

# Programsko inženjerstvo

Ak. god. 2023./2024.

<Naziv projekta>

Dokumentacija, Rev. <1 ili 2>

Grupa: *SourceresOfTheNorth*

Voditelj: *Petar Belošević*

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

Nastavnik: *Vlado Sruk*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
2.1	Primjeri u $\text{\LaTeX}$ u . . . . .	10
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>13</b>
3.1	Funkcionalni zahtjevi . . . . .	13
3.1.1	Obrasci uporabe . . . . .	15
3.1.2	Sekvencijski dijagrami . . . . .	16
3.2	Ostali zahtjevi . . . . .	17
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>18</b>
4.1	Baza podataka . . . . .	18
4.1.1	Opis tablica . . . . .	18
4.1.2	Dijagram baze podataka . . . . .	19
4.2	Dijagram razreda . . . . .	20
4.3	Dijagram stanja . . . . .	21
4.4	Dijagram aktivnosti . . . . .	22
4.5	Dijagram komponenti . . . . .	23
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>24</b>
5.1	Korištene tehnologije i alati . . . . .	24
5.2	Ispitivanje programskog rješenja . . . . .	25
5.2.1	Ispitivanje komponenti . . . . .	25
5.2.2	Ispitivanje sustava . . . . .	25
5.3	Dijagram razmještaja . . . . .	26
5.4	Upute za puštanje u pogon . . . . .	27
<b>6</b>	<b>Zaključak i budući rad</b>	<b>28</b>
	<b>Popis literature</b>	<b>29</b>

**Indeks slika i dijagrama** **30**

**Dodatak: Prikaz aktivnosti grupe** **31**

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak. Započet opis zadatka.	Petar Belošević	23.10.2023.
0.2	Dovršen opis projektnog zadatka.	Petar Belošević	28.10.2023.
0.3	Dodani dionici i aktori.	Petar Belošević	29.10.2023.
0.6	Arhitektura i dizajn sustava, algoritmi i strukture podataka	*	26.08.2013.
0.8	Povijest rada i trenutni status implementacije, Zaključci i plan daljnjeg rada	*	28.08.2013.
0.9	Opisi obrazaca uporabe	*	07.09.2013.
0.10	Preveden uvod	*	08.09.2013.
0.11	Sekvencijski dijagrami	*	09.09.2013.
0.12.1	Započeo dijagrame razreda	*	10.09.2013.
0.12.2	Nastavak dijagrama razreda	*	11.09.2013.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene: Timer - Brojilo vremena	*	15.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

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

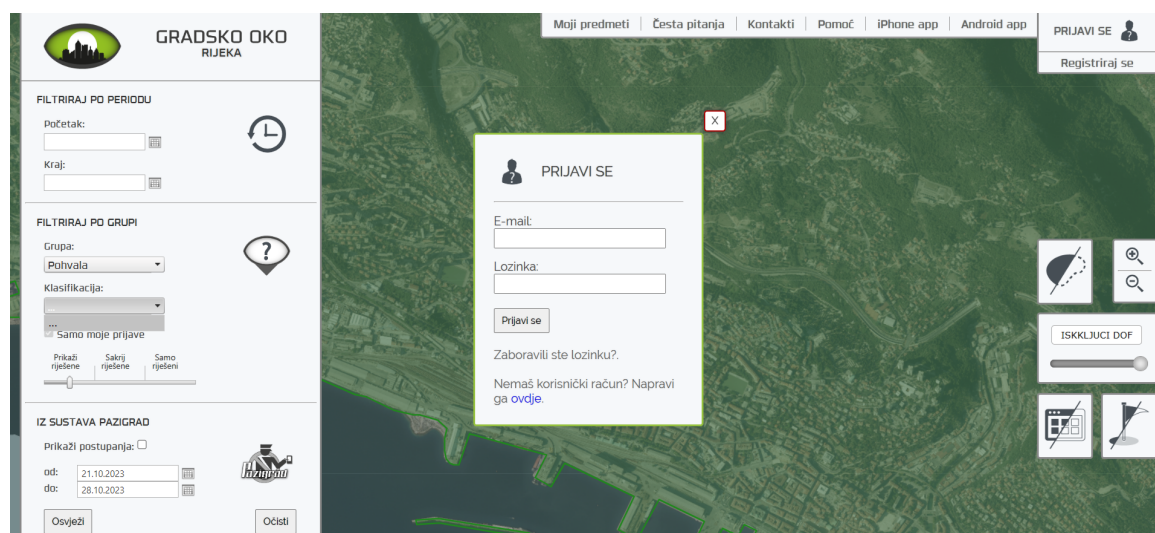
*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih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

## 2. Opis projektnog zadatka

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

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

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

Slika 2.1: Aplikacija *Gradsko oko* Grada Rijeke

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

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

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

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

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

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

## 2.1 Primjeri u L<sup>A</sup>T<sub>E</sub>Xu

*Ovo potpoglavlje izbrisati.*

U nastavku se nalaze različiti primjeri kako koristiti osnovne funkcionalnosti L<sup>A</sup>T<sub>E</sub>Xa koje su potrebne za izradu dokumentacije. Za dodatnu pomoć obratiti se asistentu na projektu ili potražiti upute na sljedećim web sjedištima:

- Upute za izradu diplomskog rada u L<sup>A</sup>T<sub>E</sub>Xu - [https://www.fer.unizg.hr/\\_download/repository/LaTeX-upute.pdf](https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf)
- L<sup>A</sup>T<sub>E</sub>X projekt - <https://www.latex-project.org/help/>
- StackExchange za Tex - <https://tex.stackexchange.com/>

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

primjer primjer primjer primjer primjer primjer

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

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

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

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

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

naslov unutar tablice		
ime	VARCHAR	
primjer	VARCHAR	

Tablica 2.1: Naslov s referencom izvan tablice

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



Slika 2.2: Primjer slike s potpisom



Slika 2.3: Primjer slike s potpisom 2

Referenciranje slike 2.3 u tekstu.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### *dio 1. revizije*

##### Dionici:

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

##### Aktori i njihovi funkcionalni zahtjevi:

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

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

3. Djelatnik gradskog ureda (inicijator) može:

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

4. Baza podataka (sudionik) može:

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

### 3.1.1 Obrasci uporabe

#### *dio 1. revizije*

#### **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 tijekom obrasca vratio na osnovni tijek.*

#### UC<broj obrasca> -<ime obrasca>

- **Glavni sudionik:** <sudionik>
- **Cilj:** <cilj>
- **Sudionici:** <sudionici>
- **Preduvjet:** <preduvjet>
- **Opis osnovnog tijeka:**
  1. <opis korak jedan>
  2. <opis korak dva>
  3. <opis korak tri>
  4. <opis korak četiri>
  5. <opis korak pet>
- **Opis mogućih odstupanja:**
  - 2.a <opis mogućeg scenarija odstupanja u koraku 2>
    1. <opis rješenja mogućeg scenarija korak 1>
    2. <opis rješenja mogućeg scenarija korak 2>
  - 2.b <opis mogućeg scenarija odstupanja u koraku 2>
  - 3.a <opis mogućeg scenarija odstupanja u koraku 3>

#### **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.*



### 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.*

## 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.

## 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-programске zahtjeve. Po točkama razraditi i popratiti odgovarajućim skicama:

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

### 4.1 Baza podataka

#### dio 1. revizije

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

#### 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

korisnik - ime tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

korisnik - ime tablice		
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	

#### 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".*

## 4.2 Dijagram razreda

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

### **dio 1. revizije**

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

### **dio 2. revizije**

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

## 4.3 Dijagram stanja

### *dio 2. revizije*

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

## 4.4 Dijagram aktivnosti

### *dio 2. revizije*

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

## 4.5 Dijagram komponenti

### *dio 2. revizije*

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



## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### *dio 2. revizije*

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

## 5.2 Ispitivanje programskog rješenja

### dio 2. revizije

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

### 5.2.1 Ispitivanje komponenti

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

### 5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>1</sup>. 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.*

---

<sup>1</sup><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

2.1	Aplikacija <i>Gradsko oko</i> Grada Rijeke . . . . .	6
2.2	Primjer slike s potpisom . . . . .	11
2.3	Primjer slike s potpisom 2 . . . . .	12

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

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

#### 1. sastanak

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

#### 2. sastanak

- Datum: u ovom formatu: 29. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme



## Tablica aktivnosti

### Kontinuirano osvježavanje

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

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

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

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

## 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](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*