

**Jesenji semestar**

**2016/17 PROJEKTNI ZADATAK**

Dokumentacija sistema za evidenciju gostiju hotela

**AUTOR**

**Darko Stojanović - 2725**

**OAS**

**Smer: Softversko Inženjerstvo**

[**darko.stojanovic.2725@metropolitan.ac.rs**](mailto:darko.stojanovic.2725@metropolitan.ac.rs)

Beograd, februar 2017

Sadržaj

[**1. Uvod** 3](#_Toc479104841)

[1.1 Svrha dokumenta 3](#_Toc479104842)

[1.2 Pregled dokumenta 3](#_Toc479104843)

[1.3 Razvojne metodologije 4](#_Toc479104844)

[**2. Opis** 4](#_Toc479104845)

[2.1 Funkcije sistema i moduli 4](#_Toc479104846)

[2.2 Ograničenja sistema 4](#_Toc479104847)

[**3. Zahtevi i specifikacije** 5](#_Toc479104848)

[3.1 Uvod 5](#_Toc479104849)

[3.2 Korisnici sistema 5](#_Toc479104850)

[3.3 Funkcionalni zahtevi 5](#_Toc479104851)

[3.3.1 Administratorski deo 5](#_Toc479104852)

[3.3.2 Deo za recepcionera 6](#_Toc479104853)

[3.4 Nefunkcionalni zahtevi 7](#_Toc479104854)

[3.4.1 Performanse 8](#_Toc479104855)

[3.4.2 Lakoća upotrebe 8](#_Toc479104856)

[3.4.3 Zaštita 8](#_Toc479104857)

[3.4.4 Raspoloživost 8](#_Toc479104858)

[3.4.5 Lakoća održavanja 8](#_Toc479104859)

[3.4.6 Bezbednost 8](#_Toc479104860)

[3.5 Slučajevi upotrebe (UseCase dijagrami) 10](#_Toc479104861)

[3.5.1 Administrator sistema 10](#_Toc479104862)

[3.5.2 Radnik na recepciji hotela 16](#_Toc479104863)

[**4.0 Dizajn** 23](#_Toc479104864)

[4.1 Strukturni dizajn 23](#_Toc479104865)

[4.1.1 Klasni dijagram 23](#_Toc479104866)

[4.1.2 Dizajn baze podataka 24](#_Toc479104867)

[4.2 Arhitektura sistema 26](#_Toc479104868)

[4.3 Infrastruktura sistema 27](#_Toc479104869)

[4.4 Procesni dizajn (Dizajn ponašanja) 28](#_Toc479104870)

[4.4.1 Dijagrami sekvenci 28](#_Toc479104871)

[4.4.2 Dijagrami aktivnosti 32](#_Toc479104872)

[**5.0 Implementacija i testiranje** 34](#_Toc479104873)

[5.1 Dizajn pattern 34](#_Toc479104874)

[5.1.1 Korišćene tehnologije 34](#_Toc479104875)

[5.1.2 Korišćeni dizajn paterni 34](#_Toc479104876)

[5.2 Testiranje jedinica 35](#_Toc479104877)

[**6.0 Zaključak** 35](#_Toc479104878)

# 1. Uvod

Tema ovog projekta je izrada I dokumentovanje sistema za evidentiranje gostiju jednog hotela. Može se izmeniti I prilagoditi za slične oblasti poslovanja . Sistem je napravljen tako da što više olakša posao budućim korisnicima.

## 1.1 Svrha dokumenta

Svrha ovog dokumenta jeste da pruži uvid u način funkcionisanja ovog sistema. Način rada biće objašnjen korišćenjem UML dijagrama od kojih će svaki imati svoj opis I način primene u celokupnom sistemu. Alat koji je korišćen za izradu dijagrama je SAP PowerDesigner. Za realizaciju biće korišćen Java programski jezik jer je najprikladniji za softver ovakvog tipa. Ovaj dokument predstavlja samu predrazvojnu fazu.

## 1.2 Pregled dokumenta

Ovaj dokument sadrži sledeće:

1. Detaljan uvid u dijagrame slučajeva korišćenja

2. Detaljan uvid u svaki slučaj korišćenja

3. Sekvencijalne dijagrame slučajeva korišćenja

4. Sistem sekvencijalne dijagrame slučajeva korišćenja

5. Detaljan uvid u dijagram klasa

6. Prikaz svake klase sa pojašnjenjem metoda i atributa

7. Konceptualni model baze podataka

8. Fizički model baze podataka

9. Detaljan prikaz modela infrastrukture

10. Detaljan uvid u dijagram arhitekture sistema

11. Rezultate testiranja pojedinih klasa

## 1.3 Razvojne metodologije

Zahtevi korisnika za ovaj sistem su jasno definisani od samog početka tako da bi najprikladniji model razvoja bio model vodopada u kombinaciji sa inkrementalnim razvojem. Akcenat je stavljen na projektovanje i izradu osnovnih funkcija sistema kako bi bio funkcionalan, dok će se u narednim verzijama (inkrementima) proizvoda dodavati nove funkcionalnosti u skladu sa željama i potrebama budućih korisnika.

# 2. Opis

## 2.1 Funkcije sistema i moduli

Sistem za evidentiranje gostiju hotela predstavlja jednostavno napravljen softverski sistem sačinjen od dva modula. Moduli omogućavaju različitim tipovima korisnika da koriste ovaj sistem. Svaki tip korisnika ima svoju ulogu.

Podsistem za administratora – Administrator sistema ima svoj nalog kojim pristupa sistemu. Administrator može dodavati nove zaposlene, koji su recepcioneri u hotelu. Takođe može bristati zaposlenog iz evidencije o radu, menjati njegove podatke i zahtevati pregled svih zaposlenih. Svi podaci koje administrator unese za radnike čuvaju se u bazi podataka.

Podsistem za radnika na recepciji hotela – Recepcioner koristi nalog koji je za njega otvorio administrator. Ovaj sistem je najviše pravljen za radnike na recepciji hotela. Oni mogu rezervisati sobe, upravljati sobama (dodati, izmeniti, obrisati i tražiti sobu). Takođe mogu upravljati evidencijom gostiju i rezervacijama. Sve promene i rezervacije čuvaju se u bazi podataka.

## 2.2 Ograničenja sistema

Ovaj sistem napravljen je za samo za jedan hotel a ne za ceo lanac hotela tako da više hotela ne može biti povezano preko njega. Trenutno postoji samo jedna lokalna baza podataka koju koristi jedan hotel. Integracioja više hotela u ovaj sistem trenutno nije moguća. Ovaj nedostatak se može prevazići kasnijim izmenama i redizajniranjem određenih komponenti ovog sistema. Po potrebi korisnika razvojni tim može uzeti u obzir razvijanje delova sistema koji bi omogućili njegovo korišćenje u svim ostalim hotelima ovog lanca. Ovakva izmena bi se izvršila kao nadogradnja posotjećeg sistema kao jedan od inkremenata.

# 3. Zahtevi i specifikacije

## 3.1 Uvod

Osnovni cilj ovog sistema je da zaposlenima jednog hotela olakša svakodnevni rad na poslu. Od velike važnosti je brzina i efikasnost obavljanja radnji sistema. Korisnicima će biti omogućeno lako korišćenje. Od njih će se samo tražiti da popunjavaju jednostavne forme i evidentiraju goste. Obuka zaposlenih nije potrebna za korišćenje sistema.

## 3.2 Korisnici sistema

Analizirajući sve funkcionalnosti koje ovaj sistem treba da obezbedi budućim korisnicima, možemo razlikovati dve osnovne grupe korisnika. Prvu grupu predstavljaju zaposleni u hotelu tj. radnici na recepciji, dok drugu grupu predstavljaju administratori sistema. Njihova uloga je da sistem održavaju i menjaju podatke u bezi podataka vezane za radnike. Svaki od ovih modula biće analiziran i objašnjen u posebnom delu ovog dokumenta.

Dve role koje postoje u okviru ovog sistema su:

1. Administrator
2. Radnik na recepciji hotela

Zahtevi odnosno slučajevi korišćenja će se definisati na nivou svake role ovog sistema.

## 3.3 Funkcionalni zahtevi

### 3.3.1 Administratorski deo

Administrator će imati mogućnost da se prijavi na sistem korišćenjem svog naloga. Pored toga, on može da registruje naloge svim radnicima tog hotela. Odabirom opcije ‚‚menadžment osoblja“ administrator otvara prozor za dodavanje novih radnika. Kako bi dodao novog radnika od informacija mora uneti njegov identifikacioni broj, šifru koju će koristiti za pristup nalogu, njegovo zvanje, ime, prezime, kontakt telefon i adresu stanovanja.

***Prijava administratora***

Kako bi administrator bio u mogućnosti da pristupa sistemu, neophodno je da se prijavi na svoj nalog unošenjem korisničkog imena i lozinke. Nakon uspešne autentikacije (provera da li se poklapaju unete informacije) i autorizacije (pravo na manipulisanje sistemom i bazom podataka), administrator može obavljati sve aktivnosti koje mu sistem obezbeđuje. Prijava administratora je neophodna kako bi se ograničilo neautorizovano korišćenje funkcionalnosti kojima samo administrator ima pristup.

***Pravljenje novog naloga***

*Staff management – add new* – Administrator unosi potrebne informacije za registraciju naloga za novog radnika. Od informacija potrebno je da unese ime, prezime, identifikacioni broj, šifru koju radnik koristi za pristup, kontakt telefon, zvanje i adresu stanovanja.

***Pretraga zaposlenih***

*Search employee –* Administrator odabirom ove opcije može pretraživati bazu podataka zaposlenih. Od informacija potrebno je da unese identifikacioni broj zaposlenog, ime zaposlenog ili prezime zaposlenog. Klikom na dugme ‚‚search“ dobija sve podatke o traženom zaposlenom.

***Brisanje zaposlenog***

*Delete employee –* Administrator može ukloniti zaposlenog iz baze podataka i tako prekinuti radni odnos. Potrebno je da dok je u delu za prikaz svih zaposlenih, odabere jednog zaposlenog koga će obrisati. Pre nego što se brisanje izvrši, administrator će dobiti obaveštenje o brisanju i moraće da klikne dugme ‚‚ok“ kako bi potvrdio brisanje.

***Izmena podataka zaposlenih***

*Edit employee –* Aministrator ima mogućnost da izmeni podatke o zaposlenom. Od informacija o zaposlenom može promeniti sve osim identifikacionog broja zaposlenog i broja telefona. Kako bi bio u mogućnosti da menja informacije o zaposlenom, administrator se mora nalaziti u delu za prikaz svih zaposlenih. Nakon što popuni sve informacije, mora potvrditi izmene klikom na dugme ‚‚ok“.

### 3.3.2 Deo za recepcionera

Na početku recepcioner se mora prijaviti na nalog koji mu je administrator napravio. Unosi svoje korisničko ime i lozinku.

***Prijava recepcionera***

*Log in –* Recepcioner se prijavljuje na svoj nalog. Nakon uspešne prijave može koristiti sistem. Takođe proveravaju se input polja, ne mogu biti prazna.

***Rezervacija sobe***

*Room booking –* Od potrebnih informacija, recepcioner unosi datum dolaska (bira datum iz kalendara kom pristupa klikom na ikonicu) i datum odlaska (datum bira na isti način kao kod datuma dolaska). Nakon toga iz liste soba bira jednu koju će rezervisati. Pre nego što je rezerviše, mora proveriti dostupnost sobe klikom na dugme ‚‚check room“. Ukoliko je soba dostupna za rezervaciju, klikom na dugme ‚‚book room“ recepcioner rezerviše izabranu sobu.

***Upravljanje sobama***

*Room management –* Radnik može dodavati, pretraživati, menjati i brisati sobe. Kako bi dodao sobu, potrebno je da unese broj sobe, tip sobe, cenu sobe po jednom noćenju i tip kreveta u sobi. Ukoliko želi da menja sobu, radnik se mora nalaziti u delu programa u kome se nalazi lista soba. Menja podatke o onoj sobi koja je trenutno izabrana. Od podataka može izmeniti sve sem broja sobe. Procedura za brisanje sobe je slična kao kod izmene sobe. Korisnik se mora nalaziti u istom delu programa i obrisati izabranu sobu iz selekcije. Ukoliko želi da pretražuje sobe, potrebno je da unese broj sobe za koju traži informacije.

***Upravljanje evidencijom gostiju***

*Visitor management –* Odabirom ove opcije, recepcioner može pregledati evidenciju gostiju. Može pretraživati goste hotela i tražiti od sistema spisak svih gostiju. Od podataka za pretragu korisnik može uneti ime, grad, državu, okrug ili identifikacioni broj gosta. Listu svih gostiju dobija samim odabirom opcije za upravljanje gostima. Kroz listu se kreće pomoću dugmadi ‚‚next“ i ‚‚previous“.

***Upravljanje rezervacijama***

*Booking management –* U ovom delu sistema recepcioner može modifikovati, otkazivati i pregledati sve rezervacije. Pregledanje svih rezervacija je obezbeđeno samim odabirom opcije booking management. Korisnik se kreće kroz rezervacije pomoću dugmadi ‚‚next“ i ‚‚previous“. Modifikovanje rezervacija može se izvršiti unosom potrebnih podataka i pritiskom na dugme ‚‚modify booking“. Otkazivanje rezervacije vrši se klikom na dugme ‚‚cancel booking“ dok se recepcioner nalazi u delu u kom je prikazana lista svih rezervacija.

## 3.4 Nefunkcionalni zahtevi

Nefunkcionalni zahtevi predstavljaju zahteve koji nisu direktno u vezi sa servisima i funkcionalnostima koje sistem treba da obezbedi korisnicima, već su povezani sa svojstvima tih servisa. Neka od tih svojstava su performanse, pouzdanost, brzina reagovanja sistma na određeni zahtev korisnika, količina memorije koju sistem koristi, bezbednost, raspoloživost, prenosivost i lakoća održavanja. Oni definišu ograničenja implementaciji sistema, kao što su na primer, svojstva U/I uređaja, ili predstavljanje podataka na interfejsu sa drugim sistemima.

### 3.4.1 Performanse

Performanse imaju veliki značaj u većini sistema. Za svaki sistem potrebno je obezbediti što bolje performanse kako bi korisnik mogao da koristi sistem na najbolji način. U slučaju ovog sistema, performanse mogu imati značajan uticaj na rad samog sistema. Dobre performanse dovode do lakšeg i bržeg korišćenja, što u praksi znači manje čekanja gostiju hotela da se evidentiraju u bazu gostiju. Dobre performanse ovog sistema najviše zavise odbaze podataka. Upisivanje i čitanje podataka treba da bude konkurentno, da ne dolazi do gubitka podataka ili velikog čekanja za upis ili čitanje. Smanjenje redudantnosti podataka u bazi takođe dovodi do poboljšanja performansi.

### 3.4.2 Lakoća upotrebe

Za svaki podsistem je potrebna kraća obuka budučih korisnika. Sistem je po sebi intuitivan za korišćenje ali je ipak korisnicima koji se teže snalaze za računarom potrebna obuka.

### 3.4.3 Zaštita

Sistemu nije potreban visok nivo zaštite jer je pristup bazi podataka ograničen. Sam pristup bazi ograničen je autentikacijom i autorizacijom oba tipa korisnika (radnika i administratora).

### 3.4.4 Raspoloživost

Sistem treba da bude dostupan za korišćenje na laptop i desktop uređajima.

### 3.4.5 Lakoća održavanja

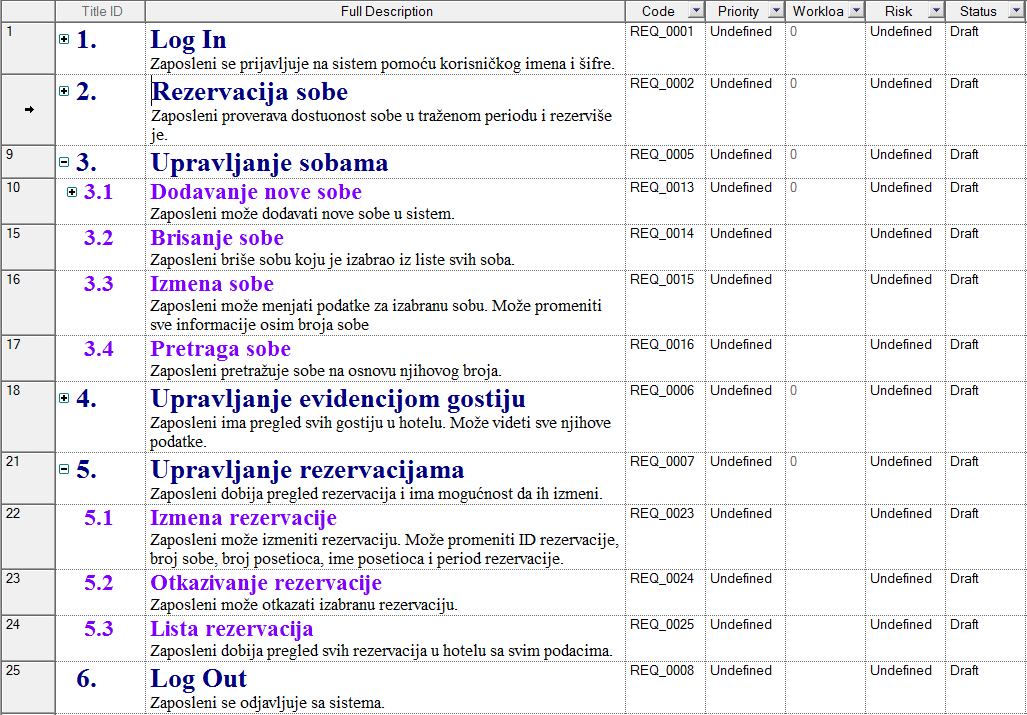
Akcenat je na tome da se olakša razvoj eventualnih naknadnih funkcionalnosti tako što treba težiti da se u što većoj meri koriste postojeće tj. već razvijene komponente.

### 3.4.6 Bezbednost

Najkritičnija komponenta sistema je ustvari sama konekcija sa bazom podataka, zbog toga što funkcionisanje podsistema administratora i radnika na recepciji direktno zavisi od informacija koje su smeštene u bazi podataka sistema.



Slika 1: Funkcionalni zahtevi za administratora sistema

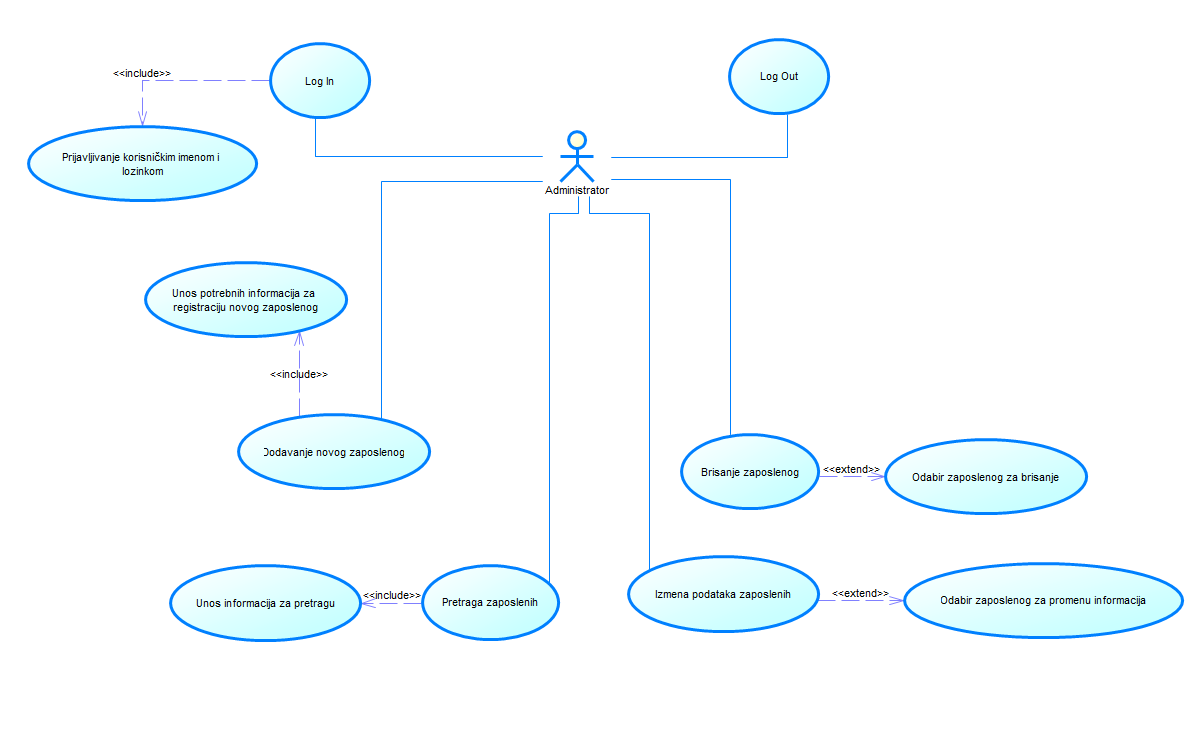


Slika 2: Funkcionalni zahtevi za radnika na recepciji

## 3.5 Slučajevi upotrebe (UseCase dijagrami)

### 3.5.1 Administrator sistema

Na sledećim slikama biće prikazani dijagrami slučajeva korišćenja za korisnike sistema, za administratora i radnika na recepciji hotela.



Slika 3: UseCase dijagram za administratora sistema

* ***Prijava administratora na sistem***

**Učesnici:**

Administrator i sistem.

**Opis slučaja korišćenja:**

Administrator unosi svoje korisničko ime i lozinku pomoću kojih je u mogućnosti da koristi sistem.

**Preduslovi:**

Korisnik ima pristup sistemu, konekcija na internet nije neophodna.

**Osnovni scenario:**

1. Administrator unosi svoje korisničko ime i lozinku u za to predviđena input polja i prijavljuje se na sistem pritiskom dugmeta ‚‚Log In“.
2. Sistem proverava da li su uneti podaci validni i u zavisnosti od toga obaveštava administratora.

**Alternativni scenariji:**

1. Administrator unosi pogrešne podatke za log in ili ostavlja neka polja nepopunjena
2. Sistem prikazuje obaveštenje o grešci.

**Postuslovi:**

1. Administrator je obavešten o uspešnom prijavljivanju, preusmeren je na svoj podsistem i može da nastavi sa korišćenjem aplikacije.

* ***Dodavanje novog zaposlenog***

**Učesnici:**

Administrator sistema i sistem.

**Opis slučaja korišćenja:**

Administrator želi da doda novog zaposlenog koji je dobio posao u hotelu.

**Preduslovi:**

1. Administrator ima pristup bazi podataka i konekcija sa bazom je uspešno uspostavljena i stabilna.
2. Administrator je iz menija ponuđenih opcija izabrao ‚‚Dodavanje novog zaposlenog“.

**Osnovni scenario:**

1. Administrator unosi podtrebne podatke za dodavanje novog zaposlenog. Ti podaci su ime, prezime, identifikacioni broj, šifra koju radnik koristi za pristup, kontakt telefon, zvanje i adresa stanovanja.
2. Sistem šalje obaveštenje administratoru da je uspešno napravio nalog zaposlenom.

**Alternativni scenariji:**

1. Administrator nije uneo dobre informacije za dodavanje novog zaposlenog ili se neke informacije poklapaju (ukoliko je zaposleni već dodat).
2. Sistem prikazuje poruku o grešci.

**Postuslovi:**

1. Administrator je uspešno napravio novi nalog i dodao zaposlenog u evidenciju.

* ***Brisanje zaposlenog***

**Učesnici:**

Administrator sistema i sistem.

**Opis slučaja korišćenja:**

Administrator želi da ukloni radnika iz sistema koji više nije u radnom odnosu.

**Preduslovi:**

1. Administrator ima pristup bazi podataka.
2. Administrator je odabrao opciju za brisanje naloga zaposlenih.

**Osnovni scenario:**

1. Administrator mora biti u delu za prikaz svih zaposlenih, odabira jednog zaposlenog koga će obrisati.
2. Sistem šalje obaveštenje administratoru da je uspešno obrisao izabranog zaposlenog.

**Alternativni scenariji:**

1. Administrator ne uspeva da obriše zaposlenog zbog greške u komunikaciji sa bazom.
2. Sistem izbacuje poruku o grešci i koracima kako da se greška otkloni.

**Postuslovi:**

1. Administrator je uspešno obrisao nalog zaposlenog.

* ***Pretraga zaposlenih***

**Učesnici:**

Administrator sistema i sistem.

**Opis slučaja korišćenja:**

Administrator želi da pronađe određenog zaposlenog u bazi zaposlenih.

**Preduslovi:**

1. Administrator ima pristup bazi podataka.
2. Administrator je izabrao opciju za pretraživanje zaposlenih.

**Osnovni scenario:**

1. Administrator mora biti u delu za prikaz svih zaposlenih, u polje za pretragu unosi bar jedan uslova za pretraživanje.
2. Sistem šalje povratne informacije o traženom zaposlenom.

**Alternativni scenariji:**

1. Administrator ne uspeva da pronađe zaposlenog na osnovu pretraživanih kriterijuma.
2. Sistem šalje poruku administratoru da traženi zaposleni ne postoji u bazi podataka.

**Postuslovi:**

1. Administrator je uspešno pronašao informacije o traženom zaposlenom.

* ***Izmena podataka zaposlenih***

**Učesnici:**

Administrator sistema i sistem.

**Opis slučaja korišćenja:**

Administrator sistema želi da izmeni određene podatke za zaposlene.

**Preduslovi:**

1. Administrator ima pristup bazi podataka.
2. Administrator se nalazi u delu sistema koji mu omogućuje izmene podataka korisnicima.

**Osnovni scenario:**

1. Administrator mora biti u delu za prikaz svih zaposlenih, odabira jednog zaposlenog kome će menjati podatke.
2. Sistem šalje obaveštenje administratoru da je uspešno promeni podatke izabranom zaposlenom.

**Alternativni scenariji:**

1. Administrator ne može menjati podatke zaposlenom jer je došlo do greške sa konekcijom na bazu podataka.
2. Sistem šalje obaveštenje o grešci sa uputstvima za njeno rešavanje.

**Postuslovi:**

1. Administrator je uspešno promenio podatke izabranom zaposlenom i sašuvao promene u bazi podataka.

* ***Odjava administratora***

**Učesnici:**

Administrator i sistem.

**Opis slučaja korišćenja:**

Administrator želi da se odjavi sa sistema.

**Preduslovi:**

1. Administrator je prethodno uspešno log inovan na sistem.

2. Administrator je odabrao opciju ‚‚Log Out“.

**Osnovni scenario:**

1. Administrator klikom na dugme za odjavljivanje sistemu ukazuje da želi da se odjavi.

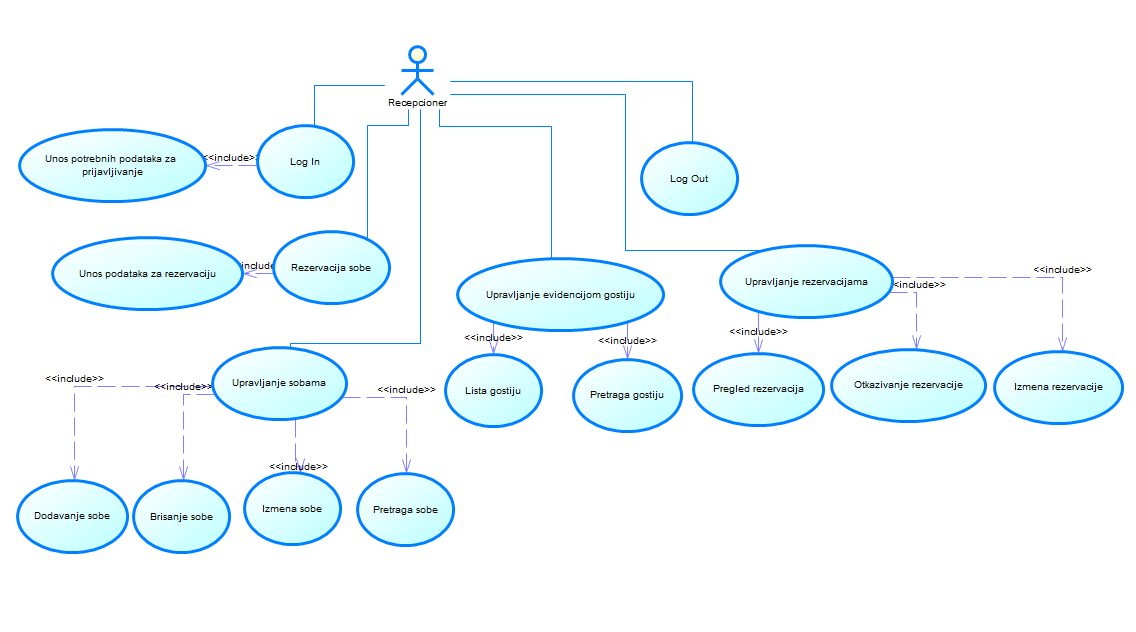
2. Sistem odjavljuje administratora.

**Postuslovi:**

1. Administrator više nije prijavljen na sistem.

### 3.5.2 Radnik na recepciji hotela

Na sledećoj slici prikazan je UseCase dijagram za radnika na recepciji u hotelu.



Slika 4: UseCase dijagram za radnika na recepciji

* ***Prijava recepcionera na sistem***

**Učesnici:**

Recepcioner (korisnik) i sistem.

**Opis slučaja korišćenja:**

Recepcioner se prijavljuje na sistem nalogom koji mu je administrator napravio.

**Preduslovi:**

1. Recepcioner ima pristup bazi podataka.

**Osnovni scenario:**

1. Recepcioner unosi svoje korisničko ime i lozinku kako bi mogao da pristupi sistemu.
2. Sistem šalje poruku o uspešnom prijavljivanju na sistem.

**Alternativni scenariji:**

1. Korisnik je uneo pogrešno korisničko ime ili lozinku.
2. Sistem izbacuje poruku o grešci.

**Postuslovi:**

1. Korisnik je uspešno prijavljen na sistem i može koristiti sve funkcionalnosti.

* **Rezervacija sobe**

**Učesnici:**

Recepcioner (korisnik) i sistem.

**Opis slučaja korišćenja:**

Recepcioner želi da rezerviše jednu od slobodnih soba za goste hotela.

**Preduslovi:**

1. Recepcioner ima pristup bazi podataka.
2. Recepcioner se nalazi u delu sistema koji mu omogućuje rezervacije soba.

**Osnovni scenario:**

1. Recepcioner se nalazi u delu za rezervisanje soba, bira jednu sobu koja je slobodna i rezerviše je.
2. Sistem šalje obaveštenje recepcioneru da je uspešno rezervisao sobu.

**Alternativni scenariji:**

1. Recepcioner pokušava da rezerviše sobu koja je već rezervisana.
2. Sistem obaveštava receocionera da je soba zauzeta i da rezerviše neku drugu.

**Postuslovi:**

1. Recepcioner je uspešno rezervisao sobu za goste hotela.

* ***Upravljanje sobama***

**Učesnici:**

Recepcioner (korisnik) i sistem.

**Opis slučaja korišćenja:**

Recepcioner želi da izvrši promene u sistemu za upravljanje soba.

**Preduslovi:**

1. Recepcioner se nalazi u delu sistema koji mu obezbeđuje upravljanje sobama.
2. Recepcioner ima pristup bazi podataka.

**Osnovni scenario:**

1. Recepcioner se nalazi u delu za upravljanje sobama i bira jednu od opcija: dodavanje sobe, brisanje sobe, izmena sobe i pregled soba i menja izabrane podatke.
2. Sistem šalje povratnu poruku da su sve izmene sačuvane.

**Alternativni scenariji:**

1. Recepcioner pokušava da promeni informacije ali sistem prijavljuje problem sa konekcijom na bazu podataka.
2. Recepcioner pokušava da promeni podatke za nepostojeću sobu.
3. Recepcioner pokušava da doda sobu ali je kapacitet premašen.
4. Sistem u svakom od ovih slučajeva vraća odgovarajuće obaveštenje.

**Postuslovi:**

1. Recepcioner je uspešno izmenio željene podatke o sobama.

* ***Upravljanje evidencijom gostiju***

**Učesnici:**

Recepcioner (korisnik) i sistem.

**Opis slučaja korišćenja:**

Recepcioner želi da izvrši promene u evidenciji gostiju.

**Preduslovi:**

1. Recepcioner ima pristup bazi podataka.
2. Recepcioner se nalazi u delu sistema koji mu omogućuje rad na evidenciji gostiju.

**Osnovni scenario:**

1. Recepcioner se nalazi u delu programa za upravljanje evidencijom gostiju. Bira između sledećih opcija: pretraga gostiju i lista gostiju.
2. Sistem šalje poruku o uspešnim promenama.

**Alternativni scenariji:**

1. Recepcioner pokušava da promeni informacije ali sistem prijavljuje problem sa konekcijom na bazu podataka.
2. Recepcioner pokušava da pronađe nepostojećeg gosta.
3. U listi gostiji trenutno nema gostiju.
4. Sistem recepcioneru vraća odgovarajuće obaveštenje za svaki od slučajeva.

**Postuslovi:**

1. Recepcioner je uspešno pretražio evidenciju gostiju hotela.

* ***Upravljanje rezervacijama***

**Učesnici:**

Recepcioner (korisnik) i sistem.

**Opis slučaja korišćenja:**

U ovom delu sistema recepcioner može modifikovati, otkazivati i pregledati sve rezervacije. Pregledanje svih rezervacija je obezbeđeno samim odabirom opcije booking management.

**Preduslovi:**

1. Recepcioner se nalazi u delu sistema za upravljanje rezervacijama.
2. Recepcioner ima uspostavljenu konekciju sa bazom podataka.

**Osnovni scenario:**

1. Recepcioner se nalazi u delu za upravljanje rezervacijama i bira opciju: pregled svih rezervacija i izmena rezervacija.
2. Sistem šalje odgovor da su promene uspešno izvršene i sačuvane.

**Alternativni scenariji:**

1. Recepcioner pokušava da promeni informacije i pristupi listi rezervacija ali sistem prijavljuje problem sa konekcijom na bazu podataka.
2. Recepcioner pokušava da vidi rezervacije a u evidenciji nema unetih rezervacija.
3. Recepcioner pokušava da promeni podatke za izabranu rezervaciju ali ne uspeva jer je došlo do greške u komunikaciji sa bazom.
4. Sistem šalje odgovarajuću potuku za prethodne slučajeve.

**Postuslovi:**

1. Recepcioner je uspešno pregleda listu rezervacija i izmenio željene rezervacije.

* ***Odjavljivanje recepcionera sa sistema***

**Učesnici:**

Recepcioner (korisnik) i sistem.

**Opis slučaja korišćenja:**

Korisnik (recepcioner) želi da se odjavi sa sistema.

**Preduslovi:**

1. Korisnik je prijavljen na sistem.

**Osnovni scenario:**

1. Recepcioner pritiskom na dugme ‚‚Log Out“ ukazuje sistemu da želi da se odjavi.
2. Sistem odjavljuje korisnika sa njegovog naloga.

**Postuslovi:**

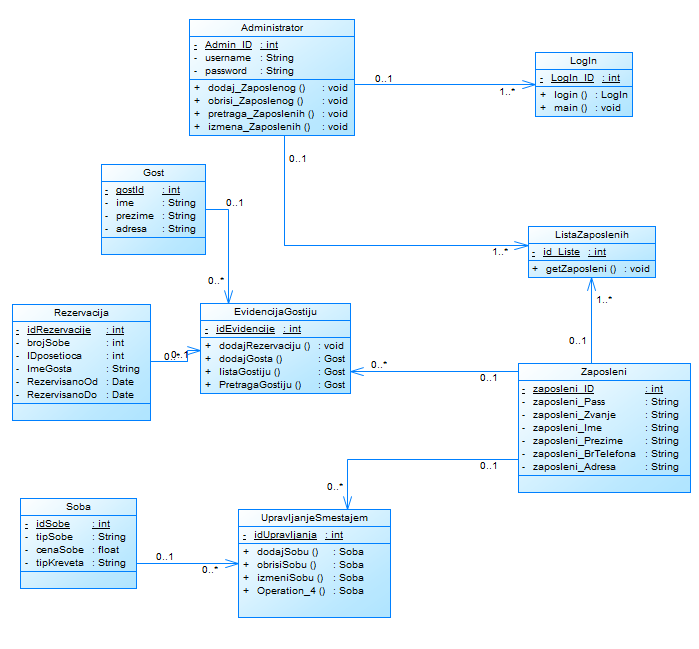
1. Korisnik se uspešno odjavio sa sistema.

# 4.0 Dizajn

## 4.1 Strukturni dizajn

### 4.1.1 Klasni dijagram

Na sledećoj slici može se videti prikaz klasnog dijagrama gde je definisana primarna struktura klasa kao i relacije između njih.



Slika 5: Klasni dijagram

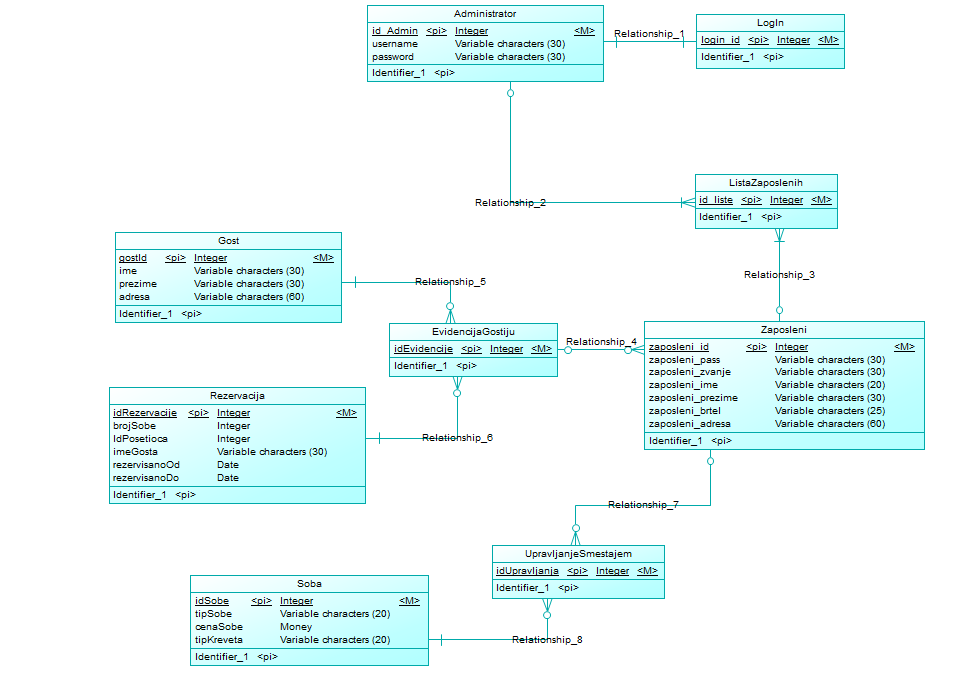
Na slici se može videti povezanost klasa sistema. Sve klase predstavljaju Java POJO (*Plain Old Java Object*) klase što podrazumeva da sadrže metode za get i set za svaki atribut kao i toString metodu koje su izostavljene u dijagramu zbog preglednosti.

Klasa ‘‘Zaposeni“ eksterno dobija popunjavanje svojih atributa tj. kao sto je vec spomenuto ranije u slučajevima upotrebe i opisu projekta, administrator dodaje sve informacije i podatke pri kreiranju naloga za zaposlenog. Svaki korisnik u zavisnosti od toga kojem podsistemu pripada, upisuje potrebne informacije.

### 4.1.2 Dizajn baze podataka

***Konceptualni dijagram***

Na sledećoj slici predstavljen je konceptualni model (E-R) dijagrama klasa.

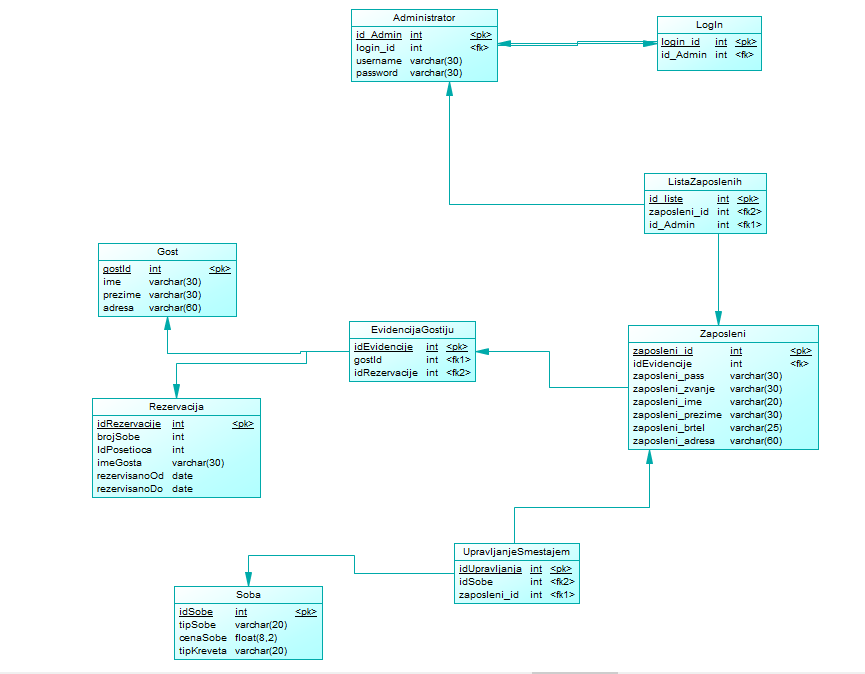


Slika 6: Konceptualni klasni model

Na slici 6 može se videti konceptualni model ovog sistema za evidenciju gostiju hotela i predstavlja prvi prikaz modela baze gde vidimo prirodu veza između entiteta sistema. Prikazan je tok podataka za administratora sistema i radnika na recepciji. Na slici se vidi koja polja su obavezna i koji atributi su primarni i strani ključevi.

***Fizički model baze podataka***

Na sledećoj slici predstavljen je fizički model baze podataka.

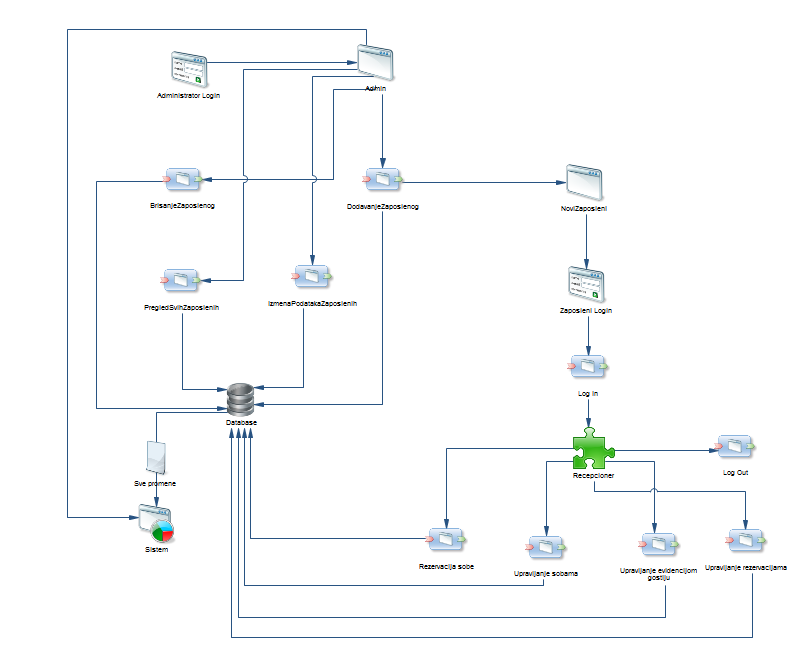


Slika 7: Fizički model baze podataka

Na osnovu fizičkog modela možemo uvideti relacije između tabela kao i u koje tabele se spuštaju primarni ključevi drugih tabela. Svaka od tabela ima svoj primarni ključ. Deo podsistema za administratora definisan je tabelama administrator, login, lista zaposlenih i zaposleni. U međutabelu lista zaposlenih spuštaju se primarni ključevi tabela zaposleni i administrator. Deo podsistema definisan je tabelama koje se nalaze uglavnom na levoj strani prethodne slike. Zaposleni kontroliše upravljanje smeštajem i evidenciju gostiju. Evidencija gostiju je međutabela koja spaja tabele rezervacija i gost. U nju se spuštaju primarni ključevi ove dve tabele. Upravljanje smeštajem je međutabela u koju se spuštaju primarni ključevi tabela zaposleni i soba.

## 4.2 Arhitektura sistema

Na sledećoj slici predstavljen je dijagram arhitekture sistema.



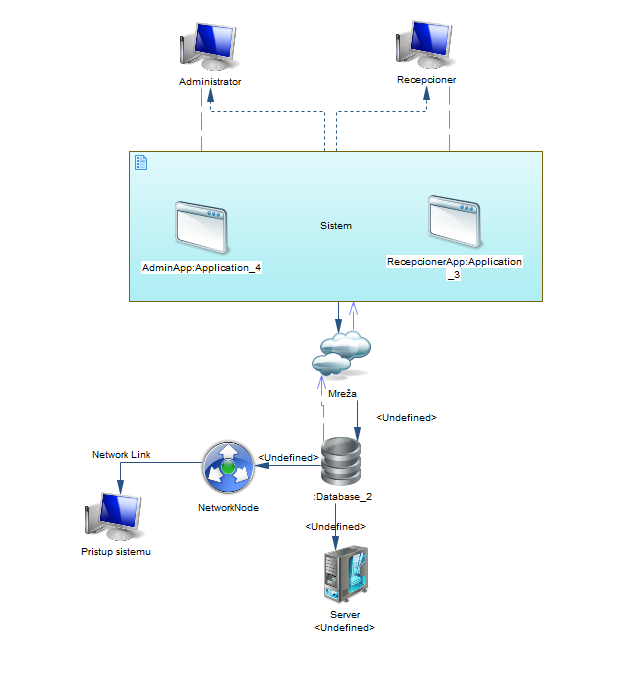
Slika 8: Dijagram arhitekture sistema

Arhitektura ovog softverskog sistema je predstavljena kroz par komponenti dijagrama arhitekture. Komponente su sistem, ulazi, veze I tok podataka, aplikacije, posebne komponente sistma, rad aplikacija i baza podataka.

Najvažnija komponenta ovog sistema je baza podataka. Svi podaci i sve izmene čuvaju se u bazi podataka. Front-end komponenta Log in admin dozvoljava administratoru da se uloguje na nalog I napravi aplikaciju Novi zaposleni. Nakon pravljenja naloga od strane administratora, korisnik sistema postaje radnik na recepciji hotela. On dalji rad izvršava odabirom jedne od njegovih mogućih funkcionalnosti. Nakon završetka rada svi podaci se automatski skladište u bazu podataka.

## 4.3 Infrastruktura sistema

Na sledećoj slici predstavljen je dijagram infrastrukture sistema.



Slika 9: Dijagram Infrastrukture sistema

Ovaj dijagram pokazuje krajnje jednostavnu infrastrukturu ovog sistema. Uključuje najosnovnije elemente, neopohodne za funkcionisanje jednog softverskog sistema. Administrator može pristupiti sistemu putem interneta preko bilo kog računara, dok je drugim zaposlenima u sistemu omogućen pristup takođe koristeći računar. Program označen plavom bojom sadrži podsisteme potrebne za funkcionisanje celog sistema. Podsistemi su povezani sa računarima na kojima rade korisnici sistema sa jedne strane, dok se sa druge strane program povezuje sa bazom podataka.

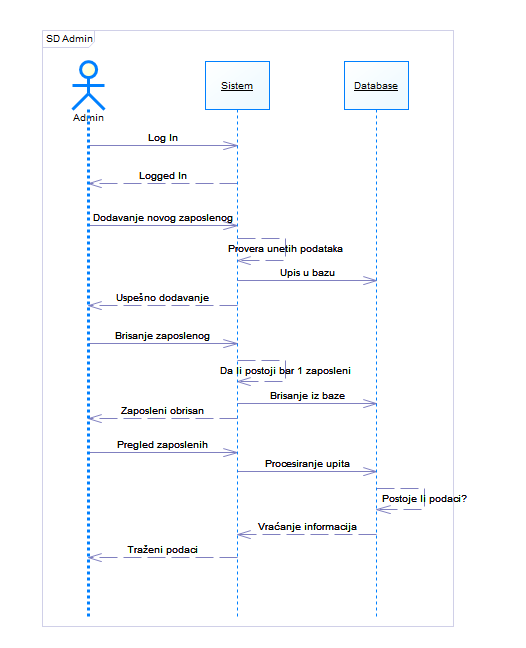
## 4.4 Procesni dizajn (Dizajn ponašanja)

### 4.4.1 Dijagrami sekvenci

Za svaki od podsistema softvera, koji su opisani, biće prikazan dijagram sekvenci i pored toga će biti opisan tok interakcije u alternativnim putanjama i situacijama.

Prvo će biti prikazan sekvencijalni dijagram za administratora sistema na sledećoj slici.

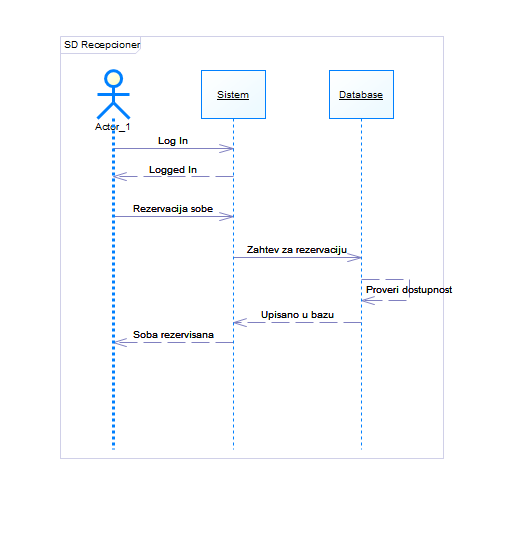
***Sekvencijalni dijagram za administratora***



Slika 10: Sekvencijalni dijagram administratora

Komunikacija se vrši između administratora, sistema i baze podataka. Definisane su provere da li se izvršenje akcije odvija po planu. Jedna od provera jeste provera unetih podataka. Ostale provere su da li postoje zaposleni i da li postoje podaci u bazi podataka.

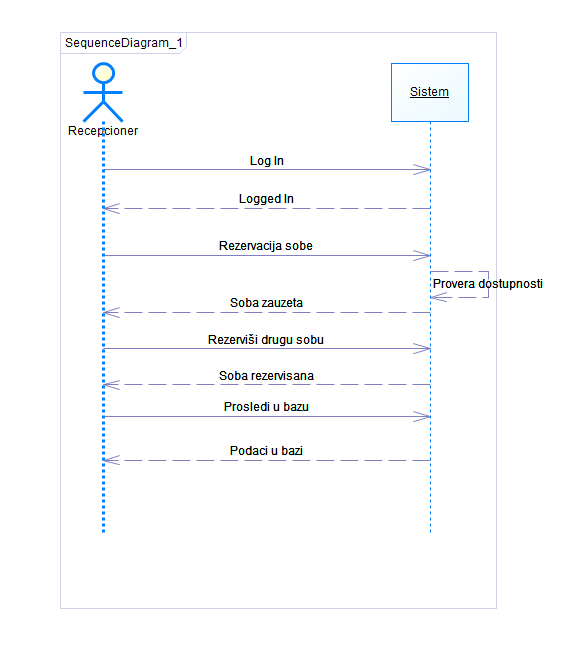
***Sekvencijalni dijagram za radnika na recepciji***



Slika 11: Sekvencijalni dijagram recepcionera

Komunikacija se vrši između recepcionera, sistema i baze podataka. Definisane su provere da li se izvršenje akcije odvija po planu. Provera je dostupnost sobe, tj da li u bazi podataka već postoji rezervacija sa istim brojem sobe.

***Sistem sekvencijalni dijagram***

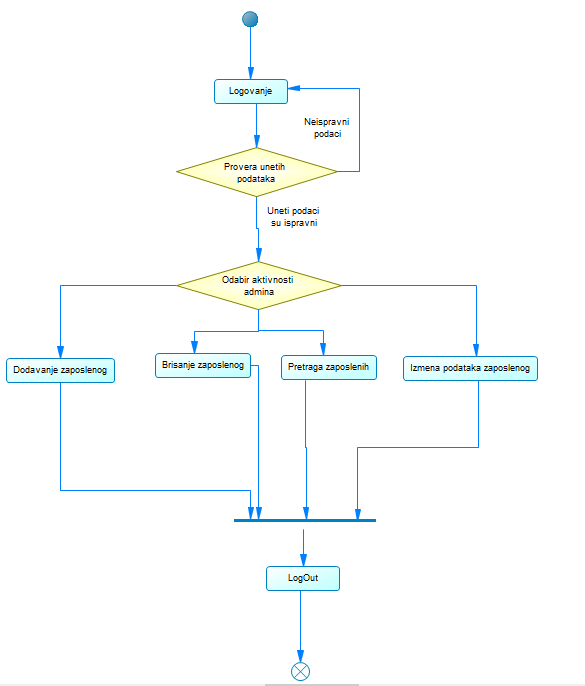


Slika 12: Sistem sekvencijalni dijagram

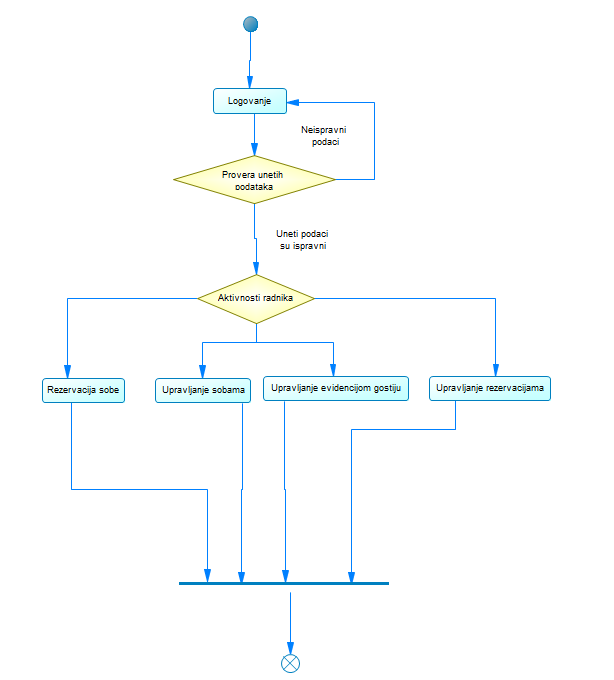
Ovaj Sistem Sekvencijalni Dijagram pokazuje razmenu poruka između korisnika, koji je u ovom slučaju recepcioner hotela, i sistema. SSD je pogled na funkcionisanje komunikacije između sistema i nekog korisnika.

### 4.4.2 Dijagrami aktivnosti

U okviru ovog dela dokumentacije biće prikazani dijagrami aktivnosti za ovaj sistem.



Slika 13: Dijagram aktivnosti za administratora sistema

****

Slika 14: Dijagram aktivnosti za recepcionera

# 5.0 Implementacija i testiranje

## 5.1 Dizajn pattern

### 5.1.1 Korišćene tehnologije

Razvoj je izvršen Java Spring okvirom što uključuje korišćenje dizajn paterna koji su implementirani unutar pomenutog okvira. Kada je u pitanju arhitektura sistema okvir nameće MVC (Model View Controller) arhitektonski patern.

### 5.1.2 Korišćeni dizajn paterni

Ovaj sistem nije implementiran tako da je uvid u implementaciju pružen na osnovu prethodnih iskustava. Za implementaciju ovog softverskog sistema najprikladniji bio bi *Model View Controller Architercture* patern. MVC deli softverski sistem u tri dela po opstoj definiciji ovog paterna:

* ***View*** - Obezbeđuje korisniku interfejs pomoću koga korisnik unosi podatke i poziva odgovarajuće operacije koje treba da se izvrše nad modelom. View prikazuje korisniku stanje modela.
* ***Controller*** - Osluškuje i prihvata zahtev od klijenta za izvršenje operacije. Nakon toga poziva operaciju koja je definisana u modelu, ukoliko model promeni stanje, obaveštava view o promeni stanja.
* ***Model*** - Predstavlja stanje sistema koje mogu promeniti operacije modela.



Slika 15: Model MVC Arhitekture

## 5.2 Testiranje jedinica

Kada se radi o testiranju samog projekta važno je istaći da sam se opredelio za metodu testiranja jedinica. (Unit testing). Kao krajnji cilj ovog metoda testiranja postavlja se verifkacija ispravnosti samog koda (samim tim testiranje nivoa uspešnosti realizacije zahteva). Testiranje jedinica bi se vršilo postepeno. Prvenstveno bi se istestirale sve CRUD operacije vezane uz administratorski modul i modul za recepcionera (dodavanje, brisanje, izmena i čitanje iz baze).

Testiranje CRUD operacija je vežno jer se funkcionisanje celog sistema oslanja na to. Nakon ovog testiranja poželjno bi bilo testirati pojedinačne metode. Ovo je jako važan deo jer se može desiti da određeni metodi vraćaju pogrešne vrednosti što bi dovelo do neupotrebljivosti sistema. Ovakvo testiranje je najbolje sprovesti dok je sistem još uvek u izradi.

Na kraju potrebno je sprovesti testiranje celokupnog sistema i u tom poslednjem koraku testiranja pronaći što više grešaka i otkloniti ih. Nijedan sistem ne može biti potpuno oslobođen grešaka, ali se detaljnim testiranjem broj tih grešaka može smanjiti. Svaki dački inkrement sistema i nadogradnja zahtevaće dodatna testiranja kako bi se obezbedilia i poboljšala funkcionalnosti sistema.

# 6.0 Zaključak

Iako je ovaj softverski sistem jednostavan i prost za razumeti, uz ovaj dokument njegovo razumevanje bi trebalo biti olakšano. Ono što je prednost ovakvog dokumenta je to što daje detaljan uvid čak i u bazu podataka i način ralizacije čitvog projekta što podrazumeva i korišćene tehnologije (što je bitno ljudima koji su u pozadini i vrše održavanje sistema). Dokumentacija sistema je veoma bitna korisnicima sistema, ljudima koji se nalaze iza samog sistema. Sistem ima prostora za još modula i različitih podsistema i nadogradnji. Početna verzija sistema je napravljenja tako da zadovolji potrebe korisnika i predstavi projekat koji se drži pravila izrade softverskih sistema i kao takav daje inženjerima uvid i mogućnost da ovaj sistem još više usavrše.