

Univerzitet u Beogradu
Matematički fakultet

Unapređenje informacionog sistema za
evidenciju računarske opreme u
Republičkom geodetskom zavodu

Master rad

Boris Radovanović

Beograd, 30.11.2011.

Univerzitet u Beogradu - Matematički fakultet
Master rad

Autor: Boris Radovanović
Naslov: Unapređenje informacionog sistema za
evidenciju računarske opreme u
Republičkom geodetskom zavodu
Mentor: dr Saša Malkov
Članovi komisije: dr Nenad Mitić
dr Miodrag Živković
Datum: 30.11.2011.

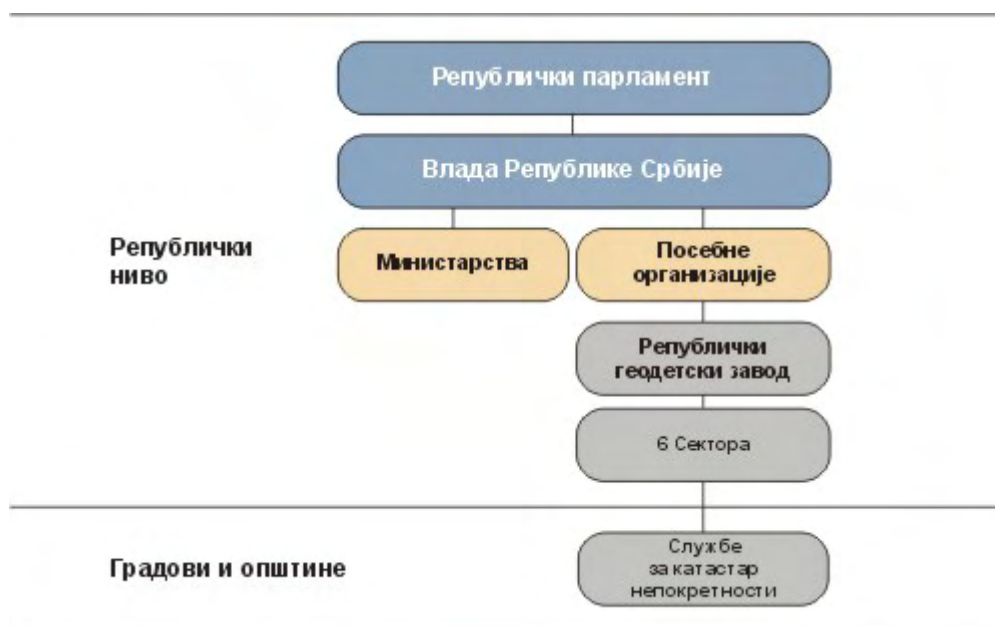
Sadržaj

1	Uvod	3
2	Opis metoda i alata	5
2.1	Metodologija razvoja	5
2.2	Dijagram slučajeva korišćenja	6
2.3	Dijagram klasa	7
2.4	Dijagram entiteta i odnosa	8
2.5	Notacija za modeliranje poslovnih procesa	9
2.6	Korišćeni alati	10
3	Analiza i definisanje zahteva	11
3.1	Preduzete aktivnosti na sagledavanju postojećeg stanja	11
3.2	Uočeno stanje	11
3.3	Organizaciona struktura Zavoda	17
3.4	Diskusija o slabostima i kvalitetima baze i aplikacija	17
3.5	Definisanje ciljeva	19
4	Projektovanje izmena	21
4.1	Entiteti <i>Komponenta</i> i <i>Instanca</i>	21
4.2	Troslojna klasifikacija računarskih komponenti	23
4.3	Podaci o: <i>Zaposlenima</i> , <i>Dobavljačima</i> i <i>Statusima instanci</i>	25
4.4	Nova organizaciona struktura	26
4.5	Prijava i servis kvarova	27
4.6	Korisnici aplikacije i forme za prikazivanje podataka	29
5	Implementacija	31
5.1	Primenjene tehnologije	31
5.1.1	Platforma	31
5.1.2	Programski jezik	31
5.1.3	Tehnologija <i>Ajax</i>	32
5.1.4	Baza podataka	32
5.1.5	Razvojno okruženje	32
5.2	Komponenta za povezivanje sa bazom podataka	33
5.3	Forme za prikaz podataka	35
5.4	Tranzicija	37
6	Zaključak	41

7	Dodaci	42
7.1	Dodatak A - Formulari	42
7.2	Dodatak B - Baza podataka	48
7.2.1	Dodatak B1 - Dijagrami baza podataka	48
7.2.2	Dodatak B2 - Odnosi između relacija	50
7.2.3	Dodatak B3 - Indeksi tabela	52
7.2.4	Dodatak B4 - Provera ograničenja	52
8	Reference	54

1 Uvod

Republički geodetski zavod (u daljem tekstu: Zavod) je zasebna organizacija pri Vladi Republike Srbije koja vrši stručne poslove, poslove državne uprave i geodetske radove u inženjersko-tehničkim oblastima. Sadašnji oblik funkcionisanja Zavoda na snazi je od decembra 1992. godine, dok su njegove preteče bile geodetske uprave koje su funkcionisale kao sastavni delovi Opštinskih uprava.



Slika 1: Organizaciona struktura

Zavod danas broji oko 2500 zaposlenih lica. Delokrug čine geodetski radovi i poslovi državne uprave koji se odnose na osnovne geodetske radove, katastarski i komasacioni premer, osnivanje, obnovu i održavanje katastra nepokretnosti, premer državne granice, vođenje registra državne granice i ostale delatnosti. Poslove iz svog delokruga Zavod obavlja iz sedišta u Beogradu, kao i van sedišta u svim opštinama. Van sedišta, u okviru Sektora za katastar nepokretnosti, funkcionišu Službe za katastar nepokretnosti. One čine najveći deo Zavoda i broje 156 službi, teritorijalno organizovanih po opštinama u Republici Srbiji.

Novom sistematizacijom krajem decembra 2010. godine poslovanje Zavoda organizovano je podelom na šest sektora među kojima je Sektor za informatiku i komunikacije. Jedan od poslova koje obavlja Odeljenje za informatičku podršku, unutar ovog sektora, jeste evidencija kompletne računarske

opreme koju Zavod poseduje. Za tu svrhu implementiran je informacioni sistem kako bi se organizovalo upravljanje računarskom opremom.

Zavod poseduje 2500-3000 računara koji sa pratećom opremom broje preko 10.000 jedinica računarske opreme. Jedna od glavnih celina informacionog sistema je deo za prijavu i servisiranje kvarova na opremi. Pored računarske opreme sistem skladišti podatke o katastrima i zaposlenim licima. Vremenom se organizacija Zavoda menjala te su određene funkcije sistema zastarele. Uzimajući u obzir i njegove nedostatke koji su predstavljeni u ovom radu, javila se potreba za implementacijom novog unapređenog sistema.

Cilj ovog rada je unapređenje informacionog sistema koji se koristi za tu svrhu.

2 Opis metoda i alata

2.1 Metodologija razvoja

Zbog specifičnosti problema i potrebe da se postojeći materijali i dizajn baze podataka u što manjoj meri izmeni, kako bi bio olakšan prelazak sa starog na novi sistem, korišćena je hibridna metodologija razvoja, odnosno modifikacija metodologije vodopada i inkrementalna strategija razvoja. Inkrementalni razvoj spada u savremene koncepte razvoja objektno orijentisanih metodologija [1]. Ove metodologije se odlikuju postavljanjem objekata (entiteta, subjekata, servisa i interfejsa) u centar pažnje.

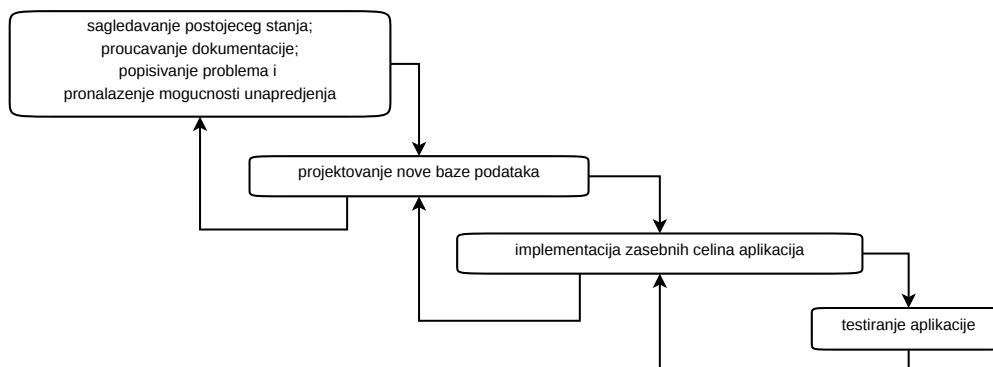
Kako je priroda informacionog sistema takva da se celine nadovezuju jedna na drugu i da zavise od ispravnosti prethodnih celina, inkrementalnim razvojem su implementirane, isporučivane i testirane generacije novog sistema. Omogućen je povratak na projektovanje celine koja je trenutno u fazi implementacije, ako bi se ispostavilo da postoji bolje rešenje za njenu funkcionalnost, što predstavlja osobinu modifikovane metodologije vodopada.

Pošto se rad zasniva na unapređenju postojećeg, a ne na pravljenju novog sistema, nije bilo potrebe za korišćenjem novijih razvojnih metodologija, kao što je *RUP* metodologija.

Na početku rada na projektu postavljeni su sledeći zadaci koji su predstavljali smernice tokom rada:

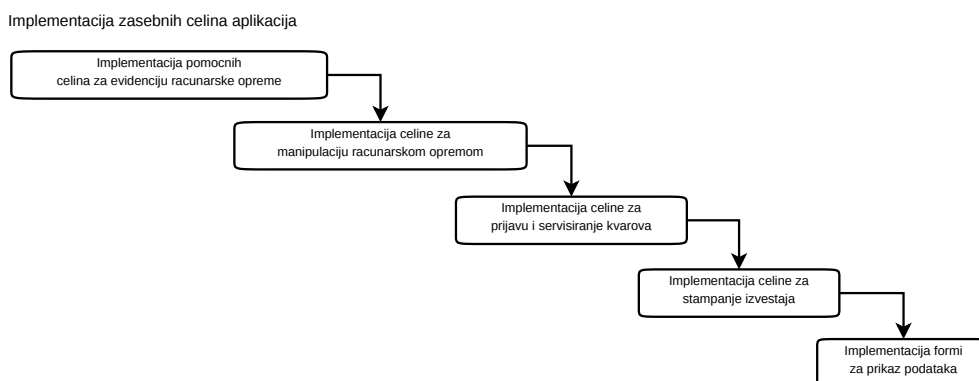
- sagledavanje postojećeg stanja; proučavanje dokumentacije; popisivanje problema i pronalaženje mogućnosti za unapređenje
- projektovanje nove baze podataka čiji dizajn mora biti pogodan za uvoz podataka iz stare baze
- implementacija zasebnih celina aplikacije
 - pomoće celine za evidenciju računarske opreme (upravljanje lokacijama, organizacionim jedinicama, zaposlenim licima, dobavljačima računarske opreme, i sl.)
 - upravljanje računarskom opremom
 - prijavljivanje i servisiranje kvarova računarske opreme
 - štampanje izveštaja
 - forme za prikaz podataka različitim nivoima korisnika
- testiranje aplikacije

Opisani postupci mogu se predstaviti grafički u obliku modifikovane metodologije *vodopada* gde se, pri nailasku na problem u kasnijim fazama, projekat vraća na ponovnu proveru u fazi projektovanja.



Slika 2: Faze kroz koje projekat prolazi

Zbog svoje obimnosti treća faza projekta može se predstaviti detaljnije (slika 3). Iz dijagrama se vidi zavisnost svake od celina aplikacije. Nakon svake celine neophodno je ispitati ispravnost njegove funkcije pošto se svaka naredna celina nadovezuje na prethodne korake i zavisi od uspešnosti njihove izrade.

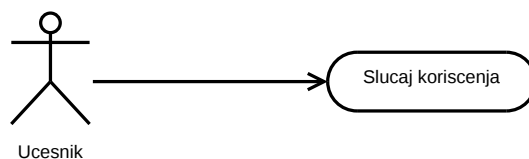


Slika 3: Povezanost zasebnih celina aplikacije

2.2 Dijagram slučajeva korišćenja

Slučajevi korišćenja (eng. *use case*) predstavljaju tehniku modeliranja ponašanja sistema, iz aspekta interakcije između korisnika i sistema.

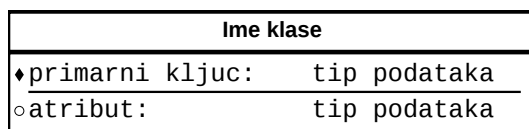
Dijagram slučajeva korišćenja (eng. *use case diagram*) prikazuje učesnike i njihove slučajeve korišćenja. Jednostavan je za razumevanje čak i bez formalnog poznavanja notacije, što ima veliki značaj u komunikaciji između razvojnog tima i korisnika, kao i u diskusiji sa klijentima. U radu se ovaj dijagram koristi prilikom detaljnijeg pojašnjenja procedure za prijavu i servisiranje kvarova. Na slici 4 je prikazan osnovni oblik dijagrama slučajeva korišćenja.



Slika 4: Dijagram slučajeva korišćenja (eng. *use case diagram*)

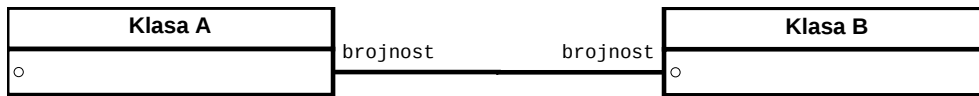
2.3 Dijagram klasa

Klasa predstavlja apstrakciju objekta sa istim svojstvima (atributima) i ponašanjem (operacijama). Tokom faze projektovanja za predstavljanje klasa podataka korišćeni su *dijagrami klasa*. Oni su jedni od najčešće korišćenih strukturnih dijagrama *UML*-a [2]. Pored naziva klasa sadrže attribute, metode i odnose između klasa. Pri izradi ovog projekta prepoznavanje klasa vršilo se na osnovu podataka iz tehničke dokumentacije starih aplikacija. Više klasa je zadržano iz stare aplikacije, neke klase su dodate, a nekima su izmenjeni atributi i odnosi između ostalih klasa.



Slika 5: Predstavljanje klase

Osnovni oblik predstavljanja klase je prikazan na slici 5. Odnosi između klasa se predstavljaju *asocijacijama* (slika 6). Asocijacija je najopštiji odnos između klasa, koji označava da je za punu funkcionalnost objekta jedne od klasa neophodan jedan ili više objekata druge klase.



Slika 6: Asocijacija - odnos između klasa

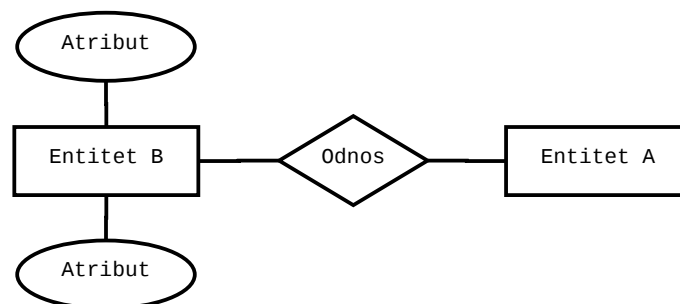
Brojnost označava broj objekata jedne klase koji mogu biti u odnosu sa jednim objektom druge klase i može imati sledeće vrednosti:

- 0..1: 0 ili 1 objekat
- 0..*: 0 ili više objekata
- 1: tačno jedan objekat
- 1..*: 1 ili više objekata

2.4 Dijagram entiteta i odnosa

Dijagram entiteta i odnosa (eng. *Entity Relationship Diagram*) je tehnika za vizuelni prikaz logičke strukture baze podataka. Svaki entitet na dijagramu odgovara tabeli u bazi podataka dok odnosi između entiteta predstavljaju strane ključeve.

Entiteti se prikazuju u obliku pravougaonika koji sadrži ime entiteta. Svojstva (atributi) se prikazuju u obliku elipse koji sadrži naziv atributa i koja je punom linijom povezana sa relevantnim entitetom. Odnosi se prikazuju u obliku romba u kome je upisano ime odnosa. Svaka linija je označena sa *1* ili *n* radi naznake da li je odnos *jedan-jedan*, *jedan-više*, itd.

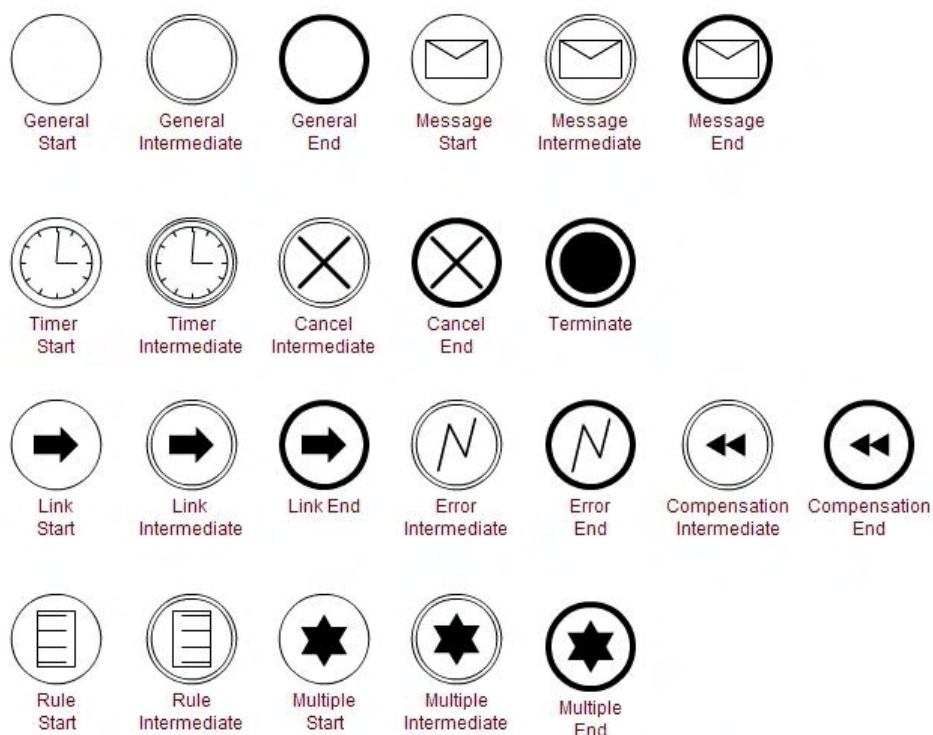


Slika 7: Dijagram entiteta i odnosa (ER diagram)

2.5 Notacija za modeliranje poslovnih procesa

U radu je korišćen dijagram saradnje, koji predstavlja jednu od tehnika za modeliranje poslovnih procesa (eng. *Business Process Modeling Notation* – *BPMN*) [3]. Ova tehnika obuhvata i različite oblike zapisivanja nefunkcionalnih zahteva. Pored dijagrama saradnje u tehnike modeliranja procesa spadaju i dijagrami procesa.

U dijagramu saradnje svakom subjektu odgovara po jedan dugačak pravougaonik u kojem se nalaze ime subjekta i operacije koje subjekat izvršava. Početak procesa označava se tankom kružnicom, dok je kraj procesa obeležen podebljanom kružnicom. Svaki subjekat izvršava određene operacije. Strelicama se označava redosled izvršavanja operacija. Račvanje puteva u zavisnosti od ispunjenosti nekog uslova se označava romбом. Dodatnim simbolima moguće je predstaviti podatke koji se koriste u operacijama ili poruke koje subjekti razmenjuju. Najčešći simboli i njihovo značenje su prikazani na slici 9.



Slika 8: Simboli BPMN dijagrama

2.6 Korišćeni alati

Aplikacija je implementirana na programskom jeziku *C#* [4], za platformu *ASP.NET* [5], pomoću razvojnog alata *Microsoft Visual Studio 2010* [6].

Svi dijagrami izrađeni su u pomoćnom programu *Dia* [7].

3 Analiza i definisanje zahteva

3.1 Preduzete aktivnosti na sagledavanju postojećeg stanja

Kako bih se upoznao sa postojećim stanjem sistema dogovorio sam više sastanaka sa predstavnicima Zavoda. Na sastancima su mi predstavljene aplikacije koje je Zavod koristio tokom prethodnih godina, njihove mane i funkcije koje su nedostajale. Tom prilikom preuzeo sam tehničku dokumentaciju aplikacija [8] [9] [10], čijim sam proučavanjem upoznao principe njihovog funkcionisanja. Sve dodatne informacije o sistemu dobijao sam u komunikaciji sa zaposlenim licima.

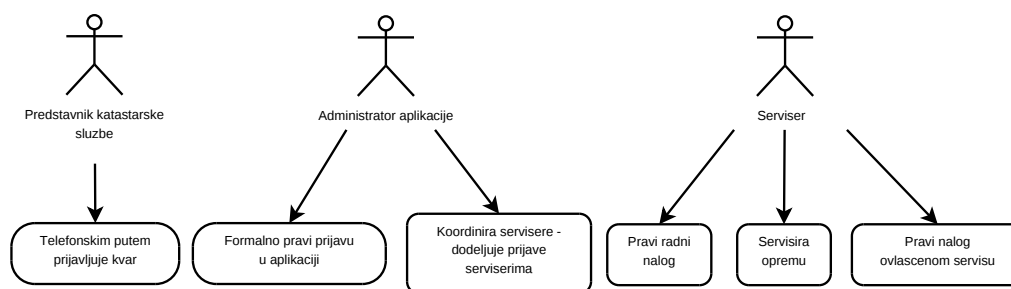
3.2 Uočeno stanje

Izradi starih aplikacija prethodilo je obimno prikupljanje podataka o inventaru koji Zavod ima na raspolaganju. U prvoj fazi zaposleni u Zavodu prikupljali su podatke na terenu i zapisivali su ih u formulare. Izgled formulara je priložen u dodatku *Dodatak A - Formulari*. Druga faza se sastojala u unošenju podataka iz formulara u bazu podataka *Access*. U poslednjoj trećoj fazi podaci su objedinjavani u jednu bazu i klasifikovani su po kategorijama. Kako Zavod koristi *Windows* servere, veb aplikacija je napisana u jeziku *Visual Basic* i koristi *Microsoft SQL Server 2000*.

Počevši od 2002. godine izrađene su dve aplikacije. Prva je veb aplikacija, nazvana *Računarska oprema*, i njena namena je evidentiranje računarske i komunikacione opreme kao i softvera koji se koristi u Zavodu. Glavne funkcije su unošenje, brisanje, ažuriranje i pretraživanje podataka o opremi. U *Računarskoj opremi* se takođe nalaze podaci o lokacijama na kojima se oprema nalazi, zaposlenima koji koriste opremu, dobavljačima od kojih je oprema dobavljena i koji su odgovorni za servis opreme u garantnom roku. Druga je desktop aplikacija, nazvana *HelpDesk*, koja predstavlja proširenje funkcija već postojeće baze koristeći iste podatke iz prve aplikacije. Namena joj je evidentiranje prijave kvarova i intervencija na računarskoj opremi i softveru. Pošto je pristup omogućen samo autorizovanim korisnicima, nisam imao svakodnevni samostalan pristup aplikacijama, već isključivo pod nadzorom tokom redovnih poseta Zavodu. Iz tog razloga njihovo unapređenje sam najviše zasnivao na informacijama dobijenih tokom poseta u komunikaciji sa predstavnikom Zavoda i pri proučavanju tehničke dokumentacije.

U dokumentaciji je najveći akcenat stavljen na dizajn baze podataka. Detaljno su opisani odnosi između entiteta, njihove namene i ograničenja. Navedeni su nivoi korisnika i njihova prava pristupa svakoj od tabela. Nešto manji deo zauzima uputstvo za korišćenje aplikacije od strane zaposlenih u Zavodu i formulari koji su u upotrebi prilikom prijave kvarova, pravljenje naloga servisu i sličnim situacijama.

Trenutno se u Zavodu "Sektor za informatiku i komunikacije", pored ostalog, bavi i održavanjem računarske opreme. Pored njihove evidencije, po unapred utvrđenom pravilniku, vrši se prijava i servisiranje kvarova. Pošto udaljeni korisnici nisu u mogućnosti da neposredno koriste aplikaciju, procedura je takva da katastri telefonskim putem obaveštavaju centralu Zavoda o nastalom kvaru. Centrala tada vrši *Prijavu* problema pomoću desktop aplikacije *HelpDesk*. Prilikom *Prijave* potrebno je evidentirati inventarski broj opreme, datum prijave, ime i prezime osobe koja je prijavila problem, opis samog problema, organizaciona jedinica gde se oprema nalazi i kontakt telefon. Na osnovu *Prijave* štampa se *Radni nalog* za intervenciju koji se daje zaposlenom u Zavodu zaduženom za servisiranje računara. Serviser je dužan da otkloni problem ili konstatuje kvar, te da preporuči servisiranje ili rashodavanje uređaja. Takođe kod intervencije potrebno je upisati i *Vrstu intervencije* prema zadanom šifarniku. Dodeljivanje *Naloga servisu* vrši se kada serviser u Zavodu označi na *Radnom nalogu* da se određena oprema treba popraviti u servisu. Servis vraća računare sa *Servisnim izveštajem* na kome se nalazi broj i datum izveštaja, vreme utrošeno za intervenciju, tip servisiranja, opis rada servisera i spisak ugrađenih delova.



Slika 9: Akteri i slučajevi upotrebe u proceduri za prijavu i servis kvarova

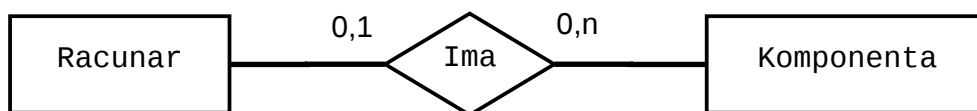
Jedna od sekundarnih funkcija koje poseduje stari sistem jeste štampanje izveštaja različitih namena i postavljanje novosti na naslovnu stranu veb aplikacije. Izveštaji se mogu pregledati u internet pretraživaču ili se mogu odštampati na papiru. Klasifikovani su po grupama te se, primera radi, u klasifikacionoj grupi "statistika" nalaze podaci o broju perifernih uređaja u sva-

koj organizacionoj jedinici posebno. Novosti mogu čitati isključivo korisnici aplikacije. One se koriste kako bi se prenele važne informacije korisnicima o radu sistema, novim mogućnostima koje su naknadno implementirane, o dopunama uputstva za korišćenje i ostalim sličnim stvarima.

Redovno pravljenje rezervnih kopija podataka u određenim vremenskim intervalima veoma je značajno za sistem, jer omogućava brz oporavak u slučaju oštećenja baze. Kopije se prave na tri načina, kojima se gubitak informacija svodi na minimum u slučaju pada sistema:

- *Puna rezervna kopija* podataka pravi se jednom nedeljno – nedeljom pre podne kada nema aktivnosti nad bazom podataka
- *Diferencijalna rezervna kopija* pravi se na kraju svakog radnog dana i obuhvata promene nad bazom koje su nastale od prethodnog pravljenja rezervne kopije
- radnim danima, svakog sata, arhivira se *dnevnik transakcija*

Pošto se u radu često pominju različite verzije baze podataka u daljem tekstu koristićemo izraz *prvobitna baza podataka* za bazu podataka veb aplikacije, dok ćemo bazu podataka desktop aplikacije zvati *nadograđena baza podataka* pošto predstavlja proširenu bazu veb aplikacije. Termin *nova baza podataka* biće korišćen za bazu podataka nove aplikacije koja je rezultat ovog rada.



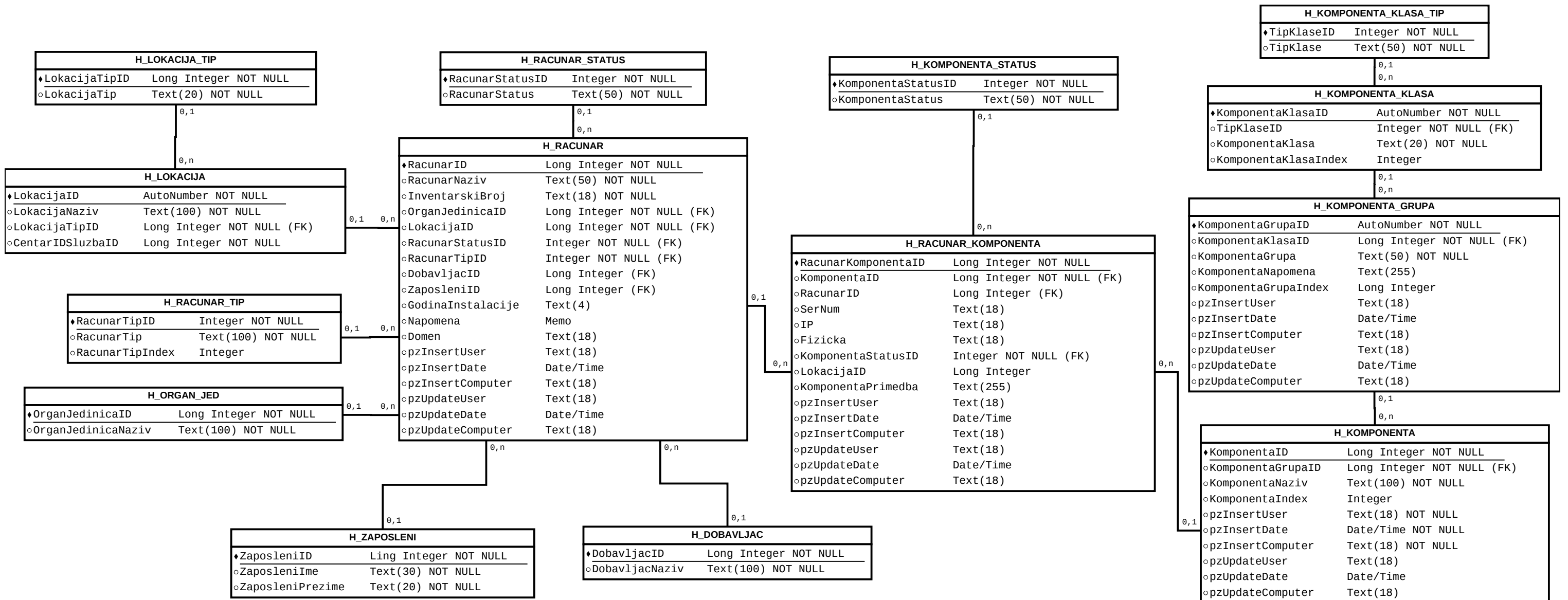
Slika 10: Entiteti *Računar* i *Komponenta*

Dva glavna entiteta prvobitne i nadograđene baze podataka su *Računar* i *Komponenta*. *Računar* se sastoji iz *Komponenti*, koje su klasifikovane po grupama, klasama i tipovima klasa. Tipovi klasa su hardver i softver. U okviru tipa klase „hardver“ postoje klase: monotori, komponente, periferije i ostalo. U okviru klase „komponente“ postoje grupe: procesori, memorije, diskovi, oprički uređaji i ostalo. Računarska oprema može imati određen *Status*: ispravna, neispravna, delimično ispravna, na servisu ili na terenu. Za nju je zadužen *Korisnik* koji može biti zaposlen u Zavodu ili može biti angažovan, na primer, ugovorom o delu ili preko omladinske zadruge. Oprema se nalazi na nekoj *Lokaciji* i pripada određenoj *Organizacionoj jedinici*. *Dobavljač* je

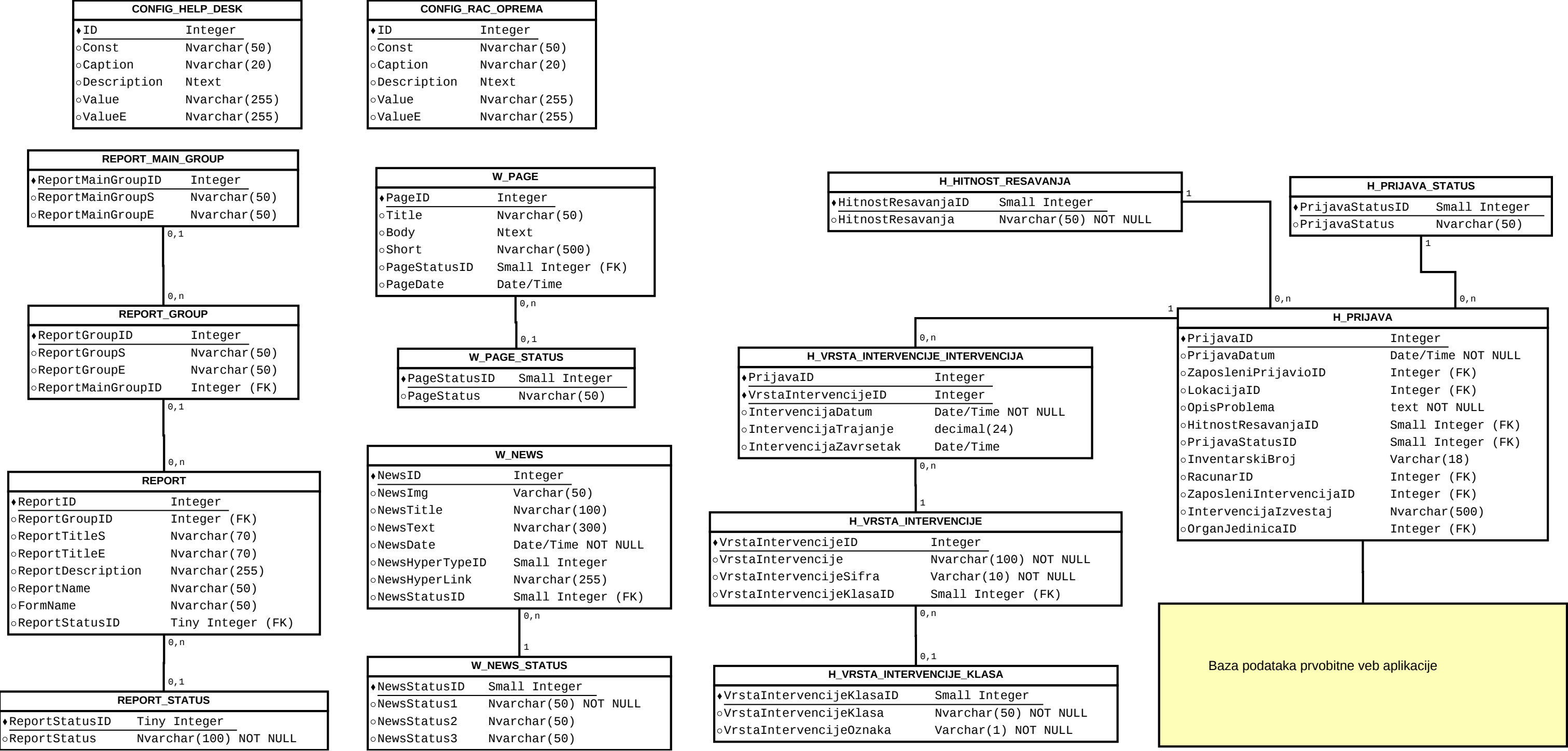
firma ili pojedinac od koje je oprema nabavljena i koji je odgovoran za servisiranje opreme u garantnom roku. Svi odnosi navedenih entiteta u prvobitnoj veb aplikaciji jesu odnosi *nula ili jedan prema nula, jedan ili više*.

Nadograđena baza, koju koristi desktop aplikacija, ima nešto konzervativnije odnose te se pored navedenog pojavljuje i odnos *jedan prema nula, jedan ili više*. Time su veze između pojedinih entiteta rigoroznije te postojanje jedne instance uključuje obaveznu povezanost sa drugim entitetom. U nadograđenoj bazi najbitnije proširenje napravljeno je za potrebe prijave i servisiranja kvarova na opremi. Iz prethodno opisane procedure za prijavu kvarova proilazi potreba za entitetima za *Prijavu kvarova*, *Status* u kojem se problematična oprema nalazi, *Spisak i Klasifikacije intervencija* izvršene nad opremom i *Hitnost rešavanja* prijavljenog problema. Pored navedenih entiteta u nadograđenoj bazi podataka nalaze se i sporedni entiteti koji se koriste za čuvanje podataka o *Izveštajima* različitih tipova i *Novostima* koje se prikazuju na naslovnoj strani aplikacije.

Slika 11 prikazuje model prvobitne baze podataka veb aplikacije, dok je na slici 12 prikazan model nadograđene baze podataka koju koristi desktop aplikacija.



Slika 11: Model baze podataka veb aplikacije



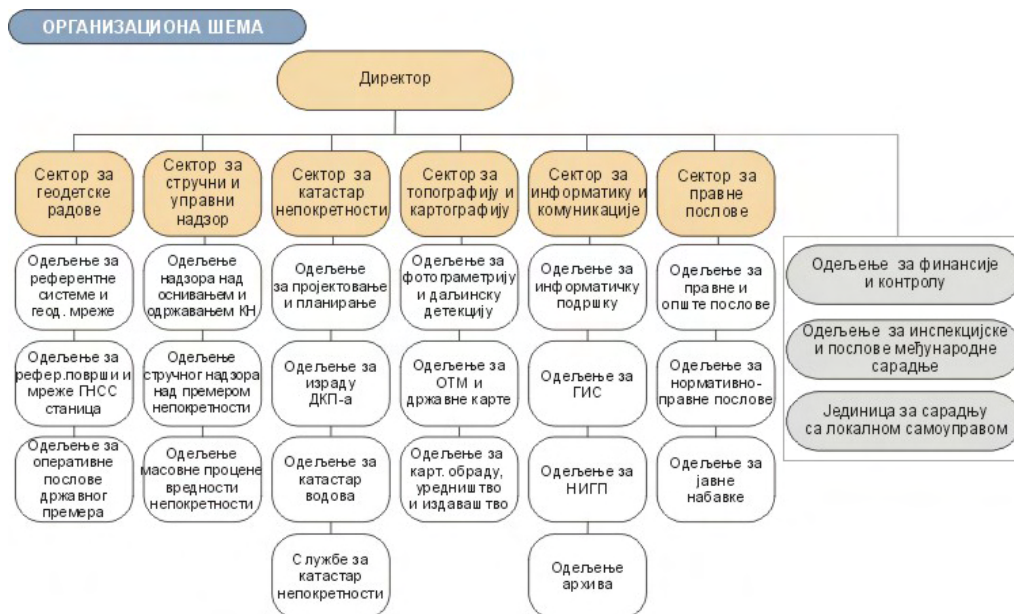
Slika 12: Model baze podataka desktop aplikacije

3.3 Organizaciona struktura Zavoda

Trenutna organizaciona struktura Zavoda [11] može se videti na slici 13. Organizacija ima oblik *drveta*. Stari sistem nije omogućavao hijerarhijsku organizaciju organizacionih jedinica te su sektori, službe i odeljenja čuvani bez informacija o pripadnosti drugim organizacionim jedinicama.

Lokacija opreme i zaposlenih evidentirala se na dva načina:

- pripadnost *Lokaciji* koja predstavlja gradove i opštine u Srbiji, na primer: Beograd, Novi Sad, Niš, Smederevo, itd. Posebno je naglašen tip lokacije koji može biti: sektor, služba, odeljenje ili grupa.
- pripadnost *Organizacionoj jedinici* u kojoj je navedena pripadnost na nivou sektora (slika 13), na primer: sektor za geodetske radove, sektor za stručni i upravni nadzor, sektor za katastar nepokretnosti, itd.



Slika 13: Organizaciona šema RGZ, decembar 2010.

3.4 Diskusija o slabostima i kvalitetima baze i aplikacija

Opisani sistem umnogome ispunjava potrebe Zavoda. Međutim, postoje delovi baze podataka i aplikacije koji nisu projektovani na najbolji način te

dolazi do različitih poteškoća u radu.

Najveći propusti napravljeni su pri samom projektovanju baze podataka. Njeno razmatranje i razumevanje je od velikog značaja pošto ona predstavlja temelj nad kojim se ceo projekat nadograđuje. Sam koncept baze nije dobro zamišljen jer ne preslikava realno stanje koje se primenjuje u praksi. Računarska oprema u Zavodu identifikuje se inventarskim brojem i njega poseduje svaka jedinica opreme kojoj je fizički moguće dodeliti broj. Dakle, svakom kućištu računara, monitoru, štampaču, skeneru, habu, softveru i sličnim komponentama dodeljuje se jedinstven inventarski broj. Međutim, u staroj bazi (slika 11) samo jedan entitet (H_RACUNAR) poseduje inventarski broj. Sve ostale komponente ne poseduju inventarski broj i vezuju se za neku jedinicu računara. Bez povezivanja perifernih komponenti sa računarom ne postoje informacije o njihovoj lokaciji, dobavljaču ili zaposlenom koji je odgovoran za tu komponentu.

Drugi problem jeste izostavljanje osobine "*jedinstvenosti*" (eng. *unique*) za atribut inventarski broj. Time se omogućava postojanje više stavki opreme sa istim inventarskim brojem. U praksi se iz različitih razloga dešavalo da za jedan inventarski broj budu unešene dve ili više jedinice opreme te je dolazilo do ponovljenih podataka. Takođe u entitetu H_RACUNAR_KOMPONENTA uveden je *veštački* primarni ključ. Kako se taj entitet koristi za povezivanje druga dva entiteta ovime je automatski omogućeno da za jedan računar i jednu komponentu postoji više upisa u bazi. Isti entitet proširen je redundantnim informacijama koje se mogu dobiti spajanjem sa drugim tabelama.

Ostatak baze podataka sa manjim izmenama i dopunama zadovoljava potrebe Zavoda.

Procedura za prijavu i servisiranje kvarova je centralizovana i njeno sprovođenje u velikoj meri zavisi od efikasnosti i ažurnosti Odeljenja za informatičku podršku. Odeljenje prima sve prijave problema nad opremom i koordinira servisere. Upravo dužnosti ovog odeljenja čine "usko grlo" u proceduri. Ne postoji direktna komunikacija prilikom prijave problema između katastarsa i servisera, što bi umnogome ubrzalo popravku opreme.

U funkcionisanju aplikacija najveći problem predstavlja neophodnost paralelnog korišćenja obe aplikacije. Tako je recimo u veb aplikaciji omogućen unos novih računara i komponenti dok je za štampanje raznih izveštaja neophodno koristiti deksktop aplikaciju. Postoje nelogičnosti u definisanim kategorijama opreme te je mrežna oprema izdvojena iz hardvera. Na kraju, u formama za prikaz podataka desktop aplikacije korisniku se prikazuju nepotrebne informacije. Na slici 14 prikazana je forma za prikazivanje podataka na kojoj se vidi *ID* kolona entiteta *Računar*. Korisniku je ta informacija ne-

potrebna pošto računare jedinstveno određuje njihov inventarski broj, koji se koristi u svakodnevnom radu.

The screenshot shows a web form titled "Ispravka računara" (Computer Repair). The form contains the following fields and options:

- Računar ID:** 1750 (highlighted with a red circle)
- Inventarski broj*:** 35375
- Naziv računara*:** FSCp351035375
- Org.jedinica*:** Sektor za informatiku i komunikacije
- Lokacija:** BEOGRAD
- Zaposleni/korisnik*:** Damir Mihajlović - Sektor za informatiku i komunikacije
- Status računara*:** 1 - Ispravan
- Dobavljač*:** SIEMENS D.O.O. BEOGRAD
- Tip računara*:** 5 - Radna stanica
- Godina instalacije*:** 2009
- Datum prijema:** 9.4.2009
- Garancija (ukucati broj meseci):** 36 (with a date field showing 9.4.2012)
- Napomena:** 01.06.2010. koristi ga Damir Mihajlović iz kancelarije 115
ugovor SAM-RECRP-YF3908-ICB-G-08-176/2
nabavka preko Svetske banke ugovor iz kraja

At the bottom, there are buttons for "Sačuvaj" (Save), "Otkazi" (Cancel), and "Dodaj" (Add).

Slika 14: Jedna od formi za prikaz podataka koja sadrži ID podatka

3.5 Definisane ciljeve

Nakon sagledavanja postojećeg stanja definisani su ciljevi koje nova aplikacija mora ispunjavati.

Osnovni zadatak je ispravljanje postojećih nedostataka. Navedeni problemi moraju biti otklonjeni što je preduslov za pouzdan rad sistema. On će biti proširen novim funkcijama, praćenjem različitih statistika i poboljšanjem prikaza i izdavanja izveštaja.

Novo rešenje bi trebalo da objedini sve funkcije u jednu veb aplikaciju,

koja će zameniti postojeće dve aplikacije. Tako će biti olakšan i unapređen rad Zavoda na polju evidentiranja i upravljanja računarskom opremom.

Takođe postavlja se zadatak da se omogući uvoz podataka iz zatečene baze u novi sistem kako bi se izbeglo ručno unošenje velike količine podataka. Iz tog razloga je poželjno da glavnica baze podataka zadrži sadašnju strukturu i oblik unosa.

Krajnji cilj jeste pravljenje multifunkcionalnih formi za prikaz podataka različitim tipovima korisnika čime će biti odvojeni podaci koje može videti zaposleni (koji prijavljuje kvar na svojoj opremi), serviser opreme ili administrator.

4 Projektovanje izmena

4.1 Entiteti *Komponenta* i *Instanca*

U opisu zatečenog stanja već je objašnjeno koje su bile uloge entiteta *Računar* i *Komponenta* u starom sistemu i koje su mane takve organizacije. Ubuduće ova dva entiteta biće spojeni u jedan jer nije potrebno razlikovati jedinice računara od ostalih komponenti.

Novi sistem mora oslikavati realno stanje koje Zavod primenjuje na polju evidentiranja računarske opreme. Evidencija opreme organizovaće se pomoću dva entiteta različite namene:

- entitet *Komponenta* predstavlja apstraktnu komponentu računarske opreme koja može postojati u Zavodu. U aplikaciji su komponente predstavljene "katalogom komponenti" iz kojeg korisnik aplikacije može po potrebi izabrati određenu komponentu i zadužiti je u Zavodu. Komponente imaju sledeće atribute: proizvođač, model, specifikacija i identifikacioni broj grupe komponenti kojoj pripada (npr. *proizvođač*: "Samsung", *Model*: "2233bw", *specifikacija*: "22' inča, rezolucija 1920x1080", *grupa*: "Monitor LCD")
- entitet *Instanca* predstavlja jedinicu računarske opreme koja postoji u Zavodu i koja je zapravo samo konkretna instanca određene apstraktne komponente. Kao takva ona predstavlja samo dopunu prethodnog entiteta nakon što se on zaduži u Zavodu po nekom kriterijumu. Instance poseduju sledeće atribute: inventarski broj, serijski broj, naziv, pripadnost drugoj instanci, zaduženost kod zaposlenog, zaduženost na određenoj lokaciji, dobavljača od kojeg je nabavljena, itd.

Prilikom korišćenja aplikacije potrebno je povremeno dopunjavati "katalog komponenti", odnosno entitet *Komponenta*, novim komponentama koje Zavod bude naručivao. Jedino obavezno polje za koje je neophodno uneti vrednost kako bi se napravio objekat komponente, jeste polje *Proizvođač*, dok polja *Model* i *Specifikacija* ne moraju imati dodeljene vrednosti. Radi identifikacije komponente i izbegavanja dupliranih zapisa, postavljena je osobina jedinstvenosti (eng. *unique*) nad skupom polja (*GrupaID*, *Proizvođač*, *Model*).

Entitet *Instanca* ima gotovo iste atribute kao i zatečeni entitet *Računar* dok će najveću promenu predstavljati dodavanje osobine jedinstvenosti (eng. *unique*) za atribut *Inventarski broj*. U praksi se iz različitih razloga dešavalo

da za jedan inventarski broj budu unešene dve ili više jedinice opreme te je dolazilo do dupliranih podataka. Međutim, postavljanjem jedinstvenog inventarskog broja biće napravljen problem za kasnije faze u kojima je potrebno uvoziti podatke iz stare u novu bazu podataka. Postojeći duplirani podaci sa dva ista inventarska broja neće moći da se unesu u novu bazu. Radi rešavanja ovog problema duplirani inventarski brojevi će se proširivati određenim specijalnim simbolima (na primer: "x1", "x2", ...).

Instance su proširene sa dva atributa koji opisuju garanciju te jedinice opreme. Pored informacije o dobavljaču od koga je oprema nabavljena, one sada poseduju informaciju o početku garancije (tipa *date*) i vremenu trajanja garancije (tipa *tinyint*) u kome se po dogovoru unosi trajanje garancije u mesecima. Dosadašnja praksa svodila se na upisivanje garancije u polju *napomene*. Ovime se olakšava posao u narednim fazama, pre svega prilikom prijave kvarova na opremi, kada će korisnik biti obaveštavan da li se jedinica opreme u tom trenutku nalazi pod garancijom.

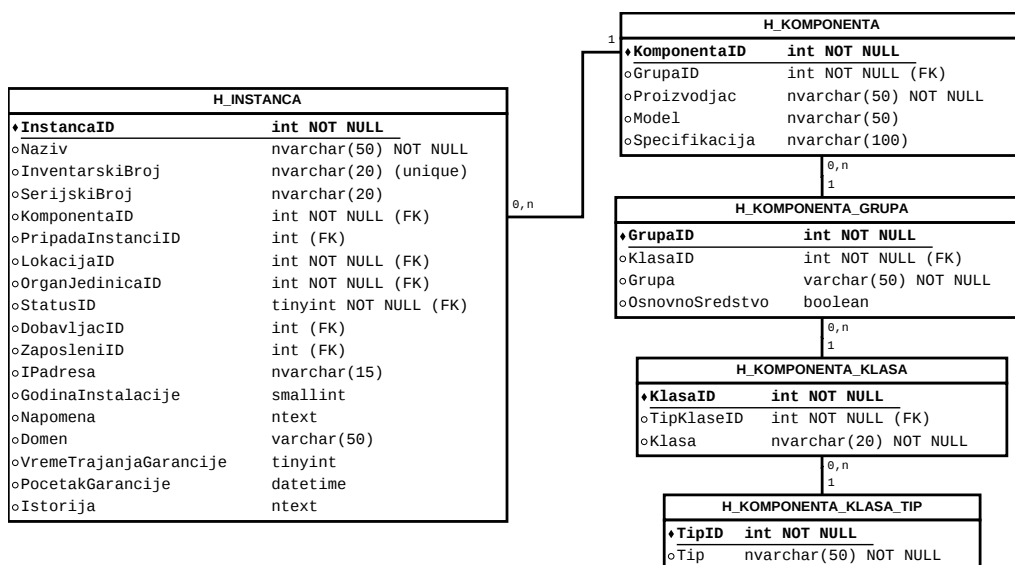
Poslednja izmena nad ovim entitetom je uvođenje podatka o *istoriji* instance (tipa *text*). Tokom vremena u njoj se automatski beleže stanja koje je instanca prošla od njenog nastanka. Korisnik nije u mogućnost da menja vrednost ovog polja. Svaka promena stanja instance se zapisuje novim slogovima formata:

Datum: 17-Sep-2011: instanci su promenjene sledeće osobine:

- promenjena lokacija iz: *121* u: *magacin*
- promenjen status iz: *Ispravno* u: *Na servisu*

Za razliku od starog sistema, ovaj način evidentiranja opreme prouzrokuje da za komponente (monitori, štampači, mrežna oprema, i sl.) postoje informacije o lokaciji na kojoj se nalaze, zaposlenom koji ih koristi, dobavljaču od koga su nabavljene i dužini garancije. Do sada to nije bio slučaj jer su sve te komponente bile vezane za određeno kućište računara od koga su zavisile. Tako će biti omogućeno lakše upravljanje navedenom opremom – premeštanje na druge lokacije ili davanje drugom zaposlenom na korišćenje.

Na dijagramu klase (slika 15) prikazan je deo nove baze podataka u kome se čuvaju podaci o instancama, komponentama i troslojnoj klasifikaciji komponenti.

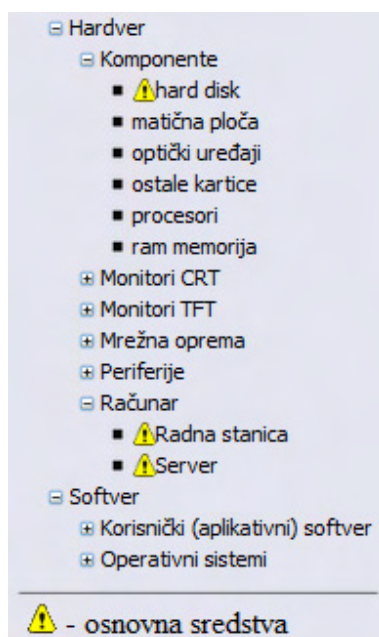


Slika 15: Klase *Instanca*, *Komponenta* i troslojna hijerarhija komponenti

4.2 Troslojna klasifikacija računarskih komponenti

Iz starog sistema biće zadržana troslojna klasifikacija komponenti uz jednu praktičnu promenu. Troslojnu klasifikaciju čini podela po *Tipovima*, *Klasama* i *Grupama* (slika 16), kao što je predstavljeno u uvodnom poglavlju.

Jedina promena jeste uvođenje pojma *osnovnih sredstava* u poslednjoj podeli po *Grupama*. Primera radi, *osnovna sredstva* jesu komponente koje pripadaju sledećim *Grupama*: CRT monitori, TFT monitori, štampači, radne stanice, serveri, svičevi, ruteri i sl., dok procesori, memorije, optički uređaji, matične ploče i slična oprema, koja obično može da se upotrebljava samo u okviru neke druge opreme, nisu *osnovna sredstva*. U praktičnoj primeni komponente od veće važnosti jesu *osnovna sredstva* i njima je najčešće fizički moguće dodeliti inventarski broj. Zavod odlučuje koje će komponente imati status osnovnog sredstva u zavisnosti od potreba. Sa slike se vidi da i tvrdi diskovi (eng. *hard disk*) mogu imati ovaj status ako u Zavodu postoji potreba za prenosnim tvrdim diskovima kao nezavisnim jedinicama sa inventarskim brojem.



Slika 16: Troslojna klasifikacija komponenti

Pored toga što je od veće važnosti da se ovim komponentama upravlja odvojeno u različitim izveštajima, svrha uvođenja *osnovnih sredstava* sastoji se i u zaduživanju samih jedinica opreme. Nakon razmatranja, a u dogovoru sa predstavnicima Zavoda, došao sam do zaključka da je za postojanje instance *osnovnog sredstva* (na primer monitora) neophodno ispuniti jedan od sledećih kriterijuma:

- posedovanje inventarskog broja ili
- zaduženost kod zaposlenog ili
- zaduženost na određenoj lokaciji

dok je za ostale instance (na primer optičkog uređaja) neophodno ispunjavanje jednog od kriterijuma:

- pripadnost nekoj instanci koja je *osnovno sredstvo* ili
- zaduženost kod zaposlenog

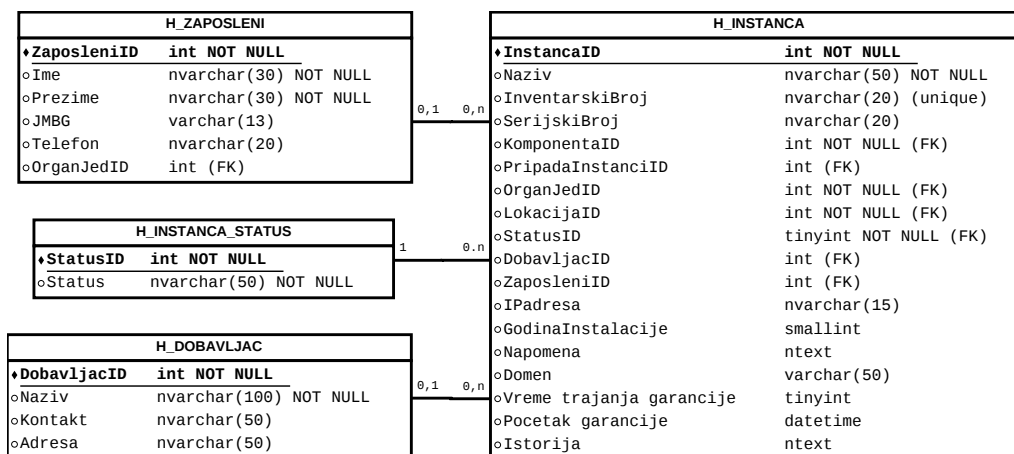
Troslojna klasifikacija u bazi podataka je organizovana pomoću tri entiteta predstavljenih na slici 15: *H_KOMPONENTA_KLASA_TIP*, *H_KOMPONENTA_KLASA* i *H_KOMPONENTA_GRUPA*.

4.3 Podaci o: *Zaposlenima, Dobavljačima i Statusima instanci*

Podaci o zaposlenima se neće mnogo razlikovati od načina na koji su čuvani u staroj aplikaciji. Pored već postojećih atributa o imenu i prezimenu zaposlenog, entitet *Zaposleni* je sada proširen informacijama o JMBG-u, kontakt informacijama i lokaciji na kojoj je lice zaposleno. Posebno je važno naglasiti da je do sada bilo nemoguće razlikovati dva zaposlena lica koja imaju isto ime i prezime i koji su zaposleni na istoj lokaciji. Uvođenjem osobine "jedinstvenosti" (eng. *unique*) za tri polja (ime, prezime, lokacija i JMBG) sprečeno je da se to ponovi. U slučaju da se pojave dva lica sa istim imenom i prezimenom na istoj lokaciji, neophodno je uneti JMBG jednog od njih, ili u imenu napisati dodatne informacije (npr. ime jednog roditelja zaposlenog lica).

Čuvanje podataka o dobavljačima služi pre svega pri servisiranju računarske opreme u garantnom roku. Zato je pored imena dobavljača potrebno čuvati kontakt podatke (telefon, email i sl.) i njegovu adresu. To su ujedno i svi atributi koje poseduje ovaj entitet.

U bazi podataka takođe čuvamo podatke o statusu koje određena instanca može imati, i on može imati sledeće vrednosti: ispravno, delimično ispravno, na servisu, na terenu i neispravno. Administratori aplikacije su u mogućnosti da dodaju nove ili izmene postojeće statuse. Stari sistem posedovao je entitete *H_RACUNAR_STATUS* i *H_KOMPONENTA_STATUS* te će oni sada biti objedinjeni jednim entitetom *H_STATUS*.

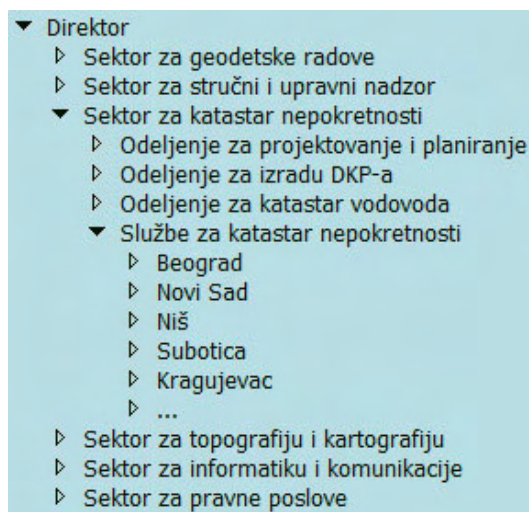


Slika 17: Povezanost entiteta *Zaposleni*, *Dobavljači* i *Status instanci* sa entitetom *Instanca*

4.4 Nova organizaciona struktura

Radi lakšeg predstavljanja i razumljivije organizacije dizajnirana je nova struktura organizacionih jedinica u obliku stabla koja objedinjuje dosadašnja dva načina evidentiranja pripadnosti opreme. U prvobitnoj bazi podataka podaci o katastrima su čuvani u posebnom entitetu H_LOKACIJA. U praksi svi katastri pripadaju "Službi za katastar nepokretnosti" (slika 13). Ta veza je iskorišćena za povezivanje dosadašnje podele, pa su tako sve *Lokacije* u novom sistemu našle mesto u "Službi za katastar nepokretnosti" koja je dodata kao novi nivo u hijerarhiji.

Nova hijerarhija *Organizacionih jedinica* i način na koji je ona predstavljena u novoj bazi podataka može se videti na slikama 18 i 19.



Slika 18: Hijerarhija organizacionih jedinica

H_ORGAN_JED	
♦OrganJedID	int NOT NULL
◦Naziv	nvarchar(100) NOT NULL
◦PripadaID	int (FK)

Slika 19: Predstavljanje organizacionih jedinica u bazi podataka

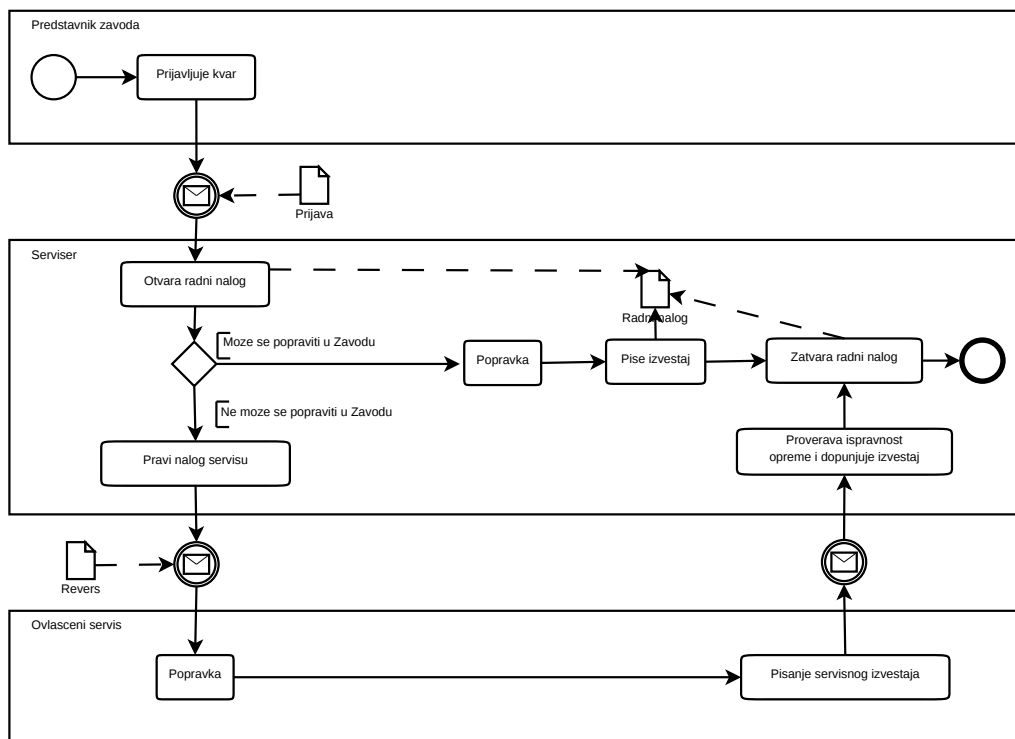
Drugi korak u reorganizaciji pripadnosti ogledava se u izmenjenoj nameni entiteta *Lokacije*. Umesto dosadašnjih podataka o lokacijama katastara (Beograd, Novi Sad, Niš, ...) u njemu će biti smešteni nazivi kancelarija, magacina i ostalih prostorija u kojima je smeštena oprema. Vrednosti se mogu unositi i ažurirati iz administracionog panela. Razlog uvođenja ove promene jeste

pretraživanje komponenti. U novom sistemu biće omogućeno da se za organizacionu jedinicu izabere "SKN¹ Beograd" a za lokaciju broj kancelarije "112", i na taj način se mogu prikazati podaci o opremi koja pripada tačno određenoj lokaciji u Zavodu.

4.5 Prijava i servis kvarova

Za potrebe celine koja se odnosi na prijavu i servisiranje kvarova koriste se dva glavna entiteta *Prijava* i *Intervencija*. Pored njih koriste se pomoćni entiteti *Vrsta intervencije* i *Ttp vrste intervencije* kojima se klasifikuju intervencije po određenom kriterijumu.

Procedura prijave i servisiranja nije mnogo izmenjena u odnosu na stari sistem već su samo dodate određene olakšice. Subjekti i njihovi poslovi mogu se videti na dijagramu saradnje (slika 20).



Slika 20: Dijagram saradnje: Tok prijave i servisiranja kvarova

¹Služba za katastar nepokretnosti

Prijava kvarova može se vršiti na dva načina:

- telefonskim putem obaveštavanjem centrale Zavoda, što je jedini način prijave u starom sistemu
- novom veb aplikacijom od strane predstavnika katastarskih službi ²

Nakon prijave problema, centrala ili predstavnik katastarske službe pravi formalnu *Prijavu* u sistemu sa sledećim informacijama: podaci o jedinici opreme nad kojoj se prijavljuje problem, datum prijave, ime i prezime osobe koja je prijavila problem, opis problema, organizaciona jedinica i lokacija na kojoj se oprema nalazi, kontakt telefon i hitnost rešavanja problema. Već opisanim uvođenjem podatka o garanciji na nivou instance, automatski se u opisu problema prilaže informacija koja govori da li je instanca trenutno pokrivena garancijom, i ako jeste, za koliko dana garancija ističe.

Na osnovu *Prijave* serviser u Zavodu pravi i štampa *Radni nalog* koji sadrži početne informacije o zaposlenom, odnosno serviseru, koji je preuzeo odgovornost nad zadatom prijavom i datumu pravljenja *Radnog naloga*. Nakon toga serviser je odgovoran za dalje postupke sa problematičnom jedinicom opreme. Mogući scenariji su otklanjanje kvara od strane serviseru u samom Zavodu, konstatovanje da je iz nekog razloga nemoguće popraviti opremu nakon čega sledi njeno rashodavanje i, ako je oprema pokrivena garancijom, prosleđivanje opreme ovlašćenom serviseru. U svakom od scenarija serviser je dužan da u *Radnom nalogu* dopuni izveštaj, naznači vrstu intervencije i obrazloži postupak koji je preduzeo. Ono što neće biti menjano u odnosu na stari sistem je klasifikacija intervencija pošto one zadovoljavaju sadašnje potrebe i nije neophodno menjati već dobro ustaljene navike.

U slučaju da je potrebno opremu poslati u ovlašćeni servis, serviser pokreće proceduru pravljenja naloga servisu. Ona se ogleda u popunjavanju *Reversa* koji potpisuje i sam ovlašćeni servis kako bi se potvrdilo da je oprema preuzeta. Ovlašćeni servis vraća opremu sa servisnim izveštajem koji se dopunjava u *Radnom nalogu*. Na kraju serviser sa izveštajem ovlašćenog servisa konačno zatvara *Radni nalog* još jednom dopunom izveštaja. Rezultat je popravljena ili rashodavana jedinica opreme.

²Detaljnije o *predstavnicima katastarskih službi* u poglavlju koje se odnosi na korisnike aplikacije

4.6 Korisnici aplikacije i forme za prikazivanje podataka

Za razliku od starog sistema koji je predviđao samo jedan tip korisnika aplikacije, pri projektovanju novog sistema predviđena su tri različita tipa korisnika aplikacije. U zavisnosti od njihovih ovlašćenja prikazivaće im se različite forme za prikaz podataka i funkcija. To su sledeći tipovi korisnika:

- **Administrator.** Ima najveća ovlašćenja nad sistemom. Može dodavati, ažurirati i brisati podatke iz svih entiteta. Poželjno je postojanje samo jednog administratora aplikacije zbog sigurnosti sistema. Pošto je ceo sistem centralizovan, administrator jedini poseduje sledeće mogućnosti:
 - pravljenje novih i ažuriranje postojećih jedinica računarske opreme
 - dopunjavanje "kataloga komponenti"
 - menjanje organizacione strukture Zavoda
 - menjanje troslojne klasifikacije komponenti i osnovnih sredstava
 - dodavanje i ažuriranje informacija o zaposlenima
 - dodavanje i ažuriranje informacija o dobavljačima
 - dodavanje i ažuriranje statusa instanci
 - dodavanje i ažuriranje statusa i hitnosti rešavanja prijave
 - dodavanje i ažuriranje novih korisnika aplikacije
- **Serviser.** Određen broj zaposlenih u Zavodu bavi se servisiranjem opreme. Njima su omogućene osnovne funkcije upravljanja entitetima *Prijava* i *Intervencija* dok ostale podatke nisu u mogućnosti da koriste. Kao što je opisano u odeljku za prijavu i servis kvarova, poslovi serviseru sastoje se u sledećem:
 - Pregledavanje svih *Prijava* kvarova. Mogućnost odabira određene *Prijave* čiji problem želi razrešiti. U tom slučaju *Status prijave* se menja iz "prijavljano" u status "na servisu" i automatski se pravi i štampa *Radni nalog* serviseru³.
 - Pravljenje *Radnog naloga* odgovara pravljenju novog podatka u tabeli *Intervencija*. Serviser tada proverava da li je on u mogućnosti da otkloni kvar ili je potrebno da se jedinica opreme prosledi u ovlašćeni

³formular "Radni nalog za intervenciju nad opremom" u odeljku *Dodaci*

servis. U zavisnosti od toga na *Radnom nalogu* beleži *Vrstu intervencije* i počinje pisati *Izveštaj*.

- Ako je potrebno da se određena oprema popravi u servisu, vrši se *Pravljenje naloga servisu*⁴ i na *Radnom nalogu* se označava da je oprema poslata u servis. Pri povratku opreme iz servisa na *Radnom nalogu* se popunjava *Servisni izveštaj*, a *Status prijave* se može promeniti na "Završeno" a *Status instance* na "Ispravno" ili "Neispravno" u zavisnosti od rezultata.

- **Predstavnik katastarske službe.** Predstavlja drugi novouvedeni tip korisnika aplikacije koji bi trebalo da olakša dosadašnju proceduru prijave i servisiranja kvarova. Novi sistem predviđa da svaka katastarska služba izabere jednog predstavnika, koji će moći da prijavi problem na opremi kojom taj katastar raspolaže. Pri novoj prijavi automatski se šalje poruka serviseru putem elektronske pošte da je pristigla nova prijava čime on može otpočeti opisanu proceduru servisiranja opreme. Osnovne funkcije kojima ovaj tip korisnika raspolaže su:

- Pregledanje opreme kojom raspolaže katastarska služba u kojoj je zaposlen. Zbog povezanosti entiteta *Korisnika* i *Zaposlenih* prostim filtriranjem opreme po lokaciji zaposlenog lako se može dobiti oprema koju ta katastarska služba poseduje.

- Pravljenje i pregledanje *Prijave* kvarova nad opremom kojom raspolaže.

- Pregledanje *Radnih naloga* ako su isti napravljeni na osnovu njegovih prijava ili nekog drugog korisnika iz te organizacione jedinice. U njemu se može informisati o trenutnom stanju prijavljene opreme.

⁴formular "Revers serviseru za preuzimanje opreme" u odeljku *Dodaci*

5 Implementacija

5.1 Primenjene tehnologije

5.1.1 Platforma

Nova aplikacija za evidenciju računarske opreme implementirana je za platformu *ASP.NET*. Platforma *ASP.NET* predstavlja okvir za razvoj veb aplikacija (eng. *web application framework*), koja podržava razvoj dinamičkih veb stranica, aplikacija i servisa. Razvijena je od strane *Microsoft*-a januara 2002. godine, kada je javnosti postala dostupna prva verzija 1.0 *.NET* platforme. Njena preteča je prva tehnologija kompanije *Microsoft* za dinamički generisane veb stranice, nazvane *ASP* (eng. *Active Server Pages*).

Glavna odlika ove platforme je razdvajanje veb stranice, koja se prikazuje korisniku u pretraživaču, i izvornog koda, koji je vezan za odgovarajuću veb stranicu. Veb stranice se označavaju ekstenzijama *.aspx*. Ranije se dinamički izvorni kod, koji se izvršava na serveru, nalazio unutar veb stranica u posebno označenom bloku `< % - dynamic code - % >`, slično drugim skript jezicima, kao što su *PHP*, *JSP* i *ASP*. Počevši od verzije 2.0 platforme *ASP.NET* *Microsoft* je uveo novi model koji je podrazumevao razdvajanje statičnog sadržaja stranica od dinamičkog koda. Svakoј veb stranici, čija je uobičajena ekstenzija *.aspx*, odgovara dodatni programski modul sa ekstenzijom *.aspx.cs* ili *.aspx.vb* zavisno od programskog jezika.

5.1.2 Programski jezik

Aplikacija je razvijena na programskom jeziku *C#*. Naziv je inspirisan muzičkom notacijom gde simbol *#* ispred note označava da ton te note treba biti povišen. Prethodno je postojala ideja da se ovaj jezik nazove *Cool* što bi trebalo da predstavlja skraćenicu za *C-like Object Oriented Language*. Razvijen je od strane *Microsoft*-a i prva zvanična verzija ovog programskog jezika predstavljena je u januaru 2002. godine.

C# je zamišljen kao jednostavan i moderan, objektno orijentisan i strogo tipiziran programski jezik opšte namene. Podržava i ostale paradigme kao što su strukturna, imperativna i funkcionalna paradigma. Oslobođanje memorije se vrši automatskim "sakupljačem otpadaka" (eng. *garbage collection*). Izvorni kod je veoma sličan izvornim kodovima programskih jezika *C*, *C++* i *Java*, kako bi programerima olakšalo učenje ovog programskog jezika.

5.1.3 Tehnologija *Ajax*

Ajax (eng. *Asynchronous JavaScript and XML*) [12] je složena veb tehnologija i čine je sledeći elementi:

- *HTML* i *CSS* za prikazivanje sadržaja u pretraživaču,
- *DOM* (eng. *Document Object Model*) za dinamičko menjanje sadržaja,
- *XML* za razmenu podataka između pretraživača i servera,
- *JavaScript* za povezivanje svih tehnologija.

Koristi se isključivo u pretraživaču na klijentskoj strani kako bi se povećala interakcija veb stranica i korisnika. Pomoću ove tehnologije veb aplikacija je u mogućnosti da šalje i prima podatke serveru bez potrebe za ponovno učitavanje čitavog sadržaja veb stranice. To je ujedno i glavna odlika koja *Ajax* čini popularnim. U novoj aplikaciji *Ajax* je korišćen kako bi se olakšalo popunjavanje određenih formulara, koji čine veliki deo interakcije sa korisnicima.

5.1.4 Baza podataka

Baza podataka informacionog sistema je implementirana na sistemu za upravljanje relacionim bazama podataka *Microsoft SQL Server*. Postoji u više izdanja, koje su namenjene različitim profilima korisnika: *Express*, *Workgroup*, *Web*, *Standard*, *Enterprise* i *Datacenter*. Za implementaciju baze podataka sistema za upravljanje računarskom opremom, korišćena je verzija *Express*.

5.1.5 Razvojno okruženje

Aplikacija je razvijena pomoću razvojnog okruženja *Microsoft Visual Studio* čija je namena razvoj konzolnih i *GUI* (eng. *graphical user interface*) aplikacija, veb aplikacija i veb servisa. Podržava više programskih jezika među kojima su *C*, *C++*, *C#*, *Visual Basic* i *F#*. Podršku za programske jezike kao što su *Python* i *Ruby* moguće je naknadno instalirati. Pomoću ovog alata moguće je razviti aplikacije za sledeće platforme: *Microsoft Windows*, *Windows Mobile*, *.NET Framework*, *Microsoft Silverlight* i druge.

Visual Studio postoji u više izdanja sa različitim osobinama: *Express*, *Professional*, *Premium*, *Ultimate* i *Test Professional*. Neizbežne celine sa

kojima se korisnik susreće u ovom alatu jeste deo za pisanje izvornog koda (eng. *Code Editor*) i za dizajn veb stranica (eng. *Designer*). U najvećoj meri su dobro prilagođeni korisnicima. Baza podataka se može dizajnirati pomoću alata *Class designer*. Razvojno okruženje poseduje i alat za pronalaženje i otklanjanje grešaka tokom rada aplikacije (eng. *Debugger*).

5.2 Komponenta za povezivanje sa bazom podataka

LinQ (eng. *Language Integrated Query*) [13] je komponenta *Microsoft .NET* platforme koja omogućava postavljanje upita nad bazom podataka. Može se koristiti u više programskih jezika koji se koriste u okviru ove platforme, počevši od verzije 3.5. Dobavljene podatke iz baze podataka *LinQ* može transformisati u podatke pogodne za obradu u nizovima, klasama, *XML* strukturama ili ostalim strukturama za skladištenje podataka.

Osnovne dve klase koje su potrebne za funkcionisanje jesu *DataContext* i *LinQDataSource*. Klasa *DataContext* poseduje metode za osnovno komuniciranje sa bazom podataka - *Dispose*, *ExecuteQuery*, *InsertOnSubmit*, *SubmitChanges*, i druge. *LinQ* takođe pravi klase za svaku tabelu posebno sa atributima koja ta tabela poseduje. Na kraju komponenta *LinQDataSource*, povezujući se pomoću *DataContext*-a na bazu podataka, preuzima podatke i transformiše ih u format pogodan za predstavljanje u veb komponentama za prikaz podataka.

Komponenta koja je prethodila korišćenju nove komponente *LinQDataSource* jeste *SqlDataSource*. U implementaciji novog sistema, korišćena je zbog lakoće postavljanja složenih ili ugnježđenih upita. Sledi primer korišćenja komponente *SqlDataSource* i postavljanja nešto složenijeg upita nad bazom podataka:

```
<asp:SqlDataSource ID="sqlKlasaTipDataSource" runat="server"
    ConnectionString="<%= ConnectionStrings:
    DatabaseConnectionString %>"
    SelectCommand="SELECT '-- Izaberi tip --' as TipKlase,
                    null as TipKlaseID
                    union
                    SELECT H_KOMPONENTA_KLASA_TIP.TipKlase,
                    H_KOMPONENTA_KLASA_TIP.TipKlaseID FROM
                    H_KOMPONENTA_KLASA_TIP">
</asp:SqlDataSource>
```

Naredni primer predstavlja osnovno korišćenje komponente *LinqDataSource*. U njemu se može videti atribut *TableName* koji referiše na tabelu H.DOBAVLJAC. Na taj način se bez dodatnih podešavanja i obrade mogu preuzeti podaci iz navedene tabele.

```
<asp:LinqDataSource ID="DobavljaciDataSource" runat="server"
    ContextTypeName="DataClassesDataContext"
    EnableDelete="True" EnableUpdate="True"
    EntityTypeName="" TableName="H_DOBAVLJACs">
</asp:LinqDataSource>
```

Omogućeno je sortiranje podataka. U ovom primeru sortiranje se vrši redom po grupi, proizvođaču i modelu komponente računarske opreme.

```
<asp:LinqDataSource ID="KomponenteDataSource" runat="server"
    ContextTypeName="DataClassesDataContext" EntityTypeName=""
    OrderBy="H_KOMPONENTA_GRUPA.Grupa, Proizvodjac, Model"
    TableName="H_KOMPONENTAs" EnableDelete="True"
    EnableUpdate="True">
</asp:LinqDataSource>
```

Ako postoji potreba za filtriranjem podataka ono se može obaviti naznačavanjem nad kojom kolonom se vrši filtriranje u atributu *Where* i uvođenjem odeljka *WhereParameters* unutar komponente *LinqDataSource*. Tu su navedene komponente od kojih zavisi filtriranje. Sledeći primer filtrira podatke o komponentama računarske opreme koje pripadaju određenoj grupi komponentata. Podaci će se prikazati u zavisnosti od izabrane vrednosti u *DropDownList* komponenti sa nazivom *ddlGrupaFiltriranje*.

```
<asp:LinqDataSource ID="KomponenteFilterGrupaDataSource"
    runat="server" ContextTypeName="DataClassesDataContext"
    EnableDelete="True" EnableUpdate="True" EntityTypeName=""
    TableName="H_KOMPONENTAs" Where="GrupaID == @GrupaID">
    <WhereParameters>
        <asp:ControlParameter ControlID="ddlGrupaFiltriranje"
            Name="GrupaID" PropertyName="SelectedValue"
            Type="Int32" />
    </WhereParameters>
</asp:LinqDataSource>
```

5.3 Forme za prikaz podataka

Jedna od najvažnijih celina nove aplikacije je celina za upravljanje računarskom opremom i prijavljivanje kvarova na opremi. Radi ilustracije formi za prikaz i upravljanje podataka, na slici 21 je prikazan deo stranice za prikaz podataka jedinice računarske opreme.

Detaljni podaci o instanci:

Naziv	Novi skener
Inventarski broj	3212
Serijski broj	
Proizvođač/Model	Plustek / Be@rPaw
Pripada Instanci	-
Organizaciona jedinica	SKN Beograd
Lokacija	
Status	Ispravno
Dobavljač	Gigatron
Početak garancije	9/7/2011 0:00:00 AM
Vreme trajanja garancije (u mesecima)	12
Zaposleni	
IPadresa	
Godina Instalacije	
Domen	

Istorija komponente:

Instanca kreirana: 09-Sep-2011

Datum: 09-Sep-2011
Instanci su promenjene sledeće osobine:
- promenjena **organ. jed.** iz: / u: SKN Beograd

Datum: 09-Sep-2011
Instanci su promenjene sledeće osobine:
- promenjen **dobavljač** iz: / u: Gigatron

Datum: 09-Sep-2011
Instanci su promenjene sledeće osobine:
- promenjen **početak kagancije** iz: 09-Sep-2011 u: 07-Sep-2011
- promenjen **vreme trajanja garancije** iz: u: 12

Datum: 11-Oct-2011 Instanci su promenjene sledeće osobine:
- promenjen **naziv** iz: nova u: Novi skener

Slika 21: Forma za prikazivanje podataka o jedinici računarske opreme

Sa slike se mogu videti podaci o jedinici računarske opreme. Prikazana jedinica je skener koji u Zavodu ima status osnovnog sredstva i jednoznačno se identifikuje inventarskim brojem. Posедуje informacije o pripadnosti organizacionoj jedinici, kupljena je ili nabavljena kod konkretnog dobavljača i poseduje informacije o garanciji. Na desnoj strani nalaze se informacije o kompletnoj istoriji stanja komponente. Pri pravljenju instance prvi zapis beleži datum pravljenja, dok se ostali zapisi odnose na promene stanja.

Pretraga instanci:

Po inventarskom broju:

Po serijskom broju:

Po zaposlenima:

Status:

☒ samo osnovna sredstva

Po organizacionim jedinicama: ☐ Direktor

- ☐ Sektor za geodetske radove
- ☐ Sektor za informatiku i komunikacije
- ☐ Sektor za katastar nepokretnosti
- ☐ Sektor za pravne poslove
- ☐ Sektor za strucni i upravni nadzor
- ☐ Sektor za topografiju i kartografiju

Po lokacijama:

Filteri: zaposleni: Aca Stojanović; status: Ispravno; samo osnovna sredstva;

		Naziv	InventarskiBroj	Vrsta komponente	Status	Zaposleni
Detaljno	✗			Maxtor, 360GB	Ispravno	Aca Stojanović
Detaljno	✗	Acin monitor	678	Samsung, SyncMaster 310	Ispravno	Aca Stojanović

Slika 22: Forma za pretragu jedinica računarske opreme

Slika 22 predstavlja osnovni formular za pretraživanje jedinica računarske opreme. Koristi se u više slučajeva među kojima su osnovna pretraga, prijavljivanje kvarova ili pronalaženje jedinice osnovnog sredstva kako bi se druga oprema povezala sa istom.

Ponuđena je pretraga po konkretnom inventarskom ili serijskom broju. Pretraga po ovim kriterijumima poništava filtere koji su kreirani po ostalim atributima. Drugi deo pretrage, na formi u delu ispod horizontalne linije, koristi se za pravljenje filtera po više kriterijuma. U konkretnom primeru sa slike 22 napravljen je filter u kojem su izlistane sve ispravne jedinice računarske opreme koje imaju status osnovnog sredstva i zadužene su kod zaposlenog Ace Stojanovića. Moguće je filtrirati prikazane podatke i po organizacionim jedinicama i lokacijama (kancelarija, magacin i slično) na kojima se oprema može nalaziti.

Podaci o novoj prijavi:

Instanca nad kojom je prijavljen problem: **inventarski broj: 777**
vrsta komponente: [Radna stanica] CoolerMaster /
Garancija: Instanca nije bila pod garancijom.

Hitnost prijave:

Status prijave:

Datum prijave: ☐ Izaberi drugi datum

Ime i prezime osobe koja je prijavila problem:

Organ. jed. na kojoj se oprema nalazi: ☐ Izaberi drugu organ. jed.

Lokacije na kojoj se oprema nalazi: ☐ Izaberi drugu lokaciju

Kontakt telefon:

Opis problema:

Instanca nije bila pod garancijom.

Slika 23: Forma za prijavljivanje problema za jedinicu računarske opreme

Forma za prijavljivanje problema (slika 23) prikazuje se nakon što se u već opisanoj formi za pretragu jedinica računarske opreme (slika 22) izabere jedinica na kojoj se želi prijaviti problem. U ovoj formi kopiraju se osnovni podaci o jedinici opreme: vrsta komponente, inventarski i serijski broj. Podrazumevana vrednosti za hitnost prijave je *Hitno*, dok je za status prijave podrazumevana vrednost *Aktuelno*. Datum prijave sadrži dan kada je prijavljen problem ali se može promeniti ako je to potrebno. U opisu problema automatski se upisuje informacija o garanciji te jedinice, ako je poseduje, ili informacija da jedinica nikad nije ni bila pod garancijom. Pripadnost organizacionoj jedinici i lokaciji se takođe automatski upisuje ako ta jedinica poseduje te informacije, inače ostaje otvorena da se prilikom prijave izmeni. Na kraju, korisniku je ostavljeno da dopopuni opis problema i kontakt telefon pre nego što potvrdi prijavu.

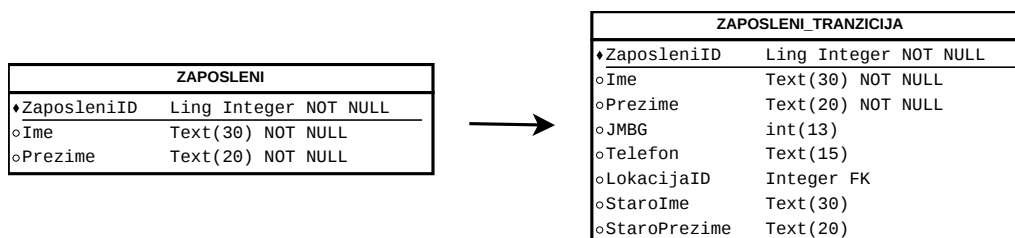
5.4 Tranzicija

Nakon završene implementacije novog sistema najvažniji i najosetljiviji korak predstavlja prebacivanje podataka iz stare baze podataka u novi sistem. Sadržaj većine novih tabela može se uvesti iz odgovarajućih tabela preslikavanjem "jedan prema jedan", dok je kod pojedinih potrebno spajanje dve tabele.

Za potrebe tranzicije tabele u novoj bazi podataka proširuju se pomoćnim atributima koje je odgovarajuća tabela posedovala u starom sistemu. Prva faza tranzicije podrazumeva prenos podataka u navedene pomoćne attribute kako bi se podaci proverili i po potrebi ažurirali. U drugoj fazi podaci se prebacuju iz pomoćnih u ostale attribute novih tabela koji će biti korišćeni u aplikacijama. Na kraju preostaje da se izmeni dizajn tabela izostavljanjem pomoćnih atributa.

Jedan od razloga ovog postupka je dodatna provera podataka pre konačnog potvrđivanja njihove ispravnosti. Na taj način, u novu tabelu koja čuva informacije o instancama računarske opreme, mogu se uneti stari podaci sa dupliranim inventarskim brojevima koji će naknadno biti proveravani i ažurirani pre konačnog potvrđivanja podataka. U suprotnom duplirani inventarski brojevi ne bi mogli da se prenesu u nove attribute zbog osobine jedinstvenosti.

Na slici 24 prikazan je primer tabele u kojoj se smeštaju informacije o zaposlenim licima. U staroj bazi podataka ona je posedovala samo attribute *ime* i *prezime*. Tokom tranzicije, pored novih informacija o *JMBG-u*, *kontakt telefonu* i *pripadnosti lokaciji*, tabela će posedovati attribute *staroIme* i *staroPrezime*. U mnogim tabelama tokom tranzicije poželjno je dopisati još informacija koje su naknadno dodate u dizajnu tabela.



Slika 24: Tabela *Zaposleni* tokom tranzicije podataka

U tabeli 1 navedene su tabele u novoj bazi podataka, načinu na koji su uvezeni podaci iz stare baze podataka i proširenja tabela novim atributima.

Tabela	Način uvoženja podataka	Dodatna proširenja
Zaposleni	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	Dopuniti sledećim podacima: JMBG, telefon i pripadnost lokaciji
Dobavljač	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	Dopuniti sledećim podacima: kontakt i adresa
Lokacija	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele izostavljajući podatke o tipu lokacije	
Organizaciona Jedinica	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	Dopuniti sledećim podacima: pripadnost drugoj organizacionoj jedinici
Status instance	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	
Tip komponente	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	
Klasa komponente	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	
Grupa komponente	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	Dopuniti sledećim podacima: osnovno sredstvo
Komponenta	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele izostavljajući podatke o indeksu i nazivu	Dopuniti sledećim podacima: proizvođač, model i specifikacija
Instanca	Spajanje podataka iz tabela Računar i Računar_Komponenta po atributu <i>RacunarID</i>	Dopuniti sledećim podacima: pripadnost drugoj instanci, početku garancije, vremenu trajanja garancije i istoriji

Tabela	Način uvoženja podataka	Dodatna proširenja
Prijava	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele izostavljajući podatke o inventarskom broju, serviseru i izveštaju	Dopuniti sledećim podacima: kontakt telefon
Status prijave	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	
Hitnost rešavanja	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	
Intervencija	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	Dopuniti sledećim podacima: serviser, izveštaj serviseru i izveštaj ovlašćenog servisa
Vrsta intervencije	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	
Tip vrste intervencije	Preslikavanje podataka <i>jedan prema jedan</i> iz stare tabele	

Tabela 1: Način uvoženja podataka i dodatna proširenja tabela u novoj bazi podataka

6 Zaključak

Opisanim unapređenjem informacionog sistema za evidenciju računarske opreme povećavaju se njegove mogućnosti i olakšava se rad u određenim segmentima. Za razliku od prethodnog sistema otklonjeni su nedostaci koji su se javljali prilikom upravljanja računarskom opremom a koji se odnose na: (1) opremu sa dupliranim inventarskim brojevima, (2) komponente koje ne poseduju informacije o pripadnosti lokacijama i zaposlenima, kao i (3) nemogućnost posmatranja komponente kao nezavisne celine u odnosu na računar. Procedura za prijavu i servisiranje kvarova je uprošćena uvođenjem dva nova tipa korisnika aplikacije - serviseri i predstavnika karastarske službe. Time se olakšava posao zaposlenog lica koji je bio zadužen za primanje prijave telefonskim putem, a samim tim se i decentralizuje organizacija pravljenjem naloga serviserima u više regionalnih centara. Olakšano je i praćenje istorije kretanja i promene stanja opreme. Sada se istorija automatski beleži prilikom svake promene stanja i korisnik je oslobođen ikakve obaveze zapisivanja istorije. Olakšana je pretraga opreme novim filterima po organizacionim jedinicama, lokacijama i zaposlenima. Time se detaljnije može pratiti statistika prijavljenih problema nad opremom. Ostavljena je fleksibilnost pri eventualnoj izmeni organizacione strukture Zavoda, podrazumevajući da se svaka buduća organizacija može predstaviti u obliku stabla.

Nakon svake implementirane celine aplikacija je testirana od strane predstavnika Zavoda. Ipak, očekujem da se pri praktičnom korišćenju aplikacije uvidi gde eventualno postoji prostor za dalje unapređenje. Pre svega se to odnosi na prilagođavanje dizajna korisničkog interfejsa željama samih korisnika. U budućnosti može se povećati bezbednost aplikacije uvođenjem *SSL* protokola [14] za enkripciju podataka. Time bi sva komunikacija između korisnika i veb aplikacije bila šifrovana što bi u velikoj meri onemogućilo trećem licu pristup informacijama koje se razmenjuju.

7 Dodaci

7.1 Dodatak A - Formulari

RAČUNARSKA OPREMA¹

Org. jedinica*: _____
Broj kancelarije*: _____ Korisnik*: _____

1. Oprema

Naziv računara*:	Domen:	Inventarski broj* ² :	Godina instalacije:

Dobavljač*:	
-------------	--

Status računara*:	1) Ispravan 2) Neispravan 3) Delimično ispravan 4) Na servisu
-------------------	---

Monitor*:	1) 14" 2) 15" 3) 17" 4) 19" 15) 21"	
Procesor*:		MHz*: _____
Matična ploča:		
RAM*:		
Grafička kartica:		
Modem:		Provajder: _____

Tastatura:	1) DIN 2) PS/2
Miš:	1) COM 2) PS/2 3) USB

	Proizvođač:	Veličina:
HD 1:		
HD 2:		
HD 3:		
HD 4:		

CD/DVD	Tip:	Model:
-	1) R 2) RW	

LAN (tip):	Fizička adresa:	IP adresa:

2. Periferijska oprema

	Model:	Inventarski broj:	Status ³ :
Štampač:			
Skener:			
Digitajzer:			
Ploter:			

1 Obrazac se popunjava upisivanjem ili zaokruživanjem odgovarajućeg podatka. Ukoliko se podatak ne nalazi u spisku upisati ga na liniji. Obavezni podaci su označeni zvezdicom*.

2 Upisati i inventarski broj monitora.

3 1) Ispravan 2) Neispravan 3) Delimično ispravan 4) Na servisu

3. Operativni sistemi⁴

1.	Windows 95	_____	8.	Windows 2000 Server	_____
2.	Windows 98	_____	9.	Windows 2000 Advanced Server	_____
3.	Windows NT 4.0	_____	10.	Windows 2003 Server	_____
4.	Windows 2000	_____	11.	Novel	_____
5.	Windows XP Professional ed.	_____	12.	_____	_____
6.	Windows XP Home ed.	_____	13.	_____	_____
7.	Windows NT 4.0 Server	_____	14.	_____	_____

4. Aplikativni softver⁵

1.	MS Office	10.	Visual Studio	19.	_____
2.	Auto CAD	11.	ErWin	20.	_____
3.	Auto Desk Map	12.	Visio	21.	_____
4.	Microstation	13.	WinZip	22.	_____
5.	MapSoft	14.	Acrobat Reader	23.	_____
6.	Geomedia	15.	Pisarnica	24.	_____
7.	Adome Ilustrator	16.	GeodIS-NET	25.	_____
8.	Corel	17.	GeodIS-Tahimetrija	26.	_____
9.	Photoshop	18.	GeodIS-Orto	27.	_____

5. Mrežna oprema: UTP/BNC

Vrsta uređaja	Proizvođač i model	Br. Portova					Inv./Serijski broj	Status ⁶
Router	_____						_____	_____
Switch	_____	4	8	16	24	48	_____	_____
HUB	_____	4	8	16	24	48	_____	_____
Patch panel	_____	4	8	16	24	48	_____	_____
Firewall	_____						_____	_____
RACK (Unit)	_____						_____	_____
UPS (VA)	_____	_____ VA					_____	_____

6. Administracija⁷

1. _____
2. _____
3. _____

7. Naopmena

Datum: _____

Podatke prikupio: _____

4 Upisati verziju Service Pack-a

5 Upisati verziju

6 1) Ispravan 2) Neispravan 3) Delimično ispravan 4) Na servisu

7 Upisati lokalne administratore

Slika 27: Formular za prijavu problema na računarskoj opremi

RADNI NALOG ZA INTERVENCIJU

Broj prijave: 9515

Datum prijave: 28.07.2011.

Izvršilac: Andrija Nikolić

Prijavio problem: Aco Markešević

Lokacija: 0

Sprat:

Hitnos rešavanja: Nije hitno

Opis problema:

Izveštaj:

Datum početka
intervencije:

Datum završetka
intervencije:

Dužina trajanja
intervencije (u minutima):

Andrija Nikolić

Slika 28: Radni nalog za intervenciju nad opremom



РЕПУБЛИКА СРБИЈА
РЕПУБЛИЧКИ ГЕОДЕТСКИ ЗАВОД
СЕКТОР ЗА ИНФОРМАТИКУ И КОМУНИКАЦИЈЕ

06 број:
Датум: . . 2011. године
Булевар војводе Мишића 39
Београд
тел. (011) 2650-538, факс (011) 2650-538

РЕВЕРС

Предмет: Преузимање и задуживање ИТ опреме, делова и потрошног материјала.

Евиденција о преузимању опреме и потрошног материјала за потребе _____:

Р.Б.	Опрема	ЈМ	Количина	Из собе	ИБ/СН	Утрађена код
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						

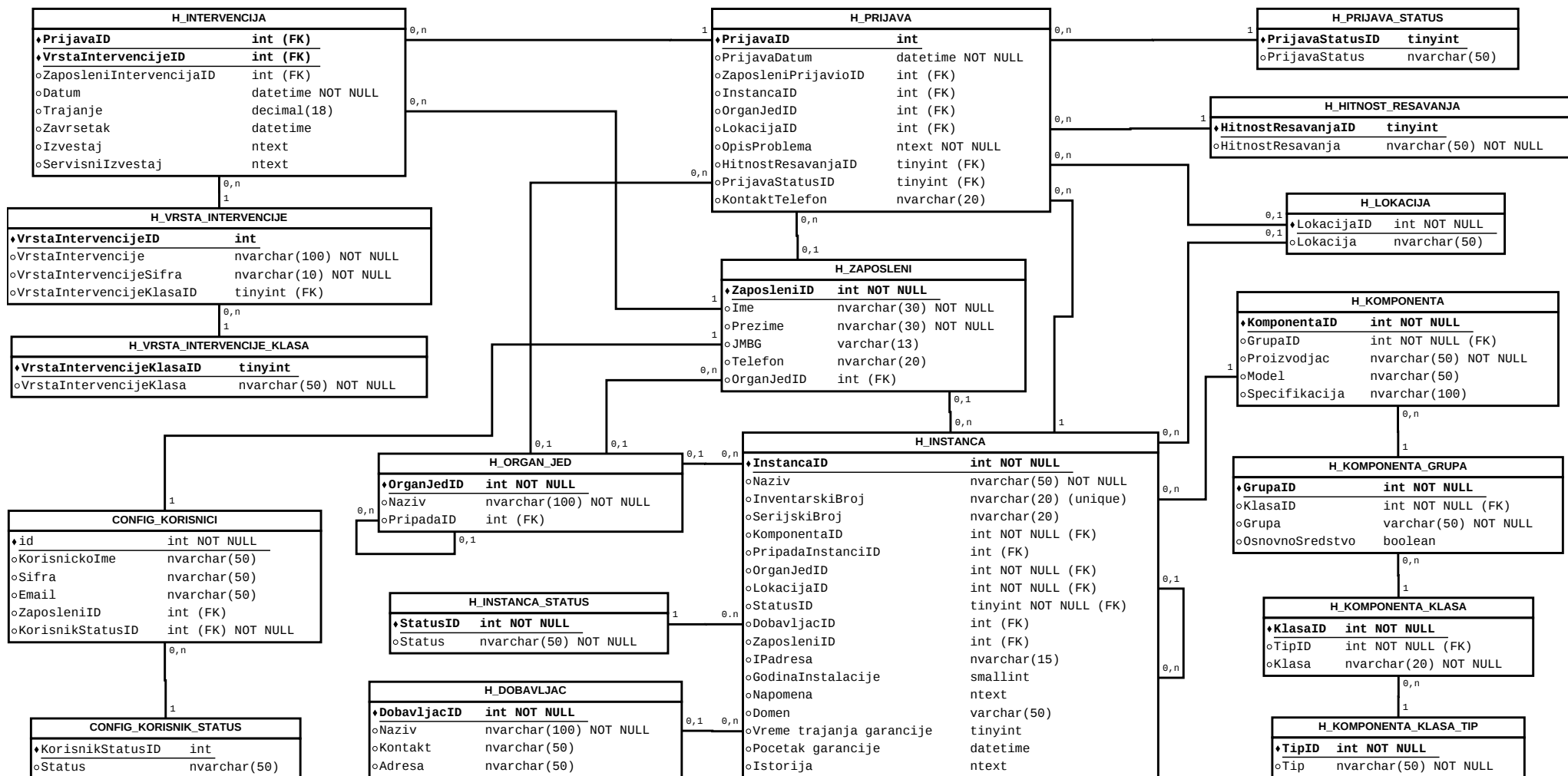
Предао

Примио

Слика 29: Revers servisu za preuzimanje opreme

7.2 Dodatak B - Baza podataka

7.2.1 Dodatak B1 - Dijagrami baza podataka



Slika 30: Model nove baze podataka

7.2.2 Dodatak B2 - Odnosi između relacija

Ime stranog ključa	Nedostajuće vrednosti	Brojnost
FK_H_INSTANCA_H DOBAVLJAC	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_INSTANCA_H INSTANCA	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_INSTANCA_H INSTANCA.STATUS	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_INSTANCA_H KOMPONENTA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_INSTANCA_H LOKACIJA	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_INSTANCA_H ORGAN_JED	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_INSTANCA_H ZAPOSLENI	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_PRIJAVA_H INSTANCA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_PRIJAVA_H HITNOST_RESAVANJA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_PRIJAVA_H LOKACIJA	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_PRIJAVA_H ORGAN_JED	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_PRIJAVA_H PRIJAVA.STATUS	nisu dozvoljene NULL vrednosti	jedan prema nula ili više

Ime stranog ključa	Nedostajuće vrednosti	Brojnost
FK_H_PRIJAVA_H_ZAPOSLANI	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_PRIJAVA_H_INTERVENCIJA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_INTERVENCIJA_H_VRSTA_INTERVENCIJE	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_INTERVENCIJA_H_ZAPOSLANI	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_VRSTA_INTERVENCIJE_H_VRSTA_INTERVENCIJE_KLASA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_KOMPONENTA_H_KOMPONENTA_GRPUPA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_KOMPONENTA_GRPUPA_H_KOMPONENTA_KLASA	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_KOMPONENTA_KLASA_H_KOMPONENTA_KLASA_TIP	nisu dozvoljene NULL vrednosti	jedan prema nula ili više
FK_H_ORGAN_JED_H_ORGAN_JED	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_H_ZAPOSLANI_H_ORGAN_JED	dozvoljene NULL vrednosti	nula ili jedan prema nula ili više
FK_CONFIG_KORISNICI_H_ZAPOSLANI	nisu dozvoljene NULL vrednosti	jedan prema jedan
FK_CONFIG_KORISNICI_CONFIG_KORISNIK_STATUS	nisu dozvoljene NULL vrednosti	jedan prema nula ili više

Tabela 2: Odnosi između relacija

7.2.3 Dodatak B3 - Indeksi tabela

Ime indeksa	Kolona
IX_H.INSTANCA_1	InventarskiBroj (ASC)
IX_H.INSTANCA_2	SerijskiBroj (ASC)

Tabela 3: Indeksi tabela

7.2.4 Dodatak B4 - Provera ograničenja

Ime ograničenja	Izraz
CK_H.INSTANCA	([InventarskiBroj] IS NOT NULL OR [OrganJedID] IS NOT NULL OR [ZaposleniID] IS NOT NULL OR [PripadaInstanciID] IS NOT NULL) AND [KomponentaID] IS NOT NULL AND [StatusID] IS NOT NULL
CK_H.ZAPOSLENI	[Ime] IS NOT NULL AND [Prezime] IS NOT NULL
CK_H.ORGAN_JED	[OrganJedID] <> [OrganJedParentID] AND [Naziv] IS NOT NULL
CK_H.PRIJAVA	[HitnostResavanjaID] IS NOT NULL AND [PrijavaStatusID] IS NOT NULL AND [InstancaID] IS NOT NULL
CK_H.KOMPONENTA_KLASA_TIP	[TipKlase] IS NOT NULL
CK_H.KOMPONENTA_KLASA	[Klasa] IS NOT NULL AND [TipKlaseID] IS NOT NULL
CK_H.KOMPONENTA_GRPUPA	[Grupa] IS NOT NULL AND [KlasaID] IS NOT NULL
CK_H.KOMPONENTA	[Proizvodjac] IS NOT NULL AND [GrupaID] IS NOT NULL
CK_H.LOKACIJA	[Lokacija] IS NOT NULL
CK_H.HITNOST_RESAVANJA	[HitnostResavanja] IS NOT NULL
CK_H.PRIJAVA_STATUS	[PrijavaStatus] IS NOT NULL
CK_H.DOBAVLJAC	[Naziv] IS NOT NULL
CK_H.INSTANCA_STATUS	[InstancaStatus] IS NOT NULL

Ime ograničenja	Izraz
CK_H_INTERVENCIJA	[ZaposleniIntervencijaID] IS NOT NULL AND [PrijavaID] IS NOT NULL AND [VrstaIntervencijaID] IS NOT NULL
CK_H_VRSTA _INTERVENCIJE	[VrstaIntervencije] IS NOT NULL AND [VrstaIntervencijeSifra] IS NOT NULL AND [VrstaIntervencijeKlasaID] IS NOT NULL
CK_H_VRSTA _INTERVENCIJE_KLASA	[VrstaIntervencijeKlasa] IS NOT NULL

Tabela 4: Provera ograničenja (*check constraints*)

8 Reference

- [1] OOD - David Avison, Guy Fitzgerald: Information Systems Development, McGraw Hill Higher Education, 2003. ISBN 978-0077096267
- [2] UML - Martin Fowler, Kendall Scott: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Addison-Wesley, 1999. ISBN 978-0201657838
- [3] BPMN - Business Process Modeling Notation: http://en.wikipedia.org/wiki/Business_Process_Model_and_Notation (posećeno oktobra 2011.)
- [4] C Sharp - Jesse Liberty: Programming C#, 4th Edition, ISBN: 0-596-00699-3
- [5] ASP.NET - Matthew MacDonald, Adam Freeman: Pro ASP.NET 4 in C# 2010, 4th Edition, 2010. ISBN: 978-1430225294
- [6] Microsoft Visual Studio - http://en.wikipedia.org/wiki/Microsoft_Visual_Studio (posećeno oktobra 2011.)
- [7] Dia: [http://en.wikipedia.org/wiki/Dia_\(software\)](http://en.wikipedia.org/wiki/Dia_(software))
- [8] Predrag Živić: Računarska oprema, maj 2005.
- [9] Predrag Živić: Računarska oprema i Help Desk, maj 2005.
- [10] Računarska oprema - Uputstvo za korišćenje veb aplikacije, decembar 2006.
- [11] Organizaciona šema Zavoda: http://www.rgz.gov.rs/template0.asp?PageName=org_sema_2005&MenuID=none (posećeno oktobra 2011.)
- [12] AJAX - Nicholas C. Zakas, Jeremy McPeak, Joe Fawcett: Professional Ajax - Programmer to Programmer, Wrox 2006. ISBN: 978-0471777786
- [13] LINQ - <http://msdn.microsoft.com/en-us/library/bb308959.aspx> (posećeno oktobra 2011.)
- [14] Secure Sockets Layer: http://en.wikipedia.org/wiki/Secure_Sockets_Layer (posećeno oktobra 2011.)