Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/01BE35Paunovic

WPF IMPLEMENTACIJA KONFIGURABILNE RIBBON KONTROLE

WPF IMPLEMENTATION OF THE CONFIGURABLE RIBBON CONTROL

Srđan Paunović, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *U radu je prezentovan projekat u kom je implementirana komandna traka sa kontrolama, pomoću koje se ubrzava korišćenje aplikacije. Traka je korisnički interfejs koji povećava mogućnost otkrivanja aplikacionih mogućnosti i funkcija. Rad je realizovan u programskom jeziku C# korišćenjem WPF grafičkog podsistema.*

Ključne reči: *WPF, DataTemplate, CustomControls, UserControls, XML, Ribbon*

Abstract – *The document presents a project in which a command ribbon with controls is implemented, thereby speeding up the use of the application. The ribbon user interface increases discoverability of features and functions. The project was realized in the C# programming language using the graphical subsystem WPF.*

Keywords: *WPF, DataTemplate, CustomControls, UserControls, XML, Ribbon*

1. UVOD

Ribbon je komandna traka koja organizuje funkcije aplikacije u serije kartica (eng. tab) na vrhu aplikativnog prozora. Traka je korisnički interfejs koji pruža mogućnost lakšeg otkrivanja funkcionalnosti, omogućava brže učenje aplikacije i korisnici stiču osećaj veće kontrole prilikom korišćenja iste. Traka ujedno zamenjuje tradicionalne menije i trake sa alatkama. Sve funkcionalnosti su odjednom prikazane i korisnik na lakši način stiže do njih. Srodne funkcije su grupisane i postavljene u određeni tab. Implementirane funckije su prikazane preko dugmića koji su predstavljeni su ikonicom i nazivom. Naziv jasno govori o tome kakvo je njeno ponašanje, dok ikonica ilustruje to ponašanje. Prelaskom kursora preko dugmeta dobija se *ToolTip* koji detaljno opisuje koja je uloga funkcije. Zbog takvog načina prikazivanja uloge dugmeta naziv i nije obavezan, što je pogodno kod manjih dugmića. Veličina i pozicija dugmeta se određuju na osnovu toga koliko je ono važno za korišćenje aplikacije, kao i koliko se često koristi. Funkcije koje su relevantnije za aplikaciju i koje se češće koriste su predstavljene dugmićima koji su veći u odnosu na one funkcionalnosti koje se ili retko koriste ili nisu od posebnog značaja za domen problema kojim se sama aplikacija bavi.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bio dr Dragan Ivetić, red. prof.

Funkcije koje imaju iste ili slične osobine se mogu smestiti unutar grupe. Grupa ima svoj naziv koji bi trebalo da približno i asocijativno opisuje ponašanje funkcija unutar nje. Unutar jednog taba se, pored dugmića, mogu smestiti i grupe koje su slične na određenom nivou.. Naziv taba bi trebalo da asocira korisnika koje se to grupe i funkcije nalaze unutar njega. Kartice su raspoređene na sličan način kao što bi to bilo i u meniju. Organizovanje kartica na ovakav način pogodan je korisnicima koji su se već navikli na korišćenje starog menija i trake sa alatima, što je od velikog značaja jer korisniku omogućava komforniji, sigurniji i brži rad.

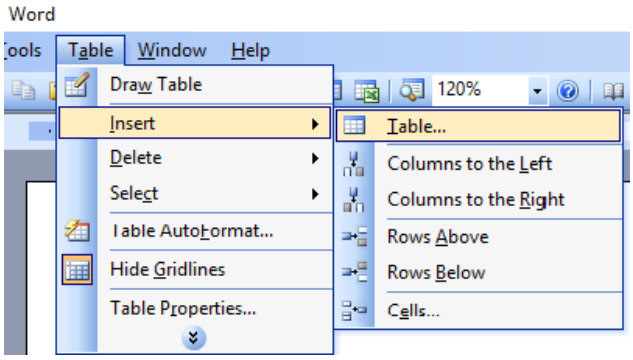
Jedna od većih prednosti *Ribbon* kontrole je ta da su korisniku prikazane samo one funkcije koje korisnik može da koristi u trenutno aktivnom prozoru. Na taj način se izbegavaju dugmići koji su konstantno prikazani u meniju i zauzimaju mesto, a nisu aktivni. Skup aktivnih kontrola se dinamički menja u zavisnosti od aktivnog konteksta u kom se aplikacija nalazi.

Traka takođe ima i funkciju sakrivanja, i u tom slučaju korisniku će se proširiti radni prostor. U sakrivenom režimu vidi se samo nazivi kartica). Klikom na određenu karticu prikazaće se *PopUp* meni koji prikazati sadržaj kartice. Na takav način se ne smanjuje radni prostor, a *PopUp* je vidljiv dokle god korisnik ne promeni fokus sa trake.

Prilikom smanjivanja veličine prozora aplikacije dolazi do transformisanja kontrola unutar trake. Tačnije, veće kontrole se smanjuju, dok se manje kontrole premeštaju u *PopUp* kontrolu unutar grupe u kojoj se prethodno kontrola nalazila. Ukoliko neka kontrola vizuelno ne može da stane u grupu kojoj pripada, premešta se u *PopUp*. Nakon što se premesti prva kontrola sakrije, pojavljuje se dugme pomoću kojeg je moguće otvoriti padajući prozor. Na taj način se pristupa skrivenim kontroloma. Traka prvo sakriva one manje prioritetne akcije, a one sa većim prioritetom pokušava da održi vidljive što je duže to moguće.

2. Ubrzavanje interakcije

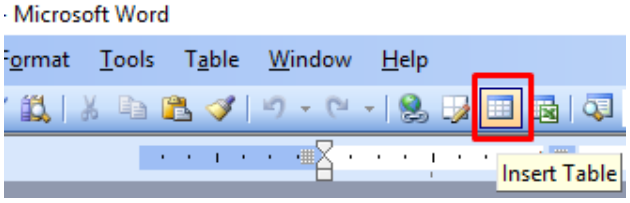
Kod starih menija sve se nalazilo sakriveno u svega par opcija, na koje se prvo moralo kliknuti kako bi se pokazao *PopUp* meni sa ostatkom mogućih akcija. Dolazak do željenih akcija zahtevao je kretanje kroz razgranato stablo menija i podmenija, što je oduzimalo prilično veliku količinu korisnikovog vremena. Primer korišćenja *Menubar*-a za umetanje nove tabele u dokument pomoću *Microsoft Word*-a verzije iz 2003. godine, prikazan je na Slici 1.



Slika 1 *Prikaz umetanja tabele iz menija u Microsoft Word-u iz 2003. godine*

Za izvršavanje pomenute akcije je bilo potrebno da korisnik prvo odabere opciju *Table* koja se nalazi na *Menubar*-u. Zatim se kursorom pomeri na *Insert* kako bi mu se pokazao ostatak funkcionalnosti, i tek na kraju odabere da ubaci tabelu. Opisani način je vremenski prilično zahtevan, iz razloga što korisnik mora da iz tri različita menija odabere željenu stavku.

Takav nedostatak je ispravio je *Toolbar*. Na *Toolbar*-u su se nalazile prečice frekventnijih akcija koje su se takođe već nalazile i u meniju. U sebi je sadržavao dugmad, koja su predstavljena ikonicom koja ilustruje ponašanje funckije. Primer umetanja tabele pomoću *Toolbar*-a prikazan je na Slici 2.



Slika 2 Prikaz umetanja tabele pomoću *Toolbar*-a u *Microsoft Word*-u iz 2003. godine

Upotreba *Toolbar*-a je drastično smanjila vreme potrebno za umetanje nove tabele, ali potencijalno samo za napredne korisnike koji već imaju iskustvo u radu sa aplikacijom ili nekom sličnom iz istog domena problema. Aplikacije koje u svom korisničkom interfejsu imaju i *Toolbar* pružaju korisnicima bar dva različita načina kako mogu da obave istu funkcionalnost: preko tradicionalnog menija ili upotrebom prečice. Međutim, ikonice su male i nekada nije jasno koja se tačno funkcionalnost krije iza određene prečice. Korisnik dok bi tražio prečicu koja odgovara funkcionalnosti koju želi da izvrši, morao je kursorom da prelazi preko svake ikonice i da čita *ToolTip*, kako bi bio siguran šta tačno ona radi. Ovo prelaženje kursorom je takođe povećavalo korisnikovo vreme potrebno za ispunjenje željene akcije. Iz tog razloga je uveden *Ribbon*.

Ribbon je nastao kao zamena za prethodno navedene kontrole. Traka izgledom podseća na obe kontrole. Kartice su raspoređene na sličan način kao što su bile glavne opcije na meniju. Svaka kartica u sebi sadrži grupe sa raznim kontrolama.

Samo je jedna kartica u jednom trenutku selektovana, a njegove kontrole su prikazane korisniku. Kontrole su veće i preglednije nego što su bile u *Toolbar*-u, što korisniku preciznije govori koja je njena svrha, i tako mu omogućava da na brži način dolazi do željenih akcija.

Primer umetanja tabele pomoću *Ribbon*-a prikazan je na Slici 3.

Slika 3 *Prikaz umetanja tabele pomoću Ribbon-a u Microsoft Word-u iz 2010. godine*

Ikonice na *Ribbon*-u su dosta veće i jasnije, što značajno doprinosi razumevanju koja je uloga akcije. Ispod dugmeta se nalazi naziv akcije, a prelaskom kursora preko ikonice se prikazuje detaljan opis koja je njena uloga.

Procena efikasnosti izvršavanja definisanog zadatka (umetanje tabele na specificarane načine) je predstavljena preko KLM-GOMS modela. U analizi je razmatran prosečan, manje vešt korisnik (40 r/m)[1]. U zavisnosti od odabranog načina korisniku bi trebalo za umetanje tabele:

1. Preko menubar-a: 7.95s
2. Preko toolbara-a: 2.65s
3. Preko ribbon-a: 2.65s

Iz priloženog jasno se vidi da će najsporije do željene akcije doći putem *Menubar*-a, a da će mu trebati jednako vremena ukoliko za to bude koristio *Toolbar* ili *Ribbon.*

3. IMPLEMENTACIJA

Traka je osmišljena da služi kao unapređenje tradicionalnih menija i traka sa alatkama. Kao takva ona je napravljena od nekoliko komponenti:

- kartice,
- grupe,
- sekcije,
- kontrole i
- *toolbar* za brzi pristup

Glavni cilj implementiranog projekta jeste da se korisnicima pruži mogućnost da, koristeći *Ribbon* kontrolu, na osnovu predefinisanih komandi (akcija) mogu sebi da kreiraju i uređuju izgled radnog prostora po svojim preferencijama. Kako je već napomenuto, sve komande su serijalizovane u XML dokument. Podsistem za čitanje XML sadržaja pročita ovaj dokument i pruža korisniku mogućnost da pročitane komande rasporedi i grupiše po svojoj volji. Kada napravi željenu konfiguraciju, korisnik je serijalizuje nazad u XML format dokumenta i sačuva je na disku računara. Kao takva, traka se dalje koristi u aplikaciji umesto menija i traki sa alatkama.

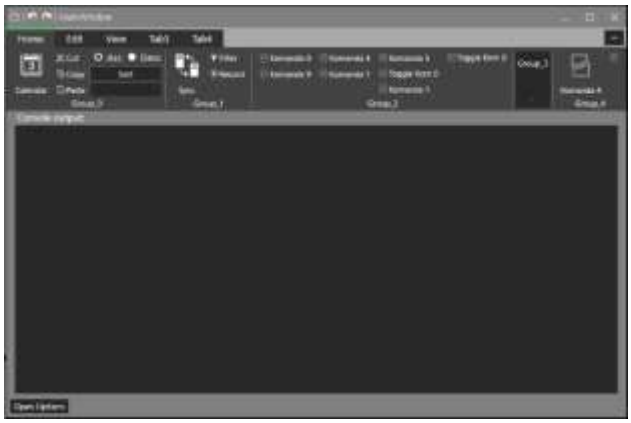
[1] Jedinica za merenje brzine kucanja. Označava broj otkucanih reči u jednom minutu.

3.1. Korisnički interfejs

Prilikom dizajniranja korisničkog interfejsa, velika pažnja je posvećena njegovom prilagođavanju tipu korisnika koji će zapravo koristiti aplikaciju. Boje, fontovi i nazivi opcija su konzistentni na svakom fragmentu korisničkog interfejsa, pružajući korisniku komforniji i brži rad. U svakom trenutku, aplikacija prikazuje optimalan skup kontrola koje su korisniku potrebne za izvršavanje željenog cilja.

Optimalan skup vidljivih komandi je kontekstno zavistan, odnosno određuje se na osnovu selektovane kartice i zadataka koji su izvršeni.

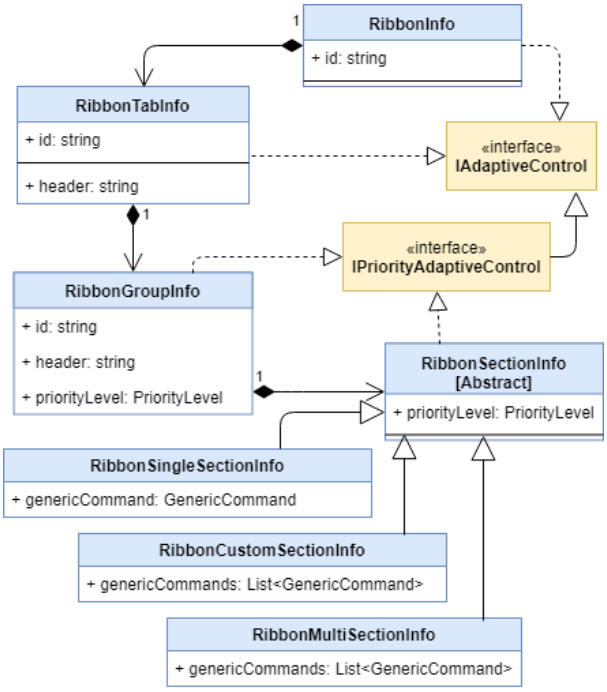
Ovaj skup je sa jedne strane dovoljno velik da zadovolji korisnikovu potrebu za obavljanjem zadatka, dok je sa druge strane dovoljno mali da korisnički interfejs ostane čist, a ne pretrpan raznolikim opcijama koje nisu relevantne u datom trenutku. Na Slici 4 prikazan je kompletan izgled glavnog prozora aplikacije.



Slika 4 *Prikaz korisničkog interfejsa aplikacije*

3.2. Model podataka

Svaka kontrola koja može biti deo glavne *Ribbon* kontrole ima i svoj odgovarajući model. Diagram klasa modela komponenti je prikazan na Slici 5.



Slika 5 *Dijagram klasa modela Ribbon aplikacije*

3.3 Implementacija kontrola

Kontrole imaju mnoge osobine koje se mogu iskoristiti za prilagođavanje njihovog izgleda. Dugme, na primer, ima opcije da promeni boju pozadine ili teksta, kartice *TabControl*-e mogu biti premeštene postavljanjem *TabStripPlacement* funkcije, i tako dalje. Ali to je sve što se može učiniti sa takvim svojstvima [1].

Šabloni (*Template*), sa druge strane, omugućavaju da se u potpunosti zameni vizuelno stablo elementa sa bilo čim što može pasti na pamet, a da sve funkcionalnosti ostanu netaknute. Šabloni, kao i mnoge druge stvari u WPF-u, nisu samo neki dodatni mehanizmi. Podrazumevani vizuelni prikazi za svaku kontrolu u WPF-u su definisani u šablonima i prilagođeni za svaku temu *Windows*-a. Izvorni kod za svaku kontrolu je potpuno odvojen od podrazumevanih vizuelnih prikaza stabla [1].

Šabloni i želja da se razdvoji vizuelni prikaz od logike su takođe razlozi da WPF kontrole ne izlažu jednostavnije osobine za podešavanje izgleda. Na primer, bilo bi lepo da postoji mogućnost da se promeni boja strelice *Expander*-a, ali siva boja se neće lepo prikazati na ljubičastoj pozadini. Ova relativno jednostvana promena može se postići samo definisanjem novog šablona za *Expander*. *On* u sebi nema definisan *ArrowBrush* ili *ArrowColor* [1].

Postoje nekoliko različitih vrsta šablona. Jedan od njih je i šablon podataka (*DataTemplate*). Šabloni podataka predstavljeni su klasom *DataTemplate* koja potiče, kao i većina šablona, od abstraktne klase *FrameworkTemplate*. Šabloni podataka prilagođavaju izgled nekog .NET objekta[2].

RibbonWindow je prozor na kom se nalazi *Ribbon* kontola. *Template* tog prozora je izmenjen kako bi se omogućile sve funkcionalnosti koje *Ribbon* ima. *Template* se sa stoji od *Border*-a koji se nalazi oko celog prozora i predstavlja deo koji omogućava promenu veličine prozora (*resize*). Unutar *ResizeBorder*-a nalaze se dva dela, *Title* deo i sadržaj tog prozora. *Title*, naslov, se nalazi na gornjem delu, kao i kod standardnog prozora. Jedini razlog zbog kog se i pravi kompletno ceo *template* prozora jeste to što *Title* deo nije moguće promeniti. Kod standarnog prozora predefinisana je pozadina *TitleBar*-a i nju dodeljuje *Windows* operativni sistem i jedino je moguće menjati tekst naslova. Zbog potrebe da se kontrola za brzi pristup nađe u *TitleBar*, potrebno je promeniti i njegov *template*. *Template* se sastoji iz ikonice prozora, kontrole za brzi pristup, naslova i dugmadi za minimizaciju, maksimizaciju i zatvaranje prozora.

QuickAccessToolbar je u osnovi *UserControl*-a koja u sebi ima *StackPanel*. *StackPanel* ima horizontalnu orijentaciju i u njemu se nalaze kontrole. Kontrole koje se mogu nalaziti su obično dugme (*Button*), isključivo dugme (*ToggleButton*) i korisnički definisana kontrola (CustomControl). *Button* i *ToggleButton* imaju posebno definisan izgled za svoju prezentaciju na *QuickAccessToolbar*-u. *CustomControl*-a će biti prikazana onako kako je i napravljena, što nekad može biti problem. Problem bi bio ukoliko je kontrola većih dimenzija nego što je prostor u koji se smešta. To bi dovelo da se određeni deo kontrole ne vidi, što gubi

smisao same kontrole. Rešenje problema će biti objašnjeno kasnije u okviru ovog poglavlja.

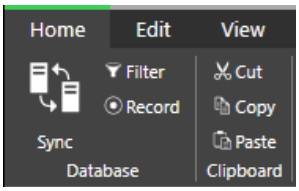
TabControl-a ima svoje kartice i svaki taj kartica ima svoj sadržaj. Taj sadržaj je *UserControl*-a koja se zove *TabContent*. *TabContent* je u osnovi *StackPanel*-a koji svoj sadržaj skladišti u horizontalnoj orijentaciji. Elementi unutar njega se nižu sa leva na desno. Ima javnu metodu *AddControl* pomoću koje se dodaju *Ribbon Group Control*-e. Metoda *AddControl* kao parametar prima *FrameworkElement* i može da primi bilo koju kontrolu koja je u osnovi *FrameworkElement*-a (kao što je i *RibbonGroupControl*).

RibbonGroupControl je takođe *UserControl*-a koja u osnovi isto ima *StackPanel* i *TextBlock*. Na isti način kao i kod *TabContol*-e, *RibbonGroupControl* ima svoju implementaciju metode *AddControl*, pomoću koje se dodaju elementi grupe. *TextBlock* se nalazi u donjem delu grupe i prikazuje naziv grupe. Grupa unutar sebe može da ima proizvoljan broj sekcija. Sekcije su *UserControl*-e i postoje tri razlicite vrste.

Vrste sekcija koje postoje: *multi*, *single* i *custom*. Svaka od njih je predstavljena odgovarajućim kontrolama. *RibbonSingleSectionControl* je kontola koja vizuelno predstavlja sekciju koja ima jednu komandu unutar sebe. U osnovi je *Grid* kontrola, jer se prikazuje samo jedan element. *Grid* podrazumevano proširi i uklopi elemente unutar sebe, i zbog toga je jako pogodan za ovakav vid prikaza. Postavljanje kontrole na sekciju je omogućeno pomoću *SetControl* metode. Kontrole koje prikazuju u *single* sekciji su ili *Button* ili *ToggleButton*, u zavisnosti da li komanda *push* ili *toggle*.

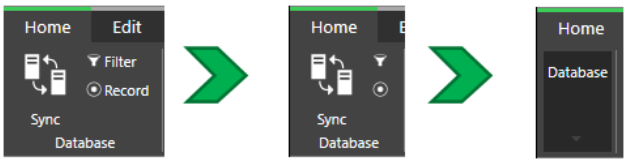
RibbonMultiSectionControl je kontrola koja može maksimalno da prikazuje do tri kontrole. Isto kao i kod *single* sekcije, kontrole mogu biti samo *Button* ili *ToggleButton*. U osnovi se nalazi *StackPanel* sa vertikalnom orijentacijom, koji niže kontrole od gore ka dole. Dodavanje omogućuje metoda *AddControl*, koja dodaje odgovarajuće kontrole u sekciju.

Button i *ToggleButton* imaju posebno definisan izgled za prikaz u odgovorajućoj sekciji. *Template* za oba dugmeta je isti, sa tim da *ToggleButton* ima mogućnost da ostane u pritisnutom stanju i da promeni boju kontrole kako bi dao indikaciju da je kontrola pritisnuta. Izgled kontrola kada se nalaze u *single* sekciji je takav da se ikonica nalazi na sredini u gornjem delu, a tekst kontrole se nalazi ispod nje. Visina takve kontrole je uvek fiksna dok se širina može menjati u zavisnosti od dužine teksta. Izgled kontrole za *multi* sekciju je takav da se ikonica nalazi sa leve strane, a pored nje sa desne strane se nalazi tekst. Visina i ovakve kontrole je fiksna i omogućava da mogu da stanu sve tri kontrole unutar sekcije. Širina je promenljiva u zavisnosti od teksta, sa tim da sve tri kontrole u sekciji zauzimaju širinu najšire kontrole. Izgled je moguće videti na Slici 6.



Slika 6 *Prikaz single i mutli sekcije sa svojim kontrolama u 2 grupe*

Adaptacija je implementirana da kontrole same brinu o tome kako će da se uklope u slobodan prostor. Prilikom skupljanja prozora u kom se *Ribbon* kontrola nalazi, kontrole se dinamički skupljaju, kolko god je to moguće, tako se uklapaju u raspoloživ prostor. Svaka kontrola od koje je sačinjen *Ribbon* ima implementiran interfejs *IAdaptiveControl.* To obezbeđuje da svaka kontrola ima implementiranu metodu za širenje i skupljanje, kao i informaciju da li je kontrola skroz skupljena ili ne. Kada kontrole dostignu maksimum svog skupljanja, i u okviru grupe ne postoji ni jedna kontrola koja može još da se skupi, cela grupa prelazi u *DropDown* kontrolu. Proces skupljanja je prikazan na Slici 7.



Slika 7. *Proces skupljanja kontrola unutar grupe*

4. ZAKLJUČAK

Projekat koji je tema ovoga rada predstavlja primenu stečenog znanja iz oblasti grafičkog korisničkog interfejsa. Posebna pažnja je posvećena dizajnu, tako da aplikacija bude dostupna i pogodna širokom spektru korisnika.

Glavni fokus ovog rada jeste na primeni stilova i šablona za pravljenje prilagođenih, nestandardnih kontrola. Ideja je bila da se što vernije napravi kontrola koja će imati sličnu namenu kao i *Microsoft*-ova *Ribbon* komandna traka. Pojedinačne kontrole unutar trake pravljene su svaka na specifičan način i prikazane su svojim šablonom. Glavna razlika, a samim tim i prednost, u odnosu na originalnu Microsoft-ovu traku jeste proizvoljna konfiguracija dugmadi prilikom rada aplikacije.

Mogućnost konfiguracije interfejsa je od posebnog značaja pre svega naprednijim korisnicima, koji mogu da prilagode svaki segment svojim potrebama i na taj način ubrzaju svoj rad. Uvođenjem ove funkcionalnosti, cela oblast dizajna se diže na jedan viši nivo i otvara čitav niz mogućnosti za dalja unapređenja.

U nekim budućim verzijama aplikacije bi bilo poželjno da postoji opcija premeštanja *DragAndDrop*-om, gde bi korisnik prevlačenjem kontrole određivao gde će kontrola da se nalazi na traci, a nova pozicija bi se automatski serijalizovala u XML dokument.

5. LITERATURA

[1] A. Nathan, WPF 4 Unleashed, Indianapolis: Sams, 2010 , pp430-432

[2] Microsoft, „C# documentation“ https://docs.microsoft.com/en-us/dotnet/csharp/

Kratka biografija:

Srđan Paunović rođen je 04.04.1994. god. u Novom Sadu. Osnovne akademske studije na Fakultetu tehničkih nauka u Novom Sadu završio je 2017. godine na smeru Primenjeno softversko inženjerstvo. Iste godine, na istom fakultetu, upisuje master akademske studije**,** smer Primenjeno softversko inženjerstvo.
Kontakt: srdjan.paunovic94@gmail.com