

# Programsko inženjerstvo

Ak. god. 2020./2021.

## Parkiraj Me

Dokumentacija, Rev. 1.0

Grupa: NDB

Voditelj: Daniel Ranogajec

Datum predaje: 13. 11. 2020.

Nastavnik: Nikolina Frid

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>2</b>
<b>2 Opis projektnog zadatka</b>	<b>3</b>
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	10
3.1.2 Sekvencijski dijagrami . . . . .	24
3.2 Ostali zahtjevi . . . . .	29
<b>4 Arhitektura i dizajn sustava</b>	<b>30</b>
4.1 Baza podataka . . . . .	30
4.1.1 Opis tablica . . . . .	31
4.1.2 Dijagram baze podataka . . . . .	34
4.2 Dijagram razreda . . . . .	36
4.3 Dijagram stanja . . . . .	40
4.4 Dijagram aktivnosti . . . . .	41
4.5 Dijagram komponenti . . . . .	42
<b>5 Implementacija i korisničko sučelje</b>	<b>43</b>
5.1 Korištene tehnologije i alati . . . . .	43
5.2 Ispitivanje programskog rješenja . . . . .	44
5.2.1 Ispitivanje komponenti . . . . .	44
5.2.2 Ispitivanje sustava . . . . .	44
5.3 Dijagram razmještaja . . . . .	45
5.4 Upute za puštanje u pogon . . . . .	46
<b>6 Zaključak i budući rad</b>	<b>47</b>
<b>Popis literature</b>	<b>48</b>
<b>Indeks slika i dijagrama</b>	<b>49</b>

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Jovanović	13.10.2020
0.2	Opis projektnog zadatka	Jovanović Marušić Fabijanić	16.10.2020
0.3	Specifikacije programske potpore, funkcionalni zahtjevi, obrasci uporabe i dijagrami obarazaca uporabe	Jovanović Erceg Fabijanić Vuica	17.10.2020
0.3.1	Uvod u sekvencijske dijagrame	Jovanović	19.10.2020
0.3.2	Sekvencijski dijagrami i njihovi opisi	Erceg	23.10.2020
0.4	Baza podataka	Briševac Marušić Ranogajec	24.10.2020
0.4.1	Dijagram razreda	Briševac Marušić	24.10.2020
0.5	Ostali zahtjevi	Jovanović	31.10.2020
0.6	Prepravljanje opisa projektnog zadatka	Fabijanić	9.11.2020.
0.7	Prepravljanje specifikacija programske potpore	Vuica	10.11.2020.
0.8	Revizija pred predaju	Marušić Jovanović Fabijanić	13.11.2020.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	na	na

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije „Parkiraj Me“ koja korisnicima omogućuje praćenje stanja parkirališnih mjesta u gradu te njihovu rezervaciju. Aplikacija će objedinjavati parkirališne površine diljem Zagreba svih tvrtki zainteresiranih za ponudu svojih površina putem aplikacije. Ovakvim rješenjem bi se uvelike olakšalo nalaženje parkirališnih mjesta u gusto naseljenim urbanim sredinama među koje se svrstaje i grad Zagreb.

Prilikom pokretanja aplikacije prikazuje se karta s ucrtanim parkirališnim površinama koje se nalaze u blizini. Pomoću pametnih kamera se omogućuje praćenje zauzeća pojedinih parkirališnih mjesta.

Sustav je povezan sa Google Maps uslugom, tj. pomoću toga će se korisnicima prikazivati karta i lokacije parkinga na njoj. Vezano za tu uslugu implementirana je i opcija prijedloga parkirališnih površina na temelju trenutne lokacije korisnika. Kod davanja prijedloga u situaciji kada je na geografski najbližoj lokaciji slobodan mali broj mjesta, prednost je dana sljedećoj najbližoj lokaciji s većim brojem slobodnih mjesta, jer se može dogoditi da kad vozač stigne do parkinga više nema slobodnih mjesta.

Neregistriranom korisniku odabirom parkirališne površine prikazuju se informacije o statusu parkirališne površine:

- ukupan broj parkirališnih mjesta
- broj zauzetih mjesta
- broj slobodnih mjesta
- cijena za 1h parkinga

Također se prikazuje „grayed-out“ opcija 'REZERVIRAJ'. Ona se omogućuje kada se neregistriran korisnik uspješno registrira i prijavi u sustav.

Svaki punoljetni neregistrirani korisnik ima opciju registracije. Moguće se registrirati kao privatni korisnik, tj. vozač ili kao tvrtka koja nudi svoja parking mjesta na svojim parkirališnim površinama.

Pri postupku registracije za obe vrste računa unose se slijedeće podatci:

- adresa e-pošte
- korisničko ime
- lozinka
- broj telefona

Korisnici koji žele napraviti račun kao vozač dodatno unose slijedeće podatke:

- broj kreditne kartice
- registarsku oznaku jednog ili više automobila za koje će vršiti rezervacije

Korisnici koji žele napraviti račun kao vlasnici parking površina dodatno unose slijedeće podatke:

- informacije o jednoj ili više parking lokacija
  - adresa lokacije
  - broj mjesta
  - cijena za pojedinu vrstu rezervacije (vrste rezervacije su kasnije detaljnije objašnjene)
- IBAN računa za isplatu

Što se tiče samih rezervacija postoje tri opcije:

- jednokratna za vremenski period kraći od 24 sata koja se mora obaviti barem 6h unaprijed
  - naplata u trenutku rezervacije
- ponavljajuća rezervacija koja mora trajati najmanje 1h i ponavljati se barem jednom tjedno tijekom mjesec dana
  - naplata svakih 30 dana
- trajna rezervacija mjesta (0-24h svaki dan na neodređeni period)

Naplata će se vršiti po principu da korisnici plaćaju direktnom uplatom aplikaciji što je ostvareno pomoću vanjskog servisa naplate. Dalje se te uplate prosljeđuju vlasnicima parking lokacija.

Vozač ima slijedeće mogućnosti prilikom korištenja aplikacije:

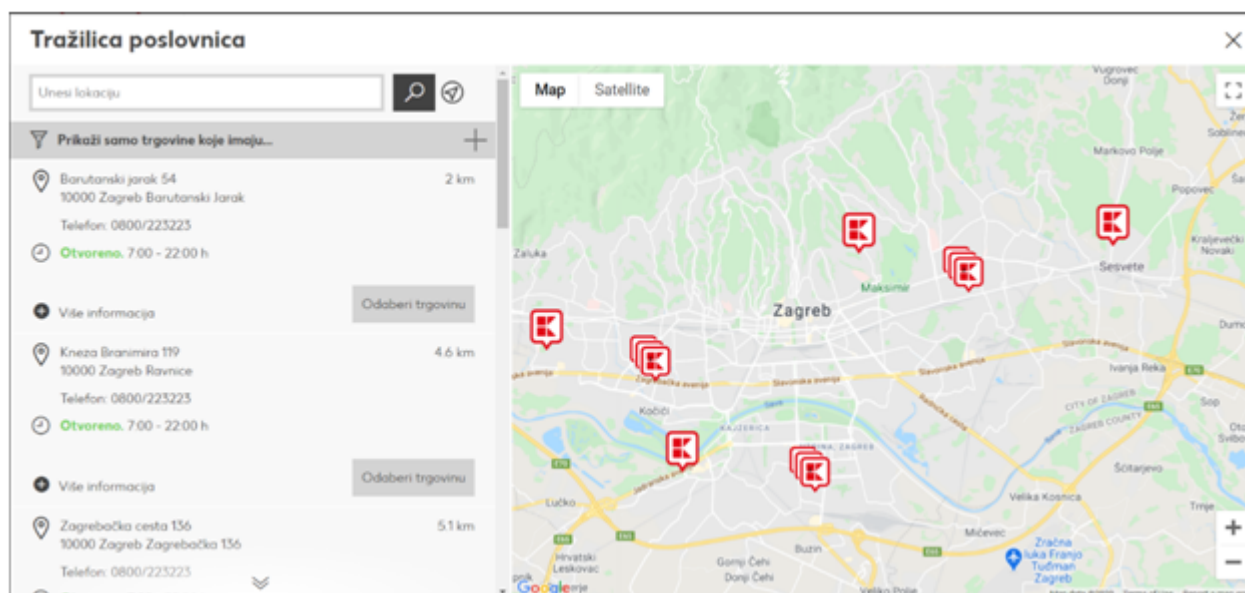
- rezervacija parking mjesta
- pregled i izmjena:
  - osobnih podataka
  - registriranih vozila
  - aktivnih rezervacija

Tvrtkama se omogućuju slijedeće mogućnosti prilikom korištenja aplikacije:

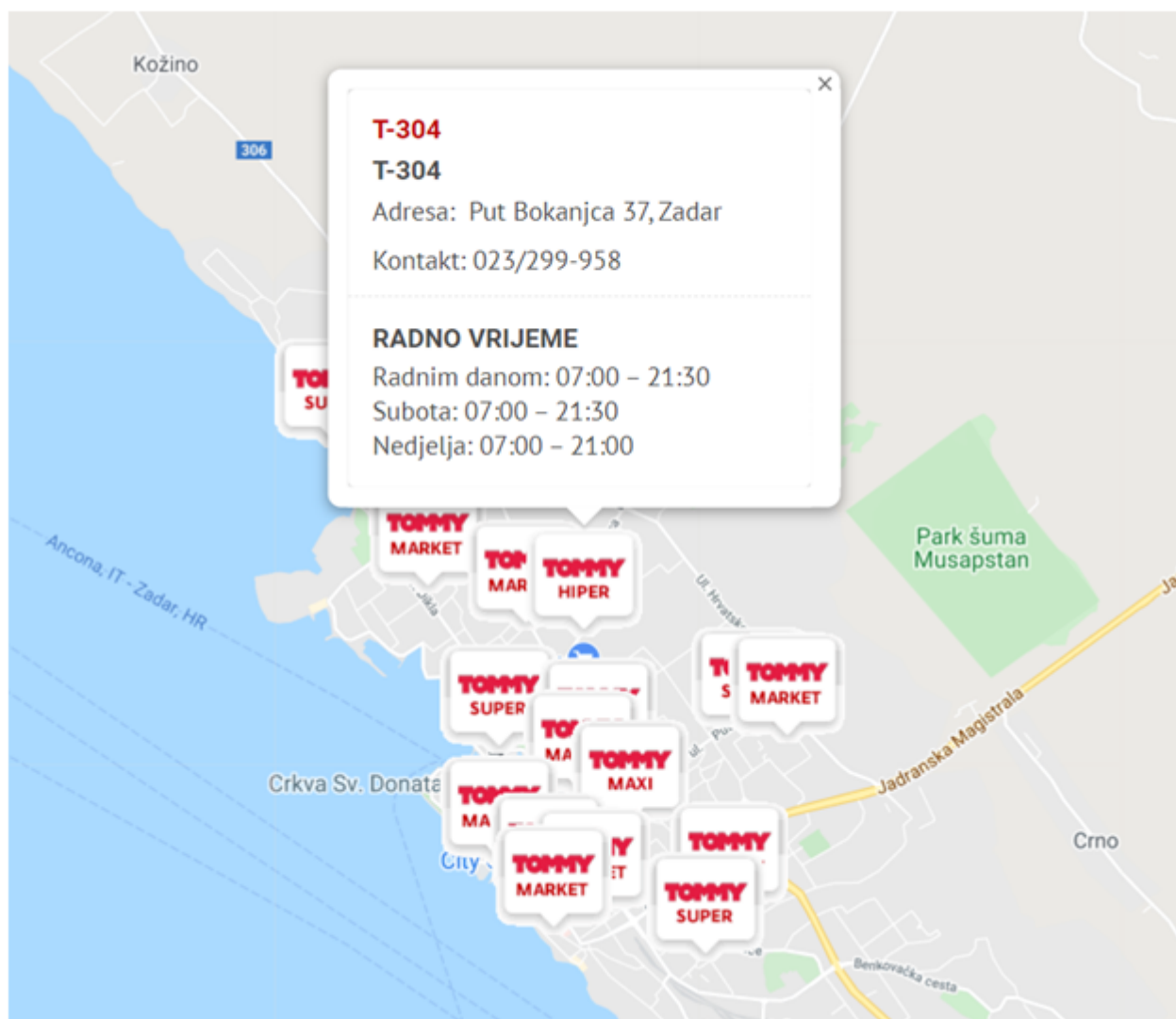
- pregled i izmjena:
  - osobnih podataka o računu
  - prijavljenih parking površina
  - postavljenih cijena rezervacija

Administrator je zadnji tip korisnika. Administrator sustava ima najveće ovlasti te može upravljati i računima klijenata i računima tvrtki. Ima mogućnost i brisanja tih računa. Prilikom brisanja računa klijenata, ako ima neiskorištenih rezervacija uplata bi se refundirala za preostalo (neiskorišteno) vrijeme. Prilikom brisanja računa tvrtki obrisale bi se i sve njihove parkirališne površine, a sve preostale neiskorištene rezervacije bi se otkazale i refundirale.

Slična implementacija lokacija na karti i prikazivanja nekih informacija o njima prikazana je na donjim slikama.



Slika 2.1: Odabir poslovnica tvrtke Kaufland, <https://www.kaufland.hr/usluge/poslovnica.html>



Slika 2.2: Odabir poslovnica tvrtke Tommy, <https://tommy.hr/hr/prodajna-mjesta>



## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

Dionici:

1. Tvrtke
2. Korisnici
  - (a) Neregistrirani
  - (b) Vozač
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Tvrtka (inicijator) može:
  - (a) registrirati se u aplikaciji unoseći OIB, ime, adresu sjedišta, adresu e-pošte
  - (b) prijaviti, pregledati, urediti i obrisati parkirališnu površinu
2. Neregistrirani korisnik (inicijator) može:
  - (a) pregledati parkirališne površine
  - (b) stvoriti korisnički račun unoseći OIB, ime, prezime, adresu e-pposte, broj registracije svojeg automobila i broj kreditne kartice
  - (c) dobiti ovlasti registriranog korisnika
3. Vozač (inicijator) može:
  - (a) pregledati parkirališne površine
  - (b) pregledati i promijeniti osobne podatke
  - (c) pregledati, dodati, obrisati registarsku oznaku automobila u aplikaciju
  - (d) dobiti prijedlog najbližih parkirališnih površina za rezervaciju
  - (e) rezervirati parkirališno mjesto

- (f) platiti rezervaciju
- (g) pregledati, urediti, obrisati rezervacije

4. Administrator (inicijator) može:

- (a) vidjeti popis odabranih korisnika
- (b) urediti i obrisati odabrane korisnike
- (c) vidjeti popis odabranih tvrtki
- (d) urediti i obrisati odabrane tvrtke

5. Baza podataka (inicijator) može:

- (a) pohranjuje sve podatke o korisnicima
- (b) pohranjuje sve podatke o tvrtkama

### 3.1.1 Obrasci uporabe

#### UC1 - Pregled parkirališnih površina

- **Glavni sudionik:** Vozač, neregistrirani korisnik
- **Cilj:** Pregledati parkirališne površine
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Karta je prikazana prilikom otvaranja aplikacije
  2. Korisnik na karti odabire parkirališnu površinu
  3. Prikazuju se informacije o odabranoj površini

#### UC2 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik, Tvrtka
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju (vozač ili tvrtka)
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Odabir već zauzetog korisničkog imena i/ili e-pošte, unos korisničkog podataka u nedozvoljenom formatu ili pružanje neispravne e-pošte
    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

#### UC3 - Prijava u sustav

- **Glavni sudionik:** Vozač
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
  1. Unos korisničkog imena i lozinke

2. Potvrda ispravnosti unesenih podataka
3. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
  - 2.a Neispravno uneseni podatci
    1. Sustav obavještava korisnika o neuspjeloj prijavi

#### UC4 - Pregled osobnih podataka

- **Glavni sudionik:** Vozač, Tvrtka
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „osobni podatci“
  2. Aplikacija prikazuje osobne podatke korisnika

#### UC5 - Promjena osobnih podataka

- **Glavni sudionik:** Vozač, tvrtka
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za promjenu podataka
  2. Korisnik mijenja svoje osobne podatke
  3. Korisnik sprema promjene
  4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 2.a Korisnik promijeni svoje osobne podatke, ali ne odabere opciju „Spremi promjenu“
    1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora
  - 2.b Korisnik promijeni svoje osobne podatke, ali je nešto neispravno uneseno
    1. Sustav obavještava korisnika da je nemoguće izmijeniti podatke pri likom odabira „Spremi promjene“

#### UC6 - Registracija auta

- **Glavni sudionik:** Vozač
- **Cilj:** Registrirati automobil
- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen
- **Opis osnovnog tijeka:**
  1. Vozač odabire opciju za registrirati auto
  2. Vozač unosi potrebne podatke
  3. Vozač prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
  - 2.a Unesena neispravna/već registrirana registracija
    1. Sustav obavijesti vozača da nije moguće registrirati auto, te ga vraća na stranicu za registraciju automobila
    2. Vozač mijenja podatka ili odustaje

#### UC7 - Pregled registriranih automobila

- **Glavni sudionik:** Vozač
- **Cilj:** Pregledati registrirane automobile
- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen
- **Opis osnovnog tijeka:**
  1. Vozač odabire opciju za pregledati registrirane automobile
  2. Aplikacija prikazuje automobile koje je vozač prijašnje unio ako postoje

#### UC8 - Promjena registriranih automobila

- **Glavni sudionik:** Vozač
- **Cilj:** Promijeniti registrirane automobile
- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen
- **Opis osnovnog tijeka:**
  1. Vozač odabere opciju za promjenu registriranih automobila
  2. Vozač mijenja registrirane automobile
  3. Vozač sprema promjene
  4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
  - 2.a Vozač izmjeni podatke, ali ne odabere opciju da spremi izmjene

1. Sustav obavještava vozača da nije spremio podatke prije izlaska iz prozora
- 3.a Vozač promijeni podatke, ali nešto je neispravno
  1. Sustav obavještava vozača da nije moguće spremiti promjene

#### **UC9 - Prijedlog najbližih slobodnih parkirališnih površina**

- **Glavni sudionik:** Vozač
- **Cilj:** Dobiti prijedlog najbližih parking mjesta za rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Vozač odabire opciju „Predloži parking mjesto“
  2. Sustav traži od vozača dopuštenje za korištenje trenutne lokacije
  3. Sustav prikaže nekoliko najbližih slobodnih parkirališnih površina vodeći računa o slobodnim mjestima
- **Opis mogućih odstupanja:**
  - 2.a Vozač ne dopusti pristup trenutnoj lokaciji
    1. Sustav obavještava vozača da nije moguće dati prijedlog najbližih parkirališnih površina sa slobodnim mjestima
  - 3.a Ne postoji slobodno parking mjesto
    1. Sustav obavještava vozača da su sva parking mjesta popunjena

#### **UC10 - Rezervacija parkirališnog mjesta**

- **Glavni sudionik:** Vozač
- **Cilj:** Rezervirati parking mjesto
- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen
- **Opis osnovnog tijeka:**
  1. Vozač odabere parkirališnu površinu na kojoj želi rezervirati mjesto
  2. Vozač odabere opciju „Rezerviraj mjesto“
  3. Vozač odabire vrstu rezervacije
  4. Vozač pregleda potrebne informacije za pojedinu vrstu rezervacije
  5. Vozač odabire mjesto koje želi rezervirati
  6. Vozač odabire opciju „Potvrdi rezervaciju“
  7. Sustav proslijedi vozača na naplatu rezervacije
- **Opis mogućih odstupanja:**

#### 4.a Na parkingu nema slobodnog mjesta

1. Sustav obavještava vozača da nema slobodnog mjesta i daje korisniku opciju mijenjanja datuma rezervacije

### UC11 - Naplata rezervacije

- **Glavni sudionik:** Vozač
- **Cilj:** Platiti rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** - Vozač je prijavljen i napravio je rezervaciju
- **Opis osnovnog tijeka:**
  1. Sustav prikazuje vozaču detalje rezervacije
  2. Vozač odabire „Plati i završi rezervaciju“
  3. Sustav naplati rezervaciju
  4. Sustav obavijesti vozača o uspješnoj rezervaciji i vrati ga nazad na home-page
- **Opis mogućih odstupanja:**
  - 2.a Vozač izađe iz prozora bez da je završio rezervaciju
    1. Sustav obavještava vozača da rezervacija nije dovršena i daje mu opciju da se vrati nazad u prozor ili odustane
  - 3.a Dode do greške pri naplati rezervacije
    1. Sustav obavještava vozača da rezervacija nije dovršena radi kritične pogreške pri obradi transakcije i vraća ga na home-page ili ponovi transakciju ako je moguće.

### UC12 - Pregled aktivnih rezervacija

- **Glavni sudionik:** Vozač
- **Cilj:** Pregledati aktivne rezervacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen
- **Opis osnovnog tijeka:**
  1. Vozač odabire opciju „Pregledaj aktivne rezervacije“
  2. Sustav prikaže vozaču aktivne rezervacije ukoliko takve postoje

### UC13 - Uređivanje aktivnih rezervacija

- **Glavni sudionik:** Vozač
- **Cilj:** Urediti rezervaciju

- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen i napravio je barem jednu rezervaciju
- **Opis osnovnog tijeka:**
  1. Vozač odabire opciju „Moje rezervacije“
  2. Sustav prikaže listu rezervacija vozača
  3. Vozač odabire željenu rezervaciju
  4. Prikaže se lista slobodnih mjesta
  5. Vozač izmjeni svoju rezervaciju i odabire „Spremi“
  6. Ažurira se baza podataka
- **Opis mogućih odstupanja:**
  - 4.a Nema slobodnih mjesta
    1. Sustav obavještava vozača da ne postoje druga slobodna parking mjesta u tom terminu kad je rezervacija

#### **UC14 - Brisanje aktivnih rezervacija**

- **Glavni sudionik:** Vozač
- **Cilj:** Obrisati rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Vozač je prijavljen i napravio je barem jednu rezervaciju
- **Opis osnovnog tijeka:**
  1. Vozač odabire opciju „Moje rezervacije“
  2. Sustav prikaže listu rezervacija vozača
  3. Vozač odabire opciju „Obriši rezervaciju“
  4. Rezervacija se briše iz baze podataka
  5. Vozaču se vraćaju novci ako ima neiskorištenih rezervacija
- **Opis mogućih odstupanja:**
  - 5.a Greška prilikom refundiranja novca
    1. Sustav obavještava vozača da je došlo do greške prilikom refundiranja novca na njegov račun, te se odustaje od brisanja rezervacije.

#### **UC15 - Prijava parkirališne površine**

- **Glavni sudionik:** Tvrtka
- **Cilj:** Prijaviti parking objekt
- **Sudionici:** Baza podataka
- **Preduvjet:** Tvrtka je prijavljena
- **Opis osnovnog tijeka:**



1. Tvrtka odabire opciju „Prijavi parking“
  2. Unos podataka o parkirališnoj površini
  3. Tvrtka unese sve podatke i odabire „Spremi“
  4. Ažurira se baza podataka
- **Opis mogućih odstupanja:**
    - 2.a Unesen neispravan/već prijavljen parking
      1. Sustav obavijesti korisnika da nije moguće prijaviti parking te ga vraća na stranicu za prijavu parkirališne površine
      2. Korisnik mijenja podatka ili odustaje

#### **UC16 - Pregled prijavljenih parkirališnih površina**

- **Glavni sudionik:** Tvrtka
- **Cilj:** Pregled prijavljenih parkirališnih površina
- **Sudionici:** Baza podataka
- **Preduvjet:** Tvrtka je prijavljena i postoji prijavljena parkirališna površina
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „Popis parkirališnih površina“
  2. Otvara se lista prijavljenih parkirališnih površina

#### **UC17 - Uređivanje parkirališnih površina**

- **Glavni sudionik:** Tvrtka
- **Cilj:** Urediti parkirališnu površinu
- **Sudionici:** Baza podataka
- **Preduvjet:** - Tvrtka je prijavljena i ima barem jednu prijavljenu parkirališnu površinu
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „Popis parkirališnih površina“
  2. Sustav prikaže listu prijavljenih parkirališnih površina
  3. Korisnik odabire željenu parkirališnu površinu
  4. Prikažu se podaci parkirališne površine
  5. Korisnik izmjeni podatke i odabire „Spremi“
  6. Ažurira se baza podataka
- **Opis mogućih odstupanja:**
  - 5.a Korisnik promijeni podatke parkirališne površine, ali ne odabere opciju „Spremi promjenu“

1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora
- 5.b Korisnik promijeni podatke parkirališne površine, ali je nešto neispravno uneseno
  1. Sustav obavještava korisnika da je nemoguće izmjeniti podatke prilikom odabira „Spremi promjene“

#### **UC18 - Brisanje parkirališnih površina**

- **Glavni sudionik:** Tvrtka
- **Cilj:** Obrisati parkirališnu površinu
- **Sudionici:** Baza podataka
- **Preduvjet:** - Tvrtka je prijavljena i ima barem jedan prijavljeni parkirališnu površinu
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju „Popis parkirališnih površina“
  2. Sustav prikaže listu prijavljenih parkirališnih površina
  3. Korisnik briše željenu parkirališnu površinu
  4. Ažurira se baza podataka

#### **UC19 - Pregled korisnika**

- **Glavni sudionik:** Administrator
- **Cilj:** Vidjeti popis odabranih registriranih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** - Administrator prijavljen
- **Opis osnovnog tijeka:**
  1. Opis osnovnog tijeka:
  2. Administrator odabire opciju „Pregled korisnika“
  3. Administrator odabire filtere prema kojima želi filtrirati korisnike ili ih upisuje u tražilicu
  4. Pojavi se lista s odabranim korisnicima
  5. Administrator odabire željenog korisnika te upravlja njegovim podacima
  6. Ažurira se baza podataka

#### **UC20 - Uređivanje odabranih korisnika**

- **Glavni sudionik:** Administrator
- **Cilj:** Urediti odabrane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator prijavljen i postoji barem jedan korisnik
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju „Pregled korisnika“
  2. Administrator odabire filtere prema kojima želi filtrirati korisnike ili ih upisuje u tražilicu
  3. Sustav prikazuje listu odabranih korisnika
  4. Administrator odabire željenog korisnika te upravlja njegovim podacima
  5. Ažurira se baza podataka

#### UC21 - Brisanje odabranih korisnika

- **Glavni sudionik:** Admin
- **Cilj:** sudionik: Obrisati odabrane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Admin prijavljen i postoji barem jedan korisnik
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju „Pregled korisnika“
  2. Administrator odabire filtere prema kojima želi filtrirati korisnike ili ih upisuje u tražilicu
  3. Sustav prikazuje listu odabranih korisnika
  4. Administrator odabire željenog korisnika te ga briše
  5. Ažurira se baza podataka

#### UC22 - Pregled tvrtki

- **Glavni sudionik:** Admin
- **Cilj:** Vidjeti popis odabranih tvrtki
- **Sudionici:** Baza podataka
- **Preduvjet:** Admin prijavljen
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju „Pregled korisnika“
  2. Administrator Odabire filtere prema kojima želi filtrirati tvrtke ili ih upisuje u tražilicu
  3. Pojavi se lista s odabranim tvrtkama

**UC23 - Uređivanje odabranih tvrtki**

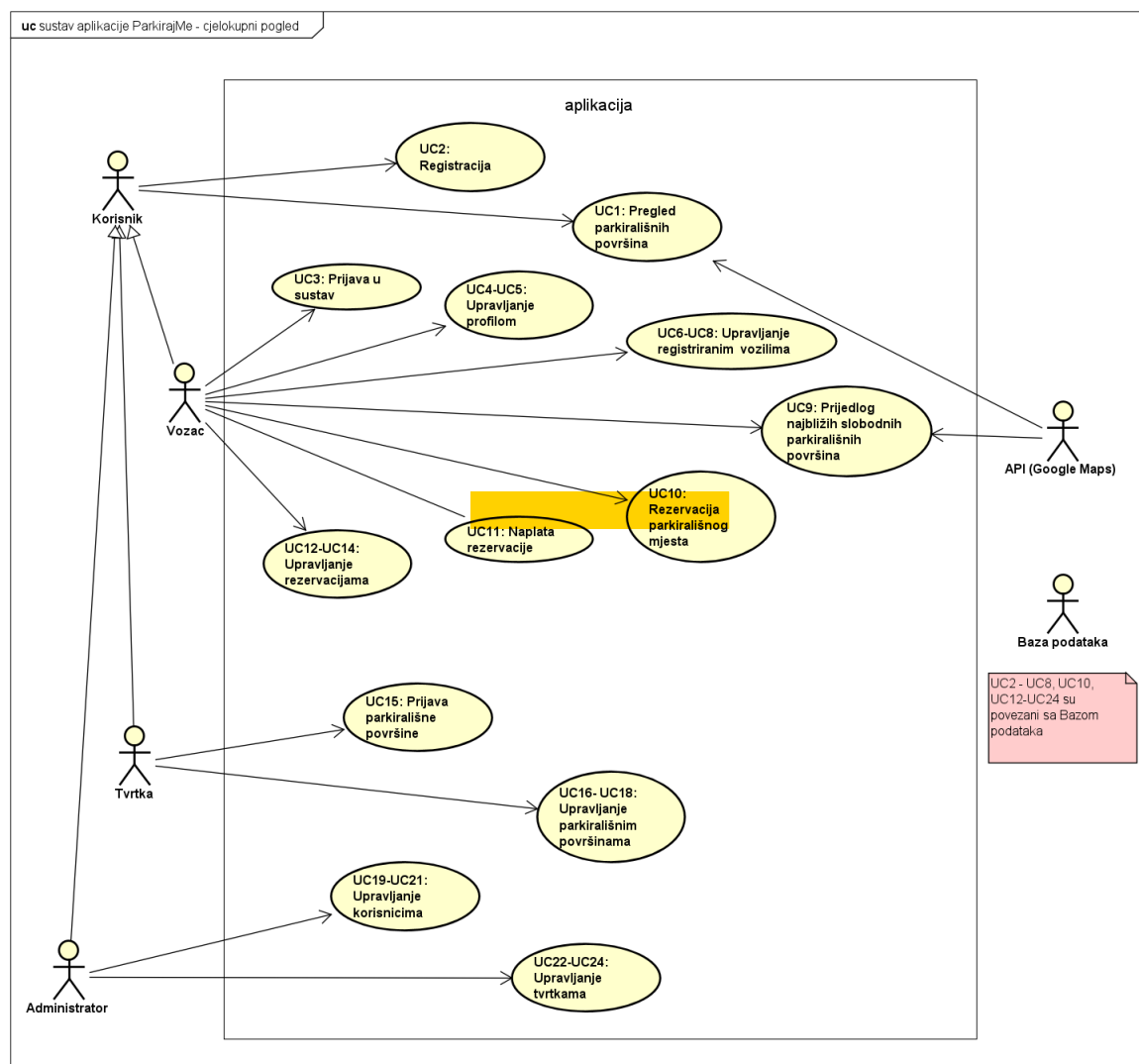
- **Glavni sudionik:** Administrator
- **Cilj:** Urediti odabrane tvrtke
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator prijavljen i postoji barem jedna tvrtka
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju „Pregled tvrtki“
  2. Odabire filtere prema kojima želi filtrirati tvrtke ili ih upisuje u tražilicu
  3. Sustav prikazuje listu odabranih tvrtki
  4. Administrator odabire željenu tvrtku te upravlja njenim podacima
  5. Ažurira se baza podataka

**UC24 - Brisanje odabranih tvrtki**

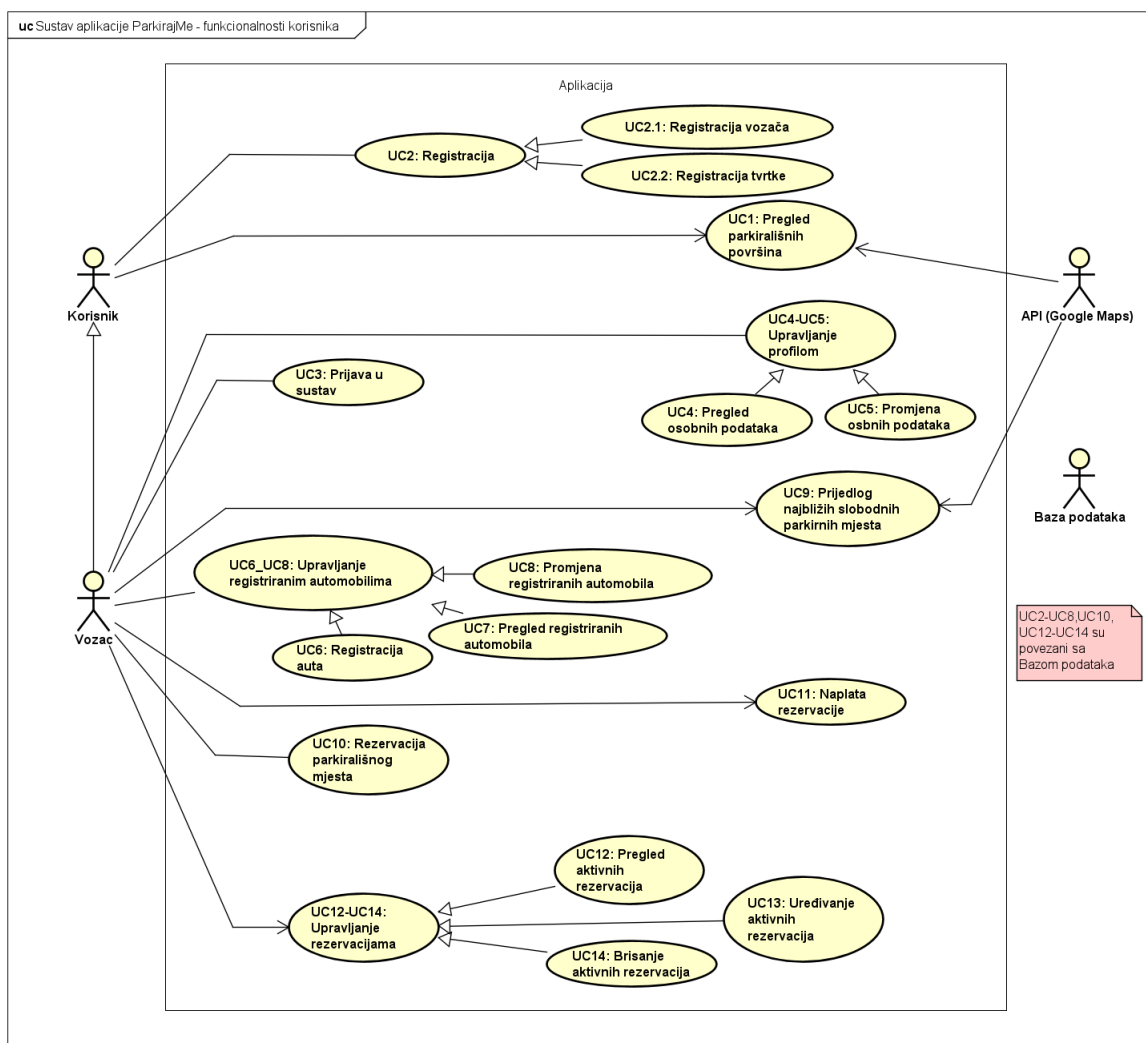
- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati odabrane tvrtke
- **Sudionici:** Baza podataka
- **Preduvjet:** - Administrator je prijavljen i postoji barem jedan korisnik
- **Opis osnovnog tijeka:**
  1. Administrator odabire opciju „Pregled tvrtki“
  2. Administrator odabire filtere prema kojima želi filtrirati tvrtke ili ih upisuje u tražilicu
  3. Sustav prikazuje listu odabranih tvrtki
  4. Administrator odabire željenu tvrtku te ju briše
  5. Ažurira se baza podataka

## Dijagrami obrazaca uporabe

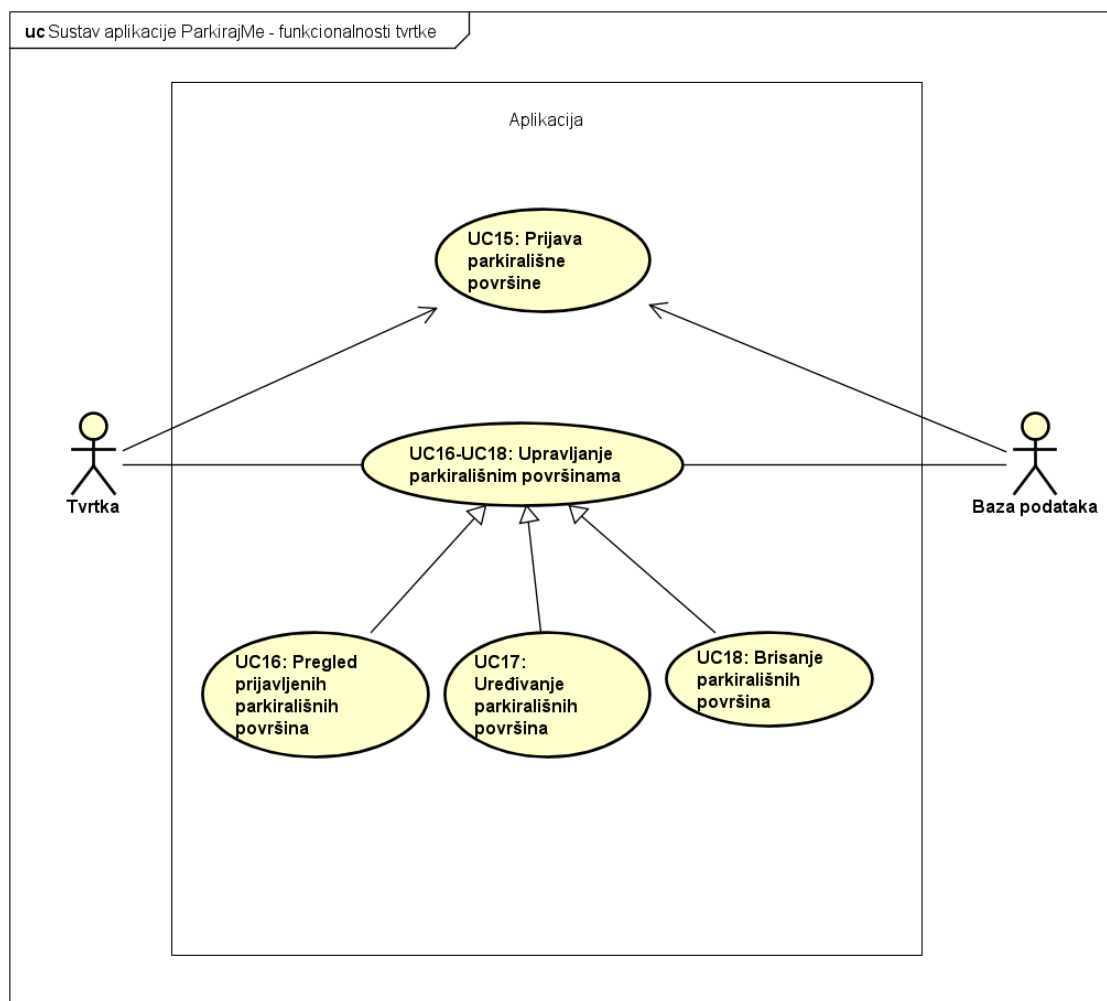
Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.



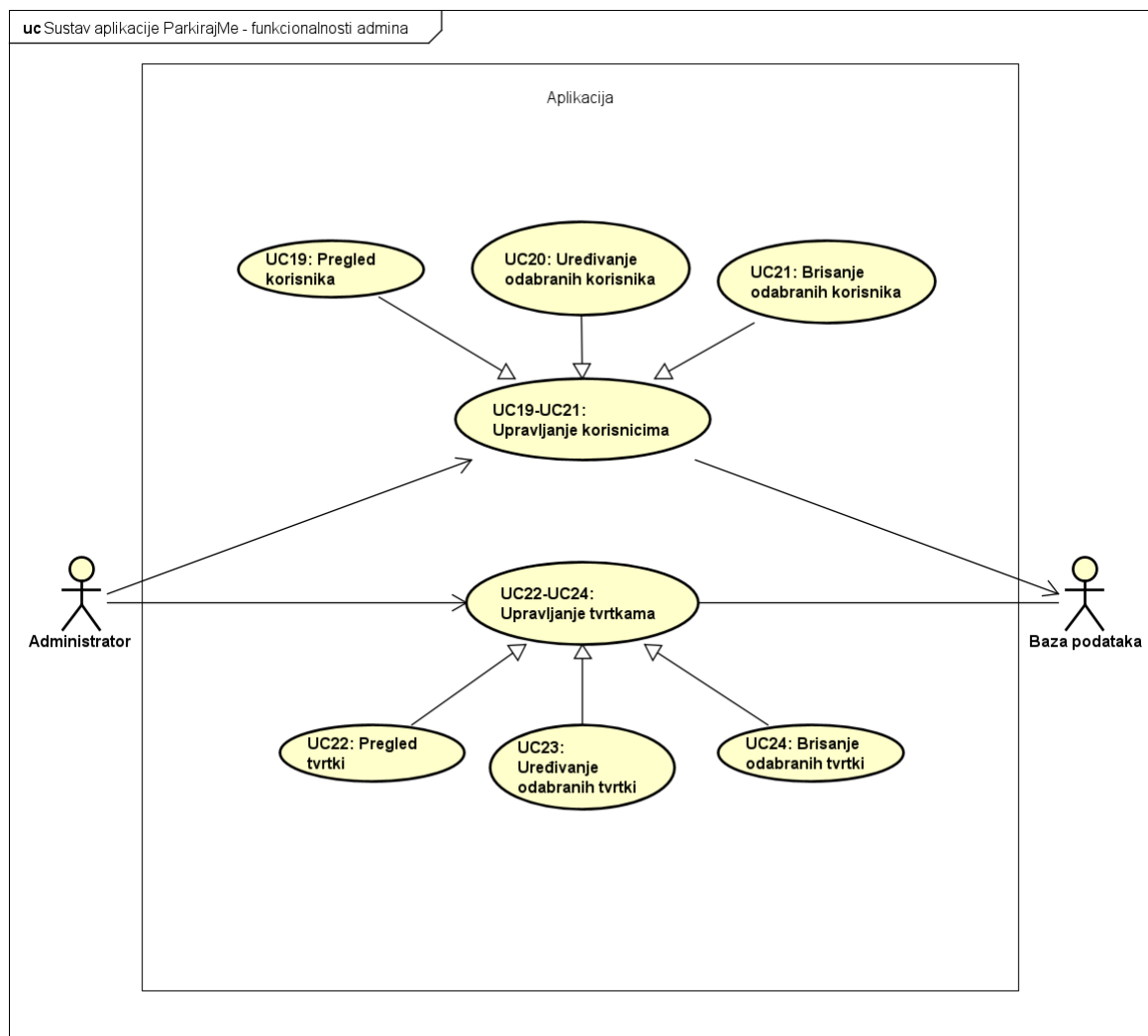
Slika 3.1: Cjelokupni pogled



Slika 3.2: Funkcionalnosti korisnika



Slika 3.3: Funkcionalnosti tvrtke



Slika 3.4: Funkcionalnosti admina

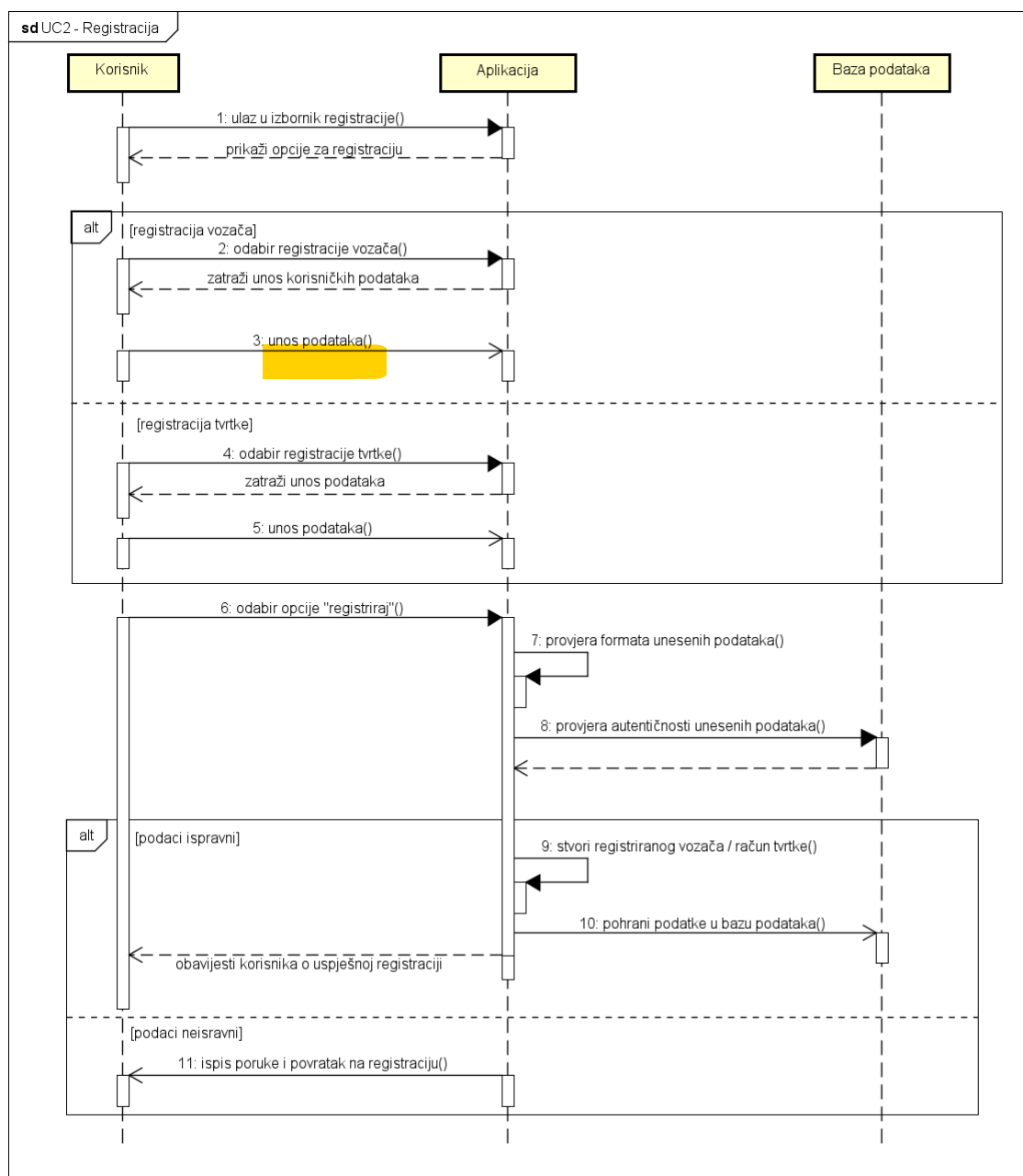


### 3.1.2 Sekvencijski dijagrami

*Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava (max. 4 dijagrama). Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.*

#### **Obrazac uporabe UC2 - Registracija**

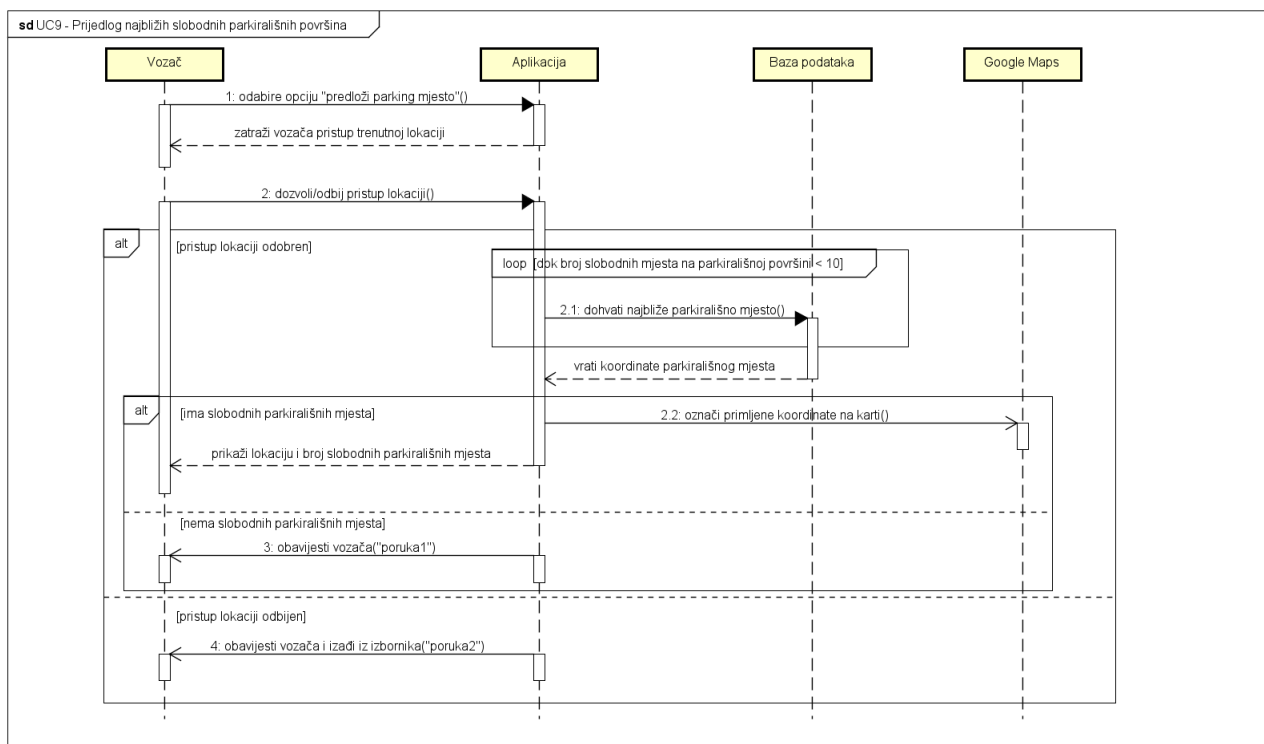
Korisnik ulazi u izbornik registracije i bira jednu od dvije opcije: "Registracija vozača" ili "Registracija tvrtke". Nakon odabira željene registracije aplikacija ga traži unos potrebnih podataka. Aplikacija provjerava jesu li sva polja popunjena i je li format unesenih podataka ispravan (broj znamenki OIB-a, format registracijske oznake i adrese e-pošte). Nakon toga aplikacija pristupa bazi podataka i provjerava jedinstvenost unesenih podataka (korisnik ne može napraviti dvije registracije sa istim automobilom). Ako su podaci ispravni, ovisno o vrsti registracije, stvara se novi vozač ili korisnički račun tvrtke te se podaci pohranjuju u bazu podataka. Ako su uneseni podaci neispravni, korisnik prima obavijest i aplikacija ga vraća na izbornik registracije.



Slika 3.5: Sekvencijski dijagram - Registracija

**Obrazac uporabe UC9 - Prijedlog najbližih slobodnih parkirališnih površina**

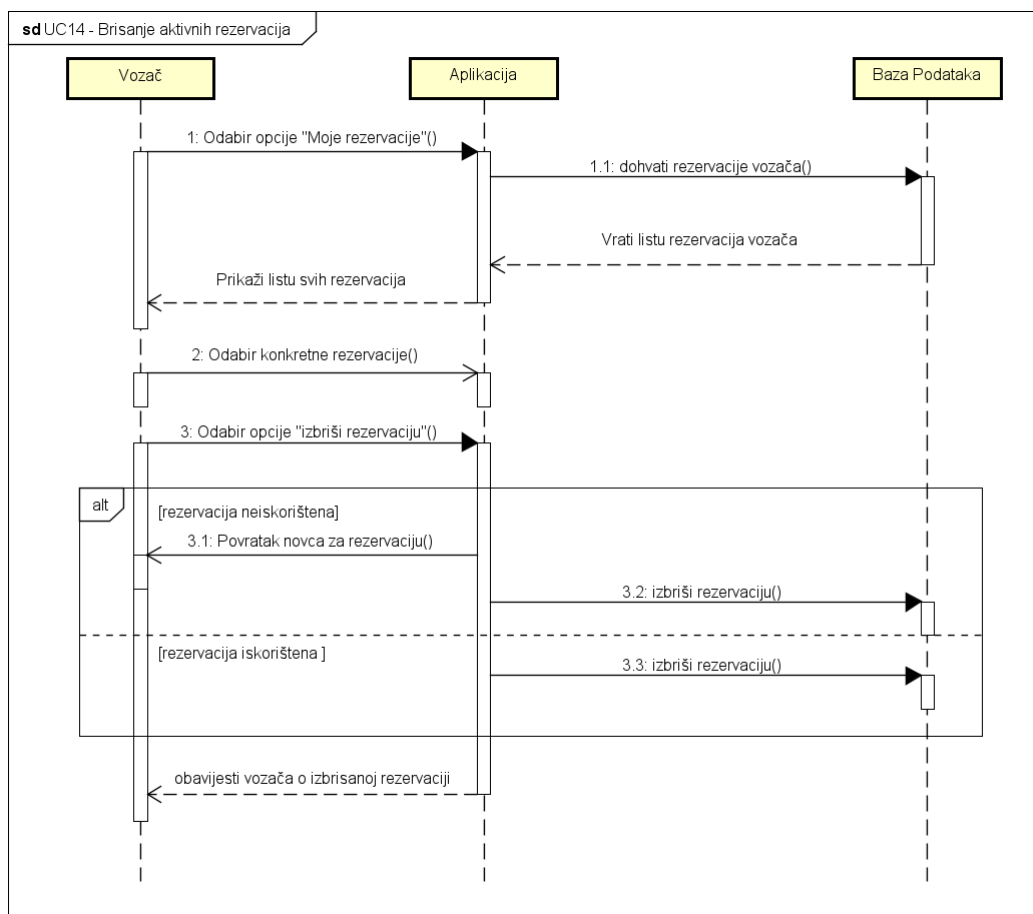
Aplikacija vozaču nudi opciju prijedloga najbližih slobodnih parkirališnih površina. Nakon što vozač odabere opciju "predloži parkirališno mjesto" poslužitelj od njega traži pristup trenutnoj lokaciji. Ako vozač dozvoli pristup, poslužitelj pronalazi 3 najbliže parkirališne površine sa minimalno 10 slobodnih parkirališnih mjesta. Prag od 10 slobodnih mjesta postavljen je kako bi se osiguralo da se od trenutka vozačevog zaprimanja lokacije do njegovog dolaska na parkirališnu površinu ne bi popunila sva parkirališna mjesta. Poslužitelj najbliže parkirališne površine pronalazi tako što uspoređuje vozačevu trenutnu lokaciju sa koordinatama parkirališnih površina pohranjenim u bazi podataka i bira 3 najmanje udaljenosti (?? ne znam još kako će to bit implementirano). Poslužitelj prikazuje dohvaćene lokacije i broj slobodnih parkirališnih mjesta. Aplikacija vozaču šalje poruku ako ne uspije pronaći ni jedno slobodno parkirališno mjesto. Ako vozač poslužitelju ne dozvoli pristup trenutnoj lokaciji, aplikacija ga obavještava da ne može pružiti uslugu i vraća vozača na glavni izbornik.



Slika 3.6: Sekvencijski dijagram - Prijedlog najbližih slobodnih parkirališnih površina

### Obrazac uporabe UC14 - Brisanje aktivnih rezervacija

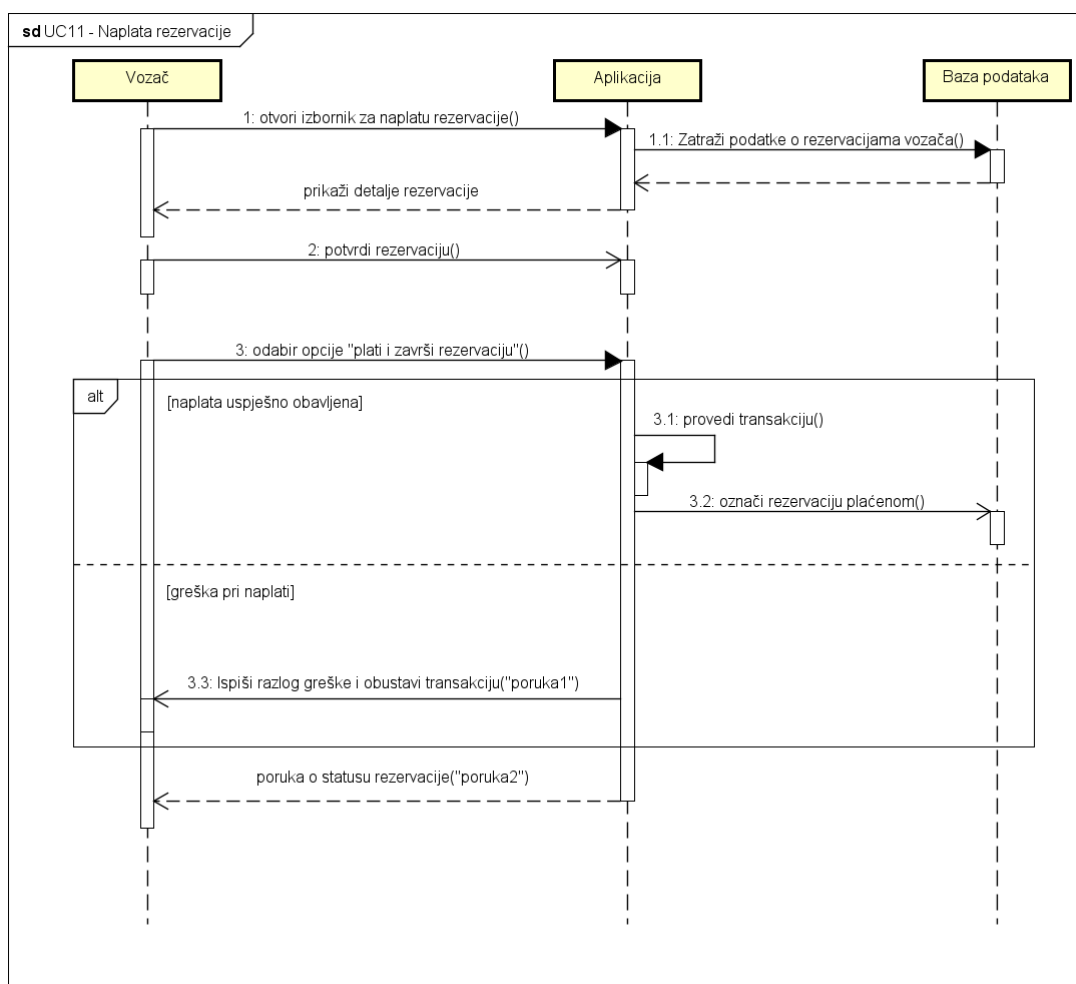
Vozač odabire opciju "moje rezervacije" nakon čega poslužitelj pristupa bazi podataka i dohvaća listu svih vozačevih rezervacija. Vozač odabire jednu rezervaciju koju želi obrisati i bira "izbriši rezervaciju". Ako rezervacija još nije iskorištena, aplikacija uplaćuje cijenu rezervacije na vozačev račun i briše ju iz baze podataka. Ako je rezervacija već iskorištena, aplikacija ju uklanja iz baze podataka. Nakon brisanja rezervacije, vozač prima obavijest.



Slika 3.7: Sekvencijski dijagram - brisanje aktivnih rezervacija

### Obrazac uporabe UC11 - Naplata rezervacije

Vozaču je omogućena naplata rezervacije putem aplikacije. Ulaskom u izbornik za naplatu rezervacije, aplikacija pristupa bazi podataka i dohvaća podatke o svim rezervacijama vozača. Vozač može pregledati detalje rezervacije i potvrditi ju odabirom opcije "plati i završi rezervaciju". Tek kada se uspostavi sigurna veza sa poslužiteljem za transakciju, ona se provodi. Nakon uspješno obavljene transakcije, rezervacija se u bazi podataka označava plaćenom i aplikacija obaviještava vozača o statusu rezervacije. Ako prilikom pokušaja naplate rezervacije dođe do greške, transakcija se zaustavlja i vozač prima odgovarajuću poruku. Prilikom naplate rezervacije, vozači plaćaju direktno aplikaciji, a nakon toga aplikacija novčana sredstva raspodjeljuje tvrtkama koje su vlasnici pojedinih parkirališnih površina.



Slika 3.8: Sekvencijski dijagram - Naplata rezervacije

## 3.2 Ostali zahtjevi

Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju **kako se sustav treba ponašati** i koja **ograničenja** treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.

sto ako je više ljudi odjednom odabere isto mjesto

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Sustav mora podržavati hrvatsku abecedu, ali i znakove engleske abecede (Q,W,X,Y) koji se mogu pojavljivati u korisničkim imenima, e-mail adresama i nazivima objekata na karti grada Zagreba
- Komunikacija sa bazom podataka (slanje upita i primanje odgovora od baze) ne smije trajati dulje od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija (koristeći ??? jezike)
- Korištenje sustava na način na koji nije zamišljeno ne smije narušiti funkcionalnost i rad
- Sustav treba biti intuitivan za korištenje
- Nadogradnja sustava provodi se na temelju zadnje verzije bez mijenjanja ključnih dijelova sustava
- Sustav kao valutu treba podržati HRK
- Cijene rezervacija prikazuju točan iznos do drugog decimalnog mjesta
- Baza podataka treba biti otporna na napade i greške
- Pristup sustavu treba biti omogućen iz javne mreže pomoću HTTPS-a
- Sustav korisniku mora omogućiti registraciju neograničenog broja automobila
- Lokacija parkirališnih površina ima odstupanje od maksimalno 30 metara
- Sustav je dostupan 24 sata dnevno, 365 dana u godini (osim u trenucima obnavljanja ili nadogradnje sustava)

## 4. Arhitektura i dizajn sustava

### *dio 1. revizije*

*Arhitektura je podjeljena na tri sustava:*

- *Web Aplikacija*
- *Web Poslužitelj*
- *Baze podataka*

*Web aplikacija* je dinamička, i zato je potrebno uvesti *web poslužitelj* odnosno server. Pomoću web poslužitelja korisnik može komunicirati sa web aplikacijom. Komunikacija se odvija preko HTTP protokola.

Kada korisnik koristi web aplikaciju, on šalje određene zahtjeve koje web aplikacija obrađuje. Ovisno o zahtjevu, web aplikacija će dohvatiti podatke iz baze podataka. Na svaki zahtjev, web aplikacija je dužna poslati odgovor korisniku ili javiti pogrešku.

*Baza podataka* služi za pohranu, izmjenu i dohvat podataka koji će se obrađivati u aplikaciji. Za njenu implementaciju koristiti ćemo PostgreSQL.

Za izradu web aplikacije odlučili smo koristiti *Node.js*, te ćemo većinu funkcionalnosti aplikacije ostvariti uz pomoć *Express frameworka*. Za spajanje aplikacije s bazom podataka koristiti ćemo *Connect-pg-Simple*.

Aplikaciju ćemo razvijati u razvojnom okruženju Microsoft Visual Studio. Arhitektura sustava temeljit će se na *Model-View-Controller (MVC)* arhitekturi.

### 4.1 Baza podataka

Za naš sustav koristiti ćemo relacijsku bazu podataka, ostvarenu u aplikaciji PostgreSQL. Baza je prilagođena brzom pohranjivanju i dohvaćanju podataka, te su entiteti napravljeni tako da odgovaraju modelima u dijagramu razreda.

Baza podataka sastoji se od sljedećih entiteta:

- *Korisnik*

- *Tvrtka*
- *Parking*
- *Vozilo*
- *Rezervacija*
- *Sjednica*

#### 4.1.1 Opis tablica

Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ

**Korisnik** je entitet koji čuva informacije o korisniku aplikacije *ParkirajMe*. Sadrži attribute: Korisničko ime i OIB korisnika, broj kreditne kartice, e-mail korisnika, te njegovo ime i prezime. Ovaj entitet u vezi je *One-to-Many* s entitetom vozilo preko korisničkog imena, zato jer svaki korisnik može posjedovati više vozila, a jedno vozilo također može imati više korisnika koji ga parkiraju.

korisnik		
Korisničko ime	VARCHAR	Jedinstveno korisničko ime korisnika
Lozinka	VARCHAR	Hash jedinstvene lozinke korisnika
OIB	CHAR(11)	Osobni identifikacijski broj korisnika
Broj kred. kartice	VARCHAR	Broj kreditne kartice korisnika
Email	VARCHAR	adresa elektroničke pošte korisnika
Ime	VARCHAR	Ime korisnika
Prezime	VARCHAR	Prezime korisnika
Razina ovlasti	INT	Razina ovlasti korisnika

**Tvrtka** je entitet koji predstavlja korisnički račun tvrtke koji je u njeno ime otvorio ovlašteni zaposlenik tvrtke. Sadrži attribute: korisničko ime, lozinka, OIB tvrtke, Email, ime te adresu sjedišta. Povezan je vezom *One-To-Many* s entitetom parking, zato jer je jedna tvrtka može imati više parkinga.



tvrtka		
Korisničko ime	VARCHAR	Jedinstveno korisničko ime ovlaštenog zaposlenika tvrtke
Lozinka	VARCHAR	Hash jedinstvene lozinke ovlaštenog zaposlenika tvrtke
OIB tvrtke	CHAR(11)	Osobni identifikacijski broj tvrtke
Email	VARCHAR	adresa elektroničke pošte tvrtke
Ime	VARCHAR	Ime tvrtke
adresaTvrtka	VARCHAR	Adresa sjedišta tvrtke

**Vozilo** je entitet koji pohranjuje koja sve vozila je u aplikaciju unio neki korisnik, te su to jedina dva atributa ovog entiteta. U vezi je *Many-to-One* s entitetom korisnik preko atributa korisničko ime, te u vezi *One-to-Many* s entitetom rezervacija preko atributa registracija.

vozilo		
Korisničko ime	VARCHAR	Jedinstveno korisničko ime vlasnika vozila
Registracija	VARCHAR	Registracija automobila

**Rezervacija** je entitet koji sadržava podatke o rezervaciji jednog parkirnog mjesta. Njezini atributi su: ID rezervacije, ID grupe rezervacija kojoj ova rezervacija pripada, Korisničko ime onog tko je rezervirao, registracija vozila, ID parkinga, datum i vrijeme početka i kraja rezervacije. Povezan je vezom *Many-to-One* sa entitetom Vozilo preko atributa Korisničko ime i rezervacija (Ključ entiteta vozilo), te s entitetom Parking vezom *Many-to-One* preko atributa Parking ID.

rezervacija		
IDRezervacija	SERIAL	Jedinstveni ID rezervacije
ID grupe	INT	Jedinstvena oznaka grupe rezervacija
Korisničko ime	VARCHAR	Jedinstveno korisničko ime vlasnika vozila
Registracija vozila	VARCHAR	Registracija automobila koji će zauzeti mjesto
IDParking	INT	Jedinstvena oznaka parkinga
Pocetak rezervacije	TIMESTAMP	Početak rezervacije
Kraj rezervacije	TIMESTAMP	Završetak rezervacije

**Parking** je entitet koji sadržava podatke o jednom parkingu. Njegovi atributi su: ID parkinga, OIB tvrtke koja ga je objavila, ukupan kapacitet, broj slobodnih mjesta, cijena parkirnog mjesta, adresa parkinga, te geografska širina i dužina njegove lokacije. Povezan je sa entitetom tvrtka vezom *Many-to-One* pošto jedna tvrtka može imati više parkinga, te s entitetom rezervacije vezom *One-to-Many* zato jer na jednom parkingu može biti više rezervacija. Također je povezan s entitetom Lokacija vezom *One-to-One* preko atributa ID lokacija.

parking		
ID parkinga	SERIAL	Jedinstvena oznaka parkinga
Korisničko ime tvrtke	VARCHAR	korisničko ime tvrtke koja je vlasnik parkinga
Kapacitet	INT	Ukupan kapacitet parkinga
Broj slob. mjesta	INT	Broj slobodnih mjesta na parkingu
Cijena po satu	NUMERIC	Cijena rezervacije parkinga po satu
ID lokacija	INT	Jedinstvena oznaka lokacije parkinga

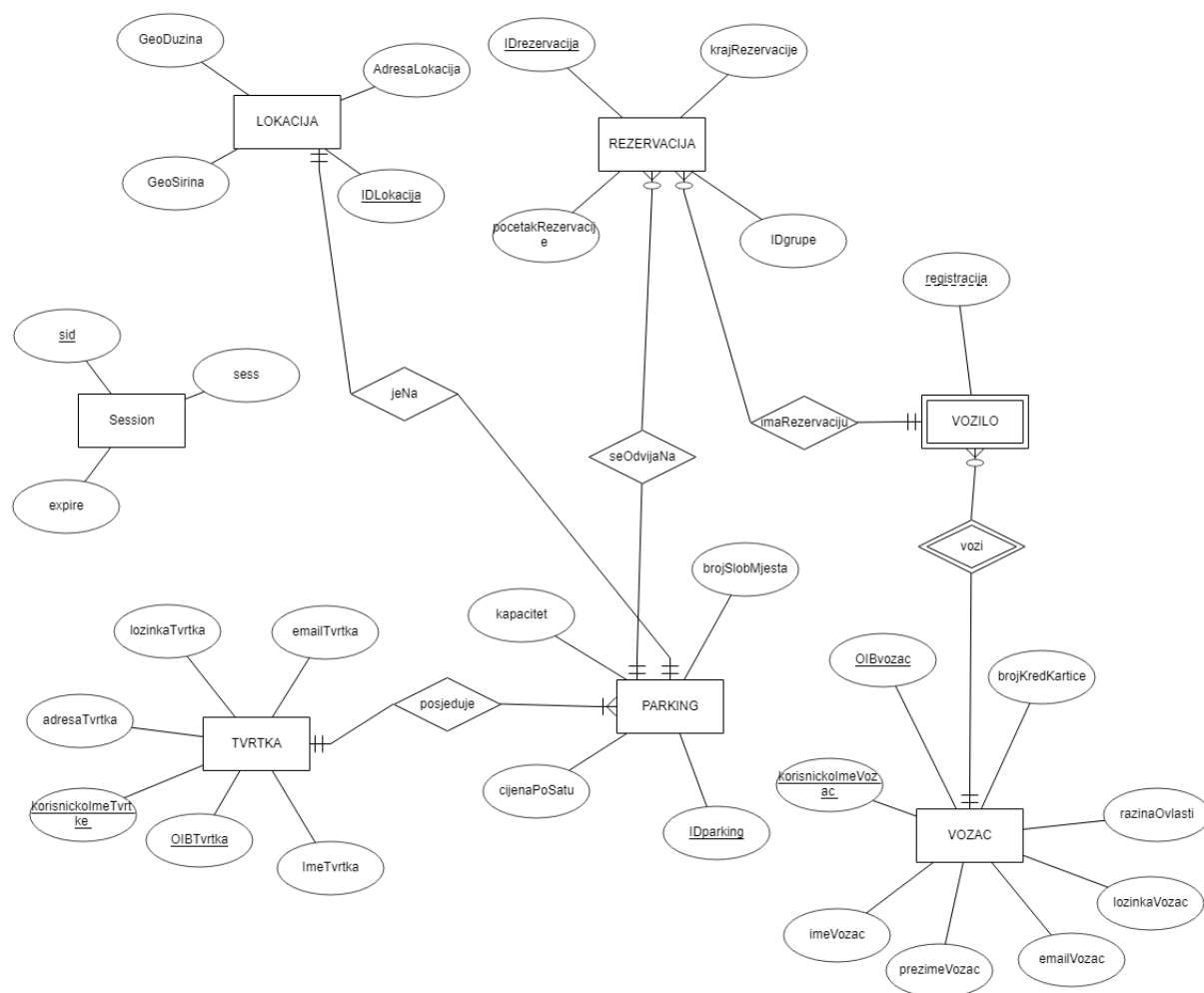
**Lokacija** je entitet koji sadržava podatke o jednoj lokaciji. Njegovi atributi su ID lokacije, njena adresa, te geografska širina i dužina. Povezan je vezom *One-to-One* s entitetom Parking zato jer se na jednoj lokaciji može nalaziti samo jedan parking.

lokacija		
ID lokacije	SERIAL	Jedinstvena oznaka lokacije
Adresa lokacije	VARCHAR	Adresa lokacije
Geo. širina	NUMERIC	Geograska širina lokacije
Geo. dužina	NUMERIC	Geograska dužina lokacije

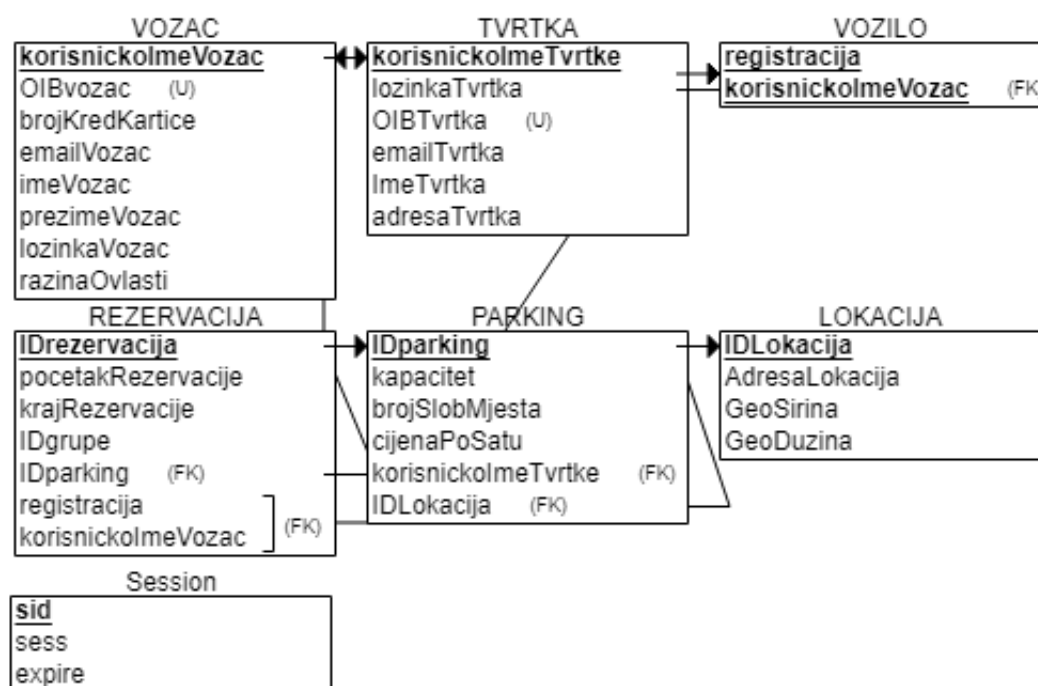
**Sjednica** je entitet koji sadrži informacije o sjednicama koje se stvaraju na stranici ParkirajMe. Sadrži attribute ID sjednice, informacije o sjednici ti vrijeme trajanja sjednice.

sjednica		
ID sjednice	VARCAHR	Jedinstvena oznaka sjednice
Info	JSON	Informacije o sjednici
Trajanje	TIMESTAMP	Trajanje sjednice

### 4.1.2 Dijagram baze podataka



Slika 4.1: ER model



Slika 4.2: relacijska shema

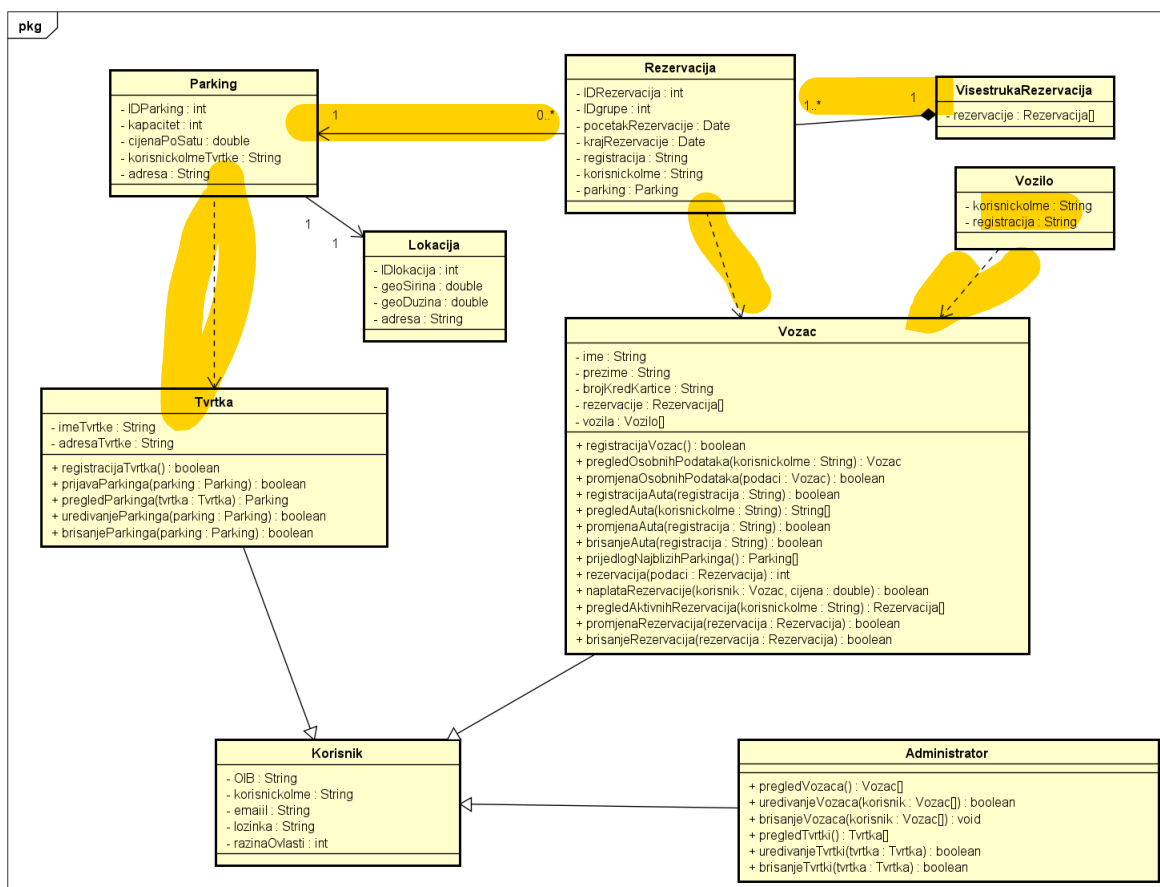
## 4.2 Dijagram razreda

Dijagrami razreda MVC arhitekture sustava prikazani su na slikama 4.4, 4.5 i 4.6. Razredi na slici 4.4 predstavljaju klase modela, koji su uglavnom direktno povezani s entitetima u bazi podataka.

Razredi na slici 4.5 su pomoćne **klase** koje pomažu upravljanju podacima, te najčešće služe za komunikaciju s bazom podataka.

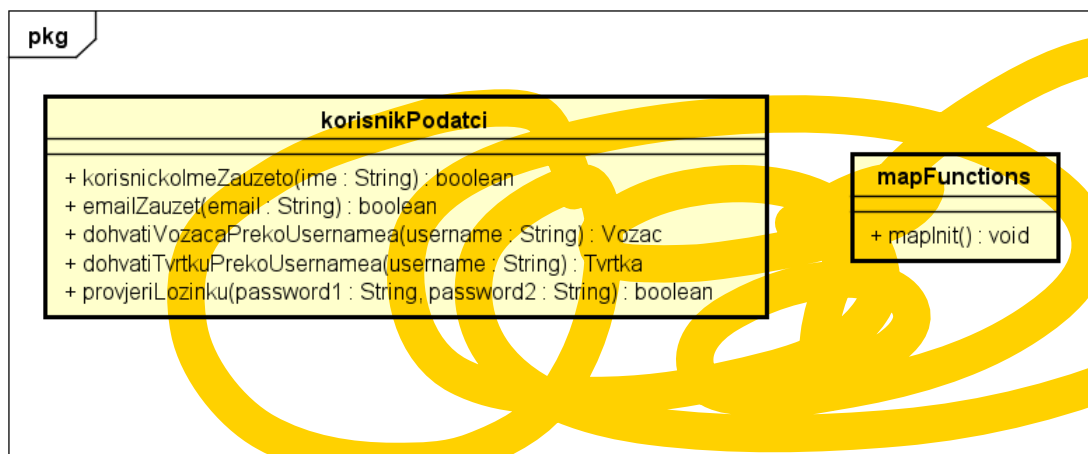
Na slici 4.6 se vide razredi koji implementiraju funkcionalnosti routera te oni prikazuju pug file-ove koji prikazuju stranice.

Razred **Korisnik** predstavlja bilo kojeg korisnika aplikacije, bio to vozač, tvrtka, administrator ili ne-registrirani korisnik. **Vozač** predstavlja korisnika aplikacije koji želi rezervirati parkirališno mjesto, a **Tvrtka** korisnika koji želi iznajmljivati svoju parkirališnu površinu. **Administrator** je korisnik s najvećim ovlastima na aplikaciji. Jedina klasa koja se razlikuje od odgovarajućeg entiteta u bazi podataka je **Rezervacija** zato što također postoji klasa **Višestruka rezervacija** koja označava skup rezervacija koji su dio iste ponavljajuće ili trajne rezervacije.



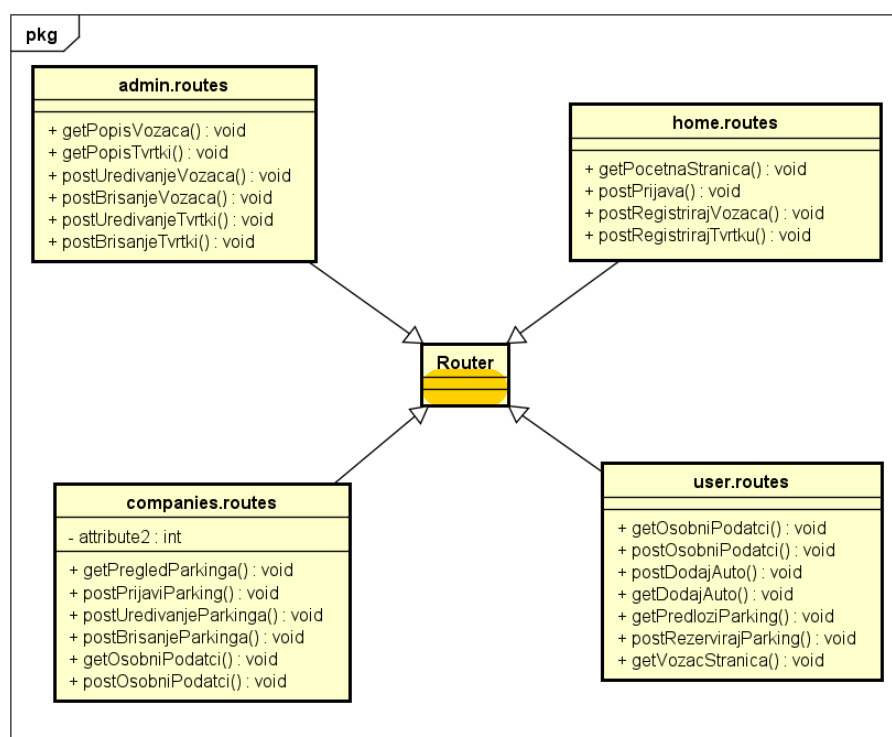
Slika 4.3: Dijagram razreda za modele

Pomoćne klase su klase koje u sebi sadrže pomoćne funkcije koje potpomažu ostvarivati funkcionalnosti drugih klasa, te najčešće služe za direktno slanje upita bazi podataka.



Slika 4.4: Dijagram razreda za pomoćne razrede

Routeri su klase koji služe za povezivanje sa frontendom, odnosno ostvaruju prikazivanje pug-ova. Postoje routeri za **home-page**, stranicu za vozače (radno ime **user.routes**), stranicu za tvrtke te administratora.



Slika 4.5: Dijagram razreda za routere

*dio 2. revizije*

*Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije*



## 4.3 Dijagram stanja

### *dio 2. revizije*

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

## 4.4 Dijagram aktivnosti

### *dio 2. revizije*

*Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.*

## 4.5 Dijagram komponenti

### *dio 2. revizije*

*Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.*

## 5. Implementacija i korisničko sučelje

### 5.1 Korištene tehnologije i alati

#### *dio 2. revizije*

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

1. Visual Studio Code, <https://code.visualstudio.com/download>

## 5.2 Ispitivanje programskog rješenja

### dio 2. revizije

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.*

### 5.2.1 Ispitivanje komponenti

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

### 5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium<sup>1</sup>. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

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

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

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

---

<sup>1</sup><https://www.seleniumhq.org/>

## 5.3 Dijagram razmještaja

### *dio 2. revizije*

*Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.*

## 5.4 Upute za puštanje u pogon

### *dio 2. revizije*

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

*Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.*

## 6. Zaključak i budući rad

### *dio 2. revizije*

*U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.*

*Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.*



# Popis literature

## *Kontinuirano osvježavanje*

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

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

# Indeks slika i dijagrama

2.1	Odabir poslovnica tvrtke Kaufland, <a href="https://www.kaufland.hr/usluge/poslovnica.html">https://www.kaufland.hr/usluge/poslovnica.html</a> . . . . .	6
2.2	Odabir poslovnica tvrtke Tommy, <a href="https://tommy.hr/hr/prodajna-mjesta">https://tommy.hr/hr/prodajna-mjesta</a> . . . . .	7
3.1	Cjelokupni pogled . . . . .	20
3.2	Funkcionalnosti korisnika . . . . .	21
3.3	Funkcionalnosti tvrtke . . . . .	22
3.4	Funkcionalnosti admina . . . . .	23
3.5	Sekvencijski dijagram - Registracija . . . . .	25
3.6	Sekvencijski dijagram - Prijedlog najbližih slobodnih parkirališnih površina . . . . .	26
3.7	Sekvencijski dijagram - brisanje aktivnih rezervacija . . . . .	27
3.8	Sekvencijski dijagram - Naplata rezervacije . . . . .	28
4.1	ER model . . . . .	34
4.2	relacijska shema . . . . .	35
4.3	Dijagram razreda za modele . . . . .	37
4.4	Dijagram razreda za pomoćne razrede . . . . .	38
4.5	Dijagram razreda za routere . . . . .	38