

Programsko inženjerstvo

Ak. god. 2022./2022.

MapTasker

Dokumentacija, Rev. 1

Grupa: Armada4BrodaKeti

Voditelj: Bojan Puvača

Datum predaje: 18.11.2022.

Nastavnik: Hrvoje Nuić

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	19
3.2 Ostali zahtjevi	26
4 Arhitektura i dizajn sustava	27
4.1 Baza podataka	28
4.1.1 Opis tablica	29
4.1.2 Dijagram baze podataka	34
4.2 Dijagram razreda	35
4.3 Dijagram stanja	36
4.4 Dijagram aktivnosti	38
4.5 Dijagram komponenti	40
5 Implementacija i korisničko sučelje	41
5.1 Korištene tehnologije i alati	41
5.2 Ispitivanje programskog rješenja	42
5.2.1 Ispitivanje komponenti	42
5.2.2 Ispitivanje sustava	42
5.3 Dijagram razmještaja	43
5.4 Upute za puštanje u pogon	44
6 Zaključak i budući rad	54
Indeks slika i dijagrama	57
Dodatak: Prikaz aktivnosti grupe	58

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Bojan Puvača	26.10.2022.
0.2	Dijagrami obrazca uporabe	Matija Jelavić	29.10.2022.
0.5	Opis projektnog zadatka	Jurica Uglešić	30.10.2022.
0.6	Dodani funkcionalni zahtjevi i opisi za prvih 10 obrazaca uporabe	Katarina Šabić	30.10.2022.
0.8	Dodana 2 sekvencijska dijagrama	Ema Vlainić	01.11.2022.
0.9	Opis 5 tablica baze podataka	Matija Jelavić	02.11.2022.
0.10	Dodani opisi za preostalih 10 obrazaca uporabe	Danijel Kovačević	02.11.2022.
0.11	Opis arhitekture sustava	Antonio Mišić	02.11.2022.
0.12	Manje izmjene: izmijenjena prvog dijagrama obrazaca uporabe	Matija Jelavić	03.11.2022.
0.13.	Manje izmjene: korekcija prikaza dijagrama obrazaca uporabe	Matija Jelavić	05.11.2022.
0.14.	Manje izmjene: ispravak sekvencijskih dijagrama	Ema Vlainić	06.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.15.	Manje izmjene: ispravak funkcionalnih zahtjeva	Katarina Šabić	08.11.2022.
0.16.	Dijagram objekata	Bojan Puvača	08.11.2022.
0.17.	Dodani ostali zahtjevi sustava	Jurica Uglešić	09.11.2022.
0.18.	Ispravljeni opisi tablica pobaze podataka i dodani opisi za preostalih 5 tablica	Matija Jelavić Danijel Kovačević	12.11.2022.
0.19.	Ispravak opisa tablica baze podataka i dodani opisi za preostalih 5 tablica	Matija Jelavić Danijel Kovačević	12.11.2022.
0.20.	Dodana presotala 2 sekvencijska dijagrama	Ema Vlainić	16.11.2022.
1.0	Korigiranje teksta i provjera dokumentacije	Katarina Šabić	18.11.2022.

2. Opis projektnog zadatka

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 voditelji.

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 vodicima 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
 - Fotografija
 - Opis nestale osobe
2. Poslati zahtjev za registraciju na aplikaciju pri čemu je potrebno navesti:
 - Uloga za koju se prijavljuje (spasioc, voditelj, 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.

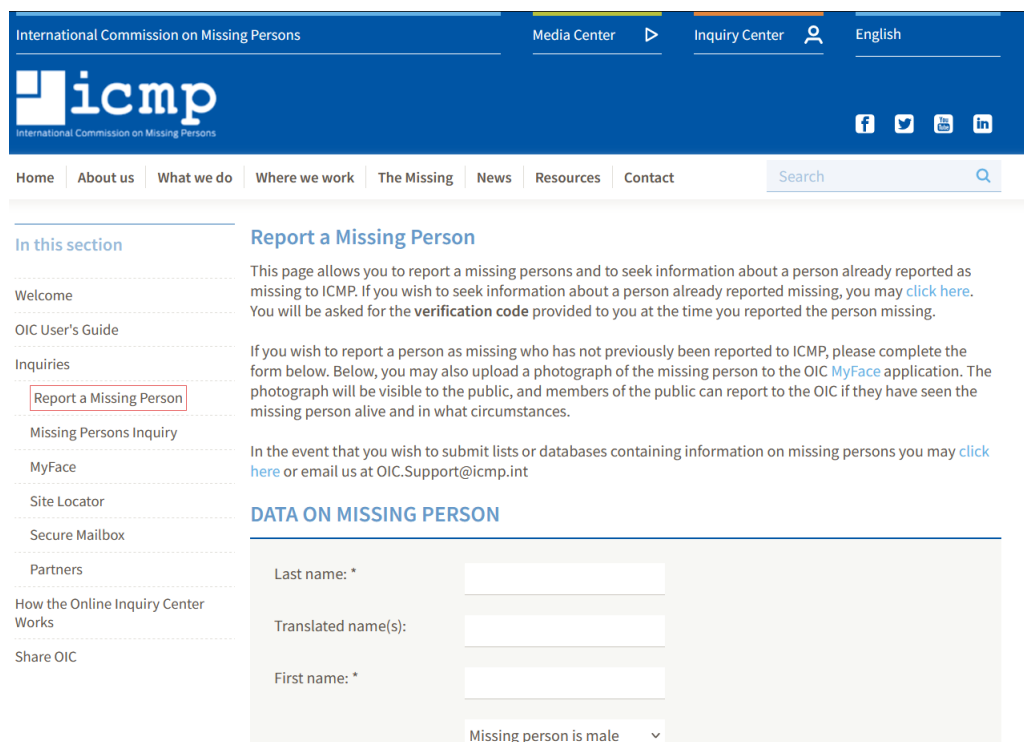
Administrator aplikacije ima pristup popisu svih registriranih korisnika te njihovim osobnim podacima. Uloga mu je potvrđivanje ili odbacivanje zahtjeva za registraciju korisnika te može mijenjati dodijeljena prava korisnicima.

Na internetu već je dostupna aplikacija slična *MapTaskeru* od **Međunarodne komisije za nestale osobe** (*International Commission on Missing Persons*). Prijave nestalih osoba preko *MapTaskera* i *ICMP-a* postoje mnoge sličnosti i razlike.

Sličnosti: Informacije koje je potrebno popuniti tijekom prijave, kao što su ime, prezime te opis nestale osobe. Moguće je dodati fotografiju osobe koja će postati vidljiva javnosti te njezini članovi mogu komentirati jesu li vidjeli nestalu osobu.

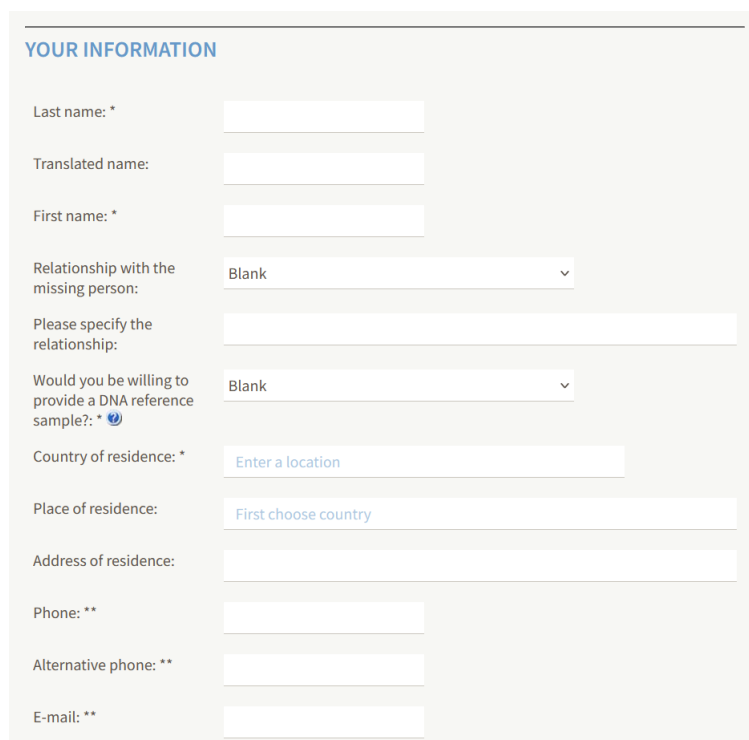
Razlike: U formi za prijavu su dostupna dodatna polja za popunjavanje: spol, datum rođenja, država i grad kojoj osoba pripada, datum i mjesto nestanka osobe, visina i težina. Preko *ICMP-a* anonimna prijava nestale osobe nije moguća pošto je obavezno popuniti i polja s podacima osobe koja stvara prijavu. Također je moguće poslati liste ili baze podataka koje sadrže informacije o nestalim osobama.

Moguća proširenja bi uključivala dodatna polja informacije o nestaloj osobi kao što su mjerenja, spol te gdje i kada je zadnji put osoba viđena. Također bi se mogao dodati sustav za doniranje obiteljima nestalih.



The screenshot shows the ICMP (International Commission on Missing Persons) website. The header includes the ICMP logo and navigation links: Home, About us, What we do, Where we work, The Missing, News, Resources, and Contact. A search bar is also present. The main content area is titled 'Report a Missing Person' and contains instructions on how to report a missing person. A sidebar on the left lists various services: Welcome, OIC User's Guide, Inquiries, Report a Missing Person (highlighted), Missing Persons Inquiry, MyFace, Site Locator, Secure Mailbox, Partners, How the Online Inquiry Center Works, and Share OIC. The 'Report a Missing Person' form includes fields for Last name, Translated name(s), First name, and a dropdown for 'Missing person is male'.

Slika 2.1: Prijava nestale osobe preko ICMP



The screenshot shows the 'YOUR INFORMATION' form on the ICMP website. The form includes fields for Last name, Translated name, First name, Relationship with the missing person, Please specify the relationship, Would you be willing to provide a DNA reference sample?, Country of residence, Place of residence, Address of residence, Phone, Alternative phone, and E-mail. The form is designed to collect personal information from the user reporting a missing person.

Slika 2.2: Popunjavanje informacije korisnika

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

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
 - (c) komentirati prijavu o nestaloj osobi
2. Voditelj operacije može:
 - (a) imati sve ovlasti kao i neregistrirani korisnik
 - (b) zaključati prijavu o nestaloj osobi nakon što osoba bude pronađena
 - (c) obrisati prijavu o nestaloj osobi i komentare na prijavi
 - (d) definirati područja na karti koja je potrebno pretražiti i kartografirati
 - (e) imati pristup statistici humanitarne akcije
 - (f) mijenjati podatke u svom profilu
3. Kartograf može:
 - (a) imati sve ovlasti kao i neregistrirani korisnik
 - (b) promijeniti status odabranog bloka na karti iz nezapočetog u aktivno, i iz aktivnog u provjeru

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

4. Spasioc može:

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

5. Administrator može:

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

3.1.1 Obrasci uporabe

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

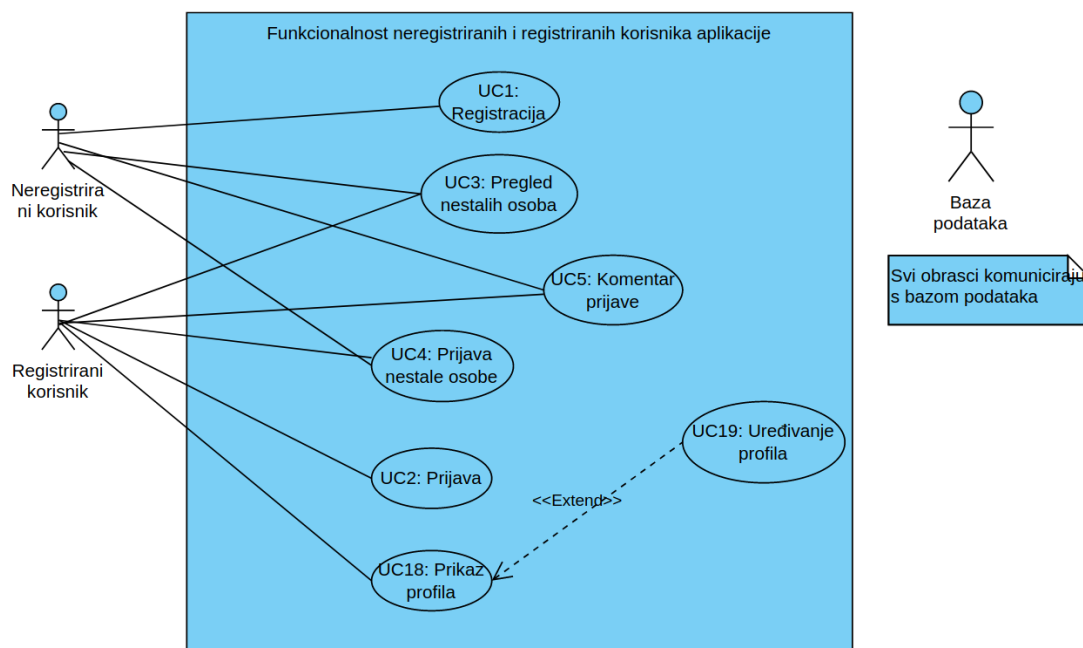
UC19 - Uređivanje profila

- **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

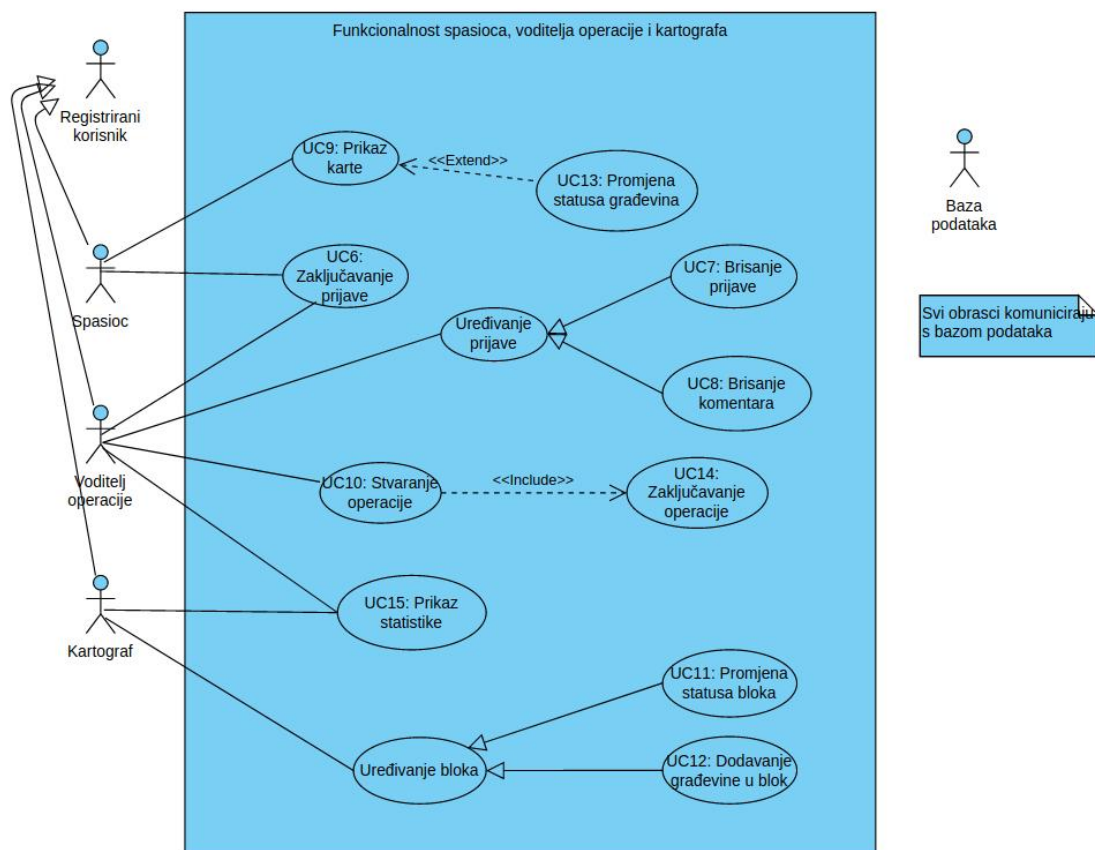
UC20 - Potvrda registracije

- **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 ispisiuje prigodna poruka

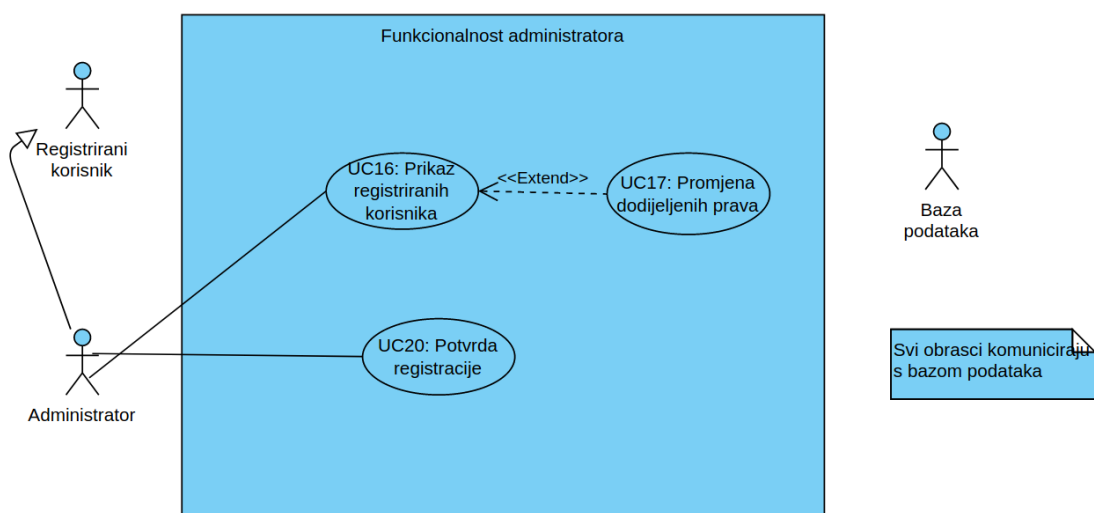
Dijagrami obrazaca uporabe



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



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

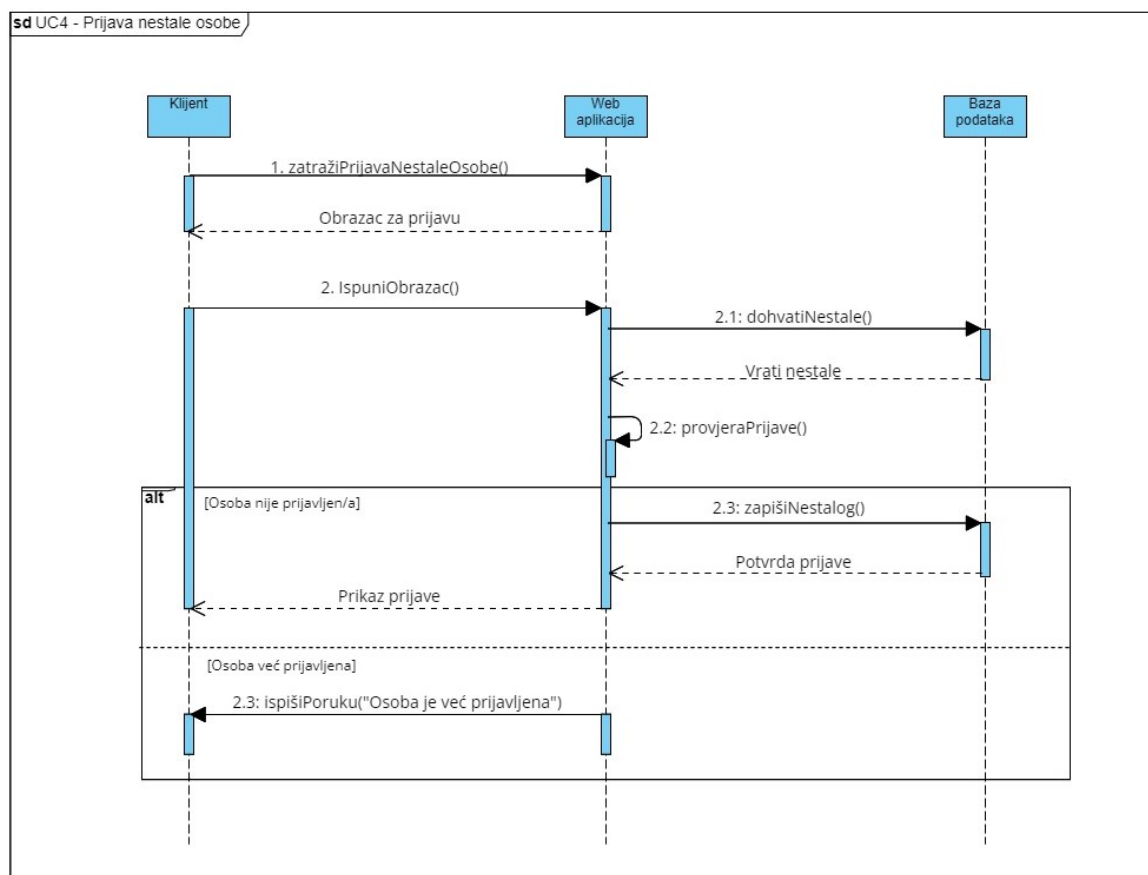


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC4 - Prijava nestale osobe

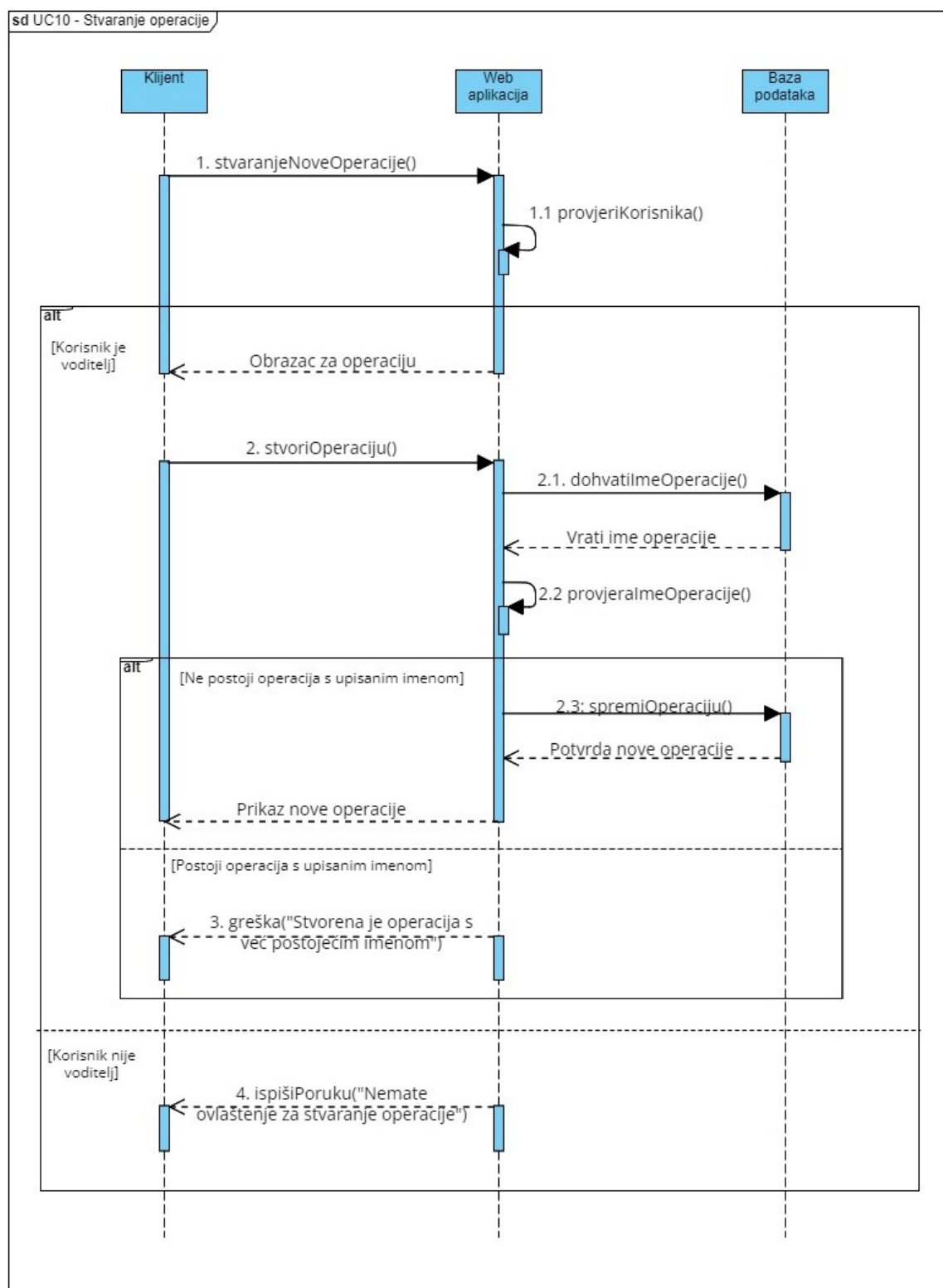
Klijent šalje zahtjev za prijavu nestale osobe. Poslužitelj prikazuje obrazac za prijavu. Klijent ispunjava obrazac (ime, prezime, fotografija i opis) za prijavu nestale osobe. Poslužitelj dohvaća podatke iz baze te provjerava je li nova prijavljena osoba u bazi. Ako osoba dosad nije prijavljena, zapisuje 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

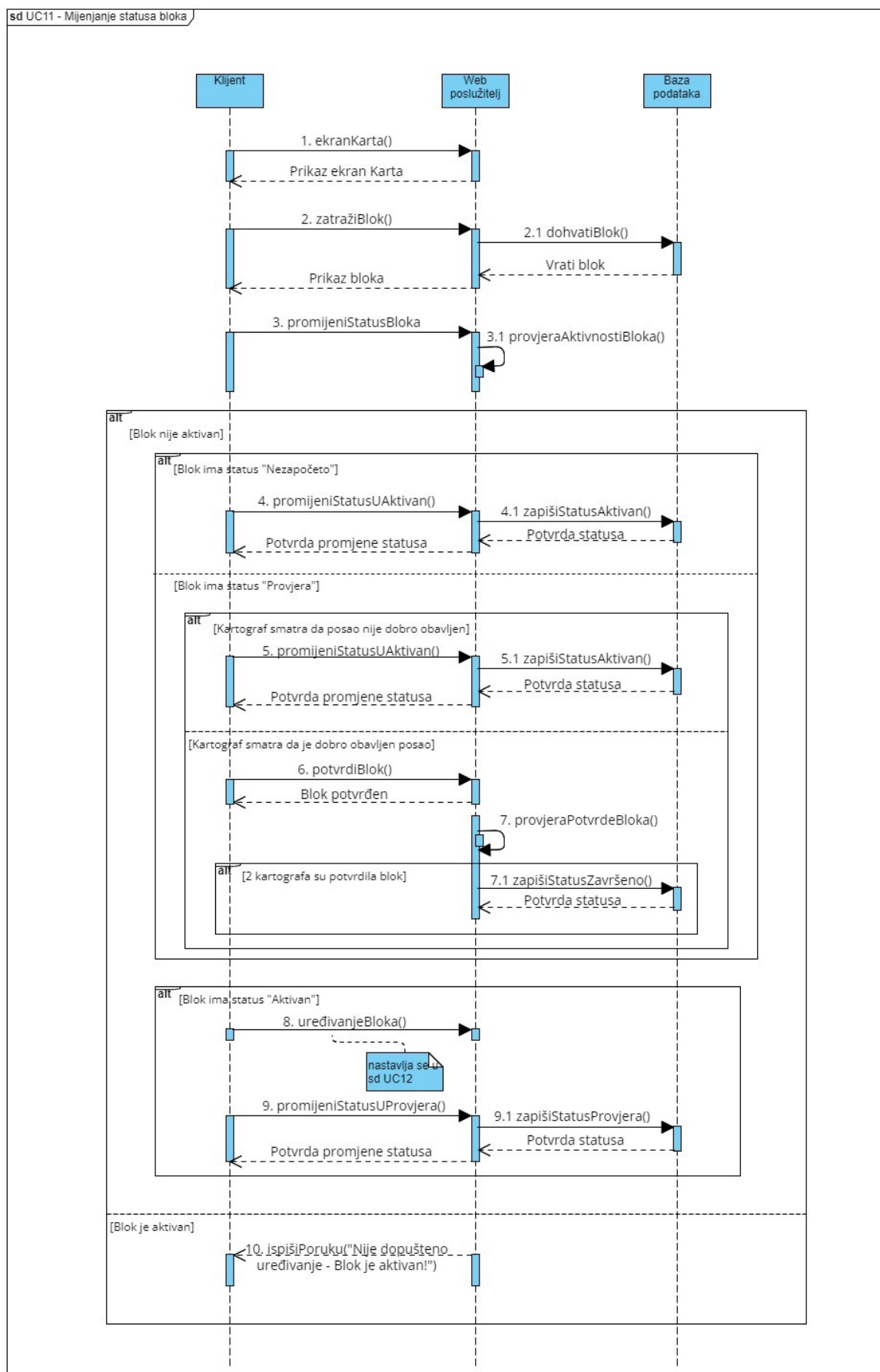
Klijent šalje zahtjev za stvaranje operacije. Sustav provjerava je li korisnik vo-
ditelj. Ako korisnik nije vođitelj, poslužitelj će ispisati poruku da nema ovlaštenje
za stvaranje nove operacije, a ukoliko je, poslat će obrazac za stvaranje nove ope-
racije. Klijent stvara novu operaciju, tj. upisuje ime operacije i područje (regije i
blokove) koje je potrebno pretražiti. Poslužitelj dohvaća podatke iz baze te provje-
rava 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

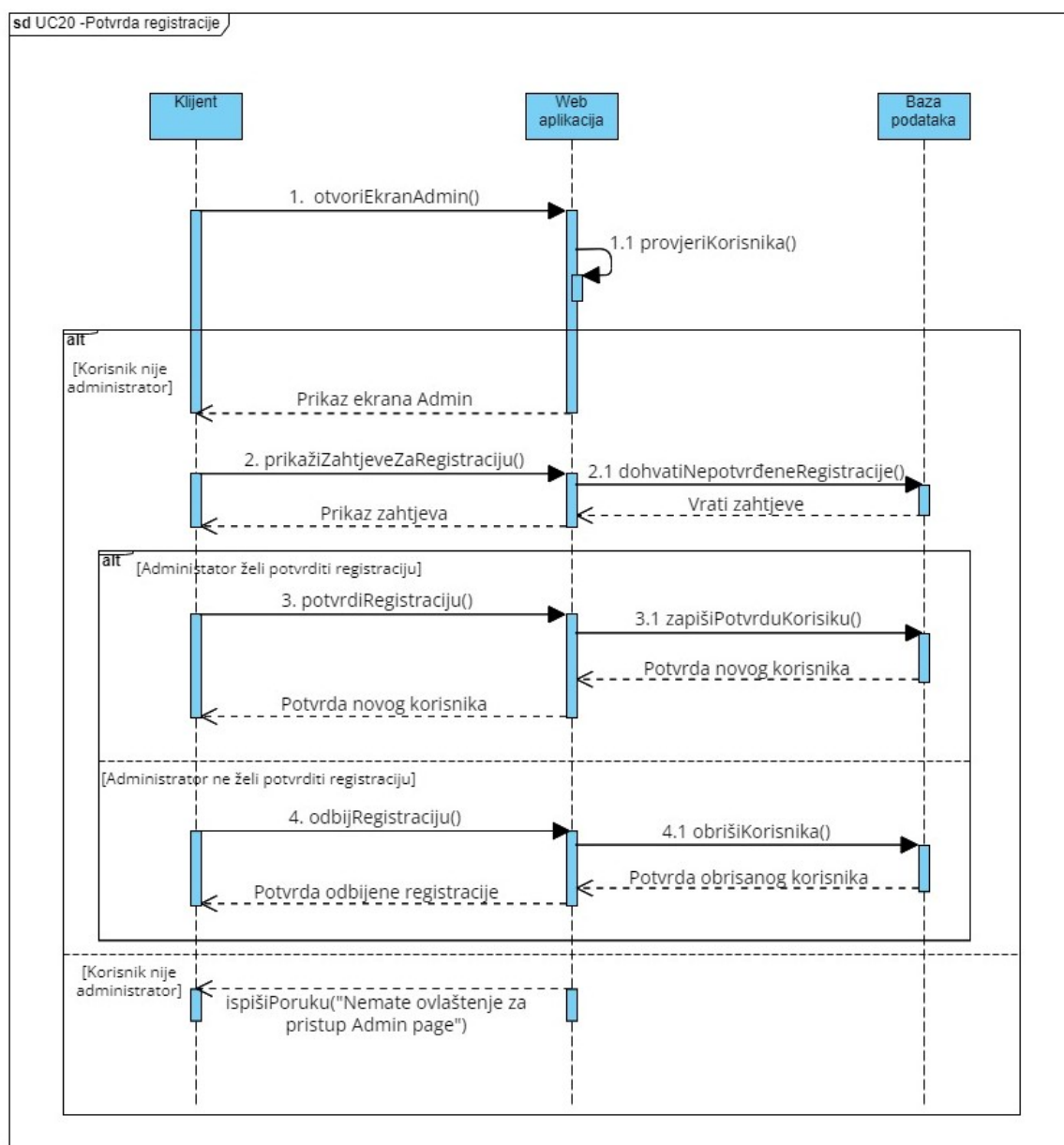
Obrazac uporabe UC11 - Mijenjanje statusa bloka

Klijent šalje zahtjev za prikaz karte. Web poslužitelj prikazuje kartu s opcijama, oviseći o korisniku. Klijent šalje zahtjev za prikaz bloka koji želi uređivati. Sustav dohvaća iz baze taj blok i prikazuje ga klijentu. Klijent šalje zahtjev za promjenu statusa bloka. Web poslužitelj provjerava status bloka te ako je blok aktivan ispisuje poruku da nije dopušteno uređivanje bloka. Ako je blok u statusu "Nezapočeto", klijent šalje zahtjev za promjenu statusa u "Aktivan". Sustav zapisuje status u bazu te prikazuje potvrdu promjene statusa. Ako je blok u statusu "Provjera", kartograf može smatrati da je posao dobro obavljen ili nije. Ukoliko kartograf smatra da posao nije dobro obavljen, šalje zahtjev sustavu za promjenu statusa bloka u "Aktivan" koji se zapisuje u bazu. Ukoliko kartograf smatra da je posao dobro obavljen, potvrđuje blok. Ukoliko su 2 kartografa potvrdila blok, mijenja se status bloka u "Završeno" i zapisuje u bazu. Kad je status bloka aktivan klijent može uređivati blok što se nastavlja u UC12 - Dodavanje građevine u blok. Kad je kartograf gotov s uređivanjem bloka, šalje zahtjev za promjenu statusa u "Provjera". Sustav zapisuje promjenu u bazu te šalje potvrdu.



Obrazac uporabe UC20 - Potvrda registracije

Klijent šalje zahtjev za otvaranjem ekrana admin. Sustav provjerava je li korisnik administrator te ako nije, web poslužitelj ispisuje poruku da nema ovlaštenja za pristup Admin pageu. Ukoliko je korisnik administrator, sustav prikazuje ekran Admin. Klijent šalje zahtjev za nepotvrđene registracije. Sustav dohvaća iz baze nepotvrđene korisnike te ih prikazuje. Ako administrator želi potvrditi registraciju, šalje zahtjev sustavu koji zapisuje zapisuje potvrdu korisnika u bazu. Sustav prikazuje potvrdu novog korisnika. Ukoliko administrator ne želi potvrditi registraciju, šalje zahtjev sustavu. Sustav briše korisnika iz baze te šalje potvrdu odbijene registracije.



Slika 3.7: Sekvencijski dijagram za UC20 - Potvrda registracije

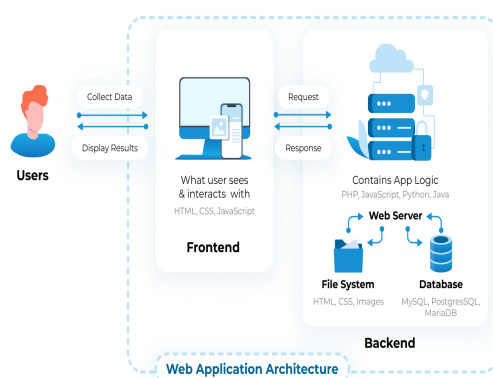
3.2 Ostali zahtjevi

- Učitavanje početne stranice, stranice prijave korisnika ili nestalih osoba ne smije trajati duže od nekoliko sekundi
- Izgled aplikacije treba biti jednostavan kako bi se novi korisnici s lakoćom služili njezinim funkcionalnostima bez dodatnih pomagala (*user-friendly*)
- Proces prijave ili registracije korisnika ne smije trajati duže od nekoliko sekundi
- Sustav treba podržavati rad i operacije više korisnika u isto vrijeme bez ikakvih poteškoća
- Operacije vezane uz baze podataka trebaju biti brzo i efikasno izvršene
- Baza podataka treba biti pouzdana i dobro zaštićena
- U bazi podataka ne smiju postojati više operacija, korisnika te prijavljenih nestalih osoba s istim imenom
- Proces prijave nestale osobe ne smije predugo trajati
- Sustav treba podržavati korištenje diakritičkih znakova pri unosu imena osoba ili operacija
- Promjena statusa ili faza blokova u bilo kojem smjeru ne smije narušavati funkcionalnost sustava

4. Arhitektura i dizajn sustava

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.

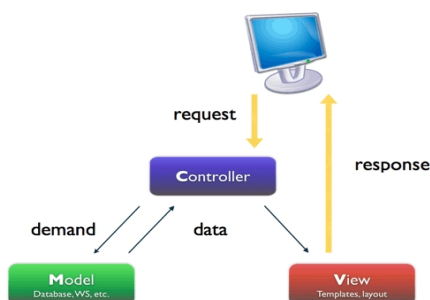
Web aplikacija se najčešće dijeli na dva dijela:

- **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-end-a.

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

Za potrebe našeg sustava koristimo relacijsku Microsoft SQL Server bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana te izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Users
- Role
- Missing Report
- Comment
- Operation
- Area
- Region
- Block
- Building
- Point

4.1.1 Opis tablica

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, u vezi *One-to-Many* s entitetom Comment preko atributa identifikatora osobe, u vezi *One-To-Many* s entitetom Area preko atributa identifikatora osobe te u vezi *One-To-Many* s entitetom Block preko atributa identifikatora osobe.

Users		
Username	VARCHAR	korisničko ime korisnika
OIB	BIGINT	jedinstveni identifikator osobe
FirstName	VARCHAR	ime korisnika
LastName	VARCHAR	prezime korisnika
Photo	VARCHAR	fotografija korisnika
PhoneNumber	VARCHAR	telefonski broj korisnika
EMail	VARCHAR	adresa elektroničke pošte korisnika
Password	VARCHAR	hash lozinke

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Users		
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, opis prijave, datum i vrijeme prijave te datum i vrijeme pronalaska. 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
ReportedAt	DATETIME	datum i vrijeme prijave nestanka osobe
FoundAt	DATETIME	datum i vrijeme pronalaska nestale osobe

Comment Ovaj entitet sadrži sve važne informacije o komentarima prijava 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 te u vezi *One-To-Many* s entitetom Region preko atributa identifikatora operacije.

Operation		
Id	INT	jedinstveni identifikator operacije
Status	VARCHAR	status operacije
LeaderOIB	BIGINT	jedinstveni identifikator voditelja operacije

Area Ovaj entitet sadrži sve važne informacije o područjima. Sadrži attribute: identifikator područja, datum i vrijeme nastanka područja, datum i vrijeme zatvaranja područja te identifikator osobe koja je zadnja uređivala područje. Ovaj entitet u vezi je *Many-to-One* s entitetom Users preko atributa identifikatora osobe, u vezi *One-To-Many* s entitetom Block preko atributa identifikatora područja, u vezi *One-To-Many* s entitetom Building preko atributa identifikatora područja, u vezi *One-To-Many* s entitetom Point preko atributa identifikatora područja te u vezi *One-To-Many* s entitetom Region preko atributa identifikatora područja.

Area		
Id	INT	jedinstveni identifikator područja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Area		
CreatedAt	DATETIME	datum i vrijeme nastanka područja
ClosedAt	DATETIME	datum i vrijeme zatvaranja područja
UpdatedLastByOIB	BIGINT	identifikator osobe koja je zadnja uređivala područje

Region Ovaj entitet sadrži sve važne informacije o regiji. Sadrži attribute: identifikator regije te identifikator operacije. Ovaj entitet u vezi je *Many-to-One* s entitetom Area preko atributa identifikatora regije, u vezi *Many-To-One* s entitetom Operation preko atributa identifikatora operacije te u vezi *One-To-Many* s entitetom Block preko atributa identifikatora regije.

Region		
AreaId	INT	jedinstveni identifikator regije
OperationId	INT	jedinstveni identifikator operacije

Block Ovaj entitet sadrži sve važne informacije o blokovima. Sadrži attribute: identifikator bloka, status bloka, identifikator regije te identifikator osobe koja uređuje blok. Ovaj entitet u vezi je *Many-to-One* s entitetom Users preko atributa identifikatora osobe, u vezi *Many-To-One* s entitetom Area preko atributa identifikatora bloka, u vezi *Many-To-One* s entitetom Region preko atributa identifikatora regije te u vezi *One-To-Many* s entitetom Building preko atributa identifikatora bloka.

Block		
AreaId	INT	jedinstveni identifikator bloka
Status	VARCHAR	status bloka
RegionId	INT	jedinstveni identifikator regije
ActiveForOIB	BIGINT	identifikator kartografa koji trenutno uređuje blok

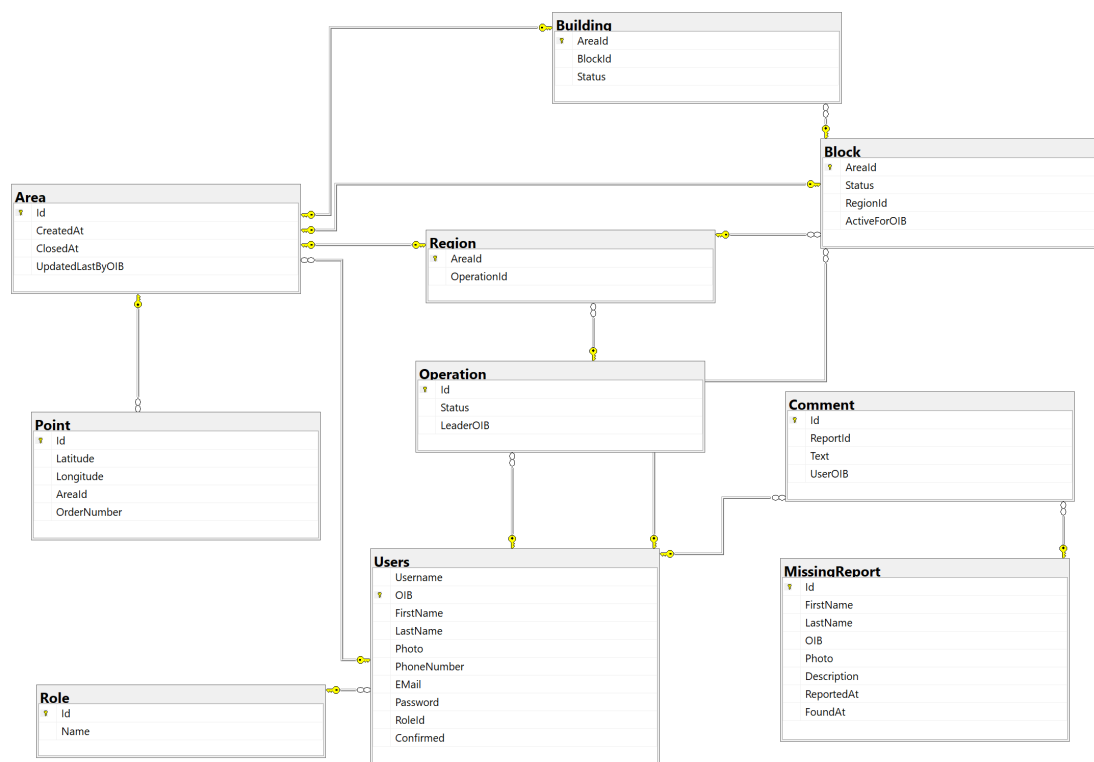
Building Ovaj entitet sadrži sve važne informacije o građevinama. Sadrži atribute: identifikator građevine, identifikator bloka te status građevine. Ovaj entitet u vezi je *Many-to-One* s entitetom Area preko atributa identifikatora građevine te u vezi *Many-To-One* s entitetom Block preko atributa identifikatora bloka.

Building		
AreaId	INT	jedinstveni identifikator građevine
BlockId	INT	jedinstveni identifikator bloka
Status	VARCHAR	status građevine

Point Ovaj entitet sadrži sve važne informacije o točkama i njihovim koordinatama. Sadrži atribute: identifikator točke, geografska širina, geografska dužina, identifikator područja te redni broj. Ovaj entitet u vezi je *Many-to-One* s entitetom Area preko atributa identifikatora područja.

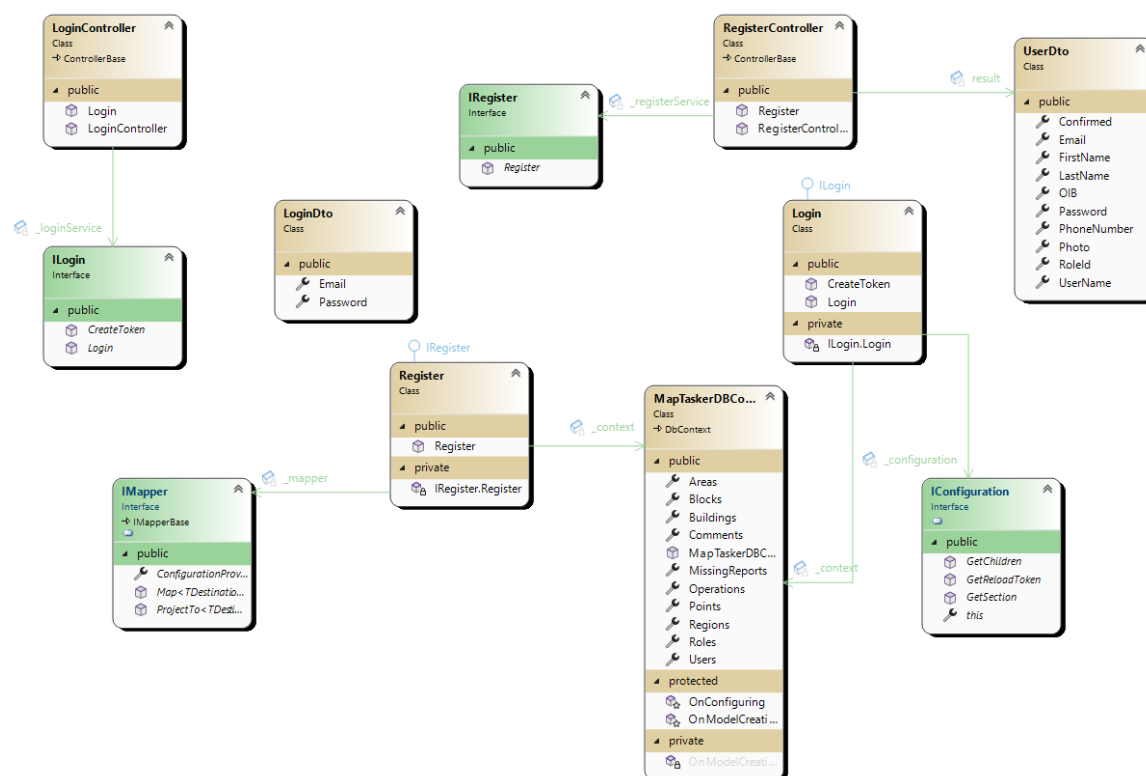
Point		
Id	INT	jedinstveni identifikator točke
Latitude	FLOAT	geografska širina točke
Longitude	FLOAT	geografska dužina točke
AreaId	INT	jedinstveni identifikator područja
OrderNumber	INT	redni broj točke

4.1.2 Dijagram baze podataka

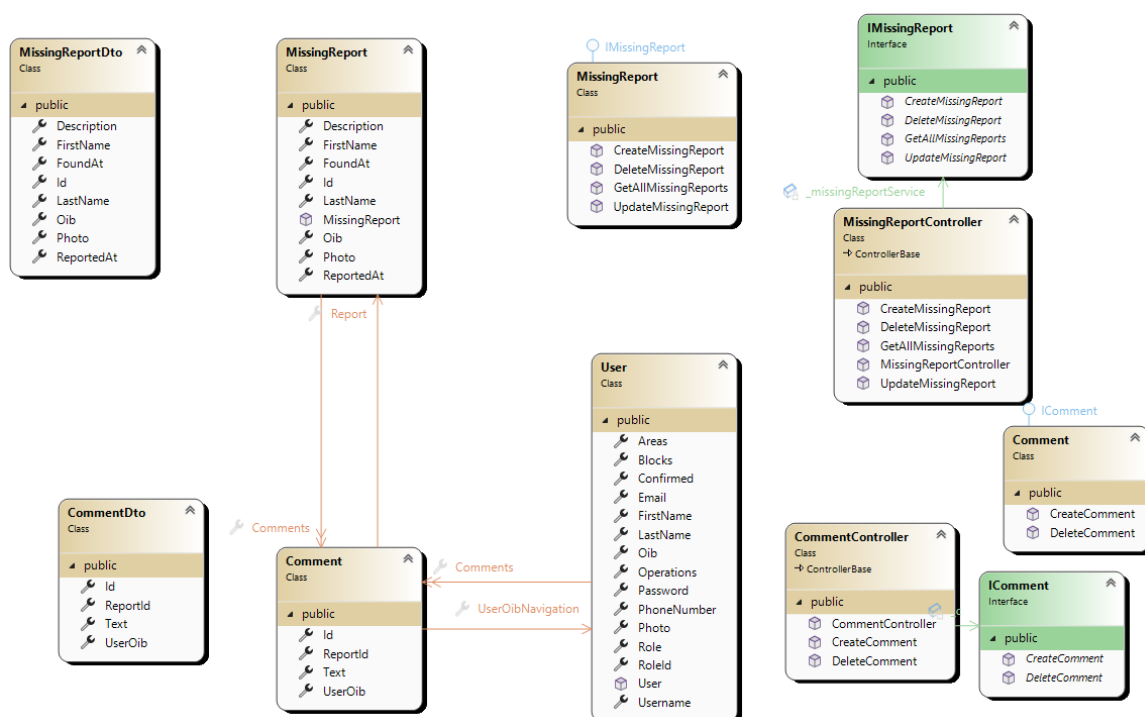


Slika 4.3: ER dijagram baze podataka

4.2 Dijagram razreda



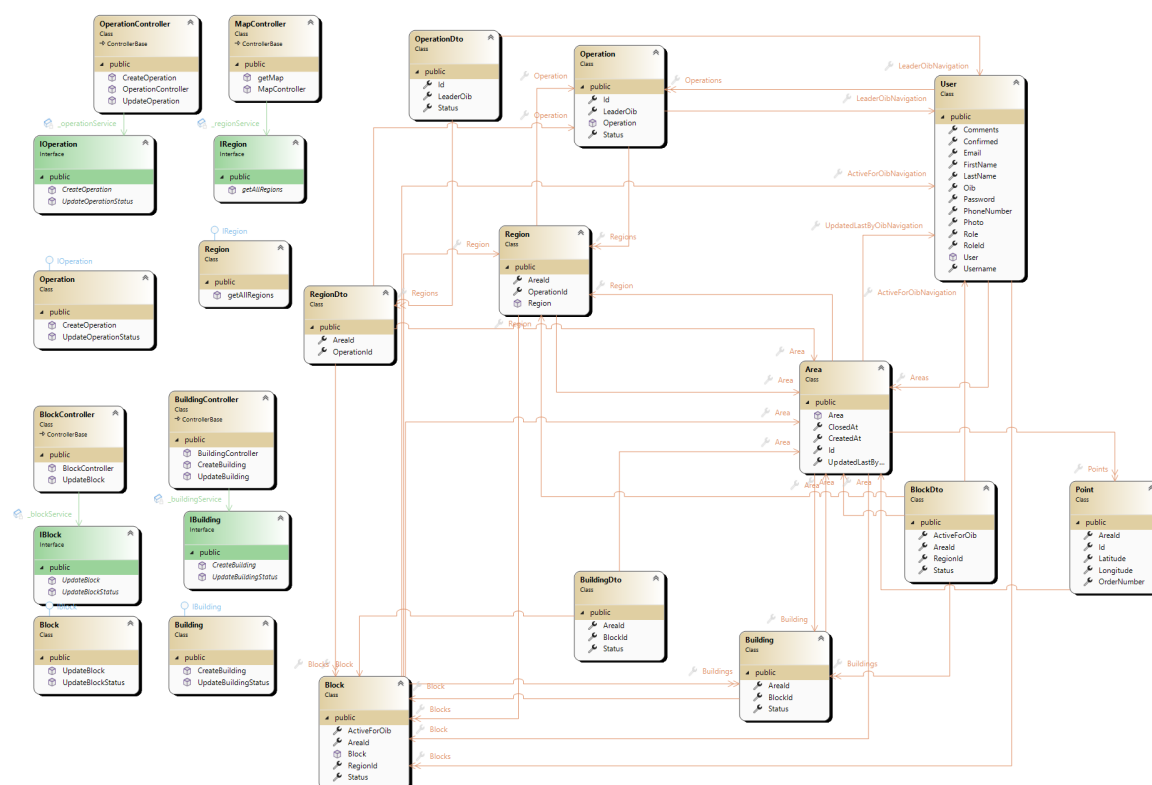
Slika 4.4: Dijagram razreda 1 - Register i Login



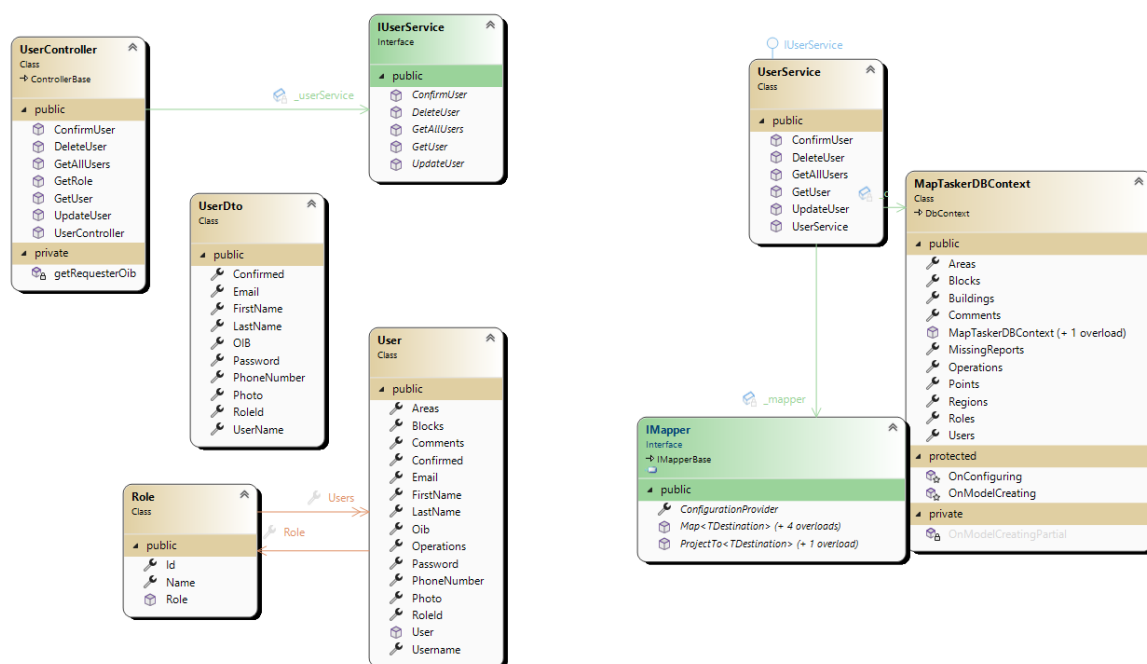
Slika 4.5: Dijagram razreda 2 - Missing Report i Comment

4.3 Dijagram stanja

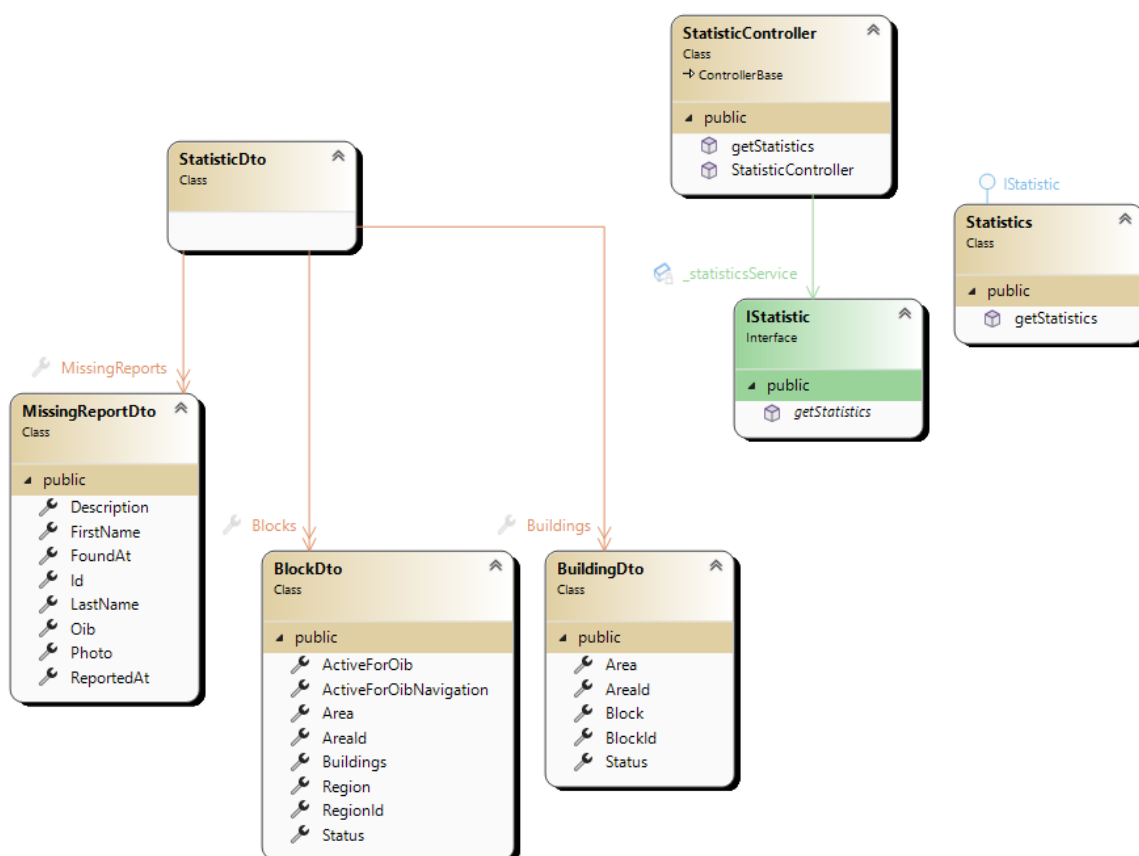
Dijagram stanja koristi se kako bi se opisalo dinamičko ponašanje nekog dijela sustava u vremenu. Njime se prikazuju prijelazi između stanja pojedinih objekata. Korisnik prilikom registracije unosi svoje podatke i bira ulogu za koju se prijavljuje. Na slici 4.9 prikazan je dijagram stanja za registriranog korisnika koji je odabrao ulogu voditelja operacije. Nakon prijave u sustav, voditelju se prikazuje karta na kojoj može stvarati nove operacije, to jest definirati područja koja je potrebno pretražiti. Svakoj operaciji daje ime. Voditelj može u bilo kojem trenutku pregledati podatke o svom profilu i po potrebi ih izmijeniti. Također, voditelj osim što može prijaviti nestalu osobu, ima uvid u popis nestalih osoba i može zaključiti prijavu o nestaloj osobi nakon što je ona pronađena, i dodavati i brisati komentare na prijave. Uz to, voditelj može i obrisati prijavu o nestaloj osobi. Na kraju, voditelj ima pristup statistici o nestalim osobama.



Slika 4.6: Dijagram razreda 3 - Operations, Areas, Regions, Blocks, Buildings



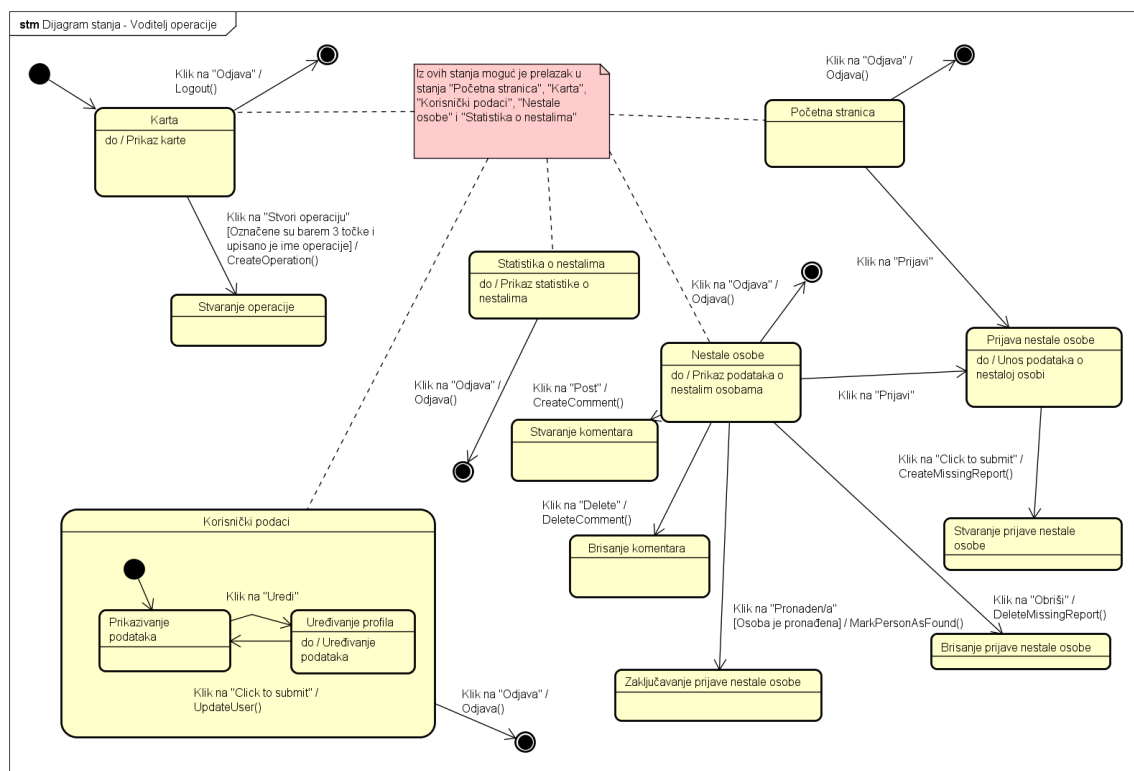
Slika 4.7: Dijagram razreda 4 - User i Role



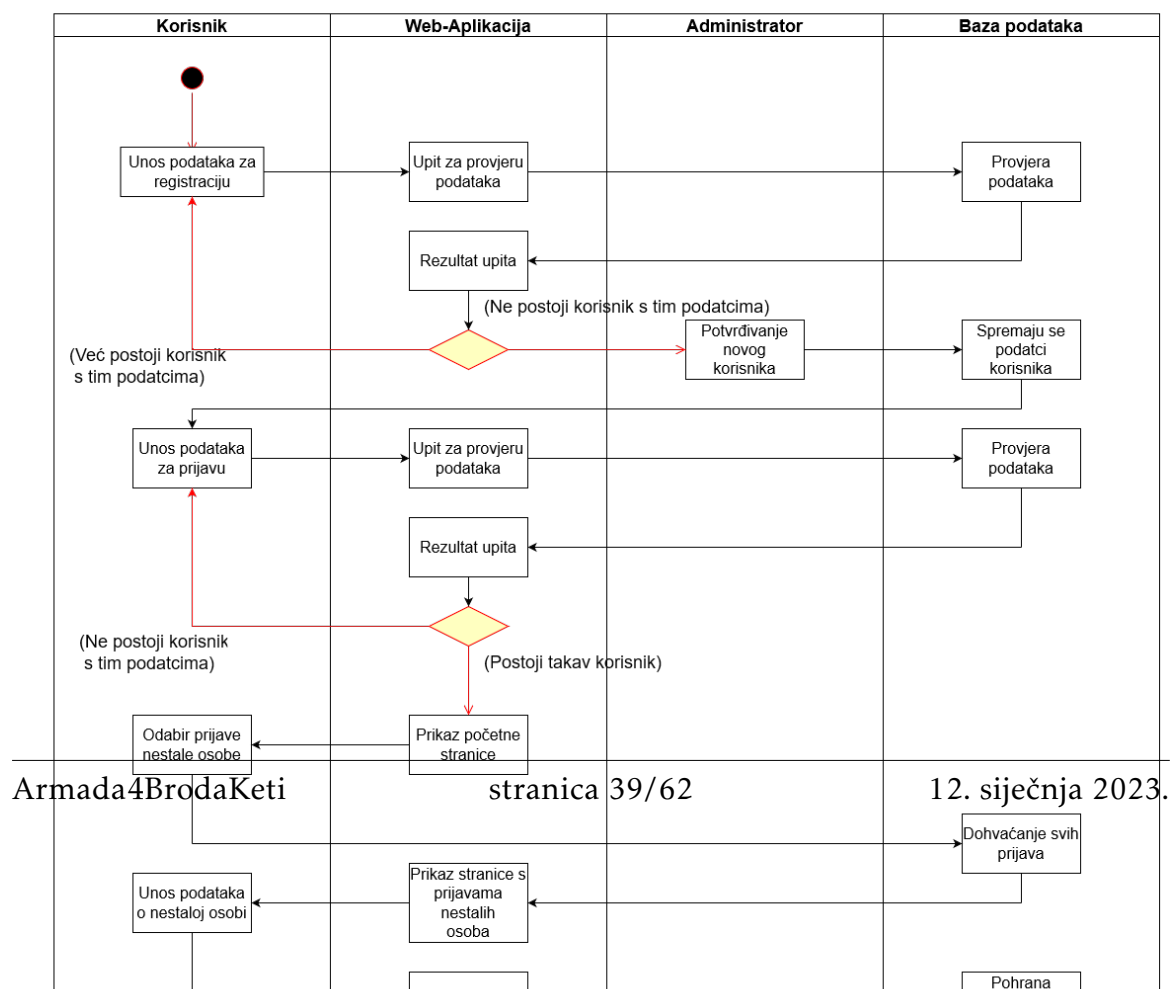
Slika 4.8: Dijagram razreda 5 - Statistics

4.4 Dijagram aktivnosti

Dijagram aktivnosti 4.9 opisuje proces registracije korisnika koju administrator potvrđuje. Nakon što se uspješno provede prijava odobrenog korisnika, korisnik ide na stranicu gdje može objaviti prijavu nestale osobe nakon što su uneseni svi potrebni podatci.



Slika 4.9: Dijagram stanja - Vodiitelj operacije



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 Selenium¹. 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:

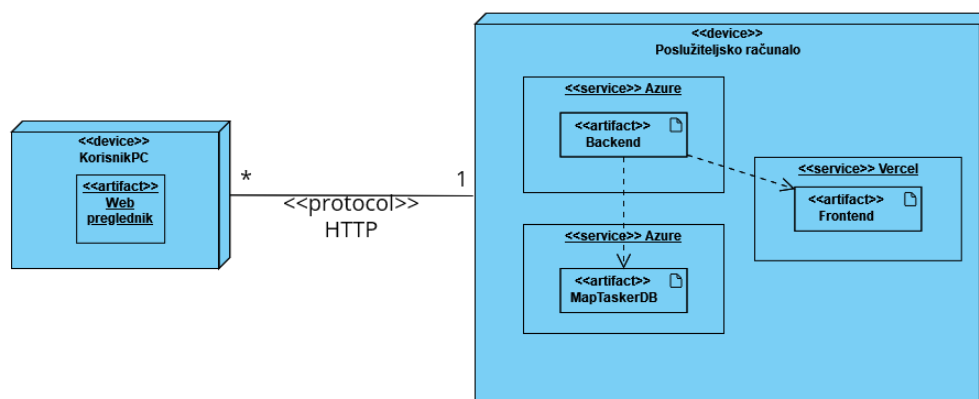
- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

Korisnik se sa svog uređaja spaja na poslužiteljsko računalo HTTP protokolom. Na računalu se nalazi web aplikacija čiji se *backend* i *baza podataka* postavljene na Azure serveru dok se *frontend* nalazi na Vercel serveru.



Slika 5.1: Dijagram razmještaja

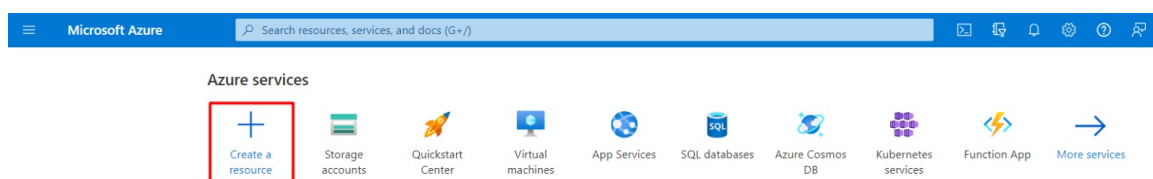
5.4 Upute za puštanje u pogon

Prijava u Azure račun

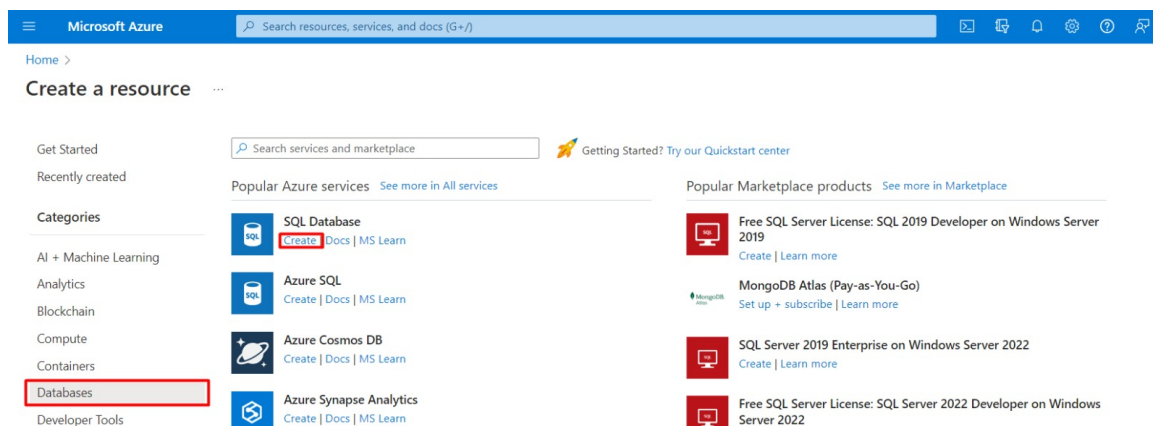
Korisnik se treba prijaviti u svoj Microsoft Azure račun.

Stvaranje baze podataka na Azure serveru

Nakon prijave, korisnik treba u izborniku odabrati opciju "Create a resource", te od ponuđenih odabrati "SQL Database".



Slika 5.2: Odabir opcije "Create a resource"



Slika 5.3: Odabir opcije "SQL Database"

Korisnik treba dati ime bazi, napraviti njezin resource group i server. Također potrebno je odabrati i pricing tier, odnosno cjenovni rang koji ovisi o željenim performansama.

Microsoft Azure

Search resources, services, and docs (G+/)

Home > Create a resource >

Create SQL Database

Microsoft

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure subscription 1

Resource group * ⓘ Maptasker

Create new

Database details

Enter required settings for this database, including picking a logical server and configuring the compute and storage resources

Database name * Enter database name

Server * ⓘ maptasker-db-server (West Europe)

Create new

Want to use SQL elastic pool? ⓘ ☐ Yes ☒ No

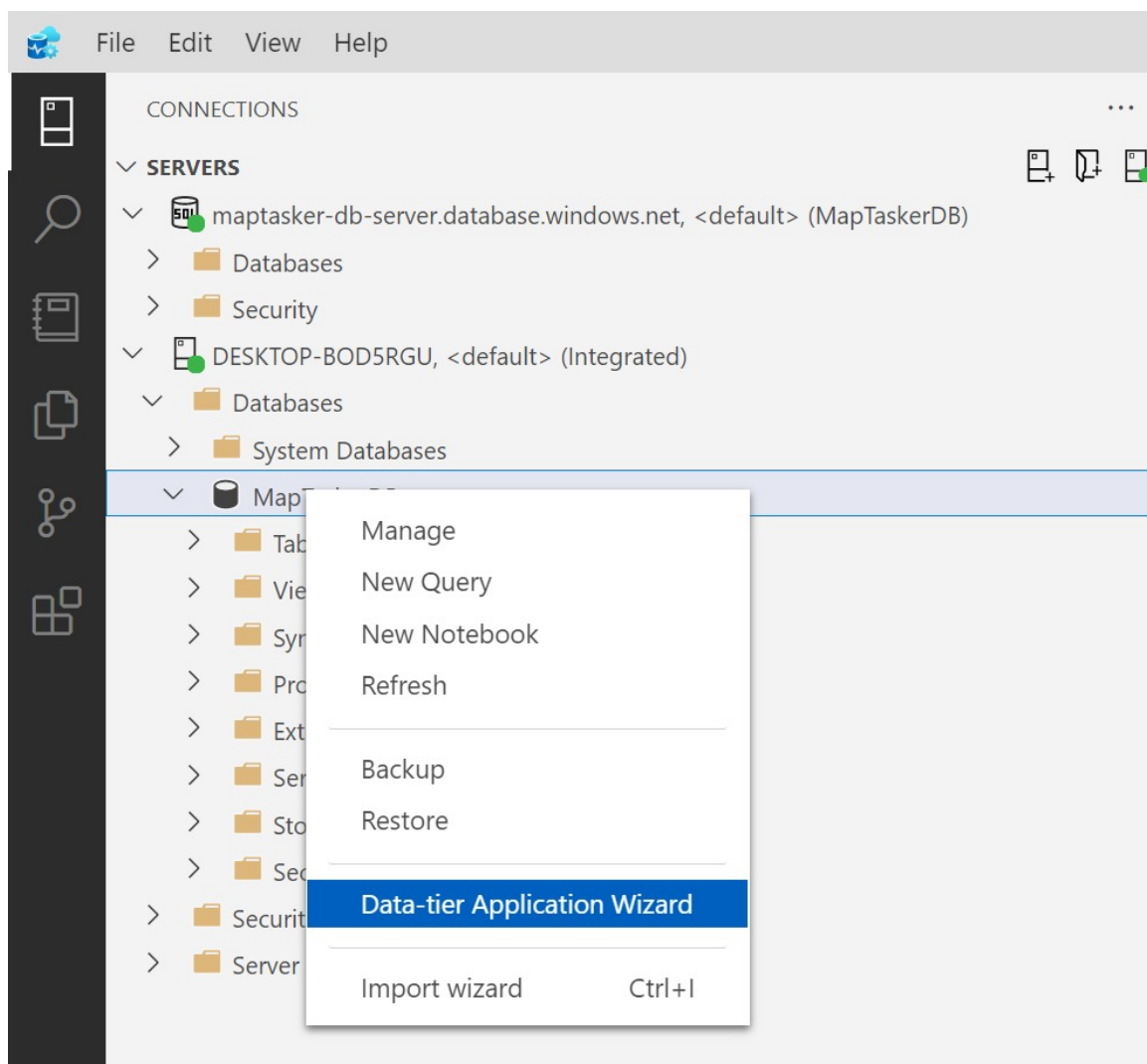
Workload environment ☐ Development ☒ Production

Review + create

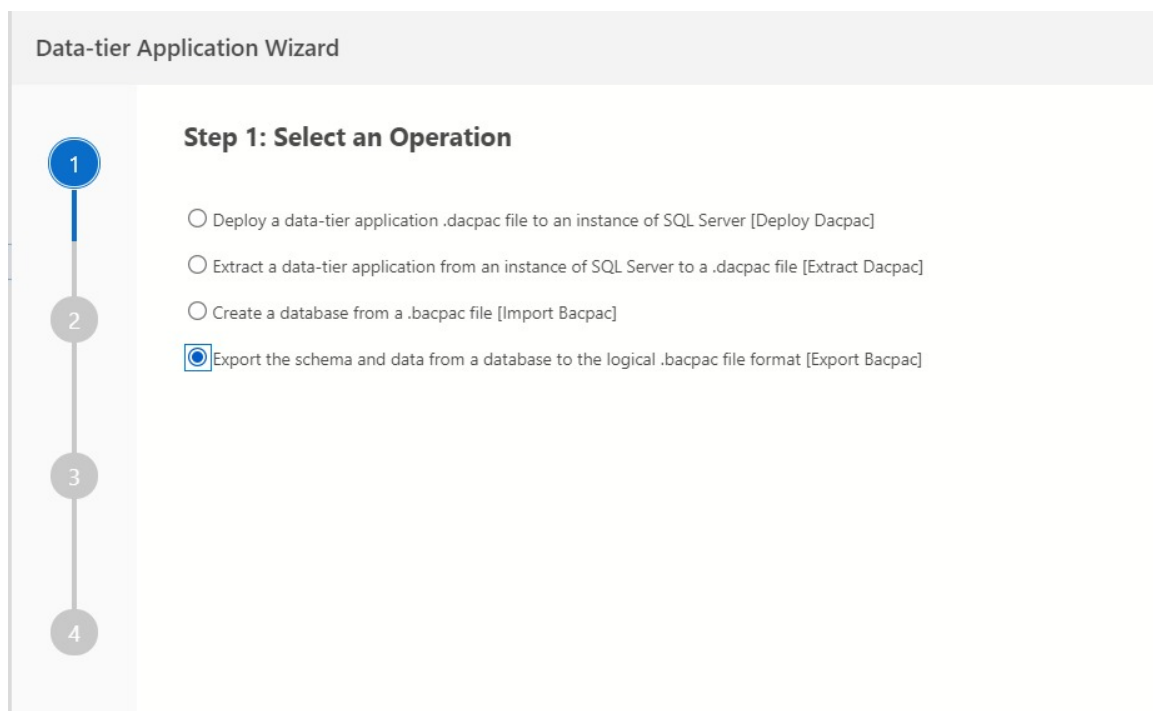
Next : Networking >

Slika 5.4: Dodavanje željenih svojstava bazi

Potrebno je otvoriti Azure Data Studio i spojiti se na lokalni server, zatim desnim klikom na lokalnu bazu na padajućem izborniku odabrati opciju "Data-tier Application Wizard". Nakon toga korisnik treba označiti zadnju navedenu operaciju ("Export the schema..."), kliknuti Next, te na kraju odabrati lokaciju kreirane .bac-pac datoteke i pričekati dok se datoteka ne stvori.

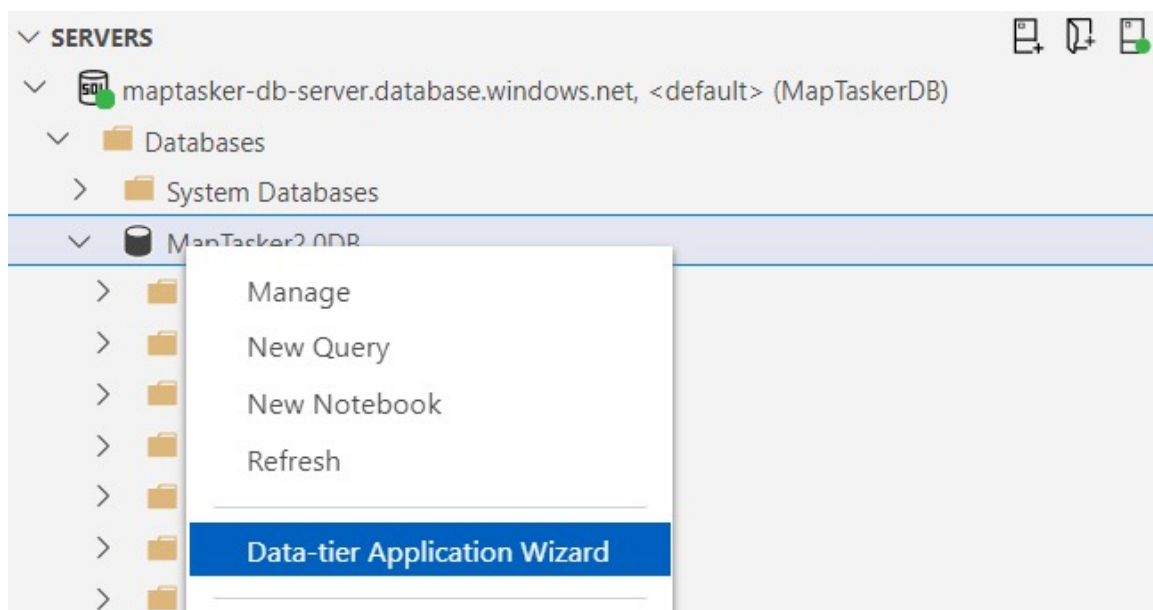


Slika 5.5: Odabir opcije "Data-tier Application Wizard"

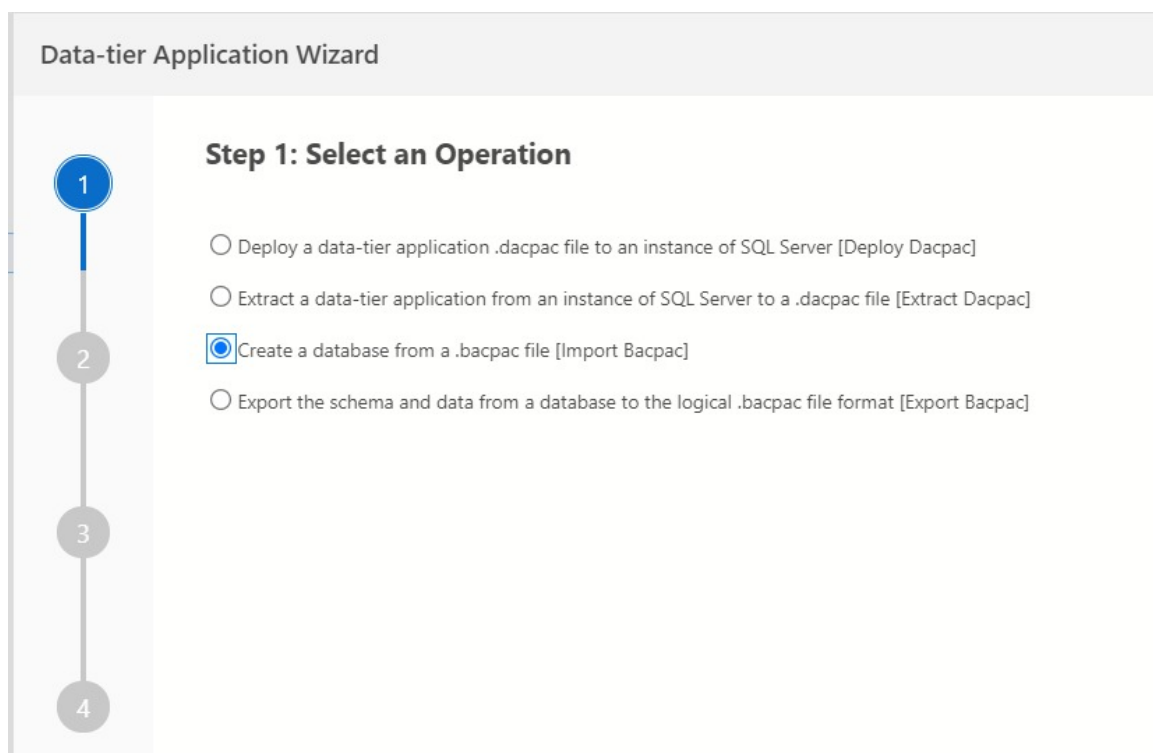


Slika 5.6: Odabir željene operacije

Na kraju je potrebno spojiti se na Azure server i ponovno otvoriti "Data-tier Application Wizard". Zatim korisnik treba odabrati opciju "Create a database" kojom će stvoriti bazu, pratiti daljne korake i odabrati prethodno stvoreni .bacpac file.



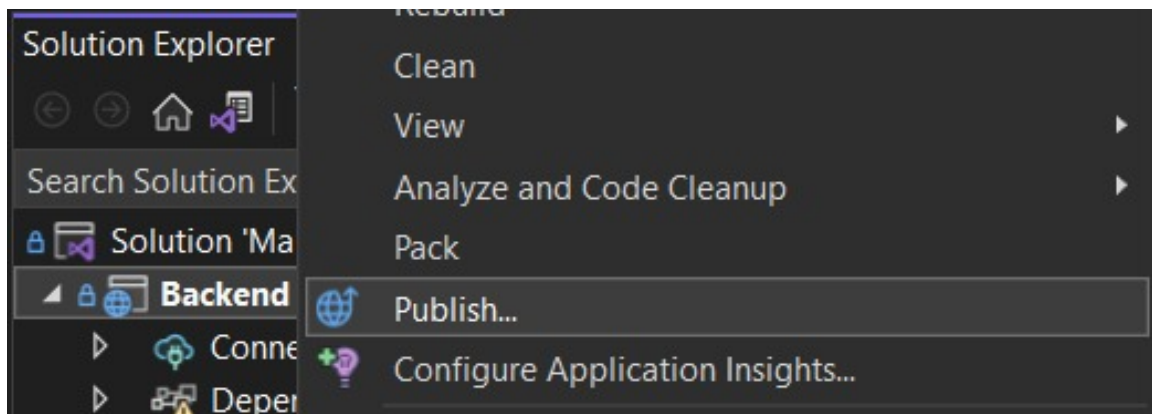
Slika 5.7: Odabir opcije "Data-tier Application Wizard"



Slika 5.8: Odabir operacije stvaranja baze

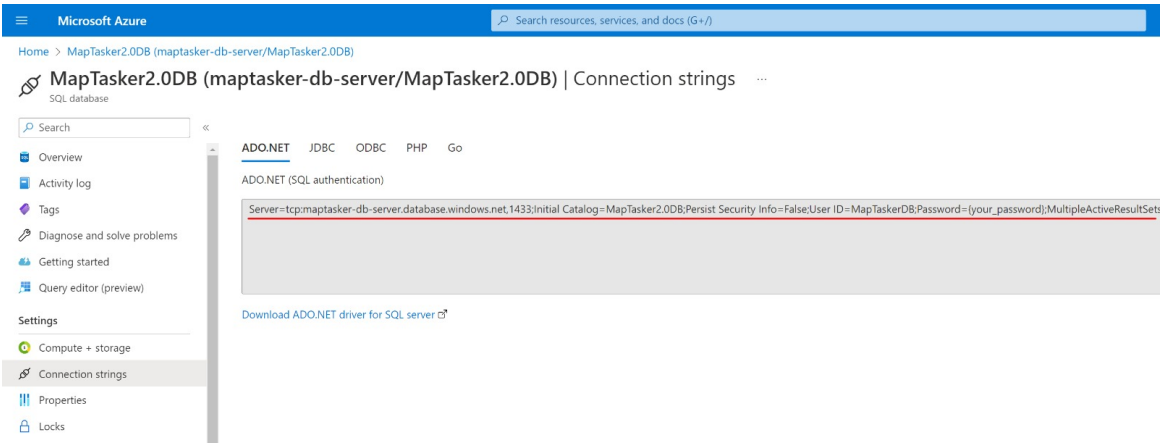
Stvaranje backend API-ja na Azure serveru

Za početak, potrebno je otvoriti projekt u Visual Studiu, i desnim klikom na Backend projekt u padajućem izborniku odabrati opciju "Publish". Zatim se preko grafičkog sučelja ulogirati u Azure i označiti isti resource group kao i za bazu.

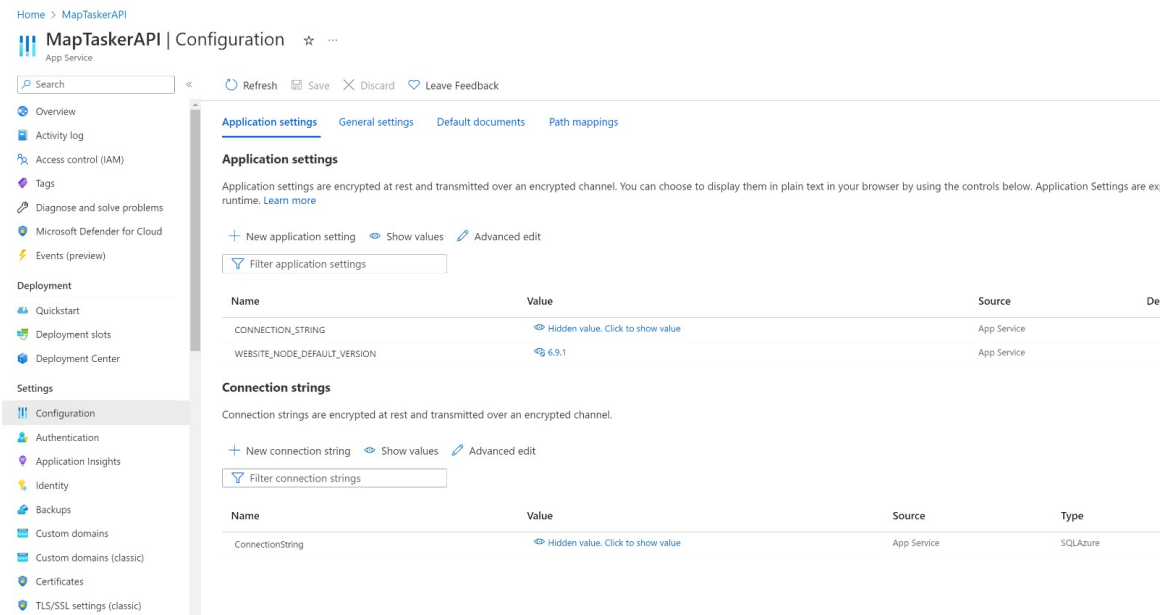


Slika 5.9: Odabir opcije "Publish"

Za ispravno pokretanje backenda, potrebno je otići u Azure na novonastali App Service u Configuration sučelje. Potrebno je dodati novi Application Setting naziva CONNECTION_STRING i vrijednosti connection stringa dobivenog iz Azure postavka za bazu. Ponoviti postupak umetanja connection stringa za podizbornik Connection String, ovoga puta s nazivom ConnectionString.



Slika 5.10: Lokacija vrijednosti connection stringa



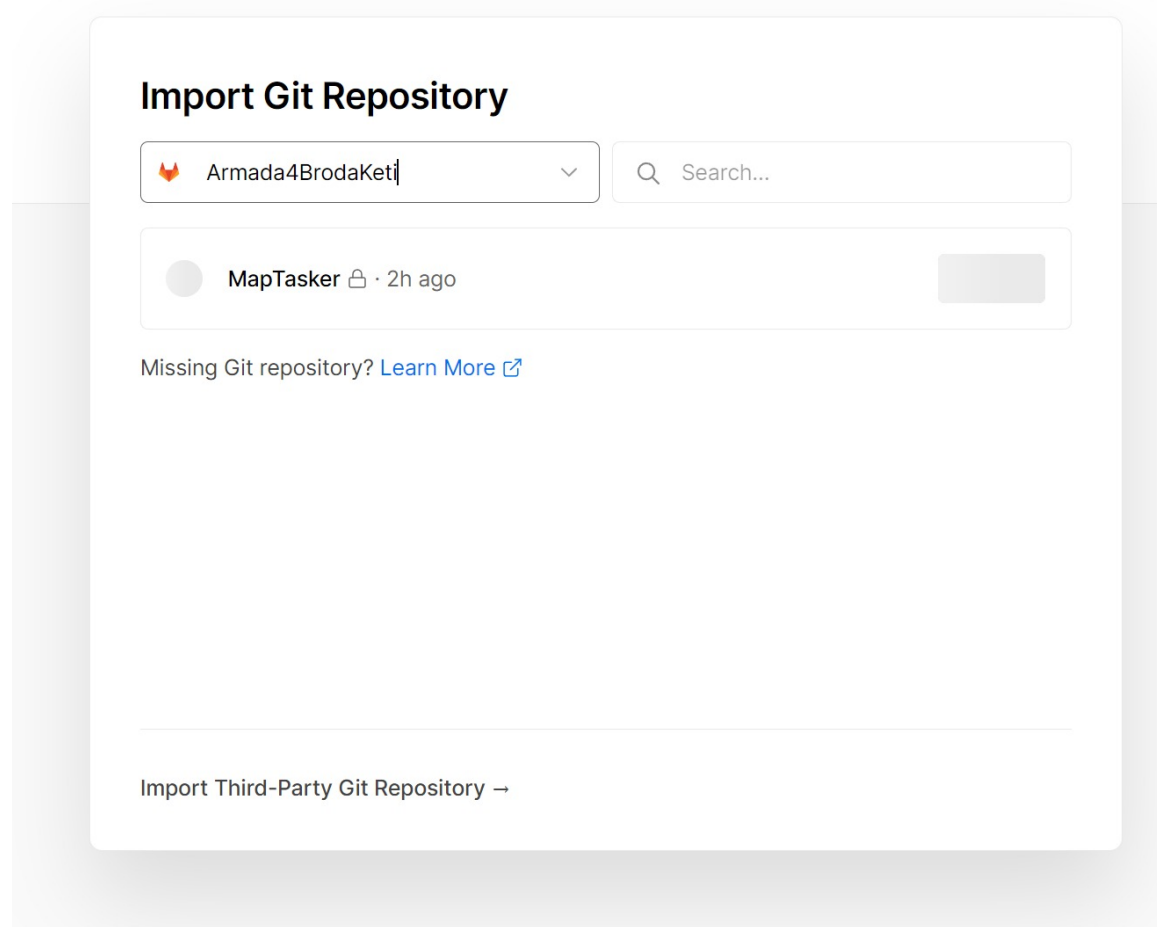
Slika 5.11: Unos vrijednosti connection stringa

Stvaranje frontenda aplikacije na Vercel platformi

Za početak nužno je uvesti git repozitorij projekta.

Let's build something new.

To deploy a new Project, import an existing Git Repository or get started with one of our Templates.



Slika 5.12: Uvoz git repozitorija projekta

Nakon toga potrebno je otići u "Build & Development Settings" sučelje i postaviti naredbe potrebne za pokretanje npm skripti koje su prikazane na slici 5.12.

Build & Development Settings

When using a framework for a new project, it will be automatically detected. As a result, several project settings are automatically configured to achieve the best result. You can override them below.

Framework Preset

Create React App

Build Command ?	react-scripts build	Override <input checked="" type="checkbox"/>
Output Directory ?	build	Override <input type="checkbox"/>
Install Command ?	npm install	Override <input checked="" type="checkbox"/>
Development Command ?	react-scripts start-prod	Override <input checked="" type="checkbox"/>

[Learn more about Build and Development Settings](#) [Save](#)

Slika 5.13: Naredbe za pokretanje npm skripti

U sučelju "Environment Variables" kopirati link na backend API (vidljiv u Azureu) pod ključem REACT_APP_API_URL.

Environment Variables

In order to provide your Deployment with Environment Variables at Build and Runtime, you may enter them right here, for the Environment of your choice. [Learn more](#)

A new Deployment is required for your changes to take effect.

[Create new](#) [Link Shared Environment Variables](#)

Key

e.g. CLIENT_KEY

+ Add another

Value

⊖

Environment

☒ Production

☒ Preview [Select custom branch](#)

☒ Development

⬇ Import

TIP: Paste a .env above to populate the form

Save

☒ Automatically expose [System Environment Variables](#)

🔍 Search...

All Environments

☰ Last Updated

<>

REACT_APP_API_URL
Development, Preview, Production

🗑 [https://maptaskerapi.azurewebsi...](#)

Updated 53d ago

🟣 ⋮

Slika 5.14: Podešavanje varijabli okruženja

Spremanjem postavki razmještanje frontenda će se dogoditi automatski i ponovno se pokrenuti pushanjem na main branch.

6. Zaključak i budući rad

Naš zadatak bio je razvoj web aplikacije koja služi za pronalaženje nestalih osoba te implementacija karte kako bi se olakšao posao spasiočima. Za samu izradnju karte bili su odgovorni kartografi dok su voditelji bili zaduženi za održavanje operacija te određivanja regija koje je potrebno kartografirati. Zadani cilj smo postigli nakon 13 tjedana timskog rada.

Prvi dio projekta se sastojao od okupljanja i upoznavanja članova tima. Na tom sastanku članovi su se podijelili i međusobno dogovorili tko je zadužen za koji dio projekta. Tijekom ove faze većina se članova koncentrirala na pisanje i obradu dokumentacije, što je bilo učinjeno bez većih poteškoća. Na kasnijim sastancima raspravljalo se o tome kako bi *baza podataka*, *frontend* i *backend* trebali izgledati. Najveći dio tijekom ove faze projekta su bili deployment te implementacija slike tijekom registracije korisnika.

Drugi dio projekta se manje oslanjao na dokumentaciju te više na samu implementaciju *use-caseova*. Pošto je već nekoliko članova imalo iskustva s *backendom*, lako su svoje znanje prenijeli i na ostale članove kojima je zadaća bila programiranje i implementiranje funkcija web aplikacije. Takve upute su nam pomogle da se svi uskladimo kako ne bi kasnije došlo do komplikacija tijekom pisanja koda. Kod dokumentacije je još bilo potrebno dovršiti ostale UML dijagrame, ispitivanje programskog rješenja te upute za puštanje u pogon. Sa *frontendom* je bilo manje problema nego s *backendom*, s time da je najteži dio implementacije ove faze projekta bila sama karta kojom se služe kartografi, voditelji i spasioči.

Za komunikaciju među članovima služili smo se aplikacijom *WhatsApp* gdje smo često postavljali pitanja i pomagali drugima oko dokumentacije ili manjih problema u vezi projekta. Za komunikaciju tijekom rješavanja većih problema projekta kao što su bili karta ili implementacija kompleksnijih funkcionalnosti koristili smo aplikaciju *Discord*. Budući rad bi mogao uključivati mobilno proširenje kako bi se povećala pristupačnost i pomoć kod pronalaženja nestalih osoba.

Izrada ovog projektnog zadatka dobro je doprinijela razvoju vještina članovima tima koji se prvi put upoznavaju s ovakvim zadatcima dok su već vješti članovi iskoristili ovu priliku kako bi usavršili svoje sposobnosti te pomogli drugima tako

što su im prenijeli svoje znanje o razvoju web aplikacija.

Indeks slika i dijagrama

2.1	Prijava nestale osobe preko ICMP	7
2.2	Popunjavanje informacije korisnika	7
3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranih i registriranih korisnika aplikacije	17
3.2	Dijagram obrasca uporabe, funkcionalnost spasioca, voditelja operacije i kartografa	18
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	18
3.4	Sekvencijski dijagram za UC4 - Prijava nestale osobe	19
3.5	Sekvencijski dijagram za UC10 - Stvaranje operacije	21
3.6	Sekvencijski dijagram za UC11 - Mijenjanje statusa bloka	23
3.7	Sekvencijski dijagram za UC20 - Potvrda registracije	25
4.1	Arhitektura sustava	27
4.2	MVC model	28
4.3	ER dijagram baze podataka	34
4.4	Dijagram razreda 1 - Register i Login	35
4.5	Dijagram razreda 2 - Missing Report i Comment	36
4.6	Dijagram razreda 3 - Operations, Areas, Regions, Blocks, Buildings	37
4.7	Dijagram razreda 4 - User i Role	37
4.8	Dijagram razreda 5 - Statistics	38
4.9	Dijagram stanja - Voditelj operacije	39
4.10	Dijagram aktivnosti registracije, prijave korisnika te prijava nestale osobe	39
5.1	Dijagram razmještaja	43
5.2	Odabir opcije "Create a resource"	44
5.3	Odabir opcije "SQL Database"	44
5.4	Dodavanje željenih svojstava bazi	45
5.5	Odabir opcije "Data-tier Application Wizard"	46
5.6	Odabir željene operacije	47

5.7	Odabir opcije "Data-tier Application Wizard"	48
5.8	Odabir operacije stvaranja baze	48
5.9	Odabir opcije "Publish"	49
5.10	Lokacija vrijednosti connection stringa	50
5.11	Unos vrijednosti connection stringa	50
5.12	Uvoz git repozitorija projekta	51
5.13	Naredbe za pokretanje npm skripti	52
5.14	Podešavanje varijabli okruženja	53

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 20. listopada 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - sastanak s asistentom i demonstratorom
 - analiziranje zadatka
 - razrješavanje nejasnoća i osnovnih dilema funkcionalnosti

2. sastanak

- Datum: u ovom formatu: 21. listopada 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - podjela dužnosti i zadataka za naredni tjedan
 - konačan izbor alata i tehnologija

3. sastanak

- Datum: 01. studenoga 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - izrada baze podataka
 - daljnja podjela rada (raspisivanje dokumentacije i izrada sekvencijskih i dijagrama obrazaca uporabe)

4. sastanak

- Datum: 03. studenoga 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - drugi sastanak s asistentom i demonstratorom
 - komentiranje napisane dokumentacije i dijagrama
 - daljnja podjela rada (front end i back end)

5. sastanak

- Datum: 14. studenoga 2022.
- Prisustvovali: Bojan Puvača, Jurica Uglešić, Antonio Mišić, Matija Jelavić
- Teme sastanka:
 - izrada dijagrama razreda

Tablica aktivnosti

	Bojan Puvača	Matija Jelavić	Danijel Kovačević	Antonio Mišić	Katarina Šabić	Jurica Uglešić	Ema Vlainić
Upravljanje projektom	5						
Opis projektnog zadatka						2	
Funkcionalni zahtjevi	2				1		
Opis pojedinih obrazaca			4		3		
Dijagram obrazaca		6					
Sekvencijski dijagrami							6
Opis ostalih zahtjeva						1	
Arhitektura i dizajn sustava				2			
Baza podataka			2				
Dijagram razreda		2	2	2		2	
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja					3		
Zaključak i budući rad							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Bojan Puvača	Matija Jelavić	Danijel Kovačević	Antonio Mišić	Katarina Šabić	Jurica Uglešić	Ema Vlanić
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
Izrada početne stranice							6
Izrada baze podataka	1	1	1	1	1	1	1
Spajanje s bazom podataka	6						
Back end	1	3		10		2	
Registracija, validacija, prijava korisnika u sustav	6				6		
Puštanje aplikacije u pogon	10						

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.