



Univerzitet u Novom Sadu
Tehnički fakultet »Mihajlo Pupin«
Zrenjanin



IZVEŠTAJ O REALIZOVANOM SOFTVERU U OKVIRU STRUČNE PRAKSE U GRADSKOJ UPRAVI GRADA ZRENJANINA

Tema: Izrada Desktop C# softverske aplikacije za pretragu dokumenata

mentor: Mirča Sekošan
komentor: Doc. dr. Ljubica Kazi

Student: Miloš Trošić IT 21/13
Smer: Informacione tehnologije, osnovne studije

Zrenjanin, 2017.

Contents

1. Projektni zadatak.....	3
2. Korisničko uputstvo	4
3. Proces kreiranja aplikacije	6
3.1. Opis primenjene tehnologije	6
3.2. Opis kreiranja aplikacije	6
4. Zaključak.....	8
5. Literatura.....	9

1. Projektni zadatak

Na osnovu sporazuma koji su Gradska uprava Grada Zrenjanina i Tehnički fakultet „Mihajlo Pupin“ Zrenjanin potpisali u vezi stručne prakse koju bi studenti Tehničkog fakulteta „Mihajlo Pupin“ Zrenjanin realizovali u okviru organizacionih jedinica Gradske uprave Grada Zrenjanina, 20.3.2017. godine organizovan je sastanak studenata, predstavnika Gradske uprave Grada Zrenjanina i predstavnika Tehničkog fakulteta „Mihajlo Pupin“ Zrenjanin radi definisanja projektnih zadataka.

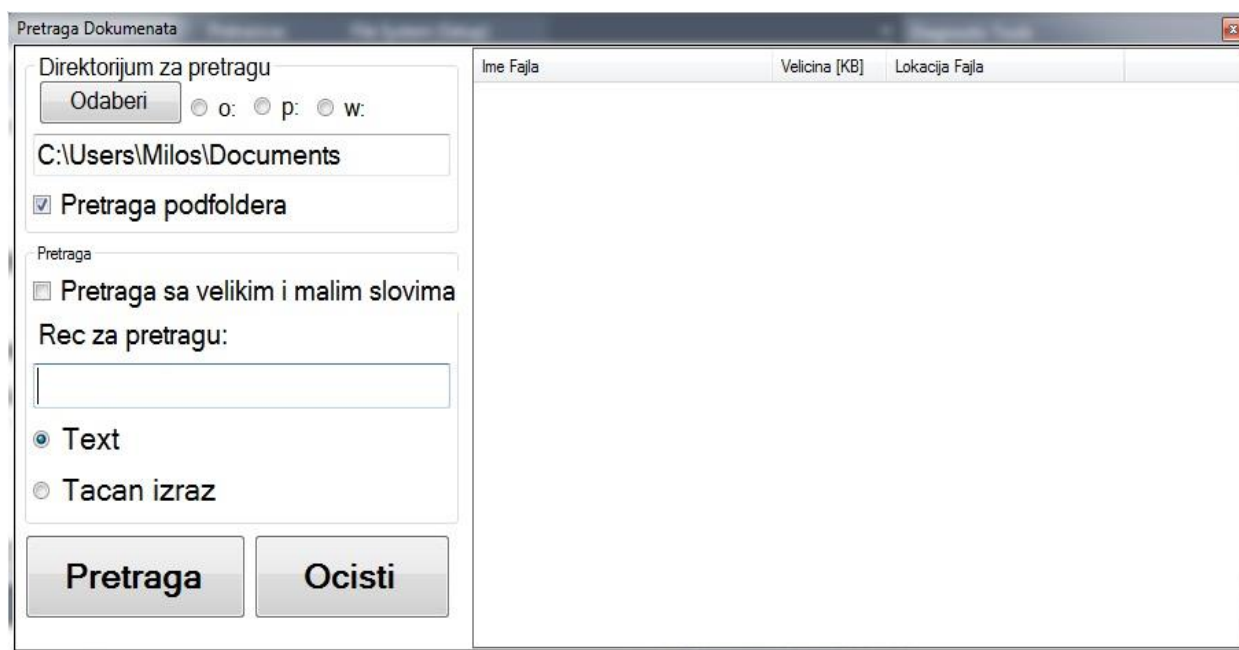
U ovom radu, projektni zadatak je definisan na sledeći način:

Problem koji treba rešiti je izrada računarskog programa za pretragu većeg broja dokumenata prema raznim ključnim rečima. Potrebno je softversku aplikaciju uraditi i prilagoditi potrebama Gradske uprave Grada Zrenjanina, prvenstveno radi pretrage odluka Gradske skupštine, ali sa mogućnošću korišćenja i od strane drugih organizacionih jedinica Gradske uprave Grada Zrenjanina. Potrebno je realizovati softversku aplikaciju u saradnji sa IT sektorom Gradske uprave. Aplikacija treba korisniku da omogućava pretragu reči unutar word fajlova. Aplikacija treba da je prilagodjena i za novije i za starije verzije word-a, tačnije za fajlove sa ekstenzijom .doc i .docx.

2. Korisničko uputstvo

Korisnik ima mogućnost rada u jednoj formi u kojoj se nalaze sledece dodatne opcije:

1. Selektovanje foldera ili celog direktorijuma.
Čekiranjem ove opcije omogućuje se pretraga svih fajlova koji se nalaze unutar foldera.
2. Pretraga sa velikim i malim slovima
3. Pretraga podfoldera



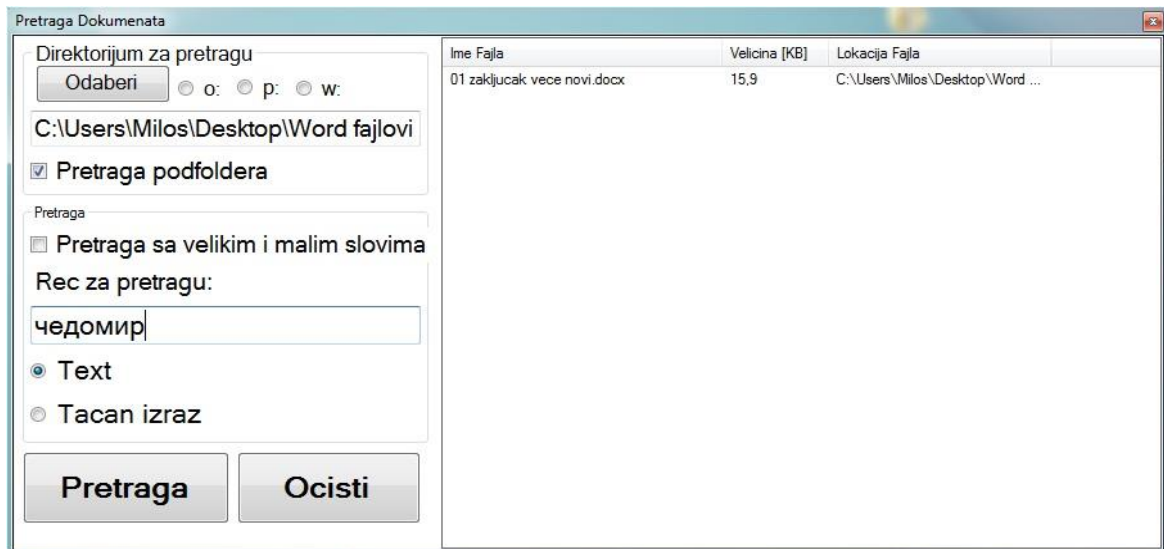
Slika 1. Izgled Aplikacije.

Nakon pokretanja aplikacije korisnik dobija početnu formu koja izgleda kao na Slici 2.



Slika 2. Izgled početne forme

Prvi korak nakon pokretanja je odabir direktorijuma za pretragu. To se vrši klikom na dugme Odaberi ili klikom na jedno od 3 radio button-a. Ukoliko zelimo pretragu podfoldera, onda se čekira „Pretraga podfoldera“. Nakon toga u praznom prozoru ispod „Rec za pretragu“ kucamo tekst koji zelimo pronaći u odabranom folderu. Zatim klikom na dugme „Pretrazi“ vrsimo pretragu i izlistavanje svih word fajlova u kojima se nalazi ukucani tekst. Slika 3. prikazuje izgled aplikacija nakon izlistavanja.



Slika 3. Aplikacija nakon izlistavanja

Čekiranjem „Pretraga sa velikim i malim slovima“ uključujemo pretragu koja izlistava dokumenta koja imaju tacan raspored sa velikim i malim slovima. Nakon izlistavanja dokumenata, klikom na konkretan dokument otvara se MS Word u okviru kog se prikazuje taj dokument.

3. Proces kreiranja aplikacije

3.1. Opis primenjene tehnologije

Za kreiranje ove aplikacije korišten je programski jezik C# i .NET okruženje.

Microsoft.Net framework je softverska platforma koja se može instalirati na bilo kom računarku koji ima microsoft operativni sistem. Microsoft.Net sadrži veliku biblioteku kodova za uobičajne probleme u programiranju i virtuelnu mašinu koja upravlja izvršavanjem programa pisanih za .Net. .Net podržava rad sa više programskih jezika poput Visual Studio ili Java. Kako bi moglo da se radi u .Net-u pored njega je potrebno instalirati SDK ili *Software development kit* i naravno program u kojem će se programirati.

Baza poseduje širok spektar mogućnosti u koje se ubrajaju rad sa bazama, podacima, interfejsom itd. Biblioteka klasa se koristi od strane programera koji ga u kombinaciji sa svojim kodom koriste u izradi raznih aplikacija poput ove.

Programi pisani za .Net se izvršavaju u CLR-u ili poznatijem kao Common Language Runtime, koji obezbeđuje izgled virtuelne mašine same aplikacije tako da programeri ne moraju da razmišljaju o specijalnim procesorima koji će izvršiti program. Zadatak virtuelne mašine je prevođenje instrukcija koje unese korisnik u mašinski jezik koji razume računar i to je odlika programskih jezika viših nivoa. Ukupno gledano CLD i biblioteka klasa čine .Net.

Kad je u pitanju programski jezik C#, to je jedan od novijih programskih jezika razvijen od strane Microsoft-a. C# je objektno orjentisan programski jezik kao i C++ i Java. Jezik je smišljen za pravljenje aplikacije koje podržavaju upravo .Net okruženje

Bitnije karakteristike C# su:

- ne postoji ni globalne metode niti promenljive, sve što je na raspolaganju jeste ono što je korisnik uneo unutar klasa,
- ne postoji više nasleđivanja, jedna klasa može da bude potomak samo jednoj klasi
- poseduje različite konverzije između različitih tipova (int u float) itd.
- jezik je case sensitive, razlikuje velika od malih slova

3.2. Opis kreiranja aplikacije

Osnovni koncept aplikacije preuzet je sa LINKA [(Sourceforge, n.d.)]. S obzirom da je dat slobodno (besplatno) kao izvorni kod, data je mogućnost izmene.

Na početnoj formi imamo sledeće elemente (neki od elemenata su dodati u okviru izmene):

1. Dugme „Pretraži“.
2. Dugme „Očisti“.
3. Dugme „Odaberi“
4. Dva Text Box-a
5. Dva Check Box-a

6. Dve grupe radio button-a.
7. List View – za prikaz rezultata.

1. Klikom na dugme Pretraži pokrece se akcija btnSerach_Click. Prvo se izvrsava sledeca linija koda:

```
this.resultListView.Items.Clear();
```

Ovim se brisu svi prehodni rezultati iz ListView-a. Nakon toga se metodom IndexOf upoređuje uneti tekst sa tekстом unutar fajlova. Zatim se izlistavaju fajlovi koji sadrže uneti tekst.

```
private static IEnumerable<string> Search(string directory, string searchString, bool searchSubdirectories, bool caseSensitive, bool useRegex)
{
    var isMatch = useRegex ? new Predicate<string>(x => Regex.IsMatch(x,
searchString, caseSensitive ? RegexOptions.None : RegexOptions.IgnoreCase))
    : new Predicate<string>(x => x.IndexOf(searchString, caseSensitive ?
StringComparison.Ordinal : StringComparison.OrdinalIgnoreCase) >= 0);

    foreach (var filePath in Directory.GetFiles(directory, "*.doc", searchSubdirectories ?
SearchOption.AllDirectories : SearchOption.TopDirectoryOnly))
    {
        string docxText;

        TextExtractor extractor = new TextExtractor(filePath);
        docxText = extractor.ExtractText();

        if (isMatch(docxText))
            yield return filePath;}}}
```

Da bi ovo radilo, pre toga moramo da dodamo u projekat code7248.Word_Reader. To je dll fajl u kome se nalazi metoda za citanje word fajlova (NAVEDENI DLL FAJL JE PREUZET SA LINKA2 [(codeproject, n.d.)] i slobodan je za koriscenje. Desnim klikom na projekat zatim

Add -> Reference. Takodje pre namespace-a moramo dodati dll.

```
using Code7248.word_reader;
```

Nakon izlistavanje fajlova, svakom fajlu mozemo pristupiti duplim klikom na fajl. Ovo se postize metodom resultListView_ItemActivate.

```
string filePath = ((ListView)sender).SelectedItem[0].SubItems[2].Text;
if (File.Exists(filePath))
    Process.Start(filePath);
```

Brisanje izlistanih fajlova se vrši klikom na dugme Ocisti. Kod koji se izvršava klikom na dugme Ocisti je sledeci:

```
resultListView.Items.Clear()
```

4. Zaključak

Radeći na ovoj aplikaciji upoznao sam se sa programskim jezikom C#. U ranijim godinama studija upoznao sam se sa Visual Studio programom, tako da mi je rad sa bazom i najjednostavnijim komponentama poput textbox, label, combobox bio poznat. U ovoj aplikaciji upoznala sam se sa radom sa word dokumentima.

Što se tiče same aplikacije, podložna je promenama i unapređivanju, moguće je dodati još elemenata u zavisnosti od želje i potreba korisnika. Rad sa aplikacijom nije komplikovan. Aplikacija je prilagodjena svim korisnicima tako što upućuje korisnika koje podatke treba da upiše, ukoliko ne unese sve podatke ili unese pogrešan tip podatka, ispisuje se obaveštenje o greški. Tako da sa aplikacijom mogu raditi i osobe koje se ne bave programiranjem.

Po mom mišljenju je veoma bitan dizajn aplikacije kao i da ona bude prilagodjena svim korisnicima. Neophodno je da korisnici razumeju aplikaciju kako bi je što lakše i uspešnije koristili.

5. Literatura

1. <https://www.sourceforge.net/p/word-reader/wiki/Home/> - preuzeto - 10.05.2017
2. <https://www.codeproject.com/Tips/651158/Find-Text-in-Word-Documents> - 10.05.2017