Univerzitet u Novom Sadu 

Tehnički fakultet „Mihajlo Pupin“

Zrenjanin

**Predmet: Projektovanje softvera**

**Tema: Naručivanje hrane**

Profesor: Doc. dr. Eleonora Brtka Student: Mirko Popović SI 21/21 Asistent: MSc Siniša Mihajlović

Sadržaj

[1. Predmetni cilj 3](#_Toc136176517)

[2. Uvod 4](#_Toc136176519)

[3. Opis aplikacije 5](#_Toc136176520)

[4. Zaključak 11](#_Toc136176521)

[5. Literatura 12](#_Toc136176522)

# Predmetni cilj

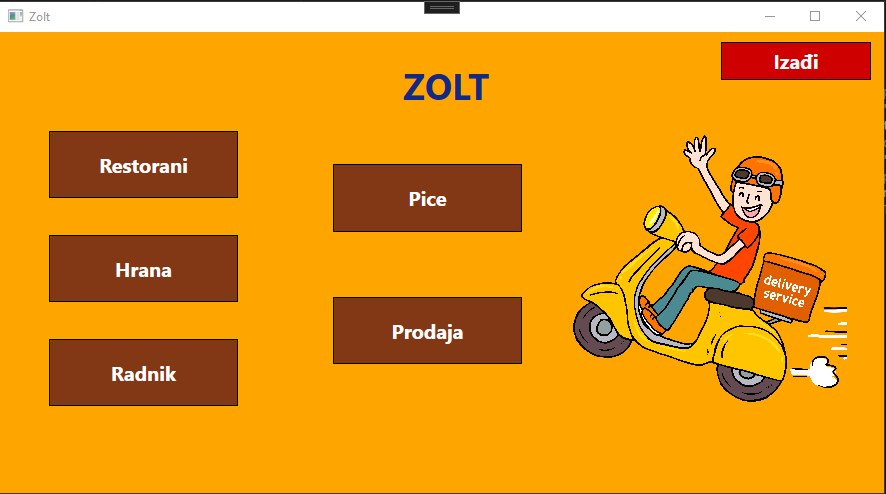
# Cilj predmeta Projektovanje softvera je da student stekne znanja neophodna za kreiranje efikasnog softvera koji uključuje arhitekturu softvera, implementaciju i testiranje kao i za kreiranje dokumentacije za softverske sisteme. Student stečena znanja treba da iskoristi za samostalnu konstrukciju softvera koja je bazirana na standardnim procesima implementacije I testiranja kao i za kreiranje projektne dokumentacije.

# 2. Uvod

U današnjem digitalnom dobu, brza i efikasna usluga postala je ključni faktor za uspeh u industriji ugostiteljstva. Kako bi se olakšalo naručivanje hrane i efikasno upravljalo restoranom, sve veći broj ugostiteljskih objekata koristi aplikacije za naručivanje hrane i vođenje restorana. U ovom seminarskom radu fokusiraću se na razvoj takve aplikacije koristeći C# programski jezik i WPF (Windows Presentation Foundation) tehnologiju u okviru Visual Studio Community razvojnog okruženja. Glavni cilj ovog rada je razviti kompletnu aplikaciju koja omogućava korisnicima da jednostavno i brzo naruče hranu iz restorana, kao i da se restoran efikasno vodi i upravlja. Za izradu korisničkog interfejsa korišćena je WPF tehnologija koja pruža bogat skup alata za kreiranje modernih i atraktivnih korisničkih interfejsa. Jedna od ključnih komponenti ove aplikacije je integracija sa SQL bazom podataka. Baza podataka se koristi za skladištenje informacija o radnicima, restoranima, hrani, piću i prodaji. Ovakav pristup omogućava efikasno upravljanje podacima i brz pristup relevantnim informacijama. Aplikacija ima glavni prozor koji pruža pregled svih funkcionalnosti sistema. Pored toga, implementirani su i prozori za unos, izmenu i brisanje radnika, restorana, hrane i pića. Takođe, postoji prozor za prodaju koji omogućava brzo i efikasno evidentiranje narudžbi i praćenje stanja prodaje. Jedna od ključnih karakteristika ove aplikacije je mogućnost povratka na prethodni ekran, što korisnicima omogućava jednostavno navigiranje kroz sistem. Takođe, implementirana je funkcionalnost brisanja teksta iz textbox-ova, čime se olakšava unos i izmena podataka. U nastavku ovog seminarskog rada, detaljnije ćemo opisati arhitekturu i funkcionalnosti aplikacije za naručivanje hrane i vođenje restorana, kao i postupak razvoja i implementacije u C# koristeći WPF tehnologiju.

# 3. Opis aplikacije

Pokretanjem aplikacije vidimo glavni prozor za dostupnim opcijama, u gornjem desnom uglu vidimo opciju da izađemo iz programa kao I opcije za hranu, radnike, piće…



Slika 1 – Glavni prozor aplikacija, tj. Početna stranica

Kada koristimo Visual Studio naša WPF svaki prozor se sastoji iz dva koda, .xaml gde mi sređujemo korisnički interfejs aplikacije I xaml.cs gde se dešava sva “magija” tj. Gde se nalazi kod aplikacije koji omogućava funkcionalnosti.

|  |
| --- |
| <Window x:Class="NarucivanjeHrane.MainWindow"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"  xmlns:local="clr-namespace:NarucivanjeHrane"  mc:Ignorable="d"  Title="Zolt" Height="500" Width="900" Background="White">  <Grid Background="Orange">  <Grid.ColumnDefinitions>  <ColumnDefinition Width="45\*"/>  <ColumnDefinition Width="16\*"/>    <Button x:Name="BtnHrana" Content="Hrana" HorizontalAlignment="Left" Height="67" Margin="49,203,0,0" VerticalAlignment="Top" Width="189" Background="#FF833815" FontSize="20" Foreground="White" FontWeight="Bold" Click="BtnHrana\_Click" Cursor="Hand" BorderBrush="Black"/>  </Grid></Window> |

U primeru koda možemo praktično videti kako .xaml izgleda za većinu prozora, kada se piše kod za prozore sve do dugmeta će biti isto za sve prozore, sem naravno naziva, i nekih željenih dekoracija. Sada stižemo do samih funkcionalnosti, jer naravno nema poente kada nešto lepo izgleda u interfejsu ako ne radi.

|  |
| --- |
| **using** **System.Windows.Data**;  **using** **System.Windows.Documents**;  **using** **System.Windows.Input**;  **using** **System.Windows.Media**;  **using** **System.Windows.Media.Imaging**;  **using** **System.Windows.Navigation**;  **using** **System.Windows.Shapes**;  **namespace** **NarucivanjeHrane**  {  **public** **partial** **class** **MainWindow** : Window  {  **public** MainWindow()  {  InitializeComponent();  }  **private** **void** BtnRadnja\_Click(object sender, RoutedEventArgs e)  {  Radnja objRadnja = **new** Radnja();  Visibility = Visibility.Hidden;  objRadnja.Show();  }  **private** **void** BtnHrana\_Click(object sender, RoutedEventArgs e)  {  Hrana objHrana = **new** Hrana();  Visibility = Visibility.Hidden;  objHrana.Show();  } |

Kod u tabeli predstavlja ono što se dešava iza intefejsa, bitno je reći da svaki .xaml.cs kod mora imati deo gde se navode svi „namespace-ovi“ koje ćemo koristiti upotrebom „using“ direktive, ovo radimo kako ne bi morali npr. Da pišemo isti kod više puta. Kada kliknemo na dugme za restoran otvara se novi prozor.

A screenshot of a computer

Description automatically generated with medium confidence

Slika 2 – Prozor za radnje, tj. Restorane

Kliknom na dugme “Restorani” otvara se novi prozor gde možemo unositi podatke za naš restoran, ili za više njih ako je u pitanju neki lanac restorana. U principu svi prozori funkcionišu na isti način gde mi možem unositi, menjati I brisati podatke iz naše baze podataka. Baze se povezuje preko konekcionog stringa koji se zapisuje u kodu:

|  |
| --- |
| SqlConnection sqlCon = **new** SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=NarucivanjeHrane;Integrated Security=True"); |

Bitno je napomenuti da se konekcioni string prvo treba kreirati u app.config, takođe se treba ustanoviti konekcija sa našom bazom preko View > Server Explorer > Add Connection. Kada je baza povezana I poželimo da radimo sa podacima u bazi korisno je da vidimo na čemu smo. To radimo preko ubacivanja Data Grida u naš .xaml kod.

|  |
| --- |
| <DataGrid x:Name="GridRadnja" HorizontalAlignment="Left" Height="320" Margin="419,101,0,0" VerticalAlignment="Top" Width="456" Opacity="0.8" Grid.ColumnSpan="2"/> |

Konačno dolazimo do ubacivanja podataka, za primer ćemo koristiti prozor za hranu.

A screenshot of a phone

Description automatically generated with low confidence

Slika 3 – Prozor za hranu

Nakon što otvorimo prozor možemo unositi željene podatke.

A screenshot of a computer

Description automatically generated with medium confidence

Slika 4 – Unos podataka

Uneti podaci se beleže u našu bazu podataka. Ukoliko poželimo mi te podatke možemo menjati ili brisati:

A screenshot of a computer

Description automatically generated with medium confidence

Slika 5 – Brisanje Podataka u Bazi

Kako bi se lakše razumela baza podataka, najbolje bi je bilo prikazati u vidu dijagrama:

A screenshot of a computer

Description automatically generated with medium confidence

Slika 6 – Dijagram SQL baze podataka

Zadnji prozor koji je vredan spomena je prozor za prodaju.

A screenshot of a computer

Description automatically generated with medium confidence

Slika 7 – Prozor za prodaju

Ovaj prozor funkcioniše na isti princip kao I prošli prozori sa tim što ovde možemo zapisati predmete koje smo prodali. Kada su podaci za prodaju uneti možemo ih lako proveriti u bazi podataka preko SQL Server Management Studija

A screenshot of a computer

Description automatically generated with medium confidence

Slika – Provera Prodaje U Bazi

# 4. Zaključak

U ovom seminarskom radu sam se fokusirao na razvoj aplikacije za naručivanje hrane i vođenje restorana koristeći C# programski jezik i WPF tehnologiju. Ova aplikacija pruža mogućnost korisnicima da jednostavno i brzo prate porudžbine iz restorana, kao i da restoranom efikasno upravljaju. Integracija sa SQL bazom podataka nam je omogućila efikasno upravljanje podacima, čime smo omogućili brz pristup relevantnim informacijama o radnicima, restoranima, hrani, piću i prodaji. Aplikacija za naručivanje hrane i vođenje restorana ima potencijal da unapredi rad ugostiteljskih objekata i pruži korisnicima brzu i efikasnu uslugu. Daljnji razvoj aplikacije može uključivati dodatne funkcionalnosti poput online plaćanja, praćenja porudžbina i generisanja izveštaja, čime bi se dodatno unapredila upravljanje restoranom. U zaključku, razvoj ove aplikacije predstavlja način da se restoranima olakša I modernizuje način poslovanja. C# i WPF tehnologija su se pokazali kao pouzdani alati u razvoju ove vrste aplikacija, a njihova primena pruža mogućnost daljeg unapređenja i prilagođavanja zahtevima tržišta.

# 5. Literatura

# http://www.tfzr.uns.ac.rs/Predmet/Projektovanje%20softvera/--1 – [Zvanicna literatura za predmet]

# 