
Dokumentacija sistemskog dizajna

Hotel Customer Care

Verzija: 1.1

Autori: Aleksandar Lekić, Bojan Suvajac, Daniel Crnovčić,
Dragan Bunić, Ljubiša Milinčić, Svetozar Vuković

Datum: 22/9/2017

Sadržaj

1	Uvod	1
1.1	Namjena sistema	1
1.2	Projektni ciljevi	1
1.3	Definicije i skraćenice	2
1.4	Referentni dokumenti	3
1.5	Pregled	3
2	Arhitektura postojećeg sistema	4
3	Predložena arhitektura	5
3.1	Kratak pregled arhitekture i funkcionalnosti podsistema	5
3.2	Dekompozicija sistema	5
3.3	HW/SW mapiranje	6
3.4	Perzistentni sloj	8
3.5	Kontrola prava pristupa i sigurnost	10
3.6	Kontrola toka	11
3.7	Granična stanja sistema	12

1 Uvod

1.1 Namjena sistema

Softverski sistem koji će biti proizveden je nazvan **Hotel Customer Care** (u daljem tekstu HCC). Sistem će biti proizveden za sve zainteresovane klijenete koji žele da na jasan, intuitivan i prije svega moderan način obogate usluge hotelijerstva svojim cijenjenim gostima. HCC je u osnovi namijenjen kako velikim lancima hotela tako i malim hotelima čija je ponuda raznolika. Svaka hotelska soba će biti opremljena tako da će postojati interfejs prema aplikaciji što će omogućiti da gosti imaju efikasnu realizaciju svojih zahtjeva. Sistem će biti realizovan kao desktop aplikacija čiji je glavni cilj da olakša komunikaciju između gostiju hotela i zaposlenih radnika koji će biti ažurniji u pružanju usluga. Ova aplikacija je pogodna za hotele koji žele svojim gostima da obezbijede bržu rezervaciju usluga hotela i samim tim da smanje broj problema koje nosi direktno naručivanje usluga.

1.2 Projektni ciljevi

HCC je **klijent-server** aplikacija koja podržava veći broj funkcija za interakciju korisnika sa sistemom i adekvatnim **grafičkim korisničkim interfejsom (GUI)**. Aplikacija će se moći koristiti na srpskom i na engleskom jeziku. Aplikacija je dostupna svakom ko ima napravljen i aktivan nalog na sistemu.

Glavne karakteristike ove aplikacije su:

- Pruža operativnu podršku za najpoznatije operativne sisteme.
- Podržava istovremeni rad sa većim brojem korisnika.
- Pruža sve usluge koje se očekuju od jednog modernog hotela.
- Omogućava kontaktiranje recepcionara za svaki problem koji nije naveden u FAQ sekciji.

HCC će obezbijediti sljedeće funkcije:

- Dozvoljava korisnicima da se prijave i odjave sa sistema.
- Dozvoljava svakom prijavljenom korisniku da koristi usluge hotela.
- Pružanje sobnih usluga korisnicima sistema.
- Rezervisanje termina za:
 - Restoran,
 - Wellness centar,
 - Sportske aktivnosti.
- Kreiranje popusta.
- Čuvanje informacija o korisnicima.
- Prikaz slobodnih soba.
- Naplata računa gostima hotela.
- Prikaz obavještenja.
- Pregled knjige utisaka, kao i kreiranje novog utiska.

Postoje dvije klase korisnika HCC sistema:

- **Gost** - ne posjeduje nikakve privilegije na sistemu, prijavljuje se u hotel kako bi mogao dobiti pristup sistemu.
- **Recepcionar** - posjeduje najviše privilegije i prava pristupa na sistemu, može dodavati, brisati i mijenjati korisničke naloge.

Korisnik sistema prilikom rezervacije sobe dobija korisničko ime i lozinku pomoću kojih se može prijaviti na sistem. Sistem će omogućiti gostima hotela uvid u sve usluge koje su koristili tokom boravka u hotelu. Takođe, sistem će omogućiti korisnicima naručivanje svih usluga koje hotel pruža u odgovarajućim terminima. Pored usluga, sistem će omogućiti gostima kompletan uvid u troškove, gdje će gosti moći pregledati datum, vrijeme, naziv i cijenu svake od iskorištenih usluga, kao i ukupnu vrijednost računa. Korisnik sistema će imati mogućnost da iskoristi popust na sve usluge jednostavnim unosom odgovarajućeg koda. Sistem će omogućiti prikaz oglasa kreiranih od strane recepcionara te prikaz digitalne knjige utisaka u kojoj će gosti moći kreirati nove i pregledati postojeće utiske.

Prijava na sistem biće omogućena i recepcionaru. Prilikom rezervacije sobe, recepcionar, preko sistema, kreira novi korisnički nalog za gosta i daje mu korisničko ime i lozinku. Recepcionar će imati prikaz svih registrovanih gostiju koji su trenutno u hotelu, te uvid u njihove račune. Kada gost napušta hotel, recepcionar mora da označi odgovarajući račun korisnika kao plaćen. Kada korisnik rezerviše uslugu, sistem će generisati obavještenje koje će biti prikazano recepcionaru. Sistem će obezbijediti recepcionaru mogućnost manipulacije proizvodima koje gost može da naruči, kreiranje i prikaz popusta, oglasa, te brisanje nedostojanstvenih utisaka.

Sistem će imati neka ograničenja. Plaćanje putem kreditne kartice neće biti omogućeno. Nije dozvoljeno da nalog predviđen za recepcionara bude ujedno i nalog predviđen za gosta. Nalozi svih recepcionara će biti kreirani tokom izrade aplikacije, a ukoliko se javi potreba za kreiranje novog naloga za recepcionara, potrebno je kontaktirati razvojni tim.

1.3 Definicije i skraćenice

Tabela 1: Popis definicija i skraćenica

HCC	Hotel Customer Care
Klijent-Server	Predstavlja način organizacije informacionog sistema u kom jedna komponenta aplikacije, koja se izvodi na jednom računaru (klijent), zahtjeva servis od druge aplikativne komponente koja se izvršava na drugom računaru (server).
FAQ	Skraćenica od <i>Frequently Asked Questions</i> ili u prevodu Često postavljana pitanja. Predstavlja spisak najčešćih pitanja o nekoj temi, i sažetih odgovora na ta pitanja.
GUI	<i>Graphical User Interface</i> – Grafički korisnički interfejs. Omogućava interakciju između korisnika i računara preko grafičkih ikona i indikatora.
CSS	<i>Cascading Style Sheets</i> – Jezik formatiranja pomoću kojeg se definiše izgled dokumenta napisanog jezikom za označavanje (eng. <i>Markup language</i>), kao što su HTML ili XML.
Baza podataka	Kolekcija podataka organizovanih za brzo pretraživanje i pristup.

DBMS	<i>Database Management System</i> ili sistem za upravljanje bazom podataka je sistem opšte namjene za rješavanje zajedničkih i aplikativnih problema i zadataka u radu sa bazom.
MySQL	<i>My Structured Query Language</i> – je DBMS otvorenog koda kompanije MySQL AB dostupan za različite operativne sisteme (Windows, Linux, MacOS i druge).
Windows	Operativni sistem kompanije Microsoft.
Java	Objektno-orijentisani jezik opšte namjene razvijen 1995. godine od strane Džejmsa Goslinga iz kompanije Sun Microsystems.
JVM	<i>Java Virtual Machine</i> – je virtuelna mašina koja omogućava da računar, na kojem je instalirana, pokreće program napisan u Java programskom jeziku.
TCP/IP	Transmission Control Protocol/Internet Protocol – Grupa protokola koja se koristi za komunikaciju na globalnoj računarskoj mreži – Internetu.
Hash	Heš funkcija je funkcija koja se koristi za mapiranje podataka proizvoljne veličine u podatke fiksne dužine, na način da se iz izlaza funkcije ne može dobiti (ili vrlo teško) ulaz. Vrijednost koju ovakva funkcija vraća se naziva <i>hash value</i> ili samo <i>hash</i> .

1.4 Referentni dokumenti

- [1] Lekić A., Suvajac B., Crnovčić D., Bunić D., Milinčić Lj., Vuković S. *Specifikacija korisničkih zahtjeva - Hotel Customer Care*

1.5 Pregled

Dokument je podijeljen na tri glave, a to su: **Uvod**, **Arhitektura postojećeg sistema** i **Predložena arhitektura**.

U prvom dijelu dokumenta navodi se namjena sistema i projektni ciljevi kao i spisak skraćenica, definicija i referenci.

Drugi dio ovog dokumenta čini opis postojeće arhitekture sistema.

Treću glavu dokumenta čine kratak pregled arhitekture i funkcionalnosti podsistema, dekompozicija sistema, HW/SW mapiranje i perzistentni sloj. Treći dio takođe obuhvata kontrolu prava pristupa i sigurnost te kontrolu toka i granična stanja sistema.

2 Arhitektura postojećeg sistema

HCC sistem će biti razvijen sa pretpostavkom da ne postoji relevantna arhitektura postojećeg sistema u smislu informatičkog načina pružanja usluga. Od hotela se očekuje samo da ima lokalnu *wireless* mrežu, tj. pristup internetu i da obezbijedi personalni računar za svaku gostinsku sobu kao i radnika na recepciji. Sistem je **greenfield investicija** i nije moguća implementacija sistema pomoću reinženjeringa.

Greenfield investicija je vrsta investicije za koju je značajno da se sa poslom počinje posve od početka, bez prethodne infrastrukture, poslovnih prostora i radnika.

Reinženjering predstavlja radikalan redizajn poslovnih procesa radi njihovog dramatičnog poboljšanja. Radikalni redizajn znači otpočinjanje od početka umjesto mijenjanja ili modifikovanja postojećih načina rada.

3 Predložena arhitektura

3.1 Kratak pregled aritekture i funkcionalnosti podsistema

Odabrana arhitektura za implementaciju HCC sistema je **klijent-server** u kombinaciji sa **MVC** arhitekturnim stilom. Navedeni arhitekturni stil najbolje odgovara potrebama ciljne aplikacije.

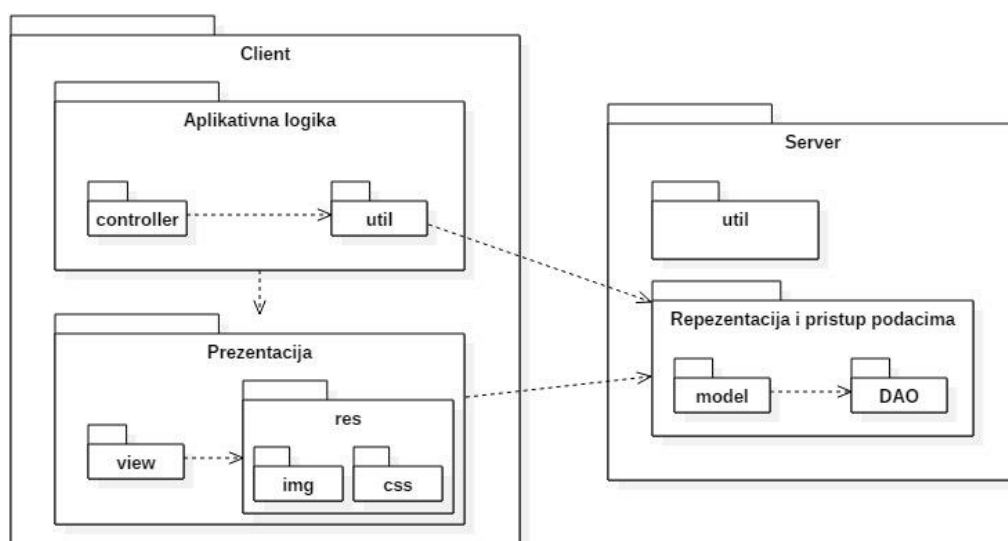
Klijent-server je arhitektura gdje su korisnik tj. klijent i server odvojeni ili neravnopravni. Klijent je najčešće aktivan korisnik, koji šalje zahtjeve i čeka dok se zahtjevi ne ispune. Server je pasivan, on čeka na zahtjeve te ih ispunjava i šalje klijentu. Serveri su obično veoma jaki računari sa moćnim konfiguracijama i karakteristikama zbog toga što istovremeno moraju prerađivati mnogo zahtjeva koji pristižu veoma često.

MVC (*Model-View-Controller*) arhitektura je softverski šablon koji odvajava prikaz informacija od interakcije korisnika sa tim informacijama. Model se sastoji od podataka aplikacije, poslovnih pravila, logike i funkcija. View može da bude bilo koji izlazni prikaz podataka, kao što je dijagram, grafik ili *GUI* interfejs. *Controller* uzima ulazne podatke i prilagođava ih odgovarajućim modelima ili view-ovima.

3.2 Dekompozicija sistema

Kako je već navedeno u prethodnom naslovu, dekompozicija sistema je izvršena prema MVC arhitekturnom stilu na podsisteme zadužene za aplikativnu logiku, prezentaciju i reprezentaciju i pristup podacima.

Aplikativna logika i prezentacija se nalaze na klijentskoj strani dok je reprezentacija i pristup podacima smješten na serverskoj strani. Aplikativna logika se sastoji iz *Controller*-a i *Util*-a koji su zaduženi za manipulaciju podacima i interfejsu prema pomoćnim klasama respektivno. Prezentacija se sastoji iz *View*-a koji će biti zadužen za prikaz informacija i podataka korisniku sistema. U podsistemu koji je zadužen za prezentaciju osim *View*-a nalaze se i *Img* i *CSS* paketi u kojima se nalaze slike potrebne za funkcionisanje aplikacije i fajlovi zaduženi za formatiranje izgleda elemenata *View*-a. Reprezentacija i pristup podacima se sastoji iz *Model*-a koji predstavlja perzistentne objekte sistema kao i transfer i pristup istih u bazu podataka. Na sljedećoj slici je dat dijagram paketa ciljnog sistema:



Slika 2: Dijagram paketa ciljnog sistema

3.3 HW/SW mapiranje

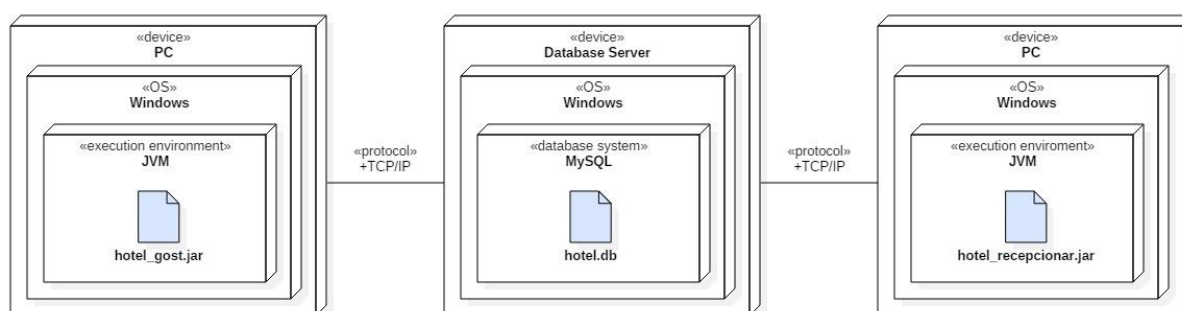
Kao što je bitno izvršiti dekompoziciju sistema na podsisteme, bitno je i prikazati koje veze postoje između tih podsistema, kao i kako se oni realizuju: hardverski ili softverski. HW/SW (*hardware/software*) mapiranje omogućava upravo ovo.

UML-om se HW/SW mapiranje predstavlja sa dva dijagrama:

- Dijagram razmještaja (eng. *deployment diagram*).
- Dijagram komponenata (eng. *component diagram*),

Dijagram razmještaja je strukturni dijagram kojim modelujemo raspored komponenata u eksploataciji sistema. Sastoji se iz čvorova, koji predstavljaju apstrakcije hardverskih ili softverskih komponenata i veza između njih.

Na slici 2 je dat dijagram razmještaja za ciljni sistem. Imamo tri glavna čvora (stereotip <<device>>) koji predstavljaju računar gosta, recepcionara i server sa bazom podataka. Na računarima gosta i recepcionara će biti instalirani *Windows* operativni sistemi (stereotip <<OS>>) na kojim će se pokretati klijentske aplikacije gosta odnosno recepcionara. Kako se radi o Java aplikacijama, one će se izvršavati na Java virtuelnim mašinama (stereotip <<execution environment>>). Server sa bazom podataka će sadržati *MySQL* sistem za upravljanje bazom podataka (stereotip <<database system>>) koji će se izvršavati takođe na *Windows* operativnom sistemu (stereotip <<OS>>). Veza između klijentskih računara i servera je predstavljena punom linijom između čvorova sa oznakom *TCP/IP*, da se označi da se komunikacija odvija preko mreže.



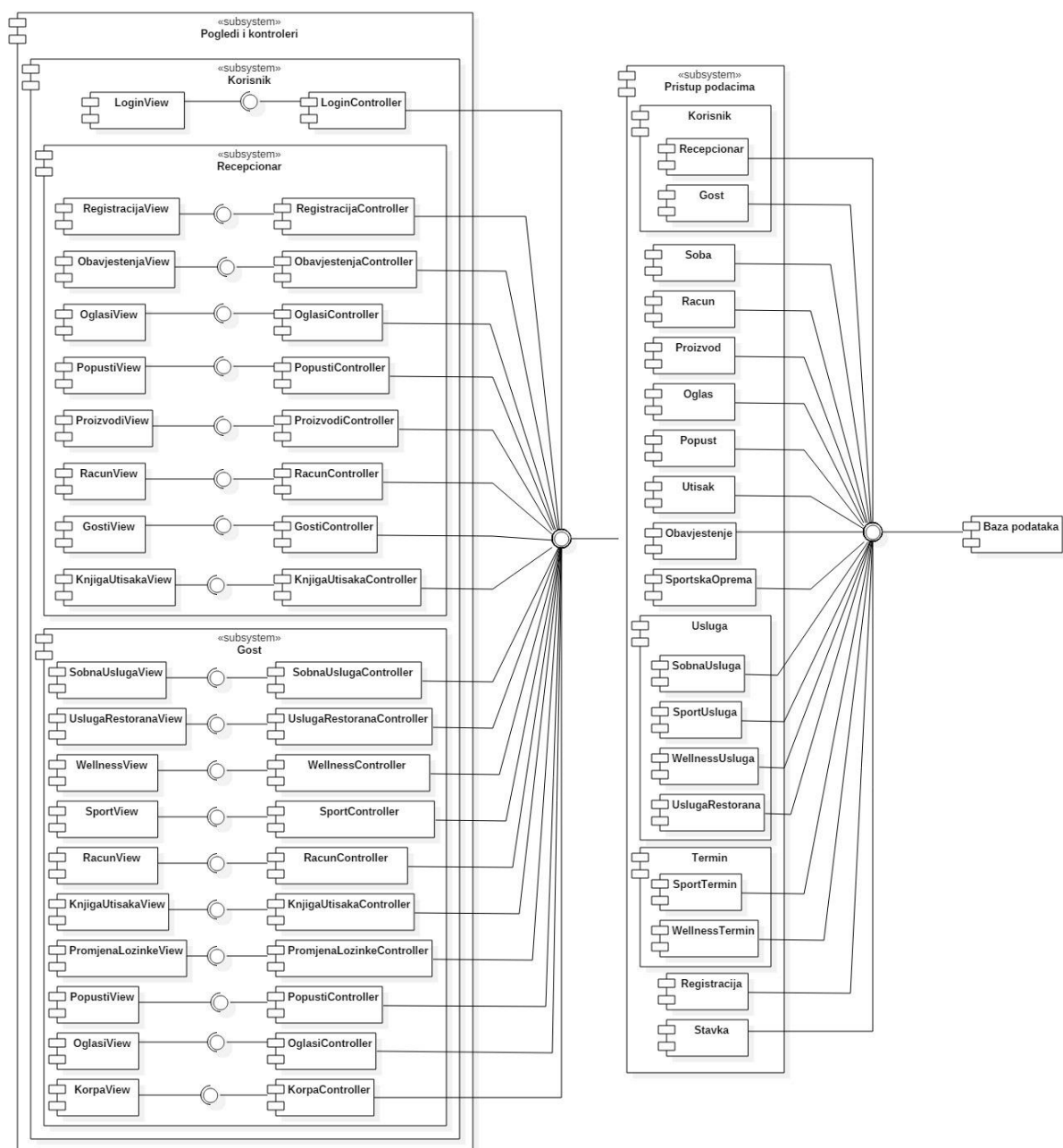
Slika 1: Dijagram razmještaja ciljnog sistema

Na slici 3 je prikazan dijagram komponenata za HCC sistem. Tri glavna podsistema su: Pogledi i kontroleri, Pristup podacima i Baza podataka.

Podsistem Pogledi i kontroleri se dalje dekomponuje na komponente Recepcionar i Gost koji sadrže *View* i *Controller* komponente pomoću kojih se realizuje pristup, obrada i prikaz perzistentih objekata koji se čuvaju u bazi podataka. Veza između *View*-a i odgovarajućeg *Controller*-a je prikazana korištenjem „ball-and-socket“ notacijom da se naglasi da *Controller* komponente predstavljaju *provided* interfejs prema *View* komponentama, tj. da pružaju servis *View* komponentama.

Podsistem Pristup podacima se sastoji iz komponenata sa *design time* životnim vijekom i predstavljaju ekvivalent klasama koje su navedene u dijagramu klasa (pogledati dokument pod [1]).

Podsistem Baza podataka, kao i što samo ime govori, predstavlja podsistem koji sadrži bazu podataka u kojoj će se čuvati svi relevantni podaci za ciljni sistem. Predstavlja *provided* interfejs prema podsistemu Pristup podacima, odnosno njegovim podkomponentama.



Slika 3: Dijagram komponenti ciljnog sistema

3.4 Perzistentni sloj

Niti jedan ozbiljan softverski sistem ne funkcioniše bez dobro obrađenog perzistentnog sloja. Perzistentni sloj upravlja perzistentnim objektima, odnosno, objektima čije stanje mora da se trajno čuva na fajl sistemu ili u bazi podataka.

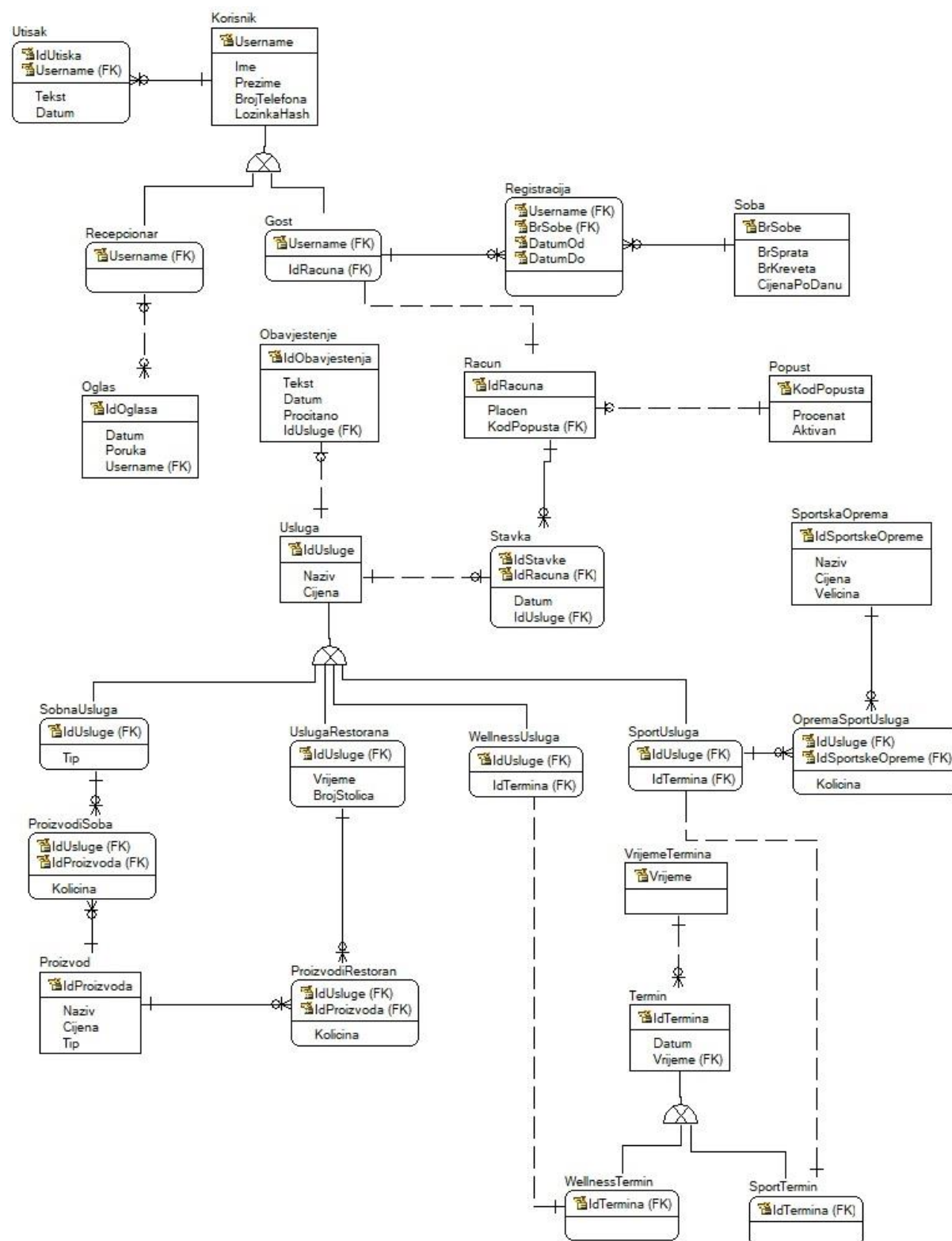
HCC treba da čuva podatke o korisnicima, odnosno gostima i recepcionarima, za koje je potrebno voditi informacije neophodne za funkcionisanje sistema. Osim ovih podataka neophodno je čuvati podatke o sobama, podatke o korištenim uslugama, podatke od stanju računa, podatke o popustima koje gost može da ostvari, podatke o oglasima te podatke vezane za knjigu utisaka.

Pristup ovim podacima treba da bude konkurentan jer je HCC zamišljen kao klijent-server aplikacija koja opslužuje veći broj klijenata. Konkurentnim pristupom moguće je da klijenti istovremeno pristupaju određenim podacima, a HCC treba da obezbijedi konzistentnost i integritet kako goste hotela ne bi doveli u neprijatnu situaciju.

Podaci će se čuvati na serverskom dijelu aplikacije. Klijenti pristupaju podacima tako što pošalju odgovarajući zahtjev serveru, dok server, zapravo, vrši direktan pristup odgovarajućim podacima te ih dostavlja klijentu koji je uputio zahtjev. Na taj način obezbjeđena je transparentnost podataka, odnosno, korisnik aplikacije ima percepciju centralizovane organizacije podataka.

Zbog navedenih razloga HCC će manipulirati perzistentnim objektima putem *DBMS*-a. *DBMS* obezbjeđuje mehanizme za konkurentan pristup podacima i uz ispravno korištenje na jednostavan način obezbjeđuje konzistentnost i integritet podataka. Za potrebe funkcionisanja sistema odabrana baza podataka biće relacionala, a za konkretni *DBMS* sistem biće odabran *MySQL*.

Na sljedećoj stranici je prikazan *IE (Information Engineering)* dijagram baze podataka dobijen na osnovu dijagrama klasa ciljnog sistema.



Slika 4: Relaciona šema baze podataka ciljnog sistema

3.5 Kontrola prava pristupa i sigurnost

Kao što se tokom faze analize softvera različiti učesnici vežu za različite slučajeve upotrebe, čime se ograničava njihovo učešće u sistemu, tako se prilikom faze dizajna mora razviti mehanizam za ograničavanje prava pristupa podacima učesnicima sistema.

Za ograničavanje pristupa operacijama unutar klasa iskorištena je **lista mogućnosti** (eng. *capability list*). Lista mogućnosti omogućava laku provjeru da li učesnik ima pravo pristupa nekoj operaciji tako što se provjerava da li pripadajuća lista sadrži odgovarajući par <klasa, operacija>. Ako ne sadrži onda dati učesnik nema pravo pristupa operaciji koja se nalazi unutar korespondentne kalse.

U nastavku su date liste prava pristupa za učenike „Gost“ i „Recepcionar“.

Tabela 3: Lista mogućnosti za učesnika „Gost“

Gost	
Korisnik	login()
Korisnik	logout()
Gost	promjenaLozinke()
Oglas	prikazi()
Popust	unesiKod ()
KnjigaUtisaka	dodajUtisak()
KnjigaUtisaka	prikazi()
Racun	prikazi()
SportUsluga	prikaziTermine()
SportUsluga	prikaziOpremu ()
UslugaRestorana	prikaziMeni()
SobnaUsluga	prikaziMeni()
SobnaUsluga	pranjeVesa()
SobnaUsluga	pospremanjeSobe()
WellnessUsluga	prikaziTermine ()
Usluga	rezervisi()

Tabela 2: Lista mogućnosti za učesnika „Recepcionar“

Recepcionar	
Korisnik	login()
Korisnik	logout()
Gost	prikaziRacun()
Racun	plati()
Popust	dodaj ()
Popust	obrisi ()
Registracija	registrujGosta ()
KnjigaUtisaka	prikazi()
KnjigaUtisaka	obrisiUtisak()
Obavjestenje	prikazi()
Proizvod	dodaj()
Proizvod	obrisi()
Oglas	dodaj()
Oglas	obrisi()

Dodatna zaštita sistema će biti osigurana autentikacijom korisnika koji žele da koriste aplikaciju. Autentikacija će se sastojati iz unosa korisničkog imena i lozinke. Treba napomenuti da se lozinke u bazi podataka neće čuvati u otvorenom obliku već će biti čuvan samo *heš* (eng. *hash*) lozinki.

3.6 Kontrola toka

Kontrola toka u okviru HCC sistema je dizajnirana na centralizovani način pri čemu se koristi mehanizam zasnovan na upravljanju događajima (eng. *event-driven*).

U centralizovanom dizajnu postoji jedinstvena softverska komponenta koja može da prihvati kod, a sve druge komponente sinhronišu svoj rad prema njoj. Dakle, kontrolu toka je prepuštena radu jednog upravljačkog objekata.

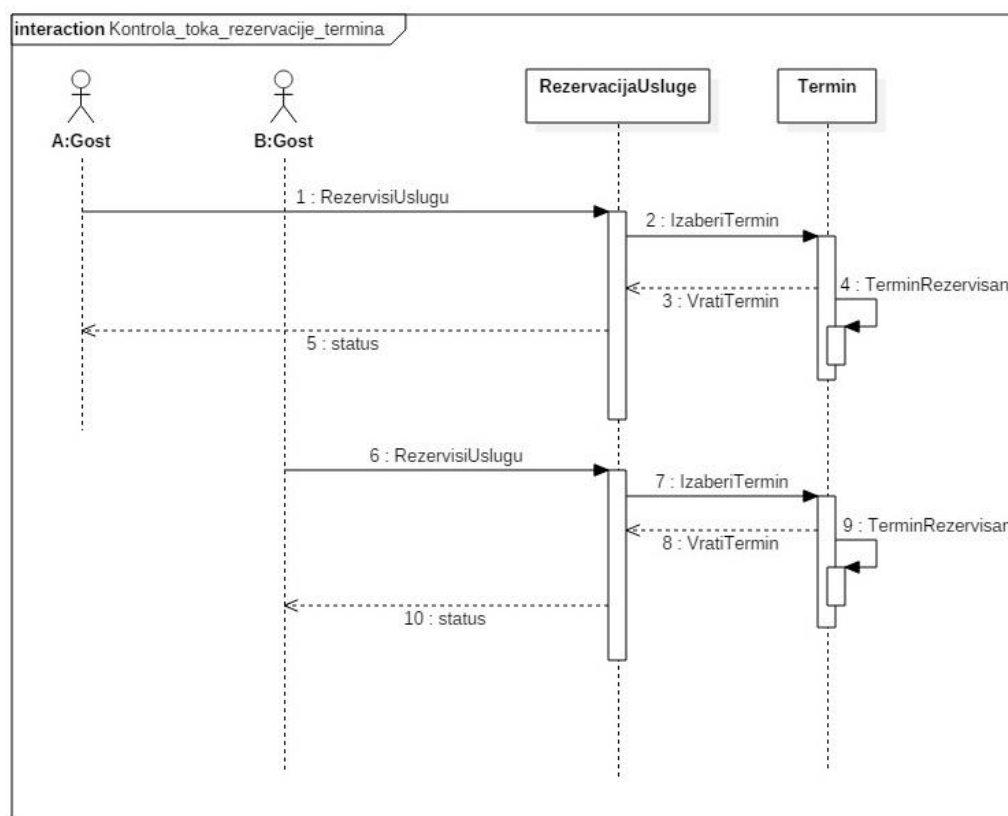
U programiranju koje je vođeno događajima tok programa je određen aktivnostima koje se obavljaju kao odgovori na neku akciju iniciranu od strane korisnika, poput klika mišem ili pritiskanja tastera, ili aktivnostima koje su zasnovane na porukama iz drugih programa, senzorima, alarmima itd.

Ovakva kontrola toka se dominantno upotrebljava kod aplikacija sa grafičkim korisničkim interfejsima kao što je slučaj sa HCC sistemom.

U aplikaciji vođenoj događajima, uobičajeno postoji jedna glavna, upravljačka petlja koja osluškuje događaje, a zatim kad se detektuje neki događaj reaguje na njega tako što dispečer aktivira povratni poziv funkcije, odnosno šalje odgovarajuće zahtjeve i poruke.

Da bi se održalo konzistentno stanje sistema u slučaju konkurentnog izvršavanja određenih operacija u sistemu potrebno je obezbediti da se transakcije realizuju u izolaciji, odnosno da druga transakcija ide tek kada se prva transakcija kompletira.

U okviru HCC sistema konkurentna aktivnost se dešava u slučaju da više korisnika pokuša da rezerviše isti termin u okviru jedne usluge. Tada je potrebno serijski i izolovano izvršavati navedene transakcije što se ogleda u tome da će kontroler *rezervacijaUsluga* dozvoliti pristup terminima samo jednom od gostiju. Izvršiće se ažuriranje termina rezervacije, kontroler će poslati odgovarajući status o usluzi gostu hotela, i tek poslije toga dozvoliti pristup rezervaciji usluga drugom gostu. To je ilustrovano na sljedećem dijagramu:



Slika 5: Dijagram sekvence za kontrolu toka rezervacije termina

3.7 Granična stanja sistema

HCC sadrži sljedeće vrste graničnih stanja:

- **Početak rada sistema** - sistem prikazuje formu za prijavu na sistem odakle se prelazi u stanje opsluživanja korisnika. Korisnički interfejs nudi opcije korisnicima sa nalogom da uđu u sistem, promijene lozinku ili obavještenje ako korisničko ime i/ili lozinka nisu tačni.
- **Terminacija rada sistema** - sistem oslobađa sve zauzete resurse kao što su mrežna konekcija i veza sa bazom podataka. Pri tome su podsistemi obaviješteni o terminaciji rada sistema.
- **Otkazi** - obuhvataju greške, *bug*-ove i nepredvidive pojave kao što su nestanak struje, hardverski otkazi ili eventualno pad čitavog sistema. HCC je obezbijeđen od ovoga obradom raznih vrsta izuzetaka kako bi se unaprijed predvidjele neželjene situacije i zamjenom odgovarajućih dijelova nefunkcionalne opreme u što kraćem vremenskom intervalu. Za pravilno funkcionisanje sistema odgovoran je administrator. U cilju što boljeg funkcionisanja sistema on obavlja sljedeće aktivnosti:
 - Analiza sistema i identifikacija potencijalnih problema u sistemu.
 - Izrada rezervnih kopija podataka i sistema.
 - Inoviranje i nadogradnja sistema novim tehnologijama.
 - Instalacija novih verzija, parametrizacija i prekonfiguracija komponenti sistema.
 - Instalacija i konfiguracija novog softvera i hardvera.
 - Dodavanje, oduzimanje i ažuriranje korisničkih naloga u sistemu.
 - Obezbjeđivanje zaštite sistema.
 - Izrada dokumentacije o računarskom sistemu.
 - Rješavanje zastoja u sistemu i izvještavanje.
 - Poboljšavanje performansi sistema.