

*PREDMET: SE201 – Uvod u softversko inženjerstvo*

# Projektni zadatak na temu

Privatna Bolnica

Profesor: Ime i prezime:

##### Prof. Dragan Domazet Nikola Ranđelović

Asistent: Broj indeksa:

##### Bogdan Janković 5033

1. Uvod

Cilj dokumenta je da detaljno predstavi način rada privatne bolnice. Ovaj projektni zadatak baviće se definisanjem osnovnih funkcionalnih i nefunkcionalnih zahteva koje softver mora da zadovolji kako bi se smatrao uspešnim. U dokumentu će biti detaljno opisan rad sistema za rad privatne bolnice. Za bolje razumevanje problema sa kojim se susrećemo kako bi projekat bio realizovan biće objašnjeni putem korišćenja dijagrama (Klasni, UseCase, sekvencijalni) koji bi trebalo detaljnije da objasne način i zamisao funkcionisanja sistema. Za potrebe izrade dijagrama koristiće se alata **PowerDesigner.** U projektnom zadatku biće predstavljene i mogućnosti svih korisnika sistema. Korisnici ovog sistema su korisnici, doktori i radnici.

Projekat je tako zamišljen kako bi olakšao zakazivanje pregleda i vodjenje evidencije u magacinu.

# Opis i svrha dokumenta

Svrha dokumenta je da jasno opiše moguće funkcionalnosti i mogućnosti softvera za rad bolnice. Dokument se sastoji od samog uvoda gde se daje kratak opis cilja za kreiranje ovog sistema. Nakon toga, dokument će se sastojati od opisa sistema sa identifikovanim klasama korisnika i njihovim ovlašćenjima, detalje zahteva u kojima su opisani funkcionalni i nefunkcionalni zahtevi. Dokument će se sastojati i od specifikacije dizajna u okviru kog se nalaze dijagrami slučajeva korišćenja, klasni dijagram i svih mogućih sekvencijalnih dijagrama koji su vezani za rad sistema.

Takođe, jako bitna stavka je razvoj test klasa u java programskom jeziku u kojem ćemo simulirati nekoliko funkcija od kojih će se sistem sastojati.

3. Opis sistema i metodologija rada

Sistem je osmišljne tako da korisnik u svakom trenutku može da zakaže pregled kod doktora,naravno pri zakazivanju mora da sačeka odobrenje samog pregleda,ali ukoliko korisnik redovno pregledava nalog,moći će da vidi da li je pregled odobren ili nije.

S druge strane, postojaće sistem koji će biti fokusiran na rad magacina. Radnici će moći da upisuju i ispisuju robu sa sistema, sve sa ciljem što boljeg evidentiranja transakcije robe kroz magacin.

3.1. Funkcije softvera

Funkcije sistema su:

* Kreiranje korisnia
* Zakazivanje pregleda
* Pregled zakazanog pregleda
* Odobravanje pregleda
* Odbijanje pregleda
* Administracija proizvoda u magacinu
* Pregled transakcija u magacinu

## Korisnici sistema

Postoje dve vrste korisnika sistema za iznajmljivanje automobila:

1. Korisnik
2. Doktor
3. Radnik

3.2.1. Opis sistema za korisnika

Sistem je zamišljen da funkcioniše uz pomoć softvera koji će korisnik upotrebljavati. Korisnik će imati mogućnost pregleda dostupnih doktora, zakazivanja pregleda. Ukoliko odabere opciju za pregled dostupnih doktora prikazaće mu se sve doktore koji su dostupni, a ako nema dostupnih doktora korisnik će biti obavešten.

Opcijom za zakazivanje pregleda korisnik će unosom određenih podataka zakazati pregled, naravno svi podaci moraju biti validni kako bi uspešno zakazao pregled. Opcijom pregled zakazanih pregled će imati uvid u preglede koje je zakazao, a ako ih nema biće obavešten da nema zakazanih pregleda.

3.2.2. Opis sistema za doktora

Doktor će imati mogućnost pregleda svih zahteva za pregled nakon čega će moći da odabere da li želi da odobri pregled ili ne.

Ukoliko se doktor smatra da je obrazloženje korisnika **validan**. Doktor ce uneti sve podatke o predstojećem pregledu koje će naknadno korisnik dobiti i biće uspešno upoznat sa svim detaljima predstojećeg pregleda.

Ukoliko doktor smatra da obrazloženje korisnije **nije validno.** Doktor će uneti obrazloženje u vidu teksta gde će obrazložiti svoju odluku korisniku.

3.2.3. Opis sistema za radnika

Radnik će imati mogućnost upisivanja robe u sistem. Ima mogućnost da upiše sve podatke o proizvodu. Ako su svi podaci validni, radnik je uspešno upisao robu.

Takođe, radnik će imati mogućnost ispisivanje robe, gde će uneti razlog uzimanja robe iz magacina i količinu. Količina će se posebno proveravati.

Radnik će na kraju imati i mogućnost pregleda svih transakcija robe gde će imati mogućnost pretraživanja po datumima gde će sistem vratiti taj dan sa svim transakcijama u toku tog dana.

## Metodologija izrade

Metod korišćen i implementiran je hibridni model razvoja. Delom je korišćen model vodopada zato što se većina funkcija neće menjati i ostaće slična prvoj verziji vezanoj za zakazivanje preko sistema, jer se tim potrudio da taj deo bude što prostiji i logičniji.

# Specifikacija zahteva

Specifični softverski zahtevi predstavljaju način na koji bi bilo potrebno razvijati neki sistem. Dobro definisanje specifičnih zahteva dovodi do ispravnog rešenje. U ovaj tip softverskih zahteva spadaju funkcionalni i nefunkcionalni zahtevi.

4.1.1. Korisnički interfejs

Izgled korisničkog interfejsa će zavisiti od privilegija korisnika koji se prijavljuje na sistem. Intefejs je jednostavan, intuitivan i lak za upravljanje. Korisnički intefejs poseduje:

* + - Organizaciju ekrana: KI ima osnovnu formu za logovanje. Ukoliko je logovanje uspešno urađeno, KI intefejs se sastoji od jednog prozora na kojem su vidljive sve funkcije koje korisnik sa određenim privilegijama može da koristi.
    - Povratne poruke: Za svaku nastalu grešku ili neku uspešnu radnju korisnika, softver će izbaciti poruku u vidu novog prozora i time obavestiti korisnika.

4.1.2. Hardverski interfejs

Izlazno-ulazni uređaji moraju biti podržani od strane sistema. Na primer izlazni uređaj može biti monitor ili displej. Ulazni uređaj može biti miš, tastatura i displej telefona za samu aplikaciju sistema.

4.1.3. Softverski interfejs

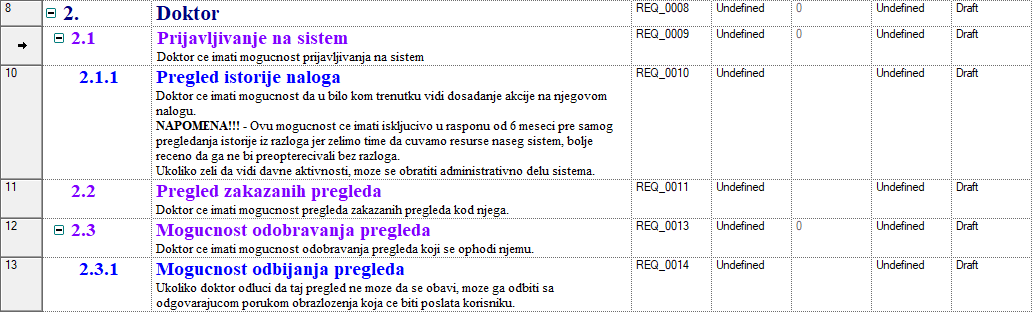
Aplikacija sistema će pristupati bazi podataka za izvršavanje različitih upita.

## Funkcionalni zahtevi

U delu funkcionalnih zahteva biće prikazani svi funkcionalni zahtevi koje sistem za iznajmljivanje aautomobila treba da zadovolji. Funkcionalni zahtevi će biti prikazani u tri sektora, funkcionalne zahteve koje mora da zadovolji doktor, funkcionalne zahteve koje mora da zadovolji korisnik i one koje mora da zadovolji radnik. Za izradu i prikaz funkcionalnih zahteva se koristi Requirements model koji zadovoljava sve standarde za pisanje istih generisan u PowerDesigner alatu.

4.2.1. Funkcionalni zahtevi doktora

U ovom poglavnju prikazani su funkcionalni zahtevi koje doktor treba da zadovolji. Doktor je zadužan je za greške koje se pojave. Ima uvid u sve zakazane preglede kod njega. Svi zahtevi koje administrator pruža se nalaze na slici 4.2.1.1.



Slika 4.2.1.1 – Funkcionalni zahtevi doktora

#### Pregled istorije naloga

#### Doktor će imati mogućnost da u bilo kom trenutku vidi dosadašnje akcije na njegovom nalogu.

#### **NAPOMENA!!!** - Ovu mogućnost će imati isključivo u rasponu od 6 meseci pre samog pregledanja istorije iz razloga jer želimo time da čuvamo resurse naseg sistema, bolje rečeno da ga ne bi preopterećivali bez razloga.

#### Ukoliko želi da vidi davne aktivnosti, može se obratiti administrativnom delu sistema.

#### 4.2.1.2 Pregled zakazanih pregleda

#### Doktor će imati mogućnost pregleda zakazanih pregleda kod njega.

#### 4.2.1.3 Odobravanje pregleda

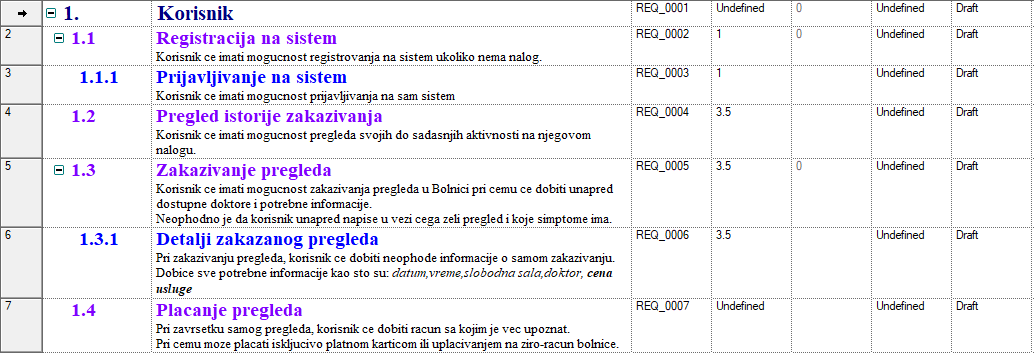
#### Doktor će imati mogucnost odobravanja pregleda koji se ophodi njemu.

#### 4.2.1.4 Odbijanje pregleda

#### Ukoliko doktor odluči da taj pregled ne može da se obavi, može ga odbiti sa

#### odgovarajućom porukom obrazloženja koja će biti poslata korisniku.

4.2.2. Funkcionalni zahtevi korisnika



Slika 4.2.2.1 – Funkcionalni zahtevi korisnika

#### Ragistracija na sistem

Korisnik će imati mogucnost registrovanja na sistem ukoliko nema nalog.

#### Pregled istorije zakazivanja

Korisnik će imati mogucnost pregleda svojih do sadašnjih aktivnosti na njegovom nalogu.

#### 4.2.2.3 Zakazivanje pregleda

Korisnik će imati mogucnost zakazivanja pregleda u Bolnici pri čemu će dobiti unapred dostupne doktore i potrebne informacije.

Neophodno je da korisnik unapred napiše u vezi čega želi pregled i koje simptome ima.

#### Detalji zakazanog pregleda

Pri zakazivanju pregleda, korisnik će dobiti neophode informacije o samom zakazivanju.

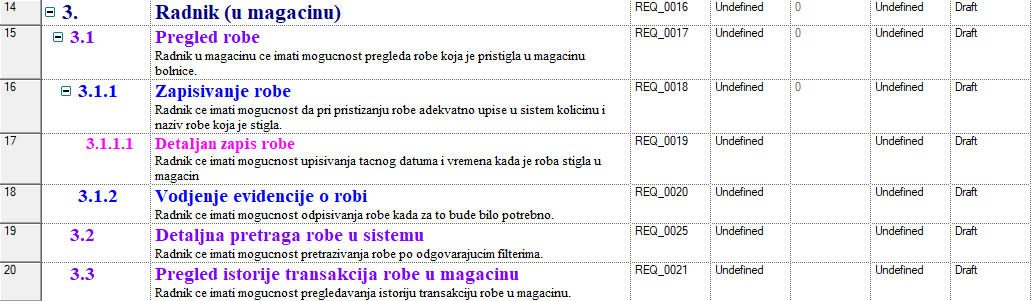
Dobiće sve potrebne informacije kao što su: datum,vreme,slobodna sala, cena usluge.

#### Plaćanje pregleda

Pri završetku samog pregleda, korisnik će dobiti račun sa kojim je već upoznat.

Pri čemu moze plaćati isključivo platnom karticom ili uplaćivanjem na žiro-račun bolnice.

4.2.3. Funkcionalni zahtevi radnika



Slika 4.2.3.1 – Funkcionalni zahtevi radnika

#### Pregled robe

#### Radnik u magacinu će imati mogućnost pregleda robe koja je pristigla u magacinu bolnice.

#### Zapisivanje robe

#### Radnik će imati mogućnost da pri pristizanju robe adekvatno upiše u sistem količinu i naziv robe koja je stigla.

#### Detaljan zapis robe

#### Radnik će imati mogućnost upisivanja tacnog datuma i vremena kada je roba stigla u magacin

#### Odpisivanje robe

#### Radnik će imati mogućnost odpisivanja robe kada za to bude bilo potrebno.

#### Detaljna pretraga robe u sistemu

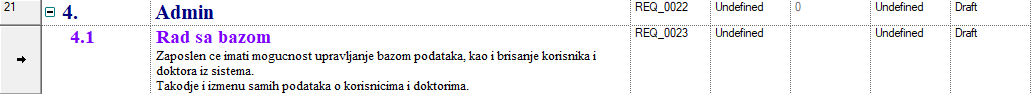
#### Radnik će imati mogućnost pretrazivanja robe po odgovarajucim filterima.

#### Pregled istorije transakcije robe u magacinu

#### Radnik će imati mogućnost pregledavanja istoriju transakcije u magacinu

.

4.2.4. Funkcionalni zahtevi admina



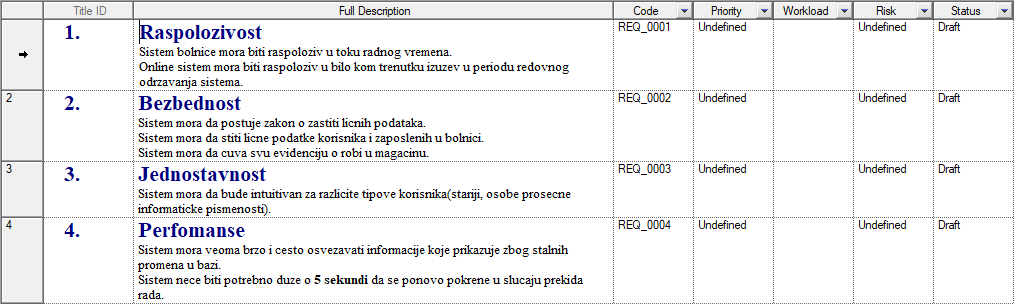
#### Rad sa bazom

#### Admin će imati mogućnost upravljanje bazom podataka, kao i brisanje korisnika i doktora iz sistema.

#### Takođe i izmenu samih podataka o korisnicima i doktorima.

4.2.5. Nefunkcionalni zahtevi

Nefunkcionalni zahtevi su zahtevi koji su vezani za ceo sistem, to nisu neke akcije koje sistem preduzima. Pod nefunkcionalnim zahtevima spadaju raspolozivost, praktičnost sistema, bezbednost sistema i perfromanse.



Slika 4.2.5.1 – Nefunkcionalni zahtevi

#### Raspoloživost

Sistem bolnice mora biti raspoloziv u toku radnog vremena.

Online sistem mora biti raspoloziv u bilo kom trenutku izuzev u periodu redovnog odrzavanja sistema.

#### Bezbednost

Sistem mora da postuje zakon o zastiti licnih podataka.

Sistem mora da stiti licne podatke korisnika i zaposlenih u bolnici.

Sistem mora da cuva svu evidenciju o robi u magacinu.

#### Jednostavnost

Sistem mora da bude intuitivan za razlicite tipove korisnika(stariji, osobe prosecne informaticke pismenosti).

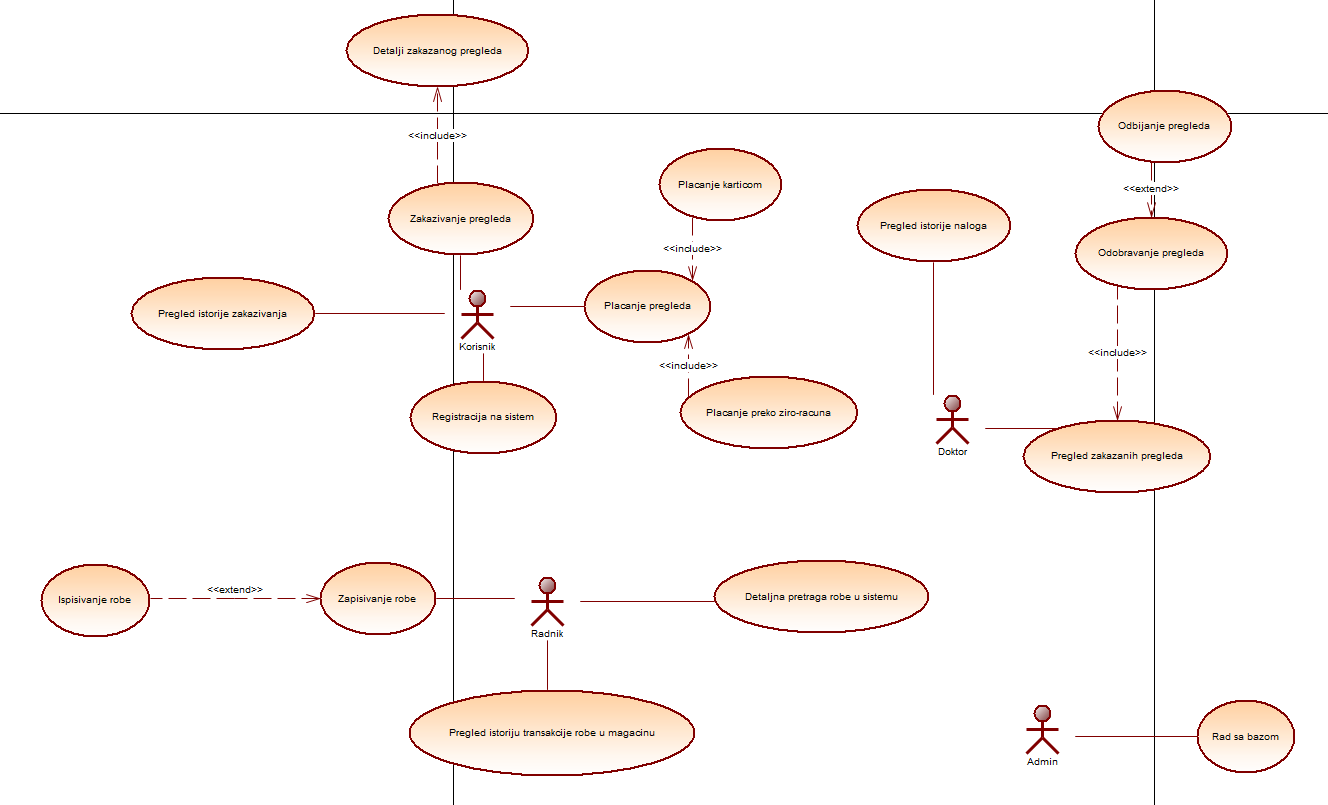
#### Performanse

Sistem mora veoma brzo i cesto osvezavati informacije koje prikazuje zbog stalnih promena u bazi.

Sistem nece biti potrebno duze o **5 sekundi** da se ponovo pokrene u slucaju prekida rada.

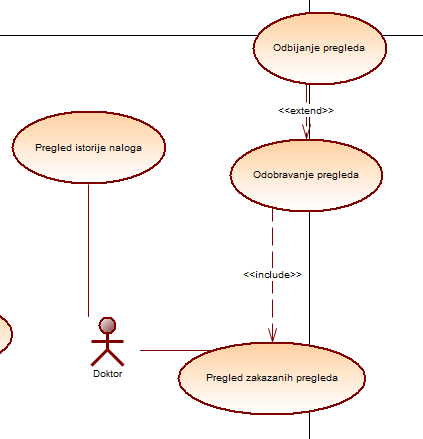
## Slučajevi korišćenja

Slučajevi korišćenja predstavljaju akcije koje akteri mogu da izvrše sa funkcionalnostima softvera. Slučajevi korišćenja su prikazani u UseCase dijagramu koristeći PowerDesigner alat.



Slika 4.3.1 – Slučajevi korišćenja svih aktera sistema

4.3.1. UseCase doktora



Slika 4.3.1.1 – Slučajevi korišćenja doktora

#### Pregled istorije naloga

##### Preduslovi:

##### Doktor ima nalog na sistemu.

##### Osnovni scenario:

##### Prikazana stranica za prikaz svih vozila.

##### [Izuzeci:

##### [EXC-1]

##### NAPOMENA!!! - Ovu mogucnost ce imati iskljucivo u rasponu od 6 meseci pre samog pregledanja istorije iz razloga jer zelimo time da cuvamo resurse naseg sistem, bolje receno da ga ne bi preopterecivali bez razloga.

##### Ukoliko zeli da vidi davne aktivnosti, moze se obratiti administrativno delu sistema

##### Postuslovi:

Doktor je upoznat sa istorijom naloga.

#### Pregled zakazanih pregleda

##### Preduslovi:

##### Doktor bi trebalo da ima zakazane preglede.

##### Osnovni scenario:

##### Doktor pritiskom na dugme "Zahtevi za pregled" dobija prozor sa listom pregleda koje zele da se odviju kod njega.[EXC-1]Zaposleni unosi nove tablice automobila.

##### Doktor pritiskom na odgovarajuci pregled moze otvoriti zasebni pregled.

##### Izuzeci:

##### [EXC-1]

##### Ukoliko doktoru ne izlazi nikakva lista sa zakazanim pregledima, to je iz razloga jer ne postoji ni jedan zakazani pregled u tom trenutku.

##### Postuslovi:

Doktor je upoznat sa zakazanim pregledima.

#### 4.3.1.3 Odobravanje pregleda

##### Preduslovi:

##### Pregled je zakazan kod odgovarajuceg doktora.

##### Osnovni scenario:

##### Doktor klikom na korisnika u listi dobija formular koji treba da popuni.

##### Doktor unosi vreme kada ce se odvijati pregled.

##### Doktor unosi slobodnu salu u kojoj ce se pregled odvijati.

##### Doktor klikom na dugme "Odobri" uspesno salje obavestenje korisniku da je odobren pregled sa svim informacijama o buducem pregledu.

##### Postuslovi:

#### Doktor uspesno salje obavestenje korisniku da je pregled odobren,sa svim potrebnim informacijama.

#### 4.3.1.4 Odbijanje pregleda

##### Preduslovi:

##### Pregled je zakazan kod odgovarajućeg doktora.

##### Osnovni scenario:

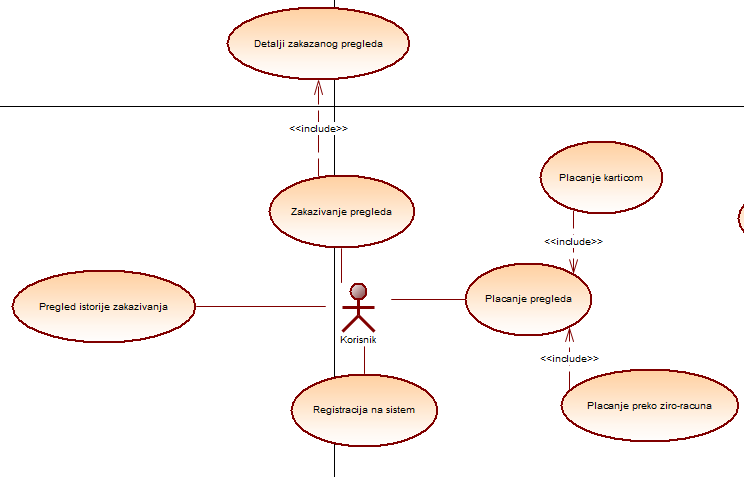
##### Nakon klika na dugme "Odbijanje pregleda", doktor dobija formular u kom mora da obrazlozi iz kog razloga nije moguc pregled.

##### Doktor klikom na dugme "Potvrdi" salje obavestenje sa porukom iz kog razloga se pregled odbija.

##### Postuslovi:

Doktor je uspesno obavestio korisnika iz kog razloga nije moguc pregled.

4.3.2. UseCase korisnika

4.3.2.1 – Slučajevi korišćenja za korisnika

#### Pegistracija na sistem

##### Preduslovi:

##### Korinsik ne sme da ima vec kreirani nalog na sistemu.

##### Osnovni scenario:

##### Korisnik unosi ime

##### Korisnik unosi prezime

##### Korisnik unosi e-mail

##### Korisnik unosi sifru

##### Korisnik prilaze odgovarajuci PDF fajl sa istorijom bolesti

##### Korisnik potvrđuje unos klikom na dugme “Registruj se”. [EXC-1]

##### Izuzeci:

##### [EXC-1] Korisnik ima već postojeći nalog

##### Postuslovi:

1. Sistem je uspešno registrovao korisnika i obaveštava korisnika o

uspešnoj registraciji na sistem.

#### Pregled istorije zakazivanja

##### Preduslovi:

##### Korisnik je uspesno ulogovan na svoj nalog.

##### Osnovni scenario:

##### Korisnik pritiskom na dugme "istorija naloga" dobija listu do sadanjis

##### akcija na njegovom nalogu.[EXC-1]

##### Izuzeci:

##### [EXC-1] Ukoliko korisnik nije imao do sadasnje akcije na nalogu nece imati

##### nikakvu listu na svom prozoru.

##### Postuslovi:

1. Korisnik je upoznat sa istorijom svog naloga.

#### Zakazivanje pregleda

##### Preduslovi:

##### Korisnik je uspesno ulogovan na svoj nalog.

##### Osnovni scenario:

##### Korisnik pritiskom na dugme "zakazi pregled" dobija novi prozor sa odgovarajucim stavkama

##### koje mora da popuni.

##### Korisnik unosi poruku zbog cega zeli pregled.

##### Korisnik unosi zeljenog doktora.[EXC-1]

##### Korisnik unosi metodu placanja pregleda.

##### Korisnik potvdjuje unos klikom na dugme ''Zakazi'.

##### Izuzeci:

##### [EXC-1]

##### Ukoliko doktor nije dostupan,sistem ce poslati odgovarajucu poruku korisniku

##### Postuslovi:

Korisnik je uspesno kreirao pregled

#### Detalji zakazanog pregleda

##### Preduslovi:

##### Korisnik mora da ima kreiran zakazan pregled

##### Osnovni scenario:

##### Korisnik klikom na dugme "detalji zakazanog pregleda" dobija informacije o zakazanom pregledu:

##### - vreme i datum

##### - slobodna sala

##### - cena pregleda

##### Postuslovi:

1. Korisnik je upoznat sa svim detaljima o pregledu.

#### Placanje pregleda

##### Preduslovi:

* + - * 1. Korisnik treba da ima zaduzenja na sistem.

##### Osnovni scenario:

* Korisnik klikom na dugme "Zaduzenja" dobija informacije o zaduzenjima.

- Korisnik moze da izabere opciju placanja:

1. Online placanje.

2. Placanje preko ziro-racuna.

##### Postuslovi:

##### Korisnik je uspesno izabrao metodu placanja.

##### Korisnik je upoznat sa svojim zaduzenjima

#### Placanje karticom

##### Preduslovi:

##### Korisnik je izabrao nacin placanja - ONLINE.

##### Osnovni scenario:

##### Korisnik dobija odgovarajuci prozor sa adekvatnim formama za popunjavanje.

##### Korisnik unosi informacije o kartici.[EXC-1]

##### Korisnik potrvdjuje uslove koriscenja.

##### Korisnik pritiskom na dugme "Plati" potvrdjuje transakciju.

##### Izuzeci:

##### [EXC-1] - Korisnik nije uneo validne podatke o kartici

##### Postuslovi:

1. Korisnik je uspesno izmirio svoja zaduzenja.

#### Placanje preko žiro-računa

##### Preduslovi:

##### Korisnik je odabrao modul placanja - ZIRO-RACUN

##### Osnovni scenario:

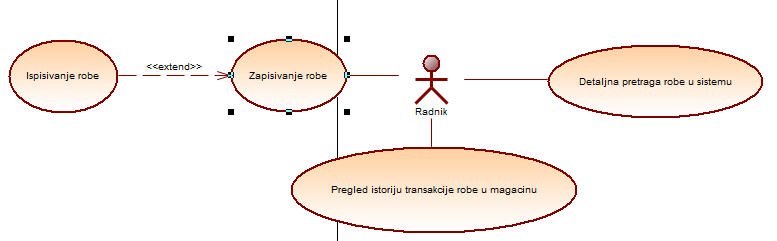
##### Korisnik pritiskom na dugme "Detalji ziro-racuna" dobija informacije

##### o ziro racunu koje treba fizicki da popuni i da plati u odgovarajucoj posti.

##### Postuslovi:

##### Korisnik je dobio JPG žiro-računa

4.3.3. UseCase radnika



Slika 4.3.3.1 – Slučajevi korišćenja radnika

#### Zapisivanje robe

##### Preduslovi:

##### Radnik mora da ima nalog koji kreira iskljucivo Admin

##### Osnovni scenario:

##### Radnik klikom na dugme "Zapisi robu", otvara prozor u kome treba da unese potrebne informacije o samoj robi:

##### - vreme i datum kada je roba stigla u magacin

##### - naziv robe

##### - kolicina

##### - vrsta robe (lek, oprema, odeca itd.)

##### Radnik klikom na dugme "Potvrdi zapis" potvdjuje upisivanje robe u bazu tj. magacin

##### Postuslovi:

1. Radnik je uspesno upisao robu u bazu tj. Magacin

#### Odpisivanje robe

##### Preduslovi:

##### Radnik mora da ima nalog koji kreira iskljucivo Admin

##### Osnovni scenario:

##### Radnik klikom na dugme "Odpisi robu", dobija odgovarajuci prozor u kom mora da upise naziv ili ID robe koja napusta magacin.

##### Klikom na dugme "Potvrdi", sistem otvara prozor u kome mora da popuni sva polja koja se ticu robe:

##### - vreme kada roba bude uzeta iz magacina

##### - razlog uzimanja robe iz magacina

##### - kolicinu koja napusta magacin [EXC-1]

##### Klikom na "Potvrdi odpis", radnik potvrdjuje odpis robe iz magacina

##### Izuzeci:

##### [EXC-1] - Ukoliko radnil zeli vecu kolicinu date robe da uzme iz magacina nego sto ima, sistem ce izbaciti odgovarajucu poruku.

##### 

##### Postuslovi:

#### Radnik je uspesno isporucio robu iz magacina.

#### Pregled istorije transakcija u magacinu

##### Preduslovi:

##### Radnik mora da ima nalog koji kreira iskljucivo Admin

##### Osnovni scenario:

##### Radnik klikom na dugme "Istorija transakcija", dobija prozor u kome se nalazi dugme "Datum transakcije".

##### Radnik klikom na dugme "Datum transakcije" dobija mogucnost unosa datuma koji zeli da pretrazi.

##### Klikom na dugme "Potvrdi" dobija informacije o tom datumu tj. dobija informacije o svim transakcijama u toku tog dana.[EXC-1]

##### Izuzeci:

##### [EXC-1] - ukoliko korisnik unese datum u kome ne postoje transakcije,sistem ce odgovoriti odgovarajucom porukom

##### Postuslovi:

#### Radnik je uspesno upoznat sa svim detaljima transakcije

#### Detaljna pretraga robe u sistemu

##### Preduslovi:

##### Radnik mora da ima nalog koji kreira iskljucivo Admin

##### Osnovni scenario:

##### Radnik pritiskom na dugme "Pretraga robe" dobija novi prozor sa listom svih artikla u magacinu.

##### Radnik uz pomoc filtera moze da pronadje potrebne artikle koji moze da filtrira po:[EXC-1]

##### - nazivu

##### Izuzeci:

##### [EXC-1] - Ukoliko korisnik unese nepostojeci artikal,sistem ce izbaciti odgovarajucu poruku

##### Postuslovi:

#### Radnik je upoznat sa detaljima datog artikla.

#### Radnik dobija informacije kao sto su:

#### - kolicina koja je preostala u magacinu

#### - datum zapisivanja

#### - postojanje artikla u magacinu

4.3.4. UseCase admina

#### 

#### Rad sa bazom

##### Preduslovi:

##### Mora da bude prijavljen kao Admin

##### Osnovni scenario:

##### Admin pritiska dugme "Izmena podataka"

##### Admin dobija listu svih doktora i korisnika.

##### Admin menja informacije o korisniku/doktoru

##### Postuslovi:

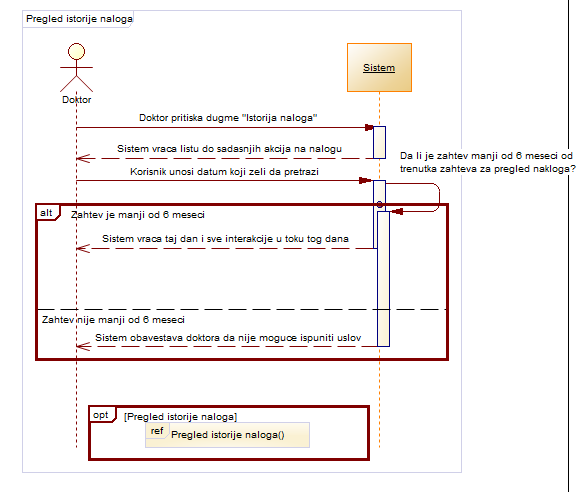
#### Admin je uspesno promenio informacije o korisniku/doktoru

## Apstraktni dijagram sekvenci

Apstraktni dijagram sekvenci ili sistem sekvencijalni dijagrami su interakcioni dijagrami koji detaljno opisuju kako se neke operacije izvršavaju. U ovom projektnom zadataku, sistem sekvencijalni dijagrami će biti vezani za sve operacije koje smo objasnili u dijagramu slučajeva korišćenja sistema. Sistem sekvencijalni dijagrami prikazuju interakciju između objekata tokom izvršenja neke operacije. Hijerarhijski prikazuju sve interakcije koje se dešavaju na vertikalnoj osi u dijagramu.

4.4.1. Sistem sekvencijalni dijagram za doktora

#### Pregled istorije naloga



Doktor pritiskom na dugme “Istorija naloga” dobija informacije o prethodnim obavljenim pregledima

#### Pregled zakazanih pregleda

#### 

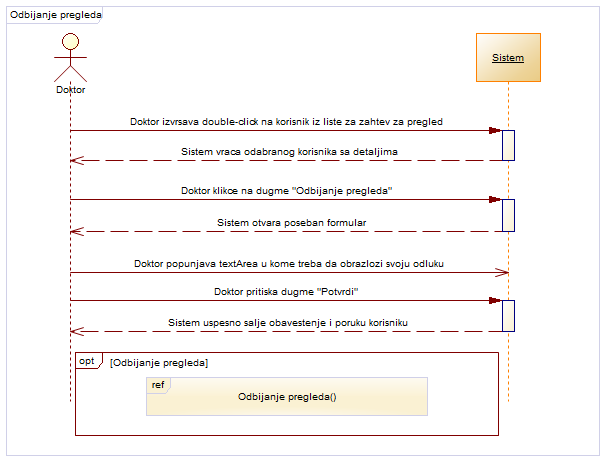
Doktor ima mogucnost pregleda svih zahteva za pregled.

#### Odobravanje pregleda

#### 

Doktor ima mogucnost odobravanje pregleda i unosenje svih potrebnih informacija.

#### Odbijanje pregleda



Takođe, doktor ima mogućnost odbijanja pregleda, s time što mora da obrazloži svoju odluku u vidu teksta.

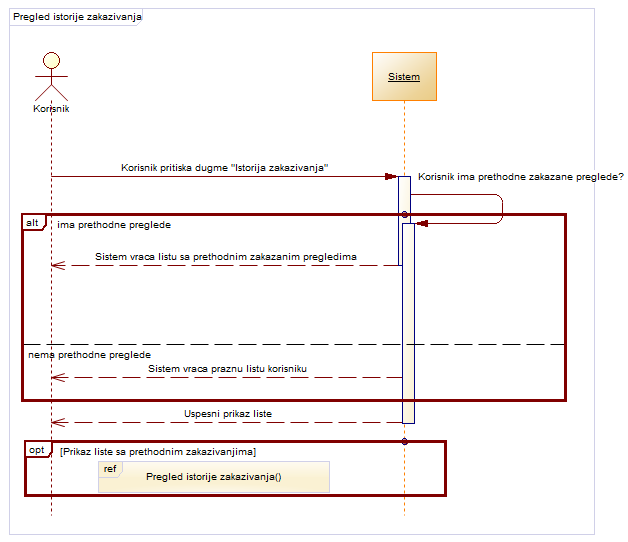
4.4.2. Sistem sekvencijalni dijagrami za korisnika

#### Registracija na sistem

#### 

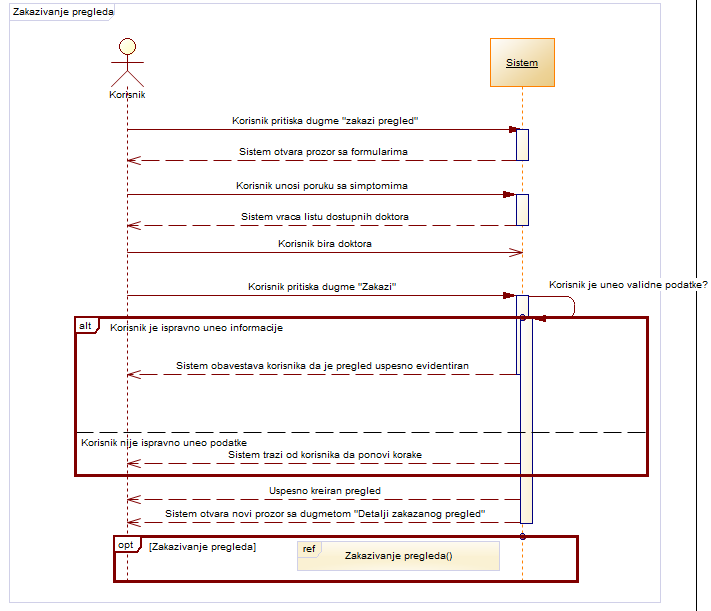
Korisnik ima mogućnost da se registruje na sistem

#### Pregled istorije zakazivanja



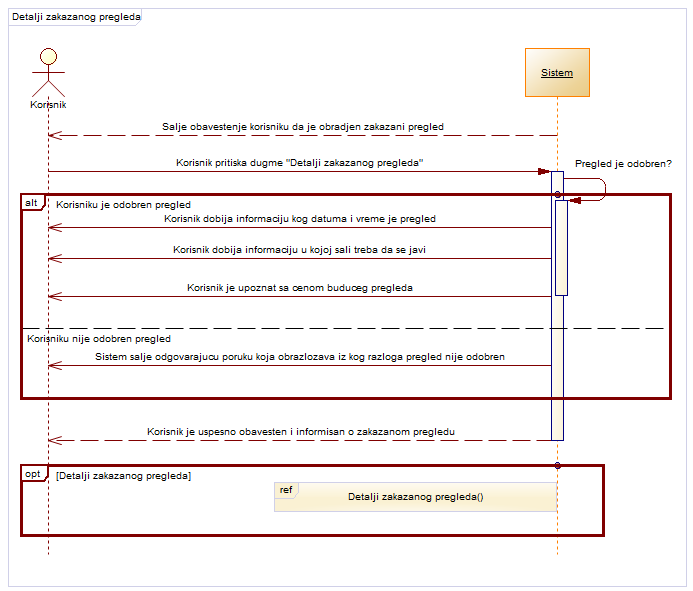
Korisnik ima mogućnost pregledavanja do sadasnjih zakazanih pregled.

#### Zakazivanje pregleda



Korisnik ima mogućnost zakazivanja pregleda.

#### Detalji zakazanog pregleda



Kada korisnik zakaze pregled, otvara mu se nova strana, zvana Detalji zakazanog pregleda, gde korisnik dobija informaciju o predstojecem pregledu.

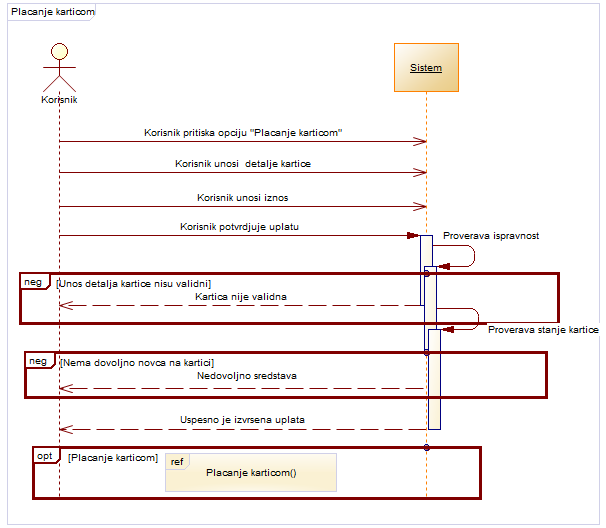
.

#### Plaćanje pregleda

#### 

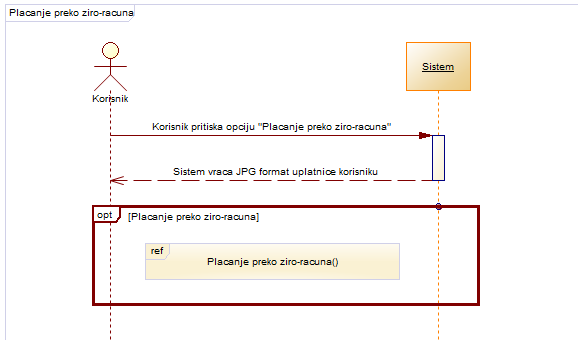
Korisnik ima mogucnost pregleda svih zaduženja prema bolnici.

#### Plaćanje karticom



Korisnik ima mogućnost plaćanje karticom u sistemu.

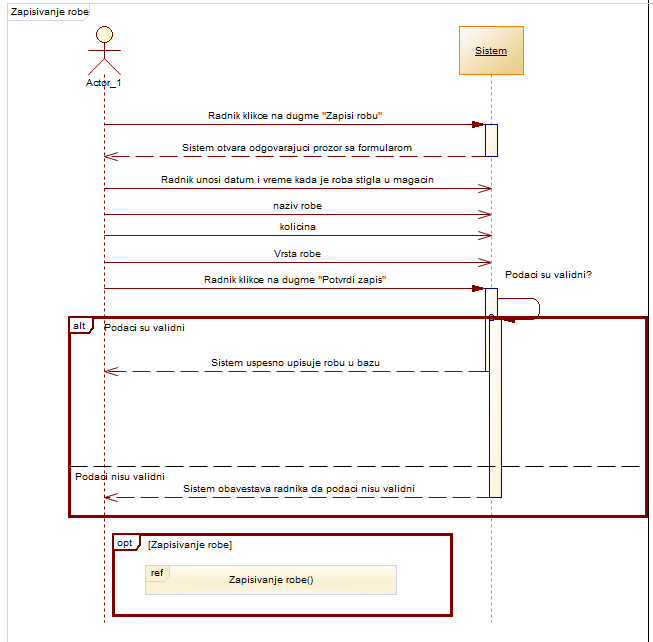
#### Plaćanje preko žiro-računa



Korisnik takođe ima mogućnost plaćanje preko žiro računa, u ovom delu sistem samo vraća JPG žiro-računa sa svim potrebnim informacijama.

4.4.3. Sistem sekvencijalni dijagrami za radnika

#### Zapisivanje robe



Radnik ima mogućnost upisivanje robe u sistem.Radnik je u obavezi da sve podatke o proizvodu unese validno.

#### Odpisivanje robe

#### 

#### Radnik ima opciju odpisivanja robe, tj. kada se uzima roba iz magacina radnik je u obavezi da upise razlog,količinu i naravno naziv leka koji je uzet iz magacina.

#### Detaljna pretraga robe u sistemu

#### 

#### 

#### Radnik će imati mogućnost pretrage robe u sistemu, sistem će vratiti taj proizvod sa svim potrebnim

#### informacijama o njemu

#### Pregled istorije transakcije robe u magacinu

#### 

#### Radnik će imati mogućnost pregledavanja transakcija robe u magacinu. Radnik će imati mogućnost unosa datuma koji želi da proveri, sistem će vratiti sve transakcije vezane za taj datum.

4.4.4. Sistem sekvencijalni dijagrami za admina

#### Rad sa bazom

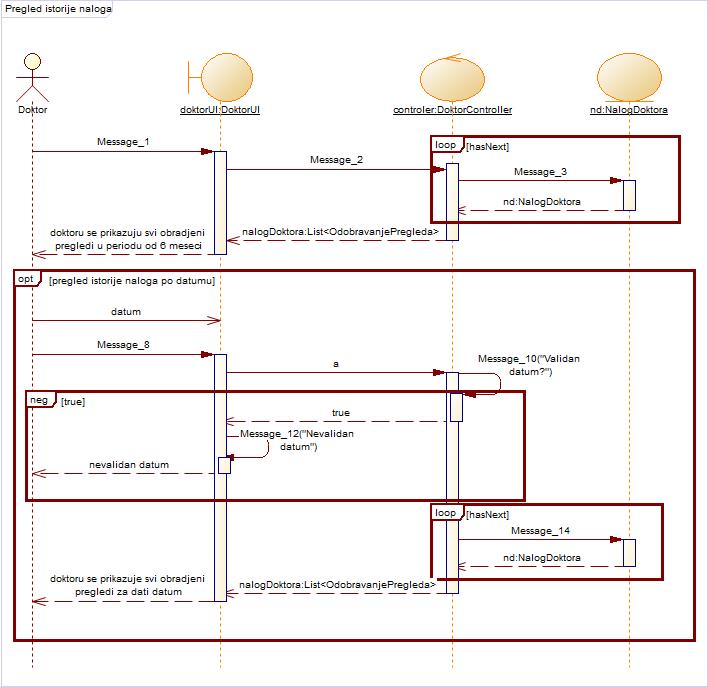
#### 

#### Admin će imati mogućnost rada sa bazom podataka

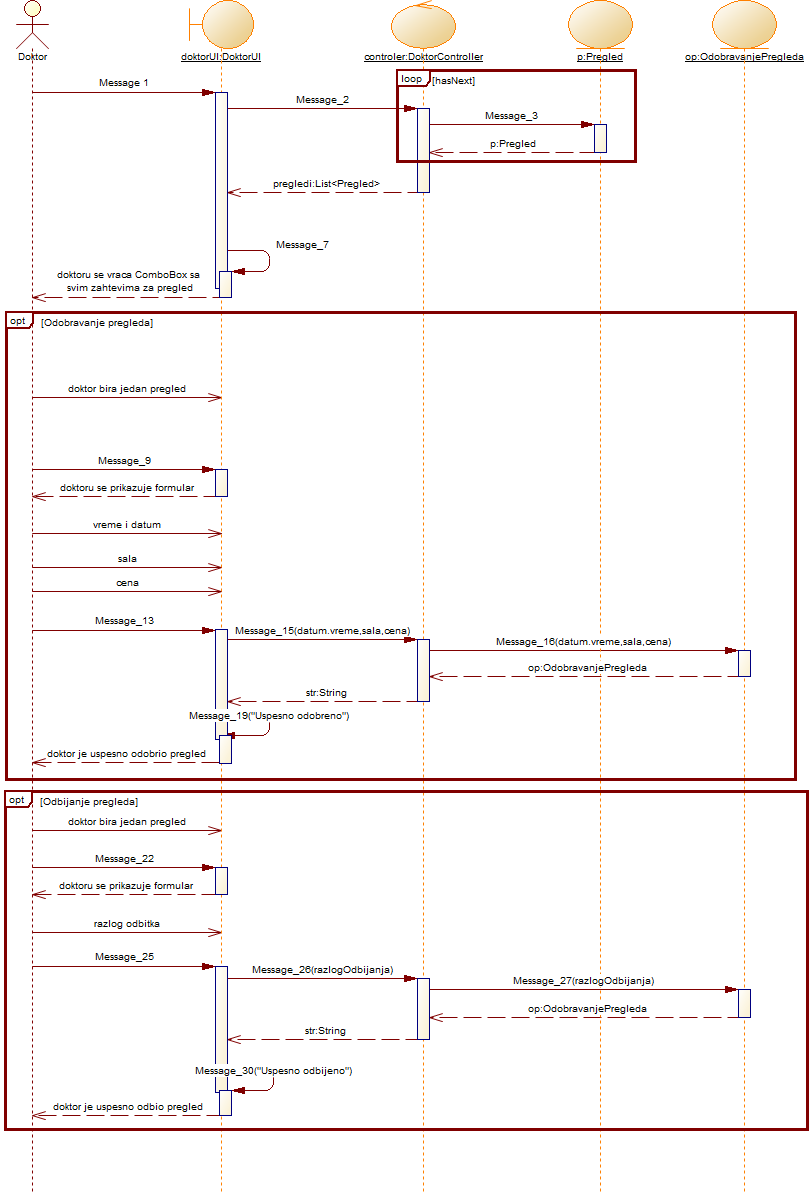
## Sekvencijalni dijagram(detaljni)

U narednom delu projektno zadatka biće predstavljeni detaljni sekvencijalni dijagrami na kojima se prikazuju sve klase koje učestvuju u određenom procesu.

4.5.1. Detaljni sekvencijalni dijagram doktora

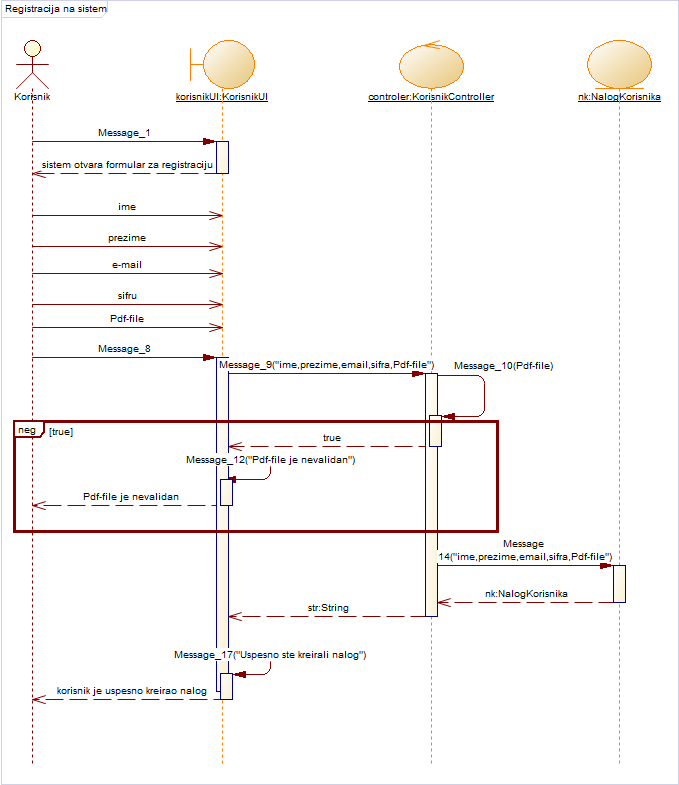


Slika 4.5.1.1 – Detaljni sekvencialni dijagram doktora

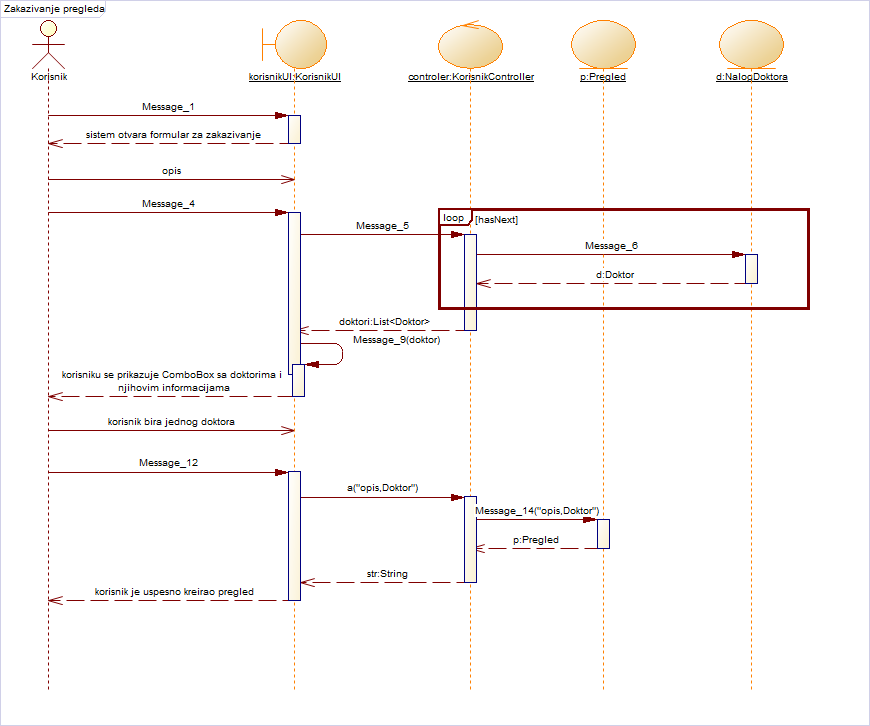


Slika 4.5.1.2 – Detaljni sekvencialni dijagram doktora

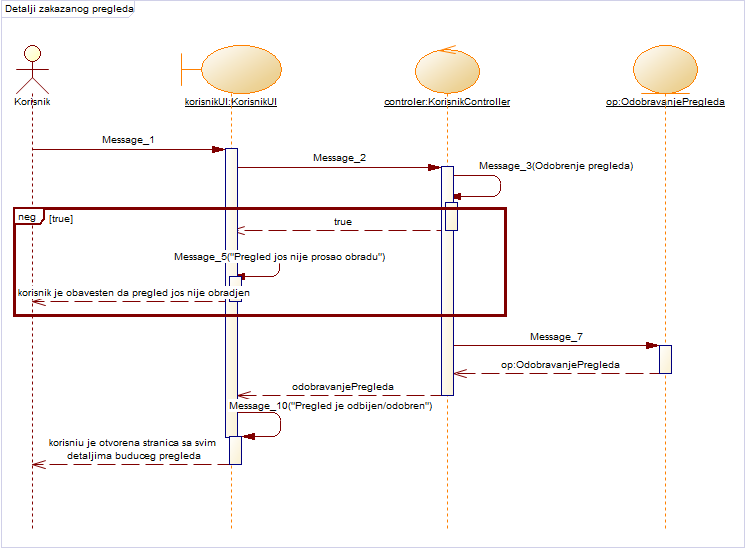
4.5.2. Detaljni sekvencijalni dijagram korisnika



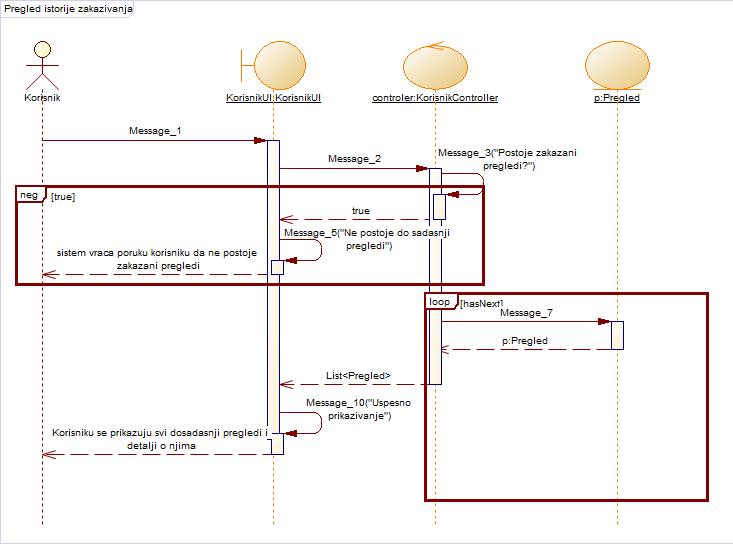
Slika 4.5.2.1 – Detaljni sekvencialni dijagram korisnika



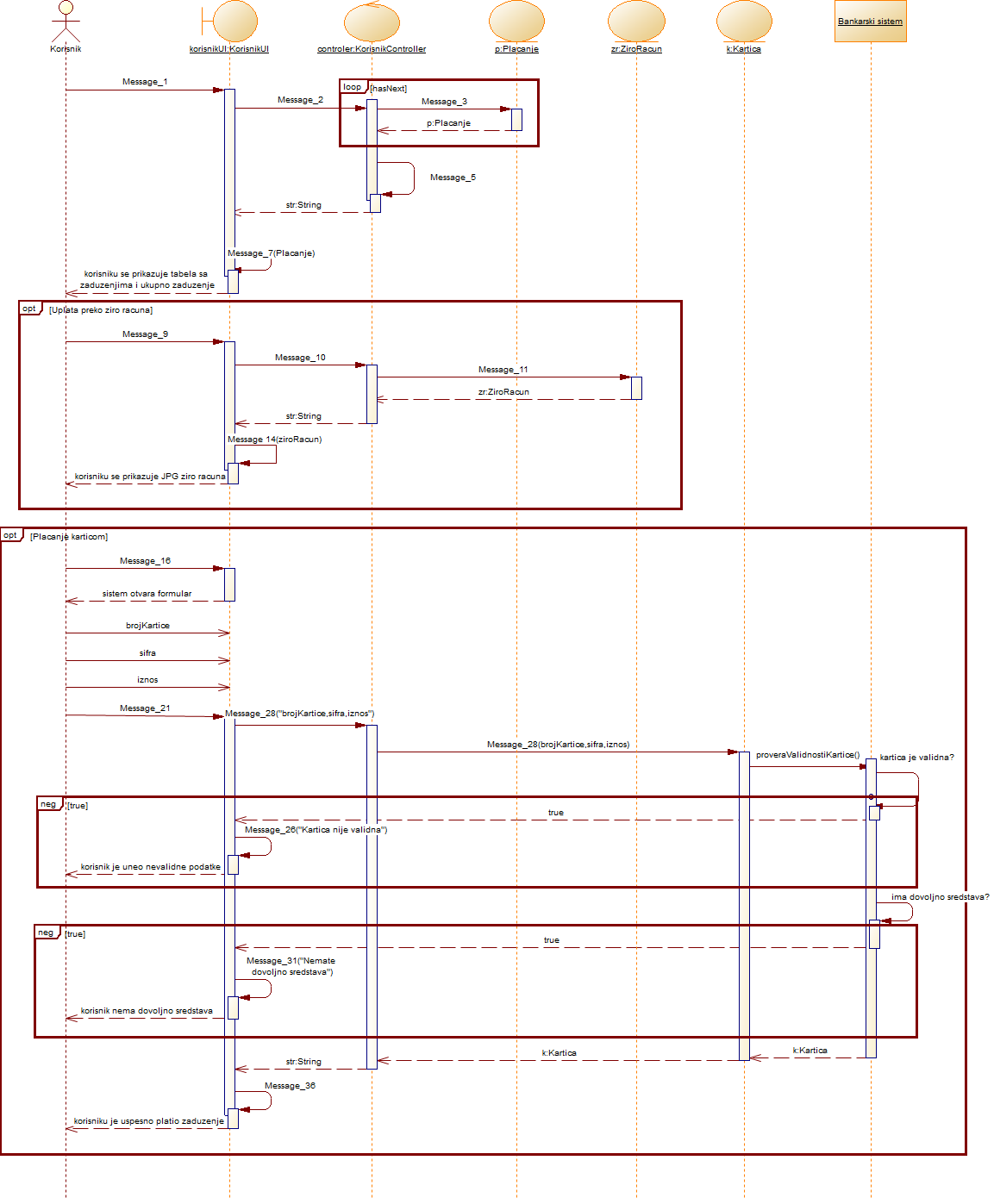
Slika 4.5.2.2 – Detaljni sekvencialni dijagram korisnika



Slika 4.5.2.3 – Detaljni sekvencialni dijagram korisnika

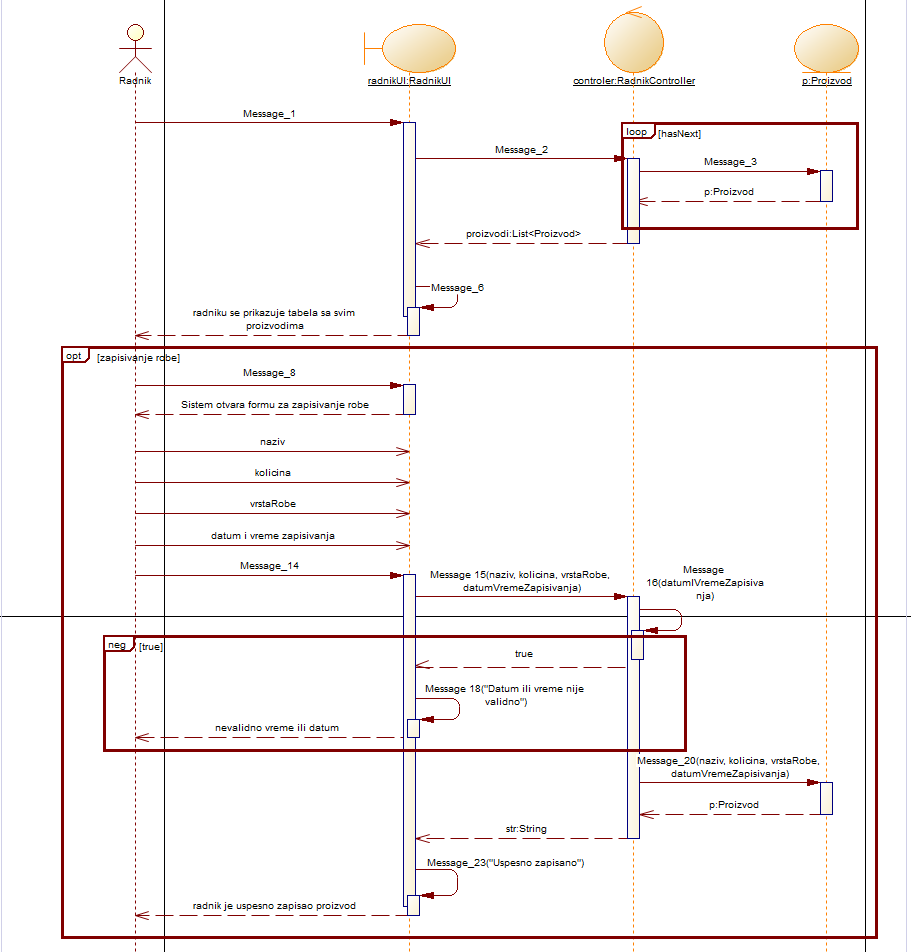


Slika 4.5.2.4 – Detaljni sekvencialni dijagram korisnika

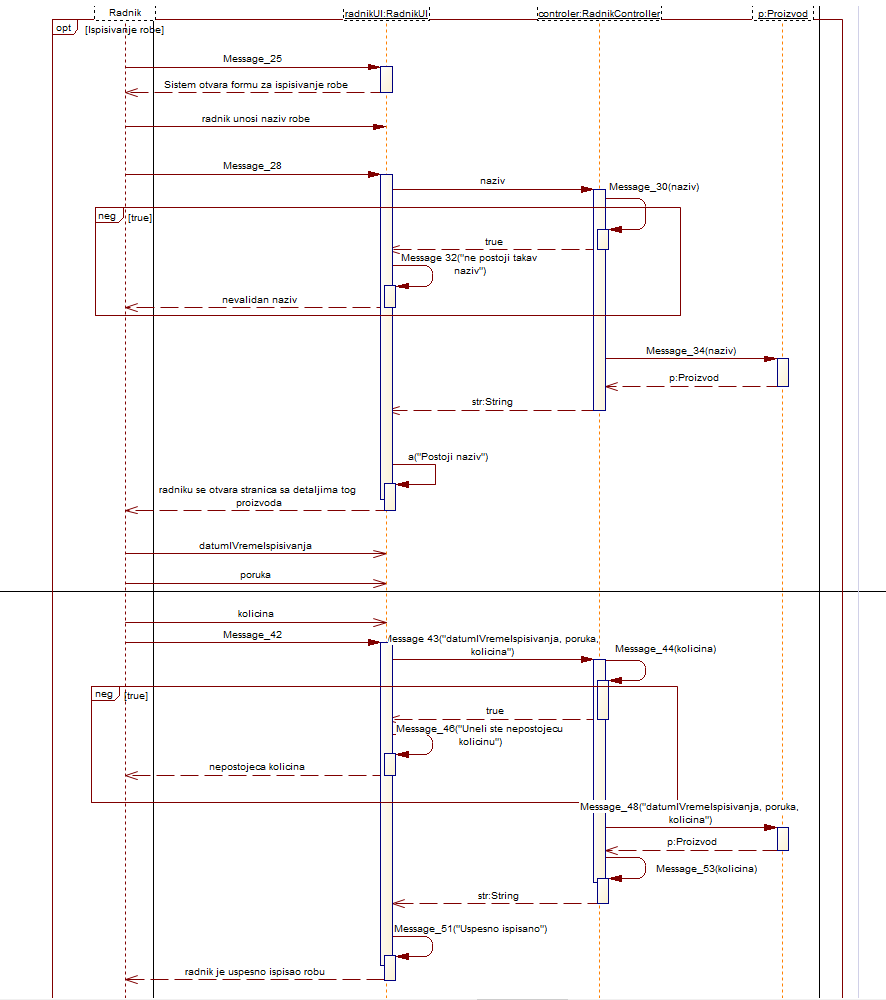


Slika 4.5.2.5 – Detaljni sekvencialni dijagram korisnika

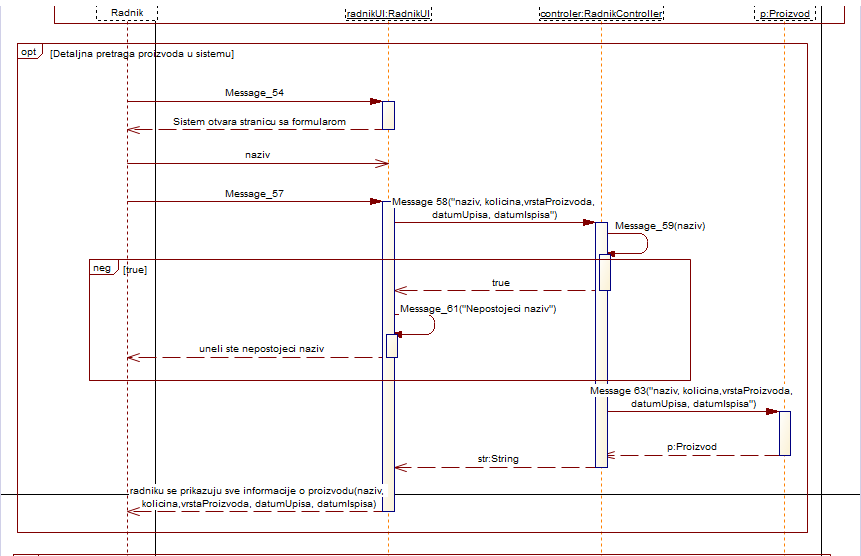
4.5.3. Detaljni sekvencijalni dijagram radnika



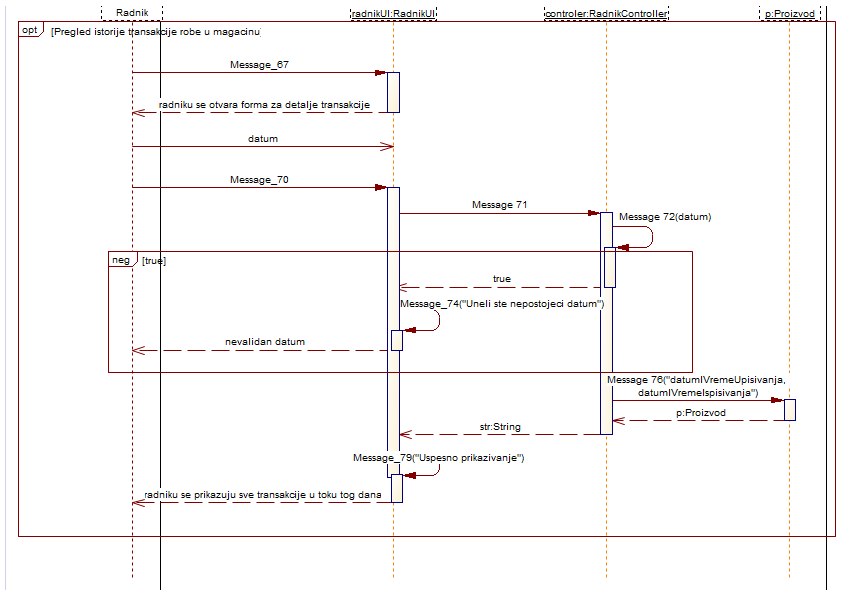
Slika 4.5.2.1 – Detaljni sekvencialni dijagram radnika (1.deo)



Slika 4.5.2.2 – Detaljni sekvencialni dijagram radnika (2.deo)

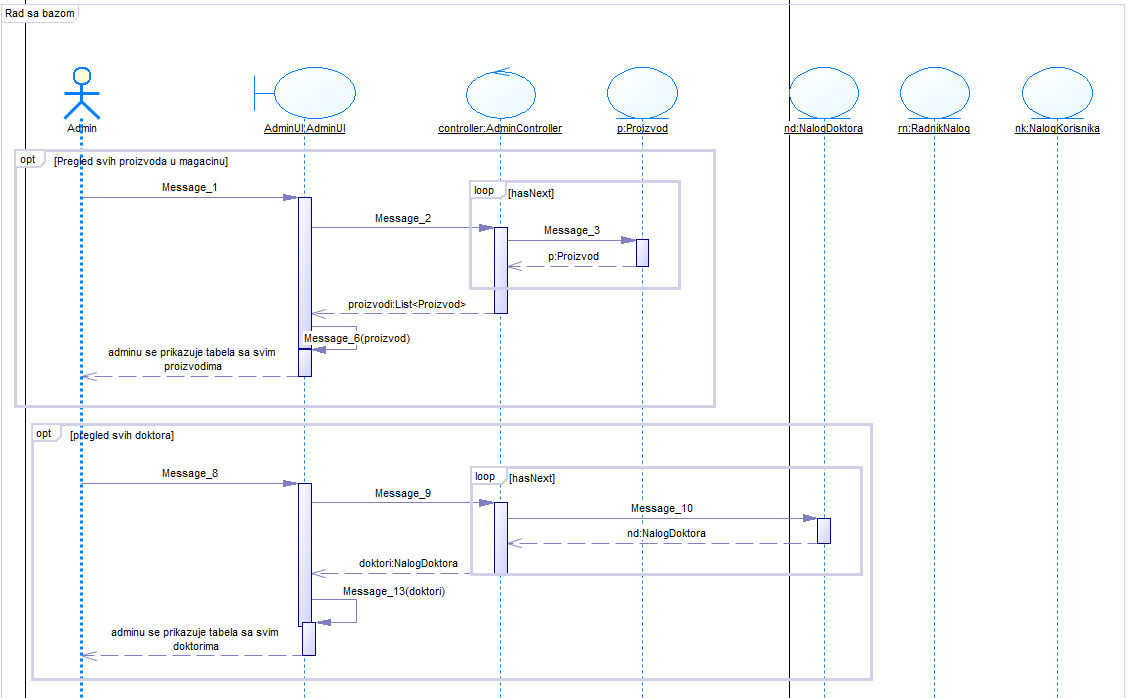


Slika 4.5.2.3 – Detaljni sekvencialni dijagram radnika (3.deo)

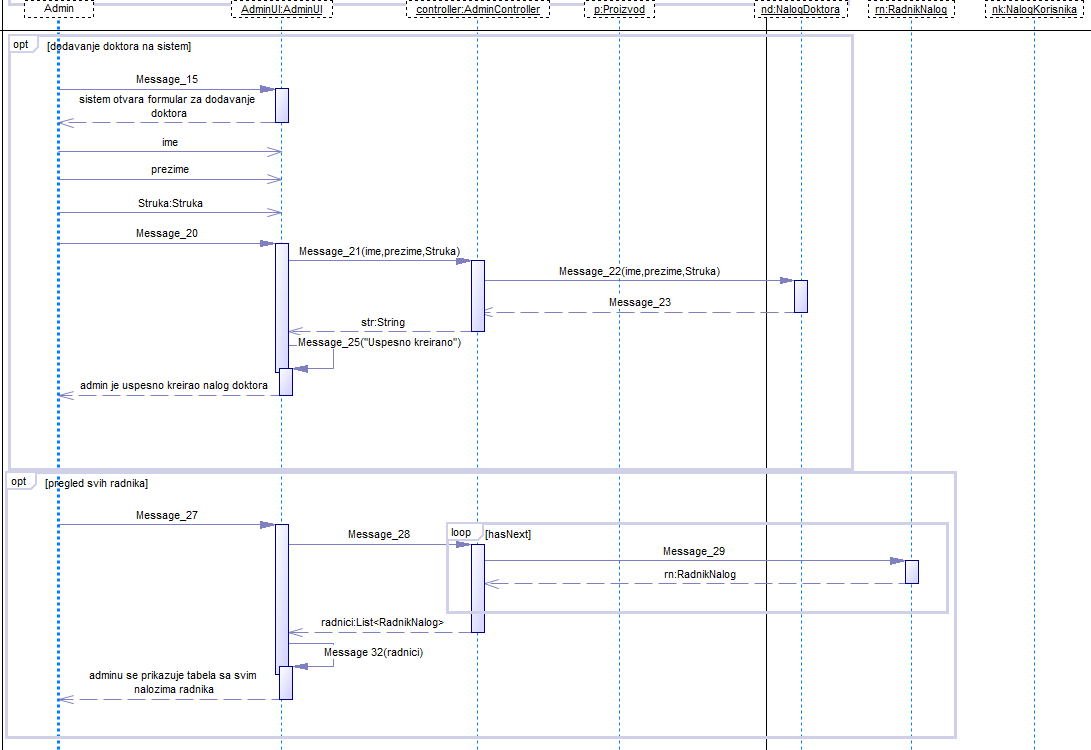


Slika 4.5.2.4 – Detaljni sekvencialni dijagram radnika (4.deo)

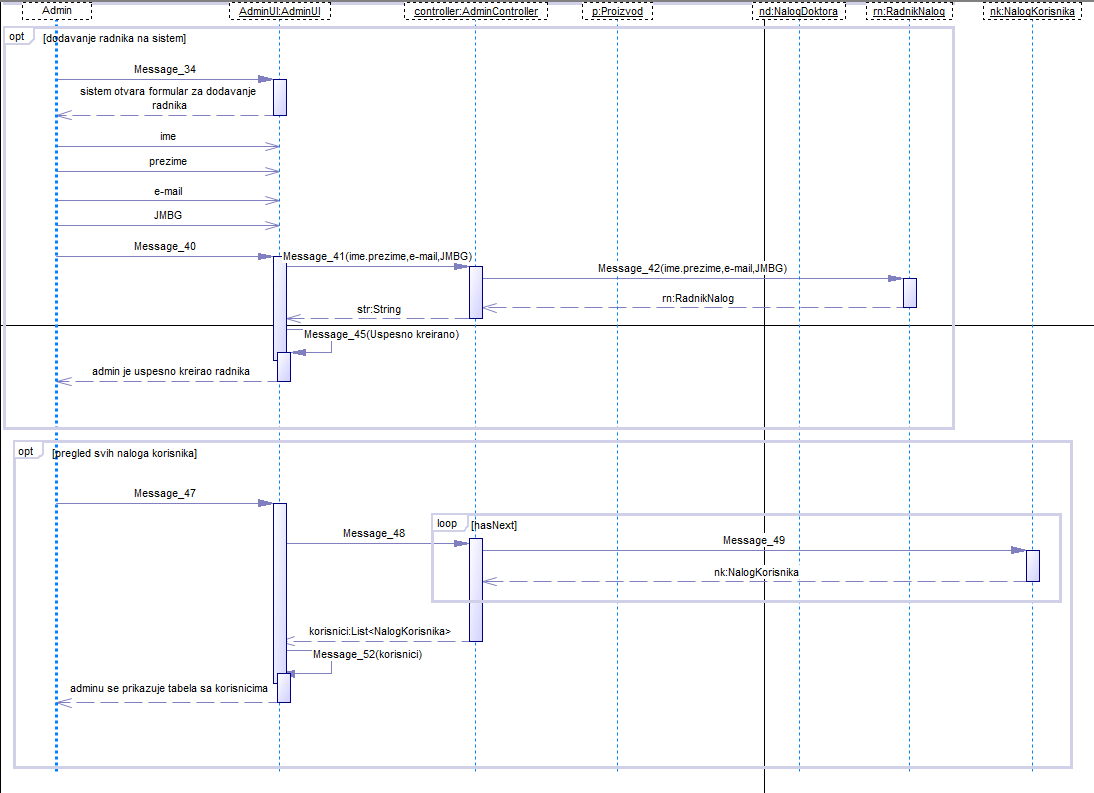
4.5.4. Detaljni sekvencijalni dijagram admina



Slika 4.5.4.1 – Detaljni sekvencialni dijagram admina (1.deo)



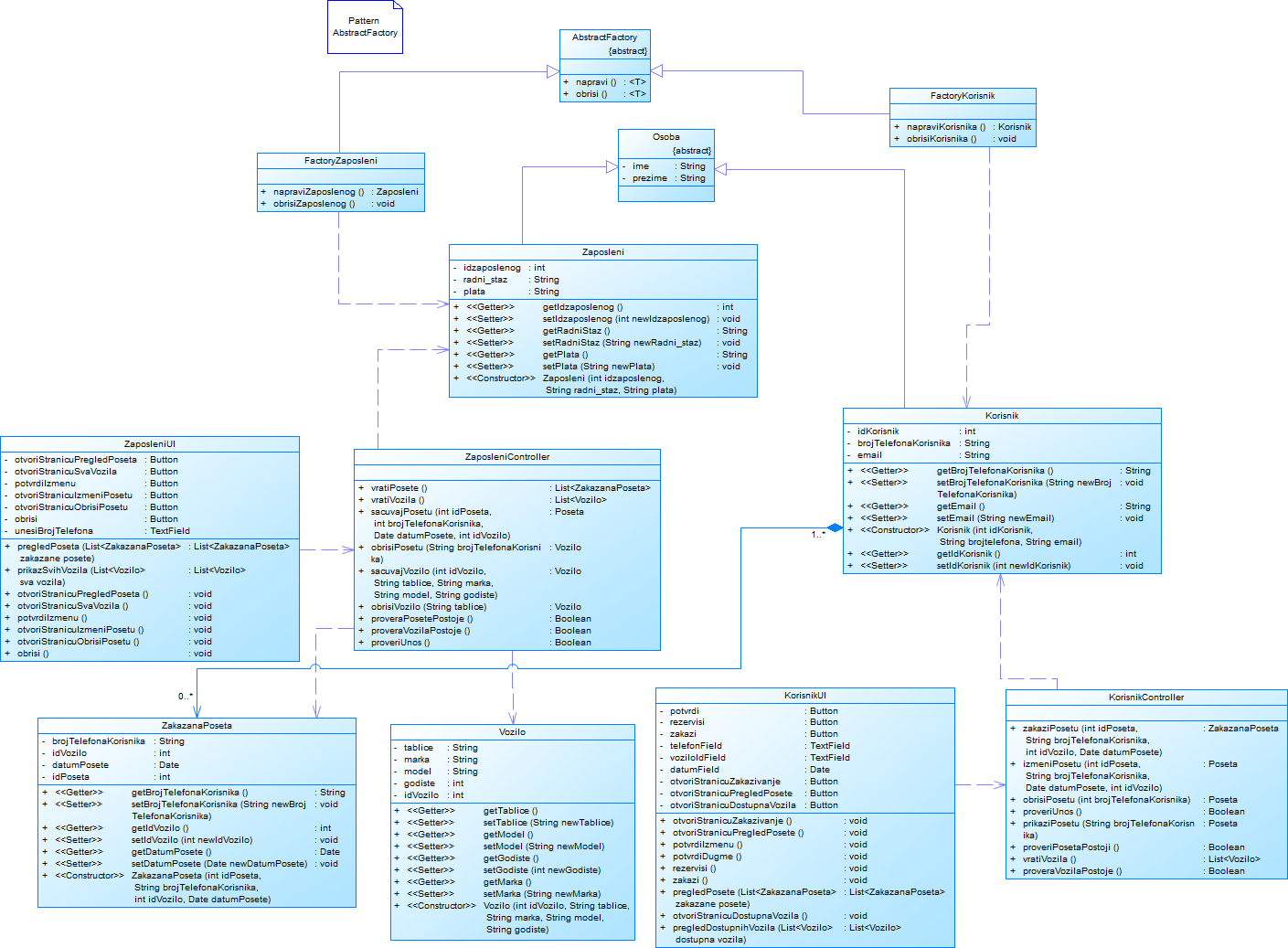
Slika 4.5.4.2 – Detaljni sekvencialni dijagram admina (2.deo)



Slika 4.5.2.3 – Detaljni sekvencialni dijagram admina(3.deo)

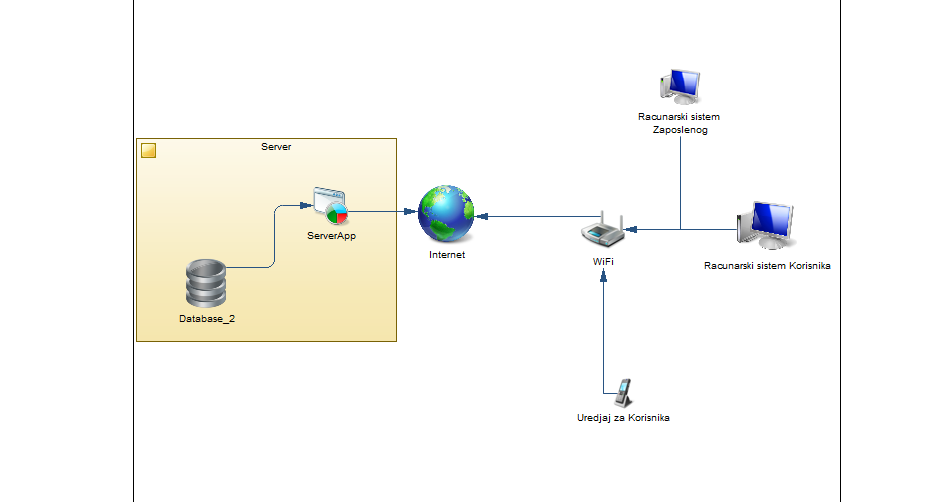
## Klasni dijagram

U klasnom dijagramu se nalaze predviđene klase za kreiranje objekata kako bi se omogućilo potpuno funkcionisanje sistema. Pored samih klasa prikazane su i veze među njima. Takođe je primenjen i pattern AbstractFactory.



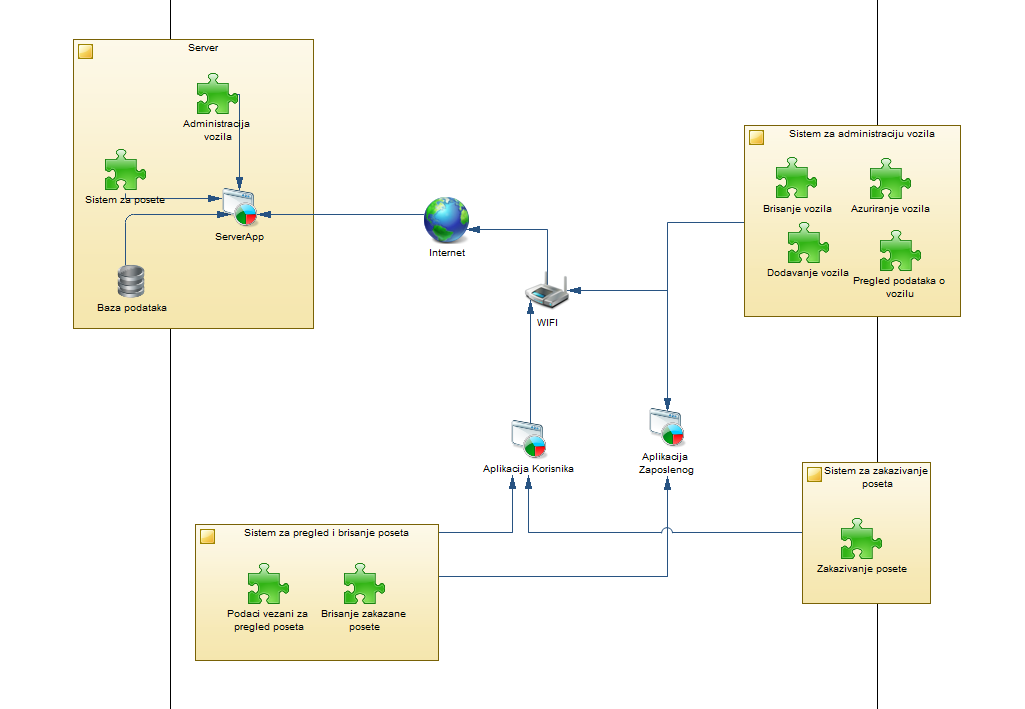
Slika 4.6.1 – Klasni dijagram

4.6.1. Dijagram arhitekture



Slika 4.7.1.1 – Dijagram arhitekture

4.6.2. Infrastrukturni dijagram



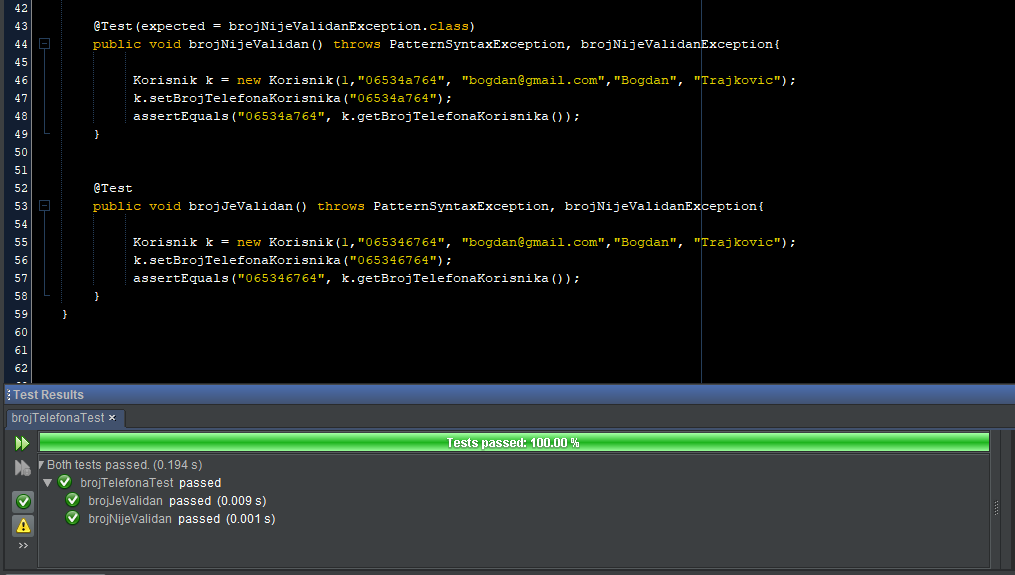
Slika 4.7.2.1 – Dijagram infrastrukture

## Testiranje

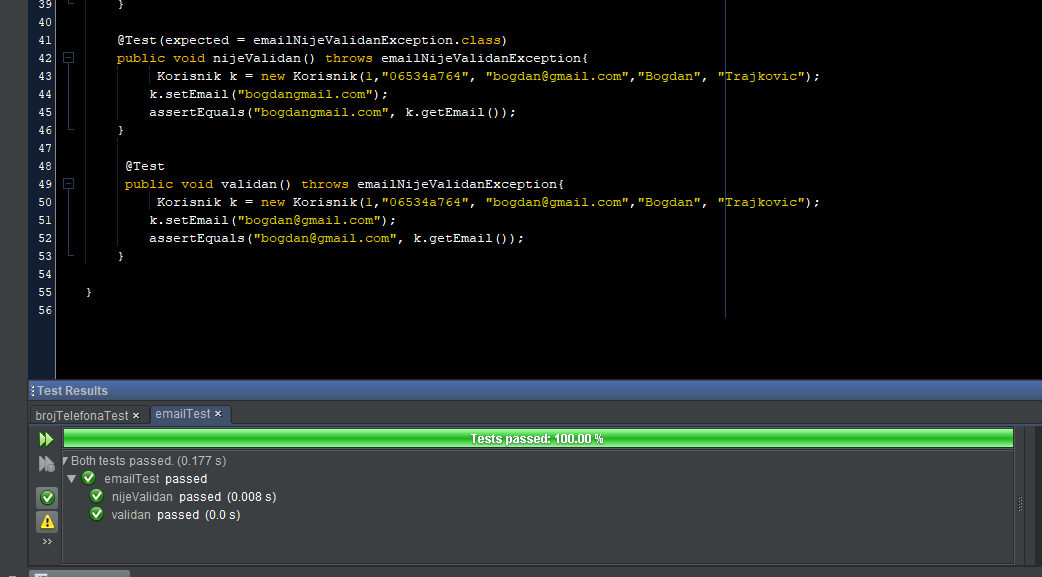
Testiranje modula (jedinično testiranje, unit test) proverava da li pojedinačne komponente ispravno funkcionišu sa svim očekivanim tipovima ulaza, u skladu sa dizajnom komponente. Koristićemo JUnit testove koji su dostupni u NetBeans-u.

U našem slučaju, kreiraćemo metodu koja će biti korišćenja za proveru unetih podataka konretno provera broja telefona koji mora sadržati samo brojeve. Osim toga proveravaćemo da li je godiste automobila broj veći od nule jer ne sme biti negativan broj, validnost email-a kao i da li su određene liste prazne.

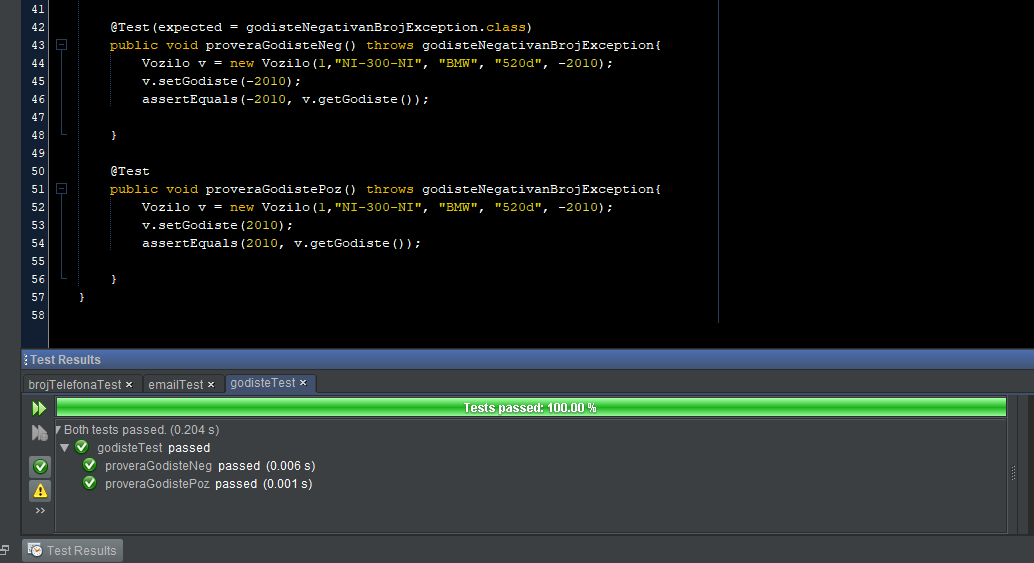
Definisaćemo i metodu koja će se baviti proverom da li je broj telefona validan. Validan je samo ako se satoji samo od brojeva.



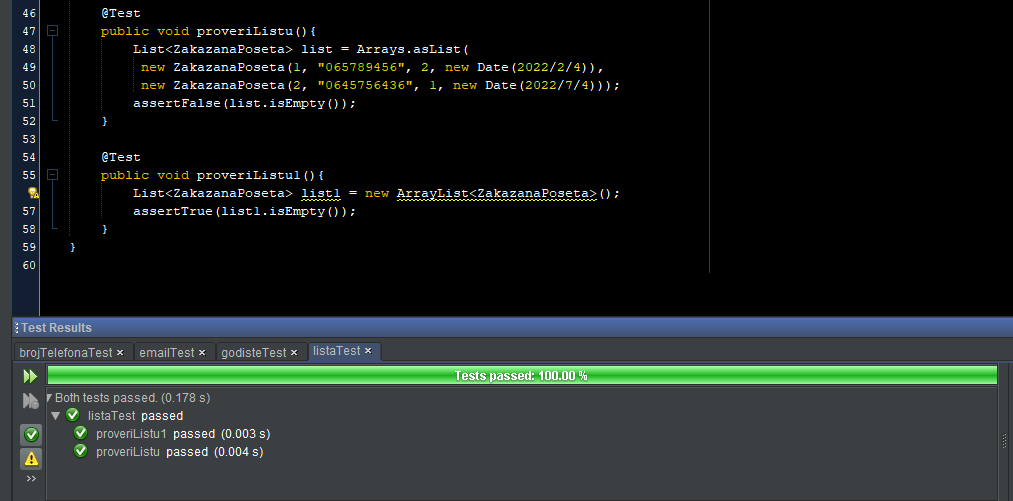
Slika 4.9.2 – Metoda za validaciju broja telefona



Slika 4.9.2 – Testiranje email-a



Slika 4.9.2 – Testiranje godišta



Slika 4.9.2 – Testiranje da li je lista prazna

# Zaključak

Dokumentacija projekta je kao cilj imala što jednostavniji prikaz i objašnjenje rada aplikacije. Ukratko se glavni delovi mogu videti, kako klasni, tako i svi ostali dijagrami korišćeni u projektu, kao i sama arhitektura i infrastruktura istog. Projekat je rađen postepeno uz primenjivanje znanje koje sam stekao. Projekta bih unapredio dodavanjem još dodatnih funkcionalnosti kako bi sistem bio što potupuniji.

# Reference

[1]*SE201- Uvod u softversko inženjerstvo*, Prof.dr Dragan Domazet, 2017.