



PROJEKTNİ ZADATAK

Predmet: **KI301 Konstruisanje softvera**

Polaznik: **Branislav Manojlović kp11**
Datum: **14.06.2018.**

Sadržaj:

Opis problema	3
Model zahteva	5
Slučajevi korišćenja za korisničke role prikazani use-case dijagramima	8
Opis slučajeva korišćenja	9
Dijagram arhitekture	10
Sekvencijalni model za korisnika Rentijera kao glavni model	11
Glavni Klasni model za korisnika Rentijera proistekao iz Sekvencijalnog modela	12
Klasni model za korisnika Rentijera pri kreiranju parkinga za izdavanje	12
Klasni model za korisnika Rentijera pri pregledu detalja parkinga	12
Modelovanje klasnih metoda pomoću pseudokoda	13
Kreiranje aplikacije	15
Test slučajevi za testiranje Modula I i Modula II	17
Debagovanje aplikacije upotrebom debbugera u NetBeans-u	25
Test slučajevi Integracionog Testiranja Modula I i Modula II	28

Opis problema

Klijent je naručio izradu aplikacije koja će se koristiti za potrebe izdavanja i zakupa parking mesta građanima.

Aplikacija je namenjena korisnicima Rentijerima koji imaju parking mesta za iznajmljivanje i koji putem aplikacije kreiraju oglas za izdavanje mesta kojima raspolažu, unose osnovne podatke o parking mestu koji uključuju: adresu parkinga, sliku parkinga, cenu izdavanja parkinga, kao i datum i vreme raspoloživosti parkinga, mogu da vide spisak parkinga kojima raspolažu, zatim spisak rezervacija koje su izvršene na njihovim parkinzima, mogu da vide Rewieve ostavljene od strane korisnika Zakupaca o njihovim parkng mestima, kao i novčani prihod koji pri tome ostvaruju. Pored njih postoje i korisnici Zakupci koji imaju mogućnost pretraživanja raspoloživih parkinga koji su u ponudi i mogu da iznajme parkinge koji su slobodni. Takođe, Zakupci mogu da vrše pretragu parkinga po adresama i prema vremenskom periodu, mogu da otkazu rezervaciju ukoliko se predomisle i ne žele više da zadrže rezervaciju na datom parking, mogu da ostave Review o parkingu koji su koristili, mogu da vide Review-ove ostavljene od strane drugih korisnika i mogu da vide listu svih parkinga koje su oni sami rezervisali. Obe grupe korisnika imaju mogućnosti kreiranja korisničkih profila. Korisnički profil sadrži sledeće informacije: ime, prezime, email, kontakt telefon, sliku, podatak o raspoloživim novčanim sredstvima korisnika.

Podaci koje korisnici unose se čuvaju u bazi podataka. Parkinzi se prikazuju korisnicima pomoću Google mape, tačnije u vidu pinova na mapi. Svaki parking je označen bojom. Boje parkinga se razlikuju u zavisnosti od statusa parkinga i to: Slobodan – zelena boja, Zauzet – crvena boja.

Za ovaj problem, treba da uradite sledeće zadatke:

1. Definišite model zahteva. Navesti funkcionalne i nefunkcionalne zahteve i ukratko ih opisati.
2. Definišite arhitekturu sistema i kratko navedite funkciju svake komponente te arhitekture. Obrazložite zašto ste izabrali baš takvu arhitekturu.
3. Detaljno definišite sve UML slučajeve korišćenja.
4. Identifikovati glavne objekte sistema i modelovati sekvencijalni dijagram.
5. Na osnovu sekvencijalnog dijagrama modelovati klasni dijagram.

Nastavak projekta koji se odnosi na predmet KI301

Kroz projektni zadatak potrebno je obuhvatiti:

- Naslov i kratak opis problema
- Kratka specifikacija korisničkih zahteva (ukratko da bi se stekao uvid u aplikaciju o kojoj se govori)
- Kratak opis arhitekture aplikacije (razviti aplikaciju tako da ima minimum pet komponenti)
- Modelovati klasni dijagram aplikacije

- Modelovanje klasnih metoda pomoću pseudokoda
- Izvršiti podelu aplikacije na module (predstaviti svaki modul odvojenim klasnim dijagramom)
- Modelovati sekvencijalni dijagram aplikacije
- Na osnovu klasnog i sekvencijalnog dijagrama izvršiti implementaciju (minimum dva modula u Java programskom jeziku)
- Osmisliti pet test slučajeva i predstaviti ih tabelom (odabrati ad hoc ili deterministički način)
- Izvršiti testiranje, debugovanje i integraciju aplikacije
- Osmisliti pet test slučajeva integracionog testiranja i predstaviti ih tabelom (korsteći top down ili bottom up tehniku)

Za odabrani problem potrebno je prvo ukratko opisati problem a zatim izvršiti specifikaciju korisničkih zahteva. Na osnovu korisničkih zahteva potrebno je izvršiti modelovanje dijagrama arhitekture kao i klasnog dijagrama koji prikazuje sve klase unutar aplikacije. Potrebno je modelovati klasne metode pomoću pseudokoda. Sledeći korak je podela sistema na module gde je prvobitno modelovani klasni dijagram potrebno podeliti na module (odvojene klasne dijagrame koji prikazuju klase tog modula i veze između klasa modula). Modelovati i sekvencijalni dijagram aplikacije. Izvršiti implementaciju minimum dva modula u Java programskom jeziku. Nakon toga u okviru projektnog zadatka potrebno je izvršiti definisanje test slučajeva. Prvo je potrebno osmisliti pet test slučajeva (ad hoc ili determinističkim načinom testiranja) i pet test slučajeva integracionog testiranja (korišćenjem top down ili bottom up tehnike). Izvršiti debugovanje i prikazati rezultate debugovanja.

Pri radu, koristite Power Designer i NetBeans.

Korisničke role:

- Rentijer parking mesta,
- Zakupac parking mesta.

Funkcionalni zahtevi – korisnička rola Rentijer:

1. Rentijer treba da ima mogućnost kreiranja korisničkog profila,
2. Rentijer treba da ima mogućnost kreiranja parking mesta za iznajmljivanje,
3. Rentijer treba da ima mogućnost određivanja cene iznajmljivanja parkinga,
4. Rentijer treba da ima mogućnost izmene cena iznajmljivanja parkinga,
5. Rentijer treba da ima mogućnost pregleda liste svojih iznajmljenih parkinga,
6. Rentijer treba da ima mogućnost pregleda Review-a ostavljenih od strane Zakupaca,
7. Rentijer treba da ima mogućnost pregleda pristiglih novčanih sredstava od iznajmljenih parkinga,
8. Rentijer treba da ima mogućnost brisanja kreiranih parking mesta iz ponude.

Funkcionalni zahtevi – korisnička rola Zakupac:

1. Zakupac treba da ima mogućnost kreiranja korisničkog profila,
2. Zakupac treba da ima mogućnost pretrage parkinga,
3. Zakupac treba da ima mogućnost pretrage parkinga po adresi.
4. Zakupac treba da ima mogućnost pretrage parkinga po datumu izdavanja,
5. Zakupac treba da ima mogućnost pregleda detalja parkinga,
6. Zakupac treba da ima mogućnost pregleda ostavljenih Review-a od strane drugih zakupaca,
7. Zakupac treba da ima mogućnost rezervisanja parkinga,
8. Zakupac treba da ima mogućnost plaćanja parkinga karticom on-line,
9. Zakupac treba da ima mogućnost ostavljanja Review-a o parkingu koji je koristio nakon isteka rezervacije,
10. Zakupac treba da ima mogućnost pregleda učinjenih rezervacija,
11. Zakupac treba da ima mogućnost otkazivanja rezervacije.

Funkcionalni zahtevi – Sistem:

1. Sistem treba da ima mogućnost generisanja dokaza o izvršenoj novčanoj transakciji od strane Zakupca, tj. o plaćanju parkinga,
2. Sistem treba da ima mogućnost pozivanja Google map servisa i prikazivanja parking na mapi.

Nefunkcionalni zahtevi:

- Zahtevi proizvođača: Sistem treba da bude dostupan korisnicima preko mobilnih uređaja koji rade sa Android operativnim sistemom a razvija se i kao servletska web aplikacija. Sistem mora imati konekciju sa internetom kako bi pravilno funkcionisao.
- Zahtevi organizacije: nema.
- Spoljni zahtevi: nema.

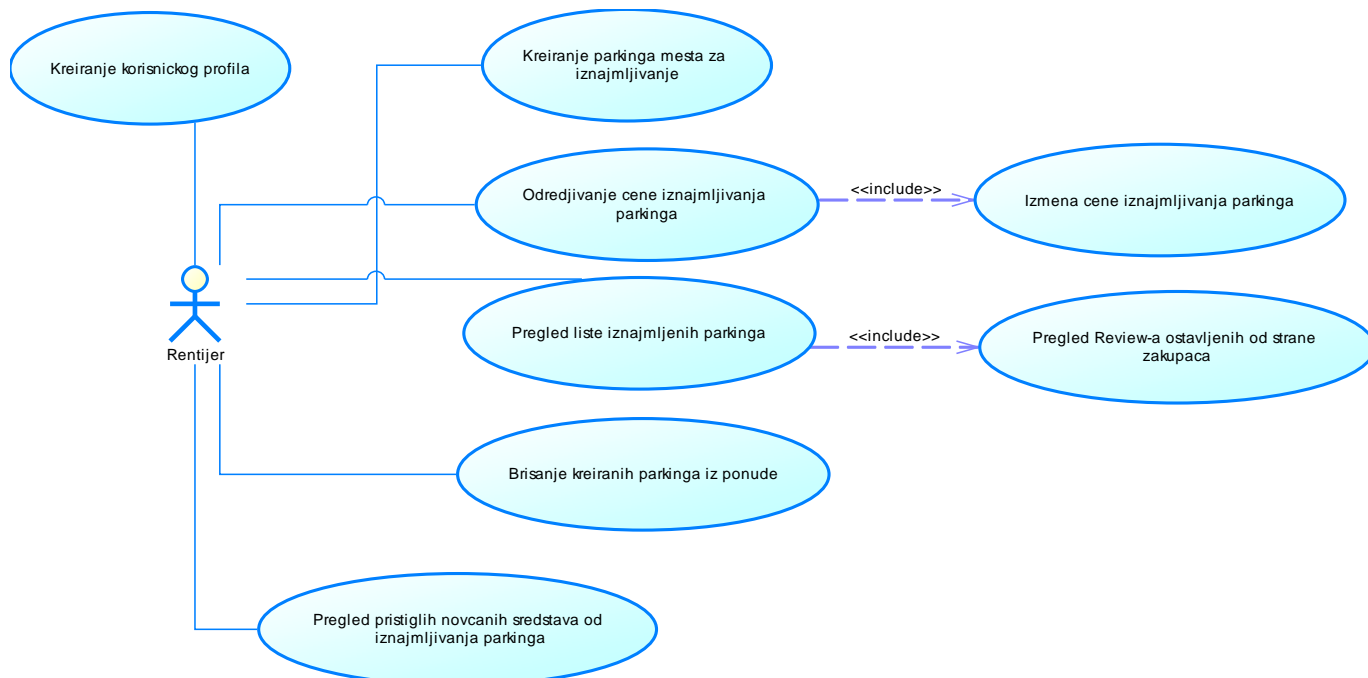
Model zahteva

Title ID	Full Description
1	Overview
2	Use Cases
3	Identified Users
3.1	Rentijer
3.2	Zakupac
4	Identified Use Cases
4.1	Rentijer
4.1.1	Rentijer treba da ima mogucnost kreiranja korisnickog profila Rentijer treba da ima pristup sistemu Rentijer treba da upotrebi jedinstveno korisnicko ime, email i lozinku
4.1.2	Rentijer treba da ima mogucnost kreiranja parking mesta za iznajmljivanje Rentijer treba da ima pristup sistemu Rentijer treba da ima minimum jedan parking za iznajmljivanje
4.1.3	Rentijer treba da ima mogucnost odredjivanja cene iznajmljivanja parkinga Rentijer treba da ima pristup sistemu Rentijer treba da je izdao bar jedan parking Rentijer treba da nije vec prethodno postavljao cenu parkinga
4.1.4	Rentijer treba da ima mogucnost izmene cena iznajmljivanja parkinga Rentijer treba da ima pristup sistemu Rentijer treba da je izdao bar jedan parking Rentijer treba da jeste vec prethodno postavljao cenu parkinga
4.1.5	Rentijer treba da ima mogucnost pregleda liste svojih iznajmljenih parkinga Rentijer treba da ima pristup sistemu Rentijer treba da je izdao jedan ili vise parkinga Rentijer treba da nije obrisao parkinge koje je kreirao

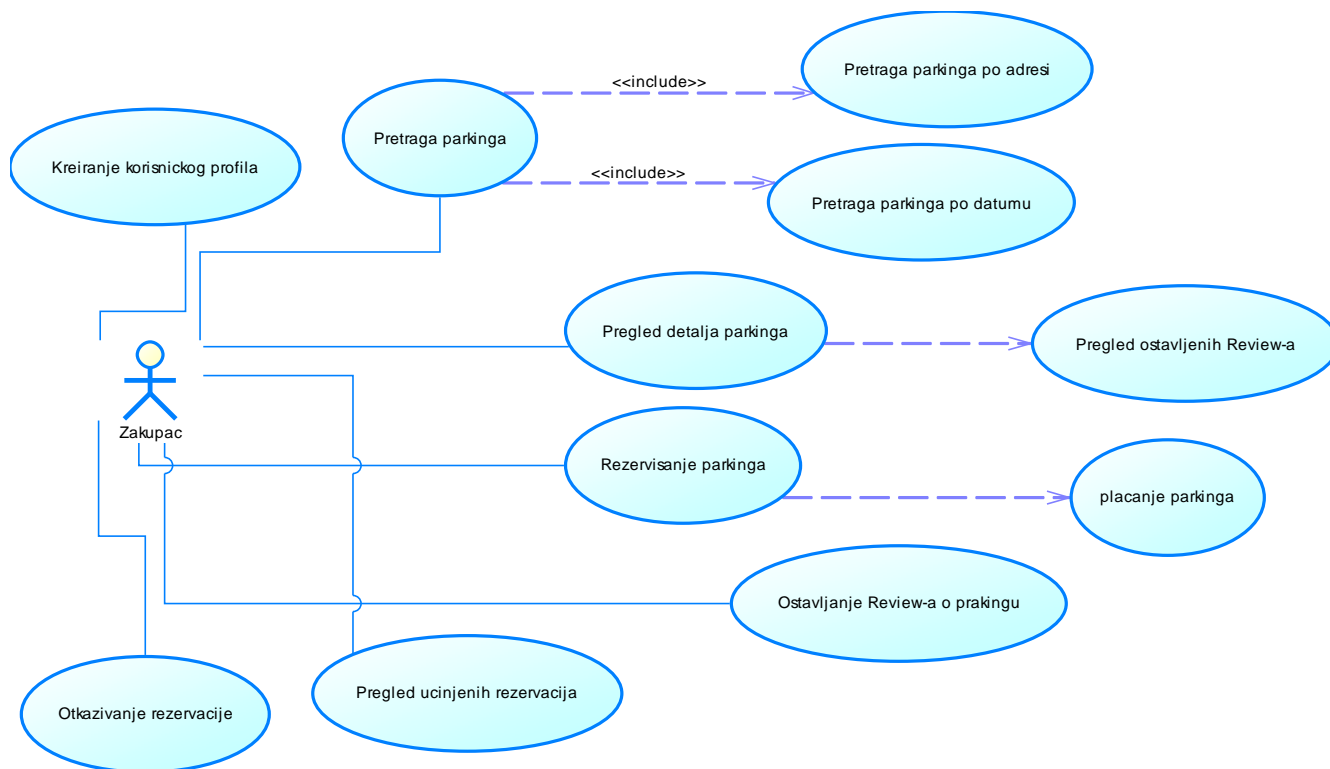
4.1.6	Rentijer treba da ima mogućnost pregleda Review-a ostavljenih od strane Zakupaca Rentijer treba da ima pristup sistemu Rentijer treba da je izdao bar jedan parking Zakupac koji je rezervisao parking od Rentijera treba da je ostavio bar jedan Review
4.1.7	Rentijer treba da ima mogućnost pregleda pristiglih novčanih sredstava iznajmljenih parkinga Rentijer treba da ima pristup sistemu Rentijer treba da je uspešno izdao bar jedan parking Zakupac treba da je platio novčani iznos Rentijeru za izdavanje parkinga
4.1.8	Rentijer treba da ima mogućnost brisanja kreiranih parking mesta iza ponude Rentijer treba da ima pristup sistemu Rentijer treba da je kreirao bar jedan parking
4.2	Zakupac
4.2.1	Zakupac treba da ima mogućnost kreiranja korisničkog profila Zakupac treba da ima pristup sistemu Zakupac treba da upotrebi jedinstveno korisničko ime, email i lozinku
4.2.2	Zakupac treba da ima mogućnost pretrage parkinga Zakupac treba da ima pristup sistemu Zakupac treba da upotrebi neki od kriterijuma za pretragu, setovanje datuma ili adrese
4.2.3	Zakupac treba da ima mogućnost pretrage parkinga po adresi Zakupac treba da ima pristup sistemu Zakupac treba da upotrebi kao kriterijum za pretragu, setovanje adrese
4.2.4	Zakupac treba da ima mogućnost pretrage parkinga po datumu izdavanja Zakupac treba da ima pristup sistemu Zakupac treba da upotrebi kao kriterijum za pretragu, setovanje datuma
4.2.5	Zakupac treba da ima mogućnost pregleda detalja parkinga Zakupac treba da ima pristup sistemu Trebalo bi da postoji barem jedan kreiran parking od strane jednog ili više Rentijera

4.2.6	Zakupac treba da ima mogucnost pregleda ostavljenih Review-a o parkingu od strane drugih zakupaca Zakupac treba da ima pristup sistemu U sistemu treba da postoji jedan ili vise parkinga koji su iznajmljivani i za koje je ostavljan Review
4.2.7	Zakupac treba da ima mogucnost rezervisanja parkinga Zakupac treba da ima pristup sistemu Zakupac treba da ima novcana sredstva sa kojima moze da izvrši plaćanje
4.2.8	Zakupac treba da ima mogucnost plaćanja parkinga on-line karticom Zakupac treba da ima pristup sistemu Zakupac treba da ima novcana sredstva i karticu za internet plaćanje sa kojima moze putem on-line servisa da izvrši plaćanje
4.2.9	Zakupac treba da ima mogucnost ostavljanja Review-a o parkingu koji je koristio nakon isteka rezervacije Zakupac treba da ima pristup sistemu Zakupac treba da je rezervisao i platio parking
4.2.10	Zakupac treba da ima mogucnost pregleda učinjenih rezervacija Zakupac treba da ima pristup sistemu Zakupac treba da je ima jednu ili više izvršenih rezervacija u prošlosti
4.2.11	Zakupac treba da ima mogucnost otkazivanja rezervacije Zakupac treba da ima pristup sistemu Zakupac treba da je izvršio rezervaciju nekog parkinga Zakupac treba da je pomoću odgovarajuće forme obustavio rezervisanje parkinga

Slučajevi korišćenja za korisničke role, prikazani dijagramima slučajeve korišćenja (*Use-Case dijagramima*)

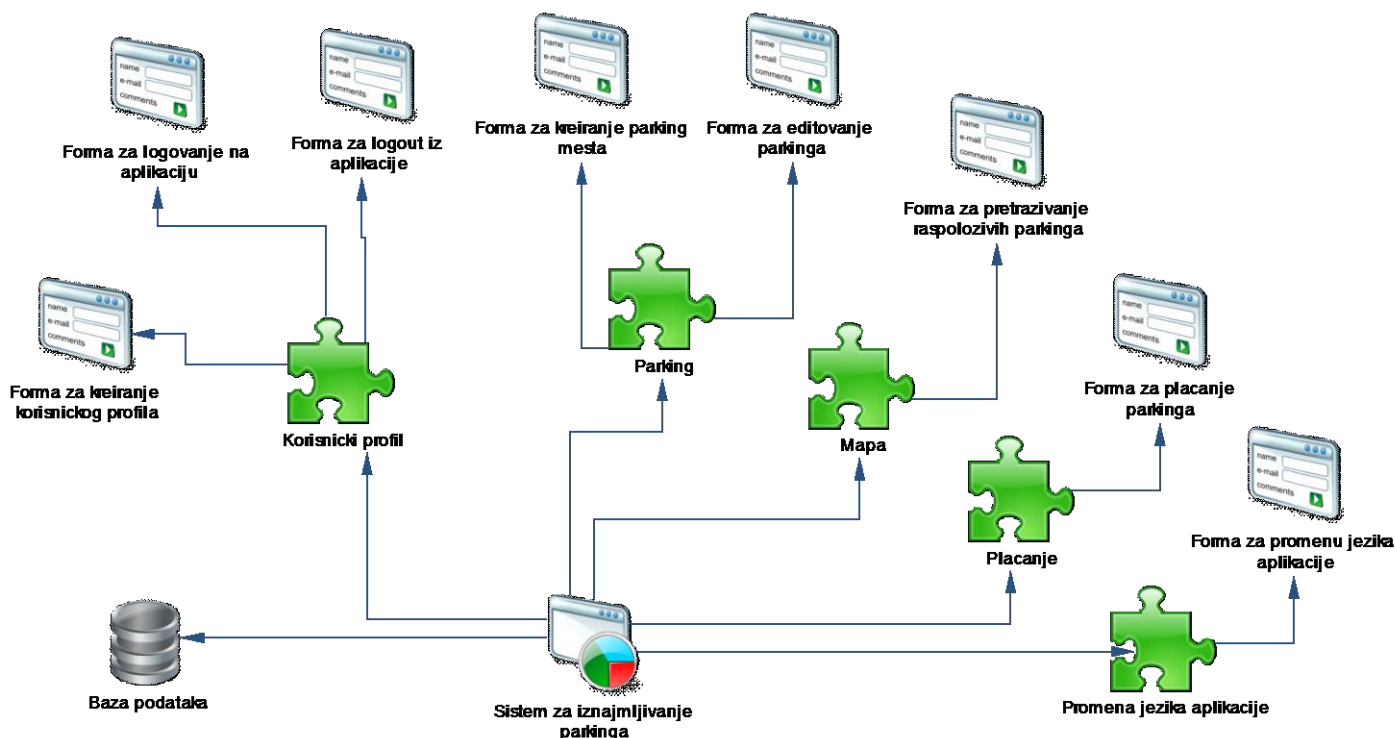


Slika 1. Use-Case Dijagram Rentier Parkinga



Slika 2. Use-Case Diagram Zakupac Parkinga

Dijagram Arhitekture



Slika 3. Dijagram Arhitekture sistema sa 5 komponentata

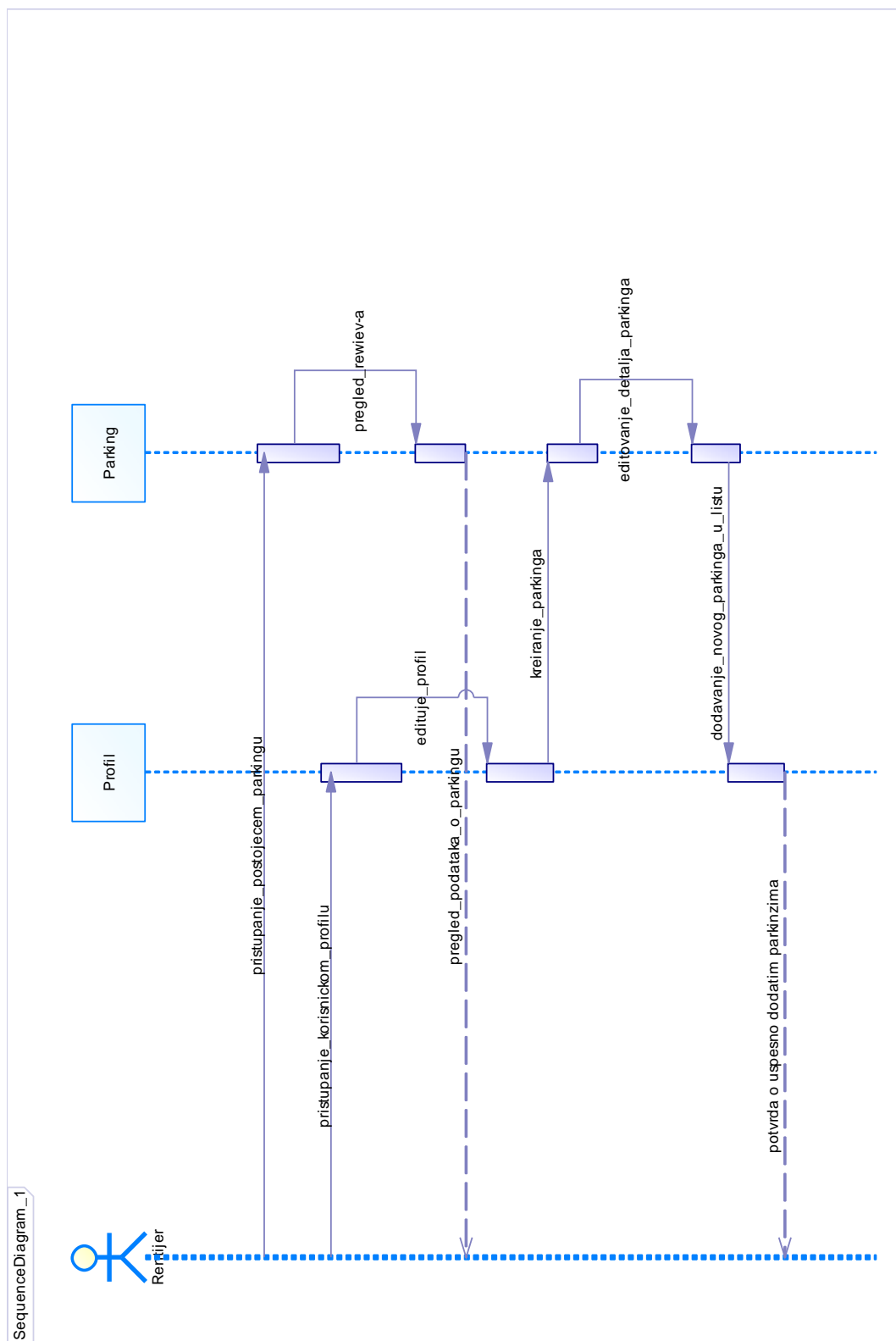
Aplikacija ima pet komponentata od kojih svaka ima odgovarajuće forme putem kojih korisnici komuniciraju sa sistemom. Pošto su dve korisničke role suštinski nezavisne jedna od druge, moguće je podeliti sistem na module prema ovim dvema korisničkim grupama.

Tako da možemo posmatrati sistem iz dve perspektive i to:

- Korisnika Rentijera i
- Korisnika Zakupca.

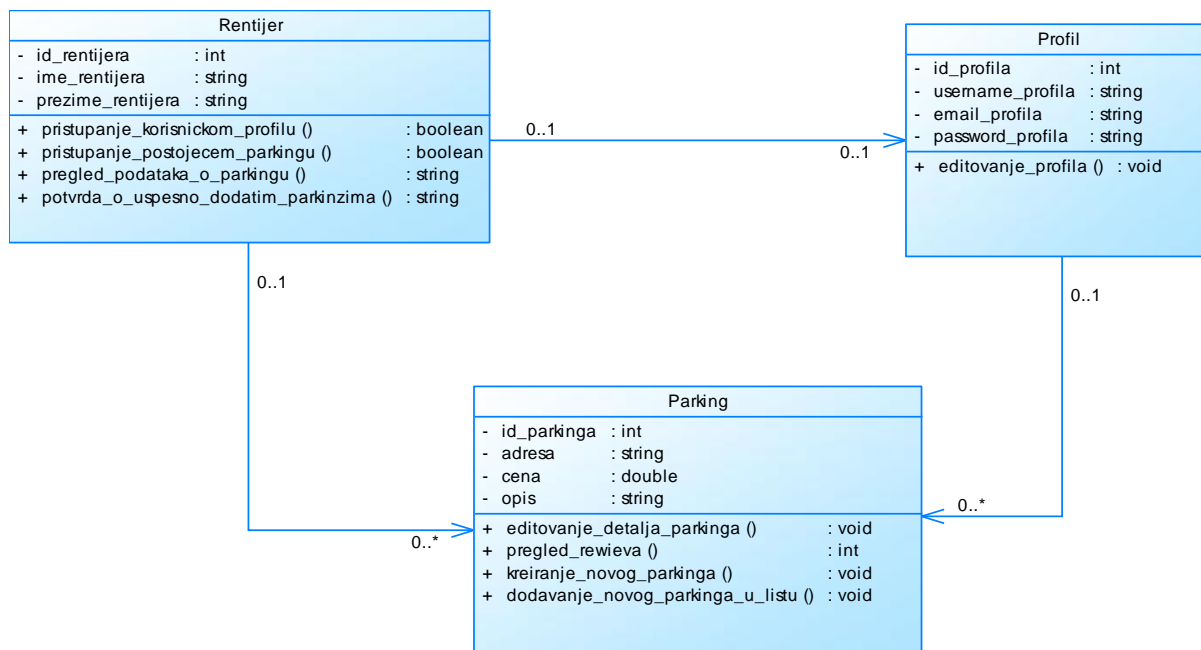
Pošto korisnik Zakupac ima obavezu da plaća parking prilikom zakupa, to je ova perspektiva previše kompleksna za opis u projektu, pa sam se ja opredelio da posmatram sistem samo kroz perspektivu korisnika Rentijera koji kreira i iznajmljuje parking.

Sekvencijalni model za korisnika Rentijera kao glavni model



Slika 4. Sekvencijalni model dela sistema oko korisnika Rentijera

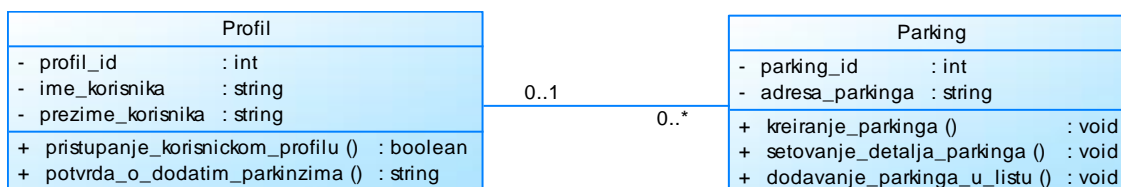
Glavni Klasni model za korisnika Rentijera proistekao iz glavnog Sekvencijalnog modela



Slika 5. Glavni klasni dijagram odnosno glavni Modul ovog dela sistema

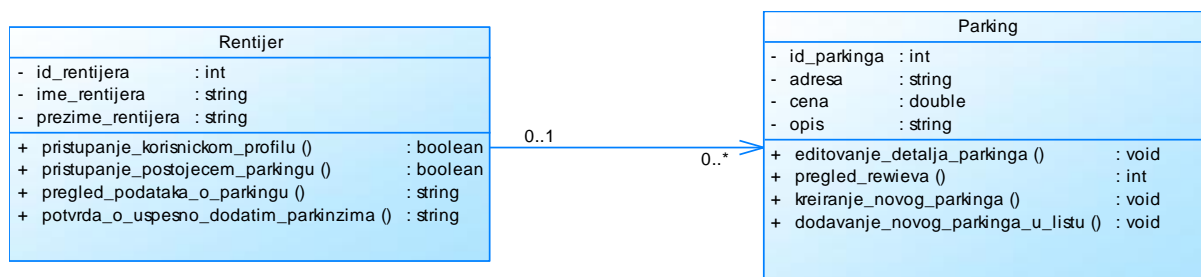
Prikazani Glavni klasni model se može podeliti na dva manja modela koji se mogu posmatrati i testirati kao zasebne celine.

Klasni model za korisnika Rentijera pri kreiranju parkinga za izdavanje



Slika 6. Modul I

Klasni model za korisnika Rentijera pri pregledu detalja parking sa kojima raspolaže



Slika 7. Modul II

Modelovanje klasnih metoda pomoću pseudokoda

1. Pseudokod algoritma za pristup i editovanje korisničkog profila:

korak 1: START programme,
korak 2: CLICK Login to user account,
korak 3: INPUT valid Username,
korak 4: INPUT valid Password,
korak 5: IF(Username==TRUE && Password==TRUE),
korak 6: User exists,
korak 7: INSPECT User profile,
korak 8: EDIT User profile,
korak 9: SAVE changes,
korak 10: BACK to Homescreen,
korak 11: ELSE User don't exists,
korak 12: BACK to Homescreen,
korak 13: CHOOSE option SIGN UP or CREATE PROFILE,
korak 13: IF(SIGN UP ==TRUE),
korak 14: CREATE User PROFILE,
korak 15: ELSE Exit programme,
korak 16: END programme.

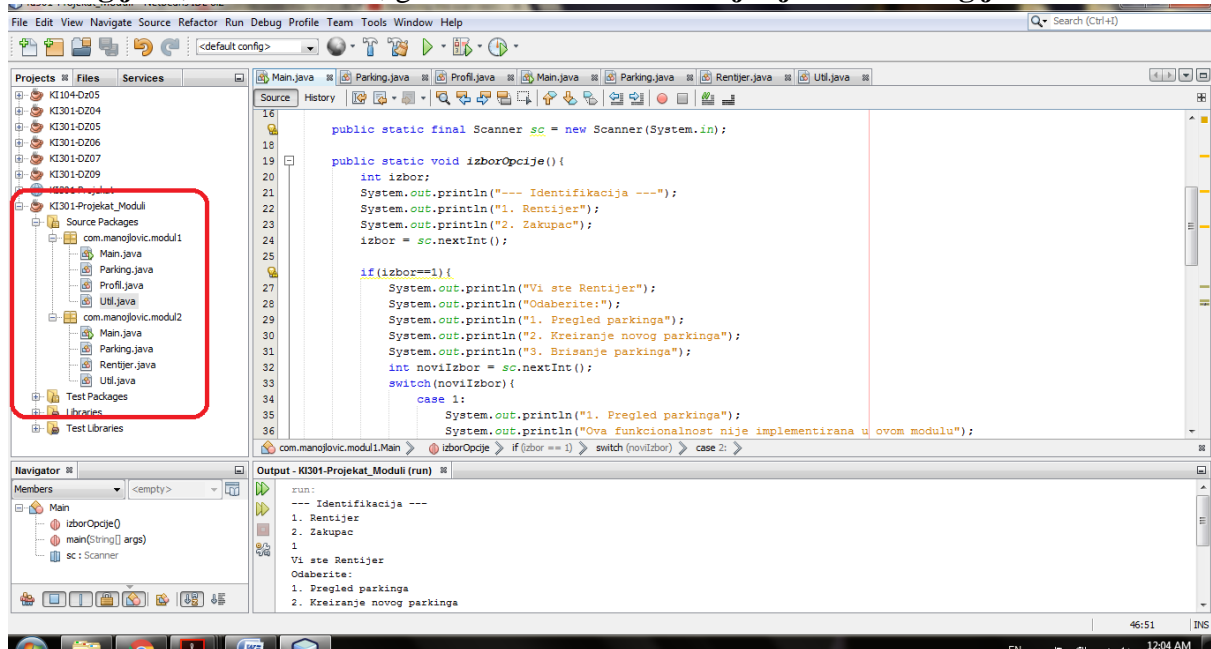
2. Pseudokod algoritma za pristup i editovanje parkinga:

korak 1: START programme,
korak 2: CLICK Login to user account,
korak 3: INPUT valid Username,
korak 4: INPUT valid Password,
korak 5: User exists,
korak 6: INSPECT User profile,
korak 7: CLICK OPEN List of User parking places,

korak 8: IF(List == TRUE),
korak 9: SHOW List of User parking places,
korak 10: CLICK to choose parking,
korak 11: Parking details OPENS UP,
korak 12: CLICK to edit parking details,
korak 13: EDIT User parking details,
korak 14: SAVE changes,
korak 15: ELSE List of User parking places don't exists,
korak 16: CREATE new parking place,
korak 17: SAVE new parking place,
korak 18: BACK to Homescreen,
korak 19: CHOOSE option Exit or OPEN List of User parking places,
korak 20: IF(OPEN == TRUE),
korak 21: SHOW List of User parking places,
korak 22: ELSE Exit programme,
korak 23: END programme.

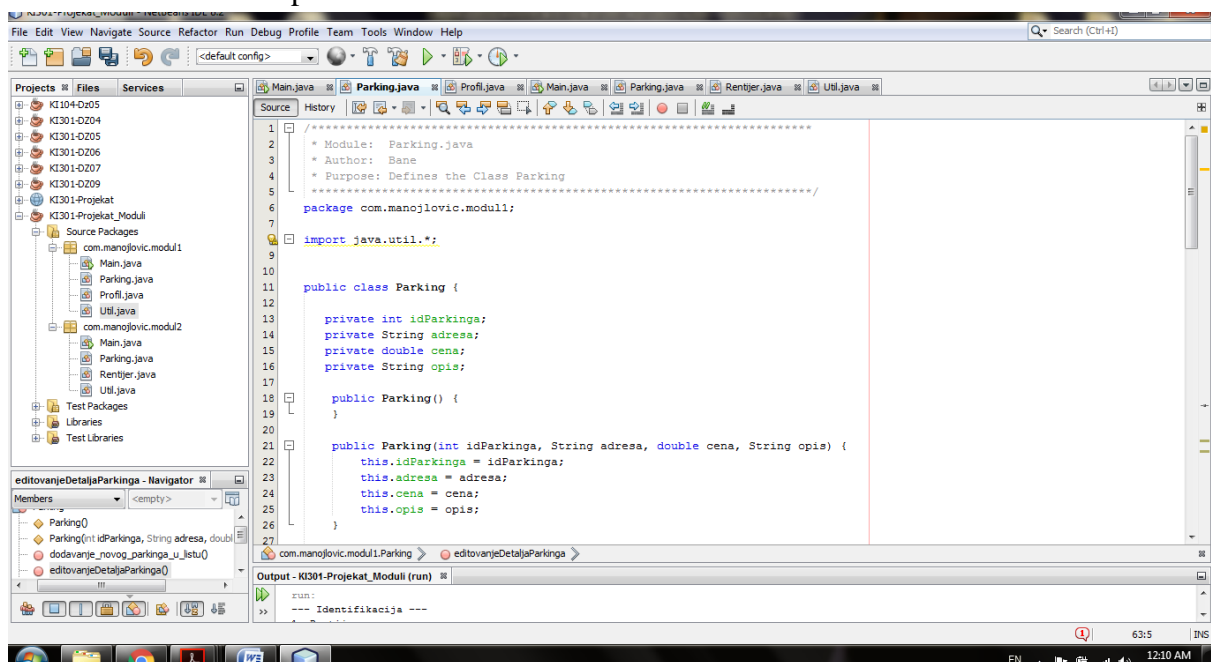
Kreiranje aplikacije

Na osnovu klasnog i sekvencijalnog dijagrama cemo izvršiti implementaciju (minimum dva modula u Java programskom jeziku). Na osnovu modula generišemo Java programski kod i kreiramo aplikaciju. Za **Modul I**, generisane su klase: **Profil.java** i **Parking.java**, a za **Modul II**, generisane su klase: **Rentijer.java** i **Parking.java**.

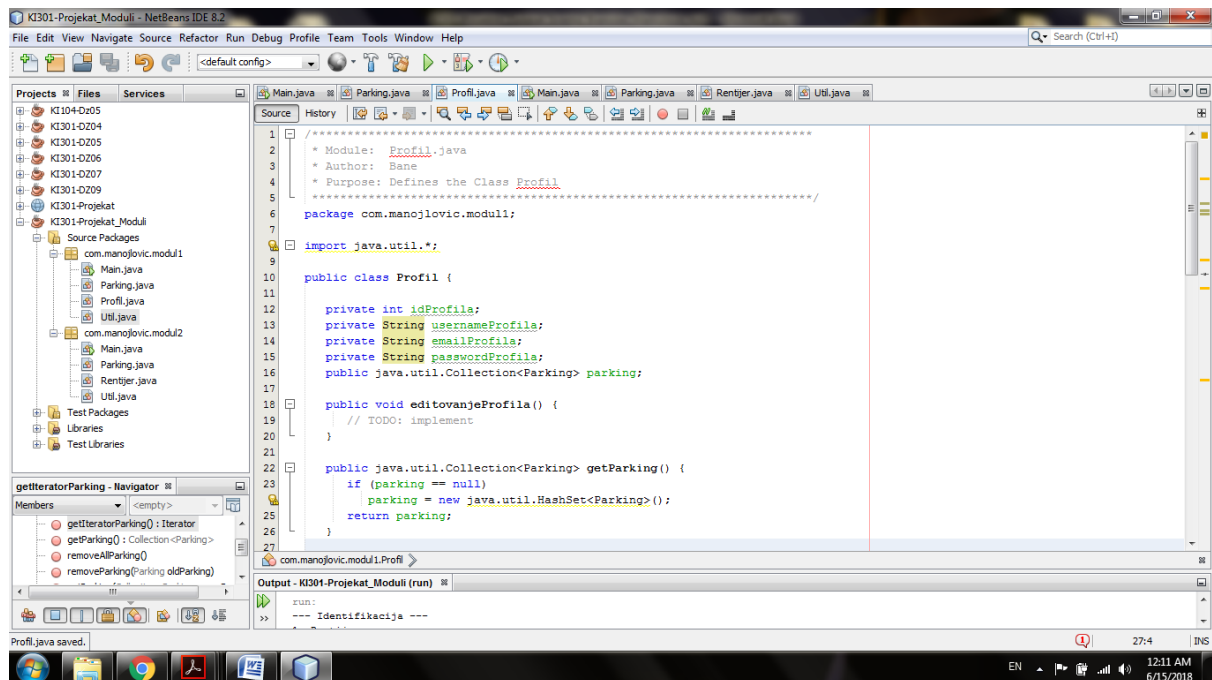


Slika 8. Implementacija generisanih klasa u Java programu

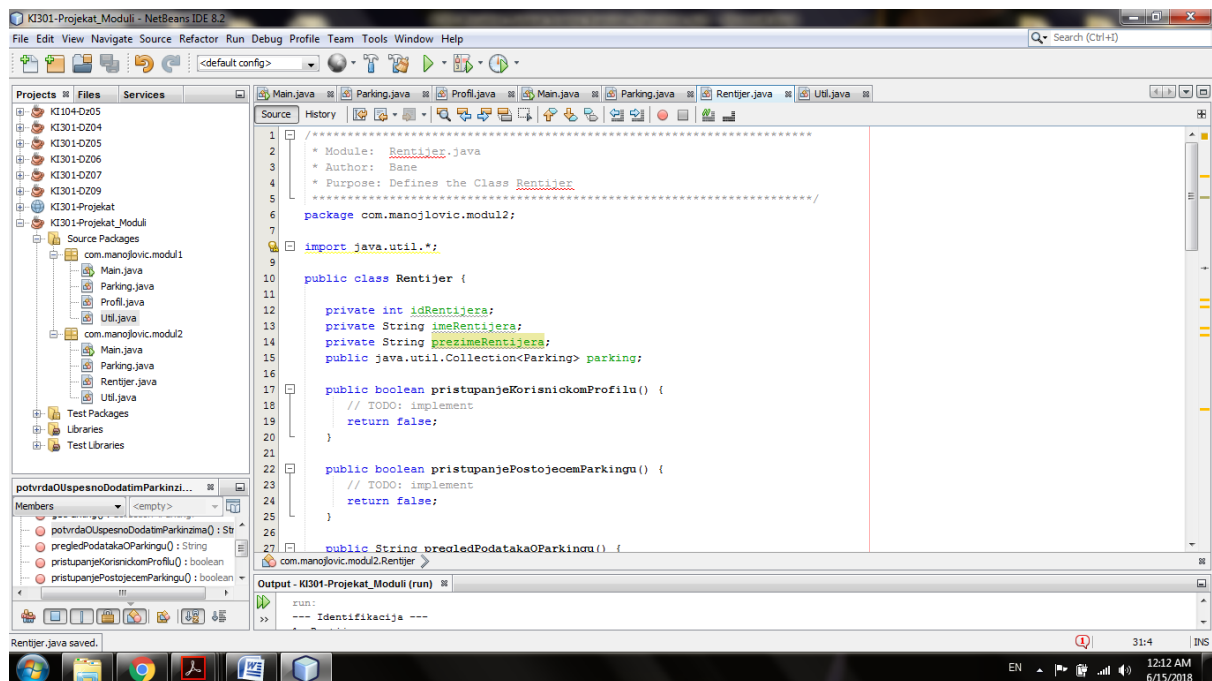
Kreirao sam dva paketa, sa sopstvenim **Main.java** pokretačkim klasama, kako bih testirao svaki od modula posebno. Takođe, sam dodao i **Util.java** klase koje mi omogućavaju pisanje podataka koje unosi korisnik u tekstualni fajl **parkinzi.txt**, koji mi za potrebe ovog zadatka simulira bazu podataka.



Slika 9. Klasa Parking.java generisana iz klasnog dijagrama sa svim atributima



Slika 10. Klasa Profil.java generisana iz klasnog dijagrama sa svim atributima



Slika 11. Klasa Rentijer.java generisana iz klasnog dijagrama sa svim atributima

Test slučajevi za testiranje Modula I i Modula II

Naslov	Kreiranje parkinga	Rev 01	Autor	Branislav	Datum	14.06.2018
Cilj	Verifikacija forme za kreiranje parkinga		Reference	/		
Test uslovi	/	Vreme neophodno za izradu test slučaja	5 min	Neophodno vreme za izvršenje test slučaja		8 min

	Opis postavke za testiranje
•	Aplikacija je instalirana na uređaj
•	Korisnik je ulogovan kao Rentijer
•	Korisnik odabira opciju za kreiranje parkinga

Definicija testa				Izvršenje testa	
	Uslovi	Ulazni podaci	Očekivani rezultati	Aktuelni rezultati	Broj problema
•	Unet negativan ID broj parkinga	invalid	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik ostavio polje prazno	null	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik uneo ID broj u nevalidnom formatu	Double, float, string	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik uneo validan ID broj	int	Kursor aplikacije prešao u naredno polje	Kursor aplikacije prešao u naredno polje	0
•	Korisnik je ostavio prazno polje za adresu	null	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik je uneo adresu u nevalidnom formatu	Double, float	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik je uneo validnu adresu	String	Kursor aplikacije prešao u naredno polje	Kursor aplikacije prešao u naredno polje	0
•	Korisnik je ostavio	null	Poruka o grešci	Prikazana poruka o grešci sa dodatnim	0

	prazno polje za cenu			objašnjenjem	
•	Korisnik je uneo cenu u nevalidnom formatu	String	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik je uneo negativnu vrednost za cenu	invalid	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik je uneo validnu vrednost za cenu	double	Kursor aplikacije prešao u naredno polje	Kursor aplikacije prešao u naredno polje	0
•	Korisnik je ostavio prazno polje za opis parkinga	null	Polje nije obavezno, poruka o grešci se ne prikazuje	Poruka o grešci se nije prikazala	0
•	Korisnik uneo opis parknga	String	Poruka o grešci se ne prikazuje	Poruka o grešci se ne prikazuje	0

	Opis post uslova
•	Korisnik je uspešno kreirao parking mesto

Naslov	Editovanje parkinga	Rev 02	Autor	Branislav	Datum	14.06.2018
Cilj	Verifikacija forme za Edit parkinga	Reference	/			
Test uslovi	/	Vreme neophodno za izradu test slučaja	10 min	Neophodno vreme za izvršenje test slučaja	10 min	

	Opis postavke za testiranje
•	Aplikacija je instalirana na uređaj
•	Korisnik je ulogovan kao Rentijer
•	Korisnik odabira opciju za Editovanje parkinga

Definicija testa				Izvršenje testa	
	Uslovi	Ulazni podaci	Očekivani rezultati	Aktuelni rezultati	Broj problema
•	Korisnik kliknuo na polje za ID parkinga	null	Polje nije editabilno	Polje nije editabilno	0
•	Korisnik kliknuo na polje za adresu parkinga	null	Polje nije editabilno	Polje nije editabilno	0
•	Korisnik je kliknuo na polje za cenu	double	Polje je editabilno i omogućena je promena cene	Polje je editabilno i omogućena je promena cene	0
•	Korisnik je obrisao cenu i nije uneo novu	null	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik je uneo cenu u nevalidnom formatu	String	Poruka o grešci	Prikazana poruka o grešci sa dodatnim objašnjenjem	0
•	Korisnik je uneo validnu vrednost za cenu	double	Kursor aplikacije prešao u naredno polje, poruka o grešci nije prikazana	Kursor aplikacije prešao u naredno polje, poruka o grešci nije prikazana	0
•	Korisnik je	String	Polje je	Polje je editabilno i	0

	kliknuo na polje za opis parkinga		editabilno i omogućena je promena opisa	omogućena je promena opisa	
•	Korisnik je obrisao opis i nije uneo novi	null	Polje nije obavezno, poruka o grešci se ne prikazuje	Poruka o grešci se nije prikazala	0
•	Korisnik uneo novi opis parkinga	String	Poruka o grešci se ne prikazuje	Poruka o grešci se ne prikazuje	0

	Opis post uslova
•	Korisnik je uspešno editovao parking mesto i sve promene su sačuvane

Naslov	Brisanje parkinga	Rev 03	Autor	Branislav	Datum	14.06.2018
Cilj	Verifikacija forme za brisanje parkinga		Reference	/		
Test uslovi	/	Vreme neophodno za izradu test slučaja	10 min	Neophodno vreme za izvršenje test slučaja		10 min

	Opis postavke za testiranje
•	Aplikacija je pokrenuta
•	Korisnik je ulogovan kao Rentijer
•	Korisnik odabira opciju za brisanje parkinga

Definicijatesta				Izvršenje testa	
	Uslovi	Ulaznipodaci	Očekivani rezultati	Aktuelnirezultati	Brojproblema
•	Korisnik ima bar jedan kreiran parking	true	Poruka o dozvoljenom brisanju parkinga	Briisanje parking je dozvoljeno	0
•	Korisnik nema ni jedan kreiran parking	true	Poruka da za taj korisnički profil ne postoji kreiran ni jedan parking	Aplikacija je prikazala poruku da za taj korisnički profil ne postoji kreiran ni jedan parking	0
•	Odabrani parking ima aktivnu rezervaciju	string	Poruka da brisanje parkinga nije dizvoljeno jer on ima trenutno aktivnu rezervaciju	Prikazana je poruka da brisanje parkinga nije dizvoljeno jer on ima trenutno aktivnu rezervaciju	0

	Opis post uslova
•	Korisnik je uspešno obrisao izabrani parking i on se više ne nalazi u ponudi

Naslov	Pregled Rewiev-a	Rev 04	Autor	Branislav	Datum	14.06.2018
Cilj	Verifikacija forme za pregled ocena parkinga		Reference	/		
Test uslovi	/	Vreme neophodno za izradu test slučaja	10 min	Neophodno vreme za izvršenje test slučaja		3 min

	Opis postavke za testiranje
•	Aplikacija je pokrenuta
•	Korisnik je ulogovan kao Rentijer ili Zakupac
•	Korisnik odabira opciju za pregled parkinga

Definicijatesta				Izvršenje testa	
	Uslovi	Ulazni podaci	Očekivani rezultati	Aktuelni rezultati	Broj problema
•	Odabrani parking ima validno ostavljen Rewiev	true	Ispisan sadržaj Rewiev-a	Prikazan sadržaj Rewiev-a u vidu tekstualne poruke	0
•	Odabrani parking nema ostavljen Rewiev	true	Poruka o tome da parking nema ostavljen Rewiev	Prikazana je poruka o tome da parking nema ostavljen Rewiev	0

	Opis post uslova
•	Korisniku je prikazan sadržaj ostavljenog rewiev-a u vidu tekstualne poruke

Naslov	Pristup profilu	Rev 05	Autor	Nebojša	Datum	14.06.2018
Cilj	Verifikacija forme za pristup korisničkom profilu		Reference	/		
Test uslovi	/	Vreme neophodno za izradu test slučaja	10 min	Neophodno vreme za izvršenje test slučaja		5 min

	Opis postavke za testiranje
•	Aplikacija je pokrenuta
•	Korisnik unosi svoje korisničke podatke
•	Korisnik odabira profilnu stranu

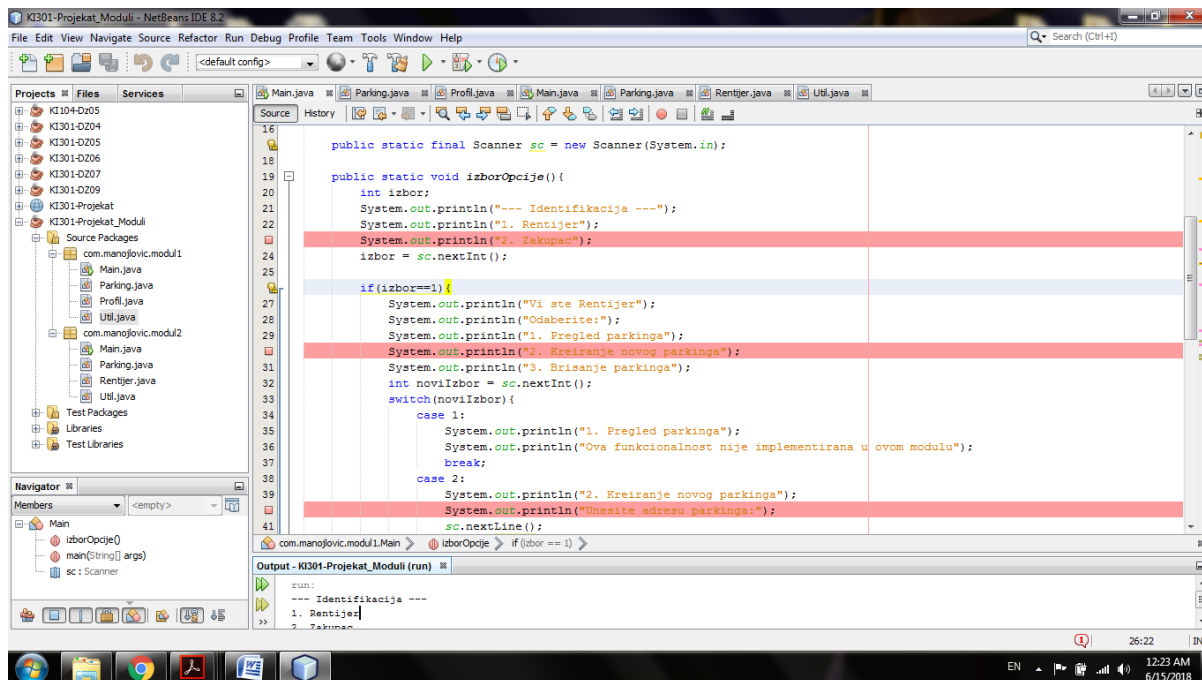
Definicija testa				Izvršenje testa	
	Uslovi	Ulazni podaci	Očekivani rezultati	Aktuelni rezultati	Broj problema
•	Korinik je uneo pogrešno korisničko ime	string	Poruka o grešci	Prikazana je poruka o grešci	0
•	Korisnik je ostvio polje za korisničko ime prazno	null	Poruka o grešci	Prikazana je poruka o grešci	0
•	Korisnik je uneo validno Korisničko ime	string	Poruka o grešci se ne prikazuje i kursor aplikacije prelazi u naredno polje	Poruka o grešci se nije prikazala i kursor aplikacije je prešao u naredno polje	0
•	Korinik je uneo pogrešnu lozinku	string	Poruka o grešci	Prikazana je poruka o grešci	0
•	Korisnik je ostvio polje za lozinku prazno	null	Poruka o grešci	Prikazana je poruka o grešci	0
•	Korisnik je uneo validnu	string	Poruka o grešci se ne prikazuje	Poruka o grešci se nije prikazala	0

	lozinku				
•	Korisnik nije kliknuo dugme za otvaranje profilne stranice već se samo ulogova na aplikaciju	null	Korisnik ostaje na Početnoj stranici aplikacije	Samo početna strana aplikacije je otvorena	0
•	Korisnik je kliknuo dugme za otvaranje profilne stranice	true	Profilna stranica je otvorena	Profilna stranica je otvorena	0

	Opis post uslova
•	Korisniku je prikazana profilna stranica njegovog korisničkog profila

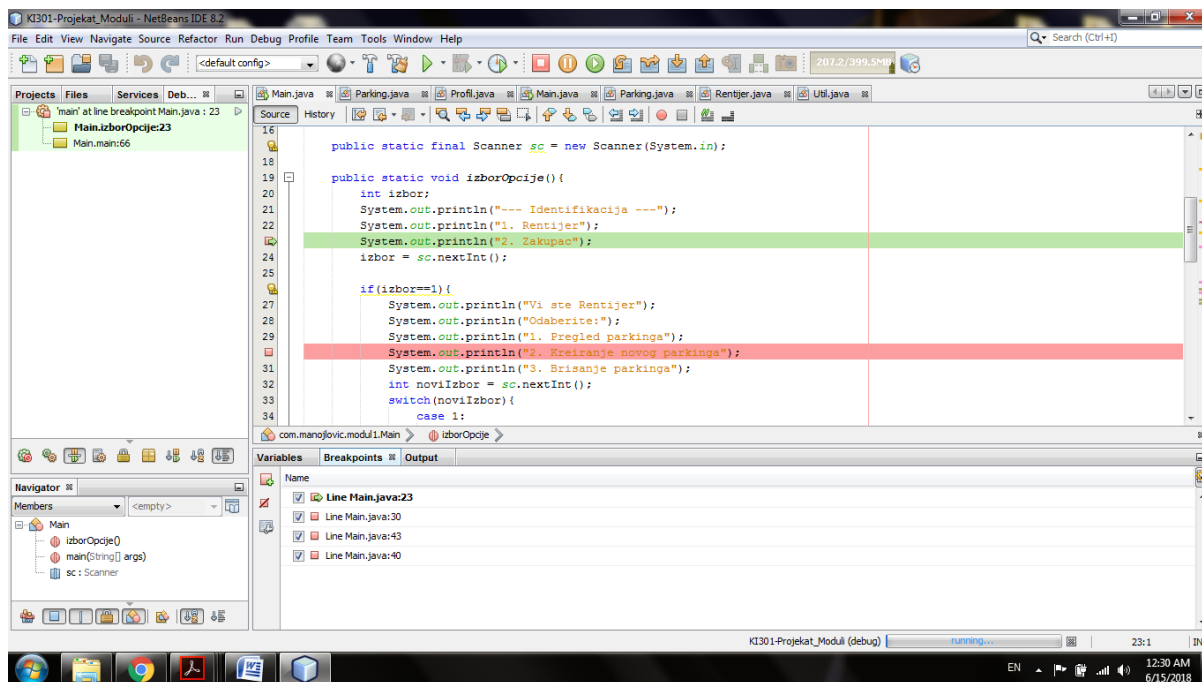
Debagovanje aplikacije upotrebom debbugera u NetBeans-u

1. Postavimo Breakpoints (tačke prekida) na linijama koda za koje sumnjamo da imaju grešku. Idem: desni klik → Breakpoint → Enabled.



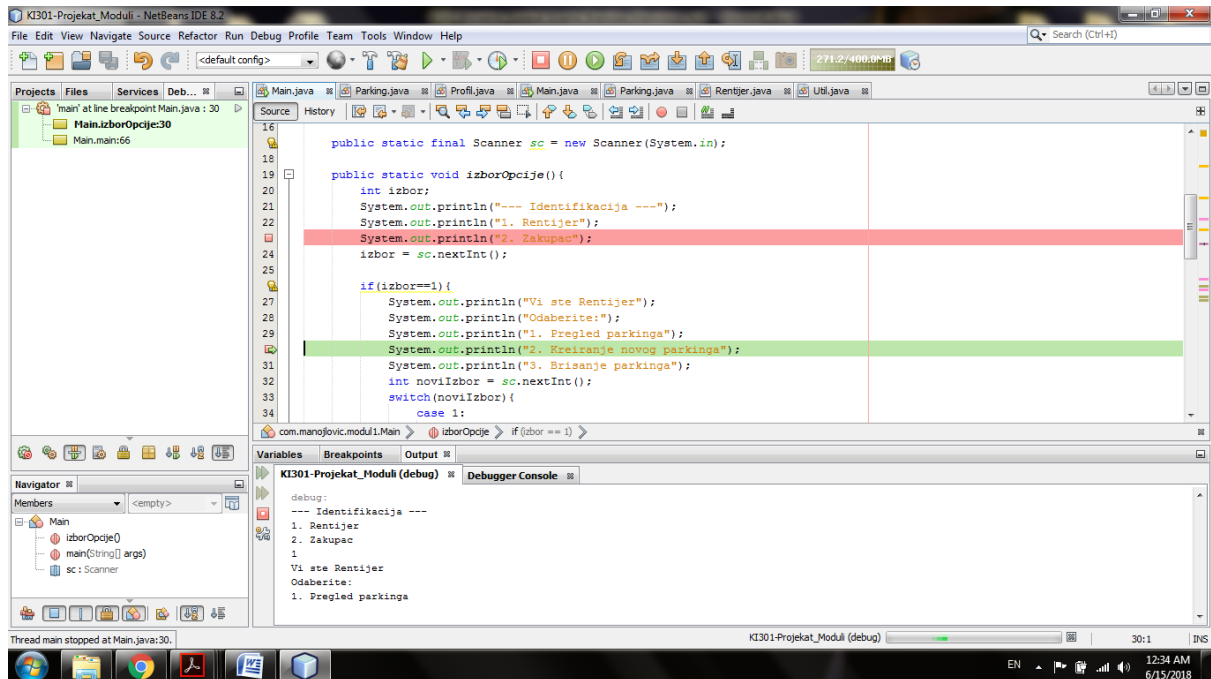
Slika 12. Postavljanje Breakpoint-a na problematičnim mestima u kodu

2. Listu svih Breakpoint-a vidim i na kartici „Breakpoints“ u NetBeans-u,



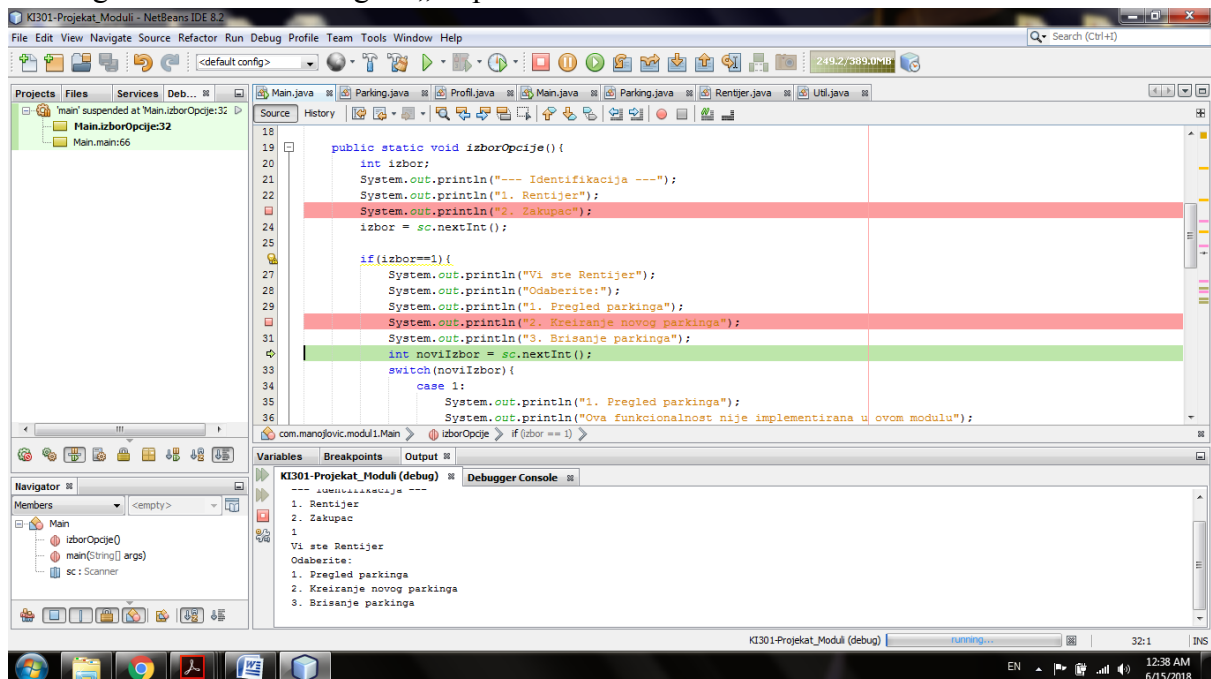
Slika 13. Lista postavljenih Breakpoint-a

3. Program se pusta u rad klikom na dugme “Debug project”. Njegovo izvršavanje i debugovanje traje do prvog Breakpoint-a, kada staje a taj Breakpoint na kom je program stao je oznacen zelenom bojom. Kretanje kroz program i njegovo izvršavanje pratimo u izlaznoj konzoli NetBeans-a



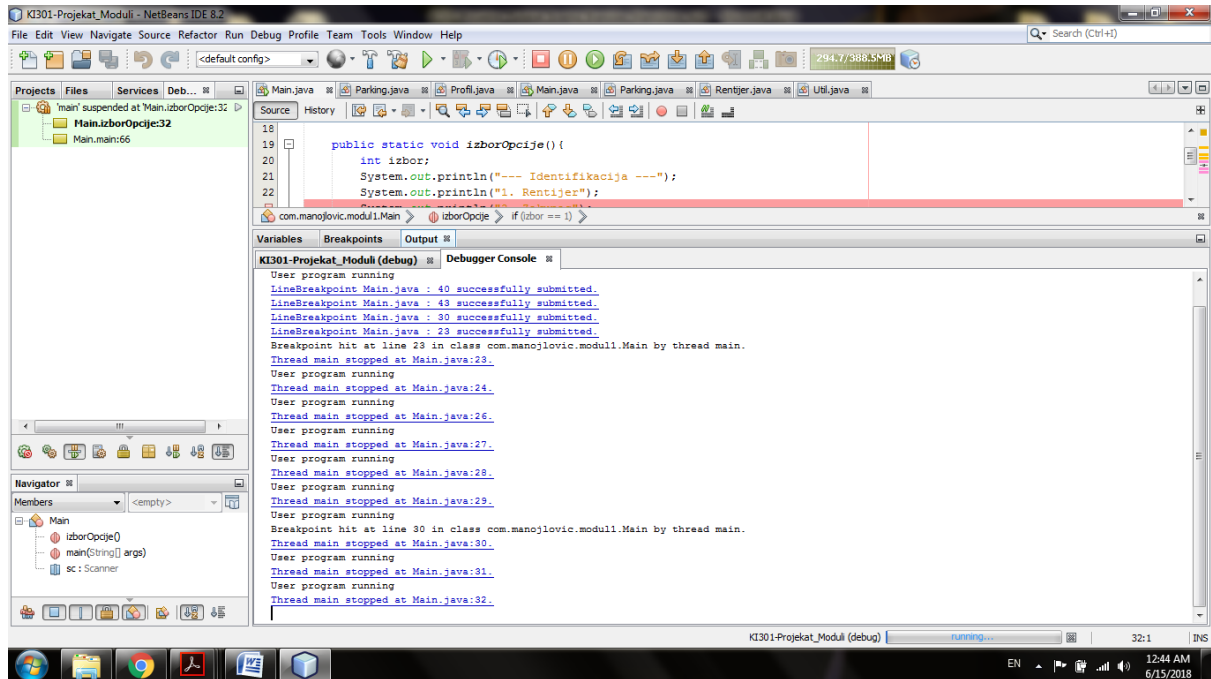
Slika 14. Program staje na Breakpoint-u i omogućava detaljan pregled koda

4. Klikom na dugme „Step Over“ ukoliko želim da nastavim da prolazim kroz kod liniju po liniju, kako bih debugovao program tražeći problem. Prelaz je označen zelenom linijom na slici 15. Ukoliko želim da sa jednog Breakpoint-a odmah prođem kroz celu metodu, mogu da kliknem na dugme „Step Out“.



Slika 15. Prelazak na sledecu liniju koda klikom na dugme “Step Over”

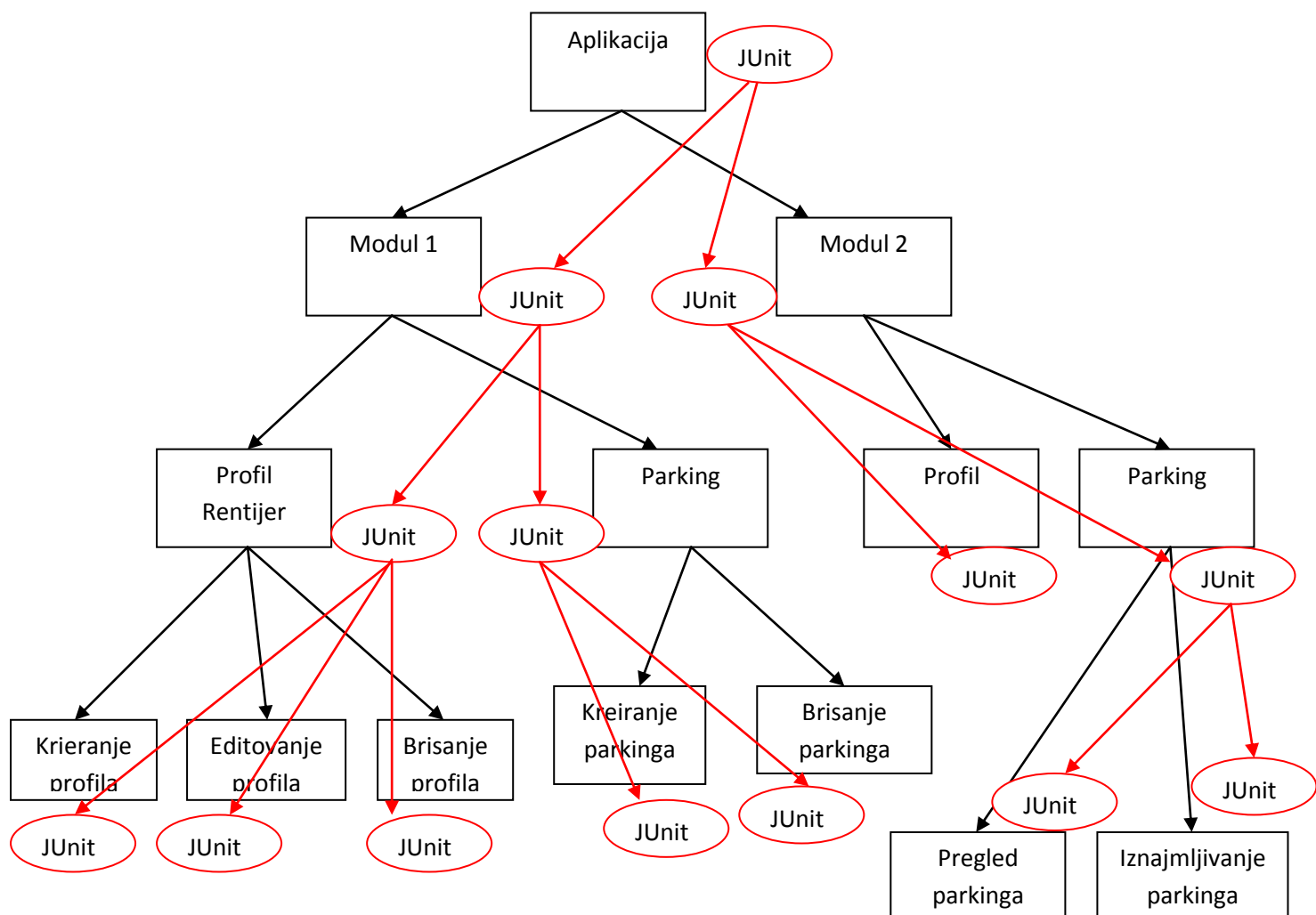
5. Svaki korak koji učinimo prilikom debugovanja programskog fajla možemo pratiti na konzoli za debugovanje (Debugger Console). Prolazimo kroz kod Breakpoint po Breakpoint i polako ispravljamo bug po bug dok ne uklonimo sve greške u kodu na koje naiđemo.



Slika 16. Ispis na Debugre Console

Korišćenje debagera u NetBeans random okruženju sam debugovao program, prolazeci kroz programski kod liniju po liniju gde sam polako pregledao šta se dešava i da tako uočio gde se javljala greška ili problem.

Test slučajevi Integracionog Testiranja Modula I i Modula II



Slika 17. Dijagram Integracionog testiranja Top-down pristupom

Primenjen pristup je Integraciono testiranje **odozgo-nadole** (top-down). Testiranje treba izvršiti primenom JUnit framework-a u NetBeans random okruženju, budući da je kompletna aplikacija pisana u programskom jeziku Java.

MSc Branislav Manojlović
Novi Sad 18.05.2018.