

# Programsko inženjerstvo

Ak. god. 2020./2021.

<Naziv projekta>

Dokumentacija, Rev. <1 ili 2>

Grupa: <Naziv grupe>

Voditelj: <Ime i prezime voditelja>

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

Nastavnik: <Ime i prezime nastavnika zaduženog za vašu grupu>

# Sadržaj

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	*	22.08.2013.
0.2	Dopisane upute za povijest dokumentacije. Dodane reference.	*	24.08.2013.
0.5	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	*	25.08.2013.
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.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.2	Manje izmjene:Timer - Brojilo vremena	*	15.09.2013.
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

### *dio 1. revizije*

*Na osnovi projektnog zadatka detaljno opisati korisničke zahtjeve. Što jasnije opisati cilj projektnog zadatka, razraditi problematiku zadatka, dodati nove aspekte problema i potencijalnih rješenja. Očekuje se minimalno 3, a poželjno 4-5 stranica opisa. Teme koje treba dodatno razraditi u ovom poglavlju su:*

Cilj je razviti programsku potporu za web aplikaciju "MapTasker" koja služi za pronalaženje nestalih osoba tijekom raznim elementarnih nepogoda. Kako bi se takve operacije uspješno izvele, potrebno je spasiocima napraviti kartu s pozicijama kuća koju proizvode kartografi. Područje koje je potrebno kartografirati postavljaju vođaitelji.

Početna stranica sadrži formu za registraciju ili prijavu te tipku za prijavu nestalih osoba. Taj gumb vodi korisnika na stranicu na kojoj se nalaze već prijavljene nestale osobe te gumb za prijavu nove osobe. Prijavljenim korisnicima se na stranici također nalazi karta čije se funkcionalnosti razlikuju, oviseći o ulozi korisnika. Uz kartu se nalazi statistika koja vođaiteljima i kartografima prikazuje podatke o nestalim i pronađenim osobama, broj završenih blokova kroz vrijeme (*engl. Burn down chart*) te broj pretraženih i nepretraženih građevina.

Prijavljeni korisnici imaju pristup stranici koja im omogućuje pregled i uređivanje njihovog profila. Pri pokretanju aplikacije neregistrirani korisnik ima dvije opcije:

1. Ostati neregistriran te prijaviti nestalu osobu te njihove informacije:
  - Ime i prezime
  - Fotograija
  - Opis nestale osobe
2. Poslati zahtjev za registraciju na aplikaciju pri čemu je potrebno navesti:
  - Uloga za koju se prijavljuje (spasioc, vođaitelj, kartograf)
  - Korisničko ime

- Fotografija
- Lozinka
- Ime i prezime
- Broj mobitela
- Email adresa

U prvom slučaju gdje korisnik prijavljuje nestalu osobu, ostali korisnici mogu komentirati prijavu i odgovoriti na nju. Kad je prijavljena osoba pronađena, spasioc ili voditelj zaključavaju prijavu dok voditelj jedini ima ulogu brisanja prijave i komentara.

Voditelj je odgovoran za proces stvaranja operacije te definiranja područja koje će se kartografirati i pretraživati. To uključuje određivanje područja na karti koje će biti definirano poligonom. Kako bi se izbjegli problemi, samo jedan voditelj može raditi na jednoj regiji i blokovima. Postoje tri razine područja:

- Regija
- Blok
- Građevina

Na jednom poligonu može postojati više regija u kojima se nalaze razni blokovi. Svaki od blokova ima status o tome u kojoj se fazi trenutno nalazi:

1. **Nezapočeto**
2. **Aktivno**
3. **Provjera**
4. **Završeno**

Blok će biti prikazan na karti u različitim bojama ovisno o fazi. Nakon što je blok **završen**, voditelj zaključava operaciju.

Kartograf jedini može mijenjati statuse blokova iz **nezapočetog** u **aktivnog** i nakon toga u **provjeru**. U fazu **završeno** blok prelazi jedino kad su dva kartografa sigurna da je posao dobro obavljen. Ako kartograf smatra da se posao mogao bolje odraditi, može vratiti stanje bloka iz **provjere** u **aktivno**. U **aktivnom** stanju kartograf može dodavati nove građevine. Kartograf može tijekom kartografiranja

imati samo jedan **aktivan** blok kako bi se izbjegli konflikti. Blok može biti **aktivan** samo kod jednog kartografa.

Spasioc na svojoj stranici ima pristup karti s građevinama koje imaju jedan od dva statusa:

- Pretraženo
- Nepretraženo

Nakon promjene statusa iz **nepretraženo** u **pretraženo**, kartograf i voditelj primaju obavijest na stranici da je osoba za koju je objavljena prijava pronađena te voditelj dobiva ime i prezime spasioca. U tom slučaju, voditelj zatvara prijavu za pronađenu osobu. Time se ažurira statistika *Burn down chart*, broj nestalih osoba i pretraženih građevina.

Administrator aplikacije ima pristup popisu svih registriranih korisnika te njihovim osobnim podacima. Uloga mu je potvrđivanje ili odbacivanje zahtjeva za registraciju korisnika te može mijenjati dodijeljena prava korisnicima.

Sustav treba podržavati rad te više zahtjeva od više korisnika u stvarnom vremenu.

## 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	

Nastavljeno na idućoj stranici




Nastavljeno od prethodne stranice

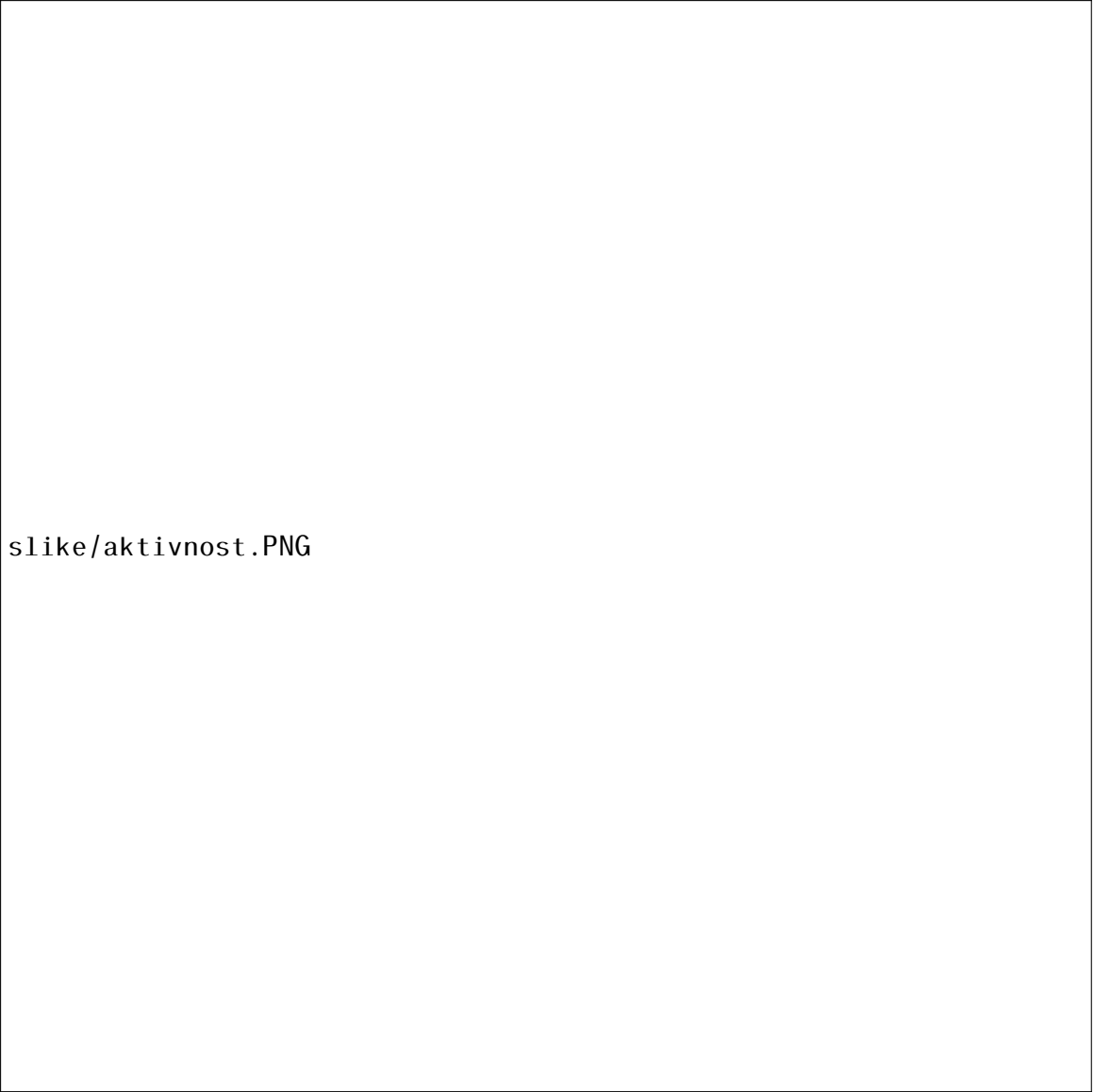
naslov unutar tablice		
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	

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	

slike/aktivnost.PNG

Slika 2.1: Primjer slike s potpisom



slike/aktivnost.PNG

Slika 2.2: Primjer slike s potpisom 2

Referenciranje slike ?? u tekstu.

## 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.

Navesti **aktore** koji izravno **koriste ili komuniciraju sa sustavom**. Oni mogu imati inicijatorsku ulogu, tj. započinju određene procese u sustavu ili samo sudioničku ulogu, tj. obavljaju određeni posao. Za svakog aktora navesti funkcionalne zahtjeve koji se na njega odnose.

#### **Dionici:**

1. Neregistrirani korisnik
2. Voditelj operacije
3. Kartograf
4. Spasioc
5. Administrator

#### **Aktori i njihovi funkcionalni zahtjevi:**

1. Neregistrirani/neprijavljeni korisnik može:
  - (a) poslati zahtjev za registraciju s potrebnim osobnim podacima: korisničko ime, fotografija, ime, prezime, lozinka, broj mobitela i adresa, i s ulogom za koju se prijavljuje
  - (b) prijaviti nestalu osobu s imenom, prezimenom, fotografijom i opisom
2. Voditelj operacije može:
  - (a) zaključati prijavu o nestaloj osobi nakon što osoba bude pronađena
  - (b) obrisati prijavu o nestaloj osobi i komentare na prijavi

- (c) definirati područja na karti koja je potrebno pretražiti i kartografirati
- (d) imati pristup statistici humanitarne akcije
- (e) mijenjati podatke u svom profilu

3. Kartograf može:

- (a) promijeniti status odabranog bloka na karti iz nezapočetog u aktivno, i iz aktivnog u provjeru
- (b) promijeniti status odabranog bloka na karti iz provjera u aktivno ako posao nije dobro odrađen, te nastavljati uređivati područje
- (c) dodavati nove građevine bloku sa statusom aktivno
- (d) imati pristup statistici humanitarne akcije
- (e) mijenjati podatke u svom profilu

4. Spasioc može:

- (a) pregledavati kartu sa unesenim građevinama i promijeniti im status u pretraženo ili nepretraženo
- (b) zaključati prijavu o nestaloj osobi nakon što osoba bude pronađena
- (c) mijenjati podatke u svom profilu

5. Administrator može:

- (a) potvrditi željena prava neregistriranog korisnika
- (b) vidjeti popis registriranih korisnika i njihovih osobnih podataka
- (c) mijenjati dodijeljena prava registriranih korisnika
- (d) imati sva prava i ovlasti kao registrirani korisnici

### 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.*

#### UC1 - Registracija u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Registracija u sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju u sustav
  2. Korisnik upisuje tražene osobne podatke i ulogu za koju se prijavljuje
  3. Korisnik prima obavijest o uspješno obavljenoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Korisnik odabire već postojeće korisničko ime i/ili e-mail, ili unosi podatke u nedozvoljenom (neispravnom) formatu
    1. Sustav onemogućuje registraciju, te vraća korisnika na stranicu za registraciju
    2. Korisnik ispravlja pogrešno upisane podatke te registracija završava uspješno, ili odustaje od registracije

#### UC2 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Stjecanje odgovarajućih ovlasti u aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** Uspješna registracija
- **Opis osnovnog tijeka:**
  1. Korisnik unosi korisničko ime i lozinku
  2. Potvrda o ispravnosti unesenih podataka

3. Dobivanje odgovarajućih ovlasti u aplikaciji
- **Opis mogućih odstupanja:**
  - 2.a Neispravnost unesenog korisničkog imena i/ili lozinke
    1. Sustav obavještava korisnika o neuspjeloj prijavi u sustav i vraća ga na homepage

### UC3 - Pregled nestalih osoba

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled podataka nestalih osoba
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire ekran "Nestali"
  2. Aplikacija daje prikaz podataka nestalih osoba

### UC4 - Prijava nestale osobe

- **Glavni sudionik:** Korisnik
- **Cilj:** Prijaviti nestanak osobe
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik unosi podatke o nestaloj osobi u sustav
- **Opis mogućih odstupanja:**
  - 1.a Prijava već prijavljene nestale osobe
    1. Sustav obavještava korisnika da je osoba već prijavljena kao nestala i vraća ga na stranicu "Nestali"

### UC5 - Komentiranje prijave nestale osobe

- **Glavni sudionik:** Korisnik
- **Cilj:** Komentiranje prijave nestale osobe
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik piše komentar na prijavu o nestaloj osobi

### UC6 - Zaključavanje prijave

- **Glavni sudionik:** Spasioc ili voditelj operacije
- **Cilj:** Zaključavanje prijave o nestaloj osobi
- **Sudionici:** Baza podataka
- **Preduvjet:** Osoba je pronađena
- **Opis osnovnog tijeka:**
  1. Spasioc ili voditelj operacije odabire ekran "Nestali"
  2. Spasioc ili voditelj operacije zaključava prijavu o nestaloj osobi
  3. Status prijave mijenja se u "zaključana".
  4. Prijava se više ne prikazuje u sustavu

#### UC7 - Brisanje prijave

- **Glavni sudionik:** Voditelj operacije
- **Cilj:** Uklanjanje prijave o nestaloj osobi
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj operacije je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Voditelj operacije odabire ekran "Nestali"
  2. Voditelj operacije odabire gumb "Obriši prijavu"
  3. Prijava o nestaloj osobi briše se iz baze podataka

#### UC8 - Brisanje komentara

- **Glavni sudionik:** Voditelj operacije
- **Cilj:** Brisanje komentara sa prijave o nestaloj osobi
- **Sudionici:** Baza podataka
- **Preduvjet:** Postoji barem 1 komentar na prijavi o nestaloj osobi
- **Opis osnovnog tijeka:**
  1. Voditelj operacije odabire željenu prijavu o nestaloj osobi
  2. Voditelj operacije odabire gumb "Obriši komentar" za željeni komentar

#### UC9 - Prikaz karte

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz područja na karti
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**

1. Nakon prijave korisniku je vidljiva karta sa označenim regijama i blokovima svih operacija

#### UC10 - Stvaranje operacije

- **Glavni sudionik:** Voditelj operacije
- **Cilj:** Definiranje područja koje je potrebno pretražiti
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj operacije je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Voditelj operacije odabire ekran "Karta"
  2. Voditelj operacije odabire gumb "Stvori novu operaciju"
  3. Voditelj operacije unosi potrebne podatke o operaciji: ime operacije i područje (regije i blokove) koje je potrebno pretražiti
  4. Podaci o stvorenoj operaciji spremaju se u bazu podataka
- **Opis mogućih odstupanja:**
  - 2.a Stvorena je operacija s već postojećim imenom
    1. Sustav onemogućuje stvaranje nove operacije, te javlja grešku.

#### UC11 - Mijenjanje statusa bloka

- **Glavni sudionik:** Kartograf
- **Cilj:** Promjena statusa bloka
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Kartograf odabire ekran "Karta"
  2. Kartograf odabire blok na kojem želi raditi
  3. Kartograf odabire gumb "Promijeni status bloka"
  4. Kartograf mijenja status bloka iz "nezapočeto" u "aktivno" te potom uređuje blok
  5. Kartograf mijenja status bloka iz "aktivno" u "provjera" nakon što je odradio posao
  6. Status bloka mijenja se iz "provjera" u "dovršeno" ako su dva kartografa provjerila odrađeni posao
  7. Promjena statusa bloka sprema se u bazu podataka
- **Opis mogućih odstupanja:**



2.a Kartograf odabire blok sa statusom "aktivan" kojeg uređuje drugi kartograf

1. Nije mu dopušteno uređivanje bloka uz ispis prigodne poruke

5.a Kartograf smatra da posao nije dobro odrađen

1. Kartograf mijenja status bloka iz "provjera" u "aktivno"

### UC12 - Dodavanje građevine u blok

- **Glavni sudionik:** Kartograf
- **Cilj:** Dodavanje nove građevine u blok
- **Sudionici:** Baza podataka
- **Preduvjet:** Blok ima status "aktivan"
- **Opis osnovnog tijeka:**
  1. Kartograf odabire ekran "Karta"
  2. Kartograf odabire blok na kojem želi unijeti građevinu
  3. Kartograf odabire gumb "Unesi novu građevinu"
  4. Kartograf unosi podatke o građevini
  5. Podaci o unesenoj građevini spremaju se u bazu podataka
- **Opis mogućih odstupanja:**
  - 4.a Građevina s tim podacima već postoji
    1. Sustav onemogućava dodavanje građevine te javlja grešku

### UC13 - Mijenjanje statusa građevini

- **Glavni sudionik:** Spasioc
- **Cilj:** Promjena statusa građevine
- **Sudionici:** Baza podataka
- **Preduvjet:** Spasioc je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Spasioc odabire ekran "Karta"
  2. Spasioc odabire građevinu kojoj želi promijeniti status
  3. Spasioc odabire gumb "Promijeni status građevine"
  4. Spasioc mijenja status u "pretraženo" ako je građevina pretražena ili u "nepretraženo" ako građevina nije pretražena
  5. Podaci o statusu građevine spremaju se u bazu podataka

### UC14 - Zaključavanje operacije

- **Glavni sudionik:** Voditelj operacije

- **Cilj:** Zaključavanje područja koja su uspješno kartografirana i pretražena
- **Sudionici:** Baza podataka
- **Preduvjet:** Područje je uspješno pretraženo
- **Opis osnovnog tijeka:**
  1. Voditelj operacije odabire ekran "Karta"
  2. Voditelj operacije klikne na operaciju koju želi zaključati
  3. Voditelj operacije odabire gumb "Zaključaj operaciju"
  4. Područje se više ne prikazuje kao područje koje je potrebno pretražiti

#### UC15 - Prikaz statistike

- **Glavni sudionik:** Voditelj operacije ili kartograf
- **Cilj:** Prikaz statistike humanitarne akcije
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj operacije ili kartograf je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Voditelj operacije ili kartograf odabire ekran "Statistika"
  2. Prikazuje se broj završenih blokova, broj prijavljenih nestalih i broj pronađenih osoba, te broj pretraženih i nepretraženih građevina kroz vrijeme

#### UC16 - Prikaz registriranih korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Prikaz svih registriranih korisnika te njihovih osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Administrator odabire ekran "Admin page"
  2. Administratoru se prikazuje popis korisnika te njihovih osobnih podataka

#### UC17 - Mijenjanje prava korisnicima

- **Glavni sudionik:** Administrator
- **Cilj:** Mogućnost mijenjanja prava registriranim korisnicima
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Administrator odabire ekran "Admin page"

2. Administrator kod željenog korisnika odabire gumb "Promijeni prava"
3. Administrator bira novo pravo tom korisniku
4. Novi podaci o pravu korisnika spremaju se u bazu podataka

### UC18 - Prikaz profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Prikaz osobnih podataka korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire ekran "Profile page"
  2. Korisniku se prikazuju njegovi osobni podaci

### UC19 - Uređivanje profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Mogućnost uređivanja osobnih podataka
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik odabire ekran "Profile page"
  2. Korisnik kod podatka koji želi promijeniti odabire gumb "Uredi"
  3. Korisnik unosi novi podatak na to mjesto
  4. Novi podaci o korisniku spremaju se u bazu podataka

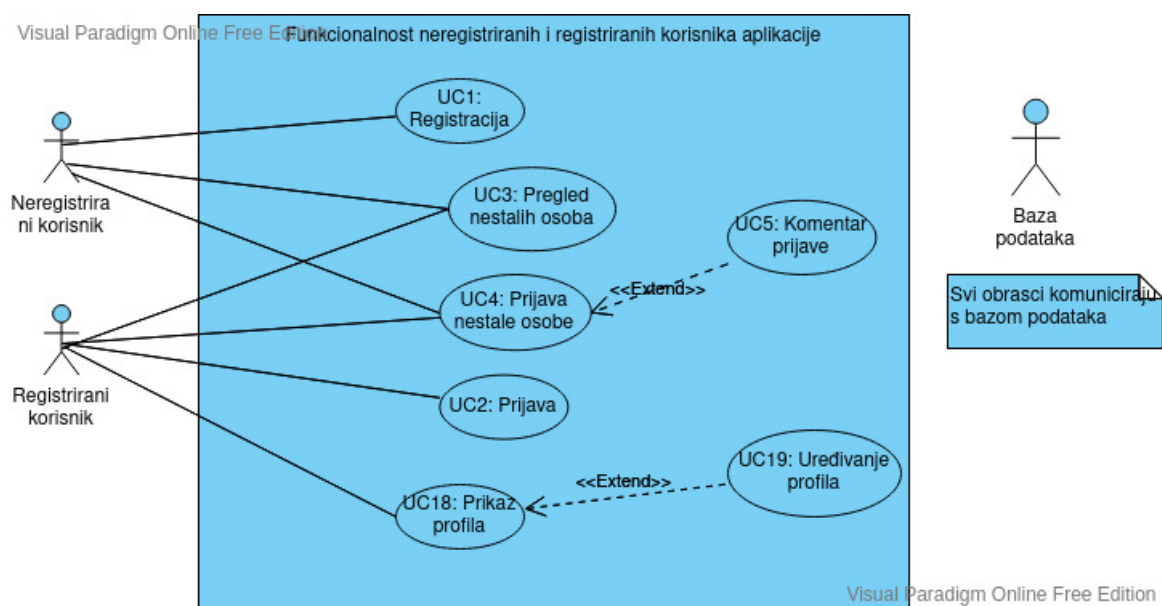
### UC20 - Potvrda registracije

- **Glavni sudionik:** Administrator
- **Cilj:** Mogućnost potvrde ili odbijanja registracije korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Administrator odabire ekran "Admin page"
  2. Administrator odabire gumb "Zahtjevi za registraciju"
  3. Administrator odabire gumb "Potvrdi registraciju"
  4. Podaci o korisniku spremaju se u bazu podataka
- **Opis mogućih odstupanja:**
  - 3.a Administrator ne želi potvrditi registraciju

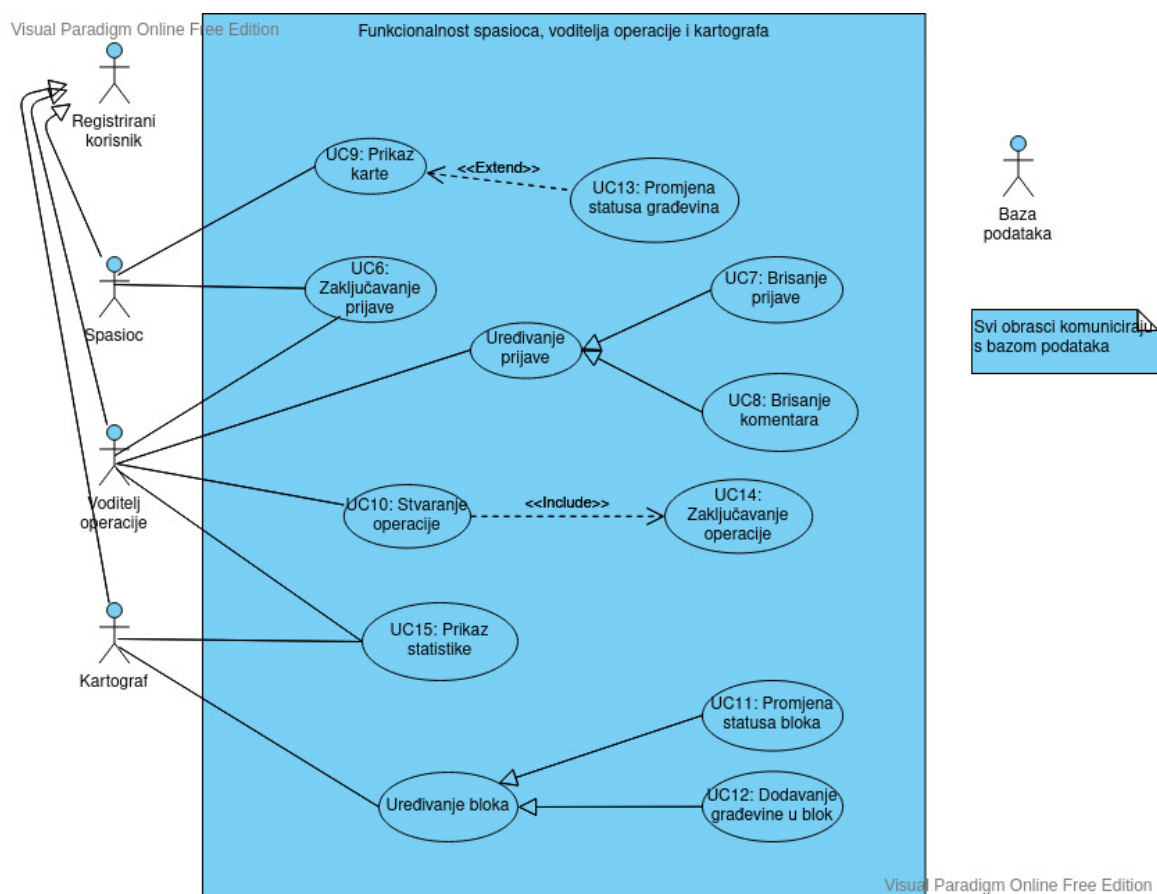
1. Administrator odabire gumb "Odbij registraciju"
2. Zahtjev za registracijom je odbijen te se korisniku ispisuje prigodna poruka

## Dijagrami obrazaca uporabe

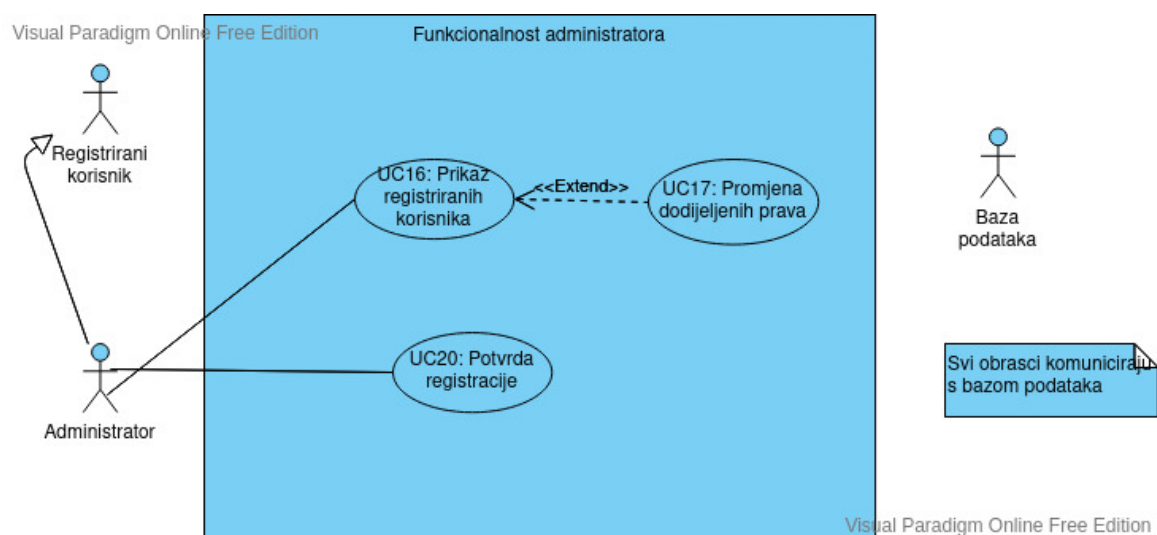
Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost neregistriranih i registriranih korisnika aplikacije



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost spasioca, voditelja operacije i kartografa

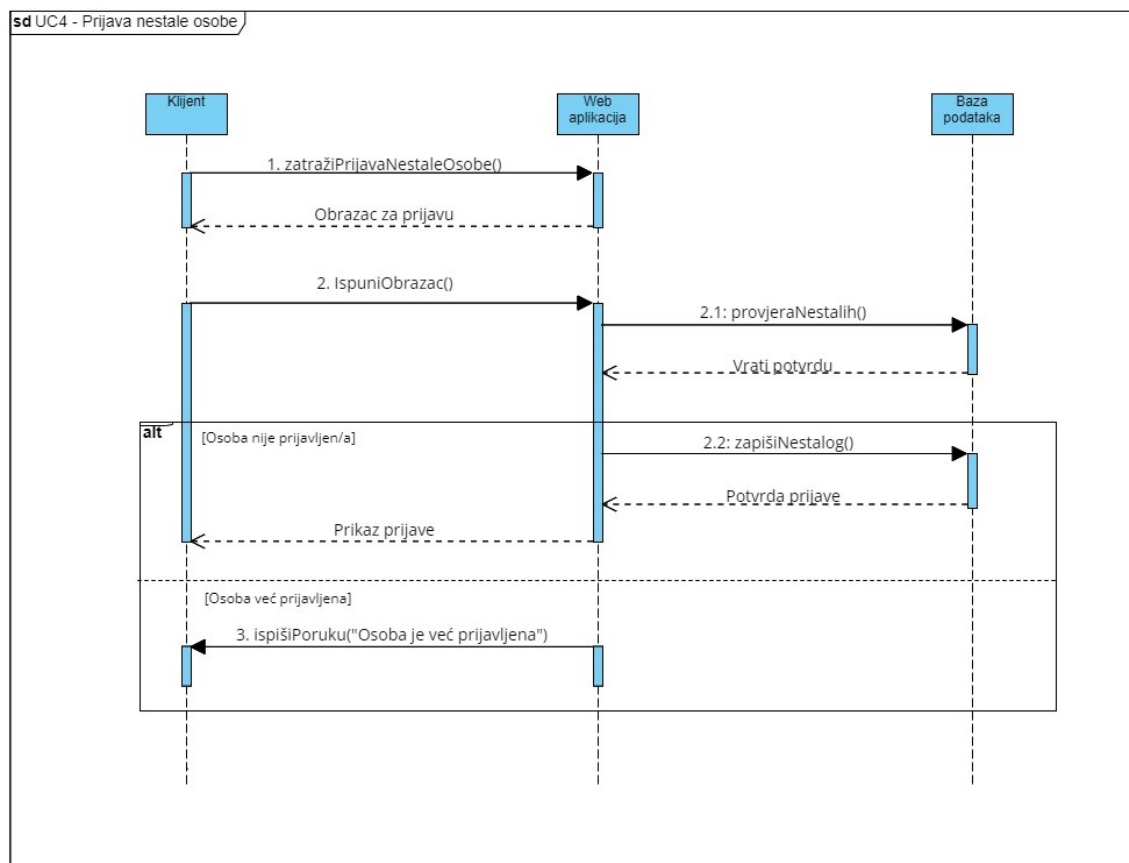


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC4 - Prijava nestale osobe

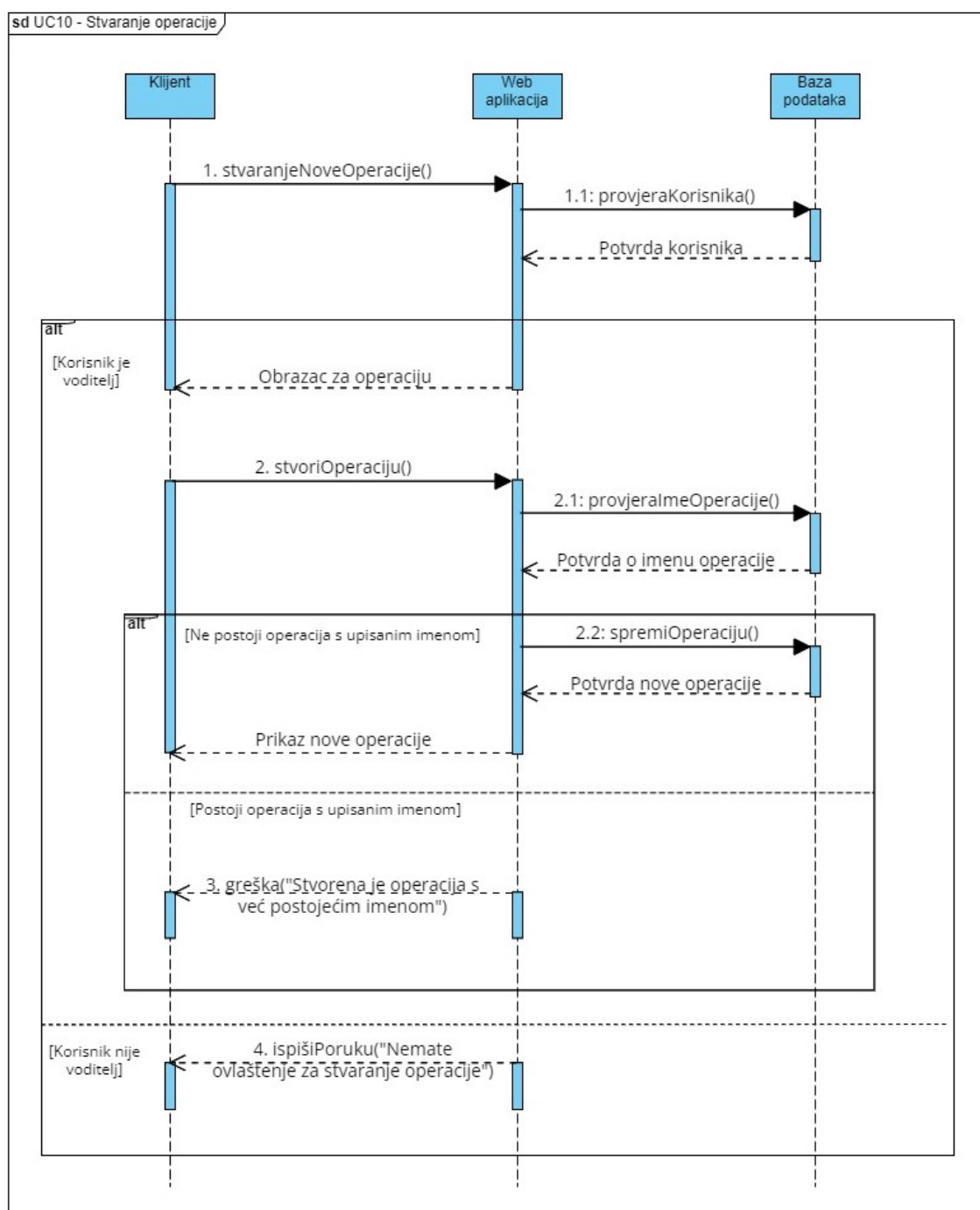
Klijent šalje zahtjev za prijavu nestale osobe. Poslužitelj prikazuje obrazac za prijavu. Klijent ispunjava obrazac (ime, prezime, fotografija i opis) za prijavu nestale osobe. Poslužitelj dohvaća podatke iz baze te provjerava je li nova prijavljena osoba u bazi. Ako osoba dosad nije prijavljena, zapisuje se u bazu nestalih te sustav šalje potvrdu i prikaz prijave. Ukoliko je osoba već prijavljena, poslužitelj obavještava klijenta uz poruku.



Slika 3.4: Sekvencijski dijagram za UC4 - Prijava nestale osobe

**Obrazac uporabe UC10 - Stvaranje operacije**

*Klijent šalje zahtjev za stvaranje operacije. Sustav provjerava je li korisnik voditelj. Ako korisnik nije voditelj, poslužitelj će ispisati poruku da nema ovlaštenje za stvaranje nove operacije, a ukoliko je, poslat će obrazac za stvaranje nove operacije. Klijent stvara novu operaciju, tj. upisuje ime operacije i područje (regije i blokove) koje je potrebno pretražiti. Poslužitelj dohvaća podatke iz baze te provjerava postoji li već operacija s tim imenom. Ako ne postoji, nova stvorena operacija se zapisuje u bazu podataka te sustav šalje potvrdu i prikaz nove operacije. Ukoliko već postoji operacija s tim imenom, sustav javlja grešku da je stvorena operacija s već postojećim imenom.*



Slika 3.5: Sekvencijski dijagram za UC10 - Stvaranje operacije



## 3.2 Ostali zahtjevi

### *dio 1. revizije*

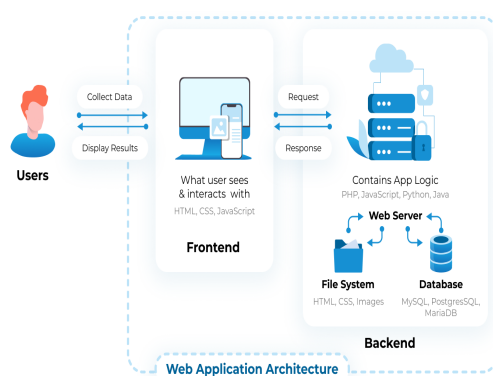
Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju **kako se sustav treba ponašati** i koja **ograničenja** treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: 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*

*Arhitektura sustava može se podijeliti na tri podsustava:*

- Web preglednik
- Web aplikacija
- Baza Podataka



Slika 4.1: Arhitektura sustava

**Web preglednik** je aplikacija za prikaz web stranica. Kada korisnik posjeti neku web stranicu, internetski preglednik zatraži podatke sa web poslužitelja, koje interpretira i prikaže na zaslon računala ili nekog drugog pametnog uređaja.

**Web poslužitelj** osnovni podsustav u izradi web aplikacije. Njegova uloga u sustavu je komunikacija korisnika(web preglednika) s aplikacijom. Komunikacija se ostvaruje preko HTTP (engl. Hyper Text Transfer Protocol) protokola. Uloga HTTP protokola je prijenos sadržaja s poslužitelja na preglednik, koji dalje odrađuje svoj posao.

Korisnik preko preglednika, koristi korisničko sučelje, te tako šalje HTTP zahtjeve na poslužitelj. Neki od zahtjeva su HTTP GET, HTTP POST itd. Svaki takav zahtjev u 99 % slučajeva na poslužitelju izaziva, njegovu komunikaciju s bazom podataka.

**Baza podataka** je skup međusobno povezanih podataka, pohranjeni u vanjskoj memoriji računala. Njezina uloga u sustavu je brza i efikasna pohrana podataka, koji se propagiraju iz sloja web preglednika, preko web poslužitelja.

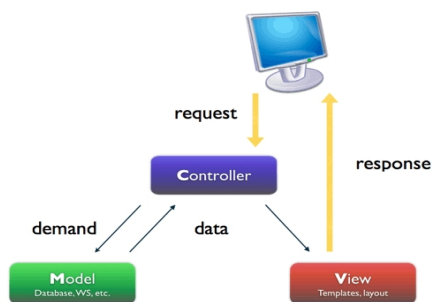
*Web aplikacija se najčešće dijeli na dva dijela:*

- **Front-end** koji je zadužen za razvoj korisničkog iskustva na webu.
- **Back-end** koji je zadužen za obradu i spremanje podataka, koji dođu sa front-enda.

Tehnologije koje smo uzeli za razvoj web aplikacije su React.js za Front-end u razvojnom okruženju Microsoft Visual Studio Code, te .NET za Back-end u razvojnom okruženju Microsoft Visual Studio. Uz .Net na back-endu, smo uzeli SQL server za bazu podataka. Arhitektura sustava će biti bazirana na MVC(engl. Model-View-Controller) konceptu.

**MVC** model sastoji se od tri komponenti:

- **Model** - komponenta modela koja je zadužena za dohvat i manipulacijom podataka. Često za obavljanje svojih zadaća koristi bazu podataka.
- **View** - komponenta kojoj je zadaća prikaz dobivenih podataka korisniku.
- **Controller** - komponenta zadužena za primanje zahtjeva od korisnika, koje dalje propagira komponenti Model.



Slika 4.2: MVC model

## 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

**Users** Ovaj entitet sadrži sve važne informacije o korisniku aplikacije. Sadrži attribute: korisničko ime, OIB, ime, prezime, fotografiju, broj telefona, adresu elektroničke pošte, lozinku, ID uloge korisnika te status je li potvrđen. Ovaj entitet u vezi je *Many-to-One* s entitetom Role preko atributa identifikatora uloge, u vezi *One-To-Many* s entitetom Operation preko atributa identifikatora osobe i u vezi *One-to-Many* s entitetom Comment preko atributa identifikatora osobe.

Users		
Username	VARCHAR	korisničko ime korisnika
OIB	BIGINT	jedinstveni identifikator osobe
FirstName	VARCHAR	ime korisnika
LastName	VARCHAR	prezime korisnika
PhoneNumber	VARCHAR	telefonski broj korisnika
EMail	VARCHAR	adresa elektroničke pošte korisnika
Password	VARCHAR	hash lozinke
RoleId	INT	identifikator uloge
Confirmed	BIT	status je li administrator potvrdio korisnika

**Role** Ovaj entitet sadrži sve važne informacije o ulogama korisnika aplikacije. Sadrži attribute ID uloge i ime uloge. Ovaj entitet u vezi je *One-to-Many* s entitetom Users preko atributa identifikatora uloge.

Role		
Id	INT	jedinstveni identifikator uloge
Name	VARCHAR	naziv uloge

**Missing Report** Ovaj entitet sadrži sve važne informacije o prijavama nestalih

osoba. Sadrži attribute: ID prijave, ime osobe, prezime osobe, OIB osobe, fotografiju osobe i opis prijave. Ovaj entitet u vezi je *One-to-Many* s entitetom Comment preko atributa identifikatora prijave nestale osobe.

Missing Report		
Id	INT	jedinstveni identifikator prijave nestale osobe
FirstName	VARCHAR	ime nestale osobe
LastName	VARCHAR	prezime nestale osobe
OIB	BIGINT	jedinstveni identifikator nestale osobe
Photo	VARCHAR	fotografija nestale osobe
Description	VARCHAR	opis prijave nestale osobe

**Comment** Ovaj entitet sadrži sve važne informacije o komentarima prijave nestalih osoba. Sadrži attribute ID komentara, ID prijave koja se komentira, sadržaj komentara, OIB korisnika. Ovaj entitet u vezi je *Many-to-One* s entitetom Missing Report preko atributa identifikatora prijave nestale osobe i u vezi *Many-to-One* s entitetom Users preko atributa identifikatora osobe.

Comment		
Id	INT	jedinstveni identifikator komentara
ReportId	INT	jedinstveni identifikator prijave nestale osobe
Text	VARCHAR	sadržaj komentara
UserOIB	BIGINT	jedinstveni identifikator osobe koja komentira

**Operation** Ovaj entitet sadrži sve važne informacije o operacijama. Sadrži attribute: ID operacije, status operacije i OIB voditelja operacije. Ovaj entitet u vezi je *Many-to-One* s entitetom Users preko atributa identifikatora osobe.

Operation		
Id	INT	jedinstveni identifikator operacije

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Operation		
Status	VARCHAR	status operacije
LeaderOIB	BIGINT	jedinstveni identifikator voditelja operacije

#### 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

# 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: 3. studenoga 2022.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

#### 2. sastanak

- Datum: u ovom formatu: 3. studenoga 2022.
- 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.*

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<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.*