

Maša Šimičić¹
Patrizia Poščić²
Danijela Jakšić³

Stručni rad
UDK 004.65
004.658

PLANIRANJE DIZAJNA I RAZVOJ POSLOVNE APLIKACIJE NAD RELACIJSKOM BAZOM PODATAKA – IZ STUDENTSKE PERSPEKTIVE⁴

SAŽETAK

U ovom radu opisuje se planiranje, dizajn i razvoj poslovne aplikacije „Praonica“ nad relacijskom bazom podataka za udrugu „Terra“ u Rijeci, iz studentske perspektive. Sama ideja izrade poslovne aplikacije nastala je prilikom razgovora sa zaposlenikom udruge „Oaza“ u Rijeci. Planiranju aplikacije prethodilo je intervjuiranje osoba ključnih za prikupljanje korisničkih zahtjeva, koji su također potrebni za izradu relacijske baze podataka u Clarion-u, dijagrama Entitet-veza te na temelju njega napravljenog Relacijskog modela. Ovaj članak sadržava slike kao primjer najbitnijih prozora aplikacije nastale u programskom jeziku Clarion s prikladnim opisom. U konačnici rad je rezultirao cjelovitom aplikacijom za poslovni sustav praonice udruge „Terra“ u Rijeci. Po završetku aplikacije dogovoren je sastanak sa zaposlenicima praonice spomenutih udruge na kojem je prezentirana gotova aplikacija s ciljem da krajnji korisnici istaknu što bi htjeli izmijeniti, nadodati ili ukloniti. Sukladno tome modificirana je aplikacija te instalirana u računalni sustav praonice udruge „Terra“. Uz samu aplikaciju instaliran je i program za automatsko stvaranje sigurnosne kopije aplikacije.

Ključne riječi: baza podataka, praonica, aplikacija, prezentacija, instalacija, Clarion

1. UVOD

Što je to aplikacija? Što je baza podataka, a što relacijska baza podataka? Što je Clarion?

Aplikacija (engl. *application*) je programski proizvod nad bazom podataka koji obavlja određenu poslovnu funkciju (Pavlič, 2011). Životni ciklus programskog proizvoda često se provodi prema nekom od modela procesa. Jedan od klasičnih modela procesa je i vodopadni (engl. *waterfall*) model. Postoji mnogo verzija tog modela pa se model najčešće sastoji od faze definiranja korisničkih zahtjeva, oblikovanja softvera, implementacije programskog koda, testiranja i uvođenja u rad te održavanja kroz vrijeme. Za proces je zapravo karakteristično da se razvija slijedom koraka gdje završetkom jedne faze počinje druga. Baza podataka (engl. *Database*) je neredundantni

¹ Univ. bacc. Inf., E-mail: masa.simicic@student.uniri.hr

² Dr. sc., izvanredni profesor, Odjel za informatiku, Sveučilište u Rijeci, Radmile Matejčić 2, 51 000 Rijeka, Hrvatska.
E-mail: patrizia@inf.uniri.hr

³ Dr. sc., poslijedoktorand, Odjel za informatiku, Sveučilište u Rijeci, Radmile Matejčić 2, 51 000 Rijeka, Hrvatska.
E-mail: danijela.kasic@inf.uniri.hr

⁴ Datum primitka rada: 21. 12. 2017.; datum prihvaćanja rada: 1. 3. 2018.

skup podataka o stanju sustava. Ona je skup podataka o svim pojavljivanjima entiteta, veza i njihovih atributa (Pavlič, 2011). Relacijska baza podataka (engl. *relational database*) je skup u vremenu promjenjivih relacija (veza) opisanih u shemi baze podataka. Shema baze podataka je zapis podatkovne strukture i ograničenja među podacima poslovne organizacije metodom za modeliranje podataka (Pavlič, 2011).

Kod izrade aplikacije koja radi nad relacijskom bazom podataka, koristit će se Clarion koji je posebno dizajniran za izradu takve vrste aplikacije. Clarion je komercijalni programski jezik tvrtke SoftVelocity koji je kompatibilan sa SQL jezikom za rad s bazama podataka. Radno okruženje (engl. *Development Environment*) programa Clarion počiva na programskom jeziku (engl. *Programming Language*) Clarion-a. Radno okruženje pruža obrasce generirane kodom koji se temelje na sistemu zadanih uzoraka što omogućuje programerima da rade na razini koja je viša od samog pisanja koda. Generator tada pretvara tu višu razinu u sam kod. Također, u programu Clarion moguće je i upisivati kod u postojeći kako bi izmijenili ili proširili neki zadani uzorak aplikacije prema vlastitim potrebama. Tako je omogućeno stvaranje sofisticirane baze podataka bez ikakvog upisivanja programskog koda ili dodavanjem samo minimalne količine istog.

2. KORISNIČKI ZAHTJEVI ZA IZRADU APLIKACIJE

Kroz intervju sa zaposlenicima Udruge „Oaza“ prikupljeni su korisnički zahtjevi za izradu poslovne aplikacije. Udruga „Oaza“ provodi programe usmjerene zaštiti beskućnika i socijalno osjetljivih osoba te prevenciji socijalne izolacije i rizičnih ponašanja. Njihovo područje djelovanja za punoljetne građane je odgoj i obrazovanje, socijalna skrb te program prihvatilišta za beskućnike. Oni također nude mnoštvo aktivnosti volontiranja u koje se svi građani mogu uključiti. Usko su povezani s udrugom „Terra“ čiji je cilj djelovanja prevencija i suzbijanje ovisnosti, AIDS-a i ostalih spolno prenosivih bolesti, posebice kod rizičnih i marginaliziranih skupina, olakšavanje i omogućavanje procesa reintegracije ovisnika nakon završetka tretmana za liječenje, prevencija ovisnosti kod rizičnih skupina mladih te senzibilizacija i edukacija javnosti o problematici zlouporabe droga.

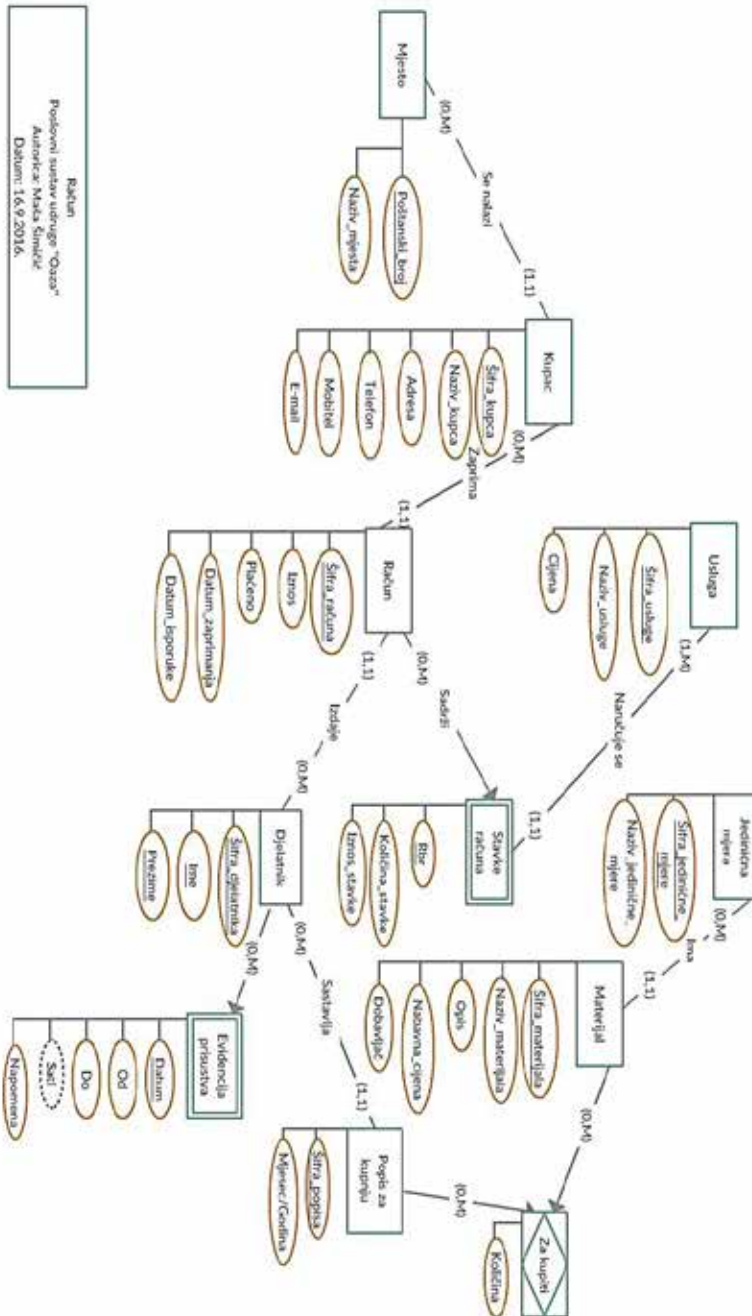
Kroz zajednički razgovor zaključili smo kako bi im od najvećeg značaja bila baza podataka i aplikacija za njihov novi poslovni projekt koji je praonica. Praonica dakle pruža usluge pranja, sušenja i peglanja svih vrsti robe za svoje klijente te bi ima od velikog značaja bila baza podataka pomoću koje bi zaposlenici mogli voditi evidenciju o praonici i njezinim poslovima.

Nadalje određeni su zahtjevi za samu aplikaciju „Praonica“, a to su sljedeći:

1. pregled kupaca, računa i njegovih stavki, djelatnika i njihovih evidencija prisustva
2. upisivanje informacija o kupcima kao što su Naziv, Adresa, Telefon, Mobitel i E-mail adresa
3. bilježenje datuma zaprimanja te datuma isporuke robe kupcu
4. automatsko računanje ukupnog iznosa stavki računa te polje za vođenje evidencije o tome da li je iznos plaćen ili nije od strane kupca
5. evidencija prisustva djelatnika
6. popis za kupovinu potrebnog materijala

Sukladno navedenim zahtjevima nastao je dijagram koji vidimo na slici 1.

Slika 1. DEV



Izvor: obrada autora

Životni ciklus aplikacije proveden je prema klasičnom faznom modelu odnosno takozvanom vodopadnom modelu procesa. Model započinje formiranjem korisničkih zahtjeva kako je prethodno opisano te se nastavlja s dizajniranjem aplikacije točnije specifikacijom komponenti poslovnog sustava i njihovih veza što ćemo vidjeti u sljedećem poglavlju. U poglavlju nakon opisuje se pak implementacija programskog koda te odgovarajući izgled aplikacije. Prije kraja ili same instalacije aplikacije i testiranja održana je iznimno prezentacija kako bi se utvrdilo da li su potrebne ikakve njene izmjene. Unatoč vraćanju na prvi korak i dobivanju novih zahtjeva te njihovom implementacijom u program, životni ciklus aplikacije nastavlja se ondje gdje je stao.

3. MODELIRANJE PODATAKA

3.1 Dijagram Entitet-veza

Metoda entiteti – veze, skraćeno EV, odnosno model entiteti - veze (engl. Entity – relationship model) je jedna od metoda za modeliranje podataka. EV je grafički prikaz međusobno povezanih grupa podataka nekog sustava.

Dijagram strukture modela naziva se dijagram entiteta i veza, skraćeno DEV (Pavlič, 2011).

Koncepti strukture metode EV koji se koriste u dijagramu za poslovnu aplikaciju „Praonica“ su: Entitet i tip entiteta, veza i tip veze, atribut tipa entiteta, slab tip entiteta, agregirani tip entiteta.

3.2 Relacijski model

DEV se dalje prevodi u relacijski model, skraćeno RM, s ciljem praktičnijeg modeliranja podataka (Pavlič, 2011).

Osnovni koncepti relacijskog modela podataka su relacije. Relacije su iste onima u matematici što prikazuje slika 2, s tom razlikom da su relacije u relacijskom modelu podataka vremenski promjenjive (Pavlič, 2011).

Slika 2. Matematička definicija relacije

Definicija [uredi | uredi izvor]

Binarna relacija R između dva skupa A i B je podskup kartezijevog proizvoda

$$A \times B = \{(a, b) \mid a \in A, b \in B\}; \quad R \subseteq A \times B.$$

Izvor: <https://bs.wikipedia.org>

Primarni ključ (engl. *primary key*) je jedan od ključeva tipa entiteta odabran za zamjenika entiteta zbog nekog značajnog razloga (Pavlič, 2011). U DEV-u na slici 1 i relacijskom modelu koji slijedi prikazan je tako što je taj atribut u oba slučaja podvučen linijom.

Vanjski ključ nastaje iz tipa veze (1,1):(0,M) ili (1,1):(1,M) kada se primarni ključ tipa entiteta koji ulazi u vezu sa strane s brojnošću M umeće kao vanjski ključ, odnosno atribut u relaciju dobivenu

prevođenjem DEV-a u relacijski model. U relacijskom modelu taj je atribut prikazan tako što je podvučen isprekidanom linijom i naziv toga atributa ukošen je.

Entiteti se iz DEV-a u RM prevode tako što je njihov naziv naveden ispred zagrade. U zagradi koja slijedi navedeni su redom: primarni ključ (ili više njih), atributi tog entiteta te vanjski ključ (ili više njih).

Račun(Šifra_rachuna, Iznos, Plaćeno, Datum_zaprimanja, Datum_isporuke, Šifra_djelatnika, Šifra_kupca)

Stavka(Šifra_rachuna, Rbr, Količina_stavke, Iznos_stavke, Šifra_usluge)

Usluga(Šifra_usluge, Naziv_usluge, Cijena_usluge)

Kupac(Šifra_kupca, Naziv_kupca, Adresa, Telefon, Mobitel, E-mail, Poštanski_broj)

Mjesto(Poštanski_broj, Naziv_mjesta)

Materijal(Šifra_materijala, Naziv_materijala, Opis, Nabavna_cijena, Dobavljač, Šifra_mjere)

Mjera(Šifra_mjere, Naziv_mjere)

Za kupiti(Šifra_materijala, Šifra_popisa, Količina)

Popis za kupovinu(Šifra_popisa, Mjesec/Godina, Šifra_djelatnika)

Djelatnik(Šifra_djelatnika, Ime, Prezime)

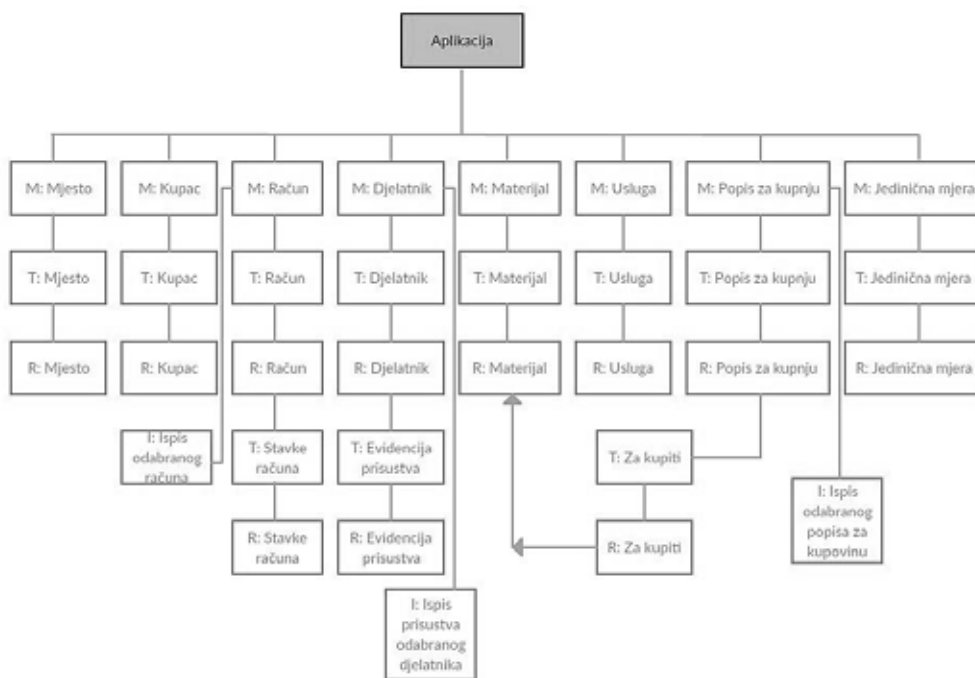
Evidencija prisustva(Šifra_djelatnika, Datum_ep, Od, Do, Sati, Napomena)

4. APLIKACIJA

U ovom poglavlju pružit ćemo pogled na samu aplikaciju relacijske baze podataka, u programskom jeziku Clarion. Predstaviti ćemo vizualni izgled gotove aplikacije te manje programski kod jer zbog praktične upotrebe u poslovanju nije moguće objaviti veći dio koda.

Dijagram programskog proizvoda, skraćeno DPP, prikazuje strukturu aplikacije. Izrađen na temelju EV modela DPP prikazuje kako su prozori u aplikaciji vezani. DPP razlikuje četiri prozora te navodi slova M, T, R te I prije naziva tablice. Slovo M označava meni prozor iz kojega se pristupa pregledu prozora ili procedure. Takav pregled označava slovo T i preko njega se pristupa ažuriranju prozora ili procedure te je on označen slovom R. Slovo I označava izvješća pa se u dijagramu vidi preko kojih tablica se njima pristupa.

Slika 3. DPP



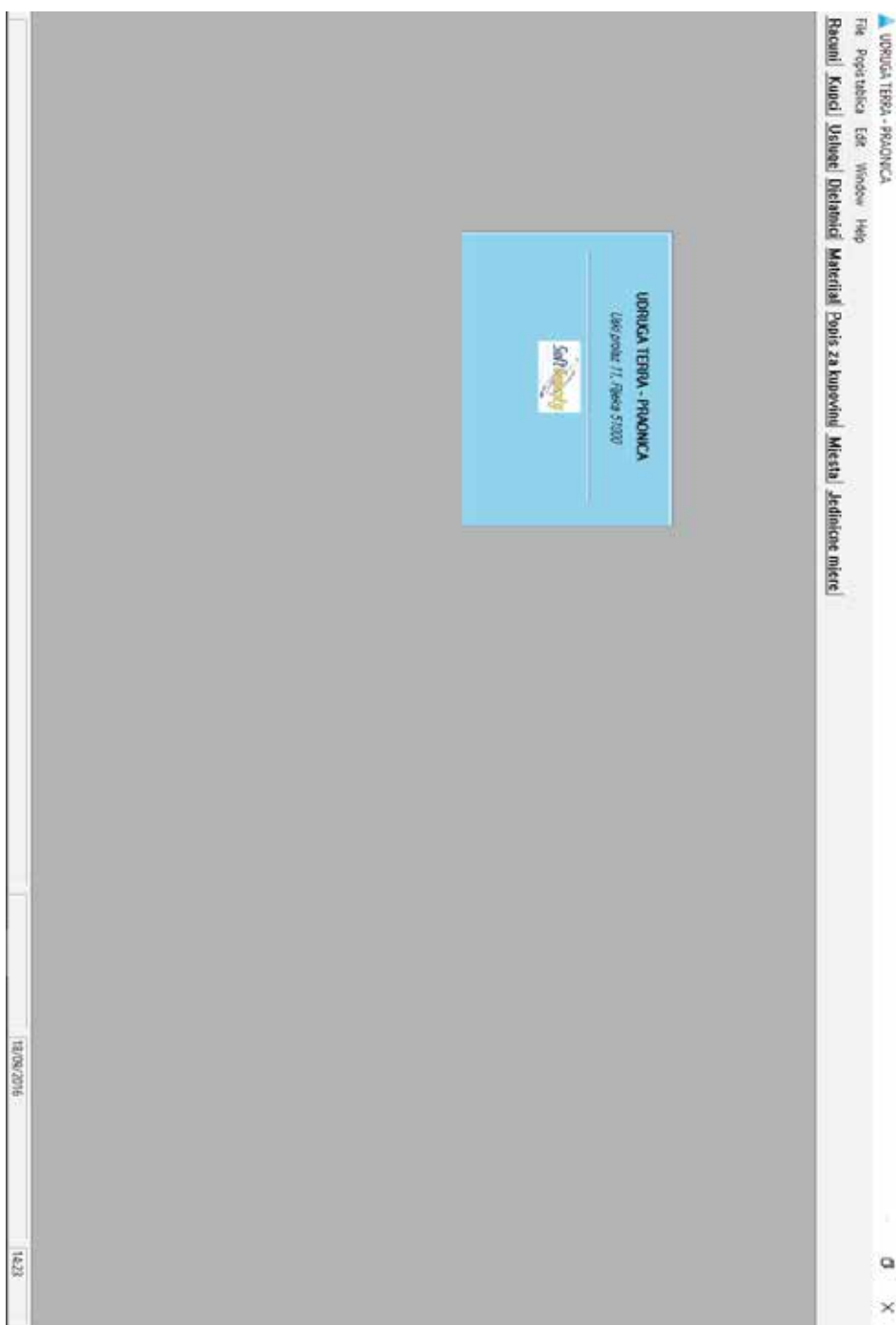
Izvor: obrada autora

4.1 Glavni izbornik

Na slici 3 prikazan je glavni izbornik aplikacije. Do njega se dolazi odmah pri ulasku u samu aplikaciju.

Okvir prozora nosi ikonu Clarion-a te naziv aplikacije u gornjem lijevom kutu. U sredini prozora nalazi se skočni prozor. Skočni prozor je podešen da se prikazuje vremenski 5 sekundi ili se gasi pritiskom mišem na njega. Prozor je uređen u skladu s aplikacijom i na njemu se nalazi naziv udruge, svrha i adresa. Ispod teksta je slika koja je logo tvrtke SoftVelocity čiji je programski jezik Clarion. Slika je obično postavljena po default-u na skočni prozor.

Slika 4. Glavni izbornik



Izvor: obrada autora

4.2 Izvješća

Korisnički zahtjev za evidencijom prisustva djelatnika realizirao se na način koji vidimo na slici 4.

Na Evidenciji prisustva uz uobičajeni naziv udruge i naslov izvješća nalaze se i podaci o pojedinom djelatniku kao što su njegova šifra, ime i prezime te informacije o evidenciji njegovog prisustva. Za pojedini datum ispisuje se početak radnog vremena (Od) i kraj radnog vremena (Do) te ukupan broj radnih sati za taj dan uz opcionalnu napomenu na kraju.

Slika 5. Izvješće Evidencije prisustva

The screenshot shows a 'Report Preview' window with a menu bar (File, View, Zoom) and a toolbar with options for Page, Across, Down, and Zoom (set to Tilepages). The main content area displays a report titled 'LDRUGA TERENA - PRIONDA' with the subtitle 'Evidencija prisustva'. The report is organized into three sections, one for each employee (Situacija 1, 2, and 3). Each section includes the employee's name and surname, followed by a table of attendance data for the date 12.11.2016. The columns in the table are Date, Start Time (Od), End Time (Do), Hours (Sati), and Remarks (Napomena).

LDRUGA TERENA - PRIONDA					
Evidencija prisustva					
Situacija 1					
Ime: Marko					
Prezime: MARKO					
Date	Od	Do	Sati	Napomena	
12.11.2016	13:00	16:00	1	nista	
12.11.2016	13:00	16:00	1	nista	
Situacija 2					
Ime: Ivan					
Prezime: IVO					
Date	Od	Do	Sati	Napomena	
12.11.2016	14:00	16:00	1	nista	
Situacija 3					
Ime: Ana					
Prezime: ANA					
Date	Od	Do	Sati	Napomena	
12.11.2016	16:00	18:00	2	nista	

Izvor: obrada autora

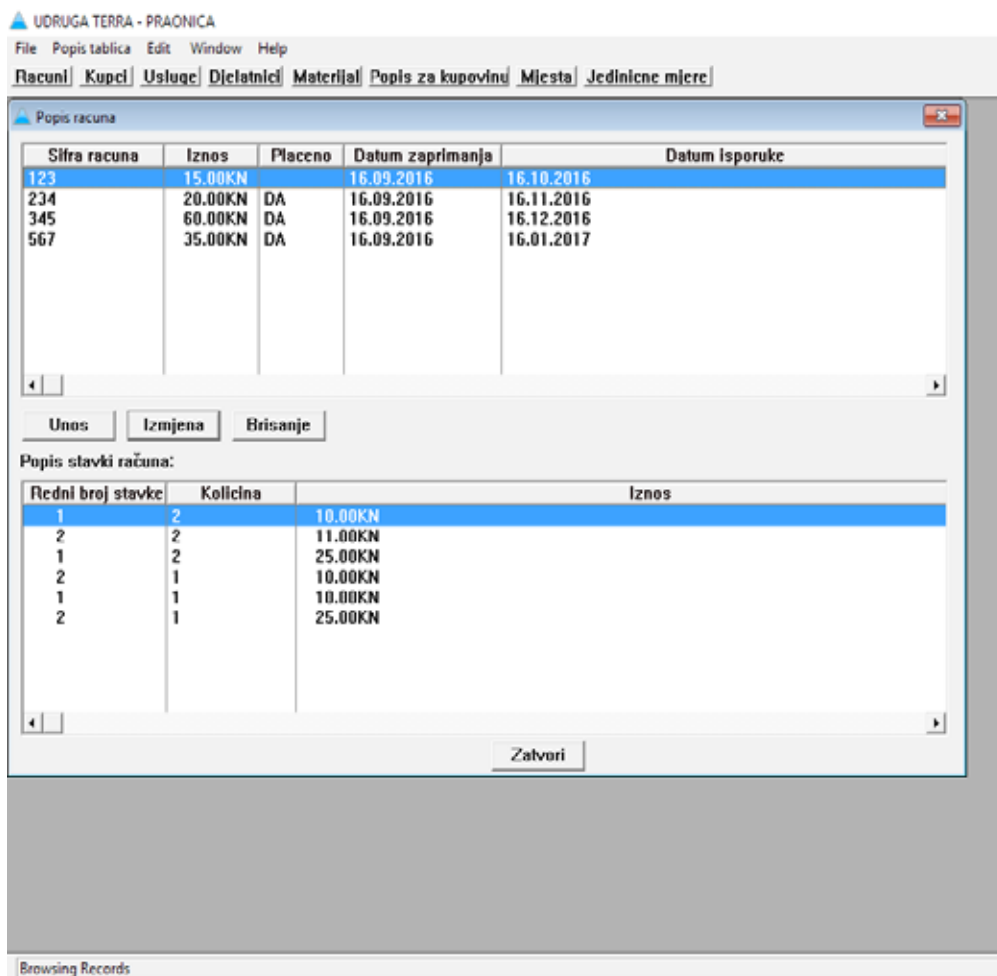
4.3 Prozori za pregled i unos zapisa

Na sličan način izvedeni su svi prozori za pregled i unos zapisa u bazu podataka te ćemo navesti primjer nekolicine.

Pritiskom na gumb *Računi*, na glavnom izborniku, otvara se prozor *Popis računa* prikazan na slici 5. Na njemu možemo vidjeti veliku tablicu sa zapisima svih računa u bazi te gumbima za njihovo uređivanje (*Unos*, *Izmjena*, *Brisanje*). U istom prozoru nalazi se i druga tablica vezana za tablicu *Račun* a to su *Stavke računa* te tako korisnici mogu jednostavno vidjeti popis svih stavki ako to požele no ne mogu je direktno uređivati (nema gumba *Unos*...).

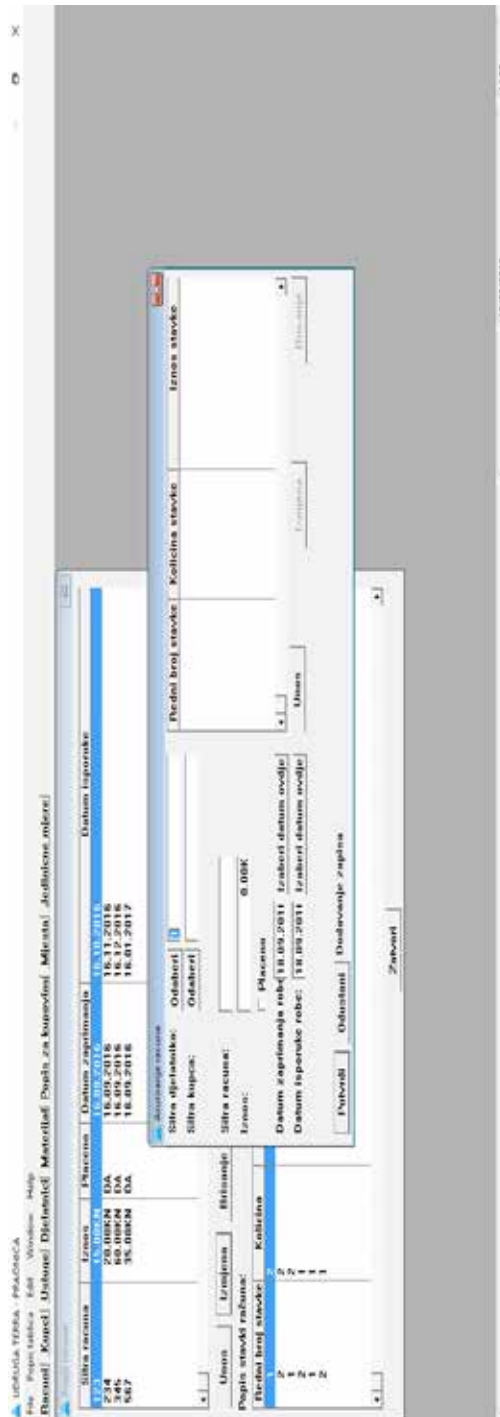
Pritiskom na gumb za uređivanje pristupa se novome prozoru kojeg vidimo na slici 6, a to je *Ažuriranje računa*.

Slika 6. Popis računa



Izvor: obrada autora

Slika 7. Ažuriranje računa



Izvor: obrada autora

U ovome prozoru unose se informacije o samome računu. Korisnici aplikacije moći će ovdje unijeti račune koje su prethodno izdali kako bi imali evidenciju o njima na jednom mjestu, kao i one novonastale. Na samome početku korisnik će unijeti šifru djelatnika koji sastavlja račun i šifru kupca kojemu je namijenjen taj račun. Šifra djelatnika se pritiskom na susjedni gumb *Odaberi* može odabrati direktno iz tablice *Djelatnici* bez ručnog upisivanja iste, kao i Šifra kupca iz tablice *Kupci* na isti način. Šifru računa upisuje korisnik. Ona može biti bilo koji proizvoljan broj ili niz brojeva. Polje iznos se računa automatski tako da zbraja sve iznose stavke računa koji se unose nakon što se popune podaci o računu. Stoga polje iznos možemo ostaviti prazno te se vratiti na njega nakon unosa stavki.

Datum zaprimanja i isporuke važan je članovima praonice te ga ne moraju ručno upisivati nego ga mogu izabrati pritiskom na gumb *Izaberi datum ovdje*. Po default-u u tim poljima već će se nalaziti trenutni datum u slučaju da se želi ostaviti istog za pospremanje.

Na dnu prozora nalaze se gumbi *Potvrdi* i *Odustani* koji su sami po sebi jasni te nakon njih slijedi systemska poruka. Za prethodni pritisak na gumb *Unos*, systemska poruka će biti „Dodavanje zapisa“, za *Izmjena* bit će „Izmjena zapisa“, a za *Brisanje* će biti „Brisanje zapisa“.

Ono što slijedi nakon popunjavanja podataka o računu je popunjavanje podataka o stavkama računa pritiskom na neki od gumba za uređivanje ispod tablice ovog prozora koje vidimo na slici 7.

Ažuriranje stavke podrazumijeva dodavanje šifre usluge. Kada pritisnemo gumb *Odaberi* pored šifre, omogućava nam se odabir usluge (iz tablice *Usluge*) na koju se odnosi ova stavka računa. Jednom kada je usluga odabrana, u polju šifre usluge prikaže nam se taj jedinstveni broj usluge te njen naziv. Redni broj stavke koja će biti na tom računu unosi se ručno, a količina te stavke može se birati putem polja SPIN tako da povećavamo ili smanjujemo broj sa strelicama gore i dolje. Količina se može upisati i ručno ako se tako želi. Na temelju količine stavke i jedinične cijene usluge iz tablice *Usluge*, automatski se izračunava iznos stavke računa. Takav automatski izračun postignut je dodavanjem programskog koda na polje iznosa stavke u trenutku kada je polje odabrano. Iznos stavke dobiven je pritom iz umnoška količine stavke iz tablice *Ažuriranje stavke računa* i cijene usluge iz tablice *Usluga*.

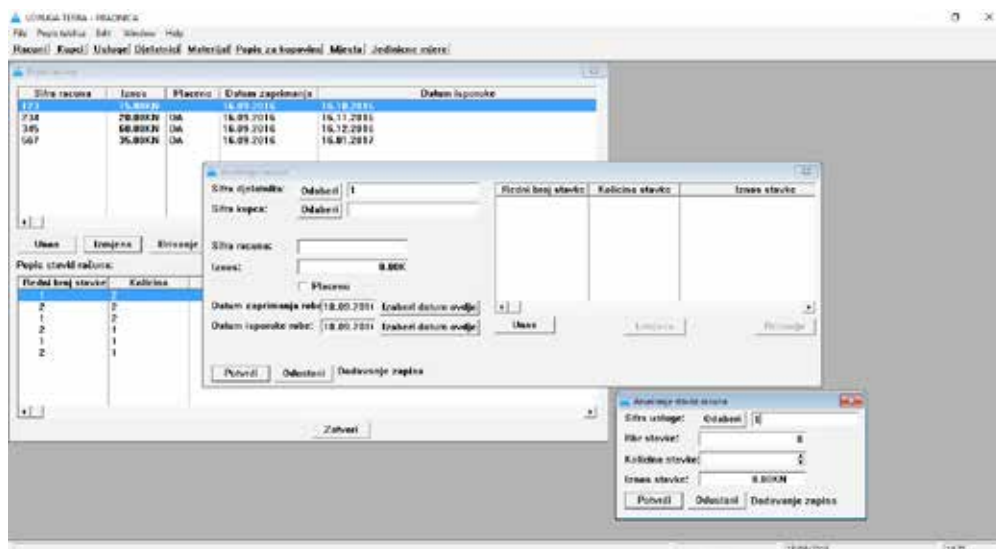
Tablica 1. Automatski izračun iznosa stavke

$STA:Iznos_stavke=STA:Kolicina_stavke*USL:Cijena_usluge$

Izvor: obrada autora

Korisnici aplikacije, u tablici *Usluge*, u svakom trenutku mogu unositi i mijenjati iznose usluga koje pružaju. Važno je također napomenuti da ako se cijena usluga promijeni, iznos već unesenog računa neće se izmijeniti već će nova izmjena utjecati samo na novo unesene račune. Ako se iz nekog razloga odluči izmjenjivati podatke o starom računu, odnosno izračunati iznos stavke prema novoj cijeni, to se također može napraviti osvježavanjem pojedinog polja Iznos stavke odabranog računa.

Slika 8. Ažuriranje stavke računa



Izvor: obrada autora

5. PREZENTACIJA APLIKACIJE

5.1 Općenito

Nakon što je izrađena aplikacija kontaktiran je predstavnik udruge „Oaza“ i dogovoren je termin prezentacije aplikacije koja se održala potom ispred tri zaposlenika praonice i samim predstavnikom Udruge. Prezentacija se održala u prostorima „Oaze“ i „Terre“ na adresi Krešimirova 12 u Rijeci.

Napominjemo da je za prezentaciju aplikacija bila posebno prilagođena. Naime uklonjen je skočni prozor u potpunosti, data joj je nova slika pozadine i promijenjene su veličine i raspored gumbiju na početnom zaslonu. Svi prozori povećani su te su sukladno tome organizirane njihove stavke, radi jasnijeg pregleda. Izvješća su sada upotpunjena i isključivo na gumbima unutar pojedinih pregleda te ista mogu biti ispisana samo za odabrani zapis umjesto za sve zapise u bazi podataka. Nadalje tablice „Popis stavki računa“ na prozoru „Popis računa“, „Evidencija prisustva“ na „Popis djelatnika“ i „Za kupiti“ na prozoru „Popis za kupovinu“ uklonjeni su.

5.2 Dodatni korisnički zahtjevi

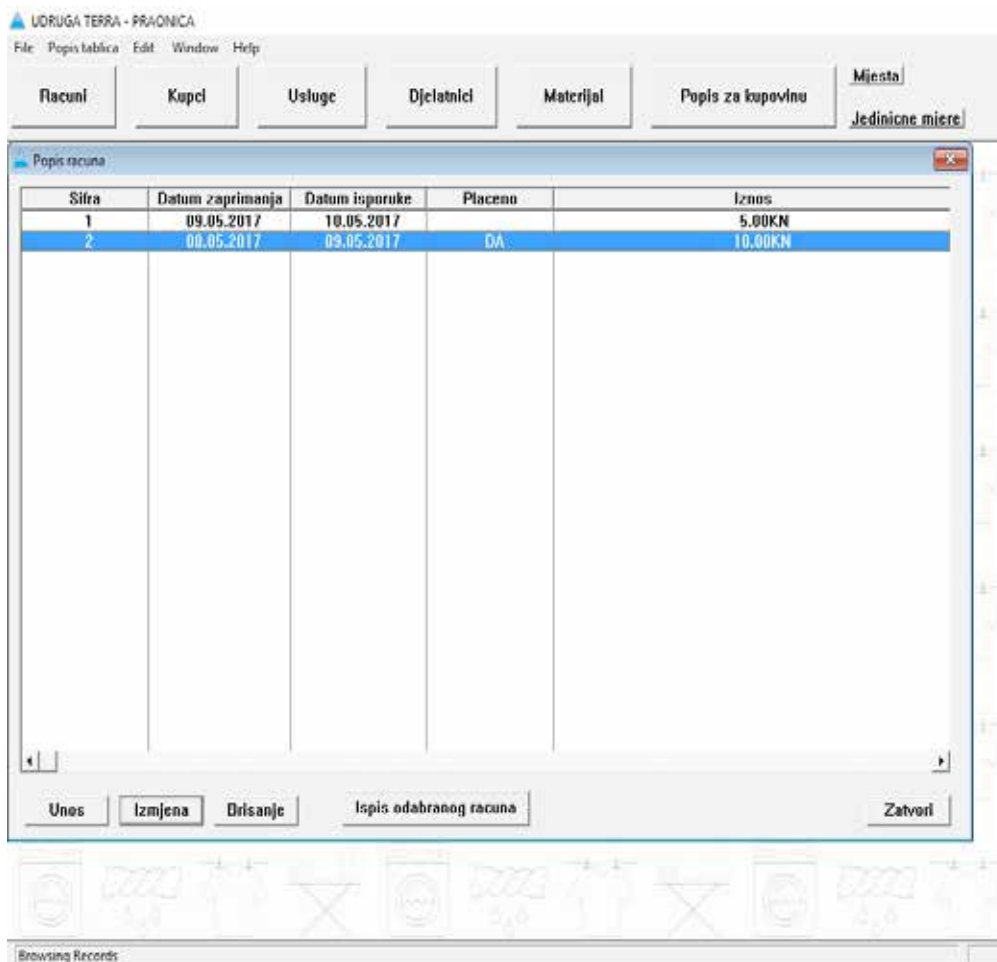
Nakon predstavljanja aplikacije uslijedila su pitanja. Najveću zabrinutost stvarale su šifre pojedinog atributa. Njihov prijedlog je bio da se šifre automatski upisuju sa svakim novim unosom bilo kojeg atributa, kao redni broj. Sljedeće što ih je zanimalo bilo je vezano uz samu instalaciju aplikacije te su se htjeli osigurati da slučajno ne izbrišu sve zapise iz baze podataka, odnosno da postoji „backup“ datoteka te iste aplikacije.

Naposljetku zanimalo ih je ako se mogu zapisi iz aplikacije na računalu u praonici pregledavati s nekog drugog udaljenog računala poput onoga u samoj Udruzi.

6. PRILAGOĐENI IZGLED APLIKACIJE PREMA NOVIM ZAHTJEVIMA

S novim zahtjevima dobivenima na prezentaciji aplikacije ona je poprimila izgled koji vidimo na slici 8. Svaka šifra jakog entiteta sada je bila automatski redni broj, a izuzetak je bila šifra „Poštanski broj“ koja se uvijek mora sama unositi te nije prikladno da bude redni broj.

Slika 9. Novi izgled aplikacije



Izvor: obrada autora

7. INSTALACIJA APLIKACIJE

7.1 Instalacija

Kako bi aplikacija napravljena u Clarionu ispravno radila na nekom računalu koje nema instaliran sami program ona mora sadržavati datoteke s nastavkom .dll. Dynamic-link library (ili DLL) je Microsoft-ova implementacija koncepta dijeljenih datoteka za operacijske sustave Microsoft Windows i OS/2 te predstavlja skup funkcija i procedura koje mogu koristiti drugi programi. (<http://lecto-player.lss.hr>)(<https://en.wikipedia.org>)

Prilikom izrade aplikacije ta opcija bila je prethodno postavljena tako da je moj zadatak bio samo smjestiti kopiju aplikacije na računalo praonice.

Datoteku je postavljena na računala udruge te je stvoren prečac na datoteku s nastavkom .exe odnosno na izvršni program.

Aplikacija je isprobana tada s nekoliko testnih podataka i radila je ispravno.

7.2 Automatski „Backup“

Uz instalaciju programa, prema dodatnim postavljenim zahtjevima slijedila je instalacija programa za automatski backup aplikacije. Program „Everyday auto backup“ koji je besplatno preuzet s interneta instaliran je direktno na računala praonice. Pomoću navedenog programa aplikacija „Praonica“ svakog dana u 23 sata i 55 minuta dobiti će svoju kopiju zadnje spremljene verzije.

8. ZAKLJUČAK

Konačni rezultat ovog rada relacijska je baza podataka u Clarion-u za poslovni sustav Praonice Udruge „Terra“ u Rijeci. Prvi korak bio je dogovaranje sastanka sa zaposlenicima iz Udruge „Oaza“. Zaključeno je kako bi im od najvećeg značaja bila baza podataka za njihov novi poslovni projekt koji je praonica. Praonica bi dakle pružala usluge pranja, sušenja i peglanja svih vrsti robe za svoje klijente. Potrebna je bila stoga baza podataka pomoću koje bi zaposlenici mogli voditi evidenciju o praonici i njezinim poslovima. Na temelju korisničkih zahtjeva napravljen je Dijagram entitet – veza i Relacijski model. Naposljetku nastala je sama aplikacija relacijske baze podataka s mogućnošću spremanja svih zapisa o računima i njegovim stavkama, kupcima, uslugama, djelatnicima i njihovim evidencijama o prisustvu. Također u bazu podataka moguće je spremati zapise o svom potrebitom radnom materijalu ove praonice. Pomoću aplikacije korisnici sada mogu pregledavati te zapise, uređivati ih ili brisati te mogu i ispisivati podatke o računima, djelatnicima ili potrebitom materijalu praonice.

Možemo zaključiti kako je cijeli proces izrade ove poslovne aplikacije bio od velikog iskustvenog značaja, iz studentske perspektive: prilika za samostalno izrađivanje relacijske baze podataka u programskom jeziku Clarion, suradnja s ljudima iz poslovnog sektora, prikupljanje korisničkih zahtjeva, smišljanje dizajna aplikacije te testiranje i uvođenje u rad.

ZAHVALA

Zahvalila bih se prvo svojoj mentorici prof. dr. sc. Patriziji Poščić i asistentici dr.sc. Danijeli Jakšić što su mi pružile priliku za rad u praksi te na njihovim smjernicama i pomoći pri izradi aplikacije. Potom bih se zahvalila gospodinu Dejanu Travici i ostalim članovima udruge „Oaza“ i „Terra“ koji su mi učinili uslugu i opskrbili me svim potrebnim informacijama i dokumentima za izradu aplikacije.

LITERATURA

Pavlić, M. P. (2011). Oblikovanje baza podataka, Rijeka: Odjel za informatiku Sveučilišta u Rijeci

SoftVelocity: <http://www.softvelocity.com/> (12. 5. 2017)

Seminar: <http://lecto-player.lss.hr/recordings/fer/predmeti/racfor/2016/seminari/mveizovic/seminar.pdf> (12. 5. 2017)

Udruga Terra: <http://www.udrugaterra.hr/> (12. 5. 2017)

Wikipedia: [https://en.wikipedia.org/wiki/Clarion_\(programming_language\)](https://en.wikipedia.org/wiki/Clarion_(programming_language)) (12. 5. 2017)

Wikipedia: [https://bs.wikipedia.org/wiki/Relacija_\(matematika\)](https://bs.wikipedia.org/wiki/Relacija_(matematika)) (12. 5. 2017)

Wikipedia: https://en.wikipedia.org/wiki/Dynamic-link_library (12. 5. 2017)

Volonterski centar Rijeka: <http://www.volonterski-centar-ri.org/udruga-za-beskucnike-i-socijalno-ugrozene-osobe-oaza/> (12. 5. 2017)

Maša Šimičić¹
Patrizia Poščić²
Danijela Jakšić³

Professional paper
UDC 004.65
004.658

DESIGN PLANNING AND DEVELOPMENT OF BUSINESS DATABASE APPLICATION – FROM THE STUDENT’S PERSPECTIVE⁴

ABSTRACT

This scientific article describes design planning and development of the business database application “Laundry” for an association named “Terra” in the city of Rijeka. The idea for a business application came during a conversation with the “Oaza” association employee in Rijeka. Application design planning was preceded by interviewing people which was indispensable in order to make a list of user requirements. These user requirements were needed for developing a relational database in Clarion, Entity – relationship model and for relation model. This article contains pictures as examples of the most important parts of the application made in Clarion program with a matching description. The work resulted in a complete application for the business system association “Terra’s Laundry” in Rijeka. When the application was done, the meeting was arranged so that the application could be presented to “Terra’s Laundry’s” employees. At the presentation of the application the employees were expected to comment and say what should be changed, added or removed from the application. In line with what was said at the meeting, the application was modified and finally it was ready to be installed on the “Terra’s Laundry” main computer. It was also necessary to install a program for automatic backup in order to make its security copy on daily basis.

Key words: database, laundry, application, presentation, installation, Clarion

⁵ Univ. bacc. Inf, E-mail: masa.simicic@student.uniri.hr

⁶ PhD, Associate Professor, Department of Informatics, University of Rijeka, Radmile Matejčić 2, 51 000 Rijeka, Croatia.
E-mail: patrizia@inf.uniri.hr

⁷ PhD, Postdoc, Department of Informatics, University of Rijeka, Radmile Matejčić 2, 51 000 Rijeka, Croatia.
E-mail: danijela.kasic@inf.uniri.hr

⁸ Received: 21 December 2017; Accepted: 1 March 2018