



SISTEM ZA AUTOMATIZACIJU PROCESA IZRADA LIČNIH DOKUMENATA

SYSTEM FOR AUTOMATIZATION PROCESS OF ISSUING PERSONAL DOCUMENTS

Dorđo Kalaković, *Fakultet tehničkih nauka, Novi Sad*

Oblast: INŽENJERSTVO INFORMACIONIH SISTEMA

Kratak sadržaj - U modernom dobu jedan od najbitnijih resursa postalo je vreme. Mnoge tehnologije, alati, procedure uvođeni su ne bi li se uspelo uštedeti na vremenu. Ušteda ovog resursa bila je osnovni motiv nastanka aplikacije opisane u daljem radu. Aplikacija je napisana u programskom jeziku C# i napravljena je da omogući jednostavnije apliciranje za neki od ličnih dokumenata. Korisnik aplikacije bi mogao uvek da ima uvid na vreme čekanja u nekoj od ustanova, kao i na vreme čekanja na zahtev, za koji je prethodno aplicirao.

Ključne reči – Visual Studio; Microsoft SQL Server; MVC; Twilio; WCF service; Windows service;

Abstract - In recent years, time has become our most valuable resource. Many technologies, tools and procedures have been implemented in order to save certain amount of time. The same motive has been the main catalyst behind the software application described in this paper. This software has been written in C# programming language and it is developed to simplify the process of application for obtaining personal documents. By using the application, user can have insight into waiting time in the appropriate institution, as well as the waiting time for the document request which user previously made.

Keywords – Visual Studio; Microsoft SQL Server; MVC; Twilio; WCF service; Windows service

1. UVOD

Danas je C# jedan od najpopularnijih programskih jezika. Koristi se od strane programera širom sveta. C# je razvijen od strane Microsoft kompanije, kao moderna platforma za razvoj .NET aplikacija.

C# jezik se distribuira zajedno sa *Common Language Runtime* (CLR), specijalnim okruženjem na kojem se izvršava. Ovo okruženje je deo platforme .NET Framework (eng. okvir) koji uključuje CLR, skup standardnih biblioteka, kompajler, debugger i ostale razvojne alate [1].

ASP.NET MVC predstavlja jednostavan framework za razvoj Web aplikacija u Microsoft .NET okruženju. Kada ljudi upotrebe akronim MVC najčešće misle na design pattern (eng. dizajn obrasce). Model-View-Controller je dizajn obrazac koji omogućava razdvajanje prikaza, podataka, kao i kontrole i tokove upravljanja. View predstavlja prikaz na ekranu i ono što korisnik vidi.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji je mentor dr Srđan Sladojević, docent.

Controller upravlja unesenim podacima na View. Model predstavlja strukturu podataka koja se prenosi između prethodna dva navedena. Dolazi se do zaključka da obrazac pomaže razdvajanju prikaza, interakcije i podataka.

U ASP.NET MVC obrazac je malo drugačiji. Svaki zahtev uslužuje Controller. Zatim Controller vrši neku izmenu nad modelom i aktivira View za prikaz. U ASP.NET MVC View predstavlja .aspx stranice koje sadrže HTML elemente [2]. Model mogu biti podaci iz baze podataka, što je i slučaj u aplikaciji koja se upisuje u radu.

Aplikacija ima podršku logovanja. Logovanje se startuje odmah prilikom startovanja aplikacije i kreira se na zadatoj lokaciji, a ime log fajla se kreira po sledećem obrascu {Log}{dan}{mesec}{godina}{milisekunde}. Kako bi se fajl kreirao samo jednom i sve vreme se radilo sa istim fajlom upotrebljen je i Singleton obrazac.

Singleton obrazac se koristi kada treba da postoji samo jedna instanca klase i mora biti pristupačna klijentu kroz poznatu tačku pristupa. Takođe, obrazac se koristi kada instanca klase trebalo da bude proširiva, a klijenti bi trebali biti u mogućnosti da koriste proširenu instancu bez izmene sopstvenog koda [3].

2. PRIMENJENE SOFTVERSKJE TEHNOLOGIJE

U radu su se koristile sledeće tehnologije:

.Net razvojni okvir predstavlja razvojnu programsku platformu opšte namene. Microsoft kompanija je započela razvoj ove platforme krajem 1990-te godine, a prva beta verzija .Net 1.0 je puštena krajem 2000-te godine. Od početka razvoja, pa sve do danas Microsoft je usavršavao i otpuštao različite verzije .Net framework-a. Ovaj framework se koristi isključivo na Windows operativnim sistemima, a minimalna verzija je Windows 7 Service Pack 1. Glavne karakteristike .Net framework-a su da podržava više programskih jezika, asinhrono i konkurentno programiranje modela i interoperabilnost koja omogućava širok spektar scenarija kroz više platformi [4].

Microsoft SQL Server je sistem za upravljanje relacionim bazama podataka u vlasništvu kompanije Microsoft. Kao server za baze podataka, ovo je softverski proizvod sa primarnom namenom za skladištenje i konzumiranje podataka preko zahteva, najčešće upućenih od strane drugih aplikacija, koje se mogu nalaziti na istom računaru ili na drugom računaru, kada se interakcija odvija putem Interneta. SQL Server Management Studio je alat za korisničku interakciju sa Microsoft SQL Server-om. Služi za konfigurisanje, upravljanje i administraciju svih komponenti unutar Microsoft SQL Server-a. Ovaj alat ima ugrađen editor za skripte i grafičke alate koji rade sa objektima i osobinama koje nudi server.

Windows Communication Foundation (WCF) predstavlja deo .NET softverskog okruženja koji je namenjen za razvoj i upravljanje servisima. WCF je u osnovi Microsoft-ova implementacija skupa industrijskih standarda koji definišu interakciju između servisa, konverziju tipova, kao i upravljanje različitim protokolima što omogućava WCF-u da podrži interoperabilnost između servisa. Kako bi WCF servis bio funkcionalan i pristupačan neophodno je da bude hostovan u odgovarajućem okruženju, koje je najčešće realizovano u okviru nekog procesa [5].

Windows service predstavlja osnovnu komponentu Microsoft Windows operativnog sistema i omogućava kreiranje i upravljanje dugotrajnim procesima. Za razliku od običnih programa koji su pokrenuti od strane korisnika i jedino se izvršavaju kada je korisnik ulogovan, Windows servisi se mogu startovati bez intervencije korisnika i mogu trajati mnogo duže nakon što se korisnik izloguje. Servis radi u pozadini i obično se startuju kada se mašina pokrene. Usluge upravljaju širokom paletom funkcija uključujući mrežu, zvukove, boje, praviti rezervu podataka i mnoge druge [6].

U aplikaciji windows servis služi za slanje sms poruka ukoliko vreme čekanja na neki zahtev bude manje od 20 minuta. Servis će iz baze pretražiti vreme apliciranja svakog neizvršenog zahteva, uporediti da li je vreme čekanja na zadati zahtev manje od 20 minuta. Ukoliko je vreme čekanja manje od 20 minuta servis će poslati poruku obaveštenja na broj telefona podnosioca zahteva. U bazi podataka će se polje *IsNotify* podesiti na true, kako taj broj više ne bi bio obaveštavan. Servis nakon toga pravi minut pauze.

Twilio predstavlja razvojnu platformu za komuniciranje. Softverski timovi koriste Twilio za dodavanje zvuka, videa, razmene poruka u njihovim aplikacijama. U pozadini Twilio Api-a je Super Network, softverski sloj koji povezuje i optimizuje komunikacione mreže širom sveta. Ovo omogućava korisnicima da pouzdano pozovu i šalju poruke bilo gde u svetu [7].

Kako bi iskoristili pogodnosti slanja poruka, potrebno je prvo registrovati se na sajtu ove platforme. Nakon registracije se unosi broj telefona, na koji će stići kod za potvrdu. Potrebno je podesiti zemlju u kojoj se dati broj nalazi kao i pozivni broj. Dobija se broj koji će se koristiti u aplikaciji kao broj telefona, API i token. API i token izgledaju kao *hash* (mešavina), a služe kako bi se aplikacija povezala sa profilom koji je registrovan i kreiran na sajtu.

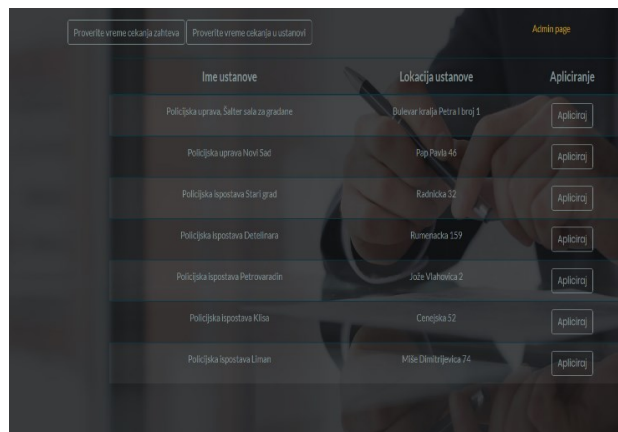
3. FUNKCIONALNOSTI

Osnovna funkcionalnost aplikacije jeste da korisnik preko Web-a aplicira za izradu nekog od ličnih dokumenata u određenoj ustanovi. Na početnoj stranici korisniku su navedene sve ustanove u kojima može da aplicira. Korisnik ima mogućnost da vidi tabelu u kojoj je navedeno ime ustanove kao i njena lokacija. Poslednja kolona sadrži dugme koje će klikom korisnika odvesti na stranicu za dodavanje zahteva.

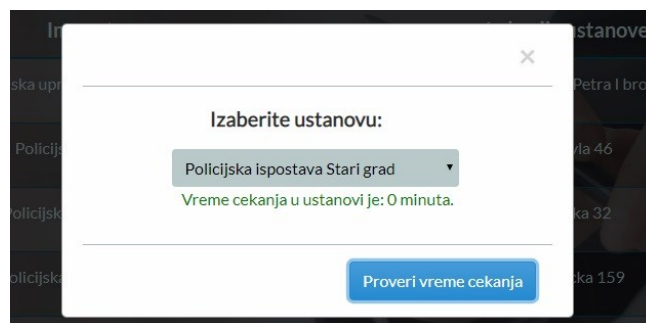
Početna stranica se može videti na slici 1. Pre nego što kreira zahtev korisnik može da proveri da li u ustanovi, u kojoj želi da aplicira ima gužve i da li postoji određeno

čekanje. Ovim se želelo postići da korisnik ne dođe u situaciju da aplicira a da je pritom odmah sledeći na redu. Na početnoj stranici postoji dugme sa nazivom „proverite vreme čekanja u ustanovi”.

Klikom na ovo dugme pojavice se modal na kom korisnik ima mogućnost da izabere ustanovu u kojoj želi da proveri vreme čekanja. Klikom na dugme „proveri vreme čekanja” pojavice se labela sa ispisom vremena čekanja. Prikaz na slici 2.



Slika 1: Index stranica



Slika 2: Provera vremena ;ekanja u ustanovi

Kada korisnik izabere ustanovu u kojoj želi da aplicira prikazaće se stranica za dodavanje zahteva. Korisnik popunjava svoje lične podatke na formi. Kada korisnik pritisne dugme za kreiranje zahteva automatski će se u bazu podataka, u posebnu tabelu dodati i korisnik sa svojim podacima, gde će JMBG biti identifikator. Identifikator zahteva biće identifikator izabranog dokumenta kao i JMBG korisnika.

Ovim se želelo postići da jedan korisnik ne može da aplicira za jedan dokument više puta. Uvek postoji mogućnost gubitka nekog od dokumenata tako da korisnik ima pravo da aplicira na neki dokument ponovo, ali tek nakon što je aplicirani zahtev izvršen. U svakom trenutku, pre nego što kreira zahtev, korisnik može da se vrati na prethodnu stranicu.

Obavezna polja koja korisnik treba da popuni su obojena crvenom bojom. Ukoliko ne unese neki podatak, ili neki unos nije validan, prikazaće se određene validacione poruke. Slika forme za dodavanje zahteva nalazi se na slici 3. Na slici 4. su prikazane validacione poruke.

Slika 3: Forma za dodavanje zahteva

Slika 4: Validacija na formi za dodavanje zahteva

Nakon što je zahtev kreiran, na broj telefona i na email, ukoliko ga je korisnik popunio, treba da stigne poruka obavještenja. U poruci se nalazi identifikacioni broj zahteva, a poruka sadrži i informacije gde je aplicirano i za koji dokument. Primer poruke koja je u formatu sms poruke se nalazi na slici 5.

Postovani Djordjo,
 Uspesno ste kreirali zahtev u ustanovi SUP Zmaj Ognjena Vuka, za izradu dokumenta: Vozacka dozvola. Identifikacioni broj Vaseg zahteva je [23757171](#). Na sajtu mozete proveriti potencijalno vreme cekanja za Vas zahtev. Srdacan pozdrav, Adminstracija

Slika 5: SMS obavještenje za kreiran zahtev

Nakon dobijanja identifikacionog zahteva korisnik u svakom trenutku može da ima uvid u vreme čekanja na dati zahtev. Na početnoj stranici se nalazi dugme pod nazivom „proverite vreme čekanja zahteva”. Klikom na pomenuto dugme pojavljuje se modal gde korisnik ima mogućnost da unese identifikacioni broj zahteva. Validan unos zahteva će prikazati labelu sa ispisom vremena čekanja na zahtev kao i broj osoba na čekanju. Nevalidan unos će prikazati validacionu poruku da je potrebno uneti postojeći zahtev. Unos identifikacionog broja zahteva koji je izvršen će prikazati poruku da je zahtev već izvršen. Prikaz validne poruke na slici 6.

Slika 6: Prikaz vremena čekanja na zahtev

Aplikacija je kreirana kako bi pružila podršku i ustanovama za obradu zahteva. Na početnoj stranici se nalazi dugme sa nazivom „admin page”. Klikom na pomenuto dugme pojavljuje se modal koji služi za login (eng. prijavu).

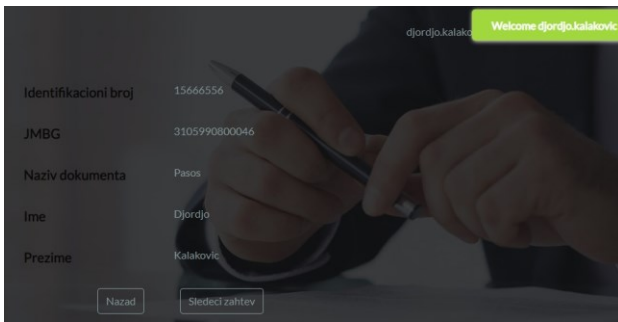
Slika modala za prijavu može se videti na slici 7. Korisnik unosi korisničko ime i šifru nakon čeka potvrđuje unos. Nakon toga se prikazuje stranica za obradu zahteva. Korisnik je prijavljen kao radnik u samo jednoj ustanovi i može da obrađuje zahteve samo iz te ustanove.

U desnom uglu je prikazana notifikacija dobrodošlice prijavljenom korisniku. Notifikacija se isključuje klikom bilo gde na formi. Istog momenta kako se korisnik prijavio tada je izvršen (promenjen status u bazi podataka na izvršen) prvi zahtev koji je bio na čekanju. Informacije o zahtevu se nalaze na formi. Forma za obradu zahteva se može videti na slici 8.

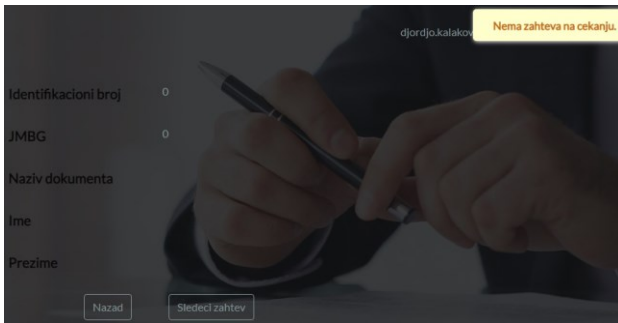
Ukoliko nema zahteva prikazaće se poruka da nema zahteva na čekanju. Dugme na kome je pisalo „admin page” nakon prijave ima prikazano ime prijavljenog zaposlenog, sve dok se isti ne odjavi. Odjavljivanje se vrši tako što korisnik klikne na dugme „logout” na stranici za obradu zahteva.

Nakon toga biće preusmeren na početnu stranicu. Forma za obradu zahteva, gde je ispisano da nema zahteva na čekanju, se može videti na slici 9.

Slika 7: Prijava za obradu zahteva



Slika 8: Prikaz informacija o zahtevu na formi



Slika 9: Prikaz stranice za obradu zahteva kada nema zahteva na cekanju

4. ZAKLJUČAK

Svima se bar jednom desilo da se našao u gužvama prilikom vađenja ličnih dokumenata. Aplikacija može biti alat putem kog ćemo ovakve probleme rešiti. Sličan alat postoji, gde je moguće zakazati izradu nekog dokumenta nekoliko dana unapred, ali smatram da je ovim softverskim rešenjem obuhvaćena veća celina. Aplikacija pruža dosta benefita. Od prikaza koliki su redovi čekanja u određenim ustanovama, apliciranja za određena dokumenta u ustanovama, do toga da pruža podršku zaposlenima u ustanovama, kako bi izvršavali zahteve po redu, i imali informacije o zahtevu.

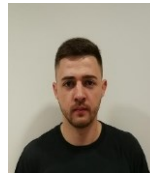
Prilikom prijavljivanja zaposlenog u ustanovi pristupa se stranici za obradu zahteva. Ovo znači da zaposleni može od kuće, tačnije van posla, pristupiti sajtu i obraditi zahtev. Ovo predstavlja jedan od nedostataka aplikacije. Potrebno je uvesti jedan nivo verifikacije, gde će se proveravati sa koje lokacije se zaposleni prijavljuje, kako ne bi bilo dozvoljeno obraditi zahtev van radnog mesta. Svakim pristupom stranici za obradu zahteva obrađuje se novi zahtev koji je na redu. Može se desiti da zaposleni nekada ne želi odmah nakon prijavljivanja startovati obradu zahteva, tako da i ovo predstavlja polje gde bi moglo da se izvrši unapređenje.

Ideja izbegavanja čekanja u redovima se može primeniti i na neke druge grane. Aplikaciju je moguće unaprediti tako da se može aplicati u zdravstvu, kako bi se izbegla ogromna čekanja kod lekara. Veliki redovi su problem u katastru, kao i u nekim drugim ustanovama. Primenom aplikacije se u navedenim institucijama mogu rešiti pomenuti problemi.

5. LITERATURA

- [1] Svetlin Nakov, Veselin Kolev & Co. (2013, Sofia), "Fundamentals of computer programming with C#".
- [2] Jonathan McCracken, The Pragmatic Bookshelf Raleigh „Test-Drive ASP.NET MVC", North Carolina Dallas, Texas.
- [3] Erich Gamma, Richard Helm, Ralph Johnson, John Vissedes, "Design Patterns Elements of Reusable Object-Oriented Software".
- [4] Sempf B., Sphar C., Davis S.R., (2013), "C# 5.0 All - In-One For Dummies", John Wiley & Sons, Inc., Hoboken, NJ, USA.
- [5] Stopper, Markus, and Bernd Gastermann. "Service-oriented communication concept based on WCF .NET for industrial applications." International multi conference of engineers and computer scientists. 2010.
- [6] Stackify , What are Windows Services? How Windows Services Work, Examples, Tutorials and More, dostupno na: <https://stackify.com/what-are-windows-services> [pristupljeno 07. decembar 2018]
- [7] Twilio, dostupno na: <https://www.twilio.com/learn/twilio-101/what-is-twilio> [datum pristupa 07.12.2018]

Kratka biografija:



Dordo Kalaković rođen je 1990. godine u Novom Sadu. Srednju tehničku školu "Mileva Marić" završava 2009. godine, nakon čega svoje obrazovanje nastavlja na Fakultetu tehničkih nauka u Novom Sadu, smer Inženjerski menadžment. Nakon završenih osnovnih akademskih studija 2015. godine, upisuje master akademske studije, program Inženjerstvo informacionih sistema.