Dizajn sistema

Smart ZEV

Mrđan Poletanović
Dimitrije Kučuk
Bojan Bulatović
Darko Prelić
Slaviša Stojaković

ETF BANJA LUKA 2020.

1. Uvod

Svrha ovog dokumenta je da detaljno opiše dizajn Smart ZEV sistema i da, kao doupna specifikaciji korisničkih zahtjeva, upotpuni opis sistema.

1.1. Namjena sistema

Smart ZEV će biti sistem za finansijsko poslovanje zajednica etažnih vlasnika na prostoru Republike Srpske. Korisnicima ovog sistema biće omogućen jednostavan prikaz svih neophodnih informacija vezanih za finansije jednog ZEV-a, trenutno stanje računa, mjesečni i godišnji izvještaji, kao i pojedinačni izvještaji i dugovanja korisnika. Predsjednik jedne zajednice etažnih vlasnika moći će na jednostavan način generisati izvještaje, pratiti stanje računa, te dugovanja ostalih korisnika. Podaci potrebni za obračunavanje će se unositi pri kreiranju naloga za jednu zajednicu, a podaci o korisnicima i zajednicama etažnih vlasnika biće čuvani u bazi podataka, koja će biti postavljena na server.

1.2. Projektni ciljevi

Ciljevi koje Smart ZEV sistem treba da ispuni su:

- **Pouzdanost**

  Pouzdanost je obezbijeđena time što je server distribuiran, a podaci su replicirani na više čvorova. Na taj način dobijamo sistem koji je otporan na otkaze pojedinih komponenata.

- **Efikasnost**

  Efikasnost Smart ZEV sistema se ogleda u efikasnosti finansijskog poslovanja zajednica etažnih vlasnika što podrazumijeva prikaz trenutnog stanja računa, mjesečnog i godišnjeg izvještaja, kao i pojedinačnog izvještaja i dugovanja korisnika. Takođe, generisanje izveštaja i računa će se vršiti po unaprijed zadatim podacima za obračunavanje koji će se, po potrebi, moći ažurirati.

- **Lakoća upotrebe**

  Sistem mora biti lak za upotrebu kako bi ga mogli korisiti korisnici različitih nivoa poznavanja rada na računaru. Lakoća upotrebe se ogleda u jednostavnom i intuitivnom grafičkom korisničkom interfejsu.

- **Pristupačnost**

  Smart ZEV sistem će biti implementiran kao web aplikacija. Za izvršavanje sistema biće neophodna internet konekcija i internet pretraživač.

- **Sigurnost**

  Web aplikacija Smart ZEV koristiće *https* protokol čime će komunikacija između sistema i korisnika aplikacije biti zaštićena. Korisničke lozinke će se čuvati u bazi podataka zajedno sa korisničkim imenom, te će biti heširane, čime je obezbijeđena tajnost kredencijala neophodnih za prijavu na sistem. U budućim verzijama, možemo očekivati upotrebu digitalnih sertifikata kako bismo u potpunosti obezbijedili sigurnost sistema. Pristup podacima o zajednicama neregistrovanim korisnicima neće biti omogućen.

- **Robusnost**

  Pošto se na serveru čuvaju vrlo značajni podaci, ne smije se dozvoliti da se neki podatak izgubi zbog pada servera. Zbog toga je server distribuiran na više fizički udaljenih čvorova, čime se postiže robusnost sistema.

- **Transparentnost**

  Sistem mora biti transparentan, tj. svaka aktivnost koja uključuje finansijsko poslovanje mora biti vidljiva svim korisnicima. To se postiže generisanjem mjesečnog, odnosno godišnjeg izvještaja koji je dostupan na uvid svim korisnicima.

1.3. Definicije i skraćenice

- Korisnik – stanari i etažni vlasnici
- Predsjednik - predsjednik zajednice etažnih vlasnika
- Administrator – osoba koja upravlja, održava i uređuje sistem, dodaje i uklanja ZEV-ove
- ZEV – zajednica etažnih vlasnika
- DBMS – Database Management System - Sistem za upravljanje bazom podataka
- HTTPS - Hypertext Transfer Protocol Secure - sigurna (kriptovana) verzija protokola za prenos informacija na internetu

1.4. Referentni dokumenti

Referentni dokument je specifikacija korisničih zahtjeva Smart ZEV sistema.

1.5. Kratak pregled dokumenta i predložene arhitekture

Ostatak dokumeta sastoji se iz dva dijela. U prvom dijelu opisana je arhitektura postojećeg sistema, dok je u drugom dijelu dat opis arhitekture za sistem koji se projektuje. U ovom dijelu je opisana dekompozicija sistema i funkcionalnosti svakog podsistema. Osim toga, dat je konceptualni model baze podataka koja će biti korištena, kao i dijagram slučajeva upotrebe koji se odnosi na granična stanja sistema.

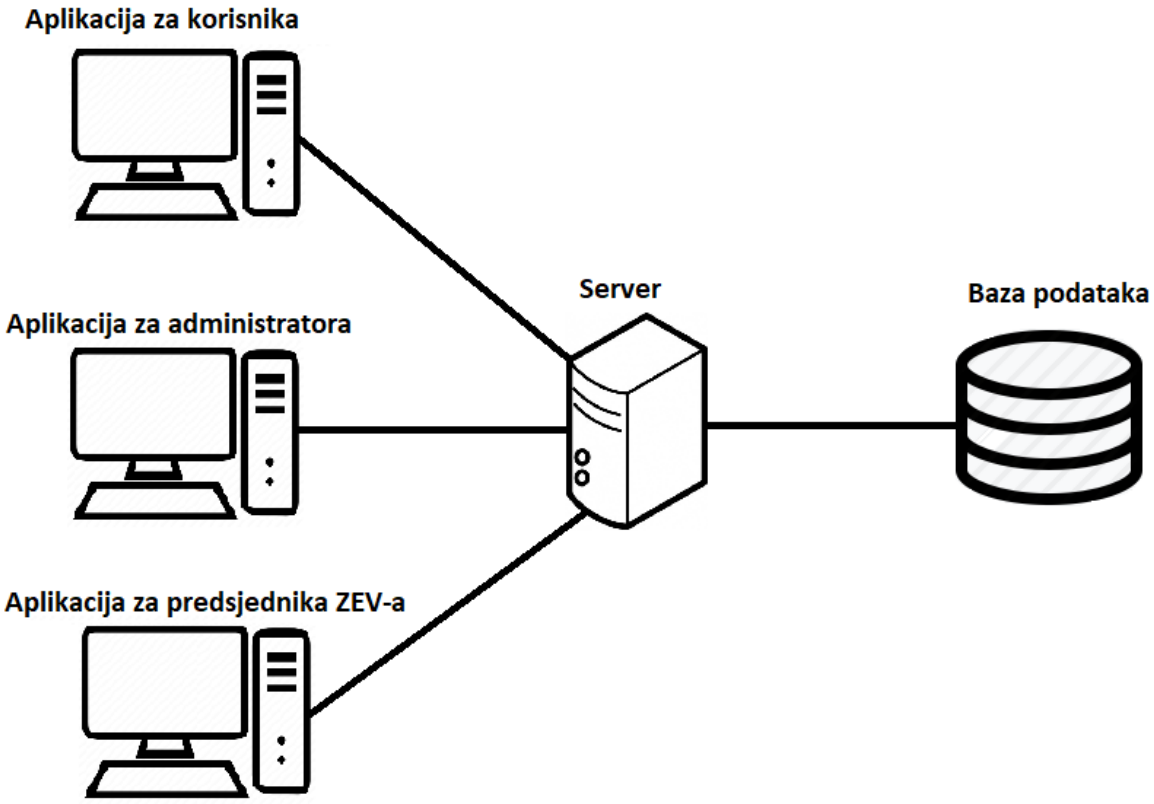
2. Arhitektura postojećeg sistema

Pošto se radi o novom sistemu arhitektura postojećeg sistema ne postoji. Na tržištu postoji nekoliko aplikacija sa različitim arhitekturama koje ne zadovoljavaju sve potrebe korisnika. Uglavnom su to desktop aplikacije čije korišćenje nije tako jednostavno i intuitivno. U nastavku dokumenta slijedi opis predložene arhitekture našeg budućeg sistema.

3. Predložena arhitektura

3.1. Kratak pregled arhitekture i funkcionalnosti podsistema

Sistem će se sastojati iz tri podsistema, a to su: administratorska aplikacija, aplikacija za obične korisike, te aplikacija za predsjednike zajednica etažnih vlasnika (Slika 1). Sva tri sistema će koristiti istu bazu podataka. Nijedan od navedenih podsistema ne zavisi od nekog drugog spoljašnjeg sistema.



Slika 1

Podsistem za predsjednika ZEV-a namijenjen je svim predsjednicima zajednica etažnih vlasnika. Oni su osobe koje uspostavljaju kontakt sa administratorima sistema, te na taj način registruju svoju zajednicu. Korisnik ovog sistema ima veće privilegije u odnosu na ostale stanare, te tako on može unositi račune, generisati izvještaje i slično. Svaki predsjednik na osnovu lozinke i jedinstvenog korisničkog imena može pristupiti sistemu, i tako uređivati poslovanje svoje zajednice etažnih vlasnika.

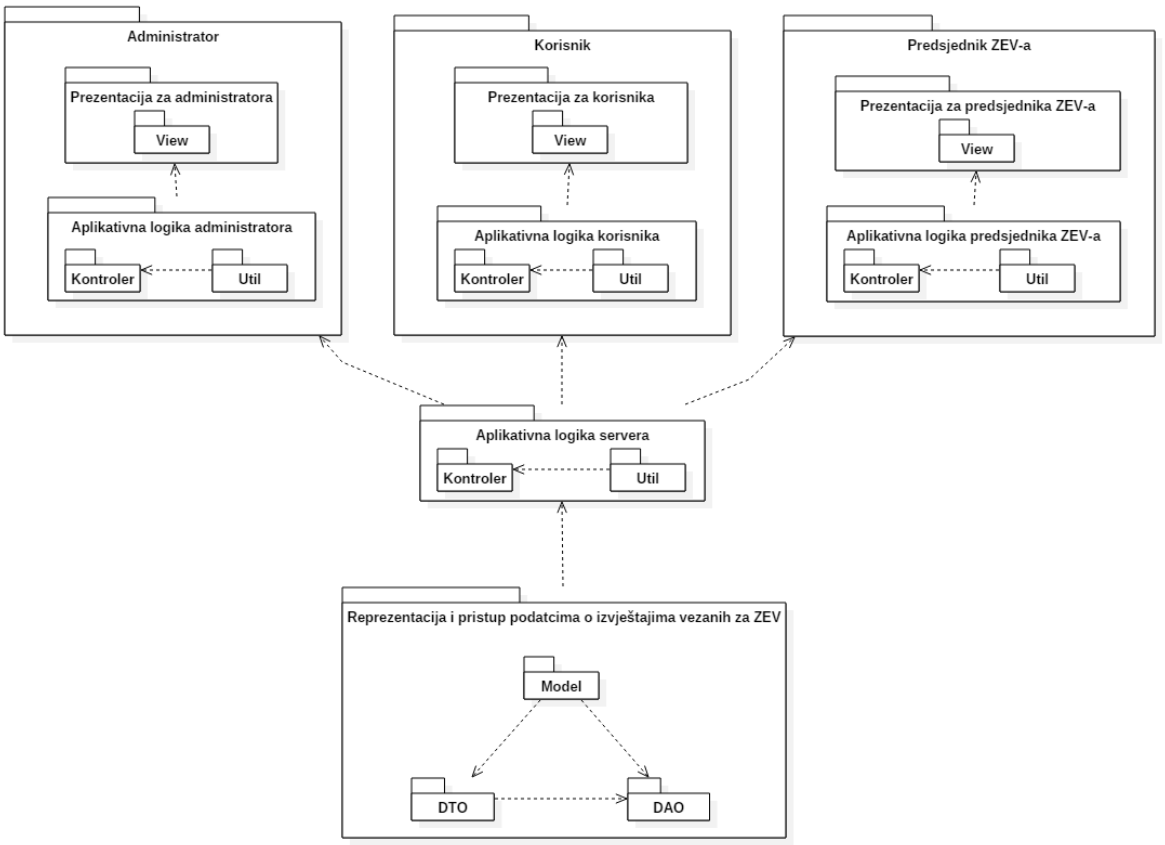
Podsistem za administratora je aplikacija čija je svrha uređivanje sistema, dodavanje novih zajednica, kao i održavanje sistema. Svaki administrator ima svoje korisničko ime i lozinku, te na osnovu tih kredencijala pristupa svom nalogu. On dodaje i uklanja zajednice etažnih vlasnika na sistem, te vrši pregled informacija o već registrovanim zajednicama.

Podsistem za korisnika je namijenjen svim stanarima i etažnim vlasnicima. Registraciju na ovaj sistem može ostvariti bilo koji građanin koji je vlasnik stana u registrovanoj zajednici etažnih vlasnika. Na osnovu unesenih podataka, kreira se profil korisnika i njegovu verifikaciju vrši predsjednik zajednice. Korisnik ima mogućnosti pregleda svih svojih mjesečnih obaveza, kao i godišnjih finansijskih izvještaja zajednice. Korisniku se mogu dodijeliti dodatne privilegije, ako on izvršava neku funkciju u zajednici, npr. blagajnik. Sa tim dodatnim privilegijama, on može unositi račune i evidentirati plaćanje mjesečnih obaveza.

Server čuva podatke o svim ZEV-ovima i njihovim korisnicima u bazi podataka. Takođe, server čuva i podatke za obračunavanje, te na osnovu zadatih parametara od strane korisnika generiše potreban račun ili izvještaj.

3.2. Dekompozicija sistema

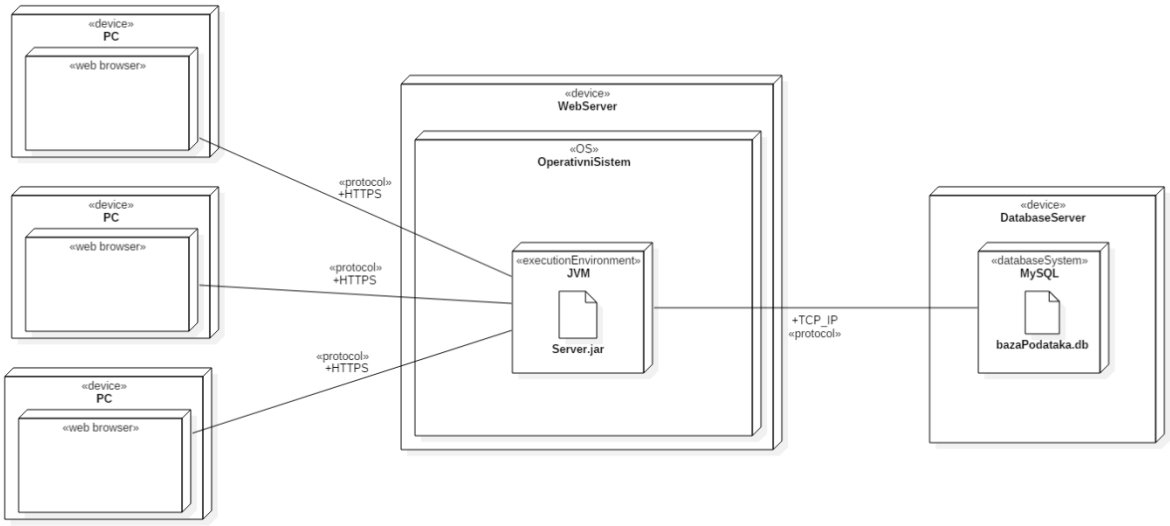
Na Slici 2. Prikazan je UML dijagram paketa (Package Diagram) kojim je prikazana dekompozicija sistema.



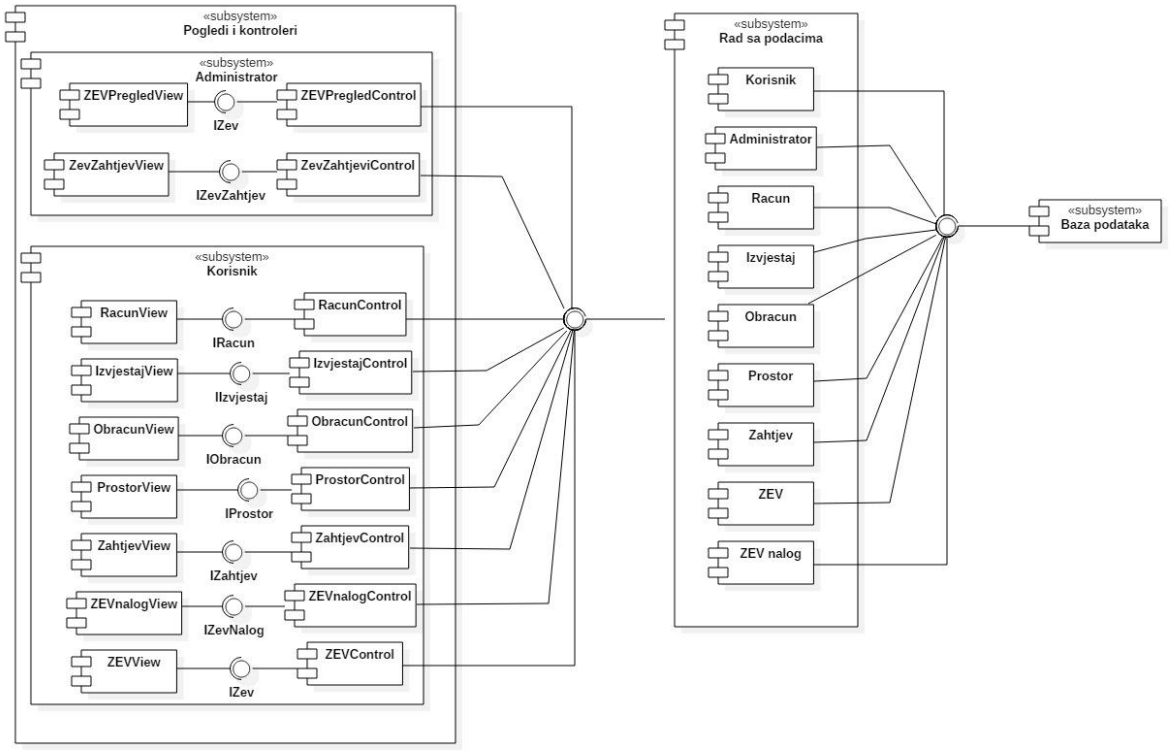
Slika 2

3.3. Hardversko/softversko mapiranje

Na Slici 3 prikazan je UML dijagram razmještaja (Deployment Diagram) a na Slici 4 UML dijagram komponenti (Component Diagram).



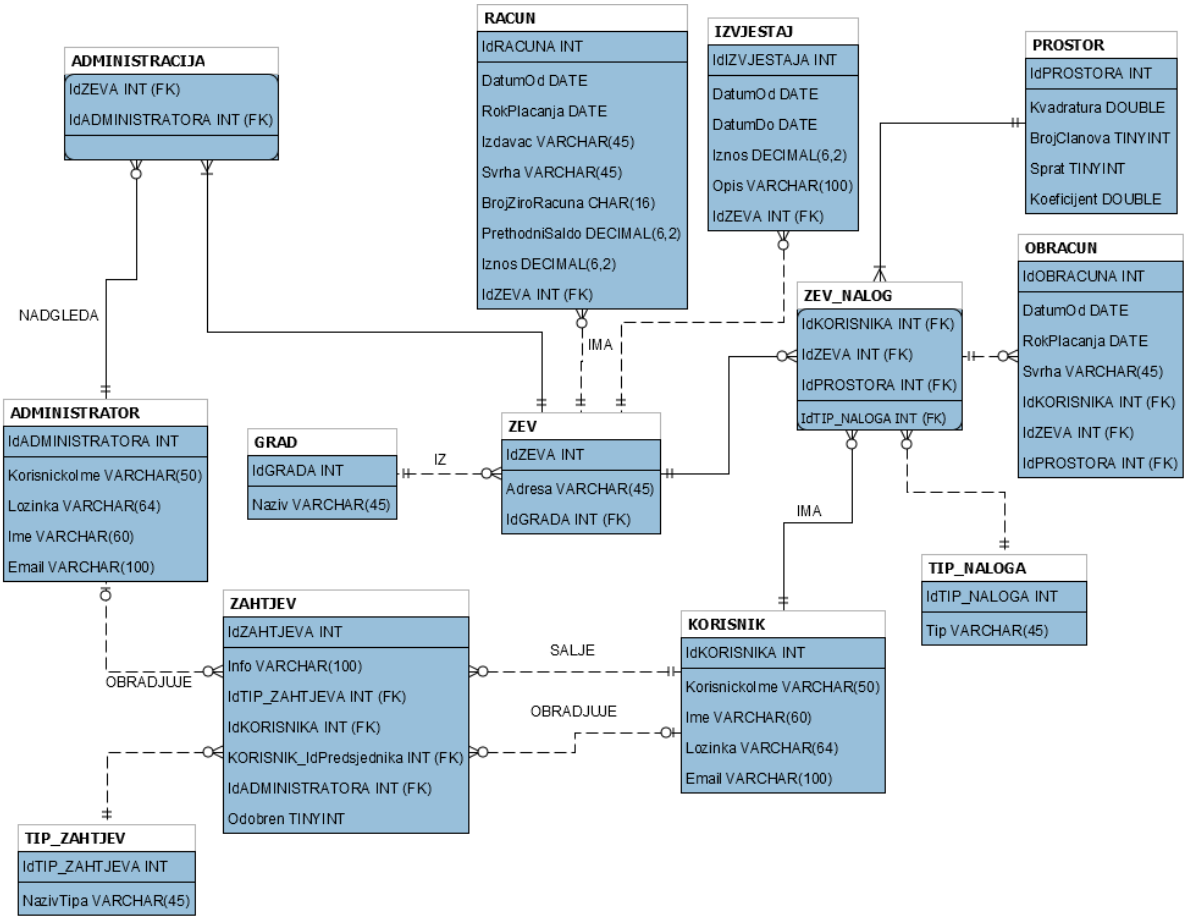
Slika 3



Slika 4

3.4. Perzistentni sloj

Na serveru se čuva baza podataka koja sadrzi informacije o korisnicima, ZEV-ovima, računima i izvještajima. Na Slici 5 prikazan je konceptualni model ove baze.


ADMINISTRACIJA
IdZEVA INT (FK)
IdADMINISTRATORA INT (FK)
RACUN
IdRACUNA INT
DatumOd DATE
RokPlacanja DATE
Izdavac VARCHAR(45)
Svrha VARCHAR(45)
BrojZiroRacuna CHAR(16)
PrethodniSaldo DECIMAL(6,2)
Iznos DECIMAL(6,2)
IdZEVA INT (FK)
IZVJESTAJ
IdIZVJESTAJA INT
DatumOd DATE
DatumDo DATE
Iznos DECIMAL(6,2)
Opis VARCHAR(100)
IdZEVA INT (FK)
PROSTOR
IdPROSTORA INT
Kvadratura DOUBLE
BrojClanova TINYINT
Sprat TINYINT
Koeficijent DOUBLE
OBRACUN
IdOBRACUNA INT
DatumOd DATE
RokPlacanja DATE
Svrha VARCHAR(45)
IdKORISNIKA INT (FK)
IdZEVA INT (FK)
IdPROSTORA INT (FK)
ZEV_NALOG
IdKORISNIKA INT (FK)
IdZEVA INT (FK)
IdPROSTORA INT (FK)
IdTIP_NALOGA INT (FK)
NADGLEDA
IMA
ADMINISTRATOR
IdADMINISTRATORA INT
Korisnickolme VARCHAR(50)
Lozinka VARCHAR(64)
Ime VARCHAR(60)
Email VARCHAR(100)
GRAD
IdGRADA INT
Naziv VARCHAR(45)
IZ
ZEV
IdZEVA INT
Adresa VARCHAR(45)
IdGRADA INT (FK)
IMA
TIP_NALOGA
IdTIP_NALOGA INT
Tip VARCHAR(45)
ZAHTJEV
IdZAHTJEVA INT
Info VARCHAR(100)
IdTIP_ZAHTJEVA INT (FK)
IdKORISNIKA INT (FK)
KORISNIK_IdPredsjednika INT (FK)
IdADMINISTRATORA INT (FK)
Odobren TINYINT
SALJE
OBRADJUJE
KORISNIK
IdKORISNIKA INT
Korisnickolme VARCHAR(50)
Ime VARCHAR(60)
Lozinka VARCHAR(64)
Email VARCHAR(100)
OBRADJUJE
TIP_ZAHTJEV
IdTIP_ZAHTJEVA INT
NazivTipa VARCHAR(45)

Slika 5

3.5. Kontrola prava pristupa

U sledećoj tabeli je prikazana kontrola prava pristupa korisnika sistema.

Učesnici	Klase	
	Server	GUIKontroler
Administrator	ukloniZEV() deaktivirajNalog() odobriZahtjev() odbijZahtjev() kreirajZev() dodajAdministratora() ukloniAdministratora()	pregledajZahtjeve() nadgleda() prijava() odjava()
Predsjednik ZEV-a	generisiObracun() generisiIzvjestaj() dodijeliPrivilegije() konfigurisiPodesavanja() obradiZahtjev() unesiPrihod() unesiRashod() dodajNalog() izbrisiNalog() azurirajInformacije()	pregledajStanje() pregledajZahtjeve() prijava() odjava()
Korisnik	azurirajInformacije() uclanjivanje() registracija() generisiNalog()	pregledajObracun() pregledajZevove() prijava() odjava()
Odgovorno lice	azurirajInformacije() uclanjivanje() registracija() generisiNalog() obradiZahtjev() unesiPrihod() unesiRashod()	pregledajObracun() pregledajZevove() prijava() odjava() pregledajZahtjeve()

3.6. Identifikacija konkurentnosti

Svi korisnici mogu pristupati bazama podataka konkurentno, a sinhronizaciju obezbjeđuje DBMS.

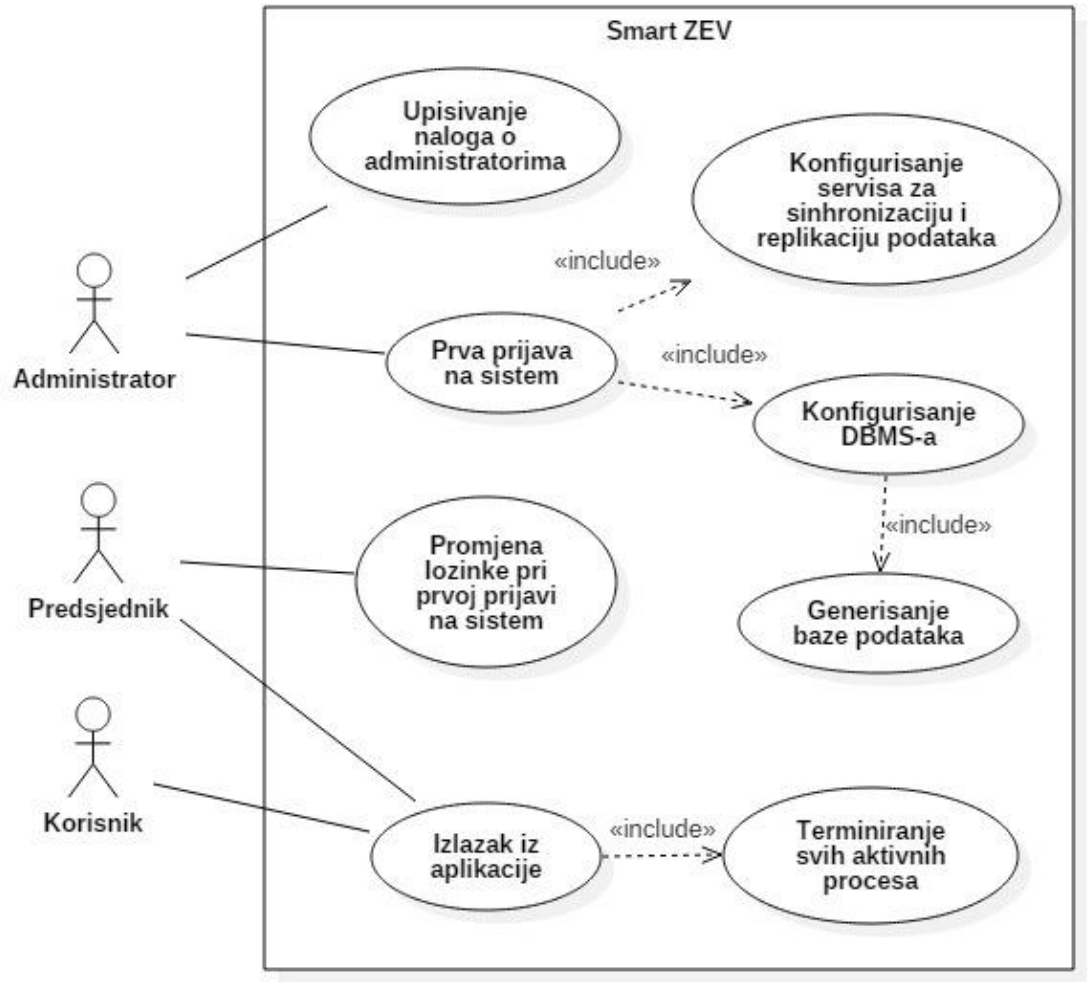
3.7. Kontrola toka

Kontrola toka je centralizovana, pri čemu server ima kontrolu pristupa podacima i obrađuje zahtjeve klijenata. Mehanizam kontrole toka je event-driven, jer se server aktivira na zahtjeve klijenata.

3.8. Granična stanja sistema

- Pri pokretanju servera konfiguriše se servis za automatsku sinhronizaciju i replikaciju podataka. Osim toga potrebno je generisati baze podataka na serveru i podesiti DBMS.
- Prije prvog pokretanja sistema potrebno je da u bazi sa nalozima bude upisan nalog o administratorima
- Pri prvom pokretanju sistema potrebno je da predsjednik promijeni lozinku
- Ukoliko na nekom od serverskih čvorova dođe do nestanka električne energije, potrebno je priključiti mašinu na alternativno napajanje u što kraćem vremenskom roku, a ostatak serverskih čvorova se automatski sinhronizuje.
- Ukoliko dođe do trajnog oštećenja nekog od serverskih čvorova, ostali čvorovi se automatski sinhronizuju, dok se teži ka što bržoj zamjeni oštećene mašine.
- Ukoliko korisnik izađe iz aplikacije svi aktivni procesi biće terminirani
- Ukoliko korisnik izgubi internet konekciju svi aktivni procesi biće terminirani

Na Slici 6 prikazan je UML Use Case Diagram za granične slučajeve upotrebe.



Slika 6