

RAZVOJ DESKTOP I VEB APLIKACIJE ZA PODRŠKU RADA STUDENTSKE SLUŽBE**DEVELOPMENT OF DESKTOP AND WEB APPLICATION FOR STUDENT ADMINISTRATION OFFICE**Milan Đukić, *Fakultet tehničkih nauka, Novi Sad***Oblast – Inženjerstvo informacionih sistema**

Kratak sadržaj – U okviru ovog rada je izvršeno projektovanje i razvoj desktop i veb aplikacije koje bi mogle olakšati i ubrzati proces poslovanja kao i doprineti boljoj organizaciji rada studentske službe. Prilikom implementacije aplikacija korišćene su tehnologije ASP.NET MVC i WPF koje predstavljaju sastavni deo Microsoft-ovog .NET okruženja.

Ključne reči: Baza podataka, informacioni sistemi, .NET, desktop aplikacija, veb aplikacija, MVC dizajnerski obrazac

Abstract – Task of the master thesis was the design and development of a desktop and web application that will facilitate and accelerate the business process as well as contribute to better organization of the student administration office. Implementation of the applications was achieved using ASP.NET and WPF technologies which are part of .NET framework designed by Microsoft company.

Keywords: Database, information system, ASP.NET, WPF, web applicaiton, MVC design pattern

1. UVOD

Moderno poslovanje nije moguće zamisliti bez upotrebe savremenih softvera (desktop aplikacija), a posebno razvoja veb aplikacija različitih namena. Razlog za potrebu veb aplikacija jeste potreba da bilo kada, bilo gde i u bilo kojem momentu aplikacija može konzumirati usluge preduzeća koju ono nudi.

Pored ekspanzije veb aplikacija, desktop aplikacije i dalje imaju važnu ulogu u poslovnom svetu. Iako veb aplikacije poseduju značajne prednosti, postoje pak situacije kada su desktop aplikacije nezamenljive i neophodno ih je koristiti kako bi se efikasno odvijao proces poslovanja.

Čestu potrebu vođenja evidencije, transparentnog pružanja informacija klijentima, gotovo je nemoguće obaviti u modernom svetu a da to nije u elektronskoj formi i dostupno putem interneta.

S tim u vezi, predmet ovog master rada jeste realizacija kako desktop tako i veb aplikacije na primeru rada studentske službe kao i evidencije položenih ispita.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Darko Stefanovic, vanr. prof.

U cilju efikasnijeg rada, bolje organizovanosti i pružanja usluga o evidenciji položenih ispita, kao i njihove pretrage, razvijene su dve aplikacije koje predstavljaju sistem za automatizaciju evidencije položenih ispita.

2. OPIS KORIŠĆENIH TEHNOLOGIJA

Funkcionalnosti desktop i veb aplikacije razvijeni su u WPF, odnosno ASP.NET MVC tehnologiji. Ove tehnologije biće predstavljene u nastavku rada.

2.1. .NET Framework

Microsoft .NET je softverski *framework* (okruženje) koji se instalira na operativnom sistemu *Windows*. Softver koji je razvijen korišćenjem .NET tehnologija zahteva da .NET okruženje bude instaliran na određenom sistemu kako bi mogao da funkcioniše. Instalirano .NET okruženje stvara potrebno softversko okruženje koje obezbeđuje odgovarajuće uslovi softveru tokom vremena izvršavanja (*runtime*) [1].

.NET Framework ima dve komponente [2]:

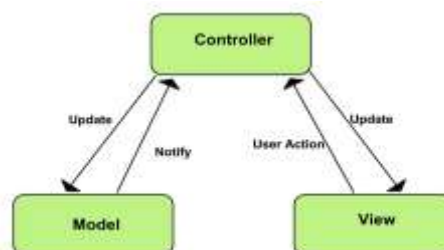
- .NET Framework class library (FCL) i
- Common Language Runtime (CLR)

2.2. ASP.NET MVC

ASP.NET predstavlja platformu u okviru .NET okruženja. ASP.NET se koristi za razvoj veb aplikacija kao i veb servisa koji se izvršavaju u IIS-u (*Internet Information Services*). ASP.NET obezbeđuje, između ostalog i visok nivo doslednosti tokom implementacije veb aplikacija i servisa [3].

MVC (*Model-View-Controller*) obrazac je jedan od najprisutnijih dizajnerskih obazaca u modernom razvoju aplikacija. Njegova filozofija je razdvajanje odgovornosti i enkapsulacija između izvršavanja logike aplikacije i njene prezentacije. U suprotnom, ukoliko se odgovornosti ne razdvoje aplikacija će biti usko spregnuta što dovodi do otežanog održavanja i proširenja u budućnosti.

Na slici 3 prikazane su komponente MVC obrasca.



Slika 3: MVC obrazac

MVC obrazac čine sledeće komponente [3]:

1. **Model** predstavlja podatke aplikacije i povezanu poslovnu logiku, koja se odnosi na ažuriranje modela. Takođe sadrži i logiku aplikacije, definišući šta se sve može uraditi sa datim podacima. Mnoge aplikacije koriste mehanizme za trajno čuvanje podataka (bazu podataka, na primer).
2. **View** komponenta prikazuje podatke u aplikaciji. Ona nije trajna, aktivna je samo dok je aplikacija pokrenuta. Ova komponenta može da sadrži logiku prikazivanja vizuelnih elemenata ali nikako ne sme da sadrži poslovnu logiku, što je zadatak kontrolne komponente.
3. **Controller** ima zadatak da inicijalizuje zahtevane podatke, da ih održava i prosledi View komponenti. Controller komponenta povezuje View i Model i određuje tok izvršavanja aplikacije. Controller interpretira input korisnika i prosleđuje ih do Model-a ili View-a. On odlučuje kako model treba da se promeni kao rezultat inputa korisnika i koji View treba da se koristi.

2.3. WPF

WPF predstavlja grafički podsistem .NET Framework-a i direktno je u vezi sa XAML-om (*eXtensible Application Markup Language*) [4]. Prvi put se pojavio sa verzijom 3.0 .NET Framework-a.

WPF omogućava programski model za razvoj aplikacija gde postoji jasna razlika između korisničkog interfejsa i poslovne logike. Ono što je značajno jeste da se može koristiti za razvoj kako desktop tako i veb softvera.

Između ostalog, WPF omogućava izuzetno bogat dizajn, kontrolu ali i razvoj kastomizovanih vizuelnih dodataka kada su u pitanju Windows programi.

WPF pretenduje da spoji u celinu kako korisnički interfejs, 3D crtanje, animacije, audio, video, tako i *data binding*.

3. OPIS FUNKCIONALNOSTI WPF i ASP.NET APLIKACIJE

U ovom poglavlju biće predstavljene bitne funkcionalnosti desktop i veb aplikacije razvijene u WPF, odnosno u ASP.NET tehnologiji koja se bazira na MVC dizajnerskom obrascu. Aplikacijama se želi omogućiti radnicima visokoškolskih ustanova u okviru studentske službe, mogućnost vođenja evidencije o studentima, profesorima, predmetima, departmanima kao i onome što je najbitnije, a to je evidencija o položenim ispitima.

U nastavku rada biće više reči o funkcionalnostima aplikacija, te će biti i slikovito predstavljena strana korisničkog interfejsa ovih aplikacija. Pošto je reč o dve aplikacije, prvo će biti predstavljena veb aplikacija sa bitnim funkcionalnostima, a potom će biti reči o funkcionalnostima desktop aplikacije koja je prvenstveno namenjena radnicima u studentskoj službi.

3.1. VEB APLIKACIJA STUDENTSKE SLUŽBE RAZVIJENA U ASP.NET MVC TEHNOLOGIJI

Veb aplikacija studentske službe jeste namenjena kako radnicima u studentskoj službi tako i samim studentima. Stranica koja se prvobitno prikazuje korisniku pri pristupanju na veb sajt jeste stranica za logovanje. Korisnik je u obavezi da unese validnu e-mail adresu kao i šifru kako bi pritiskom na dugme "Sign In" pristupio sadržaju sajta studentske službe. U slučaju da korisnik prvi put pristupa veb sajtu i nema korisnički nalog, potrebno je izvršiti registraciju korisnika.

Nakon uspešne registracije i logovanja na veb sajt, korisnik se automatski redirektuje na glavnu stranicu. Glavna stranica predstavlja inicijalni spisak evidencije studenata, profesora, departmana, predmeta, kao i položenih ispita. Svako od evidencija pojedinačno pristupa se pritiskom na dugme sa nazivom evidencije za koju se želi informisati. Na slici 1. biće slikovito prikazan korisnički interfejs za početnu stranu (*home page*).



Slika 1. Home page ASP.NET MVC aplikacije

Klikom na bilo koje ponuđeno dugme proslediće korisnika na stranicu gde će imati kompletan uvid u evidenciju selektovane opcije kao i mogućnost unosa, promene ili brisanja novih podataka ako mu je dodeljena privilegija za to.

Osnovna ideja svake stranice, pa i ove, jeste mogućnost korisnicima da izvrše *CRUD* (*create, read, update, delete*) operacije. U zavisnosti od tipa korisnika dodeljene su razne privilegije za ove operacije. Radnicima u studentskoj službi biće omogućeno da koriste sve operacije, dok će studentima biti omogućen samo uvid u podatke bez da imaju pravo modifikovati ili brisati već postojeće podatke iz evidencije.

U slučaju da radnik studentske službe želi upisati u bazu nove podatke za neku od mogućih kategorija (student, profesor, predmet, departman, ispitni rok, prijavljeni ispiti), neophodno je da klikne na dugme „Add“ te će mu se otvoriti *modal*-ni prozor u kojem će popunjavati podatke o kategoriji. Ukoliko pak neki od podataka ostanu nepopunjeni, korisniku će biti ispisano obaveštenje da je neophodno popuniti sva obavezna polja kako bi se podaci o kategoriji mogli sačuvati u bazu podataka.

3.2. DESKTOP APLIKACIJA STUDENTSKE SLUŽBE RAZVIJENA U WPF TEHNOLOGIJI

Nakon što je predstavljen rad same veb aplikacije za podršku rada studentske službe, u ovom poglavlju biće predstavljena i desktop aplikacija identičnih funkcionalnosti. Ono što jeste razlika jeste da desktop aplikacije za ciljnu grupu ima isključivo radnike u studentskoj službi, dok je veb aplikacija bila namenjana i delom za krajnje korisnike tj. studente.

Primenom ove desktop aplikacije, radnici u studentskoj službi imaju priliku da se interno upoznaju sa podacima koji su uneti preko same veb aplikacije. Razlog za to jeste što i desktop i veb aplikacija koriste jednu bazu podataka, tako da će isti podaci biti predstavljeni na obe aplikacije. Time se želi dočarati realna potreba radnicima u studentskoj službi da i u uslovima rada bez interneta mogu imati sve potrebne informacije u datom trenutku vremena.

Ukoliko se osvrnemo na tehnički deo poput korisničkog interfejsa, bitno je napomenuti da je za izradu modernog dizajna primenjen *Metro UI*.

Metro UI predstavlja besplatne skupove alata za razvoj *HTML*, *CSS* i *JavaScript*-a. *Metro UI* pomaže pri izradi cele desktop aplikacije primenom živopisnog *grid* sistema, ugrađenih i predefinisanih komponenti kao i zaista funkcionalnih dodataka baziranih na *jQuery*-u [5].

Ono što je izuzetno značajna karakteristika ove desktop aplikacije, jeste da se prilikom izrade koristio princip *Entity Framework*-a predstavljen kao „*Database-First*“, odnosno prvo je kreirana baza a potom se iz već postojeće baze generisale klase. Generisanje modela omogućeno je putem *Entity Framework Designer*-a oslanjajući se na već kreiranu bazu podataka [6].

Stranica koja se prvobitno prikazuje korisniku pri pristupanju na aplikaciju jeste stranica za logovanje. Korisnik je u obavezi da unese validnu e-mail adresu kao i šifru kako bi pritiskom na dugme „*Login*“ pristupio sadržaju aplikacije studentske službe. U slučaju da korisnik prvi put pristupa veb sajtu i nema korisnički nalog, potrebno je kreirati nalog za novog korisnika. Kreiranje naloga se vrši pritiskom na link „*Create account*“ te je potrebno popuniti potrebna polja i kliknuti na dugme „*Submit*“.

Nakon uspešne registracije i logovanja na desktop aplikaciju, korisnik se automatski redirektuje na glavnu stranicu. Glavna stranica predstavlja inicialni spisak evidencije studenata, profesora, departmana, predmeta, kao i položenih ispita. Svako od evidencija pojedinačno pristupa se pritiskom na dugme sa nazivom evidencije za koju se želi informisati. Na Slici 2. biće slikovito prikazan korisnički interfejs za početnu stranu (*home page*) sa već selektovanom opcijom evidencije profesora.

	NAME	DATE OF BIRTH	EDIT	DELETE
Subjects	Inđar Stoković	7/11/1983 12:00:00 AM	EDIT	DELETE
Students	Senja Ristić	7/10/1982 12:00:00 AM	EDIT	DELETE
Exams	Olgja Branković	5/27/1982 12:00:00 AM	EDIT	DELETE
Departments	Anđela Anđelić	6/16/1981 12:00:00 AM	EDIT	DELETE
Professors	Darko Đorđević	4/18/1974 12:00:00 AM	EDIT	DELETE

Slika 2. Home page i tabelarni prikaz evidencije profesora

Ono što karakteriše vizuelni izgled početne stranice ove desktop aplikacije jeste jednostavnost. Radnici studentske službe će imati početnu stranicu aplikacije koja će ujedno predstavljati i jedinu osnovnu stranicu pri korišćenju aplikacije.

Ono što se želi omogućiti korisniku jeste brzo, efikasno i jednostavno korišćenje i rukovođenje aplikacijom. S' obzirom da ova aplikacija predstavlja sekundarnu opciju pri podršci u radu studentske službe, želi se izbeći otežano korišćenje kao i sama kompleksnost desktop aplikacije.

Takođe, želi se omogućiti korisniku aplikacije da odmah sa početka ima otvorene sve opcije i sve mogućnosti. U startu se korisnik može odlučiti za koju evidenciju je zainteresovan. Sa leve strane nalaze se opcije za odabir željene evidencije dok je u središnjem delu tabelarni prikaz odabrane evidencije.

Sa leve strane ekrana, korisniku se daje mogućnost selektovanja evidencije:

- Predmeta (*Subjects*),
- Studenata (*Students*),
- Položenih ispita (*Exams*),
- Departmana (*Departments*) i
- Profesora (*Professors*).

Ukoliko korisnik aplikacije želi uvid u bilo koje raspoložive podatke, potrebno je kliknuti na neko od dugmića sa leve strane, te će se korisniku na sredini ekrana prezentovati tabelarni prikaz željenih podataka. Pošto su korisnici desktop aplikacije radnici u studentskoj službi, imaće mogućnost da unutar tabelarnog prikaza podataka vrše modifikaciju ili pak brisanje željenih redova u tabeli.

Kao i u već predstavljenoj veb aplikaciji korisnik će imati omogućene *CRUD* operacije sa svih validacionim pravilima. Pored operacije čitanja, korisnik aplikacije će imati mogućnosti kreiranja novih, modifikaciju postojeći kao i brisanje već kreiranih informacija o studentima, predmetima, profesorima, departmanima i ispitima.

Glavna prednost ove desktop aplikacije jeste upravo ova početna stranica jer sve informacije koje korisnik može da dobije se nalaze upravo na njoj, i time se omogućava maksimalna brzina informisanja korisnika.

4. ZAKLJUČAK

Kroz rad je prikazan razvoj desktop i veb aplikacije za podršku rada studentske službe. Obe aplikacije su imale svrhu i bile su zamišljene kao osnovni alat koji će omogućiti svojim korisnicima (radnicima studentske službe i samim studentima, kada je reč o veb aplikaciji) značajniji napredak u obavljanju svakodnevnih obaveza. Između ostalog, omogućen je uvid u celokupnu evidenciju podataka o studentima, profesorima, predmetima, ispitnim rokovima, kao i prijavljenim ispitima u veb aplikaciji, odnosno položenim ispitima ako je reč o desktop aplikaciji.

Ono što je posebno važno i što karakteriše ove aplikacije jeste jednostavan a opet moderan korisnički interfejs, lakoća upotrebe, pouzdanost pri čuvanju podataka kao i brzina odziva same aplikacije što u mnogome štedi vreme samim korisnicima. Ako se uporedi korišćenje ovih aplikacija sa dosadašnjim načinom poslovanja sa hrpom papira, jasno je da se prikupljanje i skladištenje potrebnih informacija vrši isključivo na jednom mestu, te sama centralizacija skladištenja podataka predstavlja veliku prednost.

Izuzetna organizacija posla, koja je omogućena primenom ovih aplikacija, doprinosi uvećanoj efikasnosti i efektivnosti samih korisnika tj. radnika u studentskoj službi, dok s' druge strane obezbeđuje jasne i pouzdane informacije studentima kao krajnjim korisnicima.

Ovim radom težilo se prikazati koliko je upotrebom savremenih tehnologija kao što su *WPF* i *ASP.NET MVC* arhitekture jednostavno kreirati kvalitetanu i funkcionalnu veb i desktop aplikaciju koje će doprineti i pospešiti poslovanje manjeg ili srednjeg biznisa.

4. LITERATURA

- [1] Srđan Sladojević, Mirjana Dulić, Predrag Jelovac, Ivan Edelinški, Darko Stefanovic (2015), „One solution of web application for data acquisition from remote sensing devices“, Infotech – Jahorina 2015, Republika Srpska, Jahorina, Vol. 14, str. 788-792.
- [2] Internet: Microsoft “Introduction to the C# Language and the .NET Framework”. Dostupno na: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework> [pristupljeno 03.oktobar 2018.]
- [3] Internet: Tom Dykstra, Rick Anderson, “Getting Started with Entity Framework 6 Code First using MVC 5” dostupno na: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/getting-started-with-ef-using-mvc/creating-an-entity-framework-data-model-for-an-asp-net-mvc-application>, [pristupljeno 6.oktobar 2018.]
- [4] Internet: “XAML in WPF”, Dostupno na: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/xaml-overview-wpf>, [pristupljeno 06.oktobar 2018.]
- [5] Internet: “Metro 4 – Raise Your Work to the Next Level”. Dostupno na: <https://metroui.org.ua/>, [pristupljeno 10. oktobar 2018.]
- [6] Internet: “EF Basics and EF6 Code-First”. Dostupno na: <http://www.entityframeworktutorial.net/code-first/what-is-code-first.aspx>, [pristupljeno 06.oktobar 2018.]

Kratka biografija:



Milan Đukić rođen je u Novom Sadu 1993. god. Master rad na Fakultetu tehničkih nauka iz oblasti Inženjerstva informacionih sistema odbranio je 2018.god.
kontakt: djukacho93@gmail.com