

# Programsko inženjerstvo

Ak. god. 2023./2024.

## SpotPicker

Dokumentacija, Rev. 1

Grupa: *Šargarepoljupci*

Voditelj: *Matko Krnić*

Datum predaje: *17. studenog. 2023.*

Nastavnik: *Hrvoje Nuić*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>10</b>
3.1 Funkcionalni zahtjevi . . . . .	10
3.1.1 Obrasci uporabe . . . . .	12
3.1.2 Sekvencijski dijagrami . . . . .	22
3.2 Ostali zahtjevi . . . . .	26
<b>4 Arhitektura i dizajn sustava</b>	<b>27</b>
4.1 Baza podataka . . . . .	29
4.1.1 Vrsta i implementacija . . . . .	29
4.1.2 Glavne komponente baze podataka . . . . .	29
4.1.3 Opis tablica . . . . .	29
4.1.4 Dijagram baze podataka . . . . .	33
4.2 Dijagram razreda . . . . .	35
4.3 Dijagram stanja . . . . .	36
4.4 Dijagram aktivnosti . . . . .	37
4.5 Dijagram komponenti . . . . .	38
<b>5 Implementacija i korisničko sučelje</b>	<b>39</b>
5.1 Korištene tehnologije i alati . . . . .	39
5.2 Ispitivanje programskog rješenja . . . . .	40
5.2.1 Ispitivanje komponenti . . . . .	40
5.2.2 Ispitivanje sustava . . . . .	40
5.3 Dijagram razmještaja . . . . .	41
5.4 Upute za puštanje u pogon . . . . .	42
<b>6 Zaključak i budući rad</b>	<b>43</b>
<b>Popis literature</b>	<b>44</b>

<b>Indeks slika i dijagrama</b>	<b>45</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>46</b>

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	matkokrnic	16.10.2023.
0.2	Dodan opis projektnog zadatka.	JureRoncevic	29.10.2023.
0.3	Dodani opis obrazaca uporabe u specifikaciji programske potpore.	matkokrnic	30.10.2023.
0.4	Dodane neke osnovne funkcionalosti za backend	piilip	11.11.2023.
0.4.1	Popravljene greške za backend	piilip	11.11.2023
0.5	Dodani dijagrami obrazaca uporabe	MegaNoris	12.11.2023.
0.5.1	Ispravak pravopisnih pogrešaka	MegaNoris	12.11.2023.
0.6	Napravljen backend login	matkokrnic	13.11.2023.
0.6.1	Ostvarena registracija za backend	matkokrnic	13.11.2023.
0.7	Napravljene izmjene u dijagramima obrazaca uporabe	MegaNoris	14.11.2023.
0.8	Dodane slike za ER dijagram i relacijsku shemu	piilip	14.11.2023.
0.8.1	Napravljene tablice i opis baze podataka	piilip	14.11.2023.
0.9	Dodana mapa .idea pod mapom Izvorni-Kod sa svim potrebnim datotekama	piilip	14.11.2023.
0.10	Dodani sekvencijski dijagrami u mapu slike	romgko	15.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.11	Dodana prva verzija frontenda	domajenjic	16.11.2023.
0.11.1	Dodani komentari za frontend	domajenjic	16.11.2023.
0.12	Dodana arhitektura i dizajn sustava	JureRoncevic	17.11.2023.
0.12.1	Dodani obrasci uporabe i sekvencijski dijagrami u projektnu dokumentaciju	JureRoncevic	17.11.2023.
0.13	Dodan backend API i pomoćne klase	piilip	17.11.2023.
0.14	Dodani opisi uz sekvencijske dijagrame	JureRoncevic	17.11.2023.
0.15	Ažuriran i popravljen dnevnik promjena dokumentacije	JureRoncevic	17.11.2023.
<b>1.0</b>	Verzija samo s bitnim dijelovima za 1. ciklus	*	17.11.2023.

*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.*

## 2. Opis projektnog zadatka

Ideja našeg projekta je ostvariti aplikaciju koja će omogućiti korisniku da pregledava, rezervira i naplaćuje parkirna mjesta za automobile i bicikle. Aplikacija to ostvaruje na način da prilikom njezinog otvaranja prikazuje korisniku kartu nad kojom korisnik može birati željeno odredište, tip vozila koje koristi te trajanje parkiranja. Nakon svih odabira, na karti mu se iscrtava najbrža ruta do slobodnog parkirališnog mjesta te ako je mjesto slobodno, ono se rezervira. Parkirališta za bicikle nemaju ucrtana pojedinačna mjesta te se ne rezerviraju ni naplaćuju, već aplikacija jedino prati ukupan broj slobodnih mjesta. Za računanje najbližeg dostupnog parkirališnog mjesta aplikacija koristi OSRM, što je kratica za Open Source Routing Machine. Riječ je o modernom C++ algoritmu za usmjeravanje koji se koristi za izračunavanje najkraćeg puta u cestovnom prometu.

Aplikacija koristi API za više funkcija:

- implementacija novčanika u koji korisnik uplaćuje sredstva
- osvježavanje parkirališnog mjesta radi provjere zauzetosti
- dohvaćanje početnih informacija o stanju parkirališnih mjesta pri čemu se koristi overpass API

Aplikacija se može pokrenuti bez registracije, no za pristup svim alatima aplikacije, potrebna je registracija. Korisnici koji nisu registrirani mogu na karti vidjeti sva dostupna parkirališta i parkirališna mjesta, ali nemaju informacije u njihovom stanju u stvarnom vremenu. Neregistrirani korisnik koji se želi registrirati šalje zahtjev za registraciju pri čemu ima opciju birati između dvije uloge (voditelj parkirališta i klijent). Za registraciju su potrebni:

- korisničko ime
- ime
- prezime
- slika osobne
- IBAN račun
- email adresa

Za registraciju kao klijent dovoljna je samo potvrda preko email adrese. Ako je riječ o voditelju parkirališta tada je potrebna i potvrda administratora. Registrirani korisnici također imaju mogućnost pregledavanja te izmjene vlastitih podataka, uključujući i opciju da obrišu svoj račun.

Klijent otvaranjem aplikacije dobiva pregled nad kartom te odabire lokaciju odredišta, tip vlastitog vozila, i trajanje parkiranja. Klijent može rezervirati parkirališno mjesto na dva načina. Može prvo označiti parkirališna mjesta koja mu odgovaraju te mu tada aplikacija nudi moguće termine u kojima će ta mjesta biti slobodna. Druga opcija je da prvo definira odgovarajući termin nakon čega će mu aplikacija prikazati sva parkirališna mjesta koja su slobodna u tom terminu. Aplikacija razlikuje parkirališna mjesta čije su razine različite te korisnik ima mogućnost odabira. Kada je klijent zadovoljan s parkirališnim mjestom, rezervira mjesto na proizvoljno vrijeme ako je slobodno te pri tome ima opciju da napravi rezervaciju ponavljajućom. Klijent jedino može rezervirati mjesto u budućnosti, odnosno, mjesto ne može biti rezervirano u istom datumu u kojem je korisnik otvorio aplikaciju. Plaćanje parkiranja korisnik obavlja tijekom rezervacije ili po dolasku na parkirališno mjesto. Plaćanje se obavlja preko novčanika koji je implementiran unutar aplikacije u koji korisnik može proizvoljno ubaciti novac.

Voditelj parkirališta ima veće ovlasti od klijenta. On ima dozvolu unijeti podatke o vlastitom parkiralištu koje uključuju:

- naziv
- opis
- fotografija
- cjenik i sl.

Voditelj nad svojim parkiralištem može individualno ucrtati svako parkirališno mjesto prema vlastitim željama. Nakon što ucrtava mjesta, on odlučuje je li mjesto dostupno za rezervaciju. Dodatno, voditelj postavlja i senzor koji služi za dohvaćanje informacije o zauzetosti mjesta. Za svoje parkiralište voditelj određuje cijenu rezervacije ovisnu o vremenu. Voditelj također ima i pristup statističkim podacima o svom parkiralištu. Moguće je promatrati zauzetost cijelog parkirališta te pojedinačnih parkirališnih mjesta kroz vrijeme uz pomoć grafa.

Administrator aplikacije ima najveće ovlasti i sposoban je vidjeti popis svih registriranih korisnika i njihovih osobnih podataka te im po potrebi može mijenjati osobne podatke.

Motivacija za razvoj ovog projekta dolazi od mnogih prednosti koje aplikacija za parkiranje može donijeti u odnosu na klasičan način rezerviranja i plaćanja parkiranja. Za početak, korištenjem ove aplikacije korisniku se značajno smanjuje vrijeme potrebno za pronalaženje parkirališnog mjesta. Dodatno, plaćanje parkirališnog mjesta može biti izvor frustracije zbog korištenja kovanica i gomilanja sitniša ili zbog potrebe za vađenjem kartice. Aplikacija za taj problem nudi praktično rješenje implementacijom novčanika što omogućuje korisniku da sva svoja plaćanja obavi unaprijed preko mobilnog telefona. Jednostavnost i elegantnost ove aplikacije pomaže i pri smanjenju stresa kod korisnika. Pitanja poput: "Gdje ćemo parkirati?", "Smijem li ja parkirati na ovom mjestu?" ili "Ima li uopće slobodnih mjesta?" više neće biti razlog za brigu jer aplikacija obavlja sve razmišljanje za vas. Problem traženja parkirališnog mjesta u nepoznatom gradu će se svesti na samo par klikova na aplikaciji te će uštedjeti korisnicima i novac budući da neće gubiti vrijeme vožajući se po gradu, tražeći slobodno mjesto. Konačno, aplikacija pridonosi i okolišu jer manje vremena potrošeno na potražnju mjesta za parkiranje znači manje vremena koje trošimo na sagorijevanje goriva.

Neke od aplikacija koje pružaju slična rješenja su:

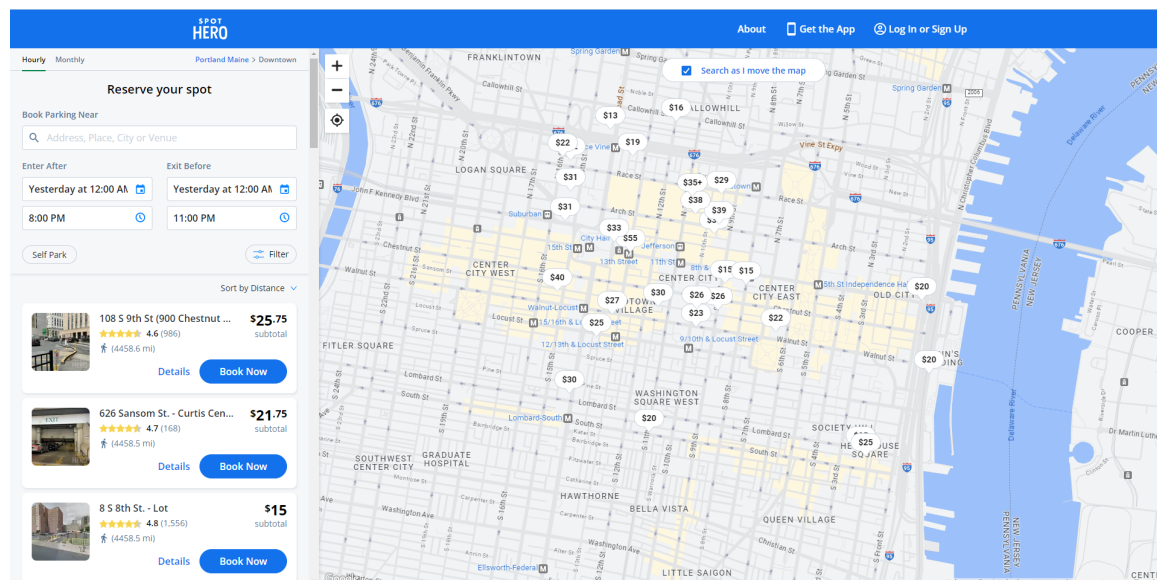
- SpotHero
- Best in Parking
- ParkWhiz
- Parkopedia
- Way

Kako bi bolje ilustrirali prednosti koje naša aplikacija ima u odnosu na neke od postojećih rješenja, uzet ćemo kao primjer *SpotHero*. *SpotHero* korisniku prikazuje dostupna parkirališna mjesta u blizini nakon što korisnik upiše adresu na kojoj se nalazi. Korisnik tada može kartu rezervirati i tada dobiva propusnicu koju skenira na parkirališnom mjestu. *SpotHero* omogućuje korisniku mjesečno parkiranje te parkiranje u zračnoj luci. Bitna razlika između naše aplikacije i *SpotHero* je to da kod korištenja *SpotHero-a* korisnik mora odmah definirati termin i vrijeme parkiranja dok naša aplikacija dozvoljava korisniku da prvo označi parkirališna mjesta, a nakon toga mu se nude dostupni termini. Aplikacije se razlikuju i u tome što *SpotHero* ne daje opciju za bicikle, već samo za osobna vozila. Dodatno, *SpotHero* je jedino dostupan u SAD-u i Kanadi.

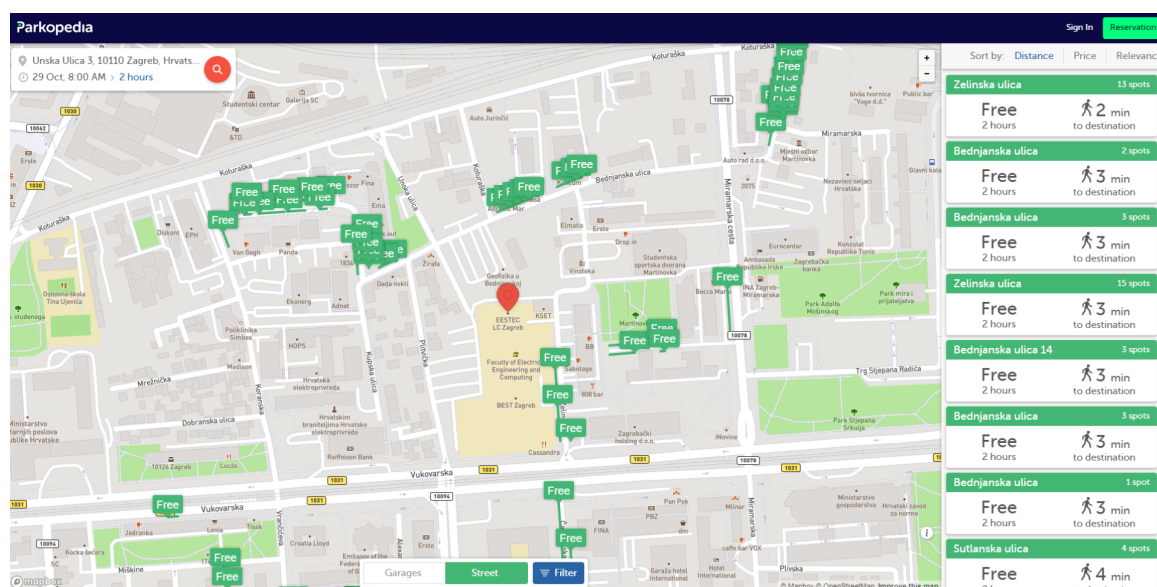
*Parkopedia* je još jedna takva slična aplikacija. *Parkopedia* koristi Google Maps kako bi iscrtala najbližu rutu korisniku. Razlika između našeg projekta i ove apli-



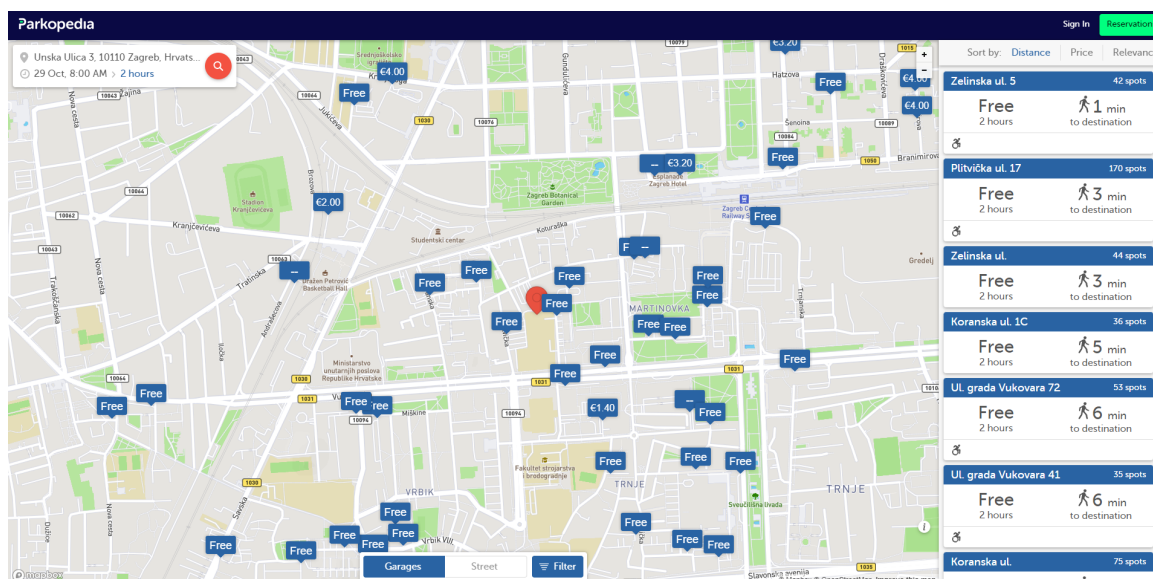
kacije je što *Parkopedia* zahtijeva da se prvo unese adresa odredišta nakon čega nudi popis dostupnih parkirališta. *Parkopedia* na prikazu karte daje opciju između prikazivanja parkirališnih mjesta na ulici i u garaži. Kao i kod *SpotHero-a*, *Parkopedia* ne nudi opciju za bicikle te pretraga prvo po terminu parkiranja nije moguća.



Slika 2.1: Prikaz aplikacije SpotHero



Slika 2.2: Prikaz aplikacije Parkopedia s označenim parkiralištima na ulici



Slika 2.3: Prikaz aplikacije Parkopedia s označenim parkiralištima na garaži

Vjerujemo da našom aplikacijom možemo zainteresirati ljude koji su već upoznati s parking aplikacijama, ali smatraju da postoje područja u kojima trenutno najpopularnije aplikacije nisu dovoljno prilagodljive korisničkim željama. Konačan plan i opseg ovog projekta je učiniti ovu aplikaciju dostupnu cijelom svijetu. Ovisno o potrebi i potražnji korisnika, aplikaciju je moguće u budućnosti i nadograditi funkcionalnostima koje druge aplikacije imaju poput mjesečnog rezerviranja parkiranja i slično.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

#### Dionici:

1. Voditelj parkirališta
2. Klijent
3. Administrator
4. Razvojni tim

#### Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) pregledati na karti pozicije svih parkirališta i dostupnim mjestima na parkiralištima
  - (b) poslati zahtjev za registraciju, a potrebni su: korisničko ime, lozinka, ime, prezime, slika osobne, IBAN račun i email adresa
    - i. moguće se registrirati kao klijent
    - ii. moguće se registrirati kao voditelj parkirališta
2. Klijent (inicijator) može:
  - (a) uz pregled parkirališta te dostupnih mjesta istih ima uvid u aktualnu zauzetost parkirnih mjesta
  - (b) unijeti odredište, tipa vozila i procijenjenu duljinu trajanja parkiranja čime generira rutu do najbližeg slobodnog parkirališta te rezervira parkirno mjesto ako postoji slobodno mjesto za rezervaciju
  - (c) rezervirati parkiranje:
    - i. prema slobodnim terminima tražene lokacije
    - ii. prema slobodnim lokacijama u traženom terminu
  - (d) nadopuniti novčanik sredstvima za plaćanje parkiranja
  - (e) pregledavati i mijenjati osobne podatke
  - (f) obrisati račun

1. Voditelj parkirališta(inicijator) može:

- (a) unijeti informacije o svom parkiralištu (naziv, opis, fotografija, cjenik i sl.)
  - i. u kartu ucrtati svako dostupno parkirališno mjesto za to parkiralište
- (b) definirati je li moguće rezervirati parkirališno mjesto
- (c) definirati cijenu ovisno o trajanju rezervacije
- (d) pristupiti statistici zauzetosti parkirališta i parkirališnih mjesta kroz vrijeme
- (e) obrisati parkirno mjesto
- (f) mijenjati informacije o svom parkiralištu
- (g) obrisati račun

1. Administrator (inicijator) može:

- (a) potvrditi neregistriranog korisnika kao voditelja parkirališta
- (b) pristupiti statistici zauzetosti parkirališta i parkirališnih mjesta kroz vrijeme
- (c) dodati ili obrisati parkiralište i parkirna mjesta
- (d) pristupiti i izmijeniti podatke o parkiralištu
- (e) korisnike brisati i mijenjati im razinu pristupa aplikaciji
- (f) obrisati parkiralište

1. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o parkiralištima i stanja zauzetosti parkirališnih mjesta

### 3.1.1 Obrasci uporabe

#### *dio 1. revizije*

#### Opis obrazaca uporabe

*Funkcionalne zahtjeve razraditi u obliku obrazaca uporabe. Svaki obrazac je potrebno razraditi prema donjem predlošku. Ako u nekom koraku može doći do odstupanja, potrebno je to odstupanje opisati i po mogućnosti ponuditi rješenje kojim bi se tijekom obrasca vratio na osnovni tijek.*

#### UC1 -Pregled parkirališta i parkirališnih mjesta na karti

- **Glavni sudionik:** Klijent, Neregistrirani korisnik
- **Cilj:** Pregledati dostupna parkirna mjesta
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Učitavanjem aplikacije prikazuje se karta s ucrtanim parkiralištima
  2. Korisnik odabire parkiralište
  3. Prikazuju se dostupna parkirališna mjesta za odabrano parkiralište te informacije o parkiralištu
- **Opis mogućih odstupanja:**
  - 3.a Glavni sudionik je registrirani Klijent
    1. Dodatno se prikazuje zauzetost parkirališnih mjesta u stvarnom vremenu

#### UC2 -Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za registraciju
  2. Korisnik unosi potrebne korisničke podatke
  3. Korisnik odabire vrstu registracije
  4. Korisnik registraciju potvrđuje preko maila

- **Opis mogućih odstupanja:**

2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila

1. Sustav upozorava korisnika na probleme u registraciji i vraća ga na stranicu registracije

3.a Odabrana je registracija računa Voditelja parkirališta

(a) Administrator mora dodatno potvrditi takvu registraciju

### **UC3 -Prijava u sustav**

- **Glavni sudionik:** Klijent

- **Cilj:** Dobiti pristup korisničkom sučelju

- **Sudionici:** Baza podataka

- **Preduvjet:** Registracija

- **Opis osnovnog tijeka:**

1. Unos korisničkog imena i lozinke

2. Potvrda o ispravnosti unesenih podataka

3. Pristup korisničkim funkcijama

- **Opis mogućih odstupanja:**

2.a Neispravno korisničko ime/lozinka

1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za prijavu

### **UC4 -Pregled osobnih podataka**

- **Glavni sudionik:** Klijent

- **Cilj:** Pregledati osobne podatke

- **Sudionici:** Baza podataka

- **Preduvjet:** Klijent je prijavljen

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Osobni podatci"

2. Otvara se stranica s osobnim podacima korisnika

### **UC5 -Promjena osobnih podataka**

- **Glavni sudionik:** Klijent

- **Cilj:** Promijeniti osobne podatke

- **Sudionici:** Baza podataka

- **Preduvjet:** Klijent je prijavljen

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Osobni podatci"
2. Otvara se stranica s osobnim podacima korisnika
3. Korisnik odabere opciju za promjenu podataka
4. Korisnik mijenja svoje osobne podatke
5. Korisnik sprema promjene
6. Baza podataka se ažurira

– **Opis mogućih odstupanja:**

- 2.a Korisnik promijeni svoje osobne podatke, ali ne odabere opciju "Spremi promjene"
  1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

#### UC6 -Brisanje korisničkog računa

- **Glavni sudionik:** Klijent
- **Cilj:** Obrisati korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Osobni podatci"
  2. Otvara se stranica s osobnim podacima korisnika
  3. Korisnik odabere opciju brisanja računa
  4. Korisnički račun se izbriše iz baze podataka
  5. Otvara se stranica za registraciju

#### UC7 -Rezervacija parkirnog mjesta

- **Glavni sudionik:** Klijent
- **Cilj:** Rezervirati parkirališno mjesto
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen, Klijent (ili aplikacija) je odabrao mjesto, datum i vrijeme parkiranja
- **Opis osnovnog tijeka:**
  1. Baza podataka zapiše podatke o rezervaciji
  2. Parkiranje se naplati pri rezervaciji ili po dolasku na parkirališno mjesto
- **Opis mogućih odstupanja:**
  - 1.a Klijent rezervaciju definira kao ponavljajuću
    1. Rezervacija se sprema u bazu podataka u posebnom obliku

**UC8 -Odabir rezervacije parkirnog mjesta**

- **Glavni sudionik:** Klijent
- **Cilj:** Rezervirati parkirališno mjesto
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent bira termin ili parkirališno mjesto te upisuje predviđeno trajanje rezervacije
  2. Ako je Klijent odabrao termin nude mu se odgovarajuća slobodna parkirališna mjesta, a ako je Klijent odabrao parkirališno mjesto nude mu se dostupni termini za dotično mjesto
  3. Klijent odabire mjesto, datum i trajanje rezervacije
  4. Klijent odabire je li rezervacija ponavljajuća
- **Opis mogućih odstupanja:**
  - 1.a ILI 3.a Klijent odabere današnji datum
    1. Sustav upozori korisnika da se ne može rezervirati mjesto na datum odabira

**UC9 -Dohvat parkirališta**

- **Glavni sudionik:** Klijent
- **Cilj:** Pronaći put do najbližeg parkirališta i rezervirati ako je moguće
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
  1. Klijent odabire odredište na karti, tip vozila i procjenu trajanja parkiranja
  2. Aplikacija iscrta rutu na mapi do najbližeg slobodnog parkirališnog mjesta
  3. Mjesto se rezervira ako je slobodno za rezervaciju

**UC10 -Pregled rezervacija**

- **Glavni sudionik:** Klijent
- **Cilj:** Pregled aktivnih rezervacija
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen i napravio je barem jednu rezervaciju
- **Opis osnovnog tijeka:**



1. Klijent odabire opciju "Moje rezervacije"
2. Prikaže se lista rezervacija korisnika

#### **UC11 -Uređivanje rezervacije**

- **Glavni sudionik:** Klijent
- **Cilj:** Uređivanje aktivne rezervacija
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen i napravio je barem jednu rezervaciju
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju "Moje rezervacije"
  2. Prikaže se lista rezervacija korisnika
  3. Klijent odabire rezervaciju koju želi urediti
  4. Prikaže se lista dostupnih parkirnih mjesta i slobodnih termina za iste
  5. Korisnik čini promjene te ih sprema

#### **UC12 -Brisanje rezervacije**

- **Glavni sudionik:** Klijent
- **Cilj:** Otkazati zakazanu rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen i postoji barem jedna aktualna rezervacija
- **Opis osnovnog tijeka:**
  1. Klijent odabire opciju "Moje rezervacije"
  2. Prikaže se lista rezervacija korisnika
  3. Klijent bira opciju za brisanje rezervacije
  4. Rezervacija se uklanja iz Baze podataka

#### **UC13 -Plaćanje**

- **Glavni sudionik:** Klijent
- **Cilj:** Platiti parkiranje
- **Sudionici:** Baza podataka, Voditelj parkirališta
- **Preduvjet:** Klijent je prijavljen, parkiralište se plaća aplikacijom
- **Opis osnovnog tijeka:**
  1. Uzme se određena količina iz računa Klijenta
  2. Stavlja se na račun Voditelja parkirališta koje se rezervira
  3. Rezervacija se označi plaćenom

– **Opis mogućih odstupanja:**

- 1.a Klijent nema dovoljno novca na računu
  - 1. Sustav upozori korisnika da nema dovoljno novca
  - 2. Sustav prekine sa tijekom izvođenja

**UC14 -Uplata**

– **Glavni sudionik:** Klijent

– **Cilj:** Uplatiti novac na račun

– **Sudionici:** Baza podataka

– **Preduvjet:** Klijent je prijavljen

– **Opis osnovnog tijeka:**

- 1. Klijent odabere opciju uplate novca
- 2. Klijent upiše podatke računa/kartica s koje se uzima novac i koliko novca želi prenijeti
- 3. Potvrđuje se transakcija
- 4. Dodaju se novi novci na račun

– **Opis mogućih odstupanja:**

2.a Krivo upisane informacije računa/kartice

- 1. Sustav upozori korisnika da je krivo upisao podatke
- 2. Sustav vrati korisnika na stranicu upisa podataka

3.a Nedovoljno novca na računu

- 1. Sustav upozori korisnika da nema dovoljno novca na računu
- 2. Sustav vrati korisnika na stranicu upisa podataka

**UC15 -Pregled aktivnih rezervacija parkirališta**

– **Glavni sudionik:** Voditelj parkirališta

– **Cilj:** Pregledati sve aktualne rezervacije

– **Sudionici:** Baza podataka

– **Preduvjet:** Rezervacije su zaprimljene i plaćene

– **Opis osnovnog tijeka:**

- 1. Voditelj odabere opciju pregleda aktivnih rezervaciju
- 2. Prikaže se lista aktivnih rezervacija

**UC16 -Ucrtavanje novog parkirnog mjesta**

– **Glavni sudionik:** Voditelj parkirališta

– **Cilj:** Ucrtati novo parkirno mjesto u sklopu postojećeg parkirališta

– **Sudionici:** Baza podataka

- **Preduvjet:** Voditelj je prijavljen
- **Opis osnovnog tijeka:**
  1. Voditelj odabire opciju prikaza tlocrta parkirališta
  2. Prikazuje mu se tlocrt parkirališta
  3. Voditelj odabire opciju za ucrtavanje novog parkirnog mjesta
  4. Voditelj ucrtava na tlocrt novo parkirno mjesto
  5. promjene se upisuju u Bazu podataka

#### **UC17 -Uređivanje podacima o parkiralištu**

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Izmijeniti informacije o parkiralištu
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj parkirališta je prijavljen
- **Opis osnovnog tijeka:**
  1. Voditelj parkirališta bira prikaz opciju prikaza informacija o parkiralištu
  2. Prikazuju mu se ranije unesene informacije
  3. Voditelj bira opciju izmjene dotičnih podataka
  4. Izmijenjeni podatci se spremaju u Bazu podataka

#### **UC18 -Uklanjanje parkirnog mjesta**

- **Glavni sudionik:** Voditelj parkirališta
- **Cilj:** Ukloniti željena parkirna mjesta
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj parkirališta je prijavljen, postoje ucrtana parkirna mjesta
- **Opis osnovnog tijeka:**
  1. Voditelj bira opciju prikaza tlocrta parkirališta
  2. Voditelj potom bira opciju uklanjanja parkirnog mjesta
  3. Odabrana uklonjena parkirna mjesta se uklanjaju iz Baze podataka

#### **UC19 -Potvrđivanje voditelja parkirališta**

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi registraciju računa Voditelja parkirališta
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je administrator i postoji nepotvrđena registracija za račun Voditelja parkirališta

– **Opis osnovnog tijeka:**

1. Administrator otvara listu registraciju na čekanju
2. Administrator potvrđuje određene registracije
3. U bazu podataka se spremaju promijene

**UC20 -Pregled Korisnika**

– **Glavni sudionik:** Administrator

– **Cilj:** Pregledati registrirane korisnike

– **Sudionici:** Baza podataka

– **Preduvjet:** Prijavljeni korisnik ima administratorska prava

– **Opis osnovnog tijeka:**

1. Administrator odabire opciju pregledavanja korisnika
2. Prikaže se lista svih ispravno registriranih korisnika s osobnim podacima

**UC21 -Brisanje korisnika**

– **Glavni sudionik:** Administrator

– **Cilj:** Ukloniti korisnika iz Baze podataka

– **Sudionici:** Baza podataka

– **Preduvjet:** Prijavljeni korisnik ima administratorska prava

– **Opis osnovnog tijeka:**

1. Administrator odabire opciju uklanjanja korisnika
2. Administrator pronalazi željenog korisnika
3. Administrator uklanja željenog korisnika i njegove podatke iz baze podataka

**UC22 -Promjena prava pristupa**

– **Glavni sudionik:** Administrator

– **Cilj:** Promijeniti prava korisnika

– **Sudionici:** Baza podataka

– **Preduvjet:**

– **Preduvjet:** Prijavljeni korisnik ima administratorska prava

– **Opis osnovnog tijeka:**

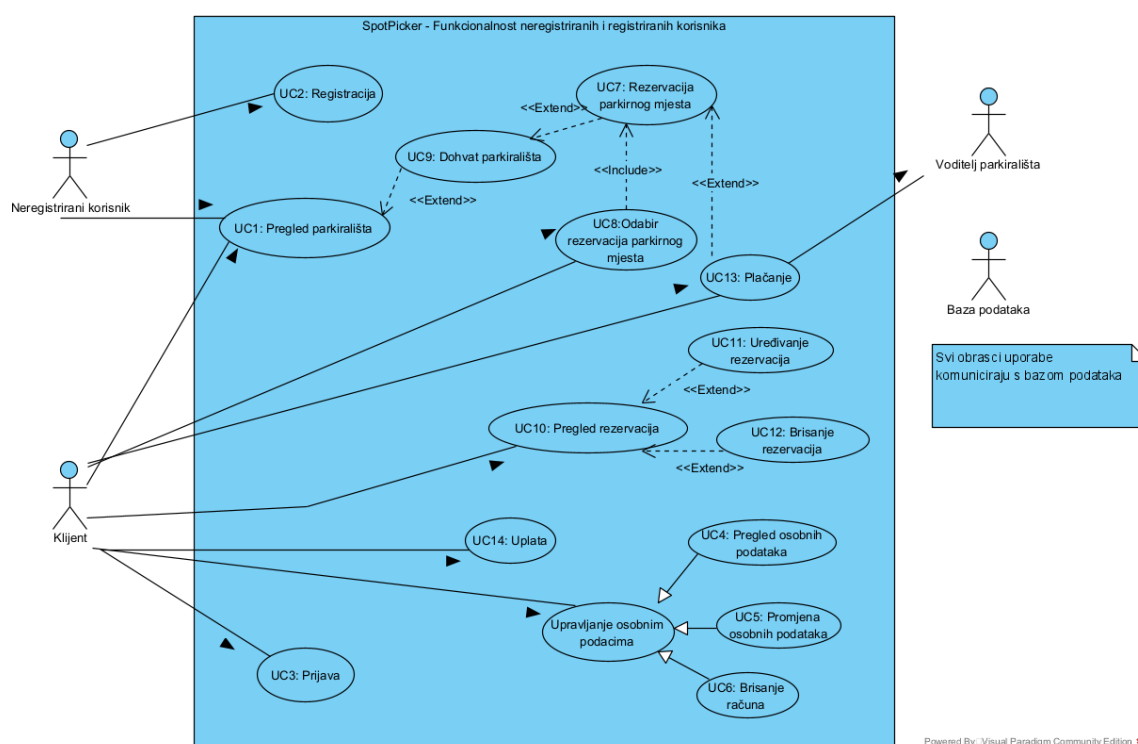
1. Administrator pronalazi željenog korisnika
2. Administrator mijenja razinu pristupa željenom korisniku

**UC23 -Pregled statistike**

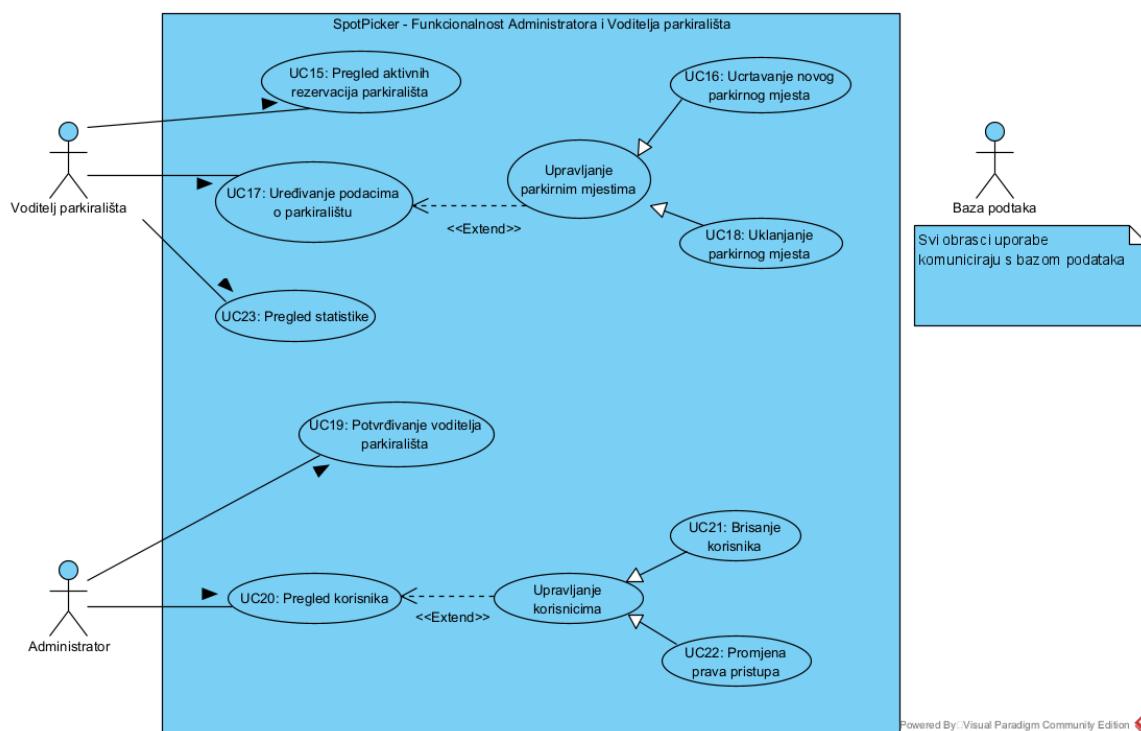
– **Glavni sudionik:** Voditelj parkirališta

- **Cilj:** Pregledati statistiku popunjenosti parkirališnih mjesta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen kao Voditelj parkirališta
- **Opis osnovnog tijeka:**
  1. Izabere se opcija prikaza statistike za parkiralište
  2. Prikazuje se statistika zauzetosti mjesta dotičnog parkirališta

### Dijagrami obrazaca uporabe



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

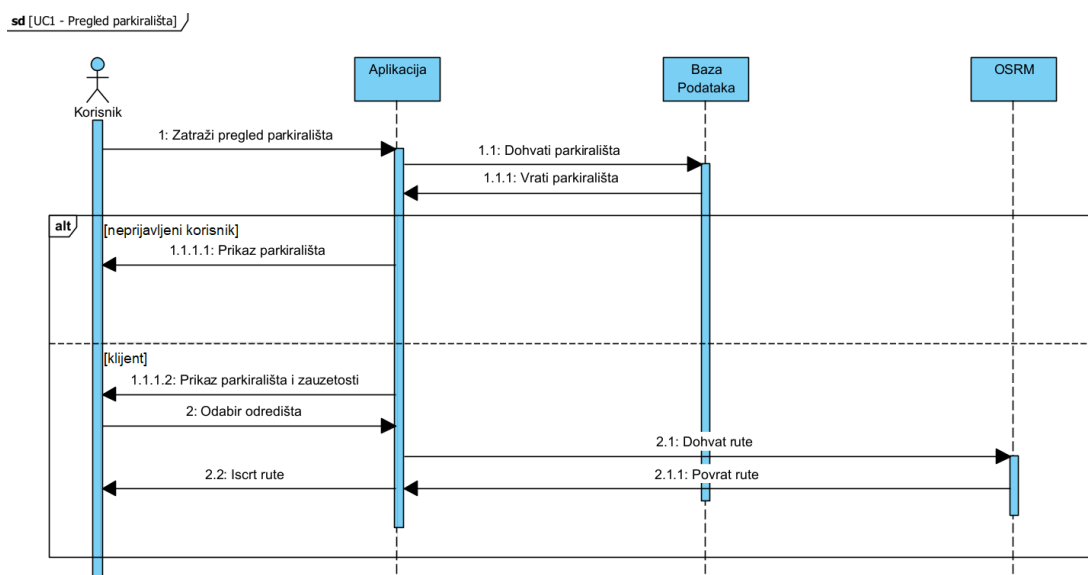


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost administratora i voditelja parkirališta

### 3.1.2 Sekvencijski dijagrami

#### Obrazac uporabe UC1 - Pregled parkirališta

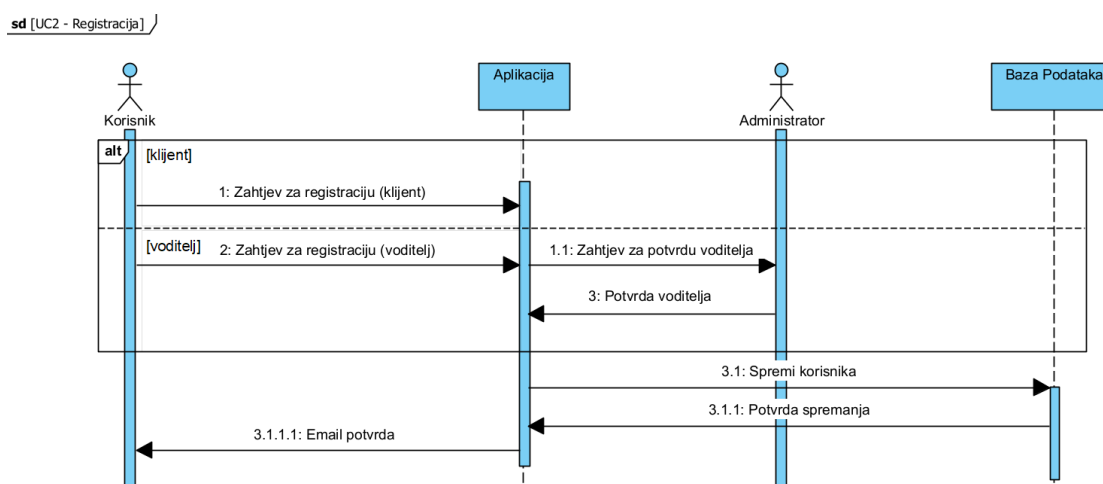
Korisnik šalje zahtjev za pregledom parkirališta kako bi mogao odabrati parkirališno mjesto. Poslužitelj šalje zahtjev prema bazi podataka i dohvaća iz nje parkirališta. Ako je korisnik neprijavljen tada mu poslužitelj samo prikazuje parkirališta. Ako je riječ o prijavljenom korisniku, odnosno klijentu tada mu poslužitelj prikazuje parkirališta s informacijom o zauzetosti. Klijent zatim odabire odredište pri čemu poslužitelj šalje zahtjev prema OSRM (Open Source Routing Machine) za dohvat rute. OSRM vraća rutu poslužitelju koji ju iscrtava klijentu.



Slika 3.3: Sekvencijski dijagram za UC1

## Obrazac uporabe UC2 - Registracija

Korisnik šalje zahtjev za registraciju sa željenom ulogom za koju se prijavljuje prema poslužitelju. Ako je željena uloga voditelj parkirališta, tada poslužitelj prima zahtjev za registraciju od korisnika i šalje prema administratoru zahtjev za potvrdu voditelja. Ako administrator potvrdi korisnika, potvrda se šalje poslužitelju. Ako je željena uloga klijent samo se šalje zahtjev za registraciju prema poslužitelju te nema potvrde administratora. Neovisno o ulozi koju je korisnik izabrao proces registracije se nastavlja. Poslužitelj šalje zahtjev prema bazi podataka za spremanjem korisnika. Baza podataka sprema korisnika i šalje nazad prema poslužitelju potvrdu spremanja. Konačno, poslužitelj šalje email potvrdu korisniku čime proces registracije završava.

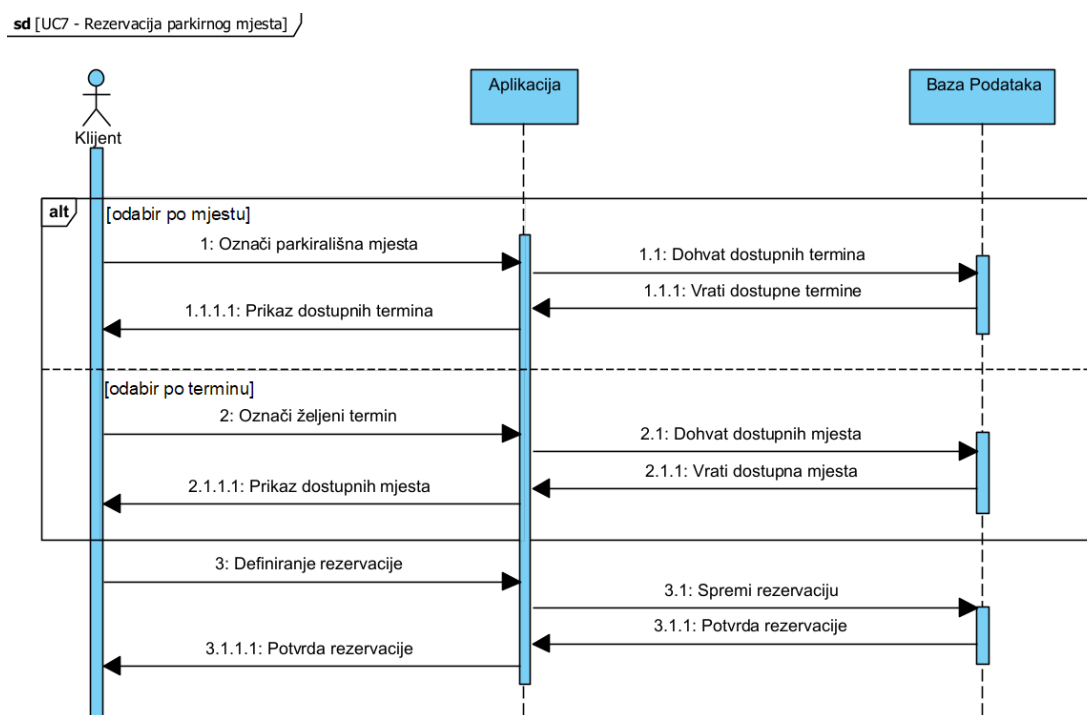


Slika 3.4: Sekvencijski dijagram za UC2

## Obrazac uporabe UC7 - Rezervacija parkirališnog mjesta



Klijent može rezervirati parkirališna mjesta na dva načina. Prvi način je odabir po mjestu, gdje korisnik označuje željena parkirališna mjesta te šalje zahtjev prema poslužitelju. Poslužitelj zatim šalje zahtjev prema bazi podataka za dohvat dostupnih termina. Baza podataka vraća dostupne termine poslužitelju nakon čega ih poslužitelj prikazuje klijentu. Drugi način je odabir po terminu, gdje korisnik odabire željeni termin i šalje zahtjev prema poslužitelju. Poslužitelj šalje zahtjev za dohvat dostupnih mjesta prema bazi podataka. Baza podataka vraća dostupna mjesta poslužitelju koji ih prikazuje klijentu. Nakon odabira načina rezervacije klijent šalje zahtjev za definiranje rezervacije prema poslužitelju. Poslužitelj šalje zahtjev za spremanjem rezervacije prema bazi podataka te baza podataka vraća potvrdu rezervacije. Poslužitelj potvrdu rezervacije prosljeđuje nazad prema klijentu čime rezervacija parkirališnog mjesta završava.

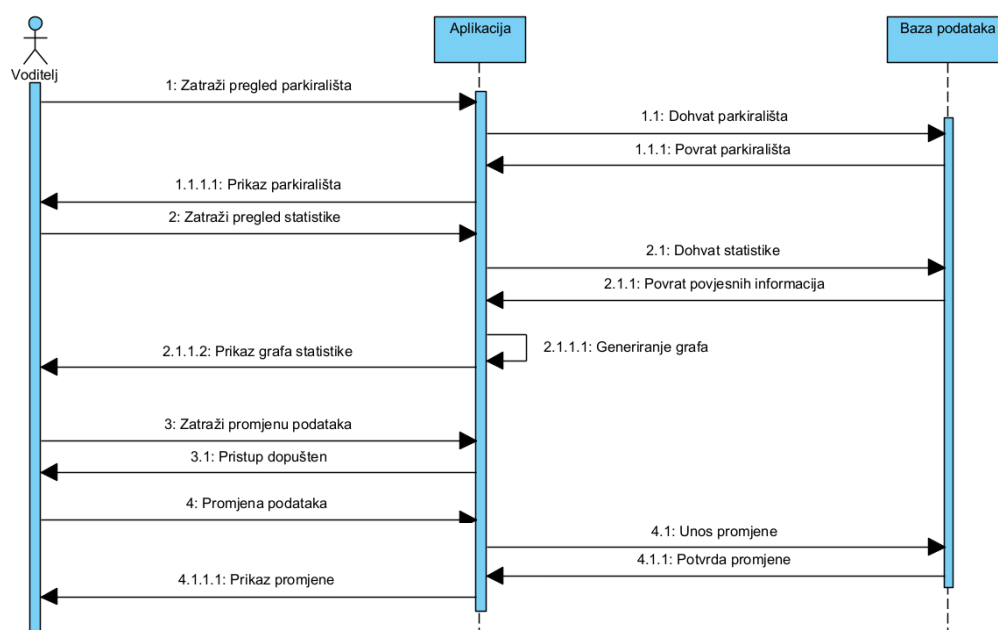


Slika 3.5: Sekvencijski dijagram za UC7

### Obrazac uporabe UC17 - Uređivanje podataka o parkiralištu

Vlasnik šalje zahtjev za pregledom parkirališta. Poslužitelj šalje zahtjev za dohvat parkirališta prema bazi podataka. Baza podataka vraća parkirališta poslužitelju koji ih prikazuje vodicelju. Voditelj traži pregled statistike zauzetosti parkirališta i parkirališnih mjesta kroz vrijeme. Poslužitelj šalje zahtjev za dohvaćanje statistike prema bazi podataka. Baza podataka vraća povijesne informacije o zauzetosti parkirališta poslužitelju. Poslužitelj na temelju dobivenih povijesnih informacija generira statistiku u obliku grafa. Dobiveni graf statistike prikazuje se vlasniku. Vlasnik šalje zahtjev za promjenu podataka poslužitelju te mu poslužitelj vraća dozvolu za pristup. Vlasnik mijenja podatke i šalje promjene poslužitelju koji promjene prosljeđuje bazi podataka. Baza podataka potvrđuje promjene i vraća potvrdu poslužitelju koji prikazuje promjene vlasniku.

sd [UC17 - Uređivanje podataka o parkiralištu]



Slika 3.6: Sekvencijski dijagram za UC17

## 3.2 Ostali zahtjevi

### *dio 1. revizije*

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.

1. Koristiti OSRM za dohvat rute na mapi
2. Koristiti Overpass API za dohvat početne informacije o parkirališnim mjestima
3. Sustav mora biti implementiran kao web aplikacija koristeći isključivo objektno-orijentirane jezike
4. Statistika parkirališta se prikazuje u obliku grafa

## 4. Arhitektura i dizajn sustava

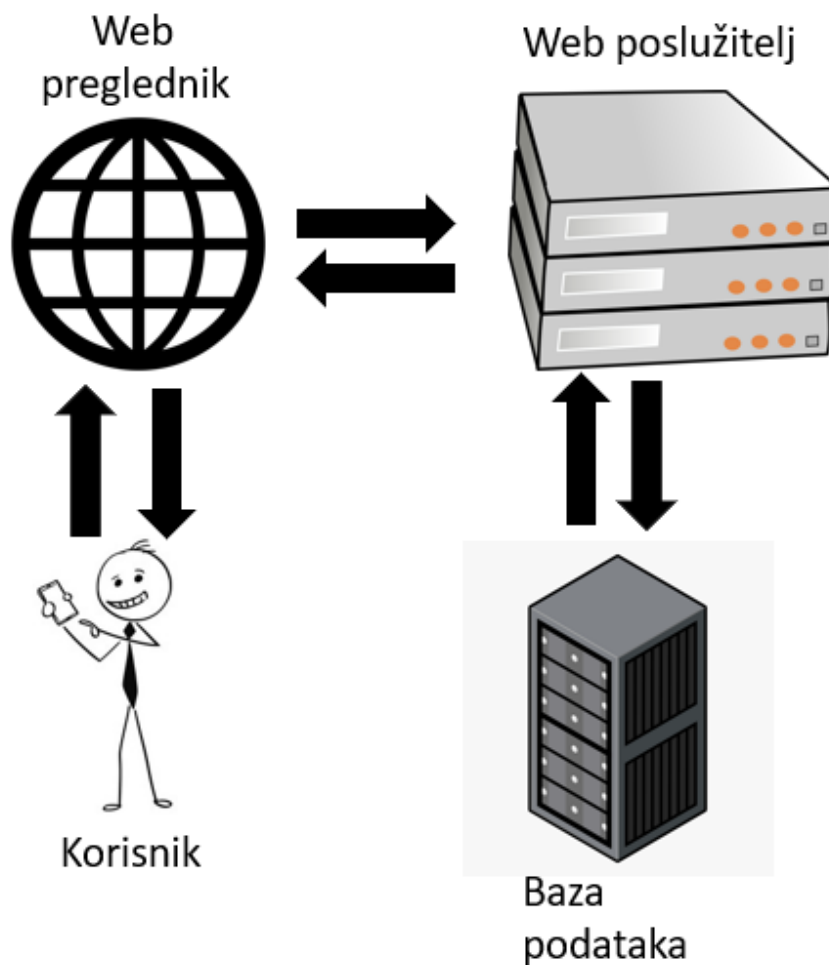
Arhitektura sustava je struktura sustava koja sadrži: elemente programa, njihove izvana vidljive karakteristike te odnose među njima. Arhitektura sustava se dijeli na tri manja sustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

Web preglednik je program preko kojega korisnik može pregledavati web-stranice te multimedijalne sadržaje od tih web-stranica. Web-stranica je napisana u kodu koji web preglednik prevodi tako da svatko razumije web-stranicu. Web preglednik također služi kao kanal između korisnika i web poslužitelja.

Web poslužitelj služi za komunikaciju između klijenta i aplikacije. Ta komunikacija se odvija preko HTTP-a. HTTP je protokol koji se koristi za prijenos podataka na webu. Web poslužitelj također pokreće web aplikaciju i prosljeđuje joj zahtjeve.

Web aplikacija obrađuje zahtjeve od korisnika, a za pojedine zahtjeve ona pristupa bazi podataka nakon čega se odgovor šalje korisniku putem web poslužitelja te web preglednika u HTML obliku.



Slika 4.1: Arhitektura sustava

U našem sustavu za razvoj radnog okvira na poslužiteljskoj strani (backend-u) odlučili smo koristiti *Spring Boot*, a na klijentskoj strani (frontend-u) smo odlučili koristiti *React*. Programske jezike za koje smo se odlučili su *Java* i *JavaScript*.

Korišteni razvojni okvir Spring se služi tzv. MVC arhitekturom. Takva arhitektura podrazumijeva sljedeću podjelu:

- **Model:** Središnja komponenta sustava, dohvat te manipulacija podataka
- **View:** dostupni razni prikazi podataka (tablično, grafom)
- **Controller:** upravlja zahtjevima korisnika te na temelju njih izvodi daljnju interakciju s ostalim komponentama

Naša web aplikacija koristi višeslojni stil arhitekture. Prednosti višeslojnog stila arhitekture su brojne:

- olakšava razvoj programa
- timovi se mogu raspodijeliti na razvoj zasebnih slojeva arhitekture
- podjela briga odnosno svaki sloj se mora brinuti za samo svoju zadaću
- moguće je jednostavno povećanje i poboljšanje sustava

## 4.1 Baza podataka

### 4.1.1 Vrsta i implementacija

Za modeliranje sustava je korištena relacijska baza podataka, a nju smo implementirati pomoću open-source sustava za upravljanje bazama podataka PostgreSQL. Za izradu dijagrama ER i generiranje relacijske sheme je korišten besplatan online alat ERDplus (<https://erdplus.com/>) koji je korišten i u sklopu predmeta Baze podataka.

### 4.1.2 Glavne komponente baze podataka

Baza podataka sastoji se od sljedećih entiteta:

- **Korisnik** (*Korisnicko\_ime, Lozinka, Ime, Prezime, Slika\_osobne, IBAN, email, Uloga, stanjeNaRacunu*)
- **ParkiralisteAutomobili** (*fotografija, Naziv, Opis, CijenaPoSatu, ParkiralisteId*)
- **Parking\_mjesto** (*identifikacija, Oznaka, Slobodno, Dostupno, ParkiralisteId*)
- **Voditelj** (*jePrihvacen, korisnickoIme, ParkiralisteId*)
- **Rezervacija** (*RezervacijaID, cijena, pocetakRezervacije, krajRezervacije, RRule, KorisnickoIme, Identifikacija*)
- **BicikliMjesto** (*identifikacija, BrojMjesta, duljina, sirina*)

### 4.1.3 Opis tablica

**Korisnik:** sadržava sve bitne informacije o registriranim korisnicima u sustavu. Sadrži attribute: ID korisnika, korisničko ime, lozinka, ime, prezime,

slika osobne iskaznice, IBAN, email, uloga (enum za 'klijent', 'voditelj', 'administrator') i stanje na računu. Ovaj entitet je preko korisnickogImena u vezi One-to-Many s entitetom Rezervacija i u One-to-One vezi s entitetom Voditelj.

Korisnik		
KorisničkoIme	VARCHAR	Jedinstveno korisničko ime
email	VARCHAR	Email korisnika
lozinka	VARCHAR	Hash lozinke dobiven s Bcrypt encoderom
Ime	VARCHAR	Ime korisnika
Prezime	VARCHAR	Prezime korisnika
IBAN	VARCHAR	IBAN korisnika
slikaOsobne	BYTEA	Byte array slike osobne iskaznice
Uloga	INT	Enum koji označava ulogu korisnika ('klijent', 'voditelj', 'administrator')
stanjeNaRacunu	NUMERIC	Decimalni broj koji označava koliko novaca se nalazi u novčaniku

**ParkiralisteAutomobili:** sadrži ključne informacije vezane za parkirališta automobila. Sadrži attribute koje postavlja voditelj parkirališta: fotografija, naziv, opis i cijena po satu. Uz to sadrži i ID parkirališta. Ovaj entitet je u vezi One-to-One s entitetom voditelj putem ID-a parkirališta i One-to-Many vezi s ParkingMjesto preko ID-a parkirališta.

ParkiralisteAutomobili		
ParkiralisteId	INT	Jedinstveni identifikator parkirališta
fotografija	BYTEA	Slika parkirališta koju voditelj može priložiti
Naziv	VARCHAR	Naziv parkirališta
Opis	VARCHAR	Opis parkirališta
CijenaPoSatu	NUMERIC	Cijena po satu koju definira voditelj parkirališta

**Parking\_mjesto:** predstavlja informacije o pojedinačnim parkirnim mjestima unutar parkirališta za automobile. U ovu tablicu ćemo unijeti početne informacije o mjestima za automobile koje smo dobili od overpass API-ja. Sadrži attribute kao što su identifikacija i koordinate (duljina, širina) koje ćemo dobiti pozivom overpassAPI-a. (@id i "coordinates" iz GeoJSON-a dobivenog pozivom overpassAPI-a), oznaka parkirnog mjesta, informacija o dostupnosti koju postavlja vođa i slobodnom statusu parkirnog mjesta te ID parkirališta kojem pripada. U U vezi je Many-to-One s entitetom "ParkirališteAutomobili" putem ID-a parkirališta i One-to-many s entitetom Rezervacija putem identifikacije parking mjesta.

Parking_mjesto		
identifikacija	VARCHAR	Identifikacija mjesta koje generira overpassAPI koji generira overpassAPI. Npr. "node/11310562209"
Oznaka	VARCHAR	Oznaka ili broj parkirnog mjesta
Duljina	NUMERICAL	Označuje duljinu (longitude)
Širina	NUMERICAL	Označuje širinu (longitude)
Slobodno	BOOLEAN	Je li mjesto slobodno ili nije
Dostupno	BOOLEAN	Je li vođa učinio mjesto dostupnim
ParkirališteId	INT	Jedinstveni identifikator parkirališta (ParkirališteAutomobili.ParkirališteID)

**Voditelj:** sadrži informacije o vođama parkirališta. Sadrži status prihvatanosti vođe i povezuje vođu sa parkiralištem. Ima vezu s One-to-one s entitetima Korisnik i ParkirališteAutomobila.

Voditelj		
jePrihvacen	BOOLEAN	Je li vođa prihvaćen
korisnickoIme	VARCHAR	Jedinstveno korisničko ime
ParkirališteId	INT	Jedinstven ID parkinga (ParkirališteAutomobili.ParkirališteID)



**Rezervacija:** sadrži informacije o parking rezervacijama koje su napravili korisnici. Uz cijenu i timestampe za pocetak i kraj rezervacije sadrži i string (TEXT) RRule kojim se definira učestalost ponavljanja rezervacije. Taj string se može parsirati pa generirati instance ponavljajućih rezervacija. Ovaj entitet ima vezu Many-to-One s entitetom Korisnik putem jedinstvenog korisničkog imena. Također, ima vezu Many-to-One s entitetom "ParkingMjesto" putem lokacije parkirnog mjesta.

Rezervacija		
RezervacijaID	INT	Jedinstven identifikator rezervacije
Cijena	NUMERIC	Ukupna cijena rezervacije
pocetakRezervacije	TIMESTAMP	Jedinstveni identifikator rezervacije
krajRezervacije	TIMESTAMP	Ukupna cijena rezervacije
RRule	TEXT	String koji će definirati učestalost ponavljanja rezervacije na osnovi formata iCalendar (RFC 5545) (ako je NULL onda je jednokratna)
korisnickoIme	INT	Jedinstveno korisničko ime
identifikacija	INT	Jedinstven identifikator parkirnog mjesta

**BicikliParking:** unosimo početne informacije o biciklističkim mjestima. Pošto se parkirališta za bicikle ne mogu rezervirati niti se naplaćuju, ovaj entitet je odvojen od ostatka sustava. Sadrži attribute: koordinate (širina i duljina) te identifikaciju koje dobivamo pozivom overpassAPI-ja. Duljina je prva vrijednost, a širina druga vrijednost u coordinates arrayu.

Npr. "coordinates": [16.0166069 (duljina), 45.8387721 (sirina)]

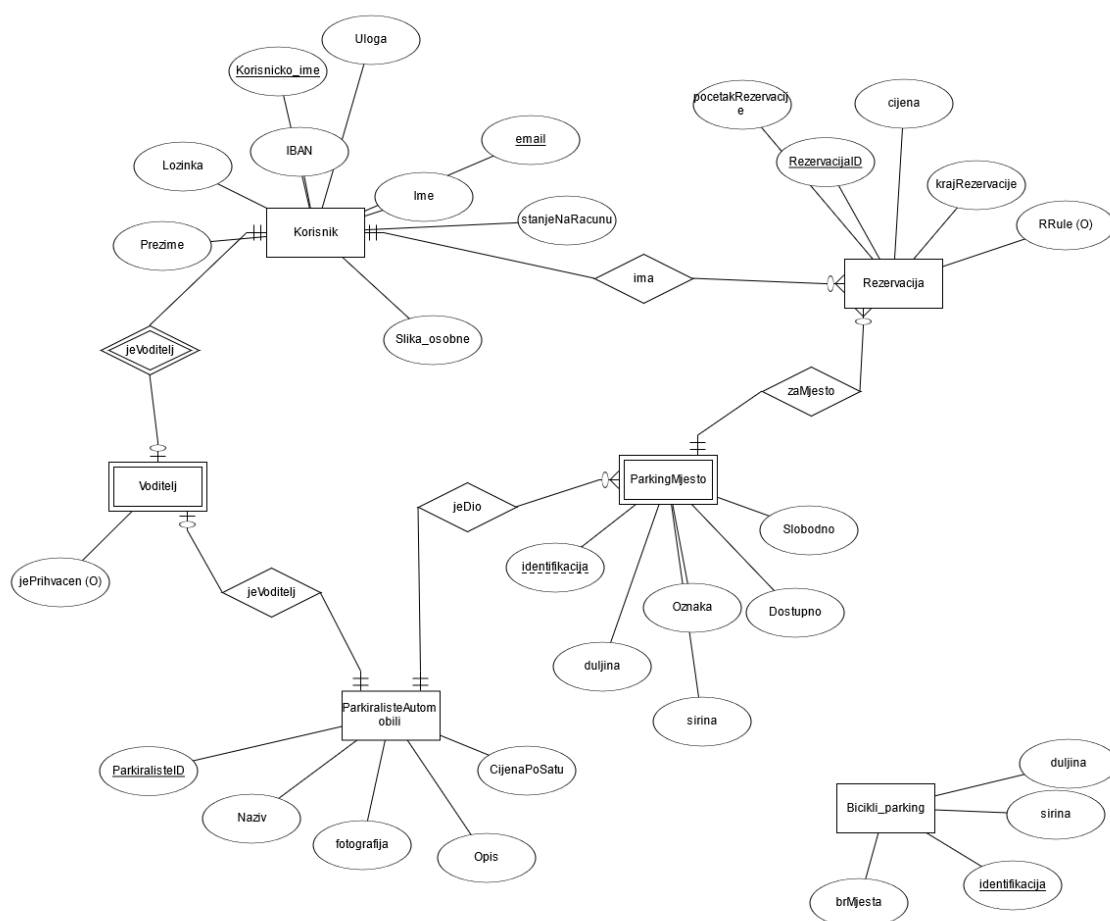
BicikliParking		
identifikacija	VARCHAR	Jedinstven identifikator mjesta (@id iz GeoJSON-a)

Nastavljeno na idućoj stranici

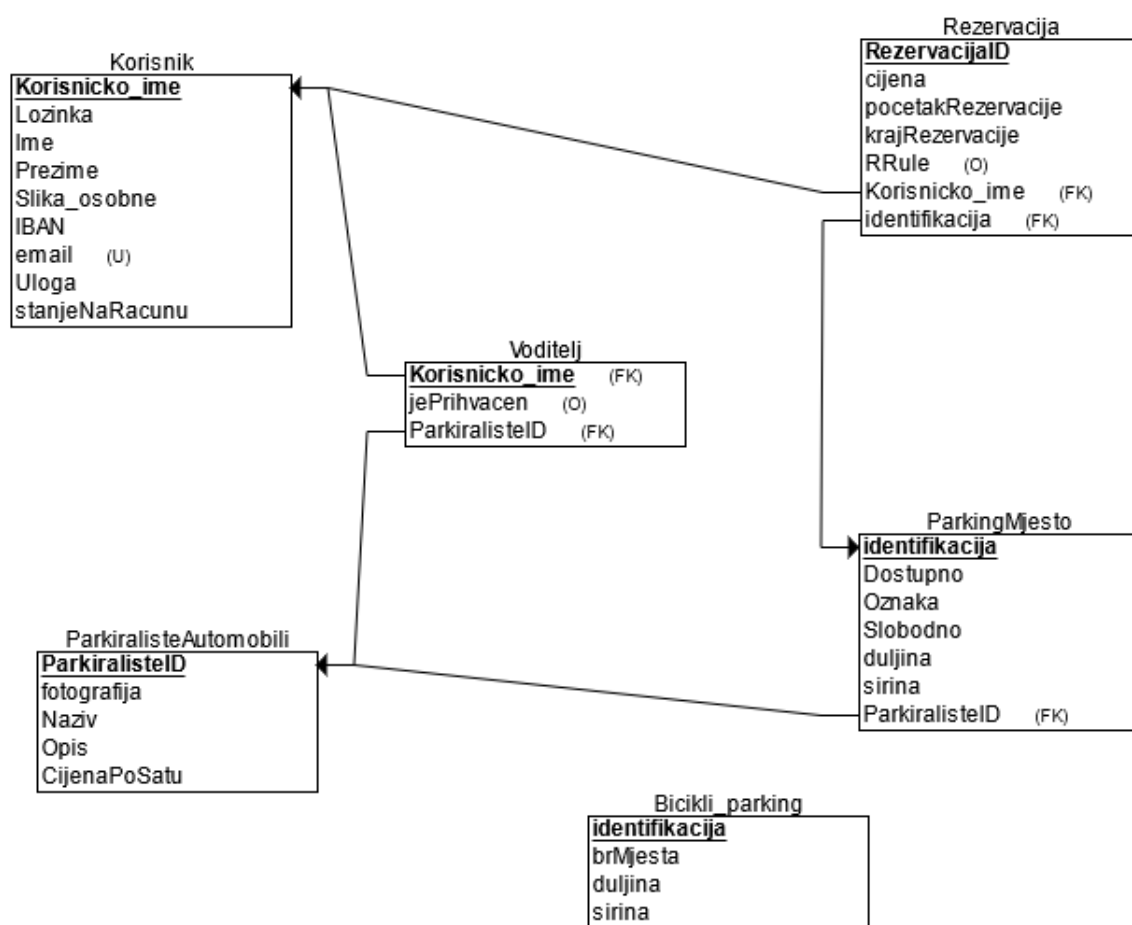
Nastavljeno od prethodne stranice

BicikliParking		
brojMjesta	INT	Broj dostupnih mjesta na parkingu za bicikle
duljina	NUMERICAL	(longitude) prva vrijednost u "coordinates" nizu
sirina	NUMERICAL	(latitude) druga vrijednost u "coordinates" nizu

#### 4.1.4 Dijagram baze podataka



Slika 4.2: E-R dijagram baze podataka



Slika 4.3: Relacijska shema

## 4.2 Dijagram razreda

*Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.*

### **dio 1. revizije**

*Prilikom prve predaje projekta, potrebno je priložiti potpuno razrađen dijagram razreda vezan uz **generičku funkcionalnost** sustava. Ostale funkcionalnosti trebaju biti idejno razrađene u dijagramu sa sljedećim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazivi atributa razreda, veze i odnosi između razreda.*

### **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.*



## 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	Prikaz aplikacije SpotHero . . . . .	8
2.2	Prikaz aplikacije Parkopedia s označenim parkiralištima na ulici	8
2.3	Prikaz aplikacije Parkopedia s označenim parkiralištima na garaži	9
3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranih i re- gistriranih korisnika . . . . .	20
3.2	Dijagram obrasca uporabe, funkcionalnost administratora i vo- ditelja parkirališta . . . . .	21
3.3	Sekvencijski dijagram za UC1 . . . . .	22
3.4	Sekvencijski dijagram za UC2 . . . . .	23
3.5	Sekvencijski dijagram za UC7 . . . . .	24
3.6	Sekvencijski dijagram za UC17 . . . . .	25
4.1	Arhitektura sustava . . . . .	28
4.2	E-R dijagram baze podataka . . . . .	33
4.3	Relacijska shema . . . . .	34

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

*U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.*

#### 1. sastanak

- Datum: u ovom formatu: 17. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - \* opis prve teme
  - \* opis druge teme

#### 2. sastanak

- Datum: u ovom formatu: 17. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - \* opis prve teme
  - \* opis druge teme

## Tablica aktivnosti

### *Kontinuirano osvježavanje*

*Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.*

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							

Nastavljeno na idućoj stranici



Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

## 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](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*