DIZAJN SISTEMA

Lanac autobuskih stanica

Verzija: 1.0
Tim: Vanja Ćulum, Anela Crljenković, Marko Knežić, Milan Paspalj
Datum: 20/05/2018

SADRŽAJ

1. UVOD

1.1 Namjena sistema

Sistem **BUST** ima za cilj da omogući lakšu organizaciju autobuskog saobraćaja na nivou autobuskih stanica, međusobnu komunikaciju i pojedinačnu organizaciju svake od njih. Ovaj sistem je namjenjen kako autobuskim stanicama, tako i svim radnicima u okviru autobuskih stanica. Omogućavati će radnicima jednostavno upravljane svim aktivnostima u okviru autobuske stanice, kao i pružanje usluga korisnicima koji koriste usluge autobuske stanice. U svrhu ovoga, sistem će morati da ima niz funkcija kojima će zadovoljiti zahtjeve koji se pred njega postave. Drugim riječima, omugućiti će potpunu automatizaciju rada autobuske stanice. Sistem će biti realizovan kao desktop aplikacija čiji je glavni cilj da omogući što jednostavniju komunikaciju između radnika i korisnika usluga autobuske stanice. Ova aplikacija je pogodna za sve autobuske stanice koje žele da usavrše organizaciju i upravljanje samom stanicom i svim njenim jedinicama, kao i da pojednostave pružanje usluga korisnicima.

1.2 Projektni ciljevi

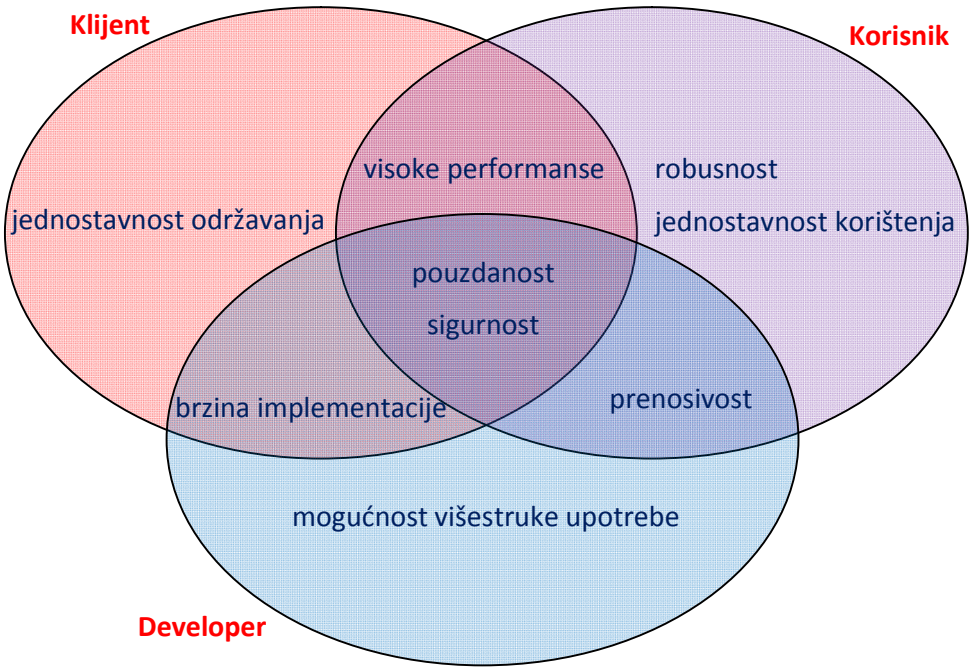
BUST je softverska aplikacija koja omogućava veliki broj funkcionalnosti za interakciju radnika sa sistemom pomoću grafičkog korisničkog interfejsa (***GUI***).

Dizajniran je da tako da ispuni sledeće projektne ciljeve:

- Pouzdanost - podrazumjeva da je softver u mogućnosti da izvrši traženu funkciju pod određenim uslovima za određeni period vremena,
- Sigurnost - odnosi se na zaštitu podataka i informacija namjenjenih samo određenim osobama, a biće omogućena upotrebom korisičkih naloga pri čemu će određene funkcije biti dostupne samo specijalizovanim korisnicima, kao i korištenjem ***DBMS***, te ***backup***-om svih podataka i informacija kojima sistem raspolaže,
- Brzina implementacije - cilj je implementirati aplikaciju sa svim potrebnim funkcionalnostima u što kraćem vremenskom periodu,
- Jednostavnost korištenja - odnosi se na upravljanje sistemom i svim njegovim funkcijama na što jednostavniji način, a biće omogućena dizajniranjem jednostavnog grafičkog korisničkog interfejsa (*GUI*),
- Robusnost - mogućnost sistema da radi pri velikom opterećenju ili nevalidnim situacijama (otpornost na greške), a biće omogućena predviđanjem i implementacijom tehnika za razrješavanje nastalih izuzetaka,
- Mogućnost višestruke upotrebe - predstavlja mogućnost da se dio i cijeli programski kod može koristiti u drugim projektima bez ikakvih ili sa

minimalnim modifikacijama, a biće omogućena korištenjem projektnih obrazaca,

- Visoke performanse - softver mora izvršava sve funkcionalnosti u vremenskom intervalu prihvatljivom korisniku, pri čemu ne zahtjeva previše resursa,
- Prenosivost - mogućnost korištenja sistema i svih njegovih funkcionalnosti pod različitim uslovima i na različitim okruženjima, a biće omogućena implementacijom sistema korištenjem ***Java*** programskog jezika pod pretpostavkom da na okruženju na kome se aplikacija izvršava nalazi ***Java Runtime Environment***,
- Jednostavnost održavanja - mjera koliko je lako ispraviti propuštene greške i nadogradnju novih funkcija sistema, a biće omogućena korištenjem projektnih obrazaca.



Slika 1: projektni ciljevi po kategorijama stakeholder-a

1.3 Definicije i skraćenice

Pojam	Opis
DBMS	*Database Management System* odnosno sistem za upravljanje bazom podataka. Softver namjenjen za upravljanje bazom podataka.
GUI	*Graphical User Interface* odnosno grafički korisnički interfejs. Omogućava interakciju između korisnika i aplikacije preko grafičkih komponenti.
Java	Objekto-orijentisani programski jezik opšte namjene.
Java Runtime Environment	Softver koji sadrži sve potrebne komponente da bi se izvršavao Java program.
Projektni obrazac	Opšte, ponovno upotrebljivo rješenje za česte probleme koji se sreću prilikom projektovanja softvera.
TCP/IP	Transmission Control Protocol/Internet Protocol. Grupa protokola koja se koristi za komunikaciju na gobalnoj računarskoj mreži - Internetu.
Klijent	Osoba koja je naručilac sistema.
Korisnik	Osoba koja koristi usluge sistema.
Developer	Osoba koja projektuje sistem.
Greenfield projekat	Predstavlja projekat čije izvođenje ne prati nikakav prijašnji posao, tj. izgradnja se vrši od nule.
UML	Unified Modeling Language. Modelacioni jezik opšte namjene za prikazivanje vizuelnog dizajna sistema u softverskom inženjerstvu.

Tabela 1:definicije i skraćenice

1.4 Referentni dokumenti

[1] Specifikacija korisničkih zahtjeva – Lanac autobuskih stanica

1.5 Kratak pregled dokumenata i predložene arhitekture

Dokument je podjeljen na tri glave, Uvod, Arhitektura postojećeg sistema i Predložena arhitektura.

U prvom dijelu dokumenta definisana je namjena sistema kao i projektni ciljevi koji se teže ispuniti. Takođe, sadrži spisak definicija, skraćenica i referenci.

Drugi dio dokumenta čini opis postojeće arhitekture sistema.

Treći dio čine kratak pregled arhitekture i funkcionalnosti podsistema, dekompozicija sistema, HW/SW mapiranje, perzistentni sloj, kontorla prava pristupa i sigurnost, kontrola toka, te granična stanja sistema.

2. ARHITEKTURA POSTOJEĆEG SISTEMA

BUST sistem se gradi od nule, tj. predstavlja ***greenfield*** projekat. Zbog ovoga neće biti moguće izvršiti reinženjering pa se pažljivo mora odabrati arhitekurni stil.

Arhitekturni stilovi koji se obično koriste su:

- Klijent-Server arhitekturni stil
- Peer-to-Peer arhitekturni stil
- MVC (Model-View-Controller) arhitekturni stil
- Troslojni arhitekturni stil
- Četvoroslojni arhitekturni stil

Sistem **BUST** će se oslanjati na MVC arhitekturni stil, a motivacija za izbor ovog arhitekturng stila je:

- interfejs sistema mijenja se mnogo češće nego aplikativna logika
- aplikativna logika mijenja se mnogo češće nego domenski objekti

3. PREDLOŽENA ARHITEKTURA

3.1 Kratak pregled arhitekture i funkcionalnosti podsistema

MVC (Model-View-Controller) je arhitekturni stil koji razdvaja podsisteme u tri kategorije:

- Model – reprezentacija i manipulacija podacima (domenskim objektima)
- View – prezentacija podataka krajnjem korisniku
- Controller – aplikativna logika zadužena za upravljanje svim funkcionalnostima sistema te prikazivanje rezultata krajnjem korisniku

Kontroler mjenja model a svaka promjena treba da se manifestuje prikazom. Izborom ovog stila omogućeni su nezavisan razvoj, testiranje i održavanje sistema.

Slika 2: MVC arhitekturni stil

Identifikovani podsistemi su navedeni i opisani u daljem tekstu.

3.2 Dekompozicija sistema

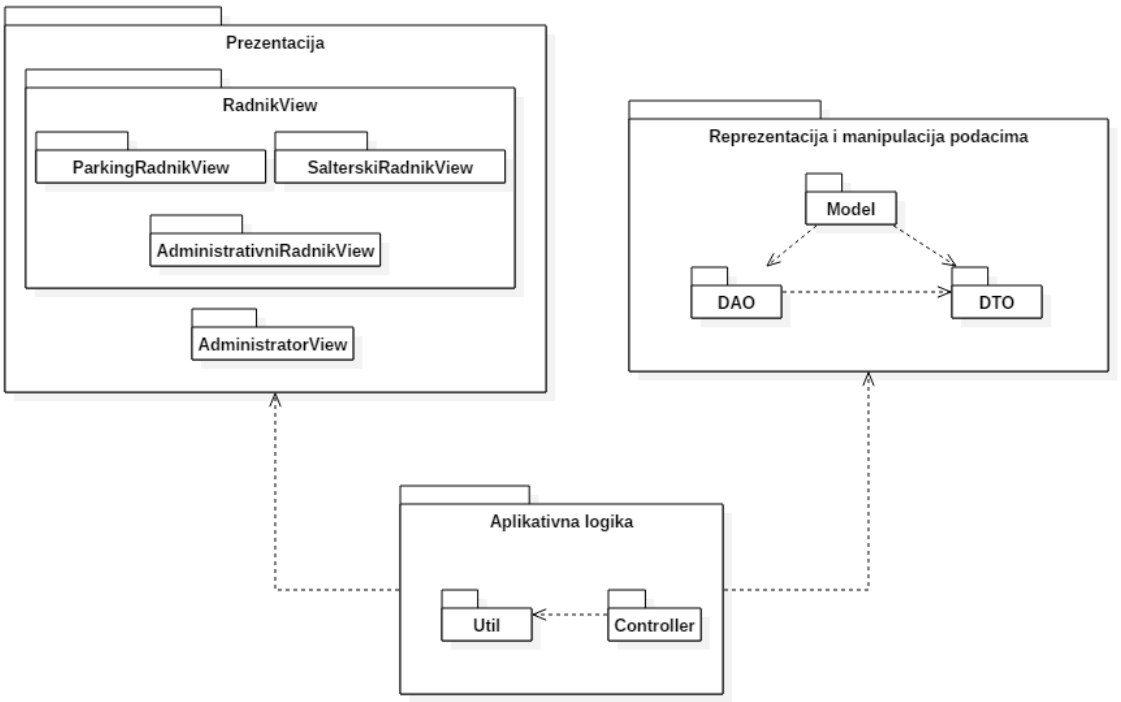
Podsistem predstavlja kolekciju blisko povezanih klasa, asocijacija, operacija, događaja i ograničenja. Dekompozicija sistema je pojednostavljivanje odnosno „razbijanje“ sistema na podsisteme u cilju rješavanja više jednostavnijih problema.

Dekompozicija sistema **BUST** je zasnovana na MVC arhitekturnom stilu a razbijanje je izvršeno na podsisteme zadužene za aplikativnu logiku, prezentaciju te reprezentaciju i manipulaciju podacima.

Podsistem prezentacije služi za prikaz podataka krajnjem korisniku. Podjeljen je na dva podsistema: RadnikView i AdministratorView.

Podsistem aplikativne logike služi za interakciju sa korisnikom preko podsistema Controller i pomoćnih klasa podistema Util.

Podsistem reprezentacije i pristupa podacima obezbjeđuje interakciju sa DBMS preko odgovarajućih podsistema za transfer: DAO i DTO.



Slika 3: Zamišljena MVC arhitektura

3.3 HW/SW mapiranje

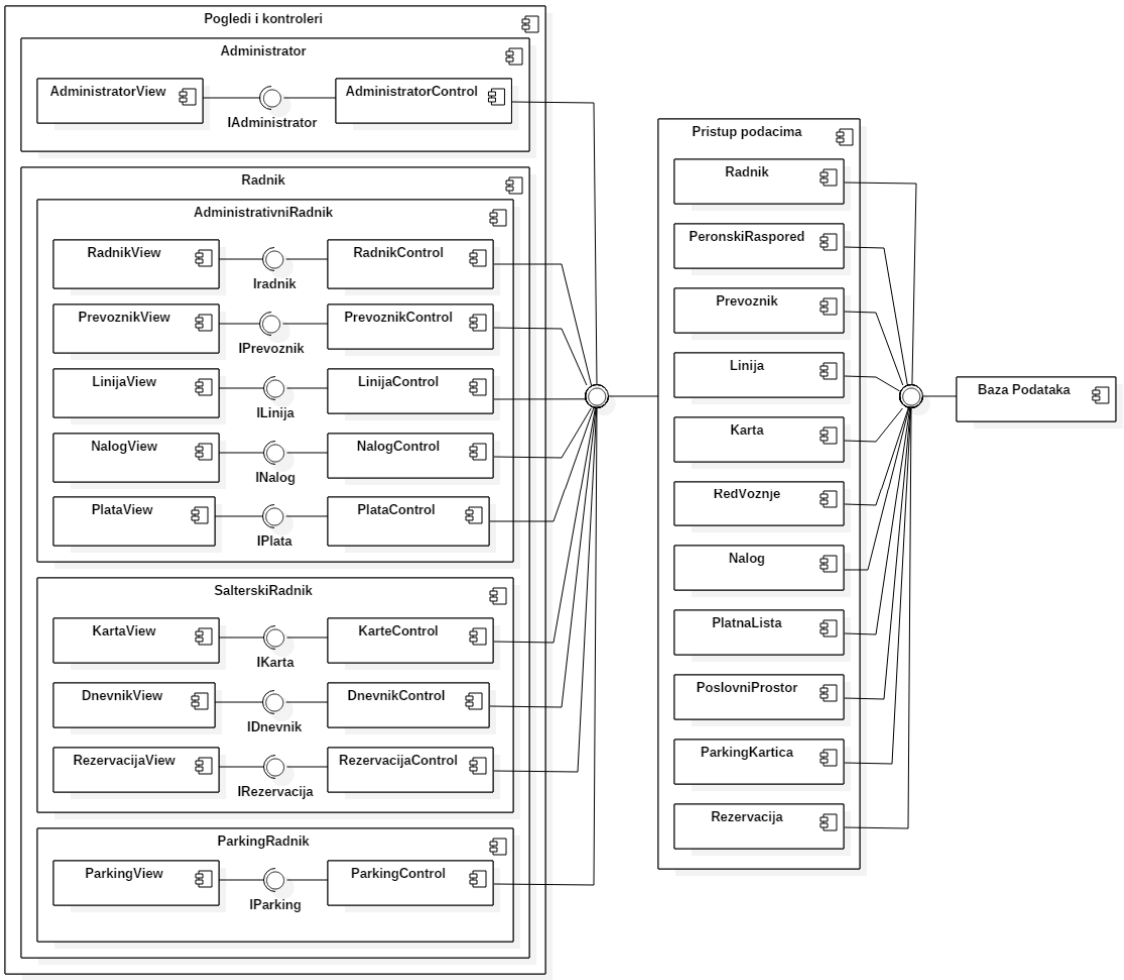
Nakon što se izvrši dekompozicija sistema na podsisteme, potrebno je prikazati koje veze postoje između tih podsistema te kako se oni realizuju: hardverski ili softverski. U tu svrhu se vrši HW/SW mapiranje.

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

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

Komponenta je zamjenljivi dio sistema koji realizuje skup interfejsa. Dijagram komponenata prikazuje organizaciju i zavisnosti između komponenata.

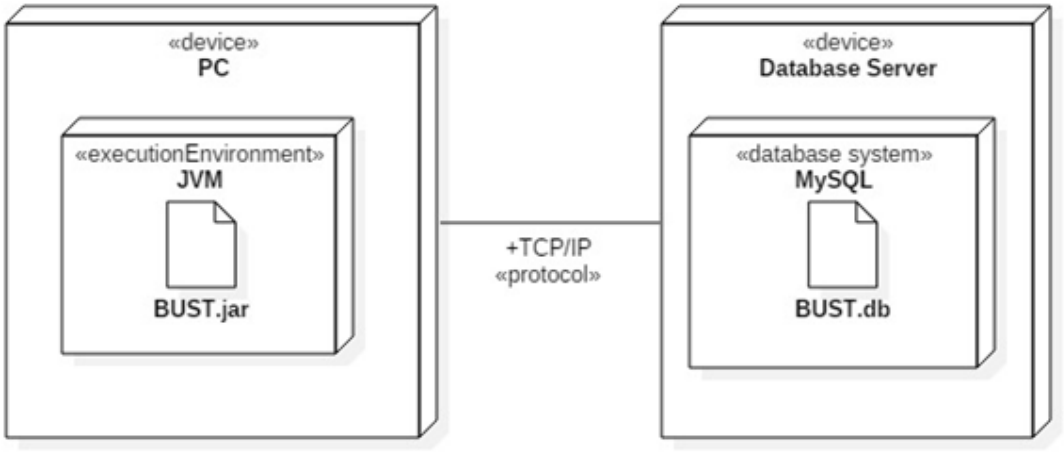
Komponente sistema su podjeljene na tri podsistema: pogledi i kontroleri, pristup podacima i bazu podataka.



Slika 4: Dijagram komponenata

Dijagram razmještaja predstavlja fizički razmještaj artifakta na čvorovima.

Aplikacija je instalirana na računarima radnika, koji moraju imati instaliranu Java virtuelnu mašinu (JVM). Svi podaci neophodni za rad nalaze se na globalnoj bazi podataka.



Slika 5: Dijagram razmještaja

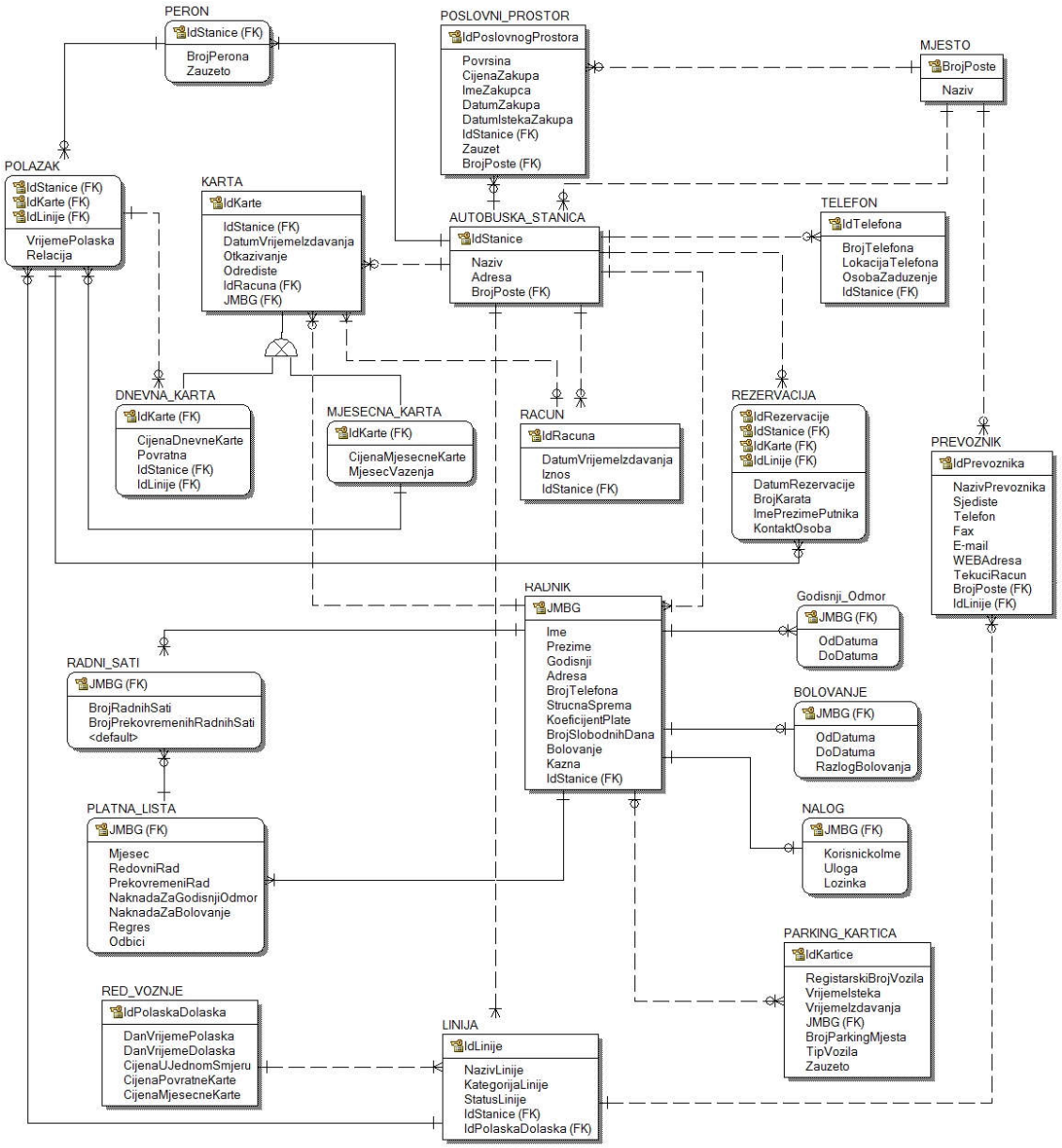
3.4 Perzistentni sloj

Perzistentni sloj upravlja perzistentnim objektima, odnosno objektima čije stanje mora trajno da se čuva i nakon završetka aplikacije.

Ovaj sistem treba trajno da čuva evidenciju o autobuskim stanicama, poslovnim prostorima autobuskih stanica, zaposlenim radnicima, prodanim, rezervisanim, kao i otkazanim kartama, a takođe i red vožnje, linijama i prevoznicima koji saobraćaju tim linijama te računima izdatim za naplatu karata. Trajno čuvanje ovih podataka je od izuzetne važnosti za normalno funkcionisanje autobuskih stanica.

Zbog navedenih razloga **BUST** će manipulisati perzistentnim objektima pomoću *DBMS*-a. *DBMS* obezbjeđuje mehanizme za konkurentan pristup podacima i uz ispravno korištenje na jednostavan način obezbjeđuje konzistentnost i integritet podataka. Za trajno čuvanje podataka koristiće se relaciona baza podataka a *MySQL* će se koristiti kao *DBMS*.

Pomoću ER (Entity Relationship) dijagrama prikazana je struktura baze podataka dobijena na osnovu dijagrama klasa ciljnog sistema.



Slika 6: Relaciona šema baze podataka

3.5 Kontrola prava pristupa i sigurnost

S obzirom da nemaju svi korisnici sistema ista prava, da se ne bi narušila sigurnost, potrebno je razviti mehanizam za ograničavanje prava pristupa podacima.

Za ograničavanje prava pristupa operacijama unutar klasa iskorištena je lista mogućnosti. Lista mogućnosti omogućava jednostavnu provjeru da li određeni korisnik ima pravo pristupa nekoj operaciji tako što se provjerava da li lista mogućnosti sadrži odgovarajući par *<klasa, operacija>*. Ako ne sadrži onda dati korisnik nema pravo pristupa operaciji koja se nalazi unutar posmatrane klase.

Prava pristupa korisnika su prikazana na sledećim listama mogućnosti:

Administrator	
Nalog	prijava()
Nalog	odjava()
Naog	promjenaLozinke()
Radnik	naBolovanju()
Radnik	naGodisnjem()
Radnik	brojSlobdnihDana()
Radnik	koeficijentPlate()
PlatnaLista	kreiraj()
PlatnaLista	prikazi()
PlatnaLista	stampaj()
RadniSati	unesi()
Linija	getStatus()
RedVoznje	cijenaVoznje()
RedVoznje	trajanjeVoznje()
Karta	popunjavanjeKarte()
Karta	stampanjeKarte()
Karta	prodaja()
Karta	vracanje()
Karta	otkazivanje()
DnevnaKarta	formiranjeUkupneCjene()
MjesecnaKarta	formiranjeCijene()
Racun	stampanjeRacuna()
Rezervacija	popunjavanje()
Rezervacija	otkazivanje()
Peron	jeZauzet()
ParkingKartica	odgovarajucaParkingMjesta()
ParkingKartica	istekla()
ParkingKartica	unosPodataka()
ParkingKartica	stampaj()

Tabela 2: Lista mogućnosti za administratora

AdministrativniRadnik	
Nalog	prijava()
Nalog	odjava()
Nalog	promjenaLozinke()
Radnik	naBolovanju()
Radnik	naGodisnjem()
Radnik	brojSlobdnihDana()

Radnik	koeficijentPlate()
PlatnaLista	kreiraj()
PlatnaLista	prikazi()
PlatnaLista	stampaj()
Linija	getStatus()
RedVoznje	cijenaVoznje()
RedVoznje	trajanjeVoznje()
Peron	jeZauzet()

Tabela 3: Lista mogucnosti za administrativnog radnika

SalterskiRadnik	
Nalog	prijava()
Nalog	odjava()
Nalog	promjenaLozinke()
Linija	getStatus()
RedVoznje	cijenaVoznje()
RedVoznje	trajanjeVoznje()
Karta	popunjavanjeKarte()
Karta	stampanjeKarte()
Karta	prodaja()
Karta	vracanje()
Karta	otkazivanje()
DnevnaKarta	formiranjeUkupneCjene()
MjesecnaKarta	formiranjeCijene()
Racun	stampanjeRacuna()
Rezervacija	popunjavanje()
Rezervacija	otkazivanje()

Tabela 4: Lista mogućnosti za šalterskog radnika

ParkingRadnik	
Nalog	prijava()
Nalog	odjava()
Nalog	promjenaLozinke()
ParkingMjesto	jeZauzeto()
ParkingMjesto	getTipVozila()
ParkingKartica	odgovarajucaParkingMjesta()
ParkingKartica	istekla()
ParkingKartica	unosPodataka()
ParkingKartica	stampaj()

Tabela 5: Lista mogućnosti za parking radnika

Zbog dodatnog povećanja sigurnosti sistema, korisnici će morati izvršiti prijavu na sistem kako bi korstili sve funkcionalnosti koje su za njih definisane ranije. Autentikacija korisnika će se satojati iz unosa korisničkog imena i lozinke. Lozinke će se čuvati kao heš vrijednosti u bazi podataka.

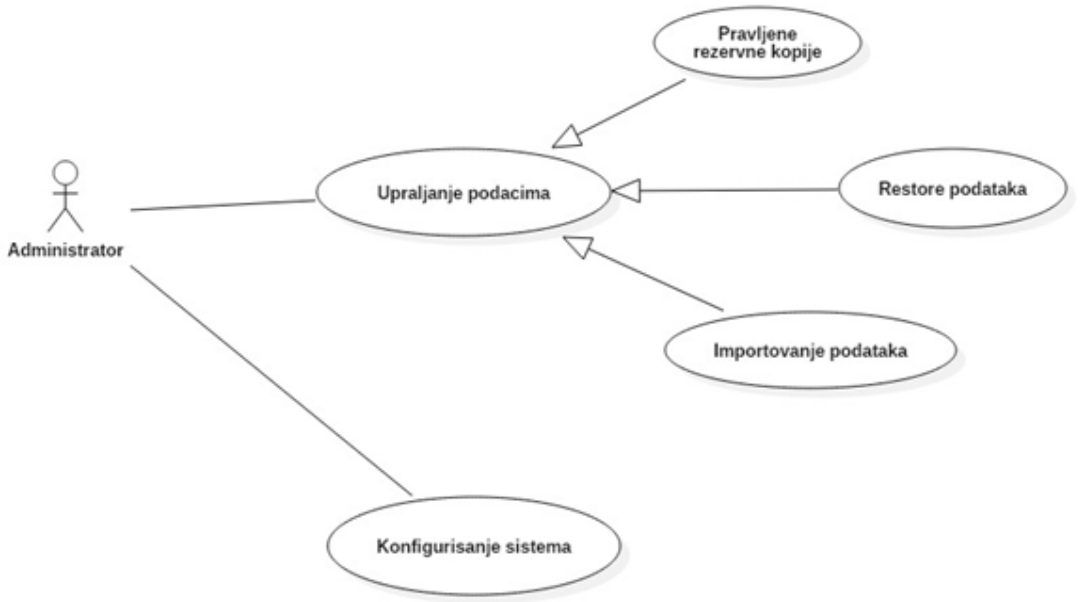
3.6 Kontrola toka

Kontrola toka u okviru **BUST** sistema je centralizovana pri čemu se koristi mehanizam zasnovan na upravljanju događajima (eng. *event-driven*). Ovim se podrazumjeva postojanje kontrolne petlje u kojoj se osluškuju i detektuju događaji, poput klika mišem ili pritiska tastera, i na osnovu toga delegiraju odgovarajući pozivi. S obzirom da sistem koristi grafički korisniči interfejs, ovakva kontrola toka je veoma pogodna. Da bi se održalo konzistentno stanje sistema u slučaju konkurentnog izvršavanja određenih operacija, potrebno je obezbjediti da se one izvršavaju izolovano, odnosno da druga operacija može da se izvršava tek kada se prva operacija kompletira.

3.7 Granična stanja sistema

Granična stanja sistema prilikom uključivanja, isključivanja i izuzetaka, identifikovana u toku dizajna su prikazana na dijagramu graničnih slučajeva upotrebe. Ova stanja se moraju pažljivo obraditi da bi sistem mogao nesmetano funkcionisati.

Graničnim stanjima sistema se uglavnom bavi administrator, koji je ujedno i glavni učesnik u dijagramu graničnih slučajeva upotrebe.



Slika 7: Granična stanja sistema

Na dijagramu su prikazani sledeći slučajevi upotrebe:

- konfigurisanje sistema,
- upravljanje podacima, koji uključuje:
    - importovanje podataka,
    - pravljenje rezervne kopije
    - restore podataka

Konfigurisanje sistema podrazumjeva kreiranje, uništavanje i arhiviranje perzistentnih objekata kao i dealokaciju resursa.

Pravljenje rezervne kopije (nakon unaprijed specifikovanog vremenskog perioda) je neophodna akcija radi očuvanja integriteta podataka i omogućavanja restore-a podataka. Rezerva kopija služi za restore podataka u slučaju otkaza sistema ili neke druge nepredviđene situacije.

Restore podataka se odnosi na vraćanje stanja sistema kakvo je bilo prije nego se dogodila neka nepridviđena situacija ili, u najgorem slučaju, otkaz sistema. Ovim postupkom se podaci, koji su učitani u rezervnu kopiju baze podataka, kopiraju u bazu podataka kako ne bi došlo do gubitka informacija.

Importovanje podataka podrazumjeva učitavanje podataka iz nekih drugih baza podataka ili nekog fajla.