ukloniti
  2. Sustav ga za provjeru pita da potvrdi odluku
  3. Administrator potvrđuje i profil se uklanja iz baze podataka

UC11 - Pregledavanje povijesti svojih prijava

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati svu povijest privedenih prijava
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Korisnik bira opciju za prikazivanje povijesti prijava
  2. Sustav mu na sučelju prikazuje sve njegove prijave
  3. Korisnik dodatno može filtrirati iste po temi i lokaciji

UC12 - Pregledavanje profila gradskih ureda

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati popis svih postojećih gradskih ureda
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju za pregled svih gradskih ureda zapisanih u sustavu
  2. Sustav mu na sučelje prikazuje gradske urede zapisane u bazi podataka zajedno sa njihovim opisima

UC13 - Povezivanje na postojeću prijavu

- **Glavni sudionik:** Korisnik
- **Cilj:** Vezati se na vremenski blisku prijavu na istom području
- **Sudionici:** Baza podataka
- **Opis osnovnog tijeka:**
  1. Korisnik predaje prijavu
  2. Sustav mu javlja za vremenski blisku prijavu na toj lokaciji
  3. Korisnik bira hoće li se vezati na postojeću prijavu ili kreirati vlastitu

UC14 - Odabir aktivnih prijava sa karte

- **Glavni sudionik:** Klijent
- **Cilj:** Pogledati na karti neriješene prijave i detalje za iste
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje aktivnih prijava
- **Opis osnovnog tijeka:**
  1. Klijent na karti odabire opciju za uvid u aktivne prijave
  2. Sustav mu sve lokacije aktivnih prijava prikazuje na karti
- **Opis mogućih odstupanja:**

  1.a U sustavu nema aktivnih prijava
  1. Sustav korisnika izbacuje iz pregleda karte i javlja mu da nema aktivnih prijava

UC15 - Mijenjanje sadržaja aktivne prijave

- **Glavni sudionik:** Klijent
- **Cilj:** Promijeniti sadržaj vlastite aktivne prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje aktivne prijave
- **Opis osnovnog tijeka:**
  1. Klijent odabire svoju aktivnu prijavu koju želi izmjeniti
  2. Klijent mijenja atribut u prijavi
  3. Prijava s ažuriranim podakom se zapisuje u bazu podataka
- **Opis mogućih odstupanja:**

  1.a U trenutku odabira prijave, prijava više nije aktivna
  1. Sustav korisniku javlja da prijava više nije aktivna
  2. Korisnik je preusmjeren na početnu stranicu

UC16 - Pregled zaprimljenih prijava

- **Glavni sudionik:** Gradski ured
- **Cilj:** Pregledati sve zaprimljene prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje zaprimljenih prijava
- **Opis osnovnog tijeka:**
  1. Gradski ured bira izabire pregled svih aktivnih prijava
  2. Sustav mu iz baze podataka omogući uvid u sve prijave namijenjene njemu

- **Opis mogućih odstupanja:**
  2.a Nepostojanje prijava namijenjenih njemu
    1. Sustav mu javlja da trenutno nema aktivnih prijava

UC17 - Prihvaćanje ili odbijanje zaprimljene prijave

- **Glavni sudionik:** Gradski ured
- **Cilj:** Privatiti ili odbiti prijavu iz pregleda prijava
- **Sudionici:** Baza podataka, Gradski uredi
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Gradski ured u pregledu svih prijava bira za pojedinu prijavu hoće li je prihvatiti ili odbiti
  2. U slučaju prihvaćanja prijave, ta se ista uklanja it popisa aktivnih drugim uredima za koje je bila namijenjena

UC18 - Promjena statusa određene prijave

- **Glavni sudionik:** Gradski ured
- **Cilj:** Promijeniti status prihvaćene prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Prihvat određene prijave
- **Opis osnovnog tijeka:**
  1. Gradski ured za prihvaćaenu prijavu mijenja njen status ovisno je li uspješno odrađena ili ne
  2. Status se zapisuje u bazu podatka
  3. U slučaju neuspješno odrađene prijave, ta se ista vraća u popis aktivnih svim uredima za koje je i prije bila namijenjena

UC19 - Povezivanje nepovezanih prijava

- **Glavni sudionik:** Gradski ured
- **Cilj:** Spojiti više pojedinih prijava u jednu cjelinu
- **Sudionici:** Baza podataka, Korisnik (inicjaitaor prijave)
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Gradski ured iz pregleda prijava odabire prijave koje želi povezati
  2. Svaki prijavitelj čija je prijava povezana na druge sada to vidi u sustavu

UC20 - Prihvaćanje zahtjeva za ulazak registriranog korisnika u gradski ured

- **Glavni sudionik:** Gradski ured
- **Cilj:** Prihvatiti ili odbiti zahtjev kroisnika za ulazak u gradski ured
- **Sudionici:** Baza podataka, Korisnik (inicijator zahtjeva)
- **Preduvjet:** Poslan zahtjev za ulazak u gradski ured
- **Opis osnovnog tijeka:**
  1. Gradski ured gleda poslane zahtjeve vezane za učlanjenje korisnika u taj određeni ured
  2. U slučaju da zahtjevi postoje gradski ured bira hoće li prihvatiti novog člana ili ne
- **Opis mogućih odstupanja:**

  1.a Nepostojanje aktivnih zahtjeva za ulazak u ured
    1. Sustav javlja da trenutno nema poslanih zahtjeva

UC21 - Slanje zahtjeva za ulazak u gradski ured

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Poslati određenom gradskom uredu zahtjev za ulaz u isti
- **Sudionici:** Baza podataka, Gradski ured
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire u tražilici određeni gradski ured
  2. Pri pronalasku traženog ureda korisnik šalje zahtjev za ulaz u isti

UC22 - Registracija novog gradskog ureda

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Kreirati novi gradski ured
- **Sudionici:** Baza podataka, Administrator
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju kreiraj gradski ured
  2. Pri kreiranju novog gradskog ureda unosi podatke za isti
  3. Na temelju podataka administrator i baza podataka prihvaćaju ili odbijaju zahtjev za kreiranjem ureda

UC23 - Uređivanje osobnih podataka

- **Glavni sudionik:** Klijent
- **Cilj:** Promijeniti osobne podatke u bazi podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik bira opciju uređavanje osobnih podataka
  2. Mijenja podatak, te se novi zapisuje u bazu podataka
- **Opis mogućih odstupanja:**

  2.a Unos neispravnog podatka
  1. Baza podatka mu dojavljuje o grešci i traži ga da promijeni unos
  2. Korisnik odustaje od unosa ili prepravlja unos podataka

UC24- Praćanje prijave jedinstvenim ID-om

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Pregledati svou prijavu unosom jedinstvenog broja
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje jedinstvenog broja
- **Opis osnovnog tijeka:**
  1. Neregistrirani korisnik u tražilicu unosi ID koji je dobio pri unosu prijave
  2. Sustav mu pokaže njegovu prijavu i status ako je unesen broj važeći
- **Opis mogućih odstupanja:**

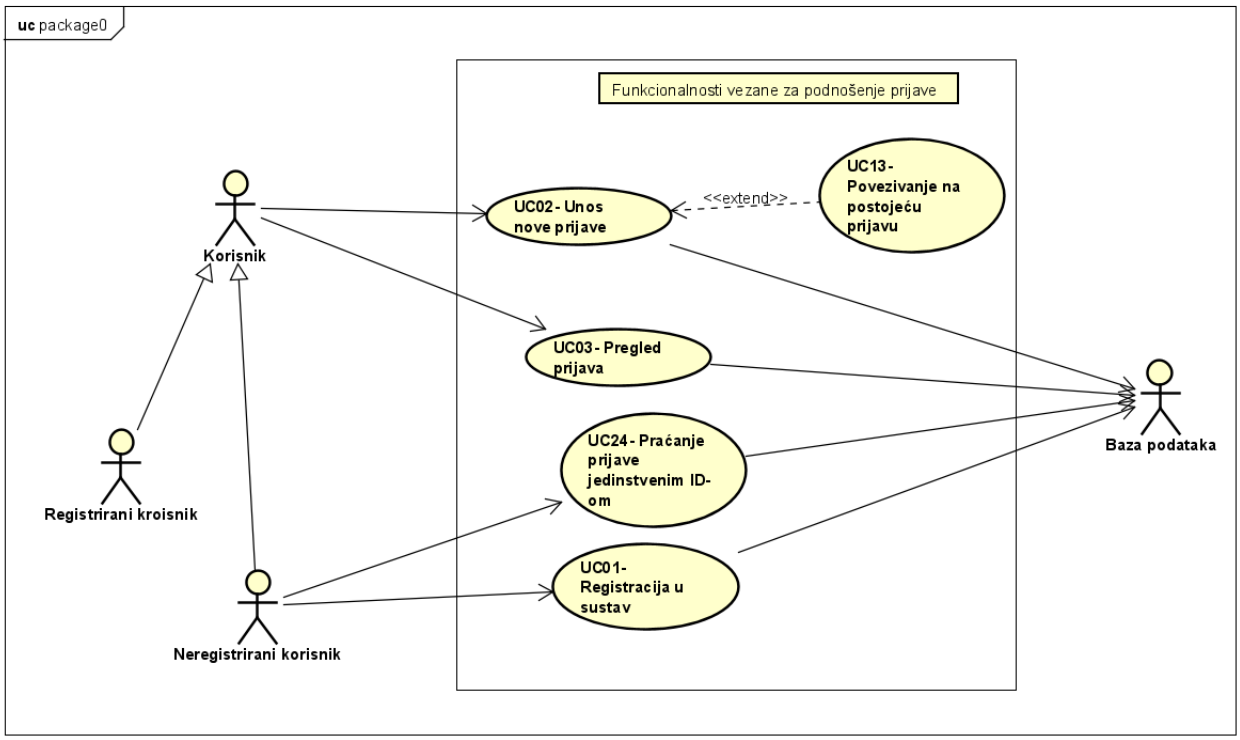
  1.a Unos neispravnog podatka
  1. Baza podatka mu dojavljuje o grešci i traži ga unese ispravan ID ili da odustane od prijave
  2. Korisnik bira opciju odustani ili opet unosi ID

UC25- Slanje obavijesti vezane za odrađenu prijavu na E-mail klijent

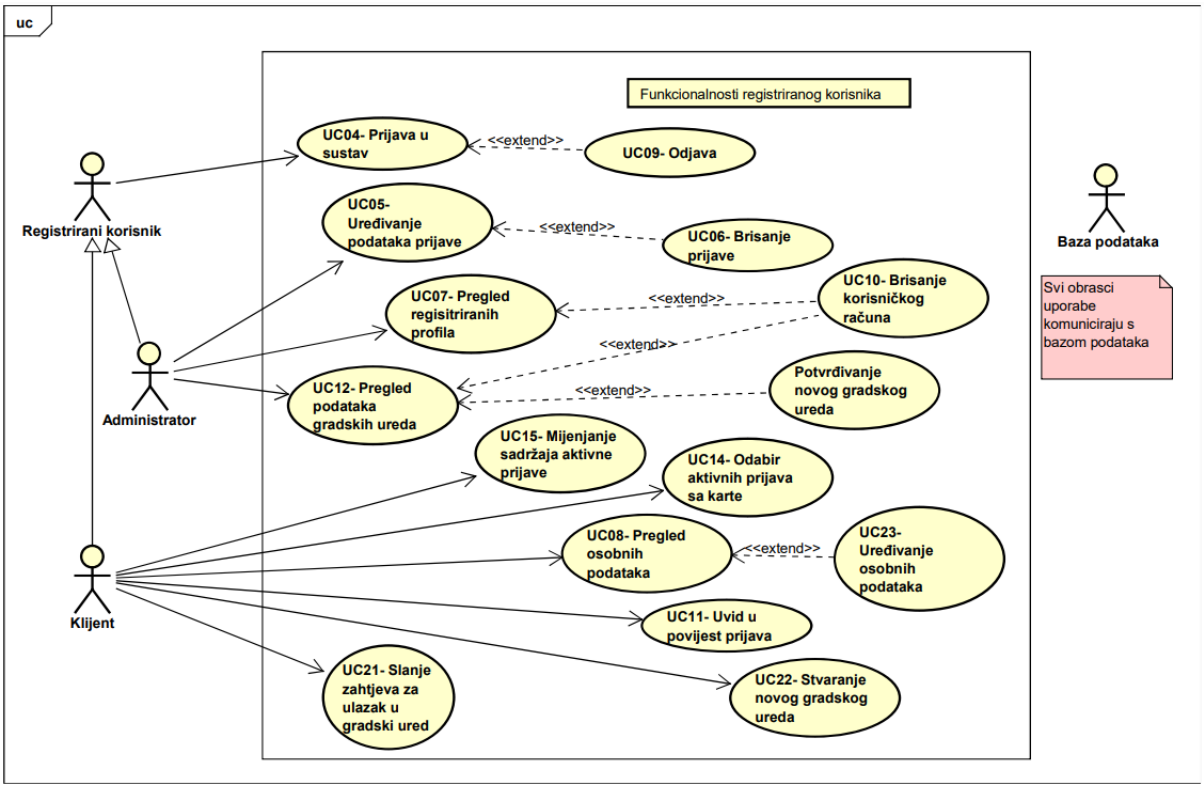
- **Glavni sudionik:** Gradski ured
- **Cilj:** Isporučiti objavu vezanu za odrađenu prihvaćenu prijavu na E-mail klijenta
- **Sudionici:** E-mail klijent
- **Preduvjet:** -
- **Opis osnovnog tijeka:**

1. Gradski ured šalje prouku o prijavi na E-mail klijent
2. E-mail klijent prikazuje podnositelju prijave obavijest

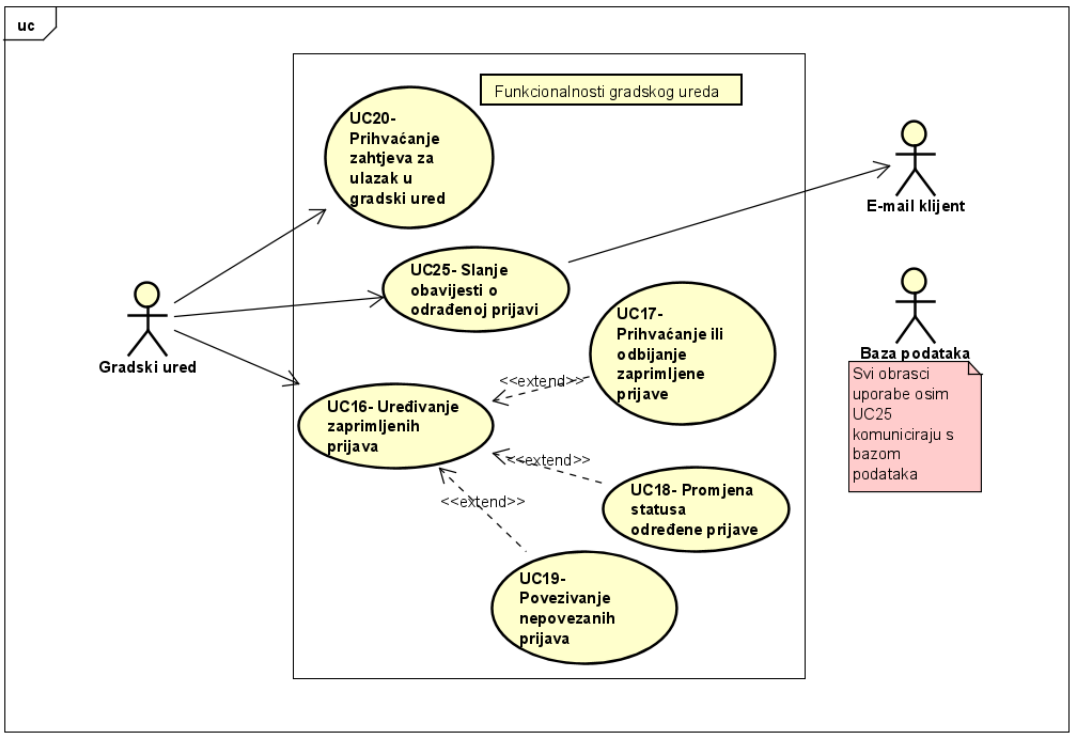
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe - Funkcionalnosti vezane za podnošenje prijave



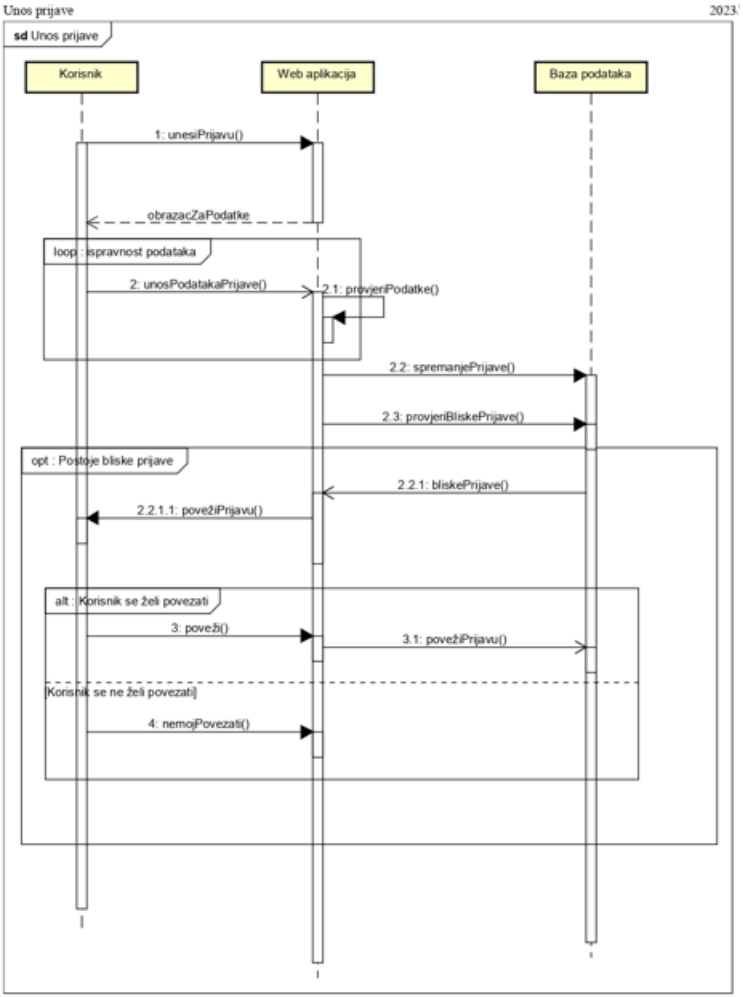
Slika 3.2: Dijagram obrasca uporabe - Funkcionalnosti registriranog korisnika (administrator i klijent)



Slika 3.3: Dijagram obrasca uporabe - Funkcionalnosti gradskog ureda

3.1.2 Sekvencijski dijagrami

Obrasci uporabe UC02, UC03, UC13, UC14 - unos nove prijave, pregled postojećih prijava, povezivanje na postojeću prijavu, uvid u aktivne prijave sa karte
Korisnik (prijavljeni ili anonimni) može podnijeti novu prijavu sustavu. Korisnik odabire opciju unosa nove prijave nakon čega poslužitelj vraća formu za upis podataka prijave. Korisnik šalje prijavu te sustav provjerava unos korisnika i provjerava u postoje li vremenski i prostorno bliske aktivne prijave te ako postoje, vraća formu korisniku gdje se može spojiti na neku od bliskih prijava. Korisnik šalje poslužitelju hoće li se i na koju povezati te poslužitelj sprema novu prijavu u bazu podataka. Korisnik također može poslati zahtjev za pregledom svih postojećih prijava uz opcionalne filtere po temi i lokaciji na što poslužitelj reagira dohvaćanjem tih podataka iz baze te slanjem istih korisniku.

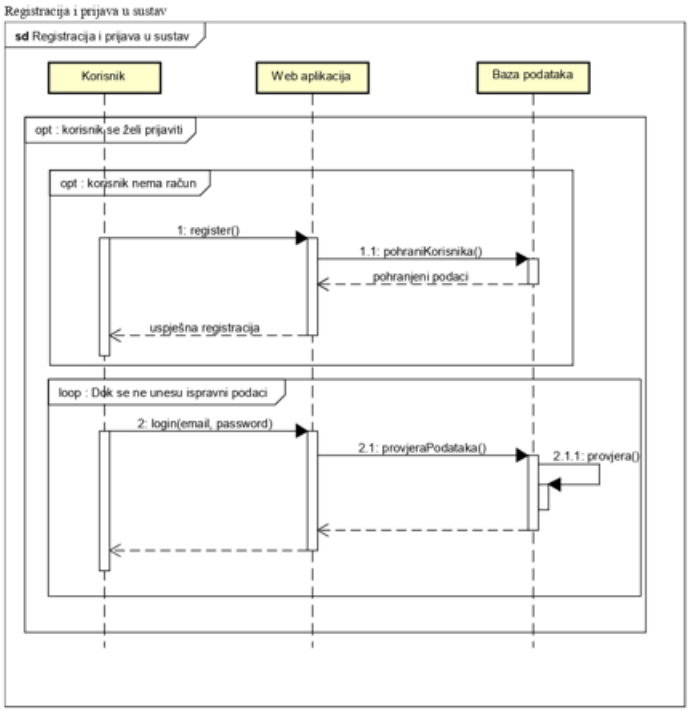


Slika 3.4: Sekvencijski dijagram - UC02, UC03, UC13, UC14

Obrasci uporabe UC01, UC04, UC08, UC11, UC15 - registracija i prijava korisnika u sustav

Neprijavljeni korisnik može poslati zahtjev za registracijom na što poslužitelj odgovara formom za registraciju. Kad korisnik pošalje podatke, poslužitelj ih validira te sprema u bazu podataka. Prijava korisnika u sustav, pregled osobnih podataka, pregled povijesti vlastitih prijava, mijenjanje sadržaja aktivne prijave.

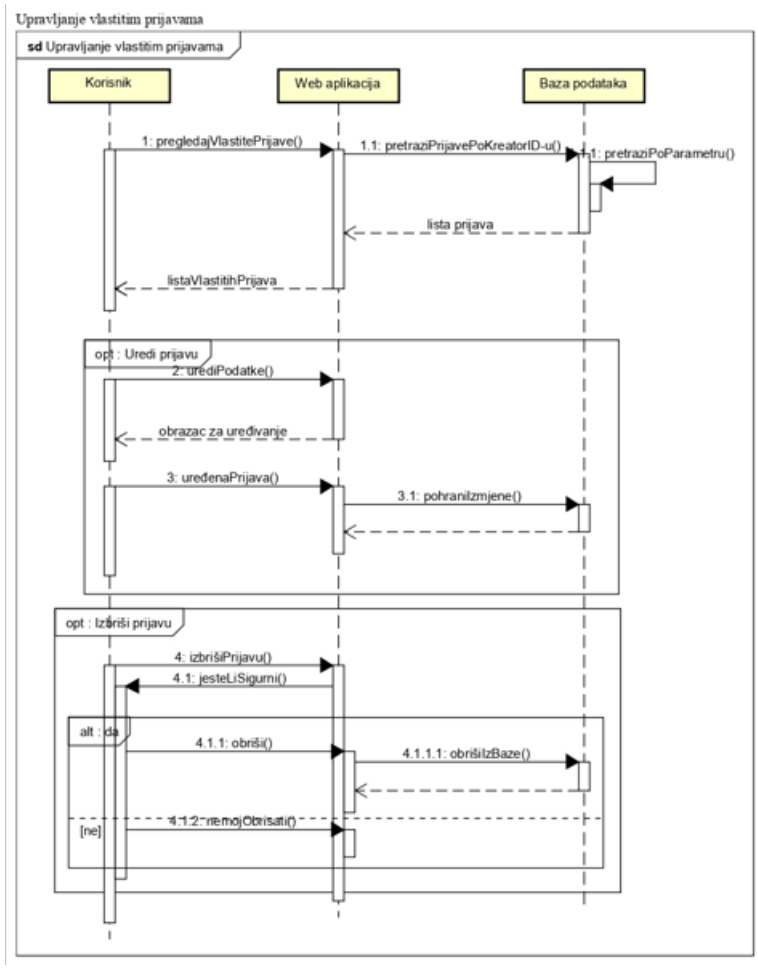
Registrirani korisnik može poslati zatjev za prijavom u sustav na što poslužitelj odgovara formom za prijavu. Kad korisnik prijavu pošalje, poslužitelj ju validira uz pomoć baze podataka te obaviještava korisnika o uspiješnosti prijave. Prijavljeni korisnik može poslati poslužitelju zahtjev za pregledom povijesti vlastitih prijava na što ih poslužitelj dohvaća iz baze podataka te ih vraća korisniku. Na isti način prijavljeni korisnik može pregledati osobne podatke.



Slika 3.5: Sekvencijski dijagram - UC01, UC04, UC08, UC11, UC15

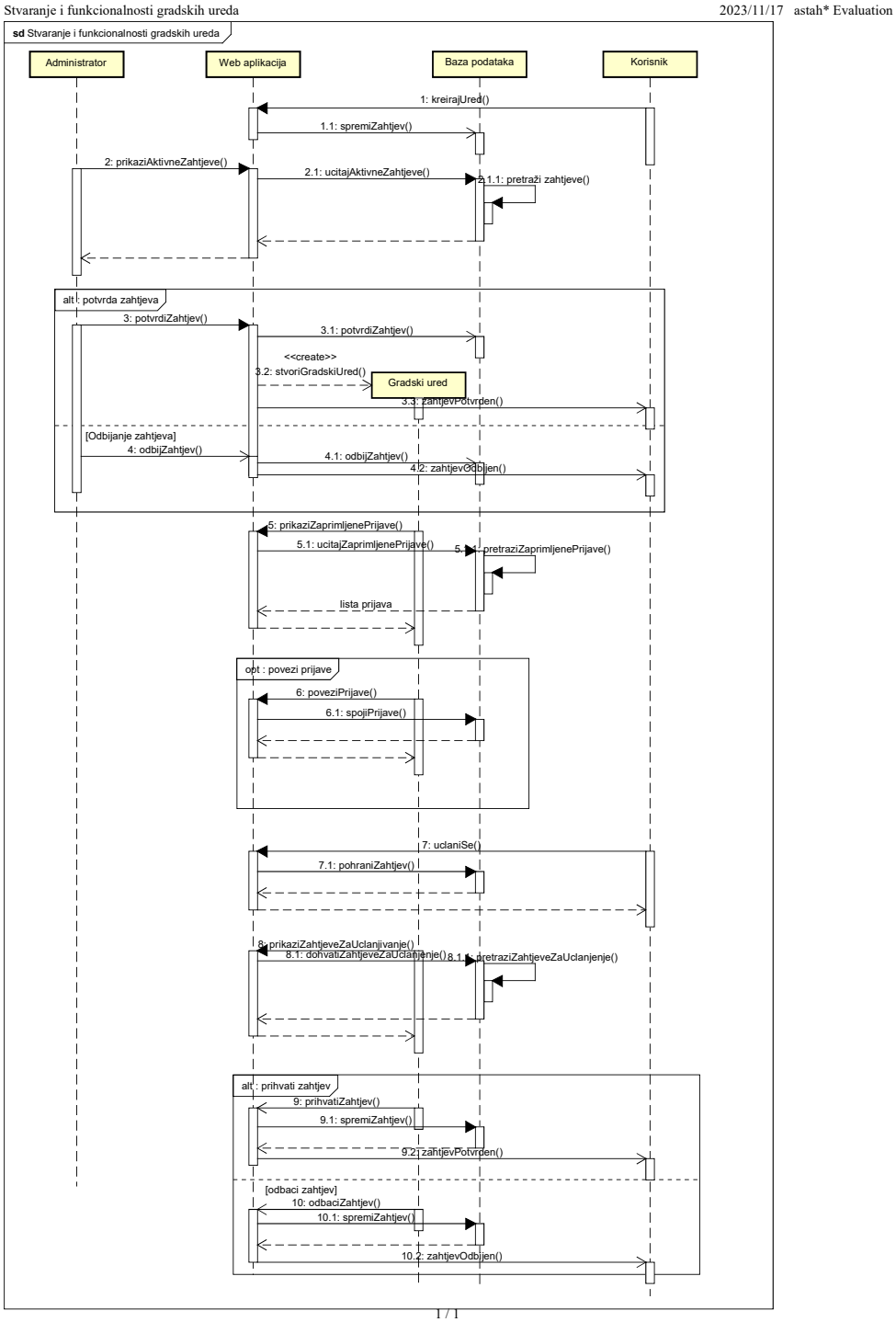
Obrazac uporabe UC05, UC06, UC11 - uređivanje podataka prijave, brisanje prijave, pregled korisnika

Korisnik u svakom trenutku može pregledati povijest svojih prijava. Prijave koje su još aktivne, tj. nisu odrađene može izbrisati u potpunosti iz sustava ili im mijenjati podatke.



Slika 3.6: Sekvencijski dijagram - UC05, UC06, UC11, UC15

Obrasci uporabe UC22, UC20, UC16, UC17, UC18, UC19 Prijavljeni korisnik može odlučiti kreirati novi gradski ured. Administrator to prvo mora odobriti da bi se transakcija uspješno dovila. Gradski ured može prihvatiti prijave koje je zaprimio, te listu istih uređivati po volji - mijenjati status ovisno o uspješnosti odrađene prijave, povezivati nepovezanih prijava. Također prijavljeni korisnici imaju mogućnost slanja zahtjeva za ulazak u gradski ured, što gradski ured može prihvatiti ili odbiti.



Slika 3.7: Sekvencijski dijagram - UC22, UC20, UC16, UC17, UC18, UC19

3.2 Ostali zahtjevi

- Sustav treba biti implementiran u obliku web aplikacije
- Aplikacija treba biti uvijek dostupna
- Aplikacije treba pružati usluge u stvarnom vremenu
- Učitavanje aplikacije ne smije trajati duže od 2 sekunde
- Pristup bazi podataka ne smije trajati duže od 2 sekunde
- Sustav treba biti organiziran u obliku MVC
- Sustav na poslužiteljskoj strani je napisan u programskom jeziku Java te radnom okviru Spring Boot
- Sustav na klijentskoj strani je implementiran programskim jezikom JavaScript te radnom okviru React.js
- Podaci se spremaju u bazu podataka koristeći JPA
- Arhitektura aplikacije mora biti u obliku klijent-poslužitelj
- Pri pristupu aplikaciji se koristi protokol HTTPS
- Aplikacije treba biti prilagođena i desktop uređajima kao i mobilnim uređajima
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogucen iz javne mreže pomoću HTTPS

4. Arhitektura i dizajn sustava

Arhitektura ovog projekta se može razlučiti na tri podsustava:

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

Web poslužitelj je podsustav kojemu je zadaća spremanje, obrađivanje i dostavljanje klijentima sadržaj web stranica. Preko web aplikacije poslužitelj komunicira sa klijentom koji je u ovom slučaju web preglednik. Komunikacija se odvija putem protokola aplikacijskog sloja interneta - HTTP (Hypertext Transfer Protocol). On je tu da ”reagira” na akcije koje mu web preglednik proslijedi te o ovisno o potrebi prolježuje zahtjev na web aplikaciju.

Web preglednik je program koji fizičkom korisniku omogućuje učitavanje i pregled web aplikacije. On je tu kao prevoditelj što znači da interpretira prikaz koda pisanog i uređivanog u HTML-u u izgled razumljiv korisnicima. Putem web preglednika korisnik šalje HTTP zahtjeve web poslužitelju, no isto tako web poslužitelj je tu da prilagodi prikaz HTTP odgovora i prikaže ih korisniku.

Baza podataka je podsustav u podatkovnom sloju kojem je primaran uloga sigurno spremanje podataka, a detaljnije o njoj se govori u poglavlju 4.1.

Web aplikacija je dio sustava kojeg korisnik koristi za obrađivanje željenih zahtijeva koji uzrokuju pristupanje podacima u bazi podataka. Web aplikacija za interpretiranje dostavlja web pregledniku određeni HTML kod, a podijeljena je na back-end i na front-end o kojima je riječ u nastavku.

- **Front-end** ili klijentska strana je dio aplikacije koji služi za sve ono vidljivo na web pregledniku. On je prezentacijski sloj koji korisniku omogućuje jednostavnu komunikaciju sa sustavom. Tehnologija koja je korištena za realizaciju ovog dijela web aplikacije je programski jezik JavaScript, preciznije radni okvir React u kombinaciji sa TypeScriptom.

- **Back-end** ili poslužiteljska stran je dio aplikacije koji je zadužan za obrađivanje dobivenih zahtjeva i vršenja funkcionalnih akcija. Tehnologija u kojom je pisan je programski jezik Java, preciznije u radnom okviru SpringBoot.

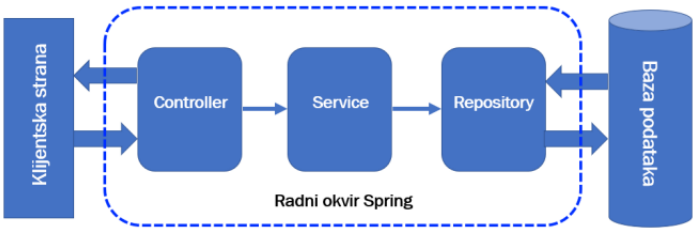
Okruženje u kojem je pisan front-end je Visual Studio Code, a backe-end u Intellij-u. To nisu fiksna okruženja i može se koristiti bilo koji drugi IDE ili text editor. Arhitektura sustava se temelji na višeslojnom arhitekturnom stilu koji je podržan od strane SpringBoot tehnologije.

Višeslojni stil arhitekture

Višeslojna arhitektura sastoji se od idućih slojeva:

- klijentske strane koja omogućuje prikaz korisničkog sučelja
- nadglednika koji korisničku i poslužiteljsku stranu
- usluge koja obavlja poslovnu logiku i ostvaruje temeljnu funkcionalnost i zadaaću web aplikacije
- repozitorija koji definira način pristupanja podacima
- baze podataka koja u našem slučaju podatke sprema u relacijsku bazu Postgre

Osnovna značajka ovakvog stila arhitekture su da svaki sloj pruža uslugu drugom sloju, a skriva svoj skup usluga. Jedna od prednosti ovog stila je olakšavanje ostvarivanja podjele brige u web aplikaciji jer se svaki sloj brine isključivo o svojoj funkcionalnosti. Klijentska strana razgovara sa web sučeljem, nadglednik putem REST API-ja pruža vanjsko sučelje i prihvaća HTTP zahtjeve te poziva odgovarajuće usluge, a potom te iste vraća kao odgovor klijentu. Sloj usluge definira pristup uslugama web aplikacije(tko i kako pristupa). Sloj repozitorija nam osigurava pristup podacima, te preslikavanje domenskih objekata u bazu podataka koristeći formu 1:1. Skica ovakve arhitekure je dana na slici 4.1



Slika 6.5. Povezanost slojeva u radnom okviru Spring.

Slika 4.1: Primjer višeslojne arhitekture korištenjem radnog okvira Spring

4.1 Baza podataka

dio 1. revizije

Za potrebe sustava koristit će se relacijska baza podataka koja nam omogućuje jednostavnije modeliranje stvarnog svijeta. Baza je implementirana u PostgreSQL-u zbog njegove jednostavnosti i jer je tim najbolje upoznat s tim sustavom; njegovim limitacijama i pravilima. Entiteti stvarnog svijeta su prevedeni kao tablice (relacije) koje imaju ime i svoj skup atributa. Baza podataka osigurava nam jednostavnu pohranu, umetanje, izmjenu i dohvat podataka, te garantira njihovu sigurnost. Baza podataka koristi sljedeće atribute:

- prijave
- lokacije
- slike
- sorisnici
- tipovi_ostecenja
- gradski_uredi

4.1.1 Opis tablica

prijave je entitet koji sadrži sve važne informacije o podnesenim prijavama. Sastoji se od atributa:id, ostecenje_id, lokacija_id, kreator_id, parent_prijava_id, prvo_vrijeme_prijave, vrijeme_otklona, naziv, opis. Ovaj entitet u vezi je One-to-One s tablicom lokacije preko atributa lokacija_id, sa entitetom slike je u vezi One-to-Many preko atributa id. Sa entitetom tipovi_ostecenja je u vezi One-to-Many preko atributa ostecenje_id, u vezi Many-to-One sa entitetom korisnici preko atributa kreator_id, te je sama sa

sobom u refleksivnoj vezi One-to-Many preko atributa id.

prijave		
id	INT	jedinstveni identifikator prijave
ostecenje_id	INT	jedinstveni identifikator tipa ostecenja
kreator_id	INT	jedinstvani identifikator korisnika koji je poslao prijavu
parent_prijava_id	INT	jedinstveni identifikator prijave na koju se prijava nadovezala
prvo_vrijeme_prijave	INT	vrijeme slanja prijave
vrijeme_otklona	INT	vrijeme otklona prijave
opis	VARCHAR	opis poslane prijave
naziv	INT	konkretno ime poslane prijave

lokacije je entitet koji sadrži osnovne podatke o geografskoj lokaciji pojedine prijave. Sastoji se od atributa: lokacija_id, latitude, longitude. Preko atributa id je povezana vezom One-to-One sa relacijom Prijave.

Lokacije		
lokacija_id	INT	jedinstveni identifikator prijave
latitude	INT	geografska latituda lokacije
longitude	INT	geografska longituda lokacije

korisnici je entitet koji sadrži osobne podatke o registriranim korisnicima kao i njihovu pripadnost vijecu i ulogu koju obnašaju u sustavu. Sastoji se od atributa: id, ime, prezime, username, email, password, ured_id, role, active. Povezani su sa relacijom prijave u vezi One-to-Many preko atributa id, te sa relacijom gradski_uredi u vezi Many-to-One preko atributa ured_id.

korisnici		
id	INT	jedinstveni identifikator korisnika
ime	VARCHAR	ime registriranog korisnika
prezime	VARCHAR	prezime registriranog korisnika
username	VARCHAR	korisničko ime registriranog korisnika
email	VARCHAR	e-mail adresa registriranog korisnika
password	VARCHAR	lozinka za prijavu registriranog korisnika
ured_id	INT	jedinstvani identifikator vijeca/ureda kojem pripada
active	INT	jedinstveni token sesije prijave korisnika
role	VARCHAR	uloga koju korisnik obnaša u sustavu

slike je entitet koji sadrži podatke vezane za sliku podnešene prijave. Sastoji se od atributa: id, podatak, prijava_id. Preko entiteta prijava_id je povezan tablicom prijave u vezi Many-to-One.

slike		
id	INT	jedinstveni identifikator slike pojedine prijave
podatak	VARCHAR	string bitova koji sadrže sliku
prijava_id	INT	jedinstveni identifikator prijave za koju je poslana slika

tipovi_osetecenja je entitet koji sadrži podatke vezane za opis oštećenja kao i id ureda koji se bavi tim tipom oštećenja. Sastoji se od atributa: id, naziv. Preko atributa id je povezan sa relacijom prijave u vezi Many-to-One, te je sa relacijom gradski_uredi u vezi One-to-One povezan preko istog atributa id.

tipovi_ ostecenja		
id	INT	jedinstveni identifikator vrste oštećenja

Nastavljeno na idućoj stranici

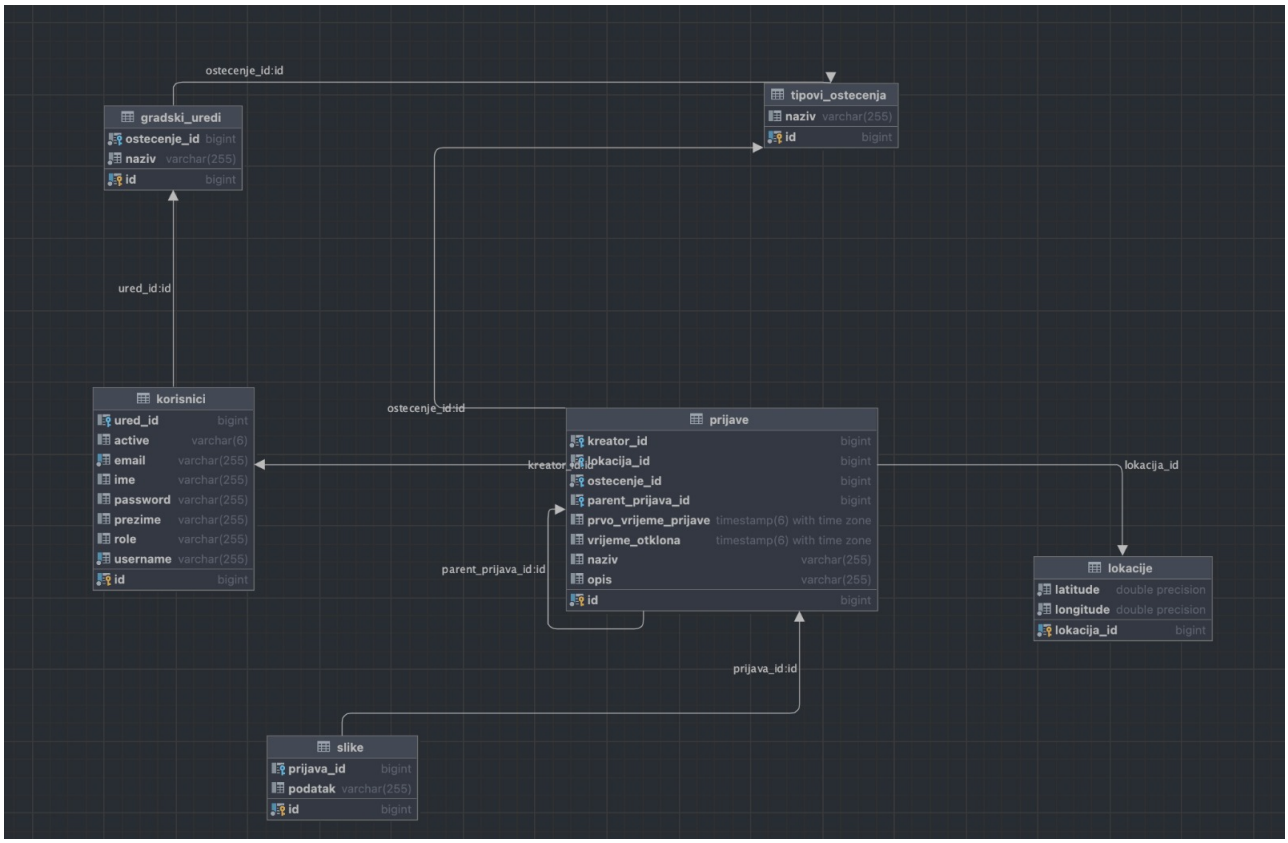
Nastavljeno od prethodne stranice

tipovi_ ostecenja		
naziv	VARCHAR	naziv konkretnog oštećenja

gradski_uredi je entitet koji sadrži podatke vezane za opis pojedinog vijeca/ureda koji je zadužen za otklanjanje određenog tipa oštećenja. Sastoji se od atributa: id, naziv i ostecenje_id. Preko atributa ostecenje_id je povezan sa relacijom tipovi_ostecenja u vezi Many-to-One, te je povezan sa relacijom korisnici preko atributa id u vezi One-to-Many.

gradski_uredi		
id	INT	jedinstveni identifikator pojedinog registriranog vijeća/ureda
naziv	VARCHAR	puni naziv vijćea u aplikaciji
ostecenje_id	INT	identifikator vrste oštećenja koje za koje je taj ured zadužen

4.1.2 Dijagram baze podataka

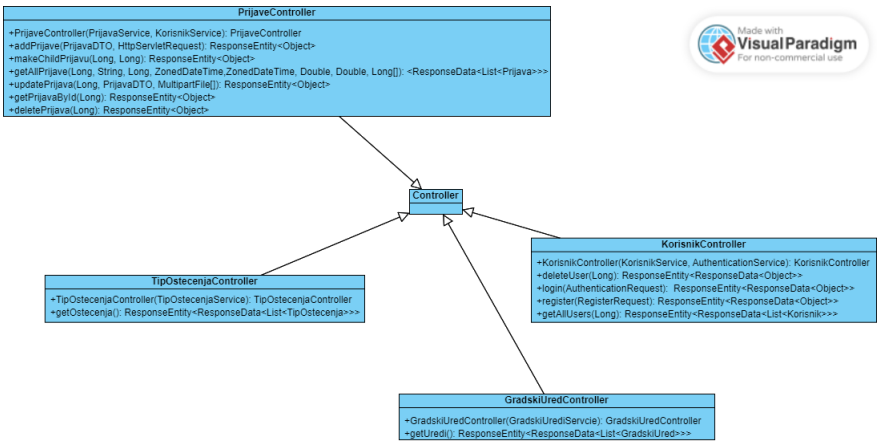


Slika 4.2: Relacijski dijagram baze podataka

4.2 Dijagram razreda

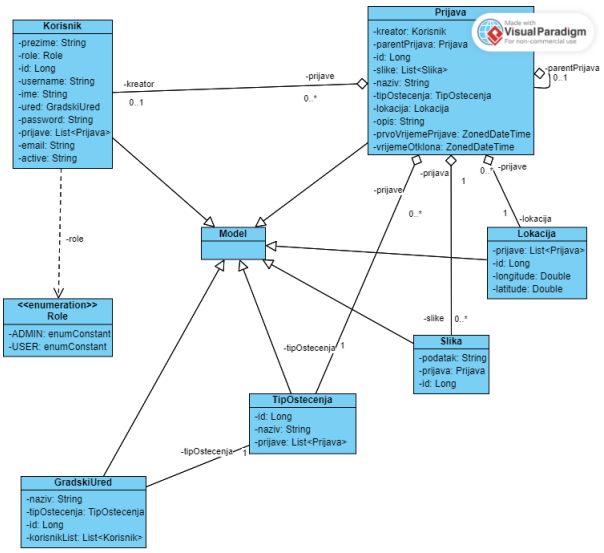
Na slikama 4.3 i 4.4 prikazani su razredi i njihove relacije na serverskoj strani aplikacije. Konkretnije rečeno prikazani su modeli i kontroleri.

Svi kontroleri nasljeđuju razred Controller. Metode koje su u njima implementirane ovise o DTO objektima, koji se realiziraju pomoću metoda implementiranih u Domain dijelu razreda.



Slika 4.3: Dijagram razreda - Controllers dio

Domain dio razreda preslikava strukturu baze podataka, te svi razredi nasljeđuju razred Model. Ovako ilustrirani razredi su zapravo odraz onih relacija koji su zapisani u bazi podataka, te direktno komunciraju sa bazom podataka. Razred Korisnik je ilustracija bilo kojeg registriranog usera koji ima svoju ulogu (povezan sa enumom Role), te može podnijeti prijavu u sustav (pvezanost sa razredom Prijava preko private atributa prijave). Prijave kada se konstruiraju zadaje im se lokacija (pa su preko atributa lokacija povezane sa razredom Lokacija), slika (agregatna povezanost sa razredom Slika preko varijable slike) i tip ostecenja (agregatna povezanost sa razredom tipOstecenja preko atributa tipOstecenja). Razred GradskiUred je reprezentacija gradskig ureda koje je povezan sa tipom oštećenja preko atributa tipOstecenja.



Slika 4.4: Dijagram razreda - Controllers dio

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. The Unified Modeling Language, https://www.uml-diagrams.org/
6. Astah Community, http://astah.net/editions/uml-new

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: 23. listopada 2023.
    - Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Ivan Šimunić, Dominik Zoričić
    - Teme sastanka:
        - raspodjela uloga po članovima
        - odabir izvedbenih tehnologija
2. sastanak
    - Datum: 26. listopada 2023.
    - Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Dominik Zoričić
    - Teme sastanka:
        - definiranje osnovnih zahtjeva
3. sastanak
    - Datum: 30. listopada 2023.
    - Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Dominik Zoričić, Ivan Šimunić
    - Teme sastanka:
        - Kreacija baze podataka
4. sastanak
    - Datum: 2. studenoga 2023.
    - Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Dominik Zoričić, Ivan Šimunić
    - Teme sastanka:

  - Demo verzija front-enda
  - Ostali zahtjevi sustava

5. sastanak
   - Datum: 11. studenoga 2023.
   - Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Dominik Zoričić, Ivan Šimunić
   - Teme sastanka:
     - Komentiranje dijagrama obrazaca uporabe
     - Uspoređivanje obrazaca uporabe sa funkcijama back-end servisa
6. sastanak
   - Datum: 16. studenoga 2023.
   - Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Dominik Zoričić, Ivan Šimunić
   - Teme sastanka:
     - Autentifikacija korisnika
     - Plan za deployment

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Ivan Šimunić	**Nino Ćurko**	**Davor Najev**	**Nikola Botić**	**Dominik Zoričić**	**Josip Šare**	**Ivan Elez**
Upravljanje projektom	10	10	11	12	9	10	8
Opis projektnog zadatka		2					
Funkcionalni zahtjevi		2.5					
Opis pojedinih obrazaca		4					
Dijagram obrazaca		2					
Sekvencijski dijagrami							
Opis ostalih zahtjeva		1					
Arhitektura i dizajn sustava		2					
Baza podataka		1.5					
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati		1					
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ivan Šimunić	**Nino Ćurko**	**Davor Najev**	**Nikola Botić**	**Dominik Zoričić**	**Josip Šare**	**Ivan Elez**
Dnevnik sastajanja	1	1					
Zaključak i budući rad							
Popis literature							
Postavljanje klijentske strane (front-end)			20	15			
Postavljanje serverske strane (front-end)					15	13	11
izrada baze podataka	1	1	1	1	2	1	1
spajanje s bazom podataka			4		4	4	
Autorizacija i autentikacija			3		5	5	
Punjenje baze podataka					1	1	2
Implementacija Google maps API-ja			4			1	
Deployment			6		6	6	

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.