

Programsko inženjerstvo

Ak. god. 2022./2023.

MapTasker

Dokumentacija, Rev. 1

Grupa: Armada4BrodaKeti

Voditelj: Bojan Puvača

Datum predaje: 13.1.2023.

Nastavnik: Hrvoje Nuić

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	6
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	21
3.2 Ostali zahtjevi	28
4 Arhitektura i dizajn sustava	29
4.1 Baza podataka	30
4.1.1 Opis tablica	31
4.1.2 Dijagram baze podataka	36
4.2 Dijagram razreda	37
4.3 Dijagram stanja	42
4.4 Dijagram aktivnosti	43
4.5 Dijagram komponenti	45
5 Implementacija i korisničko sučelje	46
5.1 Korištene tehnologije i alati	46
5.2 Ispitivanje programskog rješenja	47
5.2.1 Ispitivanje komponenti	47
5.2.2 Ispitivanje sustava	54
5.3 Dijagram razmještaja	59
5.4 Upute za puštanje u pogon	60
6 Zaključak i budući rad	70
Popis literature	72
Indeks slika i dijagonama	74

Dodatak: Prikaz aktivnosti grupe

75

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	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: izmijena 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/dodataka	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 sekvensijska dijagrama	Ema Vlainić	16.11.2022.
1.0	Korigiranje teksta i provjera dokumentacije	Katarina Šabić	18.11.2022.
1.1	Dodan dijagram aktivnosti	* Jurica Uglešić *	07.01.2023.
1.2	Dodani zaključak i dijagram razmještaja	Jurica Uglešić	10.01.2023.
1.3	Dodane upute za puštanje aplikacije u pogon	Katarina Šabić	10.01.2023.
1.4	Dodan dijagram stanja	Katarina Šabić	11.01.2023

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodataka	Autori	Datum
1.5	Dodan dijagram komponenti, korišteni alati i ispitivanje korištenog rješenja	Danijel Kovačević	12.01.2023
1.5.1	Manja revizija (dijagram komponenti)	Danijel Kovačević	13.01.2023.
1.6	Ažurirani dnevnik sastajanja i dnevnik promjena dokumentacije, dodan dijagram pregleda promjena	Katarina Šabić	13.01.2023
2.0	Konačni tekst predloška dokumentacije	Bojan Puvača, Katarina Šabić	13.01.2023.

2. Opis projektnog zadatka

Cilj je razviti programsku potporu za web aplikaciju "MapTasker" koja služi za pronalaženje nestalih osoba tijekom raznim elementarnim 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 voditeljima 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 poligону 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 **provjera**. 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 podatcima. 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 podatcima 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 mjerjenja, 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 website's header with links for Media Center, Inquiry Center, English, and social media icons. Below the header, a navigation bar includes Home, About us, What we do, Where we work, The Missing, News, Resources, Contact, and a search bar. On the left, a sidebar titled 'In this section' lists various links, with 'Report a Missing Person' highlighted by a red box. The main content area is titled 'Report a Missing Person' and contains instructions for reporting missing persons. It features fields for Last name, Translated name(s), First name, and a dropdown for gender (Missing person is male/female). A note indicates that photographs can be uploaded via the OIC MyFace application.

Slika 2.1: Prijava nestale osobe preko ICMP

This screenshot shows the 'Your Information' section of the ICMP form. It includes fields for Last name, Translated name, First name, Relationship with the missing person (with a dropdown menu showing 'Blank'), Please specify the relationship (a text input field), Would you be willing to provide a DNA reference sample? (with a dropdown menu showing 'Blank'), Country of residence (a text input field with placeholder 'Enter a location'), Place of residence (a dropdown menu with placeholder 'First choose country'), Address of residence (a text input field), Phone, Alternative phone, and E-mail, all marked with double asterisks (**).

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 sudsionik:** Voditelj operacije
- **Cilj:** Uklanjanje prijave o nestaloj osobi
- **Sudsionici:** 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 sudsionik:** Voditelj operacije
- **Cilj:** Brisanje komentara sa prijave o nestaloj osobi
- **Sudsionici:** 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 sudsionik:** Korisnik
- **Cilj:** Prikaz područja na karti
- **Sudsionici:** 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 sudsionik:** Voditelj operacije
- **Cilj:** Definiranje područja koje je potrebno pretražiti
- **Sudsionici:** 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 odradjeni posao
 7. Promjena statusa bloka spremaju 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 odradjen
 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 želenog 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

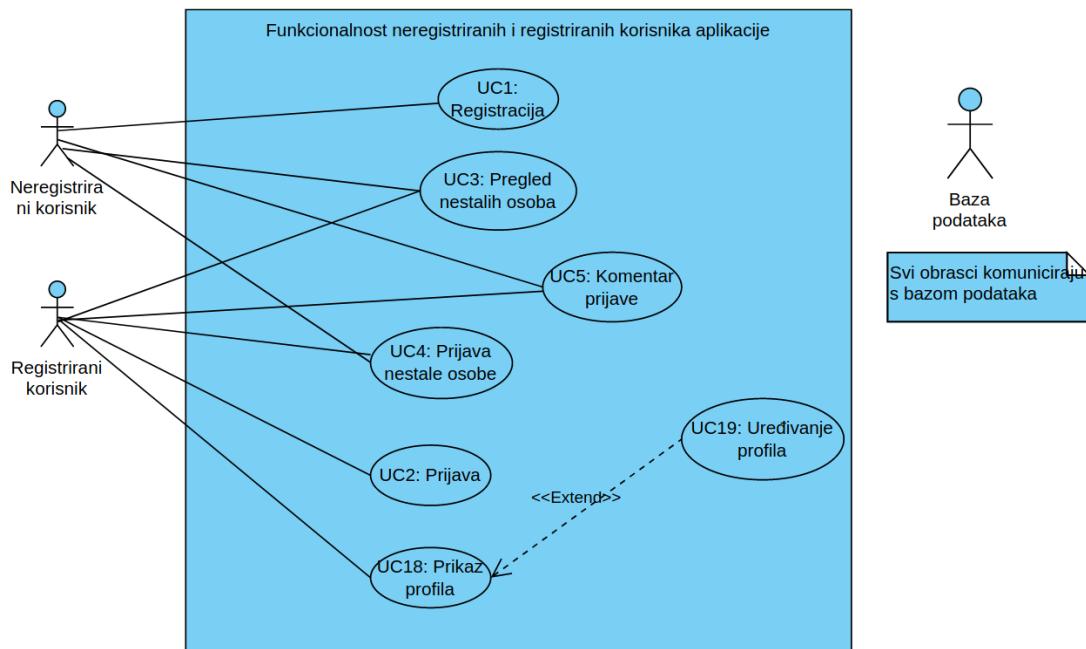
UC19 - Uređivanje profila

- **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

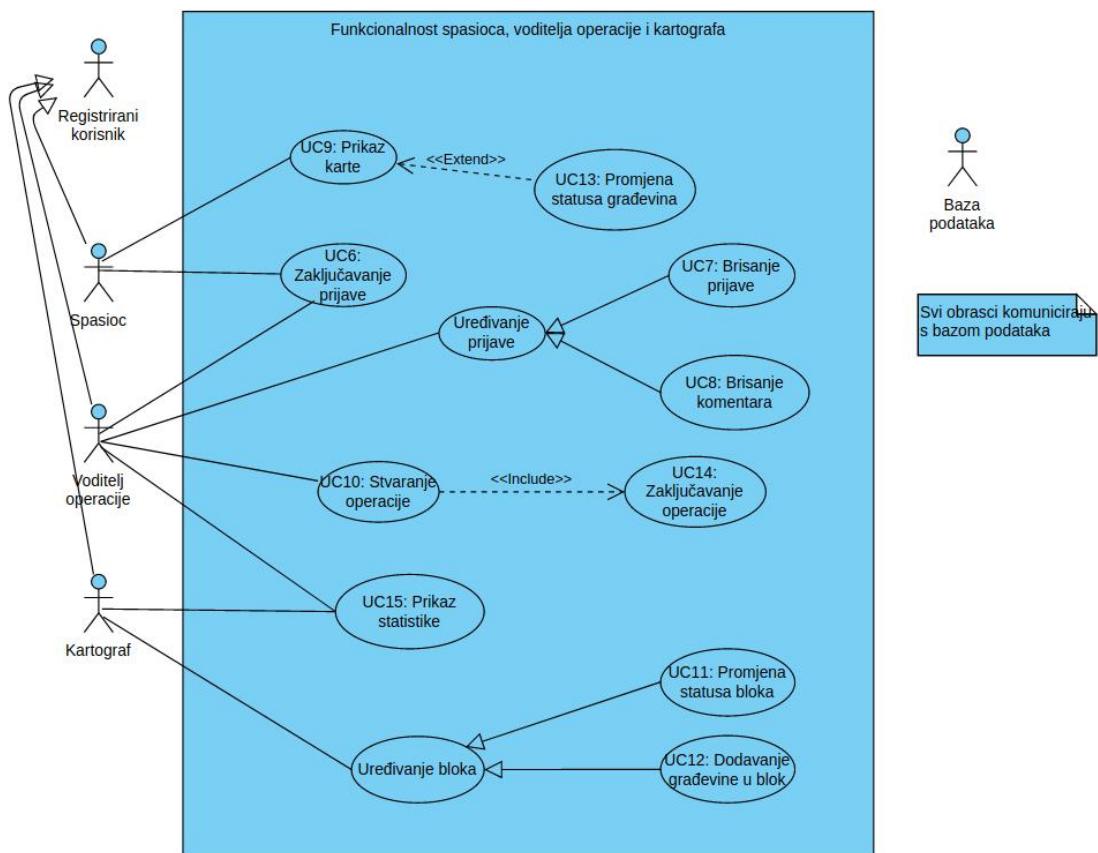
UC20 - Potvrda registracije

- **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

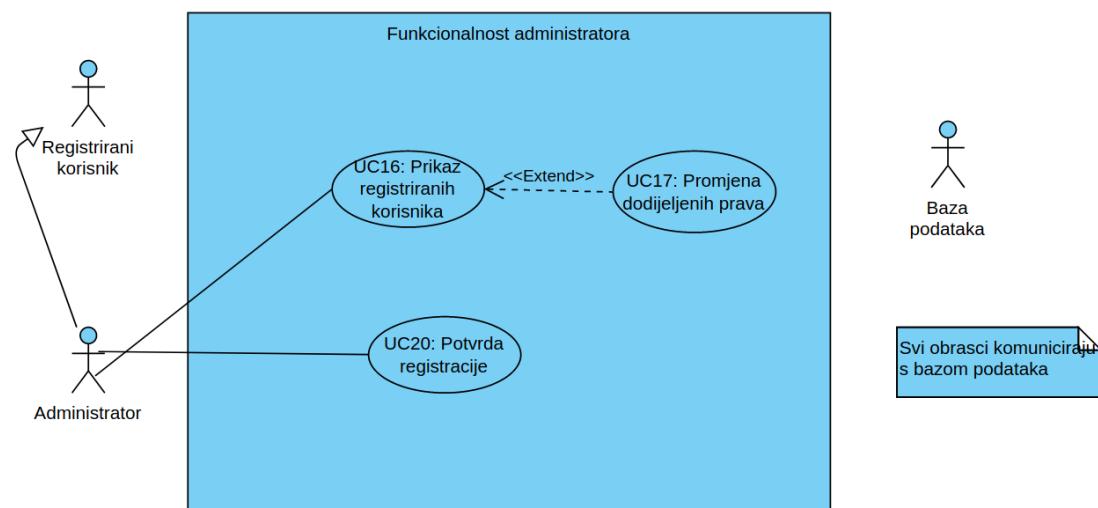
Dijagrami obrazaca uporabe



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



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

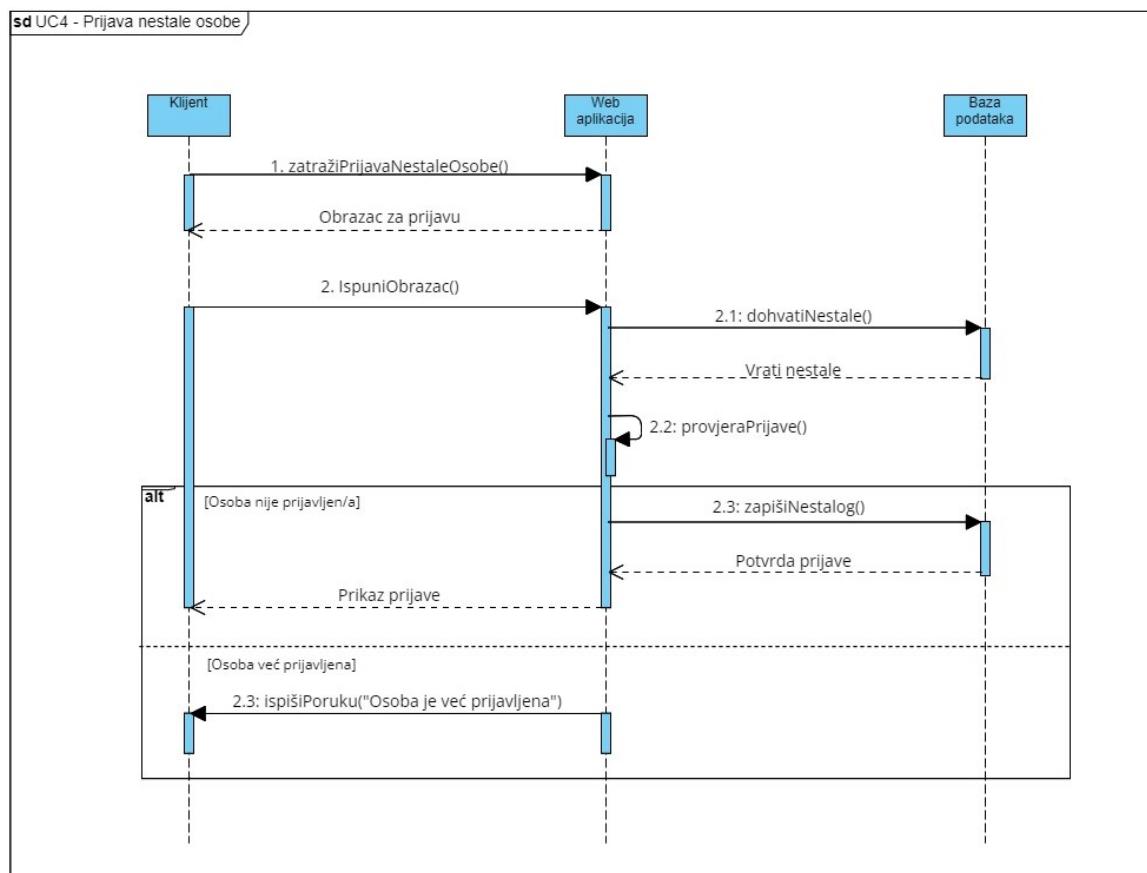


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC4 - Prijava nestale osobe

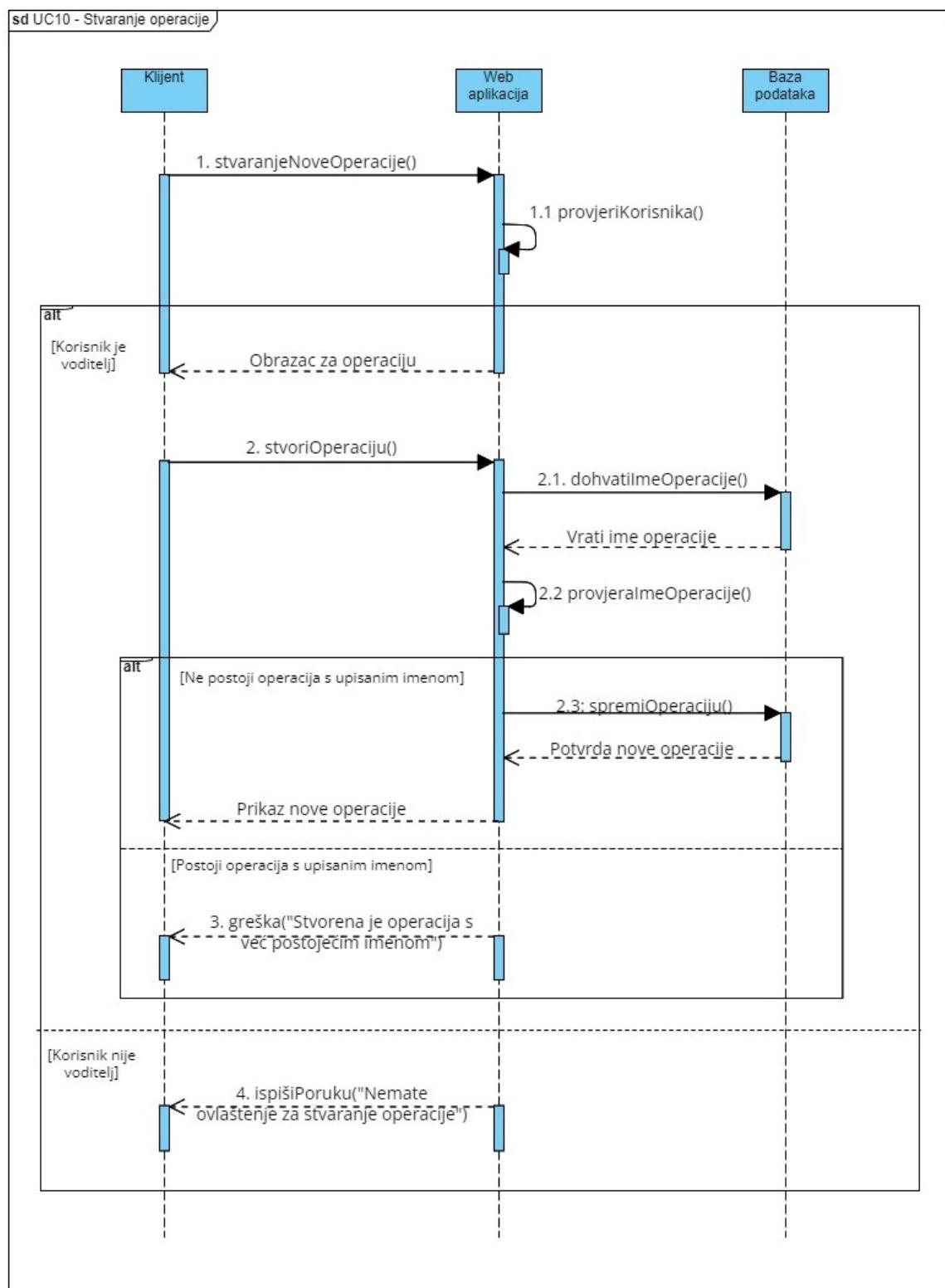
Klijent šalje zahtjev za prijavu nestale osobe. Poslužitelj prikazuje obrazac za prijavu. Klijent ispunjava obrazac (ime, prezime, fotografija i opis) za prijavu nestale osobe. Poslužitelj dohvaća podatke iz baze te provjerava je li nova prijavljena osoba u bazi. Ako osoba dosad nije prijavljena, zapisuje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

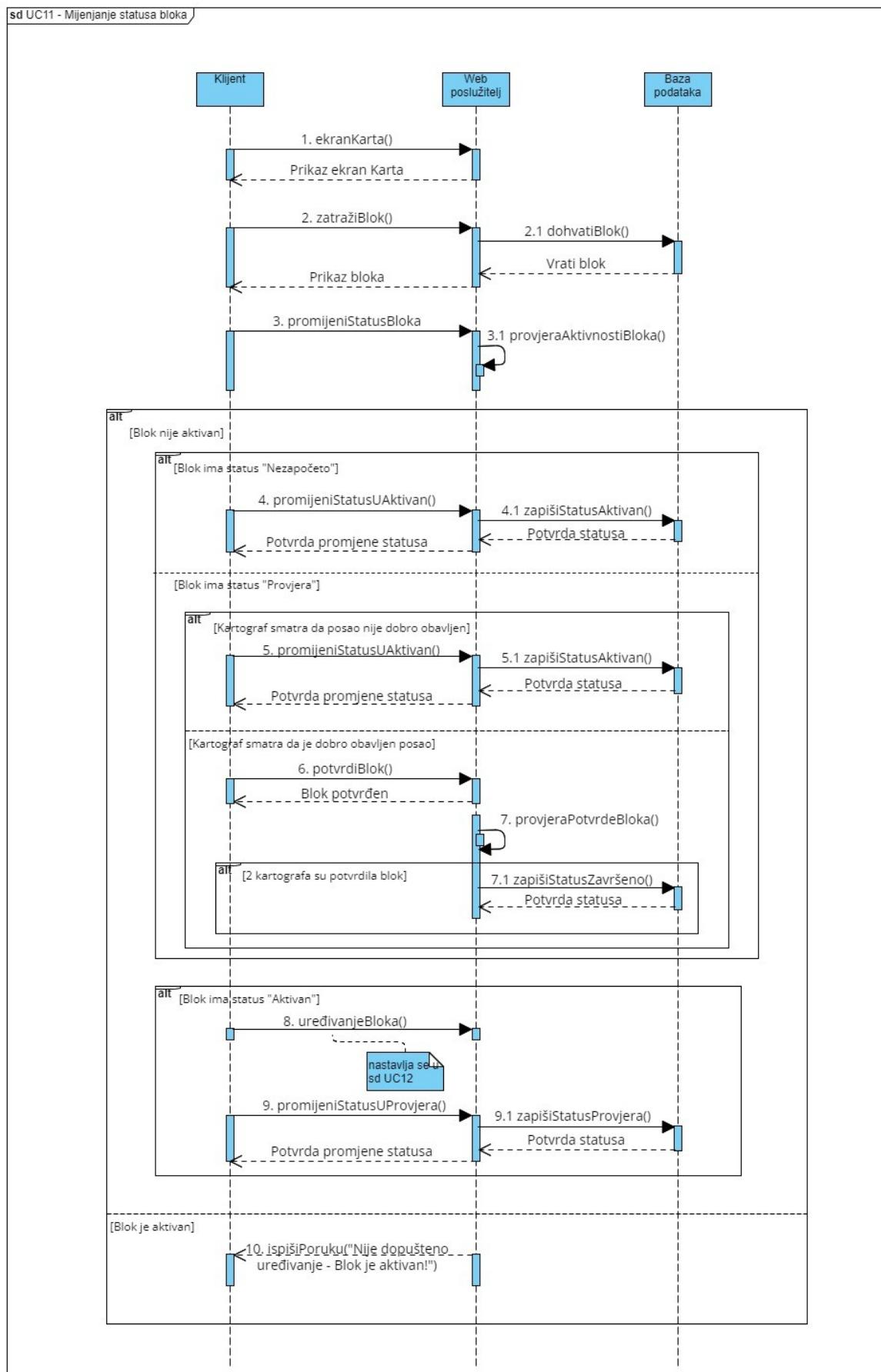
Klijent šalje zahtjev za stvaranje operacije. Sustav provjerava je li korisnik voditelj. Ako korisnik nije voditelj, poslužitelj će ispisati poruku da nema ovlaštenje za stvaranje nove operacije, a ukoliko je, poslat će obrazac za stvaranje nove operacije. Klijent stvara novu operaciju, tj. upisuje ime 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

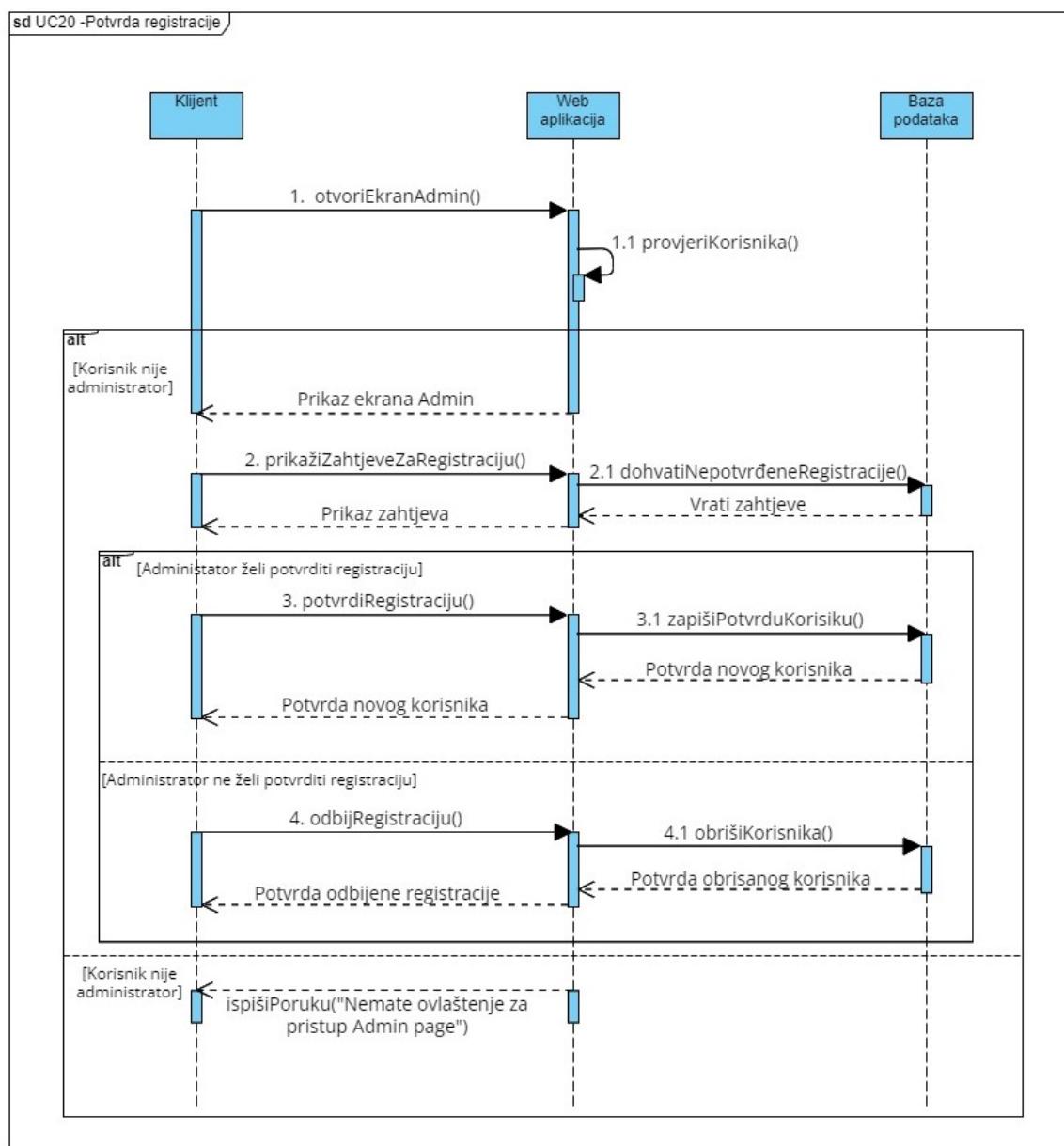
Obrazac uporabe UC11 - Mijenjanje statusa bloka

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 aktiviran 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 aktiviran 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 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

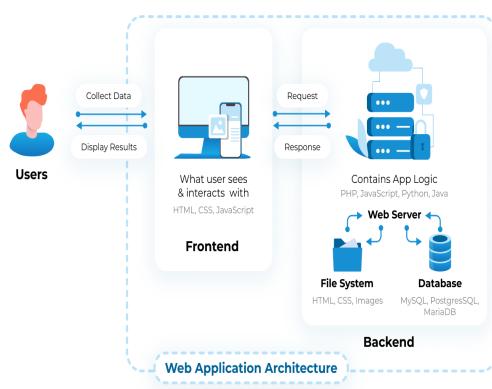
3.2 Ostali zahtjevi

- 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 dijakritičkih 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 odradjuje 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.

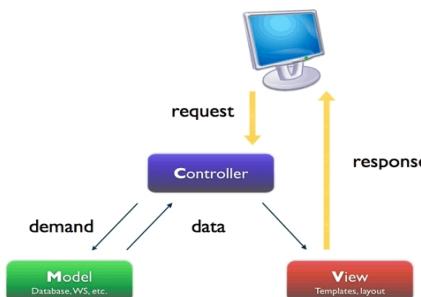
Web aplikacija se najčešće dijeli na dva dijela:

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

Tehnologije koje smo uzeli za razvoj web aplikacije su React.js za Front-end u razvojnog 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 dohvati 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 atribute 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 atribute: 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 atribute 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 atribute: 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 atribute: 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 atribute: 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 atribute: 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

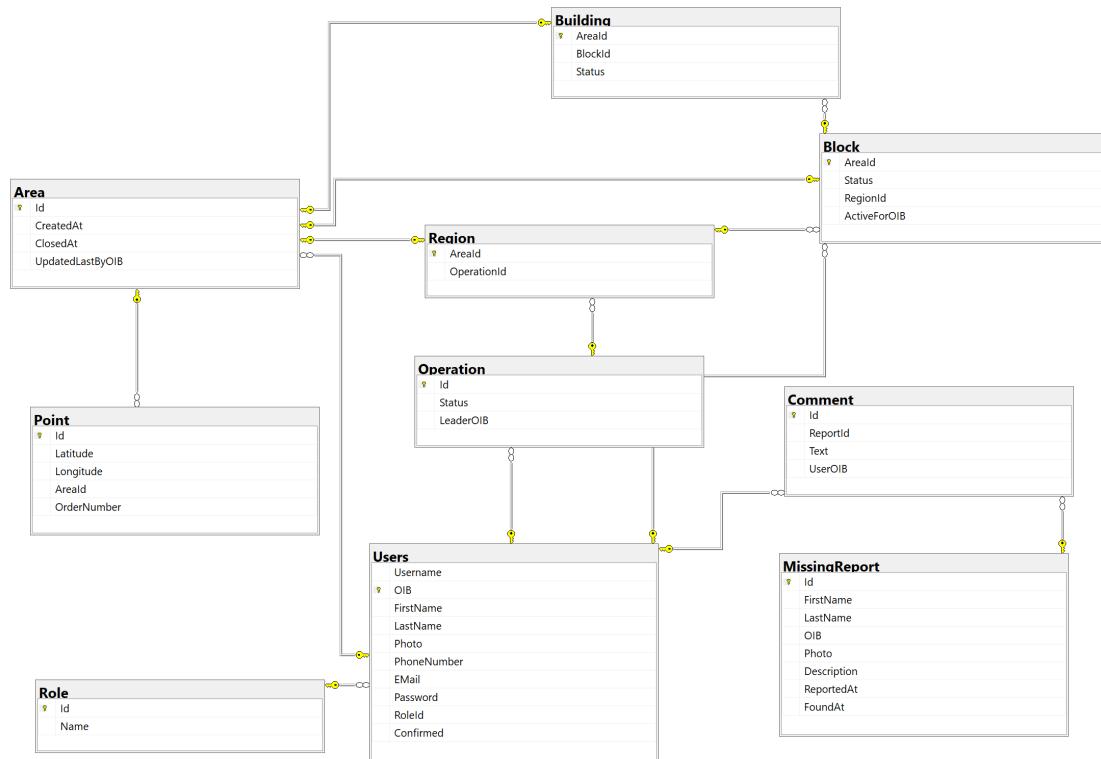
Building Ovaj entitet sadrži sve važne informacije o građevinama. Sadrži atribute: identifikator građevine, identifikator bloka te status građevine. Ovaj entitet u vezi je *Many-to-One* s entitetom Area preko atributa identifikatora građevine te u vezi *Many-To-One* s entitetom Block preko atributa identifikatora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

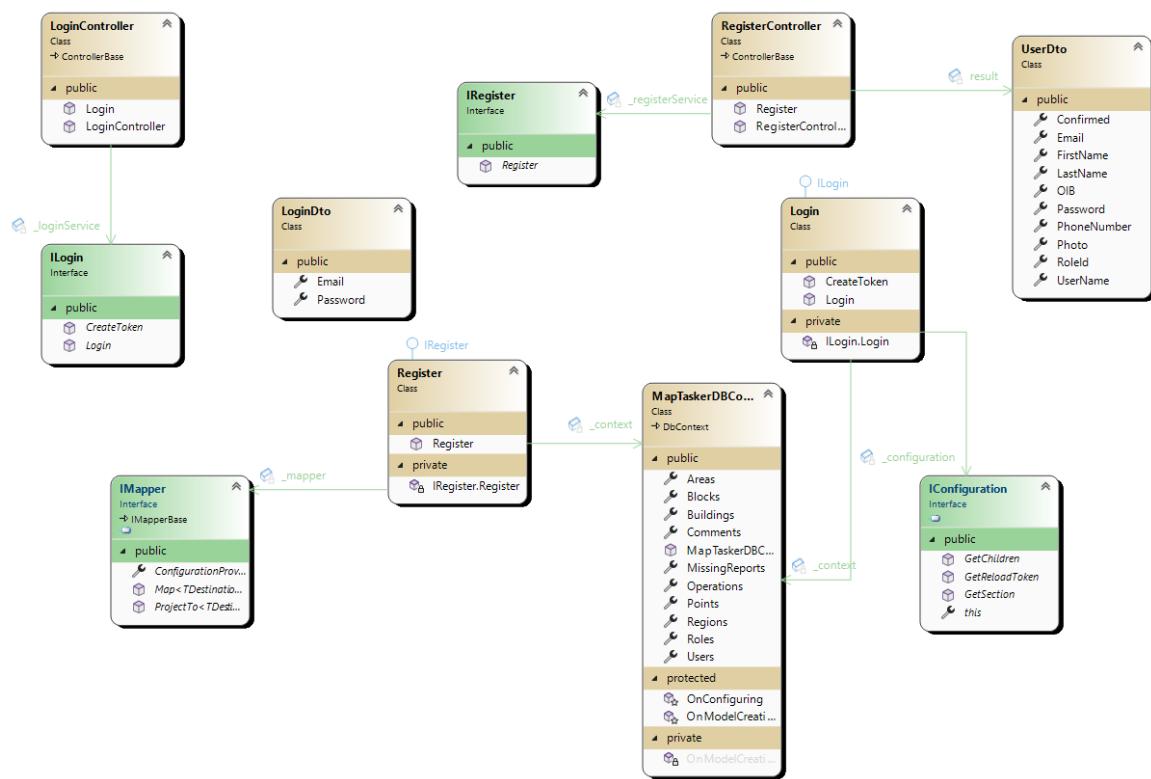


Slika 4.3: ER dijagram baze podataka

4.2 Dijagram razreda

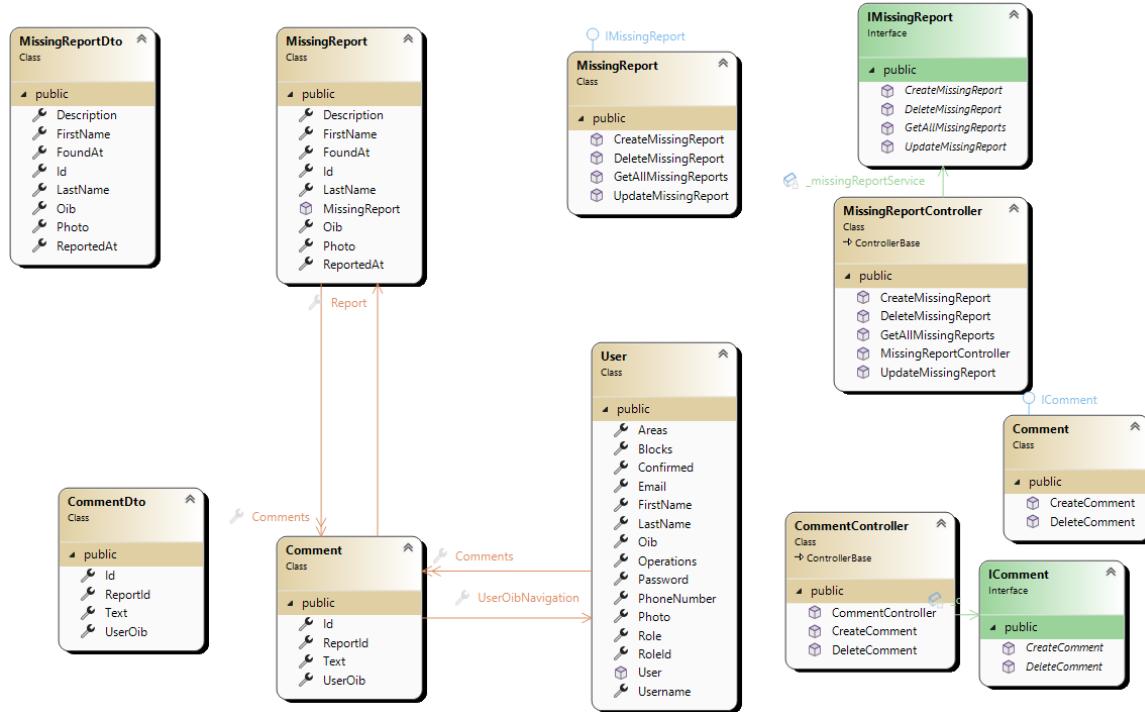
Na slikama 4.4, 4.5, 4.6, 4.7 i 4.8 prikazani su razredi koji pripadaju *backend* dijelu MVC arhitekture. Da bi se izbjegla prenapučenost, razredi su podijeljeni s obzirom na funkcionalnosti. Tako na prvom dijagramu imamo razrede vezane za registraciju i prijavu, na drugom su razredi vezani za prijave i komentare, na trećem se dijagramu nalaze svi razredi potreбni za karte i mapiranje, a na četvrtom razredi vezani za korisnike i njihove uloge, a na petom su dijagramu smješteni razredi vezani za statistiku.

Na slici 4.4 nalazi se dijagram koji prikazuje odnos Controllera, Modela i DTO-ova vezanih uz Login i Register. Vidimo da se, nakon registracije korisnika, podaci spremaju u bazu podataka, a također se iz RegisterControllera podaci šalju u UserDto. RegisterController i LoginController pomoću registerServicea i loginService-a realiziraju sučelja IRegister i ILogin te nasljeđuju njihove metode.



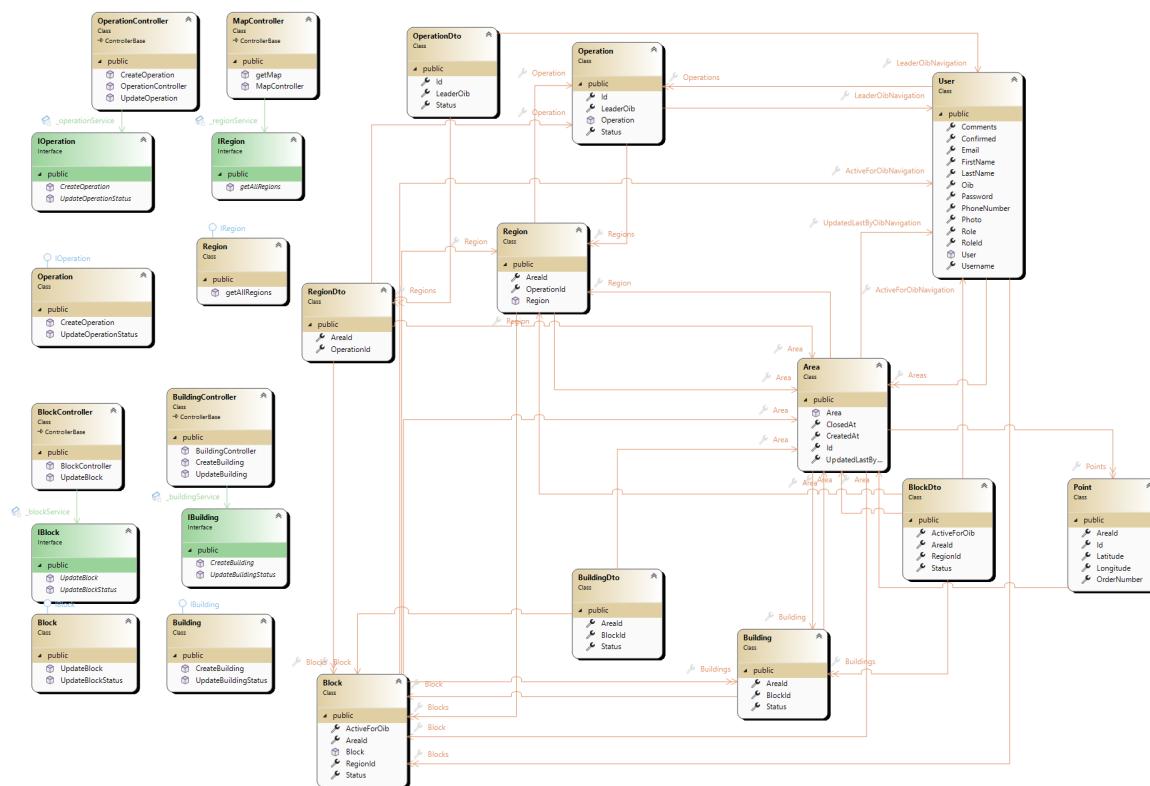
Slika 4.4: Dijagram razreda 1 - Register i Login

Ovaj dijagram prikazuje prijave nestalih i komentare. Promatrajući Controllere, koji nasljeđuju sučelja, vidimo koje sve metode postoje, a to su metode za stvaranje, brisanje, ažuriranje i dohvata svih podataka, te isto tako za stvaranje i brisanje komentara. One manipuliraju s MissingReportDto i CommentDto. Modeli MissingReport i Comment primaju ulazne podatke od Controllera te su međusobno povezani relacijama pridruživanja.



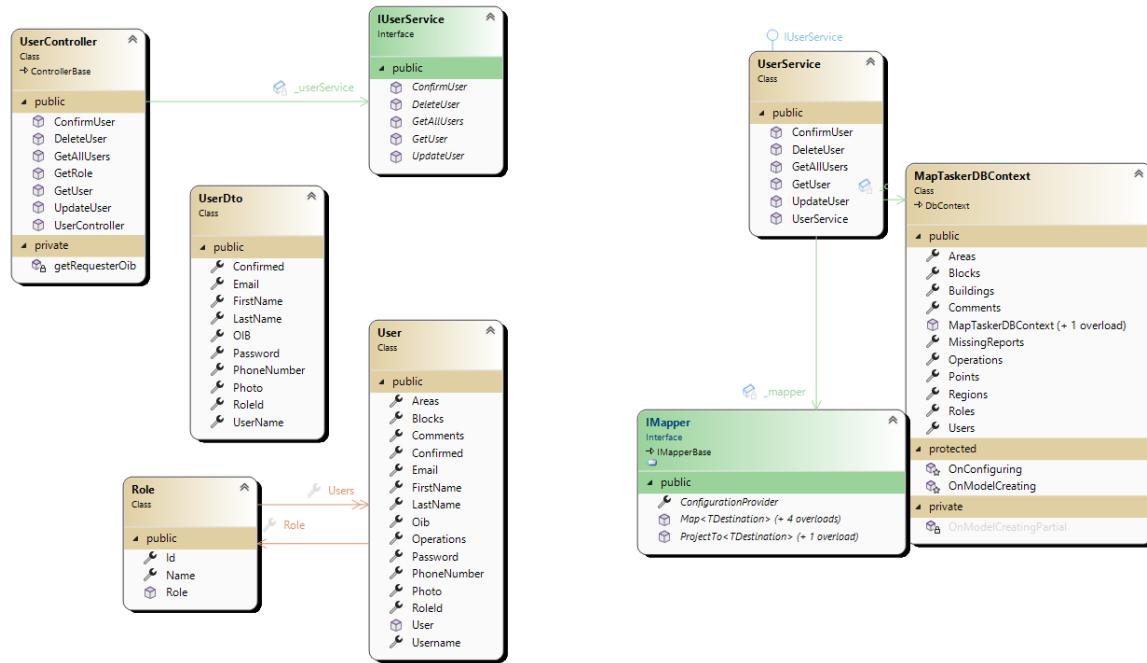
Slika 4.5: Dijagram razreda 2 - Missing Report i Comment

Ovaj dijagram prikazuje razrede vezane za mapu, operacije i područja te kao takav čini ključni dio aplikacije. Iz ovog dijagrama razvidna je hijerarhija svih područja. Controlleri OperationController, MapController, BlockController i BuildingController nasljeđuju metode za stvaranje operacija i područja, kao i za mijenjanje njihova statusa. Modeli operacije, regije, blokovi i zgrade međusobno su povezani relacijama generalizacija i pridruživanje. Modeli i DTO-ovi također su povezani s Userom relacijama pridruživanja te su prikazane operacije, područja i blokovi povezani s tim Userom.



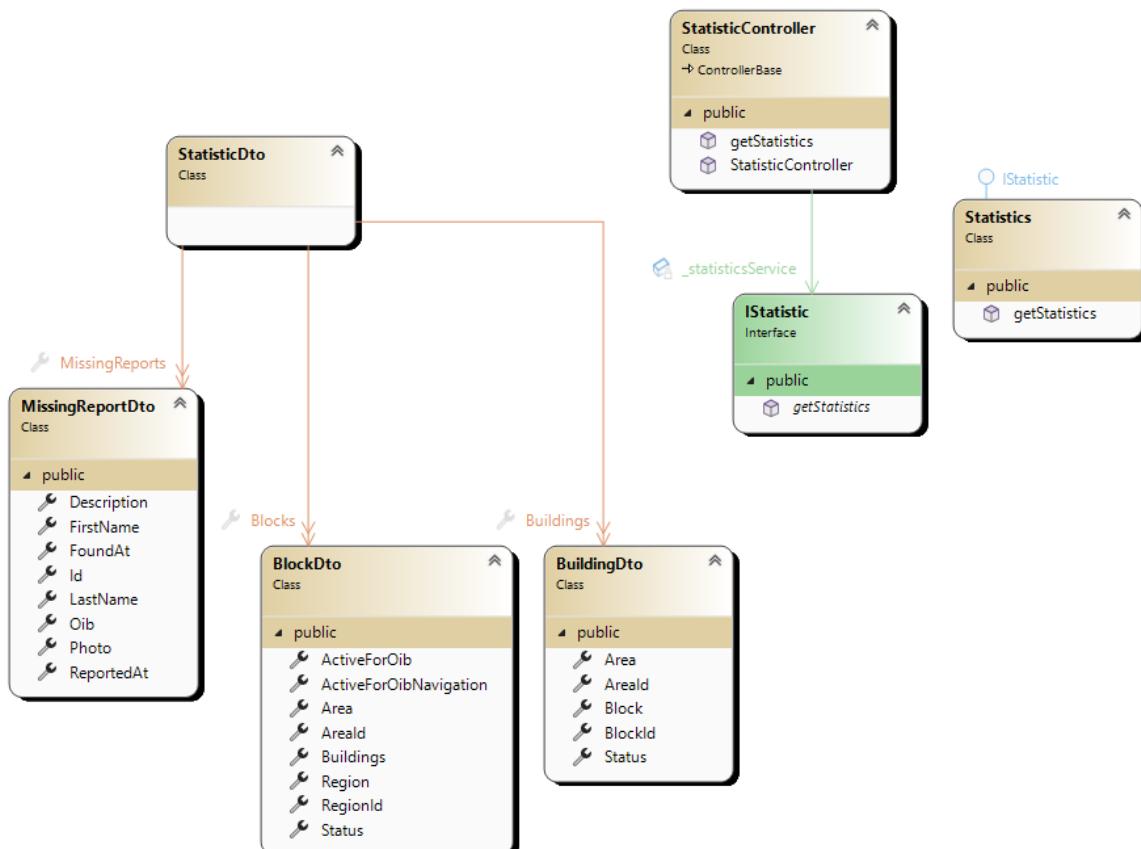
Slika 4.6: Dijagram razreda 3 - Operations, Areas, Regions, Blocks, Buildings

Na slici 4.7 prikazani su Controlleri, Modeli i Servicei User i Role. UserController implementira IServiceUser te implementira metode za potvrdu, brisanje, dohvata, ažuriranje te dohvata svih korisnika. Modeli User i Role također su međusobno povezani pridruživanjem. Vidimo i da se podaci o Useru spremaju u bazu podataka.



Slika 4.7: Dijagram razreda 4 - User i Role

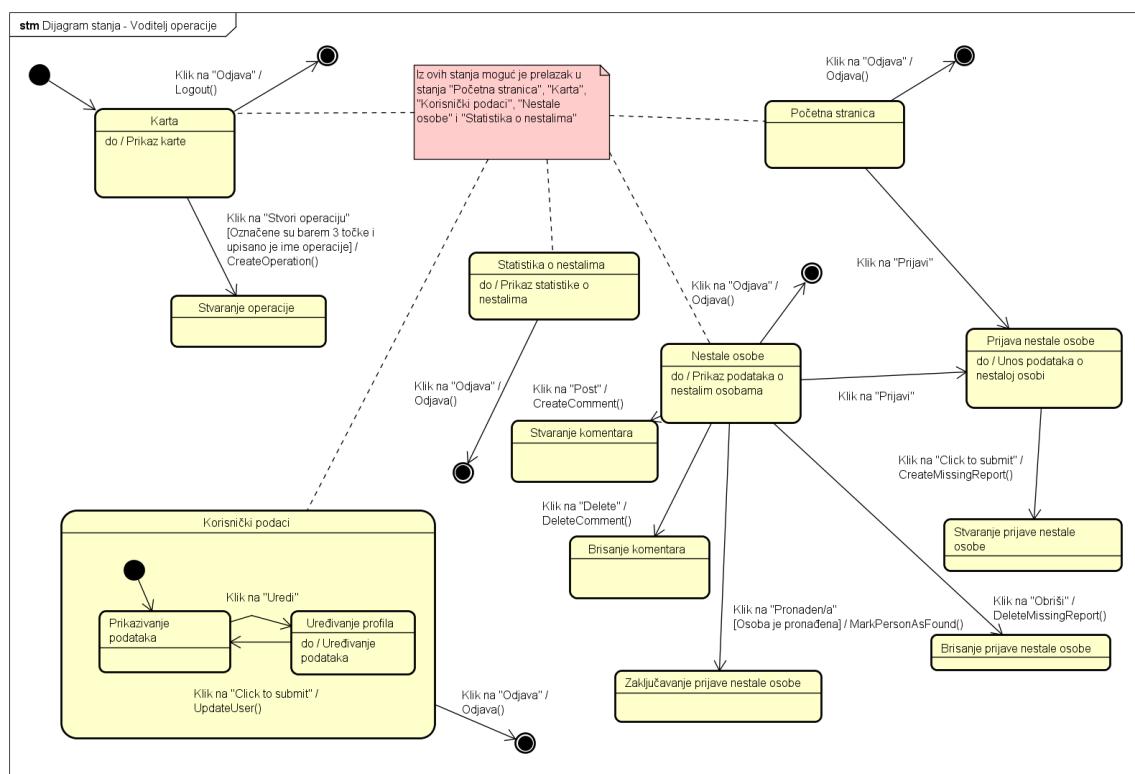
Na slici 4.8 nalazi se dijagram koji prikazuje statistiku. StatisticsController realizira sučelje IStatistic te njegovu metodu getStatistics za dohvat statistike. Data transfer object StatisticDto povezan je s DTO-ovima MissingReport, Block i Building kako bi mogao prikazati potrebne podatke.



Slika 4.8: Dijagram razreda 5 - Statistics

4.3 Dijagram stanja

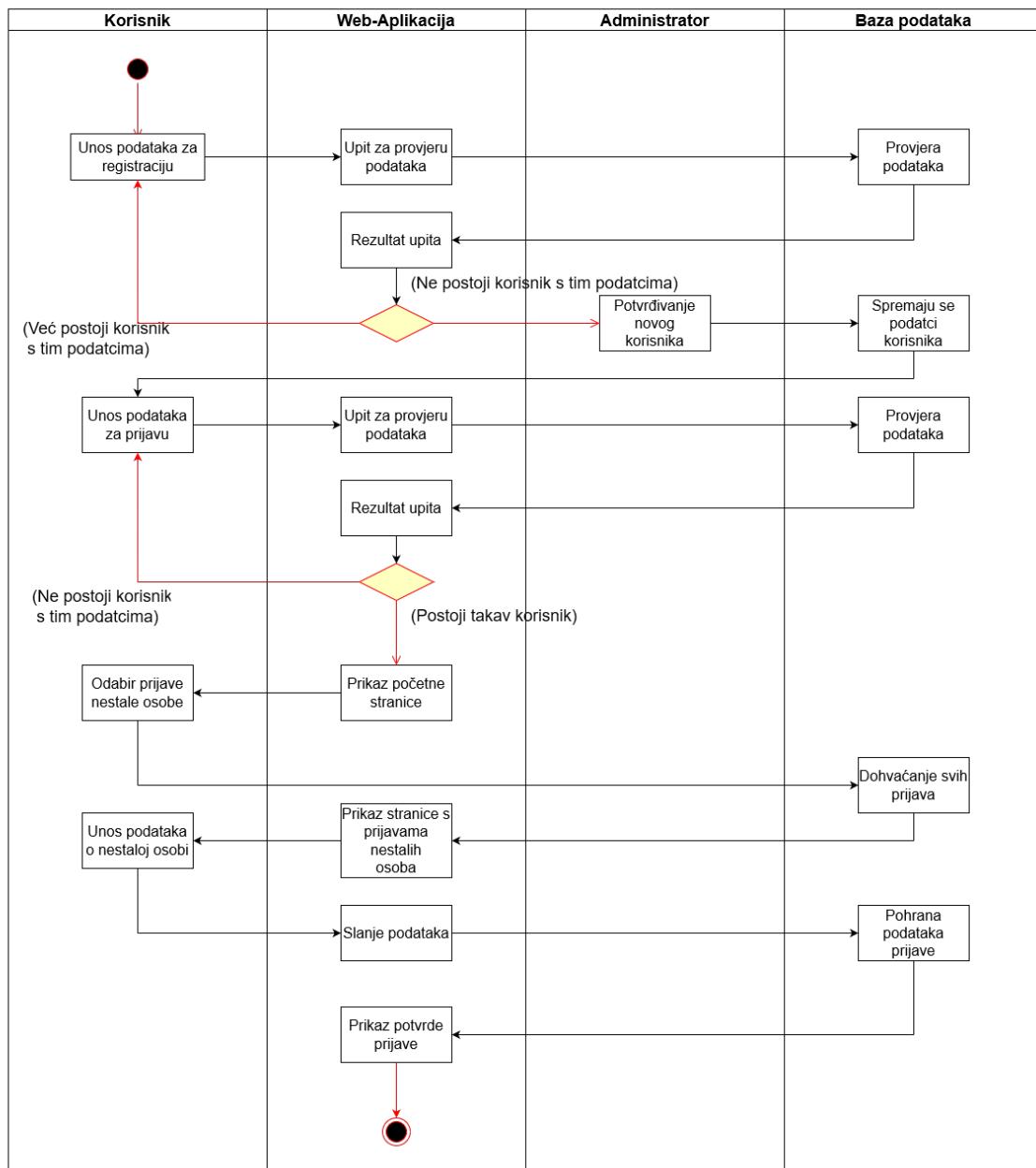
Dijagram stanja koristi se kako bi se opisalo dinamičko ponašanje nekog dijela sustava u vremenu. Njime se prikazuju prijelazi između stanja pojedinih objekata. Korisnik prilikom registracije unosi svoje podatke i bira ulogu za koju se prijavljuje. Na slici 4.9 prikazan je dijagram stanja za registriranog 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 izmejniti. Također, voditelj osim što može prijaviti nestalu osobu, ima uvid u popis nestalih osoba i može zaključati 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 4.9: Dijagram stanja - Voditelj operacije

4.4 Dijagram aktivnosti

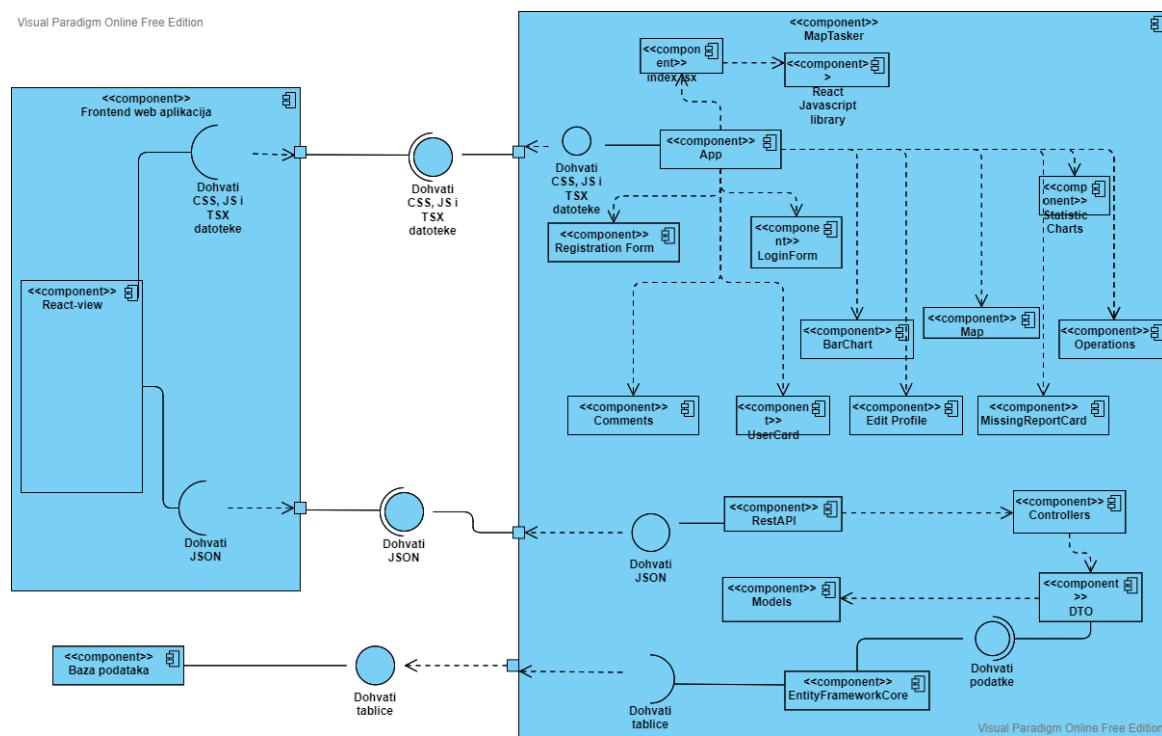
Dijagram aktivnosti opisuje proces registracije korisnika koju administrator potvrđuje. Nakon što se uspješno provede prijava odobrenog korisnika, korisnik ide na stranicu gdje može objaviti prijavu nestale osobe nakon što su uneseni svi potrebni podatci.



Slika 4.10: Dijagram aktivnosti registracije, prijave korisnika te prijava nestale osobe

4.5 Dijagram komponenti

Ovaj dijagram pokazuje organizaciju i međuodnos komponenti, unutarnje strukturu te odnos prema okolini. Postoje 3 vrste sučelja: za dohvat CSS, JS i TSX datoteka, za dohvat JSON podataka kojim se pristupa REST API komponenti te za dohvat tablica iz baze podataka. Preko glavne komponente App pristupa se ostalim komponentama. Preko drugog sučelja za dohvat JSON-a pristupa se Rest API-ju koji zatim poslužuje podatke koji pripadaju *backend* dijelu aplikacije. Tablice se iz baze podataka dohvaćaju preko sučelja za dohvat podataka iz baze te se šalju u obliku DTO-a MVC arhitekturi na daljnju manipulaciju.



Slika 4.11: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacija WhatsApp¹ i Discord². Za izradu UML dijagrama korišten je alat Visual Paradigm Online³. Kao sustav za upravljanje izvornim kodom korišten je Git⁴ te je udaljeni repozitorij projekta dostupan na platformi GitLab⁵. Za izradu dokumentacije korišten je softverski sustav za pripremu dokumenata LaTeX⁶.

Kao razvojno okruženje korišteno je integrirano razvojno okruženje Microsofta - Microsoft Visual Studio⁷. Čini ga opsežan skup alata za izgradnju ASP.NET aplikacija, desktop aplikacija i mobilnih aplikacija. Kao urednik izvornog koda za rad na *frontendu* korišten je i Microsoftov uređivač izvornog koda - Visual Studio Code⁸.

Za izradu *frontend* dijela aplikacije korišten je .NET Framework⁹ te C sharp¹⁰, dok je *backend* dio aplikacije napravljen koristeći React¹¹ i JavaScript¹². .Net Framework jest radni okvir razvijen od strane Microsofta koji programerima uvelike pomaže u prevođenju i izvođenju programa. React je besplatna JavaScript biblioteka otvorenog koda za izgradnju korisničkih sučelja.

Baza podataka nalazi se na poslužitelju u oblaku Microsoft Azureu¹³. To je računalna platforma koja omogućuje pristup i upravljanje aplikacijama i uslugama.

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://online.visual-paradigm.com/>

⁴<https://git-scm.com/>

⁵<https://gitlab.com/>

⁶<https://www.latex-project.org/>

⁷<https://visualstudio.microsoft.com/>

⁸<https://code.visualstudio.com/>

⁹<https://dotnet.microsoft.com>

¹⁰<https://docs.microsoft.com/en-us/dotnet/csharp/>

¹¹<https://reactjs.org/>

¹²<https://www.javascript.com/>

¹³<https://portal.azure.com/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

Svi unit testovi izvršeni su automatski. Ispitivanje je provedeno po serviceima te su se tako provjeravale pojedine komponente sustava. U svakom su testu na početku dohvaćeni DbContext i Mapper.

Ispitni slučaj 1: Dohvaćanje svih korisnika

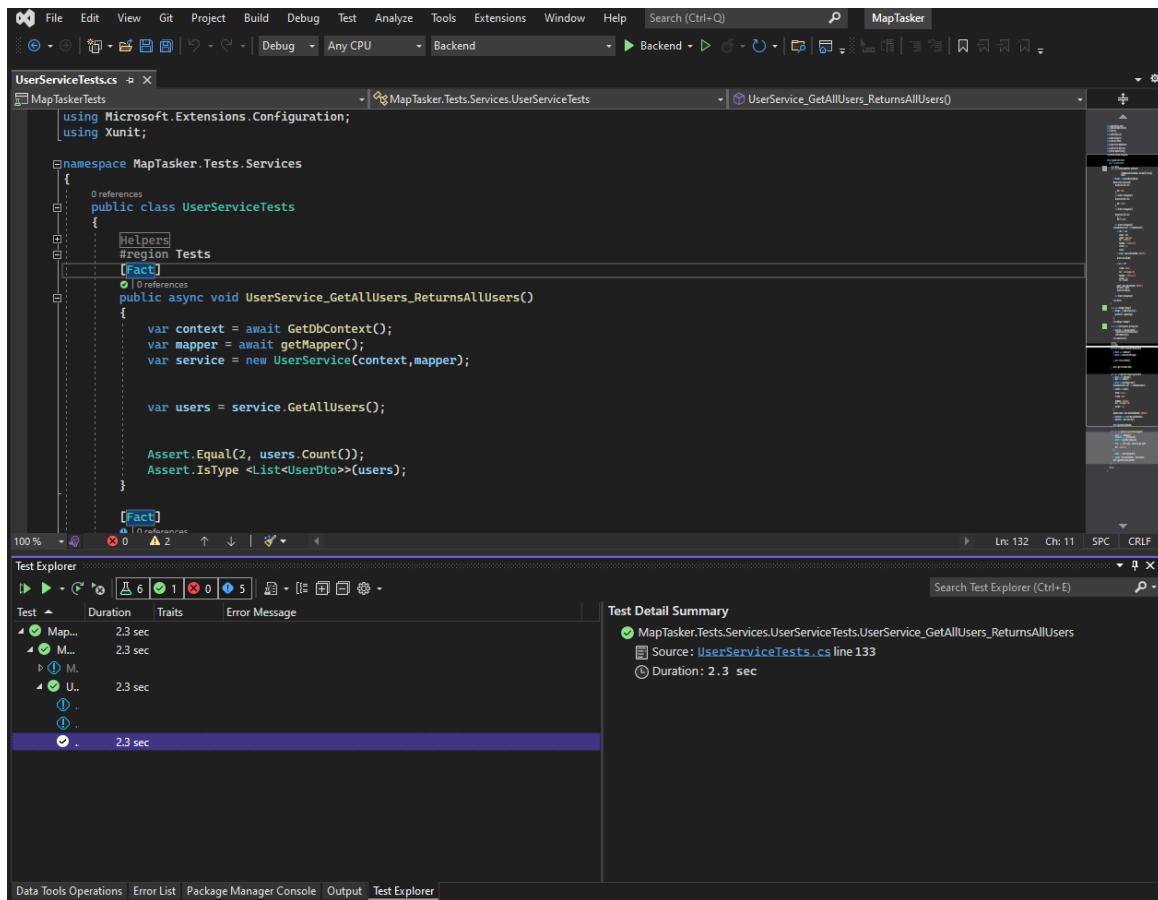
Ulaz:

1. Stvaranje novog UserServicea
2. Dohvaćanje metode GetAllUsers

Očekivani izlaz:

1. Stvara se novi UserService
2. Dohvaćaju se svi korisnici
3. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.



Slika 5.1: GetAllUsers - ReturnsAllUsers Test

Ispitni slučaj 2: Uspješna registracija korisnika

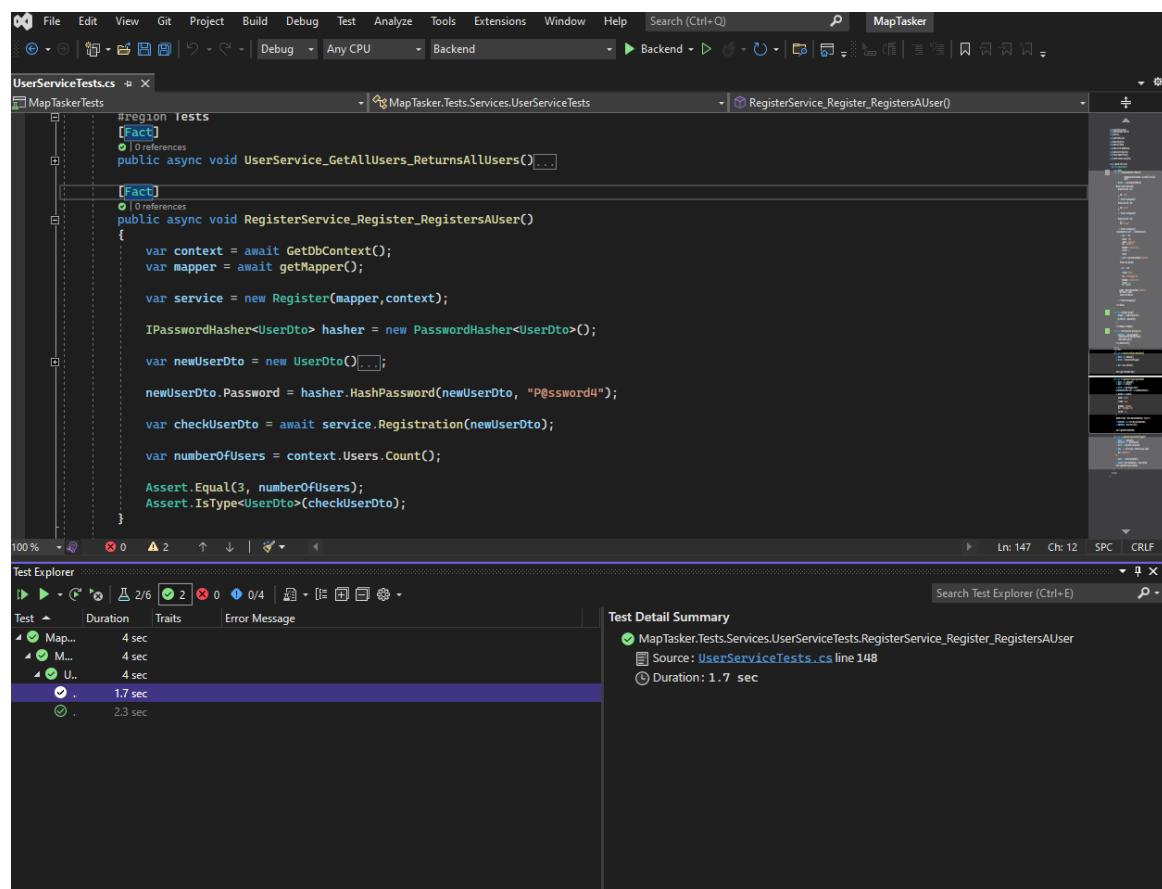
Ulaz:

1. Stvaranje novog RegisterServicea
 2. Kreiranje password hashera
 3. Instanciranje novog korisnika
 4. Postavljanje lozinke korisniku
 5. Dohvaćanje DTO-a Usera i ukupnog broja korisnika za provjeru

Očekivani izlaz:

1. Stvara se novi RegisterService
 2. Kreira se novi password hasher
 3. Novi korisnik se uspješno stvara
 4. Lozinka se uspješno postavlja korisniku
 5. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.



Slika 5.2: Register - RegistersAnUser Test

Ispitni slučaj 3: Uspješna prijava korisnika

Ulaz:

1. Stvaranje novog LoginServicea
2. Kreiranje tokena koji se sastoji od e-mail adrese i lozinke
3. Kreiranje handlera koji upravlja tokenima
4. Dohvaćanje tokena

Očekivani izlaz:

1. Stvara se novi LoginService
2. Kreira se token za provjeru
3. Kreira se handler koji upravlja tokenima
4. Token se uspješno dohvata
5. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

```

    UserServiceTests.cs
    MapTaskerTests
    MapTasker.Tests.Services.UserServiceTests
    LoginService_Login_UserSuccessfullyLoggedIn()

    [Fact]
    public async void LoginService_Login_UserSuccessfullyLoggedIn()
    {
        var context = await GetDbContext();
        var configuration = await getConfiguration();

        var service = new Login(context, configuration);

        var token = await service.LogIn(new Backend.Data.Login.LoginDto
        {
            Email = "admin@fer.hr",
            Password = "P@ssword1"
        });

        var handler = new JwtSecurityTokenHandler();

        var jsonToken = handler.ReadToken(token) as JwtSecurityToken;

        Assert.IsType<JwtSecurityToken>(jsonToken);
    }

```

Slika 5.3: UserSuccessfullyLoggedIn Test

Ispitni slučaj 4: Dohvaćanje svih prijava nestanka**Ulaz:**

1. Stvaranje novog generatora Id-a
2. Kreiranje novog MissingReport Servicea
3. Dohvaćanje svih prijava nestanaka metodom GetAllMissingReports

Očekivani izlaz:

1. Stvara se generator Id-a
2. Uspješno kreiranje MissingReporta
3. Uspješno se dohvataju sve prijave nestanaka
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q). The title bar says "MapTasker". The main area displays the code for "MissingReportAndCommentServiceTests.cs". The code contains a test method named "MissingReport_GetAllMissingReports_GetsAllReports()". The "Test Explorer" window at the bottom left shows four tests: "Map...", "M...", "M...", and ".". The test "Map..." has a duration of 7 sec. The test "M..." has a duration of 7 sec. The test "M..." has a duration of 1.5 sec and is highlighted in purple. The test "." has a duration of 1.5 sec. The "Test Detail Summary" window on the right shows the selected test "MissingReportAndCommentServiceTests.cs line 85" with a duration of 1.5 sec.

Slika 5.4: GetAllMissingReports Test

Ispitni slučaj 5: Stvaranje prijave nestanka

Ulaz:

1. Stvaranje novog generatora Id-a
2. Kreiranje MissingReport Servicea
3. Instanciranje prijave nestanka

Očekivani izlaz:

1. Stvara se generator Id-a
2. Uspješno kreiranje MissingReport Servicea
3. Prijava nestanka uspješno je instancirana
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and a search bar. The toolbar below has icons for file operations like Open, Save, and Print. The main window has tabs for 'MissingReport...serviceTests.cs' and 'UserServiceTests.cs'. The left pane shows the 'Test Explorer' with a list of tests: 'Map... 8.3 sec', 'M... 8.3 sec', 'M... 2.8 sec', 'M... 1.3 sec', 'M... 1.5 sec', and 'U... 5.5 sec'. The right pane shows the 'Test Detail Summary' for the selected test, which passed with a duration of 1.3 sec. The bottom status bar shows 'Ln: 100 Ch: 1 SPC CRLF'. The code editor on the right contains the following C# test code:

```

[Fact]
public async void MissingReport_CreateMissingReport_CreatesMissingReport()
{
    var context = await GetDbContext();
    var mapper = await getMapper();
    var generator = new IdGenerator();

    var service = new MissingReport(context, mapper, generator);

    var missingReport = await service.CreateMissingReport(new MissingReportDto
    {
        Id = 3,
        FirstName = "Marko",
        LastName = "Maric",
        Oib = "123456789918",
        Photo = "https://asfjkjh",
        Description = "Visok 180 plava majica zelene patike",
        ReportedAt = DateTime.Now,
    });

    Assert.IsType<MissingReportDto>(missingReport);
    Assert.Equal(3, context.MissingReports.Count());
}

```

Slika 5.5: CreatesMissingReport Test

Ispitni slučaj 6: Kreiranje komentara

Ulaz:

1. Stvaranje novog generatora Id-a
2. Kreiranje Comment Servicea
3. Instanciranje novog komentara

Očekivani izlaz:

1. Stvara se generator Id-a
2. Uspješno se kreiran Comment Service
3. Komentar je uspješno instanciran
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

The screenshot shows the Microsoft Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search (Ctrl+Q). The toolbar has icons for file operations like Open, Save, and Print. The status bar at the bottom shows 'Ln: 124 Ch: 12 SPC CRLF'.

The main area shows the code for `CommentService_CreateComment_CreatesComment`:

```

    Assert.IsType<MissingReportDto>(missingReport);
    Assert.Equal(3, context.MissingReports.Count());
}

[Fact]
public async void CommentService_CreateComment_CreatesComment()
{
    var context = await GetDbContext();
    var mapper = await getMapper();
    var generator = new IdGenerator();

    var service = new Comment(context, mapper, generator);

    var comment = await service.CreateComment(new CommentDto
    {
        Id = 2,
        ReportId = 1,
        Text = "Vidio mladica na glavnom kolodvoru."
    });

    Assert.IsType<CommentDto>(comment);
    Assert.Equal(2, context.Comments.Where(a=> a.ReportId == 1).Count());
}

```

The Test Explorer window at the bottom shows the test results:

Test	Duration
Map... 9.6 sec	
M... 9.6 sec	
M... 4 sec	
MapTasker.Tests.Services.MissingReportAndCommentServiceTests.CommentService_CreateComment_CreatesComment 1.3 sec	
U... 1.3 sec	
U... 1.5 sec	
U... 5.5 sec	

The 'Test Detail Summary' pane shows the details for the successful test:

- Source: `MissingReportAndCommentServiceTests.cs` line 125
- Duration: 1.3 sec

Slika 5.6: CreatesComments Test

5.2.2 Ispitivanje sustava

Svi Selenium testovi izvršeni su automatski. Ispitivanje je provedeno po obrascima uporabe. Ispitani su obrasci uporabe:

1. UC1 - Registracija u sustav
2. UC2 - Prijava u sustav
3. UC4 - Prijava nestale osobe
4. UC5 - Komentiranje prijave nestale osobe

Ispitni slučaj 1: Registracija

Ulaz:

1. Učitavanje početne stranice i namještanje veličine prozora
2. Pritisak na gumb "Register"
3. Korisnik unosi tražene podatke te odabire ulogu koju želi

Očekivani izlaz:

1. Početna stranica uspješno se učitava
2. Učitavanje forme za registraciju
3. Korisnik prima obavijest o uspješno obavljenoj registraciji
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

The screenshot shows the MapTasker interface with the following details:

- Project:** MapTaskerTests
- Test:** CreateCommentOnMissingReport
- Target URL:** http://localhost:3000
- Test Log (Log tab):**

Index	Command	Target	Value
1	✓ open	/	
2	✓ set window size	825x982	
3	✓ mouse over	css=MuiButtonBase-root:nth-child(4)	
4	✓ click	css=MuiButtonBase-root:nth-child(4)	
5	✓ type	name=username	antonio.misic@skole.
6	✓ type	name=password	RYNXscFWUUM4RKxX
7	✓ click	name=firstname	
8	✓ type	name=firstname	Antonio
9	✓ click	name.lastname	
10	✓ type	name.lastname	Mišić
11	✓ click	name=OIB	
12	✓ type	name=OIB	12345678911

Below the table are input fields for Command, Target, Value, and Description, each with a clear button.

Log:

```

9. click on name.lastname OK
10. type on name.lastname with value Mišić OK
11. click on name=OIB OK
12. type on name=OIB with value 12345678911 OK
13. click on name=phonenum OK
14. type on name=phonenum with value 0923554250 OK
15. type on name=email with value antoniohej@gmail.com OK
16. click on name=role OK
17. select on name=role with value label=Kartograf OK
18. click on css=.submit-button OK
'Registration' completed successfully
    
```

Reference:

Timestamps for the log entries:

 - 02:29:59
 - 02:29:59
 - 02:30:00
 - 02:30:00
 - 02:30:00
 - 02:30:00
 - 02:30:01
 - 02:30:01
 - 02:30:02
 - 02:30:02

Slika 5.7: Registration Test

Ispitni slučaj 2: Prijava

Ulaz:

1. Učitavanje početne stranice i namještanje veličine prozora
2. Pritisak na gumb "Login"
3. Korisnik unosi e-mail adresu i lozinku

Očekivani izlaz:

1. Početna stranica uspješno se učitava
2. Učitavanje forme za prijavu
3. Korisnik je uspješno prijavljen u sustav
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

The screenshot shows the MapTasker test runner interface. The main window displays a table of test steps for a 'Login' test case. The steps include opening the application, setting the window size, clicking the login button, entering email and password, and clicking the submit button. The target and value columns show the specific CSS or XPath for each action. Below the table are input fields for Command, Target, Value, and Description. At the bottom, there are tabs for Log and Reference, and a log window showing the execution details of each step.

Command	Target	Value
✓ open	/	
✓ set window size	750x826	
✓ click	css=MuiButtonBase-root:nth-child(3)	
✓ type	name=email	antonio.misic@skole.RYNXsdFWUM4RKxX
✓ type	name=password	
✓ click	css=login-page	
✓ type	name=email	admin@fer.hr
✓ click	css=login-page	
✓ type	name=password	P@ssword1
✓ click	css=submit-button	

Log

```

1. open on / OK
2. setWindowSize on 750x826 OK
3. click on css=MuiButtonBase-root:nth-child(3) OK
4. type on name=email with value antonio.misic@skole.RYNXsdFWUM4RKxX OK
5. type on name=password with value P@ssword1 OK
6. click on css=login-page OK
7. type on name=email with value admin@fer.hr OK
8. click on css=login-page OK
9. type on name=password with value P@ssword1 OK
10. click on css=submit-button OK
>Login completed successfully

```

Slika 5.8: Login Test

Ispitni slučaj 3: Kreiranje prijave nestanka

Ulaz:

1. Učitavanje početne stranice i namještanje veličine prozora
2. Pritisak na gumb "Prijavi"
3. Unošenje podataka o nestaloj osobi

Očekivani izlaz:

1. Početna stranica uspješno se učitava
2. Učitavanje forme za prijavu nestanka osobe
3. Obavijest o uspješnoj prijavi nestale osobe
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

Command	Target	Value
✓ type	name=firstName	
✓ click	name=lastName	
✓ type	name=lastName	
✓ click	name=oib	
✓ type	name=oib	12345678911
✓ click	css=form-input:nth-child(9)	
✓ type	css=form-input:nth-child(9)	Nestao prije 10 min
✓ click	name=reportedAt	
✓ type	name=reportedAt	2023-01-03
✓ click	name=photo	
✓ click	css=submit-button	
✓ assertAlert	Uspešna prijava nestale osobel	

Log:

```

7. type on name=lastName with value Mišić OK
8. click on name=oib OK
9. type on name=oib with value 12345678911 OK
10. click on css=form-input:nth-child(9) with value Nestao prije 10 min OK
11. type on css=form-input:nth-child(9) with value 2023-01-03 OK
12. click on name=reportedAt OK
13. type on name=reportedAt with value 2023-01-03 OK
14. click on name=photo OK
15. click on css=submit-button OK
16. assertAlert on Uspešna prijava nestale osobel OK
'CreateMissingReport' completed successfully
    
```

Slika 5.9: CreateMissingReport Test

Ispitni slučaj 4: Kreiranje komentara na prijavu nestanka

Ulaz:

1. Učitavanje početne stranice i namještanje veličine prozora
2. Pritisak na gumb "Nestale osobe"
3. Stvaranje komentara kraj nestale osobe

Očekivani izlaz:

1. Početna stranica uspješno se učitava
2. Učitavanje stranice s nestalim osobama
3. Komentar je prikazan na stranici
4. Rezultat testa je ispravan

Izlaz: Test je zadovoljen. Aplikacija je prošla test.

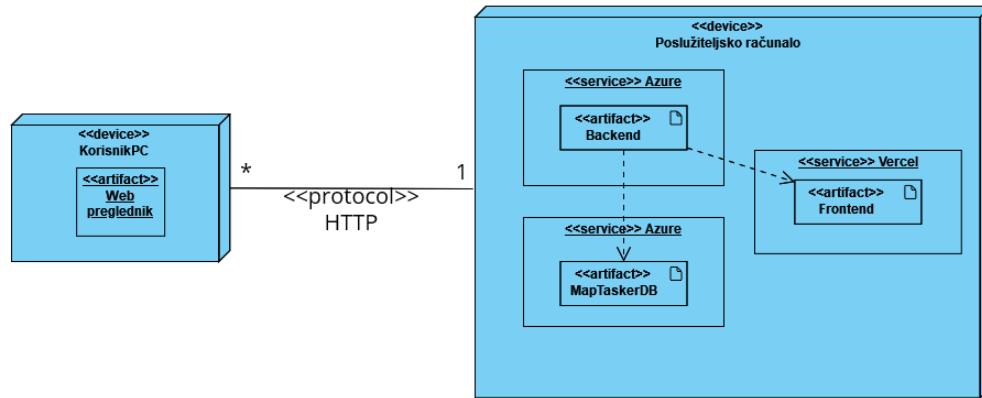
The screenshot shows the MapTasker software interface with the following details:

- Project:** MapTaskerTests
- Test:** CreateCommentOnMissingReport
- Log:**
 - 'CreateMissingReport' completed successfully
 - Running 'CreateCommentOnMissingReport'
 - 1. open on / OK
 - 2. setWindowSize on 825x982 OK
 - 3. click on css=MuiButton-text:nth-child(2) OK
 - 4. mouseOver on css=MuiButtonBase-root:nth-child(2) OK
 - 5. mouseOut on css=MuiButton-text:nth-child(2) OK
 - 6. click on id=r0: OK
 - 7. type on id=r0: with value Vidio sam ga kod Glavnog kolodvora OK
 - 8. click on css=MuiButton-contained OK
 - 'CreateCommentOnMissingReport' completed successfully

Slika 5.10: CreateComment Test

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* postavljeni na Azure serveru dok se *frontend* nalazi na Vercel serveru.



Slika 5.11: Dijagram razmještaja

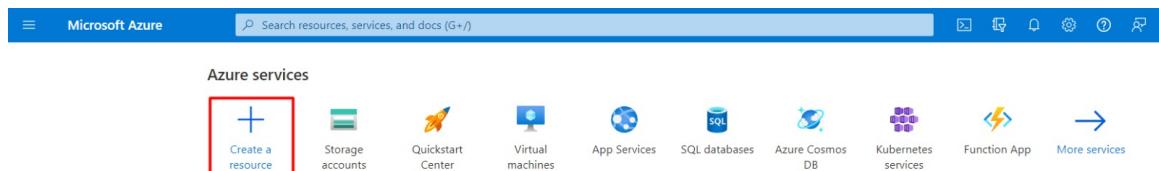
5.4 Upute za puštanje u pogon

Prijava u Azure račun

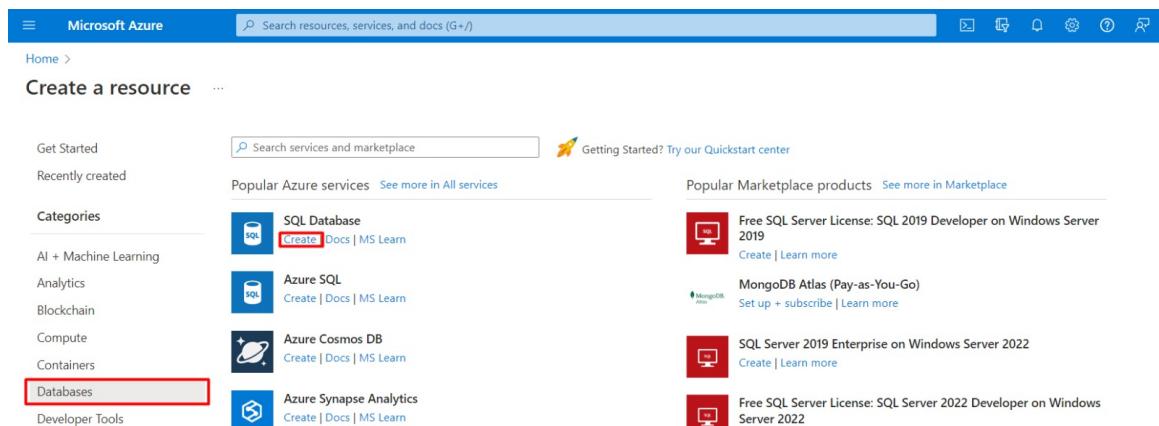
Korisnik se treba prijaviti u svoj Microsoft Azure račun.

Stvaranje baze podataka na Azure serveru

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



Slika 5.12: Odabir opcije "Create a resource"



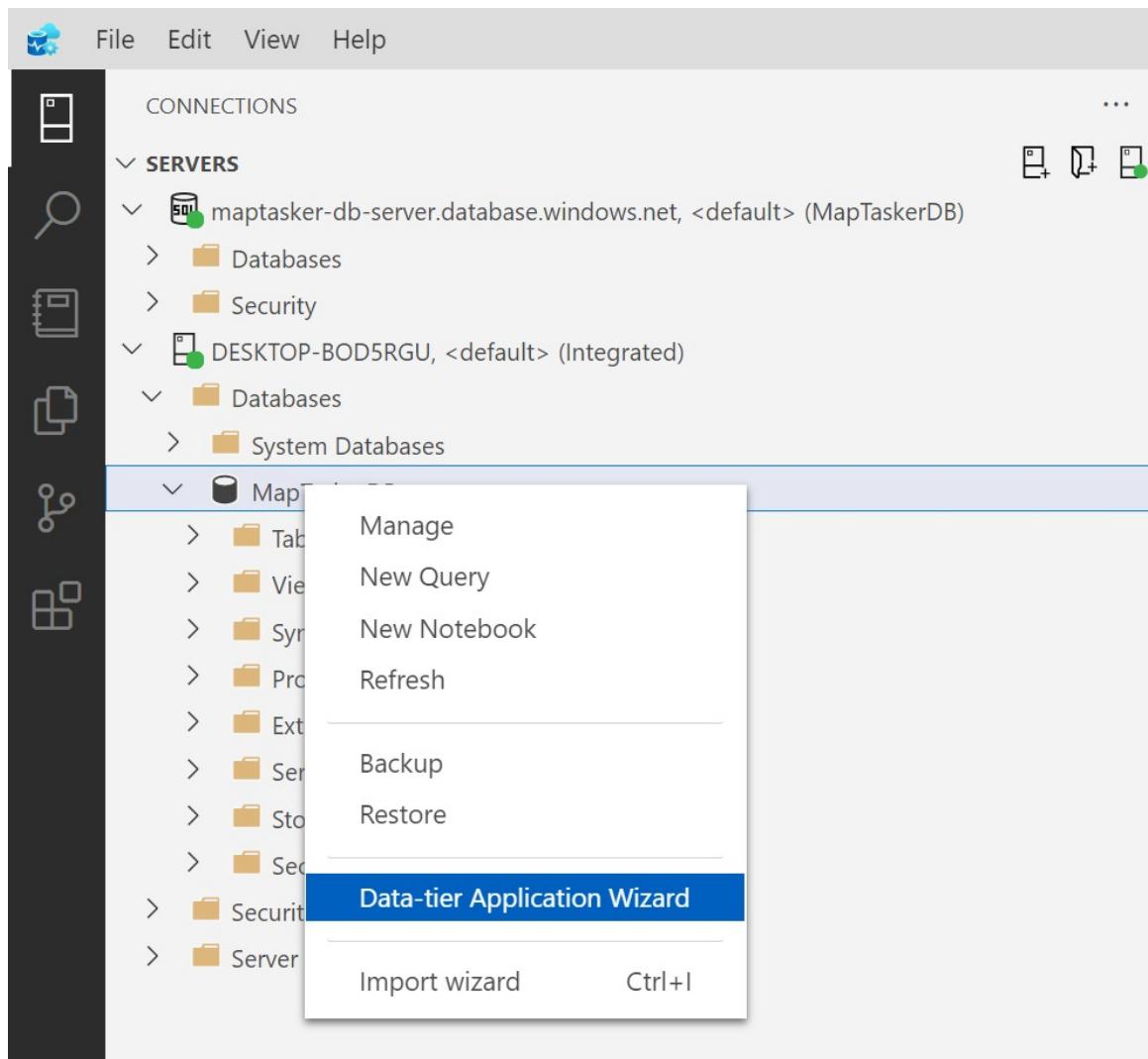
Slika 5.13: Odabir opcije "SQL Database"

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.

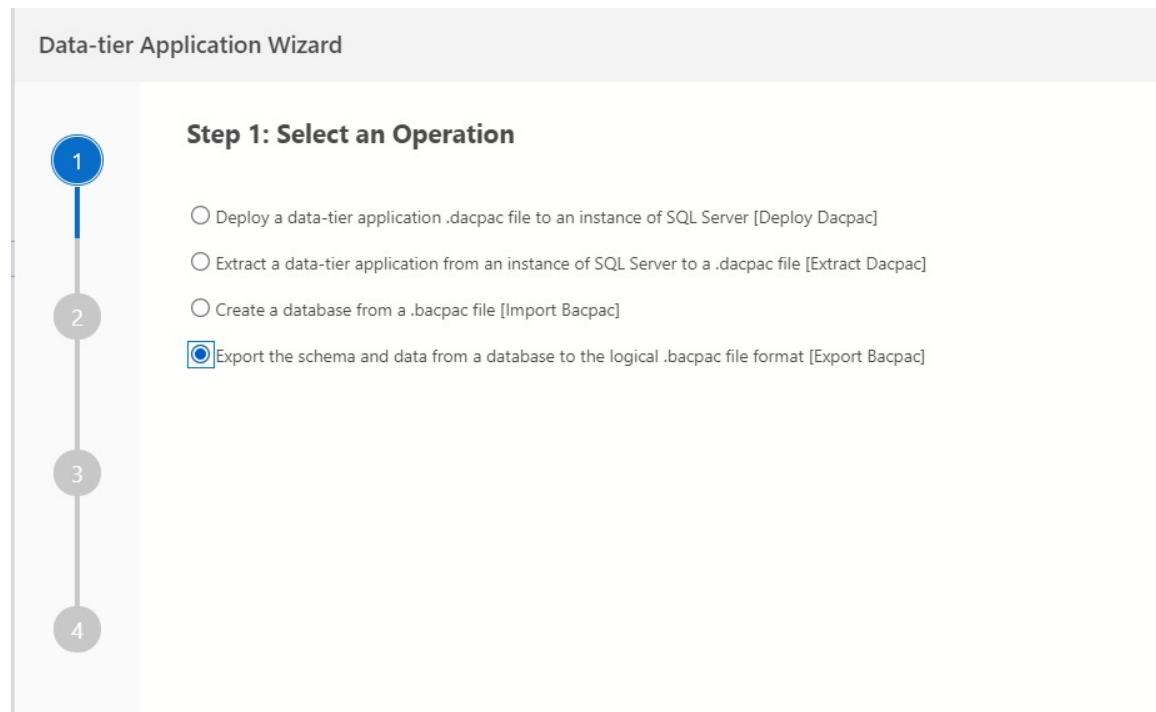
The screenshot shows the 'Create SQL Database' wizard in the Microsoft Azure portal. In the 'Project details' section, the 'Subscription' dropdown is set to 'Azure subscription 1'. The 'Resource group' dropdown shows 'Maptasker' with a red box around the 'Create new' button. In the 'Database details' section, the 'Database name' field is empty. The 'Server' dropdown shows 'maptasker-db-server (West Europe)' with a red box around the 'Create new' button. Below these, there's a question about using a SQL elastic pool with 'No' selected. Under 'Workload environment', 'Production' is selected. At the bottom, a red box surrounds the 'Review + create' button, while the 'Next : Networking >' button is white.

Slika 5.14: 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 .bacpac datoteke i pričekati dok se datoteka ne stvori.

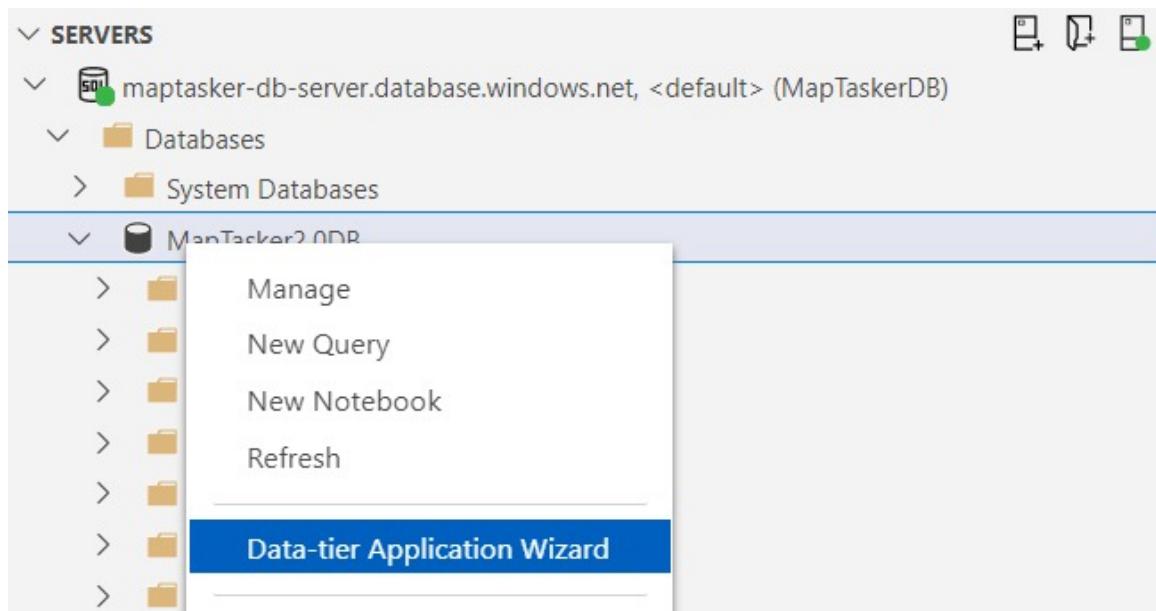


Slika 5.15: Odabir opcije "Data-tier Application Wizard"

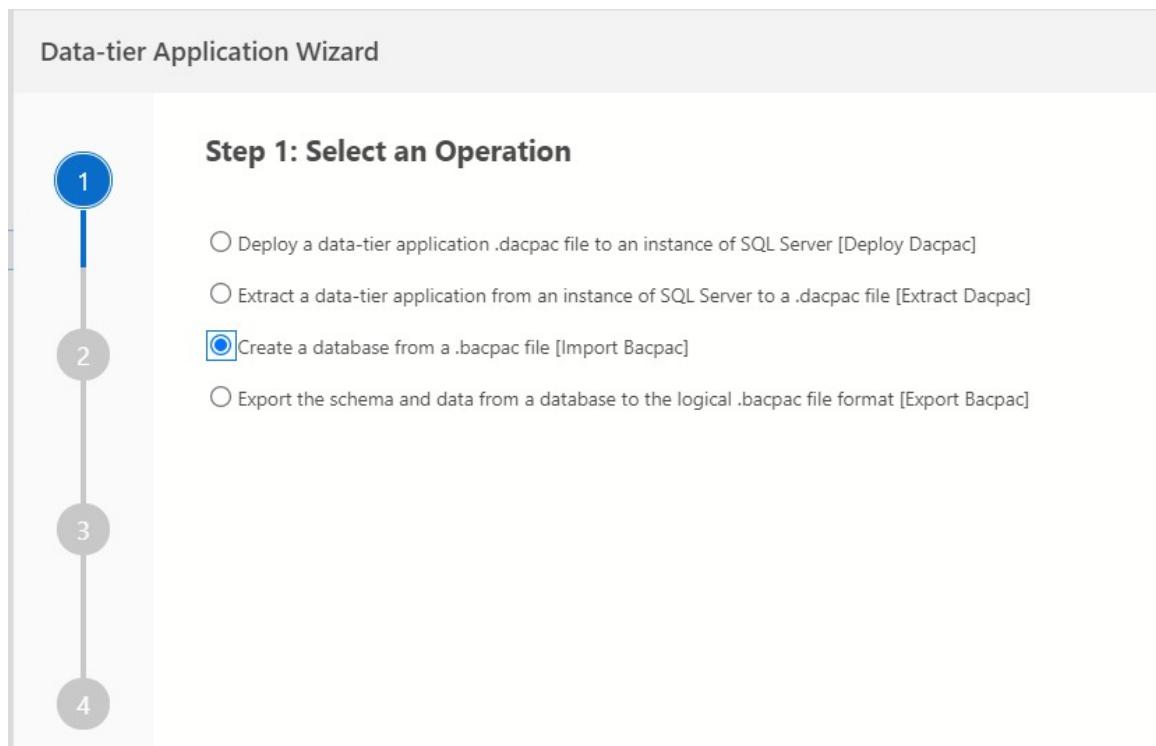


Slika 5.16: Odabir željene operacije

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.



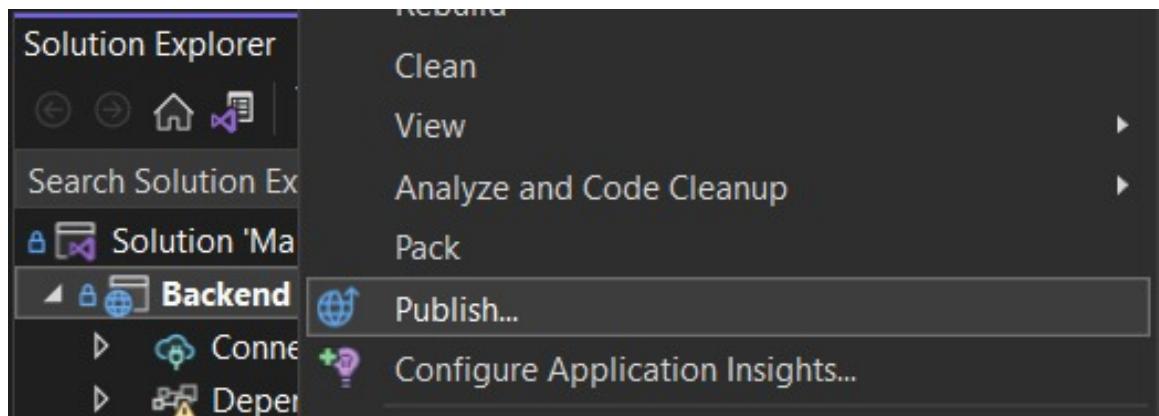
Slika 5.17: Odabir opcije "Data-tier Application Wizard"



Slika 5.18: Odabir operacija stvaranja baze

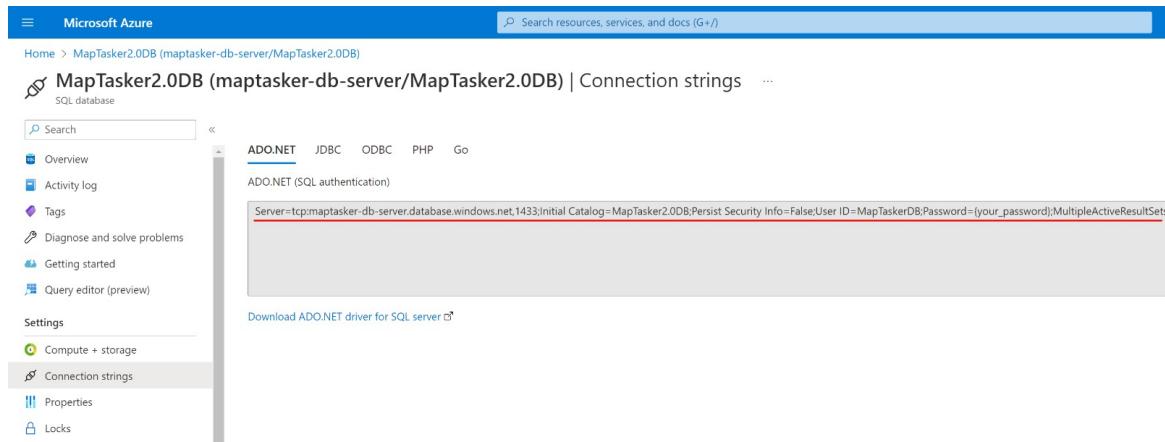
Stvaranje backend API-ja na Azure serveru

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



Slika 5.19: 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 5.20: Lokacija vrijednosti connection stringa

The screenshot shows the Azure App Service configuration page for 'MapTaskerAPI'. The top navigation bar includes the Azure logo, a search bar, and a 'Save' button. Below the navigation bar, the URL 'Home > MapTaskerAPI | Configuration' is visible. On the left, a sidebar menu lists Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Microsoft Defender for Cloud, Events (preview), Deployment (Quickstart, Deployment slots, Deployment Center), Settings (Configuration, Authentication, Application Insights, Identity, Backups, Custom domains, Custom domains (classic), Certificates, TLS/SSL settings (classic)), and Configuration (which is selected and highlighted in grey). The main content area is titled 'Application settings' and shows two entries: 'CONNECTION_STRING' (Value: 'Hidden value. Click to show value') and 'WEBSITE_NODE_DEFAULT_VERSION' (Value: '6.9.1'). Below this, there is a section titled 'Connection strings' with a table:

Name	Type	Source
ConnectionString	SQLAzure	App Service

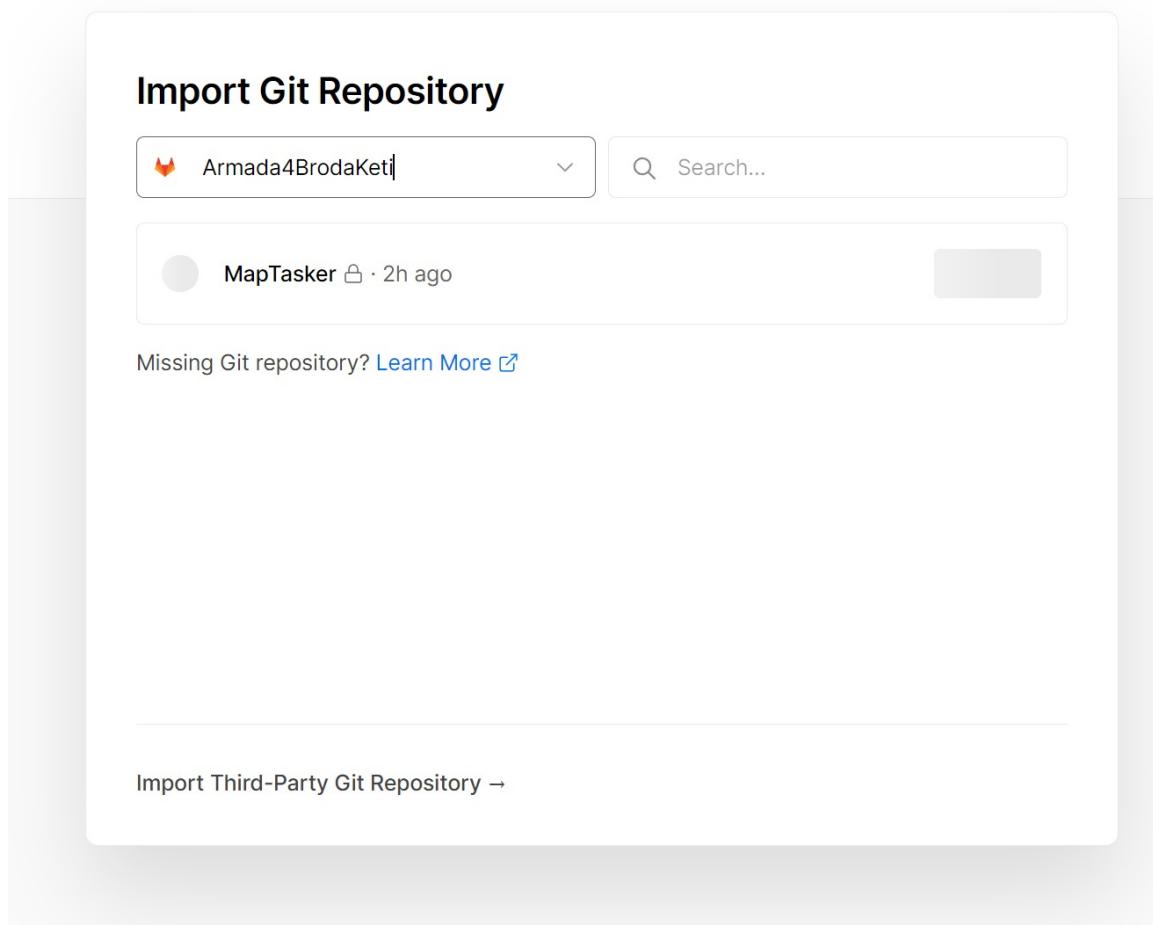
Slika 5.21: Unos vrijednosti connection stringa

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 5.22: 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

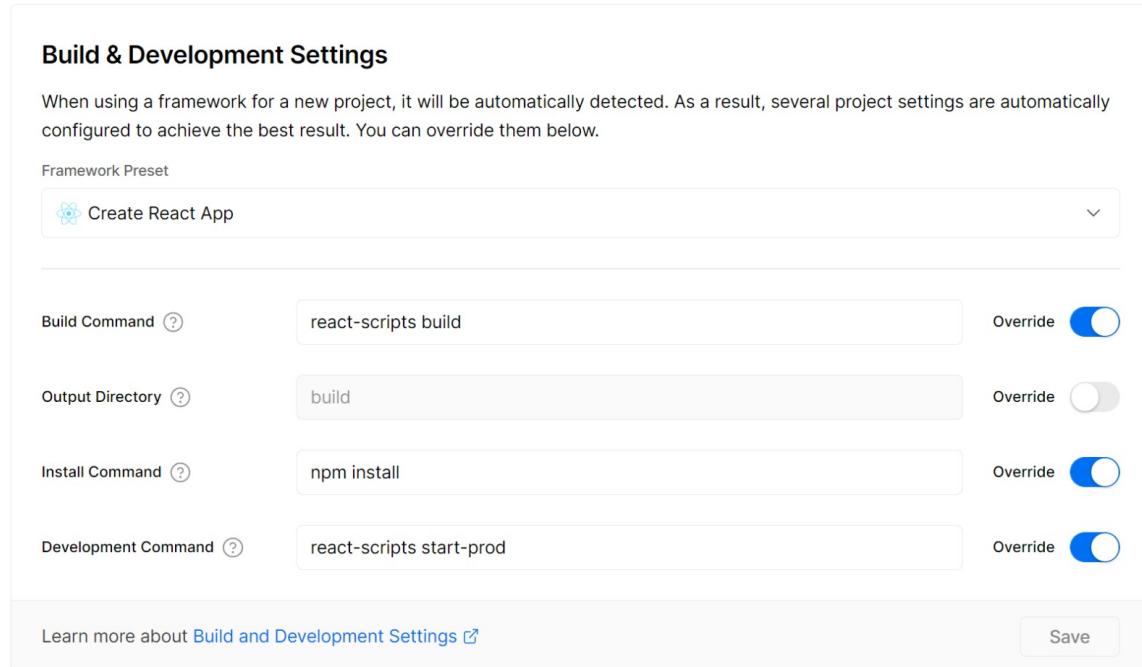
When using a framework for a new project, it will be automatically detected. As a result, several project settings are automatically configured to achieve the best result. You can override them below.

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 5.23: 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	Value	⋮
e.g. CLIENT_KEY		<input type="button" value="⊖"/>
⊕ Add another		

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 5.24: 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 spasiocima. 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 programske 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 spasioci.

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 pomoći kod pronalaženja nestalih osoba.

Izrada ovog projektnog zadatka dobro je doprinijela razvoju vještina članovima tima koji se prvi put upoznavaju s ovakvim zadatokm 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.

Popis literature

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinzh>
2. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Prijava nestale osobe preko ICMP	9
2.2	Popunjavanje informacije korisnika	9
3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranih i registriranih korisnika aplikacije	19
3.2	Dijagram obrasca uporabe, funkcionalnost spasioca, voditelja operacije i kartografa	20
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	20
3.4	Sekvencijski dijagram za UC4 - Prijava nestale osobe	21
3.5	Sekvencijski dijagram za UC10 - Stvaranje operacije	23
3.6	Sekvencijski dijagram za UC11 - Mijenjanje statusa bloka	25
3.7	Sekvencijski dijagram za UC20 - Potvrda registracije	27
4.1	Arhitektura sustava	29
4.2	MVC model	30
4.3	ER dijagram baze podataka	36
4.4	Dijagram razreda 1 - Register i Login	37
4.5	Dijagram razreda 2 - Missing Report i Comment	38
4.6	Dijagram razreda 3 - Operations, Areas, Regions, Blocks, Buildings .	39
4.7	Dijagram razreda 4 - User i Role	40
4.8	Dijagram razreda 5 - Statistics	41
4.9	Dijagram stanja - Voditelj operacije	42
4.10	Dijagram aktivnosti registracije, prijave korisnika te prijava nestale osobe	44
4.11	Dijagram komponenti	45
5.1	GetAllUsers - ReturnsAllUsers Test	48
5.2	Register - RegistersAnUser Test	49
5.3	UserSuccessfullyLoggedIn Test	50
5.4	GetAllMissingReports Test	51
5.5	CreatesMissingReport Test	52

5.6	CreatesComments Test	53
5.7	Registration Test	55
5.8	Login Test	56
5.9	CreateMissingReport Test	57
5.10	CreateComment Test	58
5.11	Dijagram razmještaja	59
5.12	Odabir opcije "Create a resource"	60
5.13	Odabir opcije "SQL Database"	60
5.14	Dodavanje željenih svojstava bazi	61
5.15	Odabir opcije "Data-tier Application Wizard"	62
5.16	Odabir željene operacije	63
5.17	Odabir opcije "Data-tier Application Wizard"	64
5.18	Odabir operacije stvaranja baze	64
5.19	Odabir opcije "Publish"	65
5.20	Lokacija vrijednosti connection stringa	66
5.21	Unos vrijednosti connection stringa	66
5.22	Uvoz git repozitorija projekta	67
5.23	Naredbe za pokretanje npm skripti	68
5.24	Podešavanje varijabli okruženja	69
6.1	Prikaz aktivnosti na repozitoriju	80
6.2	Prikaz aktivnosti na repozitoriju	80
6.3	Prikaz aktivnosti na repozitoriju	81

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

6. sastanak

- Datum: 11. prosinca 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - podjela rada (podjela posla na frontend i backend po ekranima)

7. sastanak

- Datum: 17. prosinca 2022.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - prezentacija napravljenog ostalim članovima tima
 - daljnja podjela zadataka

8. sastanak

- Datum: 21. prosinca 2022.
- Prisustvovali: Antonio Mišić, Ema Vlainić, Jurica Uglešić
- Teme sastanka:
 - prezentacija alfa inačice projekta asistentu i demonstratoru

9. sastanak

- Datum: 05. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - dorađivanje alfa inačice
 - podjela pisanja dokumentacije
 - komentiranje izrade dijagrama stanja, komponenti i aktivnosti

10. sastanak

- Datum: 02. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - dorađivanje alfa inačice
 - podjela pisanja dokumentacije
 - komentiranje izrade dijagrama stanja, komponenti i aktivnosti

11. sastanak

- Datum: 05. siječnja 2023.
- Prisustvovali: Bojan Puvača, Matija Jelavić
- Teme sastanka:
 - sastanak s asistentom i demonstratorom
 - komentiranje izrade dijagrama i implementacije projekta

12. sastanak

- Datum: 09. siječnja 2023.
- Prisustvovali: Bojan Puvača, Antonio Mišić, Matija Jelavić
- Teme sastanka:
 - ispravljanje funkcija na backendu

13. sastanak

- Datum: 12. siječnja 2023.
- Prisustvovali: svi članovi tima
- Teme sastanka:
 - zadnji sastanak s asistentom i demonstratorom
 - prezentacija projekta

Tablica aktivnosti

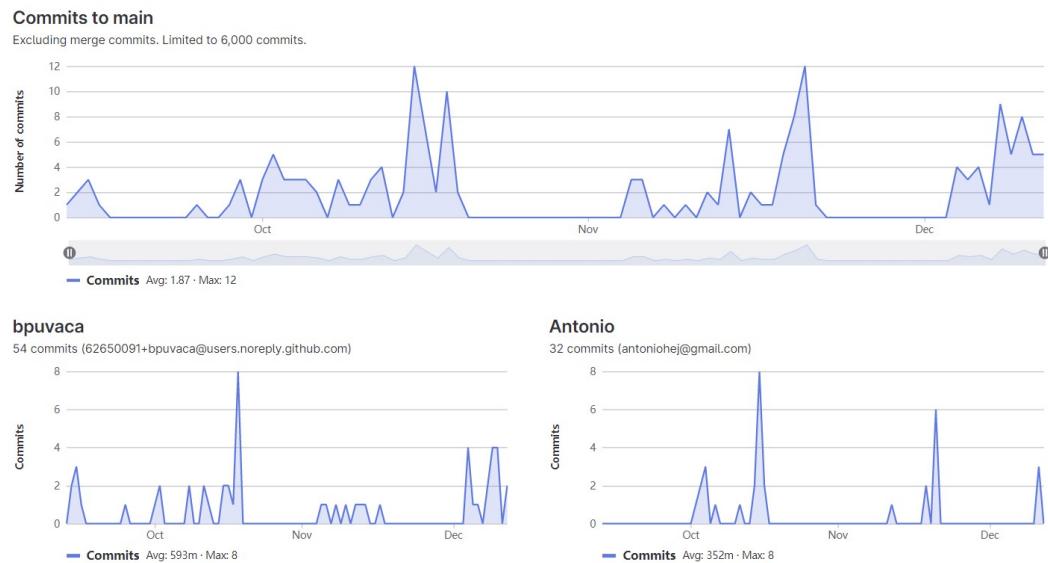
	Bojan Puvača	Matija Jelavić	Danijel Kovačević	Antonio Mišić	Katarina Šabić	Jurica Uglešić	Ema Vlainić
Upravljanje projektom	8						
Opis projektnog zadatka						2	
Funkcionalni zahtjevi	2			1			
Opis pojedinih obrazaca		4		3			
Dijagram obrazaca	6						
Sekvencijski dijagrami						6	
Opis ostalih zahtjeva					1		
Arhitektura i dizajn sustava			2				
Baza podataka		2					
Dijagram razreda	2	2	2		2		
Dijagram stanja				5			
Dijagram aktivnosti					3		
Dijagram komponenti		3					
Korištene tehnologije i alati		1					
Ispitivanje programskog rješenja		2	8				
Dijagram razmještaja					1		
Upute za puštanje u pogon							
Dnevnik sastajanja				4			
Zaključak i budući rad					1		

Nastavljeno na idućoj stranici

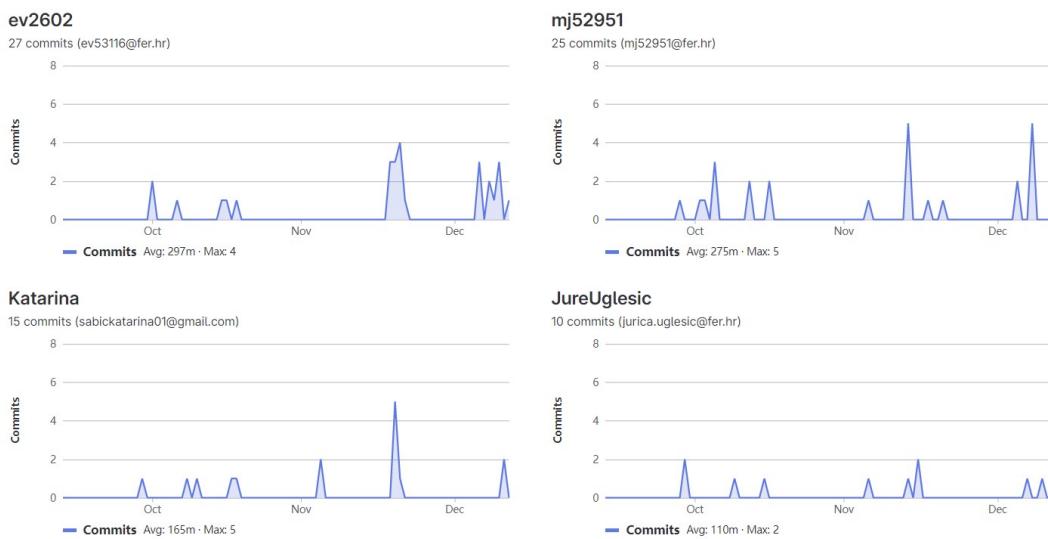
Nastavljeno od prethodne stranice

	Bojan Puvača	Matija Jelavić	Danijel Kovačević	Antonio Mišić	Katarina Šabić	Jurica Uglešić	Ema Vlainić
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						
Backend	3	12		15		6	
Frontend	20				7		15
Registracija, validacija, prijava korisnika u sustav	6				6		
Puštanje aplikacije u pogon	13						

Dijagrami pregleda promjena



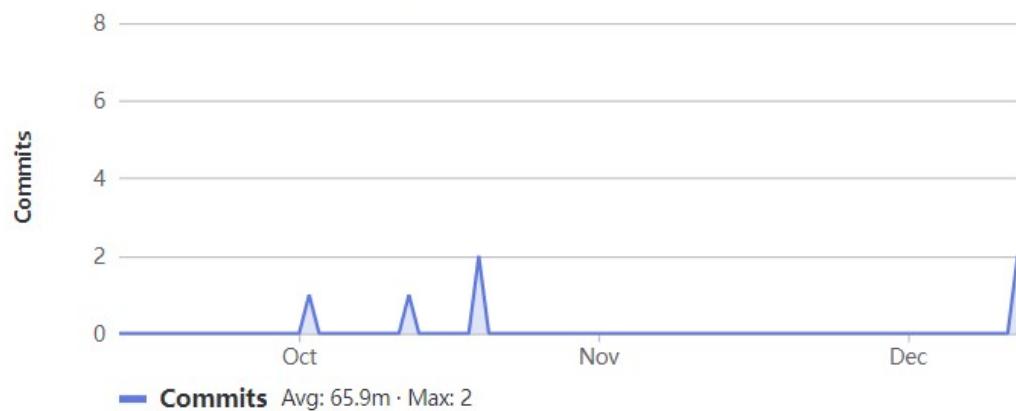
Slika 6.1: Prikaz aktivnosti na repozitoriju



Slika 6.2: Prikaz aktivnosti na repozitoriju

dkova12314

6 commits (danijel.kovacevic@fer.hr)



Slika 6.3: Prikaz aktivnosti na repozitoriju