

Programsko inženjerstvo

Ak. god. 2023./2024.

SpotPicker

Dokumentacija, Rev. 2

Grupa: Leteći medvjedići

Voditelj: Mario Olčar

Datum predaje: 17. 11. 2023.

Nastavnik: Hrvoje Nuić, mag. ing.

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Dnevnik sastanaka	5
3	Opis projektnog zadatka	6
4	Specifikacija programske potpore	10
4.1	Funkcionalni zahtjevi	10
4.1.1	Obrasci uporabe	11
4.1.2	Dijagrami obrazaca uporabe	16
4.1.3	Sekvencijski dijagrami	18
4.2	Ostali zahtjevi	20
5	Arhitektura i dizajn sustava	21
5.1	Baza podataka	23
5.1.1	Opis tablica	23
5.1.2	Dijagram baze podataka	26
5.2	Dijagram razreda i opis razreda	26
5.3	dijagram stanja	29
5.4	dijagram aktivnosti	30
5.5	Dijagram komponenti	33
6	Implementacija i korisničko sučelje	34
6.1	Korištene tehnologije i alati	34
6.2	Ispitivanje programskog rješenja	35
6.3	Dijagram razmještaja	35
6.4	Upute za puštanje u pogon	36
6.4.1	Pokretanje backend dijela aplikacije	36
7	Zaključak i budući rad	38
	Popis literature	39

Indeks slika i dijagrama	40
7.1 Tablica aktivnosti	41

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.01	Napravljen predložak	Mario Olčar	22.10.2023
0.10	Dodan opis zadatka	Mario Olčar	29.10.2023
0.19	Dodani opisi obrazaca uporabe	Paula Močinić	8.11.2023
0.28	Dodani dijagrami obrazaca uporabe	Paula Močinić	8.11.2023
0.37	Dodani sekvencijski dijagrami i njihovi opisi	Paula Močinić	9.11.2023
0.46	Stvorena mapa sa slikama dijagrama	Paula Močinić	12.11.2023
0.55	Dodani ostali zahtjevi	Paula Močinić	13.11.2023
0.61	Dodani funkcijski zahtjevi	Ivan Bušljeta	15.11.2023
0.70	Dodan ER dijagram	Lovro De Villa	15.11.2023
0.79	Dodan opis arhitekture i dizajna sustava	Lovro De Villa	16.11.2023
0.75	Dodan opis baze podataka i tablice	Tomislav Marenić	16.11.2023
0.79	Dodan opis arhitekture i dizajna sustava	Lovro De Villa	16.11.2023
0.80	Ispravak pogreške, dodan dijagram modela i njegov opis	Tomislav Marenić	17.11.2023

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.88	Dodan dijagram DTO i njegov opis	Ivan Bušljeta	17.11.2023
0.89	Dodan dijagram kontrolera i njegov opis	Lovro De Villa	17.11.2023
0.99	Ispravak gramatičkih pogrešaka	Ivan Bušljeta, Mario Olčar	17.11.2023
1.01	Ispravak dijagrama obrazaca uporabe	Paula Močinić	17.12.2023
1.02	Ispravak funkcionalnih zahtjeva i opisa obrazaca uporabe	Paula Močinić	17.12.2023
1.1	Dodan dijagram stanja i njegov opis	Paula Močinić	8.1.2024
1.2	Dodane Korištene tehnologije i alati	Mario Olčar	8.1.2024
1.3	Dodan dijagram aktivnosti	Ivan Bušljeta	8.1.2024
1.5	Dodan zaključak	Paula Močinić	12.01.2024
2.0	Predaja druge revizije	Mario Olčar	19.01.2024

2. Dnevnik sastanaka

Datum sastanka	Opis	Voditelj
2.11.2023	Prvi uvodni sastanak za početak rada, raspodjela poslova i obaveza	Mario Olčar
7.11.2023	izvještaji o napretku	Mario Olčar
11.11.2023	backend sastanak	Matija Huđin
10.12.2023	stvari koje ćemo raditi nakon međuispita i analiza dokumentacije nakon predaje za drugi ciklus	Mario Olčar
17.12.2023	sastanak	Mario Olčar
18.12.2023	backend sastanak	Matija Huđin
20.12.2023	backend sastanak	Matija Huđin
27.12.2023	sastanak	Mario Olčar
8.1.2023	sastanak	Mario Olčar
15.1.2023	sastanak	Mario Olčar

3. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "SpotPicker" koji će omogućiti rezervaciju, naplatu parkiranja i pregled slobodnih parkirališnih mjesta za automobile i bicikle.

Aplikacija za rezerviranje parkinga pruža niz koristi kako vozačima, tako i vlasnicima parkirališta. Vozači često gube vrijeme tražeći slobodna parkirališta, pogotovo u gusto naseljenim područjima ili tijekom gužvi. Aplikacija za rezerviranje parkinga im omogućuje da rezerviraju svoje mjesto, čime se smanjuje potreba za traženjem parkirnog prostora i eliminira stres povezan s time. Također rezervacija pruža vozačima sigurnost i garanciju parkirnog prostora kad stignu na odredište. To je posebno korisno tijekom raznih događanja, koncerata illi sportskih manifestacija, kada je potražnja za parkirnim mjestima visoka. S druge strane vlasnici parkirališta mogu bolje upravljati svojim resursima koristeći informacije o rezervacijama. Aplikacija omogućuje praćenje popunjenosti parkirališta, što pomaže u planiranju i optimizaciji korištenja prostora. Ovo može rezultirati boljim iskorištavanjem kapaciteta parkirališta, povećanjem prihoda i poboljšanjem općeg iskustva korisnika.

Kako bi se ovo postiglo, stranica mora biti pregledna, razumljiva i korisnicima lako dostupna. Do svega na stranici trebalo bi moći doći u samo nekoliko klikova. Sučelje mora biti vizualno atraktivno i jednostavno za korištenje. Korisniku moraju biti lako dostupne i vidljive sve najvažnije informacije o pojedinom parkingu: koliko je mjesta slobodno na pojedinom parkingu, njena cijena i moguća ograničenja vezana uz vozila.

Posebnu pažnju trebalo bi posvetiti i originalnosti web stranice po čemu istaknuti u usporedbi s konkurencijom i time privukla više korisnika. Također tada bi i više parkirališta moglo odabrati našu stranicu za rezervaciju i naplatu parkinga.

Korist od stranice imat će prvenstveno kupci, jer je izgradnja stranice usmjerena njima kao najbrojnijoj skupini korisnika. Također, korist će imati i voditelji

parkirališta kojima će stranica omogućiti ostvarenje većeg profita. Stranica će im poslužiti kao oblik oglašavanja te će im olakšati nalaženje kupaca. Time će im se povećati konkurentnost na tržištu.

Korisnike stranice smo podijelili na više skupina. To su administratori, voditelji parkinga, registrirani korisnici ili klijenti i neregistrirani korisnici.

Neregistrirani korisnik može poslati zahtjev za registraciju sa željenom ulogom za koju se prijavljuje (voditelj parkinga ili klijent), a potrebni su:

- korisničko ime
- lozinka
- ime
- prezime
- slika osobne iskaznice
- IBAN
- email adresa

Administrator može vidjeti popis svih registriranih korisnika i njihovih osobnih podataka te im mijenjati osobne podatke. Registracija se završava potvrdom preko email adrese. a ako se korisnik registrirao kao voditelj dodatno ga mora potvrditi administrator.

Voditelj parkinga ima mogućnost unijeti informacije o svom parkiralištu (naziv, opis, fotografija, cjenik i sl.) i u kartu ucrtati svako dostupno parkirališno mjesto za to parkiralište. Voditelj definira je li moguće rezervirati parkirališno mjesto te postavlja senzor koji osvježava informaciju o zauzetosti parkirališnog mjesta.

Neregistrirani korisnici u aplikaciji mogu pregledati sva parkirališta i parkirališna mjesta koja su dostupna, dok se klijentima (prijavljenim korisnicima) dodatno prikazuje informacija o njihovoj zauzetosti u stvarnom vremenu.

Pregledavanjem karte, klijent može odabrati lokaciju svog odredišta, tip vozila i procjenu trajanja parkinga, a aplikacija mu na karti iscrta rutu do najbližeg slobodnog parkirališnog mjesta i rezervira ga ako je slobodno za rezervaciju. Za dohvat rute do parkirališnog mjesta koristiti se OSRM1.

Klijent može rezervirati parkirališna mjesta na dva načina:

- Prvi način je da na karti označi parkirališna mjesta za koja je zainteresiran i potom mu se otvori kalendar s dostupnim terminima.
- Drugi način je da označi željeni termin te da mu se na karti prikažu parkirališna mjesta koja su slobodna za rezervaciju u tom terminu. Rezervacije mogu trajati proizvoljno dugo i biti definirane kao ponavljajuće, a vođa za svoje parkiralište definira cijenu ovisno o trajanju rezervacije. Korisnik mjesto može rezervirati samo u budućnosti (dakle, ne uključujući datum za vrijeme kojeg korisnik pristupa aplikaciji).

Plaćanje parkinga preko aplikacije izvršava se prilikom rezervacije, a klijent u aplikaciji posjeduje novčanik kojeg može nadopuniti u bilo kojem trenutku.

Osim navedenih zahtjeva, važno je naglasiti dodatne specifikacije koje će definirati funkcionalnosti i karakteristike sustava za rezerviranje parkinga. Sustav treba omogućiti rad više korisnika istovremeno u stvarnom vremenu. To znači da više vozača može istovremeno pristupati aplikaciji, pregledavati dostupna parkirališta i obavljati rezervacije bez značajnog gubitka performansi. Ova funkcionalnost pridonosi učinkovitom korištenju resursa, posebno u situacijama s visokim prometom korisnika.

Pristup bazi podataka, koji je ključan dio funkcionalnosti sustava, mora biti brz i učinkovit. Izvršavanje dijela programa koji pristupa bazi podataka ne smije trajati duže od nekoliko sekundi, čime se osigurava odzivnost sustava i izbjegava frustracija korisnika uslijed dugih vremena čekanja. Sustav treba biti implementiran kao web aplikacija koristeći objektno orijentirane jezike. Ova arhitektura olakšava održavanje, proširivost i razumijevanje koda, čineći sustav skalabilnim i prilagodljivim budućim promjenama.

Sustav koristi EURO kao valutu. To omogućuje jednostavnu i konzistentnu razmjenu informacija o cijenama rezervacija te osigurava dosljednost pri plaćanjima.

Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške. Ovo osigurava sigurnost podataka, stabilnost sustava te sprječava gubitak podataka uslijed neželjenih događaja. Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS protokola, čime se osigurava sigurna komunikacija između

korisnika i sustava. Ova mjera dodatno štiti privatnost podataka i osigurava da su informacije sigurne tijekom prijenosa.

4. Specifikacija programske potpore

4.1 Funkcionalni zahtjevi

Dionici:

1. Voditelj parkirališta (naručitelj)
2. Klijenti
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi su sljedeći:

1. Voditelj parkirališta može:
 - unijeti informacije o svom parkiralištu (naziv, opis, fotografija, cjenik)
 - ucrtati u kartu svako dostupno parkirališno mjesto
 - označiti da je mjesto moguće rezervirati
 - vidjeti informacije o zauzetosti dostupnih parkirališnih mjesta u realnom vremenu
 - dodati ili obrisati parkirališno mjesto
2. Neregistrirani korisnik može:
 - pregledati na karti dostupna parkirališta i parkirališna mjesta (bez informacije o zauzetosti u stvarnom vremenu)
 - odabrati parkiralište i dobiti prikaz općih informacija (naziv, opis, fotografija, cjenik)
 - poslati zahtjev za registraciju u sustav sa željenom ulogom za koju se prijavljuje (voditelj parkinga ili klijent)
3. Klijent može:
 - vidjeti informacije o zauzetosti dostupnih parkirališnih mjesta u realnom vremenu
 - nadopuniti svoj novčanik
 - odabrati lokaciju svog odredišta, tip vozila i procjenu trajanja parkinga

- rezervirati parkirališna mjesta označavanjem mjesta na karti
- rezervirati parkirališna mjesta odabirom označavanjem željenog termina

4. Administrator može:

- vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- vidjeti i promijeniti razinu pristupa aplikaciji korisnicima (klijent, vođitelj parkirališta)
- potvrditi vođitelja parkirališta
- dodati ili obrisati parkiralište

4.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pregled dostupnih parkirališnih mjesta

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati dostupna parkirališta i njihovu zauzetost
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Klijent ulazi u aplikaciju i pregledava kartu sa svim dostupnim parkiralištima
 2. Klijent bira parkiralište koje ga zanima.
 3. Aplikacija prikazuje informacije o parkiralištu i zauzetosti parkirališnih mjesta na tom parkiralištu.
- **Opis mogućih odstupanja:**
 - 2.a Klijent odabere parkiralište koje trenutno nema dostupnih slobodnih mjesta.
 1. Aplikacija obavještava klijenta da nema slobodnih mjesta na odabranom parkiralištu.
 - 2.b Klijent pokuša pregledati parkiralište koje ne postoji u sustavu.
 1. Aplikacija obavještava klijenta o nepostojećem parkiralištu.

UC2 - Rezervacija parkirališta

- **Glavni sudionik:** Klijent
- **Cilj:** Rezervirati parkiralište za svoje vozilo
- **Sudionici:** Baza podataka

- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Klijent odabire parkiralište na karti i željeni datum i vrijeme rezervacije
 2. Aplikacija prikazuje dostupna slobodna parkirališna mjesta na odabranoj lokaciji za navedeni datum i vrijeme
 3. Klijent odabire slobodno parkirališno mjesto i potvrđuje rezervaciju
 4. Aplikacija omogućuje plaćanje rezervacije
 5. Nakon uspješne rezervacije, klijent prima potvrdu rezervacije putem e-maila
- **Opis mogućih odstupanja:**
 - 2.a Klijent odabire parkirališne na kojem nema slobodnih mjesta.
 1. Aplikacija obavještava klijenta da na parkiralištu nema slobodnih mjesta.
 - 3.a Klijent odabire slobodno parkirališno mjesto koje u međuvremenu postane zauzeto.
 1. Aplikacija obavještava klijenta da se parkiralište promijenilo i predlaže novo slobodno mjesto.

UC3 - Dodavanje informacija o parkiralištu (za vođitelje parkinga)

- **Glavni sudionik:** Vođitelj parkinga
- **Cilj:** Dodati informacije o svom parkiralištu
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao vođitelj parkinga
- **Opis osnovnog tijeka:**
 1. Vođitelj parkinga unosi informacije o svom parkiralištu, uključujući naziv, opis, fotografiju, cjenik i sl
 2. Vođitelj parkinga može ucrtati svako dostupno parkirališno mjesto za svoje parkiralište
 3. Vođitelj parkinga definira je li moguće rezervirati parkirališno mjesto i postavlja senzor koji osvježava informaciju o zauzetosti parkirališnog mjesta
- **Opis mogućih odstupanja:**
 - 1.a Vođitelj parkinga pokuša dodati informacije o parkiralištu koja već postoje u sustavu
 1. Aplikacija obavještava vođitelja parkinga o već postojećim informacijama i omogućava izmjenu postojećih podataka

3.a Voditelj parkinga pokušava postaviti senzore na nepostojeće parkiralište.

1. Aplikacija obavještava voditelja parkinga o nepostojećem parkiralištu i sugerira unos postojećeg parkirališta.

UC4 - Statistika zauzetosti parkirališta

- **Glavni sudionik:** Voditelj parkinga
- **Cilj:** Pregledati statistiku zauzetosti parkirališta i parkirališnih mjesta kroz vrijeme
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao voditelj parkinga
- **Opis osnovnog tijeka:**
 1. Voditelj parkinga bira parkiralište za koje želi pregledati statistiku
 2. Aplikacija prikazuje grafički prikaz statistike zauzetosti parkirališta i parkirališnih mjesta tijekom vremena

UC5 - Administracija korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Upravljanje korisnicima i njihovim osobnim podacima
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav kao administrator
- **Opis osnovnog tijeka:**
 1. Administrator pregledava popis svih registriranih korisnika
 2. Administrator može mijenjati osobne podatke korisnika
- **Opis mogućih odstupanja:**
 - 2.a Administrator pokušava izmijeniti podatke korisnika na nedozvoljen način.
 1. Sustav obavještava administratora o neispravnoj izmjeni podataka i onemogućava ju.

UC6 - Prikaz parkirališta za bicikle

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati dostupna parkirališta za bicikle
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Klijent pregledava dostupna parkirališta za bicikle na karti
 2. Aplikacija prikazuje informacije o parkiralištima za bicikle i ukupnom broju slobodnih mjesta
- **Opis mogućih odstupanja:**
 - 2.a Klijent pregledava parkiralište za bicikle koje trenutno nema slobodnih mjesta
 1. Aplikacija obavještava klijenta da nema slobodnih mjesta na odabranom parkiralištu za bicikle.

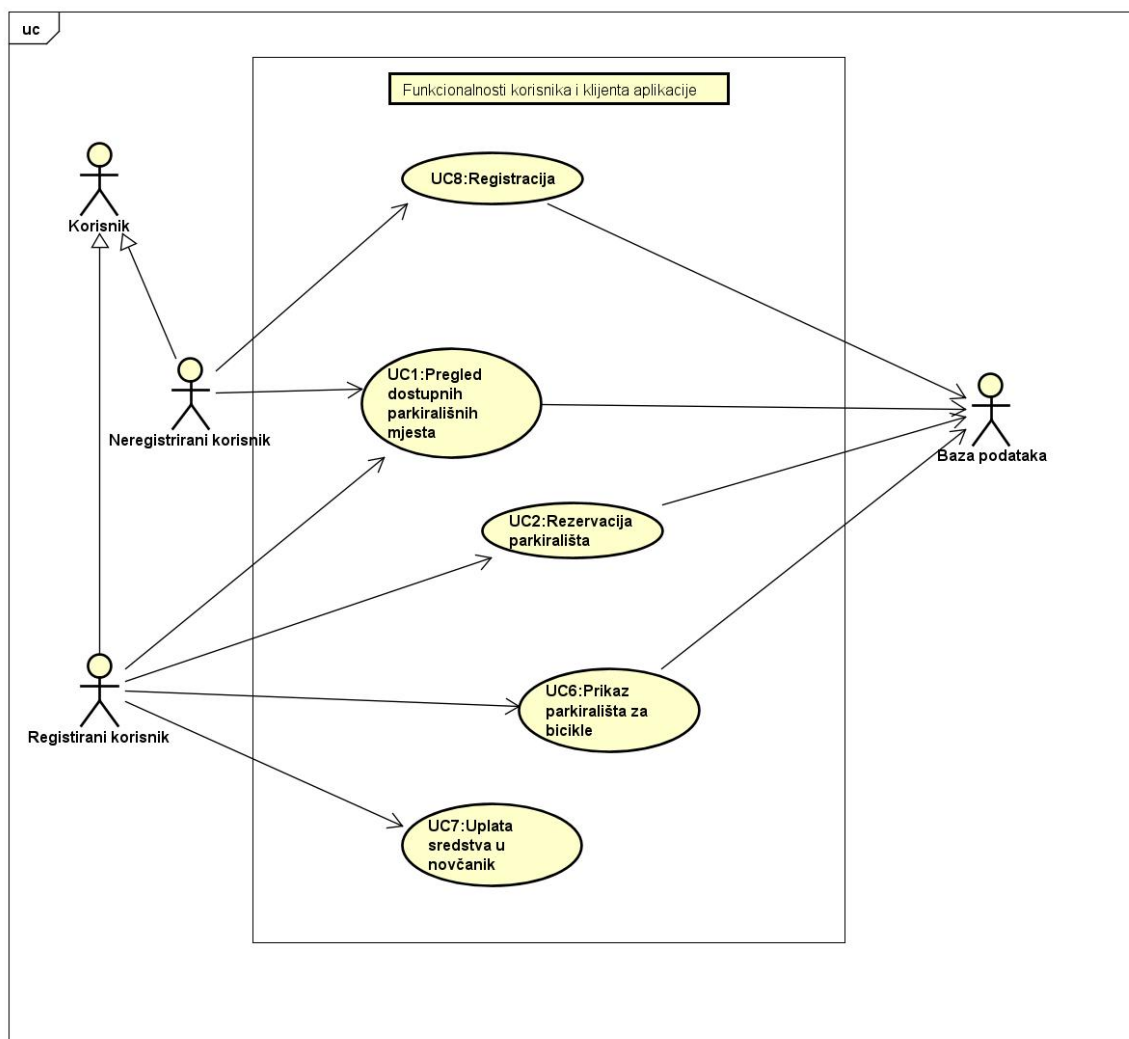
UC7 - Uplata sredstava u novčanik

- **Glavni sudionik:** Klijent
- **Cilj:** Nadopuniti novčanik sredstvima za plaćanje parkinga
- **Sudionici:** Bankovni sustav
- **Preduvjet:** Prijava u sustav
- **Opis osnovnog tijeka:**
 1. Klijent odabire opciju za uplatu sredstava
 2. Klijent unosi iznos koji želi uplatiti
 3. Aplikacija preusmjerava korisnika na sigurnu stranicu za plaćanje gdje unosi bankovne podatke
 4. Nakon uspješne uplate, sredstva se dodaju u novčanik korisnika
- **Opis mogućih odstupanja:**
 - 3.a Klijent pokušava izvršiti uplatu, ali bankovni sustav ne uspijeva obraditi transakciju
 1. Klijent prima obavijest o neuspjeloj uplati i dobiva priliku ponovno pokušati uplatu.

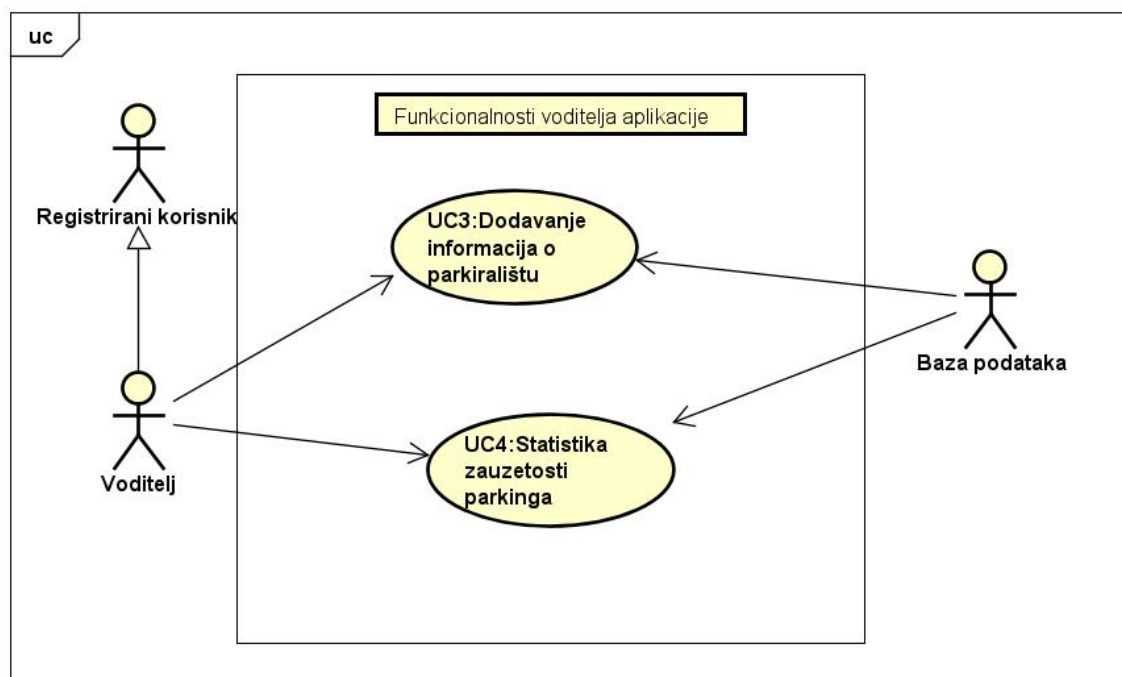
UC8 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

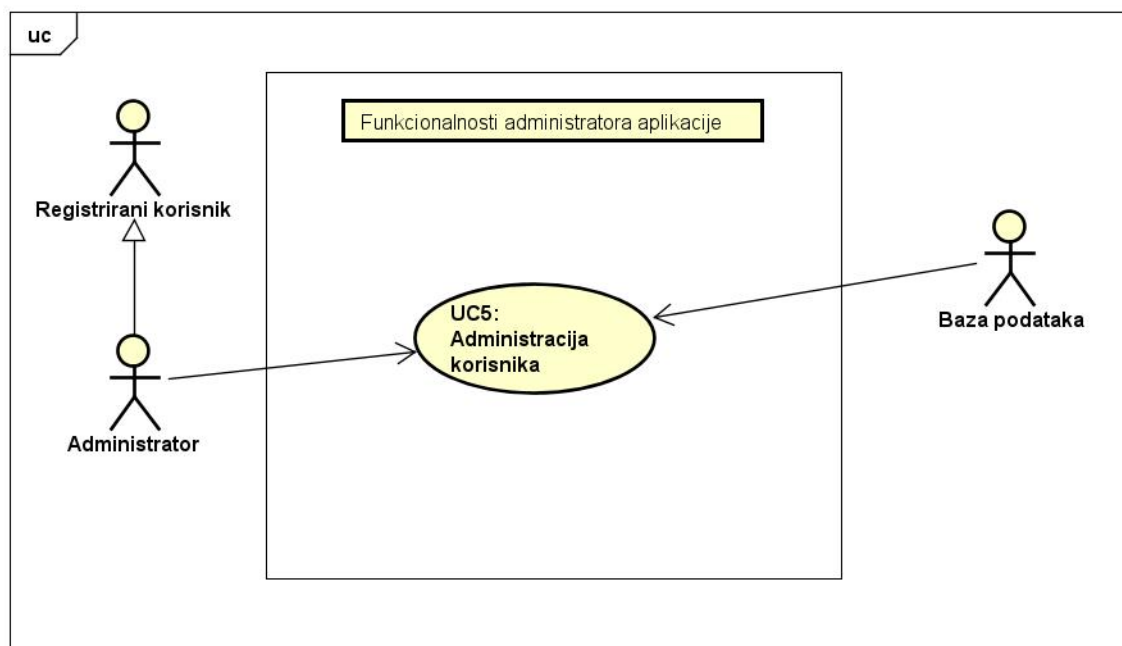
4.1.2 Dijagrami obrazaca uporabe



Slika 4.1: Dijagram obrasca uporabe, funkcionalnost korisnika i klijenta



Slika 4.2: Dijagram obrasca uporabe, funkcionalnost vlasnika

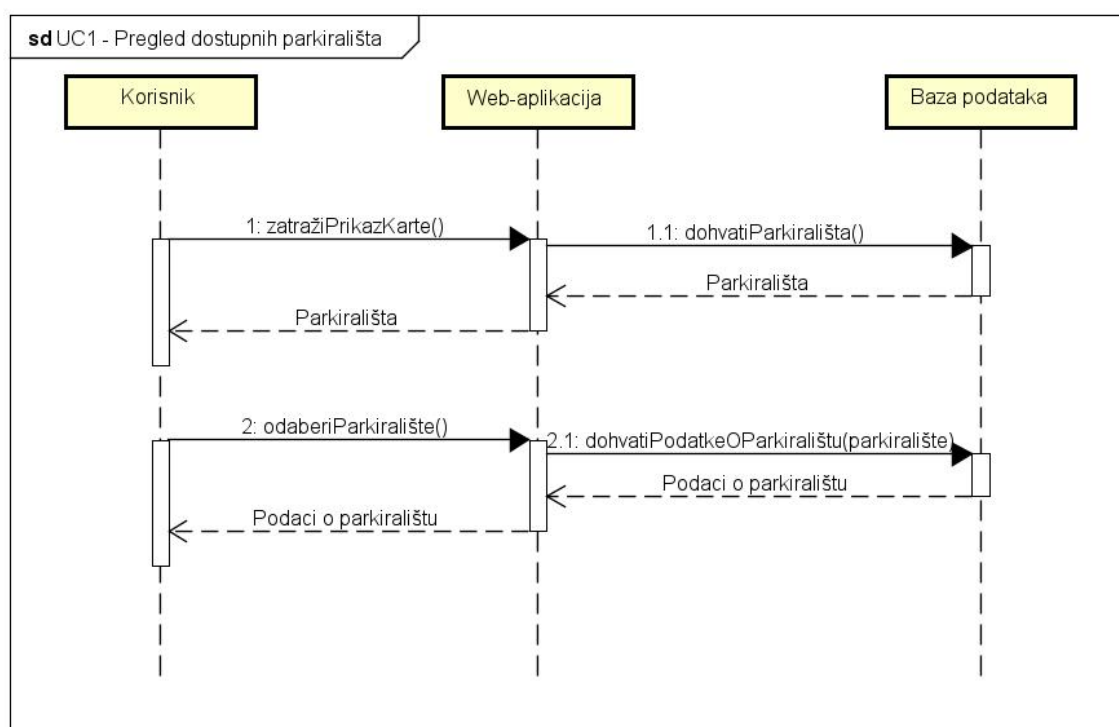


Slika 4.3: Dijagram obrasca uporabe, funkcionalnost administratora

4.1.3 Sekvencijski dijagrami

Obrazac uporabe UC1 - Pregled dostupnih parkirališta

Klijent šalje zahtjev za kartografskim prikazom s dostupnim parkiralištima kako bi odabrao parkiralište. Aplikacija dohvaća trenutne podatke o svim parkiralištima iz baze podataka i prikazuje ih korisniku. Nakon što klijent odabere parkiralište na karti, aplikacija šalje upit bazi podataka kako bi dohvatila osnovne informacije o odabranom parkiralištu. Baza podataka odgovara na upit i šalje informacije o parkiralištu i zauzetosti parkirališnih mjesta, što aplikacija prikazuje korisniku.

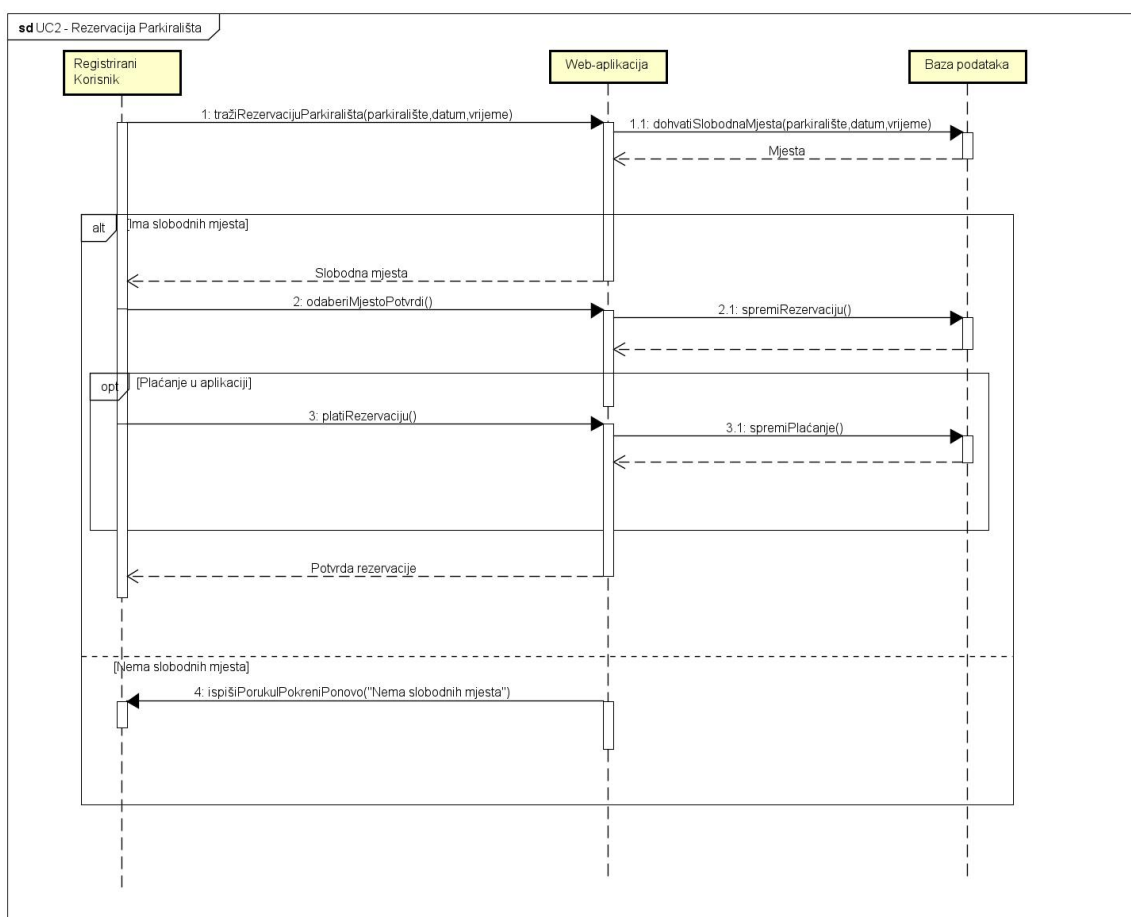


Slika 4.4: Sekvencijski dijagram za UC1

Obrazac uporabe UC2 -Rezervacija parkirališta

Klijent šalje zahtjev za rezervaciju parkirališta odabirom parkirališta na karti i specificiranjem datuma i vremena rezervacije. Aplikacija šalje upit bazi podataka kako bi provjerila dostupnost slobodnih parkirališnih mjesta na odabranoj lokaciji za navedeni datum i vrijeme. Baza podataka provjerava dostupna mjesta i šalje informacije o slobodnim parkirališnim mjestima aplikaciji. Ako nema slobodnih

mjesta na odabranom parkiralištu aplikacija obavještava klijenta o tome uz poruku. Aplikacija prikazuje klijentu dostupna slobodna mjesta i omogućuje odabir. Klijent bira slobodno parkirališno mjesto i potvrđuje rezervaciju. Aplikacija šalje upit bazi podataka za rezervaciju parkirališta. Baza podataka rezervira parkiralište i šalje potvrdu rezervacije aplikaciji. Aplikacija omogućuje klijentu plaćanje rezervacije. Aplikacija šalje potvrdu plaćanja bazi podataka. Baza podataka ažurira status rezervacije i potvrđuje plaćanje. Aplikacija šalje potvrdu rezervacije klijentu. Klijent prima potvrdu rezervacije.



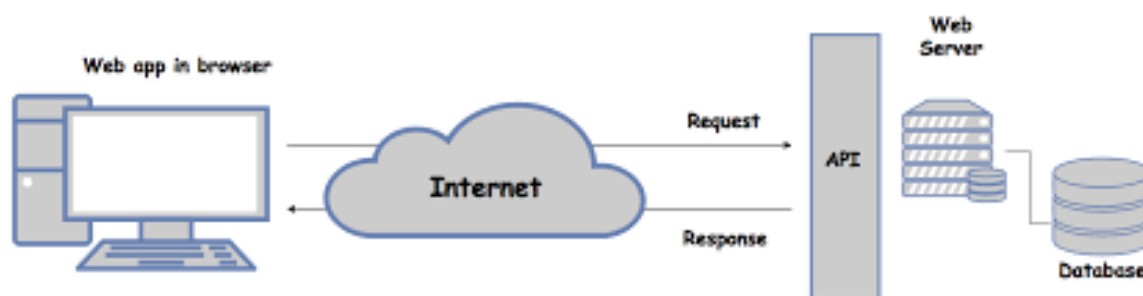
Slika 4.5: Sekvencijski dijagram za UC2

4.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Sustav kao valutu koristi EURO
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS

5. Arhitektura i dizajn sustava

Web aplikacija je softverski program koja se izvršava na web poslužitelju i koja omogućuje korisnicima pristup i interakciju s raznim funkcijama, uslugama ili informacijama putem web preglednika.



Slika 5.1: Arhitektura sustava

Povezanost između web aplikacije i web poslužitelja je ključna za funkcioniranje web aplikacija. Evo osnovne poveznice:

- **Klijentska strana (Frontend):**

- Web aplikacija obično ima klijentsku stranu koja se izvršava u web pregledniku korisnika. Ova klijentska strana može biti izrađena korištenjem HTML-a, CSS-a i JavaScripta, a može koristiti i frontend okvire poput React-a kojeg smo odabrali u izradi naše web aplikacije.

- **Poslužiteljska strana (Backend):**

- Backend dio web aplikacije izvršava se na web poslužitelju. Ovdje se izvršava poslovna logika aplikacije, upravlja se bazom podataka, obrađuju se zahtjevi korisnika, provjerava se autentikacija, autorizacija itd. Backend se obično piše u jezicima poput Node.js, Pythona, Rubyja, Java-e, a mi smo se odlučili za C# zajedno s .NET radnim okvirom.

- **Komunikacija između klijenta i poslužitelja:**
 - Web aplikacija komunicira s poslužiteljem putem HTTP (Hypertext Transfer Protocol) zahtjeva i odgovora. Kada korisnik interagira s web aplikacijom putem svog preglednika (klikne na gumb, ispuni obrazac), preglednik šalje zahtjev web poslužitelju. Web poslužitelj obrađuje taj zahtjev i šalje odgovor natrag pregledniku.
- **Povratne informacije korisniku:**
 - Kada web poslužitelj obradi zahtjev, šalje odgovor natrag na klijentsku stranu (preglednik). To može uključivati HTML, CSS, JavaScript, podatke iz baze podataka ili druge resurse koji se prikazuju korisniku putem web preglednika.

Odabrali smo Microsoft Visual Studio kao naše razvojno okruženje. Koristit ćemo arhitekturu sustava temeljenu na MVC (Model-View-Controller) konceptu. Ovaj koncept podržan je unutar .NET radnog okvira te pruža gotove predloške koji olakšavaju razvoj web aplikacija. Jedna od glavnih karakteristika MVC koncepta je njegova sposobnost za nezavisni razvoj pojedinih dijelova aplikacije. Ovo omogućuje lakše testiranje aplikacije kao i jednostavnije dodavanje novih svojstava u sustav bez potrebe za velikim modifikacijama.

MVC arhitektura se sastoji od:

1. Model: Model predstavlja sloj podataka aplikacije i poslovnu logiku. Ovdje se podaci obrađuju, pohranjuju i pristupa im se. Model je odgovoran za interakciju s bazom podataka ili nekim drugim izvorom podataka, kao i za manipulaciju tim podacima prema zahtjevima poslovne logike.
2. View: View predstavlja korisničko sučelje aplikacije, odnosno ono što korisnik vidi i s čime interagira. Pogled prikazuje podatke iz Modela na način koji je razumljiv i koristan korisniku. Obično se radi o HTML-u, CSS-u, JavaScriptu ili nekom drugom obliku sučelja koje korisnik može vidjeti.
3. Controller: Kontroler je posrednik između Modela i View-a. On obrađuje korisničke zahtjeve primljene putem korisničkog sučelja (View), upravlja tim zahtjevima i ažurira Model prema tim zahtjevima. Kontroler reagira na akcije korisnika, ažurira podatke u Modelu i određuje koji View će se prikazati korisniku.

5.1 Baza podataka

Za potrebe razvoja aplikacije SpotPicker koristi se relacijska baza podataka. Osnovne zadaće baze podataka su pohrana i organizacija podataka te brzo pretraživanje i dohvaćanje podataka kako bi ih se moglo dalje obraditi. Svaki je entitet korištene relacijske baze naveden i opisan u daljnjem tekstu, a ispod opisa nalazi se tablični prikaz opisanog entiteta i njegovih atributa.

Baza podataka aplikacije SpotPicker sastoji se od sljedećih entiteta:

- ConfirmationLink
- Korisnik
- Parking
- Rezervacija
- Wallet

5.1.1 Opis tablica

ConfirmationLink

Entitet ConfirmationLink sadrži informacije vezane za potvrđene korisnike i sadrži sljedeće attribute: ConfirmationLinkID, KorisnikID, Link i isValid. Primarni ključ entiteta ConfirmationLink je atribut ConfirmationLinkID, a strani ključ je atribut KorisnikID. Navedeni je entitet u vezi *One-to-One* s entitetom Korisnik preko atributa KorisnikID.

ConfirmationLink		
Confirmation LinkID	INT	jedinstveni ID svakog linka sa potvrdu pri registraciji
Korisnik ID	INT	jedinstveni ID korisnika
Link	NVARCHAR	točan link koji se koristio za potvrdu registracije
isValid	BIT	informacija o potvrdi korisnika

Korisnik

Entitet Korisnik sadrži informacije vezane za registrirane korisnike aplikacije i sadrži sljedeće attribute: KorisnikID, Username, Password, RazinaPristupa, Name,

Surname, PictureData, BankAccountNumber, Email, AccountEnabled i EmailVerified. Primarni ključ entiteta Korisnik je KorisnikID. Navedeni je entitet u vezi *One-to-Many* s entitetom ConfirmationLink preko atributa KorisnikID, vezi *One-to-Many* s entitetom Parking preko atributa KorisnikID, u vezi *One-to-Many* s entitetom Rezervacija preko atributa KorisnikID i u vezi *One-to-One* s entitetom Wallet preko atributa KorisnikID.

Korisnik		
KorisnikID	INT	jedinstveni ID korisnika
Username	NVARCHAR	korisničko ime
Password	NVARCHAR	lozinka
Razina Pristupa	INT	voditelj parkinga(1) ili običan korisnik(0)
Name	VARCHAR	ime korisnika
Surname	VARCHAR	prezime korisnika
PictureData	VARBINARY	slika osobne
BankAccount Number	VARCHAR	broj bankovnog računa korisnika
Email	VARCHAR	korisnikova e-mail adresa
Account Enabled	BIT	informacija o potvrdi profila korisnika od strane administratora
Email Verified	BIT	informacija o potvrdi profila korisnika putem e-maila

Parking

Entitet Parking sadrži informacije vezane za opis i konfiguraciju parkirališta te sadrži sljedeće atribute: ParkingID koji je primarni ključ entiteta, Name, Description, Photo, PricePerHour, Capacity i KorisnikID. Strani ključ entiteta *Parking* je KorisnikID. Navedeni je entitet u vezi *Many-to-One* s entitetom Korisnik preko atributa KorisnikID, i u vezi *One-to-Many* s entitetom Rezervacija preko atributa ParkingID.

Parking		
ParkingID	INT	jedinstveni ID parkirališta
Name	NVARCHAR	naziv parkirališta
Description	VARCHAR	opis parkirališta
Photo	VARBINARY	fotografija parkinga
PricePerHour	INT	cijena parkirališnog mjesta po satu
Capacity	INT	broj parkirališnih mjesta na parkiralištu
KorisnikID	INT	jedinstveni ID korisnika

Rezervacija

Entitet Rezervacija sadrži informacije vezane za rezervacije pojedinih parkirnih mjesta i sadrži sljedeće atribute: ReservationID koji je primarni ključ entiteta, KorisnikID, ParkingID, DateTimeStart, DateTimeEnd i ParkingPlaceId. Strani ključevi entiteta *Rezervacija* su KorisnikID i ParkingID. Navedeni je entitet u vezi *Many-to-One* s entitetom Korisnik preko atributa KorisnikID.

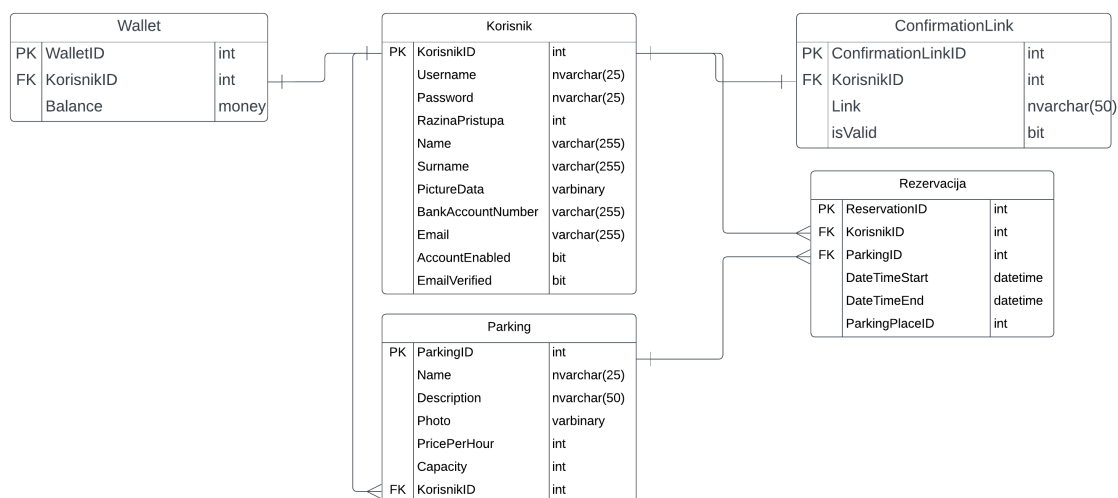
Rezervacija		
ReservationID	INT	jedinstveni ID rezervacije
KorisnikID	INT	jedinstveni ID korisnika
ParkingID	INT	jedinstveni ID parkirališta
DateTimeStart	DATETIME	početno datum-vrijeme rezervacije
DateTimeEnd	DATETIME	završno datum-vrijeme rezervacije
ParkingPlace ID	INT	jedinstveni ID parkirališnog mjesta

Wallet

Entitet Wallet sadrži informacije vezane za novčanik korisnika i njegova sredstva. Sadrži sljedeće atribute: WalletID koji je primarni ključ entiteta, KorisnikID koji je strani ključ entiteta i Balance. Navedeni je entitet u vezi *One-to-One* s entitetom Korisnik preko atributa KorisnikID.

Wallet		
WalletID	INT	jedinstveni ID novčanika
KorisnikID	INT	jedinstveni ID korisnika
Balance	MONEY	iznos sredstava u novčaniku korisnika

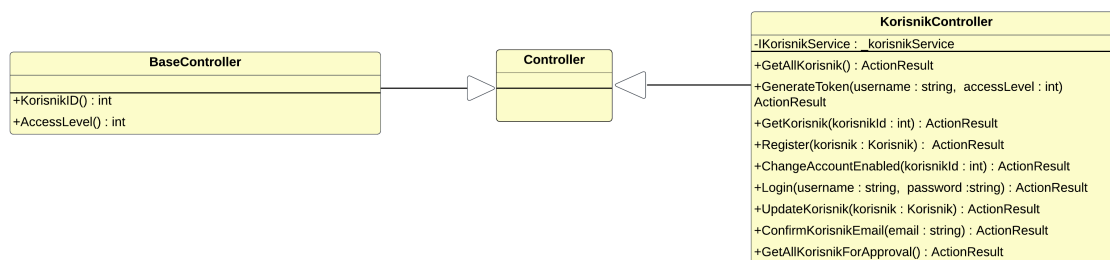
5.1.2 Dijagram baze podataka



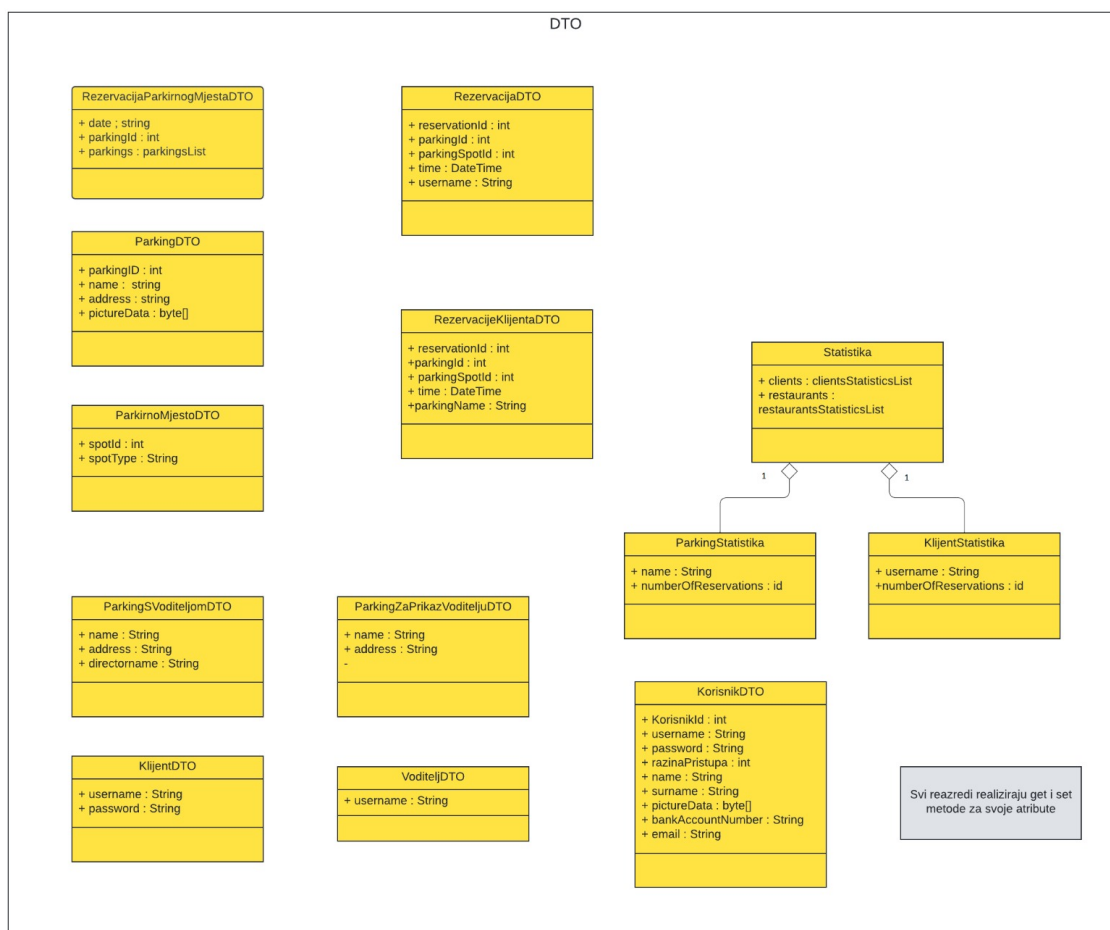
Slika 5.2: ER dijagram baze podataka

5.2 Dijagram razreda i opis razreda

Na slikama 4.3, 4.4 i 4.5 su prikazani razredi koji pripadaju *backend* dijelu MVC arhitekture. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Metode implementirane u tim razredima manipuliraju s DTO (Data transfer object), a oni su dohvaćeni pomoću metoda implementiranih u Model razredima. Metode implementirane u Controller razredima vraćaju JSON datoteke s html status kodom.



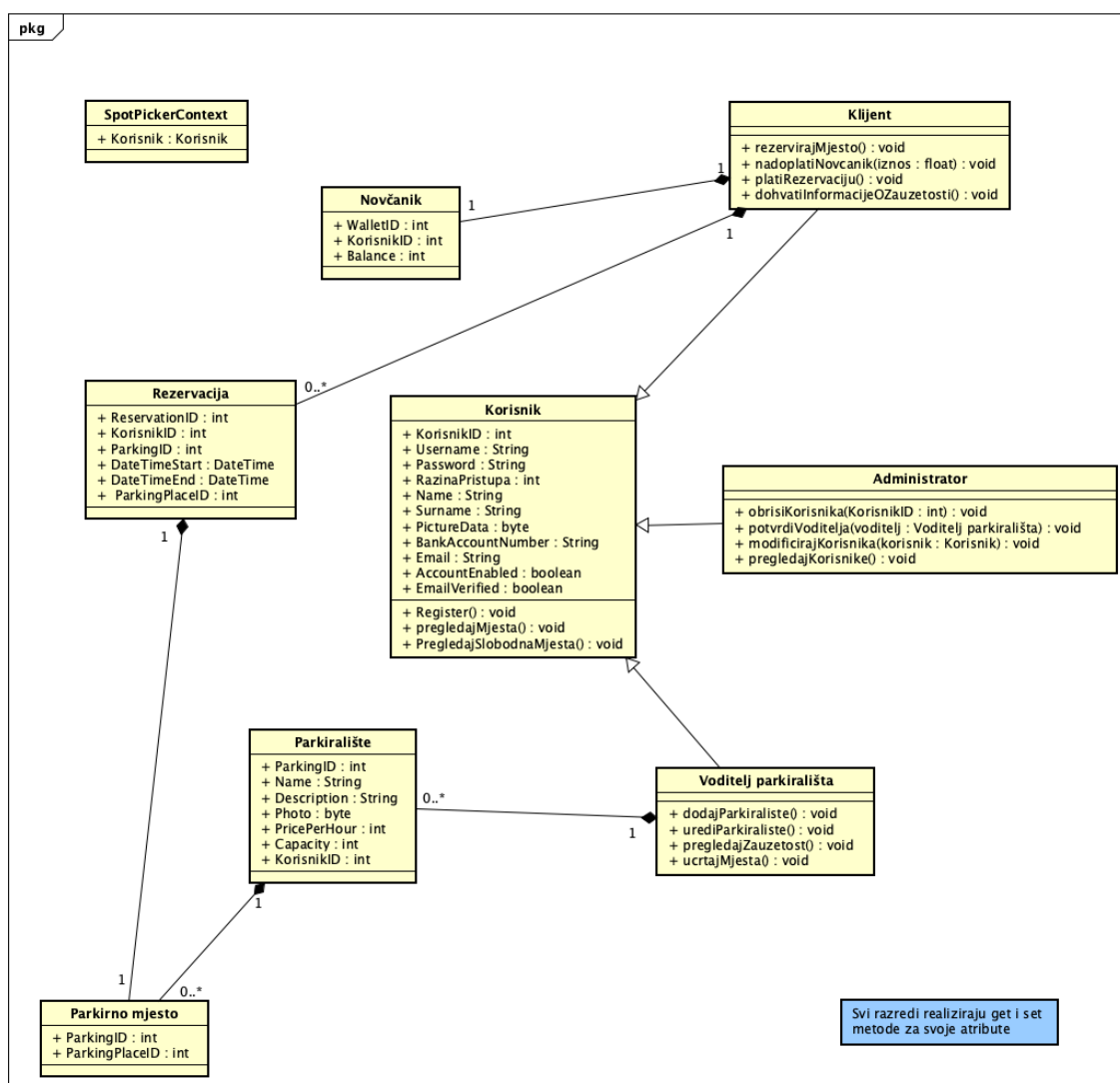
Slika 5.3: Dijagram razreda - dio Controllers



Slika 5.4: Dijagram razreda - dio Data transfer objects

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke. Razred Korisnik predstavlja neregistriranog korisnika koji se može registrirati u sustav kao klijent ili vođa parkinga, unoseći osnovne informacije. Neregistri-

rani korisnik može vidjeti sva parkirna mjesta i posebno slobodna parkirna mjesta. Razred Klijent predstavlja korisnika koji je registriran u sustav. On može koristiti osnovne funkcionalnosti sustava kao i neregistrirani korisnik i posebno dobiti informacije o zauzetosti pojedinih mjesta u stvarnom vremenu. Osim toga može nadoplatiti svoj novčanik i rezervirati parkirno mjesto. Razred Voditelj parkinga predstavlja registriranog korisnika koji može dodati parkirališta. Razred Administrator predstavlja administratora sustava koji ima najveće ovlasti. Ima mogućnost mijenjanja podataka registriranih korisnika i uklanjanje istih. Posebno potvrđuje voditelje parkirališta. Razred Novčanik predstavlja novčanik klijenta te prikazuje iznos njegovih raspoloživih sredstava. Razred Parkiralište predstavlja parkiralište koje je dodao voditelj parkinga.

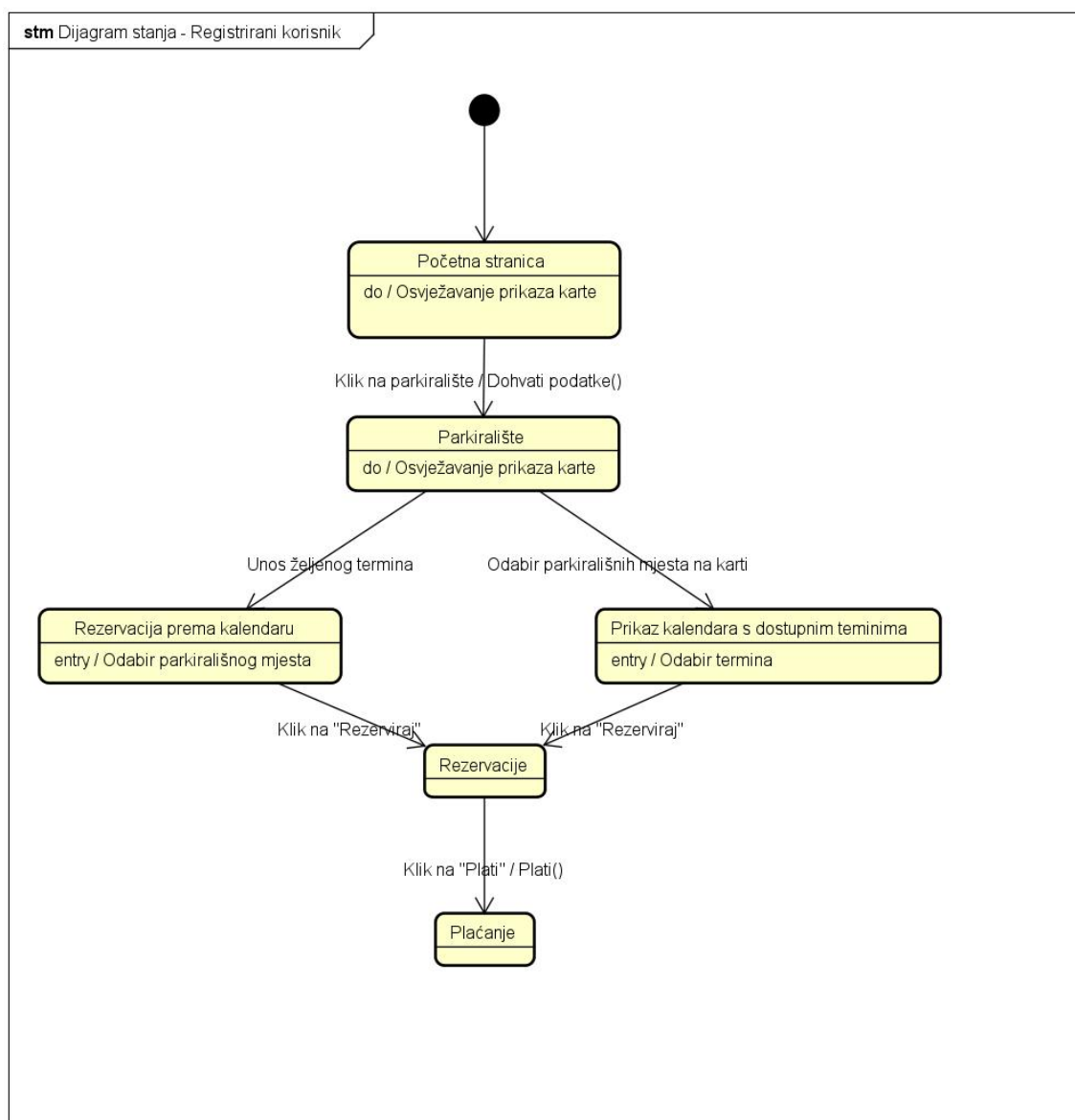


Slika 5.5: Dijagram razreda - dio Models

5.3 dijagram stanja

Dijagram stanja za aplikaciju SpotPicker prikazuje moguća stanja korisnika i sustava te prijelaze između tih stanja na temelju različitih događaja. Na slici 4.6 prikazan je dijagram stanja za registriranog korisnika. Nakon uspješne prijave u aplikaciju, registriranom korisniku se prikazuje početna stranica koja omogućuje pregled dostupnih parkirališta. Korisnik ima mogućnost pregleda informacija o svakom parkiralištu, njegovoj zauzetosti, te vrsti vozila koje se može parkirati. Za odabrano parkiralište korisnik može provjeriti informacije o cijenama, trajanju par-

kinga te dostupnim terminima za rezervaciju parkirališnih mjesta. Korisniku je omogućeno nadopunjavanje novčanika putem aplikacije radi plaćanja rezervacija ili uplate parkinga prilikom dolaska na lokaciju slobodnog parkirališnog mjesta.

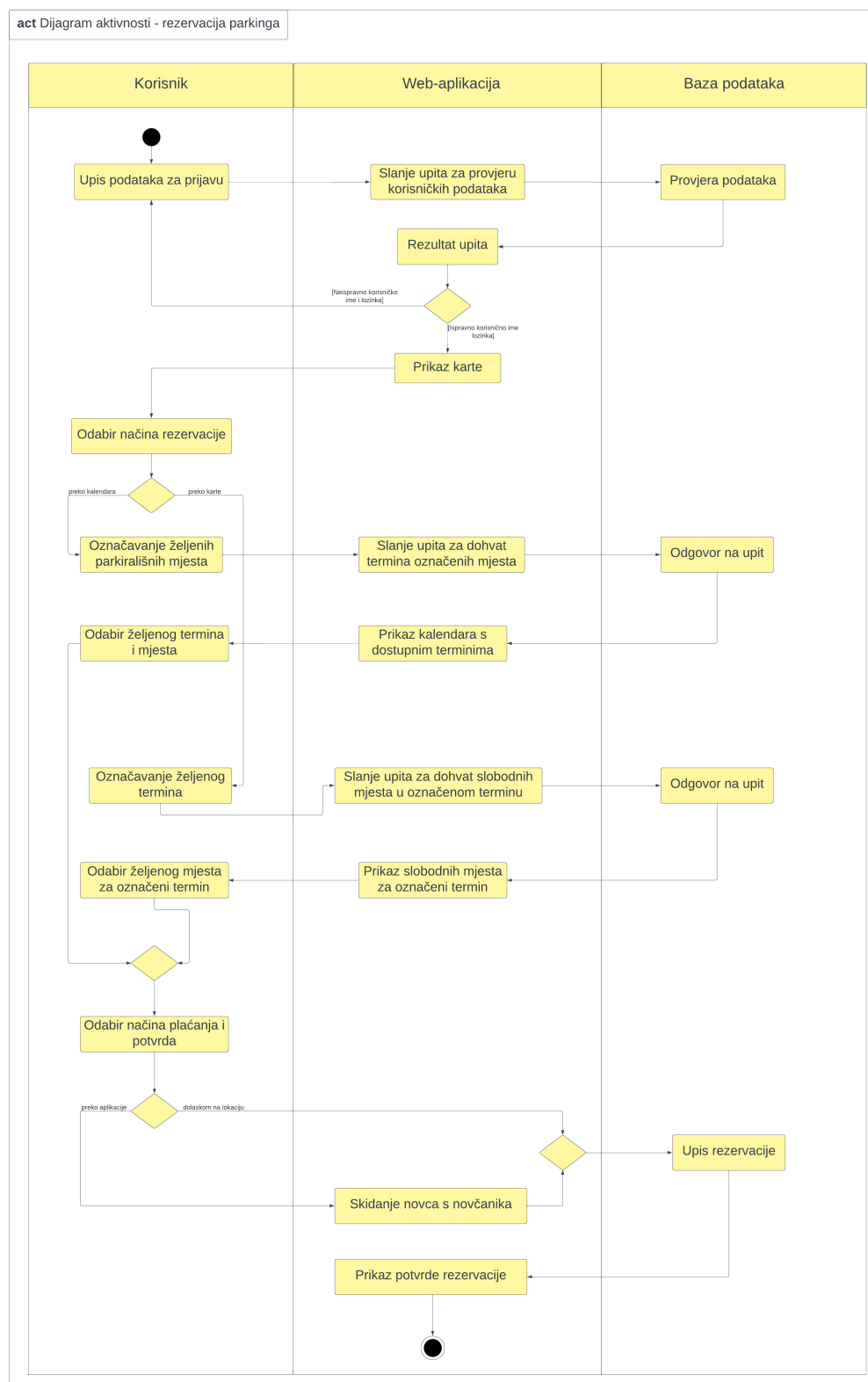


Slika 5.6: Dijagram stanja

5.4 dijagram aktivnosti

Na dijagramu aktivnosti 4.7. prikazan je proces kreiranja rezervacije parkinga. Korisnik se prijavi u sustav te na jedan od dva načina kreira rezervaciju. Prvi način

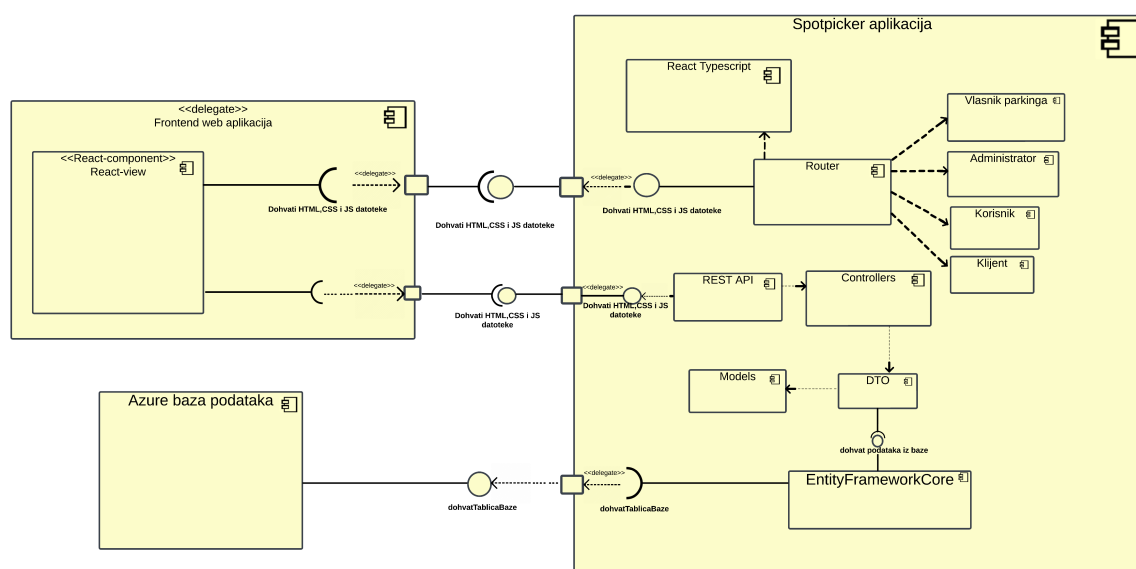
je da odabere željeni termin nakon čega mu se na karti prikazuju dostupna parkirališna mjesta u tom terminu između kojih onda odabire. Drugi način je da na karti odabere parkirališna mjesta za koja je zainteresiran, nakon čega mu se otvara kalendar s dostupnim terminima. Nakon toga korisnik odabire između plaćanja u aplikaciji i plaćanja na licu mjesta. Naposljetku mu se prikazuje potvrda rezervacije.



Slika 5.7: Dijagram aktivnosti

5.5 Dijagram komponenti

Na dijagramu komponenti 5.7. prikazan je vizualni prikaz arhitekture sustava kroz identifikaciju ključnih komponenata i njihovih međusobnih veza. Ovaj dijagram jasno prikazuje fizičke ili logičke module, njihove sučelja te način kako komponente surađuju radi ostvarenja funkcionalnosti sustava. Sustavu se pristupa preko dva različita sučelja. Preko *frontend* dijela aplikacije poslužuju se sve HTML, CSS i TYPESCRIPT datoteke. *Frontend* dio aplikacije ima i sučelje za slanje JSON datoteka aplikaciji preko REST API komponente, koja dalje te podatke poslužuje *backend* dijelu aplikacije. Pomoću EntityFrameworkCore komponente aplikacija pristupa podacima iz baze podataka, a ti pristigli podatci se šalju u obliku DTO (*Data transfer object*).



Slika 5.8: Dijagram komponenti

6. Implementacija i korisničko sučelje

6.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacije WhatsApp ¹. Za izradu UML dijagrama korišten je alat Astah Professional², a kao sustav za upravljanje izvornim kodom Git³. Udaljeni repozitorij projekta dostupan je na web platformi GitHub⁴.

Kao razvojno okruženje korišten je Microsoft Visual Studio ⁵- integrirano je razvojno sučelje (IDE) tvrtke Microsoft. Prvenstveno se koristi za razvoj računalnih programa za operacijski sustav Windows, kao i za web-stranice, web-aplikacije, web-usluge i mobilne aplikacije. Visual Studio za razvoj softvera koristi Microsoftove platforme kao što su windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight.

Aplikacija je napisana koristeći radni okvir jezik C# ⁶ za izradu backenda te React ⁷ i jezik JavaScript za izradu frontenda. React, također poznat kao React.js je biblioteka u JavaScriptu za izgradnju korisničkih sučelja. Održana je od strane Facebooka. React se najčešće koristi kao osnova u razvoju web ili mobilnih aplikacija. Složene aplikacije u Reactu obično zahtijevaju korištenje dodatnih biblioteka za interakciju s API-jem. Radni okvir Baza podataka se nalazi na poslužitelju u oblaku Microsoft Azure ⁸.

¹<https://www.whatsapp.com/>

²<https://astah.net/products/astah-professional/>

³<https://git-scm.com/>

⁴<https://github.com/>

⁵<https://visualstudio.microsoft.com/>

⁶<https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>

⁷<https://react.dev/>

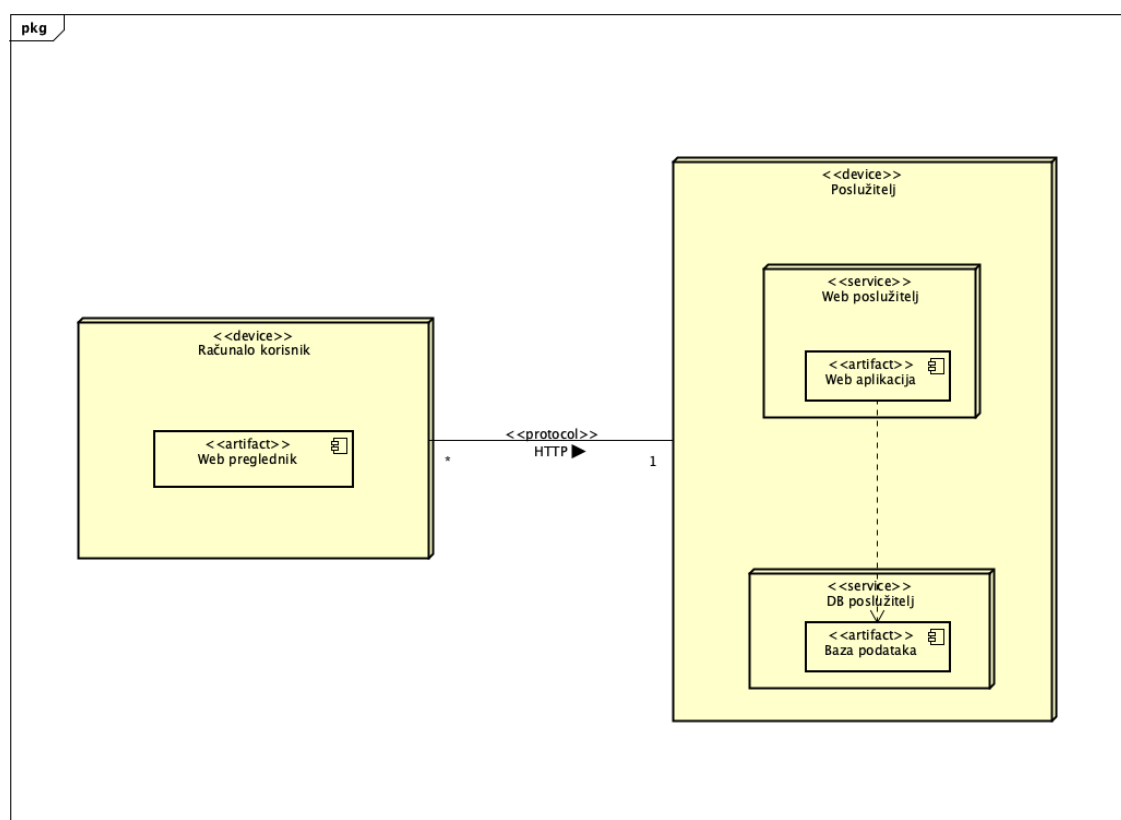
⁸<https://azure.microsoft.com/en-us>

6.2 Ispitivanje programskog rješenja

6.3 Dijagram razmještaja

UML-dijagrami razmještaja (engl. deployment diagrams) prikazuju fizičku arhitekturu programskog sustava, prikazujući razmještaj programskih artefakata na sklopovskim čvorovima ili na virtualnim okruženjima.

Naš se sustav bazira na "klijent – poslužitelj" arhitekturi, te se komunikacija između računala korisnika i poslužitelja odvija pomoću HTTP veze. Pristup web aplikaciji odvija se preko web preglednika računala korisnika. Na poslužitelju se nalaze web poslužitelj i poslužitelj baze podataka.



Slika 6.1: Dijagram razmještaja

6.4 Upute za puštanje u pogon

6.4.1 Instalacija poslužitelja baze podataka

Pokretanje baze podataka radili smo unutar razvojnog okruženja Microsoft Visual Studio. Pokretanjem debug procesa

7. Zaključak i budući rad

Projekt SpotPicker predstavlja sveobuhvatnu web aplikaciju koja pruža korisnicima mogućnost jednostavne rezervacije, naplate parkiranja te pregleda slobodnih parkirališnih mjesta za automobile i bicikle. Tijekom razvoja ovog sustava, uspješno smo ostvarili zadane ciljeve, pridržavajući se specifičnih funkcionalnosti i zahtjeva postavljenih u okviru projektnog zadatka.

U prvoj fazi projekta, usmjerili smo se na temeljito dokumentiranje zahtjeva, koristeći se UML dijagramima i modeliranjem. Ovi dokumenti, poput obrazaca uporabe, sekvencijskih dijagrama te dijagrama razreda, bili su od ključne pomoći u daljnjem razvoju sustava. Kvalitetna priprema u prvoj fazi značajno je olakšala implementaciju u drugoj fazi, pružajući jasan smjer svakom članu tima.

U drugoj fazi, članovi tima su samostalno radili na implementaciji, suočavajući se s izazovima te koristeći priliku za učenje novih alata i jezika. Kroz intenzivan rad, ostvarili smo funkcionalnosti poput rezervacije parkirališta, pregleda statistika, te integracije s OSRM API-jem za dohvat ruta. Osim toga, dovršili smo i sve potrebne UML dijagrame te izradili popratnu dokumentaciju.

Jedan od ključnih aspekata našeg projekta je briga o korisničkom iskustvu. Kroz jednostavno sučelje, korisnici imaju pristup svim potrebnim informacijama, mogu rezervirati parkirališta na različite načine, a voditelji parkirališta imaju uvid u statistike zauzetosti.

U budućnosti, mogli bismo proširiti funkcionalnosti aplikacije dodavanjem mobilne aplikacije, omogućujući korisnicima veću fleksibilnost i dostupnost. Također, kontinuirano ažuriranje sustava i praćenje povratnih informacija korisnika bit će ključno za održavanje i unaprjeđenje kvalitete aplikacije.

Suradnja u timu često može biti izazovna, ali smo kroz ovaj projekt naučili kako pravilno komunicirati, dijeliti zadatke i usklađivati različite perspektive. Osim toga, rješavanje izazova koji su se pojavili tokom projekta omogućilo nam je razvijanje kreativnih i analitičkih vještina.

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. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

4.1	Dijagram obrasca uporabe, funkcionalnost korisnika i klijenta . . .	16
4.2	Dijagram obrasca uporabe, funkcionalnost vlasnika	17
4.3	Dijagram obrasca uporabe, funkcionalnost administratora	17
4.4	Sekvencijski dijagram za UC1	18
4.5	Sekvencijski dijagram za UC2	19
5.1	Arhitektura sustava	21
5.2	ER dijagram baze podataka	26
5.3	Dijagram razreda - dio Controllers	27
5.4	Dijagram razreda - dio Data transfer objects	27
5.5	Dijagram razreda - dio Models	29
5.6	Dijagram stanja	30
5.7	Dijagram aktivnosti	32
5.8	Dijagram komponenti	33
6.1	Dijagram razmještaja	35

Dodatak: Prikaz aktivnosti grupe

7.1 Tablica aktivnosti

	Mario Olčar	Paula Močinić	Tomislav Marenić	Lucija Perković	Ivan Bušljeta	Lovro De-Villa	Matija Huđin
Upravljanje projektom	40						
Opis projektnog zadatka	5						
Funkcionalni zahtjevi					5		
Opis pojedinih obrazaca		4					
Dijagram obrazaca		4					
Sekvencijski dijagrami			5				
Opis ostalih zahtjeva	3						
Arhitektura i dizajn sustava		15					
Baza podataka					12		
Dijagram razreda					2		
Dijagram stanja						2	
Dijagram aktivnosti		1					
Dijagram komponenti			2				
Korištene tehnologije i alati	1						
Ispitivanje programskog rješenja							12

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Mario Olčar	Paula Močinić	Tomislav Marenčić	Lucija Perković	Ivan Bušljeta	Lovro De-Villa	Matija Hudin
Dijagram razmještaja			2				
Upute za puštanje u pogon							1
Dnevnik sastajanja	2						
Zaključak i budući rad		3					
Popis literature	1						
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>		12					
<i>izrada baze podataka</i>					3	5	7
<i>spajanje s bazom podataka</i>		2					3
<i>back end</i>	4				4	7	8