Redis baza podataka implementirana na C# aplikaciji

Šarčević, Tomislav

Undergraduate thesis / Završni rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:204708

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Preddiplomski sveučilišni studij računarstva

REDIS BAZA PODATAKA IMPLEMENTIRANA NA C# APLIKACIJI

Završni rad

Tomislav Šarčević

Osijek, 2016.

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek,06.09.2016.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada

Ime i prezime studenta:	Tomislav Šarčević
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3517, 02.08.2013.
OIB studenta:	32975316739
Mentor:	Doc.dr.sc. Ivica Lukić
Sumentor:	
Naslov završnog rada:	Redis baza podataka implementirana na C# aplikaciji
Znanstvena grana rada:	**Programsko inženjerstvo (zn. polje računarstvo)**
Predložena ocjena završnog rada:	Izvrstan (5)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 Postignuti rezultati u odnosu na složenost zadatka: 2 Jasnoća pismenog izražavanja: 2 Razina samostalnosti: 3
Datum prijedloga ocjene mentora:	06.09.2016.
Datum potvrde ocjene Odbora:	12.09.2016.

Potpis mentora za predaju konačne verzije rada u Studentsku službu pri završetku studija:	Potpis:
	Datum:

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTIRADA

Osijek,13.09.2016.

Ime i prezime studenta:	Tomislav Šarčević
Studij:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	R3517, 02.08.2013.
Ephorus podudaranje [%]:	0

Ovom izjavom izjavljujem da je rad pod nazivom**: Redis baza podataka implementirana na C# aplikaciji**

izrađen pod vodstvom mentora Doc.dr.sc. Ivica Lukić

i sumentora

mojvlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osimonih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada,osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, ašto je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ:

1. UVOD 1
1.1 Zadatak završnog rada 1
2. ALATI 2
2.1 Microsoft Visual Studio 2
2.2 C# 2
2.3 Redis 3
3. APLIKACIJA 7
4. ZAKLJUČAK 12
LITERATURA 13
SAŽETAK 14
ABSTRACT 15
ŽIVOTOPIS 16
PRILOZI 17

1. UVOD

Zadatak ovog završnog rada izrada je C# aplikacije za vođenje informacija o zaposlenicimauz pomoć Redis baze podataka. Glavni cilj ove aplikacije je na što jednostavniji način pomoći poslodavcima u vođenju baze podataka o zaposlenicima, njihovim privatnim podacima, te o podacima vezanima uz njihov posao. Kako bi ova aplikacija bila funkcionalna, bilo je potrebno odrediti neke osnovne podatke o radnicima koji će pripomoći razlikovanju između radnika, tu se kao glavna varijabla koja ih razlikuje uzeo osobni identifikacijski broj (OIB).

Kako bi cijela aplikacija funkcionirala, koristilo se znanje stečeno na fakultetu iz kolegija „Objektno orijentirano programiranje“, „Baze podataka“ i „Programiranje 2“. Završni rad se dijeli na 2 dijela, na korisnički i programerski dio. Korisnik vidi samo aplikaciju, gumbove za rad s bazom podataka, dok se sav bitan dio odrađuje u programerskom dijelu koji je nevidljiv krajnjem korisniku aplikacije (poslodavcu).

U nastavku završnog rada dati će se teoretska podloga o svakom od dijelova aplikacije i nakon toga će se opisati način i faze nastanka aplikacije. Opisat će se vizualni nastanak aplikacije, kao i dijelovi koda koji su sastavni dio ove aplikacije.

1.1 Zadatak završnog rada

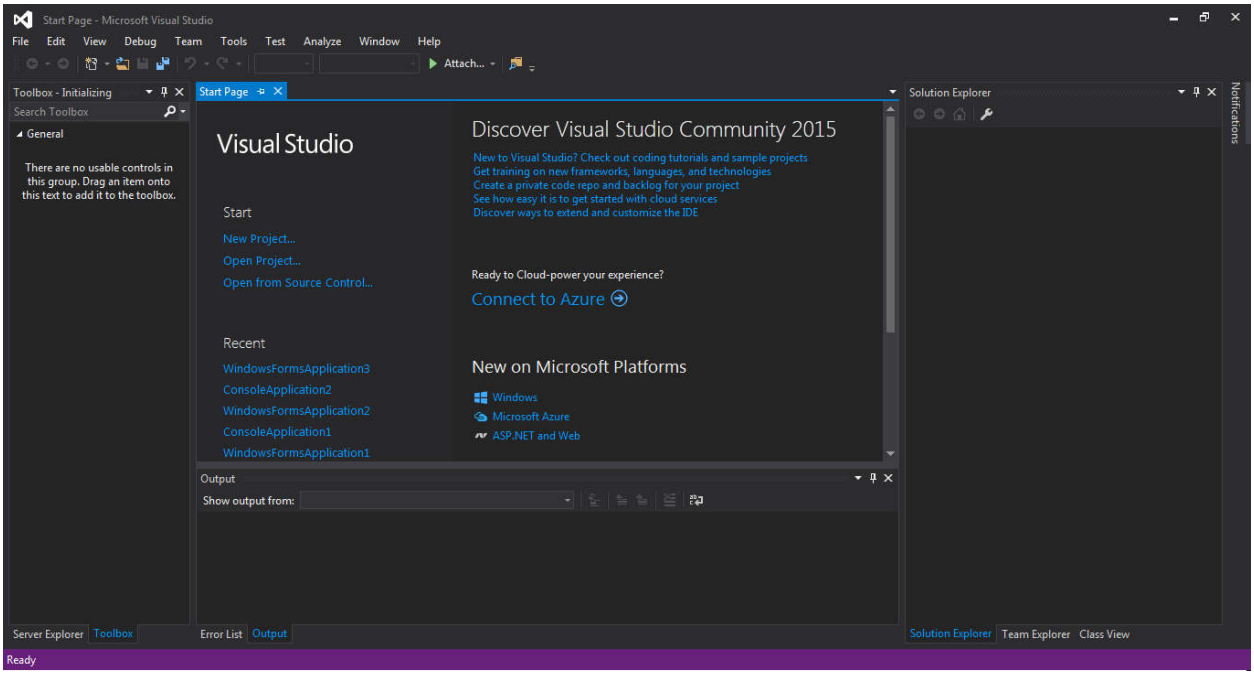
Zadatak završnog rada je napraviti C# aplikaciju koja će pripomoći poslodavcima voditi bazu podataka o zaposlenicima u tvrtci. Poslodavci će, osim unošenja podataka u bazu podataka, moći provjeravati unešene zaposlenike te zadavati poslove svakom od zaposlenika. Aplikacija je pisana C# jezikom, izgrađena je u programskom okruženju Microsoft Visual Studio i koristi se Redis bazom podataka.

2. ALATI

2.1 Microsoft Visual Studio

Microsoft Visual Studio je razvojno okruženje razvijeno od strane Microsofta. Koristi se najviše za razvoj računalnih aplikacija i web stranica. Prva verzija Visual Studija izašla je 1997. godine, najvažnija stvar koju je sadržavala je podržavanje C++ programskog jezika. Najnovije verzije Visual Studija sadrže integrirani debugger koji radi na razini izvornog koda, alate za pomoć dizajniranja aplikacija s grafičkim korisničkim sučeljem, koje će biti korišteno u ovom radu, i web stranica.

Visual Studio je pogodan za razvoj raznoraznih aplikacija i programa iz razloga što podržava mnogo različitih programskih jezika. Neki od tih programskih jezika su C, C++, C# (koji će biti korišten za ovaj rad), Python i mnogi drugi. Osim tih programskih jezika, postoji i mogućnost instaliranja jezičnih servisa uz pomoć kojih dobivamo puno više novih mogućnosti.



Sl. 2. 1 Početna stranica Microsoft Visual Studio okruženja

2.2 C#

C# je programski jezik napisan od strane Andresa Hejlsberga i tima koji je okupio oko sebe 1999. godine. U početku se zvao „Cool“ (C-like Object Oriented Language), a nakon godinu dana mijenja naziv u C#. Sintaksa C# jezika je vrlo slična sintaksi C i C++ jezika.

Tipovi podataka koji se mogu koristiti u C# jeziku su također slični poput C i C++ jezika. Osnovni tipovi su char, int, short, long, float, double, bool, string, decimal. Osim tih tipova postoje i izvedeni tipovi, kao što su naprimjer uint, ushort i ulong. Svaki od tih tipova sprema određeni tip podatka i zauzima određenu količinu memorije u računalu. U tablici 2.1 može se vidjeti što koji tip predstavlja i koliko memorije zauzimasvaki od osnovnih tipova.

Tip podatka	Vrsta podatka	Memorija (bitovi)
char	jedan znak	16
int	cijeli broj	32
short	cijeli broj	16
long	cijeli broj	64
float	cijeli broj jednostruke preciznosti	32
double	cijeli broj dvostruke preciznosti	64
bool	logički podatak	8
string	slijed znakova	/
decimal	decimalni broj	128

Tablica 2. 1 Popis tipova podataka

Osim tipova podataka, važno je napomenuti i operacije koje su omogućene u C#. Operacije se mogu podijeliti na aritmetičke (zbrajanje, oduzimanje, množenje, dijeljenje, inkrementiranje, dekrementiranje itd.), relacijske (jednako ==, nije jednako !=, veće >, manje <, veće/jednako >=, manje/jednako <=), logičke (negacija !, logički I &, logički ILI | itd.).

Za ispis na konzolu koristi se naredba **Console.WriteLine(„tekst koji želimo ispisati“);**. Ako želimo od korisnika preuzeti neki upis s tipkovnice i spremiti ga u varijablu x, to možemo napraviti idućom naredbom **string x = Console.ReadLine();**.

U Visual Studiju postoji nekoliko načina za primjenu C# programskog jezika, 2 glavna načina su kao aplikacija i kao program u konzoli. U ovom radu se koristila C# aplikacija, za koju je procijenjeno da će dati najbolje rezultate za zadani problem.

2.3 Redis

Redis je jedna od najzastupljenijih NoSQL baza podataka. Redis predstavlja REmote DIctionary Server. Razvijen je 2009. godine, razvio ga je Salvatore Sanfilippo u samo 20 000

linija koda. Pisan je u ANSI C jeziku. Baza podataka je „open-source“, a jedna od većih prednosti je to što je u potpunosti besplatna.

Redis je baza podataka koja radi sa sljedećim strukturama podataka:

1. Stringovi

2. Skupovi

3. Sortirani skupovi

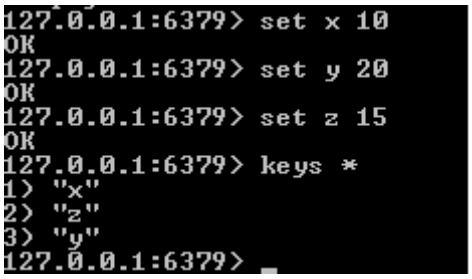
4. Hashevi

5.Liste

Svaki od ovih tipova struktura podataka ima svoje prednosti i nedostatke, svaki tip je dobar za određeni tip podataka koje korisnik želi spremiti.Za spremanje svakog određenog tipa podatka postoje naredbe koje se upisuju.

STRING

Stringovi predstavljaju najjednostavniji tip podatka koji se sastoji od 2 jednostavna dijela, ključa i vrijednosti. Za stringove naredba za postavljanje vrijednosti je **set ključ vrijednost**, a ako tu vrijednost korisnik želi ispisati naredba je **get ključ**. Ako želimo ispisati sve ključeve koje smo zapisali u neku bazu podataka, koristimo naredbu **keys ***.

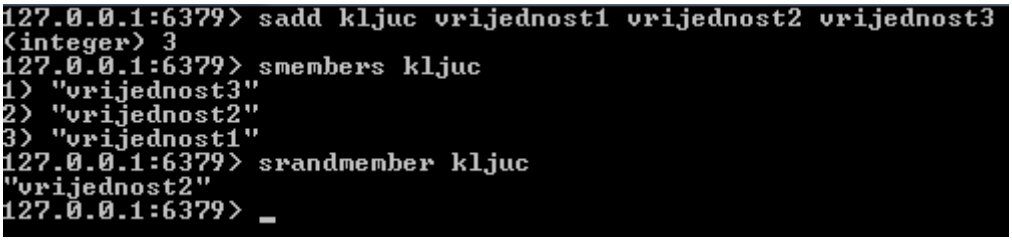


Sl. 2.2 Osnovne naredbe

SKUP

Skup je neuređena kolekcija varijabli i pripadnih vrijednosti. Svaka od varijabli može imati jednu ili više pripadnu vrijednost. Kako bi se zapisao skup u bazu podataka, koristi se naredba **sadd varijabla vrijednost vrijednost**, za primjer ako želimo zapisati dvije vrijednosti za određenu varijablu. Za ispisivanje svih vrijednosti određene varijabli koristimo naredbu

smembers varijabla. Vezano uz skupove, postoje još raznorazne naredbe kojima možemo pretraživati presjek ili uniju između 2 različita skupa (skupovi s različitim nazivom varijabli). Za ispis random vrijednosti iz varijable koristimo naredbu **srandmember varijabla**. Glavna prednost skupova je u tome što sam Redis ne dopušta ponavljanje iste vrijednosti u određenoj varijabli, tako da programer ne mora brinuti o uvjetima za ponavljanje vrijednosti prilikom pisanja programa.



Sl. 2.3 Naredbe vezane uz skup

SORTIRANI SKUP

Sortirani skup je uređena kolekcija varijabli i pripadnih vrijednosti. Kao i običan skup, sortirani skup ne može imati ponavljajuće elemente. Glavna naredba za dodavanje elemenata u sortirani skup je **zadd**, a osim toga, postoje naredbe pomoći kojih možemo pronaći određene elemente u skupu, kao naprimjer broj elemenata između minimalne i maksimalne vrijednosti (**zcount ključ min max)**.

HASH

Hash je niz podataka koji ne zauzima puno prostora u memoriji, ključevi i varijable su nizovi stringova, a unutar jednog hasha može se nalaziti više ključeva. Naredba za upis podataka u hash je **hset naziv ključ vrijednost**, dok se za upis više ključeva i vrijednosti unutar hasha koristi naredba **hmset naziv ključ1 vrijednost ključ2 vrijednost**. Za ispis svih vrijednosti nekog hasha koristi se naredba **hgetall naziv**.

LISTA

Lista predstavlja niz stringova. Elementi su sortirani redoslijedom dodavanja. Za dodavanje vrijednosti na početak liste koristi se naredba **lpush varijabla vrijednost,** dok se za dodavanje vrijednosti na kraj liste koristi naredba **rpush varijabla vrijednost**. Za brisanje zadnjeg upisanog unutar liste koristi se naredba **lpop varijabla**.

Neke od glavnih prednosti Redisa kao baze podataka su brzina izvršenja naredbi, jednostavnost u konfiguraciji i učenju, jednostavnost operacija nad bazom podataka i aktivan razvoj. Redis se svakodnevno razvija, *open-source* je i dostupna je svima.

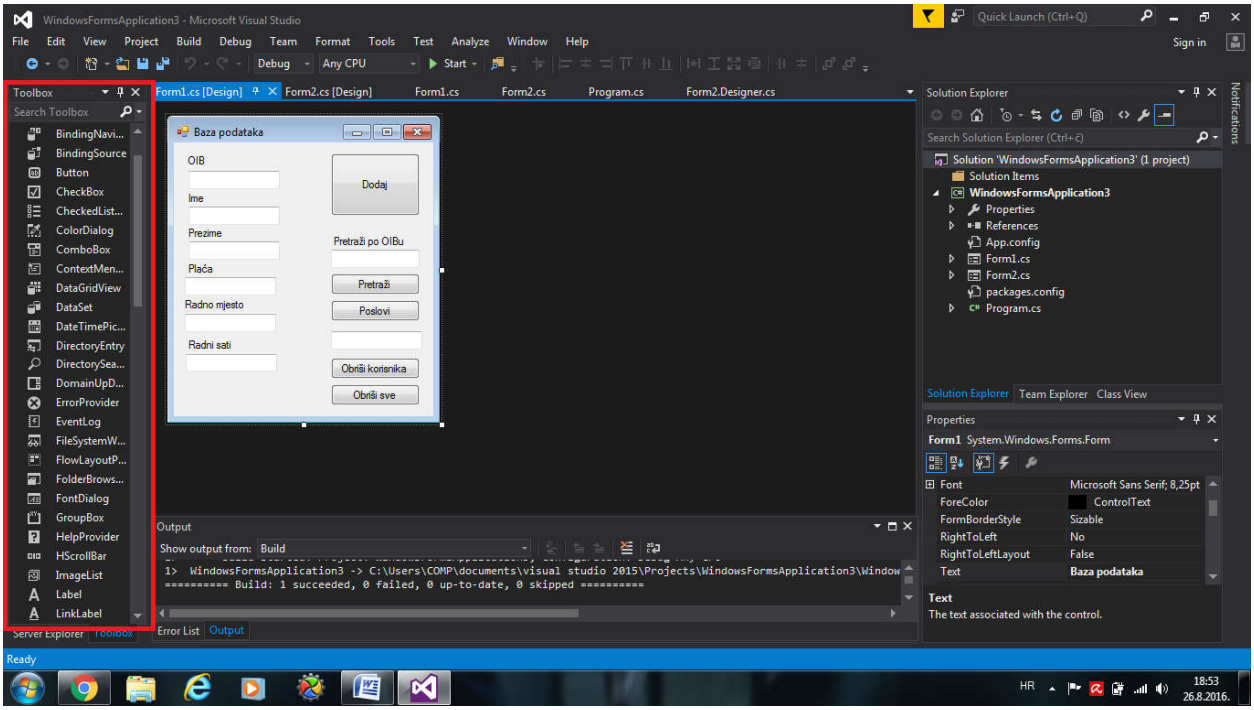
Nedostaci Redisa, kao i ostalih NoSQL baza podataka, je u tome što postoje problemi s administracijom baze podataka, ne postoji standardizacija i ne zadovoljavaju se ACID svojstva (atomarnost, konzistentnost, izolacija i trajnost).

3. APLIKACIJA

Kako bi zadatak bio uspješno izvršen, za početak se treba analizirati zadani problem. U suradnji s mentorom dogovoreno je da se način funkcioniranja Redis baze podataka prikaže uz pomoć C# aplikacije koja će koristiti poslodavcu za unos podataka o svojim radnicima.

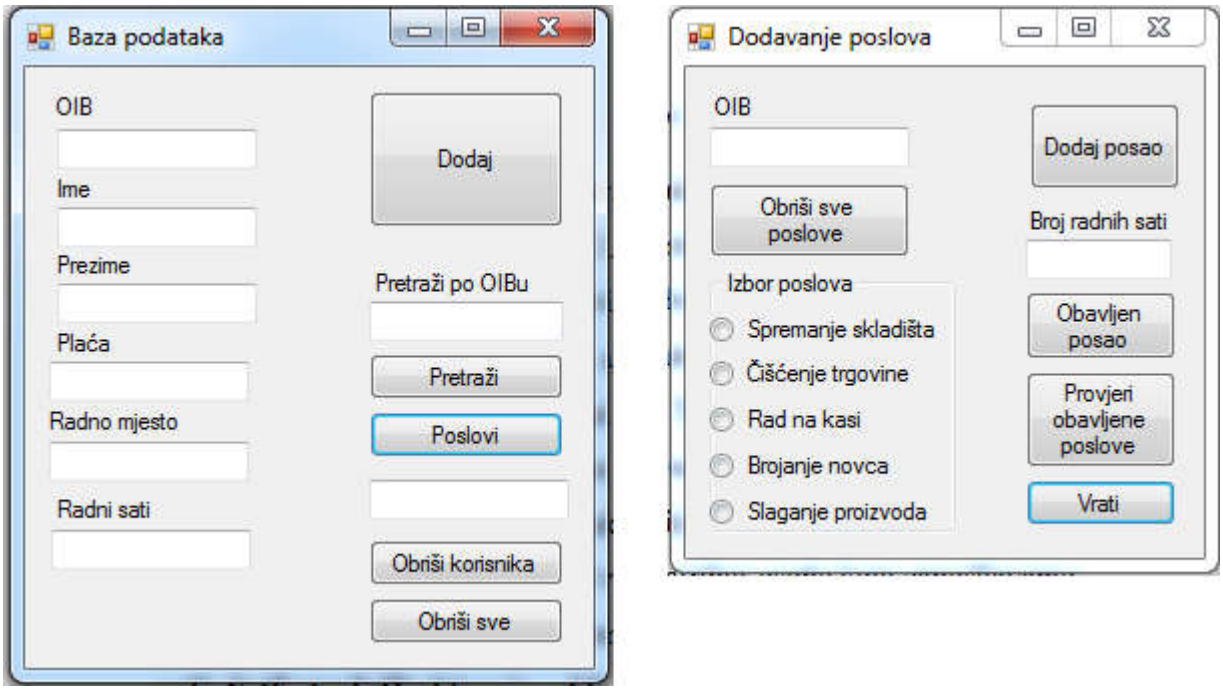
Prije pokretanja aplikacije potrebno je pokrenuti redis-server i zatim redis-client iz WinRar datoteke prethodno skinute s adrese http://redis.io/download.

Za početak se mora napraviti vizualni izgled aplikacije koja mora biti jednostavna za uporabu. Potrebno je u C# aplikaciji dodati određene textboxove, labele i buttone. Na slici 3.1 se može vidjeti izgled aplikacije i u crvenom kvadratu je označen izbornik „Alati“ gdje se može odabrati puno različitih opcija za aplikaciju. Također, u dogovoru s mentorom napravljen je i drugi dio aplikacije čiji zadatak je da obavlja sve funkcije vezane uz zadavanje poslova određenom zaposleniku.



Sl. 3. 1 Izgled aplikacije i izbornik „Alati“

Na slici 3.2 može se vidjeti izgled oba dijela aplikacije. Prvi dio aplikacije zadužen je za dodavanje u bazu podataka, a drugi dio za zadavanje poslova zaposlenicima.



Sl. 3.2 Izgled aplikacije

Gumb „Dodaj“ ima ulogu dodavanja osobe u bazu podataka. Kod koji opisuje postupak dodavanja može se vidjeti na slici 3.3. Kako bi se dodala osoba u bazu podataka, potrebno je prvo provjeriti jesu li svi textboxovi nužni za dodavanje osobe u bazu podataka popunjeni, ako jedan od textboxova nije ispunjen, aplikacija će izbaciti poruku upozorenja.Potrebno je i provjeriti postoji li već osoba u bazi podataka s tim OIBom. Također, nakon dodavanje osobe u bazu dobije se potvrda od programa da je korisnik dodan u bazu podataka. Kako bi dodali osobu u bazu podataka, potrebno je iz svakog TextBoxa preuzeti slijed slova koji je zapisan i spremiti ga u određenu varijablu. Svaka od varijabli predstavlja neki atribut osobe koju zapošljavamo (OIB, ime, prezime...) i tada to spremamo u redis bazu podataka pomoću naredbe **redis.SetEntryInHash(naziv, „ključ“, vrijednost).**

```
private void dodaj_Click(object sender, EventArgs e)
{
    using (var redis = new RedisClient())
        if (tb_oib.Text == "" || tb_ime.Text == "" || tb_prez.Text == "" || tb_placa.Text == "" || tb_mjesto.Text == "" || tb_sati.Text ==
        {
            MessageBox.Show("Niste dobro unijeli podatke!", "Greška!");
        }

        else if (redis.SetEntryInHashIfNotExists(oib, "Ime", ime))
        {
            redis.SetEntryInHash(oib, "Ime", ime);
            redis.SetEntryInHash(oib, "Prezime", prezime);
            redis.SetEntryInHash(oib, "Placa", placa);
            redis.SetEntryInHash(oib, "Radno mjesto", mjesto);
            redis.SetEntryInHash(oib, "Broj radnih sati", sati);
            tb_oib.Text = String.Empty;
            tb_ime.Text = String.Empty;
            tb_prez.Text = String.Empty;
            tb_placa.Text = String.Empty;
            tb_mjesto.Text = String.Empty;
            tb_sati.Text = String.Empty;
            MessageBox.Show("Dodali ste osobu u bazu podataka.");
        }

        else MessageBox.Show("Već postoji osoba u bazi podataka s tim OIBom.");
}
```

Sl. 3.3 Kod za dodavanje osobu u bazu podataka

Osim gumba „Dodaj“, postoje i gumbovi „Pretraži“ čija funkcija je da pretraži postoji li osoba s određenim OIBom već u bazi podataka. Ako osoba postoji, korisnik dobije poruku na ekranu sa svim podacima o toj osobi. Kod koji pretražuje osobu u bazi podataka i ispisuje na ekran može se vidjeti na slici 3.5.

```
private void pretraga_Click(object sender, EventArgs e)
{
    string provjeri = Convert.ToString(tb_trazi.Text);
    using (var redis = new RedisClient())
    {
        if (redis.HashContainsEntry(provjeri, "Ime")==false)
        {
            tb_trazi.Text = String.Empty;
            MessageBox.Show("U bazi podataka ne postoji osoba s navedenim OIBom.", "Greška!");
        }

        else
        {
            tb_trazi.Text = String.Empty;
            string pime = redis.GetValueFromHash(provjeri, "Ime");
            string pprezime = redis.GetValueFromHash(provjeri, "Prezime");
            string pplaca = redis.GetValueFromHash(provjeri, "Placa");
            string pradno = redis.GetValueFromHash(provjeri, "Radno mjesto");
            string psati = redis.GetValueFromHash(provjeri, "Broj radnih sati");
            MessageBox.Show(string.Format("Postoji osoba s navedenim OIBom.\n Ime: {0} \n Prezime: {1} \n Plaća: {2} \n Radno mjesto: {3} \n Broj radnih sat
        }
    }
}
```

Sl. 3.5 Kod za pretraživanje baze podataka

Također postoji i gumb „Obriši sve“ čiji zadatak je da obriše sve osobe iz baze podataka, u slučaju da poslodavac želi započeti neku novu bazu podataka ili jednostavno obrisati sve. Da ne bi došlo do krivog pritiska gumba, nakon što se stisne gumb, izbaci se izbornik sa Da/Ne opcijama kako bi korisnik potvrdio brisanje cijele baze podataka. Na slici 3. 6 može se vidjeti kod za brisanje cijele baze podataka.

```
private void brisisve_Click(object sender, EventArgs e)
{
    using (var redis = new RedisClient())
    {
        DialogResult brisanje = MessageBox.Show("Jeste li sigurni da želite obrisati sve podatke iz baze podataka?",
              "Brisanje baze podataka", MessageBoxButtons.YesNo);
        switch (brisanje)
        {
            case DialogResult.Yes:
                redis.FlushAll();
                MessageBox.Show("Obrisali ste sve podatke iz baze podataka.");
                break;
            case DialogResult.No:
                MessageBox.Show("Niste obrisali sve podatke iz baze podataka.");
                break;
        }

    }
}
```

Sl. 3.6 Kod za brisanje svih podataka iz baze

Gumb „Poslovi“ otvara novi prozor aplikacije gdje postoje opcije za zadavanje poslova, ispis poslova i za vraćanje na prethodni prozor. Za svaku od tih opcija prvo se mora provjeriti postoji li već osoba u bazi podataka. Za biranje određenih poslova korišteni su *RadioButtoni* koji rade na principu da se samo jedan od elemenata može označiti. Za svaku naredbu koju korisnik odabere dobije potvrdu da je obavljena.

```
else if (radioButton1.Checked)
{
    if (redis.HashContainsEntry(oib1, "Obavljeni poslovi") == false)
    {
        redis.SetEntryInHash(oib1, "Obavljeni poslovi", "Spremanje skladišta");
        redis.RemoveEntryFromHash(oib1, "Posao u tijeku");
        redis.SetEntryInHash(oib1, "Broj potrebnih sati za posao", radnisati);
        MessageBox.Show("Posao je obavljen!");
        tb_oib1.Text = String.Empty;
        tb_radnisati.Text = String.Empty;
    }

    else
    {
        string x = redis.GetValueFromHash(oib1, "Obavljeni poslovi") + ", " + "Spremanje skladišta";
        redis.SetEntryInHash(oib1, "Obavljeni poslovi", x);
        redis.SetEntryInHash(oib1, "Broj potrebnih sati za posao", radnisati);
        MessageBox.Show("Posao je obavljen!");
        tb_oib1.Text = String.Empty;
        tb_radnisati.Text = String.Empty;
    }
}
```

Sl. 3.7 Primjer korištenja *RadioButtona*

Kao što se može vidjeti iz navedenih dijelova koda, kako bi baza podataka funkcionirala potrebno je koristiti, uz sintaksu C# jezika, i sintaksu Redisa. Svaka od naredbi Redisa je opisana na stranici [1]. Neke od naredbi koje su korištene u aplikaciji su **redis.FlushAll()** čiji zadatak je obrisati cijelu bazu podataka, **redis.GetValueFromHash(naziv, „ključ“)** koja daje vrijednost

elementa varijable „ključ“ i **redis.HashContainsEntry(naziv, „ključ“)** koja je tipa *bool* i koristi se najviše unutar petlji.

4. ZAKLJUČAK

Cilj ovog završnog rada bio je napraviti funkcionalnu bazu podataka koja će omogućiti poslodavcu jednostavno vođenje evidencije o svojim zaposlenicima. Glavni problem rada je bio naučiti sintaksu i način korištenja Redisa kao baze podataka, uz to je problem bio konstruirati aplikaciju i učiniti ju funkcionalom. Moralo se za svaki pritisak na *button* uzeti u obzir koji *textboxovi* su ispunjeni, to jest, moralo se uzeti u obzir i to je li određeni OIB već zapisan u bazi podataka, ako je, što dalje, a ako nije, obavijestiti korisnika o pogrešci. U dogovoru s mentorom dodan je dio o zadavanju poslova, a izbor poslova je odabran nasumično, samo kako bi se pokazao način funkcioniranja *radiobuttona.*Aplikacija je vrlo jednostavna za korištenje. Poslodavac mora znati neke osnovne informacije o svakom zaposleniku kako bi ga evidentirao u bazu podataka.

Korišteni programi za aplikaciju su dostupni svakome tko ima pristup računalu te je svaki od programa besplatan za korištenje, osim u slučaju korištenja aplikacije u komercijalne svrhe. Također, na internetu postoji relativno puno uputa kako koristiti svaki od navedenih programa, s raznoraznim primjerimakorištenja i načinima kako doći do rješenja nekog problema. Prilikom razvijanja ove aplikacije bilo je potrebno osnovno znanje C# jezika i korištenja Visual Studija. Aplikacija je uspješno napravljena, u dogovoru s mentorom je svaki dio zadatka obavljen bez nekih većih poteškoća.

LITERATURA

[1] https://github.com/ServiceStack/ServiceStack.Redis/wiki/IRedisClient (posjećeno 22. lipnja 2016.)

[2] https://en.wikipedia.org/wiki/Redis (posjećeno 10. lipnja 2016.)

[3] http://www.milanpopovic.me/redis-moc-jednostavnosti/ (posjećeno 20. lipnja 2016.)

[4] http://redis.io (posjećeno 10. lipnja 2016.)

[5] https://en.wikipedia.org/wiki/Microsoft_Visual_Studio (posjećeno 12. lipnja 2016.)

[6] https://en.wikipedia.org/wiki/C_Sharp_(programming_language) (posjećeno 13. lipnja 2016.)

[7] http://mrav.ffzg.hr/zanimanja/book/part2/node1407.htm (posjećeno 15. lipnja 2016.)

[8] http://www.tutorialspoint.com/csharp/csharp_data_types.htm (posjećeno 13. lipnja 2016.)

SAŽETAK

Glavni cilj ovog rada je bio napraviti aplikaciju koja će pojednostaviti vođenje baze podataka o zaposlenicima. Aplikacija omogućava poslodavcu jednostavan pregled određenih zaposlenika i informacije o njima. Također dopušta poslodavcu zadavanje poslova svakom zaposleniku, ispis obavljenih poslova pojedinog zaposlenika, kao i brisanje svih obavljenih poslova zaposlenika. Uz to, poslodavac ima mogućnost, nakon što da otkaz zaposleniku, obrisati ga također i u bazi podataka. Može se obrisati i cijela baza podataka, u slučaju da poslodavac želi prenamijeniti bazu podataka i krenuti ispočetka. Aplikacija je napravljena u Windows okruženju, uzevši u obzir da veliki dio današnjih poslodavaca koristi upravo to okruženje. Teoretski dio sadrži opis temeljnih obilježja programa Microsoft Visual Studija, programskog jezika C#, kao i temeljna obilježja i način funkcioniranja Redisa kao baze podataka.

Ključne riječi: Microsoft Visual Studio, C#, Redis, aplikacija, poslodavac, zaposlenik

ABSTRACT

Redis database implemented with C# application

Main goal of this assigment was to make an application that will simplify maintaining the database about the employees. Application helps employer with simplified view about the employees and it also can show the information about the employees. It also allow employer to assign the tasks for employees, to print finished tasks of every individual employee and to delete all the assigned and finished tasksfro every employee. Employer also has the possibility, after he fire employee, to delete that employee from database. The whole database also can be fully deleted, in case the employer wants to reassign the database and start from the beggining. Application is made in Windows surrounding, knowing that most of employers use that surrounding. Theoretical part contains the description of basic features of Microsoft Visual Studio, C# programming language, and the basic features and functioning of Redis as a database.

Keywords: Redis, database, application, Visual Studio, C#

ŽIVOTOPIS

Tomislav Šarčević rođen je 20. svibnja 1994. godine u Đakovu, Hrvatska. Živi u Đakovu, na adresi Pavićeva 7. 2001. godine počinje s osnovnoškolskim obrazovanjem u OŠ Vladimir Nazor u Đakovu. Nakon završetka osnovnoškolskog obrazovanja, 2009. se upisuje u Gimnaziju Antuna Gustava Matoša u Đakovu, smjer Opća gimnazija. 2013. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija, smjer Računarstvo, koji još uvijek pohađa. Dobitnik je stipendije za deficitarna zanimanja Nacionalne zaklade za potporu učeničkom i studentskom standardu za akademsku godinu 2015./2016. Također posjeduje određeno znanje u govoru, čitanju i pisanju engleskog jezika, vozačku dozvolu B kategorije. Posjeduje i određeno znanje u radu s Microsoft Office alatima, osnovno znanje opisnog jezika HTML, osnovno znanje programskih jezika C, C++ i C#.

Tomislav Šarčević

PRILOZI

DVD

- Microsoft Visual Studio projekt
- Rad u .docx i .pdf formatu