Oblikovanje programske potpore

Ak. god. 2018./2019.

*ParkirajMe*

Dokumentacija, Rev. 1

Grupa: *ferovci*

Voditelj: *Hrvoje Ditrih*

Datum predaje: *<dan>. <mjeseca> <godine>.*

Nastavnik: *Nikolina Frid*

Sadržaj

[1.](#_30j0zll) Dnevnik promjena dokumentacije 3

[2.](#_2et92p0) Opis projektnog zadatka 5

[3.](#_3dy6vkm) Pojmovnik 7

[4.](#_4d34og8) Funkcionalni zahtjevi 8

[4.1 Opis obrazaca uporabe 9](#_2s8eyo1)

[4.2 Dijagrami obrazaca uporabe 15](#_17dp8vu)

[4.3 Sekvencijski dijagrami 18](#_26in1rg)

[5.](#_lnxbz9) Ostali zahtjevi 26

[6.](#_1ksv4uv) Arhitektura i dizajn sustava 27

[6.1.](#_44sinio) Svrha, opći prioriteti i skica sustava 28

[6.2.](#_z337ya) Dijagram razreda s opisom 29

[6.3.](#_1y810tw) Dijagram objekata 30

[6.4.](#_2xcytpi) Ostali UML dijagrami 31

[7.](#_3whwml4) Implementacija i korisničko sučelje 32

[7.1.](#_qsh70q) Dijagram razmještaja 32

[7.2.](#_1pxezwc) Korištene tehnologije i alati 33

[7.3.](#_49x2ik5) Isječak programskog koda vezan za temeljnu funkcionalnost sustava 34

[7.4.](#_147n2zr) Ispitivanje programskog rješenja 35

[7.5.](#_23ckvvd) Upute za instalaciju 36

[7.6.](#_32hioqz) Korisničke upute 37

[8.](#_2grqrue) Zaključak i budući rad 38

[9.](#_3fwokq0) Popis literature 39

[Chapter 2 Dodatak A: Indeks (slika, dijagrama, tablica, ispisa kôda) 40](#_4f1mdlm)

[Chapter 3 Dodatak B: Dnevnik sastajanja 41](#_2u6wntf)

[Chapter 4 Dodatak C: Prikaz aktivnosti grupe 42](#_19c6y18)

[Chapter 5 Dodatak D: Plan rada / Pregled rada i stanje ostvarenja 44](#_3tbugp1)

*Sadržaj bi se trebao automatski osvježavati prema tekstu (desni klik, „Update Field“) ako se bude držalo zadanih formata poglavlja.*

# 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 |  | Horvat | 10.09.2013. |
| 0.92 | Nastavak dijagrama razreda | Horvat | 11.09.2013. |
| 1.0 | Verzija samo s bitnim dijelovima za 1. ciklus | Ivošević | 11.09.2013. |
| 1.1 | Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi | Grudenić | 14.09.2013. |
| 1.11 | Manje izmjene: Timer > Brojilo vremena | Grudenić | 15.09.2013. |
| 1.12 | Popravljeni dijagrami obrazaca uporabe | Jović | 15.09.2013. |
| 1.5 | Generalna revizija strukture dokumenta | Ivošević | 19.09.2013. |
| 1.55 | Manja revizija (dijagram razmještaja) | Jović | 20.09.2013. |
| 2.0 | Konačni tekst predloška dokumentacije | Ivošević | 28.09.2013. |
| 3.0 | Nadopuna poglavlja o arhitekturi | Jović | 25.11.2013. |
| 4.0 | Nadopune po pitanju praćenja aktivnosti u grupi | Ivošević | 01.10.2015. |
| 4.1 | Definirani Dodaci C i D | Ivošević | 05.10.2015. |

*Moraju postojati glavne revizije (verzije, inačice) 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, (ukoliko bude više od toga nastavlja se sa 0.10, 0.11…) sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd. Ukoliko se želi naglasiti veličina promjene/dodatka unutar revizije može se napraviti različita „granulacija“ u označavanju revizija. Npr. nakon 0.1 može biti 0.11, 0.12, 0.15 pa tek onda 0.2. Ili se za veće promjene/zahvate može nakon 0.1 odmah skočiti na 0.5. itd. Bitno je da na krajevima ciklusa revizije koje se predaju budu s oznakama* ***1.0*** *i* ***2.0*** *(Preporuka je da te revizije budu podebljane unutar liste).*

# 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 aplikacije 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. Registrirani korisnik ima na raspolaganju sljedeće opcije:

1. Traženje najbližih slobodnih parkirališta

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.

1. Rezervacija parkirališnih mjesta

Korisnik odabire gumb za rezervaciju, željeno parkiralište te način 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.

# 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

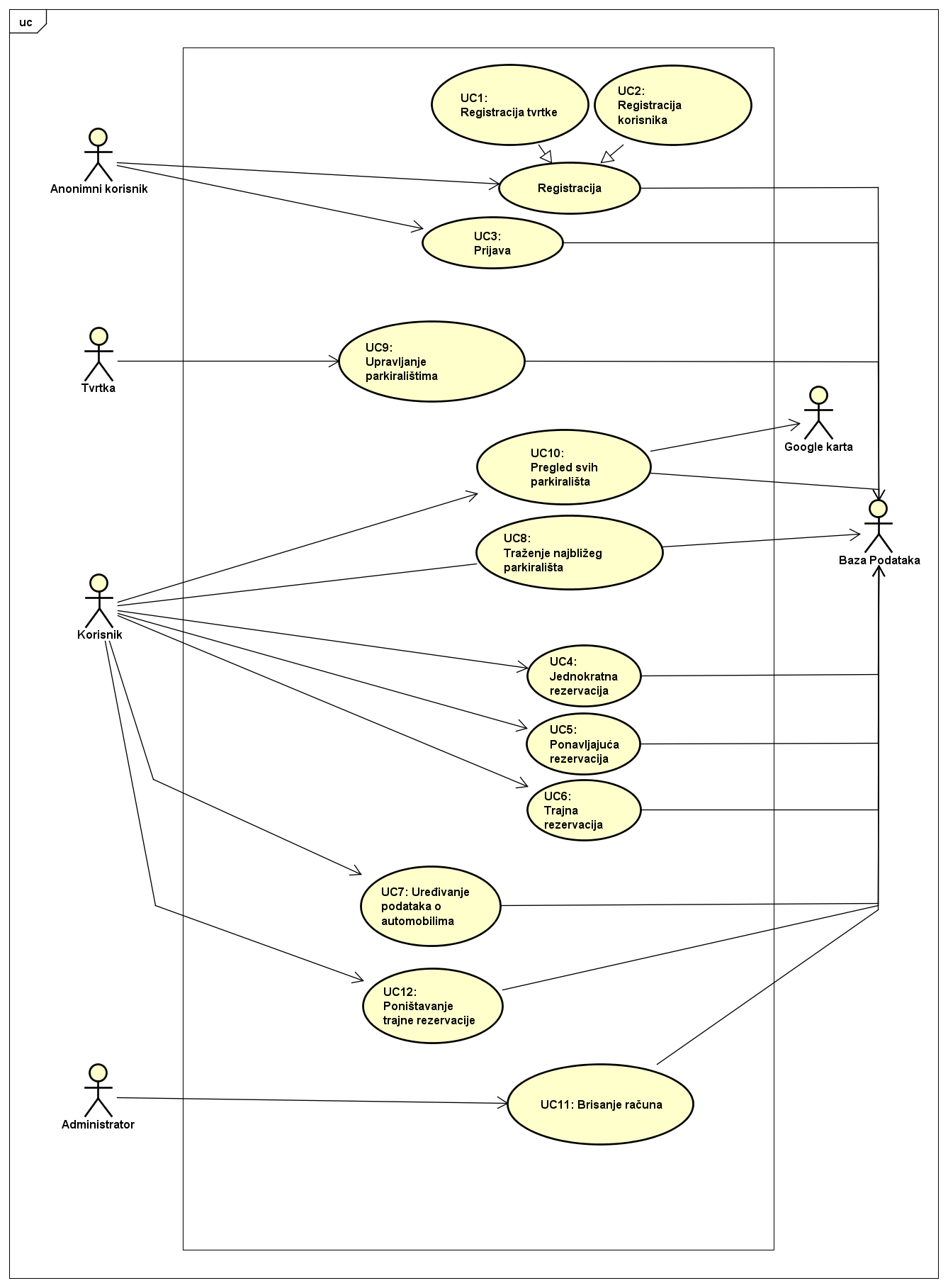
## 4.1 Opis obrazaca uporabe

* 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 uspješno sprema podatke
  + Opis mogućih odstupanja:
    - Korisnik upisuje svoje podatke
    - Baza podataka ne sprema podatke zbog neispravnih podataka ili zato što već postoji takva registrirana tvrtka
* 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 uspješno sprema podatke
  + Opis mogućih odstupanja:
    - Korisnik upisuje svoje podatke
    - Baza podataka ne sprema podatke zbog neispravnih podataka ili zato što već postoji takav registrirani korisnik

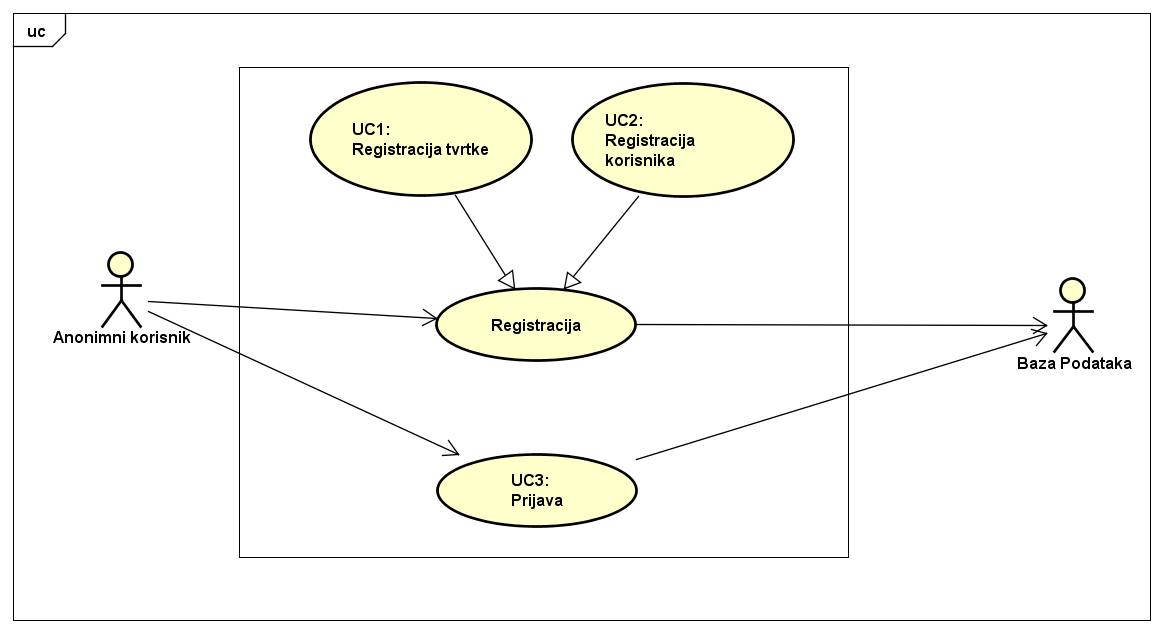
Rbr.: UC3

* + 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:
    - Korisnik pogrešno upisuje svoje podatke
    - Sustav provjerava ispravnost podataka i javlja grešku
* 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 i rezervira mjesto
  + Opis mogućih odstupanja:
    - Korisnik odabire vremenski blok u kojem nema slobodnih mjesta u garaži
    - 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, obavlja plaćanje i rezervira mjesto
  + Opis mogućih odstupanja:
    - Korisnik odabire barem jedan vremenski blok u kojem nema slobodnih mjesta u garaži
    - 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:
    - Korisnik potvrđuje da mjesto želi trajno rezervirati
    - 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
    - Korisnik klikne na gumb za brisanje postojećeg automobila
  + Opis mogućih odstupanja:
    - Korisnik unosi neispravne podatke o automobilu
    - Neki drugi korisnik već posjeduje automobil koji trenutni korisnik pokušava dodati
* Rbr.: UC8
  + Naziv: Traženje najbližeg parkinga
  + Opis: Na temelju trenutne lokacije korisnika aplikacija određuje najbliže dostupno parkiralište
  + Glavni aktor: Korisnik
  + Preduvjeti: Korisnik je prijavljen
  + Pokretač: Pritisak na gumb *ParkirajMe*
  + Opis osnovnog tijeka:
    - Aplikacija gleda najbliže parkiralište po udaljenosti te gleda broj dostupnih mjesta na njemu
    - Ako aplikacija procjeni da na parkiralištu ima dovoljan broj slobodnih mjesta, nudi korisniku to parkiralšte
  + Opis mogućih odstupanja:
    - Nedovoljno slobodnih mjesta na najbližem parkiralištu pa aplikacija nudi prvo sljedeće parkiralište po blizini sa dovoljno slobodnih mjesta
* Rbr.: UC9
  + Naziv: Upravljanje parkiralištima
  + Opis: Tvrtka dodaje novi parking ili uređuje podatke o postojećem parkingu
  + Glavni aktor: Tvrtka
  + Preduvjeti: Tvrtka je prijavljena
  + Opis osnovnog tijeka:
    - Aplikacija izlistava tvrtki sva njena parkirališta
    - Tvrtka odabire neko parkiralište te klikne na gumb za uređivanje
    - Uređuje podatke o parkiralištu i potvrđuje promjene
    - Alternativno: tvrtka kline na gumb za dodavanje novog parkirališta te upisuje podatke i potvrđuje dodavanje
  + Opis mogućih odstupanja:
    - Istekla sjednica korisnika
* 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:
    - Korisniku je istekla sjednica
* 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:
    - Istekla je sjednica korisnika
* 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

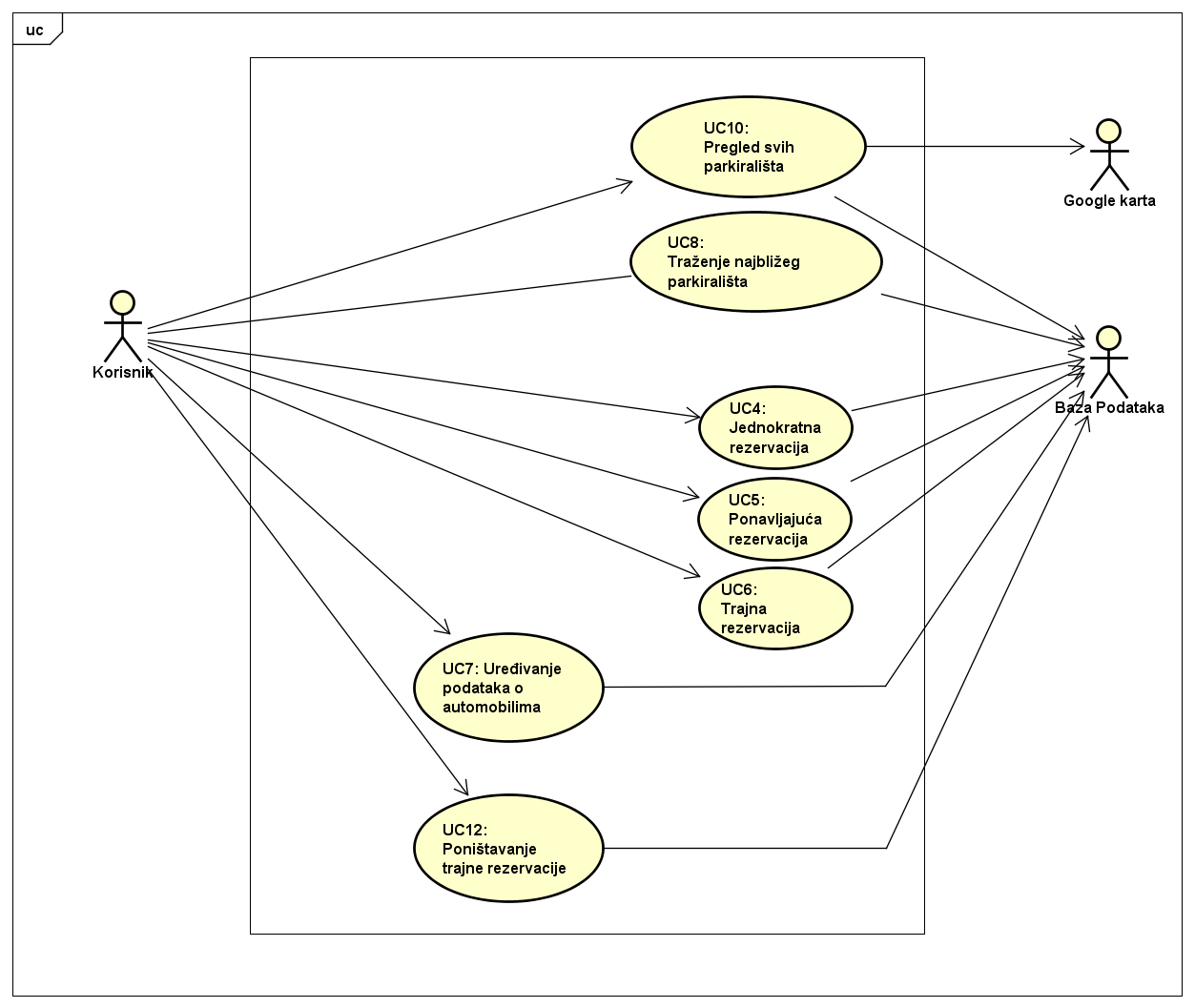
## 4.2 Dijagrami obrazaca uporabe



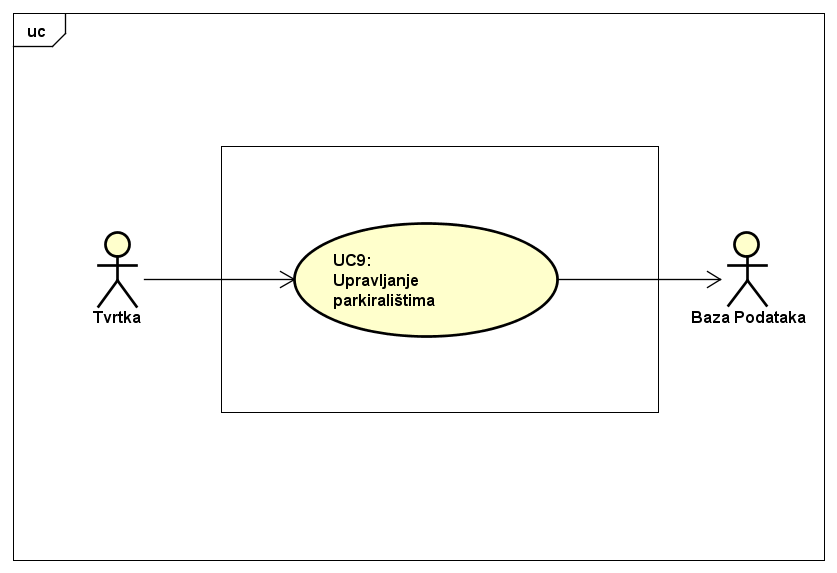
*Slika 4.2.1 Dijagram svih obrazaca uporabe*



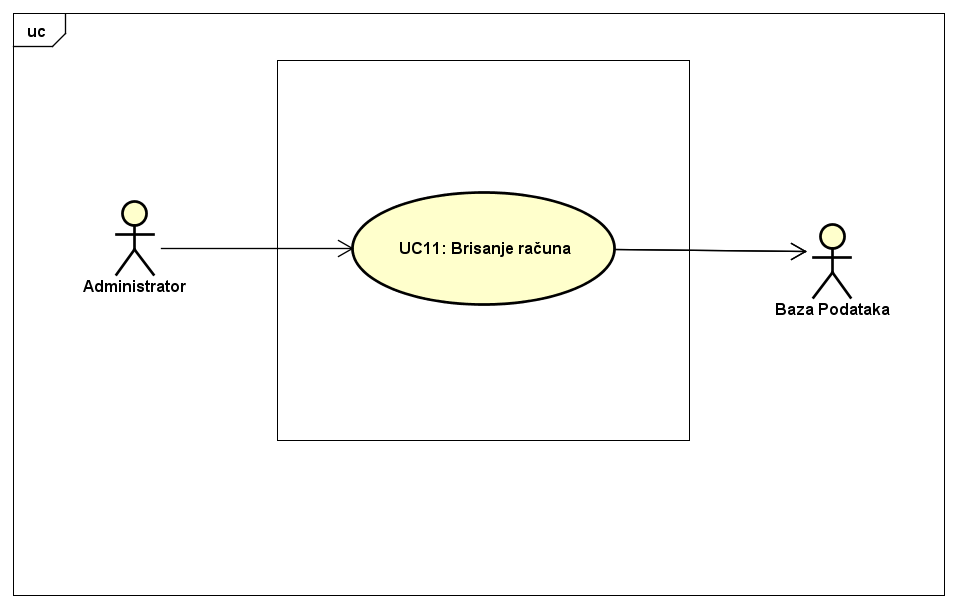
*Slika 4.2.2 Dijagram obrazaca uporabe za anonimnog korisnika – UC1, UC2, UC3*



*Slika 4.2.3 Dijagram obrazaca uporabe za korisnika – UC4-8, UC10*



*Slika 4.2.4 Dijagram obrazaca uporabe za tvrtku – UC9*



*Slika 4.2.5 Dijagram obrazaca uporabe za administratora i sustav – UC11, UC12*

## 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:**

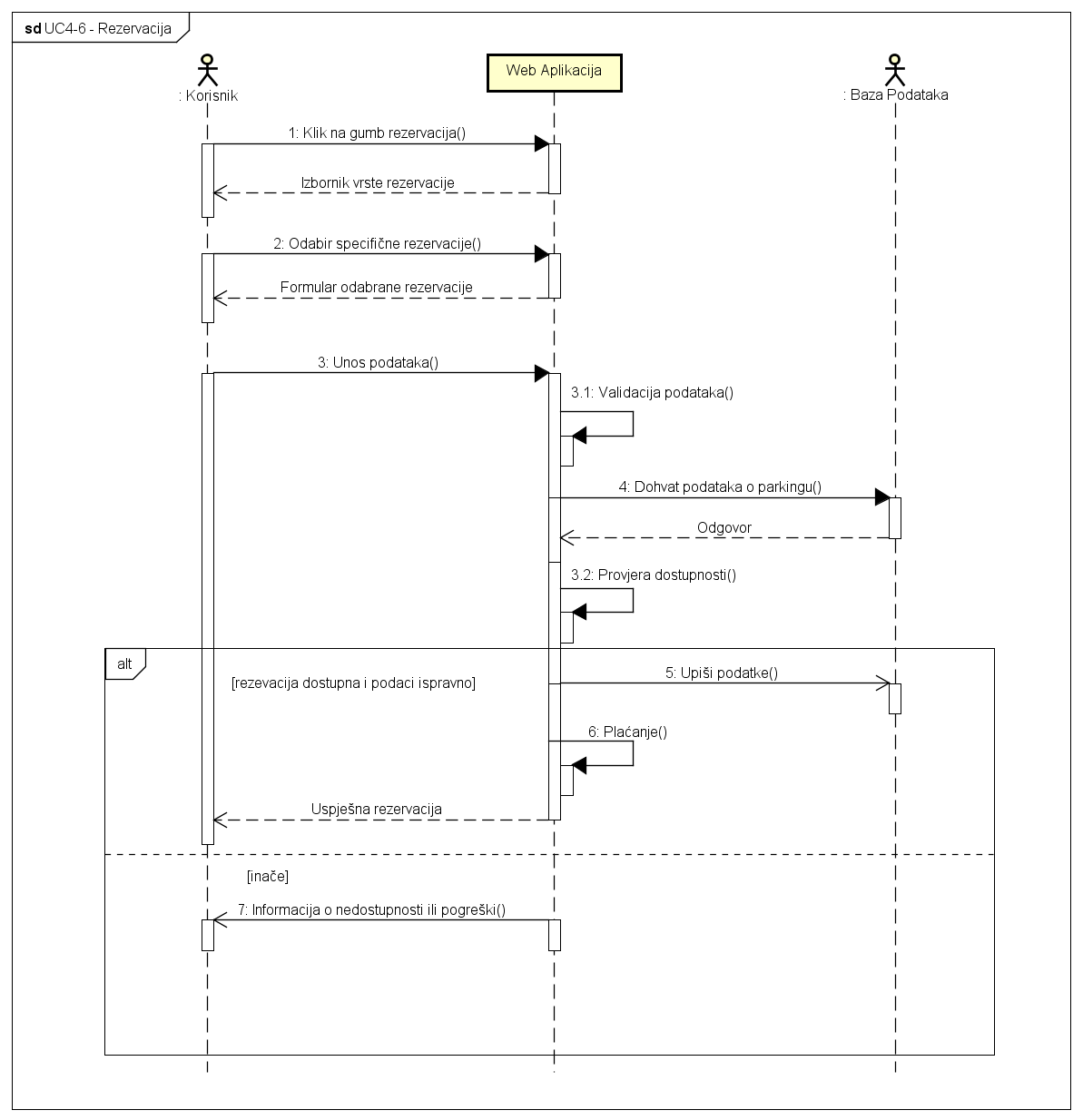
Anonimni korisnik klikne na gumb 'prijava' te zatim ispunjava obrazac s podacima nakon čega sustav provjerava ispravnost tih podataka. Ako sustav utvrdi da su podaci ispravni korisnik se prosljeđuje njegovo sučelje ovisno o vrsti korisnika: korisnik, tvrtka ili administrator. Inače, korisnik dobiva poruku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

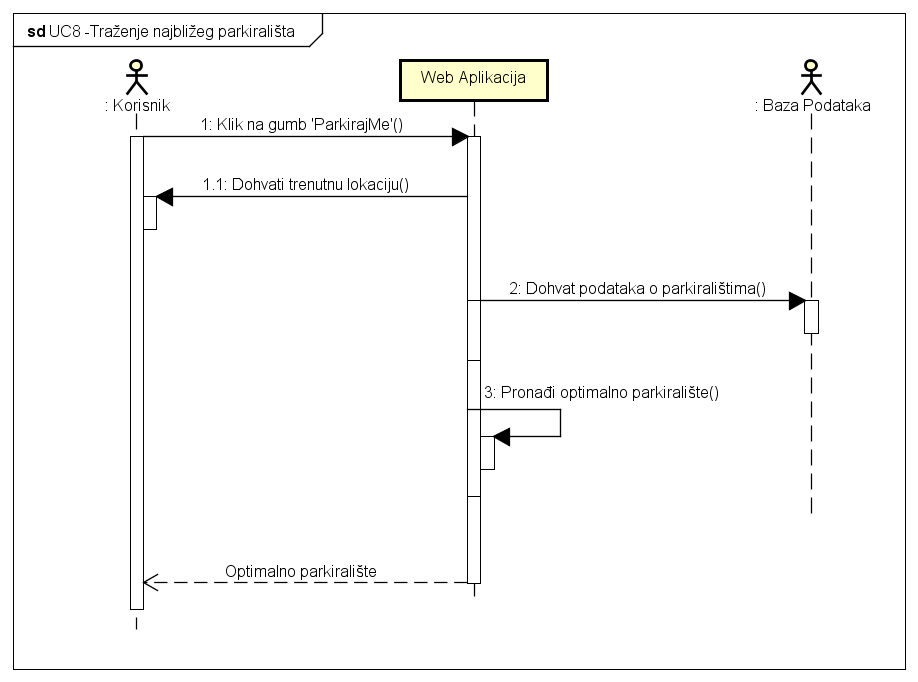
Korisnik klikne na gumb 'Upravljanje automobilima' te se prosljeđuje na popis svih svojih automobila. Korisnik zatim može dodati novi automobil ili obrisati postojeći. Klikom na gumb za brisanje postojećeg automobila od korisnika se traži potvrda za traženu akciju. Ako korisnik klikne potvrdno automobil 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:**

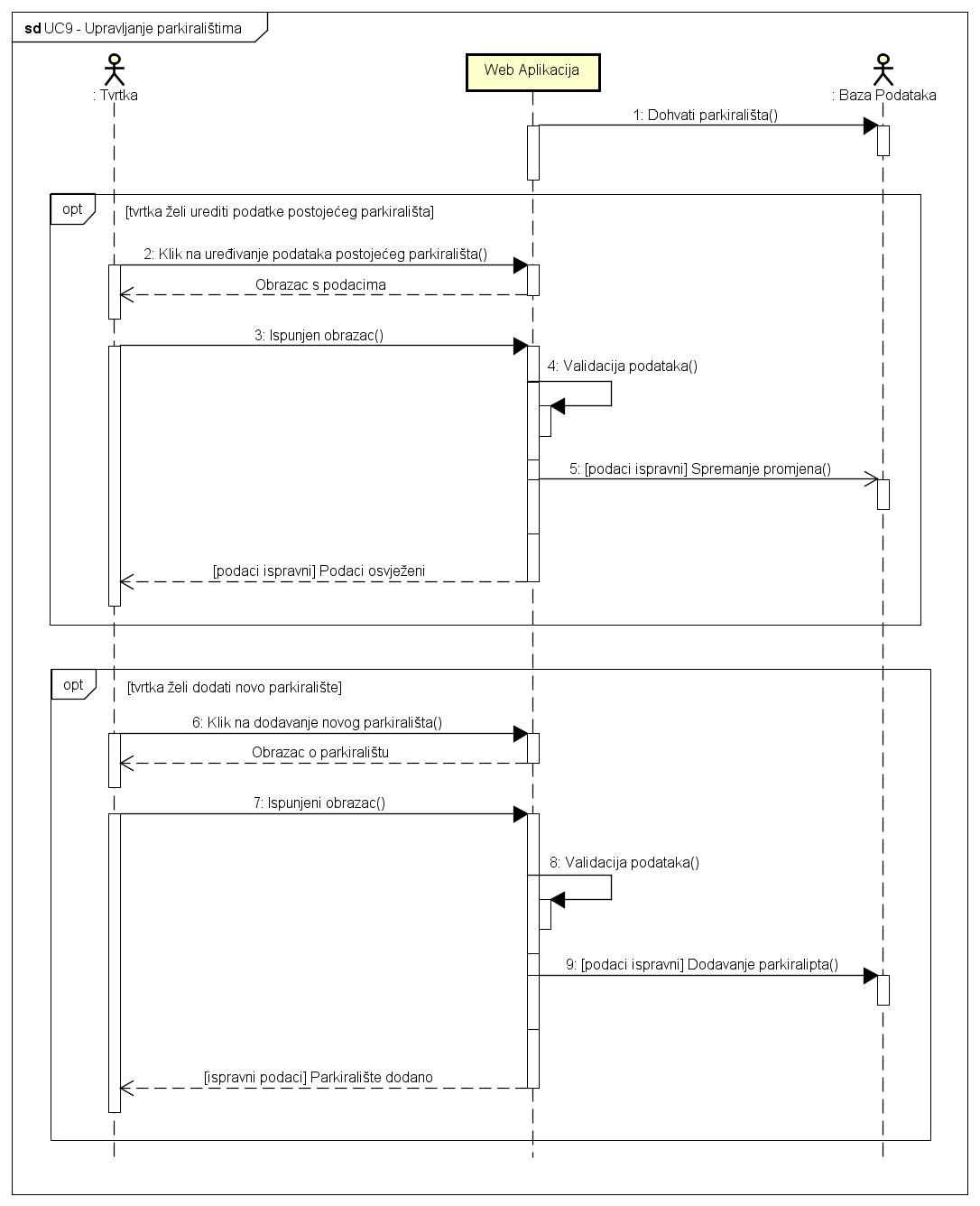
Korisnik klikne na gumb 'ParkirajMe'. Dohvate se podaci o svim parkiralištima nakon čega se određuje optimalno parkiralište na osnovu trenutne lokacije korisnika i popunjenosti najbližih parkirališta. Korisnika se prosljeđuje na određeno parkiralište.



*Slika 4.3.5 Sekvencijski dijagram za traženje parkirališta – UC8*

**Obrasci uporabe UC9:**

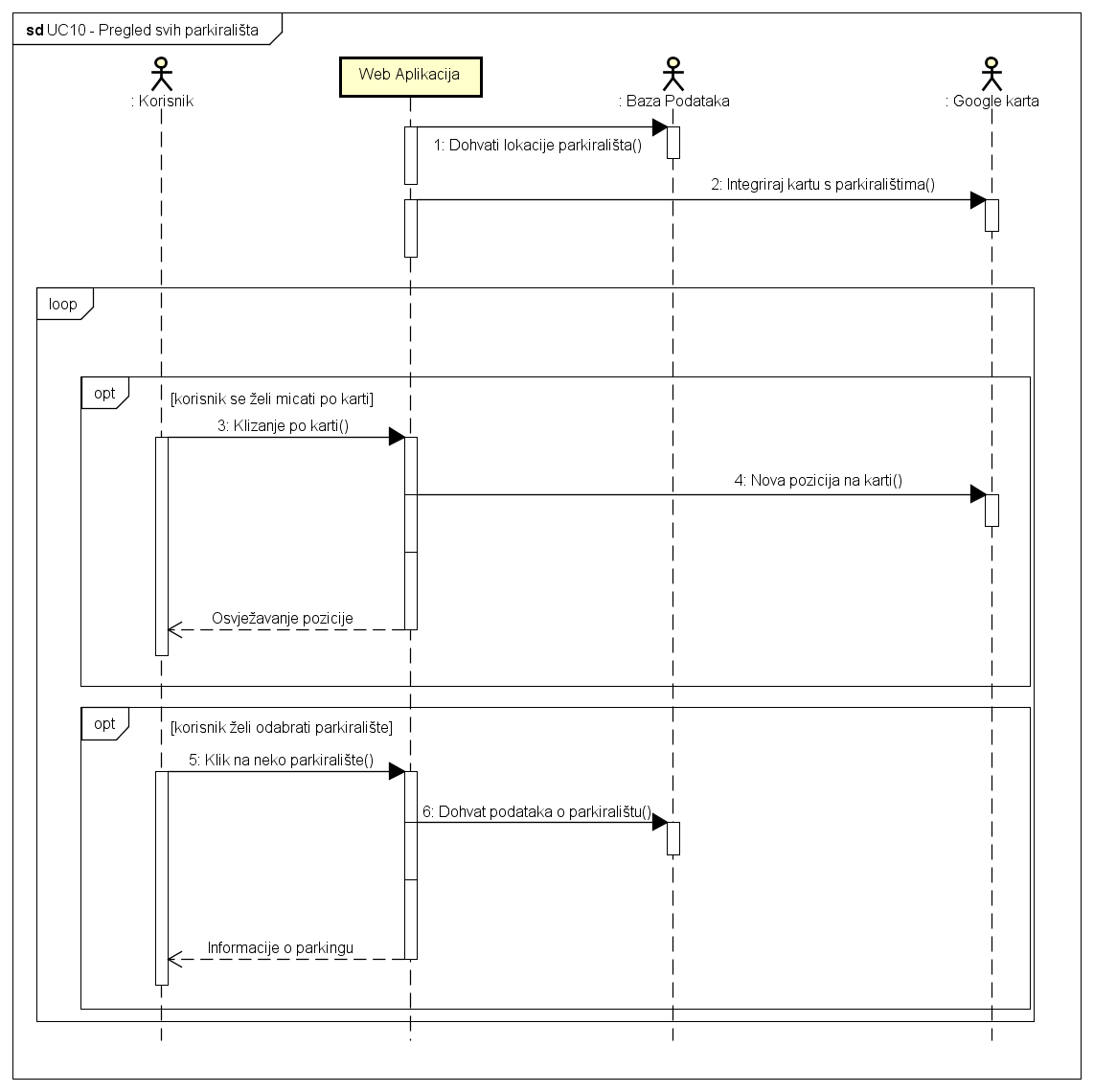
Tvrtka može dodati novo parkiralište ili urediti podatke o postojećem. Klikom na uređivanje postojećeg parkirališta tvrtka ispunjava obrazac s promjenama te sprema promjene. Klikom na gumb za dodavanje novog parkirališta tvrtki se nudi obrazac s ispunjavanjem podataka o parkiralištu. Ako su podaci ispravni, novo parkiralište se dodaje u bazu podataka.



*Slika 4.3.6 Sekvencijski dijagram za upravljanje parkiralištima – 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*

# Ostali zahtjevi

*Ostali ili nefunkcionalni zahtjevi nisu izravno vezani za primarnu funkciju sustava. Govore o određenim mjerama kvalitete i svojstvima i ograničenjima platforme i procesa ostvarenja sustava.*

*Pod mjerama i zahtjevima na kvalitetu sustava misli se na nekoliko aspekata: neosjetljivost na pogreške, (is)koristivost, pouzdanost, performanse, (p)održivost.*

*Pod ograničenjima se uglavnom misli na svojstva tehnologije i metodologiju implementacije: ograničenja tehnologije, vrijeme izrade, (ne)usvojeni standardi, informacije o procesu razvoja i oblikovanja.*

*Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.*

Svakih 60 sekundi se azurira nstanje parkiraliosta.

Responzivno.

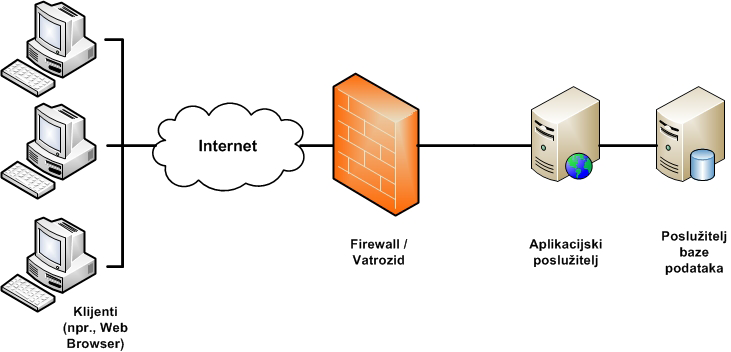
Sustav treba omogućiti paralelni rad više korisnika ∙ ∙ Neispravno korištenje sustava ne smije utjecati na stanje sustava ili baze podataka ∙ Sustav mora biti neosjetljiv na eventualne pogreške ∙ Sustav mora podržavati znakove hrvatske abecede ∙

# 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.

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.



## 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 loyinke 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 rezervraciju
* rezervirajPonavljajuce – generira ponavljajuću rezervaciju
* rezervirajTrajno – generira trajnu rezervaciju
* urediPodatke – omogućava uređivanje korisničkih podataka
* ponistiRezervaciju – omogućava brisanje postojeće rezervacije
* obrisiRacun – omogućava brisanje korisničkog računa

**KorisnikTvrtka –** proširuje razred registriranog korisnika, modelira tvrtku

Atributi:

* imeTvrtke – ime tvrtke
* adresaSjedista – adresa sjedišta
* parkiralista – lista registriranih parkirališta koja pripadaju tvrtki

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

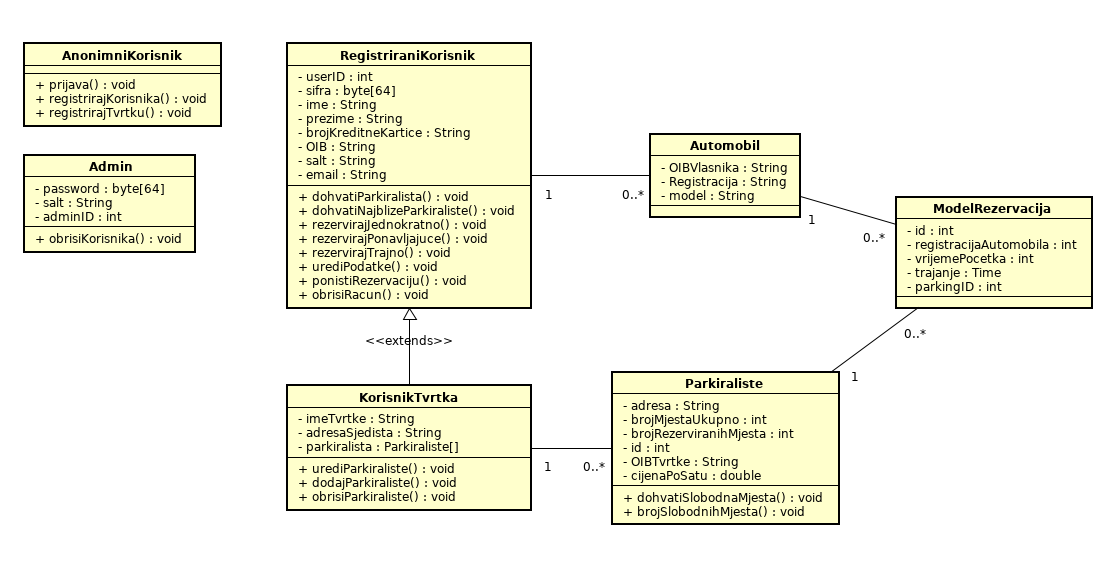
Metode:

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

**ModelRezervacija –** modelira postojeću rezervaciju

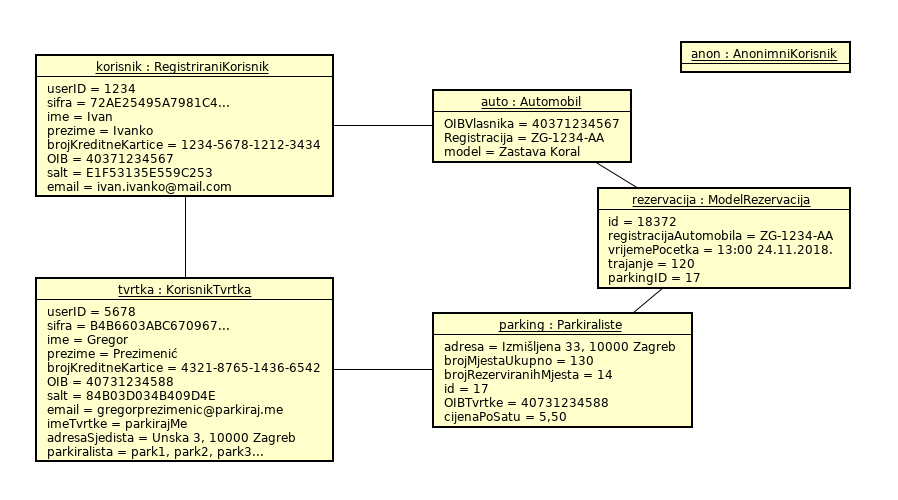
Atributi:

* 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

*Ovdje počinju sadržaji Revizije 2.*

*U ovom potpoglavlju potrebno je uvrstiti dodatna četiri dijagrama koji prikazuju arhitekturu sustava i to: komunikacijski dijagram, dijagram stanja, dijagam aktivnosti i dijagram komponenti. Potrebno je dati barem po jedan primjerak svakog od dijagrama. Potrebno je da svaki od dijagrama prikazuje (jedan) bitan dio funkcionalnosti sustava. Dijagram komponenti treba prikazivati sve komponente sustava. Prema potrebi, ovdje se mogu dodati i neki dodatni sekvencijski dijagrami za objekte.*

# 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

*Potrebno je umetnuti dijagram razmještaja i po potrebi ga opisati.*

## Korištene tehnologije i alati

*Navesti sve tehnologije i alate koji su primijenjeni pri izradi projekta te ih ukratko opisati; njihovo značenje i mjesto i način primjene.*

## Isječak programskog koda vezan za temeljnu funkcionalnost sustava

*U ovom poglavlju potrebno je prikazati isječak programa koji prema mišljenju studenta ostvaruje temeljnu funkcionalnost u sustavu (ili nekom modulu).*

## Ispitivanje programskog rješenja

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti s prikazom odabira ispitnih slučajeva.*

*Prilikom prezentacije svojih Ispitnih scenarija (minimalno četiri) studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete. Poželjno je da se napravi i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane te da se vidi na koji način sustav reagira kada nešto nije u potpunosti ostvareno.*

*Različiti ulazi za ispitne scenarije trebaju pokrivati temeljnu funkcionalnost nekog modula i nekoliko rubnih uvjeta.*

## Upute za instalaciju

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

## Korisničke upute

*Korisničke upute ovisit će o količini implementirane funkcionalnosti. Očekuje se da upute budu na oko pet A4 stranica koje će dati potpuni opis funkcionalnosti sustava sa stajališta krajnjeg korisnika.*

# Zaključak i budući rad

*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.*

# Popis literature

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

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>

1. Astah Community, [http://astah.net/editions/community/](http://argouml.tigris.org/)

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

*U ovom dijelu dodatka potrebno je ispisati (numerirane) liste slika, dijagrama, tablica, ispisa kôda i ostalih pomoćnih sadržaja iz teksta dokumentacije.*

# Dodatak B: Dnevnik sastajanja

*U ovom dijelu dodatka potrebno je prenijeti sadržaj dnevnika sastajanja, tj. datoteke 'dnevnik.txt' iz repozitorija grupe. Datoteku 'dnevnik.txt' potrebno je voditi kroz repozitorij, a ovdje će to biti preneseno kao dio konačne dokumentacije.*

# Dodatak C: Prikaz aktivnosti grupe

*U ovom dijelu dodatka potrebno je navesti*

* *popis članova grupe i njihovih zaduženja, tj. odrađenih poslova (u postocima ukupno odrađenog posla)*
* *prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je u* BitBucket *sučelju instalirati dodatak* AwesomeGraph *koji daje prikaz grafički prikaz aktivnosti grupe, tj. '*commit' *akcija tijekom trajanja projekta, te na kraju projekta generirane grafove prenijeti u ovo poglavlje dokumentacije.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Popis aktivnosti** | **Članovi grupe** (abecednim redom) | | | | | | |
| Ime Prezime voditelja | Ime Prezime 2 | Ime Prezime 3 | Ime Prezime 4 | Ime Prezime 5 | Ime Prezime 6 | Ime Prezime 7 |
| **Upravljanje projektom** |  |  |  |  |  |  |  |
| **Opis projektnog zadatka** |  |  |  |  |  |  |  |
| **Rječnik pojmova** |  |  |  |  |  |  |  |
| **Opis funkcionalnih zahtjeva** |  |  |  |  |  |  |  |
| **Opis ostalih zahtjeva** |  |  |  |  |  |  |  |
| **Arhitektura i dizajn sustava** |  | | | | | | |
| Svrha, opći prioriteti i skica sustava |  |  |  |  |  |  |  |
| Dijagram razreda s opisom |  | | | | | | |
| Dijagram objekata |  |  |  |  |  |  |  |
| Ostali UML dijagrami |  |  |  |  |  |  |  |
| **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** |  |  |  |  |  |  |  |
| **Zaključak i budući rad** |  |  |  |  |  |  |  |
| **Popis literature** |  |  |  |  |  |  |  |
| **Dodaci** |  | | | | | | |
| Indeks |  |  |  |  |  |  |  |
| Dnevnik sastajanja |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

*Napomena: Doprinose u aktivnostima treba navesti u postocima po članovima grupe. Zbroj postotaka u svakom retku treba biti 100%.*

Pregled pohrana kroz vrijeme trajanja projekta (primjer):



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

*U ovom poglavlju potrebno je navesti:*

* *(u rev. 1) koji je plan rada za rev. 2,*
* *(u rev. 2) koji je status implementacije u odnosu na postavljene ciljeve, procjenu vremena dovršetka projekta (ako zadatak nije u potpunosti ispunjen), koje bi bile smjernice za daljnji rad kad bi se nastavilo s projektom te u čemu bi se sastojale buduće nadogradnje.*