

Oblikovanje programske potpore

Ak. god. 2019./2020.

Manje smeće više sreće

Dokumentacija, Rev. 1

Grupa: *BashCrash*

Voditelj: *Marko Lazarić*

Datum predaje: 15.11.2019.

Nastavnik: *Tomislav Jukić*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	18
3.2 Ostali zahtjevi	26
4 Arhitektura i dizajn sustava	27
4.1 Baza podataka	30
4.1.1 Opis tablica	30
4.1.2 Dijagram baze podataka	33
4.2 Dijagram razreda	35
4.3 Dijagram stanja	38
4.4 Dijagram aktivnosti	39
4.5 Dijagram komponenti	40
5 Implementacija i korisničko sučelje	42
5.1 Korištene tehnologije i alati	42
5.2 Ispitivanje programskog rješenja	44
5.2.1 Ispitivanje komponenti	45
5.2.2 Ispitivanje sustava	53
5.3 Dijagram razmještaja	63
5.4 Upute za puštanje u pogon	64
6 Zaključak i budući rad	69
Popis literature	70
Indeks slika i dijagrama	72

Dodatak: Prikaz aktivnosti grupe

73

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Započeo opis projektnog zadatka i obrasce uporabe.	Vale	26.10.2019.
0.2	Ostali zahtjevi.	Lazarić	26.10.2019.
0.3	Funkcionalni zahtjevi.	Benotić	27.10.2019.
0.4	Obrasci uporabe za direktora.	Pavliš	28.10.2019.
0.5	Obrasci uporabe za neregistrirane korisnike.	Jakopec	28.10.2019.
0.6	Obrasci uporabe za zaposlenike komunalne službe. Opis direktora komunalne službe.	Novak	28.10.2019.
0.7	Dijagram baze podataka.	Lazarić	30.10.2019.
0.8	Baza podataka i opis tablica.	Vale	31.10.2019.
0.9	Sekvencijski dijagrami za UC1 i UC8.	Benotić	01.11.2019.
0.10	Opis projektnog zadatka.	Pavliš & Lazarić	02.11.2019.
0.11	Sekvencijski dijagrami za UC9 i UC13.	Pavliš	03.11.2019.
0.12	Dijagram obrazaca uporabe.	Teofilović	04.11.2019.
0.13	Ispravak grešaka u dokumentaciji i ažuriranje dnevnika sastajanja.	Vale	08.11.2019.
0.14	Arhitektura i dizajn sustava.	Jakopec	09.11.2019.
0.15	Ispravak dokumentacije.	Jakopec	15.11.2019.
1.0	Prva verzija dokumentacije.	Jakopec & Lazarić	15.11.2019.
1.1	Korištene tehnologije i alati.	Pavliš	30.12.2019.
1.2	Zaključak i budući rad.	Pavliš	31.12.2019.
1.3	Ispitivanje komponenti.	Lazarić	01.01.2020.
1.4	Dijagram komponenti.	Benotić	03.01.2020.
1.5	Dijagrami stanja i aktivnosti.	Vale	04.01.2020.
1.6	Dijagram razmještaja.	Teofilović	06.01.2020.
1.7	Ispitivanje sustava.	Lazarić	06.01.2020.
1.8	Dijagrami razreda.	Lazarić	11.01.2020.

Rev.	Opis promjene/dodatka	Autori	Datum
1.9	Upute za puštanje u pogon	Jakopec	12.01.2020.
1.10	Korištene tehnologije i alati.	Pavliš	15.01.2020.
1.11	Dnevnik sastajanja.	Vale	15.01.2020.
1.12	Dijagram komponenti.	Benotić	16.01.2020.
1.13	Ispitivanje sustava Selenium.	Jakopec	16.01.2020.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za web aplikaciju "*Manje smeće više sreće*" koja će građanima omogućiti ocjenjivanje i komentiranje stanja kontejnera te će djelatnicima komunalnih službi omogućiti praćenje tih recenzija. Aplikacija bi time povećala povezanost građana s djelatnicima komunalnih službi, što bi omogućilo rješavanje problema odvoza smeća na efikasniji i brži način. Osim toga, aplikacija bi omogućila građanima transparentniji uvid u rad komunalnih službi.

Prilikom pokretanja sustava, prikazuje se karta s ucrtanim kontejnerima koji se nalaze u blizini korisnika te traka za pretraživanje kontejnera. Neregistriranom korisniku, odabirom kontejnera na karti, otvaraju se opće informacije o istom; ID kontejnera, adresa te povijest stanja koja je vidljiva kroz recenzije korisnika i slike koje su korisnici objavljivali.

Osim pregleda kontejnera odabirom na karti, korisnicima je ponuđena opcija pretraživanja putem jedinstvenog ID-a koji je prikazan na kontejneru. Upisom valjanog ID-a, otvaraju se opće informacije o kontejneru, a u suprotnom korisnik dobiva odgovarajuću poruku.

Neregistriranom korisniku omogućeno je prijavljivanje u sustav s postojećim računom ili registracija novog računa. Za registraciju novog računa potrebni su sljedeći podatci:

- korisničko ime
- lozinka
- e-mail adresa

Registracijom u sustav, korisnik dobiva mogućnost ocjenjivanja i komentiranja stanja kontejnera kao građanin. Korisnik, osim kao građanin, može biti djelatnik ili direktor komunalne službe.

Građanin je registrirani korisnik koji ima mogućnost ocjenjivanja i komentiranja stanja kontejnera. Nakon što je odabrao kontejner, građaninu je omogućeno davanje numeričke ocjene punoće kontejnera te ocjene neurednosti kontejnera. Osim navedenog, omogućeno mu je ostavljanje tekstualnih komentara, te slika samog

kontejnera kako bi po potrebi bolje prikazao stanje kontejnera. Građaninu je omogućeno praćenje kontejnera čime dobiva e-mail obavijesti pri promjeni stanja kontejnera te pregled kontejnera koje trenutno prati i mogućnost prestanka praćenja pojedinih kontejnera. Obavijesti o promjeni stanja kontejnera zahtijevaju potvrđenu e-mail adresu kako ne bi slali e-mail poruke na krivu adresu.

Djelatnik komunalne službe kroz aplikaciju ima pristup dodijeljenim kontejnerima svoje komunalne službe. Omogućen mu je pregled svojih kontejnera i njihovih stanja, te može označiti da je ispraznio određeni kontejner čime se šalje obavijest svim pretplaćenom korisnicima tog kontejnera. Osim toga, omogućen mu je prikaz svoje rute koju aplikacija generira na temelju dodijeljenih kontejnera i njihovih stanja. Područje grada podijeljeno je na zone za koje su zadužene ekipe komunalnih djelatnika.

Direktor komunalne službe ima širi spektar mogućnosti rada u aplikaciji. Njemu je omogućen pregled svih kontejnera, zaposlenika, zona i ruta svoje komunalne službe. Odabirom kontejnera, omogućeno mu je brisanje, premještanje i označavanje da je kontejner poslan u skladište. Omogućeno mu je dodavanje kontejnera na nekoj od lokacija po izboru pri čemu mora ručno postaviti adresu i koordinate kontejnera. Uz dodavanje, omogućeno mu je i brisanje kontejnera te premještanje kontejnera na neku drugu lokaciju. Osim toga, direktoru je omogućeno dodavanje i uklanjanje djelatnika iz svoje službe.

Kontejneri u sebi imaju senzor koji detektira je li kontejner očišćen ili nije. Pomoću tog senzora moguće je otkriti koliko dugo neki kontejner nije očišćen, ali nije moguće otkriti treba li kontejner čišćenje ili ne, zato su potrebni građani i njihove recenzije kako bi ovaj sustav bio efikasan i kvalitetan.

Svi kontejneri u ovom projektu bit će smješteni u gradu Zagrebu te se očekuje da će korisnici aplikacije biti građani grada Zagreba i da će oni imati najveću korist od same aplikacije. U budućnosti je, međutim, moguće da proširenje na cijelu Hrvatsku ili šire, što bi omogućilo više korisnika koji će imati izravnu korist od ove aplikacije kao i veću uštedu u pogledu vremena, ali i novca, za komunalne radnike na čijem će području biti ostvaren rad ove aplikacije.

Jedna od sličnih aplikacija koja se bavi sličnim problemima je aplikacija Zagrebačkog holdinga. Njihova aplikacija se, naravno, bavi puno većim opsegom poslova nego što je specificirano u ovom projektnom zadatku, ali se u domeni održavanja kontejnera poklapa s domenom ovog projektnog zadatka. Ključna razlika između ovog projekta i aplikacije Zagrebačkog holdinga je transparentnost, odnosno ne-

dostatak iste. Na njihovoj aplikaciji nije moguće pratiti kada je (i je li) neki kontejner očišćen niti se može ocijeniti stanje pojedinog kontejnera. Zagrebački holding, također, nema način na koji će pratiti koliko je neki kontejner očišćen te se, prema tome, odvoz smeća uvijek vrši u isto vrijeme i na istim kontejnerima, što može biti neefikasno. Jedan od ciljeva ove aplikacije je, uz transparentnost, veća efikasnost nego što je trenutno efikasnost aplikacije Zagrebačkog holdinga. Uz to, naravno, važno je dati građanima mogućnost pridonijeti cijelom tom procesu jer su ipak oni u središtu ove aplikacije.

Evolucija je neizostavni dio aplikacija programske potpore. Kao takva, aplikacija je podložna promjeni i nadogradnjama, stoga će u nastavku biti navedene poneke nove mogućnosti i svojstva za svaku vrstu korisnika aplikacije.

Moguće buduće nadogradnje projektnog zadatka za građanina uključuju:

- pronalazak kontejnera preko QR koda
- opcije zaboravio sam lozinku/zapamti kako bi proces prijave bio što lakši i bolji
- potvrđivanje e-mail adrese preko tokena u e-mail poruci
- obavijesti unutar aplikacije za praćene kontejnere (uz obavijesti preko e-maila)

Moguće buduće nadogradnje projektnog zadatka za djelatnika komunalne službe uključuju:

- brzo skeniranje QR koda kako bi se potvrdilo da je taj kontejner sa sigurnošću isprazan
- generiranje optimalne rute čišćenja kontejnera koja će se mijenjati dinamički tijekom vožnje
- paljenje točkice na ekranu kada se skenira QR kod kontejnera

Moguće buduće nadogradnje projektnog zadatka za direktora komunalne službe uključuju:

- karta s lokacijama kontejnera na osobnoj aplikaciji na kojoj će biti oznake u boji ovisno o stanju kontejnera
- očitavanje trenutne lokacije te dodavanje kontejnera na trenutnoj lokaciji

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Direktor službe
2. Korisnici stranice
 - (a) Neregistrirani korisnici
 - (b) Registrirani korisnici
3. Službenici
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) na karti pregledati pozicije kontejnera
 - (b) pronaći kontejner pomoću njegovog ID-a ili adrese
 - (c) vidjeti trenutno i stanje kontejnera do 3 mjeseca unatrag
 - (d) registrirati se u sustav te stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, lozinka i e-mail adresa
2. Registrirani korisnik (inicijator) može:
 - (a) pregledavati i mijenjati svoje osobne podatke
 - (b) izbrisati svoj korisnički račun
 - (c) ocijeniti stanje i pisati komentare o kontejneru
 - (d) postaviti fotografije kontejnera
 - (e) odabrati kontejnere za praćenje njihovog stanja
3. Službenik (inicijator) može:
 - (a) označiti da je ispraznio kontejner
 - (b) premjestiti kontejner

(c) povući kontejner u skladište

4. Direktor službe (inicijator) može:

- (a) dodati i izbrisati korisnika iz svoje službe
- (b) dodati, izbrisati i premjestiti kontejner

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o kontejnerima

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pretraživanje i pregled kontejnera putem ID-a

- **Glavni sudionik:** Svi korisnici
- **Cilj:** Pregledati kontejnere i njihovo stanje kroz vremenski period
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik u tražilicu upisuje ID kontejnera
 2. Prikazuju se informacije o kontejneru
- **Opis mogućih odstupanja:**
 - 2.a Neispravan/nepostojeći ID
 1. Sustav obavještava korisnike o neuspjelom pokušaju te navodi korisnika na upis valjanog ID-a
 2. Korisnik upisuje ispravan ID ili odustaje od pretraživanja

UC2 - Pretraživanje i pregled kontejnera odabirom na karti/putem adrese

- **Glavni sudionik:** Svi korisnici
- **Cilj:** Pregledati kontejnere i njihovo stanje kroz vremenski period
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Karta je prikazana prilikom učitavanja aplikacije
 2. Korisnik na karti odabire kontejner
 3. Prikazuju se informacije o kontejneru

UC3 - Registracija

- **Glavni sudionik:** Neregistrirani korisnici
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisničke podatke

3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 3.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC4 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnici, zaposlenici, direktor
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke
 2. Potvrda o ispravnosti unesenih podataka
 3. Pristup funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime ili lozinka
 1. Sustav obavještava korisnike o neuspješnom upisu
 2. Korisnik upisuje ispravne podatke ili odustaje od prijave

UC5 - Pregled osobnih podataka

- **Glavni sudionik:** Registrirani korisnici, zaposlenici, direktor
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Osobni podatci"
 2. Prikazuju se osobni podatci korisnika

UC6 - Objava slike kontejnera

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Objaviti trenutno stanje kontejnera
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabere kontejner
 2. Korisnik učitava sliku kontejnera
 3. Korisnik objavljuje sliku kontejnera
- **Opis mogućih odstupanja:**
 - 2.a Korisnik učitava sliku, ali ju ne objavi
 1. Sustav obavještava korisnika da slika nije objavljena prije obustavljanja radnje

UC7 - Ocjenjivanje kontejnera

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Ocijeniti trenutno stanje kontejnera
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabere kontejner
 2. Korisnik daje ocjenu kontejneru

UC8 - Komentiranje kontejnera

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Komentirati stanje kontejnera
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabere kontejner
 2. Korisnik piše komentar
 3. Korisnik objavljuje komentar
- **Opis mogućih odstupanja:**
 - 2.a Korisnik napiše komentar, ali ga ne objavi
 1. Sustav obavještava korisnika da komentar nije objavljen prije obustavljanja radnje

UC9 - Dodavanje korisnika

- **Glavni sudionik:** Direktor
- **Cilj:** Dodati novog korisnika

- **Sudionici:** Baza podataka
- **Preduvjet:** Direktor je prijavljen
- **Opis osnovnog tijeka:**
 1. Direktor u aplikaciji odabire opciju dodavanja novog korisnika
 2. Direktor upisuje ime, prezime, ID novog korisnika (i ostale podatke koji idu u bazu podataka)
 3. Direktor potvrđuje svoj unos
- **Opis mogućih odstupanja:**
 - 2.a Korisnik već postoji u bazi
 1. Sustav obavještava direktora o tome da već postoji korisnik s tim ID-om

UC10 - Brisanje korisnika

- **Glavni sudionik:** Direktor
- **Cilj:** Izbrisati postojećeg korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Direktor je prijavljen
- **Opis osnovnog tijeka:**
 1. Direktor u aplikaciji pretražuje korisnike
 2. Direktor odabire opciju brisanja korisnika
 3. Direktor potvrđuje svoj izbor
- **Opis mogućih odstupanja:**
 - 2.a Ne postoji korisnik koji zadovoljava uvjete pretrage
 1. Sustav ispisuje poruku da korisnik nije pronađen

UC11 - Dodavanje kontejnera

- **Glavni sudionik:** Direktor
- **Cilj:** Dodati novi kontejner
- **Sudionici:** Baza podataka
- **Preduvjet:** Direktor je prijavljen
- **Opis osnovnog tijeka:**
 1. Direktor u aplikaciji odabire opciju dodavanja novog kontejnera
 2. Direktor upisuje adresu i ID novog kontejnera
 3. Direktor potvrđuje svoj unos
- **Opis mogućih odstupanja:**
 - 2.a Na odabranoj adresi već postoji kontejner

1. Pitati direktora želi li postaviti još jedan kontejner na tu adresu
 2. Direktor odabire opciju da ili ne
- 2.b Već postoji kontejner s tim ID-om
1. Ispisati poruku da već postoji kontejner s tim ID-om

UC12 - Brisanje kontejnera

- **Glavni sudionik:** Direktor
- **Cilj:** Dodati novog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Direktor je prijavljen
- **Opis osnovnog tijeka:**
 1. Direktor u aplikaciji pretražuje kontejner po ID-u ili ulici i odabire opciju za brisanje kontejnera
 2. Direktor potvrđuje svoj unos
- **Opis mogućih odstupanja:**
 - 2.a Kontejner s traženim ID-em ne postoji
 1. Ispisati poruku da traženi kontejner nije pronađen
 - 2.b U traženoj ulici nema niti jednog kontejnera
 1. Ispisati da u traženoj ulici nema niti jednog kontejnera

UC13 - Premještanje kontejnera

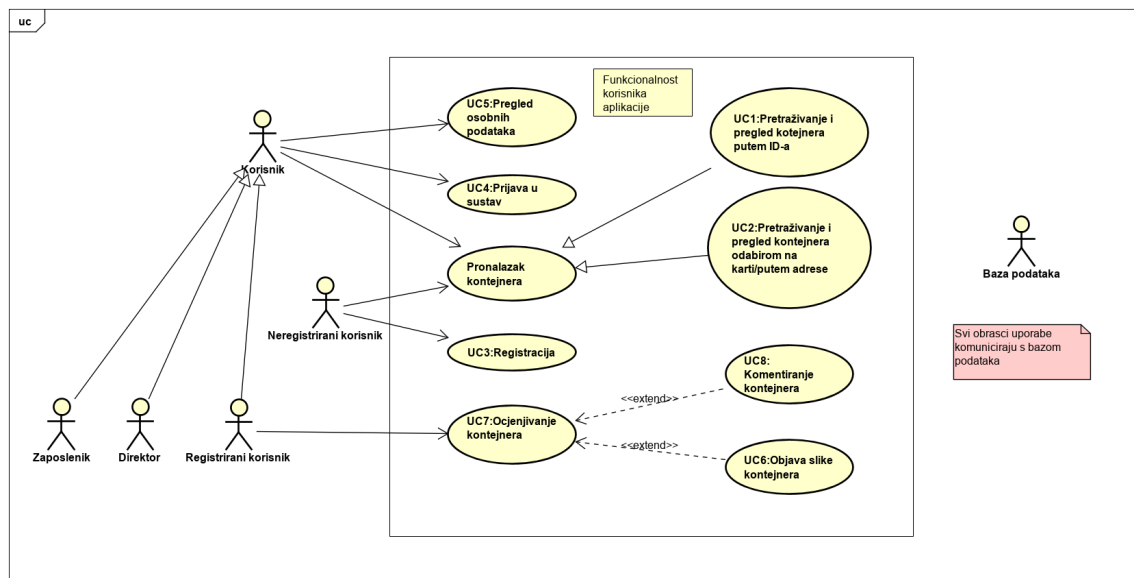
- **Glavni sudionik:** Direktor
- **Cilj:** Premjestiti kontejner s jedne lokacije na drugu
- **Sudionici:** Baza podataka
- **Preduvjet:** Direktor je prijavljen
- **Opis osnovnog tijeka:**
 1. Direktor u aplikaciji pretražuje kontejner po ID-u ili ulici i odabire opciju za premještanje kontejnera
 2. Direktor upisuje adresu gdje želi da kontejner bude premješten
 3. Direktor potvrđuje svoj unos
- **Opis mogućih odstupanja:**
 - 2.a Traženi ID kontejnera ne postoji
 1. Ispisati poruku da traženi kontejner nije pronađen
 - 2.b U traženoj ulici nema niti jednog kontejnera
 1. Ispisati da u traženoj ulici nema niti jednog kontejnera
 - 2.c Na odabranoj adresi već postoji kontejner

1. Pitati direktora želi li postaviti još jedan kontejner na tu adresu
2. Direktor odabire opciju da ili ne

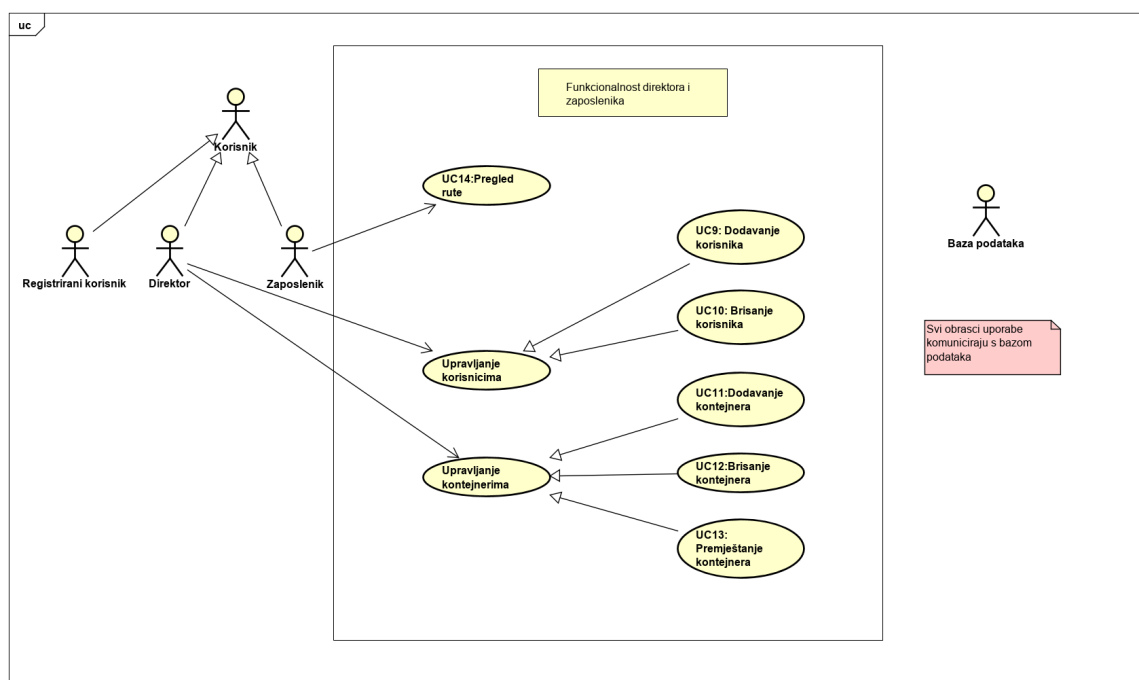
UC14 - Pregled rute

- **Glavni sudionik:** Zaposlenici
- **Cilj:** Pregledati rutu uz prikazivanje svih kontejnera na ruti
- **Sudionici:** Baza podataka
- **Preduvjet:** Zaposlenik je prijavljen
- **Opis osnovnog tijeka:**
 1. Zaposlenik pritisne na gumb moja ruta
 2. Prikazuju mu se informacije ruti zaposlenika
- **Opis mogućih odstupanja:**
 - 2.a Neispravna ili nepostojeća ruta
 1. Sustav obavještava zaposlenika o nepostojećoj ruti

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe, funkcionalnost korisnika aplikacije

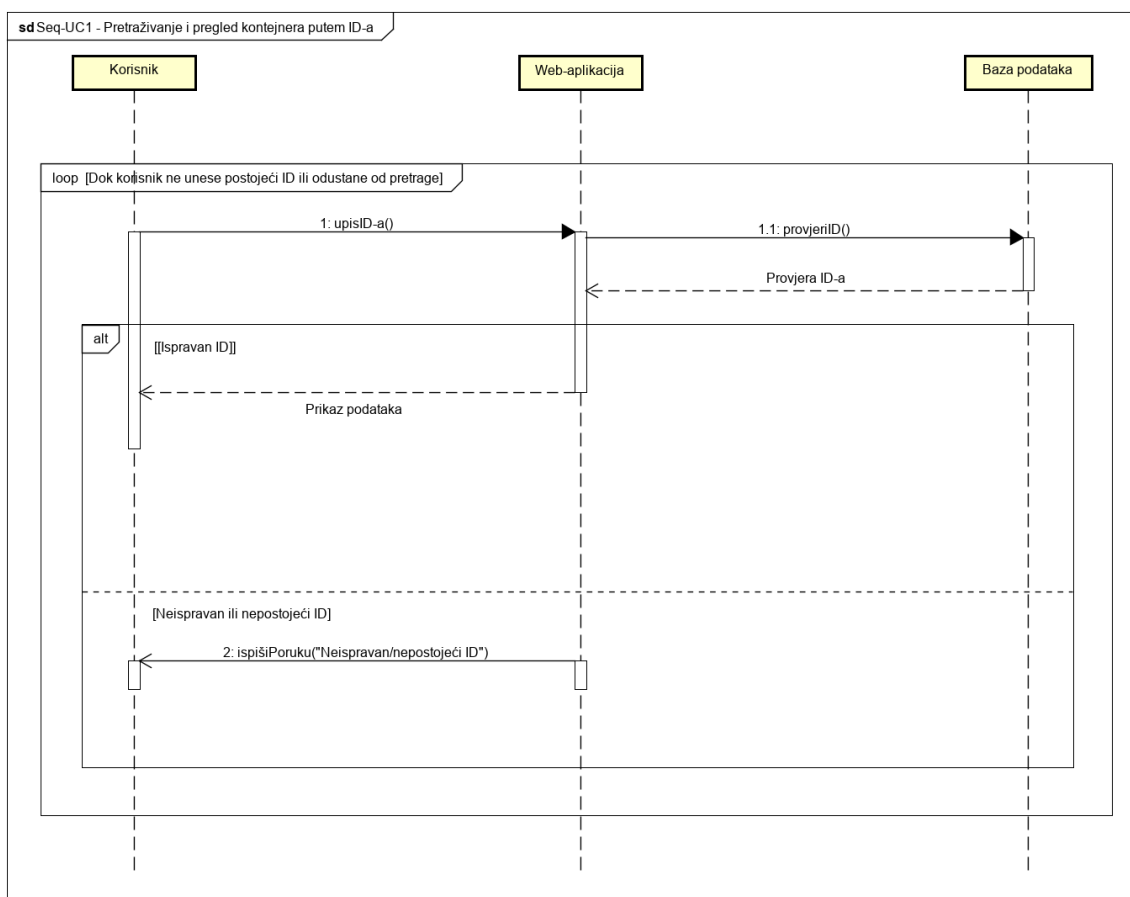


Slika 3.2: Dijagram obrazaca uporabe, funkcionalnost direktora i zaposlenika

3.1.2 Sekvencijski dijagrami

UC1 - Pretraživanje i pregled kontejnera putem ID-a

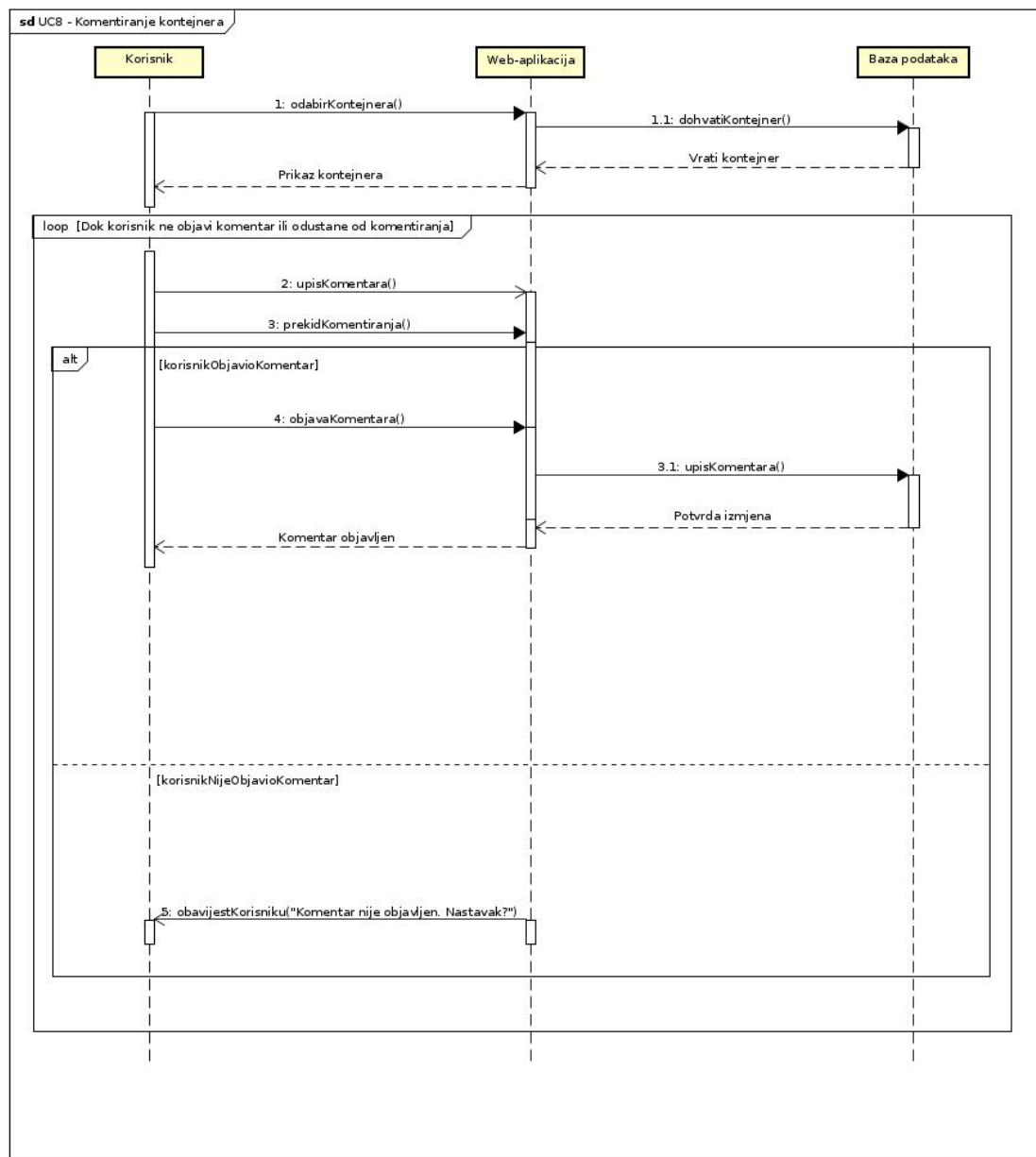
Korisnik upisuje ID kontejnera za koji želi pregledati podatke. Poslužitelj prosledjuje upisani ID bazi koja provjerava je li ID ispravan te postoji li kontejner s tim ID-om. Ako postoji, poslužitelj dohvaća podatke o kontejneru i prikazuje ih korisniku. Ako je ID neispravan ili kontejner s tim ID-om ne postoji u bazi, sustav obavještava korisnika te ga navodi na ispravan unos ID-a. Proces se ponavlja dok korisnik ne unese ispravan ID za koji postoji kontejner ili dok ne odustane od pretraživanja.



Slika 3.3: Sekvencijski dijagram za UC1

UC8 - Komentiranje kontejnera

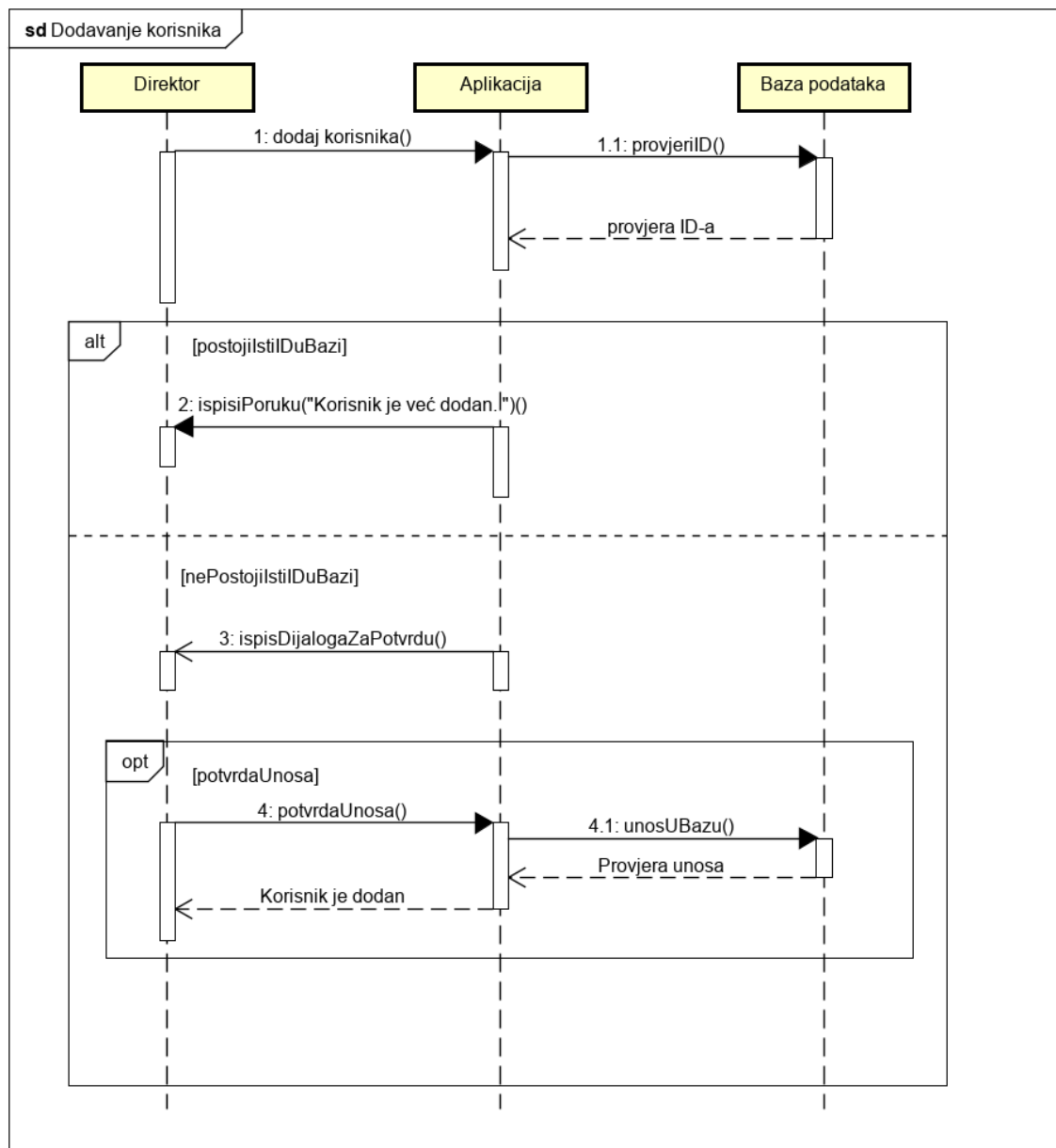
Korisnik odabire kontejner koji želi komentirati. Poslužitelj dohvaća i prikazuje željeni kontejner. Korisnik počinje pisati komentar i u jednom trenutku prekida s komentiranjem. Ako je korisnik objavio komentar, poslužitelj ga upisuje u bazu te korisniku šalje potvrdu o uspješnoj objavi. Ako korisnik prekine komentiranje, a nije objavio komentar, sustav ga o tome obavještava i upita ga želi li nastaviti s komentiranjem. Proces se ponavlja dok korisnik ne objavi komentar ili dok ne odustane od komentiranja.



Slika 3.4: Sekvencijski dijagram za UC8

Obrazac uporabe UC9 - Dodavanje korisnika

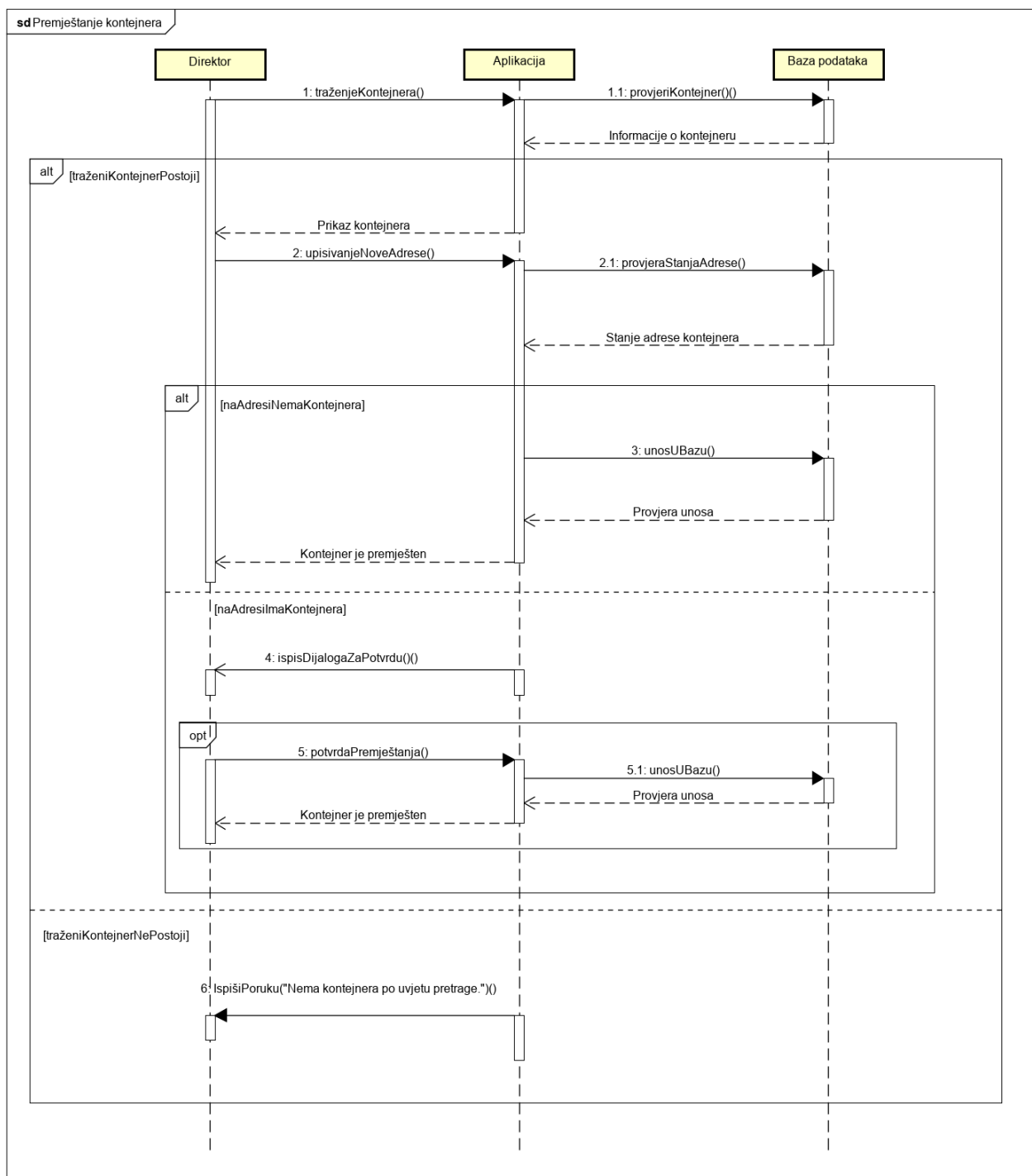
Direktor želi dodati novog korisnika. U aplikaciji bira opciju za unos korisnika, gdje predaje sve njegove podatke, što uključuje i njegov ID. Nakon toga aplikacija njegov ID prosljeđuje do baze podataka gdje se provjerava postoji li već korisnik s tim ID-em. Ukoliko ne postoji, Direktor dobiva opciju za potvrdu unosa te ako izabere tu opciju, novi se korisnik unosi u bazu. Ako korisnik s tim ID-em već postoji, ispisuje se odgovarajuća poruka. Ako Direktor nije potvrdio unos, postupak će se prekinuti.



Slika 3.5: Sekvencijski dijagram za UC9

Obrazac uporabe UC13 - Premještanje kontejnera

Direktor želi premjestiti određeni kontejner. U aplikaciji traži kontejner po ID-u ili ulici i odabire opciju za premještanje tog kontejnera. Provjerava se postoji li kontejner s traženim ID-em ili u traženoj ulici. Nakon toga Direktor odabire adresu na koju će kontejner biti premješten. Ako na toj adresi već postoji kontejner onda se treba prikazati opcija kojom Direktor potvrđuje namjeru da se 2 kontejnera nalaze na toj adresi. Nakon toga mijenja se adresa kontejnera u bazi podataka. Ako kontejner ne postoji, ispisuje se odgovarajuća poruka. Ako Direktor nije potvrdio unos, postupak će se prekinuti.



Slika 3.6: Sekvencijski dijagram za UC13

3.2 Ostali zahtjevi

- Sustav treba biti izveden kao responzivna web aplikacija prilagođena mobilnim uređajima, te se prikladno skalirati za ekrane računala
 - Sustav treba omogućiti rad više korisnika u stvarnom vremenu
 - Pogreške pri izvođenju se ne smiju direktno prikazivati korisniku
 - Sustav treba provjeravati ispravnost unesenih podataka, te ne smije dopustiti korisniku da unese podatke koji bi narušili funkcionalnost sustava
 - Korisničko sučelje sustava mora biti na hrvatskom jeziku, te podržavati hrvatsku abecedu pri unosu i prikazu podataka
 - Korisničko sučelje mora biti intuitivno i jednostavno za korištenje
 - Korisniku treba omogućiti pronalazak kontejnera u manje od 10 akcija
 - Učitavanje stranice ne smije trajati duže od 6 sekundi
-
- Korisničko ime i e-mail adresa korisnika moraju biti jedinstveni
 - Lozinke se ne smiju spremati u bazu podataka kao nešifrirani tekst (engl. *plaintext*), nego moraju biti šifrirane algoritmom raspršivanja
 - Nakon registracije ili promjene e-mail adrese, potrebno je potvrditi vlasništvo e-mail adrese
 - E-mail obavijesti (osim slanja upita za potvrdu e-maila) se ne smiju slati na nepotvrđene e-mail adrese.

4. Arhitektura i dizajn sustava

Glavni prioriteti kod izgradnje sustava i odabira arhitekture su nam jednostavnost korištenja sučelja za sve korisnike (osobito za građane), paralelni rad više korisnika u stvarnom vremenu te korištenje na različitim uređajima (prvenstveno na mobilnim). Osim toga, građani najvjerojatnije neće biti voljni instalirati dodatnu programsku podršku, već će se htjeti koristiti onim što već imaju, a to je standardni web preglednik. Zbog navedenih zahtjeva, odlučili smo se za arhitekturu responzivne web aplikacije.

Arhitekturu možemo podijeliti na tri osnovna podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Web preglednik je program koji omogućuje korisnicima pregled web-stranica i sadržaja vezanih uz njih. Web preglednici su prevoditelji, tj. stranice su pisane u kodu koji preglednici nakon toga interpretiraju u nešto razumljivo. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj je osnova rada web aplikacije. Glavna mu je zadaća komunikacija klijenta s aplikacijom. Komunikacija se vrši putem HTTP (engl. Hyper Text Transfer Protocol) protokola, glavne i najčešće metode prijenosa informacija na Webu. Osnovna namjena ovog protokola je omogućavanje objavljivanja i prezentacije HTML dokumenata, tj. web stranica. Poslužitelj pokreće web aplikaciju te joj prosljeđuje zahtjev.

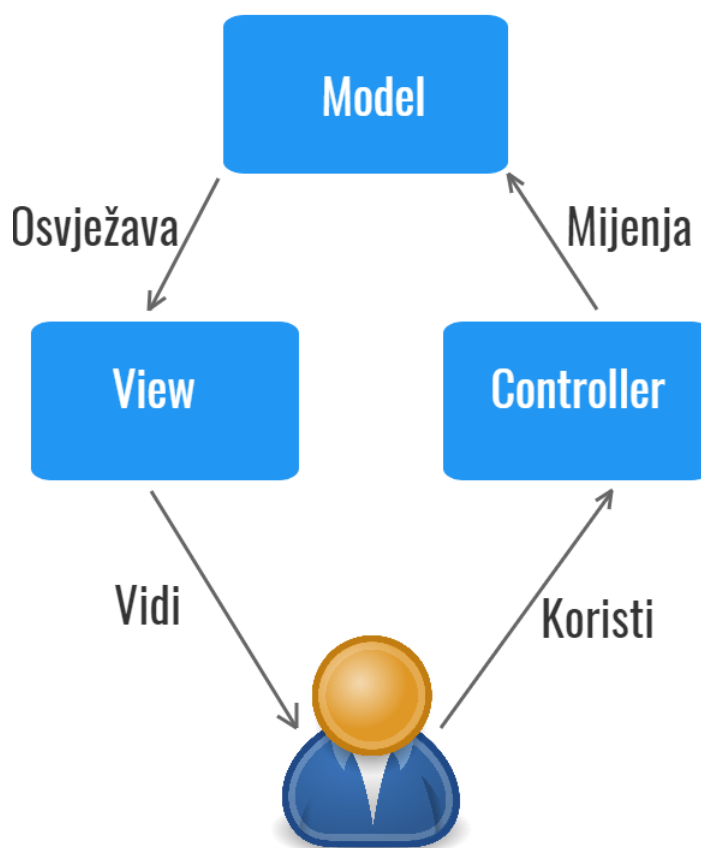
Korisnik koristi **web aplikaciju** za obrađivanje željenih zahtijeva. Web aplikacija obrađuje zahtjev, pristupa bazi podataka te preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku.

Programski jezici koje smo odabrali za izradu naše web aplikacije su Java zajedno sa Spring Boot radnim okvirom za izradu backenda te React i jezik Javascript za izradu frontenda.

Arhitektura sustava razvijat će se prema **MVC** (Model-View-Controller) konceptu. MVC odvaja korisničko sučelje od ostatka sustava, pogodan je za uporabu razinske kohezije te smanjuje međuovisnost između U/I sučelja i ostatka sustava. Posljedica toga je jednostavnije ispitivanje i razvijanje i dodavanje novih svojstava u sustav.

MVC se sastoji od:

- **Model**
- **View**
- **Controller**



Slika 4.2: MVC skica

Model čuva podatke web aplikacije, logiku i pravila aplikacije te je centralna komponenta sustava.

Controller prima ulaze i prilagođava ih za prosljeđivanje Modelu ili Viewu. Nadalje, upravlja korisničkim zahtjevima te pomoću njih izvodi daljnju komunikaciju i interakciju s ostalim elementima sustava.

View predstavlja bilo kakav prikaz podataka te su mogući različiti prikazi istih.

Korisnik koristi Controller te vidi View. Controller mijenja Model, a model osvježava View.

4.1 Baza podataka

Aplikacija koristi SQL relacijsku bazu podataka. Ona se sastoji od relacija (tablica) koje se sastoje od atributa. Razlog uporabe relacijske baze podataka je olakšavanje modeliranja entiteta i događaja iz stvarnog svijeta. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Kontejner
- Komunalna služba
- Pretplata
- Recenzija
- Senzor
- Zona
- Grad
- Država

4.1.1 Opis tablica

Korisnik. Ovaj entitet sadrži informacije o korisniku aplikacije. Uključeni atributi su ID korisnika, ID komunalne službe u kojoj je korisnik zaposlen (ako je zaposlen u komunalnoj službi), korisničko ime, e-mail adresa, informacije je li e-mail adresa potvrđena te lozinka. Ovaj entitet u vezi je *One-to-Many* s entitetom Komunalna služba preko ID-a korisnika te u vezi *Many-to-One* s entitetom Komunalna služba preko ID-a komunalne službe u kojoj je korisnik zaposlen.

Korisnik		
IDkorisnik	INTEGER	Jedinstveni identifikator korisnika
IDsluzba	INTEGER	Jedinstveni identifikator komunalne službe
korisnickoIme	VARCHAR	Korisničko ime
email	VARCHAR	E-mail adresa korisnika
emailPotvrda	BOOLEAN	Stanje potvrđenosti e-mail adrese
lozinka	VARCHAR	Hash lozinke

Kontejner. Ovaj entitet sadrži informacije o kontejneru. Uključeni atributi su ID kontejnera, ID komunalne službe, ID zone u kojoj se kontejner nalazi, adresa kontejnera, geografska dužina te geografska širina kontejnera. Ovaj entitet u vezi

je *One-to-Many* s entitetom Pretplata, Recenzija i Senzor preko ID-a kontejnera, *Many-to-One* s entitetom Komunalna služba preko ID-a komunalne službe te u vezi *Many-to-One* s entitetom Zona preko ID-a zone.

Kontejner		
IDkontejner	INTEGER	Jedinstveni identifikator kontejnera
IDsluzba	INTEGER	Jedinstveni identifikator komunalne službe
IDzona	INTEGER	Jedinstveni identifikator gradske zone
adresa	VARCHAR	Adresa kontejnera
geoDuzina	NUMERIC	Geografska dužina kontejnera
geoSirina	NUMERIC	Geografska širina kontejnera

Komunalna služba. Ovaj entitet sadrži informacije o komunalnoj službi. Uključeni atributi su ID komunalne službe, ID direktora komunalne službe, ime komunalne službe te opis komunalne službe. Ovaj entitet u vezi je *One-to-Many* s entitetom Korisnik i Kontejner preko ID-a komunalne službe te u vezi *Many-to-One* s entitetom Korisnik preko ID-a direktora komunalne službe.

Komunalna služba		
IDsluzba	INTEGER	Jedinstveni identifikator komunalne službe
IDdirektor	INTEGER	Jedinstveni identifikator direktora komunalne službe
imeSluzba	VARCHAR	Ime komunalne službe
opisSluzba	VARCHAR	Opis komunalne službe

Pretplata. Ovaj entitet sadrži informacije o korisničkim pretplatama na kontejnere. Uključeni atributi su ID korisnika i ID kontejnera. Ovaj entitet u vezi je *Many-to-One* s entitetom Korisnik preko ID-a korisnika te u vezi *Many-to-One* s entitetom Kontejner preko ID-a kontejnera.

Pretplata		
IDkorisnik	INTEGER	Jedinstveni identifikator komunalne službe
IDkontejner	INTEGER	Jedinstveni identifikator direktora komunalne službe

Recenzija. Ovaj entitet sadrži informacije o recenzijama kontejnera. Uključeni atributi su ID recenzije, ID korisnika, ID kontejnera, ocjena punoće, ocjena

urednosti, komentar, slika te vrijeme objave recenzije. Ovaj entitet u vezi je *Many-to-One* s entitetom Korisnik preko ID-a korisnika te u vezi *Many-to-One* s entitetom Kontejner preko ID-a kontejnera.

Recenzija		
IDrecenzija	INTEGER	Jedinstveni identifikator recenzije
IDkorisnik	INTEGER	Jedinstveni identifikator korisnika
IDkontejner	INTEGER	Jedinstveni identifikator kontejnera
ocjenaPunoce	INTEGER	Ocjena punoće kontejnera
ocjenaUrednosti	INTEGER	Ocjena urednosti kontejnera
komentar	VARCHAR	Komentar na stanje kontejnera u trenutku objavljivanja recenzije
slika	LONGBLOB	Slika kontejnera u trenutku objavljivanja recenzije
vrijemeObjave	TIMESTAMP	Vrijeme u trenutku objavljivanja recenzije

Senzor. Ovaj entitet sadrži informacije o senzoru. Uključeni atributi su ID senzora, ID kontejnera te token. Ovaj entitet u vezi je *Many-to-One* s entitetom Kontejner preko ID-a kontejnera.

Senzor		
IDSenzor	INTEGER	Jedinstveni identifikator senzora
IDkontejner	INTEGER	Jedinstveni identifikator kontejnera
token	VARCHAR	Token senzora

Zona. Ovaj entitet sadrži informacije o gradskoj zoni u kojoj je smješten kontejner. Uključeni atributi su ID zone, ID grada te ime zone. Ovaj entitet u vezi je *One-to-Many* s entitetom Kontejner preko ID-a zone te u vezi *Many-to-One* s entitetom Grad preko ID-a grada.

Zona		
IDzona	INTEGER	Jedinstveni identifikator gradske zone
IDgrad	INTEGER	Jedinstveni identifikator grada
imeZona	VARCHAR	Ime zone

Grad. Ovaj entitet sadrži informacije o gradu u kojoj je smješten kontejner. Uključeni atributi su ID grada, ID države te ime grada. Ovaj entitet u vezi je *One-*

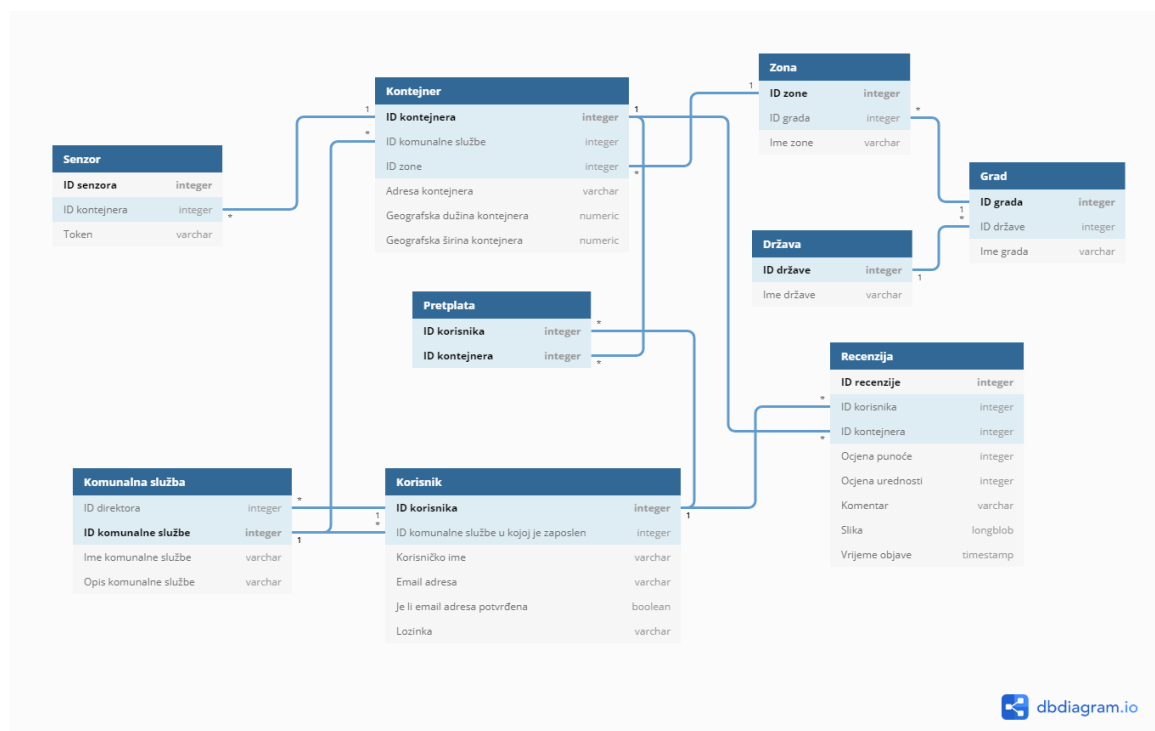
to-Many s entitetom Zona preko ID-a grada te u vezi *Many-to-One* s entitetom Država preko ID-a države.

Grad		
IDgrad	INTEGER	Jedinstveni identifikator grada
IDdrzava	INTEGER	Jedinstveni identifikator države
imeGrad	VARCHAR	Ime grada

Država. Ovaj entitet sadrži informacije o državi u kojoj je smješten kontejner. Uključeni atributi su ID države te ime države. Ovaj entitet u vezi je *One-to-Many* s entitetom Grad preko ID-a države.

Država		
IDdrzava	INTEGER	Jedinstveni identifikator države
imeDrzava	VARCHAR	Ime države

4.1.2 Dijagram baze podataka

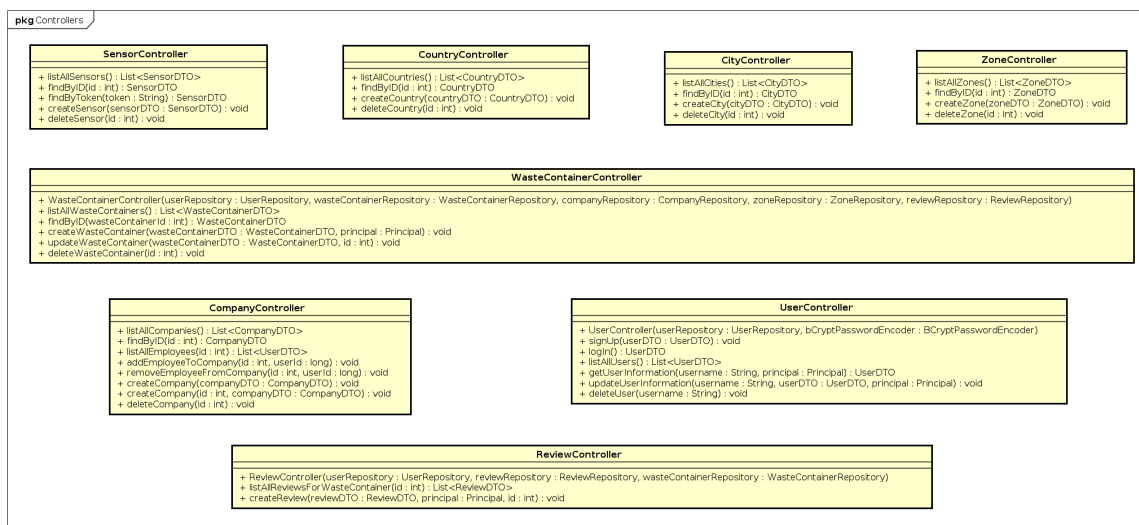


Slika 4.3: Dijagram baze podataka

Dijagram baze podataka je izrađen alatom dbdiagram.io. Svjetlo plavo su označeni strani ključevi, a primarni ključevi su podebljani.

4.2 Dijagram razreda

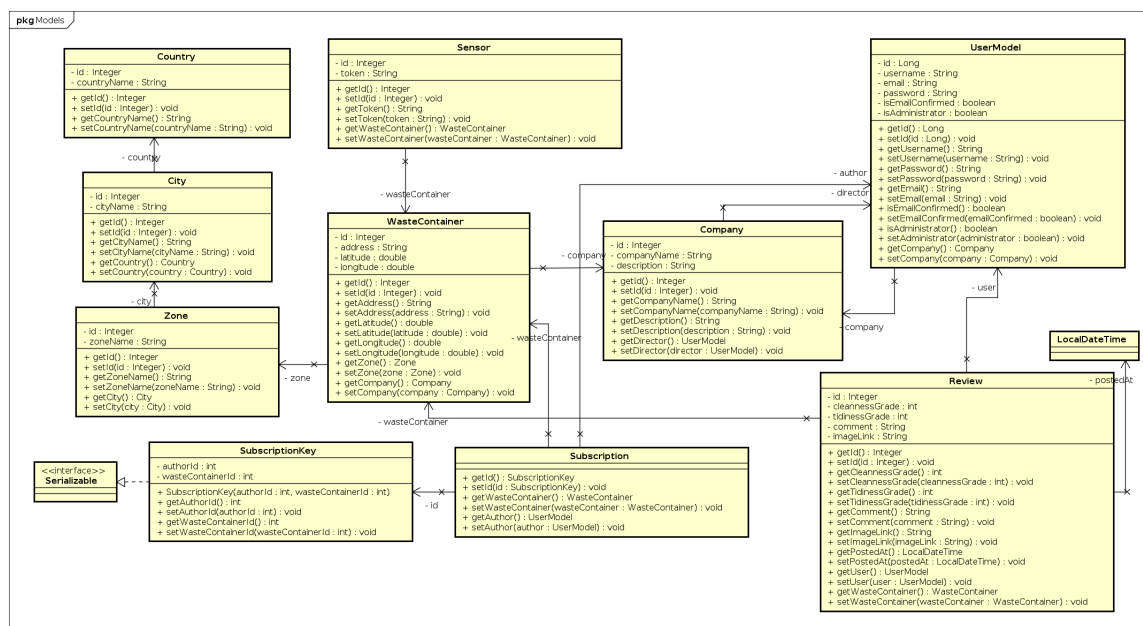
Na slikama 4.4, 4.5 i 4.6 su prikazani razredi *backenda*. Razredi su podijeljeni u dijagrame prema njihovim odgovornostima: controller razredi odgovaraju na upite *frontenda* te koriste razrede models za komunikaciju s bazom podataka i razrede DTO (engl. *Data Transfer Object*) za komunikaciju s *frontendom* u JSON formatu. Radi preglednosti na dijagramima razreda su prikazane samo veze između razreda prikazanih na istom dijagramu.



Slika 4.4: Dijagram razreda - dio Controllers



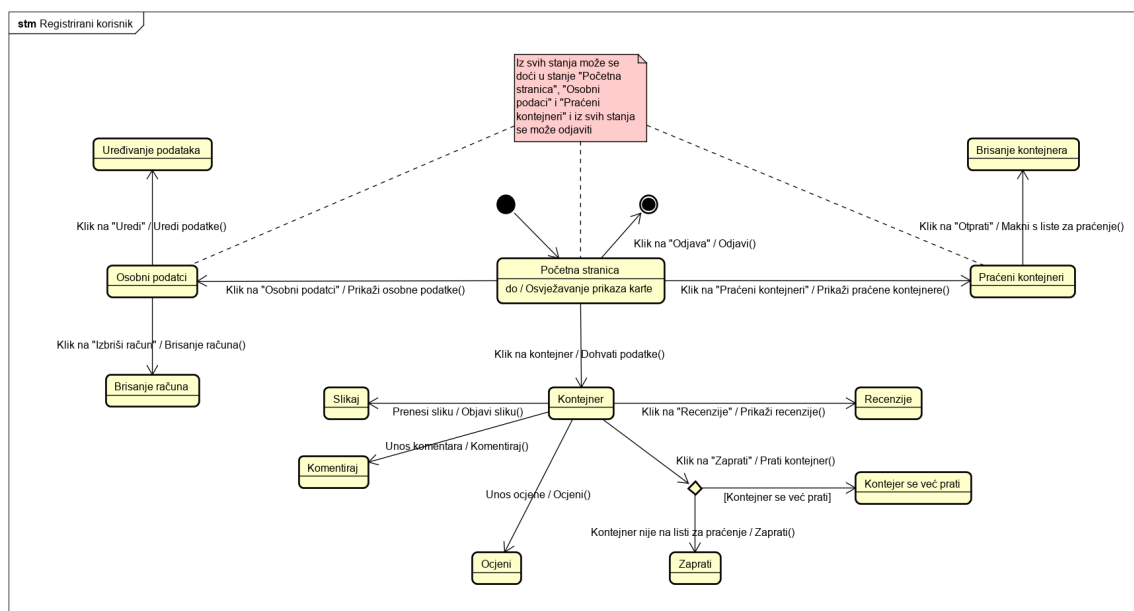
Slika 4.5: Dijagram razreda - dio DTO



Slika 4.6: Dijagram razreda - dio Models

4.3 Dijagram stanja

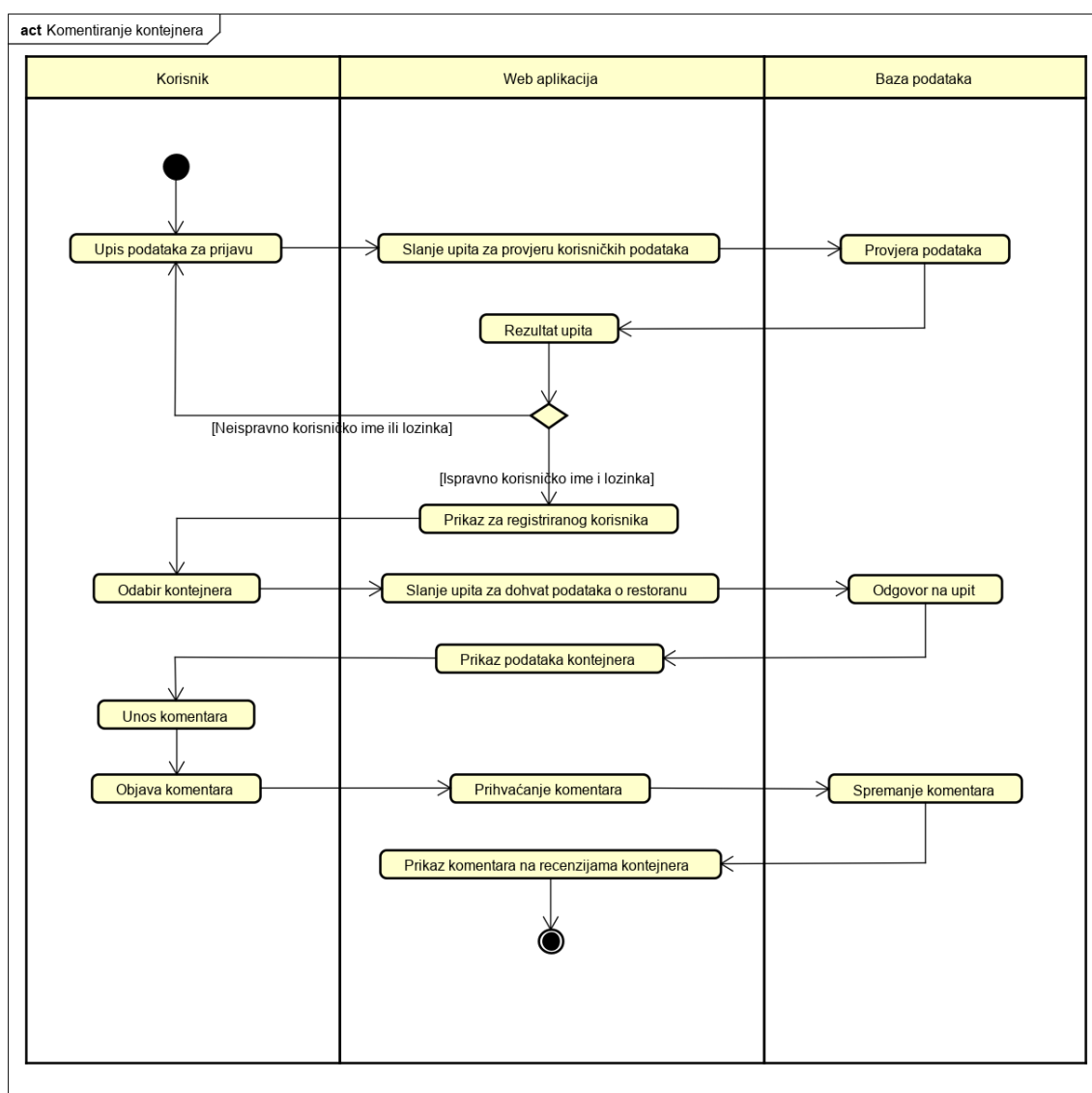
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.6 prikazan je dijagram stanja za registriranog korisnika - građanina. Nakon prijave, korisniku se prikazuje početna stranica na kojoj može pregledati kartu i kontejnere. Za odabrani kontejner ima opciju pregledavanja stanja (slika, komentara, ocjene urednosti), opciju za praćenje kontejnera te dodavanje vlastitog komentara, slike i ocjenjivanje kontejnera. Klikom na "Osobni podatci" prikazuju mu se njegovi podatci, koje može urediti ili odabrati opciju brisanja računa. Odabirom "Praćeni kontejneri" korisniku se otvara popis praćenih kontejnera koje po potrebi može obrisati.



Slika 4.7: Dijagram stanja

4.4 Dijagram aktivnosti

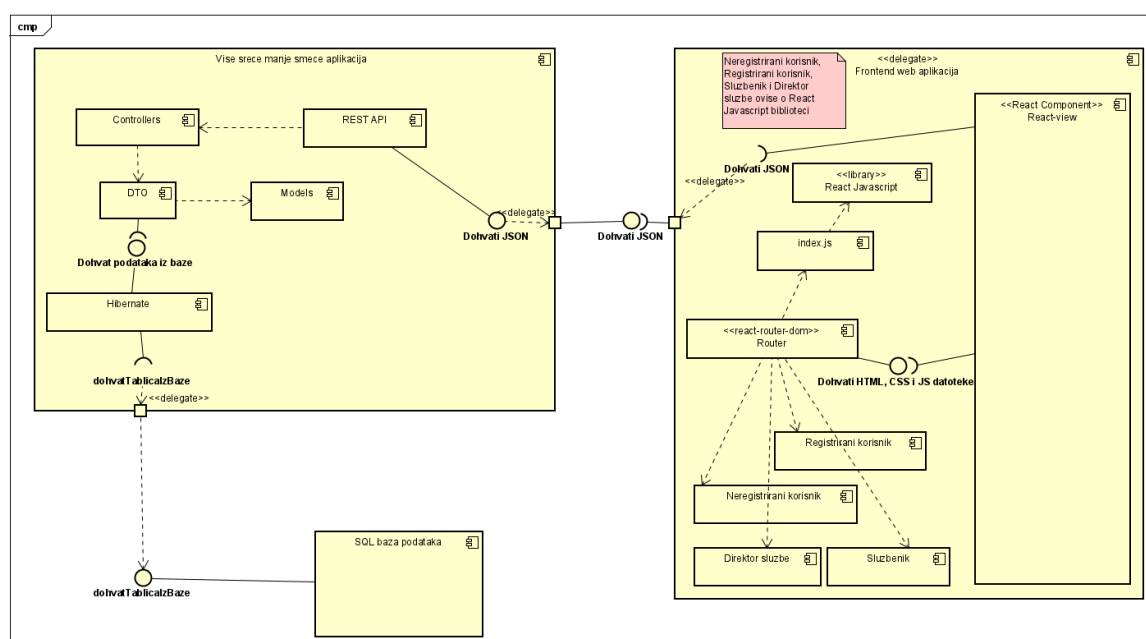
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti 4.7 prikazan je proces komentiranja stanja kontejnera. Korisnik se prijavi u sustav, odabere kontejner za koji želi ostaviti komentar, unese komentar. Kada je završio s pisanjem komentara, isti objavljuje te taj komentar postaje javan u recenzijama kontejnera.



Slika 4.8: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazan na sljedećoj stranici opisuje organizaciju i međusobnu ovisnost komponenti, interne strukture i odnose prema okolini. Sustavu se pristupa preko dva različita sučelja: sučelja za dohvat HTML, CSS i JS datoteka, preko kojega se poslužuju datoteke koje pripadaju *frontend* dijelu aplikacije te sučelja za dohvat JSON podataka, preko kojega se pristupa REST API komponenti. Router je komponenta koja na upit s url određuje koja će se datoteka poslužiti na sučelje. *Frontend* dio sastoji se od niza JavaScript datoteka koje su raspoređene u logičke cjeline nazvane po tipovima aktera koji im pristupaju. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente (gumbe, forme, i sl.). REST API poslužuje podatke koji pripadaju *backend* dijelu aplikacije. Hibernate je zadužen za dohvaćanje tablica iz baze podataka koristeći SQL upite. Ti podaci se dalje šalju MVC arhitekturi u obliku DTO (*Data Transfer Object*) React-view komponenta komunicira preko dostupnih sučelja s Više Sreće Manje Smeće aplikacijom te osvježava prikaz i dohvaća nove podatke ili datoteke ovisno o korisnikovim akcijama.



Slika 4.9: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacije Slack¹. Slack omogućuje laku podjelu u podskupine, brz način komuniciranja kao i mogućnost povezivanja s web platformom na kojoj se nalazi projekt kako bi se u svakom trenutku moglo vidjeti nove promjene na projektu. Za izradu UML dijagrama korišten je alat Astah Professional² koji služi za vizualizaciju ideja i stvaranje dijagrama koji pomažu pri lakšem razvoju i razumijevanju projektnog zadatka. Za sustav za upravljanje izvornim kodom korišten je Git³ koji je najpopularniji sustav za upravljanje izvornim kodom na svijetu. Udaljeni repozitorij projekta dostupan je na web platformi Gitlab⁴ koja služi za upravljanje git repozitorijima na centraliziranom serveru. Gitlab omogućuje potpunu kontrolu nad repozitorijima i članovi tima mogu odlučiti hoće li ti repozitoriji biti javno dostupni ili ne.

Kao razvojno okruženje korišten je Eclipse⁵ - integrirano razvojno okruženje (IDE) tvrtke Eclipse Foundation. Služi prvenstveno za razvoj aplikacija u Javi, ali aplikacije se mogu raditi i u drugim programskim jezicima pomoću plug-ina.

Aplikacija je napisana koristeći radni okvir Spring Boot⁶ i jezik Java⁷ za izradu backenda. Radni okvir Spring Boot ima mikro servis arhitekturu koja omogućuje da svaki servis ima svoj vlastiti proces što je vrlo korisno pri razvoju aplikacija. Java je brz, siguran i pouzdan objektno orijentirani programski jezik kojeg je dizajnirala tvrtka Sun Microsystems. Za izradu frontenda korišteni su React⁸ i jezik

¹<https://slack.com/>

²<http://astah.net/editions/professional>

³<https://git-scm.com/>

⁴<https://gitlab.com/>

⁵<https://www.eclipse.org/>

⁶<https://spring.io/projects/spring-boot>

⁷<https://www.java.com/en/>

⁸<https://reactjs.org/>

JavaScript⁹. React je biblioteka u JavaScriptu za izgradnju korisničkih sučelja. Posjeduje ga tvrtka Facebook koja ga također i održava, a najčešće koristi kao osnova u razvoju web ili mobilnih aplikacija. Javascript je skriptno programski jezik koji se izvršava u web pregledniku na strani korisnika.

Baza podataka nalazi se na poslužitelju u oblaku Heroku¹⁰. Heroku je cloud platforma koja omogućuje tvrtkama da izgrađuju, dostavljaju, nadziru i skaliraju aplikacije.

⁹<https://www.javascript.com/>

¹⁰<https://www.heroku.com/>

5.2 Ispitivanje programskog rješenja

Sustav se ispituje jediničnim testovima (engl. *unit tests*) i integracijskim testovima (engl. *integration tests*). Jediničnim testovima se testiraju komponente backenda, dok se integracijskim testovima testira frontend i obrasci uporabe.

U sljedeća dva potpoglavlja razloženi su ispitni slučajevi za ispitivanje komponenti i cjelokupnog sustava. Cilj takvog ispitivanja je ranije otkrivanje i jednostavnije ispravljanje neispravnih i nepoželjnih ponašanja aplikacije te kao provjera zadovoljava li aplikacija obrasce uporabe.

5.2.1 Ispitivanje komponenti

Ispitivanje komponenti se kontinuirano prati koristeći GitLab CI/CD¹¹. U nastavku su prikazani ispitni slučajevi komponenti koji se prate, njihovi očekivani i trenutni rezultati, a na kraju potpoglavlja je prikazan rezultat izvođenja ispita u razvojnom okruženju JUnit¹².

Ispitni slučaj 1: Registracija korisnika sprema korisnika u bazu podataka

Ulaz:

1. Pozivanje metode za registraciju korisnika s ispravnim podacima.

Očekivani rezultat:

- 1.a Korisnik je spremljen u bazu podataka.

Izvorni kod:

```
1 @Test
2 void testRegistrationSavesUser() {
3     UserDTO firstUser = new UserDTO("username", "email@email.com", "password");
4
5     userController.signUp(firstUser);
6
7     // User was saved
8     verify(userRepository, times(1))
9         .save(any());
10 }
```

Programski isječak 1: Izvorni kod za ispitni slučaj 1

Rezultat: Očekivani rezultat je zadovoljen. Registracija korisnika s ispravnim podacima je uspješno spremila tog korisnika u bazu podataka.

¹¹<https://docs.gitlab.com/ee/ci/>

¹²<https://junit.org/>

Ispitni slučaj 2: Registracija dva korisnika s istom e-mail adresom**Ulaz:**

1. Pozivanje metode za registraciju korisnika.
2. Pozivanje metode za registraciju korisnika s istom e-mail adresom.

Očekivani rezultat:

- 1.a Korisnik je spremljen u bazu podataka.
- 2.a Izazvana je iznimka zbog duplicirane e-mail adrese.

Izvorni kod:

```
1  @Test
2  void testRegistrationWithDuplicateEmailAddress() {
3      UserDTO firstUser = new UserDTO("username1", "email@email.com"
4                                     "password1");
5      UserDTO secondUser = new UserDTO("username2", "email@email.com",
6                                     "password2");
7
8      userController.signUp(firstUser);
9
10     // First user was saved
11     verify(userRepository, times(1))
12         .save(any());
13
14     // Second user causes an exception
15     assertThrows(Exception.class, () -> userController.signUp(secondUser));
16 }
```

Programski isječak 2: Izvorni kod za ispitni slučaj 2

Rezultat: Očekivani rezultat je zadovoljen. Registracija prvog korisnika uspješno sprema tog korisnika u bazu podataka, dok registracija drugog korisnika rezultira iznimkom.

Ispitni slučaj 3: Registracija dva korisnika s istim korisničkim imenom**Ulaz:**

1. Pozivanje metode za registraciju korisnika.
2. Pozivanje metode za registraciju korisnika s istim korisničkim imenom.

Očekivani rezultat:

- 1.a Korisnik je spremljen u bazu podataka.
- 2.a Izazvana je iznimka zbog dupliciranog korisničkog imena.

Izvorni kod:

```
1  @Test
2  void testRegistrationWithDuplicateUsername() {
3      UserDTO firstUser = new UserDTO("username", "email1@email.com"
4                                     "password1");
5      UserDTO secondUser = new UserDTO("username", "email2@email.com",
6                                     "password2");
7
8      userController.signUp(firstUser);
9
10     // First user was saved
11     verify(userRepository, times(1))
12         .save(any());
13
14     // Second user causes an exception
15     assertThrows(Exception.class, () -> userController.signUp(secondUser));
16 }
```

Programski isječak 3: Izvorni kod za ispitni slučaj 3

Rezultat: Očekivani rezultat je zadovoljen. Registracija prvog korisnika uspješno sprema tog korisnika u bazu podataka, dok registracija drugog korisnika rezultira iznimkom.

Ispitni slučaj 4: Korisnik smije dohvatiti vlastite osobne podatke**Ulaz:**

1. Pozivanje metode za dohvat osobnih podataka za trenutno prijavljenog korisnika.

Očekivani rezultat:

- 1.a Vraćeni su osobni podaci prijavljenog korisnika.

Izvorni kod:

```
1  @Test
2  void testUserCanFetchItsOwnData() {
3      userController.getUserInformation("logged_in_user",
4          () -> "logged_in_user");
5
6      // Verify interaction with database
7      verify(userRepository, times(1))
8          .findByUsername(anyString());
9  }
```

Programski isječak 4: Izvorni kod za ispitni slučaj 4

Rezultat: Očekivani rezultat je zadovoljen. Dohvaćeni su osobni podaci prijavljenog korisnika.

Ispitni slučaj 5: Korisnik ne smije moći dohvatiti osobne podatke drugog korisnika**Ulaz:**

1. Pozivanje metode za dohvat osobnih podataka za korisnika koji nije trenutno prijavljen.

Očekivani rezultat:

- 1.a Izazvana je iznimka zbog pokušaja dohvata tuđih podataka.

Izvorni kod:

```
1  @Test
2  void testUserCannotFetchOthersData() {
3      assertThrows(Exception.class, () ->
4          userController.getUserInformation("different_user",
5              () -> "logged_in_user"));
6
7      // Verify no interaction with database
8      verify(userRepository, times(0))
9          .findByUsername(anyString());
10 }
```

Programski isječak 5: Izvorni kod za ispitni slučaj 5

Rezultat: Očekivani rezultat je zadovoljen. Izazvana je iznimka pri pokušaju dohvata osobnih podataka drugog korisnika.

Ispitni slučaj 6: Dohvat podataka o postojećem kontejneru**Ulaz:**

1. Pozivanje metode za dohvat podataka o kontejneru s valjanim identifikatorom kontejnera.

Očekivani rezultat:

- 1.a Vraćeni su podaci o kontejneru s tim identifikatorom.

Izvorni kod:

```
1  @Test
2  void testFetchingInformationAboutExistingWasteContainer() {
3      wasteContainerController.findById(EXISTING_WASTE_CONTAINER_ID);
4
5      // Verify interaction with database
6      verify(wasteContainerRepository, times(1))
7          .findById(EXISTING_WASTE_CONTAINER_ID);
8  }
```

Programski isječak 6: Izvorni kod za ispitni slučaj 6

Rezultat: Očekivani rezultat je zadovoljen. Dohvaćeni su podaci o kontejneru s tim identifikatorom.

Ispitni slučaj 7: Dohvat podataka o nepostojećem kontejneru**Ulaz:**

1. Pozivanje metode za dohvat podataka o kontejneru s nevaljanim identifikatorom kontejnera.

Očekivani rezultat:

- 1.a Izazvana je iznimka zbog pokušaja dohvata podataka o kontejneru koji ne postoji.

Izvorni kod:

```
1  @Test
2  void testFetchingInformationAboutNonExistantWasteContainer() {
3      assertThrows(Exception.class,
4          () ->
5              wasteContainerController.findById(NON_EXISTANT_WASTE_CONTAINER_ID));
6
7      // Verify interaction with database
8      verify(wasteContainerRepository, times(1))
9          .findById(NON_EXISTANT_WASTE_CONTAINER_ID);
10 }
```

Programski isječak 7: Izvorni kod za ispitni slučaj 7

Rezultat: Očekivani rezultat je zadovoljen. Dohvat podataka o nepostojećem kontejneru je izazvao iznimku.

```
1 [INFO] --- maven-surefire-plugin:2.22.2:test (default-test) @
   ↳ manje-smece-vise-srece-backend ---
2 [INFO]
3 [INFO] -----
4 [INFO]  T E S T S
5 [INFO] -----
6 [INFO] Running
   ↳ hr.fer.opp.bashcrash.manjesmecevisesrece.rest.WasteContainerControllerTest
7 [INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.656 s -
   ↳ in hr.fer.opp.bashcrash.manjesmecevisesrece.rest.WasteContainerControllerTest
8 [INFO] Running hr.fer.opp.bashcrash.manjesmecevisesrece.rest.UserControllerTest
9 [INFO] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.118 s -
   ↳ in hr.fer.opp.bashcrash.manjesmecevisesrece.rest.UserControllerTest
10 [INFO]
11 [INFO] Results:
12 [INFO]
13 [INFO] Tests run: 7, Failures: 0, Errors: 0, Skipped: 0
14 [INFO]
15 [INFO] -----
16 [INFO] BUILD SUCCESS
17 [INFO] -----
18 [INFO] Total time: 6.019 s
19 [INFO] Finished at: 2020-01-02T23:31:26+01:00
20 [INFO] -----
```

Programski isječak 8: Prikaz rezultata izvođenja ispita u razvojnem okruženju

5.2.2 Ispitivanje sustava

Ispitivanje sustava prati se ručno upotrebom Selenium IDE alata integriranog u Google Chrome preglednik¹³. Ispitni slučajevi ispituju obrasce uporabe UC4, UC5, UC6, UC7, UC8, UC11, UC12 i UC13. U nastavku su prikazani koraci ispitivanja, njihovi očekivani i trenutni rezultati te programski isjecci ispitnih slučajeva.

Ispitni slučaj 8: Prijava i odjava korisnika

Ulaz:

1. Otvaranje početne stranice u web pregledniku.
2. Pritisak na ikonu korisnika u gornjem desnom kutku.
3. Odabir opcije "Prijava" iz padajućeg izbornika.
4. Upisivanje korisničkog imena i lozinke te pritisak gumba "Prijava".
5. Pritisak na ikonu korisnika u gornjem desnom kutku.
6. Odabir opcije "Odjava" iz padajućeg izbornika.

Očekivani rezultat:

- 1.a Prikazuje se početna stranica s kartom.
- 1.b Na karti su prikazani kontejneri koji su uneseni u aplikaciju.
2. Prikazuje se padajući izbornik s opcijama "Prijava" i "Registracija".
3. Prikazuje se forma za prijavu korisnika s tekstualnim poljima za unos korisničkog imena i lozinke.
- 4.a Prikazuje se početna stranica s dodatnim gumbima sa strane.
- 4.b Korisnik je uspješno prijavljen.
5. Prikazuje se padajući izbornik s opcijama "Postavke" i "Odjava".
- 6.a Prikazuje se početna stranica bez dodatnih gumbi sa strane.
- 6.b Korisnik je odjavljen.

Izvorni kod:

```
1 //otvaranje pocetne stranice
2 driver.get("http://localhost:3000/");
3 driver.manage().window().setSize(new Dimension(1051, 806));
4
5 //prijava
6 driver.findElement(By.cssSelector(".dropdown-toggle")).click();
```

¹³<https://selenium.dev/selenium-ide/>

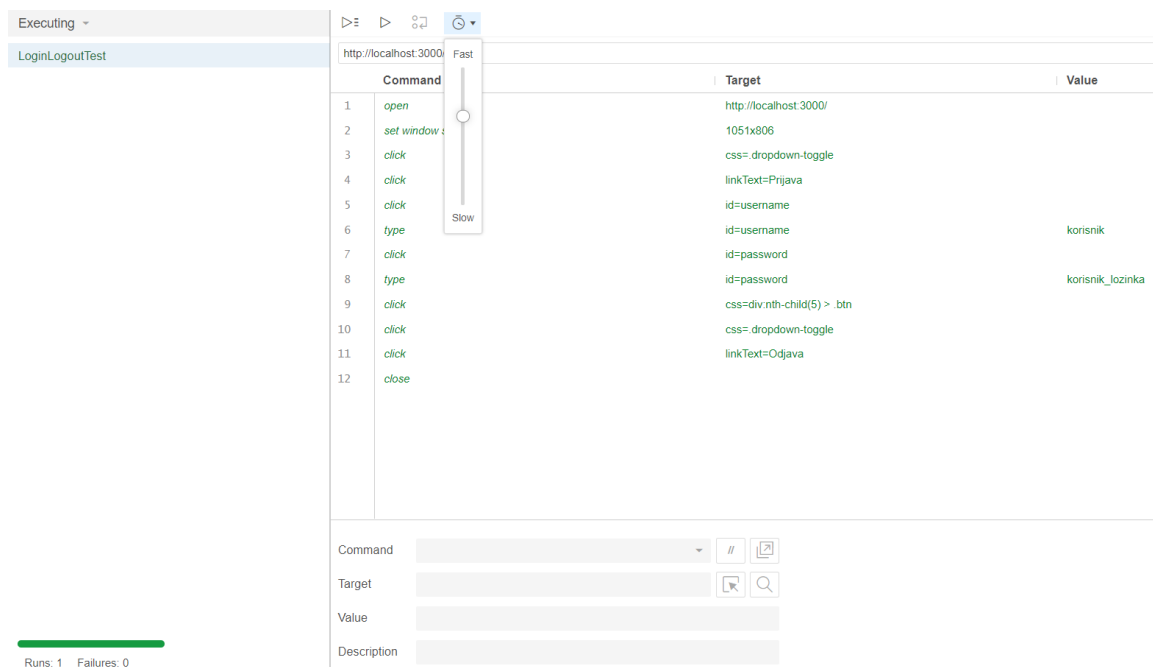
```

7  driver.findElement(By.linkText("Prijava")).click();
8
9      //upis korisnickog imena
10 driver.findElement(By.id("username")).click();
11 driver.findElement(By.id("username")).sendKeys("korisnik");
12
13     //upis lozinke
14 driver.findElement(By.id("password")).click();
15 driver.findElement(By.id("password")).sendKeys("korisnik_lozinka");
16
17     //potvrda
18 driver.findElement(By.cssSelector("div:nth-child(5) > .btn")).click();
19
20     //odjava
21 driver.findElement(By.cssSelector(".dropdown-toggle")).click();
22 driver.findElement(By.linkText("Odjava")).click();
23
24     //zatvaranje
25 driver.close();

```

Programski isječak 9: Izvorni kod za ispitni slučaj 8

Rezultat: Očekivani rezultat je ostvaren. Korisnik je uspješno prijavljen i odjavljen.



Slika 5.1: Rezultat za ispitni slučaj 8

Ispitni slučaj 9: Pregled osobnih podataka**Ulaz:**

1. Otvaranje početne stranice u web pregledniku te prijava u sustav¹⁴.
2. Pritisak na ikonu korisnika u gornjem desnom kutku.
3. Odabir opcije "Postavke" iz padajućeg izbornika.
4. Odjava iz sustava¹⁴.

Očekivani rezultat:

- 1.a Prikazuje se početna stranica.
- 1.b Korisnik je prijavljen.
2. Prikazuje se padajući izbornik s opcijama "Postavke" i "Odjava".
3. Prikazuju se osobni podaci korisnika.
4. Korisnik je odjavljen iz sustava.

Izvorni kod:

```
1      //prijava
2
3      //klik na padajuci izbornik
4  driver.findElement(By.cssSelector(".dropdown-toggle")).click();
5
6      //klik na korisnika
7  driver.findElement(By.linkText("korisnik")).click();
8
9      //pregled osobnih podataka
10
11     //odjava
```

Programski isječak 10: Izvorni kod za ispitni slučaj 9

Rezultat: Očekivani rezultat je ostvaren. Pregled osobnih podataka bio je uspješan.

¹⁴Postupak prijave i odjave korisnika u sustav je prikazan u ispitnom slučaju 8.

The screenshot displays the Selenium IDE interface for a test suite named 'PersonalDataTest'. The test is running at 'http://localhost:3000/'. A speed slider is visible, set to 'Fast'. The test consists of 14 steps:

Step	Command	Target	Value
1	open	http://localhost:3000/	
2	set window size	1051x806	
3	click	css= dropdown-toggle	
4	click	linkText=Prijava	
5	click	id=username	
6	type	id=username	korisnik
7	click	id=password	
8	type	id=password	korisnik_lozinka
9	click	css=div:nth-child(5) > .btn	
10	click	css= dropdown-toggle	
11	click	linkText=korisnik	
12	click	css= dropdown-toggle	
13	click	linkText=Odjava	
14	close		

At the bottom, the status bar shows 'Runs: 1' and 'Failures: 0'.

Slika 5.2: Rezultat za ispitni slučaj 9

Ispitni slučaj 10: Komentiranje kontejnera**Ulaz:**

1. Otvaranje početne stranice u web pregledniku te prijava u sustav¹⁵.
2. Pritisak na ikonu povećala u lijevom izborniku.
3. Upisivanje ispravnog identifikatora kontejnera, pritisak na ikonu povećala.
4. Pritisak na ikonu kontejnera s tim identifikatorom.
5. Upisivanje komentara kontejnera, ocjenjivanje kontejnera i pritisak na ikonu diskete.
6. Odjava iz sustava¹⁵.

Očekivani rezultat:

- 1.a Prikazuje se početna stranica.
- 1.b Korisnik je prijavljen.
2. Prikazuje se tekstualno polje za upis identifikatora kontejnera.
3. Odabrani kontejner se centrira na karti.
4. Prikazuje se stranica kontejnera s objavljenim recenzijama tog kontejnera.
- 5.a Komentar je objavljen.
- 5.b Ocjena je zabilježena.
6. Korisnik je odjavljen iz sustava.

Izvorni kod:

```
1 //prijava
2
3 //klik na gumb za traženje kontejnera
4 driver.findElement(By.cssSelector(".btn:nth-child(1) path")).click();
5 {
6     WebElement element = driver.
7         findElement(By.cssSelector(".btn:nth-child(1)"));
8     Actions builder = new Actions(driver);
9     builder.moveToElement(element).perform();
10 }
11
12 //upisivanje id-a kontejnera
13 driver.findElement(By.cssSelector(".form-control")).click();
14 driver.findElement(By.cssSelector(".form-control")).sendKeys("93");
15
16 //potvrda
```

¹⁵Postupak prijave i odjave korisnika u sustav je prikazan u ispitnom slučaju 8.

```
17 driver.findElement(By.id("addon-wrapping")).click();
18
19 //klik na kontejner
20 driver.
21     findElement(By.cssSelector("div:nth-child(2) > div > svg > path")).
22         click();
23
24 //upis ocjene
25 driver.findElement(By.name("userOcjena")).click();
26 driver.findElement(By.name("userOcjena")).sendKeys("3");
27
28 //upis komentara
29 driver.findElement(By.name("userKomentar")).click();
30 driver.findElement(By.name("userKomentar")).
31     sendKeys("Kontejner je u prosjecnom stanju.");
32
33 //potvrda
34 driver.findElement(By.cssSelector("div:nth-child(6) svg")).click();
35
36 //odjava
```

Programski isječak 11: Izvorni kod za ispitni slučaj 10

Rezultat: Očekivani rezultat je ostvaren. Ocjena i komentar su pohranjeni.

Executing ▾

ContainerCommentTest

http://localhost:3000/ Fast

	Command	Target	Value
1	open	http://localhost:3000/	
2	set window size	1054x808	
3	click	css= dropdown-toggle	
4	click	linkText=Prijava	
5	click	id=username	
6	type	id=username	korisnik
7	click	id=password	
8	type	id=password	korisnik_lozinka
9	click	css=div:nth-child(5) > .btn	
10	click	css=.btn:nth-child(1) path	
11	mouse over	css=.btn:nth-child(1)	
12	click	css=.form-control	
13	type	css=.form-control	93
14	click	id=addon-wrapping	
15	click	css=div:nth-child(2) > div > svg > path	
16	click	name=userOcjena	
17	type	name=userOcjena	3

Command //

Target

Value

Description

Runs: 1 Failures: 0

Slika 5.3: Rezultat za ispitni slučaj 10, prvi dio

17	type	name=userOcjena	3
18	click	name=userKomentar	
19	type	name=userKomentar	Kontejner je u prosječnom stanju.
20	click	css=div:nth-child(6) svg	
21	click	css= dropdown-toggle	
22	click	linkText=Odjava	
23	close		

Command //

Target

Value

Description

Runs: 1 Failures: 0

Slika 5.4: Rezultat za ispitni slučaj 10, drugi dio

Ispitni slučaj 11: Upravljanje kontejnerima**Ulaz:**

1. Otvaranje početne stranice u web pregledniku te prijava u sustav¹⁶.
2. Pritisak na ikonu kontejnera u lijevom izborniku.
3. Pritisak na ikonu plusa.
4. Upisivanje podataka o kontejneru i pritisak na ikonu diskete.
5. Pritisak na ikonu kante za smeće u kartici novostvorenog kontejnera.
6. Odjava iz sustava¹⁶.

Očekivani rezultat:

- 1.a Prikazuje se početna stranica.
- 1.b Korisnik je prijavljen.
2. Prikazuje se popis kontejnera koji su uneseni u aplikaciju.
3. Prikazuje se forma za unos podataka o novom kontejneru.
- 4.a Kontejner je spremljen u bazu podataka.
- 4.b Prikazuje se popis kontejnera koji sadrži novostvoreni kontejner.
- 5.a Kontejner je izbrisan iz baze podataka.
- 5.b Prikazuje se popis kontejnera koji ne sadrži obrisani kontejner.
6. Korisnik je odjavljen iz sustava.

Izvorni kod:

```
1 //prijava
2
3 //klik na gumb za traženje kontejnera
4 driver.findElement(By.cssSelector(".btn:nth-child(1) > svg")).click();
5
6 //klik na gumb kontejnera
7 driver.
8     findElement(By.cssSelector(".btn:nth-child(5) > svg")).click();
9
10 //klik na gumb za dodavanje kontejnera
11 driver.
12     findElement(By.cssSelector("div:nth-child(1) > svg > path")).click();
13
14 //unos podataka za kontejner i potvrda unosa
15 driver.findElement(By.name("adresa")).click();
16 driver.findElement(By.name("adresa")).sendKeys("Savska Ulica 1");
```

¹⁶Postupak prijave i odjave korisnika u sustav je prikazan u ispitnom slučaju 8.

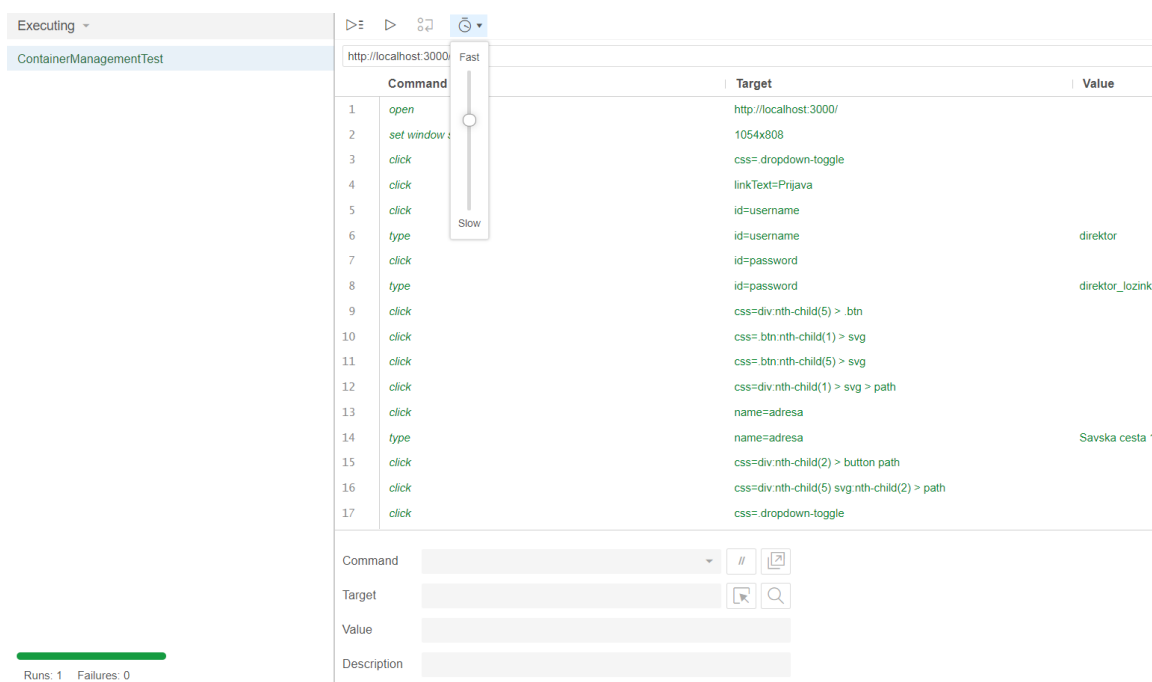
```

17 driver.
18     findElement(By.cssSelector("div:nth-child(2) > button path")).
19         click();
20
21     //brisanje kontejnera
22 driver.
23 findElement(By.cssSelector("div:nth-child(5) svg:nth-child(2) > path")).
24     click();
25
26     //odjava

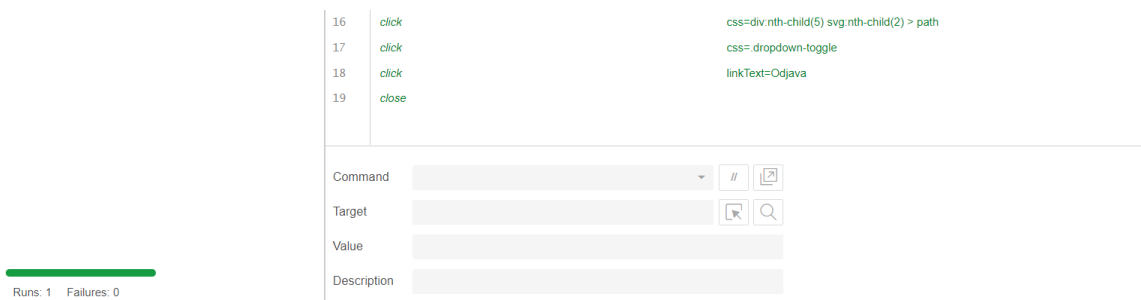
```

Programski isječak 12: Izvorni kod za ispitni slučaj 11

Rezultat: Očekivani rezultat je ostvaren. Upravljanje kontejnerima bilo je uspješno.



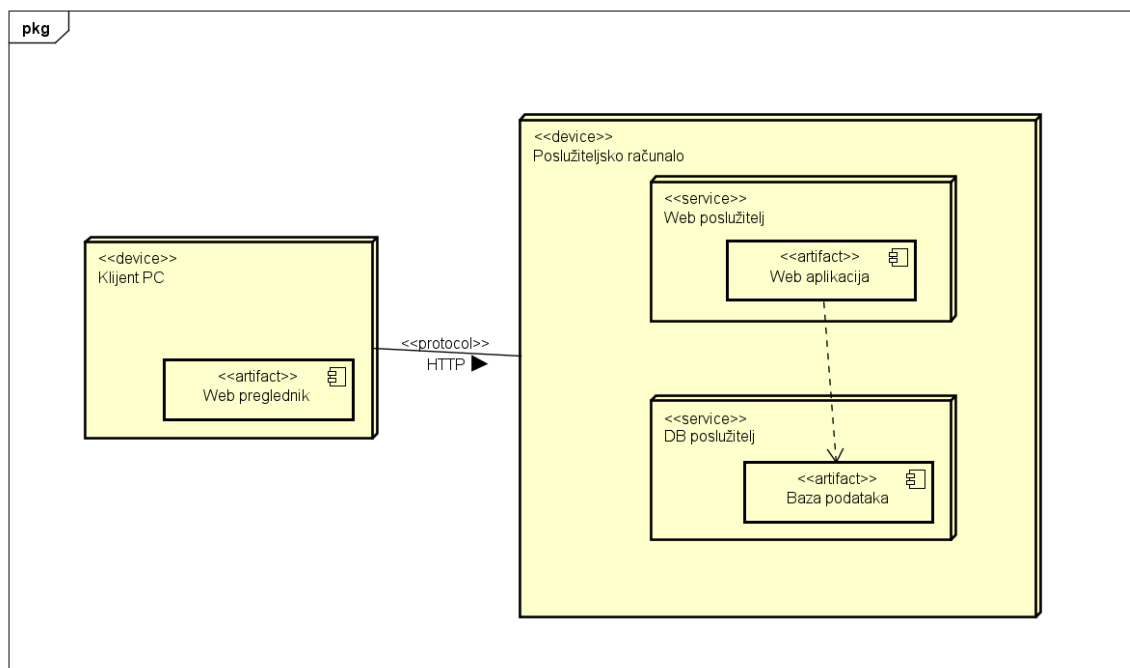
Slika 5.5: Rezultat za ispitni slučaj 11, prvi dio



Slika 5.6: Rezultat za ispitni slučaj 11, drugi dio

5.3 Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju sustava te su usredotočeni na odnos sklopovskih i programskih dijelova. Sustav je baziran na arhitekturi "klijent - poslužitelj". Klijenti preko web preglednika pristupaju web aplikaciji dok se na poslužiteljskom računalu nalaze web poslužitelj i poslužitelj baze podataka. Komunikacija između klijenta i poslužitelja odvija se preko HTTP veze.



Slika 5.7: Dijagram razmještaja

5.4 Upute za puštanje u pogon

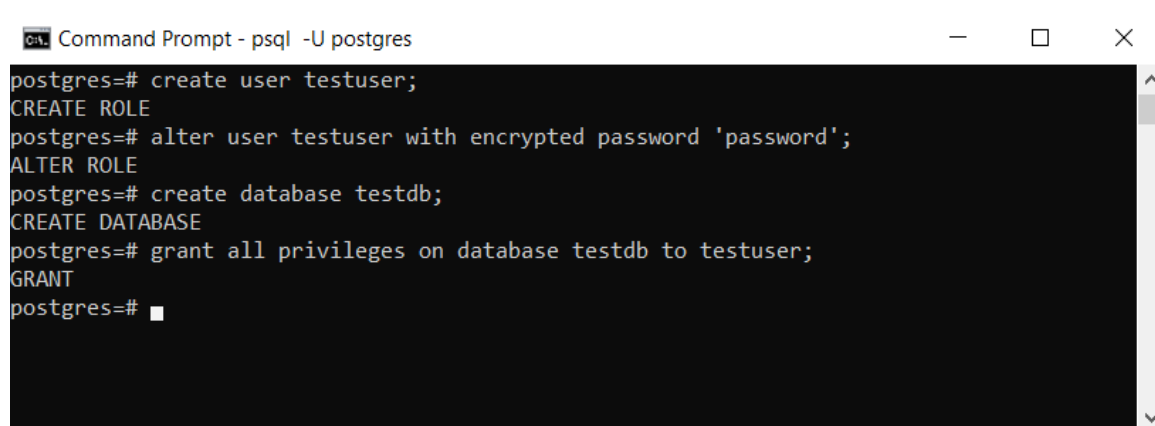
Backend:

1. Instalacija programske podrške(Java, Apache Maven, PostgreSQL)

- Java (<https://www.oracle.com/technetwork/java/javase/downloads/index.html>)
- Apache Maven (<https://maven.apache.org/install.html>)
- PostgreSQL (<https://www.postgresql.org/download/>)

2. Postavljanje baze podataka (Stvaranje korisnika te baze podataka)

```
create user testuser;  
alter user testuser with encrypted password 'password';  
create database testdb;  
grant all privileges on database testdb to testuser;
```



```
Command Prompt - psql -U postgres  
postgres=# create user testuser;  
CREATE ROLE  
postgres=# alter user testuser with encrypted password 'password';  
ALTER ROLE  
postgres=# create database testdb;  
CREATE DATABASE  
postgres=# grant all privileges on database testdb to testuser;  
GRANT  
postgres=#
```

Slika 5.8: Postavljanje baze podataka

3. Konfiguracija *pom.xml* i *application.properties* datoteka

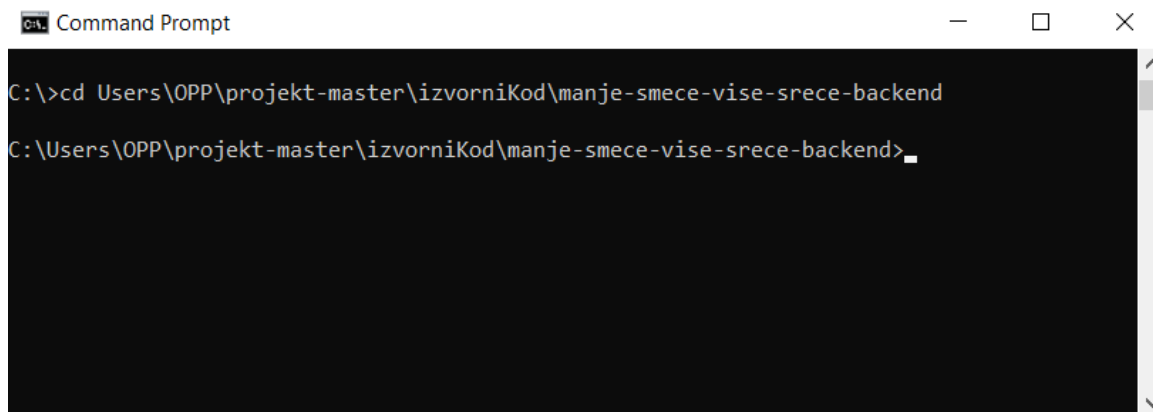
- U *pom.xml* potrebno je dodati:

```
<dependency>  
    <groupId>org.postgresql</groupId>  
    <artifactId>postgresql</artifactId>  
</dependency>
```

- U application.properties potrebno je dodati:

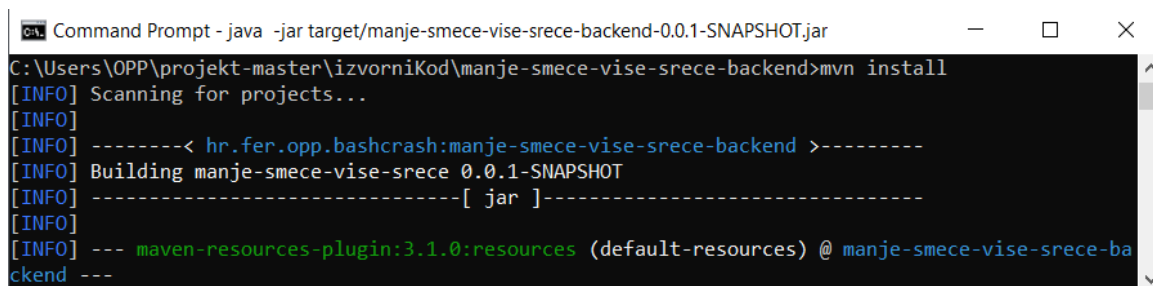
```
spring.datasource.url = jdbc:postgresql://localhost:5432/testdb
spring.datasource.username = testuser
spring.datasource.password = password
```

4. Pozicioniranje u radni direktorij backend-a putem cmd terminala



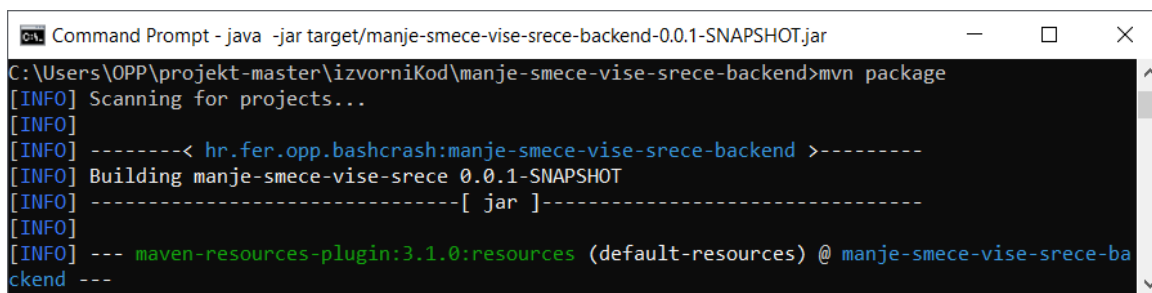
Slika 5.9: Pozicioniranje u radni direktorij

5. mvn install



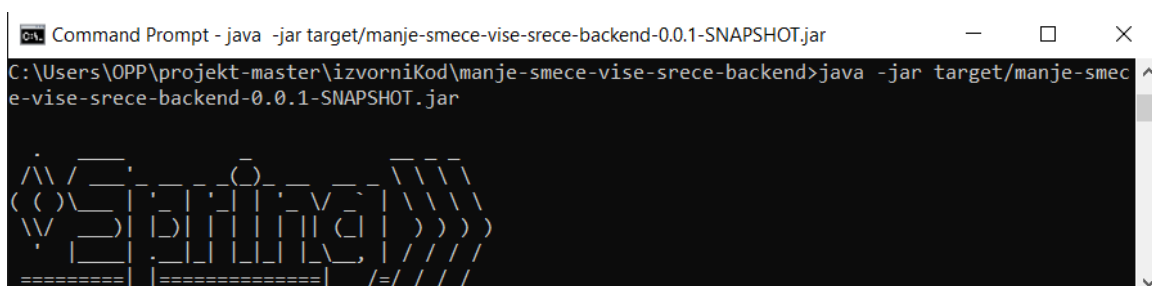
Slika 5.10: mvn install

6. mvn package



Slika 5.11: mvn package

7. `java -jar target/manje-smece-vise-srece-backend-0.0.1-SNAPSHOT.jar`

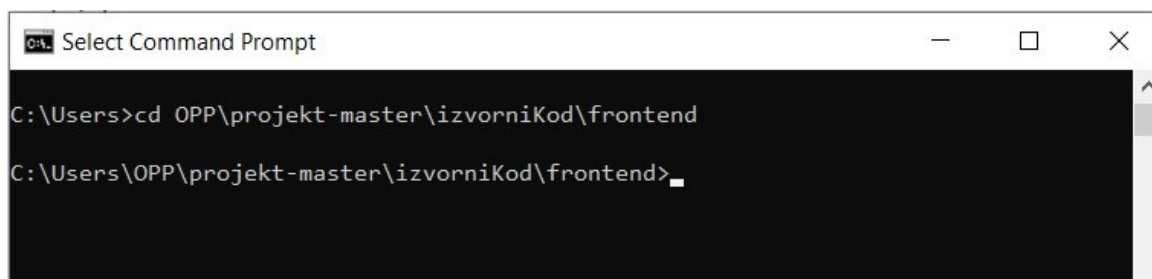


Slika 5.12: Pokretanje

Frontend:

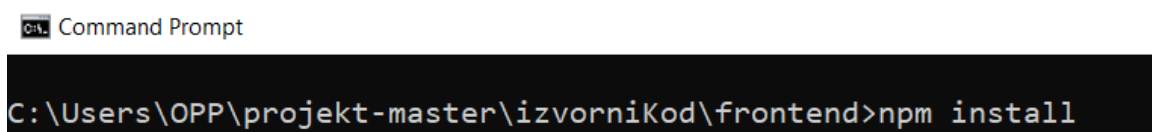
1. Instalacija programske podrške(Node.js)
 - Node.js (<https://nodejs.org/en/download/>)
2. Konfiguracija package.json datoteke
 - U package.json datoteku potrebno je dodati:

```
"proxy": "http://localhost:8080"
```
3. Pozicioniranje u radni direktorij frontenda-a putem cmd terminala



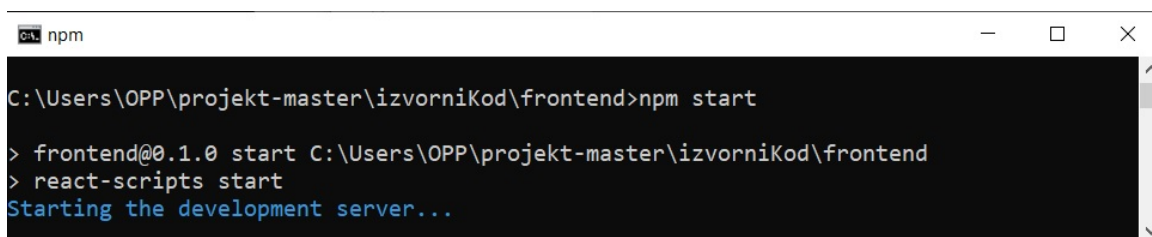
Slika 5.13: Pozicioniranje u radni direktorij

4. npm install

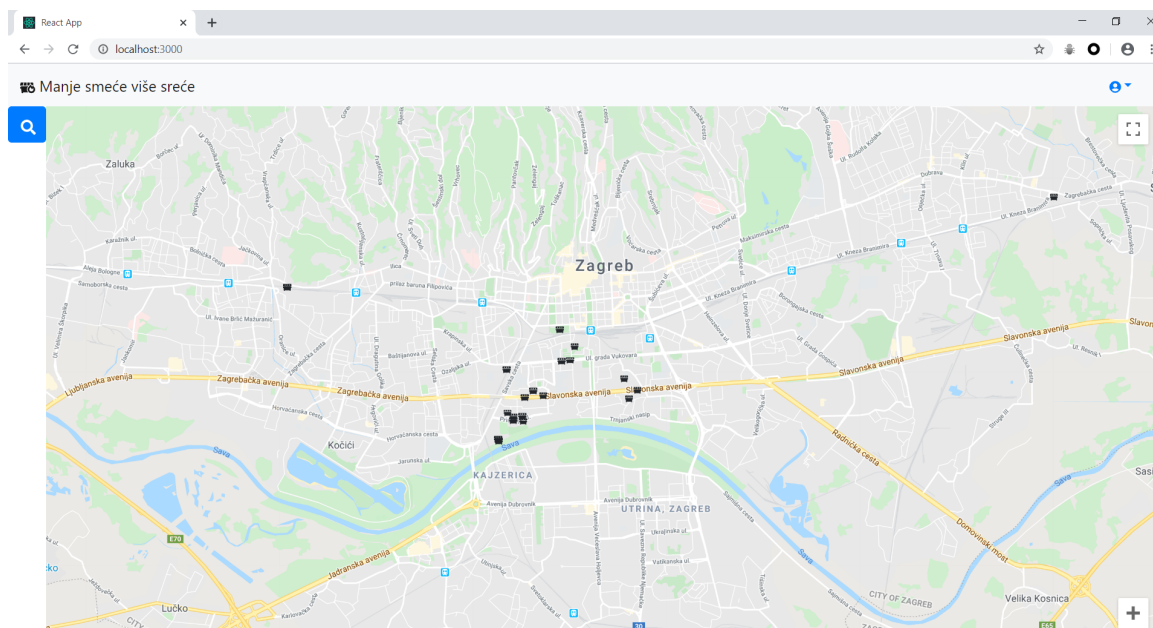


Slika 5.14: npm install

5. npm start



Slika 5.15: npm start



Slika 5.16: Konačno

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije za efikasniji i transparentniji odvoz smeća koja se postiže dodavanjem senzora na kontejnere te mogućnosti recenziranja istih kako bi se doznalo njihovo stanje i urednost. Izrada aplikacije (te popratne dokumentacije) trajala je 17 tjedana i bila je podijeljena u dvije faze.

Prva faza započela je okupljanjem tima za razvoj aplikacije, ali većina vremena utrošena je na planiranje razvoja i dokumentaciju. Znanje stečeno na ovom predmetu pomoglo nam je pri izradi dokumentacije, što je uvelike pomoglo pri brzjoj i efikasnijoj izradi same aplikacije. Ogroman broj raznih dijagrama koji variraju po kompleksnosti, veličini i funkciji pružao je pogled iz svih perspektiva, kako korisničkih, tako i sistemskih, na cijelu aplikaciju. Radi bolje koordinacije, timovi su bili podijeljeni u podskupine koje su, ovisno o fazi projekta, imale razne zadatke, kao što su razvoj frontenda i backenda.

Druga faza bila je nešto kraća od prve, ali činjenica da je trajala kraće ne znači da je bila lakša, dapače, tempo rada bio je puno intenzivniji jer je u pitanju bio sam razvoj aplikacije, što je (za većinu članova tima) uključivalo i učenje novih tehnologija i alata. Druga faza uključivala je i završetak projektne dokumentacije koja je, unatoč tome što najveći fokus nije bio na njoj, uspješno odrađena.

Komunikacija među članovima vršila se pomoću aplikacije Slack koja je bila izuzetno korisna jer se moglo praktično i brzo podijeliti zadatke na sve članove tima.

Iako se aplikacija zadržava na području jednog grada - Zagreba, moguće je buduće proširenje na druge gradove Hrvatske, ali i šire.

Sudjelovanje u izradi ove aplikacije bilo je neprocjenjivo iskustvo za sve članove razvojnog tima. Naučili smo raditi u raznim novim tehnologijama te smo shvatili koliki je vremenski (ali i timski) pothvat izrada ovakve aplikacije. Koordiniranost među članovima ekipe bila je izuzetno bitan faktor koji je omogućio da veći dio izrade aplikacije i dokumentacije prođe brzo i bez većih problema. Zadovoljni smo krajnjim rezultatom i, iako uvijek postoji prostor za poboljšanje, možemo reći da nam je svima drago što smo radili zajedno i što smo uspjeli postići ovakav monumentalni pothvat kao što je razvoj aplikacije.

Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

3.1	Dijagram obrazaca uporabe, funkcionalnost korisnika aplikacije . .	16
3.2	Dijagram obrazaca uporabe, funkcionalnost direktora i zaposlenika	17
3.3	Sekvencijski dijagram za UC1	19
3.4	Sekvencijski dijagram za UC8	21
3.5	Sekvencijski dijagram za UC9	23
3.6	Sekvencijski dijagram za UC13	25
4.1	Arhitektura sustava	27
4.2	MVC skica	29
4.3	Dijagram baze podataka	33
4.4	Dijagram razreda - dio Controllers	35
4.5	Dijagram razreda - dio DTO	36
4.6	Dijagram razreda - dio Models	37
4.7	Dijagram stanja	38
4.8	Dijagram aktivnosti	39
4.9	Dijagram komponenti	41
5.1	Rezultat za ispitni slučaj 8	54
5.2	Rezultat za ispitni slučaj 9	56
5.3	Rezultat za ispitni slučaj 10, prvi dio	59
5.4	Rezultat za ispitni slučaj 10, drugi dio	59
5.5	Rezultat za ispitni slučaj 11, prvi dio	61
5.6	Rezultat za ispitni slučaj 11, drugi dio	62
5.7	Dijagram razmještaja	63
5.8	Postavljanje baze podataka	64
5.9	Pozicioniranje u radni direktorij	65
5.10	mvn install	65
5.11	mvn package	66
5.12	Pokretanje	66
5.13	Pozicioniranje u radni direktorij	67
5.14	npm install	67

5.15 npm start	67
5.16 Konačno	68

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 9. listopada 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - upoznavanje članova
 - definiranje rokova

2. sastanak

- Datum: 15. listopada 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - određivanje funkcionalnih i nefunkcionalnih zahtjeva aplikacije
 - prioritizacija zahtjeva

3. sastanak

- Datum: 17. listopada 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - određivanje funkcionalnih i nefunkcionalnih zahtjeva aplikacije
 - prioritizacija zahtjeva
 - odabir tehnologije za izradu web aplikacije

4. sastanak

- Datum: 22. listopada 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:

- razrada specifikacije programske potpore
- obrasci uporabe i sekvencijski dijagrami

5. sastanak

- Datum: 29. listopada 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - baza podataka
 - komentiranje dosadašnje verzije dokumentacije

6. sastanak

- Datum: 5. studenog 2019.
- Prisustvovali: M. Lazarić, R. Pavliš, F. Vale
- Teme sastanka:
 - komentiranje obrazaca uporabe i sekvencijskih dijagrama

7. sastanak

- Datum: 12. studenog 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - komentiranje dosadašnje verzije dokumentacije
 - demonstracija generičkih funkcionalnosti

8. sastanak

- Datum: 10. prosinca 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - podjela poslova na dokumentaciji i samoj aplikaciji
 - rasprava o preostalim poglavljima dokumentacije

9. sastanak

- Datum: 17. prosinca 2019.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - rad sa Selenium testovima

10. sastanak

- Datum: 7. siječnja 2020.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - demonstracija alfa inačice aplikacije

11. sastanak

- Datum: 14. siječnja 2020.
- Prisustvovali: M. Benotić, P. Jakopec, M. Lazarić, A. Novak, R. Pavliš, I. Teofilović, F. Vale
- Teme sastanka:
 - završno komentiranje dokumentacije
 - konačna dorada dokumentacije

Tablica aktivnosti

	Marko Lazarić	Aleksandar Novak	Filip Vale	Ivan Teofilović	Matija Benotić	Petar Jakopec	Robert Pavliš
Upravljanje projektom	11	9	10	9	9	10	10
Opis projektnog zadatka	2	1	1	-	-	2	2
Funkcionalni zahtjevi	-	-	-	-	2	-	-
Opis pojedinih obrazaca	-	1	2	-	-	2	2
Dijagram obrazaca	-	-	2	2	-	-	-
Sekvencijski dijagrami	-	-	1	-	2	2	2
Opis ostalih zahtjeva	1	-	-	1	-	-	-
Arhitektura i dizajn sustava	-	-	-	-	-	3	-
Baza podataka	2	-	3	-	-	-	-
Dijagram razreda	3	-	-	-	-	1	-
Dijagram stanja	-	-	3	-	-	-	-
Dijagram aktivnosti	-	-	3	-	-	-	-
Dijagram komponenti	-	-	-	-	4	-	-
Korištene tehnologije i alati	-	-	-	-	-	-	4
Ispitivanje programskog rješenja	6	-	-	-	5	-	-
Dijagram razmještaja	-	-	-	3	-	-	-
Upute za puštanje u pogon	-	-	-	-	-	4	-
Dnevnik sastajanja	1	-	4	-	-	-	-
Zaključak i budući rad	-	-	-	-	-	-	3
Popis literature	-	-	-	-	-	-	-
Autentikacija	-	3	-	-	3	-	-
Frontend	-	10	-	-	-	-	-
Backend	5	-	-	-	-	-	-

Dijagrami pregleda promjena