Programsko inženjerstvo

Ak. god. 2020./2021.

ParkShare

Dokumentacija, Rev. 2

Grupa: Proggers

Voditelj: Marko Barbir

Datum predaje: 17.11.2021.

Nastavnik: Hrvoje Nuić, mag. ing.

Sadržaj

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	**Opis promjene/dodatka**	**Autori**	**Datum**
0.1	Napravljen predložak.	M.B.	12.10.2021.
0.2	Ažuriran predložak dokumentacije	M.B.	20.10.2021.
0.3	Dodan dnevnik.txt	M.B.	27.10.2021.
0.4	Dodane *Use Case* deklaracije	M.P.	28.10.2021.
0.5	Dodan opis projektnog zadatka	M.Ž.	29.10.2021.
0.6	Dodani *Use Case* obrasci	M.I., F.B., Z.R.	28.10.2021.
0.7	Dodani *Use Case* dijagrami	M.I., F.B., Z.R.	12.11.2021.
0.8	Dodani sekvencijski dijagrami	M.I.	12.11.2021.
0.9	Arhitektura i dizajn sustava	Z.R.	13.11.2021.
0.10	Dodan opis baze podataka i tablica	Z.R.	16.11.2021.
0.11	Uređena početna stranica	Z.R.	16.11.2021.
0.12	Dodani nefunkcionalni zahtjevi	M.Ž.	16.11.2021.
0.13	Dodani dijagrami razreda	M.I., F.B.	17.11.2021.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	17.11.2021.

2. Opis projektnog zadatka

Cilj projekta je razviti programsku podršku za stvaranje aplikacije „ParkShare“ koja će korisniku omogućiti rezervaciju i plaćanje parkinga. Korisnik će moći također pregledati sva slobodna parkirališna mjesta za automobile i bicikle. Prilikom pokretanja sustava korisnik će na mapi vidjeti obližnja parkirališta i dobit će rutu do najbližeg te će moći vidjeti ima li slobodnih mjesta kako ne bi došao na popunjeno parkiralište. Neregistrirani korisnik šalje zahtjev za registraciju s odabranom ulogom koju želi, dakle voditelj parkinga ili klijent. Neregistriranom korisniku pokazuju se dostupna parkirališta i dostupna mjesta, a klijentima se prikazuje i zauzetost mjesta u stvarnom vremenu. Za registraciju korisniku su potrebni sljedeći podaci:

- *Korisničko ime*
- *Lozinka*
- *Ime*
- *Prezime*
- *Slika osobne*
- *IBAN račun*
- *Email adresa*

Ukoliko se podaci podudaraju s nekim postojećim korisnikom u bazi podataka sustav šalje grešku korisniku koji se pokušao registrirati i upozorava ga da takav korisnik već postoji. Registracija završava potvrdom putem emaila korisnika, a voditelje mora potvrditi dodatno i administrator. Svaki korisnik aplikacije može vidjeti svoje privatne podatke u aplikaciji i može ih mijenjati. Također korisnik može izbrisati svoj korisnički račun. Postoje 3 vrste korisnika:

- *Klijent*
- *Administrator*
- *Voditelj parkirališta*

Administrator može vidjeti popis svih klijenata i voditelja i njihove osobne podatke i može ih mijenjati po volji ako je potrebno. Administrator iz baze podataka

dobiva popis svih nepotvrđenih voditelja parkirališta te ih on ukoliko smatra da su im podaci valjani potvrđuje. Administrator može također pregledati sva parkirališta i njihova mjesta u aplikaciji i može mijenjati njihove podatke. Administrator može ukloniti ili promijeniti korisnički račun bilo kojeg ne-administrativnog korisnika. Voditelj zapisuje specifikacije svog parkirališta kao što su:

- *Naziv parkirališta*
- *Opis*
- *Fotografija*
- *Cjenik*

Voditelj ucrtava na karti svako parkirališno mjesto svog parkirališta i može za po želji odabrano parkirališno mjesto odrediti može li se rezervirati i postavlja senzore. Ukoliko automobil dođe na određeno parkirališno mjesto gdje se nalazi senzor, senzor javlja da je mjesto sada zauzeto.Ako automobil napusti parkirališno mjesto senzor javlja da se mjesto oslobodilo. Nakon ucrtavanja parkirališnih mjesta zapisuje se ukupan kapacitet parkirališta. Klijent na karti pregledava parkirališta u svojoj blizini i klikom na njih može vidjeti dostupne informacije o tom parkiralištu ujedno i je li ima slobodnih mjesta na parkiralištu. Ukoliko klijent želi biti siguran, on može rezervirati parkirališno mjesto tako da upiše termin u kojem želi parkirališno mjesto i pokazat će mu sva dostupna parkirališna mjesta na karti koja se mogu rezervirati. Drugi način za rezervaciju je, klijent unaprijed odabere parkirališna mjesta koja želi i onda će mu se prikazati kalendar s dostupnim terminima rezervacije. Klijent može rezervirati mjesto na proizvoljno dugo vremena i može rezervacija biti definirana kao ponavljajuća. Voditelj za svoje parkiralište određuje cijenu ovisno o trajanju rezervacije. Ukoliko se klijent odluči rezervirati parkirališno mjesto, parking mu se odmah naplaćuje putem aplikacije, ako pak klijent dođe bez rezervacije i odabere slobodno parkirališno mjesto, parking se naplaćuje na parkiralištu u tom trenutku. U aplikaciji svaki klijent posjeduje novčanik kojim plaća parking, a u svakom trenutku može uplatiti određenu svotu novca u novčanik kako bi mogao platiti parking. Prilikom naplate parkinga ukoliko klijent nema dovoljno sredstava za platiti na računu, preusmjerava ga se na stranicu za nadoplatu novca u novčanik kako bi mogao podmiriti usluge parkirališta. Službena valuta kojom sustav raspolaže jest Hrvatska kuna - HRK. Svakom korisniku se prikazuje stanje njegovog novčanika i rezervacije koje ima u sustavu parkirališta, administrator nema novčanik i mogućnost rezervacije parkilašnog mjesta pa se njemu prikazuje broj korisnika i broj parkirališta u sustavu. Voditelji mogu

vidjeti statistiku zauzetosti parkirališnih mjesta kroz vrijeme na svom parkiralištu. Prikupljaju se povijesne informacije kako je kroz vrijeme parkiralište bilo zauzeto i statistički se izračunava zauzetost u tom periodu vremena te se voditelju to prikazuje u obliku grafa. Voditelj može ucrtati najviše jedno parkiralište u aplikaciji, a parkirališna mjesta može ucrtavati i brisati po želji, isto tako može izbrisati cijelo parkiralište. Voditelj parkinga može izmijeniti mogućnost rezervacije nekog parkirališnog mjesta te promijeniti cijenu parkinga. Napomenuli smo da će na svakom parkiralištu biti moguće ponuditi parking i biciklima. Voditelj parkirališta ne ucrtava parkirališna mjesta kao za automobile već samo određuje ukupan kapacitet parkirališta za bicikle. Korisnici bicikla neće morati brinuti o naplati parkinga jer se on neće naplaćivati takvim korisnicima samim tim neće imati ni opciju za rezervaciju parkinga. Svaki voditelj mora na karti naznačiti je li parkiralište dostupno i za bicikle kako bi korisnici znali na karti gdje mogu parkirati svoje bicikle. Parkirališna mjesta za bicikle neće imati senzor za očitavanje dostupnosti mjesta. U bazu podataka spremamo sve podatke o korisnicima i njihovim ovlastima. Pohranjujemo sve podatke o parkiralištima(ime, cijena, opis), također sve informacije o parkirališnim mjestima(zauzetost, tip). Svaki klijent u aplikaciji nakon korištenja određenog parkirališta dobiva „Like“ opciju, ikonu palčića gore i dolje te odabire klikom je li mu se parkiralište svidjelo ili ne. Također može i zanemariti danu opciju nije obvezno odgovoriti. Aplikacija mora podržavati rad više korisnika u stvarnom vremenu.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Asistent (Naručitelj)
2. Korisnici
3. Vlasnici parkinga
4. Administrator(i)
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Voditelj parkinga (inicijator) može:
   (a) Napraviti (max 1) parkiralište, te ga izbrisati, ili promijeniti ime i cjenik
   (b) Ucrtati nova i izbrisati postojeća parking mjesta za svoj parking
   (c) Definirati i promijeniti mogućnost rezervacije pojedinog parkirališnog mjesta
   (d) Pregledati svoje osobne podatke i promijeniti ih
   (e) Izbrisati svoj korisnički račun
   (f) Pregledati statistiku zauzetosti svog parkirališta i parkirališnih mjesta kroz vrijeme u obliku grafa
2. Neregistrirani/neprijavljeni korisnik (inicijator) može:
   (a) Registrirati se kao klijent ili kao voditelj parkinga
   (b) Prijaviti se
   (c) Pregledati sva parkirališta i parkirališna mjesta dostupna u aplikaciji
3. Prijavljeni klijent (inicijator) može:
   (a) Odjaviti se
   (b) Pregledati svoje osobne podatke i promijeniti ih
   (c) izbrisati korisnički račun

(d) Pregledati sva parkirališta i parkirališna mjesta dostupna u aplikaciji te njihovu zauzetost u realnom vremenu i rezervati ih
(e) Upisati destinaciju, tip prijevoza i procjenu trajanja parkinga te dobiti mogućnost rezervacije najbližeg parkirališnog mjesta
(f) Uplatiti novac u račun
(g) Pregledati povijest transakcija i rezervacija

4. Administrator (inicijator) može:

(a) Odjaviti se
(b) Izbrisati ili promijeniti korisnički račun bilo kojeg ne-administratorskog korisnika
(c) Pregledati sva parkirališta i parkirališna mjesta dostupna u aplikaciji, te promijeniti njihove podatke
(d) Potvrditi novi račun voditelja parkirališta

5. Baza podataka (sudionik):

(a) Pohranjuje sve podatke o korisnicima i njihovim ovlastima
(b) Pohranjuje sve podatke o parkiralištima (ime, cjenik)
(c) Pohranjuje sve podatke o parkirališnim mjestima (zauzetost, tip)

6. Senzor zauzetosti (inicijator):

(a) Šalje signal kad se parkirališno mjesto zauzme ili oslobodi

7. Banka (sudionik):

(a) Na klijentov zahtjev prenosi novac s klijentovog računa na račun aplikacije

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Dodavanje parkirališta

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Dodavanje parkirališta u bazu podataka
- **Sudionici:** Voditelj parkirališta, baza podataka
- **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga
- **Opis osnovnog tijeka:**
  1. Voditelj parkirališta odabire opciju za dodavanje parkinga
  2. Voditelj parkirališta unosi informacije o svom parkiralištu
  3. Voditelj parkirališta na karti unosi dostupna parkirališna mjesta
  4. Voditelj parkirališta postavlja senzor koji osvježava informaciju zauzetosti parkirališnog
- **Opis mogućih odstupanja:**

  2.a Unešeno već postojeće ime parkinga
  1. Aplikacija ispisuje grešku

  3.a Označavanje mjesta na karti na kojem već postoji parking
  1. Aplikacija ispisuje grešku

  4.a Aplikacija se ne može povezati sa senzorom
  1. Aplikacija ispisuje grešku

UC2 - Brisanje parkirališta

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Brisanje parkirališta
- **Sudionici:** Voditelj parkirališta, baza podataka
- **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište već postoji u bazi podataka
- **Opis osnovnog tijeka:**
  1. Voditelj zatraži brisanje parkirališta
  2. Parkiralište se briše iz baze podataka
  3. Voditelju se javlja da je parkiralište uspješno izbrisano
- **Opis mogućih odstupanja:**

  2.a Neuspješno brisanje podataka iz baze podataka
  1. Aplikacija ispisuje grešku

UC3 - Izmjena podataka o parkiralištu

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Izmjena podataka o parkiralištu
- **Sudionici:** Voditelj parkirališta, baza podataka
- **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište već postoji u bazi podataka
- **Opis osnovnog tijeka:**
  1. Voditelj zatraži izmjenu podataka
  2. Voditelj unosi nove izmjenjene podatke o parkiralištu
  3. Novi podatci se spremaju u bazu podataka
  4. Voditelju se javlja da su podatci uspješno izmjenjeni
- **Opis mogućih odstupanja:**

  3.a Neuspješno spremanje podataka u bazu podataka
  1. Aplikacija ispisuje grešku

UC4 - Izmjena podataka o parkirališnom mjestu

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Izmjena podataka o parkirališnom mjestu
- **Sudionici:** Voditelj parkirališta, baza podataka
- **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište i parkirno mjesto već postoji u bazi podataka
- **Opis osnovnog tijeka:**
  1. Voditelj zatraži izmjenu podataka
  2. Voditelj unosi nove izmjenjene podatke o parkirališnom mjestu
  3. Novi podatci se spremaju u bazu podataka
  4. Voditelju se javlja da su podatci uspješno izmjenjeni
- **Opis mogućih odstupanja:**

  3.a Neuspješno spremanje podataka u bazu podataka
  1. Aplikacija ispisuje grešku

UC5 - Dodavanje parkirališnog mjesta

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Dodavanje parkirališnog mjesta
- **Sudionici:** Voditelj parkirališta, baza podataka
- **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište i parkirno mjesto već postoji u bazi podataka
- **Opis osnovnog tijeka:**

1. Voditelj zatraži dodavanje novog parkirališnog mjesta na parkiralištu
2. Voditelj na karti ucrtaje parkirališno mjesto
3. Voditelj parkirališta definira je li moguće rezervirati parkirališno mjesto
4. Podatci se spremaju u bazu podataka
5. Voditelju se javlja da su podatci uspješno spremljeni u bazi podataka

– **Opis mogućih odstupanja:**

2.a Voditelj dodaje parkirališno mjesto na lokaciji na kojoj vec postoji parkirališno mjesto
   1. Voditelja se upozorava o grešci

4.a Neuspješno spremanje podataka u bazu podataka
   1. Aplikacija ispisuje grešku

UC6 - Brisanje parkirališnog mjesta

– **Glavni sudionik:** Voditelj parkirališta
– **Cilj:** Brisanje parkirališnog mjesta
– **Sudionici:** Voditelj parkirališta, baza podataka
– **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište već postoji u bazi podataka
– **Opis osnovnog tijeka:**
   1. Voditelj zatraži brisanje parkirališnog mjesta
   2. Parkirališno mjesto se briše iz baze podataka
   3. Voditelju se javlja da je parkiralište uspješno izbrisano
– **Opis mogućih odstupanja:**

2.a Neuspješno brisanje podataka iz baze podataka
   1. Aplikacija ispisuje grešku

UC7 - Definiranje mogućnosti rezervacije pojedinog parkirališnog mjesta

– **Glavni sudionik:** Voditelj parkirališta
– **Cilj:** Definiranje mogućnosti rezervacije pojedinog parkirališnog mjesta
– **Sudionici:** Voditelj parkirališta, baza podataka
– **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište već postoji u bazi podataka
– **Opis osnovnog tijeka:**
   1. Voditelj odabire parkirališno mjesto
   2. Voditelj definira mogućnost rezervacije parkirališnog mjesta
   3. Informacije se spremaju u bazi podataka

4. Voditelju se javlja da su podatci uspješno spremljeni

– **Opis mogućih odstupanja:**

3.a Neuspješno spremanje podataka u bazu podataka

1. Aplikacija ispisuje grešku

UC8 - Promjena mogućnosti rezervacije pojedinog parkirališnog mjesta

– **Glavni sudionik:** Senzor zauzetosti
– **Cilj:** Omogućuje ponovnu rezervaciju novog tek oslobođenog mjesta
– **Sudionici:** Baza podataka
– **Preduvjet:**
– **Opis osnovnog tijeka:**
1. Vozilo dođe ili ode sa parkirališnog mjesta
2. Senzor registrira promjenu
3. Promjena se pohranjuje u bazu podataka
– **Opis mogućih odstupanja:**

2.a Kvar na senzoru zauzetosti

1. Hitan popravak ili zamjena senzora od strane voditelja parkinga

UC9 - Pregled osobnih podataka

– **Glavni sudionik:** Administrator
– **Cilj:** Dati administratoru posebnu mogućnost pregleda osobnih podataka svih registriranih korisnika
– **Sudionici:** Baza podataka
– **Preduvjet:** Glavni sudionik nužno mora biti prijavljen kao administrator
– **Opis osnovnog tijeka:**
1. Administrator zatraži pregled osobnih podataka određenog registriranog korisnika
2. Određeni podaci se dohvaćaju iz baze podataka
3. Podaci se formatirano prikazuju administratoru
– **Opis mogućih odstupanja:**

2.a Neuspjelo dohvaćanje podataka iz baze podataka

1. Aplikacija ispisuje grešku

UC10 - Promjena osobnih podataka

– **Glavni sudionik:** Administrator
– **Cilj:** Dati administratoru posebnu mogućnost izmjene osobnih poda-

taka svih registriranih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Glavni sudionik nužno mora biti prijavljen kao administrator
- **Opis osnovnog tijeka:**
  1. Administrator zatraži pregled osobnih podataka određenog registriranog korisnika
  2. Određeni podaci se dohvaćaju iz baze podataka
  3. Administrator odluči mijenjati osobne podatke odabranog korisnika
  4. Administrator potvrdi promjene
  5. Promjene se pohranjuju u bazu podataka
- **Opis mogućih odstupanja:**

  2.a Neuspjelo dohvaćanje podataka iz baze podataka
    1. Aplikacija ispisuje grešku

  3.a Administrator je unio krivi format podataka
    1. Aplikacija upozorava na krivi format i ne dopušta potvrdu

  5.a Neuspjela pohrana u bazu podataka
    1. Aplikacija ispisuje grešku

UC11 - Registracija u sustav

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Omogućuje neregistriranom korisniku registraciju u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Glavni sudionik nije registriran od prije
- **Opis osnovnog tijeka:**
  1. Neregistrirani korisnik šalje zahtjev za registraciju sa određenom ulogom za koju se hoće prijaviti (voditelj parkinga ili klijent)
  2. Neregistrirani korisnik upisuje potrebne podatke (ime, prezime email,...)
  3. Neregistrirani korisnik potvrđuje svoje podatke
  4. Ako se korisnik prijavio kao voditelj parkinga administrator ga mora potvrditi
  5. Podaci o novoregistriranom korisniku se pohranjuju u bazu podataka
  6. Registracija se završava potvrdom preko email adrese
- **Opis mogućih odstupanja:**

  2.a Upisani podaci nisu u pravilnom formatu

1. Aplikacija upozorava na krivi format i ne dopušta nastavak

5.a Neuspjela pohrana u bazu podataka

1. Aplikacija dojavljuje grešku i ne dopušta nastavak

UC12 - Prijava u sustav

- **Glavni sudionik:** Bilo koji registrirani korisnik (klijent, voditelj, admin)
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je već registriran u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik unosi korisničko ime i lozinku
  2. Provjerava se postoje li upisani podaci u bazi podataka
  3. Korisnik dobiva pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**

  2.a Upisano korisničko i lozinka su netočni

  1. Sustav obavještava korisnika o neispravnom upisu

UC13 - Odjava iz sustava

- **Glavni sudionik:** Bilo koji registrirani korisnik (klijent, voditelj, admin)
- **Cilj:** Odjavljivanje iz korisničkog sučelja
- **Sudionici:**
- **Preduvjet:** Korisnik prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik zatraži odjavu iz sustava
  2. Korisnik izlazi iz korisničkog sučelja

UC14 - Brisanje korisničkog računa

- **Glavni sudionik:** Bilo koji registrirani korisnik (klijent, voditelj, admin)
- **Cilj:** Omogućuje brisanje korisničkog računa
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. Korisnik zatraži brisanje korisničkog računa
  2. Korisničko ime i lozinka korisnika se brišu iz baze podataka
  3. Izlazak iz korisničkog sučelja

UC15 - Pregled statistike zauzetosti

- **Glavni sudionik:** Voditelj

- **Cilj:** Voditelj može voditi evidenciju o zauzetosti svojih parkirališnih mjesta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao voditelj
- **Opis osnovnog tijeka:**
  1. Voditelj odluči vidjeti zauzetost svih svojih parkirališnim mjesta
  2. Dohvaćanje podataka o zauzetosti iz baze podataka
  3. Voditelj pregledava podatke te donosi daljnje odluke
- **Opis mogućih odstupanja:**

  2.a Nema nijedno parkirno mjesto
    1. Ispis pogreške
    2. Voditelju omoguceno dodavanje parkirnih mjesta

UC16 - Pregled dostupnih parkirališnih mjesta

- **Glavni sudionik:** Registrirani i neregistrirani korisnici
- **Cilj:** Uvid u dostupna parkirališna mjesta
- **Sudionici:** Baza podataka
- **Preduvjet:** Nema
- **Opis osnovnog tijeka:**
  1. Odabir parkinga te parkirališnog mjesta
  2. Dohvaćanje podataka iz baze podataka
  3. Daljnje registriranje u slučaju da korisnik želi izvršiti rezervaciju
- **Opis mogućih odstupanja:**

  2.a Nema slobodnih parkirališnih mjesta
    1. Poruka o popunjenosti parkinga

UC17 - Pregled zauzetosti parkirališnih mjesta

- **Glavni sudionik:** Klijent
- **Cilj:** Svaki klijent može evidentirati zauzetost o svim parkirnim mjestima
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik prijavljen kao klijent te postoje parkirna mjesta
- **Opis osnovnog tijeka:**
  1. Klijent odluči vidjeti podatke o zauzetosti
  2. Dohvat svih parkirnih mjesta
  3. Odabir pojedinog parkirnog mjesta
  4. Dohvat podataka iz baze podataka

5. Uvid u dostavljene podatke

– **Opis mogućih odstupanja:**

2.a Nema nijedno parkirno mjesto

1. Poruka

UC18 - Rezervacija parkirališnog mjesta

– **Glavni sudionik:** Klijent
– **Cilj:** Klijent rezervira parkirališno mjesto
– **Sudionici:** Baza podataka
– **Preduvjet:** Korisnik rezerviran kao klijent
– **Opis osnovnog tijeka:**
1. Korisnik odluči rezervirati parkirno mjesto
2. Otvaranje karte te odabir tipa prijevoza te trajanje parkinga
3. Iscrtavanje rute do najbliže slobodnog parkirališnog mjesta
4. Klijent odabire jedan od 2 načina rezerviranja
5. Nastavak na plaćanje
– **Opis mogućih odstupanja:**

3.a Nema slobodnih parkirališnih mjesta

1. Ispis pogreške
2. Preusmjeravanje nazad na ispunjavanje obrasca o traženju slobodnih parkirnih mjesta

UC19 - Upis destinacije, tipa prijevoza i procjene trajanja parkinga

– **Glavni sudionik:** Klijent
– **Cilj:** Unos podataka za rezervaciju parkinga
– **Sudionici:** Baza podataka
– **Preduvjet:** Korisnik registriran kao klijent
– **Opis osnovnog tijeka:**
1. Upis destinacije, tipa prijevoza i procjene trajanja parkinga
2. Filtriranje slobodnih parkirnih mjesta
3. Rezervacija parkirnog mjesta
– **Opis mogućih odstupanja:**

2.a Unos krivih podataka

1. Poruka te ponovna mogućnost unosa podataka

UC20 - Uplata novca na račun

– **Glavni sudionik:** Klijent

- **Cilj:** Uplata klijenta na račun voditelja parkinga
- **Sudionici:** Baza podataka, Voditelj parkinga
- **Preduvjet:** Korisnik registriran kao klijent te klijent izvršio rezervaciju
- **Opis osnovnog tijeka:**
  1. Uvid u cijenu željenog parkirališnog mjesta
  2. Klijent dodaje novac u novčanik
  3. Klijent plaća novac na voditeljeov račun
  4. Ažuriranje novčanika u bazi podataka
  5. Potvrda o uspješnoj transakciji
- **Opis mogućih odstupanja:**

  2.a Klijent nema dovoljno novca u novčaniku
  1. Ispis poruke
  2. Preusmjeravanje na dodavanje novca u novčanik

  3.a Neuspjelo ažuriranje podataka
  1. Ispis poruke
  2. Preusmjeravanje na ponovno izvršavanje transakcije

UC21 -Pregled povijesti transakcija i rezervacija

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:**Pregled povijesti transakcija i rezervacija
- **Sudionici:** Voditelj parkirališta i baza podataka
- **Preduvjet:** Korisnik mora imati ulogu voditelja parkinga, voditelj mora upravljati tim parkingom, parkiralište već postoji u bazi podataka
- **Opis osnovnog tijeka:**
  1. Voditelj zatraži pregled povijesti transakcija i rezervacija
  2. Podatci se dohvaćaju iz baze podataka
  3. Podatci se ispisuju voditelju
- **Opis mogućih odstupanja:**

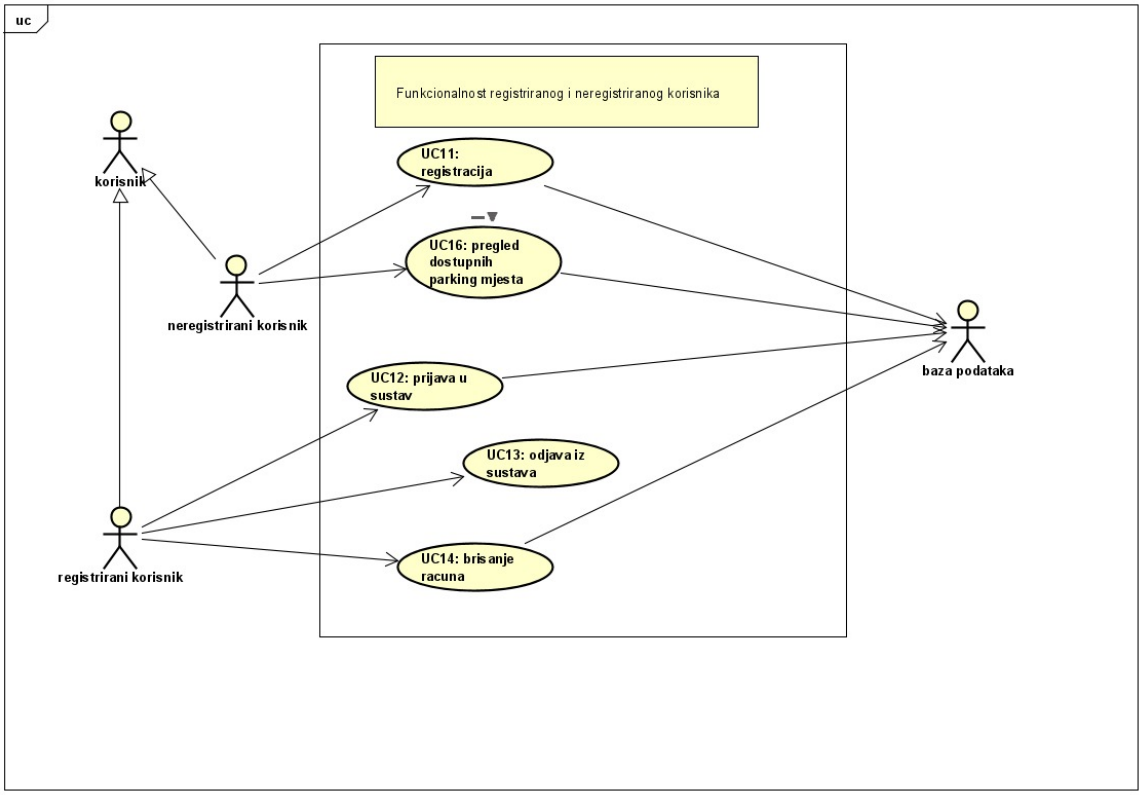
  2.a Neuspješno spremanje podataka u bazu podataka
  1. Aplikacija ispisuje grešku

UC22 - Potvrda računa voditelja

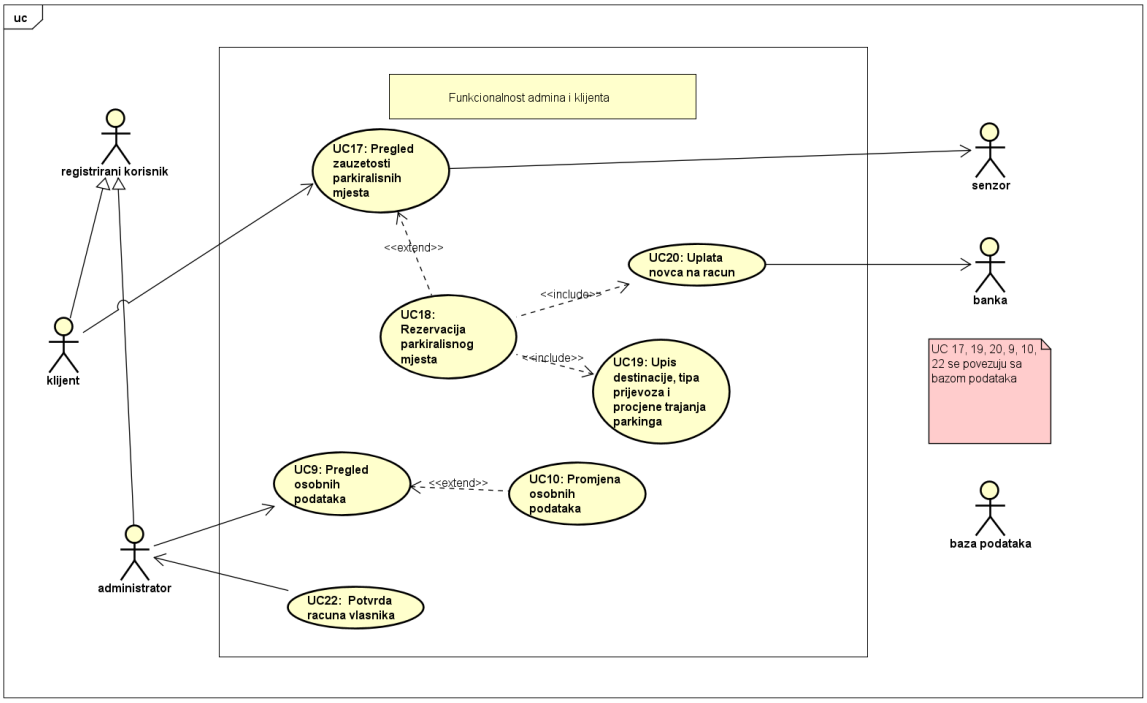
- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi račun voditelja parkinga
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je napravio račun potvrdom email adrese te korisnik je prijavljen kao administrator

– **Opis osnovnog tijeka:**
  1. Administrator pregledava voditeljev račun
  2. Odabire funkcionalnost potvrđivanja u slučaju valjanosti računa suprotno odbacuje račun
– **Opis mogućih odstupanja:**

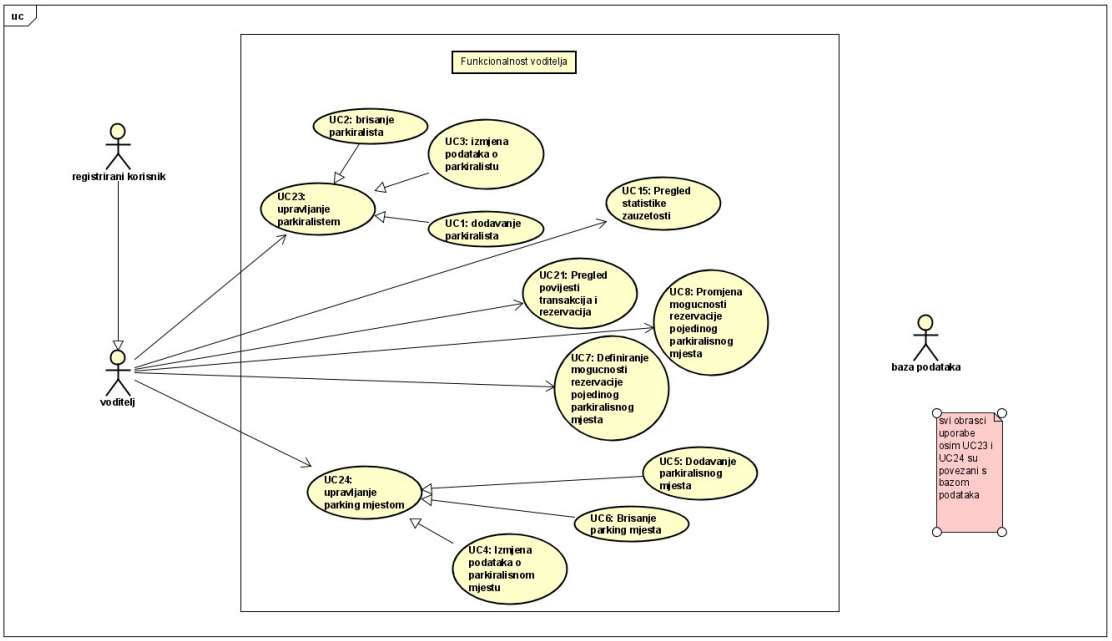
Dijagrami obrazaca uporabe



Slika 3.1: UC registrirani i neregistrirani korisnik



Slika 3.2: UC funkcionalnost klijenta i administratora

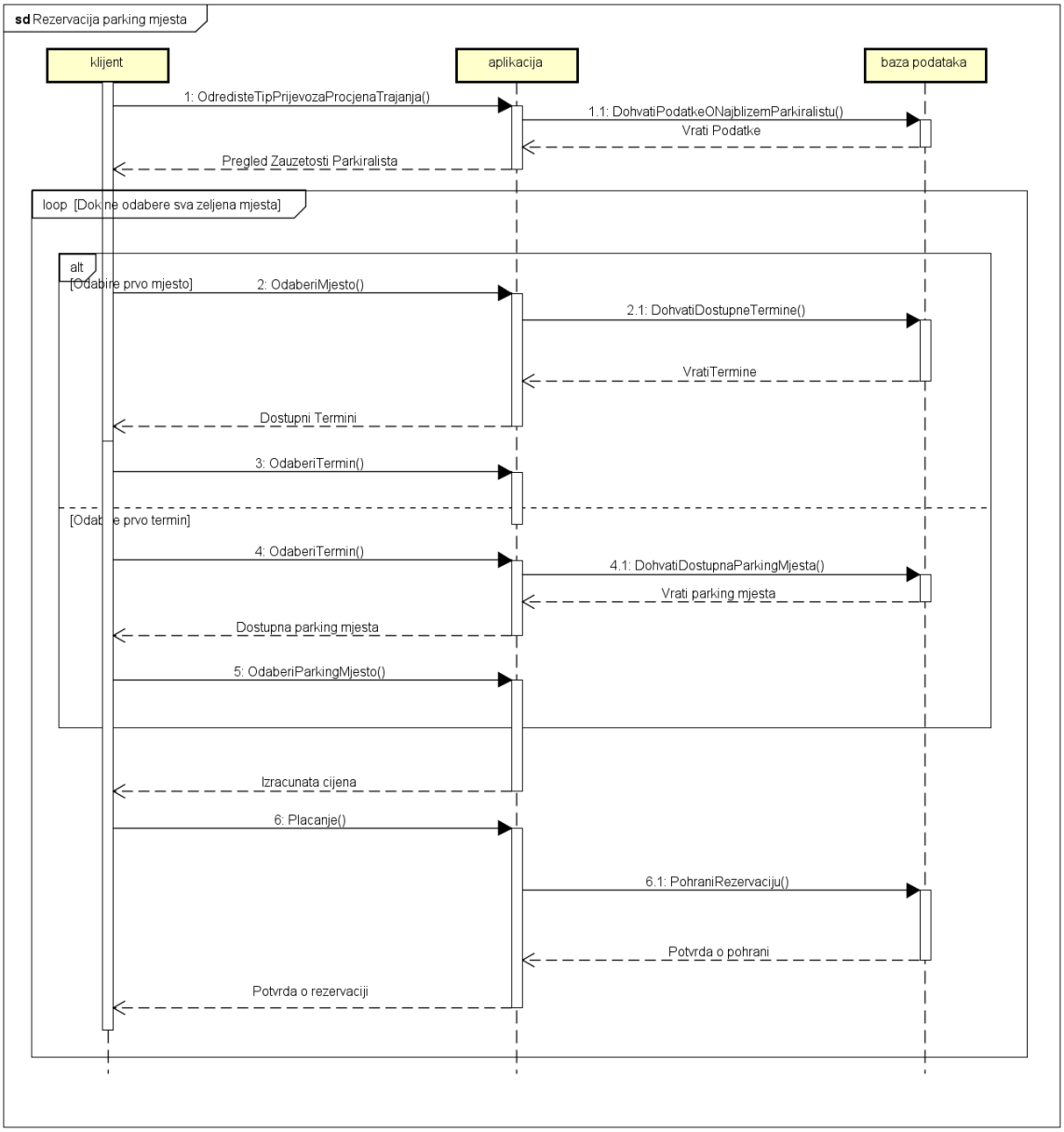


Slika 3.3: UC funkcionalnost voditelja

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC18 - Rezervacija parkirališnog mjesta

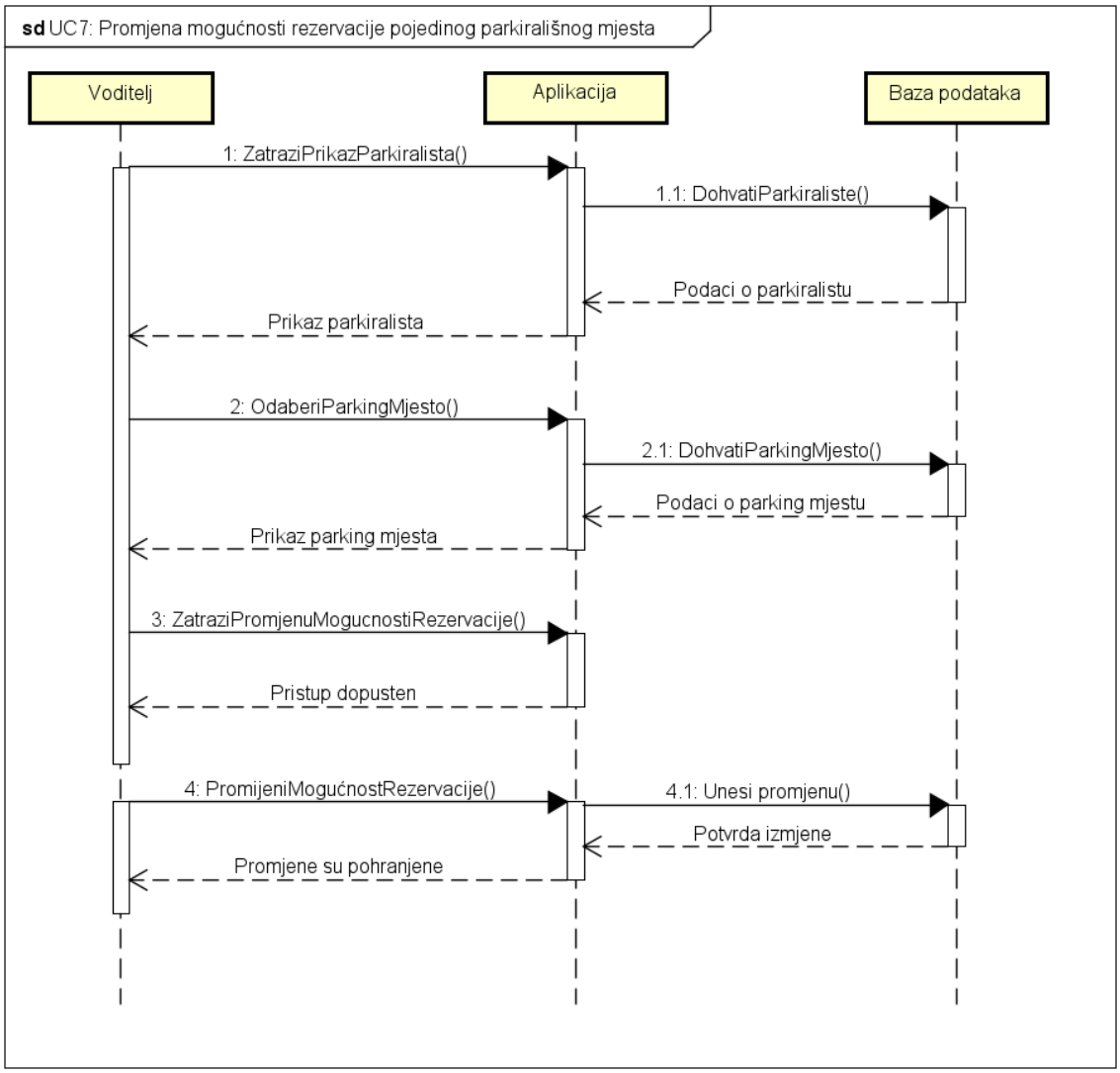
Klijent odabire svoje odredište, tip prijevoza i procjenu trajanja parkinga. Aplikacija mu pronalazi najbliži parking te prikazuje njegovu zauzetost. Klijent sada ima dvije mogućnosti: prvo odabrati termin nakon čega mu aplikacija dohvaća iz baze podataka slobodna mjesta u tom terminu ili najprije odabrati parking mjesto nakon čega mu aplikacija dohvaća iz baze podataka slobodne termine za to mjesto. Nakon klijentovog odabira aplikacija izračunava cijenu parkinga. U ovom slučaju uzeli smo opciju da klijent plaća putem aplikacije. Nakon plaćanja rezervacija se pohranjuje u bazu podataka i ispisuje se potvrda o rezervaciji. Postupak je iterativan.



Slika 3.4: Sekvencijski dijagram za UC18

Obrazac uporabe UC8 - Promjena mogućnosti rezervacije pojedinog parkirališnog mjesta

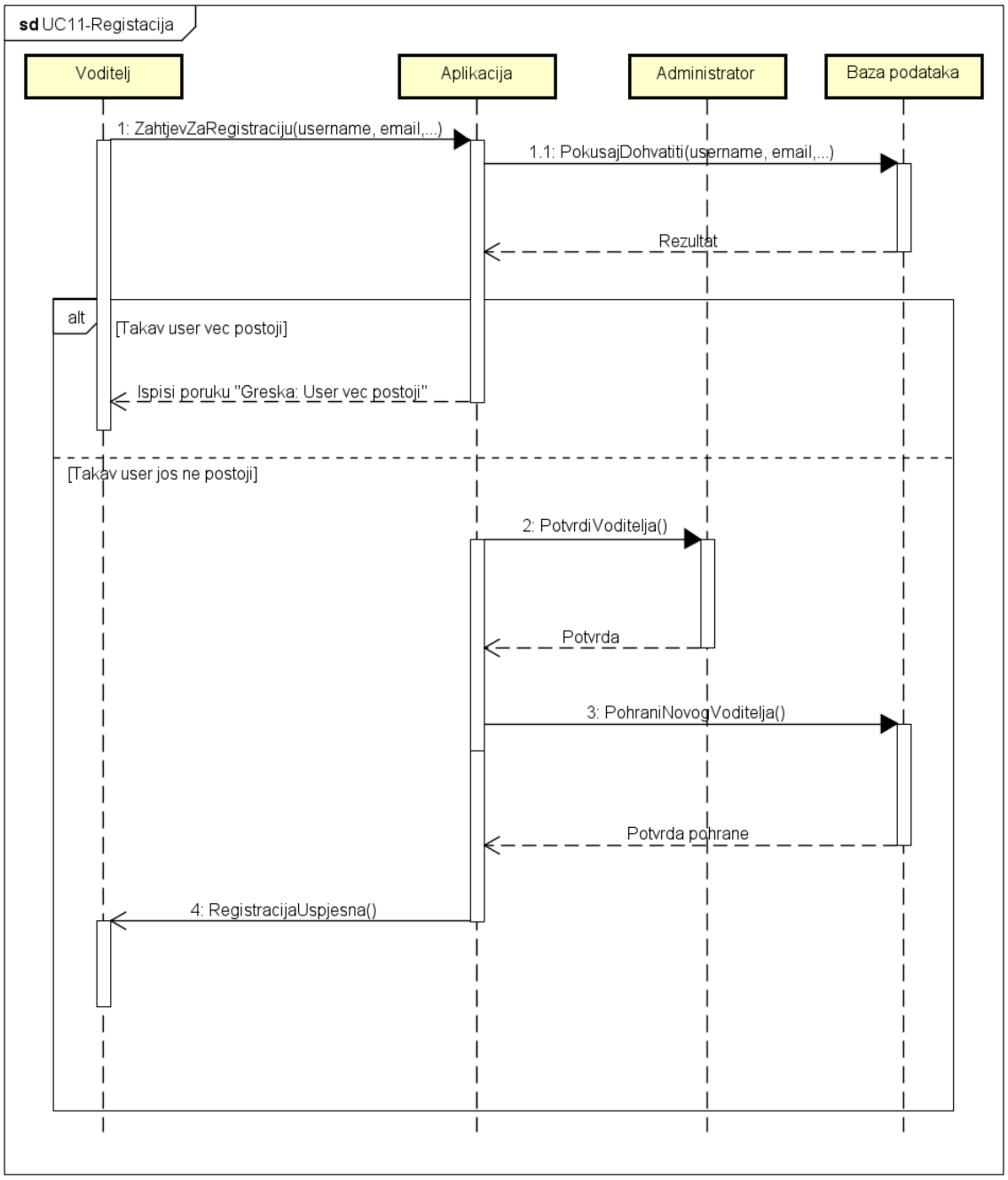
Voditelj zatraži prikaz parkirališta nakon čega aplikacija dohvaća iz baze podataka podatke o parkiralištu. Voditelj dobiva prikaz tog parkirališta. Voditelj odabere parkirališno mjesto. Aplikacija dohvaća podatke o parkirališnom mjestu iz baze podataka. Voditelj želi promjeniti njegovu mogućnost rezervacije. Aplikacija mu to odobrava jer je u ulozi voditelja. Voditelj mijenja mogućnost rezervacije parking mjesta, aplikacija promjene pohranjuje u bazu i javlja voditelju da su promjene obavljene.



Slika 3.5: Sekvencijski dijagram za UC8

Obrazac uporabe UC11 - Registracija

U primjeru smo izabrali registraciju voditelja parkinga jer uz standardan tijek registracije korisnika, voditelja dodatno mora potvrditi i administrator. Voditelj šalje zahtjev za registraciju s potrebnim podacima. Aplikacija provjerava postoji li već u bazi podataka taj korisnik. Ako postoji registracija je neuspješna i aplikacija ispisuje grešku. Ako još ne postoji takav korisnik aplikacija šalje zahtjev za potvrdu administratoru. Nakon njegove potvrde aplikacija pohranjuje novog korisnika u bazu podataka u ulozi voditelja. Baza podataka potvrđuje pohranu i registracija je uspješna.



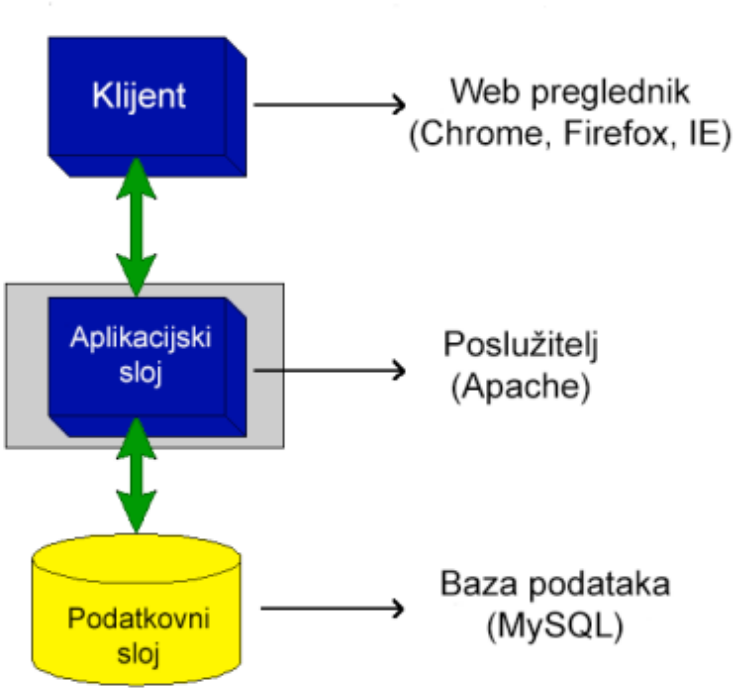
Slika 3.6: Sekvencijski dijagram za UC11

3.2 Ostali zahtjevi

- Sustav treba biti oblikovan i implementiran kako bi imao sva svojstva web aplikacije koristeći objektno orijentirane jezike
- Aplikacija bi trebala podržavati rad više korisnika u stvarnom vremenu
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Korisničko sučelje i sustav moraju podržavati sve znakove hrvatske abecede pri radu na aplikaciji, unos podataka, ispis podataka, u to se ubrajaju specijalni(dijaktrički) znakovi
- Ako se korisnik sustavom koristi na neispravan način, sustav mora i dalje ostati funkcionalan i raditi bez odstupanja
- Kako ne bismo korisnike dodatno opteretili s učenjem na koji način sustav radi, korisničko sučelje mora biti jednostavno za korištenje bez dodatnih uputa
- Nadodavanjem novih funkcionalnosti u sustav ne smijemo narušiti rad ranije implementiranih funkcionalnosti
- Službena i jedina valuta sustava jest hrvatska kuna - HRK
- Pristup sustavu mora biti omogucen iz javne mreže pomoću HTTPS
- Veza s bazom podataka mora biti jako dobro zaštićena od neželjenih vanjskih utjecaja, mora biti brza kako bi sustav mogao biti funkcionalan i mora biti otporna na vanjske greške

4. Arhitektura i dizajn sustava

Sve web aplikacije sastavljene su od klijentske strane(eng.frontend) i od serverske strane(eng.backend). Kod u klijentskoj strani je pisan u HTML, CSS i JS i on se izvodi u web pregledniku. Dok serverska strana se sastoji od poslovne logike i koda koji odgovara na HTTP zahtjeve. Kod na serverskoj strani može biti pisan u Javi, PHP, Python, itd.



Slika 4.1: Arhitektura web aplikacije

Web aplikacije podijeljene su na tri glavna sloja:

- **Prvi sloj predstavlja prezentacijski sloj**, to je korisničko sučelje i komunikacijski sloj aplikacije gdje korisnik stupa u interakciju s aplikacijom. Glavna svrha prezentacijskog sloja je prikazivanje i prikupljanje informacija od korisnika.
- **Drugi sloj predstavlja aplikacijski sloj**, to je glavni sloj web aplika-

cije. U ovom sloju se informacije prikupljene u prezentacijskom sloju obrađuju korištenjem poslovne logike. Te se također u ovom sloju dohvaćaju unose i izmjenjuju podatci koji se nalaze u podatkovnom sloju

– **Treći podatkovni sloj**, ili baza podataka je sloj u kojem se pohranjuju podatci koje koristi o obrađuje web aplikacija i u kojem se pristupa istim.

Svi slojevi međusobno komuniciraju preko standardiziranih Internetskih protokola, kojih se prilikom razvoja aplikacija treba pridržavati.Glavne prednosti razvijanja aplikacija u tri sloja su:

– Svaki sloj može biti napravljen u različitom programskom jeziku i okruženju.
– Svaki sloj može biti pokrenut na vlastitom serveru sto znači da slojevi ne ovise jedan o drugom.
– Brži razvoj aplikacije zato sto se svaki sloj može razvijati istovremeno.
– Poboljšana sigurnost zato sto možemo vršiti provjeru nad upitima i podatcima na tri razine i zbog toga što korisnik ne radi direktno sa SQL serverom.

Programski jezik kojeg smo odabrali za izradu naše web aplikacije je **Java** zajedno s **Spring Boot**-om i **Thymeleaf**-om, koristeći sustava baza podataka **PostgreSQL**. Odabrano razvojno okruženje je IntelliJ IDEA. Arhitektura sustava temeljiti će se na MVC (Model-View-Controller) konceptu. MVC koncept podržan je od strane Spring Boot radnog okvira i kao takav ima gotove predloške koji nam olakšavaju razvoj web aplikacije. MVC koncept sastoji se od:

– **Model** - Komponenta Model sadrzi cijelu logiku vezanu za podatke s kojom korisnik radi. To mogu biti podatci koji se prenose između komponenti View i Controler ili bilo koje druge podatke vezane za poslovnu logiku.
– **View** - Komponenta View koristi se za svu logiku korisničkog sučelja aplikacije. Na primjer, korisnički prikaz uključivat će sve komponente korisničkog sučelja kao što su tekstualni okviri, padajući izbornik itd. s kojima je u interakciji krajnji korisnik.
– **Controller** - Kontroleri djeluju kao sučelje između komponenti Modela i Viewa za obradu sve poslovne logike i dolaznih zahtjeva, manipuliranje podacima pomoću komponente Model i interakciju s Viewsima kako bi prikazali konačni izlaz.

4.1 Baza podataka

Za našu web aplikaciju koristit ćemo relacijsku bazu podataka. Relacijska model baze podataka se sastoji od više tablica podataka takozvanih relacija. Svaka relacija ima svoje ime koje je jedinstveno u toj bazi podataka i koristimo ga za razlikovanje relacija u jednoj bazi podataka. Svaka relacija ima stupce koji sadrže vrijednost nekog atributa. Atribut ima svoje ime po kojem ga razlikujemo od drugih atributa u toj relaciji. Svaka tablica ima jedan ili više jednistvenih atributa ili skup atributa koji je jedinstven koji se koristi kao primarni ključ tablice. Entiteti u našoj bazi podataka su:

- **Korisnik**(usertable)
- **Voditelj parkinga**(parkingowner)
- **Klijent**(client)
- **Rezervacija**(clientreservation)
- **Parkiralište**(parking)
- **Parkirališno mjesto**(parkingspot)
- **Zauzetost parkirnog mjesta**(parkingspotoccupancy)

4.1.1 Opis tablica

Korisnik Ovaj entitet sadrži sve važne informacije o korisniku aplikacije. Sadrži atribute: identifikator korisnika, korisničko ime, ime, prezime, email korisnika, lozinku, tip korisnika aplikacije i oznaku je li administrator potvrdio taj korisnički račun. Ovaj entitet ima tri specijalizacije na klijenta, voditelja parkinga i administratora, s tim da je ta specijalizacija na administratora nepotpuna, administrator nema nikakve dodatne atribute te ne postoji tablica za njega.

Korisnik(usertable)		
userid	INT	Jedinstveni identifikator korisnika
username	VARCHAR	Korisničko ime

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Korisnik(usertable)		
userfirstname	VARCHAR	Ime korisnika
usersurename	VARCHAR	Prezime korisnika
useremail	VARCHAR	Email korinika
temppassword	VARCHAR	Privremena lozinka
usertype	VARCHAR	Tip korisnika aplikacije (voditelj ili klijent)
confirmed	BOOLEAN	Varijabla koja pokazuje je li korisnik potvrđen od strane administratora

Voditelj parkirališta Ovaj entitet je specijalizacija entiteta korisnik on sadrži dodatne atribute: iban i poveznicu na sliku osobne iskaznice. Povezan je sa parkiralištem vezom 1 naprema 1.

Voditelj parkirališta(parkingowner)		
userid	INT	Jedinstveni identifikator korisnika
iban	CHARACTER	IBAN voditelja parkinga
idpicture	VARCHAR	Poveznica na sliku osobne iskaznice

Klijent Ovaj entitet je specijalizacija entiteta korisnik on sadrži dodatni atribut stanje računa klijenta. Povezan je sa entitetom rezervacija vezom n (rezervacija) naprema 1(klijent).

Klijent(client)		
userid	INT	Jedinstveni identifikator korisnika
walletbalance	NUMERIC	Stanje računa klijenta

Parkiralište Ovaj entitet sadrži sve bitne informacije o parkiralištu. Atributi su mu: korisnički id voditelja parkirališta, ime parkinga, poveznica na sliku parkinga, cijena parkinga po satu i opis parkirališta. Povezan je sa voditeljem parkiralista vezom 1 naprema 1 i sa parkirališnim mjestom vezom n (mjesto)

naprema 1 (parkiralište).

Parkiralište(parking)		
userid	INT	Jedinstveni identifikator voditelja parkinga
parkingname	VARCHAR	Ime parkirališta
parkingphoto	VARCHAR	Poveznica na sliku parkirališta
hourlyprice	VARCHAR	Cijena parkiranja po satu
description	VARCHAR	Opis parkirališta

Parkirališno mjesto Ovaj entitet sadrži sve bitne informacije o parkirališnom mjestu. Atributi su mu: korisnički id voditelja parkirališta, broj parkirališnog mjesta na parkiralištu, tip parkirališnog mjesta, mogućnost rezervacije te x i y koordinate za 4 točke koje označavaju granice parkirališnog mjesta. Povezan je vezon 1 naprema n sa parkiralištem i rezervacijom a vezom n(zauzetost) naprema 1(mjesto) sa zauzetosti parkirališnog mjesta.

Parkirališno mjesto(parkingspot)		
userid	INT	Jedinstveni identifikator voditelja parkinga
parkingspotnumber	INT	Broj parkirališnog mjesta na parkiralištu
parkingspottype	VARCHAR	Vrsta parkirališnog mjesta(Za automobile ili za bicikle)
canbereserved	BOOLEAN	Mogućnost rezerviranja parkirališnog mjesta
pointx1	NUMERIC	Koordinata x prve točke parkirnog mjesta
pointy1	NUMERIC	Koordinata y prve točke parkirnog mjesta
pointx2	NUMERIC	Koordinata x druge točke parkirnog mjesta
pointy2	NUMERIC	Koordinata y druge točke parkirnog mjesta

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Parkirališno mjesto(parkingspot)		
pointx3	NUMERIC	Koordinata x treće točke parkirnog mjesta
pointy3	NUMERIC	Koordinata y treće točke parkirnog mjesta
pointx4	NUMERIC	Koordinata x četvrte točke parkirnog mjesta
pointy4	NUMERIC	Koordinata y četvrte točke parkirnog mjesta

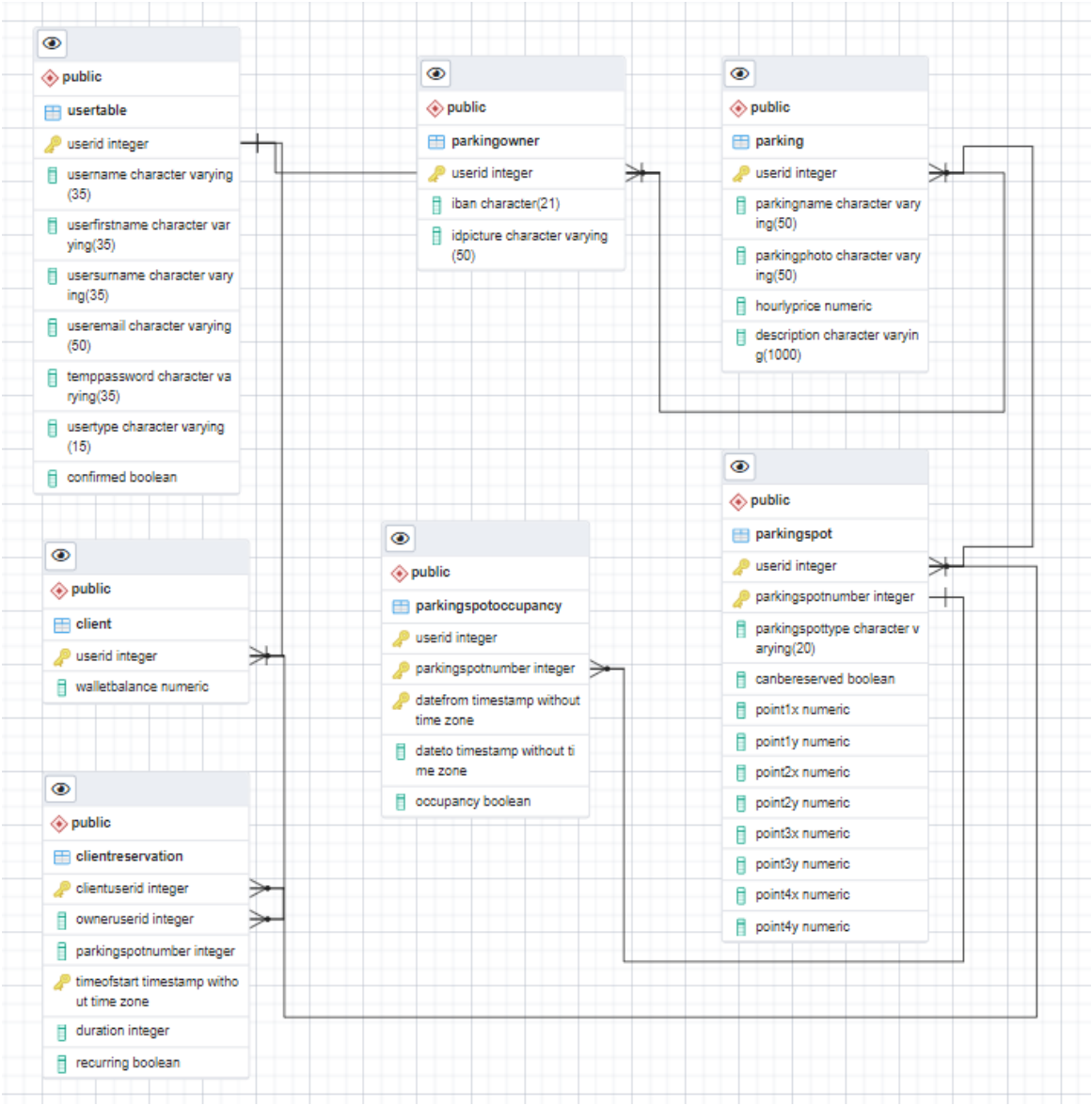
Zauzetost parkirališnog mjesta Ovaj entitet sadrži informacije koje se koriste za statističku analizu zauzetosti parkirališnog mjesta. Atributi su mu: korisnički id voditelja parkirališta, broj parkirališnog mjesta na parkiralištu, vrijeme od, vrijeme do i zauzetost u tom razdoblju. Povezan je vezom 1(mjesto) naprema n(zauzetost) sa parkirališnim mjestom.

Zauzetost parkirnog mjesta(parkingspotoccupancy)		
userid	INT	Jedinstveni identifikator voditelja parkinga
parkingspotnumber	INT	Broj parkirnog mjesta na parkiralištu
datefrom	TIMESTAMP	Početak zauzetosti parkirališnog mjesta
dateto	TIMESTAMP	Kraj zauzetosti parkirališnog mjesta
occupancy	BOOLEAN	Varijabla koja nam govori da li je parkirališno mjesto bilo zauzeto

Rezervacija Ovaj entitet sadrži informacije o rezervacijama parkirališnih mjesta. Sadrži atribute: jedinstveni identifikator klijenta, vrijeme početka rezervacije, jedinstveni identifikator voditelja parkinga, broj parkirnog mjesta na parkiralištu, duljina rezervacije i varijablu koja nam govori je li rezervacija ponavljajuća. Povezan je vezon 1(klijent) naprema n(rezervacija) sa klijentom i vezom 1(mjesto) naprema n(rezervacija) sa parkirališnim mjestom.

Rezervacija(clientreservation)		
userid	INT	Jedinstveni identifikator klijenta
timeofstart	TIMESTAMP	Vrijeme početka rezervacije
userid	INT	Jedinstveni identifikator voditelja parkinga
parkingspotnumber	INT	Broj parkirnog mjesta na parkiralištu
duration	INT	Duljina rezervacije
reccuring	BOOLEAN	Varijabla koja definira je li rezervacija ponavljajuća

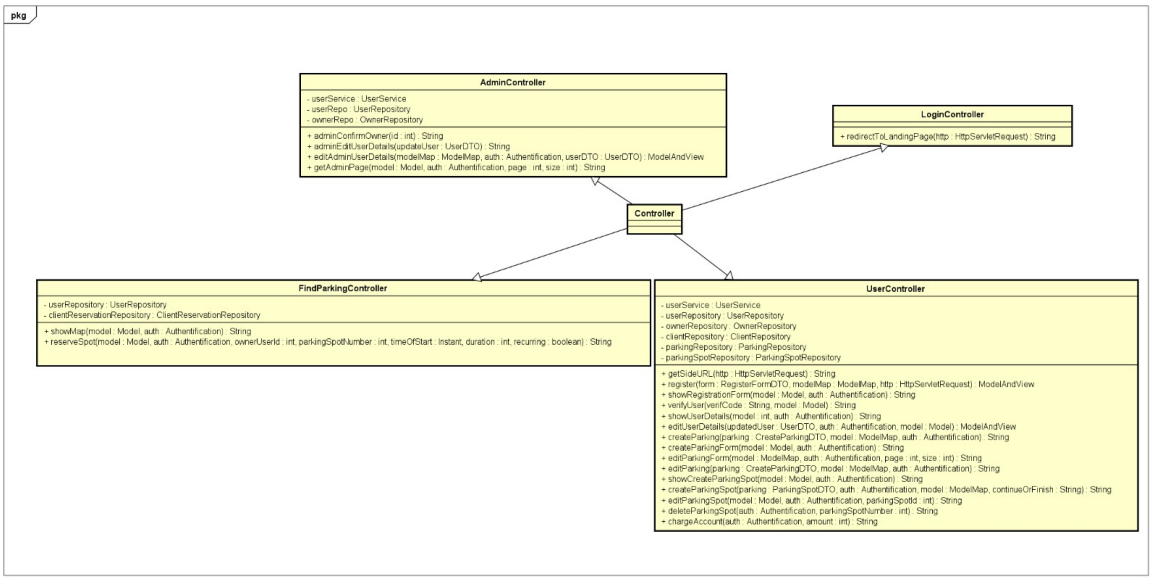
4.1.2 Dijagram baze podataka



Slika 4.2: E-R dijagram baze podataka

4.2 Dijagram razreda

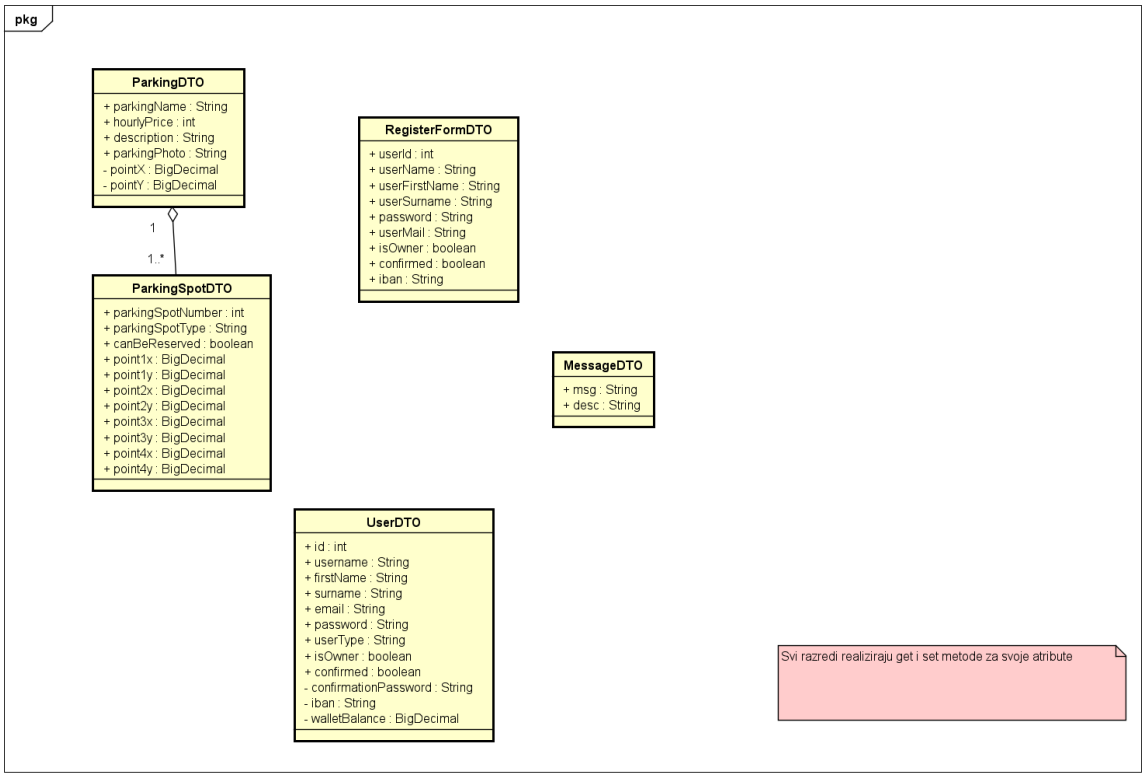
Na slikama 4.3, 4.4 i 4.5 su prikazani razredi koji pripadaju backend dijelu MVC arhitekture. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Metode implementirane u tim razredima manipuliraju s DTO (Data transfer object). Metode implementirane u Controller razredima vraćaju objekt tipa String kojim se framework povezuje sa frontendom. Zbog lakše organizacije, razredi su podijeljeni logički po pravu pristupa metodama određenih aktora. Iz naziva i tipova atributa u razredima moze se zaključiti vrsta ovisnosti među različitim razredima.



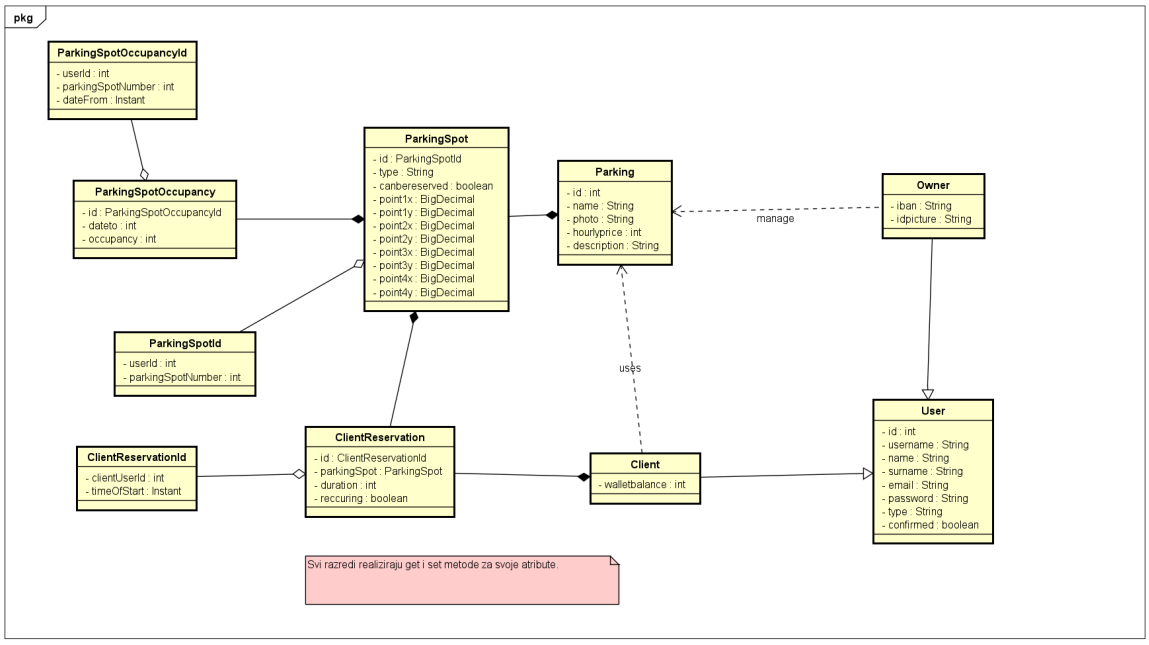
Slika 4.3: Dijagram razreda - dio Controllers

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Razred User predstavlja neregistriranog korisnika koji se moze registrirati u sustav unoseći osnovne informacije. Razred Client predstavlja klijenta koji je registriran u sustav i koji može koristiti njegove osnovne funkcionalnosti. Razred Owner predstavlja voditelja parkinga koji ima mogućnost upravljanja svojim parkiralištima i parkirališnim mjestima. Razred Parking prikazuje parking sa definirajućim atributima i parkirališnim mjestima koji su zaseban razred. Razred ClientReservation služi za ispis statističkih podataka. Razredi ParkingSpot, ParkingSpotOccupancy i ClientReservation imaju složene ključeve koji su izdvojeni u zasebne razrede.

1



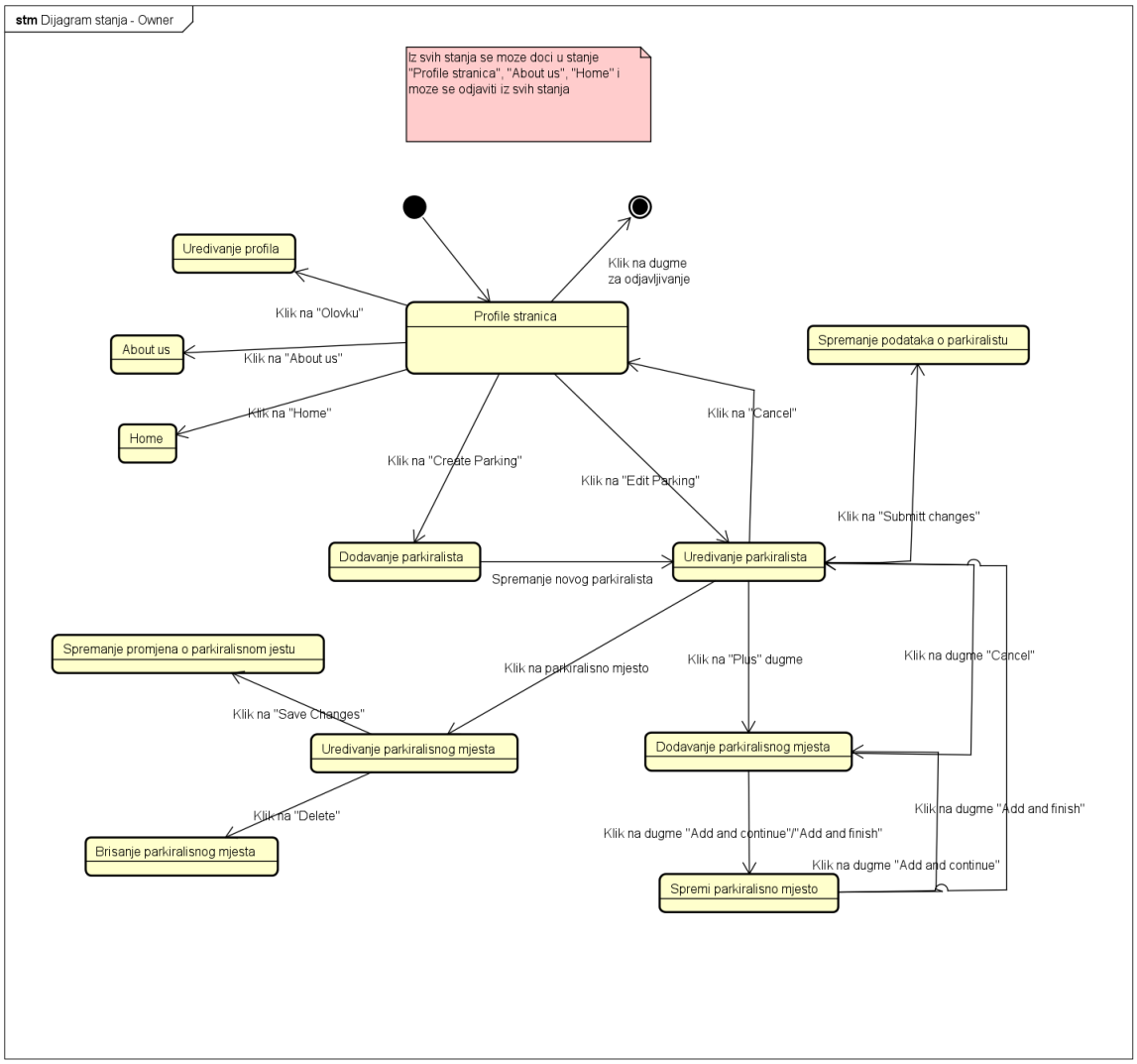
Slika 4.4: Dijagram razreda - dio Data transfer objects



Slika 4.5: Dijagram razreda - dio Models

4.3 Dijagram stanja

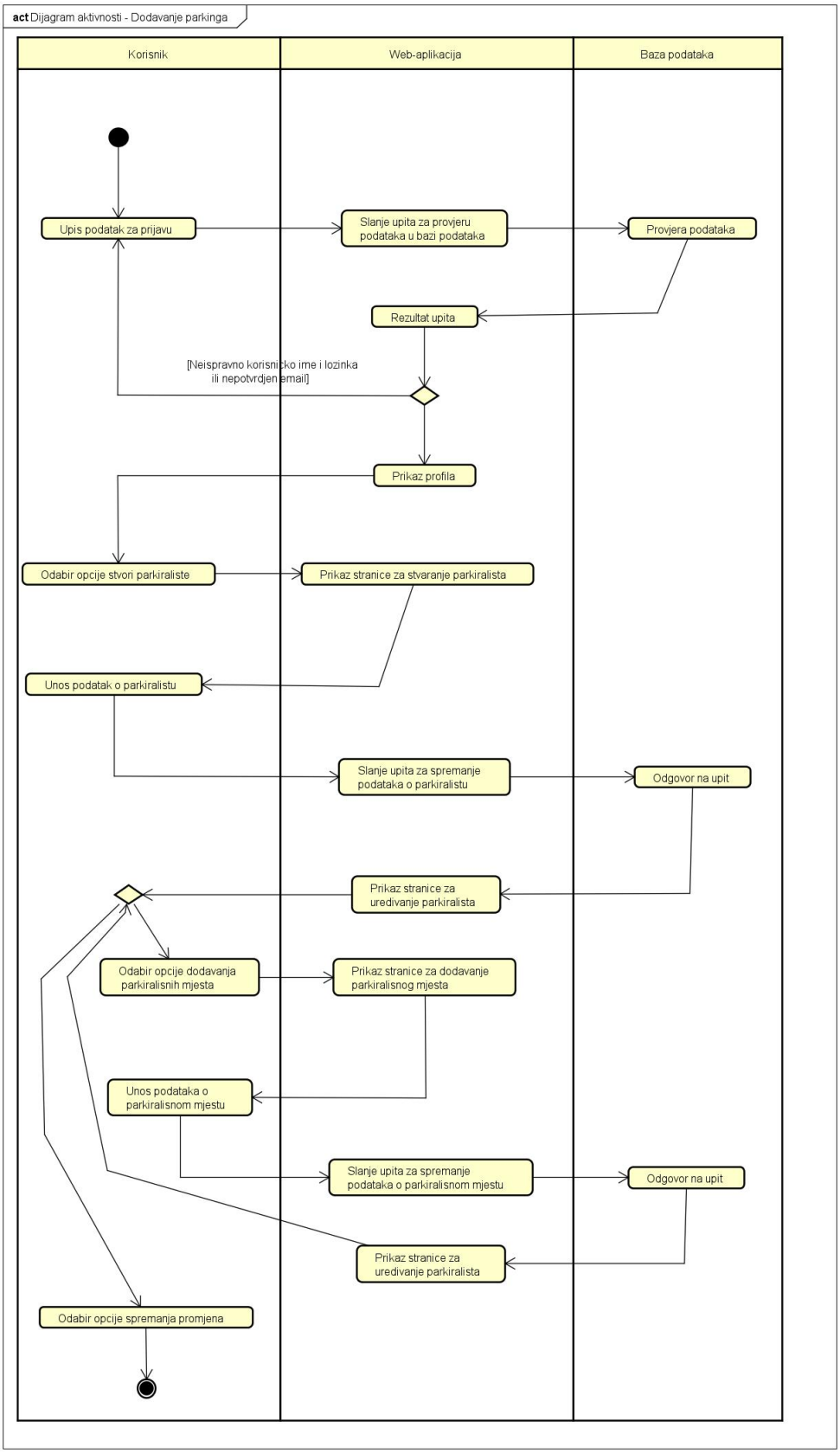
Dijagrami stanja pokazuju različita stanja entiteta. Također mogu pokazati kako entitet reagira na različite događaje mijenjajući se iz jednog stanja u drugo. Dijagram stanja je UML dijagram koji se koristi za modeliranje dinamičke prirode sustava. Obično se koristi za opisivanje ponašanja objekta ovisno o stanju. Dijagrami stanja obično se primjenjuju na objekte, ali se mogu primijeniti na bilo koji element koji ima ponašanje prema drugim entitetima kao što su: akteri, slučajevi upotrebe, metode, sustavi podsustava i itd. i obično se koriste zajedno s dijagramima interakcije (obično dijagrami sekvenci).
U nasem dijagramu stanja prikazujemo kako se mijenjaju stranice za voditelja parkinga.



Slika 4.6: Dijagram stanja

4.4 Dijagram aktivnosti

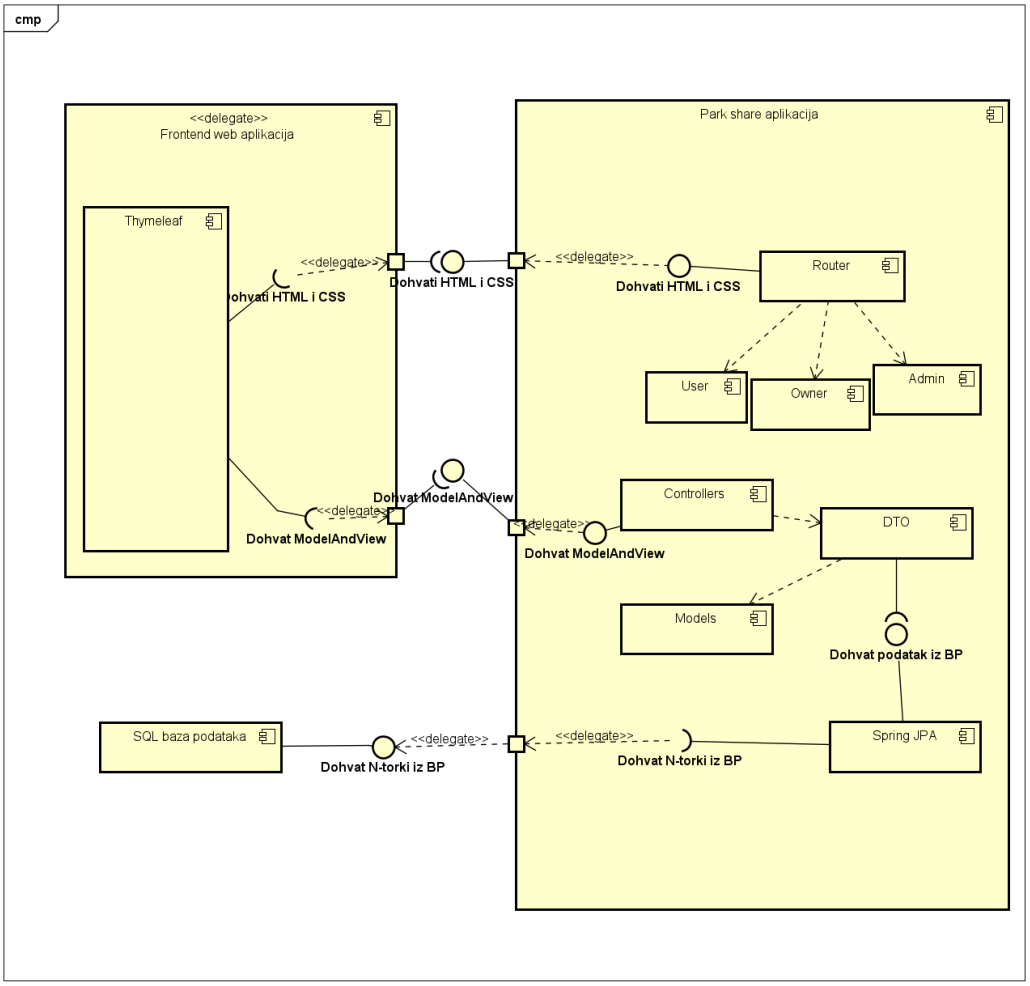
Dijagram aktivnosti je UML dijagram ponašanja. Predstavlja kako se svaka aktivnost odvija jedan za drugim. Aktivnost je neka vrsta operacije sustava. Nadalje, dijagrami aktivnosti pomažu poslovnim i razvojnim timovima organizacije da razumiju procese i ponašanje sustava.
Na nasem dijagramu aktivnosti prikazan je proces kreiranja parkiralista i dodavanja parkiralisnih mjesta u samo parkiraliste.



Slika 4.7: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti koristi se za prikaz velikog objektno orijentiranog sustava, kako bi se njima lakše upravljalo. On vizualizira odnose kao i organizaciju između komponenti prisutnih u sustavu. Pomaže u formiranju izvršnog sustava. Komponenta je jedna jedinica sustava, koja je zamjenjiva i izvršna. Detalji implementacije komponente su skriveni i potrebno je sučelje za izvršavanje funkcije. To je poput crne kutije čije je ponašanje objašnjeno predviđenim i potrebnim sučeljima. Router je komponenta koja na upit s url odreduje koja datoteka ce se posluziti na sucelje. Preko sucelja za dohvat ModelAndView posluzuju se podatci sa backenda koji su preko spring JPA-a sa suceljem za dohvat N-torki dohvaceni iz baze podataka.



Slika 4.8: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Za komunikaciju medu clanovima tima smo koristili discord[1]i whatsapp[2]. Za izradu Uml dijagrama korisitli smo Astah[3]. Za upravljanje izvornim kodom koristili smo Git[4], a udaljeni repozitorij nalazio se na GitLabu[5].
Razvojno okruzenje nam je bio Intellij[6]. IntelliJ IDEA je integrirano razvojno okruženje (IDE) napisano u Javi za razvoj računalnog softvera. Razvio ga je JetBrains (prije poznat kao IntelliJ), a dostupan je besplatno preko licence za studente za koju smo se prijavili sa FER-ovim mailom.
Web aplikaciju smo napisali u Javi[7] koristeci Spring Boot[8] framework za izradu backenda, a thymeleaf[9], html, css i javascript[10] za izradu frontenda.
Spring Boot je open-source framework koji održava tvrtka pod nazivom Pivotal. Pruža Java programerima platformu za početak rada s aplikacijom koja se može automatski konfigurirati.
Thymeleaf je Java teamplate engine koji omogucuje rad sa HTML, XML, JavaScript i CSS datotekama.

[1] https://discord.com/download
[2] https://www.whatsapp.com/download
[3] https://astah.net/products/astah-professional/
[4] https://git-scm.com/downloads
[5] https://about.gitlab.com/install/
[6] https://www.jetbrains.com/idea/download
[7] https://www.java.com/download/ie_manual.jsp
[8] https://spring.io/projects/spring-boot
[9] https://www.thymeleaf.org/
[10] https://www.javascript.com/

5.2 Ispitivanje programskog rješenja

Testove smo izvodili na strukturirani nacin pomocu selenium testova. Ispitivanje se radilo po obrascima uporabe kako bi se provjerila osnovna funkcionalnost sustava, ali i nasumicnim kretanjima po aplikaciji kako bi se pronasle neocekivane greske ili nepredvidena ponasanja.

5.2.1 Ispitivanje komponenti

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

5.2.2 Ispitivanje sustava

Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium[11]*. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

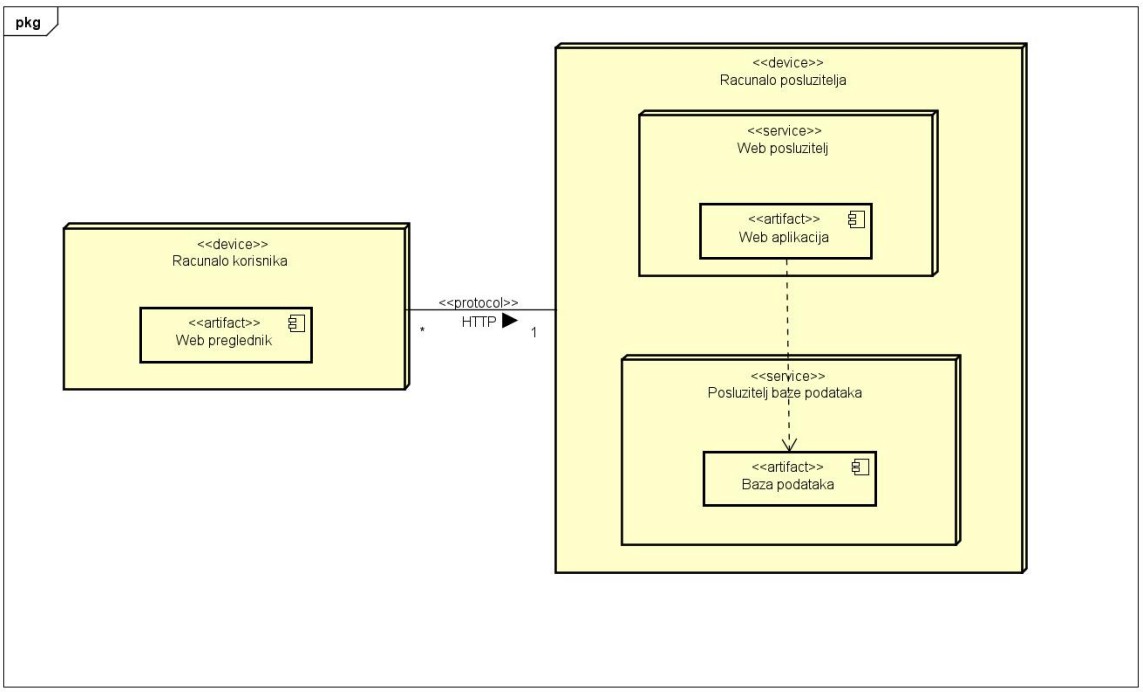
- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

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

[11] https://www.seleniumhq.org/

5.3 Dijagram razmještaja

Dijagram razmjestaja je strukturni statički UML dijagram koji opisuje topologiju sustava i usredotočen je na odnos sklopovskih i programskih dijelova. Imamo dijagram razmještaja, specifikacijski dijagram razmještaja, dijagram razmještaja instanci, implementacijski dijagram i dijagram mrežne arhitekture.



Slika 5.1: Dijagram razmjestaja

5.4 Upute za puštanje u pogon

Pustanje aplikacije u pogon odradeno je koristenjem alata Heroku. Posluziteljski i klijentski dio podignuti su zasebno. Za dizanje posluziteljskog dijela pratili smo upute na stranici [12]. U projektu radimo s PostgreSql bazom podataka koja je podrzana na Heroku platformi. Integrira se u projekt dodavanjem Heroku Postgres Add-Onsa te se dobiju podatci za povezivanje s tom bazom koji se upisuju u application.properties u projektu.

Slika 5.2: Add-Ons

Prvo treba skinuti Heroku CLI(Command Line Interface) preko kojeg se koristi naredba za prijavu na Heroku. Nakon toga treba pripremiti repozitorij za dizanje na Heroku koristenjem git naredbi.

```
$ git init
$ git add .
$ git commit -m "first commit"
```

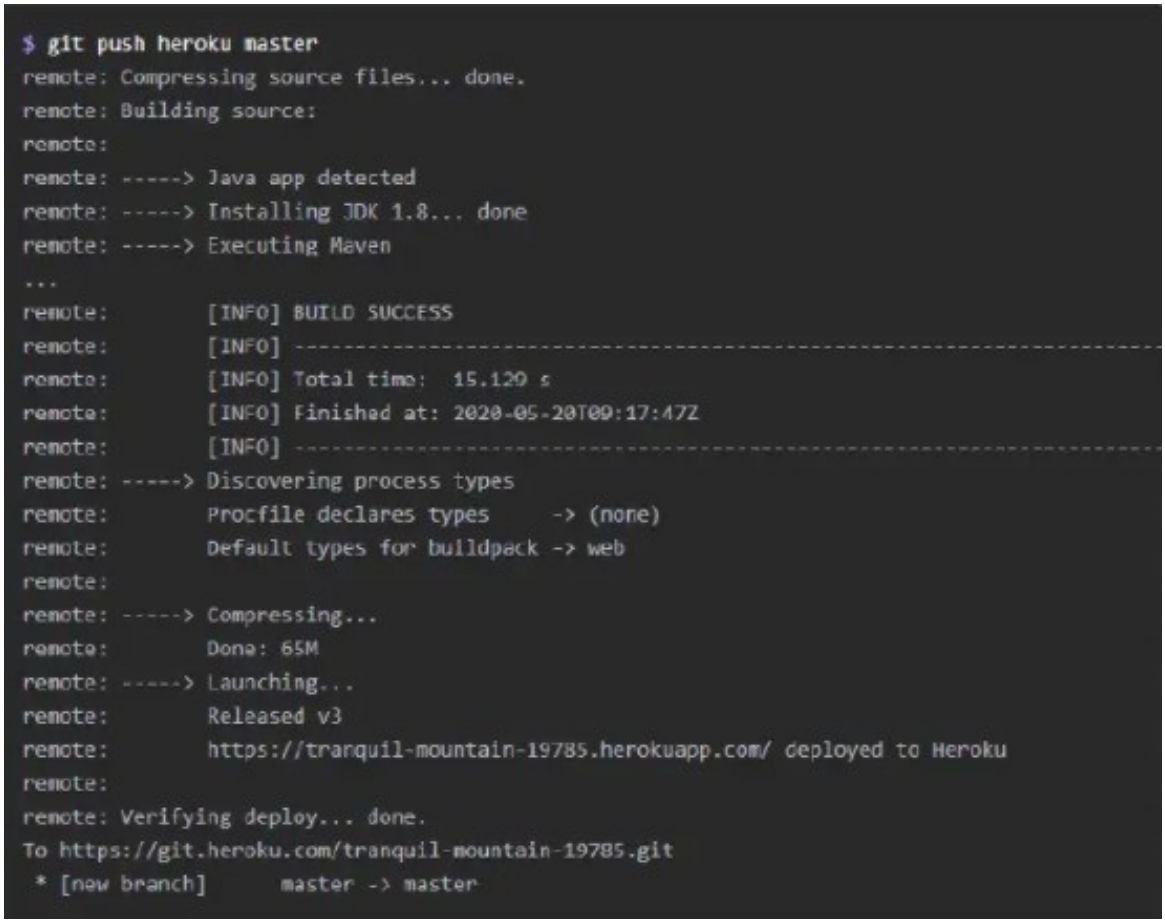
Slika 5.3: Heroku CLI

Zatim treba kreirati udaljeni Heroku repozitorij i konacno podignuti nas kod na taj repozitorij sto je prikazano na slikama 5.4 i 5.5

```
$ heroku create
Creating app... done, tranquil-mountain-19785
https://tranquil-mountain-19785.herokuapp.com/ | https://git.heroku.com/tranquil-mountain-197
```

Slika 5.4: Stvaranje Heroku repozitorija

[12] https://devcenter.heroku.com/articles/deploying-spring-boot-apps-to-heroku



Slika 5.5: Deplojanje posluziteljskog koda na Heroku

Nasoj aplikaciji mozete pristupiti na linku [13]

[13] https://park-share-proggers.herokuapp.com/

6. Zaključak i budući rad

Zadatak nase grupe bio je razvoj web aplikacije za online rezerviranje parkiralisnog mjesta uz mogucnost upravljama korisnicima, voditeljima parkiralista, parkiralistima, parkiralisnim mjestima. Nakon 14 tjedana rada u timu i razvoja web aplikacije djelomicno smo uspjeli u ostvarenju naseg cilja. Provedba projekta je tekla kroz dvije faze.
U prvoj fazi smo se okupili kao tim i upoznali, dodijeljen nam je projektni zadatak i radili smo na dokumentaciji projekta. Zbog dobro provedene prve faze uvelike nam je olaksala daljnji put ka realizaciji web aplikacije. Izradeni obrasci, dijagrami i baza podataka bili su od velike pomoci svim sudionicima na projektu pri osmisljanju rijesenja i rijesavanju projektnog zadatka.
Druga faza projekta, je bila dosta teza od prve faze zbog nedostatka vremena i slabijeg znanja koristenih alata clanovi tima su gubili dosta vremena pri ucenju samih alata i pri debuggiranju koda. Osim realizacije rjesenja, u drugoj fazi je bilo potrebno dokumentirati ostale UML dijagrame i izraditi popratnu dokumentaciju kako bi buduci korisnici mogli lakse koristiti ili vrsiti preinake na sustavu. Clanovi tima su komunicirali putem whatsapp i discord grupa gdje smo obavijestavali ostale clanove o tijeku rada na projektu. Sudjelovanje na ovom projektu bilo je vrijedno i na neki nacin naporno iskustvo svim kolegama u timu jer smo kroz intenzivnih nekoliko tjedana rada iskusili zajednicki rad na istom projektu. Zbog losije vremenske organiziranosti imali smo manjih problema no sve u svemu dosta smo naucili i dosta naucenog ce mo koristiti u daljnjoj karijeri. Za nastavak rada prvo bi dovrsili cijeli zadatak, a onda bi ugradili i pravi sustav za naplatu.
Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji

Popis literature

Kontinuirano osvježavanje

Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

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

Indeks slika i dijagrama

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

1. sastanak
   - Datum: 14.10.2021.
   - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
   - Teme sastanka:
     * Prvi sastanak tima
     * Formiranje grupe
2. sastanak
   - Datum: 26.10.2021.
   - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
   - Teme sastanka:
     * Raspodjela na frontend i backend
     * Raspodjela rada na dokumentaciji
3. sastanak
   - Datum: 2.11.2021.
   - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
   - Teme sastanka:
     * Rasprava o stanju na projektu
4. sastanak
   - Datum: 15.11.2021.
   - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
   - Teme sastanka:

        * Rasprava o stanju na projektu
        * Pregled dokumentacije
        * Dogovor o deployju aplikacije
5. sastanak
    - Datum: 3.12.2021.
    - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
    - Teme sastanka:
        * Rasprava o stanju na projektu
        * Raspodijela poslova
        * Dogovor o razvoju alfa verzije
6. sastanak
    - Datum: 13.12.2021.
    - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
    - Teme sastanka:
        * Rasprava o stanju na projektu
        * Rjesavanje problema na projektu
7. sastanak
    - Datum: 4.1.2022.
    - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
    - Teme sastanka:
        * Rasprava o stanju na projektu
        * Pregled dokumentacije
        * Dogovor o razvoju zavrsne verzije aplikacije
8. sastanak
    - Datum: 14.1.2022.
    - Prisustvovali: M.Barbir, M.Ivić, F.Bura, M.Žura, M.Puharić, L.Gjurić, Z.Ravlić
    - Teme sastanka:
        * Rasprava o stanju na projektu
        * Pregled dokumentacije
        * Dogovor o deploju aplikacije
        * Rjesavanje problema

Tablica aktivnosti

Kontinuirano osvježavanje

	Marko Barbir	**Luka Gjurić**	**Marko Žura**	**Filip Bura**	**Marin Puharić**	**Zvonimir Ravlić**	**Mate Ivić**
Upravljanje projektom	7	5	5	5	8	2	2
Opis projektnog zadatka	0	0	12	0	0	0	0
Funkcionalni zahtjevi	2	0	0	0	4	0	0
Opis pojedinih obrazaca	1	0	0	4	1.5	5	4
Dijagram obrazaca	0	0	0	0	0	15	10
Sekvencijski dijagrami	0	0	0	0	0	0	7
Opis ostalih zahtjeva	0	0	4	0	0	0	0
Arhitektura i dizajn sustava	0.5	1	1	1	0.5	5	1
Baza podataka	2	0	0	0	0	9	0
Dijagram razreda	0	0	0	0	0	0	11
Dijagram stanja	0	0	0	0	0	0	0
Dijagram aktivnosti	0	0	0	0	0	0	0
Dijagram komponenti	0	0	0	0	0	0	0
Korištene tehnologije i alati	0	0	0	0	0	0	0
Ispitivanje programskog rješenja	0	0	0	0	0	0	0
Dijagram razmještaja	0	0	0	0	0	0	0
Upute za puštanje u pogon	0	0	0	0	0	0	0
Dnevnik sastajanja	0.5	1	1	0.5	0.5	0	1

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Marko Barbir	**Luka Gjurić**	**Marko Žura**	**Filip Bura**	**Marin Puharić**	**Zvonimir Ravlić**	**Mate Ivić**
Zaključak i budući rad	0	0	0	0	0	0	0
Popis literature	0	0	0	0	0	0	0
Izrada početne stranice	10	15	1	0	10	0	2
Izrada baze podataka	5	0	0	0	0	3	0
Spajanje s bazom podataka	4	2	0	5	4	0	0
Back end	60	15	21	20	80	16	17
Front end	30	60	10	0	10	11	10
Deployment na Heroku	2	0	0	15	0	0	0

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.