

Školska godina *2022 /2023*

PREDMET*:* **SE201 – Uvod u softversko inžinjerstvo**

*Tehnička dokumentacija*

Projektni zadatak: **Lanac hotela**

Ime i prezime: **Uros Mirkovic**

Broj indeksa: **5078**

Asistent: **Bogdan Janković**

Datum izrade: **18.01.2023.**

SADRŽAJ

[SADRŽAJ 2](#_Toc125113469)

[OPIS TEME 3](#_Toc125113470)

[POTREBAN FIZIČKE KOMPONENTE 3](#_Toc125113471)

[POTREBAN SOFTVER 3](#_Toc125113472)

[FUNKCIONALNI ZAHTEVI 4](#_Toc125113473)

[NEFUNKCIONALNI ZAHTEVI 6](#_Toc125113474)

[SLUČAJEVI KORIŠĆENJA 7](#_Toc125113475)

[ARHITEKTURA\INFRASTRUKTURA 10](#_Toc125113476)

[KLASNI DIJAGRAM 12](#_Toc125113477)

[SISTEM SEKVECNIJANI DIJAGRAMI 13](#_Toc125113478)

[TESTIRANJE 15](#_Toc125113479)

[LITERATURA 18](#_Toc125113480)

OPIS TEME

Lanac Hotela sadrzi vise hotela povezanih u jednu celinu.Klijenti mogu birati neki od ponudjenih hotela u kojim zele da odsedaju.Klijenti hotela imaju dva nacina rezirvacije preko aplikacije I na samoj recepciji. Kako bi rezervisao sobu klijent mora da izvrsi registraciju na sam sistem.Prilikom rezervacije klijent dobija qr kod koji moze da skenira i dobija kljuc sobe.Mobilna aplikacija moze da omoguci pristup klijentu restoranu samog hotela gde klijent ima mogucnost uplate dorcka,rucka I vecere.Hotel sadrzi I spa koji moze da se rezervise bez rezervacije sobe.Zaposleni na recepciji imaju pristup aplikaciji tako da mogu da vide sve rezervacije.Hotel poseduje i iznajmljivanje automobila ako je to klijentu potrebno.Klijent ima mogucnost iznajmljivanja automobila u jednom hotelu a da automobile vrati u drugom hotelu ako nastavlja tamo odestanje.Hotel sadrzi i spa deo koji klijenti mogu da rezervisu uz sobu i uplate odredjeni paket.

POTREBAN FIZIČKE KOMPONENTE

Racunar za recepciju,Mrezni uredjaj koji povezuje hotele,Racunar za restoran hotela,Citaci QR koda na vratima hotela,GPRS za svaki automobil,Racunar za spa deo.

POTREBAN SOFTVER

1.Baza podataka klijenata i zaposlenih- sadrzi sve trenutne klijente, klijente koji su rezervisali sove i zaposlene u hotelu.

2.Baza svih povezanih hotela – sadzi sve hotele datog lanca.

3.Aplikacija za rezervaciju soba – preko aplikacije izlaze sve raspolozive sobe u datom periodu kojem odaberemo.Kada rezervisemo I platimo preko online bankinga dobijamo qr kod koji mozemo da iskoristimo bez pristupa recepciji da uzmemo datu sobu.

4.Aplikacija za komunikaciju medju zaposlenima – aplikacija sadrzi radna vremena zaposlenih I zadruzenja za dati dan.

5.Aplikacija za dostavu hrane u sobama hotela – sadrzi meni koji je za dati dan raspolozen I cenu samog menija ako klijent nije uplatio.

6.Aplikacija za iznajmljivanje automobile – aplikaciji imaju samo pristup vec registrovani klijenti hotela, aplikacija sadrzi dostupne automobile u tom periodu i u tom hotelu.

7.Aplikacija za rezivisanje spa dela – aplikaciji imaju pristup samo vec registrovani klijenti hotela,aplikacija sadrzi dostupne pakete u spa delu.

FUNKCIONALNI ZAHTEVI





1. *Funkcionalni zahtevi klijenta*

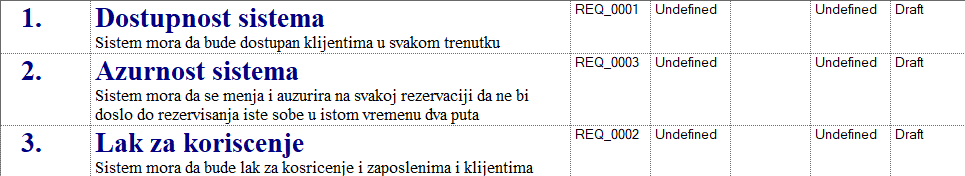
* Klijent ima mogućnost izbora hotela iz menija
* Klijent ima mogućnost rezervacije sobe iz menija
* Klijent ima mogućnost doplate za pansion u hotelu
  + Klijent može da bira između polupanisona,punogpansiona ili noćenja sa doručkom
* Klijent ima mogućnost doplate za spa u hotelu
  + Klijent može da bira između bazena i saune
* Klijent ima mogućnost iznajmljivanja automobila u hotelu
* Klijent ima mogućnost rezervacije online
  + Klijent ima mogućnost plaćanja kreditnom karticom
  + Klijent ima mogućnost plaćanja na samoj recepciji
* Klijent ima mogućnost rezervacije na recepciji
  + Klijent ima mogućnost plaćanja gotovinom
  + Klijent ima mogućnost plaćanja kreditnom karticom
* Klijent ima mogućnost rezervacije samo spa dela
  + Klijent može izabrati između bazena i saune

1. *Funkcionalni zahtevi zaposlenog*

* Zaposleni ima mogućnost kreiranja rezervacije na zahtev klijenta ili na samoj recepciji
* Zaposleni ima mogućnost pregleda svih rezervacija,dodavanje rezervacija i brisanje određenih rezervacija

NEFUNKCIONALNI ZAHTEVI

Nefunkcionalni zahtevi su zahtevi vezani za sam sistem,pod tim zahtevima spadaju dostupnost sistema,ažurnost sistema i lakoća korišćenja.



***DOSTUPNOST SISTEMA***

* Sistem mora da bude dostupan svim klijentima u svakom trenutku kako bi mogli da provere raspoloživost sistema i njihove već kreirane rezervacije.

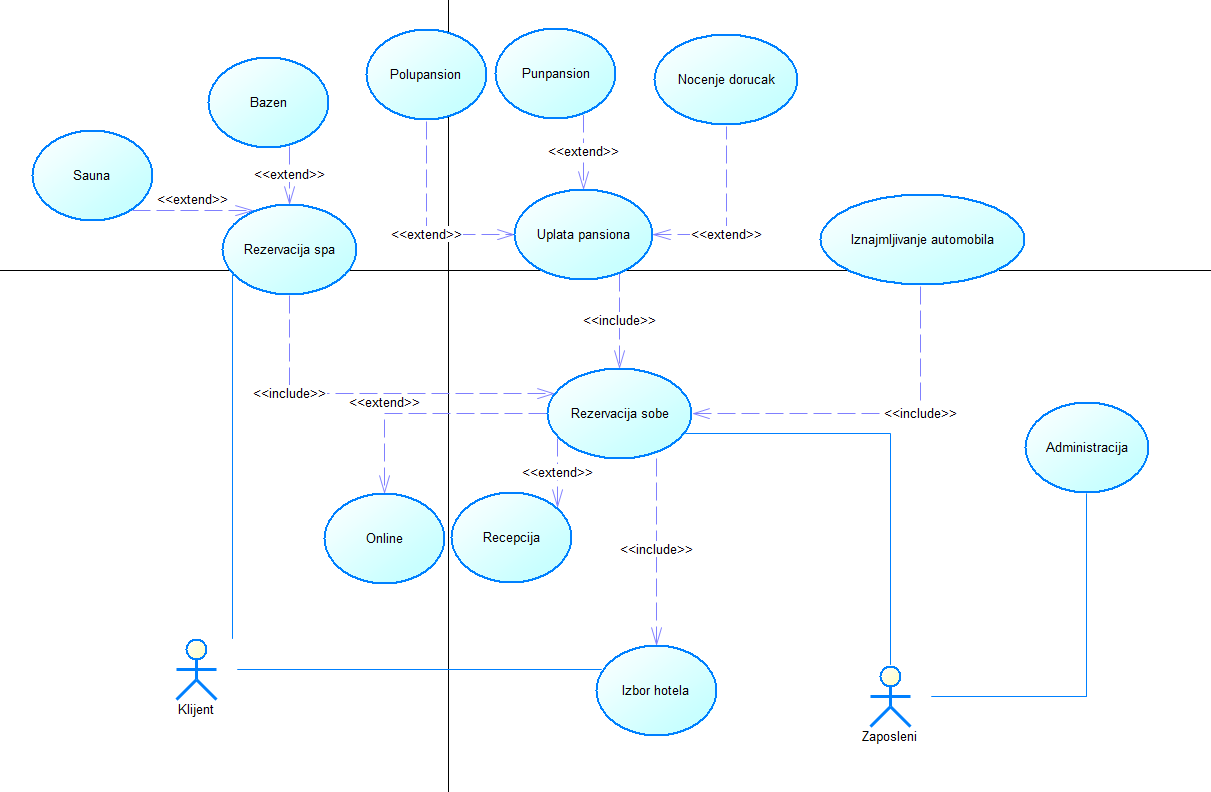
***AŽURNOST SISTEMA***

* Sistem mora da se menja i ažurira na prilikom svake rezervacije tako da ne bi doslo do rezervisanja iste sobe dva puta u istom vremenu.

***LAK ZA KORIŠĆENJE***

* Sistem mora da bude lak i intuitivan za korišćenje, da bi se klijenti lakše snalazili i bolje upravljali aplikacijom. Takođe mora da bude i lak za korišćenje zaposlenima da bi se lako adaptirali na način rada

SLUČAJEVI KORIŠĆENJA



Slučajevi korićenja predstavljaju funcionalne zahteve koji akteri mogu da izvrše sa sistemom prikazane na ***Use Case*** dijagramu.

1. ***SLUCAJEVI KORIŠĆENJA KLIJENTA***

|  |  |
| --- | --- |
| USE CASE | **Izbor hotela** |
| Glavni actor | Klijent |
| Cilj | Omogućiti klijentu da izabere hotel |
| Preduslovi | Klijent mora biti ulogovan |
| Okidač | Klijent je kliknuo na izaberi hotel |
| Primarni scenario | 1. Klijent iz menija bira opciju pregled svih hotela 2. Klijent iz menija bira opciju za pregled određenog hotela |
| Sekundarni scenario | Klijent želi da promeni određeni hotel |

|  |  |
| --- | --- |
| USE CASE | **Rezervacija sobe** |
| Glavni actor | Klijent |
| Cilj | Omogućiti klijentu da izvrši rezervaciju sobe |
| Preduslovi | Klijent mora biti ulogovan i mora da odabere određeni hotel |
| Okidač | Klijent je kliknuo na rezerviši |
| Primarni scenario | 1. Klijent unosi tip sobe koji želi da rezerviše 2. Klijent unosi datum za koji želi da rezerviše sobu 3. Klijent ima mogućnost rezervacije pansiona(USE CASE:Uplata pansiona) 4. Klijent ima mogućnost rezervacije spa dela(USE CASE: Uplata pansiona) 5. Klijent ima mogućnost iznajmljivanje automobile(USE CASE: Iznajmljivanje automobila) 6. Klijent ima mogućnost plaćanja kreditnom karticom online(USE CASE: Online) 7. Klijent potvrđuje rezervaciju |
| Sekundarni scenario | Klijent želi da izmeni određenu rezervaciju |

|  |  |
| --- | --- |
| USE CASE | **Rezervacija spa** |
| Glavni actor | Klijent |
| Cilj | Omogućiti klijentu da rezerviše spa |
| Preduslovi | Klijent mora biti ulogovan, može rezervisati samo spa ili rezervisati spa uz određenu rezervaciju sobe |
| Okidač | Klijent je kliknuo na rezervacija spa |
| Primarni scenario | 1. Klijent iz menija bira opciju pregled svih paketa(Bazen,Sauna) 2. Klijent iz menija bira opciju za pregled određenog paketa 3. Klijent iz menija bira opciju plati onlile(opciono) 4. Klijent bira opciju potvrdi rezervaciju |
| Sekundarni scenario |  |

|  |  |
| --- | --- |
| USE CASE | **Rezervacija pansiona** |
| Glavni actor | Klijent |
| Cilj | Omogućiti klijentu da rezerviše pansion |
| Preduslovi | Klijent mora biti ulogovan i rezervisati pansion uz određenu rezervaciju sobe |
| Okidač | Klijent je kliknuo na rezervacija pansiona |
| Primarni scenario | 1. Klijent iz menija bira opciju pregled svih pansiona(Nocenje doručak,Polupansion,Punpansion) 2. Klijent iz menija bira opciju za pregled određenog pansiona 3. Klijent bira opciju potvrdi izbor pansiona |
| Sekundarni scenario |  |

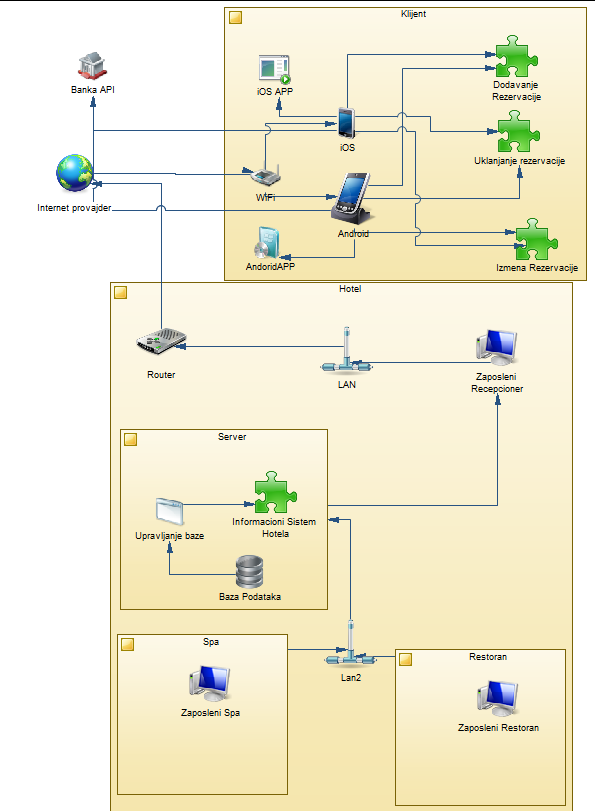
|  |  |
| --- | --- |
| USE CASE | **Iznajmljivanje automobila** |
| Glavni actor | Klijent |
| Cilj | Omogućiti klijentu da iznajmi automobil |
| Preduslovi | Klijent mora biti ulogovan i iznajmiti automobil uz određenu rezervaciju sobe |
| Okidač | Klijent je kliknuo na iznajmljivanje automobila |
| Primarni scenario | 1. Klijent iz menija bira opciju pregled svih automobila 2. Klijent iz menija bira opciju za pregled određenog automobila 3. Klijent bira opciju potvrdi izbor automobila |
| Sekundarni scenario |  |

|  |  |
| --- | --- |
| USE CASE | **Online placanje** |
| Glavni actor | Klijent |
| Cilj | Omogućiti klijentu da plati online |
| Preduslovi | Klijent mora biti ulogovan i imati već rezervaciju koju hoće da plati |
| Okidač | Klijent je kliknuo na plati online |
| Primarni scenario | 1. Klijent iz menija bira opciju plati online 2. Klijent unosi podatke svoje kreditne kartice 3. Klijent bira opciju potvrdi podatke kartice |
| Sekundarni scenario |  |

1. ***SLUCAJEVI KORIŠĆENJA ZAPOSLENOG***

|  |  |
| --- | --- |
| USE CASE | **Administracija** |
| Glavni actor | Zaposleni |
| Cilj | Omogućiti zaposlenom da upravlja rezervacijama |
| Preduslovi | Zaposleni mora da bude ulogovan |
| Okidač | Zaposleni je kliknuo na pregled svih rezervacija |
| Primarni scenario | 1. Zaposleni bira iz menija određenu rezervaciju 2. Zaposleni bira opciju izmeni rezervaciju(opcionalno) 3. Zaposleni bira opciju izbrisi rezervaciju(opcionalno) 4. Zaposleni bira opciju sačuvaj izmene |
| Sekundarni scenario | Zaposleni bira opciju dodaj rezervaciju |

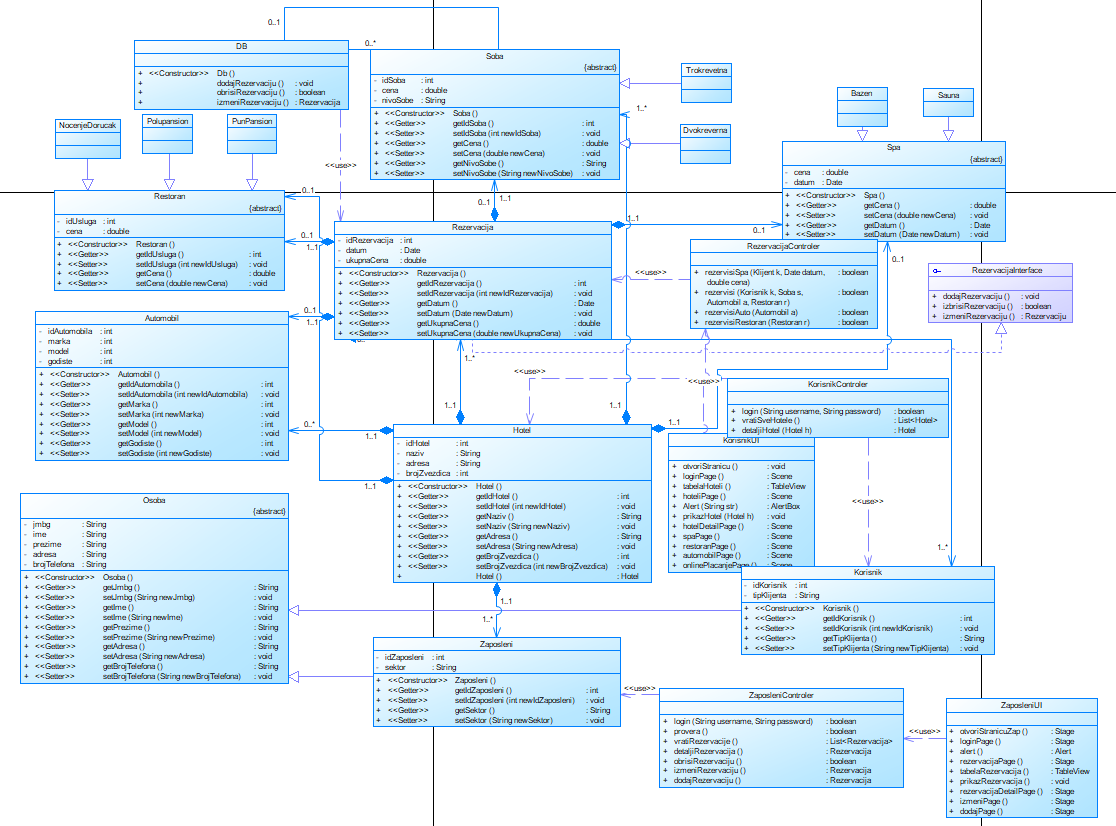
ARHITEKTURA\INFRASTRUKTURA



Dijagram pokazuje na arhitekturu i na infrastrukturu koja je potrebna lancu hotela. Glavne oblasti su spa i restoran koje su u sklopu hotela,informacioni sistem na koji se svaki hotel povezuje. Lanac hotela poseduje bazu koja čuva sve podatke o rezervacijama hotela,zaposlenim,klijentima. Njom se upravlja posebnom aplikacijom koja mora da bude implementirana na računarima u hotelu. U hotelu u spa delu i u restoranu mora da postoji po jedan računar koji služi zaposlenima da vide date rezervacije za taj dan. Klijent moze preko aplikacije da rezervise spa deo,odredđeni pansion za taj dan. Na samoj recepciji mora da postoji računar koji služi za uravljanje celim hotelom. Zaposleni na recepciji može da vidi sve tipove rezervacija i upravlja njima. Svaki hotel ima svoju bazu podataka u kojoj čuva podatke. Sve te baze hotela čine distribuirani sistem hotela i tako olakšava rad samog sistema. Za to komuniciranje potrebna nam je dobra internet konecija. Za online korisnike, koriste se aplikacije u zavisnosti on operativnog sistema može biti Android ili iOS. Aplikacije su pogodne za inkrementalni razvoj jer u samim aplikacijama može dodavati opcije naknadno ako hotel ima potrebu za tim. Ovaj način rezervacije pogodna je jer se kočna rezervacija čuva u bazi tek kada izvršimo celu rezervaciju na uređaju.

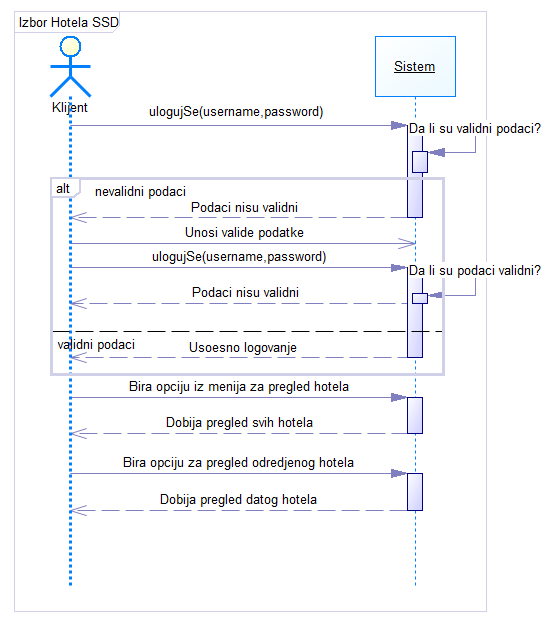
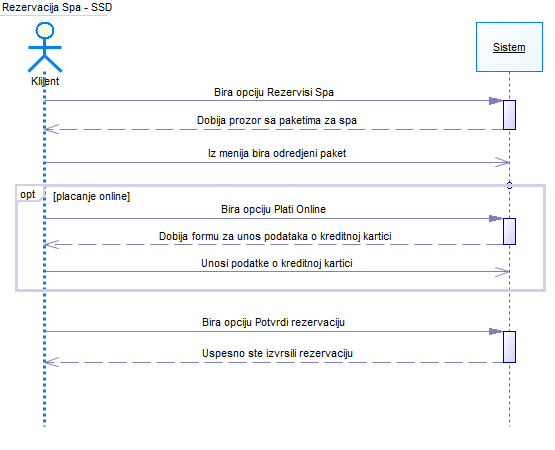
KLASNI DIJAGRAM

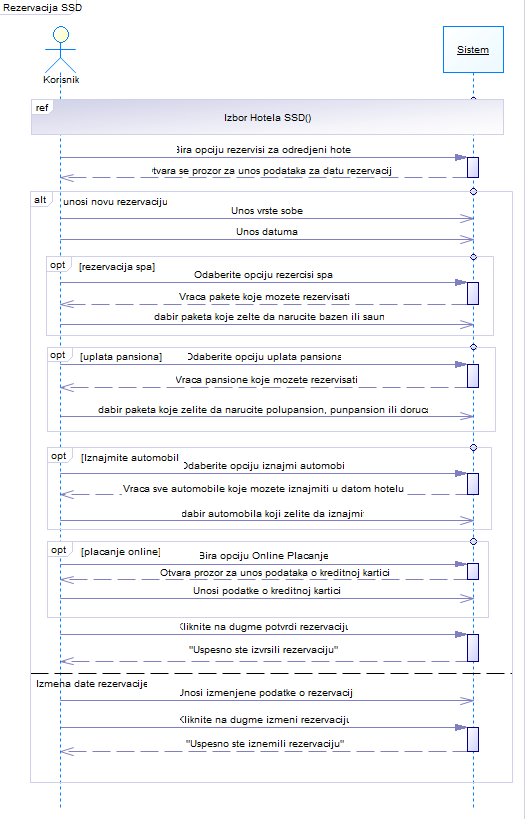
U klasnom dijagramu se nalaze predviđene klase za kreiranje objekata kako bi se omogućilo potpuno funkcionisanje Lanca hotela. Svaka klasa ima određene atribute i metode koje su potrebne za funkcionisanje sistema.Na klasnom dijagramu su prikazane i veze između svih klasa kao i primena patterna ***Singleton,Builder i Decorator***.



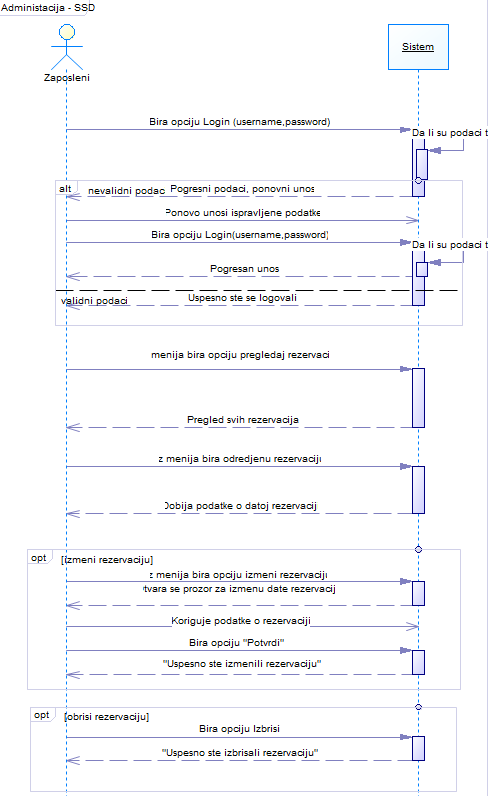
SISTEM SEKVECNIJANI DIJAGRAMI

1. KLIJENT





1. ZAPOSLENI



TESTIRANJE

Testiranje je bitno za softver da bi proverili da li softver radi ono sto mi želimo da radi ili da otkrijemo greške u kodu.Testiranje ovog sistema se zasniva na dodavanje cene na već formiranu cenu rezervacije. Testiranje vršimo JUnit testovima.

*Metoda koja će se testirati:*

public double dodajNaCenu(double cena) throws CenaException{

if (cena>0) {

return this.ukupnaCena+cena;

}else{

throw new CenaException("Ne mozete dodati manje od 1");

}

}

Metoda bi trebala da na već definisanu cenu doda novu cenu. Na cenu možemo dodati vrednosti veće od 0. U slučaju da dodajemo 0 na cenu dobićemo CenaException koji smo prethodno napravili. Za ovu metodu smo napravili 3 testa da bi bili sigurni da radi ono što želimo.

/\*\*

\* Test of dodajNaCenu method, of class Rezervacija.

\*/

@Test

public void testDodajNaCenu() throws CenaException {

System.out.println("dodajNaCenu");

Rezervacija instance = new Rezervacija();

instance.setUkupnaCena(50);

double expResult = 100.0;

double result = instance.dodajNaCenu(50);

assertEquals(expResult, result, 0.0);

}

/\*\*

\* Test of dodajNaCenu method, of class Rezervacija.

\*/

@Test

public void testDodajNaCenu2() throws CenaException {

System.out.println("dodajNaCenu");

Rezervacija instance = new Rezervacija();

instance.setUkupnaCena(50);

double expResult = 110.0;

double result = instance.dodajNaCenu(50);

assertNotEquals(expResult, result, 0.0);

}

/\*\*

\* Test of dodajNaCenu method, of class Rezervacija.

\*/

@Test(expected = CenaException.class)

public void testDodajNaCenu3() throws CenaException {

System.out.println("dodajNaCenu");

Rezervacija instance = new Rezervacija();

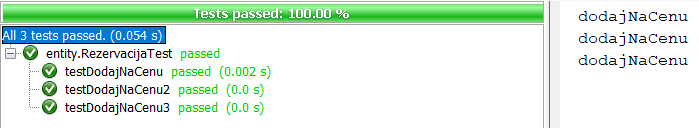
instance.setUkupnaCena(50);

double expResult = 100.0;

double result = instance.dodajNaCenu(0);

assertEquals(expResult, result, 0.0);

}



LITERATURA

METROPOLITAN LEARNING SYSTEM : <http://lams.metropolitan.ac.rs>