

Programsko inženjerstvo

Ak. god. 2023./2024.

SpotPicker

Dokumentacija, Rev. 1

Grupa: *Lukax4*

Voditelj: *Luka Dikić*

Datum predaje: 17. 11. 2023.

Nastavnik: *Hrvoje Nuić, mag. ing.*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
3	Specifikacija programske potpore	10
3.1	Funkcionalni zahtjevi	10
3.1.1	Obrasci uporabe	12
3.2	Dijagram obrazaca uporabe	18
3.2.1	Sekvencijski dijagrami	20
3.3	Ostali zahtjevi	22
4	Arhitektura i dizajn sustava	23
4.1	Baza podataka	26
4.1.1	Opis tablica	26
4.1.2	Dijagram baze podataka	30
4.2	Dijagram razreda	30
4.3	Dijagram stanja	33
4.4	Dijagram aktivnosti	34
4.5	Dijagram komponenti	35
4.6	Dijagram razreda	35
4.7	Dijagram stanja	38
4.8	Dijagram aktivnosti	39
4.9	Dijagram komponenti	40
5	Implementacija i korisničko sučelje	47
5.1	Korištene tehnologije i alati	47
5.2	Ispitivanje programskog rješenja	48
5.2.1	Ispitivanje komponenti	48
5.2.2	Ispitivanje sustava	48
5.3	Dijagram razmještaja	49
5.4	Upute za puštanje u pogon	50

6 Zaključak i budući rad	51
Popis literature	52
Indeks slika i dijagrama	53
Dodatak: Prikaz aktivnosti grupe	54

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	*	25.10.2023.
0.2	Dopisane upute za povijest dokumentacije. Dodane reference.	*	24.08.2013.
0.5	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	*	26.10.2023.
0.6	Arhitektura i dizajn sustava, algoritmi i strukture podataka	*	27.10.2013.
0.8	Povijest rada i trenutni status implementacije, Zaključci i plan daljnjeg rada	*	2.11.2023.
0.9	Opisi obrazaca uporabe	*	07.11.2023.
0.10	Preveden uvod	*	10.11.2023.
0.11	Sekvencijski dijagrami	*	15.11.2023.
0.12.2	Napravljen dijagrama razreda	*	16.11.2023.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	17.11.2023.

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

0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.

2. Opis projektnog zadatka

Cilj projekta je napraviti web aplikaciju “SpotPicker” koja bi omogućila vozačima osobnih automobile i bicikla da unaprijed rezerviraju svoje parkirno mjesto u garaži i time se riješe svakodnevne muke lutanja po parkingu i traženja mjesta.

S korisničke strane, aplikacija se otvara u neregistriranom obliku. Neregistrirani korisnik može pregledavati parkirališta i njihova parkirna mjesta, ali bez informacije o dostupnosti. Zato se novim korisnicima nudi mogućnost registracije, odnosno izrade korisničkog računa i mogućnost prijave za one koji već imaju izrađen račun. Za registraciju su potrebni:

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

Prilikom prve registracije potrebno je potvrditi svoje podatke preko poruke poslane na prethodno unesenu e-mail adresu. Za prijavu su potrebni samo korisničko ime i lozinka. Najvažnija dodatna mogućnost koju aplikacija

nudi prijavljenim korisnicima je pregled dostupnih parkirnih mjesta u realnom vremenu. Puni postupak rezervacije parkirnog mjesta izgleda ovako:

1. korisnik na karti grada bira lokaciju do koje želi doći
2. aplikacija mu na temelju njegovog odredišta bira najbliži parking

Sljedeći se korak može odviti na dva načina:

3. a) korisnik bira parkirna mjesta koja mu se sviđaju, a zatim se otvara kalendar u kojem može točno odabrati termin rezervacije (datum i vrijeme)
3. b) korisnik bira termin rezervacije (datum i vrijeme), a zatim mu se prikazuju parkirna mjesta koja će tada biti dostupna

Ovdje je bitno napomenuti da i u a) i u b) varijanti najranija moguća rezervacija je dan nakon onog u kojem korisnik vrši rezervaciju. Znači nije moguće rezervirati na isti dan.

4. korisnik sada dobiva dodatnu mogućnost odabira želi li da mu rezervacija bude ponavljajuća ili ne. To je pogotovo pogodno ljudima koji znaju da imaju neke obaveze na npr. tjednoj bazi
5. korisniku se daje opcija plaćanja karticom unaprijed ili plaćanje prilikom dolaska na parking
6. potvrđuje se rezervacija parkirnog mjesta, termina, trajanja i načina plaćanja

Korisniku se pruža mogućnost plaćanja direktnim prijenosom sredstava s kartice ili uplaćivanjem novca u novčanik unutar aplikacije nakon kojeg se sredstva mogu

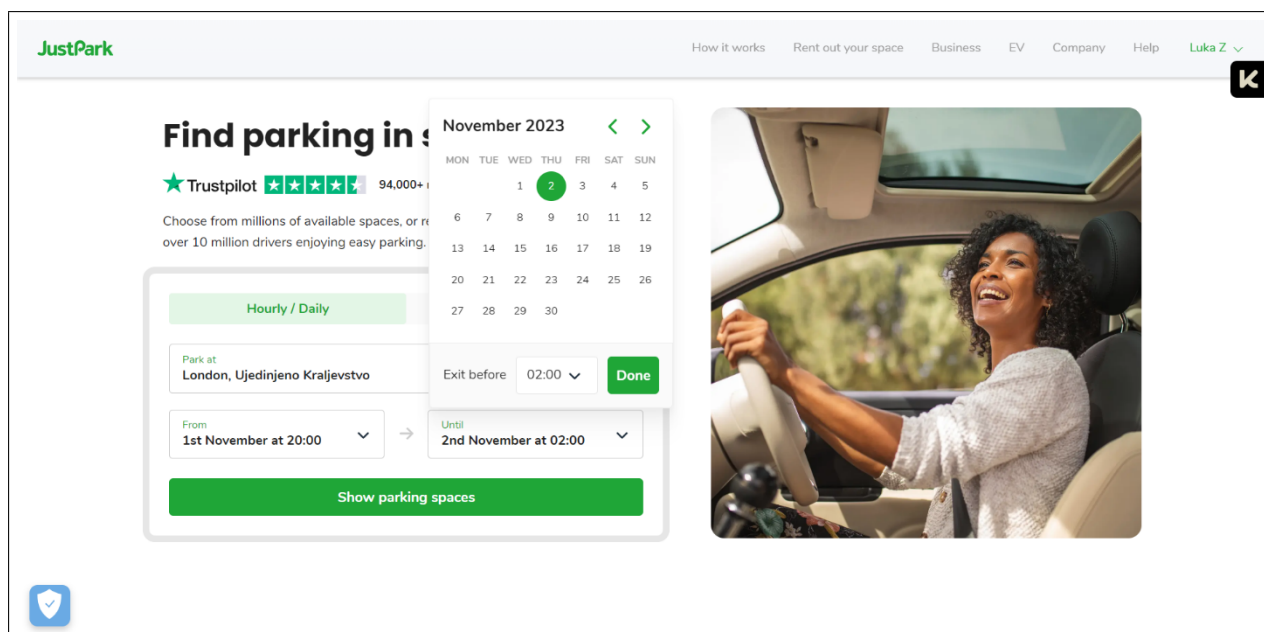
koristiti u bilo kojem trenutku. Drugi način štedi vrijeme jer nije potrebno unositi podatke o kartici i vršiti potvrde prilikom svake transakcije.

Šta se tiče parkirnih mjesta za bicikle, njih nije potrebno rezervirati jer se ne naplaćuju. Moguće je jedino pratiti u realnom vremenu ukupan broj dostupnih mjesta.

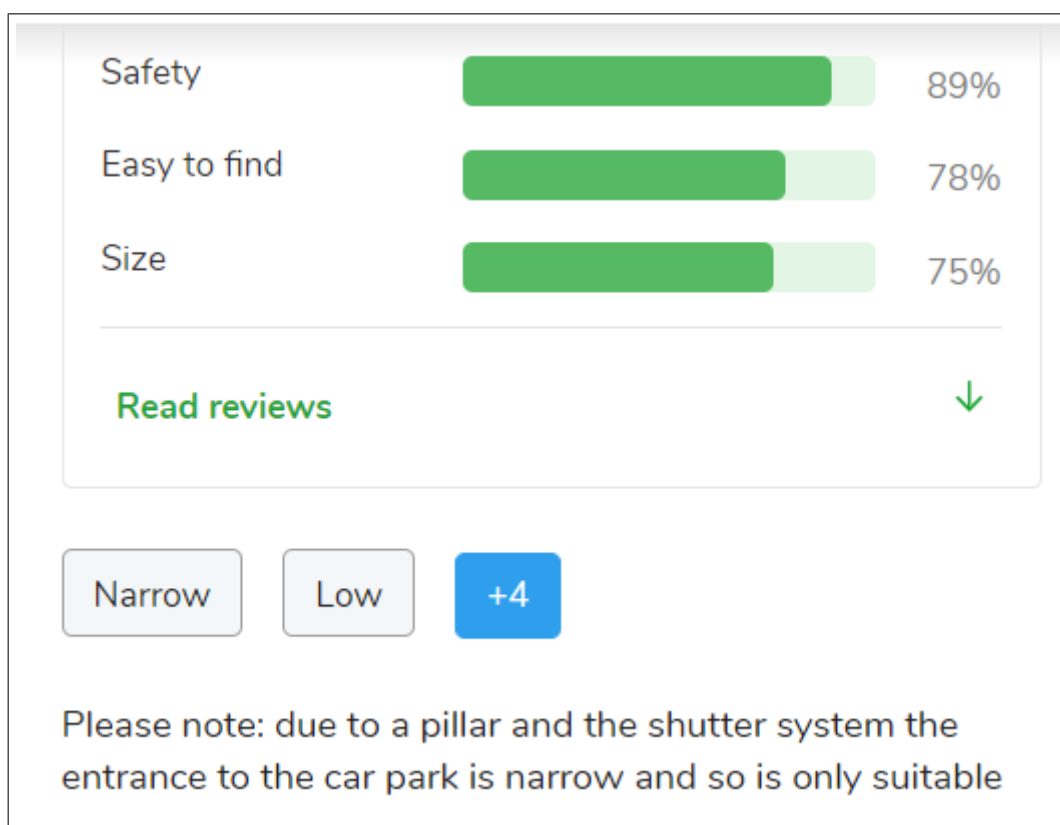
U slučaju da se korisnik prijavljuje kao vođa parkinga, da bi dobio vođa parkinga prvo je potrebno odobrenje administratora. Vođa parkinga ima mogućnost promjene podataka o svom parkingu. To uključuje izmjenu imena parkinga, dodavanje i uklanjanje slika te promjene cijena. Također može dodavati nova mjesta u slučaju proširenja parkinga.

Uloga administratora donosi pravo pregledavanja i mijenjanja osobnih podataka registriranih korisnika. Također, on potvrđuje identitet vođe parkinga prilikom prijave u aplikaciju.

Slično programsko rješenje nudi aplikacija JustPark koja omogućuje rezervaciju parkirnih mjesta u Ujedinjenom Kraljevstvu. Glavno sučelje (slika1) nudi jednostavnu mogućnost odabira lokacije i termina. Zatim se otvara popis parkirališta sortiranih po udaljenosti koju treba pješice prijeći od parkinga do finalne destinacije. SpotPicker se razlikuje od JustParka po tome šta JustPark nudi ostavljanje komentara i ocijenjivanje parkinga (slika2) po raznim kriterijima poput sigurnosti, lakoće snalaženja, širine mjesta itd.



Slika 2.1: Sučelje za odabir termina u JustPark



Slika 2.2: Dodatna mogućnost ocjene parkinga u JustPark

SpotPicker je aplikacija savršena za ljude koji planiraju svoja putovanja ili obaveze po nekoliko dana unaprijed jer će si rezervacijom mjesta ukloniti onaj najgori dio vožnje, a to je lutanje u potrazi za mjestom. Također, pogodno je ljudima koji moraju negdje doći na vrijeme jer ovako mogu značajno bolje procijeniti koliko im je vremena potrebno u vožnji. Isto tako, aplikaciju mogu koristiti i ljudi kojima je parking potreban stalno (npr. jer rade u blizini) jer je moguće da rezervacija bude ponavljajuća iz tjedna u tjedan.

Naravno, bitno je napomenuti da je ovaj projekt moguće nadograditi i poboljšati neke njegove stavke. Tako bi korisni dodatak bio mogućnost poništavanja rezervacije uz povrat novca.

Šta se tiče prilagodbe raznim tržištima, za manje gradove koji imaju puno privatnih kuća u centru, bila bi dobra mogućnost da se vlasnici kuća s dvorištem mogu prijaviti da iznajmljuju svoje dvorište kao parkirno mjesto. Vlasnik dvorišta bi se jednako vodio kao i vlasnik većeg parkirališta, a odabir mjesta bi djelovao na isti način kao i kod velikih parkinga ovisno o veličini dvorišta. Bitno je primijetiti da projekt uključuje rješenje za parkirališta s jednom razinom, tako da bi jedno dodatno moguće proširenje bilo dodati mogućnost odabira kata garaže i tek onda parkirnog mjesta.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Korisnik
2. Administrator
3. Vlasnik parkirališta(naručitelj)
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) na karti pregledati lokacije parkirališta i tlocrt parkirališta
 - (b) stvoriti korisnički račun, tj. registrirati se za šta su mu potrebni korisničko ime, lozinka, ime, prezime, slika osobne iskaznice, IBAN i email adresa
2. Registrirani korisnik (sudionik) može:
 - (a) prijaviti se u sustav koristeći korisničko ime i lozinku
 - (b) na karti odabrati adresu do koje želi doći
 - (c) na karti pregledati lokacije parkirališta, tlocrt parkirališta i dostupna parkirna mjesta
 - (d) rezervirati parkirno mjesto, termin i trajanje za koje je zainteresiran
 - (e) platiti rezervaciju parkinga
3. Vlasnik parkinga (sudionik) može:
 - (a) unijeti i mijenjati informacije o svom parkiralištu (naziv, opis, fotografija, cjenik)
 - (b) dodavati nova parkirna mjesta
 - (c) vidjeti statistiku zauzetosti parkirališta i parkirališnih mjesta kroz vrijeme

4. Administrator (sudionik) može:

- (a) pregledavati osobne podatke registriranih korisnika
- (b) mijenjati osobne podatke registriranih korisnika
- (c) potvrditi prijavu korisnicima koji se prijavljuju kao vlasnik parkinga

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - registracija korisnika

- **Glavni sudionik:** neregistrirani korisnik
- **Cilj:** izraditi korisnički račun
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. korisnik odabire opciju za registraciju
 2. korisnik upisuje potrebne podatke
 3. u bazu podataka se upisuje novi korisnik
 4. web stranica se preusmjerava na stranicu za prijavu
- **Opis mogućih odstupanja:**
 - 2.a korisničko ime i/ili email su već zauzeti
 1. korisnik dobiva povratnu informaciju o specifičnoj greški i preusmjeren je nazad na stranicu za registraciju
 - 2.b upisana lozinka nije važeća(nema barem 8 znakova ili ne sadrži barem jedno slovo i brojku)
 1. korisnik dobiva povratnu informaciju o specifičnoj greški i preusmjeren je nazad na stranicu za registraciju

UC2 -prijava korisnika

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** prijava u korisnički račun
- **Sudionici:** baza podataka
- **Preduvjet:** prethodna registracija
- **Opis osnovnog tijeka:**
 1. korisnik odabire opciju za prijavu
 2. korisnik upisuje svoje korisničko ime i lozinku
 3. korisnik je preusmjeren na početnu stranicu sa svim ovlastima prijavljenog korisnika
- **Opis mogućih odstupanja:**
 - 2.a upisana kombinacija korisničkog imena i lozinke je neispravna
 1. korisnik je vraćen na stranicu za prijavu

UC3-promjena lozinke

- **Glavni sudionik:** registrirani korisnik
- **Cilj:** izmijeniti lozinku na već postojećem korisničkom računu
- **Sudionici:** baza podataka
- **Preduvjet:** prethodna registracija
- **Opis osnovnog tijeka:**
 1. korisnik izabire opciju za mijenjanje lozinku
 2. korisnik upisuje email adresu na koju želi da mu se pošalje verifikacijski kod
 3. korisnik upisuje verifikacijski kod
 4. korisnik upisuje novu lozinku i potvrđuje ju ponovnim upisom
 5. korisnik je preusmjeren na stranicu za prijavu
- **Opis mogućih odstupanja:**
 - 3.a upisani verifikacijski kod nije isti onom poslanom na email
 1. dolazi poruka o grešci i korisnik može opet upisati kod
 - 4.a lozinke se ne poklapaju
 1. dolazi poruka o grešci i korisnik može opet upisati loznike
 - 4.b upisana lozinka nije važeća(nema barem 8 znakova ili ne sadrži barem jedno slovo i brojku)
 1. dolazi poruka o grešci i korisnik može opet upisati loznike

UC4-odabir parkirališta

- **Glavni sudionik:** prijavljeni korisnik
- **Cilj:** izabrati parkiralište na kojem će kasnije rezervirati mjesto
- **Sudionici:** sudionici
- **Preduvjet:** prethodna prijava
- **Opis osnovnog tijeka:**
 1. korisnik upisuje adresu do koje želi doći
 2. korisnik klikne gumb za pretraživanje
 3. korisniku se na karti iscrta ruta do parkirališta koje je najbliže njegovom odredištu

UC5.1 -rezervacija mjesta s uvjetom određenog mjesta

- **Glavni sudionik:** prijavljeni korisnik
- **Cilj:** rezervirati specifično parkirno mjesto

- **Sudionici:** sudionici
- **Preduvjet:** odabrano parkiralište
- **Opis osnovnog tijeka:**
 1. korisnik klikne na ikonu parkirališta
 2. na karti parkirališta odabire mjesto za koje je zainteresiran
 3. u kalendaru izabire datum
 4. korisniku se prikazuju slobodni sati za odabrano mjesto i datum
 5. korisnik bira vremensko razdoblje
 6. korisnik potvrđuje rezervaciju
 7. web stranica se preusmjerava na stranicu za plaćanje
- **Opis mogućih odstupanja:**
 - 4.a sva vremena za taj datum i mjesto su zauzeti
 1. korisnik sam mora odabrati neki drugi datum ili mjesto

UC5.2-rezervacija mjesta s uvjetom datuma

- **Glavni sudionik:** prijavljeni korisnik
- **Cilj:** rezervirati mjesto u specifičnom terminu
- **Sudionici:** sudionici
- **Preduvjet:** odabrano parkiralište
- **Opis osnovnog tijeka:**
 1. korisnik klikne na ikonu parkirališta
 2. korisnik u kalendaru odabire datum i vrijeme početka parkiranja
 3. korisnik u kalendaru odabire datum i vrijeme završetka parkiranja
 4. korisniku se prikažu dostupna mjesta za odabrani termin na karti parkirališta
 5. korisnik odabire mjesto na karti
 6. korisnik potvrđuje rezervaciju
 7. web stranica se preusmjerava na stranicu za plaćanje
- **Opis mogućih odstupanja:**
 - 4.a niti jedno mjesto se ne prikazuje kao dostupno
 1. korisnik mora odabrati drugi termin

UC6.1-plaćanje prilikom dolaska

- **Glavni sudionik:** prijavljeni korisnik
- **Cilj:** plaćanje parkinga prilikom dolaska na lokaciju
- **Sudionici:** baza podataka

- **Preduvjet:** registracija korisnika
- **Opis osnovnog tijeka:**
 1. Korisnik dolazi rezervira parkirališno mjesto
 2. Korisnik dolazi na lokaciju parkirališnog mjesta
 3. Korisnik plaća rezervirano mjesto

UC6.2-plaćanje prilikom rezervacije

- **Glavni sudionik:** prijavljeni korisnik
- **Cilj:** plaćanje karticom prilikom rezervacije
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika
- **Opis osnovnog tijeka:**
 1. Korisnik odabire parkirališno mjesto i slobodan termin
 2. Korisnik odabire mogućnost plaćanja karticom prilikom rezervacije
 3. Korisnik unosi podatke s kartice
 4. Nakon uspješnog plaćanja mjesto je rezervirano
- **Opis mogućih odstupanja:**
 - 2.a odbijena kartica
 1. pojavljuje se poruka o nevaljanosti kartice

UC6.3-plaćanje sredstvima u novčaniku aplikacije

- **Glavni sudionik:** prijavljeni korisnik
- **Cilj:** plaćanje parkirnog mjesta sredstvima unutar aplikacije
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika, dodavanje novaca u novčanik
- **Opis osnovnog tijeka:**
 1. Korisnik sa svog računa uplaćuje željeni iznos u novčanik aplikacije
 2. Korisnik prilikom rezervacije odabire način plaćanja sredstvima iz novčanika aplikacije
 3. Nakon uspješnog plaćanja mjesto je rezervirano
- **Opis mogućih odstupanja:**
 - 2.a nedovoljno sredstva u novčaniku
 1. pojavljuje se poruka o nedovoljnoj količini sredstava u novčaniku

UC7-dodavanje novaca u novčanik aplikacije

- **Glavni sudionik:** prijavljeni korisnik

- **Cilj:** korištenje sredstava novčanika aplikacije za plaćanje
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za dodavanje sredstava u novčanik aplikacije
 2. Korisnik upisuje podatke kartice te iznos koji želi uplatiti
 3. Nakon uspješne uplate stanje u novčaniku aplikacije se ažurira

UC8-izmjena osobnih podataka korisnika

- **Glavni sudionik:** administrator
- **Cilj:** izmjena osobnih podataka postojećih korisnika
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregled korisnika
 2. Administrator otvara željeni profil
 3. Administrator odabire opciju uredi
 4. Administrator izmjenjuje podatke
 5. Promjene se ažuriraju u bazi podataka

UC9-potvrđivanje prijave vlasniku

- **Glavni sudionik:** administrator
- **Cilj:** potvrda uspješne prijave vođitelju
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregled korisnika
 2. Administrator otvara željeni profil
 3. Administrator odabire opciju potvrdi vođitelja
 4. Promjene se ažuriraju u bazi podataka

UC10-brisanje korisnika

- **Glavni sudionik:** administrator
- **Cilj:** izmjena osobnih podataka postojećih korisnika
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika

- **Opis osnovnog tijeka:**

1. Administrator odabire opciju pregled korisnika
2. Administrator otvara željeni profil
3. Administrator odabire opciju uredi
4. Administrator izmjenjuje podatke
5. Promjene se ažuriraju u bazi podataka

UC11-unos podataka o parkiralištu

- **Glavni sudionik:** voditelj parkinga
- **Cilj:** unos podataka o parkirnom mjestu
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika, potvrda uloge voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj otvara popis parkirališnih mjesta
 2. Voditelj odabire željeno parkirališno mjesto
 3. Voditelj odabire opciju uredi
 4. Voditelj unosi naziv, opis, fotografiju, cijenu itd
 5. Promjene se ažuriraju u bazi podataka

UC12-dodavanje dostupnosti parkirnog mjesta

- **Glavni sudionik:** voditelj parkinga
- **Cilj:** ucrtava dostupna parkirališna mjesta u kartu
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika, potvrda uloge voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj otvara kartu parkirališnih mjesta
 2. Voditelj odabire željeno parkirališno mjesto
 3. Voditelj odabire opciju dostupno
 4. Promjene se ažuriraju u bazi podataka

UC13-micanje dostupnosti parkirnog mjesta

- **Glavni sudionik:** voditelj parkinga
- **Cilj:** : ucrtava dostupna parkirališna mjesta u kartu
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika, potvrda uloge voditelja
- **Opis osnovnog tijeka:**

1. Voditelj otvara kartu parkirališnih mjesta
2. Voditelj odabire željeno parkirališno mjesto
3. Voditelj odabire opciju nedostupno
4. Promjene se ažuriraju u bazi podataka

UC14-pregledavanje statistike za parkiralište

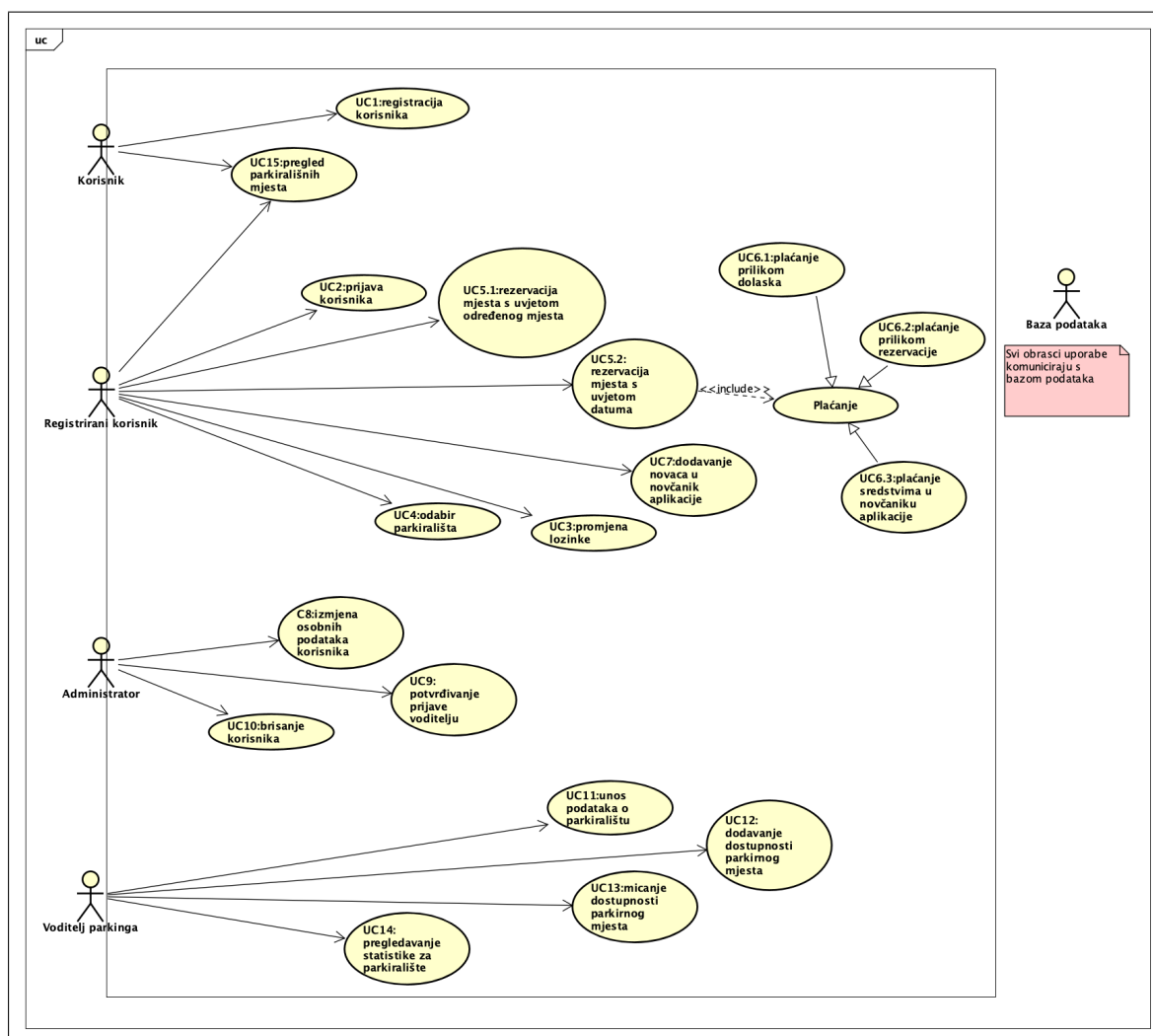
- **Glavni sudionik:** voditelj parkinga
- **Cilj:** pregled statistike zauzetosti parkirališnog mjesta
- **Sudionici:** baza podataka
- **Preduvjet:** registracija korisnika, potvrda uloge voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj otvara kartu parkirališnog mjesta
 2. Voditelj odabire željeno parkirališno mjesto
 3. Voditelj odabire opciju statistika
 4. Prikazuje se statistika zauzetosti parkirališnog mjesta u obliku gafa

UC15 -pregled parkirališnih mjesta

- **Glavni sudionik:** korisnik
- **Cilj:** pregled parkirališnih mjesta i njihova dostupnost
- **Sudionici:** baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire kartu parkirališta
 2. Otvara se karta sa prikazom dostupnih mjestas

3.2 Dijagram obrazaca uporabe

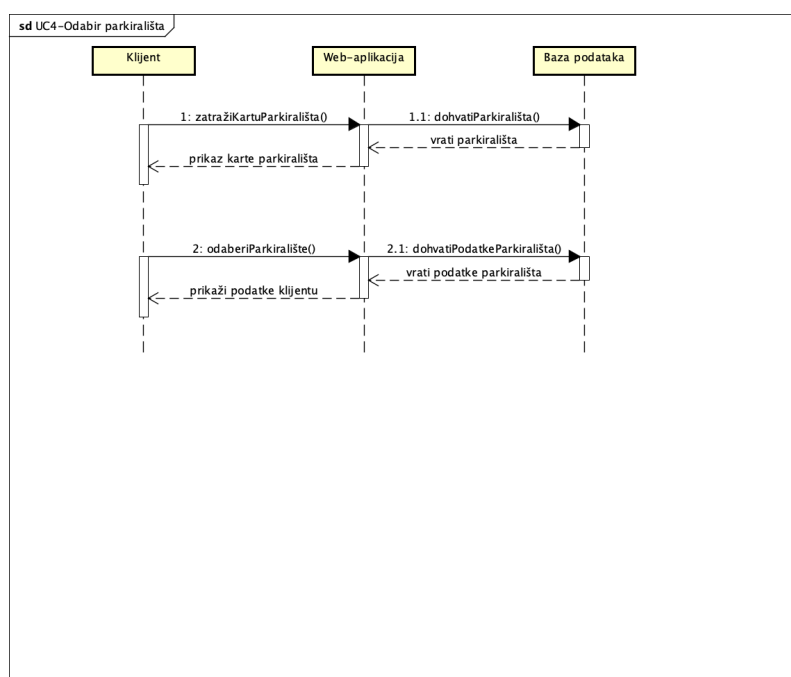
Na slici je prikazan dijagram obrazaca uporabe točnije funkcionalnosti korisnika, registriranog korisnika, administratora te voditelja parkinga.



3.2.1 Sekvencijski dijagrami

Obrazac uporabe UC4 - Odabir parkirališta

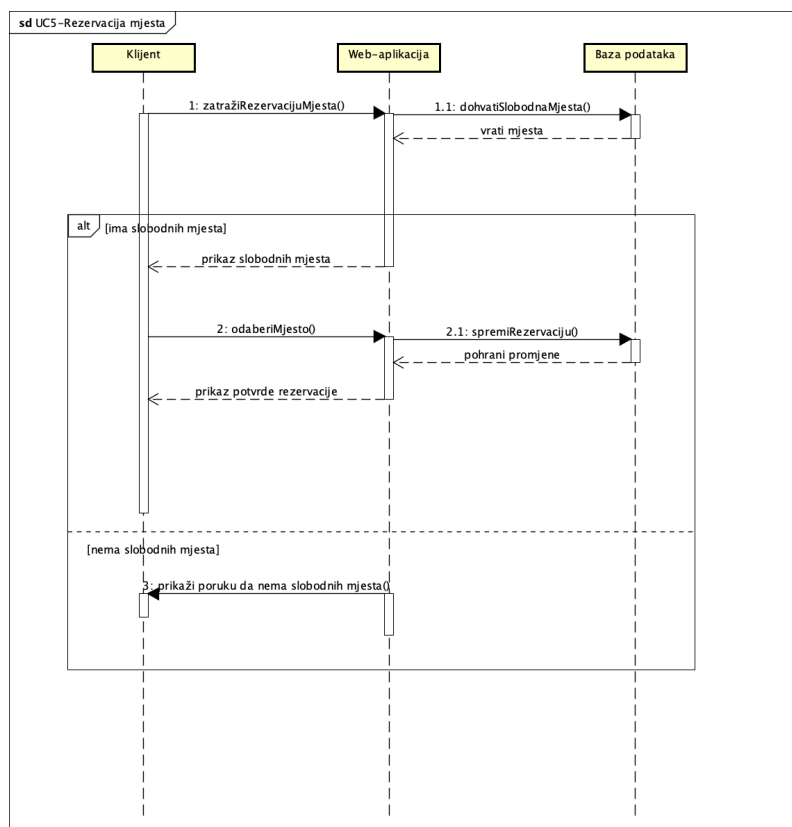
Klijent, nakon što pokrene aplikaciju, šalje zahtjev za pregled svih dostupnih parkirališta. Aplikacija, koristeći bazu podataka, dohvaća popis parkirališta te ih prikazuje korisniku. Nakon pregleda, klijent odabire specifično parkiralište koje ga zanima. Aplikacija zatim prikazuje dodatne informacije o odabranom parkiralištu, uključujući naziv, opis, fotografiju, cjenik i slobodna parkirališna mjesta. Dodatno, klijent može pregledati dostupna parkirališna mjesta na karti povezanoj s odabranim parkiralištem, a ovisno o odabiru, informacije o zauzetosti pojedinih mjesta



Slika 3.1: Sekvencijski dijagram za UC4

Obrazac uporabe UC5 - Rezervacija mjesta

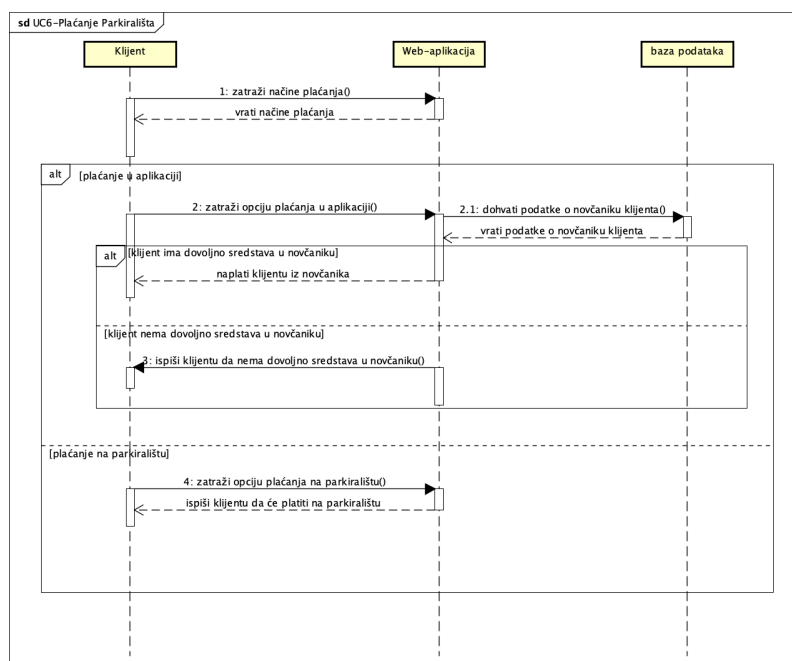
Klijent, nakon što je odabrao željeno parkiralište, pristupa informacijama o dostupnim parkirališnim mjestima. Aplikacija mu prikazuje kartu parkirališta s označenim slobodnim i zauzetim mjestima. Tada klijent šalje upit za odabranim mjestom, a aplikacija provjerava raspoloživost tog mjesta. Ukoliko je mjesto slobodno, klijentu se potvrđuje rezervacija, a sustav ažurira informacije o zauzetosti i evidentira rezervaciju. U slučaju da mjesta nisu slobodna, aplikacija obavještava klijenta o nedostupnosti te ga vraća na korak odabira novih mjesta ili termina.



Slika 3.2: Sekvencijski dijagram za UC5

Obrazac uporabe UC6 - Plaćanje parkirališta

Kada klijent odluči platiti parkiranje, sustav prikazuje opcije plaćanja prilikom rezervacije ili prilikom dolaska na lokaciju parkirališta. Klijent odabire željeni način plaćanja. Ako odabere plaćanje unutar aplikacije, tada aplikacija provjerava stanje novčanika. Rezervacija će se naplatiti ako klijent ima dovoljno sredstava u novčaniku, a u protivnom će mu se prikazati da treba nadoplatiti svoj novčanik. Ako se odluči za opciju plaćanja parkinga na parkiralištu, tada će mu aplikacija to prihvatiti i potvrditi.

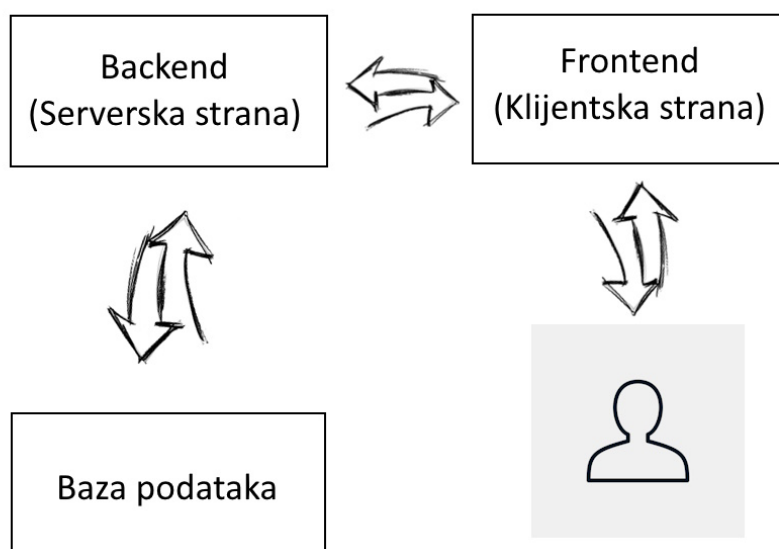


Slika 3.3: Sekvencijski dijagram za UC6

3.3 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.
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava.
- Sustav treba biti jednostavan za korištenje; korisnici se moraju znati koristiti sučeljem bez opsežnih uputa.
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava.
- Sustav kao valutu koristi evro.

4. Arhitektura i dizajn sustava



Slika 4.1: Arhitektura sustava

Korisnik aplikaciji pristupa putem web preglednika. Interakciju s aplikacijom ostvaruje preko korisničkog sučelja pomoću kojeg šalje zahtjeve web poslužitelju i prima odgovore.

Programski jezik pomoću kojeg je ostvaren backend web aplikacije je C#, a korišteni radni okvir je .NET. Frontend aplikacije ostvaren je programskim jezikom TypeScript i radnim okvirom Angular. Za razvojno okruženje backend-a odabran je Visual Studio, a frontend-a Visual Studio Code.

.NET je radni okvir namijenjen stvaranju mikroservisa. Mikroservisi su arhitektura razvoja softvera koja podrazumijeva izgradnju aplikacije pomoću niza malih, nezavisnih servisa. Svaki mikroservis obavlja određeni posao i može komunicirati s drugim mikroservisima preko standardiziranih protokola. Primjena mikroservisne arhitekture u razvoju web aplikacija omogućuje agilniji pristup razvoju, povećava skalabilnost, pojednostavljuje održavanje i prilagodljivost sustava promjenama. Svaki mikroservis može se smatrati zasebnim servisom unutar web

aplikacije, obavljajući specifične funkcionalnosti, poput autentifikacije, upravljanja korisnicima, poslovnih logika, itd.

Arhitektura sustava temelji se na MVC (Model-View-Controller) konceptu. MVC koncept predstavlja arhitekturni obrazac koji je podržan od strane .NET radnog okvira, pružajući gotove predloške koji značajno olakšavaju razvoj web aplikacija.

MVC koncept omogućuje nezavisan razvoj pojedinih dijelova aplikacije, što rezultira jednostavnijim ispitivanjem i olakšanim dodavanjem novih svojstava u sustav. Takav pristup razdvaja sustav na tri ključna dijela:

- **Model:** Središnja komponenta sustava koja predstavlja dinamičke strukture podataka neovisne o korisničkom sučelju. Model izravno upravlja podacima, logikom i pravilima aplikacije. Također, prima ulazne podatke od Controllera.
- **View:** Odgovoran je za prikaz podataka, poput grafa ili tabličnog prikaza. View omogućuje različite prikaze istih informacija, poput grafičkog ili tabličnog prikaza podataka.
- **Controller:** Prima ulaze od korisnika i prilagođava ih za prosljeđivanje Modelu ili Viewu. Controller upravlja korisničkim zahtjevima i temeljem njih izvodi daljnju interakciju s ostalim elementima sustava. Osim posredovanja između Modela i Viewa, Controller igra ključnu ulogu u usmjeravanju tijeka aplikacije.

Ovaj trojni raspored omogućuje jasno definiranje odgovornosti svake komponente, čime se postiže modularnost sustava. Model upravlja podacima, View se brine o prezentaciji, dok Controller koordinira interakcijom između njih. Ovaj pristup često rezultira čistim, lako održivim kodom i olakšava daljnji razvoj aplikacije.

Web aplikaciju čine tri osnovna dijela:

- **Frontend**
- **Backend**
- **Baza podataka**

Frontend se sastoji od komponenata i logike. Istu komponentu je moguće koristiti za različite namjene (engl. reusability). Angular koristi koncept komponenata

i servisa koji omogućuje ponovnu upotrebu koda. Angular koristi svojstva poput Dependency Injection i Change Detection za poboljšanje performansi. Struktura je ostvarena povezivanjem različitih komponenata unutar hijerarhijske strukture.

Backend je ključna komponenta web aplikacije, sastavljena od sljedećih dijelova:

- **Programsko sučelje za reprezentacijski prijenos stanja (REST API):** Controller je odgovoran za obradu zahtjeva koji dolaze od korisnika ili drugih dijelova aplikacije. On prima HTTP zahtjeve, interpretira ih, te ih prosljeđuje odgovarajućim dijelovima aplikacije. Nakon obrade zahtjeva, Controller šalje odgovor korisniku, koji se sastoji od statusnog koda, tijela poruke i zaglavlja. Sadržaj u tijelu poruke predstavlja informacije koje korisnik konzumira nakon što su prikazane u web pregledniku.
- **Sloj poslovne logike (Service):** Service je odvojeni sloj koji sadrži poslovnu logiku aplikacije. Ovdje se obrađuju i provjeravaju podaci, izvršavaju poslovne operacije te se komunicira s ostalim dijelovima sustava. Service često služi kao posrednik između Controllera i Repository sloja. Komunikaciju sa slojem poslovne logike Controller ostvaruje pomoću umetanja ovisnosti (dependency injection). Dependency injection je obrazac prema kojemu se u određeni objekt/funkciju umeće neki drugi objekt/funkcija na koji se prvobitno spomenuti objekt/funkcija oslanja.
- **Sloj za pristup bazi podataka (Repository):** Repository je odgovoran za komunikaciju s bazom podataka. Ovaj sloj omogućuje izvođenje operacija poput čitanja, pisanja, brisanja i ažuriranja podataka u bazi. Koristi se SQL upitima ili ORM (Object-Relational Mapping) tehnikama za pretvaranje podataka iz relacijske baze podataka u objekte koji se koriste unutar aplikacije. Repository omogućuje komunikaciju s bazom podataka pomoću SQL-a. Objekti iz relacijske baze podataka pretvaraju se u objekte programskog jezika Java pomoću tehnike ORM.

Ova trostruka podjela omogućuje jasnu strukturu i odvajanje odgovornosti unutar backend-a, čineći ga modularnim i lakšim za održavanje.

4.1 Baza podataka

Baza podataka za aplikaciju implementirana je kao relacijska baza podataka, gdje se podaci pohranjuju u obliku redaka ili n-torki te stupaca ili atributa koji zajedno čine tablicu. Ovaj pristup omogućuje strukturiranu organizaciju podataka, gdje svaka tablica predstavlja određenu vrstu informacija, a svaki redak u tablici predstavlja konkretan zapis ili entitet. Relacijski model olakšava učinkovito upravljanje podacima, omogućuje jednostavno pretraživanje i izvođenje kompleksnih upita nad podacima, što čini bazu podataka temeljem za pouzdano čuvanje i manipulaciju informacijama u aplikaciji.

4.1.1 Opis tablica

Glavna komponenta baze je tablica User, koja se puni osobnim podacima unesenim pri registraciji korisnika u sustav. Svakom novom korisniku dodjeljuje se primarni ključ, unikatni identifikacijski broj pomoću kojeg se korisnici u bazi podataka međusobno razlikuju. Budući da korisnik može biti primatelj usluge, davatelj usluge i menadžer usluge, postoji atribut roleId koji ovisno o broju označava ulogu korisnika u sustavu. Ako je korisnik menadžer, onda je povezan na tablicu Manager preko stranog ključa userId. Tablica Manager dodatno sadrži atribut confirmedByAdmin koji opisuje je li menadžer potvrđen od strane administratora aplikacije. Kako bi korisnik opće rezervirao parkirno mjesto, postoji tablica Parking koja sadrži svoj unikatni id, id vlasnika. Svako parkiralište ima n parkirališnih mjesta koja su opisana u tablici ParkingPlace sa stranim ključem na id parkirališta u čijem su vlasništvu, te tip mjesta uz cijenu i pokazatelj zauzetosti. U sustavu baze podataka također postoji tablica Vehicle koja opisuje vozilo. Vozilo ima strani ključ na id vlasnika, te vlastiti id i naziv. Kako bi sustav rezervacija funkcionirao, postoji tablica Reservation koja povezuje korisnika, parking, parkirno mjesto, vozilo te vrijeme same rezervacije. Svaki korisnik ima svoj novčanik opisan tablicom Wallet sa stranim ključem na korisnika i količinom novca kao atributom. Na samom kraju postoji tablica Transaction koja prati uplate i isplate u sustavu.

USER		
user ID	INT	jedinstveni identifikator korisnika

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

USER		
username	VARCHAR	korisničko ime
password	VARCHAR	hash lozinke
name	VARCHAR	ime korisnika
surname	VARCHAR	prezime korisnika
IBAN	VARCHAR(34)	IBAN korisnika koji se koristi prilikom plaćanja
email	VARCHAR	email korisnika
is email confirmed	BOOLEAN	potvrda je li korisnik potvrdio svoj račun preko mail-a
role id	INT	broj korisnikove uloge u sustavu
id image path	VARCHAR	fotografija osobne iskaznice korisnika

MANAGER		
user ID	INT	jedinstveni identifikator korisnika koji je menadžer parkinga, strani ključ na user ID user tablice
confirmed by admin	BOOLEAN	vrijednost je li korisnik potvrđen od strane administratora da je menadžer

PARKING		
parking ID	INT	jedinstveni identifikator parkinga
owner ID	INT	ID na korisnika vlasnika parkinga, strani ključ
parking name	VARCHAR	ime parkinga
parking description	VARCHAR	opis parkinga, koliko ima mjesta, lokacija, je li natkriven, je li osvijetljen...

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

PARKING		
parking photo	VARCHAR	fotografija parkinga

PARKING SPACE		
parking space ID	INT	jedinstveni identifikator parkirnog mjesta
parking ID	INT	ID na parking u čijem je vlasništvu odabrano parkirno mjesto, strani ključ
type	INT	id vozila čiji tip odgovara odabranom parkirnom mjestu (npr: automobil, bicikl...), strani ključ
status	ID	status zauzetosti parkinga, vrsta podatka je integer jer je s brojem označen status parkinga
latitude	INT	brojčana vrijednost fizičke širine parkinga, u metrima
longitude	INT	brojčana vrijednost fizičke duljine parkinga, u metrima
price	INT	cijena parkirnog mjesta, po satu

VEHICLE		
vehicle ID	INT	jedinstveni identifikator vozila
vehicle name	VARCHAR	ime vozila

RESERVATION		
reservation ID	INT	jedinstveni identifikator rezervacije
user ID	INT	ID na korisnika koji je napravio rezervaciju, strani ključ
reservation date	DATETIME	datum i vrijeme rezervacije
duration	ID	vremenska duljina rezervacije, u satima
vehicle type	INT	tip vozila na čije ime glasi rezervacija, strani ključ na tablicu vehicle
parking space id	INT	jedinstveni identifikator koje je točno parkirno mjesto rezervirano, strani ključ
repeating	BOOLEAN	vrijednost koja pokazuje treba li se rezervacija periodički ponavljati (na dnevnoj, tjednoj ili mjesečnoj razini...)

TRANSACTION		
transaction ID	INT	jedinstveni identifikator transakcije
user ID	INT	ID na korisnika koji je obavio transakciju, strani ključ
type	INT	tip transakcije
amount	FLOAT	količina novca koji je prošao kroz transakciju

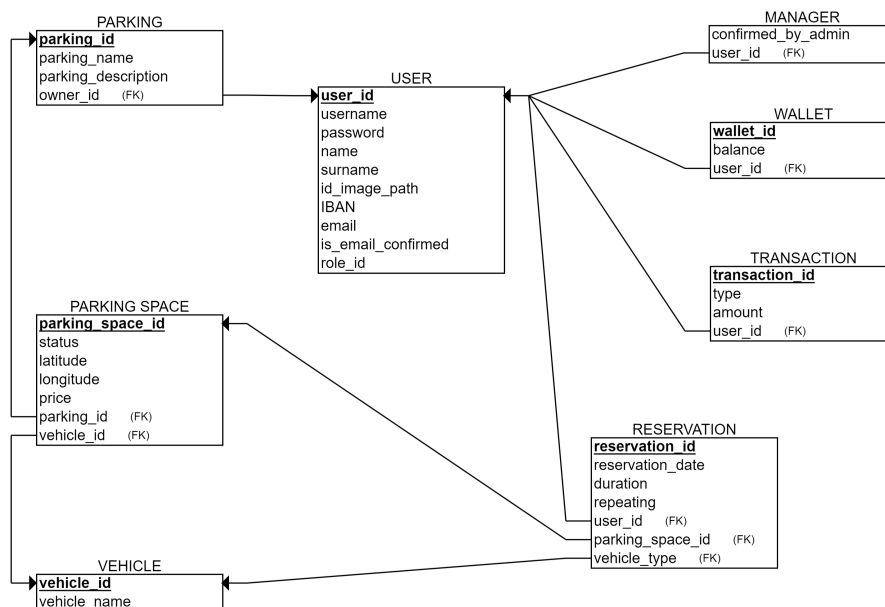
WALLET		
wallet ID	INT	jedinstveni identifikator novčanika
user ID	INT	ID na korisnika u čijem je vlasniku virtualni novčanik, strani ključ

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

WALLET		
balance	FLOAT	količina novca koji se nalazi u novčaniku

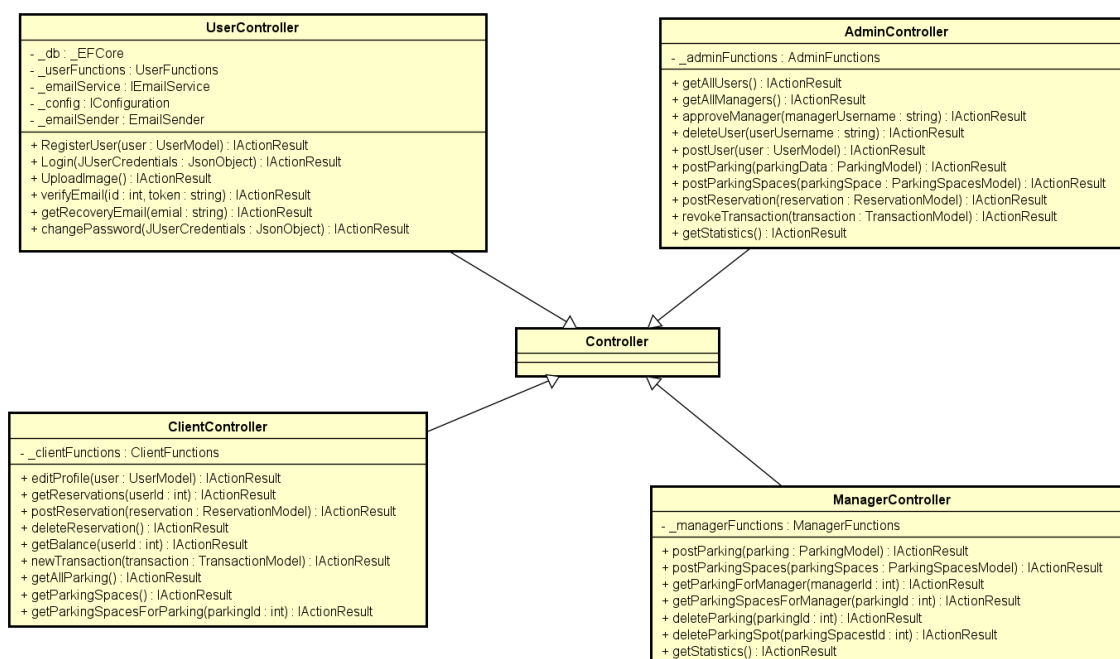
4.1.2 Dijagram baze podataka



Slika 4.2: ER dijagram baze podataka

4.2 Dijagram razreda

Na slikama su prikazani razredi („Class“) koji su korišteni za implementaciju backend-a. Na slici 4.7 su prikazani razredi koji nasljeđuju Controller razred. Sve funkcije implementirane u Controller razredu vraćaju IActionResult (podatke i odgovarajući kod). Također, sve funkcije ne komuniciraju direktno s bazom podataka nego pozivaju funkcije iz određenih servisa koji imaju implementiranu tu funkcionalnost. Funkcije u kontrolerima kao parametre primaju odgovarajući model.



Slika 4.3: dio Controllers

Modeli razreda odražavaju strukturu baze podataka unutar aplikacije. Metode implementirane unutar tih razreda izravno komuniciraju s bazom podataka kako bi dobile tražene informacije. Razred User predstavlja generičnog korisnika aplikacije koji se može registrirati. Na taj razred referira se razred Manager (jer je svaki Manager ujedno i User).

U dijagramu razreda na slici 4.9 prikazani je dio servisa. Svi servisi kao svoj atribut, između ostalih, imaju instancu objekta `_EFCore` koji predstavlja kontekst za bazu podataka i instancu `IConfiguration` objekta koji služi za dohvaćanje konstanti. Funkcije definirane u pojedinom razredu dohvaćaju, mijenjaju i dodaju podatke u bazu podataka i manipuliraju podacima koje vraćaju kontroleru koji šalje nazad do korisnika.

4.3 Dijagram stanja

dio 2. revizije

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

4.4 Dijagram aktivnosti

dio 2. revizije

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

4.5 Dijagram komponenti

dio 2. revizije

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

4.6 Dijagram razreda

Na slikama su prikazani razredi („Class“) koji su korišteni za implementaciju backend-a. Na slici 4.7 su prikazani razredi koji nasljeđuju Controller razred. Sve funkcije implementirane u Controller razredu vraćaju IActionResult (podatke i odgovarajući kod). Također, sve funkcije ne komuniciraju direktno s bazom podataka nego pozivaju funkcije iz određenih servisa koji imaju implementiranu tu funkcionalnost. Funkcije u kontrolerima kao parametre primaju odgovarajući model.

Modeli razreda odražavaju strukturu baze podataka unutar aplikacije. Metode implementirane unutar tih razreda izravno komuniciraju s bazom podataka kako bi dobile tražene informacije. Razred User predstavlja generičnog korisnika aplikacije koji se može registrirati. Na taj razred referira se razred Manager (jer je svaki Manager ujedno i User).

U dijagramu razreda na slici 4.9 prikazani je dio servisa. Svi servisi kao svoj atribut, između ostalih, imaju instancu objekta `_EFCore` koji predstavlja kontekst za bazu podataka i instancu `IConfiguration` objekta koji služi za dohvaćanje konstanti. Funkcije definirane u pojedinom razredu dohvaćaju, mijenjaju i dodaju podatke u bazu podataka i manipuliraju podacima koje vraćaju kontroleru koji šalje nazad do korisnika.

4.7 Dijagram stanja

dio 2. revizije

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

4.8 Dijagram aktivnosti

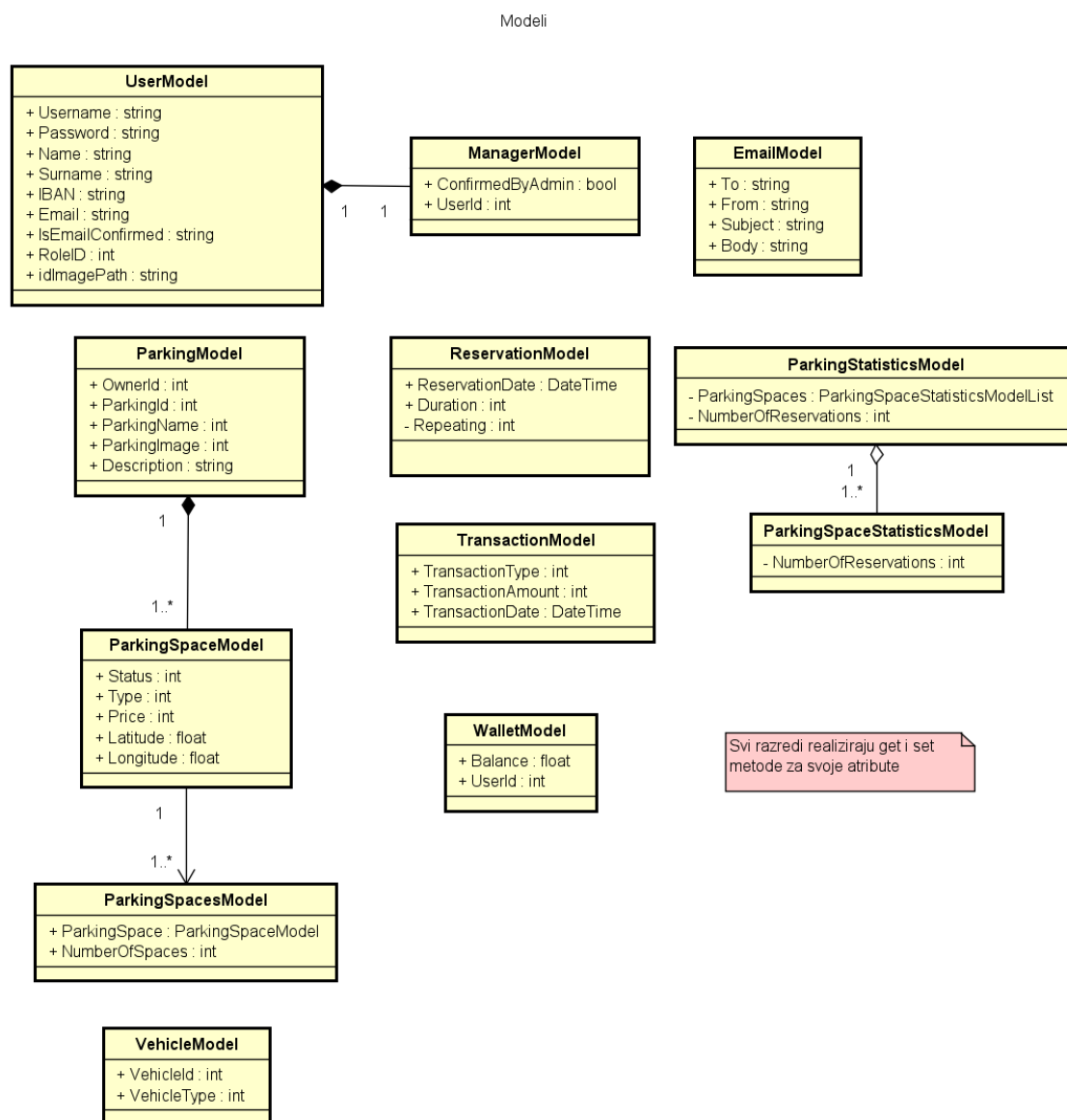
dio 2. revizije

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

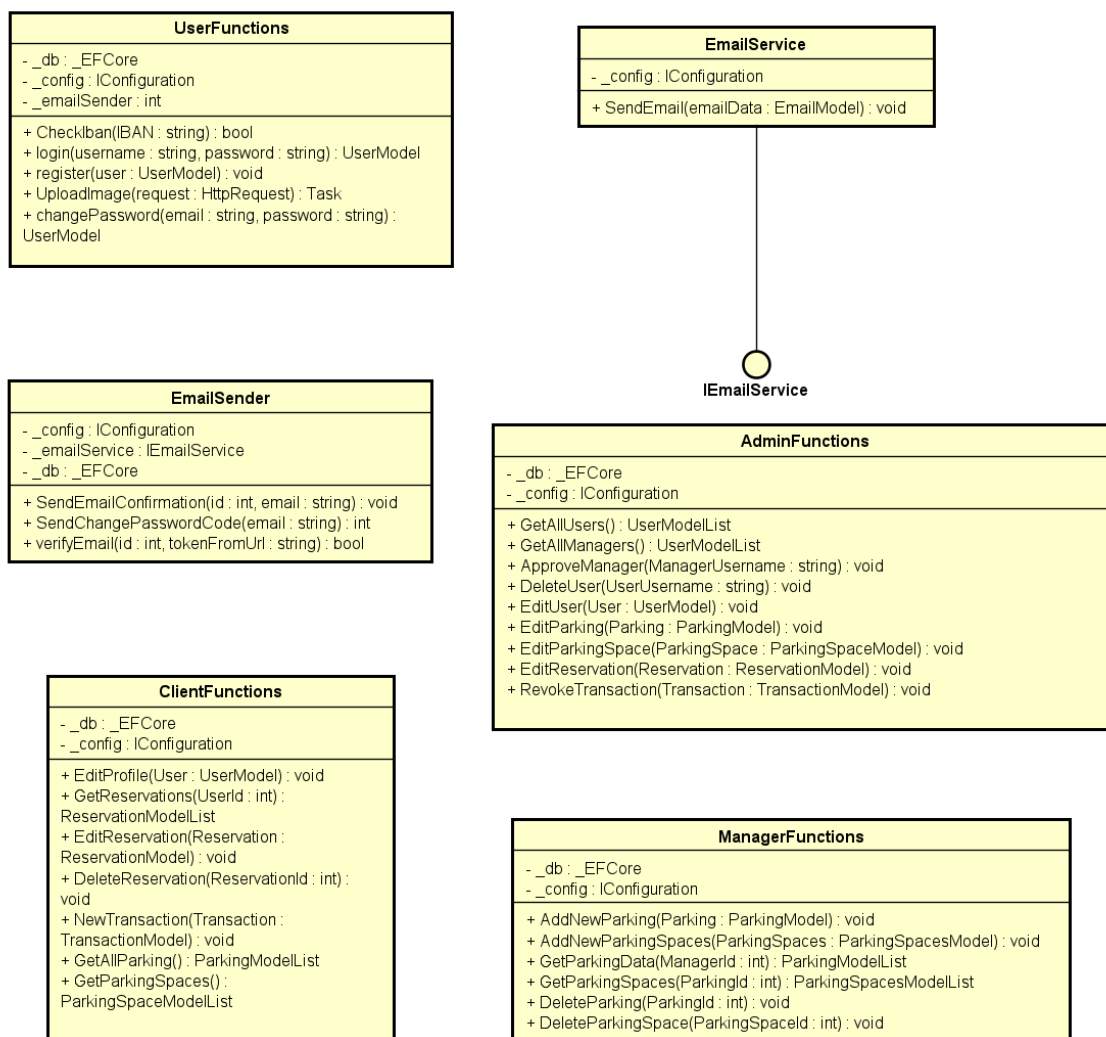
4.9 Dijagram komponenti

dio 2. revizije

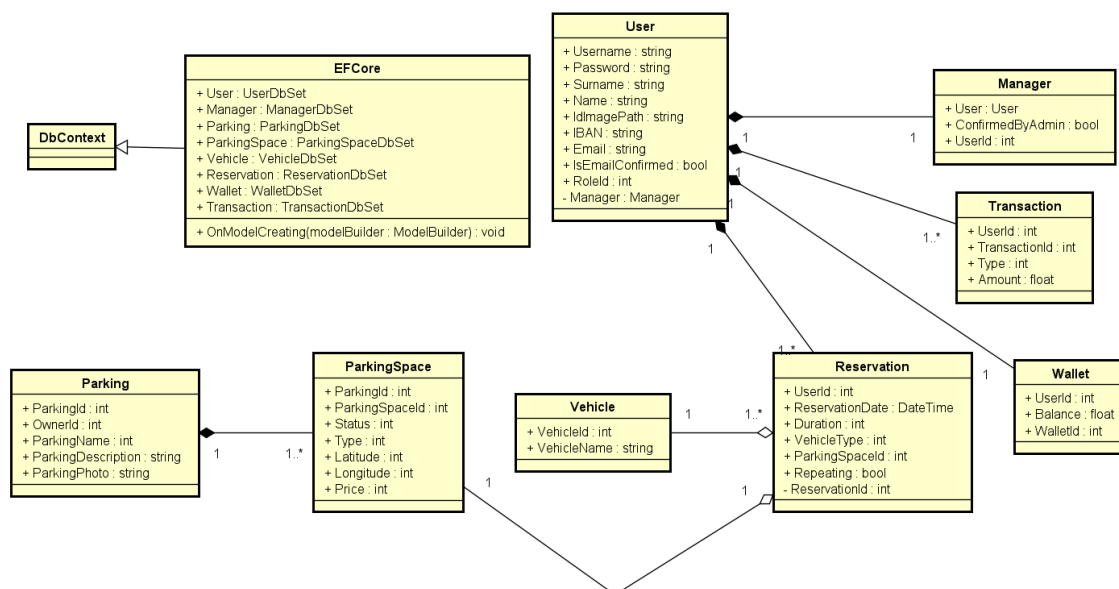
Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.



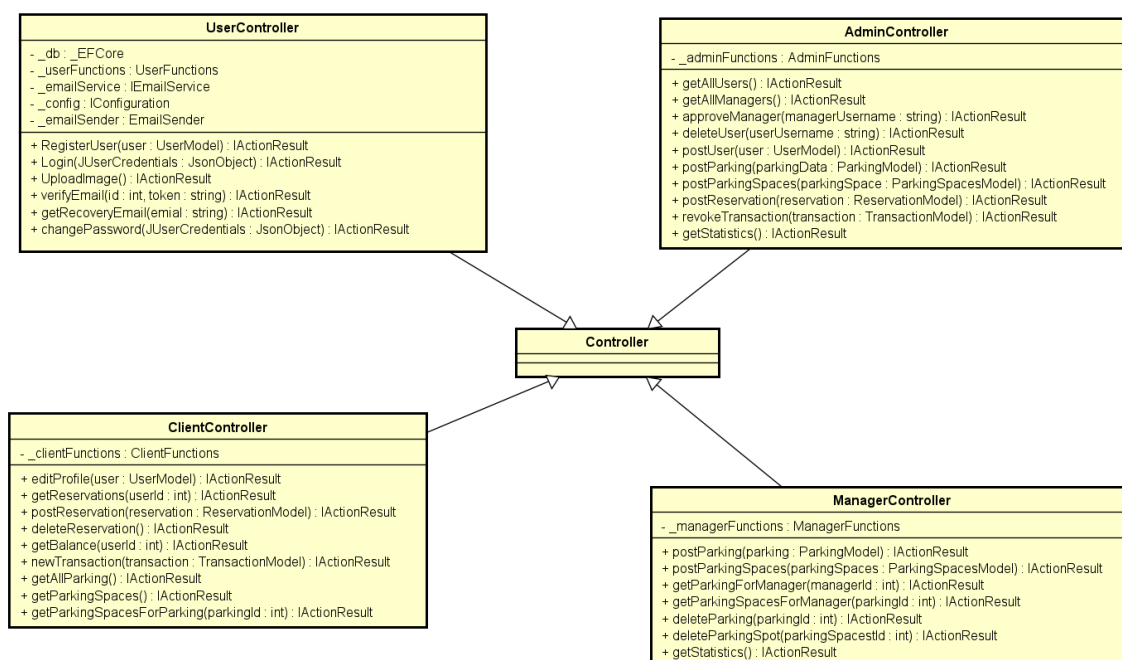
Slika 4.4: dio Models



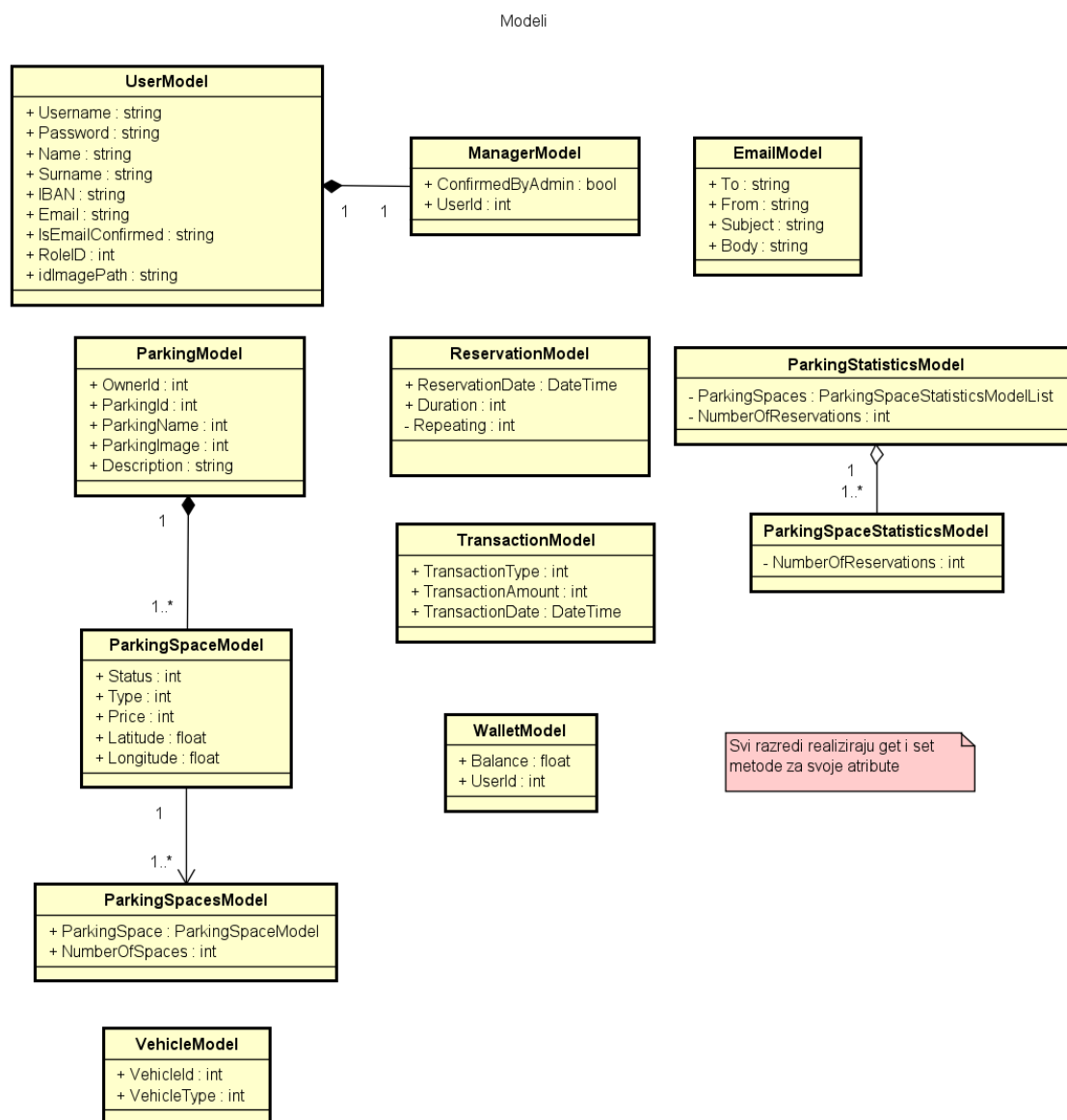
Slika 4.5: dio Services



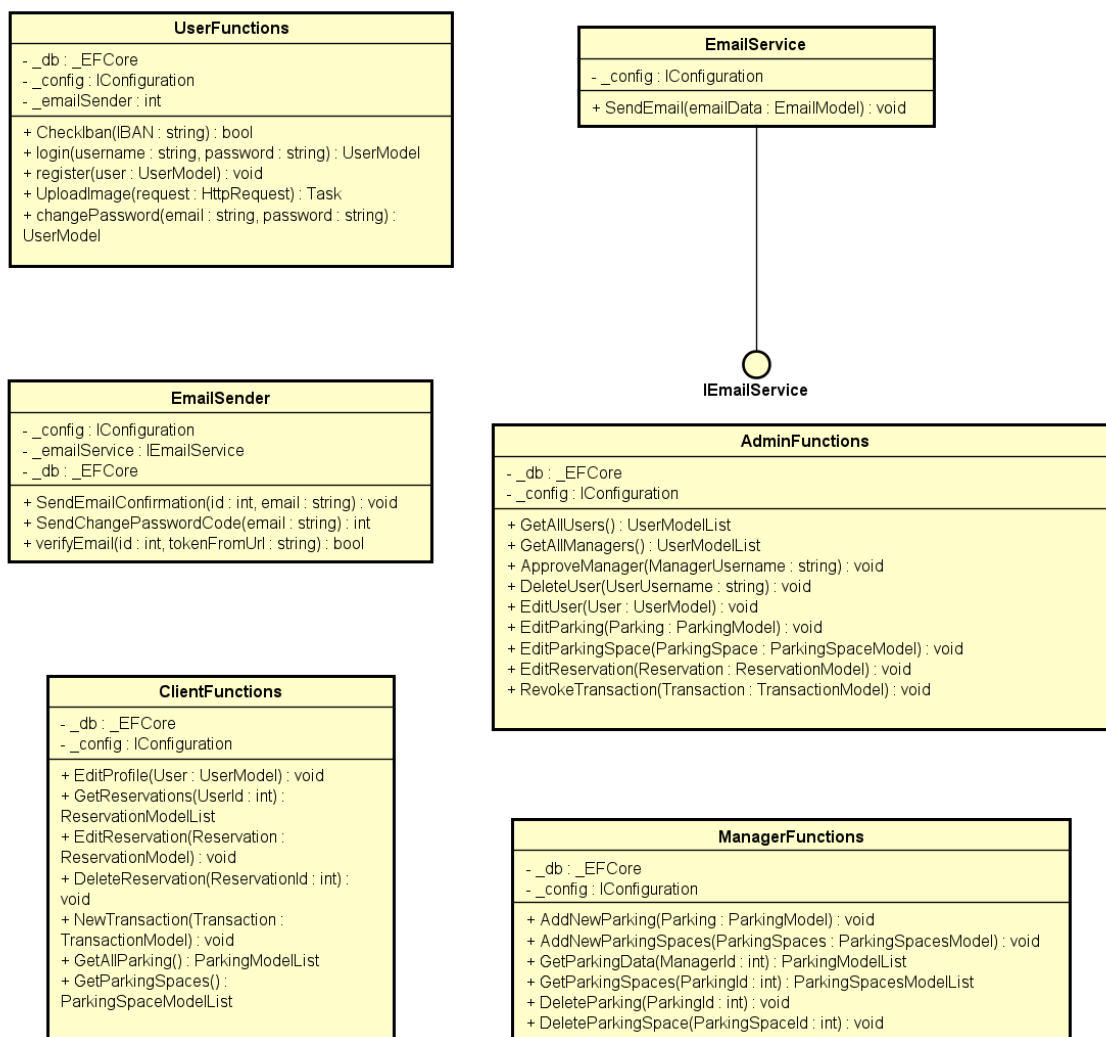
Slika 4.6: Reprezentacija baze podataka



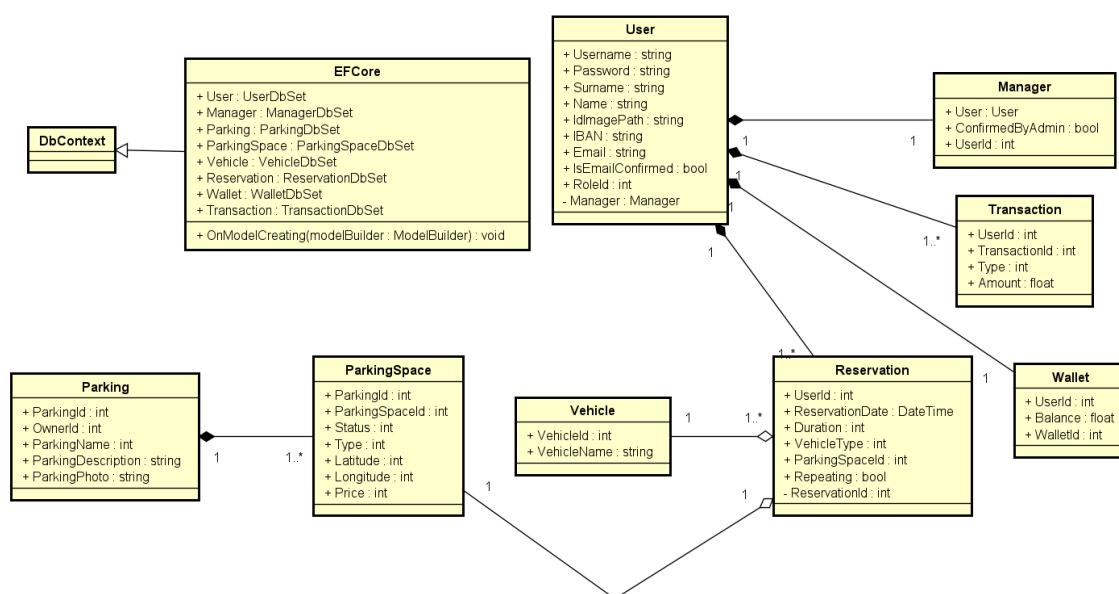
Slika 4.7: dio Controllers



Slika 4.8: dio Models



Slika 4.9: dio Services



Slika 4.10: Reprezentacija baze podataka

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

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

5.2 Ispitivanje programskog rješenja

dio 2. revizije

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

5.2.1 Ispitivanje komponenti

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

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

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

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

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

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

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

5.4 Upute za puštanje u pogon

dio 2. revizije

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

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

6. Zaključak i budući rad

dio 2. revizije

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

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

Popis literature

Kontinuirano osvježavanje

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

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

Indeks slika i dijagrama

2.1	Sučelje za odabir termina u JustPark	8
2.2	Dodatna mogućnost ocjene parkinga u JustPark	8
3.1	Sekvencijski dijagram za UC4	20
3.2	Sekvencijski dijagram za UC5	21
3.3	Sekvencijski dijagram za UC6	22
4.1	Arhitektura sustava	23
4.2	ER dijagram baze podataka	30
4.3	dio Controllers	31
4.4	dio Models	41
4.5	dio Services	42
4.6	Reprezentacija baze podataka	43
4.7	dio Controllers	43
4.8	dio Models	44
4.9	dio Services	45
4.10	Reprezentacija baze podataka	46

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

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

2. sastanak

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

Tablica aktivnosti

Kontinuirano osvježavanje

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

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

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

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

Dijagrami pregleda promjena

dio 2. revizije

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