Oblikovanje programske potpore

Ak. god. 2018./2019.

*ParkirajMe*

Dokumentacija, Rev. 1

Grupa: *ferovci*

Voditelj: *Hrvoje Ditrih*

Datum predaje: *28. studeni 2018.*

Nastavnik: *Nikolina Frid*

Sadržaj

[1. Dnevnik promjena dokumentacije 4](#_Toc535392094)

[2. Opis projektnog zadatka 5](#_Toc535392095)

[3. Pojmovnik 7](#_Toc535392096)

[4. Funkcionalni zahtjevi 8](#_Toc535392097)

[4.1 Opis obrazaca uporabe 9](#_Toc535392098)

[4.2 Dijagrami obrazaca uporabe 16](#_Toc535392099)

[4.3 Sekvencijski dijagrami 17](#_Toc535392100)

[5. Ostali zahtjevi 26](#_Toc535392101)

[6. Arhitektura i dizajn sustava 27](#_Toc535392102)

[6.1. Svrha, opći prioriteti i skica sustava 27](#_Toc535392103)

[6.2. Dijagram razreda s opisom 29](#_Toc535392104)

[6.3. Dijagram objekata 32](#_Toc535392105)

[6.4. Ostali UML dijagrami 33](#_Toc535392106)

[7. Implementacija i korisničko sučelje 40](#_Toc535392107)

[7.1. Dijagram razmještaja 40](#_Toc535392108)

[7.2. Korištene tehnologije i alati 41](#_Toc535392109)

[7.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava 42](#_Toc535392110)

[7.4. Ispitivanje programskog rješenja 44](#_Toc535392111)

[7.4.1 Test 1: Registracija 44](#_Toc535392112)

[7.4.2 Test 2: Prijava 47](#_Toc535392113)

[7.4.3 Test 3: Dodavanje automobila 48](#_Toc535392114)

[7.4.4 Test 4: Uređivanje podataka o parkiralištu 49](#_Toc535392115)

[7.4.5. Test 5: Jednokratna rezervacija 51](#_Toc535392116)

[7.4.6. Test 6: Ponavljajuća rezervacija 52](#_Toc535392117)

[7.4.7. Test 7: Trajna rezervacija 53](#_Toc535392118)

[7.4.8. Test 8: Brisanje tvrtke 54](#_Toc535392119)

[7.4.9. Test 9: Ukidanje trajne rezervacije 55](#_Toc535392120)

[7.4.10. Test 10: Parkiraj Me 56](#_Toc535392121)

[7.5. Upute za instalaciju 57](#_Toc535392122)

[7.6. Korisničke upute 58](#_Toc535392123)

[7.6.1 Anonimni korisnik 58](#_Toc535392124)

[7.6.2 Prijavljeni korisnik 61](#_Toc535392125)

[7.6.3 Prijavljena tvrtka 64](#_Toc535392126)

[7.6.4 Administrator 66](#_Toc535392127)

[8. Zaključak i budući rad 68](#_Toc535392128)

[9. Popis literature 69](#_Toc535392129)

[Chapter 2 Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 70](#_Toc535392130)

[Chapter 3 Dodatak B: Dnevnik sastajanja 71](#_Toc535392131)

[Chapter 4 Dodatak C: Prikaz aktivnosti grupe 72](#_Toc535392132)

[Chapter 5 Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 74](#_Toc535392133)

# Dnevnik promjena dokumentacije

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev.** | **Opis promjene/dodatka** | **Autor(i)** | **Datum** |
| 0.1 | Napravljen predložak | Ditrih | 16.10.2018. |
| 0.2 | Opis projektnog zadatka | Požeg, Bilanović | 17.10.2018. |
| 0.3 | Funkcionalni zahtjevi | Bilanović,  Požeg | 18.10.2018. |
| 0.4 | Use Case | Škudar,  Hudiček | 29.10.2018. |
| 0.5 | UML dijagrami | Pavlović,  Požeg | 30.10.2018. |
| 0.6 | Dorađeni funkcionalni zahtjevi | Ditrih | 2.11.2018. |
| 0.61 | Sekvencijski dijagrami | Ditrih, Požeg | 3.11.2018. |
| 0.7 | Arhitektura i dizajn sustava | Bilanović | 11.11.2018. |
| 0.91 | Dodatak A i B | Bilanović | 26.11.2018. |
| 0.92 | Dijagram razreda | Škudar | 27.11.2018. |
| 0.93 | ER model baze podataka | Pavlović | 27.11.2018. |
| **1.0** | **Zaključak, literatura, dodaci i finalna revizija** | Bilanović, Ditrih | **28.11.2018.** |
| 1.1 | Ispravci dokumentacije | Požeg, Bilanović | 7.1.2019. |
| 1.2 | Komunikacijski dijagram | Bilanović | 8.1.2019. |
| 1.3 | Dijagram aktivnosti | Bilanović | 9.1.2019. |
| 1.4 | Dijagram stanja, korisničke upute, tehnologije | Požeg | 13.1.2019. |

# Opis projektnog zadatka

Cilj ovog projekta je izraditi web aplikaciju koja će korisnicima pružati informacije o slobodnim parkirnim mjestima te njihovu rezervaciju na području grada Zagreba. Raspoloživa parkirališta bit će sva ona koje će zainteresirane tvrtke registrirati putem iste aplikacije. Aplikacija će raditi u stvarnom vremenu te koristiti korisnikovu lokaciju pomoću koje će se odrediti najbliže parkiralište sa slobodnim mjestom. Nakon određivanja parkirnog mjesta korisniku se šalje informacija o lokaciji parkinga.

Razlog pokretanja projekta je olakšavanje i ubrzavanje procesa traženja parkirnog mjesta u gradu. Želja je korisniku ponuditi individualni odabir te cijene odabranog parkirališta. Također pružamo lakši način zainteresiranim tvrtkama za promidžbu svojih parkirališta te lakšu digitalnu naplatu.

Glavni aktori uz korisnike su tvrtke koje nude svoja parkirna mjesta. Kako bi koristio usluge rezervacije parkirališta, korisnik se prvo mora registrirati na web stranici, što je omogućeno svakoj punoljetnoj osobi. Korisnik tijekom regristracije unosi sljedeće podatke: OIB, ime, prezime, adresa e-pošte, broj registracije svog automobila te broj kreditne kartice.

Korisnik ima na raspolaganju opciju traženja najbližih slobodnih parkirališta, što se odvija na sljedeći način: korisnik odabire gumb za pretragu najbližih slobodnih mjesta temeljenih na njegovoj lokaciji. Zatim dobiva na odabir nekoliko najboljih opcija zajedno sa informacijom o cijeni te on na temelju toga odabire parking. Registrirani korisnik ima na raspolaganju sljedeće opcije rezervacije:

* 1. Jednokratna

-za vremenski period kraći od 24 sata

-obavlja se minimalno 6 sati unaprijed

-naplaćuje se odmah u trenutku rezervacije izravnim terećenjem kreditne kartice

* 1. Ponavljajuća

-mora trajati najmanje 1 sat i ponavljati se barem jednom tjedno tijekom mjesec dana ili trajno (0-24h svaki dan na neodređeni period)

-trajna rezervacija naplaćuje se svakih 30 dana

Tvrtke koje žele nuditi svoja parkirališta unutar aplikacije također se trebaju registrirati putem aplikacije. Za registraciju ovlašteni zaposlenik unosi sljedeće podatke: OIB, ime, adresa sjedišta, adresa e-pošte. Nakon registracije unose se podatci o određenim parkiralištima koje ta tvrtka nudi. Kada korisnik izvrši rezervaciju on plaća izravno aplikaciji koja ta sredstva raspodjeljuje tvrtkama vlasnicima parkinga.

# Pojmovnik

WEB (World Wide Web) – najkorištenija usluga interneta koja omogućava dohvaćanje hipertekstualnih dokumenata.

Frontend – obuhvaća prezentacijski sloj aplikacije, točnije, onaj dio aplikacije koji je vidljiv korisniku i preko kojeg korisnik radi sa aplikacijom.

Backend – obuhvaća sloj podatkovne poveznice aplikacije, točnije, onaj dio aplikacije koji nije vidljiv korisniku i koji je zadužen za obavljanje većine operacija u aplikaciji.

Open-source softver je vrsta računalnog softvera u kojem se izvorni kod izdaje pod licencom u kojoj nositelj autorskih prava korisnicima daje pravo na proučavanje, promjenu i distribuciju softvera bilo kome i za bilo koju svrhu.

Run-time okruženje je okruženje koje osigurava prevođenje i izvršavanje programa.

API (Aplikacijsko programsko sučelje) je skup određenih pravila i specifikacija koje programeri slijede tako da se mogu služiti uslugama i resursima operacijskog sustava.

Hipertekst je jedan od ključnih temeljnih koncepata World Wide Weba, gdje su web stranice često pisane na jeziku hipertekstualnog označavanja.(HTML)

# Funkcionalni zahtjevi

Dionici:

* Korisnici
* Tvrtke
* Administratori

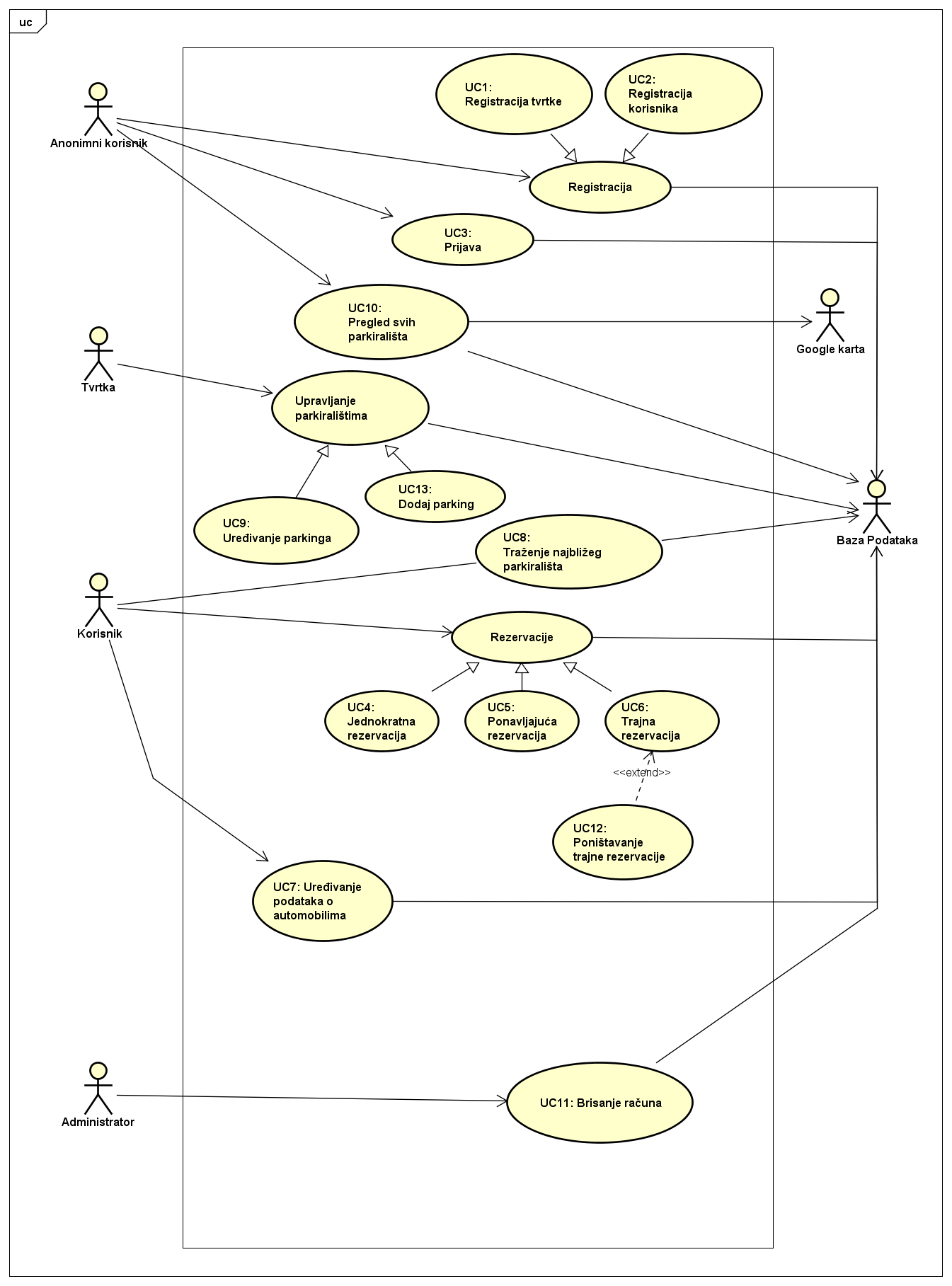
Aktori:

* Svi posjetitelji WEB stranice (anonimni korisnici)
  + Mogu se registrirati kao korisnik ili tvrtka te prijaviti u aplikaciju
* Korisnici
  + Može se prijaviti u sustav
  + Može zatražiti informacije o slobodnim parkirnim mjestima
  + Može zatražiti cijenu određenih parkirnih mjesta
  + Može odabrati određeni parking od ponuđenih
  + Može rezervirati parkirno mjesto jednokratno
  + Može rezervirati ponavljajuće parkirno mjesto
  + Može izvršiti uplatu rezervacije
* Tvrtke
  + Može se prijaviti u sustav
  + Može ponuditi informacije o svojim parkiralištima
* Administratori
  + Može upravljati korisnicima sustava
  + Može brisati korisničke račune(korisnike ili tvrtke)
* Baza podataka:
  + Sprema podatke o registriranim korisnicima i tvrtkama
  + Sprema podatke o rezervacijama
* Poslužitelj:
  + Obrađuje zahtjeve korisnika
  + Prati trenutna slobodna mjesta svih parkirališta
* Parkirna mjesta:
  + Prate trenutnu popunjenost parkirališta
* Google karta:
  + Pruža uslugu pregledavanja parkirnih mjesta na karti

## 4.1 Opis obrazaca uporabe

* Rbr.: UC1
  + Naziv: Registracija tvrtke
  + Opis: Registracija tvrtke na web aplikaciju
  + Glavni aktor: Anonimni korisnik
  + Preduvjeti: nema
  + Pokretač: Pritisak na gumb registracija
  + Opis osnovnog tijeka:
    - Korisnik odabire da je tvrtka
    - Korisnik upisuje podatke tvrtke
    - Baza podataka provjerava ispravnost podataka te postoji li već takva registrirana tvrtka
    - Baza podataka uspješno sprema podatke
  + Opis mogućih odstupanja:
    - Baza podataka ne sprema podatke
* Rbr.: UC2
  + Naziv: Registracija klijenta
  + Opis: Registracija klijenta na web aplikaciju
  + Glavni aktor: Anonimni korisnik
  + Preduvjeti: Klijent je punoljetan
  + Pokretač: Pritisak na gumb registracija
  + Opis osnovnog tijeka:
    - Korisnik odabire da je klijent
    - Korisnik upisuje podatke tvrtke
    - Baza podataka provjerava ispravnost podataka te postoji li već takav registrirani korisnik
    - Baza podataka uspješno sprema podatke
  + Opis mogućih odstupanja:
    - Baza podataka ne sprema podatke
* Rbr.: UC3
  + Naziv: Prijava
  + Opis: Prijava korisnika na web aplikaciju
  + Glavni aktor: Anonimni korisnik
  + Preduvjeti: nema
  + Pokretač: Pritisak na gumb prijava
  + Opis osnovnog tijeka:
    - Korisnik upisuje svoje podatke
    - Sustav provjerava ispravnost podataka te proslijeđuje klijenta na stranicu za prijavljene klijente
    - Klijenti su: Korisnik, Tvrtka i Administrator
  + Opis mogućih odstupanja:
    - Sustav javlja grešku zbog neispravnosti podataka
* Rbr.: UC4
  + Naziv: Jednokratna rezervacija
  + Opis: Jednokratna rezervacija parkirnog mjesta
  + Glavni aktor: Korisnik
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Odabir parkirališta i klik na gumb rezervacija
  + Opis osnovnog tijeka:
    - Korisnik odabire vrijeme početka i trajanje rezervacije
    - Sustav provjerava dostupnost parkirnog mjesta u tom vremenskom periodu i rezervira mjesto
  + Opis mogućih odstupanja:
    - Sustav provjerava dostupnost i javlja grešku
* Rbr.: UC5
  + Naziv: Ponavljajuća rezervacija
  + Opis: Ponavljajuća rezervacija parkirnog mjesta
  + Glavni aktor: Korisnik
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Odabir parkirališta i klik na gumb rezervacija
  + Opis osnovnog tijeka:
    - Korisnik odabire vrijeme početka, trajanje i dane u tjednu u kojima se rezervacija ponavlja sljedećih mjesec dana
    - Sustav provjerava dostupnost odabranog mjesta za sve željene vremenske periode, obavlja plaćanje i rezervira mjesto
  + Opis mogućih odstupanja:
    - Sustav provjerava dostupnost i javlja grešku
* Rbr.: UC6
  + Naziv: Trajna rezervacija
  + Opis: Trajna rezervacija parkirnog mjesta
  + Glavni aktor: Korisnik
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Odabir parkirališta i klik na gumb rezervacija
  + Opis osnovnog tijeka:
    - Korisnik potvrđuje da mjesto želi trajno rezervirati
    - Sustav provjerava dostupnost, obavlja plaćanje i rezervira mjesto na 30 dana, te automatski ponavlja proces svakih 30 dana sve dok korisnik ne poništi trajnu rezervaciju
  + Opis mogućih odstupanja:
    - Sustav provjerava dostupnost i javlja grešku
* Rbr.: UC7
  + Naziv: Uređivanje podataka o automobilima
  + Opis: Korisnik dodaje ili briše svoje automobile
  + Glavni aktor: Korisnik
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Pritisak na gumb *Upravljanje automobilima*
  + Opis osnovnog tijeka:
    - Korisnik klikne na gumb *Dodaj* te unese podatke o novom automobilu
    - Sustav provjerava unosi li korisnik ispravnost podataka o automobilu i postoji li već korisnik koji posjeduje automobil kakav se pokušava dodati
    - Korisnik klikne na gumb za brisanje postojećeg automobila
  + Opis mogućih odstupanja:
    - Sustav provjerava ispravnost i javlja grešku
* Rbr.: UC8
  + Naziv: Traženje najbližeg parkinga
  + Opis: Na temelju trenutne lokacije korisnika aplikacija određuje najbliže dostupno parkiralište
  + Glavni aktor: (Anonimni) Korisnik
  + Preduvjeti: nema
  + Pokretač: Pritisak na gumb *ParkirajMe*
  + Opis osnovnog tijeka:
    - Aplikacija gleda najbliže parkiralište po udaljenosti te gleda broj dostupnih mjesta na njemu
    - Provjera broja slobodnih parkirališnih mjesta
    - Ako aplikacija procjeni da na parkiralištu ima dovoljan broj slobodnih mjesta, nudi korisniku to parkiralište
  + Opis mogućih odstupanja:
    - Nedovoljno slobodnih mjesta na najbližem parkiralištu pa aplikacija nudi prvo sljedeće parkiralište po blizini sa dovoljno slobodnih mjesta
* Rbr.: UC9
  + Naziv: Uređivanje parkinga
  + Opis: Tvrtka uređuje podatke o postojećem parkingu
  + Glavni aktor: Tvrtka
  + Preduvjeti: Tvrtka je prijavljena
  + Opis osnovnog tijeka:
    - Tvrtka odabire gumb za upravljanje parkirališta te zatim gumb *Uređivanje parkinga*
    - Aplikacija izlistava tvrtki sva njena parkirališta
    - Tvrtka odabire neko parkiralište
    - Uređuje podatke o parkiralištu i potvrđuje promjene
  + Opis mogućih odstupanja:
    - Nema
* Rbr.: UC10
  + Naziv: Pregled svih parkirališta
  + Opis: Korisnik na karti može pregledavati sva dostupna parkirališta i njihove informacije
  + Glavni aktor: Korisnik
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Pritisak na gumb *Karta*
  + Opis osnovnog tijeka:
    - Korisnik može kliziti po karti i vidjeti sva raspoloživa parkirališta
    - Klikom na neko parkiralište postaju dostupne informacije o tom parkiralištu te se otvara mogućnost za rezervaciju na tom parkiralištu
  + Opis mogućih odstupanja:
    - Nema
* Rbr.: UC11
  + Naziv: Brisanje računa
  + Opis: Trajno brisanje korisničkog računa
  + Glavni aktor: Administrator
  + Preduvjeti: Administrator je prijavljen u sustav
  + Pokretač: Odabir računa i klik na gumb *Briši*
  + Opis osnovnog tijeka:
    - Aplikacija može izlistati sve tvrtke ili korisnike
    - Administrator odabire račun i briše ga
    - Ako briše tvrtku, brišu se sva njena parkirališta i sve rezervacije na njenim parkiralištima uz samo brisanje računa tvrtke
    - Ako se briše korisnik, brišu se sve njegove rezervacije uz samo brisanje računa korisnika
  + Opis mogućih odstupanja:
    - Nema
* Rbr.: UC12
  + Naziv: Poništavanje trajne rezervacije
  + Opis: Korisnik poništava trajnu rezervaciju mjesta na parkiralištu
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Pritisak na gumb *Poništi rezervaciju*
  + Opis osnovnog tijeka:
    - Korisnik na popisu svojih trajnih rezervacija odabire jednu i poništava ju
  + Opis mogućih odstupanja
    - Korisnik nema nijednu trajnu rezervaciju
* Rbr.: UC13
  + Naziv: Dodaj parking
  + Opis: Tvrtka dodaje novi parking
  + Glavni aktor: Tvrtka
  + Preduvjeti: Tvrtka je prijavljena
  + Opis osnovnog tijeka:
    - Tvrtka odabire gumb za upravljanje parkirališta te zatim gumb *Dodaj parking*
    - Tvrtka upisuje podatke novog parkirališta i potvrđuje dodavanje
  + Opis mogućih odstupanja:
    - Nema

## 4.2 Dijagrami obrazaca uporabe

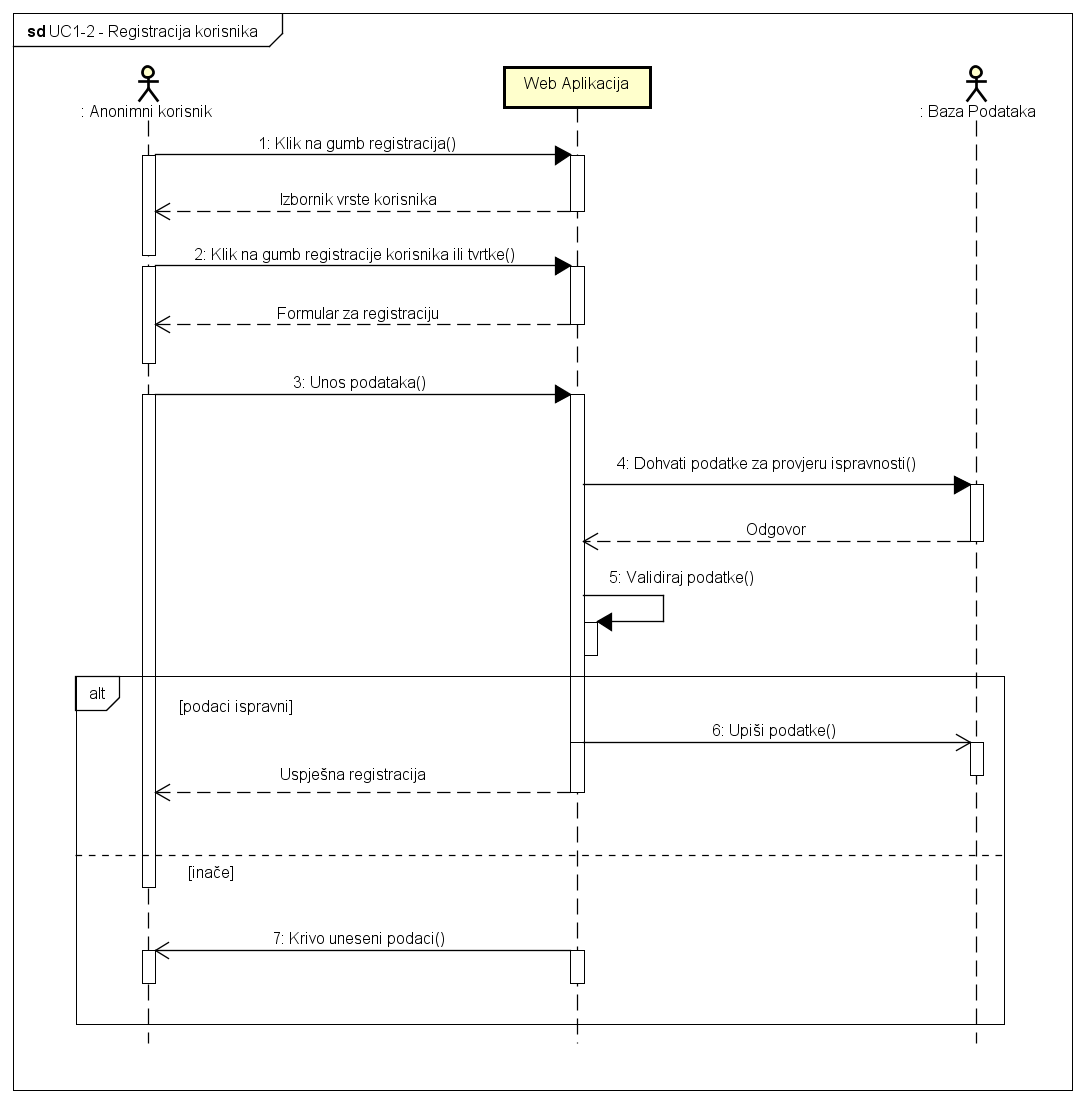


*Slika 4.2.1 Dijagram svih obrazaca uporabe*

## 4.3 Sekvencijski dijagrami

**Obrasci uporabe UC1 i UC2:**

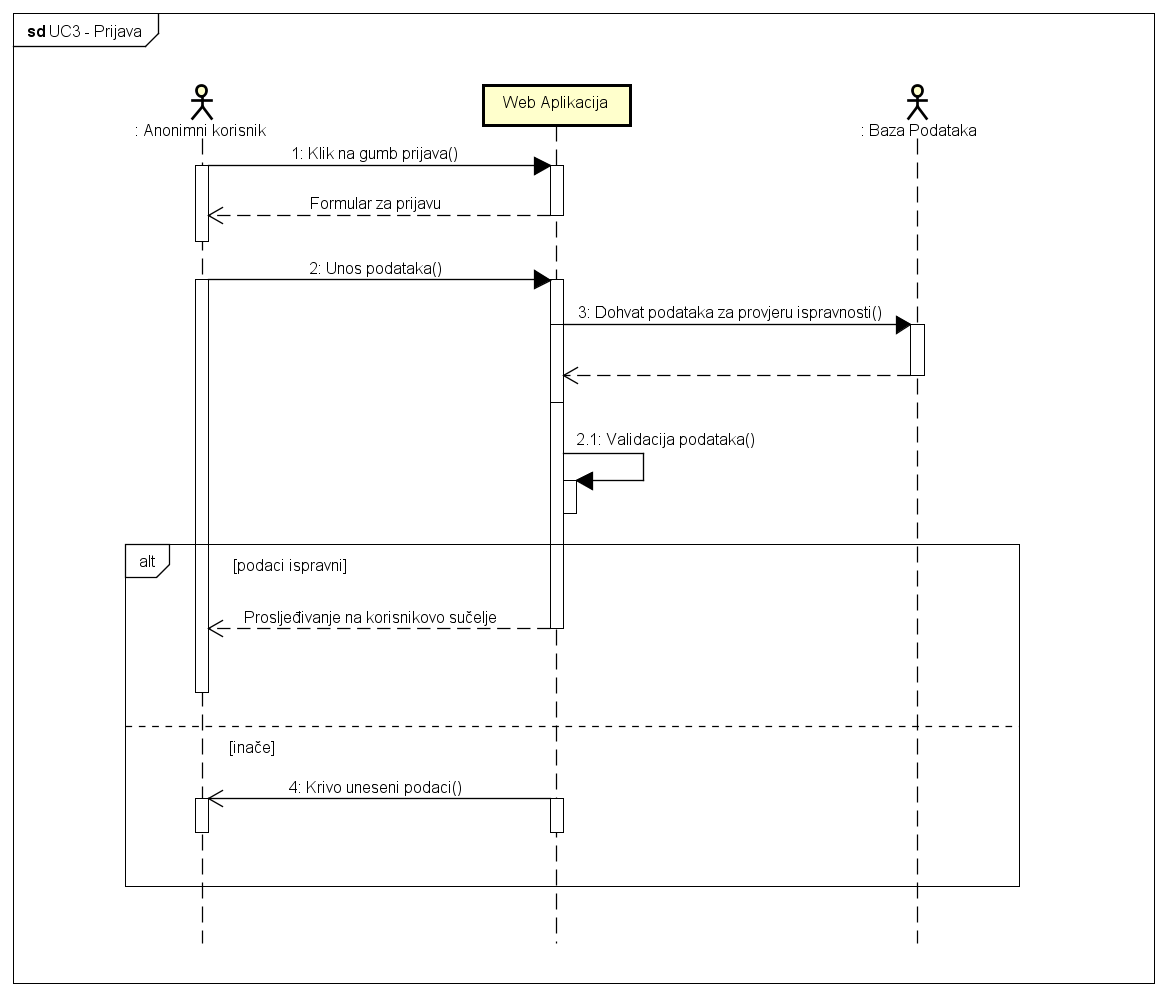
Anonimni korisnik klikne na gumb 'registracija' te mu se nudi opcija da se registrira kao korisnik ili tvrtka. Korisnik zatim ispunjava obrazac s podacima nakon čega sustav provjerava ispravnost tih podataka odnosno postoji li već takav registrirani korisnik. Ako sustav utvrdi da su podaci ispravni korisnik dobiva poruku o uspješnoj registraciji. Inače, korisnik dobiva poruku o neispravnosti podataka.



*Slika 4.3.1 Sekvencijski dijagram registracije korisnika – UC1, UC2*

**Obrazac uporabe UC3:**

Anonimni korisnik klikne na gumb 'prijava' te zatim ispunjava obrazac s podacima nakon čega sustav provjerava ispravnost tih podataka. Ako sustav utvrdi da su podaci ispravni korisnik se prosljeđuje njegovo sučelje ovisno o vrsti korisnika: korisnik, tvrtka ili administrator. Inače, korisnik dobiva poruku o neispravnosti podataka.



*Slika 4.3.2 Sekvencijski dijagram za prijavu korisnika – UC3*

**Obrasci uporabe UC4, UC5 i UC6:**

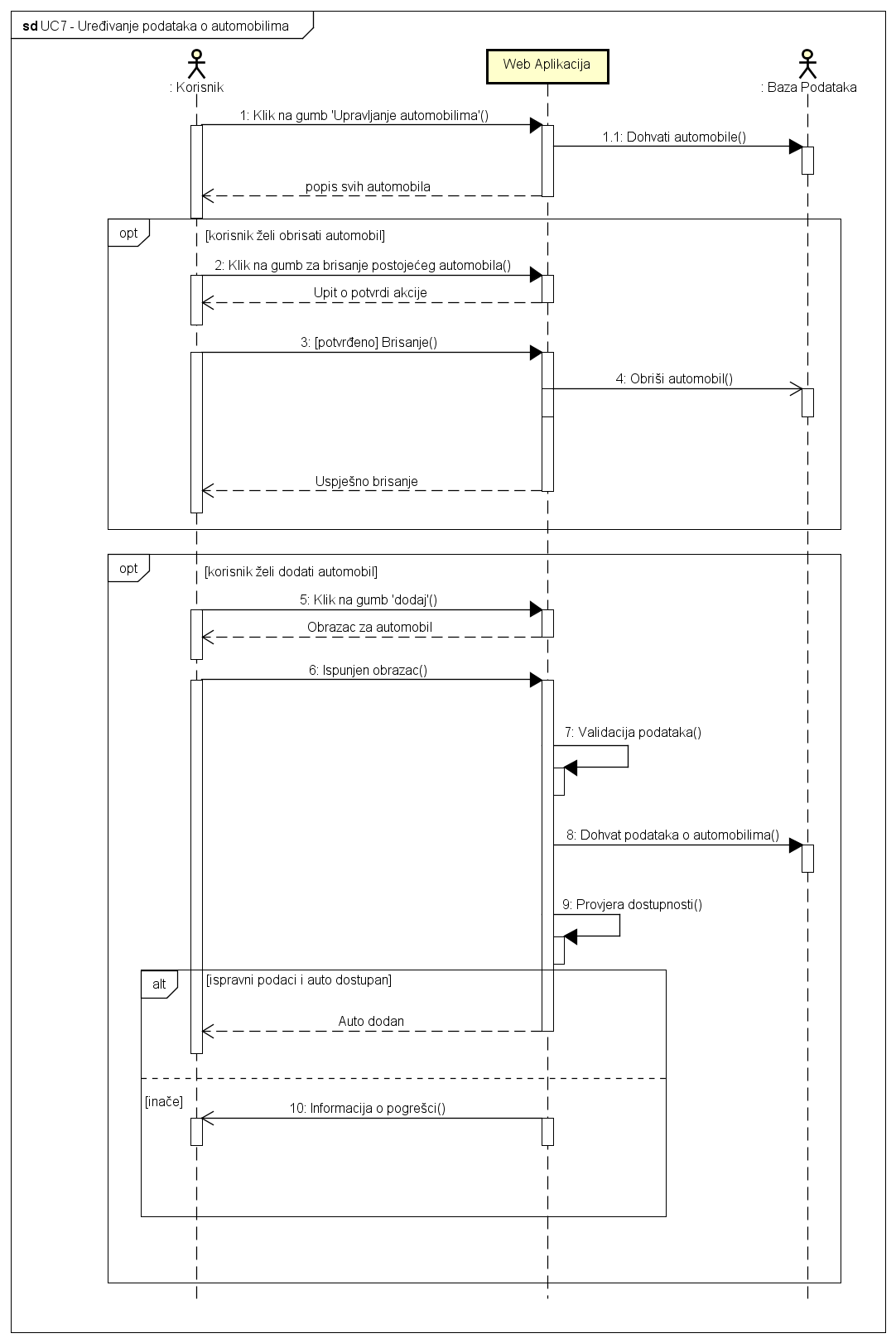
Korisnik odabire jedno od dostupnih parkirališta na karti te klikne na gumb 'rezervacija' te zatim bira vrstu rezervacije. Nakon odabira jedne od dostupnih vrsta rezervacija (jednokratna, ponavljajuća i trajna), ispunjava obrazac za rezervaciju nakon čega sustav validira ispunjeni obrazac te provjerava dostupnost rezervacije. Ako sustav utvrdi da je rezervacija dostupna, obavlja plaćanje te šalje korisniku poruku o uspješnoj rezervaciji. Inače, korisnik dobiva poruku o nedostupnosti rezervacije ili obavijest o krivo unesenim podacima.



*Slika 4.3.3 Sekvencijski dijagram za rezervacije – UC4-6*

**Obrasci uporabe UC7:**

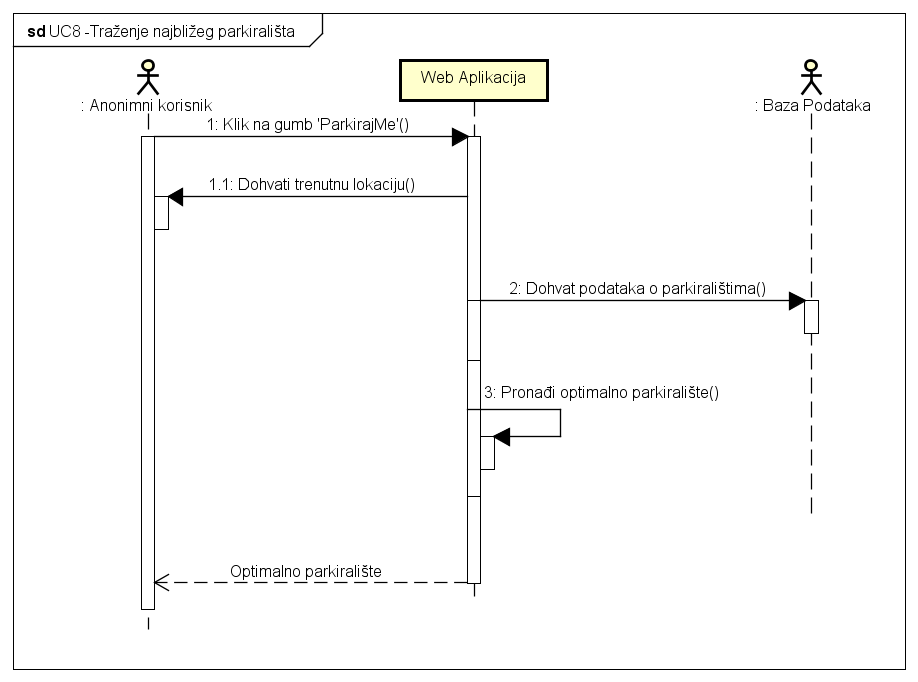
Korisnik klikne na gumb 'Upravljanje automobilima' te se prosljeđuje na popis svih svojih automobila. Korisnik zatim može dodati novi automobil ili obrisati postojeći. Klikom na gumb za brisanje postojećeg automobila od korisnika se traži potvrda za traženu akciju. Ako korisnik klikne potvrdno automobil se briše iz njegovog popisa. Dodatno, korisnik može kliknuti na gumb za dodavanje novog automobila nakon čega ispunjava obrazac. Ako je unesao ispravne podatke i nijedan drugi korisnik ne posjeduje taj automobil korisnik dobiva obavijest o uspješnom dodavanju automobila.



*Slika 4.3.4 Sekvencijski dijagram za upravljanje automobilima – UC7*

**Obrasci uporabe UC8:**

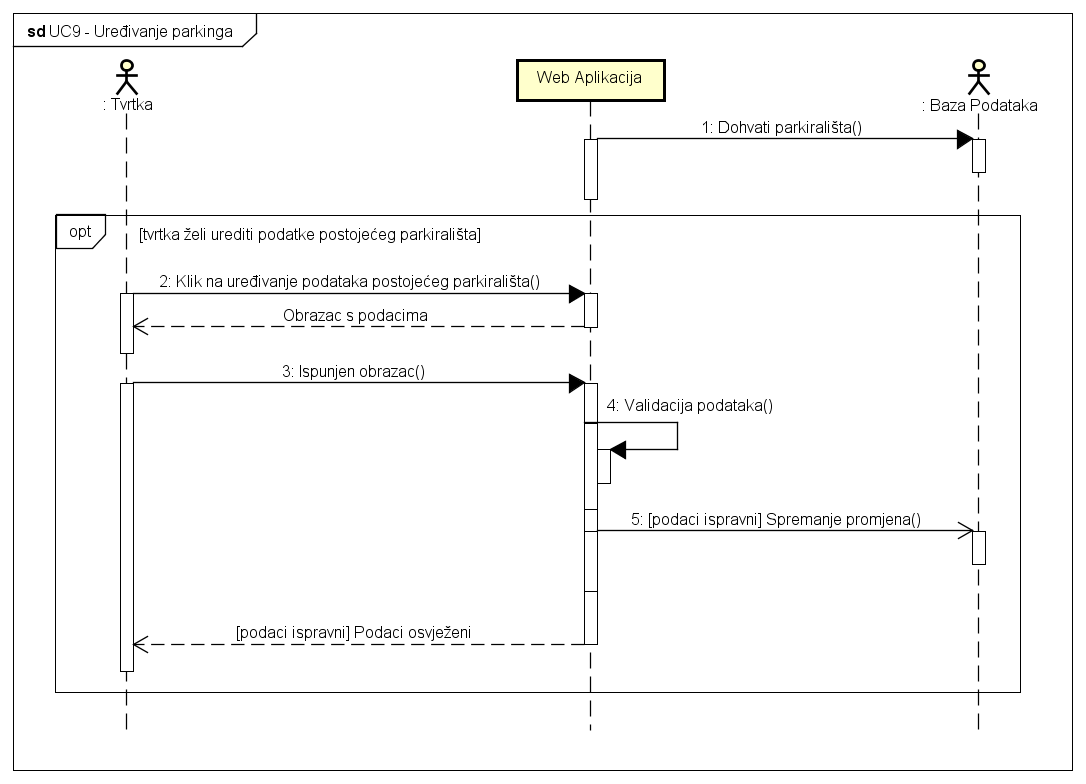
Korisnik klikne na gumb 'ParkirajMe'. Dohvate se podaci o svim parkiralištima nakon čega se određuje optimalno parkiralište na osnovu trenutne lokacije korisnika i popunjenosti najbližih parkirališta. Korisnika se prosljeđuje na određeno parkiralište.



*Slika 4.3.5 Sekvencijski dijagram za traženje parkirališta – UC8*

**Obrasci uporabe UC9:**

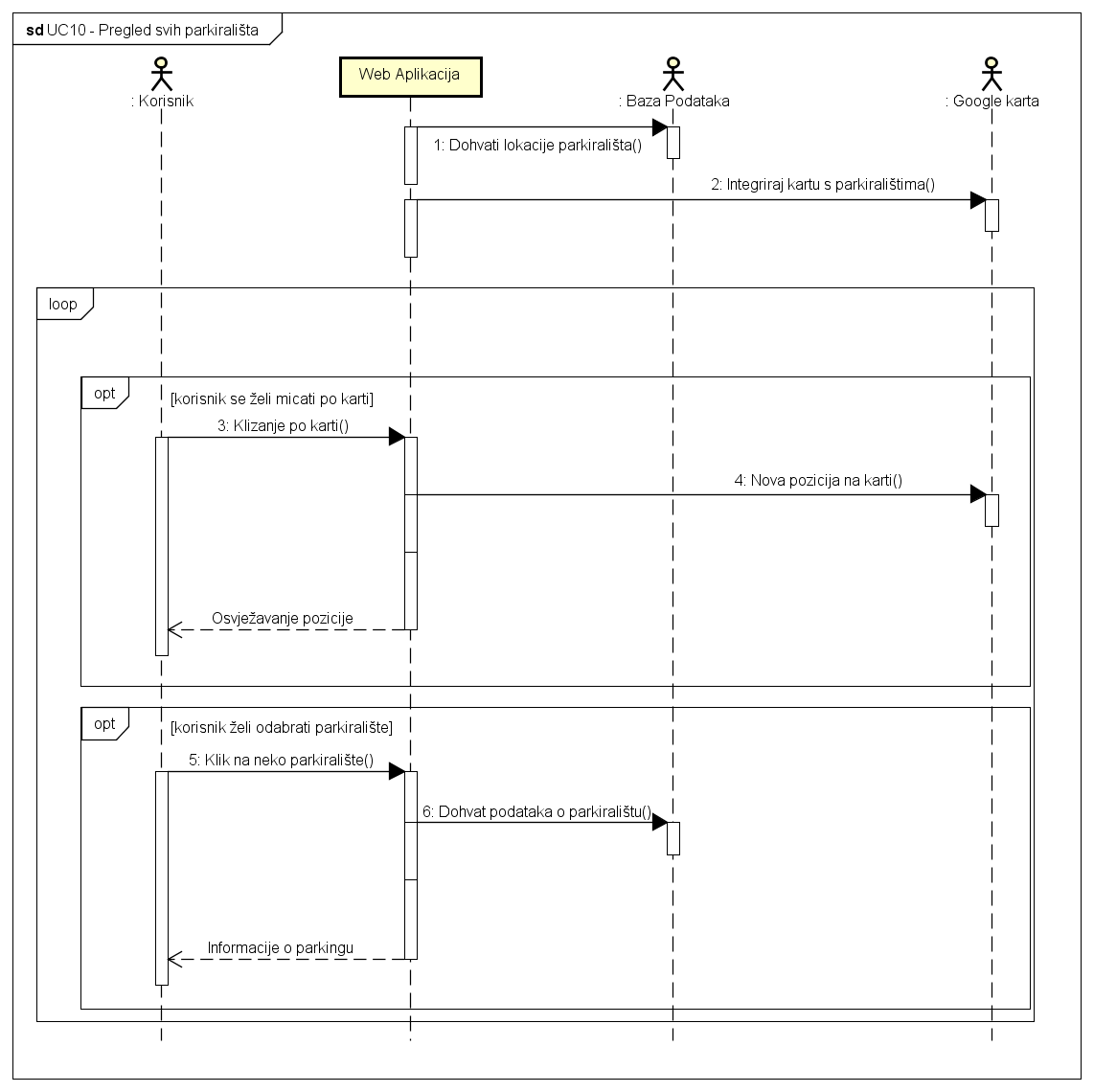
Tvrtka može urediti podatke o postojećem parkiralištu. Klikom na upravljanje parkiralištima te zatim klikom na gumb uređivanje parkinga tvrtki se izlistavaju sva parkirališta. Tvrtka odabire parkiralište čije podatke želi izmijeniti, upise nove podatke te ako su podatci ispravni, novo parkiralište se dodaje u bazu podataka.



*Slika 4.3.6 Sekvencijski dijagram za uređivanje parkinga – UC9*

**Obrasci uporabe UC10:**

Nakon što se karta s lokacijama parkirališta učita korisnik može kliziti po karti kako bi ju lakše mogao pregledavati te kliknuti na pojedino parkiralište kako bi mogao vidjeti dodatne informacije ili napraviti rezervaciju.



*Slika 4.3.7 Sekvencijski dijagram za pregled parkirališta – UC10*

**Obrasci uporabe UC11:**

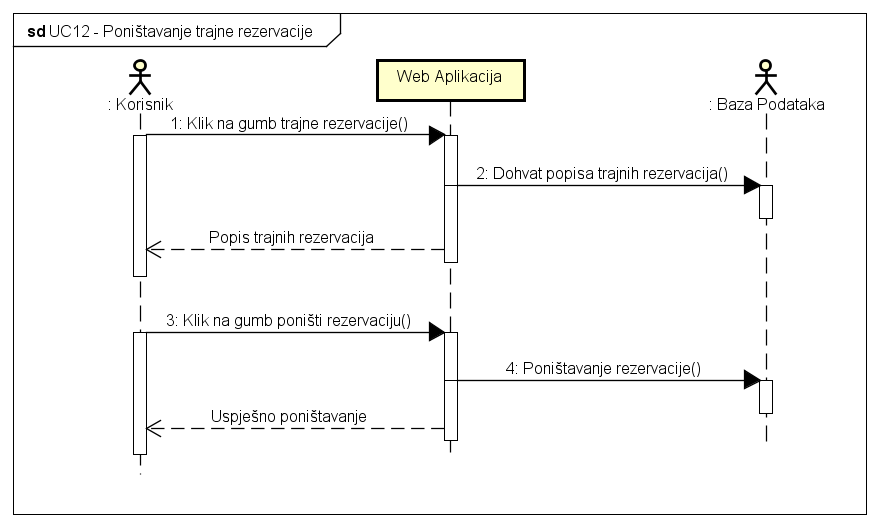
Administrator može pregledati listu tvrtki ili korisnika. Klikom na brisanje računa se iz baze podataka brišu rezervacije vezane uz korisnika ili uz sva parkirališta tvrtke, potom se brišu sva parkirališta ako je odabrana tvrtke te na posljetku brisanje podataka samog računa.



*Slika 4.3.8 Sekvencijski dijagram za brisanje računa – UC11*

**Obrasci uporabe UC12:**

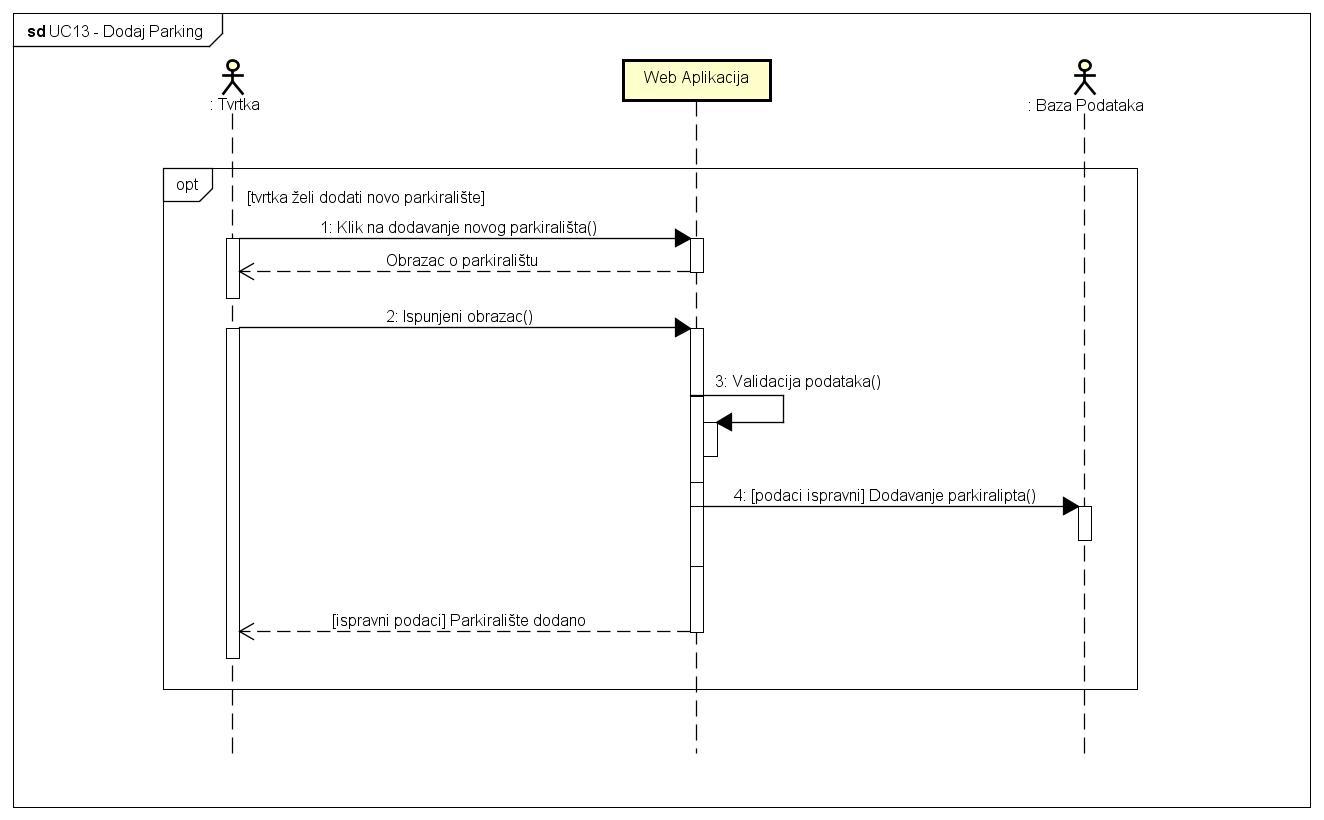
Korisnik može pregledati popis svojih trajnih rezervacija te ih poništiti, tako da se prestanu automatski obnavljati svakih 30 dana.



*Slika 4.3.9 Sekvencijski dijagram za poništavanje trajne rezervacije – UC12*

**Obrasci uporabe UC13:**

Tvrtka može dodati novo parkiralište. Klikom na upravljanje parkiralištima tvrtka zatim odabire dodavanje novog parkinga. Tvrtki se potom nudi obrazac s ispunjavanjem podataka o parkiralištu. Ako su podaci ispravni, novo parkiralište se dodaje u bazu podataka.

 *Slika 4.3.9 Sekvencijski dijagram za dodavanje parkirališta– UC13*

# Ostali zahtjevi

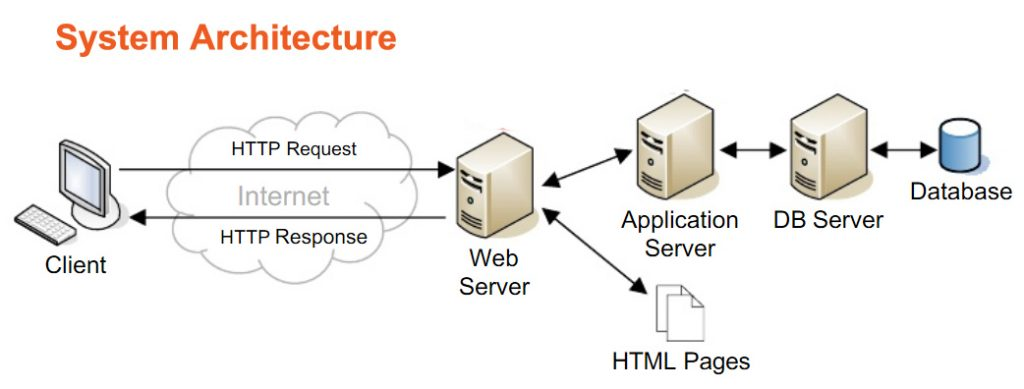
* Sustav mora jamčiti točnost informacija
* Svakih 60 sekundi treba se ažurirati stanje parkirališta
* Sustav mora biti responzivan
* Sustav mora omogućavati paralelan rad više korisnika
* Pogrešno korištenje aplikacije ne smije utjecati na stanje sustava ili baze podataka
* Sustav mora podržavati znakove hrvatske abecede
* Nadogradnja novih funkcionalnosti ne smije narušiti funkcionalnost rada sustava
* Sustav ne smije dopustiti korisnicima pristup administratorskim opcijama

# Arhitektura i dizajn sustava

## Svrha, opći prioriteti i skica sustava

Tijekom projektiranja sustava jedna od važnijih odluka je izbor arhitekture. Karakteristike dobre arhitekture su mogućnost nadogradnje, povoljno održavanje i fleksibilnost sustava. Kako nam je cilj omogućiti korisniku obavljanje potrebnih radnji preko interneta te interakciju korisnika sa sustavom učiniti što jednostavniju i manje podložnu greškama, odlučili smo se za izradu web-aplikacije.

Arhitektura sustava se može podijeliti na dva podsustava: web-aplikacija i baza podataka. Dakle, sustav će se sastojati od web aplikacije koja će biti povezana s bazom podataka. Baza podataka će spremati podatke o korisnicima, tvrtkama, parkiralištima i rezervacijama.



*Slika 6.1.1 Arhitektura sustava*

**Web aplikacija**

Kao programski jezim odabrali smo objektno orijentirani jezik Java. Specifičnije koristiti ćemo tehnologije Spring Framework i Bootstrap. Spring Framework nudi mnoge alate i olakšava pisanje backenda za web aplikacije dok nam Bootstrap olakšava pisanje frontenda. Povezivanje frontenda i backenda će se odvijati preko RESTful API-ja, a upravljač baze podataka za backend će biti H2.

**Baza podataka**

U bazu podataka pohranjivat će se svi podatci relevantni za prijavu korisnika i uporabu same aplikacije. Prlikom kreiranja baze podataka isključit čemo mogućnost redundancije podataka tako što ćemo sve entitete i relacije svesti na treću normalnu formu. Za prikaz strukture baze podataka kreiran je ER model baze podataka.



*Slika 6.1.2 ER model baze podataka*

## Dijagram razreda s opisom

Dijagram razreda prikazuje sve postojeće razrede koje koristimo u implementaciji, zajedno s njihovim odnosima. Pretpostavlja se da su svi atributi privatni, i da za svaki atribut postoje pripadajuće metode *get* i *set*. (Slika 6.2.1)

**Administrator -** modelira administratora

Atributi:

* adminID – jedinstveni identifikator administratora
* sifra – hash kombinacije lozinke koju administrator unosi tijekom prijave i *salta*
* salt – kriptografski *nonce* koji se koristi tijekom verifikacije šifre administratora

Metode:

* obrisiKorisnika – omogućava brisanje korisničkog računa, a ako je korisnik tvrtka, onda također briše sva pripadajuća parkirališta

**Anonimni korisnik –** modelira anonimnog korisnika

Metode:

* prijava – omogućuje prijavu registriranim korisnicima
* registrirajKorisnika – omogućuje registraciju računa običnog korisnika
* registrirajTvrtku – omogućuje registraciju računa tvrtke

**Registrirani korisnik –** modelira registriranog korisnika

Atributi:

* ID – jedinstveni identifikator korisnika
* sifra – hash kombinacije lozinke koju korisnik unosi tijekom prijave i *salta*
* ime – ime korisnika
* prezime – prezime korisnika
* brojKreditneKartice – broj kreditne kartice
* OIB – osobni identifikacijski broj korisnika
* salt – kriptografski *nonce* generiran tijekom registracije korisnika koji se koristi u verifikaciji lozinke
* email – adresa e-pošte korisnika

Metode:

* dohvatiParkiralista – dohvaća popis parkirališta, sortirano uzlazno po udaljenosti
* dohvatiNajblizeParkiraliste – vraća parkiralište najbliže zadanoj lokaciji
* rezervirajJednokratno – generira jednokratnu rezervaciju
* rezervirajPonavljajuce – generira ponavljajuću rezervaciju
* rezervirajTrajno – generira trajnu rezervaciju
* urediPodatke – omogućava uređivanje korisničkih podataka
* ponistiRezervaciju – omogućava brisanje postojeće rezervacije
* obrisiRacun – omogućava brisanje korisničkog računa

**KorisnikTvrtka –** modelira tvrtku

Atributi:

tvrtkaID - jedinstveni identifikator tvrtke

* imeTvrtke – ime tvrtke
* adresaSjedista – adresa sjedišta
* OIB - OIB tvrtke
* parkiralista – lista registriranih parkirališta koja pripadaju tvrtki
* sifra – hash kombinacije lozinke koju korisnik (npr. tvrtkin web administrator) unosi tijekom prijave i *salta*
* salt – kriptografski *nonce* generiran tijekom registracije tvrtke koji se koristi u verifikaciji lozinke

Metode:

* urediParkiraliste – omogućuje uređvanje podataka registriranog parkirališta
* dodajParkiraliste – omogućuje registraciju parkirališta
* obrisiParkiraliste – omogućuje brisanje registriranog parkirališta

**Automobil –** modelira automobil registriranog korisnika

Atributi:

* registracija – registracija automobila, ujedno i jedinstveni identifikator
* OIBVlasnika – osobni identifikacijski broj vlasnika
* model – model automobila

**Parkiralište –** modelira parkiralište

Atributi:

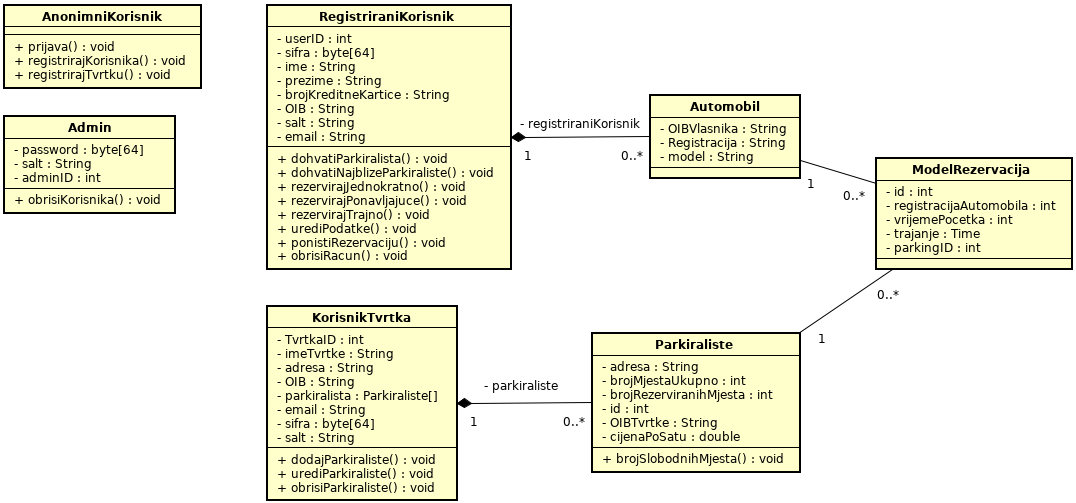
* adresa – adresa parkirališta
* brojMjestaUkupno – kapacitet parkirališta
* brojRezerviranihMjesta – broj mjesta koja su trenutno rezervirana
* ID – jedinstveni identifikator parkirališta
* OIBTvrtke – osobni identifikacijski broj tvrtke kojoj parkiralište pripada
* cijenaPoSatu – cijena parkiranja po satu

Metode:

* dohvatiSlobodnaMjesta – dohvaća mjesta koja trenutno nisu rezervirana
* brojSlobodnihMjesta – dohvaća broj slobodnih mjesta

**ModelRezervacija –** modelira postojeću rezervaciju

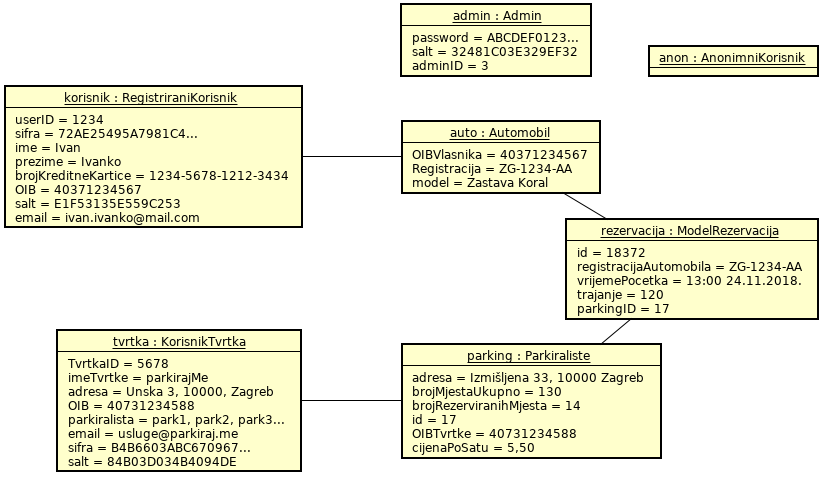
Atributi:

* ID – jedinstveni identifikator rezervacije
* registracijaAutomobila – registracija automobila
* vrijemePocetka – vrijeme pocetka rezervacije
* trajanje – trajanje rezervacijeu minutama
* parkingID – identifikator parkirališta u kojem se rezervira
* 

*Slika 6.2.1 Dijagram razreda*

## Dijagram objekata

Dijagramom objekata (Slika 6.3.1) opisujemo stanje objekata u aplikaciji nakon što je korisnik uspješno rezervirao parkirno mjesto.



*Slika 6.3.1 Dijagram objekata*

## Ostali UML dijagrami

**Komunikacijski dijagram**

Komunikacijski dijagram je dijagram međudjelovanja. Pripada nadskupini ponašajnih dijagrama, uz dijagrame stanja, aktivnosti i obrazaca uporabe. Komunikacijski dijagram stavlja veći naglasak na pregled scenarija, na međusobnu komunikaciju. Sažeti su te je redoslijed poruka sudionika određen brojem na poruci.

Komunikacijski dijagram na slici prikazuje jednokratnu rezervaciju koja uključuje prijavu u sustav. Pretpostavljam da je korisnik već registriran, tako da ću izostaviti registraciju korisnika. Korisnik se prijavljuje tako da šalje zahtjev za prijavom i svojim podatcima kao parametrima. Sustav provjerava u bazi podataka ispravnost te ako su podatci ispravni, prijavljuje korisnika. Prijavljeni korisnik može slati zahtjev za rezervacijom u kom navodi vrijeme početka i trajanje rezervacije. Sustav provjerava dostupnost parkirnog mjesta u bazi podataka. Ako je mjesto dostupno, rezervira ga i šalje korisniku informaciju o uspješnoj rezervaciji.

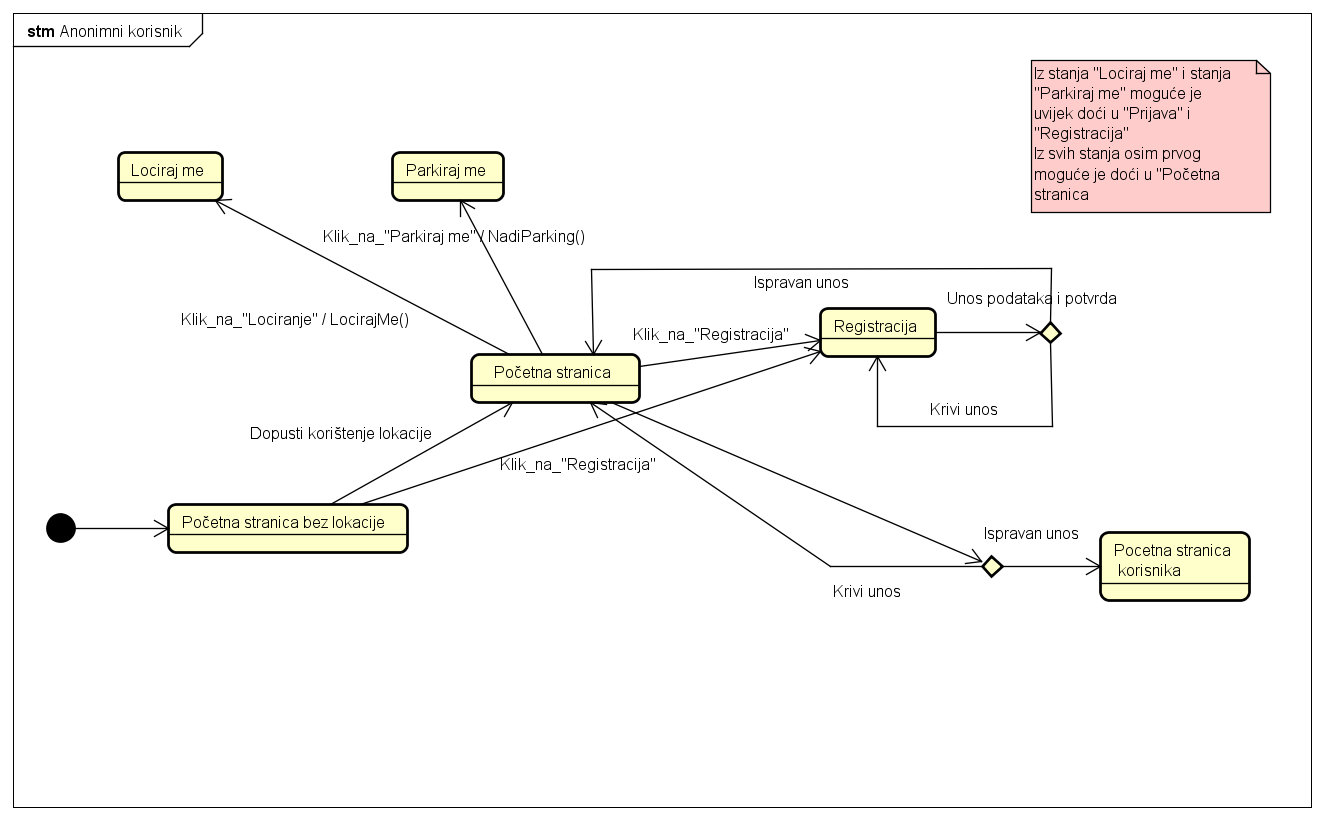


*Slika 6.4.1 Komunikacijski dijagram*

**Dijagram stanja**

Dijagramom stanja modeliramo ponašanje sustava koje je upravljano događajima. On opisuje diskretna stanja sustava i prijelaze između tih stanja.

Kada neka osoba pristupi web aplikaciji tada je ona anonimni korisnik. Prilikom pristupanja stranici nudi se dopuštenje za korištenje lokacije. Ukoliko korisnik odabere da ne dopušta, tada je njoj na raspolaganju dostupna samo prijava i registracija. Za korisnika koji dopusti korištenje lokacije osim prethodnih opcija dostupne su još i opcije poput lociranja na karti te traženja najbližeg parkinga.



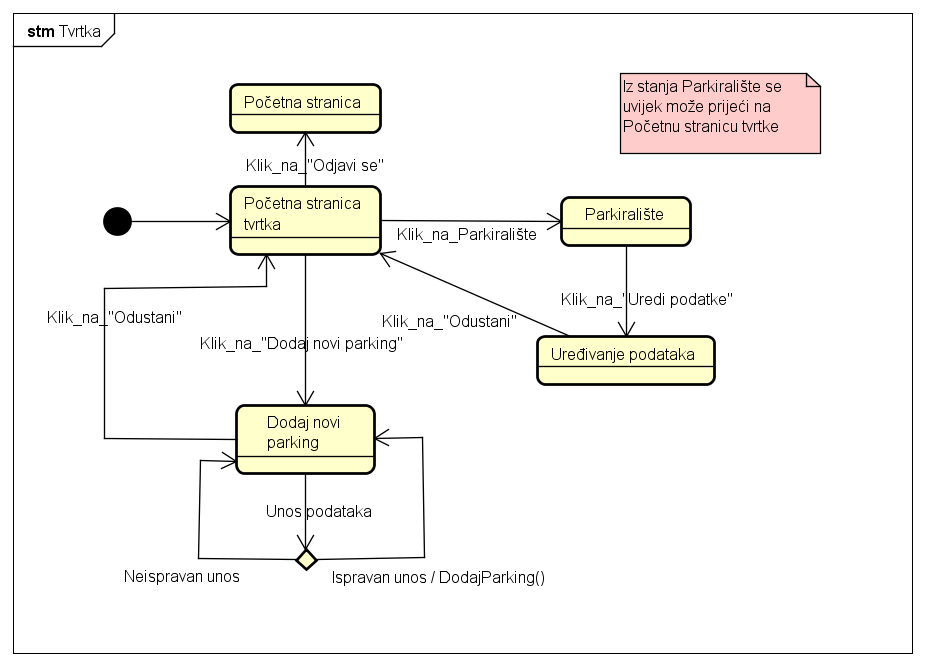
*Slika 6.5.1 Dijagram stanja – anonimni korisnik*

Nakon unosa ispravnih podataka i odabira prijave korisnika se prebacuje na početnu stranicu za korisnika koja nudi dodatne opcije. Korisnik može odabrati neki parking sa slike te ga rezervirati trajno, jednokratno ili ponavljajuće. U svakom trenutku moguće je odustati i vratiti se na početnu stranicu. Isto tako moguće je dodavati nove automobile kao i njihovo brisanje. Sa početne stranice dostupna je i lista rezervacija ali samo trajnih zbog mogućnosti ukidanja.



*Slika 6.5.2 Dijagram stanja – Korisnik*

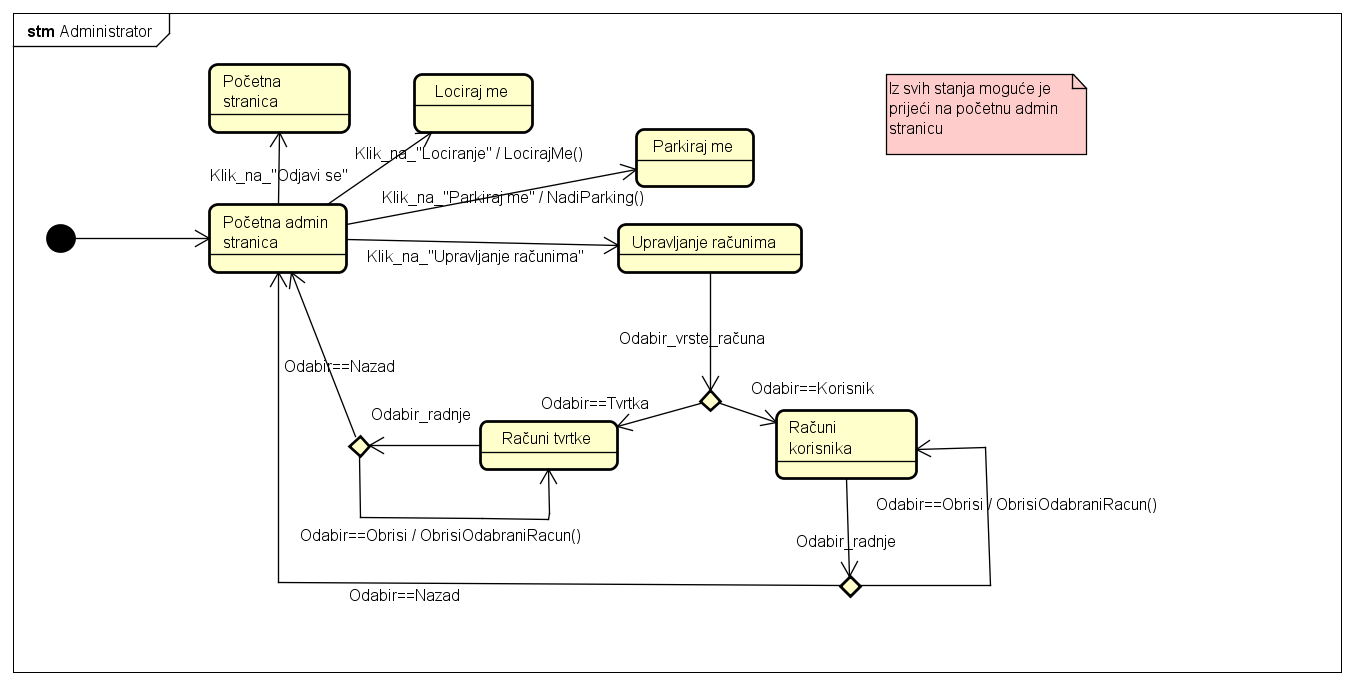
Razlika je prijavljujemo li se sa računom tvrtke ili korisnika. Ukoliko smo korisnik možemo koristi opcije navedene na slici iznad, no ukoliko smo tvrtka imamo neke sasvim druge opcije. Moguće je odabirati parkirališta te ukoliko su ona naša moguće ih je urediti ili npr. dodati novi parking. Ukoliko se dogodi neispravan unos sustav nam daje da ponovno upisujemo.



*Slika 6.5.3 Dijagram stanja – Tvrtka*

Jedini način za brisanje računa je preko administratora. Administrator se prijavljuje sa posebnom korisničkom oznakom i lozinkom. On ima mogućnost brisanja računa korisnika i tvrtka.

Tvrtke, korisnici i administratori imaju mogućnost odjavljivanja i time vraćanja na početnu stranicu kakvu ima anonimni korisnik.



*Slika 6.5.3 Dijagram stanja – Administrator*

**Dijagram aktivnosti**

Dijagram aktivnosti, uz dijagram stanja, prikazuje softversku funkcionalnost iz perspektive sustava, ne prikazujući aktore ili vanjsko korisničko sučelje. Prikazuje korak po korak radni tok aktivnosti obavljanih u sustavu, stavljajući naglasak na jednostavnost i slijednost operacija.

Dijagram aktivnosti prikazan na slici predstavlja postupak registracije korisnika. Korisnik početno odabire polje Registracija, unosi svoje podatke u polja za registraciju, koje sustav zatim provjerava slanjem upita bazi podataka. Baza provjerava podatke, a ovisno o ispravnosti podataka, proces može ići u dva smjera. Ako su podatci neispravni, sustav šalje obavijest korisniku i zahtjeva ponovan unos. Ako su ispravni, sustav šalje zahtjev za pohranom podataka bazi. U slučaju da je pohrana neuspješna, sustav će ponoviti zahtjev. Ako je pohrana podataka korisnika uspješna, a time i registracija, obavještava o uspješnoj registraciji.

A close up of a map

Description automatically generated

*Slika 6.4.6 Dijagram aktivnosti – Registracija korisnika*

Sljedeći dijagram aktivnosti prikazuje postupak uređivanja parkinga. Uređivati podatke o parkiralištima mogu tvrtke koje su prijavljene te mogu uređivati podatke onih parkirališta čiji su vlasnici. Tvrtka prilikom prijave vidi svoja parkirališta, odabere jedno za koje sustav vraća podatke i klikne na gumb Uredi podatke. Nakon uređivanja podataka, tvrtka potvrđuje promjene klikom na gumb Spremi koje, ako su podatci ispravni, se upisuju u bazu. Ako tvrtka ipak ne želi spremiti promjene, odabire gumb Odustani. Ovisno o uspješnosti pohrane, dobiva se obavjest o uspješnoj izmjeni.

A close up of a map

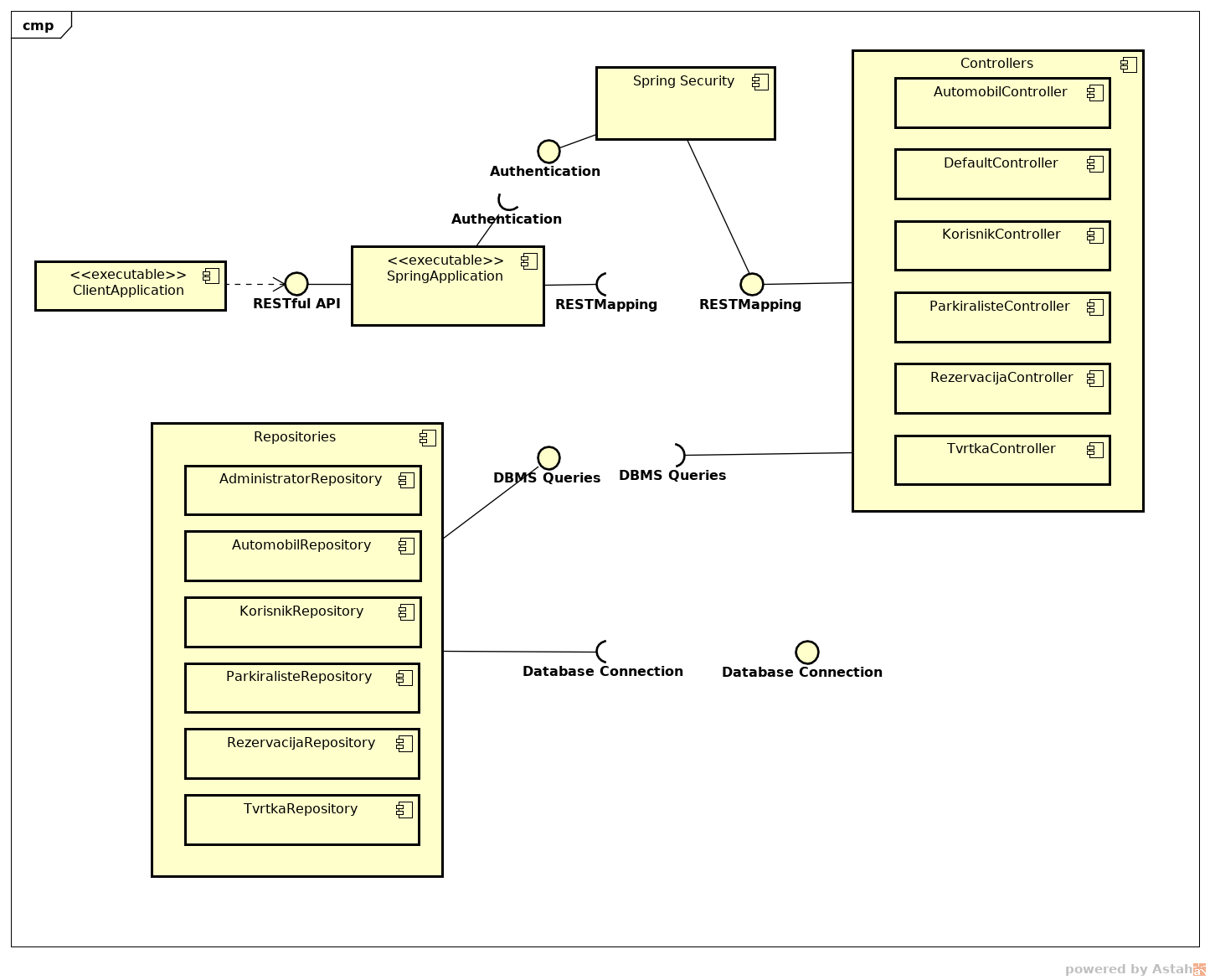
Description automatically generated

*Slika 6.4.7 Dijagram aktivnosti – Uređivanje parkirališta*

**Dijagram komponenti**

Konačno, slika 6.6.1 prikazuje dijagram komponenti. Same aplikacije poslužitelja i korisnika komuniciraju koristeći *RESTful API*. Budući da je za većinu obrazaca uporabe potrebna registracija korisnika ili tvrtke, koristimo razvojni okvir *Spring Security* u sklopu razvojnog okvira *Spring* koji omogućuje autentifikaciju korisnika. Samo mapiranje REST zahtjeva odvija se preko kontrolera, a u zahtjevima za koje je potrebna autentifikacija sudjeluje i *Spring Security*. Repozitoriji služe kao posrednici između baze podataka, s kojom su spojeni, i već navedenih kontrolera.

//TODO: je li potrebno dodati i sve razrede koji implementiraju konkretne operacije tipa svi DTO, svi JPA itd., također numerirati sliku



# Implementacija i korisničko sučelje

*U ovom poglavlju potrebno je:*

*dati dijagram razmještaja (deployment dijagram)*

*navesti koje su tehnologije i alati korišteni u razvoju sustava*

*dati isječak programskog koda koji implementira neku od temeljnih funkcionalnosti u sustavu*

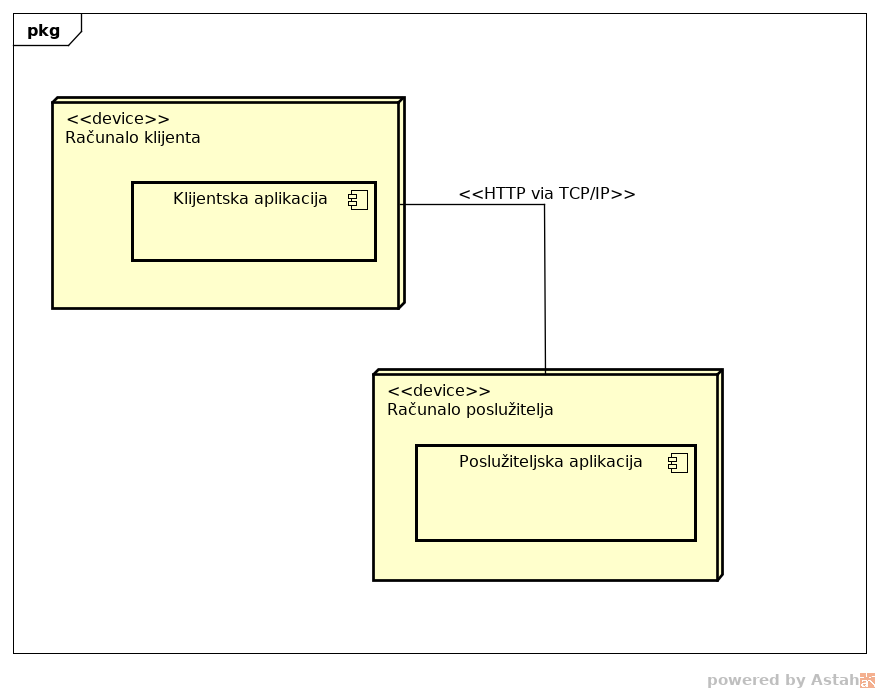
*objasniti kako je ispitano implementirano rješenje i pokazati bar 4 ispitna scenarija*

*dati upute za instalaciju*

*dati upute za korištenje*

## Dijagram razmještaja

Slika 7.1.1 prikazuje dijagram razmještaja koji opisuje fizički razmještaj sustava *ParkirajMe*. Na računalu klijenta pokreće se *web* preglednik kojim se potom pristupa poslužitelju, prilikom čega se koristi HTTP protokol preko TCP/IP protokola. Na računalu poslužitelja pokrenuta je *web* aplikacija koja uključuje samu aplikaciju (logiku), bazu podataka i sustav upravljanja bazom podataka.



## Korištene tehnologije i alati

React je *JavaScript* biblioteka za izradu korisničkog sučelja. Na održavanju sustava rade kompanija Facebook i grupa individualnih *developera* i kompanija. Može se koristiti kao osnova za razvijanje jedno strančnih ili mobilnih aplikacija. Aplikacije koje koriste React uglavnom moraju koristiti dodatne biblioteke za povezivanje sa *API*-jem. Također ima extenzije i nadogradnje poput *React Bootstrap* kojeg koristimo.

Spring Boot je konvencijsko-konfiguracijsko rješenje za kreiranje samostalnih, proizvodnih programa na temelju Spring-a koje možete lagano pokrenuti. Cilj izrade Boot-a je omogućiti programeru da krene s minimalnim naporom pošto se on brine o najboljoj konfiguraciji Spring platforme i *third-party* bibliotekama.

Leaflet je vrlo proširena *open-source JavaScript* biblioteka koja se koristi za izradu aplikacija koje koriste karte. Biblioteka podržava HTML5 i CSS3. Jedna je od najpoznatijih biblioteka pošto je pomoću nje vrlo lagano integrirati neku mapu. Koriste ju poznate stranice poput Pinterest-a i Flickr-a.

JavaScript ili JS kako ga zovemo skraćeno je visoki programski jezik koji koristi *ECMAScript* specifikaciju. Pored HTML-a i CSS-a on je jedan od 3 najbitnijih jezika za *World Wide Web*. Omogućava interaktivne web stranice što je ključ web programiranja.

JSON (JavaScript Object Notation) je *open-standard* format za podatke. Koristi tekst koji je čitljiv korisniku, npr. Objekti atribut-vrijednost i polja podataka.

REST je softverski stil arhitekture koji definira skup ograničenja koja se koriste za stvaranje web-usluga. Web stranice napravljene pomoću te arhitekture pružaju interoperabilnost između računalnih sustava na Internetu.

Fetch API omogućuje *JavaScript* sučelje za pristupanje i manipuliranje dijelova HTTP-a kao npr. upit i odgovor. API omogućuje vrlo lagan logički način za dohvaćanje resursa kroz mrežu.

Hibernate H2 je sustav za upravljanje bazom podataka napisan u Javi.

CSS (Cascading Style Sheets) je jezik koji se koristi za opisivanje prezentacije dokumenta pisanog u označni jezik kao što je HTML. Osmišljen je kako bi omogućio odvajanje prezentacije i sadržaja, uključujući izgled, boje i fontove. Odvajanje znači poboljšanje pristupaćnosti sadržaju, veću fleksibilnost i kontrolu u specifikaciji karakteristika prezentacije.

HTML (Hypertext Markup Language) je standardni označni jezik za izradu web stranica i web aplikacija. Web-preglednici primaju HTML dokumente s web-poslužitelja ili iz lokalne pohrane i prenose dokumente na multimedijalne web-stranice. HTML opisuje strukturu web stranice semantički i izvorno uključuje znakove za izgled dokumenta.

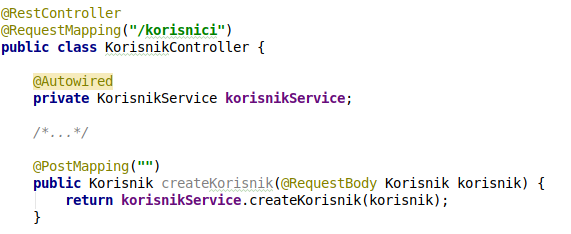
IntelliJ je integrirano razvojno okruženje za razvoj računalnih softvera pisanih u Javi.

Node.js je *open-source*, više platformno JavaScript *run-time* okruženje koje izvršava JavaScript kod izvan preglednika.

## Isječak programskog koda vezan za temeljnu funkcionalnost sustava

**Registracija korisnika**

Budući da je većina obrazaca uporabe definirana za registrirane korisnike, najprije je bilo potrebno implementirati samu registraciju. Stvaranje novog korisnika propisuje sučelje *KorisnikService* metodom *createKorisnik*. Nadalje, sam zahtjev mapira se na sufiks adrese “/korisnici”, koristeći *KorisnikController* (slika 7.3.1.).



*Slika 7.3.1. prikazuje isječak koda u razredu KorisnikController koji je odgovoran za registraciju korisnika.*

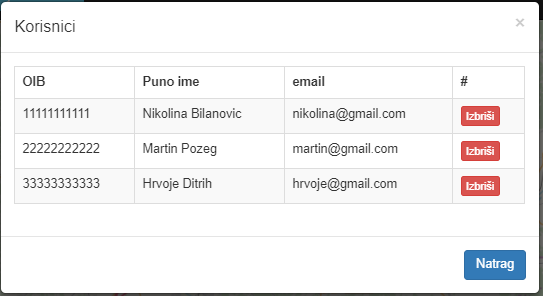
KorisnikController je jedna konkretna implementacija *Spring Controllera*, čija je glavna zadaća, okvirno govoreći, presresti dolazeće zahtjeve, iz njih izvući potrebne informacije i dalje ih proslijediti na procesiranje. Ovdje koristimo *RESTful API*, pa nad razred stavljamo anotaciju *@RestController*, čime označavamo da se radi o *RESTful* zahtjevima. Oni se svi mapiraju na sufiks adrese “/korisnici”, što je vidljivo iz anotacije *@RequestMapping(“/korisnici”)*. Dalje, budući da implementiramo ponašanje definiramo sučeljem *KorisnikService*, razred kontrolera “spajamo” sa sučeljem koristeći anotaciju *“@Autowired*” nad atributom koji implementira navedeno sučelje. Konačno, zahtjev za stvaranje korisnika se ne mapira na nikakav dodatan sufiks osim vršnog, i svi podaci koji su predani u POST *request* metodi u JSON formatu automatski se parsiraju i predaju metodi *createKorisnik*, što je označeno anotacijom *@RequestBody*.

Prije pohrane novog korisnika u bazu podataka, potrebno je provjeriti njegove podatke. Zato se oni privremeno spremaju u objekt *korisnik* razreda *Korisnik* i predaju metodi *createKorisnik* koja obavlja provjeru i pohranu (slika 7.3.2.). Ona je implementirana u razredu *KorisnikServiceJpa*, gdje JPA označava da se koristi *Spring Data JPA API* i *JPA Based Repository*. U toj se metodi prvo obavljaju provjere poput valjanosti *e-mail* adrese, jedinstvenosti korisnika , valjanosti broja kreditne kartice i sl. Ako su svi podaci valjani, korisnikova lozinka se *hashira* radi sigurnosti te se podaci korisnika spremaju u bazu podataka. On se nakon toga može *logirati* koristeći iste podatke. Slična procedura ponavlja se i prilikom registracije tvrtke.

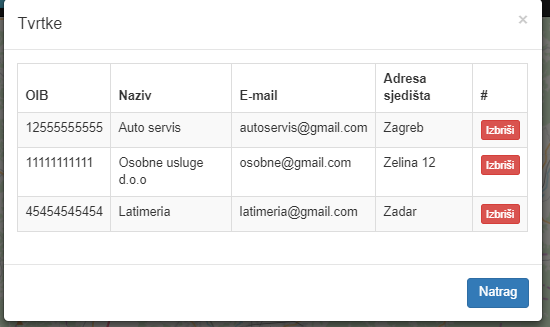
 *Slika 7.3.2. prikazuje isječak koda koji je odgovoran za provjeravanje i spremanje podataka o korisniku*

## Ispitivanje programskog rješenja

Za potrebe ispitivanja registrirana su tri korisnika i tri tvrtke.



Slika 7.4.1 Popis korisnika



Slika 7.4.2 Popis tvrtki

## 7.4.1 Test 1: Registracija

Registracija za korisnika i tvrtku su gotovo identične, razlikuju se u nekoliko podataka koji se upisuju, odnosno ne upisuju. Na primjerima se pokazuju dijelovi registracije tvrtke i korisnika.

Ulaz:

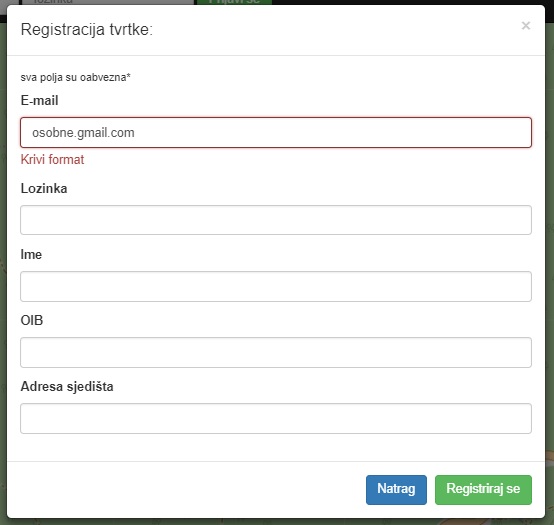
* Odabir registracija, unos podataka u polja registracije

Očekivani izlaz:

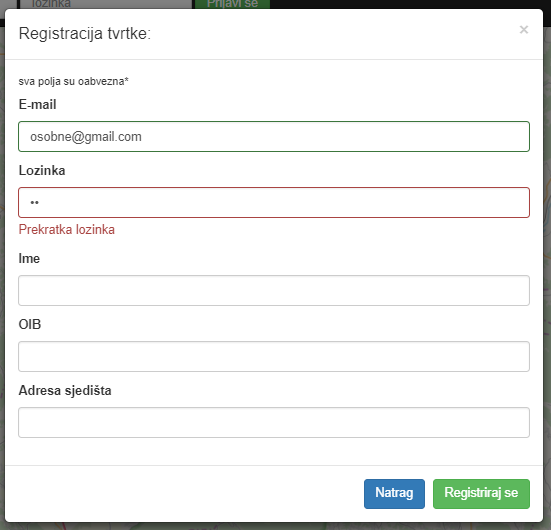
* Neispravan mail
* Prekratka lozinka
* Neispravan oib
* Korisnik već postoji
* Uspješna registracija

Izlaz:

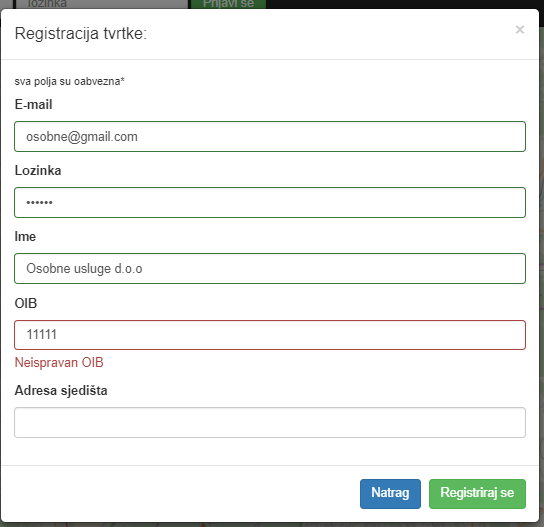
* Slike pokazuju kako su svi izlazi jednaki očekivanima



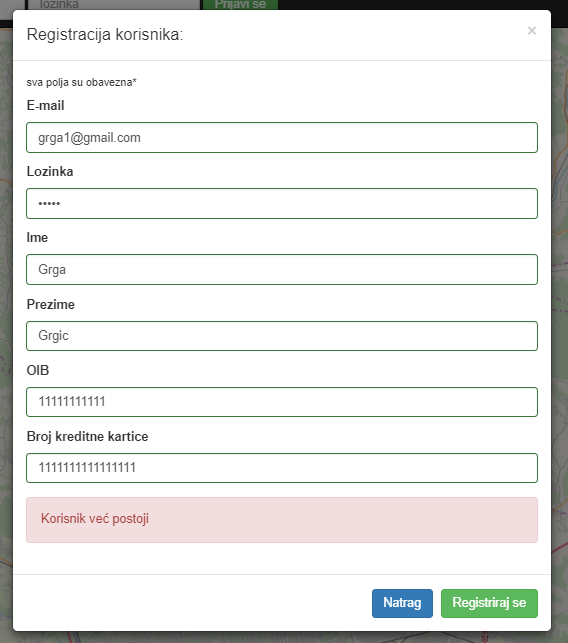
Slika 7.4.1.1 Unos maila u krivom formatu



Slika 7.4.1.2 Unos prekratke lozinke



Slika 7.4.1.3 Unos neispravnog OIB-a



Slika 7.4.1.4 Unos već postojećeg OIB-a



Slika 7.4.1.5 Uspješna registracija

## 7.4.2 Test 2: Prijava

Ulaz:

* Unos podataka u polja prijave

Očekivani izlaz

* Neispravni podatci
* Prikaz početne stranice korisnika/tvrtke

Izlaz

* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.2.1 Unos neispravnih podataka za prijavu



Slika 7.4.2.2 Unos ispravnih podataka za prijavu

## 7.4.3 Test 3: Dodavanje automobila

Ulaz:

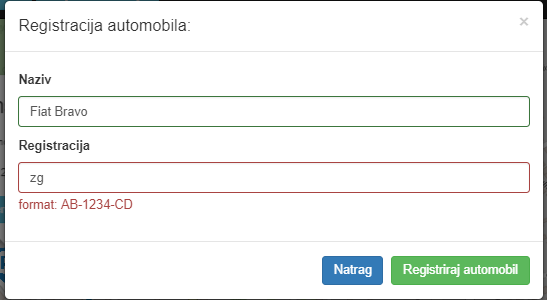
* Odabir dodavanja automobila, unos podataka o automobilu

Očekivani izlaz:

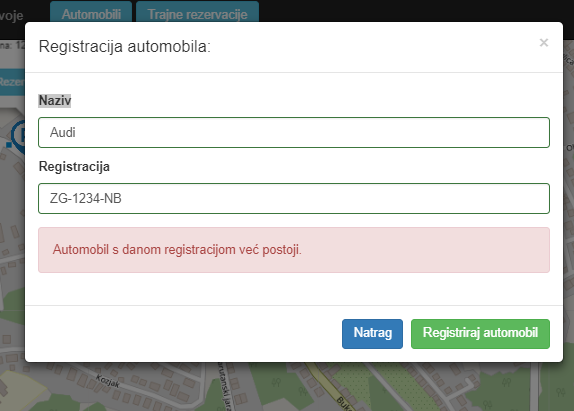
* Nesipravan format registracijske oznake
* Automobil s tom registracijskom oznakom već postoji
* Uspješna registracija automobila

Izlaz:

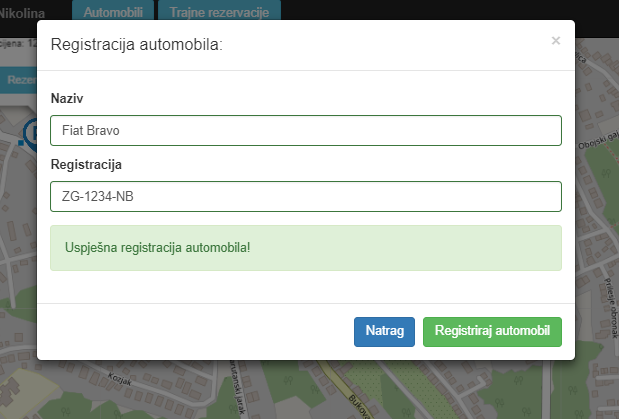
* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.3.1 Unos pogrešnog formata registracijske oznake



Slika 7.4.3.2 Unos već postojeće registracijske oznake

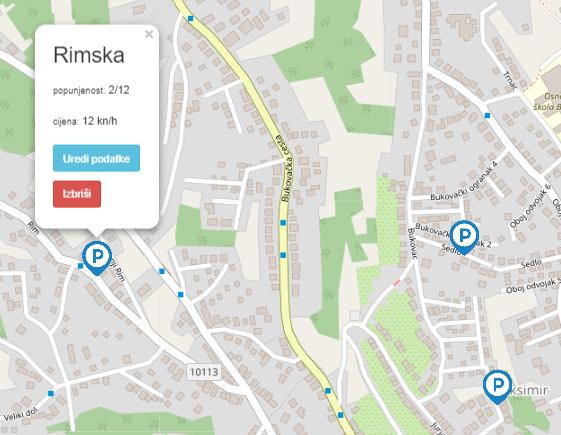


Slika 7.4.3.3 Uspješna registracija automobila

## 7.4.4 Test 4: Uređivanje podataka o parkiralištu

Ulaz:

* Prijavljena tvrtka vidi svoja parkirališta, odabere jedno i odabere Uredi podatke



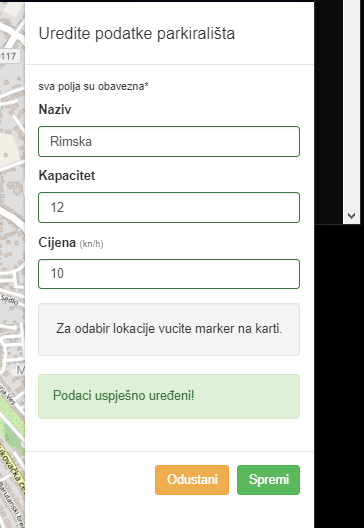
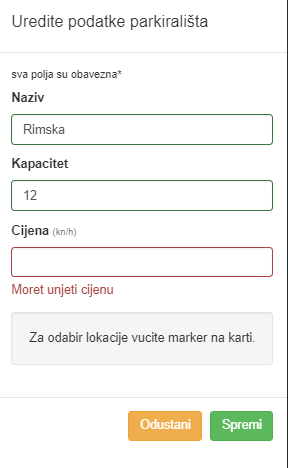
Slika 7.4.4.1 Prikaz svih lokacija te podataka o odabranom parkiralištu

Očekivani izlaz:

* Potrebno unijeti cijenu
* Uspješno uređivanje

Izlaz:

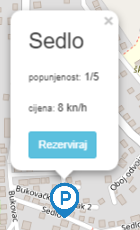
* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.4.2 Nedostatak cijene Slika 7.4.4.1 Uspješno uređivanje podataka

**Rezerviranje parkirnog mjesta**

Korisnik rezervira parkirno mjesto tako da klikne na lokaciju, odnosno parkiralište koje želi i odabere gumb Rezervacija. Zatim bira između jednokratne, ponavljajuće i trajne rezervacije, a ovisno o tom odabiru, ponašanje sustava je različito.



Slika 7.4.3. Prikaz nakon što korisnik odabere parking

## 7.4.5. Test 5: Jednokratna rezervacija

Ulaz:

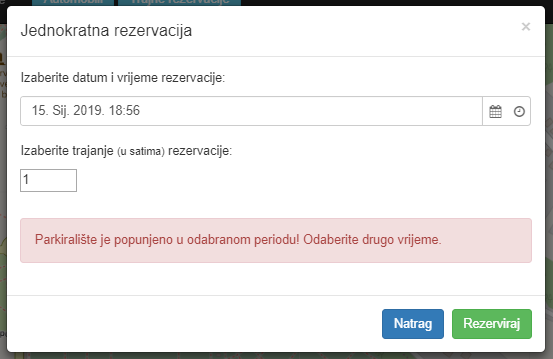
* Odabir datuma, vremena i trajanja rezervacije

Očekivani izlaz:

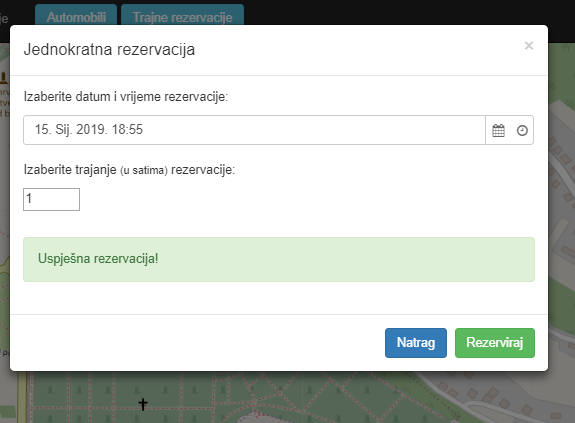
* Parkiralište je popunjeno
* Uspješna rezervacija

Izlaz:

* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.5.1 Parkiralište popunjeno u odabranom periodu



Slika 7.4.5.2 Uspješna jednokratna rezervacija

## 7.4.6. Test 6: Ponavljajuća rezervacija

Ulaz:

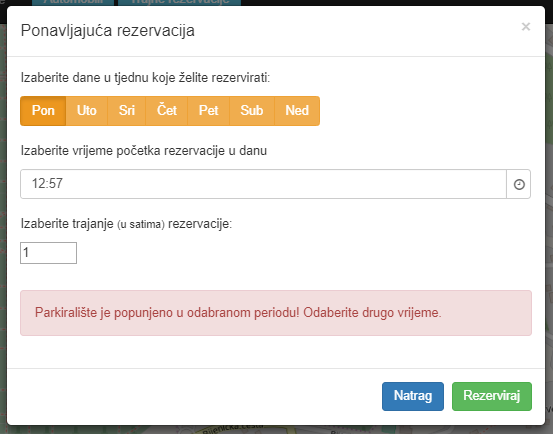
* Odabir dana u tjednu, vremena početka rezervacije u danu i trajanja rezervacije u satima

Očekivani izlaz:

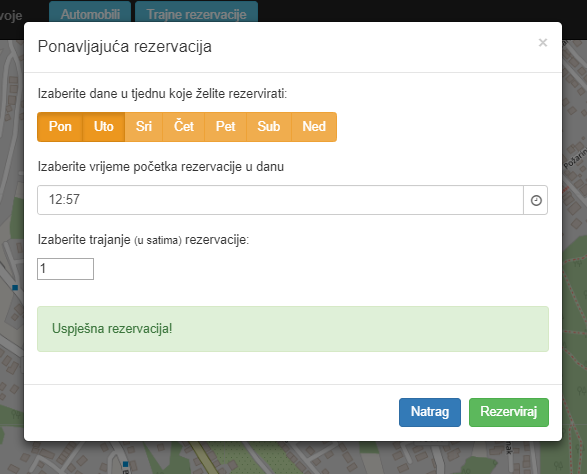
* Parkiralište je popunjeno
* Uspješna rezervacija

Izlaz:

* Slike pokazuju kako su svi izlazi jednaki očekivanima



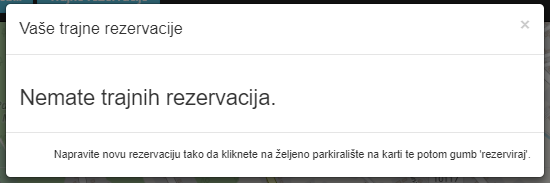
Slika 7.4.6.1 Parkiralište popunjeno u odabranom periodu



Slika 7.4.6.2 Uspješna ponavljajuća rezervacija

## 7.4.7. Test 7: Trajna rezervacija

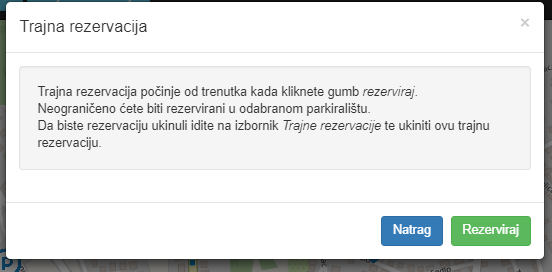
Korisnik može pregledati popis svojih trajnih registracija, koji je početno prazan.



Slika 7.4.7.1 Prazan popis trajnih registracija

Ulaz:

* Odabir jednog parkinga parkinga, odabir gumba Rezerviraj



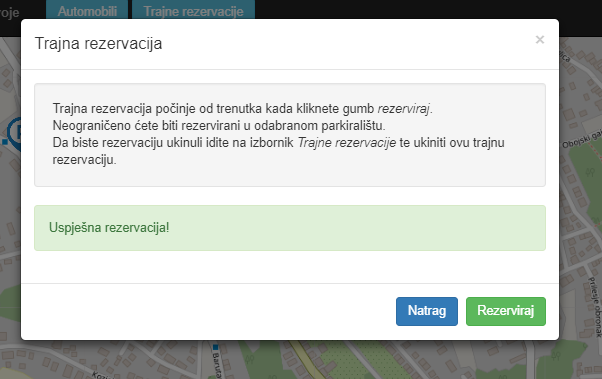
Slika 7.4.7.2. Prikaz nakon odabira Rezerviraj

Očekivani izlaz:

* Uspješna trajna rezervacija

Izlaz:

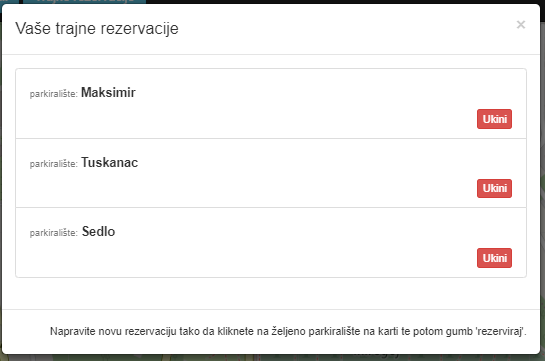
* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.7.3 Uspješna trajna rezervacija

## 7.4.8. Test 8: Brisanje tvrtke

Admin može brisati tvrtke. U slučaju brisanja tvrtke, potrebno je obrisati sva njena parkirališta i ukinuti sve korisničke trajne rezervacije. Kao primjer, izbrsana je tvrtka koja je imala tri parkirališta od kojih je korisnik na dva imao trajne rezervacije. Na prikazu korisnikovih trajnih rezervacija, parkirališta Maksimir i Tuskanac su u vlasništvu tvrtke koja će se izbrisati.



Slika 7.4.8.1 Prikaz korisnikovih trajnih registracija

Ulaz:

* Administrator odabire Obriši uz tvrtku koju želi obrisati

Očekivani izlaz:

* Tvrtka je obrisana, sve trajne rezervacije na njenim parkiralištima su ukinute

Izlaz:

* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.8.2 Prikaz korisnikovih trajnih registracija nakon brisanja tvrtke

## 7.4.9. Test 9: Ukidanje trajne rezervacije

Ulaz:

* Korisnik odabere Trajne registracije, što mu daje popis njegovih trajnih registracija, odabere Ukloni



Slika 7.4.9.1 Popis trajnih rezervacija korisnika

Očekivani izlaz:

* Rezervacija uklonjena

Izlaz:

* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.9.1 Popis trajnih rezervacija korisnika nakon brisanja rezervacije

## 7.4.10. Test 10: Parkiraj Me

Ulaz:

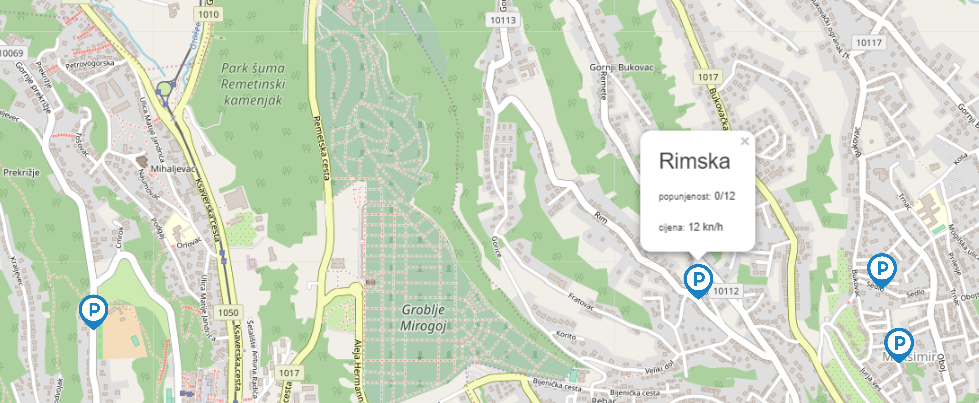
* Odabir Parkiraj me

Očekivani izlaz:

* Na temelju lokacije ponuđen najbliži parking s više od 10 slobodnih mjesta

Izlaz:

* Slike pokazuju kako su svi izlazi jednaki očekivanima



Slika 7.4.10.1 Odgovor na izbor Parkiraj me

## 

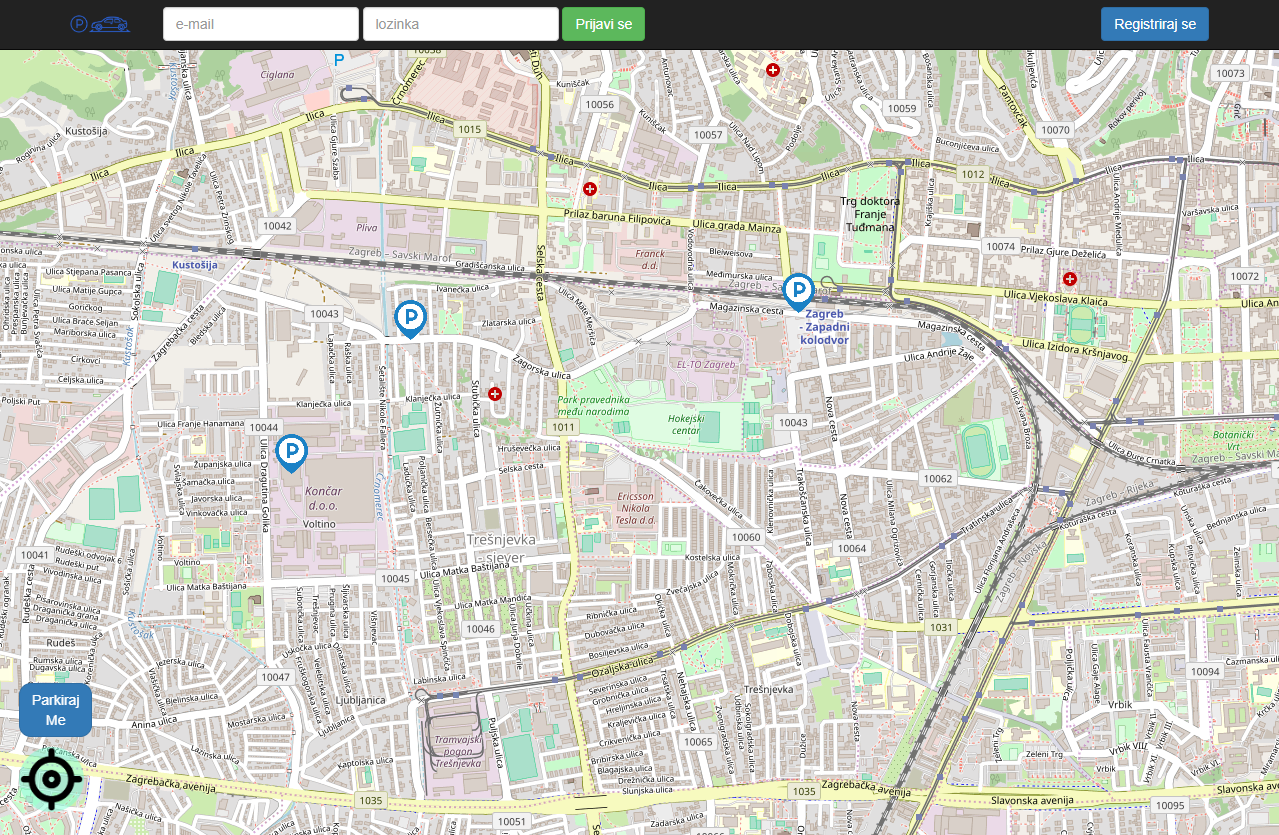
## Upute za instalaciju

*U ovom poglavlju potrebno je dati upute za instalaciju ostvarenog prototipa.*

## Korisničke upute

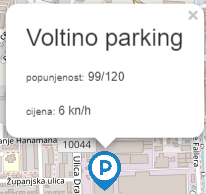
## 7.6.1 Anonimni korisnik

Najmanje mogućnosti na web aplikaciji ima anonimni korisnik. Sve što treba je upisati web adresu aplikacije i to je to, spojen je. Anonimni korisnik može lijevim pritiskom miša kliziti po karti i gledati gdje sve ima parkirališta koje nudi aplikacija.



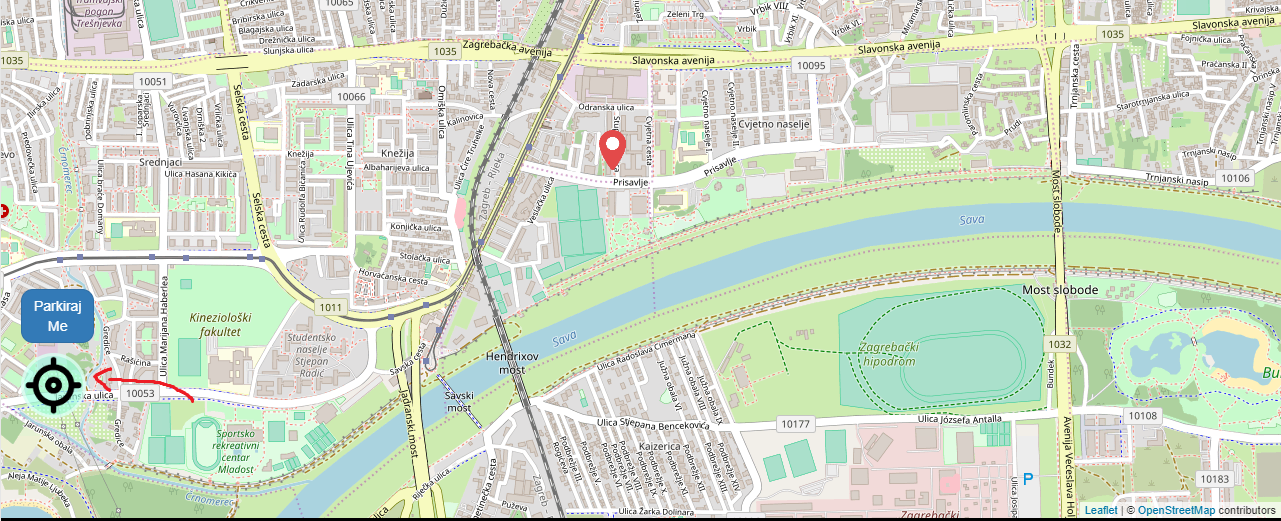
*Slika 7.6.1.1 – Početna stranica anonimnog korisnika*

Desnim pritiskom miša na parkiralište iskače prozorčić sa imenom parkirališta te cijenom po satu.



*Slika 7.6.1.2 –Informacije o parkiralištu*

Jednim klikom miša na oznaku za aplikaciju korisnika se pozicionira na karti tj. odvede na njegovu lokaciju. Također pritiskom na gumb „ParkirajMe“ korisnika se pozicionira na njemu najoptimalniji parking koji nije popunjen.

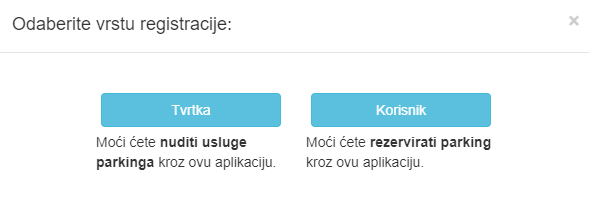


*Slika 7.6.1.3 –Mogućnost lociranja*

Prvi vrhu stranice nalazi se mogućnost prijave i mogućnost registracije. Klikom na gumb „Registriraj se“ korisniku se nude mogućnosti registracije kao tvrtka ili kao korisnik te je objašnjeno što će tko moći koristiti.

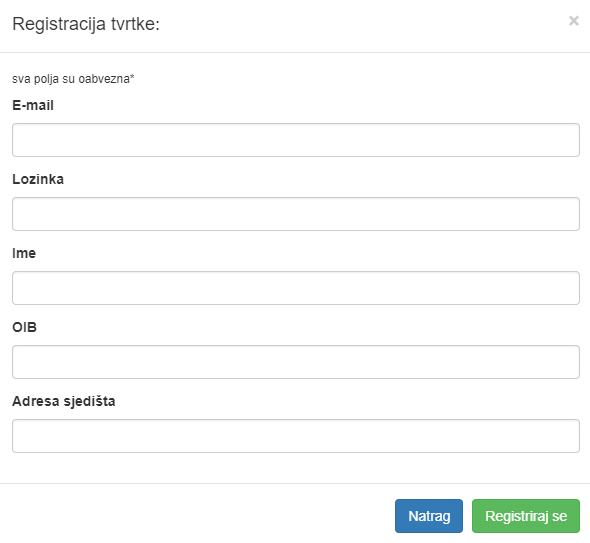
C:\Users\marti\Desktop\68dce00431fa845dc215961118224b1a.png

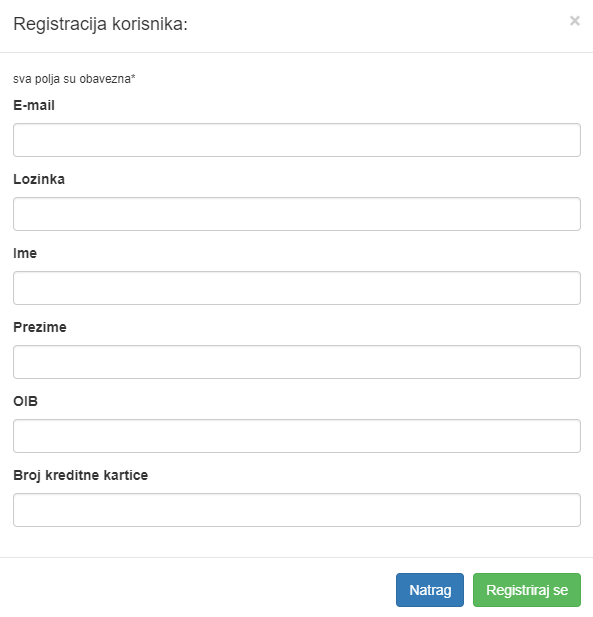
*Slika 7.6.1.4 –Prijava i registriranje*



*Slika 7.6.1.5 – Vrsta registracije*

Ovisno o odabiru vrste registracije potrebno je unijeti drugačije podatke u formular te pritiskom na gumb „Registriraj se“ potvrditi unos.





*Slika 7.6.1.6 – Formular korisnika i tvrtke*

Zadnje što anonimni korisnik može je prijaviti se. U mjesto za prijavu upise svoj e-mail i lozinku te stisne na gumb „Prijavi se“. Ako su upisani podatci točni njega se odvodi na početnu stranicu prijavljenog korisnika te dobiva nove mogućnosti.

C:\Users\Martin\Desktop\4f432774610679366cae87d10784734f.png

*Slika 7.6.1.7 – Prijava korisnika*

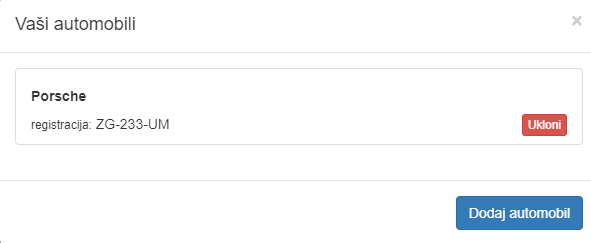
## 7.6.2 Prijavljeni korisnik

Prijavljeni korisnik ima istu stranicu kao i anonimni korisnik samo bez prijave i registracije te sa novim mogućnostima. Mogućnost poput „Parkiraj me“ i „Lociraj me“ su mu omogućene ali su objašnjenje u prethodnom dijelu.

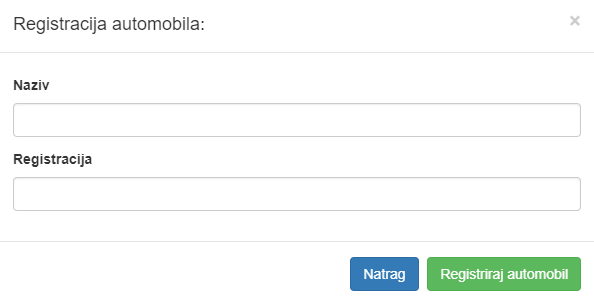
C:\Users\Martin\Desktop\01e85ad4a3f6d2b99ebf7bc315c2b206.png

*Slika 7.6.2.1 – Mogućnosti prijavljenog korisnika*

Odabirom automobili korisniku se izlistavaju svi automobili koje posjeduje te mu se nudi gumb „Dodaj automobil“. Nakon pritiska gumba za dodavanje automobila pojavljuje nam se formular za registraciju automobila. Potrebno ga je potvrditi i pritisnuti gumb „Registriraj automobil“ za potvrdu registracije.



*Slika 7.6.2.2 – Lista automobila*

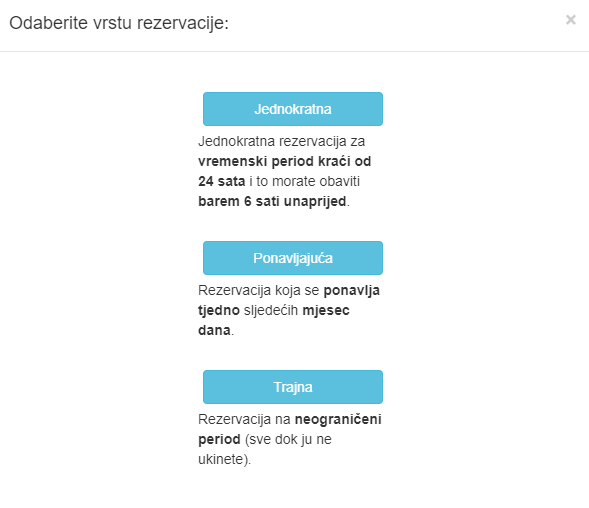


*Slika 7.6.2.3 – Formular za registraciju automobila*

Iduće što prijavljeni korisnik može je Rezervirati parkiralište. Klikom na željeno parkiralište korisnik dobije podatke o parkiralištu te mogućnost rezervacije. Klikom na gumb „Rezerviraj“ korisnik odabire hoće li rezervacija biti jednokratna, ponavljajuća ili trajna. Trajna vrijedi od trenutka kada pritisnemo gumb dok za jednokratnu i ponavljajuću treba popuniti odgovarajuće formulare te potvrditi unos.

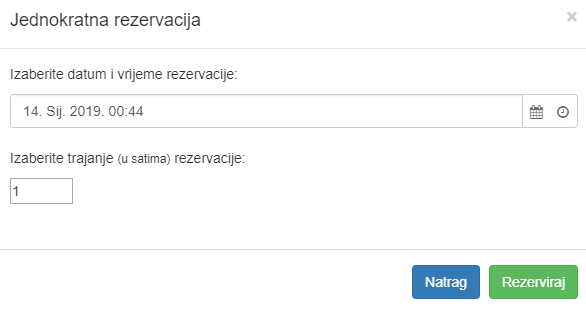


*Slika 7.6.2.4 – Mogućnost rezervacije*

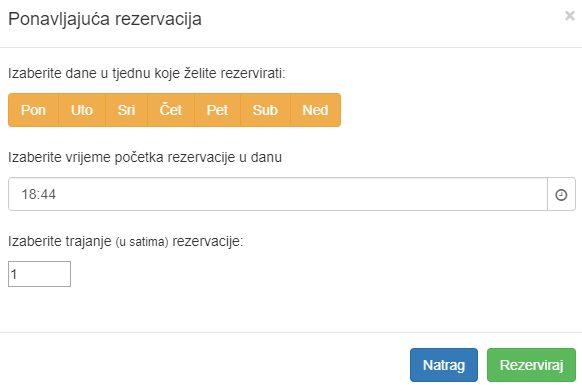


*Slika 7.6.2.5 – Odabir vrste rezervacije*

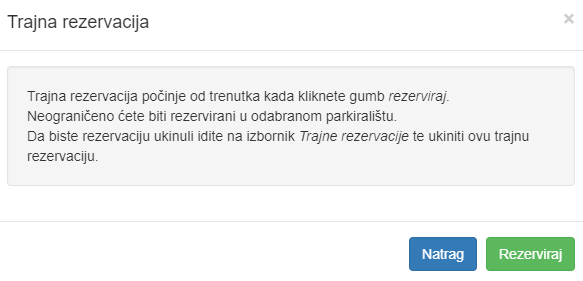
U svakom trenutku bilo korisnik, tvrtka, admin, anonimni korisnik moguće je pritiskom na „X“ u gornjem desnom rubu skočnog prozora odustati od naše radnje ili se klikom na gumb „Natrag“ vratiti korak unatrag.



*Slika 7.6.2.6 – Formular jednokratne rezervacije*



*Slika 7.6.2.7 – Formular ponavljajuće rezervacije*



*Slika 7.6.2.8 – Potvrda trajne rezervacije*

Nakon što je korisnik završio sa svojim radom klikom na gumb odjavi se on se vraća na početnu stranicu anonimnog korisnika te mu je ponovno dostupna prijava s nekim drugim računom.

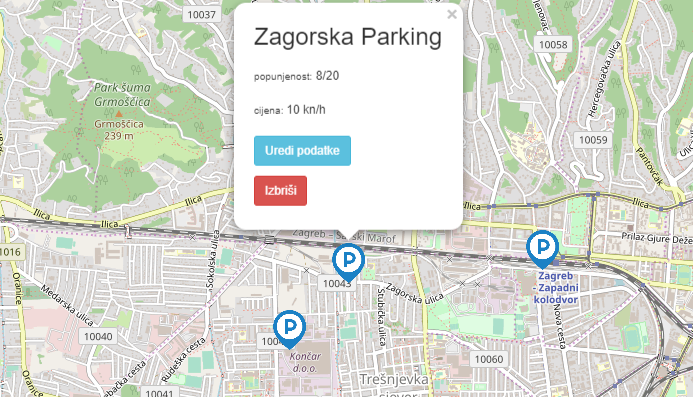
## 7.6.3 Prijavljena tvrtka

Prijavljena tvrtka nema mogućnost lociranja niti opcije „Parkiraj me“ nego može samo dodavati parkinge te mijenjati podatke o postojećim parkiralištima.

C:\Users\Martin\Desktop\5a417d00ab3d53f69afc673cb324011a (1).png

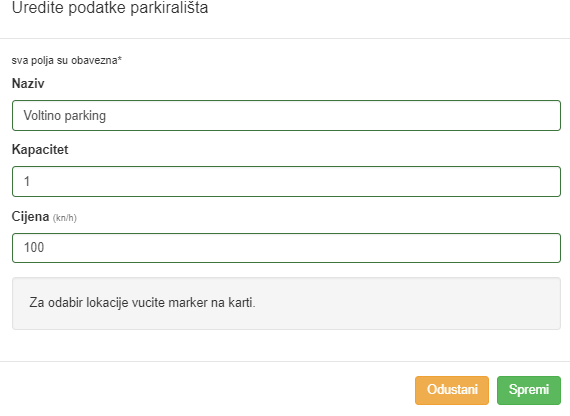
*Slika 7.6.3.1 – Mogućnosti prijavljene tvrtke*

Tvrtka na karti vidi sva svoja parkirališta te pritiskom na neko od njih dobiva opciju uređivanja podataka te brisanje.



*Slika 7.6.3.2 – Uređivanje parkinga*

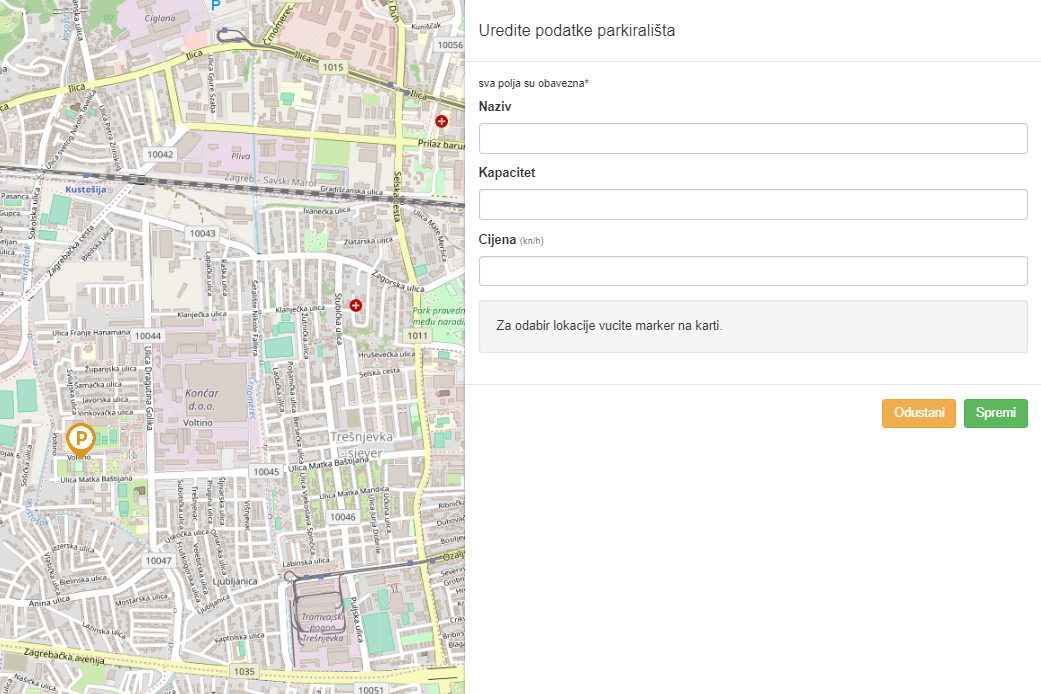
Nakon pritiska gumba za uređivanje parkinga potrebno je ispuniti odgovarajući formular te potvrditi unos.



*Slika 7.6.3.3 – Formular za uređivanje parkirališta*

Ako želimo obrisati parkiralište pritisnemo na gumb „Izbriši“ te ne dobijemo nikakvu obavijest nego se parkiralište obriše i nestane sa karte na koju gledamo.

Ukoliko nemamo parkirališta tada ih na karti neće biti. Potrebno je pritiskom na gumb „Dodaj novi parking“ popuniti formular za kreiranje novog parkinga te na karti kliziti žutim markerom za lokaciju do mjesta na kojem se nalazi naš parking. Potrdimo unos te se parking kreira i pojavljuje na karti.



*Slika 7.6.3.4 – Formular za dodavanje novog parkirališta*

Jedino što nam preostaje je opcija odjavljivanja kada smo gotovi s poslom.

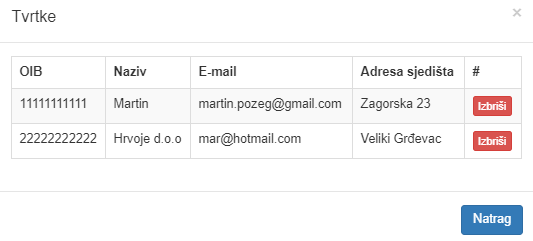
## 7.6.4 Administrator

Administrator se sa posebnim mail-om i sifrom ulogirava u sustav te dobiva maksimalne privilegije.

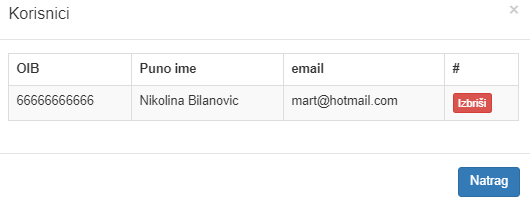
https://i.gyazo.com/31d6e9aaee05e807408aa4f1afbf0c72.png

*Slika 7.6.4.1 – Formular za dodavanje novog parkirališta*

Ukoliko administrator želi pregledati račune registriranih korisnika klikom na gumb „Upraljanje računima“ otvara mu se novi izbornik u kojem je potebno odabrati želimo li pregledati račune tvrtke ili račune korisnika. Nakon izlistavanja odabranih računa administrator ima mogućnost brisanja tog računa klikom na gumb „Izbriši“. Nakon što je kliknuo iz liste računa nestaje odabrani račun bez ikakve potvrde zajedno sa svim njegovim rezervacijama/parkiralištima/automobilima.



*Slika 7.6.4.2 – Lista računa tvrtki*



*Slika 7.6.4.3 – Lista računa korisnika*

# Zaključak i budući rad

Tijekom izrade ove dokumentacije prvi puta svi članovi rade na timskom projektu izrade programske potpore što se pokazao kao izazov. U prvoj fazi smo dokumentirali zahtjeve zadatka i preliminarno modelirali sustav. Slijedi implementacija web aplikacije te proširenje i produbljenje dokumentacije. Nitko od članova tima nikada nije programirao *frontend* ili *backend* pa se moramo upoznati s tehnologijama i dobrim praksama prije nego krenemo sa samom implementacijom.

# Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. Oblikovanje programske potpore, FER ZEMRIS, <http://www.zemris.fer.hr/predmeti/opp>
3. I. Sommerville, „Software engineering“, 8th ed, Addison Wesley, 2007.
4. T.C.Lethbridge, R.Langaniere, „Object-Oriented Software Engineering“, 2nd ed. McGraw-Hill, 2005.
5. Software engineering ,Rutgers University, <http://www.ece.rutgers.edu/~marsic/Teaching/SE>
6. I. Marsic, „Software engineering book“, Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
7. Concepts: Requirements, <http://www.upedu.org/upedu/process/gcncpt/co_req.htm>
8. UML 2 Class Diagram Guidelines, <http://www.agilemodeling.com/style/classDiagram.htm>
9. Domain Class Diagram Modeling Standards and Guidelines, <http://www.bced.gov.bc.ca/imb/downloads/classdiagramstandards.pdf>
10. Astah Community, [http://astah.net/editions/community/](http://argouml.tigris.org/)

# Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda)

[Slika 4.3.1 Sekvencijski dijagram registracije korisnika – UC1, UC2 19](#_lnxbz9)

[Slika 4.3.2 Sekvencijski dijagram za prijavu korisnika – UC3 19](#_35nkun2)

[Slika 4.3.3 Sekvencijski dijagram za rezervacije – UC4-6 21](#_1ksv4uv)

[Slika 4.3.4 Sekvencijski dijagram za upravljanje automobilima – UC7 22](#_44sinio)

[Slika 4.3.5 Sekvencijski dijagram za traženje parkirališta – UC8 23](#_2jxsxqh)

[Slika 4.3.6 Sekvencijski dijagram za upravljanje parkiralištima – UC9 24](#_z337ya)

[Slika 4.3.7 Sekvencijski dijagram za pregled parkirališta – UC10 25](#_3j2qqm3)

[Slika 4.3.8 Sekvencijski dijagram za brisanje računa – UC11 26](#_1y810tw)

[Slika 4.3.9 Sekvencijski dijagram za poništavanje trajne rezervacije – UC12 27](#_4i7ojhp)

[Slika 6.1.1 Arhitektura sustava 28](#_qsh70q)

[Slika 6.1.2 ER model baze podatala 29](#_1pxezwc)

[Slika 6.2.1 Dijagram razreda 32](#_147n2zr)

[Slika 6.3.1 Dijagram objekata 33](#_23ckvvd)

Slika 6.4.1 Dijagram

# Dodatak B: Dnevnik sastajanja

#9.10.2018.

Prvi sastanak. Upoznavanje.

#23.10.2018.

Drugi sastanak. Odabir tehnologije, rasprava o funkcionalnostima aplikacije, popis svih obrazaca uporabe.

#8.11.2018.

Treći sastanak. Ostali zahtjevi, razrada funkcionalnosti aplikacije, model baze podataka, podjela poslova.

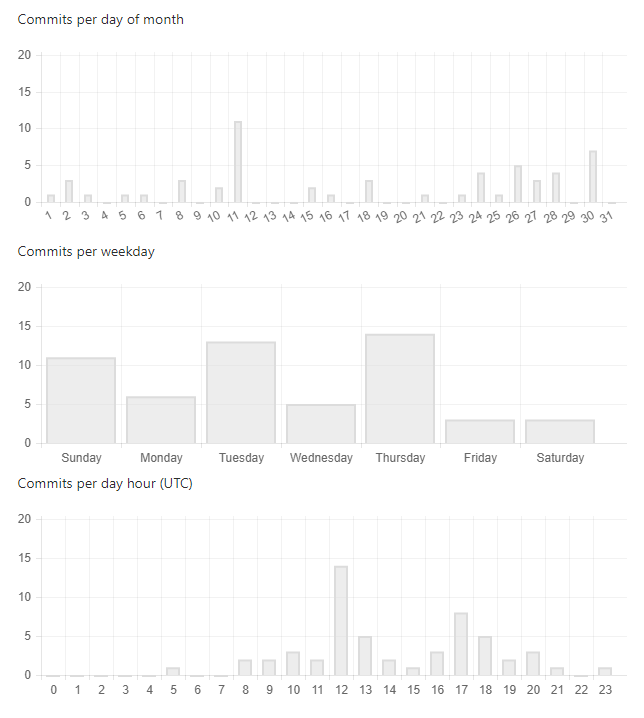
#26.11.2018.

Četvrti sastanak. Dogovorena podjela poslova (Front-end, Back-end), dodaci A-D, dodatno uređivanje dokumentacije.

# Dodatak C: Prikaz aktivnosti grupe

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Popis aktivnosti** |  |  |  |  |  |  |
| Hrvoje Ditrih (voditelj) | Nikolina Bilanović | Domagoj Pavlović | Martin Požeg | Damjan Hudiček | Silvije Škudar |
| **Upravljanje projektom** | 100% |  |  |  |  |  |
| **Opis projektnog zadatka** |  | 50% |  | 50% |  |  |
| **Rječnik pojmova** |  | 33% |  | 34% |  | 33% |
| **Opis funkcionalnih zahtjeva** | 10% | 40% |  | 40% | 10% |  |
| **Opis ostalih zahtjeva** |  |  |  | 100% |  |  |
| **Arhitektura i dizajn sustava** |  |  |  |  |  |  |
| Svrha, opći prioriteti i skica sustava |  | 90% | 10% |  |  |  |
| Dijagram razreda s opisom |  |  |  |  |  |  |
| Dijagram objekata |  |  |  |  |  | 100% |
| Ostali UML dijagrami |  | 50% |  | 50% |  |  |
| **Implementacija i korisničko sučelje** |  |  |  |  |  |  |
| Dijagram razmještaja |  |  |  |  |  |  |
| Korištene tehnologije i alati |  |  |  |  |  |  |
| Isječak programskog kôda |  |  |  |  |  |  |
| Ispitivanje programskog rješenja |  |  |  |  |  |  |
| Upute za instalaciju |  |  |  |  |  |  |
| Korisničke upute |  |  |  |  |  |  |
| **Plan rada** |  |  |  |  |  |  |
| **Pregled rada i stanje ostvarenja** | 100% |  |  |  |  |  |
| **Zaključak i budući rad** | 100% |  |  |  |  |  |
| **Popis literature** | 100% |  |  |  |  |  |
| **Dodaci** |  |  |  |  |  |  |
| Indeks |  | 100% |  |  |  |  |
| Dnevnik sastajanja |  | 100% |  |  |  |  |
|  |  |  |  |  |  |  |

Pregled pohrana kroz vrijeme trajanja projekta u mjesecu studenom:



# Dodatak D: Plan rada / Pregled rada i stanje ostvarenja

* Podjela rada
* Učenje tehnologija: Java Spring, Bootstrap
* Implementacija zadatka
* Nadopuna i produbljenje dokumentacije